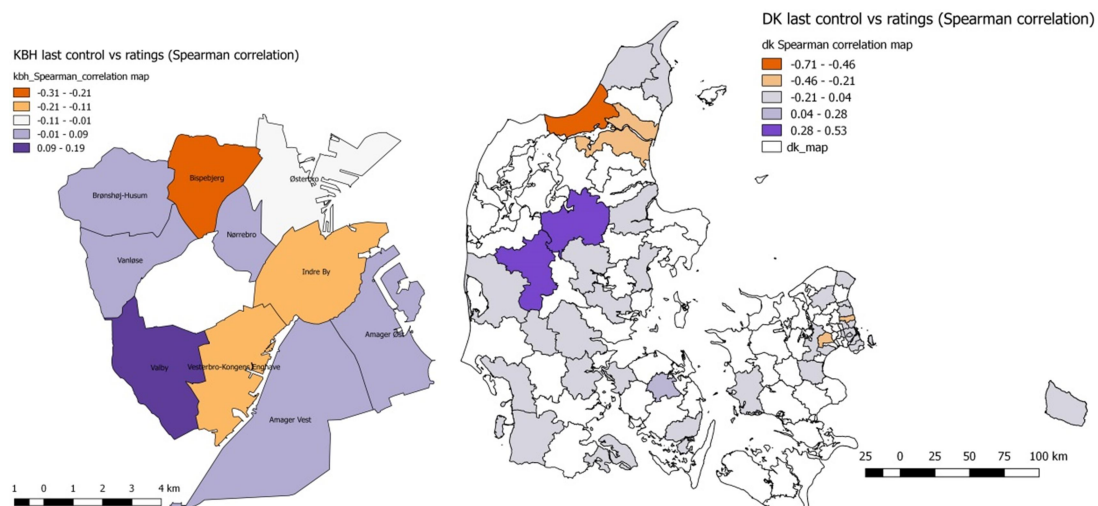


Spatial Analyses of Food Safety, Customer Ratings and Price of Restaurants in Denmark



Student:

Oana Baescu

This page intentionally left blank

Aalborg University - Copenhagen Campus

The School of Architecture, Design and Planning

Semester: 4th, 2017

A.C.Meyers Vænge 15, M2

Secretary: Janni Rise Larsen

Project Title:

Spatial analyses of food safety, customer ratings and price of restaurants in Denmark

Theme:

Master's thesis project

Project period:

1st February- 9th June 2017

Submission date:

9th June 2017

Supervisor:

Associate Professor Carsten Kessler

Attendees:



Oana Baescu (20152813)

This project was built around the idea of determining whether there is any connection between Denmark's restaurants state of cleanliness, their price range and the ratings they receive from the customers, all displayed in a spatial context.

To serve this purpose, 1492 Danish restaurants, of which 778 located in Copenhagen area, were analysed.

The process included data validation for the datasets obtained from the smiley database, data extraction from Foursquare platform using Python, data analysis and visualisation using ArcMap, QGIS and R.

The project had two foci, one regarding restaurants in Denmark and the other one regarding restaurants located in Copenhagen area.

Number of copies: 1

Number of pages: 92

Number of appendix: 3

This page intentionally left blank

Table of Contents

Acknowledgment	1
1. Introduction.....	1
1.1. Problem statement.....	3
1.2. Research questions	3
2. Background.....	3
2.1. Restaurant ratings	4
2.2. Restaurant state of cleanliness	4
3. Theory and Methodology.....	6
3.1. Restaurant Food Safety	6
3.1.1. Danish Food Administration (Fødevarestyrelsen).....	6
3.1.2. Inspection Control of Food Businesses – Restaurants	7
3.2. Smiley Scheme.....	10
3.2.1. Smiley Symbols	11
3.2.2. Elite Smiley	12
3.3. Restaurant – the industry code	13
3.4. Ratings online platform – Foursquare.....	14
3.4.1. Foursquare API and the URL request design.....	15
3.4.2. Foursquare parameters.....	16
3.5. Spatial analysis	19
3.6. Software	28
3.6.1. ArcMap	28
3.6.2. QGIS.....	29
3.7. Programming Languages	29
3.7.1. Python	29
3.7.2. R.....	30
4. Implementation Stage.....	32
4.1. Data processing	32
4.1.1. Raw data preparation: smiley data base.....	33
4.1.2. Foursquare platform data request in Python.....	38
4.1.3. Data processing in ArcMap and QGIS.....	47
4.2. Data analysis and visualisation.....	50

5. Results	57
5.1 Data distribution.....	57
5.2. Data analysis.....	63
5.3. Data visualisation	67
6. Discussion	72
6.1. Answers to the research questions	72
6.2. Limitations.....	75
6.3. Further Research and Improvements	76
7. Conclusion	78
8. Bibliography.....	79
9. Appendix.....	86

List of Figures:

Figure 1 Hygiene ratings: an interactive map of UK food outlets (Hygiene ratings: an interactive map of UK food outlets, www.theguardian.com , 2017)	5
Figure 2. The points of the inspection control report (Kontrolrapporter på engelsk og tysk, www.findsmiley.dk , 2017).....	9
Figure 3.Example of a control report at a restaurant that received a '2' at one of the inspected points. This determined the general 'smiley' of the inspection result (Islands Brygge Wok control rapport from 02.03.2016, www.findsmiley.dk , 2017).....	10
Figure 4. The explanation of the smiley scheme as presented on inspection report. (Kontrolrapporter på engelsk og tysk , www.findsmiley.dk , 2017)	12
Figure 5. Other symbols from the smiley scheme as presented on www.findsmiley.dk (Symboler og ikoner , www.findsmiley.dk , 2017).....	12
Figure 6 The elite smiley symbol (Kontrolrapporter på engelsk og tysk , www.findsmiley.dk , 2017) .	13
Figure 7.Food categories and subcategories on foursquare.com (Foursquare Category Hierarchy, developer.foursquare.com , 2017)	18
Figure 8 A restaurant's page on Foursquare and three of the parameters present in the response file highlighted in yellow: price, categories, ratings (Amass, foursquare.com , 2017)	19
Figure 9 Ripley's K-function plot (Comber, L. & Brunsdon, C., 2015, pp. 189)	26
Figure 10 Voronoi diagram for eight sites in the plane (Aurenhammer, F., 1991, pp. 347)	27
Figure 11 Implementation's stage flow chart	32
Figure 12. Filtering the database in excel after the industry code '56.10.00 A' corresponding to restaurants	33
Figure 13 'Custom Auto Filter' tool from excel used for the queries on 'name' dataset' column.....	35
Figure 14. 'Find and Replace' excel' tool used to remove characters or set of characters associated with the restaurants' names.	36

Figure 15. GPS coordinates of Restaurant Uformel, located on Studiestræde 69, 1455, København K (www.gps-coordinates.net 2017).....	37
Figure 16 Restaurants from different cities with the same longitude and latitude values	38
Figure 17. Creating a new app on Foursquare (New Developer App, foursquare.com , 2017)	39
Figure 18. User credentials from Foursquare, (App Details, foursquare.com, 2017)	39
Figure 19 The join of the two csv file: the one containing the smiley information and the one containing the response from Foursquare.....	47
Figure 20 Adding x and y to the joined csv file.....	48
Figure 21 Join between 'dk_ratings' and 'dk_kort' files.....	49
Figure 22. Saving the shapefile with the UTM 32N CRS.....	50
Figure 23 Denmark's restaurants distribution	51
Figure 24 Copenhagen's restaurants distribution.....	51
Figure 25 Voronoi diagram input form in QGIS for Copenhagen.....	53
Figure 26 Settings for the choropleth map representing the Spearman correlation between 'last control' and 'ratings' for Denmark.....	56
Figure 27 Settings for the choropleth map representing the mean values for the 'price' variable from Copenhagen	56
Figure 28 Kernel density estimation for Denmark (the left map) and Copenhagen (the right map) ...	57
Figure 29 K-function plot for Denmark	58
Figure 30 K-function plot for Copenhagen area.....	58
Figure 31 Voronoi diagram for Denmark, with elite smiley variable	59
Figure 32 Voronoi diagram of Denmark with last control variable.....	59
Figure 33 Voronoi diagram of Denmark with price variable.....	60
Figure 34 Voronoi diagram of Denmark with ratings variable.....	60
Figure 35 Voronoi diagram of Copenhagen with last control variable	61
Figure 36 Voronoi diagram of Copenhagen with ratings variable	61
Figure 37 Voronoi diagram of Copenhagen with price variable	62
Figure 38 Voronoi diagram of Copenhagen with elite smiley variable	62
Figure 39 Denmark, Spearman's correlation for last control versus ratings	68
Figure 40 Denmark, Spearman's correlation for last control versus price.....	68
Figure 41 Denmark, Spearman's correlation for elite smiley versus ratings	68
Figure 42 Denmark, mean value for last control (left) and the mean value for ratings (right)	69
Figure 43 Denmark, mean value for price (left) and the mean value for elite smiley (right)	69
Figure 44 Copenhagen, Spearman's correlation for last control versus ratings (left) and for last control versus price (right)	70
Figure 45 Copenhagen, Spearman's correlation for elite smiley versus ratings.....	70
Figure 46 Copenhagen, mean value for last control (left) and the mean value for ratings (right).....	71
Figure 47 Copenhagen, mean value for price (left) and the mean value for elite smiley (right).....	71

List of Tables:

Table 1. Basic control frequency based on the enterprise risk group (Risikobaseret kontrol, www.findsmiley.dk , 2017).....	8
Table 2. New industry codes established in 2007 for the restaurant industry (DANSK BRANCHE-NOMENKLATUR - DANSK BRANCHEKODE , www.dst.dk , 2017).....	14
Table 3. Foursquare API Request and response parameters used in the present project (Search Venues , developer.foursquare.com , Versioning & Internationalization , developer.foursquare.com , Venue Response, developer.foursquare.com , 2017).....	16
Table 4 Correlation coefficient values and the correspondent correlation type (Mazzocchi, M., 2008, pp. 174)	21
Table 5 Correlation coefficient values corresponding to the strength of the relationship (Xiao, C., et al. 2015, pp. 3868).....	21
Table 6. The columns from the smiley database for the current project	34
Table 7 Data overview.....	50
Table 8 Spearman correlation values for Copenhagen districts	63
Table 9 Spearman correlation values for Danish municipalities	65
Table 10 Mean values for Copenhagen districts, calculated for four variables	66
Table 11 Mean values for Denmark's municipalities, calculated for four variables	67

List of Equations:

Equation 1 Covariance formula (Mazzocchi, M., 2008, pp. 173)	20
Equation 2 Correlation coefficient formula (Mazzocchi, M., 2008, pp. 174).....	20
Equation 3 Spearman rank correlation coefficient formula (Xiao, C., et al. 2015, pp. 3869)	22
Equation 4 Mean calculation formula (Welkowitz, J., et al., 2011, pp.58)	22
Equation 5 Kernel density estimation formula (Anderson, T. K., 2008, pp. 360).....	23
Equation 6 Bandwidth formula for Kernel density estimation in R (Comber, L. & Brunsdon, C., 2015, pp. 176)	24
Equation 7 K-function formula (Comber, L. & Brunsdon, C., 2015, pp. 185)	24
Equation 8 K-function formula for complete spatial randomness (Comber, L. & Brunsdon, C., 2015, pp. 185)	25
Equation 9 Formula of estimated K- function (Comber, L. & Brunsdon, C., 2015, pp. 186)	25
Equation 10 Formula for the estimate of intensity (Comber, L. & Brunsdon, C., 2015, pp. 186).....	25
Equation 11 Formula for the edge-corrected estimated K-function (Comber, L. & Brunsdon, C., 2015, pp. 187,189)	26
Equation 12 Voronoi region formula (Aurenhammer, F., et al. 2013, pp. 8).....	27

Acronyms:

API – Application Programming Interface

CRS – Coordinate Reference System

UTM – Universal Transverse Mercator

Acknowledgment

I would like to thank to Carsten Keßler, my thesis supervisor, for his tremendous help, dedication and support throughout the whole project.

I would also like to thank to Troels Schwarz-Linnet, volunteer at Codher, for his patience, consistency and professionalism.

1. Introduction

Eating out is becoming a popular activity nowadays. The reasons for dining out are various: pleasure, special occasions or work associated reasons. The popularity of eating out of the house increased also the number of food services establishments in the recent years.

(Edwards, J.S.A, 2012, pp. 223) When talking about the type of food one can choose when eating out, it was remarked that food service industry is keeping up with the food movement changes. From eating locally, seasonal or organic to molecular gastronomy or fusion cuisine, there is a large variety of food styles, the consumer can select from. There are a large number of restaurants focused only on making profit too and on selling unsophisticated food that is not following any food trends. (Edwards, J.S.A, 2012, pp. 225)

Dining out in a restaurant is not limited to putting together the food and the customer but to an entire algorithm that combines “the food, the consumer and the situation under which consumption takes place” (Edwards, J.S.A, 2012, pp. 225).

The dining out experience is expressed online, and the ratings and reviews are ways used to characterize it. Online environment hosts websites that evaluate products or services by summing ratings. This helps customers to have a fast grasp of the product or service without spending much time with reading reviews. (Hu et al., 2016, pp. 133)

Food industry services are no different. “Online, every restaurant has a chance of being reviewed” (Davis, M. 2009, pp. 223) or rated.

When talking about restaurants' state of cleanliness and public access to this type of information, a study published in 2003 by Jin, Ginger Zhe and Leslie, Phillip describes the effect of introducing hygiene grade cards on restaurants and consumers in 1998 in Los Angeles. The main conclusion of the study was that introducing the hygiene grade cards raised awareness and restaurants improved the quality of the food hygiene. This was demonstrate by showing how the grade cards increased in inspection scores, made consumer more observant about restaurant hygiene and decrease the number of illnesses caused by spoiled food. (Jin, G. Z. & Leslie, Ph., 2003, pp. 449)

Disclosure of restaurants' state of cleanliness helped the consumer to avoid the places with poor hygiene. This information disclosure had an impact on the business too, restaurants with great hygiene being frequented by more customers than the restaurants that scored a poorer grade. (Jin, G. Z. & Leslie, Ph., 2003, pp. 412)

In Denmark, for example, all the places that serve food are obliged to display, in a visible place, closer to the entrance, the latest food safety control report. (About the Danish Smiley Scheme, www.findsmiley.dk , 2017) In this way, the potential customers are notified about the cleanliness of the place and they can decide if the place worth the visit or not. The food safety report from the last inspection can be found online too, on the Danish Food Inspection Administration website or, in some cases, on the restaurants' website.

During the project documentation, it was not encountered information regarding restaurant ratings or their state of cleanliness in a spatial context. The spatial context was found, however, in connection to crime or car accidents data analyse. Because there is no connection between them and the project research, it was decided the spatial context will be treated in the theory and methodology chapter.

In connection to all the arguments presented above, the project goal is to find out if there is any relation between restaurants state of cleanliness, online ratings and prices and how to present this connection in a spatial context. To serve this purpose, 1492 Danish restaurants, of which 778 located in Copenhagen area, were analysed. The process included data validation for the datasets obtained from the smiley database, data extraction from Foursquare platform using Python, data analysis and visualisation using ArcMap, QGIS and R. The project has two foci, one regarding restaurants in Denmark and the other one regarding

restaurants located in Copenhagen area. Based on this were formulated the research questions listed in subchapter 1.2.

1.1. Problem statement

How food safety, customer ratings and price of restaurants in Denmark are related and how this relationship can be presented in a spatial context?

1.2. Research questions

1. Given Denmark's administrative division, what are the municipalities holding the cleaner restaurants that are also best rated by the customers?

This question will be answered using R to calculate the correlation between significant variables and the mean for every significant variable of Danish restaurants, for each municipality. Choropleth maps for both types of values will present the results in a spatial context.

2. What is the relation between the income per municipality and the restaurant cleanliness, the price and the customer ratings?

To answer this question data analyses made in R and choropleth maps produced in QGIS will be used. Also the answer will include information about the restaurants distribution in Denmark and correlation between significant variables.

3. How Copenhagen is represented in the relation between food safety, price and customer reviews in each of its districts?

This question will be answered first from the data distribution point of view. Then it will be calculated Spearman's correlation coefficient and mean. The results will be presented in a spatial context too, using choropleth maps.

2. Background

This chapter will introduce general information on how restaurant ratings are calculated on three ratings websites: TripAdvisor, Zagat and Yelp. Also the food safety system with public impact is described from the United Kingdom and the State of New York.

2.1. Restaurant ratings

A general definition of what a rating is in the online context of customer-product relationship is presented in an article about the connection between online ratings and reviews and the products sales. In this article Hu, N. et al. state that “numeric ratings can be viewed as codified assessments on a standardized scale”(Hu, N. et al. from 2013, pp.42). In the same article they remind the abundance of information the consumer is facing and how the ratings and price information increase the visibility of a product and decrease the time spent online while reading reviews increases the time spent online and could positively affect the choice and the quality of the product. (Hu, N. et al. from 2013, pp.43)

Ratings and restaurant ratings in particular uses special calculation algorithms and almost every rating website have its own.

TripAdvisor, for example, uses values from 1 to 5, 5 being the best and, in case of restaurants, ratings summary consist of four predefined aspects: food, service, value and atmosphere. (What is a rating?, www.tripadvisorsupport.com, 2017, Restaurant Details, www.tripadvisor.com , 2017)

For the rating system, Zagat uses also values from 1 to 5, but decimals are taken into account too. Ratings summary has three predefined aspects: food, décor and service. (Understand Zagat ratings, support.google.com, 2017)

Yelp has a rating system based on values from 1 to 5 but the algorithm behind the ratings generation involves the use of recommended reviews. (Do reviews that aren't currently recommended impact the business's star rating?, www.yelp-support.com , 2017)

2.2. Restaurant state of cleanliness

Public disclosure about restaurants state of cleanliness can be acknowledge as a new fact if the following are considered: in the state of New York, it started in July 2010, in Great Britain in 2011 and in Denmark in 2001. (How We Score and Grade, www1.nyc.gov , 2017, UK-wide Food Hygiene Ratings Steering Group – Minutes for 10 March 2011 meeting, www.food.gov.uk, 2017 Om smiley-ordningen, www.findsmiley.dk , 2017)

For each of the above mentioned states, data of food safety control report can be downloaded for free.

For the state of New York, food safety scores are divided into three categories, labelled with A, with points from 1 to 13, B with points from 14 to 27 and C, with 28 points or more. The restaurant that has the lower scores has the better hygiene. (How We Score and Grade, www1.nyc.gov , 2017)

The food hygiene grades in Great Britain is based on ratings from 0 to 5, 5 being the grade for “very good” hygiene standards. (Find out more about food hygiene ratings www.food.gov.uk , 2017). In 2013, The Guardian displayed on an interactive map, presented below, the restaurants with the food hygiene grades. At the time the map webpage was consulted, it was noticed that the information was not updated to present year.

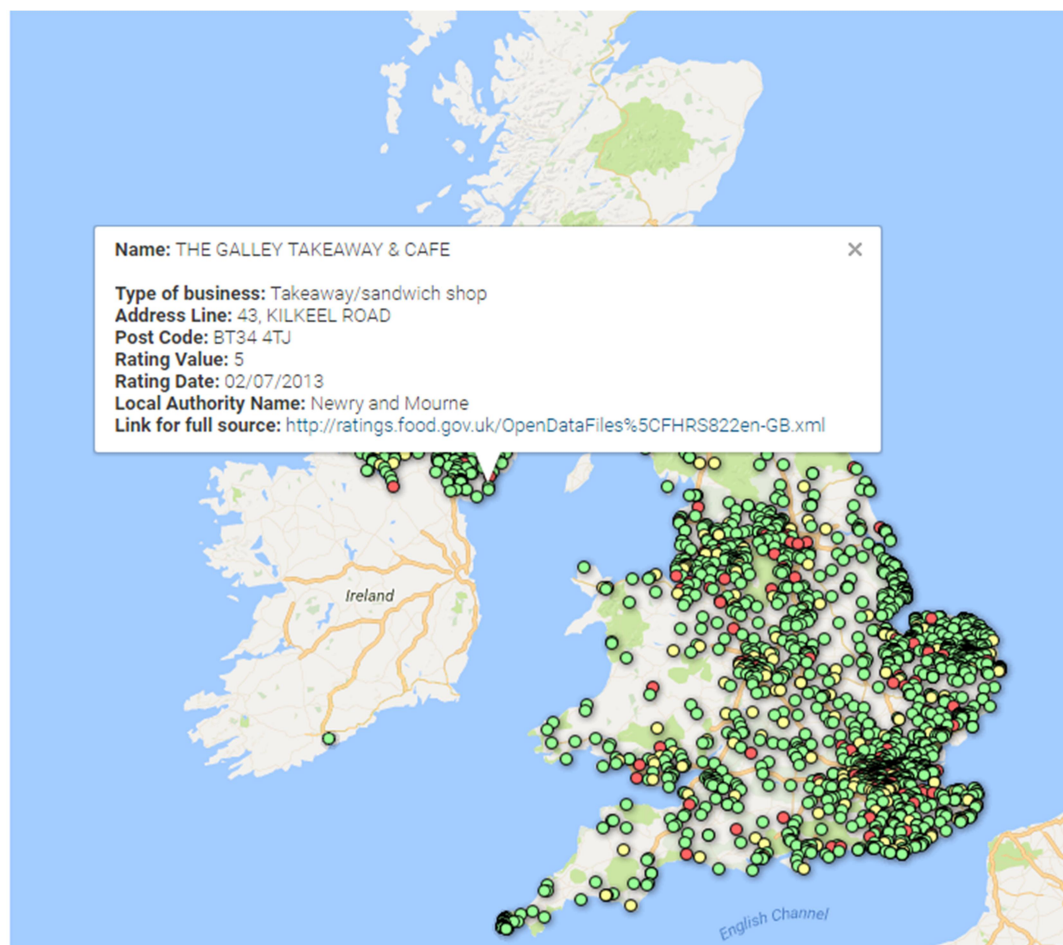


Figure 1 Hygiene ratings: an interactive map of UK food outlets (Hygiene ratings: an interactive map of UK food outlets, www.theguardian.com, 2017)

3. Theory and Methodology

This chapter introduces the tools that were used to fulfil the project objectives. It is described the theory behind it, from general aspects of food safety in Denmark and Foursquare ratings platform to software and programming languages used to process, analyse and visualize the data and the results.

3.1. Restaurant Food Safety

This subchapter introduces general information about food safety in Denmark. It describes the role of Danish Food Administration in the food control inspection and how this inspection is done. It is also presented what smiley system is and how it works.

3.1.1. Danish Food Administration (Fødevarestyrelsen)

Danish Food Administration (Fødevarestyrelsen) is part of the Danish Ministry of Environment and Food (Miljø- og Fødevareministeriet) and takes care of the rules governing the veterinary and food industry. It also makes sure the rules are respected via guidance and monitoring of food products, animal nutrition, slaughterhouses and veterinary conditions (Fødevarestyrelsen, www.foedevarestyrelsen.dk, 2017).

The responsibilities inside Danish Food Administration are divided among five departments. Three of these departments (veterinary, food products and meat inspection) are specialized business lines whilst the other two (customer and development and finance - which also includes the laboratory part) are for business support (Organisation, www.foedevarestyrelsen.dk, 2017).

- Veterinary department is responsible for the control and emergency situations. It manages also the rules formulation and the policy development in relation to animal health, animal welfare and veterinary medicine.

- Food products department is responsible for the control and emergency situations. It manages also the rules formulation and the policy development in relation to animal nutrition and food chemistry and food quality.
- Meat control department is responsible for the inspection of all slaughtered animals before and after the slaughter. It also supervises the tasks related to small abattoirs;
- Customer and development department is responsible for business development, planning control, communication and quality related to customer consultancy, international coordination, international trade, and law and control management.
- Finance department is responsible for finance, human resources, IT, laboratory and business.

(Organisation, www.foedevarestyrelsen.dk, 2017)

3.1.2. Inspection Control of Food Businesses – Restaurants

Food business companies are responsible for their products and Danish Food Administration controls if they comply with the rules (Kontrol, www.foedevarestyrelsen.dk, 2017).

The Danish Food Low (from the 8th of March 2013, with the amendments from § 9 of Law No. 1459 from 17th of December 2013) is the one that ensures the consumers are provided with healthy and quality foods, that they are protected from misleading in matters regarding food and promotes healthy eating habits by controlling and supporting the production and marketing of food at all stages (Bekendtgørelse af lov om fødevarer, www.retsinformation.dk, 2017). The general rules for publication of the inspection control results are presented in the § 56 of the Danish Food Low (Regler, www.findsmiley.dk, 2017).

All the food companies are inspected regularly and verified how well they comply with the food regulations. The inspections are unannounced and their frequency can vary “from once every two years to one to three times a year” (About the Danish Smiley Scheme, www.findsmiley.dk, 2017) depending on factors like the risk group the food company belongs to, if the company has or not the elite status award or if the report control has remarks with additional checks. The control can be also requested by the food businesses

and in this case it is done for a fee. The companies that have an elite smiley award will get fewer controls (Risikobaseret kontrol, www.findsmiley.dk , 2017).

Additionally to the basic inspection controls, there are a large number of campaign controls and sample checks carried out. Also, the consumer complaints are often followed by a control at the food enterprise in question (Opfølgende og bestilte controller, www.findsmiley.dk , 2017).

Example of a basic control frequency for retail food companies based on the risk group is presented in

Table 1.

Risk group	Industry	Standard frequency	Elite frequency
High	Restaurant, slaughter, fish shop	2	1
Medium	Bakery	1	0.5
Low	Grocery store	0.5	0.5
Ultra low	Pharmacy, drugstore, pub	As required	X

Table 1. Basic control frequency based on the enterprise risk group (Risikobaseret kontrol, www.findsmiley.dk , 2017).

