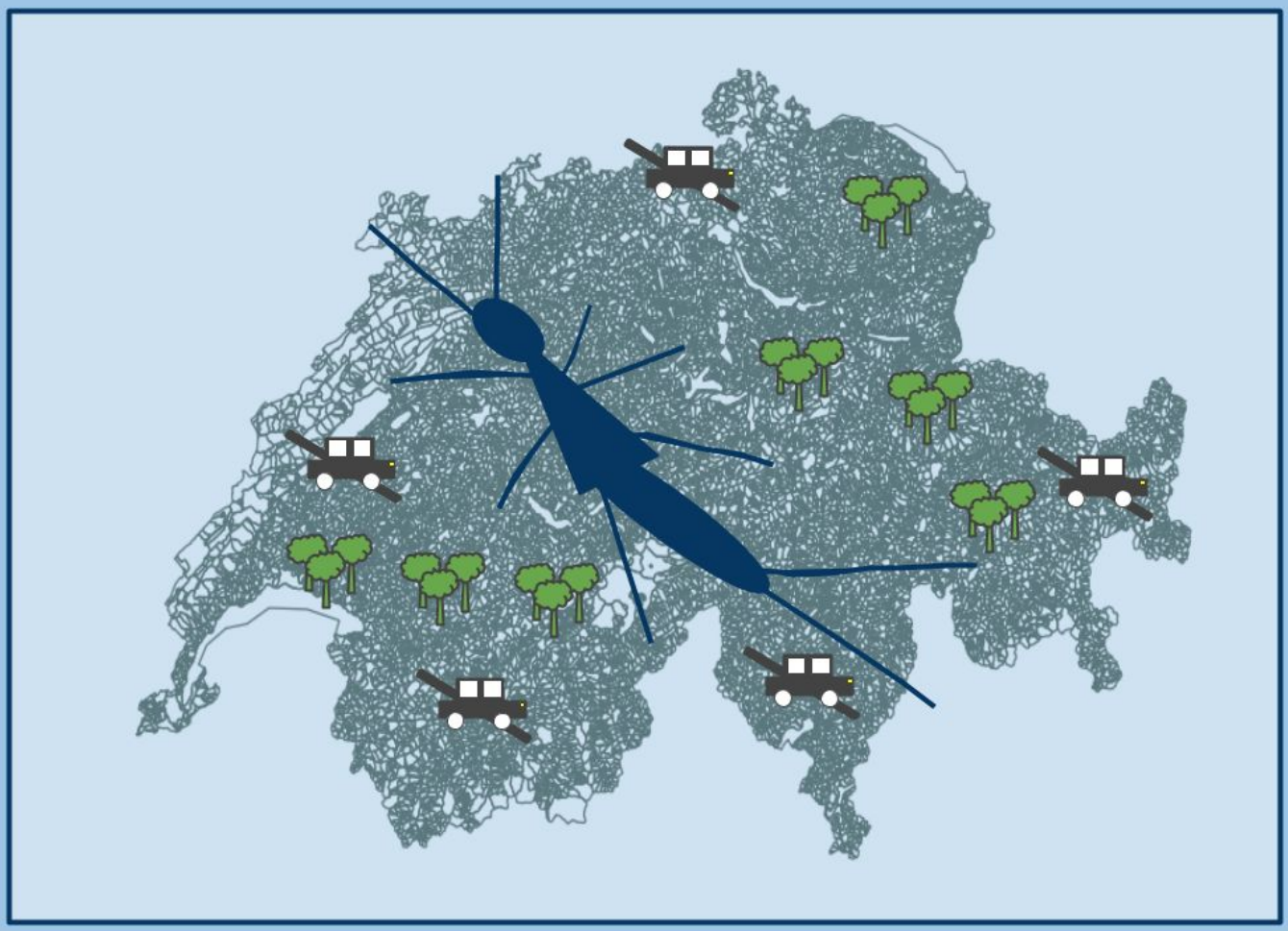
Modelling aquatic macroinvertebrate richness distribution at landscape level in Swiss watercourse networks using environmental variables

Katharina Kaelin (09-729-385)
Department of Geography, UZH
Master thesis GEO 511, 30.9.2015



Supervised by
Prof. Florian Altermatt (Eawag/ UZH) and Prof. Jan Seibert (UZH)

Abteilung Aquatische Ökologie
Eawag
Überlandstrasse 133
CH-8600 Dübendorf
florian.altermatt@eawag.ch

Geographisches Institut
Universität Zürich - Irchel
Winterthurerstrasse 190
CH-8057 Zürich
jan.seibert@geo.uzh.ch

Table of content

Source of Swiss catchment map on cover image: Bundesamt für Umwelt (n.d): Einzugsgebietsgliederung Schweiz EZGG-CH

Abstract

Aquatic biodiversity in rivers and streams is threatened in many regions worldwide. As biodiversity loss has severe consequences on environmental processes it is important to understand the cause of decline and to predict the state of change of biodiversity in space and time. In this study a spatially explicit habitat distribution model for aquatic macroinvertebrates in Swiss watercourse networks was developed using the national biodiversity monitoring data (BDM). The aim was to identify the spatial environmental variable datasets available nationwide, which explain the diversity of macroinvertebrates, in order to predict and upscale their nationwide distribution.

The modelling was carried out with a generalized linear model that related the richness of the Ephemeroptera, Plecoptera and Trichoptera species (EPT) and the IBCH family and higher order taxa (indice biologique global normalisé - IBGN, adapted to Swiss watercourses) to 38 environmental variables. The model parameters were then used to predict the EPT species and IBCH taxa richness at both levels to the landscape scale.

The results show that land-use (forest, pasture, cultivated land and developed area) and topology (elevation and slope) variables influence the Swiss macroinvertebrate richness distribution at a landscape level. However, they do not influence all macroinvertebrate orders equally: the EPT species react more sensitively towards land-use changes than the IBCH taxa, resulting in opposing spatial predictions on richness at the landscape scale. This indicates firstly, that the diversity pattern of one macroinvertebrate group cannot be used as a proxy of another macroinvertebrate group and secondly, that a better understanding of the the relationship between environmental variables and macroinvertebrate richness is gained when the focus is placed on a few sensitive macroinvertebrate species with a clearly defined ecological niche (eg. EPT species), than when a broad mixture of macroinvertebrate taxa with varying sensitivity and less clearly defined ecological niche (eg. IBCH taxa) are considered.

1. Introduction

Taxa richness distribution is characterized by spatial heterogeneity. Even though the spatial patterns of various taxa are increasingly well documented, the understanding of why areas vary in taxa richness imposes challenges (Gaston 2000). Often, environmental variables and spatial patterns have been cited as primary determinants of taxa richness distribution. Environmental variables influence the habitat area and spatial patterns affect dispersal. The relative importance of these determinants is thought to vary considerably across communities and regions (Lin et al. 2013).

Changes in taxa richness alter ecosystem processes and affect the resilience of ecosystems to environmental changes (Chapin 2000). Hence, improving the understanding of taxa richness distribution is important, especially with regard to the observed impact of human induced environmental changes on taxa richness distribution. Currently observed environmental changes have an especially large impact on freshwater taxa richness distribution. 65% of the global watercourse habitats are threatened (Vörösmarty 2010) and freshwater habitats experience a far greater decline than observed for habitats in the most affected terrestrial ecosystems (Dudgeon 2006). Given that freshwater supports almost 6% of all described species despite only covering 0.8% of the earth's surface (Dudgeon 2006) it is important to understand the cause of decline.

The documentation of general taxa richness distribution by classical sampling techniques in the field is logistically very challenging. Thus, as a simplification and best possible approximation, a subset of bioindicator taxa is used. Bioindicator taxa are easily studied and their pattern are likely to be representative of many other species (Pearson & Carroll 1998). Watercourse macroinvertebrates (hereafter simply referred to as macroinvertebrate) are often used bioindicators to study watercourses: their sampling and identification is relatively easy, their sensitivity and immobility makes them vulnerable to unfavorable local environments and their role in the aquatic food web is vital (EPA 2012). Often the number of Ephemeroptera, Plecoptera and Trichoptera species (EPT) and the indice biologique global normalisé (IBGN) are used to quantify the macroinvertebrate taxa and to reflect the richness (i.e. biodiversity) at a watercourse site.

For a long time macroinvertebrates were studied in a non-spatial perspective (Altermatt 2013). Thereby, environmental variables were thought to be the major determinants of macroinvertebrate richness distributions. Observed macroinvertebrate richness distributions were related to instream habitat features and surrounding land-use characteristics (eg. Death and Winterbourn 1995, Kennen 1999). Later, macroinvertebrates were mostly studied in a linear spatial perspective (eg. Grubaugh et al. 1996, Delong & Brusven 1998). It was chiefly the river continuum concept (Vannote et al. 1980) that lead to the change of perspective. This concept states that fluvial geomorphic processes have to be considered when trying to understand the macroinvertebrate richness distribution: the continuous geomorphic changes from headwater to lower reaches alter the habitat characteristics. Only recently, ecologists recognized that the dendritic networks of watercourses influence the macroinvertebrate richness distribution by modifying dispersal (eg. Carrara et al. 2012, Altermatt et al. 2013). Due to data scarcity, macroinvertebrates are still mainly studied at small spatial scales (watercourse section scale, linear spatial perspective). As larger scale studies (landscape scale, dendritic network spatial perspective) shed light on different processes (Heino et al. 2003) their inclusion is likely to improve the holistic understanding of the macroinvertebrate richness distribution.

A previous study in the Rhine catchment in Switzerland related the EPT taxa richness distribution to environmental variables and spatial patterns at a landscape scale (Seymour et al. 2015). The environmental variables and spatial pattern data were collected with help of a geographic information system (GIS). Environmental variables considered in this study were: elevation, calcareous rock exposure, land-use type (agriculture, settlement, wooded areas, meadows, other types), mean annual temperature and mean annual precipitation. Spatial patterns that were considered are pairwise-distances between sampling sites measured as Euclidean distance and as topological distance. The EPT data were collected by the biodiversity monitoring of Switzerland (BDM). The study tried to disentangle the effects of environmental variables and spatial pattern by decomposing the source of observed variability with variance partitioning. Most variation, in this study, was explained by spatial pattern (mean = 9.4%, SD = 3.1). Both environmental variables (mean = 5.4%, SD = 1.4) and environmental-spatial variation (mean = 6.2%, SD = 3.0) explained less variation. The low explanatory power of environmental variables overall was explained by the exclusion of relevant environmental variables or by stochastic and historic effects that shape taxa richness.

To check whether relevant environmental variables were excluded a more extensive set of environmental variable data should be analyzed. Taxa habitat distribution models provide a good tool to quantify the relationship between species and environment at landscape scale. They have experienced a rapid usage increase since their first application in 1970 (Guisan & Thuiller 2005). This is mainly attributed to advances in statistics and GIS (Guisan & Zimmermann 2000).

This master thesis conducts a spatially explicit nationwide macroinvertebrate habitat distribution model for the EPT species and IBGN taxa in Swiss watercourse networks, with help of regression analysis. The aim is to identify spatial environmental variable datasets available nationwide that explain the diversity of macroinvertebrates, in order to predict their nationwide distribution. Figure 1 demonstrates the approach. First, the regression parameters are estimated by associating monitored macroinvertebrate richness data to environmental variables available at the sampling site vicinity. Secondly, the estimated parameters are used to predict the macroinvertebrate richness distribution nationwide. The regression analysis is conducted using geographical information systems (ArcGIS: Version 10.2.2, ESRI 2015; QGIS: Version 2.8.2-Wien, QGIS Development Team 2015) and the programming languages R and Python (R: Version 0.98.107; R Core Team 2014; Python: Version 2.7.9; Python Team 2014).

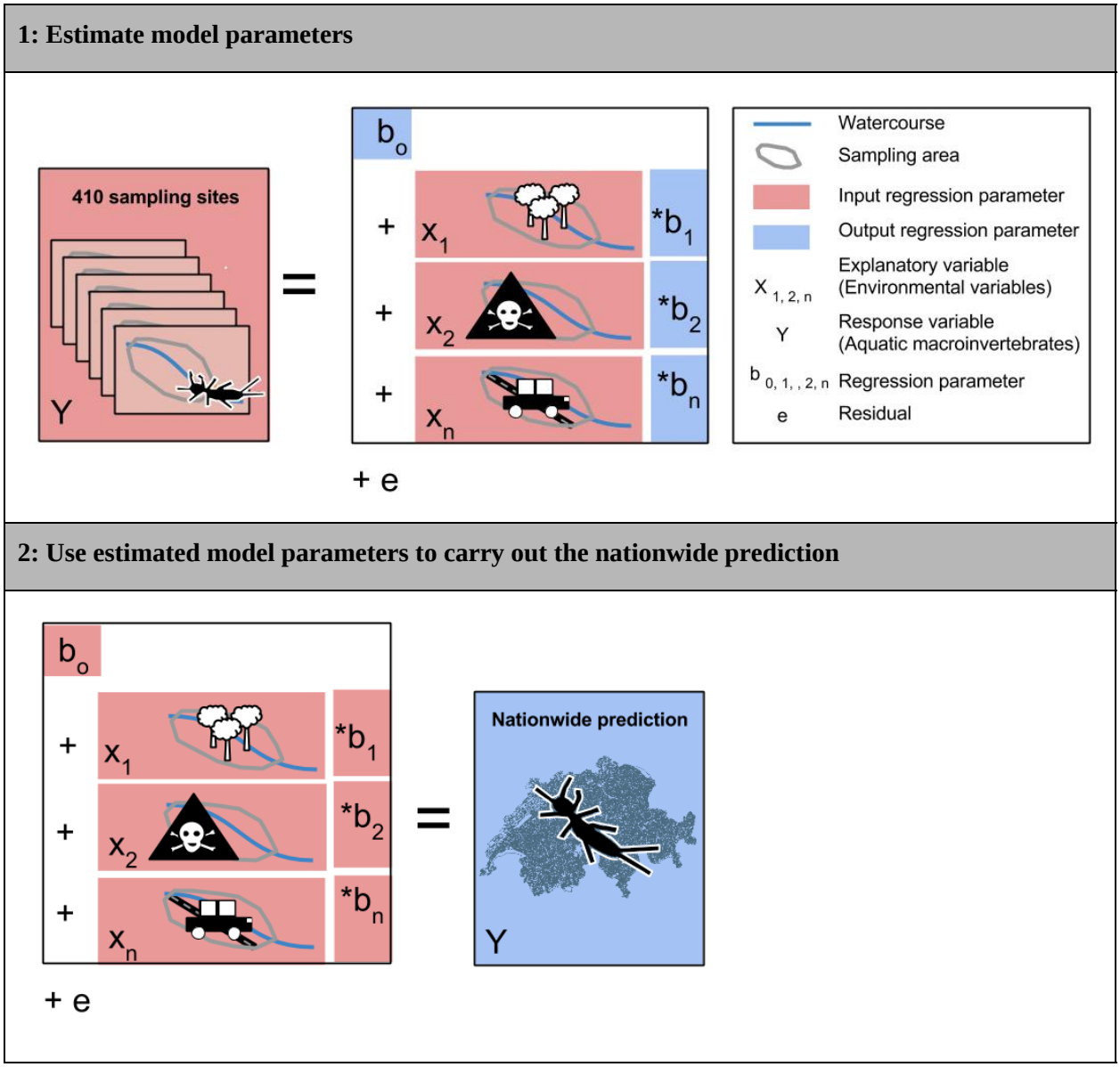


Figure 1: Regression Model. 1: Estimation of the regression parameters; 2: Utilization of estimated regression parameters to carry out the nationwide prediction. *(Source of Switzerland image: Swisstopo (n.d.): swissBoundaries3D, Hoheitsgrenze)*

2. Data and methods

The following chapter describes the material and methods that are applied in this study (Figure 2). First, the macroinvertebrate dataset is introduced (chapter 2.1). Secondly, the selection of the sampling area (chapter 2.2) and the selection of the environmental variable datasets are described (chapter 2.3). Thirdly, the model building, selection, prediction and evaluation are discussed (chapter 2.4). All modelling steps were carried out with programmed scripts to automate repetitive steps and ensure reproducibility (Appendix 1).

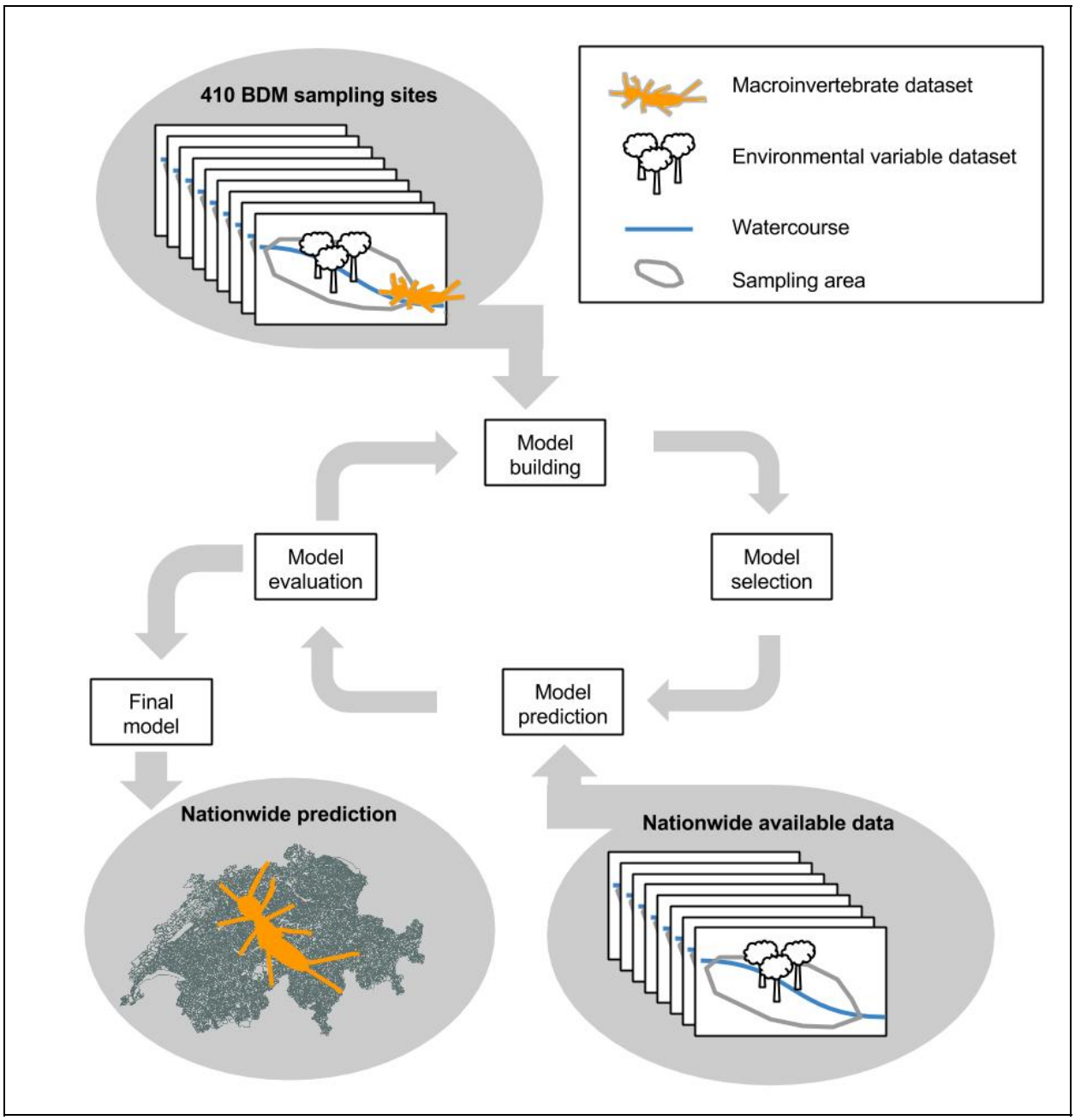


Figure 2: Workflow. The final model is obtained using model building, selection, prediction & evaluation. *(Source of Switzerland image: Swisstopo (n.d.): swissBoundaries3D, Hoheitsgrenze)*

2.1. Response variable: macroinvertebrate dataset

The nationwide macroinvertebrate richness distribution dataset of Switzerland is acquired by biodiversity monitoring (BDM; chapter 2.1.1.). In this study the number of EPT species and the number of IBGN taxa was used (chapter 2.1.2). For technical reasons, the coordinates of the BDM sampling sites do not always lie on a watercourse, even though the watercourse is sampled. To correct for this small inconsistency and to assign each BDM point to a watercourse the raw data is cleansed (chapter 2.1.3).

2.1.1. Biodiversity monitoring of macroinvertebrates

Since 2010 the nationwide BDM observes macroinvertebrates in watercourses at 570 sampling sites (Figure 3) (Stucki 2010, Koordinationsstelle BDM 2014). Yearly, a distinct and random 20% of the 570 BDM sampling sites are monitored. At the time this study was carried out, 410 monitored sites have been monitored. The sampling sites are chosen with help of a systematic sampling grid. Considering that the origin of the systematic sampling grid is chosen at random, it can be statistically treated like a random sample. This approach has the advantage that the regional subsamples are proportional to the size of the regions. Only watercourses that appear on the swiss 1:25´000 map and have a Strahler number (measure of watercourse branching complexity) of at least two are monitored. The width of the sampling site is defined by the width of the watercourse. The upstream length of the sampling site is defined as having ten times the length of the watercourse width. Within this sampling site 8 probes are taken at different habitat types and flow velocities. At some especially diverse habitats 4 additional probes are taken. The sum of the probes is considered. Each BDM sampling site has an unique identification number which is formed by the first three digits of the X-and Y-coordinate (CH1904, EPSG-Code 21781) of the most downstream point of the sampling site. The coordinates are taken in the middle of the watercourse. Probes are collected with help of a kicknet-samplings. For each taxa the presence and absence within the sampling site is recorded.

2.1.2. Macroinvertebrate considered in this study

The BDM monitors the larvae of the orders Ephemeroptera, Plecoptera and Trichoptera (EPT) and the larvae of the taxa that are needed to calculate the to Swiss watercourses adapted IBGN (IBCH) (Stucki 2010, Koordinationsstelle BDM 2014). While the EPT orders are identified to the species level, most of the 142 macroinvertebrate taxa which are used to calculate the IBCH index are identified at family level. Some hard to identify taxa, however, are only distinguished at phyla, class or order level. In this study the number of EPT species and the number of IBCH taxa was used. In comparison to the EPT species, the IBCH taxa are easier to identify. Moreover, they have a larger ecological niche and are less well-defined.

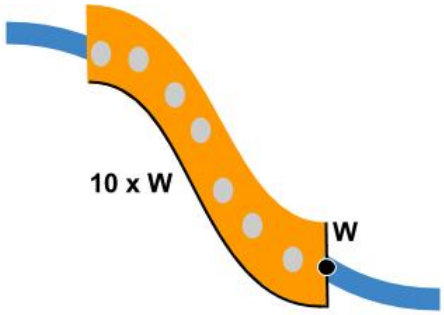
Nationwide distribution of the BDM sampling sites	Single BDM sampling site in more detail
	
BDM sampling site	W Width of the watercourse BDM probe Most downstream BDM probe of sampling site Watercourse BDM sampling site

Figure 3: BDM sampling sites. Left side: nationwide distribution of the BDM sampling sites; Right side: single BDM sampling site in more detail. *(Source of nationwide distribution of BDM sampling sites image: Koordinationsstelle BDM 2014)*

2.1.3. Spatial data cleansing of the BDM macroinvertebrate dataset

Due to measurement inaccuracies not all BDM sampling site coordinates lie on watercourses. Therefore, the BDM sampling sites had to be assigned to the nearest feature of the watercourse dataset (WC; Swisstopo (2007): Vector 25, Gewässernetz). This was done with GIS. To check whether the assignment to a watercourse is unambiguous two checks were carried out:

1) Find the three nearest watercourses (Near_FID) of the GPS coordinate and check whether their distances to the GPS coordinate (In_FID) are similar by dividing the second nearest distance by the nearest distance and the third nearest distance by the nearest distance. If the result of the division is near one (<5) verify the position of the sampling point visually.

2) Find the nearest watercourse to the GPS coordinate (Near_FID) and check whether the difference between the X- or Y- coordinates of the GPS coordinates (In_FID) and the closest watercourse is >1m. If this is the case verify the position of the sampling point visually.

The checks demonstrated that the watercourse assignment was ambiguous for two BDM sampling sites. These BDM sampling sites were moved to the nearest watercourse.

2.2. Sampling area

The borders of the sampling area which will be used to compute the environmental variables (hereafter referred to as sampling area) have to enclose all ground surface points that affect the macroinvertebrate habitat conditions at the BDM sampling sites. These borders are located within the total catchments of the BDM sampling points (blue area in Figure 4 that does not have a purple outline). In the following section, the catchment datasets (chapter 2.2.1), the sampling area for the nationwide prediction (chapter 2.2.2), the sampling area for the BDM sampling sites (chapter 2.2.3) and the problems involved in these steps (chapter 2.2.4) are described.

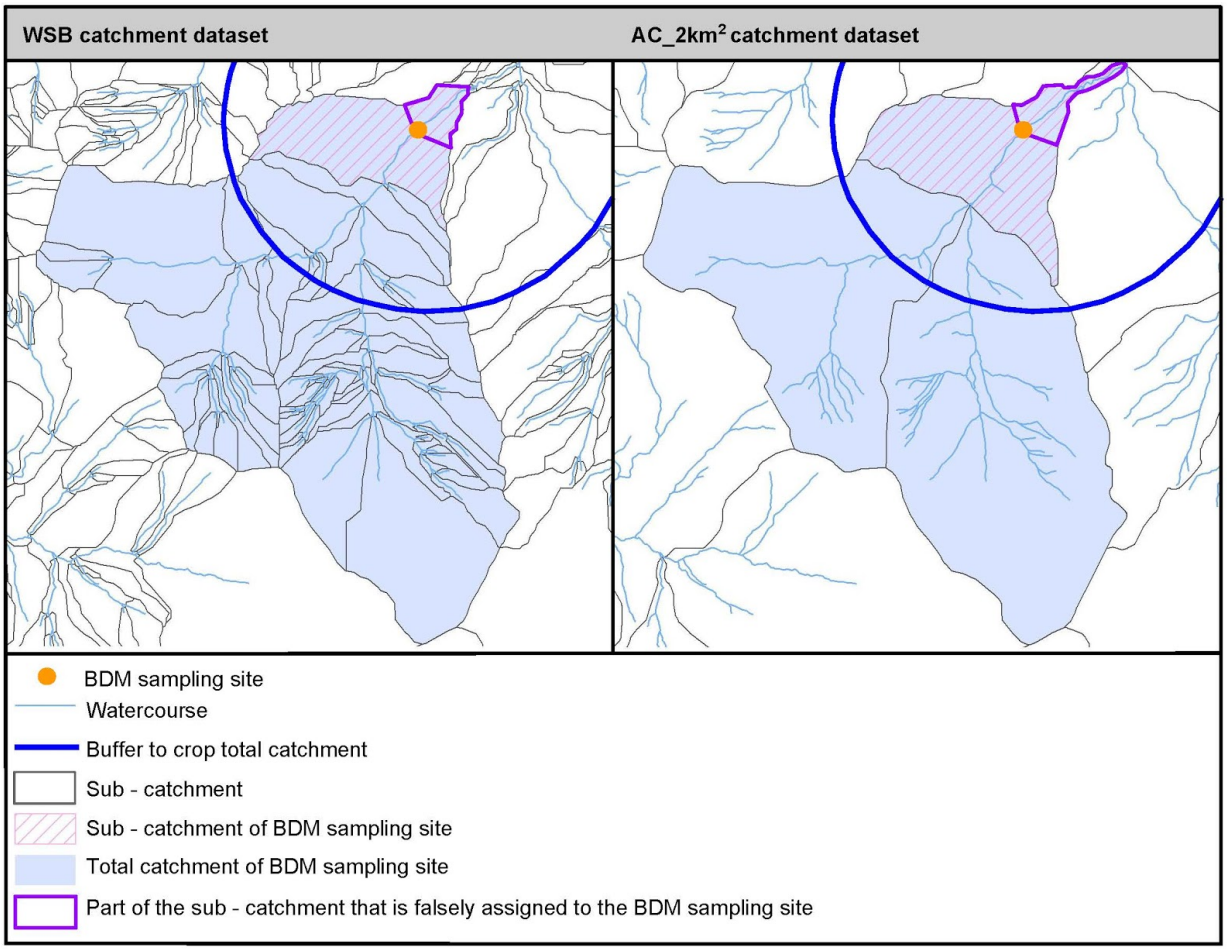


Figure 4: Available nationwide catchment datasets; Left side: WSB; Right side: AC_2km² (*Source: WSB = Bundesamt für Umwelt (n.d.): Gewässerabschnittsbasierte Einzugsgebietsgliederung der Schweiz GAB-EZGG-CH; AC_2km² Bundesamt für Umwelt (n.d): Einzugsgebietsgliederung Schweiz EZGG-CH; WC = Swisstopo (2007): Vector 25, Gewässernetz)*

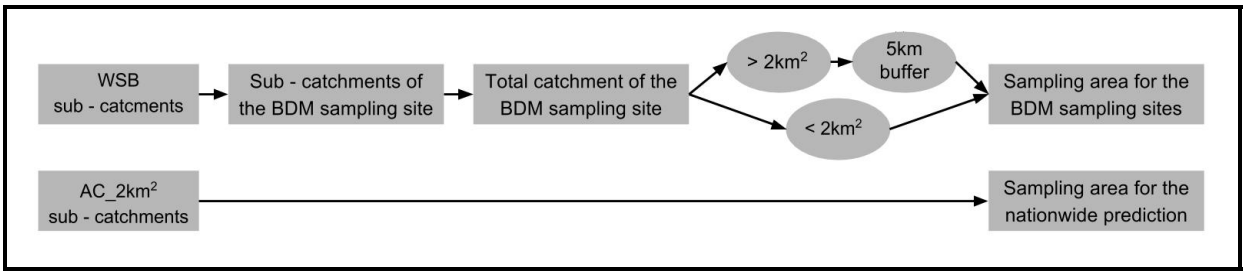


Figure 5: Sampling area definition. Above: sampling area for BDM sampling sites; Below: sampling area for nationwide prediction.

2.2.1. Catchment datasets

The Federal Office for the Environment of Switzerland (FOEN) created two nationwide catchment datasets. They provide a basis for different hydrological and water management studies. The first catchment dataset is a watercourse segment based dataset (WSB; Bundesamt für Umwelt (n.d.): Gewässerabschnittsbasierte Einzugsgebietsgliederung der Schweiz GAB-EZGG-CH). The second catchment dataset is an aggregated catchment dataset (AC; Bundesamt für Umwelt (n.d): Einzugsgebietsgliederung Schweiz EZGG-CH) (Table 1).

Table 1: Comparison of the available nationwide catchment datasets; Left side: WSB; Right side: AC.

	Watercourse segment based dataset (WSB)	**Aggregated dataset (AC)**
Description	The WSB dataset contains a catchment for each overground and underground river and creek segment. Segments start and end where watercourses intersect and/or where watercourses change their watercourse characteristic attribute (eg. underground creek, overground river).	The AC dataset is an aggregated catchment dataset. Catchments can be aggregated to several aggregation levels: 2km^2, 40km^2, 150km^2 and 1000km^2.
Application	Useful to determine the catchments of small watercourses.	Useful to determine the catchments of large watercourses.
Number of features	In total 181´182 catchments.	In total 22’169 catchments.

Making use of this official datasets offers the advantage of comparability between different catchment-based studies. Moreover, several useful environmental variables have already been calculated for these catchment datasets. Thus, the idea of recalculating the catchments by means of a digital terrain models (DTM) is rejected. Nevertheless, making use of the existing catchment datasets brings some disadvantages. The biggest disadvantage is that the BDM sampling points do not lie on the outlet of the catchment datasets. This means that a part of the catchment is mistakenly assigned to the BDM sampling site although it lies downstream of it (purple outlined area in Figure 4). Since the sub-catchments of the AC dataset are larger than the sub-catchments of the WSB dataset, the

mistakenly assigned area is larger for the AC dataset (Figure 4). An inaccurate definition of the sampling area results in an inaccurate estimate of the environmental variables and hence in an inaccurate model. Therefore, the sampling area should preferably be defined with the smaller sub-catchments of the WSB dataset. Using the WSB dataset for the nationwide prediction, however, is not considered to be appropriate since the sample size of the BDM sampling sites is too small to make a meaningful prediction for all catchments of the WSB dataset (in total 181´182 catchments). For this reason the $2km^2$ aggregated AC dataset (AC_2km^2) was chosen to carry out the nationwide prediction (in total 22'169 catchments) (Figure 6).

2.2.2. Sampling area for the nationwide prediction (Nationwide prediction sampling area)

The nationwide prediction was carried out for each sub-catchment of the $2km^2$ aggregated AC catchment dataset (AC_2km^2) at the outlet (Figure 4 and 6). Seymour et al. (2014) found that the total variation explained for the EPT species diversity remained relatively consistent for different sampling area sizes. Therefore it is assumed that the influence of the environmental variables on the macroinvertebrates is negligible beyond an area of $2km^2$.

2.2.3. Sampling area for the BDM sampling sites (BDM sampling area)

The sampling areas of the BDM sampling sites were defined with help of a GIS approach. In the following the three steps to define the sampling area for the BDM sampling sites are described:

1) Assign a sub-catchment to each BDM sampling site (sub-catchment of the BDM sampling site; pink shaded area in Figure 4): Using a spatial query the BDM sampling sites were assigned to the sub-catchment that surrounds them. To assure that the sampling sites are assigned to the correct sub-catchment the watercourse ID (WC OBJECTID) that was joined to the BDM sampling site (chapter 2.1.3) and the watercourse ID that is associated with the sub-catchment of the WSB dataset (OBJECTID_G) were compared. These IDs did not match for 29 BDM sampling sites. This is mostly due to the shift of the watercourse and catchment junctions: The calculation of a sub-catchment requires the definition of an outlet which has to be transferred back to a junction in order to get a meaningful result (personal communication, with Ivo Strahm, October 25, 2014). However, the non-existence of the watercourse ID in the catchment dataset also led to an ID mismatch (sub-catchments are not calculated for every watercourse). The most suitable watercourse ID was applied to each such BDM sampling site after a visual inspection.

2) Generate the total catchments of the BDM sampling sites (total catchment of BDM sampling site; blue area in the Figure 4): The total catchments of the BDM sampling sites were generated with help of the nested set model attributes of the sub-catchments of the BDM sampling sites (h1 and h2). Considering the total catchment might however be unsuitable since the influence of the environmental variables diminishes with distance (Walker et al. 2012).

3) Think about the appropriate size of the BDM sampling area: The sampling area for the nationwide prediction and the sampling area for the BDM sampling sites should be area-wise comparable. Therefore, the total catchments of the BDM sampling sites that have a total

catchment surface area < 2km^2 (182 BDM sampling sites) were not be reduced. The total catchments of the BDM sampling sites that have a total catchment surface area > 2km^2 (228 BDM sampling sites), however, were reduced in order to assure area-wise comparability and the distance-wise diminution of the environmental variables. The sub-catchments of the total catchments that lie in the vicinity of the BDM sampling points can be located with buffers that are placed around the BDM sampling points (dark blue line in Figure 4). Buffers are a frequently used tool to carry out proximity analyses (ESRI 2014). The environmental variables used in this study are considered to have an influence of several kilometers. Thus, circular 5km and 10km buffers around the BDM sampling points were compared (Figure 7 and Table 2). All sub-catchments of the total catchments that are partly overlapped by buffers were considered to have an influence on the BDM sampling points water condition. If only sub-catchment of the total catchments that are entirely overlapped by buffers are considered to be influential, the form of the sub-catchment plays an important role (Figure 8). While in comparison to the 10km buffers, the number of cropped catchments is much higher for the 5km buffers, the number of catchments that have a small area and do not overlap is smaller for the 5km buffers than for the 10km buffers. Since the area of the catchments of the BDM sampling sites should not be much larger than 2km^2 and overlaps between the catchments are inappropriate for statistical reasons, the 5km buffers were selected.

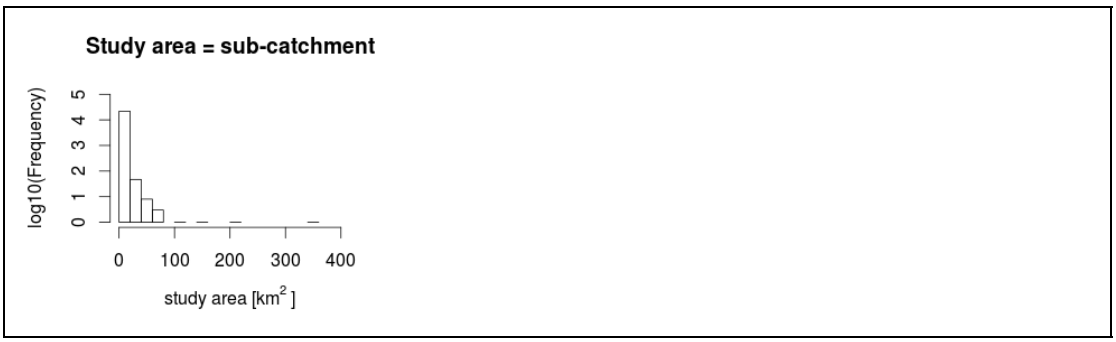


Figure 6: Surface area histogram of the nationwide prediction sampling areas. Only sub-catchments that lie within Switzerland are considered (21'825 sub-catchments).

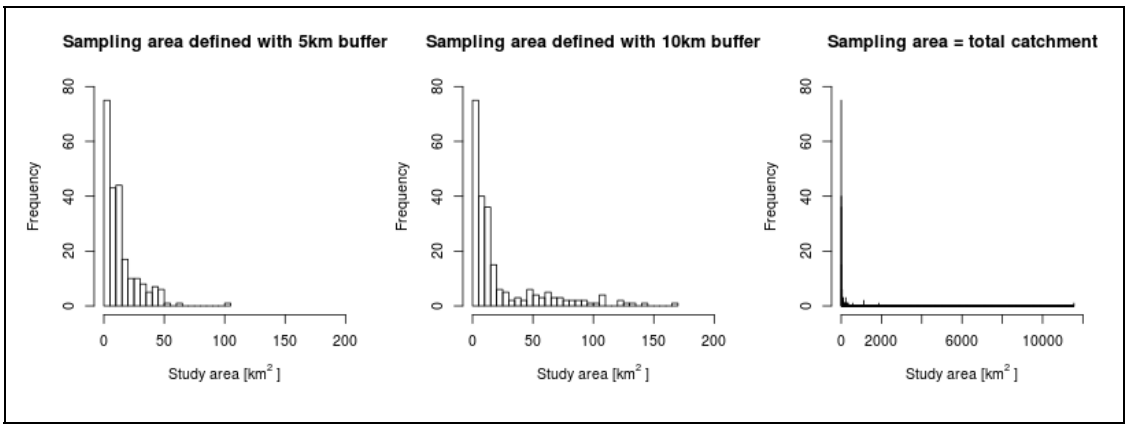


Figure 7: Surface area histograms of different possibilities to define the BDM sampling areas > 2km2 (228 BDM sampling sites). Left: BDM sampling areas that are defined with help of a 5km buffer; Middle: BDM sampling areas that are defined with help of a 10km buffer; Right: BDM sampling areas that are defined using the total catchment.

Table2: Comparison of 5km- and 10km buffer radius to define the BDM sampling areas.

Number of BDM sampling areas that...	**5km buffer**	**10km buffer**
… are cropped	74	39
… have an area > 20km^2	49	62
… have an area > 50km^2	3	38
… overlap	7	24

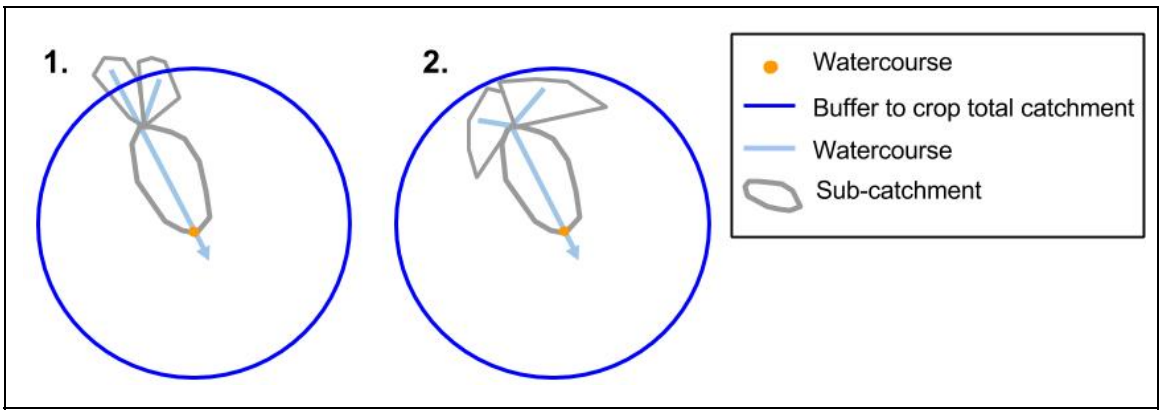


Figure 8: Influence of the catchment form. 1: If only sub-catchments that are entirely overlapped by buffers are considered to be influential one sub-catchment is selected; 2: If only sub-catchments that are entirely overlapped by buffer are considered to be influential all sub-catchments are selected.

2.2.4. Problems

The procedure described above to calculate the BDM sampling areas is a simple automated procedure and causes some problems which are described below:

1) The BDM sampling points do not lie on the outlet of the catchment datasets (purple outlined area in Figure 4). Therefore, the environmental variables might not be representative for the BDM sampling points. However Tobler's first law of geography states that "Everything is related to everything else, but near things are more related than distant things" (Tobler 1970). Since environmental variables mostly correlate in space it is assumed that this problem is negligible for most environmental variables. There are, however, some man-made environmental variables in this study that do not correlate in space (dam_count, disposalsite_2004_percentage, hydropower_count, stormsewage_m3.a, wastewater_m3.a). The presence and absence of these variables in the BDM sampling areas were checked visually. If frequent GIS analysis with the BDM sampling points are planned in the future, the BDM sampling points should be moved to the outlets of the catchments.

2) The result is dependent on the accuracy of the catchment data. During a visual control of the catchments that are assigned to the BDM sampling sites some inaccuracies of the WSB dataset are found: The nested set model attributes do not always return the total catchment,

the sub-catchments are not calculated for all watercourse segments, the junctions of the watercourse dataset and the sub-catchment dataset often do not match and some watercourses lie on and also cross the watershed line.

3) Buffers do not do a good job at cropping the nearest watercourses when the watercourse meanders. This is mostly the case in flat areas. Since the proximity of the watercourses cannot be determined easily with another method the described approach was nevertheless favoured.

4) Scale dependency. It is important to remember that the optimal scale to capture macroinvertebrates is likely to be different for each environmental variable. The influence of the environmental variables is moreover affected by the spatial variation of the physical, ecological and land-use variables within the catchment (Sliva & Williams 2001).

5) Disagreement between the catchment datasets: The boundaries of the AE and WSB catchments do not always agree. Therefore, the macroinvertebrate richness cannot be predicted for every AC_2km^2 dataset.

Some of the problems described above might be corrected manually. With the exception of problem 1, however, no manual correction was carried out since it would be hard to justify why certain corrections were made while others were omitted. Nevertheless, the catchments where problems are noted were documented.

2.3. Explanatory variables: environmental variables

Only environmental variables for which nationwide spatial data are available were considered in this study. The area-wide nationwide coverage is a mandatory requirement for the nationwide prediction. Many previous studies primarily related macroinvertebrates to instream habitat features (eg. Aguiar et al. 2002, Heino et al. 2003, Miserendino 2001). As nationwide spatial instream habitat feature data are rare this study mainly related macroinvertebrates to land-use data. After extensive data research 38 environmental variables were selected. These variables are characterized by ecological relevance and previous usage in literature (eg. Richards et al. 1997, Sliva & Williams 2001, Sawyer et al. 2004, Egler et al. 2012, Wahl et al. 2013, Seymour et al. 2015) (Table 3). Chapter 2.3.1 describes the procedure to obtain the explanatory variables. Chapter 2.3.2 relates the explanatory variables to the observed macroinvertebrate richness and chapter 2.3.3 explains how an appropriate number of explanatory variables was chosen.

Table 3: Summary of the explanatory variables used in this study: name, description and source.

Explanatory variables			
1	**area_bdm_m2 [m^2]** Area of sampling area *Dataset: WSB (V), Attribute: A_SUBEZG*	**20**	**Masl [m.a.s.l]** Mean meters above sea level of sampling area *Dataset: swissALTI3D (R)*
2	**area_total_m2 [m^2]** Area of total catchment *Dataset: WSB (V), Attribute: A_SUBEZG*	**21**	**potato_percentage [%]** Proportion of potato cultivation area within sampling area *Dataset: WSB (V), Attribute: KART*
3	**canal_percentage [m]** Length of canal watercourse within sampling area *Dataset: WC (V), Attribute: OBJECTVAL = Kanal*	**22**	**Q_amax_m3.s [%]** Maximum annual discharge within sampling area *Dataset: MQ-GWN-CH (V), Attribute: mqn_Jan, mqn_Feb, mqn_Mar, mqn_Apr, mqn_Mai, mqn_Jun ,mqn_Jul, mqn_Aug, mqn_Sep, mqn_Okt, mqn_Nov, mqn_Dez*
4	**carbonate_per_carbonatesilicate [%]** Proportion of carbonate rock in sampling area: carbonate area/(carbonate area+silicate area) *Dataset:* Watercourse typification source data: geology, *Attribute: GEO*	**23**	**Q_amean_m3.s [%]** Mean annual discharge within sampling area *Dataset: MQ-GWN-CH (V), Attribute: mqn_Jah*
5	**cereal_percentage [%]** Proportion of cereal cultivation area within sampling area *Dataset: WSB (V), Attribute: GETR*	**24**	**Qvar_amean_m3.s [%]** Mean annual discharge variability within sampling area *Dataset: MQ-GWN-CH (V), Attribute: abflussvar*
6	**corn_percentage [%]** Proportion of corn cultivation area within sampling area *Dataset: WSB (V), Attribute: MAIS*	**25**	**rapeseed_percentage [%]** Proportion of rapeseed cultivation area within sampling area *Dataset: WSB (V), Attribute: RAPS*

7	**dam_count [count]** Number of dams within sampling area *Dataset: Dam (V)*	**26**	**roof_percentage [%]** Proportion of roof area within sampling area *Dataset: WSB (V), Attribute: DACH*
8	**decidious_per_forest [%]** Proportion of deciduous forest area in sampling area: deciduous forest area/(deciduous forest+coniferous forest) *Dataset: Forest type (R), Attribute: VALUE*	**27**	**rootvegetable_percentage [%]** Proportion of root vegetables cultivation area within sampling area *Dataset: WSB (V), Attribute: RUEB*
9	**disposalsite_190207_percentage [%]** Proportion of disposal site area within sampling area *Dataset: WSB (V), Attribute: DEPO*	**28**	**settlement_percentage [%]** Proportion of settlement area within sampling area *Dataset: WSB (V), Attribute: SIED*
10	**disposalsite_2004_percentage [%]** Proportion of disposal site area within sampling area *Dataset: Disposal site, 2004 (R)*	**29**	**slope_max [%]** Maximum watercourse gradient within sampling area *Dataset:* Watercourse typification source data: slope *(V), Attribute: slope*
11	**facade_percentage [%]** Proportion of building facade area within sampling area *Dataset: WSB (V), Attribute: FASS*	**30**	**slope_mean [%]** Mean watercourse gradient within sampling area *Dataset:* Watercourse typification source data: slope *(V), Attribute: slope:*
12	**facaderoof_percentage [%]** Proportion of building shell area within sampling area *Dataset: WSB (V), Attribute: FA_DACH*	**31**	**stormsewage_m3.a [m^3/a]** Storm sewage quantity which is annually conveyed into watercourse within sampling area *Dataset: WSB (V), Attribute: MW*
13	**field_percentage [%]** Proportion of cultivated area within sampling area *Dataset: WSB (V), Attribute: ACK*	**32**	**street_percentage [%]** Proportion of street area within sampling area *Dataset: WSB (V), Attribute: STR_A*
14	**floodplainwetland_percentage [%]** Proportion of floodplains and wetlands within sampling area *Dataset: Floodplain & Wetland (V)*	**33**	**track_percentage [%]** Proportion of railway track area within sampling area *Dataset: WSB (V), Attribute: GEL_A*
15	**forest_percentage [%]** Proportion of forest area within sampling area *Dataset: WSB (V), Attribute: WALD*	**34**	**vegetable_percentage [%]** Proportion of vegetable cultivated area within sampling area *Dataset: WSB (V), Attribute: GEM*
16	**fruit_percentage [%]** Proportion of orchard area within sampling area *Dataset: WSB (V), Attribute: OBST*	**35**	**vine_percentage [%]** Proportion of vineyard area within sampling area *Dataset: WSB (V), Attribute: REB*

17	**green_percentage [%]** Proportion of green area (Ley, Alpine- & Jura pasture, natural meadows) within sampling area *Dataset: WSB (V), Attribute: GRUE*	**36**	**wastewater_m3.a [m^3/a]** Wastewater quantity which is annually conveyed into watercourse within sampling area *Dataset: WSB (V), Attribute: ARA*
18	**hydropower_count [count]** Number of hydropower plants within sampling area *Dataset: Hydropower (V)*	**37**	**watercourse_bdm_m [m]** Length of watercourses within sampling area *Dataset: WSB (V), Attribute: FLSTR_SUBE*
19	**legume_percentage [%]** Proportion of legume cultivation area within sampling area *Dataset: WSB (V), Attribute: HUEF*	**38**	**watercourse_total_m [m]** Length of watercourses within total catchment *Dataset: WSB (V), Attribute: FLSTR_SUBE*

Abbreviations	
R	Raster data
V	Vector data
Dataset source	
Dam	Bundesamt für Energie (2013): Stauanlagen unter Bundesaufsicht
Disposal site	Swisstopo (2007): Vector 25, Gewässernetz
Floodplain & Wetland	Bundesamt für Umwelt (n.d.): Bundesinventar der Auengebiete von nationaler Bedeutung Bundesamt für Umwelt (n.d.): Bundesinventar der Flachmoore von nationaler Bedeutung Bundesamt für Umwelt (n.d.): Bundesinventar der Hoch- & Übergangsmoore von nationaler Bedeutung
Forest type	Bundesamt für Statistik (n.d.): Waldmischungsgrad der Schweiz
Watercourse (WC)	Swisstopo (2007): Vector 25, Gewässernetz
Hydropower	Bundesamt für Energie (2012): Statistik der Wasserkraftanlagen (WASTA)
MQ-GWN-CH	Bundesamt für Umwelt (2013): MQ-GWN-CH
swissALTI3D	Swisstopo (n.d.): swissAlti3D
Watercourse segment based catchment (WSB)	Bundesamt für Umwelt (n.d.): Gewässerabschnittsbasierte Einzugsgebietsgliederung der Schweiz GAB-EZGG-CH mit zusätzlicher Darstellung von Landnutzungsdaten (Swisstopo (2010): Vector25, Vector-25 Daten; Bundesamt für Statistik (n.d.): Areal-Statistik; Bundesamt für Statistik (n.d.): Landwirtschaftliche Betriebszählung 2008, Ackerkulturen; Amtliche Vermessung Schweiz / FL (n.d.): DM.01-AV_CH, Daten der Amtlichen Vermessung; Swisstopo (n.d.): swissBUILDINGS 3D, Gebäude; Bundesamt für Umwelt (n.d.): ARA-Datenbank, ARA-Daten)
Watercourse typification & source data	Bundesamt für Umwelt (n.d.): Fliessgewässertypisierung (Swisstopo (2007): VECTOR25, Swisstopo (n.d.): Digitales Höhenmodell dtm-AV - neu swissAlti3D; Bundesamt für Umwelt (2013): MQ-GWN-CH) mit zusätzlichen Gewässerdaten (slope: Swisstopo (n.d.): swissAlti3D; geology: Schweizerische geotechnische Kommission (n.d.): Vereinfachte geotechnische Karte der Schweiz)

2.3.1. Procedure to obtain explanatory variables

The environmental variables of the sampling areas were all obtained using GIS and R. With exception of *decidious_per_forest*, *Masl* and *disposalsite_2004_percentage* all environmental variables were obtained from vector data. The procedure to obtain them can be broadly classified into four groups (Table 4). Each procedure must work for the BDM sampling area and the nationwide prediction sampling area. The accuracy of the calculation of the environmental variables that are obtained from raster images depends upon the cell size resolution. This study used a cell size resolution of 10m. Since the environmental variables are spatially extensive a decrease in cell size does not improve the result. As described in chapter 2.2.4 the presence and absence of the man-made environmental variables were checked visually for the BDM sampling areas.

Table 4: GIS procedure to obtain explanatory variables.

Procedure of how to obtain WSB and FGT environmental variables (with exception of *area_total_m2* and *watercourse_total_m*)
❖ Join the WSB, MQ_GWN_CH and FGT attributes to the WSB features based on the common attribute field ❖ Transform the WSB polygon features to point features ❖ Select all point feature that lie within the sampling area and calculate the desired environmental variables
Procedure of how to obtain *dam_count, hydropower_count, canal_percentage, geology* and *floodplainwetland_percentage*
❖ Make sure that the sampling areas do not overlap ❖ Use the “Intersect” tool of ArcGis ❖ Append the results and calculate the desired environmental variables
Procedure of how to obtain *decidious_per_forest, Masl* and *disposalsite_2004_percentage*
❖ Use the “Tabulate Area” and “Zonal Statistics” tool of ArcGis
Procedure how how to obtain *area_total_m2* and *watercourse_total_m*
❖ Spatially join the WSB attributes to the BDM sampling sites

2.3.2. Relating the explanatory variables to the observed macroinvertebrate richness

The appendix (Appendix 2 & 3) visualizes the relationship between each explanatory variable and the observed EPT species and IBCH taxa richness at sampling areas. *Forst_percentage, decidious_per_forest, carbonate_per_carbonatesilicate* and *green_percentage* have the highest correlation coefficients for the EPT species (> 0.2). *Mas*l and *carbonate_per_carbonatesilicate* have the highest correlation coefficient for the IBCH taxa (> 0.5).

2.3.3. Choosing appropriate number of explanatory variables for the prediction

As a rule of thumb no more than n/3 (where n = number of response variables, in this study n = 410) explanatory variables should be related to the response variable during multiple regressions (Crawley 2007). If all explanatory variables and their two-way interactions are included 741 regression parameter values would have to be estimated with the data (38*37/2+38). Interactions occur when an explanatory variable has a different effect on the outcome depending on the values of another explanatory variable (Crawley 2007). The presence of interactions can have important implications on the model and thus should not be neglected. Since it is impossible to estimate more regression parameter values than response variables the number of explanatory variables had to be reduced drastically. In this study two steps were carried out to reduce the number of explanatory variables:

1) In a first step only explanatory variables that can be obtained for most BDM sampling areas were considered. As the discharge variables (*Q_amax_m3.s, Q_amean_m3.s and Qvar_amean_ m3.s*) are only available for approximately 40% of the BDM sampling areas they were excluded from this study (Appendix 4 and 5).

2) In a second step the correlating explanatory variables should be eliminated. Explanatory variables that correlate with each other tell the same story, even if the relationship is spurious (Crawley 2007). Since, according to the Shapiro-Wilk test none of the explanatory variables were normally distributed (Appendix 6) the Kendall and Spearman test was used to identify correlating explanatory variables (‘cor’ function of R; R Core Team 2014) (Appendix 7 & 8). If the correlation coefficients were < -0.7 or > 0.7 it was assumed that the variables correlate strongly with each other (Rumsey 2011). The number of strongly correlating variables was larger for the Spearman correlation test (69 correlating variables) than for the Kendall test (42 correlating variables). All variables that strongly correlate with each other in the Kendall correlation test also strongly correlated with each other in the Spearman correlation test. These variables are listed in Table 5. Where possible the most powerful explanatory variables per correlation group were selected considering data quality. Hence, disposalsite_2004_ percentage and wastewater_m3.a are selected since disposalsite_190207_percentage was calculated with an obsolete dataset and stormsewage_m3.a was modelled on the basis of wastewater_m3.a (Table 5: A, B). For explanatory variables that cannot be easily selected on the basis of data quality information (Table 5: C, D, E) tree models were carried out with help of the tree model function of R (‘tree’; Brian Ripley 2015) (Appendix 9 & 10). Tree models give guidance about which explanatory variables to include by indicating which explanatory variables explain the greatest amount of deviance in the response variable (Crawley 2007). For each correlation group the explanatory variable that explains the highest amount of deviance in the response variable was selected (Table 6). As the correlation group E is large, two explanatory variables were selected for this group.

Table 5: Correlating explanatory variables.

Name	Correlating explanatory variables
A	*disposalsite_190207_percentage, disposalsite_2004_percentage*
B	*stormsewage_m3.a, wastewater_m3.a*
C	*area_bdm_m2, area_total_m2, watercourse_bdm_m, watercourse_total_m*
D	*facade_percentage, facaderoof_percentage, roof_percentage, settlement_percentage*
E	*cereal_percentage, corn_percentage, field_percentage, legume_percentage, potato_percentage, rapeseed_percentage, rootvegetable_percentage, vegetable_percentage*

Table 6: Result of tree model for correlating explanatory variables

Response variable	Correlating explanatory variables
EPT	*watercourse_bdm_m, roof_percentage, corn_percentage, vegetable_percentge*
IBCH	*area_bdm_m2, facade_percentage, rapeseed_percentage, vegetable_percentage*

3) In a third step a tree model was carried out for all correlating explanatory variables with the highest explanatory power and all non-correlating explanatory variables (Appendix 9 & 10). Table 7 summarizes the explanatory variables that according to the tree model explain the greatest amount of deviance in the response variables. Considering that *slope_max* and *slope_mean* correlate highly according to the Spearman correlation test (correlation coefficient: 0.81) and moderately according to the Pearson correlation test (correlation coefficient: 0.62) only *slope_mean* was considered since it has more explanatory power. Table 8 sums up the explanatory variables that were used to model the macroinvertebrate richness distribution. This explanatory variables are visualized with help of maps in appendix 11. Considering more than two-way interactions would lead to overparameterization. Hence only two-way interactions were considered.

Table 7: Result of tree model for correlating and non-correlating explanatory variables. For the correlating explanatory variables only the variables with the highest explanatory power are considered.

Response variable	**Explanatory variables that, according to step 1 and 2, have the highest explanatory power**
EPT	*carbonate_per_carbonatesilicate, corn_percentage, decidious_per_forest, forest_percentage, green_percentage, Masl, roof_percentage, slope_max, slope_mean, street_percentage, wastewater_m3.a, watercourse_bdm_m*
IBCH	*area_bdm_m2, disposalsite_2004_percentage, facade_percentage, forest_percentage, fruit_percentage, green_percentage, Masl, slope_max, slope_mean, vegetable_percentage, vine_percentage*

Table 8: Most important explanatory variables. Explanatory variables marked in bold visualize the explanatory variables that are considered to be important for the EPT species and the IBCH taxa.

Response variable	**Most important explanatory variables**
EPT	*carbonate_per_carbonatesilicate, corn_percentage, decidious_per_forest,* ***forest_percentage****,* ***green_percentage****,* ***Masl****, roof_percentage,* ***slope_mean****, street_percentage, wastewater_m3.a watercourse_bdm_m*
IBCH	*area_bdm_m2, disposalsite_2004_percentage, facade_percentage,* ***forest_percentage****, fruit_percentage,* ***green_percentage****,* ***Masl****,* ***slope_mean****, vegetable_percentage, vine_percentage*

2.4. Model

This chapter discusses the model building (chapter 2.4.1), the model selection (chapter 2.4.2), the model prediction (chapter 2.4.3) and the model evaluation (chapter 2.4.4)

2.4.1. Model building

Regressions are used to analyse the relationship between a response variable and one or several explanatory variables (Crawley 2007). The simplest and most frequently used regression is the linear regression. Yet, this approach is rarely useful for taxa habitat distribution models for several reasons. First, the response variable is often described by means of presence, cover and count data. These data have properties that violate the basic requirements of linear regressions: the data are strictly bounded (in all cases there is a lower and often an upper limit of possible data range), the variance is not constant and the residuals are not normally distributed. Second, the values of the explanatory variables are often not continuous. Generalized linear models (GLM) have been developed to cope with these problems. They represent a generalization of the classical linear regression methods and are characterized by three important properties: the error structure (specifies the distribution of the residuals), the linear predictor (represents the linear sum of the effects of the explanatory variables) and the link function (relates the mean value of the response variable to its linear predictor). GLMs are often used to model taxa distribution and were also be used in this study.

The response variable in this study is a count data, which implies that the GLM model should be fitted using the Poisson family (Poisson errors and a log link function) (Crawley 2007). To check whether the Poisson distribution is suitable, the model-checking plots (‘plot’ function, R Core Team 2014) of the Poisson and Gaussian family were compared (Figure 9 and 10). The model-checking plots include a plot of residuals against fitted values, a scale–location plot of residuals against fitted values, a normal QQ plot and a plot of residuals against leverages. With help of the plots the heteroscedasticity, non-normal distribution of the models residuals, influential data points and outliers can be detected (Crawley 2007).

The model-checking plots (Figure 9 and 10) indicate that the GLM using a Gaussian family is characterized by better normal distribution for both the EPT species and the IBCH taxa. As the heteroscedasticity and the influential data points and outliers of the GLM using Gaussian and Poisson family are comparable for the EPT species and the IBCH taxa, the Gaussian family was chosen over the Poisson family.

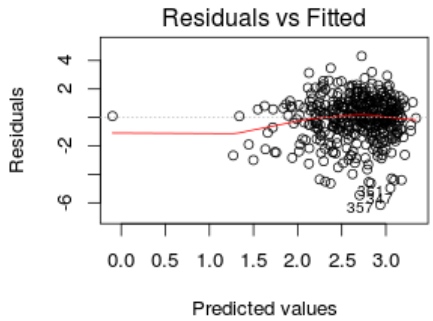
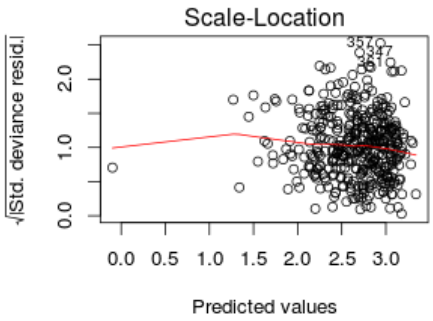
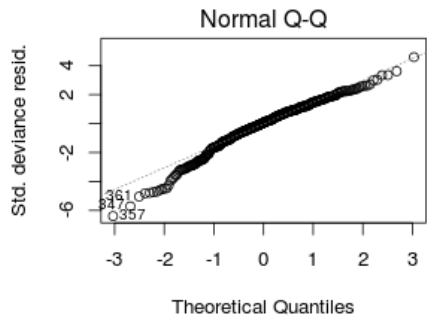
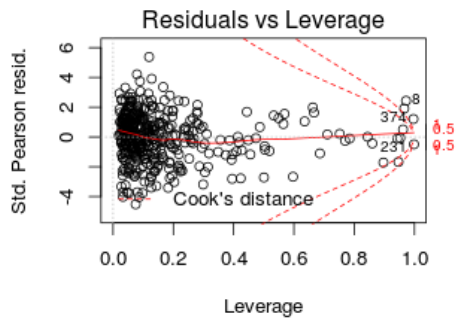
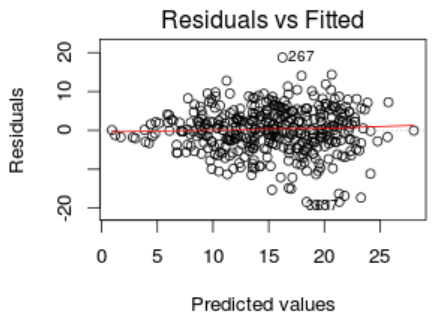
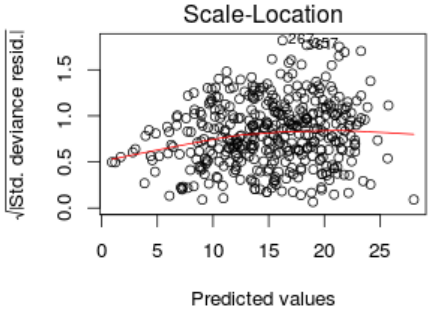
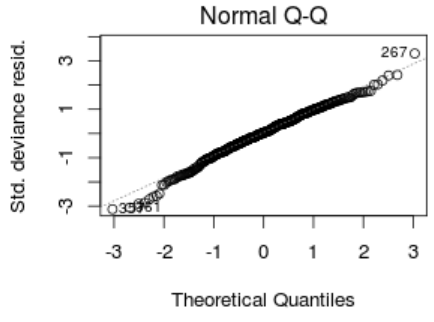
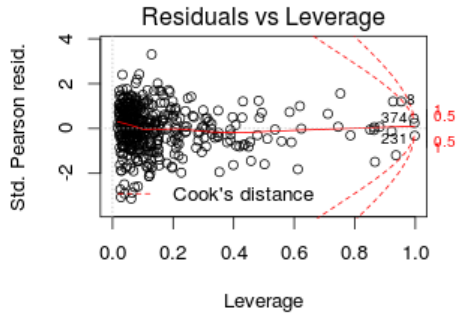
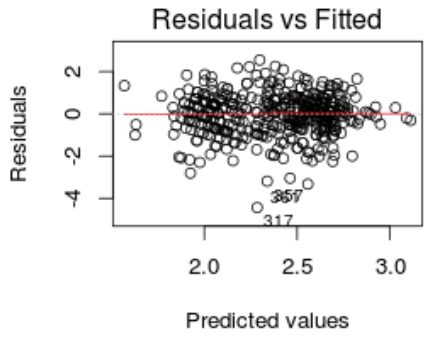
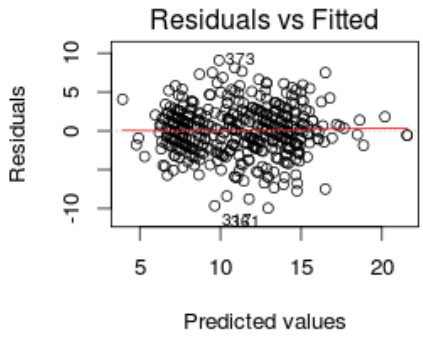
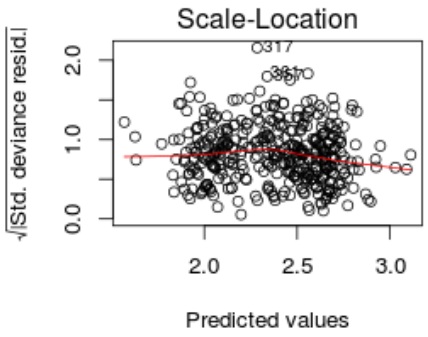
EPT	GLM with Poisson family	GLM with Gaussian family
	   	   

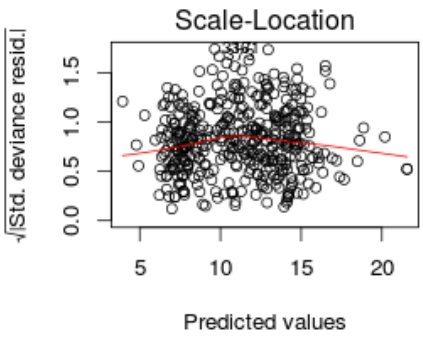
Figure 9: Model checking plots for the EPT species. Left side: GLM with Poisson family; Right side: GLM with Gaussian family. Predicted values and residuals displayed for Poisson family are ln(values).

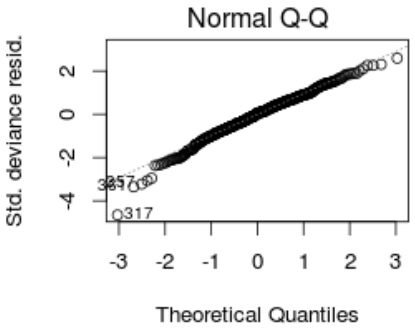
IBCH	GLM with Poisson family	GLM with Gaussian family

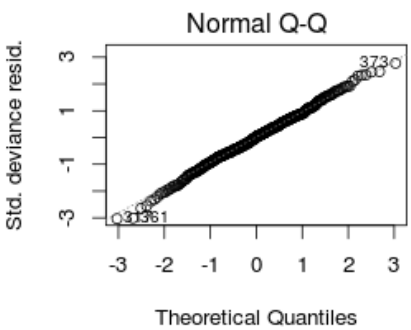


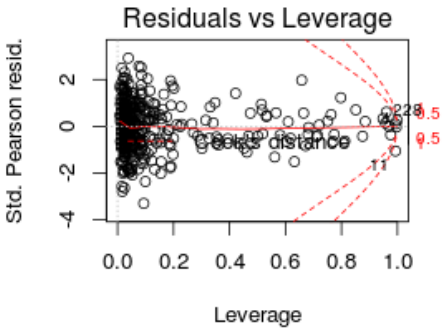












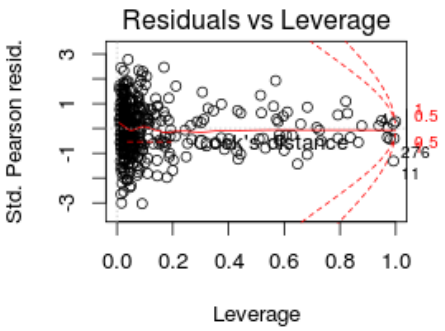


Figure 10: Model checking plots for the IBCH taxa. Left side: GLM with Poisson family; Right side: GLM with Gaussian family. Predicted values and residuals displayed for Poisson family are ln(values).

