



**University of
Zurich**^{UZH}

Uncertainty aware density estimation with user-generated tick bite data

GEO 620 Master's Thesis

Author

Marius Lechtenböhmer
13-616-651

Supervised by

Prof. Dr. Patrick Laube (patrick.laube@zhaw.ch)

Faculty representative

Prof. Dr. Ross Purves

30.09.2020

Department of Geography, University of Zurich



**University of
Zurich^{UZH}**

Department of Geography

Uncertainty aware density estimation with user-generated tick bite data

Master Thesis in the specialized Geography Master: GIScience

By

Marius Lechtenböhrer

Dornstrasse 8

9464 Rüthi

marius.lechtenboehmer@uzh.ch

Supervisor:

Prof. Dr. Patrick Laube

Faculty member:

Prof. Dr. Ross Purves

September 2020

Acknowledgements

I would like to thank all the people, who helped me with the preparations, execution, and writing of this paper.

My biggest gratitude goes out to Prof. Patrick Laube for his guidance, encouragement, and gracious support throughout the whole course of my master thesis.

I would also like to thank Nils Ratnaweera for instructing me and providing the help I needed to write the code in R. I learned an immense amount of coding in R from him, which gave me more possibilities for my master thesis.

I would also like to thank my parents for supporting and believing in me and giving me the space I needed. They were always there to help me when I needed it.

Abstract

Ticks can be found almost everywhere in Switzerland, but very little research has been done on the distribution of it. For this the ZHAW developed an app called "Zecke", which gathered crowdsourced data for about 5 years. Users could report tick bites on the app with a circle, that marks the area where they were bitten. This data contains many variables including the circle given in meters of radius around the centre. This inaccuracy is uncertain spatial data, which needs to be considered. For this inaccuracy three different methods were developed to calculate the density of the tick reports, with concern to the inaccuracy radius. In all methods the KDE was used. The *normal* KDE was used as a standard measure to which comparisons could be made. The *weighted* method puts a weight on the inaccuracies; the bigger the radius the smaller the impact and vice versa. The *concurrent* and *sequential* methods use a combination of the MCS and KDE to represent the spatial properties of the inaccuracy of the tick reports. To analyse the methods and find the differences between them small multiples were created, each with four different kernel parameters as well as increases of inaccuracy radii to simulate what impact the inaccuracies have on the density. To compare the methods better, all experiments were done on a local and national scale. The *normal* KDE calculates the highest density, followed closely by the *concurrent* and *sequential* KDE. The *weighted* KDE calculates the overall lowest KDE.

Index

Acknowledgements	I
Abstract	II
Index	III
1. Introduction	1
1.1 Relevance	1
1.2 Research questions	1
1.3 Structure	2
2. Related Work	4
2.1 Tick research in Switzerland & other countries	4
2.2 Data Quality of Crowdsourced Data	7
3. Data settings	12
3.1 Used data	12
3.2 Standard report	13
4. Methodology	14
4.1 Conventional KDE	14
4.2 Uncertainty weighted KDE	14
4.3 Monte-Carlo-KDE	15
4.3.1 <i>Concurrent</i> KDE	16
4.3.2 <i>Sequential</i> KDE	16
4.4 Statistical Measurements	17
4.4.1 Mean (Absolute) Error (ME/MAE)	17
4.4.2 Root Mean Square Error (RMSE)	18
4.4.3 Standard Deviation (SD)	18
5. Experiment	19
5.1 Explorative analysis	19
5.2 Sensitivity analysis	19
5.2.1 Local scale	20
5.2.2 National scale	23
5.3 Differential analysis	23
6. Results	24

6.1	Explorative data analysis	24
6.1.1	Temporal	24
6.1.2	Spatial	26
6.2	Sensitivity analysis	30
6.2.1	Local Scale (Sursee region, LU)	30
6.2.2	National scale (Switzerland)	40
6.3	Differential analysis	47
6.3.1	Local scale (Sursee region, LU)	47
6.3.2	National Scale (Switzerland)	58
7.	Discussion	63
7.1.	What spatio-temporal patterns are visible from crowdsourced tick data?	63
7.2.	How to estimate tick density from user-generated tick bite data with positional uncertainty?	65
7.3	What is the relation between the created spatial patterns of the different methods, the kernel parameter, and the inaccuracy radii of the tick reports?	67
8.	Conclusion	70
9.	References	72
10.	Table of figures	77
11.	Appendix	78
11.1	Code snippets	78
11.1.1	Histograms	78
11.1.2	KDE	83
11.1.3	creating small multiples	86
11.2	Declaration of authorship	95

1. Introduction

1.1 Relevance

Citizen science has gained popularity over the last few years. Citizen scientists are non-professional participants in research. They can be involved in any part of the project, yet mostly data collection (Bonney et al., 2009; Haklay, 2015). Large amounts of data can be costly and difficult to obtain. Citizen science solves both problems and engages citizens in science and helps sensitise them to the topic (Turrini, Dörler, Richter, Heigl, & Bonn, 2018).

According to the World Health Organization (WHO) vectors are living organisms capable of transmitting infectious diseases from animals to humans or between each other. All of them can carry pathogens which are a serious threat to food security, public health, and economic activities worldwide (World Health Organization, 2014). Nine vectors (i.e. mosquitoes, ticks, sandflies, black flies, triatomine bugs, tsetse flies, mites, lice, and snails) have been identified by the WHO (World Health Organization, 2017). Not all vector-borne diseases are lethal, but patients suffering from one might have temporal or chronic disabilities, which prevent victims from providing for their households (World Health Organization, 2014).

Ticks can be found almost everywhere in Switzerland. To help reduce the amount of tick bites happening, the ZHAW created an app called *"Zecke"*. With this app, users can report tick bites (A. S. G. ZHAW Spin-off, n.d.). Because tick bites are hard to spot and difficult to know when and where exactly the users were bitten, an accuracy circle was added. The radius and centre coordinates of this circle including other information are sent to a server with each tick report. The collection of data with the help from the public is called crowdsourcing. With this crowdsourced data, over 40'000 tick reports were sent.

1.2 Research questions

As stated above, this master thesis analyses the crowdsourced tick data for Switzerland. The used program for the analysis is R as it gives more possibilities than ArcGIS, QGIS, SPSS, and tasks could be written to run automatically, which

was necessary (11.1 Code). However, a few visualisations at the end were done in QGIS. The main goal is to propose methods that estimate density maps which take spatial uncertainty factors into account. Following this the research questions are the following:

- What spatio-temporal patterns are visible from crowdsourced tick data?
- How to estimate tick density from user-generated tick bite data with positional uncertainty?
- What is the relation between the created spatial patterns of the different methods, the kernel parameter, and the inaccuracy radii of the tick reports?

1.3 Structure

The presented thesis is divided into eight main chapters, as well as the index, references, table of figures, appendix, and at the beginning the abstract.

The two subchapters before depict the introduction and the research questions to the topic. A small insight into crowdsourcing and ticks was given.

The second chapter gives background information of the current research state of ticks in Switzerland and some tick related projects in other countries. Later, the data quality of crowdsourced data will be covered. Examples of upsides and downsides of other crowdsourcing projects and how they deal with uncertainties are explained later, as well as the kernel density estimation (KDE).

The next chapter talks about the received crowdsourced data provided by the ZHAW (Zürcher Hochschule für Angewandte Wissenschaft). Before using the data, they had to be cleaned up of unwanted data points. The standard reports were one factor of which the data needed to be cleaned of.

The 4th chapter discloses the methods used for the analysis. The KDE is explained, and how it was adjusted in different ways. The four used methods are going to be explained including the used statistical measurements.

The 5th chapter explains the experiments that were done for the analysis. Here it declares how the methods were used to conduct experiments. The experiments of sensitivity and differential analysis were done on a local and national scale. The local scale will be visualized by a map in this subchapter 5.2.1 Local scale.

The results can be found chapter 6. The same structure will be retained from the previous chapter. The explorative analysis will show multiple histograms. First, the temporal analysis will be shown, then the spatial variables. Afterwards, the small multiples of each method for the local and national scale will be investigated. The last part of chapter 5.3 Differential analysis.

Once the small multiples were created, the difference within the small multiples and across the methods could be determined.

To analyse the variance within the small multiples a map was chosen and one adjacent differential map to the right/left and top/bottom of it. The differential map helps in combination with the statistical measurements to distinguish the deviation between the maps within the small multiples.

For the analysis of the difference between the methods, maps with the same kernel parameter and inaccuracy modifier were chosen for both methods. The difference between those two maps was calculated including the used statistical measurements.

In chapter 6 are the differential maps, where the different methods will be compared by their impact on the density map.

Followed by the results is the discussion. There, the most important findings will be reviewed, as well as research questions are going to be discussed in the same order the results are structured and listed at the beginning in 1.2.

The last main chapter is the conclusion. Here a summary of the master thesis is given. The findings followed by the new research gaps are going to be exemplified.

2. Related Work

2.1 Tick research in Switzerland & other countries

The research on ticks in Switzerland has been limited. However, it is commonly known that ticks are found everywhere in Switzerland, yet very little has been done to prevent tick bites.

One of the first research projects in Switzerland took place in 2010. A paper was published with the focus on the molecular epidemiology of tick-borne encephalitis virus in Switzerland (Gäumann, 2010). For that research project ticks were collected in specific locations in Switzerland by the Swiss military and researchers in spring 2009. Over 62'000 ticks were found in 2009 distributed over 176 locations (see [Figure 1](#)). This was the first tick collection project in Switzerland. To collect the data, the soldiers ran white cloth through the bushes and grass, then counted the ticks hanging on the cloth (Gäumann, 2010). There has been no other Swiss-wide tick collection until the start of the mobile app "Zecke".

In the last 18 years a fluctuation of reports of tick-borne infectious diseases and medical consultations was found, but the number of reports has risen in the last few years. In the year 2018, the state issued a static tick risk map on its geo-webserver (map.geo.admin.ch). The state currently uses the same crowdsourced data as for this project, but only up until 2018. The most recent available tick sampling data is generated by the app "Zecke", hence is crowdsourced data. Users submitted over 30'000 tick bite reports with various degrees of precision (ZHAW, 2019). The state's data is not as elaborated as the one in this master thesis, otherwise it is the same. In 2019 the state declared Switzerland a nation-wide risk area for tick-borne encephalitis except the Cantons of Geneva and Ticino. Tick reports with at least one FSME confirmed case are seen in [Figure 2](#).

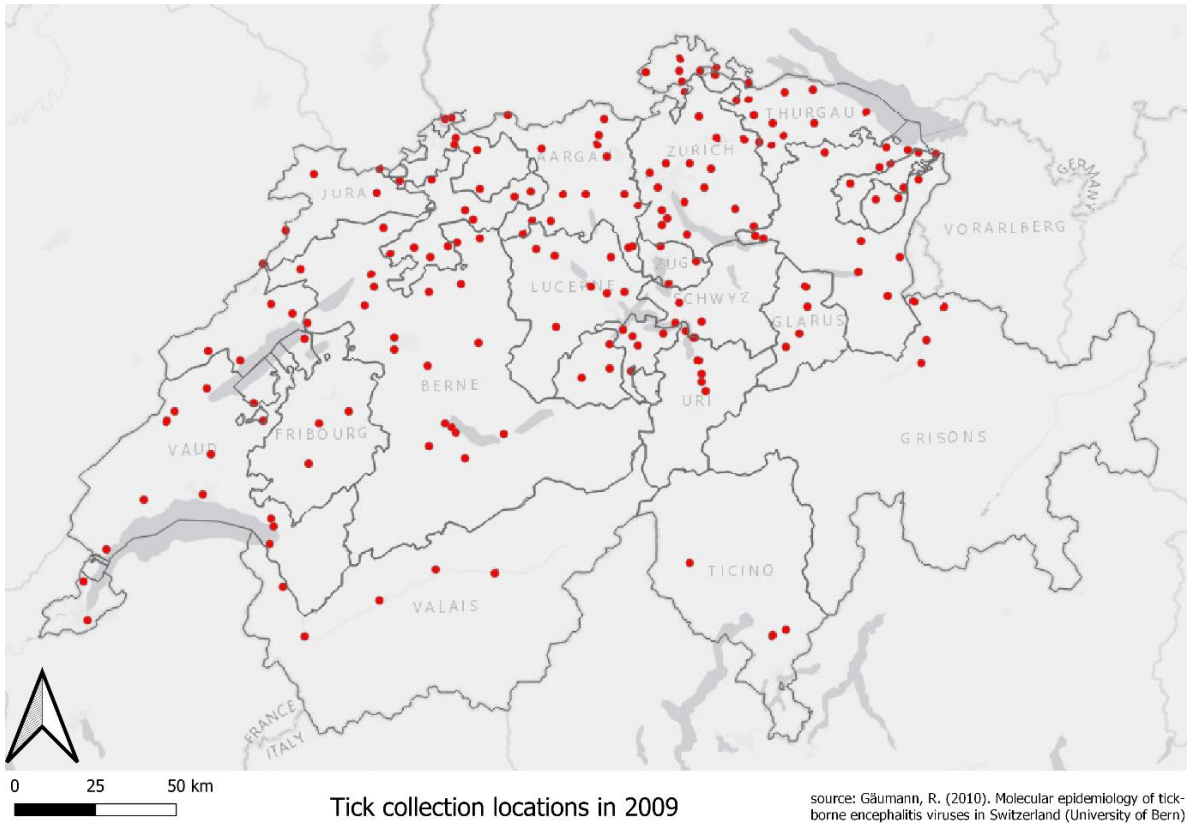


Figure 1: Tick locations of Gäumann data from 2010

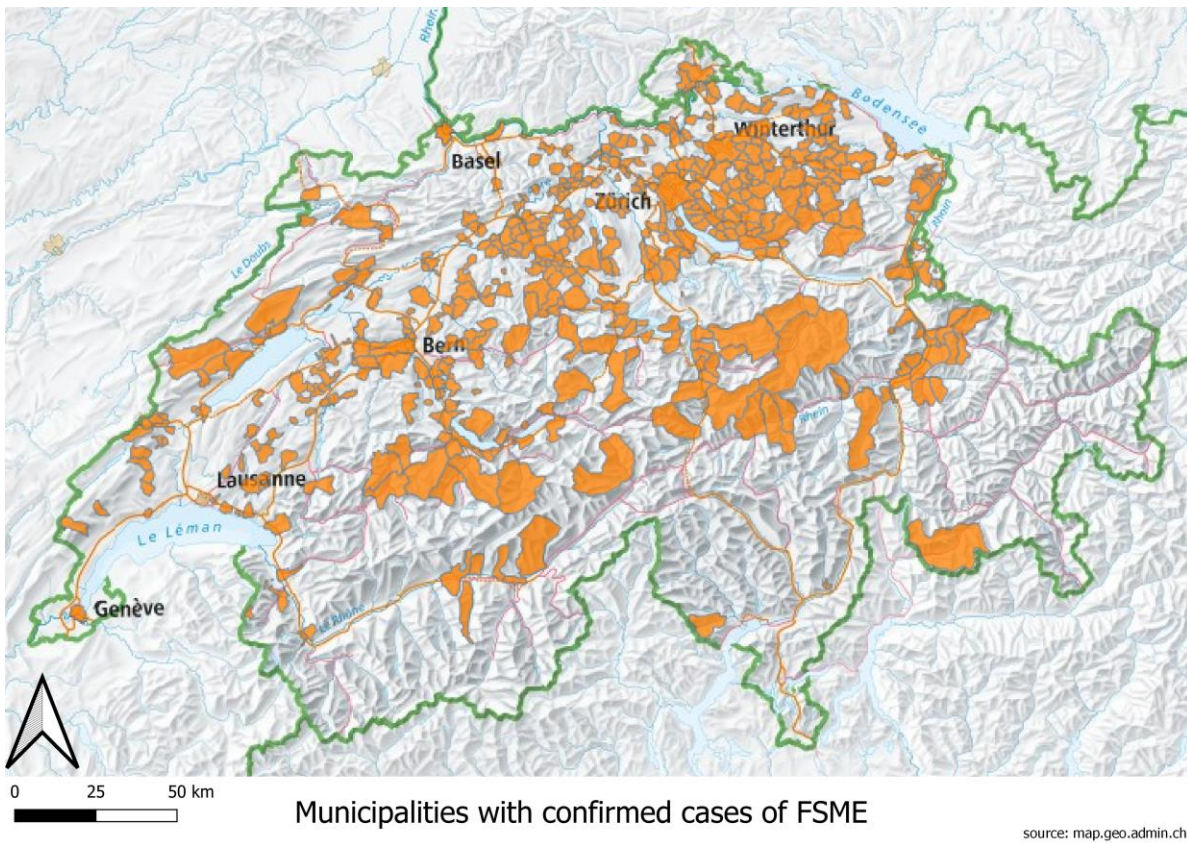


Figure 2: Municipalities with FSME (source: map.geo.admin.ch)

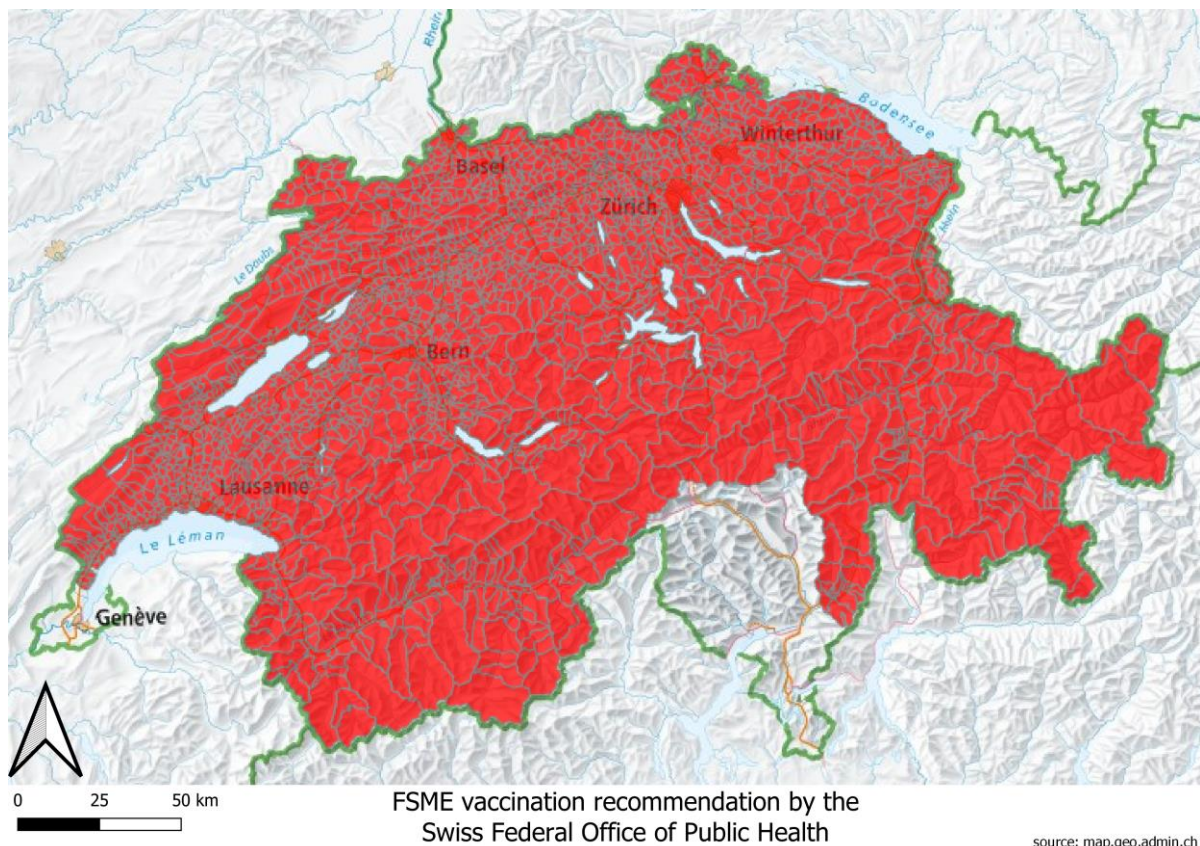


Figure 3: FSME vaccination recommendation (source: map.geo.admin.ch)

It is recommended by the Swiss government for every citizen to get a vaccination against this disease (see [Figure 3](#)) (Bern, 2019; Bern & Bern, 2018). The Netherlands has conducted more research about the spread of ticks than Switzerland. One project which focuses on mapping tick dynamics and tick bite risk using data-driven approach and volunteered observations is by Garcia Martí (Garcia Martí, 2019). There, volunteers went monthly to their allocated forest site and collected ticks by dragging a white cloth through the bushes. The data collection was done between 2006 and 2016.

In the north east of the United States, the Department of Biological Sciences from the Northern Arizona University launched a campaign in January 2016 for citizen scientists to submit and send in ticks. The campaign then investigated the species, which stage the ticks were in, their sex, and if they contain any tick-borne bacteria. The program ran until August 2017. The submissions peaked from the 20th to the

23rd week of each year. A total of 4261 ticks that were sent in have been found on humans (Porter et al., 2019).

Another tick-borne related study from Hungary collected road-kill carcasses in urban areas to test whether these are a carrier of tick-borne pathogens. Yet further investigations need to be done to see if these animals also increase the spread of the bacteria (Szekeres et al., 2019).

While there has been done some research on ticks and their respective diseases in other countries, their results can only help predict the circumstances in Switzerland slightly. Research on how to cure the diseases is great and helps a lot; progress has been made. But it does not protect the citizens from getting the disease in the first place. Informing people where ticks lurk could decrease the chance of getting bitten which would lower the cases of tick-borne diseases in Switzerland. It is generally known that ticks can be found in the forest, but is the spread of ticks evenly distributed over all forests in Switzerland and how to deal with the spatial uncertainty of the crowdsourced data? This research gap will be looked into in this Master-Thesis.

2.2 Data Quality of Crowdsourced Data

Citizen scientists can be of help in any scientific field. Although some might say that there are criteria for it to count as citizen science (Robinson L.D., Cawthray, J.L., West, S.E., Bonn, A., & Ansine, 2018), there are many different types of citizen science. For example, the participants can help collect data or do the research themselves. Citizen science categories can be active (OpenStreetMap, n.d.) or passive (Bundesamt für Gesundheit BAG, 2020c) and voluntary or involuntary. Citizen science categories can include a data framework (Strobl, Etter, Van Meerveld, & Seibert, 2019). Citizen science categories can require training. The collection of data can be implicit or explicit. The amount and type of information collected can differ.

As the name of the website <https://citizensciencegames.com> already suggests there are citizen science games to be found. Some are quite simple; others are very complicated. One of them is called *Foldit* where people all around the globe try to

restructure protein models. Because of citizen science new algorithms of how proteins can fold were discovered. The quality of the results rivals the experts' and can lead to scientific breakthroughs (Eiben et al., 2012; Khatib et al., 2011).

A part of Citizen Science is also the data collection. When citizens voluntarily collect data for a project, then this data can be called *crowdsourced data*. If there happens to be a spatial component in the data, then it is the subtype of crowdsourced data namely *volunteered geographic information (VGI)*.

VGI can be explicit or implicit. Explicit geographic information is when the contributors lay the focus on mapping the content of the data (e.g. appending information on OpenStreetMap). Implicit geographic information is when the location of the content is minor matter (e.g. geotagged microblogs) (Craglia, Ostermann, & Spinsanti, 2012).

Crowdsourced data can be divided into either active or passive contribution and into data with framework (e.g: Hiking trails) and without framework (sky / stars). In the case of the app "Zecke", the VGI falls into the same category of very active crowdsourced GI and leans more towards having a framework than not (See et al., 2016). *Senaratne* differs between implicit and explicit, which is like active and passive contribution. Implicit VGI brings higher quality concerns than explicit VGI, yet more research has been done to resolve quality issues with explicit VGI. However, a higher utilization of implicit VGI in geospatial research has been seen. Therefore, nowadays more attention should be on resolving quality issues with the implicit VGI (Senaratne, Mobasheri, Ali, Capineri, & Haklay, 2017).

As data gets collected uncertainties arise. There are four different classifications of spatial data uncertainties: (a) *the inherent uncertainty in nature*, (b) *uncertainty due to limitations of human cognition of the natural world*, (c) *object measurement error*, and (d) *uncertainties arising through spatial analyses and spatial data processing* (Shi, 2010). In the case of the collected tick data, most of the existing uncertainties arise due to a mix of *measurement errors* and *inherent uncertainty in nature*. Most humans do not realize when they have been bitten by a tick, thus it is extremely difficult to know when and where exactly they were bitten.

There are many ways to assure high quality data in research in general, yet significantly less for crowdsourced data. Questions may be asked about quality assurance like development of credibility of crowdsourced data. Fortunately there are a variety of broad approaches to secure quality assessments in this field (See et al., 2016).

Using crowdsourced geospatial data scientifically might raise concerns given that people report data without attention to scientific standards. The collected tick data by the app "Zecke" contains all the typical data quality issues that crowdsourced data brings: heterogeneous contribution, uncertain positional records, different data capture devices, etc (Goodchild & Li, 2012).

Some of the crowdsourcing projects implemented a tool to increase the likelihood of good quality data, such as the *CrowdWater* project app where citizen scientists took pictures of water streams, and reported the respective water level (Strobl et al., 2019). It has a feature where users could report low quality pictures or false evaluations; for example, if the same picture was reported 15 times it was deleted automatically¹. Unfortunately, there was no quality guarantee measure in the app "Zecke". However, by changing the accuracy radius, it leaves space to set our own standards for quality of the data.

The VGI collected through the app "Zecke" was available at the beginning of this master thesis yet were being updated through the progress of this paper. The used data was collected from March 2015 to 24.08.2020 and it is prone to have the same problems as other VGI, since the data structures share some related traits such as positional accuracies, temporal accuracies, variable quality of the observations, and information on participants. Albeit these expected issues, the reported data is of sufficient quality if to take only the reports with an uncertainty radius of 2'130m or smaller (90% of all reports).

The one main advantage of crowd sourced data is that it should be easily gathered by the public. However, the quality of the data might be lacking because of that.

¹ Personal communication via e-mail

Training the participants could have a big effect on the quality, however it might faze potential users, resulting in much less citizen scientists participating hence less data. Training can also be seen as an indicator of the level of complexity of the role a citizen scientist will take in that project (See et al., 2016).

When dealing with data quality separating the given data into high and low quality is important to increase the quality of the overall project. There are different ways to triage the data. Some projects require manual triaging, others can be fully automated (Goodchild & Li, 2012). A very good example of manual triaging is Wikipedia. Contributors review the articles and improve them if necessary. This can hardly be done automatically. A well-designed database can eliminate unwanted, hence low-quality inputs automatically already before the document gets reported.

The crowdsourced tick reports fall in-between the manual and automatic triaging, since the app "*Zecke*" reduced the possible temporal inputs on the device but had no limit to its spatial component. The radius could be increased immensely, and the centre could be placed anywhere on the world. After the initial triaging of the device the rest was done mostly manually.

A lot of research exists on how to prevent low quality data when crowdsourcing, yet very little papers can be found on how to deal with uncertainty once low-quality data is obtained. One example of the latter is a paper by Amatulli (Amatulli, Pérez-Cabello, & de la Riva, 2007). The projects talk about how wildfire occurrences can be mapped with the uncertainty of where the ignition point location was. There they came up with several ways to calculate the density of the points. One of them is the kernel density estimation (KDE) with a *moving window* of a fixed size. As explained by Silverman in 1986, the KDE calculates the density of an area by creating a round moving window around each point. All points lying within the windows get weighted as to how close they are to the centre. The closer to the centre the higher its impact. It gradually decreases outwards until it hits zero outside the windows. The amount of points and their proximity to the centre determine the density for that specific location. The window moves to the next point and the cycle repeats. The size of the moving window is defined by the search radius of the kernel. This method is particularly useful when the data consists of finite individual point locations (Bailey,

T.C., & Gatrell, 1995; Koutsias, Kalabokidis, & Allgöwer, 2004). However, there is little search done on spatial density estimation models under uncertainties.

3. Data settings

In the following paragraphs, the data set is being shown. In addition, a standard report gets shown in order to explain why certain data is not being taken into account. Then at the end, some old collected data will be discussed.

3.1 Used data

The original data was received on the 24th August 2020 and started with 47'723 tick reports. After only taking reports originating in Switzerland, deleting the standard reports, and removing outliers whose inaccuracy radius were above 2'130 meters 40'579 tick reports are left. That is a 14.967% difference in the number of ticks, which is a total of 7'144 non-considered reports.

The original data shows when the report was sent exact to the minute, the geographical, temporal inaccuracy of the report, and the coordinates among other things. The temporal inaccuracy of the report shows if the tick bite happened on the day the report was sent, the day before, or earlier. About 80% of all reports were done on the same day the tick bite happened or the day after. Based on [Figure 4](#) it was assumed that most people report tick bites in the evening hours, because it can take some time after the initial bite to spot them on their bodies. This is because of the ticks' small size. Therefore, the reported time is not considered.

