Universität Zürich[UZH]

Geographisches Institut

Ecological impact of charcoal production on Miombo woodlands and the role of governance

ESS 511 Master's Thesis

Author: David Wick, 18-934-455

Supervised by: Prof. Dr. Maria J. Santos

Faculty representative: Prof. Dr. Maria J. Santos

24.10.2025

Table of Content

Figures

Tables

1 Abstract

Charcoal production links rural livelihoods, urban energy demand, and forest ecosystems, demonstrating the complex feedbacks that shape tropical drylands. In Tanzania's Miombo woodlands, these feedbacks determine whether the system remains resilient or shifts toward degradation. This study combines functional trait analysis and spectral variability assessment to examine how charcoal production and governance influence the ecological structure and resilience of Miombo woodlands. Plant functional trait data from the TRY database were linked with Sentinel-2 satellite imagery to connect species-level characteristics with landscape-level vegetation composition and condition. Results show that species used for charcoal production occupy a distinct region of trait space, characterized by conservative strategies such as high wood density and low specific leaf area. These traits confer structural stability but may limit recovery potential, indicating selective pressure that shapes ecosystem resilience. At the landscape scale, spectral metrics revealed that community-managed forests exhibited weaker signals of degradation under charcoal extraction than open-access areas. Governance therefore exerts a measurable effect on the ecological expression of disturbance. By linking species traits with landscape patterns, this study shows that resilience in the Miombo woodlands depends not only on what species are used but on how forest use is governed. Strengthening community-based management can align charcoal production with long-term ecosystem stability.

2 Introduction

Charcoal production represents a regionally significant but often overlooked component of the global energy and land-use transition. It supplies energy to more than two billion people worldwide and remains the dominant household fuel across sub-Saharan Africa, meeting up to 80 % of urban energy demand (FAO, 2017; Bailis et al., 2015). While vital for livelihoods and energy security, the charcoal economy is also a major driver of woodland degradation, carbon emissions, and biodiversity loss (Chidumayo and Gumbo, 2013; Ribeiro et al., 2020). Nowhere are these trade-offs more visible than in Tanzania, where charcoal underpins both rural economies and urban energy supply (Malimbwi et al., 2005; van ’t Veen, 2022). Understanding how charcoal production interacts with forest ecosystems and governance systems is therefore crucial for developing sustainable management pathways within the Miombo woodlands.

2.1 From global trends to local realities: The charcoal context

Humans have always depended on nature to sustain their livelihoods and meet daily needs (Ostrom, 2009; Scoones, 1998). Across the world, societies and ecosystems form interconnected social–ecological systems (Colding and Barthel, 2019), where the sustainable use of natural resources depends on maintaining a balance between exploitation and regeneration (Ostrom, 2009). Yet this balance is increasingly challenged by population growth, urbanization, and rising consumption (Venter et al., 2016; Wackernagel et al., 2021).

One example of this challenge lies in the global demand for energy. As fossil fuel dependence drives climate change, renewable energy sources are expanding rapidly. Among them, biomass-based fuels such as wood and crop residues remain the most widespread renewable energy source, providing about 9% of global primary energy and accounts for 55% of wood harvest worldwide (Bailis et al., 2015) 2015). For 2.6 billion people, mainly in the Global South, wood fuels remain the primary source of household energy (Guta et al., 2022; Smith et al., 2019).

Charcoal is not a major contributor to global energy production, yet it plays an vast role in regional energy systems, especially in sub-Saharan Africa, where it supports urban energy security and rural livelihoods (Chidumayo and Gumbo, 2013; FAO, 2017). Historical evidence shows that transitions from traditional biomass to modern energy sources have typically spanned 80–120 years, depending on economic development and policy support (Fouquet, 2010). Howerver, rapid substitution with fossil-based alternatives may increase dependency on external energy sources and fail to reduce emissions in the long term (Harfoot et al., 2018). On top of that, charcoal production can be locally managed, economically accessible, and socially embedded, reducing external dependencies and strengthening

local economies (FAO, 2017; Schure et al., 2014; van 't Veen, 2022). If forests are managed sustainably, woodfuel systems can approach carbon neutrality or even achieve net carbon gains. Studies show that when harvesting cycles allow full regrowth, Miombo woodlands can recover 2–5 Mg C ha^{-1} yr^{-1}, offsetting combustion emissions within 20–30 years (Chidumayo and Gumbo, 2013; Ribeiro et al., 2020). Life-cycle analyses further indicate that sustainably sourced charcoal can yield near-zero or slightly negative net emissions, ranging between –0.2 and +0.1 t CO_2e per MWh (Hektor et al., 2016). This highlights the need to improve and transition charcoal systems toward sustainability, rather than abandoning them altogether. Charcoal is not inherently unsustainable; its impact depends on how production is managed and governed. While sustainably harvested woodfuel systems can be carbon-neutral or potentially net carbon negative under long regrowth cycles (Chidumayo and Gumbo, 2013; Hektor et al., 2016; Ribeiro et al., 2020), rapid urbanization and rising demand are increasingly straining this balance. Tanzania's urban population grows by nearly 5% per year, and charcoal demand is projected to double by 2030 relaive to 2010 levels (Bailis et al., 2015). As extraction scales up to meet this demand, weak regulation and shortened harvest cycles risk exceeding natural regeneration rates, leading to progressive woodland degradation (Ahrends et al., 2010; Syampungani et al., 2009).

2.2 Charcoal in a global context

Charcoal plays vastly different roles across the world's regions, shaped by local economic conditions, energy infrastructure, cultural practices, and industrial needs. While it serves as a primary cooking fuel in some regions, it functions as a recreational commodity or industrial input in others. Globally, charcoal provides cooking fuel for hundreds of millions of people and direct or secondary income for over 40 million producers (FAO, 2017). For 2.6 billion people, mainly in the Global South woodfuels remain the main source of household energy (Guta et al., 2022; Smith et al., 2019). Beyond households, charcoal is used in small-scale industries and metal processing, such as the pig iron industry in Brazil (Morello, 2015), and continues to play a vital role in urban energy systems.

Despite its local production and use, charcoal is a globally significant energy commodity. Its production accounts for roughly 7% of global deforestation and contributes to forest degradation and biodiversity loss (Ahrends et al., 2010; Chidumayo and Gumbo, 2013). Charcoal-related emissions are estimated at 71 million tons of CO_2 and 1.3 million tons of CH_4 annually, most of which arise from inefficient carbonization rather than use (Chidumayo and Gumbo, 2013).

Charcoal demand is closely tied to processes of urbanization and demographic change. As cities expand, especially in developing regions, urban households increasingly depend on charcoal because it is accessible, transportable, and energy-dense, offering a practical alternative to firewood in densely populated environments (Santos et al., 2017). In many cases, charcoal use persists not because of a

lack of modern energy options, but because of its convenience, affordability, and cultural familiarity (Santos et al., 2017). Globally, rising urban populations have therefore become the main driver of increasing charcoal demand, particularly in Sub-Saharan Africa, where both urbanization rates and population growth are projected to be among the highest in the world (Santos et al., 2017). Urban centers act as consumption hubs that draw production outward, generating spatial waves of forest degradation and deforestation around major cities (Ahrends et al., 2010; Zorrilla-Miras et al., 2018). This urban demand dynamic links local forest use to broader socioeconomic transitions, reinforcing the importance of understanding charcoal not only as a rural livelihood issue, but as an urban–rural energy system embedded in global change.

At the governance level, charcoal systems operate under contrasting management regimes and reform pathways across regions. In parts of Asia and South America, production has become increasingly commercialized, often linked to private plantations or industrial supply chains (Piketty et al., 2009). In contrast, in much of Sub-Saharan Africa, production remains largely informal and locally organized, taking place in open-access or communal forests where regulatory enforcement is limited (Schure et al., 2014; van ’t Veen, 2022). In recent decades, global efforts to make charcoal production more sustainable have targeted all stages of this governance spectrum. These include technological interventions, such as the introduction of improved kilns to enhance conversion efficiency (Bailis et al., 2015), as well as institutional reforms like licensing schemes, charcoal bans, and community-based forest management initiatives (FAO, 2017; Kamwilu et al., 2021). Other strategies address consumption, promoting efficient stoves or alternative fuels to reduce demand (Broto et al., 2018; Kojima, 2011). The effectiveness of these interventions varies widely, depending on social, political, and economic contexts, and transitions toward sustainable charcoal systems remain non-linear and region-specific (Branch and Martiniello, 2018).

Overall, charcoal represents both a development necessity and an environmental challenge. Its global context reveals how deeply energy demand, forest governance, and livelihoods are linked. Understanding these interconnections—particularly how governance mediates ecological outcomes—is essential for supporting sustainable transitions. This becomes especially relevant in the African context, where charcoal remains the dominant household energy source and a cornerstone of rural livelihoods.

2.2.1 Charcoal in africa

Africa is both the largest producer and consumer of charcoal globally, accounting for more than half of the world’s total production (FAO, 2017). Charcoal remains the dominant household energy source across much of the continent, supporting daily cooking and heating for both urban and rural

populations (FAO, 2017). Its persistence reflects the intersection of energy poverty, population growth, and rapid urbanization, which together sustain high and rising demand (Santos et al., 2017). Access to alternatives such as electricity and gas remains limited and uneven, making charcoal the most affordable, accessible, and reliable option for most households (Guta et al., 2022; Smith et al., 2019).

Charcoal demand is closely tied to urban expansion. As cities grow, charcoal serves as a convenient and transportable fuel for urban households (Santos et al., 2017). Major urban centers such as Dar es Salaam and Maputo act as consumption hubs, drawing production outward and creating waves of forest degradation that spread along transport routes and rural supply zones (Ahrends et al., 2010; Zorrilla-Miras et al., 2018). This dynamic underscores charcoal's role as an urban–rural energy system, linking rural producers to urban consumers through extensive informal value chains.

Economically, charcoal production provides critical livelihood support for millions of rural households (FAO, 2017). It offers income diversification and serves as a safety net during agricultural off-seasons or market downturns (Jones et al., 2016; Vollmer et al., 2017). However, benefits are unevenly distributed along the value chain: producers, often the poorest actors, receive the least income, while transporters and traders capture a larger share of profits (Agyei et al., 2018; Baumert et al., 2016). This imbalance limits poverty reduction potential and exposes producers to resource depletion risks and long-term livelihood insecurity (Schure et al., 2014).

Environmentally, charcoal production is a major driver of deforestation and forest degradation in Africa (Chidumayo and Gumbo, 2013). The impacts, however, vary by country and governance system. In Tanzania and Kenya, open-access forest use and weak enforcement have led to widespread woodland depletion (Sander et al. 2013). In Zambia and Malawi, community-based forest management has emerged as an alternative governance approach, showing potential to promote sustainable production when benefits are equitably shared (FAO, 2017; van 't Veen, 2022). In Ghana and Nigeria, charcoal trade has become increasingly commercialized, feeding regional and cross-border markets (Schure et al., 2014).

Governance diversity is therefore central to understanding Africa's charcoal sector. Systems range from open-access and permit-based frameworks to community-managed regimes, each with distinct ecological and social outcomes. Weak institutional capacity, overlapping formal and informal regulations, and political incentives often enable illegal or unregulated production, undermining sustainability goals (Branch and Martiniello, 2018; Schure et al., 2014). Nonetheless, locally grounded governance reforms and more secure tenure arrangements have potential in reconciling energy needs with forest conservation.

Charcoal in Africa thus represents both an essential livelihood resource and an environmental challenge. Its future depends on how effectively governance systems can adapt to rapid demographic and urban transitions while promoting sustainable forest management.

2.2.2 Charcoal production and use in Tanzania

Tanzania is among Africa's largest charcoal producers and consumers, with charcoal serving as the primary energy source for urban households and a crucial income stream for rural communities (FAO, 2017; van 't Veen, 2022). In Dar es Salaam, more than 90 % of households depend on charcoal for daily cooking, making it indispensable to the country's energy security (Malimbwi and Zahabu, 2008). This dual dependence—urban consumption and rural production—places charcoal at the center of the country's livelihoods, energy access, and forest management nexus (Sander et al., 2013).

Unlike in many neighboring countries, Tanzania has made charcoal governance an explicit component of national forest and energy policy. The Forest Act of 2002 and its regulations establish a licensing and taxation framework for production and transport, while the National Forest Policy (United Republic of Tanzania, 1998) recognizes charcoal as part of the formal energy economy. However, enforcement remains weak, and a large share of charcoal is still produced and traded informally (Schure et al., 2014; van 't Veen, 2022). Government bans and permit restrictions have periodically sought to curb deforestation, yet such measures often displace production geographically rather than reducing total output (Sander et al., 2013; van 't Veen, 2022).

Tanzania is widely recognized as a regional leader in community-based natural resource management (CBNRM) (FAO, 2017). Through frameworks such as Community-Based Forest Management (CBFM) and Joint Forest Management, local communities can obtain legal rights to manage forest reserves, develop bylaws, and issue harvesting permits—including for charcoal—under approved management plans (Blomley et al., 2008; United Republic of Tanzania, 1998). These programs have demonstrated positive outcomes where effectively implemented, including reduced illegal logging, improved regeneration, and greater local participation in decision-making (Blomley et al., 2008; Lund and Treue, 2008). However, outcomes remain uneven across regions: limited institutional capacity, elite capture, and unequal benefit-sharing have constrained success and hindered the integration of charcoal value chains into local governance systems (Sander et al., 2013; van 't Veen, 2022).

To clarify how governance structures influence charcoal production, it is useful to distinguish between the two dominant management regimes examined in this study:

- Open Access (OA):
  Forests under open-access conditions are nominally state-owned but lack effective regulation or enforcement. Extraction is unrestricted, and users have no formal management

responsibilities or tenure security. Such systems often lead to unsustainable harvesting, as individual incentives to exploit outweigh collective incentives to conserve (Hardin, 1968; Ostrom, 2009; Sander et al., 2013).

- Community-Based Natural Resource Management (CBNRM):
  This approach devolves management authority to local communities through frameworks such as CBFM and JFM. Communities establish bylaws, enforce harvesting quotas, and benefit from forest revenues, creating incentives for sustainable use and regeneration (Blomley et al., 2008; FAO, 2017; Lund and Treue, 2008; United Republic of Tanzania, 1998). When implemented effectively, CBNRM improves forest condition and governance accountability, though challenges remain regarding institutional capacity and equitable benefit-sharing (van ’t Veen, 2022).

Charcoal production in Tanzania remains predominantly rural and small-scale, relying on traditional earth kilns with conversion efficiencies as low as 10–15 % (Bailis et al., 2015; FAO, 2017). Nonetheless, a gradual shift toward more organized and commercially structured value chains distinguishes Tanzania from countries such as Mozambique or Malawi, where production is largely subsistence-based (van ’t Veen, 2022). Around Dar es Salaam, Morogoro, and Dodoma, charcoal networks have become increasingly formalized, connecting rural producers, transporters, and traders (Sander et al., 2013). Despite this growing organization, producers typically capture only a small share of market value, while middlemen and traders retain the majority of profits (Baumert et al., 2016; Schure et al., 2014).

Environmentally, charcoal production remains a major driver of woodland degradation, particularly in the Miombo woodlands that cover much of central and southern Tanzania (Ahrends et al., 2010). Production follows spatial patterns of extraction that radiate from urban centers, leaving behind degraded landscapes and diminished regeneration capacity (Ahrends et al., 2010; Zorrilla-Miras et al., 2018). These dynamics have intensified debates over how to reconcile forest conservation with energy and livelihood needs within the country’s development agenda (FAO, 2017; van ’t Veen, 2022).

Overall, Tanzania’s charcoal sector illustrates both the potential and limits of governance reform in achieving sustainable energy transitions. The coexistence of formal regulation, widespread informality, and community-based management captures the complexity of energy transitions in rapidly developing economies—where sustainability depends not only on policy design, but also on the social and ecological realities of implementation (Branch and Martiniello, 2018; van ’t Veen, 2022).

2.3 The Miombo ecosystem

The miombo woodland is the most extensive tropical seasonal woodland formation in Africa. This chapter provides an overview of its geographic extent, ecological structure, and socioecological significance to contextualize its role in the charcoal production system

2.3.1 Geographic extent

The Miombo woodlands form the ecological backbone of southern and central Africa. Spanning across countries such as Angola, Zambia, Tanzania, Mozambique, Malawi, and Zimbabwe, they represent the largest tropical dry forest formation on the continent and one of the most extensive woodland ecosystems globally (Campbell, 1996; Chidumayo and Gumbo, 2010; Frost, 1996). Historically, the Miombo region covered approximately 2.7 million km² across the Zambezian ecoregion (Campbell, 1996; Chidumayo and Gumbo, 2010; Frost, 1996). However, recent remote-sensing assessments indicate that this area has declined to around 1.9 million km² by 2020, largely due to deforestation and woodland degradation (Ribeiro et al., 2020). Despite this contraction, Miombo remains one of the most extensive continuous dryland forest systems on Earth.

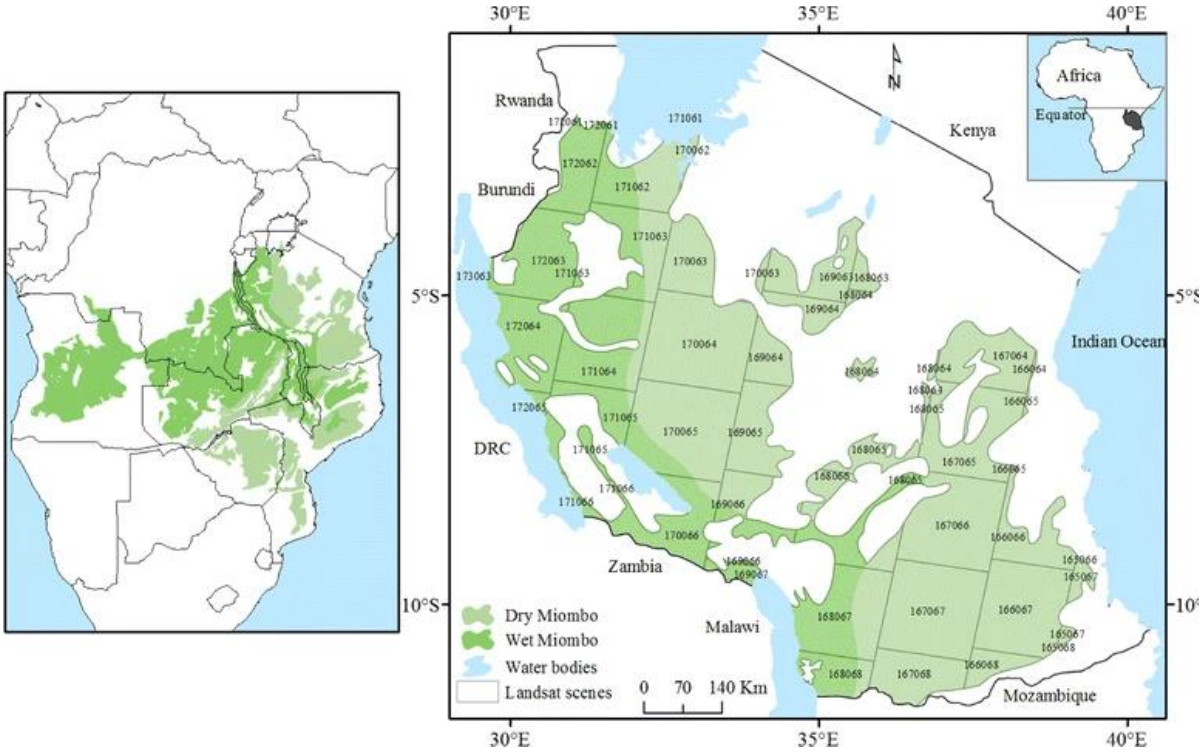

Figure 1 Extent of Miombo woodlands (Tarimo et al., 2015).

2.3.2 Ecology of the miombo woodlands

Ecologically, the Miombo woodlands are defined by the dominance of leguminous tree genera such as *Brachystegia*, *Julbernardia*, and *Isoberlinia* (Frost, 1996; White, 1983). These trees form open-canopy, seasonally dry forests adapted to nutrient-poor soils, pronounced rainfall seasonality, and frequent fires (Campbell, 1996; Frost, 1996). Fire is a natural and recurrent ecological process in Miombo, maintaining the woodland structure and stimulating regeneration through coppicing, a resprouting strategy that allows trees to recover after cutting or burning (Ryan et al., 2016; Syampungani et al., 2017). This capacity for regeneration underpins both the resilience and vulnerability of the Miombo: while it can recover from low-intensity disturbances, repeated overexploitation and shortened recovery cycles reduce its regenerative potential (Chidumayo and Gumbo, 2010).

The ecological structure and function of the Miombo woodlands are shaped by strong climatic seasonality, nutrient-poor soils, and recurrent disturbances such as fire and harvesting. Rainfall is concentrated in a single wet season lasting three to five months, followed by a prolonged dry season with high evapotranspiration and limited soil moisture (Campbell, 1996; Frost, 1996). This climatic pattern, coupled with generally ferrallitic and nutrient-depleted soils, constrains productivity but favours tree species that are drought-tolerant, deep-rooted, and slow-growing (Chidumayo and Gumbo, 2010; Frost, 1996).

Vegetation structure and dominant species

The Miombo canopy typically ranges between 10–20 m in height and is dominated by deciduous leguminous species of the genera *Brachystegia*, *Julbernardia*, and *Isoberlinia* (Frost, 1996; White, 1983). These genera account for most of the basal area in mature woodlands and define the ecosystem's floristic identity within the Zambezian Phytochorion (White, 1983). The understorey is often composed of shrubs and perennial grasses such as *Hyparrhenia* and *Andropogon* species, which contribute to the frequent occurrence of fire (Campbell, 1996).

Fire ecology and regeneration

Fire is a central ecological process in the Miombo, influencing species composition, structure, and nutrient cycling (Frost, 1996; Ryan et al., 2016). Most trees are fire-tolerant, possessing thick bark, underground rootstocks, and the ability to resprout (coppice) after disturbance (Syampungani et al., 2017). Resprouting allows rapid vegetative regeneration following cutting or burning, giving Miombo woodlands a high capacity for recovery after low-intensity disturbance events (Chidumayo and Gumbo, 2010). However, frequent or high-intensity fires, particularly when combined with repeated wood extraction, can deplete root carbohydrate reserves and reduce resprouting vigor (Ryan et al., 2016).

Over time, this leads to the replacement of mature woodland by more open, degraded systems dominated by grasses and shrubs (Syampungani et al., 2009).

Nutrient and carbon dynamics

Miombo ecosystems are nutrient-conservative systems, where most nutrients are stored in biomass rather than soil (Campbell, 1996; Frost, 1996). The dominance of leguminous trees enables symbiotic nitrogen fixation, which plays a key role in maintaining soil fertility under low-nutrient conditions (Chidumayo and Gumbo, 2010). Leaf litter decomposition during the wet season provides short pulses of nutrient availability that sustain primary productivity (Frost, 1996). In carbon terms, Miombo woodlands store substantial amounts in below-ground biomass and soils, estimated at 60–70% of total ecosystem carbon stocks (Ribeiro et al., 2020). Disturbances such as charcoal production and frequent burning reduce above-ground carbon but may stimulate below-ground fluxes through enhanced root turnover and regrowth (Ryan et al., 2016).

Successional dynamics

The successional trajectory of Miombo follows a slow transition from grassy fallow to closed-canopy woodland, a process that can take 30–50 years depending on soil fertility and disturbance frequency(Chidumayo and Gumbo, 2010; Syampungani et al., 2009). Early successional stages are dominated by fast-growing species such as *Combretum* and *Terminalia*, which are later replaced by *Brachystegia* and *Julbernardia* in mature stands (Campbell, 1996). Human disturbances, particularly shifting cultivation and charcoal production often interrupt this cycle, preventing full canopy recovery and promoting secondary woodland mosaics (Goncalves, 2019). These ecological and social dynamics form the foundation for understanding how charcoal production interacts with and transforms the Miombo ecosystem.

2.3.3 Ecosystem functions and services

The Miombo's ecological significance extends beyond its flora. These woodlands sustain a remarkably high biodiversity, with more than 8,000 plant species, roughly a quarter of which are endemic to the Zambezian region (Linder et al., 2012; White, 1983). They provide habitat for diverse fauna, serve as carbon reservoirs, and regulate hydrological cycles across southern Africa (Chidumayo and Gumbo, 2010; Ribeiro et al., 2020). Estimates suggest that Miombo woodlands store between 18 and 24 petagrams of carbon in biomass and soils, underscoring their importance for regional and global carbon budgets (Ribeiro et al., 2020).

Socioecologically, Miombo woodlands are integral to the livelihoods of over 100 million people (Campbell, 1996; Syampungani et al., 2009). They supply fuelwood, charcoal, timber, food, honey, and medicinal plants, forming the primary source of energy and material resources for rural populations

(Chidumayo and Gumbo, 2010). At the same time, they are increasingly under pressure from charcoal production, shifting cultivation, and agricultural expansion, which collectively drive deforestation and forest degradation (Ahrends et al., 2010). The dual role of Miombo woodlands as both a source of livelihood and an ecological regulator makes them a focal point in regional debates on sustainable land use and climate mitigation.

Overall, Miombo woodlands are fire- and disturbance-dependent ecosystems characterized by slow biomass accumulation, high resilience to episodic stress, and strong coupling between ecological processes and human use. Their persistence under increasing pressure depends on maintaining the natural rhythms of fire, regeneration, and nutrient cycling that define their ecological equilibrium.

2.3.4 Charcoal in Tanzania's Miombo

In Tanzania, Miombo woodlands dominate national forest cover and form the ecological foundation for both rural livelihoods and the charcoal economy. The country's long-standing experience with CBNRM provides a unique governance context in which forest use and conservation are closely intertwined (Blomley et al., 2008; FAO, 2017).

Ecologically, Tanzania's Miombo landscapes encompass both wet and dry woodland zones that differ in structure, productivity, and response to disturbance (Chidumayo and Gumbo, 2010; Frost, 1996; White, 1983).

- **Wet Miombo**, dominant in the southern highlands and parts of Morogoro and Ruvuma, receives over 1,000 mm of rainfall annually and features taller, denser woodlands dominated by *Brachystegia spiciformis* and *Julbernardia globiflora*.
- **Dry Miombo**, prevalent in central regions such as Kilosa, Dodoma, and Tabora, receives 700–1,000 mm of rainfall and is characterized by shorter canopies and slower-growing species such as *Brachystegia boehmii* and *Julbernardia paniculata* (Frost, 1996; Syampungani et al., 2009).

This ecological variation directly affects the availability, density, and regeneration potential of wood for charcoal. Wet Miombo tends to yield more biomass in the short term but is more sensitive to repeated harvesting, whereas dry Miombo recovers slowly yet demonstrates stronger long-term resilience through vigorous coppicing. Consequently, sustainable charcoal management must adapt to local ecological conditions rather than treating Miombo as a uniform resource base (Campbell, 1996; Ryan et al., 2016).

Charcoal production has emerged as one of the dominant disturbance regimes in Tanzania's Miombo, interacting with fire, grazing, and shifting cultivation to shape woodland structure and carbon dynamics (Chidumayo and Gumbo, 2010; Ryan et al., 2016). Studies in central Tanzania show that repeated

harvesting without adequate recovery leads to declines in above-ground biomass and shifts in species composition toward more fire-tolerant, lower-density species (Ahrends et al., 2010). Yet, when managed through regulated harvesting cycles and longer rotation periods, Miombo woodlands can maintain their natural regeneration and carbon recovery capacity, indicating that degradation results primarily from weak governance rather than charcoal production per se (FAO, 2017; Syampungani et al., 2017).

Taken together, charcoal production and Miombo woodlands constitute a tightly coupled socio-ecological system. Charcoal remains both a livelihood necessity and a source of ecological strain, while the Miombo provides the material and ecological foundation for resilience or degradation depending on how it is governed. Achieving sustainability therefore depends on governance systems capable of integrating energy access, rural livelihoods, and ecosystem conservation within a coherent management framework.

To understand how these governance regimes influence ecological outcomes, it is essential to ground the analysis in ecological theory. The way charcoal harvesting alters forest structure and species composition can be interpreted through the lens of disturbance ecology and trait-based theory, which together explain how species characteristics mediate ecosystem responses to human use. The following section introduces these theoretical foundations, focusing on the relationships between functional traits, disturbance, and biodiversity as the ecological core of this study.

2.4 Ecological theory: traits, disturbance, and biodiversity

Understanding how ecosystems respond to disturbance requires linking community composition to species' functional characteristics. Modern ecology increasingly focuses on functional traits—measurable features of organisms that determine how they respond to environmental change and influence ecosystem processes (Díaz et al., 2016; Violle et al., 2007). Trait-based frameworks provide a mechanistic bridge between species-level responses and system-level functioning, allowing ecologists to predict how disturbance regimes, such as fire, drought, or selective harvesting, shape biodiversity and resilience. Within this perspective, both disturbance intensity and trait composition act as central drivers of ecosystem dynamics, determining whether a system recovers, transforms, or collapses following pressure.

2.4.1 Functional traits and ecosystem functioning

In ecology, functional traits are defined as measurable morphological, physiological, or phenological features of organisms that influence their performance and ecosystem roles (Violle et al., 2007). Unlike taxonomic classifications, trait-based approaches describe organisms through characteristics that directly affect resource acquisition, growth, reproduction, and survival. This framework allows

researchers to move beyond species identities toward a mechanistic understanding of how communities assemble and ecosystems function (Díaz et al., 2016; Lavorel and Garnier, 2002).

Plant traits are often organized along a “fast–slow” continuum of ecological strategies (Reich, 2014). Species with fast traits such as high specific leaf area, rapid growth, and low wood density typically dominate resource-rich or frequently disturbed environments. In contrast, slow species exhibit dense wood, long leaf lifespans, and low nutrient turnover, enabling persistence under nutrient-poor or drought-prone conditions. The distribution of these traits across communities forms a functional trait space that reflects trade-offs between productivity, resource use, and stress tolerance (Díaz et al., 2016).

At the ecosystem level, functional traits govern processes such as primary production, nutrient cycling, and carbon storage. For instance, variation in wood density and leaf area can affect biomass accumulation and decomposition rates, while traits related to nitrogen fixation influence soil fertility (Díaz et al., 2007; Lavorel and Garnier, 2002). Consequently, changes in trait composition may alter ecosystem functioning and resilience.

In managed or exploited ecosystems, such as the Miombo woodlands, human activities can act as selective filters that favor certain functional traits over others. Selective harvesting refers to the disproportionate removal of individuals or species based on desirable traits such as high wood density or large stem diameter which can gradually shift community composition. Over time, such filtering may modify the community’s functional trait space and its capacity to recover after disturbance.

2.4.2 Disturbance: trait interactions and resilience

Disturbance is a fundamental ecological process shaping community composition and ecosystem dynamics (Pickett et al., 1989; Turner, 2010). It can be defined as any discrete event that disrupts ecosystem structure, resource availability, or species interactions, leading to shifts in community composition over time. Examples include natural drivers such as fire, drought, and herbivory, as well as anthropogenic pressures like logging, grazing, and woodfuel extraction. The frequency, intensity, and spatial scale of disturbance determine whether ecosystems recover, reorganize, or shift toward alternative states (Folke et al., 2004; Holling, 1973).

From a trait-based perspective, disturbance acts as an environmental filter that selects for species with traits conferring tolerance or rapid recovery (Lavorel and Garnier, 2002). In frequently disturbed systems, species tend to exhibit fast traits such as high growth rates, resprouting ability, and early reproduction, enabling rapid colonization and turnover (Díaz et al., 2007; Reich, 2014). In contrast, systems experiencing infrequent or low-intensity disturbance often favor slow, conservative species

with dense wood, deep root systems, and longer lifespans, which enhance persistence but limit short-term recovery capacity.

The Miombo woodlands exemplify ecosystems where disturbance–trait interactions strongly influence resilience. Fire, drought, and wood extraction act as recurrent filters shaping community composition. Many Miombo species exhibit adaptive traits such as thick bark and vigorous coppicing that allow recovery after fire or cutting (Chidumayo and Gumbo, 2010; Syampungani et al., 2017). However, when disturbance regimes exceed ecological thresholds trait filtering may drive shifts toward communities dominated by more disturbance-tolerant but functionally limited species. Such changes can reduce ecosystem functions like biomass accumulation, carbon storage, and seedling recruitment (Ryan et al., 2016).

Resilience, defined as the capacity of an ecosystem to absorb disturbance and reorganize while retaining its basic structure and function (Holling, 1973), is therefore tightly linked to the diversity and distribution of functional traits (Elmqvist et al., 2003). Communities with a wide range of response traits tend to exhibit greater resilience because they maintain ecological functions even as species composition changes (Díaz et al., 2016). In contrast, low trait diversity or strong directional filtering, such as through targeted wood extraction, can reduce functional redundancy and make systems more vulnerable to collapse under cumulative pressures (Elmqvist et al., 2003). Rather than viewing disturbance as inherently destructive, this perspective emphasizes that ecosystem outcomes depend on the balance between disturbance intensity, recovery time, and the diversity of traits that enable adaptation and regeneration (Folke et al., 2004; Lavorel and Garnier, 2002).

2.4.3 Biodiversity responses to disturbance

Disturbance plays a central role in structuring biodiversity by altering species composition, richness, and evenness across space and time (Connell, 1978; Sousa, 1984). According to the intermediate disturbance hypothesis, biodiversity tends to peak at intermediate levels of disturbance intensity or frequency, where both competitive exclusion and local extinction are minimized (Connell, 1978). At low disturbance levels, competitive dominants suppress diversity, while at high disturbance levels, only a few tolerant species persist. This theoretical relationship has been observed across ecosystems, from coral reefs and grasslands to tropical woodlands (Mackey and Currie, 2001; Molino and Sabatier, 2001).

Biodiversity can be described across three hierarchical scales. Alpha diversity, first conceptualized by Whittaker (Whittaker, 1972), refers to the number and relative abundance of species within a specific site or community, capturing local richness and evenness. Beta diversity measures the degree of species turnover or differentiation between sites—how distinct communities are across space—and reflects spatial heterogeneity in composition (Whittaker, 1972). Gamma diversity represents total

diversity at the regional scale, integrating both within-site (alpha) and between-site (beta) components (Jost, 2007). Together, these measures provide a multiscale perspective for understanding how disturbance shapes ecological patterns from local to landscape levels.

Biodiversity can be described across three hierarchical scales. Alpha diversity refers to the number and relative abundance of species within a specific site or community. Beta diversity measures the turnover or differentiation of species between sites, reflecting how distinct communities are across a landscape. Gamma diversity represents total diversity at the regional scale, integrating variation within and between local communities (Jost, 2007; Whittaker, 1972).

Moderate disturbance can enhance beta diversity by creating spatial heterogeneity and a mosaic of successional stages, increasing the range of available niches (Chase and Myers, 2011). In contrast, intense or chronic disturbance tends to homogenize communities, reducing both alpha and gamma diversity. Thus, biodiversity responses are scale-dependent, governed by the interplay between disturbance intensity, spatial heterogeneity, and recovery dynamics (Sousa, 1984; Villéger et al., 2008).

Functional diversity, representing the range and distribution of traits within a community, provides a more mechanistic understanding of how disturbance affects ecosystem functioning (Petchey and Gaston, 2006). High functional redundancy where multiple species share similar ecological roles buffers ecosystems against disturbance by maintaining key processes even when species are lost (Elmqvist et al., 2003). However, directional filtering through repeated harvesting can erode this redundancy, leading to lower resilience and simplified community structures (Cadotte et al., 2011). In systems like the Miombo, where coppicing and fire tolerance are widespread adaptive traits, biodiversity outcomes depend on how disturbance regimes interact with these trait-mediated recovery processes (Ryan et al., 2016; Syampungani et al., 2017).

Understanding biodiversity responses to disturbance therefore requires integrating taxonomic (species richness, evenness, turnover) and functional (trait diversity, redundancy, dispersion) dimensions. This dual perspective links species-level patterns to ecosystem-level processes, allowing more accurate predictions of how anthropogenic pressures such as charcoal production alter ecological stability and resilience. It also provides the theoretical foundation for assessing whether governance systems can maintain diversity within sustainable disturbance thresholds.

2.5 From research gap to hypotheses

Building on the ecological and governance context outlined above, this section synthesizes the key research gaps emerging from current knowledge on charcoal production and Miombo woodlands. It outlines how this thesis addresses these gaps by formulating specific research questions, defining corresponding objectives, and developing hypotheses that guide the analytical framework of the study.

2.5.1 Research gap

Building on the preceding sections, this study focuses on how charcoal production interacts with the ecological functioning of Tanzania's Miombo woodlands under different governance regimes. Charcoal production in Tanzania's Miombo woodlands unfolds across interconnected social, economic, and ecological dimensions. Previous studies have emphasized its socioeconomic importance, highlighting how governance arrangements shape livelihood diversification, income distribution, and poverty alleviation potential (Schure et al., 2014; van 't Veen, 2022; Vollmer et al., 2017). As outlined in the preceding section, disturbance theory predicts that human harvesting alters species composition and trait distributions by selectively removing individuals with certain functional characteristics. Yet, the role of governance in mediating these ecological outcomes within charcoal production systems remains comparatively underexplored.

Against this background, this thesis examines the ecological dimension of charcoal production systems under contrasting governance regimes of OA and CBNRM to clarify how governance mediates ecological outcomes.

Ecological impacts can be conceptualized through four key variables: soil properties, carbon stock, tree species composition, and functional traits (Chidumayo and Gumbo, 2013). Charcoal production can alter soil structure and water-holding capacity (Lasota et al., 2021), increase organic matter and nutrient concentrations (Adio et al., 2022), and modify exchangeable cations such as Mn, Fe, Cu, and Zn (Chima et al., 2013). Similarly, its influence on carbon stocks depends on recovery time and disturbance frequency, which vary with climate, soil fertility, and stand structure (Cole et al., 2014).

However, much less is known about charcoal's influence on tree species composition and functional traits. Evidence from Kenya suggests that species diversity declines in charcoal-producing areas (Kiruki et al., 2017), while qualitative accounts from Ethiopia describe selective depletion of Acacia and Combretum species (Garedew and Simon, 2018). Selective harvesting which often targets dense-wood or slow-growing species prized for their high calorific value, can lead to long-term shifts in trait distributions and ecosystem functioning (Díaz et al., 2007). From an ecological perspective, such extraction acts as a directional filter on plant functional traits: species with traits conferring high wood density, deep rooting, or slow growth are disproportionately removed, while those with lighter wood, rapid growth, or higher reproductive turnover become more dominant. Over time, this could alters the community's functional composition, reducing its capacity for carbon storage, drought tolerance, and post-disturbance regeneration. In systems like the Miombo woodlands, where resilience relies on the balance between slow and fast species strategies, repeated selective removal may therefore erode the very traits that sustain long-term ecosystem stability.

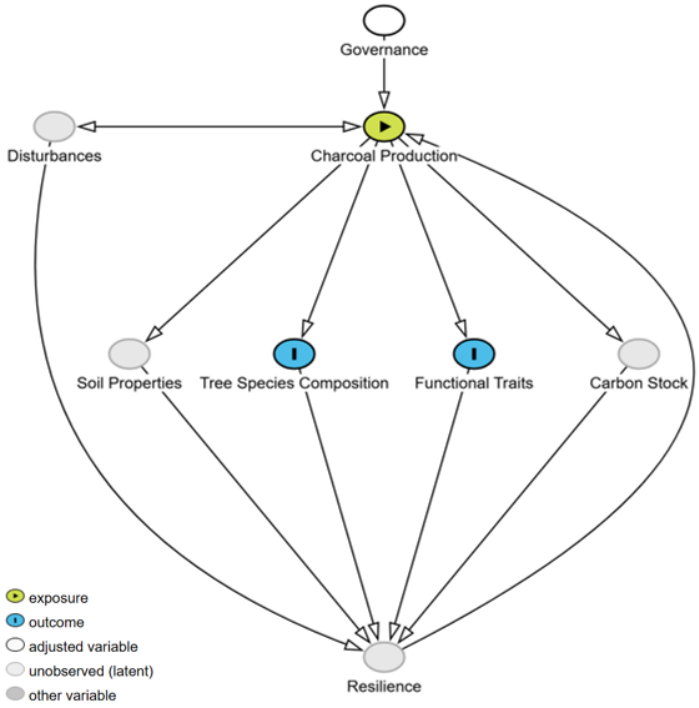

Figure 2 Conceptual framework of the charcoal system

This framework tries to illustrate the relationships among governance, charcoal production, ecological variables, and resilience. Yellow: exposure variable; blue: ecological outcomes; gray: mediating or latent variables.

The conceptual framework situates charcoal production as one among several interacting disturbance processes in the Miombo system. However, the intensity and frequency of disturbances determine whether the system remains resilient or shifts toward degradation. Charcoal extraction becomes ecologically critical when recovery intervals shorten or when cumulative biomass loss exceeds the system's regenerative capacity. Under these conditions, resilience—the capacity of the woodland to regain its structure and function—declines.

In this framework, governance acts as the key mediating factor. It cannot control natural disturbances, but it determines how charcoal production is regulated. Governance influences who extract, where, and how much, thereby shaping both the intensity and spatial distribution of charcoal-related pressures. Effective governance can maintain extraction within ecological thresholds, allowing the Miombo system to sustain its regenerative dynamics. Weak or absent governance, by contrast, amplifies degradation and can push the system beyond recovery limits.

Importantly, the framework highlights the feedback dynamics between resilience and charcoal production. In a resilient woodland, moderate extraction can occur without ecological collapse, biomass regrows, soils recover, and livelihoods are maintained. This represents a stabilizing (negative)

feedback, where ecosystem renewal counterbalances resource use. However, as extraction intensifies or recovery intervals shorten, the feedback can shift toward a destabilizing (positive) loop: reduced resilience leads to lower productivity, prompting further expansion of charcoal harvesting to meet demand. This accelerates degradation and diminishes the woodland's capacity to recover, locking the system into a downward spiral. In this context, governance could become the critical balancing mechanism—it determines whether the feedback remains stabilizing by enforcing sustainable harvest limits and regeneration intervals or turns destabilizing through unregulated exploitation. Effective governance can thus convert a potentially self-reinforcing degradation cycle into a self-correcting system that maintains ecological and social stability.

In summary, the framework conceptualizes charcoal production not merely as a driver of degradation but as part of a broader disturbance–resilience continuum. This thesis seeks to clarify the role governance plays within that continuum by examining how charcoal production influences key ecological dimensions of the Miombo woodlands. Specifically, it addresses whether charcoal use and governance regimes are associated with measurable differences in tree species composition and functional traits, as outlined in the research questions.

2.5.2 Research questions

- **RQ1:** Is there a difference in Miombo woodlands tree species composition and traits used and not used for charcoal production?
- **RQ2:** Is there a difference in tree species composition and traits under different management regimes, i.e. open access versus community based natural resource management?

2.5.3 Objective

This study aims to assess the ecological impacts of charcoal production and governance on the Miombo woodlands of Tanzania. Specifically, it investigates how charcoal use and management regimes shape tree species composition and functional trait distributions, and how these ecological patterns reflect broader differences in forest resilience. To achieve this, the research combines two complementary analytical approaches: (1) a trait-based analysis using data from the TRY Plant Trait Database to compare the trait space of species used and not used for charcoal production at the national scale, and 2) a spectral variability analysis (SVA) using Sentinel-2 imagery to evaluate ecological variability across governance contexts at multiple spatial scales. At the national scale, SVA compares protected and unprotected Miombo areas to capture large-scale patterns of forest condition and species richness, evenness and diversity metrics. At the local scale, the analysis zooms in on village-level governance systems, contrasting CBNRM with open-access (OA) areas to assess how governance influences ecosystem resilience. Together, these approaches provide an integrated perspective on how ecological

structure and function respond to different charcoal production systems, offering insights into the sustainability and resilience of Tanzania's Miombo ecosystems.

2.5.4 Hypothesis

For the first research question, I hypothesize that areas subject to charcoal production will exhibit lower tree species richness, diversity, and evenness than those not used for charcoal. This decline is expected because charcoal producers preferentially harvest species with dense wood that yield more energy per unit volume and produce high-quality charcoal (Cazzolla Gatti et al., 2015; Chidumayo and Gumbo, 2013). Over time, this directional filtering could alter the community's functional composition and potentially its regeneration capacity (Díaz et al., 2007; Syampungani et al., 2017). Specifically, I predict shifts in three functional traits. Wood density is expected to decline in harvested areas due to the preferential removal of dense-wood species that yield high-quality charcoal. In contrast, specific leaf area is likely to increase, reflecting a community shift toward fast-growing, acquisitive species that invest less in structural tissues. Likewise, seed mass is anticipated to decrease in more intensely disturbed areas, as the community becomes dominated by small-seeded species adapted to rapid colonization and high turnover. Together, these changes could lead to a reduction of functional diversity, thereby weakening the woodland's capacity to recover from disturbance.

For the second research question, I hypothesize that CBNRM systems will maintain higher species richness, diversity, and functional diversity than OA systems. Regulated harvesting, rotation cycles, and management oversight within CBNRM frameworks (Blomley et al., 2008; FAO, 2017; Ishengoma et al., 2016) are expected to moderate disturbance intensity and allow recovery of high wood density and large-seeded species.

In contrast, OA systems where recovery time might be shortened will favor species with high SLA and low wood density, reflecting fast growth but reduced ecosystem stability (Ahrends et al., 2010; Chidumayo and Gumbo, 2013). As a result, functional composition in OA areas will show stronger trait convergence and lower resilience, while CBNRM systems retain a more balanced representation of slow- and fast-strategy species.

Together, these hypotheses test how governance mediates trait–disturbance dynamics, linking specific ecological mechanisms to the sustainability of charcoal production in Tanzania's Miombo woodlands.

3 Theoretical and analytical framework

Understanding the ecological effects of charcoal production requires an integrative framework that connects functional ecology, biodiversity theory, and remote sensing. The theoretical background of this study aims to provide the conceptual link between species-level functional traits, landscape-scale spectral variability, and governance-mediated disturbance regimes. Charcoal production is therefore conceptualized as a recurring anthropogenic disturbance that modifies forest structure and species composition, while spectral and trait-based indicators allow these changes to be quantified across scales, from trees and communities to ecosystems.

Two key concepts underpin this framework: functional traits, which describe how species respond to and recover from disturbance, and spectral diversity, which provides a remote-sensing proxy for ecological variability and forest condition.

Functional Traits: Building on the ecological theory introduced above, this study uses functional trait data to quantify how disturbance and governance shape community composition in the Miombo woodlands. Traits such as wood density, specific leaf area, and seed mass are employed as indicators of species' ecological strategies capturing trade-offs between growth, resource use, and regeneration potential (Díaz et al., 2016; Reich, 2014). These traits are analyzed using Principal Component Analysis (PCA) to represent the functional structure of communities and to identify shifts in trait space associated with charcoal production intensity and governance type.

Spectral Diversity: Spectral diversity complements the trait-based approach by linking field-level ecological variation to landscape-scale vegetation patterns. Variability in canopy reflectance measured through indices such as Normalized difference vegetation index (NDVI), normalized difference moisture index (NDMI), and the Canopy Chlorophyll Index (CCI) is used as a proxy for differences in vegetation structure and composition (Asner and Martin, 2009; Ustin and Gamon, 2010). Through Spectral Variability Analysis (SVA), these data could capture how ecological heterogeneity changes under different disturbance regimes, providing an integrative, spatially continuous measure of biodiversity and forest condition.

In this study, spectral metrics such as NDVI, NDMI, and CCI, along with SVA, are used to quantify differences in canopy structure and vegetation composition across governance regimes. Combined with trait-based approaches, these metrics allow a multiscale assessment of biodiversity and forest resilience. Statistical tools such as PCA, k-means clustering, and distance metrics are employed to characterize separability between ecological states, while diversity indices and tests such as PERMANOVA validate the robustness of observed patterns.

Together, these approaches create a theoretical and methodological foundation for linking disturbance ecology, functional traits, and remote sensing, enabling a comprehensive assessment of how charcoal production and governance shape the resilience of Tanzania's Miombo woodlands.

3.1 Foundation of principal component analysis

The functional traits and the sentinel data were analyzed by the means of a Principal Component Analysis (PCA). PCA is a multivariate statistical method used to reduce the dimensionality of complex datasets while retaining most of the original variation (Jolliffe and Cadima, 2016). In the context of plant functional traits, PCA allows for visualizing and quantifying how species differ across multiple traits simultaneously, by summarizing correlated variables into a few orthogonal axes called principal components that capture the dominant gradients of trait variation.

PCA can be applied to any multidimensional dataset, where each principal component (PC) represents an eigenvector of that data cloud. The first principal component (PC1) accounts for the largest share of total variation, while each subsequent component explains the maxim remaining variation orthogonal to the preceding axes (Jolliffe and Cadima, 2016). PCA is particularly useful when the first few components (typically PC1 and PC2) capture most of the variation, allowing the multidimensional relationships among variables or species to be visualized effectively in a two-dimensional space.

A landmark study (Díaz et al., 2016) used PCA to map the global spectrum of plant form and function, analyzing six key traits related to growth, survival, and reproduction across thousands of vascular plant species. Their analysis revealed that roughly three-quarters of global trait variation could be summarized in just two dimensions. In the resulting PCA space (Figure 3), woody and non-woody species separated clearly along the first principal component (PC1), which was most strongly associated with stem specific density (SSD), plant height, and seed mass. This axis therefore represents a gradient from small, fast-growing, low-density species to large, structurally robust species with conservative growth strategies. The visualization in Figure 3 exemplifies how PCA can condense multidimensional trait data into interpretable ecological gradients, revealing major axes of plant functional diversity across ecosystems.

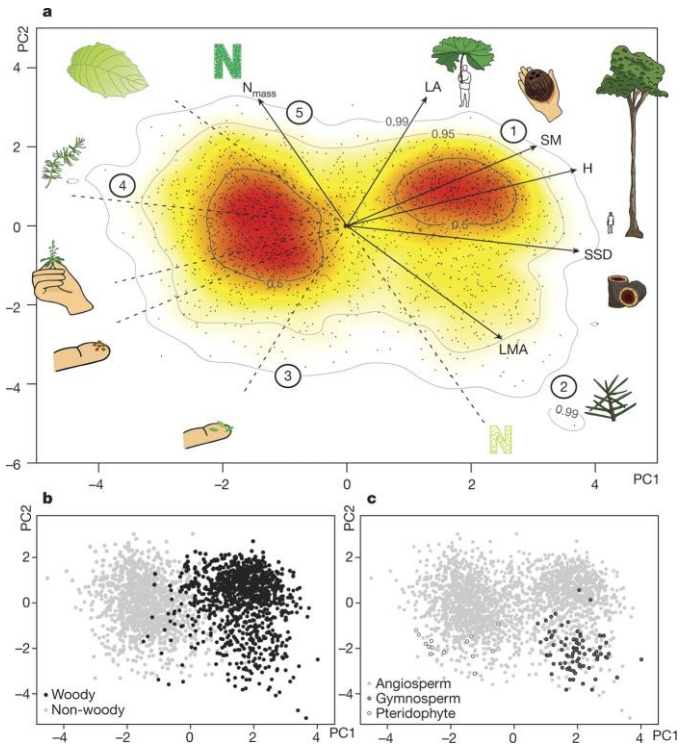

Figure 3 Example PCA from Díaz et al. 2016.

For the functional trait analysis of this thesis, multiple PCA configurations were conducted using different combinations of traits to test the robustness and interpretability of trait-space patterns. Trait selection was guided by three complementary criteria: (1) data completeness, prioritizing traits with sufficient coverage across species.; (2) ecological reasoning, ensuring inclusion of traits functionally linked to charcoal production, forest structure, and resilience; and (3) statistical relevance, assessed using the Boruta algorithm (Kursa and Rudnicki, 2010).

The Boruta algorithm was used as a wrapper-based feature selection method built around random forests (Kursa and Rudnicki, 2010). It iteratively compares the importance of real traits against randomized “shadow” features to identify all variables that are statistically relevant to the response. This approach is particularly suited for ecological data because it retains all important predictors, rather than only a minimal subset, thereby preserving the multidimensional structure of functional variation that underlies PCA interpretation.

While PCA is a powerful tool for visualizing multidimensional relationships and identifying potential group structures, it remains primarily exploratory. It can suggest differences between groups, but it does not quantify their statistical significance or magnitude. For the functional trait analysis, additional statistical testing was performed using Permutational Multivariate Analysis of Variance (PERMANOVA), which formally evaluates whether species groups differ in multivariate trait composition.

In contrast, for the spectral data, where sample sizes are orders of magnitude larger and PCA serves as a dimensionality reduction step, separability metrics such as Mahalanobis, Bhattacharyya, and Jeffries–Matusita distances were applied instead. These provide a direct, quantitative measure of how distinct spectral clusters are within multivariate space, complementing the visual interpretation of PCA results. Together, these approaches provide statistical and geometric evidence for the patterns revealed.

3.1.1 Permutational multivariate analysis of variance

Permutational Multivariate Analysis of Variance (PERMANOVA) was developed by Marti J. Anderson (Anderson, 2001) as a non-parametric method for testing multivariate differences between groups. Originally designed for ecological community data, it extends the logic of traditional ANOVA to distance matrices, allowing comparisons based on any dissimilarity measure (e.g., Euclidean) rather than relying on normality or homogeneity of variance assumptions. PERMANOVA partitions the total variation in a dataset into within-group and between-group components and uses permutations to determine the statistical significance of group separation.

In this study, PERMANOVA is used to test whether the functional trait composition of tree species differs significantly between charcoal and non-charcoal species. The analysis provides several key statistics that together describe the strength, direction, and reliability of group differences. Following are these key statistics with the example number of Figure 9:

- *F* Comparable to the F-ratio in classical ANOVA, the pseudo-F measures the ratio of between-group variance to within-group variance, based on a chosen distance matrix. Values are usually >1 when groups differ meaningfully.
    - *F* = 3.417 suggests moderate but statistically significant separation between charcoal and non-charcoal species.
- R^2 represents the proportion of total variation in the dataset explained by the grouping variable. Values can range between 0-1.
    - R^2 = 0.046 indicates that 4.6 % of total trait-space variation is attributable to whether species are used for charcoal or not.
- *p* the probability that an observed F-value could arise by chance if group labels were randomly assigned. Values can range between 0-1.
    - *p* = 0.008 indicates a low probability that group differences in trait composition occurred by chance, supporting a real ecological separation.

- *Dispersion p* tests whether group dispersions (spreads) within multivariate space differ significantly, ensuring that PERMANOVA results reflect centroid differences rather than unequal variance. Values can range between 0-1.
    - *Dispersion p* = 0.215 suggests that group spreads are not significantly different, confirming that the separation in PCA space reflects true compositional differences rather than unequal variance.

While PERMANOVA provides a robust statistical test for group differences in multivariate space, its interpretive power diminishes with very large sample sizes. In datasets containing thousands of data points, such as spectral imagery, PERMANOVA can become computationally intensive, and even small, ecologically negligible effects may yield statistically significant p-values due to high replication. In such cases, separability metrics offer a more direct and interpretable alternative. Accordingly, separability metrics were introduced as a complementary approach to quantify the distinctness of ecological groups directly within multivariate feature space.

3.1.2 Separability metrics

Separability metrics provide quantitative measures of how distinct two or more classes or groups are within a multidimensional feature space. Unlike permutation-based tests, they assess the degree of separation between distributions directly from their statistical properties. These measures are particularly useful for large datasets such as Sentinel-2 imagery, where the number of pixels makes formal hypothesis testing computationally expensive and statistically overpowered. In this study, separability metrics were used to quantify how clearly areas under different governance types can be distinguished based on their spectral signatures. The following metrics were used in this thesis. The example numbers stem form Figure 18:

- Centroid distance (Euclidean)
    - Measures the straight-line distance between the mean spectral signatures (centroids) of two groups in PCA space.
    - Higher values indicate greater overall spectral separation.
    - Centroid distance = 1.532 shows a moderate difference between the spectral means.
- Mahalanobis distance (D_m) (Mahalanobis, 1936)
    - Accounts for covariance among bands, describing how many multivariate standard deviations apart the two centroids are.
    - Typical values range from 0 (overlap) to > 2 (clear separation).
    - D_m = 0.701 indicates partial spectral distinctness between the groups.
- Bhattacharyya distance (D^B) (Bhattacharyya, 1943)

  - Combines mean and variance information to estimate class overlap in probability space.
  - Values near 0 imply almost complete overlap; values > 2 imply strong separability.
  - D^B = 0.088 suggests substantial spectral overlap, meaning the two groups share many similar reflectance features.
- Jeffries–Matusita distance (D^{JM}) (Jeffreys, 1997)
  - A scaled transformation of the Bhattacharyya distance ranging from 0 to 2, where 2 represents perfect separability.
  - D^{JM} = 0.168 confirms weak spectral separability—some distinction exists, but large areas of overlap remain.

These separability metrics were applied to quantify how distinct spectral signatures are between pixels belonging to different groups, namely protected and unprotected areas on a national scale and CBNRM and OA on a local scale. The origin of the protected area layer used for this classification is described in the following section.

3.2 Spectral analysis framework

Building upon the multivariate framework established through PCA and its complementary metrics, the next part of the analysis focuses on spectral data derived from Sentinel-2 imagery. While trait analyses characterize functional differences among species, the spectral analysis extends this logic to the landscape scale, capturing ecological variation through differences in canopy reflectance patterns. To structure this analysis, Miombo woodland areas were first categorized according to protection status or governance regime, before deriving vegetation indices and spectral diversity metrics.

3.2.1 Protected area classification

To differentiate between protected and unprotected Miombo woodlands, this study employed the International Union for Conservation of Nature (IUCN) protected area classification system (Dudley, 2008).

The IUCN framework categorizes protected areas according to their primary management objectives, ranging from strict nature protection to sustainable resource use. These categories were used to create the protection mask that distinguishes charcoal-free (protected) and charcoal-accessible (unprotected) zones in the spectral analysis.

Table 1 IUCN Protected Area Management Categories.

In subsequent analyses, categories Ia–IV were classified as protected, since categories V and VI allow certain forms of regulated resource use, including charcoal harvesting (Dudley, 2008).

Category	Designation	Primary Management Objective
Ia	Strict Nature Reserve	Strictly protected for biodiversity and geological/geomorphological features; human visitation, use, and impacts are strictly controlled.
Ib	Wilderness Area	Large unmodified or slightly modified areas retaining natural character without permanent or significant human habitation, protected to preserve their natural condition.
II	National Park	Large natural or near-natural areas set aside to protect large-scale ecological processes, species, and ecosystems, while allowing for compatible education and recreation.
III	Natural Monument or Feature	Areas set aside to protect a specific natural monument, such as a landform, sea mount, or geological feature of outstanding value.
IV	Habitat/Species Management Area	Managed mainly for conservation through active intervention to protect particular species or habitats.
V	Protected Landscape/Seascape	Areas where the interaction of people and nature has produced a distinct ecological, cultural, or scenic character; safeguarding integrity while allowing sustainable traditional use.
VI	Protected Area with Sustainable Use of Natural Resources	Areas that conserve ecosystems and habitats together with cultural values and traditional natural resource management systems; sustainable resource use is permitted if compatible with conservation objectives.

The initial phase of the analysis focused on exploring spectral differences between protected and unprotected areas by calculating three vegetation indices for each group. This step served to evaluate boundary conditions and identify preliminary contrasts in vegetation structure and condition. Based on this classification, the next analytical step examined how these two management zones differ in vegetation structure and physiological condition using spectral indices derived from Sentinel-2 imagery.

3.2.2 Vegetation indices

Vegetation indices derived from multispectral data provide simple yet powerful proxies for assessing vegetation condition, canopy density, and moisture dynamics. In this study, three commonly used indices the Normalized Difference Vegetation Index (NDVI), the Normalized Difference Moisture Index

(NDMI), and the Canopy Chlorophyll Index (CCI*)* were calculated from Sentinel-2 reflectance data to capture structural and physiological differences in vegetation cover.

Table 2 Summary of vegetation indices used from Sentinel-2 data.

	NDVI	**NDMI**	**CCI**
Purpose	Proxy for canopy greenness and photosynthetic activity	Indicator of canopy water content and moisture status	Sensitive to chlorophyll concentration and canopy pigment status
Formula	(NIR-RED)/ (NIR+RED)	(NIR-SWIR)/ (NIR+SWIR)	(NIR-GREEN)/ (NIR+GREEN)
Spectral Bands	B8 (NIR), B4 (RED)	B8 (NIR), B11 (SWIR)	B8 (NIR), B3 (GREEN)
Range	-1 to 1	−1 to +1	−1 to +1
Ecological meaning	High = dense, photosynthetically active vegetation; low = bare soil or degraded vegetation	High = moist, healthy vegetation; low = water stress, dry or senescent canopy	High = high chlorophyll content and canopy vitality; low = pigment stress or senescence
Reference	(Rouse Jr et al., 1974)	(Gao, 1996)	(Gitelson and Merzlyak, 1998)

After characterizing vegetation structure and physiological condition through vegetation indices and PCA, the final component of the analysis focused on quantifying tree species composition. To capture diversity patterns across ecological and spatial scales, a suite of diversity metrics.

3.2.3 Diversity metrics

After characterizing vegetation structure and physiological condition, I use diversity metrics to quantify how tree species composition varies across governance gradients. In ecological research, they can be used to compare community structure across sites, management systems, or disturbance gradients (Magurran, 2004). For this study, diversity indices were calculated for tree species clusters at both the national and local scales to assess how charcoal production and governance influence Miombo woodland diversity.

Diversity can be expressed in terms of Hill numbers, a unified framework that quantifies diversity (Jost, 2006). This approach allows traditional indices to be interpreted on a common scale and makes differences across sites more intuitive.

- **Species Richness (S)**
    - **Definition:** The number of distinct species in a sample or area (Hill number of order 0).

- **Formula:** $S = \sum_{i=1}^{S} 1$
- **Range:** 1 to the total number of species observed.
- **Ecological interpretation:** Captures total diversity but ignores relative abundance; sensitive to rare species.

- **Hill Number of Order 1 (D_1)**
  - **Definition:** The exponential of Shannon entropy; represents the *effective number of common species*.
  - **Formula:** $D_1 = e^{-\sum_{i=1}^{S} p_i \ln(p_i)}$
  - **Range:** 1 to *S*.
  - **Ecological interpretation:** Gives the number of equally abundant species needed to produce the observed diversity; moderately sensitive to rare species.

- **Hill Number of Order 2 (D_2)**
  - **Definition:** The inverse of Simpson's concentration index; emphasizes dominant species.
  - **Formula:**
  - $D_2 = \frac{1}{\sum_{i=1}^{S} p_i^2}$
  - **Range:** 1 to *S*.
  - **Ecological interpretation:** Represents the effective number of dominant species; sensitive to species dominance and evenness.

- **Evenness Based on Hill Numbers (E_1 and E_2)**
  - **Definition:** Evenness expresses how evenly individuals are distributed among species. Hill-based evenness measures (E_1, E_2) standardize D_1 and D_2 relative to species richness.
  - **Formulas:** $E_1 = \frac{D_1 - 1}{S - 1}, E_2 = \frac{D_2 - 1}{S - 1}$
  - **Range:** 0 to 1.
  - **Ecological interpretation:** Values near 1 indicate evenly distributed communities; lower values suggest dominance by few species.

Together, these indices describe complementary aspects of community structure. *S* captures species presence: D_1 and D_2 represent abundance-weighted diversity emphasizing common and dominant species, respectively; and E_1 and E_2 quantify how evenly individuals are distributed. Analyzing all five

metrics provides a detailed view of how charcoal production and governance systems influence tree community composition and ecological balance.

While classical diversity metrics such as S, D_1, and D_2 rely on discrete species identities and abundance data, the spectral data used in this study consist of continuous reflectance values across multiple bands. In such datasets, every pixel has a unique spectral signature, and calculating richness directly from pixel values would trivially yield maximum diversity (S = total number of pixels). To translate continuous spectral information into ecologically interpretable units, k-means clustering was applied to group pixels with similar spectral characteristics into discrete "spectral species." These clusters were then treated similar to biological species, allowing the computation of richness and diversity metrics within protected and unprotected areas, as well as CBNRM and OA. The clustering process that underlies this transformation is described in the following section.

3.2.4 K-Means clustering

K-means clustering (MacQueen, 1967) is an unsupervised machine learning algorithm that separates a dataset into k groups (clusters) such that each observation belongs to the cluster with the nearest mean (centroid). In this study, k-means was used to discretize continuous Sentinel-2 spectral data into a finite set of spectral types, each representing a distinct combination of canopy and soil reflectance characteristics.

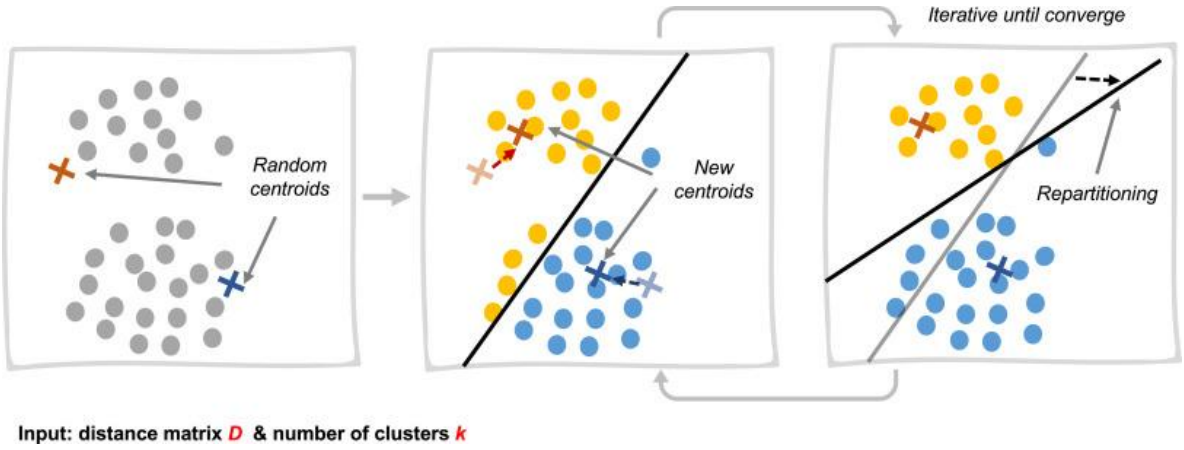

Figure 4 Illustration of K-means Clustering (Gao et al., 2023)

The estimation routine of K-means involves: (i) randomly initialize centroids of a pre-specified number of clusters and partition data points into groups according to their distance to the centroids; (ii) re-estimate the centroids (calculated as the mean of all the data points of the cluster) using points for each cluster; (iii) re-partition data into clusters; and (iv) iteratively repeat (ii) and (iii) until no more changes were observed in the location of centroids (convergence).

The number of clusters (k) determines the granularity of classification. Each resulting cluster was treated as a "spectral species," enabling the calculation of spectral richness (S) and diversity (D_1, D_2)

within the different governance categories. Although often illustrated in two dimensions for simplicity the k-means algorithm operates equivalently in higher-dimensional feature spaces. In this study, clustering was performed in a 10-dimensional spectral space defined by the selected Sentinel-2 bands, allowing for fine differentiation of vegetation reflectance patterns in their original, non-collapsed state (i.e., prior to PCA). The resulting clusters provided a discrete ecological basis for subsequent diversity calculations, linking continuous spectral variation to interpretable measures of richness, evenness and diversity across governance.

4 Data

This thesis integrates data from multiple independent sources to enable a combined analysis of charcoal production, forest governance, and ecological resilience in Tanzania. Broadly, two complementary domains of data were used: (i) spectral data derived from satellite imagery and spatial boundaries to capture ecosystem-level patterns at both national and local scales, and (ii) trait data derived from plant trait databases and forest inventory plots to characterize species-level functional differences. The data span different spatial scales and resolutions. At the national scale, remote sensing imagery was combined with vegetation maps and protected area datasets to assess spectral variability of Miombo woodlands under different governance regimes. At the local scale, satellite imagery was analyzed within village boundaries in Kilosa district to compare CBNRM with OA systems. The trait analysis, in turn, combined harmonized global plant trait data with Tanzanian forest inventory and species lists to explore functional trait spaces relevant to charcoal production and forest management. The following sections are organized by analytical domain. Section 4.1 describes the trait analysis data, including global plant trait data (TRY), Tanzanian forest plot data, and species reference lists. Section 4.2 introduces the data used for the spectral analysis, including vegetation extent, protected areas, village boundaries, and Sentinel-2 imagery.

4.1 Species and trait data

For the trait-based analysis, I used species-level functional trait data from the TRY Plant Trait Database and combined them with two national datasets: the National Forest Plot Data (Section 4.1.2) and the Tanzania Species List (Section 4.1.3). This integration ensured that only species occurring in Tanzania were included and that their local uses, such as charcoal production, were properly identified. The following subsections describe each dataset and how they were combined to construct the final trait dataset used in this study.

4.1.1 TRY data

The functional trait data used in this study were obtained from the TRY Plant Trait Database (Kattge et al., 2020), an open-access global repository of plant traits compiled from hundreds of individual datasets. TRY provides standardized trait observations across species, with harmonized units and metadata.

The traits present in this dataset encompass a wide range of plant functions, including leaf morphology and physiology (e.g., specific leaf area, nitrogen and phosphorus content, transpiration rate), seed traits (e.g., seed dry mass), and stem and wood traits (e.g., conduit density, stem diameter, wood density). These traits are ecologically relevant for understanding forest productivity, regeneration, and biomass

potential, and thus provide a functional basis for comparing species used in charcoal production with those not used.

4.1.2 National forest plot data

To ensure that the trait analysis only included species that occur within Tanzania, I cross-referenced the TRY trait dataset with the National Forest Inventory (NFI) species list (Rajala, 2022). The NFI, conducted under the National Forestry Resources Monitoring and Assessment of Tanzania (NAFORMA) program, is based on a systematic sampling design with 3,468 plots of 0.07 ha each, distributed at 5 km grid intervals across the country. In each plot, all trees with a diameter at breast height (DBH) ≥ 10 cm were measured and identified to species level. This inventory aimed to provides a comprehensive and nationally representative overview of Tanzania's forest composition and structure, allowing the TRY trait dataset to be filtered to species empirically observed in Tanzanian forests.

4.1.3 Tanzania species list

In addition to trait and plot data, this study used the NAFORMA Species List (MNRT et al., 2011) . The list was compiled by the Forestry and Beekeeping Division of the MNRT and provides detailed records of Tanzania's tree species, including information on their local uses. In this thesis, the list was used to identify and classify tree species that are utilized for charcoal production. Each species in the dataset was flagged as either "charcoal" or "non-charcoal" based on matches to this list, creating a binary classification that underpins the subsequent trait-based comparisons.

4.2 Study area extent

This analysis relied on four main data sources that together define the spatial extent and governance context of the Miombo woodlands. At the national scale, the Miombo extent was combined with protected area datasets to generate masks for the Sentinel-2 imagery. At the local scale, Miombo woodlands were intersected with village boundaries to delineate analytical units for comparing CBNRM and OA systems.

4.2.1 Miombo woodland extent

To determine the extent of Miombo woodland, I used the VegetationMap4Africa dataset (Lillesø et al., 2024), which provides a harmonized map of potential natural vegetation for Eastern Africa. The VECEA project was developed by the World Agroforestry Centre in collaboration with national partners, and integrates ecological, climatic, and floristic information to represent the vegetation types that would occur under natural conditions without major human disturbance.

For this study, only polygons classified as "Drier Miombo" (Wmd) and "Wetter Miombo" (Wmw) were retained, as these represent the ecologically dominant forms of Miombo woodland in Tanzania.

Polygons assigned to mixtures of Drier Miombo with North Zambezian woodland (Wmd/Wn) or Drier Miombo with Deciduous bushland (Wmd/Bd) were excluded to reduce potential spectral confusion and noise.

4.2.2 Protected areas

To classify Miombo woodlands according to governance regimes, I combined the VECEA vegetation layer with spatial data on protected areas in Tanzania. Protected area boundaries were compiled from WDPA-derived shapefiles which provide a global dataset of protected areas (UNEP-WCMC and IUCN, 2025). The WDPA assigns categories defined by the IUCN (see section 3.2.1) and designation types that distinguish levels of legal protection and land-use restrictions.

For this study, Miombo polygons were overlaid with the protected area layer, and each polygon was assigned one of three protection categories:

- **Strictly Protected**: Areas with legal frameworks that prohibit extractive uses such as charcoal harvesting, including IUCN categories Ia–IV and designations of national or international status.
- **Uncertain Protection:** Areas under partial or mixed governance, including IUCN Category VI, Game Reserves, Game Controlled Areas, and sites with missing or ambiguous IUCN classification. These areas may allow limited or regulated resource extraction.
- **Unprotected**: All Miombo woodland located outside any protected area boundary.