Only very few zero values were present in the response variable data (EPT species: 1.7% zero values; IBCH taxa: 0.2% zero values). This explains why the GLM with Gaussian family was more suitable for the nationwide macroinvertebrate richness modelling than the GLM with Poisson family. Yet, the usage of the GLM with Gaussian family brings along disadvantages. In contrast to the GLM with Poisson family, the GLM with Gaussian family results in continuous and not strictly bounded predictions (Crawley 2007).

A GLM with Gaussian family is very similar to a multiple linear regression. The difference between the two is the statistical optimization (GLM with Gaussian family: maximum likelihood; Multiple linear regression: least-squares) (Crawley 2007). The GLMs are carried with the 'glm' function of R (R Core Team 2014).

2.4.2. Model selection

According to the principle of parsimony, when choosing among two equivalent explanations, the simpler one is chosen over the more complex one (Crawley 2007). Therefore, the GLMs should not contain any redundant explanatory variables. Model selection methods help to achieve this by applying alternative fitting procedures to the maximum likelihood fitting. Alternative fitting procedures often yield better prediction accuracy and model interpretability (James et al. 2013).

A study that compared different model selection methods for predictive habitat modelling found that no model selection method performed best under all circumstances (Reineking & Schröder 2006). They recommend using stepwise selection methods with caution and indicate that shrinkage methods perform well under a wide range of underlying species-habitat relationships.

In this study a stepwise selection method (backward stepwise selection method) and shrinkage method (lasso model selection method) were carried out and compared. Both methods start with regression models that contain all predictors (James et al. 2013). The backward stepwise selection method iteratively removes the least useful predictor. The lasso selection method shrinks the parameter estimates towards zero which reduces the variance.

2.4.2.1. Backward stepwise selection method

This study used the automated backward selection method of R ('step'; R Core Team 2014) that is based on the Akaike information criterion (AIC). AIC is a model fit measure that penalizes the presence of superfluous parameters. As the step function errs on the side of generosity, a manual model simplification should be carried out after the automated procedure (Crawley 2007). The manual model simplification did not lead to a better AIC value in this study. The macroinvertebrate richness prediction that was carried out with the GLM, which is simplified using the backward stepwise selection method, is referred to as *Step Model* in this study. The GLM output of the *Step Model* are visualized in appendix 12 & 13. The anova table of the *Step Model* are visualized in figure 11, 12.

2.4.2.2. Lasso selection method

The lasso selection method in this study was carried out with the glmnet package of R (Friedman et al. 2010). A ten-fold cross-validation was used to automatically find the tuning parameter (λ) that returns

the smallest mean cross validation error. This method depends on finding the best λ since this parameter controls the impact of the shrinkage penalty (James et al. 2013). The macroinvertebrate richness prediction that was carried out with the GLM, which is simplified using the lasso selection method, is referred to as *Lasso Model* in this study. The *Mean(Step, Lasso) Model* stands for the mean of the *Step* and *Lasso Model*. The GLM output of the *Lasso Model* are visualized in appendix 14 & 15.

2.4.3. Model prediction

The macroinvertebrate richness prediction models are carried out with the R function 'predict.glm' (R Core Team 2014). Table 9 lists all models that have been carried out. The prediction could only be carried out for the sampling areas for which all needed explanatory variable (Table 8) are available. Important explanatory variables are revealed through analysis of deviance tables (anova tests) for GLM fits. The anova tests indicate if the amount of deviance that the environmental variables reduce is statistically significant.

2.4.4. Model evaluation

The concordance correlation, the estimated standard error and the residuals between the *Mean(Lasso,Step)Prediction-BDMsites-WSB* and the recorded BDM macroinvertebrate richness values during the BDM were used to evaluate the model.

The concordance correlation coefficient (Lin 1998) is a measure of agreement between two variables. It determines how far observation pairs deviate from the perfect line of concordance. This coefficient is calculated with help of the 'epi.ccc' function of the R package epiR (Mark Stevenson et al. 2015).

The estimated standard errors are only available for the Step Model. For strongly biased estimates such as the Lasso Model, the standard errors are not meaningful (Goeman 2014). The estimated standard errors of the Step Models are calculated with the 'predict.glm' function of R (R Core Team 2014).

The residuals between the *Mean(Lasso,Step)Prediction-BDMsites-WSB* and the recorded BDM macroinvertebrate richness during the BDM are visualized spatially for each AC_2km^2 catchment and for each large river catchment of Switzerland (Aare, Adda, Adige, Inn, Limmat, Reuss, Rhein, Rhone, Ticino) (Bundesamt für Umwelt (n.d.): Hydrografische Gliederung – nachbearbeitete Version (basis04)).

Table 9: Macroinvertebrate richness prediction models: name and description.

Name	Description
Step Prediction - Nationwide - AC_2km²	Nationwide macroinvertebrate richness prediction that was carried out with a GLM that was simplified using the backward stepwise selection method. The AC_2km² catchment dataset was used for this prediction.
Lasso Prediction - Nationwide - AC_2km²	Nationwide macroinvertebrate richness prediction that was carried out with a GLM that was simplified using the Lasso selection method. The AC_2km² catchment dataset was used for this prediction.
Step Prediction - BDM sites - AC_2km²	Macroinvertebrate richness prediction for the 410 BDM sampling sites that was carried out with a GLM that was simplified using the backward stepwise selection method. The AC_2km² catchment dataset was used for this prediction. The AC_2km² catchments for the 410 BDM sampling sites were determined with help of a spatial join of the AC_2km² catchment dataset with the BDM sampling sites.
Lasso Prediction - BDM sites - AC_2km²	Macroinvertebrate richness prediction for the 410 BDM sampling sites that was carried out with a GLM that was simplified using the Lasso selection method. The AC_2km² catchment dataset was used for this prediction. The AC_2km² catchments for the 410 BDM sampling sites were determined with help of a spatial join of the AC_2km² catchment dataset with the BDM sampling sites.
Step Prediction - BDM sites - WSB	Macroinvertebrate richness prediction for the 410 BDM sampling sites that was carried out with the GLM that was simplified using the backward stepwise selection method. The BDM sampling areas were used for this prediction
Lasso Prediction - BDM sites - WSB	Macroinvertebrate richness prediction for the 410 BDM sampling sites that was carried out with the GLM that was simplified using the Lasso selection method. The BDM sampling areas were used for this prediction
Mean(Lasso, Step) Prediction - BDM sites - WSB	Mean of the *Step Prediction - BDM sites - WSB* and *Lasso Prediction - BDM sites - WSB*

3. Results

3.1 Important environmental variables

The anova tables for the GLM identify the important environmental variables of the recorded macroinvertebrate richness in Swiss watercourses (data from BDM, Swiss Biodiversity Monitoring; referred to as *Monitoring* in the following), by identifying significant residual deviance (p-value < 0.01) (Table 10, 11 and 12). Important environmental variables for the EPT species include, from most important to least, *forest_percentage*, *green_percentage*, *corn_percentage*, *street_percentage*, *decidious_per_forest*, *carbonate_per_carbonatesilicate and watercourse_bdm_m.* Important environmental variables for the IBCH taxa are, in order of importance, *green_percentag*e, *Masl*, *forest_percentage*, *facade_percentage*, *vegetable_percentage*, *slope_mean*, *fruit_percentage and vine_percentage*. Although some environmental variables are important for both the EPT species and the IBCH taxa most environmental variables are important to only one of them. The important variables of the land-use categories arable land (EPT: *corn_percentage*; IBCH: *vegetable_percentage*) and developed area (EPT: *street_percentage*; IBCH: *facade_percentage*) differ for the EPT species and IBCH taxa. The environmental variables *forest_ percentage* and *green_percentage*, in contrast, are very important for both the EPT species and the IBCH taxa. For the EPT species, agriculture (*corn_percentage*) is more important than man-made constructions (*street_percentage*). The opposite holds true for the IBCH taxa: settlements (*facade_percentage*) are more important than agriculture (*vegetable_percentage*). While *Masl* and *slope_mean* are important for the IBCH taxa on their own they are only considered to be important for the EPT species in interaction with other variables. C*arbonate_per_carbonatesilicate* and *decidious_ per_forest* are important for the EPT species but not for the IBCH species.

3.2 Nationwide macroinvertebrate prediction

Figure 11 & 12 and table 10 show the nationwide macroinvertebrate richness predictions (*StepPrediction- Nationwide-AC_2km*^{2}, *LassoPrediction-Nationwide-AC_2km*^{2}). The EPT nationwide prediction maps suggest that the EPT species are most likely to occur in wooded and livestock farming areas. Only a few EPT species occur in cultivated land (Ajoie, Mittelland, Rheintal, Valleys of Wallis, Rheintal and Magadinoebene) or in as glaciated and fin covered areas (high-altitude regions of the Alps). Nonetheless there are some exceptions to this observations. For example, the Step model predicts high EPT species richness occurrence at the Aletsch glacier. Settlements do not seem to heavily impact the presence of EPT species. The *Step* and *Lasso Model*, however, do not agree for the populated areas in lake vicinity (see below). According to the IBCH nationwide taxa prediction map the IBCH taxa richness distribution is mainly determined through the terrrain elevation (explanatory variable Masl). While many IBCH taxa are predicted to be found in low altitudes, few IBCH taxa are predicted to be found at high altitudes. As the valley of Wallis demonstrates, there are some exceptions to this conclusion. This area is characterized by the presence of the cultivated land-use variables *vine_ percentage* and *fruit_percentage*. A comparison between the nationwide prediction of EPT species and of IBCH taxa shows firstly, that the EPT species nationwide predictions are much more detailed and secondly, that the predicted EPT species and IBCH taxa richness values do not correlate spatially.

Table 10: Environmental variables that reduce a significant amount of deviance (p-value < 0.01) of the recorded macroinvertebrate richness during the biodiversity monitoring (EPT species, IBCH taxa) ordered by importance (most to least significant from top to bottom) with indication if the variable has a positive (+) or negative (-) or unknown (?) effect on the macroinvertebrate richness according to the nationwide prediction.

EPT		**IBCH**	
+	*forest_percentage*	**?**	*green_percentage*
+	*green_ percentage*	-	*Masl*
-	*corn_percentage*	**?**	*forest_percentage*
?	*street_percentage*	**?**	*facade_percentage*
?	*decidious_per_forest*	**?**	*vegetable_percentage*
?	*carbonate_per_carbonatesilicate*	**?**	*slope_mean*
?	*watercourse_bdm_m*	-	*fruit_percentage*
		-	*vine_percentage*

3.3 Comparison of the model selection methods

Figure 14 illustrates the comparison between the *StepPrediction-Nationwide-AC_2km²* and the *LassoPrediction-Nationwide-AC_2km²* predictions. The concordance correlation coefficient for the EPT species (0.36) is much higher than the concordance correlation coefficient for the IBCH taxa (0.07). Yet, according to the concordance coefficient classification of McBride (2005) both model predictions are poor. The range of the predicted macroinvertebrate richness is higher for the Step Model (*StepPrediction-Nationwide-AC_2km²*, *StepPrediction-BDMsites-AC_2km²*) than for the Lasso Model (*LassoPrediction-Nationwide-AC_2km²*, *LassoPrediction-BDMsites-AC_2km²*) (This can be seen in greater detail in table 13 and 14).

The large prediction deviations between the *StepPrediction-Nationwide-AC_2km²* and *LassoPrediction-Nationwide-AC_2km²* coincide with high estimated standard errors of the *StepPrediction-Nationwide- AC_2km²* prediction (Figure 14). The range of the estimated standard error of the *StepPrediction-Nationwide-AC_2km²* is larger for the IBCH taxa richness prediction than for the EPT species richness prediction. The spatial distribution of the estimated standard errors are visualized in figure 13. The maps suggest that high estimated standard errors for the EPT specie are often found at lake inflows and outflows. The same pattern was also observed for the IBCH taxa. Large estimated standard errors were, however, also found at other sites (eg. valley of Wallis and Rheintal).

3.4 Evaluation of the prediction at the BDM sampling areas

Figure 15 compares the *StepPrediction-BDMsites-WSB*, the *LassoPrediction-BDMsites-WSB* and the *Mean(Lasso,Step)Prediction-BDMsites-WSB* predictions with the *Monitoring* values. The figure trends suggest that the macroinvertebrate richness is overpredicted by the models when few

macroinvertebrates are present and underpredicted by the models when numerous macroinvertebrates are present. According to McBrides (2005) classification of the concordance correlation coefficient the agreement between the values of the models and the *Monitoring* are poor. They are slightly higher for the *StepPrediction-BDMsites-WSB* (EPT: 0.58; IBCH: 0.64) than for the *LassoPrediction-BDMsites-WSB* (EPT: 0.5; IBCH: 0.52) and slightly higher for the IBCH taxa (*Step Model*: 0.64, *Lasso Model*: 0.52) than for the EPT species (*Step Model*: 0.58, *Lasso Model*: 0.5). The estimated standard error of the *StepPrediction-BDMsites-WSB* prediction do not show a meaningful pattern.

The spatial distribution of the residuals between the values of the *Mean(Lasso,Step)-Prediction-BDMsites-WSB* and the *Monitoring* are illustrate in figure 17 and 18. According to these figures the residuals do not correlate spatially. The largest residuals occur at catchments where most monitoring was carried out (EPT species: Aare, Limmat and Rhein; IBCH taxa: Aare, Rhein and Rhone) (Figure 16). However, also some sites were little was carried out are characterized by high residual values (eg. EPT species: Adige and Inn; IBCH taxa: Adda and Inn).

3.5 Influence of the catchment dataset choice

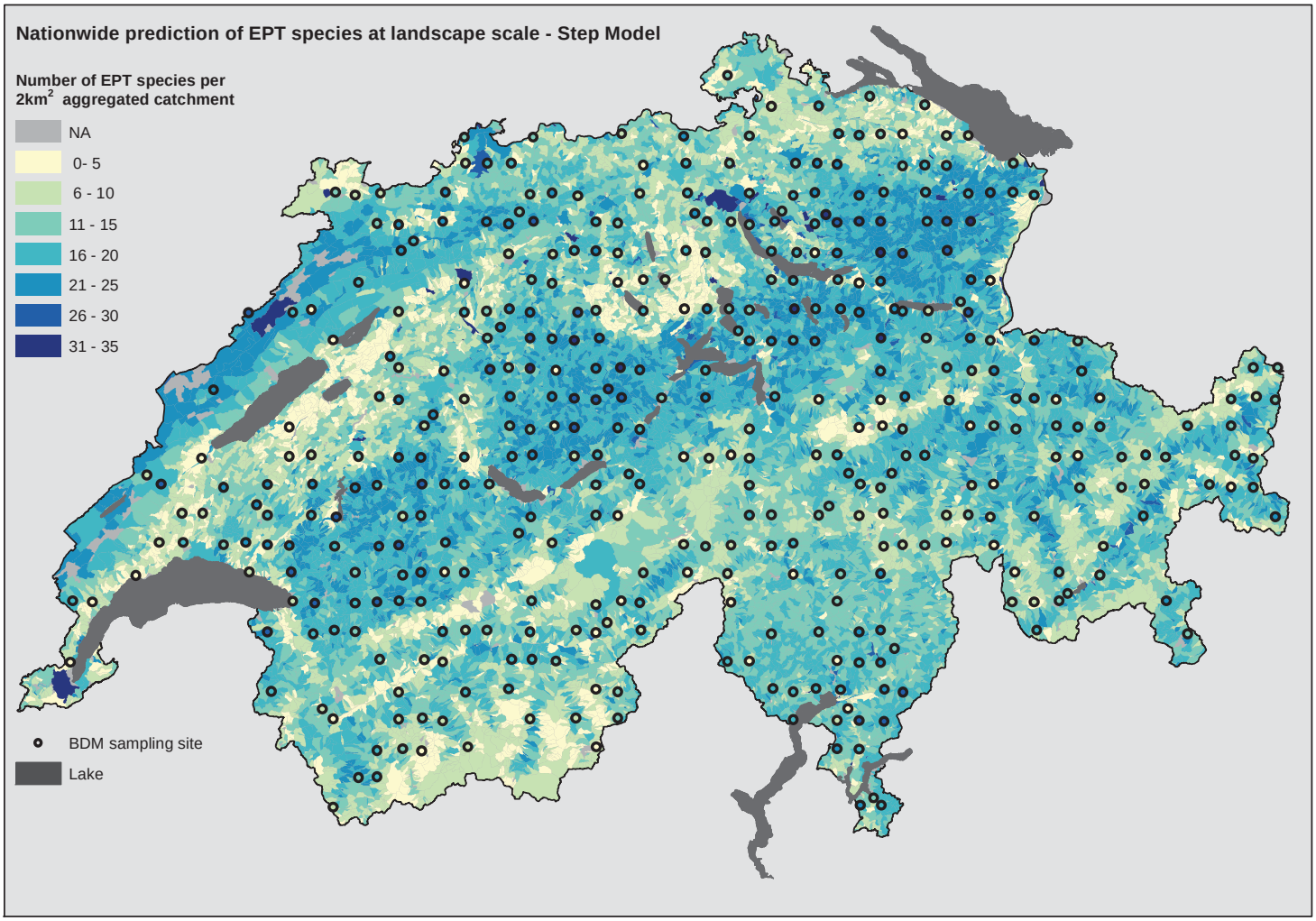
While the BDM sampling areas are defined with the WSB catchment dataset the nationwide prediction sampling areas are defined with the AC_2km^2 catchment dataset. To evaluate if the catchment dataset choice affects the macroinvertebrate richness prediction, the macroinvertebrate richness at the BDM sampling sites was predicted firstly with the WSB catchment dataset (*StepPrediction-BDMsites-WSB*, *LassoPrediction-BDMsites-WSB*) and secondly with the AC_2km^2 catchment dataset (*StepPrediction- BDMsites-AC_2km^2*, *LassoPrediction-BDMsites-AC_2km^2*) (Figure 19). The concordance correlation coefficients are higher for the *Lasso Models* (EPT: 0.64; IBCH: 0.92) than for the Step Models (EPT: 0.54; IBCH: 0.23). For the *Step Models* the concordance correlation coefficient is higher for the EPT species (0.54) than for the IBCH taxa (0.23). The opposite holds true for the *Lasso Models*: the concordance correlation coefficient is higher for the IBCH taxa (0.92) than for the EPT species (0.64). With exception of the *Lasso Models* for the IBCH taxa the model agreements are poor (McBride, 2005). The model agreement of the *Lasso Models* for the IBCH taxa is moderate.

Table 11: Anova output of the GLM for the EPT species

```
> anova(step_glm_2wayinteraction_gaussian, test="F")
Analysis of Deviance Table

Model: gaussian, link: identity

Response: BDM_a_EPT

Terms added sequentially (first to last)

                                          Df Deviance Resid. Df Resid. Dev       F    Pr(>F)
NULL                                                      409      22797
forest_percentage                          1  2196.19       408      20601 62.5559 2.802e-14 ***
green_percentage                           1  2211.54       407      18389 62.9930 2.316e-14 ***
decidious_per_forest                       1   378.87       406      18011 10.7916 0.0011136 **
street_percentage                          1   455.49       405      17555 12.9740 0.0003574 ***
slope_mean                                 1   116.42       404      17439  3.3159 0.0693909 .
Masl                                       1    27.33       403      17411  0.7784 0.3781745
carbonate_per_carbonatesilicate            1   250.31       402      17161  7.1297 0.0079047 **
watercourse_bdm_m                          1   189.48       401      16972  5.3972 0.0206917 *
roof_percentage                            1    93.31       400      16878  2.6579 0.1038620
wastewater_m3.a                            1    19.33       399      16859  0.5505 0.4585830
corn_percentage                            1   601.78       398      16257 17.1410 4.272e-05 ***
forest_percentage:green_percentage         1   579.37       397      15678 16.5028 5.895e-05 ***
forest_percentage:street_percentage        1    91.62       396      15586  2.6098 0.1070287
forest_percentage:Masl                     1    86.25       395      15500  2.4566 0.1178588
forest_percentage:wastewater_m3.a          1   394.54       394      15105 11.2381 0.0008812 ***
green_percentage:street_percentage         1     0.91       393      15104  0.0260 0.8719102
green_percentage:Masl                      1     6.80       392      15098  0.1938 0.6600269
green_percentage:corn_percentage           1     2.92       391      15095  0.0831 0.7733232
decidious_per_forest:street_percentage     1     4.97       390      15090  0.1416 0.7068711
decidious_per_forest:slope_mean            1   434.87       389      14655 12.3869 0.0004844 ***
street_percentage:corn_percentage          1   521.29       388      14134 14.8482 0.0001366 ***
slope_mean:Masl                            1   103.16       387      14030  2.9385 0.0872986 .
Masl:carbonate_per_carbonatesilicate       1   204.47       386      13826  5.8240 0.0162779 *
Masl:roof_percentage                       1    81.68       385      13744  2.3265 0.1280147
Masl:corn_percentage                       1   158.96       384      13585  4.5279 0.0339847 *
carbonate_per_carbonatesilicate:roof_percentage  1  139.05   383      13446  3.9605 0.0472893 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Table 12: Anova output of the GLM for the IBCH taxa

```
> anova(step_glm_2wayinteraction_gaussian, test="F")
Analysis of Deviance Table

Model: gaussian, link: identity

Response: BDM_a_IBCH

Terms added sequentially (first to last)

                                          Df Deviance Resid. Df Resid. Dev       F    Pr(>F)
NULL                                                        409     7929.4
area_bdm_m2                                1     3.53       408     7925.8  0.3267 0.5679621
facade_percentage                          1   488.04       407     7437.8 45.1888 6.553e-11 ***
vegetable_percentage                       1   173.33       406     7264.5 16.0490 7.424e-05 ***
disposalsite_2004_percentage               1    11.59       405     7252.9  1.0728 0.3009770
forest_percentage                          1   539.47       404     6713.4 49.9509 7.561e-12 ***
fruit_percentage                           1    59.43       403     6654.0  5.5032 0.0194939 *
green_percentage                           1   876.48       402     5777.5 81.1557 < 2.2e-16 ***
Masl                                       1   812.83       401     4964.7 75.2626 < 2.2e-16 ***
slope_mean                                 1    91.30       400     4873.4  8.4539 0.0038557 **
vine_percentage                            1    42.17       399     4831.2  3.9045 0.0488792 *
area_bdm_m2:disposalsite_2004_percentage   1     4.17       398     4827.0  0.3861 0.5347034
area_bdm_m2:slope_mean                     1    19.59       397     4807.4  1.8137 0.1788621
facade_percentage:vegetable_percentage     1    13.75       396     4793.7  1.2729 0.2599397
facade_percentage:disposalsite_2004_percentage  1  11.25     395     4782.4  1.0421 0.3079853
facade_percentage:vine_percentage          1     5.77       394     4776.7  0.5340 0.4653811
vegetable_percentage:forest_percentage     1     6.50       393     4770.2  0.6019 0.4383173
vegetable_percentage:green_percentage      1     2.22       392     4767.9  0.2054 0.6506309
vegetable_percentage:Masl                  1    41.97       391     4726.0  3.8865 0.0494004 *
disposalsite_2004_percentage:forest_percentage  1  57.71     390     4668.3  5.3438 0.0213288 *
disposalsite_2004_percentage:fruit_percentage   1 149.50     389     4518.8 13.8426 0.0002287 ***
disposalsite_2004_percentage:slope_mean    1    80.76       388     4438.0  7.4774 0.0065394 **
disposalsite_2004_percentage:vine_percentage    1  16.89     387     4421.1  1.5637 0.2118971
forest_percentage:fruit_percentage         1     8.06       386     4413.1  0.7461 0.3882450
forest_percentage:green_percentage         1    69.07       385     4344.0  6.3954 0.0118442 *
fruit_percentage:green_percentage          1    12.11       384     4331.9  1.1209 0.2903914
fruit_percentage:Masl                      1    74.26       383     4257.6  6.8758 0.0090873 **
green_percentage:vine_percentage           1    83.09       382     4174.5  7.6935 0.0058146 **
Masl:slope_mean                            1    59.76       381     4114.8  5.5333 0.0191664 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```



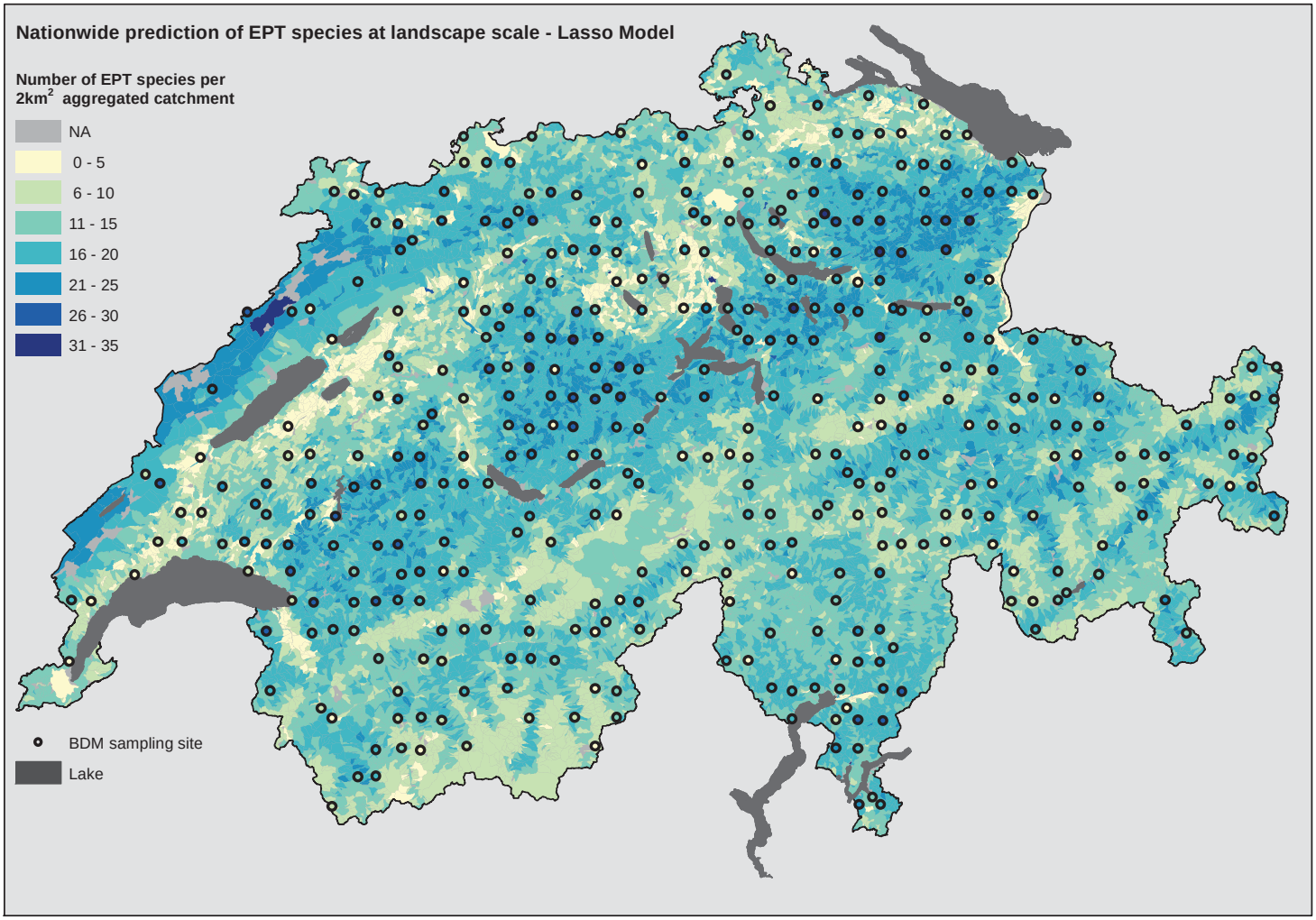
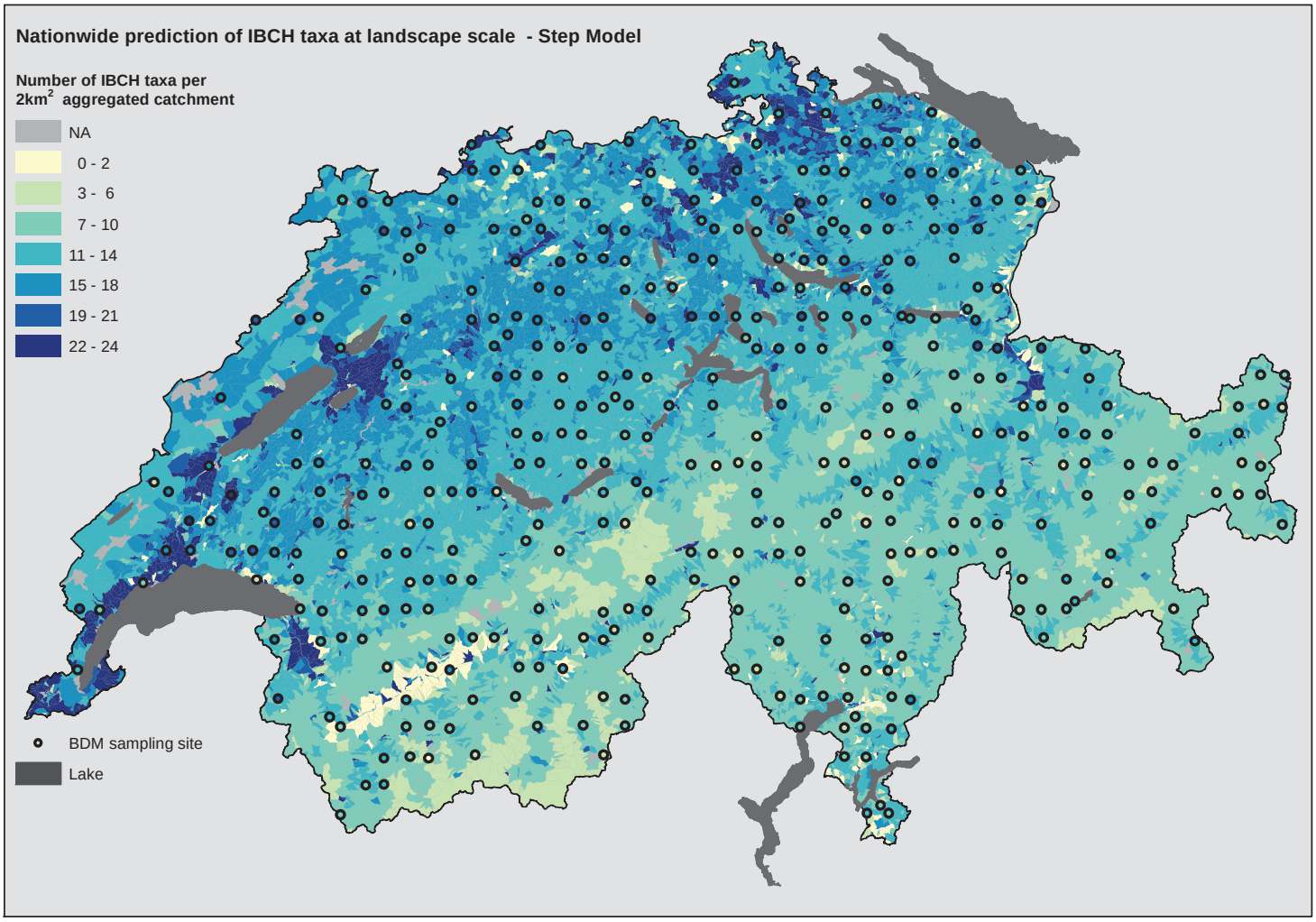


Figure 11: Nationwide EPT species prediction (bounded EPT species values: All predicted EPT species values that are smaller than the minimum value of the Monitoring are assigned to the minimum value of the Monitoring and all EPT species that are larger than the maximum value of the Monitoring are assigned to the maximum value of the Monitoring). *(Source: Bundesamt für Umwelt (n.d): Einzugsgebietsgliederung Schweiz EZGG-CH; Swisstopo (2007): Vector 25, Primärflächen; Koordinationsstelle BDM (2014))*



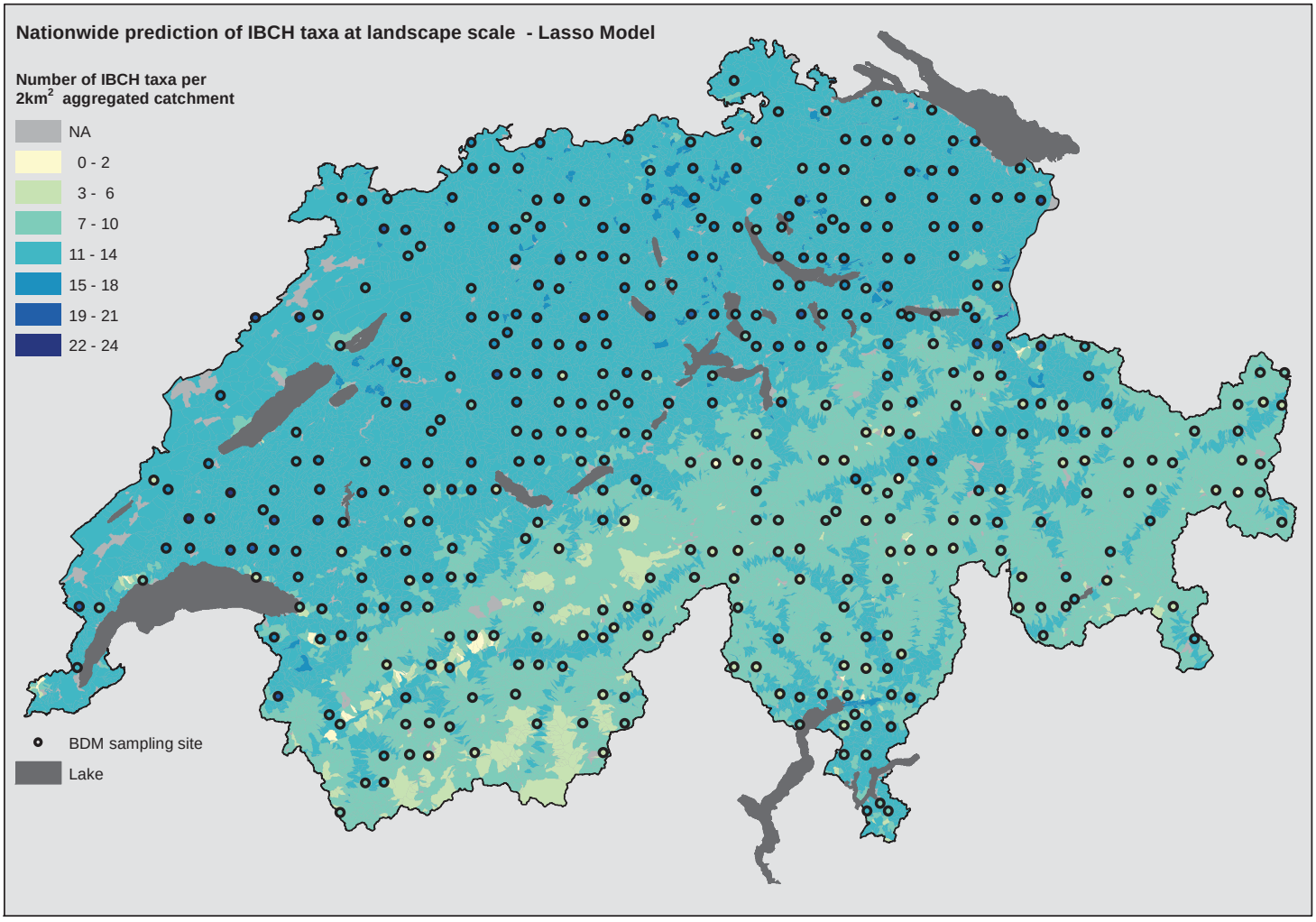
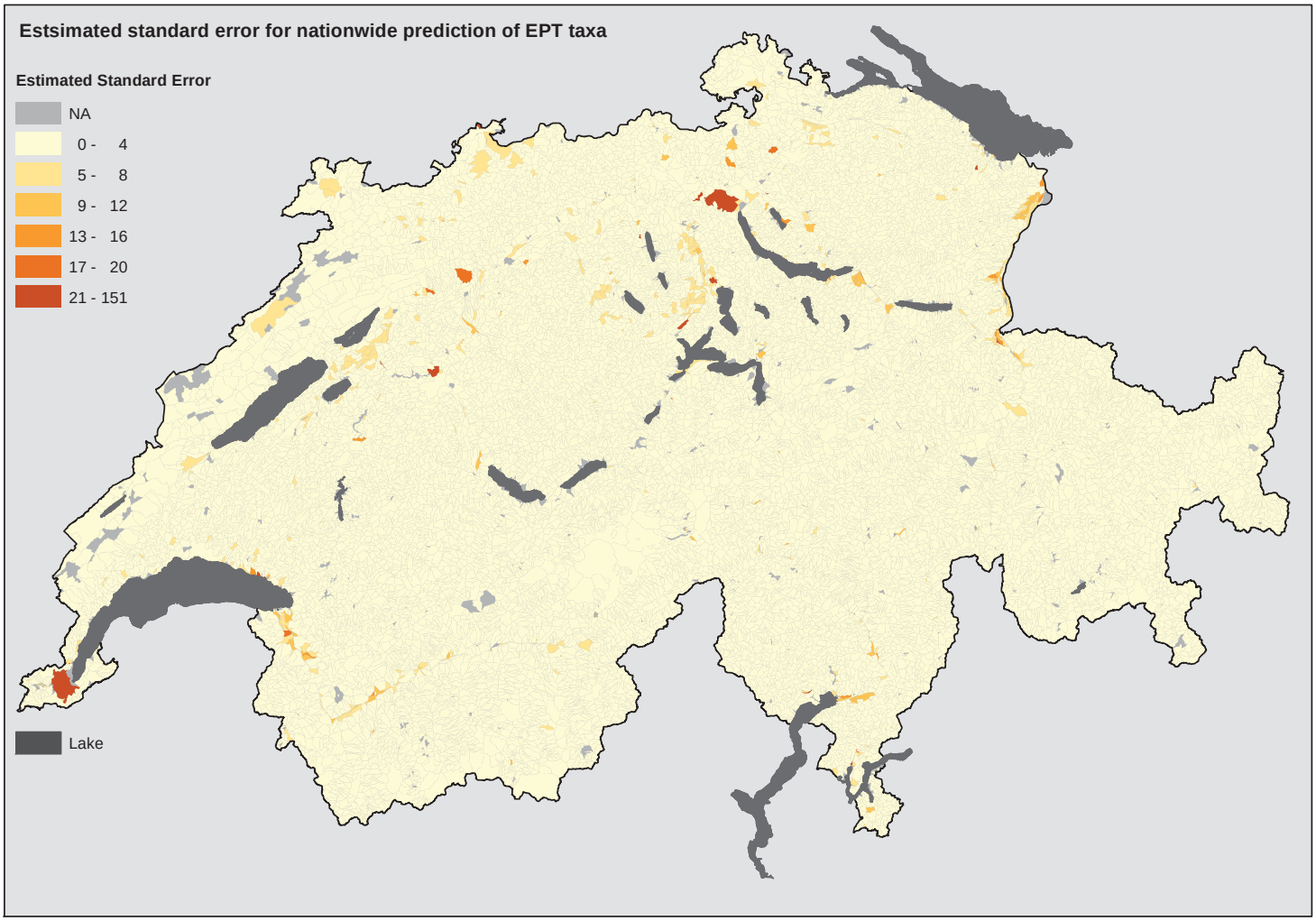


Figure 12: Nationwide IBCH taxa prediction (bounded IBCH taxa values: All predicted IBCH taxa values that are smaller than the minimum value of the Monitoring are assigned to the minimum value of the Monitoring and all IBCH taxa that are larger than the maximum value of the Monitoring are assigned to the maximum value of the Monitoring). (Source: Bundesamt für Umwelt (n.d): Einzugsgebietsgliederung Schweiz EZGG-CH; Swisstopo (2007): Vector 25, Primärflächen; Koordinationsstelle BDM (2014))



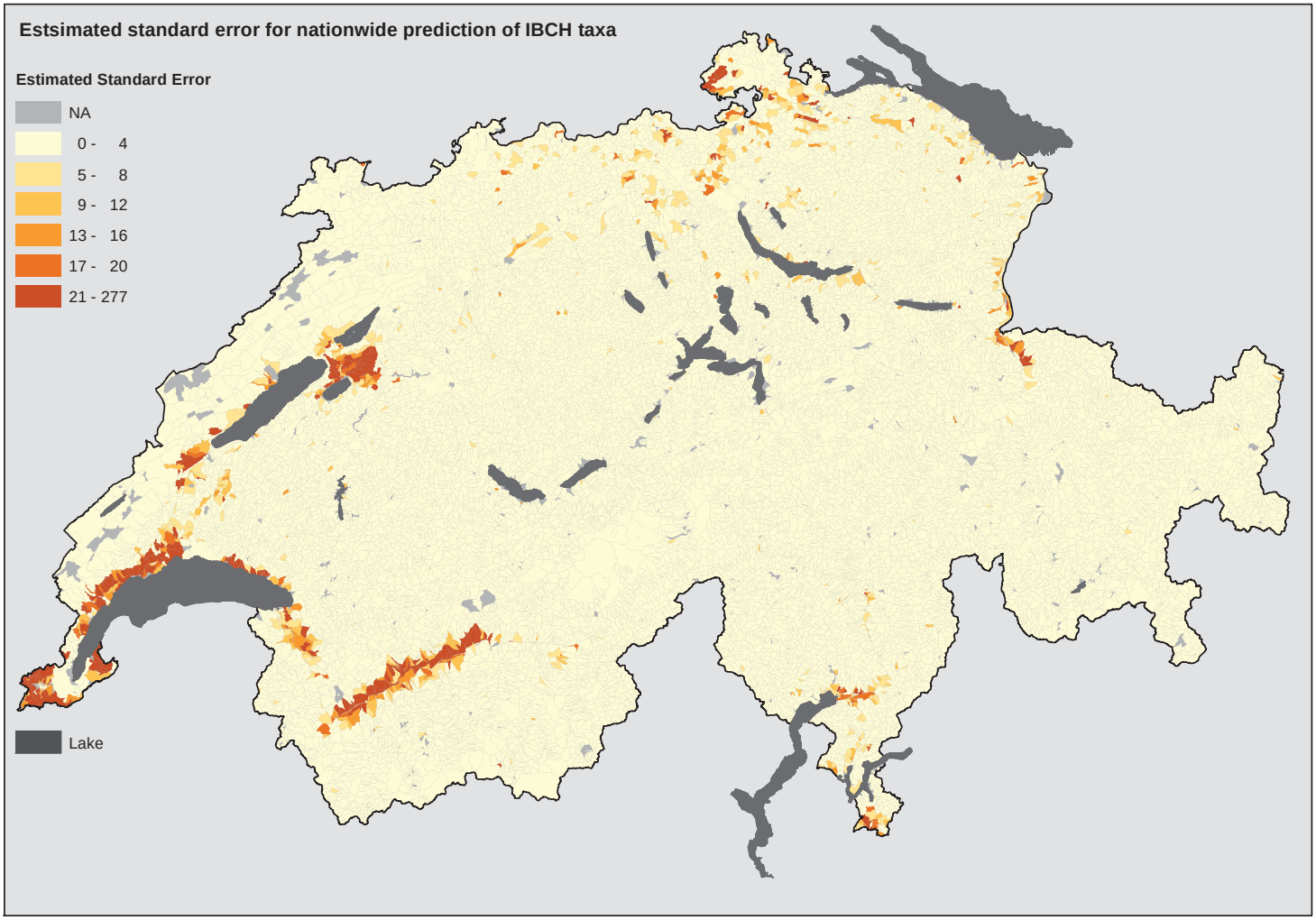


Figure 13: Estimated standard errors for the Step Model. *(Source: Bundesamt für Umwelt (n.d): Einzugsgebietsgliederung Schweiz EZGG-CH; Swisstopo (2007): Vector 25, Primärflächen)*

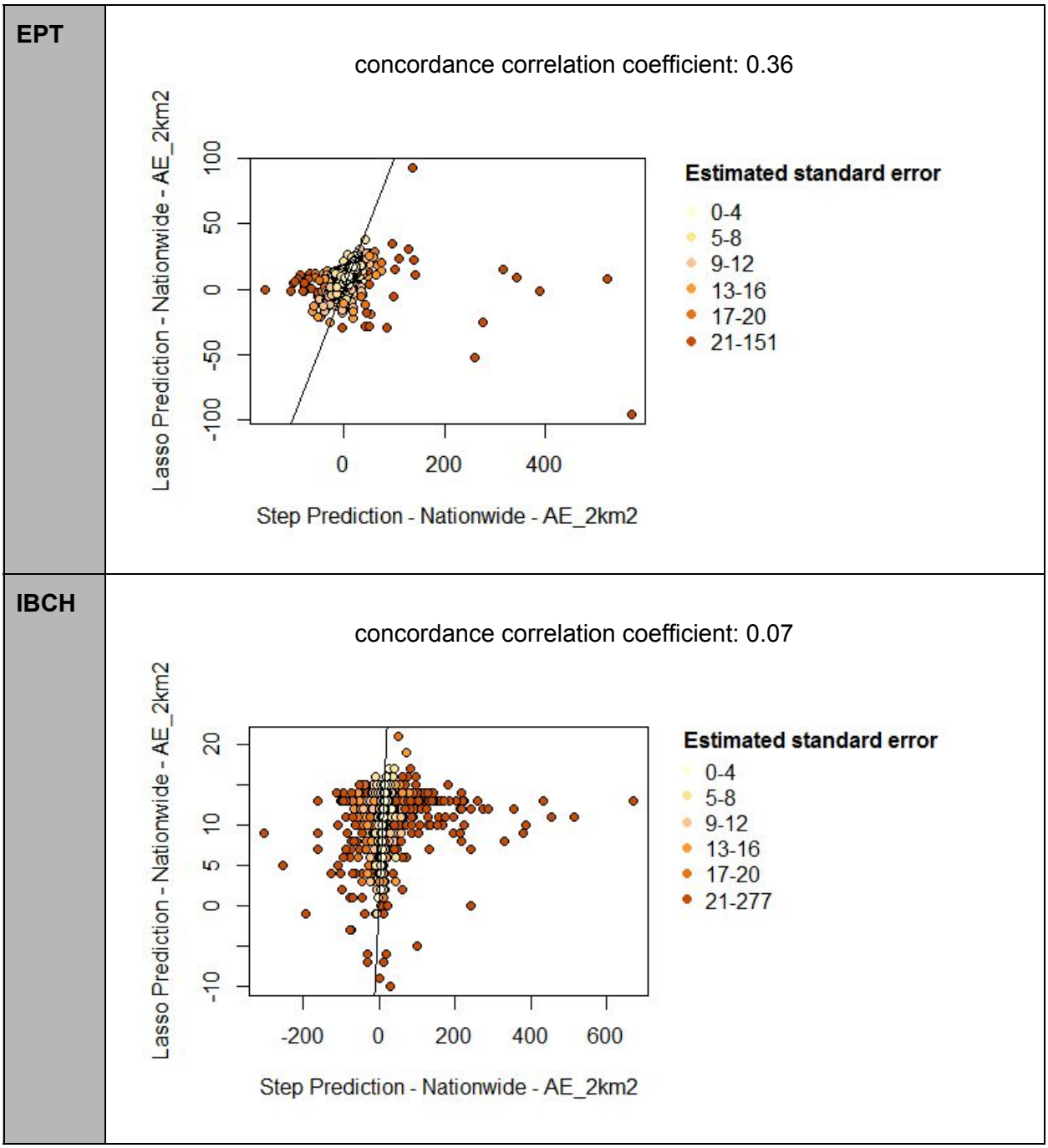


Figure 14: Comparison of the predicted macroinvertebrate richness values: *StepPrediction-Nationwide-AE_km²* vs.*LassoPrediction-Nationwide-AE_2km²* (boundless species and taxa richness values) and visualization of the estimated standard errors of the *StepPrediction-Nationwide- AE_2km2*

Table 13: EPT species richness histograms for monitoring and predictions (boundless EPT species richness values; values that appear very seldomly are cut off in the histograms).

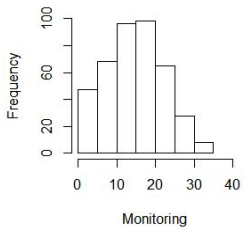
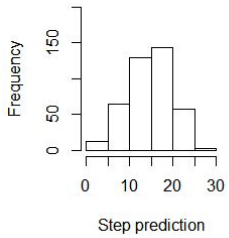
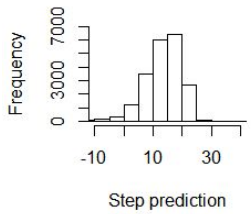
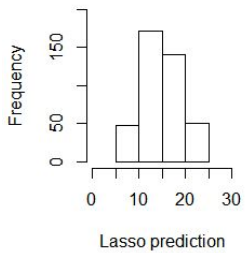
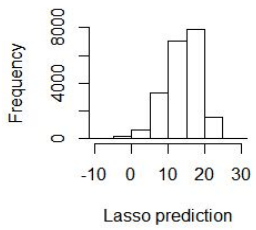
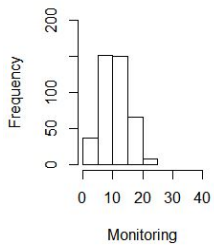
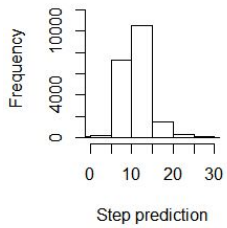
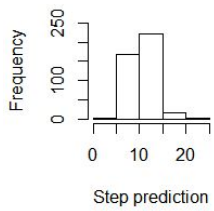
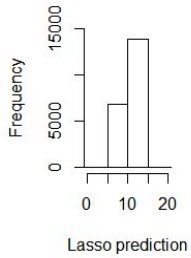
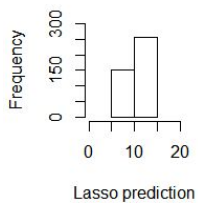
EPT	Histogramm	Boundary
Monitoring		Lower boundary: 0 Upper Boundary: 35 Mean: 15.2
Step Prediction - BDM sites - AE_2km²		Lower boundary:2 Upper Boundary: 26 Mean: 15.2
Step Prediction - Nationwide - AE_2km²		Lower boundary: -154 Upper Boundary: 569 Mean: 13.9
Lasso Prediction - BDM sites - AE_2km²		Lower boundary: 5 Upper Boundary: 25 Mean: 15.2
Lasso Prediction - Nationwide - AE_2km²		Lower boundary: -96 Upper Boundary: 93 Mean: 14.3

Table 14: IBCH taxa richness histograms for monitoring and predictions (boundless IBCH taxa richness values; values that appear very seldomly are cut off in the histograms).

IBCH	Histogramm	Boundary
Monitoring		Lower boundary: 0 Upper Boundary: 24 Mean:11.2
Step Prediction - BDM sites - AE_2km²		Lower boundary: 4 Upper Boundary: 21 Mean: 11.2
Step Prediction - Nationwide - AE_2km²		Lower boundary: -301 Upper Boundary: 671 Mean: 12.6
Lasso Prediction - BDM sites - AE_2km²		Lower boundary: 6 Upper Boundary: 15 Mean: 11.2
Lasso Prediction - Nationwide - AE_2km²		Lower boundary: -10 Upper Boundary: 21 Mean: 11.5

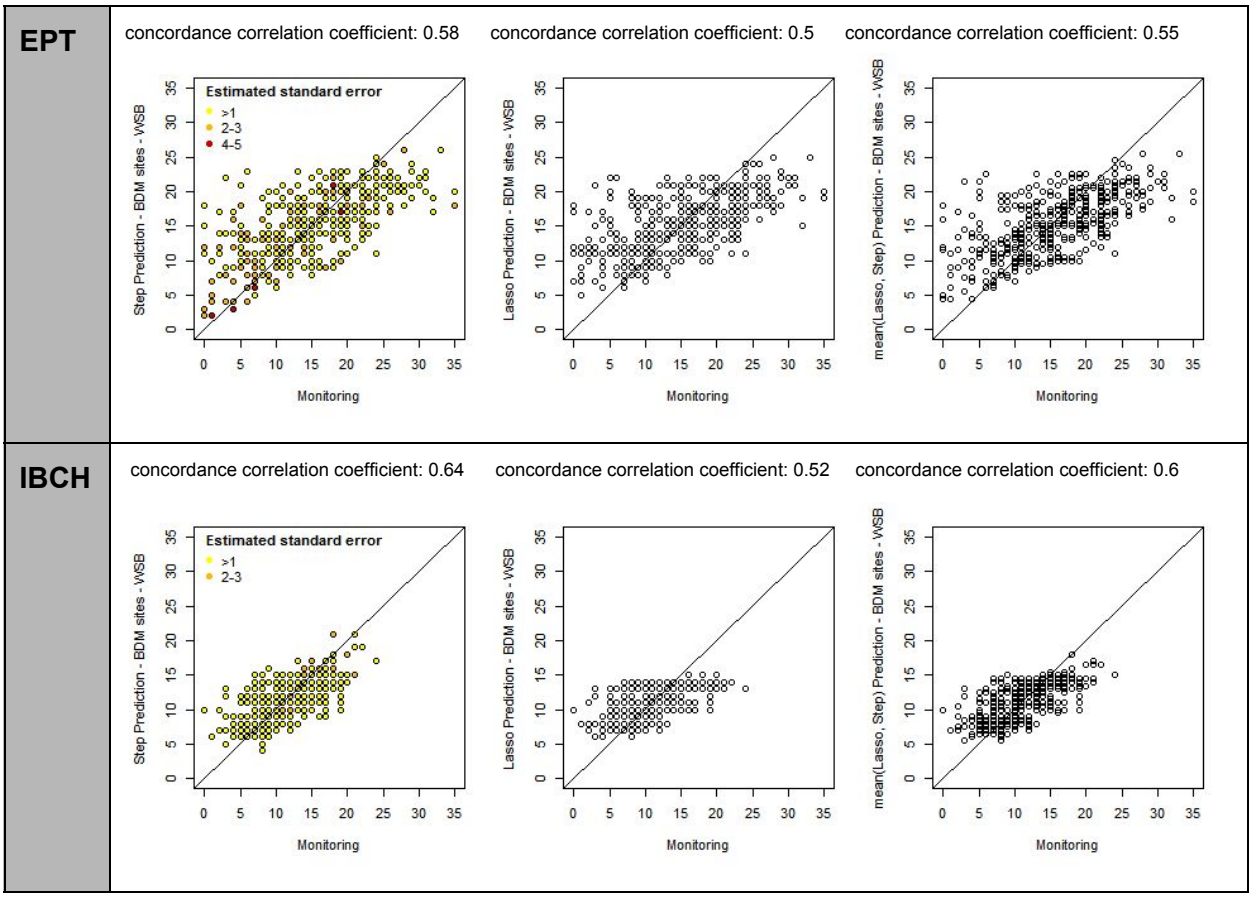


Figure 15: Comparison of the predicted macroinvertebrate richness values with the observed macroinvertebrate richness values: *StepPrediction-BDMsites-WSB* vs. *Monitoring*, *LassoPrediction-BDMsites-WSB* vs. *Monitoring* and *Mean(Lasso,Step)Prediction-BDMsites-WSB* vs. *Monitoring* (boundless species and taxa richness values). For the *StepPrediction-BDMsites-WSB* prediction the estimated standard errors are visualized.

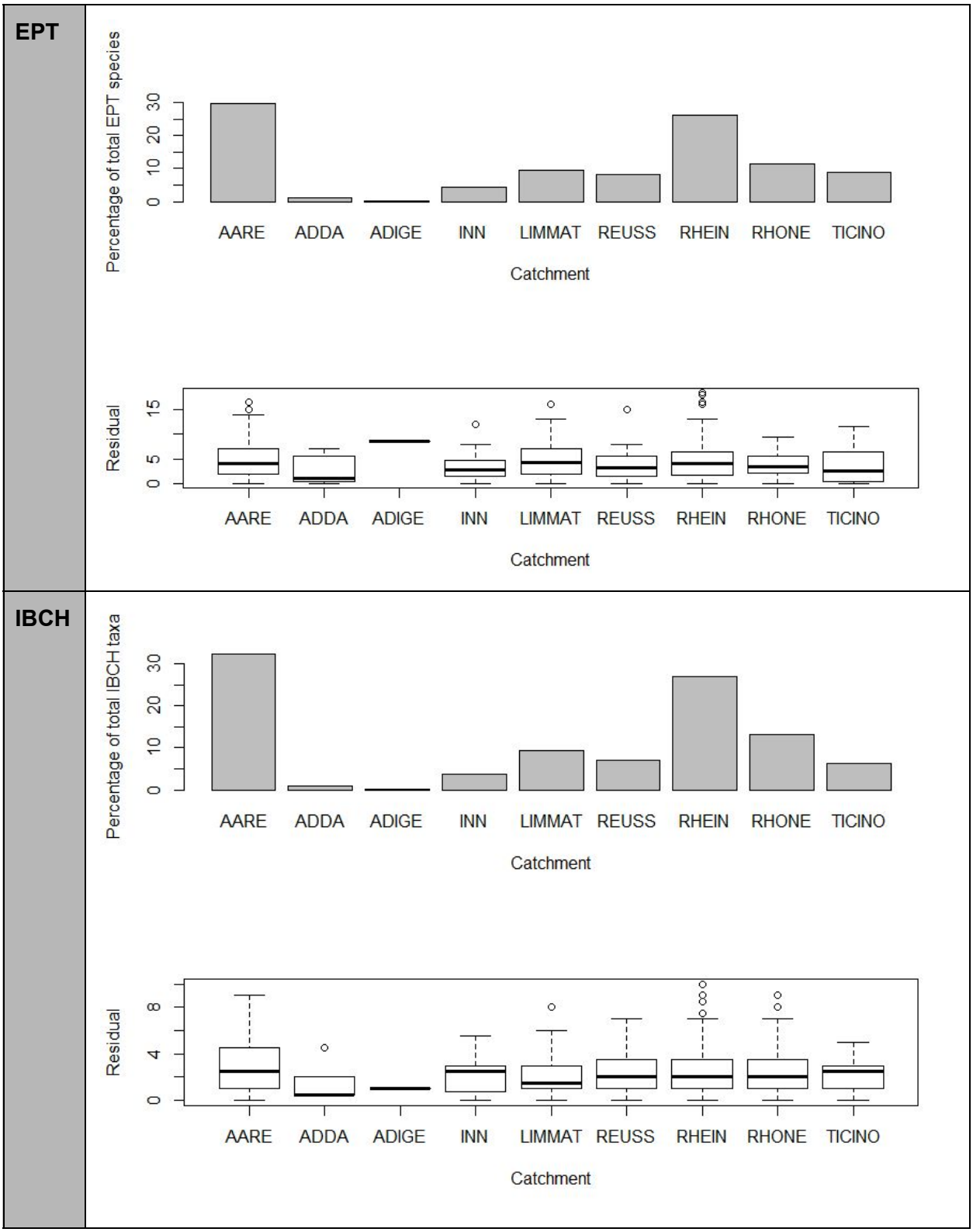


Figure 16: Percentage of the observed macroinvertebrate richness values (*Monitoring*) per catchment and occurrence of residuals (difference between *Mean(Lasso,Step) Prediction-BDMsites-WSB* and *Monitoring*) per catchment (boundless values).

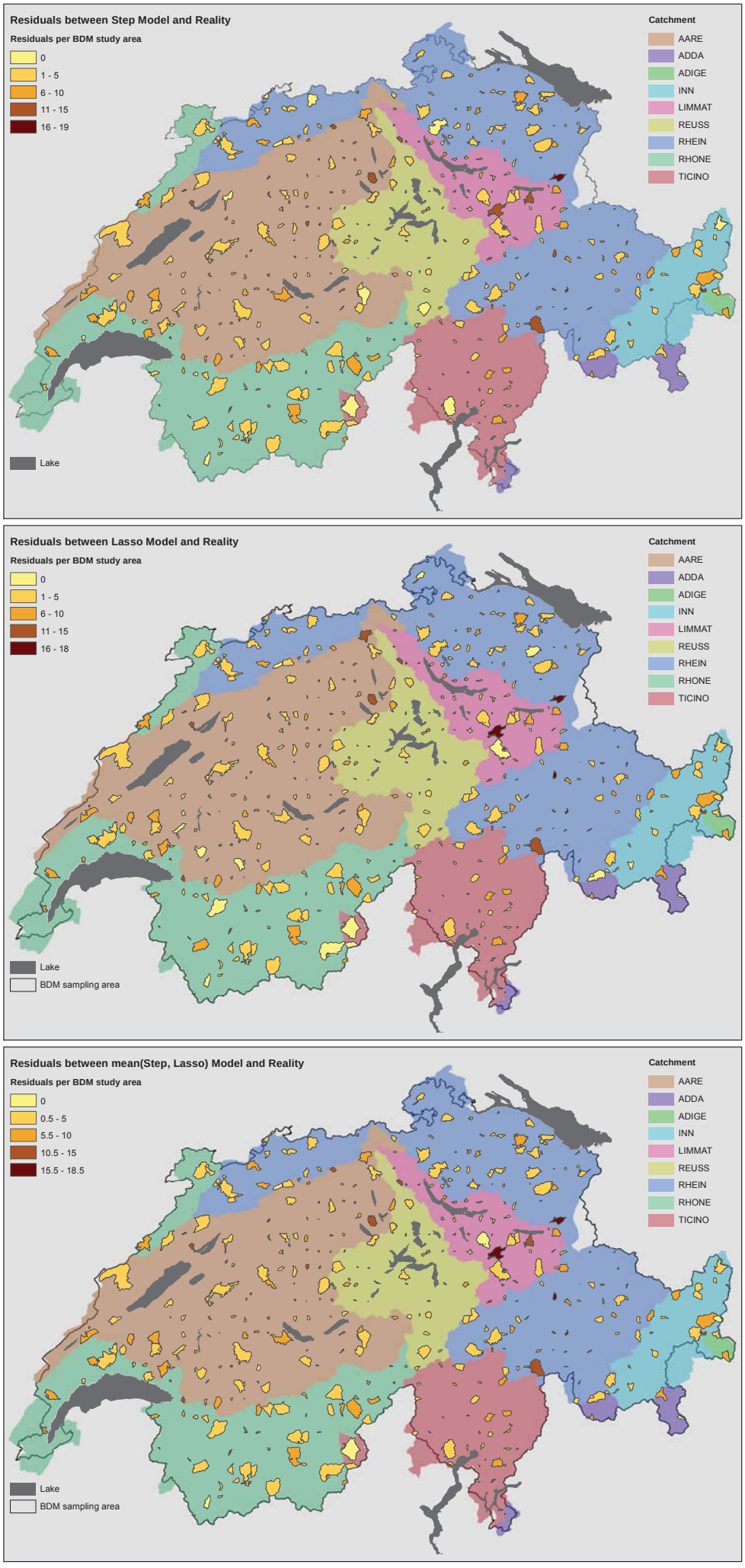
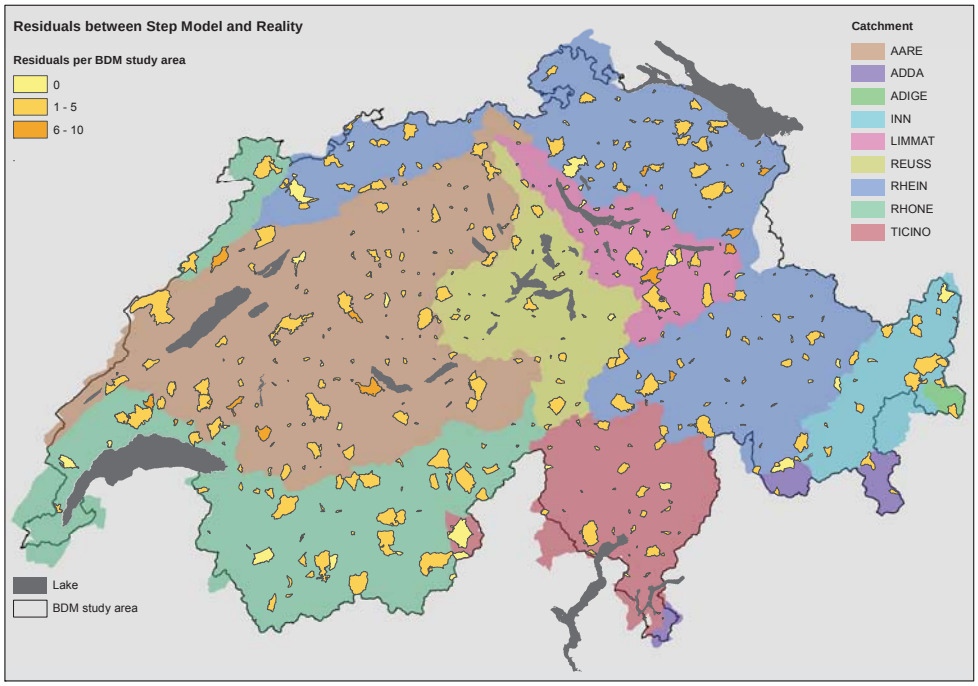
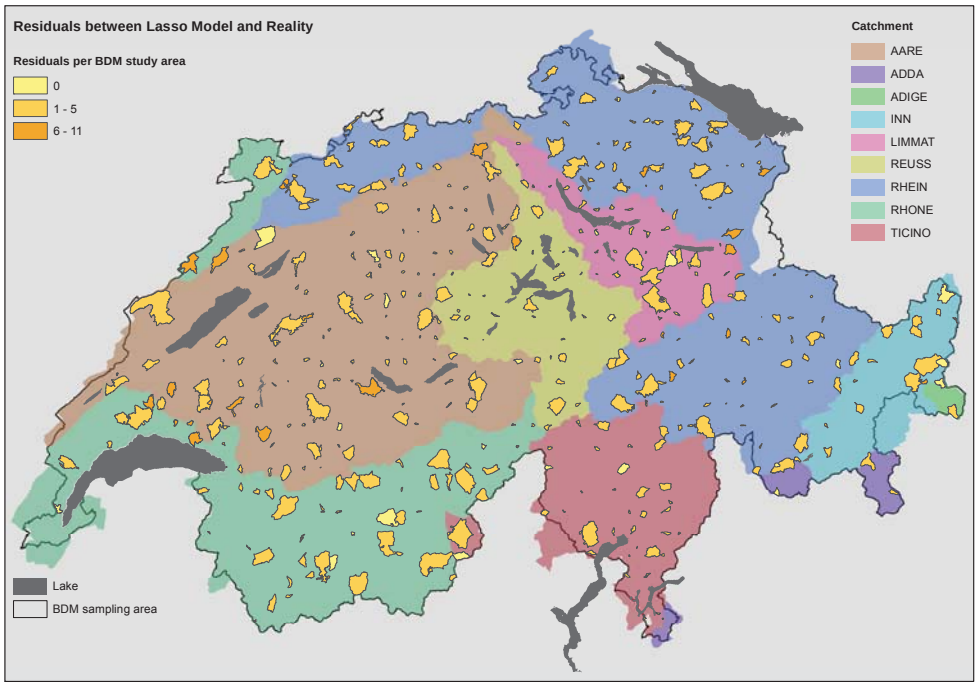


Figure 17: Residuals between the Mean(Lasso,Step)Prediction-BDMsites-WSB and the Monitoring values of the EPT species (boundless EPT species values). *(Source: Bundesamt für Umwelt (n.d.): Gewässerabschnittsbasierte Einzugsgebietsgliederung der Schweiz GAB- EZGG-CH; Bundesamt für Umwelt (n.d.): Hydrografische Gliederung – nachbearbeitete Version (basis04); Swisstopo (2007): Vector 25, Primärflächen)*





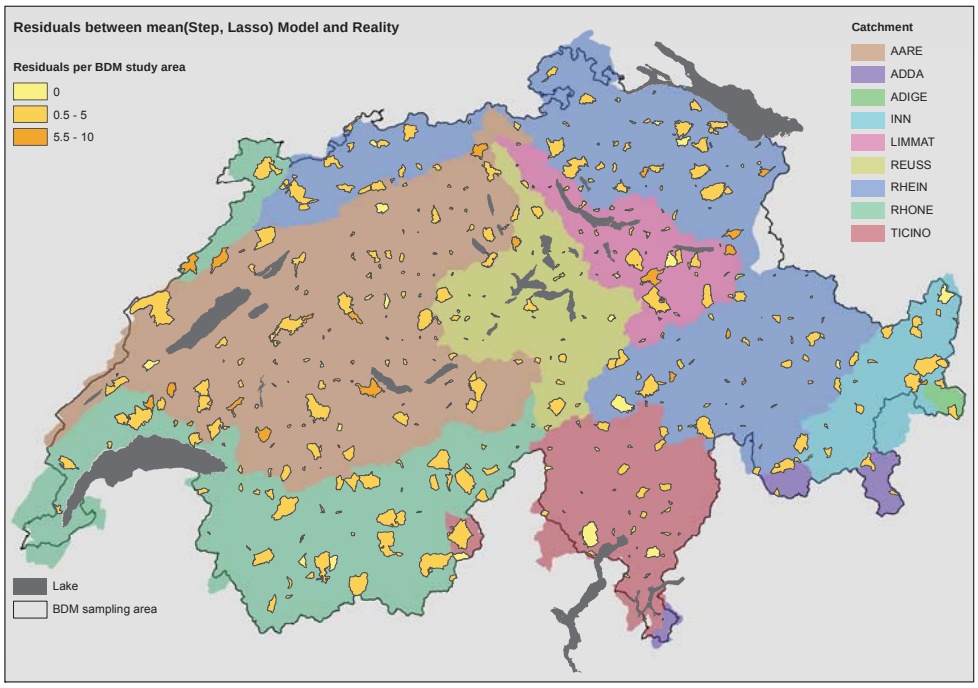


Figure 18: Residuals between the *Mean(Lasso,Step)Prediction-BDMsites-WSB* and the *Monitoring* values of the IBCH taxa (boundless IBCH taxa values). *(Source: Bundesamt für Umwelt (n.d.): Gewässerabschnittsbasierte Einzugsgebietsgliederung der Schweiz GAB- EZGG-CH; Bundesamt für Umwelt (n.d.): Hydrografische Gliederung – nachbearbeitete Version (basis04); Swisstopo (2007): Vector 25, Primärflächen)*

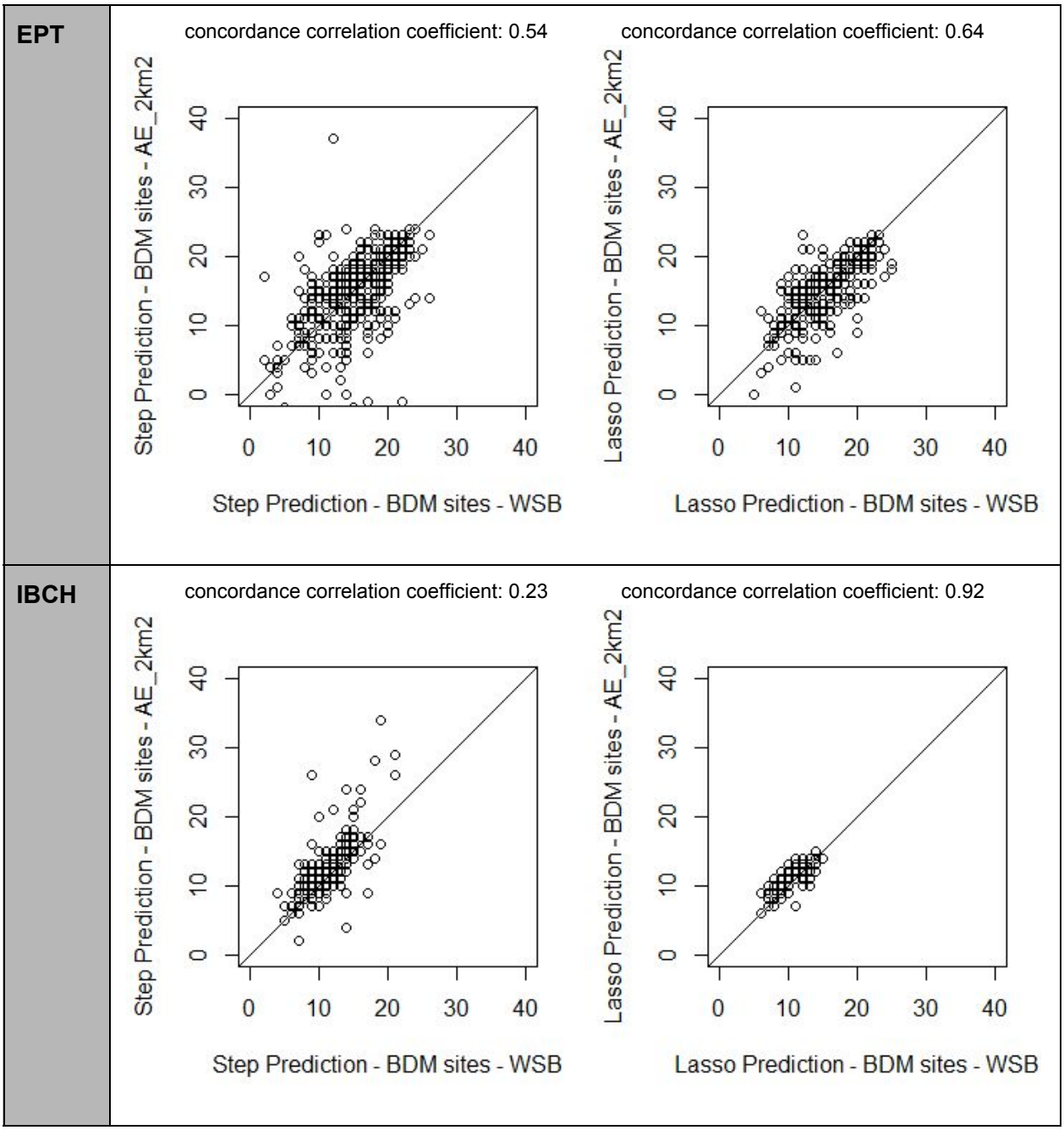


Figure 19: Comparison of the predicted macroinvertebrate richness values using different sampling areas: *StepPrediction-BDMsites-WSB* vs. *StepPrediction-BDMsites-AE_2km* and^2 *LassoPrediction-BDMsites- AE_2km*^{2} vs. *LassoPrediction-BDMsites-AE_2km*^{2} (boundless species and taxa richness values).