The results and the details of the inspection are filled by the inspector in a control report form ('kontrolrapport' in Danish). This document is displayed by the food businesses so the consumer can see the status of the cleanliness place before deciding "to enter a shop or a restaurant" (About the Danish Smiley Scheme, www.findsmiley.dk , 2017). The control report is also present on the food businesses web-site.

Using the 'search' window of www.findsmiley.dk one can find the control report of a specific food business using key words like 'address', 'zip code', 'city', 'company name' (Velkommen til findsmiley.dk , www.findsmiley.dk , 2017).

The smiley data base, that holds data of all the companies across Denmark that has been controlled by the Danish Food Administration inspectors, can be found at www.findsmiley.dk/Statistik/Smiley_data/Sider/default.aspx. The data are public and can be downloaded in two formats type (xls and xml) (Hent smiley-data, www.findsmiley.dk , 2017).

The control report has general control points, as seen in Figure 2. However, not all the points are verified at once during a food safety inspection control. The results of the control are written on the right side of each inspected point and they are numbers from 1 to 4. What each number represents and how they are associated with the smiley symbols it is presented in the next sub-chapter.

Inspected	Result
Hygiene: Handling of food	
Cleaning	
Maintenance	
Own-check system	
Publication of Inspection Report	
Hygiene education	
Labelling and information	
Approvals etc.	
Special labelling and ID requirements	
Food standards	
Additives etc.	
Chemical contaminants	
Packaging etc.	
Other	
Not all rules are inspected each time	

Figure 2. The points of the inspection control report (Kontrolrapporter på engelsk og tysk, www.findsmiley.dk , 2017)

The result of the report is determined by the worst value of the inspected point and it is displayed on the top-left side report control as seen in the figure below (Islands Brygge Wok control rapport from 02.03.2016, <http://www.findsmiley.dk> , 2017).



Fødevarestyrelsen

Kontrolrapport

Virksomhed **Islands brygge wok**

Adresse **Leifsgade 13**

Postnr./By **2300 København S**

CVR-nr. **36549106**

Denne kontrol, dato
02-03-2016

Tidligere kontrol
Dato **04-09-2015**

Kontrolleret	Resultat
Hygiejne: Håndtering af fødevarer	1
Rengøring	1
Vedligeholdelse	
Virksomhedens egenkontrol	2
Offentliggørelse af kontrolrapport	1
Uddannelse i hygiejne	
Mærkning og information	
Godkendelser m.v.	
Særlige mærkningsordninger	
Varestandarder	

Tilsynsførendes bemærkninger side 1 af 2

Hygiejne: Håndtering af fødevarer: Følgende er kontrolleret uden anmærkninger: Opbevaring og emballering af fødevarer i div kølefaciliteter samt målte temperaturer. Faciliteter til hygiejnisk håndvask. Vejledt om procedurer for nedkøling.

Figure 3. Example of a control report at a restaurant that received a '2' at one of the inspected points. This determined the general 'smiley' of the inspection result (Islands Brygge Wok control rapport from 02.03.2016, www.findsmiley.dk, 2017)

Restaurants with inadequate results at the inspection control will have to pay for the next follow-up inspection. If the restaurants have good food safety compliance proved at several controls, they will receive the 'elite' smiley award (About the Danish Smiley Scheme, www.findsmiley.dk, 2017).

Both the consumers and the controlled food companies could write comments regarding the inspection report. The comments could be introduced using the contact form link from www.foedevarestyrelsen.dk and they could be visualized along with the control report (Kommentarer til kontrolrapporten, www.findsmiley.dk, 2017).

A blank inspection report in English is attached in the appendix section.

3.2. Smiley Scheme

In 2001 the smiley scheme was introduced in Denmark and it made it easy for the consumers to look after the retail food business that respect the food safety rules. From that time on,

Food Safety Administration's inspection reports have been published and they helped on creating more transparency of the food businesses, more competitiveness inside the industry, all these for the consumers benefit (Om smiley-ordningen , www.findsmiley.dk , 2017).

A short timeline of smiley scheme development is presented below:

- October 2001 - launching of the smiley scheme with compulsory publication of good and bad inspection control results.
- January 2004 - the smiley scheme is loaded on the internet.
- March 2008 - the elite smiley is introduced; the smiley scheme is made more visible in the way that it can be readable from the outside of the food establishment. The food businesses can link the inspection control reports to their own websites.
- August 2016 – Findsmiley.dk gets a new and interface and new functionality.

(Om smiley-ordningen, www.findsmiley.dk , 2017)

The smiley system has four smiley types and it describes the degree the food establishment complies with the food safety rules. The smiley types will be briefly described in the next subchapter.

3.2.1. Smiley Symbols

The smiley symbols are part of the smiley scheme. They are graphically represented in Figure 5 together with the explanation for each smiley figure. Also, each smiley symbol has a number associated with it. From 1 to 4, number 1 is associated with the 'happy' smiley and means the food business complies with the rules, to number 4, associated with the 'sad' smiley, representing the poorest performance in terms of food safety regulation. The smileys and their associated values are presented in Figure 4 (Kontrolrapporter på engelsk og tysk , www.findsmiley.dk , 2017).






Result		Explanation
1		No remarks
2		Enjoining order
3		Injunction or prohibitory order
4		Administrative penalties, reported to the police or approval withdrawn
Poorest result determines the smiley		
Elite		No remarks on the latest 4 reports and within the latest 12 months

Figure 4. The explanation of the smiley scheme as presented on inspection report. (Kontrolrapporter på engelsk og tysk , www.findsmiley.dk , 2017)

Other symbols that accompany the smiley scheme are presented in Figure 5 and they are displayed by the food business if its premises have not been controlled yet (the top symbol) or if it cannot get the elite-status (the bottom symbol).



Virksomheden har endnu ikke haft kontrol



Kan ikke få elite-status. Vises for ultralavrisiko-virksomheder, der får kontrol efter behov.

Figure 5. Other symbols from the smiley scheme as presented on www.findsmiley.dk (Symboler og ikoner , www.findsmiley.dk , 2017)

3.2.2. Elite Smiley

The elite smiley (Figure 6) is awarded to the restaurants that received at least four 'happy' smileys consecutively and did not get any negative remarks during the inspection control in the last twelve months.



Figure 6 The elite smiley symbol (Kontrolrapporter på engelsk og tysk , www.findsmiley.dk , 2017)

An elite smiley award is not 'better' than a 'happy' smiley. A new restaurant that has not been controlled yet for four times cannot have the elite status but its performance could be appreciated to be as good as an elite status if it is complying with the food safety regulation (Om elite-smiley, www.findsmiley.dk , 2017).

There are also some food businesses that cannot achieve the elite status if they are classified like ultra-low -risk companies. Among the companies included in this category are hostels, bodegas and wine shops (Om elite-smiley, www.findsmiley.dk , Ultralavrisiko-virksomheder , www.findsmiley.dk, 2017)

3.3. Restaurant – the industry code

Industry code or 'branchekode' (in Danish) represents the official manner of classifying the economic activities and it is used in official documents, business databases. They were originally developed at a national level and progressively adapted to an international market (Virksomhedsinformation: Branchekoder , libguides.cbs.dk , 2017).

In the smiley database contained in the file named 'SmileyStatus.xml' there are listed all the businesses related to food, no matter if they process it, transport it, sell it as it is, deposit or preserve it with their associated industry codes.

From this database was extracted only the food businesses corresponding to '56.10.00.A Servering: Restauranter mv' ('Serving : restaurants etc.') industry code. This code refers to restaurants.

The other codes two codes '56.10.00.B Servering: Restauranter, mv. – begrænset' ('Serving : restaurants etc. - limited') and '56.10.00.C Servering: Restauranter mv. - åbent op til 6 måneder om året' ('Serving : restaurants etc. – opened up to 6 months a year') were not

taken into consideration in the current project. This decision was made after considering the examples found on vejledninger.foedevarestyrelsen.dk , the section about activities in retail companies ('Aktiviteter i detailvirksomheder') for 'Restaurant, begrænset' (restaurant, limited) : grill bars, canteens and cafes with limited range, gas stations, hot dog stands (Aktiviteter i restaurant, begrænset , vejledninger.foedevarestyrelsen.dk, 2017).

The industry codes were changed in 2007 and start being used from 1st of January 2008 (DANSK BRANCHE-NOMENKLATUR - DANSK BRANCHEKODE, www.dst.dk, 2017). This change, however, is not reflected in the smiley database downloaded from the www.findsmiley.dk and therefore it will not be referred as such.

The new industry codes extracted from the 'file db07_struktur-(2016).xml' can be seen in Table 2 (DANSK BRANCHE-NOMENKLATUR - DANSK BRANCHEKODE, www.dst.dk , 2017).

Industry code	Description
56.10.10	Restauranter (Restaurants)
56.10.20	Pizzeriaer, grillbarer, isbarer mv. (Pizzerias, grill bars, ice cream shops etc)
56.21.00	Event catering
56.29.00	Anden restaurationsvirksomhed (Other restaurant businesses
56.30.00	Cafeér, værtshuse, diskoteker mv.(Cafes, hostels, discos, etc.)

Table 2. New industry codes established in 2007 for the restaurant industry (DANSK BRANCHE-NOMENKLATUR - DANSK BRANCHEKODE , www.dst.dk , 2017)

3.4. Ratings online platform – Foursquare

Yelp, TripAdvisor, Google Places and Foursquare are the online platforms with rated restaurants that were taken into consideration for obtaining the ratings and price data for the current project.

TripAdvisor specifically mentioned that their data cannot be used for academic research while Yelp was offering specific dataset to be used for academic purposes (FAQ, www.yelp.com , FAQs, developer-tripadvisor.com , 2017).

Google Places had¹ the Premium Data feature (Zagat) that offered extra content, like ratings or reviews. The feature was, however part of the paid Google Maps APIs Premium Plan (Premium Data, developers.google.com, 2017).

Foursquare had no specific terms for data use for academic purposes nor asking for a fee so it was chosen to be used for the dataset extraction.

Foursquare started in 2009 in New York and ‘uses location intelligence to build meaningful consumer experiences and business solutions’ (About Us, foursquare.com/about, Foursquare, techcrunch.com ,2017). The company has two apps built to enhance the user experience. ‘Foursquare City Guide’ app is based on trusted network recommendations when exploring new places. ‘Foursquare Swarm’ app is turning the check-ins into a game and encourages users to compete with their friends to earn badges and other types of prizes (About Us, foursquare.com/about, Foursquare Swarm, www.swarmapp.com, 2017).

Foursquare offers hosted technology and data for developers and enterprises ‘to context-smart, location-aware apps’ (About Us, foursquare.com/about, 2017).

3.4.1. Foursquare API and the URL request design

The datasets used for this project were extracting using Foursquare API Venues Platform. To retrieve the necessary data types from the online platform a URL request was build using parameters from Foursquare repository. The general elements that compose this URL for the project in case are presented as follows:

“url=<https://api.foursquare.com/v2/>” - standard url from Foursquare API

“[venues/](#)” – the resource that will be accessed for further data extraction

“[search?](#)” – general information about the resource

¹ Zagat service is no longer available from the 2nd of May 2017 (Premium Data , developers.google.com, 2017)

`"ll="+str(lat)+"", "+str(lng)+"&query="+rname.encode("utf8")+"&client_id=CLIENT_ID&client_secret=CLIENT_SECRET&v=20170201&limit=1"` – required and optional parameters, delimited by ‘&’ sign.

(Search Venues, developer.foursquare.com , API Endpoints , developer.foursquare.com 2017)

3.4.2. Foursquare parameters

The parameters used in the request url and the parameters present in the response retrieved from the online platform are briefly described in the table below.

Type	Parameter	Description
request	ll (required)	“Latitude and longitude of the user's location
request	query	A search term to be applied against venue names”
request	client_id	User’s id
request	client_secret	User’s password
request	v	“Is a date representing the API ‘version’ expected from Foursquare
request	limit	Number of results to return, up to 50
response	ratings	Numerical rating of the venue (0 through 10). Returned as part of an explore result, excluded in search results. Not all venues will have a rating.
response	price	An object containing the price tier from 1 (least pricey) - 4 (most pricey) and a message describing the price tier.
response	categories	An array, possibly empty, of categories that have been applied to this venue. One of the categories will have a field primary indicating that it is the primary category for the venue. For the complete set of categories, see venues/categories.
response	(venue)id	A unique string identifier for this venue.”

Table 3. Foursquare API Request and response parameters used in the present project (Search Venues , developer.foursquare.com , Versioning & Internationalization , developer.foursquare.com , Venue Response, developer.foursquare.com, 2017)

In the request URL, all the parameters are optional unless otherwise it is mentioned (Search Venues, developer.foursquare.com , 2017)

The ratings parameter, is “sometimes” (Venue Response, developer.foursquare.com , 2017) present in the response file. But the algorithm behind this parameter is described on Foursquare – Help Centre in the ‘Business Basics’ section. According to this, the ratings of a place are a combination of variables gathered from Foursquare datasets: likes and dislikes, negative and positive tips. Businesses owners are advised to encourage their customers to react with likes or tips to increase the business ratings. It might happen that not all the places have ratings and this could occur because of the type of the place or because the algorithm did not reach a certain data amount to redraw the ratings out of it (Place ratings, support.foursquare.com , 2017).

On Foursquare platform there are ten main venues’ categories. Each category can have subcategories and each subcategory can have more subdivisions. The categories are: ‘Arts & Entertainment’, ‘College&University’, ‘Event’, ‘Food’, ‘Nightlife Spot’, ‘Outdoors&Recreation’, ‘Professional&Other Places’, ‘Residence’, ‘Shop&Service’, ‘Travel&Transport’. The ‘Food’ category has 89 subcategories and some of the subcategories have more subdivisions in case of, for example the Asian or Indian restaurants. Figure 7 displays how the ‘Food’ category is presented on Foursquare platform (Foursquare Category Hierarchy, developer.foursquare.com , 2017).

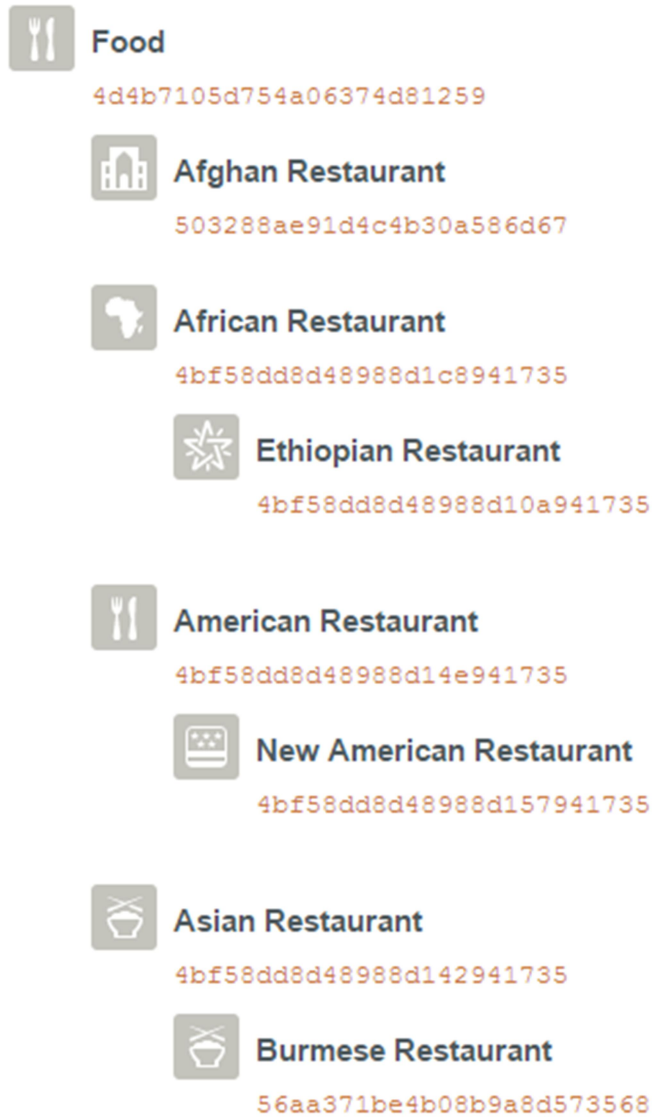


Figure 7. Food categories and subcategories on foursquare.com (Foursquare Category Hierarchy, developer.foursquare.com, 2017)

An example of how a restaurant page looks like when the place is searched on foursquare.com is showed in Figure 8. In the same figure, highlighted in yellow are some of the parameters present in the response file used in the current project.

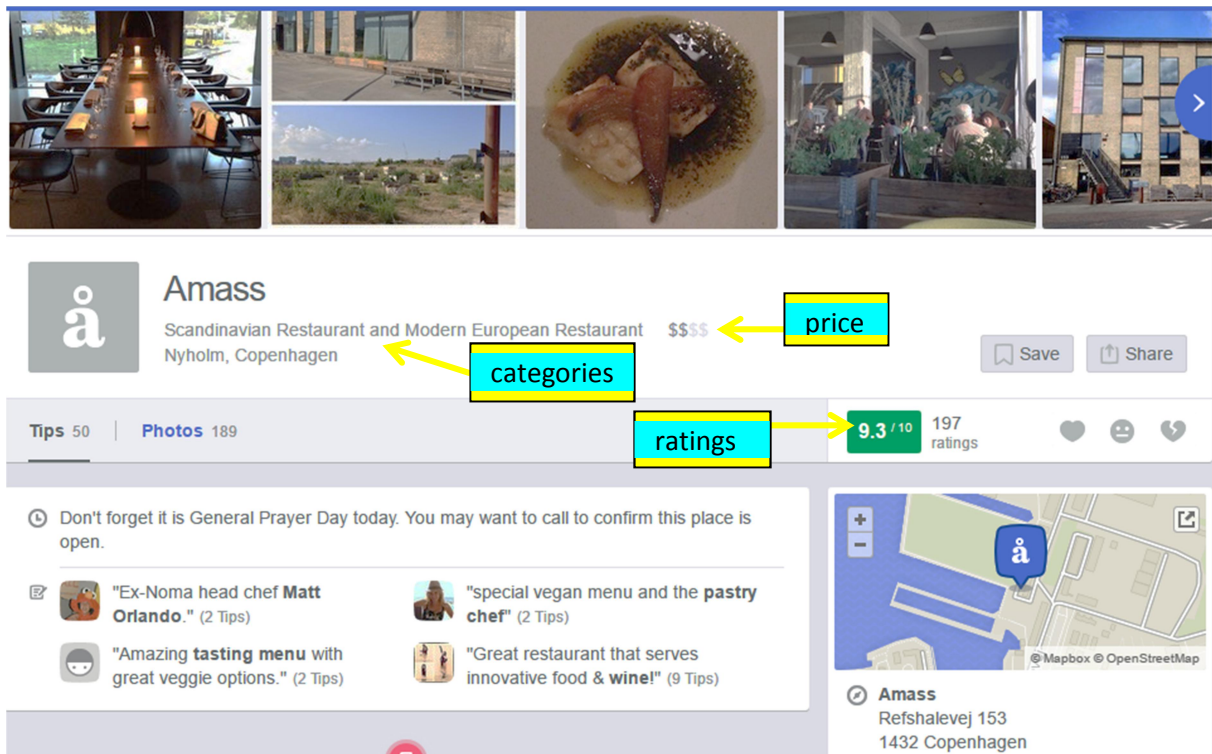


Figure 8 A restaurant's page on Foursquare and three of the parameters present in the response file highlighted in yellow: price, categories, ratings (Amass, foursquare.com, 2017)

3.5. Spatial analysis

This subchapter introduces the statistical concepts and visualizing tools used during the project implementation stage to analyse the relationships among the variables of the dataset. The subchapter is divided into two parts, the first one referring to the data analysis in a statistical environment, the second one to the spatial data distribution and data visualisation.

Data analysis

Statistics provide important tools to understand the relationships between two or more variables. Correlation interprets these relationships without presuming a causal connection between the variables. Regression, on the other hand, enters a causal aspect, by relating a dependent variable to an explanatory variable, expecting the second one to cause the first one. (Mazzocchi, M., 2008, pp. 172)

Simple (bivariate) correlation is a statistical concept that measures the relationship between two or more variables. To identify this relationship it is important to verify if the two variables vary together across the dataset. Covariance is used to measure this variation and it evaluate the 'co-movement of the two variables x and y ' (Mazzocchi, M., 2008, pp. 172-173). Covariance uses the following formula:

$$COV(x, y) = s_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{n - 1}$$

Equation 1 Covariance formula (Mazzocchi, M., 2008, pp. 173)

The result of the covariance can have both positive and negative values. If the two variables will always go in opposite direction, all the terms of the above summation will be negative thus the result will be negative too, whilst if the two variables will go in the same direction, the result will be positive. If there is no relationship among the variables, the covariance result will be close to zero. (Mazzocchi, M., 2008, pp. 173)

Covariance relies upon the measurement units and, to avoid a possible problem caused by the different measurement units of the variables, standardization is utilized. Standardization subtracts the mean and divides by the standard deviation. All this operations will produce the formula to calculate Pearson's correlation coefficient or correlation coefficient. Correlation coefficient is symbolized with r and its formula is presented below. (Mazzocchi, M., 2008, pp. 173 - 174)

$$CORR(X, Y) = r_{xy} = \frac{s_{xy}}{s_x s_y} = \frac{\frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{n - 1}}{\sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}} \sqrt{\frac{\sum_{i=1}^n (y_i - \bar{y})^2}{n - 1}}} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}}$$

Equation 2 Correlation coefficient formula (Mazzocchi, M., 2008, pp. 174)

Correlation coefficient value varies from 1 to -1 . The differences are explained in Table 4 and Table 5.

<i>r</i> value	Correlation type
$r = -1$	Perfect negative correlation
$r = 0$	No correlation
$r = 1$	Perfect positive correlation

Table 4 Correlation coefficient values and the correspondent correlation type (Mazzocchi, M., 2008, pp. 174)

<i>r</i> value	Strength of relationship
-1.0 to -0.5 or 1.0 to 0.5	Strong
-0.5 to -0.3 or 0.3 to 0.5	Moderate
-0.3 to -0.1 or 0.1 to 0.3	Weak
-0.1 to 0.1	None or very weak

Table 5 Correlation coefficient values corresponding to the strength of the relationship (Xiao, C., et al. 2015, pp. 3868)

The correlation that uses this coefficient is called Pierson correlation. To validate this correlation, together with r , another element should be considered. This is the significance of the correlation or the p -value. p -value helps on deciding if there is a confidence level in the variables relationship. For example, if $r=0.1$ (meaning a weak relationship between variables) and p -value below 0.01 then there is confidence (of 99%) that between the two variables there is a positive relationship. (Mazzocchi, M., 2008, pp. 175)

Pearson correlation coefficient is conditioned by a linear relationship between the variables (Mazzocchi, M., 2008, pp. 174).

The relationship of the current project variables was observed to be not linear.

For this reason another correlation coefficient was chosen and this was Spearman's rank correlation coefficient. This type of coefficient is "a nonparametric (distribution-free) rank statistic proposed as a measure of the strength of the association between two variables." (Hauke, J. &Kossowski, T., 2011, pp. 89) Spearman's rank correlation coefficient or r_s does not require a linear relationship between the variables and it can be considered a version of Person's correlation coefficient where the data are transformed into ranks before the coefficient is calculated. (Hauke, J. &Kossowski, T., 2011, pp. 89)

The formula to calculate Spearman's rank correlation coefficient is presented below, where $d_i = X'_i - Y'_i$ represents the difference between each pair of ranked variables and N stands for the total samples number. (Xiao, C., et al. 2015, pp. 3869)

$$r_s = 1 - \frac{6 \sum d_i^2}{N(N^2 - 1)};$$

Equation 3 Spearman rank correlation coefficient formula (Xiao, C., et al. 2015, pp. 3869)

To emphasize certain features of the datasets, statistics use *central tendency* concept. This returns a single number that is the most representative for a determined data interval. (Welkowitz, J., et al., 2011, pp. 54-55)

There are several ways of calculating the central tendency, but for the current project dataset was chosen the mean called also the arithmetic mean or average. The mean sums up all the values of a dataset and divides the sum with the number of values, as seen in the formula below. μ is the symbol for *mean*, $\sum x$ represents the sum of the values and N stands for the number of values summed up. (Welkowitz, J., et al., 2011, pp.58)

$$\mu = \frac{\sum x}{N}$$

Equation 4 Mean calculation formula (Welkowitz, J., et al., 2011, pp.58)

The mean considers all the values of the dataset, it is a consistent and it is used to calculate other statistical variables but it is sensitive to extreme values, especially when the dataset is not very large. (Welkowitz, J., et al., 2011, pp.61)

Data distribution and visualisation

To analyse the data distribution, to visualize it and to visualize the results of the data analysis the following tools were used in the current project: Kernel density estimation, K-function,

Voronoi diagram and choropleth map. These are tools used in point pattern analysis and they will be shortly detailed in the lines below.

To detect if there is a pattern in the project dataset, Kernel density estimation was used and R was the software that run the tool.

Kernel density estimation works by covering each point with a symmetrical surface. After this it is evaluated the distance from the point to a reference position that was calculated using a mathematical function. All the points are covered in this way. (Anderson, T. K., 2008, pp. 360) Then the density of a map cell is calculated by dividing the number of restaurants by the unit area (Steenberghen, T., et al., 2004, pp. 172).