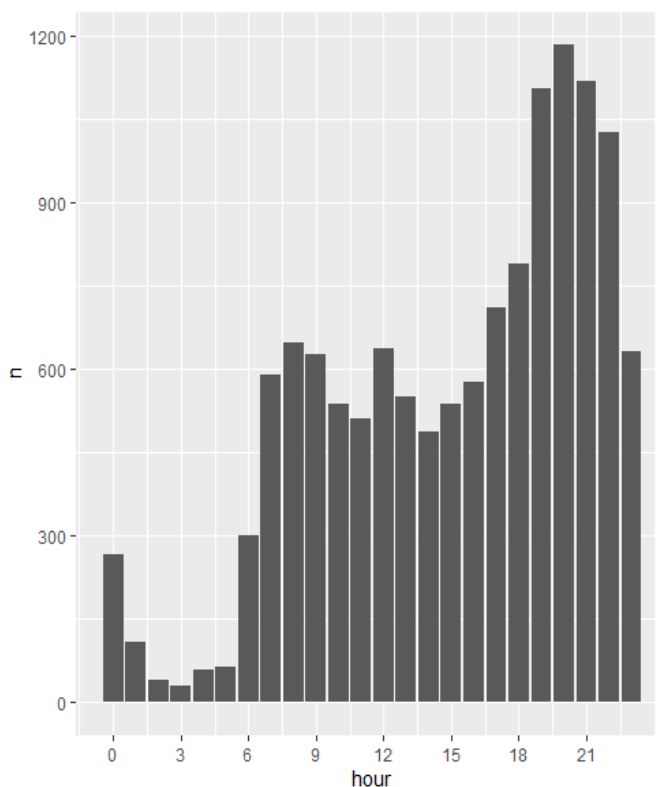


Figure 4: Daily reported hours

3.2 Standard report

The name (*Wer: Ich*) and date (*Wann: Heute*) would remain constant on each device unless changed. A report could only be made if a body part and a location was chosen. For the body part the user had to choose on which one he/she got bitten; there are no standard settings possible. For the location the situation changes depending on the device and if geolocation is on or off. In a standard report the location and uncertainty radius of the tick bite remained unchanged and were determined by the device. The radius of uncertainty of a standard report on a device is determined by the size of its screen, the bigger the screen, the bigger the radius. A standard tick report with geolocation switched off and selected body part can be seen in [Figure 5](#).

A standard report could deviate the outcome of the KDE, a selection had to be made. As it is unknown if a standard report was done on purpose or by accident. All standard reports were filtered out. To find out what a standard report looked like, a small experiment was performed. Eleven smartphones were gathered at one place. Each device reported a tick bite with and without geolocation on/off and without changing the default radius set by the smartphone. Since these were all reports from the same location and sent within a minute, the standard reports were found for at least 11 smartphones. From the received information the default radius could be extracted which was later used to filter out all the standard reports with those radii.



Durch Ihre Meldung dieses Zeckenstichs leisten Sie einen wertvollen Beitrag zur Zeckenprävention. Andere Benutzer dieser App können den Stichort auf der Karte sehen.

Figure 5: Standard report

4. Methodology

The main methods used for this project were the kernel density estimation (KDE), and the Monte Carlo Simulation (MCS). The KDE was used to find out the density of the given data points. Whereas the MCS served the purpose of creating more points in similar locations to simulate cases where the point was not located in the centre of the uncertainty radius. To use a non-spatial way of utilizing the uncertainty, the normal KDE was given a weight of the uncertainty.

To take these uncertainties into consideration 4 different density estimations were used. The first one is a simple KDE without weighting, the second with weighting the inaccuracies of the tick reports and the third and fourth are slight modifications of each other, whereas the main idea is to create MCS and use those to calculate the KDE.

4.1 Conventional KDE

In this project the KDE search window is a square and is specified by a unit of meters. The reason to use a squared moving window is to model a fixed extent more easily with a specific number of cells. This made comparing the different KDE's of the same location easier.

Additionally, to the kernel size, there are different types of kernel weighting methods. The used kernel is the Gaussian (often also known as the *normal*) kernel. It takes all points within the windows into consideration and weights them according to the standard deviation. This means that points near the centre have proportionally a higher degree of influence on the density than points further away. Other methods weigh all the points inside the window equally (uniform) or have the influence increase statically to the centre (triangular) (Silverman, 1986).

4.2 Uncertainty weighted KDE

Two sets of KDE were used. The first is a standard KDE by just using the data points. The second is to take the inaccuracy radii of the points into consideration. To ensure

that the point with the smallest radius gets weighed the highest and vice versa, the inverse was calculated using the following formula:

$$Y=1-(X - \min(X)) / (\max(X) - \min(X))$$

This formula ensures that all tick reports' inaccuracies fall within the range of 0 to 1, where 1 is giving to the smallest inaccuracy and 0 to the largest inaccuracy.

Example:

$$X = 300m \quad \min(X) = 2m \quad \max(X) = 2130m$$

$$Y=1 - (300m - 2m) / (2130m - 2m) = 0.86$$

In order to change the inaccuracy and therefore the weight, one could not just increase the radii. If the radii would be increased ten times, then the $\min(X)$ and $\max(X)$ values had to be adjusted. This in turn would negate the increase of inaccuracy because the relationship between for example 3 to 4 is the same as 30 to 40. Another suitable option could have been to not adjust the values. Then the $\max(X)$ would be smaller than $10X$. That would produce numbers bigger than 1 and favour tick reports with bigger inaccuracies over smaller ones, which is not the desired result. By taking the formula shown above and dividing the resulting numbers by 0.5, 1, 2, and 10 the problem was solved. The assumption is made that if a tick report had a weight of Y then the weight Y should be decreased by the factor K if the radii is increased by the same factor K .

$$X = 300m \rightarrow Y=0.86$$

$$10X = 3000m \rightarrow Y/10= 0.086$$

By comparing the first and the second KDE map under [Figure 25](#), the impact of the inaccuracy radii is shown.

4.3 Monte-Carlo-KDE

To help find out which method is best to simulate different outcomes of the KDE, the MCS was used. The MCS creates another instance of a data set similar to the original one. It does so by estimating the properties of the distribution by combing the

existing distribution of random samples (van Ravenzwaaij, Cassey, & Brown, 2018). In this project however, the properties or boundaries were defined by the inaccuracy radius of each tick report. The amount of which MCS could move within the radii was regulated with a Gaussian distribution around the original tick report (example shown in [Figure 6](#)). The assumption is being made that the user tries to set the centre to the closest most probable tick bite location, yet due to human error could be mistaken. This enables the simulation instances where the happening of the tick bite is not situated in the middle of the inaccuracy circle. This in combination with the KDE enables to incorporate the inaccuracy of the tick reports into the density estimation.

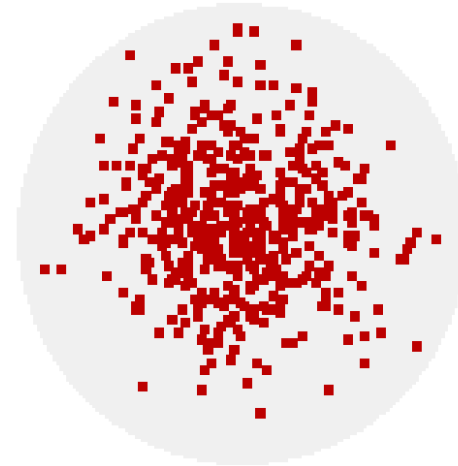


Figure 6: Gaussian distribution of MCS (source: https://simion.com/info/monte_carlo.html)

Here two other methods were applied together with the conventional KDE. The first method is referred to in the following as "*Concurrent*". The second method is referred to in the following as "*Sequential*". Both methods were used to embody the inaccuracies of the tick reports into the KDE by simulating 50-times more tick reports.

4.3.1 *Concurrent* KDE

The *concurrent* KDE simulates each tick report 50 times around the original tick report with a Gaussian distribution. Every iteration creates another instance of tick reports on the same map. Once all iterations are done, the KDE will be calculated for one map with 50-times more tick reports. But because the map now contains 50-times more tick reports the density is also increased by 50-times, that is why the density of the map was then divided by 50 again to be able to compare the different maps and methods (see [Figure 7](#)).

4.3.2 *Sequential* KDE

The *sequential* KDE works in a similar way, but instead of simulating the additional tick reports on one map, each iteration will be projected onto a separate map. After 50 iterations 50 maps are created. The KDE is calculated for each map the same

way as for the conventional KDE. At the end the mean KDE map is calculated from all 50 maps (see [Figure 7](#)).

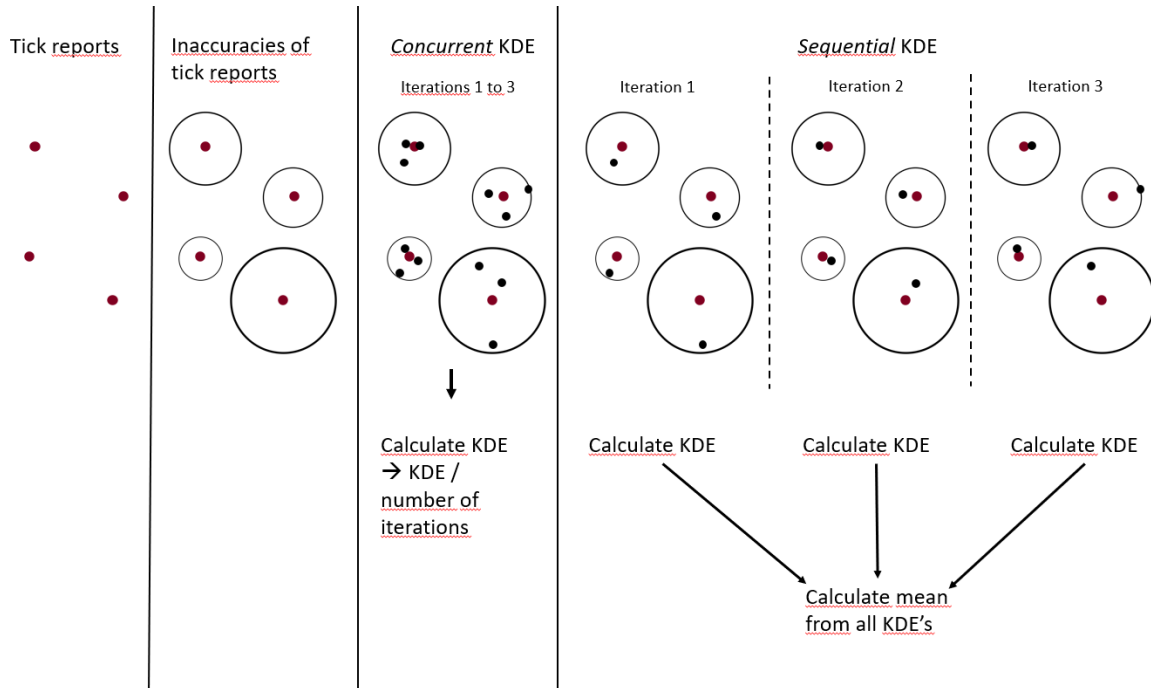


Figure 7: Monte-Carlo-Simulation for inaccuracy

4.4 Statistical Measurements

To measure the differences in between the maps and methods the following four measurements were used.

Mean Error: $ME = 1/n \sum_{i=1}^n \Delta h_i$

Mean Absolute Error: $MAE = 1/n \sum_{i=1}^n |\Delta h_i|$

Root Mean Square Error: $RMSE = \sqrt{(1/n \sum_{i=1}^n (\Delta h_i)^2)}$

Standard Deviation: $SD = \sqrt{(1/(n-1) \sum_{i=1}^n (\Delta h_i)^2)}$

4.4.1 Mean (Absolute) Error (ME/MAE)

Mean Error: $ME = 1/n \sum_{i=1}^n \Delta h_i$

The ME calculates the difference between raster A and raster B and adds/ subtracts the distance from raster A to raster B. Then it divides by the resulting number by the amounts of n-values.

Mean Absolute Error: $MAE = 1/n \sum_{i=1}^n |\Delta h_i|$

The MAE does the same but calculates the absolute distance, therefore only adds the absolute distances (Chai & Draxler, 2014; Kenney, 1939).

4.4.2 Root Mean Square Error (RMSE)

Root Mean Squared Error: $RMSE = \sqrt{(1/n \sum_{i=1}^n (\Delta h_i)^2)}$

The Root Mean Square Error (RMSE) calculates the MAE, but instead of taking the absolute distance values, these numbers are squared. Then the resulting number is also divided by the number of units of the original raster. At the end the square root is taken, to make the number comparable the MAE. It measures how spread out the residuals are; in this project how spread out the values from raster B are compared to raster A. Where the MAE gives the same relative weight to all errors, the RMSE punishes variance as errors with greater absolute values receive a heavier weight than smaller errors (Chai & Draxler, 2014; Kenney, 1939).

4.4.3 Standard Deviation (SD)

Standard Deviation: $SD = \sqrt{(1/(n - 1) \sum_{i=1}^n (\Delta h_i)^2)}$

The Standard Deviation (SD) calculation is very similar to the RMSE. Except when dividing by n-values, here it is divided by n-1 values. Since the formulas are very similar, the resulting numbers are as well. The SD calculates the spread around the mean, while the RMSE calculates the distance between raster A and raster B, so how much does one raster from another deviate (Kenney, 1940; Papoulis, 2002).

5. Experiment

5.1 Explorative analysis

To explore multiple factors of the data set, it was sorted. The date a tick report was sent was separated into the year, month, week, weekday, and hours. The locations of the tick reports were separated into cantons/municipalities they were lying in. The population for each canton/municipality was divided by the population for each canton/municipality to normalize the data (Bundesamt für Statistik, 2020). To find the height at which each data point was set, a Z-coordinate from a height model was added (swisstopo, 2014).

5.2 Sensitivity analysis

A main part of this master thesis is to compare the impact of inaccuracy radii to the parameters of the kernel on density maps, as well as comparing different methods that take the inaccuracy radii into account. To compare the influence of the inaccuracy against the influence of the kernel parameter, small multiples were created to compare the rasters more easily. Small multiples allow for a visual comparison and help orient between the rasters. With the visual comparison only estimates can be made, therefore quantitative measurements (SD, ME, MAE, RMSE, including the sum of the positive and negative values) were also used.

Because the kernel parameter is a rather arbitrary value, multiple maps with different parameters were created. In the subset 4 parameters were chosen to be 100m, 250m, 500m, and 1'000m. On the scale for Switzerland, the parameters were increased to 1'000, 3'000, 5'000m, and 7'000m. To see what influence the inaccuracy has on the outcome, they were increased and decreased by a multiplicative of 0,5, 1, 2, and 10, further referred to as an *inaccuracy modifier*, whereas 1 is the original size of the radii. This led to the creation of small multiples with the different kernels and inaccuracy modifiers.

On each scale four small multiples were created, each with one method. First the normal KDE where only the kernel parameters were changed since the inaccuracy does not play a role in this method. The second method is the *concurrent* KDE,

which can be seen in [Figure 8](#). The third is *sequential* and the fourth method is the *weighted* method. Both, the *concurrent* and *sequential* method had 50 iterations. As a *concurrent* method simulates multiple tick reports on a single map, the KDE would have been 50 times higher. To correct this, the resulting raster of this method was divided by 50 again.

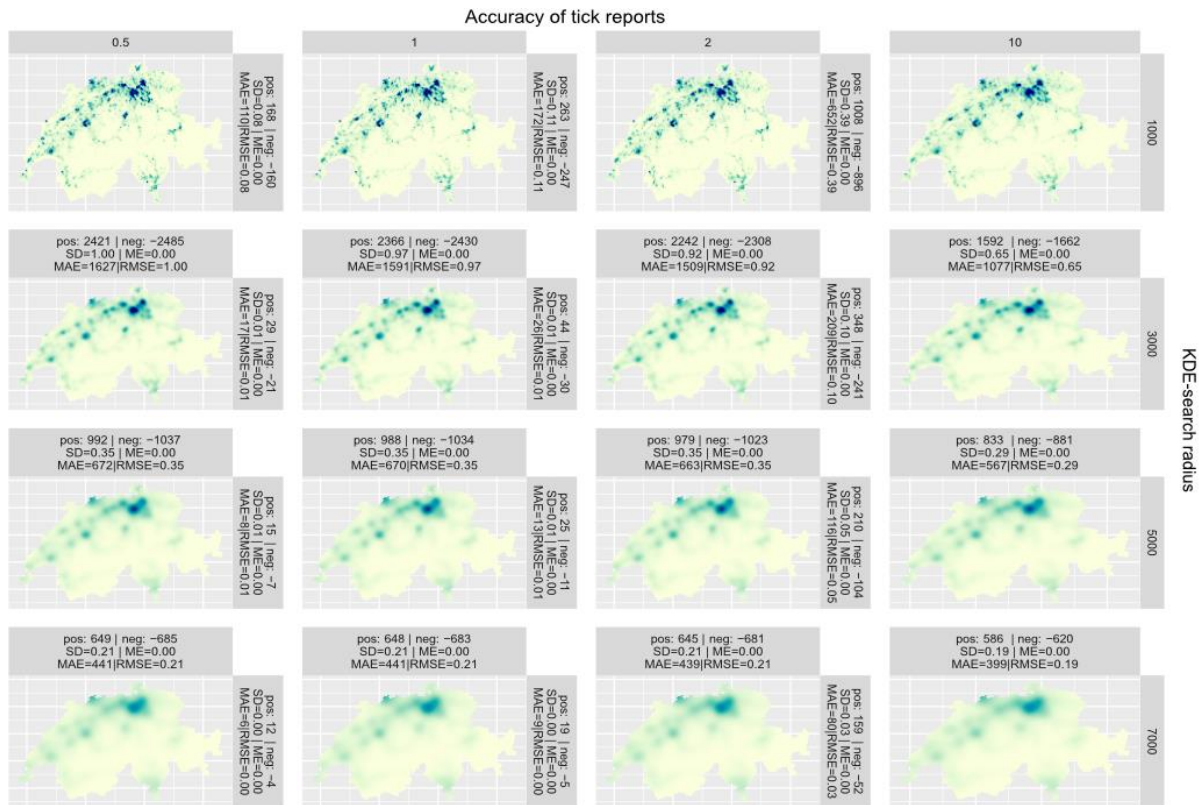
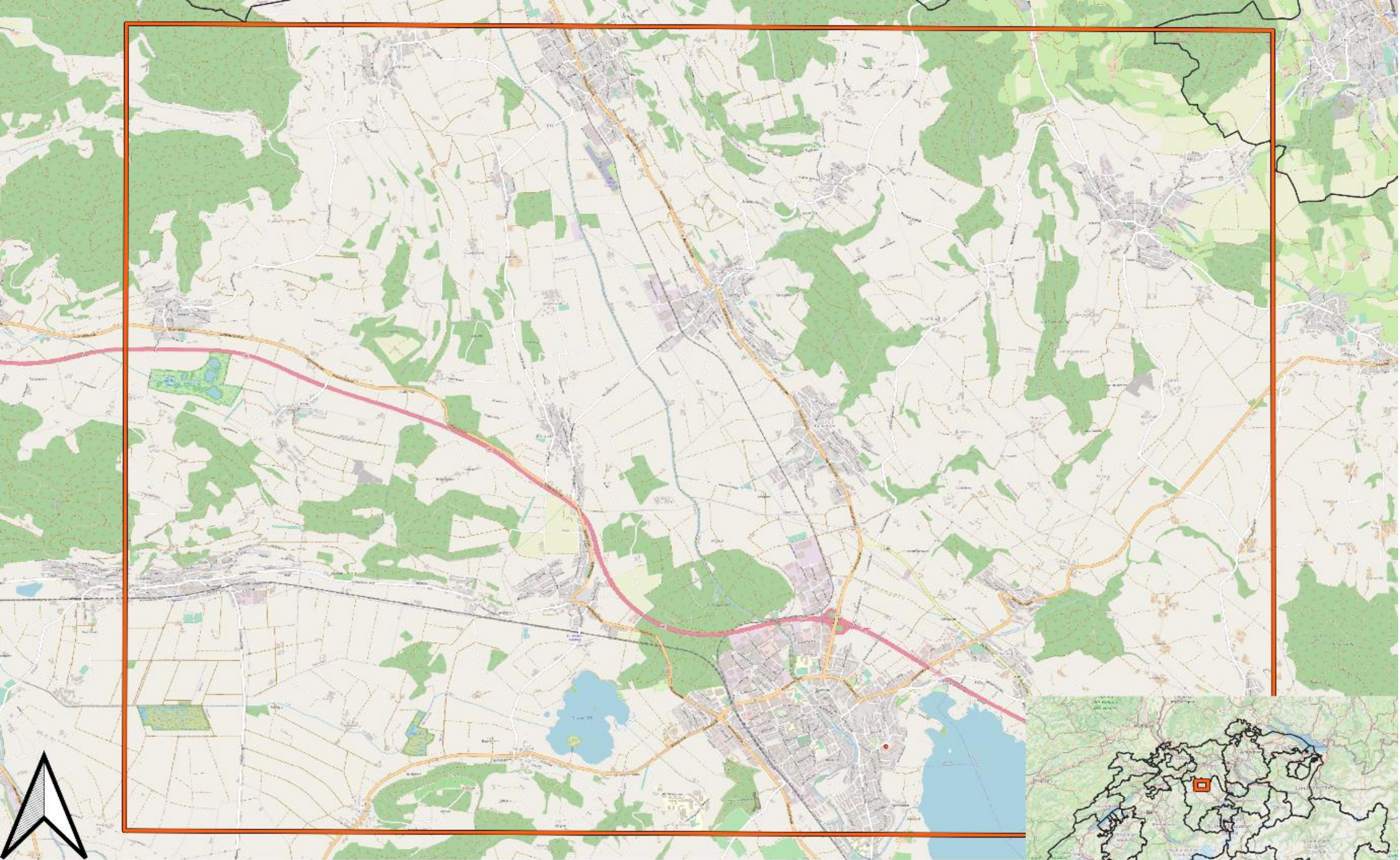


Figure 8: National *concurrent* KDE

5.2.1 Local scale

For the analysis, two investigation areas were used: the national and local scale. The local scale is a subset of the national scale and features a rural area. The selected rural area had an extent of 8.9km high and 12km wide and is situated around Knutwil, Büron, Geuensee, and Sursee in the north of the canton Lucerne (see [Figure 9](#)). It is in the midland of Switzerland, had an average looking amount of tick reports (n=198) and didn't have any high mountains in the area. Mountains could have distorted the tick locations by accidentally choosing the midpoint of the report being above 1'500m. Forests were close by each town, where ticks are commonly found, and the local people use as recreation areas.

The kernels were chosen to be 100m, 250m, and 1'000m, as they were close to some peaks of the inaccuracy of reports. The 500m was chosen for the opposite reason and as to have a gap between 250m and 1'000m. The cell size for the local scale is a hectare, resulting in the following maps to show the tick density per hectare.



Extent of subset

Extent: 12km wide / 8.9km high
Background source: OpenStreetMap

The subset is situated in the north of the Canton Luzern.

Figure 9: Local extent

5.2.2 National scale

The cell size for the national scale is 3.828km², resulting in the following maps to show the tick density per 3.828km². The reason for this seemingly arbitrary number is that this results in the same amounts of cells for the national extent as for the local scale. This in turn makes it easier to compare the numbers between the local and national scale.

5.3 Differential analysis

Once the small multiples were created, the difference within the small multiples and across the methods could be determined.

To analyse the variance within the small multiples a map was chosen and one adjacent differential map to the right/left and top/bottom of it. The differential map helps in combination with the statistical measurements to distinguish the deviation between the maps within the small multiples.

For the analysis of the difference between the methods, maps with the same kernel parameter and inaccuracy modifier were chosen for both methods. The difference between those two maps was calculated including the used statistical measurements.

6. Results

The results are structured into two main parts. The data exploration is separated into the temporal and spatial part of the data. Once the data has been examined, then the created matrixes will be set into focus. To get a good view of how the results vary in scale, local and national extents were used.

6.1 Explorative data analysis

6.1.1 Temporal

To get familiar with the tick data, some data exploration needed to be done. The data was collected starting in 2015. In the first 3 years about 1'800 and 3'800 tick bites were reported. The first major increase of tick reports happened in the year 2018. The app "*Zecke*" got a lot of attention and more than 9'000 tick reports were sent that year. The app retained its popularity in 2019 and still more than 8'000 tick bites got registered. Around 13'000 tick reports happened in 2020 up until the 10th August (see [Figure 10](#)).

As the temperature rises throughout the year, the tick reports also increase. An interesting correlation is that even in winter there are still tick reports happening, but much less than during other seasons. The high peak of reports is from May to June and declines dramatically in July across all years. (see [Figure 11](#)).

Fifty percent of all tick bites get reported between the weeks of 21 up until 27, which is comparable to what Porter et al. (2019) has examined. The 21st week of the year is around mid-May. Also, almost all reports (90%) have been sent between week 12 (mid-March) and 36 (beginning of September) of the year (see [Figure 12](#)).

[Figure 13](#) shows the amount of tick reports depending on the days of the week. Weekday 1 represents Monday and Weekday 7 represents Sunday. It is visible that during the weekday the amount of tick reports coming in are relatively equal. A slight increase happens on Saturday, but the highest amount of reports happens on Sunday. The most likely explanation is that many people have more time to go out into nature during weekends than weekdays.

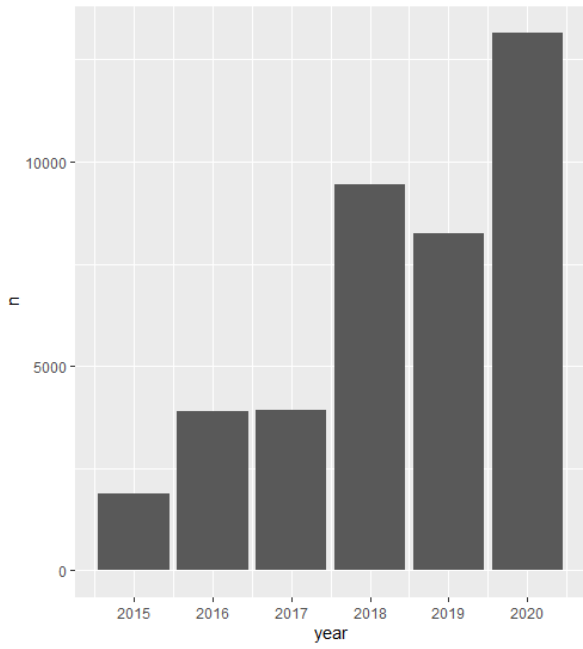


Figure 10: Count per year

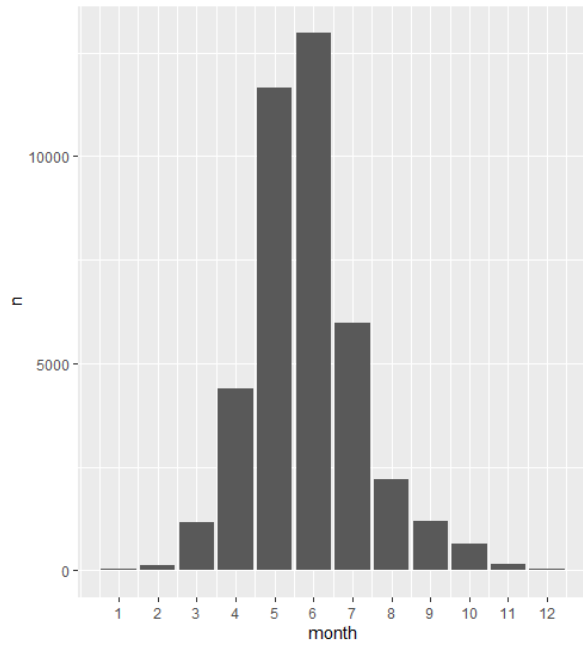


Figure 11: Count per month

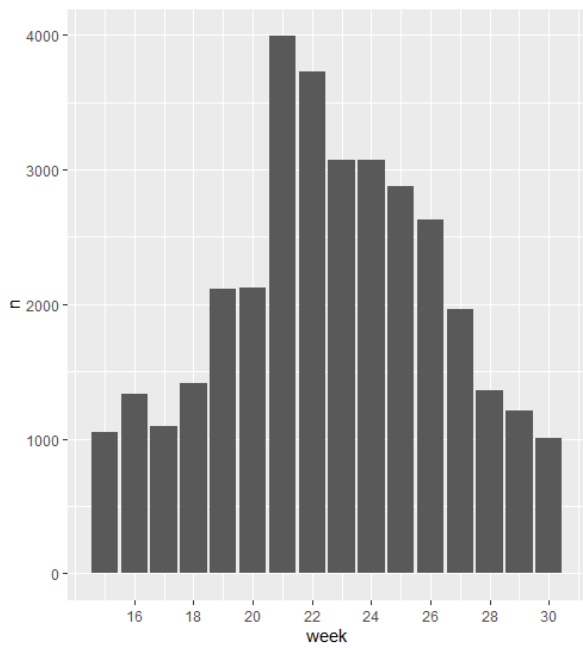


Figure 12: Count per week

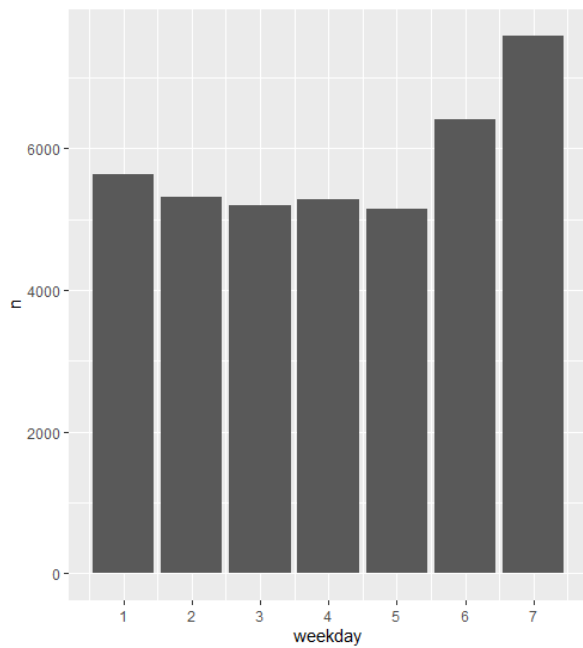


Figure 13: Count per weekday

6.1.2 Spatial

Ticks can be found almost everywhere in Switzerland. To find out where they are and how they are distributed, some plots were created.

When comparing [Figure 14](#) and the areas with urban character (Bundesamt für Statistik (BFS), 2014) it is visible that ticks appear everywhere in Switzerland where bigger groups of people are. This is logical, because without humans, no tick reports could be made. Big clusters are visible around Lausanne, the whole canton of Zurich, Basel, and Bern.