4. Discussion

4.1 Nationwide macroinvertebrate prediction and important environmental variables

Past studies that related macroinvertebrate richness distribution to land-use variables found that macroinvertebrate richness is highest in woods, followed by grassland/pasture, agricultural and developed areas (Wahl et al. 2013, Egler et al. 2012, Roy et al. 2003, Moore & Palmer 2005).

4.1.1 EPT species

In agreement with these findings, this study predicts that the sites with highest EPT species richness are found in forested and pasture covered areas (Figure 11). In disagreement with these findings, however, this study did not find the lowest EPT species richness at populated sites but in cultivated areas. This suggests that pollutants from cultivated land affects the EPT species richness in Switzerland more than urban pollution. The anova table for the EPT species (Table 11) confirms this observation: arable land (*corn_ percentage*) reduces more residual deviance of the recorded EPT species richness during the BDM than urban sites (*roof_percentage*). Moreover, a study that analyzed the presence of pesticides (plant protection products and biozide) in five medium sized watercourses of Switzerland found that a significant proportion of pesticides can be attributed to plant protection products used in agriculture (Wittmer et. al 2014). This indicates that agriculture impacts water quality more than populated areas and coincides with the findings described above.

The low EPT species richness predictions at glaciated and fin covered areas (Figure 11) are explainable by the fact that glaciated and fin covered areas are too cold for their survival. It has been shown that EPT species are found only a few hundred meters downstream the glacier (Lords-Crozet et al. 2001). Glaciated area is not explicitly considered as an explanatory variable in this study but other explanatory variables implicitly relate to it (eg. *Masl*).

The numerous EPT species richness prediction at livestock farming areas (Figure 11) does not agree with past studies. According to Kyriakeas & Watzin (2006) more EPT species are found near cultivated land (cornfields) than near livestock area (cows). McIver & McInnis (2007), who compared EPT richness in grazing and non-grazing areas, also found that the EPT richness is higher in non-grazing units.

In summary, the results of the EPT species suggest that land-use variables reduce the residual deviance of the recorded EPT species richness during the BDM significantly and do a got job at predicting the EPT species. While land-use variables significantly reduce the residual deviance of the EPT species richness recorded during the BDM on their own, topological variables (*Masl*, *slope_mean*) only significantly reduce the deviance when interactions are considered (Table 11). This suggests that the EPT species-topological variable relationship is not linear but convex quadratic. The fact that c*arbonate_per_carbonatesilicate* and *decidious_ per_forest* significantly reduce the residual deviance of the EPT species richness recorded during the BDM (Table 11) suggests that EPT species are not only influenced by land-use and topological variables but also by other factors.

4.1.2 IBCH taxa

In contrast to the nationwide EPT species richness prediction the nationwide IBCH taxa richness prediction does not follow the land-use ranking found in literature and is mainly determined by the topological variable *Masl* (Figure 12). Nevertheless, the anova table (Table 12) indicates that land-use variables significantly reduce the residual deviance of the recorded IBCH taxa richness during the BDM. As the IBCH taxa consists of numerous orders that are characterized by a variety of different ecological niches it is likely that orders that are sensitive towards land-use variables are blurred by less sensitive orders. This might also explain why developed area land-use variables reduce more residual deviance of the recorded IBCH species richness during the BDM than cultivated land-use variables. The fact, that different cultivated-land (EPT: *corn_percentage*; IBCH: *vegetable_ percentage*) and developed area variables (EPT: *street_percentage*; IBCH: *facade_percentage*) explain the largest amount of deviance in the response variable according to tree models for the EPT species and IBCH taxa (Appendix 9 and 10) reinforces the assumption that different macroinvertebrate orders are characterized by distinct sensitivities and habitat preferences.

4.2 Comparison of the model selection methods

The better agreement between the nationwide *StepPrediction-Nationwide-AC_2km^2 and LassoPrediction-Nationwide-AC_2km^2* (Figure 14) for the EPT species than for the IBCH taxa is an additional indication that the environmental niche is better defined for the EPT species than for the IBCH taxa and thus easier to model and predict. The result that the *StepPrediction-BDMsites-WSB* and *LassoPrediction-BDMsites-WSB* predictions agree slightly better for the IBCH taxa *Monitoring* than for the EPT species *Monitoring* (Figure 15), however, would suggest that the ecological niche of the IBCH taxa is better modelled. Nonetheless the poor agreement between the *StepPrediction-Nationwide-AC_2km^2 and LassoPrediction-Nationwide-AC_2km^2* for the IBCH taxa suggests that the better fit of the *StepPrediction-BDMsites-WSB* and *LassoPrediction-BDMsites-WSB* with the IBCH taxa *Monitoring* is misleading and likely caused by overfitting.

The comparison of the *StepPrediction-BDMsites-WSB* and the *LassoPrediction-BDMsites-WSB* with the *Monitoring* values suggests that the *Step Model* does a better job than the *Lasso Model* (Figure 15). Figure 19, however, indicates that the *Lasso Model* might be stabler since the *Lasso Models* that are carried out for different sampling areas agree better than the *Step Models* that are carried out for different sampling areas.

The observation that areas with high estimated standard errors of the *StepPrediction-Nationwide-AC_2km^2* (Figure 13) are primarily located at lake inflows and outflow might be explainable by dispersion of lake macroinvertebrate taxa into adjacent watercourses. Lake macroinvertebrate taxa are characterized by different ecological niche preferences than watercourse taxa and are thus not modelled well.

4.3 Evaluation of the prediction at the BDM sampling areas

The observed model over- and underprediction of the *StepPrediction-BDMsites-WSB* and *LassoPrediction-BDMsites-WSB* prediction (Figure 15) indicates that explanatory variables which are not considered in the model influence the macroinvertebrates. An explanation for this observation might be that while the BDM data represent the realized niche, the models only consider the fundamental niche and neglect interspecific competition. This is especially plausible in view of the fact that population size proves to be an effective antipredator defense (Wrona & Dixon 1990). Other possible explanations for the model over- and underprediction might be the omission of population growth consideration (Snider & Brimlow 2013) and the omission of spatial pattern consideration (dendritic networks; Altermatt 2013).

The presence of relatively high residuals of the *Mean(Lasso,Step)Prediction-BDMsites-WSB* and the *Monitoring* (Figure 16) at sites where few monitorings took place might be explainable by the presence of macroinvertebrates that are endemic to these catchments.

4.4 Evaluation of environmental variables choice

As only environmental variables for which nationwide spatial data are available are considered in this study, the results of the macroinvertebrates predictions could be improved through the availability of more nationwide spatial data. If, for example, discharge is included as an explanatory variable, it is selected by the tree model that is carried out for the EPT species. This indicates that it may be an important EPT species richness predictor. It would also be helpful to have more nationwide spatial data available that coincide with the environmental variables that are recorded during the BDM (chiefly instream habitat features, eg. riverbed substrate, flow rate). Nevertheless there is also uncertainty regarding the selection of explanatory variable for which nationwide available data are available. It is likely that not all nationwide available data that have an effect on the macroinvertebrate are recognized by this study. Livestock, for example could not be included, because of unawareness of the existence of this dataset until late in the study.

4.5 Evaluation of habitat distribution model choice

Regression methods are frequently applied in taxa habitat distribution models because of their statistical foundation and their ability to realistically model ecological relationships. Yet, numerous other approaches exist to model taxa distributions. The most comprehensive set of model comparison up to date suggests that novel methods (eg. machine learning techniques) consistently outperform more established methods (eg. regression analysis) (Elith et al. 2006). However, since the models in this study are built with conventional procedures and occurrence-only data, it is difficult to interpret the above finding. Established regression methods with more sophisticated procedures might yield similar results as novel methods that are carried out with conventional procedures. Moreover another study (Guisan et al. 2007) found that while there are important differences between the model performances, the variance in model performance is greater among species than among techniques. Species that grow slowly and/or have a specialized niche tend to be better modeled independent of the chosen method. This indicates that suitability knowledge of species is important for modeling purposes.

4.6 Uncertainties

Uncertainties are not only present during the environmental variable selection and model choice, but also during all other modelling steps. A good prediction depends on reliable input data (BDM, explanatory variable, catchment and watercourse data), the choice of the catchment dataset, the definition of the sampling area and the modelling decisions (choice of GLM family and model selection method). Even though these uncertainties are very hard to quantify, it is important to be aware of them.

4.7 Conclusion

Understanding of the relationship between environmental variables and macroinvertebrates is an important milestone in understanding ecosystem processes that affect watercourse health. This study shows that although land-use (forest, pasture, cultivated land and developed area) and topology (elevation and slope) variables influence the Swiss macroinvertebrate richness distribution at a landscape level, they do not influence all macroinvertebrate taxa equally. The EPT species, which in contrast to the IBCH taxa are characterized by a clearly defined niche, react more sensitively towards land-use changes than the IBCH taxa. This results in opposing spatial predictions on richness at the landscape scale and suggests that the diversity pattern of one macroinvertebrate group (eg. EPT species) cannot be used as a proxy of another macroinvertebrate group (eg. IBCH taxa). It furthermore indicates that a better understanding of the the relationship between environmental variables and macroinvertebrate richness is gained when the focus is placed on a few sensitive macroinvertebrate species with a clearly defined ecological niche (eg. EPT species), than when a broad mixture of macroinvertebrate taxa with varying sensitivity and less clearly defined ecological niche (eg. IBCH taxa) are considered.

4.8 Outlook

Future studies could focus on modelling the species of the orders Ephemeroptera, Plecoptera and Trichoptera individually to test their sensitivity towards land-use variables separately. Moreover, it would also be interesting to test if and how the inclusion of instream habitat features and the dispersal of organisms changes the macroinvertebrate richness prediction.

5. Lessons learned

Throughout this study I learned a lot of lessons. In the following section I want to share the most important ones.

- Data management is very important. It should always be very clear how and where the data is stored and finding the most recent file should be evident. Backups are crucial.

- Scripts are very powerful as they automate repetitive steps and ensure reproducibility. Writing them can be time-intensive but once they work they save a lot of time since most steps have to be carried out several times.

- The processing time of the scripts can be an important issue and should be considered.

- Manual checks are mandatory. Never trust a result without checking it´s plausibility.

- Be careful to avoid copy-paste mistakes and joining files by the wrong attribute.

- In the future revision control should be used for scripts and for data.

6. Acknowledgement

I would like to thank Florian Altermatt, Rosi Sieber and Mat Seymour for all their advice. Moreover, I would like to thank the Eawag ECO team and specifically the Altermatt lab for being such a supportive, interested and cheerful crowd. Thanks also to Jan Seibert who gave me permission to carry out my master thesis at Eawag.

7. Personal declaration

I hereby declare that the submitted thesis is the result of my own, independent work. All external sources are explicitly acknowledged in the thesis.

_____

8. References

8.1 Literature

Aguiar, F. C., Ferreira, M. T. & Pinto, P. (2002): Relative influence of environmental variables on macroinvertebrate assemblages from an Iberian basin. Journal of the North American Benthological Society, Vol. 21, No. 1, p. 43-53.

Altermatt, F. (2013): Diversity in riverine metacommunities: A network perspective. Aquatic Ecology, Vol 47, No. 3, p. 365-375.

Altermatt, F., Seymour, M. & Martinez, N. (2013): River network properties shape α-diversity and community similarity patterns of aquatic insect communities across major drainage basins. Journal of Biogeography, Vol. 40, No. 3, p. 2249–2260.

Carrara, F., Altermatt, F., Rodriguez-Iturbe, I. & Rinaldo, A. (2012): Dendritic connectivity controls biodiversity patterns in experimental metacommunities. Proceedings of the National Academy of Sciences, Vol. 109, No. 15, p. 5761–5766.

Chapin III, F. S., Zavaleta E. S., Eviner V. T. , Naylor, R. L., Vitousek, P. M., Reynolds, H. L., Hooper D. U. Lavorel. S., Sala, O. E., Hobbie, S. E., Mack, M. C &, Diaz, S. (2000): Consequences of changing biodiversity. Nature, Vol 405, p. 234-242.

Crawley, M. J. (2007): The R book. England: John Wiley & Sons Ltd.

Death, R. G. & Winterbourn, M. J. (1995): Diversity patterns in stream benthic invertebrate communities: the influence of habitat stability. Ecology, Vol. 76, p. 1446-1460.

Delong, M. D. & Brusven, M. A. (1998): Macroinvertebrate community structure along the longitudinal gradient of an agriculturally impacted stream. Environmental management, Vol. 22, No. 3, p. 445-457.

Dudgeon, D., Arthington, A. H., Gessner, M. O., Kawabata, Z.-I., Knowler, D. J., Lévêque, C., Naiman, R. J., Prieur-Richard A.- H., Soto D., Stiassny, M. L. J. & Sullivan, C. A. (2006). Freshwater biodiversity: importance, threats, status and conservation challenges. Biological Reviews, Vol. 81, No.2, p. 163–182.

Egler, M., Buss, D. F., Moreira, J. C. & Baptista, D. F. (2012): Influence of agricultural land-use and pesticides on benthic macroinvertebrate assemblages in an agricultural river basin in southeast Brazil. Brazilian Journal of Biology, Vol. 72, No. 3, p. 437–443.

EPA (2012): Chapter 4: Macroinvertebrates and habitat. http://water.epa.gov/type/rsl/monitoring/vms40.cfm, (accessed: 2.8.15).

ESRI (2014): An overview of the Proximity toolset. http://resources.arcgis.com/en/help/main/10.2/index.html#//000800000018000000, (accessed: 2.8.15).

Gaston, K. J. (2000): Global patterns in biodiversity. Nature, Vol. 405, p. 220–227.

Goeman, J., Meijer, R. & Chaturvedi, N. (2014): L1 and L2 penalized regression models. p. 1–20.

Guisan, A. & Thuiller, W. (2005): Predicting species distribution: Offering more than simple habitat models. Ecology Letters, Vol. 8, p. 993–1009.

Guisan, A. & Zimmermann, N. E. (2000): Predictive habitat distribution models in ecology. Ecological Modelling, Vol. 135, p. 147–186.

Grubaugh, J. W., Wallace, J. B. & Houston, E. S. (1996): Longitudinal changes of macroinvertebrate communities along an Appalachian stream continuum. Canadian Journal of Fisheries and Aquatic Sciences, Vol. 53, No. 4, p. 896-909.

Heino, J., Melo, A. S., Bini, L. M., Altermatt, F., Al-Shami, S. A., Angeler, D. G., Bonada, N., Brand, C., Callisto, M., Cottenie, K. Dangles, O., Dudgeon, D. Encalada, A. Göthe, E., Grönroos, M., Hamada, N. Jacobsen, D. Landeiro, V. L., Ligeiro, R., Martins, R. T., Miserandino, M. L., Rawi, C. S. M. Rodrigues, M. E., Roque, F. O., Sandin, L., Schmera, D., Sgbari, L. F., Simaika, J. P., Siqueira, T., Thompson, R. M. & Tosnwend, C. R. (2015): A comparative analysis reveals weak relationships between ecological factors and beta diversity of stream insect metacommunities at two spatial levels. Ecology and Evolution, Vol. 5, No. 6, p. 1235-1248.

Heino, J., Muotka, T. & Paavola, R. (2003): Determinants of macroinvertebrate in headwater diversity streams : regional and local influences. Journal of Animal Ecology, Vol. 72, p. 425–434.

James, G., Witten, D., Hastie, T. & Tibshirani, R. (2013): An introduction to statistical learning. New York: Springer.

Kennen, J. G. (1999): Relation of Macroinvertebrate Community Impairment to Catchment Characteristics in New Jersey Streams. Journal of American Water Resources Association, Vol. 35, No. 4, p. 939-955.

Koordinationsstelle BDM (2014): Biodiversitätsmonitoring Schweiz BDM. Beschreibung der Methoden und Indikatoren, Bern: Bundesamt für Umwelt.

Kyriakeas, S. A. & Watzin, M. C. (2006): Effects of adjacent agricultural activities and watershed characteristics on stream macroinvertebrate communities. Journal of the American Water Resources Association, Vol. 42, No. 2, 425–441. 425-441.

Lin, L. I.-K. (1989): A concordance correlation coefficient to evaluate reproducibility. Biometrics, Vol. 45, No. 1, p. 255-268.

Lin, G., Stralberg, D., Gong, G., Huang, Z., Ye, W. & Wu, L. (2013): Separating the Effects of Environment and Space on Tree Species Distribution: From Population to Community. Plos One, Vol. 8, No. 2, p. 1-10.

Lods-Crozet, B., Castella, E., Cambin, D., Ilg, C., Knispel, S. & Mayor-Simeant, H. (2001): Macroinvertebrate community structure in relation to environmental variables in a Swiss glacial stream. Freshwater Biology, Vol. 46, No. 2, p. 1641-1661.

McIver, J. D. & McInnis, M. L. (2007): Cattle grazing effects on macroinvertebrates in an Oregon mountain stream. Rangeland Ecology & Management, Vol. 60, No. 3, p. 293-303.

Miserendino, M. L. (2001): Macroinvertebrate assemblages in Andean Patagonian rivers and streams: Environmental relationships. Hydrobiologia, Vol. 444, No. 1-3, p. 147–158.

Moore, A. A, Palmer, M. A (2005): Invertebrate Biodiversity in Agricultural and Urban Headwater Streams : Implications for Conservation and Management. Ecological Application, Vol. 15., No. 4, p. 1169-1177.

Pearson, D. L. & Carroll, S. S. (1998). Global patterns of species richness: spatial models for conservation planning using bioindicator and precipitation data. Conservation Biology, Vol. No. 4. p. 809-821.

Reineking, B. & Schröder, B. (2006): Constrain to perform: Regularization of habitat models. Ecological Modelling, Vol. 193, No. 3-4, p. 675–690.

Richards, C., Haro, R. J., Johnson, L. B. & Host, G.E. (1997): Catchment and reach-scale properties as indicators of macroinvertebrate species traits. Freshwater Biology, Vol. 37, No. 1, p. 219-230.

Roy, A. H., Rosemond, A. D., Paul, M. J., Leigh, D. S. & Wallace, J. B. (2003): Stream macroinvertebrate response to catchment urbanisation (Georgia, U.S.A.). Freshwater Biology, Vol. 48, No. 2, p. 329–346.

Rumsey, D. J. (2011): Statistics for dummies, 2nd edition, USA: John Wiley & Sons.

Sawyer, J. A., Stewart, P. M., Mullen, M. M., Simon, T. P. & Bennett, H. H. (2004): Influence of habitat, water quality, and land use on macro-invertebrate and fish assemblages of a southeastern coastal plain watershed, USA. Aquatic Ecosystem Health & Management, Vol. 7, No. 1, p. 85–99.

Seymour, M., Deiner, K. & Altermatt, F. (2015): Scale and scope matter when explaining varying patterns of community diversity in riverine metacommunities. In review.

Sliva, L. & Williams, D. D. (2001): Buffer zone versus whole catchment approaches to studying land use impact on river water quality. Water Research, Vol. 35, No. 14, p. 3462–3472.

Snider, S. & Brimlow, J. (2013): An Introduction to Population Growth. Nature Education Knowledge, Vol. 4, No. 4, p. 3.

Stucki P. (2010): Methoden zur Untersuchung und Beurteilung der Fliessgewässer. Makrozoobenthos Stufe F, Bern: Bundesamt für Umwelt.

Tobler, W. R. (1970): A computer movie simulating urban growth in the Detroit region. Economic geography, Vol. 46, p. 234-240.

Vannote, R. L., Minshall, G. W. , Cummins, K. W., Sedell, J. R. & Cushing, C. E. (1980): The river continuum concept. Canadian Journal of Fisheries and Aquatic Sciences, Vol. 37, No. 1, 130–137.

Vörösmarty, C. J., McIntyre, P. B., Gessner, M. O., Dudgeon, D., Prusevich, A., Green, P., Glidden. S., Bunn, S. E, Sullivan, C. A., Liermann, C. R. & Davies, P. M. (2010): Global threats to human water security and river biodiversity. Nature, 467(7315), p. 555–561.

Wahl, C. M., Neils, A. & Hooper, D. (2013): Impacts of land use at the catchment scale constrain the habitat benefits of stream riparian buffers. Freshwater Biology, Vol. 58, No. 11, p. 2310-2324.

Walker, C. H., Sibly, R. M., Hopkin, S. P. & Peakall, D. B. (2012): Principles of ecotoxicology. 4th edition, USA: CRC Press.

Wittmer, I., Moschet, C.. Simovic, J., Singer, H., Stamm, C., Hollender, J., Junghans M. & Leu, C. (2014): Über 100 Pestizide in Fliessgewässern - Programm Nawa Spez zeigt die hohe Pestizidbelastund der Schweizer Fliessgewässer auf. Aqua & Gas, Vol. 94, No. 3, p. 32-43.
f
Wrona, F. J. & Dixon, W. J. (1991): Group size and predation risk: a field analysis of encounter and dilution effects. American Naturalist, Vol. 137, No. 2, p. 86-201.

8.2 Software and packages

Brian Ripley (2015). tree: Classification and Regression Trees. R package version 1.0-36. http://CRAN.R-project.org/package=tree (accessed: 2.8.15).

ESRI (2015): ArcGIS Desktop. Environmental Systems Research Institute, California. http://www.esri.com (accessed: 2.8.15).

Jerome Friedman, Trevor Hastie, Robert Tibshirani (2010): Regularization Paths for Generalized Linear Models via Coordinate Descent. Journal of Statistical Software, 33(1), 1-22. http://www.jstatsoft.org/v33/i01/ (accessed: 2.8.15).

Mark Stevenson with contributions from Telmo Nunes, Cord Heuer, Jonathon Marshall, Javier Sanchez, Ron Thornton, Jeno Reiczigel, Jim Robison-Cox, Paola Sebastiani, Peter Solymos, Kazuki Yoshida and Simon Firestone (2015): epiR: Tools for the Analysis of Epidemiological Data. R package version 0.9-62. http://CRAN.R-project.org/package=epiR (accessed: 2.8.15).

Python Team (2014): Software Foundation. http://www.python.org(accessed: 2.8.15).

QGIS Development Team (2015): QGIS Geographic Information System. Open Source Geospatial Foundation Project. http://qgis.osgeo.org (accessed: 2.8.15).

R Core Team (2014). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Austria. http://www.R-project.org (accessed: 2.8.15).

8.3 Data

Bundesamt für Energie (2012): Statistik der Wasserkraftanlagen (WASTA).

Bundesamt für Energie (2013): Stauanlagen unter Bundesaufsicht.

Bundesamt für Statistik (2004): Arealstatistik nach Nomenklatur 2004 – Standard.

Bundesamt für Umwelt (2013): MQ-GWN-CH.

Bundesamt für Umwelt (n.d.): Gewässerabschnittsbasierte Einzugsgebietsgliederung der Schweiz GAB-EZGG-CH mit zusätzlicher Darstellung von Landnutzungsdaten (Swisstopo (2010): Vector25, Vector-25 Daten; Bundesamt für Statistik (n.d.): Areal-Statistik; Bundesamt für Statistik (n.d.): Landwirtschaftliche Betriebszählung 2008, Ackerkulturen; Amtliche Vermessung Schweiz / FL (n.d.): DM.01-AV_CH, Daten der Amtlichen Vermessung; Swisstopo (n.d.): swissBUILDINGS 3D, Gebäude; Bundesamt für Umwelt (n.d.): ARA-Datenbank, ARA-Daten).

Bundesamt für Umwelt (n.d): Einzugsgebietsgliederung Schweiz EZGG-CH.

Bundesamt für Umwelt (n.d.): Fliessgewässertypisierung (Swisstopo (2007).

Bundesamt für Statistik (n.d.): Waldmischungsgrad der Schweiz.

Bundesamt für Umwelt (n.d.): Hydrografische Gliederung – nachbearbeitete Version (basis04).

Bundesamt für Umwelt (n.d.): Bundesinventar der Auengebiete von nationaler Bedeutung.

Bundesamt für Umwelt (n.d.): Bundesinventar der Flachmoore von nationaler Bedeutung.

Bundesamt für Umwelt (n.d.): Bundesinventar der Hoch- und Übergangsmoore von nationaler Bedeutung.

Bundesamt für Umwelt (n.d.): ARA-Datenbank.

Koordinationsstelle BDM (2014): Biodiversitätsmonitoring Schweiz BDM. Beschreibung der Methoden und Indikatoren, Bern: Bundesamt für Umwelt.

Schweizerische geotechnische Kommission (n.d.): Vereinfachte geotechnische Karte der Schweiz.

Swisstopo (2007): Vector 25, Gewässernetz.

Swisstopo (2007): Vector 25, Primärflächen.

Swisstopo (n.d.) swissAlti3D.

Swisstopo (n.d.): swissBoundaries3D, Hoheitsgebiet & Hoheitsgrenze.

9. Figure Index

Figure 14	Comparison of the predicted macroinvertebrate richness values: *StepPrediction-Nationwide-AC_2km²* vs. *LassoPrediction-Nationwide-AC_2km²* (boundless EPT species richness values) and visualization of the estimated standard errors of the *StepPrediction-Nationwide-AC_2km²*.
Figure 15	Comparison of the predicted macroinvertebrate richness values with the observed macroinvertebrate richness values: *StepPrediction-BDMsites-WSB* vs. *Monitoring*, *LassoPrediction-BDMsites- WSB* vs. *Monitoring* and *Mean(Lasso,Step)Prediction-BDMsites-WSB* vs. *Monitoring* (boundless EPT species richness values).
Figure 16	Percentage of the observed macroinvertebrate richness values (*Monitoring*) per catchment and occurrence of residuals (difference between *Mean(Lasso,Step) Prediction-BDMsites-WSB* and *Monitoring*) per catchment (boundless values).
Figure 17	Residuals between the Mean(Lasso,Step)Prediction-BDMsites-WSB and the Monitoring values of the EPT species (boundless EPT species values)
Figure 18	Residuals between the Mean(Lasso,Step)Prediction-BDMsites-WSB and the Monitoring values of the IBCH taxa (boundless IBCH taxa values)
Figure 19	Comparison of the predicted macroinvertebrate richness values using different sampling areas: *StepPrediction- BDMsites-WSB* vs. *StepPrediction-BDMsites-AC_2km²* and *LassoPrediction-BDMsites-* AC_2km² vs. *LassoPrediction-BDMsites- AC_2km²*.

10. Table Index

11. Appendix

1 Scripts

1. BDM sampling site

1.1 BDMSamplingSites_NearGwn25.py

```
# ---------------------------------------------------------------------------
# BDMSamplingSites_NearGwn25.py
# Description: Moves BDM sampling sites to nearest river
# ---------------------------------------------------------------------------

# Import arcpy module
import arcpy
import csv
from dbfpy import dbf
import os
import sys
# Path
Path = "..."

# Set workspace
arcpy.env.workspace = Path

# Allow to overwrite output
arcpy.gp.overwriteOutput = True

# Define Variables: Input
Data_in = Path + "Data_in"
BDMPints = BDMPoints = Data_in + ".../BDMSamplingSites.shp"
GWN = Data_in + ".../gwn_25_l.shp"

# Define Variables: Output
NearTabeClosestDBF = "Data_out/NearTabeClosest.dbf"
NearTabe3ClosestDBF = "Data_out/NearTabe3Closest.dbf"

# Generate Near Table: Find the closest river
arcpy.GenerateNearTable_analysis (BDM, GWN, NearTabeClosestDBF,"", "LOCATION" , "NO_ANGLE",
"CLOSEST", "", "PLANAR")
print "NearTabeClosestDBF is created"

# Generate Near Table: Find the three closest rivers
arcpy.GenerateNearTable_analysis (BDM, GWN, NearTabe3ClosestDBF,"", "LOCATION" , "NO_ANGLE",
"ALL", "3", "PLANAR")
print "NearTabe3ClosestDBF is created"


# Convert .dbf Files to .csv Files
path = "..."
for dirpath, dirnames, filenames in os.walk(path):
    for filename in filenames:
        if filename.endswith('.dbf'):
            print "Converting %s to csv" % filename
            csv_fn = filename[:-4]+ ".csv"
            with open(csv_fn,'wb') as csvfile:
                in_db = dbf.Dbf(os.path.join(dirpath, filename))
```

```
        out_csv = csv.writer(csvfile)
        names = []
        for field in in_db.header.fields:
            names.append(field.name)
        out_csv.writerow(names)
        for rec in in_db:
            out_csv.writerow(rec.fieldData)
        in_db.close()
        print "Conversion %s to csv is done" % filename
    else:
      print "Filename does not end with .dbf"
```

1.2 BDMSamplingSites_NearGwn25_JoinBDMGWNByID.py

```
# ---------------------------------------------------------------------------
# BDMSamplingSites_NearGwn25_JoinBDMGWNByID.py
# Description: Joins BDM and GWN attributes to BDM sampling sites via common ID
# ---------------------------------------------------------------------------

# Import arcpy module
import arcpy
import itertools

# Path
Path = "..."

# Set workspace
arcpy.env.workspace = Path

# Allow to overwrite output
arcpy.gp.overwriteOutput = True

# Define Variables: Input
IN_FID = "IN_FID"
FID = "FID"
NEAR_FID = "NEAR_FID"
Data_in = Path + "Data_in"
BDMPoints_NearGWN = Data_in + ".../BDMSamplingSites_NearGwn25.shp"
BDM = Data_in + ".../BDMSamplingSites.shp"
GWN = Data_in + ".../gwn_25_l.shp"
print "Input Variables are defined"

# Define Variables: Output
Data_out = Path + "Data_out"
BDMPoints_NearGwn25_JoinAttributeByID_BDM_GWN = Data_out +
".../BDMPoints_NearGwn25_JoinAttributeByID_BDM_GWN.shp"
print "Output Variables are defined"

# Create Output
arcpy.CopyFeatures_management (BDMPoints_NearGWN,
BDMPoints_NearGwn25_JoinAttributeByID_BDM_GWN)
print "Output is created"
```

```
# Join BDM and GWN
arcpy.JoinField_management (BDMPoints_NearGwn25_JoinAttributeByID_BDM_GWN, IN_FID, BDM,
FID)
arcpy.JoinField_management (BDMPoints_NearGwn25_JoinAttributeByID_BDM_GWN, NEAR_FID, GWN,
FID)
print "Join is done"
```

1.3 BDMSamplingSites_NearGWN25_JoinGABByLocation.py

```
# ---------------------------------------------------------------------------
# BDMSamplingSites_NearGWN25_JoinGABByLocation.py
# Description: Joins GAB attributes to BDM sampling sites via spatial query
# ---------------------------------------------------------------------------

# Import arcpy module
import arcpy
import itertools

# Path
Path = "..."

# Set workspace
arcpy.env.workspace = Path

# Allow to overwrite output
arcpy.gp.overwriteOutput = True

# Define Variables: Input
Data_in = Path + "Data_in"
BDMPoints_NearGwn25_JoinAttributeByID_BDM_GWN = Data_in +
".../BDMPoints_NearGwn25_JoinAttributeByID_BDM_GWN.shp"
GAB = Data_in + ".../GAB_EZGG_CH.shp"
print "Input Variables are defined"

# Define Variables: Output
Data_out = Path + "Data_out"
BDMPoints_NearGWN25_JoinAttributeByLocation_GAB = Data_out +
".../BDMPoints_NearGWN25_JoinAttributeByLocation_GAB.shp"
print "Output Variables are defined"

# Create Output
arcpy.CopyFeatures_management (BDMPoints_NearGwn25_JoinAttributeByID_BDM_GWN,
BDMPoints_NearGWN25_JoinAttributeByLocation_GAB)
print  "Output is created"

# Spatial Join GAB
arcpy.SpatialJoin_analysis(BDMPoints_NearGwn25_JoinAttributeByID_BDM_GWN, GAB,
BDMPoints_NearGWN25_JoinAttributeByLocation_GAB , "JOIN_ONE_TO_MANY", "KEEP_COMMON")#
macth option = Intersect
print "Join is done"
```

1.4 BDMSamplingSites_NearGwn25_CorrectGWNROWID ArcGis Field Calculator Query

```
# ---------------------------------------------------------------------------------------------------------------------
# BDMSamplingSites_NearGwn25_CorrectGWNROWID ArcGis Field Calculator Query
# Description: Checks the plausibility of the “Generate Near Table tool” by comparing the watercourse
# IDs of the GWN dataset (attribute: ‘GWN_OBJECT)  with the watercourse IDs of the GAB dataset
# (attribute: ‘GABObID_G’). If ‘GWN_OBJECT’ !=  ‘GABObID_G’  (GWN_GAB_ID = 0) the most
# suitable watercourse ID is applied to the BDM sampling site after a visual inspection (attribute:
# ‘GWN_CorrID’).
# ---------------------------------------------------------------------------------------------------------------------

# Compare ‘GWN_OBJECT with ‘GABObID_G’
CASE
    WHEN "OBJECTID_1" = "OBJECTID_G" THEN 1
    ELSE 0
END

# The attributes do not match for 29 features
# --> For 27 of the 29 mismatching attributes the ‘GWN_OBJECT‘ was most suited
# --> For the BDMRowIDs 172 and 308 neither the  most suitable watercourse ID is added manually
```

1.5 BDMPoints_NearGwn25_CorrectGWNROWID_JoinBDMGWNByID.py

```
# ---------------------------------------------------------------------------------------------------------------------
# 5_BDMPoints_NearGwn25_CorrectGWNROWID_JoinAttributeByID.py
# Description: 1) Joins correct BDM and GWN attributes to BDM sampling sites via a common ID;
# 2)  Separates catchments based on their area (> 2km2 and < 2km2)
# ---------------------------------------------------------------------------------------------------------------------

# Import arcpy module
import arcpy
import itertools

# Path
Path = "..."

# Set workspace
arcpy.env.workspace = Path

# Allow to overwrite output
arcpy.gp.overwriteOutput = True

# Define Variables: Input
GWN_CorrID = "GWN_CorrID"
OBJECTID  = "OBJECTID"
OBJECTID_G = "OBJECTID_G" # ObjekctID Gewässernetz
FID = "FID"
IN_FID = "IN_FID"
Data_in = Path + "Data_in"
BDMPoints_NearGwn25_CorrectGWNROWID = Data_in +
".../BDMPoints_NearGwn25_CorrectGWNROWID.shp"
GWN = Data_in + ".../gwn_25_l.shp"
GAB = Data_in + ".../GAB_EZGG_CH.shp"
BDM = Data_in + ".../BDMSamplingSites.shp"
Result = "Result/BDMPoints.shp" # In the Result the Attributes where renamed so that understanding is
easier
print "Input Variables are defined"

# Define Variables: Output
Data_out = Path + "Data_out"
BDMPoints_NearGwn25_JoinAttributeByID_BDM_GWN_GAB = Data_out +
".../BDMPoints_NearGwn25_JoinAttributeByID_BDM_GWN_GAB.shp"
BDMPoints_Gr2km2 = "Result/BDMPoints_Gr2km2.shp"
BDMPoints_Kl2km2 = "Result/BDMPoints_Kl2km2.shp"
print "Output Variables are defined"

# Create Output
arcpy.CopyFeatures_management (BDMPoints_NearGwn25_CorrectGWNROWID,
BDMPoints_NearGwn25_JoinAttributeByID_BDM_GWN_GAB)
print "Output is created"

### Delete fields that are not wished
List1 = [f.name for f in arcpy.ListFields(BDMPoints_NearGwn25_JoinAttributeByID_BDM_GWN_GAB)]
print "\nList1\n: " + str(List1)
List2 = [elem for elem in List1 if 'IN_FID' not in elem if 'Near_FID' not in elem if 'OBJECTID_1' not in elem if
'OBJECTID_G' not in elem if 'GWN' not in elem if 'Check' not in elem if 'Shape' not in elem if 'FID' not in
elem]
print "\nList2\n: " + str(List2)
arcpy.DeleteField_management (BDMPoints_NearGwn25_JoinAttributeByID_BDM_GWN_GAB, List2)
List3 = [f.name for f in arcpy.ListFields(BDMPoints_NearGwn25_JoinAttributeByID_BDM_GWN_GAB)]
print "\nList3\n" + str(List3)
print "Fields are deleted"

# Join BDM via IN_FID
arcpy.JoinField_management (BDMPoints_NearGwn25_JoinAttributeByID_BDM_GWN_GAB, IN_FID,
BDM, FID)
# Join GWN via GWN_CorrID
arcpy.JoinField_management (BDMPoints_NearGwn25_JoinAttributeByID_BDM_GWN_GAB,
GWN_CorrID, GWN, OBJECTID)
# Join GAB via GWN_CorrID
arcpy.JoinField_management (BDMPoints_NearGwn25_JoinAttributeByID_BDM_GWN_GAB,
GWN_CorrID, GAB, OBJECTID_G)
print "Join is done"

# Select catchments where A_EZG > 2km2
where_clause1 = "GAB_A_EZG > " +  str(2000000)
arcpy.Select_analysis(Result, BDMPoints_Gr2km2, where_clause1)
# Select catchments where A_EZG < 2km2
where_clause2 = "GAB_A_EZG < " +  str(2000000)
arcpy.Select_analysis(Result, BDMPoints_Kl2km2, where_clause2)
#print "catchments are selected"
```

2. BDM sampling area

2.1 BDMSamplingAreaTotalCatchment.py

```
# ---------------------------------------------------------------------------
# BDMSamplingArea_TotalCatchment.py
# Description: Defines sampling areas = total catchments
# ---------------------------------------------------------------------------

# Import arcpy module
import arcpy

# Path
Path = "..."

# Set workspace
arcpy.env.workspace = Path

# Allow to overwrite output
arcpy.gp.overwriteOutput = True

# Define Variables: Input
BDM_RowID = "BDM_RowID"
GWN_CORRID = "GWN_CorrID"
Data_in = Path + "Data_in"
BDMPoints = Data_in + ".../BDMPoints_Kl2km2.shp"
catchment = Data_in + ".../GAB_EZGG_CH.shp"
print "Input Variables are defined"

# Define Variables: Output
Data_out = Path + "Data_out"
print "Output Variables are defined"

# Create Feature Class for Append
catchment_append = "catchment_append.shp"
catchment_dissolved_all = "catchment_dissolved_all_kl2km2.shp"
catchment_all = "catchment_all_kl2km2.shp"
arcpy.CreateFeatureclass_management(Data_out, catchment_append, "POLYGON", "", "DISABLED",
"DISABLED", catchment)
arcpy.CreateFeatureclass_management(Data_out, catchment_dissolved_all, "POLYGON", "",
"DISABLED", "DISABLED", catchment)
arcpy.CreateFeatureclass_management(Data_out, catchment_all, "POLYGON", "", "DISABLED",
"DISABLED", catchment)
print "Feature Classes for Final Results are created"

### Create Field to relate Final Result to BDM_ROWID
arcpy.AddField_management(Data_out + ".../" + catchment_append, "GWN_CorrID",
"LONG","#","#","#","#","#","#","#")
arcpy.AddField_management(Data_out + ".../" + catchment_dissolved_all, "BDM_RowID",
"LONG","#","#","#","#","#","#","#")
print "BDM_RowID fields are created"

# List all GWN_CorrID field values
# GWN_CorrID_fieldvalue List
cursor = arcpy.da.SearchCursor(BDMPoints, GWN_CORRID)
GWN_CorrID_fieldvalue = sorted({int(row[0]) for row in cursor})
print "\nGWN_CorrID_fieldvalue: \n" + str(GWN_CorrID_fieldvalue) + "\n"
# Length of GWN_CorrID_fieldvalue List
print "length of GWN_CorrID_fieldvalue_list: " + str(len(GWN_CorrID_fieldvalue))

# Append all catchments where fieldvalue of GAB_ObID_G = GWN_CorrID_fieldvalue and add fieldvalue
GWN_CorrID
# Select all catchments where fieldvalue of GAB_ObID_G = GWN_CorrID_fieldvalue
counter = -1
for i in GWN_CorrID_fieldvalue:
    where_clause1 = "OBJECTID_G = " +  str(i)
    outFC1 = Data_out + ".../catch_"+ str(i)+ ".shp" # Catchment for one site, use for checking
    arcpy.Select_analysis(catchment, outFC1, where_clause1)
    print ".../catch_"+ str(i)+ ".shp is created"
    counter +=1
    print "FID= " + str(counter)
    # Append Selection to Shapefile
    arcpy.Append_management(outFC1,Data_out + ".../" + catchment_append,"NO_TEST","","")
    AddGWN_CorrID = str(i)
    print "GWN_CorrID= " +  AddGWN_CorrID
    features = arcpy.UpdateCursor(Data_out + ".../" + catchment_append)
    for feature in features:
        if feature.FID == counter:
            feature.GWN_CorrID = AddGWN_CorrID
            features.updateRow(feature)
    del feature,features
    print "Append " + str(i) + " is done and fieldvalue 'GWN_CorrID' is added"
print "Append catchment is done"

# Join GAB_h1 and GAB_h2 from BDM to catchment_append
arcpy.JoinField_management (Data_out + ".../" + catchment_append, GWN_CORRID, BDMPoints,
GWN_CORRID, ["GAB_h1", "GAB_h2", "BDM_RowID"])
print "GAB_h1 and GAB_h2 are joined by attribute GWN_CORRID to catchment_append"

# Find all subcatchments upstream of the BDM Points with help of the Attributes GAB_h1 and GAB_h2 of
the catchment layer
cursor = arcpy.SearchCursor(Data_out + ".../" + catchment_append)
counter = -1
for poly in cursor:
    counter +=1
    # Define Attributes
    val_name = poly.getValue(BDM_RowID)
    val_h1 = poly.getValue("GAB_h1")
    val_h2 = poly.getValue("GAB_h2")
    # Select all base upstream of the BDM Points
    where_clause2 = "h1 >= " + str(val_h1) + " AND " + "h1 < " + str(val_h2)
    outFC2 = Data_out + ".../catch_"+ str(val_name)+ ".shp" # Catchment for one site, use for checking
    arcpy.Select_analysis(catchment, outFC2, where_clause2)
    print "catch_"+ str(val_name)+ ".shp is created"
    # Dissolve the subcatchments into one BDM_ROW ID catchment
    catchment_dissolved =  Data_out + ".../catchment_dissolved_"+ str(val_name)+ ".shp" #catchment for
one site, use for checking
```

```
    arcpy.Dissolve_management(outFC2, catchment_dissolved,
"#","#","MULTI_PART","DISSOLVE_LINES")
    print "catchment_dissolved_" + str(val_name)+ " is done (" + str(counter) + ")"
print "All catchments are dissolved"

# Access folder and list items in the folder
import os
list = os.listdir(Data_out)
# Human sorting = Natural sorting
# .sort() does not sort number ascendingly and therefore we have to use the human sorting
import re
def atoi(text):
    return int(text) if text.isdigit() else text
def natural_keys(text):
    #alist.sort(key=natural_keys) sorts in human order
    #http://nedbatchelder.com/blog/200712/human_sorting.html
    #(See Toothy's implementation in the comments)
    return [ atoi(c) for c in re.split('(\d+)', text) ]

# Final Results: Append the results to one layer
# Append catchment_dissolved to catchment_dissolved_all.shp
catchment_dissolved_list = [elem for elem in list if 'catch' in elem if 'dissolved' in elem if 'shp' in elem if 'xml'
not in elem if 'lock' not in elem if 'all' not in elem]
catchment_dissolved_list.sort(key=natural_keys)
print "\catchment_dissolved_list: \n" + str(catchment_dissolved_list) + "\n"
print "length of catchment_dissolved_list: " + str(len(catchment_dissolved_list)) + "\n"
counter = -1
for i in catchment_dissolved_list:
    counter +=1
    print "FID= " + str(counter)
    arcpy.Append_management(Data_out + ".../" + i, Data_out + ".../" + catchment_dissolved_all,
"NO_TEST","","")
    AddBDM_RowID = filter(lambda x: x.isdigit(), i)
    print "BDM_ROWID= " + AddBDM_RowID
    features = arcpy.UpdateCursor(Data_out + ".../" + catchment_dissolved_all)
    for feature in features:
        if feature.FID == counter:
            feature.BDM_RowID = AddBDM_RowID
            features.updateRow(feature)
    del feature,features
    print "Append " + i + " is done and BDM_ROWID " + AddBDM_RowID+ " is added"
print "Append catchment_dissolved is done" + "\n"
# Append catchments to catchment_all.shp
catchment_list = [elem for elem in list if 'catch' in elem if 'shp' in elem if '_' in elem if 'xml' not in elem if 'lock'
not in elem if 'all' not in elem if 'dissolved' not in elem if 'append' not in elem]
catchment_list.sort(key=natural_keys)
print "\catchment_list: \n" + str(catchment_list) + "\n"
print "length of catchment_list: " + str(len(catchment_list)) + "\n"
counter = -1
for i in catchment_list:
    arcpy.Append_management(Data_out + ".../" + i, Data_out + ".../" + catchment_all, "NO_TEST","","")
    counter +=1
    print "Append " + i + " is done and BDM_ROWID is added (" + str(counter) + ")"
print "Append catchment is done \n"
```

2.2 BDMSamplingArea_BufferedCatchment.py

```
# ---------------------------------------------------------------------------
# BDMSamplingArea_BufferedCatchment.py
# Description: Defines sampling areas with buffers
# ---------------------------------------------------------------------------

# Import arcpy module
import arcpy
import os.path

# Path
Path = "..."

# Set workspace
arcpy.env.workspace = Path

# Allow to overwrite output
arcpy.gp.overwriteOutput = True

# Define Variables: Input
BufferSize = "5 Kilometers"
BDM_RowID = "BDM_RowID"
GWN_CorrID = "GWN_CorrID"
Data_in = Path + "Data_in"
BDMPoints = Data_in + ".../BDMPoints_Gr2km2.shp"
catchment = Data_in + ".../GAB_EZGG_CH.shp"
print "Input Variables are defined"

# Define Variables: Output
Data_out = Path + "Data_out"
print "Output Variables are defined"

### Create Feature Class for Append
catchment_append = "catchment_append.shp"
catchmentbuf_dissolved_all = "catchmentbuf_dissolved_all_buf5km.shp"
buffer_all = "buffer_all_buf5km.shp"
catchment_all = "catchment_all_buf5km.shp"
arcpy.CreateFeatureclass_management(Data_out, catchment_append, "POLYGON", "", "DISABLED",
"DISABLED", catchment)
arcpy.CreateFeatureclass_management(Data_out, catchmentbuf_dissolved_all, "POLYGON", "",
"DISABLED", "DISABLED", catchment)
arcpy.CreateFeatureclass_management(Data_out, buffer_all, "POLYGON", "", "DISABLED", "DISABLED",
catchment)
arcpy.CreateFeatureclass_management(Data_out, catchment_all, "POLYGON", "", "DISABLED",
"DISABLED", catchment)
##print "Feature Classes are created"

# Create Field to realte Final Result to BDM_ROWID
arcpy.AddField_management(Data_out + ".../" + catchment_append, "GWN_CorrID",
"LONG","#","#","#","#","#","#","#","#")
```

```
arcpy.AddField_management(Data_out + ".../" + catchmentbuf_dissolved_all, "BDM_RowID",
"LONG","#","#","#","#","#","#","#")
arcpy.AddField_management(Data_out + ".../" + buffer_all, "BDM_RowID",
"LONG","#","#","#","#","#","#","#")
print "BDM_RowID fields are created"

# Create Buffer
Shape = "Shape"
cursor1 = arcpy.SearchCursor(BDMPoints)
counter = -1
for row in cursor1:
    # Define Attributes
    inFC = row.getValue(Shape) # This gives you the geometry
    val_name1 = row.getValue(BDM_RowID) # There should be no Null-values in "Plot_No", the values have
to be unique
    outFC1 = Data_out + ".../buf_"+ str(val_name1)+ ".shp" # Buffer for one site, use for checking
    # Create a Buffer around each BDM Point
    arcpy.Buffer_analysis(inFC , outFC1, BufferSize, "FULL", "ROUND", "NONE")
    counter +=1
    print "buf_" + str(val_name1) + ".shp is created (" + str(counter) + ")"
print "All buffers are created"

# List all GWN_CorrID field values
# GWN_CorrID_fieldvalue List
cursor = arcpy.da.SearchCursor(BDMPoints, GWN_CorrID)
GWN_CorrID_fieldvalue = sorted({int(row[0]) for row in cursor})
print "\nGWN_CorrID_fieldvalue: \n" + str(GWN_CorrID_fieldvalue) + "\n"
# Length of GWN_CorrID_fieldvalue List
print "length of GWN_CorrID_fieldvalue_list: " + str(len(GWN_CorrID_fieldvalue))

# Append all catchments where fieldvalue of GAB_ObID_G = GWN_CorrID_fieldvalue and add fieldvalue
GWN_CorrID
# Select all catchments where fieldvalue of GAB_ObID_G = GWN_CorrID_fieldvalue
counter = -1
for i in GWN_CorrID_fieldvalue:
    where_clause1 = "OBJECTID_G = " +  str(i)
    outFC1 = Data_out + ".../catch_"+ str(i)+ ".shp" # Catchment for one site, use for checking
    arcpy.Select_analysis(catchment, outFC1, where_clause1)
    print ".../catch_"+ str(i)+ ".shp is created"
    counter +=1
    print "FID= " + str(counter)
    # Append Selection to Shapefile
    arcpy.Append_management(outFC1,Data_out + ".../" + catchment_append,"NO_TEST","","")
    AddGWN_CorrID = str(i)
    print "GWN_CorrID= " +  AddGWN_CorrID
    features = arcpy.UpdateCursor(Data_out + ".../" + catchment_append)
    for feature in features:
        if feature.FID == counter:
            feature.GWN_CorrID = AddGWN_CorrID
            features.updateRow(feature)
    del feature,features
    print "Append " + str(i) + " is done and fieldvalue 'GWN_CorrID' is added"
print "Append catchment is done"
```



```
# Join GAB_h1 and GAB_h2 from BDM to catchment_append
arcpy.JoinField_management (Data_out + ".../" + catchment_append, GWN_CorrID, BDMPoints,
GWN_CorrID, ["GAB_h1", "GAB_h2", "BDM_RowID"])
print "GAB_h1 and GAB_h2 are joined by attribute GWN_CORRID to catchment_append"

# Find all subcatchments upstream of the BDM Points with help of the Attributes GAB_h1 and GAB_h2 of
the catchment layer
cursor = arcpy.SearchCursor(Data_out + ".../" + catchment_append)
counter = -1
for poly in cursor:
    counter +=1
    # Define Attributes
    val_name = poly.getValue(BDM_RowID)
    val_h1 = poly.getValue("GAB_h1")
    val_h2 = poly.getValue("GAB_h2")
    # Select all base upstream of the BDM Points
    where_clause2 = "h1 >= " + str(val_h1) + " AND " + "h1 < " + str(val_h2)
    outFC2 = Data_out + ".../catch_"+ str(val_name)+ ".shp" # Catchment for one site, use for checking
    arcpy.Select_analysis(catchment, outFC2, where_clause2)
    print "catch_"+ str(val_name)+ ".shp is created"
print "All catchments are created"

# Comparison of catchment list and buffer list: If they do not have the same length something went wrong
# Access folder and list items in the folder
import os
list1 = os.listdir(Data_out)
# Human sorting = Natural sorting
# .sort() does not sort number ascendingly and therefore we have to use the human sorting
import re
def atoi(text):
    return int(text) if text.isdigit() else text
def natural_keys(text):
    #alist.sort(key=natural_keys) sorts in human order
    #http://nedbatchelder.com/blog/200712/human_sorting.html
    #(See Toothy's implementation in the comments)
    return [ atoi(c) for c in re.split('(\d+)', text) ]
# Catchment List
catchment_list = [elem for elem in list1 if 'catch' in elem if 'shp' in elem if 'xml' not in elem if 'lock' not in elem
if 'append' not in elem if 'buf' not in elem if 'all' not in elem]
catchment_list.sort(key=natural_keys)
print "\ncatchment_list: \n" + str(catchment_list)
# Buffer List
buffer_list = [elem for elem in list1 if 'buf' in elem if 'shp' in elem if 'xml' not in elem if 'lock' not in elem if
'append' not in elem if 'catch' not in elem if 'all' not in elem]
buffer_list.sort(key=natural_keys)
print "\nbuffer_list: \n" + str(buffer_list)
# Length of catchment List
print "\nlength of catchment_lis: " + str(len(catchment_list))
# Length of buffer List
print "length of buffer_list: " + str(len(buffer_list))

# List all BDM_RowID field values
```

```
# BDM_RowID_fieldvalues List
cursor = arcpy.da.SearchCursor(BDMPoints, BDM_RowID)
BDM_RowID_fieldvalues = sorted({int(row[0]) for row in cursor})
print "\nBDM_RowID_fieldvalues: \n" + str(BDM_RowID_fieldvalues) + "\n"

# For a given BDM_Row ID select all subcatchment which a buffer of XKM intersects and dissolve the
subcatchments into one BDM_ROW ID catchment.
counter = -1
for i in BDM_RowID_fieldvalues:
    catchment = Data_out + ".../catch_" + str(i) + ".shp"
    buffer = Data_out + ".../buf_" + str(i) + ".shp"
    # Select Layer By Location
    # Select all subcatchment which a buffer of XKM intersects
    arcpy.MakeFeatureLayer_management(buffer, "buffer_lyr")
    arcpy.MakeFeatureLayer_management(catchment, "catchment_lyr")
    arcpy.SelectLayerByLocation_management("catchment_lyr", "intersect", "buffer_lyr")
    # Export Selection to Shapefile: Dissolve catchments
    # Dissolve the subcatchments into one BDM_ROW ID catchment.
    catchmentbuf_dissolved =  Data_out + ".../catchmentbuf_dissolved_"+ str(i)+ ".shp" #catchment for one
site, use for checking
    arcpy.Dissolve_management("catchment_lyr", catchmentbuf_dissolved,
"#","#","MULTI_PART","DISSOLVE_LINES")
    counter +=1
    print "catchmentbuf_dissolved_" + str(i)+ " is done(" + str(counter) + ")"
print "All catchments are dissolved"

# Final Results: Append the results to one layer and add fieldvalue BDM_ROWID
list2 = os.listdir(Data_out)
# Append catchmentbuf_dissolved to catchmentbuf_dissolved_all.shp
catchmentbuf_list = [elem for elem in list2 if 'catchmentbuf_dissolved_' in elem if 'shp' in elem if 'xml' not in
elem if 'lock' not in elem if 'append' not in elem if 'catchmentbuf_dissolved_all' not in elem]
catchmentbuf_list.sort(key=natural_keys)
print "\ncatchmentbuf_dissolved_list: \n" + str(catchmentbuf_list)+ "\n"
print "length of catchmentbuf_dissolved_list: " + str(len(catchmentbuf_list)) + "\n"
counter = -1
for i in catchmentbuf_list:
    counter +=1
    print "FID= " + str(counter)
    arcpy.Append_management(Data_out + ".../" + i, Data_out + ".../" + catchmentbuf_dissolved_all,
"NO_TEST","","")
    AddBDM_RowID = filter(lambda x: x.isdigit(), i) # filter only numbers of catchmentbuf_list to obtain
BDM_ROWID
    print "BDM_ROWID= " + AddBDM_RowID
    features = arcpy.UpdateCursor(Data_out + ".../" + catchmentbuf_dissolved_all)
    for feature in features:
        if feature.FID == counter:
            feature.BDM_RowID = AddBDM_RowID
            features.updateRow(feature)
    del feature,features
    print "Append " + i + " is done and BDM_ROWID " + AddBDM_RowID+ " is added"
print "Append catchmentbuf_dissolved is done" + "\n"
# Append buffers to buffer_all.shp
buffer_list2 = [elem for elem in list2 if 'buf' in elem if 'shp' in elem if 'xml' not in elem if 'lock' not in elem if
'append' not in elem if 'catchment' not in elem if 'buffer_all' not in elem]
buffer_list2.sort(key=natural_keys)
print "\buffer_list2: \n" + str(buffer_list2) + "\n"
print "length of buffer_list2: " + str(len(buffer_list2)) + "\n"
counter = -1
for i in buffer_list2:
    counter +=1
    print "FID= " + str(counter)
    arcpy.Append_management(Data_out + ".../" + i, Data_out + ".../" + buffer_all, "NO_TEST","","")
    AddBDM_RowID = filter(lambda x: x.isdigit(), i)
    print "BDM_ROWID= " + AddBDM_RowID
    features = arcpy.UpdateCursor(Data_out + ".../" + buffer_all)
    for feature in features:
        if feature.FID == counter:
            feature.BDM_RowID = AddBDM_RowID
            features.updateRow(feature)
    del feature,features
    print "Append " + i + " is done and BDM_ROWID " + AddBDM_RowID+ " is added"
print "Append buffer is done \n"
#Append catchments to catchment_all.shp
catchment_list2 = [elem for elem in list2 if 'catch' in elem if 'shp' in elem if 'xml' not in elem if 'lock' not in
elem if 'append' not in elem if 'buf' not in elem if 'catchment_all' not in elem]
catchment_list2.sort(key=natural_keys)
print "\catchment_list2: \n" + str(catchment_list2) + "\n"
print "length of catchment_list2: " + str(len(catchment_list2)) + "\n"
counter = -1
for i in catchment_list2:
    counter +=1
    print "FID= " + str(counter)
    arcpy.Append_management(Data_out + ".../" + i, Data_out + ".../" + catchment_all, "NO_TEST","","")
    AddBDM_RowID = filter(lambda x: x.isdigit(), i)
    print "Append " + i + " is done and BDM_ROWID "
print "Append catchment is done \n"
```

2.3 BDMSamplingArea_AreaCalculationGr2km2.py

```
# ---------------------------------------------------------------------------
# BDMSamplingArea_AreaCalculationGr2km2.py
# Description: Calculates surface area of possible BDM sampling area definitions for total BDM sampling
sites that have a total catchment surface area >2km2, Total  Catchment, 5km Buffer, 10km Buffer)
# ---------------------------------------------------------------------------

# Import arcpy module
import arcpy

# Path
Path = "..."

# Set workspace
arcpy.env.workspace = Path

# Allow to overwrite output
arcpy.gp.overwriteOutput = True
```

```
# Define Variables: Input
BDM_RowID = "BDM_RowID"
Data_in = Path + "Data_in"
NearestCatchment =  Data_in + ".../catchment_append.shp"
Buffer5km = Data_in + ".../catchmentbuf_dissolved_all_buf5km.shp"
Buffer10km = Data_in + ".../catchmentbuf_dissolved_all_buf10km.shp"
TotalCatchment = Data_in + ".../catchment_dissolved_all_gr2km2.shp"
print "Input Variables are defined"

# Define Variables: Output
Data_out = Path + "Data_out"
AreaComparison = Data_out + ".../AreaComparison_Gr2km2.shp"
print "Output Variables are defined"

# Calculate Areas
# Nearest Catchment
arcpy.AddField_management(NearestCatchment, "AreaNearC", "DOUBLE")
arcpy.CalculateField_management(NearestCatchment, "AreaNearC",
"!SHAPE.AREA@SQUAREKILOMETERS!" , "PYTHON_9.3")
print "Area for NearestCatchmentis calculated"
# 5km Buffer
arcpy.AddField_management(Buffer5km, "AreaB5km", "DOUBLE")
arcpy.CalculateField_management(Buffer5km, "AreaB5km", "!SHAPE.AREA@SQUAREKILOMETERS!" ,
"PYTHON_9.3")
print "Area for Buffer 5km is calculated"
# 10km Buffer
arcpy.AddField_management(Buffer10km, "AreaB10km", "DOUBLE")
arcpy.CalculateField_management(Buffer10km, "AreaB10km", "!SHAPE.AREA@SQUAREKILOMETERS!"
, "PYTHON_9.3")
print "Area for or Buffer 10km is calculated"
# Total Catchment
arcpy.AddField_management(TotalCatchment, "AreaTotalC", "DOUBLE")
arcpy.CalculateField_management(TotalCatchmen, "AreaTotalC",
"!SHAPE.AREA@SQUAREKILOMETERS!" , "PYTHON_9.3")
print "Area for TotalCatchment is calculated"

# Create Feature Class for Area Comparison
arcpy.CopyFeatures_management (NearestCatchment, AreaComparison)
print "Feature Class for Append is created"

# Join via BDM_RowID
arcpy.JoinField_management (AreaComparison, BDM_RowID, Buffer5km, BDM_RowID, [ "AreaB5km"])
arcpy.JoinField_management (AreaComparison, BDM_RowID, Buffer10km, BDM_RowID, [ "AreaB10km"])
arcpy.JoinField_management (AreaComparison, BDM_RowID, TotalCatchment, BDM_RowID, [
"AreaTotalC"])
print "All Areas are appended to AreaComparison"
```