Kernel density estimation is calculated after the following formula:

$$f(x, y) = \frac{1}{nh^2} \sum_{i=1}^n K \left(\frac{d_i}{h} \right)$$

Equation 5 Kernel density estimation formula (Anderson, T. K., 2008, pp. 360)

The terms of the formula represents the estimated density at the location - $f(x,y)$, the number of observations- n , kernel size or the band width – h , the kernel function – K and the distance between the point location and its observation location – d_i . The method I got its name as Kernel density estimation because of the defined bandwidth circular area (kernel) produced around each dataset point. The density estimation is then calculated by summing all the values calculated for each individual point and the result is a density estimates surface. (Anderson, T. K., 2008, pp. 360)

There are two parameters that are influencing the outcome of Kernel density estimations. These are the bandwidth and the cell size. The bandwidth size will influence the clustered areas, the larger they will be, the larger those areas will be. (Anderson, T. K., 2008, pp. 361)

As mentioned at the beginning of this section, Kernel density estimation was calculated in R and the bandwidth was not mentioned, therefor this was calculated by default after the following formula:

$$h_x = \sigma_x \left(\frac{2}{3n} \right)^{\frac{1}{6}}$$

Equation 6 Bandwidth formula for Kernel density estimation in R (Comber, L. & Brunsdon, C., 2015, pp. 176)

The formula terms are the bandwidth for coordinate x of the point – h_x , the standard deviation for the x_i coordinated of the point σ_x . To calculate the bandwidth for the y coordinates of the point, it is used the same formula, replacing x with y when necessary. (Comber, L. & Brunsdon, C., 2015, pp. 176)

Another approach to point pattern analysis was through the use K-function. K-function principle considers “that *marginal* distribution of points has a fixed intensity, but that the *joint* distribution of all points is such that individual distributions of points are not independent” (Comber, L. & Brunsdon, C., 2015, pp. 184-185). This is opposite to Kernel density estimation that assumes “the spatial distributions for a set of points are independent but have a varying intensity” (Comber, L. & Brunsdon, C., 2015, pp. 184-185).

K-function is a distance function and below its calculation formula.

$$K(d) = \lambda^{-1} E(N_d)$$

Equation 7 K-function formula (Comber, L. & Brunsdon, C., 2015, pp. 185)

The variables of this formula are as follows:

N_d - the number of events x_i for a distance d of an event randomly chosen from all the events $\{x_1, \dots, x_n\}$;

λ – the process intensity, measured in events per unit area.

(Comber, L. & Brunsdon, C., 2015, pp. 185)

For a situation where the data distribution is independent and the marginal densities are uniform, there is another function called K_{CSR} – complete spatial randomness with formula (Comber, L. & Brunsdon, C., 2015, pp. 185):

$$K_{CSR}(d) = \pi d^2$$

Equation 8 K-function formula for complete spatial randomness (Comber, L. & Brunsdon, C., 2015, pp. 185)

If d has a given value, K_{CSR} function result will give information about the expected number of events located around an event randomly chosen and the assumption is that there is a uniform density of each observation that is independently distributed of others observations. For example, if $K(d) > K_{CSR}(d)$ then there is a cluster distribution of events located at d distance. And if $K(d) < K_{CSR}(d)$ the existence of one point indicate that there are few chances that other points to be in the nearby distance. (Comber, L. & Brunsdon, C., 2015, pp. 185)

In case it is working with a sample data points of an unknown the distribution, it is made an estimate by using the sample and K-function will add the estimation factors to the formula, as seen below (Comber, L. & Brunsdon, C., 2015, pp. 185-186):

$$\hat{K}(d) = \hat{\lambda}^{-1} \sum_i \sum_{j \neq i} \frac{I(d_{ij} < d)}{n(n-1)}$$

Equation 9 Formula of estimated K- function (Comber, L. & Brunsdon, C., 2015, pp. 186)

I variable from the estimated K-function formula is a function indicator that takes values of 0 (if the logical expression in the brackets is false) or 1 (if the logical expression in the brackets is true). (Comber, L. & Brunsdon, C., 2015, pp. 186)

The formula for estimated intensity, shown below, introduces area of the study region $|A|$ that is defined by a polygon A (Comber, L. & Brunsdon, C., 2015, pp. 186).

$$\hat{\lambda} = \frac{n}{|A|}$$

Equation 10 Formula for the estimate of intensity (Comber, L. & Brunsdon, C., 2015, pp. 186)

The estimated K-function is dependent on the study area form. The theoretical formula of $K_{CSR}(d)$ assumes the points appear “in an infinite two-dimensional plane” (Comber, L. & Brunsdon, C., 2015, pp. 186). In real life, the points will come from a finite area. To overcome this situation, Ripley (Ripley, B.D., 1976, pp. 255-266) proposes adjustments to the estimate of K-function presented in the following formula:

$$\hat{K}(d) = \hat{\lambda}^{-1} \sum_i \sum_{j \neq i} \frac{2I(d_{ij} < d)}{n(n-1)w_{ij}}$$

Equation 11 Formula for the edge-corrected estimated K-function (Comber, L. & Brunsdon, C., 2015, pp. 187,189)

The new variable introduced in the equation is the area of intersection between a circle having its center at x_i and the study area A . The symbol for this variable is w_{ij} . (Comber, L. & Brunsdon, C., 2015, pp. 187)

The k-function was run in R and an example of its outcome is presented in Figure 9 where the estimated function is labelled with K_{bord} and the theoretical function (CSR) is labeled with K_{pois} . (Comber, L. & Brunsdon, C., 2015, pp. 189)

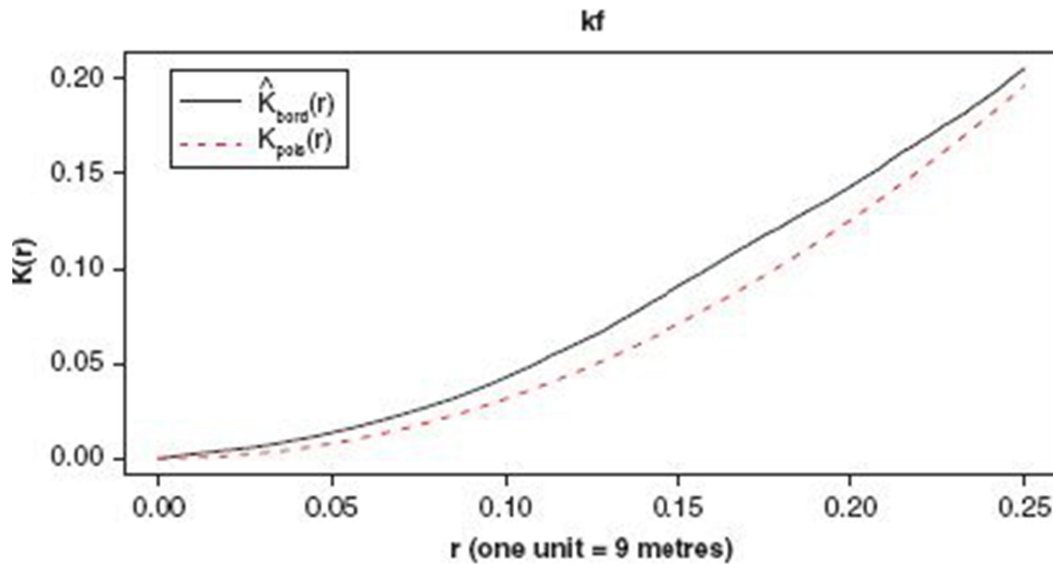


Figure 9 Ripley's K-function plot (Comber, L. & Brunsdon, C., 2015, pp. 189)

Another tool used in this project for data distribution is represented by the Voronoi diagram.

Voronoi diagram is a geometrical construction based on the nearest neighbour rule. Given a plane with a number of points in it, Voronoi diagram of the points will divide the plane following “the nearest neighbour rule: Each point is associated with a region of the plane closest to it” (Aurenhammer, F., 1991, pp. 345-346). An example of a Voronoi diagram is displayed in Figure 10.

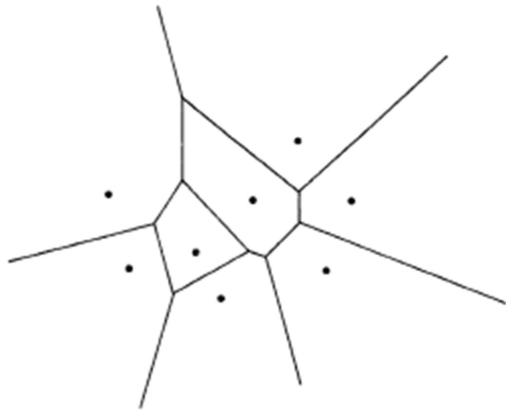


Figure 10 Voronoi diagram for eight sites in the plane (Aurenhammer, F., 1991, pp. 347)

A Voronoi diagram of S , where S is the number of sites, is defined by the union of all the Voronoi edges. A Voronoi edge is the common boundary of two Voronoi regions. Voronoi vertices are the extremity of the Voronoi edges and they can be shared between three or more Voronoi regions. (Aurenhammer, F., et al. 2013, pp. 7-8)

A formal way of producing a Voronoi region is presented below:

$$VR(p, S) = \bigcap_{q \in S, q \neq p} D(p, q).$$

Equation 12 Voronoi region formula (Aurenhammer, F., et al. 2013, pp. 8)

Where VR (Voronoi region) for p point among a given S sites is the intersection of $n-1$ halfplanes D of the two points, p and q . $VR(p, S)$ is formed by of all points x belonging to Euclidean plane R^2 . (Aurenhammer, F., et al. 2013, pp. 7-8)

When used for determine the point pattern, the density of points vary with the shape of Voronoi regions: the clustered the points, the smaller the Voronoi area will be.

(Aurenhammer, F., et al. 2013, pp. 211) Another characteristic of Voronoi diagrams when

applied to point pattern analysis is that the endpoints of the Voronoi regions have polygons of infinite area. (Comber, L. & Brunsdon, C., 2015, pp. 202)

Choropleth maps are used in the current project to visualize the data pattern and different variables associations and they were produced in QGIS.

Choropleth maps are commonly used for areal data display. Using different colours and patterns associations, sets or subsets of variables values can be presented on a map. The map legends indicate correspondences between the colour or pattern range and the datasets values range. (Waller, L. A. & Gotway, C.A., 2004, pp. 78-79)

Choropleth maps can be classed, where the value range of data is associated with colour or pattern in non-overlapping intervals of defined lengths or unclassed, where the data range is not split into intervals and all the data have the same colour or pattern. (Waller, L. A. & Gotway, C.A., 2004, pp. 80)

Choropleth maps can also have a 3D display, where different parts of the map are raised by different heights, and this, in association with colours or patterns, are indicators of different data values. (Waller, L. A. & Gotway, C.A., 2004, pp. 81)

Dot density maps are another type of choropleth maps. In these maps, the different data values are represented by a number of dots instead of a colour or a pattern. (Waller, L. A. & Gotway, C.A., 2004, pp. 81-82)

3.6. Software

Joining csv files was not possible in QGIS –‘None’ data were displayed in the joined table.

Running any other tools than ‘Join’ in ArcMap was accompanied by the ‘Not responding’ message.

These reasons determined to use both software programs to achieve the project goal.

3.6.1. ArcMap

ArcMap is the principal application of ArcGIS and it is used to execute GIS tasks with different ranges of complexity. The application allows users to visualize GIS datasets, to assign symbols to the elements displayed on the map and to produce map layouts for printing or publication purposes (What is ArcMap? desktop.arcgis.com , 2017).

In ArcMap the geographical information is represented as group of layers and map elements. Typical map elements are the north arrow, the symbol legend, the scale bar and the map title (What is ArcMap? desktop.arcgis.com , 2017).

Most common operations executed in ArcMap are: working with maps and printing them, refining and gathering GIS datasets, analysing and automatizing datasets using geoprocessing tools, using map services to publish map documents, sharing GIS data with other users, customizing the user experience with Python add-in (What is ArcMap? desktop.arcgis.com , What is a Python add-in? desktop.arcgis.com , 2017).

For this project, ArcMap was used to join two csv files and a csv file with a shapefile.

3.6.2. QGIS

QGIS is open, free software utilised for GIS data manipulation. The platforms that can run QGIS are UNIX, Mac, Linux, Windows and Android. QGIS supports a large number of vector, raster and database format and processes. (QGIS - The Leading Open Source Desktop GIS www.qgis.org/ , 2017)

In this project, QGIS was used to produce choropleth maps, to run Voronoi tool and to work with the datasets when ArcMap performance was too slow.

3.7. Programming Languages

This subchapter introduces the two programming languages used in this project: Python and R. Python was used to extract the data from the Foursquare platform and R was used to analyse them and run calculations on different variables and to produce visual outcomes.

3.7.1. Python

Python is a programming language ‘object-oriented, interpreted and interactive’ (The Python Wiki, wiki.python.org/moin/, 2017). It works with modules, classes, exceptions, dynamic data types and has a dynamic typing. (The Python Wiki, wiki.python.org/moin/, 2017)

Python is free of charge and all its releases are open source. (History and License, docs.python.org/3/license.html, 2017)

To extend Python’s usability, one can use modules. Modules are Python files with .py extension that brings new functions to the software. (Modules and Packages, www.learnpython.org , 2017)

Python modules used in this project are the following:

- ‘urllib2’ – helps opening URLs by defining specific functions and classes (urllib2, docs.python.org , 2017);
- ‘json’ – is used to work with JavaScript information (json, docs.python.org , 2017);
- ‘os.path’ –responsible for function types on pathnames (os.path, docs.python.org , 2017);
- ‘csv’ – used in working with CSV format files (csv, docs.python.org , 2017);
- ‘codecs’ – helps in defining the standard python codecs and gives access to the internal Python codec archive (codecs, docs.python.org, 2017);

3.7.2. R

R is free open source programming language used for statistical computing and to generate graphics. The language works with a large variety of statistical instruments (linear and nonlinear data modelling, runs classical tests for statistics, does time-series analysis, classification, clustering) and graphical techniques and it is very flexible. (What is R?, www.r-project.org , 2017)

R functionality can be expanded using packages and the ones used in the current project are briefly introduced below. (What is R?, www.r-project.org , 2017)

- rgdal – gives access to GDAL library (Geospatial Data Abstraction Library) and PROJ.4 library (projection and transformation operations) (rgdal: Bindings for the Geospatial Data Abstraction Library cran.r-project.org , 2017);
- spatstat – is a toolbox for Spatial Point Patterns analysis (spatstat: Spatial Point Pattern Analysis, Model-Fitting, Simulation, Tests, cran.r-project.org , 2017)
- GISTools – tools for mapping and manipulation of spatial data (GISTools: Some further GIS capabilities for R, cran.r-project.org , 2017)
- sp – spatial data access through classes and methods, plus functions for retrieving coordinates, for subsetting, for plotting data as maps; (sp: Classes and Methods for Spatial Data, cran.r-project.org , 2017)
- FNN – “cover-tree and kd-tree fast k-nearest neighbour search algorithms and related applications” (FNN: Fast Nearest Neighbor Search Algorithms and Applications, cran.r-project.org , 2017)
- maptools – tools for working with geographic data, ‘ESRI Shapefiles’ in particular (maptools: Tools for Reading and Handling Spatial Objects, cran.r-project.org , 2017)
- plyr - aggregate and disaggregate data subsets (plyr: Tools for Splitting, Applying and Combining Data cran.r-project.org , 2017)

4. Implementation Stage

This chapter will present the evolution of this project, from analysing the raw data from the smiley database to the script build for the data extraction from Foursquare platform, the processing data in ArcMap and QGIS and data analysis in R and QGIS.

Overall of this chapter is illustrated in the figure below.

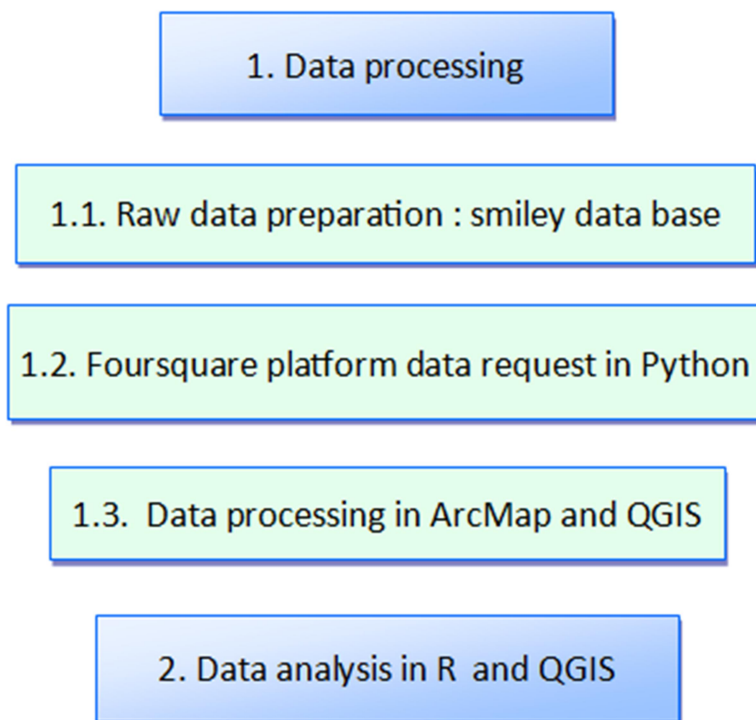


Figure 11 Implementation's stage flow chart

4.1. Data processing

The project implementation stage started with downloading smiley dataset from www.findsmiley.dk, from the section called 'Hent smiley-data'. The file was downloaded on the 1st of February 2017 in xml format.

In the following part of the project it will be presented the evolution of raw data to spatial analysis, passing through the processing stage of smiley data, data extraction using a Python

script, data set up for spatial analysis in ArcMap and QGIS, ending with analysing the data in R and QGIS.

4.1.1. Raw data preparation: smiley data base

The first step of preparing smiley data for the processing step was to select the meaningful data for the project.

The file contained data of companies from all over Denmark that interact, in different manners, with processed or unprocessed food type and therefore it was performed a selection of the data after the 'industry code' ('branchekode'). The dataset to work with had '56.10.00.A' as the industry code, corresponding to 'Servering: Restauranter mv' ('Serving: Restaurants etc.'), for the industry type. The selection of the data was done using the 'Filter' tool from excel, as seen in Figure 12.

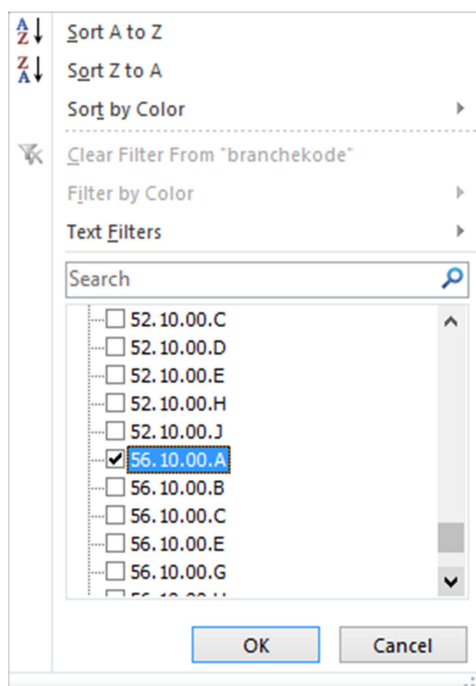


Figure 12. Filtering the database in excel after the industry code '56.10.00 A' corresponding to restaurants

After the selection, 9968 records out of 57305 were extracted for the use of the current project.

On the second step it was establish the column with unique value for each restaurant. This was done because a food company could have restaurants in more than one location (this

may be it in the same city or in different cities). The column with unique value for each restaurant was 'navnelbnr'.

The next step was to delete columns that were not relevant for the project purpose. There were deleted fourteen columns as follows: 'cvnrnr', 'pnr', 'foedevareafdeling', 'virksomhedstype', 'branchekode', 'branche', 'seneste_kontrol_dato', 'naestseneste_kontrol', 'naestseneste_kontrol_dato', 'tredjeseneste_kontrol', 'tredjeseneste_kontrol_dato', 'fjerdeseneste_kontrol', 'fjerdeseneste_kontrol_dato', 'URL'.

Then for the remaining column, column heads were translated from Danish to English, when relevant. Table 6 presents the columns from the smiley database the project work will be based on.

A	B	C	D	E	F	G	H	I
navnelb	name	adresse	zip code	city	Latest_ctrl	Elitesmiley	Geo_Lng	Geo_Lat
560167	Dino's Pizzeria	Frederiksgade 17	7870	Roslev	1	1	8.981	56.7013
533031	Saray Pizza Kebabhus	Frederiksborgvej 4	2400	København NV	1	1	12.5339	55.702
569975	Halifax Amager	Amagerbro Torv 13	2300	København S	1	0	12.5339	55.702
577636	Minigrillen	Nørregade 8	4100	Ringsted	1	0	12.5339	55.702
72958	Kaffehuset	Houmarksvej 13, Åstrup	7130	Juelsminde	1	1	12.5339	55.702
123030	Benløse Pizza	Benløseparken 2	4100	Ringsted	2	0	12.5339	55.702
561587	Agerbæk Pizza og Keb	Storegade 12	6753	Agerbæk	1	1	12.5339	55.702

Table 6. The columns from the smiley database for the current project

Below it is described what each column of the dataset represents.

- navnelb: a numeric code, unique for each restaurant;
- name – the restaurant' name;
- address – the restaurant' address;
- zip code – the restaurant' zip code;
- city – the restaurant' city;
- Latest_ctrl – the information about the last control performed by the inspectors of The Danish Veterinary and Food Administration (Fødevarestyrelsen). It refers to the restaurants that were controlled in 2016. It has a numerical value from 1 to 4 where 1 stands for the happy smiley and 4 stands for the upset smiley.
- Elitesmiley – shows if the restaurant has or not the elite status; value '0' stands for no elite smiley and value '1' stands for elite smiley;

- Geo_lng – longitude of the restaurants' location, in decimal degrees;
- Geo_lat – latitude of the restaurants' location, in decimal degrees.

Continuing with the smiley dataset processing, the data was verified for consistency and several operations were performed to insure the accuracy of the information.

The industry code '56.10.00 A' supposed to include only restaurants. However, they were performed queries on the 'name' dataset' column using 'Custom AutoFilter' tool from excel (Figure 13) with following key words: 'kantine', 'dagcenter', 'hospital', 'kursuscenter', 'idrætscenter', 'kommune', 'skole', 'institution', 'region', 'neuro', 'Udbetaling', 'idræts', 'Aramark', gas stations ('OK Plus', 'Shell', 'q8', 'statoil', 'Circle K'), '7-Eleven', 'KFC', 'eurest', 'Sunset Boulevard', 'Kirkens Korshær', 'pølsevogn', 'McDonald', universities ('SDU', 'AAU', 'DTU'), 'CELF', 'UCH', 'ISS', 'Kollegiet', 'Soldater (hjem)', 'børne (center)', 'syg', 'campus', 'rehabilitering', 'aktiveringshuset', 'pleje', 'aktivitetcenter', 'Psyk', 'Tulip TimeOut', 'ESS Offshore', 'burger king'. This set of queries was meant to remove fast food chains, gas stations and canteens associated with different institutions.

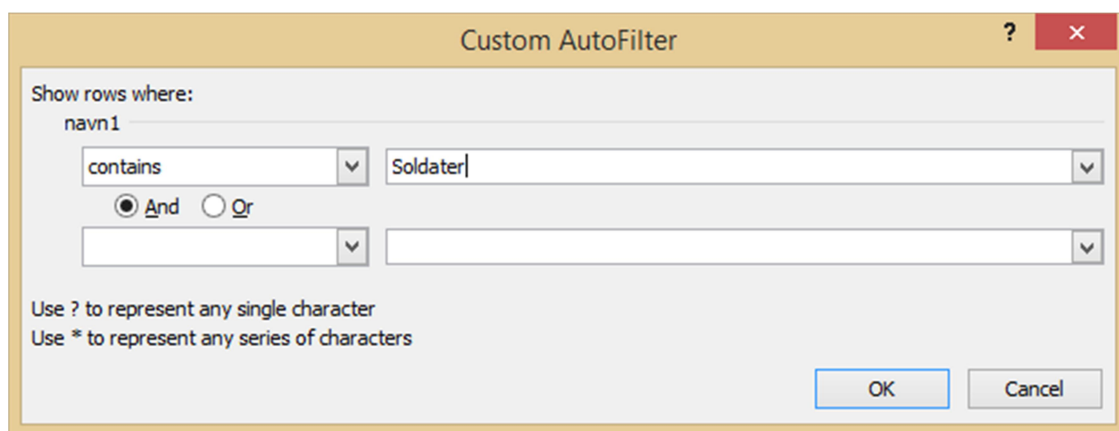


Figure 13 'Custom Auto Filter' tool from excel used for the queries on 'name' dataset' column

However, the queries were not exhaustive and more about this matter will be debated in the limitation part of the project.

At the end of the queries, the smiley dataset diminish from 9968 to 9256 data.

It was notice that part of the restaurants' names were containing characters and set of characters as follows: 'ApS', 'v/name name', 'v/ name ApS', 'A/S', 'lVS', '.dk', ',', 'v.', 'c/o', 'm/s..', 'm/f...', 'k/s', 'p/s', '. DK', 'i/s'. This could have been an issue with the script

running over the restaurant names and matching them with restaurant names from the Foursquare platform. Therefore these characters and set of characters were removed from the restaurants 'names column using 'Find and Replace' tool from excel. Figure 14 shows an example of found and replaced set of characters 'v/'. The '*' symbol is the part of the text in the cell that will be kept and the rest will be deleted/replaced with blank space.

