Data Science in Life Science

SS20

Quentin Quarantino

PUBLISHED BY PUBLISHER

BOOK-WEBSITE.COM

First printing, March 2019

Contents

Part 1

1. Introduction

1.1 Goal of the Project

In this weeks project each group member was assigned one overarching topic pertaining to the current Covid-19 epidemic. The current epidemic is globalized, with severe consequences to social, health and economic order. As of today 212 countries are affected, with a total of 4,215,274 confirmed cases and a death toll of 284,672 [41].

1.2 Outcome

Within each topic a short introduction to the general concept is given. The understanding of these concepts is then deepened by real world code examples, showing a glimpse of what is possible in each topic in regards to Covid-19.

2. Topic1: Spreading Models

2.1 Background

Modeling the spread of infectious diseases is not only an essential tool in understanding the transmission rates and the trajectory of future cases but also has a significant influence on the appropriate guidelines to control the course of an epidemic. The approaches towards modeling the spread can range from computational models (e.g. agent-based) to mathematical modeling (e.g. compartmentalized models). In this short introduction we will focus on a SIR-model which is part of the compartmentalized subgroup. These models follow a deterministic pattern where each sub-population is divided into groups. In SIR-models each letter stands for one group: $S = susceptible$, $I = infectious$ and $R = recovered/death$. Then it follows that for each time independent point t the rates for each subgroup can be calculated by:

$$dS/dt = \nu N - \beta SI/N - \mu S$$
$$dI/dt = \beta SI/N - \gamma I - \mu I$$
$$dR/dt = \gamma I - \mu R$$

with γ denoting the time rate of death/recovery, β denoting the number of new infections one case causes per time point t, μ denoting general death rate and ν denoting being the birthrate. hey

2.2 Data and Methods

This code is based on the kaggle notebook from *https://www.kaggle.com/lisphilar/covid-19-data-with-sir-model*. It uses python and a basic framework of libraries e.g pandas, sklearn, datetime etc.. The main data used is from the World Health Organization showing novel corona infections by country. Furthermore supplementary data is used to include the age pyramid for each country. The WHO Data set is preprocessed to include the variables: Date, Country, Province, Confirmed, Infected, Deaths and Recovered. A first visualization shows the global rate of infected, deaths and recovered people (Figure 2.1). Next the growth factor is calculated, which is given by: G_n/G_{n-1} with $G = confirmedcases$. Countries with growth factor higher then one have an increasing number

of cases. In contrast growth factor lesser then one shows a declining number of cases. The actual analysis is done for 5 countries: Italy, Japan, India, USA and New Zealand. Giving one case as an

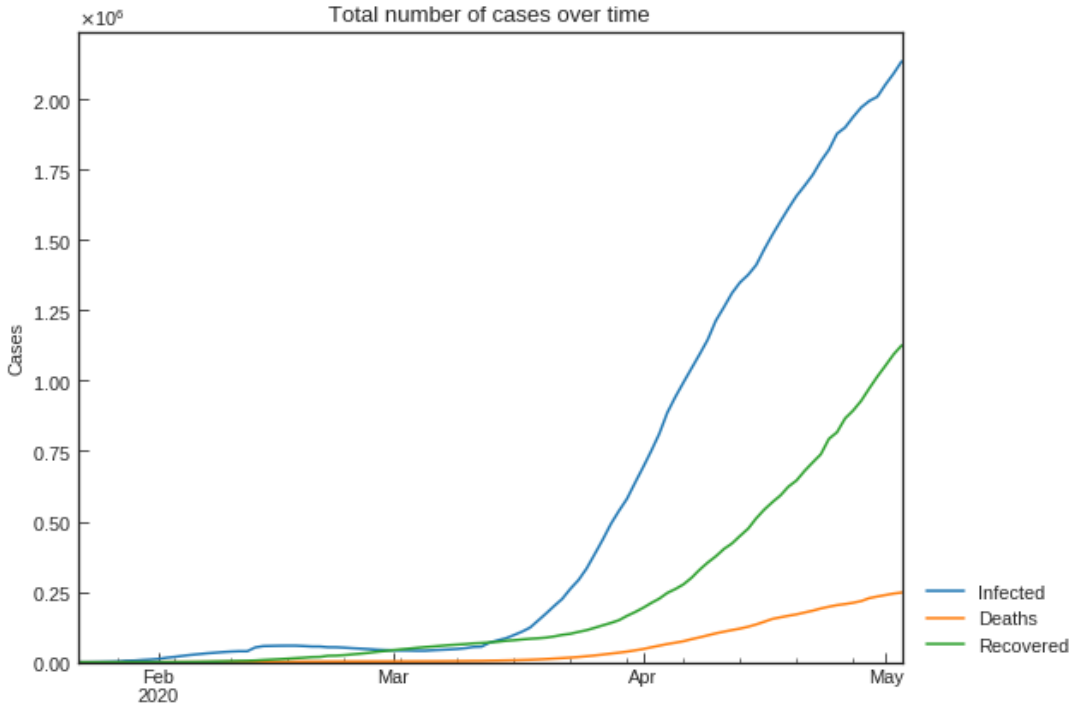


Figure 2.1: Global infections, deaths and recovered people for Covid-19. Infected people follow an exponential trajectory. Recovered people follow a delayed increase, which is due to the long incubation and illness period of 14 days.

example as a first step a S-R trend is plotted (Figure 2.2). It shows the trend of susceptible against recovered people. 5 change points can be identified. Next the SIR-F model parameters are estimated for each change point. As a last step the changes in the p value are contrasted with measures taken by the country. While these results are interesting SIR modeling also has its limitations.

2.3 Discussion & Results

Even though the SIR-Model is one of the most basic infectious disease models available it can show promising results with careful consideration for parameter selection and data processing steps. In the case of Italy 3 measures could be shown to reduce the p value: quarantine of person contacted with positive patients, school closure and lock-down. It also has to be considered that the SIR model is based on very basic assumptions. For example the number of susceptible people is treated as fixed as well as the rates of change.

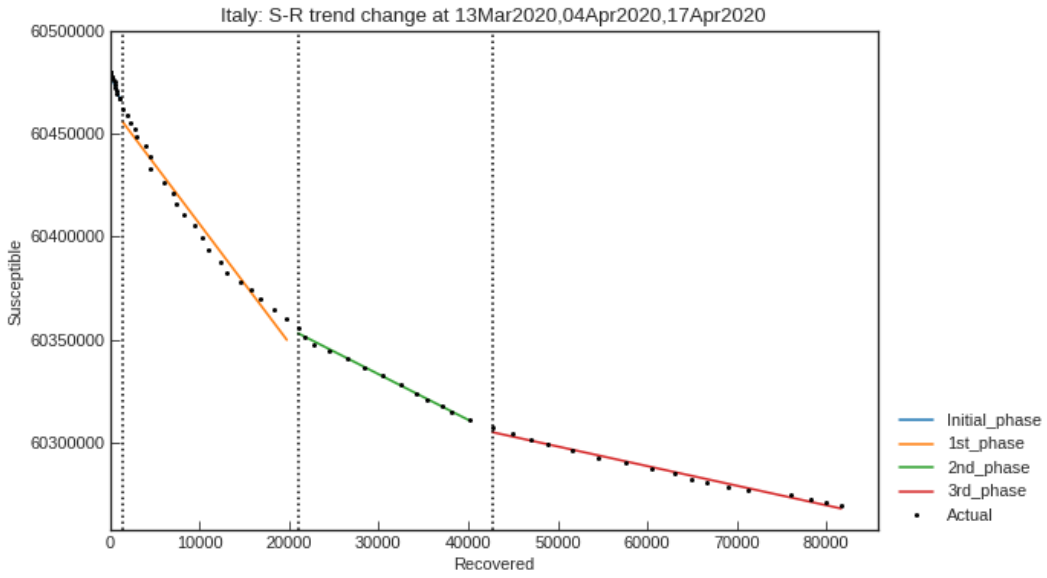


Figure 2.2: Trend of susceptible people versus recovered. 5 distinct change points can be identified.

3. Topic 2: Data-based Time Series Prediction

3.1 Background

Many governments around the world are building their political decisions around the number of current confirmed cases of people infected by COVID-19. Nonetheless, not only the current number of confirmed cases is from greater interest, but also how the virus spreads in the future. One way of forecasting the spread of the virus is by using data-based time series prediction.

3.2 Data and Methods

Therefore machine learning models are calibrated using publicly available data sources like the WHO health report. Time series forecasting can be framed as a supervised learning problem. Other than agent-based spreading simulation such as the SIR model, the models used here do not simulate a population. The forecasting is performed using pythons numpy and sklearn libraries. At first the data is downloaded. Since no data points are missing no preprocessing is performed with the exception of converting integers into date times and reorganizing dataframes. The data contains a wide range of countries with the number of infected people per day starting January 22. A support vector machine model is implemented to forecast the number of infected people. The parameters that have been set can be seen in figure 3.1 in line 2. The test and test training data sets are generated by splitting them without shuffling them, such that the time series is preserved.

```
[ ]  1  # svm_confirmed = svm_search.best_estimator_
     2  svm_confirmed = SVR(shrinking=True, kernel='poly',gamma=0.01, epsilon=1,degree=5, C=0.1)
     3  svm_confirmed.fit(X_train_confirmed, y_train_confirmed)
     4  svm_pred = svm_confirmed.predict(future_forcast)
```

Figure 3.1: Parameters set for SVM Model.

The model has been trained using the first 75 days since January 22. In figure 3.2 it can be seen that the model over estimates the number of confirmed infections by over 1.5 Million.

3.3 Results

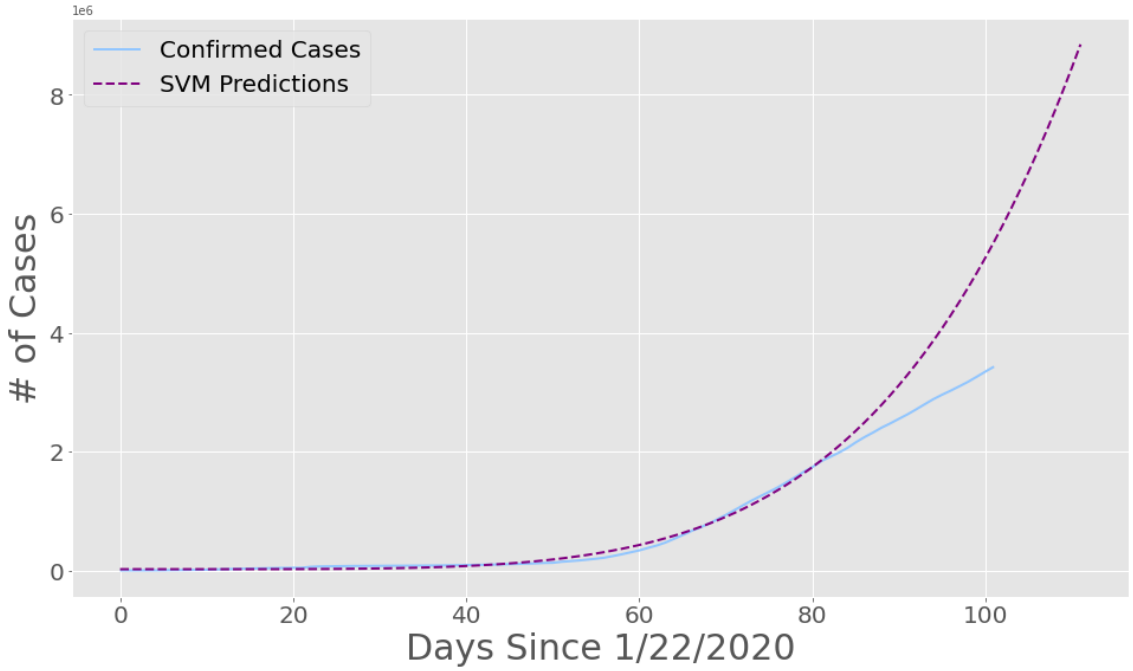


Figure 3.2: Comparison between the observed and the estimated number of infected people. The SVM prediction overshoots the number of cases in contrast to actual confirmed cases.

3.4 Discussion

The shown results are quite underwhelming. This fact has several reasons. One pandemic curves usually increase at the beginning exponentially but then flatten down e.g. because of restrictions in society to decrease the spread of the virus. The model is trained using data from the beginning of the crises where the number of cases rapidly grow. Based on this assumption the estimated number of infected people overshadows the confirmed number. Nonetheless due to different test capacities around the world the estimated might be closer to the real number than it seems to be the case shown in figure 3.2. Still the used model was quite simple and no testing was shown how the parameters were found. A more complex model may gives a better insight to the spread of the virus.

4. Topic 3: Risk Factor Analysis

4.1 Background

The potential dangers of 2019-nCoV have prompted a number of studies on its epidemiological characteristics. It is essential to estimate the number of infections (including those that have not been diagnosed), to be able to analyze the spread of the diseases. To better assess the epidemic risk of 2019-nCoV, among the key parameters to be approximated are the basic reproduction number R0 and the incubation period . Initially we estimate the cumulative number of cases in China outside Hubei province after 23 January, using a time-dependent compartmental model of the transmission dyamics and then we use that number as an input to the global transportation network to generate probability distributions of the number of infected travellers arriving at destinations outside China. Finally using a Galton–Watson branching process to model the initial spread of the virus.

4.2 Data and Methods

The analysis is performed using the python libraries namely numpy, matplot,kiwis solver, scipy and cycler. We computed the risk of the of the individual countries with the selected possible parameters like connectivity and Rloc where Rloc is the local reproduction number of the infection, Getting all the combination of the variables from the data surrounds the neighbour of the china to generate the Heat map.

4.3 Results

Heat map generated gives the information about the outbreak risks as functions of Θ and Rloc, when C = 200,000. The arrows show the directions corresponding to the largest reductions in the risk, which is shown in the figure 4.1

4.4 Discussion

By combining three different modelling approaches helps to assess the risk of 2019-nCoV outbreaks in countries outside of China. This risk depends on three key parameters: the cumulative number of

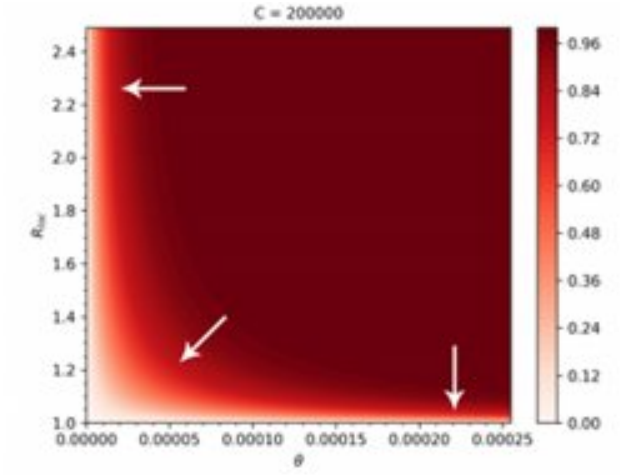
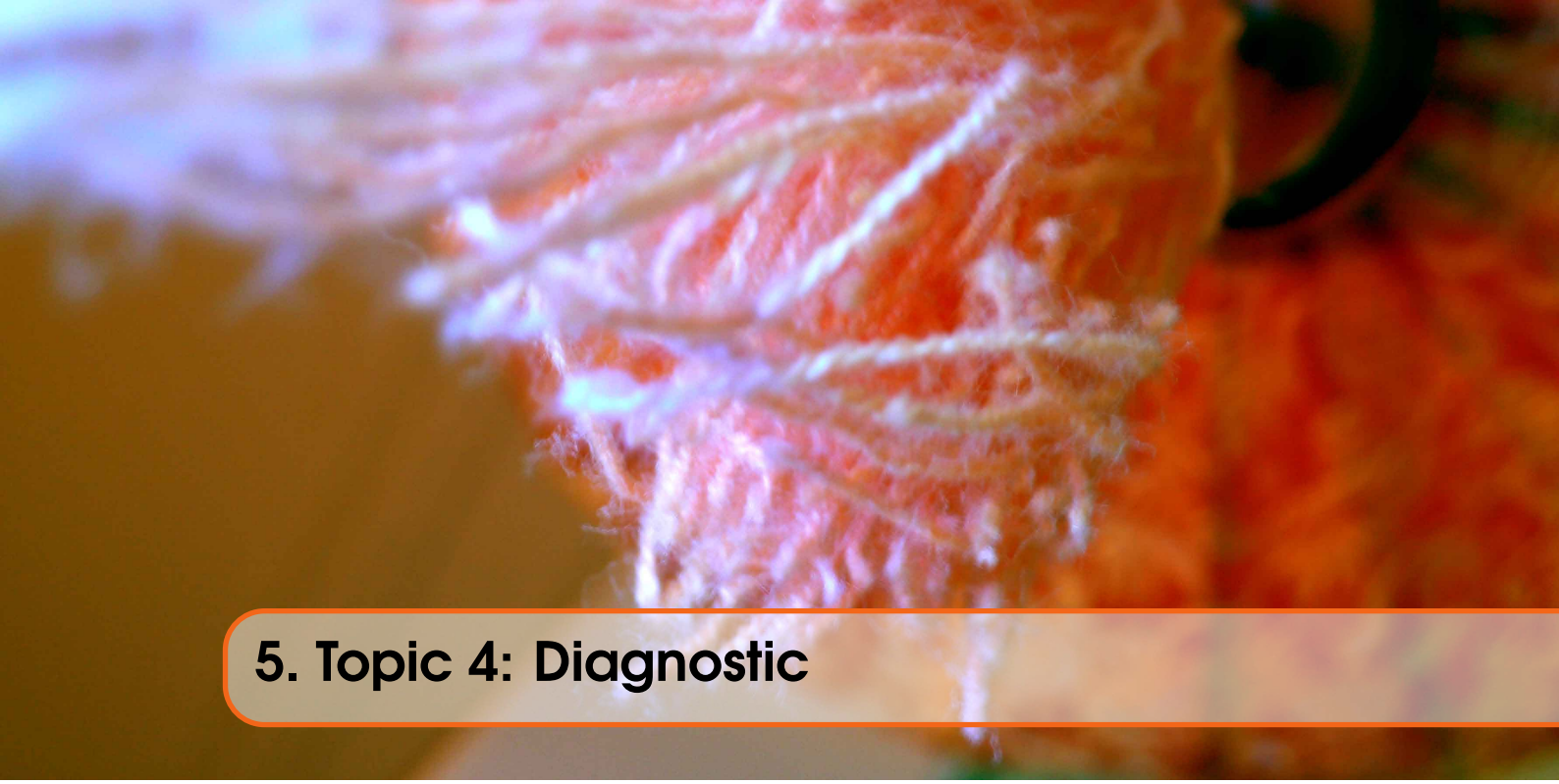


Figure 4.1: Heatmap of the outbreak risks as functions of Theta and Rloc

cases in areas of China which are not closed,the connectivity between China and the destination country, and the local transmission potential of the virus in countries with low connectivity to China but with relatively high Rloc, the most benefecial control measure to reduce the risk of outbreaks is a further reduction in their importation number either by entry screening or travel restrictions. Knowing Rloc and the generation interval are needed not only to have a better quantitative risk estimation, but also for guidance as to which types of control measures may reduce the outbreak risk the most effective.



5. Topic 4: Diagnostic

5.1 Background

The objective of diagnostics is to help effectively diagnose COVID-19 disease. Diagnostics based on RT-PCR-analysis is not very secure due to a high number of false positives. Diagnosis using X-Ray / CT scan images has objective to help effectively diagnose COVID-19 disease with the help of X-Ray/CT scan images in order to improve speed accuracy and scale of diagnosis.

5.2 Data and Methods

The X-Ray Detection method reproduced here is done by training a deep learning model using x-ray images (see Figure 5.1) with TensorFlow and Keras in Python to predict whether a patient has COVID-19. The full list of required tools are here (see Figure 5.2)

5.3 Results

As a first step X-Ray data was downloaded from the source and python scripts were downloaded. In the next step,anaconda was installed as it contains a lot of preinstalled packages. In separate environment all the packages listed (see Fig 5.2) were installed with needed help tools and also other needed packages needed (like cuda toolkit and cudnn) to run tensorflow were installed) Then the step augmentation of given X-Ray images was performed for both classes covid positive and normal respectively. In this step using 70 covid and 28 normal X-Ray data were 5088 covid and 2424 normal augmented data generated. In the next step the model was trained and tested using augmented data. The augmented data were divided in train and test data. 80% (6009 data) of augmented data were used as train data and were included in model and 20% (1503 data) of data were used as test data for predictions. The model was validated for 1503 test data 100 times with 1746 repetitions . Using confusion matrix specificity, sensitivity and accuracy values were estimated and plotted.

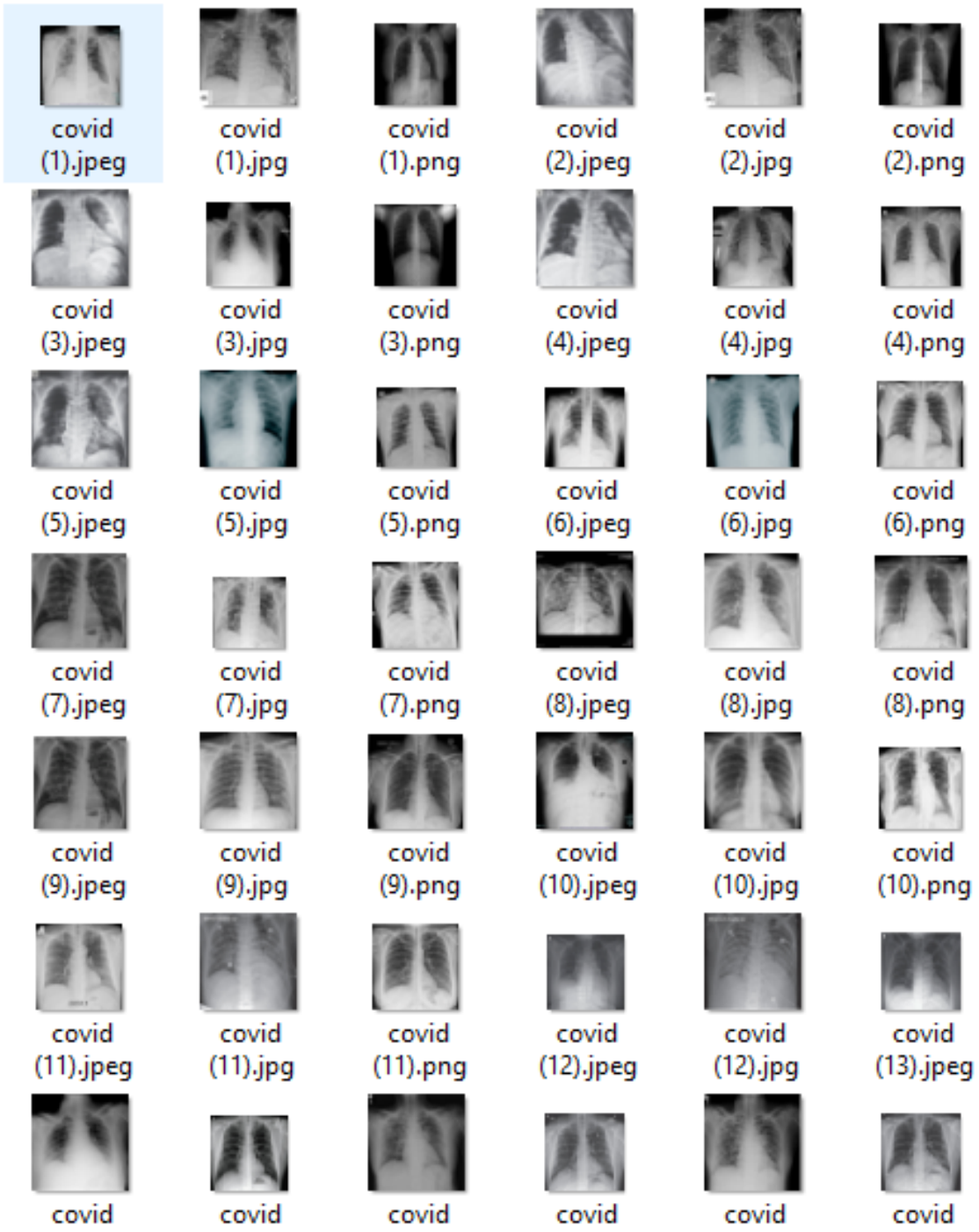


Figure 5.1: Xray data of different patients with lung desease. A white obstruction within the image implactes an infection

```
absl-py==0.9.0
astor==0.8.1
cachetools==4.0.0
certifi==2019.11.28
chardet==3.0.4
cycler==0.10.0
gast==0.2.2
google-auth==1.11.3
google-auth-oauthlib==0.4.1
google-pasta==0.2.0
grpcio==1.27.2
h5py==2.10.0
idna==2.9
imutils==0.5.3
joblib==0.14.1
Keras==2.3.1
Keras-Applications==1.0.8
Keras-Preprocessing==1.1.0
kiwisolver==1.1.0
Markdown==3.2.1
matplotlib==3.2.0
numpy==1.18.2
oauthlib==3.1.0
opencv-python==4.2.0.32
opt-einsum==3.2.0
pandas==1.0.2
Pillow==7.0.0
protobuf==3.11.3
pyasn1==0.4.8
pyasn1-modules==0.2.8
pyparsing==2.4.6
python-dateutil==2.8.1
pytz==2019.3
PyYAML==5.3
requests==2.23.0
requests-oauthlib==1.3.0
rsa==4.0
scikit-learn==0.22.2.post1
scipy==1.4.1
six==1.14.0
sklearn==0.0
tensorboard==2.1.0
tensorflow==2.1.0
tensorflow-estimator==2.1.0
tensorflow-gpu==2.1.0
tensorflow-gpu-estimator==2.1.0
termcolor==1.1.0
urllib3==1.25.8
Werkzeug==1.0.0
wrapt==1.12.1
```

Figure 5.2: List of required tools for the python script

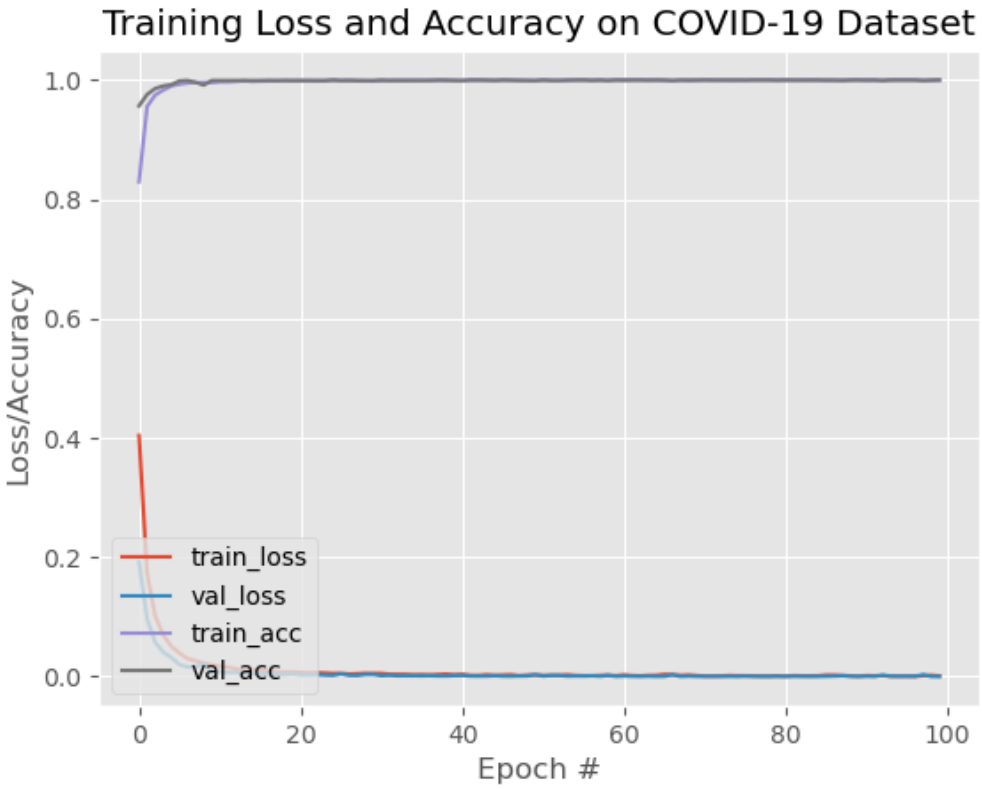


Figure 5.3: Plot of validation loss, training loss and accuracy. After 10 epochs a limit is reached

5.4 Discussion

Approach based on deep Learning described here is a very promising tool for Covid-19 detection in lungs. But on the other hand it is very time consuming. All the steps of this pipeline are very time consuming. Augmentation of images took 2.5 hours. Training and testing using model took about 40 hours. The accuracy, specificity and sensitivity are very high (Figure 5.3) and prove that this approach is very useful.

6. Topic 5: Origin Analysis

6.1 Background

Phylogenetic analysis aims to reconstruct phylogenies both for a group of species and also for the individuals within those species. For pathogens, we are interested in the evolution of the diseases at the genetic level, and what this can tell us about their past and present diversity [38]. An important fact about the Coronaviviridae family is that it's member tend to "jump" from one species to another. When the transmission occurs from a non-human host to a human host it is called zoonosis [11]. The determination of the most recent common ancestor of the human SARS-CoV-2 and the zoonotic transmission can provide important information about biological features, key mutations and properties of the virus. A detailed understanding of how an animal virus jumped species boundaries to infect humans will help in the prevention of future zoonotic events. [2].

6.2 Data and Methods

We will compare the genetic sequence of SARS-CoV-2 with other viruses of the Coronaviridae family in different hosts. The following analysis is based on a Github repository of Simon Burgermeister [8]. Six complete genomes were considered, whose names and hosts are listed in Table 6.1. The sequence data (fasta files) were downloaded from the NCBI Virus public library [15]. To compare the genetic sequences, a multiple sequenced alignment needed to be performed. Clustal Omega is a software that uses seeded guide trees and HMM profile-profile techniques to generate alignments between multiple sequences. Unfortunately, my local computer was not able to compute the alignment due to RAM exceedance. Therefore, I submitted a request to the online version of Clustal Omega [26]. Based on the resulting alignment, a distance matrix was calculated with the *TreeConstruction* package from Biopython. Afterwards, the same package was used to create the phylogenetic tree base on the UPGMA algorithm.

6.3 Results

The resulting phylogenetic tree (Figure 6.1) shows that our human SARS-CoV-2 sequence is most similar to the SARS-like coronavirus sequence of the Rhinolophus (horseshoe bat) with a similarity

Accession number	Host	Description
MN996528	H. Sapiens	Human SARS-CoV-2
NC_019843	H.Sapiens	Human MERS-CoV
JQ065048	Anatidae	Ducks, geese and swans
MG772934	Rhinolophus	Horseshoe bats
NC_034972	Apodemus chevrieri	Rodent
KX38909	Gallus gallus	Chicken
MT084071	Manis javanica	Pangolin

Table 6.1: Considered Coronaviridae strains and hosts.

of 96%. The host with the next similar sequence is the Manis javanica (Pangolin) with a similarity of 0.89% between their genomes. The human MERS-Cov genome and the SARS-CoV-2 genome share only a sequence similarity of 0.74%.

6.4 Discussion

As many early cases of COVID-19 were linked to the Huanan market in Wuhan [43], it is possible that an animal source was present at this location. Given the similarity of SARS-CoV-2 to bat SARS-CoV-like coronaviruses, it is likely that bats serve as reservoir hosts for its progenitor. Although the similarity of 96% to the coronavirus sequence hosted by the Rhinolophus, Andersen et al. [2] identified that its spike protein diverges in the receptor binding domain (RBD), which suggests that it may not bind efficiently to the human ACE2 receptor. Furthermore, it is assumed in this and other studies [20, 23] that an intermediate host was probably involved.

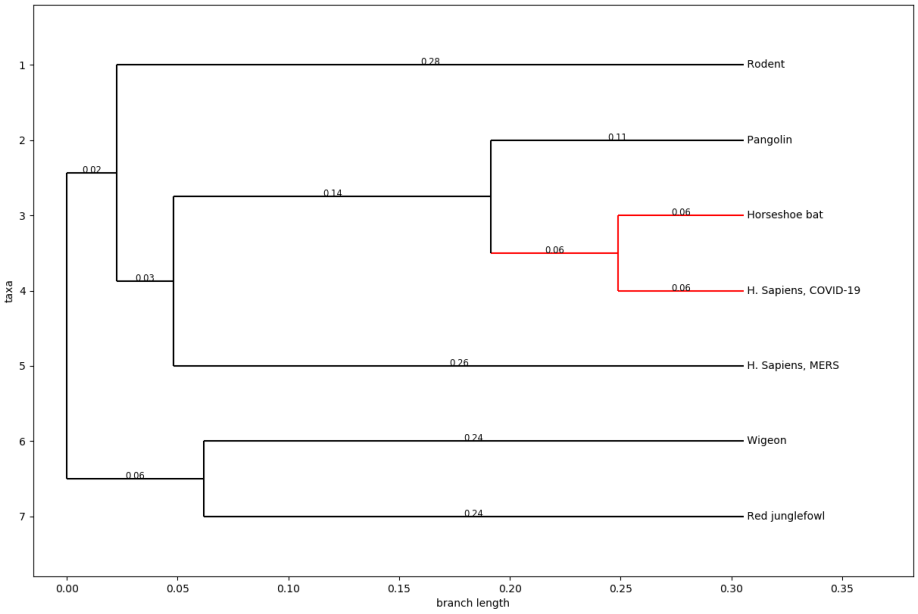


Figure 6.1: Phylogenetic tree of the origin detection analysis.

II

Part 2

7. Introduction

7.1 Goal of the Project

The overwhelming amount of daily published papers correlated to the corona virus makes it difficult, even for health professionals, to keep up with new information about the virus. One way of managing the flood of information is by clustering them according to their topics to simplify the search. Therefore, we have performed a cluster analysis of the CORD-19 dataset, which contains roughly 60.000 articles.
After parsing the body of each article in the dataset, the extracted information is transformed into a feature vector. We afterwards apply dimensionality reduction using PCA and performed a k-means clustering. Subsequently, t-SNE is applied to project the original feature vector into two dimensions such that clusters become visible in the two dimensional space.
Each course participant selected five scientific papers that cluster in the same group as the article they introduced in *Part* I. The submissions have been used to create a new dataset. K-meanswas also applied to this dataset, to determine the cluster assignments and investigate patterns in the data. Finally, the selected articles of *Part* I were added to the CORD-19 dataset. The clustering was redone to see if the papers will be assigned to the expected clusters. In addition to re performing the clustering, two methods for selecting the best k value and two distance metrics were compared: Silhouette Scoring vs Distortion and Euclidean vs Cosine Similarity.

7.2 Outcome

The five papers we selected in *Part* I were clustered into 3 different groups, in which three of the papers have been assigned to the same cluster. Comparing both, the method of choosing k by elbow point or silhouette scoring and the distance metrics euclidean and cosine similarity, we determined that silhouette scoring and euclidean distance performed better. 10 clusters with unique topics were found (Table 8.2).

