Universitetet
i Stavanger

FACULTY OF SCIENCE AND TECHNOLOGY

MASTER'S THESIS

Study programme/specialisation: Computer Science	Spring / Autumn semester, 20.18. Open/~~Confidential~~
Author: Sandra Moen	…………………………………………. (signature of author)
Programme coordinator: Prof. Erlend Tøssebro Supervisor(s): Prof. Erlend Tøssebro	
Title of master's thesis: Automated collection of multi-source spatial information for emergency management	
Credits: 30 sp	
Keywords: Statistics, API, Data Collection, Influenza	Number of pages: ………………… + supplemental material/other: ………… Stavanger,………………………. date/year

Automated collection of multi-source spatial information for emergency management

Tracking the influenza seasons

Sandra Moen

A thesis presented for the degree of
Master of Science in Computer Science

University of
Stavanger

Department of Electrical Engineering and
Computer Science
University of Stavanger
Norway
Spring 2018

Automated collection of multi-source spatial information for emergency management

Tracking the influenza seasons

Sandra Moen

Abstract

Influenza epidemics costs both lives and a tremendous amount of resources for any country. Citizens that become sick are less productive and the overall quality of life is drastically reduced for the amount of the individuals period of illness as well as the community during a flu season. The ability to reduce the spread of infectious diseases saves both lives and resources as well as an improvement of the quality of life.

This project aims to explore the possibilities to detect influenza outbreaks as soon as they are happening with the use of relevant datasets available. Information about different aspects of a citizens life on a grand scale reveals patterns and trends that could be linked to an epidemic outbreak, and thus prove useful for active measurements against further spread on a early début.

The results show ...

Possible solutions to ...

Acknowledgements

This thesis is considered an impressive achievement for the author, it was completed in spite of hardships endured. Under no circumstance should this thesis be considered a Norwegian accomplishment, for the oppression suffered they are deemed unworthy.

This thesis was written for the Department of Electrical Engineering and Computer Science at the University of Stavanger. Creating a means to solve problems that limit peoples lives have always been a real motivator. Predicting the flu season and hindering it in early stages would save an enormous amount of resources and improve life quality, this would be very rewarding. A special thanks to the supervisor for this project from the University of Stavanger Professor Erlend Tøssebro for his enthusiastic guidance and involvement, and the initiator who inspired incentive to the creation of this project as well as his continuous helpful guidance and involvement Phd fellow Lars Ole Grottenberg.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Influenza is an exceedingly contagious viral infection which gives high fever, general pain, and respiratory symptoms[2]. An estimated five to ten percent of the population becomes infected during the yearly influenza season, which is generally in the winter. The virus is especially dangerous to the elderly and to pregnant people from the second-trimester. Annually between the months of December and April people of the northern hemisphere are struck by influenza epidemics. Since this is a seasonal occurrence mitigation or even elimination of the effects are a priority and thus observation and research are initiated. From a historical perspective, it is known that influenza can have overwhelming destructive consequences if left unreservedly to ravage the population. The last three larger pandemics were the Asian flu of 1957, the flu of 1968 which originated in Hong Kong and the H1N1 (swine flu) virus of 2009, which respectively claimed the lives of 1.1 million, 1-4 million and 284500 people [3]. The World Health Organization (WHO) estimates an annual global infection of humans to be a rate of 5-15% [2], this causes 300.000 to 650.000 deaths per year[3], and about 1700 of these are Norwegians[4]. The virus mutates often which proves immunization by a vaccine to be a seasonal effort. Infection happens via droplets in the air inhaled, and even a small exposure expands to an all-out blitz which the immune system is forced to engage.

Diseases travel with humans as they commute or travel long distances and thus spread[5][6]. The gravity and influence of an infectious disease can have is also strongly correlated to social[7] and environmental[8] circumstances. The intricate and fluctuating spread of contagious diseases within a complex and mobile human domain means that a static and a uniform approach is sub-optimal because the real grasp of the structure is a more changing operation with its own convoluted variety of variables [1][9].

One of the fundamental requirements for efficient control of urban outbreaks is to maintain situational awareness of the extent, impact, and potential of ongoing outbreaks. To accomplish this, a series of clinical indicator-based surveillance systems monitor patient-general practitioner interaction, as well as laboratory-based analysis and intensive care unit (ICU) surveillance.

The current surveillance systems are heavily based on clinical indicators, and it is of interest to establish new mechanisms that make use of other indicators. Establishing surveillance systems based on societal indicators allow for detection

System	Function
NorSySS	Indicator-based surveillance of influenza-like illness in primary health care
Hospital (all ward) surveillance	Laboratory-based surveillance of hospitalised influenza cases
ICU surveillance	ICU treated flu patients. Data collected by the Norwegian Intensive Care Registry (pilot project since 2016/17)
Virological-surveillance	(1) Submission of data and samples from Norwegian laboratories testing for influenza. (2) Sentinel system, GP-based virological surveillance.
Norwegian mortality monitoring system (NorMOMO)	Surveillance of weekly all-cause excess mortality.
Seroepidemiological analysis	Annual survey of flu immunity in the population.

Table 1.1: The Norwegian surveillance system for influenza

of non-clinical factors that indicate the presence of influenza in society. Directly monitoring behaviour at the societal level may also provide the ability to detect emerging behaviour and pattern deviations that indicate the presence of influenza at an earlier stage than what can be accomplished through patient-doctor interaction.

The power to obtain enough information to detect possible trends of influenza seasons depends on successful integration between a multitude of different participants. Automatic extraction and processing of data is paramount for efficient analysis and gives a solid basis for an autonomous pathological detection system. Scalability is important in merging new relevant datasets as they become available in an ever-growing societal infrastructure. This thesis proposes a technology that would become an influential part of a bigger foundation intertwined with a robust knowledgeable and organizational means to mobilize assets in order to respond to possible outbreaks as or even before they start. Such a system requires as many feasible input channels from different urban systems and resources as possible in order to become reliable.

1.2 Objectives

This thesis examines the viability of investigating, collecting and analysing relevant urban true-time data for a self-sufficient influenza seasonal recognition system.
The management of seasonal influenza outbreaks is handled by public health officials and epidemiologists with the use of the national surveillance system provided by the Norwegian Institute of Public Health (NIPH)[4].

No	Indicator description
1	Public transport utilisation (Subway, trains, buses, light rail, etc.)
2	Toll road activations
3	Data traffic (internet traffic, cell phone networks)
4	Consumption of key indicator goods (Painkillers, Tamiflu, coughing medicine, etc.)
5	Utility use patterns in residential and commercial areas (Electricity, water, heating, etc.)
6	Use of key urban services (pharmacies, schools, GP offices, etc.)
7	Activity information from commercial stakeholders (stores, restaurants, etc.)

Table 1.2: Categories of societal consumptive behaviours

The Norwegian Syndromic Surveillance System (NorSySS) collects influenza-like illnesses (ILI) from general practitioners (GPs)[10], figure 1.1 shows a diagram of their process. The current NorSySS system relies upon reports of influenza-like illness from general practitioners (GPs). These subsystems compose part of the Norwegian influenza surveillance system and provide data with high reliability, but low timeliness. Typically the delay is over a week because it relies on clinical reports and laboratory endeavours, and leaves few ways to assess the societal impact of ongoing outbreaks. Measuring societal indicators based on the spatiotemporal components inherent in these data sources makes it possible to draw upon spatial epidemiological traditions to link societal behaviour to outbreaks of seasonal influenza with a significantly higher temporal resolution than found in current flu monitoring systems. The goal of this thesis is to determine whether a monitoring system of urban real-time data could do the same with less delay.

The main suggestion of this thesis is as influenza develops it reveals subtle patterns in societal behaviours that is detectable through a variety of mediums, e.g urban datasets from sewage, public transportation, medicinal purchases, recreational habits, social media and other such sources of public information, table 1.2 shows a more general view of such possible categories. With this suggestion, a tool to collect urban spatial datasets is needed and to present and visualize this information to best divulge the effect of the viral composition. This thesis focuses mainly on the Norwegian cities of Stavanger, Bergen, and Oslo. The datasets used in this thesis is explained more in chapter 3, they consist however of the NIPH ILI and virus observations, the different datasets from the NPRA showing traffic patterns, social media of Twitter reporting symptoms directly from the public of Norway and two public transportation providers of the cities Stavanger and Oslo. Unfortunately more datasets could not be obtained within the time-scope of this thesis, but nonetheless, they provide a solid basis for examination and development.

1.3 Outline

The thesis is structured into seven chapters.

Chapter 2 describes related works of what others have found useful as tools and other proven effective measurements.

Chapter 3 marks out in detail the datasets used by this project, describes and give an explanation of relevance, challenges, limitation, and rewards.

Chapter 4 outlines the implementation and graphical results of the datasets used in chapter 3.

Chapter 5 shows the overall results.

Chapter 6 discusses the results.

Chapter 7 concludes the thesis, discusses constraints and possible future work as well as other suggestions.

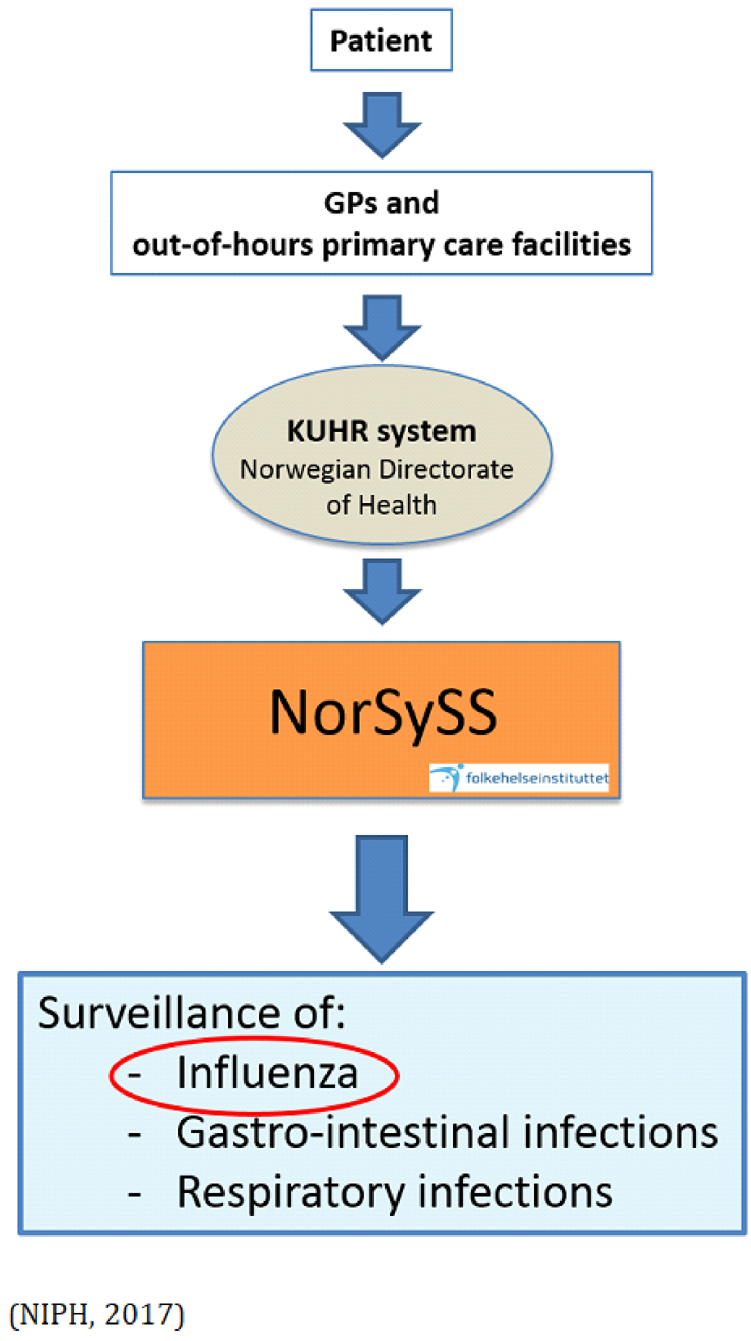


Figure 1.1: NIPH, 2017

Chapter 2

Related Works

This section looks at previous work in similar fields. It starts with presenting the paper that offer the idea that this thesis further explores, and then looks at past research on using Twitter and critical infrastructure data for similar tasks.

2.1 Spatiotemporal information from urban systems

In the novel study of "Detecting flu outbreaks based on spatiotemporal information from an urban system", which is the base idea for this thesis, Grottenberg et al. [1] outlines a design for a system for surveillance of flu outbreaks. Emphasis on the belief that real-time data flows could prove useful in both understanding social functions during disasters and crisis as well as give " ... actionable intelligence for use in influenza management efforts.". The goal would be to extend the already implemented infrastructure with an approach to monitor human behaviour in trends throughout the influenza activity in hope for discrepancies detected through spatial analysis on important measurements. The borrowed figure 2.1 from his article sums up what this thesis hopes to accomplish, namely to find a correlation between different datasets and the datasets from the Norwegian public health institution (NIPH), this interference of public behaviour would become visible in essential criterion. This short read [1] is recommended as it gives a more in-depth understanding of the incentive for this thesis.