2.4 BDMSamplingArea_AreaComparisonGr2km2.R

```
# ---------------------------------------------------------------------------------------------------------------------
# BDMSamplingArea_AreaComparisonGr2km2.R
# Description: Compares surface area of different BDM sampling area definitions for total BDM
sampling sites that have a total catchment surface area >2km2, Total  Catchment, 5km Buffer, 10km
Buffer)
# ---------------------------------------------------------------------------------------------------------------------

# Clear R's memory
rm(list=ls())

# Import Data
EZG_Gr2km2_AreaComparison <- read.csv(".../AreaComparison_Gr2km2.csv", header=T, sep="\t")
AE_2km2 <- read.csv(".../AE_2km2_Area_CH_2.txt", header=T, sep="\t")

## Check Data
str(EZG_Gr2km2_AreaComparison)
str(AE_2km2)
str(AE_2km2)
dim(AE_2km2)

# Histogram
# http://stackoverflow.com/questions/3541713/how-to-plot-two-histograms-together-in-r

options(scipen=10000) # Force R to stop plotting abbreviated axis labels: e.g. 1e+00

AE_2km2_Area_km <- AE_2km2$Shape_Area/1000000

p1 <- hist(EZG_Gr2km2_AreaComparison$AreaB5km, xlim=c(0,250),  breaks = 25)
p2 <- hist(EZG_Gr2km2_AreaComparison$AreaB10km, xlim=c(0,250),  breaks = 50)
p3 <- hist(EZG_Gr2km2_AreaComparison$AreaTotalC, xlim=c(0,250),  breaks = 2000)
p4 <- hist(AE_2km2_Area_km, xlim=c(0,100),  breaks = 200)

plot(p1)
plot(p2)
plot(p3)
plot(p4)

range(EZG_Gr2km2_AreaComparison$AreaB5km)
range(EZG_Gr2km2_AreaComparison$AreaB10km)
range(EZG_Gr2km2_AreaComparison$AreaTotalC)
range(AE_2km2_Area_km)

par(mfrow = c(1,3))
plot( p1, col=rgb(1,1,1,1/4), xlim=c(0,200),  ylim=c(0,80), main = "Sampling area defined with 5km
buffer", xlab = expression("Study area [km"^2 ~ "]"))  # first histogram
plot( p2, col=rgb(1,1,1,1/4), xlim=c(0,200),  ylim=c(0,80), main = "Sampling area defined with 10km
buffer", xlab = expression("Study area [km"^2 ~ "]"))  # first histogram
plot( p3, col=rgb(1,1,1,1/4), xlim=c(0,12000),  ylim=c(0,80), main = "Sampling area = total catchment",
xlab = expression("Study area [km"^2 ~ "]"))  # first histogram

p4 <- hist(AE_2km2_Area_km, plot=F)
p4$counts
p4$counts <- log10(p4$counts)
plot(p4, ylab='log10(Frequency)', main = "Study area = sub-catchment", xlab = expression("study area
[km"^2 ~ "]"), xlim=c(0.001,400), ylim=c(0.001,5))
```

```
# Area Compariosn
sum(EZG_Gr2km2_AreaComparison$AreaB5km < 0) # how many features have a smaller area than
0km sum(EZG_Gr2km2_AreaComparison$AreaB5km > 0) # how many features have a bigger area
than 0km sum(EZG_Gr2km2_AreaComparison$AreaB5km < 2) # how many features have a smaller
area than 0km sum(EZG_Gr2km2_AreaComparison$AreaB5km > 2)
sum(EZG_Gr2km2_AreaComparison$AreaB5km > 10)
sum(EZG_Gr2km2_AreaComparison$AreaB5km > 20)
sum(EZG_Gr2km2_AreaComparison$AreaB5km > 30)
sum(EZG_Gr2km2_AreaComparison$AreaB5km > 40)
sum(EZG_Gr2km2_AreaComparison$AreaB5km > 50)
sum(EZG_Gr2km2_AreaComparison$AreaB5km > 60)
sum(EZG_Gr2km2_AreaComparison$AreaB5km > 70)
sum(EZG_Gr2km2_AreaComparison$AreaB5km > 80)
sum(EZG_Gr2km2_AreaComparison$AreaB5km > 100)

sum(EZG_Gr2km2_AreaComparison$AreaB10km < 0)
sum(EZG_Gr2km2_AreaComparison$AreaB10km > 0)
sum(EZG_Gr2km2_AreaComparison$AreaB10km < 2)
sum(EZG_Gr2km2_AreaComparison$AreaB10km > 2)
sum(EZG_Gr2km2_AreaComparison$AreaB10km > 10)
sum(EZG_Gr2km2_AreaComparison$AreaB10km > 20)
sum(EZG_Gr2km2_AreaComparison$AreaB10km > 30)
sum(EZG_Gr2km2_AreaComparison$AreaB10km > 40)
sum(EZG_Gr2km2_AreaComparison$AreaB10km > 50)
sum(EZG_Gr2km2_AreaComparison$AreaB10km > 60)
sum(EZG_Gr2km2_AreaComparison$AreaB10km > 70)
sum(EZG_Gr2km2_AreaComparison$AreaB10km > 80)
sum(EZG_Gr2km2_AreaComparison$AreaB10km > 100)

sum(AE_2km2_Area_km > 0)
sum(AE_2km2_Area_km < 2)
sum(AE_2km2_Area_km > 2)
sum(AE_2km2_Area_km < 5)
sum(AE_2km2_Area_km > 10)
```

2.5 BDMSamplingArea_Append.py

```
# ---------------------------------------------------------------------------
# BDMSamplingArea_Comparison.py
# Description: Appends sampling areas  (EZG_Kl2km2 and EZG_Gr2km2)
# ---------------------------------------------------------------------------

# Import arcpy module
import arcpy
import itertools

# Path
Path = "..."

# Set workspace
arcpy.env.workspace = Path

# Allow to overwrite output
arcpy.gp.overwriteOutput = True

# Define Variables: Input
BDM_RowID = "BDM_RowID"
AreaTotalC = "AreatTotalC"
AreaNearC = "AreaNearC"
Data_in = Path + ".../Data_in"
EZG_Kl2km2_Area = Data_in +  ".../EZG_Kl2km2_Area.shp"
EZG_Gr2km2_Area = Data_in + ".../EZG_Gr2km2_Area.shp"
print "Input Variables are defined"

# Define Variables: Output
AreaCheck = "AreaCheck" # Checks if AreaTotalC has the right value
AreaCheck2 = "AreaCheck2" # Checks if AreaTotalC = AreaCheck2
AreaNT = "AreaNT" # Calculates ratio of AreaNearC/AreaTotalC
Data_out = Path + ".../Data_out"
EZG_All = Data_out + ".../EZG_All_GAB.shp"
EZG_All_test = Data_out + ".../EZG_All_test.shp"
print "Output Variables are defined"

### List fields withtout BDM_RowID, AreaNearC and Area TotalC
# EZG_Kl2km2_Area
EZG_Kl2km2_Area_List1 = [f.name for f in arcpy.ListFields(EZG_Kl2km2_Area)]
print "\nEZG_Kl2km2_Area_List1\n: " + str(EZG_Kl2km2_Area_List1)
EZG_Kl2km2_Area_List2 = [elem for elem in EZG_Kl2km2_Area_List1 if 'BDM' not in elem if 'Shape' not in
elem if 'FID' not in elem if 'Area' not in elem]
print "\nEZG_Kl2km2_Area_List2\n: " + str(EZG_Kl2km2_Area_List2)
# EZG_Gr2km2_Area
EZG_Gr2km2_Area_List1 = [f.name for f in arcpy.ListFields(EZG_Gr2km2_Area)]
print "\nEZG_Gr2km2_Area_List1\n: " + str(EZG_Gr2km2_Area_List1)
EZG_Gr2km2_Area_List2 = [elem for elem in EZG_Gr2km2_Area_List1 if 'BDM' not in elem if 'Shape' not
in elem if 'FID' not in elem if 'Area' not in elem]
print "\EZG_Gr2km2_Area_List2: " + str(EZG_Gr2km2_Area_List2)

### Delete all fields except BDM_RowID, AreaNearC and Area TotalC
### EZG_Kl2km2_Area
arcpy.DeleteField_management (EZG_Kl2km2_Area, EZG_Kl2km2_Area_List2)
EZG_Kl2km2_Area_ListNew = [f.name for f in arcpy.ListFields(EZG_Kl2km2_Area)]
print "\nEZG_Kl2km2_Area_ListNew\n" + str(EZG_Kl2km2_Area_ListNew)
# EZG_Kl2km2_Area
arcpy.DeleteField_management (EZG_Gr2km2_Area, EZG_Gr2km2_Area_List2)
EZG_Gr2km2_Area_ListNew = [f.name for f in arcpy.ListFields(EZG_Gr2km2_Area)]
print "\nEZG_Gr2km2_Area_ListNew\n" + str(EZG_Gr2km2_Area_ListNew)
# !!!The field AreaB5km was deleted manually!!!

# Add Field Gr/Kl 2km2
# EZG_Kl2km2_Area
arcpy.AddField_management(EZG_Kl2km2_Area, "GrKl2km2", "Text")
cursor1 = arcpy.da.UpdateCursor (EZG_Kl2km2_Area, "GrKl2km2")
for row in cursor1:
    row[0] = "Kl2km2"
    cursor1.updateRow(row)
del row
```

```
del cursor1
print "Field Gr/Kl 2km2 is added to EZG_Kl2km2_Area"
# EZG_Gr2km2_Area
arcpy.AddField_management(EZG_Gr2km2_Area, "GrKl2km2", "Text")
cursor2 = arcpy.da.UpdateCursor (EZG_Gr2km2_Area, "GrKl2km2")
for row in cursor2:
    row[0] = "Gr2km2"
    cursor2.updateRow(row)
del row
del cursor2
print "Field Gr/Kl 2km2 is added to EZG_Gr2km2_Area"

# Create Feature Class for Append
arcpy.CopyFeatures_management (EZG_Kl2km2_Area, EZG_All)
print "Feature Class EZG_All is created"

# Append
# Polygon2 is appended to Polygon1
arcpy.Append_management(EZG_Gr2km2_Area, EZG_All, "TEST","","")
print "Append is done"

# Check if appended Polygons overlap1.4:
https://arcpy.wordpress.com/2012/02/01/find-overlapping-features/
with arcpy.da.SearchCursor(EZG_All, ["BDM_RowID", "SHAPE@"]) as cur:
    for e1,e2 in itertools.combinations(cur, 2):
        if e1[1].overlaps(e2[1]):
            print "{} overlaps {}".format(e1[0],e2[0])


# Check if appended Polygons contain each other:
https://arcpy.wordpress.com/2012/02/01/find-overlapping-features/
with arcpy.da.SearchCursor(EZG_All, ["BDM_RowID", "SHAPE@"]) as cur:
    for e1,e2 in itertools.combinations(cur, 2):
        if e1[1].contains(e2[1]):
            print "{} contains {}".format(e1[0],e2[0])

# Check if AreaTotalC is correct
# Calculate Area
arcpy.AddField_management(EZG_All, AreaCheck, "DOUBLE")
arcpy.CalculateField_management(EZG_All, AreaCheck, "!SHAPE.AREA@SQUAREKILOMETERS!" ,
"PYTHON_9.3")
print "Area check calculated"

# Compare field values: Check if AreaTotalC = AreaCheck
arcpy.AddField_management(EZG_All, "Check", "TEXT")
# Create update cursor for feature class
with arcpy.da.UpdateCursor(EZG_All, ('AreaTotalC', 'AreaCheck', 'Check')) as cursor:
    # For each row, evaluate the Shape_Ar_1 and Shape_Ar_2 values (index position
    #  of 0 and 1), and update Same (index position of 2)
    for row in cursor:
        if (row[0] == row[1]):
            row[2] = 1
        else:
            row[2] = 0
        cursor.updateRow(row)
print "Check if AreaTotalC = AreaCheck is done"

# Add Field AreaNT
arcpy.AddField_management(EZG_All, AreaNT, "Double")
arcpy.CalculateField_management(EZG_All, AreaNT, "!AreaNearC!/!AreaTotalC!" , "PYTHON_9.3")
AreaTotalC = "AreatTotalC"
AreaNearC = "AreaNearC"
```

2.6 BDMSamplingArea_JoinBDMByID.py

```
# ---------------------------------------------------------------------------
# BDMSamplingArea_JoinBDMByID.py
# Description: Joins BDM attributes (BDMPoints) to sampling areas (EZG_All_GAB ) via common ID
# ---------------------------------------------------------------------------

# Import arcpy module
import arcpy

# Path
Path = "..."

# Set workspace
arcpy.env.workspace = Path

# Allow to overwrite output
arcpy.gp.overwriteOutput = True

# Define Variables: Input
Data_in = Path + "Data_in"
BDMPoints = Data_in + ".../BDMPoints.shp"
EZG_All_GAB = Data_in + ".../EZG_All_GAB.shp"
print "Input Variables are defined"

# Define Variables: Output
Data_out = Path + "Data_out"
EZG_All_GAB_IntersectBDM = Data_out + ".../EZG_All_GAB_SpatialJoinBDMPoints.shp"
print "Output Variables are defined"

# Join BDM_Point Attributes to EZG_All_GAB
arcpy.CopyFeatures_management (EZG_All_GAB, EZG_All_GAB_IntersectBDM)
arcpy.JoinField_management (EZG_All_GAB_IntersectBDM, "BDM_RowID", BDMPoints, "BDM_RowID")
print "Join is done"
```

3. Explanatory variables

3.1 GAB_JoinAttributesByID ArcGis Field Calculator Query

```
# ---------------------------------------------------------------------------
# GAB_JoinAttributesByID
# Description:Joins GAB and FGT attributes to GAB geometry via common ID
# ---------------------------------------------------------------------------
```

```
# Join via Python takes forever. Therefore the join has been carried out with the ArcGis Field calculator

The GAB Attributes are joined to GAB via EZG_NR (GAB_JoinAttribute_GAB.shp):
181182 of 181182 records matched by joining [EZG_NR] from <GAB_EZG_CH> with [EZG_NR] from
<LW_SUB>
181182 of 181182 records matched by joining [GAB_EZG_CH.EZG_NR] from <GAB_EZG_CH_LW_SUB>
with [EZG_NR] from <SIED_DEPO_SUB>
181182 of 181182 records matched by joining [GAB_EZG_CH.EZG_NR] from
<GAB_EZG_CH_LW_SUB_SIED_DEPO_SUB> with [EZG_NR] from <ARA_MW_SUB>
181182 of 181182 records matched by joining [GAB_EZG_CH.EZG_NR] from
<GAB_EZG_CH_LW_SUB_SIED_DEPO_SUB_ARA_MW_SUB> with [EZG_NR] from <VERK_SUB>
181182 of 181182 records matched by joining [GAB_EZG_CH.EZG_NR] from
<GAB_EZG_CH_LW_SUB_SIED_DEPO_SUB_ARA_MW_SUB_VERK_SUB> with [EZG_NR] from
<MJA_SUB>

The discharge Attributes are joined via OBJECTID_gwn25 (GAB_JoinAttribute_GAB_FGT.shp): :
30224 of 181182 records matched by joining [OBJECTID_G] from <GAB_JoinAttribute_GAB> with
[OBJECTID_gwn25] from <MQ_GWN_CH.txt>

The slope Attribute are joined via  OBJECTID_gwn25 (GAB_JoinAttribute_GAB_FGT.shp):
181179 of 181182 records matched by joining [GAB_JoinAttribute_GAB.OBJECTID_G] from
<GAB_JoinAttribute_GAB_txt> with [OBJECTID] from <gwn_3D>
```

3.2 GAB_ToPoint.py

```
# ---------------------------------------------------------------------------------------------------------------------
# GAB_ToPoint.py
# Description: Converts GAB Polygons to GAB Points
# ---------------------------------------------------------------------------------------------------------------------

# Import arcpy module
import arcpy
import itertools

# Path
Path = "..."

# Set workspace
arcpy.env.workspace = Path

# Allow to overwrite output
arcpy.gp.overwriteOutput = True

# Define Variables: Input
Data_in = Path + "Data_in"
GAB_Polygon = Data_in + ".../GAB_JoinAttribute_GAB_FGT.shp"
print "Input Variables are defined"

# Define Variables: Output
Data_out = Path + "Data_out"
GAB_FGT_Point = Data_out + ".../GAB_FGT_Point.shp"
print "Output Variables are defined"

# GAB Polygon to GAB Point
arcpy.FeatureToPoint_management(GAB_Polygon, GAB_FGT_Point, "INSIDE")
print "GAB_Point is created"
```

3.3 IntersectGABandFGTwithBDMSamplingAreas.py

```
# ---------------------------------------------------------------------------------------------------------------------
# IntersectGABandFGTwithBDMSamplingAreas.py
# Description: Intersects the GAB and FGT dataset with the BDM sampling areas (The intersect of the dam
and hydropower datasets with the BDM sampling areas are obtained in the same manner)
# ---------------------------------------------------------------------------------------------------------------------

# Import arcpy module
import arcpy

# Path
Path = "..."

# Set workspace
arcpy.env.workspace = Path

# Allow to overwrite output
arcpy.gp.overwriteOutput = True

# Define Variables: Input
Data_in = Path + "Data_in"
EZG_All_Corr_NoOverlap1_GAB = Data_in + ".../EZG_All_Corr_NoOverlap1_GAB.shp"
EZG_All_Corr_NoOverlap2_GAB = Data_in + ".../EZG_All_Corr_NoOverlap2_GAB.shp"
GAB_FGT_Point = Data_in + ".../GAB_FGT_Point.shp"
print "Input Variables are defined"

# Define Variables: Output
Data_out = Path + "Data_out"
GAB_FGT_Attributes_NoOverlap1 = Data_out + ".../GAB_FGT_Attributes_NoOverlap1.shp"
GAB_FGT_Attributes_NoOverlap2 = Data_out + ".../GAB_FGT_Attributes_NoOverlap2.shp"
print "Output Variables are defined"

# Intersect EZG_All with GAB_Point
arcpy.Intersect_analysis([EZG_All_Corr_NoOverlap1_GAB, GAB_FGT_Point],
GAB_FGT_Attributes_NoOverlap1, "All")
arcpy.Intersect_analysis([EZG_All_Corr_NoOverlap2_GAB, GAB_FGT_Point],
GAB_FGT_Attributes_NoOverlap2, "All")
print "Intersect is done"
```

3.4 IntersectCanalWithBDMSamplingAreas.py

```
# ---------------------------------------------------------------------------------------------------------------------
# IntersectCanalWithBDMSamplingAreas.py
# Description: Intersects the canal dataset with the BDM sampling areas
# ---------------------------------------------------------------------------------------------------------------------

# Import arcpy module
import arcpy
```

```
# Path
Path = "..."

# Set workspace
arcpy.env.workspace = Path

# Allow to overwrite output
arcpy.gp.overwriteOutput = True

# Define Variables: Input
Data_in = Path + "Data_in"
EZG_All_Corr_NoOverlap1_GAB = Data_in + ".../EZG_All_Corr_NoOverlap1_GAB.shp"
EZG_All_Corr_NoOverlap2_GAB = Data_in + ".../EZG_All_Corr_NoOverlap2_GAB.shp"
Kanal = Data_in + ".../Kanal.shp"
print "Input Variables are defined"

# Define Variables: Output
Data_out = Path + "Data_out"
Kanal_NoOverlap1 = Data_out + ".../Kanal_NoOverlap1.shp"
Kanal_NoOverlap2 = Data_out + ".../Kanal_NoOverlap2.shp"
print "Output Variables are defined"

# Intersect EZG_All with GAB_Point
arcpy.Intersect_analysis([EZG_All_Corr_NoOverlap1_GAB, Kanal], Kanal_NoOverlap1, "All")
arcpy.Intersect_analysis([EZG_All_Corr_NoOverlap2_GAB, Kanal], Kanal_NoOverlap2, "All")
#print "Intersect is done"

#Calculate Kanal Length within BDM EZG
arcpy.AddField_management(Kanal_NoOverlap1, "LenKanal", "DOUBLE")
arcpy.CalculateField_management(Kanal_NoOverlap1, "LenKanal", "!SHAPE.LENGTH@METERS!" ,
"PYTHON_9.3")
arcpy.AddField_management(Kanal_NoOverlap2, "LenKanal", "DOUBLE")
arcpy.CalculateField_management(Kanal_NoOverlap2, "LenKanal", "!SHAPE.LENGTH@METERS!" ,
"PYTHON_9.3")
print "Length Kanal is calculated"
```

3.5 IntersectHydroRegionWithBDMSamplingAreas.py

```
# ---------------------------------------------------------------------------
# IntersectHydroRegionWithBDMSamplingAreas.py
# Description: Intersects the hydrological regions with the BDM sampling areas (The intersect of the
BioRegion dataset with the BDM sampling areas is obtained in the same manner)
# ---------------------------------------------------------------------------

# Import arcpy module
import arcpy

# Path
Path = "..."

# Set workspace
arcpy.env.workspace = Path

# Allow to overwrite output
arcpy.gp.overwriteOutput = True

# Define Variables: Input
Data_in = Path + "Data_in"
BDMPoints = Data_in + ".../BDMPoints.shp"
BioggRegionen = Data_in + ".../biogreg.shp"
print "Input Variables are defined"

# Define Variables: Output
Data_out = Path + "Data_out"
BioggRegionen2 = Data_out + ".../BioggRegionen2.shp"
print "Output Variables are defined"

# Intersect EZG_All with GAB_Point
arcpy.Intersect_analysis([BDMPoints, BioggRegionen], BioggRegionen2, "All")
print "Intersect is done"
```

3.6 IntersectCanalWithBDMSamplingAreas.py

```
# ---------------------------------------------------------------------------
# IntersectCanalWithBDMSamplingAreas.py
# Description: Intersects the canal dataset with the BDM sampling areas
# ---------------------------------------------------------------------------

# Import arcpy module
import arcpy

# Path
Path = "..."

# Set workspace
arcpy.env.workspace = Path

# Allow to overwrite output
arcpy.gp.overwriteOutput = True

# Define Variables: Input
Data_in = Path + "Data_in"
EZG_All_Corr_NoOverlap1_GAB = Data_in + ".../EZG_All_Corr_NoOverlap1_GAB.shp"
EZG_All_Corr_NoOverlap2_GAB = Data_in + ".../EZG_All_Corr_NoOverlap2_GAB.shp"
Kanal = Data_in + ".../Kanal.shp"
print "Input Variables are defined"

# Define Variables: Output
Data_out = Path + "Data_out"
Kanal_NoOverlap1 = Data_out + ".../Kanal_NoOverlap1.shp"
Kanal_NoOverlap2 = Data_out + ".../Kanal_NoOverlap2.shp"
print "Output Variables are defined"

# Intersect EZG_All with GAB_Point
arcpy.Intersect_analysis([EZG_All_Corr_NoOverlap1_GAB, Kanal], Kanal_NoOverlap1, "All")
arcpy.Intersect_analysis([EZG_All_Corr_NoOverlap2_GAB, Kanal], Kanal_NoOverlap2, "All")
#print "Intersect is done"
```

```
#Calculate Kanal Length within BDM EZG
arcpy.AddField_management(Kanal_NoOverlap1, "LenKanal", "DOUBLE")
arcpy.CalculateField_management(Kanal_NoOverlap1, "LenKanal", "!SHAPE.LENGTH@METERS!" ,
"PYTHON_9.3")
arcpy.AddField_management(Kanal_NoOverlap2, "LenKanal", "DOUBLE")
arcpy.CalculateField_management(Kanal_NoOverlap2, "LenKanal", "!SHAPE.LENGTH@METERS!" ,
"PYTHON_9.3")
print "Length Kanal is calculated"
```

3.7 IntersectGeologywithBDMSamplingAreas.py

```
# ---------------------------------------------------------------------------------------------------------------------
# IntersectGeologywithBDMSamplingAreas.py
# Description: Intersects the geology dataset with the BDM sampling areas  (The intersect of the
# floodplainwetland dataset with the BDM sampling areas is obtained in the same manner)
# ---------------------------------------------------------------------------------------------------------------------

# Import arcpy module
import arcpy

# Path
Path = "..."

# Set workspace
arcpy.env.workspace = Path

# Allow to overwrite output
arcpy.gp.overwriteOutput = True

# Define Variables: Input
Data_in = Path + "Data_in"
EZG_All_Corr_NoOverlap1_GAB = Data_in + ".../EZG_All_Corr_NoOverlap1_GAB.shp"
EZG_All_Corr_NoOverlap2_GAB = Data_in + ".../EZG_All_Corr_NoOverlap2_GAB.shp"
Geologie = Data_in + ".../Geologie.shp"
print "Input Variables are defined"

# Define Variables: Output
Data_out = Path + "Data_out"
Geologie_NoOverlap1 = Data_out + ".../Geologie_NoOverlap1.shp"
Geologie_NoOverlap2 = Data_out + ".../Geologie_NoOverlap2.shp"
print "Output Variables are defined"

# Intersect EZG_All with GAB_Point
arcpy.Intersect_analysis([EZG_All_Corr_NoOverlap1_GAB, Geologie], Geologie_NoOverlap1, "All")
arcpy.Intersect_analysis([EZG_All_Corr_NoOverlap2_GAB, Geologie], Geologie_NoOverlap2, "All")
print "Intersect is done"

# Calculate Area
# Overlap 1
arcpy.AddField_management(Geologie_NoOverlap1, "AreaKM", "DOUBLE")
arcpy.CalculateField_management(Geologie_NoOverlap1, "AreaKM",
"!SHAPE.AREA@SQUAREKILOMETERS!" , "PYTHON_9.3")
# Overlap 2
arcpy.AddField_management(Geologie_NoOverlap2, "AreaKM", "DOUBLE")
arcpy.CalculateField_management(Geologie_NoOverlap2, "AreaKM",
"!SHAPE.AREA@SQUAREKILOMETERS!" , "PYTHON_9.3")
print "Area is calculated"
```

3.8 AppendFloodplainWetland.py

```
# ---------------------------------------------------------------------------------------------------------------------
# AppendFloodplainWetland.py
# Description: Appends the floodplain and wetland datasets
# ---------------------------------------------------------------------------------------------------------------------

# Import arcpy module
import arcpy
import itertools

# Path
Path = "..."

# Set workspace
arcpy.env.workspace = Path

# Allow to overwrite output
arcpy.gp.overwriteOutput = True

# Define Variables: Input
Data_in = Path + ".../Data_in"
Aue = Data_in +  ".../au.shp"
Flachmoor = Data_in + ".../fm.shp"
Hochmoor = Data_in + ".../hm.shp"
print "Input Variables are defined"

# Define Variables: Output
Data_out = Path + "Data_out"
AueMoor = Data_out + ".../AueMoor.shp"
print "Output Variables are defined"

# Add Field Type
# Aue
arcpy.AddField_management(Aue, "Type", "Text")
cursor1 = arcpy.da.UpdateCursor (Aue, "Type")
for row in cursor1:
    row[0] = "Aue"
    cursor1.updateRow(row)
del row
del cursor1
# Flachmoor
arcpy.AddField_management(Flachmoor, "Type", "Text")
cursor2 = arcpy.da.UpdateCursor (Flachmoor, "Type")
for row in cursor2:
    row[0] = "Flachmoor"
    cursor2.updateRow(row)
del row
```

```
del cursor2
# Hochmoor
arcpy.AddField_management(Hochmoor, "Type", "Text")
cursor3 = arcpy.da.UpdateCursor (Hochmoor, "Type")
for row in cursor3:
    row[0] = "Hochmoor"
    cursor3.updateRow(row)
del row
del cursor3
print "Field Type is added"

# Create Feature Class for Append
arcpy.CopyFeatures_management (Aue, AueMoor)
print "Feature Class EZG_All is created"

### List fields withtout Type
# EZG_Kl2km2_Area
AueMoor_List1 = [f.name for f in arcpy.ListFields(AueMoor)]
print "\AueMoor_List1\n: " + str(AueMoor_List1)
AueMoor_List2 = [elem for elem in AueMoor_List1 if 'Type' not in elem if 'Shape' not in elem if 'FID' not in
elem]
print "\AueMoor_List2\n: " + str(AueMoor_List2)

### Delete all fields except BDM_RowID, AreaNearC and Area TotalC
### EZG_Kl2km2_Area
arcpy.DeleteField_management (AueMoor, AueMoor_List2)
AueMoor_ListNew = [f.name for f in arcpy.ListFields(AueMoor)]
print "\nEZG_Kl2km2_Area_ListNew\n" + str(AueMoor_ListNew)

# Append
# Polygon2 is appended to Polygon1
arcpy.Append_management([Flachmoor, Hochmoor], AueMoor, "NO_TEST","Join", "")
print "Append is done"
```

3.9 AppendNoOverlapDatsets.R

```
# ------------------------------------------------------------------------------------------------------------------------
# AppendNoOverlapDatasets.R
# Description: Appends NoOverlap datasets
# ------------------------------------------------------------------------------------------------------------------------

 # Clear R's memory
rm(list=ls())

 # Import Data
NoOverlap1 <- read.csv(".../GAB_FGT_Attributes_NoOverlap1_QGis.csv", header=T, sep="\t")
NoOverlap2 <- read.csv(".../GAB_FGT_Attributes_NoOverlap2_QGis.csv", header=T, sep="\t")
# Check Data
str(NoOverlap1)
str(NoOverlap2)
dim(NoOverlap1)
dim(NoOverlap2)
```

```
 # Append
# https://kb.iu.edu/d/bcrr
datafile <- rbind(NoOverlap1,NoOverlap2) # rbind = append
dim(datafile)

 # Write appended data to csv file
write.table(datafile,"...GAB_FGT_Attributes_QGis.csv", sep="\t", quote = F, row.names = F)
```

3.10 ExplanatoryVariable_GAB.py

```
# ------------------------------------------------------------------------------------------------------------------------
# ExplanatoryVariable_GAB.py
# Description: Obtains explanatory variables from the GAB dataset
# ------------------------------------------------------------------------------------------------------------------------

# !!! trailing blank line in data file causes IndexError "list index out of range"

# Create dictionary
f = open('/GAB_FGT_Attributes_QGis.csv') # open file
data = f.read()   # read file
datalines = data.split('\n')   # split file into lines
dd = {}   #initialize dictionary
for line in datalines[1:]:
    items = line.split('\t')   # split lines into items using tab as delimiter
    # define dictionary using item[1] as key and append necessary items:
    dd.setdefault(items[1], []).append((float(items[18]),   #00 GAB: A_SUBEZG
                                        float(items[17]),   #01 GAB: FLSTR_SUBE
                                        float(items[26]),   #02 GAB: ACK
                                        float(items[27]),   #03 GAB: HUEF
                                        float(items[28]),   #04 GAB: KART
                                        float(items[29]),   #05 GAB: GETR
                                        float(items[30]),   #06 GAB: MAIS
                                        float(items[31]),   #07 GAB: OBST
                                        float(items[32]),   #08 GAB: RAPS
                                        float(items[33]),   #09 GAB: RUEB
                                        float(items[34]),   #10 GAB: GEM
                                        float(items[35]),   #11 GAB: REB
                                        float(items[39]),   #12 GAB: WALD
                                        float(items[40]),   #13 GAB: GRUE
                                        float(items[44]),   #14 GAB: FASS
                                        float(items[45]),   #15 GAB: DACH
                                        float(items[46]),   #16 GAB: FA_DACH
                                        float(items[50]),   #17 GAB: SIED
                                        float(items[51]),   #18 GAB: DEPO
                                        float(items[54]),   #19 GAB: ARA
                                        float(items[55]),   #29 GAB: MW
                                        float(items[59]),   #21 GAB: GEL_A
                                        float(items[67])))  #22 GAB: STR_A
print dd # Microsoft cannot print dictionary; it crashes
print "dictionary is  is created"

# Create output
keys = sorted(dd.keys())
```

```
output = open('/EV_GAB_20150302.csv', 'w')
output.write('BDMRowID, A_TEZG, FLSTR_SUBE_ANT, ACK_ANT, HUEF_ANT, KART_ANT,
GETR_ANT, MAIS_ANT, OBST_ANT, RAPS_ANT, RUEB_ANT,  GEM_ANT, REB_ANT, WALD_ANT,
GRUE_ANT, FASS_ANT, DACH_ANT, FA_DACH_ANT, SIED_ANT, DEPO_ANT, ARA_ANT, MW_ANT,
GEL_A_ANT, STR_A_ANT\n')
for key in keys:
    output.write(key)
    output.write(',')
    output.write(str((sum( item[0] for item in dd [key])))) #SUM A_SUBEZG   A_TEZG
    output.write(',')
    output.write(str((sum( item[1] for item in dd [key])))) #01 FLSTR_SUBE_ANT
    output.write(',')
    output.write(str((sum( item[2] for item in dd [key]) / sum( item[0] for item in dd [key]) *100))) #02
ACK_ANT
    output.write(',')
    output.write(str((sum( item[3] for item in dd [key]) / sum( item[0] for item in dd [key]) *100))) #03
HUEF_ANT
    output.write(',')
    output.write(str((sum( item[4] for item in dd [key]) / sum( item[0] for item in dd [key]) *100))) #04
KART_ANT
    output.write(',')
    output.write(str((sum( item[5] for item in dd [key]) / sum( item[0] for item in dd [key]) *100))) #05
GETR_ANT
    output.write(',')
    output.write(str((sum( item[6] for item in dd [key]) / sum( item[0] for item in dd [key]) *100))) #06
MAIS_ANT
    output.write(',')
    output.write(str((sum( item[7] for item in dd [key]) / sum( item[0] for item in dd [key]) *100))) #07
OBST_ANT
    output.write(',')
    output.write(str((sum( item[8] for item in dd [key]) / sum( item[0] for item in dd [key]) *100))) #08
RAPS_ANT
    output.write(',')
    output.write(str((sum( item[9] for item in dd [key]) / sum( item[0] for item in dd [key]) *100))) #09
RUEB_ANT
    output.write(',')
    output.write(str((sum( item[10] for item in dd [key]) / sum( item[0] for item in dd [key]) *100))) #10
GEM_ANT
    output.write(',')
    output.write(str((sum( item[11] for item in dd [key]) / sum( item[0] for item in dd [key]) *100))) #11
REB_ANT
    output.write(',')
    output.write(str((sum( item[12] for item in dd [key]) / sum( item[0] for item in dd [key]) *100))) #12
WALD_ANT
    output.write(',')
    output.write(str((sum( item[13] for item in dd [key]) / sum( item[0] for item in dd [key]) *100))) #13
GRUE_ANT
    output.write(',')
    output.write(str((sum( item[14] for item in dd [key]) / sum( item[0] for item in dd [key]) *100))) #14
FASS_ANT
    output.write(',')
    output.write(str((sum( item[15] for item in dd [key]) / sum( item[0] for item in dd [key]) *100))) #15
DACH_ANT
    output.write(',')
    output.write(str((sum( item[16] for item in dd [key]) / sum( item[0] for item in dd [key]) *100))) #16
FA_DACH_ANT
    output.write(',')
    output.write(str((sum( item[17] for item in dd [key]) / sum( item[0] for item in dd [key]) *100))) #17
SIED_ANT
    output.write(',')
    output.write(str((sum( item[18] for item in dd [key]) / sum( item[0] for item in dd [key]) *100))) #18
DEPO_ANT
    output.write(',')
    output.write(str((sum( item[19] for item in dd [key])))) #19 ARA_ANT
    output.write(',')
    output.write(str((sum( item[20] for item in dd [key])))) #20 MW_ANT
    output.write(',')
    output.write(str((sum( item[21] for item in dd [key]) / sum( item[0] for item in dd [key]) *100))) #21
GEL_A_ANT
    output.write(',')
    output.write(str((sum( item[22] for item in dd [key]) / sum( item[0] for item in dd [key]) *100))) #22
STR_A_ANT
    output.write('\n')
output.close() # without this command output will not be written
print "output is written"
```

3.11 ExplanatoryVariable_FGT.py

```
# ---------------------------------------------------------------------------------------------------------------------
# ExplanatoryVariable_FGT.py
# Description: Obtains explanatory variables from the FGT dataset
# ---------------------------------------------------------------------------------------------------------------------

# !!! trailing blank line in data file causes IndexError "list index out of range"

# Create dictionary
f = open('/GAB_FGT_Attributes_QGis.csv') # open file
data = f.read()   # read file
datalines = data.split('\n')   # split file into lines
dd = {}   #initialize dictionary
for line in datalines[1:]:
    items = line.split('\t')   # split lines into items using tab as delimiter
    # define dictionary using item[1] as key and append necessary items:
    dd.setdefault(items[1], []).append((float(items[72]),   #00 Abfluss: mqn_Jahr
                                        float(items[73]),   #01 Abfluss: mqn_Jan
                                        float(items[74]),   #02 Abfluss: mqn_Feb
                                        float(items[75]),   #03 Abfluss: mqn_Mar
                                        float(items[76]),   #04 Abfluss: mqn_Apr
                                        float(items[77]),   #05 Abfluss: mqn_Mai
                                        float(items[78]),   #06 Abfluss: mqn_Jun
                                        float(items[79]),   #07 Abfluss: mqn_Jul
                                        float(items[80]),   #08 Abfluss: mqn_Aug
                                        float(items[81]),   #09 Abfluss: mqn_Sep
                                        float(items[82]),   #10 Abfluss: mqn_Okt
                                        float(items[83]),   #11 Abfluss: mqn_Nov
                                        float(items[84]),   #12 Abfluss: mqn_Dez
```

```
                                float(items[87]),   #13 Abfluss: abflussvar
                                float(items[109]))) #14 Flussneigung: slope
# print dd # Microsoft cannot print dictionary, it crashes
print "dicionary is defined"

keys = sorted(dd.keys())
output = open('/EV_FGT.csv', 'w')

# create output
output.write('BDMRowID, mqn_Jahr_MEAN, mqn_Jan_MAX, mqn_Feb_MAX, mqn_Mar_MAX,
mqn_Apr_MAX, mqn_Mai_MAX, mqn_Jun_MAX, mqn_Jul_MAX, mqn_Aug_MAX, mqn_Sep_MAX,
mqn_Okt_MAX, mqn_Nov_MAX, mqn_Dez_MAX, abflussvar_MEAN, slope_MEAN, slope_MAX\n')
for key in keys:
    output.write(key)
    output.write(',')
    output.write(str(sum( item[0] for item in dd [key]) / len(dd [key]))) #23 mqn_Jahr_MEAN
    output.write(',')
    output.write(str(max( item[1] for item in dd [key]))) #24 mqn_Jan_MAX
    output.write(',')
    output.write(str(max( item[2] for item in dd [key]))) #25 mqn_Feb_MAX
    output.write(',')
    output.write(str(max( item[3] for item in dd [key]))) #26 mqn_Mar_MAX
    output.write(',')
    output.write(str(max( item[4] for item in dd [key]))) #27 mqn_Apr_MAX
    output.write(',')
    output.write(str(max( item[5] for item in dd [key]))) #28 mqn_Mai_MAX
    output.write(',')
    output.write(str(max( item[6] for item in dd [key]))) #29 mqn_Jun_MAX
    output.write(',')
    output.write(str(max( item[7] for item in dd [key]))) #30 mqn_Jul_MAX
    output.write(',')
    output.write(str(max( item[8] for item in dd [key]))) #31 mqn_Aug_MAX
    output.write(',')
    output.write(str(max( item[9] for item in dd [key]))) #32 mqn_Sep_MAX
    output.write(',')
    output.write(str(max( item[10] for item in dd [key]))) #33 mqn_Okt_MAX
    output.write(',')
    output.write(str(max( item[11] for item in dd [key]))) #34 mqn_Nov_MAX
    output.write(',')
    output.write(str(max( item[12] for item in dd [key]))) #35 mqn_Dez_MAX
    output.write(',')
    output.write(str(sum( item[13] for item in dd [key]) / len(dd [key]))) #36 abflussvar_MEAN
    output.write(',')
    output.write(str(sum( item[14] for item in dd [key]) / len(dd [key]))) #37 slope_MEAN
    output.write(',')
    output.write(str(max( item[14] for item in dd [key]))) #37 slope_MAX
    output.write('\n')
output.close() # without this command output will not be written
print "output is written"
```


3.12 ExplanatoryVariable_FGT_MaxQ.py

```
# ---------------------------------------------------------------------------------------------------------------------
# ExplanatoryVariable_FGT_MaxQ.py
# Description: Obtains explanatory variable MaxQ from the FGT dataset
# ---------------------------------------------------------------------------------------------------------------------

# !!! trailing blank line in data file causes IndexError "list index out of range"

# Create dictionary
f = open('/EV_FGT.csv')   # open file
data = f.read()   # read file
datalines = data.split('\n')   # split file into lines

# Create output
output = open('/EV_MaxAbfluss.csv', 'w')
output.write('BDMRowID, MaxAbfluss\n')
for line in datalines[1:]:
    items = line.split(',')   # split lines into items using comma as delimiter
    # define dictionary using item[0] as key and append necessary items:
    BDMRowID = items[0]
    list_str = (items[2:13])
    list_int = map(float, list_str) # converts list values to float values; max() does not yield correct result
    for list values
    print list_int
    MaxAbfluss = str(BDMRowID) + "," +str(max(list_int))
    print MaxAbfluss
    output.write(MaxAbfluss + "\n")
output.close() # without this command output will not be written
print "output is written"
```

3.13 ExplanatoryVariable_DamCount.py

```
# ---------------------------------------------------------------------------------------------------------------------
# ExplanatoryVariable_DamCount.py
# Description: Obtains explanatory variable dam_count (hydropower_count is obtained in the same
# manner)
# ---------------------------------------------------------------------------------------------------------------------

# Import module
import numpy

# !!! trailing blank line in data file causes IndexError "list index out of range"

# Create dictionary
f = open('/Dam.csv')
data = f.read()   # read file
datalines = data.split('\n')   # split file into lines
dd = {}   #initialize dictionary
for line in datalines[1:]:
    items = line.split('\t')   # split lines into items using tab as delimiter
    # define dictionary using item[2] as key and append necessary items:
    dd.setdefault(items[1], []).append((float(items[5]))) # 00 Area_Check
print dd # Microsoft cannot print dictionary, it crashes
```

```
print "dicionary is defined"

# Create output
keys = sorted(dd.keys())
output = open('EV_Dam.csv', 'w')
output.write('BDM_RowID, AnzDam\n')
for key in keys:
    output.write(key)
    output.write(',')
    output.write(str(len(dd [key])))
    output.write('\n')
output.close() # without this command output will not be written
print "output is written"
```

3.14 ExplanatoryVariable_CanalPercentage.py

```
# ----------------------------------------------------------------------------------------------------------------------
# ExplanatoryVariable_CanalPercentage.py
# Description: Obtains explanatory variable canal_percentage
# ----------------------------------------------------------------------------------------------------------------------

# !!! trailing blank line in data file causes IndexError "list index out of range"

# Import module
import numpy

# !!! trailing blank line in data file causes IndexError "list index out of range"

# Create dictionary
f = open(/Kanal.csv')
data = f.read()  # read file
datalines = data.split('\n')  # split file into lines
dd = {}  #initialize dictionary
for line in datalines[1:]:
    items = line.split('\t')  # split lines into items using tab as delimiter
    # define dictionary using item[2] as key and append necessary items:
    dd.setdefault(items[1], []).append((float(items[26]))) # 00 LenKanal
print dd # Microsoft cannot print dictionary, it crashes
print "dicionary is defined"

# Create output
keys = sorted(dd.keys())
output = open('/EV_Kanal.csv', 'w')
output.write('BDM_RowID, LenKanal\n')
for key in keys:
    output.write(key)
    output.write(',')
    output.write(str((sum( item for item in dd [key])))) # item[0] won't work
    output.write('\n')
output.close() # without this command output will not be written
print "output is written"
```

3.15 ExplanatoryVariable_CarbonatePerCarbonatesilicate.R

```
# ----------------------------------------------------------------------------------------------------------------------
# ExplanatoryVariable_CarbonatePerCarbonatesilicate.py
# Description: Obtains explanatory variable carbonate_per_carbonatesilicate
# ----------------------------------------------------------------------------------------------------------------------

# Clear R's memory
rm(list=ls())

# Import library
library(reshape2)

# Import Data
GeologieIn <- read.csv(".../Geologie.csv", header=T, sep="\t")

# Check Data
str(GeologieIn)
dim(GeologieIn)

# Sum all "AreaKM" values that  have the same 'BDMRowID ' and  'GEO Value '
#http://stackoverflow.com/questions/11597542/r-how-to-create-pivot-table-like-data-frame-while-3-
variables -are-involved
#http://stackoverflow.com/questions/18622854/how-to-create-a-pivot-table-in-r-with-more-than-3-variables
Pivot <- dcast(GeologieIn, BDM_RowID ~ GEO, value.var = "AreaKM", fun.aggregate=sum)
str(Pivot)

# Add a new column to dataframe and calculate the  carbonate per carbonatesilicate ratio
# https://lembra.wordpress.com/2010/03/12/adding-new-column-to-a-data-frame-in-r/
Pivot[c("GeologyRatio")]<- Pivot$karbonatisch/(Pivot$karbonatisch + Pivot$silikatisch)
str(Pivot)

# Write data to csv file
write.table(Pivot,".../EV_Geologie.csv", sep="\t", quote = F, row.names = F)
```

3.16 ExplanatoryVariable_FloodplainwetlandPercentage.py

```
# ----------------------------------------------------------------------------------------------------------------------
# ExplanatoryVariable_FloodplainwetlandPercentage.py
# Description: Appends the datasets that are needed to obtain the explanatory variable
# floodplainwetland_percentage
# ----------------------------------------------------------------------------------------------------------------------

# !!! trailing blank line in data file causes IndexError "list index out of range"

# Create dictionary
f = open(/AueMoore.csv') # open file
data = f.read()  # read file
datalines = data.split('\n')  # split file into lines
dd = {}  #initialize dictionary
for line in datalines[1:]:
```

```
    items = line.split('\t')   # split lines into items using tab as delimiter
    # define dictionary using item[2] as key and append necessary items:
    dd.setdefault(items[1], []).append((float(items[10]),  #00 Area AueMoore (km)
                                float(items[2])))  #01 Area TotalC (km)
# print dd # Microsoft cannot print dictionary, it crashes
print "dicionary is defined"

# Create output
keys = sorted(dd.keys())
output = open('...EV_AueMoore.csv', 'w')
output.write('BDM_RowID, AueMoore\n')
for key in keys:
    output.write(key)
    output.write(',')
    output.write(str((sum( item[0] for item in dd [key])/dd[key][0][1]) *100)) # Area AueMoore Area TotalC
    output.write('\n')
output.close() # without this command output will not be written
print "output is written"
```

3.17 ExplanatoryVariable_DecidiousPerForest.py

```
# ---------------------------------------------------------------------------------------------------------------------
# ExplanatoryVariable_DecidiousPerForest.py
# Description: Calculates cross-tabulated areas between sampling area and forest dataset
# ---------------------------------------------------------------------------------------------------------------------

# Import arcpy module
import arcpy
import os.path

# Check in Spatial Extension
if arcpy.CheckExtension("Spatial") == "Available":
    arcpy.CheckOutExtension("Spatial")

# Path
Path = "..."

# Set workspace
arcpy.env.workspace = Path

# Allow to overwrite output
arcpy.gp.overwriteOutput = True

# Raster Environment Settings
arcpy.env.extent = arcpy.Extent(485859, 75238, 834100, 295918)

# Define Variables: Input
Data_in = Path + "Data_in"
EZG_All_Corr_NoOverlap1_GAB = Data_in + ".../EZG_All_Corr_NoOverlap1_GAB.shp"
EZG_All_Corr_NoOverlap2_GAB = Data_in + ".../EZG_All_Corr_NoOverlap2_GAB.shp"
Waldmischungsgrad = Data_in + ".../Waldmischungsgrad/wmg100"
print "Input Variables are defined"
```

```
# Define Variables: Output
Data_out = Path + "Data_out"
EZG_All_Corr_NoOverlap1_Raster = Data_out + ".../EZG_All_Corr_NoOverlap1_10m.img"
EZG_All_Corr_NoOverlap2_Raster = Data_out + ".../EZG_All_Corr_NoOverlap2_10m.img"
Waldmischungsgrad_TabulateAreaTable_NoOverlap1_2 = Data_out +
".../Waldmischungsgrad_TabulateAreaTable_NoOverlap1_10m.txt"
Waldmischungsgrad_TabulateAreaTable_NoOverlap2 = Data_out +
".../Waldmischungsgrad_TabulateAreaTable_NoOverlap2_10m.txt"
print "Output Variables are defined"

# Make sure that the newly created raster images are aligned to the geology raster image
#!!! Setting Snap is MANDATORY!!!
Snap = Waldmischungsgrad
arcpy.env.snapRaster = Snap
# print "Snap is defined"

# Convert EZG_All_GAB to raster
arcpy.PolygonToRaster_conversion (EZG_All_Corr_NoOverlap1_GAB, "BDM_RowID",
EZG_All_Corr_NoOverlap1_Raster, "MAXIMUM_AREA" , "" ,10)
arcpy.PolygonToRaster_conversion (EZG_All_Corr_NoOverlap2_GAB, "BDM_RowID",
EZG_All_Corr_NoOverlap2_Raster, "MAXIMUM_AREA" , "" ,10)
print "Polygon to Raster conversion is done"

# Tabulate Area: Output table that will contain the summary of the area of each class in each zone.
arcpy.sa.TabulateArea(EZG_All_Corr_NoOverlap1_Raster, "Value", Waldmischungsgrad,"Value",
Waldmischungsgrad_TabulateAreaTable_NoOverlap1_2, 10)
arcpy.sa.TabulateArea(EZG_All_Corr_NoOverlap2_Raster, "Value", Waldmischungsgrad,"Value",
Waldmischungsgrad_TabulateAreaTable_NoOverlap2, 10)
print "Tabulate Area is done"

# Check out Spatial Extension
arcpy.CheckInExtension("Spatial")
```

3.18 ExplanatoryVariable_DecidiousPerForest. R

```
# ---------------------------------------------------------------------------------------------------------------------
# ExplanatoryVariable_DecidiousPerForest.R
# Description: Obtains explanatory variable decidious_per_forest
# ---------------------------------------------------------------------------------------------------------------------

# Clear R's memory
rm(list=ls())

# Import Data
# Numeric variables converted to factors when reading a CSV file: use colClasses=rep("numeric")
#
http://stackoverflow.com/questions/20060706/numeric-variables-converted-to-factors-when-reading-a-csv-fil
e
# Tipp von Emanuel: Use colClasses: Wenn ich den Typ der Spalten (z.B. Nummer, String) schon hier
definiere spare ich sehr viel Speicherplatz
WMGIn_10m <- read.csv(".../Waldmischungsgrad_TabulateAreaTable_10m.csv", header=T, sep="\t",
```

```
colClasses=rep("numeric"))

# Check Data
str(WMGIn_10m)
dim(WMGIn_10m)

# Add a new column to dataframe and calculate the Nadel/Laub Ratio
# https://lembra.wordpress.com/2010/03/12/adding-new-column-to-a-data-frame-in-r/
WMGIn_10m[c("Nadelwaldanteil")] <- (as.numeric(WMGIn_10m$VALUE_1) +
as.numeric(WMGIn_10m$VALUE_2)) / (as.numeric(WMGIn_10m$VALUE_1) +
as.numeric(WMGIn_10m$VALUE_2) + as.numeric(WMGIn_10m$VALUE_3) +
as.numeric(WMGIn_10m$VALUE_4)) # Da die Zahlen Faktoren sind, kann man keine Mathe mit ihnen
machen; as.numeric wandelt die Faktorzahlen zu Nummerzahlen um
str(WMGIn_10m)

# Write appended data to csv file
write.table(WMGIn_10m,".../EV_Waldmischungsgrad_10m.csv", sep="\t", quote = F, row.names = F)
# quote = F verhindert das Nummern (Integer/Float) zu Text (String) umgewandelt werden; R kann nicht
zwischen Nummern und Text unterscheiden
# row.names = F verhindert das eine neue ID Spalte eingefügt wird
```

3.19 ExplanatoryVariable_Masl.py

```
# ---------------------------------------------------------------------------------------------------------------------
# ExplanatoryVariable_Masl.py
# Description: Obtains explanatory variable Masl
# ---------------------------------------------------------------------------------------------------------------------

# Import arcpy module
import arcpy
import os.path
from arcpy import env
from arcpy.sa import *

# Check in Spatial Extension
if arcpy.CheckExtension("Spatial") == "Available":
    arcpy.CheckOutExtension("Spatial")

# Set workspace
arcpy.env.workspace = Path

# Allow to overwrite output
arcpy.gp.overwriteOutput = True

# Raster Environment Settings
arcpy.env.extent = arcpy.Extent(485859, 75238, 834100, 295918)

# Define Variables: Input
EZG_All_Corr_NoOverlap1_GAB = ".../EZG_All_Corr_NoOverlap1_10m.img"
EZG_All_Corr_NoOverlap2_GAB = ".../EZG_All_Corr_NoOverlap2_10m.img"
MuM = ".../swissALTI3D_2m_LV03_LN02.gdb/swissALTI3D_2m"
print "Input Variables are defined"

# Define Variables: Output
MuM_ZonalStatisticsAsTable_Overlap1_10m = ".../MuM_ZonalStatisticsAsTable_Overlap1_10m.txt"
MuM_ZonalStatisticsAsTable_Overlap2_10m = ".../MuM_ZonalStatisticsAsTable_Overlap2_10m.txt"
print "Output Variables are defined"

# Make sure that the newly created raster images are aligned to the geology raster image
#!!! Setting Snap is MANDATORY!!!
Snap = MuM
arcpy.env.snapRaster = Snap
print "Snap is defined"

# Execute ZonalStatisticsAsTable
ZonalStatisticsAsTable(EZG_All_Corr_NoOverlap1_GAB, "Value", MuM,
MuM_ZonalStatisticsAsTable_Overlap1_10m)
ZonalStatisticsAsTable(EZG_All_Corr_NoOverlap2_GAB, "Value", MuM,
MuM_ZonalStatisticsAsTable_Overlap2_10m)
"print Zonal Statistic is done"

# Check out Spatial Extension
arcpy.CheckInExtension("Spatial")
```

3.20 ExplanatoryVariable_Disposalsite_2004_percentage.py

```
# ---------------------------------------------------------------------------------------------------------------------
# ExplanatoryVariable_disposalsite_2004_percentage.py
# Description: Obtains explanatory variable disposalsite_2004_percentage
# ---------------------------------------------------------------------------------------------------------------------

# Import arcpy module
import arcpy
import os.path

# Check in Spatial Extension
if arcpy.CheckExtension("Spatial") == "Available":
    arcpy.CheckOutExtension("Spatial")

# Path
Path = "..."

# Set workspace
arcpy.env.workspace = Path

# Allow to overwrite output
arcpy.gp.overwriteOutput = True

# Raster Environment Settings
# Setting Snap is MANDATORY!!!
arcpy.env.extent = arcpy.Extent(485859, 75238, 834100, 295918) #
http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#//001w00000009000000

# Define Variables: Input
EZG_All_Corr_NoOverlap1_GAB = ".../EZG_All_Corr_NoOverlap1_10m.img"
EZG_All_Corr_NoOverlap2_GAB = ".../Data_out/EZG_All_Corr_NoOverlap2_10m.img"
```

```
Deponie = ".../Deponie_Eawag_2015/AS97R_721.tif"
print "Input Variables are defined"

# Define Variables: Output
Data_out = Path + "Data_out"
Deponie_TabulateAreaTable_Overlap1_10m = Data_out +
".../Deponie_TabulateAreaTable_Overlap1_10m.txt"
Deponie_TabulateAreaTable_Overlap2_10m = Data_out +
".../Deponie_TabulateAreaTable_Overlap2_10m_Test.txt"
print "Output Variables are defined"

# Make sure that the fallowing Raster datasets are alligned to Geologie Raster dataset
# Setting Snap is MANDATORY and will lead to mistakes if it is not done
Snap = Deponie
arcpy.env.snapRaster = Snap #
http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#//001w0000000m000000
print "Snap is defined"

# Tabulate Area: Output table that will contain the summary of the area of each class in each zone.
arcpy.sa.TabulateArea(EZG_All_Corr_NoOverlap1_GAB, "Value", Deponie,"Value",
Deponie_TabulateAreaTable_Overlap1_10m, 10)
arcpy.sa.TabulateArea(EZG_All_Corr_NoOverlap2_GAB, "Value", Deponie,"Value",
Deponie_TabulateAreaTable_Overlap2_10m, 10)
print "Tabulate Area is done"

# Check out Spatial Extension
arcpy.CheckInExtension("Spatial")
```

3.21 ExplanatoryVariable_AE2km2_Area.py

```
# ---------------------------------------------------------------------------
# ExplanatoryVariable_AE2km2_Area.py
# Description: Calculates surface area of AE_2km2
# ---------------------------------------------------------------------------

# Import arcpy module
import arcpy

# Path
Path = "..."

# Set workspace
arcpy.env.workspace = Path

# Allow to overwrite output
arcpy.gp.overwriteOutput = True

# Define Variables: Input
AE2km2 =  Path + ".../basisgeometrie.shp"
print "Input Variables are defined"

# Calculate Areas
arcpy.AddField_management(AE2km2, "Area_M2", "DOUBLE")
arcpy.CalculateField_management(AE2km2, "Area_M2", "!SHAPE.AREA@SQUAREMETERS!" ,
"PYTHON_9.3")
print "Area is calculated"
```

3.22 JoinEVTablesToBDMSampingArea.py

```
# ---------------------------------------------------------------------------
# JoinEVTablesToBDMSampingArea.py
# Description: Joins explanatory variables to BDM sampling areas (explanatory variables are joined to
nationwide prediction sampling area in the same manner)
# ---------------------------------------------------------------------------

# Clear R's memory
rm(list=ls())

#options
#options(StringAsFactors = F)

# Import Data
?read.csv
EV_GAB <- read.csv(".../EV_GAB.csv", header=T, sep=",")
EV_FGT <- read.csv("...EV_FGT.csv", header=T, sep=",")
EV_MaxAbfluss <- read.csv(.../EV_MaxAbfluss.csv", header=T, sep=",")
EV_ARA_MW <- read.csv(.../EV_GAB_ChecklfBDMPointUpstream.csv", header=T, sep=",")
EV_Dam <- read.csv(".../EV_Dam_ChecklfBDMPointUpstream.csv", header=T, sep=",")
EV_Wasserkraft <- read.csv(".../EV_Wasserkraft_ChecklfBDMPointUpstream.csv", header=T, sep=",")
EV_Kanal <-
read.csv("G:/MASTERTHESIS_20150324/Projekt/4_ExplanatoryVariable/4_Kanal/2_ReaedCSVAndCalcul
ate/Data_out/EV_Kanal.csv", header=T, sep=",")
EV_Geolo….EV_Geologie.csv", header=T, sep="\t")
EV_HydrografischeGliederung <- read.csv(".../HydrografischeGliederung_WithValuesOutisdeCH.csv",
header=T, sep=",")
EV_BioggRegionen <- read.csv(".../BioggRegionen_WithValuesOutisdeCH.csv", header=T, sep=",")
EV_AueMoore <- read.csv(".../EV_AueMoore.csv", header=T, sep=",")
EV_BDMPoints <- read.csv(".../BDMPoints2.csv", header=T, sep=",")
EV_Waldmischungsgrad <- read.csv("..../EV_Waldmischungsgrad_10m.csv", header=T, sep="\t")
EV_MuM <- read.csv(".../MuM_ZonalStatisticsAsTable_10m.csv", header=T, sep="\t")
EV_Deponie <- read.csv(".../3_ChecklfBDMUpstram/EV_Deponie_ChecklfBDMPointUpstream.csv",
header=T, sep=",")

# Check Data
str(EV_GAB)
dim(EV_GAB)
str(EV_FGT)
dim(EV_FGT)
str(EV_MaxAbfluss)
dim(EV_MaxAbfluss)
str(EV_Dam)
dim(EV_Dam)
str(EV_Wasserkraft)
dim(EV_Wasserkraft)
str(EV_Kanal)
dim(EV_Kanal)
str(EV_Geologie)
dim(EV_Geologie)
str(EV_HydrografischeGliederung)
dim(EV_HydrografischeGliederung)
str(EV_BioggRegionen)
```

```
dim(EV_BioggRegionen)
str(EV_AueMoore)
dim(EV_AueMoore)
str(EV_BDMPoints, list.len = 129)
dim(EV_BDMPoints)
str(EV_Waldmischungsgrad)
dim(EV_Waldmischungsgrad)
str(EV_MuM)
dim(EV_MuM)
str(EV_Deponie)
dim(EV_Deponie)

# Join data frames by BDM_RowID
# http:/stackoverflow.com/questions/24191497/left-join-only-selected-columns-in-r-with-the-merge-function
# http:/www.statmethods.net/management/merging.html
# http:/stackoverflow.com/questions/1299871/how-to-join-data-frames-in-r-inner-outer-left-right
# http:/www.princeton.edu/~otorres/Merge101R.pdf
Join1 <- merge(x = EV_GAB, y = EV_FGT[ , c("BDM_RowID", "mqn_Jahr_MEAN", "abflussvar_MEAN",
"slope_MEAN", "slope_MAX")], by= "BDM_RowID", all.x = T) # Join 4 Variables
dim(Join1)
Join2 <- merge(x = Join1, y = EV_MaxAbfluss, by= "BDM_RowID", all = T) # Join 1 Variables
dim(Join2)
Join3 <- merge(x = Join2, y = EV_Dam[ , c("BDM_RowID", "AnzDam_Upstream")], by= "BDM_RowID", all.x
= T) # Join 1 Variables
dim(Join3)
Join4 <- merge(x = Join3, y = EV_Wasserkraft [, c("BDM_RowID", "AnzWasserkraft_Upstream")], by=
"BDM_RowID", all.x = T) # Join 1 Variables
dim(Join4)
Join5 <- merge(x = Join4, y = EV_Kanal, by= "BDM_RowID", all = T) # Join 1 Variables
dim(Join5)
Join6 <- merge(x = Join5, y = EV_Geologie[ , c("BDM_RowID", "GeologyRatio")], by= "BDM_RowID", all.x
= T) # Join 1 Variables
dim(Join6)
Join7 <- merge(x = Join6, y = EV_HydrografischeGliederung[ , c("BDM_RowID", "FLUSSGEB_N")], by=
"BDM_RowID", all.x = T) # Join 1 Variables
dim(Join7)
Join8 <- merge(x = Join7, y = EV_BioggRegionen[ , c("BDM_RowID", "BIOGREG_R6")], by=
"BDM_RowID", all.x = T) # Join 1 Variables
dim(Join8)
Join9 <- merge(x = Join8, y = EV_AueMoore, by= "BDM_RowID", all = T) # Join 1 Variables
dim(Join9)
Join10 <- merge(x = Join9, y = EV_BDMPoints[ , c("BDM_RowID", "GAB_A_EZG", "GAB_FLSTRE")], by=
"BDM_RowID", all.x = T) # Join 2 Variables
dim(Join10)
str(Join10)
Join11 <- merge(x = Join10, y = EV_Waldmischungsgrad[ , c("VALUE", "Nadelwaldanteil")], by.x =
"BDM_RowID" , by.y= "VALUE", all.x = T)  # Join 2 Variables
dim(Join11)
Join12 <- merge(x = Join11, y = EV_MuM[ , c("Value", "MEAN")], by.x = "BDM_RowID" , by.y= "Value", all.x
= T)  # Join 2 Variables
dim(Join12)
Join_13 <- merge(x = Join12, y = EV_ARA_MW[ , c("BDM_RowID", "ARA_ANT_Upstream",
"MW_ANT_Upstream")], by= "BDM_RowID", all.x = T) # Join 2 Variables
```

```
dim(Join_13)
Join_Final <- merge(x = Join_13, y = EV_Deponie[ , c("VALUE", "VALUE_27_Upstream")], by.x =
"BDM_RowID" , by.y= "VALUE", all.x = T) # Join 1 Variables
dim(Join_Final)
str(Join_Final)

# Divide canal length through watercourse_bdm_m  length and delete "LenKanal" column
# http:/www.cookbook-r.com/Manipulating_data/Adding_and_removing_columns_from_a_data_frame/
str(Join_Final)
Join_Final[c("canal_percentage")] <- (Join_Final$LenKanal) / (Join_Final$FLSTR_SUBE_ANT)* 100  # Da
die Zahlen Faktoren sind, kann man keine Mathe mit ihnen machen; as.numeric wandelt die Faktorzahlen
zu Nummerzahlen um
Join_Final$LenKanal   <- NULL

# Divide Deponie area through BDM catchment Area and delete "VALUE_27_Upstream" column
# http:/www.cookbook-r.com/Manipulating_data/Adding_and_removing_columns_from_a_data_frame/
str(Join_Final)
Join_Final[c("disposalsite_2004_percentage")] <- (Join_Final$VALUE_27_Upstream) /
(Join_Final$A_TEZG) * 100 # Da die Zahlen Faktoren sind, kann man keine Mathe mit ihnen machen;
as.numeric wandelt die Faktorzahlen zu Nummerzahlen um
Join_Final$VALUE_27_Upstream   <- NULL

# Delete colmns ARA and MW from GAB
Join_Final$ARA_ANT    <- NULL
Join_Final$MW_ANT     <- NULL

# Find and replace 0 --> NA in fallowing columns
Replace_0_NA <- c("mqn_Jahr_MEAN", "abflussvar_MEAN", "MaxAbfluss")
str(Join_Final[,  Replace_0_NA])
Join_Final[, Replace_0_NA][Join_Final[, Replace_0_NA] == 0] <- NA
str(Join_Final[, Replace_0_NA])

# Find and replace NA --> 0 in fallowing columns
Replace_NA_0 <- c("AnzDam_Upstream", "AnzWasserkraft_Upstream", "ARA_ANT_Upstream",
"MW_ANT_Upstream", "Nadelwaldanteil", "canal_percentage", "disposalsite_2004_percentage",
"AueMoore")
str(Join_Final[,  Replace_NA_0])
Join_Final[, Replace_NA_0][is.na(Join_Final[Replace_NA_0])] <- 0
str(Join_Final[, Replace_NA_0])


# Rename columns
# http:/www.cookbook-r.com/Manipulating_data/Renaming_columns_in_a_data_frame/
names(Join_Final)[names(Join_Final)=="A_TEZG"] <- "area_bdm_m2"
names(Join_Final)[names(Join_Final)=="FLSTR_SUBE_ANT"] <- "watercourse_bdm_m"
names(Join_Final)[names(Join_Final)=="ACK_ANT"] <- "field_percentage"
names(Join_Final)[names(Join_Final)=="HUEF_ANT"] <- "legume_percentage"
names(Join_Final)[names(Join_Final)=="KART_ANT"] <- "potato_percentage"
names(Join_Final)[names(Join_Final)=="GETR_ANT"] <- "cereal_percentage"
names(Join_Final)[names(Join_Final)=="MAIS_ANT"] <- "corn_percentage"
names(Join_Final)[names(Join_Final)=="OBST_ANT"] <- "fruit_percentage"
names(Join_Final)[names(Join_Final)=="RAPS_ANT"] <- "rapeseed_percentage"
names(Join_Final)[names(Join_Final)=="RUEB_ANT"] <- "rootvegetable_percentage"
```

```
names(Join_Final)[names(Join_Final)=="GEM_ANT"] <- "vegetable_percentage"
names(Join_Final)[names(Join_Final)=="REB_ANT"] <- "vine_percentage"
names(Join_Final)[names(Join_Final)=="WALD_ANT"] <- "forest_percentage"
names(Join_Final)[names(Join_Final)=="GRUE_ANT"] <- "green_percentage"
names(Join_Final)[names(Join_Final)=="FASS_ANT"] <- "facade_percentage"
names(Join_Final)[names(Join_Final)=="DACH_ANT"] <- "roof_percentage"
names(Join_Final)[names(Join_Final)=="FA_DACH_ANT"] <- "facaderoof_percentage"
names(Join_Final)[names(Join_Final)=="SIED_ANT"] <- "settlement_percentage"
names(Join_Final)[names(Join_Final)=="DEPO_ANT"] <- "disposalsite_190207_percentage"
names(Join_Final)[names(Join_Final)=="ARA_ANT_Upstream"] <- "wastewater_m3.a"
names(Join_Final)[names(Join_Final)=="MW_ANT_Upstream"] <- "stormsewage_m3.a"
names(Join_Final)[names(Join_Final)=="GEL_A_ANT"] <- "track_percentage"
names(Join_Final)[names(Join_Final)=="STR_A_ANT"] <- "street_percentage"
names(Join_Final)[names(Join_Final)=="mqn_Jahr_MEAN"] <- "Q_amean_m3.s"
names(Join_Final)[names(Join_Final)=="abflussvar_MEAN"] <- "Qvar_amean_m3.s"
names(Join_Final)[names(Join_Final)=="slope_MEAN"] <- "slope_mean"
names(Join_Final)[names(Join_Final)=="slope_MAX"] <- "slope_max"
names(Join_Final)[names(Join_Final)=="MaxAbfluss"] <- "Q_amax_m3.s"
names(Join_Final)[names(Join_Final)=="AnzDam_Upstream"] <- "dam_count"
names(Join_Final)[names(Join_Final)=="AnzWasserkraft_Upstream"] <- "hydropower_count"
names(Join_Final)[names(Join_Final)=="GeologyRatio"] <- "carbonate_per_carbonatesilicate"
names(Join_Final)[names(Join_Final)=="FLUSSGEB_N"] <- "hydro_class"
names(Join_Final)[names(Join_Final)=="BIOGREG_R6"] <- "biogeo_class"
names(Join_Final)[names(Join_Final)=="AueMoore"] <- "floodplainwetland_percentage"
names(Join_Final)[names(Join_Final)=="GAB_A_EZG"] <- "area_total_m2"
names(Join_Final)[names(Join_Final)=="GAB_FLSTRE"] <- "watercourse_total_m"
names(Join_Final)[names(Join_Final)=="Nadelwaldanteil"] <- "decidious_per_forest"
names(Join_Final)[names(Join_Final)=="MEAN"] <- "Masl"
str(Join_Final)

names(Join_Final)
str(Join_Final)
Join_Final$area_total_m2 <- as.numeric(Join_Final$area_total_m2)
str(Join_Final)

# Write data to csv file
write.table(Join_Final,".../EV_BDMPoints.csv", sep="\t", quote = F, row.names = F)
# quote = F verhindert das Nummern (Integer/Float) zu Text (String) umgewandelt werden; R kann nicht
zwischen Nummern und Text unterscheiden
# row.names = F verhindert das eine neue ID Spalte eingefügt wird

# How many values does each column have? Check if it there are sufficient values for the model
Join_Final_NotNA <- colSums(!is.na(Join_Final))
Join_Final_NotZero <-colSums(Join_Final != 0)
Join_Final_NotNA
Join_Final_NotZero
```