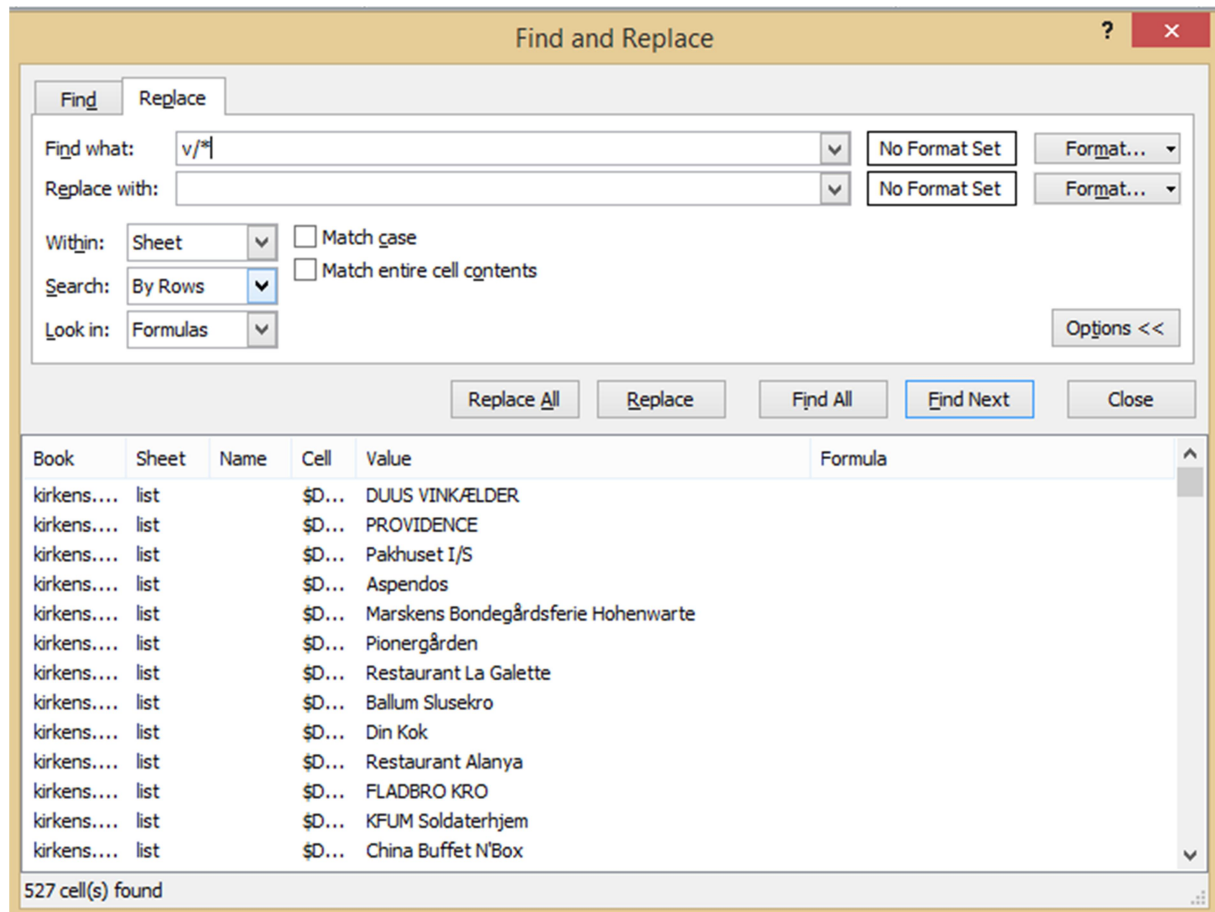


Figure 14. 'Find and Replace' excel' tool used to remove characters or set of characters associated with the restaurants' names.

The 'name' column was checked also for the name and the address not to be written in the same cell. Key words like 'torv', 'gade', 'vej', 'kælder', 'th', 'tv' were queried using the 'Find' tool from excel and the cell changes were made accordingly.

For example Restaurant Amalfi had his address written in the 'name' cell as shown below:

Restaurant Amalfi Restaurationsselskabet Gråbrødre Torv 21 A
--

The next step was to look after missing information inside the database. This was done using again the 'Filter' tool from excel. This search was applied to each column.

877 restaurants were missing coordinates. Their longitude and latitude cells were blank.

Using www.gps-coordinates.net, the gaps in the datasets were manually filled in. Figure 15 displays an example of how this task was performed: the restaurant' address was introduced in the website address window and the then, by pressing the 'Get GPS Coordinates' button, longitude and latitude were displayed. They were afterwards manually typed in the corresponding database cells.

The image shows a web interface for obtaining GPS coordinates. At the top, there is a text input field labeled 'Address' containing the text 'Stuðiestræde, 1455 København K, Denr'. Below this field is a blue button with the text 'Get GPS Coordinates'. A horizontal line separates this section from the results section below. The results section is titled 'DD (decimal degrees)*'. It contains two text input fields: one labeled 'Latitude' with the value '55.6785308' and another labeled 'Longitude' with the value '12.569309900000007'.

Figure 15. GPS coordinates of Restaurant Uformel, located on Stuðiestræde 69, 1455, København K (www.gps-coordinates.net 2017)

In the 'Elitesmiley' column the restaurants with blank cells were removed. They were also removed restaurants from the 'Latest_ctrl' column that had their last control inspection in 2014 and 2015. There were kept only the restaurants having the latest control inspection in 2016.

After these new removals, the total data to be worked with is 9245.

Going further with the data accuracy it was discovered that restaurants from different cities had the same longitude and latitude values, as seen in Figure 16. This was corrected using www.gps-coordinates.net website.

	A	B	C	D	E	F	G	H
1	navnelb	navn1	adresse1	postnr	by	Elitesmile	Geo_Lng	Geo_Lat
2	560167	Dino's Pizzaria	Frederiksgade 17	7870	Roslev	1	8.981	56.7013
3	533031	Saray Pizza Kebabhus	Frederiksborgvej 4	2400	København N	1	12.5339	55.702
4	569975	Halifax Amager	Amagerbro Torv 13	2300	København S	0	12.5339	55.702
5	577636	Minigrillen	Nørregade 8	4100	Ringsted	0	12.5339	55.702
6	72958	Kaffehuset	Houmarksvej 13, Åst	7130	Juelsminde	1	12.5339	55.702
7	123030	Benløse Pizza	Benløseparken 2	4100	Ringsted	0	12.5339	55.702
8	561587	Agerbæk Pizza og Kebabhus	Storegade 12	6753	Agerbæk	1	12.5339	55.702
9	661549	Baryl	Griffenfeldsgade 24E	2200	København N	0	12.5339	55.702
10	545962	French Chicken	Toftengårds Allé 31	2500	Valby	0	12.5339	55.702

Figure 16 Restaurants from different cities with the same longitude and latitude values

Then, as a final step in the validation of the dataset, all the cells were assured the right format, for example the cells corresponding to 'navnelb', 'zip code', 'Latest_ctrl', 'Elitesmiley', 'Geo_Lng' and 'Geo_Lat' columns will have numerical attributes.

4.1.2. Foursquare platform data request in Python

Before the project's datasets to be extracted from the Foursquare platform, it was necessary to create a new app on Foursquare (Figure 17) and to get the users credentials (Figure 18). On the user credential side, the user id is the 'CLIENT_ID' and the password is the 'CLIENT_SECRET'.

New Developer App

Get started integrating Foursquare into your app or website

bv bv

Web addresses

Application Url *

<http://www.>

Your privacy policy url

<https://www.yourapp.com/privacy>

Redirect URI(s)

https://www.yourapp.com/redirect_uri

Enter as many as you'd like, separated by commas. Like:
<https://www.foursquare.com>, <https://es.foursquare.com>, <https://fr.foursquare.com>...

* indicates required field

Figure 17. Creating a new app on Foursquare (New Developer App, [foursquare.com](https://foursquare.com/developers/new) , 2017)

bvby

Edit this App

App Information

CLIENT_ID

CLIENT_SECRET

Test this App

Get Started

Administrative information

Owner

Oana b.

Figure 18. User credentials from Foursquare, (App Details, foursquare.com, 2017)

Data about restaurant' ratings, price and category were extracted Foursquare platform using Foursquare API and a Python script. The script will read the smiley data base, line by line, and will send a URL request for each restaurant name, with a location described by the longitude and latitude columns. If the restaurant is not found on Foursquare platform, it will return no response and the script will look for the next line inside the smiley database. If the restaurant is found, a second URL request is sent to extract the desired data. By default, request and will write the response values in a json file. In the script, the response file will be transformed into a csv file, to facilitate further the data analysis.

The next pages of this project will present the code behind the data extraction and response.

The code starts with the modules import. These modules are used to support different parts of the script. A short description of each of these modules can be found in the 'Theory and Methodology' chapter, under 'Python' section.

```
import urllib2
import json
import os.path
import csv
import codecs
```

Then it indicates to Python that the code will read from an existing workbook 'restaurant_A.xlsx' (Simple usage, openpyxl.readthedocs.io, 2017)

```
from openpyxl import load_workbook
```

It is then created a json file where the response from Foursquare platform is placed. Also is created a csv file also for the response data, data that is transformed from the response json file.

```
filename = 'result_A1.json'
csvfile='csvrestA1.csv'
```

The next line verifies if the file called 'filename' exists at the indicated path and if true it will continue with the work. (Check if a file exists in a directory with Python, pythoncentral.io, 2017)

```
if not os.path.isfile(filename):
```

Then it is set the working path and the working sheet from the excel file.

```
smiley =  
load_workbook('C:/MOBILA/MAPS/GEOINFORMATICS_2016_4th/data/res  
tauranter_A.xlsx')  
sheet = smiley.get_sheet_by_name(name = 'Sheet1')
```

The following lines of the scrip are reading 'restauranter_A.xlsx' file as json file to enhance the script execution speed. This is done by making a dictionary out of the columns of the file. There are given indications on the number of rows and columns, on the encoding type for the columns having text values, and that the script should read one row of data after another. There are also instructions on how reading each column values in case they are integer, floats or text.

(Python Creating Dictionary from excel data, stackoverflow.com, 2017)

(Python Dictionary, www.tutorialspoint.com, 2017)

(Chapter 12 – Working with Excel Spreadsheets, automatetheboringstuff.com, 2017)

```
col_names = []  
for row in sheet.iter_rows(min_row=1, min_col=1,  
max_row=1, max_col=7):  
    for cell in row:  
        coln = cell.value.encode('ascii', 'ignore')  
        coll = cell.column  
        col_names.append([coln, coll])  
  
dic_keys = []  
dic = {}
```

```

    for row in sheet.iter_rows(min_row=2, min_col=1,
max_col=7):
        i = 0
        for cell in row:
            coln, coll = col_names[i]
            if (i == 0 and cell.value):
                dic_key = cell.value.encode('ascii', 'ignore')
                dic_keys.append(dic_key)
                dic[dic_key] = {}

            val = cell.value

            if coll in ['A', 'C', 'D', 'E']:
                if val == None:
                    dic[dic_key][coln] = None
                elif val== cell.value:
                    dic[dic_key][coln]= str(val.encode("utf-
8"))

                else:
                    dic[dic_key][coln] = int(val)
            elif coll in ['F', 'G']:
                dic[dic_key][coln] = float(val)
            else:
                dic[dic_key][coln] = val
            #add line by line
            i += 1

dic['col_names'] = col_names
dic['dic_keys'] = dic_keys

```

#Python dump dict to json file, stackoverflow.com , 2017

```

fileobject = open(filename, 'w')
json.dump(dic, fileobject)
fileobject.close()

```

```

else:
    fileobject = open(filename, 'r')
    dic = json.load(fileobject)
    fileobject.close()

    col_names = dic['col_names']
    dic_keys = dic['dic_keys']

    def web(lat=None, lng=None, rname=None):

        rname = urllib2.quote(rname.encode("utf-8"))
        # "search venues" request based on restaurant location
        and restaurant name; limit= Number of results to return, up to
        50.

```

The first url request is for getting the restaurants names and location (longitude and latitude). If the restaurant (venue) is not found then return 'None'.

```

    url =
    "https://api.foursquare.com/v2/venues/search?ll="+str(lat)+"",
    +str(lng)+"&query="+rname.encode("utf-
    8")+ "&client_id=CLIENT_ID&client_secret=CLIENT_SECRET&v=201702
    01&limit=1"

    response = urllib2.urlopen(url)
    jsonResponse = json.loads(response.read())
    # if the venue is not found then don't return any
    value of the four fields that are in the response
    if len(jsonResponse['response']['venues']) == 0:
        return [None, None, None, None]

```

If the restaurant (venue) is found, then look after the venue parameters: ratings, price, category, venue id and use the second url to extract them and write the response in the json response file jsonResponse2 . This explanation belongs to the code listed below.

```

venueid = jsonResponse['response']['venues'][0]["id"]

```



```

url2 =
"https://api.foursquare.com/v2/venues/"+venueid+"?client_id=
CLIENT_ID&client_secret=CLIENT_SECRET&v=20170201"
response2 = urllib2.urlopen(url2)
jsonResponse2 = json.loads(response2.read())

```

(Python Foursquare V2 library, github.com, 2017)

This part of the code presents the json response for each requested variable and it takes into considerations the errors that might appear on the way from the file reading, Foursquare platform interrogation and request. A restaurant can have several categories and the code is reading the first one. If the restaurant is not rated, then the return response will be **[None, None, None, None]**.

```

try:
    rating =
jsonResponse2['response']['venue']['rating']
except KeyError:
    rating = None
try:
    price =
jsonResponse2['response']['venue']['price']['tier']
except KeyError:
    price = None
try:
    categories =
jsonResponse2['response']['venue']['categories'][0]['id']
except KeyError:
    categories = None
except IndexError:
    categories = None
if rating == None:
    return [None, None, None, None]
else:
    return [venueid, rating, price, categories]

```

In this part the script runs through the lines of the file and send request for each line at time. Because of the limit request of Foursquare, the range was a multiple of 400 (lines) and it was run until the whole file was read. This limit was established after several trails and server error handling.

```
for i in range(9245):
    dic_key = dic_keys[i]
    dd = dic[dic_key]
    dd_keys = dd.keys()
```

If the request returns values (Webreturn), then all of the desired variables are written in the response file, in csv format. The response file was introduced at the beginning of the code.

```
if 'web' not in dd_keys:

    #if True:
        Geo_Lng = dd['Geo_Lng']
        Geo_Lat = dd['Geo_Lat']
        navn1 = dd['navn1']
        webreturn = web(lat=Geo_Lat, lng=Geo_Lng,
rname=navn1)
        dic[dic_key]['web'] = webreturn
        venueid, rating, price, categories = webreturn
        dic[dic_key]['venueid'] = venueid
        dic[dic_key]['rating'] = rating
        dic[dic_key]['price'] = price
        dic[dic_key]['categories'] = categories
```

(Storing a JSON response from an API to a CSV file stackoverflow.com, 2017)

The response file in csv format, introduced at the beginning of the code, extracts the following variables that will be used for further data analysis:

- navn1 – the restaurant name;

- Geo_Lng - longitude of the restaurants' location, in decimal degrees;
- Geo_Lat – latitude of the restaurants' location, in decimal degrees;
- rating – restaurants' ratings, decimal numbers, from 0 to 10;
- price – the price tier of the restaurant, range value from 1(not expensive) to 4 (expensive);
- categories – the restaurant category ;
- venueid – unique identification code for each restaurant.

To avoid special characters that might appear in the string, Unicode object `u""` was used (3.1.3. Unicode Strings, [docs.python.org](https://docs.python.org/3/unicodehow.html), 2017). And `"\n"` was added to the line of code to make sure each response is written on the next file line.

```

line =
u""+str(dic_key)+", "+dd['navn1']+", "+str(dd['Geo_Lng'])+", "+str
r(dd['Geo_Lat'])+", "+str(dic[dic_key]['rating'])+", "+str(dic[d
ic_key]['price'])+", "+str(dic[dic_key]['categories'])+", "+str(
dic[dic_key]['venueid'])+"\n"
    with codecs.open(csvfile, "a", "utf-8") as myfile:
        myfile.write(line)

```

(6.2. re — Regular expression operations , [docs.python.org](https://docs.python.org/3/library/re.html) , 2017)

The code ends with storing the response variable in the csv file named

'`csvrestA1.csv`'. This file will be used for the data processing and restaurants' spatial analysis.

```

# Store
fileobject = open(filename, 'w')
json.dump(dic, fileobject)
fileobject.close()

```

The data extraction returned 1840 restaurants of Denmark rated on Foursquare platform out of 9245. From these 1840 restaurants, 345 had no price value returned. Therefore they

were eliminated and the project continued with 1495 restaurants with values for all four requested variables.

The extracted restaurants had rating values were between 4.6 and 9.4.

In Denmark, there were found 74 categories and subcategories of restaurants.

4.1.3. Data processing in ArcMap and QGIS

More data processing was done in ArcMap and QGIS when the spatial variable was added to the csv file.

The csv files with the data from both sources (smiley and foursquare) were processed part in ArcMap and part in QGIS.

The smiley database was adjusted to 1495 restaurants and it was transformed into a csv file to facilitate further work.

Then the csv file containing the request from foursquare and the csv file with smiley database were joined in ArcMap using 'navnelb nr' column as seen in Figure 19.

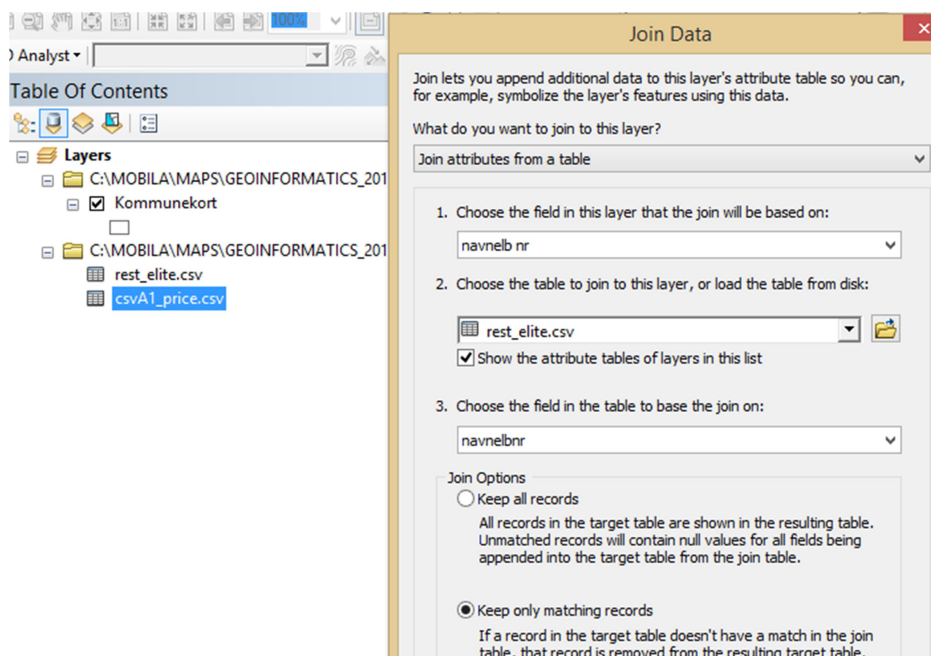


Figure 19 The join of the two csv file: the one containing the smiley information and the one containing the response from Foursquare

After the join the dataset and the final number of restaurants the project will be based on is 1492.

The joined csv file will set its 'x' and 'y' and its coordinate reference system using the tool 'Display XY data' from ArcMap (Figure 20).

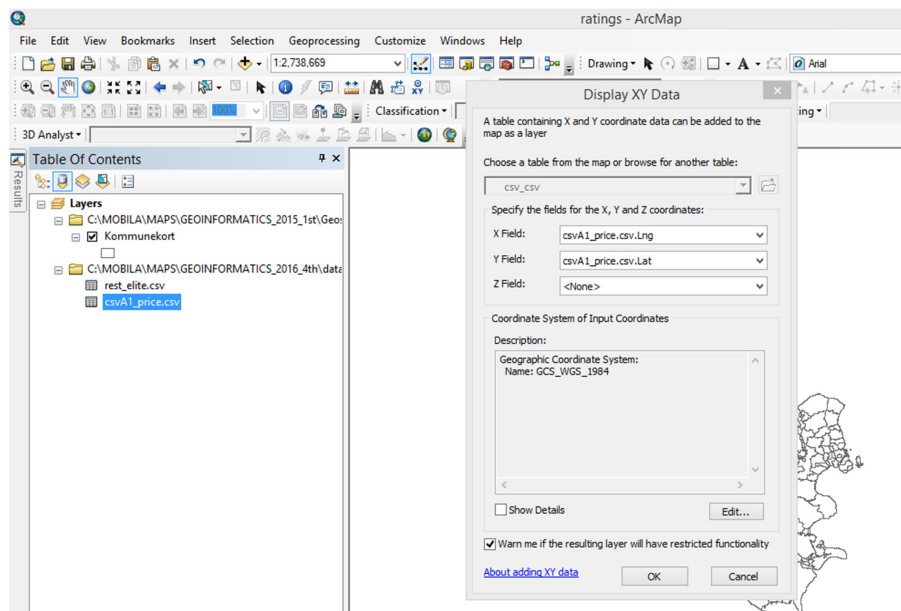


Figure 20 Adding x and y to the joined csv file

The csv file containing both smiley information and the four variables from Foursquare was transformed into a shape file named 'dk_ratings'.

From Matrikelkortet section of download.kortforsyningen.dk it was downloaded the map of Denmark with municipalities, in shapefile format. This file was added to ArcMap layer table under the name of 'dk_kort'.

Because the purpose of the project was to observe differences in restaurants variables in each Danish municipality, 'dk_kort' and 'dk_ratings' were joined as seen in Figure 21.

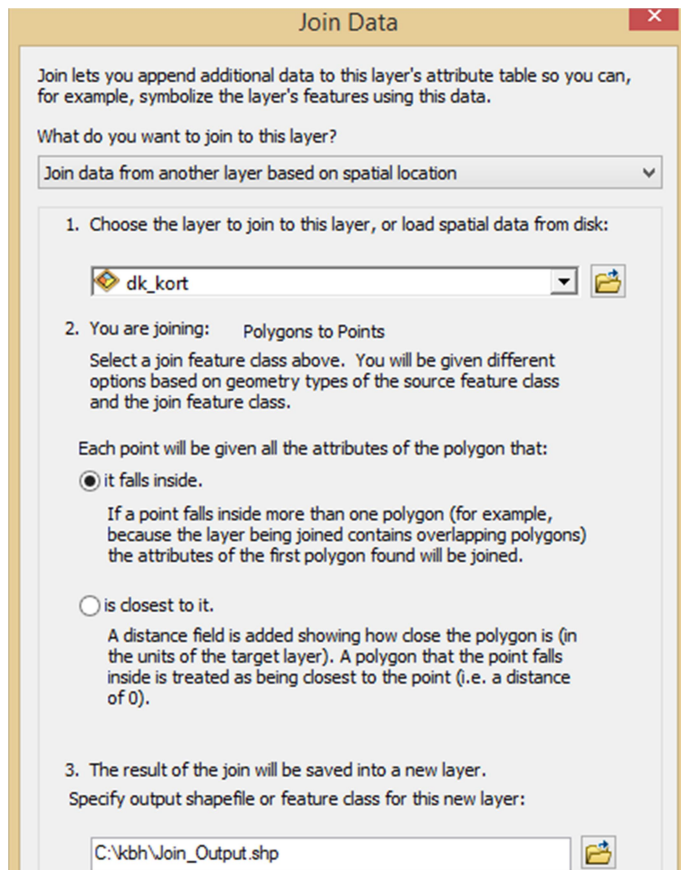


Figure 21 Join between 'dk_ratings' and 'dk_kort' files

The result was a file where each restaurant was assigned to the municipality to which it belonged and an extra column with this information was added automatically to the file attribute table. This file will be used in R and QGIS to calculate the correlations of Danish restaurants data.

For Copenhagen area, a csv file containing Copenhagen divided into districts was downloaded from portal.opendata.dk/dataset/bydele.

Copenhagen map was extracted from Denmark's map file. Then the file was joined with the csv file containing the districts. The new shape file was named 'kbh_kort'.

Restaurants located in Copenhagen municipality were extracted from the file containing all Danish restaurants and a new shape file called 'kbh_ratings'.

To assign each Copenhagen restaurant the district to which it belongs, a join was performed, the same way as in Denmark's case: 'kbh_kort' with 'kbh_ratings'. This new file will be used for further work, in the data analysis phase of the project, for Copenhagen area.

When all the shapefiles were opened in QGIS, their CRS were transformed from degrees to meters, transformation necessary for the spatial analyses. Figure 22 is presenting how this procedure was carried out.

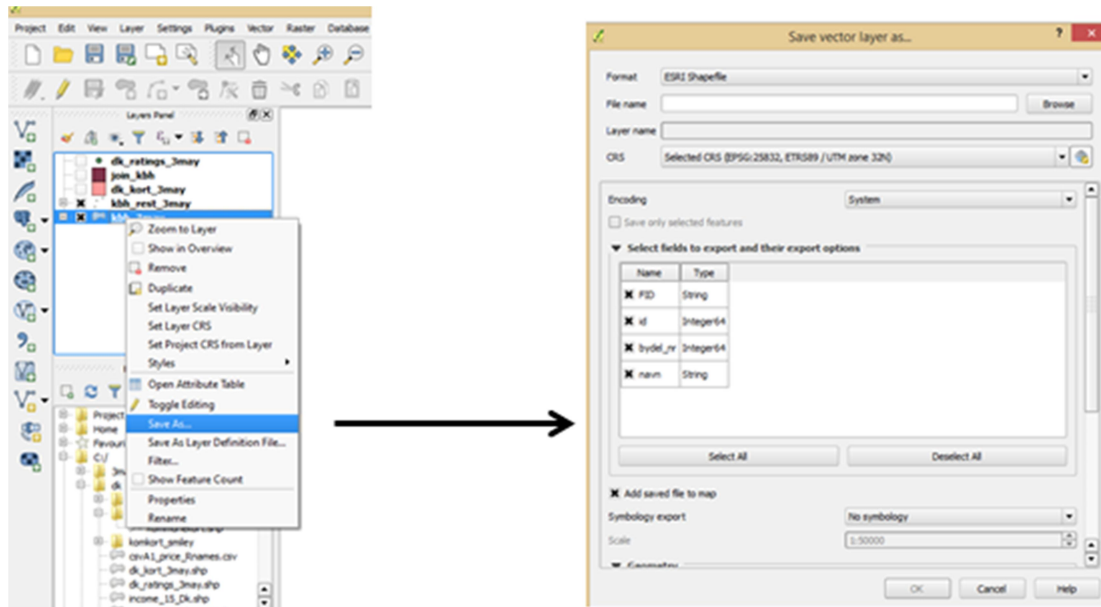


Figure 22. Saving the shapefile with the UTM 32N CRS

Before continuing with the data analysis and visualisation in Table 7 it can be seen an overview of the amount of data that will be worked with.