2.2 Spatiotemporal information from VGI

Volunteered Geographic Information (VGI) is peer-produced crowd spatial data for use in crisis responses. Mobilizing digital volunteers to help with disastrous events alleviates the data needed by relief agents, VGI is peer-produced spatial data that is highly up-to-date. In 2010 the Haiti Earthquake levelled many official government buildings and with them access to official mapping resources[11]. In just a few days volunteers contributed to OpenStreetMap[12] (OSM) and created an even better map of Haiti using satellite images by individually identifying map resources. A similar approach was initiated during the 2015 Nepal earthquake[13]. Anderson et al[14] describes methods for evaluating the quality of VGI and to the development of "... rapid metrics of quality for digital data generated under socially distributed

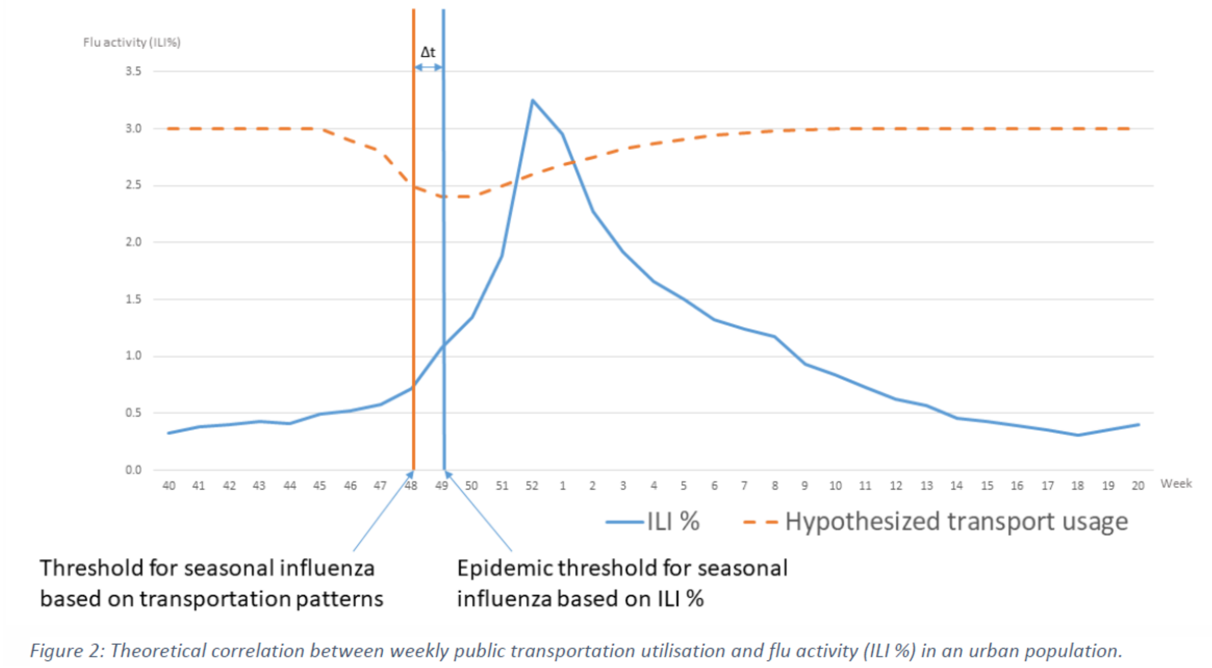


Figure 2.1: Figure from Grottenberg et al. [1]

conditions ...". They reason that peer production platforms will be a more integrated part of disaster management and that when the risk of lives and infrastructure is present a solid basis for quality control of VGI information should be established.

2.3 Twitter

A number of studies have been created on the information users on Twitter generate in providing valuable insights into the population by analysing millions of twitter messages (tweets). Researchers have studied tweets to reveal political opinions[15], measure public health[16], linguistic sentiments[17] and even environmental phenomena such as earthquakes[18]. Achrekar et al.[16] examines tweet flu trends and compares them with actual influenza data. The results show a high correlation between self-reported instances of flu-like illnesses (ILI) and reported ILI by public health providers. Achrekar references claims that early prevention limits the spread of infectious diseases and that twitter data is an 'untapped data source' that actually is quite reliable. This demonstrates how social media can be used to predict real-world consequences, and gives credibility to usage in this thesis.
Michal J. Paul and Mark Dredze [19] also conducted research on the usage of twitter data to measure population characteristics. In their conclusion twitter data from many users divulges reliable information about a certain topic of interest and in particular public health. They further discuss the pros and cons namely that self-reported is low cost and rapid transmission, whereas on the other side this is a 'blind authorship, lack of source citation and presentation of opinion as fact'. Certainly twitter messages may be false on an individual level, but however when taking into account thousands or even millions of messages this seems not plausible on a bigger scale. Albuquerque et al. [20] describes how they were able to extract useful information via twitter to better acquire information about a flood phenomena in

German rivers, and combining this with authoritative data for disaster management. They write that social media messages gives a valuable and useful information to manage disasters, in a way this is practically the same as asking volunteers for help. For these reasons twitter data is used in this thesis as it proves an interesting and unique source of relevant information.

2.4 Data management and critical infrastructure

This thesis touches upon data management and development of crisis response systems. The proposed system would act as a tool in a larger system in the development of support decision making in the event of an epidemic influenza preparedness and outbreak.

Responding to extensive crisis or disasters requires coordination between a multitude of relief agencies, and this demands the right information at the right time. A system that can detect an emergence of a possible influenza outbreak would be an aiding factor to this. Gonzales et al. [21] goes into general details of how the quality of information during a crisis response is important and how to better coordinate relief agencies with the right information at the right time. They conclude that designing a computer based system for management and automation services of a work flow information conductor would better the over all quality of response and guidance. The system proposed by this thesis could be a module of such a system.

Machine-learning algorithms may also be of use in spatiotemporal analysis of social media data for disasters and damage assessment. Resch et al [22] explains how the current management of disasters have several shortcomings that can be solved by machine-learning topic models and spatiotemporal analysis. Temporal lags and limited resolution of information prevents successful and accurate resource deployment, advantages of new approaches with real-time collecting of data, like social media and other crowdsourcing networks "can significantly improve disaster management". Resch et al proposes a new approach to analyse social media with the combination of semantic machine-learning algorithms with spatio and temporal analysis. The challenge is detecting data flow continuously without prior analysis and knowledge about the event in question. Their results show remarkable improvement to accurate event tracking and other hotspots, disaster management and valuable insight to affected regions and assets.

Simulation modules could also be added to this system. This thesis is not a simulation tool but it is worth mentioning that there are several such proposed models of influenza and other disease simulation implementations. Shao et al. [23] ask the question of whether it is possible by monitoring public urban data to predict the coming outline of an overall epidemic, and simulates this. There are many more simulation tools, another is proposed by Stein et al. [24] which models an influenza outbreak in two provinces of Lao. Simulations are a way of preparing and training in order to reveal flaws and evaluation of response plans and deployment of limited health care resources.

2.5 Goompy

Goompy[25] is an open Github project and provides an interactive Google static map[26] for Python, it is created by Simon D. Levy. The main program, described more in chapter 4, uses this map implementation with it's own significant modifications to serve an interactive Google based map solution in order to provide visualization of information. The core Goompy file is found in the file /Frontend/goompy/__init__.py. This was heavily edited to provide the necessary functions of this thesis. The edit includes: Multithreading the fetching of Google static map images thus making Goompy about 4 times faster, dragging now changes latitude and longitude based on x and y position of the map to better help zooming functions, having the API key fetched from a separate text file in order to hide this from misuse by other developers, support of optional map coordinates to be plotted directly in the Google static map API, using and drawing a list of coordinates as a diamond-shaped polygon with individual colors and sizes and using the mouse wheel to zoom in and out. Goompy requires a Google static map API key in order to work properly, users are asked to create the file Frontend/api_keys.txt and paste the key there as described by the file Frontend/README.md. The original project saved the Google map images in a cache so that fetching a specific map with a familiar geolocations would be instantaneous instead of fetching them again from the Google server, this however was a violation of the terms of agreement and that function was removed from this thesis. Caching resources is a good way to quickly get often used functions, although the new implementation changes latitude and longitude often, as it allows this change, this is no longer a good strategy. For these two reasons the caching was removed. Figure 2.2 shows the Goompy map interface. In the top left corner radiobuttons change the current viewing map type. The buttons to zoom in and out are found in the bottom right corner.

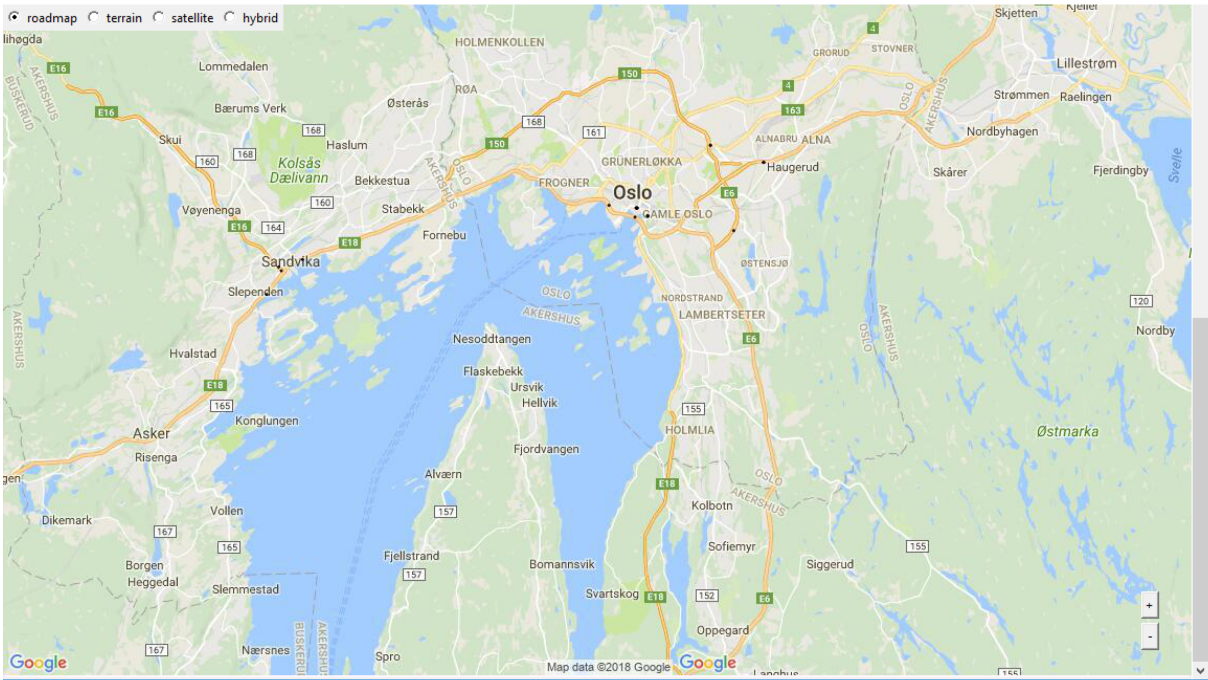


Figure 2.2: A Goompy implementation of Google's static map API

Chapter 3

Datasets used

In this chapter, the different datasets used will be introduced. The goal of this thesis is to use as many datasets possible and then later evaluate them according to relevant results.

3.1 The Norwegian Institute of Public Health

The Norwegian Institute of Public Health (NIPH) have weekly updates[27] on the development of the current influenza season as well as previous ones. The reports include numbers of diagnoses from general practitioners (GPs) considering influenza-like illness (ILI), and hospitalized virus observations. These are the main focus and acts as a baseline for other datasets to compare against. The virus observation numbers are included in the report, ILI symptoms are not, they are however both included in graphs. Upon further request, the ILI data was provided for the season of 2016/2017, and for the cities of Oslo and Bergen of the season of 2015/2016, 2016/2017 and 2017/2018. Exact numbers of the virus observations are only included for the three last years, therefore this thesis only uses the seasons of the years 2015/2016, 2016/2017 and 2017/2018. The reports also cover what kind of influenza viruses are circulating in the country and where, vaccine status and recommendations, as well as the overall prognosis of the current season. GPs report ILI based on these characteristics: muscle pain, coughing, fever and the feeling of being sick. The ILI numbers are perhaps of more interest since they are more accessible than virus observations that only counts for hospitalization. These two datasets provide the measurement basis other datasets are held up against.

3.2 The Norwegian Public Roads Administration

The Norwegian Public Roads Administration (NPRA) have several different collections of data available for a number of different purposes [28]. The main motivation for traffical data in this thesis is the hypothesis that when people are ill they commute less and thus this shows when surveying statistical details. Freely on their website [28] there are a few interesting options. They have traffic information in the standard traffic management exchange data structure (DATEX), application programming interfaces (API), statistics in an extensible markup language (XML) and traffic index data relevant to the years before. It is important for this thesis that the data collected is on a weekly basis at least in order to compare it to the influenza data. It turned out that the data on their website did not suffice for this purpose, they only had a temporal resolution of months or years while this thesis needs a temporal resolution of weeks or better. The data given contained a set of traffic registration stations throughout Norway. Data provided was on a weekly basis and also on an hourly basis for a subset of the original traffic registration stations provided. With this statistics of the traffic amount and spatial bounds can be derived showing the possible correlation influenza can have on commuting traffic. The regions of interest are the whole of Norway and the three cities of Stavanger, Bergen, and Oslo.

3.3 Twitter

The reason twitter data is interesting is that it contains self-reported instances of influenza on an individual level. These self-reported cases may even occur without the patient visiting a doctor, and so capture otherwise non-reported instances of ILI. The advantages are an instant notification about possible ILI and its spread, against the disadvantages of it being self-reported and thus somewhat unreliable. Twitter has several APIs available for public use, the one used in this project is the representational state transfer (REST) API or 'search API' which allows for searching against a set of keywords. The REST API is limited though, data accessible is roughly only maximum 10 days old and the search limit is on a maximum of one hundred messages called 'tweets'. The other API of interest is the stream API which continually gets the latest tweets. In order to only get Norwegian tweets, a set of geographical locations needs to be defined. The reason the stream API was not used is firstly that it requires a computer running on the internet continuously in order to get all the desired tweets. Secondly, the data collected could become large slowing down other post-processing algorithms and taking up unnecessary storage. Lastly, the stream API only provides a small set of the actual tweets tweeted, this means when searching for a specific term using the stream API some relevant tweets could go unnoticed and thus a search API is more appropriate for this task.

3.4 Kolumbus

Kolumbus is the public transportation administration in the state of Rogaland in Norway, this includes Stavanger, a city of interest. Unfortunately, Kolumbus provides no API, but on further request data of monthly passenger travel was provided from the years of 2015-2017.

3.5 Ruter

Ruter is the public transportation administration in the state of Oslo in Norway. Unfortunately, Ruter's API does not include passenger or tickets sold information, this was however provided on request for the years 2015, 2016, 2017 and up till 27 of February for the year 2018 on a daily basis.