This classification provides a consistent framework for comparing Miombo woodlands under different levels of formal protection. Where overlaps occurred, strictly protected status was prioritized. The resulting dataset contained 745 Miombo polygons, of which 91 were classified as Strictly Protected, 248 as Uncertain Protection, and 406 as Unprotected. In terms of coverage, Strictly Protected areas accounted for approximately 150,200 km², while Uncertain areas covered about 224,200 km²; together, these categories represent roughly 42% of Tanzania's land surface. This spatially explicit classification forms the basis for the SVA by enabling comparison of Sentinel-2 signals across contrasting governance regimes.

4.2.3 Local study area and village boundaries

The local-scale analysis focused on six villages in the Kilosa District of central Tanzania. Kilosa lies approximately 150 km east of Morogoro and is representative of the sub-humid climate of central Tanzania, characterized by mean annual rainfall of 800–1200 mm and a pronounced dry season. The dominant vegetation type is Miombo woodland.

To delineate village boundaries, I used a spatial dataset of Tanzanian administrative units provided by van ’t Veen (van ’t Veen, 2022). The boundaries were intersected with the Miombo woodland extent (4.2.1) to define the analytical units for the local-scale analysis.

4.2.4 Remote sensing data

This study used Sentinel-2 Level-2A (surface reflectance) imagery from the Copernicus Data Space Ecosystem (CDSE) (European Union, Copernicus, 2024). Images were selected within 1 December 2024–31 January 2025, restricted to ≤20% scene cloud cover, and limited to one low-cloud scene per unique Military Grid Reference System tile (MGRS).

Table 3 Overview of the 13 spectral bands

These bands were included in the downloaded Level-2A data used in this study (European Space Agency, 2015).

Band	Central bandwidth (nm)	Resolution (m)	Description
B01	443	60	Coastal aerosol
B02	490	10	Blue
B03	560	10	Green
B04	665	10	Red
B05	705	20	Red edge 1
B06	740	20	Red edge 2
B07	783	20	Red edge 3
B08	842	10	NIR
B8A	865	20	Narrow NIR
B09	945	60	Water vapor
B10	1375	60	Cirrus
B11	1610	20	SWIR1
B12	2190	20	SWIR 2

The accompanying Scene Classification Layer (SCL) and metadata (e.g. cloud probability, viewing geometry, and quality indicators) were also included in the download package but used only during preprocessing.

5 Methods

This chapter details the two-part analytical workflow used to test the study's hypotheses. First, a trait-based stream quantifies differences in functional composition between charcoal and non-charcoal species using TRY data and multivariate statistics (PCA, PERMANOVA). Second, a remote-sensing stream uses Sentinel-2 reflectance and vegetation indices to assess spectral variability across governance contexts at national and village scales. Together, these complementary approaches link species-level trait structure to landscape-level canopy patterns.

5.1 Trait analysis

This section describes the workflow used to analyze the functional trait composition of tree species relevant to charcoal production in Tanzania. The analysis was structured hierarchically, moving from data preparation to multivariate evaluation. The first part (Sections 5.2.1–5.2.4) details the data acquisition, cleaning, and harmonization steps, including the integration of TRY trait data with national forest inventory records and the derivation of species-level trait means. The second part (Sections 5.2.5–5.2.8) focuses on how to implement PCA. Finally, Section 5.2.9 presents the statistical analyses (PERMANOVA and multivariate dispersion) used. The code used to execute this work flow is in the 11Appendix.

5.1.1 Filtering

The TRY data request (see section 4.1.1) contained 36 different functional traits spread across 2'078'926 observations. Coverage was highly uneven among traits: seed dry mass had the greatest representation (314,707 observations spanning 43,073 species), whereas leaf midvein support tissue thickness was sparsely sampled (87 observations across 22 species).

To ensure ecological relevance to Tanzania's Miombo woodlands, we restricted the species pool using the National Forest Plot Inventory (NFPI) checklist (see section 4.1.2) This step confines inference to taxa documented for Tanzania and aligns the trait space with the regional flora.

5.1.2 Trait harmonization and quality assurance

At import, I applied a global quality policy: I retained records with TRY "ErrorRisk" < 4 (or missing) and excluded all records with "ErrorRisk" ≥ 4. This policy was enforced consistently throughout the workflow.

For all traits, I used TRY's trait-wise standardized values ("StdValue") to ensure cross-study comparability. For two traits with heterogeneous unit reporting—wood density and stem diameter—I increased usable coverage when "StdValue" was missing by falling back to TRY's original measurement

fields ("OrigValueStr" and "OrigUnitStr"). I normalized unit strings and converted values to canonical units:

- Wood density → g cm^{-3}. Treated g mL^{-1}, kg L^{-1}, kg dm^{-3}, t m^{-3}, and mg mm^{-3} as equivalent to g cm^{-3} (1:1). Values reported in kg m^{-3} were divided by 1000.
- Stem diameter → cm. Values in m were multiplied by 100; values in mm were divided by 10; values already in cm were retained.

This produced a unified numeric column ("StdValue_filled") that uses "StdValue" when present and the converted original value otherwise (for these two traits only). I screened distributions and unit mixes visually to flag anomalies but no additional trimming beyond the unit conversions and the "ErrorRisk" policy was applied.

Because PCA requires quantitative and continuous variables, I restricted the analysis to TRY traits with numeric standardized values and excluded binary or categorical descriptors (e.g., presence–absence variables or class labels such as growth form or leaf habit). After these steps, the dataset comprised 35’781 observations across 566 species.

5.1.3 Attribution of traits to species

I classified species with the Tanzania species list (Section 4.1.3), which provides "Latin name", "Vernacular name(s)", and a “Notes” field that remarks what a species is used for. Any species tagged as "firewood" under "Notes" was treated as charcoal-producing; all others were assigned to non-charcoal. I joined this label to the TRY table by "AccSpeciesName", creating a new "charcoal" attribute. First, I matched species via "Latin name". Then, I parsed "Vernacular name(s)" for Latin binomials and added those matches which contributed to 7 additional species. This approach declared 350 species as charcoal species and 336 species as non-charcoal species.

5.1.4 Trait renaming and aggregation

To improve readability and reproducibility in figures and reports, long descriptive trait names were systematically converted to concise short labels. This mapping included, for example, “Stem conduit density (vessels and tracheids)” to “Stem_conduit_density”, “Stem diameter (cm)” to “DBH”, and “Leaf area per leaf dry mass (specific leaf area, SLA or 1/LMA): undefined if petiole is in- or excluded” to “SLA_undefined”.

After trait naming was standardized, the data was aggregated to the species × trait level. For each species and trait, I calculated the arithmetic mean of all available standardized trait values. The resulting wide-format matrix contained one row per species and one column per trait, with the

additional binary column for charcoal. This matrix represents the mean functional trait profile for each species and serves as the foundation for subsequent principalcomponent analysis.

5.1.5 Traits available and the trade-off between trait and species coverage

After cleaning and harmonization, a total of 17 continuous traits were available, covering key plant functional categories relevant to forest structure and charcoal production systems.

- Leaf traits (e.g., nitrogen and phosphorus content, specific leaf area) describe photosynthetic capacity and resource-use strategies.
- Stem and wood traits (e.g., wood density, conduit density, stem diameter) capture hydraulic efficiency and biomass allocation.
- Seed traits (e.g., seed dry mass) reflect reproductive and dispersal strategies.

PCA requires complete data across all included traits, this led to a steep trade-off: including more traits resulted in fewer species with complete observations. For example, seed dry mass was available for 438 species, whereas combining it with wood density reduced the overlap to about 250 species.

This relationship is illustrated in Figure 5, which shows how increasing the number of traits rapidly decreases the number of species that can be retained for PCA.

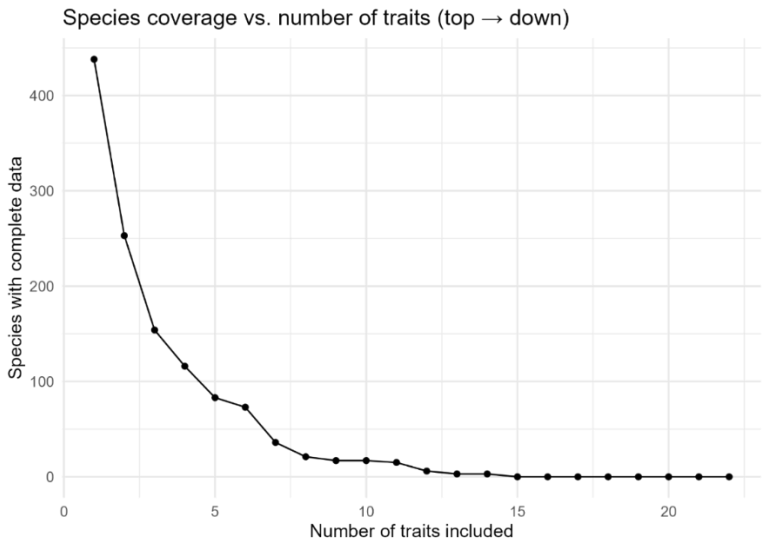

Figure 5 Trait/Species coverage curve

This curve highlights the trade-off between the number of traits included and the number of species with complete data. The line represents the number of species that could be incorporated into a PCA as more traits are added.

5.1.6 Rationale for trait selection and use of the Boruta algorithm

Because this study aimed to compare the functional trait spaces of charcoal and non-charcoal species, it was important to retain a sufficient number of species while ensuring that the selected traits captured relevant ecological information. Therefore, three complementary strategies were applied to define different PCA trait sets:

1. Data availability approach: Selecting traits that maximizes the number of species included.
2. Ecological relevance approach: Selecting traits that represent the full range of major ecological functions relevant to forest productivity, regeneration, and wood quality.
3. Data-driven approach: selecting traits identified as most informative using the *Boruta* feature selection algorithm.

The Boruta algorithm was applied to identify which functional traits most effectively discriminate between species used and not used for charcoal production. This was done using the species-by-trait mean matrix, with the binary variable charcoal (levels: “non-charcoal” and “charcoal”) as the response and all numeric traits as predictors.

The analysis was implemented in R using the Boruta package with a fixed random seed (123) for reproducibility (Kursa and Rudnicki, 2010). The algorithm iteratively evaluated the importance of each trait within a Random Forest classification, comparing real traits against randomly permuted “shadow” features. Traits consistently outperforming their shadow counterparts were labeled confirmed, those of uncertain importance tentative, and uninformative ones rejected.

Out of all tested variables, five traits were confirmed or tentatively important: specific leaf area (SLA_undefined), wood density, leaf nitrogen per area (Leaf_N_area), leaf nitrogen per mass (Leaf_N_mass), and seed mass. The results of the Boruta algorithm are shown in Figure 6, where boxplots illustrate each trait’s importance relative to the shadow features.

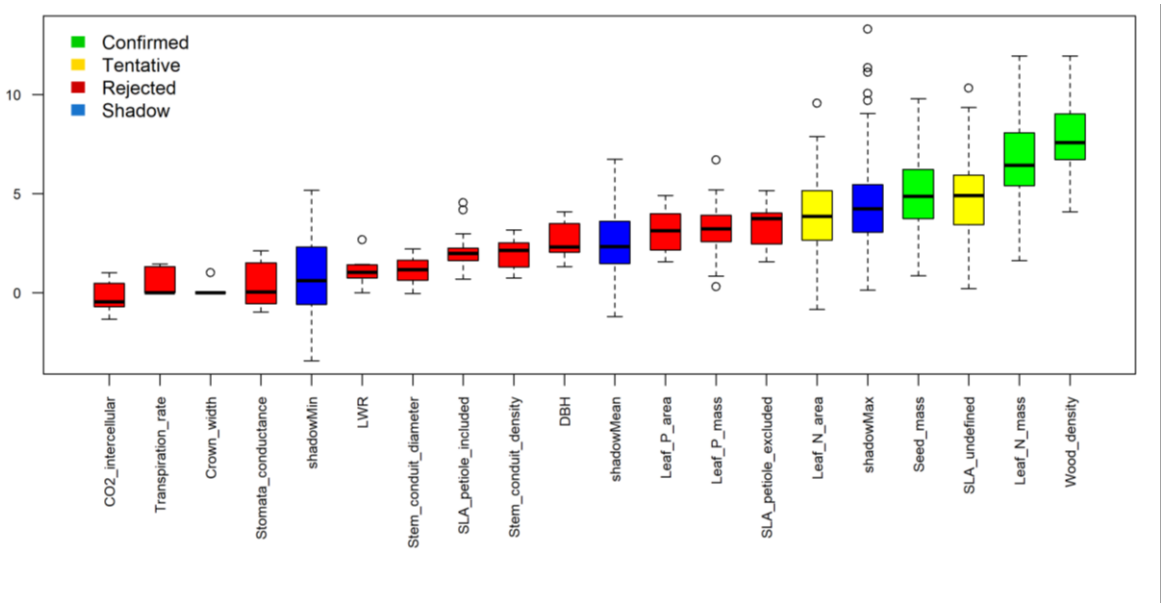

Figure 6 Boruta feature selection results.

Boxplots show trait importance ranks relative to random “shadow” features. Traits in green (confirmed) and yellow (tentative) were retained for further analysis, while red traits were rejected form the PCA.

5.1.7 Overview of PCA configurations

The different PCA runs were designed to balance trait completeness, ecological coverage, and statistical relevance. The following four configurations were analyzed:

- Top 5 traits: The five traits with the highest species coverage to maximize sample size
    - As a sensitivity measure this PCA was also run only including species that had at least 3 observations for each trait.
- Top 9 traits: this PCA maximized the number of traits included.
- Ecologically important traits: a curated selection of five traits representing essential functional dimensions of Miombo species, including growth, survival, and wood quality.
- Boruta-selected traits: the five traits identified as most informative by the Boruta algorithm, representing a data-driven selection emphasizing discrimination between charcoal and non-charcoal species.

Table 4 Traits included in each PCA configuration.

The number of species for which each trait was available and the number of observations for each species. (SLA = specific leaf area, N= Nitrogen, P=Phosphorus)

Trait name	Number of Species	Number of Observations	Top 5	Top 9	Eco	Boruta
Seed dry mass	438	5451	X	X		X
Wood density	356	4039	X	X	X	X
Leaf N per mass	232	4911	X	X	X	X
Leaf P per mass	167	873	X	X		
SLA (undefined)	146	6088	X	X	X	X
Leaf N per Area	146	3503		X		X
Stem conduit density	122	292		X	X	
SLA (petiole excluded)	117	838		X		
SLA (petiole included)	113	1391		X		
DBH	93	4610			X	

5.1.8 Execution of principal component analysis

To compare the functional trait composition of charcoal and non-charcoal species, PCA was applied. PCA is well suited for this purpose because it reduces multidimensional trait information into a few synthetic axes, allowing complex functional variation to be visualized in two-dimensional trait spaces. These visualizations make it possible to detect potential shifts or separations in trait composition between species groups.

For each trait set, the corresponding subset of the species-by-trait matrix (including the binary charcoal classification) was extracted. All numeric traits were z-scored (mean-centered and scaled to unit variance) to ensure that traits measured in different units contributed equally to the analysis.

PCA was then conducted in R using the prcomp() function on the standardized matrix. The analysis produced:

- Trait loadings, representing correlations between the original traits and principal components,
- Species scores, indicating each species' position in the reduced trait space, and
- Variance explained, quantifying how much trait variation is captured by each principal component.

The first two components (PC1 and PC2) were retained for visualization, as they captured the dominant gradients of functional variation among species and defined a two-dimensional trait space in which the relative positions of charcoal and non-charcoal species could be compared to reveal potential shifts in functional composition.

5.1.9 Statistical analysis: PERMANOVA and dispersion

To statistically test whether charcoal and non-charcoal species occupy distinct regions of functional trait space, I applied a Permutational Multivariate Analysis of Variance (PERMANOVA) using the adonis2() function from the vegan package in R.

PERMANOVA was chosen because it allows hypothesis testing on multivariate trait data without assuming normality, making it well suited for ecological datasets with mixed-scale and correlated traits. In this context, it tests whether the centroids (multivariate means) of the two groups, charcoal and non-charcoal species, differ significantly in the multidimensional trait space.

The analysis was performed on Euclidean distances computed from the standardized (z-scored) trait matrix. Each model used 9,999 permutations to assess the significance of the group effect. The output included the F-statistic, the coefficient of determination (R^2), indicating the proportion of total trait variance explained by group identity, and the p-value, which quantifies the probability of observing such group separation by chance.

To complement this, I tested for homogeneity of multivariate dispersion using the betadisper() function. This test evaluates whether the groups differ in their internal variability, that is, how widely species are spread around their group centroid in trait space. Significance was assessed via permutation tests on the average within group distances.

This step ensured that significant PERMANOVA results reflected true differences in centroid position (i.e., shifts in mean trait composition) rather than differences in within group variability (e.g., one group being more functionally heterogeneous than the other). Ecologically, this distinction is important because a centroid shift indicates that charcoal and non-charcoal species differ in their dominant functional strategies, whereas a dispersion difference suggests that one group encompasses a broader or narrower range of functional traits.

5.2 Remote sensing analysis

The remote sensing analysis complements the trait-based approach by quantifying vegetation structure and condition across spatial scales. Using Sentinel-2 multispectral data, it assesses how governance regimes influence spectral variability, vegetation indices, and canopy diversity in Tanzania's Miombo woodlands. The code for all sections can be found in the Appendix.

5.2.1 Sentinel-2 data acquisition

To obtain Sentinel-2 Level-2A surface reflectance imagery for Tanzania, I developed a Python-based data acquisition workflow using the Copernicus Data Space Ecosystem (CDSE). The workflow was executed in a Jupyter environment and automated the search, filtering, and download of suitable satellite scenes.

The area of interest (AOI) was defined as a bounding box covering the full extent of Tanzania (approximately 29.64°E to 40.40°E and 11.69°S to 2.01°S), encompassing all Miombo woodland regions relevant to this study. The search window was constrained to December 1, 2024 through January 31, 2025 to ensure consistent phenological conditions across tiles.

The script queried the CDSE Sentinel-2 catalogue for all Level-2A products intersecting the AOI and date range. From the retrieved metadata, unique Military Grid Reference System (MGRS) tile identifiers were extracted, and for each tile, the scene with ≤ 20% cloud cover was selected. This ensured low-cloud, seasonally consistent coverage while minimizing temporal heterogeneity.

All available spectral bands were downloaded, but for subsequent analyses only the bands natively provided at 20 m spatial resolution were used: B01 (443 nm, coastal aerosol), B02 (490 nm, blue), B03 (560 nm, green), B04 (665 nm, red), B05 (705 nm), B06 (740 nm), B07 (783 nm) — red-edge bands, B8A (865 nm, near-infrared), B11 (1610 nm, shortwave infrared 1), and B12 (2190 nm, shortwave infrared 2). No band resampling or recalculation was performed, ensuring that all results reflect original Sentinel-2 surface reflectance data. In total, approximately 160 Sentinel-2 L2A scenes were acquired, covering the entire extent of Tanzania.

5.2.2 Selection of satellite data

The Miombo Vegetation Layer (see section 4.2.1) and the Protected Areas Layer (see section 4.2.2) were combined to create a protection classified miombo extent which builds the basis for the subsequent SVA, ensuring that remote sensing signals could be consistently related to management context. All processing was conducted in R using the *sf* package. The coordinate reference systems of both input layers were aligned (WGS 84), and all geometries were validated to prevent topological errors.

To spatially assign protection status to each Miombo polygon, the core Miombo layer was intersected with the protected area boundaries. The intersection operation (st_intersection()) subdivided polygons where only part of a Miombo polygon overlapped a protected area, ensuring that every resulting geometry inherited the correct protection attribute.

Next, all Miombo polygons located outside any protection boundary were identified through a spatial difference operation (st_difference()), which extracted the non-overlapping portions of the Miombo extent. These polygons were subsequently assigned the protection class “Unprotected”.

Finally, the protected and unprotected subsets were combined into a single, harmonized dataset (bind_rows()), containing a unified attribute field named protection, which categorizes each polygon as either Strictly Protected, Uncertain Protection, or Unprotected.

The final dataset comprising of 745 Miombo polygons (91 Strictly Protected, 248 Uncertain Protection 406 Unprotected) was overlaid laid with the tile Id of the sentinel download to exclude tiles that did not intersect with the Miombo shapefile.

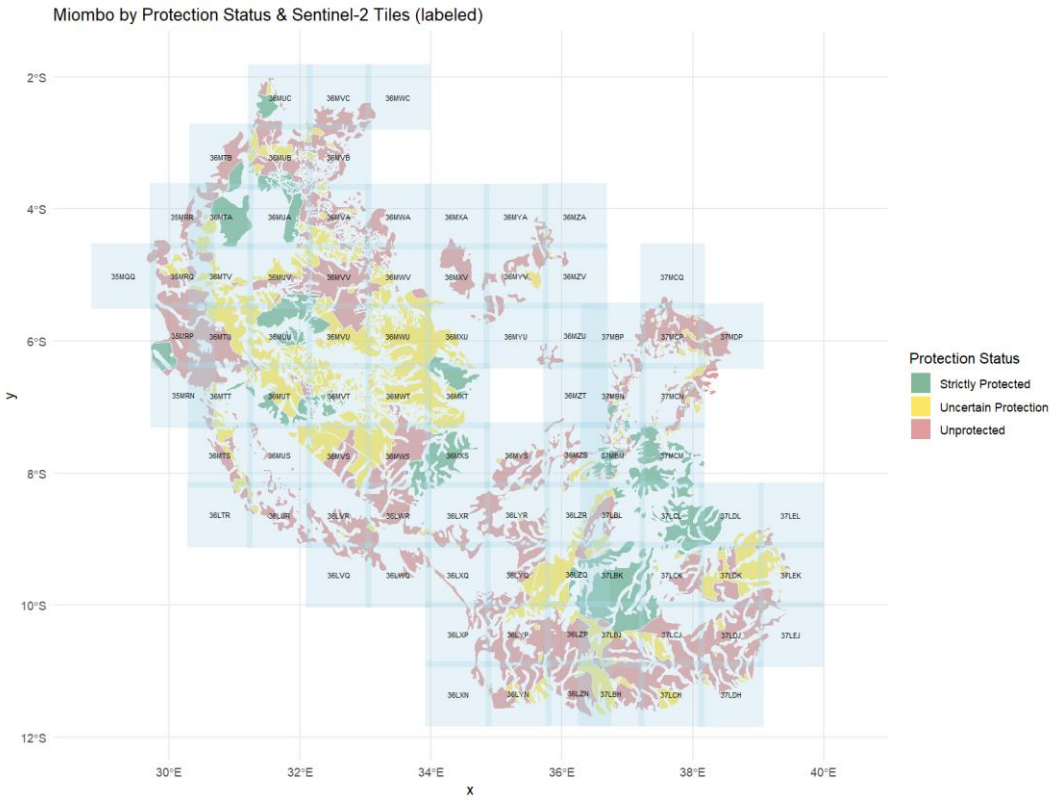

Figure 7 Map of Miombo in Tanzania

Overlayed with the corresponding Sentinel-2 tile IDs and protection classes that built the foundation of the analysis.

5.2.3 National analysis: Sentinel-2 preprocessing and sampling

The pixels used of the Sentinel-2 tiles covering Tanzania’s Miombo woodlands (Figure 7) have a spatial resolution of 20 m, which collectively represents approximately four billion pixels. Processing all pixels at this resolution would have been computationally infeasible. To address this, a two-run sampling

strategy was implemented to obtain a balanced and representative subset of Miombo pixels for the SVA. The first run calculated the proportion of pixels for each tile with the following four steps.

1. The Miombo polygons were reprojected to match the tile's coordinate system.
2. The SCL raster was cropped and masked to the Miombo extent.
3. The total number of Miombo pixels and the number of valid (clear-sky) pixels were counted, both overall and separately for the two protection categories.
4. Summary statistics were stored in a table, providing a full audit of data availability and quality across the country.

The resulting dataset quantified how much usable Sentinel-2 coverage was available for each tile before further processing. Once the valid Miombo pixels had been identified, the next step was to generate sample for analysis.

The second step was to open each tile and draw 50 random samples according to the size determined in run one. This made sampling much faster as each individual tile only needed to be opened once. During sampling, pixels were drawn randomly from areas classified as Strictly Protected and Unprotected Miombo woodlands, using clear-sky pixels only (as defined by the Sentinel-2 Scene Classification Layer). The reflectance values for each selected pixel were extracted from the ten 20 m Sentinel-2 bands available

Each sampled pixel was stored along with its tile code, spatial coordinates, replicate ID, and group label (Protected or Unprotected) in a compressed Parquet data structure, referred to as the sample bank. This sampling process was fully automated in R and designed to ensure reproducibility, proportional representation of governance classes, and minimal spatial redundancy among neighboring pixels. The resulting data bank provided the input for the subsequent analysis.

5.2.4 National scale: Vegetation indices calculation

To assess baseline differences in vegetation conditions between governance regimes, three vegetation indices were calculated directly from the sampled Sentinel-2 reflectance values stored in the data bank. Using R and the Arrow framework for efficient in-memory processing, the following indices were derived for each replicate (rep_id) and governance group (Protected vs. Unprotected):

- NDVI = (B8A − B04) / (B8A + B04)
- NDMI = (B8A − B11) / (B8A + B11)
- CCI = mean [(B8A/B05 − 1), (B8A/B06 − 1), (B8A/B07 − 1)]

Per-run mean values were computed for each group using group_by() and summarise() within Arrow, ensuring that all 20 replicates were processed consistently without loading the full dataset into memory. The resulting per-group, per-run averages were exported to a CSV file and visualized as boxplot to depict the distribution of vegetation index means across replicates.

Additionally, to visualize the variability and consistency of vegetation conditions across sampling runs, the distributions of vegetation indices were analyzed. Using R and the Arrow framework, each replicate (rep_id) was processed separately to compute pixel-wise index values for the *Protected* and *Unprotected* groups. The resulting per-pixel distributions were visualized through violin plots, which illustrate both the range and central tendency of vegetation indices within each governance category.

5.2.5 Local scale: classification, preprocessing and vegetation indices

I delineated the local Area of Interest (AOI) by intersecting village boundaries with the mapped Miombo extent. Village polygons were cleaned, made valid, and re-projected to the Miombo layer's CRS, each village was tagged to one of two governance classes based on name matching: CBNRM (Kigunga, Ulaya Mbuyuni, Ulaya Kibaoni) and OA (Kidete, Msowero). We then computed the geometric intersection between villages and the Miombo woodland shapefile to retain only Miombo inside villages, and exported three layers: all Miombo within villages, CBNRM Miombo, and Open-Access Miombo. This step mirrors the national pipeline's masking to governance categories but is constrained to the five study villages.

Per-pixel extraction and bank construction. For each Sentinel-2 L2A tile, I located the ten spectral bands used throughout the study — B01 (443 nm), B02 (490 nm), B03 (560 nm), B04 (665 nm), B05 (705 nm), B06 (740 nm), B07 (783 nm), B8A (865 nm), B11 (1610 nm), and B12 (2190 nm), corresponding to the 20 m resolution bands. These bands were chosen because they capture the key portions of the electromagnetic spectrum relevant to vegetation monitoring. After projecting the CBNRM and Open-Access AOIs to the tile CRS (terra), I cropped/masked the bands by AOI and by SCL validity (retaining only non-problematic classes; mask codes: 0, 1, 3, 8, 9, 10, 11). All remaining pixels were exported to a local parquet "bank" with coordinates, group label (CBNRM/OpenAccess), and band values; no subsampling or rarefaction was applied locally. Consistent with the national workflow, indices were computed from the same bands; for quality control I derived vegetation indexes per pixel and plotted group-wise violin graphs. Additionly here I also visualized the vegetation metrics as a map of the study area.

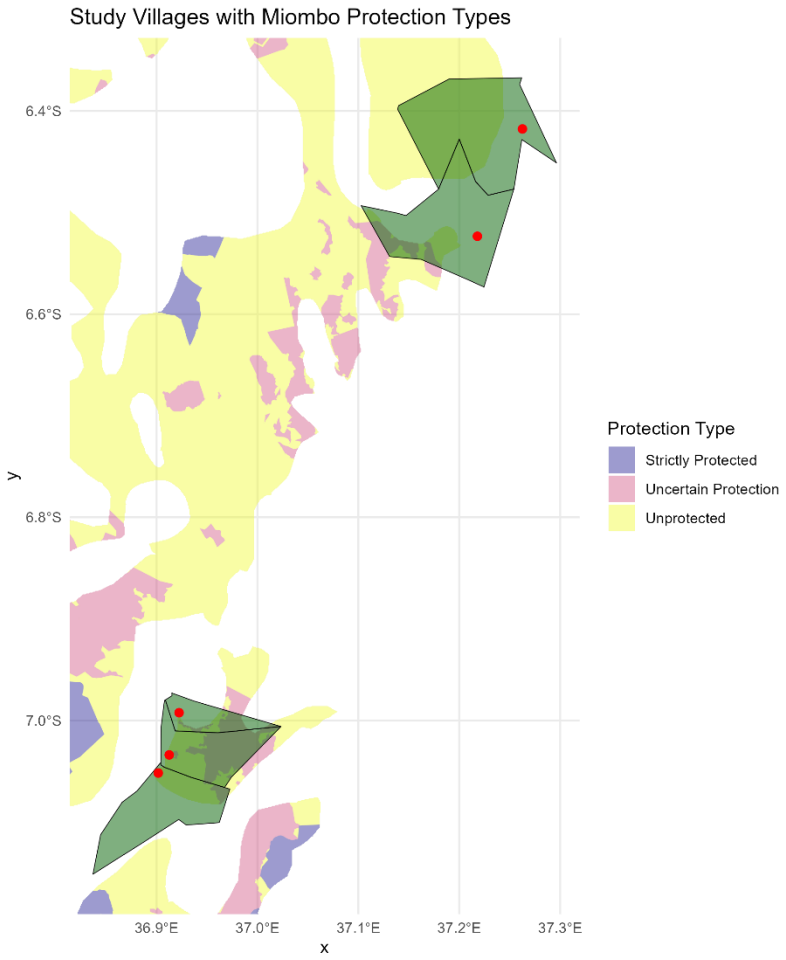

Figure 8 Miombo woodland extent with village boundaries.

Villages are classified by governance type into CBNRM (bottom green polygons) and *OA* (top green polygons) systems. Red points indicate village centers used for reference. This map defines the local Areas of Interest (AOIs) from which Sentinel-2 pixels were extracted for the local analysis.

5.2.6 Principal component analysis of Sentinel-2 data

A PCA was performed to examine the variability in Sentinel-2 reflectance data and to assess whether *Protected* and *Unprotected* Miombo woodlands reflectance separates at the national scale. The analysis used ten Sentinel-2 Level-2A bands at 20 m spatial resolution (B01–B07, B8A, B11, B12), covering the visible to short-wave infrared spectrum.

Data for each run was loaded directly from the standardized bank dataset in Parquet format (using the arrow package in R). For this analysis, a single run (Run 12) with 50 000 pixels was selected at random. Digital numbers were scaled to surface reflectance (DN / 10 000), and only complete observations were retained.

The PCA was fitted using the prcomp() function in R, with centering and unit variance scaling applied to all bands. The model was restricted to ten components (rank. = 10) corresponding to the number of

input bands. To ensure consistent axis orientation across runs, the sign of each principal component was standardized such that the sum of its loadings was positive.

Scores for the first two components (PC1 and PC2) were extracted for plotting and further statistical analyses. Variance explained by PC1 and PC2 was calculated as the proportion of the total eigenvalue sum, and loadings were exported as .csv files for traceability. It can be found in Appendix A.

Group centroids were computed in the two-dimensional PCA space defined by the first two principal components (PC1 and PC2). For each governance type ("Protected" and "Unprotected"), the centroid was obtained by averaging all sample coordinates along PC1 and PC2, respectively. These centroids represent the mean spectral position of each group within the reduced feature space. Several metrics were calculated to assess whether clusters were separable: Euclidean centroid distance, Mahalanobis distance, Bhattacharyya distance, and Jeffries–Matusita distance.

I chose these metrics because they capture complementary aspects of class separability in multivariate space. The Euclidean centroid distance provides a simple measure of the overall displacement between group means, while the Mahalanobis distance accounts for the covariance structure within each group, thereby reflecting separability relative to internal variability. The Bhattacharyya and Jeffries–Matusita distances further quantify distributional overlap between groups, integrating both mean differences and class dispersion.

This PCA was repeated using a targeted 5-band subset emphasizing vegetation sensitivity B05, B06, B07 (red-edge 1–3), B8A (narrow NIR), and B04 (red) for the same run. I chose to do this to test whether band combinations most responsive to vegetation biophysical properties improve separability between governance types, thereby emphasizing differences in vegetation condition and structure rather than overall spectral variance.

For visualization, three complementary plots were generated for both band sets (10 Bands and 5 Bands):

1. A full scatter plot displaying individual pixels color coded by governance category, including 95 % confidence ellipses and centroid markers.
2. A per-group 2D density plot showing the spectral density distribution of each group overlaid in PC1–PC2 space.
3. A loading bar plot, depicting the contribution of each Sentinel-2 band to PC1 and PC2, highlighting the dominant spectral regions driving variance.

At the village level, a local PCA was performed following the same procedure as the national analysis but applied exclusively to Sentinel-2 pixels extracted within the CBNRM and OA areas. Unlike the

national-scale PCA, the local analysis utilized all available pixels within the respective village boundaries, thereby capturing the full spectral variability present in each governance type without prior subsampling. This approach ensures that the local analysis reflects the actual spatial and spectral structure of the CBNRM and OA areas. Reflectance values were centered and standardized (mean = 0, SD = 1) prior to analysis, ensuring that all bands contributed equally to the covariance structure.

Principal components were computed using prcomp(), and the first two axes were used to visualize and compare spectral trait spaces between governance types. As in the national workflow, PCA orientation and scaling conventions were retained to ensure methodological consistency across scales. Variance explained by the first two components (PC1, PC2) and their cumulative contribution to total spectral variance were recorded, along with the loadings for all input bands. The outputs of plots and metrics were generated analogously to the national PCA.

5.2.7 K-means clustering

To characterize spectral “types” and compare their composition between governance groups, I ran unsupervised k-means clustering on 10-band Sentinel-2 reflectance (B01–B07, B8A, B11, B12) using a fixed replicate (Run 12). From the national spectral bank, I drew a training set capped at 30,000 pixels per group (Protected/Unprotected) and an evaluation pool capped at 50,000 per group. The per-group caps were applied to ensure a balanced and computationally efficient sampling of pixels across governance types while maintaining sufficient representation of spectral variability within each group. This approach avoids bias from unequal group sizes and prevents overfitting during k-means clustering due to excessively large sample sizes.

An analogous clustering procedure was conducted at the local scale using Sentinel-2 tiles covering CBNRM and OA areas. From this point onward, all analytical steps—including z-score standardization, balanced sampling, cluster assignment, and diversity metric computation—were performed in an equivalent manner for both the national and local analyses to ensure methodological consistency across scales.

All spectral bands were standardized by z-scoring (subtracting the mean and dividing by the standard deviation of the training data), and the same parameters were subsequently applied to the evaluation pool (functions: zscore_fit(), zscore_apply()). Standardization ensures that all bands contribute equally to the clustering process by removing differences in scale and magnitude, which can otherwise cause variables with larger numeric ranges to dominate the distance calculations in k-means and PCA-based analyses (Jolliffe and Cadima, 2016). Using scaling parameters derived from the training data prevents information leakage into the evaluation set and ensures consistent data normalization between phases.

Unsupervised k-means clustering (kmeans()) was then applied to the standardized reflectance data in the 10-band feature space to partition spectral variability into K clusters. The algorithm was initialized with multiple random starts (nstart = 5) and iterated up to 100 times (iter.max = 100) to improve convergence stability. Pixels in the evaluation pool were subsequently assigned to the nearest cluster centroid in z-space using a custom nearest-center routine (predict_kmeans()).

To ensure comparability between governance groups, the evaluation pool was rarefied to equal sample sizes per group (rarefy_equal()), after which cluster proportions were computed for Protected and Unprotected Miombo pixels.

Rarefying balances sample sizes and thereby removes the influence of unequal pixel counts on compositional metrics such as cluster richness. This ensures that observed differences in spectral diversity reflect true compositional variation rather than artifacts of unequal sampling intensity (Gotelli and Colwell, 2001).

The trade-off is that rarefaction involves subsampling from the larger group, which may exclude some spectral variability present in that group. However, in this context, equalization was necessary to compare spectral compositions on a per-pixel basis under consistent sampling effort, following standard practices in ecological diversity analysis and remote sensing community comparisons (Chao and Jost, 2012).

To interpret differences in cluster composition in spectral and ecological terms, cluster centers (stored in standardized *z*-space) were back-transformed to surface reflectance values using the saved *z*-score parameters (kmeans_model_zscore.rds). From these reflectance centroids, three vegetation indices were computed, providing a concise description of the vegetation greenness (NDVI), canopy moisture (NDMI), and chlorophyll content (CCI) associated with each cluster.

To visualize these patterns, clusters were sorted by the absolute difference in relative abundance between governance types (|Δ proportion|), highlighting clusters most distinct between Protected and Unprotected Miombo. A combined bar chart (showing Δ proportions) and heatmap (showing NDVI, NDMI, and CCI values) were used to display the spectral and ecological differentiation among clusters.

Analogous to the national analysis, the local *k*-means clustering was performed on the complete set of Sentinel-2 pixels extracted within the CBNRM and OA areas, without prior subsampling. Clustering was based on standardized reflectance values from ten Sentinel-2 bands. Although all available pixels were included initially to capture the full local spectral variability, the resulting cluster compositions were subsequently balanced to equal sample sizes per governance type before computing diversity, evenness, and distance metrics. This ensured comparability between groups while retaining the full spatial detail of the local dataset.

5.2.8 Richness sensitivity across runs

To evaluate the robustness of richness differences, I conducted a sensitivity analysis across varying clustering resolutions and presence thresholds. This was done to test whether observed differences in spectral richness between governance types were consistent across different levels of spectral granularity (*K*) and inclusion criteria (τ). Assessing sensitivity in this way helps confirm that richness patterns are not artifacts of specific parameter settings but reflect stable differences in spectral diversity between groups. The number of clusters (K) varied over six values (40, 60, 80, 100, 120, 150), and richness was recalculated at six presence thresholds (τ =0, 0.5%, 1%, 2%,3%, 5%). For each combination of K and τ, the difference in richness between Unprotected and Protected samples (ΔS) was summarized across 20 independent replicates.

For every cell in this parameter grid, I calculated the mean ΔS, its standard deviation (SD), standard error (SE), and corresponding 95 % confidence interval (CI = mean ± 1.96 × SE). These summaries were visualized in a heatmap, where the x-axis represents the clustering size (K) and the y-axis the presence threshold (τ). The tile colors indicate the mean richness difference (ΔS) using a diverging color ramp (blue = Protected > Unprotected, white = no difference, red = Unprotected > Protected). Each tile also reports the numerical values of the mean, CI width, and SD, allowing both the direction and variability of results to be assessed at a glance.

5.2.9 PCA-based spectral richness mapping

To spatially represent the distribution of spectral richness in the principal component (PC) space, the two leading PCA axes were subdivided into a 100 × 100 regular grid, each cell representing a defined interval of PC1 and PC2 values. Every pixel was assigned to a grid cell based on its PC coordinates and contributed to the cell's composition according to its governance group (Protected or Unprotected) and cluster membership from the k-means model. Within each occupied cell, the relative proportions of clusters were computed and converted into richness values according to the defined presence thresholds (τ = 0.5 %, 2.0 %, 5.0 %). Cells that contained fewer than the minimum number of valid pixels were excluded and rendered as a neutral gray background, ensuring that mapped areas represent only statistically stable portions of the PC space. This mapping provides an overview of where in the spectral space richness differences tend to emerge, allowing visual assessment of how spectral richness is distributed across PC space.

5.2.10 Diversity and evenness metrics

To enable the computation of diversity metrics analogous to ecological community analysis, I first summarized the spectral clustering outputs into richness tables. In these tables, each spectral cluster (derived from k-means classification in PCA space) was treated as an analogue of an ecological plot,

while the proportional occurrence of pixels from the two governance types (Protected and Unprotected) within each cluster was used as a measure of relative abundance, comparable to species counts. For each cluster, the proportion of pixels belonging to Protected and Unprotected areas was extracted, and binary presence values were assigned based on the defined presence thresholds (τ = 0.5 %, 2.0 %, 5.0 %). These binary and proportional data formed the basis of richness tables, from which the total number of occupied clusters (richness, S) was derived for each governance group.

By interpreting spectral clusters as ecological sampling units and the governance group–specific proportions as abundance values, this approach allowed the application of Hill number diversity metrics (D_1, D_2) and evenness measures (E_1, E_2). These metrics provide complementary perspectives on spectral composition: richness (S) captures the number of distinct spectral "types," diversity (D_1, D_2) accounts for both richness and the relative evenness of their distribution, and evenness (E_1, E_2) quantifies how uniformly spectral types are represented within each governance group. Together, they serve as spectral analogues to ecological community indices, enabling comparison of structural and compositional complexity between Protected and Unprotected Miombo.

Following the sensitivity analysis of richness across clustering resolutions, K = 100 was selected as the focus for subsequent analyses because richness differences stabilized beyond this point, indicating that higher K values did not substantially alter the observed spectral patterns. This allowed a targeted examination of threshold effects (τ) while maintaining an interpretable level of spectral partitioning.

This framework directly links to the research question on tree species composition by translating spectral heterogeneity into measures of ecological diversity. Higher spectral richness or diversity suggests a broader range of canopy conditions and potentially greater species or structural variety, whereas lower values indicate homogenization or dominance of fewer spectral types. The visualizations were designed to support this interpretation, with violin and box plots showing the variability in richness among clusters, bar charts summarizing total richness (S), and panels of D_1, D_2, E_1, and E_2 depicting how spectral diversity and evenness vary across thresholds and governance types.

6 Results

This chapter presents the empirical findings of the study and is structured to answer the two research questions introduced in Chapter 2. The results are organized in three parts, reflecting the multi-scale analytical design that links species-level functional traits with landscape-scale spectral and diversity patterns across different governance contexts.

6.1 Trait differences between charcoal and non-charcoal species

Charcoal-producing species show partial functional differentiation from non-charcoal species. This part addresses the functional trait dimension of charcoal production, corresponding to the first component of RQ1. To examine whether charcoal-producing and non-charcoal-producing tree species differ in their functional trait composition, a series of PCAs were performed using different subsets of traits. These analyses visualize the multivariate trait space occupied by both groups and identify the main axes of functional differentiation. The first PCA includes the five traits with the most complete data, while subsequent analyses test the robustness of this pattern by (i) excluding species with limited trait information, (ii) expanding the number of included traits, (iii) focusing on ecologically relevant traits, and (iv) selecting traits based on the Boruta machine learning algorithm.

The species included in each PCA, along with their group assignment (charcoal or non-charcoal), are listed in Appendix A. This provides an overview of the taxonomic composition underlying each analysis and ensures transparency in the selection of species subsets.

For each PCA, a PERMANOVA test was used to assess whether the centroids of charcoal and non-charcoal species differ significantly in multivariate space, and a dispersion test evaluated differences in within-group variability. Ellipses in the PCA plots represent 95% confidence intervals around group centroids, providing a visual measure of group separation and overlap. Together, these analyses assess both the strength and consistency of functional differentiation between species used and not used for charcoal production across multiple analytical perspectives.

6.1.1 PCA of top traits

Charcoal-producing and non-charcoal species exhibit a partial but meaningful separation in functional trait space. Using the five traits with the highest species coverage (N = 83; non-charcoal = 47, charcoal = 36), the PCA revealed a partial but significant separation between charcoal-producing and non-charcoal tree species. The first two principal components together explained 59.5% of total trait variation (PC1 = 37.6%, PC2 = 21.9%).

PC1 represents a trade-off between acquisitive and conservative resource-use strategies. On the negative side, it is driven by specific leaf area (SLA, –0.54), leaf phosphorus concentration

(Leaf_P_mass, –0.53), and leaf nitrogen concentration (Leaf_N_mass, –0.43), reflecting thin, nutrient-rich leaves. The positive side of PC1 is influenced by wood density (+0.45) and seed mass (+0.19), indicating denser, structurally robust species. PC2 primarily captures variation in reproductive investment relative to structural and foliar traits, with seed mass loading positively (+0.78) against wood density (–0.47) and leaf nitrogen (–0.37).

Group centroids were primarily shifted along PC1, indicating that the differentiation between charcoal-producing and non-charcoal species is mainly driven by contrasts in wood density and leaf economics rather than seed investment. A PERMANOVA confirmed that this separation was statistically significant while within-group dispersion did not differ significantly. The low R^2 value suggests that, although statistically significant, the effect size is modest, implying that charcoal use preferences correspond to only a small portion of the total functional trait variation among species.

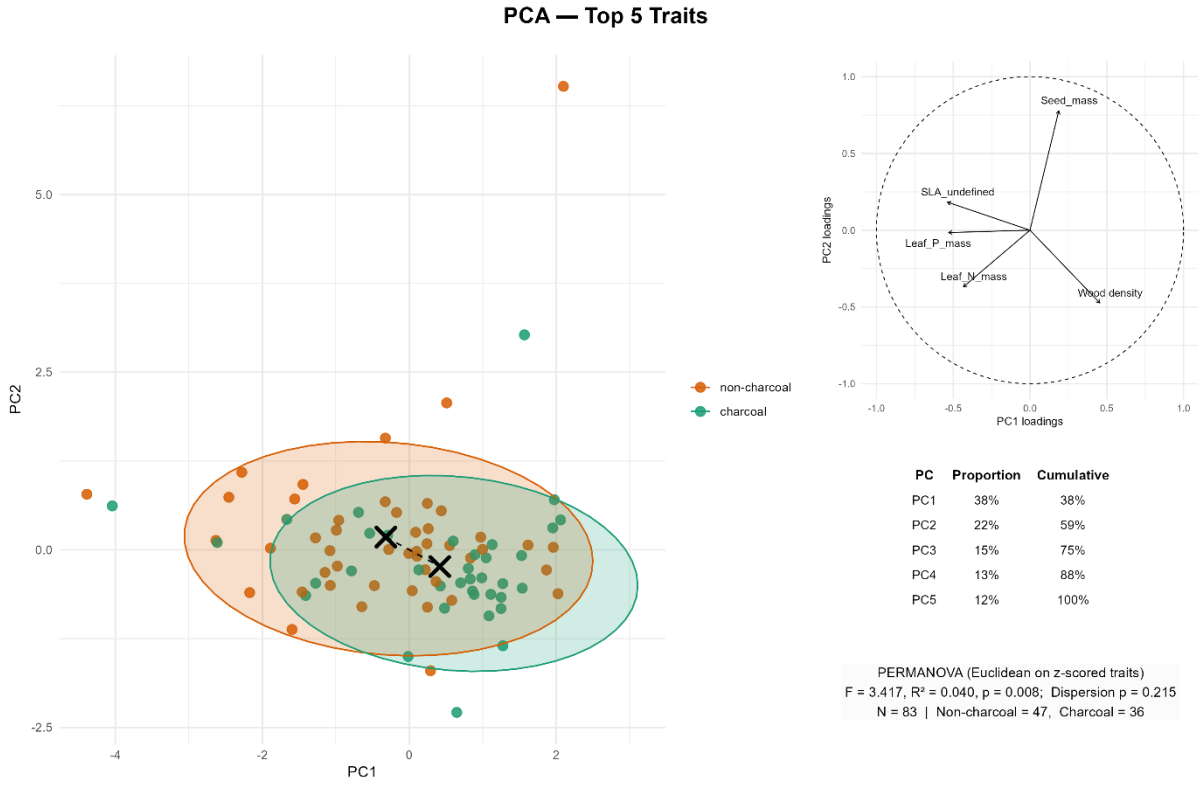

Figure 9 PCA of the five most data-complete traits

This figure compares charcoal-producing (orange) and non-charcoal (green) Miombo tree species. Group centroids are primarily separated along PC1. The difference is statistically significant, indicating a small but meaningful functional shift between groups. Ellipses show 95% CIs.

This pattern is illustrated in Figure 9, which shows PCA ellipses representing 95% confidence intervals around group centroids. Charcoal species tend to cluster toward denser wood and lower leaf nutrient concentrations, reflecting slower growth and higher structural investment. Overall, these findings

suggest that human selection for charcoal aligns with intrinsic ecological strategies, favoring conservative trait syndromes.

To assess the robustness of the trait-based differentiation, the PCA was repeated after excluding species represented by fewer than three observations per trait, reducing the dataset to N = 32 species (non-charcoal = 23, charcoal = 9). This filtering step ensured that each trait value reflected species with a more reliable sample size.

The first two components explained 66.5% of total variation (PC1 = 42.0%, PC2 = 24.5%). The trait loadings remained consistent with the full dataset: PC1 contrasted leaf phosphorus (–0.55), SLA (–0.52), and leaf nitrogen (–0.47) against wood density (+0.35) and seed mass (+0.29), again describing an acquisitive–conservative resource-use axis. PC2 separated seed mass (+0.59) from wood density (–0.66) and leaf nitrogen (–0.42), reflecting a secondary gradient between reproductive investment and structural robustness.

In contrast to the full dataset, the centroids showed greater overlap, and the PERMANOVA was not significant, with no difference in dispersion. This suggests that, once species with limited data were excluded, the previously observed separation was not statistically supported, likely reflecting reduced statistical power and the influence of sample completeness on trait-space differentiation.

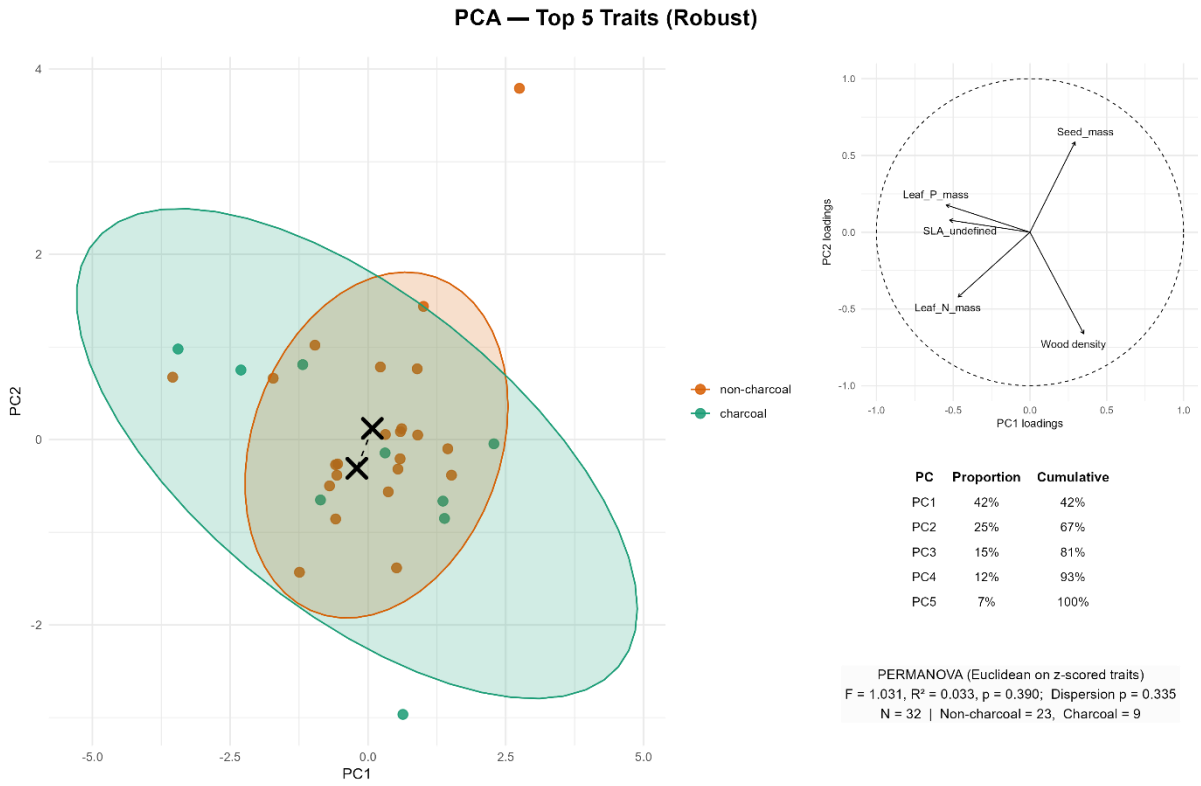

Figure 10 Robustness test of the PCA

Top 5 traits robust uses only species with ≥3 observations per trait. Groups overlap strongly. The difference is not significant. Ellipses show 95% CIs.

To further explore trait differentiation while maximizing the number of traits included, a third PCA was conducted using nine functional trait. The number of nine traits corresponded to a natural inflection point in the trait–species coverage curve Figure 5, ensuring sufficient species representation while expanding the functional scope of the analysis.

The first two principal components together explained 60.6% of total variation (PC1 = 38.9%, PC2 = 21.7%). PC1 was primarily structured by leaf and wood traits, contrasting acquisitive foliar traits (SLA variants, leaf phosphorus, and leaf nitrogen) with conservative structural traits (wood density and, to a lesser degree, seed mass). PC2 represented a secondary axis dominated by leaf nitrogen and structural properties, distinguishing species with higher leaf nitrogen and lower wood density from those with denser stems and lower foliar nutrient concentrations.

Group centroids for charcoal-producing and non-charcoal species overlapped substantially along both axes, with only a weak directional shift toward higher wood density and lower SLA among charcoal species. A PERMANOVA confirmed that the difference in multivariate trait composition was not statistically significant, and dispersion between groups did not differ significantly.

This lack of statistical separation suggests that, despite similar underlying trait correlations to the previous analyses, the inclusion of additional traits increased overall trait-space variance while reducing group distinctness. The result likely reflects the smaller number of species available for this expanded trait set (N = 17; non-charcoal = 12, charcoal = 5), which limited statistical power to detect subtle ecological differences. Nonetheless, the direction of centroid shifts remained consistent with earlier findings, indicating that species used for charcoal production tend to occupy a slightly more conservative region of the functional trait space, characterized by higher wood density and lower leaf nutrient investment.

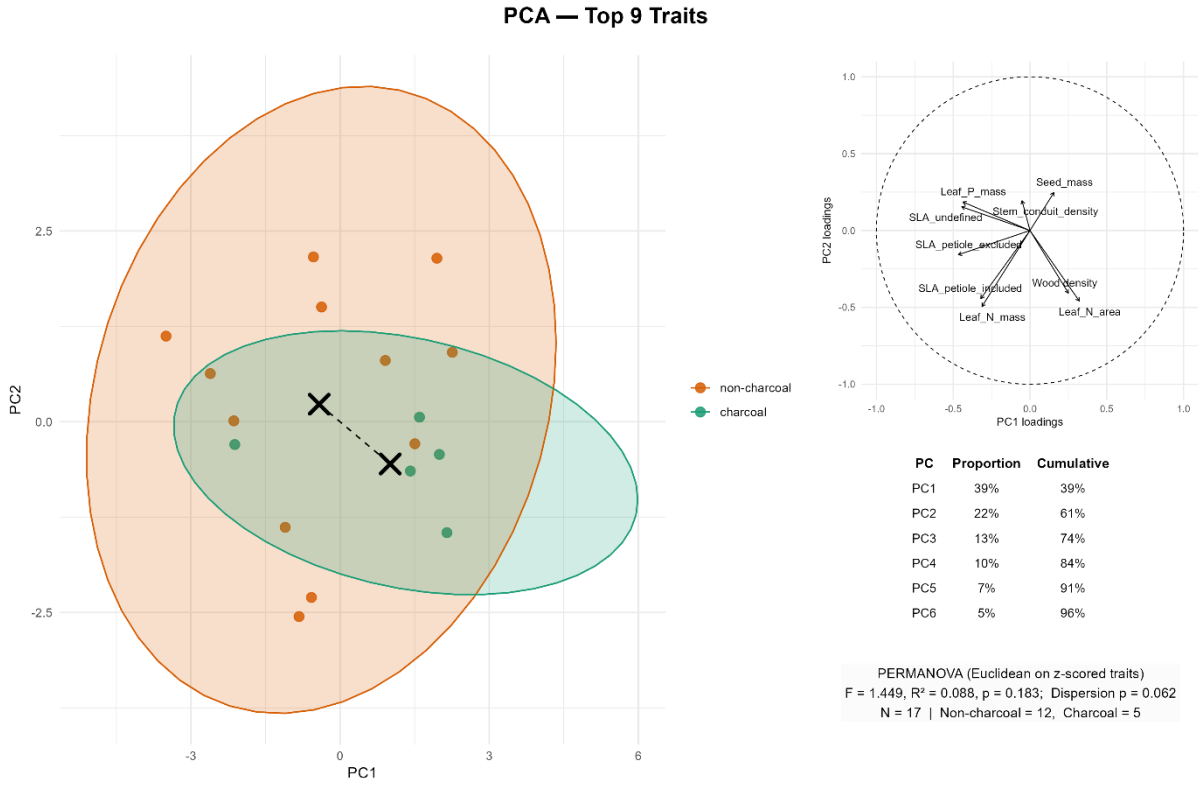

Figure 11 PCA of nine functional traits

The first two axes explain 60.6% of total variation. Centroids show minor separation along PC1, but the effect is not significant. Ellipses show 95% CIs.

6.1.2 PCA with traits selected for ecological range

As a complementary test, I repeated the PCA using a priori ecologically relevant traits regardless of data coverage. The selected traits are wood density, SLA, leaf N (Leaf_N_mass), diameter at breast height (DBH), and stem conduit density, (N = 34; non-charcoal = 22, charcoal = 12). The first two axes explained 65.7% of trait variation (PC1 = 43.6%, PC2 = 22.1%).

PC1 contrasted wood density (+0.56) and stem conduit density (+0.34) with SLA (–0.52), Leaf_N_mass (–0.39), and DBH (–0.38), capturing a conservative (dense wood, hydraulic investment) vs.

acquisitive/large-size axis. PC2 opposed Leaf_N_mass (+0.70) to DBH (–0.69), reflecting a gradient from leaf nutrient investment to stem size.

Group centroids overlapped strongly and showed no clear directional shift along either axis. PERMANOVA indicated no significant difference between groups, and dispersions were similar. Thus, focusing on these core ecological traits does not support a detectable functional separation between charcoal and non-charcoal species.

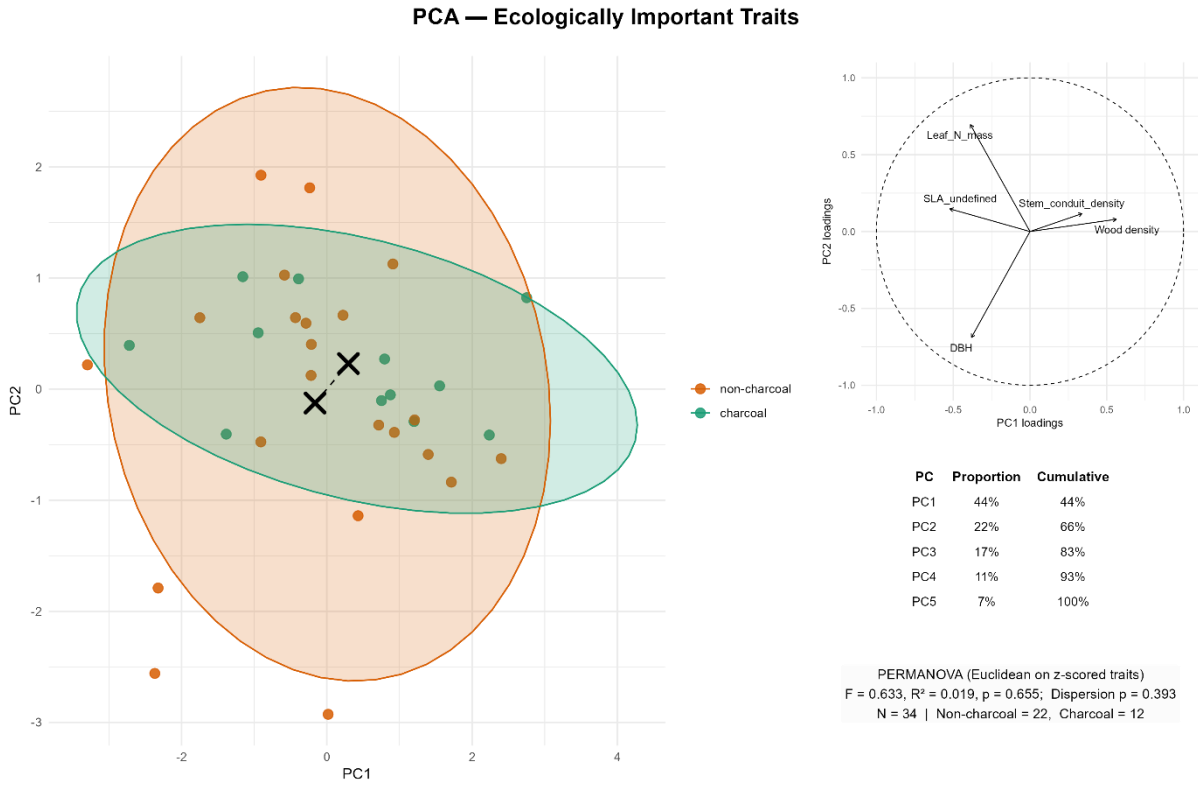

Figure 12 PCA of ecologically selected traits.

6.1.3 PCA with traits selected with the Boruta algorithm

A subsequent PCA using these five Boruta-selected traits (N = 86; non-charcoal = 48, charcoal = 38) explained 61.1% of total variation (PC1 = 35.4%, PC2 = 25.6%). PC1 contrasted SLA (+0.64) and Leaf_N_mass (+0.18) against wood density (–0.55) and Leaf_N_area (–0.51), describing a trade-off between acquisitive and conservative strategies. PC2 was driven mainly by Leaf_N_mass (+0.77) and Leaf_N_area (+0.49), representing a gradient of nutrient investment.

Group centroids showed a modest shift, primarily along PC1, with charcoal species tending toward denser wood and lower SLA and nitrogen content. The PERMANOVA indicated a marginally non-significant trend, while dispersion differences were also non-significant. This suggests a weak but

ecologically meaningful tendency for charcoal species to cluster toward conservative trait combinations.

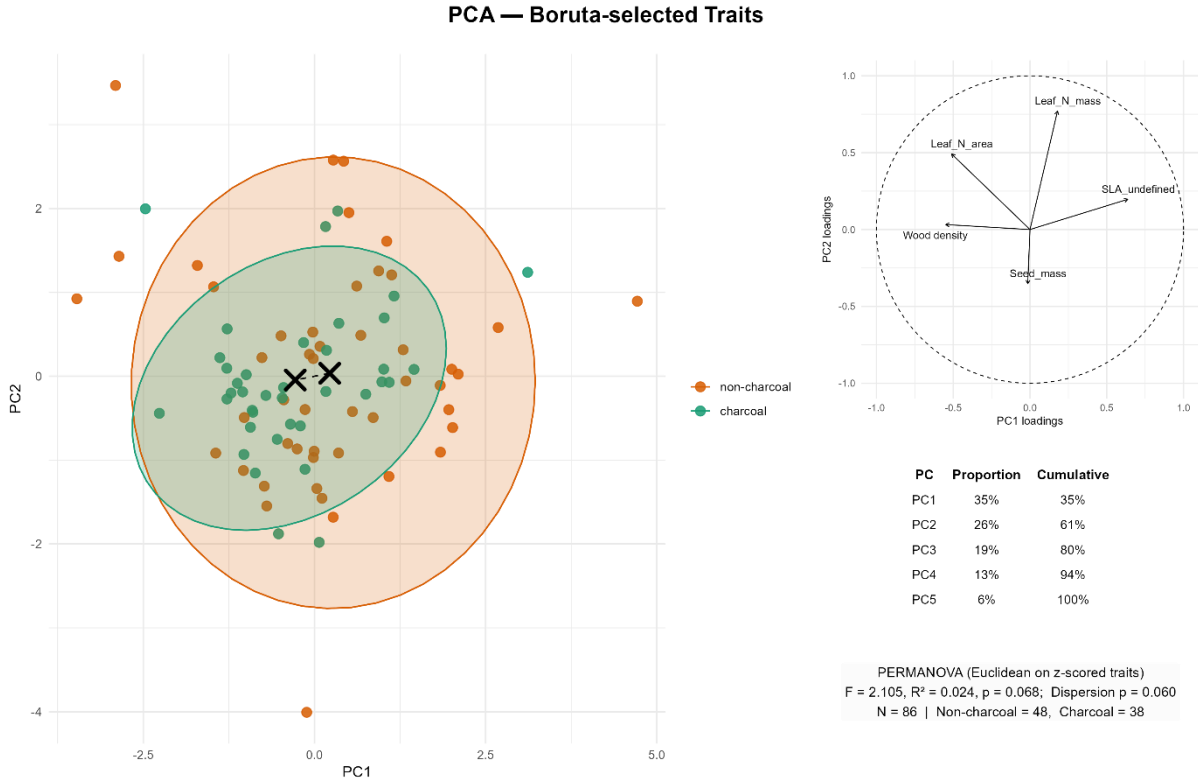

Figure 13 PCA of Boruta-selected traits

The first two axes explain 61.1% of variation. Ellipses represent 95% confidence intervals. Group separation is marginally non-significant.

Across all PCA approaches, the separation between charcoal-producing and non-charcoal species was generally weak but directionally consistent. Charcoal species tended to cluster toward traits associated with conservative resource-use strategies—higher wood density, lower specific leaf area, and reduced leaf nutrient concentrations—whereas non-charcoal species occupied the more acquisitive region of trait space. However, this distinction was statistically significant only in the most data-complete model and diminished when trait coverage was restricted or the trait set expanded. The uniformly low R^2 values across PERMANOVAs indicate that charcoal use explains only a small fraction of the total functional variation among Miombo tree species.

Collectively, these analyses suggest that while charcoal species share a general tendency toward conservative functional traits, the boundaries between groups remain diffuse, emphasizing the considerable functional diversity and ecological overlap within Miombo woodlands.

6.2 National-scale spectral and diversity patterns in the Miombo woodlands

This section aims the analysis to the national scale of Tanzania, addressing the second component of RQ1 by examining how vegetation structure, spectral composition, and diversity differ between protected and unprotected Miombo woodlands. Using Sentinel-2 imagery, vegetation indices (NDVI, NDMI, and CCI) are used to describe canopy condition and physiological status. PCA and diversity metrics derived from clustered “spectral species” further quantify how protection status influences ecosystem heterogeneity and resilience.

6.2.1 Vegetation condition: Protected vs. unprotected

On average, all three indices NDVI, NDMI, and CCI were higher in Protected Miombo woodlands compared to Unprotected areas. Across all 20 independent sampling runs, the vegetation index means remained highly stable, differing only in the third decimal place see Table 5 . This consistency indicates that the sampling process was robust.

Table 5 Min/Max and difference values for vegetation metrics over 20 runs.

	Protected (P)		**Unprotected (U)**		**P-U**
Metric	Min Value	Max Value	Min Value	Max Value	Mean ± SD
NDVI	0.46098	0.46412	0.38419	0.38707	0.0772 ± 0.0010
NDMI	0.15614	0.15889	0.06238	0.06480	0.0941 ± 0.0010
CCI	0.44063	0.44462	0.33756	0.34066	0.1035 ± 0.0011

The shape of the frequency distribution of values of NDVI, NDMI and CCI varies between the two governance types which are displayed in Figure 15 as a violin plot where each line represents one run. The lines are very much overlapping, highlighting the stability of the sampled data. Protected areas exhibited more condensed, high-centered distributions, particularly for NDVI, indicating greater uniformity and higher overall vegetation health. In contrast, Unprotected areas displayed broader and flatter distributions, suggesting greater heterogeneity in vegetation conditions. The NDMI distributions showed a similar trend, with Protected sites forming tighter, higher-valued peaks, whereas Unprotected ones were wider and more diffuse; thus, protected areas have a more restricted set of moisture values in comparison to unprotected areas. For CCI, Protected areas exhibited slightly higher mean values than Unprotected areas, with an average difference of 0.1035 ± 0.0011. This indicates that vegetation in Protected Miombo woodlands generally maintains a higher chlorophyll content and more uniform canopy condition, consistent with reduced disturbance and healthier photosynthetic activity compared to Unprotected areas.

Together, these patterns show that Protected Miombo woodlands maintain consistently higher and more homogeneous greenness, humidity and chlorophyll values than Unprotected areas, reinforcing the expected influence of protection status on vegetation conditions and spectral characteristics.

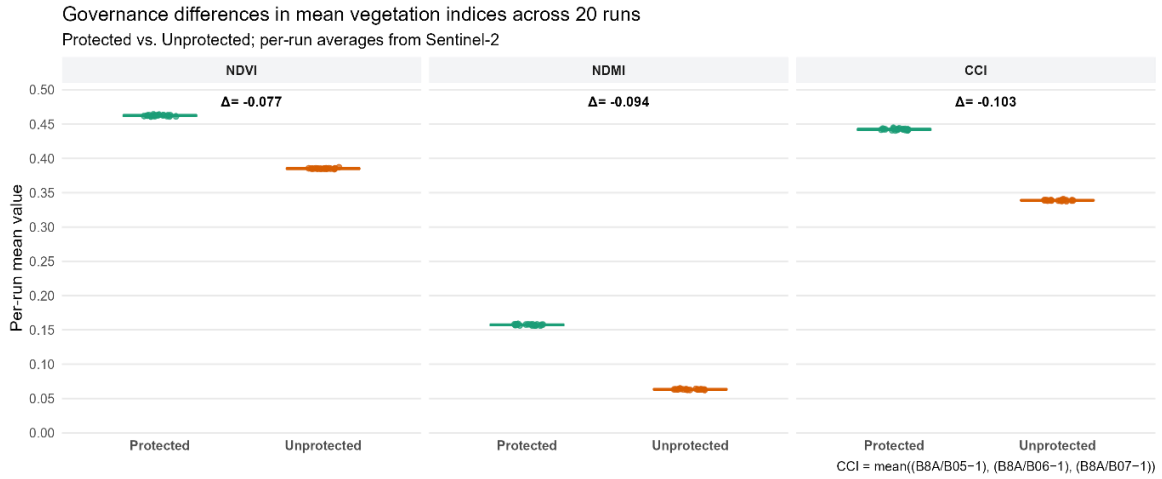

Figure 14 NDVI, NDMI, and CCI for Protected and Unprotected areas.

Boxes summarize the distribution of run-level means; points show individual runs. The annotated Δ values report the average difference across runs (Unprotected − Protected).

Figure 15 Distribution outlines of 20 runs of NDVI, NDMI, and CCI.

Horizontal width reflects the relative frequency of values (kernel density), vertically aligned to the index scale; densities are normalized to peak = 1 within each index × group. Full tails are shown so extreme values remain visible.

6.2.2 Spectral composition: Protected and unprotected

The spectral composition of Miombo woodlands varies with governance, reflecting structural and moisture-related differences between Protected and Unprotected areas. The national 10-band PCA for Run 12 explained 89.3% of total variance within the first two components (PC1 = 61.3%, PC2 = 28.0%).

The scatterplot shows a clear overlap yet with a modest but systematic separation between Protected and Unprotected Miombo woodlands, based on 47,707 observations (11,309 Protected; 36,398 Unprotected). The centroids of both groups are displaced primarily along PC1 (Euclidean distance = 1.53), while variation along PC2 remains comparatively smaller.

Spectral differences between governance types are driven mainly by variation in the red-edge and shortwave infrared regions. According to the loadings, PC1 represents a broadband brightness and vegetation intensity gradient, with strong positive contributions from the red-edge (B05–B07), red (B04), and SWIR (B11–B12) bands. This axis can therefore be associated with structural openness, reduced moisture, or exposed soil. PC2 is dominated by near-infrared (B8A) and red-edge (B06–B07) bands with opposite signs to the visible range, capturing subtle variation in canopy density and internal leaf scattering. This pattern is illustrated in Figure 16, showing the cluster of Protected pixels toward lower PC1 values and a more compact distribution.

In the PCA space, Protected pixels cluster more compactly and are shifted toward lower PC1 values, potentially indicating generally darker and more moisture-rich spectral signatures, while Unprotected pixels extend toward higher PC1 values, indicating greater heterogeneity and brighter surfaces. The 95 % confidence ellipses show that the Protected group occupies a smaller area, consistent with more uniform vegetation cover, whereas the Unprotected group is elongated along PC1 and PC2, suggesting broader variability in canopy condition.

Separability metrics confirm this moderate but systematic offset between governance types at the national level. Overall, the results indicate that differences between Protected and Unprotected Miombo woodlands are primarily driven by red-edge and SWIR reflectance variability, potentially due to Unprotected areas having more open canopies and lower vegetation moisture.