8. Tasks

8.1 Part1

The literature clustering pipeline started with the data import of the CORD-19 dataset. Since we wanted to perform the calculations in a Google Colab notebook, we decided to create an API connection to the kaggle database. Using the API, we were able to download and unzip the dataset on our personal Google drive the fastest way possible. The resulting metadata dataframe listed 59.887 entries of coronavirus related publications.
The metadata information are subsequently merged with the body text of the papers that are stored in separated json files. Due to partially missing information only 43.331 entries of the metadata could be merged with the json files. To get an overview of the average text length of the abstracts and the body text information (on which the clustering will be performed) the overall and unique number of words were calculated. The result was an average abstract length of 157 words and an average body text length of 4.528 word (1376 unique). Since the data was uploaded by many different sources, duplicates were present in the dataset. These need to be filtered out such that 30.960 publication remained in the set. The subsequent calculation steps of the pipeline will require very high computing resources. Therefore, we randomly subsamples (seed=42) the dataset to a maximum of 10.000 instances. Unfortunately, we noticed afterwards that both, entries containing null values (1073) and non-english publications (242) were still present in the data. Since these would massively reduce the interpretability of the clustering result, they were also dropped. The final dataset consisted then of 8685 entries.
Another applied proprocessing step was to detect and remove stop words. These are common words in the written text, that do not contribute to the content and act as noise in the clustering procedure. The *spacy* package was used to determine the stopwords. Additionally, a predefined list of stopwords was appended to the list, that contained frequently used words of scientific publication in general. The last step of the preprocessing was to vectorize the cleaned data. Hereby,the string formatted data is converted into a vector-based measure of how important each word is to the instance out of the literature as a whole usind the *tf-idf* package. This method creates a very high feature space and since a clusering by k-means need to be performed, a Principle Component Analysis (PCA) was applied to reduce the amount of feature by simultaneously keeping 95% of the

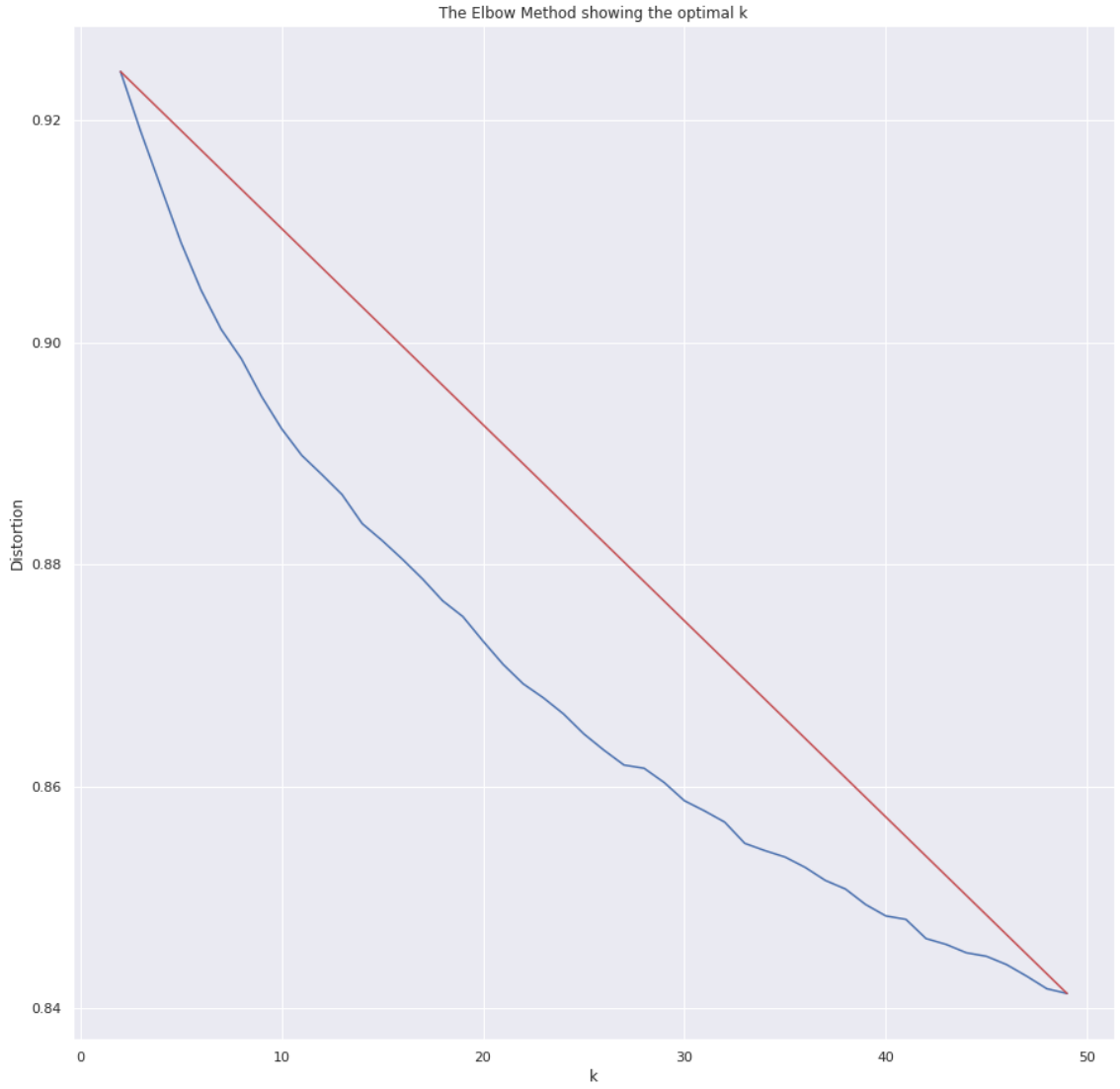


Figure 8.1: The figure shows an elbow plot of the k-means clustering the the distortion on the y-axis and the number of clusters on the x-axis. A clear elbowpoint cannot be identified.

data's variance and immensely reducing the algorithm's runtime. The best k number of clusters was determined by iterating through different values of k from two to 50. The resulting elbow plot (Figure 8.1) shows the elbow point at k=27, which is subsequently used as the best number of clusters.
A t-Distributed Stochastic Neighbor Embedding (t-SNE) was used to reduce the high dimensional features vector to two dimensions. This step provides to possibility to represent the clustering result in a plain coordinate system. The aim of the entire pipeline was to create an interactive bokeh plot. To create the plot, the results of all previous calculations are brought together. The location of each paper on the plot is determined by t-SNE while the label (color) is determined by k-means. Interestingly, the assignments match each other very well, even though they were calculated separately (Figure 8.2). Now, the clusters are calculated, but the information about the kind of papers, that are matched together is still missing. To solve this task, a Latent Dirichment Analysis (LDA) was performed to model the most important topics for each cluster. This information is also included in the final bokeh plot.

Figure 8.2: The plot shows the clustering result with the t-SNE positioning and the k-means labels. Various distinct clusters can be identified, indicating a good separation via k-means clustering.

8.2 Part2

In this task we were supposed to add the metadata information of the selected papers from *Part* I to the CORD-19 dataset and perform the clustering again. We wanted to find out if this approach improves and facilitates the search of papers for a specific topic field. In order to solve this task, we tried to generate a dataframe using the *pyPDF2* package to extract the metadata from the pdf files of the articles. Unfortunately, it did not work for all pdfs, because they did not contain uniform metadata fields. For this reason we decided to create a csv file and added all relevant metadata fields manually (Table 8.3). The manually created table was then added to the CORD-19 dataset (Figure 8.4).

Link	Title
https://www.ncbi.nlm.nih.gov/pubmed/32276116	Rapid and visual detection of 2019 novel coronavirus (SARS-CoV-2) by a reverse transcription loop-mediated isothermal amplification assay
https://www.nature.com/articles/s41598-018-37483-w	A method to identify respiratory virus infections in clinical samples using next-generation sequencing
https://www.tandfonline.com/doi/full/10.1586/14737159.2014.888313	Advances and challenges in biosensor-based diagnosis of infectious diseases
https://ann-clinmicrob.biomedcentral.com/articles/10.1186/1476-0711-7-18	Predicting the sensitivity and specificity of published real-time PCR assays
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3522074/	Application of Molecular Diagnostic Techniques for Viral Testing

Table 8.1: Papers to diagnostics

	A	B	C	D	E	F	G	H
1	paper_id	doi	abstract	body_text	authors	title	journal	abstract_summary
2	week_2_spreading_models	10 3390/ijerph 16234683	Infectious diseases are an important cause of human death The study of the pathogenesis spread regularity and development trend of infectious diseases not only provides a theoretical basis for future research on infectious diseases but also has practical guiding significance for the prevention and control of their spread In this paper a controlled differential equation and an	Infectious diseases are diseases that can be transmitted from person to person from person to animal or from animal to animal after proto-microorganisms and parasites infect human beings or animals [1–3] Infectivity epidemic and uncertainty are the three main characteristics of infectious diseases A thorough study of the spread	Bin Sheng Sun Gengxin Chen Chih-Cheng	Spread of Infectious Disease Modeling and Analysis of Different Factors on	International Journal of Environmental Research and Public Health	something
3	week_2_risk	10 3390/jcm90 20571	We developed a computational tool to assess the risks of novel coronavirus outbreaks outside of China We estimate the dependence of the risk of a major outbreak in a country from imported cases on key parameters such as: (i) the evolution of the cumulative number of cases in mainland China	A cluster of pneumonia cases in Wuhan China was reported to the World Health Organization (WHO) on 31 December 2019 The cause of the pneumonia cases was identified as a novel betacoronavirus the 2019 novel coronavirus (2019-nCoV recently renamed as SARS-CoV-2	Boldog Péter Tekeli Tamás Vizi Zsolt Dénes	Risk Assessment of Novel Coronavirus COVID-19 Outbreaks	Journal of Clinical Medicine	something
4	week2_forecasting	https://doi org/10 1016/j idm 2020 03 002	Based on the official data modelling this paper studies the transmission process of the Corona Virus Disease 2019 (COVID-19) The error between the model and the official data curve is quite small At the same time it realized forward prediction and backward inference of the epidemic situation and the relevant analysis help relevant countries to	At the end of 2019 the new coronavirus (COVID-19) spread widely in China and a large number of people became infected At present the domestic outbreak has been effectively controlled while the new coronavirus is spreading rapidly in other areas Currently Europe has become the center of	Li Lixiang Yang Zihang Dang Zhongkai Meng Cui Huang	Propagation analysis and prediction of the COVID-19	KeAi	something
5		https://ww	Testing for COVID-19 has been unable to keep up	Identifying who has the COVID-19 virus is	Hall	Finding		

Figure 8.3: csv-table

We prepossessed the dataframe and applied the previously describes clustering pipeline. Finally, the cluster membership of each paper from *Part* I could be identified. The titles of papers from the cluster were saved as .csv-file respectively (see Figure 8.5).

It could be figured out that three papers (spreading models, databased time-series prediction and risk factor analysis) from *Part* I belong to the same cluster 6 (Figure 8.6). The paper related to diagnostics was assigned to cluster 15 (Figure 8.7) and the origin analysis article was a member of cluster 9 (Figure 8.8).

2.3 Appending metadata of papers from week2 ¶

Loading metadata of papers from week2 from csv file as dataframe, preprocessing (stripping whitespace, removing char, adding word counts of abstract, body text and unique words) and appending metadata to a general dataframe containing all the papers), dropping duplicates, data summary

In [39]:
```
import pandas as pd
df_papers_week2 = pd.read_csv('C:/Users/Natalja/shared_folder/DSinLS20/week3/Week2_Papers.csv', sep=';',  dtype={'paper_id' : str
for column in df_papers_week2:
    df_papers_week2[column]= df_papers_week2[column].apply(lambda x: x.strip())
df_papers_week2
df_papers_week2['title']=df_papers_week2['title'].str.replace('<br>',' ')
df_papers_week2.drop_duplicates(subset='title', keep='first', inplace=True)
df_papers_week2['abstract_word_count'] = df_papers_week2['abstract'].apply(lambda x: len(x.strip().split()))  # word count in abs
df_papers_week2['body_word_count'] = df_papers_week2['body_text'].apply(lambda x: len(x.strip().split()))  # word count in body
df_papers_week2['body_unique_words']=df_papers_week2['body_text'].apply(lambda x:len(set(str(x).split())))  # number of unique wo
df=df_papers_week2.append(df)
df.drop_duplicates(['abstract', 'body_text'], inplace=True)
df['abstract'].describe(include='all')
```

Out[39]:
```
count     10005
unique     7205
top
freq       2794
Name: abstract, dtype: object
```

Figure 8.4: Appending metadata of week2 papers to a data frame containing metadata to CORD-19-research-challenge

Identifying of cluster for each week2 paper

using helper function save_cluster_week2_titles clusters of each paper from week2 identified and paper titles of this cluster will be saved in extra .csv-file

In [113]:
```
def save_cluster_week2_titles(title, table):
    cluster=table.loc[table['title']== title, 'y'].values
    #print(cluster)
    cluster_value=cluster[0]
    print('Cluster {}'.format(cluster_value))
    cluster=table.loc[table['y'] == cluster_value, 'title']
    nameId_week2=table.loc[table['title'] == title, 'paper_id'].values
    nameId_week2=nameId_week2[0]
    print(nameId_week2)
    savePath = 'C:/Users/Natalja/shared_folder/DSinLS20/week3/df_covid_cluster_topic_{}.csv'.format(nameId_week2)
    cluster.to_csv(savePath, index = False)
for index, row in df_papers_week2.iterrows():
      print ('Title of week 2 paper is {}'.format(row['title']))
      save_cluster_week2_titles(row['title'], df)
```

```
Title of week 2 paper is Spread of Infectious Disease Modeling and Analysis of Different Factors on Spread of Infectious Diseas
e Based on Cellular Automata
Cluster 6
week_2_spreading_models
Title of week 2 paper is Risk Assessment of Novel Coronavirus COVID-19 Outbreaks Outside China
Cluster 6
week_2_risk
Title of week 2 paper is Propagation analysis and prediction of the COVID-19
Cluster 6
week2_forecasting
Title of week 2 paper is Finding COVID-19 from Chest X-rays using Deep Learning on a Small Dataset
Cluster 15
week2_diagnostics
Title of week 2 paper is Identification of a new coronavirus
Cluster 9
week2_phylogenetic_analysis
```

Figure 8.5: Clustering and identifying cluster of week2papers

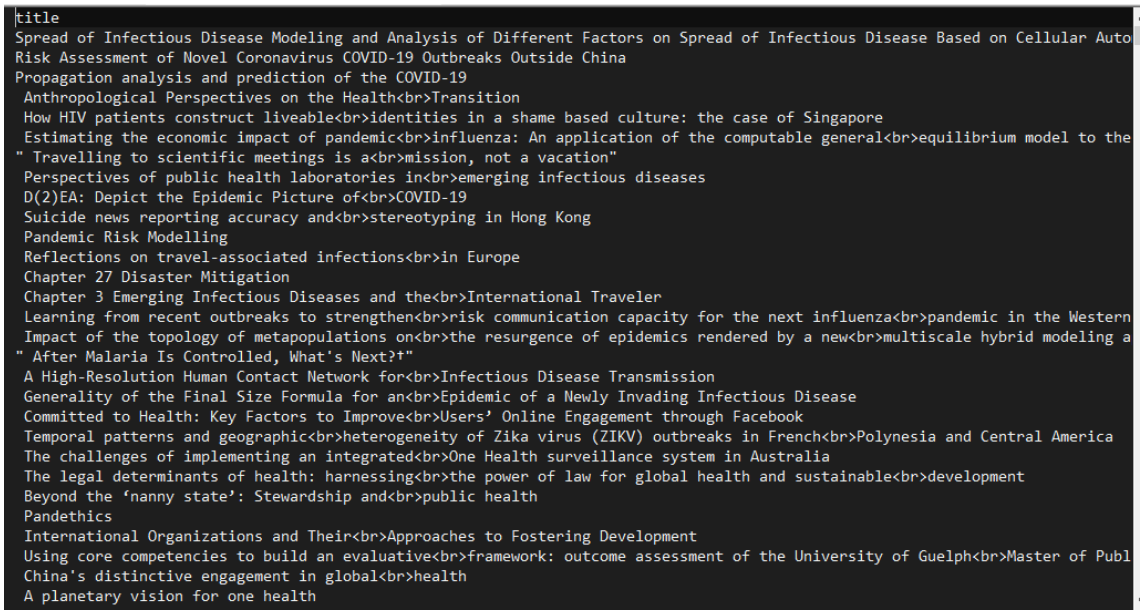


Figure 8.6: Cluster 6: titles of related papers to the papers from week2 about risk factor analysis, spreading and forecasting

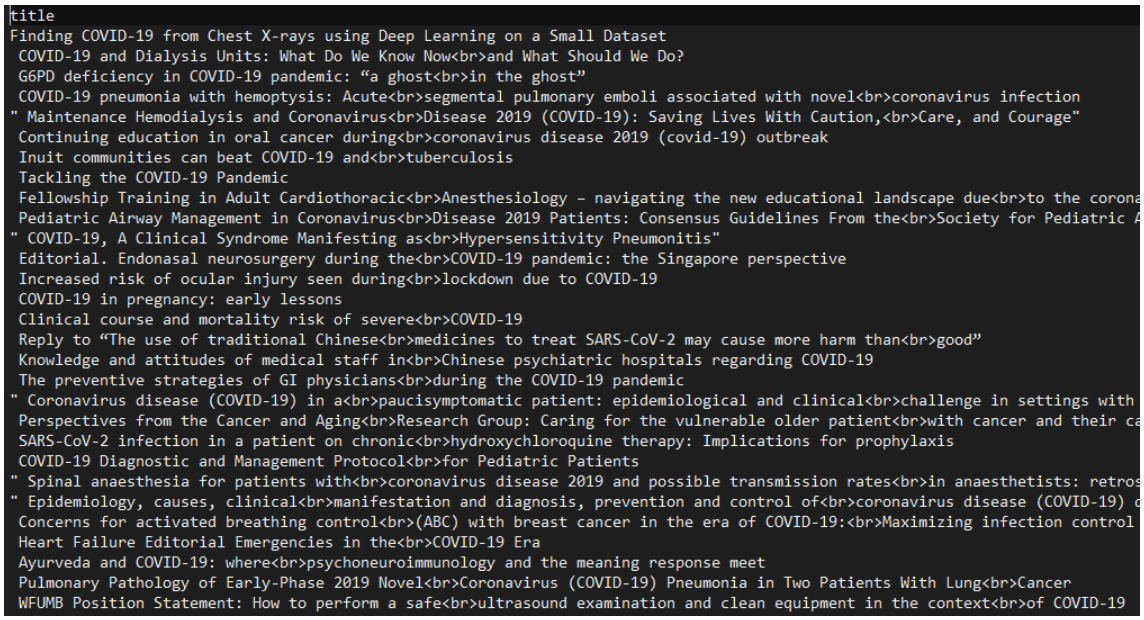


Figure 8.7: Cluster 15: titles of related papers to the paper from week 2 about diagnostics

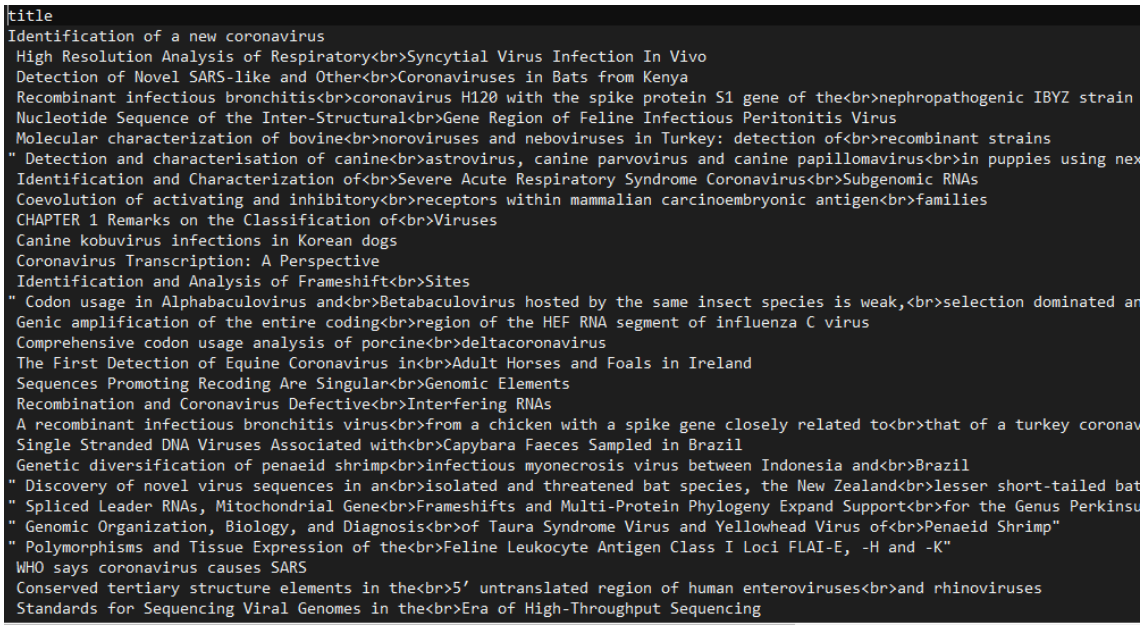


Figure 8.8: Cluster 9: titles of related papers to the paper from week 2 about origin analysis

As it can be seen the titles of the articles in the identified clusters are relates to the papers from *Part* I. Therefore, it can be a helpful approach to find a related papers.

8.3 Part3

8.3.1 Loading in the Student Dataset

After adding only the information about the five papers that we have chosen in *Part* I to the CORD-19 dataset, we created a new dataset containing all submitted papers by the course participants. It turned out that opening each of the 60.000 json files in the CORD-19 dataset to filter those jsons that do not match one of the submitted articles is too time-consuming. Title names have been stripped since traveling whitespaces result in missmatches. The runtime was dramatically decreased by first joining the course dataset with the metadata. The included paper id ("sha") was then used to search in the file names of the jsons for the papers of interest. This leads to a runtime reduction from several hours to less than 3 minutes. We matched 177 of the 195 submitted articles. And after removing duplicated entries our new dataset based on the submissions of the course participants contains 146 papers.

8.3.2 Preprocessing, Clustering and Results

Next, we proceeded with the preprocessing and final clustering step. For the preprocessing, we followed the previously describes pipeline of the kaggle notebook closely. We removed common stopwords as they act as noise. Next, a vectorizer with a noise filer of 2^{12} is applied, counting words and scoring less frequent words higher. Afterwards, the dimension of the dataset are reduced by PCA from over 4069 to 119. For the clustering we deviated from the kaggle notebook. First we tested a second method to determine the best k value. This was done by computing the silhouette score (Figure 8.9). Second we added another preprocessing step, to transform the data to a unit vector of 1, thus using the equivalent of cosine similarity as distance metric. After carefully comparing both methods with and without cosine similarity, one of elbow distortions and the other of silhouette scoring, we determine k by silhouette scoring and euclidean to be the best method (Figure 8.10). Thus we proceeded with k of 18. We choose 18, because it showed a noticeable bump in scoring, but is still a reasonable estimate based on the chosen paper data set by students. After running a t-SNE visualisation (Figure 8.11) and a Keyword extraction per cluster using LDA, with a minimum number constraint of 10 topics per cluster, we found 10 distinct clusters with an overarching topic. 8 clusters were removed due to insufficient number of data points. The results can be found in the Table 8.2.

The assignment of topics to each unique cluster was surprisingly easy, indicating a meaningful result. Some problems could be identified, for one some clusters had not enough articles assigned to them. Thus we could stipulate that the chosen k value of 18 is too high, rerunning the clustering at a lower value might give better results. On the contrary some clusters with high assignment could be identified (cluster 11). This could mean two things: Not much variation in the chosen cluster topics from the students or a more granular clustering is needed.

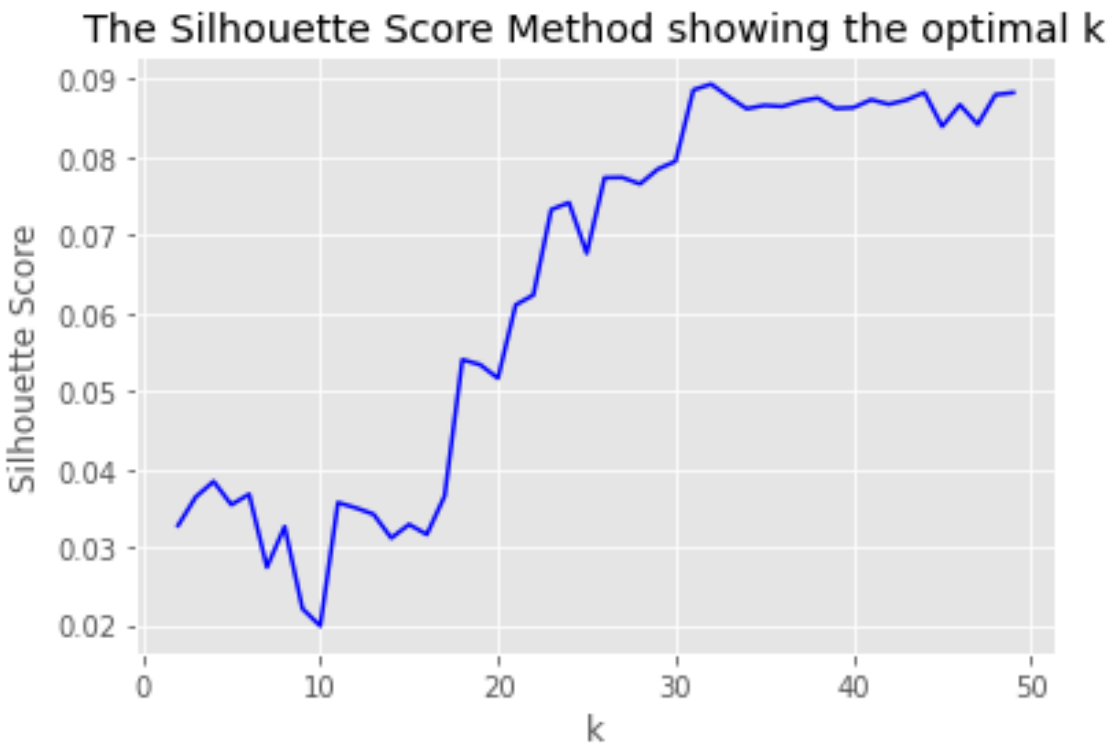


Figure 8.9: Silhouette scoring method for k-means. At the cluster point 17 a significant bump in scoring can be seen.

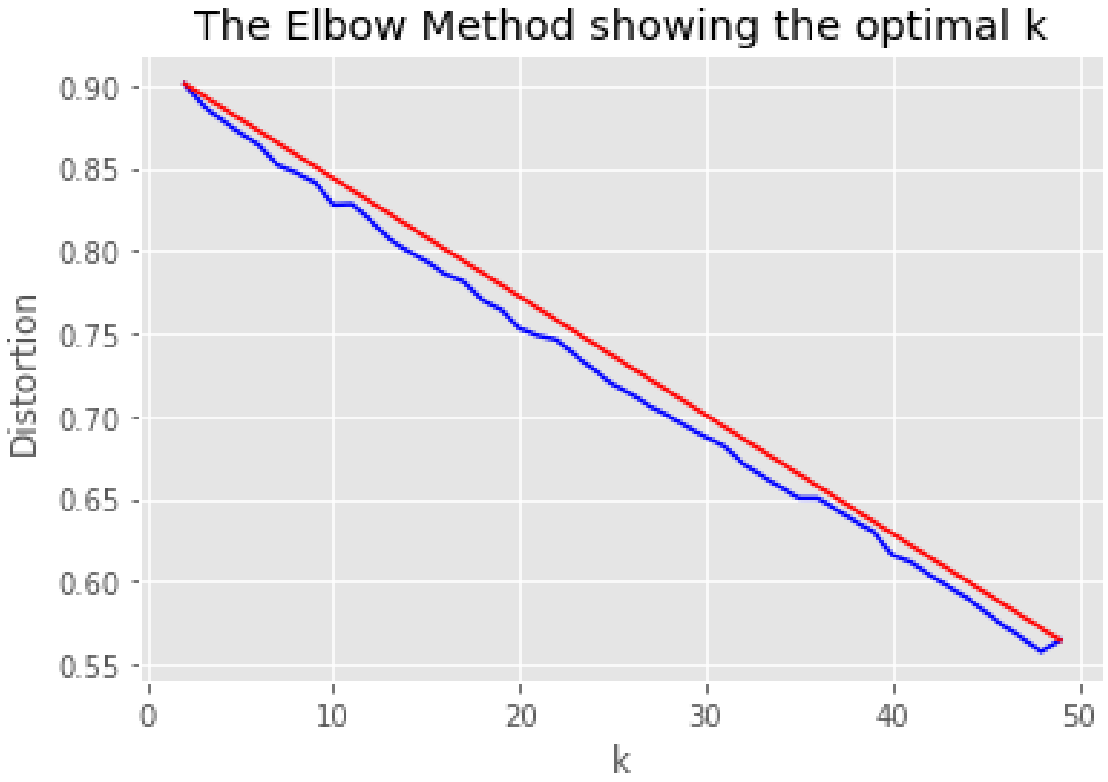


Figure 8.10: Distortion scoring method for k-means. Due to the low difference in scoring an almost linear line is produced. Thus in this case the method is unsuited to determine the optimal k for clustering

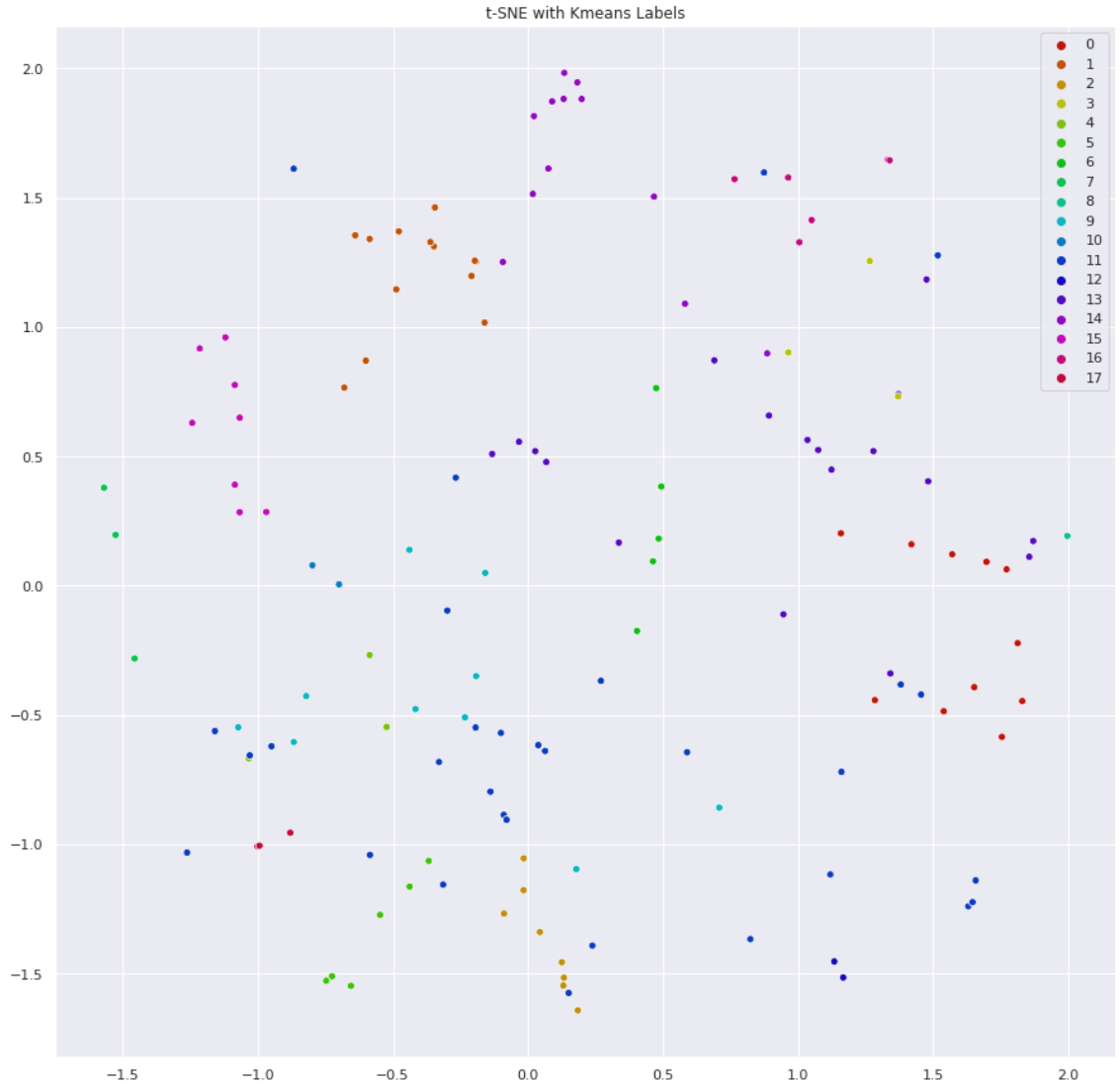


Figure 8.11: 2 dimensional tsne visualisation of the data set. Due to the low presence of articles the projection seems to be sparse. Still some clusters can be determined like: Top middle cluster 14: virus detection or middle left cluster 15: compartmentalized models. Big clusters like 11: modeling of spread are not clumped together.

Cluster	No. Articles	Assigned Topic	Keywords
0	11	phylogenetics	'sars', 'set', 'gene', 'rate', 'orf', 'recombination', 'datum', 'frequency', 'region', 'distance'
1	13	study of initial outbreak	'symptom', 'hospital', 'sars-cov-', 'china', 'country', 'mortality', 'use', 'respiratory', 'risk', 'evidence'
2	8	network-modelling of spread	'community', 'degree', 'threshold', 'mix', 'heterogeneity', 'group', 'node', 'use', 'transmission', 'size'
5	6	network-modelling of spread	'mix', 'wave', 'estimate', 'outbreak', 'total', 'community', 'human', 'delay', 'overall', 'additional'
9	11	disease forecasting	'case', 'disease', 'outbreak', 'estimate', 'influenza', 'process', 'interval', 'forecast', 'day', 'peak'
11	31	modelling of spread	'parameter', 'sequence', 'disease', 'method', 'change', 'city', 'spread', 'network', 'value', 'interaction'
12	18	origin detection	'case', 'camel', 'sars-cov-', 'human', 'protein', 'isolate', 'healthcare', 'bat', 'viral', 'sars-cov'
14	12	virus detection	'sars', 'sample', 'lung', 'serum', 'finding', 'detection', 'virus', 'care', 'study', 'swab'
15	8	compartmentalized modelling	'rate', 'risk', 'model', 'outbreak', 'death', 'datum', 'day', 'virus', 'state', 'patient'
16	6	diagnostics	'pcr', 'rsv', 'sequence', 'pneumonia', 'age', 'child', 'rhinovirus', 'associate', 'young', 'presence'

Table 8.2: Results of the clustering with unique topic assignment based on the first 10 keywords. 8 clusters are omitted due to low assignment of articles.

8.3.3 Creating a word cloud

Finally, we generated a basic word cloud. Here, we took the extracted keywords from the final clustering and used the word cloud package. The package provides a basic understanding of the word cloud with the use of some simple python libraries like numpy, pandas, matplot and pillow. The figure 8.12 shows the visualized wordcloud. Words like sars, virus and cov are bold and large which shows that the frequency of usage of these words is high and denotes their importance of it during the recent times. While these wordclouds are easy to look at, not much useful information can be extracted.

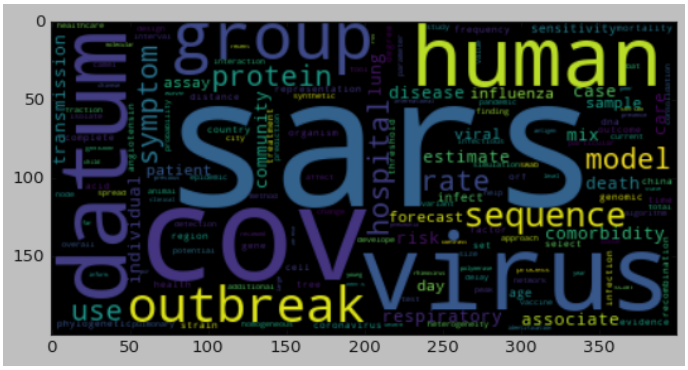


Figure 8.12: Word cloud of most frequent words. Most frequent words are clearly visable e.g. SARS and Human. More specific categories are less visible e.g. protein and estimate.

Part 3

9. Introduction

9.1 Background

Dating back to the 1920's [18] for its first inception, a classical approach towards modeling the spread of diseases in epidemiology are SIR-models. SIR-Models are based on the idea of compartmentalization, where the dynamics of an epidemic are studied by dividing the populations into distinct subgroups. The name SIR is an abbreviation for its most simple form: **S** standing for **s**usceptible (i.e. individuals not yet infected), **I** standing for **i**nfectious (i.e. infected and infectious individuals) and **R** standing for **r**ecovered (i.e. individuals which are not infected and infectious anymore). Each compartment can be understood as a state, with a flow from one state to another. By using an equation of the simple form $N = S + I + R$ the whole population N stays static, while the ratios between the states change. Each state can than be modeled by differing differential equations, thus describing the fluctuations of each state at different timesteps t. Furthermore it is possible to freely add compartments by branching out from current ones, thus making the model adaptable to very different scenarios of an epidemic.

9.2 Goal of the Project

The objective of this weeks project is the application of a SIR-model on current Covid-19 case data taken either from a city (e.g. Berlin) or national (e.g. Germany) scale. The model itself is extended beyond the simple case by integrating two new states (Exposed, Dead) to the model and studying the impact of independent features (ICUbed-capacity, Age, Smoking and Gender) on the epidemic. By fitting the model to actual case data, possible projections can be made. Furthermore different scenarios such as lockdown, reducing social contacts and wearing masks, are explored by simulating their effect on the fitted model. Each prevention method is simulated over different periods and in combination with and without wearing a mask on top.

9.3 Outcome

A simple SIR model was implemented while exploring differences in rate of infection and time to recovery. Extending the model with the independent features age, smoking and gender significantly

altered the α values (i.e. rate of death) with smoking increasing the factor by more then double from 0.07 to 0.16. Two compartments were added, simulating the incubation period and extending the recovered individuals with death. A simulation with ICU bed capacity was performed, reaching the cap after 50 days. The fitting of the model to the the data of Germany resulted in some realistic numbers as the range R0 values was kept within the possible range and showed real declined behaviour. Other predicted curves for susceptible number, exposed, dead and recovery number were also kept within the realistic bounds. The performed simulations suggests that both the duration as well as the intensity of the restrictions plays an important role when fighting the outbreak of corona. The usage of masks is even more important for minor social reduction scenarios.

10. Tasks

10.1 A simple SIR model

The simple SIR-model includes three subgroups: ***S****usceptible*, ***I****nfectious* and ***R****ecovered* cases.

Figure 10.1: A flow diagramm showing the state transtions between the subgroups. The whole population is constant, while the flow is unidirectional. (Taken from [34])

Because each compartment can be understood as a state, we can visualize their transitions as a flow diagram (Figure 10.1). The variables above the state transitions describe the rates of individuals switching between the different compartments. For susceptible people becoming infected we introduce the factor β denoting the rate of one person infecting another person and for infected people becoming recovered we introduce γ denoting the rate of infected people developing immunity any given day. Thus we can infer three differential equations for each different subgroup, with N denoting the whole population:

$$dS/dt = -\beta * S * \frac{I}{N}$$
$$dI/dt = \beta * S * \frac{I}{N} - \gamma * I$$
$$dR/dt = \gamma * I$$

By integrating these equations over the time point t using the *odeint* function from the *sklearn* package in python3, we can develop a model simulation of the developing compartments for any initial starting conditions. As an example, we can compare the impact of tripling the rate of infection by using a β value of 1.0 compared to 3.0 (Figure 10.2 and 10.3). The β value of 3.0 shows a drastic change. All three compartments are shifted to the left, while the curve of infectious people has a much higher and steeper initial incline, which in turn results in a fast drop of susceptible people. In contrast reducing the γ value from $\frac{1}{4}$ to $\frac{1}{8}$ (Figure 10.4), results in a much lower incline of recovered people and much longer period of infectious people, as shown by the higher maximum of the yellow line. This shows the importance of both, the β and γ value. Thus, it is of high value to determine the ratio $\frac{\beta}{\gamma}$ denoted as R_0 to study the dynamics of a developing epidemic.