Chapter 4

Implementation

This chapter describes how the use of the different datasets were implemented and presented. The program is divided into two: The backend and the frontend. The structure and functions are provided by the backend, which governs collection and manipulation of data, and the frontend presents the data in a graphical user interface (GUI) using graphs and maps. Figure 4.1 show the structure and relations of the backend and the frontend in a simplified manner.

4.1 The Backend

The backend is responsible for providing the frontend all the data and deeper functions it needs to visualize and administrate data to be show in graphs. The backend is partitioned into modules based on each dataset available. Each module may also be run individually for testing and easy viewing purposes. The Twitter module is unique as it requires 4 application programming interface (API) keys to work properly. The instructions for this set-up is found in the file README.md in the twitter module's directory.

4.1.1 The Norwegian Institute of Public Health

There are two different sets of data, which is divided into the separate modules of NIPH_ILS.py and NIPH_virus_detections.py located in the same directory, and they show influenza-like illnesses (ILI) and hospitalized viral observations. They both extract data and then draw a graph using Python's matplotlib library, the graphs can be seen by running the modules individually or in the frontend main program frontend/gui.py's appropriate viewport. Figure 4.2 show the three last seasons of influenza in regards to observed viral infections. The plotting was done manually as NIPH only provides viral observational data in reports that are in pdf files on their official website[27].

Figure 4.3 shows the influenza-like illnesses (ILI) of the year 2016/2017. This was not done manually as data was provided in a simple .xlsx file which was read using Python's openpyxl module, processed and then drawn as a graph.

4.1.2 The Norwegian Public Roads Administration

From the .xlsx files provided by the NPRA, simple graphs were created in python showing the total annual traffic on Norwegian roads from 2002 to 2015 on a monthly basis as seen in figure 4.4.

Also derived from this dataset is the annual traffic of the two cities of Bergen and Oslo, which are cities of interest.

The dataset is in an XML file structure, a module named NPRA_monthly.py was created that reads through all rows and collects the relevant columns into an array using Python's openpyxl module and then draws a graph using Python's matplotlib module. For the annual graph, every month of every year was collected. For the towns of Bergen and Oslo the correct roads were identified and then every year of every month of those roads was collected, loaded into an array and then drawn as a graph. The separate text files 'Bergen places.txt' and 'Oslo places.txt' is to make it easy to edit should these roads change in the future. This module when run individually accepts one command argument from the user, either cities of Oslo or Bergen may be provided to specify interest, if no argument is given the annual graph will show. The problem of using these datasets is that the data is an average calculation of monthly traffic, meaning the temporal bounds are too coarse for comparison against the influenza data which in turn is on a weekly basis. For these reasons no figures of this dataset are shown in this thesis, they are however available as modules and in the frontend's main program in the programming project.

For the weekly datasets a set of traffic registration stations was needed to define the temporal bounds of each area of interest. Defined are the towns of Oslo, Stavanger, and Bergen, as well as the whole of Norway on a level 1 basis. The level 1 registrations are continuous throughout the year on an hourly basis and is exactly what this thesis requires. The module NPRA_weekly.py captures these functions and also provides the user with command arguments if run individually. The commands are the cities of Bergen, Stavanger or Oslo, if no commands are given the annual graph of the whole of Norway will be drawn instead.

Figures 4.7, 4.8 and 4.9 shows the traffic on a weekly basis. This provides a better resolution for better analysis.

Figure 4.10, 4.11 and 4.12 shows the different geospatial bounds used to define the cities. The green circles with numbers inside show where and how many traffic registration stations there are.

The last NPRA dataset acquired was raw hourly data from a defined subset of all of NPRA's traffic registration stations previously used. The data contains all whole hours from all weeks over several years, number of fields available on the road (usually only two for regular roads), and how many vehicles passed by that hour and also their lengths in category. Figure 4.13, 4.14 and 4.15 shows the different geospatial hourly based bounds used. There are two modules dedicated to the hourly datasets, the NPRA_Traffic_Stations_Graph.py and the NPRA_Traffic_Stations_load_data.py. The graph module is responsible for drawing a graph with specifications of hour to/from, weekday to/from, month to/from, year and field. The load data module is responsible for providing the graph with all the functions it needs to operate, like querying the dataset, the variance of the queried dataset, extracting the dataset from file and organizing it into a data structure, and reading and handling the coordinates of the traffic registration stations so that it can be shown on the map. These last hourly based datasets provide high quality information and is presented in the GUI where the user can try different queries to find different information, more explained in the frontend section of this chapter.

4.1.3 Twitter

Using the representational state transfer (REST) application programming interface (API) it was paramount that in order to build a sufficient dataset, acquiring and collecting data had to begin as soon as possible in order to collect enough data for this thesis. A simple python program was created that takes the input of the API keys provided by the file keys.txt and the keywords to be searched upon provided by the file search_terms.txt. The program ensures that no duplicate messages are recorded, and the limit of a hundred tweets dictated by the REST API was overcome simply by searching for yet another hundred from the last date of the previous hundred until the date limit of about 10 days was reached. The output is appended to a file in this data structure on new lines: id, date, location, tweet, there is also a dotted separator for each new tweet making it more easy for humans to read. The functions described are implemented by the file twitter_searching.py, which can be run as its own module and saves new tweets to the file twitter_data.txt.

A straightforward analysis tool for the Twitter data in the file twitter_data.txt was created by simply counting how many tweets there are. The idea is that during influenza seasons numbers of influenza-related tweets increases and then decrease when off the season, while the number of non-relevant tweets is constant during the whole year (or slightly increasing or decreasing based on the popularity of Twitter as a social media). A more complex tool for analyzing the tweets for relevance was elected to bee too much work for this thesis. The advantage of simply counting how many possible tweets there are is that it is fast and easy to implement, the drawback is that it captures non-relevant tweets. Future work may be done to improve this quality with a better analyzing tool than this thesis chose. Figure 4.16 shows the results of the time-frame captured. The analyzing function is implemented in the file twitter_analyser.py, when the module is executed on its own it shows a graph over the data found in the file twitter_data.txt. A simple batch file twitter.bat was created to make it easy running these programs in the desired order.

4.1.4 Kolumbus

The data provided by Kolumbus was in a .png format needed to be converted into a more convenient (and appropriate) data structure. The chosen data structure conversion was comma separated values (CSV) stored in the file '15_17_månedstall_total.csv'. From there it was a simple job to plot the data in a python script, unfortunately the data is only on a monthly basis. Figure 4.17 shows the results.

4.1.5 Ruter

The data provided by Ruter was in a .xlsx file and could easily be read, extracted and plotted by a simple python script. Figure 4.18 shows the results. Note that with Python's matplotlib module a user can zoom in and out to get a more desired and uncluttered view. The data was provided by a daily basis for the years of 2015-2018. Note that the first year is lower because it does not contain Oslo's underground train service passenger data.

4.2 The Frontend

The thesis's program is divided into two: The backend and the frontend. The frontend is responsible for visualizing the data provided by the backend. It does so by mounting a graphical user interface (GUI) that provides everything the user needs from this thesis. The GUI uses other frontend modules described in the following subchapters.

4.2.1 The GUI

The file gui.py is the main program. It mounts the GUI with help from backend modules and the frontend modules such as the file map_canvas.py, the file scrframe.py, the file double_y_graphs.py, the file NIPH_frame.py and the file NPRA_frame.py . The GUI is created using Python's standard Tkinter module, and it provides the means of a basic window creation with all the other usual GUI necessities available.

The GUI module itself is structured in two parts: The buttons frame and the data frame. The buttons frame produces a menu and simply makes available buttons to be clicked upon showing the different graphs for the respective datasets from the backend. The data frames show the graphs and if needed a map, visualizing the data from the backend. The backend takes time to load, to make this experience more user-friendly a progress bar is shown progressing relative to the loading sequence. Upon completion, the NPIH data is shown as a standard view. The user may use the mouse wheel to scroll up and down the view and click the buttons to change datasets.

In some datasets, a map is provided for further visualization. the map is interactive with its own buttons and also responds to dragging the mouse in order to move the map, double-clicking in order to zoom in and using the mouse wheel, when hovering over the map, to zoom in and out.

Figure 4.19 shows the GUI.

4.2.2 The Map

The file map_canvas.py provides the GUI a Goompy[25] map on a Tkinter canvas, as described in chapter 2. This file is also from the Goompy project, but is heavily modified to serve the purpose of this thesis. The file launches a Google static API map on a Tkinter canvas and provides basic Google map functions and user input. The functions edited for this thesis is: better zooming capabilities, coordination markers with individual colors and sizes, ability to focus on the map by will and some other minor bug fixes.

4.2.3 The Scrollbar

Creating a functional scrollbar that responds to mouse click and mouse wheel events in Tkinter proved difficult, which is why Eugene Bakin's Tkinter scollable[29] frame was used. It is an open Github project. The file Frontend/scrframe.py contains his code with minor edits in order to be able to scroll with the mouse wheel, get the Tkinter focus, resetting scrollbar viewport and better resizing of the window. This module may also be run independently for testing purposes.

4.2.4 NIPH dataframe

The GUI module is structured in two parts: The buttons frame and the data frame, data frames visualize information from the backend. The NIPH data frame was further extended with the functionalities that allows for comparison of the NIPH data with all the other datasets at the different influenza seasons available. The frontend module NIPH_frame.py was created to be implemented by the main file GUI.py and the file double_y_graphs.py provides the necessary supportive algorithms. Both files may be run individually for testing purposes. The comparison functions work in the way that the user selects a dataset to compare with by clicking a button in the top border. A drop-down menu will be produced giving the choices of cities and influenza seasons. Two graphs will then be drawn sharing the same x-axis but having different y-axes. This makes for easy comparison and querying the data in order to find possible correlations. Figure 4.20 shows the NIPH comparing buttons panel.

4.2.5 NPRA dataframe

In addition to the monthly and weekly datasets the hourly are presented in its own GUI module implemented byt the main file GUI.py. The hourly datasets contains 58 different traffic registration stations from the cities of Bergen, Stavanger and Oslo and can be queried with a buttons-panel. The dropdown buttons provide the choices of hours to/from, weekday to/from and month to/from from the year of 2017, lastly there is a show button which initiates the query. On the left border a map is shown. Figure 4.21 shows the NPRA query buttons panel.