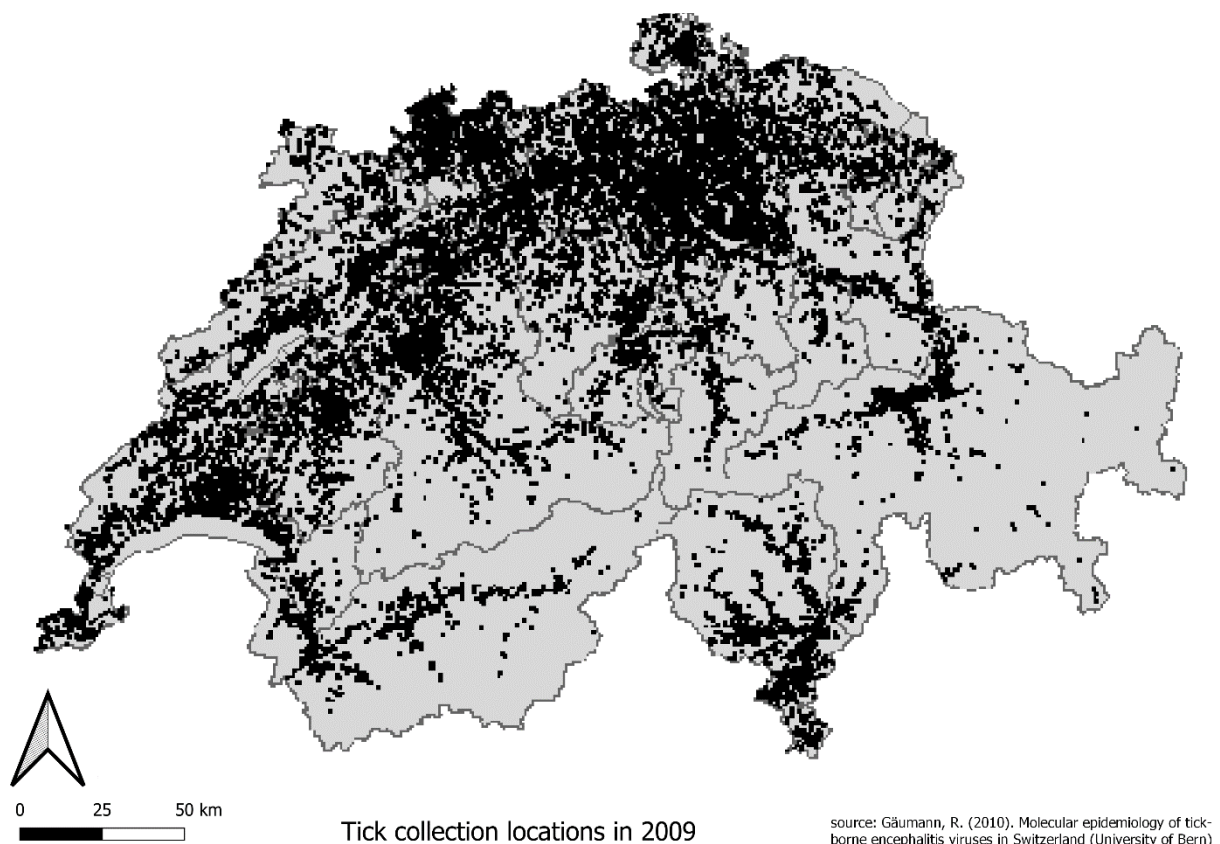


Figure 14: Crowdsourced data

All 26 cantons show reports of tick bites (see [Figure 15](#)). Some cantons have a lot more tick bites than others. This is partially due to a higher population, but maybe also more tick reports. To be able to be sure, a normalized plot was created. (see [Figure 16](#))

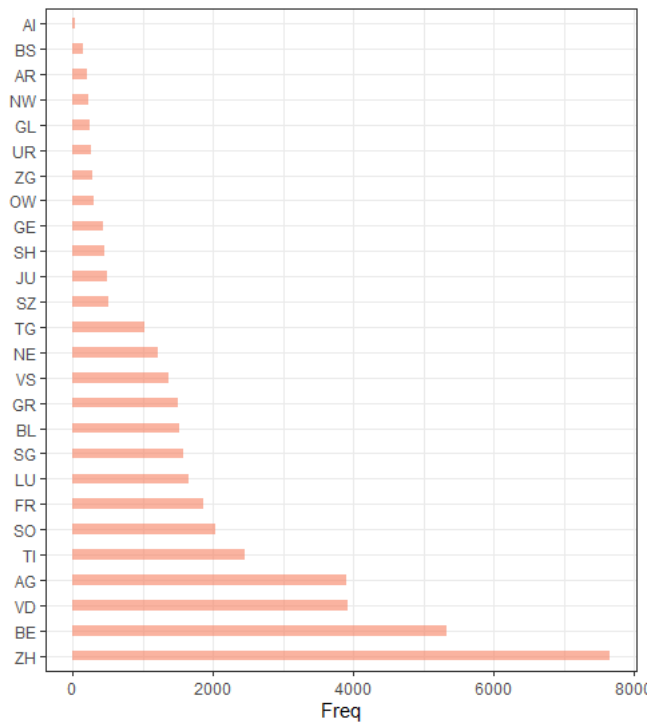


Figure 15: Ticks per canton

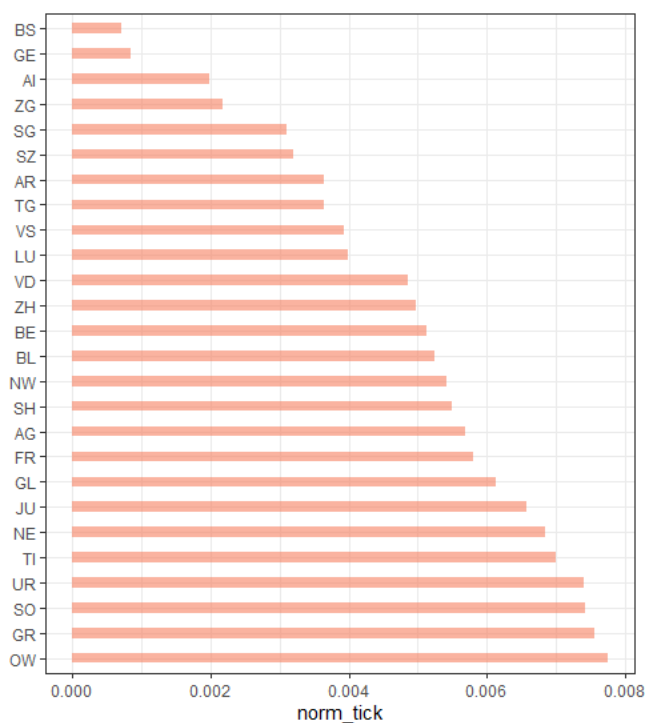


Figure 16: Ticks per canton, normalized

Basel-City almost shows no ticks per population. This is due to several factors. It is rather small and the canton only holds a few green spaces where ticks could actually be found. Ticks like a humid and moist climate (Perret, Guigoz, Rais, & Gern, 2000). The city-canton is also quiet dry due to its geographical location. It is situated between the Jura mountains and the Black Forest, which causes a funnel for the weather to go through. This in turn often creates a high pressure field, where it seldom rains (GEO-NET Umweltconsulting GmbH, 2019).

Grisons is almost the complete opposite of Basel. It is very large, has lots of forest and shows one of the highest ratio of ticks per population of the cantons; the same could be said for the canton of Obwalden, Solothurn and Uri, although they are smaller, but still big compared to Basel-City (see [Figure 16](#)).

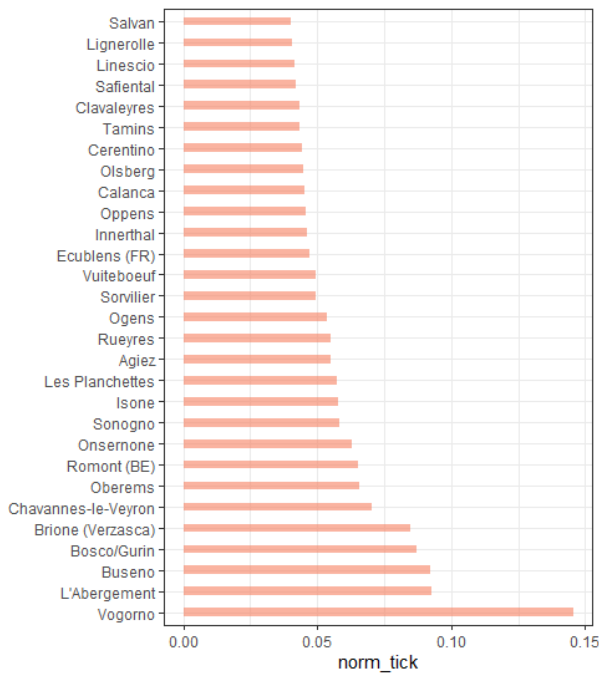


Figure 17: Ticks per municipality, normalized

What is interesting to see is that in both [Figure 15](#) and [Figure 16](#) the canton of Ticino scores relatively high. On the official webpage of the BAG it is suggested that every canton besides the canton of Ticino and Geneva should get vaccinated against the FSME (Bundesamt für Gesundheit BAG, 2020a). It seems that there are few ticks which carry the FSME virus in these cantons, even though there have been confirmation on FSME reports, but there are a lot of other ticks in Ticino (see [Figure 2](#)). The figure shows

municipalities with at least 1 confirmed case of FSME between 2009 and 2018. It is unknown to the public which municipalities had how many confirmed cases. It would have been interesting to see how this data would have compared to the received tick reports. But because of privacy reasons this is not possible.

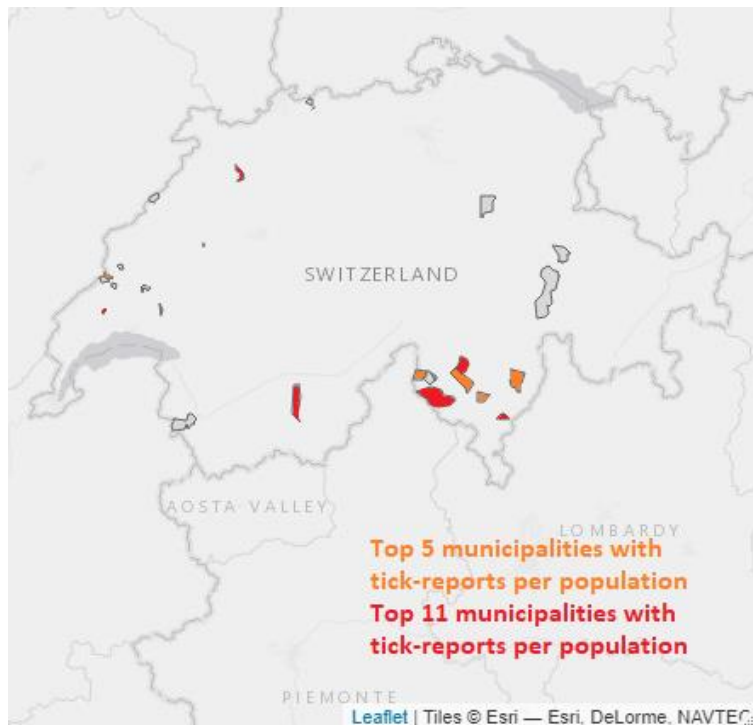


Figure 18: Top 11 municipalities

where the FSME risk is supposedly low (see [Figure 18](#)).

In [Figure 17](#) it can be seen that 29 out of 2202 municipalities in Switzerland have a higher tick report ratio per population than 0.04.

Surprisingly the municipalities with the five highest ticks per population were all from the southern part of Switzerland (with one exception); especially a lot from Ticino,

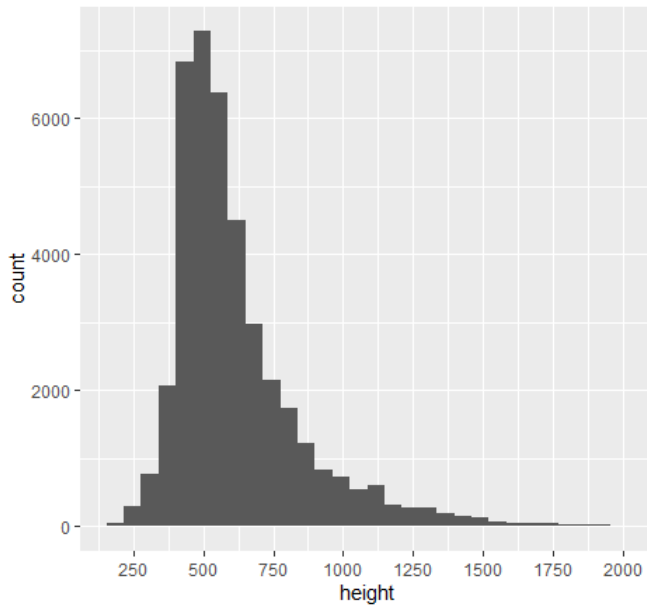


Figure 19: Height distribution

Also, few tick reports were made on the mountainous locations. The median height was 555 meters above sea level. Fifty percent of the reports happened between 466m and 697m, which is the altitude of the midland, whereas only 5% of reports surpassed 1'041m. Some outliers were even above 2'000m; a height where ticks are generally not seen (see [Figure 19](#)).

When looking at the accuracy of tick reports, a few noticeable spikes are happening. (see [Figure 20](#)) The second highest spike has the highest accuracy which is good.

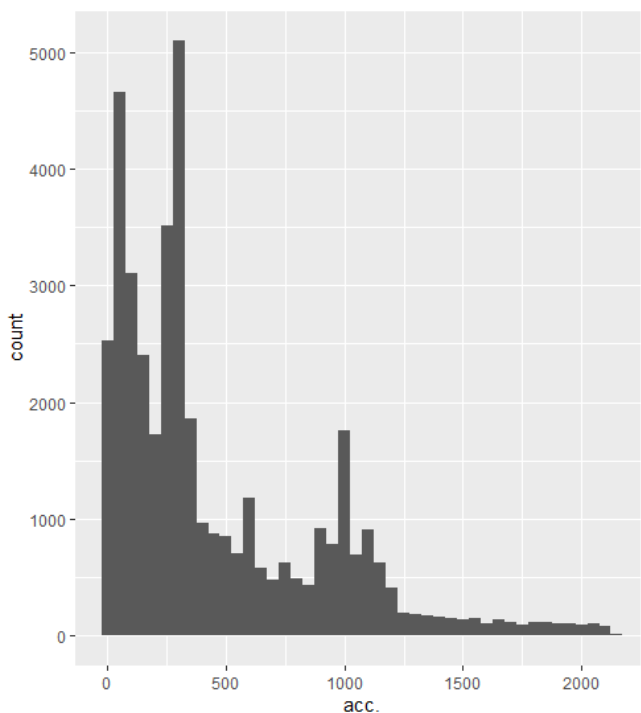


Figure 20: Accuracy distribution

This means that a lot of users are certain of the location where they have been bitten by a tick. The highest and third highest peaks ($\pm 300m$) still have a relatively high accuracy, although these were close to the standard radii of many smartphones with their geolocation turned on. The spikes with the lowest accuracy are just around 1'000m, which is also close to the standard report of at least 7 smartphone devices with their geolocation turned off. Fifty percent of tick reports lay between 122m and 715m and the median is 301m.

6.2 Sensitivity analysis

In the following small multiples, the lower map is subtracted from the upper map and the right map is subtracted from the left map. As the small multiples are getting described column and row wise, the differences between the maps are being referred to with the following code: [Kernel size, Uncertainty modifier]-[Kernel size, Uncertainty modifier].

The difference between each adjacent map was calculated by comparing the sums of all positive and negative pixels in addition to calculating the standard deviation (SD), mean error (ME), mean absolute error (MAE), and the root mean square error (RMSE). All maps are normalized in the sense that they contain the same amounts of cells/pixels, so a comparison can be made even between the local and national scale.

6.2.1 Local Scale (Sursee region, LU)

Figure 21 shows the extent of the local scale in red and the weighted KDE with a kernel of 250m. Nearly all clusters are situated near and around the local recreational areas. The ones not in the forest are inside or right at the edges of the villages. Quite often the density inside the villages doesn't reach zero, which means that ticks can also be found inside local gardens or other populated green spaces just like Mulder et al. (2013) mentions. Interestingly the density of tick reports sustains.

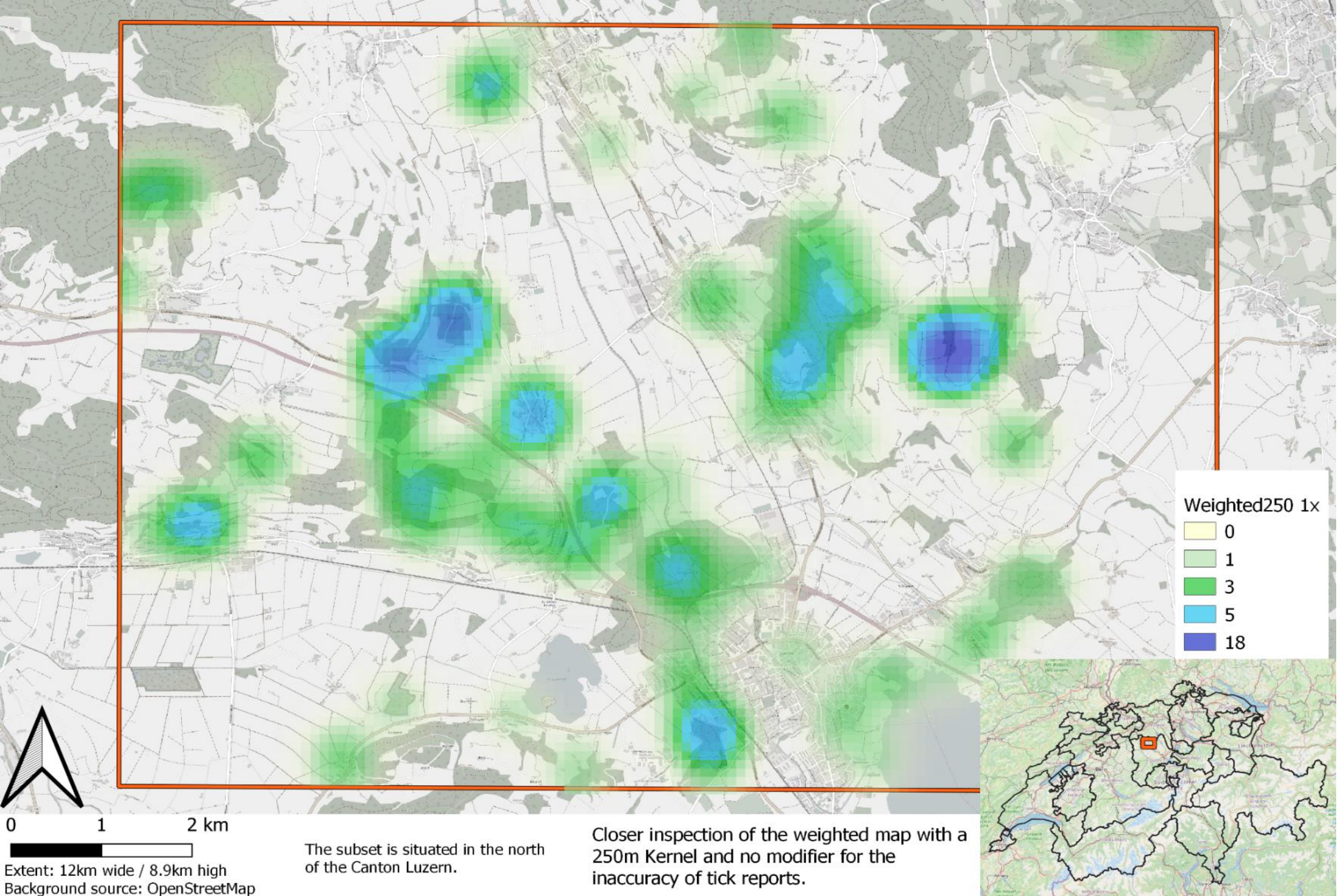


Figure 21: Local extent, weighted 250kernel

Normal KDE

When the normal KDE of tick reports on a local scale gets calculated, the area with high densities are getting smaller and denser as the kernel decreases. The SD, MAE & RMSE decrease between the maps as the kernel increases. The difference [K100,U1]-[K250,U1] has bigger SD, MAE, and RMSE then in [K250,U1]-[K500,U1], which in turn shows greater differential measurements than [K500,U1]-[K1000,U1]. Clusters are clearly visible with a kernel of 250m and 500m. With a 100m kernel the clusters turn into single dots. No clusters can be identified with a kernel of 1'000m (see [Figure 22](#)).

The KDE-search radius also varies between 100, 250, 500, 1000 meters.

"pos" respectively "neg" show the sum of positive and negative changes between the maps.

SD = Standard Deviation ME = Mean Error MAE = Mean Absolute Error RMSE = Root Mean Square Error

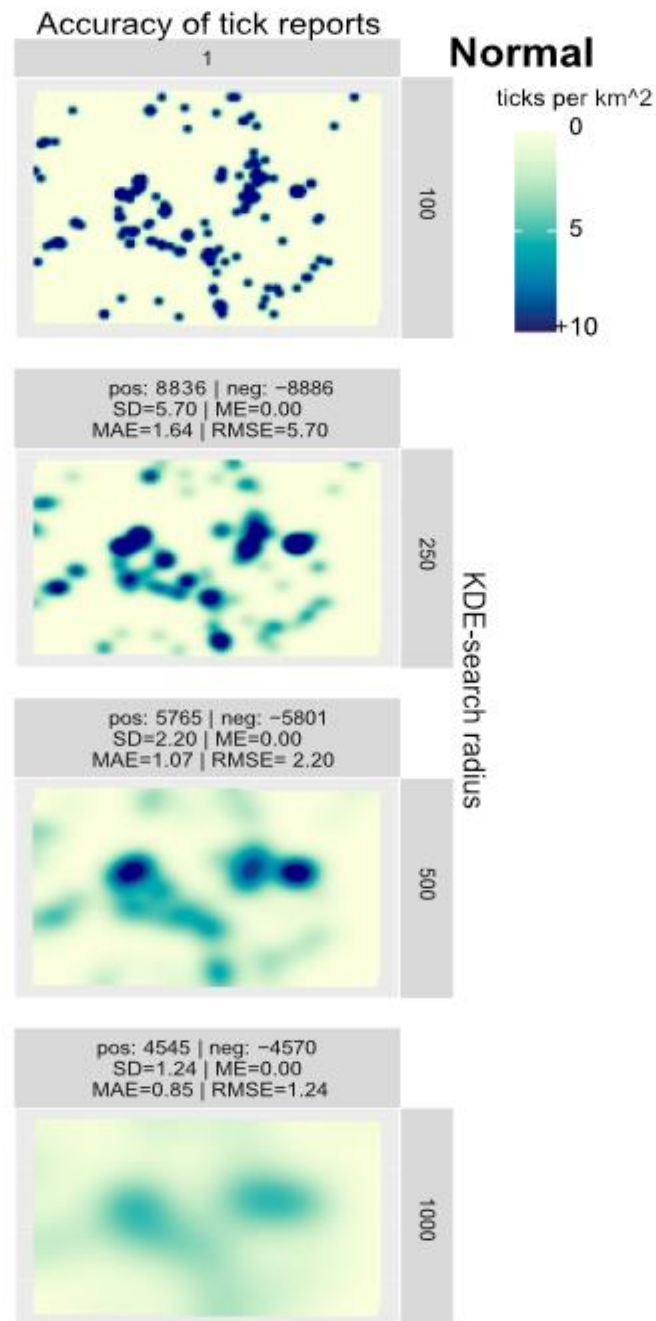


Figure 22: Local matrix, normal KDE

Concurrent KDE

Comparing the matrix *concurrent* visually column-wise, as the kernel increases the clusters get bigger, less dense, and more spread out. The difference between the maps going from top to bottom is decreasing in every measurement. A trend of having less pronounced clusters in the overall difference is visible as the kernel increases. [K100,U0.5]-[K250,U0.5] has a RMSE of 5.31 and a MAE of 1.41 whereas [K500,U0.5]-[K1000,U0.5] has a RMSE of 1.22 and a MAE of 0.83. Even when comparing [K100,U10]-[K250,U10] with [K500,U10]-[K1000,U10], the pattern persists, but the numbers are smaller.

Now looking at the matrix visually row-wise, the sums of positive and negative changes are getting bigger the higher the inaccuracy factor and the smaller the kernel gets. Yet a clear variance is only visible as the inaccuracy factor increases to 10, although this change disappears as the kernel increases to 500 or above (see [Figure 23](#)). Comparing [K100,U0.5]-[K100,U1] with [K100,U1]-[K100,U2] and [K100,U2]-[K100,U10] the change of every statistical measurement increases as the inaccuracy increases. With a kernel of 1'000m the changes become smaller when looking at [K1000,U0.5]-[K1000,U1] compared to [K100,U0.5]-[K100,U1].

Comparing the rows with the columns, the change from the 2x to the 10x modifier in the top row ([K100,U2]-[K100,U10]) is generally smaller than the change from the 100m to the 250m kernel in the most left column ([K100,U0.5]-[K250,U0.5]). Even a big increase in the inaccuracy modifier as in [K100,U2]-[K100,U10] changes the output less than the increase from 100m to 200m kernel in [K100,U0.5]-[K250,U0.5]. With a sample size of 198 tick reports most changes between kernels and inaccuracy modifications are rather small. Exceptions will be mentioned.

The accuracy of each tick report varies. In the maps below these were multiplied by the factor of 0.5, 1, 2, and 10.

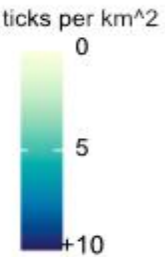
The KDE-search radius also varies between 100, 250, 500, 1000 meters.

"pos" respectively "neg" show the sum of positive and negative changes between the maps.

SD = Standard Deviation ME = Mean Error MAE = Mean Absolute Error RMSE = Root Mean Square Error

Accuracy of tick reports

Concurrent



KDE-search radius

100

250

500

1000

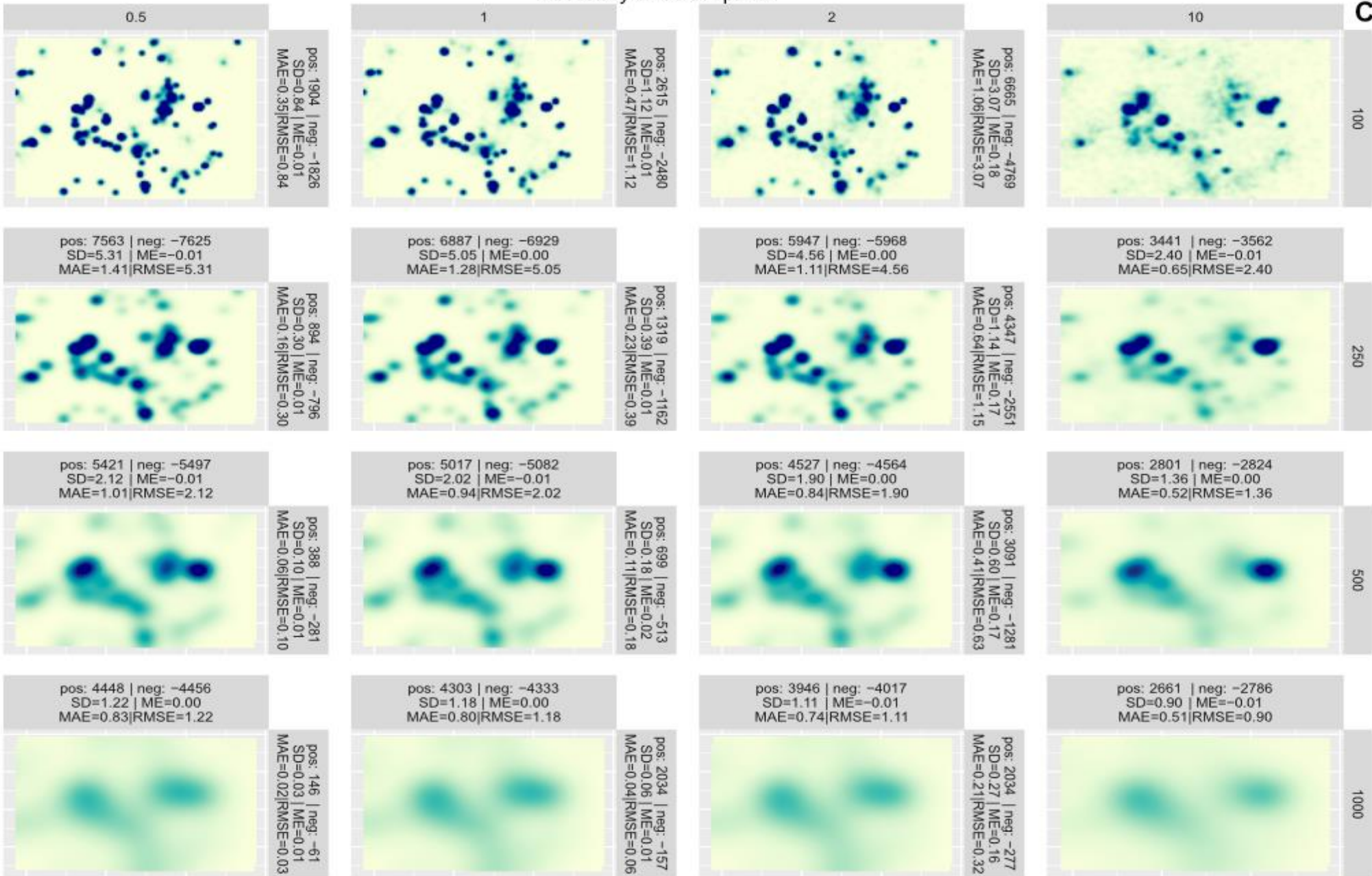


Figure 23: Local matrix, concurrent KDE

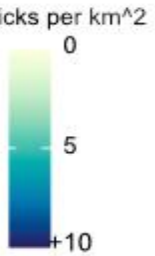
Sequential KDE

When looking at the difference between *Concurrent* and *Sequential* they are seemingly the same. In fact, the difference is so small that the numbers between them are neglectable (see [Figure 24](#)).