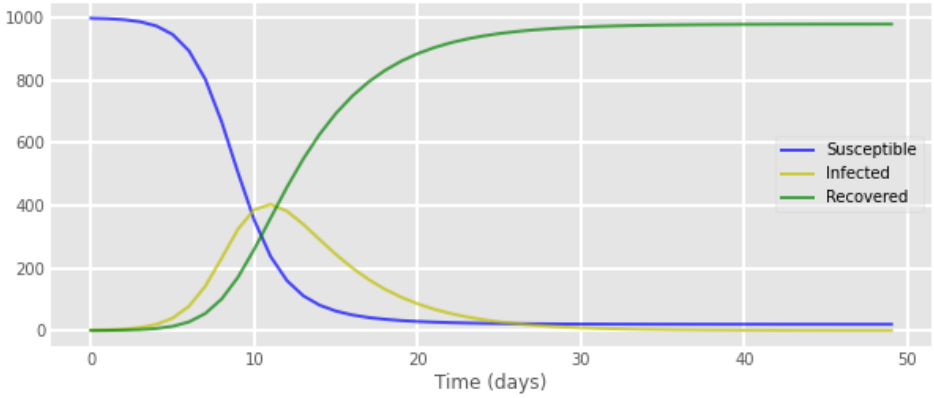


Figure 10.2: Basic SIR model simulation with starting values of S:999, I:1, R:0, β:1.0 and γ:1/4.

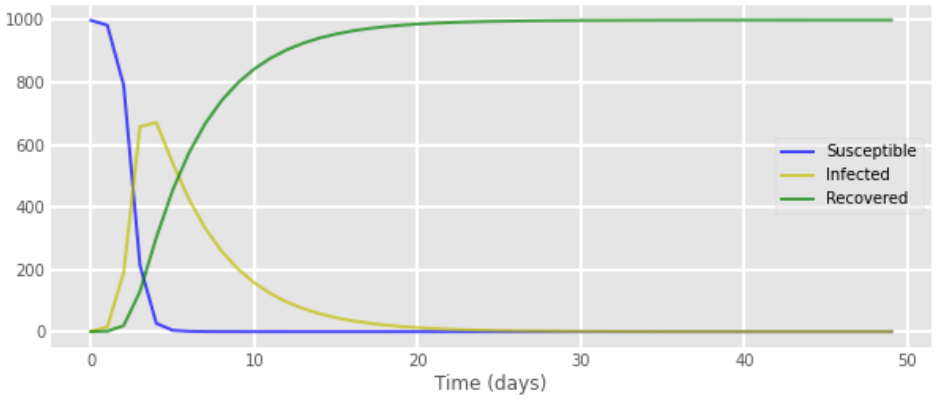


Figure 10.3: Basic SIR model simulation with starting values of S:999, I:1, R:0, β:3.0 and γ:1/4.

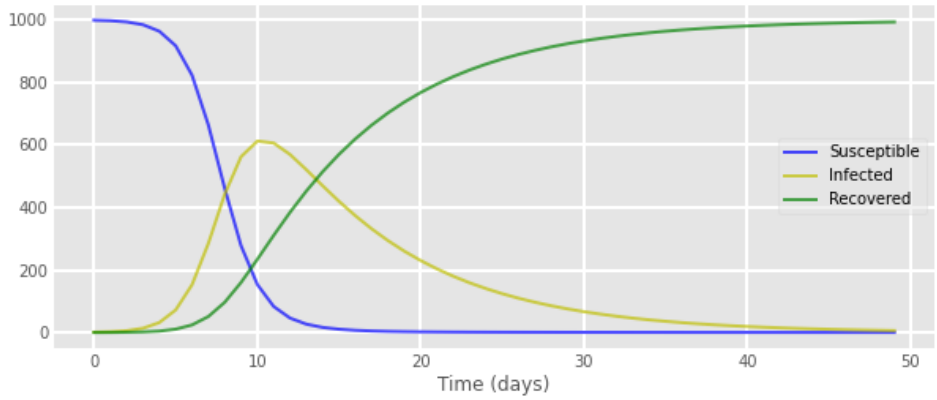


Figure 10.4: Basic SIR model simulation with starting values of S:999, I:1, R:0, β:1.0 and γ:1/8.

10.2 Extending the SIR model

The next step was to extend the basic SIR model with 2 new compartments (Figure 10.5).

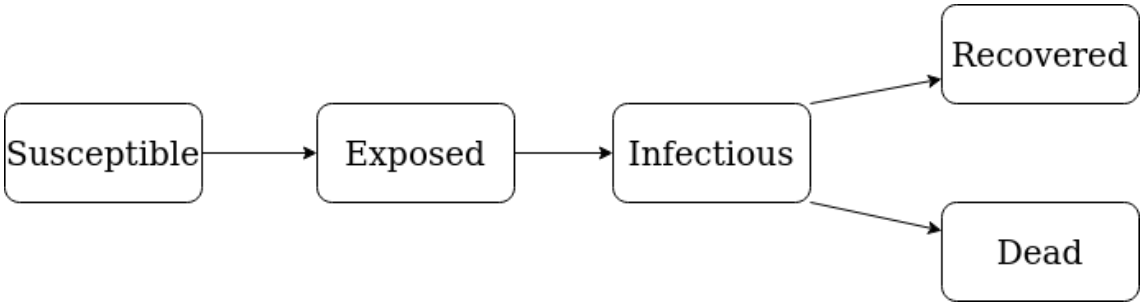


Figure 10.5: Flow diagram of the extended SIR-model. Two new compartments are added: Exposed and Dead.

Between susceptible and infectious people the exposed state is introduced. Exposed individuals carry the virus with an incubation period factor δ but are not infectious. Furthermore, the infectious group now branches out into recovered and dead. Thus, a death rate factor α is introduced to simulate the chance of death for infectious people, while also introducing a factor ρ for the length of time until death. It follows that the differential equations had to be altered:

$$dS/dt = -\beta * S * \frac{I}{N}$$
$$dE/dt = \beta * S * \frac{I}{N} - \delta * E$$
$$dI/dt = \delta * E - (1-\alpha) * \gamma * I - \alpha * \rho * I$$
$$dR/dt = (1-\alpha) * \gamma * I$$
$$dD/dt = \alpha * \rho * I$$

At next, the influence of different population proportions on the fatality rate factor α are introduced to the model. Since the age of infected people has an impact on the severity of the disease and the death rate [45], we created four age groups: 0-29, 30-59, 60-89 and 89+. Based on the death rate calculations by age groups in Italy [9], we assigned differing α values to each age group and added the percentages of the age group distribution in Germany (Table 10.1). The resulting α value for the entire population was 0.07726. Another factor that has a considerable effect on COVID-19 outcomes is the smoking behavior. We used an RKI report

about the prevalence of smoking in the adult population of Germany from 2013 [21] to integrate the proportion of daily smokers for each age group (Table 10.2). Abrams et al. [1] analyzed that current or former smokers have an increased COVID-19-related mortality by 2.4 [95% CI 1.43–4.04]. Therefore, we multiplied the α values of the smoking people in each age group by this value. This effected the overall α value to be increased to 0.16389.

Age group	% in Germany	α
0-29	30.1	0.001
30-59	41.51	0.013
60-89	28.13	0.2267
89+	0.27	0.285

Table 10.1: Alpha values assigned to different age groups.

Age group	smokers	non-smokers
0-29	31.95	68.05
30-59	26.375	73.625
60-89	8.45	91.55
89+	-	100.0

Table 10.2: Proportion of smoking in the adult population of Germany.

The third and last population proportion, we added to our model was the gender information. Zhang et al. analyzed potential risk factors in a study of n=663 Covid-19 patients and identified that male patients have an odds ratio of 0.486 [95% CI 0.311–0.758] to unimprove from the disease. We integrated this information to our model combined with the proportion of males and females in Germany from 2018 [5],[4]. Since there are 50,7% females and 49.3% males, α was slightly reduced to 0.16383. The final simulation model can be seen in Figure 10.6.

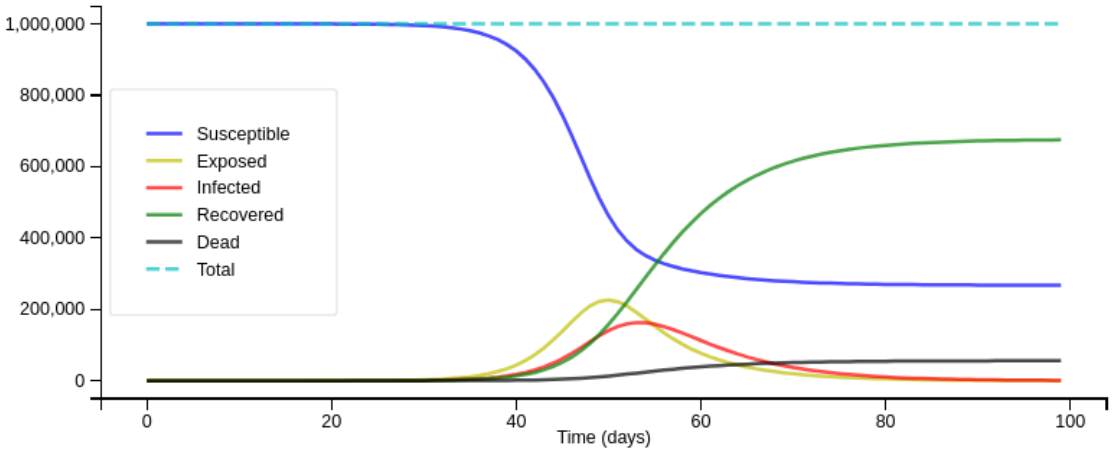


Figure 10.6: Resulting SIR model simulation with adjusted α value.

The get the information how many ICU beds are in use at each time point of the simulation, we created an equation based on two information: 17% of patients infected with Covid-19 need hospitalization in Germany [10] and 48% of the hospitalized patients need ventilation [32] and

therefore an ICU bed. The resulting equation is $occupied\,ICU\,beds = I * 0.17 * 0.48$. When the capacity of ICU beds is exceeded the death rate in our model is changed to 0.6 (Figure 10.7).

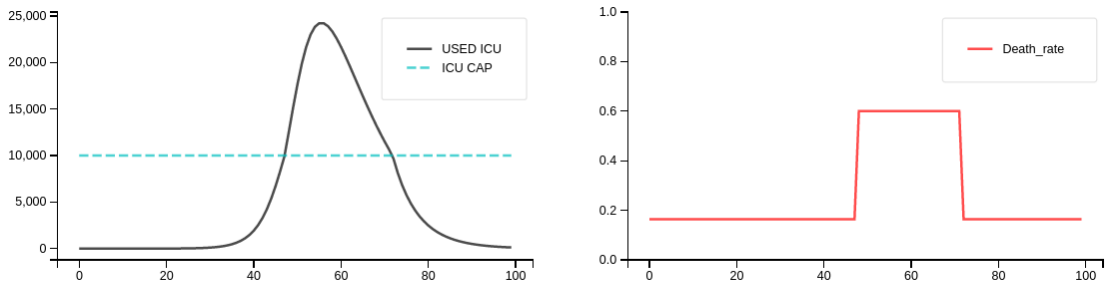


Figure 10.7: The left plot shows the occupied ICU beds (black) and the total amount of ICU beds (blue, dashed) while the right plot shows the corresponding death rate (red).

parameter	**description**	**value**
α	fatality rate	0.16
β	expected amount of people an infected person infects per day	1.25
γ	proportion of infected recovering per day	1/10 [32]
δ	incubation period (1/days)	1/5 [32]
$inf\_to\_dead\_d$	days from infection until death	50 [32]

Table 10.3: Initial parameter setup.

10.3 Parameter fitting

The next part deals with fitting the extended SIR model with time-dependent R_0 values and resource-dependent death rates to real Corona virus data of Germany, in order to come as close as possible to the real numbers and make informed predictions about possible future developments. The data we used is the data which contains the information about age groups and other parameters like fatality rates, R0 values, beginning of lockdown, etc in Germany and the data was parsed according to the range of events. The cases from 01.03.2020 – 30.04.2020 were only considered for fitting to our model to get the course idea.

Here, we initially loaded the data for the age groups, probabilities,created some look up dictionaries for easy access of the data parameters. We mainly focused on the the probabilities of infected to death. The equations of the model are translated to the coding with R_0-function and the whole model that takes the parameters to fit to calculate the curves of S, E, I, R, and D. Finally, for curve fitting we first set the parameters we knew and assumed with the upper and the lower bound values with unknown data, and defining the x values for the number of days to get the future predictions with the parameters fit.

The resulting simulations showed quite similar values with the real data with R_0 as 2.08 at the start of march 2020 and R_0 as 0.48 at the end of april, a death rate of 0.16. It is quite comparable with the real data and the many points are inline with the real data points indicating the best fit which is shown in the figure 13.6. So with the outbreak beginning on 21st January and the outbreak shift set to 30 days, so our model thinks that the main lock down took place in Germany nearly after 107 days i.e roughly in the middle of March which is very close to the real data. Figure 10.7 represents the prediction model, Here's the prediction in April — if the model is right, Germany has gone through the worst already as deaths per day is increasing which should decrease strongly over the

next months. R_0 will stay in the range, if it goes up again as lock downs are reverted, the numbers will start increasing again.

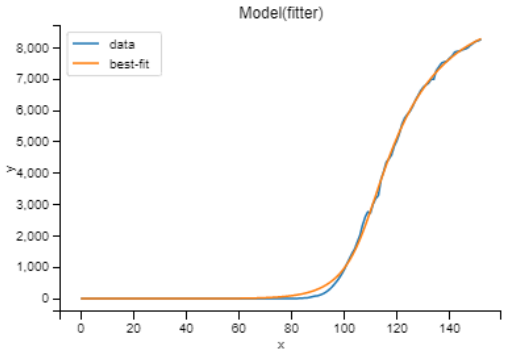


Figure 10.8: Model fit for Germany

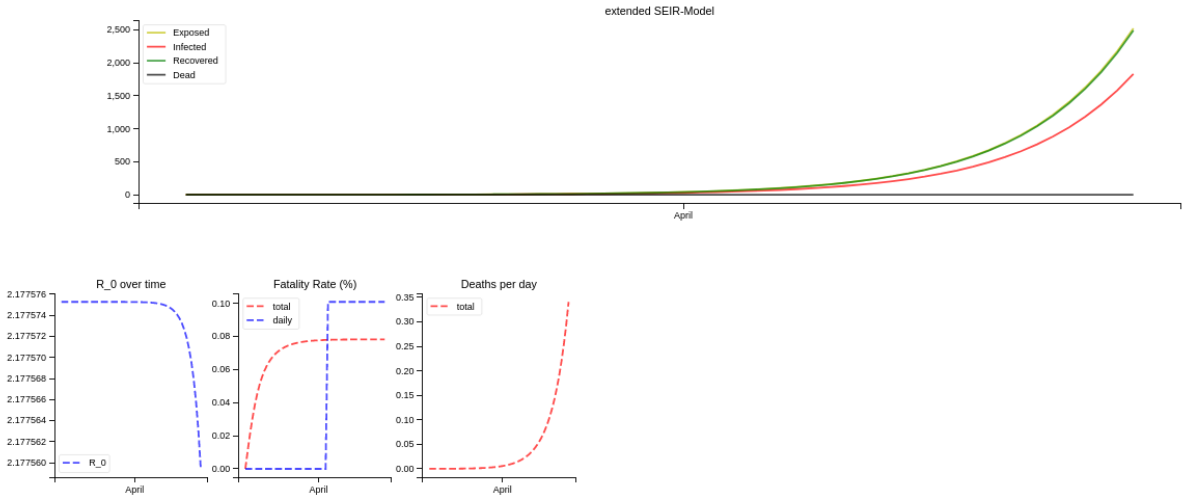


Figure 10.9: Prediction for Germany with suitable parameters

10.4 Scenario Studies

In the last step, the fitted model is used to simulate the spread of the virus with various prevention methods being implemented. Those methods affect the spread of the virus by reducing the expected amount of people an infected person infects per day (β). Recall in the equations in section 10.2, where the parameter β is obviously only present in the equations for the number of susceptible and the number of exposed people. The following prevention have been implemented to reduce β:

Reducing Social Contacts

Maybe the most effective way of decreasing the spread of the virus is to limit social contacting. A short and lose restriction period increases the risk of an uncontrolled spread of the virus or the arise of a second wave which will lead to many deaths. But a too long and tight restriction can lead to economic and psychological incisions. Performing simulations can help to see the effect of social distancing with varying periods and intensities. For our simulation we implemented social distancing by reducing β by 0/25/50/75%.

Lockdown

The so-called lockdown is a special case of social reduction where social contact is reduced to an absolute minimum by restricting the population leaving their house. But even for the case of a lockdown, peoples' social contacts cannot be stopped completely because a portion of the population like cashiers or hospital staff needs to go to work. Therefore we modelled the lockdown as a reduction of β by 90%.

Masks

Several studies [36][39] state that wearing masks can effectively reduce the spread of the virus by 8%-16%. Each of the social distancing simulations have been performed with and without the usage of masks. Wearing a mask is modelled as an additional reduction of β by 12%.

10.4.1 Methods

After a fixed unrestricted time of 30 days the restriction period starts, followed by 100 simulation days, where the length of the restriction within the simulation days is varied by 0/25/50//100% of the simulation days. As before we initialized our model with one individual being infected while the rest of the population is susceptible.

10.4.2 Results

From Figure 10.10 we can make several observations:

- With no restrictions (plot [1,1]) implemented the amount of infected people is with is by far the highest and the number of dead people is breaching linear growth
- Even minor restrictions for the smallest period reduce the number of infected people noticeably
- Wearing masks has absolute as well as relatively a higher impact when combined with less strict social restrictions
- Increasing the time of social contact reduction has less impact than the the intensity of social distancing
- The difference between 50% and 100% restricted simulated days is just minor when combined with masks and lockdown

The parameters that has not been fitted to German data have been set as shown in Table 10.3.

10.4.3 Discussion

The performed simulations suggests that both the duration as well as the intensity of the restrictions plays an important role when fighting the outbreak of corona. The usage of masks is even more important for minor social reduction scenarios. Nevertheless, the simulations seem to underestimate

the true case. The number of infected people might be underestimated. Recall, the model was fitted to Germany with the data from beginning of March to the end of April. At this point the government of Germany already introduced several restrictions to keep the spread of the virus under control, such like advising people to stay home, closing public locations and shifting many jobs to the home office. Thus, our data is already fitted to restrictions and then used to simulate restrictions, which doubles the effect of the implemented restrictions in the different scenarios. Surprisingly, without any simulated restrictions and with a model that has been fitted to data of period where restrictions were present in Germany, the model still simulates more than 10 million infected people within 130 days.

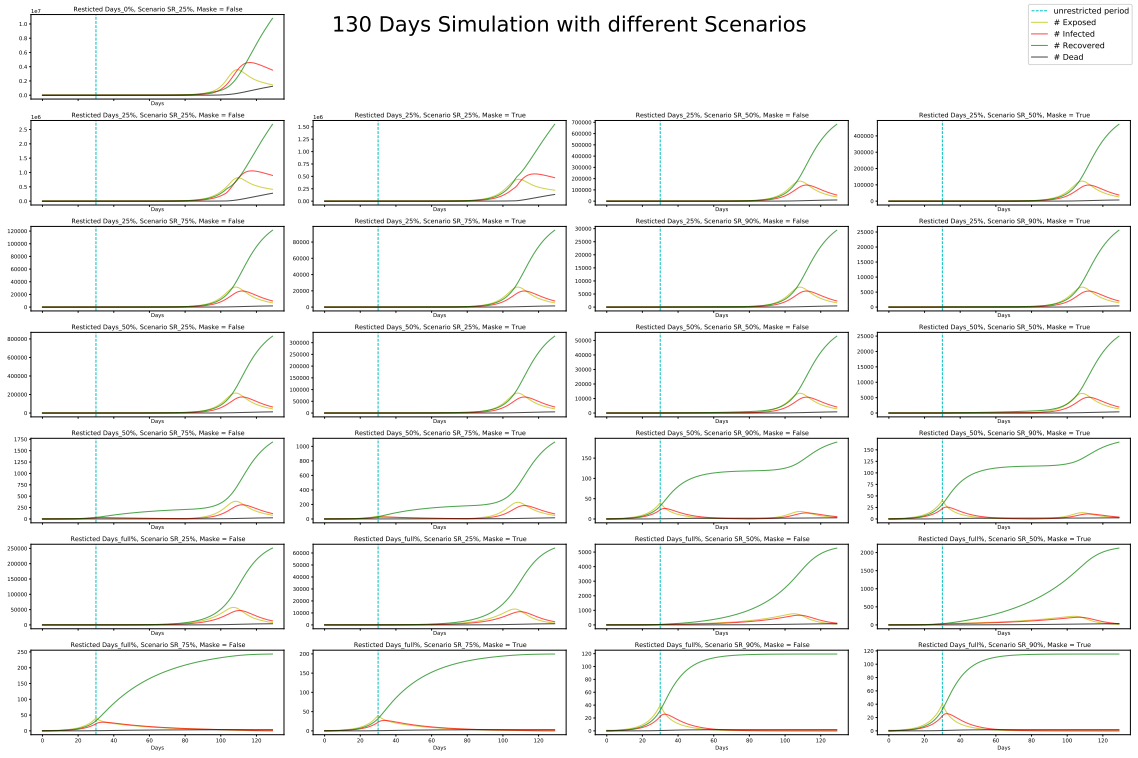


Figure 10.10: Each plot represents an independent simulation, starting with no restrictions (top left). The duration and intensity of the restrictions is varied through the simulations and once combined with the usage of a mask. The time without restrictions is denoted by the blue vertical line. Note that the number of susceptible people is not shown and the number of people in general (y-axis) is not normalized to improve visibility.

IV

Part 4

10.5 Introduction

10.5.1 Background

Agent Based Modeling (ABM) has gained significance in the last 30 years due to ever increasing computational efficiency. It has a wide variety of applications including but not limited to biology, businesses, technology, socials sciences and economics. The idea of ABM is based on simulating independent agents operating and interacting with each other within a micro scale computational model confined to a predetermined rule set. Especially within the field of epidemiology ABM's are characterized by their ability to capture heterogeneity of complex interactions between different agents [3]. The complexity of different agent behaviour can always be mirrored within the simulation by including new constraints or rules on the model. Thus, ABM's are very effective in modelling different epidemiological outcomes with different scenario rule sets.

10.5.2 Goal of the Project

The aim of this project is to use agent-based simulation to model the interactions of individuals within a population during the Covid-19 outbreak, so that one can determine how small changes in behavior and interaction can affect population level output. Different extensions (incubation and exposed state; chronic conditions and comorbidities; central locations) are implemented to refine the model. In the end, the variability of human behaviour can be shown with the purpose to understand the variability in the likely effectiveness of proposed interventions.

10.5.3 Outcome

The integration of central location had the largest impact of the added model extension. In the basic scenario without movement restrictions 90% of the population becomes infected by the virus after 21 days of simulation when supermarkets and schools are both opened. The ICU capacity was exceeded after 28 days. By applying different intervention strategies, the combination of social distancing and wearing masks has been confirmed to be the most effective.

10.6 Introduction to Agent Based Modeling for Covid 19 spreading simulations

Agent Based Models (ABM) can be implemented in very different fashions. For this week's project, we used the so-called simple billiard balls model (Silva) which is composed of a population of agents, within a loop where the agents run and interact. The agents are initialized with properties such as working place, age, or health conditions that drive their mobility patterns. As the name suggests the agents are represented as billiard balls that can transmit the virus when they get in touch with each other. The big advantage of this approach is its simplicity and modularity. On the other hand, the billiard balls model is very abstract and most likely produces wrong results. Nevertheless, all models are wrong, but some are useful (George P. Box) [7].
Taking this approach a step further the *Spatiotemporal Epidemic Model* introduced in April 2020 by Lorch [25] makes spatiotemporal predictions by making use of data from contact tracing technologies. By using Bayesian optimization the model estimates the risk of exposure based on the moving habits (e.g. going to a certain bar) of each individual, the percentage of symptomatic individuals, and the difference in transmission rate between asymptomatic and symptomatic individuals from historical longitudinal data.
Instead of modeling people as billiard balls, one can model a population as a network. The so-called *Network Based Model* (NBM) takes several assumptions into account, like social interactions, the probability to spread the disease, relationships between the individuals, immunity after infection, and many more. Based on those, a graph is built where each agent is represented as a node and the relationships between agents as an edge. The assumed baseline network structure is an input of the model but health policies, e.g. lockdowns, quarantine, etc., can be interpreted as (temporarily) changing the social network by eliminating edges.

10.7 A simple ABM

The ABM should have the agents (i.e. people) with the same characteristics of a national population. For this purpose, the age group distribution of the agents (i.e. people with the same characteristics of a country's population) should be close to that of the USA with age groups : 0-14 years: 18.62%, 15-24 years: 13.12%, 25-54 years: 39.29%, 55-64 years: 12.94%, 65 years and over : 16.03%), which were determined in 2018. This was achieved by using the beta probability distribution with parameters α= 2 and β= 5, so that the age ~β(2,5).

```
age = int(np.random.beta(2, 5, 1) * 100
```

The ABM is designed so that each agent must be in one of these states: susceptible, infected and immune-recovering. There are adjustable initial percentages of infected (0.02) and immune people (0.01) given by simulation, and the rest of the population has the status "susceptible". There is also the death status, which was created for the group of people who have the severe symptoms of SARS-COVID-2 and have not yet become immune. Spread through contagion is determined by the interaction of the infected agents through proximity or contact. This means that the faster the agent moves, the greater the probability that he will approach an infected agent and become infected as well. A Contagion Distance defines the minimum distance (set at 5) that two agents must have for virus transmission to take place. The terrain where the agents are simulated is squared and bi-dimensional (100x100). Each agent is randomly created within this terrain,like horizontal and vertical position of the agent (in code:

```
self.x = kwargs.get('x', 0), self.y = kwargs.get('y', 0)
```

).During simulation, the mobility amplitudes can be defined for any possible agent status, in each iteration, each of the agents also moves randomly within the environment. Only the steps of its position are defined by

```
x,y = np.random.normal(0, self.environment.amplitudes[self.status], 2)
self.x = int(self.x + x)
self.y = int(self.y + y)
```

The distance between two agents a1 and a2 is determined by

```
np.sqrt(((ai.x + self.positions[m][0])-(aj.x + self.positions[n][0]))
** 2 +((ai.y + self.positions[m][1])- (aj.y + self.positions[n][1]))
** 2)
```

For **all** dead agents and **all** agents with the status infected and with their severity of hospitalization or severe infection, their movement will be set to zero.
Furthermore, the effects of mobility restrictions on the economy - especially on the income and wealth of individual agents - are simulated. The agents' income is simulated as a function of their mobility. Then the mobility of the agent is defined by the Euclidean distance from his previous position. The wealth of the agents is initialized according to the equal distribution of the society. This distribution is measured using quintiles, each quintile represents a social class: 20% most poor, poor class, working class, rich class and 20% richest. The minimum income defined by the first quintile of the poorest is used as the unit of expenditure and income of each stratum. During iteration, the wealth of each agent is reduced by its minimum fixed expenses, where the constant value in units of minimum income is proportional to its actual wealth, such as expenses = minimum_income [wealth quintile]. Furthermore, in each iteration, the wealth is increased by the daily income of the agent. The income is a random value which is proportional to his actual wealth and his mobility. Then, the final income is replaced by the minimum income [wealth quintile].

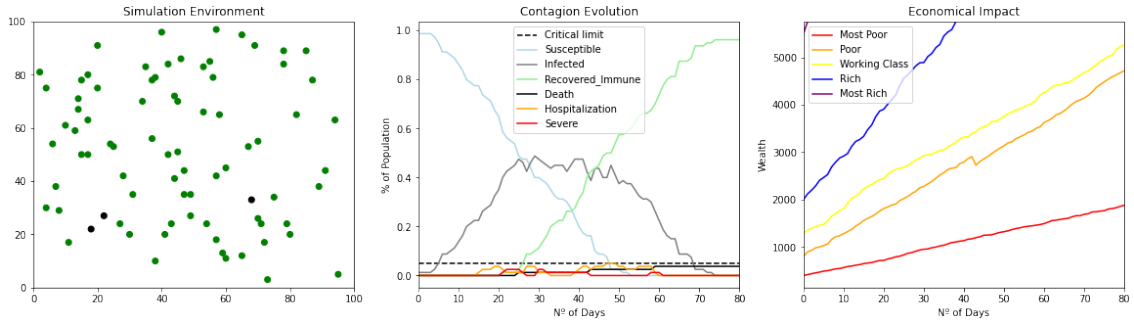


Figure 10.11: simulation with covid19 abs 1 run

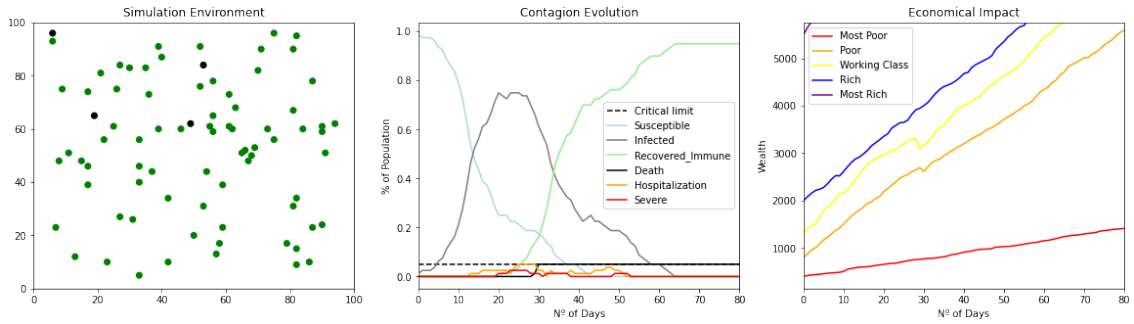


Figure 10.12: simulation with covid19 abs 2 run

Another scenario was build by setting the percentage of initially infected persons to 2% and of recovered/immune people to 1%. To simulate a lockdown the mobility amplitudes of susceptible,

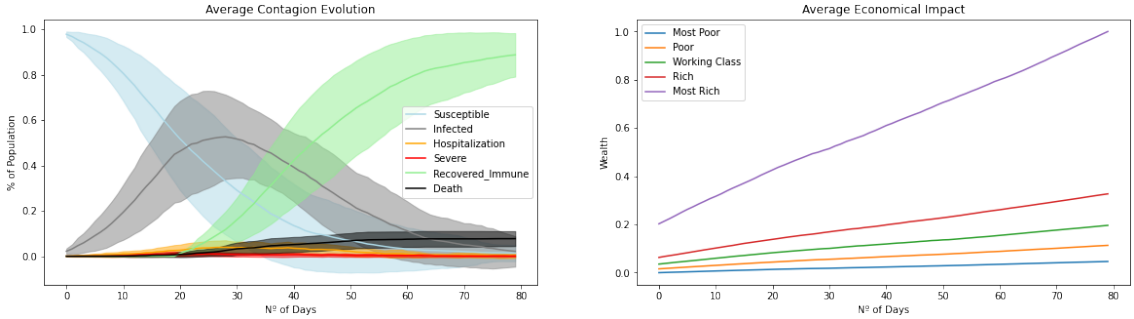


Figure 10.13: simulation with covid19 abs. Average results for 50 executions

recovered immune were set to 0.5 and to 0 for the infected one when 5% of the population is infected. To adjust the time frames from 1.3.20-30.4.20, iterations number was set to 60. Note that ABM approaches belong to the class of Monte Carlo algorithms whose results are generated using randomness and statistics. Therefore a bunch of runs is computed and the distribution within the results is evaluated. In order to obtain a reliable results for this weeks project, the simulation was executed 50 times simulations and the confidence intervals are displayed in Figure 10.14. You can see that the infected number goes down by reaching the condition of 5% infected, which is reduced to about 0% after about 30 days, which is also the start of lockdown. The number of susceptible gradually decreases over the whole period. After about 30 days, the recovering immune increases abruptly and remains at the same level. The plot, which represents economic conditions, shows the decreasing trend, which also happened in reality.

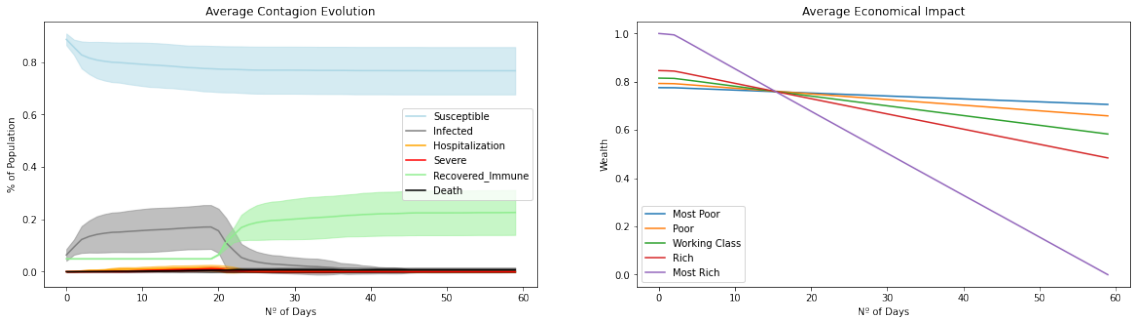


Figure 10.14: Simulation with Covid-19 ABS. Average results for 50 executions fitting to real data

Here is the plot for simple ABM 10.11. In the plot for the evolution of the contagion risk, you can see how much the critical limit of the health care system has been implemented and how many lives have been lost. This is the simulation of a catastrophic situation that will occur if nothing is done. The plot of the economic impact shows that it is not so bad, because the economy does not stop growing. If you compare the two diagrams for 2 runs 10.11 and 10.12, you can see that the curves for the contagion status "susceptible", "infected" and "immune recovered" already differ significantly. When comparing the plot for 50 executions 10.13, it is clear that curves from two plots for one run each are in the confidentiality area in the plot for 50 executions. In order to have reliable results, plot with confidentiality area should be used.

10.7.1 Fitting to real data

Unfortunately we could not fit the model to real world data. Since running the ABM is already a runtime expensive process we decided to set some parameters according to research papers in which the authors have already reported statistics for the current Corona outbreak. Values like

infection risk when being exposed, incubation time, infection duration could be set according to the data in WHOs Health Report [28]. Other values like the portion of immune individuals by the beginning of the outbreak are not detectable. Consequently we ran our simulations for those parameters with varying initial settings.

10.8 Extending the Model

One of the biggest advantages of the presented ABM approaches is its modularity. By adding properties to each agent and relationships between agents an entire society can be modelled. The simple ABM is a good starting point to understand the mechanisms of the model, but simplifies too much. The following extensions have been added to make the predictions more realistic:

10.8.1 Incubation and Exposed State

Two states were introduced to capture the characteristics of a virus spread. On the one hand the *Exposed* state is used to count all individuals that had contact with an infected person but did not infect themselves. The *Exposed* state can be used to measure how infectious the disease is by checking how many of the people that had contact with an infected person got infected by themselves. A low ratio would indicate that the disease spreads only under tight conditions. Note, that the new state is interesting for our modeling but hardly applicable to real-world scenarios since it would require a bunch of test kits, and the contact chains of infected people needed to be tracked down. Both requirements are not given at this point in any country.
On the other hand, we added another Infection state to make our model predictions more realistic by implementing an *Incubation* state. People that get infected will go through an *Incubation* period, in which they cannot infect other people. Concerning the 73th WHOs' health report [28] for the current Covid-19 virus most infected people went through an incubation period of 5-6 days. This implementation mostly delays the outbreak by preventing the spread of newly infected people.

10.8.2 Chronic Conditions and Comorbidities

A second expansion to the model was performed by taking chronic health conditions of infected individuals and their effect on the fatality rate into account. Three common german health conditions were identified: obesity, hypertension and diabetes. For each risk factor the prevalence rates within the german population were taken from the literature (54% [27] for obesity, 33% [16] for hypertension and 13% [12] for diabetes). Next, each agent is initialized by random chance with none, one or multiple conditions representing the true prevalence rates of the population. To accurately simulate the comorbidities for people dying of covid-19 we used a report by Solis et al. [35]. Individuals under the age of 18 were assumed healthy.

comorbidities	**death rate**
0	5.58%
1	13.5%
2	23.2%
3	32.9%

Table 10.4: Death rates and comorbidities, the numbers were adjusted based on age, due to inflated death values.

By simulating the extension and comparing it to the base model a visual bump in the death rate can be seen (Figure 10.15). The dead agents are more then quadrupled, which suggests, that our values for the death rate are inflated. While we account for healthy individuals as well as people under the age of 18, the over-inflation might be due to the fact, that the report measured only hospitalized people with stays over 14 days. The value ranges are to specific for one subgroup of people. Thus the base model, using death rates per age, is much more true in its simulation.

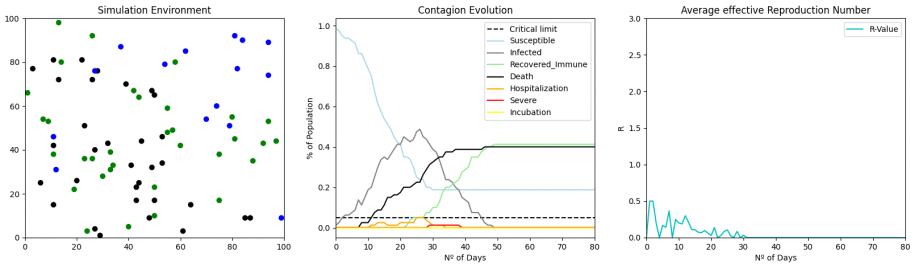


Figure 10.15: simulation of chronic conditions effect on the base simluation

10.8.3 Central locations

With the additions of central locations to the model, the agents are directed to predefined places at specific time points, instead of performing only arbitrary movement within the environment.