Reducing the analysis to vegetation-sensitive wavelengths maintains the spectral distinction between protected and unprotected Miombo woodlands. The 5-band PCA, based on red, red-edge, and near-infrared wavelengths, explained 99.2% of total variance within the first two components (PC1 = 63.4%, PC2 = 35.8%). The scatterplot shows a pattern similar to the 10-band model, with substantial overlap between groups but a consistent shift in centroid positions along PC1, while variation along PC2 remains comparatively smaller. This refined spectral differentiation is visualized in Figure 17.

The spectral differences are primarily shaped by gradients in vegetation density, canopy moisture, and chlorophyll content. According to the loadings, PC1 represents a broad vegetation density and moisture gradient, dominated by negative contributions from the near-infrared and red-edge regions, which are sensitive to canopy cover and internal scattering. PC2 is driven by stronger responses in the red and lower red-edge regions, capturing subtle variation in leaf pigment absorption and chlorophyll

concentration. Compared to the 10-band PCA, this configuration effectively rotates the spectral space, emphasizing vegetation-specific reflectance while minimizing the influence of soil brightness and background variation.

Separability metrics indicate that although the centroid distance decreased, the Mahalanobis and Jeffries–Matusita distances increased relative to the 10-band model, suggesting slightly improved class discrimination when focusing on vegetation-sensitive wavelengths. Overall, the 5-band PCA preserves the governance contrast observed in the full-band analysis while refining it along axes that more directly capture canopy condition and photosynthetic activity.

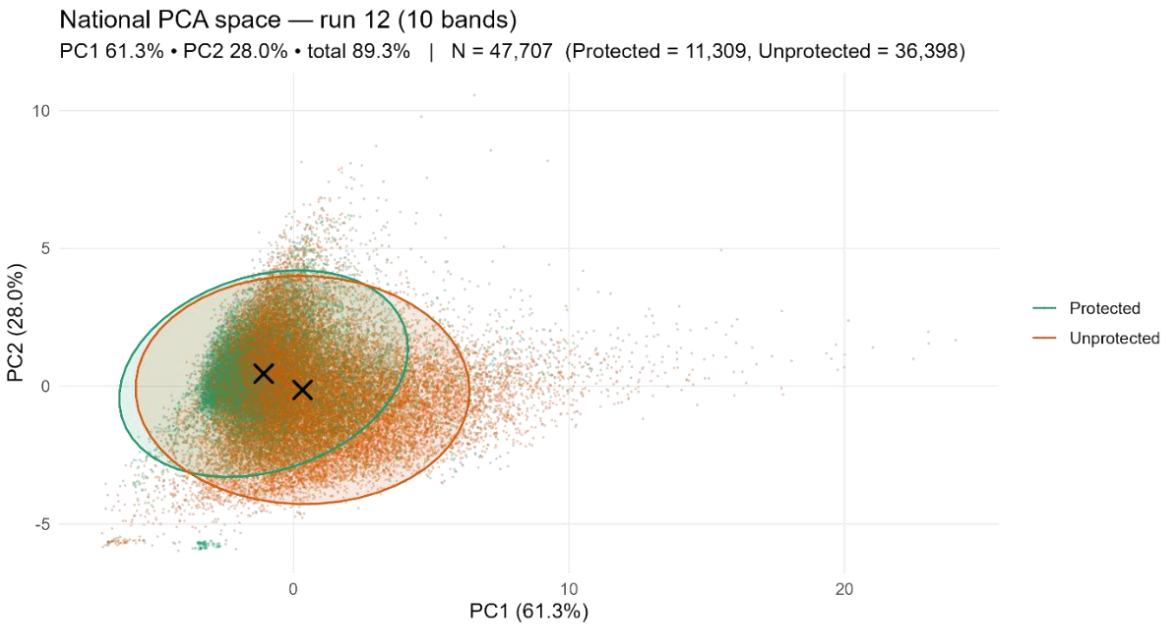

Figure 16 Scatterplot of national Miombo woodland pixels

Each point represents a sampled 20 m pixel, with 95 % confidence ellipses and group centroids (black crosses) shown. Bands used are B01, B02, B03, B04 B05, B06, B07, B8A, B09, B11, B12.

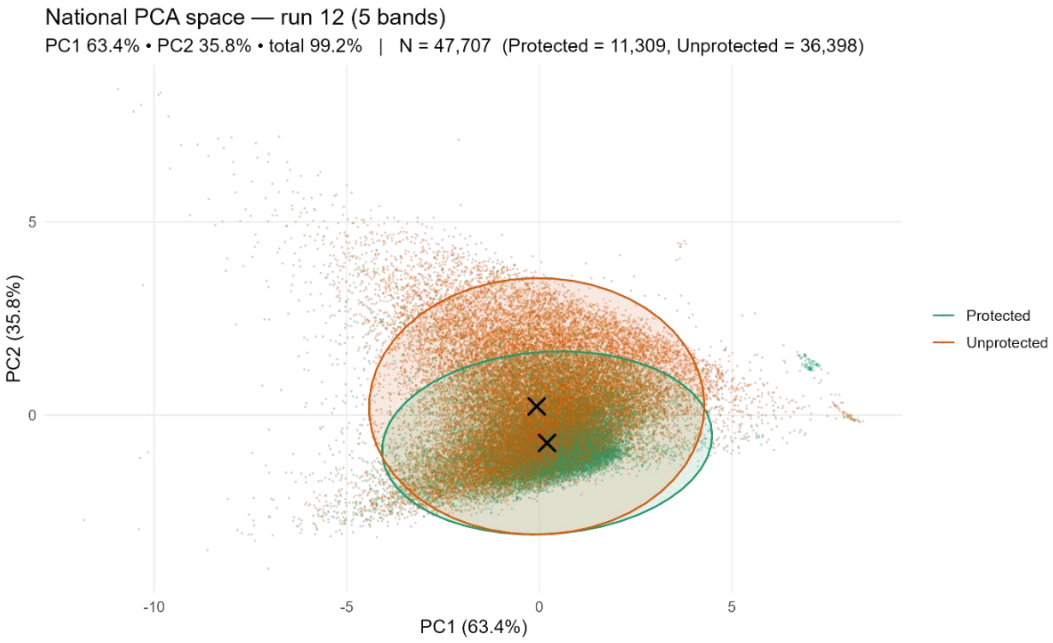

Figure 17 Scatterplot of Protected national Miombo woodland pixels

Each point represents a sampled 20 m pixel, with 95 % confidence ellipses and group centroids (black crosses) shown. Bands used are B05, B06, B07, B8A, B04.

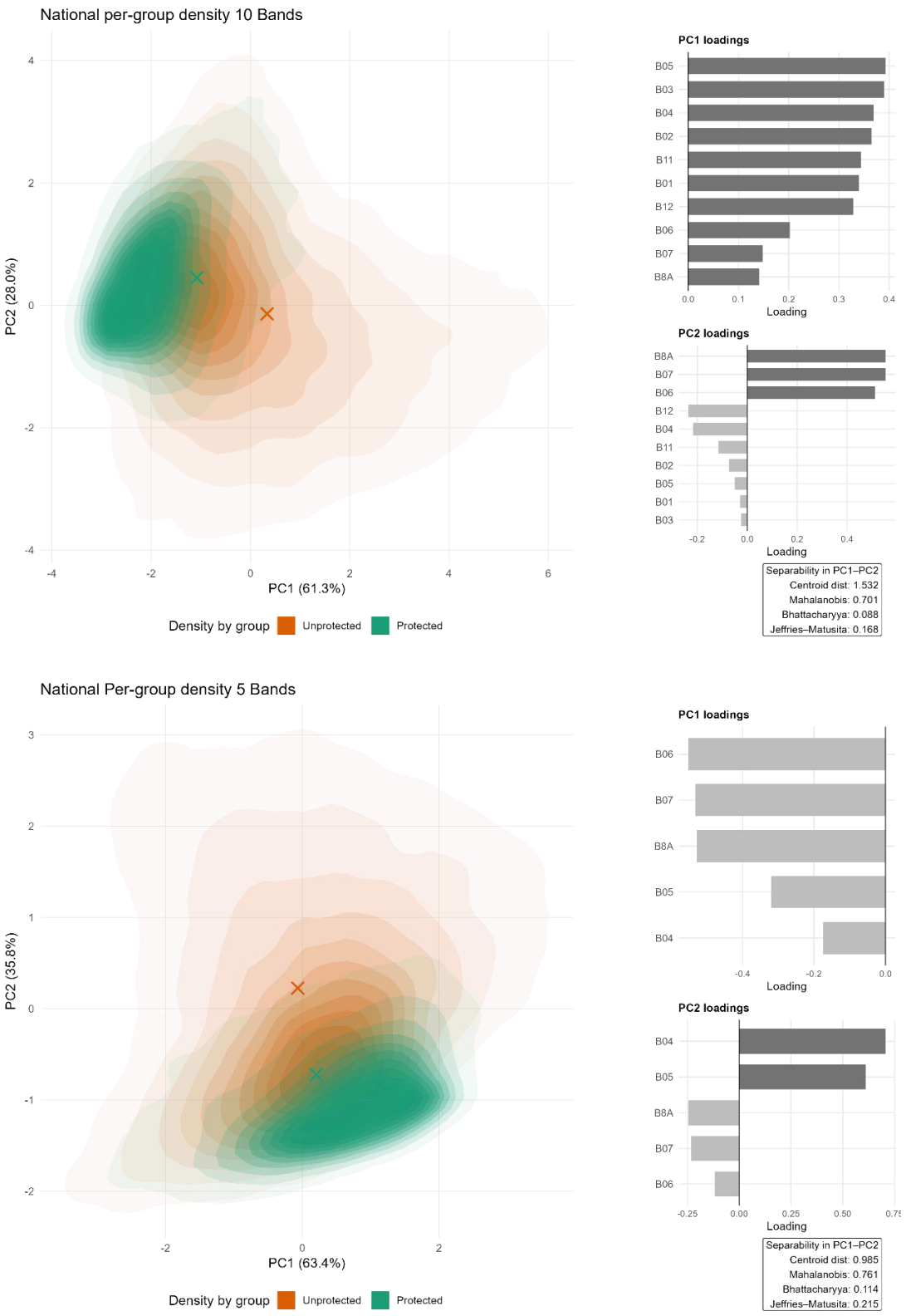

Figure 18 National density contours for ten and five band

Color intensity indicates point density, with darker areas representing higher concentrations of pixels. Black crosses mark group centroids. Bar plots on the right show the spectral band loadings that define each principal component, illustrating which Sentinel-2 bands contribute most strongly to PC1 and PC2. Positive and negative bars indicate the direction and relative influence of each band on the PCA axes

6.2.3 Diversity metrics: Protected vs. unprotected

Spectral diversity is higher in unprotected areas at permissive thresholds, reflecting fine-grained heterogeneity, while protected areas retain a core set of dominant spectral types. The delta-heatmap visualization reveals clear contrasts in the spectral composition of clusters between governance groups. Clusters most prevalent in Protected areas consistently exhibit above-average standardized values across all three vegetation indices (NDVI, NDMI, CCI), indicating higher canopy greenness, chlorophyll content, and moisture conditions. In contrast, clusters dominating Unprotected areas generally show below-average index values, reflecting comparatively lower vegetation vigor and moisture availability. This distribution is illustrated in Figure 19.

A moderate left-to-right trend is also noticeable. On the left side of the heatmap—where clusters show stronger compositional differences between governance groups—broad green and white bands dominate, indicating more consistent spectral characteristics within clusters. Toward the right, where cluster representation is more balanced between groups, the pattern becomes increasingly mosaic-like, with narrow alternating green and blue stripes that reflect smaller, more variable spectral differences.

To understand the proportion difference better here an example, cluster 5 exhibits a 6.2 % difference in representation between governance types: it contains 6.9 % of all Protected pixels which is 785 pixels but only 0.7 % of Unprotected which corresponds to 81 pixels, indicating that this spectral type is strongly associated with Protected areas.

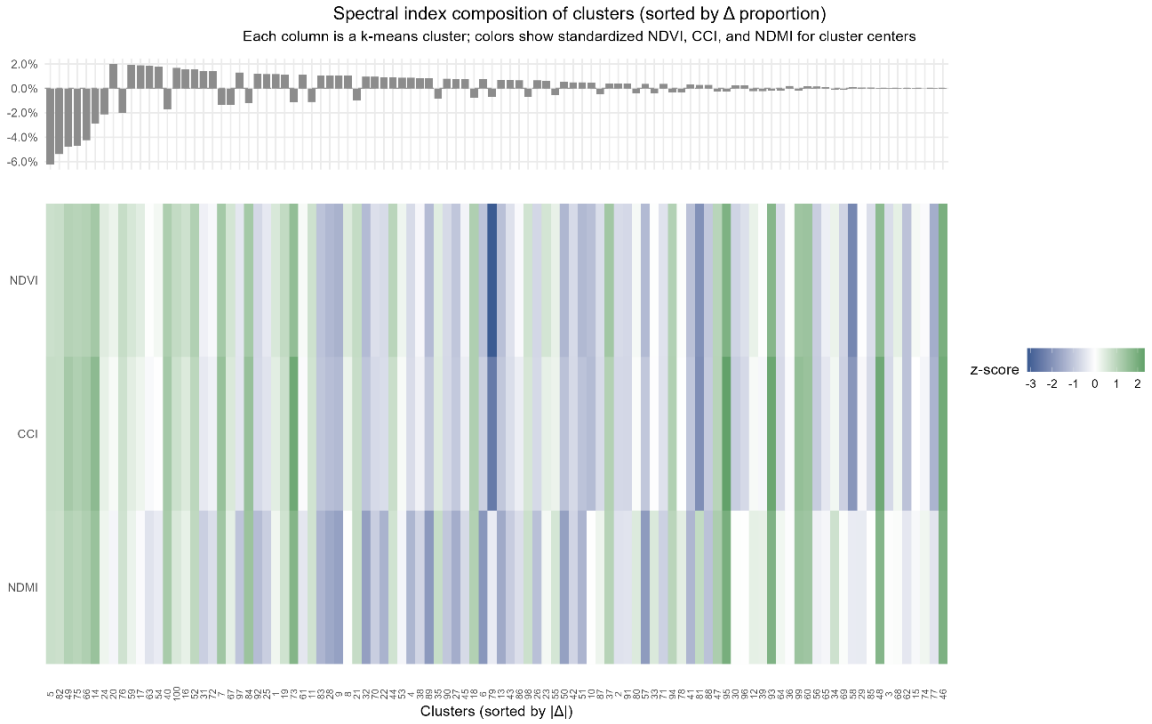

Figure 19 Cluster-based comparison of spectral indices

Each column represents one *k-means* cluster derived from a 10-band Sentinel-2 spectral feature space. Clusters were sorted by the absolute difference (|Δ|) in proportional representation between unprotected and protected pixels. The upper panel shows Δ (unprotected – protected) for each cluster, indicating whether that spectral composition is more common in either group. The lower heatmap displays standardized spectral indices — NDVI (vegetation greenness), CCI (canopy chlorophyll index), and NDMI (moisture index) calculated from the mean reflectance values of each cluster's. Values are expressed as *z-scores* (standard deviations from the mean of each metric) to ensure comparability across indices that naturally have different numeric ranges and sensitivities. Blue cells indicate values below the overall mean, green cells indicate above-average values, and white cells indicate near-mean conditions.

The richness divergence sensitivity analysis (Figure 20) illustrates how the difference in spectral richness (ΔS) between unprotected and protected areas responds to variation in clustering size (K) and global presence threshold (τ). Across most parameter combinations, ΔS is positive, particularly at lower τ values (≤ 1%), indicating that unprotected sites generally exhibit higher spectral richness. This pattern suggests that fine-grained spectral variability, captured at low presence thresholds, is more pronounced in unprotected or more heterogeneous landscapes. As τ increases, ΔS approaches zero or becomes slightly negative, implying that the dominance of a few major spectral clusters becomes comparable or even higher in protected areas when rarer spectral signals are excluded. These patterns highlight that unprotected areas are structurally heterogeneous at fine scales, while protection stabilizes the most widespread canopy types.

The influence of K on ΔS is relatively modest: richness differences remain consistent across clustering scales between K = 80 and 150, suggesting that the governance-related signal is robust to moderate changes in clustering granularity.

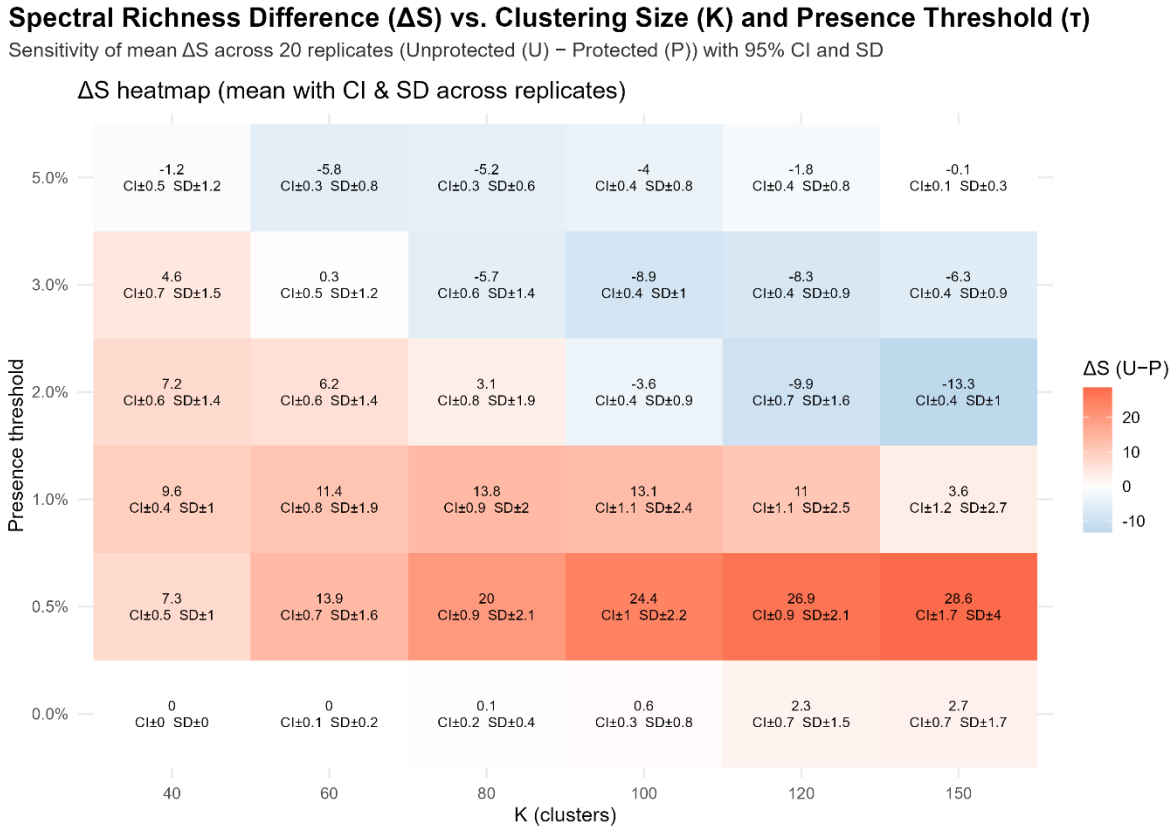

Figure 20 Sensitivity of spectral richness difference vs clustering size and threshold.

Each tile represents the mean across 20 independent replicates, with warmer tones (red) indicating higher spectral richness in unprotected areas and cooler tones (blue) indicating higher richness in protected areas. ΔS values incorporate 95% confidence intervals and standard deviations across replicates.

Figure 21 visualizes the spatial distribution of spectral richness differences (ΔS = Unprotected − Protected) within the principal component (PC) space derived from Sentinel-2 data. Each map represents a different global presence threshold (τ), which defines the minimum proportion of the entire dataset in which a spectral cluster must occur to be counted as present. Lower thresholds (e.g., τ = 0.5%) capture even rare or spatially limited spectral clusters, whereas higher thresholds (e.g., τ = 5%) include only those that are common across most pixels.

At τ = 0.5%, extensive red regions indicate that unprotected sites contain a larger number of present clusters, driven mainly by rare or heterogeneous spectral components. As τ increases, the red regions contract and blue regions become more prominent, meaning that when only frequent spectral clusters are considered, protected areas exhibit greater richness. According to the loadings, PC1 is strongly

influenced by the visible (blue to red), red-edge, and SWIR regions, which together capture gradients in canopy brightness, vegetation density, and moisture content. This could mean that the observed richness differences likely reflect contrasts in structural openness and vegetation moisture, with Protected areas characterized by darker, denser, and more homogeneous canopies, while Unprotected areas show greater spectral heterogeneity due to disturbance and canopy gaps.

Overall, the progression across thresholds shows that the contrast between unprotected and protected areas depends on the inclusion of rare versus dominant spectral types.

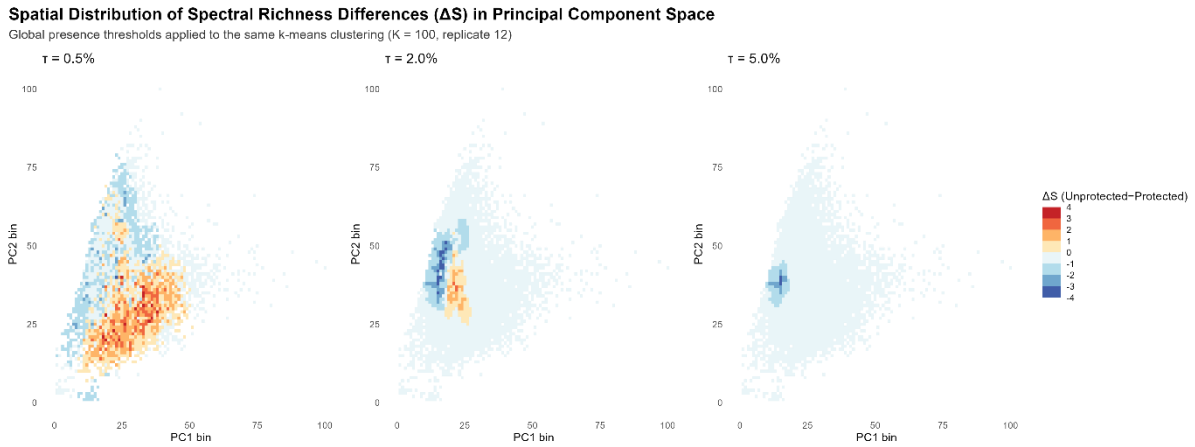

Figure 21 Spatial distribution of spectral richness difference in PC space under three thresholds.

Nationally, unprotected Miombo shows higher spectral richness and diversity at low prevalence, but protection retains the most widespread spectral types as thresholds tighten. At τ = 0.5%, Unprotected areas contain more spectral clusters than Protected and higher abundance-weighted diversity. Evenness is also slightly higher in Unprotected, indicating a broader and more even spread across spectral types.

At τ = 2.0%, richness tilts slightly toward Protected, and diversity advantages for Unprotected shrink. This indicates that as we require spectral types to be more prevalent, Protected areas keep pace and begin to dominate the persistent set.

At τ = 5.0%, Protected retains 1 high-prevalence cluster while Unprotected drops to 0, consistent with the idea that protection maintains a core, consistently represented spectral signature at national scale, whereas Unprotected areas host more rarer, patchy types that vanish under strict prevalence criteria.

Together with the PCA results, these thresholds suggest that Unprotected Miombo is spectrally heterogeneous, while Protected Miombo maintains fewer but more persistent spectral types at higher τ—consistent with higher canopy coherence and reduced disturbance in Protected areas.

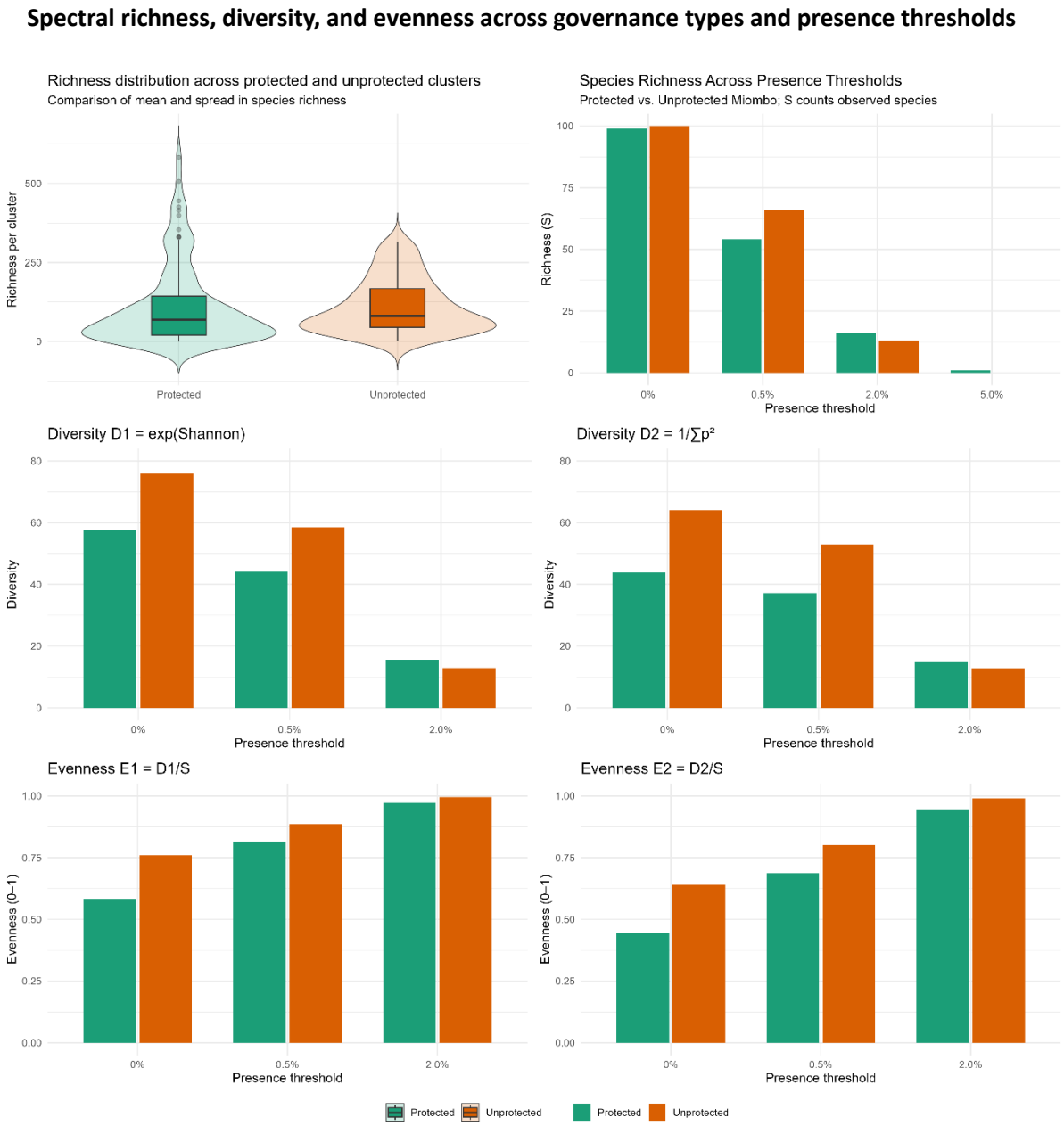

Figure 22 National spectral diversity metrics across thresholds

The upper-left panel shows the distribution of cluster-level richness (number of occupied spectral clusters) using violin and box plots, while the upper-right panel displays total spectral richness (S) across increasing presence thresholds. The middle panels illustrate Hill diversity indices representing the effective number of distinct spectral types. The lower panels show corresponding evenness measures, indicating how uniformly spectral types are distributed within each governance category.

6.3 Local scale vegetation and diversity patterns under different governances

At the local scale, this section addresses Research Question 2, examining if CBNRM and OA systems differ in their effects on forest structure, canopy condition, and diversity. The results combine

vegetation indices, spectral composition, and diversity metrics to capture differences in ecological integrity and resilience under the two governance regimes.

6.3.1 Vegetation condition: OA vs. CBNRM

Vegetation conditions differ between CBNRM and OA Miombo woodlands, with consistently higher canopy greenness, chlorophyll content, and moisture in OA areas. Across all three indices, NDVI, CCI, and NDMI, OA woodlands display higher mean values compared to CBNRM sites. NDVI increased by 0.051, CCI by 0.071, and NDMI by 0.067.

The distributions reveal that OA areas not only have higher central values but also broader variability, reflecting a mix of highly productive and more degraded vegetation patches. In contrast, CBNRM areas show more condensed distributions centered at lower index values, indicating greater homogeneity but overall lower canopy vigor and moisture. This pattern is visualized in Figure 23, showing the distribution of vegetation indices for OA and CBNRM governance in Miombo woodlands, highlighting the contrast in canopy condition. These results suggest that OA woodlands currently maintain higher vegetation activity, though with greater heterogeneity, while CBNRM areas promote structural uniformity.

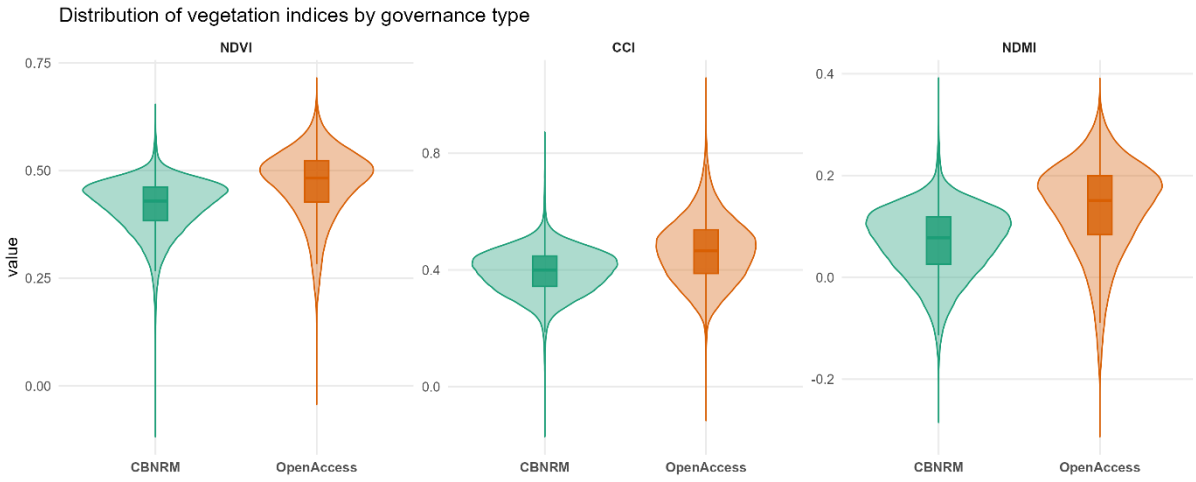

Figure 23 Vegetation indices for CBNRM and OA Governance in Miombo woodlands

Spatial patterns of vegetation indices (Figure 24) support the distributional differences shown in the violin plots. Across all three indices OA areas display generally higher and more heterogeneous values, whereas CBNRM sites appear more uniform and moderate. The maps highlight this contrast spatially: greener and moister patches are more widespread in OA woodlands, while CBNRM areas exhibit consistent mid-range values with fewer extremes. These patterns visually reinforce the statistical findings that canopy greenness, chlorophyll content, and moisture are higher but also more variable under open-access governance.

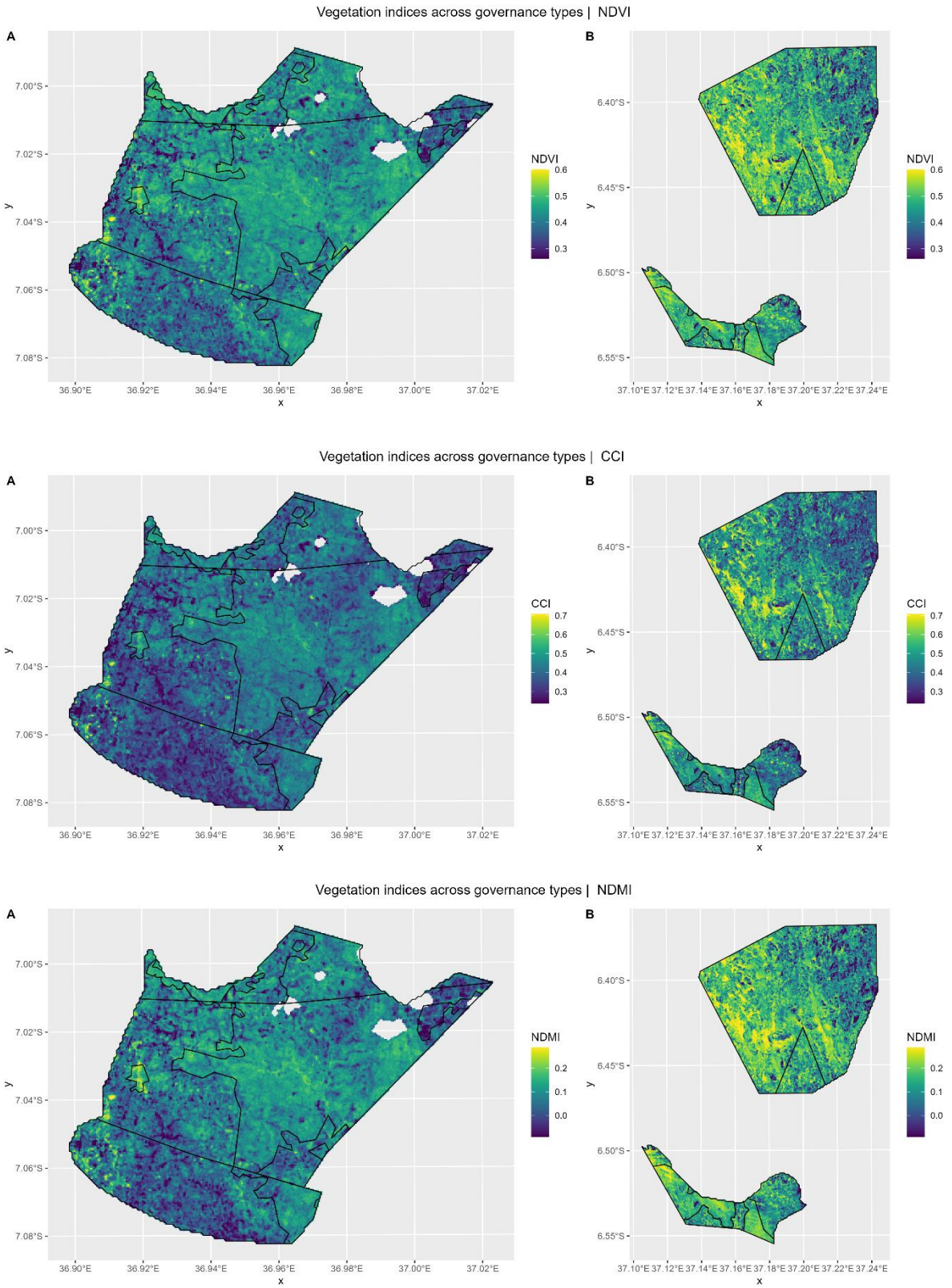

Figure 24 Spatial distribution of vegetation indices in Kilosa District.

A: CBNRM villages, B: OA villages.

6.3.2 Spectral composition: OA vs. CBNRM

CBNRM and OA woodlands differ in spectral composition at the local scale, with a modest but consistent separation primarily along PC2. The PCA for tile T37MBN shows that PC1 (62.0%) and PC2 (29.8%) together explain 91.9% of variance. The point cloud and 95% ellipses reveal substantial overlap but a clear centroid shift. The displacement occurs mainly along PC2. The loadings indicate that this PC1 shift reflects a brightness–moisture/structure gradient, while PC2 captures a red-edge/NIR canopy signal.

PC1 loads strongly and fairly uniformly on the visible (blue–red) and SWIR regions with only small weights on the NIR/red-edge trio. This axis is therefore associated with structural openness, exposed/bright backgrounds, and lower moisture versus darker, denser canopies. PC2 is dominated by red-edge/NIR responses contrasted with the visible range, capturing canopy density and internal leaf scattering. In the PCA space, OA pixels extend farther along higher PC1 and spread more broadly across PC1–PC2, indicating brighter, more heterogeneous canopies; CBNRM pixels are tighter and shifted toward lower PC1, consistent with more uniform, denser, and moister canopy conditions.

Even at village scale and different boundary conditions compared to the national sclae, governance is associated with spectral composition differences aligned with openness and moisture (PC1) and canopy/leaf optical properties (PC2), with OA more variable and shifted toward the open/bright end of the gradient, and CBNRM more compact toward the dense/dark end.

The separability metrics in the right panel of Figure 26 shows that the centroid distance indicates a moderate displacement between group means in PC space, while the Mahalanobis distance suggests partial overlap when accounting for within-group variance. The Bhattacharyya and Jeffries–Matusita distances confirm that the two spectral distributions are not fully separable, implying substantial similarity in canopy reflectance but with measurable differentiation likely linked to vegetation structure or condition.

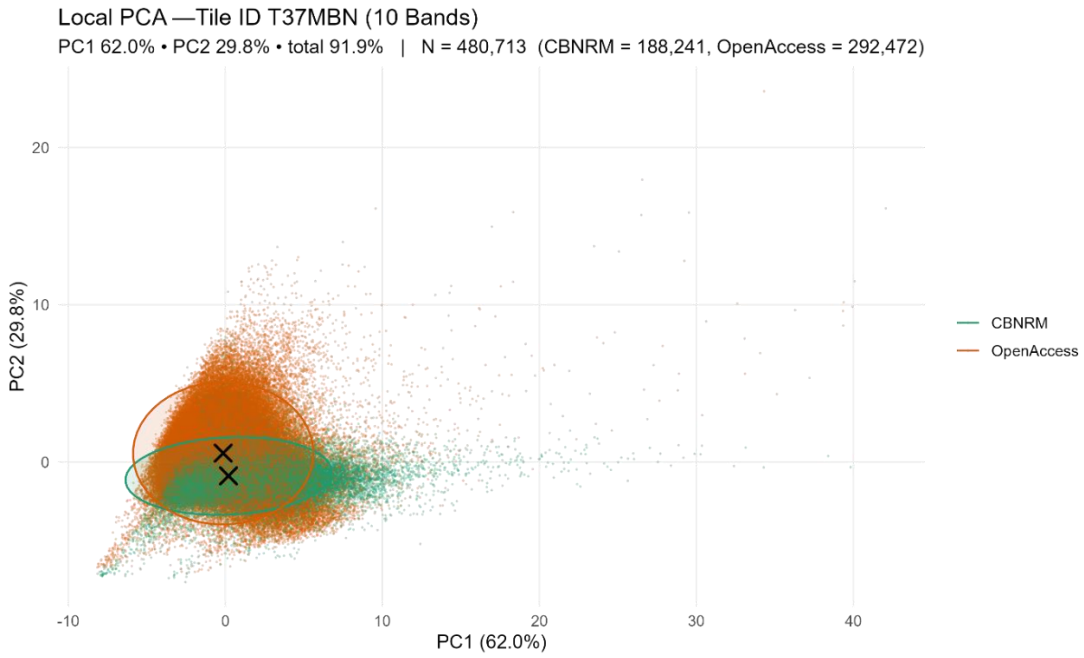

Figure 25 Scatterplot of local Miombo woodland pixels

Each point represents a 20 m pixel, with 95 % confidence ellipses and group centroids (black crosses) shown. Bands used are B01, B02, B03, B04 B05, B06, B07, B8A, B09, B11, B12.

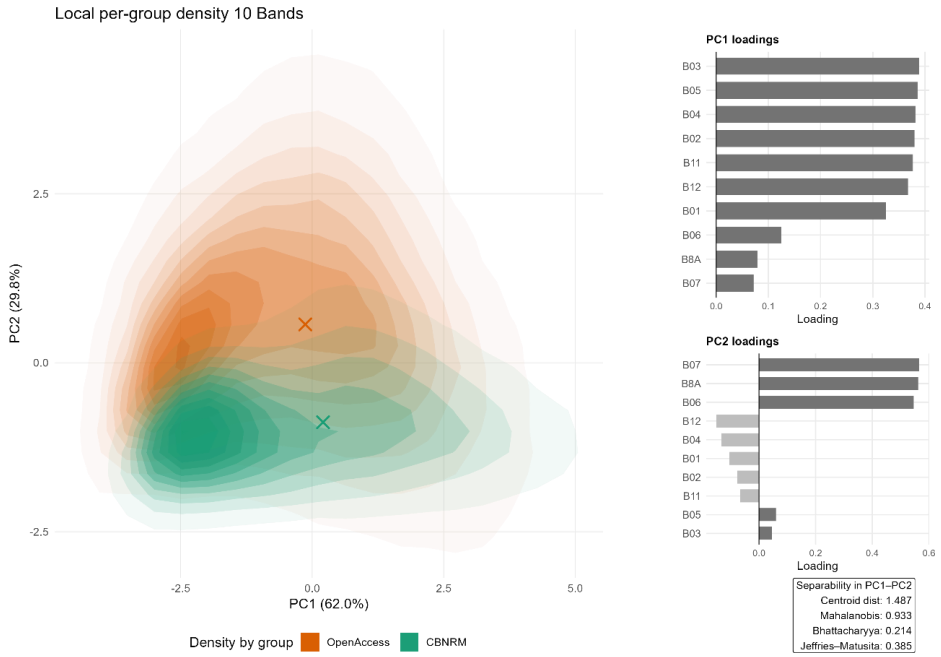

Figure 26 Local density contours of local Miombo woodland

Color intensity indicates point density, with darker areas representing higher concentrations of pixels. Crosses mark group centroids. Bar plots on the right show the spectral band loadings that define each principal component, illustrating which Sentinel-2 bands contribute most strongly to PC1 and PC2. Positive and negative bars indicate the direction and relative influence of each band on the PCA axes

6.3.3 Diversity metrics: OA vs. CBNRM

At the local scale, governance is associated with systematic differences in spectral composition: clusters with higher greenness, chlorophyll, and moisture are disproportionately represented by OA pixels, while lower-index clusters are more common in CBNRM. In the Δ–heatmap (Figure 27), columns are k-means clusters sorted by |Δ| where Δ = OpenAccess − CBNRM. On the left, where clusters have the largest absolute Δ values align with above-average standardized values (green tiles) regardless of governance types. On the right, clusters with lower absolute Δ align with below-average index values (blue tiles).

This pattern mirrors the national scale left to right structure but reverses the group ordering: whereas nationally the greener/moister clusters were more prevalent in Protected areas, locally the greener/moister clusters are more prevalent in OA than CBNRM. This indicates that, within the sampled villages, OA woodlands currently occupy a larger share of high-reflectance spectral types, while CBNRM woodlands are comparatively concentrated in lower-index spectral clusters.

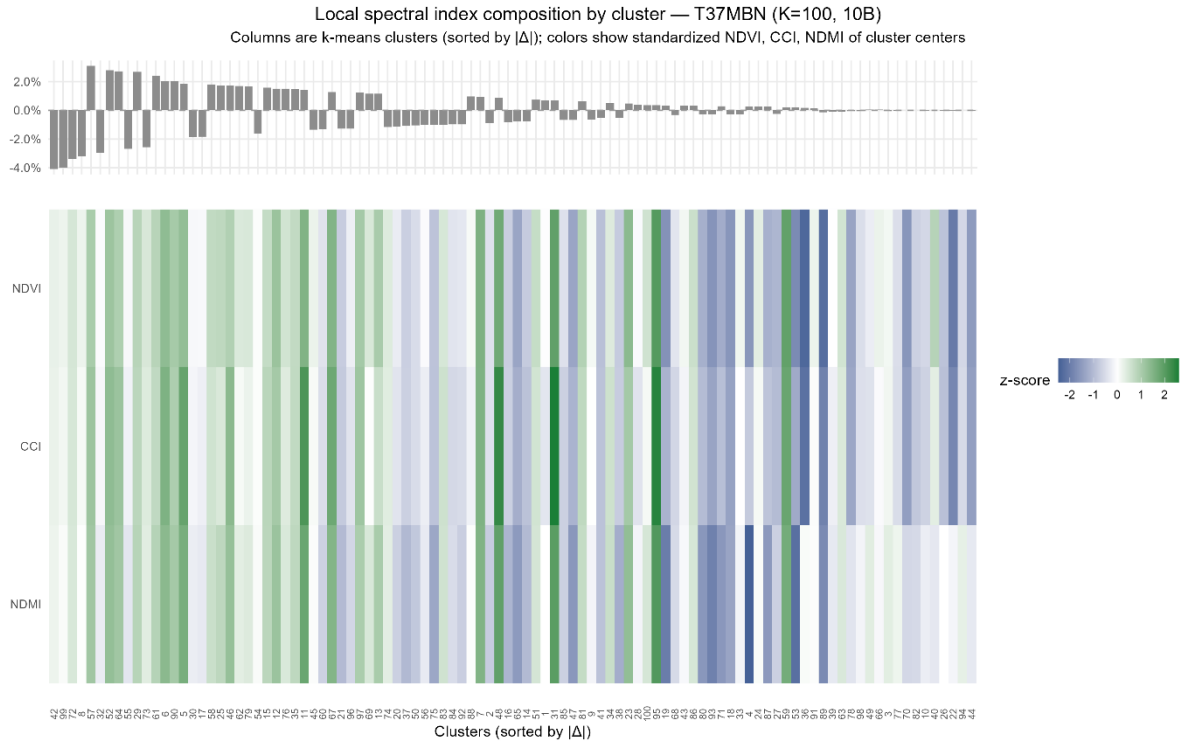

Figure 27 Cluster-based comparison of local spectral indices composition

Each column represents one *k-means* cluster derived from a 10-band Sentinel-2 spectral feature space. Clusters were sorted by the absolute difference (|Δ|) in proportional representation between OA and CBNRM. The upper panel shows Δ (OpenAccess − CBNRM) for each cluster, indicating whether that spectral composition is more common in either group. The lower heatmap displays standardized spectral indices calculated from the mean reflectance values of each cluster. Values are expressed as *z-scores* (standard deviations from the mean of each metric) to ensure comparability across indices that naturally have different numeric ranges and sensitivities. Blue cells indicate values below the overall mean, green cells indicate above-average values, and white cells indicate near-mean conditions.

OA woodlands show higher spectral “richness” at low thresholds, but this advantage reverses as the dominance threshold increases, indicating many rare spectral types in OA and fewer, more pervasive types in CBNRM. Across all cluster sizes, the count of clusters present (S) is larger in OA at low presence thresholds. As the threshold tightens (≥0.02), ΔS declines and often becomes negative, showing that fewer OA clusters exceed the stronger prevalence criterion. At the strictest level, OA frequently drops to S=0, while CBNRM still retains at least one dominant cluster in several cases.

At local scale, OA mosaics are richer in rare spectral types, consistent with greater spatial heterogeneity and mixed successional patches; CBNRM communities are characterized by fewer but more consistently represented spectral types, reflecting more uniform canopy conditions. In other words, OA’s richness is concentrated in the tails, whereas CBNRM maintains the core, dominant spectral composition.

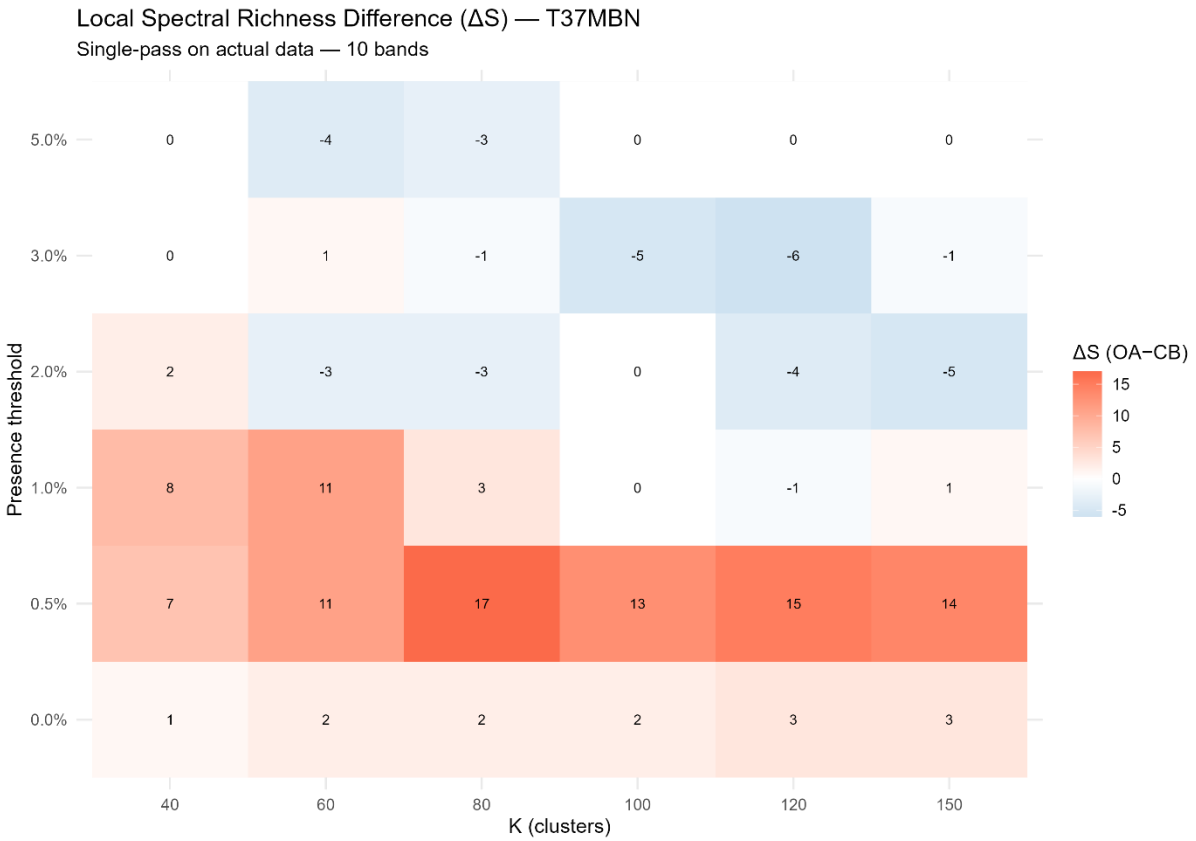

Figure 28 Sensitivity of local spectral richness difference vs clustering size and threshold

Each tile represents the mean ΔS across 20 independent replicates, with warmer tones (red) indicating higher spectral richness in unprotected areas and cooler tones (blue) indicating higher richness in protected areas.

Governance affects local spectral diversity: OA shows higher richness and diversity at low presence thresholds, but CBNRM retains more dominant spectral types as the threshold tightens. At τ = 0.5%, OA contains more spectral clusters than CBNRM, with substantially higher diversity for both common and dominant clusters. Evenness is slightly higher in OA, indicating a more even spread across its larger set of spectral types.

At intermediate prevalence (τ = 2.0%), richness equalizes while abundance-weighted diversity still favors OA. Richness is nearly the same, but OA retains a small advantage in diversity and higher evenness, suggesting OA maintains a more even distribution among the clusters that pass this threshold.

At high prevalence (τ = 5.0%), CBNRM dominates: OA richness and diversity collapse while CBNRM retains multiple pervasive spectral types. OA has only 2 qualifying clusters vs. 5 in CBNRM, with correspondingly much lower diversity. Evenness appears slightly higher in OA, but this reflects very few remaining clusters rather than a more balanced community.

These thresholds collectively indicate that OA woodlands harbor more rare spectral types that boost richness and diversity at permissive thresholds, whereas CBNRM woodlands maintain fewer but more consistently represented “core” spectral types that persist under strict prevalence criteria. This pattern aligns with earlier results: OA shows higher vegetation indices and broader spectral spread, while CBNRM is more compact and uniform in spectral space.

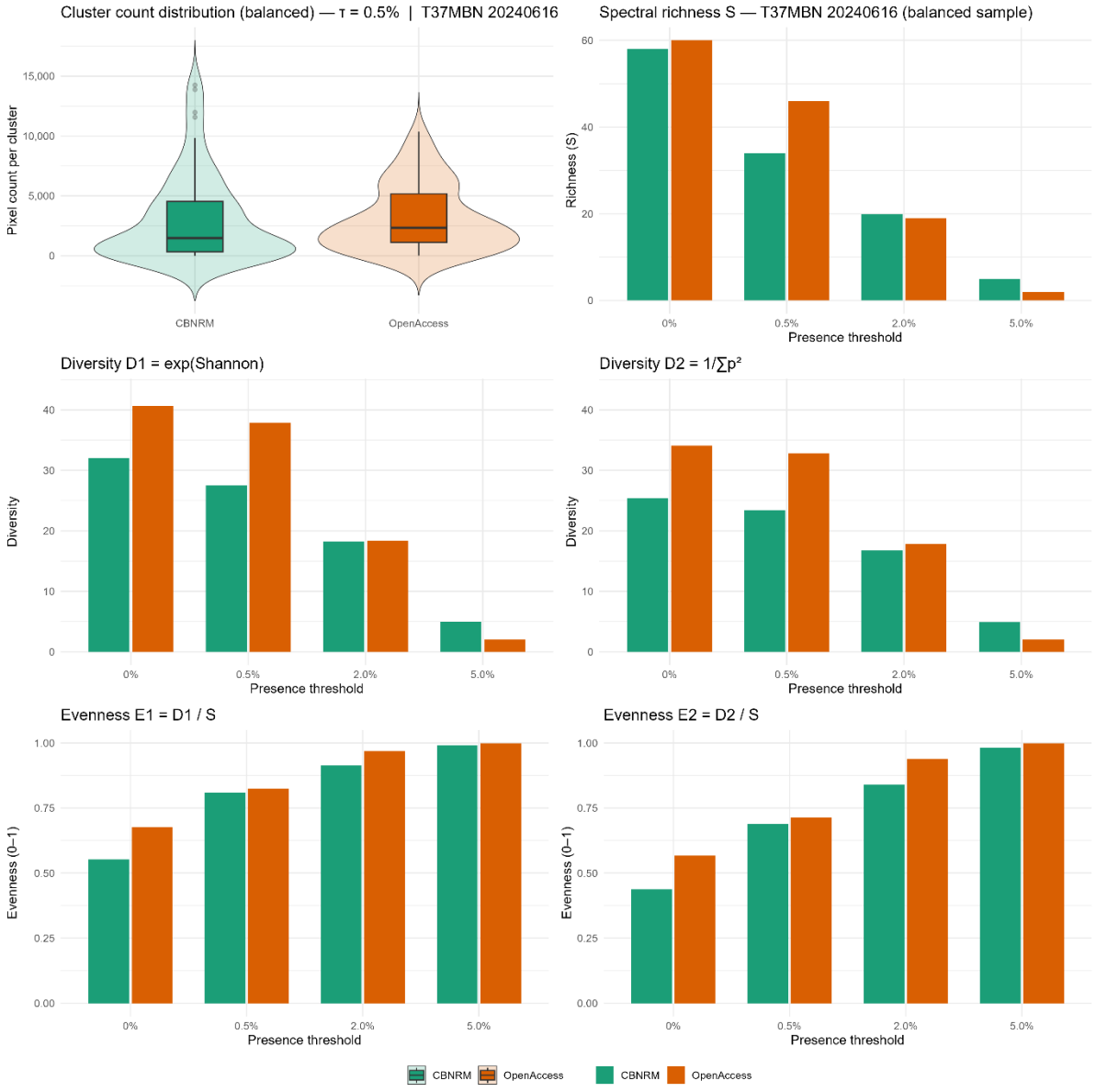

Figure 29 Local spectral diversity metrics across thresholds

The upper-left panel shows the distribution of cluster-level richness (number of occupied spectral clusters) using violin and box plots, while the upper-right panel displays total spectral richness (S) across increasing presence thresholds. The middle panels illustrate Hill diversity indices representing the effective number of distinct spectral types. The lower panels show corresponding evenness measures, indicating how uniformly spectral types are distributed within each governance category.

7 Discussion

This chapter interprets the empirical findings presented in Chapter 6 in relation to the study's research questions, conceptual framework, and existing literature. It aims to integrate results from the trait-based, spectral, and governance analyses to provide a coherent understanding of how charcoal production and forest governance influence ecological structure and resilience across scales.

7.1 Functional trait differentiation and charcoal Selectivity

This section interprets how tree species used for charcoal production differ functionally from those not utilized, focusing on their position within the broader Miombo trait space. The following subsection discusses the observed patterns and the relative strength of this functional differentiation.

7.1.1 Patterns and strength of functional differentiation

For the interpretation of trait differentiation, I will primarily draw on the PCA that included the five most data-complete traits which are specific leaf area, leaf nitrogen, leaf phosphorus, wood density, and seed mass. This model was rather balanced with 47 non-charcoal and 36 charcoal species and data completeness and was the only one to yield statistically significant group separation. The first two principal components together explained approximately 59% of total trait variation capturing the dominant ecological trade-offs underlying tree strategies in Miombo woodlands.

The first principal component represented the classical leaf–wood economics spectrum (Díaz et al., 2016), contrasting acquisitive and conservative resource-use strategies. Charcoal-producing species clustered more toward the conservative pole of this axis, characterized by high wood density, low SLA, and low foliar nitrogen and phosphorus content, whereas non-charcoal species tended toward acquisitive combinations of high SLA and nutrient-rich leaves. This pattern reflects a shift from fast-return to slow-return investment strategies (Reich, 2014), in which conservative species emphasize tissue longevity, carbon retention, and structural resilience over rapid growth. The second axis, dominated by seed mass, likely represents differences in reproductive and regeneration strategies, distinguishing species investing in few, large seeds with high establishment success from those producing many small seeds adapted to frequent disturbance (Moles and Westoby, 2006).

7.1.2 Ecological interpretation of traits shift

The weak but recurring shift toward conservative traits among charcoal-producing species may reflect the combined influence of selective harvesting and ecophysiological adaptation in Miombo woodlands. Dense, slowly growing species provide higher charcoal yields per unit volume and greater energy density (Chidumayo and Gumbo, 2013), making them consistent targets for local harvesters. Over time, such preference could impose a subtle functional filter favoring conservative trait combinations in

harvested areas. This pattern aligns with disturbance-filtering theory (Lavorel and Garnier, 2002), which posits that repeated biomass removal selects for species with stress-tolerant life-history strategies. In this context, traits such as high wood density, low specific leaf area, and low foliar nutrient content enhance survival under both anthropogenic and natural disturbance regimes (Díaz et al., 2016; Reich, 2014). These traits are associated not only with mechanical strength and tissue longevity but also with fire resistance and drought tolerance, two of the most critical adaptations for persistence in seasonally dry savannas (Pausas and Keeley, 2014).

These relationships suggest that charcoal production interacts with natural disturbance regimes rather than operating independently of them, as depicted in the conceptual framework (Figure 3). Species that persist under repeated cutting, burning, or water stress often already occupy the conservative end of the leaf–wood economics spectrum (Díaz et al., 2016). Consequently, human extraction tends to amplify existing ecological filters, reinforcing the dominance of slow-return species that invest heavily in structural tissues. This dual filtering through natural disturbance and resource-use selectivity could create a feedback loop between ecological composition and exploitation intensity. As resilient, dense-wooded species become preferentially harvested, the remaining pool increasingly reflects stress-tolerant assemblages. Such feedbacks can alter regeneration trajectories, gradually transforming the functional architecture of Miombo woodlands.

At the same time, the broad overlap between charcoal and non-charcoal species in trait space indicates substantial functional redundancy, meaning that many species share similar trait combinations or occupy adjacent ecological niches (Díaz et al., 2009). This redundancy can buffer ecosystem functioning against selective extraction in the short term, maintaining productivity and structural integrity despite species loss (Elmqvist et al., 2003). However, as disturbance persists, even minor directional filtering can lead to the erosion of redundancy, reducing the diversity of ecological strategies available for recovery after disturbance (Walker et al., 1999). This creates a trade-off between resistance and recovery: conservative communities may be stable and persistent but respond slowly to renewed disturbance or climatic variability(Holling, 1973).

In this light, the modest trait differentiation observed here may represent the emergent phase of functional reorganization, a gradual process rather than an abrupt transformation. Such reorganization aligns with broader patterns of structural fragmentation observed in unprotected landscapes, suggesting cross-scale consistency between species-level and landscape-level processes. At the species level, conservative trait dominance implies reduced functional turnover; at the landscape level, this may manifest as heterogeneous and discontinuous canopy patterns under high extraction pressure. Together, these findings provide empirical support for the first research question and hypothesis: species used for charcoal production occupy a slightly more conservative region of trait space, and this

differentiation might signify a long-term trajectory toward functional convergence and reduced ecological plasticity.

Finally, it is important to acknowledge that this pattern remains statistically modest, in part due to data heterogeneity and limited trait coverage within the TRY database. Yet, the directional consistency across analytical variants and its theoretical coherence with established ecological principles suggest a real, ecologically meaningful signal. The combined evidence points to a nuanced but consequential interaction between human resource use and natural ecological filters, where charcoal production subtly reshapes the Miombo functional landscape through repeated, scale-dependent selection pressures.

7.1.3 Critical reflection on trait analysis

The trait-based analysis presented here is constrained by several methodological and data-related limitations that warrant careful consideration. A primary limitation could stem from the use of two different species lists for data integration. While the TRY dataset provided the functional trait information for all global species, species selection and charcoal-use tagging were based on a separate compilation. This mismatch could reduce the number of overlapping taxa and therefore the analytical power of the PCA models. A more coherent approach using a single harmonized list such as the NAFRMO species list for both trait extraction and charcoal tagging, might have retained a larger subset of species, thereby improving trait coverage and group balance.

Sample size further constrained interpretive strength. Several PCA configurations included only a handful of species (in some cases fewer than ten per group), limiting the reliability of multivariate separation and precluding robust confidence ellipses or dispersion tests. Under such conditions, visualizing group centroids can still be informative but should not be overinterpreted as statistically meaningful representations of group structure. The low explained variance and small R^2 values across PERMANOVAs reinforce that any detected differentiation remains subtle and exploratory in nature

In addition, data completeness in the TRY database is uneven across both species and traits, with pronounced geographic and taxonomic biases. Tropical woody floras, including many Miombo taxa, remain underrepresented in global compilations (Kattge et al., 2020). This heterogeneity could have introduced non-random missingness that can bias the scaling of principal components. Despite these limitations, the recurring orientation of trait loadings along a conservative acquisitive axis suggests that the analyses captured a real ecological signal, albeit at low resolution.

7.2 National scale: Governance effects on vegetation and spectral diversity

This section shifts the focus from species-level traits to landscape-level vegetation patterns, examining how governance influences canopy condition and spectral organization across Tanzania's Miombo woodlands. By comparing protected and unprotected areas, it explores whether management intensity corresponds to structural gradients in vegetation density and heterogeneity. The following subsection discusses these relationships through vegetation condition metrics and their spectral correlates.

7.2.1 Vegetation condition and structural gradients

Across the national extent of the Miombo woodlands, all vegetation indices convey a consistent governance-related pattern. Protected areas exhibit higher mean values of NDVI, NDMI, and CCI, coupled with narrower distributions across the twenty independent runs. These indices jointly indicate denser, moister, and chlorophyll-richer canopies that are also more uniform spatially. Unprotected areas, in contrast, show lower mean index values but markedly wider distributions, suggesting a patchy mixture of productive and degraded vegetation states.

Such differences are not limited to mean canopy condition but also reflect the stability and coherence of vegetation structure. The smaller spread of values in protected Miombo could imply reduced exposure to selective logging, fire, or clearing, whereas the broader variability in unprotected sites could point to recurrent disturbance and regrowth cycles. These patterns are consistent with remote-sensing studies showing higher or more persistent NDVI and lower disturbance within protected areas relative to their surroundings, including Tanzania's Ruaha–Rungwa landscape and broader assessments of forest protected areas (Gizachew et al., 2020; Komba et al., 2021; Tang et al., 2011).

Ecologically, this pattern suggests that protection may reduce the frequency and spatial extent of biomass loss events, thereby allowing Miombo woodlands to approach a stable-state regime characterized by closed canopies, higher moisture retention, and lower spectral variance (Holling, 1973). In contrast, unprotected areas may represent transient or reorganization states, where disturbance frequency exceeds recovery capacity, leading to patchier canopy configurations and elevated spectral variability. In this sense, the vegetation indices capture more than just "greenness"—they reveal the functional stability of ecosystems under differing governance regimes.

7.2.2 Spectral composition and governance-related differentiation

The principal-component results suggest that governance affects how spectral variability is structured within the Miombo landscape rather than creating discrete spectral types. Protected and unprotected areas largely share the same spectral domain but differ in its internal organization: protection constrains dispersion and promotes homogeneity, whereas open-access conditions allow broader

spectral spread driven by canopy openness and heterogeneity. This distinction highlights that governance modifies the texture of spectral diversity rather than its fundamental position in spectral space.

The loadings' structure provides insight into the spectral dimensions underlying this separation. PC1 shows broadly positive contributions across the visible spectrum (B02–B04) and the red-edge and SWIR regions (B05, B11–B12), indicating a broadband brightness and vegetation-intensity gradient. This component captures overall reflectance magnitude, which tends to increase with canopy openness, reduced water content, and soil exposure. PC2, in contrast, is dominated by the red-edge and near-infrared bands (B06–B8A), which contrast with the visible range and therefore describe variation in canopy density and internal leaf scattering.

Protected Miombo woodlands cluster toward lower PC1 values, consistent with darker, moister, and more structurally cohesive canopies. Their spectral configurations are relatively compact, suggesting reduced within-class heterogeneity and more stable reflectance behavior. Unprotected woodlands occupy a broader region of spectral space, extending toward higher PC1 and PC2 values, where increased brightness, variable moisture, and mixed vegetation–soil surfaces indicate more open, disturbed, or regenerating stands. This divergence aligns with patterns observed in the vegetation indices (Section 7.3.1) and is typical of landscapes experiencing selective extraction or incomplete canopy recovery (Asner and Martin, 2009; Féret and Asner, 2014).

Although the visible bands (B02–B04) exhibit slightly higher numerical loadings on PC1, their contribution largely reflects correlated albedo across the blue–red spectrum rather than ecologically independent information. The red-edge (B05–B07) and SWIR (B11–B12) regions, despite lower magnitudes, are diagnostically more important because they are sensitive to canopy structure, water content, and lignin-cellulose absorption features—attributes that vary most strongly with disturbance intensity and management regime (Clevers and Gitelson, 2013). This explains why the red-edge and SWIR are emphasized in interpretation even when their loadings are not numerically dominant.

Overall, the PCA results indicate that protection enhances spectral cohesion and limits the spread of reflectance combinations, while unprotected Miombo exhibits a broader and more fragmented spectral envelope associated with structural variability and patchiness. These spectral contrasts provide a compositional counterpart to the functional trait differentiation described in Section 7.2, suggesting that the processes shaping species-level functional strategies, disturbance, extraction pressure, and regeneration, also govern the organization of canopy structure and reflectance properties across the landscape.

7.2.3 Spectral diversity and ecological interpretation

The diversity analysis expands on the PCA results by quantifying how governance influences the distribution and persistence of spectral types across the Miombo landscape. At permissive presence thresholds of $\tau \leq 1$ %, unprotected areas exhibit higher spectral richness, suggesting a greater number of locally unique or rare spectral clusters. Yet this advantage diminishes and often reverses at stricter thresholds, with protected areas retaining more dominant and recurrent clusters. This shift implies that unprotected landscapes harbor many small, spatially fragmented patches with distinctive reflectance signatures, while protected woodlands maintain fewer but more persistent canopy configurations.

Interpreted ecologically, high richness at low thresholds does not necessarily indicate greater biodiversity or functional complexity. Instead, it often corresponds to structural fragmentation and compositional discontinuity (Rocchini et al., 2010). The broader spectral variety in unprotected Miombo likely reflects mosaics of regrowth, bare soil, and partially cleared stands—conditions that produce multiple transient spectral types but reduce overall coherence. Protected areas, by contrast, show tighter spectral clustering and stronger dominance of common types, consistent with continuous canopy cover and more stable community composition (Gizachew et al., 2020).

The convergence of results from vegetation indices, spectral composition, and diversity metrics therefore signals a single governance-related continuum: from cohesive, moisture-retaining woodland states under protection to fragmented, spectrally heterogeneous mosaics in unprotected landscapes. This continuum parallels the functional trait differentiation described in Section 7.2, where species used for charcoal production occupied a slightly more conservative region of trait space. In both cases, disturbance and extraction appear not to erase diversity but to reshape its expression—compressing functional variability at the species level while fragmenting spectral variability at the landscape level. Spectral fragmentation and trait conservatism can thus be viewed as two manifestations of the same underlying process: the restructuring of ecological organization under resource pressure. Where governance reduces disturbance and allows regeneration, diversity becomes internally cohesive and functionally stable. Where regulation weakens, diversity becomes dispersed—still present, but increasingly partitioned into smaller fragments of structure and function.

7.2.4 Methodological reflections and data constraints

While the national-scale spectral analyses reveal consistent governance-related gradients, several methodological aspects constrain the precision and generality of these findings. The first limitation arises from the 20 m spatial resolution of Sentinel-2 Level-2A data. Miombo woodlands are characterized by a fine-grained mosaic of tree crowns, shrubs, and herbaceous cover, often interspersed with bare soil or burned patches (Frost, 1996). At this resolution, individual pixels

frequently represent mixed surfaces rather than single vegetation types, which could blur spectral contrasts between governance categories. In more open or degraded sites, this pixel mixing may dilute variability, while in dense canopies it can exaggerate apparent homogeneity, thereby moderating true structural differences.

A second consideration concerns temporal heterogeneity among the scenes analyzed. Although data were restricted to the early wet season (December–January), acquisition dates differed by up to two months across tiles. Given the strong phenological dynamics of Miombo woodlands—where leaf flush, moisture recovery, and fire events can occur within weeks, these differences may have introduced temporal noise into the spectral signal. Consequently, part of the observed spectral variation could reflect phenological differences rather than purely governance effects.

Related to this, the study captures only a single seasonal snapshot, offering a spatial but not temporal perspective on governance-related structure. Miombo ecosystems undergo pronounced intra-annual variation in greenness and canopy water content. A multi-temporal or inter-annual design could allow distinguishing persistent structural differences from transient seasonal or disturbance-driven signals, which is essential for assessing long-term ecosystem resilience.

Another potential limitation involves confounding environmental gradients. Governance categories such as “Protected” and “Unprotected” may coincide with systematic differences in wildlife, soil fertility, or proximity to infrastructure. This spatial correlation could bias interpretation by attributing vegetation differences to governance rather than to underlying biophysical or accessibility factors. As a form of omitted-variable bias. For example, higher NDVI in protected areas could partly reflect their tendency to be located in less agriculturally suitable or more remote regions rather than solely management effectiveness (Joppa and Pfaff, 2010).

Finally, the input shapefiles delineating the Miombo extent and governance boundaries form a critical foundation for all subsequent analyses but may not perfectly represent current on-the-ground conditions. The national Miombo mask, derived from existing classifications, may exclude secondary or degraded woodland types, while the protected area boundaries may not reflect recent changes in land-use or enforcement. Such uncertainties in spatial definitions propagate into sampling and classification accuracy, underscoring the need for future work to integrate updated or higher-resolution boundary datasets and local management records.

Together, these considerations suggest that while the observed spectral patterns are robust at broad scale, they should be interpreted as indicative associations rather than deterministic causal relationships. Expanding to multi-temporal, higher-resolution, and field-validated analyses would

further strengthen confidence in the observed governance-related gradients across Tanzania's Miombo woodlands.

7.3 Local scale: Governance effects on vegetation and spectral Diversity

This section discusses how governance differences between CBNRM and OA systems shape vegetation condition and spectral diversity at the local scale. By interpreting canopy greenness, chlorophyll content, and moisture variability within these two governance regimes, it evaluates how management intensity and disturbance history are expressed in spectral and structural indicators. The following subsection focuses on vegetation condition as the most direct manifestation of these local governance effects.

7.3.1 Vegetation condition under local governance systems

At the local scale, vegetation indices reveal a reversed but internally consistent pattern relative to the national-scale gradient. OA Miombo woodlands display higher mean values of NDVI, NDMI, and CCI than CBNRM sites, suggesting greener, moister, and more photosynthetically active canopies. The OA distributions, however, are considerably broader, encompassing both highly productive and degraded patches, whereas CBNRM areas show more condensed, moderate values.

This contrast can be interpreted in multiple ways. From an ecological perspective, higher mean indices in OA areas may reflect short-term regrowth or successional dynamics typical of disturbance mosaics, where recently cleared or burned sites rapidly accumulate leaf biomass and chlorophyll before reaching canopy closure (Ryan et al., 2012). In this sense, the elevated greenness of OA systems might capture transient recovery phases rather than sustained ecosystem condition. CBNRM areas, by contrast, exhibit more uniform but lower values that could indicate stable, demographically mature woodland stands with less pronounced regrowth pulses.