The accuracy of each tick report varies. In the maps below these were multiplied by the factor of 0.5, 1, 2, and 10. The KDE-search radius also varies between 100, 250, 500, 1000 meters. "pos" respectively "neg" show the sum of positive and negative changes between the maps. SD = Standard Deviation ME = Mean Error MAE = Mean Absolute Error RMSE = Root Mean Square Error

Accuracy of tick reports

Sequential



KDE-search radius

100

250

500

1000

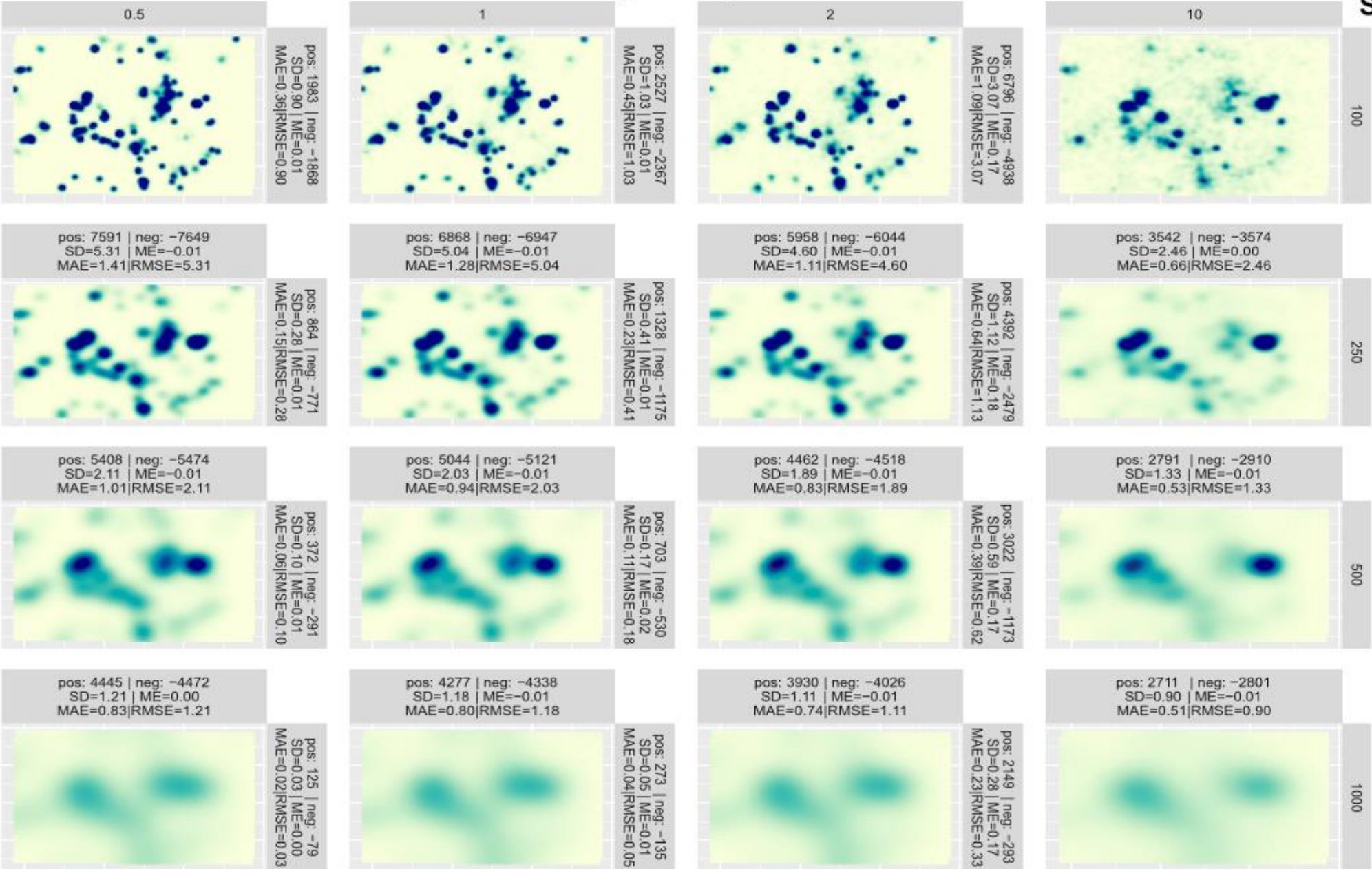


Figure 24: Local matrix, sequential KDE

Weighted KDE

With the inverse of the inaccuracy taken into account to put a weight on the KDE, the variance between the maps within the matrix increases dramatically compared to the two MCS methods. The sum of positive and negative changes between the maps decreases as the inaccuracy of the tick reports increases, but also increases as the KDE gets bigger row wise but decreases within the same column. This is different from the previous methods. The SD, MAE, and RMSE numbers start big in the top left and are getting smaller to the bottom right. The ME decreases horizontally with an increase of inaccuracy of tick reports, but does not change vertically, this is because the weight of the reports decreases by the factors of 0,5, 1, 2, and 10, hence the overall mean values decrease. This explains why horizontally there are no sums of negative value. One point to mention is that the difference between columns decreases from left to right (see [Figure 25](#)).

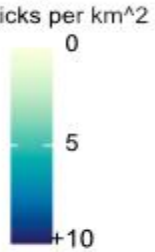
One would think that the difference between the 2x and 10x inaccuracy modifiers is bigger (for example [K100,U2]-[K100,U10]) than between 0.5 and 1 ([K100,U0.5]-[K100,U1]); visually it appears this way as well. The way the calculation of the weighting works is easily explained through an example:

The weight of a tick report with no modifier is 0.8. If the weight gets changed by the modifier 2, then the weight 0.8 gets divided by 2, hence 0.4. If the modifier is 10 the weight is therefore 0.08. The difference between 0.4 and 0.08 is 0.32. Now if the inaccuracy modifier is 0.5, then the 0.8 gets divided by 0.5, which results in 1.6. The difference between 1.6 and 0.8 is 0.8. $0.8 > 0.32$. This is why the difference decreases from left to right.

The accuracy of each tick report varies. In the maps below these were multiplied by the factor of 0.5, 1, 2, and 10. The KDE-search radius also varies between 100, 250, 500, 1000 meters. "pos" respectively "neg" show the sum of positive and negative changes between the maps. SD = Standard Deviation ME = Mean Error MAE = Mean Absolute Error RMSE = Root Mean Square Error

Accuracy of tick reports

Weighted



KDE-search radius

100

250

500

1000

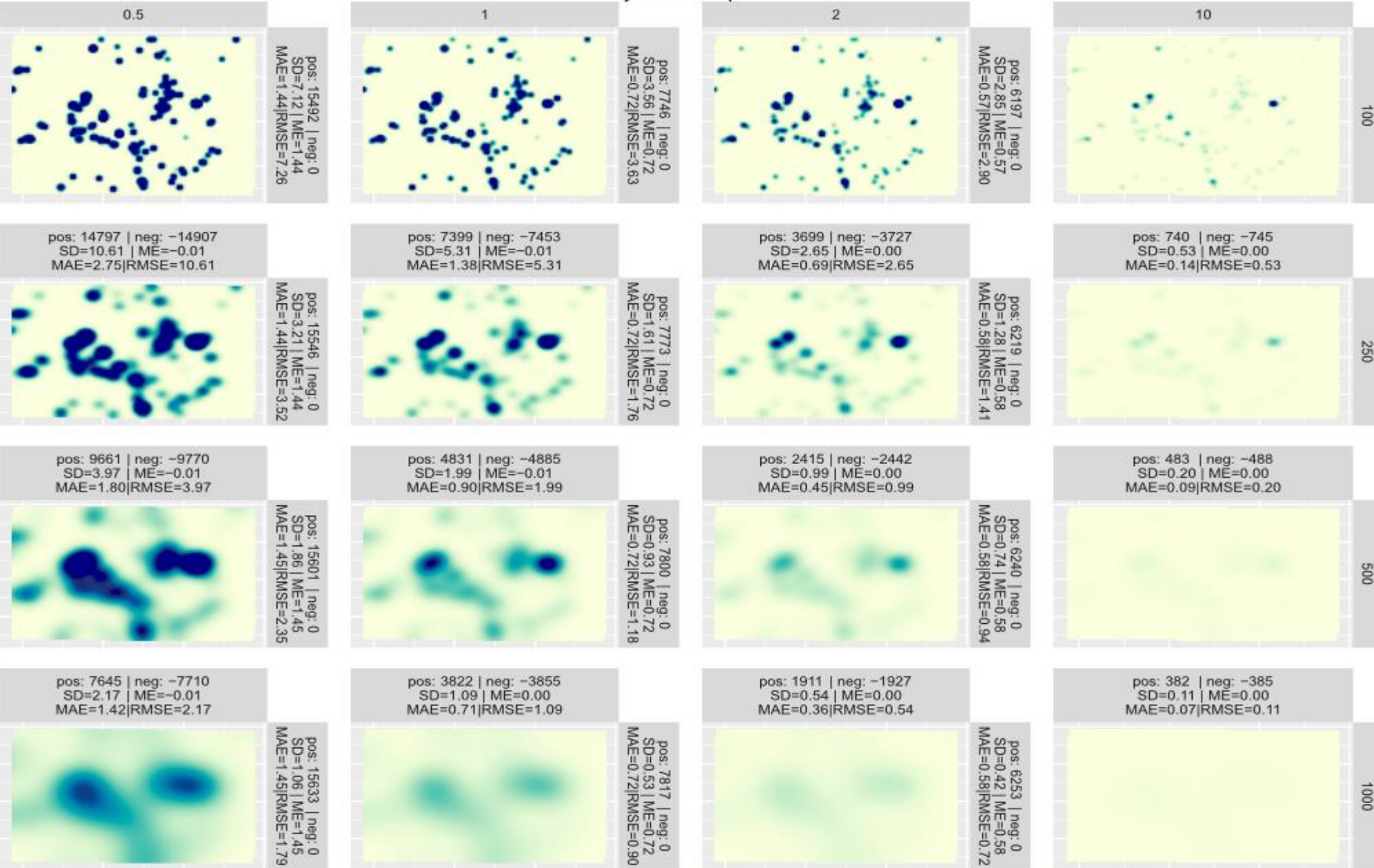


Figure 25: Local matrix, weighted KDE

6.2.2 National scale (Switzerland)

The kernels were chosen to be 1'000m, 3'000m, and 5'000m, as they were close to the multiple of 10 of the local scales. The 7'000m was chosen instead of 10'000m because the gap between 5'000m to 10'000m was too extreme; visually nothing would have been visible. Because of this I changed the previous 10x250m to 3'000m so that the increase is a steady 2'000m.

Defined clusters are less common on a large scale like Switzerland, but still occur where there are a lot of people. The Zurich area for example is one of them; many people, and a lot of reports. Though the differences between the maps are smoother than on a local scale.

Normal KDE

The clusters scatter and the differential measurements decrease as the kernel increases. On all the national scaled maps, the MAE is severely increased compared to the locally scaled maps. This is because the accumulated amounts of small differences are much bigger than when the sample size is smaller. The SD, ME, RMSE, and sums of positive and negative values are smaller on a national scale. [K1000,U1]-[K3000,U1] shows much bigger changes than [K5000,U1]-[K7000,U1] even though the increase of the kernel is constant. (see [Figure 26](#)).

The KDE-search radius also varies between 1000, 3000, 5000, 7000 meters.

"pos" respectively "neg" show the sum of positive and negative changes between the maps.

SD = Standard Deviation ME = Mean Error MAE = Mean Absolute Error RMSE = Root Mean Square Error

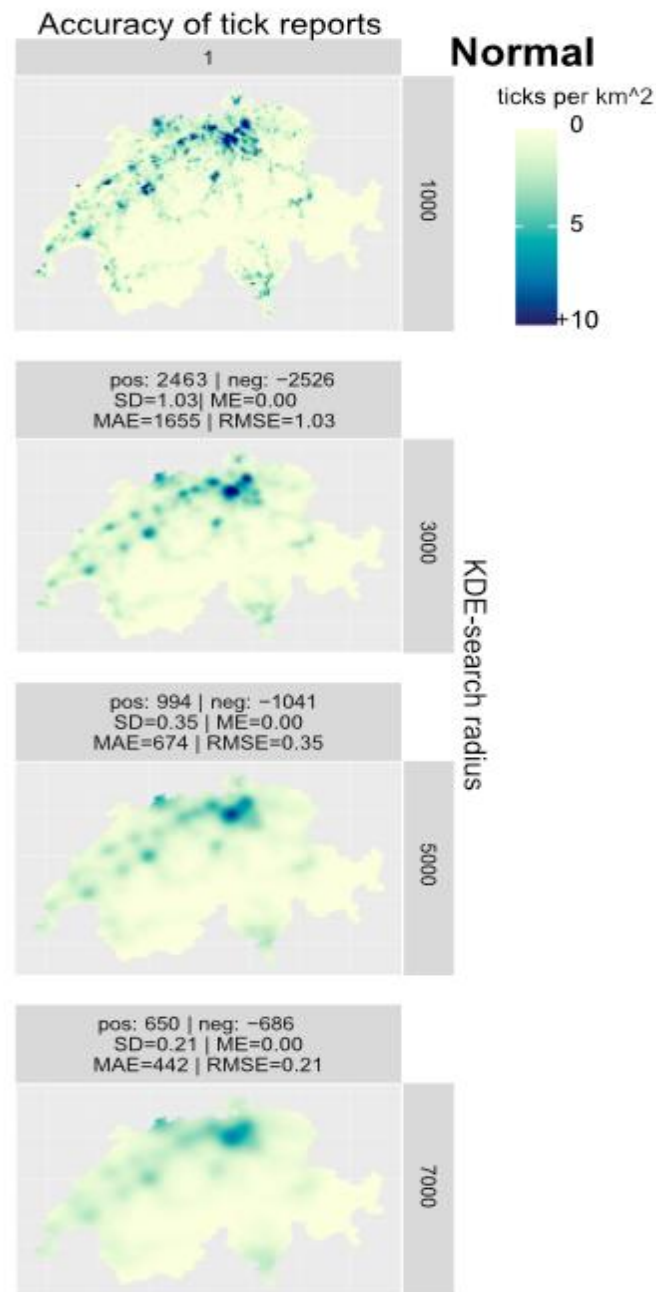


Figure 26: National matrix, normal KDE

Concurrent & Sequential KDE

As the two methods yield similar results, the descriptions of the small multiples are identical. Looking at the row of 1'000m kernel, the sharpness of the density clusters is diminishing as the inaccuracy gets bigger. With an increase of the kernel radius the clusters are getting blurred. This is confirmed by looking at [K1000,U0.5]-[K1000,U1] and [K1000,U2]-[K1000,U10]. The changes increase, but not as much as when the kernel increases from 1'000m to 3'000m, although this difference gets smaller the more the kernel increases. [K7000,U0.5]-[K7000,U1] shows almost no difference at all. (see [Figure 27](#) & [Figure 28](#)).

With the highest RMSE/SD of 1.00 with a cell size of 3.828km² no influential difference is being made by varying the kernel or the inaccuracy manipulation. The MAE is in both directions (row-/column-wise) much bigger than on a local scale, because the number of variables is higher on a national scale.

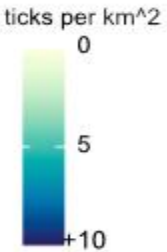
The accuracy of each tick report varies. In the maps below these were multiplied by the factor of 0.5, 1, 2, and 10. The KDE-search radius also varies between 1000, 3000, 5000, 7000 meters.

"pos" respectively "neg" show the sum of positive and negative changes between the maps.

SD = Standard Deviation ME = Mean Error MAE = Mean Absolute Error RMSE = Root Mean Square Error

Accuracy of tick reports

Concurrent



KDE-search radius

1000

3000

5000

7000

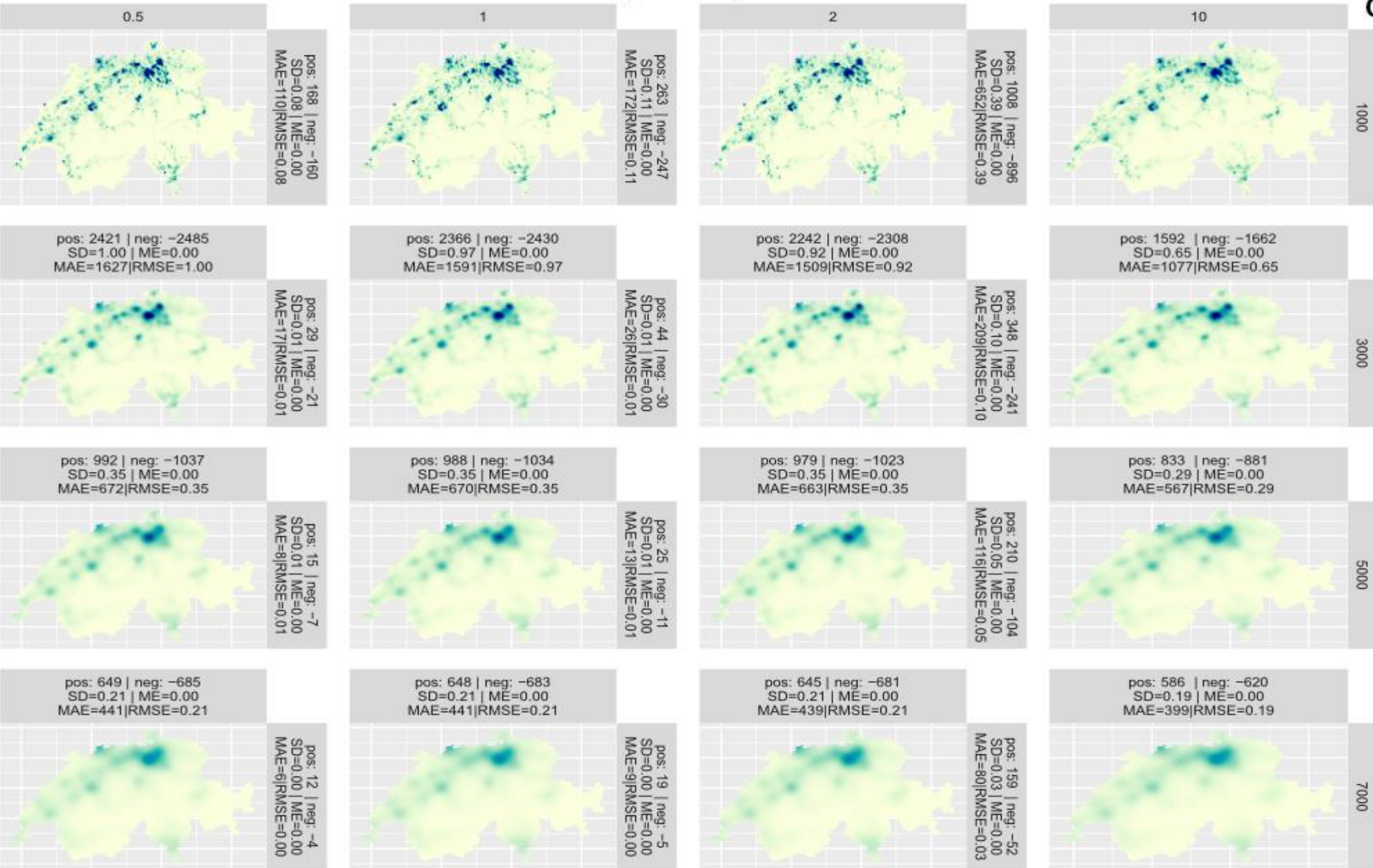
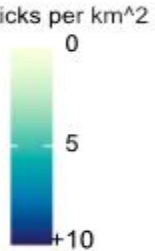


Figure 27: National matrix, concurrent KDE

The accuracy of each tick report varies. In the maps below these were multiplied by the factor of 0.5, 1, 2, and 10. The KDE-search radius also varies between 1000, 3000, 5000, 7000 meters. "pos" respectively "neg" show the sum of positive and negative changes between the maps. SD = Standard Deviation ME = Mean Error MAE = Mean Absolute Error RMSE = Root Mean Square Error

Accuracy of tick reports

Sequential



KDE-search radius

1000

3000

5000

7000

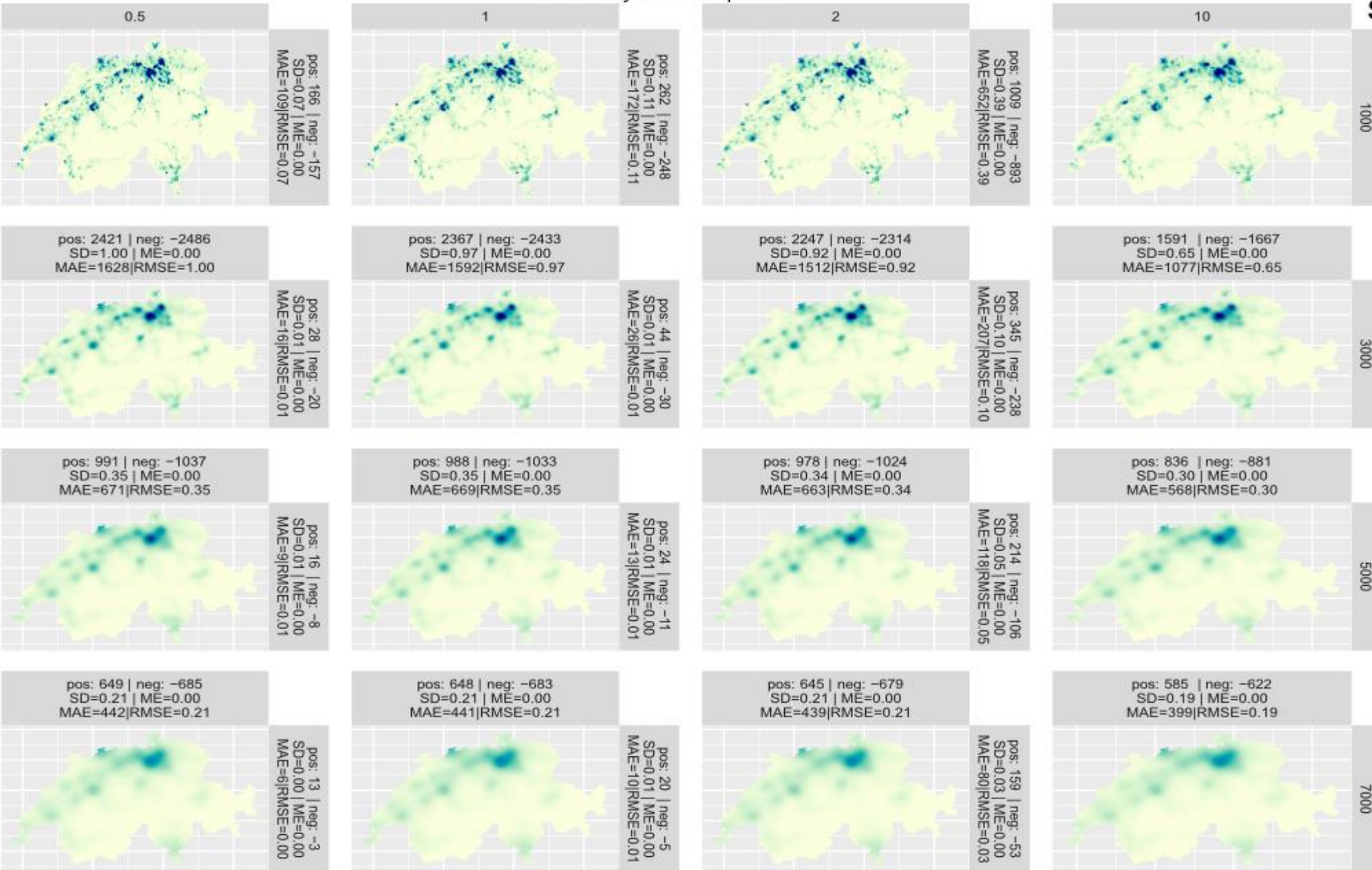


Figure 28: National matrix, sequential KDE

Weighted KDE

In contrast to the two MC methods, the *weighted* method remains its impact also on a national level. The way the formula works is that the weight of each report only decreases as the inaccuracy radii increases. Therefore, only positive values can be seen in each row. As the KDE radius rises, the clusters fade, and contours lose focus. The MAE of Switzerland is much higher compared to the *weighted* method of the local scale as the scale is bigger and more values are considered. As already explained on the local scale, horizontally no sums of negative values exist, because the map to the right only has smaller values, because the values get divided by the factor of 1, 2, or 10 respectively. Where there is almost no change between [K1000,U10]-[K3000,U10] and [K5000,U10]-[K7000,U10], there is a much greater change between [K1000,U0.5]-[K3000,U0.5] and [K5000,U0.5]-[K7000,U0.5]. (see [Figure 29](#)).

The accuracy of each tick report varies. In the maps below these were multiplied by the factor of 0.5, 1, 2, and 10. The KDE-search radius also varies between 1000, 3000, 5000, 7000 meters.

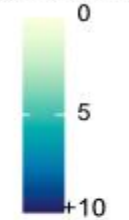
"pos" respectively "neg" show the sum of positive and negative changes between the maps.

SD = Standard Deviation ME = Mean Error MAE = Mean Absolute Error RMSE = Root Mean Square Error

Accuracy of tick reports

Weighted

ticks per km²



KDE-search radius

1000

3000

5000

7000

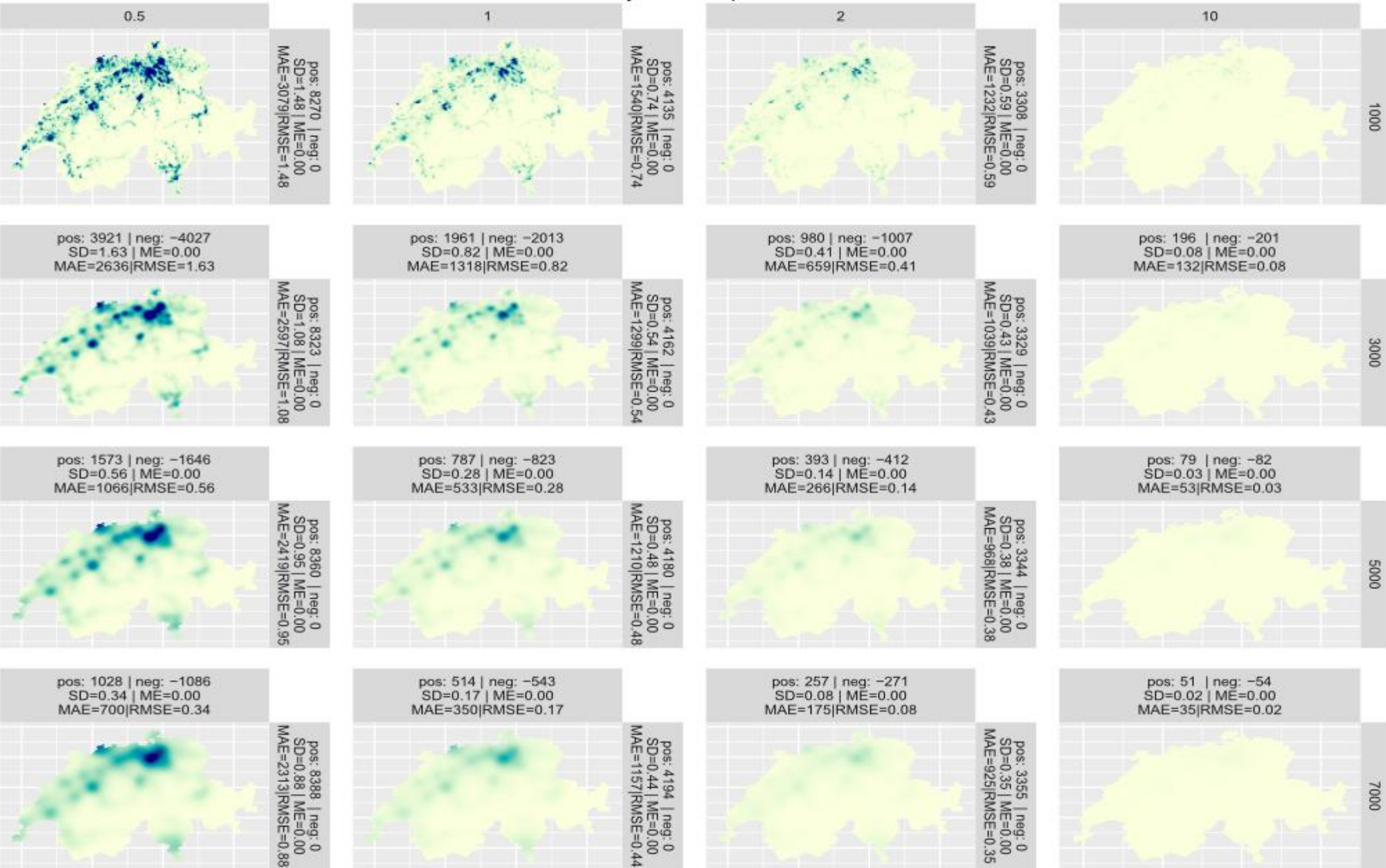


Figure 29: National matrix, weighted KDE

6.3 Differential analysis

6.3.1 Local scale (Sursee region, LU)

The maps in this chapter have different colour scales in order to show more information in each map. This way the comparison is more difficult to make between the methods, but the visual aids are much better.

Whenever a figure shows the difference between two methods ([sequential & weighted] or [sequential & normal]), then the sequential map is subtracted from the other map ([weighted-sequential] or [normal-sequential]). The red rectangle indicates which (differential) map is being described.

In the middle of [Figure 30](#) the weighted KDE method of the local scale is shown. As indicated the top map shows the difference between 100m and 250m kernel each with no inaccuracy modifier. Here it is visible that as the kernel gets bigger the density of clusters spreads outwards, as to where the clusters of the smaller kernel stay in a small area but with a higher density.