4. GLM Preparation

4.1 GLM_Preparation_ExplanatoryVariables.R

```
# ------------------------------------------------------------------------------------------------------------
# 0_GLM_Preparation_ExplanatoryVariables. R
# Description:  Checks if the explanatory variables correlate
#  ------------------------------------------------------------------------------------------------------------

# Clear R's memory
##############################################################################
rm(list=ls())

# Import Data
##############################################################################
EV <- read.csv(".../EV_BDMPoints.csv", header=T, sep="\t", stringsAsFactors=FALSE)
EV_BDMPoints <- read.csv(".../BDMPoints.csv", header=T, sep="\t")
str(EV)
str(EV_BDMPoints)
dim(EV)
dim(EV_BDMPoints)
EV_numeric <- EV[sapply(EV,is.numeric)]
dim(EV_numeric)
names(EV_numeric)

#Plot
##############################################################################
options(scipen=10000) # Force R to stop plotting abbreviated axis labels: e.g. 1e+00

# Create EV_numeric-EV_numeric Plots
##############################################################################
names(EV_numeric)
pdf(".../EV_EV_Plot.pdf", width = 5, height = 5)
k <-1
for(i in 1:39){
  for(j in k:39){
plot(EV_numeric[,i], EV_numeric[,j], xlab = names(EV_numeric)[i], ylab = names(EV_numeric)[j])
corr <- cor(as.numeric(EV_numeric[,i]), as.numeric(EV_numeric[,j]), use = "pairwise.complete.obs", method
= "spearman")
fstat <-  summary(lm(EV_numeric[,j] ~ EV_numeric[,i]))$fstatistic
pvalue <- 1-pf(fstat[1],fstat[2],fstat[3])
legend("topright", paste("pvalue = ", round(pvalue,4), # auf 4 Stellen gerundet
                         "corr = ", round(corr,4)))}
k<-k+1}
dev.off()

# Create EV-EV Correlation Table spearman
##############################################################################
# http://thomasleeper.com/Rcourse/Tutorials/NAhandling.html
dim(EV)
str(EV)
EV_numeric <- EV[sapply(EV,is.numeric)]
dim(EV_numeric)
EV_cor <- cor(EV_numeric[,2:39],EV_numeric[,2:39], use = "pairwise.complete.obs", method =
```

```
"spearman")
write.table(EV_cor,".../EV_EV_CorrTable_spearman.csv", sep="\t", quote = F, row.names = F)
EV_cor[ EV_cor < -0.7 ] <- "*"
EV_cor[ EV_cor > 0.7 ] <- "*"
write.table(EV_cor,".../EV_EV_CorrTable_grkl0.7_spearman.csv", sep="\t", quote = F, row.names = F)

# Create EV-EV Correlation Table kendall
################################################################################
# http://thomasleeper.com/Rcourse/Tutorials/NAhandling.html
EV_numeric <- EV[sapply(EV,is.numeric)]
EV_cor_kendall <- cor(EV_numeric[,2:39],EV_numeric[,2:39], use = "pairwise.complete.obs", method =
"kendall")
write.table(EV_cor_kendall,".../EV_EV_CorrTable_kendall.csv", sep="\t", quote = F, row.names = F)
EV_cor_kendall[ EV_cor_kendall < -0.7 ] <- "*"
EV_cor_kendall[ EV_cor_kendall > 0.7 ] <- "*"
write.table(EV_cor_kendall,".../EV_EV_CorrTable_grkl0.7_kendall.csv", sep="\t", quote = F, row.names =
F)
################################################################################
# Create EV-EV Correlation Table pearson
# http://thomasleeper.com/Rcourse/Tutorials/NAhandling.html
EV_numeric <- EV[sapply(EV,is.numeric)]
EV_cor_pearson <- cor(EV_numeric[,2:39],EV_numeric[,2:39], use = "pairwise.complete.obs", method =
"pearson")
write.table(EV_cor_pearson,".../EV_EV_CorrTable_pearson.csv", sep="\t", quote = F, row.names = F)
EV_cor_pearson[ EV_cor_pearson < -0.7 ] <- "*"
EV_cor_pearson[ EV_cor_pearson > 0.7 ] <- "*"
write.table(EV_cor_pearson,".../EV_EV_CorrTable_grkl0.7_pearson.csv", sep="\t", quote = F, row.names =
F)
```

4.2 GLM_Preparation_EPT.R

```
# ------------------------------------------------------------------------------------------------------------
# 0_GLM_Preparation_EPT. R
# Description: Relates the the explanatory variables to the EPT species richness
# ------------------------------------------------------------------------------------------------------------

# Clear R's memory
################################################################################
rm(list=ls())

# Import Data
################################################################################
EV <- read.csv(".../EV_BDMPoints.csv", header=T, sep="\t", stringsAsFactors=FALSE)
EV_BDMPoints <- read.csv(".../BDMPoints.csv", header=T, sep="\t")
str(EV)
str(EV_BDMPoints)
dim(EV)
dim(EV_BDMPoints)
# Join BDM Point EPT and IBCH Datato EV
RV <- merge(x = EV, y = EV_BDMPoints[ , c("BDM_RowID", "BDM_a_EPT", "BDM_a_IBCH")], by=
"BDM_RowID", all.x = T) # Join 2 Variables
str(RV)
dim(RV)
names(RV)
################################################################################
# Order RV columns alopabetically and select only numeric formatted
columns
RV_order <- RV[,order(names(RV))]
str(RV_order)
dim(RV_order)
names(RV_order)
RV_order_numeric <- RV_order[sapply(RV_order,is.numeric)]
dim(RV_order_numeric)
names(RV_order_numeric)

#Plot
################################################################################
options(scipen=10000) # Force R to stop plotting abbreviated axis labels: e.g. 1e+00

# Create EV-EPT Plot
################################################################################
# Line
names(RV_order_numeric)
pdf(".../EV_EPT_Plot_Line_EPT.pdf", width = 5, height = 5)
for(j in 1:41){
  plot(RV_order_numeric[,j], RV_order_numeric[,3], xlab = names(RV_order_numeric)[j], ylab =
names(RV_order_numeric)[3])
  if(!is.factor(RV_order_numeric[,3])){abline(equation <- lm(RV_order_numeric[,3] ~ RV_order_numeric[,j]),
col="red")} # creates line; but this line can be misleading, human eye can estimate trend much wiser
without line because outliers can have a big influence
  #legend("topright",paste0("Slope = ", round(coef(equation)[2])))
  #rsqare <- summary(lm(RV_order_numeric[,44] ~ RV_order_numeric[,j]))$r.squared
  corrS <- cor(as.numeric(RV_order_numeric[,3]), as.numeric(RV_order_numeric[,j]), use =
"pairwise.complete.obs", method = "spearman")
  corrK <- cor(as.numeric(RV_order_numeric[,3]), as.numeric(RV_order_numeric[,j]), use =
"pairwise.complete.obs", method = "kendall")
  fstat <-  summary(lm(RV_order_numeric[,3] ~ RV_order_numeric[,j]))$fstatistic
  pvalue <- 1-pf(fstat[1],fstat[2],fstat[3])
  legend("topright", paste("Kendall correlation:      ", round(corrS,4),
                           "\nSpearman correlation: ", round(corrK,4), "\n"), cex=0.5)}
dev.off()
###################################
# No Line
names(RV_order_numeric)
pdf(/EV_EPT_Plot_NoLine_EPT.pdf", width = 5, height = 5)
for(j in 1:41){
  plot(RV_order_numeric[,j], RV_order_numeric[,3], xlab = names(RV_order_numeric)[j], ylab =
names(RV_order_numeric)[3])
  #if(!is.factor(RV_order_numeric[,3])){abline(equation <- lm(RV_order_numeric[,44] ~
RV_order_numeric[,j]), col="red")} # creates line; but this line can be misleading, human eye can estimate
trend much wiser without line because outliers can have a big influence
  #legend("topright",paste0("Slope = ", round(coef(equation)[2])))
  #rsqare <- summary(lm(RV_order_numeric[,44] ~ RV_order_numeric[,j]))$r.squared
  corrS <- cor(as.numeric(RV_order_numeric[,3]), as.numeric(RV_order_numeric[,j]), use =
"pairwise.complete.obs", method = "spearman")
  corrK <- cor(as.numeric(RV_order_numeric[,3]), as.numeric(RV_order_numeric[,j]), use =
```

```
"pairwise.complete.obs", method = "kendall")
  fstat <-  summary(lm(RV_order_numeric[,3] ~ RV_order_numeric[,j]))$fstatistic
  pvalue <- 1-pf(fstat[1],fstat[2],fstat[3])
  legend("topright", paste("Kendall correlation:      ", round(corrS,4),
                           "\nSpearman correlation: ", round(corrK,4), "\n"), cex=0.5)}
dev.off()
```

4.3 Tree_EPT.R

```
# ------------------------------------------------------------------------------------------------------------------------
# Tree_EPT.R
# Description: Carries out the tree models which help to decide which explanatory variables to choose
# ------------------------------------------------------------------------------------------------------------------------

## Clear R's memory
##############################################################################
rm(list=ls())

 # Import library
##############################################################################
library(tree)

## Import Data
##############################################################################
EV <- read.csv(".../EV_BDMPoints.csv", header=T, sep="\t")
EV_BDMPoints <- read.csv(".../BDMPoints.csv", header=T, sep="\t")
str(EV)
str(EV_BDMPoints)


 ## Join BDM Point EPT and IBCH Datato EV
##############################################################################
RV <- merge(x = EV, y = EV_BDMPoints[ , c("BDM_RowID", "BDM_a_EPT", "BDM_a_IBCH")], by=
"BDM_RowID", all.x = T) # Join 2 Variables
str(RV)
names(RV)

## Tree of correlated variable
##############################################################################
# na.action
# A function to filter missing data from the model frame. The default is na.pass (to do nothing)
# as tree handles missing values (by dropping them down the tree as far as possible).
# tree1
tree1<-tree(BDM_a_EPT~ field_percentage + legume_percentage + potato_percentage +
cereal_percentage +
          corn_percentage + rapeseed_percentage + rootvegetable_percentage + vegetable_percentage,
          data=RV)
plot(tree1)
text(tree1)
print(tree1)
####################################
#tree2
tree2<-tree(BDM_a_EPT~ area_bdm_m2 + watercourse_bdm_m + area_total_m2 + watercourse_total_m,
          data=RV)
plot(tree2)
text(tree2)
print(tree2)
####################################
#tree3
tree3<-tree(BDM_a_EPT~ facade_percentage + roof_percentage + facaderoof_percentage +
settlement_percentage,
          data=RV)
plot(tree3)
text(tree3)
print(tree3)
####################################
## Not Corrlation Variables, NA included
#tree4
tree4<-tree(BDM_a_EPT~ Q_amax_m3.s + Q_amean_m3.s + Qvar_amean_m3.s + fruit_percentage +
vine_percentage + forest_percentage + green_percentage + decidious_per_forest + track_percentage +
street_percentage + canal_percentage + dam_count + hydropower_count + slope_mean + slope_max +
Masl +carbonate_per_carbonatesilicate + floodplainwetland_percentage,
          data=RV)
plot(tree4)
text(tree4)
print(tree4)
####################################
## Correlating and Not Correlating Variables, NA included
#tree5
tree5<-tree(BDM_a_EPT~ Q_amax_m3.s + Q_amean_m3.s + Qvar_amean_m3.s + fruit_percentage +
vine_percentage + forest_percentage + green_percentage + decidious_per_forest + track_percentage +
street_percentage + canal_percentage + dam_count + hydropower_count + slope_mean + slope_max +
Masl +carbonate_per_carbonatesilicate + floodplainwetland_percentage +
          watercourse_bdm_m + roof_percentage + disposalsite_2004_percentage + wastewater_m3.a +
corn_percentage + vegetable_percentage,
          data=RV)
plot(tree5)
text(tree5)
print(tree5)
####################################
## slope
#tree6
tree6<-tree(BDM_a_EPT~ slope_mean + slope_max,
          data=RV)
plot(tree6)
text(tree6)
print(tree6)
####################################
## Not Corrlation Variables, NA excluded
#tree8
tree8<-tree(BDM_a_EPT~ fruit_percentage + vine_percentage + forest_percentage + green_percentage +
decidious_per_forest + track_percentage + street_percentage + canal_percentage + dam_count +
hydropower_count + slope_mean + slope_max + Masl +carbonate_per_carbonatesilicate +
floodplainwetland_percentage,
          data=RV)
plot(tree8)
```

```
text(tree8)
print(tree8)
###################################
## Correlating and Not Correlating Variables, NA excluded
#tree9
tree9<-tree(BDM_a_EPT~ fruit_percentage + vine_percentage + forest_percentage + green_percentage +
decidious_per_forest + track_percentage + street_percentage + canal_percentage + dam_count +
hydropower_count + slope_mean + slope_max + Masl +carbonate_per_carbonatesilicate +
floodplainwetland_percentage +
        watercourse_bdm_m + roof_percentage +  corn_percentage + vegetable_percentage +
disposalsite_2004_percentage + wastewater_m3.a,
        data=RV)
plot(tree9)
text(tree9)
print(tree9)
###################################
## Only correlating variables
tree10<-tree(BDM_a_EPT~ watercourse_bdm_m + roof_percentage + disposalsite_2004_percentage +
wastewater_m3.a + corn_percentage + vegetable_percentage,
         data=RV)
plot(tree10)
text(tree10)
print(tree10)
```

4.4 Shapirotest_EV.R

```
# ------------------------------------------------------------------------------------------------------------
# Shapirotest_EV.R
# Description: Checks if the explanatory variables are normally distributed
#  ------------------------------------------------------------------------------------------------------------

## Clear R's memory
##############################################################################
##rm(list=ls())

## Import Data
##############################################################################
##
EV <- read.csv(".../EV_BDMPoints.csv", header=T, sep="\t", stringsAsFactors=FALSE)
dim(EV)
str(EV)
EV_numeric <- EV[sapply(EV,is.numeric)]

## Test for Normality
##############################################################################
##
# http://www.dummies.com/how-to/content/how-to-test-data-normality-in-a-formal-way-in-r.html
# http://stackoverflow.com/questions/18963335/looping-through-columns-of-csv-file-in-rv
## Applying shapiro.test function
##############################################################################
##
shapirotest <- apply(EV_numeric, 2, shapiro.test)
# Showing results in a nice format
shapirotest_niceformat <-sapply(shapirotest, function(x) unlist(x[c( "statistic", "p.value")]))
write.table(shapirotest_niceformat,".../EV_Shapirotest.csv", sep="\t", quote = F, row.names = F)
# Check if test for normality makes sense with help of histograms
 hist(EV[,2])
```

5. GLM

5.1 GLM_Family.R

```
#------------------------------------------------------------------------------------------------------------
# GLM_Family.R
# Description: Evaluates which GLM family  fits the data best
#------------------------------------------------------------------------------------------------------------

## Clear R's memory
##############################################################################
rm(list=ls()

## Import Data
##############################################################################
EV <- read.csv(".../EV_BDMPoints.csv", header=T, sep="\t")
EV_BDMPoints <- read.csv("...BDMPoints.csv", header=T, sep="\t")
str(EV)
str(EV_BDMPoints

## Join BDM Point EPT and IBCH Data to EV
##############################################################################
RV <- merge(x = EV, y = EV_BDMPoints[ , c("BDM_RowID", "BDM_a_EPT", "BDM_a_IBCH")], by=
"BDM_RowID", all.x = T) # Join 2 Variables
str(RV)
names(RV)

## simple GLM
##############################################################################
# glm_simple_poisson
glm_simple_poisson <- glm(BDM_a_EPT~ forest_percentage + green_percentage  + decidious_per_forest
+ street_percentage + slope_mean  + Masl + carbonate_per_carbonatesilicate + watercourse_bdm_m +
roof_percentage + wastewater_m3.a + corn_percentage, data=RV, family = poisson)
summary(glm_simple_poisson)
# Residual deviance: 277.17  on 134  degrees of freedom: Overdispersion
# Common main title of a figure panel compiled with par(mfrow)
#http://stackoverflow.com/questions/14660372/common-main-title-of-a-figure-panel-compiled-with-parmfro
w
```

```
par(mfrow = c(2,2))
plot(glm_simple_poisson)
mtext("family = poisson, no interaction", side = 3, line = -2, outer = TRUE)
# => Since Overdispersion is present quasipoisson distribution has to be used
####################################
# glm_simple_gaussian
glm_simple_gaussian <- glm(BDM_a_EPT~ forest_percentage + green_percentage  +
decidious_per_forest  + street_percentage + slope_mean  + Masl + carbonate_per_carbonatesilicate +
watercourse_bdm_m + roof_percentage + wastewater_m3.a + corn_percentage, data=RV, family =
gaussian)
summary(glm_simple_gaussian)
par(mfrow = c(2,2))
plot(glm_simple_gaussian)
mtext("family = gaussian, no interaction", side = 3, line = -2, outer = TRUE)

## 2wayinteraction GLM
#################################################################################
# glm_2wayinteraction_poisson
glm_2wayinteraction_poisson <- glm(BDM_a_EPT~ (forest_percentage + green_percentage  + decidious_per_forest  +
street_percentage + slope_mean  + Masl + carbonate_per_carbonatesilicate +
 watercourse_bdm_m + roof_percentage + wastewater_m3.a + corn_percentage)^2, data=RV, family = poisson)
summary(glm_2wayinteraction_poisson)
par(mfrow = c(2,2))
plot(glm_2wayinteraction_poisson)
mtext("family = poisson", side = 3, line = -2, outer = TRUE)
####################################
# glm_2wayinteraction_gaussian
glm_2wayinteraction_gaussian <- glm(BDM_a_EPT~ (forest_percentage + green_percentage  + decidious_per_forest
+ street_percentage + slope_mean  + Masl + carbonate_per_carbonatesilicate +
watercourse_bdm_m + roof_percentage + wastewater_m3.a + corn_percentage)^2, data=RV, family = gaussian)
summary(glm_2wayinteraction_gaussian)
par(mfrow = c(2,2))
plot(glm_2wayinteraction_gaussian)
mtext("family = gaussian", side = 3, line = -2, outer = TRUE)
```

5.2 GLM_ModelSelection_Step.py

```
#-------------------------------------------------------------------------------------------------------------------
# GLM_ModelSelection_Step.py
# Description: Applies the model selection method “step” to the GLM
#-------------------------------------------------------------------------------------------------------------------

## Clear R's memory
#################################################################################
rm(list=ls())

##  Import library
#################################################################################
library(boot)
```

```
## Import Data
#################################################################################
EV <- read.csv(".../EV_BDMPoints.csv", header=T, sep="\t")
EV_BDMPoints <- read.csv(/BDMPoints.csv", header=T, sep="\t")
str(EV)
str(EV_BDMPoints)

## Join BDM Point EPT and IBCH Data to EV
#################################################################################
RV <- merge(x = EV, y = EV_BDMPoints[ , c("BDM_RowID", "BDM_a_EPT", "BDM_a_IBCH")], by=
"BDM_RowID", all.x = T) # Join 2 Variables
str(RV)
names(RV)

## glm_2wayinteraction_gaussian
#################################################################################
# glm_2wayinteraction_gaussian
glm_2wayinteraction_gaussian <- glm(BDM_a_EPT~ (forest_percentage + green_percentage  +
decidious_per_forest  + street_percentage + slope_mean  + Masl + carbonate_per_carbonatesilicate +
watercourse_bdm_m + roof_percentage + wastewater_m3.a + corn_percentage)^2, data=RV, family =
gaussian)
summary(step_glm_2wayinteraction_gaussian)
AIC(step_glm_2wayinteraction_gaussian)
# > AIC(step_glm_2wayinteraction_gaussian)
# [1] 2650.552

## mansimp glm_2wayinteraction_gaussian
#################################################################################
# mansimp1
glm_interaction2way_gaussian_step_mansimp1 <- update(step_glm_2wayinteraction_gaussian,~. -
green_percentage:Masl)
summary(glm_interaction2way_gaussian_step_mansimp1)
AIC(glm_interaction2way_gaussian_step_mansimp1)
 # > AIC(glm_interaction2way_gaussian_step_mansimp1)
# [1] 2651.777
####################################
# mansimp2
glm_interaction2way_gaussian_step_mansimp2 <-
update(glm_interaction2way_gaussian_step_mansimp1,~. -
carbonate_per_carbonatesilicate:roof_percentage)
summary(glm_interaction2way_gaussian_step_mansimp2)
AIC(glm_interaction2way_gaussian_step_mansimp2)
 # > AIC(glm_interaction2way_gaussian_step_mansimp2)
# [1] 2653.028
 ####################################
# mansimp3
glm_interaction2way_gaussian_step_mansimp3 <-
update(glm_interaction2way_gaussian_step_mansimp2,~. - Masl:carbonate_per_carbonatesilicate)
summary(glm_interaction2way_gaussian_step_mansimp3)
AIC(glm_interaction2way_gaussian_step_mansimp3)
 # > AIC(glm_interaction2way_gaussian_step_mansimp3)
# [1] 2654.779
 ####################################
```

```
# mansimp4
glm_interaction2way_gaussian_step_mansimp4 <-
update(glm_interaction2way_gaussian_step_mansimp3,~. - slope_mean:Masl )
summary(glm_interaction2way_gaussian_step_mansimp4)
AIC(glm_interaction2way_gaussian_step_mansimp4)
 # > AIC(glm_interaction2way_gaussian_step_mansimp4)
# [1] 2656.58
 ####################################
# mansimp5
glm_interaction2way_gaussian_step_mansimp5 <-
update(glm_interaction2way_gaussian_step_mansimp4,~. - Masl:roof_percentage)
summary(glm_interaction2way_gaussian_step_mansimp5)
AIC(glm_interaction2way_gaussian_step_mansimp5)
 # > AIC(glm_interaction2way_gaussian_step_mansimp5)
# [1] 2658.931
 ####################################
# mansimp6
glm_interaction2way_gaussian_step_mansimp6 <-
update(glm_interaction2way_gaussian_step_mansimp5,~. - Masl:corn_percentage )
summary(glm_interaction2way_gaussian_step_mansimp6)
AIC(glm_interaction2way_gaussian_step_mansimp6)
 # > AIC(glm_interaction2way_gaussian_step_mansimp6)
# [1] 2660.446
 ####################################
# mansimp7
glm_interaction2way_gaussian_step_mansimp7 <-
update(glm_interaction2way_gaussian_step_mansimp6,~. - green_percentage:corn_percentage )
summary(glm_interaction2way_gaussian_step_mansimp7)
AIC(glm_interaction2way_gaussian_step_mansimp7)
 # > AIC(glm_interaction2way_gaussian_step_mansimp7)
# [1]  2662.721
 ####################################
# mansimp8
glm_interaction2way_gaussian_step_mansimp8 <-
update(glm_interaction2way_gaussian_step_mansimp7,~. -green_percentage:street_percentage )
summary(glm_interaction2way_gaussian_step_mansimp8)
AIC(glm_interaction2way_gaussian_step_mansimp8)
 # > AIC(glm_interaction2way_gaussian_step_mansimp8)
# [1] 2662.645
 ####################################
# mansimp9
glm_interaction2way_gaussian_step_mansimp9 <-
update(glm_interaction2way_gaussian_step_mansimp8,~. -forest_percentage:Masl )
summary(glm_interaction2way_gaussian_step_mansimp9)
AIC(glm_interaction2way_gaussian_step_mansimp9)
 # > AIC(glm_interaction2way_gaussian_step_mansimp9)
# [1]  2663.958
 ####################################
# mansimp10
glm_interaction2way_gaussian_step_mansimp10 <-
update(glm_interaction2way_gaussian_step_mansimp9,~. -decidious_per_forest:street_percentage )
summary(glm_interaction2way_gaussian_step_mansimp10)
AIC(glm_interaction2way_gaussian_step_mansimp10)
```



```
 # > AIC(glm_interaction2way_gaussian_step_mansimp9)
# [1]  2666.674
 ####################################
# mansimp11
glm_interaction2way_gaussian_step_mansimp11 <-
update(glm_interaction2way_gaussian_step_mansimp10,~. -street_percentage:corn_percentage )
summary(glm_interaction2way_gaussian_step_mansimp11)
AIC(glm_interaction2way_gaussian_step_mansimp11)
 # > AIC(glm_interaction2way_gaussian_step_mansimp9)
# [1]  2671.133
 ####################################
# mansimp12
glm_interaction2way_gaussian_step_mansimp12 <-
update(glm_interaction2way_gaussian_step_mansimp11,~. -forest_percentage:street_percentage )
summary(glm_interaction2way_gaussian_step_mansimp12)
AIC(glm_interaction2way_gaussian_step_mansimp12)
 # > AIC(glm_interaction2way_gaussian_step_mansimp9)
# [1] 2671.961
 ####################################
# mansimp13
glm_interaction2way_gaussian_step_mansimp13 <-
update(glm_interaction2way_gaussian_step_mansimp12,~. -decidious_per_forest:slope_mean)
summary(glm_interaction2way_gaussian_step_mansimp13)
AIC(glm_interaction2way_gaussian_step_mansimp13)
# > AIC(glm_interaction2way_gaussian_step_mansimp9)
# [1]  2676.16

## Predict EV_AE2km2_NoNA
#############################################################################################
EV_AE2km2 <- read.csv("../EV_AE2km2.txt", header=T, sep="\t")
str(EV_AE2km2)
####################################
# Are NA present?
colSums(!is.na(EV_AE2km2))
# Create subset where no NA are present
dim(EV_AE2km2)
EV_AE2km2_NoNA <- subset(EV_AE2km2,!(is.na(EV_AE2km2["carbonate_per_carbonatesilicate"])))
colSums(!is.na(EV_AE2km2_NoNA))
str(EV_AE2km2_NoNA)
dim(EV_AE2km2_NoNA)
####################################
# Predict EV_AE2km2_NoNA
Step_Predict_AE <-predict(step_glm_2wayinteraction_gaussian, EV_AE2km2, type = "response",  se.fit=T,
na.action = na.omit)
EV_AE2km2_NoNA$Step_AE_fit_EPT <- round(Step_Predict_AE$fit,0) # Predictions, as for se.fit =
FALSE.
EV_AE2km2_NoNA$Step_AE_se_EPT <- Step_Predict_AE$se.fit # Estimated standard errors.
EV_AE2km2_NoNA$Step_AE_rs_EPT <- Step_Predict_AE$residual.scale # A scalar giving the square
root of the dispersion used in computing the standard errors.
names(EV_AE2km2_NoNA)
write.table(EV_AE2km2_NoNA,".../Step_Predict_AE_EPT.csv", sep="\t", quote = F, row.names = F)

## Predict BDM Points for Verification
```

```
################################################################################
Step_Predict_BDM <-predict(step_glm_2wayinteraction_gaussian, newdata = NULL, type = "response",
se.fit=T, na.action = na.omit)
EV_BDMPoints$Step_BDM_fit_EPT <- round(Step_Predict_BDM$fit,0) # Predictions, as for se.fit = FALSE.
EV_BDMPoints$Step_BDM_se_EPT <- Step_Predict_BDM$se.fit # Estimated standard errors.
EV_BDMPoints$Step_BDM_rs_EPT <- Step_Predict_BDM$residual.scale # A scalar giving the square root
of the dispersion used in computing the standard errors.
names(EV_BDMPoints)
write.table(EV_BDMPoints,".../Step_Predict_BDM_EPT.csv", sep="\t", quote = F, row.names = F)
```

5.3 GLM_ModelSelection_Lasso.py

```
#  ------------------------------------------------------------------------------------------------------------------
# GLM_ModelSelection_Lasso.py
# Description: Applies the model selection method “lasso” to the GLM
#  ------------------------------------------------------------------------------------------------------------------

## Clear R's memory
################################################################################
rm(list=ls())

##  Import library
################################################################################
library(glmnet)

## Import Data
################################################################################
EV <- read.csv(".../EV_BDMPoints.csv", header=T, sep="\t")
EV_BDMPoints <- read.csv(".../BDMPoints.csv", header=T, sep="\t")
str(EV)
str(EV_BDMPoints)

## Join BDM Point EPT and IBCH Datato EV
################################################################################
RV <- merge(x = EV, y = EV_BDMPoints[ , c("BDM_RowID", "BDM_a_EPT", "BDM_a_IBCH")], by=
"BDM_RowID", all.x = T) # Join 2 Variables
str(RV)
names(RV)

##  Create interactions between EV
################################################################################
#http://stackoverflow.com/questions/27580267/how-to-make-all-interactions-in-r-before-using-glmnet/27583
931#27583931
# Selected EV and RV
EVRV_BDMPoints_Selected <- RV[,c("BDM_a_EPT",
"forest_percentage","green_percentage","decidious_per_forest",
"street_percentage","slope_mean","Masl","carbonate_per_carbonatesilicate",
                               "watercourse_bdm_m","roof_percentage","wastewater_m3.a","corn_percentage")]
dim(EVRV_BDMPoints_Selected)
# First step: using .*. for all interactions
EVRV_BDMPoints_Selected_Interaction <- as.formula(BDM_a_EPT ~ .*.)
RV_BDMPoints_Selected <- EVRV_BDMPoints_Selected$BDM_a_EPT
str(RV_BDMPoints_Selected)
```

```
# Second step: using model.matrix to take advantage of EVRV_BDMPoints_Selected_Interaction
EV_BDMPoints_Selected <- model.matrix(EVRV_BDMPoints_Selected_Interaction,
EVRV_BDMPoints_Selected[1,])[,-1] # test wether it works with row 1
EV_BDMPoints_Selected
length(EV_BDMPoints_Selected) # 66 (=11*10/2+11)
EV_BDMPoints_Selected <- model.matrix(EVRV_BDMPoints_Selected_Interaction,
EVRV_BDMPoints_Selected)[,-1] # for all rows
dim(EV_BDMPoints_Selected)

##  glmnet Lasso
################################################################################
glmnet_BDMPoints_Lasso = glmnet(EV_BDMPoints_Selected, RV_BDMPoints_Selected, alpha=1)
####################################
# Check
plot(glmnet_BDMPoints_Lasso)
print(glmnet_BDMPoints_Lasso)
coef(glmnet_BDMPoints_Lasso)
dim(coef(glmnet_BDMPoints_Lasso)) # one for each predictor (66) plus an intercept (1) = 67
####################################
# 10 fold cross validation
cv_EPT <- cv.glmnet(EV_BDMPoints_Selected, RV_BDMPoints_Selected, alpha=1)
plot(cv_EPT)
# lambda.min = BestLambda_EPT
BestLambda_EPT <- cv_EPT$lambda.min
BestLambda_EPT # 0.06167972
#lambda.1s
BestLambda2_EPT<- cv_EPT$lambda.1s
BestLambda2_EPT # 0.4775636
# Coefficients
coef <- coef(glmnet_BDMPoints_Lasso,s=BestLambda_EPT)
library(boot)
cvglm_10 <- cv.glm(data = RV, glmfit = coef, K = 10)
cvglm_410 <- cv.glm(data = RV, glmfit = coef, K = 410)
cvglm_10$delta
cvglm_410$delta

## Predict EV_AE2km2_NoNA
################################################################################
EV_AE2km2 <- read.csv(".../EV_AE2km2.txt", header=T, sep="\t")
str(EV_AE2km2)
dim(EV_AE2km2)
# 27 caronate_per_carnonatesilicate NA
colSums(!is.na(EV_AE2km2))
# Create subset where no NA are present
dim(EV_AE2km2)
EV_AE2km2_NoNA <- subset(EV_AE2km2,!(is.na(EV_AE2km2["carbonate_per_carbonatesilicate"])))
colSums(!is.na(EV_AE2km2_NoNA))
str(EV_AE2km2_NoNA)
dim(EV_AE2km2_NoNA)
####################################
 # Create interactions between EV
EVRV_EV_AE2km2_NoNA_Selected <- EV_AE2km2_NoNA[,c("AE_2km2_ID",
"forest_percentage","green_percentage","decidious_per_forest",
```

```
"street_percentage","slope_mean","Masl","carbonate_per_carbonatesilicate",
"watercourse_bdm_m","roof_percentage","wastewater_m3.a","corn_percentage")]
dim(EVRV_EV_AE2km2_NoNA_Selected)
# First step: using .*. for all interactions
EVRV_AE2km2_NoNA_Selected_Interaction <- as.formula(AE_2km2_ID ~ .*.)
RV_AE2km2_NoNA_Selected <- EVRV_EV_AE2km2_NoNA_Selected$AE_2km2_ID
# Second step: using model.matrix to take advantage of f
EV_AE2km2_Selected <- model.matrix(EVRV_AE2km2_NoNA_Selected_Interaction,
EVRV_EV_AE2km2_NoNA_Selected[1,])[,-1] # test wether it works with row 1
EV_AE2km2_Selected
length(EV_AE2km2_Selected) # 66 (=11*10/2+11)
EV_AE2km2_Selected <- model.matrix(EVRV_AE2km2_NoNA_Selected_Interaction,
EVRV_EV_AE2km2_NoNA_Selected)[,-1]
dim(EV_AE2km2_Selected)
####################################
# Predict EV_AE2km2_NoNA
glmnet_AE2km2_Lasso = glmnet(EV_AE2km2_Selected, RV_AE2km2_NoNA_Selected, alpha=1)
Prediction_AE <- predict(glmnet_BDMPoints_Lasso, EV_AE2km2_Selected , s=BestLambda_EPT)
EV_AE2km2_NoNA$Lasso_AE_fit_EPT <- round(Prediction_AE, 0)
str(EV_AE2km2_NoNA)
names(EV_AE2km2_NoNA)
write.table(EV_AE2km2_NoNA,".../Lasso_Predict_AE_EPT.csv", sep="\t", quote = F, row.names = F)
## Predict BDM Points for Verification
####################################
Prediction_BDM <- predict(glmnet_BDMPoints_Lasso, newx = EV_BDMPoints_Selected ,
s=BestLambda_EPT)
EV_BDMPoints$Lasso_BDM_fit_EPT <- round(Prediction_BDM, 0)
str(EV_BDMPoints)## GLM Step
names(EV_BDMPoints)
write.table(EV_BDMPoints,".../Lasso_Predict_BDM_EPT.csv", sep="\t", quote = F, row.names = F)
```

6. Prediction

6.1 NationwidePrediction_Preparation_EPT.R

```
# ------------------------------------------------------------------------------------------------------------------------
# NationwidePrediction_Preparation_EPT.R
# Description: Prepares the EPT species richness prediction
# ------------------------------------------------------------------------------------------------------------------------

## Clear R's memory
##############################################################################
rm(list=ls())

## Import library
##############################################################################
library(epiR)

## Import Data
##############################################################################

Step_Predict_AE <- read.csv(".../Step_Predict_AE_EPT.csv", header=T, sep="\t")
Step_Predict_BDM <- read.csv(".../Step_Predict_BDM_EPT.csv", header=T, sep="\t")
Lasso_Predict_AE <- read.csv(".../Lasso_Predict_AE_EPT.csv", header=T, sep="\t")
Lasso_Predict_BDM <- read.csv("...Lasso_Predict_BDM_EPT.csv", header=T, sep="\t")
EV_BDMPoints <- read.csv(".../BDMPoints.csv", header=T, sep="\t")

str(Step_Predict_AE)
str(Step_Predict_BDM)
str(Lasso_Predict_AE)
str(Lasso_Predict_BDM)
str(EV_BDMPoints)

dim(Step_Predict_AE)
dim(Step_Predict_BDM)
dim(Lasso_Predict_AE)
dim(Lasso_Predict_BDM)
dim(EV_BDMPoints)

names(Step_Predict_AE)
names(Step_Predict_BDM)
names(Lasso_Predict_AE)
names(Lasso_Predict_BDM)
names(EV_BDMPoints)

## Join
##############################################################################

# Join AE Predictions
dim(Step_Predict_AE)
AE <- merge(x = Step_Predict_AE, y = Lasso_Predict_AE[ , c("AE_2km2_ID", "Lasso_AE_fit_EPT")], by =
"AE_2km2_ID", all.x = T) # Join 2 Variables
dim(AE)
names(AE)
AE2 <- AE[c("AE_2km2_ID", "Step_AE_fit_EPT", "Step_AE_se_EPT", "Step_AE_rs_EPT",
"Lasso_AE_fit_EPT")]
dim(AE2)
#write.table(AE,".../AE_Predict_EPT.csv", sep="\t", quote = F, row.names = F)

# Join GAB Predictions
dim(Step_Predict_BDM)
BDMPoints <- merge(x = Step_Predict_BDM, y = Lasso_Predict_BDM[ , c("BDM_RowID",
"Lasso_BDM_fit_EPT")], by = "BDM_RowID", all.x = T) # Join 2 Variables
dim(BDMPoints)
names(BDMPoints)
BDMPoints2 <- BDMPoints[c("BDM_RowID", "BDM_a_EPT", "BDM_a_IBCH", "Step_BDM_fit_EPT",
"Step_BDM_se_EPT", "Step_BDM_rs_EPT", "Lasso_BDM_fit_EPT")]
dim(BDMPoints2)
#write.table(BDMPoints2,"./BDMPoints_Predict_EPT.csv", sep="\t", quote = F, row.names = F)

## Range
##############################################################################

AE_Predict<- read.csv(".../AE_Predict_EPT.csv", header=T, sep="\t")
BDMPoints_Predict <- read.csv(".../BDMPoints_Predict_EPT.csv", header=T, sep="\t")
```

```
names(AE_Predict)
names(BDMPoints_Predict)

###############################################

## AE

#Step
range(AE_Predict$Step_AE_fit_EPT)
mean(AE_Predict$Step_AE_fit_EPT)
hist(AE_Predict$Step_AE_fit_EPT, xlim=c(-10,40), ylim=c(0,7000), breaks = 200, xlab = "Step prediction",
main = NULL)

# Lasso
range(AE_Predict$Lasso_AE_fit_EPT)
mean(AE_Predict$Lasso_AE_fit_EPT)
hist(AE_Predict$Lasso_AE_fit_EPT, xlim=c(-10,30), ylim=c(0,8000), breaks = 50, xlab = "Lasso prediction",
main = NULL)

###############################################

##BDM

#Reality
range(BDMPoints_Predict$BDM_a_EPT)
mean(BDMPoints_Predict$BDM_a_EPT)
hist(BDMPoints_Predict$BDM_a_EPT, xlim=c(0,40), ylim=c(0,100), xlab = "Monitoring", main = NULL,
breaks = 10)

# Step
range(BDMPoints_Predict$Step_BDM_fit_EPT)
mean(BDMPoints_Predict$Step_BDM_fit_EPT)
hist(BDMPoints_Predict$Step_BDM_fit_EPT, xlim=c(0,30), ylim=c(0,200), xlab = "Step prediction", main =
NULL, breaks = 7)

# Lasso
range(BDMPoints_Predict$Lasso_BDM_fit_EPT)
mean(BDMPoints_Predict$Lasso_BDM_fit_EPT)
hist(BDMPoints_Predict$Lasso_BDM_fit_EPT, xlim=c(0,30), ylim=c(0,200), xlab = "Lasso prediction", main
= NULL, breaks = 7)

## Plot
#############################################################################

# Step Prediction - Nationwide - AE_2km2 vs Step Prediction - Nationwide - AE_2km2
plot(AE_Predict$Step_AE_fit_EPT, AE_Predict$Lasso_AE_fit_EPT, xlab = "Step Prediction - Nationwide -
AE_2km2", ylab = "Lasso Prediction - Nationwide - AE_2km2")
abline(0,1)
plot(AE_Predict$Step_AE_fit_EPT, AE_Predict$Lasso_AE_fit_EPT, xlab = "Step Prediction - Nationwide -
AE_2km2", ylab = "Lasso Prediction - Nationwide - AE_2km2", xlim = c(0,35), ylim = c(0,35))
abline(0,1)
concordance_correlation_ztransform <- epi.ccc(as.vector(AE_Predict$Step_AE_fit_EPT),
as.vector(AE_Predict$Lasso_AE_fit_EPT), ci = "z-transform", conf.level = 0.95)
```



```
concordance_correlation_ztransform$rho.c

## Comparison Step Lasso
#############################################################################
AE_Predict$Difference_StepLasso <- AE_Predict$Step_AE_fit_EPT -  AE_Predict$Lasso_AE_fit_EPT

min(AE_Predict$Difference_StepLasso)
max(AE_Predict$Difference_StepLasso)
hist(AE_Predict$Difference_StepLasso, breaks = 1000, xlim = c(-50,50))
min(AE_Predict$Step_AE_se_EPT)
max(AE_Predict$Step_AE_se_EPT)
hist(AE_Predict$Step_AE_se_EPT, breaks = 1000, xlim = c(0,20))

shapiro.test_1<- shapiro.test(AE_Predict$Difference_StepLasso)
shapiro.test_1$p.value
shapiro.test_2 <- shapiro.test(AE_Predict$Step_AE_se_EPT)
shapiro.test_2$p.value

m1 <- mean(AE_Predict$Difference_StepLasso)
s1 <- sd(AE_Predict$Difference_StepLasso)
ks.test_1 <- ks.test(AE_Predict$Difference_StepLasso, "pnorm", m1, s1)
ks.test_1
m2 <- mean(AE_Predict$Step_AE_se_EPT)
s2 <- sd(AE_Predict$Step_AE_se_EPT)
ks.test_2 <- ks.test(AE_Predict$Step_AE_se_EPT, "pnorm", m2, s2)
ks.test_2

EV_cor_p <- cor(AE_Predict$Difference_StepLasso, AE_Predict$Step_AE_se_EPT, use =
"pairwise.complete.obs", method = "pearson")
EV_cor_p # since data are normally distributed  according to hist and ks.test pearson should be used

# Comparison Step Lasso with Standard Errors
#############################################################################

AE_Predict$Step_AE_se_EPT
range(AE_Predict$Step_AE_se_EPT)
AE_Predict$Step_AE_se_EPT_classify <- AE_Predict$Step_AE_se_EPT
AE_Predict$Step_AE_se_EPT_classify[AE_Predict$Step_AE_se_EPT <= 4]<-
rgb(255/255,255/255,212/255)
AE_Predict$Step_AE_se_EPT_classify[AE_Predict$Step_AE_se_EPT >= 5 &
AE_Predict$Step_AE_se_EPT <= 8]<- rgb(255/255,228/255,145/255)
AE_Predict$Step_AE_se_EPT_classify[AE_Predict$Step_AE_se_EPT >= 9 &
AE_Predict$Step_AE_se_EPT <= 12]<- rgb(255/255,196/255,145/255)
AE_Predict$Step_AE_se_EPT_classify[AE_Predict$Step_AE_se_EPT >= 13 &
AE_Predict$Step_AE_se_EPT <= 16]<- rgb(255/255,155/255,41/255)
AE_Predict$Step_AE_se_EPT_classify[AE_Predict$Step_AE_se_EPT >= 17 &
AE_Predict$Step_AE_se_EPT <= 20]<- rgb(237/255,114/255,19/255)
AE_Predict$Step_AE_se_EPT_classify[AE_Predict$Step_AE_se_EPT >= 21 &
AE_Predict$Step_AE_se_EPT <= 151]<- rgb(204/255,76/255,2/255)
AE_Predict$Step_AE_se_EPT_classify
range(AE_Predict$Step_AE_se_EPT_classify)
```

```
plot(AE_Predict$Step_AE_fit_EPT, AE_Predict$Lasso_AE_fit_EPT, xlab = "Step Prediction - Nationwide - AE_2km2", ylab = "Lasso Prediction - Nationwide - AE_2km2",
    bg=AE_Predict$Step_AE_se_EPT_classify, pch=21)
par(xpd=F)
abline(0,1)
par(mar=c(5, 5, 5, 15), xpd=TRUE)
l1 <- legend(
  x = 600 , y = 110,
  legend = expression(bold("Estimated standard error")),
  , cex=1,
  bty='n')
l2 <- legend(
  x = 650 , y = 80,
  legend=c(expression("0-4")
           , expression("5-8")
           , expression("9-12")
           , expression("13-16")
           , expression("17-20")
           , expression("21-151")),
  , col=c( rgb(255/255,255/255,255/255,212/255), rgb(255/255,228/255,145/255), rgb(255/255,196/255,145/255), rgb(255/255,155/255,41/255), rgb(237/255,114/255,19/255), rgb(204/255,76/255,2/255))
  , pch = c(16,16,16)
  , cex=1,
  xjust = 0,
  bty='n')
```

6.2 NationwideSamplingArea_JoinPredictionByID_EPT ArcGis Field Calculator Query

```
# ---------------------------------------------------------------------------------------------------------------------
# NationwideSamplingArea_JoinPredictionByID_EPT ArcGis Field Calculator Query
# Description: Joins the nationwide EPT species richness prediction to the nationwide sampling area Polygons (AE_2km2)
# ---------------------------------------------------------------------------------------------------------------------

Field names containing invalid characters can cause the join to fail. The following fields contain invalid characters;
- [wastewater_m3.a] from <AE_Predict_EPT.csv> contains invalid character '.'
- [stormsewage_m3.a] from <AE_Predict_EPT.csv> contains invalid character '.'
- [Q_amean_m3.s] from <AE_Predict_EPT.csv> contains invalid character '.'
- [Qvar_amean_m3.s] from <AE_Predict_EPT.csv> contains invalid character '.'
- [Q_amax_m3.s] from <AE_Predict_EPT.csv> contains invalid character '.'

The number of matching records for the join:
- 20745 of 22169 records matched by joining [OBJECTID] from <basisgeometrie> with [AE_2km2_ID] from <AE_Predict_EPT.csv>.
Matching records may not appear in table view due to join validation errors.
```

6.3 NationwideSamplingArea_CH_EPT.py

```
# ---------------------------------------------------------------------------------------------------------------------
# NationwideSamplingArea_CH_EPT.py
# Description: This Script deletes AE_2km2 catchments which are not located in CH
# ---------------------------------------------------------------------------------------------------------------------
```

```
# Import arcpy module
import arcpy
import itertools

# Path
Path = "..."

# Set workspace
arcpy.env.workspace = Path

# Allow to overwrite output
arcpy.gp.overwriteOutput = True

# Define Variables: Input
Prediction = ".../AE_JoinPrediction_EPT.shp"
Prediction_CH_lyr = "Prediction_CH_lyr"
print "Input Variables are defined"

# Define Variables: Output
Prediction_CH = ".../AE_JoinPrediction_CH_EPT.shp"
Prediction_CH_Dissolve = /AE_JoinPrediction_CH_Dissolve.shp"
print "Output Variables are defined"

# Delete catchments which are not located in CH
# Execute CopyFeatures to make a new copy of the feature class
arcpy.CopyFeatures_management(Prediction, Prediction_CH)

# Execute MakeFeatureLayer
arcpy.MakeFeatureLayer_management(Prediction_CH, Prediction_CH_lyr)

# Execute SelectLayerByAttribute to determine which features to delete
arcpy.SelectLayerByAttribute_management(Prediction_CH_lyr, "NEW_SELECTION", "'CH" = -1')

#  Delete Selected features
arcpy.DeleteFeatures_management(Prediction_CH_lyr)

# Dissolve Prediction_CH
# arcpy.Dissolve_management(Prediction_CH, Prediction_CH_Dissolve)
```

6.4 BDMSamplingSite_JoinNationwidePredictionByLocation.py

```
# ---------------------------------------------------------------------------------------------------------------------
# BDMSamplingSite_JoinNationwidePredictionByLocation.py
# Description: Joins nationwide prediction attributes to BDM sampling sites via spatial query
# ---------------------------------------------------------------------------------------------------------------------

# Import arcpy module
import arcpy

# Path
Path = "..."

# Set workspace
arcpy.env.workspace = Path

# Allow to overwrite output
arcpy.gp.overwriteOutput = True
```

```
# Define Variables: Input
BDMPoints = ".../BDMPoints.shp"
Prediction = ".../AE_JoinPrediction_CH_EPT.shp"
print "Input Variables are defined"

# Define Variables: Output
BDMPoints_Prediction_JoinAttributeByLocation =
".../BDMPoints_Prediction_JoinAttributeByLocation_EPT.shp"
print "Output Variables are defined"

# Spatial Join BDM Point
arcpy.SpatialJoin_analysis(BDMPoints, Prediction, BDMPoints_Prediction_JoinAttributeByLocation ,
"JOIN_ONE_TO_MANY", "KEEP_COMMON") # match option = Intersect
print "Join is done"
```

6.5 NationalPrediction_Validation_EPT.R

```
# ------------------------------------------------------------------------------------------------------------------------
# NationalPrediction_Validation_EPT.R
# Description: Prepares the EPT species richness prediction
# ------------------------------------------------------------------------------------------------------------------------

## Clear R's memory
rm(list=ls())

## library
################################################################################
library(reshape2)
library(epiR)

## Import Data
################################################################################
AE <- read.csv(".../BDMPoints_Prediction_JoinAttributeByLocation_EPT.csv", header=T, sep="\t",
stringsAsFactors=FALSE)
BDM <- read.csv(".../BDMPoints_Predict_EPT.csv", header=T, sep="\t")
EV <- read.csv(".../EV_BDMPoints.csv", header=T, sep="\t", stringsAsFactors=FALSE)

str(AE)
str(BDM)
str(EV)

dim(AE)
dim(BDM)
dim(EV)

names(AE)
names(BDM)
names(EV)

## Join2
################################################################################
```



```
dim(BDM)
Join1 <- merge(x = BDM, y = AE[c("BDM_RowID", "Step_AE_fi", "Step_AE_se", "Step_AE_rs",
"Lasso_AE_f")], by = "BDM_RowID", all.x = T)
dim(Join1)
str(Join1)
Join2 <- merge(x = Join1, y = EV[c("BDM_RowID", "hydro_class")], by = "BDM_RowID", all.x = T)
dim(Join2)
str(Join2)
names(Join2)

## Plot difference between AE_2km2 and WSB dataset
################################################################################

par(mfrow = c(1,2))
plot(Join2$Step_BDM_fit_EPT, Join2$Step_AE_fi, xlab = "Step Prediction - BDM sites - WSB", ylab =
"Step Prediction - BDM sites - AE_2km2", xlim=c(0,40), ylim=c(0,40))
abline(0,1)
plot(Join2$Lasso_BDM_fit_EPT, Join2$Lasso_AE_f, xlab = "Lasso Prediction - BDM sites - WSB", ylab =
"Lasso Prediction - BDM sites - AE_2km2", xlim=c(0,40), ylim=c(0,40))
abline(0,1)

concordance_correlation_ztransform1 <- epi.ccc(as.vector(Join2$Step_BDM_fit_EPT),
as.vector(Join2$Step_AE_fi), ci = "z-transform", conf.level = 0.95)
concordance_correlation_ztransform1$rho.c

concordance_correlation_ztransform2 <- epi.ccc(as.vector(Join2$Lasso_BDM_fit_EPT),
as.vector(Join2$Lasso_AE_f), ci = "z-transform", conf.level = 0.95)
concordance_correlation_ztransform2$rho.c

## Point distance from 1:1 Linie = field data - model data
################################################################################

# GAB
names(Join2)
dim(Join2)

par(mfrow = c(1,3))

# Step
model_data <- c(Join2$Step_BDM_fit_EPT)
reality_data<- c(Join2$BDM_a_EPT) # BDM
range(Join2$Step_BDM_fit_EPT)
range(Join2$BDM_a_EPT)
plot(reality_data, model_data, xlim = c(0,35), ylim = c(0,35), xlab = "Monitoring" , ylab = "Step Prediction -
BDM sites - WSB")
abline(0,1)
Resid_S <- abs(reality_data - model_data)
Resid_S

# Lasso
model_data <- c(Join2$Lasso_BDM_fit_EPT)
reality_data<- c(Join2$BDM_a_EPT) # BDM
range(Join2$Lasso_BDM_fit_EPT)
```

```
range(Join2$BDM_a_EPT)
plot(reality_data, model_data, xlim = c(0,35), ylim = c(0,35), xlab = "Monitoring" , ylab = "Lasso Prediction - BDM sites - WSB")
abline(0,1)
Resid_L <- abs(reality_data - model_data)
Resid_L

# mean(Step,Lasso)
model_data <- c((Join2$Step_BDM_fit_EPT + Join2$Lasso_BDM_fit_EPT) / 2)
reality_data<- c(Join2$BDM_a_EPT) # BDM
range((Join2$Step_BDM_fit_EPT + Join2$Lasso_BDM_fit_EPT) / 2)
range(Join2$BDM_a_EPT)
plot(reality_data, model_data, xlim = c(0,35), ylim = c(0,35), xlab = "Monitoring" , ylab = "mean(Lasso, Step) Prediction - BDM sites - WSB")
abline(0,1)
Resid_SL <- abs(reality_data - model_data)
Resid_SL

# Write to output
BDMPoints_Predict <- read.csv(".../BDMPoints_Predict_EPT.csv", header=T, sep="\t")
BDMPoints_Predict$Resid_S <- Resid_S
BDMPoints_Predict$Resid_L <- Resid_L
BDMPoints_Predict$Resid_SL <- Resid_SL
#write.table(BDMPoints_Predict,".../BDMPoints_Predict_EPT.csv", sep="\t", quote = F, row.names = F)
# NA <- colSums(!is.na(BDMPoints_Predict))
# NotZero <-colSums(BDMPoints_Predict != 0)

concordance_correlation_ztransform3 <- epi.ccc(as.vector(Join2$BDM_a_EPT),
as.vector(Join2$Step_BDM_fit_EPT), ci = "z-transform", conf.level = 0.95)
concordance_correlation_ztransform3$rho.c

concordance_correlation_ztransform4 <- epi.ccc(as.vector(Join2$BDM_a_EPT),
as.vector(Join2$Lasso_BDM_fit_EPT), ci = "z-transform", conf.level = 0.95)
concordance_correlation_ztransform4$rho.c

concordance_correlation_ztransform5 <- epi.ccc(as.vector(Join2$BDM_a_EPT),
c((Join2$Step_BDM_fit_EPT + Join2$Lasso_BDM_fit_EPT) / 2), ci = "z-transform", conf.level = 0.95)
concordance_correlation_ztransform5$rho.c

names(Join2)

## Residual per hydro_class boxplot
##############################################################################################
BDM <- read.csv(".../BDMPoints_Predict_EPT.csv", header=T, sep="\t")
# This script has to be run two times in order to work

par(mfrow = c(2,1))
# Pie Chart
# Sum all "BDM_a_EPT" values that  have the same hydro_class name
EPT_per_hydroclass <- aggregate(Join2$BDM_a_EPT,by = list(Join2$hydro_class) , sum)
EPT_per_hydroclass$Percentage <- EPT_per_hydroclass$x/sum(EPT_per_hydroclass$x)*100
sum(EPT_per_hydroclass$Percentage)
test <- as.vector(EPT_per_hydroclass$Group.1)
```

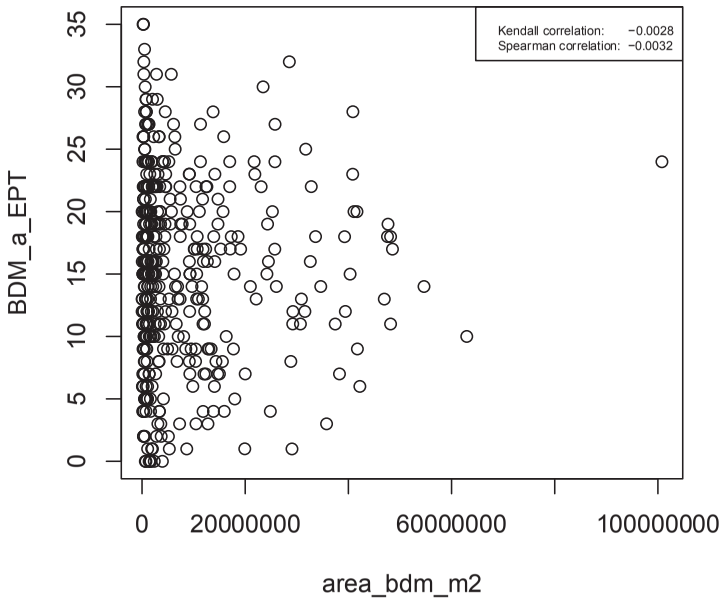
```
barplot(EPT_per_hydroclass$Percentage, names.arg = test <- as.vector(EPT_per_hydroclass$Group.1),
ylim = c(0, 30),  ylab="Percentage of total EPT species", xlab="Catchment")
# Boxplot of MPG by Car Cylinders
boxplot(Join2$Resid_SL~Join2$hydro_class,data=Join2, xlab="Catchment", ylab="Residual")
#write.table(Join2,".../BDMPoints_Resid_EPT.csv", sep="\t", quote = F, row.names = F)

## Point distance from 1:1 Linie = field data - model data with Standard Errors
##############################################################################################

# http://stackoverflow.com/questions/17551193/r-color-scatter-plot-points-based-on-values
# http://www.statmethods.net/advgraphs/parameters.html

par(mfrow = c(1,3))

# Step
model_data <- c(Join2$Step_BDM_fit_EPT)
reality_data<- c(Join2$BDM_a_EPT) # BDM
range(Join2$Step_BDM_fit_EPT)
range(Join2$BDM_a_EPT)

Join2$Step_BDM_se_EPT
range(Join2$Step_BDM_se_EPT)
Join2$Step_BDM_se_EPT_classify <- Join2$Step_BDM_se_EPT
Join2$Step_BDM_se_EPT_classify[Join2$Step_BDM_se_EPT <= 1]<- "yellow"
Join2$Step_BDM_se_EPT_classify[Join2$Step_BDM_se_EPT >= 2 & Join2$Step_BDM_se_EPT <= 3]<- "darkgoldenrod1"
Join2$Step_BDM_se_EPT_classify[Join2$Step_BDM_se_EPT >= 4 & Join2$Step_BDM_se_EPT <= 5]<- "red3"
Join2$Step_BDM_se_EPT_classify
range(Join2$Step_BDM_se_EPT_classify)


plot(reality_data, model_data, xlim = c(0,35), ylim = c(0,35), xlab = "Monitoring" , ylab = "Step Prediction - BDM sites - WSB",
    bg=Join2$Step_BDM_se_EPT_classify, pch=21)
abline(0,1)
l1 <- legend(
  x = -3 , y = 37,
  legend = expression(bold("Estimated standard error")),
  , cex=1,
  bty='n')

l2 <- legend(
  x = -1 , y = 34,
  legend=c(expression(">1")
           , expression("2-3")
           , expression("4-5")),
  , col=c("yellow", "darkgoldenrod1", "red3")
  , pch = c(16,16,16)
  , cex=1,
  xjust = 0,
  bty='n')
```

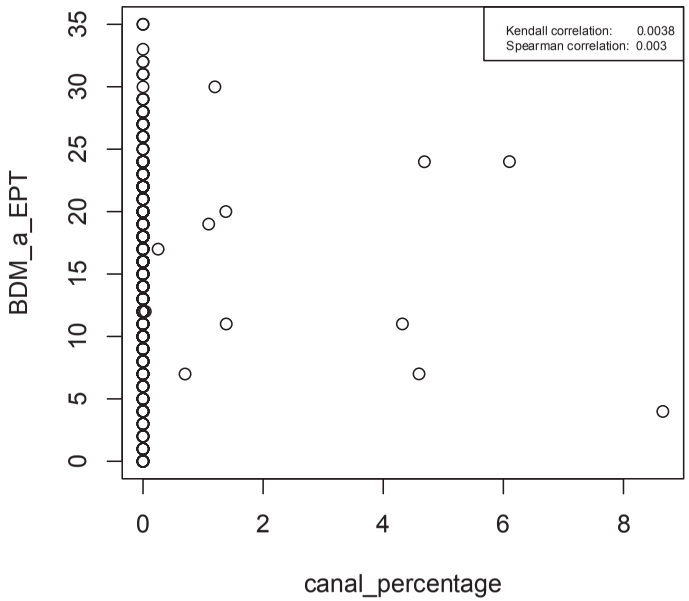
```
Resid_S <- abs(reality_data - model_data)
Resid_S

# Lasso
model_data <- c(Join2$Lasso_BDM_fit_EPT)
reality_data<- c(Join2$BDM_a_EPT) # BDM
range(Join2$Lasso_BDM_fit_EPT)
range(Join2$BDM_a_EPT)
plot(reality_data, model_data, xlim = c(0,35), ylim = c(0,35), xlab = "Monitoring" , ylab = "Lasso Prediction -
BDM sites - WSB")
abline(0,1)
Resid_L <- abs(reality_data - model_data)
Resid_L

# mean(Step,Lasso)
model_data <- c((Join2$Step_BDM_fit_EPT + Join2$Lasso_BDM_fit_EPT) / 2)
reality_data<- c(Join2$BDM_a_EPT) # BDM
range((Join2$Step_BDM_fit_EPT + Join2$Lasso_BDM_fit_EPT) / 2)
range(Join2$BDM_a_EPT)
plot(reality_data, model_data, xlim = c(0,35), ylim = c(0,35), xlab = "Monitoring" , ylab = "mean(Lasso, Step)
Prediction - BDM sites - WSB")
abline(0,1)
Resid_SL <- abs(reality_data - model_data)
Resid_SL
```

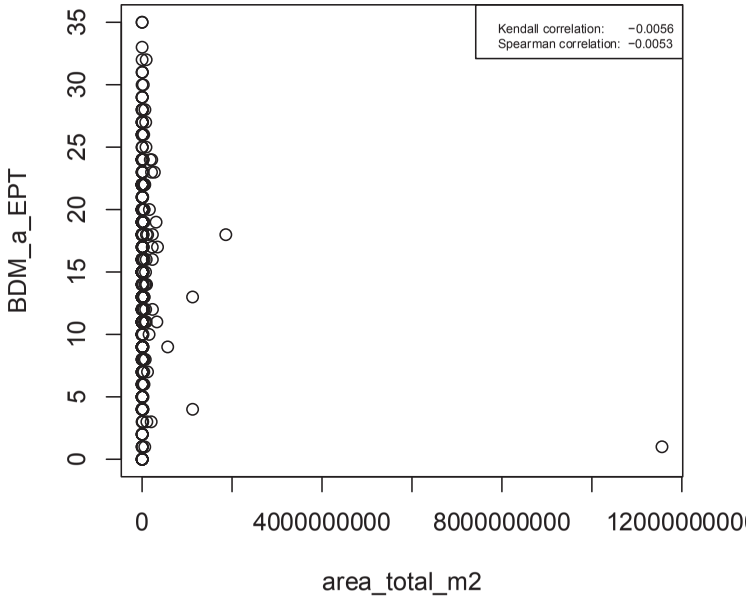
6.6 BDMSamplingArea_JoinResidualByID

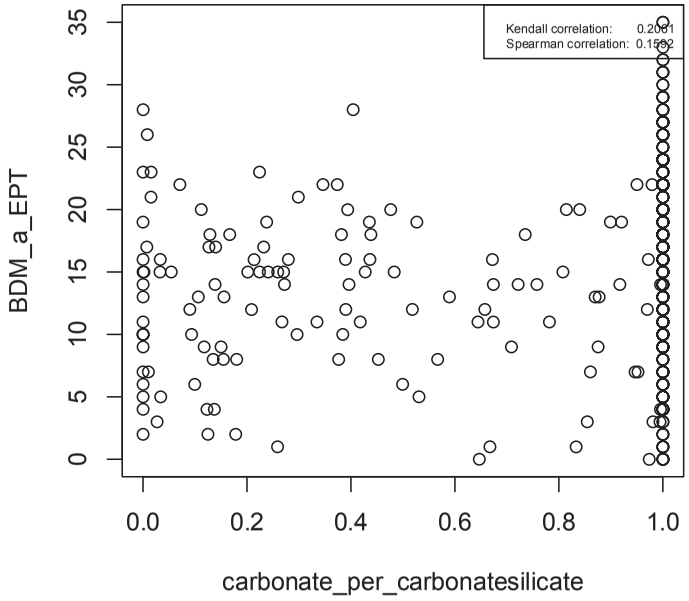
```
#---------------------------------------------------------------------------------------------------------------------------
# BDMSamplingArea_JoinResidualByID
# Description:Joins Residuals to BDMPoints via common ID
#---------------------------------------------------------------------------------------------------------------------------

# Join <BDMPoints_Predict_EPT_Join.txt> (Residuals) to <EZG_All_GAB> = BDMPoints_Resid_EPT.shp
=> <BDMPoints_Resid_EPT_FormatCellsNumber.csv>. had to be created because the numbers of
<BDMPoints_Resid_EPT.csv> are recognized as strings.

Field names that match reserved words should not be used in database schema and can cause the join to
fail. The following fields match reserved words;
- [Check] from <EZG_All_GAB>

The number of matching records for the join:
- 410 of 410 records matched by joining [BDM_RowID] from <EZG_All_GAB> with [BDM_RowID] from
<BDMPoints_Resid_EPT_FormatCellsNumber.csv>.
Matching records may not appear in table view due to join validation errors.
```

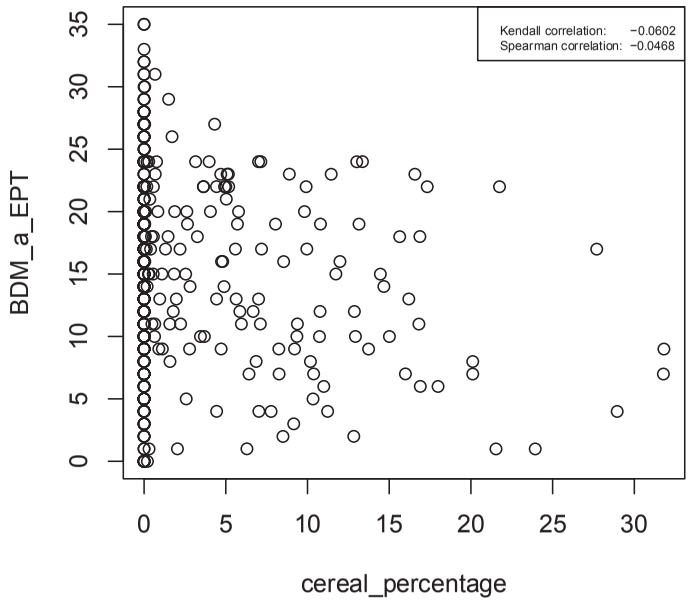
2 Scatterplots: EPT species vs. explanatory variables