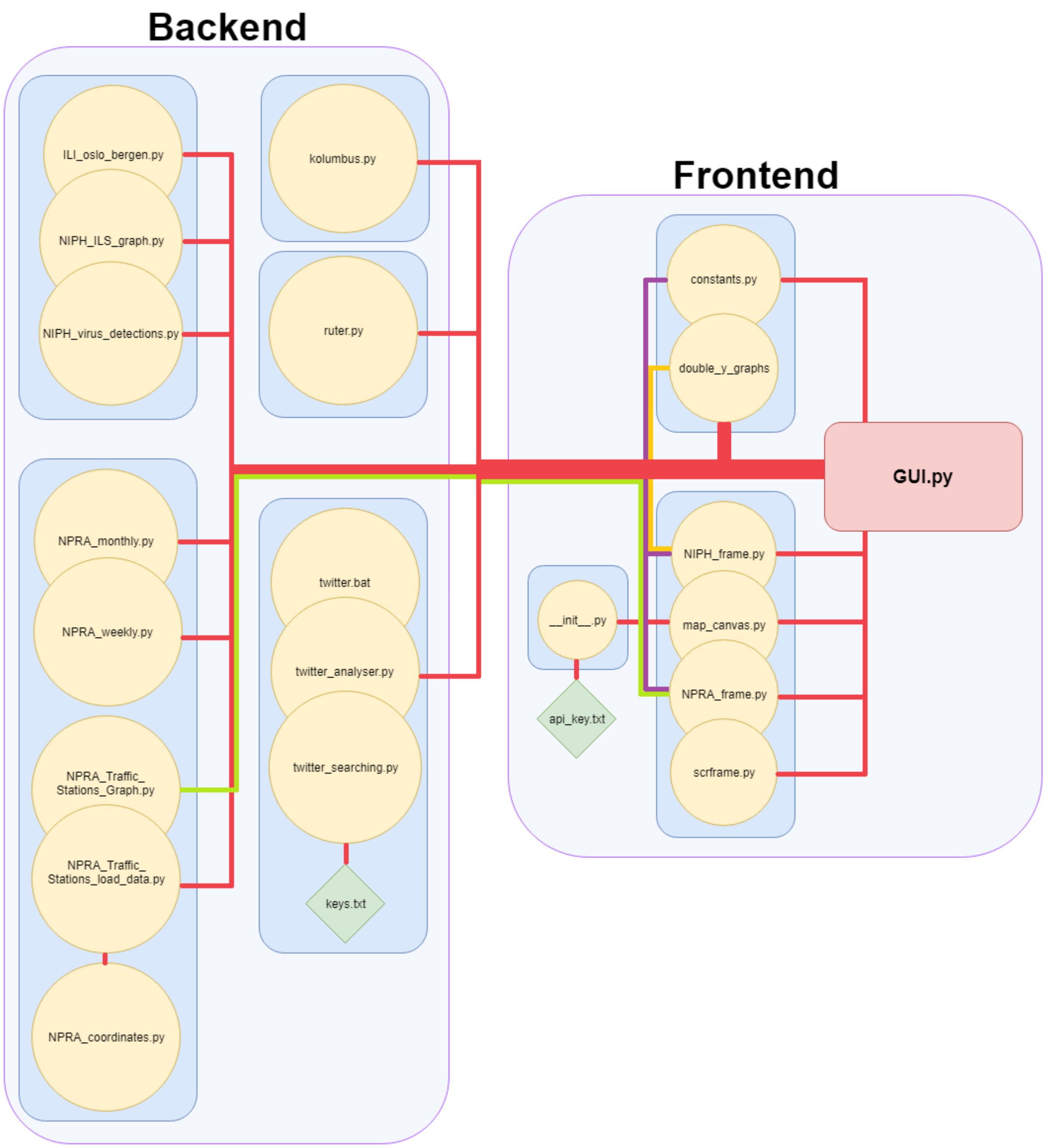


Figure 4.1: Simplification of the overall program structure and relation

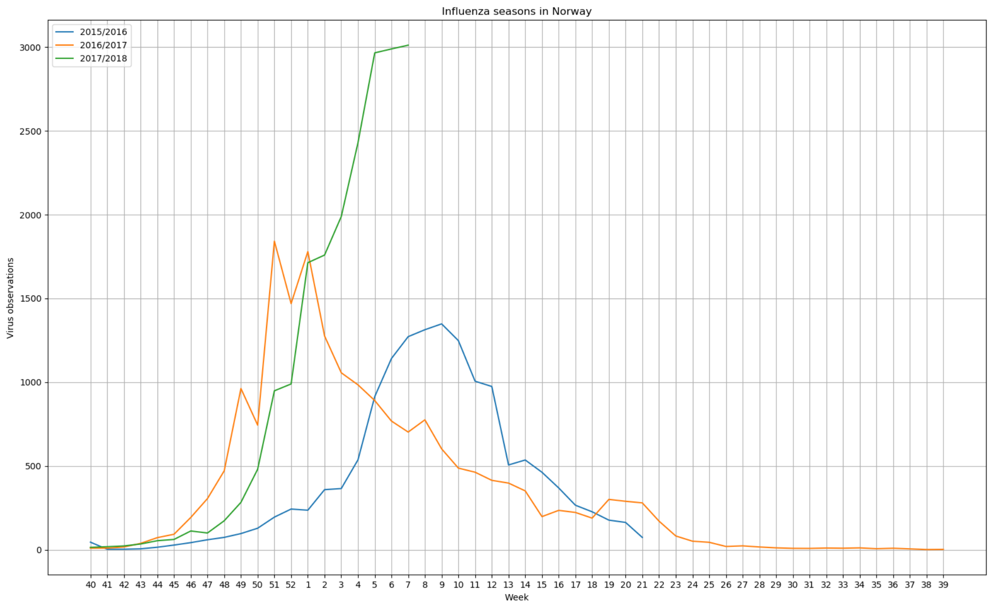


Figure 4.2: Influenza virus observation

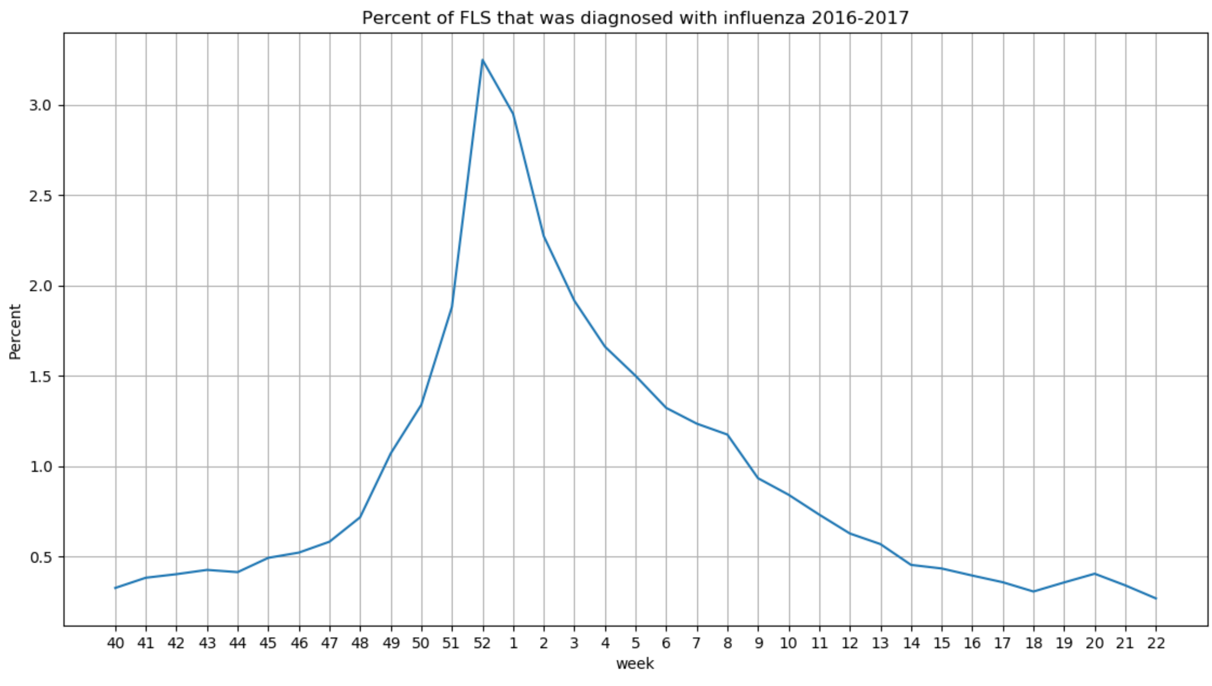


Figure 4.3: Influenza-like illnesses season 2016/2017

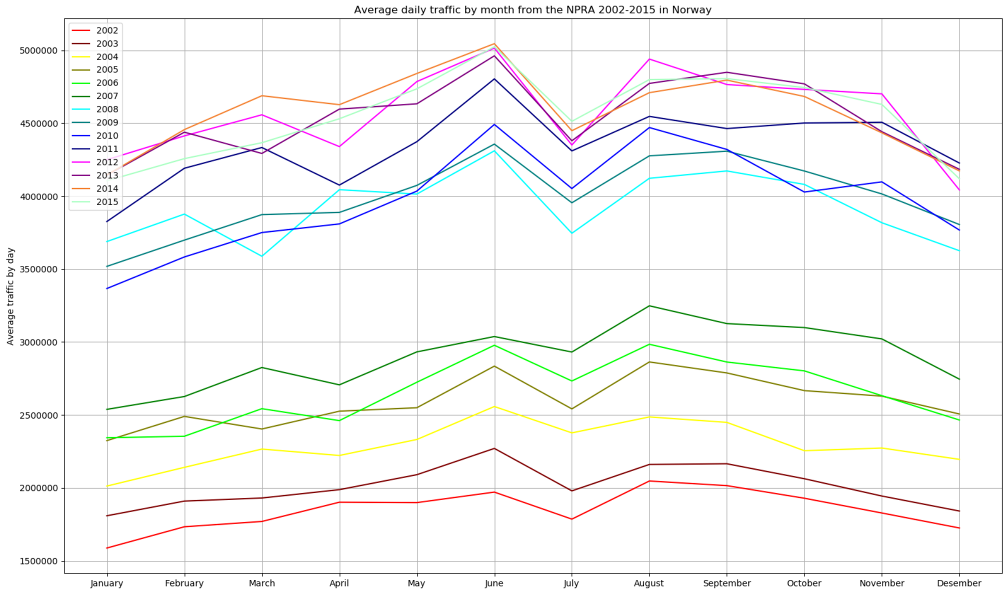


Figure 4.4: Annual traffic 2002-2015

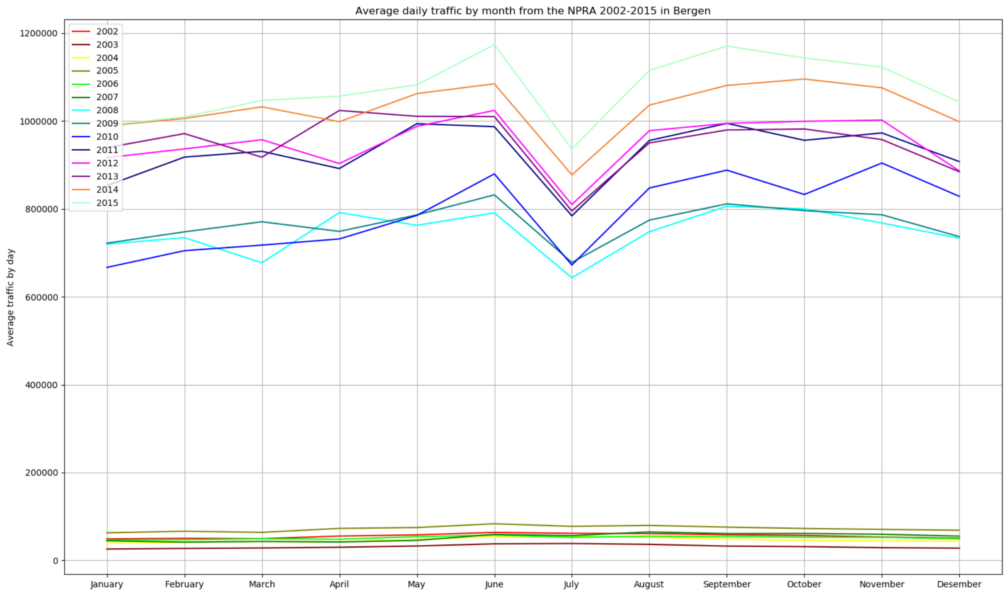


Figure 4.5: Bergen traffic 2002-2015

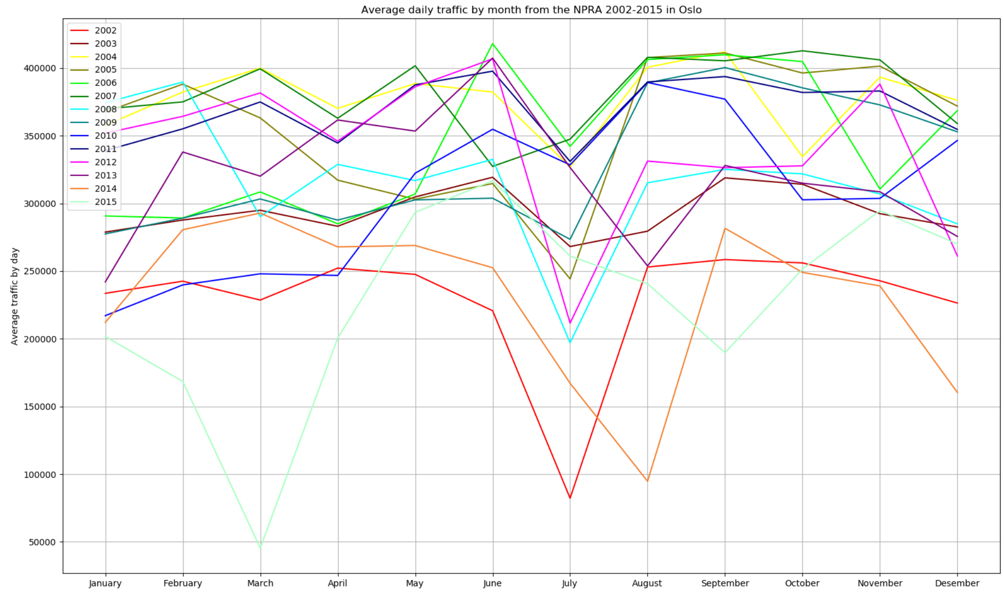


Figure 4.6: Oslo traffic 2002-2015

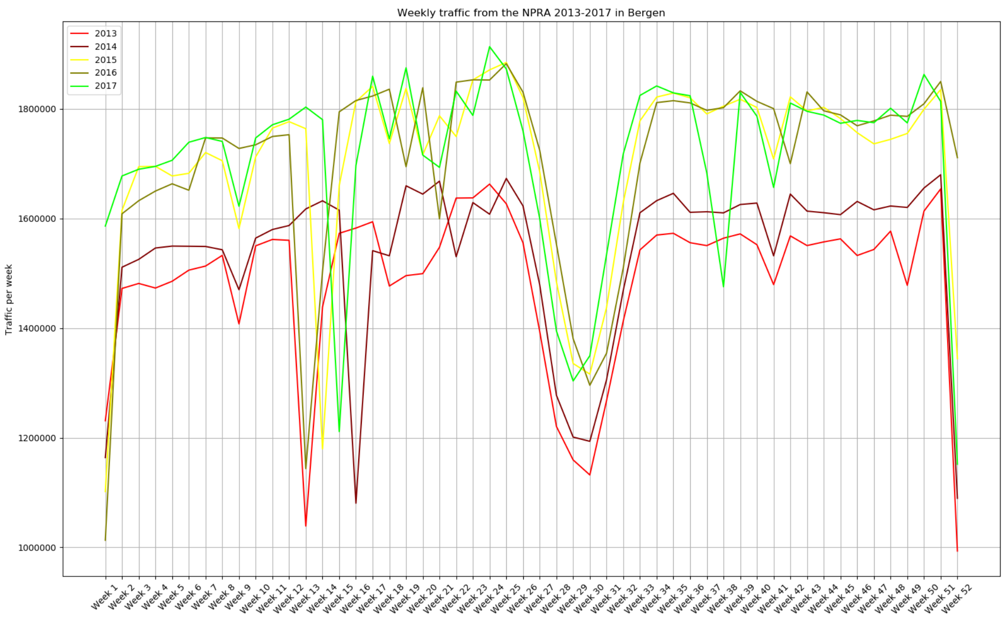


Figure 4.7: Weekly data of the city of Bergen

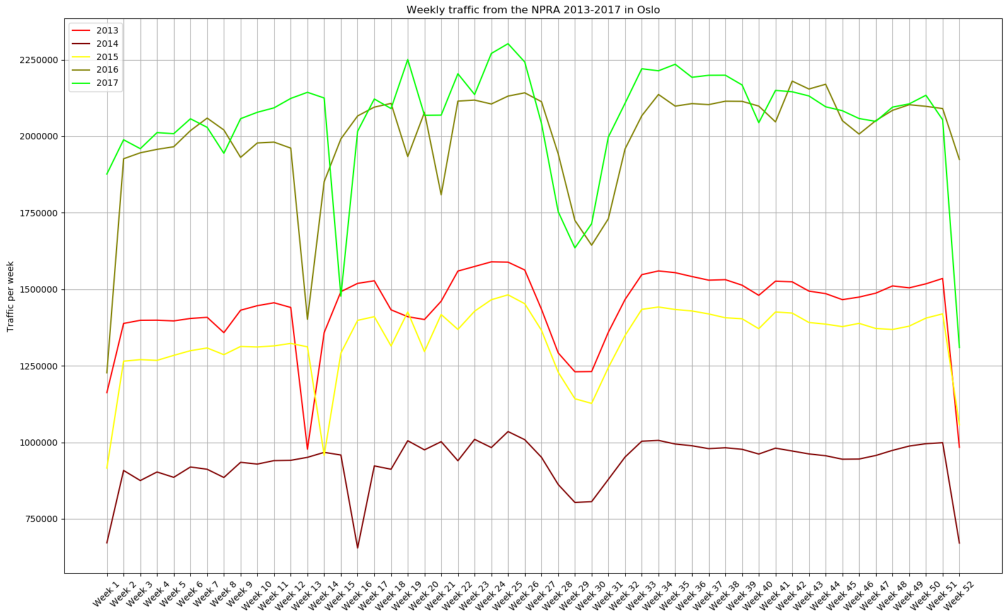


Figure 4.8: Weekly data of the city of Oslo