The bottom map shows only positive numbers, which means the map with a 250m kernel and no inaccuracy modifier has an overall higher density than the map with also a 250 kernel but a two times inaccuracy modifier. The clusters remain the same.

The SD and RMSE are much higher in the top map than in the bottom one. This is because the difference between the 100m kernel and the 250m kernel is bigger than the step from a 1x to a 2x modifier in inaccuracy.

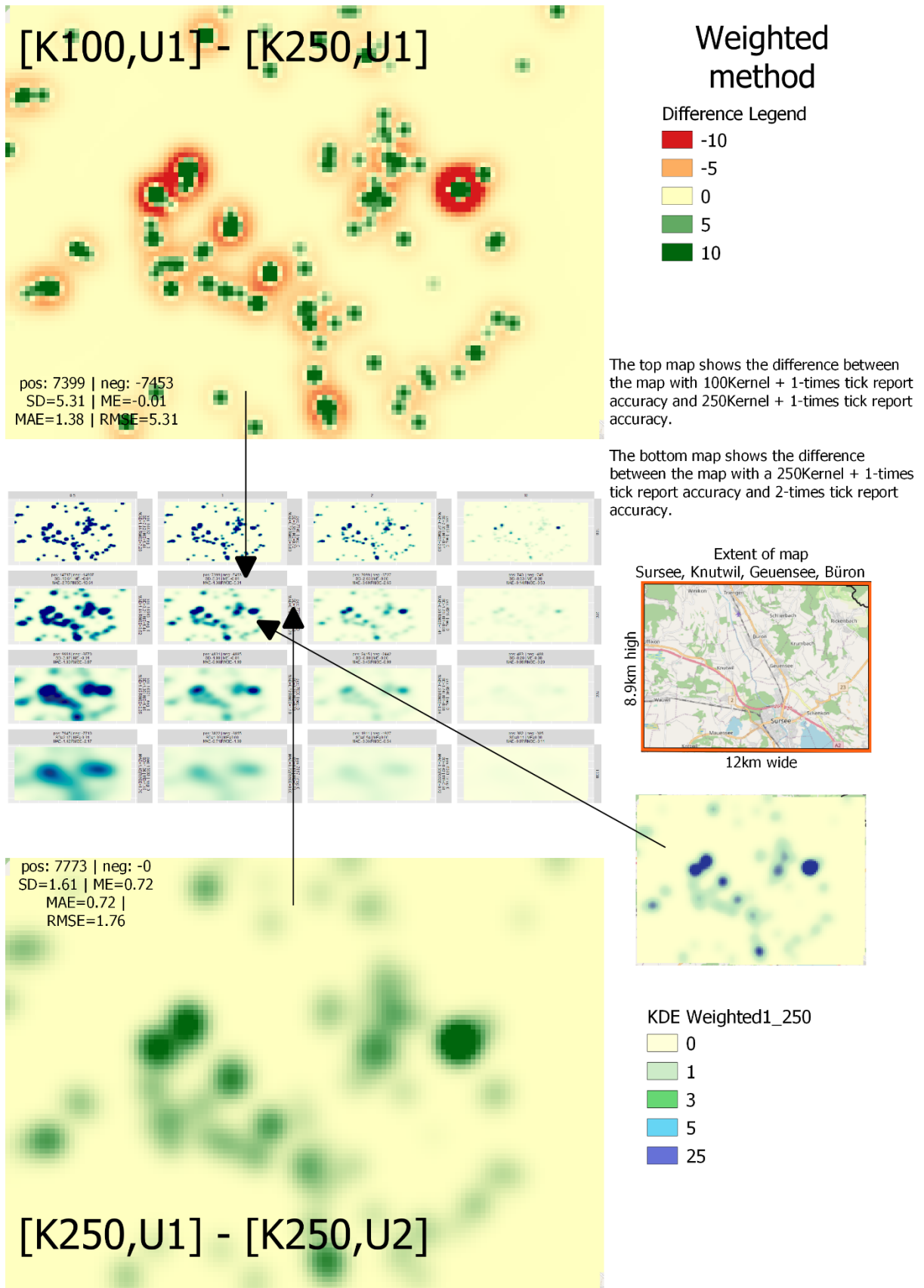
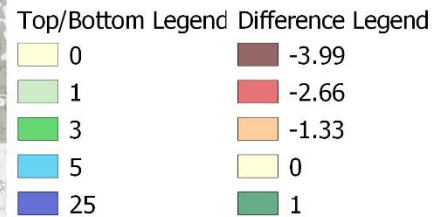
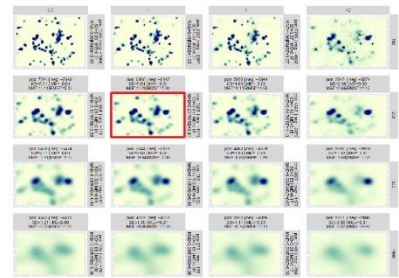
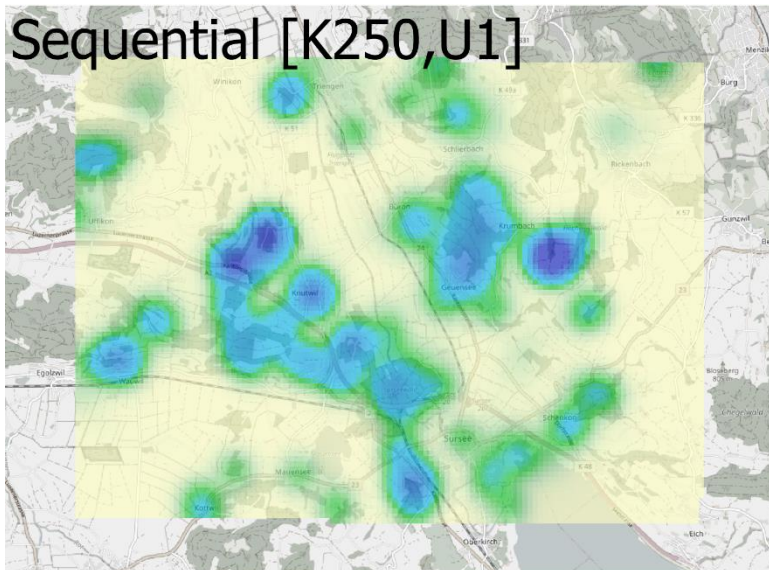


Figure 30: Local difference within *weighted*

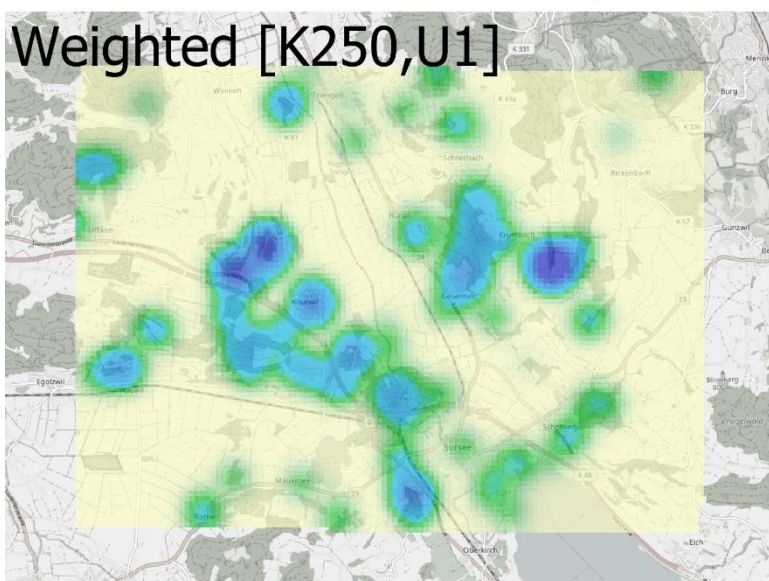
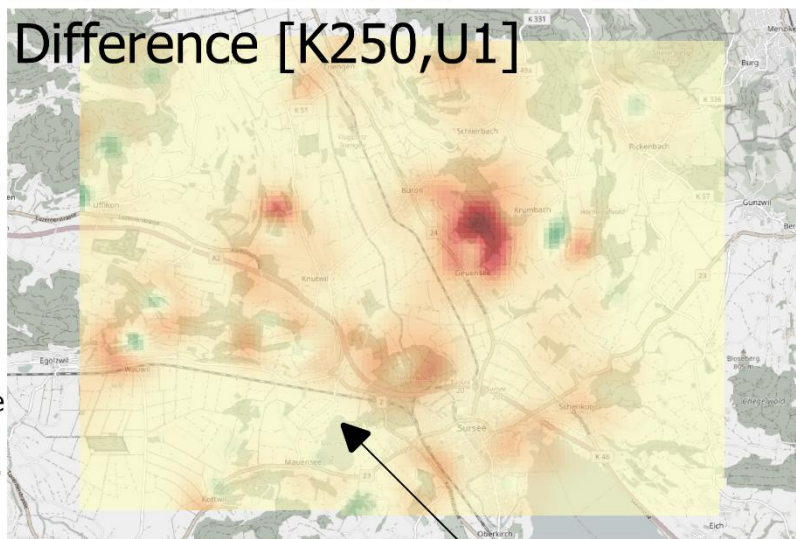
Here in [Figure 31](#) the difference between the methods with the same kernel and inaccuracy modifier is being investigated. The *weighted* map was subtracted by the *sequential*. The *sequential* method calculates a higher or the same density of tick reports almost everywhere. This is visible on the differential map as well as reflected in the measurements. The big difference on the right is located north of Geuensee in the local recreational area. A secluded view of the differential map can be seen in [Figure 32](#). There, it is visible that the area with the biggest difference has a high variety of inaccuracies and a lot of overlap. Since the *sequential* method takes the spatial features of the tick data into account it calculates a higher density than the *weighted* method. Only where there are single tick reports without any overlap with the *weighted* method calculates a higher density than the *sequential* one. The *sequential* method can simulate tick reports within the inaccuracy radii, therefore the density in the overlapping area increases the more overlaps exist in an area.



The top and bottom map each shows the KDE with a 250m search radius. The inaccuracy of the tick reports was not changed. Yet the top map shows the sequential method and the bottom one the weighted method.

The map in the middle visualizes the difference between the top and the bottom map.

The left legend counts towards the top and bottom map, whereas the right legend is for the middle map.

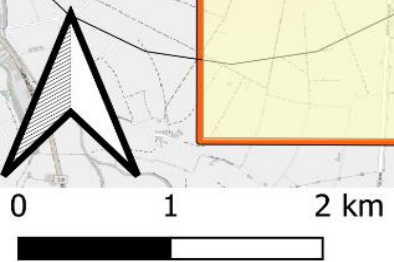
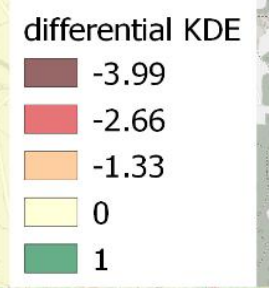
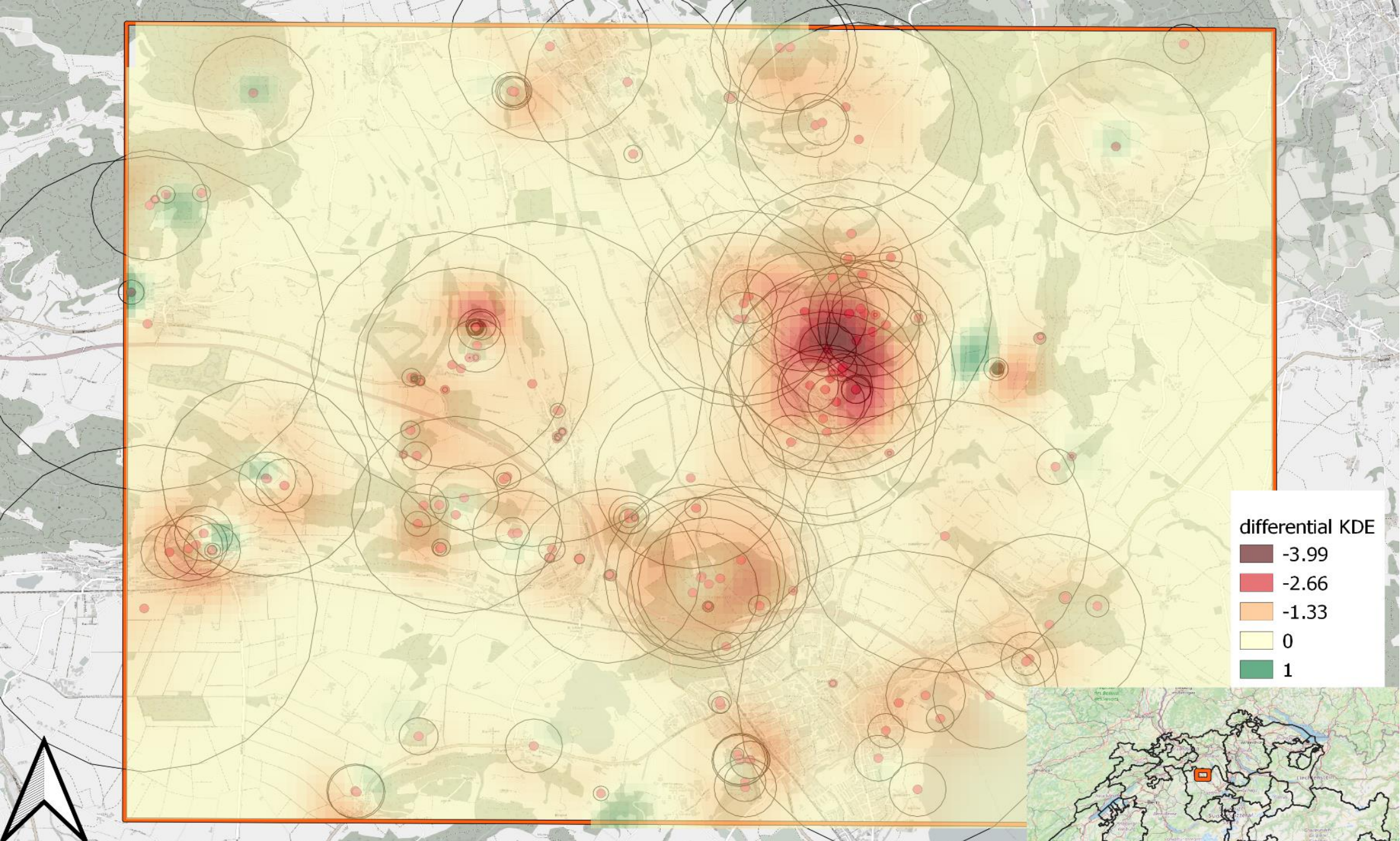


pos: 41 | neg: -3285
 SD=0.43 | ME=-0.30
 MAE=0.31 | RMSE=0.53

pos/neg show each the sum of positive and negative values respectively.

SD= Standard Deviation
 ME= Mean Error
 MAE= Mean Absolute Error
 RMSE= Root Mean Square Error

Figure 31: Local difference, *weighted vs sequential*



Extent: 12km wide / 8.9km high
Background source: OpenStreetMap

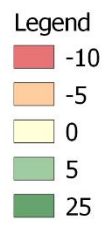
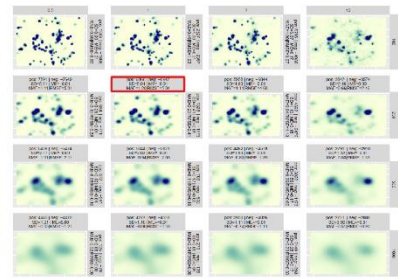
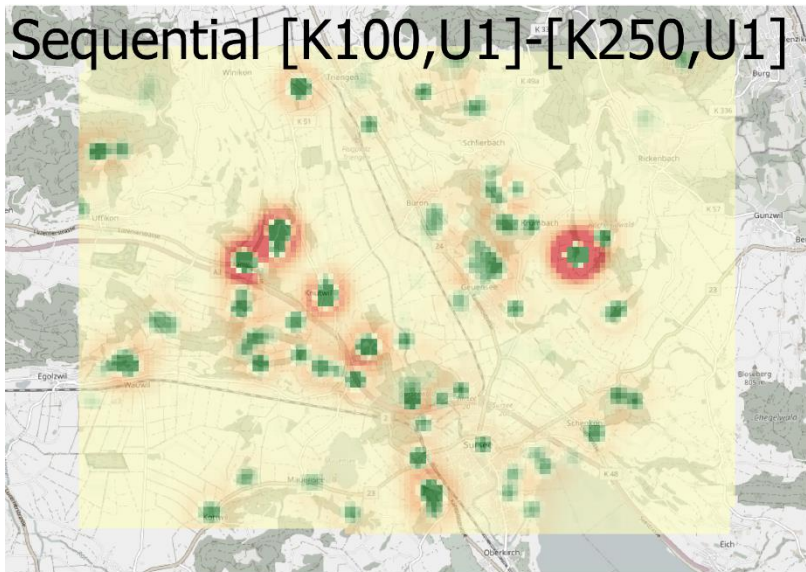
The subset is situated in the north of the Canton Luzern.

Closer inspection of the differential map [K250, U1] (weighted method - sequential method). The dots are the tick reports. The circles represent the inaccuracy radius of each tick report.

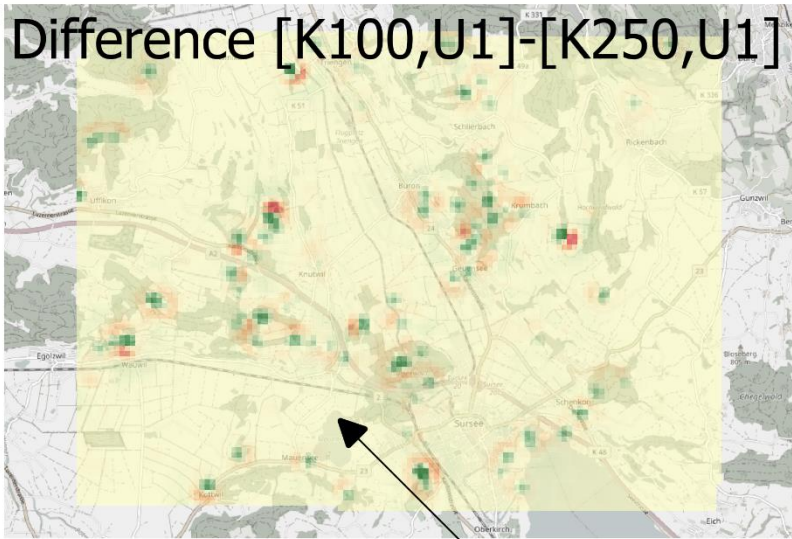


Figure 32: Difference weighted-sequential [K250, U1]

The next [Figure 33](#) displays the same methods as above, but this time the differential maps between the 100m kernel and the 250m kernel are shown. No changes were made to the inaccuracy of the tick reports. If the values are negative, then the *sequential* has a higher value, if it's green then the *weighted* method has a higher value. The *sequential* method spreads the cluster outwards, whereas the *weighted* has smaller clusters but with a higher density. Some bigger clusters have shifted sideways a bit. This could be because of the Monte Carlo Simulation used in the *sequential* method, which creates other instances of points within the inaccuracy radii of tick reports. With a RMSE of 1.08 it is rather high.

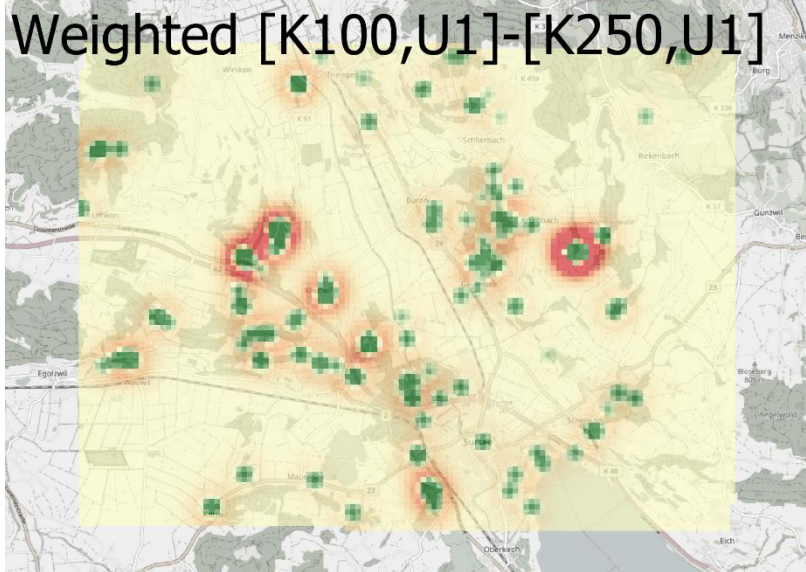


The top and the bottom map show the difference between the maps with 100m Kernel search radius and the 250m Kernel search radius. The inaccuracy of the tick reports was not changed. Yet the top map shows the sequential method and the bottom one the weighted method.



The map in the middle visualizes the difference between the top and bottom map.

The colour-code is the same for all three maps



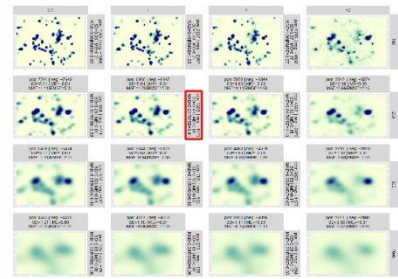
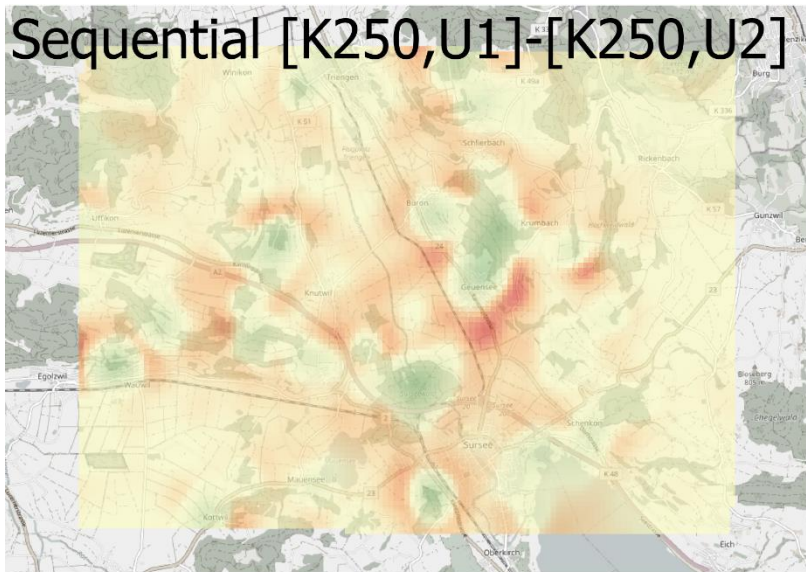
pos: 2241 | neg: -2217
 SD=1.08 | ME=0.00
 MAE=0.41 | RMSE=1.08

pos/neg show each the sum of positive and negative values respectively.

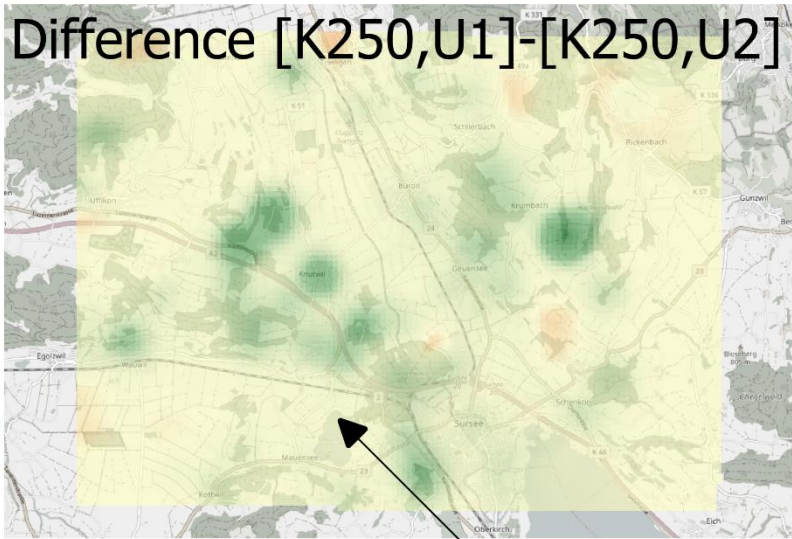
SD= Standard Deviation
 ME= Mean Error
 MAE= Mean Absolute Error
 RMSE= Root Mean Square Error

Figure 33: Local difference, *weighted vs sequential*, column

The next [Figure 34](#) displays the same methods as above, but this time the differential maps between the 1x and 2x inaccuracy of tick reports are shown. The kernel is 250m for both maps. The middle map shows the differential map between the top and bottom map. Since the differential map has mostly positive numbers, this leads to the conclusion that the *weighted* method creates a bigger difference between the 1x and 2x inaccuracy manipulation. With a SD of 1.45 and a RMSE of 1.61 the gap is considerably large.

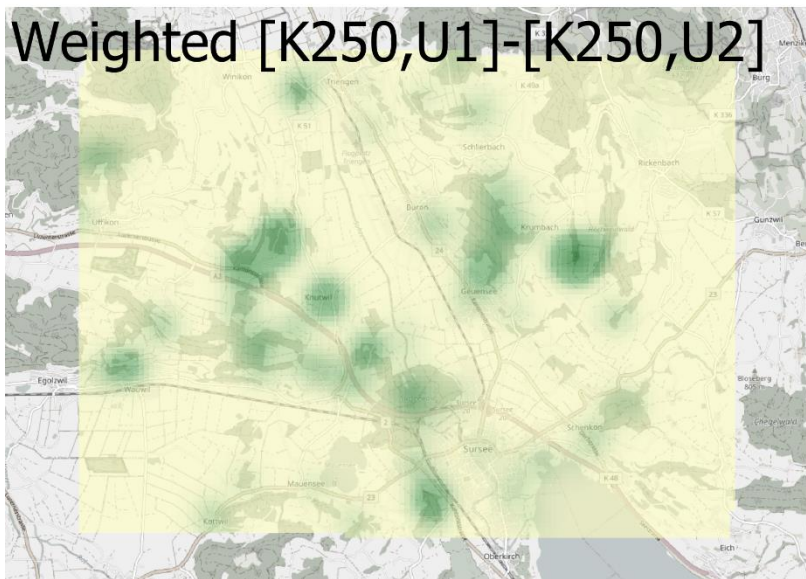


The top and the bottom map show the difference between the maps with a 1x and 2x inaccuracy modifier. The Kernel search radius is both times 250m. Yet the top map shows the sequential method and the bottom one the weighted method.



The map in the middle visualizes the difference between the top and the bottom map.

The colour-code is the same for all three maps.



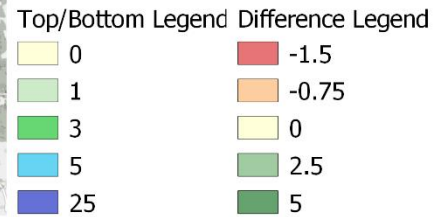
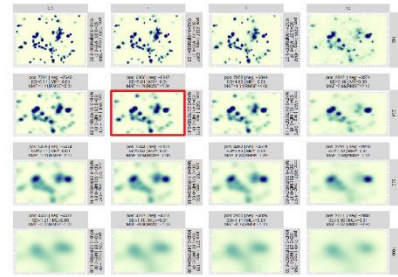
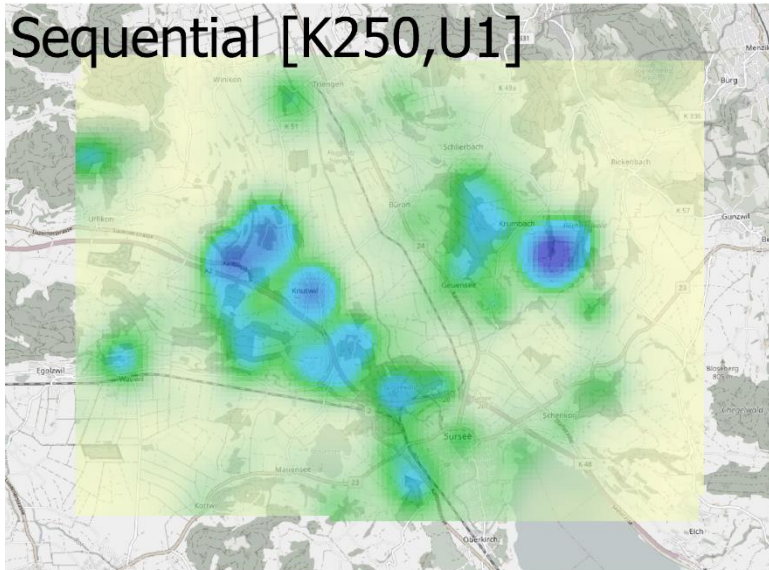
pos: 7681 | neg: -61
 SD=1.45 | ME=0.71
 MAE=0.72 | RMSE=1.61

pos/neg show each the sum of positive and negative values respectively.