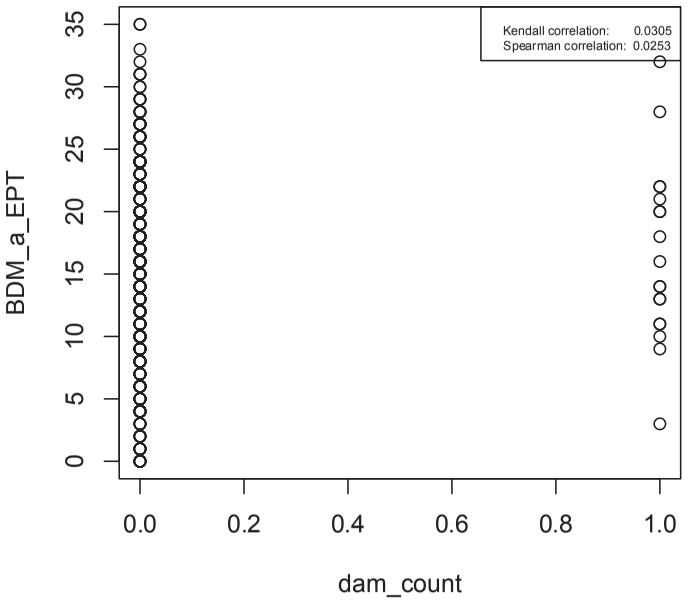


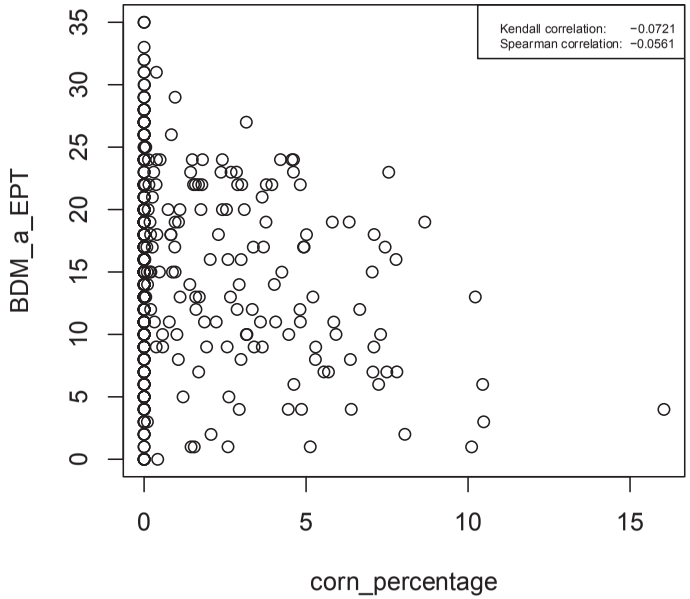


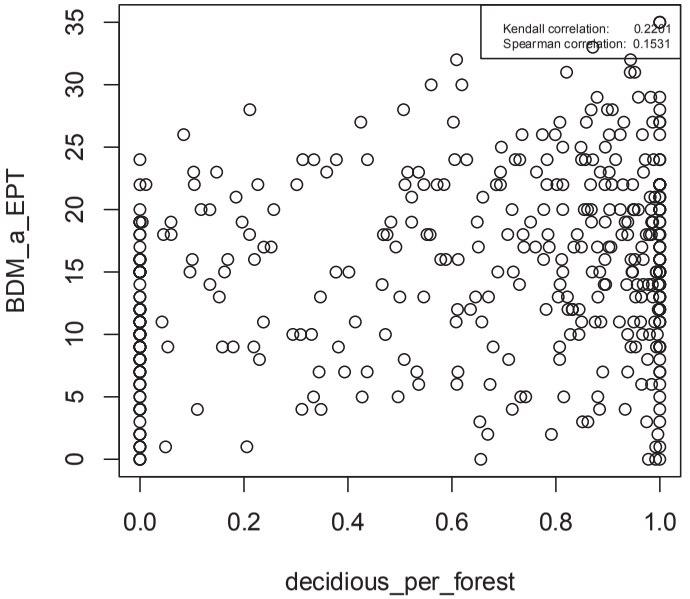


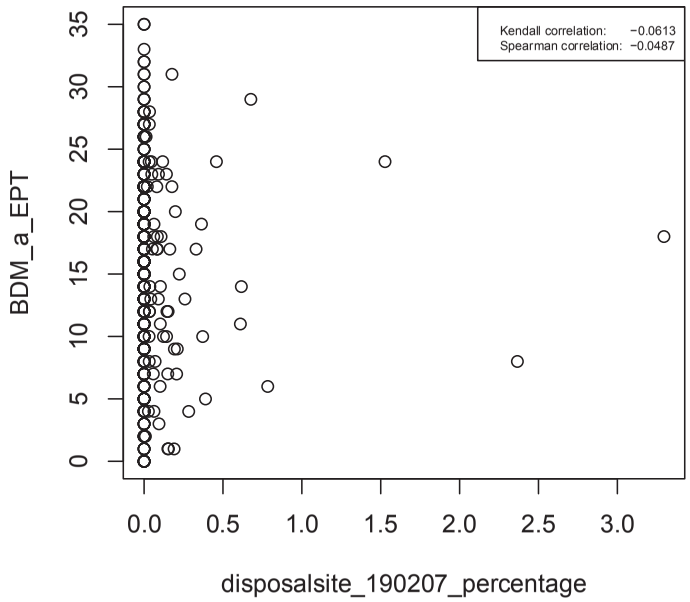


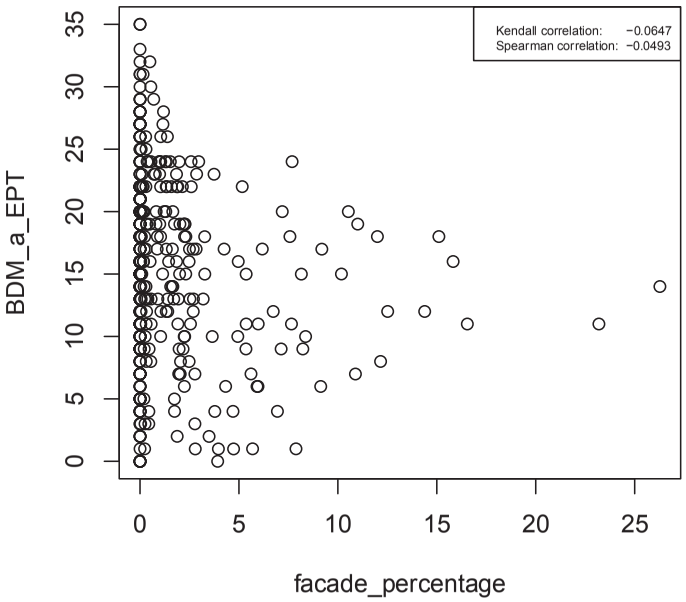

Kendall correlation: −0.0602
Spearman correlation: −0.0468
BDM_a_EPT
cereal_percentage

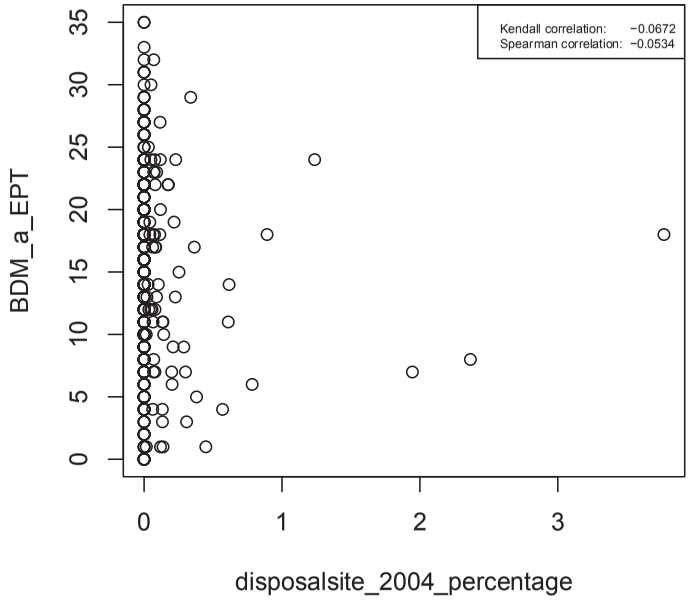

Kendall correlation: 0.0305
Spearman correlation: 0.0253
BDM_a_EPT
dam_count

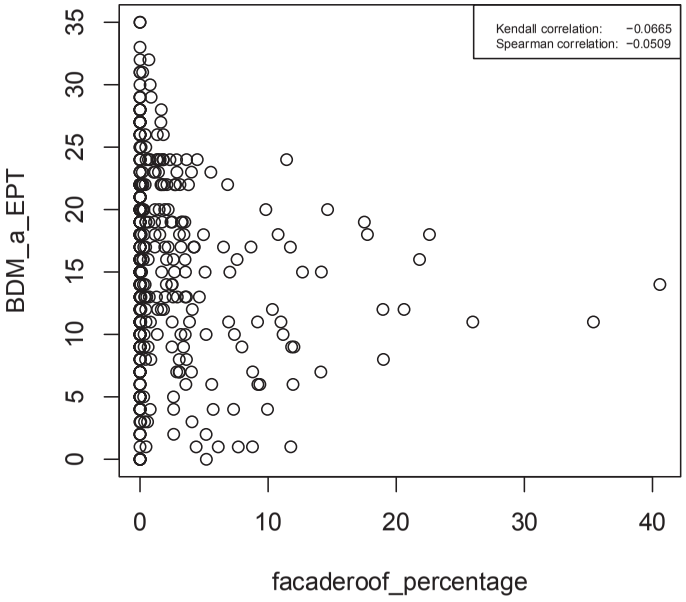

Kendall correlation: −0.0721
Spearman correlation: −0.0561
BDM_a_EPT
corn_percentage

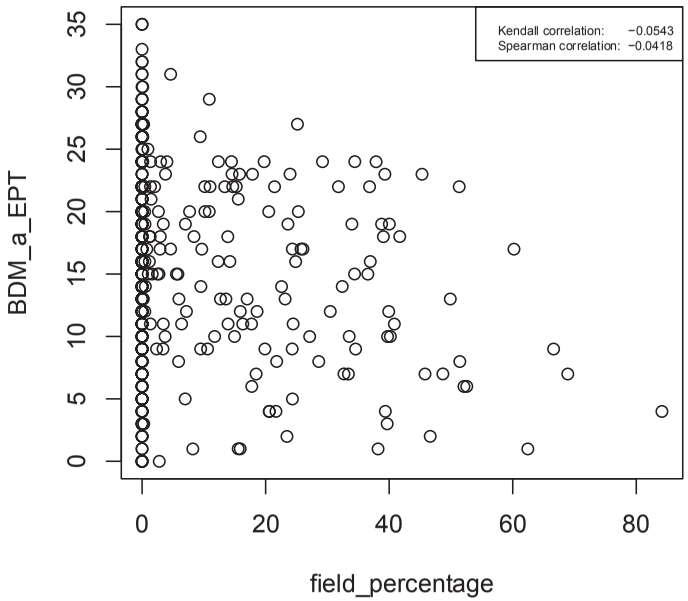

Kendall correlation: 0.2291
Spearman correlation: 0.1531
BDM_a_EPT
decidious_per_forest

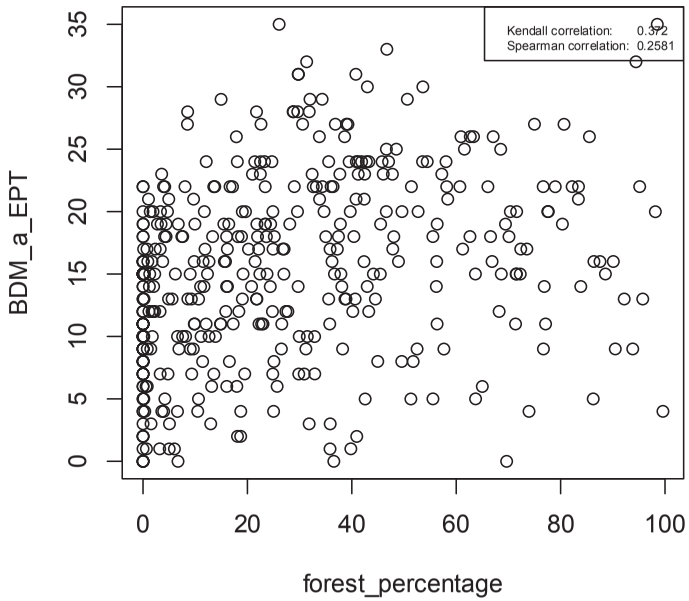

Kendall correlation: −0.0613
Spearman correlation: −0.0487
BDM_a_EPT
disposalsite_190207_percentage

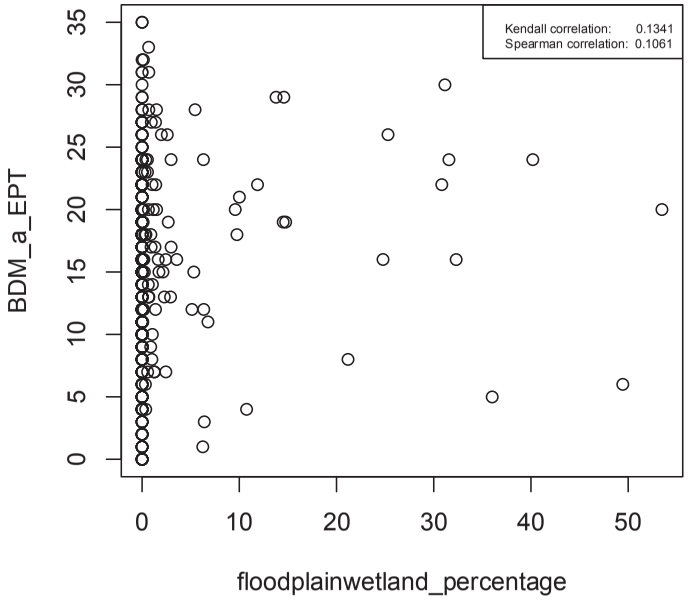

Kendall correlation: −0.0647
Spearman correlation: −0.0493
BDM_a_EPT
facade_percentage

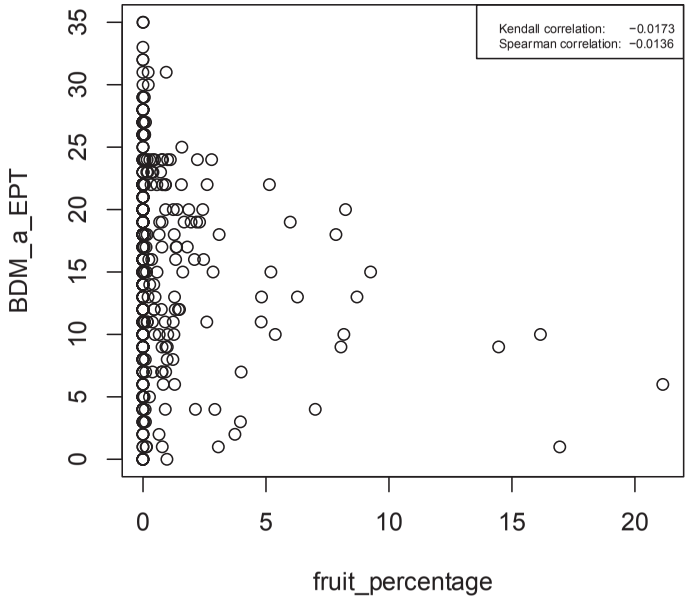

Kendall correlation: −0.0672
Spearman correlation: −0.0534
BDM_a_EPT
disposalsite_2004_percentage

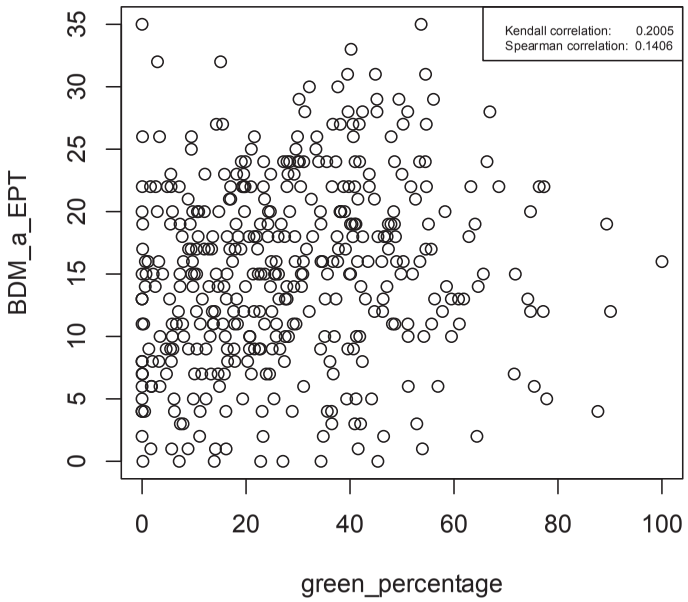

Kendall correlation: −0.0665
Spearman correlation: −0.0509
BDM_a_EPT
facaderoof_percentage

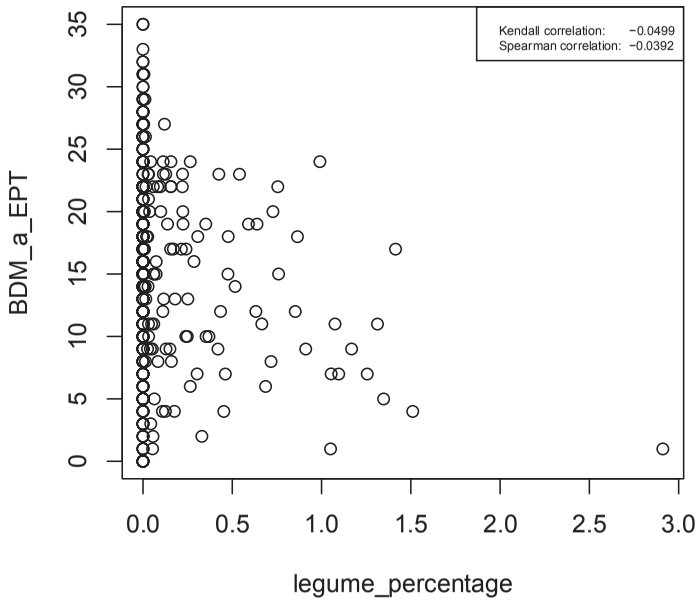

Kendall correlation: −0.0543
Spearman correlation: −0.0418
BDM_a_EPT
field_percentage
0
5
10
15
20
25
30
35
0
20
40
60
80

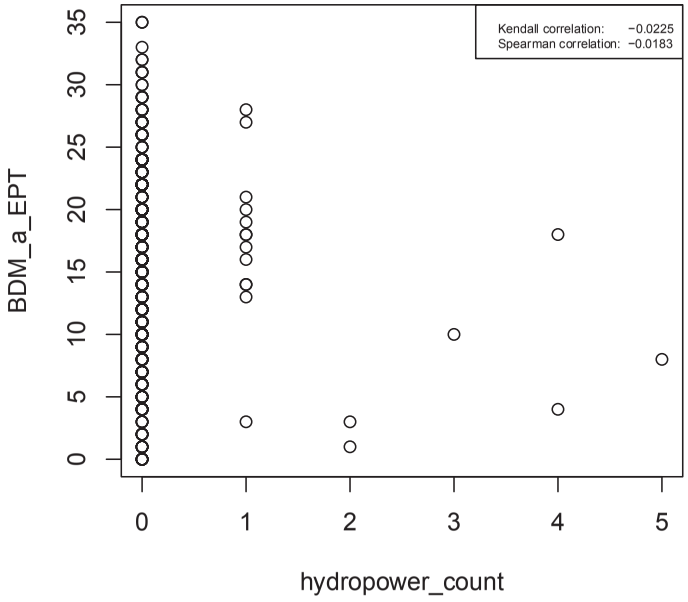

Kendall correlation: 0.372
Spearman correlation: 0.2581
BDM_a_EPT
forest_percentage
0
5
10
15
20
25
30
35
0
20
40
60
80
100

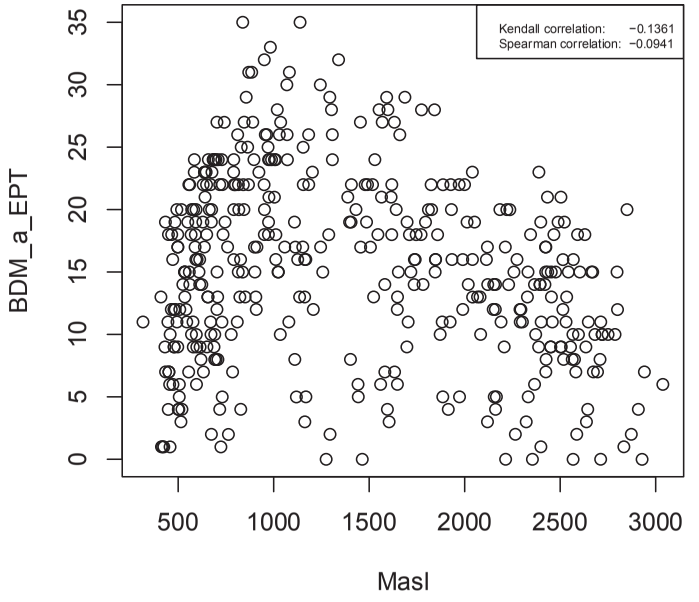

Kendall correlation: 0.1341
Spearman correlation: 0.1061
BDM_a_EPT
floodplainwetland_percentage
0
5
10
15
20
25
30
35
0
10
20
30
40
50

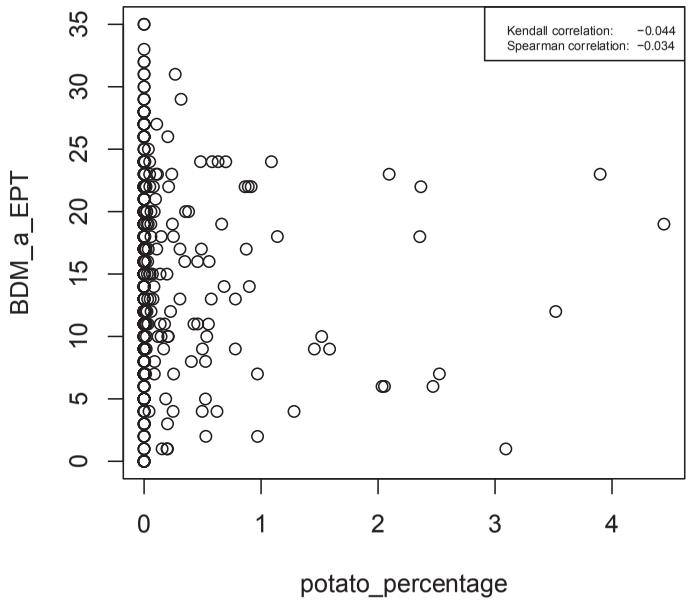

Kendall correlation: −0.0173
Spearman correlation: −0.0136
BDM_a_EPT
fruit_percentage
0
5
10
15
20
25
30
35
0
5
10
15
20

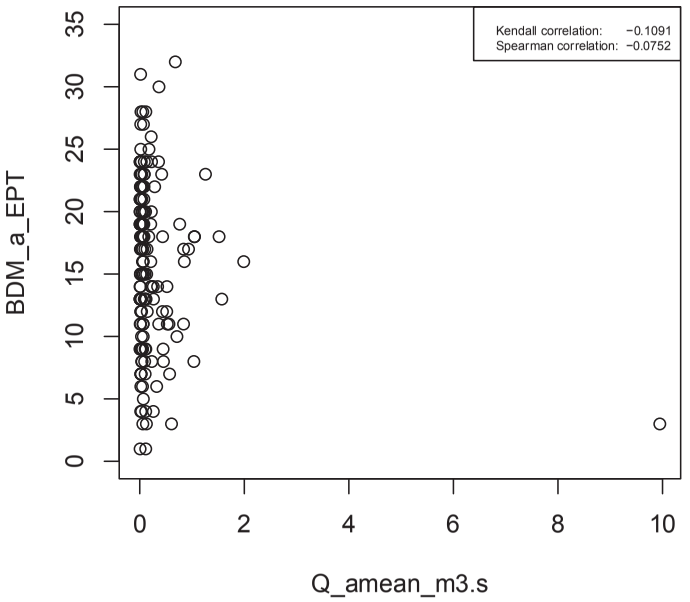

Kendall correlation: 0.2005
Spearman correlation: 0.1406
BDM_a_EPT
green_percentage

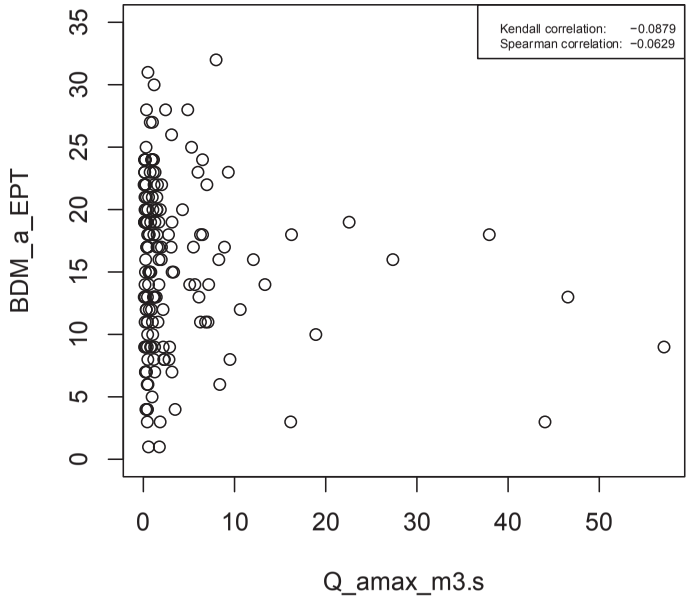

Kendall correlation: −0.0499
Spearman correlation: −0.0392
BDM_a_EPT
legume_percentage

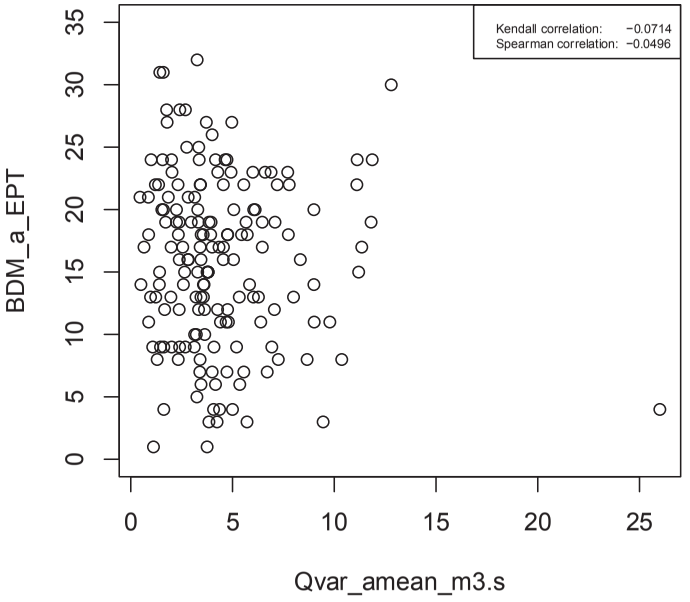

Kendall correlation: −0.0225
Spearman correlation: −0.0183
BDM_a_EPT
hydropower_count

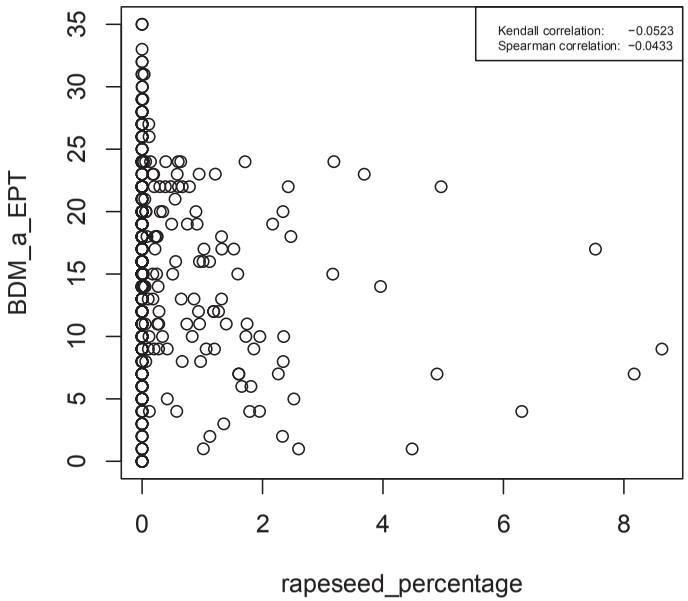

Kendall correlation: −0.1361
Spearman correlation: −0.0941
BDM_a_EPT
Masl

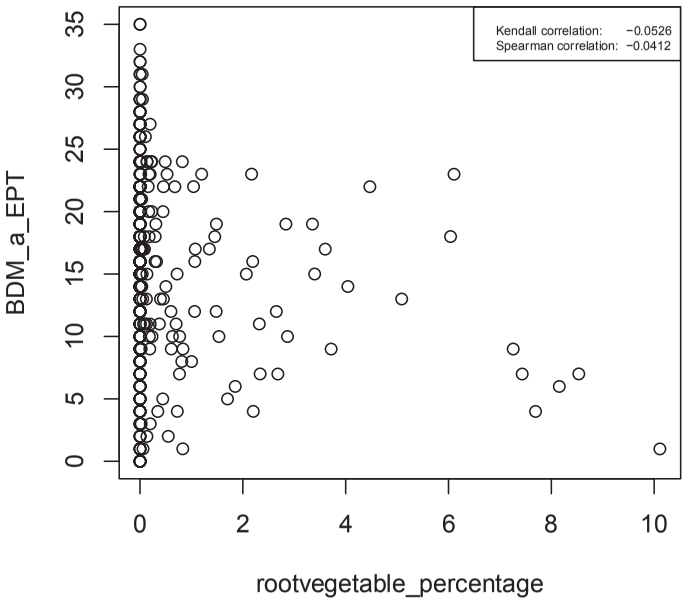

Kendall correlation: −0.044
Spearman correlation: −0.034
BDM_a_EPT
potato_percentage
0
5
10
15
20
25
30
35
0
1
2
3
4

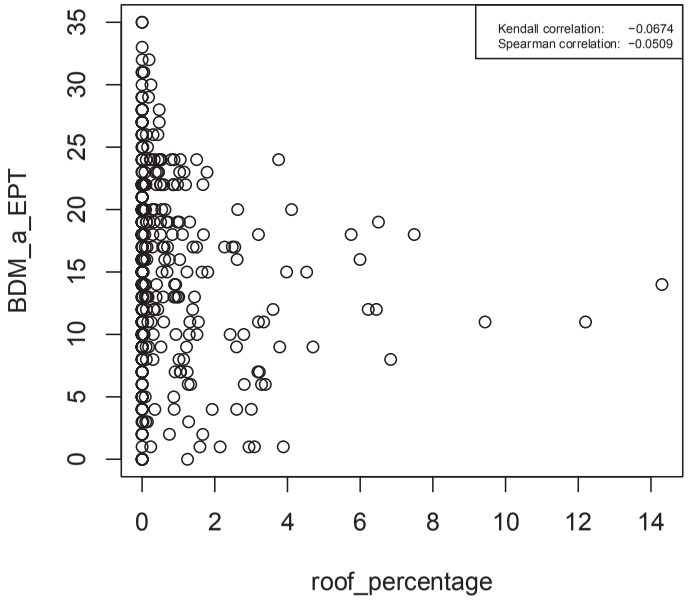

Kendall correlation: −0.1091
Spearman correlation: −0.0752
BDM_a_EPT
Q_amean_m3.s
0
5
10
15
20
25
30
35
0
2
4
6
8
10

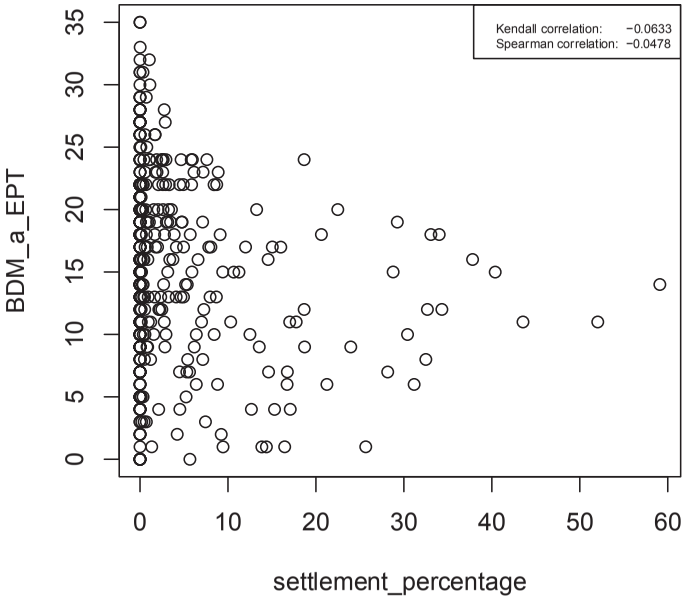

Kendall correlation: −0.0879
Spearman correlation: −0.0629
BDM_a_EPT
Q_amax_m3.s
0
5
10
15
20
25
30
35
0
10
20
30
40
50

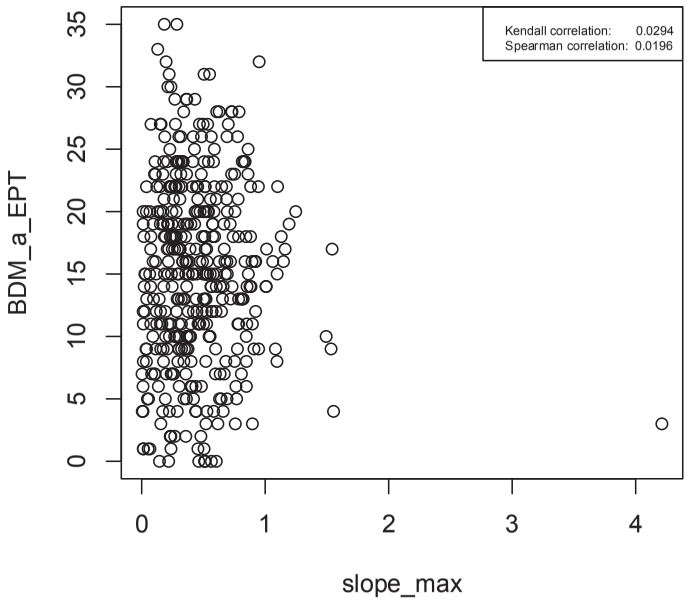

Kendall correlation: −0.0714
Spearman correlation: −0.0496
BDM_a_EPT
Qvar_amean_m3.s
0
5
10
15
20
25
30
35
0
5
10
15
20
25

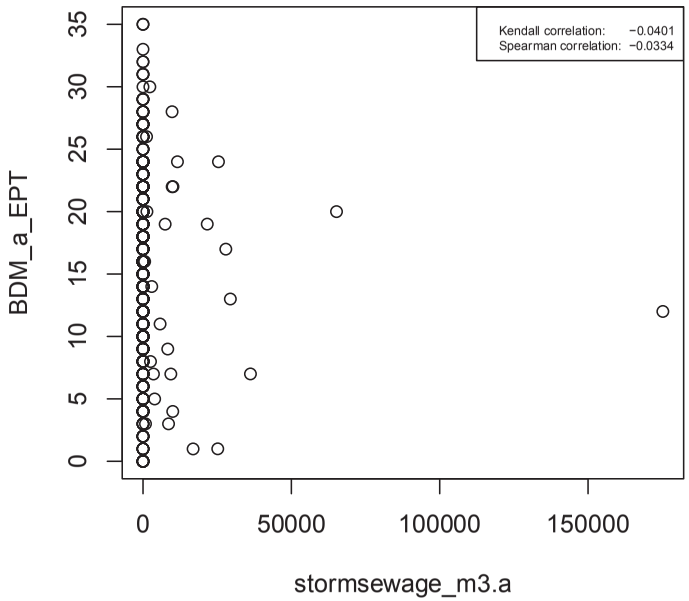

Kendall correlation: −0.0523
Spearman correlation: −0.0433
BDM_a_EPT
rapeseed_percentage

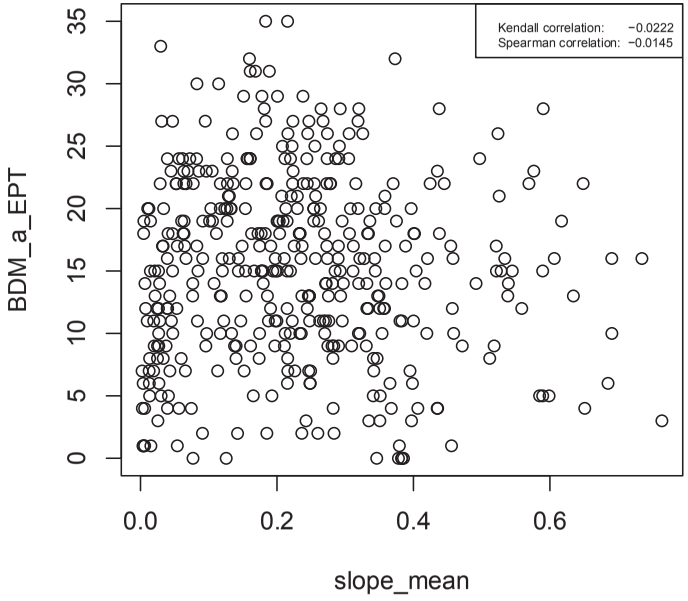

Kendall correlation: −0.0526
Spearman correlation: −0.0412
BDM_a_EPT
rootvegetable_percentage

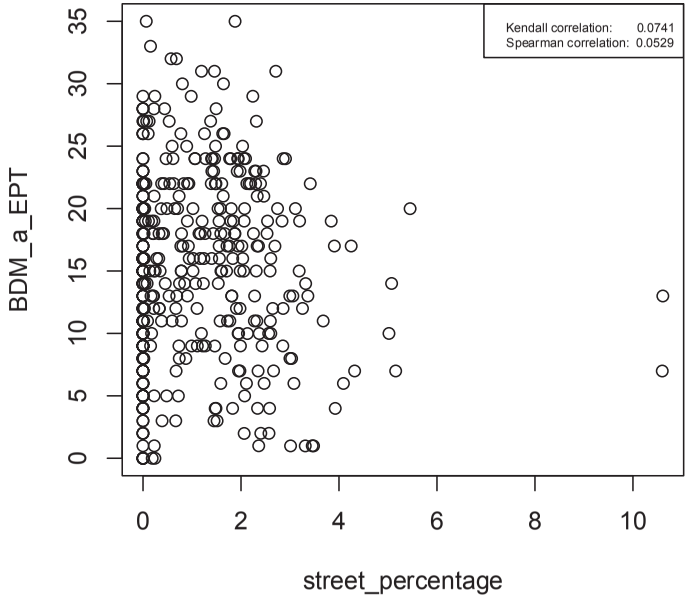

Kendall correlation: −0.0674
Spearman correlation: −0.0509
BDM_a_EPT
roof_percentage

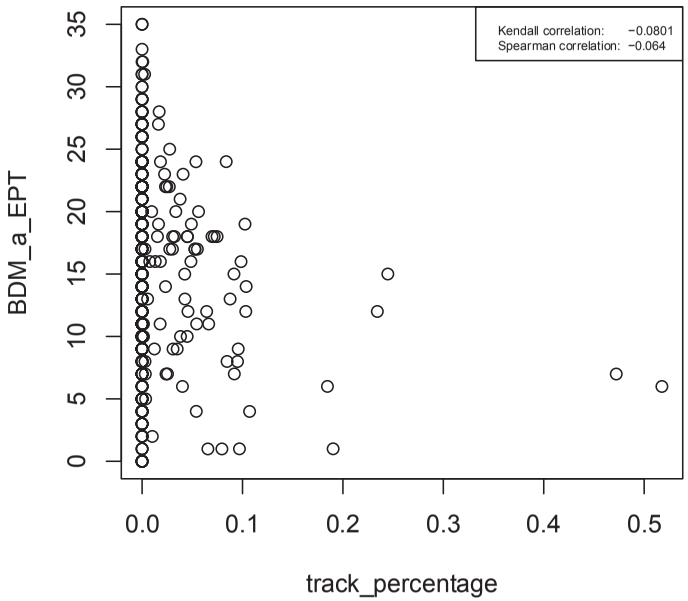

Kendall correlation: −0.0633
Spearman correlation: −0.0478
BDM_a_EPT
settlement_percentage

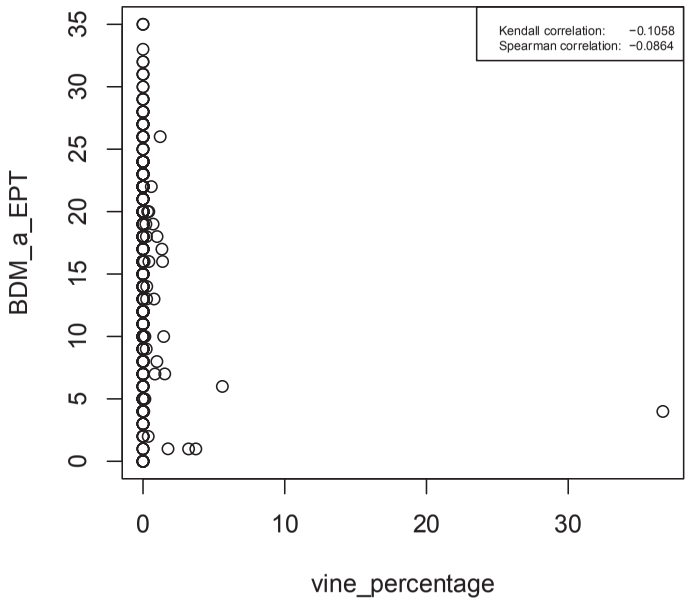

Kendall correlation: 0.0294
Spearman correlation: 0.0196
BDM_a_EPT
slope_max
0
5
10
15
20
25
30
35
0
1
2
3
4

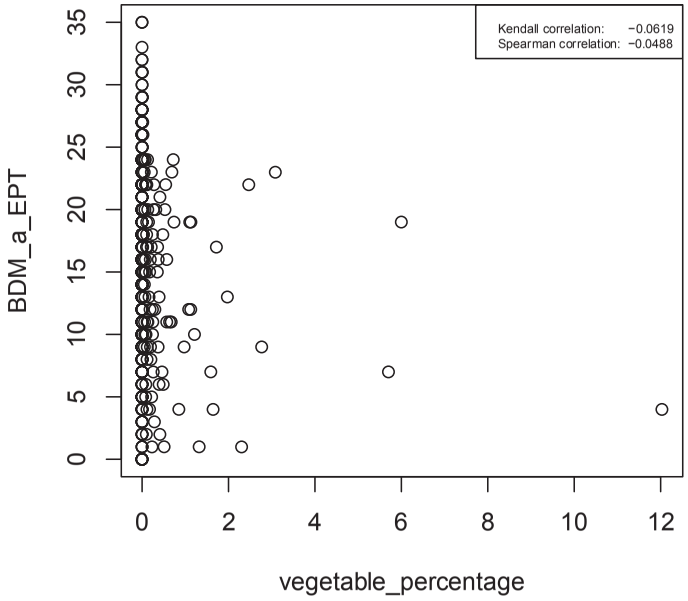

Kendall correlation: −0.0401
Spearman correlation: −0.0334
BDM_a_EPT
stormsewage_m3.a
0
5
10
15
20
25
30
35
0
50000
100000
150000

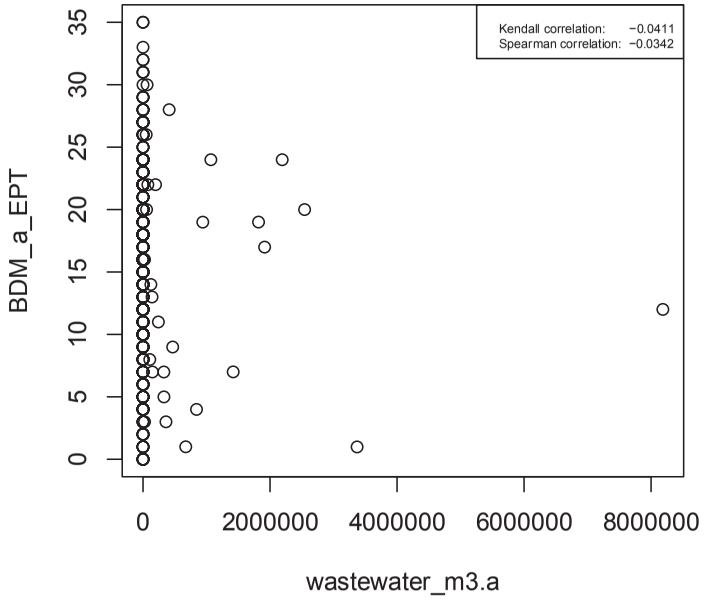

Kendall correlation: −0.0222
Spearman correlation: −0.0145
BDM_a_EPT
slope_mean
0
5
10
15
20
25
30
35
0.0
0.2
0.4
0.6


Kendall correlation: 0.0741
Spearman correlation: 0.0529
BDM_a_EPT
street_percentage
0
5
10
15
20
25
30
35
0
2
4
6
8
10


Kendall correlation: −0.0801
Spearman correlation: −0.064
BDM_a_EPT
track_percentage
0 5 10 15 20 25 30 35
0.0 0.1 0.2 0.3 0.4 0.5


Kendall correlation: −0.1058
Spearman correlation: −0.0864
BDM_a_EPT
vine_percentage
0 5 10 15 20 25 30 35
0 10 20 30

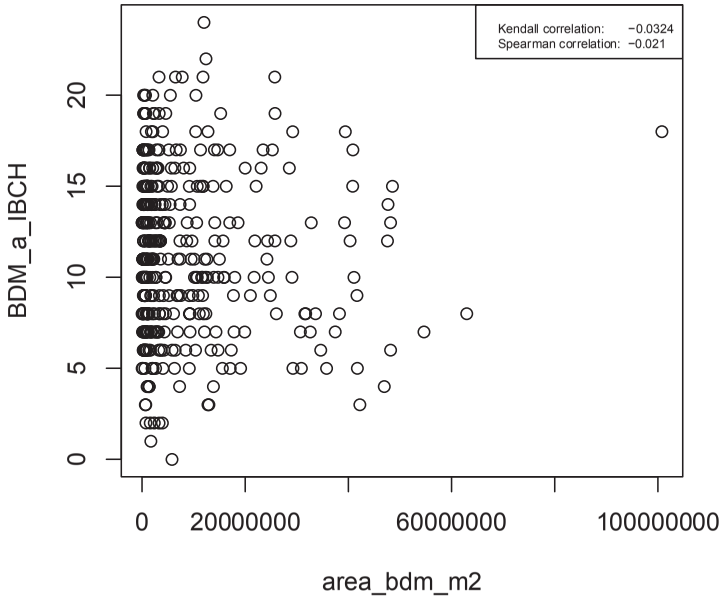

Kendall correlation: −0.0619
Spearman correlation: −0.0488
BDM_a_EPT
vegetable_percentage
0 5 10 15 20 25 30 35
0 2 4 6 8 10 12

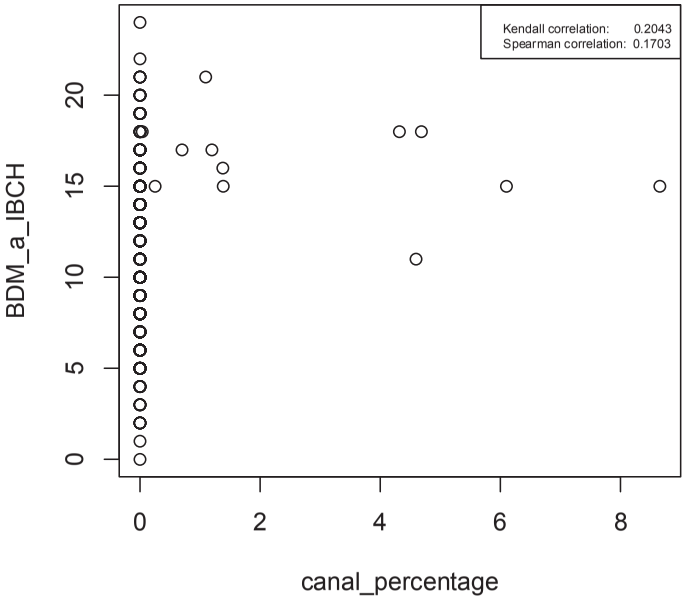

Kendall correlation: −0.0411
Spearman correlation: −0.0342
BDM_a_EPT
wastewater_m3.a
0 5 10 15 20 25 30 35
0 2000000 4000000 6000000 8000000

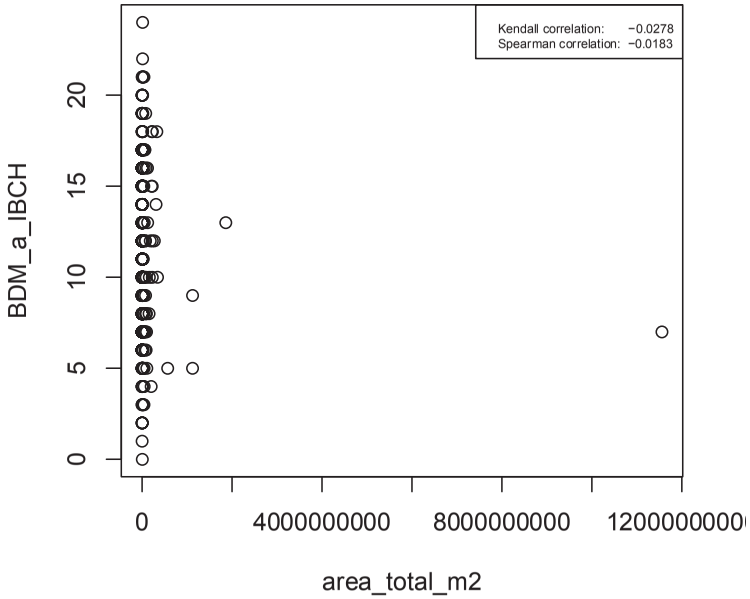
Kendall correlation: 0.0737
Spearman correlation: 0.0494
BDM_a_EPT
0
5
10
15
20
25
30
35
0
20000
60000
100000
watercourse_bdm_m

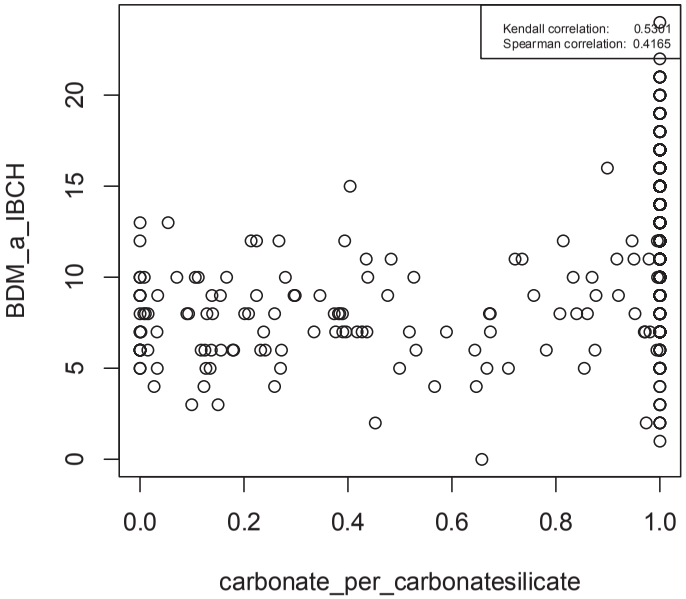
Kendall correlation: 0.0688
Spearman correlation: 0.0458
BDM_a_EPT
0
5
10
15
20
25
30
35
0
5000000
10000000
15000000
watercourse_total_m

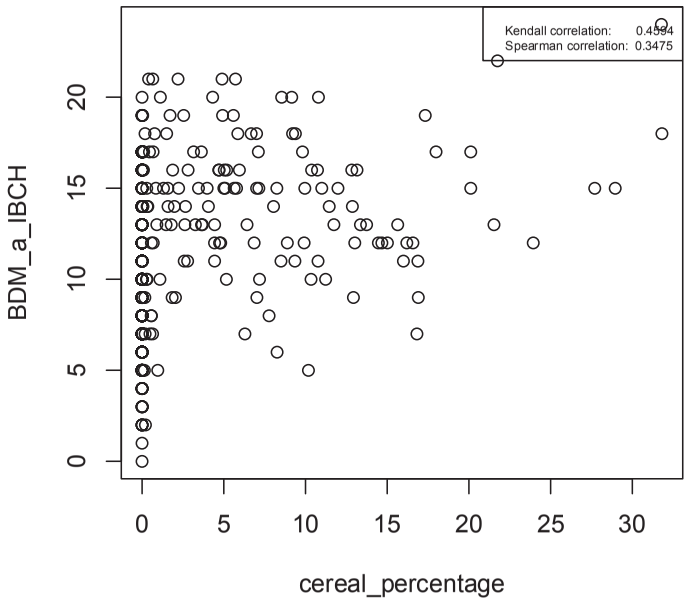
3 Scatterplots: IBCH taxa vs. explanatory variables

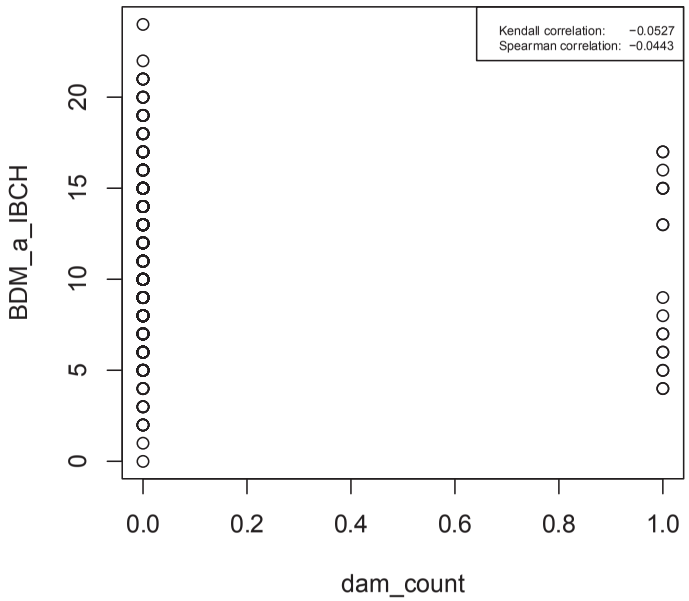


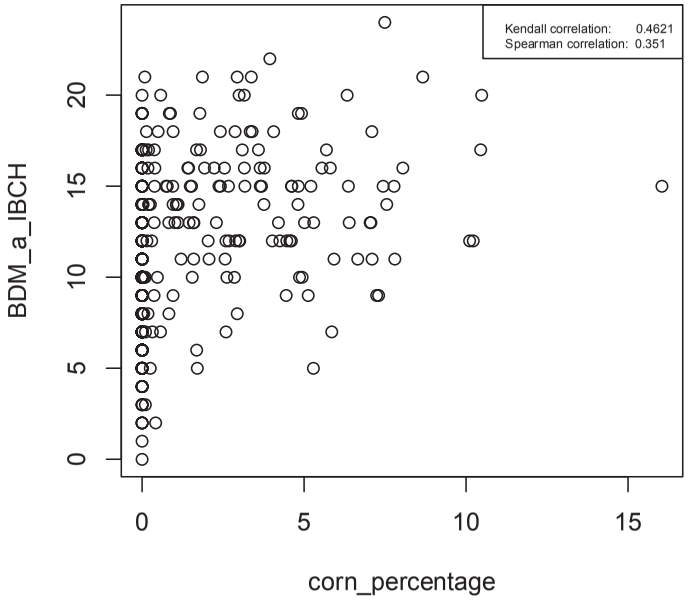


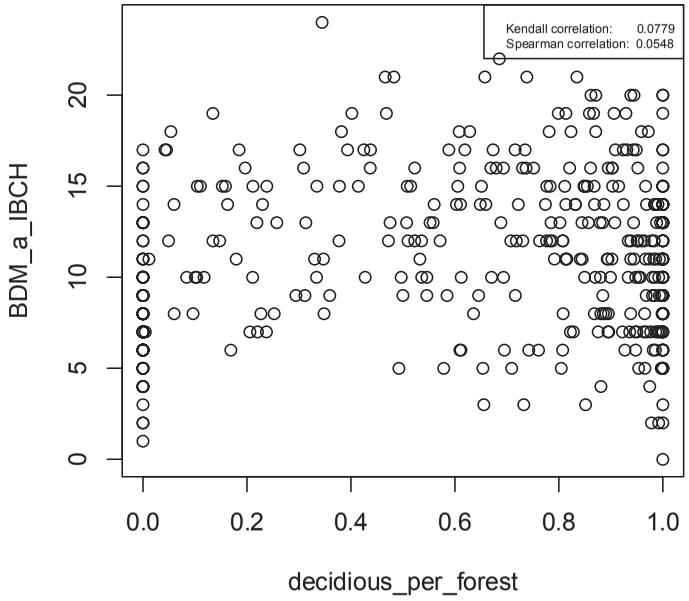


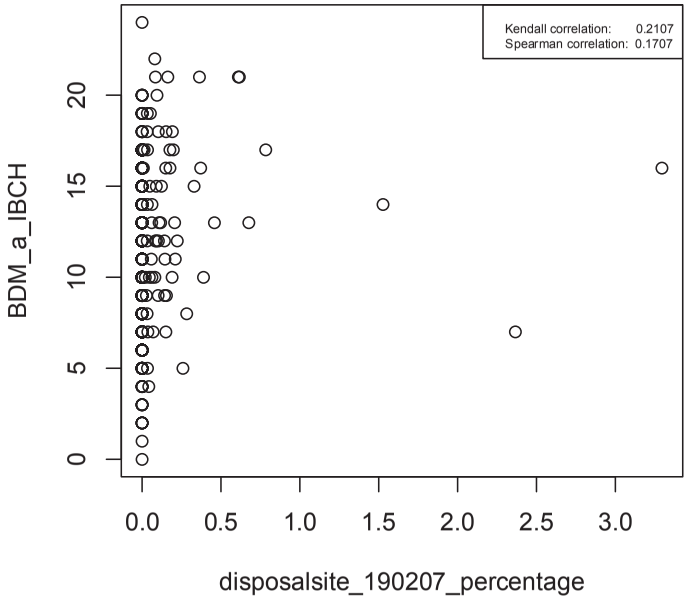


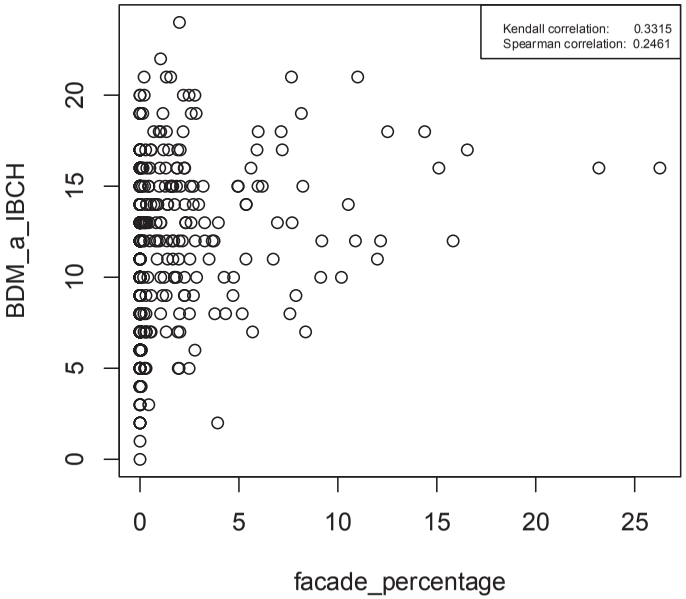

Kendall correlation: 0.4594
Spearman correlation: 0.3475
BDM_a_IBCH
cereal_percentage
0
5
10
15
20
25
30

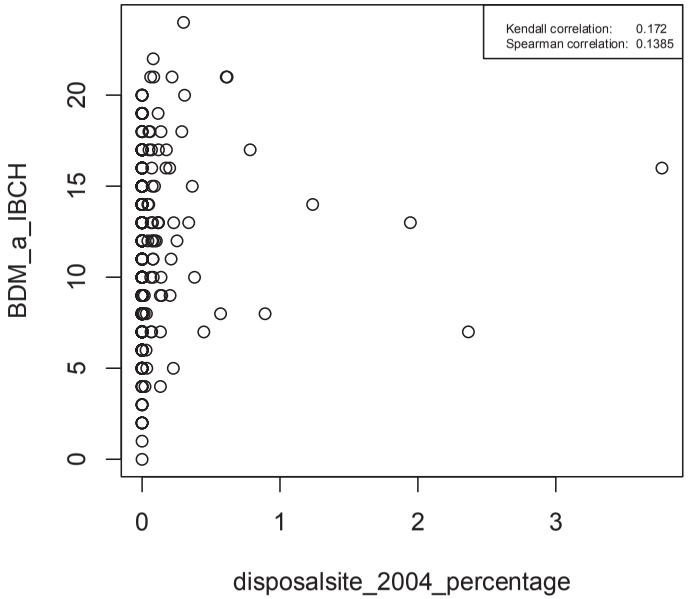

Kendall correlation: −0.0527
Spearman correlation: −0.0443
BDM_a_IBCH
dam_count
0.0
0.2
0.4
0.6
0.8
1.0

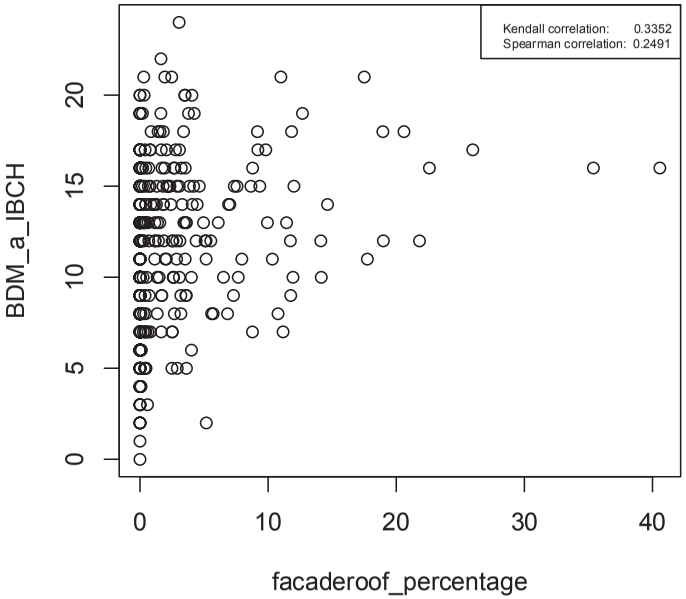

Kendall correlation: 0.4621
Spearman correlation: 0.351
BDM_a_IBCH
corn_percentage

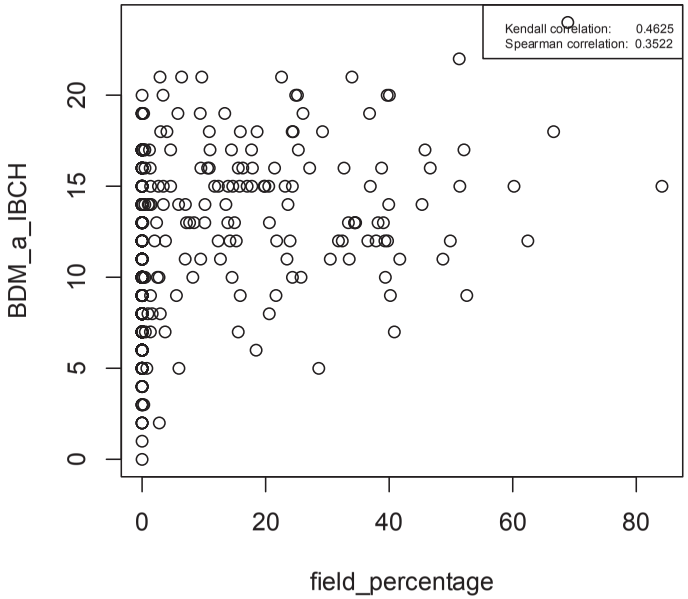

Kendall correlation: 0.0779
Spearman correlation: 0.0548
BDM_a_IBCH
decidious_per_forest

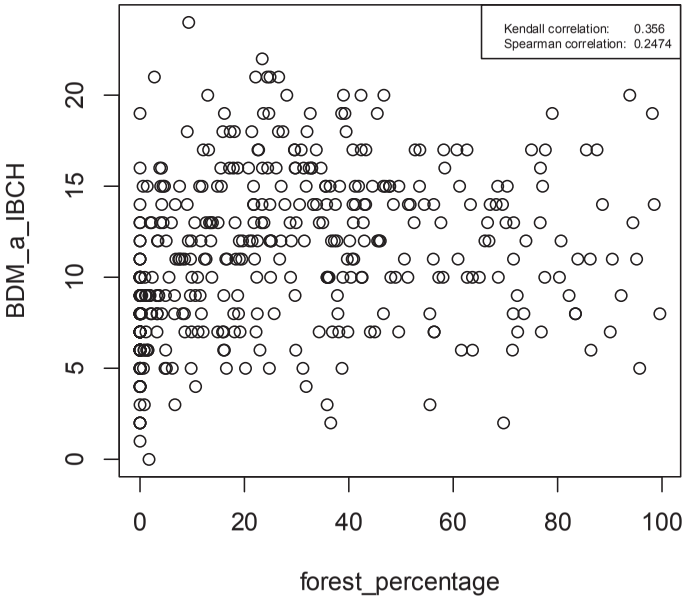

Kendall correlation: 0.2107
Spearman correlation: 0.1707
BDM_a_IBCH
disposalsite_190207_percentage

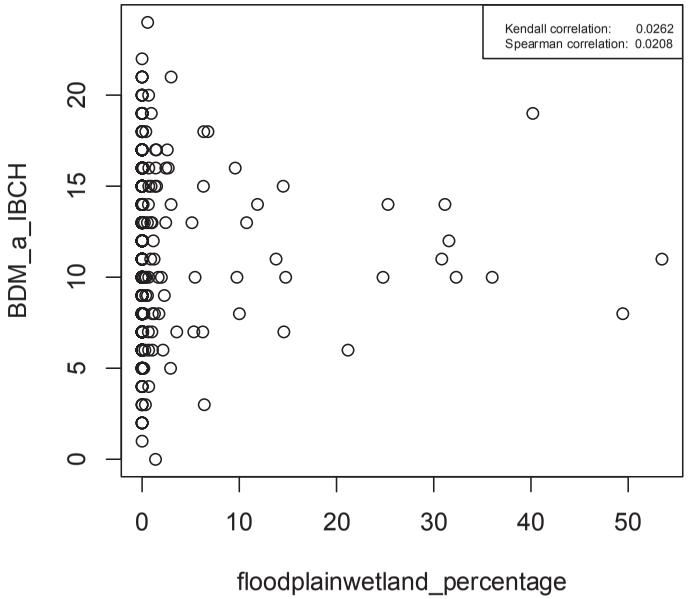

Kendall correlation: 0.3315
Spearman correlation: 0.2461
BDM_a_IBCH
facade_percentage

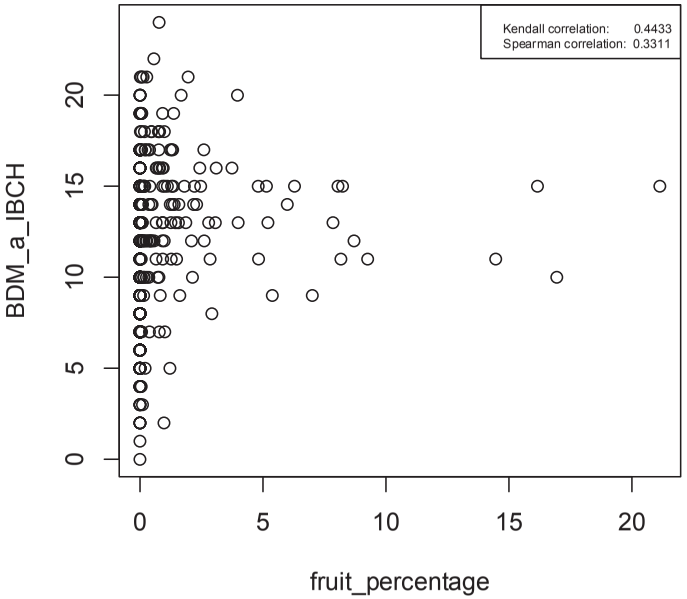

Kendall correlation: 0.172
Spearman correlation: 0.1385
BDM_a_IBCH
disposalsite_2004_percentage

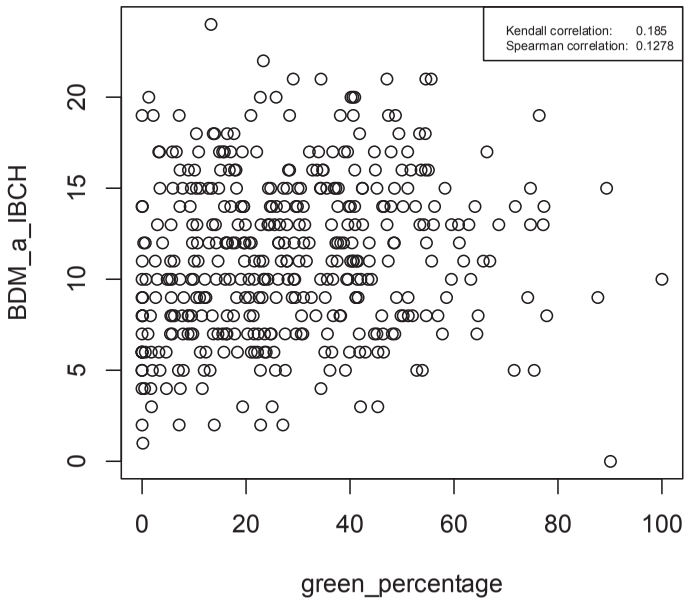

Kendall correlation: 0.3352
Spearman correlation: 0.2491
BDM_a_IBCH
facaderoof_percentage

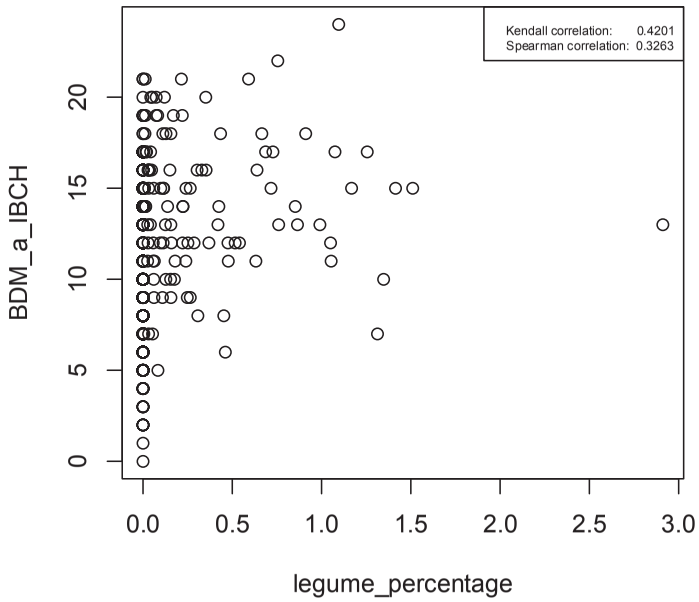

Kendall correlation: 0.4625
Spearman correlation: 0.3522
BDM_a_IBCH
field_percentage
0
5
10
15
20
0
20
40
60
80