Supermarkets

Since all people have to provide themselves or their families with groceries, we decided to include supermarkets. Each agent with an age of 18+ (assuming that younger agents are supplied with food by older family members) has to go to the supermarket at least once every seven days. Based on the average retail sales area per 1,000 inhabitants in Germany [40] the size of the supermarket is adjusted in our model. For the number of 2438 inhabitants describes later, a retail area of 3600 m^2 was created, divided into three different locations (2000 m^2, 1000 m^2 and 600 m^2). To monitor the time (in days) an agent did not visit the supermarket, an attribute was added the the *agents* package. It is a simple counter, which is reset if the agent was either placed in the supermarket after six days of absence or when the agent enters the supermarket area by random movement. The counter is initialized by random values between 1 and 6 so that all agents have to visit the supermarket at different times. If an agent is placed in one of the supermarket, his position inside that area is calculated randomly, so that he is in contagion distance to some but not all others visitor of the market. The probability that the agent goes to the biggest supermarket was set to 50% while he is placed with 30% and 20% to the medium and small supermarket respectively. After a shopping day, the next position of the agent is calculated randomly within the entire environment.

School

Since only 18+ agents visit the supermarket, younger agents still perform only random movement in our model. Therefore, we created a school. Every agent with an age between 6 and 17 attend the school five of seven days a week. Here, in contrast to the supermarket all kids visit the school at the same days. We decided to place the school in an outer area of the environment, because it is less possible that people unintended come by a school (compared to the supermarket). The area of the school is 2000 m^2 and after five consecutive days in school, the new position is again calculated randomly. For the placement of the agents within the school the same concept as for the supermarket is applied, simulating interactions on the schoolyard only to some but differing school mates. The presence of adults (i.e. teachers) is not yet implemented.

10.9 Scenario Studies

The new extensions were tested by simulating different scenarios, where each scenario represents restrictions that have been implemented to decrease the spread of the virus. The following prevention have been implemented. For the reasons outlined in section 4.4.2 we could not include the chronic expansion:

Reducing Social Contacts

Maybe the most effective way of decreasing the spread of the virus is to limit social contacting. A short and lose restriction period increases the risk of an uncontrolled spread of the virus or the arise of a second wave which will lead to many deaths. But a too long and tight restriction can lead to economic and psychological incisions. Performing simulations can help to see the effect of social distancing with varying periods and intensities. For our simulation we implemented social distancing by reducing the movement of the individuals by 60% after 10% of the population is infected. This change is reverted when only 5% of the population is still susceptible.

Lockdown

The so-called lockdown is a special case of social reduction where social contact is reduced to an absolute minimum by restricting the population leaving their house. But even for the case of a lock down, peoples' social contacts cannot be stopped completely because a proportion of the population like cashiers or hospital staff needs to go to work. Therefore we modelled the lockdown as a reduction by reducing the travel amplitudes of the agents by 90%. The travelling reduction is revoked when a individual needs to go to a supermarket.

Masks

Several studies [36][39] state that wearing masks can effectively reduce the spread of the virus by $8\% - 16\%$. Each of the social distancing simulations have been performed with and without the usage of masks. Wearing a mask is modelled by reducing the initial *contagion_rate* by 16%.

Each simulation was executed with some fixed parameters that did not change for the different scenario simulations (Table 10.5). The simulation environment was initialized to represent 1 km^2 by setting width and height to 1000. The population size of 2438 was determined to reflect the population density of a German city. We chose Hamburg's population density [14] and adjusted our population such that each point in our graph still represents 1 m^2 while being displayed as 0.5 m^2 for a better visualization. The maximal distance between agents for contagion was defined to be 5 meters. This value does not correspond with the reality, but since each agent spends the entire day at the same position, the model would not provide meaningful results with a maximal contagion distance of 1.5 or 2 meters, which corresponds with the reality [33]. Each scenario was performed for a span of 80 days.

parameter	description	value
w	Width of the environment	1000
h	Height of the environment	1000
pop_size	Population Size	2348
crit_limit	Maximum percentage of population which the Healthcare System can handle simultaneously	0.05
dist	maximal distance between agents for contagion	5
δ	Percentage of infected in initial population	0.02
β	Percentage of immune in initial population	0.01
M	Mobility ranges for agents by the beginning of simulation	5
i_risk	Prob of being exposed when being in contact with infected agent	0.9

Table 10.5: Initial parameter setup.

10.9.1 Results

The basic scenario with no restrictions as same as the implemented intervention scenarios were analyzed with and without the presence of central locations to estimate the risk of visiting supermarkets and opening schools. For all resulting figures, the left plot shows the agents represented as billiard balls while their color shows the agent's state after 80 simulation days. The centered plot displays the course of the health state portions for the entire population on each day. The right plot indicates the effective reproductive Number (R_t). It is the average number of secondary cases per infectious case in a population made up of both susceptible and non-susceptible hosts. If R_t exceeds 1.0, the virus will spread exponentially.

At first, we will compare the different model implementations for the basic scenario where no movement restrictions are applied. In Figure 10.16 it can be seen that when both schools and supermarkets are opened more than 90% of the population becomes infected by the virus after 21 days of simulation. The threshold for available ICUs is reached after 28 days, which cause an immense increase in the death rate. After 45 days, the entire population was infected with Covid: 95% recovered while 5% died. When only supermarkets are opened and schools are closed (Figure 10.17) the speed of the spread is slightly reduced. Nevertheless, the ICU capacity is exceeded after 29 days with the effect that more than 6% of the population died. At the end of the simulation, 20% of the people remain susceptible. Analyzing the model where supermarkets are excluded, but the school is opened, the infection curve is visibly flattened. The ICU capacity is never exceeded since most of the infected people are kids (6-17 years), which have a lower probability of a severe course of the disease. Due to the fact that only young people are visiting the school nearly 50% of the entire population stays susceptible until the end of the simulation.

By examining the plots for the lockdown (Figure 10.19, 10.20, 10.21) and the contact reduction scenario (Figure 10.22, 10.23, 10.24) we unfortunately noticed that our code contains an implementation error. After visiting the supermarket, the next position of the agents is calculated randomly. Instead, we should store the position from where the agents were moved to the supermarket and reassign that to that location on the next day. The random position calculation after the supermarket shopping conflicts with the concept of lockdown and social distancing. Although the movement of the agents is reduced, most of the population becomes infected after 30 days of simulation and the results do not differ as much as they should compared to the basic scenario.

However, wearing masks (combined with social distancing) has an remarkable impact on the spread of the virus. In this approach the contagion rate is reduced by 16% which lowers the impact of contacts in central locations. In the model where only schools are opened (Figure 10.27), the young population (25-30%) becomes infected after 12 days (5 days in school + weekend + 5 days in school) and recover very quickly. When the school is closed and only the supermarket is opened the infection curve is shifted to the right, because the people visit the supermarket at different days. Here, the peak of infected people is visible after 50 days with 30% of the population infected. When both institution are opened simultaneously 40% of the population is infected after 30 days, but the curve rises slowly enough that the ICU limit is never exceeded.

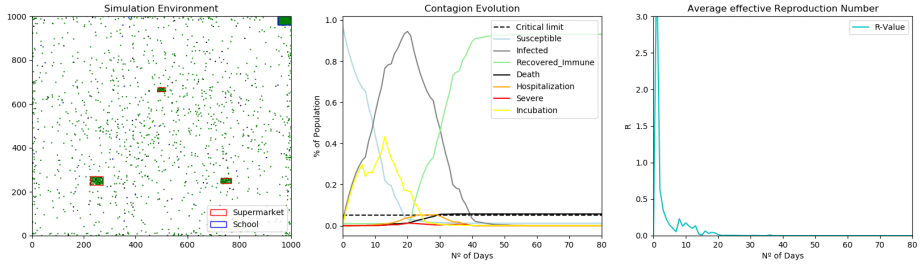


Figure 10.16: **Basic Scenario** with no restriction and the presence of three different sized **supermarkets** in the center and a **school** in an outer region of the environment.

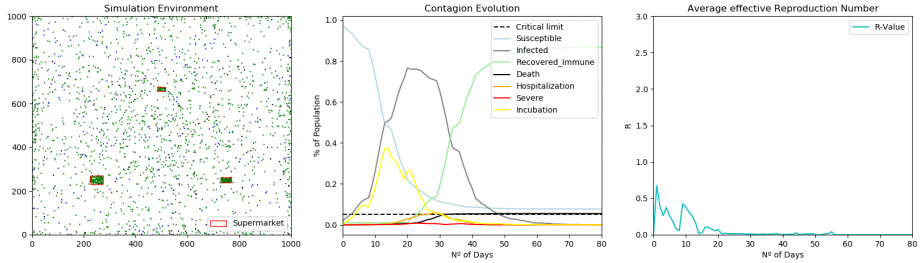


Figure 10.17: **Basic Scenario** with no restriction and the presence of three different sized **supermarkets** in the center of the environment.

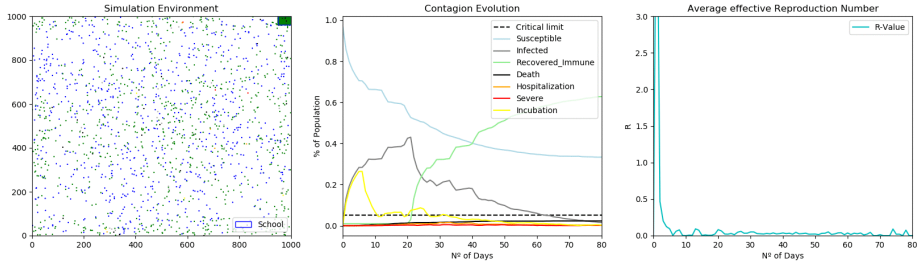


Figure 10.18: **Basic Scenario** with no restriction and the presence of a **school** in an outer region of the environment.

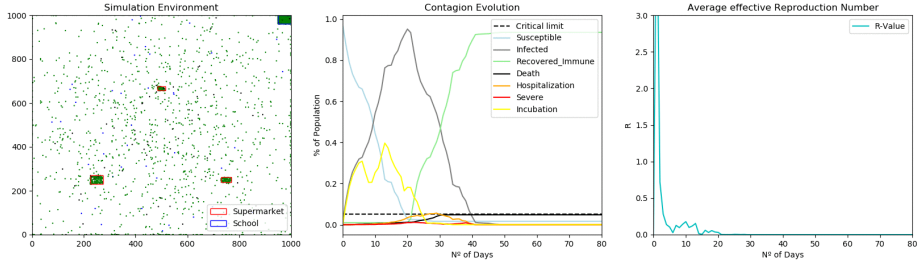


Figure 10.19: **Lockdown Scenario** with the presence of three different sized **supermarkets** in the center and a **school** in an outer region of the environment.

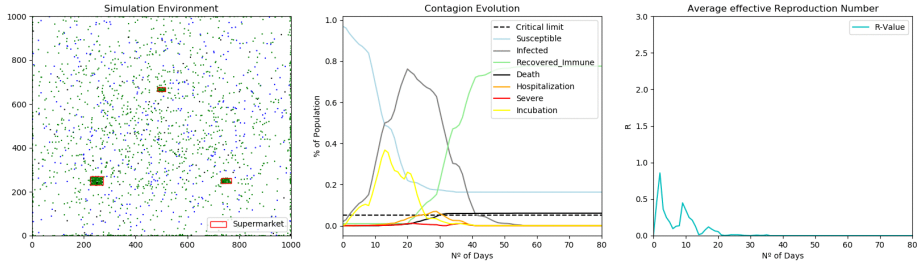


Figure 10.20: **Lockdown Scenario** with the presence of three different sized **supermarkets** in the center of the environment.

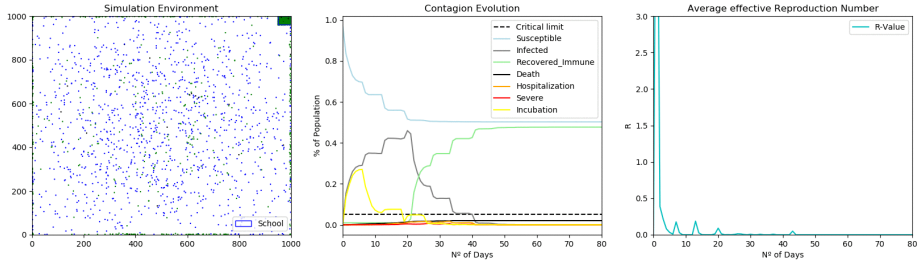


Figure 10.21: **Lockdown Scenario** with the presence of a **school** in an outer region of the environment.

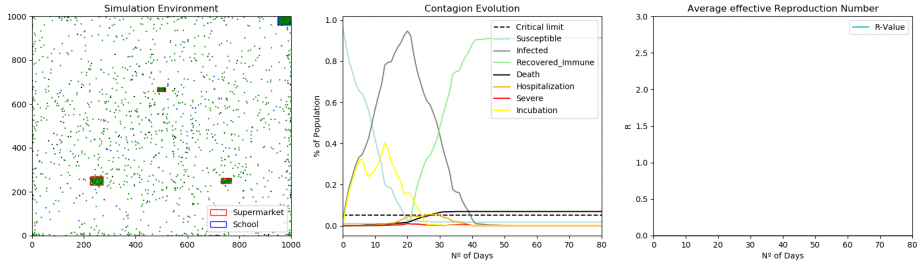


Figure 10.22: **Social contact reduction Scenario** with the presence of three different sized **supermarkets** in the center and a **school** in an outer region of the environment.

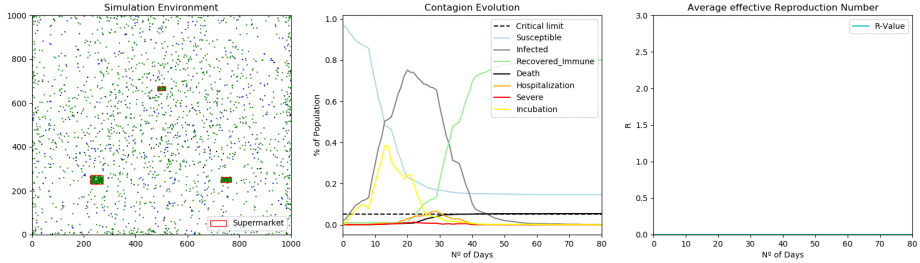


Figure 10.23: **Social contact reduction Scenario** with the presence of three different sized **supermarkets** in the center of the environment.

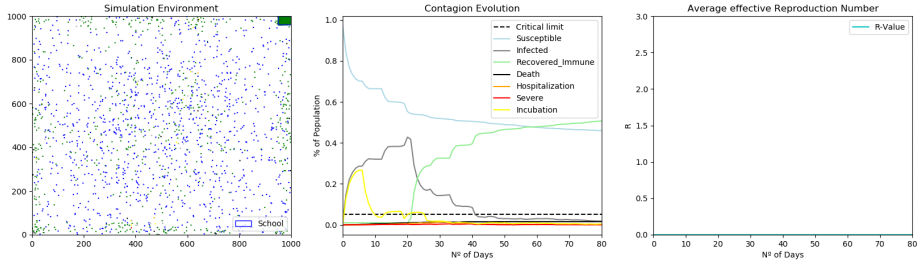


Figure 10.24: **Social contact reduction Scenario** with the presence of a **school** in an outer region of the environment.

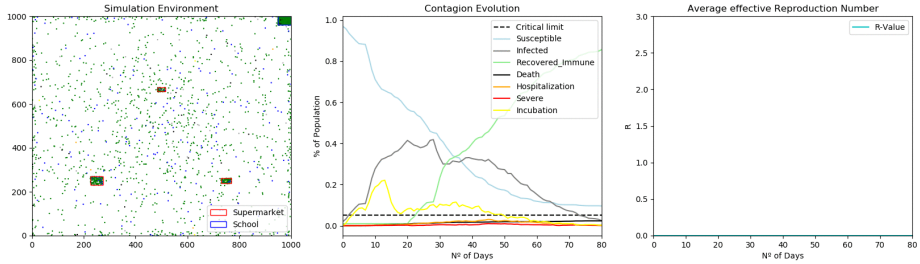


Figure 10.25: **Social contact reduction combined with wearing masks Scenario** with the presence of three different sized **supermarkets** in the center and a **school** in an outer region of the environment.

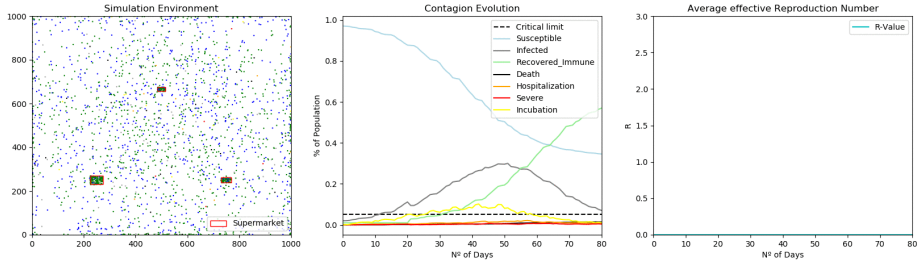


Figure 10.26: **Social contact reduction combined with wearing masks Scenario** with the presence of three different sized **supermarkets** in the center of the environment.

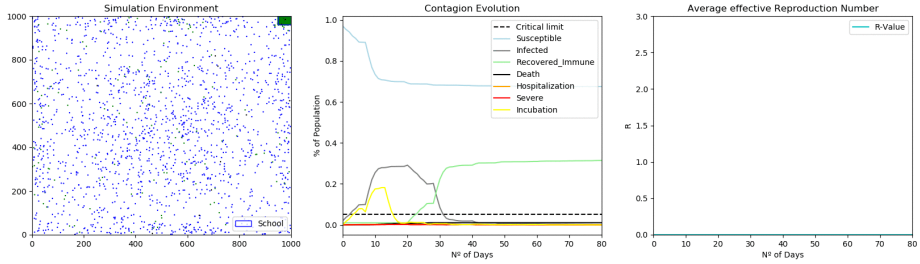


Figure 10.27: **Social contact reduction combined with wearing masks Scenario** with the presence of a **school** in an outer region of the environment.

10.9.2 Discussion

The results quickly demonstrate the need to integrate central institutions into the model, because of their large impact in agent-based systems. With different age groups (supermarket: 18+, school: 6-17) assigned to the visit of different central locations another effect becomes visible. If older people have to visit crowded places, the amount of available ICUs can be reached very fast, which need to be avoided at all costs. The analysis of the basic scenario clearly shows, that interventions are absolutely necessary. Unfortunately, the results of the lockdown and social distancing were difficult to analyze due to the implementation error. Nevertheless, the impact of wearing masks on the simulation results clearly demonstrates the effectiveness of intervention strategies to reduce the spread of the virus.

10.10 Comparison of EBM and ABM to simulate Covid-19 spreading

In the final task we draw comparisons between an ABM and EBM for the recent Covid19 outbreak. Both models are based on a common concept: modeling entities and observables within a complex system over a temporal timeline. While the basic concept is the same, their level of attention to the relationships between entities and the abstraction level of the system itself is different. Beginning with the modelling of entities, a typical approach with ABM would be going bottom up. Each agent itself is understood as an individual, with inherent properties and rule sets for interactions defined by observations. Using these definitions the model is build and simulated, showing us the macroscopic view of the individual interactions of the agents. In contrast, EBM could be defined as a top down approach. The system itself is modeled with complex equations, each entity not understood as an individual but as a compartment of subgroups.
These fundamental differences can already be observed in the computational power needed for simulation. ABM runtime grows exponentially with higher population numbers, due to the nature of individual agent modeling. EBM models are comfortable with big data sets as only simple equations need to be solved. The individual nature of ABM allows the observer to follow singular agent interactions, thus giving a unique microscopic insight into the epidemiological effects of the disease. This could lead to new insights into which factors are responsible for the spreading of the disease as well as the the infection rate. The EBM model does not allow this microscopic view and is very rigid in its simulation practices as no deviations from the set of equations are possible. On the other hand ABM can capture the inherent stochasticity of real world systems. For example agents might make decisions quite similar to real world individuals. This stochasticity can also be a negative influence, introducing noise into the system or leading to implausible outcomes. Alone with our simulations we had multiple outcomes where the epidemic did not start off at all or some subgroups exploding (See section 4.2.2). The effect of introducing a new rule set for interactions can have dramatic influences on the ABM system at a whole, thus the only way of fine tuning ABM was to subtly tune the parameters with ongoing runs. This is a very time expensive endeavour. Expending the model in population and rule sets might make it impossible.

Due to Covid19, there is a deep and continuous spread of infections between infectious and

Event-Based (Discrete) Modeling	Agent-Based Modeling
Macrospecifications reveal microstructures (top-down view)	Microspecifications generate macrostructure (bottom-up view)
Externally observable phenomenon (events)	Autonomous decision making entities (agents)
Programmed response to discrete events	Programmed functionality of agents
Events adhere to system-level observable information	Agents adhere to behavioral rules (boundedly rational)
System of interest changes state in response to events	Agents function independently and flexibly
Event impacts the entire entity	Agents interact as distinct parts of simulation
Simplicity in modeling inputs, state, and outputs	Simplicity in modeling rules
Internal behavior is unknown	Events emerge
Easy to test	Difficult to validate

Figure 10.28: Comparison of ABM and EBM characteristics. The figure is taken from [35].

susceptible people. If we show any negligence in the control measures, the outbreak will start to grow rapidly within no time. The infection is supposed to grow if the people infected by the infection is greater than one. However the only solution for this is people developing immunity. It is possible that the curve of infection starts up again after a continues period of low transmissions. This is called a second wave. This happens mainly due to the negligence of the people in not following the suitable control measures like social distancing, mobility etc.. By considering some suitable scenarios and applying the simulation with suitable parameters it is possible to simulate a second wave. The measure of immunity within a population changes the possibility of a second

wave happening. Both models take immune people into account. In ABM it is easy to create a change point within the system to model an upcoming second wave. By simply increasing certain parameters for infection rates, travel restrictions, social distancing and exposition rates a second wave can be simulated. In contrast EBM does not have an easy way to simulate a second wave. A change point has to be defined and the equations have to be introduced beforehand. Altogether ABM is more suitable for modelling a second wave then EBM.

V

Part 5

10.11 Introduction

10.11.1 Background

A time series is defined as a set of observations arranged in chronological order. The time interval between each data point remains constant, such that a sequence of discrete-time data is generated [19]. One aim is to extract meaningful statistics and interesting characteristics from the time series data structure. Thus, investigating the stationary and seasonality is part of the standard protocol when dealing with time series data. It is said to be stationary if it's mean and variance do not change over time, while seasonality can be identified depending on the data, which refers to periodic fluctuations of the values. The second major intention is to perform forecasting. Here, a model is applied to the data to predict future values based on the past observations.

10.11.2 Goal of the Project

The aim of this week's project is to forecast the COVID-19 outbreak using a classical approach, the Prophet library and a machine learning technique. The methods are applied to three different time series datasets: two datasets each generated by an SIR and an ABM model and another dataset containing the confirmed COVID-19 cases for Germany from the beginning of March until the end of April 2020. The results are evaluated by comparing the forecasting plots with the actual data as reference and analyzing the root mean square values. Additionally, a comparison between data-based and model-based prediction methods is performed by investigating the performance of the best SIR model of the previous week's project on the same data.

10.11.3 Outcome

AutoRegressive Integrated Moving Average (ARIMA) was chosen to represent the classical approach while the concept of Long Short Term Memory Neural Network (LSTM) was the machine learning technique applied to the data. LSTM turned out to be the clear winner and produced the lowest RMSE values for the prediction on the real life dataset (RMSE=0.007) and the SIR dataset (RMSE=0.0) while Prophet achieved the best prediction results on the ABM dataset (RMSE=0.0155). The comparison of data-based and model-based approaches (LSTM vs SIR model) also confirmed LSTM to be the better option for the given data.

10.12 Predicting time-series: model-vs. data-based

Data-based approaches and models are different ways to target the same problem. In the context of COVID-19, they want to predict the number of infections that will take place in the future. Data-driven predictors convinces with their simplicity of implementation. If a sufficient amount of experimental data is available, they fit the existing curve using mathematical calculations and make meaningful future predictions. However, if not enough or only bad quality data is available (e.g. at the beginning of an epidemic), there is no chance a data-driven approach performs well. This is where the advantages of model-based approaches come to the fore. Implementing a good model is indeed associated with a high cost of implementation, but it can pay off. High prediction precision can be achieved by modeling any kind of scenario, that has not yet taken place in the reality (e.g. behavioral constrains like social distancing or wearing masks). The dynamic of the states can be estimated and predicted at each time point, which makes models more versatile in comparison to data-driven approaches. Nevertheless, applying models need in general more resources of computing power and are more time consuming. Summing up, at different times in an epidemic either data-driven or model-based approaches can be the better choice.

10.12.1 Data

To evaluate the forecasting approaches three different data sets were created. Each dataset contains the accumulated confirmed cases for the COVID-19 pandemic for 60 days.

Actual data for Germany:
The actual COVID-19 case numbers for Germany were downloaded from the RKI git repository [31]. The analysis of this project was performed on the reported cases from 02.03.2020 till 30.04.2020.

SIR:
The SIR data was generated using the setup and the parameters as described in 10.7.1. The initial population size was set to the population size of Germany and the number of exposed people was set to 8000 to model an ongoing spread, which makes it more comparable to the real data from RKI. Note, that exposed means for the case of the SIR model, that the exposed individual will be infected at some point. In contrast to the ABM model, where exposed is defined as individuals that had contact to an infected person but will not necessarily get infected as well.

ABM:
To generate a third dataset, an ABM approach was used that contains central locations (supermarket and school) and models a scenario where the social contact is reduced and masks are worn. The number of agents was set to the population density of Hamburg (2438 inhabitants per square kilometer), but the results where upscaled in relation to the entire German population, which explains the high number of infections in this dataset.

In Figure 10.29 one can see that the curves of the RKI dataset and the SIR dataset are more flattened compared to the ABM data. For testing, all three data sets are split into a training and a test data set by the last 10 days.

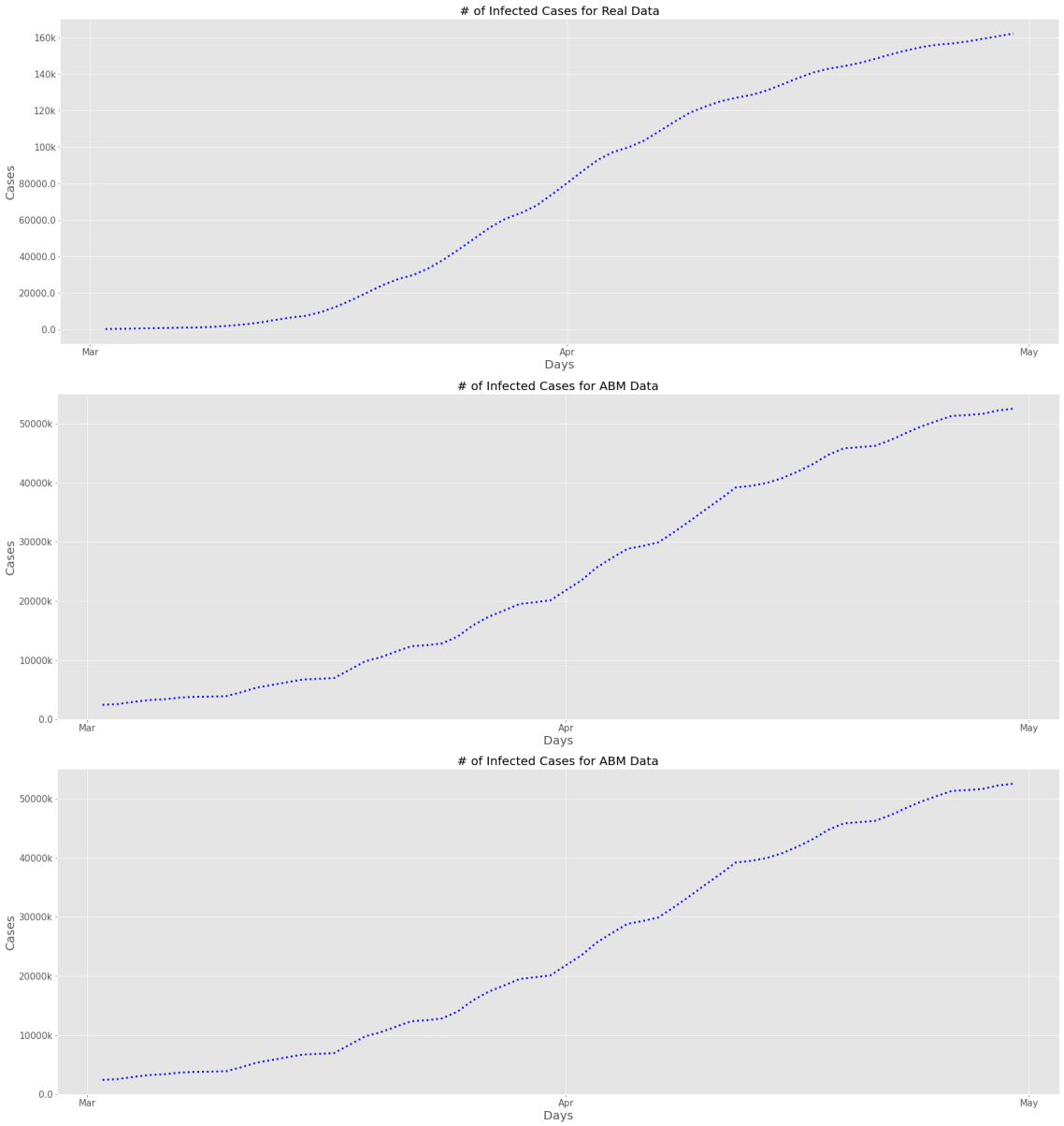


Figure 10.29: Number of confirmed cases.

10.13 Approaches for data-based time-series prediction

10.13.1 Prophet Library

The Prophet model [37] is composed of the three components *trend*, *seasonality*, and *holiday effects*, where each of its components is added together with time as its regressor:

$$y(t) = trend(t) + seasonality(t) + holidays(t) + error(t)$$

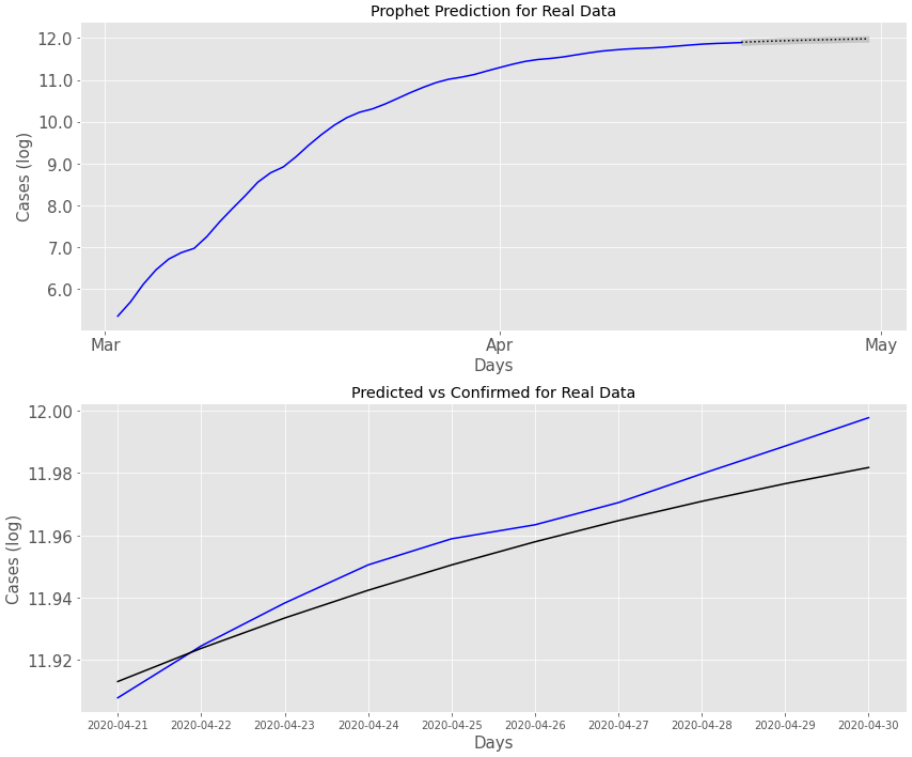
where:

Trend models *non-periodic* changes. This component is the most important one for our analysis since our data has no seasonality in our data like it is the case for i.e. the amount of rain in a certain country over many years.

Seasonality models *periodic* changes (weekly, monthly, ...). Since our data sets include only 50 days of records, there is no seasonal trend present yet. This component would be more useful when we fight COVID-19 for the next upcoming years and use this data to predict years that lie even more in the feature. Over the years there might be an increase over the colder months of the year and and decrease of new cases over warmer months of the year.

Holiday Effects allow the model to include short time intervals with an abnormal trend. When modeling Corona in Berlin with no restrictions, this could be an public event like the Karneval der Kulturen where many people from different households interact closely with each other, which would probably lead into a spike in the number of confirmed cases. Since Germany restricted most public events early in the pandemic this component is also not of greater interest for our analysis.

Prophet forecasts by fitting a curve to the input data that is then prolonged for future values with the three mentioned components. Although, Prophet aims to produce a strong forecast without much hyper-parameter tuning.For our analysis the prediction did not fit the actual trend of the test data. We achieved better results by changing the *growth* parameter from linear to logistic. Since the three datasets do not contain many breakpoints (points in the data, where the growth changes) the non linearity of a logistic function can nestle closer to the slow decrease in the growth of the confirmed cases of the test data.



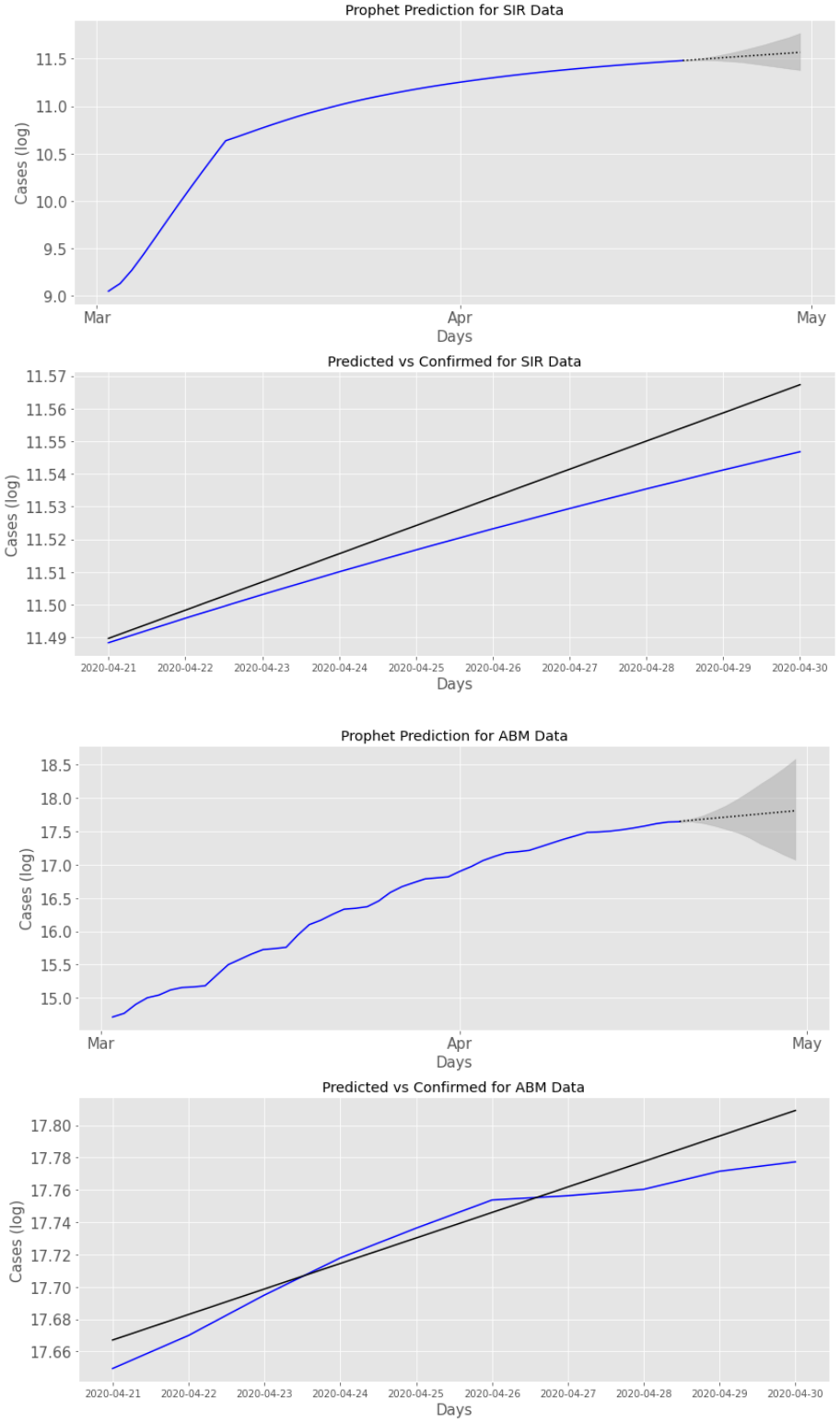


Figure 10.30: In each pair of figures the results of prophets time series prediction is displayed. The top plot shows in blue the recorded data and in black the forecast. The gray area represents the confidence intervals. The bottom plot zooms into the forecast depicts the test data to the predicted data. The results were achieved as introduced in section 10.13.