However, given that this analysis represents a single temporal snapshot, caution is warranted in attributing these differences directly to governance. The studied OA and CBNRM villages are real, geographically distinct landscapes rather than randomized samples of pixels. As such, their spectral properties may also reflect underlying biophysical differences for example, in soil fertility, or topography independent of management regime. Without multi-temporal or field-based validation, it remains uncertain whether the observed contrast arises primarily from governance, environmental variation, or their interaction.

In this sense, the higher vegetation indices in OA areas do not necessarily signal superior ecological status but rather a dynamic equilibrium between disturbance and recovery, where canopy vigor fluctuates spatially. The narrower and lower values in CBNRM sites imply stability over amplitude,

representing ecosystems that change less dramatically in response to local extraction. This reversal of the national-scale pattern, where protected woodlands held the highest indices, underscores the importance of scale and temporal context: at local resolution, immediate disturbance, regrowth dynamics can dominate spectral signals, temporarily masking longer-term governance effects.

Despite these uncertainties, the distributions of the vegetation indices mirror the structural logic observed on the national scale. Open-access systems, like unprotected Miombo, show wider and more heterogeneous index ranges, while CBNRM areas, akin to protected woodlands display more compact and internally consistent distributions. This resemblance suggests that, regardless of mean values, governance influences the organization and variability of vegetation states in a comparable manner across scales: disturbance-driven systems tend to expand spectral variability, while managed or protected systems constrain it. Even under ongoing charcoal-production pressure, this pattern indicates that CBNRM retains a stabilizing effect analogous to that of protection at larger spatial scales.

7.3.2 Spectral composition under local governance systems

At the local scale, the spectral composition of Miombo woodlands differs modestly but systematically between CBNRM and OA systems. The PCA of Sentinel-2 reflectance (tile T37MBN) shows that the first two components together explain 91.9 % of total variance (PC1 = 62.0 %, PC2 = 29.8 %). While both governance types occupy overlapping regions in spectral space, their centroids are displaced primarily along PC2, indicating that local governance influences subtle differences in canopy optical properties rather than broad brightness or moisture gradients.

The band loadings reveal that PC1 represents a broadband brightness–moisture gradient, loading strongly and fairly uniformly on the visible (B02–B04) and shortwave infrared (B11–B12) regions. Higher PC1 scores therefore correspond to structurally open and drier surfaces, whereas lower scores indicate denser, moister canopies with less soil exposure. In contrast, PC2 is dominated by red-edge and near-infrared bands (B06–B8A) that contrast with the visible range, capturing variation in canopy density, internal leaf scattering, and pigment absorption. The observed separation along PC2 suggests that CBNRM and OA differ most clearly in terms of leaf-level and canopy-scale optical structure.

In this spectral space, OA pixels extend farther along PC2 and spread broadly across both axes, reflecting greater internal heterogeneity within these landscapes, likely combining recently disturbed plots, regrowing stands, and residual closed-canopy patches. CBNRM pixels, by contrast, cluster more compactly toward lower PC2 values, consistent with more uniform canopy density and moisture conditions. This pattern indicates that local governance influences the coherence of spectral signatures, with managed systems showing less internal dispersion and disturbance-driven systems expressing higher spectral variability

From a broader perspective, this local configuration echoes the structural relationships observed at the national scale. In both analyses, systems under stronger management or protection occupy narrower, moisture-associated spectral domains, whereas unregulated or open-access systems display broader, fragmented spectral envelopes linked to canopy discontinuity and variable substrate exposure. Although the direction of the greenness gradient reverses between scales—OA woodlands being greener on average while protected woodlands dominate nationally—the underlying organization remains the same: dispersion is characteristic of disturbance, while compactness reflects ecological stability. This continuity suggests that the spectral composition of Miombo woodlands encodes a consistent governance signal, expressing how management intensity shapes the structural and optical coherence of vegetation across spatial scales.

7.3.3 Spectral diversity under local governance systems

Spectral diversity patterns at the local scale reflect the same structural logic identified in vegetation indices and spectral composition: OA woodlands exhibit higher richness and variability at low presence thresholds, whereas CBNRM systems retain fewer but more persistent spectral types at stricter thresholds. This pattern suggests that governance influences not only mean canopy conditions but also the distribution and stability of spectral diversity across spatial scales.

At permissive thresholds e.g., τ = 0.5 %, OA woodlands contain a greater number of rare spectral clusters, indicating fine-scale heterogeneity and mixed successional stages typical of disturbed or regenerating landscapes (Rocchini et al., 2010; Ustin and Gamon, 2010). The elevated richness in OA systems at this level does not necessarily imply greater functional diversity but rather a more fragmented spectral landscape, where many canopy or substrate conditions coexist. As the prevalence threshold increases, most of these rare types disappear, and CBNRM systems begin to dominate in terms of both richness and persistence. This inversion at higher τ values implies that CBNRM woodlands maintain a smaller set of core, repeatedly expressed spectral types—a hallmark of structurally cohesive ecosystems (Rocchini et al., 2021).

These findings align with the spectral composition results: OA systems are characterized by broad, patchy spectral envelopes, while CBNRM systems cluster more compactly in spectral space. Together, they indicate that OA diversity is driven by spatial turnover, whereas CBNRM diversity is driven by compositional stability. In ecological terms, this distinction may parallel different modes of resilience: OA landscapes may exhibit adaptive heterogeneity through rapid regrowth and regeneration cycles, while CBNRM woodlands reflect buffering stability through consistent canopy structure and reduced disturbance.

From a cross-scale perspective, the local pattern mirrors the national governance gradient between unprotected and protected Miombo woodlands. In both cases, less regulated systems (OA and unprotected) display higher spectral richness at low prevalence but lose coherence as rarity filters intensify, whereas protected or CBNRM systems preserve a more restricted yet enduring spectral signature. These two modes can be viewed as complementary expressions of ecosystem organization: disturbance expands diversity through spatial fragmentation, while management contracts it into cohesive and recurrent states.

Thus, the spectral diversity gradient under local governance systems represents another manifestation of the broader principal observed throughout this study: the fragmentation–cohesion continuum that connects trait-level, spectral, and structural dimensions of Miombo ecosystems. OA and CBNRM woodlands may therefore be understood as opposite expressions of the same underlying process—how governance modulates the spatial and temporal expression of diversity under human use pressure.

7.3.4 Methodological considerations and data constraints

While the local analysis provides valuable insights into how governance may shape vegetation condition and spectral diversity, several methodological and contextual limitations should be acknowledged. These considerations inform the interpretation of observed patterns and highlight areas for future refinement.

At the spatial level, the analysis was based on a single Sentinel-2 tile (T37MBN) encompassing a limited set of villages in the Kilosa District. This scope constrains the generalizability of results, as the observed contrasts may partly reflect local environmental or historical conditions rather than universal governance effects. The village boundaries used to define CBNRM OA zones are fundamental to the classification but may not perfectly capture the true spatial extent of forest use or enforcement. Small boundary inaccuracies or differences in the internal landscape configuration, such as village size, shape, and habitat heterogeneity could influence the relative spread of spectral values and diversity metrics.

From an analytical standpoint, the use of 20 m Sentinel-2 reflectance data necessarily integrates sub-pixel heterogeneity, potentially obscuring finer-scale canopy and understory variation. In mixed or mosaic landscapes typical of Miombo systems, individual pixels may represent composites of bare soil, regrowth, and mature woodland. Consequently, the observed differences in spectral diversity and composition may underestimate the true ecological contrasts between governance systems. Similarly, while PCA summarizes variance across bands, it prioritizes dominant gradients and may down-weight subtle spectral signals related to species composition or microstructural differences.

The interpretation of spectral diversity itself involves several assumptions. Richness and evenness metrics depend on clustering resolution (K) and presence thresholds (τ), parameters that structure how spectral types are defined and compared. Although sensitivity analyses were used to mitigate this effect, the absolute richness values should not be interpreted literally but as indicators of relative heterogeneity between governance contexts. Moreover, spectral heterogeneity does not exclusively reflect ecological diversity, it can also arise from soil brightness, topographic shading, or local illumination effects unrelated to vegetation structure.

A further consideration is temporal context. The Sentinel-2 scenes used represent a single date. Vegetation indices and spectral diversity measured at one moment may capture short-term differences in leaf phenology or regrowth rather than enduring structural properties. Moreover, the CBNRM systems examined here are relatively young. The Tanzania Tree Carbon Study (TTCS) project, which formalized CBNRM implementation in the three study villages, was only introduced in 2014 following the establishment of their Village Land Forest Reserves (Doggart, 2016; Ishengoma et al., 2016). Given that charcoal extraction and forest regeneration in Miombo woodlands operate over multi-year to decadal cycles (Chidumayo and Gumbo, 2013; Ryan et al., 2012; Syampungani et al., 2009), the ecological outcomes of these governance reforms may still be in their early stages. Thus, the spectral contrasts observed between CBNRM and OA areas likely reflect a combination of emerging management effects and legacy conditions from prior disturbance.

Taken together, these limitations suggest that the results should be interpreted as indicative patterns rather than definitive causal evidence. The observed differences between CBNRM and OA areas likely reflect a combination of management, disturbance history, and underlying environmental variation. Future work incorporating time-series imagery and field validation could better resolve the temporal trajectories of canopy recovery and governance influence, clarifying whether the observed spectral contrasts represent transient conditions or sustained structural divergence.

8 Conclusion

This concluding chapter joins the main findings of the study, evaluates their limitations, and outlines potential directions for future research on the ecological and governance dimensions of charcoal production in Miombo woodlands.

8.1.1 Answers to the research questions

This study set out to examine how charcoal production and forest governance shape the ecological characteristics and resilience of Miombo woodlands in Tanzania. The analyses addressed two main research questions that operate at complementary spatial scales: first, how the functional trait composition and spectral variability of Miombo vegetation differ between charcoal-associated and non-charcoal contexts at the national level; and second, how local-scale spectral diversity and structural stability vary between governance regimes in the Kilosa district.

For Research Question 1, the combined trait-based and national-scale spectral analyses revealed consistent ecological differentiation linked to charcoal production. The trait analysis showed that tree species commonly used for charcoal production occupy a slightly more conservative region of trait space than non-charcoal species. These species tended to exhibit higher wood density, lower specific leaf area, and reduced leaf nutrient concentrations, traits associated with slow growth, high carbon investment, and greater structural robustness. Although the statistical separation between groups was modest, it was robust across multiple trait combinations, suggesting a small but ecologically meaningful signal. This pattern indicates that human selection for preferred fuelwood species aligns with intrinsic ecological strategies, favoring dense, long-lived species that contribute to gradual shifts in woodland composition.

Complementing these findings, the national-scale spectral variability analysis revealed distinct structural differences between protected and unprotected Miombo woodlands. Protected areas exhibited higher mean vegetation index values (NDVI, NDMI, CCI) and a narrower distribution of spectral responses, indicating denser, more cohesive, and structurally consistent canopies. In contrast, unprotected Miombo displayed lower mean values and broader spectral distributions, reflecting greater disturbance, canopy fragmentation, and compositional heterogeneity. These national-scale contrasts reinforce the trait-based findings: ecological structure and function diverge under differing levels of extraction pressure, with protection fostering uniform vegetation strength and unprotected leading to variable, disturbed conditions. Together, the trait and spectral results illustrate how human use and management shape Miombo ecosystems at broad scales through intertwined biological and structural processes.

For Research Question 2, the local-scale spectral analysis in the Kilosa region further highlighted how governance mediates woodland structure and resilience. Comparisons between CBNRM and OA systems revealed a consistent contrast: OA areas displayed higher spectral richness under permissive thresholds, indicative of heterogeneous canopy conditions and disturbance-driven patchiness, whereas CBNRM areas maintained lower but more stable richness at stricter thresholds, suggesting greater structural persistence. Interestingly, the mean vegetation metrics (e.g., NDVI, NDMI, CCI) at the local scale showed a reversed pattern compared to the national analysis, with OA areas exhibiting slightly higher mean values than CBNRM sites. However, the shape of the distributions mirrored the national trend: CBNRM and protected areas shared narrower, more cohesive spectral profiles, while OA and unprotected areas displayed broader, more variable distributions. This parallel suggests that, although local scale mean vegetation strength may fluctuate with short-term management and disturbance, the underlying structural stability patterns remain consistent across scales. Overall, these results imply that CBNRM governance moderates disturbance impacts and supports canopy cohesion, while OA conditions foster a more fragmented form of heterogeneity reflective of reduced management control.

The convergence of trait-based and spectral evidence across spatial scales thus highlights that sustainable woodland resilience depends on aligning ecological characteristics with effective management institutions like CBNRM.

8.1.2 Limitations of the study

While the analyses provide new insight into how functional traits and governance influence Miombo woodland dynamics, several limitations constrain the generalization and precision of the results. First, the spatial and temporal scope of the satellite data restricts inference about long-term dynamics. The spectral analyses were based on single-date Sentinel-2 imagery within a limited seasonal window, meaning that short-term phenological variation or recent disturbances could affect vegetation metrics independently of management or ecological processes. Multi-temporal analyses would be required to confirm whether the observed structural differences persist through time.

Second, the resolution and sampling design introduce uncertainty in how canopy heterogeneity was represented. The 20 m spatial resolution of Sentinel-2 may obscure fine-scale structural variation and understory dynamics that contribute to biodiversity and regeneration. Likewise, the local analysis relied on a single tile covering the Kilosa district, which limits spatial representativeness and reduces statistical independence among governance classes.

Third, the trait-based dataset was constrained by uneven species coverage and variable data quality across the TRY database. Although harmonization and filtering procedures ensured comparability, the

available trait information may not capture the full ecological diversity of Miombo tree species, particularly those less studied or locally endemic.

Fourth, the analytical sensitivity of the spectral richness framework should be acknowledged. Richness and evenness estimates depend on clustering parameters and are therefore best interpreted as relative indicators of structural complexity, rather than absolute measures of biodiversity. While the replicated design and sensitivity analyses enhanced robustness, the metrics remain influenced by parameterization and data noise.

Finally, potential covariation was introduced by defining the protection status via the IUCN management categories. Differences in permitted human activity, enforcement intensity, or ecological zoning within these categories could produce vegetation contrasts independent of charcoal accessibility. Thus, some of the spectral differentiation attributed to “protection” may partially reflect variation in IUCN management regimes, rather than charcoal pressure alone.

8.1.3 Outlook and open research directions

The findings of this study demonstrate that charcoal production does not inherently lead to the ecological fragmentation often associated with resource extraction. Instead, the observed contrasts between governance types reveal that institutional design and management practices play a decisive role in shaping woodland structure and resilience. CBNRM systems, despite allowing controlled charcoal harvesting, exhibited spectral and structural characteristics more similar to protected areas than to unprotected systems. This suggests that well-regulated local governance can effectively moderate the ecological impacts of charcoal production, maintaining canopy cohesion and functional integrity even under use pressure.

Future research should therefore focus on the temporal and institutional dimensions of this governance–resilience relationship. Multi-temporal satellite analyses could reveal whether the structural stability observed under CBNRM persists through time, capturing regrowth and recovery processes after extraction events. Such analyses would help clarify whether the spectral coherence seen in community-managed woodlands reflects transient regulation or long-term ecological resilience. Integrating high-frequency remote-sensing data with field-based observations of regeneration, species composition, and biomass would provide a direct link between canopy structure and ecosystem function.

A second avenue concerns comparative analysis across governance contexts and IUCN protection categories. Expanding the study to include multiple CBNRM sites at different stages of establishment and protected areas under varying management regimes could help disentangle the effects of institutional maturity, enforcement capacity, and ecological baseline conditions. This would provide

empirical evidence on which institutional arrangements most effectively balance local livelihood needs with ecological sustainability.

Finally, the integration of socio-ecological indicators like harvesting intensity, charcoal market dependency, and community benefit distribution would enhance understanding of how governance translates into ecological outcomes. By combining ecological monitoring with institutional and economic metrics, future work could inform policies that promote equitable, low-impact charcoal production systems. The broader implication is that charcoal extraction can coexist with woodland resilience, provided that governance frameworks incentivize sustainable management and align local benefits with long-term ecological stability.

9 Synthesis

This thesis combined functional trait analysis and spectral variability assessment to examine how charcoal production and governance shape the ecological structure and resilience of Miombo woodlands in Tanzania. Across both analytical scales, the results converge on a unifying principal: the ecological consequences of charcoal extraction are not predetermined by use itself, but by the governance systems and functional composition through which use occurs. The integration of trait-based and remote-sensing evidence shows that woodland ecosystems exist along a continuum between fragmentation and cohesion, where the balance between disturbance and regulation determines resilience outcomes.

At the theoretical level, this study contributes to the broader discourse on resilience and functional diversity. Ecological resilience is shaped not only by the diversity of species, but by the functional traits that mediate how organisms respond to and recover from disturbance. The trait-based analyses demonstrated that charcoal-producing species tend to exhibit conservative functional strategies like high wood density, low specific leaf area, and low foliar nutrient content. These traits enhance structural stability and slow turnover, implying that human selection for fuelwood species, while selective, does not necessarily erode ecological function. Instead, it creates a form of anthropogenic filtering that favors species adapted to resource scarcity and disturbance, potentially lowering the resilience of the ecosystem as a whole.

The national-scale spectral analysis extended this understanding to the landscape level, linking trait-based variation with patterns of vegetation structure observable from space. Protected Miombo areas displayed higher mean vegetation indices (NDVI, NDMI, CCI) and narrower spectral distributions, indicative of dense and cohesive canopy conditions. Unprotected areas, in contrast, showed lower mean values and broader distributions patterns consistent with greater disturbance and canopy heterogeneity. These findings support theoretical expectations that disturbance promotes diversity but can undermine system-level structure if unchecked.

The local scale analysis refined this relationship by explicitly examining the moderating effect of governance. In the Kilosa district, spectral variability patterns under CBNRM and OA conditions revealed that management institutions can substantially alter ecological outcomes even under similar extraction contexts. While OA areas exhibited higher mean vegetation indices contrasting with the national scale protection gradient they also showed broader spectral distributions, indicating greater variability and patchiness. CBNRM areas, meanwhile, displayed narrower and more cohesive distributions of vegetation indexes, mirroring the structural stability of protected sites. This suggests that governance systems capable of enforcing rules and distributing resource rights equitably can

decouple charcoal production from ecological degradation. The presence of institutional control, rather than the absence of extraction, is thus the key determinant of woodland integrity.

From a theoretical perspective, this synthesis bridges two often disconnected domains, functional ecology and remote sensing within the framework of sustainability science. Functional traits describe the biological mechanisms underlying resilience, while spectral variability captures their aggregated manifestation in canopy structure. The integration of these perspectives allows resilience to be quantified across scales, linking the biological potential for recovery with the observed structural expression of stability. This approach contributes to operationalizing resilience theory by offering measurable, comparable indicators that connect species-level function, landscape pattern, and governance context.

From an applied standpoint, the findings challenge the assumption that charcoal production inherently drives forest degradation. Instead, the findings suggest that degradation occurs when responsibility for forest management is detached from those who use it. CBNRM systems that combine local participation with rule enforcement maintain ecological cohesion while supporting rural energy livelihoods. This reframes sustainable forest management from a trade-off between conservation and use toward a synergistic model in which managed use contributes to long-term ecosystem stability. In this sense, the Miombo woodlands provide a tangible example of how human activity, when structured by effective governance, can reinforce rather than erode ecological resilience.

In sum, this study shows that the ecological impact of charcoal production depends on both what species are used and how people are allowed to use them. The traits of the trees set the limits of how forests can recover, but governance decides whether that potential is realized. When responsibility is shared with local users, charcoal production can become a managed practice rather than a destructive one. Resilience in the Miombo is therefore not fixed, it grows from the relationship between people, institutions, and ecosystems.

10 References

Adio, A.A., Saliu, A.O., Akanbi-Gada, M.A., Najeemdeen, B.A., 2022. Effects of Charcoal Production on Soil Physicochemical Properties in Moro Local Government Area of Kwara State, Nigeria. Journal of Environmental Protection 13, 220–232. https://doi.org/10.4236/jep.2022.132014

Agyei, F.K., Hansen, C.P., Acheampong, E., 2018. Profit and profit distribution along Ghana's charcoal commodity chain. Energy for Sustainable Development 47, 62–74. https://doi.org/10.1016/j.esd.2018.09.002

Ahrends, A., Burgess, N.D., Milledge, S.A.H., Bulling, M.T., Fisher, B., Smart, J.C.R., Clarke, G.P., Mhoro, B.E., Lewis, S.L., 2010. Predictable waves of sequential forest degradation and biodiversity loss spreading from an African city. Proceedings of the National Academy of Sciences 107, 14556–14561. https://doi.org/10.1073/pnas.0914471107

Anderson, M.J., 2001. A new method for non-parametric multivariate analysis of variance. Austral Ecology 26, 32–46. https://doi.org/10.1111/j.1442-9993.2001.01070.pp.x

Asner, G.P., Martin, R.E., 2009. Airborne spectranomics: mapping canopy chemical and taxonomic diversity in tropical forests. Frontiers in Ecology and the Environment 7, 269–276. https://doi.org/10.1890/070152

Bailis, R., Drigo, R., Ghilardi, A., Masera, O., 2015. The carbon footprint of traditional woodfuels. Nature Clim Change 5, 266–272. https://doi.org/10.1038/nclimate2491

Baumert, S., Luz, A.C., Fisher, J., Vollmer, F., Ryan, C.M., Patenaude, G., Zorrilla-Miras, P., Artur, L., Nhantumbo, I., Macqueen, D., 2016. Charcoal supply chains from Mabalane to Maputo: Who benefits? Energy for Sustainable Development 33, 129–138. https://doi.org/10.1016/j.esd.2016.06.003

Bhattacharyya, A., 1943. On some sets of sufficient conditions leading to the normal bivariate distribution. Sankhyā: The Indian Journal of Statistics 399–406.

Blomley, T., Pfliegner, K., Isango, J., Zahabu, E., Ahrends, A., Burgess, N., 2008. Seeing the wood for the trees: an assessment of the impact of participatory forest management on forest condition in Tanzania. Oryx 42, 380–391. https://doi.org/10.1017/S0030605308071433

Branch, A., Martiniello, G., 2018. Charcoal power: The political violence of non-fossil fuel in Uganda. Geoforum 97, 242–252. https://doi.org/10.1016/j.geoforum.2018.09.012

Broto, V.C., Baptista, I., Kirshner, J., Smith, S., Neves Alves, S., 2018. Energy justice and sustainability transitions in Mozambique. Applied Energy 228, 645–655. https://doi.org/10.1016/j.apenergy.2018.06.057

Cadotte, M.W., Carscadden, K., Mirotchnick, N., 2011. Beyond species: functional diversity and the maintenance of ecological processes and services. Journal of Applied Ecology 48, 1079–1087. https://doi.org/10.1111/j.1365-2664.2011.02048.x

Campbell, B.M., 1996. The Miombo in transition: woodlands and welfare in Africa. CIFOR.

Cazzolla Gatti, R., Castaldi, S., Lindsell, J.A., Coomes, D.A., Marchetti, M., Maesano, M., Di Paola, A., Paparella, F., Valentini, R., 2015. The impact of selective logging and clearcutting on forest structure, tree diversity and above-ground biomass of African tropical forests. Ecological Research 30, 119–132. https://doi.org/10.1007/s11284-014-1217-3

Chao, A., Jost, L., 2012. Coverage-based rarefaction and extrapolation: standardizing samples by completeness rather than size. Ecology 93, 2533–2547. https://doi.org/10.1890/11-1952.1

Chase, J.M., Myers, J.A., 2011. Disentangling the importance of ecological niches from stochastic processes across scales. Philosophical Transactions of the Royal Society B: Biological Sciences 366, 2351–2363. https://doi.org/10.1098/rstb.2011.0063

Chidumayo, E.N., Gumbo, D.J., 2013. The environmental impacts of charcoal production in tropical ecosystems of the world: A synthesis. Energy for Sustainable Development, Special Issue on Charcoal 17, 86–94. https://doi.org/10.1016/j.esd.2012.07.004

Chidumayo, E.N., Gumbo, D.J., 2010. The dry forests and woodlands of Africa: managing for products and services. Earthscan.

Chima, U.D., Adedeji, G.A., Uloho, K.O., 2013. Preliminary assessment of the soil impact of charcoal production in Rivers State, Nigeria. Ethiopian Journal of Environmental Studies and Management 6, 285–293. https://doi.org/10.4314/ejesm.v6i3.9

Clevers, J.G.P.W., Gitelson, A.A., 2013. Remote estimation of crop and grass chlorophyll and nitrogen content using red-edge bands on Sentinel-2 and -3. International Journal of applied Earth Observation and Geoinformation 23, 344–351. https://doi.org/10.1016/j.jag.2012.10.008

Colding, J., Barthel, S., 2019. Exploring the social-ecological systems discourse 20 years later. Ecology and Society 24. https://doi.org/10.5751/ES-10598-240102

Cole, L.E.S., Bhagwat, S.A., Willis, K.J., 2014. Recovery and resilience of tropical forests after disturbance. Nat Commun 5, 3906. https://doi.org/10.1038/ncomms4906

Connell, J.H., 1978. Diversity in Tropical Rain Forests and Coral Reefs. Science 199, 1302–1310. https://doi.org/10.1126/science.199.4335.1302

Díaz, S., Hector, A., Wardle, D.A., 2009. Biodiversity in forest carbon sequestration initiatives: not just a side benefit. Current Opinion in Environmental Sustainability 1, 55–60. https://doi.org/10.1016/j.cosust.2009.08.001

Díaz, S., Kattge, J., Cornelissen, J.H.C., Wright, I.J., Lavorel, S., Dray, S., Reu, B., Kleyer, M., Wirth, C., Colin Prentice, I., Garnier, E., Bönisch, G., Westoby, M., Poorter, H., Reich, P.B., Moles, A.T., Dickie, J., Gillison, A.N., Zanne, A.E., Chave, J., Joseph Wright, S., Sheremet'ev, S.N., Jactel, H., Baraloto, C., Cerabolini, B., Pierce, S., Shipley, B., Kirkup, D., Casanoves, F., Joswig, J.S., Günther, A., Falczuk, V., Rüger, N., Mahecha, M.D., Gorné, L.D., 2016. The global spectrum of plant form and function. Nature 529, 167–171. https://doi.org/10.1038/nature16489

Díaz, S., Lavorel, S., de Bello, F., Quétier, F., Grigulis, K., Robson, T.M., 2007. Incorporating plant functional diversity effects in ecosystem service assessments. Proceedings of the National Academy of Sciences 104, 20684–20689. https://doi.org/10.1073/pnas.0704716104

Doggart, N., 2016. A review of policy instruments relevant to the integration of sustainable charcoal production in community based forest management in Tanzania (No. Technical Paper 51). Tanzania Forest Conservation Group (TFCG), Dar es Salaam, Tanzania.

Dudley, N., 2008. Guidelines for applying protected area management categories. Iucn.

Elmqvist, T., Folke, C., Nyström, M., Peterson, G., Bengtsson, J., Walker, B., Norberg, J., 2003. Response diversity, ecosystem change, and resilience. Frontiers in Ecology and the Environment 1, 488–494. https://doi.org/10.1890/1540-9295(2003)001%255B0488:RDECAR%255D2.0.CO;2

European Space Agency, 2015. Sentinel-2 User Handbook.

European Union, Copernicus, 2024. Copernicus Sentinel-2 Level-2A Data.

FAO, 2017. The Charcoal Transition: Greening the Charcoal Value Chain to Mitigate Climate Change and Improve Local Livelihoods.

Féret, J.-B., Asner, G.P., 2014. Mapping tropical forest canopy diversity using high-fidelity imaging spectroscopy. Ecol Appl 24, 1289–1296. https://doi.org/10.1890/13-1824.1

Folke, C., Carpenter, S., Walker, B., Scheffer, M., Elmqvist, T., Gunderson, L., Holling, C.S., 2004. Regime Shifts, Resilience, and Biodiversity in Ecosystem Management. Annual Review of Ecology, Evolution, and Systematics 35, 557–581. https://doi.org/10.1146/annurev.ecolsys.35.021103.105711

Fouquet, R., 2010. The slow search for solutions: Lessons from historical energy transitions by sector and service. Energy Policy, Energy Efficiency Policies and Strategies with regular papers. 38, 6586–6596. https://doi.org/10.1016/j.enpol.2010.06.029

Frost, P., 1996. The ecology of miombo woodlands, in: The Miombo in Transition: Woodlands and Welfare in Affrica. B. Campbell, pp. 11–58.

Gao, B., 1996. NDWI—A normalized difference water index for remote sensing of vegetation liquid water from space. Remote Sensing of Environment 58, 257–266. https://doi.org/10.1016/S0034-4257(96)00067-3

Garedew, B., Simon, L., 2018. Survey of Charcoal Production and its Impact on Plant Diversity and Conservation Challenges in Abeshige District, Gurage Zone, Ethiopia. Journal of Biodiversity & Endangered Species 6, 1–11.

Gitelson, A.A., Merzlyak, M.N., 1998. Remote sensing of chlorophyll concentration in higher plant leaves. Advances in Space Research, Synergistic Use of Multisensor Data for Land Processes 22, 689–692. https://doi.org/10.1016/S0273-1177(97)01133-2

Gizachew, B., Rizzi, J., Shirima, D.D., Zahabu, E., 2020. Deforestation and Connectivity among Protected Areas of Tanzania. Forests 11, 170. https://doi.org/10.3390/f11020170

Goncalves, F.M.P., 2019. Effect of shifting cultivation and charcoal production on structure, dynamic and above-ground biomass in the Angolan miombo and dry woodlands (doctoralThesis). Staats- und Universitätsbibliothek Hamburg Carl von Ossietzky.

Gotelli, N.J., Colwell, R.K., 2001. Quantifying biodiversity: procedures and pitfalls in the measurement and comparison of species richness. Ecology Letters 4, 379–391. https://doi.org/10.1046/j.1461-0248.2001.00230.x

Guta, D., Baumgartner, J., Jack, D., Carter, E., Shen, G., Orgill-Meyer, J., Rosenthal, J., Dickinson, K., Bailis, R., Masuda, Y., Zerriffi, H., 2022. A systematic review of household energy transition in low and middle income countries. Energy Research & Social Science 86, 102463. https://doi.org/10.1016/j.erss.2021.102463

Hardin, G., 1968. The Tragedy of the Commons. Science 162, 1243–1248. https://doi.org/10.1126/science.162.3859.1243

Harfoot, M.B.J., Tittensor, D.P., Knight, S., Arnell, A.P., Blyth, S., Brooks, S., Butchart, S.H.M., Hutton, J., Jones, M.I., Kapos, V., Scharlemann, J.P.W., Burgess, N.D., 2018. Present and future biodiversity risks from fossil fuel exploitation. Conservation Letters 11, e12448. https://doi.org/10.1111/conl.12448

Hektor, B., Backéus, S., Andersson, K., 2016. Carbon balance for wood production from sustainably managed forests. Biomass and Bioenergy 93, 1–5. https://doi.org/10.1016/j.biombioe.2016.05.025

Holling, C.S., 1973. Resilience and Stability of Ecological Systems. Annual Review of Ecology, Evolution, and Systematics 4, 1–23. https://doi.org/10.1146/annurev.es.04.110173.000245

Ishengoma, R.C., Katani, J.Z., Abdallah, J.M., Haule, O., Deogratias, K., Olomi, J., 2016. Kilosa District Harvesting Plan.

Jeffreys, H., 1997. An invariant form for the prior probability in estimation problems. Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences 186, 453–461. https://doi.org/10.1098/rspa.1946.0056

Jolliffe, I.T., Cadima, J., 2016. Principal component analysis: a review and recent developments. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 374, 20150202. https://doi.org/10.1098/rsta.2015.0202

Jones, D., Ryan, C.M., Fisher, J., 2016. Charcoal as a diversification strategy: The flexible role of charcoal production in the livelihoods of smallholders in central Mozambique. Energy for Sustainable Development 32, 14–21. https://doi.org/10.1016/j.esd.2016.02.009

Joppa, L.N., Pfaff, A., 2010. Global protected area impacts. Proceedings of the Royal Society B: Biological Sciences 278, 1633–1638. https://doi.org/10.1098/rspb.2010.1713

Jost, L., 2007. Partitioning Diversity into Independent Alpha and Beta Components. Ecology 88, 2427–2439. https://doi.org/10.1890/06-1736.1

Jost, L., 2006. Entropy and diversity. Oikos 113, 363–375. https://doi.org/10.1111/j.2006.0030-1299.14714.x

Kamwilu, E., Duguma, L.A., Orero, L., 2021. The Potentials and Challenges of Achieving Sustainability through Charcoal Producer Associations in Kenya: A Missed Opportunity? Sustainability 13, 2288. https://doi.org/10.3390/su13042288

Kattge, J., Bönisch, G., Díaz, S., Lavorel, S., Prentice, I.C., Leadley, P., Tautenhahn, S., Werner, G., 2020. TRY plant trait database - enhanced coverage and open access. Global Change Biology. https://doi.org/10.1111/gcb.14904

Kiruki, H.M., van der Zanden, E.H., Gikuma-Njuru, P., Verburg, P.H., 2017. The effect of charcoal production and other land uses on diversity, structure and regeneration of woodlands in a semi-arid area in Kenya. Forest Ecology and Management 391, 282–295. https://doi.org/10.1016/j.foreco.2017.02.030

Kojima, M., 2011. The role of liquefied petroleum gas in reducing energy poverty.

Komba, A.W., Watanabe, T., Kaneko, M., Chand, M.B., 2021. Monitoring of Vegetation Disturbance around Protected Areas in Central Tanzania Using Landsat Time-Series Data. Remote Sensing 13, 1800. https://doi.org/10.3390/rs13091800

Kursa, M.B., Rudnicki, W.R., 2010. Feature Selection with the Boruta Package. Journal of Statistical Software 36, 1–13. https://doi.org/10.18637/jss.v036.i11

Lasota, J., Błońska, E., Babiak, T., Piaszczyk, W., Stępniewska, H., Jankowiak, R., Boroń, P., Lenart-Boroń, A., 2021. Effect of Charcoal on the Properties, Enzyme Activities and Microbial Diversity of Temperate Pine Forest Soils. Forests 12, 1488. https://doi.org/10.3390/f12111488

Lavorel, S., Garnier, E., 2002. Predicting changes in community composition and ecosystem functioning from plant traits: revisiting the Holy Grail. Functional Ecology 16, 545–556. https://doi.org/10.1046/j.1365-2435.2002.00664.x

Lillesø, J.-P.B., van Breugel, P., Kindt, R., Bingham, M., Demissew, S., Dudley, C., Friis, I., Gachathi, F., Kalema, J., Mbago, F., Minani, V., Moshi, H., Mulumba, J., Namaganda, M., Ndangalasi, H., Ruffo, C., Jamnadass, R., Graudal, L., 2024. Potential Natural Vegetation of Eastern Africa (Burundi, Ethiopia, Kenya, Malawi, Rwanda, Tanzania, Uganda and Zambia): raster and vector GIS files for each country. https://doi.org/10.5281/zenodo.11125645

Linder, H.P., de Klerk, H.M., Born, J., Burgess, N.D., Fjeldså, J., Rahbek, C., 2012. The partitioning of Africa: statistically defined biogeographical regions in sub-Saharan Africa. Journal of Biogeography 39, 1189–1205. https://doi.org/10.1111/j.1365-2699.2012.02728.x

Lund, J.F., Treue, T., 2008. Are We Getting There? Evidence of Decentralized Forest Management from the Tanzanian Miombo Woodlands. World Development, Special Section: Social Movements and the Dynamics of Rural Development in Latin America (pp. 2874-2952) 36, 2780–2800. https://doi.org/10.1016/j.worlddev.2008.01.014

Mackey, R.L., Currie, D.J., 2001. The Diversity–Disturbance Relationship: Is It Generally Strong and Peaked? Ecology 82, 3479–3492. https://doi.org/10.1890/0012-9658(2001)082%255B3479:TDDRII%255D2.0.CO;2

MacQueen, J., 1967. Multivariate observations, in: Proceedings Ofthe 5th Berkeley Symposium on Mathematical Statisticsand Probability. pp. 281–297.

Magurran, A.E., 2004. Measuring Biological Diversity. Blackwell Publishing, Oxford, United Kingdom.

Mahalanobis, P.C., 1936. A note on the statistical and biometric writings of Karl Pearson. Sankhyā: The Indian Journal of Statistics (1933-1960) 2, 411–422.

Malimbwi, R., Zahabu, E., Misana, S., Monela, G., Jambiya, G., Mchome, B., 2005. CHARCOAL POTENTIAL OF MIOMBO WOODLANDS AT KITULANGALO, TANZANIA. Journal of Tropical Forest Science.

Malimbwi, R.E., Zahabu, E.M., 2008. Woodlands and the charcoal trade: the case of Dar es Salaam City.

MNRT, FAO, Ministry for Foreign Affairs of Finland, 2011. National Forestry Resources Monitoring and Assessment of Tanzania (NAFORMA) – Species List. Ministry of Natural Resources and Tourism (MNRT), Dar es Salaam, Tanzania.

Moles, A.T., Westoby, M., 2006. Seed size and plant strategy across the whole life cycle. Oikos 113, 91–105. https://doi.org/10.1111/j.0030-1299.2006.14194.x

Molino, J.-F., Sabatier, D., 2001. Tree Diversity in Tropical Rain Forests: A Validation of the Intermediate Disturbance Hypothesis. Science 294, 1702–1704. https://doi.org/10.1126/science.1060284

Morello, T.F., 2015. Carbon neutral merchant pig iron in Brazil: Alternatives that allow decoupling from deforestation. Energy for Sustainable Development 27, 93–104. https://doi.org/10.1016/j.esd.2015.04.008

Ostrom, E., 2009. A General Framework for Analyzing Sustainability of Social-Ecological Systems. Science 325, 419–422. https://doi.org/10.1126/science.1172133

Pausas, J.G., Keeley, J.E., 2014. Evolutionary ecology of resprouting and seeding in fire-prone ecosystems. New Phytologist 204, 55–65. https://doi.org/10.1111/nph.12921

Petchey, O.L., Gaston, K.J., 2006. Functional diversity: back to basics and looking forward. Ecology Letters 9, 741–758. https://doi.org/10.1111/j.1461-0248.2006.00924.x

Pickett, S.T.A., Kolasa, J., Armesto, J.J., Collins, S.L., 1989. The Ecological Concept of Disturbance and Its Expression at Various Hierarchical Levels. Oikos 54, 129–136. https://doi.org/10.2307/3565258

Piketty, M.-G., Wichert, M., Fallot, A., Aimola, L., 2009. Assessing land availability to produce biomass for energy: The case of Brazilian charcoal for steel making. Biomass and Bioenergy 33, 180–190. https://doi.org/10.1016/j.biombioe.2008.06.002

Rajala, T., 2022. NAFORMA: National Forest Resources Monitoring and Assessment of Tanzania Mainland. Sampling Design Options for 2nd Biophysical Inventory (NAFORMA II). FAO, Rome, Italy. https://doi.org/10.4060/cc0572en

Reich, P.B., 2014. The world-wide 'fast–slow' plant economics spectrum: a traits manifesto. Journal of Ecology 102, 275–301. https://doi.org/10.1111/1365-2745.12211

Ribeiro, N.S., Silva de Miranda, P.L., Timberlake, J., 2020. Biogeography and Ecology of Miombo Woodlands, in: Ribeiro, N.S., Katerere, Y., Chirwa, P.W., Grundy, I.M. (Eds.), Miombo Woodlands in a

Changing Environment: Securing the Resilience and Sustainability of People and Woodlands. Springer International Publishing, Cham, pp. 9–53. https://doi.org/10.1007/978-3-030-50104-4_2

Rocchini, D., Balkenhol, N., Carter, G.A., Foody, G.M., Gillespie, T.W., He, K.S., Kark, S., Levin, N., Lucas, K., Luoto, M., Nagendra, H., Oldeland, J., Ricotta, C., Southworth, J., Neteler, M., 2010. Remotely sensed spectral heterogeneity as a proxy of species diversity: Recent advances and open challenges. Ecological Informatics, Special Issue on Advances of Ecological Remote Sensing Under Global Change 5, 318–329. https://doi.org/10.1016/j.ecoinf.2010.06.001

Rocchini, D., Salvatori, N., Beierkuhnlein, C., Chiarucci, A., de Boissieu, F., Förster, M., Garzon-Lopez, C.X., Gillespie, T.W., Hauffe, H.C., He, K.S., Kleinschmit, B., Lenoir, J., Malavasi, M., Moudrý, V., Nagendra, H., Payne, D., Šímová, P., Torresani, M., Wegmann, M., Féret, J.-B., 2021. From local spectral species to global spectral communities: A benchmark for ecosystem diversity estimate by remote sensing. Ecological Informatics 61, 101195. https://doi.org/10.1016/j.ecoinf.2020.101195

Rouse Jr, J.W., Haas, R.H., Deering, D.W., Schell, J.A., Harlan, J.C., 1974. Monitoring the vernal advancement and retrogradation (green wave effect) of natural vegetation.

Ryan, C.M., Hill, T., Woollen, E., Ghee, C., Mitchard, E., Cassells, G., Grace, J., Woodhouse, I.H., Williams, M., 2012. Quantifying small-scale deforestation and forest degradation in African woodlands using radar imagery. Global Change Biology 18, 243–257. https://doi.org/10.1111/j.1365-2486.2011.02551.x

Ryan, C.M., Pritchard, R., McNicol, I., Owen, M., Fisher, J.A., Lehmann, C., 2016. Ecosystem services from southern African woodlands and their future under global change. Philosophical Transactions of the Royal Society B: Biological Sciences 371, 20150312. https://doi.org/10.1098/rstb.2015.0312

Sander, K., Gros, C., Peter, C., 2013. Enabling reforms: Analyzing the political economy of the charcoal sector in Tanzania. Energy for Sustainable Development, Special Issue on Charcoal 17, 116–126. https://doi.org/10.1016/j.esd.2012.11.005

Santos, M.J., Dekker, S.C., Daioglou, V., Braakhekke, M.C., van Vuuren, D.P., 2017. Modeling the Effects of Future Growing Demand for Charcoal in the Tropics. Front. Environ. Sci. 5. https://doi.org/10.3389/fenvs.2017.00028

Schure, J., Levang, P., Wiersum, K.F., 2014. Producing Woodfuel for Urban Centers in the Democratic Republic of Congo: A Path Out of Poverty for Rural Households? World Development, Forests, Livelihoods, and Conservation 64, S80–S90. https://doi.org/10.1016/j.worlddev.2014.03.013

Scoones, I., 1998. Sustainable Rural Livelihoods: A Framework for Analysis. The Institute of Development Studies and Partner Organisations.

Smith, H.E., Jones, D., Vollmer, F., Baumert, S., Ryan, C.M., Woollen, E., Lisboa, S.N., Carvalho, M., Fisher, J.A., Luz, A.C., Grundy, I.M., Patenaude, G., 2019. Urban energy transitions and rural income generation: Sustainable opportunities for rural development through charcoal production. World Development 113, 237–245. https://doi.org/10.1016/j.worlddev.2018.08.024

Sousa, W.P., 1984. The Role of Disturbance in Natural Communities. Annual Review of Ecology, Evolution, and Systematics 15, 353–391. https://doi.org/10.1146/annurev.es.15.110184.002033

Syampungani, S., Chirwa, P.W., Akinnifesi, F.K., Sileshi, G., Ajayi, O.C., 2009. The miombo woodlands at the cross roads: Potential threats, sustainable livelihoods, policy gaps and challenges. Natural Resources Forum 33, 150–159. https://doi.org/10.1111/j.1477-8947.2009.01218.x

Syampungani, S., Tigabu, M., Matakala, N., Handavu, F., Oden, P.C., 2017. Coppicing ability of dry miombo woodland species harvested for traditional charcoal production in Zambia: a win–win strategy for sustaining rural livelihoods and recovering a woodland ecosystem. J. For. Res. 28, 549–556. https://doi.org/10.1007/s11676-016-0307-1

Tang, Z., Fang, J., Sun, J., Gaston, K.J., 2011. Effectiveness of Protected Areas in Maintaining Plant Production. PLOS ONE 6, e19116. https://doi.org/10.1371/journal.pone.0019116

Tarimo, B., Dick, Ø.B., Gobakken, T., Totland, Ø., 2015. Spatial distribution of temporal dynamics in anthropogenic fires in miombo savanna woodlands of Tanzania. Carbon Balance Manage 10, 18. https://doi.org/10.1186/s13021-015-0029-2

Turner, M.G., 2010. Disturbance and landscape dynamics in a changing world. Ecology 91, 2833–2849. https://doi.org/10.1890/10-0097.1

UNEP-WCMC and IUCN, 2025. Protected Planet: The World Database on Protected Areas (WDPA).

United Republic of Tanzania, 1998. National Forest Policy. Ministry of Natural Resources and Tourism, Forestry and Beekeeping Division, Dar es Salaam, Tanzania.

Ustin, S.L., Gamon, J.A., 2010. Remote sensing of plant functional types. New Phytologist 186, 795–816. https://doi.org/10.1111/j.1469-8137.2010.03284.x

van ’t Veen, 2022. Effects of Transitions in Charcoal Production Systems on Forests and Livelihoods.

Venter, O., Sanderson, E.W., Magrach, A., Allan, J.R., Beher, J., Jones, K.R., Possingham, H.P., Laurance, W.F., Wood, P., Fekete, B.M., Levy, M.A., Watson, J.E.M., 2016. Sixteen years of change in the global terrestrial human footprint and implications for biodiversity conservation. Nat Commun 7, 12558. https://doi.org/10.1038/ncomms12558

Villéger, S., Mason, N.W.H., Mouillot, D., 2008. New Multidimensional Functional Diversity Indices for a Multifaceted Framework in Functional Ecology. Ecology 89, 2290–2301. https://doi.org/10.1890/07-1206.1

Violle, C., Navas, M.-L., Vile, D., Kazakou, E., Fortunel, C., Hummel, I., Garnier, E., 2007. Let the concept of trait be functional! Oikos 116, 882–892. https://doi.org/10.1111/j.0030-1299.2007.15559.x

Vollmer, F., Zorrilla-Miras, P., Baumert, S., Luz, A.C., Woollen, E., Grundy, I., Artur, L., Ribeiro, N., Mahamane, M., Patenaude, G., 2017. Charcoal income as a means to a valuable end: Scope and limitations of income from rural charcoal production to alleviate acute multidimensional poverty in Mabalane district, southern Mozambique. World Development Perspectives 7–8, 43–60. https://doi.org/10.1016/j.wdp.2017.11.005

Wackernagel, M., Hanscom, L., Jayasinghe, P., Lin, D., Murthy, A., Neill, E., Raven, P., 2021. The importance of resource security for poverty eradication. Nat Sustain 4, 731–738. https://doi.org/10.1038/s41893-021-00708-4

Walker, B., Kinzig, A., Langridge, J., 1999. Original Articles: Plant Attribute Diversity, Resilience, and Ecosystem Function: The Nature and Significance of Dominant and Minor Species. Ecosystems 2, 95–113. https://doi.org/10.1007/s100219900062

White, F., 1983. The vegetation of Africa., Natural Resources Research. UNESCO.

Whittaker, R.H., 1972. Evolution and Measurement of Species Diversity. TAXON 21, 213–251. https://doi.org/10.2307/1218190

Zorrilla-Miras, P., Mahamane, M., Metzger, M.J., Baumert, S., Vollmer, F., Luz, A.C., Woollen, E., Sitoe, A.A., Patenaude, G., Nhantumbo, I., Ryan, C.M., Paterson, J., Matediane, M.J., Ribeiro, N.S., Grundy, I.M., 2018. Environmental Conservation and Social Benefits of Charcoal Production in Mozambique. Ecological Economics 144, 100–111. https://doi.org/10.1016/j.ecolecon.2017.07.028

11 Appendix

Appendix Species list included in Trait PCA

Charcoal Species	Non-Charcoal Species
PCA top 5	
Alangium chinense, Albizia adianthifolia, Apodytes dimidiata, Balanites aegyptiaca, Bridelia cathartica, Cassia abbreviata, Celtis africana, Ceriops tagal, Combretum apiculatum, Combretum collinum, Combretum hereroense, Combretum imberbe, Combretum molle, Commiphora mollis, Dalbergia melanoxylon, Diospyros mespiliformis, Dodonaea viscosa, Dombeya rotundifolia, Ekebergia capensis, Englerophytum natalense, Erythrophleum africanum, Faurea saligna, Grewia monticola, Lannea schweinfurthii, Lumnitzera racemosa, Pappea capensis, Parinari excelsa, Pouteria alnifolia, Salvadora persica, Sideroxylon inerme, Spathodea campanulata, Spirostachys africana, Terminalia prunioides, Terminalia sericea, Trema orientalis, Ximenia americana	Acacia auriculiformis, Acacia crassicarpa, Acacia mangium, Acacia melanoxylon, Adansonia digitata, Albizia chinensis, Albizia gummifera, Annona squamosa, Antidesma venosum, Azadirachta indica, Bauhinia petersiana, Berchemia discolor, Brachystegia boehmii, Brachystegia spiciformis, Burkea africana, Cananga odorata, Capparis spinosa, Castilla elastica, Casuarina equisetifolia, Ceiba pentandra, Cinnamomum camphora, Coffea arabica, Dalbergia sissoo, Erica arborea, Eucalyptus camaldulensis, Eucalyptus globulus, Eucalyptus grandis, Eucalyptus saligna, Euclea divinorum, Euclea racemosa, Gliricidia sepium, Grevillea robusta, Hevea brasiliensis, Julbernardia globiflora, Kigelia africana, Lantana camara, Leucaena leucocephala, Mangifera indica, Pinus radiata, Plumeria rubra, Psidium guajava, Sclerocarya birrea, Syzygium cumini, Tamarindus indica, Terminalia catappa, Toona ciliata, Ziziphus mucronata
PCA Top 5 Robust	
Alangium chinense, Celtis africana, Ceriops tagal, Dalbergia melanoxylon, Dodonaea viscosa, Pouteria alnifolia, Sideroxylon inerme, Spathodea campanulata, Trema orientalis	Acacia auriculiformis, Annona squamosa, Antidesma venosum, Azadirachta indica, Castilla elastica, Ceiba pentandra, Cinnamomum camphora, Coffea arabica, Dalbergia sissoo, Eucalyptus globulus, Eucalyptus grandis, Eucalyptus saligna, Gliricidia sepium, Grevillea robusta, Kigelia africana, Leucaena leucocephala, Mangifera indica, Pinus radiata, Psidium guajava, Sclerocarya birrea, Syzygium cumini, Toona ciliata, Ziziphus mucronata

PCA Top 9	
Apodytes dimidiata, Balanites aegyptiaca, Dodonaea viscosa, Sideroxylon inerme, Trema orientalis	Acacia mangium, Acacia melanoxylon, Castilla elastica, Ceiba pentandra, Cinnamomum camphora, Coffea arabica, Euclea racemosa, Grevillea robusta, Lantana camara, Mangifera indica, Toona ciliata, Ziziphus mucronata
PCA Eco	
Alangium chinense, Apodytes dimidiata, Celtis africana, Ceriops tagal, Diospyros natalensis, Dodonaea viscosa, Englerophytum natalense, Pycnanthus angolensis, Sideroxylon inerme, Spathodea campanulata, Trema orientalis, Ximenia americana	Acacia mangium, Acacia melanoxylon, Azadirachta indica, Cananga odorata, Castilla elastica, Casuarina equisetifolia, Ceiba pentandra, Cinnamomum camphora, Coffea arabica, Erica arborea, Eucalyptus camaldulensis, Eucalyptus globulus, Eucalyptus grandis, Euclea racemosa, Gmelina arborea, Hevea brasiliensis, Lantana camara, Mangifera indica, Plumeria rubra, Tectona grandis, Toona ciliata, Ziziphus mucronata
PCA Boruta	
Alangium chinense, Albizia adianthifolia, Apodytes dimidiata, Balanites aegyptiaca, Bridelia cathartica, Celtis africana, Clausena anisata, Combretum apiculatum, Combretum collinum, Combretum hereroense, Combretum imberbe, Combretum molle, Commiphora mollis, Dalbergia melanoxylon, Diospyros natalensis, Dodonaea viscosa, Dombeya rotundifolia, Drypetes natalensis, Ekebergia	Acacia auriculiformis, Acacia crassicarpa, Acacia mangium, Acacia melanoxylon, Adansonia digitata, Albizia chinensis, Annona squamosa, Antidesma venosum, Azadirachta indica, Bauhinia petersiana, Berchemia discolor, Brachystegia boehmii, Brachystegia spiciformis, Burkea africana, Cananga odorata, Capparis spinosa, Castilla elastica, Casuarina equisetifolia, Ceiba pentandra, Cinnamomum camphora,

capensis, Englerophytum natalense, Erythrophleum africanum, Faurea saligna, Grewia monticola, Harungana madagascariensis, Ilex mitis, Kiggelaria africana, Lannea schweinfurthii, Pappea capensis, Parinari excelsa, Pouteria alnifolia, Sideroxylon inerme, Spathodea campanulata, Spirostachys africana, Terminalia prunioides, Terminalia sericea, Trema	Coffea arabica, Cussonia spicata, Dalbergia sissoo, Erica arborea, Eucalyptus camaldulensis, Eucalyptus globulus, Eucalyptus grandis, Euclea divinorum, Euclea racemosa, Ficus lutea, Gliricidia sepium, Grevillea robusta, Hevea brasiliensis, Julbernardia globiflora, Kigelia africana, Lantana camara, Leucaena leucocephala, Mangifera indica, Melia azedarach, Pinus radiata, Podocarpus latifolius, Prunus persica, Psidium guajava, Rapanea melanophloeos, Sclerocarya birrea, Syzygium cumini, Toona ciliata, Ziziphus mucronata

Appendix B: R and Python Code

The Structure of this section mirrors the setup of the method section first is the code for the trait analysis. Each run is always in its own box. This is followed by the python script (page 129) to download the satellite data. The SVA code is split between the National SVA Code (page 134) and then the Local SVA Code (pages 185).