Unit	Divisions	Number of restaurants
Denmark	98	1492
Copenhagen	10	778

Table 7 Data overview

4.2. Data analysis and visualisation

Before the data analysis to begin, two maps were created in QGIS to visualize the restaurants distribution in Denmark (Figure 23) and in Copenhagen (Figure 24).

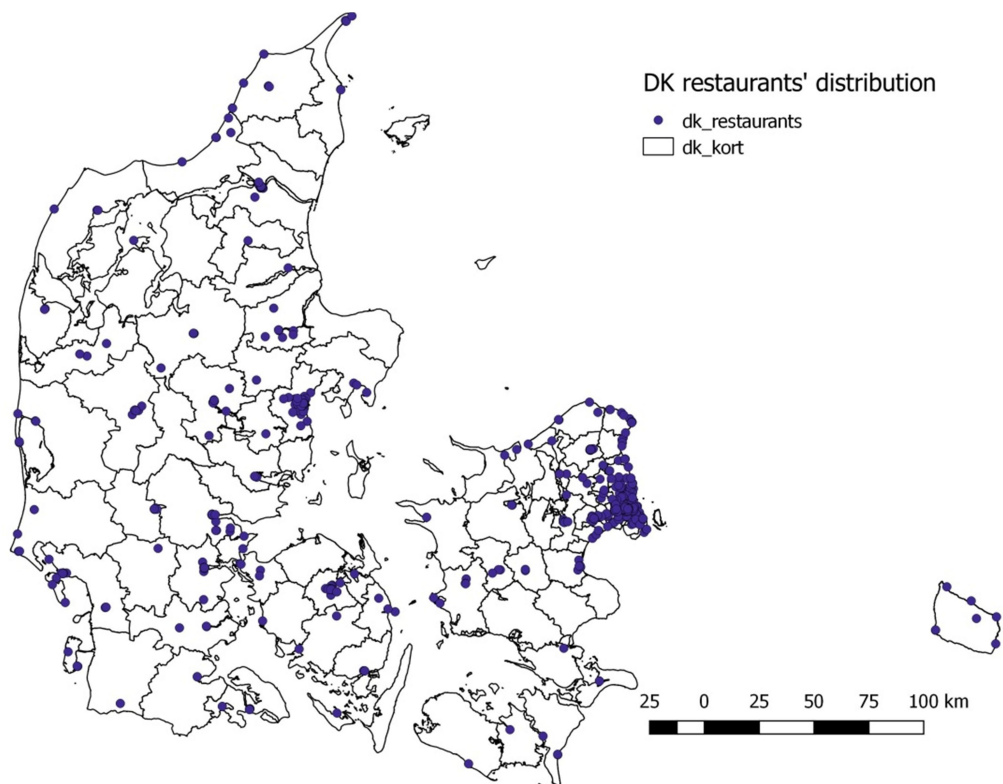


Figure 23 Denmark's restaurants distribution

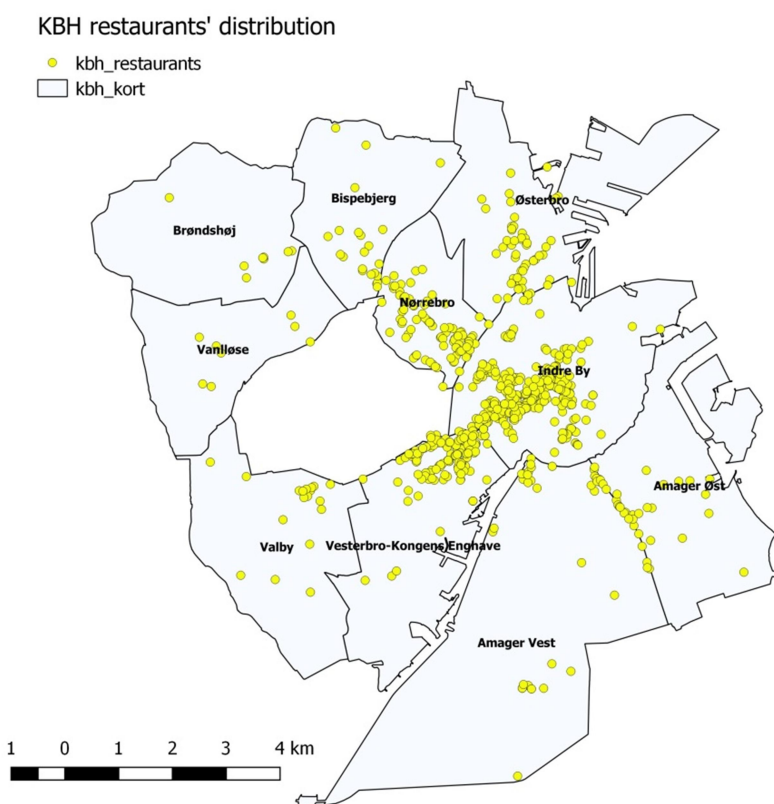


Figure 24 Copenhagen's restaurants distribution

For data distribution part R was used to run Kernel density estimation and K-function and QGIS to produce the Voronoi diagrams.

From Figure 23 it can be noticed the restaurants are clustered around major Danish cities. The cluster hypothesis of data distribution was tested using K function in R. The code for this function is written below together with code explanations.

At the beginning of the code it is set the folder where the working files are. The libraries necessary the code to run are loaded at the beginning too.

```
setwd("C:/dk")  
library(rgdal)  
library(spatstat)  
library(GISTools)  
library(sp)  
library(FNN)
```

The data are loaded.

```
kort<- readOGR('dk_kort_3may.shp','dk_kort_3may')  
dkratings<- readOGR ('dk_ratings_3may.shp', 'dk_ratings_3may')
```

'kort' file is converted to give a frame to the point pattern.

```
kort.window <- as.owin(kort)
```

The converted file is used to turn dk_ratings file into a point pattern file.

```
dkratings.ppp <- as.ppp(coordinates(dkratings), kort.window)  
(ppp, www.rdocumentation.org, 2017)
```

Run and plot the K-function.

```
kf<-Kest(dkratings.ppp, correction='border', rmax=10000)  
plot(kf)  
(Kest, www.rdocumentation.org, 2017)
```

For Copenhagen, the R code for the K-function is similar except for the working folder and the file names and the value of 'rmax'. The entire code can be found in the project appendix.

The K-function plot for Copenhagen is shown in the figure below.

Kernel density estimation was run in R for both Denmark and Copenhagen.

The lines of code for this parameter follow the code introduced above, for K-function and it consists of:

```
plot(density(dkratings.ppp))  
plot(kort, add=T)
```

For more comprehensive interpretation, Voronoi diagram was run in QGIS for Denmark and Copenhagen territory. The frame file used for this diagram was the points file one in both cases and the for the extent window it was used the polygon file, as seen in figure

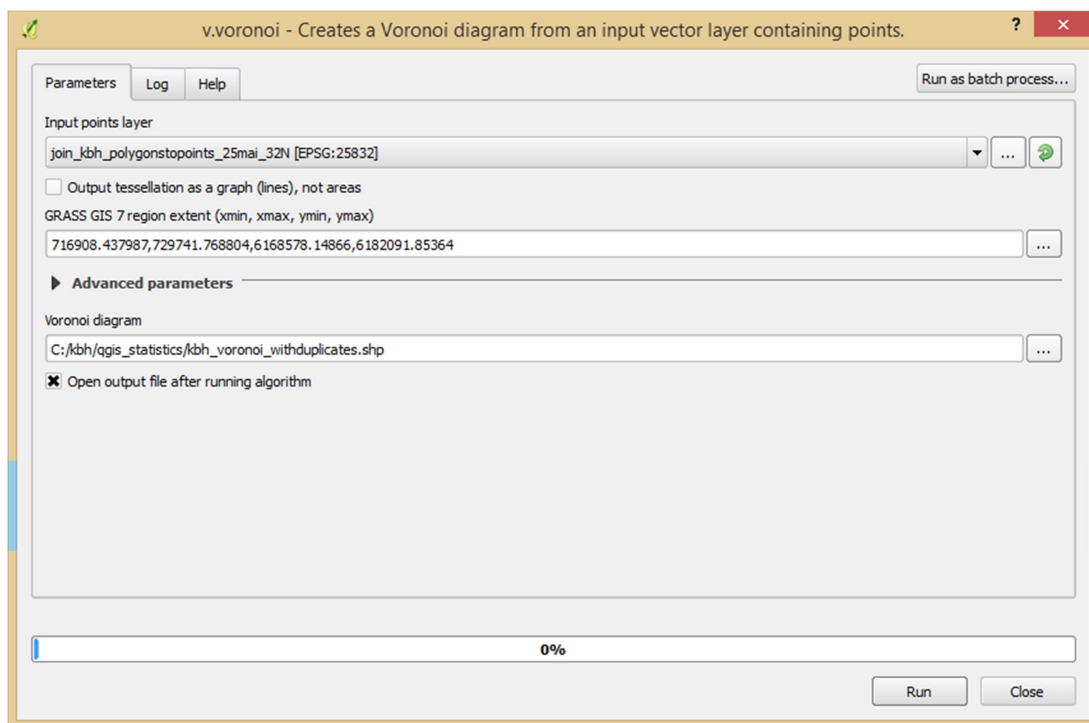


Figure 25 Voronoi diagram input form in QGIS for Copenhagen

Then a choropleth map was generated for all four variables: 'last control', 'price', 'ratings' and 'elite smiley', both for Denmark and Copenhagen.

The correlations between meaningful variables and the mean value of the four variables listed above were calculated in R for data analysis part of the project.

For this purpose, in the code introduced above will be added more lines: a line to introduce the file where each restaurant belongs to a municipality or district, lines for transforming values of working variables from the new file into numeric and the lines for calculating the Spearman correlation. In the appendix lies the entire code for both territories.

Example of the code for Spearman correlation for Copenhagen district number 10 can be read below.

```
k10<-subset(kbh_rm,kbh_rm$bydel_nr==10)
cor.test(k10$Latest_ctr,k10$rating, method = "spearman",exact=FALSE)
cor.test(k10$Latest_ctr,k10$price, method = "spearman",exact=FALSE)
cor.test(k10$Elitesmile,k10$rating, method = "spearman",exact=FALSE)
(Spearman correlation and ties, stackoverflow.com/questions, 2017 , Using R for statistical
analyses - Simple correlation, www.gardenersown.co.uk ,2017)
```

Spearman correlation was calculated for these pairs of variables: 'last control' and 'ratings', 'last control' and 'price', 'last control' and 'elite smiley'.

In case of Denmark it was noticed that the correlation done in municipalities with not enough variables was not calculated and the following error was printed by R:

```
Error in cor.test.default(dk51$1st_ctrl, dk51$ratings) :
  not enough finite observations
```

The correlation was not calculated when standard deviation was zero. Below is the R warning.

```
Warning message:
In cor(x, y) : the standard deviation is zero
```

Therefore the number of municipalities with calculated datasets changed from 98 to 37.

The correlation values were added in a csv file for each territory, and then the file was joined with the polygon layer corresponding to Denmark or Copenhagen in ArcMap. QGIS was used to visualize the data as choropleth map for each pair of calculated variables.

The mean values for Denmark's municipalities and Copenhagen districts were also calculated in R. The values were saved as a csv file that was joined, as the case of correlation values, with the polygon layer of its corresponding territory, in ArcMap.

The code below exemplifies the calculation of 'mean' for 'elite smiley', 'price', 'last control' and 'ratings' variables for each Danish municipality and also the how a new csv file is created, to incorporate all the calculated values.

```
elites <- aggregate(dk_rm$eliteS, list(dk_rm$OBJECTID), FUN=mean)
prices <- aggregate(dk_rm$price, list(dk_rm$OBJECTID), FUN=mean)
lst<-aggregate(dk_rm$lst_ctrl, list(dk_rm$OBJECTID), FUN=mean)
ratngs<-aggregate(dk_rm$ratings, list(dk_rm$OBJECTID), FUN=mean)
means <- data.frame(elites$x, prics$x, lst$x, ratngs$x)
```

(Aggregate, www.rdocumentation.org, 2017)

write.csv(means, "dk_means.csv") (Exporting a dataset from R, www.instanttr.com, 2017)

The number of Danish municipalities was 37 out of 98, being used the same csv file like the one used to calculate Spearman's correlation for the same reason of data consistency.

For Copenhagen divisions, the code is the same except the names of the files. The entire R code lies in the appendix.

Data results from both correlation and mean were visualized in QGIS using choropleth maps.

For the correlations values the settings for the choropleth map were done as in the figure below.

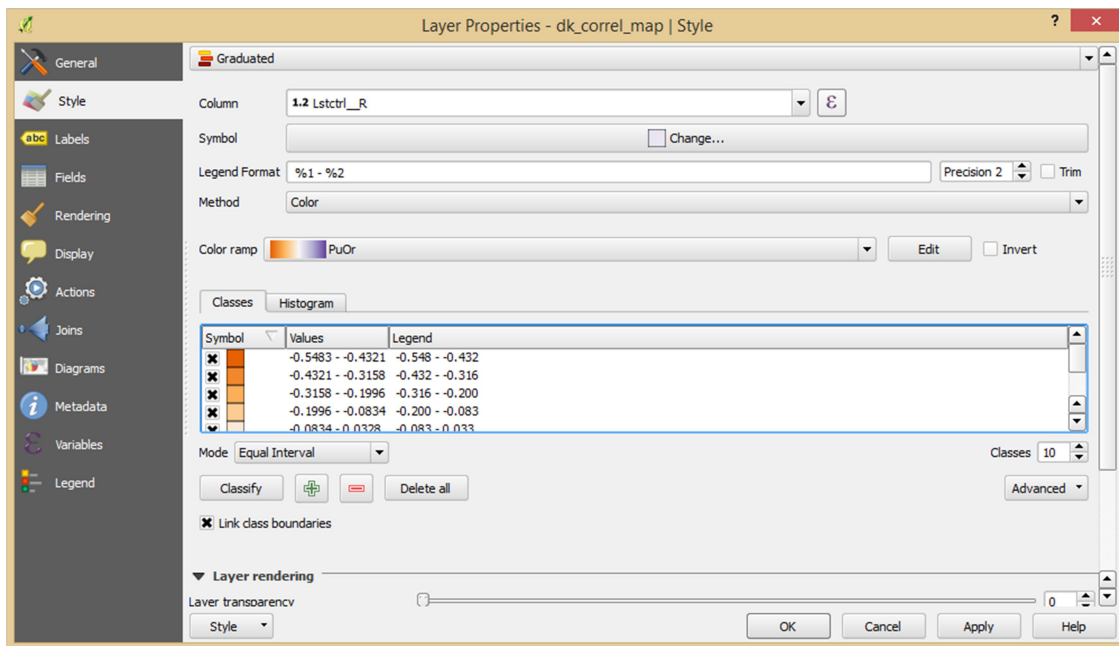


Figure 26 Settings for the choropleth map representing the Spearman correlation between 'last control' and 'ratings' for Denmark

For the mean values, the settings for the choropleth map were done as in the figure below.

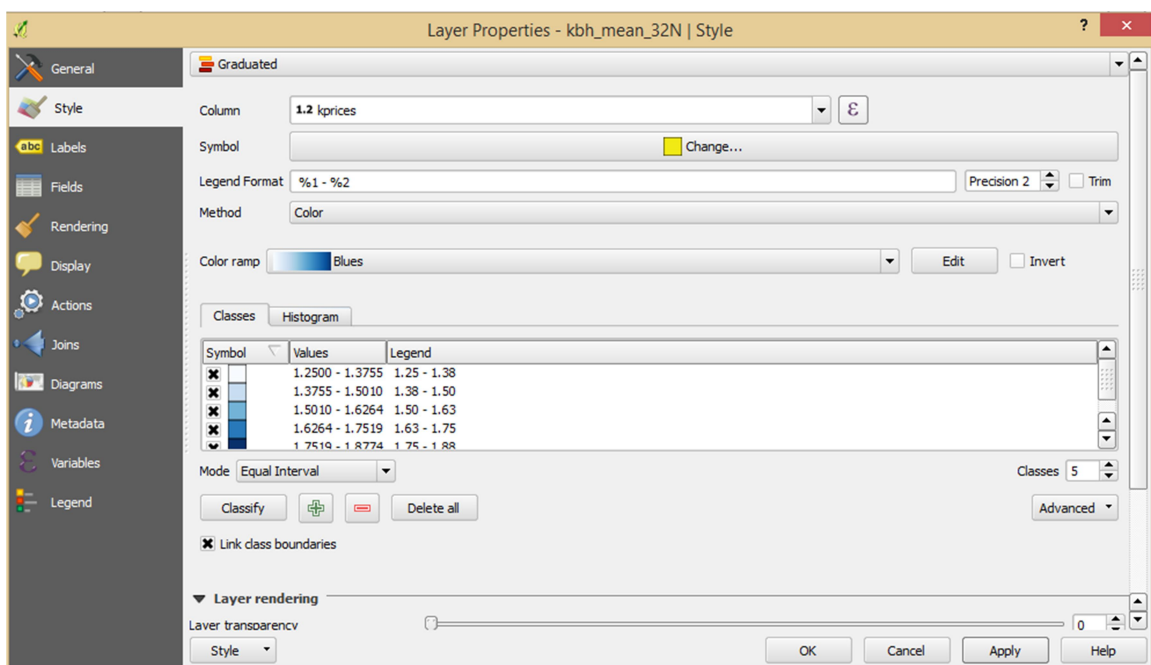


Figure 27 Settings for the choropleth map representing the mean values for the 'price' variable from Copenhagen

5. Results

This chapter includes the results of the data distribution of restaurants located in Denmark and in Copenhagen. In addition, it covers the results of Spearman's correlation and mean calculations and their visualisation on choropleth maps for both Denmark's municipalities and Copenhagen's districts.

5.1 Data distribution

Several tools were used to understand how the restaurants are distributed in Denmark and in Copenhagen area. This subchapter presents the tools and their results.

The first tool that was worked with was Kernel density estimation and it was run in R. The outcome from running this tool is a density map and its results are presented in Figure 28. The map colour code is an indication for restaurants density. The brighter the colour (yellow in this case), the denser the restaurants are in an area.

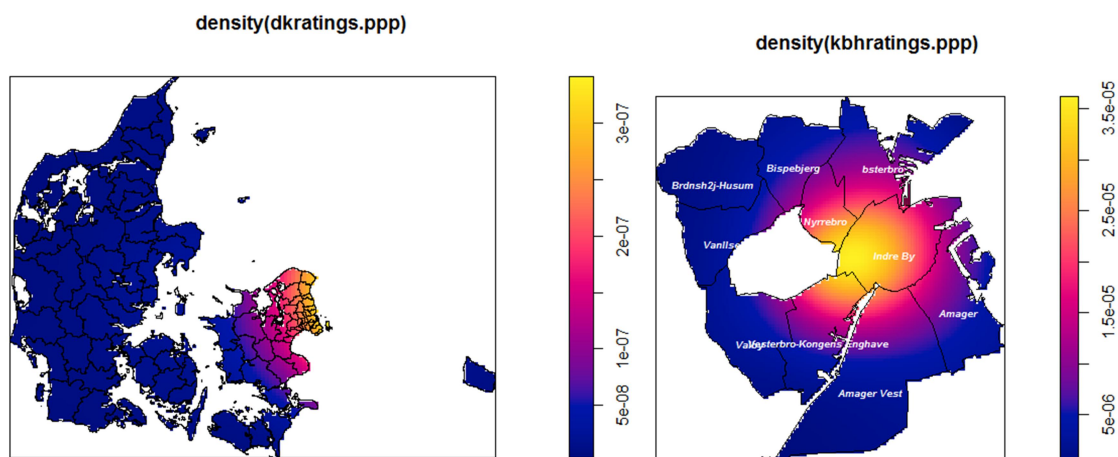


Figure 28 Kernel density estimation for Denmark (the left map) and Copenhagen (the right map)

The following tool, K-function was run in R too. Its result, presented in Figure 29 for Denmark and in Figure 30 for Copenhagen, it's a plot, where it is compared the estimated K-function (K_{bord}) with the theoretical CSR function (K_{pois}). The plot unit is meter.

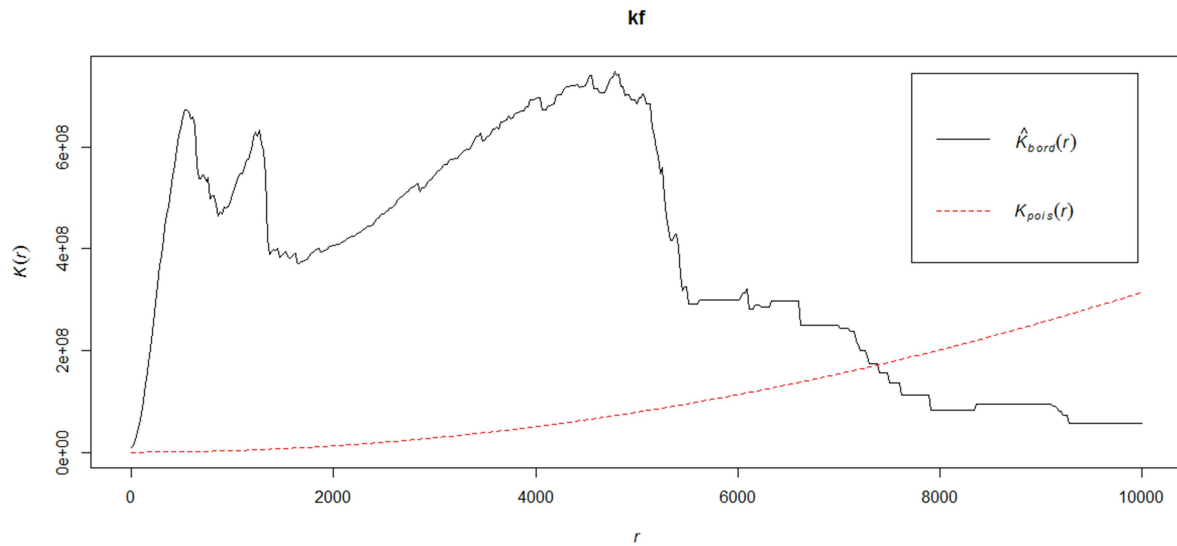


Figure 29 K-function plot for Denmark

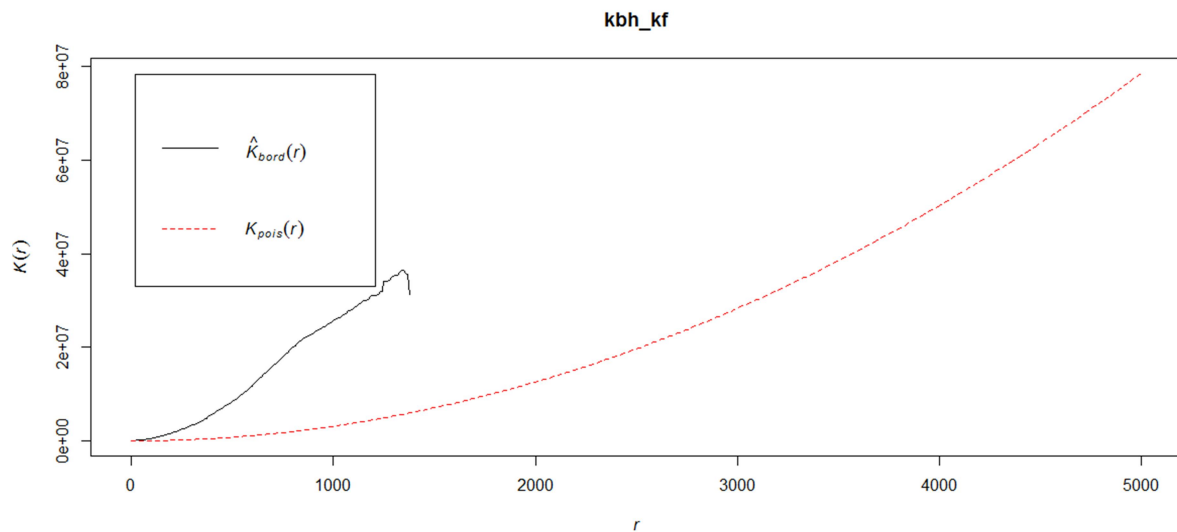


Figure 30 K-function plot for Copenhagen area

Voronoi diagram is the last tool used in this section and it was run in QGIS. The diagram was associated with choropleth map to depict the distribution of restaurants and 'last control', 'ratings', 'price' and 'elite smiley' variables in Denmark and in Copenhagen. The contour map of Denmark and Copenhagen is coloured in yellow. For Denmark, Voronoi diagrams are presented in Figure 31, Figure 32, Figure 33 and Figure 34.