10.13.2 Machine Learning (e.g. LSTM neural networks)

Long Short-Term Memory Networks (LSTM) are a prominent deep learning method for time series prediction. LSTM are part of Recurrent Neural Networks (RNN). In contrast to common feed forward networks where each layer is only connect to the subsequent layer, RNN has layers connected to itself or even previous layers. This is more closely modeled on the neural connections exhibited by the neocortex and allows the network a greater prediction accuracy for time coded data.
The LSTM network is employed by using the python library *keras*. For the prediction a LSTM of the length of the training data with a simple 1-dense hidden-layer is created. The model itself is optimized with adam and lossed by the mean_squared_error. The data needed to be preprocessed. The first step of preprocessing is called shifting. In shifting the time series data is shifted by a constant value, dividing the data in a training value and expected value. Subsequently, because LSTM need to evaluate local differences in the data, each time step is subtracted by the shifted value. As in all neural networks a common scalar is used normalizing the data between -1 and 1 with a mean of 0. We plotted the predicted and actual time series data of the last 10 days. For all three test sets the RMSE is calculated (Figure 10.31, 10.32, 10.33).

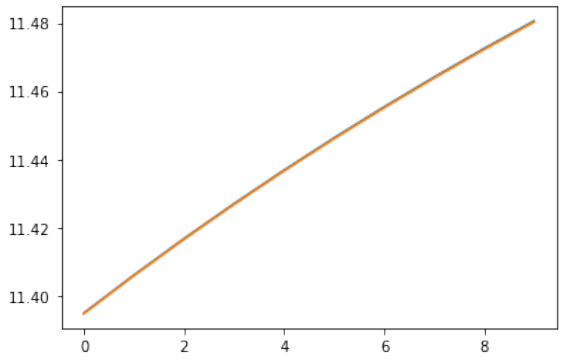


Figure 10.31: Predicted (orange) vs expected (blue) on SIR simulated data for the LSTM model as introduced in section 10.13.1. (y-axis = log of infected people, x-axis = days)

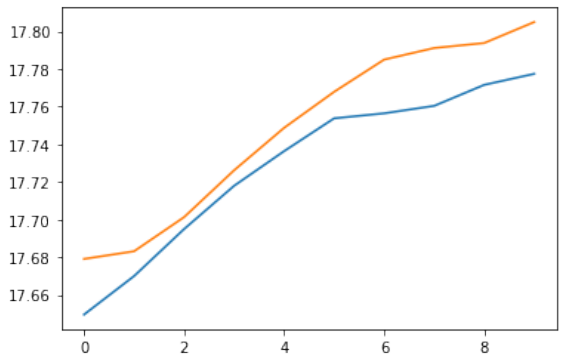


Figure 10.32: Predicted (orange) vs expected (blue) on ABM simulated data for the LSTM model as introduced in section 10.13.1.(y-axis = log of infected people, x-axis = days)

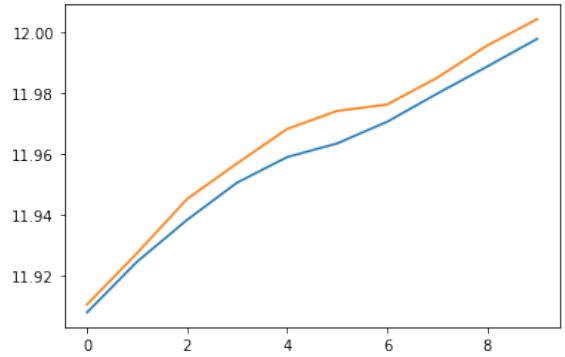


Figure 10.33: Predicted (orange) vs expected (blue) on RKI actual data for the LSTM model as introduced in section 10.13.1. (y-axis = log of infected people, x-axis = days)

10.13.3 Classical models

For the classification and forecasting on the time series problems there are various machine learning approaches. One among them is by using the classical methods. As it focus on various linear relationships, they perform well on a wide range of problems. They also perform well if the data is suitably prepared. There are nearly eleven types of classical methods, all lead to forecasting a different time series problem. From these methods we chose the Autoregressive Integrated Moving Average method (ARIMA). The reason which makes this model more significant is that it gives very low residual sum of squares (RSS) which is helpful for building the model and fit it. The lower the RSS, the lesser will be the amount of variance. However, ARIMA models the next step in the sequence as a linear function of the differenced observations and residual errors at prior time steps. It is the combination of both Autoregression (AR) and Moving Average (MA) and helps in differencing pre-processing step of the sequence stationarity called integration. The notation for the model involves specifying the order for the AR(p), I(d), and MA(q) models as parameters to an ARIMA function, e.g. ARIMA(p, d, q). An ARIMA model can also be used to develop AR, MA, and ARMA models.

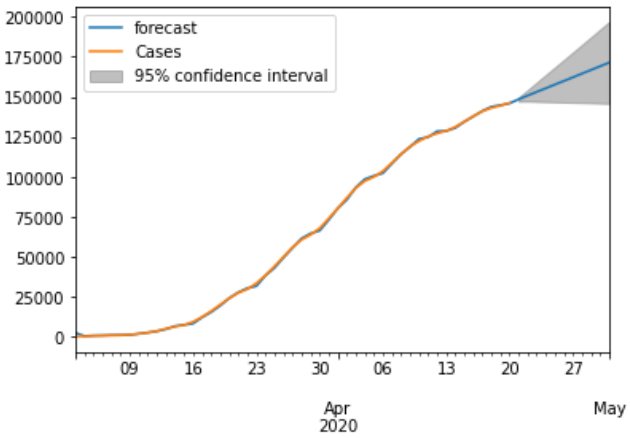


Figure 10.34: Cases (orange) vs forecast (blue) on RKI data for the ARIMA model as introduced in section 10.13.2 (y-axis = confirmed cases, x-axis = time period). The gray area represents the 95% confidence interval for the 10 days forecast.

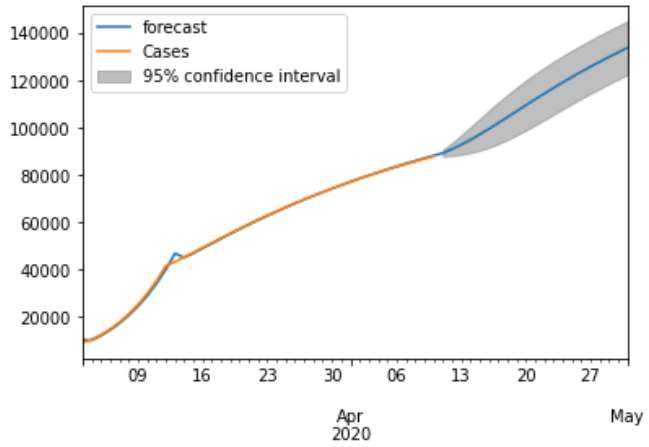


Figure 10.35: Cases (orange) vs forecast (blue) on SIR simulated data for the ARIMA model as introduced in section 10.13.2 (y-axis = confirmed cases, x-axis = time period). The gray area represents the 95% confidence interval for the 10 days forecast.

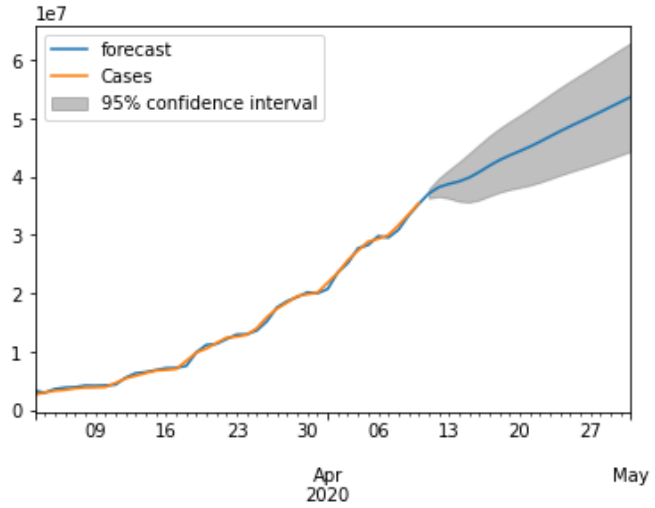


Figure 10.36: Cases (orange) vs forecast (blue) on ABM simulated data for the ARIMA model as introduced in section 10.13.2 (y-axis = confirmed cases, x-axis = time period). The gray area represents the 95% confidence interval for the 10 days forecast.

At first we imported the datasets, which contain 60 observations of the confirmed number of corona cases. We converted the dataframe to numpy (time series) for easier prediction. Then, the important part is making the time series stationary. In other words the statistical properties (mean, variance) should remain constant over time. Furthermore, we applied a smoothing function to make the time series stationarity. Afterwards, we plotted Autocorrelation (ACF) and Partial Autocorrelation (PACF) to understand the parameters (p,d,q) of the ARIMA model. They basically indicate the correlation between two time instances as well as the degree of association. Finally, we trained a certain duration of the data sets, built the ARIMA model and plotted the Root Mean Squared Error (RMSE) and forecasted the predictions for the test set. Figures 10.34, 10.35 and 10.36 represent the ARIMA model plotted for the RKI, SIR and ABM datasets respectively.

10.14 Comparison of data-based time-series prediction

Prophet :

In Figure 10.30 it can be seen that the fit is rather suboptimal, especially for the ABM data. The fluctuating trend of the AMB and the RKI dataset is not captured at all and the only reason for the small RMSE values are the low amplitudes of the fluctuations. The best fit we expected for the SIR data whose trend is almost linear because it fits the general trend of the Prophet predictions. However, the predicted period is overestimating the true values. By day ten the residual is the biggest for the SIR dataset.

LSTM :

LSTM is the *"winner"* and produced the best RMSE values for the SIR and RKI data set (Table 10.7). For the SIR simulated data it is due to the almost linear trajectory (Figure 10.31), where neural networks have an effortless fit. For the RKI data it could also reproduce a delayed flattened of the curve at day four (Figure 10.33). The high ABM value is due to the high fluctuations within the data where the influence of the local differences might be to high. Still a delayed flattening of the curve was predicted (Figure 10.32)

ARIMA (classical approach) :

From the above obtained plots and the RMSE values, we can say that the fit is bad for the SIR data set when compared with the RKI and ABM data (Figure 10.35). By interpreting the RMSE of all three predictions, it can be seen that the RMSE is very high for the SIR model showing there is a high variations with predictions when compared with the other kinds of data. The obtained RMSE differ significantly with 0.022, 0.147 and 0.028 for logarithmized RKI, SIR and ABM data sets respectively.

Dataset	Prophet	LSTM	Classical
SIR	0.011	0.000	0.147
ABM	0.0155	0.021	0.028
RKI	0.008	0.007	0.022

Table 10.6: Comparison of RMSE values for three data sets and three models as described in chapter 10.12.1.

10.15 Model-vs. data-based time-series prediction

For the fitting of the data-based model prediction a SIR model was used, which was fitted to data from Germany. The data used for the adjustment was generated for the period 03.02.2020-30.04.2020, using both the other SIR model and the ABS model. The obtained data and the parameters are already known, and it is necessary to define initial estimates and lower and upper limits for the unknown ones in order to support the curve fitter and obtain good results. For the fit, a function is needed that takes exactly one x-value as the first argument (the tag) and all parameters to be fitted. It returns the confirmed cases predicted by the model for that x-value and parameters, so that the curve fitter can compare the model prediction with the exact data. To perform the fit, it is necessary to initialize a curve fitting model, set the parameters according to the initial, minimum and maximum, specify a fitting method (e.g. *leastsq*) and perform the fit. One of the important parameters is outbreak_shift. The case data starts on 02.03.2020, so our model assumes that the virus started to spread on this day. For this reason this parameter was set to zero. Others were the

parameters R0_start (initial value of R_0 for the period), R0_end (final value of R_0), k (rate of decline of R0), x0 (start time of decline of R0), inf_to_rec_d (daily probability of transition from infected to recovered state), which influence R_0 and the infection rate (β), respectively, and inf_to_dead_p (rate of transition from infected to dead state). In figure 10.37 one can see a plot of the fit for the extracted data from the sir model and the ABM model respectively.

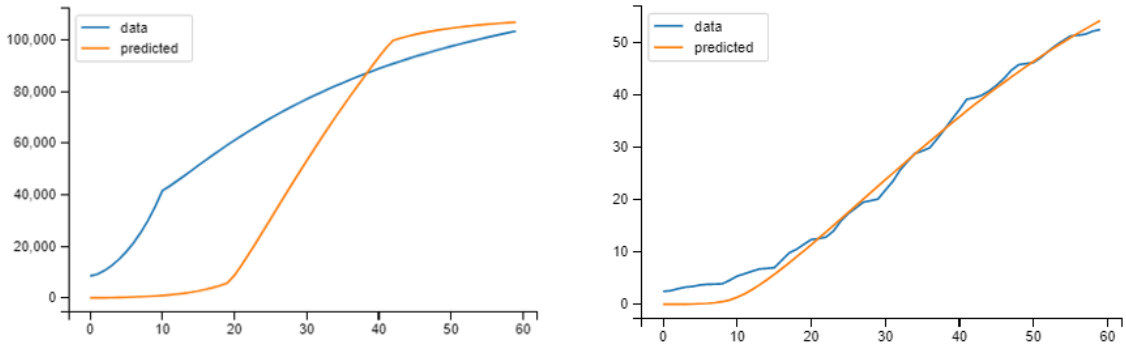


Figure 10.37: Predicted (orange) vs expected (blue) on SIR simulated data for a sir model. (y-axis = confirmed cases for SIR data, confirmed cases $*10^6$ for abm data, x-axis = days)

The parameters predicted by the fitter are shown in the Figure 10.38. The most of the parameters are nearly the same for both fitted data. Only the parameter R_0_start and x_0 differ significantly between them. The R_0_start of 2.76 for SIR data is more expected. The high value for R_0_start of 10.68 for the ABM may be possibly explained as an effect of upscaling the ABM data. Also the numbers of confirmed cases from sir data and from ABM data differ significantly, which may also be due to the upscaling of ABM data. The parameter x0 (begin of R0 decline) for SIR data was fitted to 17, and for abm data to 9. It may be as the SIR data and ABM data were generated for different scenarios. Due to the fitting to the SIR model it looks not well optimized for SIR data. The possible problem is that the data were created with a modified SIR model that was adopted for scenarios (with a parameter that changes rate of transition from susceptible to the exposed state due to the restriction and its duration). For this reason, the data may not fit together well. For ABM the fit performed much better because the model is very similar to the SIR model used for the fit.
By predicting for 10 days it is also to see that the confirmed cases for the ABM data are higher than for SIR data (see Figure 10.39 . The curves of expected and predicted for SIR data are not optimal adjusted (rmse=0.052), but they have the same trend and they approach each other and the difference will be reduced. The curves of expected and predicted for ABM data are optimal adjusted (rmse=0.015) (see Table 10.7).
In comparison to the SIR model based prediction, LSTM was able to better predict SIR data (rmse=0) but ABM data the prediction of SIR model (rmse=0.015) was better than LSTM (rmse=0.021). Taking all together, both approaches were able to deal with non-stationary data as number of confirmed cases are. However, the SIR model can not always make optimal forecasting even for data extracted from another SIR-model with a different parameter setup.

Dataset	**SIR-model**	**LSTM**
SIR	0.052	0.000
ABM	0.015	0.021

Table 10.7: Comparison of RMSE values for the models as describes in chapter 10.12.1

```
{'R_0_end': 0.3413676232485514,
 'R_0_start': 2.7584555587301667,
 'inf_to_dead_p': 0.16000000000000003,
 'inf_to_rec_d': 0.32371753521779184,
 'k': 22.681536008041032,
 'x0': 17.06903340839481}
```

```
{'R_0_end': 0.6903763998947229,
 'R_0_start': 10.67957469641348,
 'inf_to_dead_p': 0.16000000000000003,
 'inf_to_rec_d': 0.20168712000334932,
 'k': 19.99999962543517,
 'x0': 9.177538495703322}
```


Figure 10.38: fitted parameters for SIR simulated data (left) and ABM simulated data (right) for a sir model

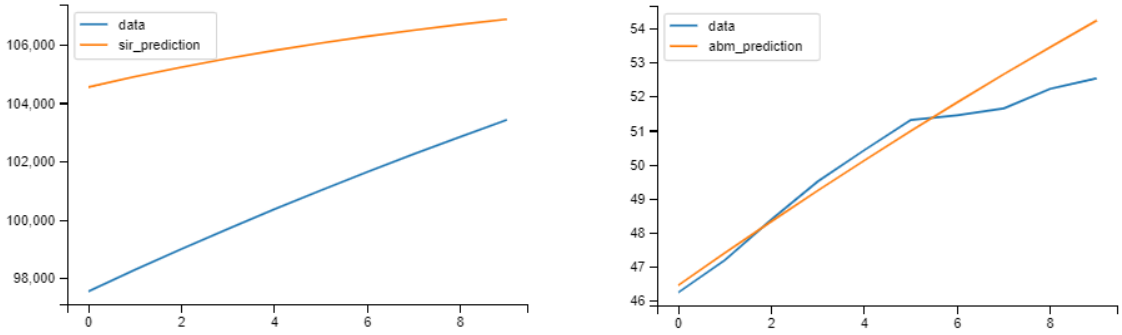


Figure 10.39: Prediction for SIR simulated data (left) and ABM simulated data (right) for a SIR model. Predicted(orange) vs expected (blue) (y-axis = confirmed cases for SIR data, confirmed cases $*10^6$ for ABM data, x-axis = days)

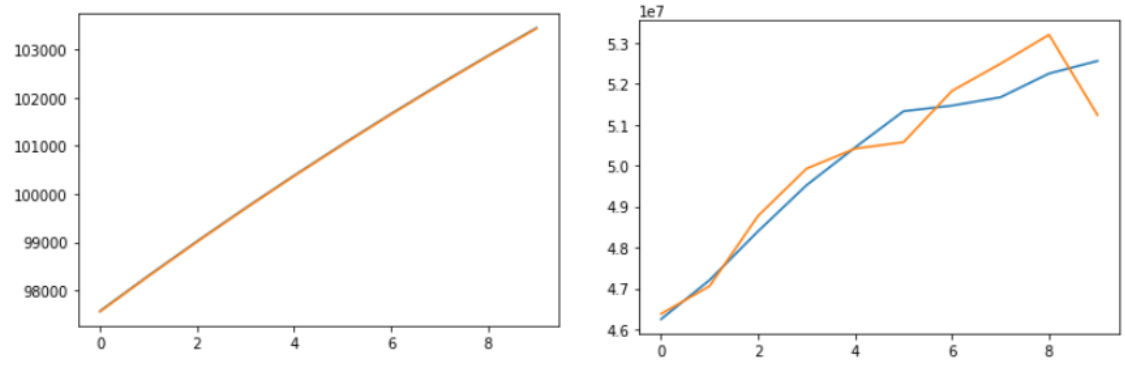


Figure 10.40: prediction for SIR simulated data (left) and ABM simulated data (right) for a LSTM model

10.16 Towards COVID-19 outbreak prediction

The results of the forecasting using the distinct time series approaches ARIMA, Prophet and LSTM as well as the model-based approach using a SIR model were evaluated by comparing the forecasting plots with the actual data as reference and analyzing the root mean square values. ARIMA performs well on forecasting stationary data of short periods. Prophet is more advanced then ARIMA and offers also the possibility to identify trends and seasonality. Unfortunately, our datasets only contained 60 days of time series data. Since seasonality is already proven to exist for other viruses (e.g. influenza [24]), it will be an interesting experiment to analyze the existence of seasonality for COVID-19 on long-term time series data. Such a study would distinguish Prophet's strengths. However, the data we used was non-stationary. That could be an explanation why ARIMA underperformed and showed the biggest RMSE values for all three datasets compared to the other methods. LSTM was the clear winner as it is optimised for dealing with non-stationary data and our results confirmed: LSTM produced the lowest RMSE values for the RKI and the SIR dataset. The results further demonstrate that, given data of confirmed COVID-19 cases, LSTM can learn and scale to more or less accurately estimate the amount of the people that will become infected in the future. Naturally, the prediction is only a tendency and need to be scrutinized very critically. Nevertheless, it is possible to predict a course of the outbreak. And the more data becomes available, the better the results of the data-driven forecasting methods will be.

VI

Part 6

11. Introduction

11.1 Background

After exploring several time-series prediction methods in the last week's project we will focus in this week's project on their visualization. Recall, forecasting is done by developing models that capture the characteristics of present data to make future predictions. The underlying mathematics and statistics of those models can be very overwhelming for people from other fields such that an easily interpretive presentation is an important but sometimes neglected part of the forecasting pipeline.

11.2 Project Description

The aim of this weeks project is to combine time-series prediction methods as introduced in the last week combined with various visualization techniques. The time-series prediction is performed with Facebook's prophet library (section 10.13) on the data given by RKI [31] for the federal states of Germany. Additionally, clustering is performed to group the federal states of Germany by the confirmed number of cases, deaths, and recovered individuals. The visual data exploration involves tree-maps, age, and federal state dependent bar plots and GIFs that illustrate the change of the confirmed number of cases over time.

11.3 Outcome

Get the visualizations with the bar charts and tree map for the number of infected cases and deaths in Germany and its federal states and distribution of the same for different ages and gender.The resulted plots represent a detailed exploration of the outbreak. A clear distinction between case trajectory curves of different federal states for germany could be drawn. The time-series prediction via prophet showed an increase in overall cases for each federal state. A **GitHub Repository** has been created that contains the three mentioned GIFs.

12. Solution Approaches

12.1 Visual Exploration

As described in the goal, the data set is taken from the Robert Koch Institute (RKI) site and for the analysis, Germany and its federal states are considered. The time-span is defined from the 28th of January till the 15th of June. The starting date is based on the first occurrence of the coronavirus within the German borders in the state of Bavaria. In order to understand the severe disease and to analyse the outbreak closely, a user friendly data visualization model with the help of different plots(use of bar charts and tree maps) related to the various criteria was used. The various criteria are based on parameters like number of confirmed cases and deaths. When performing the visualizations we had some key thoughts in mind:

Which states in Germany are mostly affected?
How the confirmed and death cases are distributed all over the Germany?
Which countries have the most deaths?
Comparison of the cases w.r.t age groups for various federal states

As a first visual overview each per state case trajectory was plotted (Figure 13.10). Two national counter measures were integrated, first being the general closure of schools and public spaces beginning with the 16th of March and the second being the introduction of mandatory masks beginning with the 27th of April. Next, we began plotting the data in the form of bar charts, here we obtained the visualizations for the number of infected cases and deaths in Germany and its federal states and distribution of the same for different ages and gender. The resulted plots represent a detailed exploration of the outbreak. Also represented the visualization of the number of confirmed, death and recovered with the tree map. Additionally, three GIFs are generated for the number of confirmed, death, and recovered cases. The time-span is defined from the 3th of March till the 19th and the code is taken from the **GitHub Repository et al. Chang Chia-huan** and extended by also plotting the number of recovered individuals.

12.2 Time-Series Prediction via Prophet

While masks are still mandatory a lot of countermeasures were eased up in the month of June, thus we performed a time series prediction for each individual state to follow up on the trajectories of cases (Figure 13.11). For the time series prediction we used a 14 day future prediction from the facebook prophet library. For the parameters seasonality was disabled because the explored time scale is to small to account for trends due to seasonal changes. All other parameters were defaulted.

12.3 Clustering

Clustering can be done in several ways. To get a quick insight into how data can be clustered, one way is to form hierarchical clusters. A hierarchy can be formed in two ways: start from the top and split or start from the bottom and merge. It was decided to perform the latter. First, the raw data containing confirmed cases, deaths, for each day for the rough period from March to 15th of June were imported for all German states. The data had to be made consistent such that the dates were synchronized for all federal states. Clustering should be done using time series data such as confirmed cases, deaths and recovered cases. To do this, you should first define the linkage function. The linkage function takes the distance information and groups pairs of objects into clusters based on their similarity. The parameters of the linkage function like metric and method may be adjusted and are set to "euclidian" and "ward" by default correspondingly. The keyword 'ward' causes the linkage function to use the algorithm to minimize the ward variance, and the keyword 'Euclidean' causes the Euclidean method to be used as a distance measure. These newly formed clusters are then linked together to form larger clusters. This process is repeated until all objects in the original data set are linked together in a hierarchical tree (dendogram). So a dendogram is a plot of clusters by hierarchical clustering, where the length of the bars represents the distance to the nearest cluster center. To find similarities between timeseries, it is worth using k-means to cluster them, since k-means is a kind of unsupervised learning and one of the most popular methods to combine unmarked data to k-clusters. The process starts with k centroids, which are randomly initialized. These centroids are used to assign points to the nearest cluster. The mean value of all points within the cluster is then used to update the position of the centroids. These steps are repeated until the centroid values stabilize. Before performing k-means clustering, it is necessary to identify the corresponding optimal number of clusters k. The elbow method as one way to estimate the value of k. To give equal importance to all features, it is needed to scale the continuous features, which will be done using the StandardScaler. For each k-value k-mean values will be initialised and the inertia attribute will be used to identify the sum of the squared distances of the samples to the nearest cluster center. As k increases, the sum of the squared distances tends toward zero. By plotting k values against the sum of the squared distances a plot should have a form of the arm and the optimal k should be then at the elbow.

13. Results

13.1 Visual Exploration

As first it can be seen that the number of deaths is for Germany is quite low compared to the confirmed cases, meaning that most infected individuals outlived the disease. Also, the confirmed cases and the recovered go hand in hand which is seen (Figure 13.1).

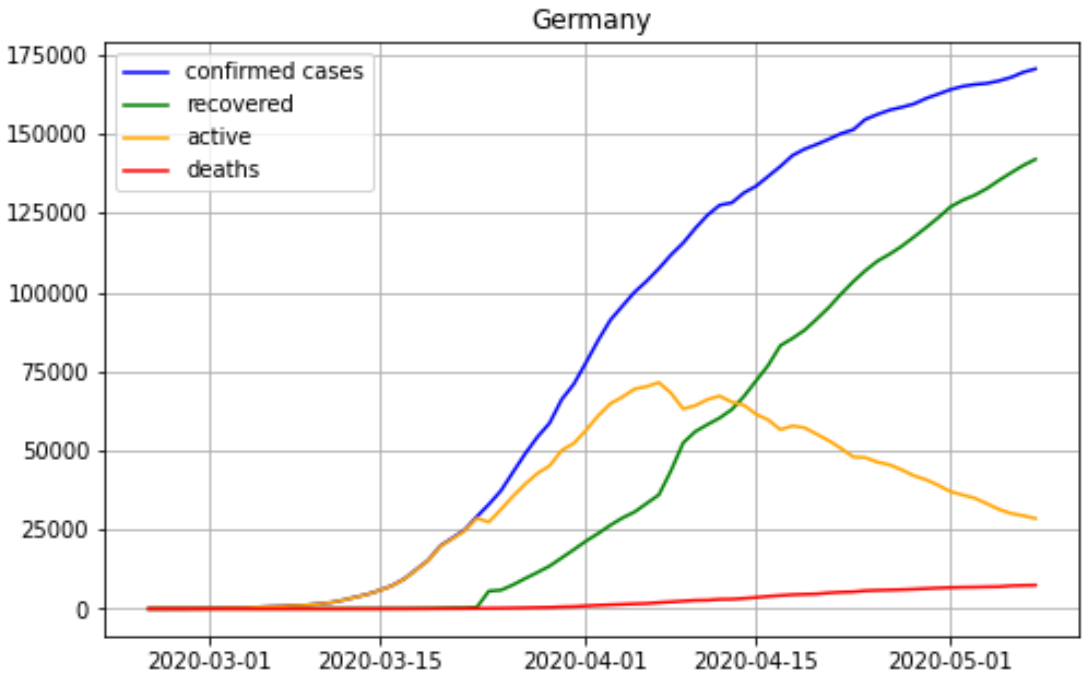


Figure 13.1: A trajectory plotting for entire Germany. The number of confirmed cases, active, recovered and deaths were visualized as dashed lines with different color codes. The number of confirmed cases and recovered cases go hand in hand

Here the three states viz. Bayern, Baden-Württemberg, and Saarland (Figure 13.2) are on the top with the highest number of cases, slightly followed by Hamburg and so on. The deaths are minimum as most of the preliminary measures were taken place soon after the spread.

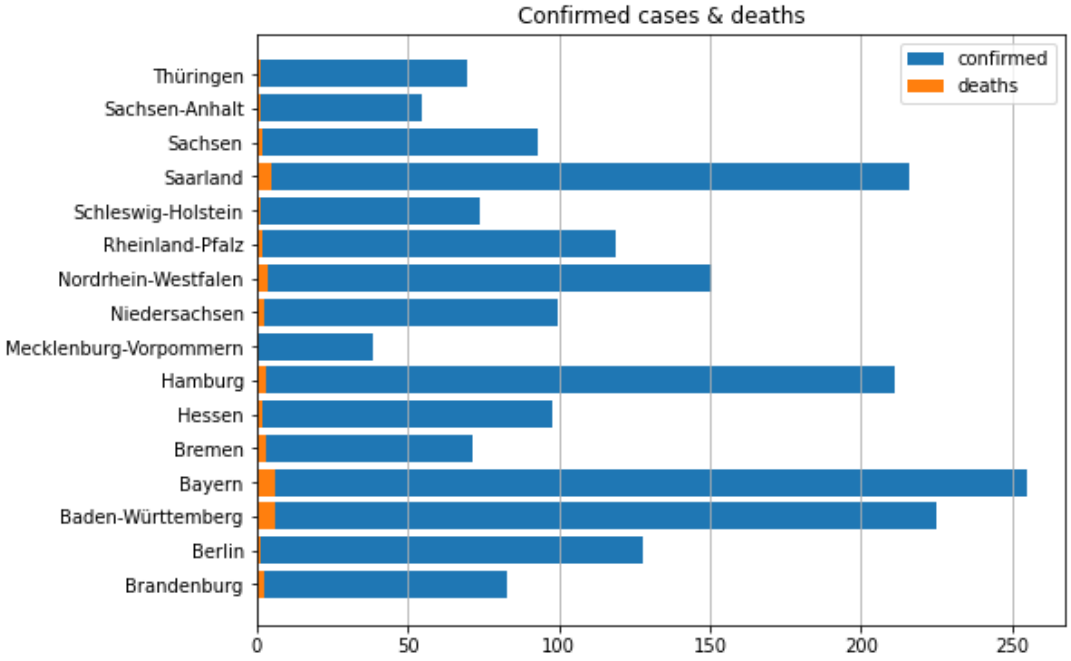


Figure 13.2: Bar chart representing comfirmed cases and fatalities per 100k population. x axis: number of cases, y axis: federal states of Germany. Blue bars: number of confirmed cases, orange: number of deaths

The next plots show the distribution of confirmed cases and deaths for different ages and gender for Germany and its federal states which are shown (Figure 13.3, Figure 13.4 and Figure 13.5). Here the age groups ranging from 35-59 has high number of cases when compared with other age groups and the population of the range between 80-99 have shown the high death cases indicating the virus easily attacks the old people who has less immunity for the disease. When the individual federal states are considered then the results vary consistently, However some states like Mecklenburg-Vorpommern, Rheinland-Pfalz and Thueringen has shown more deaths with the men signifying the dominance in the male death rate. The treemap gives a breif understandings of distribution of number of confirmed, deaths and recovered cases for the federal states of Germany which is shown in the Figure 13.6
As already pointed out in section 13 the number of confirmed cases varies through the federal states and peaks in the south of Germany. Consequently we have similar results for the death and recovered statistics. To simplify the the comparison between the federal states the reported number of cases are normalized by 1M residents. For example Bremen (top right corner of Germany) has less than 600k residents but the 1300 confirmed cases lead it to the federal state with the relatively highest number of conformed cases, while it has absolutely one of the lowest.

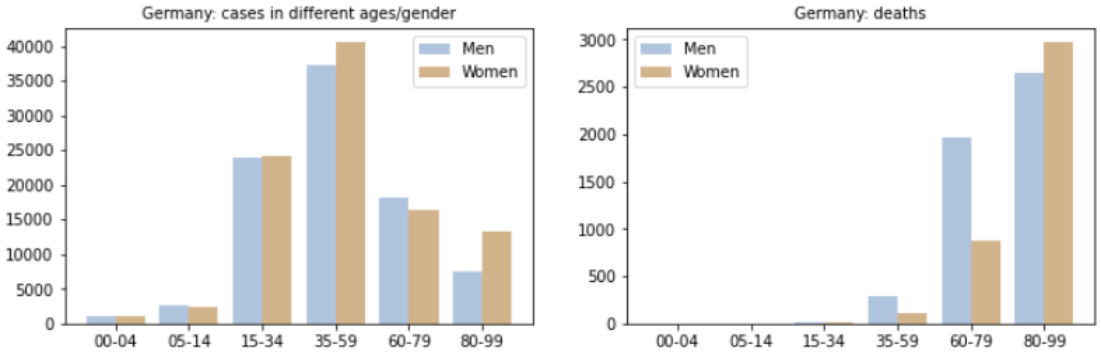


Figure 13.3: Bar charts showing the distribution of confirmed cases and deaths for different ages and gender for Germany. x axis: age groups, y axis: number of cases. Blue bars: number of cases in men, brown: number of cases in women

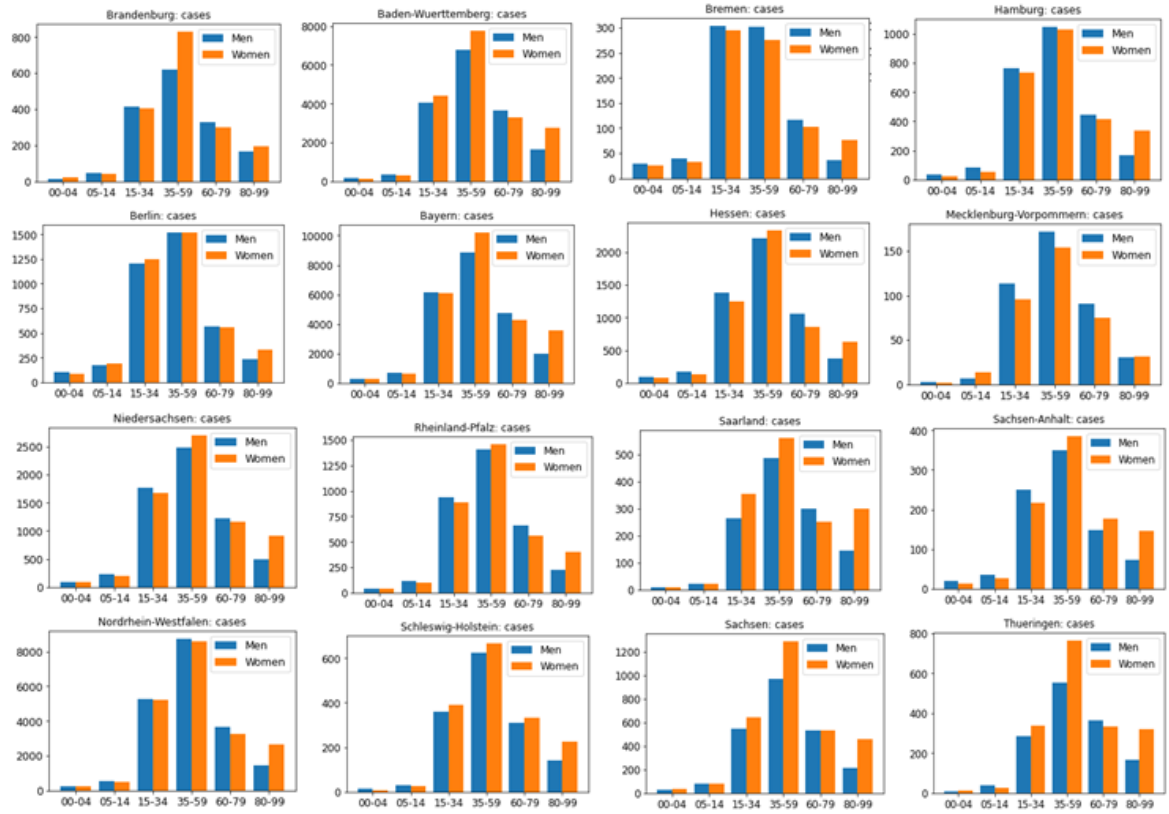


Figure 13.4: Bar charts showing the distribution of confirmed cases for different ages and gender for all the federal states of germany. x axis: age groups, y axis: number of cases. Blue bars: number of cases in men, orange: number of cases in women

The final GIFs for all three conditions can be found in our public **GitHub Repository for Week 8**.