TRY Trait space analysis

```
# ========================================
# 00_config.R — Project setup for TRY PCA
# Creates folder structure, global settings, and helpers.
# Source this file at the top of every pass:
#   source(file.path(PROJECT_DIR, "00_config.R"))
# ========================================
# ---- Repro options ----
options(stringsAsFactors = FALSE)
set.seed(123)
# ---- Base paths ----
DATA_ROOT   <- "C:/Users/david/OneDrive/Documents/Master ESS/Masterarbeit/Data"
PROJECT_DIR <- file.path(DATA_ROOT, "TRY PCA")
DIR_DATA_RAW     <- file.path(PROJECT_DIR, "data", "raw")
DIR_DATA_INTERIM <- file.path(PROJECT_DIR, "data", "interim")
DIR_DATA_DERIVED <- file.path(PROJECT_DIR, "data", "derived")
DIR_FIG          <- file.path(PROJECT_DIR, "fig")
DIR_LOGS         <- file.path(PROJECT_DIR, "logs")
DIR_SCRIPTS      <- file.path(PROJECT_DIR, "scripts")
# Charcoal species list (Excel)
SPECIES_CHARCOAL_XLSX <- file.path(
  DATA_ROOT, "Species List", "Species list.xlsx"
)
dirs_to_make <- c(PROJECT_DIR, DIR_DATA_RAW, DIR_DATA_INTERIM, DIR_DATA_DERIVED,
                  DIR_FIG, DIR_LOGS, DIR_SCRIPTS)
invisible(lapply(dirs_to_make, dir.create, recursive = TRUE, showWarnings = FALSE))
```

```
message("✅ Created/verified project folders under: ", PROJECT_DIR)
# ---- External raw sources (edit if needed) ----
# Keep pointers to your existing raw files outside this project structure.
RAW_TZ_SPECIES_XLSX <- file.path(DATA_ROOT, "ESS server", "Tree+Data__.xlsx")
# Example TRY Request 2 (TSV/CSV). Update to the actual file you use:
# In 00_config.R
RAW_TRY_REQUEST2 <- file.path(DATA_ROOT, "2. Try Data", "41327.txt")
# ---- ErrorRisk policy (consistent across passes) ----
error_risk_keep <- function(x) is.na(x) || x < 4  # keep NA or < 4
# Convenient dplyr helper
keep_errorrisk <- function(df) dplyr::filter(df, is.na(.data$ErrorRisk) | .data$ErrorRisk
< 4)
# ---- Charcoal labels (canonical) ----
# We normalize all variants to *two* labels and use this factor order globally.
CHARCOAL_LEVELS <- c("non-charcoal", "charcoal")
CHARCOAL_COLORS <- c("non-charcoal" = "#d95f02", "charcoal" = "#1b9e77")
normalize_charcoal <- function(x) {
  # Accepts yes/no/1/0/TRUE/FALSE/Y/N etc. Returns an ordered factor.
  lx <- tolower(as.character(x))
  out <- dplyr::case_when(
    lx %in% c("yes","y","1","true","t","charcoal") ~ "charcoal",
    lx  %in%  c("no","n","0","false","f","non-charcoal","noncharcoal","non  charcoal")  ~
"non-charcoal",
    TRUE ~ NA_character_
  )
  factor(out, levels = CHARCOAL_LEVELS, ordered = TRUE)
}
# ggplot helpers for consistent styling
scale_color_charcoal     <-     function(...)     ggplot2::scale_color_manual(values     =
CHARCOAL_COLORS, drop = FALSE, ...)
scale_fill_charcoal  <- function(...) ggplot2::scale_fill_manual(values  = CHARCOAL_COLORS,
drop = FALSE, ...)
theme_base <- function(base_size = 13) ggplot2::theme_minimal(base_size = base_size)
# ---- Trait naming maps (TRY long <-> short) ----
trait_rename_table <- tibble::tibble(
  original = c(
    "Leaf area per leaf dry mass (specific leaf area, SLA or 1/LMA): petiole excluded",
    "Leaf area per leaf dry mass (specific leaf area, SLA or 1/LMA): petiole included",
    "Leaf area per leaf dry mass (specific leaf area, SLA or 1/LMA): undefined if petiole
is in- or excluded",
    "Leaf nitrogen (N) content per leaf area",
    "Leaf nitrogen (N) content per leaf dry mass",
    "Leaf phosphorus (P) content per leaf area",
    "Leaf phosphorus (P) content per leaf dry mass",
    "Plant biomass and allometry: Leaf dry mass per plant dry mass (leaf weight ratio,
LWR)",
    "Seed dry mass",
    "Stem conduit density (vessels and tracheids)",
    "Stomata conductance per leaf dry mass",
    "Photosynthesis: intercellular CO2 concentration",
    "Stem conduit diameter (vessels, tracheids)",
    "Leaf transpiration rate per leaf area",
    "Crown (canopy) width",
    "Wood density (g/cm3)",
    "Stem diameter (cm)"
  ),
  short = c(
    "SLA_petiole_excluded",
    "SLA_petiole_included",
    "SLA_undefined",
    "Leaf_N_area",
    "Leaf_N_mass",
    "Leaf_P_area",
    "Leaf_P_mass",
    "LWR",
    "Seed_mass",
```

```
    "Stem_conduit_density",
    "Stomata_conductance",
    "CO2_intercellular",
    "Stem_conduit_diameter",
    "Transpiration_rate",
    "Crown_width",
    "Wood_density",
    "DBH"
  )
LONG_TO_SHORT <- stats::setNames(trait_rename_table$short, trait_rename_table$original)
SHORT_TO_LONG <- stats::setNames(trait_rename_table$original, trait_rename_table$short)

# ---- Canonical units for special traits ----
# Prefer TRY StdValue; fallback conversions only when StdValue is missing.
UNIT_CANONICAL <- list(
  wood_density = "g/cm3",
  dbh          = "cm"
)
# ---- Plot export helper ----
save_png <- function(plot_obj, filename, width = 12, height = 8, dpi = 300) {
  ggplot2::ggsave(
    filename = file.path(DIR_FIG, filename),
    plot = plot_obj, width = width, height = height, dpi = dpi
  )
  message("🖼️  Saved figure: ", file.path(DIR_FIG, filename))
}
# ---- CSV export helper ----
save_csv <- function(df, filename) {
  readr::write_csv(df, file.path(DIR_LOGS, filename))
  message("📄 Saved table: ", file.path(DIR_LOGS, filename))
}
# ---- Path builder (relative to project) ----
proj_path <- function(...) file.path(PROJECT_DIR, ...)
# ---- Quick summary to console ----
message("Project paths:")
message("  RAW:      ", DIR_DATA_RAW)
message("  INTERIM:  ", DIR_DATA_INTERIM)
message("  DERIVED:  ", DIR_DATA_DERIVED)
message("  FIG:      ", DIR_FIG)
message("  LOGS:     ", DIR_LOGS)
message("  SCRIPTS:  ", DIR_SCRIPTS)
message("Charcoal levels: ", paste(CHARCOAL_LEVELS, collapse = " < "))
message("ErrorRisk policy: keep NA or < 4")
```

```
# ========================================
# 01_ingest_filter_TZ.R — Ingest TRY R2 and filter to Tanzania
# Inputs:
#   - RAW_TRY_REQUEST2 (set in 00_config.R)  e.g., full_try_request_2.tsv/.csv
#   - RAW_TZ_SPECIES_XLSX (set in 00_config.R)  Tree+Data__.xlsx
# Outputs:
#   - data/interim/try_r2_tanzania.csv
#   - logs/trait_counts_request2.csv
#   - logs/dataset_counts_request2.csv
#   - logs/unmatched_species_request2.txt
# ========================================

suppressPackageStartupMessages({
  library(readr)
  library(readxl)
  library(dplyr)
  library(stringr)
})

# ---- wire to config ----
cfg   <-   file.path("C:/Users/david/OneDrive/Documents/Master   ESS/Masterarbeit/Data/TRY   PCA",
"00_config.R")
stopifnot(file.exists(cfg))
```

```
source(cfg)

# ---- sanity: raw sources exist ----
if (!file.exists(RAW_TRY_REQUEST2)) {
  stop("❌ RAW_TRY_REQUEST2 not found. Edit RAW_TRY_REQUEST2 in 00_config.R:\n", RAW_TRY_REQUEST2)
}
if (!file.exists(RAW_TZ_SPECIES_XLSX)) {
  stop("❌ RAW_TZ_SPECIES_XLSX not found. Edit RAW_TZ_SPECIES_XLSX in 00_config.R:\n",
RAW_TZ_SPECIES_XLSX)
}

# ---- helper: read TRY (TSV or CSV) ----
read_try_request2 <- function(path) {
  ext <- tolower(tools::file_ext(path))
  if (ext %in% c("tsv", "txt")) {
    readr::read_tsv(path, show_col_types = FALSE, progress = TRUE)
  } else if (ext %in% c("csv")) {
    readr::read_csv(path, show_col_types = FALSE, progress = TRUE)
  } else {
    message("Unknown extension '", ext, "'. Attempting TSV then CSV...")
    tryCatch(readr::read_tsv(path, show_col_types = FALSE),
             error = function(e) readr::read_csv(path, show_col_types = FALSE))
  }
}

# ---- 1) Load TRY request 2 ----
t0 <- Sys.time()
try_r2 <- read_try_request2(RAW_TRY_REQUEST2)
t1 <- Sys.time()
message("✅ TRY R2 loaded: ", paste(dim(try_r2), collapse = " x "),
        "  |  time: ", round(as.numeric(difftime(t1, t0, units = "secs")), 1), "s")

# clean species field minimally
if (!"AccSpeciesName" %in% names(try_r2)) {
  stop("❌ Column 'AccSpeciesName' not found in TRY R2.")
}
try_r2 <- try_r2 %>%
  mutate(AccSpeciesName = str_squish(AccSpeciesName))

# ---- 2) Load Tanzania species list ----
tz_xlsx <- readxl::read_excel(RAW_TZ_SPECIES_XLSX)

# try to find a species column robustly
cand_cols <- names(tz_xlsx)

pick <- cand_cols[
  stringr::str_detect(tolower(cand_cols), "species") &
    stringr::str_detect(tolower(cand_cols), "scientific|latin|name")
]

if (length(pick) == 0) {
  # fallback to known headers
  pick <- intersect(cand_cols, c("tree_species_scientific_name", "Latin name", "Latin_name"))
}
if (length(pick) == 0) {
  stop("Couldn't auto-detect the species column. Set `species_col` manually to the correct column
name.")
}

species_col <- pick[1]


tz_species <- tz_xlsx[[species_col]] %>%
  as.character() %>% str_trim() %>% str_squish()
tz_species <- tz_species[!is.na(tz_species) & nzchar(tz_species)]
tz_species <- unique(tz_species)

message("🌍 Tanzania species in Excel [", species_col, "]: ", length(tz_species))

# ---- 3) Filter TRY to Tanzanian species (exact match on AccSpeciesName) ----
try_tz <- try_r2 %>%
  filter(AccSpeciesName %in% tz_species)
```

```
message("📦 Rows kept after TZ filter: ", nrow(try_tz), " / ", nrow(try_r2))

# species coverage
matched_species <- sort(unique(try_tz$AccSpeciesName))
unmatched_species <- setdiff(tz_species, matched_species)
message("🔎 Species matched: ", length(matched_species), " | Unmatched: ",
length(unmatched_species))

# ---- 4) Summaries / logs ----
trait_counts <- try_tz %>%
  filter(!is.na(TraitName)) %>%
  count(TraitName, name = "n") %>%
  arrange(desc(n))

dataset_counts <- try_tz %>%
  count(DatasetID, name = "n") %>%
  arrange(desc(n))

save_csv(trait_counts, "trait_counts_request2.csv")
save_csv(dataset_counts, "dataset_counts_request2.csv")

# unmatched list (text file)
readr::write_lines(unmatched_species, file.path(DIR_LOGS, "unmatched_species_request2.txt"))
message("📝 Saved unmatched species list: ", file.path(DIR_LOGS, "unmatched_species_request2.txt"))

# ---- 5) Save filtered data ----
out_csv <- file.path(DIR_DATA_INTERIM, "try_r2_tanzania.csv")
readr::write_csv(try_tz, out_csv)
message("💾 Saved filtered TRY→TZ: ", out_csv)

# ---- console recap ----
message("\n== Recap ==")
message("TRY rows:          ", nrow(try_r2))
message("TRY→TZ rows:       ", nrow(try_tz))
message("Traits (non-NA):   ", nrow(trait_counts))
message("Datasets present:  ", nrow(dataset_counts))
message("Unmatched species: ", length(unmatched_species))
```

```
# ============================================
# 02_trait_QA_units.R — Trait QA & unit sanity
# Inputs:
#   - data/interim/try_r2_tanzania.rds  (preferred, if present)
#   - data/interim/try_r2_tanzania.csv  (fallback)
# Outputs:
#   - fig/traits_hist_stdvalue.pdf           (one page per trait)
#   - logs/units_by_trait.csv                (UnitName counts per trait)
#   - logs/trait_quick_stats.csv             (min/median/mean/max/sd/n)
# Notes:
#   Read-only diagnostics; uses StdValue (TRY standardized units).
# ============================================

suppressPackageStartupMessages({
  library(dplyr)
  library(readr)
  library(ggplot2)
  library(stringr)
})

# ---- wire to config ----
cfg   <-   file.path("C:/Users/david/OneDrive/Documents/Master   ESS/Masterarbeit/Data/TRY   PCA",
"00_config.R")
stopifnot(file.exists(cfg))
source(cfg)

message("🔎 Pass 2: Trait QA & units…")

# ---- load filtered TRY→TZ (prefer RDS for speed) ----
path_rds <- file.path(DIR_DATA_INTERIM, "try_r2_tanzania.rds")
path_csv <- file.path(DIR_DATA_INTERIM, "try_r2_tanzania.csv")

if (file.exists(path_rds)) {
  try_tz <- readr::read_rds(path_rds)
  message("✅ Loaded RDS: ", path_rds)
```

```
} else {
  # read only the columns we need for QA to keep it fast & light
  ct <- readr::cols(
    .default    = readr::col_skip(),
    TraitName   = readr::col_character(),
    StdValue    = readr::col_double(),
    UnitName    = readr::col_character(),
    DatasetID   = readr::col_character(),
    AccSpeciesName = readr::col_character()
  )
  try_tz <- readr::read_csv(path_csv, col_types = ct, show_col_types = FALSE, progress = TRUE)
  message("✅ Loaded CSV (skinny): ", path_csv)
}

# ---- quick summaries ----
trait_counts <- try_tz %>%
  filter(!is.na(TraitName)) %>%
  summarise(
    n_rows   = dplyr::n(),
    n_nonNA  = sum(!is.na(StdValue)),
    pct_nonNA = round(100 * n_nonNA / n_rows, 1),
    n_units  = dplyr::n_distinct(UnitName)
    , .by = TraitName) %>%
  arrange(desc(n_nonNA))

# units by trait
units_by_trait <- try_tz %>%
  filter(!is.na(TraitName), !is.na(UnitName)) %>%
  count(TraitName, UnitName, name = "n") %>%
  arrange(TraitName, desc(n))

# quick stats for StdValue
trait_stats <- try_tz %>%
  filter(!is.na(TraitName), !is.na(StdValue)) %>%
  summarise(
    n      = dplyr::n(),
    min    = suppressWarnings(min(StdValue, na.rm = TRUE)),
    q25    = suppressWarnings(quantile(StdValue, 0.25, na.rm = TRUE)),
    median = suppressWarnings(median(StdValue, na.rm = TRUE)),
    mean   = suppressWarnings(mean(StdValue, na.rm = TRUE)),
    q75    = suppressWarnings(quantile(StdValue, 0.75, na.rm = TRUE)),
    max    = suppressWarnings(max(StdValue, na.rm = TRUE)),
    sd     = suppressWarnings(sd(StdValue, na.rm = TRUE))
    , .by = TraitName) %>%
  arrange(desc(n))

# save logs
save_csv(units_by_trait,   "units_by_trait.csv")
save_csv(trait_counts,     "trait_counts_stdvalue.csv")
save_csv(trait_stats,      "trait_quick_stats.csv")

message("📄 Logged: units_by_trait.csv, trait_counts_stdvalue.csv, trait_quick_stats.csv")

# ---- histograms (StdValue) per trait: one page each ----
out_pdf <- file.path(DIR_FIG, "traits_hist_stdvalue.pdf")
grDevices::pdf(out_pdf, width = 8, height = 6)

# Plot in descending order of data availability
traits_ordered <- trait_counts$TraitName

for (tr in traits_ordered) {
  df <- try_tz %>% filter(TraitName == tr, !is.na(StdValue))
  if (nrow(df) == 0) next

  # collect units seen (top 5)
  units_str <- df %>%
    mutate(UnitName = ifelse(is.na(UnitName), "NA", UnitName)) %>%
    count(UnitName, sort = TRUE) %>%
    head(5) %>%
    mutate(label = paste0(UnitName, " (", n, ")")) %>%
    pull(label) %>%
    paste(collapse = "; ")

  p <- ggplot(df, aes(x = StdValue)) +
```

```
    geom_histogram(bins = 60, color = "black", fill = "grey70") +
    theme_base(13) +
    labs(
      title = paste0("Distribution of StdValue — ", tr),
      subtitle = paste0("n = ", nrow(df), "   |   Units (top): ", units_str),
      x = "StdValue (TRY standardized units)",
      y = "Count"
    )

  print(p)
}
grDevices::dev.off()
message("🖼️  Saved: ", out_pdf)

message("✅ Pass 2 complete.")
```

```
# ========================================
# 03_harmonize_key_traits.R — fill WD & DBH gaps with original units
# Inputs:
#   - data/interim/try_r2_tanzania.rds  (preferred) OR
#   - data/interim/try_r2_tanzania.csv
# Outputs:
#   - data/interim/try_r2_harmonized.rds  (fast to load)
#   - data/interim/try_r2_harmonized.csv
#   - logs/harmonization_report.csv  (before/after non-NA counts for WD/DBH)
# Notes:
#   * Prefer StdValue; only fallback to OrigValueStr+OrigUnitStr when StdValue is NA.
#   * Canonical units: wood density = g/cm3; DBH = cm
# ========================================

suppressPackageStartupMessages({
  library(dplyr); library(readr); library(stringr); library(tidyr)
})

# ---- wire to config ----
cfg    <-    file.path("C:/Users/david/OneDrive/Documents/Master    ESS/Masterarbeit/Data/TRY    PCA",
"00_config.R")
stopifnot(file.exists(cfg)); source(cfg)

message("🪄 Pass 3: Harmonize wood density & DBH (fallback fill only)…")

# ---- load filtered data ----
path_rds <- file.path(DIR_DATA_INTERIM, "try_r2_tanzania.rds")
path_csv <- file.path(DIR_DATA_INTERIM, "try_r2_tanzania.csv")

if (file.exists(path_rds)) {
  dt <- readr::read_rds(path_rds)
  message("✅ Loaded RDS: ", path_rds)
} else {
  # read only needed columns to keep it light
  ct <- cols(
    .default      = col_skip(),
    AccSpeciesName = col_character(),
    SpeciesName    = col_character(),
    TraitName      = col_character(),
    StdValue       = col_double(),
    UnitName       = col_character(),
    OrigUnitStr    = col_character(),
    OrigValueStr   = col_character(),
    ErrorRisk      = col_double(),
    DatasetID      = col_character(),
    Dataset        = col_character()
  )
  dt <- readr::read_csv(path_csv, col_types = ct, show_col_types = FALSE, progress = TRUE)
  message("✅ Loaded CSV (skinny): ", path_csv)
}

# ---- constants (TRY long names) ----
TRAIT_WD  <- "Stem specific density (SSD, stem dry mass per stem fresh volume) or wood density"
TRAIT_DBH <- "Stem diameter"

# ---- unit & number helpers (UTF-8 safe) ----
normalize_unit <- function(x) {
```

```
  x <- iconv(x, from = "", to = "UTF-8", sub = "")
  x <- stringr::str_to_lower(stringr::str_trim(x))
  x <- stringr::str_replace_all(x, "[[:space:]]+", "")
  x <- stringr::str_replace_all(x, "[·•\u00B7]", "")   # middots
  x <- stringr::str_replace_all(x, "per", "/")
  # normalize exponents/minus signs around -3
  x <- stringr::str_replace_all(x, "cm\\^?\\s*-?3|cm[---]?3", "cm-3")
  x <- stringr::str_replace_all(x, "m\\^?\\s*-?3|m[---]?3",   "m-3")
  x
}
parse_num_safe <- function(x) {
  x <- iconv(x, from = "", to = "UTF-8", sub = "")
  suppressWarnings(readr::parse_number(x))
}

TRAIT_WD  <- "Stem specific density (SSD, stem dry mass per stem fresh volume) or wood density"
TRAIT_DBH <- "Stem diameter"

# ---- harmonize in ONE pass (no row-index joins) ----
dt3 <- dt %>%
  mutate(
    UnitName_u     = normalize_unit(UnitName),
    OrigUnitStr_u  = normalize_unit(OrigUnitStr),
    OrigValue_num  = parse_num_safe(OrigValueStr),

    # Wood density: prefer StdValue; if NA, convert original to g/cm3
    WD_from_orig = dplyr::case_when(
      TraitName == TRAIT_WD & is.na(StdValue) &
        OrigUnitStr_u %in% c(
          "g/cm3","g/cm^3","g/cm³","g/cm-3","gcm-3","gcm^3","gcm3",
          "g/ml","g/ml","gperml","gml-1","gml^-1",
          "t/m3","t/m-3","tm-3","t/m^3",
          "kg/dm3","kg/dm-3","kgperdm3","kg/l","kgl-1",
          "mg/mm3","mg/mm-3","mgpermm3"  # 1 mg/mm3 = 1 g/cm3
        ) ~ OrigValue_num,
      TraitName == TRAIT_WD & is.na(StdValue) &
        OrigUnitStr_u %in% c("kg/m3","kg/m-3","kg/m^3","kgm-3","kgperm3") ~ OrigValue_num / 1000,
      TRUE ~ NA_real_
    ),
    WD_g_cm3_filled = dplyr::if_else(TraitName == TRAIT_WD,
                                     dplyr::coalesce(StdValue, WD_from_orig),
                                     NA_real_),

    # DBH: use StdValue+UnitName if present; else convert original to cm
    DBH_from_std = dplyr::case_when(
      TraitName == TRAIT_DBH & !is.na(StdValue) & UnitName_u == "m"  ~ StdValue * 100,
      TraitName == TRAIT_DBH & !is.na(StdValue) & UnitName_u == "cm" ~ StdValue,
      TraitName == TRAIT_DBH & !is.na(StdValue) & UnitName_u == "mm" ~ StdValue / 10,
      TRUE ~ NA_real_
    ),
    DBH_from_orig = dplyr::case_when(
      TraitName == TRAIT_DBH & is.na(DBH_from_std) & OrigUnitStr_u == "m"  ~ OrigValue_num * 100,
      TraitName == TRAIT_DBH & is.na(DBH_from_std) & OrigUnitStr_u == "cm" ~ OrigValue_num,
      TraitName == TRAIT_DBH & is.na(DBH_from_std) & OrigUnitStr_u == "mm" ~ OrigValue_num / 10,
      TRUE ~ NA_real_
    ),
    DBH_cm_filled = dplyr::if_else(TraitName == TRAIT_DBH,
                                   dplyr::coalesce(DBH_from_std, DBH_from_orig, StdValue),
                                   NA_real_),

    # Final canonical label + filled value
    HarmonizedTraitName = dplyr::case_when(
      TraitName == TRAIT_WD  ~ "Wood density (g/cm3)",
      TraitName == TRAIT_DBH ~ "Stem diameter (cm)",
      TRUE ~ TraitName
    ),
    StdValue_filled = dplyr::case_when(
      TraitName == TRAIT_WD  ~ dplyr::coalesce(WD_g_cm3_filled, StdValue),
      TraitName == TRAIT_DBH ~ dplyr::coalesce(DBH_cm_filled, StdValue),
      TRUE ~ StdValue
    )
  )

# ---- compact before/after report for Methods ----
```

```
# WD
wd_before <- with(dt,  sum(!is.na(TraitName) & TraitName == TRAIT_WD  & !is.na(StdValue)))
wd_after  <- with(dt3, sum(!is.na(TraitName) & TraitName == TRAIT_WD  & !is.na(StdValue_filled)))

# DBH
dbh_before <- with(dt,  sum(!is.na(TraitName) & TraitName == TRAIT_DBH & !is.na(StdValue)))
dbh_after  <- with(dt3, sum(!is.na(TraitName) & TraitName == TRAIT_DBH & !is.na(StdValue_filled)))

report2 <- tibble::tibble(
  Trait         = c("Wood density (g/cm3)", "Stem diameter (cm)"),
  NonNA_before  = c(wd_before, dbh_before),
  NonNA_after   = c(wd_after,  dbh_after),
  Filled_added  = NonNA_after - NonNA_before
)

save_csv(report2, "harmonization_report.csv")
report2


# ---- save outputs ----
out_rds <- file.path(DIR_DATA_INTERIM, "try_r2_harmonized.rds")
out_csv <- file.path(DIR_DATA_INTERIM, "try_r2_harmonized.csv")
readr::write_rds(dt3, out_rds)
readr::write_csv(dt3, out_csv)

message("💾 Saved harmonized data:\n  ", out_rds, "\n  ", out_csv)
message("📄 Report: ", file.path(DIR_LOGS, "harmonization_report.csv"))
message("✅ Pass 3 complete.")
```

```
# ========================================
# 04_charcoal_tagging.R — species-level charcoal tagging
# Strategy: Latin list (strict) + Latin binomials extracted from Vernacular name(s)
# Inputs:
#   - data/interim/try_r2_harmonized.(rds|csv)
#   - SPECIES_CHARCOAL_XLSX ("Latin name" + "Vernacular name(s)")
# Outputs:
#   - data/interim/try_r2_tagged.(rds|csv)
#   - logs/charcoal_obs_counts.csv
#   - logs/charcoal_species_counts.csv
#   - logs/charcoal_added_via_vernacular.csv (what the vernacular step added)
# ========================================

suppressPackageStartupMessages({
  library(dplyr); library(readr); library(readxl); library(stringr); library(tidyr); library(purrr)
})

# ---- wire to config ----
cfg    <-    file.path("C:/Users/david/OneDrive/Documents/Master    ESS/Masterarbeit/Data/TRY    PCA",
"00_config.R")
stopifnot(file.exists(cfg)); source(cfg)

message("🔥 Pass 4: Charcoal tagging (Latin + vernacular-embedded Latin)")

# ---- load harmonized data ----
path_rds <- file.path(DIR_DATA_INTERIM, "try_r2_harmonized.rds")
path_csv <- file.path(DIR_DATA_INTERIM, "try_r2_harmonized.csv")

if (file.exists(path_rds)) {
  dt <- readr::read_rds(path_rds)
  message("✅ Loaded RDS: ", path_rds)
} else {
  ct <- cols(
    .default             = col_skip(),
    AccSpeciesName       = col_character(),
    SpeciesName          = col_character(),
    TraitName            = col_character(),
    HarmonizedTraitName  = col_character(),
    StdValue             = col_double(),
    StdValue_filled      = col_double(),
    UnitName             = col_character(),
    ErrorRisk            = col_double()
  )
  dt <- readr::read_csv(path_csv, col_types = ct, show_col_types = FALSE, progress = TRUE)
```

```
  message("✅ Loaded CSV (skinny): ", path_csv)
}

# helpers
norm_name <- function(x) {
  x <- iconv(x, from = "", to = "UTF-8", sub = "")
  x <- str_squish(x)
  x <- str_replace_all(x, "\\s+", " ")
  str_to_lower(x)
}
is_binomial <- function(x) {
  # TRUE for strings like "Genus species" (capitalized genus, lower-case species, optional hyphens)
  str_detect(x, "^[A-Z][a-z-]+ [a-z][a-z-]+$")
}

# ---- read charcoal list (Excel) ----
if (!exists("SPECIES_CHARCOAL_XLSX")) stop("Set SPECIES_CHARCOAL_XLSX in 00_config.R")
stopifnot(file.exists(SPECIES_CHARCOAL_XLSX))

sl <- readxl::read_excel(SPECIES_CHARCOAL_XLSX)

# detect columns
cand <- names(sl)
latin_col <- cand[str_detect(str_to_lower(cand), "latin") & str_detect(str_to_lower(cand), "name")]
if (length(latin_col) == 0) latin_col <- intersect(cand, c("Latin name","Latin_name","LatinName"))
if (length(latin_col) == 0) stop("Couldn't find a Latin-name column in the Excel file.")
latin_col <- latin_col[1]

vern_col  <-  cand[str_detect(str_to_lower(cand),  "vernacular")  &  str_detect(str_to_lower(cand),
"name")]
if     (length(vern_col)     ==     0)     vern_col     <-     intersect(cand,     c("Vernacular
name(s)","Vernacular_name(s)","Vernacular"))
vern_col <- if (length(vern_col)) vern_col[1] else NA_character_

# ---- strict Latin list from Latin-column ----
lat_vec <- sl[[latin_col]] %>%
  as.character() %>% dplyr::coalesce("") %>% stringr::str_squish()
lat_vec <- unique(lat_vec[nzchar(lat_vec)])
lat_key <- norm_name(lat_vec)

# ---- broaden search: Latin binomials embedded in Vernacular/Notes ----
latin_regex <- "([A-Z][a-z-]+\\s+[a-z][a-z-]+)"  # Genus species

cols_to_scan <- intersect(names(sl), c(vern_col, "Notes"))
extra_text <- sl %>%
  dplyr::select(dplyr::any_of(cols_to_scan)) %>%
  dplyr::mutate(dplyr::across(dplyr::everything(), as.character)) %>%
  unlist(use.names = FALSE)

extra_text <- extra_text[!is.na(extra_text) & nzchar(extra_text)]

# extract *any* Latin binomial anywhere in the text
matches <- stringr::str_match_all(extra_text, latin_regex)
vern_binomials <- unique(unlist(lapply(matches, function(m) m[, 2])))

# drop "Genus sp." and similar (second word literally "sp" or "sp.")
vern_binomials <- vern_binomials[!grepl(" [sS][pP]\\.?$", vern_binomials)]

vern_key <- norm_name(vern_binomials)

message("📘 Latin names in Excel (Latin col): ", length(lat_key))
message("📗 Latin binomials found in Vernacular/Notes: ", length(vern_key))

# ---- union (Latin + vernacular-binomial), restricted to TRY-TZ species ----
try_species <- unique(dt$AccSpeciesName)
try_key <- norm_name(try_species)

all_lat_keys  <- unique(c(lat_key, vern_key))
all_lat_keys  <- intersect(all_lat_keys, try_key)  # only those present in TRY-TZ

latin_only_set <- intersect(lat_key, try_key)
vern_only_added_keys <- setdiff(all_lat_keys, latin_only_set)

# map keys back to canonical AccSpeciesName for logging
```

```
key_lookup <- tibble::tibble(
  key = try_key,
  AccSpeciesName = try_species
)
added_tbl <- key_lookup %>%
  dplyr::filter(key %in% vern_only_added_keys) %>%
  dplyr::select(AccSpeciesName)

# species-level lookup (consistent tagging)
species_lookup <- tibble(AccSpeciesName = unique(dt$AccSpeciesName)) %>%
  mutate(.key = norm_name(AccSpeciesName)) %>%
  mutate(charcoal = if_else(.key %in% all_lat_keys, "charcoal", "non-charcoal")) %>%
  select(-.key)

# ---- enforce consistent labels and save ----
dt_tagged <- dt %>%
  left_join(species_lookup, by = "AccSpeciesName") %>%
  mutate(charcoal = factor(charcoal, levels = c("non-charcoal","charcoal")))

# logs
obs_counts     <- dt_tagged %>% count(charcoal, name = "n_observations")
species_counts <- species_lookup %>% count(charcoal, name = "n_species")

save_csv(obs_counts,     "charcoal_obs_counts.csv")
save_csv(species_counts, "charcoal_species_counts.csv")
save_csv(added_tbl,      "charcoal_added_via_vernacular.csv")

message("📄           Logged:          charcoal_obs_counts.csv,          charcoal_species_counts.csv,
charcoal_added_via_vernacular.csv")

# save
out_rds <- file.path(DIR_DATA_INTERIM, "try_r2_tagged.rds")
out_csv <- file.path(DIR_DATA_INTERIM, "try_r2_tagged.csv")
readr::write_rds(dt_tagged, out_rds)
readr::write_csv(dt_tagged, out_csv)

message("💾 Saved tagged data:\n  ", out_rds, "\n  ", out_csv)
message("✅ Pass 4 complete.")
```

```
# ========================================
# 05_traits_to_matrix.R — species × trait means + PCA inputs
# Inputs:
#   - data/interim/try_r2_tagged.(rds|csv)
# Outputs:
#   - data/derived/species_trait_means_wide.(rds|csv)
#   - data/derived/pca_input_top5.(rds|csv)
#   - data/derived/pca_input_top9.(rds|csv)
#   - logs/trait_long_to_short_map.csv
#   - logs/trait_completeness_counts.csv
#   - logs/coverage_curve.csv
#   - fig/coverage_curve.png
# Notes:
#   Uses StdValue_filled if present else StdValue; filters ErrorRisk >= 4.
# ========================================

suppressPackageStartupMessages({
  library(dplyr); library(tidyr); library(readr); library(stringr); library(ggplot2)
})

# ---- wire to config ----
cfg    <-    file.path("C:/Users/david/OneDrive/Documents/Master    ESS/Masterarbeit/Data/TRY    PCA",
"00_config.R")
stopifnot(file.exists(cfg)); source(cfg)

message("🧱 Pass 5: species × trait means + PCA inputs")

# ---- load tagged data ----
path_rds <- file.path(DIR_DATA_INTERIM, "try_r2_tagged.rds")
path_csv <- file.path(DIR_DATA_INTERIM, "try_r2_tagged.csv")

if (file.exists(path_rds)) {
  dt <- readr::read_rds(path_rds)
```

```
  message("✅ Loaded RDS: ", path_rds)
} else {
  ct <- cols(
    .default        = col_skip(),
    AccSpeciesName  = col_character(),
    charcoal        = col_character(),
    TraitName       = col_character(),
    HarmonizedTraitName = col_character(),
    StdValue        = col_double(),
    StdValue_filled = col_double(),
    ErrorRisk       = col_double()
  )
  dt <- readr::read_csv(path_csv, col_types = ct, show_col_types = FALSE, progress = TRUE)
  message("✅ Loaded CSV (skinny): ", path_csv)
}

# ---- apply ErrorRisk policy & choose numeric value ----
dt <- dt %>%
  dplyr::filter(is.na(ErrorRisk) | ErrorRisk < 4) %>%
  dplyr::mutate(Value = dplyr::coalesce(StdValue_filled, StdValue))

# ---- trait labeling: prefer harmonized name when present ----
dt <- dt %>%
  mutate(TraitLabel = dplyr::coalesce(HarmonizedTraitName, TraitName))

# ---- long -> short trait names (for plots) ----
trait_rename_tbl <- tibble::tibble(
  original = c(
    "Leaf area per leaf dry mass (specific leaf area, SLA or 1/LMA): petiole excluded",
    "Leaf area per leaf dry mass (specific leaf area, SLA or 1/LMA): petiole included",
    "Leaf area per leaf dry mass (specific leaf area, SLA or 1/LMA): undefined if petiole is in- or
excluded",
    "Leaf nitrogen (N) content per leaf area",
    "Leaf nitrogen (N) content per leaf dry mass",
    "Leaf phosphorus (P) content per leaf area",
    "Leaf phosphorus (P) content per leaf dry mass",
    "Plant biomass and allometry: Leaf dry mass per plant dry mass (leaf weight ratio, LWR)",
    "Seed dry mass",
    "Stem conduit density (vessels and tracheids)",
    "Stomata conductance per leaf dry mass",
    "Photosynthesis: intercellular CO2 concentration",
    "Stem conduit diameter (vessels, tracheids)",
    "Leaf transpiration rate per leaf area",
    "Crown (canopy) width",
    "Wood density (g/cm3)",
    "Stem diameter (cm)"
  ),
  short = c(
    "SLA_petiole_excluded",
    "SLA_petiole_included",
    "SLA_undefined",
    "Leaf_N_area",
    "Leaf_N_mass",
    "Leaf_P_area",
    "Leaf_P_mass",
    "LWR",
    "Seed_mass",
    "Stem_conduit_density",
    "Stomata_conductance",
    "CO2_intercellular",
    "Stem_conduit_diameter",
    "Transpiration_rate",
    "Crown_width",
    "Wood density",
    "DBH"
  )
)

# Save the mapping (handy for Methods & plotting labels)
save_csv(trait_rename_tbl, "trait_long_to_short_map.csv")

# ---- species × trait means (wide) ----
species_trait_means <- dt %>%
  dplyr::filter(!is.na(Value), !is.na(TraitLabel)) %>%
```

```
  dplyr::group_by(AccSpeciesName, charcoal, TraitLabel) %>%
  dplyr::summarise(mean_value = mean(Value, na.rm = TRUE), .groups = "drop")

species_trait_wide <- species_trait_means %>%
  tidyr::pivot_wider(names_from = TraitLabel, values_from = mean_value)

# Save full wide matrix
out_wide_rds <- file.path(DIR_DATA_DERIVED, "species_trait_means_wide.rds")
out_wide_csv <- file.path(DIR_DATA_DERIVED, "species_trait_means_wide.csv")
readr::write_rds(species_trait_wide, out_wide_rds)
readr::write_csv(species_trait_wide, out_wide_csv)
message("💾 Saved wide matrix:\n  ", out_wide_rds, "\n  ", out_wide_csv)

# ---- completeness by trait (how many species have a mean) ----
comp_by_trait <- species_trait_wide %>%
  dplyr::select(-AccSpeciesName, -charcoal) %>%
  summarise(dplyr::across(dplyr::everything(), ~ sum(!is.na(.)))) %>%
  tidyr::pivot_longer(dplyr::everything(), names_to = "TraitLabel", values_to = "NonNA_Species") %>%
  dplyr::arrange(dplyr::desc(NonNA_Species))

save_csv(comp_by_trait, "trait_completeness_counts.csv")

# ---- coverage curve: species remaining as you add top-N traits ----
traits_ranked <- comp_by_trait$TraitLabel
coverage_curve <- lapply(seq_along(traits_ranked), function(k) {
  keep_traits <- traits_ranked[1:k]
  n_complete <- species_trait_wide %>%
    dplyr::select(AccSpeciesName, charcoal, dplyr::all_of(keep_traits)) %>%
    tidyr::drop_na() %>%
    nrow()
  tibble::tibble(n_traits = k, complete_species = n_complete)
}) %>% dplyr::bind_rows()

save_csv(coverage_curve, "coverage_curve.csv")

# Plot coverage curve
p_cov <- ggplot(coverage_curve, aes(x = n_traits, y = complete_species)) +
  geom_line() + geom_point() +
  theme_minimal(base_size = 12) +
  labs(title = "Species coverage vs. number of traits (top → down)",
       x = "Number of traits included",
       y = "Species with complete data")

ggsave(file.path(DIR_FIG, "coverage_curve.png"), p_cov, width = 7, height = 5, dpi = 300)
message("🖼️  Saved: ", file.path(DIR_FIG, "coverage_curve.png"))

# ---- make PCA-ready subsets: Top 5 and Top 9 traits by completeness ----
top5_traits <- head(traits_ranked, 5)
top9_traits <- head(traits_ranked, 9)

# Rename columns to short names (for PCA labels)
rename_vec <- setNames(trait_rename_tbl$short, trait_rename_tbl$original)

rename_cols <- function(df) {
  cn <- names(df)
  for (i in seq_along(cn)) {
    if (!cn[i] %in% c("AccSpeciesName","charcoal") && cn[i] %in% names(rename_vec)) {
      cn[i] <- rename_vec[cn[i]]
    }
  }
  names(df) <- cn
  df
}

pca_input_top5 <- species_trait_wide %>%
  dplyr::select(AccSpeciesName, charcoal, dplyr::all_of(top5_traits)) %>%
  tidyr::drop_na() %>%
  rename_cols()

pca_input_top9 <- species_trait_wide %>%
  dplyr::select(AccSpeciesName, charcoal, dplyr::all_of(top9_traits)) %>%
  tidyr::drop_na() %>%
  rename_cols()
```

```
# Save PCA-ready matrices
readr::write_rds(pca_input_top5, file.path(DIR_DATA_DERIVED, "pca_input_top5.rds"))
readr::write_csv(pca_input_top5, file.path(DIR_DATA_DERIVED, "pca_input_top5.csv"))
readr::write_rds(pca_input_top9, file.path(DIR_DATA_DERIVED, "pca_input_top9.rds"))
readr::write_csv(pca_input_top9, file.path(DIR_DATA_DERIVED, "pca_input_top9.csv"))

message("💾 Saved PCA inputs (top5/top9) to data/derived/")
message("✅ Pass 5 complete.")
```

```
# ========================================
# 06_pca_and_permanova_styled.R
# Reproduce "final" PCA aesthetics for 3 panels:
#   1) Top-5 (from pca_input_top5.rds to keep sample identical)
#   2) Boruta-selected set
#   3) Ecologically important set
# Inputs:
#   - data/derived/pca_input_top5.rds
#   - data/derived/species_trait_means_wide.rds
# Outputs:
#   - fig/PCA_top5_sidepanel.png
#   - fig/PCA_boruta5_sidepanel.png
#   - fig/PCA_eco_sidepanel.png
#   - logs/pca_*_permanova.csv, logs/pca_*_loadings.csv
# ========================================

suppressPackageStartupMessages({
  library(tidyverse)
  library(ggrepel)
  library(cowplot)
  library(gridExtra)
  library(vegan)
  library(scales)
  library(grid)   # for unit()
  })

# ---- constants to match your style ----
set.seed(123)
AXIS_RANGE <- c(-4, 4)
TICK_BY    <- 1
COLORS     <- c("non-charcoal" = "#d95f02", "charcoal" = "#1b9e77")

# ---- wire to config ----
cfg    <-    file.path("C:/Users/david/OneDrive/Documents/Master    ESS/Masterarbeit/Data/TRY    PCA",
"00_config.R")
stopifnot(file.exists(cfg)); source(cfg)

# ---- load inputs from the new pipeline ----
top5_path <- file.path(DIR_DATA_DERIVED, "pca_input_top5.rds")
wide_path <- file.path(DIR_DATA_DERIVED, "species_trait_means_wide.rds")
stopifnot(file.exists(top5_path), file.exists(wide_path))

df_top5 <- readr::read_rds(top5_path)
wide    <- readr::read_rds(wide_path)

# ensure group as factor with canonical order & labels
fix_group <- function(df, col = "charcoal") {
  df %>%
    mutate(!!col := factor(.data[[col]], levels = c("non-charcoal", "charcoal")))
}

df_top5 <- fix_group(df_top5)
wide    <- fix_group(wide)

# short -> long (for counting observations per species/trait from the long table)
trait_rename_table <- tibble::tibble(
  original = c(
    "Leaf area per leaf dry mass (specific leaf area, SLA or 1/LMA): petiole excluded",
    "Leaf area per leaf dry mass (specific leaf area, SLA or 1/LMA): petiole included",
    "Leaf area per leaf dry mass (specific leaf area, SLA or 1/LMA): undefined if petiole is in- or
excluded",
    "Leaf nitrogen (N) content per leaf area",
    "Leaf nitrogen (N) content per leaf dry mass",
    "Leaf phosphorus (P) content per leaf area",
    "Leaf phosphorus (P) content per leaf dry mass",
```

```
    "Plant biomass and allometry: Leaf dry mass per plant dry mass (leaf weight ratio, LWR)",
    "Seed dry mass",
    "Stem conduit density (vessels and tracheids)",
    "Stomata conductance per leaf dry mass",
    "Photosynthesis: intercellular CO2 concentration",
    "Stem conduit diameter (vessels, tracheids)",
    "Leaf transpiration rate per leaf area",
    "Crown (canopy) width",
    "Wood density (g/cm3)",
    "Stem diameter (cm)"
  ),
  short = c(
    "SLA_petiole_excluded",
    "SLA_petiole_included",
    "SLA_undefined",
    "Leaf_N_area",
    "Leaf_N_mass",
    "Leaf_P_area",
    "Leaf_P_mass",
    "LWR",
    "Seed_mass",
    "Stem_conduit_density",
    "Stomata_conductance",
    "CO2_intercellular",
    "Stem_conduit_diameter",
    "Transpiration_rate",
    "Crown_width",
    "Wood density",
    "DBH"
  )
)
short_to_long <- setNames(trait_rename_table$original, trait_rename_table$short)

# also create the inverse map (long -> short)
long_to_short <- setNames(trait_rename_table$short, trait_rename_table$original)

# rename columns in 'wide' to the short names where we have a match
nm <- names(wide)
map <- long_to_short[nm]
names(wide) <- ifelse(!is.na(map), map, nm)

# robust selector based on *observation counts* in the long tagged table
robust_species <- function(traits_short, min_per_trait = 3, min_avg = 3) {
  dt_long <- readr::read_rds(file.path(DIR_DATA_INTERIM, "try_r2_tagged.rds"))
  long_names <- unname(short_to_long[traits_short])

  obs <- dt_long %>%
    dplyr::filter(HarmonizedTraitName %in% long_names, !is.na(StdValue_filled)) %>%
    dplyr::count(AccSpeciesName, HarmonizedTraitName, name = "n") %>%
    tidyr::pivot_wider(names_from = HarmonizedTraitName, values_from = n, values_fill = 0)

  mat <- obs %>% dplyr::select(all_of(long_names))
  min_cnt <- do.call(pmin, c(as.data.frame(mat), list(na.rm = TRUE)))
  avg_cnt <- rowMeans(mat, na.rm = TRUE)

  obs$AccSpeciesName[min_cnt >= min_per_trait & avg_cnt >= min_avg]
}

# ---- trait sets (short names; must match columns in 'wide') ----
top5_traits        <- c("Seed_mass","Wood density","Leaf_N_mass","Leaf_P_mass","SLA_undefined")
boruta_traits_short <- c("Leaf_N_area","Seed_mass","SLA_undefined","Leaf_N_mass","Wood density")
eco_traits         <- c("Wood density","DBH","SLA_undefined","Leaf_N_mass","Stem_conduit_density")

# helper: keep only present numeric trait columns and drop NAs (complete cases)
# helper: keep only present *numeric* trait columns with >0 variance; drop NAs (complete cases)
mk_pca_input <- function(df, trait_names) {
  present <- intersect(trait_names, names(df))
  if (!length(present)) stop("None of the requested traits exist in the data: ",
                             paste(trait_names, collapse = ", "))

  X <- df %>% select(AccSpeciesName, charcoal, all_of(present))

  # keep only numeric present traits
  num_present <- present[vapply(X[present], is.numeric, logical(1))]
```

```
  if (!length(num_present)) stop("No numeric traits among: ", paste(present, collapse = ", "))

  # drop traits with zero variance
  keep <- vapply(X[num_present], function(v) sd(v, na.rm = TRUE) > 0, logical(1))
  num_present <- num_present[keep]
  if (!length(num_present)) stop("All requested traits have zero variance after filtering.")

  X %>% select(AccSpeciesName, charcoal, all_of(num_present)) %>% drop_na()
}

# For Top-5: use the *derived* pca_input_top5 to reproduce the exact sample
pca_input_top5 <- df_top5
pca_input_boru <- mk_pca_input(wide, boruta_traits_short)
pca_input_eco  <- mk_pca_input(wide, eco_traits)

# ================================
# Plotting helpers (match your final look)
# ================================
run_trait_pca <- function(df, trait_cols, group_col = "charcoal") {
  stopifnot(all(c("AccSpeciesName", group_col) %in% names(df)))
  mat <- df %>% select(all_of(trait_cols)) %>% as.matrix()
  Z   <- scale(mat)  # center + scale
  pca <- prcomp(Z, center = FALSE, scale. = FALSE)
  list(
    pca = pca,
    scores = as_tibble(pca$x[,1:2,drop=FALSE]) %>%
      mutate(Species = df$AccSpeciesName, Group = df[[group_col]]),
    loadings = as_tibble(pca$rotation[,1:2,drop=FALSE], rownames = "Trait"),
    imp = summary(pca)$importance
  )
}

relabel_charcoal_legend <- function(p) {
  p +
    scale_color_manual(values = COLORS, breaks = names(COLORS),
                       labels = c("Non-charcoal","Charcoal")) +try
    scale_fill_manual(values = COLORS, breaks = names(COLORS),
                      labels = c("Non-charcoal","Charcoal")) +
    guides(color = guide_legend(title = NULL), fill = guide_legend(title = NULL))
}

fix_pca_axes <- function(p, axis_range = AXIS_RANGE, tick_by = TICK_BY) {
  breaks <- seq(axis_range[1], axis_range[2], by = tick_by)
  p +
    coord_fixed(xlim = axis_range, ylim = axis_range, expand = FALSE, clip = "on") +
    scale_x_continuous(breaks = breaks, limits = axis_range, expand = c(0, 0)) +
    scale_y_continuous(breaks = breaks, limits = axis_range, expand = c(0, 0)) +
    theme(
      legend.position = "bottom",
      legend.title = element_blank(),
      legend.box = "horizontal",
      plot.margin = margin(10, 10, 12, 10)
    )
}

plot_corr_circle <- function(loadings) {
  L <- loadings %>% select(Trait, PC1, PC2)
  ggplot(L, aes(PC1, PC2, label = Trait)) +
    annotate("path",
             x = cos(seq(0, 2*pi, length.out = 200)),
             y = sin(seq(0, 2*pi, length.out = 200)),
             linetype = "dashed", linewidth = 0.3) +
    geom_segment(aes(x = 0, y = 0, xend = PC1, yend = PC2),
                 arrow = arrow(length = unit(0.012, "npc")), linewidth = 0.3) +
    ggrepel::geom_text_repel(size = 2.7, max.overlaps = 50, segment.size = 0.2) +
    coord_equal(xlim = c(-1, 1), ylim = c(-1, 1), expand = TRUE) +
    theme_minimal(base_size = 9) +
    theme(axis.title = element_text(size = 8),
          axis.text  = element_text(size = 7)) +
    labs(x = "PC1 loadings", y = "PC2 loadings")
}

scree_df <- function(imp, k = 6) {
  tibble(
```

```
    PC         = paste0("PC", seq_len(ncol(imp))),
    Proportion = scales::percent(as.numeric(imp[2, ]), accuracy = 1),
    Cumulative = scales::percent(as.numeric(imp[3, ]), accuracy = 1)
  ) %>% slice(1:min(k, n()))
}
scree_grob <- function(imp, k = 6) {
  gridExtra::tableGrob(
    scree_df(imp, k),
    rows = NULL,
    theme = gridExtra::ttheme_minimal(
      core    = list(fg_params = list(cex = 0.75, fontface = 1)),
      colhead = list(fg_params = list(cex = 0.8,  fontface = 2))
    )
  )
}

run_permanova <- function(df, trait_cols, group_col = "charcoal", permutations = 9999) {
  dat <- df %>% select(all_of(group_col), all_of(trait_cols)) %>% drop_na()
  if (nrow(dat) < 3) stop("Not enough rows for PERMANOVA.")
  keep <- sapply(dat[names(dat) %in% trait_cols], sd, na.rm = TRUE) > 0
  trait_cols <- trait_cols[keep]
  Z <- scale(as.matrix(dat[, trait_cols, drop = FALSE]))
  d <- dist(Z, method = "euclidean")
  grp <- data.frame(Group = dat[[group_col]])
  set.seed(123)
  ad <- adonis2(d ~ Group, data = grp, permutations = permutations, by = "margin")
  bd <- betadisper(d, grp$Group)
  bd_anova <- anova(bd, permutations = permutations)
  list(
    N = nrow(dat),
    n_non = sum(dat[[group_col]] == "non-charcoal"),
    n_yes = sum(dat[[group_col]] == "charcoal"),
    F  = as.numeric(ad$F[1]),
    R2 = as.numeric(ad$R2[1]),
    p  = as.numeric(ad$`Pr(>F)`[1]),
    disp_p = as.numeric(bd_anova$`Pr(>F)`[1])
  )
}

compose_figure <- function(pca_obj, df_for_perma, trait_cols, title_suffix, out_stub, fig_title =
NULL) {
  imp <- pca_obj$imp
  expl <- percent(imp[2, 1:2], accuracy = 0.1)
  scores <- pca_obj$scores

  cents <- scores %>% group_by(Group) %>%
    summarise(PC1 = mean(PC1), PC2 = mean(PC2), .groups = "drop")
  seg_layer <- if (nrow(cents) == 2) annotate("segment",
                                              x = cents$PC1[1], y = cents$PC2[1],
                                              xend = cents$PC1[2], yend = cents$PC2[2],
                                              linetype = "dashed", color = "black") else NULL
  cross_layer <- if (nrow(cents) == 2) geom_point(data = cents, aes(PC1, PC2),
                                                  inherit.aes = FALSE, shape = 4, size = 5, stroke =
2, color = "black") else NULL

  p_scatter <- ggplot(scores, aes(PC1, PC2, color = Group)) +
    geom_point(size = 3, alpha = 0.85) +
    stat_ellipse(geom = "polygon", aes(fill = Group), alpha = 0.20, show.legend = FALSE) +
    stat_ellipse(geom = "path") +
    cross_layer + seg_layer +
    scale_color_manual(values = COLORS) +
    scale_fill_manual(values = COLORS) +
    theme_minimal() +
    theme(plot.title = element_text(size = 14, face = "bold"),
          legend.title = element_blank()) +
    labs(
      title = paste("PCA Trait Space", title_suffix),
      x = paste0("PC1 (", expl[1], ")"),
      y = paste0("PC2 (", expl[2], ")")
    ) %>%
    relabel_charcoal_legend() %>%
    fix_pca_axes(axis_range = AXIS_RANGE, tick_by = TICK_BY)

  p_circle <- plot_corr_circle(pca_obj$loadings)
```

```
  tab      <- scree_grob(imp, k = 6)

  perma <- run_permanova(df_for_perma, trait_cols)
  perma_lab <- sprintf(
    paste0("PERMANOVA (Euclidean on z-scored traits)\n",
           "F = %.3f, R² = %.3f, p = %.3f;  Dispersion p = %.3f\n",
           "N = %d  |  Non-charcoal = %d,  Charcoal = %d"),
    perma$F, perma$R2, perma$p, perma$disp_p, perma$N, perma$n_non, perma$n_yes
  )
  p_label <- ggplot() +
    annotate("label", x = 0.5, y = 0.5, label = perma_lab,
             hjust = 0.5, vjust = 0.5, size = 12/3.5,
             label.size = 0, fill = "white", alpha = 0.95, colour = "black") +
    coord_cartesian(xlim = c(0,1), ylim = c(0,1), expand = FALSE) +
    theme_void()

  right <- plot_grid(
    p_circle,
    ggdraw() + draw_plot(tab, x = 0, y = 0, width = 1, height = 1),
    p_label,
    ncol = 1, rel_heights = c(0.55, 0.30, 0.15)
  )

  g <- plot_grid(p_scatter, right, ncol = 2, rel_widths = c(0.67, 0.33), align = "h")

  # 🔹 Optional figure-level title spanning the whole composition
  if (!is.null(fig_title)) {
    g <- plot_grid(
      ggdraw() + draw_label(fig_title, fontface = "bold", size = 16, hjust = 0.5, vjust = 0.5),
      g, ncol = 1, rel_heights = c(0.08, 0.92)
    )
  }

  # logs
  loadings_out <- pca_obj$loadings %>% arrange(desc(abs(PC1)))
  readr::write_csv(loadings_out, file.path(DIR_LOGS, paste0("pca_", out_stub, "_loadings.csv")))
  readr::write_csv(
    tibble::tibble(N = perma$N, n_non_charcoal = perma$n_non, n_charcoal = perma$n_yes,
                   F = perma$F, R2 = perma$R2, p_value = perma$p, disp_p_value = perma$disp_p,
                   pc1_var = imp[2,1], pc2_var = imp[2,2], cumulative_PC2 = imp[3,2]),
    file.path(DIR_LOGS, paste0("pca_", out_stub, "_permanova.csv"))
  )

  # figs
  ggsave(file.path(DIR_FIG, paste0("PCA_", out_stub, "_sidepanel.png")),
         g, width = 12, height = 8, dpi = 300)
  message("🖼️  Saved: fig/PCA_", out_stub, "_sidepanel.png")
  g
}

# Top-5
compose_figure(p_top5, pca_input_top5, traits_top5_present,
               "(Top 5 Traits)", "top5",
               fig_title = "PCA — Top 5 Traits")

# Top-5 — Robust
compose_figure(p_top5_rob, pca_input_top5_robust, traits_top5_present,
               "(Top 5 — Robust Species Only)", "top5_robust",
               fig_title = "PCA — Top 5 Traits (Robust)")

# Boruta-selected
compose_figure(p_boru, pca_input_boru, traits_boru_present,
               "(Boruta-selected Traits)", "boruta5",
               fig_title = "PCA — Boruta-selected Traits")

# Ecologically important
compose_figure(p_eco, pca_input_eco, traits_eco_present,
               "(Ecologically Important Traits)", "eco",
               fig_title = "PCA — Ecologically Important Traits")

# Top-9 (if present)
compose_figure(p_top9, df_top9, traits_top9_present,
               "(Top 9 Traits)", "top9",
               fig_title = "PCA — Top 9 Traits")
```

```
message("✅ Pass 6 complete.")
```

```
# ========================================
# 07_boruta_charcoal.R — Boruta in TRY PCA workflow
# Inputs:  data/derived/species_trait_means_wide.rds
# Outputs: fig/Boruta_importance.png
#          logs/boruta_attStats.csv
#          logs/boruta_selected_traits.csv
# ========================================

suppressPackageStartupMessages({
  library(tidyverse)
  library(Boruta)
})

# ---- wire to config ----
cfg    <-    file.path("C:/Users/david/OneDrive/Documents/Master    ESS/Masterarbeit/Data/TRY    PCA",
"00_config.R")
stopifnot(file.exists(cfg)); source(cfg)

# ---- load the wide species × trait means from Pass 5 ----
wide <- readr::read_rds(file.path(DIR_DATA_DERIVED, "species_trait_means_wide.rds"))

# keep only numeric trait columns
trait_cols <- setdiff(names(wide), c("AccSpeciesName","charcoal"))
# minimum #traits per species (same rule you used before)
min_valid_traits <- 6

df_ml <- wide %>%
  mutate(charcoal = factor(charcoal, levels = c("non-charcoal","charcoal"))) %>%
  filter(rowSums(!is.na(across(all_of(trait_cols)))) >= min_valid_traits) %>%
  select(Charcoal = charcoal, all_of(trait_cols))

message("✅ Boruta input: ", nrow(df_ml), " species × ", ncol(df_ml)-1, " traits")

# ---- run Boruta ----
set.seed(123)
boruta_result <- Boruta(Charcoal ~ ., data = df_ml, doTrace = 2)

# quick list of selected features
sel_all <- Boruta::getSelectedAttributes(boruta_result, withTentative = TRUE)
message("🤌 Selected (incl. tentative): ", paste(sel_all, collapse = ", "))

# ---- pretty base-R plot (labels readable, nothing cropped) ----
# order by median importance so highest are on the right
med_imp <- apply(boruta_result$ImpHistory, 2, function(x) stats::median(x, na.rm = TRUE))
ord     <- order(med_imp, decreasing = FALSE)
boru_plot <- boruta_result
boru_plot$ImpHistory <- boru_plot$ImpHistory[, ord, drop = FALSE]

# compute bottom margin from wrapped names (but most names are already short)
wrap_width <- 18
lbl <- vapply(colnames(boru_plot$ImpHistory),
              function(s) paste(strwrap(s, width = wrap_width), collapse = "\n"),
              character(1))
colnames(boru_plot$ImpHistory) <- lbl
max_lines  <- max(vapply(strsplit(lbl, "\n", fixed = TRUE), length, integer(1)))
bottom_mar <- max(10, 6 + 1.1 * max_lines)

# save into TRY PCA/fig
out_png <- file.path(DIR_FIG, "Boruta_importance.png")
png(out_png, width = 2800, height = 1600, res = 300)
op <- par(mar = c(bottom_mar, 5, 4, 2) + 0.1, xpd = NA)
plot(boru_plot, las = 2, cex.axis = 0.9, xlab = "", ylab = "",
     main = "Boruta feature importance for Charcoal", boxwex = 0.6)
legend("topleft", inset = 0.01, bty = "n",
       fill = c("green3","gold","red3","dodgerblue3"),
       border = NA,
       legend = c("Confirmed","Tentative","Rejected","Shadow"))
par(op); dev.off()
message("🖼️  Saved: ", out_png)

# ---- save stats to logs (robust to Boruta version) ----
```

```
att <- Boruta::attStats(boruta_result) %>% as.data.frame()
if (!"medianImp" %in% names(att)) {
  med <- apply(boruta_result$ImpHistory, 2, function(x) stats::median(x, na.rm = TRUE))
  att$medianImp <- med[match(rownames(att), names(med))]
}
att_tbl <- att %>%
  tibble::rownames_to_column("Trait") %>%
  arrange(desc(medianImp))

readr::write_csv(att_tbl, file.path(DIR_LOGS, "boruta_attStats.csv"))
readr::write_csv(tibble(Selected = sel_all),
                 file.path(DIR_LOGS, "boruta_selected_traits.csv"))

message("✅ Pass 08 complete.")
```

Python code for automized download

```
# --- Cell 0: Environment check (run once) ---

import sys, subprocess, pkgutil

print("Python:", sys.version)
print("Platform OK\n")

required = [
    "cdsetool",     # Copernicus Data Space helper
    "tqdm",
    "pandas",
    "geopandas",
    "shapely",
    "requests"
]

def ensure(pkg):
    if pkgutil.find_loader(pkg) is None:
        print(f"➡️ installing {pkg} ...")
        subprocess.check_call([sys.executable, "-m", "pip", "install", "-q", pkg])
    else:
        print(f"✅ {pkg} already installed")

for p in required:
    ensure(p)

# show cdsetool version
try:
    import cdsetool, importlib.metadata as im
    print("\ncdsetool version:", im.version("cdsetool"))
except Exception as e:
    print("\n(cdsetool version check) ->", e)

print("\n✅ Cell 0 done.")

# --- Cell 1: Imports & constants ---

import os
from datetime import date, timedelta

import requests
import pandas as pd
import geopandas as gpd
from shapely.geometry import shape, Polygon
from tqdm import tqdm

# ✅ Where to save downloads (Jupyter environment)
download_dir = "/home/jovyan/cds_downloads"
os.makedirs(download_dir, exist_ok=True)
print("Download dir:", download_dir)

# ✅ Date window (December 2024 through January 2025)
start_date = date(2024, 12, 1)
end_date   = date(2025, 1, 31)
```

```
# ✅ Cloud threshold (percent)
max_cloud = 20.0   # we can tweak later per tile if needed

# ✅ AOI bbox (covers western–eastern TZ, including MGRS zone 37)
# (xmin, ymin, xmax, ymax) ~ (29.33, -11.75, 41.80, -0.98)
aoi_bbox = (29.63833, -11.69167, 40.39583, -2.01000)

# Build WKT POLYGON for OData/RESTO-style queries if needed
xmin, ymin, xmax, ymax = aoi_bbox
bbox_poly = Polygon([(xmin, ymin), (xmin, ymax), (xmax, ymax), (xmax, ymin), (xmin, ymin)])
wkt = f"POLYGON(({xmin} {ymin}, {xmin} {ymax}, {xmax} {ymax}, {xmax} {ymin}, {xmin} {ymin}))"

print("Dates:", start_date, "→", end_date)
print("Cloud % max:", max_cloud)
print("AOI WKT:", wkt)
print("✅ Cell 1 ready.")
# --- Cell 2: Find candidate tiles in AOI/date window ---

import re
import math

xmin, ymin, xmax, ymax = aoi_bbox
box = f"{xmin},{ymin},{xmax},{ymax}"

def fetch_features_in_aoi(start_date, end_date, box, max_pages=50, page_size=200):
    """
    Query CDSE RESTO SENTINEL-2 for L2A products intersecting the bbox
    within the date window. Returns a flat list of GeoJSON features.
    """
    base = "https://catalogue.dataspace.copernicus.eu/resto/api/collections/SENTINEL-2/search.json"
    features = []

    for page in range(1, max_pages + 1):
        params = {
            "startDate": start_date.isoformat(),
            "completionDate": end_date.isoformat(),
            "box": box,
            "processingLevel": "S2MSI2A",
            "productType": "S2MSI2A",
            "maxRecords": str(page_size),
            "page": str(page),
            # You can add cloud filter here if you want to prefilter:
            # "cloudCover": f"[0,{int(max_cloud)}]"
        }
        r = requests.get(base, params=params, timeout=60)
        r.raise_for_status()
        js = r.json()

        page_feats = js.get("features", [])
        if not page_feats:
            break

        features.extend(page_feats)

        # stop early if we reached the total results
        total = js.get("properties", {}).get("totalResults")
        if total is not None:
            if page * page_size >= int(total):
                break

    return features

print("🔎 Querying catalogue…")
features = fetch_features_in_aoi(start_date, end_date, box)
print(f"Found {len(features)} features in AOI/date window.")

# Extract unique tile IDs from product titles
tile_re = re.compile(r"_T([0-9]{2}[A-Z]{3})_")
tiles = set()

for f in features:
    title = f.get("properties", {}).get("title", "")
    m = tile_re.search(title)
```

```
    if m:
        tiles.add(m.group(1))

tiles = sorted(tiles)
tiles_37 = [t for t in tiles if t.startswith("37")]

print(f"🧭 Unique MGRS tiles: {len(tiles)}")
print("Sample:", tiles[:10])
print(f"✅ Tiles in zone 37 (should include eastern TZ): {len(tiles_37)} →", tiles_37[:10])

# Keep for later steps
candidate_tiles = tiles
# --- Cell 3 (fixed): Pick the best (lowest-cloud) scene per tile, then filter already-downloaded --
-

import re

# Tiles you already have
already_downloaded = {
    '35MRS','36MUU','35LRL','36MTV','36MUA','36MVD','36MUD','36MTU',
    '35MRP','35MRN','36MVC','35MRQ','36LTN','36MVB','36MUS','36MUE',
    '36MUV','36MUT','36MTA','36LUN','35LQK','36LUM','35LRK','36LTP',
    '36LTM','35LRH','35LQH','36LUP','35LRG','35LRJ','35LQG','36MVE',
    '36MTD','35LQJ','36MTB','36LUR','36MWC','35MQN','35MRT','36MVU',
    '36MWA','36MVS','36MWD','36MVV','36MVT','36MWB','36LTR','36MUC',
    '35MQM','36MUB','36MTS','35MRR','36MWE','36MTE','36MVA','36MTC',
    '36MTT','36LTQ','35LQL','36LUQ','35MRM','35MQQ','35MQP','35MQS',
    '35MQU','35MQR','35MQT','35MRV','35MQV','35MRU'
}

# Safety checks: we need `features` and `max_cloud` from Cell 2/1
assert 'features' in globals(), "Run Cell 2 first to define `features`."
assert 'max_cloud' in globals(), "Run Cell 1 first to define `max_cloud`."

tile_re = re.compile(r"_T([0-9]{2}[A-Z]{3})_")

# 1) Best scene per tile (lowest cloud within window)
best_by_tile = {}  # tile -> (url, cloud, date, title)
for f in features:
    props = f.get("properties", {})
    title = props.get("title", "")
    m = tile_re.search(title)
    if not m:
        continue
    tile = m.group(1)
    cloud = props.get("cloudCover", 100.0) or 100.0
    if cloud > max_cloud:
        continue
    date_str = props.get("startDate")
    url = props.get("services", {}).get("download", {}).get("url")
    if not url:
        continue

    # keep the lower-cloud one (or earlier date on tie)
    if tile not in best_by_tile:
        best_by_tile[tile] = (url, cloud, date_str, title)
    else:
        u0, c0, d0, t0 = best_by_tile[tile]
        if (cloud < c0) or (cloud == c0 and (date_str or "") < (d0 or "")):
            best_by_tile[tile] = (url, cloud, date_str, title)

# 2) Build list, then filter already-downloaded
download_list = [
    (tile, info[0], info[1])  # (tile, url, cloud)
    for tile, info in best_by_tile.items()
    if tile not in already_downloaded
]

# sort for stable output
download_list.sort(key=lambda x: x[0])

print(f"After filtering, {len(download_list)} tiles remain to download.")
print("Sample:", download_list[:5])
```

```
tile_ids = [tile for tile, _, _ in download_list]
print("[" + ", ".join(f"'{t}'" for t in tile_ids) + "]")


from cdsetool.credentials import Credentials, validate_credentials

# 👇 Explicitly pass your credentials
creds = Credentials(
    username="david.wick@uzh.ch",
    password="RossoRosso_17"
)

# 🔐 Validate login
validate_credentials(username="david.wick@uzh.ch", password="RossoRosso_17")

print("✅ Authenticated successfully!")



from cdsetool.query import query_features
from cdsetool.download import download_features
from cdsetool.credentials import Credentials, validate_credentials
from datetime import date
from tqdm.notebook import tqdm
import os

# 🔐 Authenticate
#creds = Credentials()
#validate_credentials()

# 📁 Ensure download directory exists
download_path = "/home/jovyan/cds_downloads"
os.makedirs(download_path, exist_ok=True)

# 📍 Tiles to process
tile_ids = ['36LVN', '36LVP', '36LVQ', '36LVR', '36LWN', '36LWP', '36LWQ', '36LWR', '36LXN', '36LXP',
'36LXQ', '36LXR', '36LYP', '36LYQ', '36LYR', '36LZN', '36LZP', '36LZQ', '36LZR', '36MWS', '36MWT',
'36MWV', '36MXA', '36MXB', '36MXC', '36MXS', '36MXT', '36MXU', '36MXV', '36MYA', '36MYB', '36MYC',
'36MYS', '36MYT', '36MYU', '36MYV', '36MZA', '36MZB', '36MZC', '36MZS', '36MZT', '36MZU', '36MZV',
'37LBH', '37LBJ', '37LBK', '37LBL', '37LCJ', '37LDH', '37LDJ', '37LDK', '37LEH', '37LEJ', '37LEK',
'37LEL', '37LFH', '37LFJ', '37LFK', '37LFL', '37MBM', '37MBN', '37MBP', '37MBQ', '37MBS', '37MBT',
'37MCN', '37MCP', '37MCQ', '37MCR', '37MCS', '37MCT', '37MDP', '37MDQ', '37MDR', '37MDS', '37MDT',
'37MEM', '37MEN', '37MEP', '37MEQ', '37MER', '37MES', '37MET', '37MFM', '37MFN', '37MFP', '37MFQ',
'37MFR', '37MFS', '37MFT']

# 🗓️ Date range
start_date = date(2024, 12, 1)
end_date = date(2025, 1, 31)

# ☁️ Cloud settings
cloud_threshold = 20
cloud_range_str = "[0,20]"


# Save location
download_path = "/home/jovyan/cds_downloads"  # Change if needed

# Track downloaded UUIDs
downloaded = []

# Loop with progress bar
for tile in tqdm(tile_ids, desc="📦 Downloading tiles"):
    print(f"\n🔍 Searching for first valid product in tile {tile}...")
    try:
        features = query_features(
            "Sentinel2",
            {
                "startDate": start_date,
                "completionDate": end_date,
                "tileId": tile,
                "productType": "S2MSI2A",
                "processingLevel": "S2MSI2A",
                "cloudCover": cloud_range_str,
```

```
            }
        )

        features_list = list(features)
        if not features_list:
            print(f"❌ No features found for {tile}")
            continue

        # Sort chronologically
        sorted_features = sorted(
            features_list,
            key=lambda f: f["properties"]["startDate"]
        )

        # Find first ONLINE feature with cloudCover ≤ threshold
        best = next(
            (f for f in sorted_features
             if f["properties"].get("status") == "ONLINE"
             and f["properties"].get("cloudCover", 100) <= cloud_threshold),
            None
        )

        if not best:
            print(f"⚠️ No ONLINE features with <{cloud_threshold}% cloud for {tile}")
            continue

        print(f"✅          Found:          {best['properties']['title']}          |          Cloud:
{best['properties']['cloudCover']:.2f}%")

        # Download it
        for f_id in download_features([best], download_path, {"credentials": creds}):
            print(f"⬇️ Downloaded: {f_id}")
            downloaded.append(f_id)

    except Exception as e:
        print(f"❌ Error for tile {tile}: {e}")

print(f"\n Finished: {len(downloaded)} product(s) downloaded.")


import zipfile
import os

zip_path = "/home/jovyan/cds_batch2.zip"
dir_to_zip = "/home/jovyan/cds_downloads"

with zipfile.ZipFile(zip_path, 'w', zipfile.ZIP_DEFLATED) as zipf:
    for root, dirs, files in os.walk(dir_to_zip):
        for file in files:
            file_path = os.path.join(root, file)
            arcname = os.path.relpath(file_path, start=dir_to_zip)
            zipf.write(file_path, arcname)

print(f"✅ Zipped to: {zip_path}")
```