SD= Standard Deviation
 ME= Mean Error
 MAE= Mean Absolute Error
 RMSE= Root Mean Square Error

Figure 34: Local difference, *weighted vs sequential*, row

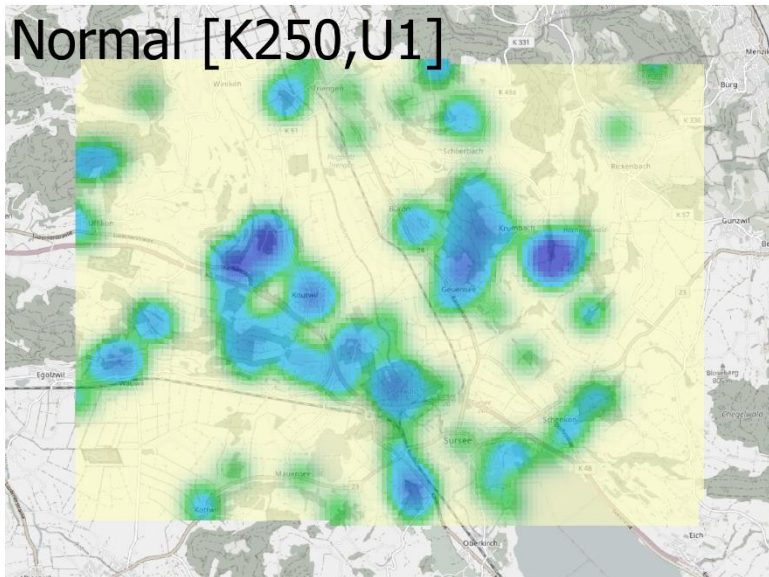
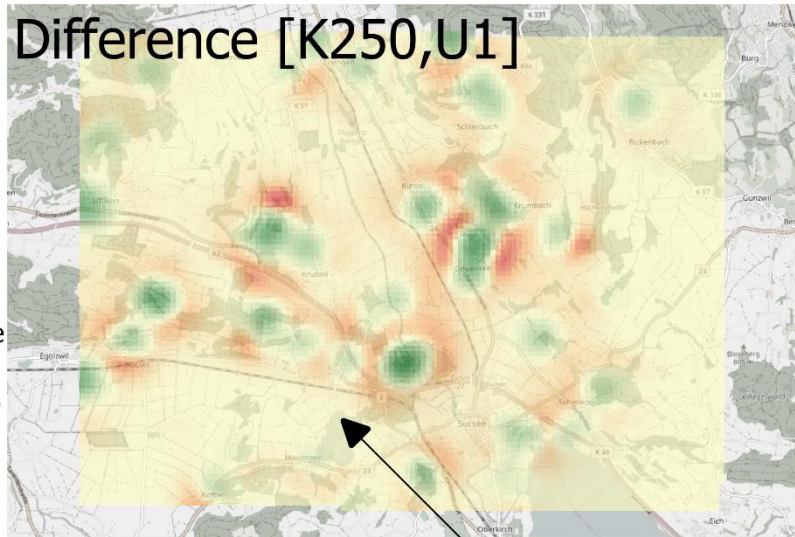
In Figure 35 the *sequential* method with a 250m kernel was subtracted from the *normal* method with also 250m kernel. The *normal* method calculates smaller and denser clusters, because of the MCS the sequential creates more dots inside their respective inaccuracy radii. This causes the *sequential* method to spread out the density, but keep it contained within an area.



The top and bottom map each shows the KDE with a 250m search radius. The top map shows the sequential method and the bottom one the normal method.

The map in the middle visualizes the difference between the top and the bottom map.

The left legend counts towards the top and bottom map, whereas the right legend is for the middle map.



pos: 1530 | neg: -1345
 SD=0.53 | ME=0.02
 MAE=0.27 | RMSE=0.53

pos/neg show each the sum of positive and negative values respectively.

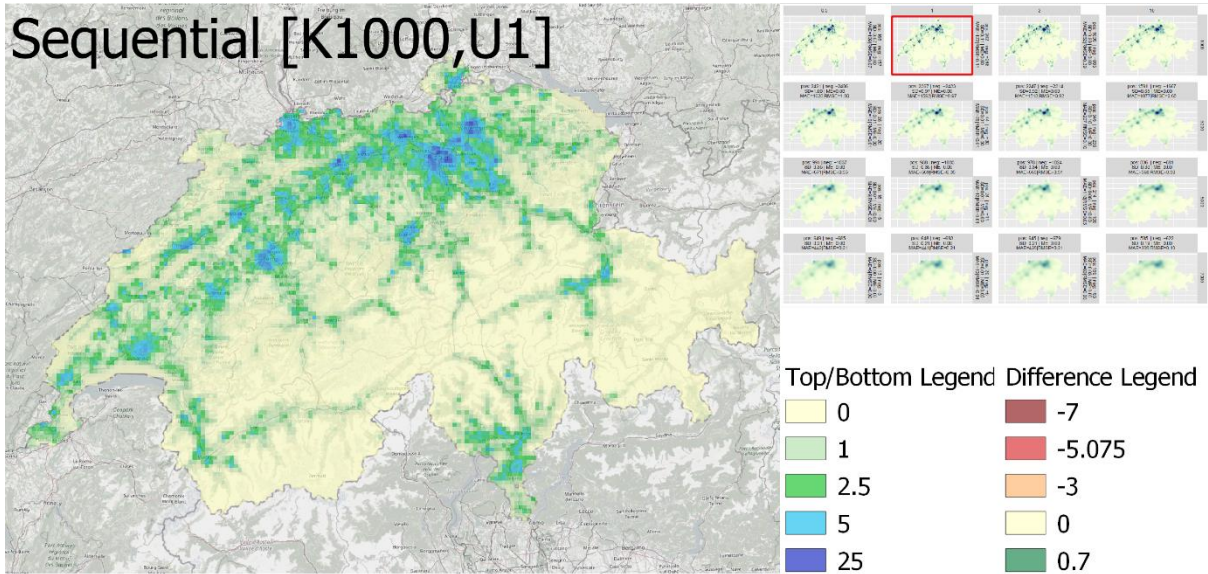
SD= Standard Deviation
 ME= Mean Error
 MAE= Mean Absolute Error
 RMSE= Root Mean Square Error

Figure 35: Local difference, normal vs sequential

6.3.2 National Scale (Switzerland)

The red square in the top right corner indicates which (differential) map is being described.

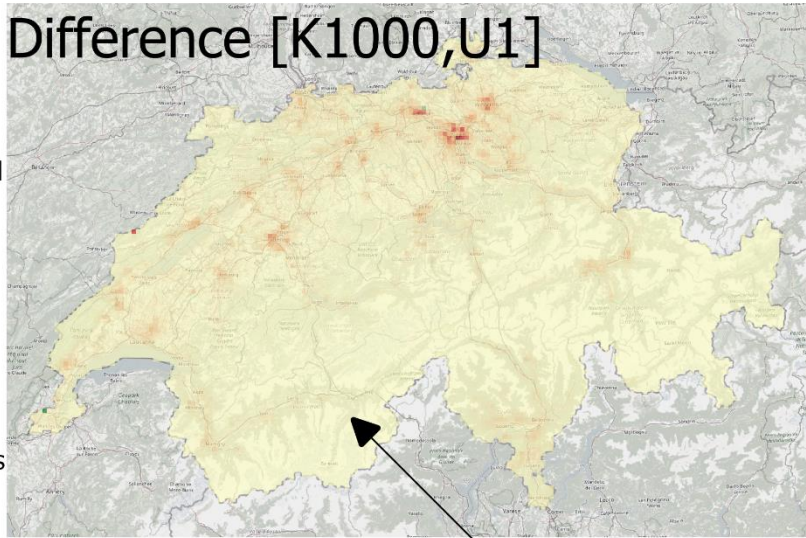
The overall difference between the two methods is that the *sequential* method calculates a higher density of tick reports than the *weighted* method. The variance might be rather small yet is still there even on a national scale (see [Figure 36](#)).



The top and the bottom map show the density of tick reports with a Kernel search radius of 1000. The inaccuracy of the tick reports were not changed. Yet the top map shows the sequential method and the bottom one the weighted method.

The map in the middle visualizes the difference between the top and the bottom map.

The left legend counts towards the top and bottom map, whereas the right legend is for the middle map.



pos: 3 | neg: -2262
 SD=0.38 | ME=-0.21
 MAE=0.21 | RMSE=0.44

pos/neg show each the sum of positive and negative values respectively.

SD= Standard Deviation
 ME= Mean Error
 MAE= Mean Absolute Error
 RMSE= Root Mean Square Error

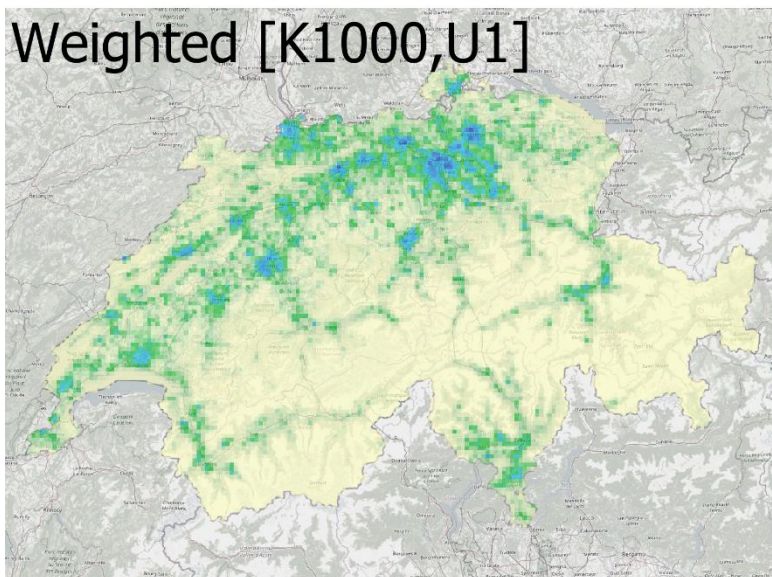


Figure 36: National difference, *weighted vs sequential*

Judging by the differential measurements, the *weighted* map changes the density from the 1'000m to the 3'000m kernel slightly more than the *sequential*. No clear trends of the way the cells change values are visible (see [Figure 37](#)).

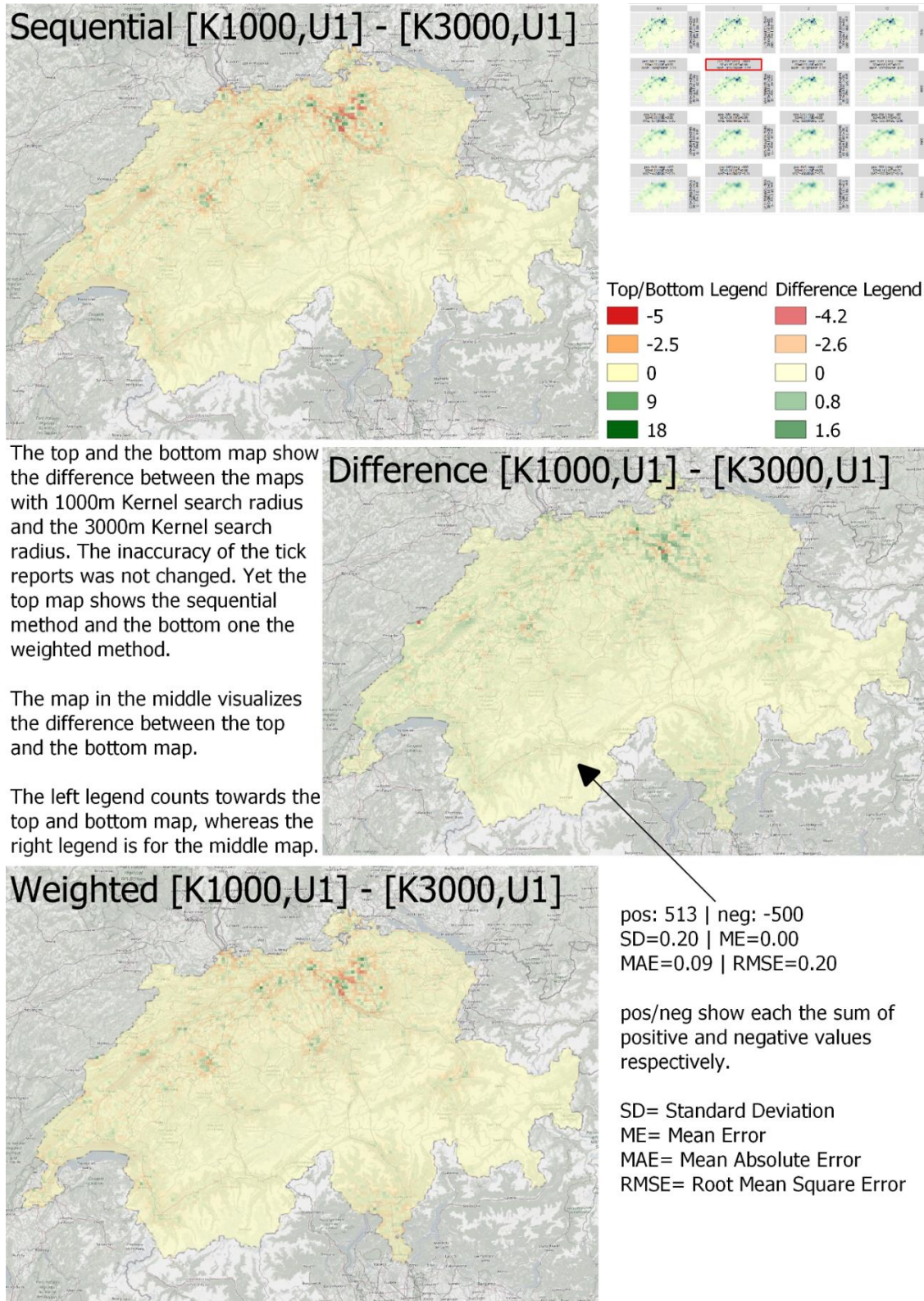


Figure 37: National difference, *weighted* vs *sequential*, column

The difference between the changes in inaccuracy in [Figure 38](#) shows that the *sequential* method changes much less by increasing the inaccuracy factor than the *weighted* method. With a SD of 0.71 and a RMSE of 0.81 on a national scale, these differences are rather large.

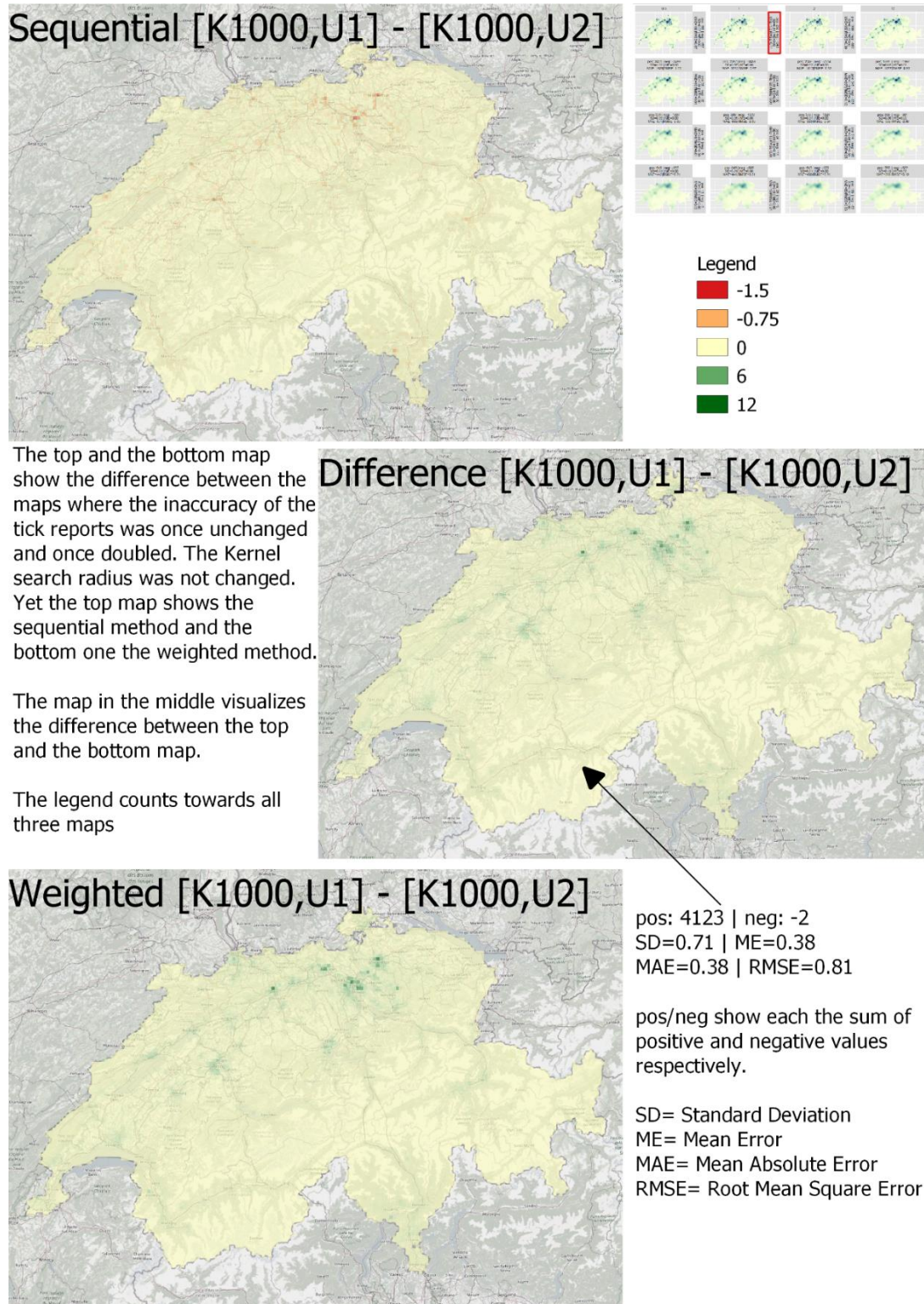


Figure 38: National difference, *weighted* vs *sequential*, row

Between the *normal* and the *sequential* methods is little change on a national level. The ME is 0.00 and the MAE is 0.05. No noticeable difference is being recorded except for 3 out of 10782 cell, where the difference is extremely high (see [Figure 39](#)).

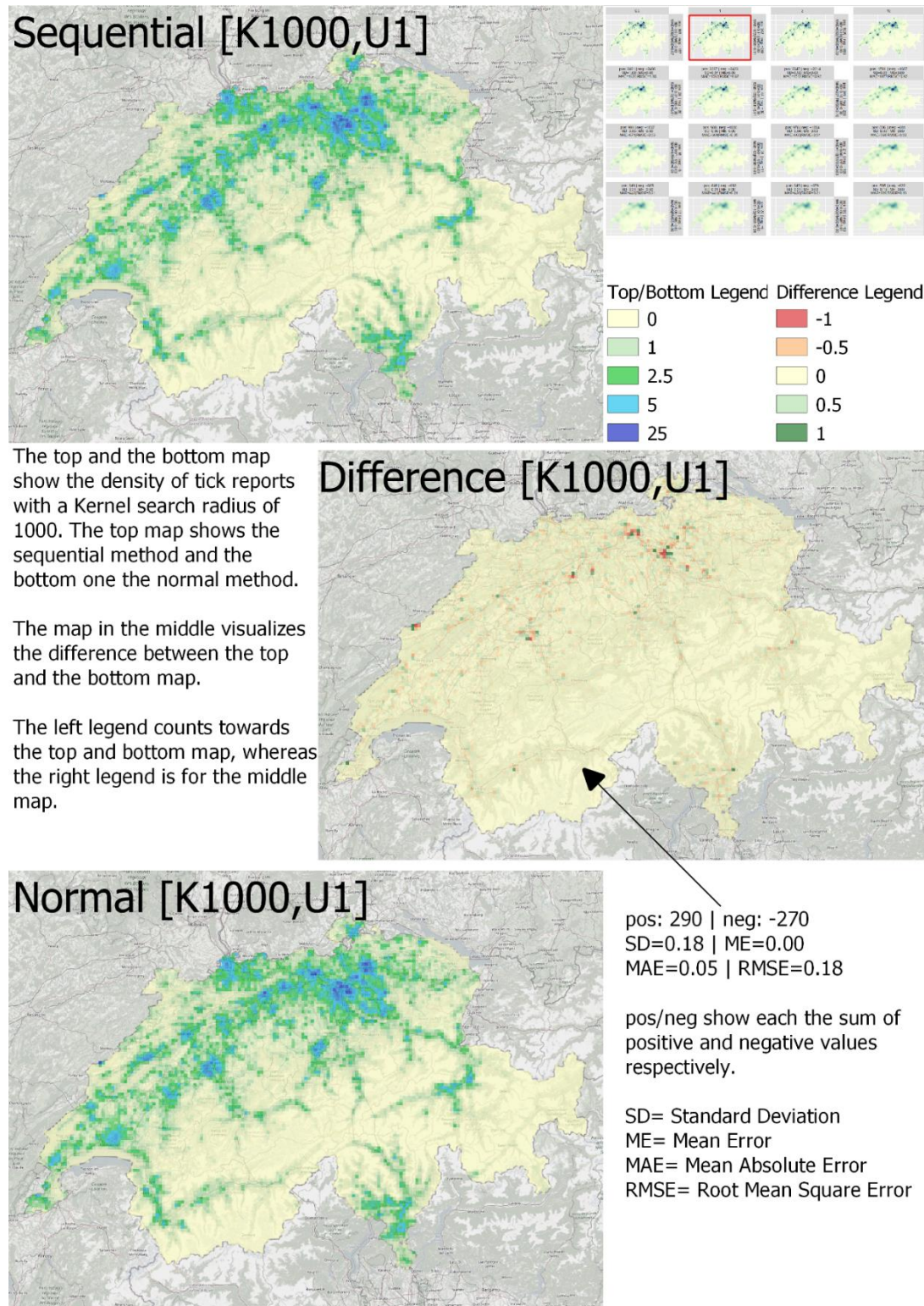


Figure 39: National difference, *normal* vs *sequential*

7. Discussion

In the discussion part all research questions will be discussed in order.

7.1. What spatio-temporal patterns are visible from crowdsourced tick data?

There has been a general trend of increasing tick bites over the years, though this is due to the app getting more popular. In the year 2020 over 13'000 tick reports were made until the 10th of August. The actual reasons for this extreme increase are unknown, but speculations can be made about the impact of COVID-19 in 2020. Because of the lockdown that happened from spring until mid-summer, almost everyone stayed at home. People were still allowed to go outside for walks in their area (Bundesamt für Gesundheit BAG, 2020b). This could have led to a major increase in visitors in the local recreational areas and their gardens. These are areas that are favoured by ticks (Hansford et al., 2017; Mulder et al., 2013; Paul, Cote, Le Naour, & Bonnet, 2016).

As all years show the same pattern in the months, May and June are the peak months where over 60% of all tick bites happen. As it gets hotter, the dry periods get longer. Ticks are sensitive to dry condition and can die if no rain falls for a long time. Over the months the temperatures drop, but the rain periods are not increasing significantly (MeteoSchweiz, 2019; Perret et al., 2000). The lower temperatures can cause ticks to die or hide. Both reasons reinforce the drop of tick reports over the following months.

As ticks can survive several years without sucking blood, it appears that this is the reason why even some tick reports happen during winter, when their habitat is usually wet.

Ticks prefer humid conditions and can die quite quickly in arid weather (Perret et al., 2000). May and June are usually the wettest months of the year (MeteoSchweiz, 2019). This explains the high number of tick reports from the 21st to the 27th week of the year.

The working weekdays show no big fluctuation, but there is an increase on Saturday and Sunday. Many people don't have to work during the weekend, which is when they have time to go to the green places. Some tick bites are discovered on the same day, but also a large portion get only discovered the next day as they are hard to spot. This is interesting to see but is not of importance to this project.

As seen in [Figure 14](#), ticks can be found all over Switzerland with varying occurrences. The maps roughly follow the same patterns as to where humans live. This is logical, because tick reports can only happen where people are, and people are mostly in habitable areas. The canton of Zurich has the most tick reports of the Swiss cantons, which is reflected with the highest human population.

Looking at the tick bites per population, 4 out of the top 5 cantons are in the southern half of Switzerland (see [Figure 17](#)). A rough pattern could be seen, that more tick bites happen in predominantly rural cantons, such as Grisons and Valais, than in urban cantons, such as Basel (Bundesamt für Statistik (BFS), 2014). Although this is only true for the cantons at the very top and bottom of [Figure 17](#). As the data shows, ticks can be found almost everywhere in Switzerland, including Ticino and Geneva. In both cantons FMSE cases are confirmed yet no vaccination against FMSE is recommended. Especially in Ticino, which scores three times in the top 5 ([Figure 15](#), [Figure 16](#), and [Figure 17](#)), a recommended vaccination is strongly advised and should be made public by the state. It isn't surprising to see most tick reports are found in the midland, which is where most of the people live and local recreational areas reside.

The reasons are unknown why so many accuracy radii are around the standard report values. Reasons could be that people were not sure that they had the ability to change the accuracy of their reports and accidentally changed it by changing the location, or they were already happy with the predefined settings and did not feel the need to change the radius a lot. This has a negative impact on the quality of the crowdsourced data. No measurement tool is implemented in the app "*Zecke*" to see how big the chosen radius is in numbers.

7.2. How to estimate tick density from user-generated tick bite data with positional uncertainty?

The easiest way to handle low quality data is to prevent it from happening at the source. Sometimes collecting uncertainties is necessary for the project. It can give you great inside information about the userbase. With the crowdsourced tick reports, the uncertainty of tick bite location is necessary, because people do not realize when they are bitten by a tick.

The first thing to do is to minimize the low-quality uncertainty from the necessary uncertainty. For this all tick reports were filtered out with an inaccuracy of more than 2'130 meters. The smaller the radius of inaccuracy, the higher the quality of the tick report. Later the standard report inaccuracies were taken out. Which then left me with 40'579 out of the starting 47'723 tick reports.

After cleaning the data, 3 methods were developed: the *weighted KDE*, *concurrent KDE*, and *sequential KDE*. By using the KDE, the density could be calculated, and the kernel and other factors could be changed, giving the option to change the kernel search radius among other things.

To have a comparison on what an impact the inaccuracy has on the density it was set in contrast to the kernel parameter. By increasing and decreasing the existing inaccuracies with the inaccuracy modifier, the simulation of lower or higher quality data could be made. A combination of the kernel parameters and the inaccuracy modifier created small multiples. These allowed for visual comparison as well as comparing the differences between other maps of the same and of another method. Statistical measurements were used for quantification of the divergences of two maps.

With the small multiples and the statistical measurements, the comparison could be made between the methods with the same extent, inaccuracy modifier, and kernel parameter.

All three methods have (dis-)advantages under circumstances. The *weighted* method has the advantage of always having an influence on the outcome, no matter the

inaccuracy radii or the scale. It calculates the overall lowest density of ticks out of the four methods. Another advantage is that any other input could be used as weights, therefore it is more flexible than the other methods, but unlike the *sequential* or *concurrent* methods it does not represent the spatial features of the inaccuracy on a spatial level but as numbers.

The *sequential* or *concurrent* methods share the same pros and cons. On a local level they show the inaccuracy of the tick reports by simulating more tick reports within their radii. They calculate the threat of tick bites in a wider area than the two methods and show that tick bites do not necessarily only happen in their specific coordinates. A big disadvantage is the lack of impact on large scale (e.g. national scale). There the additional simulations within inaccuracies are too small to create a significant difference between the *sequential / concurrent* and the *normal* method.

When it comes to the *normal* method, it calculates the highest tick density out the four methods. It was the fastest and easiest to calculate but does not consider any spatial features apart from the tick bite locations themselves.

7.3 What is the relation between the created spatial patterns of the different methods, the kernel parameter, and the inaccuracy radii of the trick reports?

The discussion of this question starts with the global relations within the different methods. Examples will be given on a local scale, but on a national scale most observations of the different methods and their relations stay true but are less prominent. Then the differential maps between the methods are inspected on a local and national scale.

With each method, the clusters behave differently when modifying the inaccuracy or the kernel bandwidth. In the case of the *normal* KDE, the inaccuracy has no impact at all, therefore only the kernel parameter can be changed. On a local scale with a search radius of 100m clear dots are visible. As the kernel increases the dots begin to merge in to bigger clusters and the density is spread out in a wider area, until it almost vanishes with a kernel of 1'000m (see [Figure 22](#)).

The *concurrent* and *sequential* methods are practically the same and show the same patterns and relation to a change in inaccuracy and kernel parameter. With a kernel of 100m an increase in inaccuracy scatters the existing small clusters in the near area. With a ten times bigger inaccuracy small clusters are so spread out that they dissipate. Bigger clusters remain, and some might have merged a little with neighbouring clusters. With higher kernel bandwidths, the change in inaccuracy decreases until almost no change is visible with a kernel of 1'000m. A 250m kernel still shows the scattering of the tick reports and some clusters are less defined (see [Figure 23](#) and [Figure 24](#)).

The local *weighted* kernel is interesting. With a decrease of inaccuracy to a 0.5 modifier and a 100m kernel, the dots become bigger and more prominent, but almost vanish with an increase of inaccuracy. This trend becomes truer the bigger the kernel gets. With an increase in the bandwidth of the kernel, the clusters become less dense but bigger (see [Figure 25](#) and [Figure 30](#)).

In every case the kernel factor had a bigger impact on the resulting map than the inaccuracies. The changes made from 1x to 2x increase of inaccuracy radii are not

very striking. Changes were evident only after the radii were increased by a factor of 10.

The difference between the *weighted* and the *sequential* map in [Figure 31](#) shows that the *sequential* method calculates almost everywhere a higher density than the *weighted* method, except for a few separated tick reports with no neighbours. If a lot of tick reports with small and large inaccuracy radii are present, like the big red area seen in [Figure 31](#), then the *sequential* method calculates a much larger cluster with a higher density than the *weighted* method. This is not the case for places with many tick reports near each other and of the same size like in the middle and very south of the map (see [Figure 32](#)). There a similar density was calculated. Only where the big radius to the right of it comes into play does the density slightly change between the methods.

When comparing the influence of the kernel parameter between the methods *weighted* and *sequential*, the change isn't as dramatic as to comparing the calculated KDE in the paragraph above (see [Figure 33](#)). The *weighted* method calculates a higher density in the centre of the clusters whereas the *sequential* method has the centre of its clusters slightly shifted due to the MCS. The overall density of the rasters stays the same seen by the ME of 0.00.