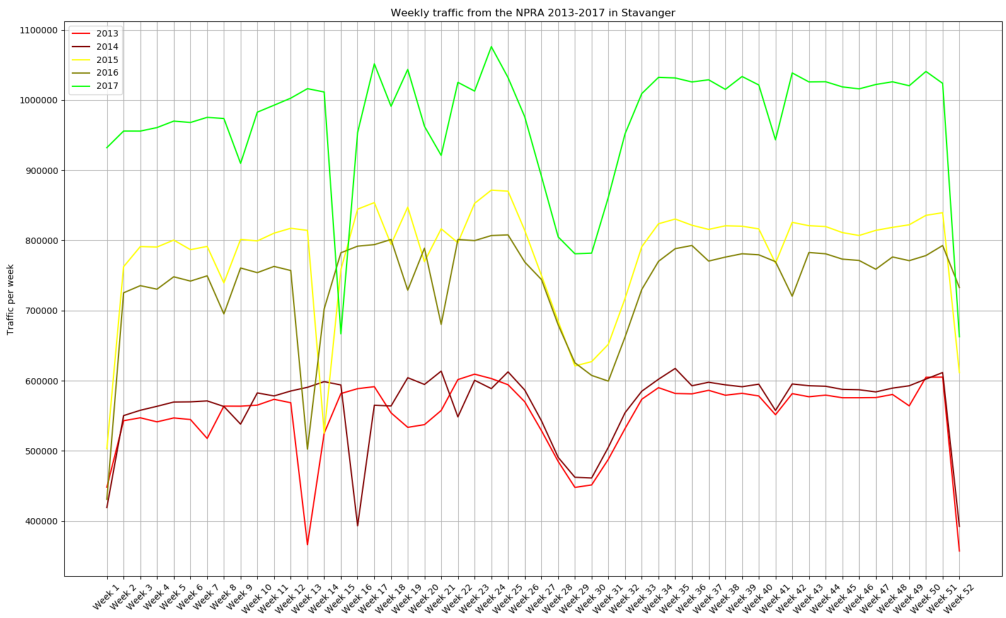


Figure 4.9: Weekly data of the city of Stavanger

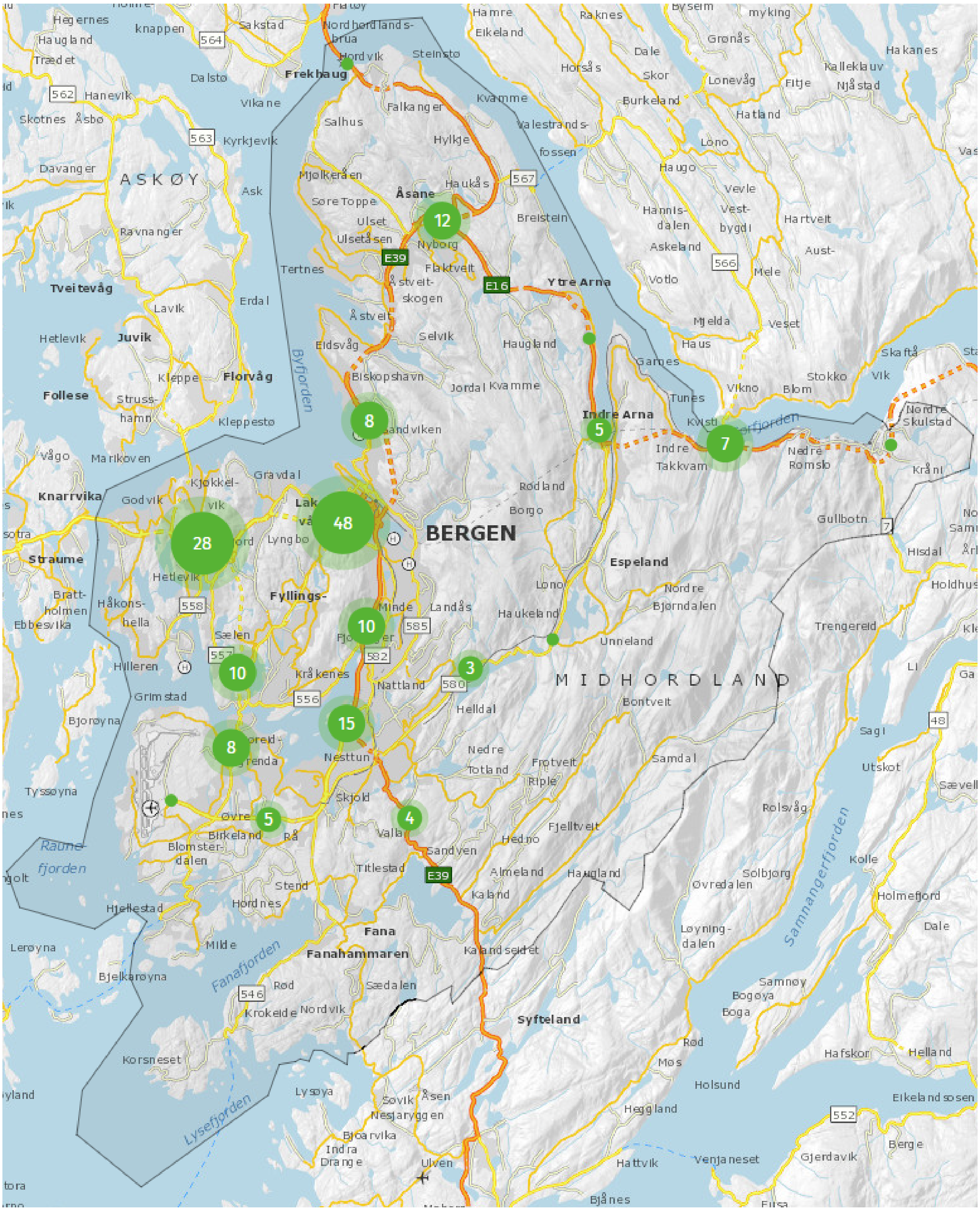


Figure 4.10: Geospatial bounds of Bergen. The green circles show where the traffic registration stations are, and the number reveals how many there are in that general area.

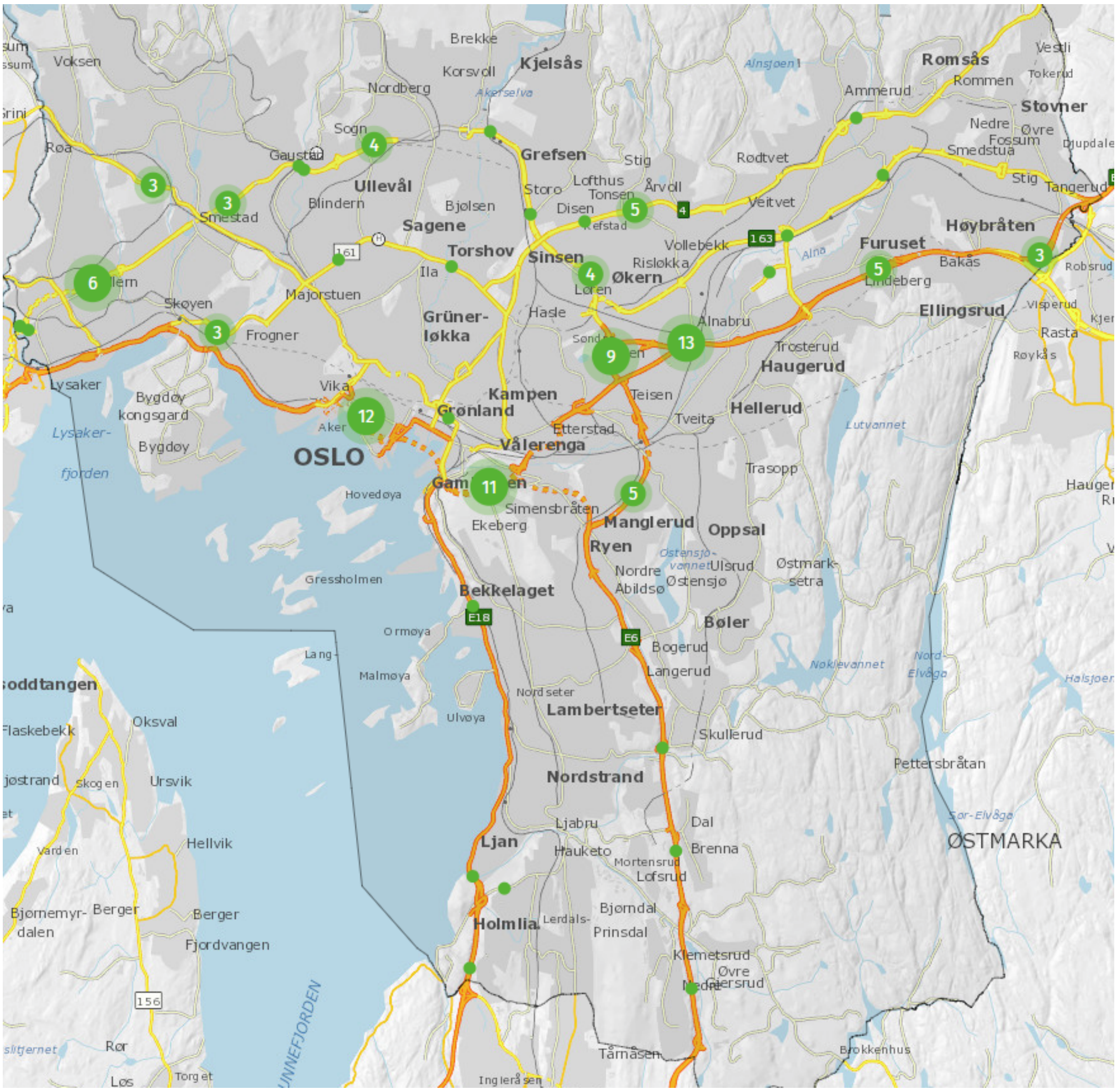


Figure 4.11: Geospatial bounds of Oslo. The green circles show where the traffic registration stations are, and the number reveals how many there are in that general area.

Figure 4.12: Geospatial bounds of Stavanger. The green circles show where the traffic registration stations are, and the number reveals how many there are in that general area.