National SVA Code

```
# Load necessary packages
library(sf)        # spatial data handling
library(dplyr)     # data manipulation
library(ggplot2)   # plotting

# ──────────────────────────────────────────────────────
# 1. Set working directory & read shapefiles
# ──────────────────────────────────────────────────────

shp_dir <- "C:/Users/david/OneDrive/Documents/Master ESS/Masterarbeit/Data/Shapefiles miombo extend
in protected areas"

# File paths
```

```
miombo_path <- file.path(shp_dir, "pnv_vecea_v2_0_Tanzania.shp")
protected_path <- file.path(shp_dir, "protected_clean_final.shp")

# Read shapefiles
miombo <- st_read(miombo_path)
protected <- st_read(protected_path)

# ──────────────────────────────────────────────────────────
# 2. Filter to core Miombo types only (Wmd = Drier, Wmw = Wetter)
# ──────────────────────────────────────────────────────────

miombo_core <- miombo %>%
  filter(CODE %in% c("Wmd", "Wmw"))

# ──────────────────────────────────────────────────────────
# 3. Harmonize coordinate systems and validate geometries
# ──────────────────────────────────────────────────────────

miombo_core <- st_transform(miombo_core, crs = st_crs(protected))
miombo_core <- st_make_valid(miombo_core)
protected <- st_make_valid(protected)

# ──────────────────────────────────────────────────────────
# 4. Intersect Miombo with protected areas → assign protection
# ──────────────────────────────────────────────────────────

# Split intersected polygons with protection info
miombo_protected <- st_intersection(miombo_core, protected) %>%
  select(CODE, LABEL, protection)

# Identify unprotected Miombo (non-overlapping parts)
miombo_unprotected <- st_difference(miombo_core, st_union(protected)) %>%
  filter(!st_is_empty(geometry)) %>%
  st_make_valid() %>%
  mutate(protection = "Unprotected")

# Combine both into a single dataset
miombo_classified <- bind_rows(miombo_protected, miombo_unprotected)

# ──────────────────────────────────────────────────────────
# 5. Visualization
# ──────────────────────────────────────────────────────────

ggplot(miombo_classified) +
  geom_sf(aes(fill = protection), color = NA, alpha = 0.7) +
  scale_fill_manual(
    values = c(
      "Strictly Protected" = "#1a9850",
      "Uncertain Protection" = "#fee08b",
      "Unprotected" = "#f46d43"
    )
  ) +
  labs(
    title = "Miombo Woodlands Classified by Protection Status",
    subtitle = "Drier and Wetter Miombo intersected with Protected Areas",
    caption = "Source: VECEA & WDPA"
  ) +
  theme_minimal()

# ──────────────────────────────────────────────────────────
# 6. Export the final classified shapefile
# ──────────────────────────────────────────────────────────

st_write(
  miombo_classified,
  dsn = file.path(shp_dir, "miombo_classified_protection.shp"),
  delete_layer = TRUE
)
```

```
## Pass 1 valid pixel per tile

# ---- Libraries ----
suppressPackageStartupMessages({
```

```
  library(sf)
  library(terra)
  library(data.table)
  library(fs)
  library(stringr)
})

# ---- CONFIG ----
root_dir    <- "C:/Users/david/OneDrive/Documents/Master ESS/Masterarbeit/Data/Sentinel 2
Bulk"
miombo_path <- "C:/Users/david/OneDrive/Documents/Master ESS/Masterarbeit/Data/Shapefiles
miombo extend in protected areas/miombo_classified_protection.shp"

# SCL classes to mask out (invalid)
SCL_BAD  <- c(0,1,3,8,9,10,11)
# Eligible protection classes (others will be excluded)
ELIGIBLE_PROT <- c("Strictly Protected", "Unprotected")

OUT_DIR       <-    file.path(root_dir,    sprintf("_SVA_RESULTS_%s",    format(Sys.time(),
"%Y%m%d_%H%M%S")))
dir_create(OUT_DIR, recurse = TRUE)
set.seed(42)

# ---- Helpers ----
list_safe_tiles <- function(root) {
  d <- list.dirs(root, full.names = TRUE, recursive = FALSE)
  d[grepl("\\.SAFE$", d)]
}
tile_code_from_path <- function(p) sub(".*_T([0-9A-Z]{5})_.*", "\\1", basename(p))

find_tile_files <- function(tile_dir) {
  all_20m <- list.files(tile_dir, pattern = "_20m\\.jp2$", recursive = TRUE, full.names =
TRUE)
  if (!length(all_20m)) return(NULL)
  scl <- grep("SCL_20m\\.jp2$", all_20m, value = TRUE)
  if (!length(scl)) return(NULL)
  list(scl = scl[1])
}

# ---- Load Miombo (once) ----
miombo_sf <- st_read(miombo_path, quiet = TRUE) |>
  subset(!st_is_empty(geometry)) |>
  st_make_valid()

# Normalize & filter protection column
if (!"protection" %in% names(miombo_sf)) {
  nms <- tolower(names(miombo_sf))
  cand <- which(nms %in% c("protection","protect","prot_status","status"))
  if (length(cand)) names(miombo_sf)[cand[1]] <- "protection"
}
# Keep only eligible classes (drop "Uncertain" and others)
miombo_sf <- miombo_sf |>
  mutate(protection = as.character(protection)) |>
  subset(!is.na(protection) & protection %in% ELIGIBLE_PROT)

# ---- PASS 1: Count valid pixels per tile (Miombo ∩ eligible protection ∩ SCL good) ----
tiles <- list_safe_tiles(root_dir)
stopifnot(length(tiles) > 0)

pb <- txtProgressBar(min = 0, max = length(tiles), initial = 0, style = 3)
rows <- vector("list", length(tiles))

for (i in seq_along(tiles)) {
  tile_dir  <- tiles[[i]]
  tile_code <- tile_code_from_path(tile_dir)
  files     <- find_tile_files(tile_dir)
```

```
  if (is.null(files)) {
    rows[[i]] <- data.frame(
      tile_code = tile_code,
      tile_path = tile_dir,
      miombo_px_total = NA_integer_,
      miombo_px_protected = NA_integer_,
      miombo_px_unprotected = NA_integer_,
      valid_px_total  = NA_integer_,
      valid_px_protected = NA_integer_,
      valid_px_unprotected = NA_integer_,
      status    = "missing_files"
    )
    setTxtProgressBar(pb, i); next
  }

  # read SCL only for counting
  r_scl <- try(terra::rast(files$scl), silent = TRUE)
  if (inherits(r_scl, "try-error")) {
    rows[[i]] <- data.frame(
      tile_code = tile_code,
      tile_path = tile_dir,
      miombo_px_total = NA_integer_,
      miombo_px_protected = NA_integer_,
      miombo_px_unprotected = NA_integer_,
      valid_px_total  = NA_integer_,
      valid_px_protected = NA_integer_,
      valid_px_unprotected = NA_integer_,
      status    = "scl_read_error"
    )
    setTxtProgressBar(pb, i); next
  }

  # reproject Miombo (already filtered to eligible classes) to SCL CRS
  miombo_t  <-  try(st_transform(miombo_sf,  crs(terra::crs(r_scl,  proj=TRUE))),  silent  =
TRUE)
  if (inherits(miombo_t, "try-error")) {
    rows[[i]] <- data.frame(
      tile_code = tile_code,
      tile_path = tile_dir,
      miombo_px_total     =     NA_integer_,     miombo_px_protected     =     NA_integer_,
miombo_px_unprotected = NA_integer_,
      valid_px_total  = NA_integer_, valid_px_protected = NA_integer_, valid_px_unprotected
= NA_integer_,
      status    = "miombo_transform_error"
    )
    setTxtProgressBar(pb, i); next
  }

  # crop & mask SCL to eligible Miombo polygons
  r_scl_c <- try(terra::crop(r_scl, terra::vect(miombo_t)), silent = TRUE)
  if (inherits(r_scl_c, "try-error")) {
    rows[[i]] <- data.frame(
      tile_code = tile_code,
      tile_path = tile_dir,
      miombo_px_total     =     NA_integer_,     miombo_px_protected     =     NA_integer_,
miombo_px_unprotected = NA_integer_,
      valid_px_total  = NA_integer_, valid_px_protected = NA_integer_, valid_px_unprotected
= NA_integer_,
      status    = "scl_crop_error"
    )
    setTxtProgressBar(pb, i); next
  }
  r_scl_m <- try(terra::mask(r_scl_c, terra::vect(miombo_t)), silent = TRUE)
  if (inherits(r_scl_m, "try-error")) {
    rows[[i]] <- data.frame(
```

```
      tile_code = tile_code,
      tile_path = tile_dir,
      miombo_px_total     =     NA_integer_,     miombo_px_protected     =     NA_integer_,
miombo_px_unprotected = NA_integer_,
      valid_px_total  = NA_integer_, valid_px_protected = NA_integer_, valid_px_unprotected
= NA_integer_,
      status    = "scl_mask_error"
    )
    setTxtProgressBar(pb, i); next
  }

  # total Miombo pixels in eligible classes (any SCL)
  miombo_px_total <- try(terra::global(!is.na(r_scl_m), "sum", na.rm=TRUE)[1,1], silent =
TRUE)
  if (inherits(miombo_px_total, "try-error")) miombo_px_total <- NA_integer_

  # Apply SCL validity (bad classes -> NA, others -> 1)
  rcl <- cbind(SCL_BAD, NA)
  r_valid <- try(terra::classify(r_scl_m, rcl = rcl, others = 1), silent = TRUE)
  if (inherits(r_valid, "try-error")) {
    rows[[i]] <- data.frame(
      tile_code = tile_code,
      tile_path = tile_dir,
      miombo_px_total  =  as.integer(miombo_px_total),  miombo_px_protected  =  NA_integer_,
miombo_px_unprotected = NA_integer_,
      valid_px_total  = NA_integer_, valid_px_protected = NA_integer_, valid_px_unprotected
= NA_integer_,
      status    = "classify_error"
    )
    setTxtProgressBar(pb, i); next
  }

  valid_px_total <- try(terra::global(r_valid, "sum", na.rm=TRUE)[1,1], silent = TRUE)
  if (inherits(valid_px_total, "try-error")) valid_px_total <- NA_integer_

  # Per-group breakdown (may be empty on one side; treat as 0)
  miombo_px_protected <- miombo_px_unprotected <- 0L
  valid_px_protected  <- valid_px_unprotected  <- 0L

  m_prot <- subset(miombo_t, protection == "Strictly Protected")
  if (nrow(m_prot) > 0) {
    rp <- try(terra::mask(r_scl_c, terra::vect(m_prot)), silent = TRUE)
    vp <- try(terra::mask(r_valid,  terra::vect(m_prot)), silent = TRUE)
    if (!inherits(rp, "try-error")) {
      miombo_px_protected     <-     as.integer(try(terra::global(!is.na(rp),     "sum",
na.rm=TRUE)[1,1], silent=TRUE))
      if (is.na(miombo_px_protected)) miombo_px_protected <- 0L
    }
    if (!inherits(vp, "try-error")) {
      valid_px_protected  <-  as.integer(try(terra::global(vp,  "sum",  na.rm=TRUE)[1,1],
silent=TRUE))
      if (is.na(valid_px_protected)) valid_px_protected <- 0L
    }
  }

  m_unprot <- subset(miombo_t, protection == "Unprotected")
  if (nrow(m_unprot) > 0) {
    ru <- try(terra::mask(r_scl_c, terra::vect(m_unprot)), silent = TRUE)
    vu <- try(terra::mask(r_valid,  terra::vect(m_unprot)), silent = TRUE)
    if (!inherits(ru, "try-error")) {
      miombo_px_unprotected     <-     as.integer(try(terra::global(!is.na(ru),     "sum",
na.rm=TRUE)[1,1], silent=TRUE))
      if (is.na(miombo_px_unprotected)) miombo_px_unprotected <- 0L
    }
    if (!inherits(vu, "try-error")) {
```

```
      valid_px_unprotected <- as.integer(try(terra::global(vu, "sum", na.rm=TRUE)[1,1],
silent=TRUE))
      if (is.na(valid_px_unprotected)) valid_px_unprotected <- 0L
    }
  }

  rows[[i]] <- data.frame(
    tile_code = tile_code,
    tile_path = tile_dir,
    # eligible-area counts only (Uncertain already excluded)
    miombo_px_total      = as.integer(miombo_px_total),
    miombo_px_protected  = as.integer(miombo_px_protected),
    miombo_px_unprotected= as.integer(miombo_px_unprotected),
    valid_px_total       = as.integer(valid_px_total),
    valid_px_protected   = as.integer(valid_px_protected),
    valid_px_unprotected = as.integer(valid_px_unprotected),
    status    = "ok",
    stringsAsFactors = FALSE
  )

  setTxtProgressBar(pb, i)
}
close(pb)

counts <- data.table::rbindlist(rows, fill = TRUE)
fwrite(counts, file.path(OUT_DIR, "pass1_valid_pixel_counts_groups.csv"))

# grand totals (eligible only)
tot_miombo_total <- sum(counts$miombo_px_total, na.rm = TRUE)
tot_miombo_P     <- sum(counts$miombo_px_protected, na.rm = TRUE)
tot_miombo_U     <- sum(counts$miombo_px_unprotected, na.rm = TRUE)

tot_valid_total  <- sum(counts$valid_px_total, na.rm = TRUE)
tot_valid_P      <- sum(counts$valid_px_protected, na.rm = TRUE)
tot_valid_U      <- sum(counts$valid_px_unprotected, na.rm = TRUE)

writeLines(c(
  sprintf("Eligible Miombo pixels (any SCL): total=%s | Protected=%s | Unprotected=%s",
          format(tot_miombo_total, big.mark=","), format(tot_miombo_P, big.mark=","),
          format(tot_miombo_U, big.mark=",")),
  sprintf("Eligible VALID pixels (SCL good): total=%s | Protected=%s | Unprotected=%s",
          format(tot_valid_total, big.mark=","), format(tot_valid_P, big.mark=","),
          format(tot_valid_U, big.mark=",")),
  sprintf("Tiles processed: %d (ok=%d, issues=%d)",
          nrow(counts),
          sum(counts$status=="ok", na.rm=TRUE),
          sum(counts$status!="ok", na.rm=TRUE))
), file.path(OUT_DIR, "pass1_summary_groups.txt"))

message("Pass 1 (eligible protection only) CSV: ", file.path(OUT_DIR,
"pass1_valid_pixel_counts_groups.csv"))
message("Summary: ", file.path(OUT_DIR, "pass1_summary_groups.txt"))
```

```
# ============================
# SVA — RUN 2B: Replicate-Labeled Sample Bank (per-tile Parquet with rep_id)
# ============================

suppressPackageStartupMessages({
  library(sf)
  library(terra)
  library(data.table)
  library(fs)
  library(stringr)
  library(arrow)     # Parquet
})
```

```
# ---------- CONFIG ----------
root_dir    <- "C:/Users/david/OneDrive/Documents/Master ESS/Masterarbeit/Data/Sentinel 2
Bulk"
miombo_path <- "C:/Users/david/OneDrive/Documents/Master ESS/Masterarbeit/Data/Shapefiles
miombo extend in protected areas/miombo_classified_protection.shp"
# -- point to your Pass 1 results folder --
PASS1_DIR  <-  "C:/Users/david/OneDrive/Documents/Master  ESS/Masterarbeit/Data/Sentinel  2
Bulk/_SVA_RESULTS_20250818_210158"

# -- full file path to the CSV --
PASS1_COUNTS_CSV <- file.path(PASS1_DIR, "pass1_valid_pixel_counts_groups.csv")

# -- read (use file= to avoid the 'system command' issue with spaces) --
counts <- data.table::fread(file = PASS1_COUNTS_CSV)

# quick sanity check
stopifnot(nrow(counts) > 0)
stopifnot(all(c("tile_code","tile_path","valid_px_total") %in% names(counts)))

# Per-run target and number of independent replicates
TARGET_PER_RUN <- 50000L
R_REPS         <- 20L

# Bands & masks (20 m workflow)
BAND_IDS <- c("B01","B02","B03","B04","B05","B06","B07","B8A","B11","B12")
SCL_BAD  <- c(0,1,3,8,9,10,11)
GROUP_MAP <- c("Strictly Protected"="Protected", "Unprotected"="Unprotected")
ELIGIBLE_PROT <- names(GROUP_MAP)  # same classes as in Pass 1

OUT_DIR       <-   file.path(root_dir,   sprintf("_SVA_BANK_REPS_%s",   format(Sys.time(),
"%Y%m%d_%H%M%S")))
BANK_DIR  <- file.path(OUT_DIR, "bank")
dir_create(BANK_DIR, recurse = TRUE)
set.seed(42)

# ---------- Helpers ----------
list_safe_tiles <- function(root) {
  d <- list.dirs(root, full.names = TRUE, recursive = FALSE)
  d[grepl("\\.SAFE$", d)]
}
tile_code_from_path <- function(p) sub(".*_T([0-9A-Z]{5})_.*", "\\1", basename(p))

find_tile_files <- function(tile_dir, band_ids=BAND_IDS) {
  all_20m <- list.files(tile_dir, pattern = "_20m\\.jp2$", recursive = TRUE, full.names =
TRUE)
  if (!length(all_20m)) return(NULL)
  pick_one <- function(id) {
    m <- grep(paste0("_", id, "_20m\\.jp2$"), all_20m, value = TRUE)
    if (length(m)) m[1] else NA_character_
  }
  bands <- setNames(vapply(band_ids, pick_one, character(1)), band_ids)
  scl   <- pick_one("SCL")
  if (is.na(scl) || anyNA(bands)) return(NULL)
  list(bands=bands, scl=scl)
}

# ---------- Load Pass 1 (eligible-only) & compute per-run allocations ----------
counts <- fread(PASS1_COUNTS_CSV)
counts <- counts[status == "ok" & !is.na(valid_px_total) & valid_px_total > 0]

stopifnot(nrow(counts) > 0)

# Pure proportional allocation to TARGET_PER_RUN (no per-tile floors)
w <- counts$valid_px_total / sum(counts$valid_px_total)
k_alloc <- floor(w * TARGET_PER_RUN)
leftover <- TARGET_PER_RUN - sum(k_alloc)
```

```
if (leftover > 0) {
  frac <- (w * TARGET_PER_RUN) - k_alloc
  add_idx <- order(frac, decreasing = TRUE)[seq_len(leftover)]
  k_alloc[add_idx] <- k_alloc[add_idx] + 1L
}
counts[, k_alloc := k_alloc]
stopifnot(sum(counts$k_alloc) == TARGET_PER_RUN)

# Save planned allocations (per run) + expected total across all reps
alloc_out <- counts[, .(tile_code, tile_path, valid_px_total,
                        valid_px_protected, valid_px_unprotected,
                        k_alloc_per_run = k_alloc,
                        k_alloc_total = k_alloc * R_REPS)]
fwrite(alloc_out, file.path(OUT_DIR, "bank_reps_allocations.csv"))

# ---------- Load Miombo once (eligible classes only) ----------
miombo_sf <- st_read(miombo_path, quiet = TRUE) |>
  subset(!st_is_empty(geometry)) |>
  st_make_valid()

if (!"protection" %in% names(miombo_sf)) {
  nms <- tolower(names(miombo_sf))
  cand <- which(nms %in% c("protection","protect","prot_status","status"))
  if (length(cand)) names(miombo_sf)[cand[1]] <- "protection"
}
miombo_sf <- miombo_sf |>
  transform(protection = as.character(protection)) |>
  subset(!is.na(protection) & protection %in% ELIGIBLE_PROT)

# ---------- Build replicate-labeled bank per tile ----------
tiles_all <- list_safe_tiles(root_dir)
tile_map  <- setNames(tiles_all, vapply(tiles_all, tile_code_from_path, ""))

manifest_rows <- list()  # per tile summary
rep_rows <- list()       # per tile per rep summary rows (for index)

pb <- txtProgressBar(min = 0, max = nrow(counts), initial = 0, style = 3)

for (i in seq_len(nrow(counts))) {
  tcode <- counts$tile_code[i]
  tpath <- tile_map[[tcode]]
  kout  <- counts$k_alloc[i]

  out_file <- file.path(BANK_DIR, paste0("bank_", tcode, ".parquet"))

  status <- "ok"; note <- ""
  total_written <- 0L

  if (is.na(tpath) || !dir_exists(tpath)) {
    status <- "tile_not_found"; note <- "SAFE dir missing"
    manifest_rows[[i]] <- data.frame(tile_code=tcode, tile_path=NA, k_alloc_per_run=kout,
                                     reps=R_REPS, rows_total=0L, status=status, note=note,
file=NA)
    setTxtProgressBar(pb, i); next
  }

  files <- find_tile_files(tpath)
  if (is.null(files)) {
    status <- "missing_bands_or_scl"; note <- "required 20m bands or SCL missing"
    manifest_rows[[i]]       <-       data.frame(tile_code=tcode,       tile_path=tpath,
k_alloc_per_run=kout,
                                     reps=R_REPS, rows_total=0L, status=status, note=note,
file=NA)
    setTxtProgressBar(pb, i); next
  }
```

```
  # Read rasters once per tile
  r_bands <- try(terra::rast(unname(files$bands)), silent = TRUE)
  if (inherits(r_bands, "try-error")) {
    status <- "band_read_error"; note <- "terra::rast bands failed"
    manifest_rows[[i]]       <-       data.frame(tile_code=tcode,       tile_path=tpath,
k_alloc_per_run=kout,
                                    reps=R_REPS, rows_total=0L, status=status, note=note,
file=NA)
    setTxtProgressBar(pb, i); next
  }
  names(r_bands) <- names(files$bands)

  r_scl <- try(terra::rast(files$scl), silent = TRUE)
  if (inherits(r_scl, "try-error")) {
    status <- "scl_read_error"; note <- "terra::rast SCL failed"
    manifest_rows[[i]]       <-       data.frame(tile_code=tcode,       tile_path=tpath,
k_alloc_per_run=kout,
                                    reps=R_REPS, rows_total=0L, status=status, note=note,
file=NA)
    setTxtProgressBar(pb, i); next
  }

  # Mask to eligible Miombo
  miombo_t <- try(st_transform(miombo_sf, crs(terra::crs(r_bands, proj=TRUE))), silent =
TRUE)
  if (inherits(miombo_t, "try-error")) {
    status <- "miombo_transform_error"; note <- "CRS transform failed"
    manifest_rows[[i]]       <-       data.frame(tile_code=tcode,       tile_path=tpath,
k_alloc_per_run=kout,
                                    reps=R_REPS, rows_total=0L, status=status, note=note,
file=NA)
    setTxtProgressBar(pb, i); next
  }

  r_bands <- try(terra::crop(r_bands, terra::vect(miombo_t)), silent = TRUE)
  if (inherits(r_bands, "try-error")) {
    status <- "band_crop_error"; note <- "crop to Miombo failed"
    manifest_rows[[i]]       <-       data.frame(tile_code=tcode,       tile_path=tpath,
k_alloc_per_run=kout,
                                    reps=R_REPS, rows_total=0L, status=status, note=note,
file=NA)
    setTxtProgressBar(pb, i); next
  }
  r_bands <- try(terra::mask(r_bands, terra::vect(miombo_t)), silent = TRUE)
  if (inherits(r_bands, "try-error")) {
    status <- "band_mask_error"; note <- "mask to Miombo failed"
    manifest_rows[[i]]       <-       data.frame(tile_code=tcode,       tile_path=tpath,
k_alloc_per_run=kout,
                                    reps=R_REPS, rows_total=0L, status=status, note=note,
file=NA)
    setTxtProgressBar(pb, i); next
  }

  r_scl <- try(terra::crop(r_scl, r_bands), silent = TRUE)
  if (inherits(r_scl, "try-error")) {
    status <- "scl_crop_error"; note <- "SCL crop failed"
    manifest_rows[[i]]       <-       data.frame(tile_code=tcode,       tile_path=tpath,
k_alloc_per_run=kout,
                                    reps=R_REPS, rows_total=0L, status=status, note=note,
file=NA)
    setTxtProgressBar(pb, i); next
  }

  # Apply SCL validity
  rcl <- cbind(SCL_BAD, NA)
  r_scl_valid <- try(terra::classify(r_scl, rcl = rcl, others = 1), silent = TRUE)
```

```
  if (inherits(r_scl_valid, "try-error")) {
    status <- "scl_classify_error"; note <- "SCL classify failed"
    manifest_rows[[i]]       <-       data.frame(tile_code=tcode,       tile_path=tpath,
k_alloc_per_run=kout,
                                    reps=R_REPS, rows_total=0L, status=status, note=note,
file=NA)
    setTxtProgressBar(pb, i); next
  }
  r_bands <- try(terra::mask(r_bands, r_scl_valid, maskvalues = NA), silent = TRUE)
  if (inherits(r_bands, "try-error")) {
    status <- "scl_mask_apply_error"; note <- "apply SCL mask failed"
    manifest_rows[[i]]       <-       data.frame(tile_code=tcode,       tile_path=tpath,
k_alloc_per_run=kout,
                                    reps=R_REPS, rows_total=0L, status=status, note=note,
file=NA)
    setTxtProgressBar(pb, i); next
  }

  # Count valid cells after masking (for replace decision)
  valid_count   <-   try(terra::global(!is.na(r_bands[[1]]),   "sum",   na.rm=TRUE)[1,1],
silent=TRUE)
  if (inherits(valid_count, "try-error") || is.na(valid_count)) valid_count <- 0L
  if (valid_count == 0L) {
    status <- "no_valid_pixels_after_mask"
    manifest_rows[[i]]       <-       data.frame(tile_code=tcode,       tile_path=tpath,
k_alloc_per_run=kout,
                                    reps=R_REPS,  rows_total=0L,  status=status,  note="",
file=NA)
    setTxtProgressBar(pb, i); next
  }

  # Build all replicates for this tile
  tile_accum <- vector("list", R_REPS)
  reps_summary <- vector("list", R_REPS)

  for (r in seq_len(R_REPS)) {
    need <- kout
    rep_df <- NULL
    tries <- 0L

    repeat {
      tries <- tries + 1L
      smp <- try(terra::spatSample(r_bands, size = need, method = "random",
                                   replace = (need > valid_count),
                                   na.rm = TRUE, xy = TRUE, useGDAL = TRUE),
                 silent = TRUE)
      if (inherits(smp, "try-error") || !nrow(smp)) {
        # If sampling failed, break to avoid infinite loop
        break
      }
      pts <- sf::st_as_sf(smp, coords = c("x","y"), crs = sf::st_crs(miombo_t))
      pts <- suppressWarnings(sf::st_join(pts, miombo_t["protection"], left = TRUE))

      grp <- unname(GROUP_MAP[as.character(pts$protection)])
      df  <- as.data.frame(smp)
      names(df)[names(df) %in% c("x","y")] <- c("x","y")
      df$group <- grp
      df$tile_code <- tcode
      df$rep_id <- r

      # Keep only recognized groups (should be all, but guard anyway)
      df <- df[!is.na(df$group), , drop = FALSE]

      if (is.null(rep_df)) rep_df <- df else rep_df <- rbind(rep_df, df)
      have <- nrow(rep_df)
      if (have >= kout || tries >= 3L) break
```

```
      need <- kout - have
    }

    # Clip to kout if we overshot; record summary
    if (!is.null(rep_df) && nrow(rep_df) > kout) rep_df <- rep_df[seq_len(kout), , drop =
FALSE]
    got <- if (is.null(rep_df)) 0L else nrow(rep_df)
    tile_accum[[r]] <- rep_df
    reps_summary[[r]] <- data.frame(tile_code=tcode, rep_id=r, rows=got)

    total_written <- total_written + got
  }

  # Write per-tile Parquet with all reps (if any rows)
  tile_df <- data.table::rbindlist(tile_accum, use.names = TRUE, fill = TRUE)
  if (nrow(tile_df)) {
    tryCatch({
      arrow::write_parquet(tile_df, out_file)
    }, error = function(e) {
      status <<- "parquet_write_error"; note <<- conditionMessage(e)
    })
  } else {
    status <- "no_rows_written"
  }

  # Record summaries
  rep_rows[[i]] <- data.table::rbindlist(reps_summary, fill = TRUE)
  manifest_rows[[i]] <- data.frame(
    tile_code = tcode,
    tile_path = tpath,
    k_alloc_per_run = kout,
    reps = R_REPS,
    rows_total = total_written,
    status = status,
    note = note,
    file = if (status=="ok") out_file else NA_character_
  )

  setTxtProgressBar(pb, i)
}
close(pb)

# ---------- Write indexes & summary ----------
bank_rep_index <- data.table::rbindlist(rep_rows, fill = TRUE)
fwrite(bank_rep_index, file.path(OUT_DIR, "bank_reps_index.csv"))  # per tile x rep rows

bank_manifest <- data.table::rbindlist(manifest_rows, fill = TRUE)
fwrite(bank_manifest, file.path(OUT_DIR, "bank_reps_manifest.csv"))

tot_rows <- sum(bank_manifest$rows_total, na.rm = TRUE)
target_total_all_reps <- TARGET_PER_RUN * R_REPS

writeLines(c(
  sprintf("Replicates: %d", R_REPS),
  sprintf("Per-run target rows: %s", format(TARGET_PER_RUN, big.mark=",")),
  sprintf("Expected   total   across   all   reps:   %s",   format(target_total_all_reps,
big.mark=",")),
  sprintf("Actual total rows written: %s", format(tot_rows, big.mark=",")),
  sprintf("Tiles   ok:   %d   /   %d",   sum(bank_manifest$status=="ok",   na.rm=TRUE),
nrow(bank_manifest))
), file.path(OUT_DIR, "bank_reps_summary.txt"))

message("Run 2B complete.")
message("Allocations: ", file.path(OUT_DIR, "bank_reps_allocations.csv"))
message("Per-tile Parquets in: ", BANK_DIR, " (now with rep_id)")
message("Per-replicate index: ", file.path(OUT_DIR, "bank_reps_index.csv"))
```

```
message("Manifest: ", file.path(OUT_DIR, "bank_reps_manifest.csv"))
message("Summary: ", file.path(OUT_DIR, "bank_reps_summary.txt"))
```

```
# ============================
# PASS Vegetation Index Per-run group means as boxplots (NDVI, NDMI, CCI)
# Computes per-run means in Arrow and plots boxplots across runs, per group
# ============================

suppressPackageStartupMessages({
  library(arrow)
  library(dplyr)
  library(tidyr)
  library(ggplot2)
  library(fs)
  library(scales)
  library(data.table)
})

# ---------- USER CONFIG ----------
BASE_DIR  <-  "C:/Users/david/OneDrive/Documents/Master  ESS/Masterarbeit/Data/Sentinel  2
Bulk/_SVA_BANK_REPS_20250818_213305"
BANK_DIR <- file.path(BASE_DIR, "bank")
OUT_DIR  <- file.path(BASE_DIR, "PASS8_IDX_RUN_MEANS")
dir_create(OUT_DIR, recurse = TRUE)

# Colors consistent with your other plots
COLS <- c(Protected = "#1b9e77", Unprotected = "#d95f02")

# Small epsilon to avoid division by zero (use addition so it stays Arrow-friendly)
eps <- 1e-9

# ---------- LOAD + COMPUTE MEANS (IN ARROW) ----------
ds <- open_dataset(BANK_DIR, format = "parquet")

# Ensure needed columns exist
need_cols <- c("rep_id","group","B04","B05","B06","B07","B8A","B11")
stopifnot(all(need_cols %in% names(ds)))

# Compute indices and per-run, per-group means entirely in Arrow, then collect
means_wide <- ds %>%
  transmute(
    rep_id,
    group,
    NDVI = (B8A - B04) / (B8A + B04 + eps),
    CCI  = ((B8A / (B05 + eps) - 1) +
              (B8A / (B06 + eps) - 1) +
              (B8A / (B07 + eps) - 1)) / 3,
    NDMI = (B8A - B11) / (B8A + B11 + eps)
  ) %>%
  group_by(rep_id, group) %>%
  summarise(
    mean_NDVI = mean(NDVI, na.rm = TRUE),
    mean_CCI  = mean(CCI,  na.rm = TRUE),
    mean_NDMI = mean(NDMI, na.rm = TRUE),
    .groups = "drop"
  ) %>%
  collect()

# Sanity: ensure both groups exist at least somewhere
stopifnot(any(means_wide$group == "Protected"), any(means_wide$group == "Unprotected"))

# Long form for plotting
means_long <- means_wide %>%
  pivot_longer(cols = starts_with("mean_"),
               names_to = "index",
               values_to = "mean_val") %>%
```

```
  mutate(
    index = recode(index,
                   mean_NDVI = "NDVI",
                   mean_NDMI = "NDMI",
                   mean_CCI  = "CCI"
    ),
    # facet order: NDVI, NDMI, CCI
    index = factor(index, levels = c("NDVI","NDMI","CCI")),
    group = factor(group, levels = c("Protected","Unprotected"))
  )

# Count how many runs were found (per group may differ if some runs miss a group)
n_runs_total <- length(unique(means_long$rep_id))

# ---------- SAVE CSV ----------
fwrite(means_long, file.path(OUT_DIR, "per_run_group_means_long.csv"))

# ---------- PLOTS ----------
LABS <- c(NDVI="NDVI", NDMI="NDMI", CCI="CCI")

PLOT_TITLE    <- "Governance differences in mean vegetation indices across 20 runs"
PLOT_SUBTITLE <- "Protected vs. Unprotected; per-run averages from Sentinel-2"
PLOT_CAPTION  <- "CCI = mean((B8A/B05−1), (B8A/B06−1), (B8A/B07−1))"

p_box <- ggplot(means_long, aes(x = group, y = mean_val, fill = group, color = group)) +
  geom_boxplot(width = 0.45, outlier.shape = NA, alpha = 0.9) +
  geom_jitter(width = 0.10, height = 0, size = 1.6, alpha = 0.7) +
  facet_wrap(~ index, ncol = 3, scales = "fixed", labeller = as_labeller(LABS)) +
  scale_fill_manual(values = COLS, guide = "none") +
  scale_color_manual(values = COLS, guide = "none") +
  scale_y_continuous(
    breaks = seq(0, 0.5, 0.05),
    labels = scales::label_number(accuracy = 0.01),
    expand = expansion(mult = c(0, 0.02))
  ) +
  coord_cartesian(ylim = c(0, 0.5), clip = "off") +   # allow annotation near edges
  labs(
    title    = PLOT_TITLE,
    subtitle = PLOT_SUBTITLE,
    x = NULL, y = "Per-run mean value",
    caption  = PLOT_CAPTION
  ) +
  theme_minimal(base_size = 12) +
  theme(
    panel.grid.minor   = element_blank(),
    panel.grid.major.x = element_blank(),
    strip.text         = element_text(face = "bold"),
    strip.background   = element_rect(fill = "#F3F4F6", color = NA),
    axis.text.x        = element_text(face = "bold"),
    plot.margin        = margin(t = 6, r = 16, b = 6, l = 6)
  )

# --- Δ across runs WITHOUT CI: (mean of Unprotected per-run means) − (mean of Protected
per-run means)
delta_summ <- means_long %>%
  group_by(index, group) %>%
  summarise(m = mean(mean_val, na.rm = TRUE), .groups = "drop") %>%
  tidyr::pivot_wider(names_from = group, values_from = m) %>%
  mutate(delta = Unprotected - Protected) %>%
  transmute(
    index,
    label = sprintf("Δ= %+.3f", delta),
    x = 1.75,   # place near 'Unprotected'
    y = 0.49   # top of the fixed 0–0.5 range
  )
```

```
# add annotation AFTER building p_box (keep coord_cartesian(..., clip = "off"))
p_box <- p_box +
  geom_text(
    data = delta_summ,
    aes(x = x, y = y, label = label),
    inherit.aes = FALSE,
    hjust = 1, vjust = 1,
    size = 3.6, fontface = "bold"
  )


ggsave(file.path(OUT_DIR, "box_per_run_means_by_group.png"),
       p_box, width = 12, height = 4.8, dpi = 300)
```

```
       # ============================
# PASS 8X — Single-run violin+box (NDVI, NDMI, CCI)
# Minimal, plot-only version with saving
# ============================

suppressPackageStartupMessages({
  library(arrow)
  library(dplyr)
  library(tidyr)
  library(ggplot2)
  library(fs)
})

# ---------- USER CONFIG ----------
BASE_DIR  <-  "C:/Users/david/OneDrive/Documents/Master  ESS/Masterarbeit/Data/Sentinel  2
Bulk/_SVA_BANK_REPS_20250818_213305"
BANK_DIR <- file.path(BASE_DIR, "bank")
OUT_DIR  <- file.path(BASE_DIR, "PASS8_IDX_REP_RUN_MIN")
dir_create(OUT_DIR, recurse = TRUE)

# Force which run to plot
FORCE_REP_ID <- 12L

# Figure text (edit these)
PLOT_TITLE    <- "Vegetation index distributions by governance for run 12"
PLOT_SUBTITLE <- "Protected vs. Unprotected; per-pixel NDVI, NDMI, and CCI from Sentinel-
2"

# Colors
COLS <- c(Protected = "#1b9e77", Unprotected = "#d95f02")

# ---------- Pick/validate run ----------
get_rep_id <- function(bank_dir, force_id) {
  ds <- open_dataset(bank_dir, format = "parquet") |>
    dplyr::select(rep_id) |>
    dplyr::distinct() |>
    collect()
  avail <- sort(unique(ds$rep_id))
  if (!force_id %in% avail) {
    stop(sprintf("rep_id %d not found in BANK_DIR. Available: %s",
                 force_id, paste(avail, collapse = ", ")))
  }
  force_id
}
REP_ID <- get_rep_id(BANK_DIR, FORCE_REP_ID)
message(sprintf("Using rep_id = %02d", REP_ID))

# ---------- Load ONE run + compute indices (in Arrow) ----------
eps <- 1e-9
ds <- open_dataset(BANK_DIR, format = "parquet")
need_cols <- c("rep_id","group","B04","B05","B06","B07","B8A","B11")
stopifnot(all(need_cols %in% names(ds)))
```

```
idx_tbl <- ds %>%
  filter(rep_id == REP_ID) %>%
  transmute(
    group,
    NDVI = (B8A - B04) / (B8A + B04 + eps),
    CCI  = ((B8A/(B05 + eps) - 1) + (B8A/(B06 + eps) - 1) + (B8A/(B07 + eps) - 1)) / 3,
    NDMI = (B8A - B11) / (B8A + B11 + eps)
  ) %>%
  collect()

stopifnot(nrow(idx_tbl) > 0)

LONG <- idx_tbl %>%
  pivot_longer(c(NDVI, NDMI, CCI), names_to = "index", values_to = "val") %>%
  filter(is.finite(val), !is.na(group)) %>%
  mutate(
    index = factor(index, levels = c("NDVI","NDMI","CCI")),
    group = factor(group, levels = c("Protected","Unprotected"))
  )

# ---------- Plot ----------
LABS <- c(NDVI = "NDVI", NDMI = "NDMI", CCI = "CCI")

p_violin <- ggplot(LONG, aes(x = group, y = val, fill = group, color = group)) +
  geom_violin(trim = TRUE, alpha = 0.30, width = 0.90, draw_quantiles = 0.5) +
  geom_boxplot(width = 0.16, outlier.shape = NA, linewidth = 0.6, fatten = 2) +
  facet_wrap(~ index, scales = "free_y", ncol = 3, labeller = as_labeller(LABS)) +
  scale_fill_manual(values = COLS, guide = "none") +
  scale_color_manual(values = COLS, guide = "none") +
  labs(
    title    = PLOT_TITLE,
    subtitle = PLOT_SUBTITLE,
    x = NULL,
    y = "Index value",
  ) +
  theme_minimal(base_size = 12) +
  theme(
    panel.grid.minor = element_blank(),
    strip.text = element_text(face = "bold"),
    strip.background = element_rect(fill = "#F3F4F6", color = NA),
    axis.text.x = element_text(face = "bold")
  )

# ---------- Save (PNG + SVG) ----------
png_path <- file.path(OUT_DIR, sprintf("violin_run%02d.png", REP_ID))

ggsave(png_path, p_violin, width = 12, height = 4.8, dpi = 300)
```

```
# ============================
# PCA (10 bands) — Run PCA 10: full scatter + per-group density + compass
# ============================
suppressPackageStartupMessages({
  library(arrow); library(dplyr); library(data.table)
  library(ggplot2); library(scales); library(fs)
  library(cowplot)      # layout + annotations
  library(ggrepel)      # labels on compass
})

# ---------- CONFIG ----------
RUN_ID         <- 12L
TARGET_PER_RUN <- 50000L
BAND_IDS       <- c("B01","B02","B03","B04","B05","B06","B07","B8A","B11","B12")

# Where your bank & run live (update timestamp if needed)
```

```
BANK_BASE_DIR <- "C:/Users/david/OneDrive/Documents/Master ESS/Masterarbeit/Data/Sentinel
2 Bulk/_SVA_BANK_REPS_20250818_213305"
BANK_DIR      <- file.path(BANK_BASE_DIR, "bank")
RUN_DIR       <- file.path(BANK_BASE_DIR, "PCA_RUNS", sprintf("RUN_%02d", RUN_ID))

# New output base
OUT_ROOT <- "C:/Users/david/OneDrive/Documents/Master ESS/Masterarbeit/Data/SVA National"
OUT_DIR  <- file.path(OUT_ROOT, sprintf("PCA_10B_RUN%02d", RUN_ID))
dir_create(OUT_DIR, recurse = TRUE)

# Optional: cache PC1–PC2 scores (so plotting is instant next time)
CACHE_SCORES <- TRUE
SCORES_PARQUET <- file.path(OUT_DIR, sprintf("scores_PC12_run%02d.parquet", RUN_ID))

# Colors
GROUP_COLS <- c("Unprotected" = "#d95f02", "Protected" = "#1b9e77")

# ---------- BUILD PCA MODEL + PC1/PC2 SCORES (INSERTED BLOCK) ----------
# We need pca_model and scores before any code that references them.

# 0) a tiny helper used later in your code
`%||%` <- function(x, y) if (!is.null(x)) x else y

# 1) Pull data for this RUN
ds <- open_dataset(BANK_DIR, format = "parquet")
df_pca <- ds %>%
  filter(rep_id == RUN_ID) %>%
  select(all_of(c("group", BAND_IDS))) %>%
  collect() %>%
  as.data.frame()

# 2) Scale to reflectance if your bank stores DN ≈ reflectance*10000
df_pca[, BAND_IDS] <- df_pca[, BAND_IDS] / 10000

# 3) Keep complete rows + non-missing group
df_pca <- df_pca[stats::complete.cases(df_pca[, BAND_IDS, drop = FALSE]) &
                   !is.na(df_pca$group), , drop = FALSE]
df_pca$group <- as.character(df_pca$group)

# 4) Fit PCA (10 components)
pca_model <- prcomp(df_pca[, BAND_IDS, drop = FALSE],
                    center = TRUE, scale. = TRUE, rank. = 10)

# 5) Save PCA model for reuse (so PASS-B can load it)
fs::dir_create(RUN_DIR, recurse = TRUE)  # ensure run folder exists
saveRDS(pca_model, file.path(RUN_DIR, sprintf("pca_model_run%02d.rds", RUN_ID)))
# optional convenience copy alongside your figures
saveRDS(pca_model, file.path(OUT_DIR, sprintf("pca_model_run%02d.rds", RUN_ID)))

# 6) Make PC1/PC2 scores data.frame `scores` used by plots/metrics below
PCs <- predict(pca_model, df_pca[, BAND_IDS, drop = FALSE])
scores <- data.frame(
  PC1   = PCs[, 1],
  PC2   = PCs[, 2],
  group = df_pca$group,
  stringsAsFactors = FALSE
)

# 7) Variance explained labels used in axis titles and subtitles
ve      <- (pca_model$sdev^2) / sum(pca_model$sdev^2)
vx1     <- round(100 * ve[1], 1)
vx2     <- round(100 * ve[2], 1)
vx12    <- round(100 * (ve[1] + ve[2]), 1)

# 8) Optional cache of scores (for instant plotting later)
if (isTRUE(CACHE_SCORES)) {
```

```
  # write as Parquet; will overwrite if exists
  arrow::write_parquet(arrow::as_arrow_table(scores), SCORES_PARQUET)
}
# ---------- LOCK PCA ORIENTATION (consistent signs for PCs) ----------
# Rule: force sum of loadings on each PC to be positive.
# If negative, flip that PC in both the rotation AND the scores.

fix_pc_sign <- function(model, scores_df) {
  rot <- model$rotation

  flip1 <- sum(rot[, 1]) < 0
  flip2 <- sum(rot[, 2]) < 0   # flip PC2 too if you want a fixed rule for both

  if (flip1) {
    model$rotation[, 1] <- -model$rotation[, 1]
    scores_df$PC1       <- -scores_df$PC1
  }
  if (flip2) {
    model$rotation[, 2] <- -model$rotation[, 2]
    scores_df$PC2       <- -scores_df$PC2
  }

  list(model = model, scores = scores_df, flips = c(PC1 = ifelse(flip1, -1, 1),
                                                    PC2 = ifelse(flip2, -1, 1)))
}

fixed <- fix_pc_sign(pca_model, scores)
pca_model <- fixed$model
scores    <- fixed$scores
pc_flips  <- fixed$flips  # keep for reference/logging if you like

# Re-save the oriented model so everything downstream uses the same signs
fs::dir_create(RUN_DIR, recurse = TRUE)
saveRDS(pca_model, file.path(RUN_DIR, sprintf("pca_model_run%02d.rds", RUN_ID)))
saveRDS(pca_model, file.path(OUT_DIR, sprintf("pca_model_run%02d.rds", RUN_ID)))


# centroids for overlays
centroids <- scores |>
  group_by(group) |>
  summarise(PC1 = mean(PC1), PC2 = mean(PC2), .groups = "drop")
# ---- separability metrics on PC1–PC2 (all points) ----
sep_metrics <- (function(df){
  ridge <- 1e-8
  P <- subset(df, group == "Protected",   select = c(PC1, PC2))
  U <- subset(df, group == "Unprotected", select = c(PC1, PC2))
  if (nrow(P) < 2 || nrow(U) < 2) return(list(DM=NA_real_, DB=NA_real_, JM=NA_real_))

  muP <- colMeans(P); muU <- colMeans(U); dmu <- as.numeric(muP - muU)

  S1 <- stats::cov(P); S2 <- stats::cov(U)
  n1 <- nrow(P); n2 <- nrow(U)
  Spooled <- ((n1 - 1) * S1 + (n2 - 1) * S2) / max(1, (n1 + n2 - 2))

  inv <- function(M) solve(M + diag(ridge, 2))
  logdet <- function(M) as.numeric(determinant(M + diag(ridge,2), log = TRUE)$modulus)

  # Mahalanobis (pooled within-class covariance)
  DM <- sqrt(t(dmu) %*% inv(Spooled) %*% dmu)

  # Bhattacharyya + Jeffries–Matusita
  Sbar  <- (S1 + S2) / 2
  term1 <- 0.125 * as.numeric(t(dmu) %*% inv(Sbar) %*% dmu)
  term2 <- 0.5 * (logdet(Sbar) - 0.5 * (logdet(S1) + logdet(S2)))
  DB <- term1 + term2
  JM <- 2 * (1 - exp(-DB))
```

```
  list(DM = as.numeric(DM), DB = as.numeric(DB), JM = as.numeric(JM))
})(scores)

# ---------- METRICS ----------
# Full-cloud cheap metrics
n_tot <- nrow(scores)
n_by  <- table(scores$group)
nU <- ifelse("Unprotected" %in% names(n_by), as.integer(n_by[["Unprotected"]]), 0L)
nP <- ifelse("Protected"   %in% names(n_by), as.integer(n_by[["Protected"]]),   0L)

# Centroid Euclidean distance in PC1–PC2 space
dist_PC12 <- {
  cU <- centroids[centroids$group=="Unprotected", c("PC1","PC2")]
  cP <- centroids[centroids$group=="Protected",   c("PC1","PC2")]
  if (nrow(cU) == 1 && nrow(cP) == 1) {
    sqrt((cU$PC1 - cP$PC1)^2 + (cU$PC2 - cP$PC2)^2)
  } else NA_real_
}



# Save metrics
fwrite(data.frame(
  run = RUN_ID,
  n_total = n_tot, n_unprotected = nU, n_protected = nP,
  centroid_dist_PC12 = dist_PC12,
  mahalanobis_PC12   = sep_metrics$DM,
  bhattacharyya_PC12 = sep_metrics$DB,
  jm_PC12            = sep_metrics$JM,
  PC1_var = vx1/100, PC2_var = vx2/100, PC12_var = vx12/100
), file.path(OUT_DIR, sprintf("metrics_run%02d.csv", RUN_ID)))


# ---------- PLOTS ----------
theme_pca <- function() {
  theme_minimal(base_size = 12) +
    theme(panel.grid.minor = element_blank(),
          panel.grid.major = element_line(size = 0.3))
}

# A) Full scatter (all points)
p_scatter <- ggplot(scores, aes(PC1, PC2, color = group)) +
  geom_point(size = 0.18, alpha = 0.20) +
  stat_ellipse(aes(fill = group), type = "norm", level = 0.95,
               geom = "polygon", alpha = 0.12, color = NA) +
  stat_ellipse(type = "norm", level = 0.95, linewidth = 0.6) +
  geom_point(data = centroids, shape = 4, stroke = 1.2, size = 4, color = "black") +
  coord_equal() +
  scale_color_manual(values = GROUP_COLS, name = NULL) +
  scale_fill_manual(values = GROUP_COLS, guide = "none") +
  labs(
    title = sprintf("National PCA space — run %d (10 bands)", RUN_ID),
    subtitle = sprintf("PC1 %.1f%% • PC2 %.1f%% • total %.1f%%   |   N = %s  (Protected =
%s, Unprotected = %s)",
                       vx1, vx2, vx12, comma(n_tot), comma(nP), comma(nU)),
    x = sprintf("PC1 (%.1f%%)", vx1),
    y = sprintf("PC2 (%.1f%%)", vx2)
  ) + theme_pca()

ggsave(file.path(OUT_DIR, sprintf("plotA_pca_scatter_allpoints_run%02d.png", RUN_ID)),
       p_scatter, width = 9.5, height = 7.5, dpi = 300)

# ---------- Left panel: per-group density (legend shown), no ellipses ----------
if (!exists("theme_pca")) {
  theme_pca <- function() {
```

```
    theme_minimal(base_size = 12) +
      theme(
        panel.grid.minor = element_blank(),
        panel.grid.major = element_line(linewidth = 0.3)
      )
  }
}

p_density <- ggplot() +
  # Unprotected layer
  stat_density_2d(
    data = subset(scores, group == "Unprotected"),
    aes(x = PC1, y = PC2, fill = "Unprotected", alpha = after_stat(level)),
    geom = "polygon", bins = 12, color = NA
  ) +
  # Protected layer
  stat_density_2d(
    data = subset(scores, group == "Protected"),
    aes(x = PC1, y = PC2, fill = "Protected", alpha = after_stat(level)),
    geom = "polygon", bins = 12, color = NA
  ) +
  # Centroids (keep the X marks)
  geom_point(
    data = centroids, aes(PC1, PC2, color = group),
    shape = 4, stroke = 1.25, size = 4
  ) +
  scale_alpha(range = c(0.03, 0.60), guide = "none") +
  scale_fill_manual(
    values = GROUP_COLS,
    breaks = c("Unprotected", "Protected"),
    name   = "Density by group"
  ) +
  scale_color_manual(values = GROUP_COLS, guide = "none") +
  coord_equal() +
  labs(
    title = "National per-group density 10 Bands",
    x = sprintf("PC1 (%.1f%%)", vx1),
    y = sprintf("PC2 (%.1f%%)", vx2)
  ) +
  theme_pca() +
  theme(legend.position = "bottom")

# ---------- Right panel: loadings as bars (PC1 & PC2) ----------
LD <- as.data.frame(pca_model$rotation[, 1:2, drop = FALSE])
LD$band <- rownames(LD)

make_loading_bar <- function(df, pc = "PC1", title_txt = NULL) {
  df$val <- df[[pc]]
  df <- df[order(abs(df$val), decreasing = FALSE), ]        # small at bottom, big at top
after coord_flip
  df$band <- factor(df$band, levels = df$band)
  ggplot(df, aes(x = band, y = val, fill = val >= 0)) +
    geom_col(width = 0.7) +
    geom_hline(yintercept = 0, linewidth = 0.3) +
    coord_flip() +
    scale_fill_manual(values = c("TRUE" = "#737373", "FALSE" = "#bdbdbd"), guide = "none")
+
    labs(title = title_txt %||% paste(pc, "loadings"), x = NULL, y = "Loading") +
    theme_minimal(base_size = 10) +
    theme(
      plot.title = element_text(size = 10, face = "bold", hjust = 0),
      axis.text.y = element_text(size = 9),
      panel.grid.minor = element_blank()
    )
}
```

```
p_bar1 <- make_loading_bar(LD, "PC1", "PC1 loadings")
p_bar2 <- make_loading_bar(LD, "PC2", "PC2 loadings")
bars_right <- cowplot::plot_grid(p_bar1, p_bar2, ncol = 1, rel_heights = c(0.55, 0.45))

# ---------- Compose (no middle % label) + SEPARABILITY box ----------
sep_text <- sprintf(
  paste(
    "Separability in PC1–PC2",
    "Centroid dist: %s",
    "Mahalanobis: %s",
    "Bhattacharyya: %s",
    "Jeffries–Matusita: %s",
    sep = "\n"
  ),
  ifelse(is.na(dist_PC12), "NA", sprintf("%.3f", dist_PC12)),
  ifelse(is.na(sep_metrics$DM), "NA", sprintf("%.3f", sep_metrics$DM)),
  ifelse(is.na(sep_metrics$DB), "NA", sprintf("%.3f", sep_metrics$DB)),
  ifelse(is.na(sep_metrics$JM), "NA", sprintf("%.3f", sep_metrics$JM)),
  format(n_tot, big.mark = ","), format(nP, big.mark = ","), format(nU, big.mark = ",")
)

label_plot <- ggplot() +
  annotate(
    "label", x = 1, y = 0, label = sep_text,
    hjust = 1, vjust = 0, size = 3.2,
    label.size = 0.25,
    label.padding = grid::unit(c(0.25,0.25,0.25,0.25), "lines"),
    colour = "black", fill = "white"
  ) +
  coord_cartesian(xlim = c(0,1), ylim = c(0,1), expand = FALSE) +
  theme_void()

base <- cowplot::plot_grid(p_density, bars_right, ncol = 2, rel_widths = c(0.74, 0.26),
align = "h")
p_composite <- cowplot::ggdraw(base) +
  cowplot::draw_plot(label_plot, x = 0.62, y = 0.02, width = 0.36, height = 0.22)

ggsave(file.path(OUT_DIR, sprintf("plotB_pca_density_bars_run%02d.png", RUN_ID)),
       p_composite, width = 12, height = 8, dpi = 320)
message("Saved pretty plot with bar loadings + separability box to: ", OUT_DIR)



# ---------- PCA stats export: variance (PC1-10) + loadings (PC1-10) ----------
# Ensure bands are named
if (is.null(rownames(pca_model$rotation))) {
  if (exists("BAND_IDS")) rownames(pca_model$rotation) <- BAND_IDS
  else rownames(pca_model$rotation) <- paste0("B", seq_len(nrow(pca_model$rotation)))
}

# Variance explained
ve_all <- (pca_model$sdev^2) / sum(pca_model$sdev^2)
k <- min(10L, length(ve_all))
VE_TBL <- data.frame(
  pc                    = paste0("PC", seq_len(k)),
  variance_proportion   = as.numeric(ve_all[1:k]),
  variance_percent      = round(100 * ve_all[1:k], 3),
  cumulative_proportion = as.numeric(cumsum(ve_all)[1:k]),
  cumulative_percent    = round(100 * cumsum(ve_all)[1:k], 3),
  stringsAsFactors = FALSE
)

# Loadings (wide, bands x PCs)
LD <- as.data.frame(pca_model$rotation[, 1:k, drop = FALSE])
LD$band <- rownames(LD)
```

```
# Optional: add wavelengths to band labels, if these names match
band_nm <- c(B01=443, B02=490, B03=560, B04=665, B05=705, B06=740, B07=783, B8A=865,
B11=1610, B12=2190)
if (all(LD$band %in% names(band_nm))) {
  LD$band_label <- paste0(LD$band, " (", band_nm[LD$band], " nm)")
} else {
  LD$band_label <- LD$band
}

# Put labels first, then the PC columns
LD_out <- LD[, c("band", "band_label", paste0("PC", seq_len(k)))]

# File paths
xlsx_path <- file.path(OUT_DIR, sprintf("pca_stats_run%02d.xlsx", RUN_ID))
ve_csv    <- file.path(OUT_DIR, sprintf("pca_variance_run%02d.csv", RUN_ID))
ld_csv    <- file.path(OUT_DIR, sprintf("pca_loadings_run%02d.csv", RUN_ID))

# Write Excel if available; otherwise CSVs
if (requireNamespace("writexl", quietly = TRUE)) {
  writexl::write_xlsx(list(variance = VE_TBL, loadings = LD_out), path = xlsx_path)
  message("📘 PCA stats workbook: ", xlsx_path)
} else {
  data.table::fwrite(VE_TBL, ve_csv)
  data.table::fwrite(LD_out, ld_csv)
  message("📄 Wrote CSVs: ", ve_csv, " and ", ld_csv, " (install {writexl} to get a single
.xlsx)")
}
```

```
# ============================
# PASS Vegetation Indexes distribution — Per-run violin+box (NDVI, NDMI, CCI)
# Loops all runs, adds 5-decimal means (printed, captioned, and saved as CSV)
# ============================

suppressPackageStartupMessages({
  library(arrow)
  library(dplyr)
  library(tidyr)
  library(ggplot2)
  library(fs)
  library(glue)
})

# ---------- USER CONFIG ----------
BASE_DIR <- "C:/Users/david/OneDrive/Documents/Master ESS/Masterarbeit/Data/Sentinel 2
Bulk/_SVA_BANK_REPS_20250818_213305"
BANK_DIR <- file.path(BASE_DIR, "bank")
OUT_DIR      <-    "C:/Users/david/OneDrive/Documents/Master    ESS/Masterarbeit/Data/SVA
National/Vegetations indexes"
dir_create(OUT_DIR, recurse = TRUE)

# Colors
COLS <- c(Protected = "#1b9e77", Unprotected = "#d95f02")

# ---------- Discover available runs ----------
ds <- open_dataset(BANK_DIR, format = "parquet")
need_cols <- c("rep_id","group","B04","B05","B06","B07","B8A","B11")
stopifnot(all(need_cols %in% names(ds)))

rep_ids <- ds |>
  dplyr::select(rep_id) |>
  dplyr::distinct() |>
  collect() |>
  {\(x) sort(unique(x$rep_id))}()

message(sprintf("Found %d runs: %s", length(rep_ids), paste(rep_ids, collapse = ", ")))
```

```
# ---------- Plot helper ----------
make_plot <- function(long_df, run_id, caption_text) {
  LABS <- c(NDVI = "NDVI", NDMI = "NDMI", CCI = "CCI")
  PLOT_TITLE   <- glue("Vegetation index distributions by governance — run {sprintf('%02d',
run_id)}")
  PLOT_SUBTITLE <- "Protected vs. Unprotected; per-pixel NDVI, NDMI, and CCI from Sentinel-
2"

  ggplot(long_df, aes(x = group, y = val, fill = group, color = group)) +
    geom_violin(trim = FALSE, alpha = 0.30, width = 0.90, draw_quantiles = 0.5) +
    geom_boxplot(width = 0.16, outlier.shape = NA, linewidth = 0.6, fatten = 2) +
    facet_wrap(~ index, scales = "free_y", ncol = 3, labeller = as_labeller(LABS)) +
    scale_fill_manual(values = COLS, guide = "none") +
    scale_color_manual(values = COLS, guide = "none") +
    labs(
      title    = PLOT_TITLE,
      subtitle = PLOT_SUBTITLE,
      x = NULL,
      y = "Index value",
    ) +
    theme_minimal(base_size = 12) +
    theme(
      panel.grid.minor = element_blank(),
      strip.text = element_text(face = "bold"),
      strip.background = element_rect(fill = "#F3F4F6", color = NA),
      axis.text.x = element_text(face = "bold"),
    )
}

# ---------- Loop runs: compute indices in Arrow, plot, save ----------
eps <- 1e-9

for (rid in rep_ids) {
  message(glue("Processing rep_id = {sprintf('%02d', rid)} ..."))

  idx_tbl <- ds %>%
    filter(rep_id == rid) %>%
    transmute(
      group,
      NDVI = (B8A - B04) / (B8A + B04 + eps),
      CCI  = ((B8A/(B05 + eps) - 1) + (B8A/(B06 + eps) - 1) + (B8A/(B07 + eps) - 1)) / 3,
      NDMI = (B8A - B11) / (B8A + B11 + eps)
    ) %>%
    collect()

  if (nrow(idx_tbl) == 0) {
    warning(glue("No rows for rep_id {rid}; skipping."))
    next
  }

  LONG <- idx_tbl %>%
    pivot_longer(c(NDVI, NDMI, CCI), names_to = "index", values_to = "val") %>%
    filter(is.finite(val), !is.na(group)) %>%
    mutate(
      index = factor(index, levels = c("NDVI","NDMI","CCI")),
      group = factor(group, levels = c("Protected","Unprotected"))
    )

  # ---- 5-decimal means by index & group ----
  means_tbl <- LONG %>%
    group_by(index, group) %>%
    summarize(mean = mean(val, na.rm = TRUE), .groups = "drop") %>%
    mutate(mean_5 = sprintf("%.5f", mean))

  # Print nicely to console
```

```
  message("Means (5 d.p.):")
  for (idx in levels(LONG$index)) {
    row_prot <- means_tbl %>% filter(index == idx, group == "Protected") %>% pull(mean_5)
    row_unpr <- means_tbl %>% filter(index == idx, group == "Unprotected") %>% pull(mean_5)
    message(glue("  {idx}: Protected={row_prot} | Unprotected={row_unpr}"))
  }

  # Save CSV with both numeric and formatted strings
  csv_path <- file.path(OUT_DIR, glue("means_run{sprintf('%02d', rid)}.csv"))
  utils::write.csv(means_tbl, csv_path, row.names = FALSE)

  cap_text <- paste(cap_lines, collapse = "   •   ")

  # Plot with caption
  p <- make_plot(LONG, rid, cap_text)

  # Save plots
  png_path <- file.path(OUT_DIR, glue("violin_indices_run{sprintf('%02d', rid)}.png"))
  ggsave(png_path, p, width = 12, height = 4.8, dpi = 300)

  message(glue("Saved: {png_path}, {csv_path}"))
}

message("Done.")


# ============================
# Combine per-run means CSVs into a single table
# Rows: run (rep_id)
# Cols: NDVI_Protected, NDVI_Unprotected, NDMI_Protected, NDMI_Unprotected,
#       CCI_Protected,  CCI_Unprotected
# ============================

suppressPackageStartupMessages({
  library(dplyr)
  library(tidyr)
  library(fs)
})

OUT_DIR     <-     "C:/Users/david/OneDrive/Documents/Master     ESS/Masterarbeit/Data/SVA
National/Vegetations indexes"

# Find the per-run CSVs
csv_files <- dir(OUT_DIR, pattern = "^means_run\\d+\\.csv$", full.names = TRUE)
if (length(csv_files) == 0L) stop("No per-run means CSVs found in OUT_DIR.")

# Helper: extract run id from filename (e.g., means_run07.csv -> 7)
get_run_id <- function(path) {
  b <- basename(path)
  as.integer(sub("^means_run(\\d+)\\.csv$", "\\1", b))
}

# Read, tag with run id, and bind
all_means <- do.call(rbind, lapply(csv_files, function(p) {
  df <- utils::read.csv(p, stringsAsFactors = FALSE)
  df$run <- get_run_id(p)
  df
}))

# Wide table with numeric means rounded to 5 decimals
wide_tbl <- all_means %>%
  mutate(col = paste(index, group, sep = "_"),
         mean5 = round(mean, 5)) %>%
  select(run, col, mean5) %>%
  tidyr::pivot_wider(names_from = col, values_from = mean5) %>%
  arrange(run)
```

```
# Ensure the column order you want
desired_cols <- c(
  "NDVI_Protected","NDVI_Unprotected",
  "NDMI_Protected","NDMI_Unprotected",
  "CCI_Protected","CCI_Unprotected"
)

# Add any missing columns (if a run lacks a group/index by chance)
for (cc in desired_cols) if (!cc %in% names(wide_tbl)) wide_tbl[[cc]] <- NA_real_

wide_tbl <- wide_tbl %>%
  select(run, all_of(desired_cols))

# Save
out_csv <- file.path(OUT_DIR, "means_all_runs.csv")
utils::write.csv(wide_tbl, out_csv, row.names = FALSE)

message(sprintf("Saved combined table: %s", out_csv))


# ========================================
# ADD-ON: overlay all runs as thin lines
# place this AFTER your for-loop that saves per-run plots
# uses objects already defined: ds, rep_ids, COLS, OUT_DIR
# ========================================

suppressPackageStartupMessages({
  library(purrr)
})

message("Building overlaid density lines across runs...")

# collect densities per run without keeping all raw pixels in memory
dens_list <- list()
eps <- 1e-9
TOT_RUNS <- length(rep_ids)

for (rid in rep_ids) {
  # compute indices for this run (same formulas you used)
  idx_tbl <- ds %>%
    filter(rep_id == rid) %>%
    transmute(
      group,
      NDVI = (B8A - B04) / (B8A + B04 + eps),
      CCI  = ((B8A/(B05 + eps) - 1) + (B8A/(B06 + eps) - 1) + (B8A/(B07 + eps) - 1)) / 3,
      NDMI = (B8A - B11) / (B8A + B11 + eps)
    ) %>%
    collect()

  if (nrow(idx_tbl) == 0) next

  long <- idx_tbl |>
    tidyr::pivot_longer(c(NDVI, NDMI, CCI), names_to = "index", values_to = "val") |>
    filter(is.finite(val), !is.na(group)) |>
    mutate(
      index = factor(index, levels = c("NDVI","NDMI","CCI")),
      group = factor(group, levels = c("Protected","Unprotected"))
    )

  # density per index × group (no quantile clipping — full tails)
  dens_df <- long %>%
    group_by(index, group) %>%
    reframe({
      d <- density(val, na.rm = TRUE, n = 512)  # adjust=... if you want smoother/sharper
      tibble(run_id = rid, x = d$x, density = d$y)
```

```
    }) %>%
    ungroup() %>%
    group_by(index, group, run_id) %>%
    mutate(density_scaled = density / (max(density, na.rm = TRUE) + 1e-12)) %>%
    ungroup()

  dens_list[[length(dens_list) + 1L]] <- dens_df
}

DENS <- dplyr::bind_rows(dens_list)
stopifnot(nrow(DENS) > 0)

# ---------- Plot 1: simple overlaid density lines ----------
p_overlaid <- ggplot(DENS, aes(x = x, y = density_scaled,
                               group = interaction(run_id, group),
                               color = group)) +
  geom_line(linewidth = 0.15, alpha = 0.15) +
  facet_wrap(~ index, scales = "free_x", ncol = 3) +
  scale_color_manual(values = COLS) +
  labs(
    title = "Overlaid kernel density lines across runs",
    subtitle = "Each thin line is one run; densities scaled to peak=1 within index × group",
    x = "Index value",
    y = "Scaled density"
  ) +
  theme_minimal(base_size = 12) +
  theme(
    panel.grid.minor = element_blank(),
    strip.text = element_text(face = "bold"),
    strip.background = element_rect(fill = "#F3F4F6", color = NA),
    legend.position = "top"
  )

ggsave(file.path(OUT_DIR, "overlay_density_lines.png"),
       p_overlaid, width = 12, height = 4.8, dpi = 300)

# ---------- Plot 2 (optional): “true violin-outline overlay” ----------
# This projects the densities sideways around group positions (like violin outlines).
DO_VIOLIN_OUTLINE_OVERLAY <- TRUE

if (DO_VIOLIN_OUTLINE_OVERLAY) {
  # ======= PLACEHOLDERS (edit these) =======
  TITLE_TEXT    <- "Consistency of vegetation index distributions across runs"
  SUBTITLE_TEXT <- "Each thin outline represents one sampling run (~50k pixels); 20 total.
Sentinel-2 L2A indices; no quantile clipping."
  # =========================================

  # map group factors to x centers (1=Protected, 2=Unprotected)
  group_to_x <- function(g) ifelse(as.character(g) == "Protected", 1, 2)

  # width factor controls half-violin width. smaller = thinner
  HALF_WIDTH <- 0.45

  # build left/right outlines per run, then plot very thin lines
  DENS_V <- DENS |>
    dplyr::mutate(
      x_center = group_to_x(group),
      # project density horizontally (scaled) to emulate violin width
      x_left  = x_center - density_scaled * HALF_WIDTH,
      x_right = x_center + density_scaled * HALF_WIDTH
    )


  p_violin_overlay <- ggplot() +
    # left side
    geom_path(
```

```
      data = DENS_V,
      aes(x = x_left,  y = x, group = interaction(run_id, index, group), color = group),
      linewidth = 0.15, alpha = 0.12
    ) +
    # right side
    geom_path(
      data = DENS_V,
      aes(x = x_right, y = x, group = interaction(run_id, index, group), color = group),
      linewidth = 0.15, alpha = 0.12
    ) +
    # enforce per-facet y ranges
    geom_blank(data = LIMITS, inherit.aes = FALSE, aes(x = x_dummy, y = y_low))  +
    geom_blank(data = LIMITS, inherit.aes = FALSE, aes(x = x_dummy, y = y_high)) +
    scale_color_manual(values = COLS) +
    scale_x_continuous(
      breaks = c(1, 2),
      labels = c("Protected", "Unprotected"),
      expand = expansion(mult = c(0.08, 0.08))
    ) +
    # keep per-facet y scales but constrained by LIMITS
    facet_wrap(~ index, ncol = 3, scales = "free_y") +
    labs(
      title = TITLE_TEXT,
      subtitle = SUBTITLE_TEXT,
      x = NULL,
      y = "Index value"
    ) +
    theme_minimal(base_size = 12) +
    theme(
      panel.grid.minor = element_blank(),
      strip.text = element_text(face = "bold"),
      strip.background = element_rect(fill = "#F3F4F6", color = NA),
      legend.position = "top"
    )

  ggsave(file.path(OUT_DIR, "overlay_violin_outlines.png"),
         p_violin_overlay, width = 12, height = 5.2, dpi = 300)
}


message("Overlay plots saved in: ", OUT_DIR)
```

```
# ============================
# SVA — PASS Klustering (10-BAND): Single-run 10D clustering (k-means)
# ============================

suppressPackageStartupMessages({
  library(arrow)
  library(dplyr)
  library(data.table)
  library(fs)
})
# ---------- CONFIG ----------
BANK_BASE_DIR <- "C:/Users/david/OneDrive/Documents/Master ESS/Masterarbeit/Data/Sentinel
2 Bulk/_SVA_BANK_REPS_20250818_213305"
BANK_DIR      <- file.path(BANK_BASE_DIR, "bank")
OUT_DIR       <- file.path(BANK_BASE_DIR, "PASS6_CLUSTER_10B_ALL")
dir_create(OUT_DIR, recurse = TRUE)

BAND_IDS <- c("B01","B02","B03","B04","B05","B06","B07","B8A","B11","B12")

K_CLUSTERS            <- 100L
TRAIN_PER_GROUP_MAX   <- 30000L
EVAL_PER_GROUP_MAX    <- 50000L
TARGET_PER_RUN        <- 50000L
```

```
set.seed(42)

# ---------- helpers ----------
open_bank <- function(dir) open_dataset(dir, format = "parquet")
zscore_fit  <-  function(X){  mu  <-  colMeans(X,  na.rm=TRUE);  sdv  <-  apply(X,2,sd);
sdv[sdv==0|!is.finite(sdv)] <- 1; list(mu=mu, sd=sdv) }
zscore_apply <- function(X, pars){ sweep(sweep(X,2,pars$mu,"-"), 2, pars$sd, "/") }
predict_kmeans <- function(X, centers){
  xsq <- rowSums(X^2); csq <- rowSums(centers^2); G <- X %*% t(centers)
  D2  <- matrix(xsq, nrow=nrow(G), ncol=ncol(G)) +
    matrix(csq, nrow=nrow(G), ncol=ncol(G), byrow=TRUE) - 2*G
  max.col(-D2, ties.method="first")
}

# ---------- fixed run to match PASS-A ----------
REP_ID <- 12L
message(sprintf("Selected run (fixed): %02d", REP_ID))

# composition metrics on proportions p (vector)
hill_numbers <- function(p) {
  p <- p[p > 0]
  if (length(p) == 0) return(list(S = 0, H = NA_real_, D1 = NA_real_, D2 = NA_real_, J =
NA_real_))
  S  <- length(p)
  H  <- -sum(p * log(p))
  D1 <- exp(H)
  D2 <- 1 / sum(p^2)
  J  <- if (S > 1) H / log(S) else NA_real_
  list(S=S, H=H, D1=D1, D2=D2, J=J)
}

# dissimilarities between two compositions
bc_distance <- function(p, q) 0.5 * sum(abs(p - q))
js_divergence <- function(p, q, eps=1e-12) {
  p <- p + eps; q <- q + eps; p <- p/sum(p); q <- q/sum(q)
  m <- 0.5*(p+q)
  0.5*(sum(p*log(p/m)) + sum(q*log(q/m)))
}
hellinger_distance <- function(p, q) {
  v <- sqrt(p) - sqrt(q); (1 / sqrt(2)) * sqrt(sum(v*v))
}

# ---------- load bank for that run ----------
ds <- open_bank(BANK_DIR)
stopifnot(all(BAND_IDS %in% colnames(ds)))
stopifnot(all(c("rep_id","group") %in% colnames(ds)))

ds_r <- ds %>% filter(rep_id == REP_ID) %>% select(all_of(c("group", BAND_IDS)))
n_r  <- ds_r %>% summarise(n = n()) %>% collect() %>% dplyr::pull(n)
if (is.na(n_r) || n_r == 0) stop("Selected run has zero rows.")

df <- as.data.frame(ds_r %>% collect())

take_g <- function(df, g, cap){
  i <- which(df$group == g)
  if (!length(i)) return(df[0,,drop=FALSE])
  if (length(i) > cap) i <- sample(i, cap)
  df[i,,drop=FALSE]
}

# Build TRAIN (balanced, per-group cap)
df_tr <- dplyr::bind_rows(
  take_g(df, "Protected",   TRAIN_PER_GROUP_MAX),
  take_g(df, "Unprotected", TRAIN_PER_GROUP_MAX)
)
```

```
# Build EVAL pool (balanced, per-group cap) → then rarefy equal
df_ev <- dplyr::bind_rows(
  take_g(df, "Protected",   EVAL_PER_GROUP_MAX),
  take_g(df, "Unprotected", EVAL_PER_GROUP_MAX)