With changing the inaccuracy modifier from 1 to 2, the *weighted* method changes more than the *sequential* one, because the *weighted* method divides all values by 2 after the normalization, whereas the increase of inaccuracy spreads out the density, but doesn't reduce it (see [Figure 34](#)).

As for the difference between the *normal* and *sequential* method, the *normal* KDE calculates a higher density with smaller, more defined clusters. The *sequential* method increases the area of the clusters, but also lowers the density of these clusters, but the overall density of both rasters is similar, seen by the ME of 0.02 (see [Figure 35](#)).

On a national scale the *sequential* method calculates a higher density than the *weighted* method (see [Figure 36](#)). The same patterns are visible on the national scale as on the local scale but to a lesser degree.

The influence of the kernel search radius of the differential maps decreased dramatically from the local to the national scale. The overall difference might still be the same (ME=0.00), but the RMSE went from a 1.08 to a 0.2 and the MAE from a 0.41 to a 0.09 (see [Figure 33](#) and [Figure 37](#)).

The impact of the 2-times inaccuracy modifier changed from the local to the national level by also dividing almost every statistical measurement by two. The SD dropped from 1.45 to 0.71, the ME from 0.71 to 0.38, the MAE from 0.72 to 0.38, and the RMSE went from a 1.61 to a 0.81. This shows that the increase of the inaccuracy has a lower significance on a national scale than on a locale scale when comparing the *weighted* with the *sequential* method (see [Figure 34](#) and [Figure 38](#)).

The same trend holds true for the comparison of the *normal* and the *sequential* method (see [Figure 39](#)). Here the difference between the local and national scale is bigger. With a RMSE of 0.18 on a national level where it was 0.53 on a local level, the difference between the methods is miniscule (see [Figure 35](#) and [Figure 39](#)).

8. Conclusion

This master thesis is the first Swiss-wide thorough analysis of crowdsourced tick data. Before the analysis could begin, the data was cleaned of standard reports and ones with an accuracy radius of over 2'130m. Then, the temporal attributes of the tick data were explored. A huge upsurge of data happened in the year 2020 compared to the years before. This could have been because of the COVID-19 virus that emerged the same year. Over the years, most tick reports happened during the week 21 and 27 which lied in May and June. Most tick reports were sent during the weekend, with the presumption that more people go to green places during the weekend than weekdays. For spatial exploration, it was found that ticks can be found almost everywhere in Switzerland. Many ticks were reported from the cantons Zurich, Bern, Vaud, and Aargau. When normalized with the cantons' population, then the top 4 are Obwald, Grisons, and Solothurn; closely followed by Ticino. The height distribution is centred around 466 and 697meters above sea level. The accuracy radii of tick reports show three spikes around 100, 300, and 1'000meters. The standard reports had similar numbers, which could lead to believe if the standard reports were changed, so would also the radii.

Afterwards the small multiples of the different methods were analysed by comparing the differences within the small multiples visually and through statistical measurements. Locally the inaccuracies of tick reports have a small impact, whereas the kernel parameter has a much bigger one. On a national scale the inaccuracy's impact was neglectable, but with the *weighted* method it could still be relevant. This method however did not take the spatial properties of the tick data correctly into account. The differential analysis showed that the *normal* KDE calculates the overall highest density, followed by the *concurrent* and *sequential* at the same position and the *weighted* method calculates the lowest density.

With the explorative data analysis, tick reports happen all over Switzerland, mostly in local recreational areas. The canton of Ticino has a lot of municipalities with very high tick reports per population numbers in Switzerland. The FSME map ([Figure 2](#)) shows that Ticino has confirmed FMSE cases. Comparing this with the high number of tick reports in that canton, making Ticino also a risk area for FSME and

recommending a vaccination might be a good idea. The months May and June are especially dangerous for getting bitten by a tick. With the *weighted* method, all types of uncertainty can be weighted and visualized with the KDE. Yet the *concurrent*, and *sequential* methods are more suitable for spatial uncertainty, because they better show the spatial properties and can simulate the extent of the uncertainty and create a KDE afterwards.

The arbitrary number of kernel search radii and the inaccuracy modifier could be varied to yield additional results. To further know when many tick bites happen, the weather could be used as an indicator to predict future tick bites. By looking at past weather reports and the following tick count, relations could maybe be seen.

What would be interesting to see next is how many of the gathered tick cases were reported with FSME. Are there clusters as well? Since this data is very private and doctors are bound to professional discretion, it would be hard to come by.

With more tick reports being send, another analyses could be made with updated data, however it is unlikely that the outcome of the maps will change a lot if the same procedure will be done with more data, especially on a big scale like Switzerland. On the small scale it will have an impact, especially in places where there haven't been any or very few tick reports.

The tick data also contains temporal uncertainties (e.g. when did the tick bite happen? Today, yesterday, or before?), which could be investigated. How accurate is the temporal perception of people when they find a tick bite?

9. References

- A. S. G. ZHAW Spin-off. (n.d.). *Zecke*. Retrieved from <https://zecke-tique-tick.ch/en/>
- Amatulli, G., Pérez-Cabello, F., & de la Riva, J. (2007). Mapping lightning/human-caused wildfires occurrence under ignition point location uncertainty. *Ecological Modelling*, *200*(3–4), 321–333. <https://doi.org/10.1016/j.ecolmodel.2006.08.001>
- Bailey, T.C., & Gatrell, A. C. (1995). *Interactive spatial data analysis*. England: Harlow Essex, Longman Scientific & Technical.
- Bern, C.-. (2019). *BAG-Bulletin 06/2019*.
- Bern, C.-, & Bern, C.-. (2018). *BAG-Bulletin 41/2018*.
- Bonney, R., Cooper, C. B., Dickinson, J., Kelling, S., Phillips, T., Rosenberg, K. V., & Shirk, J. (2009). Citizen science: A developing tool for expanding science knowledge and scientific literacy. *BioScience*. <https://doi.org/10.1525/bio.2009.59.11.9>
- Bundesamt für Gesundheit BAG. (2020a). Frühsommer-Meningoenzephalitis (FSME). Retrieved from Bundesamt für Gesundheit BAG website: <https://www.bag.admin.ch/bag/de/home/krankheiten/krankheiten-im-ueberblick/fsme.html>
- Bundesamt für Gesundheit BAG. (2020b). Neues Coronavirus: Massnahmen und Verordnungen. Retrieved September 10, 2020, from <https://www.bag.admin.ch/bag/de/home/krankheiten/ausbrueche-epidemien-pandemien/aktuelle-ausbrueche-epidemien/novel-cov/massnahmen-des-bundes.html>
- Bundesamt für Gesundheit BAG. (2020c). *Swiss Covid*. Retrieved from <https://www.bag.admin.ch/bag/de/home/krankheiten/ausbrueche-epidemien-pandemien/aktuelle-ausbrueche-epidemien/novel-cov/swisscovid-app-und-contact-tracing.html>
- Bundesamt für Statistik (BFS). (2014). *Raum mit städtischem Charakter 2012*. Retrieved from <https://www.bfs.admin.ch/bfsstatic/dam/assets/349558/master>
- Bundesamt für Statistik, B. (2020). Regionalporträts 2020: Gemeinden - Kennzahlen.

- Chai, T., & Draxler, R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE)? -Arguments against avoiding RMSE in the literature. *Geoscientific Model Development*, 7(3), 1247–1250. <https://doi.org/10.5194/gmd-7-1247-2014>
- Craglia, M., Ostermann, F., & Spinsanti, L. (2012). Digital Earth from vision to practice: Making sense of citizen-generated content. *International Journal of Digital Earth*, 5(5), 398–416. <https://doi.org/10.1080/17538947.2012.712273>
- Eiben, C. B., Siegel, J. B., Bale, J. B., Cooper, S., Khatib, F., Shen, B. W., ... Baker, D. (2012). Increased Diels-Alderase activity through backbone remodeling guided by Foldit players. *Nature Biotechnology*, 30(2), 190–192. <https://doi.org/10.1038/nbt.2109>
- Garcia Martí, I. (2019). Mapping tick dynamics and tick bite risk using data-driven approaches and volunteered observations (University of Twente). <https://doi.org/10.3990/1.9789036548595>
- Gäumann, R. (2010). *Molecular epidemiology of tick-borne encephalitis viruses in Switzerland* (University of Bern). Retrieved from <https://files.zotero.net/eyJleHBpcmVzIjoxNTg3ODExOTI1LCJoYXNoIjoiZTc3NDc5ODZlZGUzODE5ZmVky2VhMTI0OTA4OTI1M2YiLCJjb250ZW50VHlwZSI6ImFwcGxpY2F0aW9uXC9wZGYiLCJjaGFyc2V0IjoiIiwizmlsZW5hbWUiOiJBY2tlcm1hbm4tR8OkdW1hbm4gLSBNb2x1Y3VsYXlIgZXBpZGVtaW9sb2d5IG9mIHRpY2>
- GEO-NET Umweltconsulting GmbH. (2019). *Stadtklimaanalyse* (Vol. 09). Retrieved from https://map.geo.bs.ch/file_proxy/KL_Stadtklima_Windstroemungsfeld/Endbericht_Basel_Klimaanalyse_Rev09_ohne_Anhang.pdf
- Goodchild, M. F., & Li, L. (2012). Assuring the quality of volunteered geographic information. *Spatial Statistics*. <https://doi.org/10.1016/j.spasta.2012.03.002>
- Haklay, M. (2015). Citizen Science and Policy : A European Perspective by Muki Haklay. *The Woodrow Wilson Center*.
- Hansford, K. M., Fonville, M., Gillingham, E. L., Coipan, E. C., Pietzsch, M. E., Krawczyk, A. I., ... Medlock, J. M. (2017). Ticks and *Borrelia* in urban and peri-urban green space habitats in a city in southern England. *Ticks and Tick-Borne Diseases*, 8(3). <https://doi.org/10.1016/j.ttbdis.2016.12.009>
- Kenney, J. F. (1939). *Mathematics of Statistics part I* (4th ed.). Retrieved from

<https://archive.org/details/MathematicsOfStatisticsPartI/page/n3/mode/2up?q=root-mean-square>

Kenney, J. F. (1940). *Mathematics of Statistics part II* (4th ed.). Retrieved from <https://archive.org/details/MathematicsOfStatisticsPartIi/page/n211/mode/2up?q=standard+deviation>

Khatib, F., Cooper, S., Tyka, M. D., Xu, K., Makedon, I., Popović, Z., ... Players, F. (2011). Algorithm discovery by protein folding game players. *Proceedings of the National Academy of Sciences*, *108*(47), 18949–18953. <https://doi.org/10.1073/PNAS.1115898108>

Koutsias, N., Kalabokidis, K. D., & Allgöwer, B. (2004). Fire occurrence patterns at landscape level: Beyond positional accuracy of ignition points with kernel density estimation methods. *Natural Resource Modeling*, *17*(4), 359–375. <https://doi.org/10.1111/j.1939-7445.2004.tb00141.x>

MeteoSchweiz, B. für M. und K. (2019). *Klimareport 2019*. 96. Retrieved from https://www.meteoschweiz.admin.ch/content/dam/meteoswiss/de/service-und-publicationen/Publicationen/doc/klimareport_2019_de.pdf

Mulder, S., Van Vliet, A. J. H., Bron, W. A., Gassner, F., & Takken, W. (2013). High risk of tick bites in Dutch Gardens. *Vector-Borne and Zoonotic Diseases*, *13*(12). <https://doi.org/10.1089/vbz.2012.1194>

OpenStreetMap. (n.d.). OpenStreetMap. Retrieved from <https://www.openstreetmap.org>

Papoulis, A. (2002). *Probability, Random Variables and Stochastic Processes* (4th ed.). Retrieved from http://ce.sharif.edu/courses/97-98/1/ce181-1/resources/root/Text_Books_References/Papoulis_Pillai_Probability_RandomVariables_and_Stochastic_Processes-4th_Edition_2002.pdf

Paul, R. E. L., Cote, M., Le Naour, E., & Bonnet, S. I. (2016). Environmental factors influencing tick densities over seven years in a French suburban forest. *Parasites and Vectors*, *9*(1). <https://doi.org/10.1186/s13071-016-1591-5>

Perret, J. L., Guigoz, E., Rais, O., & Gern, L. (2000). Influence of saturation deficit and temperature on Ixodes ricinus tick questing activity in a Lyme borreliosis-endemic area (Switzerland). *Parasitology Research*, *86*(7), 554–557.

<https://doi.org/10.1007/s004360000209>

Porter, W. T., Motyka, P. J., Wachara, J., Barrand, Z. A., Hmood, Z., McLaughlin, M., ... Nieto, N. C. (2019). Citizen science informs human-tick exposure in the Northeastern United States. *International Journal of Health Geographics*, *18*(1), 1–14.

<https://doi.org/10.1186/s12942-019-0173-0>

Robinson L.D., Cawthray, J.L., West, S.E., Bonn, A., & Ansine, J. (2018). Ten principles of citizen science. *Citizen Science: Innovation in Open Science, Society and Policy*, 27–40.

https://doi.org/10.1007/978-94-007-6165-0_291-1

See, L., Mooney, P., Foody, G., Bastin, L., Comber, A., Estima, J., ... Rutzinger, M. (2016). Crowdsourcing, citizen science or volunteered geographic information? The current state of crowdsourced geographic information. *ISPRS International Journal of Geo-Information*, *5*(5). <https://doi.org/10.3390/ijgi5050055>

Senaratne, H., Mobasheri, A., Ali, A. L., Capineri, C., & Haklay, M. (Muki). (2017). A review of volunteered geographic information quality assessment methods. *International Journal of Geographical Information Science*, *31*(1), 139–167.

<https://doi.org/10.1080/13658816.2016.1189556>

Shi, W. (2010). *Principles of Modeling Uncertainties in Spatial Data and Spatial Analyses*. Boca Raton, FL: CRC Press.

Silverman, B. W. (1986). Density estimation: For statistics and data analysis. In *Density Estimation: For Statistics and Data Analysis*. <https://doi.org/10.1201/9781315140919>

Strobl, B., Etter, S., Van Meerveld, I., & Seibert, J. (2019). The CrowdWater game: A playful way to improve the accuracy of crowdsourced water level class data. *PLoS ONE*, *14*(9), 1–23. <https://doi.org/10.1371/journal.pone.0222579>

swisstopo. (2014). *The digital height model of Switzerland with a 200m grid*. Retrieved from <https://opendata.swiss/en/dataset/das-digitale-hohenmodell-der-schweiz-mit-einer-maschenweite-von-200-m>

Szekeres, S., Docters van Leeuwen, A., Tóth, E., Majoros, G., Sprong, H., & Földvári, G. (2019). Road-killed mammals provide insight into tick-borne bacterial pathogen communities within urban habitats. *Transboundary and Emerging Diseases*, *66*(1), 277–286. <https://doi.org/10.1111/tbed.13019>

- Turrini, T., Dörler, D., Richter, A., Heigl, F., & Bonn, A. (2018). The threefold potential of environmental citizen science - Generating knowledge, creating learning opportunities and enabling civic participation. *Biological Conservation*.
<https://doi.org/10.1016/j.biocon.2018.03.024>
- van Ravenzwaaij, D., Cassey, P., & Brown, S. D. (2018). A simple introduction to Markov Chain Monte–Carlo sampling. *Psychonomic Bulletin and Review*, 25(1), 143–154.
<https://doi.org/10.3758/s13423-016-1015-8>
- World Health Organization. (2014). A global brief on vector-borne diseases. *World Health Organization*, (September), 9. <https://doi.org/WHO/DCO/WHD/2014.1>
- World Health Organization. (2017). Global Vector Control Response 2017-2030. In *World Health Organization*. Retrieved from
<http://apps.who.int/iris/bitstream/handle/10665/259205/9789241512978-eng.pdf?sequence=1>
- ZHAW. (2019). *Research Plan - Syngery in Health*. 1–2.

10. Table of figures

Figure 1: Tick locations of Gäumann data from 2010	5
Figure 2: Municipalities with FSME (source: map.geo.admin.ch)	5
Figure 3: FSME vaccination recommendation (source: map.geo.admin.ch)	6
Figure 4: Daily reported hours	12
Figure 5: Standard report	13
Figure 6: Gaussian distribution of MCS (source: https://simion.com/info/monte_carlo.html)	16
Figure 7: Monte-Carlo-Simulation for inaccuracy	17
Figure 8: National <i>concurrent</i> KDE	20
Figure 9: Local extent	22
Figure 10: Count per year	25
Figure 11: Count per month	25
Figure 12: Count per week	25
Figure 13: Count per weekday	25
Figure 14: Crowdsourced data	26
Figure 15: Ticks per canton	27
Figure 16: Ticks per canton, normalized	27
Figure 17: Ticks per municipality, normalized	28
Figure 18: Top 11 municipalities	28
Figure 19: Height distribution	29
Figure 20: Accuracy distribution	29
Figure 21: Local extent, weighted 250kernel	31
Figure 22: Local matrix, normal KDE	33
Figure 23: Local matrix, concurrent KDE	35
Figure 24: Local matrix, sequential KDE	37
Figure 25: Local matrix, weighted KDE	39
Figure 26: National matrix, normal KDE	41
Figure 27: National matrix, concurrent KDE	43
Figure 28: National matrix, sequential KDE	44
Figure 29: National matrix, weighted KDE	46
Figure 30: Local difference within <i>weighted</i>	48
Figure 31: Local difference, <i>weighted vs sequential</i>	50
Figure 32: Difference weighted-sequential [K250, U1]	51
Figure 33: Local difference, <i>weighted vs sequential</i> , column	53
Figure 34: Local difference, <i>weighted vs sequential</i> , row	55
Figure 35: Local difference, <i>normal vs sequential</i>	57
Figure 36: National difference, <i>weighted vs sequential</i>	59
Figure 37: National difference, <i>weighted vs sequential</i> , column	60
Figure 38: National difference, <i>weighted vs sequential</i> , row	61
Figure 39: National difference, <i>normal vs sequential</i>	62

11. Appendix

11.1 Code snippets

The code snippets are structured in a similar way the paper is structured. First the loading of the data happens, then some cleaning of the data happens and after that the explorative data analysis begins by creating histograms. Afterwards the KDE function is shown. The MCS function is under section 11.2.3, rather at the beginning, after loading the subset (local scale). Then one example is showing how the MCS and KDE were combined to create the 500m kernel of the *concurrent* small multiple. If this would be continued for the other kernels as well a small multiple could be created which is shown further below in 11.2.3.

11.1.1 Histograms

```
#load packages
```{r directorySetup,comment=FALSE, message = FALSE,
echo=FALSE,warning=TRUE}
library(sf)
library(dplyr)
library(lubridate)
library(tidyverse)
library(tmap)
library(raster)
library(openxlsx)
library(rgdal)
library(ggplot2)
library(cowplot)
library(tidyr)
library(colorspace)
```

#loading data + everything into the Global Environment
```{r, warning=TRUE, message=TRUE, fig.align="center"}
getwd()
wd <- setwd(getwd())

read in main tick data
tick <- read.csv("data/20200810-tick-reports-48000.csv", sep = ",",
 encoding = "UTF-8",
 stringsAsFactors = FALSE) #load in data
tick = st_as_sf(tick, coords=c("x","y"), crs=21781)
#str(tick) ##checks how R is reading the data (geometry, numbers,
characters, etc.)
tick$date2 <-as.POSIXct(x = tick$date, format= "%d.%m.%Y %H:%M")
tick <- filter(tick, date2 >= "2015-01-01 00:00") ##there were
about 5 entries from 1970? this way they are removed
tick <- filter(tick, acc. != 1064, acc. != 68903, acc. != 1153, acc. !=
74650, acc. != 1014, acc. != 333, acc. != 974, acc. != 63040, acc. != 302,
acc. != 994, acc. != 64349, acc. != 73102, acc. != 1129, acc. != 1044, acc.
!= 67607, acc. != 972, acc. != 62963) ##removes all the standard reports
(from experiment)
```

```

mutate(tick, datum=as.Date(date2)) ##ergänzen, creates new column in
ticks. convenience variables
tick <- mutate(tick, year=year(date2), month=month(date2),
week=week(date2), day=day(date2), hour=hour(date2),
weekday=as.POSIXlt(tick$date2)$wday) # adds convenience variables year,
month, week, day, hour, and weekday 0=Sunday & 1=Monday)
tick$weekday[tick$weekday==0] <- 7
ticksf_dataframe <- st_as_sf(tick, coords = c("x", "y"), crs= 21781)
##sf is better than spatial_data_frame. Nils thinks so

#read in swiss boundaries
ch_boundaries_sf1 <-
read_sf("Schweiz/swissBOUNDARIES3D_1_3_TLM_KANTONSGBIET.shp") ##reads in
the shp of Switzerland and its Cantons
ch_boundaries_sf = st_transform(ch_boundaries_sf1, crs =
st_crs(ticksf_dataframe)) ## transforms the projection to the same as the
tick data (crs=21781)
#st_geometry(ch_boundaries_sf) ##checks the projection type (crs=?)

#only taking ticks withing Switzerland
union_CH <-st_zm(st_union(ch_boundaries_sf)) #unites the 42 polygons
(Kantone) together to one big polygon
boolian_CH <-st_contains(union_CH, ticksf_dataframe, sparse = FALSE)[1,]
#show me all the ticks withing Switzerland
zecken <-filter(ticksf_dataframe, boolian_CH) #only take the ticks withing
Switzerland
 boxplot(zecken$acc. , outline = FALSE)
 boxplot.stats(zecken$acc.)$stats
 # # Min. 1st Qu. Median 3rd Qu. Max.
 #[1] 2.0000 138.4105 301.7485 910.0000 2130
zecken <- filter(zecken, acc. <= 2130) #only shows the ticks which are
2065.5218m radius of smaller (95% of all reports in CH)

#create subset SG of zecken
SG <- subset(ch_boundaries_sf, ch_boundaries_sf$NAME == c("St.
Gallen","Appenzell Innerrhoden","Appenzell Innerrhoden")) #only takes
St.Gallen from the boundaries.
union_CH <-st_zm(st_union(SG)) #since there are 2 polygons, it unites them
boolian_CH <-st_contains(union_CH, zecken, sparse = FALSE)[1,] #yes/no
which points lie within SG?
SG <-filter(zecken, boolian_CH) #only take points within SG

tickbuffer <-st_buffer(zecken, zecken$acc.) #creates a buffer around the
tick reports --> makes them into polygons

gemeinde <- read_sf("ag-b-00.03-875-gg20/ggg_2020-LV95/shp/glg20.shp")
#reading in municipality boundaries
gemeinde_sf = st_transform(gemeinde, crs = st_crs(ticksf_dataframe))
#make gemeinde have the same csr as zecken

zecken <- st_join(zecken, ch_boundaries_sf) #works
names(zecken)[names(zecken) == "NAME"] <- "kanton" #works changes the
column name "NAME" to kanton
zecken <-
 #creates new column with the abbreviations of each
kanton
 mutate(zecken, abkurtz = case_when(kanton == "Aargau" ~ 'AG',
 kanton == "Appenzell Ausserrhoden"~ 'AR',

```

```

kanton == "Appenzell Innerrhoden"~ 'AI',
kanton == "Basel-Landschaft"~ 'BL',
kanton == "Basel-Stadt"~ 'BS',
kanton == "Bern"~ 'BE',
kanton == "Fribourg"~ 'FR',
kanton == "Genève"~ 'GE',
kanton == "Glarus"~ 'GL',
kanton == "Graubünden"~ 'GR',
kanton == "Jura"~ 'JU',
kanton == "Luzern"~ 'LU',
kanton == "Neuchâtel"~ 'NE',
kanton == "Nidwalden"~ 'NW',
kanton == "Obwalden"~ 'OW',
kanton == "Schaffhausen"~ 'SH',
kanton == "Schwyz"~ 'SZ',
kanton == "Solothurn"~ 'SO',
kanton == "St. Gallen"~ 'SG',
kanton == "Ticino"~ 'TI',
kanton == "Thurgau"~ 'TG',
kanton == "Uri"~ 'UR',
kanton == "Vaud"~ 'VD',
kanton == "Valais"~ 'VS',
kanton == "Zug"~ 'ZG',
kanton == "Zürich"~ 'ZH'))
kanton_amount <- as.data.frame(table(zecken$abkurz)) #works returns number
of reports from each Kanton
#kanton_amount

zg_join <- st_join(zecken, gemeinde_sf)
gemeinde_amount <- as.data.frame(table(zg_join$GMDNAME)) #works returns
number of reports from each municipality
#gemeinde_amount

dh = raster("datahöhenmodell/DHM200.asc") #loads in the elevation model of
Switzerland

unloadNamespace("tidyverse") #unload to unload modelr
unloadNamespace("modelr") #unload to unload broom
unloadNamespace("broom") #unload to unload tidyr
unloadNamespace("tidyr") #unload because another extract is used by the
raster package
zg_join$height <- extract(dh, zecken) #creates a new column in
zg_join and assigns every data point a specific height
zecken$height <- extract(dh, zecken)
##for extract to work tidyverse needs to be turned off.

users <- unique(zecken$X..uuid.) #17116 unique users
####
...

#Plotting histogram
```{r, warning=FALSE, message=FALSE}
zecken %>% group_by(year = year(date2)) %>% count() %>% ggplot(aes(year,n))
+ geom_col() + scale_x_continuous(breaks = 2015:2020, labels = 2015:2020) #
shows yearly count
#

```

```

zecken %>% group_by(month) %>% count() %>% ggplot(aes(month,n)) +
geom_col() + scale_x_continuous(breaks = 1:12, labels = 1:12) #shows
monthly count
#
zeck <- filter(zecken, week>=15)
zeck <- filter(zeck, week<=30 )
zeck %>% group_by(week) %>% count() %>% ggplot(aes(week,n)) + geom_col() +
scale_x_continuous(breaks = seq(16,30, 2))
#
ggplot(zecken, aes(weekday))+geom_histogram()#shows on which weekdays how
many ticks happened
zecken %>% group_by(weekday) %>% count() %>% ggplot(aes(weekday,n)) +
geom_col() + scale_x_continuous(breaks = 1:7, labels = 1:7) #shows weekday
count
#
ggplot(zecken, aes(acc.))+geom_histogram(binwidth = 50)
# min5% 1st quart median 3rd quart max 95%
# 2.0000 122.0000 301.7485 715.0000 1604.4765
ggplot(zecken, aes(date.acc.)) +geom_histogram(binwidth=216000)
zeck <- filter(zecken, height<=2000 )
ggplot(zeck, aes(height))+geom_histogram() + scale_x_continuous(breaks =
seq(0,2000, 250))
boxplot.stats(zecken$acc.)$stats
# min5% 1st quart median 3rd quart max 95%
# 193.0000 466.5025 555.6000 696.5000 1041.2980
zeck <- filter(zecken, height <=2000)
boxplot.stats(zeck$height)$stats
#
kanton_amount %>%
mutate(name= fct_reorder(Var1, desc(Freq))) %>%
ggplot(aes(x=name, y=Freq)) +
geom_bar(stat="identity", fill="#f68060", alpha=.6, width=.4) +
coord_flip() +
xlab("") +
theme_bw()
#plots all kantons and orders them in ascending order in how many
reports were taken in which canton.
gemein <- st_read("Gemeindeanzahl.xlsx")
zg_join2 <- read.csv("zg_join2.csv", sep = ",",
encoding = "UTF-8",
stringsAsFactors = FALSE)
names(gemein)[names(gemein) == "BevÄ.lkerung"] <- "population" #renames the
column to an easier name
as.data.frame(gemein)
gemeinde_amount <- as.data.frame(table(zg_join2$GMDNAME))
mergeGemein <- merge(gemeinde_amount, gemein, by.x="Var1", by.y="Gemeinde")
#joins the pop density with the amount of ticks per municipality
mergeGemein <- mutate(mergeGemein, norm_tick= Freq/population)

mergeGemein_filter <- filter(mergeGemein, norm_tick >= 0.04) #only takes
the norm_tick which are higher than 1

mergeGemein_filter %>%
mutate(name= fct_reorder(Var1, desc(norm_tick))) %>%
ggplot(aes(x=name, y=norm_tick)) +
geom_bar(stat="identity", fill="#f68060", alpha=.6, width=.4) +
coord_flip() +
xlab("") +
theme_bw()

```

```

#creates a plot that shows which Kanton has the highest tick reports per
population density (hence normalized data)
##read in data
kanton_pop <- st_read("kanton_pop_density.xlsx")
as.data.frame(kanton_pop)
zecken_INkanton <- as.data.frame(table(zecken$abkurz))
mergeKanton_pop <- merge(zecken_INkanton, kanton_pop, by.x="Var1",
by.y="Kanton") #joins the pop density with the amount of ticks per Kanton
mergeKanton_pop <- mutate(mergeKanton_pop, norm_tick= Freq/population)

mergeKanton_pop %>%
  mutate(name= fct_reorder(Var1, desc(norm_tick))) %>%
  ggplot(aes(x=name, y=norm_tick)) +
    geom_bar(stat="identity", fill="#f68060", alpha=.6, width=.4) +
    coord_flip() +
    xlab("") +
    theme_bw()
zecken_INkanton %>%
  mutate(name= fct_reorder(Var1, desc(Freq))) %>%
  ggplot(aes(x=name, y=Freq)) +
    geom_bar(stat="identity", fill="#f68060", alpha=.6, width=.4) +
    coord_flip() +
    xlab("") +
    theme_bw()
...