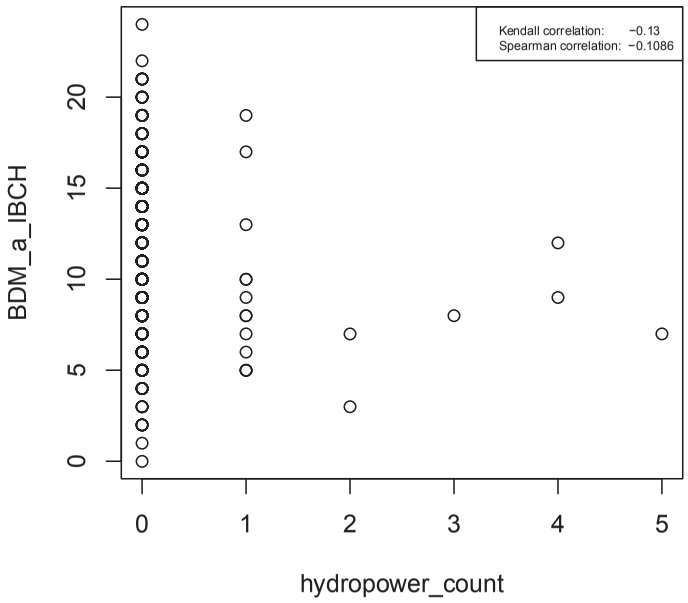

Kendall correlation: 0.356
Spearman correlation: 0.2474
BDM_a_IBCH
forest_percentage
0
5
10
15
20
0
20
40
60
80
100

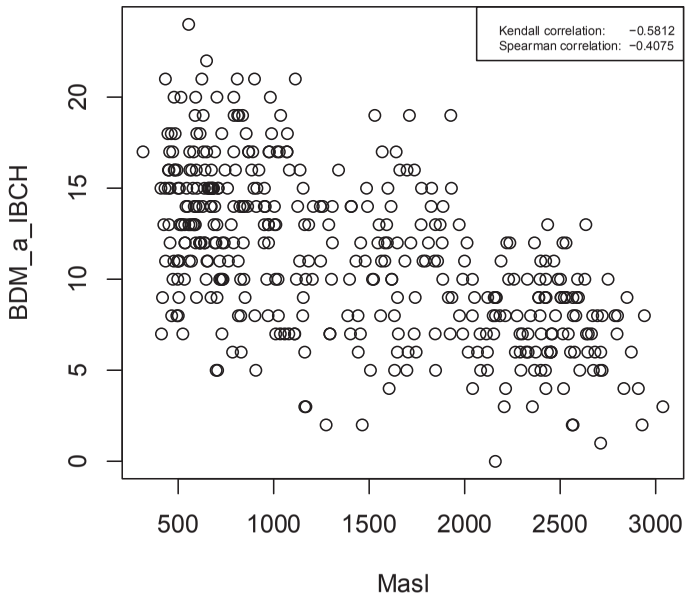

Kendall correlation: 0.0262
Spearman correlation: 0.0208
BDM_a_IBCH
floodplainwetland_percentage
0
5
10
15
20
0
10
20
30
40
50

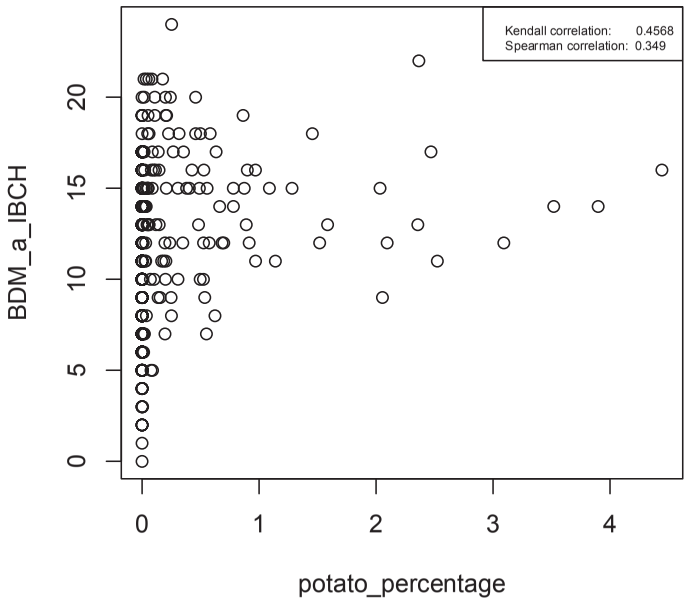

Kendall correlation: 0.4433
Spearman correlation: 0.3311
BDM_a_IBCH
fruit_percentage
0
5
10
15
20
0
5
10
15
20

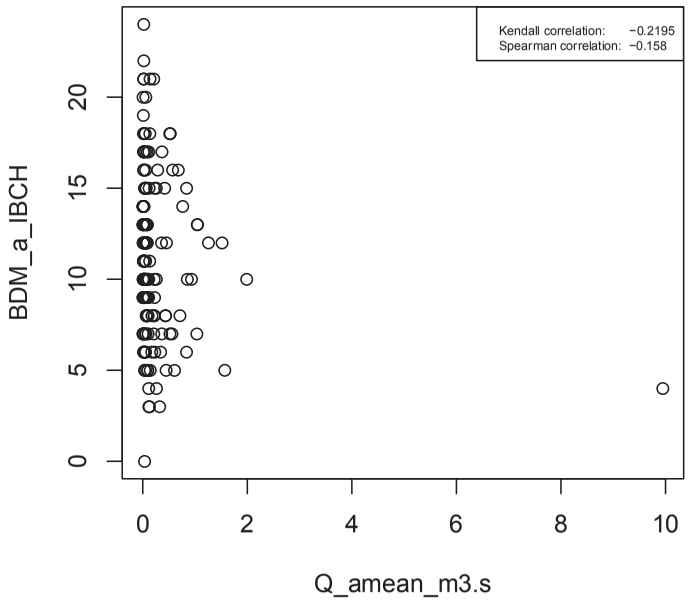

Kendall correlation: 0.185
Spearman correlation: 0.1278
BDM_a_IBCH
green_percentage
0
5
10
15
20
0
20
40
60
80
100

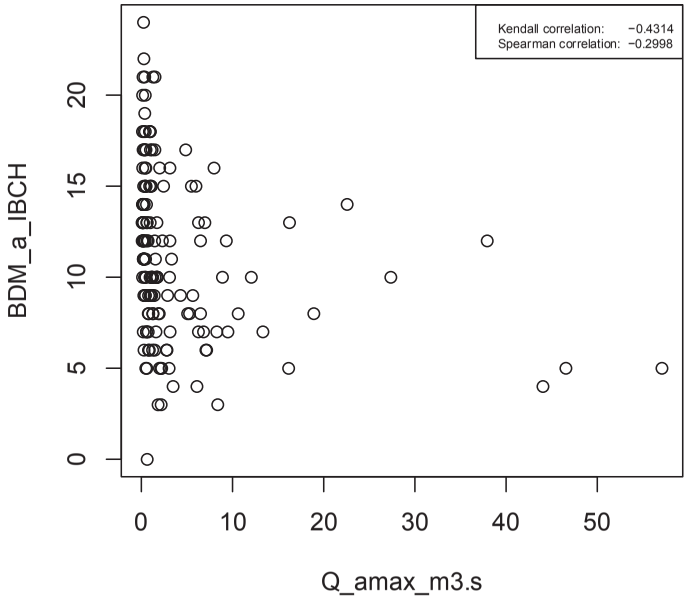

Kendall correlation: 0.4201
Spearman correlation: 0.3263
BDM_a_IBCH
legume_percentage
0
5
10
15
20
0.0
0.5
1.0
1.5
2.0
2.5
3.0

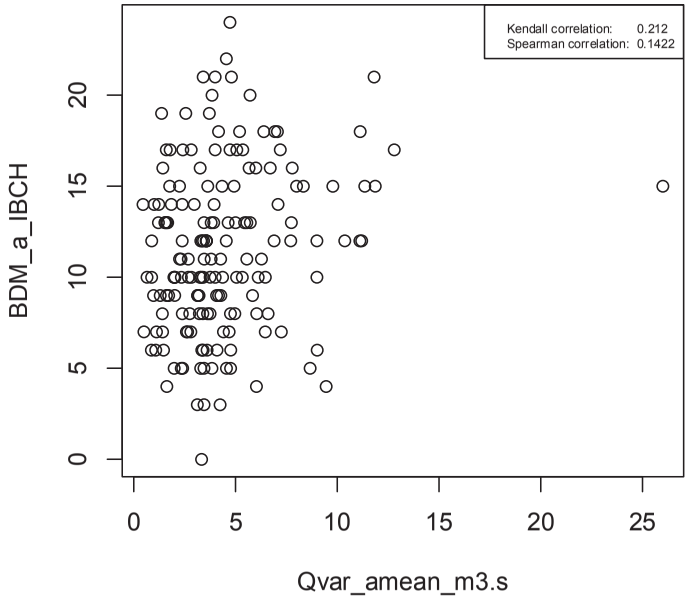

Kendall correlation: −0.13
Spearman correlation: −0.1086
BDM_a_IBCH
hydropower_count
0
5
10
15
20
0
1
2
3
4
5

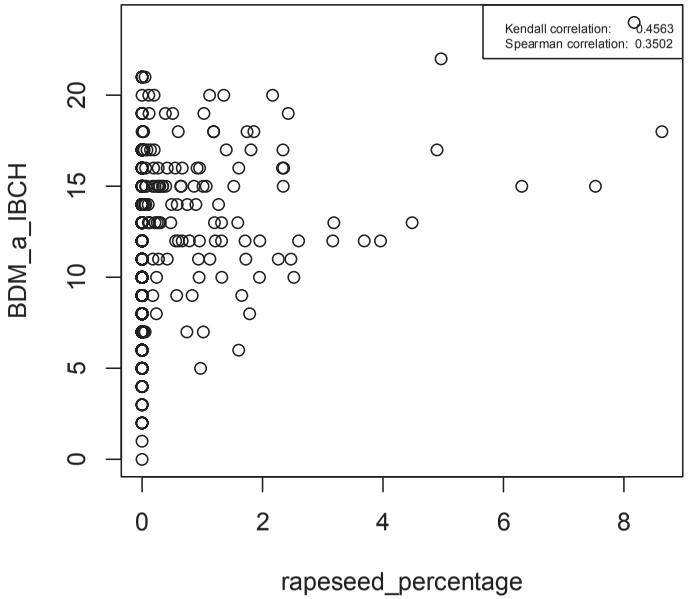

Kendall correlation: −0.5812
Spearman correlation: −0.4075
BDM_a_IBCH
MasI
0
5
10
15
20
500
1000
1500
2000
2500
3000

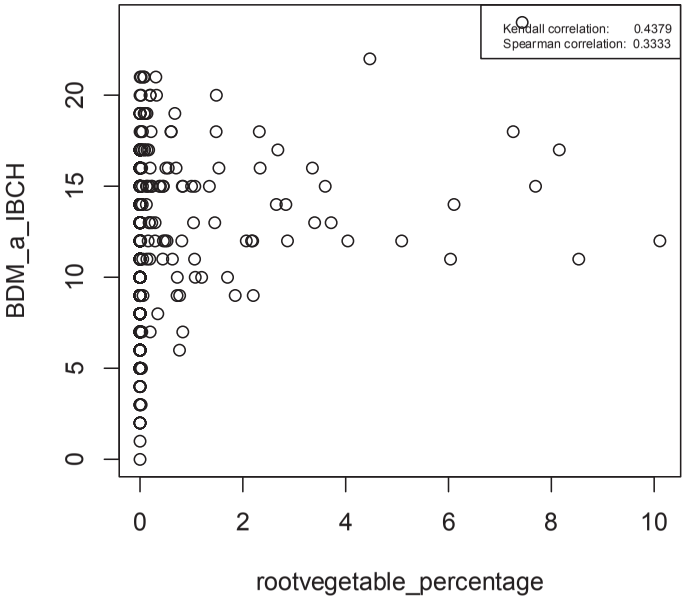

Kendall correlation: 0.4568
Spearman correlation: 0.349
BDM_a_IBCH
potato_percentage
0
5
10
15
20
0
1
2
3
4

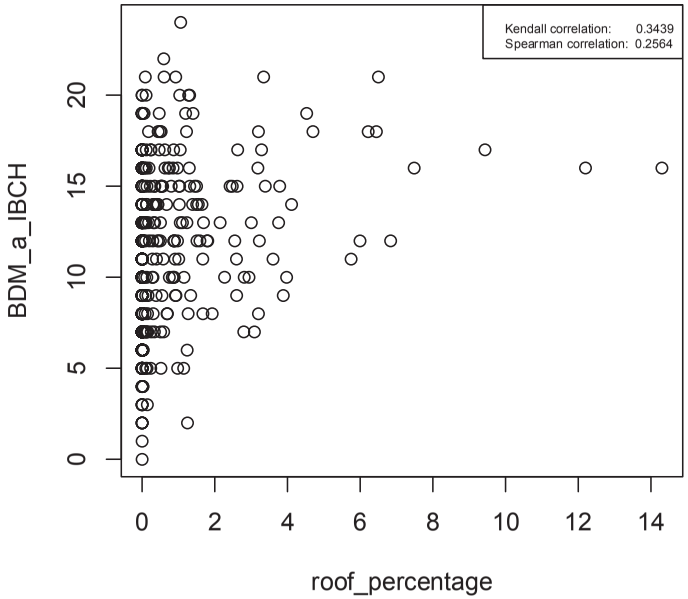

Kendall correlation: −0.2195
Spearman correlation: −0.158
BDM_a_IBCH
Q_amean_m3.s
0
5
10
15
20
0
2
4
6
8
10

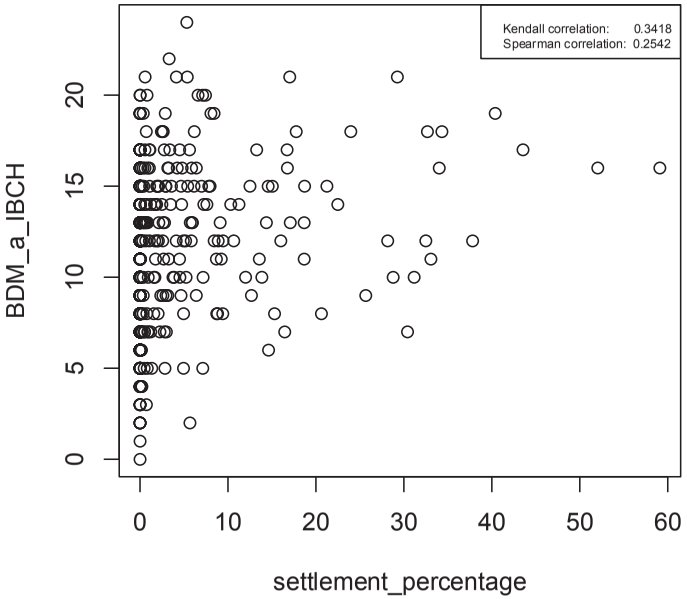

Kendall correlation: −0.4314
Spearman correlation: −0.2998
BDM_a_IBCH
Q_amax_m3.s
0
5
10
15
20
0
10
20
30
40
50

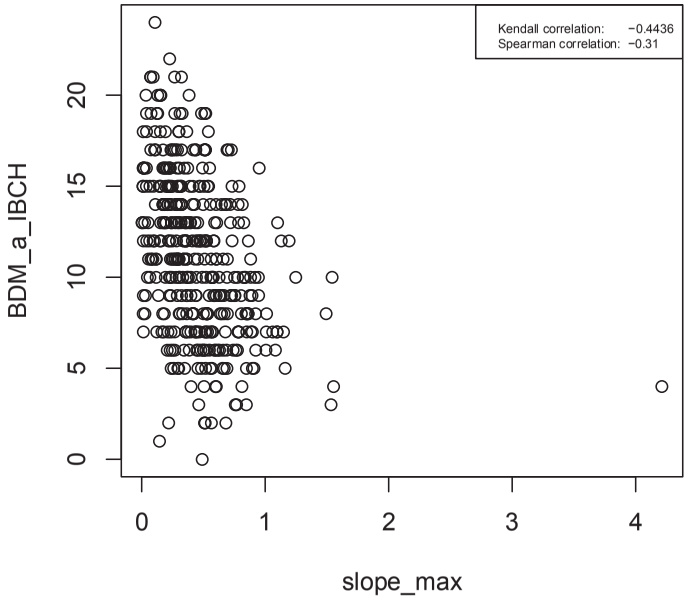

Kendall correlation: 0.212
Spearman correlation: 0.1422
BDM_a_IBCH
Qvar_amean_m3.s
0
5
10
15
20
0
5
10
15
20
25

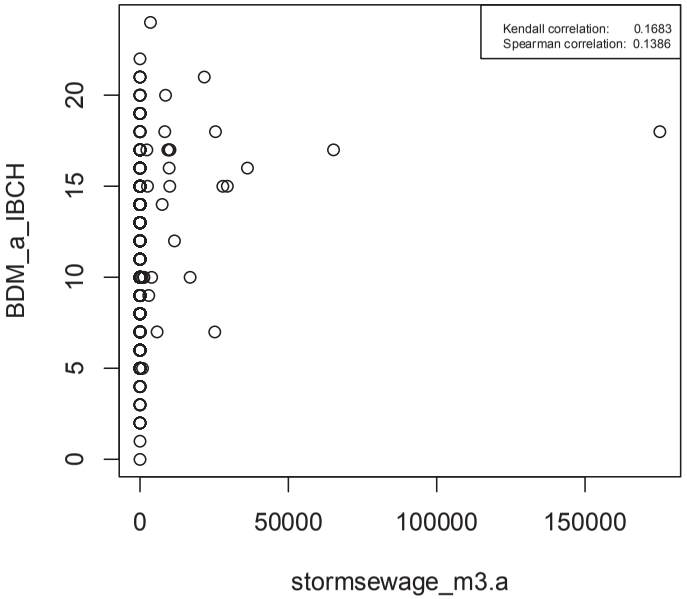

Kendall correlation: 0.4563
Spearman correlation: 0.3502
BDM_a_IBCH
rapeseed_percentage
0
2
4
6
8
0
5
10
15
20

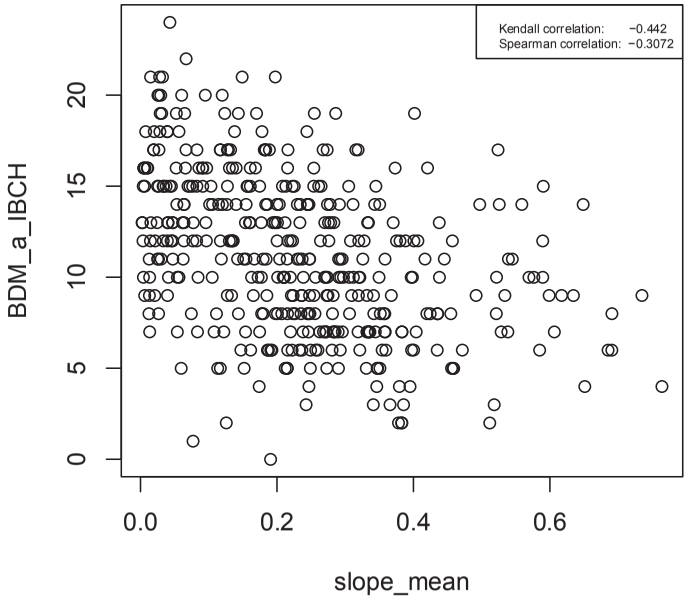

Kendall correlation: 0.4379
Spearman correlation: 0.3333
BDM_a_IBCH
rootvegetable_percentage
0
2
4
6
8
10
0
5
10
15
20

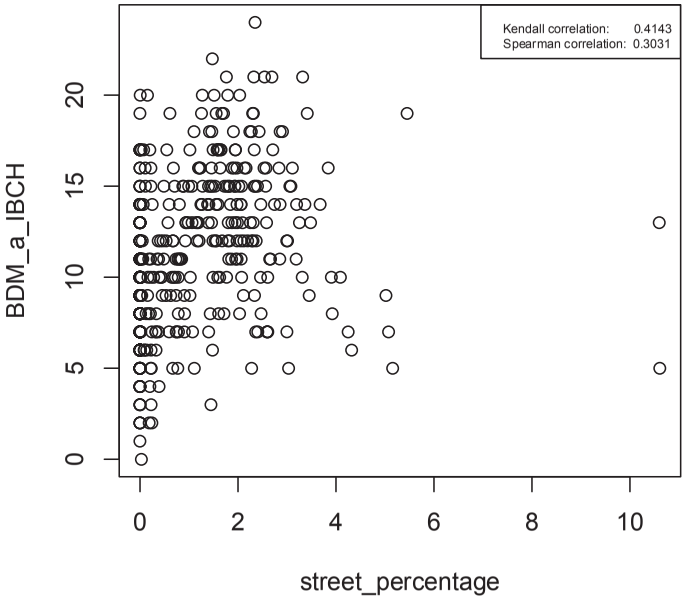

Kendall correlation: 0.3439
Spearman correlation: 0.2564
BDM_a_IBCH
roof_percentage
0
2
4
6
8
10
12
14
0
5
10
15
20

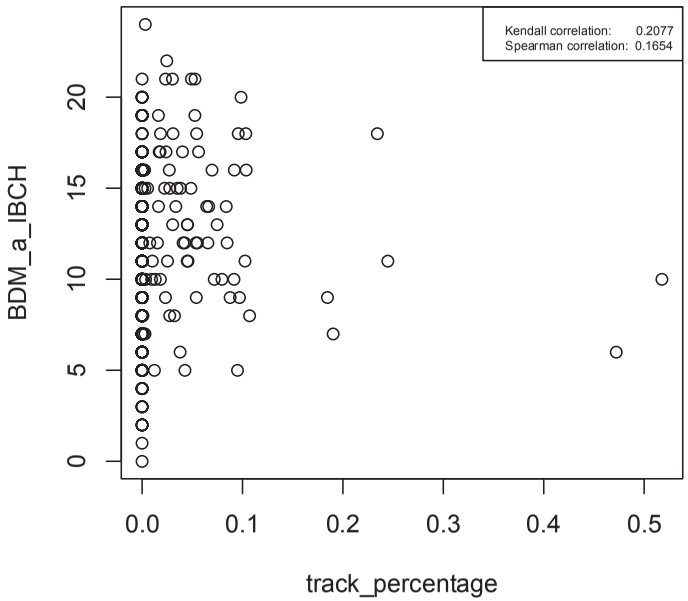

Kendall correlation: 0.3418
Spearman correlation: 0.2542
BDM_a_IBCH
settlement_percentage
0
10
20
30
40
50
60
0
5
10
15
20

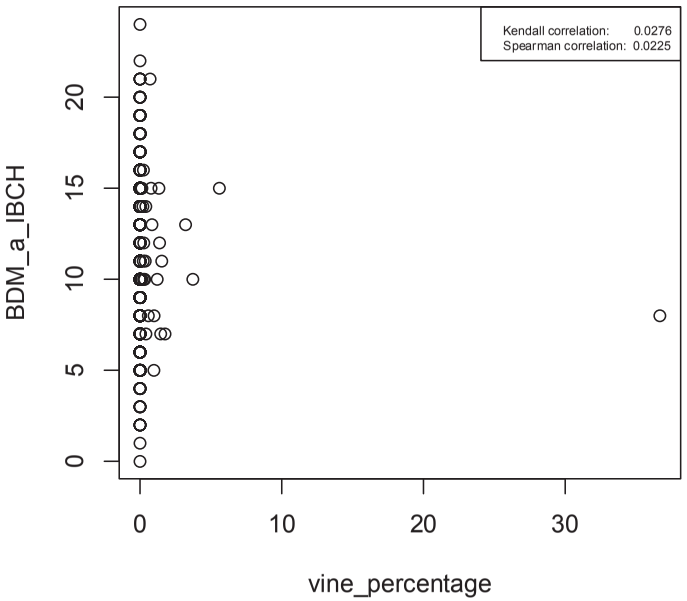

Kendall correlation: −0.4436
Spearman correlation: −0.31
BDM_a_IBCH
slope_max
0
5
10
15
20
0
1
2
3
4

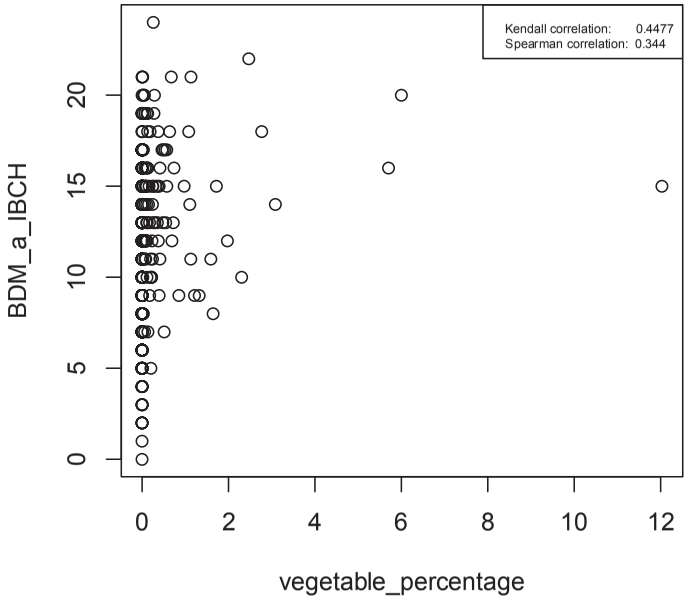

Kendall correlation: 0.1683
Spearman correlation: 0.1386
BDM_a_IBCH
stormsewage_m3.a
0
5
10
15
20
0
50000
100000
150000

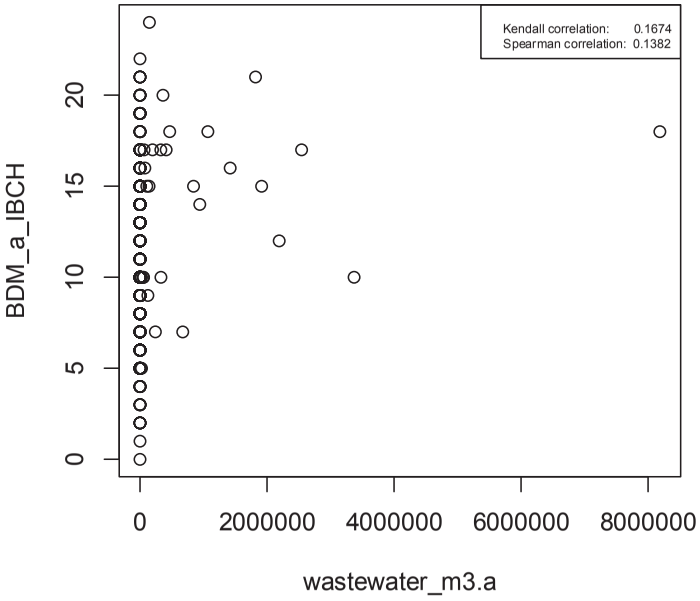

Kendall correlation: −0.442
Spearman correlation: −0.3072
BDM_a_IBCH
slope_mean
0
5
10
15
20
0.0
0.2
0.4
0.6

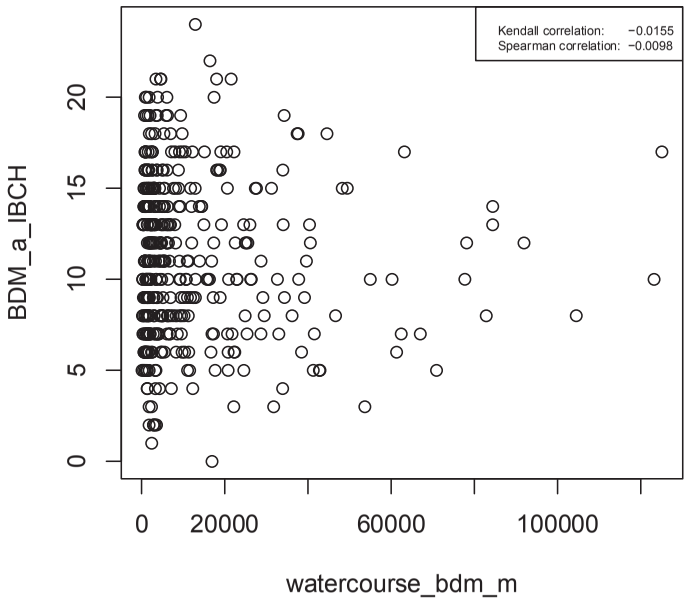

Kendall correlation: 0.4143
Spearman correlation: 0.3031
BDM_a_IBCH
street_percentage
0
5
10
15
20
0
2
4
6
8
10

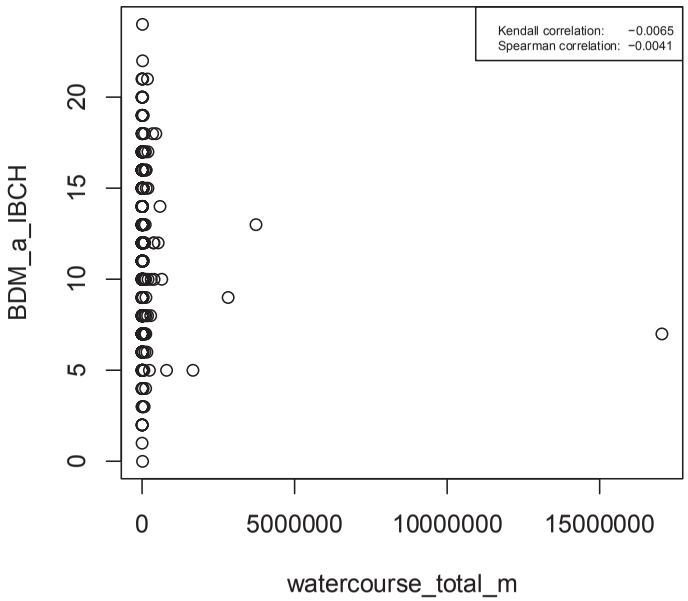

Kendall correlation: 0.2077
Spearman correlation: 0.1654
BDM_a_IBCH
track_percentage
0
5
10
15
20
0.0
0.1
0.2
0.3
0.4
0.5


Kendall correlation: 0.0276
Spearman correlation: 0.0225
BDM_a_IBCH
vine_percentage
0
5
10
15
20
0
10
20
30


Kendall correlation: 0.4477
Spearman correlation: 0.344
BDM_a_IBCH
vegetable_percentage
0
5
10
15
20
0
2
4
6
8
10
12


Kendall correlation: 0.1674
Spearman correlation: 0.1382
BDM_a_IBCH
wastewater_m3.a
0
5
10
15
20
0
2000000
4000000
6000000
8000000


Kendall correlation: −0.0155
Spearman correlation: −0.0098
BDM_a_IBCH
0
5
10
15
20
0
20000
60000
100000
watercourse_bdm_m


Kendall correlation: −0.0065
Spearman correlation: −0.0041
BDM_a_IBCH
0
5
10
15
20
0
5000000
10000000
15000000
watercourse_total_m

4 NA and zero values for explanatory varibles of BDM study area

```
> Join_Final != NA
                      BDM_RowID                       area_bdm_m2
                            410                               410
              potato_percentage                  cereal_percentage
                            410                               410
       rootvegetable_percentage             vegetable_percentage
                            410                               410
              facade_percentage                   roof_percentage
                            410                               410
               track_percentage                  street_percentage
                            410                               410
                      slope_max                       Q_amax_m3.s
                            410                               160
                    hydro_class                      biogeo_class
                            409                               410
          decidious_per_forest                             Masl
                            410                               410
   disposalsite_2004_percentage
                            410
> Join_Final != Zero
                      BDM_RowID                       area_bdm_m2
                            409                               410
              potato_percentage                  cereal_percentage
                            144                               150
       rootvegetable_percentage             vegetable_percentage
                            129                               142
              facade_percentage                   roof_percentage
                            204                               204
               track_percentage                  street_percentage
                             82                               273
                      slope_max                       Q_amax_m3.s
                            410                                NA
                    hydro_class                      biogeo_class
                             NA                               410
          decidious_per_forest                             Masl
                            336                               410
   disposalsite_2004_percentage
                             68
```

```
                watercourse_bdm_m                 field_percentage
                              410                               410
                corn_percentage                 fruit_percentage
                              410                               410
              vine_percentage                 forest_percentage
                              410                               410
          facaderoof_percentage               settlement_percentage
                              410                               410
                Q_amean_m3.s                    Qvar_amean_m3.s
                              160                               166
                   dam_count                    hydropower_count
                              410                               410
floodplainwetland_percentage                       area_total_m2
                              410                               410
          wastewater_m3.a                    stormsewage_m3.a
                              410                               410
```

```
                watercourse_bdm_m                 field_percentage
                              410                               156
                corn_percentage                 fruit_percentage
                              152                               151
              vine_percentage                 forest_percentage
                               39                               347
          facaderoof_percentage               settlement_percentage
                              204                               208
                Q_amean_m3.s                    Qvar_amean_m3.s
                               NA                                NA
                   dam_count                    hydropower_count
                               18                                19
floodplainwetland_percentage                       area_total_m2
                               88                               410
          wastewater_m3.a                    stormsewage_m3.a
                               29                                29
```

```
                legume_percentage
                              410
                rapeseed_percentage
                              410
                green_percentage
                              410
  disposalsite_190207_percentage
                              410
                      slope_mean
                              410
carbonate_per_carbonatesilicate
                              410
             watercourse_total_m
                              410
                canal_percentage
                              410
```

```
                legume_percentage
                              121
                rapeseed_percentage
                              129
                green_percentage
                              396
  disposalsite_190207_percentage
                               65
                      slope_mean
                              410
carbonate_per_carbonatesilicate
                              393
             watercourse_total_m
                              410
                canal_percentage
                               12
```

5 NA and zero values for explanatory varibles of nationwide prediction study area

```
> Join_Final_NA
                     AE_2km2_ID                     area_bdm_m2                watercourse_bdm_m                  field_percentage                 legume_percentage 
                           20772                           20772                           20772                           20772                           20772 
              potato_percentage                cereal_percentage                  corn_percentage                 fruit_percentage              rapeseed_percentage 
                           20772                           20772                           20772                           20772                           20772 
       rootvegetable_percentage             vegetable_percentage                 vine_percentage               forest_percentage                green_percentage 
                           20772                           20772                           20772                           20772                           20772 
             facade_percentage                  roof_percentage          facaderoof_percentage           settlement_percentage disposalsite_190207_percentage 
                           20772                           20772                           20772                           20772                           20772 
               wastewater_m3.a                stormsewage_m3.a                 track_percentage                 street_percentage                     Q_amean_m3.s 
                           20772                           20772                           20772                           20772                            8696 
              Qvar_amean_m3.s                      slope_mean                       slope_max                     Q_amax_m3.s carbonate_per_carbonatesilicate 
                            9132                           20772                           20772                            8563                           20745 
          decidious_per_forest                           Masl                  Area_AE2km2_m2      disposalsite_2004_percentage 
                           20772                           20772                           20772                           20772 
> Join_Final_NotZero
                     AE_2km2_ID                     area_bdm_m2                watercourse_bdm_m                  field_percentage                 legume_percentage 
                           20772                           20772                           20772                            7907                            6083 
              potato_percentage                cereal_percentage                  corn_percentage                 fruit_percentage              rapeseed_percentage 
                            7343                            7556                            7815                            6573                            6308 
       rootvegetable_percentage             vegetable_percentage                 vine_percentage               forest_percentage                green_percentage 
                            6453                            7204                            1412                           17586                           19837 
             facade_percentage                  roof_percentage          facaderoof_percentage           settlement_percentage disposalsite_190207_percentage 
                            7556                            7815                           11080                           11284                            1587 
               wastewater_m3.a                stormsewage_m3.a                 track_percentage                 street_percentage                     Q_amean_m3.s 
                             672                            2100                            4338                           15044                              NA 
              Qvar_amean_m3.s                      slope_mean                       slope_max                     Q_amax_m3.s carbonate_per_carbonatesilicate 
                              NA                           20771                           20769                              NA                              NA 
          decidious_per_forest                           Masl                  Area_AE2km2_m2      disposalsite_2004_percentage 
                           16473                           20772                           20772                            1856 
```

6 Normal distribution test for the explanatory variables

	Value of the Shapiro-Wilk statistic	
	Approximate p-value for the test	
0	0.663	area_bdm_m2
0	0.061	area_total_m2
0	0.109	canal_percentage
0	0.596	carbonate_per_carbonatesilicate
0	0.546	cereal_percentage
0	0.583	corn_percentage
0	0.206	dam_count
0	0.82	deciduous_per_forest
0	0.171	disposalsite_190207_percentage
0	0.181	disposalsite_2004_percentage
0	0.508	facade_percentage
0	0.498	facaderoof_percentage
0	0.594	field_percentage
0	0.26	floodplainwetland_percentage
0	0.896	forest_percentage
0	0.347	fruit_percentage
0	0.959	green_percentage
0	0.173	hydropower_count
0	0.386	legume_percentage
0	0.909	Masl
0	0.359	potato_percentage
0	0.451	Q_amax_m3.s
0	0.247	Q_amean_m3.s
0	0.808	Qvar_amean_m3.s
0	0.405	rapeseed_percentage
0	0.471	root_percentage
0	0.332	rootvegetable_percentage
0	0.516	settlement_percentage
0	0.789	slope_max
0	0.951	slope_mean
0	0.11	stormsewage_m3.a
0	0.762	street_percentage
0	0.295	track_percentage
0	0.204	vegetable_percentage
0	0.057	vine_percentage
0	0.121	wastewater_m3.a
0	0.619	watercourse_bdm_m
0	0.067	watercourse_total_m

7 Kendall correlation analysis for the explanatory variables

kendall	area_bdm_m2	area_total_m2	canal_percentage	carbonate_per_carbonatesilicate	cereal_percentage	corn_percentage	dam_count	decidious_per_forest	disposalsite_190207_percentage	disposalsite_2004_percentage	facade_percentage	facaderoof_percentage	field_percentage	floodplainwetland_percentage	forest_percentage	fruit_percentage	green_percentage	hydropower_count	legume_percentage	Masl	potato_percentage	Q_amax_m3.s	Q_amean_m3.s	Qvar_amean_m3.s	rapeseed_percentage	roof_percentage	rootvegetable_percentage	settlement_percentage	slope_max	slope_mean	stormsewage_m3.a	street_percentage	track_percentage	vegetable_percentage	vine_percentage	wastewater_m3.a	watercourse_bdm_m	watercourse_total_m
area_bdm_m2	1.00	0.97	0.14	-0.10	0.17	0.17	0.23	0.00	0.32	0.34	0.27	0.27	0.18	0.17	-0.10	0.18	-0.02	0.25	0.15	0.01	0.16	0.54	0.50	0.13	0.16	0.28	0.17	0.26	0.18	-0.13	0.29	0.13	0.28	0.17	0.17	0.29	0.77	0.77
area_total_m2	0.97	1.00	0.15	-0.09	0.18	0.18	0.23	-0.01	0.33	0.36	0.28	0.29	0.19	0.17	-0.09	0.19	-0.02	0.26	0.16	0.00	0.17	0.58	0.59	0.19	0.17	0.29	0.18	0.28	0.18	-0.13	0.30	0.14	0.29	0.18	0.17	0.30	0.76	0.79
canal_percentage	0.14	0.15	1.00	0.10	0.19	0.21	-0.04	-0.04	0.19	0.21	0.17	0.17	0.20	0.14	0.01	0.19	-0.02	-0.04	0.21	-0.17	0.18	-0.06	0.13	0.27	0.21	0.18	0.22	0.17	-0.16	-0.19	0.34	0.14	0.15	0.20	0.23	0.34	0.11	0.12
carbonate_per_carbonatesilicate	-0.10	-0.09	0.10	1.00	0.38	0.38	-0.06	0.15	0.18	0.15	0.27	0.27	0.37	0.07	0.29	0.38	0.25	-0.06	0.33	-0.48	0.37	-0.28	-0.14	0.17	0.36	0.28	0.36	0.28	-0.35	-0.34	0.03	0.41	0.17	0.36	0.04	0.03	-0.06	-0.06
cereal_percentage	0.17	0.18	0.19	0.38	1.00	0.92	-0.01	-0.09	0.32	0.35	0.53	0.54	0.95	-0.07	0.15	0.66	-0.03	-0.01	0.83	-0.61	0.88	-0.34	-0.13	0.30	0.88	0.55	0.85	0.54	-0.44	-0.55	0.21	0.53	0.37	0.83	0.29	0.21	0.13	0.15
corn_percentage	0.17	0.18	0.21	0.38	0.92	1.00	-0.01	-0.08	0.33	0.35	0.54	0.55	0.94	-0.07	0.14	0.68	-0.01	-0.01	0.79	-0.62	0.85	-0.32	-0.12	0.31	0.83	0.56	0.83	0.55	-0.44	-0.56	0.20	0.54	0.37	0.85	0.29	0.20	0.14	0.15
dam_count	0.23	0.23	-0.04	-0.06	-0.01	-0.01	1.00	0.05	0.09	0.17	0.11	0.11	0.00	0.14	-0.01	0.02	-0.06	0.23	-0.05	0.03	-0.01	0.30	0.24	-0.05	-0.02	0.11	-0.04	0.10	0.17	0.03	0.03	0.04	0.14	0.00	0.01	0.03	0.21	0.22
decidious_per_forest	0.00	-0.01	-0.04	0.15	-0.09	-0.08	0.05	1.00	0.00	-0.03	-0.04	-0.04	-0.08	0.06	0.17	-0.04	0.14	0.06	-0.10	0.02	-0.08	0.10	0.04	-0.08	-0.10	-0.04	-0.08	-0.05	0.05	0.04	-0.06	0.04	-0.06	-0.09	-0.10	-0.05	0.04	0.04
disposalsite_190207_percentage	0.32	0.33	0.19	0.18	0.32	0.33	0.09	0.00	1.00	0.76	0.36	0.36	0.34	0.13	0.04	0.36	0.01	0.19	0.29	-0.28	0.31	-0.03	0.07	0.17	0.30	0.37	0.33	0.36	-0.14	-0.30	0.32	0.32	0.46	0.32	0.28	0.32	0.31	0.32
disposalsite_2004_percentage	0.34	0.36	0.21	0.15	0.35	0.35	0.17	-0.03	0.76	1.00	0.38	0.38	0.36	0.15	0.04	0.36	-0.03	0.14	0.31	-0.27	0.33	0.03	0.14	0.24	0.32	0.40	0.34	0.39	-0.13	-0.29	0.38	0.32	0.48	0.33	0.29	0.38	0.32	0.35
facade_percentage	0.27	0.28	0.17	0.27	0.53	0.54	0.11	-0.04	0.36	0.38	1.00	0.99	0.54	0.06	0.16	0.58	-0.02	0.10	0.48	-0.49	0.52	-0.14	0.03	0.32	0.51	0.97	0.48	0.95	-0.21	-0.36	0.25	0.53	0.48	0.54	0.34	0.25	0.25	0.26
facaderoof_percentage	0.27	0.29	0.17	0.27	0.54	0.55	0.11	-0.04	0.36	0.38	0.99	1.00	0.55	0.06	0.15	0.58	-0.02	0.10	0.49	-0.49	0.53	-0.14	0.03	0.32	0.51	0.98	0.49	0.96	-0.22	-0.36	0.26	0.53	0.48	0.55	0.34	0.25	0.25	0.26
field_percentage	0.18	0.19	0.20	0.37	0.95	0.94	0.00	-0.08	0.34	0.36	0.54	0.55	1.00	-0.06	0.15	0.66	-0.02	-0.01	0.80	-0.60	0.87	-0.32	-0.11	0.31	0.85	0.56	0.83	0.55	-0.41	-0.54	0.22	0.54	0.37	0.83	0.31	0.22	0.15	0.16
floodplainwetland_percentage	0.17	0.17	0.14	0.07	-0.07	-0.07	0.14	0.06	0.13	0.15	0.06	0.06	-0.06	1.00	0.01	-0.02	0.10	0.18	-0.05	0.01	-0.07	0.15	0.09	-0.11	-0.06	0.06	-0.06	0.06	0.04	-0.08	0.11	0.03	0.12	-0.06	0.05	0.11	0.22	0.22
forest_percentage	-0.10	-0.09	0.01	0.29	0.15	0.14	-0.01	0.17	0.04	0.04	0.16	0.15	0.15	0.01	1.00	0.14	-0.21	0.00	0.16	-0.40	0.16	-0.14	-0.04	0.18	0.15	0.15	0.15	0.16	-0.10	-0.07	0.04	0.29	0.07	0.16	0.08	0.04	-0.05	-0.04
fruit_percentage	0.18	0.19	0.19	0.38	0.66	0.68	0.02	-0.04	0.36	0.36	0.58	0.58	0.66	-0.02	0.14	1.00	0.04	0.00	0.56	-0.58	0.64	-0.33	-0.13	0.27	0.61	0.59	0.61	0.57	-0.36	-0.49	0.19	0.56	0.36	0.68	0.30	0.19	0.16	0.18
green_percentage	-0.02	-0.02	-0.02	0.25	-0.03	-0.01	-0.06	0.14	0.01	-0.03	-0.02	-0.02	-0.02	0.10	-0.21	0.04	1.00	0.01	-0.08	0.03	-0.03	-0.08	-0.07	-0.03	-0.04	-0.02	-0.03	-0.02	-0.04	-0.05	-0.02	0.06	-0.05	-0.05	-0.14	-0.03	0.02	0.01
hydropower_count	0.25	0.26	-0.04	-0.06	-0.01	-0.01	0.23	0.06	0.19	0.14	0.10	0.10	-0.01	0.18	0.00	0.00	0.01	1.00	-0.02	0.05	-0.03	0.27	0.22	-0.02	-0.05	0.10	0.00	0.09	0.18	0.03	0.20	0.03	0.18	-0.03	0.04	0.20	0.24	0.26
legume_percentage	0.15	0.16	0.21	0.33	0.83	0.79	-0.05	-0.10	0.29	0.31	0.48	0.49	0.80	-0.05	0.16	0.56	-0.08	-0.02	1.00	-0.56	0.79	-0.32	-0.15	0.23	0.85	0.51	0.81	0.50	-0.42	-0.51	0.17	0.46	0.34	0.76	0.23	0.17	0.11	0.12
Masl	0.01	0.00	-0.17	-0.48	-0.61	-0.62	0.03	0.02	-0.28	-0.27	-0.49	-0.49	-0.60	0.01	-0.40	-0.58	0.03	0.05	-0.56	1.00	-0.58	0.29	0.12	-0.28	-0.57	-0.51	-0.57	-0.50	0.42	0.45	-0.15	-0.56	-0.33	-0.61	-0.28	-0.15	0.00	-0.02
potato_percentage	0.16	0.17	0.18	0.37	0.88	0.85	-0.01	-0.08	0.31	0.33	0.52	0.53	0.87	-0.07	0.16	0.64	-0.03	-0.03	0.79	-0.58	1.00	-0.32	-0.13	0.26	0.82	0.54	0.84	0.53	-0.42	-0.53	0.19	0.52	0.36	0.84	0.27	0.19	0.13	0.14
Q_amax_m3.s	0.54	0.58	-0.06	-0.28	-0.34	-0.32	0.30	0.10	-0.03	0.03	-0.14	-0.14	-0.32	0.15	-0.14	-0.33	-0.08	0.27	-0.32	0.29	-0.32	1.00	0.67	-0.01	-0.33	-0.14	-0.32	-0.14	0.47	0.30	0.11	-0.27	0.04	-0.32	-0.14	0.11	0.48	0.55
Q_amean_m3.s	0.50	0.59	0.13	-0.14	-0.13	-0.12	0.24	0.04	0.07	0.14	0.03	0.03	-0.11	0.09	-0.04	-0.13	-0.07	0.22	-0.15	0.12	-0.13	0.67	1.00	0.28	-0.13	0.02	-0.11	0.03	0.25	0.11	0.20	-0.11	0.13	-0.13	-0.04	0.20	0.34	0.47
Qvar_amean_m3.s	0.13	0.19	0.27	0.17	0.30	0.31	-0.05	-0.08	0.17	0.24	0.32	0.32	0.31	-0.11	0.18	0.27	-0.03	-0.02	0.23	-0.28	0.26	-0.01	0.28	1.00	0.27	0.32	0.28	0.32	-0.22	-0.26	0.21	0.18	0.19	0.27	0.19	0.20	-0.03	0.08
rapeseed_percentage	0.16	0.17	0.21	0.36	0.88	0.83	-0.02	-0.10	0.30	0.32	0.51	0.51	0.85	-0.06	0.15	0.61	-0.04	-0.05	0.85	-0.57	0.82	-0.33	-0.13	0.27	1.00	0.53	0.84	0.52	-0.43	-0.53	0.21	0.47	0.35	0.80	0.22	0.21	0.12	0.13
roof_percentage	0.28	0.29	0.18	0.28	0.55	0.56	0.11	-0.04	0.37	0.40	0.97	0.98	0.56	0.06	0.15	0.59	-0.02	0.10	0.51	-0.51	0.54	-0.14	0.02	0.32	0.53	1.00	0.51	0.96	-0.23	-0.37	0.26	0.53	0.49	0.56	0.34	0.26	0.25	0.27
rootvegetable_percentage	0.17	0.18	0.22	0.36	0.85	0.83	-0.04	-0.08	0.33	0.34	0.48	0.49	0.83	-0.06	0.15	0.61	-0.03	0.00	0.81	-0.57	0.84	-0.32	-0.11	0.28	0.84	0.51	1.00	0.49	-0.41	-0.53	0.20	0.47	0.36	0.80	0.22	0.20	0.13	0.14
settlement_percentage	0.26	0.28	0.17	0.28	0.54	0.55	0.10	-0.05	0.36	0.39	0.95	0.96	0.55	0.06	0.16	0.57	-0.02	0.09	0.50	-0.50	0.53	-0.14	0.03	0.32	0.52	0.96	0.49	1.00	-0.24	-0.38	0.25	0.53	0.49	0.55	0.33	0.25	0.24	0.26
slope_max	0.18	0.18	-0.16	-0.35	-0.44	-0.44	0.17	0.05	-0.14	-0.13	-0.21	-0.22	-0.41	0.04	-0.10	-0.36	-0.04	0.18	-0.42	0.42	-0.42	0.47	0.25	-0.22	-0.43	-0.23	-0.41	-0.24	1.00	0.62	-0.04	-0.30	-0.12	-0.42	-0.15	-0.04	0.23	0.22
slope_mean	-0.13	-0.13	-0.19	-0.34	-0.55	-0.56	0.03	0.04	-0.30	-0.29	-0.36	-0.36	-0.54	-0.08	-0.07	-0.49	-0.05	0.03	-0.51	0.45	-0.53	0.30	0.11	-0.26	-0.53	-0.37	-0.53	-0.38	0.62	1.00	-0.16	-0.40	-0.24	-0.54	-0.21	-0.16	-0.09	-0.10
stormsewage_m3.a	0.29	0.30	0.34	0.03	0.21	0.20	0.03	-0.06	0.32	0.38	0.25	0.26	0.22	0.11	0.04	0.19	-0.02	0.20	0.17	-0.15	0.19	0.11	0.20	0.21	0.21	0.26	0.20	0.25	-0.04	-0.16	1.00	0.16	0.29	0.20	0.26	0.99	0.27	0.28
street_percentage	0.13	0.14	0.14	0.41	0.53	0.54	0.04	0.04	0.32	0.32	0.53	0.53	0.54	0.03	0.29	0.56	0.06	0.03	0.46	-0.56	0.52	-0.27	-0.11	0.18	0.47	0.53	0.47	0.53	-0.30	-0.40	0.16	1.00	0.34	0.52	0.28	0.16	0.14	0.16
track_percentage	0.28	0.29	0.15	0.17	0.37	0.37	0.14	-0.06	0.46	0.48	0.48	0.48	0.37	0.12	0.07	0.36	-0.05	0.18	0.34	-0.33	0.36	0.04	0.13	0.19	0.35	0.49	0.36	0.49	-0.12	-0.24	0.29	0.34	1.00	0.37	0.28	0.29	0.25	0.27
vegetable_percentage	0.17	0.18	0.20	0.36	0.83	0.85	0.00	-0.09	0.32	0.33	0.54	0.55	0.83	-0.06	0.16	0.68	-0.05	-0.03	0.76	-0.61	0.84	-0.32	-0.13	0.27	0.80	0.56	0.80	0.55	-0.42	-0.54	0.20	0.52	0.37	1.00	0.31	0.20	0.14	0.15
vine_percentage	0.17	0.17	0.23	0.04	0.29	0.29	0.01	-0.10	0.28	0.29	0.34	0.34	0.31	0.05	0.08	0.30	-0.14	0.04	0.23	-0.28	0.27	-0.14	-0.04	0.19	0.22	0.34	0.22	0.33	-0.15	-0.21	0.26	0.28	0.28	0.31	1.00	0.26	0.18	0.18
wastewater_m3.a	0.29	0.30	0.34	0.03	0.21	0.20	0.03	-0.05	0.32	0.38	0.25	0.25	0.22	0.11	0.04	0.19	-0.03	0.20	0.17	-0.15	0.19	0.11	0.20	0.20	0.21	0.26	0.20	0.25	-0.04	-0.16	0.99	0.16	0.29	0.20	0.26	1.00	0.27	0.28
watercourse_bdm_m	0.77	0.76	0.11	-0.06	0.13	0.14	0.21	0.04	0.31	0.32	0.25	0.25	0.15	0.22	-0.05	0.16	0.02	0.24	0.11	0.00	0.13	0.48	0.34	-0.03	0.12	0.25	0.13	0.24	0.23	-0.09	0.27	0.14	0.25	0.14	0.18	0.27	1.00	0.97
watercourse_total_m	0.77	0.79	0.12	-0.06	0.15	0.15	0.22	0.04	0.32	0.35	0.26	0.26	0.16	0.22	-0.04	0.18	0.01	0.26	0.12	-0.02	0.14	0.55	0.47	0.08	0.13	0.27	0.14	0.26	0.22	-0.10	0.28	0.16	0.27	0.15	0.18	0.28	0.97	1.00

8 Spearman correlation analysis for the explanatory variables

spearman	area_bdm_m2	area_total_m2	canal_percentage	carbonate_per_carbonatesilicate	cereal_percentage	corn_percentage	dam_count	decidious_per_forest	disposalsite_190207_percentage	disposalsite_2004_percentage	facade_percentage	facaderoof_percentage	field_percentage	floodplainwetland_percentage	forest_percentage	fruit_percentage	green_percentage	hydropower_count	legume_percentage	Masl	potato_percentage	Q_amax_m3.s	Q_amean_m3.s	Qvar_amean_m3.s	rapeseed_percentage	roof_percentage	rootvegetable_percentage	settlement_percentage	slope_max	slope_mean	stormsewage_m3.a	street_percentage	track_percentage	vegetable_percentage	vine_percentage	wastewater_m3.a	watercourse_bdm_m	watercourse_total_m
area_bdm_m2	1.00	1.00	0.17	-0.13	0.22	0.22	0.28	-0.01	0.42	0.44	0.38	0.38	0.24	0.22	-0.15	0.24	-0.03	0.31	0.20	0.02	0.21	0.72	0.69	0.19	0.21	0.39	0.23	0.37	0.27	-0.19	0.36	0.19	0.36	0.22	0.21	0.36	0.93	0.93
area_total_m2	1.00	1.00	0.18	-0.12	0.24	0.24	0.28	-0.01	0.43	0.46	0.39	0.40	0.26	0.22	-0.15	0.26	-0.03	0.32	0.21	0.00	0.23	0.75	0.76	0.28	0.23	0.41	0.24	0.38	0.26	-0.20	0.37	0.20	0.37	0.24	0.22	0.37	0.93	0.94
canal_percentage	0.17	0.18	1.00	0.11	0.21	0.23	-0.04	-0.05	0.20	0.22	0.19	0.19	0.22	0.15	0.02	0.21	-0.02	-0.04	0.23	-0.21	0.20	-0.08	0.16	0.34	0.23	0.20	0.24	0.20	-0.19	-0.23	0.35	0.16	0.16	0.22	0.24	0.35	0.13	0.14
carbonate_per_carbonatesilicate	-0.13	-0.12	0.11	1.00	0.45	0.45	-0.06	0.20	0.21	0.17	0.33	0.33	0.44	0.07	0.37	0.45	0.33	-0.06	0.39	-0.62	0.44	-0.37	-0.19	0.23	0.42	0.34	0.43	0.34	-0.46	-0.44	0.04	0.50	0.20	0.42	0.05	0.04	-0.08	-0.08
cereal_percentage	0.22	0.24	0.21	0.45	1.00	0.98	-0.01	-0.11	0.37	0.40	0.65	0.66	0.99	-0.08	0.23	0.78	-0.04	-0.02	0.90	-0.77	0.96	-0.46	-0.19	0.41	0.94	0.67	0.92	0.66	-0.57	-0.71	0.23	0.67	0.42	0.93	0.33	0.23	0.18	0.20
corn_percentage	0.22	0.24	0.23	0.45	0.98	1.00	-0.01	-0.10	0.38	0.40	0.66	0.66	0.99	-0.08	0.22	0.80	-0.02	-0.01	0.88	-0.78	0.95	-0.45	-0.16	0.42	0.91	0.68	0.91	0.67	-0.57	-0.72	0.22	0.68	0.42	0.94	0.33	0.22	0.18	0.20
dam_count	0.28	0.28	-0.04	-0.06	-0.01	-0.01	1.00	0.06	0.09	0.18	0.12	0.12	0.00	0.15	-0.01	0.02	-0.07	0.23	-0.05	0.04	-0.01	0.36	0.30	-0.07	-0.02	0.12	-0.04	0.11	0.20	0.04	0.03	0.04	0.14	0.00	0.01	0.03	0.26	0.27
decidious_per_forest	-0.01	-0.01	-0.05	0.20	-0.11	-0.10	0.06	1.00	0.00	-0.03	-0.04	-0.04	-0.10	0.07	0.27	-0.05	0.20	0.07	-0.11	-0.01	-0.11	0.14	0.05	-0.12	-0.12	-0.04	-0.09	-0.05	0.08	0.06	-0.07	0.08	-0.08	-0.11	-0.11	-0.07	0.06	0.05
disposalsite_190207_percentage	0.42	0.43	0.20	0.21	0.37	0.38	0.09	0.00	1.00	0.80	0.41	0.42	0.39	0.15	0.05	0.41	0.01	0.20	0.32	-0.35	0.36	-0.04	0.09	0.22	0.33	0.43	0.38	0.42	-0.17	-0.38	0.33	0.38	0.50	0.36	0.30	0.33	0.39	0.41
disposalsite_2004_percentage	0.44	0.46	0.22	0.17	0.40	0.40	0.18	-0.03	0.80	1.00	0.44	0.44	0.42	0.16	0.06	0.40	-0.03	0.15	0.35	-0.34	0.37	0.03	0.18	0.30	0.36	0.45	0.38	0.44	-0.17	-0.36	0.40	0.38	0.52	0.38	0.30	0.40	0.41	0.44
facade_percentage	0.38	0.39	0.19	0.33	0.65	0.66	0.12	-0.04	0.41	0.44	1.00	1.00	0.66	0.07	0.22	0.69	-0.02	0.11	0.58	-0.64	0.64	-0.24	0.02	0.45	0.61	1.00	0.59	0.99	-0.31	-0.48	0.29	0.66	0.56	0.66	0.39	0.29	0.35	0.37
facaderoof_percentage	0.38	0.40	0.19	0.33	0.66	0.66	0.12	-0.04	0.42	0.44	1.00	1.00	0.67	0.07	0.22	0.69	-0.02	0.11	0.59	-0.64	0.65	-0.24	0.02	0.45	0.62	1.00	0.60	0.99	-0.32	-0.49	0.29	0.66	0.56	0.66	0.39	0.29	0.35	0.37
field_percentage	0.24	0.26	0.22	0.44	0.99	0.99	0.00	-0.10	0.39	0.42	0.66	0.67	1.00	-0.07	0.23	0.79	-0.03	-0.01	0.88	-0.77	0.95	-0.44	-0.15	0.43	0.92	0.69	0.91	0.68	-0.55	-0.70	0.24	0.69	0.42	0.93	0.35	0.24	0.20	0.22
floodplainwetland_percentage	0.22	0.22	0.15	0.07	-0.08	-0.08	0.15	0.07	0.15	0.16	0.07	0.07	-0.07	1.00	0.01	-0.02	0.13	0.19	-0.06	0.01	-0.08	0.20	0.12	-0.14	-0.06	0.07	-0.07	0.07	0.05	-0.10	0.12	0.04	0.13	-0.06	0.05	0.12	0.29	0.28
forest_percentage	-0.15	-0.15	0.02	0.37	0.23	0.22	-0.01	0.27	0.05	0.06	0.22	0.22	0.23	0.01	1.00	0.20	-0.27	0.00	0.22	-0.57	0.23	-0.20	-0.06	0.26	0.21	0.22	0.21	0.22	-0.15	-0.11	0.05	0.43	0.09	0.22	0.10	0.05	-0.08	-0.07
fruit_percentage	0.24	0.26	0.21	0.45	0.78	0.80	0.02	-0.05	0.41	0.40	0.69	0.69	0.79	-0.02	0.20	1.00	0.05	0.00	0.66	-0.73	0.76	-0.46	-0.20	0.38	0.72	0.71	0.72	0.69	-0.49	-0.64	0.21	0.70	0.41	0.79	0.34	0.21	0.22	0.24
green_percentage	-0.03	-0.03	-0.02	0.33	-0.04	-0.02	-0.07	0.20	0.01	-0.03	-0.02	-0.02	-0.03	0.13	-0.27	0.05	1.00	0.02	-0.10	0.04	-0.03	-0.12	-0.10	-0.05	-0.05	-0.02	-0.04	-0.02	-0.06	-0.09	-0.03	0.08	-0.07	-0.06	-0.18	-0.03	0.02	0.01
hydropower_count	0.31	0.32	-0.04	-0.06	-0.02	-0.01	0.23	0.07	0.20	0.15	0.11	0.11	-0.01	0.19	0.00	0.00	0.02	1.00	-0.03	0.06	-0.04	0.33	0.27	-0.03	-0.05	0.11	0.00	0.11	0.22	0.04	0.20	0.04	0.19	-0.03	0.05	0.20	0.29	0.31
legume_percentage	0.20	0.21	0.23	0.39	0.90	0.88	-0.05	-0.11	0.32	0.35	0.58	0.59	0.88	-0.06	0.22	0.66	-0.10	-0.03	1.00	-0.70	0.88	-0.43	-0.20	0.32	0.92	0.61	0.89	0.60	-0.54	-0.65	0.18	0.58	0.39	0.85	0.25	0.18	0.14	0.16
Masl	0.02	0.00	-0.21	-0.62	-0.77	-0.78	0.04	-0.01	-0.35	-0.34	-0.64	-0.64	-0.77	0.01	-0.57	-0.73	0.04	0.06	-0.70	1.00	-0.75	0.45	0.18	-0.42	-0.73	-0.65	-0.72	-0.66	0.61	0.64	-0.19	-0.75	-0.42	-0.77	-0.34	-0.19	0.00	-0.02
potato_percentage	0.21	0.23	0.20	0.44	0.96	0.95	-0.01	-0.11	0.36	0.37	0.64	0.65	0.95	-0.08	0.23	0.76	-0.03	-0.04	0.88	-0.75	1.00	-0.44	-0.18	0.36	0.91	0.66	0.92	0.65	-0.55	-0.69	0.22	0.66	0.42	0.95	0.31	0.21	0.17	0.19
Q_amax_m3.s	0.72	0.75	-0.08	-0.37	-0.46	-0.45	0.36	0.14	-0.04	0.03	-0.24	-0.24	-0.44	0.20	-0.20	-0.46	-0.12	0.33	-0.43	0.45	-0.44	1.00	0.85	-0.01	-0.44	-0.26	-0.43	-0.25	0.65	0.43	0.14	-0.41	0.05	-0.44	-0.19	0.15	0.66	0.72
Q_amean_m3.s	0.69	0.76	0.16	-0.19	-0.19	-0.16	0.30	0.05	0.09	0.18	0.02	0.02	-0.15	0.12	-0.06	-0.20	-0.10	0.27	-0.20	0.18	-0.18	0.85	1.00	0.41	-0.18	0.01	-0.16	0.02	0.36	0.16	0.25	-0.18	0.16	-0.18	-0.05	0.25	0.50	0.63
Qvar_amean_m3.s	0.19	0.28	0.34	0.23	0.41	0.42	-0.07	-0.12	0.22	0.30	0.45	0.45	0.43	-0.14	0.26	0.38	-0.05	-0.03	0.32	-0.42	0.36	-0.01	0.41	1.00	0.37	0.46	0.40	0.46	-0.34	-0.38	0.26	0.27	0.25	0.37	0.24	0.26	-0.04	0.13
rapeseed_percentage	0.21	0.23	0.23	0.42	0.94	0.91	-0.02	-0.12	0.33	0.36	0.61	0.62	0.92	-0.06	0.21	0.72	-0.05	-0.05	0.92	-0.73	0.91	-0.44	-0.18	0.37	1.00	0.63	0.92	0.62	-0.56	-0.68	0.23	0.60	0.40	0.89	0.24	0.23	0.16	0.18
roof_percentage	0.39	0.41	0.20	0.34	0.67	0.68	0.12	-0.04	0.43	0.45	1.00	1.00	0.69	0.07	0.22	0.71	-0.02	0.11	0.61	-0.65	0.66	-0.26	0.01	0.46	0.63	1.00	0.61	0.99	-0.33	-0.51	0.29	0.67	0.56	0.68	0.39	0.29	0.35	0.37
rootvegetable_percentage	0.23	0.24	0.24	0.43	0.92	0.91	-0.04	-0.09	0.38	0.38	0.59	0.60	0.91	-0.07	0.21	0.72	-0.04	0.00	0.89	-0.72	0.92	-0.43	-0.16	0.40	0.92	0.61	1.00	0.60	-0.54	-0.67	0.22	0.60	0.41	0.89	0.25	0.22	0.17	0.19
settlement_percentage	0.37	0.38	0.20	0.34	0.66	0.67	0.11	-0.05	0.42	0.44	0.99	0.99	0.68	0.07	0.22	0.69	-0.02	0.11	0.60	-0.66	0.65	-0.25	0.02	0.46	0.62	0.99	0.60	1.00	-0.34	-0.52	0.29	0.67	0.56	0.66	0.39	0.28	0.33	0.35
slope_max	0.27	0.26	-0.19	-0.46	-0.57	-0.57	0.20	0.08	-0.17	-0.17	-0.31	-0.32	-0.55	0.05	-0.15	-0.49	-0.06	0.22	-0.54	0.61	-0.55	0.65	0.36	-0.34	-0.56	-0.33	-0.54	-0.34	1.00	0.81	-0.05	-0.43	-0.16	-0.55	-0.19	-0.05	0.34	0.32
slope_mean	-0.19	-0.20	-0.23	-0.44	-0.71	-0.72	0.04	0.06	-0.38	-0.36	-0.48	-0.49	-0.70	-0.10	-0.11	-0.64	-0.09	0.04	-0.65	0.64	-0.69	0.43	0.16	-0.38	-0.68	-0.51	-0.67	-0.52	0.81	1.00	-0.20	-0.55	-0.31	-0.69	-0.26	-0.20	-0.14	-0.15
stormsewage_m3.a	0.36	0.37	0.35	0.04	0.23	0.22	0.03	-0.07	0.33	0.40	0.29	0.29	0.24	0.12	0.05	0.21	-0.03	0.20	0.18	-0.19	0.22	0.14	0.25	0.26	0.23	0.29	0.22	0.29	-0.05	-0.20	1.00	0.19	0.31	0.23	0.28	1.00	0.34	0.35
street_percentage	0.19	0.20	0.16	0.50	0.67	0.68	0.04	0.08	0.38	0.38	0.66	0.66	0.69	0.04	0.43	0.70	0.08	0.04	0.58	-0.75	0.66	-0.41	-0.18	0.27	0.60	0.67	0.60	0.67	-0.43	-0.55	0.19	1.00	0.41	0.65	0.33	0.19	0.21	0.22
track_percentage	0.36	0.37	0.16	0.20	0.42	0.42	0.14	-0.08	0.50	0.52	0.56	0.56	0.42	0.13	0.09	0.41	-0.07	0.19	0.39	-0.42	0.42	0.05	0.16	0.25	0.40	0.56	0.41	0.56	-0.16	-0.31	0.31	0.41	1.00	0.42	0.30	0.31	0.32	0.34
vegetable_percentage	0.22	0.24	0.22	0.42	0.93	0.94	0.00	-0.11	0.36	0.38	0.66	0.66	0.93	-0.06	0.22	0.79	-0.06	-0.03	0.85	-0.77	0.95	-0.44	-0.18	0.37	0.89	0.68	0.89	0.66	-0.55	-0.69	0.23	0.65	0.42	1.00	0.35	0.22	0.18	0.20
vine_percentage	0.21	0.22	0.24	0.05	0.33	0.33	0.01	-0.11	0.30	0.30	0.39	0.39	0.35	0.05	0.10	0.34	-0.18	0.05	0.25	-0.34	0.31	-0.19	-0.05	0.24	0.24	0.39	0.25	0.39	-0.19	-0.26	0.28	0.33	0.30	0.35	1.00	0.27	0.22	0.23
wastewater_m3.a	0.36	0.37	0.35	0.04	0.23	0.22	0.03	-0.07	0.33	0.40	0.29	0.29	0.24	0.12	0.05	0.21	-0.03	0.20	0.18	-0.19	0.21	0.15	0.25	0.26	0.23	0.29	0.22	0.28	-0.05	-0.20	1.00	0.19	0.31	0.22	0.27	1.00	0.34	0.35
watercourse_bdm_m	0.93	0.93	0.13	-0.08	0.18	0.18	0.26	0.06	0.39	0.41	0.35	0.35	0.20	0.29	-0.08	0.22	0.02	0.29	0.14	0.00	0.17	0.66	0.50	-0.04	0.16	0.35	0.17	0.33	0.34	-0.14	0.34	0.21	0.32	0.18	0.22	0.34	1.00	0.99
watercourse_total_m	0.93	0.94	0.14	-0.08	0.20	0.20	0.27	0.05	0.41	0.44	0.37	0.37	0.22	0.28	-0.07	0.24	0.01	0.31	0.16	-0.02	0.19	0.72	0.63	0.13	0.18	0.37	0.19	0.35	0.32	-0.15	0.35	0.22	0.34	0.20	0.23	0.35	0.99	1.00