)


# ---------- z-score on TRAIN; fit k-means; assign EVAL, then rarefy equal (MATCH PASS-A)
----------

# Keep only complete rows in the full df
X_all <- as.matrix(df[, BAND_IDS, drop = FALSE])
keep  <- stats::complete.cases(X_all)
df    <- df[keep, , drop = FALSE]

# Fit z-score on TRAIN; apply to TRAIN and EVAL
X_tr <- as.matrix(df_tr[, BAND_IDS, drop = FALSE])
X_ev <- as.matrix(df_ev[, BAND_IDS, drop = FALSE])

zs_pars <- zscore_fit(X_tr)
Z_tr    <- zscore_apply(X_tr, zs_pars)
Z_ev    <- zscore_apply(X_ev, zs_pars)

set.seed(42)
km <- kmeans(Z_tr, centers = K_CLUSTERS, iter.max = 100, nstart = 5)

# Assign clusters to the EVAL pool
cl_ev <- predict_kmeans(Z_ev, km$centers)
df_ev$cluster <- cl_ev

# Rarefy EVAL to equal N per group for fair composition (PASS-A behavior)
rarefy_equal <- function(df, group_col="group") {
  tab <- table(df[[group_col]])
  if (length(tab) < 2) return(df[0,,drop=FALSE])
  n_take <- min(as.integer(tab))
  idx <- unlist(tapply(seq_len(nrow(df)), df[[group_col]],
                       function(i) sample(i, n_take, replace = FALSE)))
  df[idx,,drop=FALSE]
}
df_eq <- rarefy_equal(df_ev, "group")
if (nrow(df_eq) == 0) stop("Only one group present after rarefaction.")

# proportions per group on clusters 1..K
tab <- as.data.table(df_eq)[, .N, by = .(group, cluster)]
all_cl <- seq_len(K_CLUSTERS)
get_prop <- function(g) {
  v <- integer(K_CLUSTERS); names(v) <- all_cl
  tg <- tab[group == g]
  if (nrow(tg)) v[as.integer(tg$cluster)] <- tg$N
  v / sum(v)
}
pP <- get_prop("Protected")
pU <- get_prop("Unprotected")

# ---------- metrics ----------
metP <- hill_numbers(pP)
metU <- hill_numbers(pU)

metrics_group <- data.table(
  metric = c("S","H","D1","D2","J"),
  Protected   = c(metP$S, metP$H, metP$D1, metP$D2, metP$J),
  Unprotected = c(metU$S, metU$H, metU$D1, metU$D2, metU$J)
)

metrics_between <- data.table(
  bray_curtis    = bc_distance(pP, pU),
```

```
  jensen_shannon = js_divergence(pP, pU),
  hellinger      = hellinger_distance(pP, pU)
)

# per-cluster table (counts & proportions)
countsP <- as.integer(round(pP * sum(tab[group=="Protected"]$N)))
countsU <- as.integer(round(pU * sum(tab[group=="Unprotected"]$N)))
clusters_tbl <- data.table(
  cluster = all_cl,
  prop_Protected   = pP,
  prop_Unprotected = pU,
  count_Protected  = countsP,
  count_Unprotected= countsU
)

# ---------- save outputs ----------
RUN_DIR <- file.path(OUT_DIR, sprintf("RUN_%02d", REP_ID))
dir_create(RUN_DIR, recurse = TRUE)

# model + preprocessing
saveRDS(list(centers = km$centers,
             zs_mu = zs_pars$mu,
             zs_sd = zs_pars$sd,
             bands = BAND_IDS,
             K = K_CLUSTERS,
             rep_id = REP_ID),
        file = file.path(RUN_DIR, "kmeans_model_zscore.rds"))

fwrite(metrics_group,   file.path(RUN_DIR, "metrics_group.csv"))
fwrite(metrics_between, file.path(RUN_DIR, "metrics_between.csv"))
fwrite(clusters_tbl,    file.path(RUN_DIR, "cluster_composition.csv"))

# small README
writeLines(c(
  "# PASS 6: single-run 10D clustering (k-means)",
  sprintf("Selected run: %02d (fixed selection)", REP_ID),
  sprintf("Bands: %s", paste(BAND_IDS, collapse=", ")),
  sprintf("K clusters: %d", K_CLUSTERS),
  sprintf("Training per group (max): %d", TRAIN_PER_GROUP_MAX),
  "Z-scoring per band performed on the selected run.",
  "Per-group composition computed after rarefying to equal N."
), file.path(RUN_DIR, "README.txt"))


# console summary
cat("\n=== PASS 6 SUMMARY (single run, 10-band) ===\n")
print(metrics_group)
cat("\nBetween-group distances:\n")
print(metrics_between)
message("\nOutputs in: ", RUN_DIR)

# ===== PASS 6 — VISUALS (place at the very end) =====
if (!requireNamespace("ggplot2", quietly=TRUE)) install.packages("ggplot2")
if (!requireNamespace("tidyr",   quietly=TRUE)) install.packages("tidyr")
if (!requireNamespace("scales",  quietly=TRUE)) install.packages("scales")
if (!requireNamespace("gridExtra",quietly=TRUE)) install.packages("gridExtra")
library(ggplot2); library(tidyr); library(scales); library(gridExtra)

# ---- 1) Per-group metrics (dumbbell) ----
label_map <- c(
  S  = "Richness (types)",
  H  = "Shannon entropy (nats)",
  D1 = "Diversity (Hill q=1)",
  D2 = "Dominance-weighted diversity (Hill q=2)",
  J  = "Evenness (Pielou’s J)"
)
```

```
mg <- data.table::copy(metrics_group)
mg[, `:=`(
  metric_lab = factor(label_map[metric], levels = label_map[c("S","H","D1","D2","J")]),
  delta      = Unprotected - Protected,
  delta_pct  = 100 * (Unprotected/Protected - 1)
)]
mg[, x_lab := pmax(Protected, Unprotected) + 0.05*abs(delta)]
mg[, delta_txt := ifelse(metric %in% c("J","H"),
                         sprintf("Δ = %.3f (%+.1f%%)", delta, delta_pct),
                         sprintf("Δ  =  %s  (%+.1f%%)",  comma(delta,  accuracy  =  0.1),
delta_pct))]

db_long <- mg |>
  dplyr::select(metric_lab, Protected, Unprotected) |>
  tidyr::pivot_longer(c(Protected, Unprotected), names_to = "group", values_to = "value")

p_db <- ggplot() +
  geom_segment(data = mg,
               aes(y = metric_lab, yend = metric_lab,
                   x = Protected,  xend = Unprotected),
               linewidth = 0.6, color = "grey65") +
  geom_point(data = subset(db_long, group == "Protected"),
             aes(x = value, y = metric_lab), shape = 21, fill = "white", size = 2.2) +
  geom_point(data = subset(db_long, group == "Unprotected"),
             aes(x = value, y = metric_lab), shape = 16, size = 2.2) +
  geom_text(data = mg,
            aes(x = x_lab, y = metric_lab, label = delta_txt),
            hjust = 0, size = 3.1) +
  labs(
    title   = sprintf("Per-group composition metrics — single run %02d (10-band)", REP_ID),
    subtitle = "Open circle = Protected; filled circle = Unprotected; grey line connects",
    x = NULL, y = NULL
  ) +
  theme_minimal(base_size = 12) +
  theme(panel.grid.minor = element_blank(),
        plot.title = element_text(hjust = 0.5))
print(p_db)
ggsave(file.path(RUN_DIR, "plot_group_metrics_dumbbell.png"),
       p_db, width = 9, height = 5, dpi = 300)

# ---- 2) Between-group separability (single-run values) ----
sep_long <- metrics_between |>
  tidyr::pivot_longer(dplyr::everything(), names_to = "metric", values_to = "value") |>
  dplyr::mutate(metric_lab = factor(metric,
                                    levels                                                    =
c("bray_curtis","jensen_shannon","hellinger"),
                                    labels = c("Bray–Curtis dissimilarity",
                                               "Jensen–Shannon divergence",
                                               "Hellinger distance")))

p_sep <- ggplot(sep_long, aes(x = value, y = metric_lab)) +
  geom_point(size = 2.2) +
  scale_x_continuous(limits = c(0, 1)) +
  labs(
    title = "Between-group separability (10-band, 10-D clusters)",
    subtitle = "Single run — dot = observed value",
    x = "Distance / divergence (0–1 scale)", y = NULL
  ) +
  theme_minimal(base_size = 12) +
  theme(panel.grid.minor = element_blank(),
        plot.title = element_text(hjust = 0.5))
print(p_sep)
ggsave(file.path(RUN_DIR, "plot_between_group_separability.png"),
       p_sep, width = 8, height = 4.5, dpi = 300)

# ---- 3) Top cluster differences with NDVI & red-edge deltas ----
```

```
# Back-transform centers from z to original reflectance scale
# ---- 3) Top cluster differences with NDVI / CCI / NDMI ----
# Back-transform centers from z to original reflectance scale
centers_z <- km$centers
colnames(centers_z) <- BAND_IDS
centers <- sweep(centers_z, 2, zs_pars$sd, "*")
centers <- sweep(centers, 2, zs_pars$mu, "+")

centers_dt <- as.data.table(centers)
centers_dt[, cluster := .I]
setcolorder(centers_dt, c("cluster", BAND_IDS))

# Convert to 0–1 reflectance units (if DN ~ reflectance*10000)
to01 <- function(v) v / 10000
centers_dt[, (BAND_IDS) := lapply(.SD, to01), .SDcols = BAND_IDS]

# Indices per cluster
eps <- .Machine$double.eps
centers_dt[, ndvi := (B8A - B04) / pmax(B8A + B04, eps)]
centers_dt[, `:=`(
  cci5 = (B8A / pmax(B05, eps)) - 1,
  cci6 = (B8A / pmax(B06, eps)) - 1,
  cci7 = (B8A / pmax(B07, eps)) - 1
)]
centers_dt[, cci := (cci5 + cci6 + cci7) / 3]
centers_dt[, ndmi := (B8A - B11) / pmax(B8A + B11, eps)]

# Δ proportions and Bray–Curtis shares (same as before)
cl_all <- as.data.table(clusters_tbl)[, .(cluster, prop_Unprotected, prop_Protected)]
cl_all[, delta := prop_Unprotected - prop_Protected]
bc_total <- 0.5 * sum(abs(cl_all$delta))
cl_all[, bc_share := if (bc_total > 0) 100 * (0.5 * abs(delta)) / bc_total else 0]

# Top-20 by |Δ|
top <- merge(cl_all, centers_dt, by = "cluster")[order(-abs(delta))][1:20]
top[, cluster := factor(cluster, levels = rev(cluster))]

# Main Δ plot (color by NDVI; change to cci/ndmi if you prefer)
library(ggplot2); library(scales)
range_x <- max(abs(top$delta), na.rm = TRUE) * 1.05
p_main <- ggplot(top, aes(x = cluster, y = delta)) +
  geom_hline(yintercept = 0, linetype = 2, color = "grey70") +
  geom_segment(aes(xend = cluster, y = 0, yend = delta),
               linewidth = 0.6, color = "grey70") +
  geom_point(aes(color = ndvi), size = 2) +
  coord_flip() +
  scale_y_continuous(labels = percent_format(accuracy = 0.1),
                     limits = c(-range_x, range_x),
                     expand = expansion(mult = c(0.05, 0.05))) +
  scale_color_gradient(low = "#9d6b53", high = "#1a7f37", name = "NDVI", guide = "none") +
  labs(title = "Top 20 cluster proportion differences (10 bands)",
       subtitle = "Δ = Unprotected − Protected",
       x = "Cluster ID", y = "Δ proportion") +
  theme_minimal(base_size = 12) +
  theme(panel.grid.minor = element_blank(),
        plot.title = element_text(hjust = 0.5),
        legend.position = "bottom")

# Side labels panel with NDVI, CCI, NDMI and BC contribution
lab_df <- data.frame(
  cluster   = top$cluster,
  ndvi      = top$ndvi,
  cci       = top$cci,
  ndmi      = top$ndmi,
  label_all = sprintf(
    "NDVI %.2f | CCI %.2f | NDMI %.2f | %.1f%% BC",
```

```
    top$ndvi, top$cci, top$ndmi, top$bc_share)
)

p_side <- ggplot(lab_df, aes(y = cluster)) +
  geom_point(aes(x = 0.03, color = ndvi), size = 2.4) +
  geom_text(aes(x = 0.07, label = label_all), hjust = 0, size = 3.1, color = "grey20") +
  scale_x_continuous(limits = c(0, 1), expand = c(0, 0)) +
  scale_color_gradient(low = "#9d6b53", high = "#1a7f37", guide = "none") +
  theme_void() +
  theme(plot.margin = margin(t = 28, r = 12, b = 26, l = 0))
# Build the wide two-panel plot once; we reuse it below
g_combined <- gridExtra::arrangeGrob(p_main, p_side, ncol = 2, widths = c(0.60, 0.40))
ggsave(file.path(RUN_DIR, "plot_cluster_top_differences_wide_indices.png"),
       g_combined, width = 12, height = 7, dpi = 300)

# --- Context table: min / median / max for NDVI, CCI, NDMI (all K vs top-20) ---
safemin   <- function(x) min(x, na.rm = TRUE)
safemed   <- function(x) median(x, na.rm = TRUE)
safemax   <- function(x) max(x, na.rm = TRUE)

# summaries across ALL cluster centers
summ_all <- data.frame(
  Metric = c("NDVI","CCI","NDMI"),
  Min_allK    = c(safemin(centers_dt$ndvi),
                  safemin(centers_dt$cci),
                  safemin(centers_dt$ndmi)),
  Median_allK = c(safemed(centers_dt$ndvi),
                  safemed(centers_dt$cci),
                  safemed(centers_dt$ndmi)),
  Max_allK    = c(safemax(centers_dt$ndvi),
                  safemax(centers_dt$cci),
                  safemax(centers_dt$ndmi))
)

# summaries within the TOP-20 clusters shown in the plot
summ_top <- data.frame(
  Metric = c("NDVI","CCI","NDMI"),
  Min_top20    = c(safemin(top$ndvi),
                   safemin(top$cci),
                   safemin(top$ndmi)),
  Median_top20 = c(safemed(top$ndvi),
                   safemed(top$cci),
                   safemed(top$ndmi)),
  Max_top20    = c(safemax(top$ndvi),
                   safemax(top$cci),
                   safemax(top$ndmi))
)

ctx_tbl <- merge(summ_all, summ_top, by = "Metric")
# nice rounding
num_cols <- setdiff(names(ctx_tbl), "Metric")
ctx_tbl[num_cols] <- lapply(ctx_tbl[num_cols], function(x) round(x, 3))

# save the numbers for reporting
data.table::fwrite(ctx_tbl, file.path(RUN_DIR, "top20_context_min_median_max.csv"))

# render as a small table and stack under your wide plot
if (!requireNamespace("gridExtra", quietly = TRUE)) install.packages("gridExtra")
library(gridExtra)

tg <- gridExtra::tableGrob(
  ctx_tbl, rows = NULL,
  theme = gridExtra::ttheme_minimal(
    base_size = 9,
    core    = list(fg_params = list(hjust = 1, x = 0.98)),
    colhead = list(fg_params = list(fontface = "bold"))
```

```
  )
)

# combine: existing wide plot (g_combined) + context table
g_final <- gridExtra::arrangeGrob(
  g_combined, tg,
  nrow = 2, heights = c(0.78, 0.22)
)

# save the new version (keep your original too if you like)
ggsave(file.path(RUN_DIR, "plot_cluster_top_differences_wide_with_context.png"),
       g_final, width = 12, height = 8, dpi = 300)


# ==== Heatmap of NDVI / CCI / NDMI per cluster, columns sorted by |Δ proportion| ====
suppressPackageStartupMessages({library(data.table);  library(ggplot2);  library(scales);
library(gridExtra); library(grid)})

# 1) Δ per cluster and sorting
cld <- as.data.table(clusters_tbl)[, .(cluster,
                                       delta = prop_Unprotected - prop_Protected)]
# sort by absolute difference (largest to smallest).
# If you prefer signed order, use order(delta) instead.
cl_order <- cld[order(-abs(delta))]$cluster
cld[, cluster_f := factor(cluster, levels = cl_order)]

# 2) Long table of center indices
M <- merge(as.data.table(centers_dt)[, .(cluster, ndvi, cci, ndmi)], cld, by = "cluster")
M_long <- melt(M, id.vars = c("cluster","cluster_f","delta"),
               variable.name = "metric", value.name = "value")

# z-score within each metric so the heatmap uses a common color scale
M_long[, value_z := (value - mean(value, na.rm=TRUE)) / sd(value, na.rm=TRUE), by = metric]

# 3) Δ bar on top
p_delta <- ggplot(cld, aes(x = cluster_f, y = delta)) +
  geom_col(width = 0.85, fill = "grey55") +
  geom_hline(yintercept = 0, linetype = 2, color = "grey50") +
  scale_y_continuous(labels = percent_format(accuracy = 0.1)) +
  labs(title = "Δ proportion (Unprotected − Protected)",
       x = NULL, y = NULL) +
  theme_minimal(base_size = 11) +
  theme(axis.text.x = element_blank(),
        axis.ticks.x = element_blank(),
        panel.grid.minor = element_blank(),
        plot.title = element_text(size = 11, hjust = 0))

# 4) Heatmap of standardized index values
metric_labels <- c(ndvi = "NDVI", cci = "CCI", ndmi = "NDMI")
p_heat   <-   ggplot(M_long,   aes(x   =   cluster_f,   y   =   factor(metric,   levels   =
c("ndvi","cci","ndmi"),
                                                          labels  =  metric_labels),  fill  =
value_z)) +
  geom_tile() +
  scale_fill_gradient2(name = "z-score", low = "#3b5b92", mid = "white", high = "#1a7f37")
+
  labs(title = "Cluster-center index values (standardized within metric)",
       x = "Clusters (sorted by |Δ|)", y = NULL) +
  theme_minimal(base_size = 11) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, size = 7),
        panel.grid = element_blank(),
        plot.title = element_text(size = 11, hjust = 0))
library(grid)
library(gridExtra)
library(ggplot2)
```

```
# 1) move the heatmap legend to the bottom (horizontal bar)
p_heat2 <- p_heat +
  theme(
    legend.position = "bottom",
    legend.direction = "horizontal",
    legend.box = "horizontal",
    legend.margin = margin(t = 4, r = 0, b = 0, l = 0),
    plot.margin = margin(t = 6, r = 6, b = 6, l = 6)
  ) +
  guides(
    fill = guide_colorbar(
      title.position = "top",
      barwidth = unit(6, "cm"),
      barheight = unit(0.35, "cm")
    )
  )

# --- Align & stack without layout errors (recommended) ---
suppressPackageStartupMessages({ library(cowplot) })

# 1) Legend at bottom (you already built p_heat2 above)
#    (If not yet, keep your p_heat2 code exactly as-is.)

# 2) Align the LEFT axes between the top-left plot (p_main) and the heatmap (p_heat2)
aligned <- cowplot::align_plots(p_main, p_heat2, align = "v", axis = "l")

# 3) Rebuild the top row with the already-aligned p_main + your side panel
top_row <- cowplot::plot_grid(aligned[[1]], p_side, ncol = 2, rel_widths = c(0.60, 0.40))

# 4) Stack top row over the aligned heatmap
final   <- cowplot::plot_grid(top_row, aligned[[2]], ncol = 1, rel_heights = c(0.65, 0.35))

# 5) Draw & save
print(final)
ggsave(file.path(RUN_DIR, "plot_top_deltas_plus_heatmap_aligned.png"),
       final, width = 12, height = 7.5, dpi = 300)


# 5) Stack the two plots (top = Δ, bottom = heatmap) and save
g_final <- gridExtra::arrangeGrob(p_delta, p_heat, ncol = 1, heights = c(0.9, 4))
grid::grid.newpage(); grid::grid.draw(g_final)

outfile <- file.path(RUN_DIR, "heatmap_indices_by_cluster_sorted_absDelta.png")
ggsave(outfile, g_final, width = 12, height = 7.5, dpi = 300)
message("Saved: ", outfile)

suppressPackageStartupMessages({
  library(data.table); library(dplyr); library(tidyr); library(ggplot2); library(scales)
})

# --- helpers (re-use if you already defined them) ---
calc_ndvi <- function(b8a, b04, eps = .Machine$double.eps) (b8a - b04) / pmax(b8a + b04,
eps)
calc_cci_three <- function(b8a, b05, b06, b07, eps = .Machine$double.eps) {
  cci5 <- (b8a / pmax(b05, eps)) - 1
  cci6 <- (b8a / pmax(b06, eps)) - 1
  cci7 <- (b8a / pmax(b07, eps)) - 1
  (cci5 + cci6 + cci7) / 3
}
calc_ndmi <- function(b8a, b11, eps = .Machine$double.eps) (b8a - b11) / pmax(b8a + b11,
eps)

# 1) Δ proportion per cluster to get the sorting
cld <- as.data.table(clusters_tbl)[, .(cluster, delta = prop_Unprotected - prop_Protected)]
cl_order <- cld[order(-abs(delta))]$cluster   # sort by |Δ|
cld[, cluster_f := factor(cluster, levels = cl_order)]
```

```
# 2) Compute per-cluster, per-group *actual index values* (means)
# --- Use the rarefied evaluation set, which HAS 'cluster' ---
DF_src <- df_eq   # or df_ev if you prefer not to rarefy

DT <- as.data.table(DF_src)
DT <- DT[!is.na(group) & !is.na(cluster) & is.finite(B8A) & is.finite(B04) & is.finite(B11)
         & is.finite(B05) & is.finite(B06) & is.finite(B07)]

# Indices per point (ratios; no need to divide by 10000)
DT[, `:=`(
  NDVI = calc_ndvi(B8A, B04),
  CCI  = calc_cci_three(B8A, B05, B06, B07),
  NDMI = calc_ndmi(B8A, B11)
)]

# Per-cluster, per-group means of each index
idx_means <- DT[, .(
  NDVI = mean(NDVI, na.rm = TRUE),
  CCI  = mean(CCI,  na.rm = TRUE),
  NDMI = mean(NDMI, na.rm = TRUE)
), by = .(cluster, group)]

# Long format + merge sort order (unchanged)
idx_long <- melt(idx_means, id.vars = c("cluster","group"),
                 variable.name = "metric", value.name = "value")
idx_long <- merge(idx_long, cld[, .(cluster, cluster_f)], by = "cluster", all.x = TRUE)
idx_long$metric <- factor(idx_long$metric, levels = c("NDVI","CCI","NDMI"),
                          labels = c("NDVI","CCI (mean of B05/06/07)","NDMI"))


# 3) Line plot across clusters (x = sorted cluster index), colored by group
p_lines <- ggplot(idx_long,
                  aes(x = as.integer(cluster_f), y = value, color = group, group = group))
+
  geom_hline(yintercept = 0, linetype = 2, color = "grey75") +
  geom_line(linewidth = 0.7, alpha = 0.9, na.rm = TRUE) +
  geom_point(size = 1.2, alpha = 0.9, na.rm = TRUE) +
  facet_wrap(~ metric, ncol = 1, scales = "fixed") +
  scale_x_continuous(
    breaks = seq(1, length(cl_order), by = 5),
    labels = function(i) as.character(cl_order[i])
  ) +
  coord_cartesian(ylim = c(-1, 1)) +
  scale_color_manual(values = c(Protected = "#2c7fb8", Unprotected = "#d95f0e")) +
  labs(
    title = "Cluster-wise index values by group (clusters sorted by |Δ proportion|)",
    subtitle = "Lines show per-cluster means of NDVI / CCI / NDMI for each group",
    x = "Cluster ID (sorted)", y = "Index value (−1 … 1)", color = "Group"
  ) +
  theme_minimal(base_size = 12) +
  theme(panel.grid.minor = element_blank(),
        axis.text.x = element_text(angle = 90, vjust = 0.5, size = 7),
        strip.text = element_text(face = "bold"))

print(p_lines)

# Save
outfile <- file.path(RUN_DIR, "lines_indices_by_cluster_group_sorted_absDelta.png")
ggsave(outfile, p_lines, width = 12, height = 8.5, dpi = 300)
message("Saved: ", outfile)

library(gtable)
library(gridExtra)
library(grid)
```

```
# Legend-bottom heatmap (keep your p_heat2 creation)
g_main <- ggplotGrob(p_main)
g_heat <- ggplotGrob(p_heat2)

# Find the panel column in each grob
panel_col_main <- unique(subset(g_main$layout, name == "panel")$l)
panel_col_heat <- unique(subset(g_heat$layout, name == "panel")$l)

# Make the left-hand area (everything left of the panel) the same width
left_idx_main <- seq_len(panel_col_main - 1)
left_idx_heat <- seq_len(panel_col_heat - 1)
common <- seq_len(min(length(left_idx_main), length(left_idx_heat)))
g_main$widths[left_idx_main[common]] <- unit.pmax(g_main$widths[left_idx_main[common]],
                                                  g_heat$widths[left_idx_heat[common]])
g_heat$widths[left_idx_heat[common]] <- g_main$widths[left_idx_main[common]]

# Rebuild the top row and stack
g_top <- arrangeGrob(g_main, p_side, ncol = 2, widths = c(0.60, 0.40))

grid.newpage()
grid.arrange(g_top, g_heat, ncol = 1, heights = c(0.65, 0.35))

ggsave(file.path(RUN_DIR, "plot_top_deltas_plus_heatmap_aligned.png"),
       arrangeGrob(g_top, g_heat, ncol = 1, heights = c(0.65, 0.35)),
       width = 12, height = 7.5, dpi = 300)


# put this near your plotting code
suppressPackageStartupMessages({ library(patchwork); library(grid) })

# make the heatmap use a horizontal legend
p_heat_bottom <- p_heat +
  theme(
    legend.position = "bottom",
    legend.direction = "horizontal",
    legend.box = "horizontal"
  ) +
  guides(fill = guide_colorbar(
    title.position = "top",
    barwidth = unit(6, "cm"),
    barheight = unit(0.35, "cm")
  ))

# stack Δ bar (top) + heatmap (bottom), keep panel widths aligned,
# and collect the legend at the bottom
p24 <- (p_delta / p_heat_bottom) +
  plot_layout(heights = c(0.9, 4), guides = "collect")

# draw + save
print(p24)
ggsave(file.path(RUN_DIR, "heatmap_indices_by_cluster_sorted_absDelta_BOTTOMlegend.png"),
       p24, width = 12, height = 7.5, dpi = 300)
```

```
# ============================================================
# Δ proportion bar (sorted by |Δ|) + NDVI/CCI/NDMI
# Heatmap of cluster-center index values (legend at bottom)
# Inputs: PASS6 outputs for a single run:
#   - cluster_composition.csv
#   - kmeans_model_zscore.rds  (has centers in Z, with zs_mu/zs_sd)
# ============================================================

suppressPackageStartupMessages({
  library(data.table); library(ggplot2); library(scales)
  library(grid); library(gridExtra); library(cowplot)
})
```

```
# ---- CONFIG ----
RUN_ID     <- 12L
BANK_BASE   <- "C:/Users/david/OneDrive/Documents/Master ESS/Masterarbeit/Data/Sentinel 2
Bulk/_SVA_BANK_REPS_20250818_213305"
PASS6_DIR  <- file.path(BANK_BASE, "PASS6_CLUSTER_10B_ALL", sprintf("RUN_%02d", RUN_ID))
IN_COMP    <- file.path(PASS6_DIR, "cluster_composition.csv")
IN_MODEL   <- file.path(PASS6_DIR, "kmeans_model_zscore.rds")
OUT_PNG                                                <-            file.path(PASS6_DIR,
"heatmap_indices_by_cluster_sorted_absDelta_BOTTOMlegend.png")

# ---- LOAD ----
stopifnot(file.exists(IN_COMP), file.exists(IN_MODEL))
clu <- fread(IN_COMP)
mod <- readRDS(IN_MODEL)

# ---- Ensure we have proportions in cluster table ----
if (!all(c("prop_Protected", "prop_Unprotected") %in% names(clu))) {
  if (all(c("count_Protected", "count_Unprotected") %in% names(clu))) {
    clu[, prop_Protected   := count_Protected   / sum(count_Protected,   na.rm = TRUE)]
    clu[, prop_Unprotected := count_Unprotected / sum(count_Unprotected, na.rm = TRUE)]
  } else stop("cluster_composition.csv must have prop_* or count_* columns.")
}

# ---- Δ per cluster and sort order (largest |Δ| first) ----
cld <- clu[, .(cluster, delta = prop_Unprotected - prop_Protected)]
cl_order <- cld[order(-abs(delta))]$cluster
cld[, cluster_f := factor(cluster, levels = cl_order)]

# ---- Get cluster centers in original reflectance space ----
# centers_z are in z-score space; back-transform with zs_sd / zs_mu
centers_z <- mod$centers
colnames(centers_z) <- mod$bands
centers <- sweep(centers_z, 2, mod$zs_sd, "*")
centers <- sweep(centers,  2, mod$zs_mu, "+")

centers_dt <- as.data.table(centers)
centers_dt[, cluster := .I]

# ---- Indices from centers (scale cancels out for ratios) ----
eps <- .Machine$double.eps
calc_ndvi <- function(b8a, b04)           (b8a - b04) / pmax(b8a + b04, eps)
calc_cci  <- function(b8a, b05, b06, b07) {
  cci5 <- (b8a / pmax(b05, eps)) - 1
  cci6 <- (b8a / pmax(b06, eps)) - 1
  cci7 <- (b8a / pmax(b07, eps)) - 1
  (cci5 + cci6 + cci7) / 3
}
calc_ndmi <- function(b8a, b11)           (b8a - b11) / pmax(b8a + b11, eps)

# Band names expected from PASS-6 (10-band @20m, omit B08)
BANDS <- c("B01","B02","B03","B04","B05","B06","B07","B8A","B11","B12")
stopifnot(all(BANDS %in% names(centers_dt)))

centers_dt[, `:=`(
  NDVI = calc_ndvi(B8A, B04),
  CCI  = calc_cci (B8A, B05, B06, B07),
  NDMI = calc_ndmi(B8A, B11)
)]

# ---- Long table + merge sort order ----
M <- merge(centers_dt[, .(cluster, NDVI, CCI, NDMI)], cld, by = "cluster")
M_long <- melt(M, id.vars = c("cluster","cluster_f","delta"),
               variable.name = "metric", value.name = "value")

# z-score within each metric (so colors are comparable across rows)
M_long[, value_z := (value - mean(value, na.rm = TRUE)) / sd(value, na.rm = TRUE),
```

```
       by = metric]

# ---- Top panel: Δ bar ----
p_delta <- ggplot(cld, aes(x = cluster_f, y = delta)) +
  geom_col(width = 0.85, fill = "grey55") +
  geom_hline(yintercept = 0, linetype = 2, color = "grey60") +
  scale_y_continuous(labels = percent_format(accuracy = 0.1)) +
  labs(title = "Δ proportion (Unprotected − Protected)", x = NULL, y = NULL) +
  theme_minimal(base_size = 11) +
  theme(axis.text.x = element_blank(),
        axis.ticks.x = element_blank(),
        panel.grid.minor = element_blank(),
        plot.title = element_text(size = 11, hjust = 0))

# ---- Bottom panel: heatmap (legend at bottom) ----
p_heat <- ggplot(M_long, aes(x = cluster_f,
                             y = factor(metric,
                                        levels = c("NDMI","CCI","NDVI"),
                                        labels = c("NDMI","CCI","NDVI")),
                             fill = value_z)) +
  geom_tile() +
  scale_fill_gradient2(name = "z-score",
                       low = "#3b5b92", mid = "white", high = "#1a7f37") +
  labs(x = "Clusters (sorted by |Δ|)", y = NULL) +
  theme_minimal(base_size = 11) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, size = 7),
        panel.grid = element_blank(),
        plot.title = element_text(size = 11, hjust = 0),
        legend.position = "bottom",
        legend.direction = "horizontal",
        legend.box = "horizontal")

# ---- Align & stack with ONE global title ----
suppressPackageStartupMessages({ library(patchwork) })

# remove per-panel titles (we'll add a global one)
p_delta_clean <- p_delta + labs(title = NULL, subtitle = NULL)
p_heat_clean  <- p_heat  + labs(title = NULL, subtitle = NULL)

final <- (p_delta_clean / p_heat_clean) +
  plot_layout(heights = c(0.9, 4), guides = "collect") +
  plot_annotation(
    title = "Spectral index composition of clusters (sorted by Δ proportion)",
    subtitle = "Each column is a k-means cluster; colors show standardized NDVI, CCI, and
NDMI for cluster centers"
  ) &
  theme(plot.title = element_text(hjust = 0.5),
        plot.subtitle = element_text(hjust = 0.5))

# ---- Save ----
ggsave(OUT_PNG, final, width = 12, height = 7.5, dpi = 300)
message("Saved: ", OUT_PNG)
```

```
# ============================
# PASS A — Richness Sensitivity heatmap across 20 runs
# ============================
suppressPackageStartupMessages({
  library(arrow); library(dplyr); library(data.table); library(fs)
  library(ggplot2); library(scales); library(tidyr); library(patchwork)
})

# ---------- CONFIG ----------
BANK_BASE_DIR <- "C:/Users/david/OneDrive/Documents/Master ESS/Masterarbeit/Data/Sentinel
2 Bulk/_SVA_BANK_REPS_20250818_213305"
BANK_DIR      <- file.path(BANK_BASE_DIR, "bank")
OUT_DIR       <- file.path(BANK_BASE_DIR, "PASS_A_RICHNESS_SENS")
```

```
dir_create(OUT_DIR, recurse = TRUE)

# Bands (10 × 20 m; omit B08)
BANDS <- c("B01","B02","B03","B04","B05","B06","B07","B8A","B11","B12")

# Bank scaling (if bank stores DN ≈ reflectance*10000)
SCALE_TO_REFLECTANCE <- TRUE
SCALE_FACTOR         <- 10000

# Sampling caps (balanced by group)
TRAIN_PER_GROUP_MAX <- 30000L   # for fitting k-means
EVAL_PER_GROUP_MAX  <- 50000L   # for evaluating richness

# Sensitivity grids
K_GRID      <- c(40L, 60L, 80L, 100L, 120L, 150L)
THRESH_GRID <- c(0.00, 0.005, 0.01, 0.02, 0.03, 0.05)  # 0%, 0.5%, 1%, 2%, 3%, 5%

# Run ALL 20 replicates
USE_REP <- 1:20

SEED <- 42
set.seed(SEED)

# ---------- Helpers ----------
open_bank <- function(dir) open_dataset(dir, format = "parquet")
zscore_fit     <-  function(X){  mu  <-  colMeans(X,  na.rm=TRUE);  sdv  <-  apply(X,2,sd);
sdv[!is.finite(sdv)|sdv==0] <- 1; list(mu=mu, sd=sdv) }
zscore_apply <- function(X, zs){ sweep(sweep(X,2,zs$mu,"-"), 2, zs$sd, "/") }

predict_kmeans <- function(X, centers){
  xsq <- rowSums(X^2); csq <- rowSums(centers^2); G <- X %*% t(centers)
  D2  <- matrix(xsq, nrow=nrow(G), ncol=ncol(G)) +
    matrix(csq, nrow=nrow(G), ncol=ncol(G), byrow=TRUE) - 2*G
  max.col(-D2, ties.method="first")
}

rarefy_equal <- function(df, group_col="group"){
  tab <- table(df[[group_col]]); if (length(tab) < 2) return(df[0,,drop=FALSE])
  n_take <- min(as.integer(tab))
  idx <- unlist(tapply(seq_len(nrow(df)), df[[group_col]],
                       function(i) sample(i, n_take, replace = FALSE)))
  df[idx,,drop=FALSE]
}

# replace your helper with this
richness_thresholded <- function(p, thr = 0.01, eps = 1e-12) {
  if (thr <= 0) {
    sum(p > eps, na.rm = TRUE)          # τ = 0 → strictly > 0
  } else {
    sum(p >= thr - eps, na.rm = TRUE)   # τ > 0 → standard threshold with tiny tolerance
  }
}

get_props_by_group <- function(df_clusters, K){
  tab <- as.data.table(df_clusters)[, .N, by=.(group, cluster)]
  get_prop <- function(g){
    v <- numeric(K); tg <- tab[group==g]; if (nrow(tg)) v[as.integer(tg$cluster)] <- tg$N
    v / sum(v)
  }
  list(P = get_prop("Protected"), U = get_prop("Unprotected"))
}

# ---------- Load bank once ----------
ds <- open_bank(BANK_DIR)
stopifnot(all(BANDS %in% colnames(ds)))
stopifnot(all(c("rep_id","group") %in% colnames(ds)))
```

```
# (Robust) Use only reps that actually exist in data
REPS <- ds %>% select(rep_id) %>% distinct() %>% collect() %>% `[[`("rep_id") %>%
  intersect(USE_REP) %>% sort()

message("Running replicates: ", paste(REPS, collapse = ", "))

# ---------- Compute richness grid across replicates ----------
RESULTS <- list()

for (r in REPS) {
  message(sprintf("\n=== RICHNESS SENS — replicate %02d ===", r))

  cols_need <- c("group", BANDS)
  df_all <- ds %>% filter(rep_id == r) %>% select(all_of(cols_need)) %>% collect() %>%
as.data.frame()
  if (SCALE_TO_REFLECTANCE) df_all[,BANDS] <- df_all[,BANDS,drop=FALSE] / SCALE_FACTOR
  df_all <- df_all[stats::complete.cases(df_all[,BANDS,drop=FALSE]), , drop=FALSE]

  take_g <- function(df, g, cap){ i <- which(df$group==g); if (!length(i))
return(df[0,,drop=FALSE]); if (length(i)>cap) i <- sample(i,cap); df[i,,drop=FALSE] }
  df_tr <- dplyr::bind_rows(
    take_g(df_all, "Protected",   TRAIN_PER_GROUP_MAX),
    take_g(df_all, "Unprotected", TRAIN_PER_GROUP_MAX)
  )
  df_ev <- dplyr::bind_rows(
    take_g(df_all, "Protected",   EVAL_PER_GROUP_MAX),
    take_g(df_all, "Unprotected", EVAL_PER_GROUP_MAX)
  )
  df_eq <- rarefy_equal(df_ev, "group")
  if (nrow(df_eq) == 0) { warning(sprintf("rep %02d has one group only; skipping.", r));
next }

  X_tr <- as.matrix(df_tr[,BANDS,drop=FALSE]); X_ev <- as.matrix(df_eq[,BANDS,drop=FALSE])
  zs <- zscore_fit(X_tr); Z_tr <- zscore_apply(X_tr, zs); Z_ev <- zscore_apply(X_ev, zs)

  set.seed(SEED) # reproducible across K within replicate
  for (K in K_GRID) {
    km <- kmeans(Z_tr, centers = K, iter.max = 100, nstart = 5)
    df_eq$cluster <- predict_kmeans(Z_ev, km$centers)

    props <- get_props_by_group(df_eq, K)
    pP <- props$P; pU <- props$U

    for (thr in THRESH_GRID) {
      S_P <- richness_thresholded(pP, thr)
      S_U <- richness_thresholded(pU, thr)
      RESULTS[[length(RESULTS)+1L]] <- data.table(
        run = r, K = K, threshold = thr,
        S_Protected = S_P, S_Unprotected = S_U,
        delta = S_U - S_P,
        delta_pct = ifelse(S_P > 0, 100 * (S_U/S_P - 1), NA_real_)
      )
    }
  }
}

# ---- Save raw results (full recompute) ----
RES <- rbindlist(RESULTS, fill = TRUE)
RES_CSV <- file.path(OUT_DIR, "richness_sensitivity_all.csv")
fwrite(RES, RES_CSV)


# keep all old rows except τ=0 (for the K in K_GRID we just recomputed)
RES_merged <- rbindlist(list(
  RES_old[!(threshold == 0.00 & K %in% K_GRID)],
```

```
  RES_patch
), use.names = TRUE, fill = TRUE)

fwrite(RES_merged, RES_CSV)
RES <- RES_merged  # continue to plotting with merged results

# ---- Summarize for heatmap
SUM_stats <- RES[, .(
  mean_delta = mean(delta, na.rm = TRUE),
  sd_delta   = sd(delta,   na.rm = TRUE),
  n          = .N
), by = .(K, threshold)][order(K, threshold)]

SUM_stats[, se := ifelse(n > 1 & is.finite(sd_delta), sd_delta / sqrt(n), NA_real_)]
SUM_stats[, `:=`(
  ci95_lo  = mean_delta - 1.96 * se,
  ci95_hi  = mean_delta + 1.96 * se,
  ci_width = ifelse(is.na(se), NA_real_, 1.96 * se)
)]
SUM_stats[, thr_lab := scales::percent(threshold, accuracy = 0.1)]

N_REPS <- length(unique(RES$run))

p_heat_ci <- ggplot(SUM_stats, aes(x = factor(K), y = thr_lab, fill = mean_delta)) +
  geom_tile() +
  geom_text(aes(label = paste0(
    round(mean_delta, 1),
    "\nCI±", ifelse(is.na(ci_width), "—", round(ci_width, 1)),
    "  SD±", ifelse(is.na(sd_delta), "—", round(sd_delta, 1))
  )), lineheight = 0.95, size = 3) +
  scale_fill_gradient2(
    low = "#6baed6", mid = "white", high = "#fb6a4a",
    midpoint = 0, name = "ΔS (U−P)"
  ) +
  labs(
    title = "ΔS heatmap (mean with CI & SD across replicates)",
    x = "K (clusters)", y = "Presence threshold"
  ) +
  theme_minimal(base_size = 12)

HM_TITLE     <- "Spectral Richness Difference (ΔS) vs. Clustering Size (K) and Presence
Threshold (τ)"
HM_SUBTITLE <- sprintf("Sensitivity of mean ΔS across %d replicates (Unprotected (U) −
Protected (P)) with 95%% CI and SD", N_REPS)

p_heat_ci_panel <- p_heat_ci + plot_annotation(
  title    = HM_TITLE,
  subtitle = HM_SUBTITLE,
  theme = theme(
    plot.background = element_rect(fill = NA, colour = NA),
    plot.title      = element_text(size = 16, face = "bold", hjust = 0.0, margin = margin(b
= 6)),
    plot.subtitle    = element_text(size = 12, colour = "grey20", hjust = 0.0, margin =
margin(b = 6))
  )
)

# ---- Save (transparent) ----
ragg::agg_png(
  filename = file.path(OUT_DIR, "heat_deltaS_panel_annot.png"),
  width = 10, height = 7, units = "in",
  res = 300, background = "transparent"
)
print(p_heat_ci_panel)
dev.off()
```

```
message("Saved CSV: ", RES_CSV)
message("Saved heatmap PNG to: ", file.path(OUT_DIR, "heat_deltaS_panel_annot.png"))
```

```
# ============================
# SUPPLEMENT — PC-space ΔS maps with GLOBAL τ (single replicate)
# ============================
suppressPackageStartupMessages({
  library(arrow); library(dplyr); library(data.table); library(fs)
  library(ggplot2); library(scales); library(tidyr); library(patchwork)
})

# ---------- CONFIG ----------
BANK_BASE_DIR <- "C:/Users/david/OneDrive/Documents/Master ESS/Masterarbeit/Data/Sentinel
2 Bulk/_SVA_BANK_REPS_20250818_213305"
BANK_DIR      <- file.path(BANK_BASE_DIR, "bank")
OUT_DIR       <- file.path(BANK_BASE_DIR, "PASS_A_RICHNESS_SENS")
dir_create(OUT_DIR, recurse = TRUE)

# Bands (10 × 20 m; omit B08)
BANDS <- c("B01","B02","B03","B04","B05","B06","B07","B8A","B11","B12")

# Bank scaling
SCALE_TO_REFLECTANCE <- TRUE
SCALE_FACTOR         <- 10000

# Sampling caps
TRAIN_PER_GROUP_MAX <- 30000L
EVAL_PER_GROUP_MAX  <- 50000L

# PC-space map settings (single replicate)
K_TARGET    <- 100L
REP_FOR_MAP <- 12L
THR_LIST    <- c(`0.5%`=0.005, `2.0%`=0.02, `5.0%`=0.05)
NBINS       <- 100L

SEED <- 42
set.seed(SEED)

# ---------- Helpers ----------
open_bank <- function(dir) open_dataset(dir, format = "parquet")
zscore_fit     <- function(X){ mu <- colMeans(X, na.rm=TRUE); sdv <- apply(X,2,sd);
sdv[!is.finite(sdv)|sdv==0] <- 1; list(mu=mu, sd=sdv) }
zscore_apply <- function(X, zs){ sweep(sweep(X,2,zs$mu,"-"), 2, zs$sd, "/") }

predict_kmeans <- function(X, centers){
  xsq <- rowSums(X^2); csq <- rowSums(centers^2); G <- X %*% t(centers)
  D2  <- matrix(xsq, nrow=nrow(G), ncol=ncol(G)) +
    matrix(csq, nrow=nrow(G), ncol=ncol(G), byrow=TRUE) - 2*G
  max.col(-D2, ties.method="first")
}

rarefy_equal <- function(df, group_col="group"){
  tab <- table(df[[group_col]]); if (length(tab) < 2) return(df[0,,drop=FALSE])
  n_take <- min(as.integer(tab))
  idx <- unlist(tapply(seq_len(nrow(df)), df[[group_col]],
                       function(i) sample(i, n_take, replace = FALSE)))
  df[idx,,drop=FALSE]
}

get_props_by_group <- function(df_clusters, K){
  tab <- as.data.table(df_clusters)[, .N, by=.(group, cluster)]
  get_prop <- function(g){
    v <- numeric(K); tg <- tab[group==g]; if (nrow(tg)) v[as.integer(tg$cluster)] <- tg$N
    v / sum(v)
  }
  list(P = get_prop("Protected"), U = get_prop("Unprotected"))
```

```
}

richness_thresholded <- function(p, thr=0.01){ sum(p >= thr, na.rm=TRUE) }

fix_pc_sign_prcomp <- function(pc) {
  stopifnot(ncol(pc$rotation) >= 2, nrow(pc$rotation) > 0)
  if (sum(pc$rotation[, 1]) < 0) { pc$rotation[, 1] <- -pc$rotation[, 1]; pc$x[, 1] <- -
pc$x[, 1] }
  if (sum(pc$rotation[, 2]) < 0) { pc$rotation[, 2] <- -pc$rotation[, 2]; pc$x[, 2] <- -
pc$x[, 2] }
  pc
}

# ---------- Load data for the chosen replicate ----------
ds <- open_bank(BANK_DIR)
stopifnot(all(BANDS %in% colnames(ds)))
stopifnot(all(c("rep_id","group") %in% colnames(ds)))

cols_need <- c("group", BANDS)
df_all <- ds %>% filter(rep_id == REP_FOR_MAP) %>% select(all_of(cols_need)) %>% collect()
%>% as.data.frame()
if (SCALE_TO_REFLECTANCE) df_all[,BANDS] <- df_all[,BANDS,drop=FALSE] / SCALE_FACTOR
df_all <- df_all[stats::complete.cases(df_all[,BANDS,drop=FALSE]), , drop=FALSE]

# Train/eval + rarefy
take_g <- function(df, g, cap){
  i <- which(df$group==g)
  if (!length(i)) return(df[0,,drop=FALSE])
  if (length(i)>cap) i <- sample(i,cap)
  df[i,,drop=FALSE]
}
df_tr <- dplyr::bind_rows(
  take_g(df_all, "Protected",   TRAIN_PER_GROUP_MAX),
  take_g(df_all, "Unprotected", TRAIN_PER_GROUP_MAX)
)
df_ev <- dplyr::bind_rows(
  take_g(df_all, "Protected",   EVAL_PER_GROUP_MAX),
  take_g(df_all, "Unprotected", EVAL_PER_GROUP_MAX)
)
df_eq <- rarefy_equal(df_ev, "group")
stopifnot(nrow(df_eq) > 0)

# Fit/assign for K_TARGET
X_tr <- as.matrix(df_tr[,BANDS,drop=FALSE]); X_ev <- as.matrix(df_eq[,BANDS,drop=FALSE])
zs <- zscore_fit(X_tr); Z_tr <- zscore_apply(X_tr, zs); Z_ev <- zscore_apply(X_ev, zs)
set.seed(SEED)
km <- kmeans(Z_tr, centers = K_TARGET, iter.max = 100, nstart = 5)
df_eq$cluster <- predict_kmeans(Z_ev, km$centers)

# GLOBAL proportions (heatmap semantics)
props_g <- get_props_by_group(df_eq, K_TARGET)
pP_g <- props_g$P; pU_g <- props_g$U

# PCA on z-scored eval + fix axis signs
pc <- prcomp(Z_ev, center = FALSE, scale. = FALSE)
pc <- fix_pc_sign_prcomp(pc)

# Equal-width PC binning
NB <- NBINS
PC12 <- data.frame(PC1 = pc$x[,1], PC2 = pc$x[,2])
brk1 <- seq(min(PC12$PC1, na.rm=TRUE), max(PC12$PC1, na.rm=TRUE), length.out = NB + 1L)
brk2 <- seq(min(PC12$PC2, na.rm=TRUE), max(PC12$PC2, na.rm=TRUE), length.out = NB + 1L)
PC1_bin <- as.integer(cut(PC12$PC1, breaks = brk1, include.lowest = TRUE, labels = FALSE))
PC2_bin <- as.integer(cut(PC12$PC2, breaks = brk2, include.lowest = TRUE, labels = FALSE))

DT_eval <- data.frame(group = df_eq$group, cluster = df_eq$cluster,
```

```
                      PC1_bin = PC1_bin, PC2_bin = PC2_bin)
DT_eval$cell_id <- DT_eval$PC1_bin * 1000L + DT_eval$PC2_bin

# Cell × group × cluster counts
cell_tab <- as.data.table(DT_eval)[, .N, by = .(cell_id, PC1_bin, PC2_bin, group, cluster)]

# Present-sets for each τ (independent)
thr_vals  <- as.numeric(THR_LIST)
thr_names <- names(THR_LIST)
ord <- order(thr_vals)
thr_vals  <- thr_vals[ord]; thr_names <- thr_names[ord]

pP_sets <- lapply(thr_vals, function(t) which(pP_g >= t))
pU_sets <- lapply(thr_vals, function(t) which(pU_g >= t))
names(pP_sets) <- thr_names; names(pU_sets) <- thr_names

build_delta_from_sets <- function(thr_name){
  presentP <- pP_sets[[thr_name]]
  presentU <- pU_sets[[thr_name]]
  cell_tab[
    , .(
      S_P = sum(cluster %in% presentP & group=="Protected"),
      S_U = sum(cluster %in% presentU & group=="Unprotected")
    ),
    by = .(cell_id, PC1_bin, PC2_bin)
  ][, delta := S_U - S_P][]
}

MAP_0p5 <- build_delta_from_sets("0.5%")
MAP_2p0 <- build_delta_from_sets("2.0%")
MAP_5p0 <- build_delta_from_sets("5.0%")

# Shared legend / scale
common_scale <- scale_fill_stepsn(
  colors = c("#313695","#4575b4","#74add1","#abd9e9","#e0f3f8",
             "#f7f7f7", "#fee090","#fdae61","#f46d43","#d73027","#a50026"),
  breaks = seq(-4, 4, by = 1),
  limits = c(-4, 4),
  oob = scales::squish,
  show.limits = TRUE,
  name = "ΔS (Unprotected−Protected)"
)

plot_map <- function(MAP, ttl){
  ggplot(MAP, aes(PC1_bin, PC2_bin, fill = delta)) +
    geom_tile(width = 1, height = 1) +
    labs(title = ttl, x = "PC1 bin", y = "PC2 bin") +
    theme_minimal(base_size = 11) +
    theme(panel.grid = element_blank()) +
    common_scale
}

p05 <- plot_map(MAP_0p5, "τ = 0.5%")
p20 <- plot_map(MAP_2p0, "τ = 2.0%")
p50 <- plot_map(MAP_5p0, "τ = 5.0%")

MAP_DIR  <-  file.path(OUT_DIR,  sprintf("PCSPACE_GLOBALtau_RUN%02d_K%d",  REP_FOR_MAP,
K_TARGET))
dir_create(MAP_DIR, recurse = TRUE)

# Save singles (transparent)
ggsave(file.path(MAP_DIR, "deltaS_pcspace_tau0p5_GLOBAL.png"), p05, width = 7.5, height =
6.5, dpi = 300,
       device = ragg::agg_png, bg = "transparent")
ggsave(file.path(MAP_DIR, "deltaS_pcspace_tau2p0_GLOBAL.png"), p20, width = 7.5, height =
6.5, dpi = 300,
```

```
       device = ragg::agg_png, bg = "transparent")
ggsave(file.path(MAP_DIR, "deltaS_pcspace_tau5p0_GLOBAL.png"), p50, width = 7.5, height =
6.5, dpi = 300,
       device = ragg::agg_png, bg = "transparent")

# Triple panel (shared legend)
PANEL_TITLE    <- "Spatial Distribution of Spectral Richness Differences (ΔS) in Principal
Component Space"
PANEL_SUBTITLE <- sprintf("Global presence thresholds; K = %d, replicate %d", K_TARGET,
REP_FOR_MAP)

p_triple <- (p05 | p20 | p50) + plot_layout(guides = "collect")
p_triple <- p_triple & theme(
  plot.background       = element_rect(fill = NA, colour = NA),
  panel.background      = element_rect(fill = NA, colour = NA),
  legend.background     = element_rect(fill = NA, colour = NA),
  legend.box.background = element_rect(fill = NA, colour = NA),
  legend.key            = element_rect(fill = NA, colour = NA),
  legend.position       = "right"
)
p_triple <- p_triple + plot_annotation(
  title    = PANEL_TITLE,
  subtitle = PANEL_SUBTITLE,
  theme = theme(
    plot.background = element_rect(fill = NA, colour = NA),
    plot.title      = element_text(size = 16, face = "bold", hjust = 0.0, margin = margin(b
= 6)),
    plot.subtitle    = element_text(size = 12, colour = "grey20", hjust = 0.0, margin =
margin(b = 6))
  )
)

ragg::agg_png(
  filename   = file.path(MAP_DIR, "deltaS_pcspace_GLOBAL_triple.png"),
  width      = 16, height = 6, units = "in",
  res        = 300, background = "transparent"
)
print(p_triple)
dev.off()

message("Saved GLOBAL-τ PC-space maps to: ", MAP_DIR)
```

```
# ========================================================
# RICHNESS TABLE CALCULATION — Run 12 (0.5%, 2%, 5%)
# Compatible with cluster_composition.csv (any variant)
# ========================================================

suppressPackageStartupMessages({
  library(data.table)
  library(fs)
})

# ---- CONFIG ----
RUN_ID  <- 12L
BANK_BASE  <-  "C:/Users/david/OneDrive/Documents/Master  ESS/Masterarbeit/Data/Sentinel  2
Bulk/_SVA_BANK_REPS_20250818_213305"
IN_FILE  <-  file.path(BANK_BASE,  "PASS6_CLUSTER_10B_ALL",  sprintf("RUN_%02d",  RUN_ID),
"cluster_composition.csv")
OUT_DIR     <-     "C:/Users/david/OneDrive/Documents/Master     ESS/Masterarbeit/Data/SVA
National/Richness Tables"
dir_create(OUT_DIR, recurse = TRUE)

# ---- LOAD ----
dt <- fread(IN_FILE)
message("Loaded: ", IN_FILE)
message("Columns: ", paste(names(dt), collapse = ", "))
```

```
# ---- ENSURE PROPORTION COLUMNS EXIST ----
if (!all(c("prop_Protected", "prop_Unprotected") %in% names(dt))) {

  if (all(c("count_Protected", "count_Unprotected") %in% names(dt))) {
    message("→ Using count_* columns to derive proportions.")
    dt[, prop_Protected   := count_Protected   / sum(count_Protected,   na.rm = TRUE)]
    dt[, prop_Unprotected := count_Unprotected / sum(count_Unprotected, na.rm = TRUE)]

  } else if (all(c("observed_Protected", "observed_Unprotected") %in% names(dt))) {
    message("→ Using observed_* columns as proportions (binary flags).")
    dt[, prop_Protected   := observed_Protected]
    dt[, prop_Unprotected := observed_Unprotected]

  } else {
    stop("❌ Could not find any usable columns (prop_*, count_*, or observed_*). Check
your input CSV.")
  }
}

# ---- VERIFY ----
stopifnot(all(c("prop_Protected", "prop_Unprotected") %in% names(dt)))

# ---- THRESHOLDS ----
thr <- c(0.005, 0.02, 0.05)  # 0.5%, 2%, 5%

# ---- FUNCTION ----
# ---- FUNCTION ----
compute_richness <- function(dt, tau) {
  # make a local copy that keeps all columns
  out <- copy(dt)

  # calculate binary presence columns
  out[, observed_Protected   := as.integer(prop_Protected   >= tau)]
  out[, observed_Unprotected := as.integer(prop_Unprotected >= tau)]

  # add approximate counts (optional, for completeness)
  out[, count_Protected   := round(prop_Protected   * sum(prop_Protected   > 0))]
  out[, count_Unprotected := round(prop_Unprotected * sum(prop_Unprotected > 0))]

  # keep relevant columns only
  out <- out[, .(cluster, observed_Protected, observed_Unprotected,
                 count_Protected, count_Unprotected)]

  # write per-threshold table
  out_file  <-  file.path(OUT_DIR,  sprintf("RUN%02d_richness_table_tau%.1fp.csv",  RUN_ID,
tau * 100))
  fwrite(out, out_file)
  message("Saved: ", out_file)

  # summarize number of clusters exceeding threshold
  S_P <- sum(out$observed_Protected)
  S_U <- sum(out$observed_Unprotected)
  message(sprintf("τ = %.1f%% → Protected: %d | Unprotected: %d | Δ = %+d",
                  tau * 100, S_P, S_U, S_U - S_P))

  data.table(threshold = tau, S_P = S_P, S_U = S_U, delta = S_U - S_P)
}


# ---- LOOP ----
summary_list <- lapply(thr, function(t) compute_richness(dt, t))
summary_dt <- rbindlist(summary_list)
fwrite(summary_dt,     file.path(OUT_DIR,     sprintf("RUN%02d_richness_summary_all.csv",
RUN_ID)))
```

```
message("\n✅ All done! Summary written to:")
message(file.path(OUT_DIR, sprintf("RUN%02d_richness_summary_all.csv", RUN_ID)))
print(summary_dt)
```

```
# --- Packages ---
library(readr)
library(dplyr)
library(tidyr)
library(ggplot2)
library(scales)

# --- Paths ---
base_dir    <-    "C:/Users/david/OneDrive/Documents/Master    ESS/Masterarbeit/Data/SVA
National/Richness Tables"
files   <- c("0.5%" = "presence_th_0p5.csv",
             "2.0%" = "presence_th_2p0.csv",
             "5.0%" = "presence_th_5p0.csv")

# --- Hill helper (vector of masked counts -> S, D1, D2, E1, E2) ---
hill_from_counts <- function(x) {
  # x = numeric vector of counts (already masked by observed flags)
  x <- x[!is.na(x) & x > 0]
  S <- length(x)
  if (S == 0) return(tibble(S = 0, D1 = NA_real_, D2 = NA_real_, E1 = NA_real_, E2 =
NA_real_))
  p <- x / sum(x)
  H <- -sum(p * log(p))
  D1 <- exp(H)
  D2 <- 1 / sum(p^2)
  tibble(S = S, D1 = D1, D2 = D2, E1 = D1 / S, E2 = D2 / S)
}

# --- Compute per-threshold metrics WITHOUT altering the CSVs ---
summ_by_thr <- lapply(names(files), function(th_label) {
  path <- file.path(base_dir, files[[th_label]])
  df   <- read_csv(path, show_col_types = FALSE)

  # sanity check (no cleaning; just verify columns exist)
  need                                                                          <-
c("observed_Protected","observed_Unprotected","count_Protected","count_Unprotected")
  if (!all(need %in% names(df))) {
    stop(sprintf("File %s is missing one of required columns: %s",
                 basename(path), paste(need, collapse = ", ")))
  }

  # mask counts by observed flags (0/1); treat NA as 0 for this computation only
  cntP_mask <- ifelse(replace_na(df$observed_Protected,   0) > 0,
                      replace_na(df$count_Protected,      0), 0)
  cntU_mask <- ifelse(replace_na(df$observed_Unprotected, 0) > 0,
                      replace_na(df$count_Unprotected,    0), 0)

  prot <- hill_from_counts(cntP_mask) %>% rename_with(~paste0(.x, "_Protected"))
  unpr <- hill_from_counts(cntU_mask) %>% rename_with(~paste0(.x, "_Unprotected"))

  tibble(threshold = th_label) %>% bind_cols(prot, unpr)
}) %>% bind_rows() %>% mutate(threshold = factor(threshold, levels = names(files)))

# ==============================
# NATIONAL — PRINT & SAVE NUMERIC SUMMARY TABLES
# ==============================

# helper to round safely
safe_round <- function(x, k = 3) ifelse(is.finite(x), round(x, k), NA_real_)

# choose delta direction: "U_minus_P" (default) or "P_minus_U"
delta_mode <- "U_minus_P"
```

```
mk_summary_nat <- function(df, mode = c("U_minus_P","P_minus_U")) {
  mode <- match.arg(mode)
  if (mode == "U_minus_P") {
    dS   <- df$S_Unprotected  - df$S_Protected
    dD1  <- df$D1_Unprotected - df$D1_Protected
    dD2  <- df$D2_Unprotected - df$D2_Protected
    dE1  <- df$E1_Unprotected - df$E1_Protected
    dE2  <- df$E2_Unprotected - df$E2_Protected
    dS_pct  <- 100 * dS  / pmax(df$S_Protected, 1e-9)
    dD1_pct <- 100 * dD1 / pmax(df$D1_Protected, 1e-9)
    dD2_pct <- 100 * dD2 / pmax(df$D2_Protected, 1e-9)
    delta_label <- "Δ = Unprotected − Protected"
  } else {
    dS   <- df$S_Protected  - df$S_Unprotected
    dD1  <- df$D1_Protected - df$D1_Unprotected
    dD2  <- df$D2_Protected - df$D2_Unprotected
    dE1  <- df$E1_Protected - df$E1_Unprotected
    dE2  <- df$E2_Protected - df$E2_Unprotected
    dS_pct  <- 100 * dS  / pmax(df$S_Unprotected, 1e-9)
    dD1_pct <- 100 * dD1 / pmax(df$D1_Unprotected, 1e-9)
    dD2_pct <- 100 * dD2 / pmax(df$D2_Unprotected, 1e-9)
    delta_label <- "Δ = Protected − Unprotected"
  }

  out <- df %>%
    dplyr::transmute(
      threshold,
      S_Protected,   S_Unprotected,   dS  = dS,   dS_pct  = dS_pct,
      D1_Protected,  D1_Unprotected,  dD1 = dD1,  dD1_pct = dD1_pct,
      D2_Protected,  D2_Unprotected,  dD2 = dD2,  dD2_pct = dD2_pct,
      E1_Protected,  E1_Unprotected,  dE1 = dE1,
      E2_Protected,  E2_Unprotected,  dE2 = dE2,
      delta_def = delta_label
    ) %>%
    dplyr::mutate(dplyr::across(-c(threshold, delta_def), safe_round, k = 3))
  out
}

summary_wide_nat <- mk_summary_nat(summ_by_thr, mode = delta_mode)

# ---- Print to console (wide)
cat("\n================ NATIONAL RICHNESS / DIVERSITY / EVENNESS ====================\n")
print(summary_wide_nat, row.names = FALSE)
cat("==============================================================================\n")

# ---- Long tidy version (easy to paste or pivot in the thesis)
summary_long_nat <- summary_wide_nat |>
  tidyr::pivot_longer(
    -c(threshold, delta_def),
    names_to  = "metric",
    values_to = "value"
  )

# ---- Save both tables next to your other national outputs
readr::write_csv(summary_wide_nat,
                 file.path(base_dir,
"NATIONAL_summary_richness_diversity_evenness_wide.csv"))
readr::write_csv(summary_long_nat,
                 file.path(base_dir,
"NATIONAL_summary_richness_diversity_evenness_long.csv"))

# ---- Optional: one-line “headlines” per threshold
cat("\n-- Headlines by threshold (", unique(summary_wide_nat$delta_def)[1], ") --\n", sep
= "")
apply(summary_wide_nat, 1, function(r){
```

```
  cat(
    sprintf("τ=%s | ΔS=%s (%.1f%%), ΔD1=%s, ΔD2=%s, ΔE1=%s, ΔE2=%s\n",
            r[["threshold"]],
            r[["dS"]], as.numeric(r[["dS_pct"]]),
            r[["dD1"]], r[["dD2"]],
            r[["dE1"]], r[["dE2"]])
  )
})
cat("================================================================================\n")

# --- Add a synthetic t = 0 (use all counts, ignore observed flags) ---
# Uses the first file in `files` as the source for raw counts.
# Assumption: counts themselves do not depend on the threshold, only observed_* does.
# If you *do* have a dedicated 0% CSV, you can swap to that path instead.
first_path <- file.path(base_dir, files[[1]])
df0 <- read_csv(first_path, show_col_types = FALSE)

cntP_all <- replace_na(df0$count_Protected,   0)
cntU_all <- replace_na(df0$count_Unprotected, 0)

prot0 <- hill_from_counts(cntP_all) %>% rename_with(~paste0(.x, "_Protected"))
unpr0 <- hill_from_counts(cntU_all) %>% rename_with(~paste0(.x, "_Unprotected"))

t0_row <- tibble(threshold = "0%") %>% bind_cols(prot0, unpr0)

# --- Insert t=0 into the existing summary and fix factor order ---
summ_by_thr <- bind_rows(t0_row, summ_by_thr) %>%
  mutate(threshold = factor(threshold, levels = c("0%", names(files))))

# --- Richness (S) by threshold ---
long_S <- summ_by_thr %>%
  select(threshold, S_Protected, S_Unprotected) %>%
  tidyr::pivot_longer(-threshold, names_to = "metric_group", values_to = "value") %>%
  tidyr::separate(metric_group, into = c("metric","group"), sep = "_")

p_S <- ggplot(long_S, aes(x = threshold, y = value, fill = group)) +
  geom_col(position = position_dodge(width = 0.75), width = 0.7) +
  scale_fill_manual(values = c("Protected" = "#1b9e77", "Unprotected" = "#d95f02")) +
  labs(
    title = "Species Richness Across Presence Thresholds",
    subtitle = "Protected vs. Unprotected Miombo; S counts observed species",
    x = "Presence threshold", y = "Richness (S)", fill = ""
  ) +
  theme_minimal(base_size = 12)

ggsave(file.path(base_dir, "richness_S_by_threshold.png"),
       p_S, width = 10, height = 5.5, dpi = 320)


# Save summary table
out_csv <- file.path(base_dir, "hill_numbers_summary_by_threshold.csv")
write_csv(summ_by_thr, out_csv)

# --- Plot: Hill diversity (D1, D2) ---
# Exclude 5.0% from diversity plots
exclude_thr <- "5.0%"

long_div <- summ_by_thr %>%
  filter(threshold != exclude_thr) %>%
  select(threshold, D1_Protected, D1_Unprotected, D2_Protected, D2_Unprotected) %>%
  pivot_longer(-threshold, names_to = "metric_group", values_to = "value") %>%
  tidyr::separate(metric_group, into = c("metric","group"), sep = "_")
ymax_div <- max(long_div$value, na.rm = TRUE)
ymax_div <- ceiling(ymax_div * 1.05)  # small headroom

p_div <- ggplot(long_div, aes(x = threshold, y = value, fill = group)) +
```

```
  geom_col(position = position_dodge(width = 0.75), width = 0.7) +
  facet_wrap(~ metric, scales = "fixed") +      # was "free_y"
  coord_cartesian(ylim = c(0, ymax_div)) +      # unified limit for D1 & D2
  scale_fill_manual(values = c("Protected" = "#1b9e77", "Unprotected" = "#d95f02")) +
  labs(
    title = "Hill diversity Under Varying Presence Thresholds (0–2%)",
    subtitle = "Magnitude-weighted diversity (D1, D2) for Protected vs. Unprotected",
    x = "Presence threshold", y = "Diversity", fill = ""
  ) +
  theme_minimal(base_size = 12)
ggsave(file.path(base_dir, "hill_diversity_D1_D2_by_threshold.png"),
       p_div, width = 10, height = 6.2, dpi = 320)

# --- Plot: Evenness (E1, E2) ---
long_even <- summ_by_thr %>%
  select(threshold, E1_Protected, E1_Unprotected, E2_Protected, E2_Unprotected) %>%
  pivot_longer(-threshold, names_to = "metric_group", values_to = "value") %>%
  tidyr::separate(metric_group, into = c("metric","group"), sep = "_")

p_even <- ggplot(long_even, aes(x = threshold, y = value, fill = group)) +
  geom_col(position = position_dodge(width = 0.75), width = 0.7) +
  facet_wrap(~ metric, scales = "free_y") +
  scale_fill_manual(values = c("Protected" = "#1b9e77", "Unprotected" = "#d95f02")) +
  coord_cartesian(ylim = c(0, 1)) +
  labs(
    title = "Evenness Under Varying Presence Thresholds (0–2%)",
    subtitle = "Relative balance of abundances (E1, E2) in Protected vs. Unprotected",
    x = "Presence threshold", y = "Evenness (0–1)", fill = ""
  ) +
  theme_minimal(base_size = 12)
ggsave(file.path(base_dir, "evenness_E1_E2_by_threshold.png"),
       p_even, width = 10, height = 6.2, dpi = 320)

message("Saved:\n- ", out_csv,
        "\n- ", file.path(base_dir, "hill_diversity_D1_D2_by_threshold.png"),
        "\n- ", file.path(base_dir, "evenness_E1_E2_by_threshold.png"))


# --- Packages ---
library(readr)
library(dplyr)
library(tidyr)
library(ggplot2)

# --- Paths ---
csv_path    <-    "C:/Users/david/OneDrive/Documents/Master    ESS/Masterarbeit/Data/SVA
National/Richness Tables/presence_th_0p5.csv"
out_dir  <- dirname(csv_path)

# --- Load & clean names robustly ---
raw <- read_csv(csv_path, show_col_types = FALSE)
nm <- names(raw)
nm <- trimws(nm); nm <- gsub("\\s+", "_", nm); nm <- gsub("[^A-Za-z0-9_]", "", nm)
names(raw) <- nm

# --- Locate columns safely ---
col_cluster <- grep("^cluster$", names(raw), ignore.case = TRUE, value = TRUE)[1]
col_P <- grep("^count_.*protected$", names(raw), ignore.case = TRUE, value = TRUE)[1]
col_U <- grep("^count_.*unprotected$", names(raw), ignore.case = TRUE, value = TRUE)[1]
if (any(is.na(c(col_cluster, col_P, col_U)))) {
  stop(
    paste0(
      "Couldn't find required columns.\nFound names:\n- ",
      paste(names(raw), collapse = "\n- "),
      "\n\nNeed   one   matching   each   regex:\n     ^cluster$\n     ^count_.*protected$\n
^count_.*unprotected$\n"
```

```
    )
  )
}

# --- Build tidy df (wide) ---
df <- raw %>%
  transmute(
    cluster = .data[[col_cluster]],
    Protected   = as.numeric(.data[[col_P]]),
    Unprotected = as.numeric(.data[[col_U]])
  )

# --- Long format for violin/box plot ---
df_long <- df %>%
  pivot_longer(cols = c(Protected, Unprotected),
               names_to = "group", values_to = "richness")

# --- Violin + boxplot ---
p_box <- ggplot(df_long, aes(x = group, y = richness, fill = group)) +
  geom_violin(trim = FALSE, alpha = 0.2, linewidth = 0.3) +
  geom_boxplot(width = 0.25, outlier.alpha = 0.3) +
  scale_fill_manual(values = c("Protected" = "#1b9e77", "Unprotected" = "#d95f02")) +
  labs(
    title = "Richness distribution across protected and unprotected clusters",
    subtitle = "Comparison of mean and spread in species richness",
    x = NULL, y = "Richness per cluster", fill = ""
  ) +
  theme_minimal(base_size = 12) +
  theme(legend.position = "none")

ggsave(file.path(out_dir, "richness_distribution_box_violin.png"),
       p_box, width = 7, height = 5.2, dpi = 320)

library(patchwork)

# ---------- Build separate panels for D1, D2, E1, E2 (for 2x3) ----------

# Ensure 5% is excluded from both diversity and evenness panels
exclude_thr <- "5.0%"

# (Re)build long_div (already done above; just reuse)
# Compute shared y-limit for D1 & D2 to make them comparable
ymax_div <- max(long_div$value, na.rm = TRUE)
ymax_div <- ceiling(ymax_div * 1.05)

# Separate D1 / D2 panels
p_D1 <- ggplot(dplyr::filter(long_div, metric == "D1"),
               aes(x = threshold, y = value, fill = group)) +
  geom_col(position = position_dodge(width = 0.75), width = 0.7) +
  coord_cartesian(ylim = c(0, ymax_div)) +
  scale_fill_manual(values = c("Protected" = "#1b9e77", "Unprotected" = "#d95f02")) +
  labs(title = "Diversity D1 = exp(Shannon)", x = "Presence threshold", y = "Diversity",
fill = "") +
  theme_minimal(base_size = 12) +
  theme(legend.position = "bottom")

p_D2 <- ggplot(dplyr::filter(long_div, metric == "D2"),
               aes(x = threshold, y = value, fill = group)) +
  geom_col(position = position_dodge(width = 0.75), width = 0.7) +
  coord_cartesian(ylim = c(0, ymax_div)) +
  scale_fill_manual(values = c("Protected" = "#1b9e77", "Unprotected" = "#d95f02")) +
  labs(title = "Diversity D2 = 1/∑p²", x = "Presence threshold", y = "Diversity", fill =
"") +
  theme_minimal(base_size = 12) +
  theme(legend.position = "bottom")
```

```
# Evenness: exclude 5% just for plotting
long_even_plot <- summ_by_thr %>%
  dplyr::filter(threshold != exclude_thr) %>%
  dplyr::select(threshold, E1_Protected, E1_Unprotected, E2_Protected, E2_Unprotected) %>%
  tidyr::pivot_longer(-threshold, names_to = "metric_group", values_to = "value") %>%
  tidyr::separate(metric_group, into = c("metric","group"), sep = "_")

p_E1 <- ggplot(dplyr::filter(long_even_plot, metric == "E1"),
               aes(x = threshold, y = value, fill = group)) +
  geom_col(position = position_dodge(width = 0.75), width = 0.7) +
  coord_cartesian(ylim = c(0, 1)) +
  scale_fill_manual(values = c("Protected" = "#1b9e77", "Unprotected" = "#d95f02")) +
  labs(title = "Evenness E1 = D1/S", x = "Presence threshold", y = "Evenness (0–1)", fill
= "") +
  theme_minimal(base_size = 12) +
  theme(legend.position = "bottom")

p_E2 <- ggplot(dplyr::filter(long_even_plot, metric == "E2"),
               aes(x = threshold, y = value, fill = group)) +
  geom_col(position = position_dodge(width = 0.75), width = 0.7) +
  coord_cartesian(ylim = c(0, 1)) +
  scale_fill_manual(values = c("Protected" = "#1b9e77", "Unprotected" = "#d95f02")) +
  labs(title = "Evenness E2 = D2/S", x = "Presence threshold", y = "Evenness (0–1)", fill
= "") +
  theme_minimal(base_size = 12) +
  theme(legend.position = "bottom")

# ---------- Assemble 2×3 panel ----------
panel_2x3 <- (p_box | p_S) /
  (p_D1  | p_D2) /
  (p_E1  | p_E2) +
  plot_layout(guides = "collect") &
  theme(legend.position = "bottom")

# Save the combined panel
ggsave(file.path(base_dir, "panel_2x3_richness_diversity_evenness.png"),
       panel_2x3, width = 14, height = 14, dpi = 320)
```