```


11.1.2 KDE

#4.6 Appendix VII: Custom functions

The following **function** is a helper **function for** "kernel_density". It takes an "extent" object and enlarges it so that it is a multiple of "cellsize". The function returns a new, adjusted "extent" object.

https://github.zhaw.ch/pages/FightingBitesWithBytes/GIS_Exposure/index.html
https://github.zhaw.ch/pages/FightingBitesWithBytes/GIS_Exposure/appendix.html#appendix-vii-custom-functions

```
```{r}
extent_by_cellsize <- function(extent_old, cellsize){
 require(raster)

 xrange <- c(extent_old[2], extent_old[1])
 yrange <- c(extent_old[3], extent_old[4])

 xrange_diff <- abs(diff(xrange))
 yrange_diff <- abs(diff(yrange))

 enlarge_x <- ((ceiling(xrange_diff/cellsize)*cellsize)-xrange_diff)/2
 enlarge_y <- ((ceiling(yrange_diff/cellsize)*cellsize)-yrange_diff)/2

 extent_new <- extent(
 matrix(
 c(
 extent_old[1]-enlarge_x,
 extent_old[2]+enlarge_x,
 extent_old[3]-enlarge_y,
 extent_old[4]+enlarge_y
), 2, 2, TRUE)
)

 extent_new
}
```
```

The following function is mainly a wrapper around `spatstat::density` to work with `sf` objects. Additionally, it also simplifies said function to suit the needs in a more succinct manner. Input:

```
```{r}
-"sf_object": A point-object of class "sf"
-"cellsize": Cellsize in units of the input data (usually meters). The
resulting raster tries to match the cellsize exactly by modifying
(enlarging) the output extent with "extent_by_cellsize"
-"sigma": is directly passed on to "spatstat::density.ppp()".
Basically, this is the search radius in units of the input data
-"boundary": An "sf"-polygon of the boundary within which densities
should be estimated (e.g. country border). The function "st_union" will
be performed on this object, and it should contain only one feature after
this operation. I don't know what will happen if this is not the
case.
-"holes": An "sf"-polygon of the holes within the boundary, where no
densities should be estimated (for example "lakes"). If the polygon
itself contains holes, these will be removed. I don't know what will
happen if this object is a multipolygon or if these holes are
outside "boundary"
-"weights", "edge", "varcov"... are directly passed on to
spatstat::density.ppp()

sf_object <- zecken
```

```

cellsize <- 2000

kernel_density <- function(sf_object, #my tick points sf_object
<- zk_join
 cellsize, #what my cellsize should
be. define before. in meters.
 sigma=NULL,
 boundary = NULL, #if NULL take the extend of
my data set and make that my boundaries
 holes = NULL,
 weights=NULL, #need to define the
inverse of my acc. if I want this to take part
 edge=TRUE,
 varcov=NULL,
 at="pixels",
 leaveoneout=TRUE,
 adjust=1,
 diggle=FALSE,
 se=FALSE,
 kernel="gaussian", #maybe I want another
version of KDE
 scalekernel=is.character(kernel),
 positive=FALSE,
 verbose=TRUE){

 require(spatstat)
 require(maptools)
 require(sf)
 require(raster)
 start <- Sys.time() #just starts a timer to show me how long
everything took until a certain point.
 stopifnot(!(is.null(boundary) & !is.null(holes))) #if I have no
boundaries, but I have holes (example: lake), it will stop the function,
because it would be weird.

 if(!is.null(boundary)){ #if i have boundaries set, do
something.
 boundary_union <- st_union(boundary)
 stopifnot(length(boundary_union)==1)
 boundary_union_coords <- (st_coordinates(boundary_union))

 poly <- list(list(x = rev(boundary_union_coords[,1]), y =
rev(boundary_union_coords[,2])))
 ex <- extent(boundary)
 } else{
 ex <- extent(sf_object) #if I don't have boundaries, take
the extend of sf_object (mydataset).
 poly <- NULL
 }
 if(!is.null(holes)){

 # holes_remove <- st_as_sfc(lapply(st_geometry(holes), function(x)
st_polygon(x[1])))
 holes_coordinates <- holes %>% #cuts out lakes etc. (everything
I dont want) and adds it to boundaries.
 st_geometry() %>%
 lapply(function(x){
 mat <- x[1]
 list(x = mat[[1]][,1],
 y = mat[[1]][,2])
 })
 }
}

```

```

 for(i in seq_along(holes_coordinates)){
 poly[[i+1]] <- holes_coordinates[[i]]
 }
 }

 ex_new <- extent_by_cellsize(extent_old = ex, cellsize = cellsize) #takes
the cellsize and enhances the extend so that the cellsize fits nicely into
it.

 xrange <- c(ex_new[1],ex_new[2])
 yrange <- c(ex_new[3],ex_new[4])

 # create an observation window from polygon and holes
 w <- owin(xrange = xrange, yrange = yrange, poly = poly, unitname =
c("meter", "meters")) #owin = observation window.
 sf_coords <- st_coordinates(sf_object)
 x <- ppp(sf_coords[,1], sf_coords[,2], window = w, marks =
st_set_geometry(sf_object, NULL)) #turns the sf_object into a ppp-object +
window

 dimyx <-
round(c((abs(diff(yrange)))/cellsize, (abs(diff(xrange)))/cellsize))
#tells me how many cells are on the X and Y axis.

 des <- density.ppp(x, sigma=sigma, dimyx = dimyx, weights=weights,
edge=edge, varcov=varcov, at=at, leaveoneout=leaveoneout, adjust=adjust,
diggle=diggle, se=se, kernel=kernel, scalekernel=scalekernel,
positive=positive, verbose=verbose) #the KDE function

 ras <- raster(des) #makes a raster out of it
 crs(ras) <- crs(sf_object) #not sure anymore
 print(difftime(Sys.time(), start)) #stops the time that we started at the
beginning
 ras #shows the KDE on the map
}
...

```

### 11.1.3 creating small multiples

```
#load in necessary data
```{r}
library(sf)
library(dplyr)
library(lubridate)
library(tidyverse)
library(tmap)
library(raster)
library(rgdal)
library(ggplot2)
library(cowplot)
library(tidyr)
library(colorspace)
library(Metrics)

getwd()
wd <- setwd(getwd())

#
knutwil_sf <- read_sf("data/Ausschnitt/new_knutwilV2.shp")
knutwil <- st_as_sf(knutwil_sf, coords = c("x", "y"), crs= 21781)
knutwil = st_transform(knutwil, crs = st_crs(21781))
knutwil <-st_cast(knutwil, "POINT", group_or_split = FALSE)
knutwil <- filter(knutwil, acc. <= 2130)
#
extent_sub <- read_sf("data/extent_subset_LU/subset_LU.shp")

...

#creating a Monte-Carlo-Simulation function
```{r}
n = 50 #how many iterations?

#tmap_mode("view") #makes the map interactive
#tmap_mode(mode = c("plot", "view")) #makes the map not interactive
point <- knutwil$geometry
sf_object <- knutwil
maxdist <- knutwil$acc.

move_point <- function(point, maxdistance){

 n_points <- length(point)

 angle_deg <- runif(n_points,1,360)

 distance <- rnorm(n_points,mean = 0,sd = maxdistance/2)

 angle_rad <- (angle_deg * pi) / (180)

 point_old <- st_coordinates(point)

 xoffset <- cos(angle_rad) * distance
 yoffset <- sin(angle_rad) * distance

 point_new <- point_old + matrix(c(xoffset,yoffset), ncol = 2)

 point_new <- point_new %>%
 split(1:nrow(.)) %>%
```

```

 map(~st_point(.x))%>%
 st_sfc()

 point_new
 }
 ...
#matrix maps for concurrent
##500
```{r}
#1 acc-radius
oneMonte <- map_dfr(1:n, function(x){          # a map with tons of points on
it
  at <-move_point(knutwil$geometry, knutwil$acc.) #assigns the move_point
function to at
  #at <- st_as_sf(at, crs=21781)          #assigns all the list-points a
projection
  at <- st_set_crs(at,21781)
  st_as_sf(at)
})

kdeoneMonte <- kernel_density(sf_object = oneMonte, cellsize =100, sigma =
500, boundary = extent_sub) #takes the oneMonte and turns it into a KDE-map
kdeoneMonte <-kdeoneMonte/50
crs(kdeoneMonte) <- st_crs(21781)$proj4string #gives correct projection
kdeoneMonte <- kdeoneMonte*1000000
##tm_shape(kdeoneMonte) + tm_raster(alpha = 0.5, midpoint = NA, palette="-
RdYlGn") ##### saved as: SG_c100_s500/50oneMCS
#writeRaster(kdeoneMonte,"knutwilkm_kde_concurrent_500.tiff")
mean_concurrent1_500 <- cellStats(kdeoneMonte, stat = 'mean')
mean_concurrent1_500
knutwil_concurrent1_500 <- kdeoneMonte

#0.5 acc-radius
knutwil$acc.2 <- knutwil$acc.*0.5
oneMonte <- map_dfr(1:n, function(x){          # a map with tons of points on
it
  at <-move_point(knutwil$geometry, knutwil$acc.2) #assigns the move_point
function to at
  #at <- st_as_sf(at, crs=21781)          #assigns all the list-points a
projection
  at <- st_set_crs(at,21781)
  st_as_sf(at)
})
kdeoneMonte500_05 <- kernel_density(sf_object = oneMonte, cellsize= 100,
sigma = 500, boundary = extent_sub) #takes the oneMonte and turns it into a
KDE-map
kdeoneMonte500_05 <-kdeoneMonte500_05/50
crs(kdeoneMonte500_05) <- st_crs(21781)$proj4string #gives correct
projection
kdeoneMonte500_05 <- kdeoneMonte500_05*1000000
##tm_shape(kdeoneMonte500_05) + tm_raster(alpha = 0.5, midpoint = NA,
palette="-RdYlGn") ##### saved as: SG_c100_s500/50oneMCS
#writeRaster(kdeoneMonte500_05,"knutwilkm_kde_concurrent0.5_500.tiff")
mean_concurrent05_500 <- cellStats(kdeoneMonte500_05, stat = 'mean')
mean_concurrent05_500
knutwil_concurrent05_500 <- kdeoneMonte500_05

#2 acc-radius
knutwil$acc.2 <- knutwil$acc.*2
oneMonte <- map_dfr(1:n, function(x){          # a map with tons of points on
it

```

```

    at <-move_point(knutwil$geometry, knutwil$acc.2) #assigns the move_point
function to at
    #at <- st_as_sf(at, crs=21781) #assigns all the list-points a
projection
    at <- st_set_crs(at,21781)
    st_as_sf(at)
  })
kdeoneMonte500_2 <- kernel_density(sf_object = oneMonte, cellsize= 100,
sigma = 500, boundary = extent_sub) #takes the oneMonte and turns it into a
KDE-map
kdeoneMonte500_2 <-kdeoneMonte500_2/50
crs(kdeoneMonte500_2) <- st_crs(21781)$proj4string #gives correct
projection
kdeoneMonte500_2 <- kdeoneMonte500_2*1000000
##tm_shape(kdeoneMonte500_2) + tm_raster(alpha = 0.5, midpoint = NA,
palette="-RdYlGn") #### saved as: SG_c100_s500/50oneMCS
#writeRaster(kdeoneMonte500_2,"knutwilkm_kde_concurrent2_500.tiff")
mean_concurrent2_500 <- cellStats(kdeoneMonte500_2, stat = 'mean')
mean_concurrent2_500
knutwil_concurrent2_500 <- kdeoneMonte500_2

#10 acc-radius
knutwil$acc.2 <- knutwil$acc.*10
oneMonte <- map_dfr(1:n, function(x){ # a map with tons of points on
it
    at <-move_point(knutwil$geometry, knutwil$acc.2) #assigns the move_point
function to at
    #at <- st_as_sf(at, crs=21781) #assigns all the list-points a
projection
    at <- st_set_crs(at,21781)
    st_as_sf(at)
  })
kdeoneMonte500_10 <- kernel_density(sf_object = oneMonte, cellsize= 100,
sigma = 500, boundary = extent_sub) #takes the oneMonte and turns it into a
KDE-map
kdeoneMonte500_10 <-kdeoneMonte500_10/50
crs(kdeoneMonte500_10) <- st_crs(21781)$proj4string #gives correct
projection
kdeoneMonte500_10 <- kdeoneMonte500_10*1000000
##tm_shape(kdeoneMonte500_10) + tm_raster(alpha = 0.5, midpoint = NA,
palette="-RdYlGn") #### saved as: SG_c100_s500/50oneMCS
#writeRaster(kdeoneMonte500_10,"knutwilkm_kde_concurrent10_500.tiff")
mean_concurrent05_500 <- cellStats(kdeoneMonte500_10, stat = 'mean')
mean_concurrent05_500
knutwil_concurrent10_500 <- kdeoneMonte500_10
...

...

#difference withing matrix
```{r}
rounding <- function(x,k) trimws(format(round(x, k), nsmall=k)) #rounding
creates a function which spits out the different distribution errors
measure <- function(diff raster, r1, r2){
 #prep
 diff raster1 <- as.data.frame(diff raster, xy = TRUE)%>% na.omit()
 r1 <- as.data.frame(r1, xy = TRUE)%>% na.omit()
 r2 <- as.data.frame(r2, xy = TRUE)%>% na.omit()
 mean <- mean(diff raster1$layer, na.rm=TRUE)
 #standard deviation = SD mean(variance^2)

```

```

 print(paste0("SD=", (rounding((sd (values(diffrafter), na.rm=TRUE)),2))
, " | ",
 #mean error = ME
 "ME=", (rounding((mean(diffrafter1$layer, na.rm=TRUE)),2))
, "\n",
 #mean absolute error = MAE
 "MAE=", (rounding(mae(r1$layer, r2$layer),2)), "|",
 #Root Mean Square Error = RMSE
 "RMSE=", (rounding(rmse(r1$layer, r2$layer),2)))) #anzahl
pixel (ohne NA)
}
#
##now just to see if the change is positive or negative. 1=negative,
2=positive
reclass_df <- c(
 -999, 0, 1,
 0, 999, 2)

reclass_m <- matrix(reclass_df,
 ncol = 3,
 byrow = TRUE)

...

#measuring differences within matrix
##concurrent
###columns (top to bottom, left ro right)
```{r}
d05a <- knutwil_concurrent05_100-knutwil_concurrent05_250
d05a_neg <- rounding(sum(d05a[d05a<0]),0)
d05a_pos <- rounding(sum(d05a[d05a>0]),0)
#d05a <- reclassify(d05a,reclass_m)
#
d05b <- knutwil_concurrent05_250-knutwil_concurrent05_500
d05b_neg <- rounding(sum(d05b[d05b<0]),0)
d05b_pos <- rounding(sum(d05b[d05b>0]),0)
#d05b <- reclassify(d05b,reclass_m)
#
d05c <- knutwil_concurrent05_500-knutwil_concurrent05_1000
d05c_neg <- rounding(sum(d05c[d05c<0]),0)
d05c_pos <- rounding(sum(d05c[d05c>0]),0)
#d05c <- reclassify(d05c,reclass_m)
#
d1a <- knutwil_concurrent1_100-knutwil_concurrent1_250
d1a_neg <- rounding(sum(d1a[d1a<0]),0)
d1a_pos <- rounding(sum(d1a[d1a>0]),0)
#d1a <- reclassify(d1a,reclass_m)
#
d1b <- knutwil_concurrent1_250-knutwil_concurrent1_500
d1b_neg <- rounding(sum(d1b[d1b<0]),0)
d1b_pos <- rounding(sum(d1b[d1b>0]),0)
#d1b <- reclassify(d1b,reclass_m)
#
d1c <- knutwil_concurrent1_500-knutwil_concurrent1_1000
d1c_neg <- rounding(sum(d1c[d1c<0]),0)
d1c_pos <- rounding(sum(d1c[d1c>0]),0)
#d1c <- reclassify(d1c,reclass_m)
#
d2a <- knutwil_concurrent2_100-knutwil_concurrent2_250
d2a_neg <- rounding(sum(d2a[d2a<0]),0)
d2a_pos <- rounding(sum(d2a[d2a>0]),0)
#d2a <- reclassify(d2a,reclass_m)

```

```

#
d2b <- knutwil_concurrent2_250-knutwil_concurrent2_500
d2b_neg <- rounding(sum(d2b[d2b<0]),0)
d2b_pos <- rounding(sum(d2b[d2b>0]),0)
#d2b <- reclassify(d2b,reclass_m)
#
d2c <- knutwil_concurrent2_500-knutwil_concurrent2_1000
d2c_neg <- rounding(sum(d2c[d2c<0]),0)
d2c_pos <- rounding(sum(d2c[d2c>0]),0)
#d2c <- reclassify(d2c,reclass_m)
#
d10a <- knutwil_concurrent10_100-knutwil_concurrent10_250
d10a_neg <- rounding(sum(d10a[d10a<0]),0)
d10a_pos <- rounding(sum(d10a[d10a>0]),0)
#d10a <- reclassify(d10a,reclass_m)
#
d10b <- knutwil_concurrent10_250-knutwil_concurrent10_500
d10b_neg <- rounding(sum(d10b[d10b<0]),0)
d10b_pos <- rounding(sum(d10b[d10b>0]),0)
#d10b <- reclassify(d10b,reclass_m)
#
d10c <- knutwil_concurrent10_500-knutwil_concurrent10_1000
d10c_neg <- rounding(sum(d10c[d10c<0]),0)
d10c_pos <- rounding(sum(d10c[d10c>0]),0)
#d10c <- reclassify(d10c,reclass_m)
#
...
###rows (left ro right, top to bottom)
```{r}
d1000a <- knutwil_concurrent05_1000-knutwil_concurrent1_1000
d1000a_neg <- rounding(sum(d1000a[d1000a<0]),0)
d1000a_pos <- rounding(sum(d1000a[d1000a>0]),0)
#d1000a <- reclassify(d1000a,reclass_m)
#
d1000b <- knutwil_concurrent1_1000-knutwil_concurrent2_1000
d1000b_neg <- rounding(sum(d1000b[d1000b<0]),0)
d1000b_pos <- rounding(sum(d1000b[d1000b>0]),0)
#d1000b <- reclassify(d1000b,reclass_m)
#
d1000c <- knutwil_concurrent2_1000-knutwil_concurrent10_1000
d1000c_neg <- rounding(sum(d1000c[d1000c<0]),0)
d1000c_pos <- rounding(sum(d1000c[d1000c>0]),0)
#d1000c <- reclassify(d1000c,reclass_m)
#
d500a <- knutwil_concurrent05_500-knutwil_concurrent1_500
d500a_neg <- rounding(sum(d500a[d500a<0]),0)
d500a_pos <- rounding(sum(d500a[d500a>0]),0)
#d500a <- reclassify(d500a,reclass_m)
#
d500b <- knutwil_concurrent1_500-knutwil_concurrent2_500
d500b_neg <- rounding(sum(d500b[d500b<0]),0)
d500b_pos <- rounding(sum(d500b[d500b>0]),0)
#d500b <- reclassify(d500b,reclass_m)
#
d500c <- knutwil_concurrent2_500-knutwil_concurrent10_500
d500c_neg <- rounding(sum(d500c[d500c<0]),0)
d500c_pos <- rounding(sum(d500c[d500c>0]),0)
#d500c <- reclassify(d500c,reclass_m)
#
d250a <- knutwil_concurrent05_250-knutwil_concurrent1_250
d250a_neg <- rounding(sum(d250a[d250a<0]),0)

```



```

d250a_pos <- rounding(sum(d250a[d250a>0]),0)
#d250a <- reclassify(d250a,reclass_m)
#
d250b <- knutwil_concurrent1_250-knutwil_concurrent2_250
d250b_neg <- rounding(sum(d250b[d250b<0]),0)
d250b_pos <- rounding(sum(d250b[d250b>0]),0)
#d250b <- reclassify(d250b,reclass_m)
#
d250c <- knutwil_concurrent2_250-knutwil_concurrent10_250
d250c_neg <- rounding(sum(d250c[d250c<0]),0)
d250c_pos <- rounding(sum(d250c[d250c>0]),0)
#d250c <- reclassify(d250c,reclass_m)
#
d100a <- knutwil_concurrent05_100-knutwil_concurrent1_100
d100a_neg <- rounding(sum(d100a[d100a<0]),0)
d100a_pos <- rounding(sum(d100a[d100a>0]),0)
#d100a <- reclassify(d100a,reclass_m)
#
d100b <- knutwil_concurrent1_100-knutwil_concurrent2_100
d100b_neg <- rounding(sum(d100b[d100b<0]),0)
d100b_pos <- rounding(sum(d100b[d100b>0]),0)
#d100b <- reclassify(d100b,reclass_m)
#
d100c <- knutwil_concurrent2_100-knutwil_concurrent10_100
d100c_neg <- rounding(sum(d100c[d100c<0]),0)
d100c_pos <- rounding(sum(d100c[d100c>0]),0)
#d100c <- reclassify(d100c,reclass_m)
#
...
...
#concurrent cowplot
##creating ggplot for cowplot
```{r}
#beide
b <- function(normal_kde, rechts_pos, rechts_neg, rechts_measure, rr1, rr2,
oben_pos, oben_neg, oben_measure, rol){
  as.data.frame(normal_kde, xy=TRUE) %>%drop_na() %>% mutate(rechts =
print(paste0("pos: ",rechts_pos," | ", "neg: ", rechts_neg, "\n",
print(measure(rechts_measure, rr1, rr2))),
                                oben=
print(paste0("pos: ",oben_pos," | ", "neg: ", oben_neg, "\n",
print(measure(oben_measure, rol, rr1)))) %>%
  ggplot() + geom_raster(aes(x=x, y=y, fill=layer)) +
  scale_fill_gradientn(colours = hcl.colors(10, "YlGnBu"), breaks = c(10,5,
-0.001), trans = "reverse", limits =c(10,-0.001), na.value="navyblue")+
  theme(legend.position = "none", axis.title.x=element_blank(),
        axis.text.x=element_blank(),axis.ticks.x=element_blank(),
axis.title.y=element_blank(), axis.text.y=element_blank(),
        axis.ticks.y=element_blank())+
  facet_grid(rechts~.~oben)
}

b5dots <- b(normal_kde = knutwil_concurrent05_250, rechts_pos = d250a_pos,
rechts_neg = d250a_neg, rechts_measure = d250a, rr1 =
knutwil_concurrent05_250, rr2 = knutwil_concurrent1_250, #done
        oben_pos = d05a_pos, oben_neg = d05a_neg, oben_measure =
d05a, rol = knutwil_concurrent05_100)

```

```

b6dots <- b(normal_kde = knutwil_concurrent1_250, rechts_pos = d250b_pos,
rechts_neg = d250b_neg, rechts_measure = d250b, rr1 =
knutwil_concurrent1_250, rr2 = knutwil_concurrent2_250, #done
      oben_pos = d1a_pos,      oben_neg = d1a_neg,      oben_measure =
d1a, ro1 = knutwil_concurrent1_100)

b7dots <- b(normal_kde = knutwil_concurrent2_250, rechts_pos = d250c_pos,
rechts_neg = d250c_neg, rechts_measure = d250c, rr1 =
knutwil_concurrent2_250, rr2 = knutwil_concurrent10_250, #done
      oben_pos = d2a_pos,      oben_neg = d2a_neg,      oben_measure =
d2a, ro1 = knutwil_concurrent2_100)

b9dots <- b(normal_kde = knutwil_concurrent05_500, rechts_pos = d500a_pos,
rechts_neg = d500a_neg, rechts_measure = d500a, rr1 =
knutwil_concurrent05_500, rr2 = knutwil_concurrent1_500, #done
      oben_pos = d05b_pos,      oben_neg = d05b_neg,      oben_measure =
d05b, ro1 = knutwil_concurrent05_250)

b10dots <- b(normal_kde = knutwil_concurrent1_500, rechts_pos = d500b_pos,
rechts_neg = d500b_neg, rechts_measure = d500b, rr1 =
knutwil_concurrent1_500, rr2 = knutwil_concurrent2_500, #done
      oben_pos = d1b_pos,      oben_neg = d1b_neg,      oben_measure =
d1b, ro1 = knutwil_concurrent1_250)

b11dots <- b(normal_kde = knutwil_concurrent2_500, rechts_pos = d500c_pos,
rechts_neg = d500c_neg, rechts_measure = d500c, rr1 =
knutwil_concurrent2_500, rr2 = knutwil_concurrent10_500, #done
      oben_pos = d2b_pos,      oben_neg = d2b_neg,      oben_measure =
d2b, ro1 = knutwil_concurrent2_250)

b13dots <- b(normal_kde = knutwil_concurrent05_1000, rechts_pos =
d1000a_pos, rechts_neg = d1000a_neg, rechts_measure = d1000a, rr1 =
knutwil_concurrent05_1000, rr2 = knutwil_concurrent1_1000, #done
      oben_pos = d05c_pos,      oben_neg = d05c_neg,      oben_measure =
d05c, ro1 = knutwil_concurrent05_500)

b14dots <- b(normal_kde = knutwil_concurrent1_1000, rechts_pos =
d1000c_pos, rechts_neg = d1000b_neg, rechts_measure = d1000b, rr1 =
knutwil_concurrent1_1000, rr2 = knutwil_concurrent2_1000,
      oben_pos = d1c_pos,      oben_neg = d1c_neg,      oben_measure =
d1c, ro1 = knutwil_concurrent1_500)

b15dots <- b(normal_kde = knutwil_concurrent2_1000, rechts_pos =
d1000c_pos, rechts_neg = d1000c_neg, rechts_measure = d1000c, rr1 =
knutwil_concurrent2_1000, rr2 = knutwil_concurrent10_1000,
      oben_pos = d2c_pos,      oben_neg = d2c_neg,      oben_measure =
d2c, ro1 = knutwil_concurrent2_500)

#####
#right

r <- function(normal_kde, rechts_pos, rechts_neg, rechts_measure, rr1, rr2,
olabel){
  as.data.frame(normal_kde, xy=TRUE) %>%drop_na() %>% mutate(rechts =
print(paste0("pos: ",rechts_pos," | ", "neg: ", rechts_neg, "\n",
print(measure(rechts_measure, rr1, rr2))))), oben = print(paste0(olabel)))
%>%
  ggplot() + geom_raster(aes(x=x, y=y, fill=layer)) +

```

```

    scale_fill_gradientn(colours = hcl.colors(10, "YlGnBu"), breaks =
c(10,5, -0.001), trans = "reverse", limits =c(10,-0.001),
na.value="navyblue")+
    theme(legend.position = "none", axis.title.x=element_blank(),
        axis.text.x=element_blank(),axis.ticks.x=element_blank(),
axis.title.y=element_blank(), axis.text.y=element_blank(),
        axis.ticks.y=element_blank())+
    facet_grid(rechts~.~oben)
}

r1dots <- r(normal_kde = knutwil_concurrent05_100, rechts_pos = d100a_pos,
rechts_neg = d100a_neg, rechts_measure = d100a, rr1 =
knutwil_concurrent05_100, rr2 = knutwil_concurrent1_100, 0.5) #done
r2dots <- r(normal_kde = knutwil_concurrent1_100, rechts_pos = d100b_pos,
rechts_neg = d100b_neg, rechts_measure = d100b, rr1 =
knutwil_concurrent1_100, rr2 = knutwil_concurrent2_100, 1) #done
r3dots <- r(normal_kde = knutwil_concurrent2_100, rechts_pos = d100c_pos,
rechts_neg = d100c_neg, rechts_measure = d100c, rr1 =
knutwil_concurrent2_100, rr2 = knutwil_concurrent10_100, 2) #done

#####
#up
o <- function(normal_kde, oben_pos, oben_neg, oben_measure, ro1, ro2,
rlabel ){
  as.data.frame(normal_kde, xy=TRUE) %>%drop_na() %>% mutate(oben=
print(paste0("pos: ",oben_pos," | ","neg: ", oben_neg, "\n",
print(measure(oben_measure, ro1, ro2))), rechts = print(paste0(rlabel)))
%>%
ggplot() + geom_raster(aes(x=x, y=y, fill=layer)) +
  scale_fill_gradientn(colours = hcl.colors(10, "YlGnBu"), breaks = c(10,5,
-0.001), trans = "reverse", limits =c(10,-0.001), na.value="navyblue")+
  theme(legend.position = "none", axis.title.x=element_blank(),
        axis.text.x=element_blank(),axis.ticks.x=element_blank(),
axis.title.y=element_blank(), axis.text.y=element_blank(),
        axis.ticks.y=element_blank())+
  facet_grid(rechts~.~oben)
}

o8dots <- o(normal_kde = knutwil_concurrent10_250, oben_pos = d10a_pos,
oben_neg = d10a_neg, oben_measure = d10a, ro1 =
knutwil_concurrent10_100, ro2 = knutwil_concurrent10_250, 250)
o12dots <- o(normal_kde = knutwil_concurrent10_500, oben_pos = d10b_pos,
oben_neg = d10b_neg, oben_measure = d10b, ro1 =
knutwil_concurrent10_250, ro2 = knutwil_concurrent10_500, 500)
o16dots <- o(normal_kde = knutwil_concurrent10_1000, oben_pos = d10c_pos,
oben_neg = d10c_neg, oben_measure = d10c, ro1 =
knutwil_concurrent10_500, ro2 = knutwil_concurrent10_1000, 1000)

#####
#nothing
#up
n <- function(normal_kde, olabel, rlabel){
  as.data.frame(normal_kde, xy=TRUE) %>%drop_na() %>% mutate(oben =
print(paste0(olabel)), rechts = print(paste0(rlabel))) %>%
ggplot() + geom_raster(aes(x=x, y=y, fill=layer)) +
  scale_fill_gradientn(colours = hcl.colors(10, "YlGnBu"), breaks = c(10,5,
-0.001), trans = "reverse", limits =c(10,-0.001), na.value="navyblue")+
  theme(legend.position = "none", axis.title.x=element_blank(),
        axis.text.x=element_blank(),axis.ticks.x=element_blank(),
axis.title.y=element_blank(), axis.text.y=element_blank(),
        axis.ticks.y=element_blank())+

```

```
    facet_grid(rechts~.~oben)
  }

n4dots <- n(normal_kde = knutwil_concurrent10_100, 10, 100)
...
##cowplot plotting
```{r}
plot_grid(r1dots, r2dots, r3dots, n4dots, b5dots, b6dots, b7dots, o8dots,
b9dots, b10dots, b11dots, o12dots, b13dots, b14dots, b15dots, o16dots)
...

```

## 11.2 Declaration of authorship

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.

Maxim Lehmann