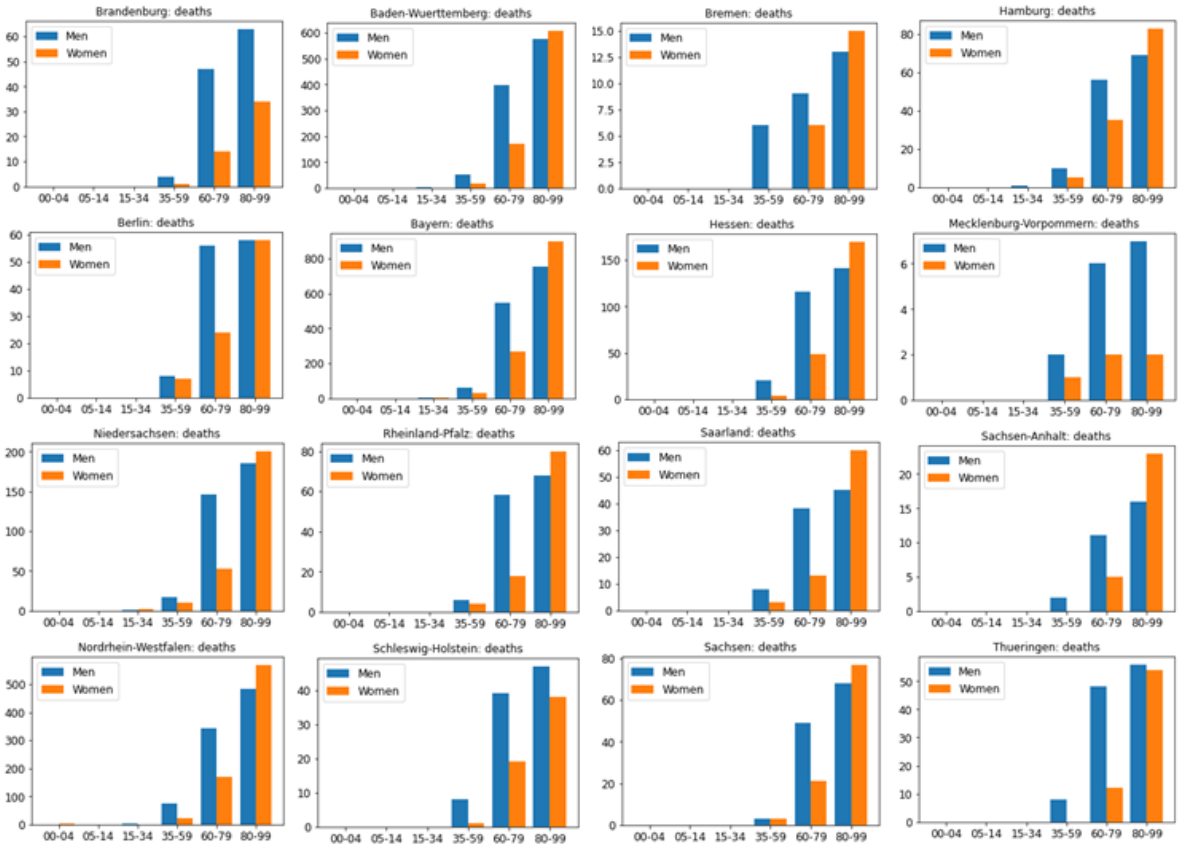


Figure 13.5: Bar charts showing the distribution of deaths for different ages and gender for all the federal states of germany. x axis: age groups, y axis: number of cases. Blue bars: number of cases in men, orange: number of cases in women

13.2 Time-series and Prophet Prediction

Following the general outline of the curves, substantial differences in overall case numbers between different states can be seen e.g. Baveria compared to Hessen (Figure 13.10). This might be due to the initial outbreak originating from the severely affected southern states of Europe. While the first introduction of general lockdown measures does not have an immediate effect it has to be noted then each introduction of measures should only be apparent after two weeks, as this is the time of infection till recovery of an individual affected by corona. Even after two weeks a full exponential increases of cases can be seen. On the contrary the introduction of mask does seem to have had an apparent effect. A noticeable downslope can be seen for each sate. Prophet predicts an increase in cases for each state (Figure (Figure 13.11) Especially for Berlin and Rheinland-Pfalz a noticeable increase in cases can be seen. Thus it might be advisable that the early easement of anti corona measures are leading to a recurrence of cases.

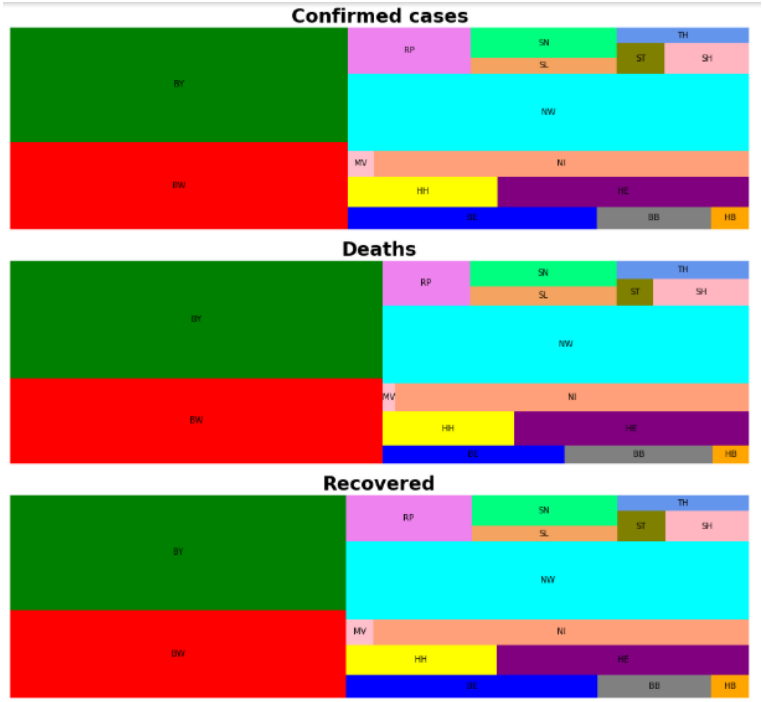


Figure 13.6: Tree map showing the distribution of confirmed death and recovered cases for all the federal states of germany. Different color codes represents the different federal states of Germany

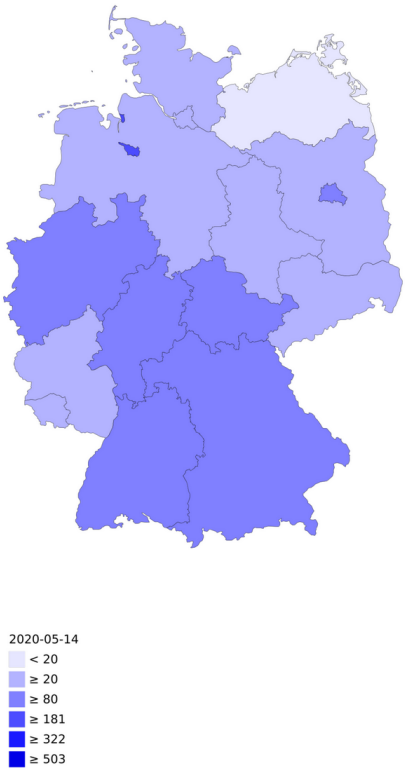


Figure 13.7: Number of confirmed cases per million residents for the federal states of Germany by the 14th of May based on the reported data in [31].

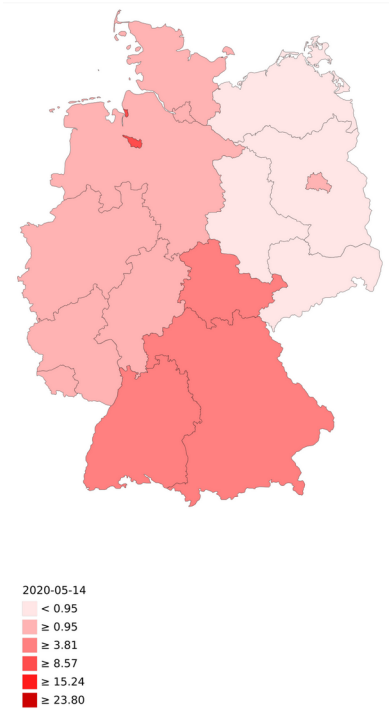


Figure 13.8: Number of cases that died to corona per million residents for the federal states of Germany by the 14th of May based on the reported data in [31].

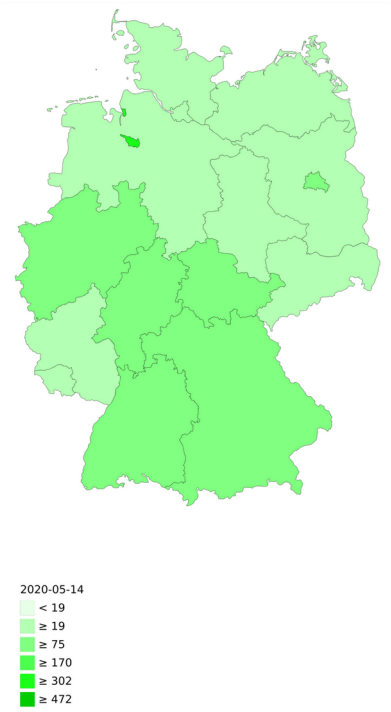


Figure 13.9: Number of recovered cases per million residents for the federal states of Germany by the 14th of May based on the reported data in [31].

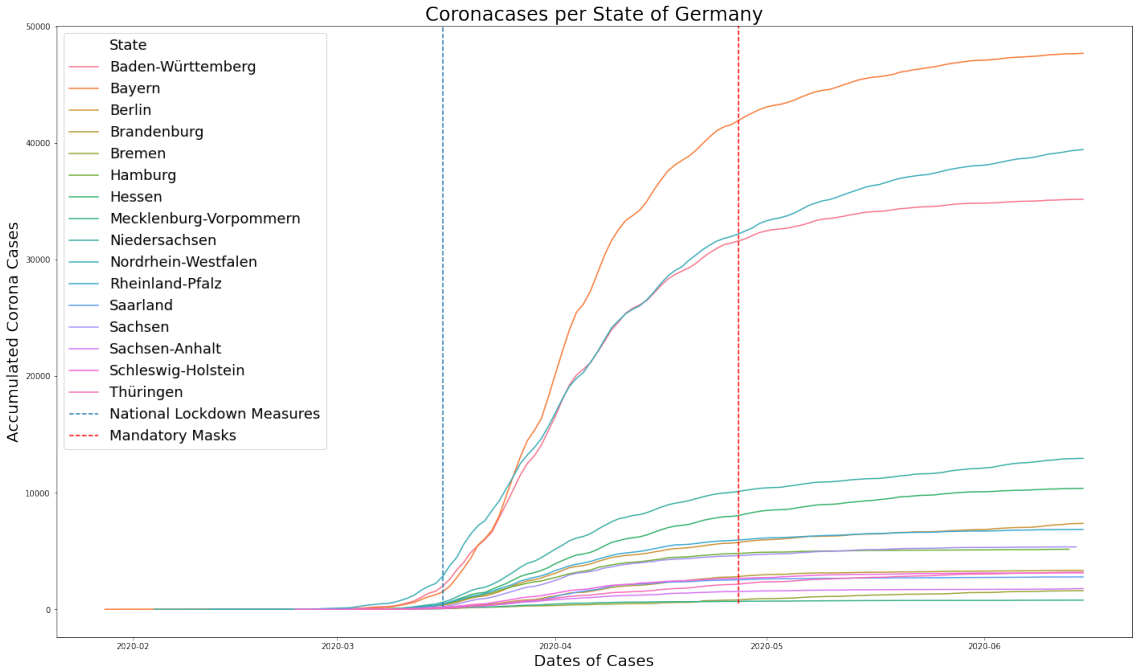


Figure 13.10: Cases trajectory plotting for each state of Germany. Two statewide coronavirus counter measures were visualized as dashed lines. The cases study follows from the 28th of January till the 15th of June. All curves follow a noticeable exponential increase with a subsequent decrease in cases.

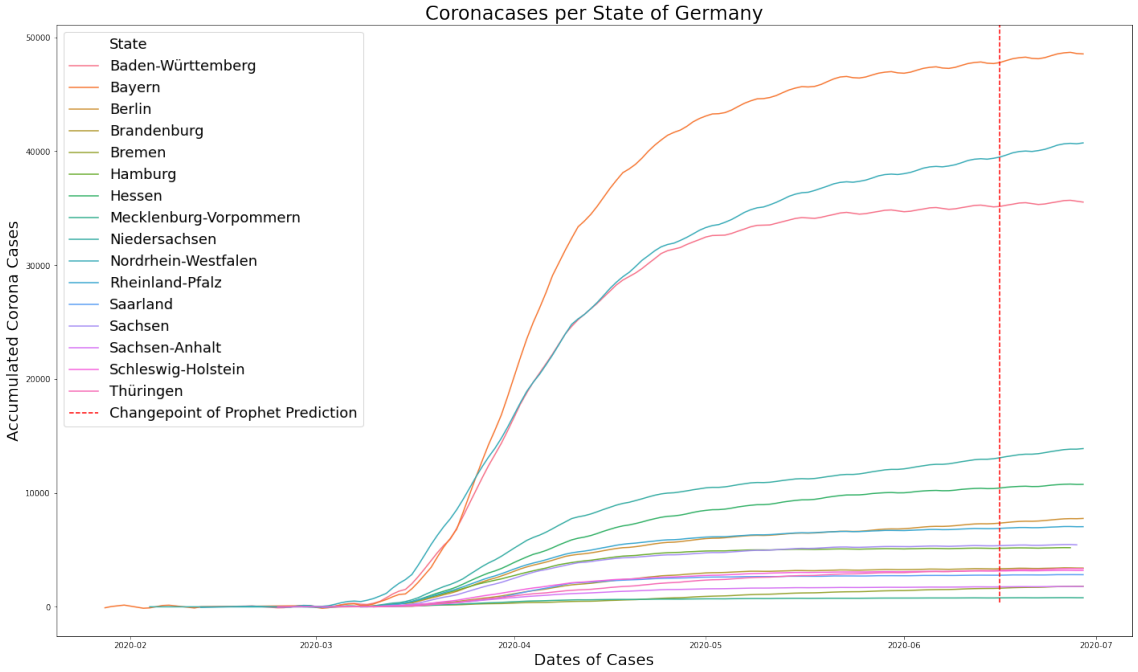


Figure 13.11: Cases trajectory plotting for each state of Germany. Predictions of the prophet library are plotted after the dashed lane, beginning with the 16th of June. For each state a noticeable increase in cases is predicted.

13.3 Clustering

In order to check the data for clusters, hierarchical clustering was done first. The figure shows the dendogram of the hierarchical clustering of confirmed cases (see 13.12,13.13 and 13.14).

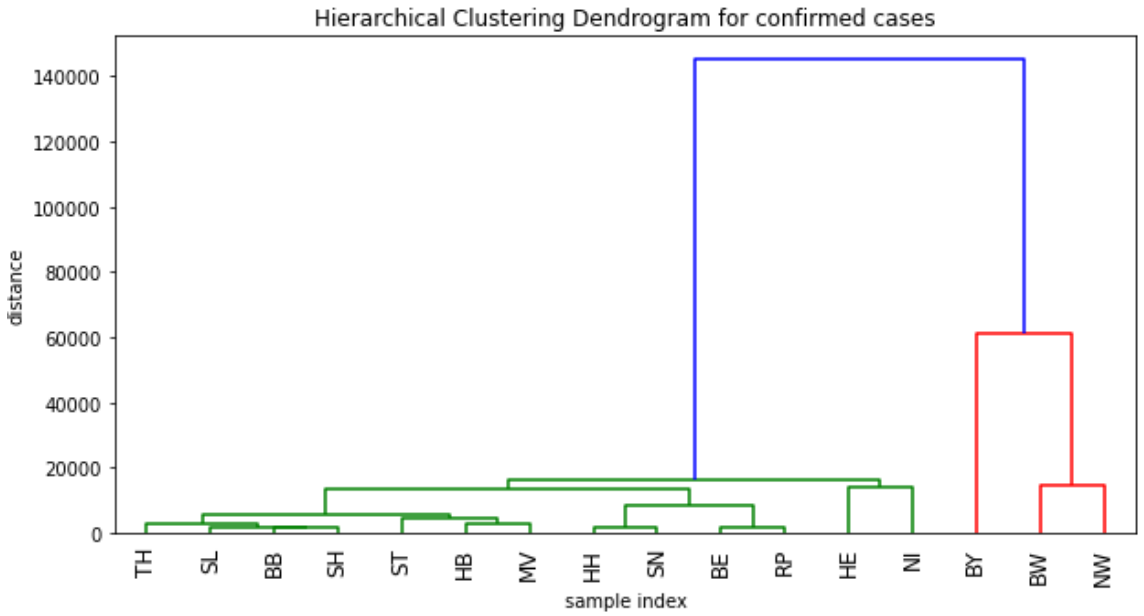


Figure 13.12: Hierarchical Clustering Dendogram confirmed cases for the period March-June 2020

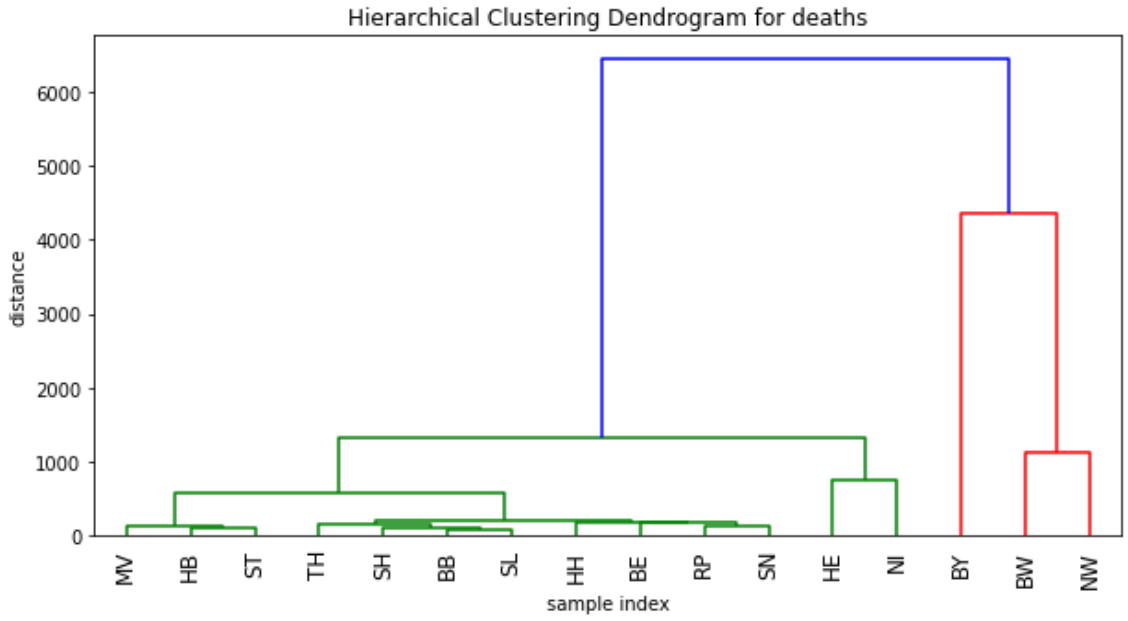


Figure 13.13: Hierarchical Clustering Dendogram to deaths for the period March-June 2020

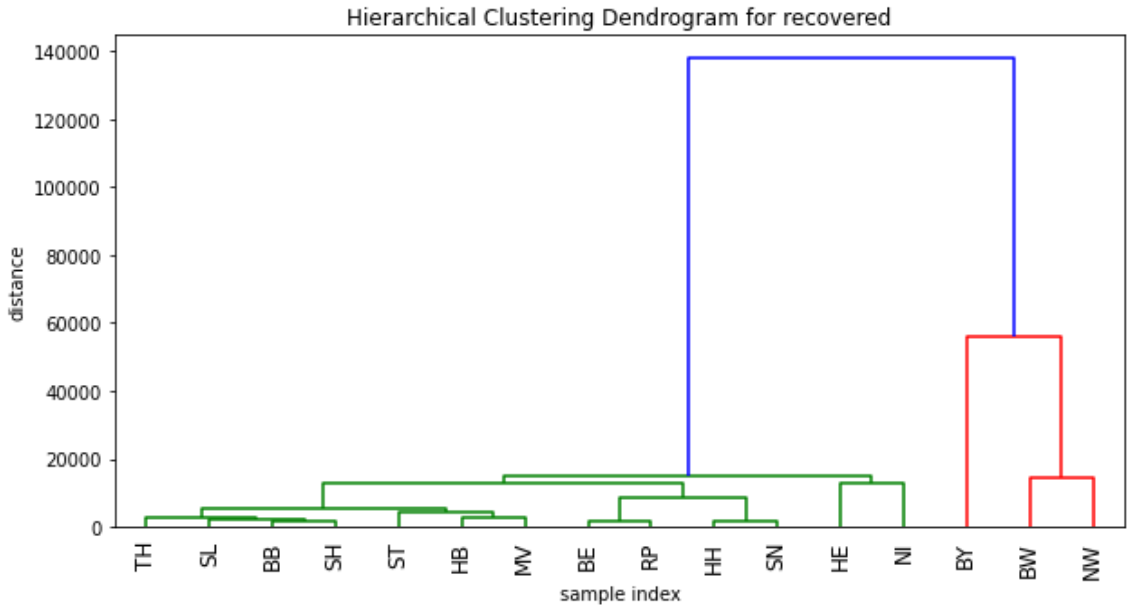


Figure 13.14: Hierarchical Clustering Dendogram to recovered for the period March-June 2020

The hiarchical clustering for confirmed cases, deaths and recoveries shows that there are two main clusters. One contains the states of Bayern(BY), Nordrein-Westfalen(NW) and Baden-Württemberg(BW) and the other the remaining states, which contains four sub-clusters (subcluster 1: Thüringen(TH), Saarland(SL), Brandenburg(BB), Schleswig-Holstein(SH); subcluster 2: Sachsen-Anhalt(ST), Bremen(HB), Meclenburg-Vorpommern(MV); subcluster 3: Berlin(BE), Rheinland-Pfalz(RP); subluster 4: Hessen(HE) and Niedersachsen(NI) . Also the cluster containing BY, NW and BW may be divided into 2 subclusters (subcluster 1: BY; subcluster2: BW and NW). This result corresponds to reality, because the largest number of infected, dead and recovered persons was found in the federal states BY, NW and BW, while other federal states had lower numbers.

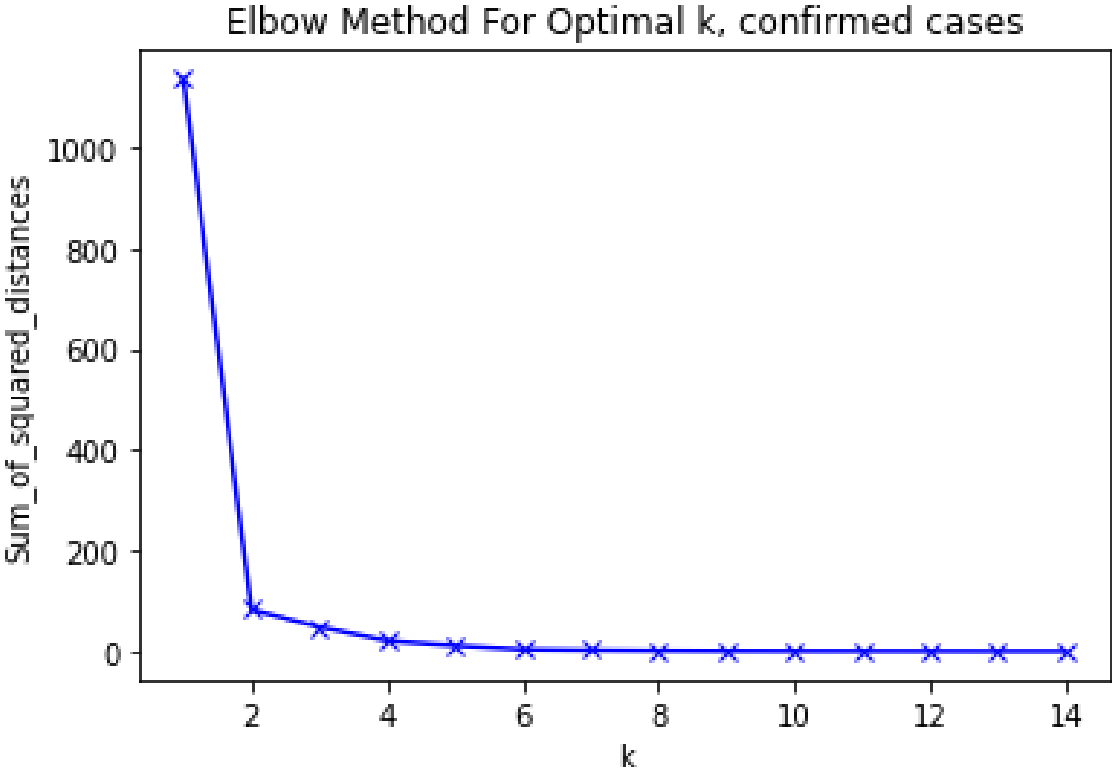


Figure 13.15: Elbow Method Plot for confirmed cases for the period March-June 2020

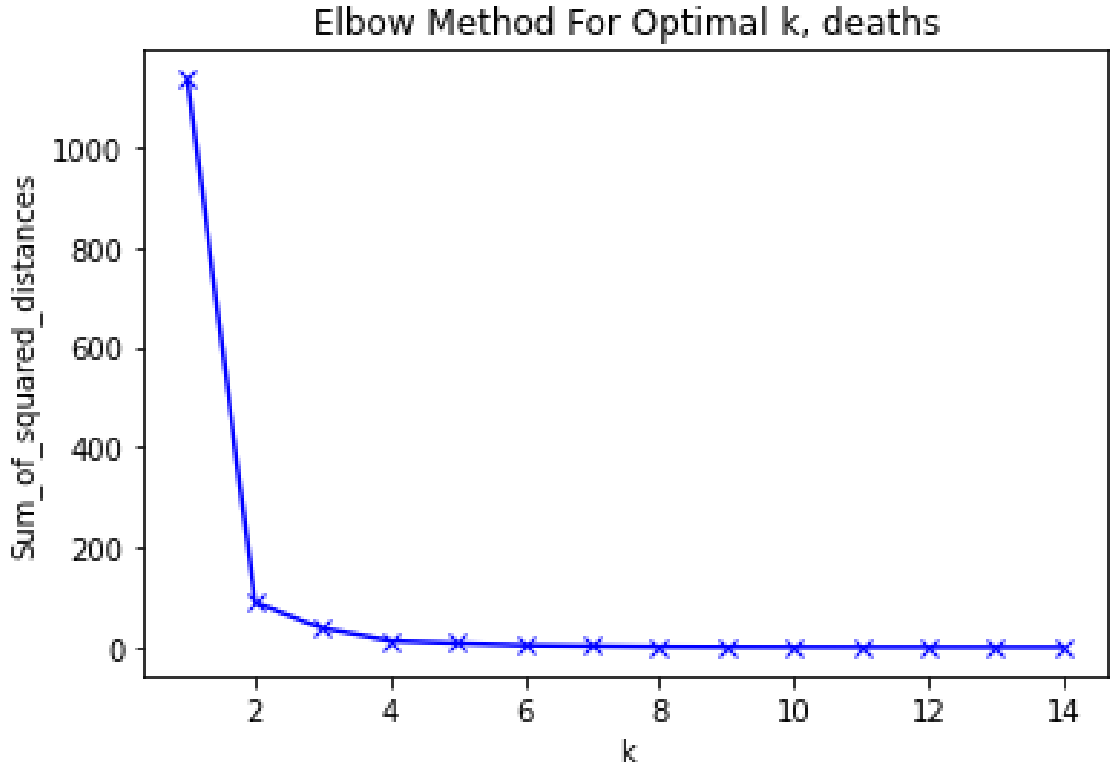


Figure 13.16: Elbow Method Plot for deaths for the period March-June 2020

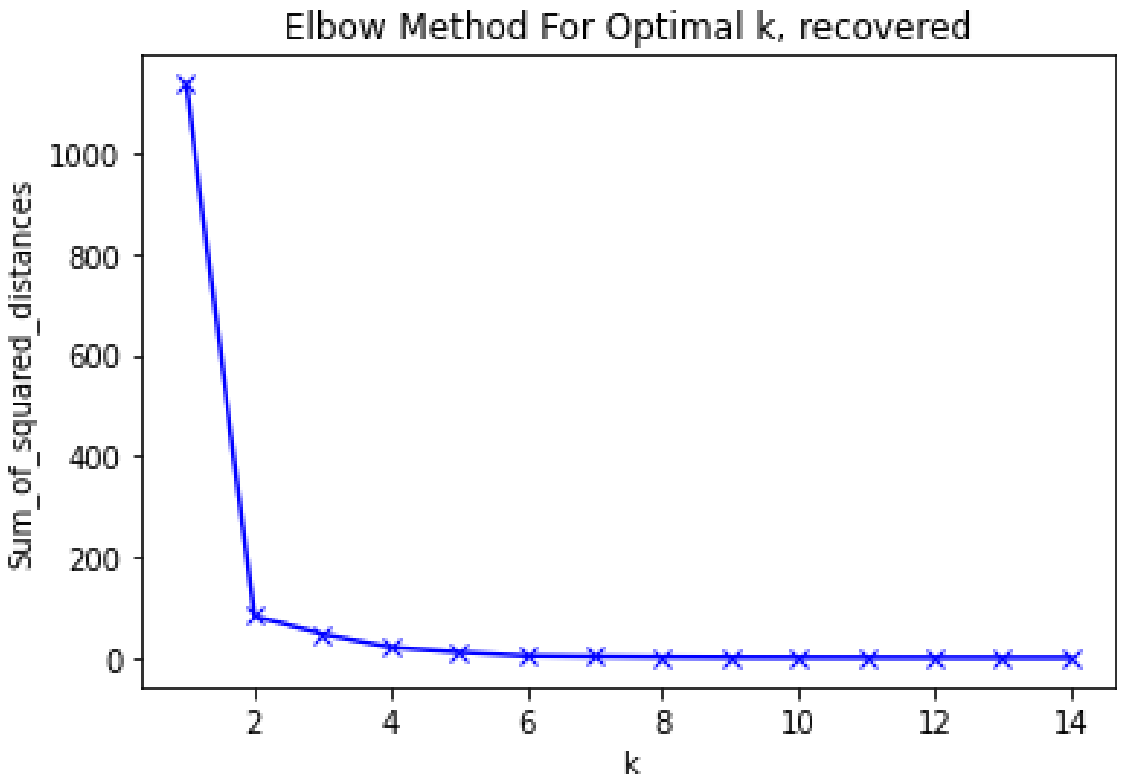


Figure 13.17: Elbow Method Plot for recovered for the period March-June 2020

The figures above (see 13.15,13.16 and 13.17) show the results of the elbow method for selectig the most optimal k-value, number of clusters. The best value seems to be 3 clusters.

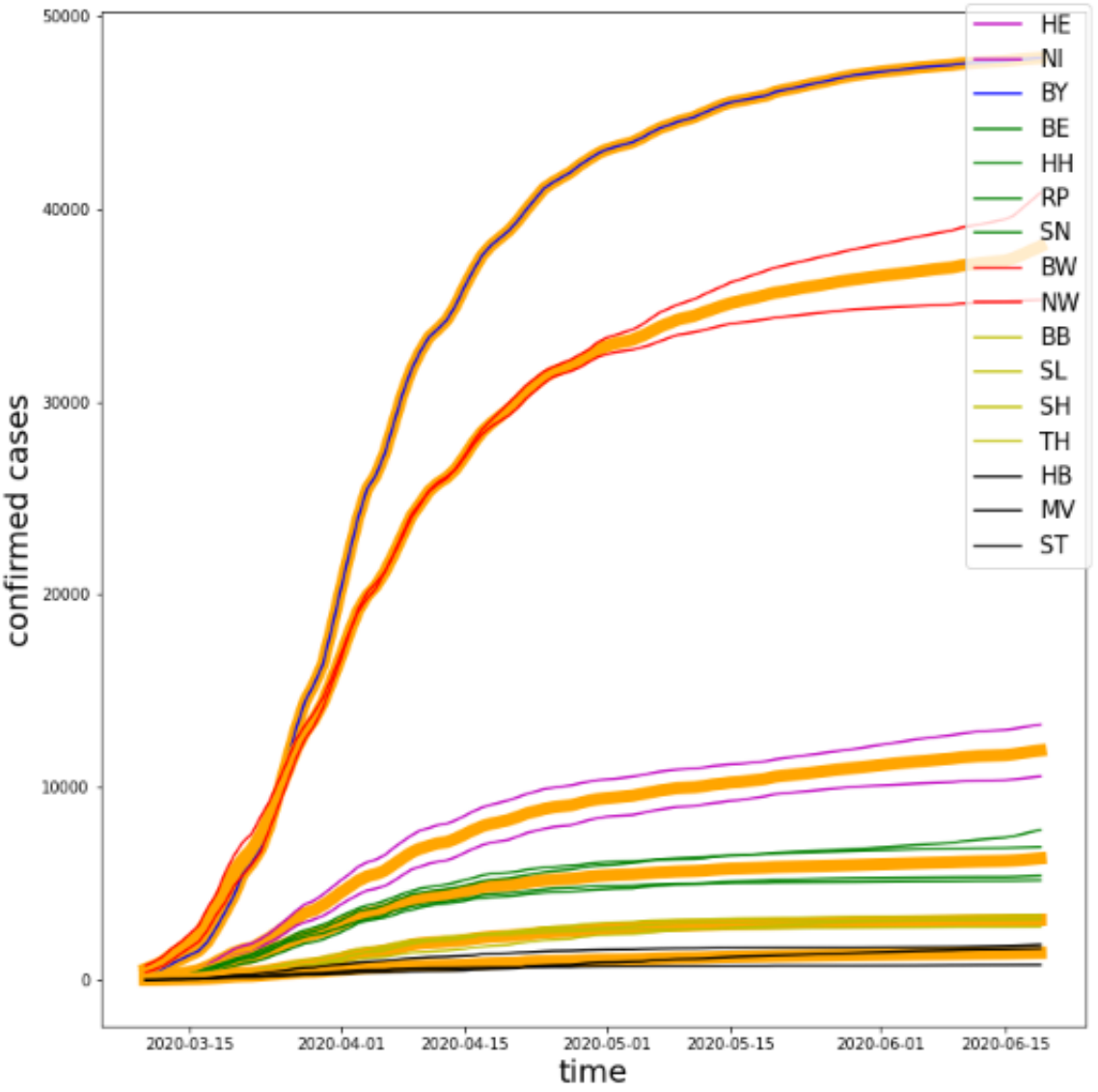


Figure 13.18: k-means-Clustering for confirmed cases for the period March-June 2020

The result of the k-means clustering show that the clusters could be identified. In order to see also subclusters, the k value was set to 6 for confirmed cases, 5 for the deaths and 6 for recovered cases (see 13.18, 13.19 and 13.20. For confimed cases and for recovered the same pattern of clusters could be identified (subcluster1: HE, NI, subcluster2: BY, subluster3: BE, HH, RP and SN; subcluster 4: BW and NW; subcluster 5: BB, SL, SH and TH; subcluster 6: HB, MV and ST) For the deaths could be 5 sublusters identified (see 13.19)

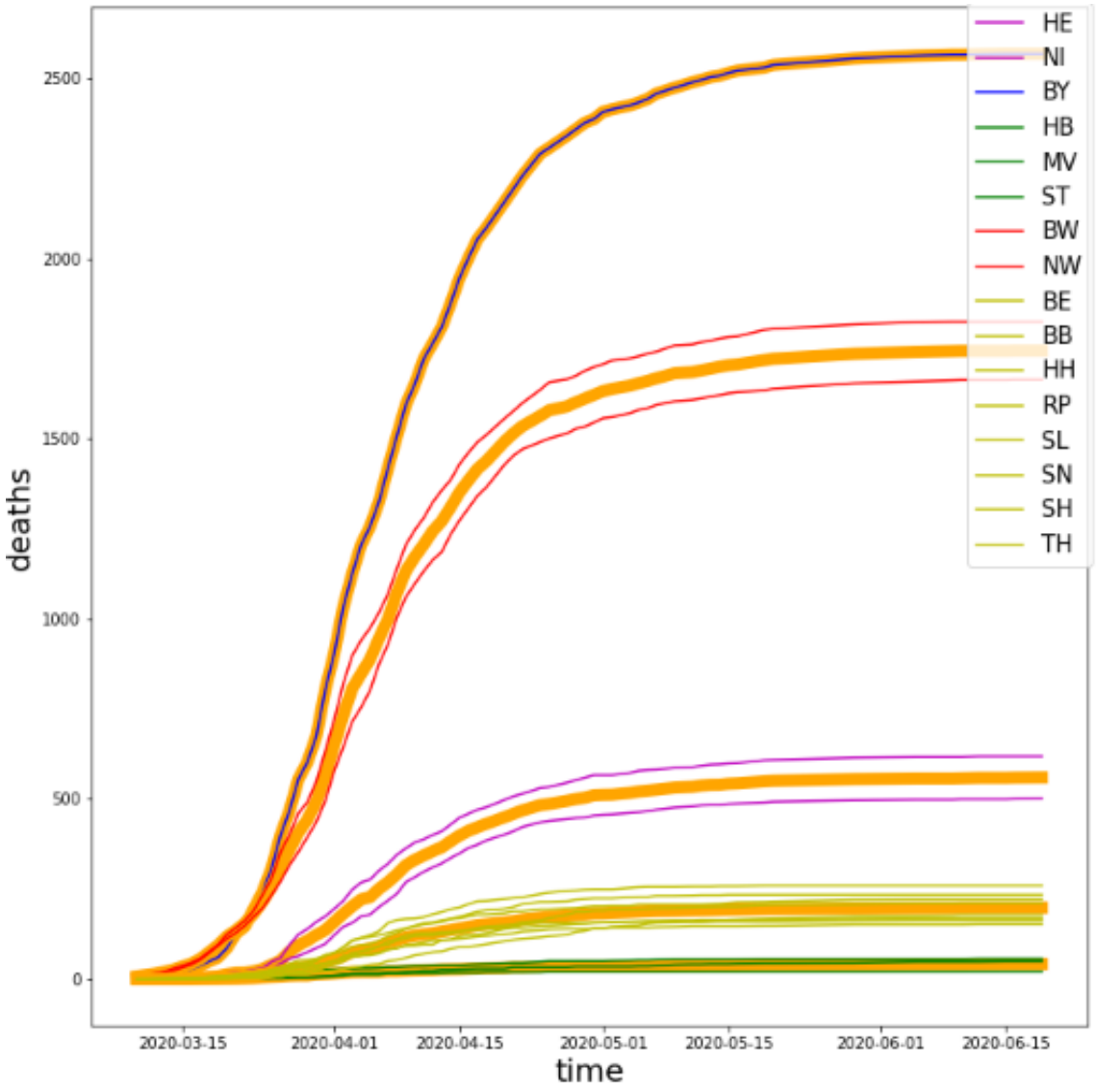


Figure 13.19: k-means Clustering for deaths for the period March-June 2020

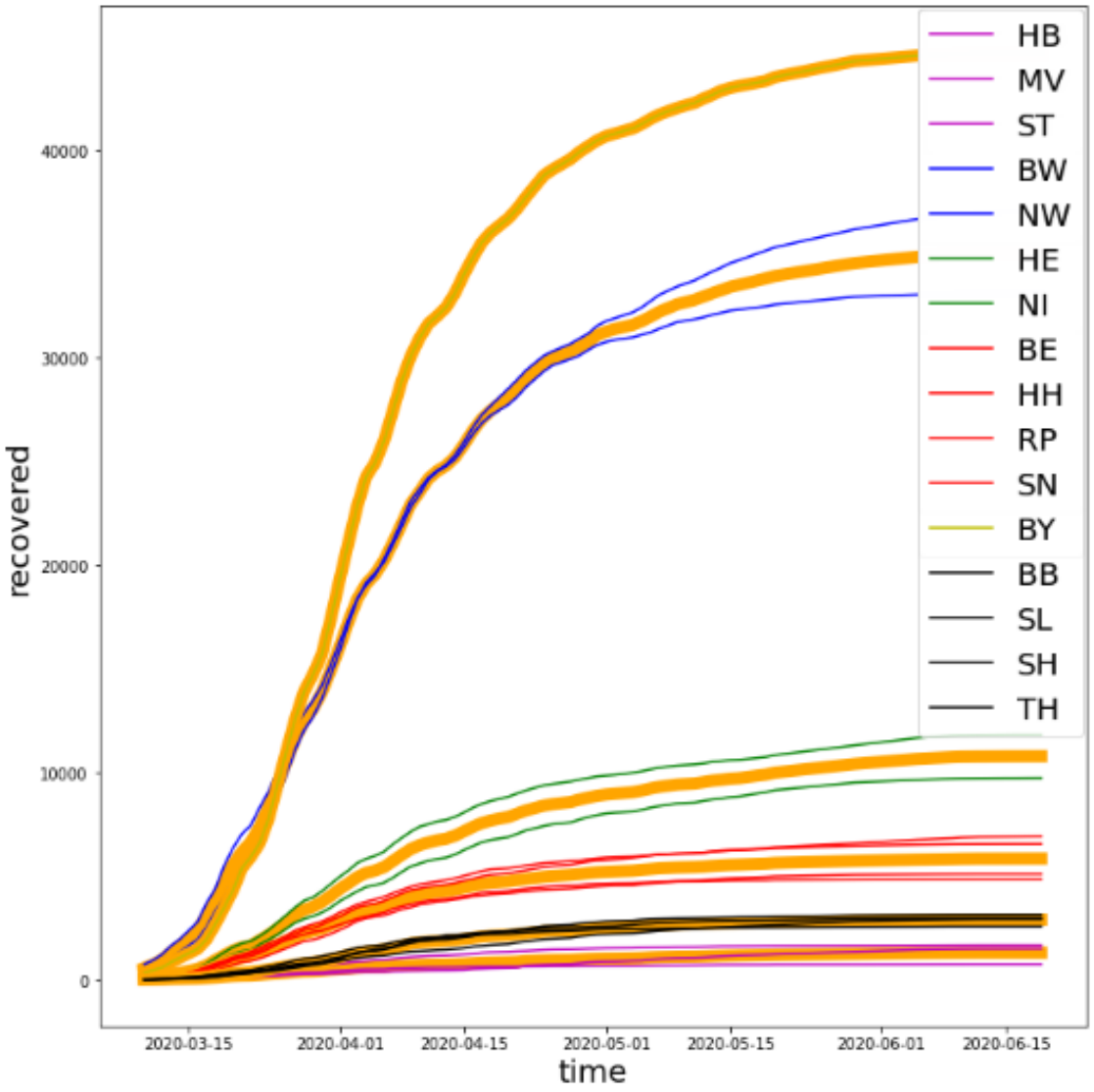


Figure 13.20: k-menas Clustering for to recovered for the period March-June 2020

14. Evaluation

The clustering using k-means method shows that there were possible to identify clusters that correspond to the reality.