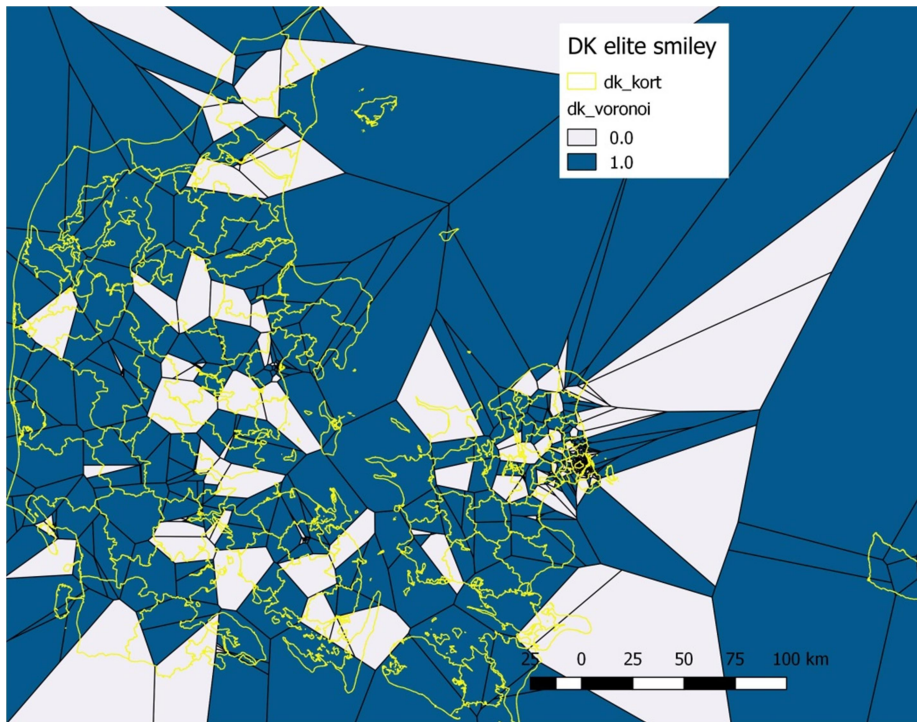


Figure 31 Voronoi diagram for Denmark, with elite smiley variable

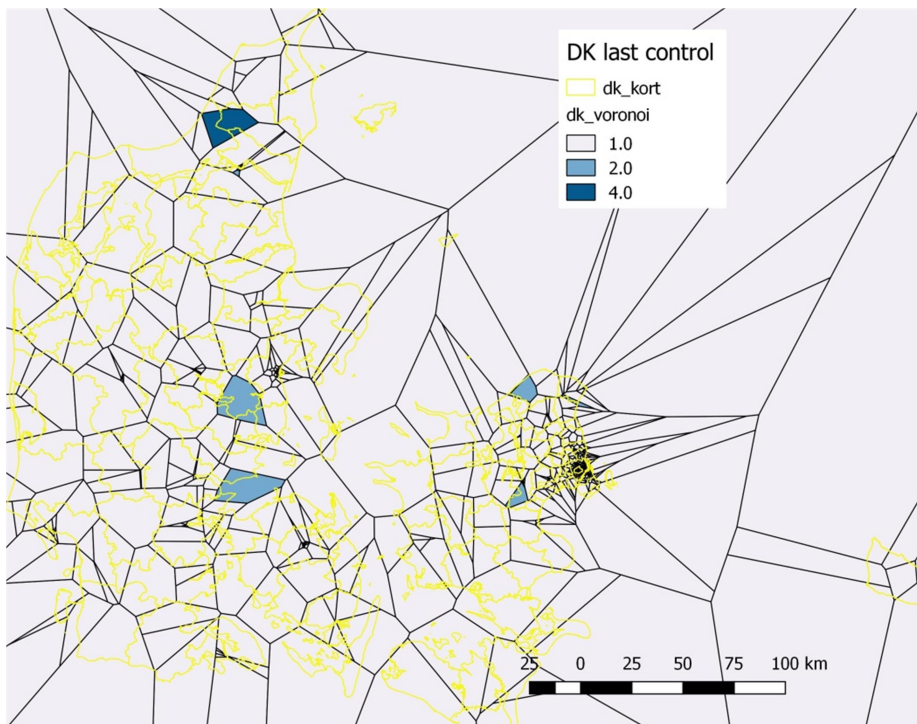


Figure 32 Voronoi diagram of Denmark with last control variable

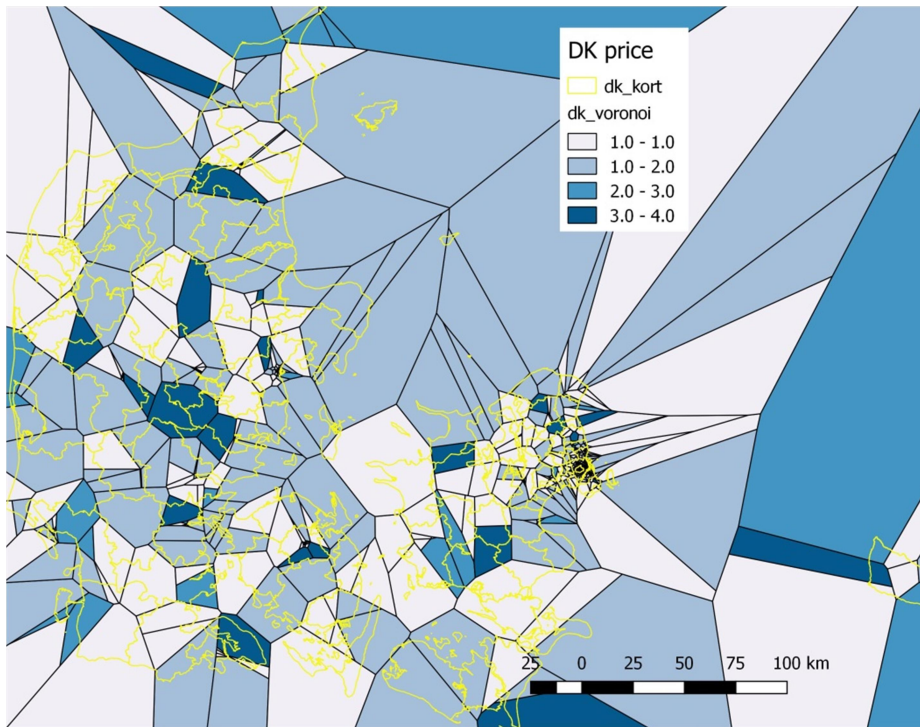


Figure 33 Voronoi diagram of Denmark with price variable

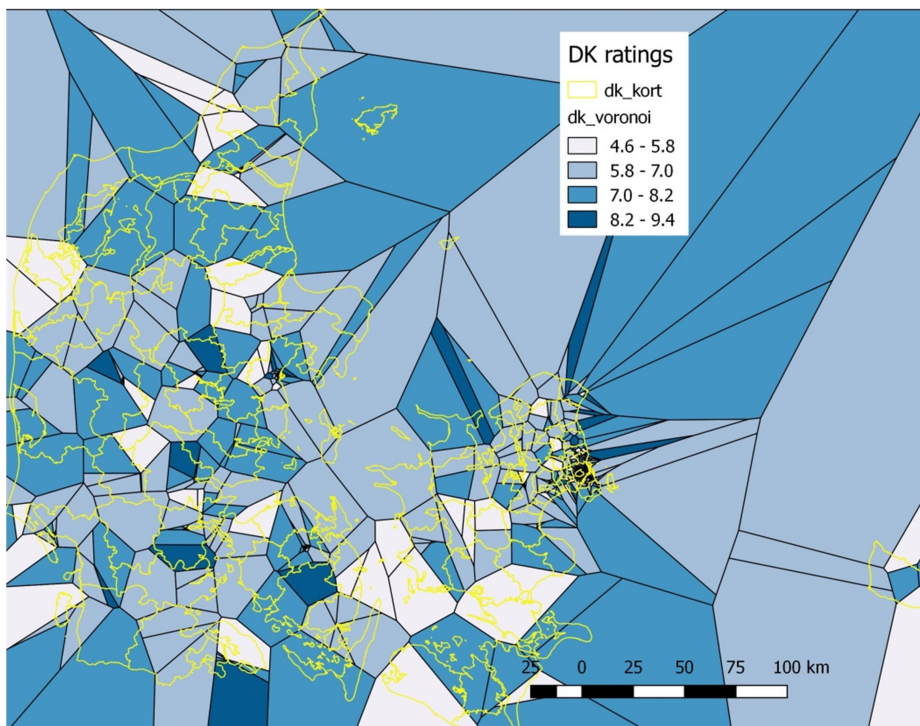


Figure 34 Voronoi diagram of Denmark with ratings variable

For Copenhagen, the Voronoi diagrams are presented in Figure 35, Figure 36, Figure 37 and Figure 38.

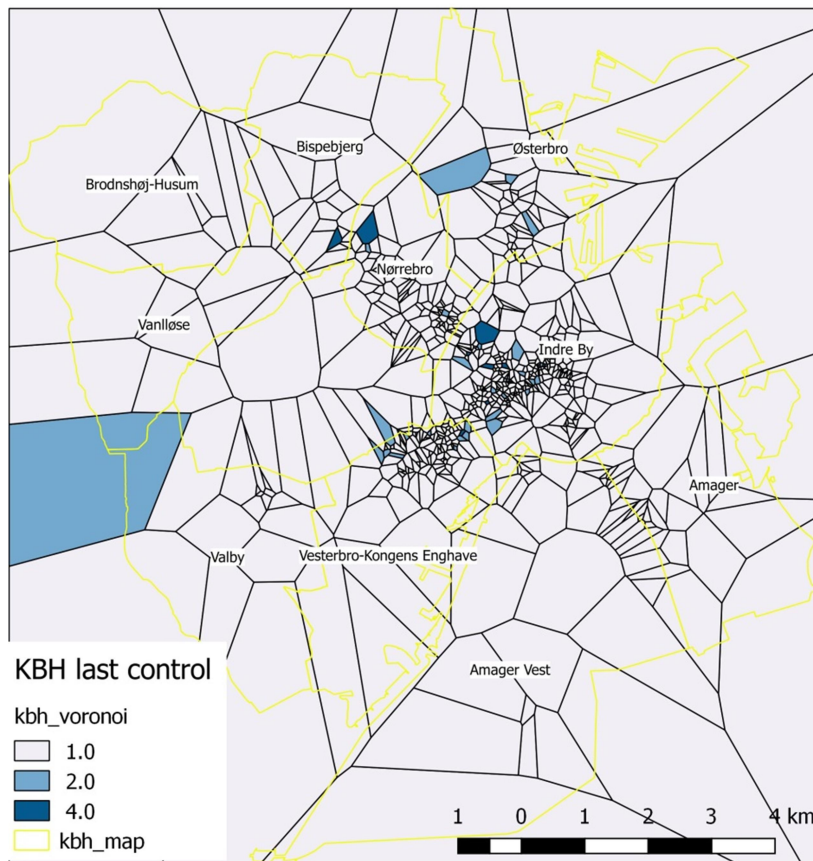


Figure 35 Voronoi diagram of Copenhagen with last control variable

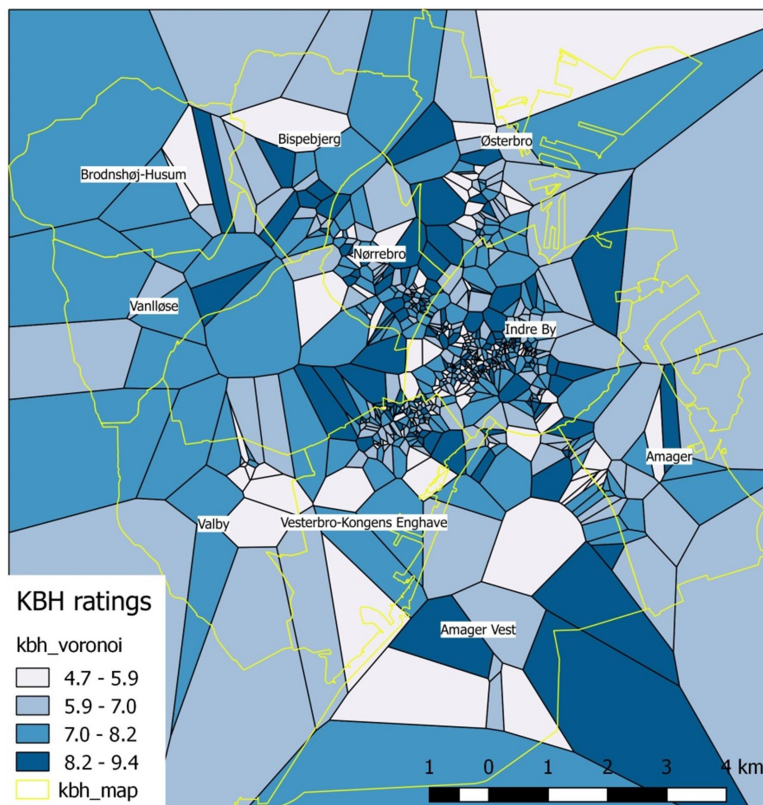


Figure 36 Voronoi diagram of Copenhagen with ratings variable

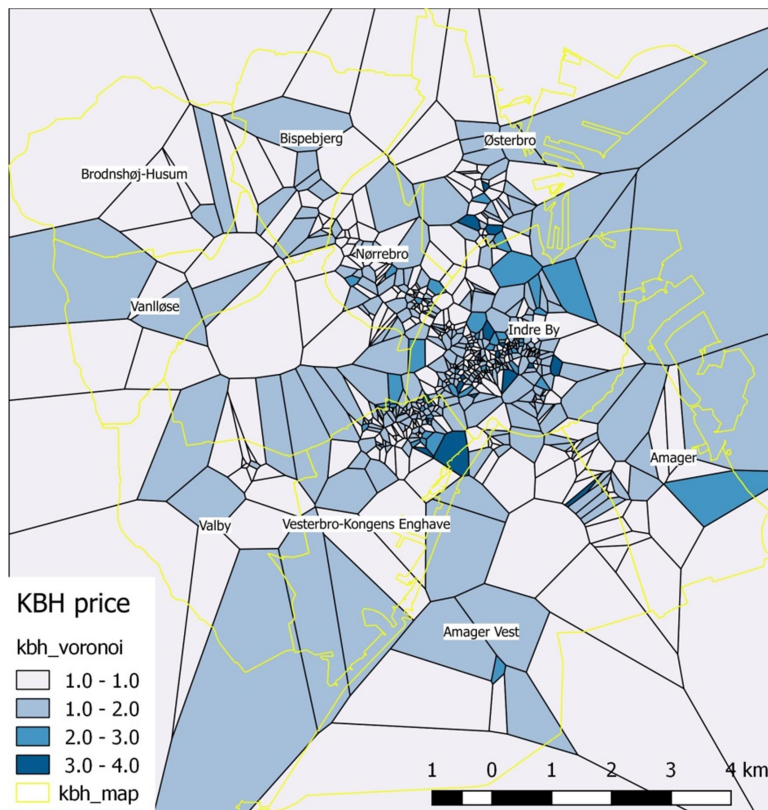


Figure 37 Voronoi diagram of Copenhagen with price variable

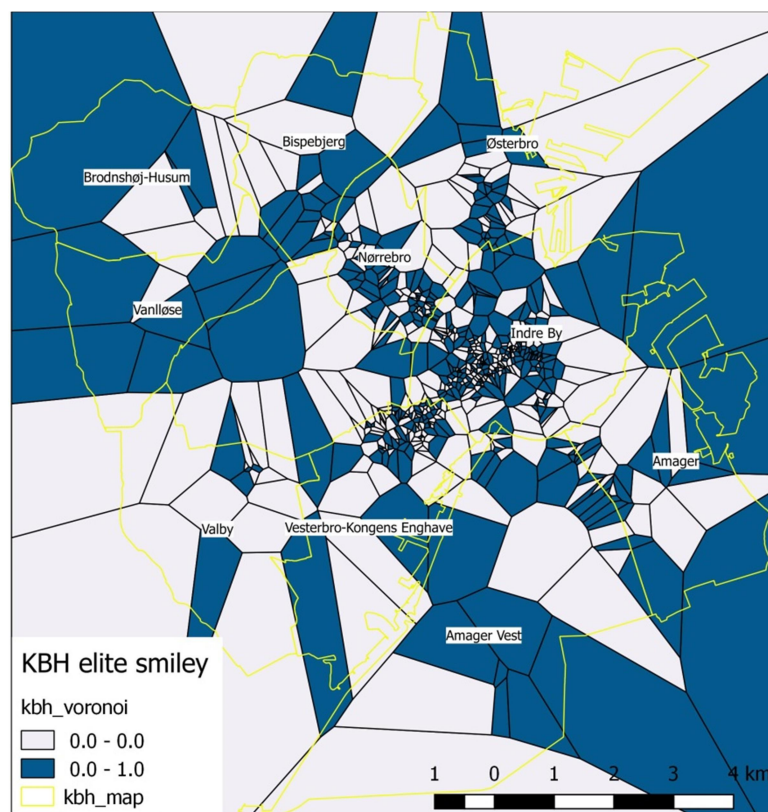


Figure 38 Voronoi diagram of Copenhagen with elite smiley variable

5.2. Data analysis

This subchapter introduces the results of the Spearman correlation values for Copenhagen (Table 8) and Denmark (Table 9) with the following pairs of variables: 'last control and ratings', 'last control' and 'price' and 'elite smiley' and 'ratings'.

The values from the table indicate that there is no relationship between: 'last control and ratings', 'last control' and 'price' pair of variables in four Copenhagen districts.

It also shows a positive relationship for two districts for 'last control and ratings' pair of variable and a negative relationship for four of the districts for the same pair of variables.

For the 'last control and price' pair there is one district with a positive relationship and five districts with negative relationships.

For the 'elite smiley and ratings' pair there are eight districts with a positive relationship and 2 districts with a negative one

No	District Name	Last control / Ratings	Last control / Price	Elite Smiley /Ratings	Restaurants	Restaurants %
1	Indre By	-0.1218362	-0.02125	0.119432	359	46
2	Østerbro	-0.07272	-0.22592	0.020453	54	7
3	Nørrebro	0.02961	-0.14673	0.094109	99	13
4	Vesterbro-Kongens Enghave	-0.13434	0.065888	0.064316	138	18
5	Valby	0.194504	-0.14086	0.164386	19	2
6	Vanløse	0	0	0.412393	8	1
7	Brønshøj-Husum	0	0	-0.52289	8	1
8	Bispebjerg	-0.31	-0.18157	0.466969	22	3
9	Amager Øst	0	0	-0.31028	21	3
10	Amager Vest	0	0	0.092904	50	6
Copenhagen restaurants total					778	100

Table 8 Spearman correlation values for Copenhagen districts

The values from Denmark's Spearman correlation table indicate there is no relationship between: 'last control and ratings', 'last control' and 'price' pair of variables in 26 of Denmark's municipalities.

It also shows a positive relationship for four municipalities for 'last control and ratings' and 'last control and price' pairs of variable and a negative relationship for seven of the municipalities for the same pairs of variables.

For the 'elite smiley and ratings' pair there are five municipalities that have no correlation relationship, 20 municipalities with a positive relationship and 12 municipalities with a negative one.

Objectid	Municipality name	Last control / Ratings	Last control / Price	Elite Smiley /Ratings	Restaurants	Restaurants %
1	København	-0.0951	-0.0142	0.0986	778	55.8
2	Frederiksberg	-0.0182	-0.084	0.3632	81	5.8
6	Gentofte	0	0	0.2853	29	2.1
7	Gladsaxe	0	0	0.3044	9	0.6
12	Høje-Taastrup	-0.3402	-0.2182	0.0581	8	0.6
13	Lyngby-Taarbæk	-0.3191	0.1324	0.2182	20	1.4
14	Rødovre	0	0	0	7	0.5
16	Tårnby	0	0	-0.5196	9	0.6
21	Helsingør	0	0	0.0867	18	1.3
22	Hillerød	0	0	0.2639	11	0.8
23	Hørsholm	0	0	-0.2185	7	0.5
24	Rudersdal	0	0	-0.2887	5	0.4
27	Greve	0	0	0.3536	5	0.4
28	Køge	0	0	0.5266	11	0.8
30	Roskilde	-0.0315	0.2819	-0.0978	19	1.4
38	Slagelse	0	0	-0.3314	7	0.5
46	Bornholm	0	0	0	7	0.5
47	Middelfart	0	0	-0.6455	5	0.4

53	Odense	0.0608	0.0515	-0.1648	47	3.4
54	Svendborg	0	0	0	7	0.5
59	Billund	0	0	0.22	6	0.4
60	Sønderborg	0	0	0.1443	7	0.5
61	Tønder	0	0	-0.3536	5	0.4
62	Esbjerg	0	0	0.2845	16	1.1
68	Horsens	0	0	0.4119	7	0.5
69	Kolding	0	0	-0.2636	15	1.1
70	Vejle	0	0	0.1264	17	1.2
71	Herning	0.3792	-0.232	0.0178	14	1
79	Randers	0	0	0	9	0.6
80	Silkeborg	0	0	0.302	17	1.2
83	Aarhus	0.0251	0.0641	-0.0606	121	8.7
85	Ringkøbing-Skjern	0	0	0	5	0.4
90	Viborg	0.5314	0.4648	-0.4951	6	0.4
92	Frederikshavn	0	0	0.5879	7	0.5
97	Jammerbugt	-0.7071	-0.6124	0.866	5	0.4
98	Aalborg	-0.3104	-0.0395	0.3496	39	2.8
99	Hjørring	0	0	-0.126	8	0.6
Denmark restaurants from 37 municipalities					1394	100

Table 9 Spearman correlation values for Danish municipalities

This section includes the results for the mean values calculated for Copenhagen (Table 10) and Denmark (Table 11).

Nr	Name	Elites	Price	Last control	Ratings	Restaurants	%
1	Indre By	0.5682	1.8774	1.0975	7.0838	359	46
2	Østerbro	0.5741	1.6296	1.0556	6.8611	54	7
3	Nørrebro	0.5455	1.4545	1.0505	7.3263	99	13
4	Vesterbro-Kongens Enghave	0.4058	1.7754	1.0942	7.0072	138	18
5	Valby	0.2632	1.2632	1.0526	6.5421	19	2
6	Vanløse	0.875	1.375	1	7.3	8	1
7	Brønshøj-Husum	0.375	1.25	1	7.1125	8	1

8	Bispebjerg	0.5909	1.4091	1.1364	7.3091	22	3
9	Amager Øst	0.5714	1.381	1	6.8286	21	3
10	Amager Vest	0.62	1.54	1	6.908	50	6
Copenhagen restaurants total						778	100

Table 10 Mean values for Copenhagen districts, calculated for four variables

OBJ_ID	Municipality	Elites	Price	Last control	Ratings	Restaurants	%
1	København	0.5347	1.7134	1.0771	7.0631	778	55.8
2	Frederiksberg	0.4815	1.5432	1.0741	6.9235	81	5.8
6	Gentofte	0.5862	1.5172	1	7.131	29	2.1
7	Gladsaxe	0.5556	1.4444	1	6.4889	9	0.6
12	Høje-Taastrup	0.375	1.25	1.125	6.6	8	0.6
13	Lyngby-Taarbæk	0.7	1.75	1.05	6.96	20	1.4
14	Rødovre	0.7143	1.7143	1	6.4857	7	0.5
16	Tørnby	0.4444	1.7778	1	6.9333	9	0.6
21	Helsingør	0.4444	1.6111	1	7.3444	18	1.3
22	Hillerød	0.8182	1.9091	1	6.5727	11	0.8
23	Hørsholm	0.4286	2	1	7.0714	7	0.5
24	Rudersdal	0.6	1.8	1	7.56	5	0.4
27	Greve	0.2	1.6	1	6.72	5	0.4
28	Køge	0.8182	2.0909	1	6.6182	11	0.8
30	Roskilde	0.5789	1.6316	1.1053	6.6579	19	1.4
38	Slagelse	0.7143	1.2857	1	6.6	7	0.5
46	Bornholm	1	2.1429	1	6.4	7	0.5
47	Middelfart	0.6	1.8	1	6.12	5	0.4
53	Odense	0.617	1.766	1.1277	6.9234	47	3.4
54	Svendborg	0.5714	1.5714	1	6.5429	7	0.5
59	Billund	0.6667	1.6667	1	6.2333	6	0.4
60	Sønderborg	0.4286	1.7143	1	7.4	7	0.5
61	Tønder	0.8	2	1	6.6	5	0.4
62	Esbjerg	0.75	1.8125	1	7.0938	16	1.1
68	Horsens	0.8571	2	1	7.7714	7	0.5
69	Kolding	0.4667	1.9333	1	7.1467	15	1.1
70	Vejle	0.6471	1.5294	1	6.7882	17	1.2
71	Herning	0.5	1.7143	1.0714	7.0643	14	1
79	Randers	0.6667	1.5556	1	6.4222	9	0.6
80	Silkeborg	0.5294	1.8824	1	6.8353	17	1.2
83	Aarhus	0.7107	1.7025	1.0248	7.1	121	8.7
85	Ringkøbing-Skjern	1	2.4	1	7.14	5	0.4
90	Viborg	0.5	1.6667	1.1667	7	6	0.4
92	Frederikshavn	0.5714	2.1429	1	7.0143	7	0.5
97	Jammerbugt	0.6	1.6	1.6	6.16	5	0.4

98	Aalborg	0.641	1.7436	1.0513	7.1974	39	2.8
99	Hjørring	0.75	2.125	1	6.6125	8	0.6
Denmark restaurants from 37 municipalities						1492	100

Table 11 Mean values for Denmark's municipalities, calculated for four variables

5.3. Data visualisation

This section the results of data analysis displayed on choropleth maps. Each area, Denmark and Copenhagen, has three maps generated for values of Spearman's correlation and four maps for values of the mean.

For Denmark, the Spearman's correlation values maps can be seen in Figure 39, Figure 40 and Figure 41 and the results for mean maps in Figure 42 and Figure 43.