Figure 4.13: Geospatial hourly bounds of Bergen

Figure 4.14: Geospatial hourly bounds of Oslo

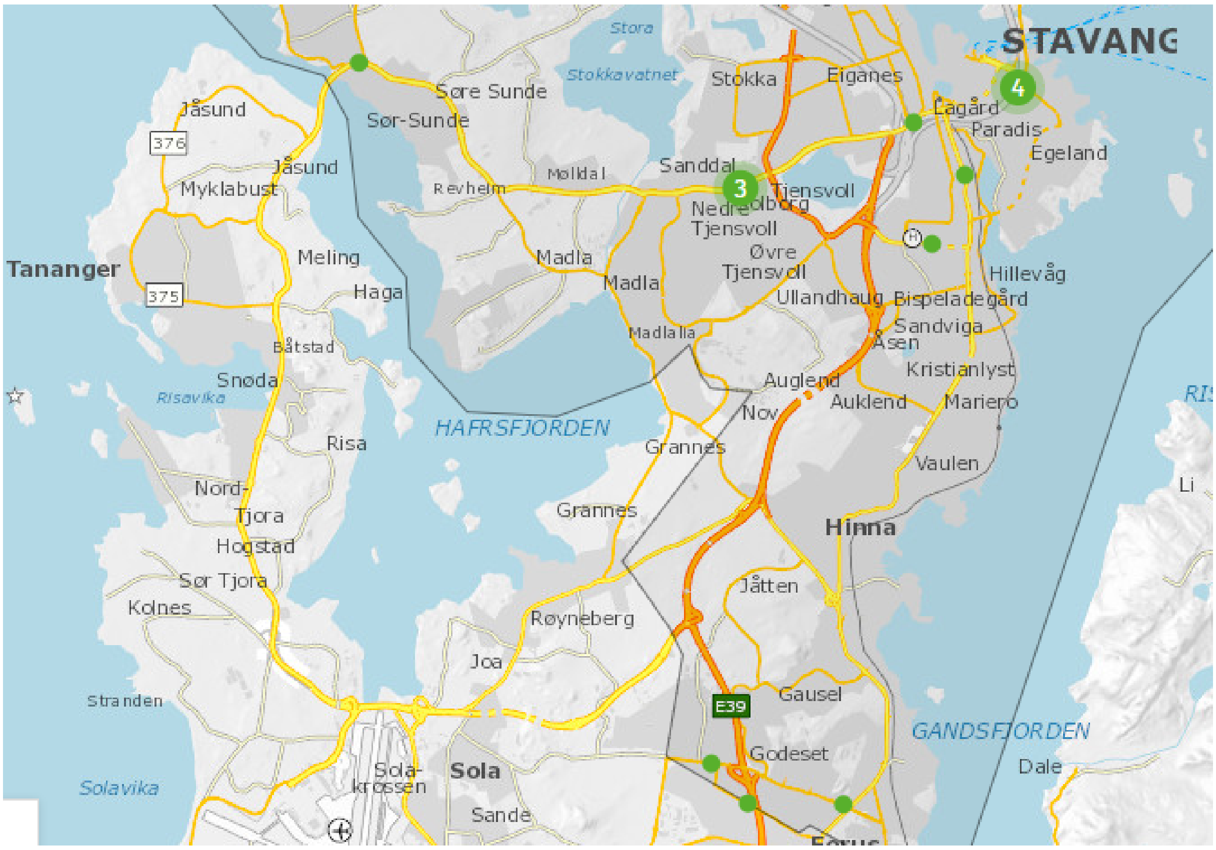


Figure 4.15: Geospatial hourly bounds of Stavanger

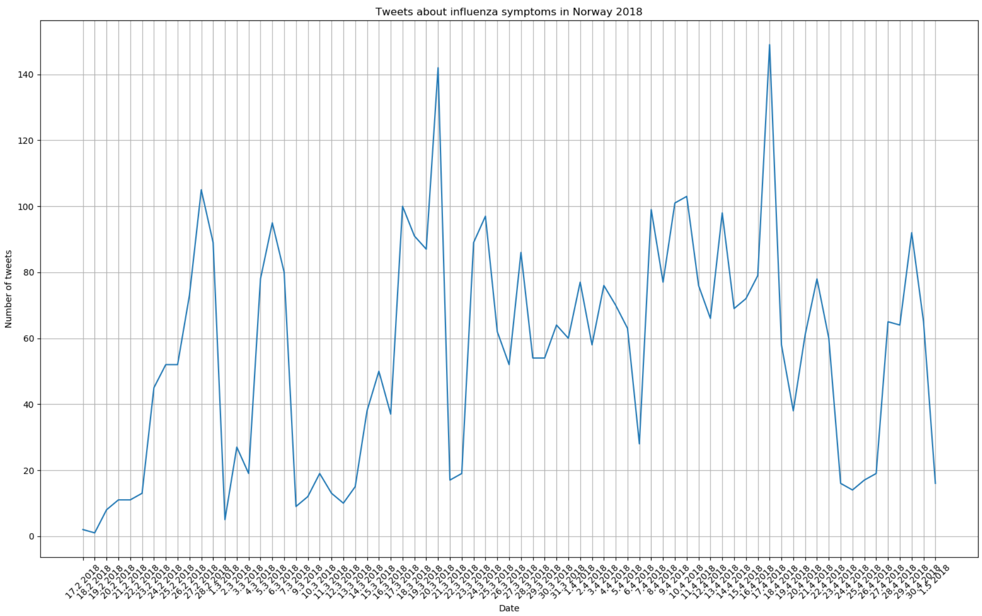


Figure 4.16: Tweets concerning ILS of 2018

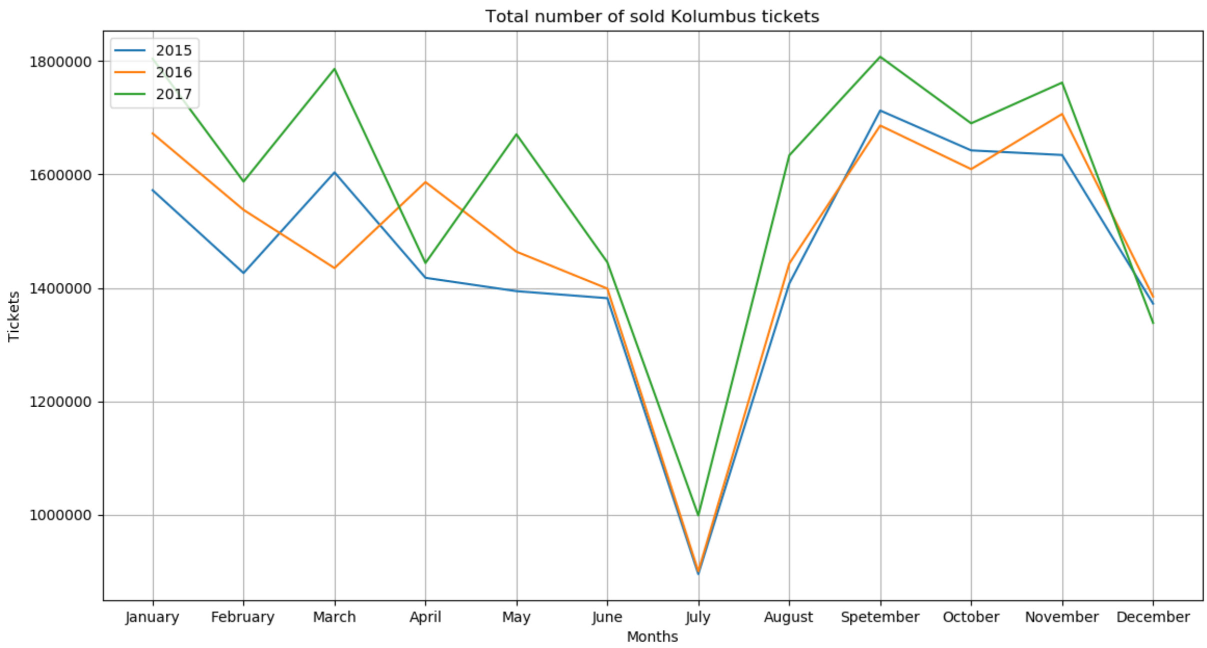


Figure 4.17: Monthly passenger travel with Kolumbus

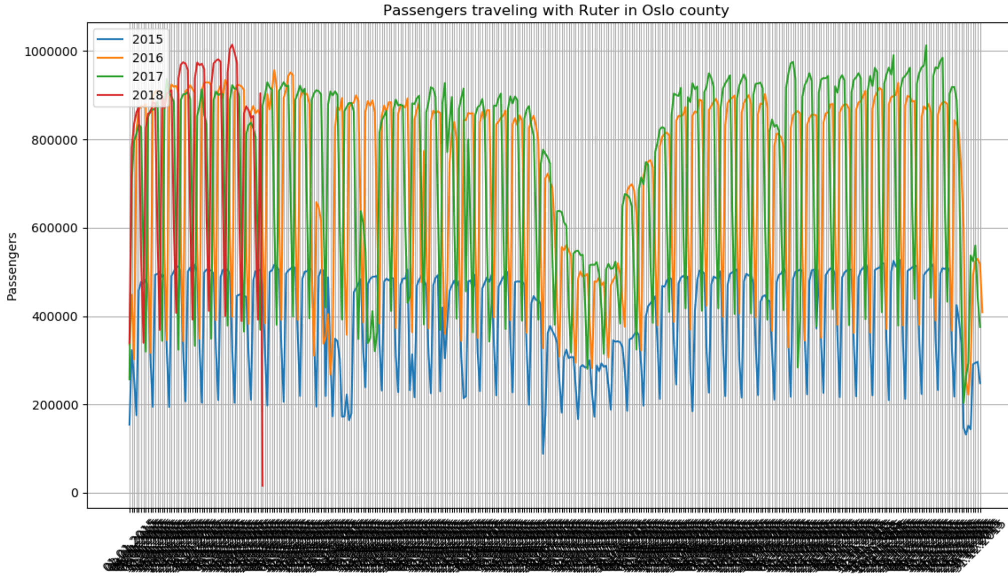


Figure 4.18: Daily tickets sold with Ruter, the year of 2015 does not contain Oslo's underground train service passenger data

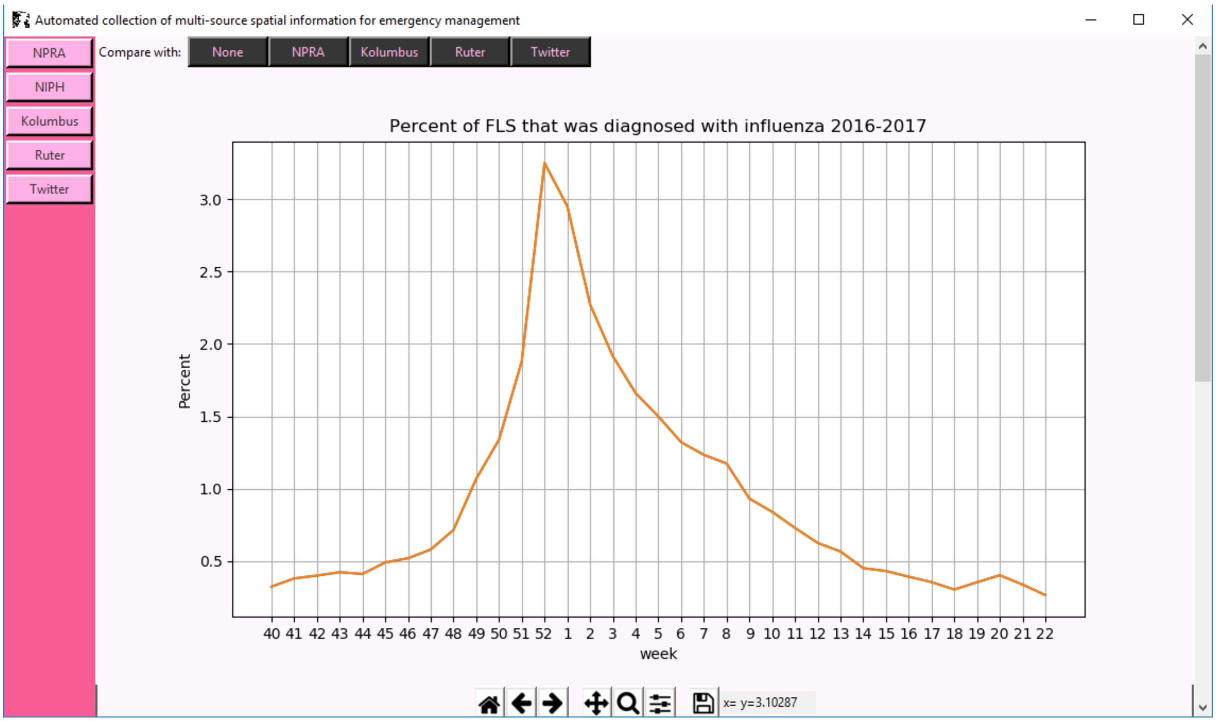


Figure 4.19: The GUI

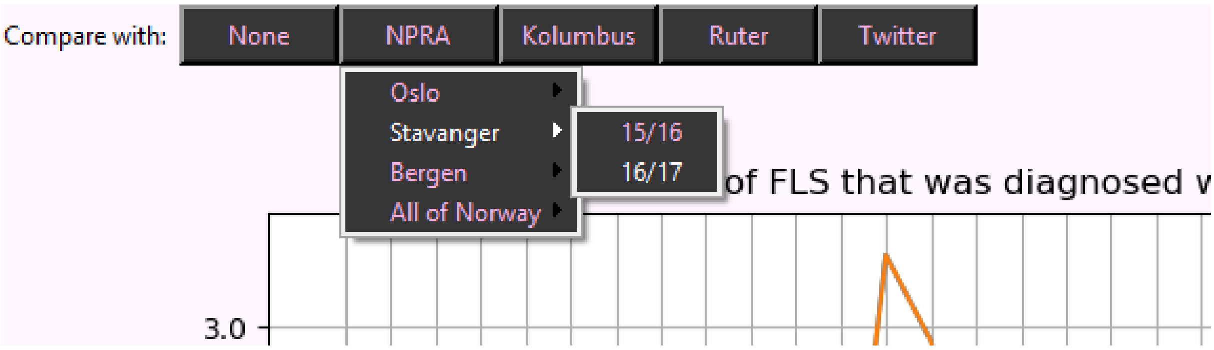


Figure 4.20: NIPH comparing buttons panel

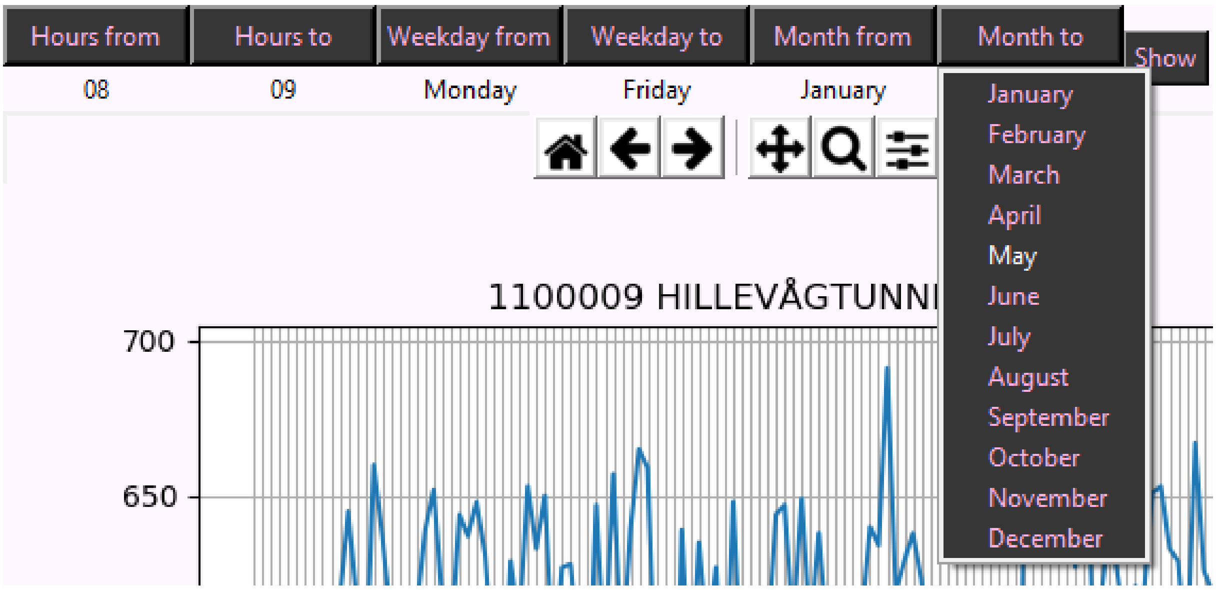


Figure 4.21: NPRA query buttons panel

Chapter 5

Results

This chapter describes the subjective view of the results derived from this thesis's program detailed in chapter three and four. Discussion about the results is elaborated upon in the following chapter.