14.1 Project Rating

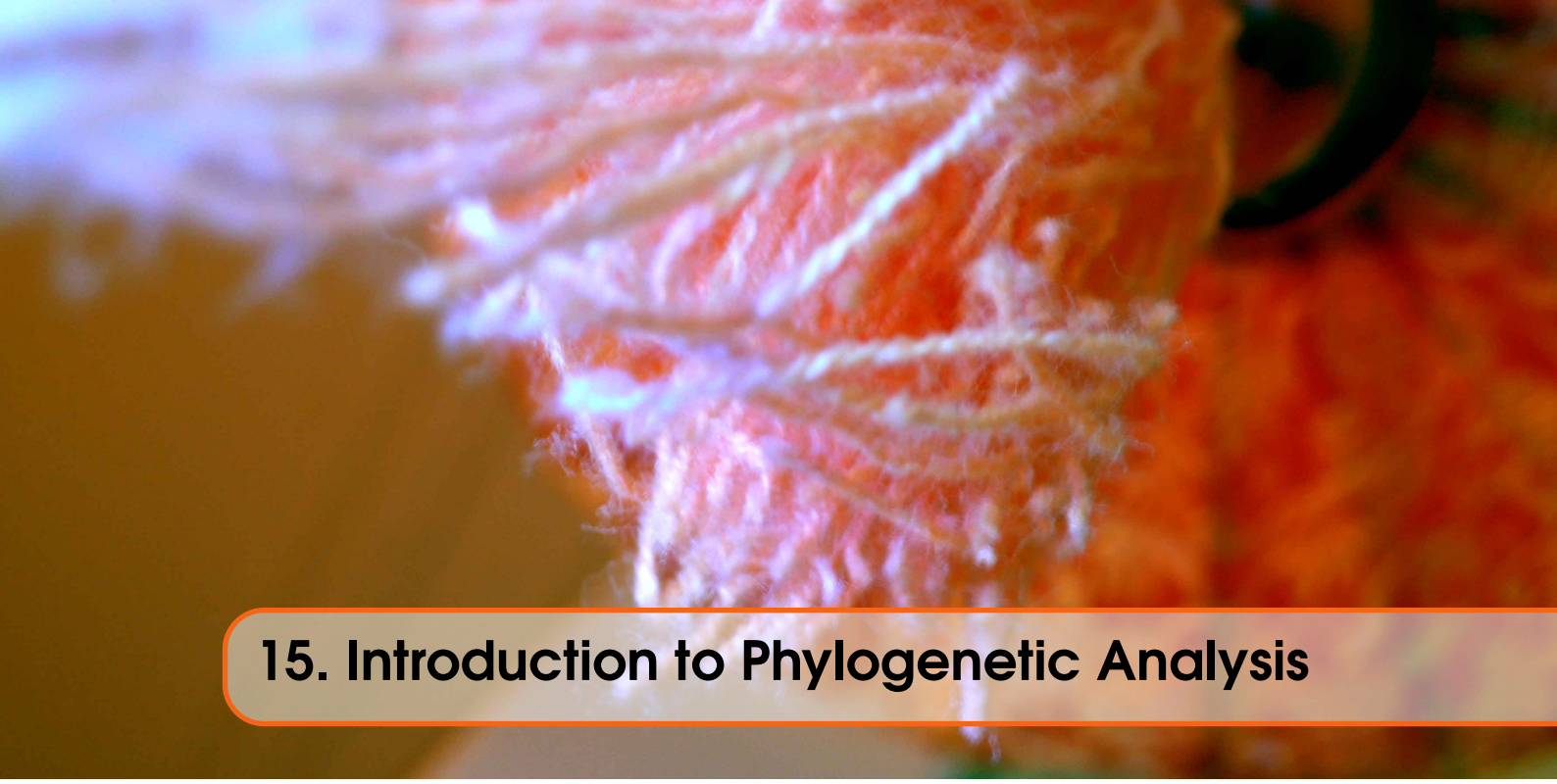
Appropriate visualization techniques for a comprehensive overview of the analysed data is an important part of being a data scientist. Finding the right graphics to emphasize certain parts of the analysis while also giving the reader a clear interpretation is not easy. Thus it is very interesting to test various methods for visualization on the recent covid-19 cases. It is apparent that it is possible to shift the point of interest for the same data simply by using a different graphic. In this regard this project was quite interesting. On the other hand doing time series prediction again was redundant. Visualization techniques can also be explored outside of the scope of time series. Overall we rate this project favourably.

14.2 Problems

Firstly in the visualization the data, we didn't face any problem as the data is directly extracted from the RKI site, and the obtaining the results with the plots was easy. Within the parts of time series prediction and curve plotting no difficulties were encountered. The preprocessing and finding of the appropriate data was easy. Furthermore, most data exploration approaches are well documented in python, such that redoing some of them for our goal of illustrating the corona cases for the federal states was not problematic either.

VII

Part 7



15. Introduction to Phylogenetic Analysis

15.1 Background

Phylogenetic analysis aims to reconstruct phylogenies both for a group of species and also for the individuals within those species. For pathogens, we are interested in the evolution of the diseases at the genetic level and want to derive information about their past and present diversity [38]. An important fact about the Coronaviridae family is that it's member tend to "jump" from one species to another. When the transmission occurs from a non-human host to a human host it is called zoonosis [11]. The determination of the most recent common ancestor of the human SARS-CoV-2 and the zoonotic transmission can provide important information about biological features, key mutations and properties of the virus. A detailed understanding of how an animal virus jumped species boundaries to infect humans will help in the prevention of future zoonotic events. [2]. Analyzing the diversions of Covid-19 sequences sampled from different human hosts all over the globe can lead to information about the order of transmission chains that have taken place.

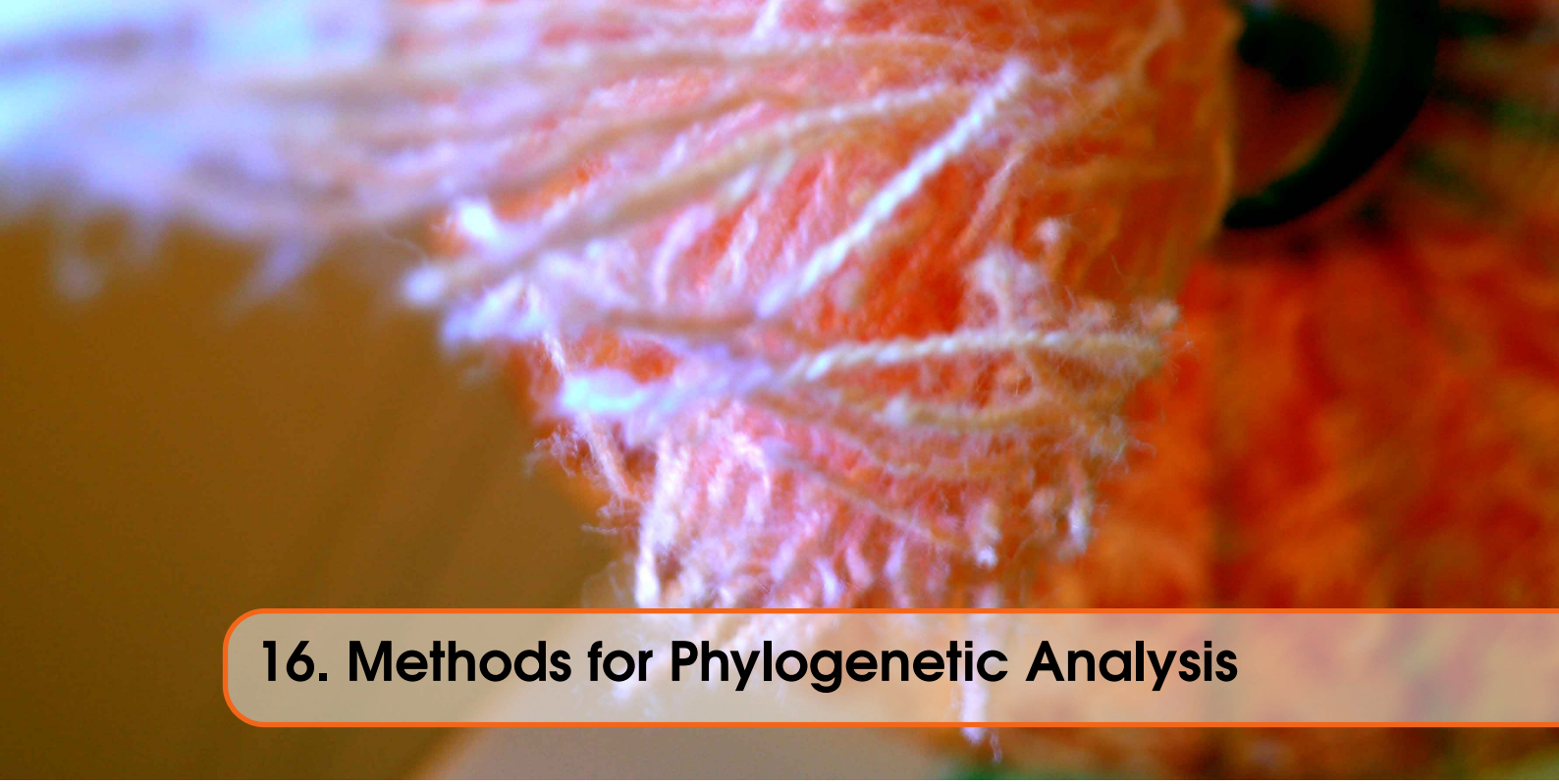
15.2 Goal of the project

We will perform two types of analysis: Origin detection and phylogeographic analysis. At first, the genetic sequence of SARS-CoV-2 is compared with six other virus sequences of the Coronaviridae family originated from different non-human hosts to gain information about the origin of the virus and the zoonosis. Regardless of that a second dataset was created containing 14 human SARS-CoV-2 sequences from different countries to identify possible pathways of the virus spread. The phylogenetic computations are performed by the hierarchical UPGMA and the TreeTime algorithm and the results are compared and analyzed afterwards. One top of that two non-hierarchical clustering methods (k-means and k-medoids) are performed to compare its results with the outcomes of the hierarchical clustering.

15.3 Outcomes

The Rhinolophus (horseshoe bat) was identified to have the most similar genomic sequence to the human SARS-CoV-2 genome with a sequence similarity of 94% and among the six Coronaviridae

family sequences of different hosts. For the phylogeographic analysis the UPGMA algorithm produced a very precise phylogenetic tree in where clear spatial patterns can be recognized. Especially, the possible infection pathways of the American samples can be explained. Furthermore, TreeTime offered the possibility to estimate the temporal transmission pattern of the virus spread with a high extent of the probability in addition to the spatial patterns. The application of non-hierarchical methods on the 14 human SARS-CoV-2 sequences resulted in the construction of five different. The biggest cluster contained mostly of samples from China and Europe. Nevertheless, the non-hierarchical approaches performed much worse than the hierarchical approaches.



16. Methods for Phylogenetic Analysis

16.1 Data

The first dataset was created to perform origin analysis (OA) and consists of one of the earliest sampled genetic sequence of SARS-CoV-2 from Wuhan and six other viruses of the Coronaviridae family in different hosts. The analysis is based on a Github repository of Simon Burgermeister [8] who originally downloaded the sequences from the NCBI Virus public library [15]. Accession numbers as same as host information can be found in table 16.1.
The second dataset was constructed to provide information about phylogeographics (PG). Here, 14 sequences are analyzed which were collected from humans all over the globe. Each continent is represented at least one often multiple sequences.To monitor the divergence of the virus at a specific time point, sequences were chosen that were all collected within March 2020 (expect one German sequence from February and another early Wuhan sequence from January). The respective metadata is listed in table 16.2.

Accession number	Host	Description
MN996528	H. Sapiens	Human SARS-CoV-2
NC_019843	H.Sapiens	Human MERS-CoV
JQ065048	Anatidae	Ducks, geese and swans
MG772934	Rhinolophus	Horseshoe bats
NC_034972	Apodemus chevrieri	Rodent
KX38909	Gallus gallus	Chicken
MT084071	Manis javanica	Pangolin

Table 16.1: Origin analysis (OA) dataset containing the human SARS-CoV-2 sequences and six other viruses of the Coronaviridae family collected in different hosts.

Accession number	Host	Location	Collection date
MT466071	Homo Sapiens	Uruguay	2020-03-13
MT499220	Homo sapiens	Tunisia	2020-03-31
MT447176	Homo sapiens	Thailand	2020-03-20
MT531537	Homo sapiens	Italy	2020-03-01
MT470177	Homo sapiens	France	2020-03-15
MT358639	Homo sapiens	Germany	2020-02-20
MT259229	Homo sapiens	China: Hubei, Wuhan	2020-01-26
MT350282	Homo sapiens	Brazil	2020-03-18
MT407659	Homo sapiens	China: Zhejiang	2020-03-24
MT451640	Homo sapiens	Australia	2020-03-25
MT434809	Homo sapiens	USA: New York	2020-03-19
MT434808	Homo sapiens	USA: New York	2020-03-19
MT633004	Homo sapiens	USA: Washington	2020-03-23
MT632947	Homo sapiens	USA: Washington	2020-03-23

Table 16.2: Phylogeographics (PG) dataset containing 14 SARS-CoV-2 sequences from different countries mostly collected in March 2020.

16.2 Methods

16.2.1 Hierarchical Approaches

Two common approaches towards constructing phylogenetic trees are the unweighted pair-group method with arthmetic mean (UPGMA) and TreeTime algorithm. The UPGMA algorithm is defined by its simplicity. Based on the assumption that the rate of mutation between different lineages stays constant over time, the so called *molecular clock hypothesis*, a phylogenetic tree with equidistant leaves to the root can be constructed. UPGMA starts with a matrix of pair wise distance objects. By iterating over the matrix, a cluster by entries i and j is defined by the smallest distance between both within the matrix. These entries are then connected through branch called the most recent common ancestor node. The distance of both to the connection node is defined as $D(i, j)/2$. Next a new cluster u is defined and its distance to each other cluster calculated as the average of the distances to all other clusters, hence the name. If no more entries are left the algorithm terminates. The TreeTimes approach belongs to the class of expectation maximization (EM) algorithms. EM algorithms use the divide and conquer method to divide a problem into simpler subproblems, thus decreasing computational time and increasing efficiency. The core idea of the TreeTime algorithm is a joint maximum likelihood assignment for each branch length. For each parameter e.g. leaf or node in the tree topology the assignment is calculated by finding the most likely value after summing or integrating over all unknown previous states. In practice the algorithm uses two steps: a post-order traversal and pre-order traversal. For the post-order traversal the maximum likelihood for node n to be at position t is calculated by taking the constraints of its children C and external constrains E e.g. collection data into account. Next the pre-order traversal follows where the branch length of each internal node is computed by finding the optimal value of time point t under the constraint of the parental node position.

The assumption of a constant mutation rate for the UPGMA algorithm is also its biggest disadvantage. Because each distance from the root to the leave within the tree is the same, UPGMA frequently generates wrong tree topologies. In reality it is very unlikely that different lineages are the same length in time and thus the hypothesis is violated. Many different factors can influence the mutation rate of an organism, bacterium or virus. In contrast the TreeTime algorithm does not rely necessarily on the *molecular clock hypothesis*, thus removing one of the biggest disadvantages.

Because TreeTime uses an expensive Baysesian approach the EM strategy is employed to strike a balance in computational efficiency. The heuristic nature of TreeTime can also be a disadvantage where a convergence to a wrong tree topology might be possible. UPGMA is robust in its implementation. The same distance matrix always results in the same tree topology. This might be an advantage over TreeTime where reproducibility is needed.

Implementation of UPGMA and TreeTime

The construction of the phylogenetic tree was performed by applying the preimplemented *upgma* function of the *Phylo* module contained in the *Biopython* package. It was executed with default parameters while the distance matrix was calculated beforehand by the *DistanceCalculator* function of the same package with the *identity* property. The multiple sequence alignment was generated by the NCBI BLAST implementation [17].
To use Nextrain Workflow with TimeTree for phylogeographic analysis, the datasets should be created according to the Nextstrain Fauna (database tool) requirements for sequence data and sample metadata. The datasets treated in this way were then processed with the Nextstrain Augur (analysis pipeline), performing multiple sequence alignment with MAFFT. A phylogeny with high probabilities is derived using TreeTime, which estimates a molecular clock. Given the derived molecular clock, TreeTime then creates a time-resolved phylogeny, derives sequence states at internal nodes and estimates the geographic migration history across the tree. These output data are exported as a JSON file that can be visualized interactively on the web with Nextstrain Auspice (vizualization platform).

16.2.2 Non-Hierarchical Approaches

While there is a broad range of clustering algorithms, the field of phylogenetic analysis is dominated by hierarchical clustering approaches like the presented UPGMA or TreeTime approach. But also other approaches like the commonly known k-means algorithm can be used to cluster sequences by similarity. Therefore we implemented two non-hierarchical clustering methods to compare its results with the ones of the hierarchical clustering.

k-means

K-means is one of the most popular clustering algorithms due to its simplicity. The data is separated by assigning each data point to cluster centers, such that the total squared error (SE) is minimized. The initial center points are set arbitrary and adjusted at each iteration step until the SE converges or after a fixed number of iterations as it can be seen in Figure 16.1. Finding the right k (number of clusters) can be challenging. Therefore k-means is performed for several k to find the one that minimizes the error.

k-medoids

The k-medoids method is a clustering methods that is related to the k-means algorithm. Both approaches break the datasets into groups. But while k-means tries in each step to minimize the total squared error, k-medoids minimizes the sum of dissimilarities between points belonging to the same cluster and its cluster center. For k-medoids the cluster centers are set to be one of the data points. This approach is less sensitive to outliers but at the cost of runtime, since every point is set to be a cluster center (medoid).

Workflow

Before we can apply both non-hierarchical clustering approaches we need to transform the sequences of the MSA such that they get comparable. Therefore we implemented the following workflow:

1) Factorize Bases:

Each sequence in the MSA contains elements of the DNA alphabet $\{A, C, T, G, N, -\}$, in

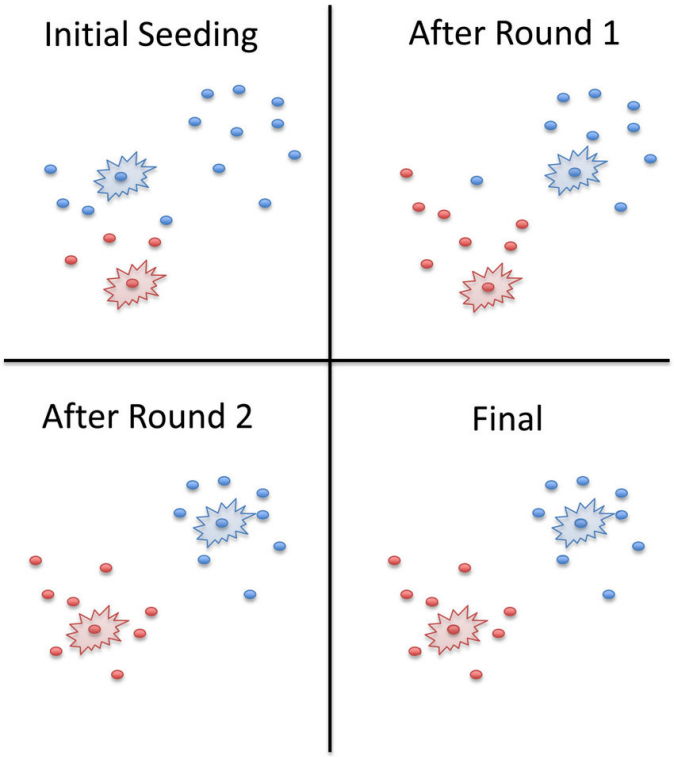


Figure 16.1: Schematic representation of the k-means algorithm for $k = 2$. The initial center points are set arbitrary and adjusted at each iteration step until the SE converges or after a fixed number of iterations. The picture is taken from [29].

which − represents gaps, N unknown bases, and the remaining four characters the DNA bases. All sequences are factorized, such that to each character of the alphabet a unique number is assigned.

2) Principal Component Analysis:

The factorized bases can be know seen as features. Consequently each of the 14 sequences consist of roughly 30.000 features/dimensions. Using PCA we can project our data onto two orthogonal axes, meaning that the feature space is reduced from roughly 30k to 2.

3) Clustering:

After performing PCA each data point is expressed by two principal components. The euclidean distance between the projected sequences is used to cluster them using k-means. For k-medoids we used the euclidean, manhattan and cosine distance.

For verification of the PCA approach, the k-Medoids method was also performed on a hamming distance matrix computed between all human sequences to compare the resulting clusters.

16.2.3 Phylodynamics

A field that gives wide knowledge of how various infected diseases are evolving over time is Phylodynamics. Phylodynamics completely depends on phylogenetic inference which acts as a tool to analyse mutations patterns which are the cause for probable spillover events. The mutations alter the phenotype which in turn infect different cell types. This allows the virus to develop different possible transmission routes. This in turn can be the driving factor of new epidemiological processes. Phylodynamics play an important role in filtering and getting the information from the genetic data. The interaction of the rapidly evolving pathogens completely depends on their ecological and evolutionary dynamics which usually happens at the same time scale. As the time

of sampling plays a crucial role in calibration of phylogenies it is an important information to incorporate in phylodynamics.

16.2.4 Phylogeography

Because infectious disease transmission is an inherently spatial process the geographic location of samples must be taken into account. The description of how the genetic signals are structured geographically within and among the species is called Phylogeography. Being the fastest growing field, it stands as a new technique in reconstructing the gene and genealogies as it detects the change or the mutation in the sequence of DNA from the individuals at the species range. [30] Visualizing these spatial relationships over geographic locations allows us to deduce how sub species are evolving, possible transmission routes and the origin place. It completely relies on ancestral lineages thus connecting the movement through space and movement through time. A few variety of applications that are entirely dependent on Phylogeography are earth historic events, distribution models and speciation processes. By linking the patterns of divergence in the population it identifies and tests the status of the diversification in an area. It gives us insight into whether and over which spatial and temporal scales the historical and the recurrent processes have shaped.

17. Analysing the Spread of SARS-CoV-2

17.1 Results

17.1.1 Origin Analysis

The resulting phylogenetic tree (Figure 17.1) of the OA dataset shows that our human SARS-CoV-2 sequence is most similar to the SARS-like coronavirus sequence of the Rhinolophus (horseshoe bat) with a similarity of 94%. The host with the next similar sequence is the Manis javanica (Pangolin) with a similarity of 0.89% between their genomes. The human MERS-Cov genome and the SARS-CoV-2 genome share only a sequence similarity of 0.74%.

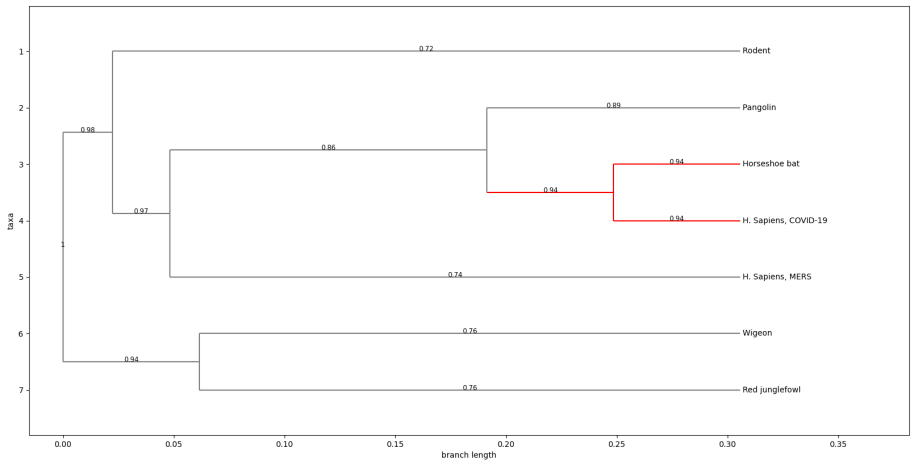


Figure 17.1: Phylogenetic tree of the origin detection analysis. The branch weights represent the sequence similarity in percent and the cluster containing the human SARS-Cov-2 seqence and the most similar sequence (horseshoe bat) is marked in red.

17.1.2 Phylodynamics and Phylogeographics

UPGMA

The UPGMA algorithm produced very plausible results (Figure 17.2). All three European samples (green) were assigned the same cluster with a maximum sequence dissimilarity of just 0.007%. The parent branch of the European cluster connects it with the South American cluster (pink) where the Brazilian and Uruguayan genomes showed a dissimilarity of only 0.003% to each other and a dissimilarity of 0.04 to the European cluster. Observing the results North American samples (blue) showed very interesting outcomes. The Washington samples were assigned together (dissimilarity 0.003%) as the as the New York samples (dissimilarity 0.007%). However, the West Coast samples were connected to the European and South American cluster, while the East Coast samples were identified to be most similar to the Wuhan sequence with a divergence of only 0.041%. The Thai sequence was clustered together with the Zhejiang (China) sequence (red) and the Tunisian (brown) and Australian (purple) sequence resulted the formed their own single sample cluster respectively.

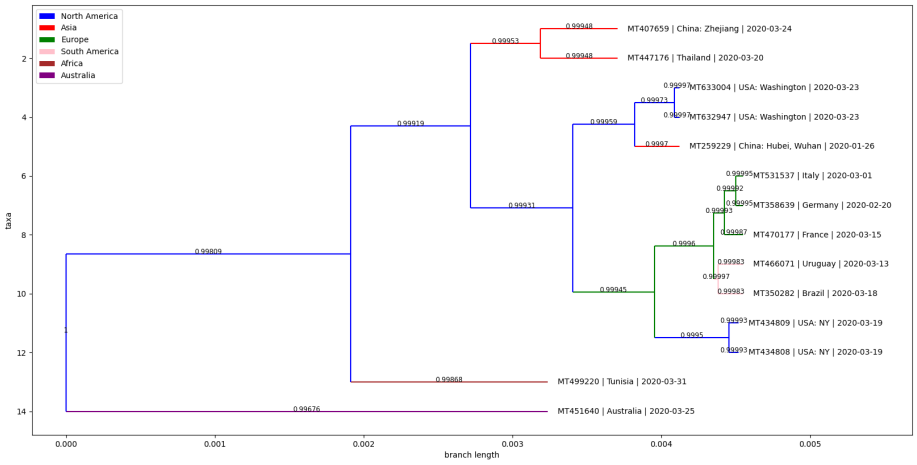


Figure 17.2: Phylogenetic tree of the phylodynamics and phylogeographics analysis. The branch weights represent the sequence similarity. The continents are marked by color as indicated in the legend while exact country and collected data is displayed in the sequence names.

17.1.3 TreeTime

A maximum likelihood phylogeography analysis of 14 ncov genomes was conducted. These sequences are sampled from distinct locations around the world. Consistent with other studies it could be found out that ncov moved from China, Wuhan Hubei to the other countries (see Figure 17.3). The ncov moved to Europe countries (France at first) via China Zhejiang and South America (Brazil) in parallel and the North American (USA) epidemic resulted from at least two introductions, one from one from Europe via Tunisia and another one from South America (Uruguay). It can be estimated that ncov moved from Europe (France) to Australia and from there to Germany. It can be estimated that the introduction in Europe (Italy, Germany and France) took place in late February 2020 and in South America one in February (Brazil, Uruguay via Thailand) and another one in North-America via Italy,Tunisia in 2020. The facts confirm results from conducted studies. Also using phylogeographic analysis it was possible to estimate 5 clades based on changes in nucleotides of the virus genome, which are associated with certain couple of countries where the virus was introduced (see Figure 17.4. The clade c1 (C3023T, C14394T, A23389G), c2(C227T), c4(G28867A, G28868A, G28869C) and c5 (C1045T, G25549T) is indicate to be distributed in European countries (Germany, France, Italy), Tunisia, Australia and North America (USA). The Clade 3 is found to be introduced South America(Uruguay and Brazil) and North America (USA) and Thailand.

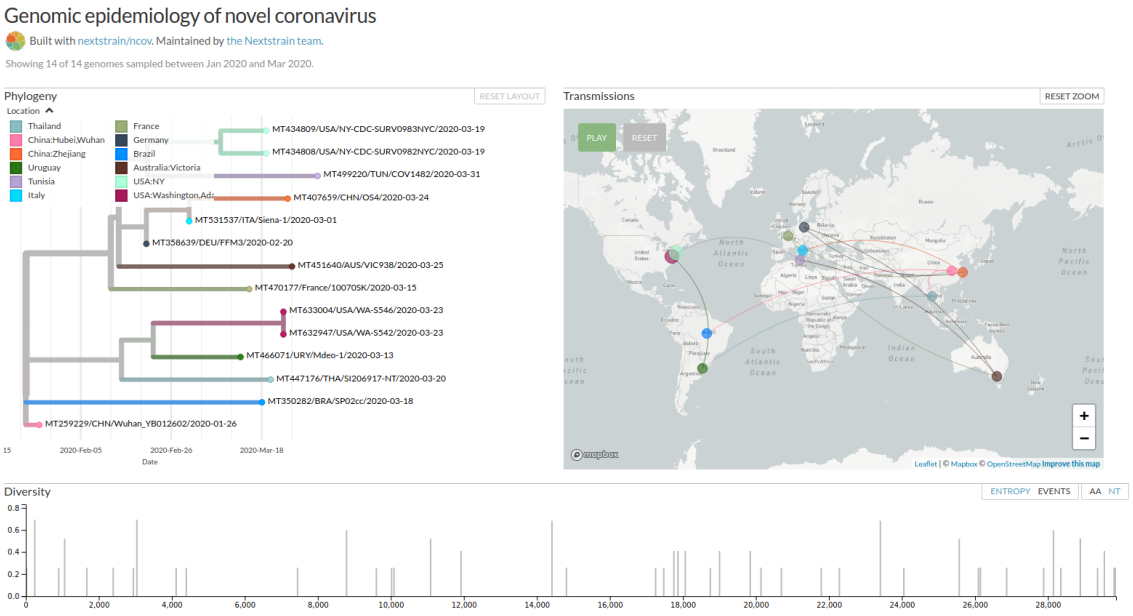


Figure 17.3: Phylogeographic analysis of 14 ncov genomes. Branch colors indicate the known country of sampling and the geographic migration history

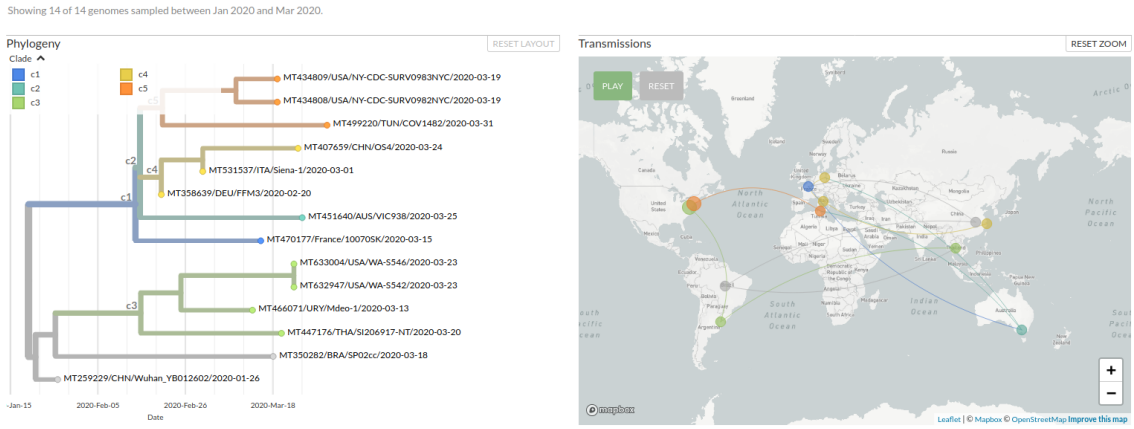


Figure 17.4: Phylogeographic analysis of 14 ncov genomes. Branch colors indicate the clades. The clade c1 nucleotide changes C3023T, C14394T, A23389G, c2 C227T, c4 G28867A, G28868A, G28869C and c5 C1045T, G25549T respectively, where at first is origin nucleotide, then the genomic site and the last letter is a nucleotide, to which the base mutated

Non-Hierarchical Approaches

After performing PCA on human sequences the resulting principal components explain 61% of the original variance. The minimal amount of variance is not easy cannot be generalized and out of the scope of this weeks project. Nevertheless in the book "Multivariate Data Analysis" Hair et al. [6] state that in many fields an explained variance of roughly 60% is sufficient. In Figure 17.5 the 2D projection of all 14 human sequences can be seen. On the left side of Figure 17.5 the

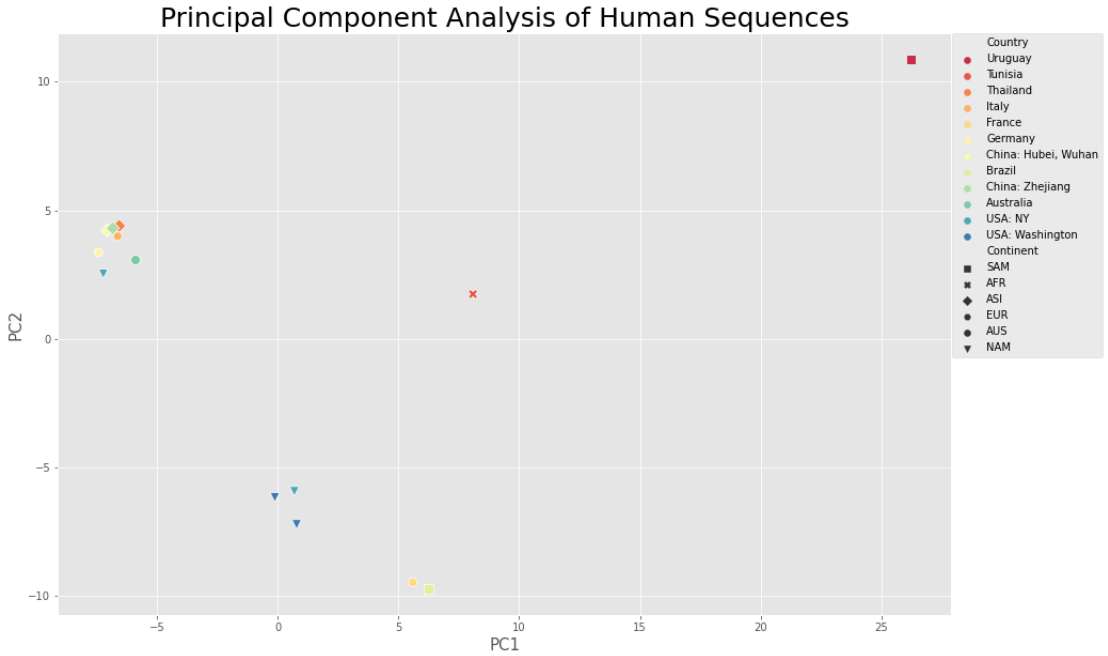


Figure 17.5: Principal Component Analysis graph of the human sequences. Each base of the roughly 30k sequences is seen as a feature and then projected on two dimensions with PCA. The shape of the markers represent the Continent the samples were taken from and the color the corresponding country. The procedure is explained in section 16.2.2.

biggest data point heap can be spotted that includes mostly sequences from China (Diamonds) and Europe (Circles), but also sample from the United States (Triangles). Another smaller group of three samples from the US can be spotted and the bottom left. Surprisingly also another heap was formed that includes the sample from French and the one from Brazil. The African and the Uruguayan sample did not group together and are generally rather separated from the other data points.