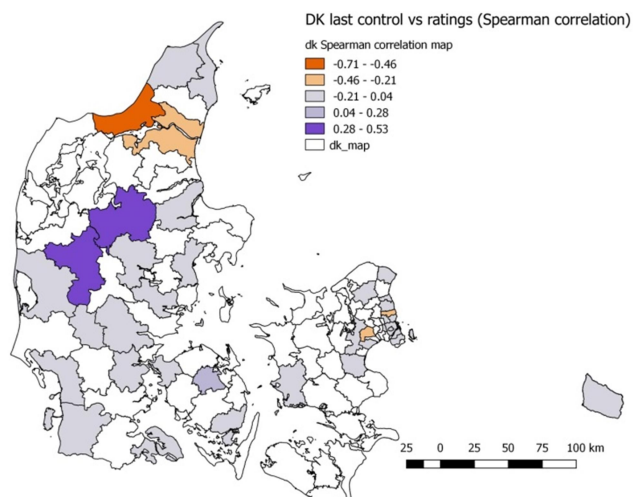


Figure 39 Denmark, Spearman's correlation for last control versus ratings

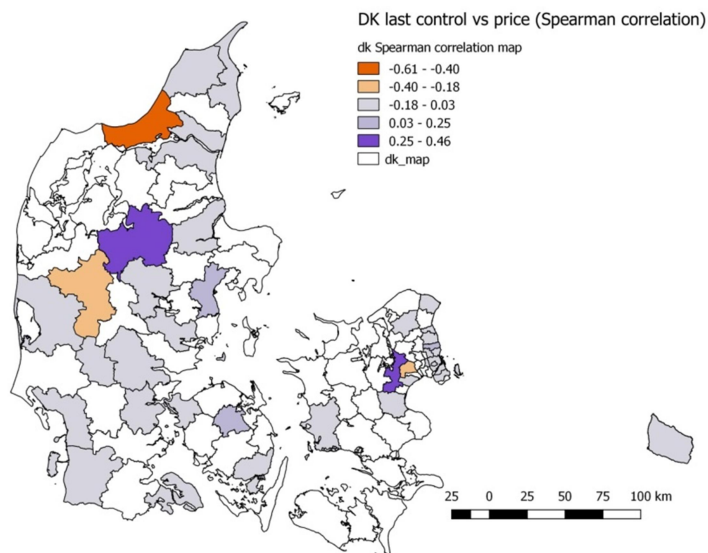


Figure 40 Denmark, Spearman's correlation for last control versus price

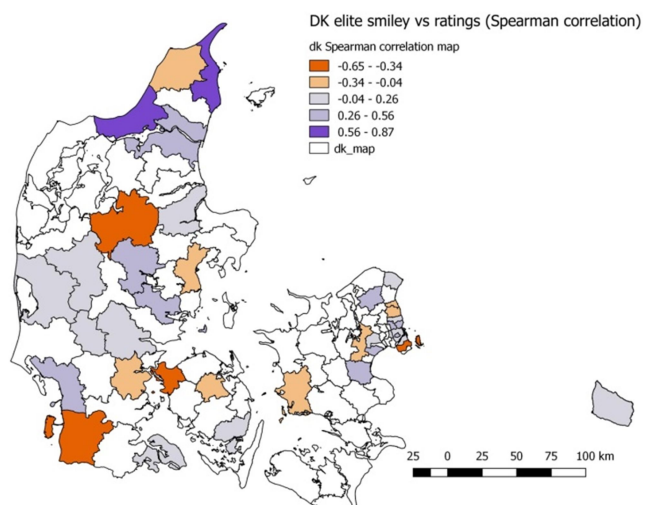


Figure 41 Denmark, Spearman's correlation for elite smiley versus ratings

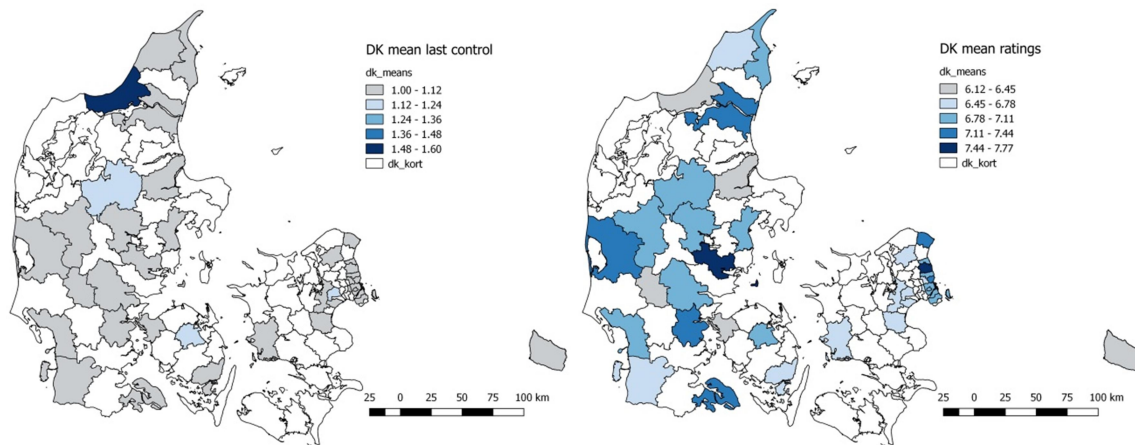


Figure 42 Denmark, mean value for last control (left) and the mean value for ratings (right)

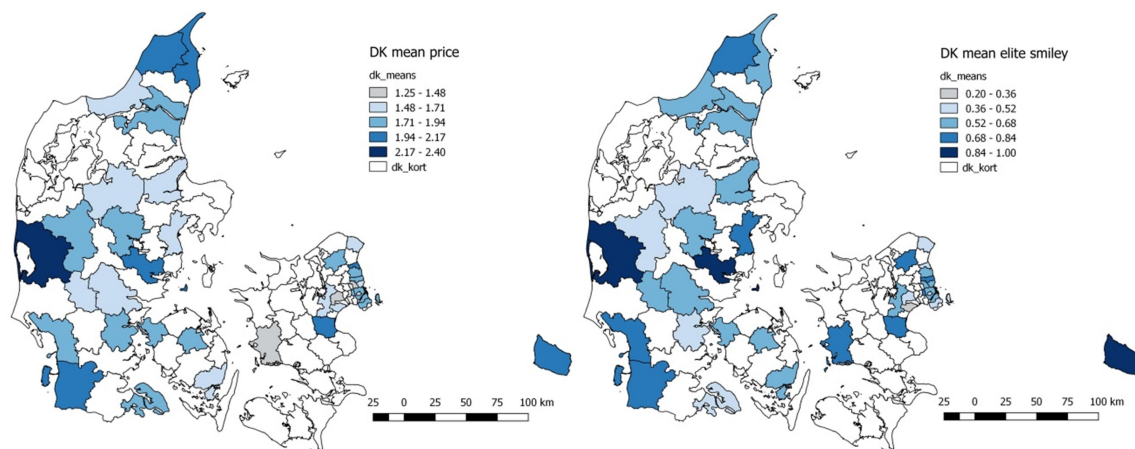


Figure 43 Denmark, mean value for price (left) and the mean value for elite smiley (right)

For Copenhagen, the Spearman's correlation values maps can be seen in Figure 44 and Figure 45 and the results for mean maps in Figure 46 and Figure 47.

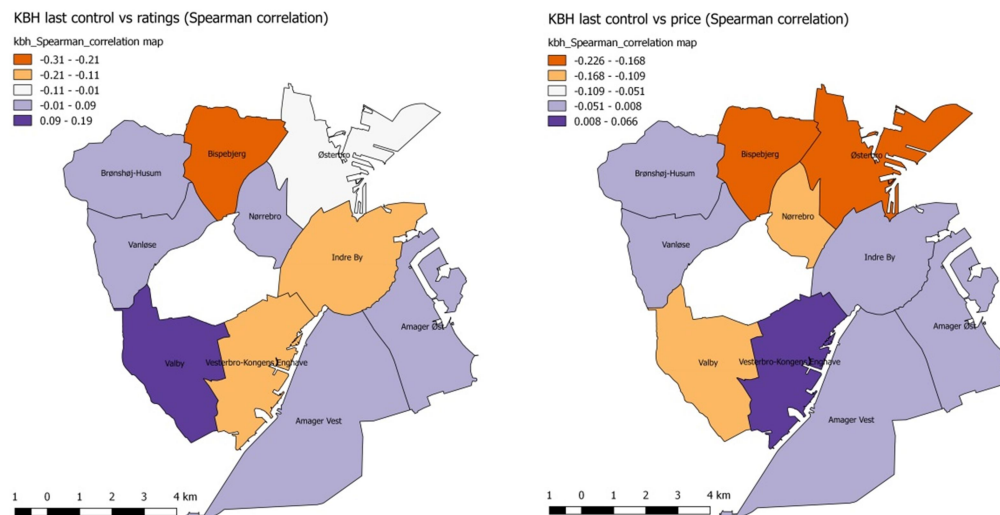


Figure 44 Copenhagen, Spearman's correlation for last control versus ratings (left) and for last control versus price (right)

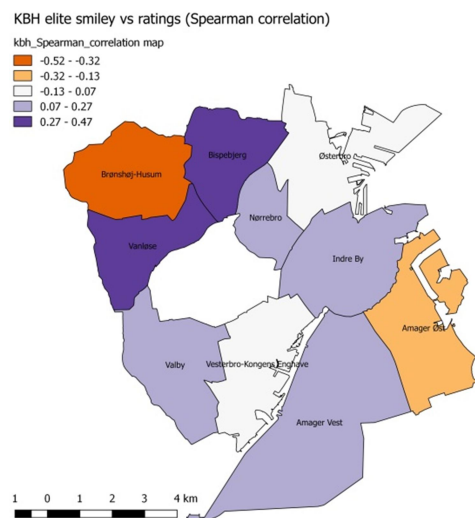


Figure 45 Copenhagen, Spearman's correlation for elite smiley versus ratings

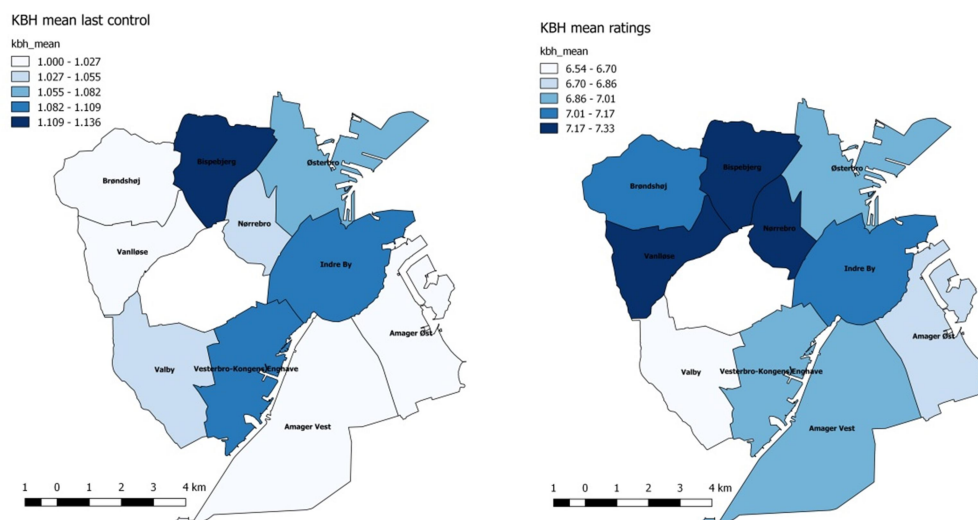


Figure 46 Copenhagen, mean value for last control (left) and the mean value for ratings (right)

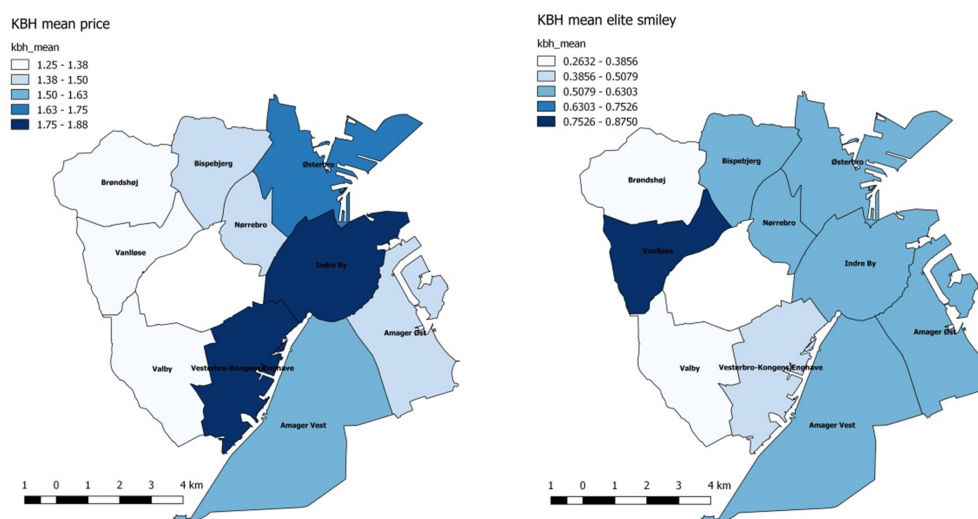


Figure 47 Copenhagen, mean value for price (left) and the mean value for elite smiley (right)

6. Discussion

This chapter is reserved for answering the research questions, analysing the project limitations and discussing the future improvements of the data analysis.

6.1. Answers to the research questions

1. Given Denmark's administrative division, what are the municipalities holding the cleaner restaurants that are also best rated by the customers?

The answer this question Spearman's correlation coefficient will be used and it will be calculated for each pair of significant variables. The mean for significant variables will be also calculated.

The relationship between the last control and ratings shows that Jammerbugt municipality is the cleanest, because the Spearman correlation values of -0.7071 indicate a negative and strong relationship, meaning the latest control values decreases and the ratings values increases. As presented in the section about last control report, the value of 1 stands for the best and the value of 4 stands for the worst.

Clean restaurants with high ratings can be found in Høje-Taastrup, Lyngby-Taarbæk and Aalborg municipality, however their correlation values indicates a moderate relationship.

If the relationship between 'elite smiley' and 'ratings' is considered, the value of 0.866 shows a strong relation between two variables, meaning the higher the elite value, the higher the ratings value are for the restaurants. Jammerbugt municipality is again indicated as having the cleanest and the best rated restaurants.

Strong relationships between the two variables have restaurants from Frederikshavn, with a value of 0.5879 and from Køge, with a value of 0.5266.

Other clean restaurants from 'elite smiley' and 'ratings' relationship point of view are located in Horsens, Frederiksberg, Greve, Aalborg, Gladsaxe and Silkeborg municipalities. However, their values from 0.302 to 0.4119 indicate a moderate relationship.

The correlation values of Danish municipalities can visualized in a spatial context in the results chapter, Figure 39, Figure 40 and Figure 41.

From the point of view of the mean values, municipality with the best scores for 'latest control', 'ratings' and 'elite smiley' are localized on the maps from Figure 42 and Figure 43.

2. What is the relation between the income per municipality and the restaurant cleanliness, the price and the customer ratings?

For time limitation reasons, the income values for each municipality were not correlated with the rest of the values. However, Spearman's correlation coefficient for 'last control' and 'price variable' variables was calculated. Also the answer of this question will include information about the restaurants spatial distribution in Denmark benefitting of the output plots created by Kernel estimation density, K-function and Voronoi diagram.

Having in mind the explanations for the 'last control' variable values, the relationship between 'last control' and 'price' show a strong negative correlation for restaurants located in Jammerbugt municipality. The value of -0.6124 indicates an increase of last control values and a decrease of price values.

Other two municipalities Herning (- 0.232) and Høje-Taastrup (-0.2182) correlation values show a weak relationship between the 'last control' and 'price'.

Restaurants distribution in Denmark was analysed using Kernel density estimation. The plot generated by this tool (Figure 28) indicates in light colour tones the restaurants are clustered in Copenhagen area. However, k-function was run to have a better understanding of the data distribution. The plot of K-function from Figure 29 indicates three clustered areas that could be associated with the biggest urban areas of Denmark: Copenhagen, Aarhus and Odense. The estimated function (K_{bord}) lines has two high peaks at 2 km and 4 km and a smaller one around 6 km. Plus, estimated function (K_{bord}) is greater than theoretical function under CSR (K_{pois}).

Also for a more clear understanding of restaurants distribution, Voronoi diagram was used. In Figure 31, Figure 32, Figure 33 and Figure 34 it can be noticed the clustering restaurants situations. The cluster areas can be easily located knowing that the smaller the area of a polygon is, the denser the restaurants inside the polygon are, so cluster zone are expected to be found around high populated urban areas like Copenhagen, Aarhus, Odense and Aalborg. The colour code in the four maps was used to indicate the variable distribution.

3. In the relation between food safety, price and customer reviews how is Copenhagen represented in each of its districts?

This question will be answered first from the data distribution point of view. Then it will be introduced the results of Spearman's correlation coefficient and the mean values together with the figure indication where these values can be seen in a spatial context.

For Copenhagen area, Kernel estimated density localizes the cluster zone approximately in the Indre By district and this can be seen in Figure 28.

The plot for K-function from Figure 30 indicates a cluster zone at about 1.5 km. The estimated function (Kbord) line is stopping after generating this peak.

Voronoi diagram for Copenhagen area follows the same pattern like in Denmark's case: the smaller the polygons, the denser the restaurants inside them. Cluster zone can be noticed in Indre by, Nørrebro and Vesterbro-Kongens Enghave districts. The colour code in the four maps was used to indicate the variable distribution. Voronoi diagrams of Copenhagen can be seen in Figure 35, Figure 36, Figure 37 and Figure 38.

The relationship between the 'last control' and 'ratings' shows that Bispebjerg district could be the cleanest, because the Spearman correlation values of -0.31 indicate a negative and moderate relationship, meaning the latest control values decreases and the ratings values increases.

For other three Copenhagen districts, Vesterbro-Kongens Enghave, Indre By and Østerbro, the correlation values indicate a negative weak relationship.

Nørrebro and Valby correlation values indicate a positive weak relationship.

The relationship between the 'elite smiley' and 'ratings' shows that Vanløse and Bispebjerg districts are the cleanest, because the Spearman correlation values of 0.4124 and respectively of 0.467 indicate a moderate and positive relationship. The higher the score of elite smiley is, the higher the ratings are.

For the same pair of variables, Brønshøj-Husum, with a correlation coefficient of -0.5229 has negative and strong relationship. Amager district can be considered in the same category but with a different value, of -0.3103.

The relationship between 'last control' and 'price' show a weak negative correlation for restaurants located in Østerbro. The value of -0.2259 indicates an increase of last control values and a decrease of price values. Bispebjerg, Nørrebro and Valby show also a weak negative correlation but with values of -0.1816, -0.1467 and -0.1409.

The correlation values of Copenhagen districts can visualized in a spatial context in the results chapter Figure 44 and Figure 45.

From the point of view of the mean values, the districts with the best scores for 'latest control', 'ratings' and 'elite smiley' are localized on the maps from Figure 46 and Figure 47.

6.2. Limitations

It is uncertain if the result of the report would have been different if more data (restaurants) would have been extracted from Foursquare rating platform. The project started with 9245 restaurants from the smiley database and ended up with 1492, of which 778 being located in Copenhagen area. A possible cause of this issue might be caused by the database accuracy information.

A list of the issues found when data were processed for this project and how these issues could be overcome is presented below:

- Many of the restaurants were discovered to have wrong address, wrong name or in some cases not existing anymore in fact. To avoid issues of this kind, the database should make more use of the column with unique values for each restaurant when data is manipulated by adding, modifying or removing. In the same category of issues falls the case of hotels that have restaurants with a different name than the name of the hotel. This situation was encountered in many cases and the restaurant was not added to the rating list because it had in the name column was written the hotel name and not the restaurant one.
- 877 restaurants were missing longitude and latitude values. To add the spatial information of this kind it is recommended to use either the tool used in the project

to add the necessary information, either to automatize the process using a Python script that can automatically introduce the coordinates after reading the name of the restaurant.

- Many restaurant names contained more words or characters than necessary: the address, type of company (a/v). The reason the address is present in the name column might have sense if the same restaurant has many locations and displays the same name. However, it is recommended to use the restaurant unique value column to overcome confusions and error in case the database is read and connected to another database.
- A situation that was discovered is the one of industry codes or branchekodes. The smiley database does not have newest codes and this caused confusion when restaurants were chosen for the processing part. Because of this, the database for this project food venues that are not restaurants per se but fast-foods although in the processing part of the project the known chains of fast food were removed from the working data. Python code and the unique value column could be used again to change the old industry codes with the new ones.

Another limitation to the project data quantity might be connected to the 'Venue Response' because not all the venues will have ratings.

A clarification it is necessary to be made in the case of elite smiley variable. If a restaurant has three smileys and it was recently opened, in the elite smiley cell the value will be zero. This is said to have in mind that a restaurant with zero elite smiley is not necessary unclean.

A limitation to the project results is considered to be the dynamics of the ratings website. If the data are not extracted all at once, they could be subject to modification due to continuous changes of the online environment.

6.3. Further Research and Improvements

- Kf function graph for Copenhagen area does not continue after a certain distance. This should be further investigating to understand the reason for the stop.
- Assess Python and R script and improve the code where necessary.

- The spatial data analysis could be improved by using ArcMap and running Geographically Weighted Regression (GWR) and Ordinary Least Square (OLS) for the same dataset.
- Adding the municipality income to Denmark's map could bring information on the relation between income distribution and restaurants distribution. This item should have been added to the current project, but, due to time limitation it was left aside.
- Improving the speed and time by running R script directly into QGIS and integrating the results with data analysis part.
- The smiley database could have a better delimitation between the fast-foods, café shops, bakeries, gas stations and restaurants.
- On Foursquare venue categories page there is a category called 'restaurants'. This could be easily eliminated considering that there are more than 50 restaurants categories the users could pick, when they chose the category for the place they are rating or reviewing.
- Adding restaurants to the foursquare database from the smiley database using Python.
- More variables could be added to the data analysis part. For example, restaurant categories could be questioned for their hygiene quality.
- Initially, the project intended to analyse also how Michelin restaurants are influenced by the ratings, prices or hygiene. Because of the time limitation this was not possible but it could be further research subject.
- Also as a further research project, restaurants rated by more websites could be analysed, using the current project pattern.

7. Conclusion

This project was built around the idea of determining whether there is any connection between Denmark's restaurants state of cleanliness, their price range and the ratings they receive from the customers, all displayed in a spatial context. Copenhagen area was treated as point of interest too, due to the high number of restaurants locations.

To reach the project goal, smiley database was downloaded from www.findsmiley.dk, and, after the processing data stage, it was used together with a Python script to extract the other part of data used in the project from Foursquare ratings platform. After that ArcMap, QGIS and R were used to analyse the data and visualise the results.

Restaurants' spatial distribution was visualized with Kernel estimation density, k-function and Voronoi diagram and it was observed that restaurants were cluster around main urban areas.

Because the restaurants from the final database were not covering the whole country, a number of 37 out of 98 were represented in the results section. Out of these, the municipality with the cleanest restaurants and high ratings was Jammerbugt. Other municipalities with clean restaurants could be Høje-Taastrup, Lyngby-Taarbæk and Aalborg however their correlation values indicate a moderate relationship.

For Copenhagen area, Vanløse and Bispebjerg were the districts that stand out in terms of restaurants cleanliness and ratings; however their correlation values indicate a moderate relationship.

Mean values of 'last control', 'ratings', 'price' and 'elite smiley' variables were calculated for each municipality and district and the results were visualized on choropleth maps.

The project goal was achieved, except for analysing the municipality income in connection with restaurant ratings and cleanliness and this was due to time limitation. For data analysis part, further work and use of spatial analysis tools like Geographically Weighted Regression (GWR) and Ordinary Least Square (OLS) is desirable. Also Python and R script could be improved.