5.1 NPRA

There are three levels of data available: monthly, weekly and hourly. For this reason, the monthly dataset will be disregarded as there are better data available. Weekly data distinctly show the Norwegian holidays described in table 5.1. It is important to take these days into account when deriving information from these data, as otherwise, it would be easy to conclude wrongly. The most dramatic drop is the summer vacation, which luckily is outside the influenza season anyway. Another challenge with holidays and vacations is that the start and duration change yearly, and because of the Gregorian calendar set dates shift one day up the next weekday for the next year. This needs to be taken into consideration, and possibly weeded out or glossed over in order to avoid misinterpretations.

Vacation/Holiday	**When**
Summer vacation	About nine weeks from the midth of June to the end of August
Autumn vacation	One week or a long weekend in September or October, usually in week 39, 40 or 41.
Christmas holiday	Usually two weeks from the end of December to the start of January
Winter vacation	usually in week 7, 8 or nine in February or March
Easter holiday	10-11 days at the end of March or beginning of April
Other and Christian holy days	Labour Day, Ascension Day, Constitution Day

Table 5.1: The Norwegian holidays and vacations

Further, the weekly graphs show a considerable increase in traffic each year, when asked about this the NPRA admitted to their action plan to increase the numbers of traffic registration stations yearly. This increase of infrastructure is transparent in the graphs shown as jumps in the amount of traffic with each new year. From the end of November to the beginning of December there is a slight drop in the amount of traffic without there being any vacations or holidays, this anomaly might be correlated with the influenza season as numbers of reported virus observations and ILI incidents seems to increase at the same time. After influenza, occurrences spike and begin to decline traffic slightly return to normal over the course of January to June. The weekly data is an aggregated set of many traffic registration stations based on the cities or the all of Norway, therefore roadworks, accidents or closed roads is not directly apparent. They are however visible, by assumption, on the hourly datasets as they only show one traffic registration station at a time. When in doubt of closed roads one could pick another traffic registration station nearby and see if it is also affected in the same manner. Another advantage with the hourly datasets is that there is not a dramatic yearly increase of traffic, which means more reliable data can be obtained, especially from the older traffic registration stations that have been operational for several years already. The map next to the hourly graph shows the available traffic registration stations to choose from.

5.2 Twitter

The way that twitter_analyser.py works are that it simply counts the number of occurrences of tweets and then draws a graph based on that count. The Twitter data collected in twitter_data.txt still contains duplicates although efforts were taken to prevent this. The duplicates may affect the graph drawn in batches as spikes where articles or hype are written about influenza or with other of the search terms. The Twitter data has a distinct pulse following the time when people post messages on social media the most by week[30], Mondays to Thursdays have a high yield of tweets, and then the weekends are calmer. This at least shows that the data collected is somewhat in accordance with other social media in other parts of the world. During the collection of tweets the event of the Norwegian Easter holiday occurred, from the graph shown the spikes even out and there is a more consistent flow of tweets throughout the holiday. When comparing the datasets of twitter and NIPH there is a clear similarity between them. The Twitter data seems to follow the trend downwards with the NIPH when the season is coming close to an end. However, the Twitter data seems to be lagging behind by 10 weeks, even less so with the ILI data from Bergen. This is in direct contradiction with both research referenced earlier in chapter two, and with this thesis's expectations.

5.3 Kolumbus

The Kolumbus data is the least interesting as it does not have spatial specific data and that the data resolution is too low on a monthly basis to see any anomalies. The longer Norwegian vacations and holidays are still somewhat visible though. This goes to show that sufficient temporal resolution is critical in order to derive any useful information from data in this thesis.

5.4 Ruter

Comparing the Ruter data with the ILI data of Oslo is especially interesting because Ruter is the public transportation administrator in that city. As with the NPRA data the Norwegian holidays and vacations are apparent as described in section 5.1. The weeks of 47, 48 and 49 show a slight decrease of passenger travel without overlapping any vacations and holidays, there is also a slight increase of reported ILI every influenza season in those weeks. This correlation might be relevant and should be investigated further. After the Christmas holiday passenger travel struggle for a few weeks to 'catch up' to a more stable level, interestingly enough the influenza seasons usually are on its peaks at that very time. The amount of passenger travel also seems to be slightly increasing as the influenza season declines.

Chapter 6

Discussion

In this chapter, the results and other constraints encountered will be discussed.

6.1 Project Management

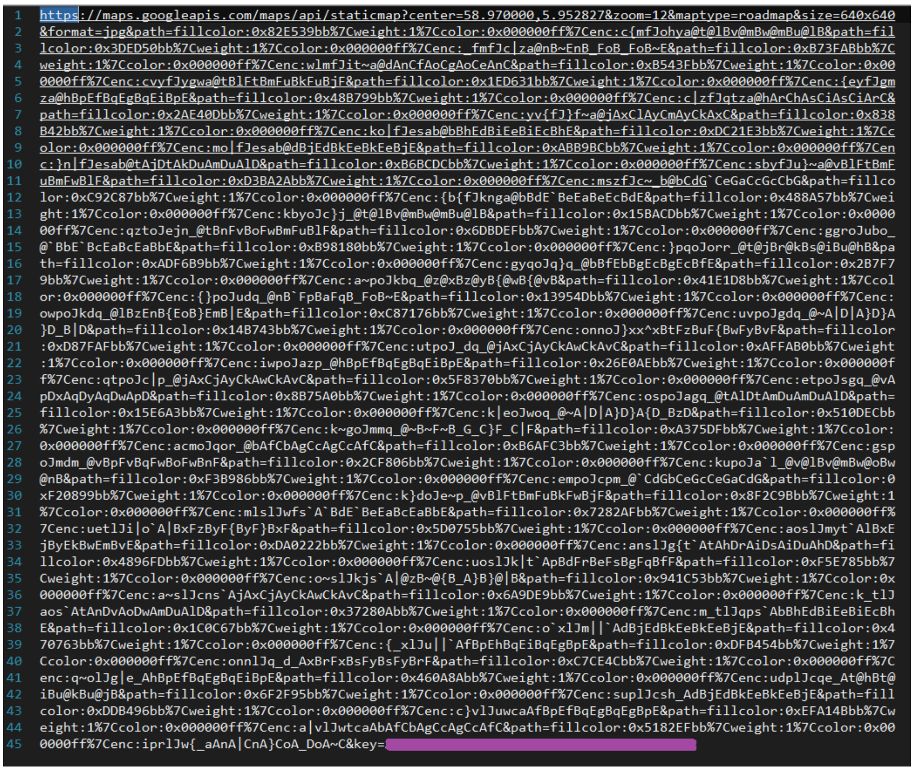
Early in the planning and management phase of this thesis, it became evident that the Norwegian infrastructure for retrieving data from various public sources by API was not sufficient for the needs of this thesis. Therefore the initial plan to automate the collection of data needed was adapted to the means of acquiring the data by manually asking the various agencies and implementing their data hard-coded. This makes the program much less scalable and flexible than hoped for, and severely inhibits future contributions as it may be difficult to couple new data with the inputs of the backend's data structure. The missing automation part will probably hinder future use of this program. The only two APIs used are of American origin, namely Google static map and Twitter. In these regards the automation element that this thesis anticipated failed, however not by a critical means as manual retrieval of data was still possible.

6.2 Project resolutions

In the middle of the time scope for this thesis, a frontend to the backend was desired and thus planning to construct this began. There were several options for choosing not only from the languages the GUI would be based upon but consideration of how a map would be projected as well. These were the main concerns and had to be compatible with each other. The first choice was between Python's Tkinter GUI module and Node Javascript GUI. The main reason Python was chosen was that it offered the easiest coupling with the backend. Javascript prohibits direct reading from local files, and thus the backend would have to be mounted on a server in order to provide its functions to a frontend. The author of this thesis had little experience with this, and learning a whole new trade was daunting and seemed insurmountable within the time scope of this thesis, therefore the enticing of the familiarity of Python triumphed. In hindsight, it would probably be better to undertake a Node/Javascript approach because some sort of database to store the backend's data is needed anyway and is probably a more feasible solution, more on this in the following chapter.

Choosing a map implementation was difficult, Python has several options like GeoPandas, ipyleaflet, Google static map, cartopy, OpenStreetMap, and basemap. All of the mentioned was hard to install and was sorely limited in function and potential, except OpenStreetMap and Google static map. Upon further investigation Goompy, as described in chapter 2, was discovered and offered a nearly effortless implementation of the map in the already applied design of the frontend.

The advantage the Python solution has is that it requires few installations of external modules and is easily downloaded and mountable on many platforms. The disadvantage with Python's Matplotlib is that drawing many graphs requires a lot of memory and processor resources, therefore it is important to manage the graphs drawn, and only load those that need be loaded at a time, flushing those that are no longer in use. The advantage of Google static map is that it is a well implemented and established service with consistent qualitative measures. Google offers fewer road details than OpenStreetMap, and that serves this thesis perfectly as the visualization needed was simply showing locations of traffic registration stations and not necessarily other roads. The disadvantages with Google static map is that there are standard usage limits (which can simply be overcome with paying for more). Pixel resolution is set to a maximum of 640x640, and the free usage is limited to 25.000 map loads per 24 hours. These two limits are not really a problem: The pixel limit is overcome by simply requesting more map loads, and the map loads limit is very high. On average Goompy does 4 map loads per zoom and 25.000 / 4 = 6250 zooms per 24 hours, average Norwegian working hours per day is 7.5 hours, this means that one would reach the limit if there are 6.250 / 7.5 / 60 = 13.9 zooms per second. This limit was never reached in testing and although it is a high limit if reached the map simply stops working for the remainder of the time to the next 24 hours. Perhaps the most severe limit Google static map have for the scope of this thesis is its maximum URL size of 8192 characters. Figure 6.1 show the programs URL that it sends to the Google map servers, containing a standard map and fifty-three traffic registration stations each with their individually different sizes and colors this surmounts to a total of 4.975 characters already, which is 60.7% of the total allowed. Although the url have encoded polylines it is already quite long, Loading all of Norway's current 10.066 traffic registration stations using Google static map with this thesis's current algorithms is not feasible, although this could be solved by clustering the traffic registration stations together, and only loading what you actually can see on the map. This would require more Goompy modifications by somehow fetching only those traffic registration stations that are actually currently visible on the map.




```
https://maps.googleapis.com/maps/api/staticmap?center=58.970000,5.952827&zoom=12&maptype=roadmap&size=640x640
&format=jpg&path=fillcolor:0x82E539bb%7Cweight:1%7Ccolor:0x000000ff%7Cenc:c{mfJohya@t@lBv@mBw@mBu@lB&path=fil
lcolor:0x3DED50bb%7Cweight:1%7Ccolor:0x000000ff%7Cenc:_fmfJc|za@nB~EnB_FoB_FoB~E&path=fillcolor:0xB73FABbb%7C
weight:1%7Ccolor:0x000000ff%7Cenc:wlmfJit~a@dAnCfAoCgAoCeAnC&path=fillcolor:0xB543Fbb%7Cweight:1%7Ccolor:0x00
0000ff%7Cenc:cvyfJygwa@tBlFtBmFuBkFuBjF&path=fillcolor:0x1ED631bb%7Cweight:1%7Ccolor:0x000000ff%7Cenc:{eyfJgm
za@hBpEfBqEgBqEiBpE&path=fillcolor:0x48B799bb%7Cweight:1%7Ccolor:0x000000ff%7Cenc:c|zfJqtza@hArChAsCiAsCiArC&
path=fillcolor:0x2AE40Dbb%7Cweight:1%7Ccolor:0x000000ff%7Cenc:yv{fJ}f~a@jAxClAyCmAyCkAxC&path=fillcolor:0x838
B42bb%7Cweight:1%7Ccolor:0x000000ff%7Cenc:ko|fJesab@bBhEdBiEeBiEcBhE&path=fillcolor:0xDC21E3bb%7Cweight:1%7Cc
olor:0x000000ff%7Cenc:mo|fJesab@dBjEdBkEeBkEeBjE&path=fillcolor:0xABB9BCbb%7Cweight:1%7Ccolor:0x000000ff%7Cen
c:}n|fJesab@tAjDtAkDuAmDuAlD&path=fillcolor:0xB6BCDCbb%7Cweight:1%7Ccolor:0x000000ff%7Cenc:sbyfJu}~a@vBlFtBmF
uBmFwBlF&path=fillcolor:0xD3BA2Abb%7Cweight:1%7Ccolor:0x000000ff%7Cenc:mszfJc~_b@bCdG`CeGaCcGcCbG&path=fillco
lor:0xC92C87bb%7Cweight:1%7Ccolor:0x000000ff%7Cenc:{b{fJknga@bBdE`BeEaBeEcBdE&path=fillcolor:0x488A57bb%7Cwei
ght:1%7Ccolor:0x000000ff%7Cenc:kbyoJc}j_@t@lBv@mBw@mBu@lB&path=fillcolor:0x15BACDbb%7Cweight:1%7Ccolor:0x0000
00ff%7Cenc:qztoJejn_@tBnFvBoFwBmFuBlF&path=fillcolor:0x6DBDEFbb%7Cweight:1%7Ccolor:0x000000ff%7Cenc:ggroJubo_
@`BbE`BcEaBcEaBbE&path=fillcolor:0xB98180bb%7Cweight:1%7Ccolor:0x000000ff%7Cenc:}pqoJorr_@t@jBr@kBs@iBu@hB&pa
th=fillcolor:0xADF6B9bb%7Cweight:1%7Ccolor:0x000000ff%7Cenc:gyqoJq}q_@bBfEbBgEcBgEcBfE&path=fillcolor:0x2B7F7
9bb%7Cweight:1%7Ccolor:0x000000ff%7Cenc:a~poJkbq_@z@xBz@yB{@wB{@vB&path=fillcolor:0x41E1D8bb%7Cweight:1%7Ccol
or:0x000000ff%7Cenc:{}poJudq_@nB`FpBaFqB_FoB~E&path=fillcolor:0x13954Dbb%7Cweight:1%7Ccolor:0x000000ff%7Cenc:
owpoJkdq_@lBzEnB{EoB}EmB|E&path=fillcolor:0xC87176bb%7Cweight:1%7Ccolor:0x000000ff%7Cenc:uvpoJgdq_@~A|D|A}D}A
}D_B|D&path=fillcolor:0x14B743bb%7Cweight:1%7Ccolor:0x000000ff%7Cenc:onnoJ}xx^xBtFzBuF{BwFyBvF&path=fillcolor
:0xD87FAFbb%7Cweight:1%7Ccolor:0x000000ff%7Cenc:utpoJ_dq_@jAxCjAyCkAwCkAvC&path=fillcolor:0xAFFAB0bb%7Cweight
:1%7Ccolor:0x000000ff%7Cenc:iwpoJazp_@hBpEfBqEgBqEiBpE&path=fillcolor:0x26E0AEbb%7Cweight:1%7Ccolor:0x000000f
f%7Cenc:qtpoJc|p_@jAxCjAyCkAwCkAvC&path=fillcolor:0x5F8370bb%7Cweight:1%7Ccolor:0x000000ff%7Cenc:etpoJsgq_@vA
pDxAqDyAqDwApD&path=fillcolor:0x8B75A0bb%7Cweight:1%7Ccolor:0x000000ff%7Cenc:ospoJagq_@tAlDtAmDuAmDuAlD&path=
fillcolor:0x15E6A3bb%7Cweight:1%7Ccolor:0x000000ff%7Cenc:k|eoJwoq_@~A|D|A}D}A{D_BzD&path=fillcolor:0x510DECbb
%7Cweight:1%7Ccolor:0x000000ff%7Cenc:k~goJmmq_@~B~F~B_G_C}F_C|F&path=fillcolor:0xA375DFbb%7Cweight:1%7Ccolor:
0x000000ff%7Cenc:acmoJqor_@bAfCbAgCcAgCcAfC&path=fillcolor:0xB6AFC3bb%7Cweight:1%7Ccolor:0x000000ff%7Cenc:gsp
oJmdm_@vBpFvBqFwBoFwBnF&path=fillcolor:0x2CF806bb%7Cweight:1%7Ccolor:0x000000ff%7Cenc:kupoJa`l_@v@lBv@mBw@oBw
@nB&path=fillcolor:0xF3B986bb%7Cweight:1%7Ccolor:0x000000ff%7Cenc:empoJcpm_@`CdGbCeGcCeGaCdG&path=fillcolor:0
xF20899bb%7Cweight:1%7Ccolor:0x000000ff%7Cenc:k}doJe~p_@vBlFtBmFuBkFwBjF&path=fillcolor:0x8F2C9Bbb%7Cweight:1
%7Ccolor:0x000000ff%7Cenc:mlslJwfs`A`BdE`BeEaBcEaBbE&path=fillcolor:0x7282AFbb%7Cweight:1%7Ccolor:0x000000ff%
7Cenc:uetlJi|o`A|BxFzByF{ByF}BxF&path=fillcolor:0x5D0755bb%7Cweight:1%7Ccolor:0x000000ff%7Cenc:aoslJmyt`AlBxE
jByEkBwEmBvE&path=fillcolor:0xDA0222bb%7Cweight:1%7Ccolor:0x000000ff%7Cenc:anslJg{t`AtAhDrAiDsAiDuAhD&path=fi
llcolor:0x4896FDbb%7Cweight:1%7Ccolor:0x000000ff%7Cenc:uoslJk|t`ApBdFrBeFsBgFqBfF&path=fillcolor:0xF5E785bb%7
Cweight:1%7Ccolor:0x000000ff%7Cenc:o~slJkjs`A|@zB~@{B_A}B}@|B&path=fillcolor:0x941C53bb%7Cweight:1%7Ccolor:0x
000000ff%7Cenc:a~slJcns`AjAxCjAyCkAwCkAvC&path=fillcolor:0x6A9DE9bb%7Cweight:1%7Ccolor:0x000000ff%7Cenc:k_tlJ
aos`AtAnDvAoDwAmDuAlD&path=fillcolor:0x37280Abb%7Cweight:1%7Ccolor:0x000000ff%7Cenc:m_tlJqps`AbBhEdBiEeBiEcBh
E&path=fillcolor:0x1C0C67bb%7Cweight:1%7Ccolor:0x000000ff%7Cenc:o`xlJm||`AdBjEdBkEeBkEeBjE&path=fillcolor:0x4
70763bb%7Cweight:1%7Ccolor:0x000000ff%7Cenc:{_xlJu||`AfBpEhBqEiBqEgBpE&path=fillcolor:0xDFB454bb%7Cweight:1%7
Ccolor:0x000000ff%7Cenc:onnlJq_d_AxBrFxBsFyBsFyBrF&path=fillcolor:0xC7CE4Cbb%7Cweight:1%7Ccolor:0x000000ff%7C
enc:q~olJg|e_AhBpEfBqEgBqEiBpE&path=fillcolor:0x460A8Abb%7Cweight:1%7Ccolor:0x000000ff%7Cenc:udplJcqe_At@hBt@
iBu@kBu@jB&path=fillcolor:0x6F2F95bb%7Cweight:1%7Ccolor:0x000000ff%7Cenc:suplJcsh_AdBjEdBkEeBkEeBjE&path=fill
color:0xDDB496bb%7Cweight:1%7Ccolor:0x000000ff%7Cenc:c}vlJuwcaAfBpEfBqEgBqEgBpE&path=fillcolor:0xEFA14Bbb%7Cw
eight:1%7Ccolor:0x000000ff%7Cenc:a|vlJwtcaAbAfCbAgCcAgCcAfC&path=fillcolor:0x5182EFbb%7Cweight:1%7Ccolor:0x00
0000ff%7Cenc:iprlJw{_aAnA|CnA}CoA_DoA~C&key=[redacted]
```