9 Tree model for EPT species

Tree model for correlating explanatory variable group C

```
tree_groupC<-tree(BDM_a_EPT~ area_bdm_m2 + watercourse_bdm_m + area_total_m2 + watercourse_total_m,
data=RV)

1) root 410 22800.0 15.19
  2) watercourse_bdm_m < 74309.6 401 22340.0 15.06 *
  3) watercourse_bdm_m > 74309.6 9   173.6 20.78 *
```

Tree model for correlating explanatory variable group D

```
tree_groupD<-tree(BDM_a_EPT~ facade_percentage + roof_percentage + facaderoof_percentage +
settlement_percentage,  data=RV)

 1) root 410 22800 15.19
   2) roof_percentage < 1.21026 346 19220 15.88
     4) facade_percentage < 0.470953 265 15580 15.35
       8) facaderoof_percentage < 0.466978 259 14960 15.49 *
       9) facaderoof_percentage > 0.466978 6   366  9.00 *
     5) facade_percentage > 0.470953 81  3319 17.63
      10) facade_percentage < 1.38932 38  1261 20.37 *
      11) facade_percentage > 1.38932 43  1521 15.21 *
   3) roof_percentage > 1.21026 64  2518 11.45 *
```

Tree model for correlating explanatory variable group E

```
tree_groupE<-tree(BDM_a_EPT~ field_percentage + legume_percentage + potato_percentage + cereal_percentage +
corn_percentage + rapeseed_percentage + rootvegetable_percentage + vegetable_percentage, data=RV)

1) root 410 22800 15.19
  2) corn_percentage < 5.06913 379 20860 15.61
    4) vegetable_percentage < 0.787182 371 20220 15.73 *
    5) vegetable_percentage > 0.787182 8   384 10.00 *
  3) corn_percentage > 5.06913 31  1043 10.03 *
```

Tree model for correlating and non-correlating explanatory variables

```
tree_correlating_noncorrelating<-tree(BDM_a_EPT~ fruit_percentage + vine_percentage + forest_percentage +
green_percentage + decidious_per_forest + track_percentage + street_percentage + canal_percentage + dam_count +
hydropower_count + slope_mean + slope_max + Masl +carbonate_per_carbonatesilicate +
floodplainwetland_percentage + watercourse_bdm_m + roof_percentage +  corn_percentage + vegetable_percentage +
disposalsite_2004_percentage + wastewater_m3.a, data=RV)

  1) root 410 22800.00 15.190
    2) forest_percentage < 19.5537 194  7907.00 12.260
      4) green_percentage < 16.1795 53  1276.00  8.660
        8) watercourse_bdm_m < 23374 44   864.90  7.545 *
        9) watercourse_bdm_m > 23374 9    88.89 14.110 *
      5) green_percentage > 16.1795 141  5688.00 13.610
       10) corn_percentage < 5.06913 125  4883.00 14.240
         20) Masl < 2119.74 63  2363.00 16.290
           40) street_percentage < 2.29651 52  1531.00 17.400
             80) watercourse_bdm_m < 2120.01 22   684.60 14.860 *
             81) watercourse_bdm_m > 2120.01 30   599.90 19.270 *
           41) street_percentage > 2.29651 11   460.00 11.000
             82) slope_max < 0.23466 6   175.30 15.330 *
             83) slope_max > 0.23466 5    36.80  5.800 *
         21) Masl > 2119.74 62  1988.00 12.160
           42) watercourse_bdm_m < 10220.6 36   993.60  9.889
             84) carbonate_per_carbonatesilicate < 0.467202 14   208.90 13.290 *
             85) carbonate_per_carbonatesilicate > 0.467202 22   520.40  7.727
              170) slope_mean < 0.344073 16   203.00  9.750 *
              171) slope_mean > 0.344073 6    77.33  2.333 *
           43) watercourse_bdm_m > 10220.6 26   551.50 15.310 *
       11) corn_percentage > 5.06913 16   367.40  8.688 *
    3) forest_percentage > 19.5537 216 11720.00 17.820
      6) green_percentage < 29.1693 140  6733.00 15.980
       12) wastewater_m3.a < 262552 134  5957.00 16.450
         24) roof_percentage < 1.21426 105  4527.00 17.400
           48) slope_mean < 0.328348 70  2567.00 18.670
             96) Masl < 547.882 8   194.90 12.880 *
             97) Masl > 547.882 62  2069.00 19.420
              194) green_percentage < 4.23055 6   184.80 26.170 *
              195) green_percentage > 4.23055 56  1582.00 18.700
                390) decidious_per_forest < 0.265811 10   297.60 14.200 *
                391) decidious_per_forest > 0.265811 46  1038.00 19.670
                  782) decidious_per_forest < 0.97257 30   410.30 21.300 *
                  783) decidious_per_forest > 0.97257 16   399.80 16.620 *
           49) slope_mean > 0.328348 35  1620.00 14.860
             98) decidious_per_forest < 0.627444 11   444.50 18.640 *
             99) decidious_per_forest > 0.627444 24   946.60 13.120 *
         25) roof_percentage > 1.21426 29   990.00 13.000 *
       13) wastewater_m3.a > 262552 6    87.50  5.500 *
      7) green_percentage > 29.1693 76  3635.00 21.220
       14) forest_percentage < 25.8828 19   732.60 17.420
         28) decidious_per_forest < 0.932128 11   204.90 20.910 *
         29) decidious_per_forest > 0.932128 8   209.90 12.620 *
       15) forest_percentage > 25.8828 57  2536.00 22.490 *
```

Tree model for explanatory variables slope

```
tree_slope<-tree(BDM_a_EPT~ slope_mean + slope_max, data=RV)

 1) root 410 22800 15.190
   2) slope_mean < 0.028612 37  1112  9.811 *
   3) slope_mean > 0.028612 373 20510 15.720
     6) slope_mean < 0.329316 275 14300 16.790 *
     7) slope_mean > 0.329316 98  5019 12.730
      14) slope_max < 0.647492 48  2331 10.830 *
      15) slope_max > 0.647492 50  2348 14.560 *
```

10 Tree model for IBCH taxa

Tree model for correlating explanatory variable group C

```
tree_groupC<-tree(BDM_a_IBCH~ area_bdm_m2 + watercourse_bdm_m + area_total_m2 + watercourse_total_m, data=RV)

 1) root 410 7929.00 11.220
   2) area_bdm_m2 < 2.92108e+007 380 7291.00 11.360
     4) area_bdm_m2 < 2.19977e+007 364 6949.00 11.240 *
     5) area_bdm_m2 > 2.19977e+007 16  230.90 13.940
      10) watercourse_bdm_m < 35075.1 7   49.71 16.570 *
      11) watercourse_bdm_m > 35075.1 9   94.89 11.890 *
   3) area_bdm_m2 > 2.92108e+007 30  545.50  9.533
     6) area_total_m2 < 1.15069e+008 18  193.10  7.778 *
     7) area_total_m2 > 1.15069e+008 12  213.70 12.170 *
```

Tree model for correlating explanatory variable group D

```
tree_groupD<-tree(BDM_a_IBCH~ facade_percentage + roof_percentage + facaderoof_percentage +
settlement_percentage,
data=RV)

1) root 410 7929.00 11.220
  2) facade_percentage < 0.0769168 229 3806.00  9.817
    4) roof_percentage < 0.00875523 218 3648.00  9.972 *
    5) roof_percentage > 0.00875523 11   48.18  6.727 *
  3) facade_percentage > 0.0769168 181 3095.00 13.010
    6) roof_percentage < 4.31985 169 2863.00 12.780 *
    7) roof_percentage > 4.31985 12  103.70 16.170 *
```

Tree model for correlating explanatory variable group E

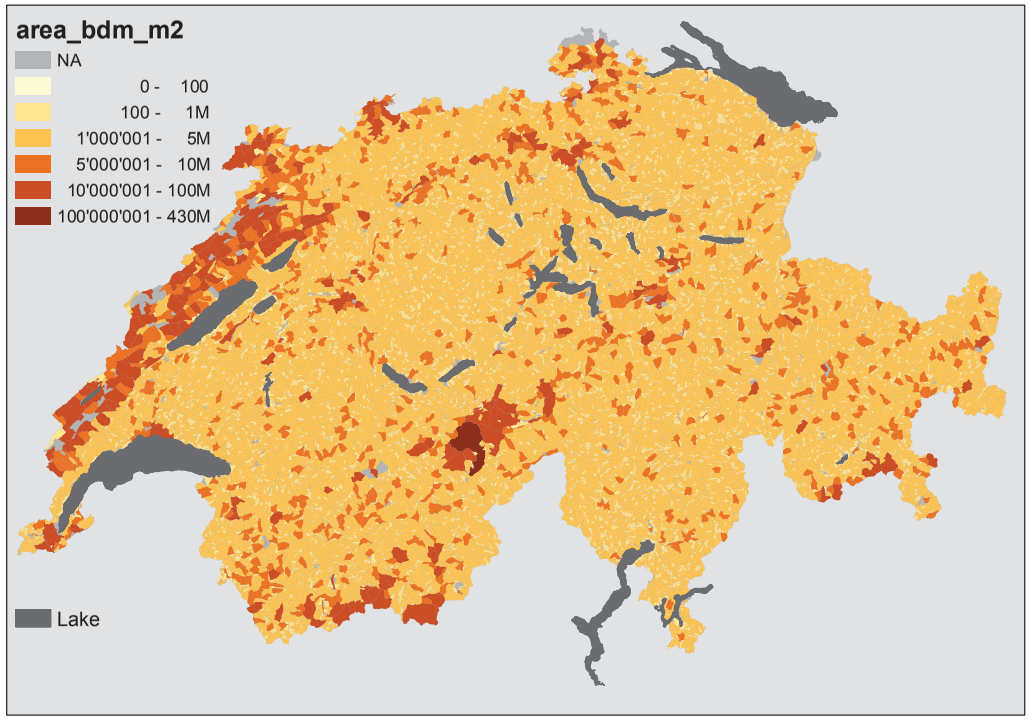
```
tree_groupE<-tree(BDM_a_IBCH~ field_percentage + legume_percentage + potato_percentage + cereal_percentage +
corn_percentage + rapeseed_percentage + rootvegetable_percentage + vegetable_percentage, data=RV)

  1) root 410 7929.00 11.220
    2) rapeseed_percentage < 3.96493e-005 283 4373.00  9.813
      4) vegetable_percentage < 0.00932866 275 4144.00  9.702 *
      5) vegetable_percentage > 0.00932866 8  109.90 13.620 *
    3) rapeseed_percentage > 3.96493e-005 127 1736.00 14.370
      6) rapeseed_percentage < 0.00723982 7   20.86 19.140 *
      7) rapeseed_percentage > 0.00723982 120 1546.00 14.090
       14) rapeseed_percentage < 4.69054 114 1354.00 13.860
         28) rootvegetable_percentage < 0.710902 74  904.50 14.490
           56) field_percentage < 2.97689 11  136.50 11.640 *
           57) field_percentage > 2.97689 63  663.00 14.980
            114) cereal_percentage < 9.86697 54  435.40 15.460 *
            115) cereal_percentage > 9.86697 9  140.90 12.110 *
         29) rootvegetable_percentage > 0.710902 40  366.40 12.700 *
       15) rapeseed_percentage > 4.69054 6   69.50 18.500 *
```

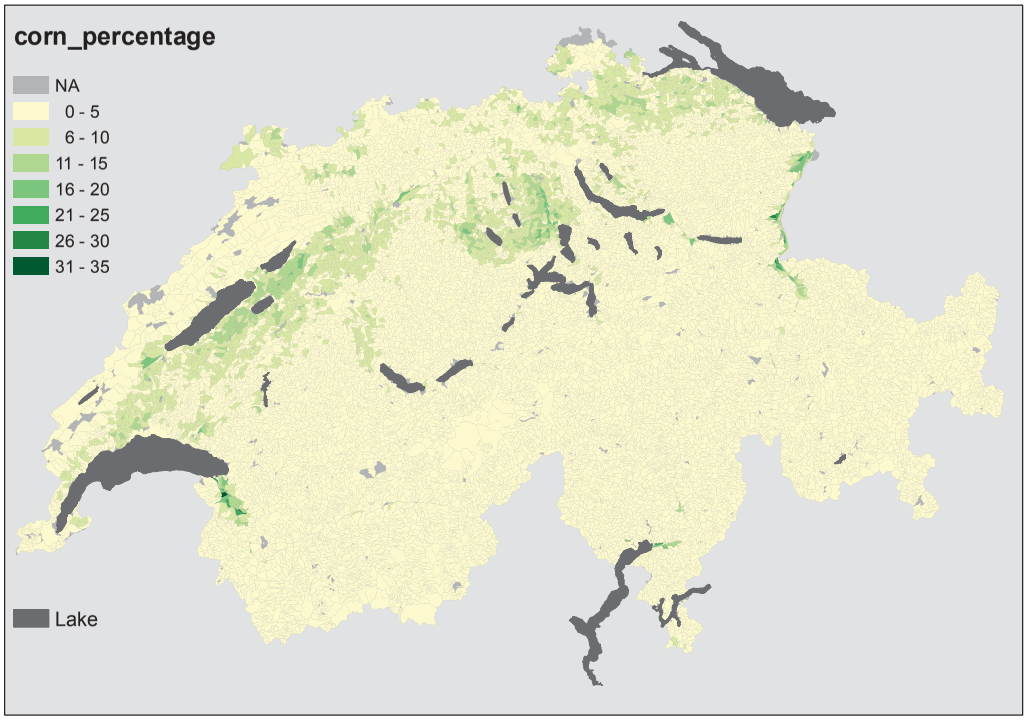
Tree model for correlating and non-correlating explanatory variables

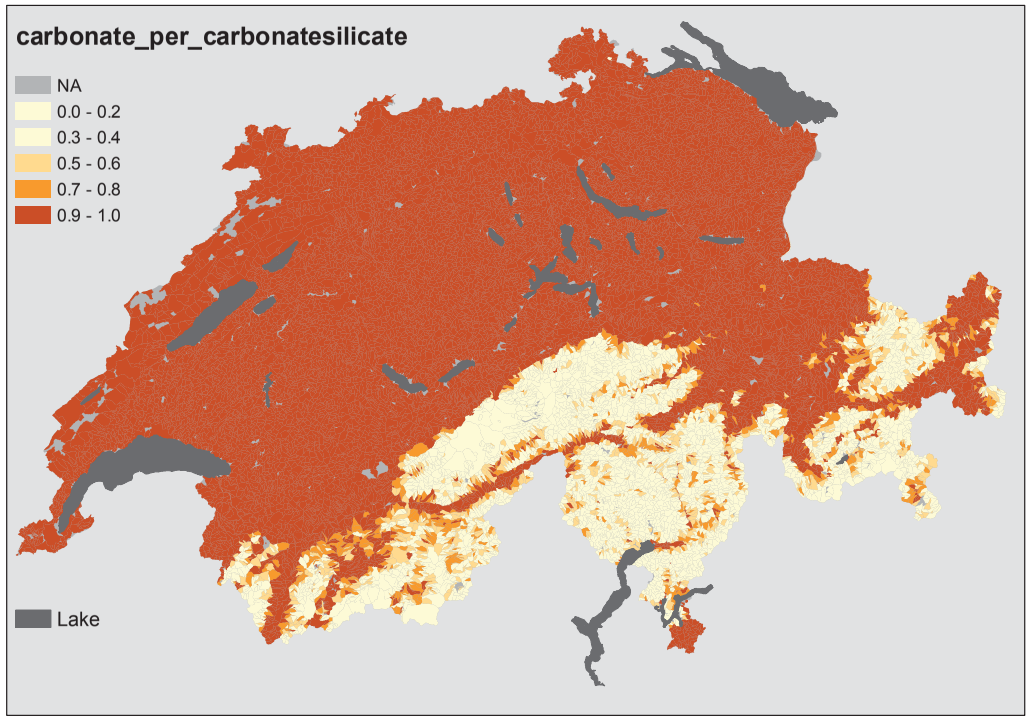
```
tree_correlating_noncorrelating<-tree(BDM_a_IBCH~ fruit_percentage + vine_percentage + forest_percentage +
green_percentage + decidious_per_forest + track_percentage + street_percentage + canal_percentage + dam_count +
hydropower_count + slope_mean + slope_max + Masl +carbonate_per_carbonatesilicate + floodplainwetland_percentage +
area_bdm_m2 + facade_percentage + rapeseed_percentage + vegetable_percentage + disposalsite_2004_percentage +
wastewater_m3.a, data=RV)

  1) root 410 7929.00 11.220
    2) Masl < 1972.79 292 4787.00 12.750
      4) slope_mean < 0.200028 163 2227.00 14.100
        8) vine_percentage < 0.910986 152 1922.00 14.390
         16) area_bdm_m2 < 2.05464e+006 60  695.60 13.200
           32) Masl < 1013.84 50  500.70 13.840
             64) forest_percentage < 19.669 13   81.08 11.380 *
             65) forest_percentage > 19.669 37  313.70 14.700 *
           33) Masl > 1013.84 10   72.00 10.000 *
         17) area_bdm_m2 > 2.05464e+006 92 1085.00 15.170
           34) disposalsite_2004_percentage < 0.270758 84  869.60 14.870
             68) fruit_percentage < 0.128541 20  150.20 16.700 *
             69) fruit_percentage > 0.128541 64  631.40 14.300
              138) vegetable_percentage < 1.12957 58  463.40 13.900
                276) disposalsite_2004_percentage < 0.189894 53  340.30 14.260 *
                277) disposalsite_2004_percentage > 0.189894 5   40.00 10.000 *
              139) vegetable_percentage > 1.12957 6   68.83 18.170 *
           35) disposalsite_2004_percentage > 0.270758 8  125.90 18.380 *
        9) vine_percentage > 0.910986 11  114.90 10.090
         18) slope_mean < 0.0466246 6   21.33 12.670 *
         19) slope_mean > 0.0466246 5    6.00  7.000 *
      5) slope_mean > 0.200028 129 1880.00 11.030
       10) slope_max < 0.736157 97 1315.00 11.730
         20) green_percentage < 9.45511 20  309.20  9.200
           40) slope_mean < 0.327494 7   68.00 12.000 *
           41) slope_mean > 0.327494 13  156.80  7.692 *
         21) green_percentage > 9.45511 77  844.30 12.390
           42) facade_percentage < 1.94036 68  643.10 12.790 *
           43) facade_percentage > 1.94036 9  106.00  9.333 *
       11) slope_max > 0.736157 32  372.70  8.906 *
    3) Masl > 1972.79 118  791.30  7.458 *
```

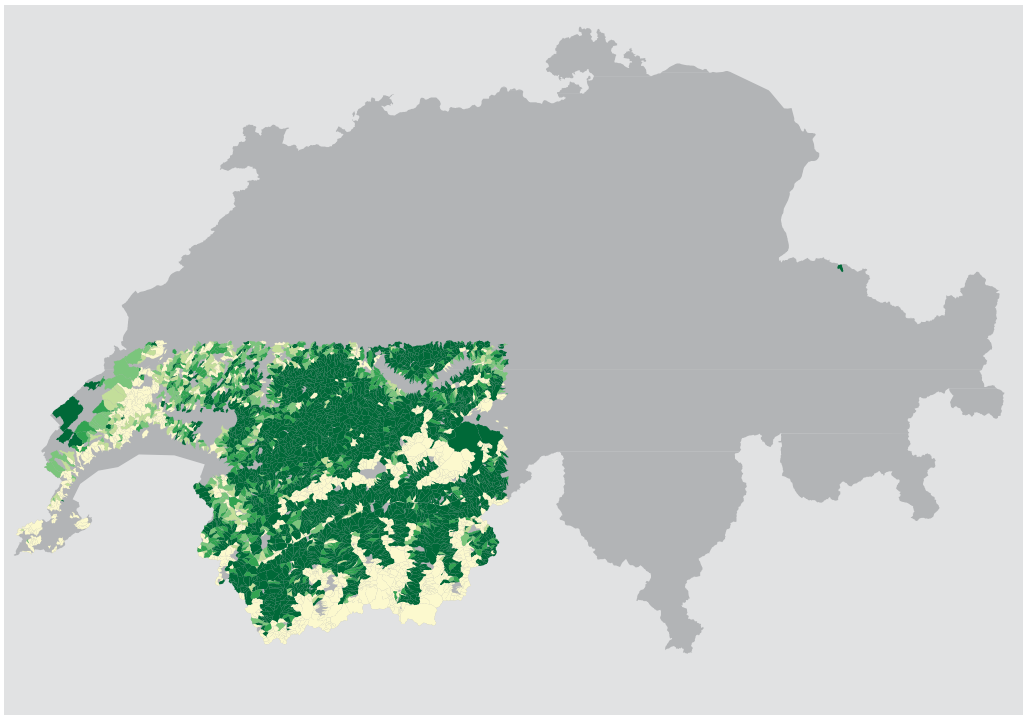
Tree model for explanatory variables slope

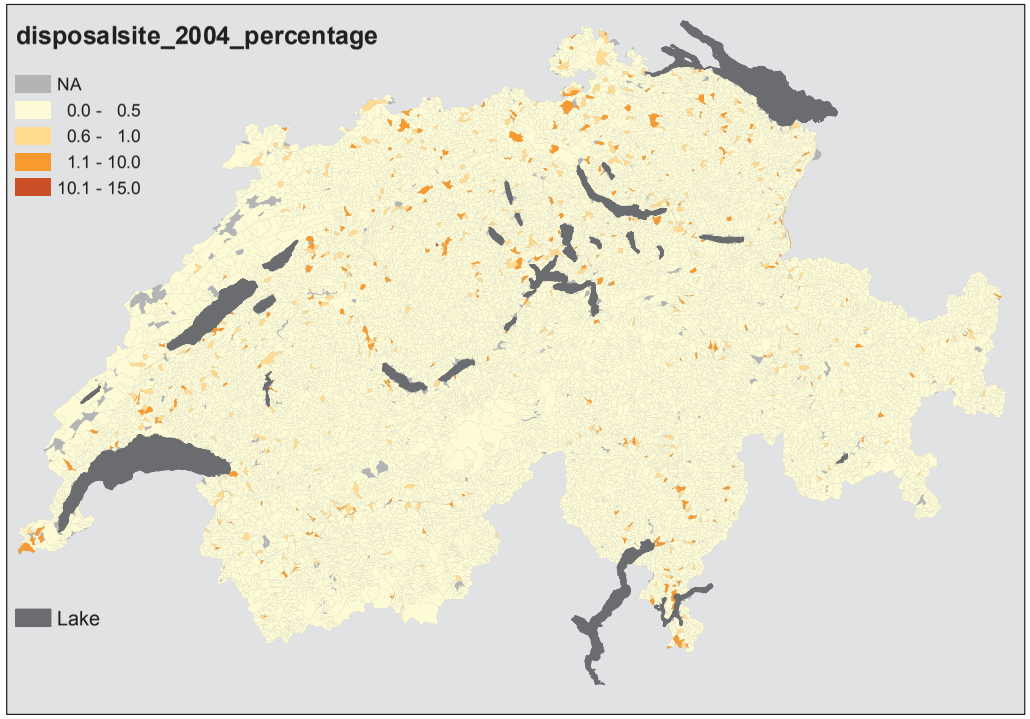
```
tree_slope<-tree(BDM_a_IBCH~ slope_mean + slope_max,  data=RV)

1) root 410 7929.0 11.220
  2) slope_mean < 0.183823 171 3008.0 13.430
    4) slope_mean < 0.0734192 82 1245.0 14.300 *
    5) slope_mean > 0.0734192 89 1643.0 12.630 *
  3) slope_mean > 0.183823 239 3491.0  9.644
    6) slope_max < 0.436869 72  939.9 11.210 *
    7) slope_max > 0.436869 167 2299.0  8.970 *
      14) slope_max < 0.647492 48  2331 10.830 *
      15) slope_max > 0.647492 50  2348 14.560 *
```

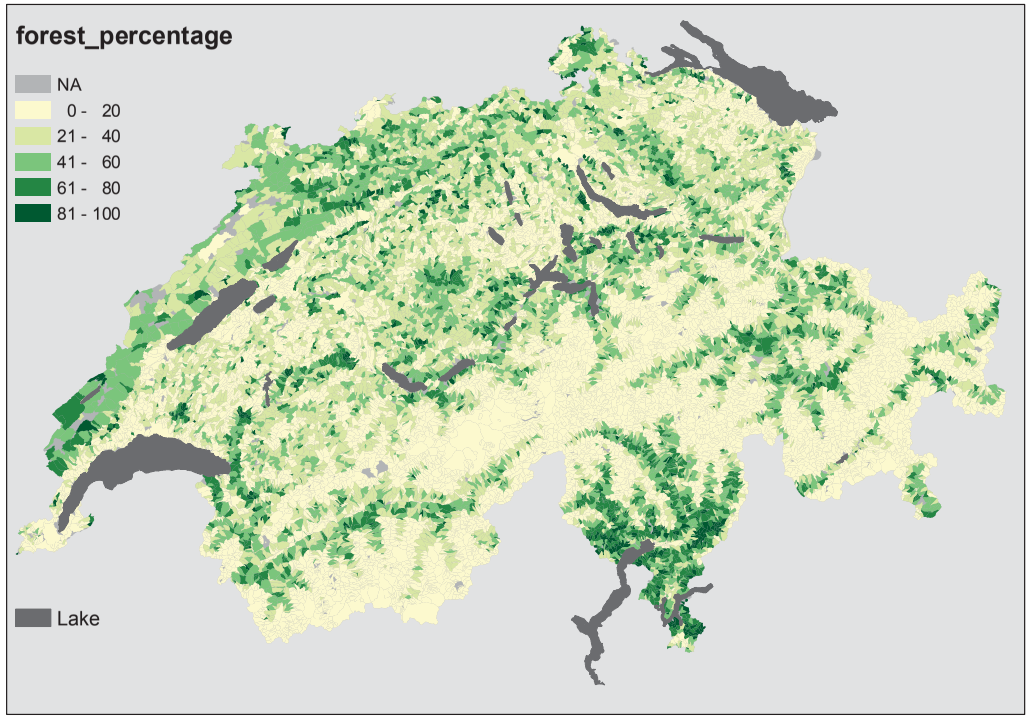

area_bdm_m2
NA
0 - 100
100 - 1M
1'000'001 - 5M
5'000'001 - 10M
10'000'001 - 100M
100'000'001 - 430M
Lake

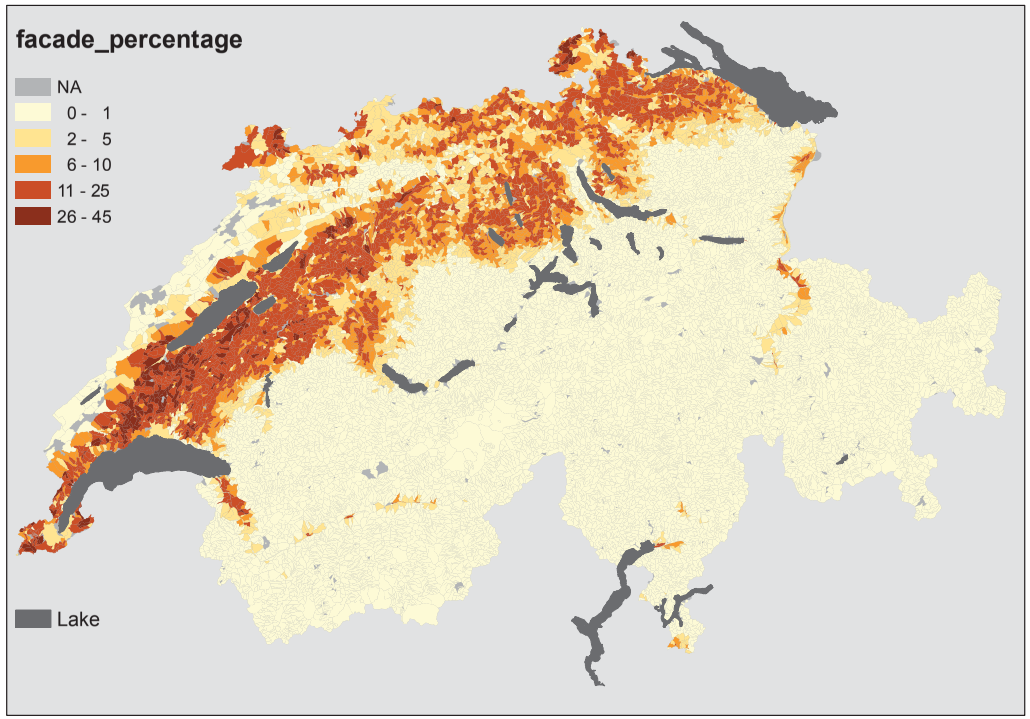

corn_percentage
NA
0 - 5
6 - 10
11 - 15
16 - 20
21 - 25
26 - 30
31 - 35
Lake

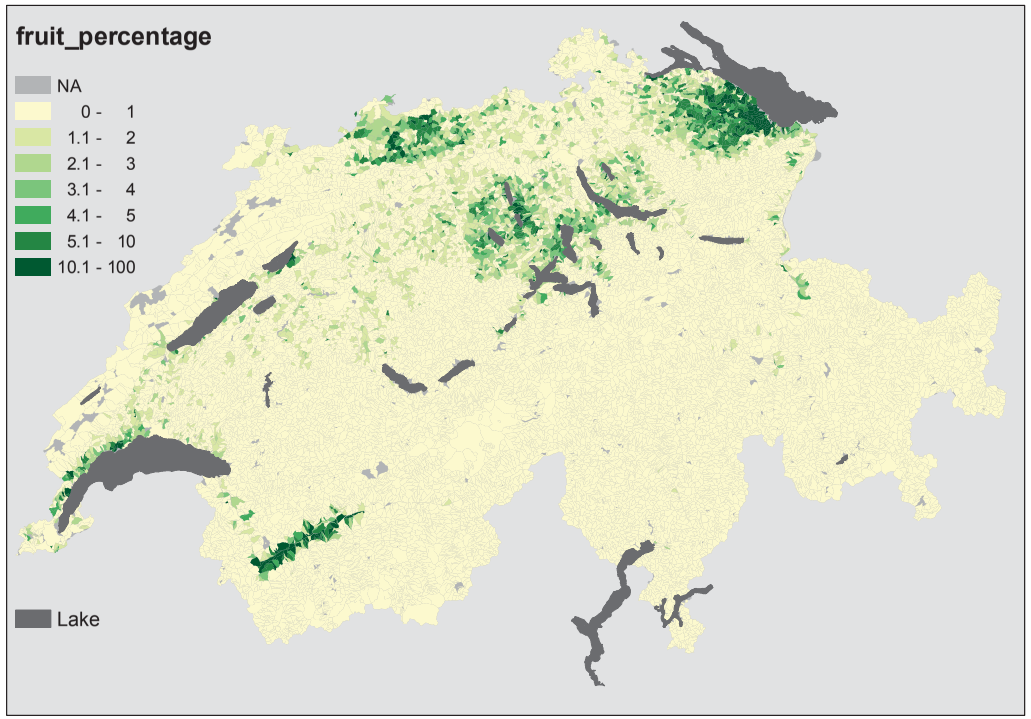

carbonate_per_carbonatesilicate
NA
0.0 - 0.2
0.3 - 0.4
0.5 - 0.6
0.7 - 0.8
0.9 - 1.0
Lake

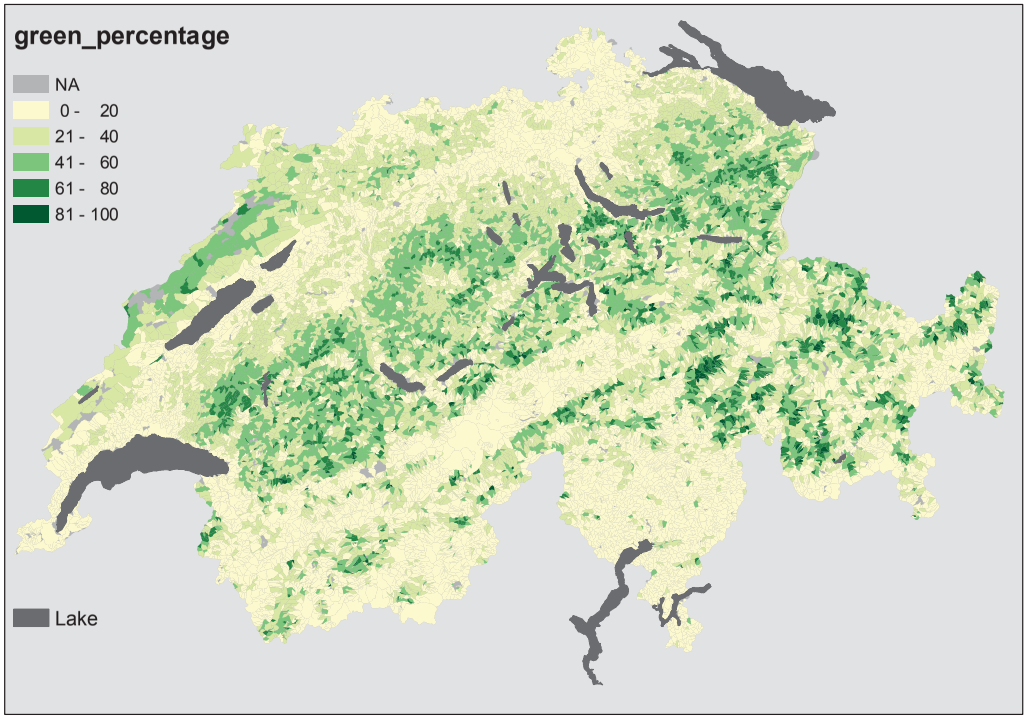


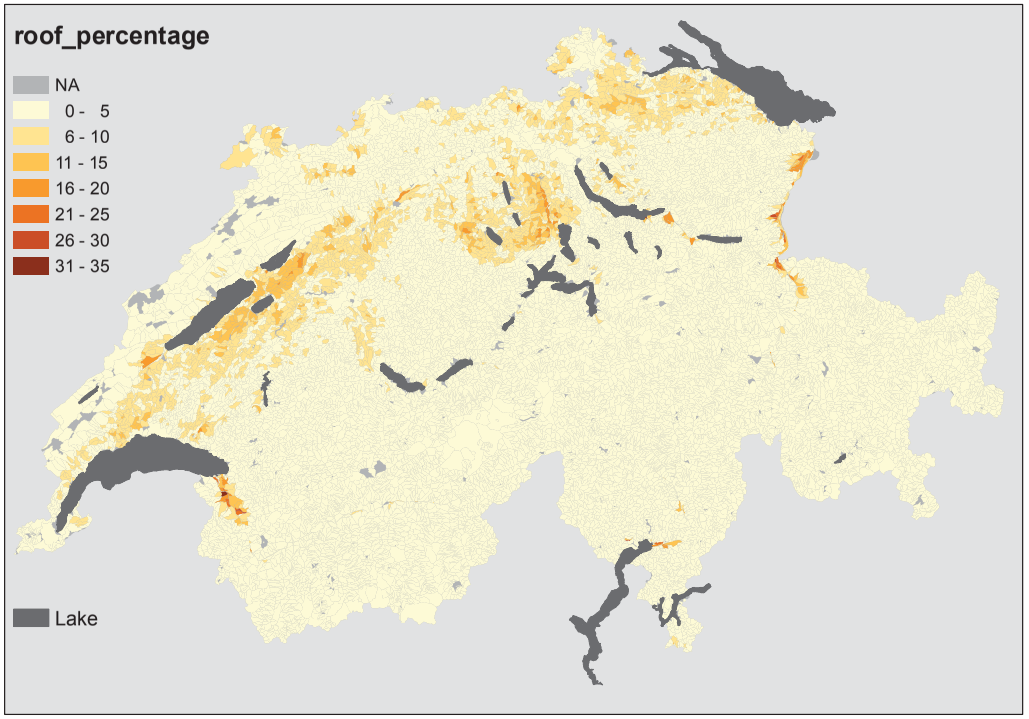

disposalsite_2004_percentage
NA
0.0 - 0.5
0.6 - 1.0
1.1 - 10.0
10.1 - 15.0
Lake

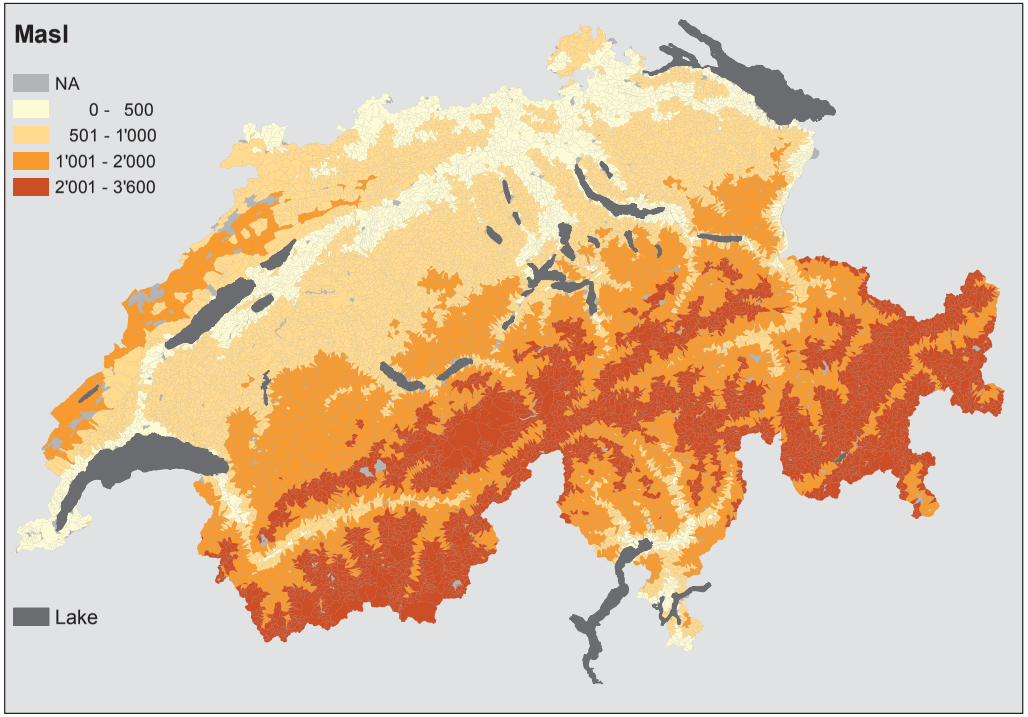

forest_percentage
NA
0 - 20
21 - 40
41 - 60
61 - 80
81 - 100
Lake

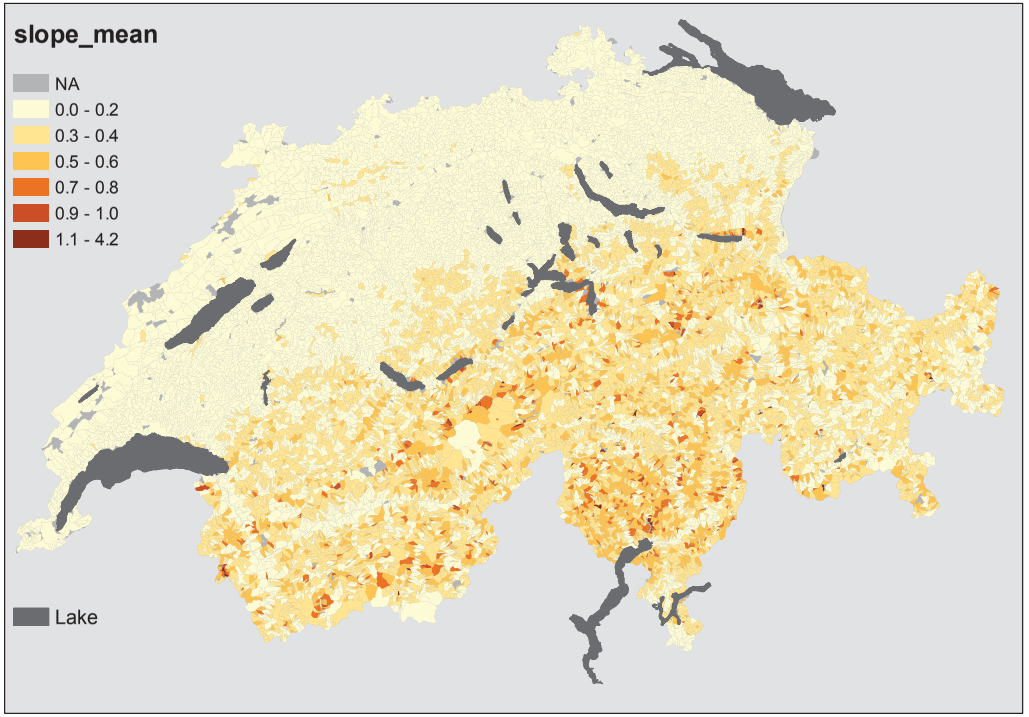

facade_percentage
NA
0 - 1
2 - 5
6 - 10
11 - 25
26 - 45
Lake

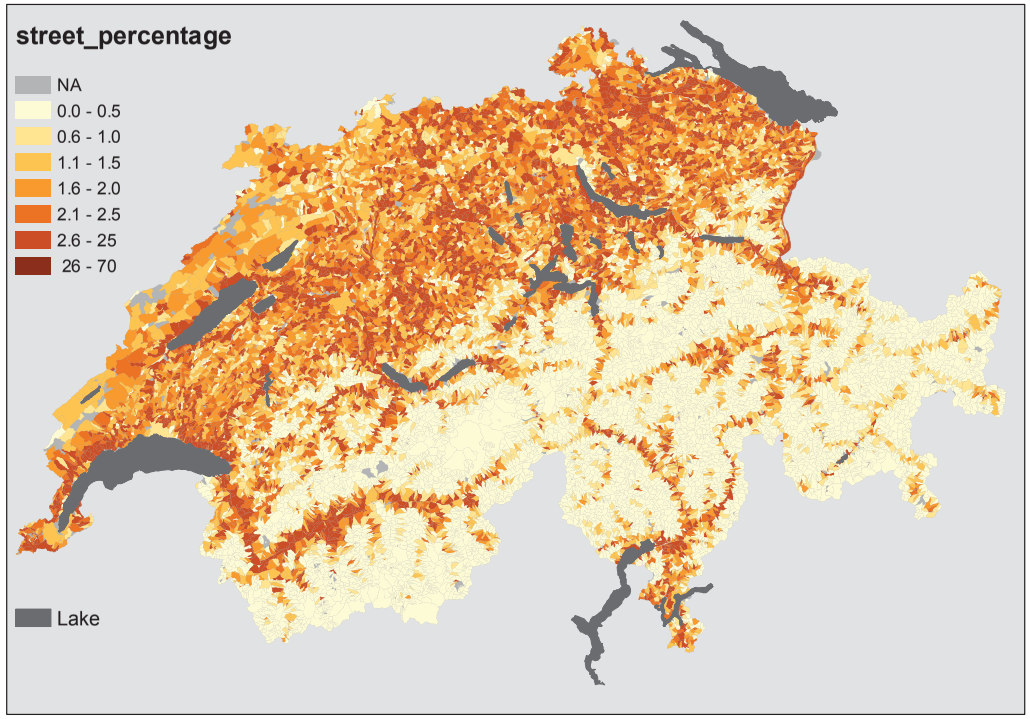

fruit_percentage
NA
0 - 1
1.1 - 2
2.1 - 3
3.1 - 4
4.1 - 5
5.1 - 10
10.1 - 100
Lake

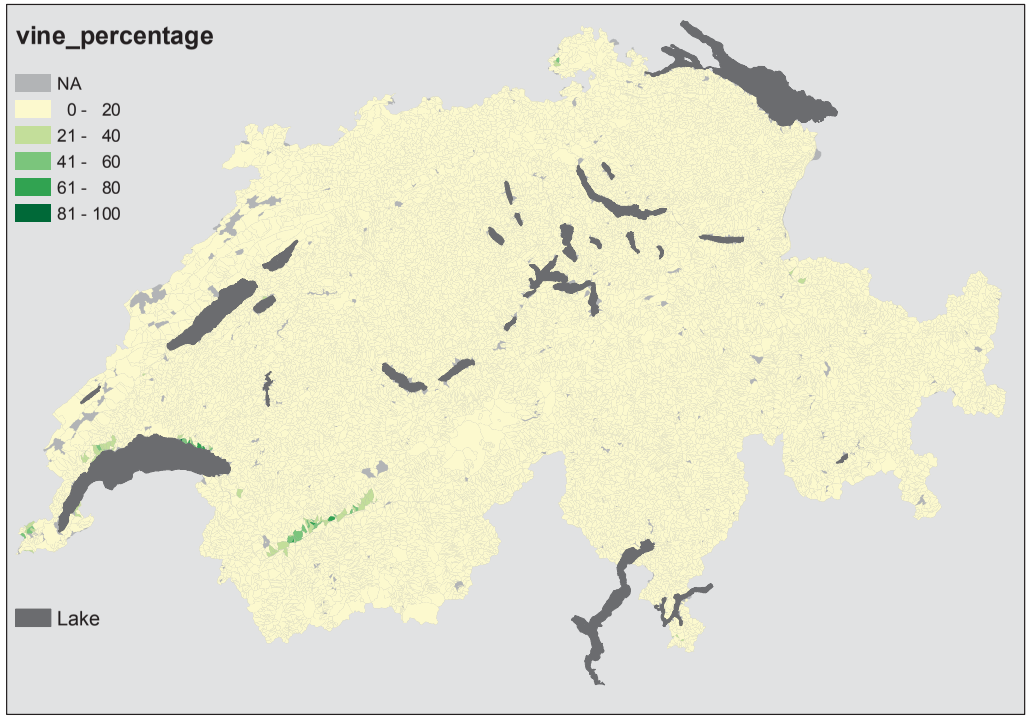

green_percentage
NA
0 - 20
21 - 40
41 - 60
61 - 80
81 - 100
Lake

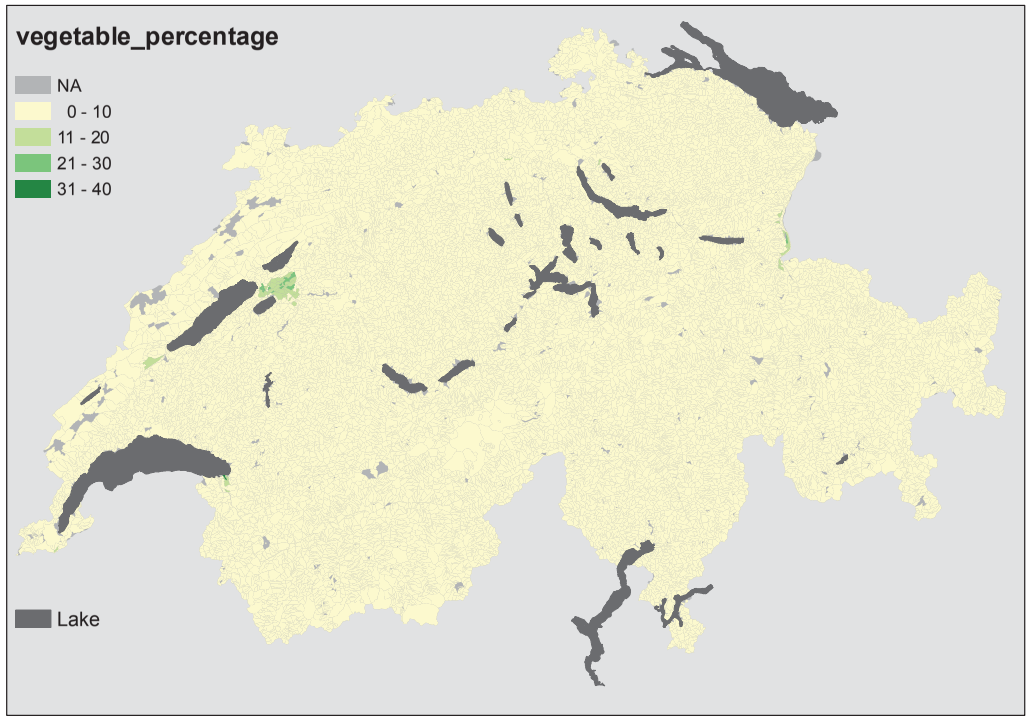

roof_percentage
NA
0 - 5
6 - 10
11 - 15
16 - 20
21 - 25
26 - 30
31 - 35
Lake

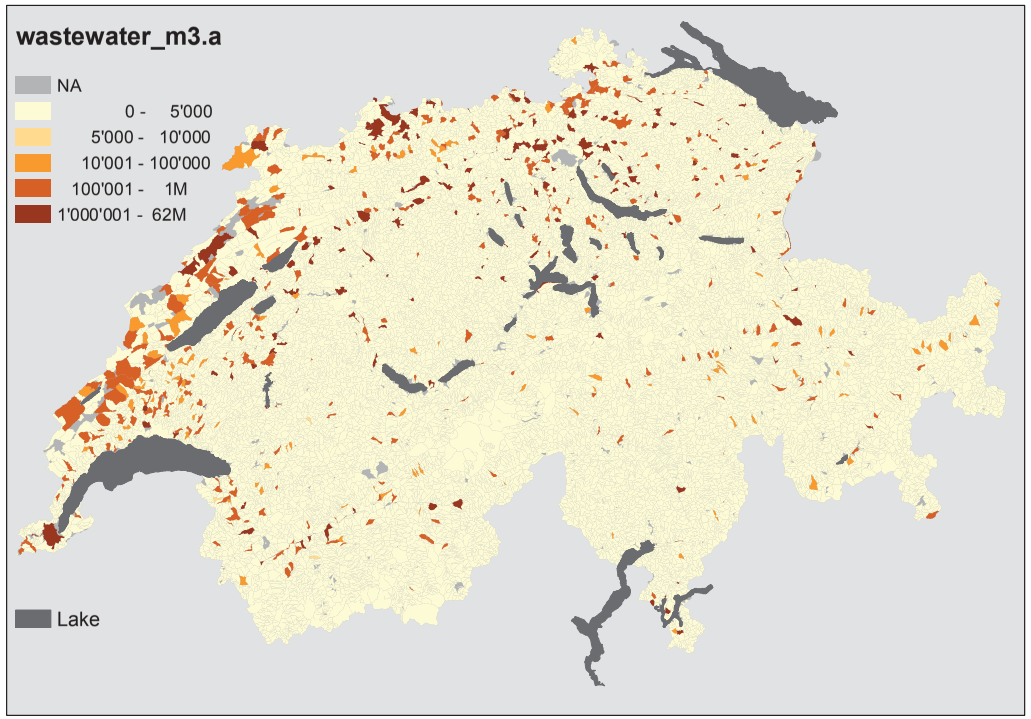

Masl
NA
0 - 500
501 - 1'000
1'001 - 2'000
2'001 - 3'600
Lake

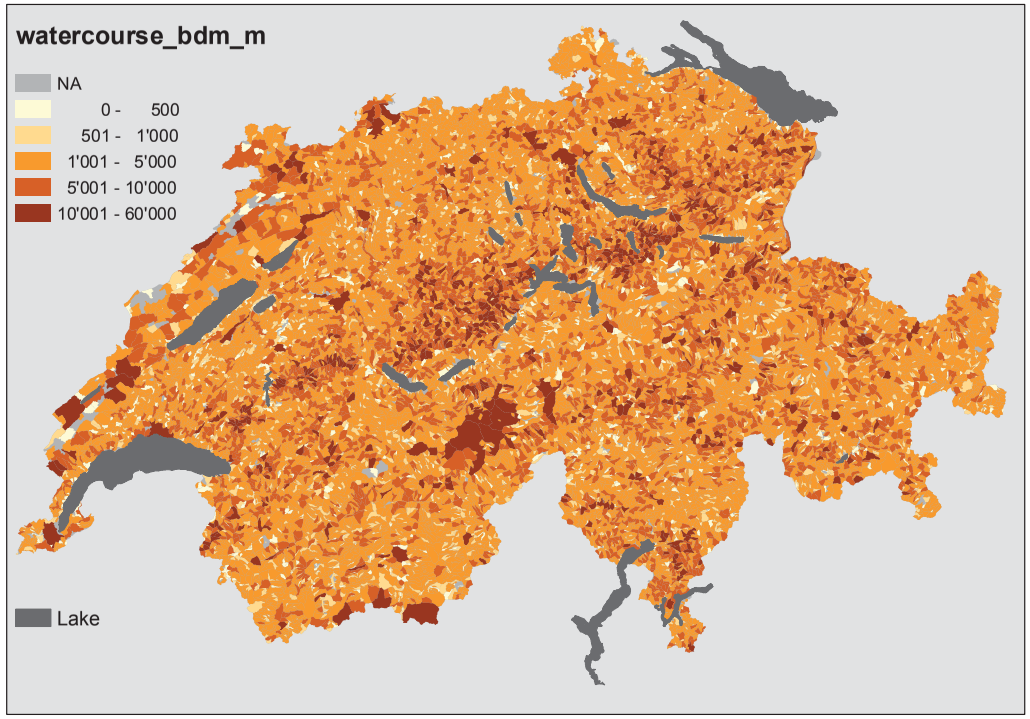

slope_mean
NA
0.0 - 0.2
0.3 - 0.4
0.5 - 0.6
0.7 - 0.8
0.9 - 1.0
1.1 - 4.2
Lake


street_percentage
NA
0.0 - 0.5
0.6 - 1.0
1.1 - 1.5
1.6 - 2.0
2.1 - 2.5
2.6 - 25
26 - 70
Lake


vine_percentage
NA
0 - 20
21 - 40
41 - 60
61 - 80
81 - 100
Lake


vegetable_percentage
NA
0 - 10
11 - 20
21 - 30
31 - 40
Lake


wastewater_m3.a
NA
0 - 5'000
5'000 - 10'000
10'001 - 100'000
100'001 - 1M
1'000'001 - 62M
Lake


watercourse_bdm_m
NA
0 - 500
501 - 1'000
1'001 - 5'000
5'001 - 10'000
10'001 - 60'000
Lake

12 GLM output of the Step Model for the EPT species

```
> summary(step_glm_2wayinteraction_gaussian)
Call:
glm(formula = BDM_a_EPT ~ forest_percentage + green_percentage +
    decidious_per_forest + street_percentage + slope_mean + Masl +
    carbonate_per_carbonatesilicate + watercourse_bdm_m + roof_percentage +
    wastewater_m3.a + corn_percentage + forest_percentage:green_percentage+
    forest_percentage:street_percentage + forest_percentage:Masl +
    forest_percentage:wastewater_m3.a +
    green_percentage:street_percentage + green_percentage:Masl +
    green_percentage:corn_percentage +
    decidious_per_forest:street_percentage +
    decidious_per_forest:slope_mean +
    street_percentage:corn_percentage + slope_mean:Masl +
    Masl:carbonate_per_carbonatesilicate +
    Masl:roof_percentage + Masl:corn_percentage +
    carbonate_per_carbonatesilicate:roof_percentage,
    family = gaussian, data = RV)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-18.5847   -3.3068    0.1917    3.7611   16.9050

Coefficients:
                                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                                    1.818e+01  6.376e+00   2.851 0.004588 **
forest_percentage                             -2.232e-01  6.814e-02  -3.276 0.001150 **
green_percentage                              -1.425e-01  1.053e-01  -1.353 0.176902
decidious_per_forest                           1.048e+01  1.963e+00   5.340 1.60e-07 ***
street_percentage                             -6.633e+00  1.671e+00  -3.969 8.61e-05 ***
slope_mean                                     2.521e+01  8.192e+00   3.077 0.002241 **
Masl                                          -2.964e-03  2.592e-03  -1.144 0.253444
carbonate_per_carbonatesilicate                1.088e+01  4.749e+00   2.291 0.022526 *
watercourse_bdm_m                              4.471e-05  1.746e-05   2.561 0.010817 *
roof_percentage                                5.382e+00  2.437e+00   2.209 0.027768 *
wastewater_m3.a                                9.060e-06  2.695e-06   3.362 0.000853 ***
corn_percentage                               -3.339e+00  8.695e-01  -3.840 0.000144 ***
forest_percentage:green_percentage             2.494e-03  9.693e-04   2.573 0.010471 *
forest_percentage:street_percentage            8.174e-02  1.882e-02   4.343 1.80e-05 ***
forest_percentage:Masl                         1.307e-04  3.794e-05   3.445 0.000633 ***
forest_percentage:wastewater_m3.a             -3.463e-07  9.652e-08  -3.588 0.000376 ***
green_percentage:street_percentage             7.550e-02  2.568e-02   2.939 0.003487 **
green_percentage:Masl                          8.268e-05  4.754e-05   1.739 0.082810 .
green_percentage:corn_percentage              -3.124e-02  1.454e-02  -2.148 0.032339 *
decidious_per_forest:street_percentage        -1.808e+00  7.133e-01  -2.535 0.011630 *
decidious_per_forest:slope_mean               -2.878e+01  6.005e+00  -4.793 2.35e-06 ***
street_percentage:corn_percentage              6.808e-01  1.554e-01   4.381 1.53e-05 ***
slope_mean:Masl                               -9.761e-03  3.811e-03  -2.561 0.010811 *
Masl:carbonate_per_carbonatesilicate          -6.697e-03  2.240e-03  -2.990 0.002970 **
Masl:roof_percentage                          -2.275e-03  1.039e-03  -2.191 0.029080 *
Masl:corn_percentage                           3.312e-03  1.527e-03   2.169 0.030701 *
carbonate_per_carbonatesilicate:roof_percentage -4.673e+00  2.348e+00  -1.990 0.047289 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 35.1077)

    Null deviance: 22797  on 409  degrees of freedom
Residual deviance: 13446  on 383  degrees of freedom
AIC: 2650.6

Number of Fisher Scoring iterations: 2
```

13 GLM output of the Step Model for the IBCH taxa

```
> summary(step_glm_2wayinteraction_gaussian)

Call:
glm(formula = BDM_a_IBCH ~ area_bdm_m2 + facade_percentage +
    vegetable_percentage + disposalsite_2004_percentage +
    forest_percentage + fruit_percentage + green_percentage + Masl +
    slope_mean + vine_percentage +
    area_bdm_m2:disposalsite_2004_percentage + area_bdm_m2:slope_mean +
    facade_percentage:vegetable_percentage +
    facade_percentage:disposalsite_2004_percentage +
    facade_percentage:vine_percentage +
    vegetable_percentage:forest_percentage +
    vegetable_percentage:green_percentage +
    vegetable_percentage:Masl +
    disposalsite_2004_percentage:forest_percentage +
    disposalsite_2004_percentage:fruit_percentage +
    disposalsite_2004_percentage:slope_mean +
    disposalsite_2004_percentage:vine_percentage +
    forest_percentage:fruit_percentage +
    forest_percentage:green_percentage +
    fruit_percentage:green_percentage +
    fruit_percentage:Masl + green_percentage:vine_percentage +
    Masl:slope_mean, family = gaussian, data = RV)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-10.3684   -2.1110    0.1179    2.0588    8.7299

Coefficients:
                                                  Estimate Std. Error t value Pr(>|t|)
(Intercept)                                      1.513e+01  1.305e+00  11.594  < 2e-16 ***
area_bdm_m2                                      4.160e-08  2.955e-08   1.408 0.160026
facade_percentage                                1.746e-02  7.797e-02   0.224 0.822989
vegetable_percentage                            -7.741e+00  2.158e+00  -3.587 0.000378 ***
disposalsite_2004_percentage                     2.064e+01  4.590e+00   4.498 9.13e-06 ***
forest_percentage                                5.136e-03  1.368e-02   0.375 0.707510
fruit_percentage                                 8.749e-01  4.573e-01   1.913 0.056473 .
green_percentage                                 2.803e-02  1.260e-02   2.224 0.026716 *
Masl                                            -3.432e-03  5.950e-04  -5.767 1.67e-08 ***
slope_mean                                      -1.045e+01  3.326e+00  -3.141 0.001816 **
vine_percentage                                  3.605e-01  1.588e+00   0.227 0.820532
area_bdm_m2:disposalsite_2004_percentage        -3.541e-07  1.673e-07  -2.116 0.035005 *
area_bdm_m2:slope_mean                          -2.073e-07  1.067e-07  -1.943 0.052719 .
facade_percentage:vegetable_percentage           2.661e-01  1.301e-01   2.046 0.041453 *
facade_percentage:disposalsite_2004_percentage  -6.399e-01  1.794e-01  -3.566 0.000408 ***
facade_percentage:vine_percentage                7.386e-01  3.032e-01   2.436 0.015320 *
vegetable_percentage:forest_percentage          -4.876e-02  2.706e-02  -1.802 0.072338 .
vegetable_percentage:green_percentage           -1.578e-01  4.990e-02  -3.163 0.001689 **
vegetable_percentage:Masl                        2.166e-02  5.670e-03   3.820 0.000156 ***
disposalsite_2004_percentage:forest_percentage  -4.719e-01  9.336e-02  -5.054 6.72e-07 ***
disposalsite_2004_percentage:fruit_percentage   -2.340e+00  9.239e-01  -2.533 0.011712 *
disposalsite_2004_percentage:slope_mean          2.032e+01  6.650e+00   3.056 0.002402 **
disposalsite_2004_percentage:vine_percentage    -3.403e+00  2.049e+00  -1.660 0.097649 .
forest_percentage:fruit_percentage               1.722e-02  8.513e-03   2.023 0.043802 *
forest_percentage:green_percentage               1.198e-03  4.308e-04   2.781 0.005685 **
fruit_percentage:green_percentage                2.550e-02  1.056e-02   2.415 0.016185 *
fruit_percentage:Masl                           -3.899e-03  1.085e-03  -3.593 0.000370 ***
green_percentage:vine_percentage                -2.499e-01  8.720e-02  -2.865 0.004396 **
Masl:slope_mean                                  4.635e-03  1.971e-03   2.352 0.019166 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 10.79995)

    Null deviance: 7929.4  on 409  degrees of freedom
Residual deviance: 4114.8  on 381  degrees of freedom
AIC: 2169.1

Number of Fisher Scoring iterations: 2
```

14 GLM output of the Lasso Model for the EPT species

```
> coef
67 x 1 sparse Matrix of class "dgCMatrix"
                                                          1
(Intercept)                                    1.467741e+01
forest_percentage                             -4.249436e-03
green_percentage                               6.113751e-02
decidious_per_forest                           6.659862e+00
street_percentage                              .
slope_mean                                     2.608757e+00
Masl                                          -1.375643e-03
carbonate_per_carbonatesilicate                .
watercourse_bdm_m                              3.988552e-05
roof_percentage                                1.142867e-03
wastewater_m3.a                                .
corn_percentage                                .
forest_percentage:green_percentage             2.634846e-03
forest_percentage:decidious_per_forest        -2.422857e-02
forest_percentage:street_percentage            8.860247e-03
forest_percentage:slope_mean                   .
forest_percentage:Masl                         4.812541e-05
forest_percentage:carbonate_per_carbonatesilicate .
forest_percentage:watercourse_bdm_m            2.574655e-07
forest_percentage:roof_percentage              .
forest_percentage:wastewater_m3.a             -1.348504e-07
forest_percentage:corn_percentage              .
green_percentage:decidious_per_forest          .
green_percentage:street_percentage             .
green_percentage:slope_mean                    .
green_percentage:Masl                          .
green_percentage:carbonate_per_carbonatesilicate .
green_percentage:watercourse_bdm_m             2.378767e-07
green_percentage:roof_percentage              -1.034029e-02
green_percentage:wastewater_m3.a               1.126917e-07
green_percentage:corn_percentage              -1.387948e-02
decidious_per_forest:street_percentage        -7.474715e-01
decidious_per_forest:slope_mean               -1.438781e+01
decidious_per_forest:Masl                      .
decidious_per_forest:carbonate_per_carbonatesilicate  6.577744e-01
decidious_per_forest:watercourse_bdm_m        -1.814561e-05
decidious_per_forest:roof_percentage           .
decidious_per_forest:wastewater_m3.a           .
decidious_per_forest:corn_percentage           .
street_percentage:slope_mean                   .
street_percentage:Masl                        -1.079370e-03
street_percentage:carbonate_per_carbonatesilicate .
street_percentage:watercourse_bdm_m            .
street_percentage:roof_percentage             -8.473401e-02
street_percentage:wastewater_m3.a              .
street_percentage:corn_percentage              4.361172e-02
slope_mean:Masl                               -2.217827e-03
slope_mean:carbonate_per_carbonatesilicate     3.275187e+00
slope_mean:watercourse_bdm_m                   .
slope_mean:roof_percentage                    -1.790512e+00
slope_mean:wastewater_m3.a                     .
slope_mean:corn_percentage                     4.031424e+00
Masl:carbonate_per_carbonatesilicate          -2.158754e-03
Masl:watercourse_bdm_m                         .
Masl:roof_percentage                          -2.280812e-04
Masl:wastewater_m3.a                           .
Masl:corn_percentage                           .
carbonate_per_carbonatesilicate:watercourse_bdm_m .
carbonate_per_carbonatesilicate:roof_percentage .
carbonate_per_carbonatesilicate:wastewater_m3.a .
carbonate_per_carbonatesilicate:corn_percentage -8.386759e-01
watercourse_bdm_m:roof_percentage             -2.744490e-05
watercourse_bdm_m:wastewater_m3.a              .
watercourse_bdm_m:corn_percentage              .
roof_percentage:wastewater_m3.a                3.204570e-07
roof_percentage:corn_percentage                4.878630e-02
wastewater_m3.a:corn_percentage                2.757282e-07
```

15 GLM output of the Lasso Model for the IBCH taxa

```
> coef * IBCH
56 x 1 sparse Matrix of class "dgCMatrix"
                                                          1
(Intercept)                                   14.4861707785
area_bdm_m2                                    .
facade_percentage                              .
vegetable_percentage                           .
disposalsite_2004_percentage                   .
forest_percentage                              .
fruit_percentage                               .
green_percentage                               0.0027353659
Masl                                          -0.0022750994
slope_mean                                    -2.3011876581
vine_percentage                                .
area_bdm_m2:facade_percentage                  .
area_bdm_m2:vegetable_percentage               .
area_bdm_m2:disposalsite_2004_percentage       .
area_bdm_m2:forest_percentage                  .
area_bdm_m2:fruit_percentage                   .
area_bdm_m2:green_percentage                   .
area_bdm_m2:Masl                               .
area_bdm_m2:slope_mean                         .
area_bdm_m2:vine_percentage                    .
facade_percentage:vegetable_percentage         .
facade_percentage:disposalsite_2004_percentage .
facade_percentage:forest_percentage            .
facade_percentage:fruit_percentage             .
facade_percentage:green_percentage             .
facade_percentage:Masl                         .
facade_percentage:slope_mean                   .
facade_percentage:vine_percentage              .
vegetable_percentage:disposalsite_2004_percentage .
vegetable_percentage:forest_percentage         .
vegetable_percentage:fruit_percentage          .
vegetable_percentage:green_percentage          .
vegetable_percentage:Masl                      .
vegetable_percentage:slope_mean                .
vegetable_percentage:vine_percentage           .
disposalsite_2004_percentage:forest_percentage .
disposalsite_2004_percentage:fruit_percentage  .
disposalsite_2004_percentage:green_percentage  .
disposalsite_2004_percentage:Masl              .
disposalsite_2004_percentage:slope_mean        .
disposalsite_2004_percentage:vine_percentage   .
forest_percentage:fruit_percentage             .
forest_percentage:green_percentage             0.0006638084
forest_percentage:Masl                         .
forest_percentage:slope_mean                   .
forest_percentage:vine_percentage             -0.0022935643
fruit_percentage:green_percentage              .
fruit_percentage:Masl                          .
fruit_percentage:slope_mean                    .
fruit_percentage:vine_percentage               .
green_percentage:Masl                          .
green_percentage:slope_mean                    .
green_percentage:vine_percentage               .
Masl:slope_mean                                .
Masl:vine_percentage                           .
slope_mean:vine_percentage                     .
```