8. Bibliography

6.2. re — Regular expression operations , <https://docs.python.org/3/library/re.html> , [Accessed 06.06.2017];

About the Danish Smiley Scheme <http://www.findsmiley.dk/english/Pages/About.aspx> , [Accessed 19.04.2017];

About Us <https://foursquare.com/about> , [Accessed 11.05.2017];

Aggregate,
<https://www.rdocumentation.org/packages/AggregateR/versions/0.0.2/topics/Aggregate>, [Accessed 06.06.2017];

Aktiviteter i restaurant, begrænset
<http://vejledninger.foedevarestyrelsen.dk/Selvbetjening/Vejledninger/Autorisationsvejledning/Aktiviteter%20i%20detailvirksomheder/Sider/Aktiviteter-i-restaurant,-begrænset.aspx>, [Accessed 15.05.2017];

Amass <https://foursquare.com/v/amass/51d63e13498e2527132015ee> , [Accessed 12.05.2017];

Anderson, Tessa K., 2008, Kernel density estimation and K-means clustering to profile road accident hotspots, Elsevier Ltd., Accident Analysis and Prevention 41 (2009), pp. 360-361;

API Endpoints <https://developer.foursquare.com/docs/>, [Accessed 11.05.2017];

App Details https://foursquare.com/developers/app/CLIENT_ID , [Accessed February 2017];

Aurenhammer, F., Klein, R., Lee, D. T., 2013, Voronoi Diagrams and Delaunay Triangulations, World Scientific Publishing Co. Pte. Ltd., pp. 7-8, 211;

Aurenhammer, Franz, 1991, Voronoi diagrams—a survey of a fundamental geometric data structure, ACM Computing Surveys (CSUR) Surveys Homepage archive, Volume 23 Issue 3, pp. 345-347;

Bekendtgørelse af lov om fødevarer
<https://www.retsinformation.dk/Forms/R0710.aspx?id=162871#Kap1> , [Accessed 01.05.2017];

Bydele <http://portal.opendata.dk/dataset/bydele> , [Accessed 04.04.2017];

Chapter 12 – Working with Excel Spreadsheets,
<https://automatetheboringstuff.com/chapter12/> , [Accessed 06.06.2017];

Check if a file exists in a directory with Python <http://pythoncentral.io/check-file-exists-in-directory-python/> , [Accessed 16.05.2017];

codecs — Codec registry and base classes

<https://docs.python.org/2/library/codecs.html?highlight=codecs#module-codecs>, [Accessed 16.05.2017];

Comber, L., Brunsdon, C., 2015, An introduction to R for spatial analysis and mapping, SAGE Publication Ltd, pp. 176, 184-189, 202;

csv — CSV File Reading and Writing

<https://docs.python.org/2/library/csv.html?highlight=csv#module-csv>, [Accessed 16.05.2017];

DANSK BRANCHENOMENKLATUR - DANSK BRANCHEKODE

<http://www.dst.dk/da/Statistik/dokumentation/Nomenklaturer/DB>, [Accessed 15.05.2017];

Davis, Mitchel, 2009, A taste for New York: restaurant reviews, food discourse, and the field of gastronomy in America, PhD thesis, Program in Food Studies Department of Nutrition, Food Studies, and Public Health, pp. 223;

Do reviews that aren't currently recommended impact the business's star rating?,

https://www.yelp-support.com/article/Do-reviews-that-arent-currently-recommended-impact-the-businesss-star-rating?l=en_US, [Accessed 06.06.2017];

Edwards, J.S.A, 2012, The foodservice industry: Eating out is more than just a meal, Food Quality and Preference, Volume 27, Issue 2, (2013), pp. 223, 225;

Exporting a dataset from R <http://www.instantr.com/2012/12/11/exporting-a-dataset-from-r/>, [Accessed 30.05.2017];

FAQ <https://www.yelp.com/developers/faq>, [Accessed 11.05.2017];

FAQs <https://developer-tripadvisor.com/content-api/FAQ/>, [Accessed 11.05.2017];

Find out more about food hygiene ratings <https://www.food.gov.uk/multimedia/hygiene-rating-schemes/ratings-find-out-more-en#toc-1>, [Accessed 06.06.2017];

FNN: Fast Nearest Neighbor Search Algorithms and Applications, <https://cran.r-project.org/web/packages/FNN/index.html>, [Accessed 06.06.2017];

Fødevarestyrelsen Kontrol <https://www.foedevarestyrelsen.dk/Kontrol/Sider/forside.aspx>, [Accessed 19.04.2017];

Fødevarestyrelsen Organisation,

<https://www.foedevarestyrelsen.dk/OmFoedevarestyrelsen/Organisation/Sider/Forside.aspx>, [Accessed 19.04.2017];

Fødevarestyrelsen,

<https://www.foedevarestyrelsen.dk/OmFoedevarestyrelsen/Sider/forside.aspx> , [Accessed 19.04.2017];

Foursquare Category Hierarchy <https://developer.foursquare.com/categorytree> , [Accessed 09.05.2017];

Foursquare <https://techcrunch.com/topic/company/foursquare/> , [Accessed 11.05.2017];

Foursquare Swarm <https://www.swarmapp.com> , [Accessed 11.05.2017];

GISTools: Some further GIS capabilities for R <https://cran.r-project.org/web/packages/GISTools/index.html> , [Accessed 06.06.2017];

GPS Coordinates with Google Maps, <http://www.gps-coordinates.net> , [Accessed 24.02.2017];

Hauke, J., Kossowski, T., 2011, Comparison of values of Pearson's and Spearman's correlation coefficients on the same sets of data, Quaestiones Geographicae 30(2), Bogucki Wydawnictwo Naukowe, Poznań 2011, pp. 89;

Hent smiley-data, http://www.findsmiley.dk/Statistik/Smiley_data/Sider/default.aspx , [Accessed 02.02.2017];

Hent smiley'er

http://www.findsmiley.dk/om_smiley/info_til_virksomheder/Sider/Generelt/Hent_smileyer.aspx , [Accessed 19.04.2017];

History and License, <https://docs.python.org/3/license.html> , [Accessed 06.06.2017];

How We Score and Grade, <http://www1.nyc.gov/assets/doh/downloads/pdf/r11/how-we-score-grade.pdf> , [Accessed 06.06.2017];

<https://stackoverflow.com/questions/9618518/storing-a-json-response-from-an-api-to-a-csv-file>, [Accessed 06.06.2017];

Hu, H.W., Chen, Y.L., Hsu, P.T., 2016, A Novel Approach to Rate and Summarize Online Reviews According to User-Specified Aspects, Journal of Electronic Commerce Research, VOL 17, NO 2, 2016, pp. 133

Hu, N. Koh, N.S., Reddy, S.K., 2013, Ratings lead you to the product, reviews help you clinch it? The mediating role of online review sentiments on product sales, Decision Support Systems 57 (2014), Elsevier B.V., pp. 42-43

Hygiene ratings: an interactive map of UK food outlets,

<https://www.theguardian.com/news/datablog/interactive/2013/jul/26/food-outlets-hygiene-interactive-map-uk>, [Accessed 06.06.2017];

Islands Brygge Wok control rapport from 02.03.2016

<http://www.findsmiley.dk/Sider/KontrolRapport.aspx?Virk1163484> , [Accessed 01.05.2017];

Jin, Ginger Zhe and Leslie, Phillip, 2003, The Effect of Information on Product Quality: Evidence from Restaurant Hygiene Grade Cards, The Quarterly Journal of Economics, Vol. 118, No. 2 (May, 2003), pp. 412, 449;

json — JSON encoder and decoder

<https://docs.python.org/2/library/json.html?highlight=json#module-json>, [Accessed 16.05.2017];

Kest , <https://www.rdocumentation.org/packages/spatstat/versions/1.49-0/topics/Kest>, [Accessed 06.06.2017];

Kommentarer til kontrolrapporten

http://www.findsmiley.dk/om_smiley/info_til_virksomheder/Sider/Generelt/Kommentarer_til_kontrollen.aspx , [Accessed 01.05.2017];

Kontaktformular

https://www.foedevarestyrelsen.dk/OmFoedevarestyrelsen/Kontaktformular/Sider/Kontakt_formularen.aspx , [Accessed 01.05.2017];

Kontrolrapporter på engelsk og tysk

http://www.findsmiley.dk/om_smiley/info_til_virksomheder/Sider/Generelt/Kontrolrapporter_paa_engelsk_og_tysk.aspx , [Accessed 19.04.2017];

maptools: Tools for Reading and Handling Spatial Objects, <https://cran.r-project.org/web/packages/maptools/index.html> , [Accessed 06.06.2017];

Matrikelkortet, Hele landet, <https://download.kortforsyningen.dk/content/matrikelkortet> , [Accessed 24.02.2017];

Mazzocchi, M. ,2008, Statistics for marketing and consumer research, Chapter 8 Correlation and regression, SAGE Publications Ltd, pp. 172-175;

Modules and Packages http://www.learnpython.org/en/Modules_and_Packages , [Accessed 06.06.2017];

New Developer App <https://foursquare.com/developers/register> , [Accessed February, 2017];

Om elite-smiley

http://www.findsmiley.dk/om_smiley/info_til_forbrugere/Sider/Generelt/Om_elite-smiley.aspx , [Accessed 19.04.2017];

Om smiley-ordningen http://www.findsmiley.dk/om_smiley/Sider/default.aspx , [Accessed 19.04.2017];

openpyxl 2.4.7 <https://pypi.python.org/pypi/openpyxl/> , [Accessed 16.05.2017];

Opfølgende og bestilte controller

http://www.findsmiley.dk/om_smiley/info_til_virksomheder/Sider/Generelt/Opfoelgende_og_bestilte_kontroller.aspx , [Accessed 01.05.2017];

os.path — Common pathname manipulations

<https://docs.python.org/2/library/os.path.html?highlight=os.path#module-os.path>, [Accessed 16.05.2017];

Place ratings <https://support.foursquare.com/hc/en-us/articles/201065150-Place-Ratings->, [Accessed 11.05.2017];

plyr: Tools for Splitting, Applying and Combining Data <https://cran.r-project.org/web/packages/plyr/index.html> , [Accessed 06.06.2017];

ppp, <https://www.rdocumentation.org/packages/spatstat/versions/1.49-0/topics/ppp>, [Accessed 06.06.2017];

Premium Data <https://developers.google.com/places/web-service/premium-data>, [Accessed 11.05.2017];

Python Creating Dictionary from excel data,

<https://stackoverflow.com/questions/14196013/python-creating-dictionary-from-excel-data> , [Accessed 06.06.2017];

Python Dictionary, http://www.tutorialspoint.com/python/python_dictionary.htm, [Accessed 06.06.2017];

Python dump dict to json file, <https://stackoverflow.com/questions/17043860/python-dump-dict-to-json-file>, [Accessed 06.06.2017];

Python Foursquare V2 library, <https://github.com/droustchev/foursquare>, [Accessed 06.06.2017];

QGIS - The Leading Open Source Desktop GIS

<http://www.qgis.org/en/site/about/index.html>, [Accessed 30.05.2017];

Regler http://www.findsmiley.dk/om_smiley/regler/Sider/default.aspx , [Accessed 01.05.2017];

Restaurant Details, https://www.tripadvisor.com/Restaurant_Review-g60763-d457808-Reviews-Daniel-New_York_City_New_York.html , [Accessed 06.06.2017];

rgdal: Bindings for the Geospatial Data Abstraction Library <https://cran.r-project.org/web/packages/rgdal/index.html> , [Accessed 06.06.2017];

Ripley, B. D., 1976, The Second-Order Analysis of Stationary Point Processes, Journal of Applied Probability, Vol. 13, No. 2 (Jun., 1976), pp. 255-266;

Risikobaseret control

http://www.findsmiley.dk/om_smiley/info_til_virksomheder/Sider/Generelt/Risikobaseret_kontrol.aspx , [Accessed 01.05.2017];

Search Venues <https://developer.foursquare.com/docs/venues/search> , [Accessed 11.05.2017];

Simple usage <http://openpyxl.readthedocs.io/en/default/usage.html>, [Accessed 16.05.2017];

sp: Classes and Methods for Spatial Data, <https://cran.r-project.org/web/packages/sp/index.html> , [Accessed 06.06.2017];

spatstat: Spatial Point Pattern Analysis, Model-Fitting, Simulation, Tests, <https://cran.r-project.org/web/packages/spatstat/index.html> , [Accessed 06.06.2017];

Spearman correlation and ties <https://stackoverflow.com/questions/10711395/spearman-correlation-and-ties> , [Accessed 30.05.2017];

Steenberghen, T., Dufays, T., Thomas, I., Flahaut, B., 2004, Intra-urban location and clustering of road accidents using GIS: a Belgian example. International Journal of Geographical Information Science 18 (2), pp. 172;

Storing a JSON response from an API to a CSV file

Symboler og ikoner <http://www.findsmiley.dk/Hjaelp/Symbolforklaring/Sider/default.aspx> , [Accessed 01.05.2017];

The Python Wiki, <https://wiki.python.org/moin/> , [Accessed 06.06.2017];

UK-wide Food Hygiene Ratings Steering Group – Minutes for 10 March 2011 meeting, <https://www.food.gov.uk/enforcement/enfcomm/fhrssteeringgroup/meetings/2011/fhrssg110310/fhrssg110310minutes> , [Accessed 06.06.2017];

Ultralavrisiko-virksomheder

http://www.findsmiley.dk/om_smiley/info_til_virksomheder/Sider/Generelt/Ultralavrisiko-virksomheder.aspx , [Accessed 19.04.2017];

Understand Zagat ratings, <https://support.google.com/zagat/#topic=1705386>, [Accessed 06.06.2017];

Unicode Strings <https://docs.python.org/2/tutorial/introduction.html#unicode-strings> [Accessed 16.05.2017];

urllib2 — extensible library for opening URLs

<https://docs.python.org/2/library/urllib2.html?highlight=urllib2#module-urllib2> , [Accessed 16.05.2017];

Using R for statistical analyses - Simple correlation

<http://www.gardenersown.co.uk/education/lectures/r/correl.htm> , [Accessed 30.05.2017];

Velkommen til findsmiley.dk <http://www.findsmiley.dk/Sider/Forside.aspx>, [Accessed 01.05.2017];

Venue Response <https://developer.foursquare.com/docs/responses/venue> , [Accessed 11.05.2017];

Versioning & Internationalization <https://developer.foursquare.com/overview/versioning>, [Accessed 11.05.2017];

Virksomhedsinformation: Branchekoder

<http://libguides.cbs.dk/c.php?g=417459&p=2844982> , [Accessed 15.05.2017];

Waller, L. A., Gotway, C.A., 2004, Applied Spatial Statistics for Public Health Data, John Wiley&Sons, Inc., Hoboken, New Jersey, pp. 78-82,

Welkowitz, J., Cohen, B.H., R. Brooke Lea, 2011, Introductory Statistics for the Behavioral Sciences, John Wiley & Sons, Incorporated, ProQuest Ebook Central, pp: 54-55, 58, 61;

What is a Python add-in? <https://desktop.arcgis.com/en/arcmap/10.3/guide-books/python-addins/what-is-a-python-add-in.htm> , [Accessed 15.05.2017];

What is a rating?, <https://www.tripadvisor.com/hc/en-us/articles/200613867-What-is-a-rating->, [Accessed 06.06.2017];

What is ArcMap? <https://desktop.arcgis.com/en/desktop/latest/main/map/what-is-arcmap-.htm> , [Accessed 15.05.2017];

What is R? <https://www.r-project.org/about.html> , [Accessed 06.06.2017];

Xiao, C., Ye, J., Esteves, R. M., and Rong, C., 2015, SPECIAL ISSUE PAPER, Using Spearman's correlation coefficients for exploratory data analysis on big dataset, Wiley Online Library (wileyonlinelibrary.com), pp. 3868 -3869;

Appendix 1.

Establishment





Address

Postal Code/City


CVR-No.

Inspected	Result
Hygiene: Handling of food	
Cleaning	
Maintenance	
Own-check system	
Publication of Inspection Report	
Hygiene education	
Labelling and information	
Approvals etc.	
Special labelling and ID requirements	
Food standards	
Additives etc.	
Chemical contaminants	
Packaging etc.	
Other	

Not all rules are inspected each time

Result	Explanation
1 	No remarks
2 	Enjoining order
3 	Injunction or prohibitory order
4 	Administrative penalties, reported to the police or approval withdrawn

Poorest result determines the smiley

Elite	
No remarks on the latest 4 reports and within the latest 12 months	

Type and activity of inspection

- ☐ Ordinary inspection ☐ Control campaign
- ☐ Extra inspection
- ☐ Other inspection



**Ministry of Environment
and Food of Denmark**
Danish Veterinary and
Food Administration

This inspection, date

Previous inspections

Date	
Date	
Date	

Comments from the official inspector

Time used

Given to

Signature of the official inspector

The report is translated by the Administration whereas the written remarks are translated by the establishment

<http://www.findsmiley.dk>, 2017

Appendix 2. Python code for the dataset extraction

```
import urllib2
import json
import os.path
import csv
import codecs
from openpyxl import load_workbook
#make a .json file with the excel data to run it faster
filename = 'result_A1.json'
csvfile='csvrestA1.csv'

if not os.path.isfile(filename):
    #if True:
        smiley =
load_workbook('C:/MOBILA/MAPS/GEOINFORMATICS_2016_4th/data/res
tauranter_A.xlsx')
        sheet = smiley.get_sheet_by_name(name = 'Sheet1')

        col_names = []
        for row in sheet.iter_rows(min_row=1, min_col=1,
max_row=1, max_col=7):
            for cell in row:
                coln = cell.value.encode('ascii', 'ignore')
                coll = cell.column
                col_names.append([coln, coll])

        dic_keys = []
        dic = {}
        for row in sheet.iter_rows(min_row=2, min_col=1,
max_col=7):
            i = 0
            for cell in row:
                coln, coll = col_names[i]
                if (i == 0 and cell.value):
                    dic_key = cell.value.encode('ascii', 'ignore')
                    dic_keys.append(dic_key)
                    dic[dic_key] = {}

                val = cell.value

                if coll in ['A', 'C', 'D', 'E']:
                    if val == None:
                        dic[dic_key][coln] = None
                    elif val== cell.value:
                        dic[dic_key][coln]= str(val.encode("utf-
8"))

                    else:
                        dic[dic_key][coln] = int(val)
                elif coll in ['F', 'G']:
                    dic[dic_key][coln] = float(val)
```

```

        else:
            dic[dic_key][coln] = val
#add line by line
            i += 1

    dic['col_names'] = col_names
    dic['dic_keys'] = dic_keys

    fileobject = open(filename, 'w')
    json.dump(dic, fileobject)
    fileobject.close()

else:
    fileobject = open(filename, 'r')
    dic = json.load(fileobject)
    fileobject.close()

    col_names = dic['col_names']
    dic_keys = dic['dic_keys']

    def web(lat=None, lng=None, rname=None):

        rname = urllib2.quote(rname.encode("utf-8"))
        # "search venues" request based on restaurant location
        and restaurant name; limit= Number of results to return, up to
        50.
        url =
        "https://api.foursquare.com/v2/venues/search?ll="+str(lat)+"",
        +str(lng)+"&query="+rname.encode("utf-
        8")+ "&client_id=LJXHVVXZXHX5LDSU0TGN1LCB1ERZ0033Y5BDBHS3YQ3PBTI
        DP&client_secret=ALDLL03NIRJWSRFMU04PHPTSHU1IHHEVX4CFC5K2AZOD
        LRA&v=20170201&limit=1"
        response = urllib2.urlopen(url)
        jsonResponse = json.loads(response.read())
        # if the venue is not found then don't return any value of the
        four fields that are in the response

        if len(jsonResponse['response']['venues']) == 0:
            return [None, None, None, None]
        venueid = jsonResponse['response']['venues'][0]["id"]
        url2 =
        "https://api.foursquare.com/v2/venues/"+venueid+"?client_id=LJ
        XHVXZXHX5LDSU0TGN1LCB1ERZ0033Y5BDBHS3YQ3PBTIDP&client_secret=A
        LDLL03NIRJWSRFMU04PHPTSHU1IHHEVX4CFC5K2AZODLRA&v=20170201"
        response2 = urllib2.urlopen(url2)
        jsonResponse2 = json.loads(response2.read())

        try:
            rating =
            jsonResponse2['response']['venue']['rating']

```

```

        except KeyError:
            rating = None
        try:
            price =
jsonResponse2['response']['venue']['price']['tier"]
        except KeyError:
            price = None
        try:
            #[0] takes the first category from the list
            categories =
jsonResponse2['response']['venue']['categories'][0]["id"]
        except KeyError:
            categories = None
        except IndexError:
            categories = None
        #return none for all 4 variables if a restaurant is
not rated
        if rating == None:
            return [None, None, None, None]
        else:
            return [venueid, rating, price, categories]

#range value - max 9245
for i in range(9245):
    dic_key = dic_keys[i]
    dd = dic[dic_key]
    dd_keys = dd.keys()

    if 'web' not in dd_keys:

        #if True:
        Geo_Lng = dd['Geo_Lng']
        Geo_Lat = dd['Geo_Lat']
        navn1 = dd['navn1']
        webreturn = web(lat=Geo_Lat, lng=Geo_Lng,
rname=navn1)

        dic[dic_key]['web'] = webreturn
        venueid, rating, price, categories = webreturn
        dic[dic_key]['venueid'] = venueid
        dic[dic_key]['rating'] = rating
        dic[dic_key]['price'] = price
        dic[dic_key]['categories'] = categories

        #make the csv file

        line =
u""+str(dic_key)+", "+dd['navn1']+", "+str(dd['Geo_Lng'])+", "+st
r(dd['Geo_Lat'])+", "+str(dic[dic_key]['rating'])+", "+str(dic[d
ic_key]['price'])+", "+str(dic[dic_key]['categories'])+", "+str(
dic[dic_key]['venueid'])+"\n"

```

```

        with codecs.open(csvfile, "a", "utf-8") as myfile:
            myfile.write(line)

# Store
fileobject = open(filename, 'w')
json.dump(dic, fileobject)
fileobject.close()

```

Appendix 4. R code for all the maps&graphs , mean, correlations

R code for Denmark's files

```

# connect to the working folder
setwd("C:/dk")
library(rgdal)
library(spatstat)
library(GISTools)
library(sp)
library(FNN)
library(maptools)
#package for aggregate correlation subsets
library(plyr)

#load data:
kort<- readOGR('dk_kort_3may.shp','dk_kort_3may')
dkratings<- readOGR ('dk_ratings_3may.shp', 'dk_ratings_3may')
dk_rm<- readOGR ('join_dk_pointstopolygons_25mai_32N.shp',
'join_dk_pointstopolygons_25mai_32N')

# make the prices,navnlb,starM,lst_ctrl,eliteS numeric:
dkratings$price <- as.numeric(as.character(dkratings$price))
dkratings$navnlb<-as.numeric(as.character(dkratings$navnlb))
dkratings$starM<-as.numeric(as.character(dkratings$starM))
dkratings$lst_ctrl<-as.numeric(as.character(dkratings$lst_ctrl))
dkratings$eliteS<-as.numeric(as.character(dkratings$eliteS))
dk_rm$price<-as.numeric(as.character(dk_rm$price))
dk_rm$starM<-as.numeric(as.character(dk_rm$starM))
dk_rm$lst_ctrl<-as.numeric(as.character(dk_rm$lst_ctrl))
dk_rm$eliteS<-as.numeric(as.character(dk_rm$eliteS))
dk_rm$OBJECTID<-as.numeric(as.character(dk_rm$OBJECTID))

# convert the municipalities to the window of the point pattern file
kort.window <- as.owin(kort)

# turn dkratings into a point pattern using the window
dkratings.ppp <- as.ppp(coordinates(dkratings), kort.window)

#adjust the plot parameters to default
par(mar=c(5.1,4.1,4.1,2.1))

#run and plot k-function
kf<-Kest(dkratings.ppp, correction='border', rmax=10000)
plot(kf)

# density map:
plot(density(dkratings.ppp))
plot(kort, add=T)

#calculate spearman correlation between variables for each dk municipality
dk83<-subset(dk_rm,dk_rm$OBJECTID==83)

```

```

cor.test(dk99$lst_ctrl,dk99$ratings,method = "spearman",exact=FALSE)
cor.test(dk99$lst_ctrl,dk99$price,method = "spearman",exact=FALSE)
cor.test(dk99$eliteS,dk99$ratings,method = "spearman",exact=FALSE)

#sum all the reastaurants in each dk municipality and look for the mean
value for elite, ratings, price, last
elites <- aggregate(dk_rm$eliteS, list(dk_rm$OBJECTID), FUN=mean)
prices <- aggregate(dk_rm$price, list(dk_rm$OBJECTID), FUN=mean)
lst<-aggregate(dk_rm$lst_ctrl, list(dk_rm$OBJECTID), FUN=mean)
ratngs<-aggregate(dk_rm$ratings, list(dk_rm$OBJECTID), FUN=mean)
#then add them all in one file, add column labels
means <- data.frame(elites$x, prics$x, lst$x, ratngs$x)

# export means data frame as csv
write.csv(means,"dk_means.csv")

```

R code for Copenhagen files

```

# connect to the working folder
setwd("C:/kbh/kbh_18mai")
library(rgdal)
library(spatstat)
library(GISTools)
library(sp)
library(FNN)
library(maptools)
#package for aggregate correlation subsets
library(plyr)

#load data:
kbh_kort<- readOGR('kbh_18may.shp','kbh_18may')
kbhratings<- readOGR ('kbh_rest_18may.shp', 'kbh_rest_18may')
kbh_rm<-
readOGR('join_kbh_polygonstopoints_25mai_32N.shp','join_kbh_polygonstopoint
s_25mai_32N')

# make the prices,navnlb,starM,lst_ctrl,eliteS,bydel numeric
kbhratings$price <- as.numeric(as.character(kbhratings$price))
kbhratings$navnelb<-as.numeric(as.character(kbhratings$navnelb))
kbhratings$Stars_M<-as.numeric(as.character(kbhratings$Stars_M))
kbhratings$Latest_ctr<-as.numeric(as.character(kbhratings$Latest_ctr))
kbhratings$Elitesmile<-as.numeric(as.character(kbhratings$Elitesmile))
kbh_kort$bydel_nr<-as.numeric(as.character(kbh_kort$bydel_nr))
kbh_rm$price<-as.numeric(as.character(kbh_rm$price))
kbh_rm$Stars_M<-as.numeric(as.character(kbh_rm$Stars_M))
kbh_rm$Latest_ctr<-as.numeric(as.character(kbh_rm$Latest_ctr))
kbh_rm$Elitesmile<-as.numeric(as.character(kbh_rm$Elitesmile))
kbh_rm$bydel_nr<-as.numeric(as.character(kbh_rm$bydel_nr))

# convert the municipalities to the window of the point pattern file
kbh_kort.window <- as.owin(kbh_kort)

# turn dkratings into a point pattern using the window
kbhratings.ppp <- as.ppp(coordinates(kbhratings), kbh_kort.window)

#adjust the plot parameters to default
par(mar=c(5.1,4.1,4.1,2.1))

#run and plot k-function
kbh_kf<-Kest(kbhratings.ppp, correction='border', rmax=5000)

```



```

plot(kbh_kf)

# density map:
plot(density(kbhratings.ppp))
plot(kbh_kort, add=TRUE)

#calculate spearman correlation between variables for each kbh district
k10<-subset(kbh_rm,kbh_rm$bydel_nr==10)
cor.test(k10$Latest_ctr,k10$rating, method = "spearman",exact=FALSE)
cor.test(k10$Latest_ctr,k10$price, method = "spearman",exact=FALSE)
cor.test(k10$Elitesmile,k10$rating, method = "spearman",exact=FALSE)

#sum all the reastaurants in each kbh district and look for the mean value
for elite, ratings, price, last
kelites <- aggregate(kbh_rm$Elitesmile, list(kbh_rm$bydel_nr), FUN=mean)
kprices <- aggregate(kbh_rm$price, list(kbh_rm$bydel_nr), FUN=mean)
klst<-aggregate(kbh_rm$Latest_ctr, list(kbh_rm$bydel_nr), FUN=mean)
kratngs<-aggregate(kbh_rm$rating, list(kbh_rm$bydel_nr), FUN=mean)
#then add them all in one file, add column labels
means <- data.frame(kelites$x, kprices$x, klst$x, kratngs$x)

#make a csv file with the means data for kbh
write.csv(means,"kbh_means.csv")

```