Figure 6.1: Size of the programs Google static map URLs

Chapter 7

Conclusion

This chapter presents possible future works and concludes this thesis.

7.1 Future works

TODO: svakheter, hvordan gjøre bedre? hva er mitt bidrag?

The program to this thesis was created by one person alone without code supervision, therefore it contains some bugs and inefficient solutions. Ameliorating these algorithm requires more work than what is this thesis's time scope. One known negligent solution is the algorithms in the program of double_y_graphs.py where redundant calculations take place. Fixing this would make the program gui.py slightly faster and be more structural preferable. Another known bug is that the buttons panel disappear when the window size is not big enough.

7.1.1 Google static map

As discussed in the previous chapter clustering traffic registration stations would solve the maximum URL problem. When presented with a map that shows all of Norway instead of showing each traffic registration stations one could cluster them together by proximity, and when zooming in present an even more fine tuned clustering until the zoom level is sufficient enough to show all of the traffic registration stations. This is already a standard way of presenting spatial data as seen in the NPRA's 'vegkart'[31] when selecting multiple elements. Further standardizing colors and sizes would significantly save url length, the thought behind different sizes and colors was only intended on a very zoomed in level and is not necessarily needed when showing cluster. The url size problem would also completely vanish if taken a Node/Javascript approach instead.

7.1.2 Database

At the very end of the time scope of this thesis it became obvious that the backend's data should have been implemented in some sort of a database in order to speed up the process of reading and extracting exact information. Data filtering is the process of refining data sets for relevant user information, different filters can be tailored for different needs. Filtering becomes particularly useful in the NPRA's hourly dataset, an example would be to filter out the different lanes available this would on average make the algorithm two times faster. In order to take advantage of data filtering and indexing the data would have to be implemented in a database. A possible more optimal solution would be to rewrite the entire project in Node/Javascript and mount the backend's data in a server.

7.2 Conclusion

much research, such wow ...

Appendix A

Appendix Title

Bibliography

[1] L. O. Grottenberg, O. Njå, E. Tøssebro, G. Braut, R. Tønnessen, and G. M. Grøneng, "Detecting flu outbreaks based on spatiotemporal information from urban systems – designing a novel study," *Icwsm*, vol. 20, pp. 1–7, 2017.

[2] "Pandemic influenza risk management: A who guide to inform and harmonize national and international pandemic preparedness and response." http://www.who.int/influenza/preparedness/pandemic/influenza_risk_management_update201 Accessed: 2018-05-23.

[3] A. D. Iuliano, K. M. Roguski, H. H. Chang, D. J. Muscatello, R. Palekar, S. Tempia, C. Cohen, J. M. Gran, D. Schanzer, B. J. Cowling, *et al.*, "Estimates of global seasonal influenza-associated respiratory mortality: a modelling study," *The Lancet*, 2017.

[4] "The norwegian institute of public health." https://www.fhi.no/en/. Accessed: 2018-06-11.

[5] C. Poletto, M. Tizzoni, and V. Colizza, "Human mobility and time spent at destination: impact on spatial epidemic spreading," *Journal of theoretical biology*, vol. 338, pp. 41–58, 2013.

[6] T. Wibisono, D. M. Aleman, and B. Schwartz, "A non-homogeneous approach to simulating the spread of disease in a pandemic outbreak," in *Simulation Conference, 2008. WSC 2008. Winter*, pp. 2941–2941, IEEE, 2008.

[7] B. Yang, H. Pei, H. Chen, J. Liu, and S. Xia, "Characterizing and discovering spatiotemporal social contact patterns for healthcare," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 8, pp. 1532–1546, 2017.

[8] C. Robertson, "Towards a geocomputational landscape epidemiology: surveillance, modelling, and interventions," *GeoJournal*, vol. 82, no. 2, pp. 397–414, 2017.

[9] M. K. Enduri and S. Jolad, "Dynamics of dengue disease with human and vector mobility," *Spatial and spatio-temporal epidemiology*, vol. 25, pp. 57–66, 2018.

[10] "The norwegian institute of public health: About the norwegian syndromic surveillance system." https://www.fhi.no/en/hn/statistics/NorSySS/about-the-norwegian-syndromic-surveillance-system/. Accessed: 2018-06-11.

[11] L. Palen, R. Soden, T. J. Anderson, and M. Barrenechea, "Success & scale in a data-producing organization: The socio-technical evolution of openstreetmap in response to humanitarian events," in *Proceedings of the 33rd annual ACM conference on human factors in computing systems*, pp. 4113–4122, ACM, 2015.

[12] OpenStreetMap, "Openstreetmap is a map of the world, created by people like you and free to use under an open licence.." https://www.openstreetmap.org, 2018.

[13] Y. Hu, K. Janowicz, and Y. Chen, "Task-oriented information value measurement based on space-time prisms," *International Journal of Geographical Information Science*, vol. 30, no. 6, pp. 1228–1249, 2016.

[14] J. Anderson, R. Soden, B. Keegan, L. Palen, and K. M. Anderson, "The crowd is the territory: Assessing quality in peer-produced spatial data during disasters," *International Journal of Human–Computer Interaction*, vol. 34, no. 4, pp. 295–310, 2018.

[15] S. B. Elson, D. Yeung, P. Roshan, S. R. Bohandy, and A. Nader, *Using social media to gauge Iranian public opinion and mood after the 2009 election.* Rand Corporation, 2012.

[16] H. Achrekar, A. Gandhe, R. Lazarus, S.-H. Yu, and B. Liu, "Predicting flu trends using twitter data," in *Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference on*, pp. 702–707, IEEE, 2011.

[17] J. Eisenstein, B. O'Connor, N. A. Smith, and E. P. Xing, "A latent variable model for geographic lexical variation," in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pp. 1277–1287, Association for Computational Linguistics, 2010.

[18] T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake shakes twitter users: real-time event detection by social sensors," in *Proceedings of the 19th international conference on World wide web*, pp. 851–860, ACM, 2010.

[19] M. J. Paul and M. Dredze, "You are what you tweet: Analyzing twitter for public health.," *Icwsm*, vol. 20, pp. 265–272, 2011.

[20] J. P. De Albuquerque, B. Herfort, A. Brenning, and A. Zipf, "A geographic approach for combining social media and authoritative data towards identifying useful information for disaster management," *International Journal of Geographical Information Science*, vol. 29, no. 4, pp. 667–689, 2015.

[21] R. A. Gonzalez and N. Bharosa, "A framework linking information quality dimensions and coordination challenges during interagency crisis response," in *System Sciences, 2009. HICSS'09. 42nd Hawaii International Conference on*, pp. 1–10, IEEE, 2009.

[22] B. Resch, F. Usländer, and C. Havas, "Combining machine-learning topic models and spatiotemporal analysis of social media data for disaster footprint and damage assessment," *Cartography and Geographic Information Science*, vol. 45, no. 4, pp. 362–376, 2018.

[23] H. Shao, K. Hossain, H. Wu, M. Khan, A. Vullikanti, B. A. Prakash, M. Marathe, and N. Ramakrishnan, "Forecasting the flu: designing social network sensors for epidemics," *arXiv preprint arXiv:1602.06866*, 2016.

[24] M. L. Stein, J. W. Rudge, R. Coker, C. van der Weijden, R. Krumkamp, P. Hanvoravongchai, I. Chavez, W. Putthasri, B. Phommasack, W. Adisasmito, *et al.*, "Development of a resource modelling tool to support decision makers in pandemic influenza preparedness: The asiaflucap simulator," *BMC public health*, vol. 12, no. 1, p. 870, 2012.

[25] "Interactive google maps for python, created by simon d. levy." https://github.com/simondlevy/GooMPy. Accessed: 2018-06-11.

[26] "Google static maps api." https://developers.google.com/maps/documentation/maps-static/intro. Accessed: 2018-05-08.

[27] "The norwegian institute of public health: Influenza information." https://fhi.no/en/id/influensa/seasonal-influenza/. Accessed: 2018-06-11.

[28] "The norwegian public roads administration: Open data, api for developers." https://www.vegvesen.no/en/the+npra/about-the-npra/open-data. Accessed: 2018-06-11.

[29] "Tkinter scrollable frame, created by eugene bakin." https://github.com/simondlevy/GooMPy. Accessed: 2018-06-11.

[30] N. Bambrick, "Analyzing time and volume trends in news content." http://blog.aylien.com/analyzing-time-volume-trends-in-news-content/, 2016.

[31] NPRA, "Information about norwegian roads from the national road databank.." https://www.vegvesen.no/vegkart/vegkart/, 2018.