Local SVA Code

```
# ================================================
# PASS L0 — AOI prep (local), minimal & close to your code
# ================================================

suppressPackageStartupMessages({
  library(sf)
  library(dplyr)
  library(ggplot2)
  library(fs)
})

# --- Paths (yours) ---
miombo_path    <- "C:/Users/david/OneDrive/Documents/Master ESS/Masterarbeit/Data/Shapefiles miombo
extend in protected areas/miombo_classified_protection.shp"
villages_path <- "C:/Users/david/Desktop/David/StudyVillages/StudyVillages_Big.shp"
output_folder <- "C:/Users/david/OneDrive/Documents/Master ESS/Masterarbeit/Data/Village Boundaries
and Miombo extent"
dir_create(output_folder, recurse = TRUE)

# --- Read ---
miombo   <- st_read(miombo_path, quiet = TRUE)
villages <- st_read(villages_path, quiet = TRUE)

# --- CRS + validity ---
miombo   <- st_make_valid(miombo)
villages <- st_make_valid(villages)
if (st_crs(villages) != st_crs(miombo)) {
  villages <- st_transform(villages, st_crs(miombo))
}

# --- VillageName (minimal guard, no fuzzy stuff) ---
# Your original used: ifelse(is.na(Name), Village_na, Name)
# We mirror that, but only if columns exist; else stop with a clear message.
if ("Name" %in% names(villages) && "Village_na" %in% names(villages)) {
  villages <- villages %>%
    mutate(VillageName    =    ifelse(is.na(Name)    |    Name    ==    "",    as.character(Village_na),
as.character(Name)))
} else if ("Name" %in% names(villages)) {
  villages <- villages %>% mutate(VillageName = as.character(Name))
} else if ("Village_na" %in% names(villages)) {
  villages <- villages %>% mutate(VillageName = as.character(Village_na))
} else {
  stop("Neither 'Name' nor 'Village_na' exists in villages. Available fields are:\n",
       paste(names(villages), collapse = ", "))
}

# --- Governance mapping (exact strings only; same as yours) ---
OA_NAMES <- c("Kidete", "Msowero")
CB_NAMES <- c("Kigunga", "Ulaya Mbuyuni", "Ulaya Kibaoni")

villages <- villages %>%
  mutate(Governance = case_when(
    VillageName %in% OA_NAMES ~ "OpenAccess",
    VillageName %in% CB_NAMES ~ "CBNRM",
    TRUE ~ NA_character_
  ))

# --- Diagnostics to see why OA might be missing ---
cat("\n[CHECK] Unique VillageName values in the shapefile:\n")
print(sort(unique(villages$VillageName)))

cat("\n[CHECK] Which of your OA names are present?\n")
print(intersect(OA_NAMES, unique(villages$VillageName)))
cat("Missing OA names (not found exactly in VillageName): ",
    paste(setdiff(OA_NAMES, unique(villages$VillageName)), collapse = ", "),
    "\n", sep = "")

cat("\n[CHECK] Which of your CBNRM names are present?\n")
print(intersect(CB_NAMES, unique(villages$VillageName)))
cat("Missing CBNRM names (not found exactly in VillageName): ",
```

```
    paste(setdiff(CB_NAMES, unique(villages$VillageName)), collapse = ", "),
    "\n", sep = "")

cat("\n[CHECK] Governance counts in villages:\n")
print(table(villages$Governance, useNA = "ifany"))

# --- Intersect Miombo with villages (keep only tagged Governance) ---
miombo_in_villages <- suppressWarnings(
  st_intersection(miombo, villages %>% select(VillageName, Governance))
) %>% filter(!is.na(Governance))

# --- Split AOIs (just like you did) ---
miombo_cb <- miombo_in_villages %>% filter(Governance == "CBNRM")
miombo_oa <- miombo_in_villages %>% filter(Governance == "OpenAccess")

cat("\nNumber of CB polygons:", nrow(miombo_cb), "\n")
cat("Number of OA polygons:", nrow(miombo_oa), "\n")

# --- Quick plot (same style you had) ---
ggplot() +
  geom_sf(data = miombo_in_villages, fill = "gray40", color = NA, alpha = 0.6) +
  geom_sf(data = villages, aes(fill = Governance), color = "black", alpha = 0.5) +
  geom_sf_text(data = villages, aes(label = VillageName), size = 4, na.rm = TRUE) +
  scale_fill_manual(values = c("CBNRM" = "forestgreen", "OpenAccess" = "tomato"), na.value = "grey85")
+
  ggtitle("Miombo Extent within Study Villages by Governance") +
  theme_minimal()

# --- Save SHP files (your original target filenames) ---
st_write(villages %>% select(VillageName, Governance),
         dsn = file.path(output_folder, "study_villages_with_governance.shp"),
         delete_layer = TRUE, quiet = TRUE)

st_write(miombo_in_villages,
         dsn = file.path(output_folder, "miombo_in_villages.shp"),
         delete_layer = TRUE, quiet = TRUE)

st_write(miombo_cb,
         dsn = file.path(output_folder, "miombo_cb.shp"),
         delete_layer = TRUE, quiet = TRUE)

st_write(miombo_oa,
         dsn = file.path(output_folder, "miombo_oa.shp"),
         delete_layer = TRUE, quiet = TRUE)

cat("\nWrote shapefiles to:\n", output_folder, "\n")
```

```
# ============================
# PASS L1 — Local per-pixel bank (CBNRM vs OpenAccess)
# + NDVI density with 0.5 threshold & %≥0.5 labels
# ============================
suppressPackageStartupMessages({
  library(sf)
  library(terra)
  library(data.table)
  library(fs)
  library(arrow)      # Parquet
  library(ggplot2)
})

# ---------- USER PATHS ----------
# .SAFE tile
tile_path      <-      "C:/Users/david/OneDrive/Documents/Master      ESS/Masterarbeit/Data/Sentinel
2/S2B_MSIL2A_20240616T073619_N0510_R092_T37MBN_20240616T100238.SAFE"

# AOIs written in L0 (your shapefiles)
aoi_cb_path <- "C:/Users/david/OneDrive/Documents/Master ESS/Masterarbeit/Data/Village Boundaries and
Miombo extent/miombo_cb.shp"
aoi_oa_path <- "C:/Users/david/OneDrive/Documents/Master ESS/Masterarbeit/Data/Village Boundaries and
Miombo extent/miombo_oa.shp"

# ---------- OUTPUT ROOT (auto from SAFE name) ----------
safe_name <- basename(tile_path)
tile_code <- sub(".*_(T[0-9A-Z]{5})_.*", "\\1", safe_name)
```

```
acq_date  <- sub(".*_(\\d{8})T\\d{6}.*$", "\\1", safe_name)
if (identical(tile_code, safe_name)) tile_code <- "TXXXX"
if (identical(acq_date,  safe_name)) acq_date  <- "YYYYMMDD"

LOCAL_BASE <- file.path("C:/Users/david/OneDrive/Documents/Master ESS/Masterarbeit/Data",
                        sprintf("Local_SVA_%s_%s", tile_code, acq_date))
L1_DIR     <- file.path(LOCAL_BASE, "L1_BANK")
dir_create(L1_DIR, recurse = TRUE)

BANK_PARQUET <- file.path(L1_DIR, "bank_local.parquet")
SUMMARY_TXT  <- file.path(L1_DIR, "bank_local_summary.txt")
PLOT_NDVI    <- file.path(L1_DIR, "ndvi_density_local.png")
PCT_TXT      <- file.path(L1_DIR, "ndvi_threshold_summary.txt")

# ---------- CONFIG ----------
BAND_IDS <- c("B01","B02","B03","B04","B05","B06","B07","B8A","B11","B12")
SCL_BAD  <- c(0,1,3,8,9,10,11)      # mask-out classes
NDVI_THRESH <- 0.5

# ---------- HELPERS ----------
find_tile_files <- function(tile_dir, band_ids=BAND_IDS) {
  all_20m <- list.files(tile_dir, pattern = "_20m\\.jp2$", recursive = TRUE, full.names = TRUE)
  if (!length(all_20m)) return(NULL)
  pick_one <- function(id) {
    m <- grep(paste0("_", id, "_20m\\.jp2$"), all_20m, value = TRUE)
    if (length(m)) m[1] else NA_character_
  }
  bands <- setNames(vapply(band_ids, pick_one, character(1)), band_ids)
  scl   <- pick_one("SCL")
  if (is.na(scl) || anyNA(bands)) return(NULL)
  list(bands = bands, scl = scl)
}
read_aoi <- function(path) {
  stopifnot(file_exists(path))
  st_make_valid(st_read(path, quiet = TRUE))
}

# ---------- 1) READ RASTERS ----------
fls <- find_tile_files(tile_path)
stopifnot(!is.null(fls))

r_bands <- terra::rast(unname(fls$bands))
names(r_bands) <- names(fls$bands)  # B01..B12

r_scl <- terra::rast(fls$scl)

# ---------- 2) READ & PROJECT AOIs ----------
aoi_cb <- read_aoi(aoi_cb_path)
aoi_oa <- read_aoi(aoi_oa_path)
aoi_cb <- st_transform(aoi_cb, crs(terra::crs(r_bands, proj = TRUE)))
aoi_oa <- st_transform(aoi_oa, crs(terra::crs(r_bands, proj = TRUE)))
v_cb <- terra::vect(aoi_cb)
v_oa <- terra::vect(aoi_oa)

# ---------- 3) SCL VALID MASK ----------
rcl <- cbind(SCL_BAD, NA)
r_valid <- terra::classify(r_scl, rcl = rcl, others = 1)  # 1 = good, NA = bad

# ---------- 4) EXTRACT ALL VALID PIXELS PER GROUP ----------
extract_group <- function(vmask, group_label) {
  rb <- terra::crop(r_bands, vmask)
  rb <- terra::mask(rb, vmask)
  rv <- terra::crop(r_valid, rb)
  rb <- terra::mask(rb, rv)        # keep only SCL-valid

  df <- as.data.frame(rb, xy = TRUE, na.rm = TRUE)
  if (!nrow(df)) return(NULL)
  setDT(df)
  setnames(df, old = c("x","y"), new = c("x","y"))
  df[, group := group_label]
  # NDVI: (B8A - B04) / (B8A + B04); scale cancels if bands are in 0..10000
  df[, ndvi := (B8A - B04) / pmax(B8A + B04, .Machine$double.eps)]
  df[]
}
```

```
dt_cb <- extract_group(v_cb, "CBNRM")
dt_oa <- extract_group(v_oa, "OpenAccess")
if (is.null(dt_cb) && is.null(dt_oa)) stop("No valid pixels found inside either AOI after SCL
masking.")

BANK <- rbindlist(list(dt_cb, dt_oa), use.names = TRUE, fill = TRUE)

# ---------- 5) WRITE BANK ----------
keep_cols <- c("x","y","group", BAND_IDS, "ndvi")
keep_cols <- keep_cols[keep_cols %in% names(BANK)]
BANK <- BANK[, ..keep_cols]
arrow::write_parquet(BANK, BANK_PARQUET)

# ---------- 6) SUMMARY ----------
n_cb <- if (is.null(dt_cb)) 0L else nrow(dt_cb)
n_oa <- if (is.null(dt_oa)) 0L else nrow(dt_oa)
summ <- c(
  sprintf("Local SVA bank — %s %s", tile_code, acq_date),
  sprintf("Pixels kept (SCL-valid only): CBNRM = %s | OpenAccess = %s",
          format(n_cb, big.mark=","), format(n_oa, big.mark=",")),
  sprintf("Total rows written: %s", format(nrow(BANK), big.mark=",")),
  sprintf("Parquet: %s", BANK_PARQUET)
)
writeLines(summ, SUMMARY_TXT)
message(paste(summ, collapse = "\n"))

# ---------- 7) NDVI DENSITY with 0.5 LINE + %≥0.5 ----------
stopifnot(all(c("group","ndvi") %in% names(BANK)))
PCT <- BANK[, .(
  n = .N,
  n_ge   = sum(ndvi >= NDVI_THRESH, na.rm = TRUE),
  pct_ge = 100 * mean(ndvi >= NDVI_THRESH, na.rm = TRUE),
  mean_ndvi   = mean(ndvi, na.rm = TRUE),
  median_ndvi = median(ndvi, na.rm = TRUE)
), by = group][order(group)]

# also write a small text summary
writeLines(c(
  sprintf("NDVI threshold = %.2f", NDVI_THRESH),
  paste(sprintf("%s: %.1f%% of pixels ≥ %.2f (n=%s)",
                PCT$group, PCT$pct_ge, NDVI_THRESH, format(PCT$n, big.mark=",")),
        collapse = "\n")
), PCT_TXT)

# build plot
g <- ggplot(BANK, aes(x = ndvi, fill = group, color = group)) +
  geom_density(alpha = 0.30, linewidth = 0.6) +
  geom_vline(xintercept = NDVI_THRESH, linetype = 2, color = "grey30") +
  annotate("text",
           x = NDVI_THRESH + 0.02, y = Inf, vjust = 1.6, hjust = 0,
           label = sprintf("Threshold = %.2f", NDVI_THRESH), size = 3.6) +
  annotate("text",
           x = NDVI_THRESH + 0.02, y = Inf, vjust = 3.0, hjust = 0,
           label = paste(sprintf("%s: %.1f%% ≥ %.2f", PCT$group, PCT$pct_ge, NDVI_THRESH),
                         collapse = "\n"),
           size = 3.6) +
  labs(title = sprintf("NDVI distribution — %s %s", tile_code, acq_date),
       subtitle = "Dashed line = 0.5; labels show share of pixels ≥ 0.5",
       x = "NDVI", y = "Density") +
  theme_minimal()

# save + show in RStudio
ggsave(PLOT_NDVI, g, width = 9, height = 5.2, dpi = 300)
print(g)

message("\nPASS L1 complete. Outputs in: ", L1_DIR)
```

```
# ============================
# PASS L2 — Local NDVI map + per-group %≥0.5 + violin+box plot
# ============================
suppressPackageStartupMessages({
  library(sf)
  library(terra)
```

```
  library(data.table)
  library(ggplot2)
  library(scales)
  library(fs)
})

# ---------- USER PATHS ----------
tile_path           <-    "C:/Users/david/OneDrive/Documents/Master    ESS/Masterarbeit/Data/Sentinel
2/S2B_MSIL2A_20240616T073619_N0510_R092_T37MBN_20240616T100238.SAFE"
aoi_cb_path <- "C:/Users/david/OneDrive/Documents/Master ESS/Masterarbeit/Data/Village Boundaries and
Miombo extent/miombo_cb.shp"
aoi_oa_path <- "C:/Users/david/OneDrive/Documents/Master ESS/Masterarbeit/Data/Village Boundaries and
Miombo extent/miombo_oa.shp"

# ---------- OUTPUT ROOT (auto from SAFE name) ----------
safe_name <- basename(tile_path)
tile_code <- sub(".*_(T[0-9A-Z]{5})_.*", "\\1", safe_name)
acq_date  <- sub(".*_(\\d{8})T\\d{6}.*$", "\\1", safe_name)
if (identical(tile_code, safe_name)) tile_code <- "TXXXX"
if (identical(acq_date,  safe_name)) acq_date  <- "YYYYMMDD"

LOCAL_BASE <- file.path("C:/Users/david/OneDrive/Documents/Master ESS/Masterarbeit/Data",
                        sprintf("Local_SVA_%s_%s", tile_code, acq_date))
L2_DIR   <- file.path(LOCAL_BASE, "L2_NDVI")
dir_create(L2_DIR, recurse = TRUE)

NDVI_TIF    <- file.path(L2_DIR, "ndvi_aoi.tif")
NDVI_PNG    <- file.path(L2_DIR, "ndvi_map.png")
VIOLIN_PNG  <- file.path(L2_DIR, "ndvi_violin_box.png")
STATS_CSV   <- file.path(L2_DIR, "ndvi_stats_by_group.csv")
THRESH_TXT  <- file.path(L2_DIR, "ndvi_threshold_summary.txt")

# ---------- CONFIG ----------
NDVI_THRESH <- 0.5
SCL_BAD     <- c(0,1,3,8,9,10,11)
BANDS_NEED  <- c("B04","B8A")

# ---------- HELPERS ----------
find_tile_files <- function(tile_dir, band_ids=BANDS_NEED) {
  all_20m <- list.files(tile_dir, pattern = "_20m\\.jp2$", recursive = TRUE, full.names = TRUE)
  if (!length(all_20m)) return(NULL)
  pick_one <- function(id) {
    m <- grep(paste0("_", id, "_20m\\.jp2$"), all_20m, value = TRUE)
    if (length(m)) m[1] else NA_character_
  }
  bands <- setNames(vapply(band_ids, pick_one, character(1)), band_ids)
  scl   <- pick_one("SCL")
  if (is.na(scl) || anyNA(bands)) return(NULL)
  list(bands = bands, scl = scl)
}
read_aoi <- function(path) { stopifnot(file_exists(path)); st_make_valid(st_read(path, quiet = TRUE))
}

# ---------- 1) READ RASTERS + AOIs ----------
fls <- find_tile_files(tile_path); stopifnot(!is.null(fls))
r_b04 <- terra::rast(fls$bands["B04"])
r_b8a <- terra::rast(fls$bands["B8A"])
r_scl <- terra::rast(fls$scl)

aoi_cb <- read_aoi(aoi_cb_path)
aoi_oa <- read_aoi(aoi_oa_path)

# align CRS
tcrs   <- terra::crs(r_b04, proj = TRUE)
aoi_cb <- st_transform(aoi_cb, tcrs)
aoi_oa <- st_transform(aoi_oa, tcrs)

v_cb  <- terra::vect(aoi_cb)
v_oa  <- terra::vect(aoi_oa)
v_all <- terra::vect(st_union(aoi_cb, aoi_oa))

# ---------- 2) VALIDITY MASK (SCL) ----------
rcl <- cbind(SCL_BAD, NA)
r_valid <- terra::classify(r_scl, rcl = rcl, others = 1)  # 1=good, NA=bad
```

```
# ---------- 3) NDVI RASTER ----------
# Note: scale (0..10000) cancels out in ratio
ndvi <- (r_b8a - r_b04) / (r_b8a + r_b04)
# mask to valid SCL and AOI union
ndvi <- terra::mask(terra::crop(ndvi, v_all), v_all)
ndvi <- terra::mask(ndvi, terra::crop(r_valid, ndvi))  # drop invalid SCL

# Write GeoTIFF (for GIS/QGIS)
terra::writeRaster(ndvi, NDVI_TIF, overwrite = TRUE)

# ---------- 4) PER-GROUP STATS ----------
stats_one <- function(v_aoi, label) {
  r <- terra::mask(terra::crop(ndvi, v_aoi), v_aoi)
  if (is.null(r) || all(is.na(values(r)))) {
    return(data.table(group = label, n = 0, mean = NA_real_, median = NA_real_, sd = NA_real_,
                      pct_ge = NA_real_))
  }
  v <- values(r, mat = FALSE, na.rm = TRUE)
  if (!length(v)) return(data.table(group = label, n = 0, mean = NA_real_, median = NA_real_, sd =
NA_real_, pct_ge = NA_real_))
  data.table(
    group   = label,
    n       = length(v),
    mean    = mean(v, na.rm = TRUE),
    median  = median(v, na.rm = TRUE),
    sd      = sd(v, na.rm = TRUE),
    pct_ge  = 100 * mean(v >= NDVI_THRESH, na.rm = TRUE)
  )
}
ST_CB <- stats_one(v_cb, "CBNRM")
ST_OA <- stats_one(v_oa, "OpenAccess")
STATS <- rbindlist(list(ST_CB, ST_OA), use.names = TRUE, fill = TRUE)
fwrite(STATS, STATS_CSV)

writeLines(c(
  sprintf("NDVI ≥ %.2f summary — %s %s", NDVI_THRESH, tile_code, acq_date),
  sprintf("CBNRM:      %.1f%% (n=%s)", STATS[group=="CBNRM"]$pct_ge, format(STATS[group=="CBNRM"]$n,
big.mark=",")),
  sprintf("OpenAccess:          %.1f%%          (n=%s)",          STATS[group=="OpenAccess"]$pct_ge,
format(STATS[group=="OpenAccess"]$n, big.mark=","))
), THRESH_TXT)

# ---------- 5) MAP (geom_tile with native cell size) ----------
df <- as.data.frame(ndvi, xy = TRUE, na.rm = TRUE)
names(df)[3] <- "ndvi"

# optional thin if > 1.5M cells
if (nrow(df) > 1.5e6) {
  set.seed(42)
  df <- df[sample.int(nrow(df), 1.5e6), , drop = FALSE]
}

rres  <- terra::res(ndvi)  # c(xres, yres)
res_x <- rres[1]; res_y <- rres[2]

aoi_cb$gov <- "CBNRM"; aoi_oa$gov <- "OpenAccess"
aoi_out <- rbind(aoi_cb[, "gov"], aoi_oa[, "gov"])

g_map <- ggplot() +
  geom_tile(data = df, aes(x = x, y = y, fill = ndvi),
            width = res_x, height = res_y) +
  scale_fill_viridis_c(option = "C", limits = c(0, 1), oob = scales::squish, name = "NDVI") +
  geom_sf(data = aoi_out, aes(color = gov), fill = NA, linewidth = 0.3, inherit.aes = FALSE) +
  scale_color_manual(values = c(CBNRM = "forestgreen", OpenAccess = "tomato"), name = "AOI") +
  coord_sf(crs = sf::st_crs(aoi_out), expand = FALSE) +
  labs(title = sprintf("NDVI map — %s %s", tile_code, acq_date),
       subtitle = "Color = continuous NDVI (0–1)", x = NULL, y = NULL) +
  theme_minimal() +
  theme(legend.position = "right", panel.grid = element_blank())

ggsave(NDVI_PNG, g_map, width = 9.5, height = 6.5, dpi = 300)
print(g_map)
```

```
# ---------- 6) VIOLIN + BOXPLOT (styled like your example) ----------
extract_ndvi_vals <- function(v_aoi, label, max_n = 5e5) {
  r <- terra::mask(terra::crop(ndvi, v_aoi), v_aoi)
  vals <- values(r, mat = FALSE, na.rm = TRUE)
  if (!length(vals)) return(data.table(group = label, ndvi = numeric(0)))
  if (length(vals) > max_n) { set.seed(42); vals <- sample(vals, max_n) }
  data.table(group = label, ndvi = vals)
}

DT_CB  <- extract_ndvi_vals(v_cb, "CBNRM")
DT_OA  <- extract_ndvi_vals(v_oa, "OpenAccess")
DT_ALL <- rbindlist(list(DT_CB, DT_OA), use.names = TRUE, fill = TRUE)

if (nrow(DT_ALL)) {
  g_violin <- ggplot(DT_ALL, aes(x = group, y = ndvi, fill = group)) +
    geom_violin(trim = FALSE, alpha = 0.2, linewidth = 0.3) +
    geom_boxplot(width = 0.25, outlier.alpha = 0.1) +
    scale_fill_manual(values = c("CBNRM" = "#e2776b", "OpenAccess" = "#23a9b7")) +
    scale_y_continuous(limits = c(0, 1), labels = number_format(accuracy = 0.01)) +
    labs(
      title = sprintf("NDVI distribution by governance — %s %s", tile_code, acq_date),
      subtitle = "Violin = distribution; box = median & IQR",
      x = NULL, y = "NDVI", fill = ""
    ) +
    theme_minimal(base_size = 12) +
    theme(legend.position = "none")

  ggsave(VIOLIN_PNG, g_violin, width = 7, height = 5.2, dpi = 320)
  print(g_violin)
} else {
  message("No NDVI values found in either AOI — violin plot skipped.")
}

message("PASS L2 complete. Outputs in: ", L2_DIR,
        "\n- ", NDVI_TIF,
        "\n- ", STATS_CSV,
        "\n- ", THRESH_TXT,
        "\n- ", NDVI_PNG,
        "\n- ", VIOLIN_PNG)
```

```
# ============================
# L3 — Local PCA (CBNRM vs OpenAccess), NATIONAL-STYLE (mirrors global script)
# ============================
suppressPackageStartupMessages({
  library(arrow); library(dplyr); library(data.table)
  library(ggplot2); library(scales); library(fs)
  library(cowplot); library(ggrepel) # (labels for future compass if needed)
})

# ---------- INPUTS ----------
# 1) Where your local bank lives (produced earlier by your L1 step)
tile_path      <-      "C:/Users/david/OneDrive/Documents/Master      ESS/Masterarbeit/Data/Sentinel
2/S2B_MSIL2A_20240616T073619_N0510_R092_T37MBN_20240616T100238.SAFE"
safe_name <- basename(tile_path)
tile_code  <-  sub(".*_(T[0-9A-Z]{5})_.*",  "\\1",  safe_name);  if  (identical(tile_code,  safe_name))
tile_code <- "TXXXX"
acq_date   <- sub(".*_(\\d{8})T\\d{6}.*$",  "\\1",  safe_name);  if  (identical(acq_date,   safe_name))
acq_date  <- "YYYYMMDD"

LOCAL_BASE   <- file.path("C:/Users/david/OneDrive/Documents/Master ESS/Masterarbeit/Data",
                          sprintf("Local_SVA_%s_%s", tile_code, acq_date))
BANK_PARQUET <- file.path(LOCAL_BASE, "L1_BANK", "bank_local.parquet")
stopifnot(file.exists(BANK_PARQUET))

# 2) Where to save all outputs for the local run
OUT_ROOT <- "C:/Users/david/OneDrive/Documents/Master ESS/Masterarbeit/Data/SVA local"
BAND_MODE <- "10B"    # set to "5B" to use the 5-band variant
BANDS_10B <- c("B01","B02","B03","B04","B05","B06","B07","B8A","B11","B12")
BANDS_5B  <- c("B05","B06","B07","B8A","B04")
BAND_IDS  <- if (BAND_MODE == "10B") BANDS_10B else BANDS_5B

OUT_DIR <- file.path(OUT_ROOT, sprintf("LOCAL_%s_%s_%s", tile_code, acq_date, BAND_MODE))
dir_create(OUT_DIR, recurse = TRUE)
```

```
# Optional: cache PC1–PC2 scores (for instant plotting later)
CACHE_SCORES   <- TRUE
SCORES_PARQUET <- file.path(OUT_DIR, sprintf("scores_PC12_%s_%s_%s.parquet", tile_code, acq_date,
BAND_MODE))

# Colors — match national palette mapping (OA=orange, CBNRM=teal)
GROUP_COLS <- c("OpenAccess" = "#d95f02", "CBNRM" = "#1b9e77")

# Plot caps (plotting only)
SCATTER_CAP <- 300000L
DENSITY_CAP <- 500000L

# ---------- HELPERS ----------
`%||%` <- function(x, y) if (!is.null(x)) x else y

# Match the national script: lock PCA orientation so signs are consistent
fix_pc_sign <- function(model, scores_df) {
  rot <- model$rotation
  flip1 <- sum(rot[, 1]) < 0
  flip2 <- sum(rot[, 2]) < 0
  if (flip1) { model$rotation[, 1] <- -model$rotation[, 1]; scores_df$PC1 <- -scores_df$PC1 }
  if (flip2) { model$rotation[, 2] <- -model$rotation[, 2]; scores_df$PC2 <- -scores_df$PC2 }
  list(model = model, scores = scores_df, flips = c(PC1 = ifelse(flip1, -1, 1), PC2 = ifelse(flip2, -
1, 1)))
}

# Shared theme
theme_pca <- function() {
  theme_minimal(base_size = 12) +
    theme(panel.grid.minor = element_blank(), panel.grid.major = element_line(size = 0.3))
}

# Loading bars builder (identical style)
make_loading_bar <- function(df, pc = "PC1", title_txt = NULL) {
  df$val <- df[[pc]]
  df <- df[order(abs(df$val), decreasing = FALSE), ]
  df$band <- factor(df$band, levels = df$band)
  ggplot(df, aes(x = band, y = val, fill = val >= 0)) +
    geom_col(width = 0.7) +
    geom_hline(yintercept = 0, linewidth = 0.3) +
    coord_flip() +
    scale_fill_manual(values = c("TRUE" = "#737373", "FALSE" = "#bdbdbd"), guide = "none") +
    labs(title = title_txt %||% paste(pc, "loadings"), x = NULL, y = "Loading") +
    theme_minimal(base_size = 10) +
    theme(plot.title = element_text(size = 10, face = "bold", hjust = 0),
          axis.text.y = element_text(size = 9), panel.grid.minor = element_blank())
}

# ---------- LOAD ----------
ds <- open_dataset(BANK_PARQUET, format = "parquet")
df <- ds %>% collect() %>% as.data.frame()
keep_cols <- intersect(c("group", BAND_IDS), names(df))
df <- df[, keep_cols, drop = FALSE]
stopifnot(all(df$group %in% c("CBNRM","OpenAccess")))

# Optional reflectance scaling to match national (auto-detect DN vs reflectance)
# If values look like DN (max > 1.5), divide by 10000.
max_any <- max(df[, BAND_IDS], na.rm = TRUE)
if (is.finite(max_any) && max_any > 1.5) {
  df[, BAND_IDS] <- df[, BAND_IDS] / 10000
}

# Keep complete rows
df <- df[stats::complete.cases(df[, BAND_IDS, drop = FALSE]), , drop = FALSE]

# ---------- PCA (ALL PIXELS) ----------
X <- as.matrix(df[, BAND_IDS, drop = FALSE])
pca_model <- prcomp(X, center = TRUE, scale. = TRUE)

# Save PCA model RDS (so other local passes can reuse it)
saveRDS(pca_model,   file.path(OUT_DIR,   sprintf("pca_model_%s_%s_%s.rds",   tile_code,   acq_date,
BAND_MODE)))

# Scores for plotting + metrics
```

```
PCs <- predict(pca_model, X)
scores <- data.frame(PC1 = PCs[, 1], PC2 = PCs[, 2], group = as.character(df$group), stringsAsFactors
= FALSE)

# Lock orientation to match national aesthetics
fixed <- fix_pc_sign(pca_model, scores)
pca_model <- fixed$model
scores    <- fixed$scores
pc_flips  <- fixed$flips
saveRDS(pca_model, file.path(OUT_DIR, sprintf("pca_model_oriented_%s_%s_%s.rds", tile_code, acq_date,
BAND_MODE)))

# Variance explained labels
ve      <- (pca_model$sdev^2) / sum(pca_model$sdev^2)
vx1     <- round(100 * ve[1], 1)
vx2     <- round(100 * ve[2], 1)
vx12    <- round(100 * (ve[1] + ve[2]), 1)

# Optional cache of scores
if (isTRUE(CACHE_SCORES)) {
  arrow::write_parquet(arrow::as_arrow_table(scores), SCORES_PARQUET)
}

# ---------- METRICS (matching national) ----------
centroids <- scores %>% group_by(group) %>% summarise(PC1 = mean(PC1), PC2 = mean(PC2), .groups =
"drop")

n_tot <- nrow(scores)
n_by  <- table(scores$group)
n_cb  <- ifelse("CBNRM"     %in% names(n_by), as.integer(n_by[["CBNRM"]]), 0L)
n_oa  <- ifelse("OpenAccess" %in% names(n_by), as.integer(n_by[["OpenAccess"]]), 0L)

dist_PC12 <- {
  cCB <- centroids[centroids$group=="CBNRM", c("PC1","PC2")]
  cOA <- centroids[centroids$group=="OpenAccess", c("PC1","PC2")]
  if (nrow(cCB) == 1 && nrow(cOA) == 1) sqrt((cCB$PC1 - cOA$PC1)^2 + (cCB$PC2 - cOA$PC2)^2) else
NA_real_
}

# Separability (PC1–PC2)
sep_metrics <- (function(df){
  ridge <- 1e-8
  A <- subset(df, group=="CBNRM",     select=c(PC1,PC2))
  B <- subset(df, group=="OpenAccess", select=c(PC1,PC2))
  if (nrow(A) < 2 || nrow(B) < 2) return(list(DM=NA_real_, DB=NA_real_, JM=NA_real_))
  muA <- colMeans(A); muB <- colMeans(B); dmu <- as.numeric(muA - muB)
  S1 <- stats::cov(A); S2 <- stats::cov(B)
  n1 <- nrow(A); n2 <- nrow(B)
  Spooled <- ((n1 - 1) * S1 + (n2 - 1) * S2) / max(1, (n1 + n2 - 2))
  inv <- function(M) solve(M + diag(ridge, 2))
  logdet <- function(M) as.numeric(determinant(M + diag(ridge,2), log = TRUE)$modulus)
  DM <- sqrt(t(dmu) %*% inv(Spooled) %*% dmu)
  Sbar  <- (S1 + S2) / 2
  DB <- 0.125 * as.numeric(t(dmu) %*% inv(Sbar) %*% dmu) + 0.5 * (logdet(Sbar) - 0.5 * (logdet(S1) +
logdet(S2)))
  JM <- 2 * (1 - exp(-DB))
  list(DM = as.numeric(DM), DB = as.numeric(DB), JM = as.numeric(JM))
})(scores)


# ---------- SAVE METRICS CSV (local) ----------
metrics_df <- data.frame(
  tile = tile_code, date = acq_date, band_mode = BAND_MODE,
  n_total = n_tot, n_CBNRM = n_cb, n_OpenAccess = n_oa,
  centroid_dist_PC12 = dist_PC12,
  mahalanobis_PC12   = sep_metrics$DM,
  bhattacharyya_PC12 = sep_metrics$DB,
  jm_PC12            = sep_metrics$JM,
  PC1_var = vx1/100, PC2_var = vx2/100, PC12_var = vx12/100,
  stringsAsFactors = FALSE
)
```

```
data.table::fwrite(metrics_df, file.path(OUT_DIR, "metrics_local.csv"))

# ---------- PCA STATS EXPORT (variance + loadings, PCs 1–10) ----------
# Ensure bands are named
if (is.null(rownames(pca_model$rotation))) {
  rownames(pca_model$rotation) <- BAND_IDS
}

ve_all <- (pca_model$sdev^2) / sum(pca_model$sdev^2)
k <- min(10L, length(ve_all))
VE_TBL <- data.frame(
  pc                    = paste0("PC", seq_len(k)),
  variance_proportion   = as.numeric(ve_all[1:k]),
  variance_percent      = round(100 * ve_all[1:k], 3),
  cumulative_proportion = as.numeric(cumsum(ve_all)[1:k]),
  cumulative_percent    = round(100 * cumsum(ve_all)[1:k], 3),
  stringsAsFactors = FALSE
)

LD <- as.data.frame(pca_model$rotation[, 1:k, drop = FALSE])
LD$band <- rownames(LD)
band_nm <- c(B01=443, B02=490, B03=560, B04=665, B05=705, B06=740, B07=783, B8A=865, B11=1610,
B12=2190)
LD$band_label <- ifelse(LD$band %in% names(band_nm), paste0(LD$band, " (", band_nm[LD$band], " nm)"),
LD$band)
LD_out <- LD[, c("band", "band_label", paste0("PC", seq_len(k)))]

xlsx_path <- file.path(OUT_DIR, sprintf("pca_stats_local_%s_%s_%s.xlsx", tile_code, acq_date,
BAND_MODE))
if (requireNamespace("writexl", quietly = TRUE)) {
  writexl::write_xlsx(list(variance = VE_TBL, loadings = LD_out), path = xlsx_path)
  message("📘 PCA stats workbook: ", xlsx_path)
} else {
  data.table::fwrite(VE_TBL, file.path(OUT_DIR, sprintf("pca_variance_local_%s_%s_%s.csv", tile_code,
acq_date, BAND_MODE)))
  data.table::fwrite(LD_out, file.path(OUT_DIR, sprintf("pca_loadings_local_%s_%s_%s.csv", tile_code,
acq_date, BAND_MODE)))
  message("📄 Wrote CSVs (install {writexl} for a single .xlsx)")
}

# ---------- PLOTS ----------
# A) Full scatter (all points; cap for speed) + ellipses + centroids
SAMP <- scores
if (nrow(SAMP) > SCATTER_CAP) { set.seed(42); SAMP <- SAMP[sample.int(nrow(SAMP), SCATTER_CAP), ] }

p_scatter <- ggplot(SAMP, aes(PC1, PC2, color = group)) +
  geom_point(size = 0.18, alpha = 0.20) +
  stat_ellipse(aes(fill = group), type = "norm", level = 0.95, geom = "polygon", alpha = 0.12, color
= NA) +
  stat_ellipse(type = "norm", level = 0.95, linewidth = 0.6) +
  geom_point(data = centroids, shape = 4, stroke = 1.2, size = 4, color = "black") +
  coord_equal() +
  scale_color_manual(values = GROUP_COLS, name = NULL) +
  scale_fill_manual(values = GROUP_COLS, guide = "none") +
  labs(
    title = sprintf("Local PCA —Tile ID %s (10 Bands) ", tile_code),
    subtitle = sprintf("PC1 %.1f%% • PC2 %.1f%% • total %.1f%%   |   N = %s  (CBNRM = %s, OpenAccess
= %s)",
                       vx1, vx2, vx12, comma(n_tot), comma(n_cb), comma(n_oa)),
    x = sprintf("PC1 (%.1f%%)", vx1), y = sprintf("PC2 (%.1f%%)", vx2)
  ) + theme_pca()

ggsave(file.path(OUT_DIR, "plotA_pca_scatter_allpoints.png"), p_scatter, width = 9.5, height = 7.5,
dpi = 300)

# B) Left: per-group density (no ellipses)  |  Right: bar loadings (PC1 & PC2) + separability box
# (cap density points if desired — stat_density_2d works on full scores, but we keep option for parity)
SCO <- scores
if (nrow(SCO) > DENSITY_CAP) { set.seed(42); SCO <- SCO[sample.int(nrow(SCO), DENSITY_CAP), ] }

p_density <- ggplot() +
  stat_density_2d(
    data = subset(SCO, group == "OpenAccess"),
    aes(x = PC1, y = PC2, fill = "OpenAccess", alpha = after_stat(level)),
```

```
    geom = "polygon", bins = 12, color = NA
  ) +
  stat_density_2d(
    data = subset(SCO, group == "CBNRM"),
    aes(x = PC1, y = PC2, fill = "CBNRM", alpha = after_stat(level)),
    geom = "polygon", bins = 12, color = NA
  ) +
  geom_point(data = centroids, aes(PC1, PC2, color = group), shape = 4, stroke = 1.25, size = 4) +
  scale_alpha(range = c(0.03, 0.60), guide = "none") +
  scale_fill_manual(values = GROUP_COLS, breaks = c("OpenAccess", "CBNRM"), name = "Density by group")
+
  scale_color_manual(values = GROUP_COLS, guide = "none") +
  coord_equal() +
  labs(title = sprintf("Local per-group density 10 Bands"), x = sprintf("PC1 (%.1f%%)", vx1), y =
sprintf("PC2 (%.1f%%)", vx2)) +
  theme_pca() + theme(legend.position = "bottom")

LD2 <- as.data.frame(pca_model$rotation[, 1:2, drop = FALSE]); LD2$band <- rownames(LD2)
p_bar1 <- make_loading_bar(LD2, "PC1", "PC1 loadings")
p_bar2 <- make_loading_bar(LD2, "PC2", "PC2 loadings")
bars_right <- cowplot::plot_grid(p_bar1, p_bar2, ncol = 1, rel_heights = c(0.55, 0.45))

# Compose right-bottom label like national: separability metrics
sep_text <- sprintf(
  paste(
    "Separability in PC1–PC2",
    "Centroid dist: %s",
    "Mahalanobis: %s",
    "Bhattacharyya: %s",
    "Jeffries–Matusita: %s",
    sep = "\n"
  ),
  ifelse(is.na(dist_PC12),      "NA", sprintf("%.3f", dist_PC12)),
  ifelse(is.na(sep_metrics$DM), "NA", sprintf("%.3f", sep_metrics$DM)),
  ifelse(is.na(sep_metrics$DB), "NA", sprintf("%.3f", sep_metrics$DB)),
  ifelse(is.na(sep_metrics$JM), "NA", sprintf("%.3f", sep_metrics$JM))
)

label_plot <- ggplot() +
  annotate("label", x = 1, y = 0, label = sep_text, hjust = 1, vjust = 0, size = 3.2,
           label.size = 0.25, label.padding = grid::unit(c(0.25,0.25,0.25,0.25), "lines"),
           colour = "black", fill = "white") +
  coord_cartesian(xlim = c(0,1), ylim = c(0,1), expand = FALSE) +
  theme_void()

base <- cowplot::plot_grid(p_density, bars_right, ncol = 2, rel_widths = c(0.74, 0.26), align = "h")
p_composite <- cowplot::ggdraw(base) +
  cowplot::draw_plot(label_plot, x = 0.62, y = 0.02, width = 0.36, height = 0.22)

ggsave(file.path(OUT_DIR, "plotB_pca_density_bars.png"), p_composite, width = 12, height = 8, dpi =
320)

message("✅ Local analysis complete. Outputs in: ", OUT_DIR)
```

```
# ============================
# PASS L5 — Local k-means clustering (CBNRM vs OpenAccess)
# ============================
suppressPackageStartupMessages({
  library(arrow)
  library(data.table)
  library(dplyr)
  library(ggplot2)
  library(scales)
  library(fs)
})

# ---------- USER: same SAFE as L0–L4 so folders line up ----------
tile_path      <-      "C:/Users/david/OneDrive/Documents/Master      ESS/Masterarbeit/Data/Sentinel
2/S2B_MSIL2A_20240616T073619_N0510_R092_T37MBN_20240616T100238.SAFE"

# ---------- OUTPUT ROOT ----------
safe_name <- basename(tile_path)
tile_code <- sub(".*_(T[0-9A-Z]{5})_.*", "\\1", safe_name)
acq_date  <- sub(".*_(\\d{8})T\\d{6}.*$", "\\1", safe_name)
```

```
if (identical(tile_code, safe_name)) tile_code <- "TXXXX"
if (identical(acq_date,  safe_name)) acq_date  <- "YYYYMMDD"

LOCAL_BASE <- file.path("C:/Users/david/OneDrive/Documents/Master ESS/Masterarbeit/Data",
                        sprintf("Local_SVA_%s_%s", tile_code, acq_date))
L1_DIR <- file.path(LOCAL_BASE, "L1_BANK")
L5_DIR <- file.path(LOCAL_BASE, "L5_CLUSTER")
dir_create(L5_DIR, recurse = TRUE)

BANK_PARQUET <- file.path(L1_DIR, "bank_local.parquet")
stopifnot(file.exists(BANK_PARQUET))

# ---------- BANDS / MODE ----------
BANDS_10B <- c("B01","B02","B03","B04","B05","B06","B07","B8A","B11","B12")
BANDS_5B  <- c("B05","B06","B07","B8A","B04")

# pick 10B if available, else 5B
cols_in_file <- colnames(read_parquet(BANK_PARQUET, as_data_frame = TRUE))
if (all(BANDS_10B %in% cols_in_file)) {
  BAND_MODE <- "10B"; BAND_IDS <- BANDS_10B
} else if (all(BANDS_5B %in% cols_in_file)) {
  BAND_MODE <- "5B";  BAND_IDS <- BANDS_5B
} else {
  stop("Neither 10B nor 5B band sets fully present in bank_local.parquet.")
}
message(sprintf("[L5] Using %s bands: %s", BAND_MODE, paste(BAND_IDS, collapse=", ")))

# ---------- K & sampling (training only) ----------
K_CLUSTERS          <- 100L
TRAIN_PER_GROUP_MAX <- 30000L   # training cap *per group* to balance centroids
SEED                <- 42L
set.seed(SEED)

# ---------- helpers ----------
zscore_fit <- function(X){
  mu <- colMeans(X, na.rm = TRUE)
  sdv <- apply(X, 2, sd)
  sdv[sdv == 0 | !is.finite(sdv)] <- 1
  list(mu = mu, sd = sdv)
}
zscore_apply <- function(X, pars){
  sweep(sweep(X, 2, pars$mu, "-"), 2, pars$sd, "/")
}
predict_kmeans <- function(X, centers){
  xsq <- rowSums(X^2); csq <- rowSums(centers^2); G <- X %*% t(centers)
  D2  <- matrix(xsq, nrow=nrow(G), ncol=ncol(G)) +
    matrix(csq, nrow=nrow(G), ncol=ncol(G), byrow=TRUE) - 2*G
  max.col(-D2, ties.method = "first")
}
# composition metrics on cluster-proportion vectors
hill_numbers <- function(p){
  p <- p[p>0]
  if (!length(p)) return(list(S=0, H=NA_real_, D1=NA_real_, D2=NA_real_, J=NA_real_))
  S <- length(p); H <- -sum(p * log(p)); D1 <- exp(H); D2 <- 1/sum(p^2); J <- if (S>1) H/log(S) else
NA_real_
  list(S=S,H=H,D1=D1,D2=D2,J=J)
}
bc_distance <- function(p,q) 0.5*sum(abs(p-q))
js_divergence <- function(p,q,eps=1e-12){
  p<-p+eps; q<-q+eps; p<-p/sum(p); q<-q/sum(q); m<-0.5*(p+q)
  0.5*(sum(p*log(p/m))+sum(q*log(q/m)))
}
hellinger_distance <- function(p,q){ v<-sqrt(p)-sqrt(q); (1/sqrt(2))*sqrt(sum(v*v)) }

# ---------- LOAD BANK (ALL PIXELS) ----------
DF <- as.data.frame(read_parquet(BANK_PARQUET))
stopifnot(all(c("group", BAND_IDS) %in% names(DF)))
stopifnot(all(DF$group %in% c("CBNRM","OpenAccess")))

# build matrix, drop NA rows
X <- as.matrix(DF[, BAND_IDS, drop = FALSE])
keep <- stats::complete.cases(X) & !is.na(DF$group)
DF   <- DF[keep, , drop = FALSE]
X    <- X[keep, , drop = FALSE]
```

```
# z-score
zs_pars <- zscore_fit(X)
Z       <- zscore_apply(X, zs_pars)

# ---------- TRAIN KMEANS on balanced subset ----------
idxC <- which(DF$group == "CBNRM")
idxO <- which(DF$group == "OpenAccess")
nC   <- length(idxC); nO <- length(idxO)
n_take <- min(TRAIN_PER_GROUP_MAX, nC, nO)
tr_idx <- c(sample(idxC, n_take), sample(idxO, n_take))
Ztr    <- Z[tr_idx, , drop = FALSE]

message(sprintf("[L5] Training k-means: K=%d on %d points (%d per group).", K_CLUSTERS, nrow(Ztr),
n_take))
km <- kmeans(Ztr, centers = K_CLUSTERS, iter.max = 100, nstart = 5)

# ---------- ASSIGN EVERY PIXEL ----------
cl <- predict_kmeans(Z, km$centers)              # 1..K
DF$cluster <- as.integer(cl)

# ---------- PER-GROUP COMPOSITIONS (ALL PIXELS) ----------
tab <- as.data.table(DF)[, .N, by = .(group, cluster)]
allK <- sort(unique(tab$cluster))
get_prop <- function(g) {
  v <- numeric(K_CLUSTERS); names(v) <- as.character(seq_len(K_CLUSTERS))
  tg <- tab[group == g]
  if (nrow(tg)) v[as.character(tg$cluster)] <- tg$N
  v / sum(v)
}
pC <- get_prop("CBNRM")
pO <- get_prop("OpenAccess")

# ---------- METRICS ----------
metC <- hill_numbers(pC)
metO <- hill_numbers(pO)
metrics_group <- data.table(
  metric = c("S","H","D1","D2","J"),
  CBNRM      = c(metC$S, metC$H, metC$D1, metC$D2, metC$J),
  OpenAccess = c(metO$S, metO$H, metO$D1, metO$D2, metO$J)
)
metrics_between <- data.table(
  bray_curtis    = bc_distance(pC, pO),
  jensen_shannon = js_divergence(pC, pO),
  hellinger      = hellinger_distance(pC, pO)
)

# ---------- ENRICH CLUSTER CENTERS (NDVI & red-edge diffs) ----------
# back-transform centers to reflectance scale
centers_z <- km$centers
colnames(centers_z) <- BAND_IDS
centers <- sweep(centers_z, 2, zs_pars$sd, "*")
centers <- sweep(centers, 2, zs_pars$mu, "+")

centers_dt <- as.data.table(centers)
centers_dt[, cluster := .I]
# put in reflectance 0–1 if your bank is scaled by 10000
to01 <- function(v) v / 10000
centers_dt[, (BAND_IDS) := lapply(.SD, to01), .SDcols = BAND_IDS]

if (all(c("B8A","B04") %in% BAND_IDS)) {
  centers_dt[, ndvi := (B8A - B04) / pmax(B8A + B04, .Machine$double.eps)]
} else {
  centers_dt[, ndvi := NA_real_]
}
# simple RE slope features if available
if (all(c("B05","B06","B07","B8A") %in% BAND_IDS)) {
  centers_dt[, re_06_05 := B06 - B05]
  centers_dt[, re_07_06 := B07 - B06]
  centers_dt[, re_8A_07 := B8A - B07]
} else {
  centers_dt[, `:=`(re_06_05 = NA_real_, re_07_06 = NA_real_, re_8A_07 = NA_real_)]
}
```

```
# proportions & Δ by group
cl_prop <- data.table(
  cluster = seq_len(K_CLUSTERS),
  prop_CBNRM      = as.numeric(pC),
  prop_OpenAccess = as.numeric(pO)
)
cl_prop[, delta := prop_OpenAccess - prop_CBNRM]
# Bray–Curtis share per cluster (how much each cluster contributes to BC)
bc_total <- 0.5 * sum(abs(cl_prop$delta))
cl_prop[, bc_share := if (bc_total > 0) 100 * (0.5 * abs(delta)) / bc_total else 0]

TOPN <- 20L
top <- merge(cl_prop, centers_dt, by = "cluster")[order(-abs(delta))][1:TOPN]
top[, cluster := factor(cluster, levels = rev(cluster))]

# ---------- SAVE TABLES ----------
fwrite(metrics_group,   file.path(L5_DIR, "metrics_group.csv"))
fwrite(metrics_between, file.path(L5_DIR, "metrics_between.csv"))
fwrite(cl_prop,         file.path(L5_DIR, "cluster_proportions.csv"))
fwrite(centers_dt,      file.path(L5_DIR, "cluster_centers_reflectance.csv"))
fwrite(top,             file.path(L5_DIR, sprintf("top%d_clusters_by_abs_delta.csv", TOPN)))

saveRDS(list(kmeans_centers = km$centers,
             zs_mu = zs_pars$mu, zs_sd = zs_pars$sd,
             bands = BAND_IDS, K = K_CLUSTERS,
             mode = BAND_MODE),
        file = file.path(L5_DIR, "kmeans_model_zscore.rds"))

# ---------- PLOTS ----------
# 1) Per-group metrics (dumbbell)
label_map <- c(
  S  = "Richness (types)",
  H  = "Shannon entropy (nats)",
  D1 = "Diversity (Hill q=1)",
  D2 = "Dominance-weighted diversity (Hill q=2)",
  J  = "Evenness (Pielou’s J)"
)
mg <- data.table::copy(metrics_group)
mg[, metric_lab := factor(label_map[metric], levels = label_map[c("S","H","D1","D2","J")])]
db_long <- mg |>
  dplyr::select(metric_lab, CBNRM, OpenAccess) |>
  tidyr::pivot_longer(c(CBNRM, OpenAccess), names_to = "group", values_to = "value")

p_db <- ggplot() +
  geom_segment(data = mg,
               aes(y = metric_lab, yend = metric_lab,
                   x = CBNRM, xend = OpenAccess),
               linewidth = 0.6, color = "grey65") +
  geom_point(data = subset(db_long, group == "CBNRM"),
             aes(x = value, y = metric_lab), shape = 21, fill = "white", size = 2.2) +
  geom_point(data = subset(db_long, group == "OpenAccess"),
             aes(x = value, y = metric_lab), shape = 16, size = 2.2) +
  labs(
    title = sprintf("Per-group composition metrics — %s %s   (K=%d, %s)", tile_code, acq_date,
K_CLUSTERS, BAND_MODE),
    subtitle = "Open circle = CBNRM; filled circle = OpenAccess",
    x = NULL, y = NULL
  ) + theme_minimal(base_size = 12) + theme(panel.grid.minor = element_blank())
ggsave(file.path(L5_DIR, "plot_group_metrics_dumbbell.png"), p_db, width = 9, height = 5, dpi = 300)
print(p_db)

# 2) Top-Δ clusters (OA–CB), colored by NDVI with RE annotations at right
library(gridExtra)
range_x <- max(abs(top$delta), na.rm = TRUE) * 1.05
p_main <- ggplot(top, aes(x = cluster, y = delta)) +
  geom_hline(yintercept = 0, linetype = 2, color = "grey70") +
  geom_segment(aes(xend = cluster, y = 0, yend = delta), linewidth = 0.6, color = "grey70") +
  geom_point(aes(color = ndvi), size = 2) +
  coord_flip() +
  scale_y_continuous(labels = percent_format(accuracy = 0.1),
                     limits = c(-range_x, range_x),
                     expand = expansion(mult = c(0.05, 0.05))) +
  scale_color_gradient(low = "#9d6b53", high = "#1a7f37", name = "NDVI", guide = "none") +
  labs(title = sprintf("Top %d cluster proportion differences — OA – CB", TOPN),
```

```
       x = "Cluster ID", y = "Δ proportion") +
  theme_minimal(base_size = 12) + theme(panel.grid.minor = element_blank(),
                                        plot.title = element_text(hjust = 0.5))
lab_df <- data.frame(
  cluster   = top$cluster,
  ndvi      = top$ndvi,
  label_all = sprintf("NDVI %.2f | RE Δ06–05=%.02f, Δ07–06=%.02f, Δ8A–07=%.02f | %.1f%% BC",
                      top$ndvi, top$re_06_05, top$re_07_06, top$re_8A_07, top$bc_share)
)
p_side <- ggplot(lab_df, aes(y = cluster)) +
  geom_point(aes(x = 0.03, color = ndvi), size = 2.4) +
  geom_text(aes(x = 0.07, label = label_all), hjust = 0, size = 3.1, color = "grey20") +
  scale_x_continuous(limits = c(0, 1), expand = c(0, 0)) +
  scale_color_gradient(low = "#9d6b53", high = "#1a7f37", guide = "none") +
  theme_void() + theme(plot.margin = margin(t = 28, r = 12, b = 26, l = 0))
g_combined <- gridExtra::arrangeGrob(p_main, p_side, ncol = 2, widths = c(0.60, 0.40))
grid::grid.newpage(); grid::grid.draw(g_combined)
ggsave(file.path(L5_DIR, "plot_cluster_top_differences_wide.png"), g_combined, width = 12, height =
7, dpi = 300)

# 3) (Optional) quick XY cluster map if bank has x,y columns
if (all(c("x","y") %in% names(DF))) {
  # thin for plotting speed if gigantic
  PLOT_CAP <- 1e6
  DFM <- if (nrow(DF) > PLOT_CAP) DF[sample.int(nrow(DF), PLOT_CAP), c("x","y","cluster","group")]
else DF[, c("x","y","cluster","group")]
  g_map <- ggplot(DFM, aes(x, y, fill = factor(cluster))) +
    geom_raster() +
    coord_equal() + scale_fill_viridis_d(option = "C", guide = "none") +
    labs(title = sprintf("Local k-means clusters (K=%d, %s)", K_CLUSTERS, BAND_MODE),
         subtitle = "Sampled for display if very large",
         x = NULL, y = NULL) + theme_minimal() +
    theme(panel.grid.minor = element_blank(), panel.grid.major = element_blank())
  ggsave(file.path(L5_DIR, "cluster_map_xy.png"), g_map, width = 9, height = 7, dpi = 300)
  print(g_map)
}

# ---------- CONSOLE SUMMARY ----------
cat("\n=== PASS L5 — Local k-means (", BAND_MODE, ", K=", K_CLUSTERS, ") ===\n", sep = "")
cat("Per-group metrics (S, H, D1, D2, J):\n"); print(metrics_group)
cat("\nBetween-group (BC, JS, HL):\n"); print(metrics_between)
message("\nOutputs in: ", L5_DIR)
```

```
# ============================================================
# LOCAL — Δ proportion bar (|Δ| order) + NDVI/CCI/NDMI heatmap
# Inputs (from L5):
#   - cluster_proportions.csv        (prop_CBNRM, prop_OpenAccess)
#   - kmeans_model_zscore.rds        (kmeans_centers, zs_mu, zs_sd, bands)
# Output:
#   - L6_VIZ/local_indices_by_cluster_sorted_absDelta.png
# ============================================================

suppressPackageStartupMessages({
  library(data.table); library(ggplot2); library(scales)
  library(patchwork);  library(fs)
})

# ---- SAFE / paths ----
tile_path      <-      "C:/Users/david/OneDrive/Documents/Master      ESS/Masterarbeit/Data/Sentinel
2/S2B_MSIL2A_20240616T073619_N0510_R092_T37MBN_20240616T100238.SAFE"
safe_name <- basename(tile_path)
tile_code  <-  sub(".*_(T[0-9A-Z]{5})_.*",  "\\1",  safe_name);  if  (identical(tile_code,  safe_name))
tile_code <- "TXXXX"
acq_date   <-  sub(".*_(\\d{8})T\\d{6}.*$",  "\\1",  safe_name);  if  (identical(acq_date,   safe_name))
acq_date  <- "YYYYMMDD"

LOCAL_BASE <- file.path("C:/Users/david/OneDrive/Documents/Master ESS/Masterarbeit/Data",
                        sprintf("Local_SVA_%s_%s", tile_code, acq_date))
L5_DIR  <- file.path(LOCAL_BASE, "L5_CLUSTER")
OUT_DIR <- file.path(LOCAL_BASE, "L6_VIZ")
dir_create(OUT_DIR, recurse = TRUE)

IN_COMP  <- file.path(L5_DIR, "cluster_proportions.csv")
IN_MODEL <- file.path(L5_DIR, "kmeans_model_zscore.rds")
```

```
OUT_PNG  <- file.path(OUT_DIR, "local_indices_by_cluster_sorted_absDelta.png")

stopifnot(file.exists(IN_COMP), file.exists(IN_MODEL))

# ---- LOAD ----
clu <- fread(IN_COMP)        # expects: cluster, prop_CBNRM, prop_OpenAccess
m   <- readRDS(IN_MODEL)     # list: kmeans_centers, zs_mu, zs_sd, bands, K, mode

# ---- Ensure proportions exist (fallback if only counts present) ----
if (!all(c("prop_CBNRM", "prop_OpenAccess") %in% names(clu))) {
  if (all(c("count_CBNRM","count_OpenAccess") %in% names(clu))) {
    clu[, prop_CBNRM      := count_CBNRM      / sum(count_CBNRM,      na.rm = TRUE)]
    clu[, prop_OpenAccess := count_OpenAccess / sum(count_OpenAccess, na.rm = TRUE)]
  } else stop("cluster_proportions.csv must have prop_* or count_* columns.")
}

# ---- Δ per cluster; sort by |Δ| (OA − CB) ----
cld <- clu[, .(cluster, delta = prop_OpenAccess - prop_CBNRM)]
cl_order <- cld[order(-abs(delta))]$cluster
cld[, cluster_f := factor(cluster, levels = cl_order)]

# ---- Back-transform centers to original space from z-score model ----
centers_z <- m$kmeans_centers
if (!is.null(colnames(centers_z))) {
  # align to model band order (usually already correct)
  centers_z <- centers_z[, m$bands, drop = FALSE]
} else {
  colnames(centers_z) <- m$bands
}
centers <- sweep(centers_z, 2, m$zs_sd, "*")
centers <- sweep(centers,  2, m$zs_mu, "+")
centers_dt <- as.data.table(centers)
centers_dt[, cluster := .I]

# ---- Indices from centers ----
# (ratios; absolute scale cancels; eps avoids 0/0)
eps <- .Machine$double.eps
calc_ndvi <- function(b8a, b04)           (b8a - b04) / pmax(b8a + b04, eps)
calc_cci  <- function(b8a, b05, b06, b07) {
  cci5 <- (b8a / pmax(b05, eps)) - 1
  cci6 <- (b8a / pmax(b06, eps)) - 1
  cci7 <- (b8a / pmax(b07, eps)) - 1
  (cci5 + cci6 + cci7) / 3
}
calc_ndmi <- function(b8a, b11)           (b8a - b11) / pmax(b8a + b11, eps)

need <- c("B04","B05","B06","B07","B8A","B11")
stopifnot(all(need %in% names(centers_dt)))

centers_dt[, `:=`(
  NDVI = calc_ndvi(B8A, B04),
  CCI  = calc_cci (B8A, B05, B06, B07),
  NDMI = calc_ndmi(B8A, B11)
)]

# ---- Merge indices with Δ and reshape ----
M <- merge(centers_dt[, .(cluster, NDVI, CCI, NDMI)], cld, by = "cluster")
M[, cluster_f := factor(cluster, levels = cl_order)]

M_long <- melt(M, id.vars = c("cluster","cluster_f","delta"),
               variable.name = "metric", value.name = "value")

# z-score per metric so heat colors are comparable across rows
M_long[, value_z := (value - mean(value, na.rm = TRUE)) / sd(value, na.rm = TRUE),
       by = metric]

# ---- Plots ----
# Top: Δ bar
p_delta <- ggplot(cld, aes(x = cluster_f, y = delta)) +
  geom_col(width = 0.85, fill = "grey55") +
  geom_hline(yintercept = 0, linetype = 2, color = "grey60") +
  scale_y_continuous(labels = percent_format(accuracy = 0.1)) +
  labs(title = "Δ proportion (OpenAccess − CBNRM)", x = NULL, y = NULL) +
  theme_minimal(base_size = 11) +
```

```
  theme(axis.text.x = element_blank(),
        axis.ticks.x = element_blank(),
        panel.grid.minor = element_blank(),
        plot.title = element_text(size = 11, hjust = 0))

# Bottom: heatmap (NDMI, CCI, NDVI)
p_heat <- ggplot(M_long, aes(x = cluster_f,
                             y = factor(metric, levels = c("NDMI","CCI","NDVI")),
                             fill = value_z)) +
  geom_tile() +
  scale_fill_gradient2(name = "z-score",
                       low = "#3b5b92", mid = "white", high = "#1a7f37") +
  labs(x = "Clusters (sorted by |Δ|)", y = NULL) +
  theme_minimal(base_size = 11) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, size = 7),
        panel.grid = element_blank(),
        legend.position = "bottom",
        legend.direction = "horizontal",
        legend.box = "horizontal")

# Stack + global title
final <- (p_delta + labs(title = NULL)) /
  (p_heat  + labs(title = NULL)) +
  plot_layout(heights = c(0.9, 4), guides = "collect") +
  plot_annotation(
    title = sprintf("Local spectral index composition by cluster — %s (K=%d, %s)",
                    tile_code, m$K, m$mode),
    subtitle = "Columns are k-means clusters (sorted by |Δ|); colors show standardized NDVI, CCI, NDMI
of cluster centers"
  ) &
  theme(plot.title = element_text(hjust = 0.5),
        plot.subtitle = element_text(hjust = 0.5))

# Save
ggsave(OUT_PNG, final, width = 12, height = 7.5, dpi = 300)
message("Saved: ", OUT_PNG)
```

```
# ============================
# PASS LA — Local ΔS heatmap (10 bands only; single pass)
# ============================
suppressPackageStartupMessages({
  library(arrow); library(dplyr); library(data.table); library(fs)
  library(ggplot2); library(scales); library(patchwork)
})

# ---------- INPUT ROOT ----------
tile_path      <-      "C:/Users/david/OneDrive/Documents/Master      ESS/Masterarbeit/Data/Sentinel
2/S2B_MSIL2A_20240616T073619_N0510_R092_T37MBN_20240616T100238.SAFE"

safe_name <- basename(tile_path)
tile_code <- sub(".*_(T[0-9A-Z]{5})_.*", "\\1", safe_name)
acq_date  <- sub(".*_(\\d{8})T\\d{6}.*$", "\\1", safe_name)
if (identical(tile_code, safe_name)) tile_code <- "TXXXX"
if (identical(acq_date,  safe_name)) acq_date  <- "YYYYMMDD"

LOCAL_BASE <- file.path("C:/Users/david/OneDrive/Documents/Master ESS/Masterarbeit/Data",
                        sprintf("Local_SVA_%s_%s", tile_code, acq_date))
L1_DIR <- file.path(LOCAL_BASE, "L1_BANK")
LA_DIR <- file.path(LOCAL_BASE, "LA_RICHNESS_SENS")
dir_create(LA_DIR, recurse = TRUE)

BANK_PARQUET <- file.path(L1_DIR, "bank_local.parquet")
stopifnot(file.exists(BANK_PARQUET))

# ---------- FORCE 10-BAND MODE ----------
BANDS <- c("B01","B02","B03","B04","B05","B06","B07","B8A","B11","B12")
cols_in_file <- colnames(read_parquet(BANK_PARQUET, as_data_frame = TRUE))
if (!all(BANDS %in% cols_in_file)) {
  missing <- setdiff(BANDS, cols_in_file)
  stop(sprintf("10-band mode requested, but missing bands: %s", paste(missing, collapse=", ")))
}
BAND_MODE <- "10B"
message(sprintf("[LA] Using 10B: %s", paste(BANDS, collapse=", ")))
```

```
# ---------- Grids & caps ----------
K_GRID      <- c(40L, 60L, 80L, 100L, 120L, 150L)
THRESH_GRID <- c(0.00, 0.005, 0.01, 0.02, 0.03, 0.05)
TRAIN_PER_GROUP_MAX <- 30000L
SEED <- 42L
set.seed(SEED)

# ---------- Helpers ----------
zscore_fit        <-   function(X){   mu   <-   colMeans(X,   na.rm=TRUE);   sdv   <-   apply(X,2,sd);
sdv[!is.finite(sdv)|sdv==0] <- 1; list(mu=mu, sd=sdv) }
zscore_apply <- function(X, zs){ sweep(sweep(X,2,zs$mu,"-"), 2, zs$sd, "/") }
predict_kmeans <- function(X, centers){
  xsq <- rowSums(X^2); csq <- rowSums(centers^2); G <- X %*% t(centers)
  D2  <- matrix(xsq, nrow=nrow(G), ncol=ncol(G)) +
    matrix(csq, nrow=nrow(G), ncol=ncol(G), byrow=TRUE) - 2*G
  max.col(-D2, ties.method="first")
}
rarefy_equal <- function(df, group_col="group"){
  tab <- table(df[[group_col]]); if (length(tab) < 2) return(df[0,,drop=FALSE])
  n_take <- min(as.integer(tab))
  idx <- unlist(tapply(seq_len(nrow(df)), df[[group_col]],
                       function(i) sample(i, n_take, replace = FALSE)))
  df[idx,,drop=FALSE]
}
richness_thresholded <- function(p, thr = 0.01, eps = 1e-12) {
  if (thr <= 0) sum(p > eps, na.rm = TRUE) else sum(p >= thr - eps, na.rm = TRUE)
}
get_props_by_group <- function(df_clusters, K, g1="CBNRM", g2="OpenAccess"){
  tab <- as.data.table(df_clusters)[, .N, by=.(group, cluster)]
  get_prop <- function(g){
    v <- numeric(K); tg <- tab[group==g]
    if (nrow(tg)) v[as.integer(tg$cluster)] <- tg$N
    v / sum(v)
  }
  list(C = get_prop(g1), O = get_prop(g2))
}

# ---------- Load bank ----------
DF <- as.data.frame(read_parquet(BANK_PARQUET))
stopifnot(all(c("group", BANDS) %in% names(DF)))
stopifnot(all(DF$group %in% c("CBNRM","OpenAccess")))

DF <- DF[stats::complete.cases(DF[,BANDS,drop=FALSE]) & !is.na(DF$group), , drop=FALSE]
X  <- as.matrix(DF[, BANDS, drop=FALSE])

idxC <- which(DF$group == "CBNRM")
idxO <- which(DF$group == "OpenAccess")
if (!length(idxC) || !length(idxO)) stop("Need both CBNRM and OpenAccess present.")
n_tr <- min(TRAIN_PER_GROUP_MAX, length(idxC), length(idxO))

# Balanced training; eval = ALL pixels → equal rarefied
tr_idx <- c(sample(idxC, n_tr), sample(idxO, n_tr))
X_tr   <- X[tr_idx,,drop=FALSE]; G_tr <- DF$group[tr_idx]

zs <- zscore_fit(X_tr)
Z_tr <- zscore_apply(X_tr, zs)

df_ev <- data.frame(group = DF$group, zscore_apply(X, zs), check.names = FALSE)
df_eq <- rarefy_equal(df_ev, "group")
if (nrow(df_eq) == 0) stop("Equal rarefaction failed (only one group present).")
Z_eq <- as.matrix(df_eq[, colnames(Z_tr), drop=FALSE])

# ---------- Sweep K & τ (single pass) ----------
RESULTS <- list()
for (K in K_GRID) {
  km <- kmeans(Z_tr, centers = K, iter.max = 100, nstart = 5)
  df_eq$cluster <- predict_kmeans(Z_eq, km$centers)
  props <- get_props_by_group(df_eq, K, g1="CBNRM", g2="OpenAccess")
  pC <- props$C; pO <- props$O

  for (thr in THRESH_GRID) {
    S_C <- richness_thresholded(pC, thr)
    S_O <- richness_thresholded(pO, thr)
    RESULTS[[length(RESULTS)+1L]] <- data.table(
```

```
      K = K, threshold = thr,
      S_CBNRM = S_C, S_OpenAccess = S_O,
      delta = S_O - S_C,
      delta_pct = ifelse(S_C > 0, 100 * (S_O/S_C - 1), NA_real_)
    )
  }
}

RES <- rbindlist(RESULTS, fill = TRUE)
CSV_PATH <- file.path(LA_DIR, "local_richness_sensitivity_10B_single.csv")
fwrite(RES, CSV_PATH)

# ---------- Plot heatmap (ΔS only) ----------
RES[, thr_lab := scales::percent(threshold, accuracy = 0.1)]

p_heat <- ggplot(RES, aes(x = factor(K), y = thr_lab, fill = delta)) +
  geom_tile() +
  geom_text(aes(label = round(delta, 1)), size = 3) +
  scale_fill_gradient2(low = "#6baed6", mid = "white", high = "#fb6a4a",
                       midpoint = 0, name = "\u0394S (OA–CB)") +
  labs(
    title    = sprintf("Local Spectral Richness Difference (\u0394S) — %s", tile_code),
    subtitle = "Single-pass on actual data — 10 bands",
    x = "K (clusters)", y = "Presence threshold"
  ) +
  theme_minimal(base_size = 12)

png_path <- file.path(LA_DIR, "heat_deltaS_local_10B_single.png")
ragg::agg_png(filename = png_path, width = 10, height = 7, units = "in", res = 300, background =
"transparent")
print(p_heat)
dev.off()

message("Saved CSV: ", CSV_PATH)
message("Saved heatmap PNG: ", png_path)
```

```
# =========================================================
# LOCAL — Richness (S), Diversity (D1, D2), Evenness (E1, E2) Plots
# Reads balanced tables from L6_RICHNESS_TABLES_BALANCED
# Produces: S bar chart, D1/D2 facets, E1/E2 facets, box+violin, 2x3 panel
# =========================================================
suppressPackageStartupMessages({
  library(readr); library(data.table); library(dplyr); library(tidyr)
  library(ggplot2); library(scales); library(fs); library(patchwork)
})

# ---------- SAFE / paths ----------
tile_path      <-      "C:/Users/david/OneDrive/Documents/Master      ESS/Masterarbeit/Data/Sentinel
2/S2B_MSIL2A_20240616T073619_N0510_R092_T37MBN_20240616T100238.SAFE"
safe_name <- basename(tile_path)
tile_code  <-  sub(".*_(T[0-9A-Z]{5})_.*",  "\\1",  safe_name);  if  (identical(tile_code,  safe_name))
tile_code <- "TXXXX"
acq_date  <- sub(".*_(\\d{8})T\\d{6}.*$", "\\1", safe_name);   if (identical(acq_date,   safe_name))
acq_date  <- "YYYYMMDD"

LOCAL_BASE <- file.path("C:/Users/david/OneDrive/Documents/Master ESS/Masterarbeit/Data",
                        sprintf("Local_SVA_%s_%s", tile_code, acq_date))
IN_DIR  <- file.path(LOCAL_BASE, "L6_RICHNESS_TABLES_BALANCED")
dir_create(IN_DIR, recurse = TRUE)

# τ files (as written by the table script)
files <- c(
  "0.5%" = sprintf("%s_%s_local_richness_tau0.5p_BALANCED.csv", tile_code, acq_date),
  "2.0%" = sprintf("%s_%s_local_richness_tau2.0p_BALANCED.csv", tile_code, acq_date),
  "5.0%" = sprintf("%s_%s_local_richness_tau5.0p_BALANCED.csv", tile_code, acq_date)
)


base_counts_file <- sprintf("%s_%s_local_cluster_counts_BALANCED.csv", tile_code, acq_date)

# ---------- helper: Hill numbers from counts vector ----------
hill_from_counts <- function(x) {
  x <- x[!is.na(x) & x > 0]
  S <- length(x)
```

```
  if (S == 0) return(tibble(S = 0, D1 = NA_real_, D2 = NA_real_, E1 = NA_real_, E2 = NA_real_))
  p <- x / sum(x)
  H <- -sum(p * log(p))
  D1 <- exp(H)
  D2 <- 1 / sum(p^2)
  tibble(S = S, D1 = D1, D2 = D2, E1 = D1 / S, E2 = D2 / S)
}

# ---------- build summary per τ from the balanced tables ----------
summ_by_thr <- lapply(names(files), function(th_label) {
  path <- file.path(IN_DIR, files[[th_label]])
  df   <- read_csv(path, show_col_types = FALSE)

  need <- c("observed_CBNRM","observed_OpenAccess","count_CBNRM","count_OpenAccess")
  if (!all(need %in% names(df))) {
    stop(sprintf("File %s is missing required columns: %s",
                 basename(path), paste(need, collapse = ", ")))
  }

  # mask counts by observed flags (0/1)
  cntC <- ifelse(replace_na(df$observed_CBNRM,      0) > 0,
                 replace_na(df$count_CBNRM,         0), 0)
  cntO <- ifelse(replace_na(df$observed_OpenAccess, 0) > 0,
                 replace_na(df$count_OpenAccess,    0), 0)

  C <- hill_from_counts(cntC) %>% rename_with(~paste0(.x, "_CBNRM"))
  O <- hill_from_counts(cntO) %>% rename_with(~paste0(.x, "_OpenAccess"))
  tibble(threshold = th_label) %>% bind_cols(C, O)
}) %>% bind_rows() %>%
  mutate(threshold = factor(threshold, levels = names(files)))  # 0.5%,2.0%,5.0%

# ==============================
# PRINT & SAVE NUMERIC SUMMARY TABLES
# ==============================

safe_round <- function(x, k = 3) ifelse(is.finite(x), round(x, k), NA_real_)

mk_summary <- function(df) {
  df %>%
    mutate(
      dS       = S_OpenAccess  - S_CBNRM,
      dS_pct   = 100 * dS  / pmax(S_CBNRM, 1e-9),
      dD1      = D1_OpenAccess - D1_CBNRM,
      dD1_pct  = 100 * dD1 / pmax(D1_CBNRM, 1e-9),
      dD2      = D2_OpenAccess - D2_CBNRM,
      dD2_pct  = 100 * dD2 / pmax(D2_CBNRM, 1e-9),
      dE1      = E1_OpenAccess - E1_CBNRM,
      dE2      = E2_OpenAccess - E2_CBNRM
    ) %>%
    transmute(
      threshold,
      S_CBNRM,          S_OA  = S_OpenAccess,   dS,   dS_pct,
      D1_CBNRM,         D1_OA = D1_OpenAccess,  dD1,  dD1_pct,
      D2_CBNRM,         D2_OA = D2_OpenAccess,  dD2,  dD2_pct,
      E1_CBNRM,         E1_OA = E1_OpenAccess,  dE1,
      E2_CBNRM,         E2_OA = E2_OpenAccess,  dE2
    ) %>%
    mutate(across(-threshold, ~ ifelse(is.finite(.x), round(.x, 3), NA_real_)))
}

summary_wide <- mk_summary(summ_by_thr)

# ---- Print to console (wide)
cat("\n================ RICHNESS / DIVERSITY / EVENNESS (OA - CBNRM) ================\n")
print(summary_wide, row.names = FALSE)
cat("==============================================================================\n")

# ---- Also provide a long tidy version (easy to copy to the thesis)
summary_long <- summary_wide |>
  tidyr::pivot_longer(
    -threshold,
    names_to  = "metric",
    values_to = "value"
  )
```

```
# Save both tables
readr::write_csv(summary_wide,                                                    file.path(IN_DIR,
"LOCAL_summary_richness_diversity_evenness_wide.csv"))
readr::write_csv(summary_long,                                                    file.path(IN_DIR,
"LOCAL_summary_richness_diversity_evenness_long.csv"))

# ---- Optional: a minimal “headline” print for each threshold
cat("\n-- Headlines by threshold (Δ = OA - CBNRM) --\n")
apply(summary_wide, 1, function(r){
  cat(
    sprintf("τ=%s | ΔS=%s (%.1f%%), ΔD1=%s, ΔD2=%s, ΔE1=%s, ΔE2=%s\n",
            r[["threshold"]],
            r[["dS"]], as.numeric(r[["dS_pct"]]),
            r[["dD1"]], r[["dD2"]],
            r[["dE1"]], r[["dE2"]])
  )
})
cat("================================================================================\n")

# ---------- add true τ = 0% from base balanced counts (count > 0) ----------
base_df <- read_csv(file.path(IN_DIR, base_counts_file), show_col_types = FALSE)
need0 <- c("count_CBNRM","count_OpenAccess")
stopifnot(all(need0 %in% names(base_df)))

C0 <- hill_from_counts(replace_na(base_df$count_CBNRM,     0)) %>% rename_with(~paste0(.x, "_CBNRM"))
O0   <-   hill_from_counts(replace_na(base_df$count_OpenAccess,   0))   %>%   rename_with(~paste0(.x,
"_OpenAccess"))
t0_row <- tibble(threshold = "0%") %>% bind_cols(C0, O0)

summ_by_thr <- bind_rows(t0_row, summ_by_thr) %>%
  mutate(threshold = factor(threshold, levels = c("0%", names(files))))

# ---------- PLOT COLORS ----------
COLS <- c("CBNRM" = "#1b9e77", "OpenAccess" = "#d95f02")

# ---------- (1) Richness S across thresholds ----------
long_S <- summ_by_thr %>%
  select(threshold, S_CBNRM, S_OpenAccess) %>%
  pivot_longer(-threshold, names_to = "metric_group", values_to = "value") %>%
  separate(metric_group, into = c("metric","group"), sep = "_")

p_S <- ggplot(long_S, aes(x = threshold, y = value, fill = group)) +
  geom_col(position = position_dodge(width = 0.75), width = 0.7) +
  scale_fill_manual(values = COLS) +
  labs(
    title = sprintf("Spectral richness S — %s %s (balanced sample)", tile_code, acq_date),
    x = "Presence threshold", y = "Richness (S)", fill = ""
  ) +
  theme_minimal(base_size = 12)

ggsave(file.path(IN_DIR, "LOCAL_richness_S_by_threshold.png"),
       p_S, width = 10, height = 5.5, dpi = 320)

# ---------- (2) Diversity D1 & D2 (exclude 5%) ----------
exclude_thr <- "5.0%"

# Include all thresholds (0%, 0.5%, 2.0%, 5.0%)
long_div <- summ_by_thr %>%
  dplyr::select(threshold, D1_CBNRM, D1_OpenAccess, D2_CBNRM, D2_OpenAccess) %>%
  tidyr::pivot_longer(-threshold, names_to = "metric_group", values_to = "value") %>%
  tidyr::separate(metric_group, into = c("metric","group"), sep = "_")


ymax_div <- max(long_div$value, na.rm = TRUE)
ymax_div <- ceiling(ymax_div * 1.05)

p_Ddiv <- ggplot(long_div, aes(x = threshold, y = value, fill = group)) +
  geom_col(position = position_dodge(width = 0.75), width = 0.7) +
  facet_wrap(~ metric, scales = "fixed") +
  coord_cartesian(ylim = c(0, ymax_div)) +
  scale_fill_manual(values = COLS) +
  labs(
    title = "Hill diversity (D1 = exp H, D2 = 1/∑p²)",
```

```
    x = "Presence threshold", y = "Diversity", fill = ""
  ) +
  theme_minimal(base_size = 12)

ggsave(file.path(IN_DIR, "LOCAL_hill_diversity_D1_D2_by_threshold.png"),
       p_Ddiv, width = 10, height = 6.2, dpi = 320)

# ---------- (3) Evenness E1 & E2 ----------
long_even <- summ_by_thr %>%
  select(threshold, E1_CBNRM, E1_OpenAccess, E2_CBNRM, E2_OpenAccess) %>%
  pivot_longer(-threshold, names_to = "metric_group", values_to = "value") %>%
  separate(metric_group, into = c("metric","group"), sep = "_")

p_Eeven <- ggplot(long_even, aes(x = threshold, y = value, fill = group)) +
  geom_col(position = position_dodge(width = 0.75), width = 0.7) +
  facet_wrap(~ metric, scales = "free_y") +
  coord_cartesian(ylim = c(0, 1)) +
  scale_fill_manual(values = COLS) +
  labs(
    title = "Evenness (E1 = D1/S, E2 = D2/S)",
    x = "Presence threshold", y = "Evenness (0–1)", fill = ""
  ) +
  theme_minimal(base_size = 12)

ggsave(file.path(IN_DIR, "LOCAL_evenness_E1_E2_by_threshold.png"),
       p_Eeven, width = 10, height = 6.2, dpi = 320)

# ---------- (4) Box + violin of per-cluster counts at τ = 0.5% ----------
tau05_file <- file.path(IN_DIR, files[["0.5%"]])
tbl05 <- read_csv(tau05_file, show_col_types = FALSE)
df_long <- tbl05 %>%
  transmute(
    CBNRM      = as.numeric(count_CBNRM),
    OpenAccess = as.numeric(count_OpenAccess)
  ) %>%
  pivot_longer(cols = everything(), names_to = "group", values_to = "count") %>%
  filter(is.finite(count))

p_box <- ggplot(df_long, aes(x = group, y = count, fill = group)) +
  geom_violin(trim = FALSE, alpha = 0.20, linewidth = 0.3) +
  geom_boxplot(width = 0.25, outlier.alpha = 0.25) +
  scale_fill_manual(values = COLS) +
  scale_y_continuous(labels = comma) +
  labs(
    title = sprintf("Cluster count distribution (balanced) — τ = 0.5%% | %s %s", tile_code, acq_date),
    x = NULL, y = "Pixel count per cluster", fill = ""
  ) +
  theme_minimal(base_size = 12) +
  theme(legend.position = "none")

ggsave(file.path(IN_DIR, "LOCAL_cluster_counts_box_violin_tau0p5.png"),
       p_box, width = 7.5, height = 5.2, dpi = 320)

# ---------- (5) Assemble 2×3 panel ----------
# split diversity/evenness facets to dedicated panels for the grid
p_D1 <- p_Ddiv + facet_wrap(~metric, nrow = 1) & theme()  # we'll filter below
p_D1 <- ggplot(dplyr::filter(long_div, metric == "D1"),
               aes(x = threshold, y = value, fill = group)) +
  geom_col(position = position_dodge(0.75), width = 0.7) +
  coord_cartesian(ylim = c(0, ymax_div)) +
  scale_fill_manual(values = COLS) +
  labs(title = "Diversity D1 = exp(Shannon)", x = "Presence threshold", y = "Diversity", fill = "") +
  theme_minimal(base_size = 12) + theme(legend.position = "bottom")

p_D2 <- ggplot(dplyr::filter(long_div, metric == "D2"),
               aes(x = threshold, y = value, fill = group)) +
  geom_col(position = position_dodge(0.75), width = 0.7) +
  coord_cartesian(ylim = c(0, ymax_div)) +
  scale_fill_manual(values = COLS) +
  labs(title = "Diversity D2 = 1/∑p²", x = "Presence threshold", y = "Diversity", fill = "") +
  theme_minimal(base_size = 12) + theme(legend.position = "bottom")

long_even_plot <- long_even  # (keep all thresholds; y is 0–1 anyway)
```

```
p_E1 <- ggplot(dplyr::filter(long_even_plot, metric == "E1"),
               aes(x = threshold, y = value, fill = group)) +
  geom_col(position = position_dodge(0.75), width = 0.7) +
  coord_cartesian(ylim = c(0, 1)) +
  scale_fill_manual(values = COLS) +
  labs(title = "Evenness E1 = D1 / S", x = "Presence threshold", y = "Evenness (0–1)", fill = "") +
  theme_minimal(base_size = 12) + theme(legend.position = "bottom")

p_E2 <- ggplot(dplyr::filter(long_even_plot, metric == "E2"),
               aes(x = threshold, y = value, fill = group)) +
  geom_col(position = position_dodge(0.75), width = 0.7) +
  coord_cartesian(ylim = c(0, 1)) +
  scale_fill_manual(values = COLS) +
  labs(title = "Evenness E2 = D2 / S", x = "Presence threshold", y = "Evenness (0–1)", fill = "") +
  theme_minimal(base_size = 12) + theme(legend.position = "bottom")

panel_2x3 <- (p_box | p_S) /
  (p_D1  | p_D2) /
  (p_E1  | p_E2) +
  plot_layout(guides = "collect") &
  theme(legend.position = "bottom")

ggsave(file.path(IN_DIR, "LOCAL_panel_2x3_richness_diversity_evenness.png"),
       panel_2x3, width = 14, height = 14, dpi = 320)

message("✅ Wrote plots to: ", IN_DIR)
```

Personal declaration

I hereby declare that the material contained in this thesis is my own original work. Any quotation or paraphrase in this thesis from the published or unpublished work of another individual or institution has been duly acknowledged. I have not submitted this thesis, or any part of it, previously to any institution for assessment purposes.

AI-Statement

In the process of creating this thesis, I used ChatGPT (Model 4 and 5) by OpenAI as a supportive tool during both the coding and writing stages. I oriented myself on AI use with the AI Guidelines from the Department of Geography University of Zurich. During coding, it assisted in translating pseudocode into functional R scripts, explaining error messages, and suggesting ways to streamline or comment my scripts. For the writing process, I used ChatGPT to help refine the clarity and flow of sentences, particularly to avoid overly long or convoluted phrasing which is something I tend to struggle with.

At no point was the AI used to generate research data, perform analyses, or write sections independently. All interpretations, analytical decisions, and final formulations are my own. The use of AI served purely as a technical and editorial aid, comparable to language editing or coding support tools.

24.10.2025 8142 Uitikon

Signature