The principal components are then used to cluster the data as described in section 16.2.2. In Figure 17.6 we can see the resulting clusters using k-means and k-medoids with different distance metrics. Setting $k = 5$ led to minimal dissimilarities.

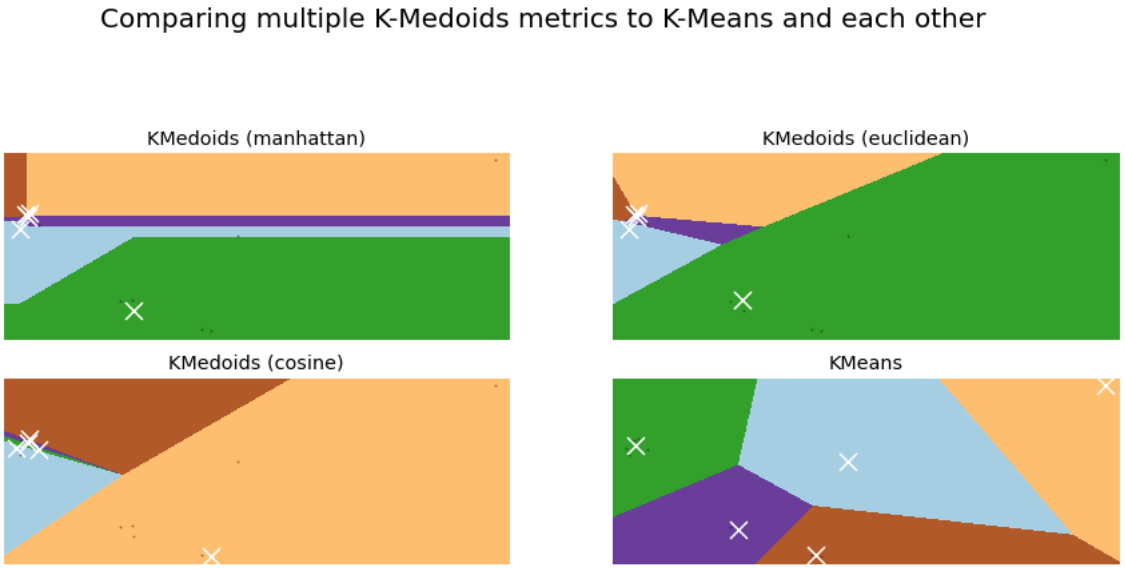


Figure 17.6: Comparison of the k-means and k-medoids clustering approaches. Each white cross represents a cluster center. The number of clusters is set to $k = 5$. The procedure is explained in section 16.2.2.

It can be seen that the four resulting clustering approaches cluster the data quite differently. While the three k-medoids approaches split the biggest data heap containing most samples from Europe and Asia into several clusters, the k-means approach assigns the same samples to a single cluster. Applying k-medoids on the hamming distance matrix with $k = 5$ produced the same clusters as it was the case with the PCA approach.

17.2 Discussion

17.2.1 Origin Analysis

As many early cases of COVID-19 were linked to the Huanan market in Wuhan [43], it is possible that an animal source was present at this location. Given the similarity of SARS-CoV-2 to bat SARS-CoV-like coronaviruses, it is likely that bats serve as reservoir hosts for its progenitor. Although the similarity of 94% to the coronavirus sequence hosted by the Rhinolophus, Andersen et al. [2] identified that its spike protein diverges in the receptor binding domain (RBD), which suggests that it may not bind efficiently to the human ACE2 receptor. Furthermore, it is assumed in this and other studies [20, 23] that an intermediate host was probably involved. Nevertheless, it has to be said that more different species of the same family can still have different mutation rates and therefore the molecular clock hypothesis has most likely been violated, which was not taken into account by UPGMA.

17.2.2 Phylodynamics and Phylogeographics

UPGMA

It was previously described that the mayor disadvantage of UPGMA is it's violation of the molecular clock hypothesis. However, in the PG dataset only human samples were examined, which could only have developed apart after the zoonosis had taken place in Wuhan at the end of 2019. It can

therefore be assumed that the molecular clock hypothesis is not, or at least only slightly, violated here.
It is not surprising that the New York samples clusters together with the European samples since new research suggests the Covid-19 outbreak in New York was mainly caused by travelers from Europe, not from Asia [13]. In contrast, the outbreak on the west coast of America is assumed to be mainly triggered by Chinese travelers, which is represented by cluster of the two samples from Washington combined with the Wuhan sample. overall, the UPGMA algorithm produce a very precise phylogenetic tree in which clear spatial patterns can be recognized.

TreeTime

Using the TreeTime based approach it was possible to describe general transmission patterns and estimate the emerging of ncov to distinct countries including the most probable introduction date. It was possible to find evidence that the outbreak begun in China, Wuhan Hubei and was widespread with movements to the countries in Europe (beginning with France, Italy, Germany), Australia, North Africa (Tunisia), North America (USA) and South America (Brazil, Uruquay). The phylogeographical analysis provides more precise view of how virus was spread in concordance with traveling, import, export and transport patterns from country to country. In addition, the phylogeographical approach provides virus genome changing information.

Non-Hierarchical Approaches

According to the results of the PCA, one can infer that the virus first spread within China starting in Wuhan and was then carried to Europe, since those samples are clustered together closely. The most distant point of this cluster is a sample from New York, that indicates the transmission from Europe/China to the US. This makes sense as medial reports about the coronavirus outbreak first started in china and then in Europe. As stated in the paper [42] from Worobey et al. the virus spread to the US mid of February when the pandemic already spread within Europe and China. The small cluster containing the French and Brazil samples does not make much sense for us. This cluster might be caused by the small sample size or the limits of non-hierarchical clustering methods in phylogenetic analysis. The clustering using the k-means approach clustered the samples according to our expectations, while k-medoids split the heap of Europe/Asia/US samples into several cluster, no matter which distance metric was used. This was surprising since papers using non-hierarchical approaches stated that k-medoids would be the better approach in phylogenetics than k-means, because k-medoids is less sensitive to outliers.

17.3 Conclusion

Even though the field of phylogenetic analysis is dominated by hierarchical clustering approaches the presented non-hierarchical methods can also provide interesting insights into the pathway of the virus. Regarding this project the hierarchical approaches produced a much more reasonable separation of the data, that allows to infer possible pathways that follow the trends described in other studies. However, non-hierarchical methods can be a better fit in the process of exploring bigger datasets for underlying structure that can afterwards be analyzed in detail since no multiple sequence alignment need to be calculated.

VIII

Part 8

18. Introduction

18.1 Background

In last week's project, we could already see how classical phylogenetic models can be used to make assumptions about the spread of the virus based on their sequence identity. Recall, those models require a multiple sequence alignment, which is not only computationally expensive but also varies due to different alignment costs. Another disadvantage lies in the evolutionary assumptions that those models make. For example, UPGMA (see 16.2) assumes a constant rate of evolution for all branches of the tree, which is unlikely, especially for sequences that are separated by larger evolutionary distances. Those problems could be obviated by a graphical representation of the virus genome that is based on the DNA sequences themselves, rather than the multiple sequence alignment.

18.2 Project Description

This week's project aims to implement one such graphical method and compare its results with the one of a classical UPGMA approach. Therefore a pyrimidine–purine graph is constructed as described by Liao et al. [22]. After building a classical phylogenetic tree (UPGMA), also a phylogenetic tree based on the method by Liao et al. is constructed. In the end, we compare both trees according to visual representation and a set of metrics. This includes weighted and unweighted Robinson-Foulds distance (symmetric differences) & Euclidean distances. The trees are constructed using roughly 50 coronavirus sequences from around the world. Both models will be used to elaborate phylogenetics as well as phylogeographics on a dataset containing human samples all over the globe. Additionally another data set was used containing samples from different hosts for which we perform the same analysis.

18.3 Outcomes

Upon visual inspection distinct differences between both methods were apparent. While the hypothesis of a spillover event from horseshoe bat to human is supported by both the method by

Liao et al. had difficulties performing on sequences which are very similar in alignment. All three distance metrics suggest significant deviations in tree topologies between both approaches.

19. Solution approach

19.1 Data

For the phylogenetic analysis, we created a dataset containing nine human samples from mainland China, in which we took three samples from January, February, and March, respectively. The remaining 39 human samples were all collected mid of April for different countries all over the globe. Another dataset was created as described in last weeks table 6.1 to perform origin analysis (OA) with samples from different hosts.

19.2 Methods

19.2.1 2D Graphical Representation of genomic Sequences

A pyrimidine-purine graph is created to analyze the phylogenetic relationships of genomes. To construct the curve, four vectors are defined to represent the purine bases Adenine and Guanine and the pyrimidine bases Thymine and Cytosine (Figure 19.1). The vectors hereby indicate the shift from the previous to the next data point of the curve when the respective base is present at the current position. The genomic sequences are converted to a set of data point bases on formula 19.1 and 19.2.

$$x_i = a_i \cdot m + g_i \cdot \sqrt{n} + c_i \cdot \sqrt{n} + t_i \cdot m \tag{19.1}$$

$$y_i = -a_i \cdot \sqrt{n} - g_i \cdot m + c_i \cdot m + t_i \cdot \sqrt{n} \tag{19.2}$$

The variable a_i, g_i, c_i, t_i correspond to the cumulative occurrence numbers of the bases until the current position while n and m real numbers representing the vector length that were set to $n = 0.5$ and $m = 0.75$ according to Yan et al.[44].

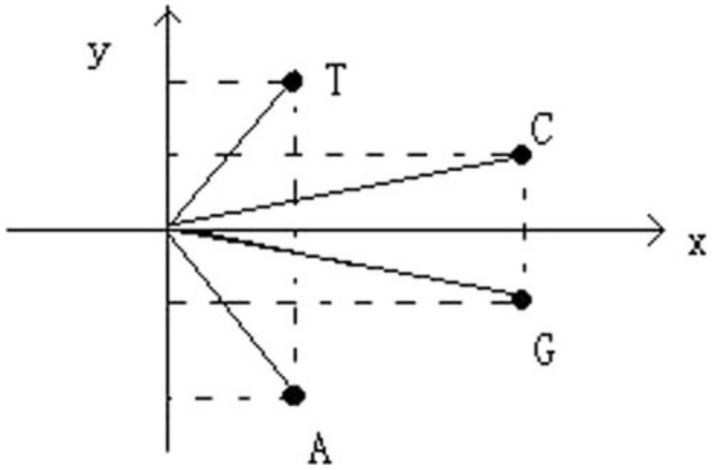


Figure 19.1: Pyrimidine-purine graph representing the four nucleotides of a DNA sequence. The vectors indicate the shift from the previous to the next data point of the curve when the respective base is present at the current position.

19.2.2 Building phylogenetic trees based on 2D curves

To create a tree using DNA sequences, we implied the 2D visualization method and created a distance matrix based on the 2D representation of sequences. The first step was to calculate the geometric center of the points using the coordinate data of DNA sequences extracted from the 2D representation:

$$x^0 = \frac{1}{N}\sum_{i=1}^{N} x_i, y^0 = \frac{1}{N}\sum_{i=1}^{N} y_i$$

The next step was to calculate the covariance matrix for the sequences:

$$\begin{cases} CM_{xx} = \frac{1}{N}\sum\limits_{i=1}^{N}(x_i - x^0)(x_i - x^0) \\ CM_{xy} = \frac{1}{N}\sum\limits_{i=1}^{N}(x_i - x^0)(y_i - y^0) = CM_{yx} \\ CM_{yy} = \frac{1}{N}\sum\limits_{i=1}^{N}(y_i - y^0)(y_i - y^0) \end{cases}$$

Then the eigenvectors of 2 eigenvalues each were to be determined from this covariance matrix using the following formula:

$$EV_k^i = (EV_{k,1}^i, EV_{k,2}^i)^T, i, j = 1, 2, ..., M, k = \lambda_1, \lambda_2$$

These are used to calculate arccosinus between sequences according to this formula:

$$\theta_{ij} = arccos(\frac{EV_k^i EV_k^j}{|EV_k^i||EV_k^j|}), i, j = 1, 2, ..., M, k = \lambda_1, \lambda_2$$

The angles are then summed for two sequences each.

$$\theta_{ij} = \theta_{ij}^{\lambda_1} + \theta_{ij}^{\lambda_2}, i, j = 1, 2, ..., M$$

To calculate the distance between the sequences the Euclidean distance between the sequences was also needed. This was calculated with this formula.

$$d_{ij} = \sqrt{(x_i^0 - x_j^0)^2 + (y_i^0 - y_j^0}, i, j = 1, 2, ..., M$$

The whole distance for the sequences will be calculated as :

$$D_{ij} = d_{ij} x \theta_{ij}, i, j = 1, 2, ..., M$$

Using this formula the distance matrix can be estimated and the phylogenetic tree generated.

19.2.3 Metrics to compare phylogenetic trees

Three different distance metrics were compared on both phylogenetic trees : Robinson foulds weighted, Euclidean and Robinson foulds unweighted distance. Robinson foulds takes the symmetric difference between two trees into account by adding all splits that are different between tree A and tree B.

20. Results & Discussion

Host Analysis

Beginning with a first visual inspection of the produced plots for the origin analysis, both methods produce a correct reproduction of the hypothesised spillover event from the horseshoe bat host to the human host for Covid-19 (Figure 20.1 and 20.2). Visualizing the sequences as 2D curves (Figure 20.3) shows a clear separation of two groups: Wigeon, Rodent and MERS on the one hand and COVID-19, Red junglefowl, Horeshoe bat and pangloin on the other. Because the tree topology is based on these curves it can easily be retraced. Interesting is the difference in grouping for MERS. While UPGMA puts MERS within reach of COVID-19, the method by Liao et al. classifies it near rodents. This is unsupported by the literature as MERS originated as a possible spillover event at the interface of human and camels. Thus UPGMA classification might be more appropriate.

Covid-19 Analysis

For the larger dataset of Covid-19 the visualization of sequence via the Liao et al. method fails (Figure 20.4). The sequence are too similar in alignment and thus no useful information can be extracted. Comparing the produced tree topology by UPGMA and 2D curve method distinct differences can be seen (Figure 20.5 and 20.6). UPGMA produces individual clusters of geographic similarity. Differences in time are also reproduced. On the other hand the Liao et al. method is also able to show geographic similarities and deviations in time. Both methods seem appropriate in their reproduction of spreading pathways for COVID-19.

The comparison of the two trees for different hosts and human respectively were done by different metrics which are mentioned in table 20.1. It is apparent that the low values for all three metrics for the COVID-19 tree suggest a lower deviation then for the host tree. The significance of these metric values is questionable as no reference value is present for comparison. Nevertheless all values suggest a difference in tree topologies for both approaches.

While the method by Liao et al. has clear advantages in computational time and efficiency it also has distinct disadvantages. If the sequences are very close in alignment as our testing of the COVID-19 dataset suggests the method falls apart. The differences are to marginal to produce meaningful results which in turn produces banal 2D curve visualisation. On the other hand, if the taxa are distinct enough the method performs quite well, showing an interesting way to visualise

distinct differences in sequences via a 2D curve visualisation. The produced tree topology while different from UPGMA seems correct.

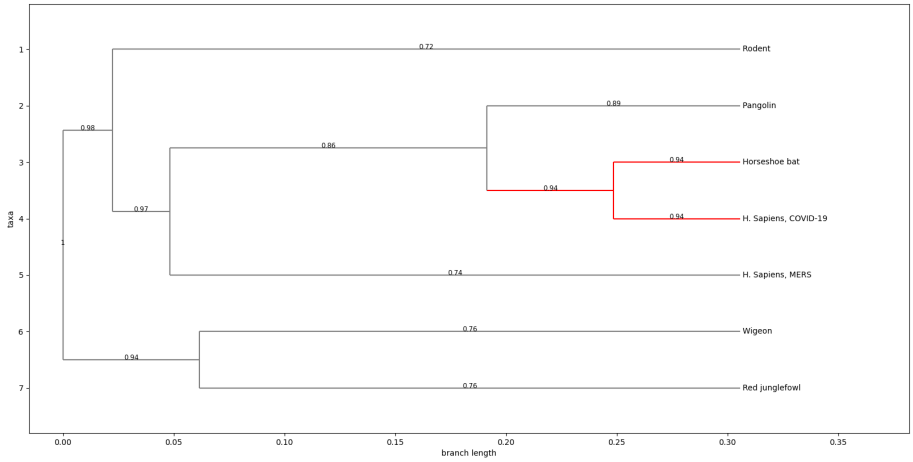


Figure 20.1: Phylogenetic tree of seven different sequences computed by the UPGMA method (see section 16.2). Visualized as red is the closet sequence of horseshoe bat and COVID-19 in humans, suggesting a possible spillover event.

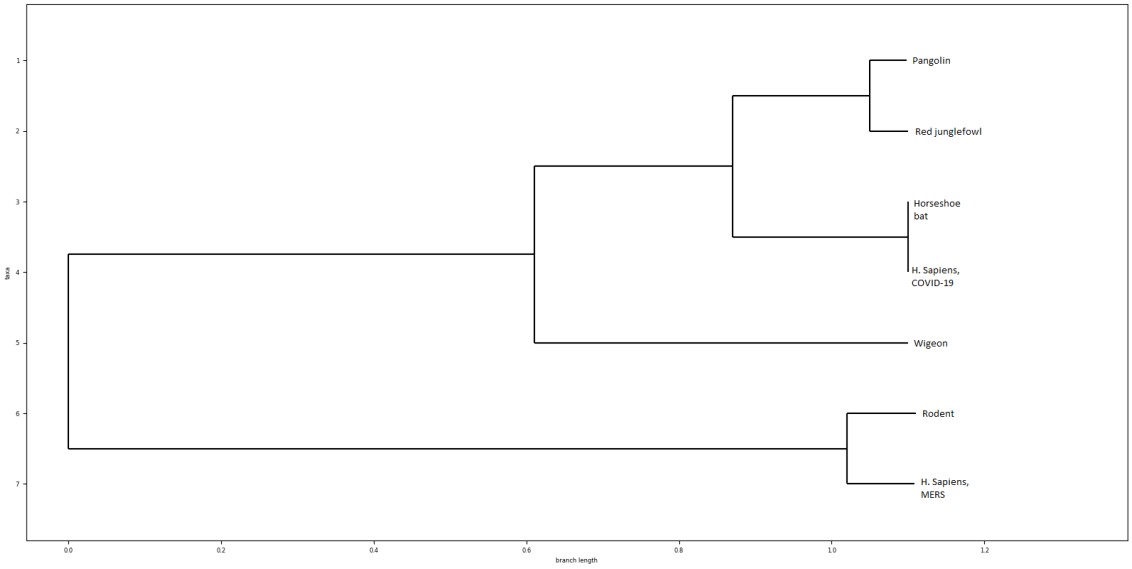


Figure 20.2: Phylogenetic tree of seven different sequences computed by the Liao et al. method (see section 19.2). The closest classification of horseshoe bat and COVID-19 in humans is reproduced, suggesting a possible spillover event too. The phylogenetic tree was created as described in section 17.1.2.

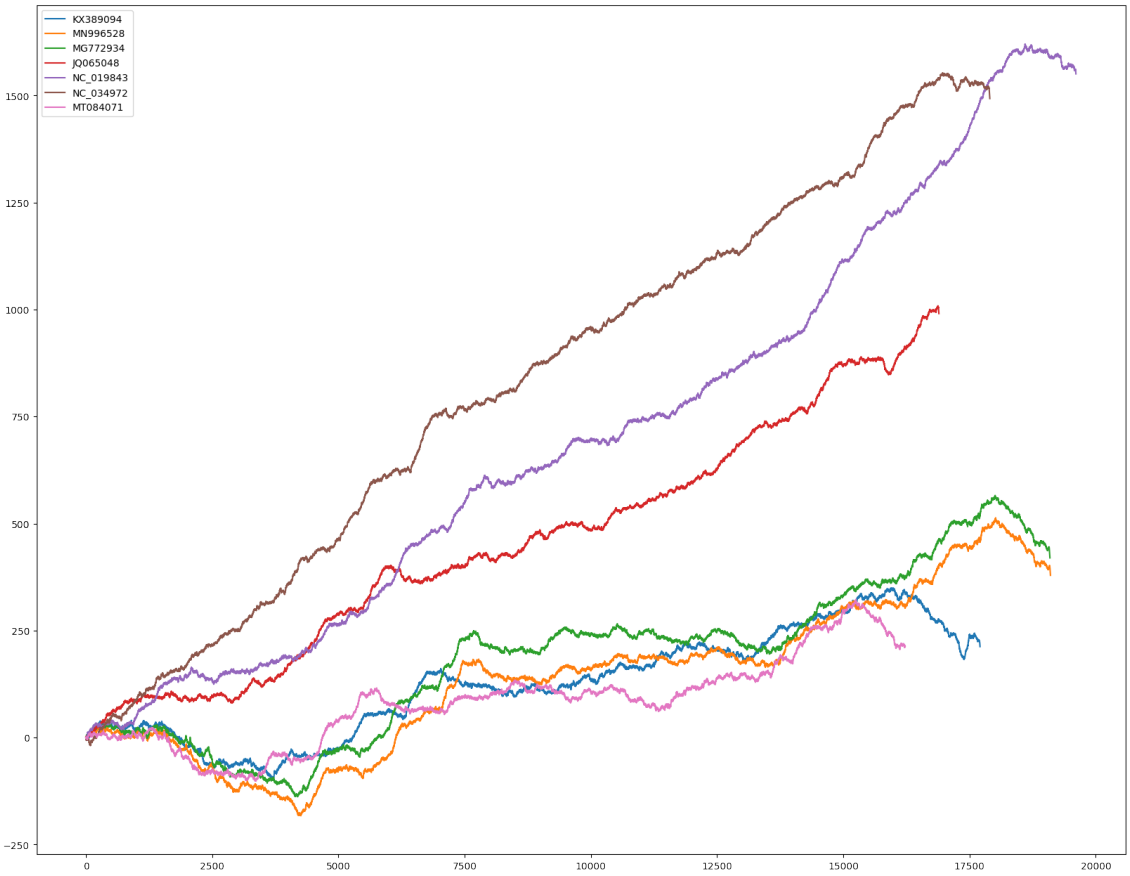


Figure 20.3: Pyrimidine-purine graph representing the shifts of individual bases for seven different sequences. While four curves a drawn together, three are deviating. The graphical representation was created as described in section 19.2. On the x-axis u can see the position in the samples and on the y-axis the corresponding score at this position.

Metrices	**phylogenetic tree different hosts**	**phylogenetic tree human**
Robinson-Fould weighted	3.475	0.075
Robinson-Fould unweighted	6	186
Euclidean metric	1.728	0.025

Table 20.1: Comparison of two trees with three different types of metric values.

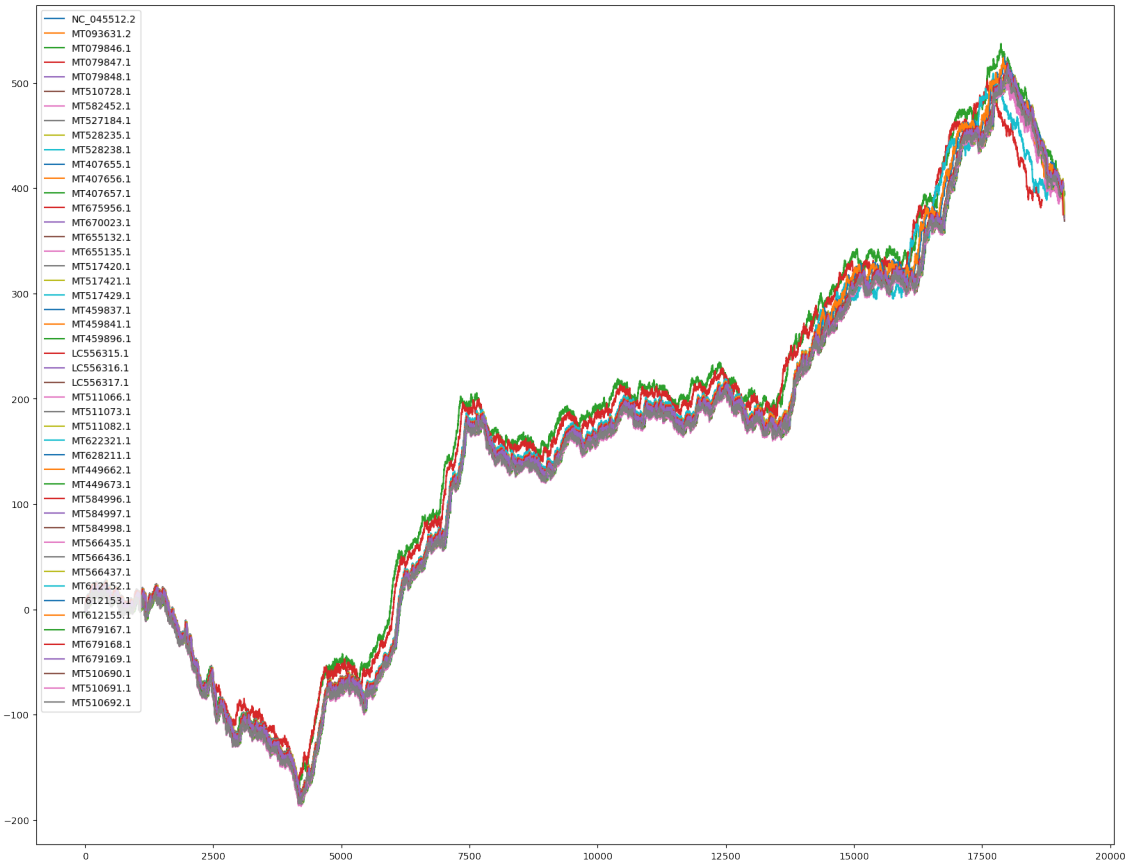


Figure 20.4: Pyrimidine-purine graph representing the shifts of individual bases for 40 different sequences of COVID-19. Because the alignment of sequences is too similar all curves are drawn together and thus it is difficult to extract useful information. The graphical representation was created as described in section 19.2. On the x-axis u can see the position in the samples and on the y-axis the corresponding score at this position.

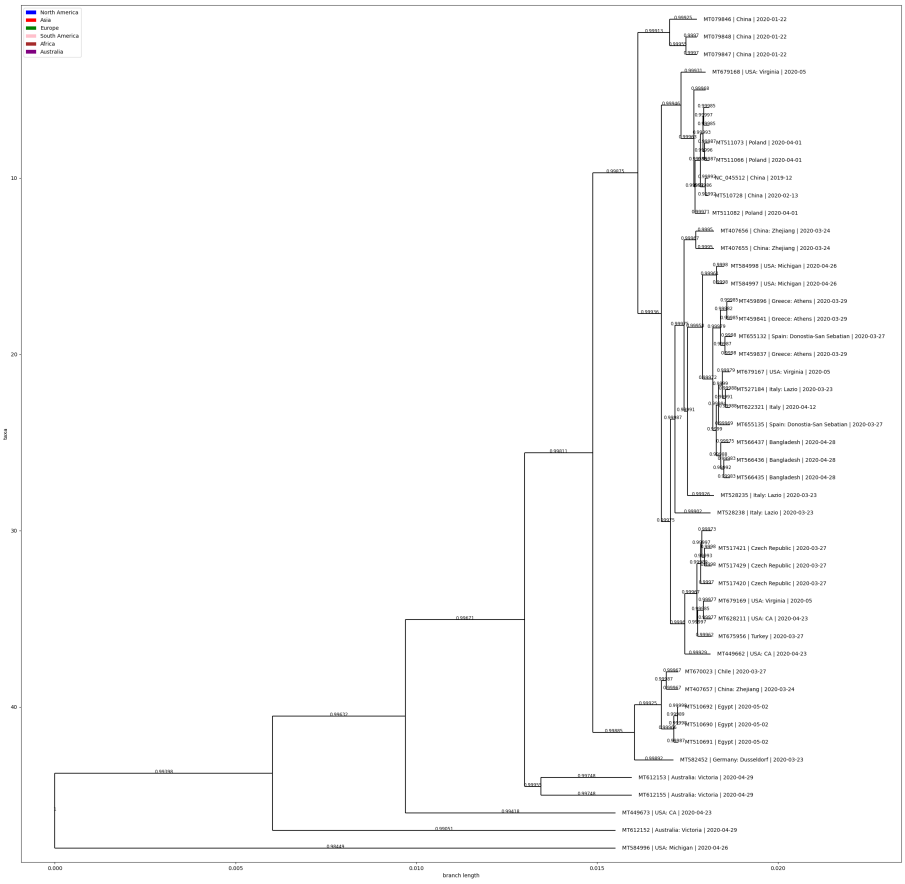


Figure 20.5: Phylogenetic tree of 40 different sequences from COVID-19 computed by the UPGMA method (see section 16.2). Distinct geographic clusters and time-based deviations can be seen.

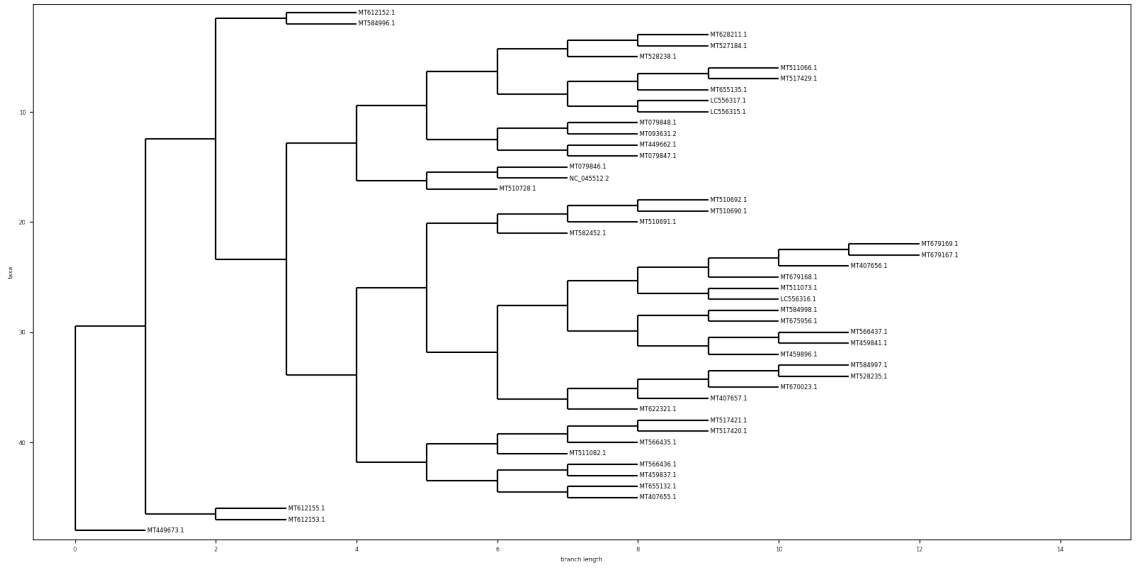


Figure 20.6: Phylogenetic tree of 40 different sequences from COVID-19 computed by the Liao et al. method (see section 19.2). Values for the distance on each branch length had to be omitted as the computed values were to small.

21. Project Evaluation

Overall this week's project was interesting to go through because it presented an unusual approach for a common problem. On top of that, the graphical approach was easy to implement due to the paper [22] that described the implementation clearly. The biggest benefit was the short running time, which allowed us to explore the effect of different parameter settings or using more diverged samples. Still, the graphical approach was not performing as well as the classical methods, thus we will most likely not use it in any future project, but when runtime becomes a major factor, due to huge sample sizes.

Bibliography

Articles

[1] Elissa M Abrams and Stanley J Szefler. "COVID-19 and the impact of social determinants of health". In: *The Lancet Respiratory Medicine* (2020) (cited on page 56).

[2] Kristian G Andersen et al. "The proximal origin of SARS-CoV-2". In: *Nature medicine* 26.4 (2020), pages 450–452 (cited on pages 27, 28, 117, 130).

[3] Ali Bazghandi. "Techniques, advantages and problems of agent based modeling for traffic simulation". In: *International Journal of Computer Science Issues (IJCSI)* 9.1 (2012), page 115 (cited on page 63).

[7] George EP Box. "All models are wrong, but some are useful". In: *Robustness in Statistics* 202 (1979), page 549 (cited on page 64).

[11] Kuldeep Dhama et al. "SARS-CoV-2: Jumping the species barrier, lessons from SARS and MERS, its zoonotic spillover, transmission to humans, preventive and control measures and recent developments to counter this pandemic virus". In: (2020) (cited on pages 27, 117).

[13] Ana S Gonzalez-Reiche et al. "Introductions and early spread of SARS-CoV-2 in the New York City area". In: *Science* (2020) (cited on page 131).

[15] Eneida L Hatcher et al. "Virus Variation Resource–improved response to emergent viral outbreaks". In: *Nucleic acids research* 45.D1 (2017), pages D482–D490 (cited on pages 27, 119).

[17] Mark Johnson et al. "NCBI BLAST: a better web interface". In: *Nucleic acids research* 36.suppl_2 (2008), W5–W9 (cited on page 121).

[20] Tommy Tsan-Yuk Lam et al. "Identifying SARS-CoV-2 related coronaviruses in Malayan pangolins". In: *Nature* (2020), pages 1–6 (cited on pages 28, 130).

[21] Thomas Lampert, Elena von der Lippe, and Stephan Müters. "Prevalence of smoking in the adult population of Germany". In: (2013) (cited on page 56).

[22] Bo Liao, Xuyu Xiang, and Wen Zhu. “Coronavirus phylogeny based on 2D graphical representation of DNA sequence”. In: *Journal of computational chemistry* 27.11 (2006), pages 1196–1202 (cited on pages 135, 147).

[23] Zhixin Liu et al. “Composition and divergence of coronavirus spike proteins and host ACE2 receptors predict potential intermediate hosts of SARS-CoV-2”. In: *Journal of medical virology* 92.6 (2020), pages 595–601 (cited on pages 28, 130).

[24] Eric Lofgren et al. “Influenza seasonality: underlying causes and modeling theories”. In: *Journal of virology* 81.11 (2007), pages 5429–5436 (cited on page 92).

[25] Lars Lorch et al. “A spatiotemporal epidemic model to quantify the effects of contact tracing, testing, and containment”. In: *arXiv preprint arXiv:2004.07641* (2020) (cited on page 64).

[26] Fábio Madeira et al. “The EMBL-EBI search and sequence analysis tools APIs in 2019”. In: *Nucleic acids research* 47.W1 (2019), W636–W641 (cited on page 27).

[28] World Health Organization et al. “Coronavirus disease 2019 (COVID-19): situation report, 73”. In: (2020) (cited on pages 67, 68).

[29] Justin Page et al. “BamBam: Genome sequence analysis tools for biologists”. In: *BMC research notes* 7 (Nov. 2014), page 829. DOI: 10.1186/1756-0500-7-829 (cited on page 122).

[35] Patricio Solıs and Hiram Carreño. “COVID-19 Fatality and Comorbidity Risk Factors among Confirmed Patients in Mexico”. In: *medRxiv* (2020) (cited on pages 68, 76).

[36] Thorsten Suess et al. “The role of facemasks and hand hygiene in the prevention of influenza transmission in households: results from a cluster randomised trial; Berlin, Germany, 2009-2011”. In: *BMC infectious diseases* 12.1 (2012), page 26 (cited on pages 59, 70).

[37] Sean J Taylor and Benjamin Letham. “Forecasting at scale”. In: *The American Statistician* 72.1 (2018), pages 37–45 (cited on page 83).

[39] Samantha M Tracht, Sara Y Del Valle, and James M Hyman. “Mathematical modeling of the effectiveness of facemasks in reducing the spread of novel influenza A (H1N1)”. In: *PloS one* 5.2 (2010) (cited on pages 59, 70).

[42] Michael Worobey et al. “The emergence of SARS-CoV-2 in Europe and the US”. In: *bioRxiv* (2020) (cited on page 131).

[43] Fan Wu et al. “A new coronavirus associated with human respiratory disease in China”. In: *Nature* 579.7798 (2020), pages 265–269 (cited on pages 28, 130).

[44] Stephen S-T Yau et al. “DNA sequence representation without degeneracy”. In: *Nucleic acids research* 31.12 (2003), pages 3078–3080 (cited on page 137).

[45] Jin-jin Zhang et al. “Clinical characteristics of 140 patients infected with SARS-CoV-2 in Wuhan, China”. In: *Allergy* (2020) (cited on page 55).

Books

[6] William C Black, Barry J Babin, Rolph E Anderson, et al. *Multivariate data analysis*. Volume 5. 3 (cited on page 129).

[18] McKendrick Kermack. *A contribution to the mathematical theory of epidemics*. Proc. Roy. Soc. A, Band 115, 1927 (cited on page 51).

[19] Gebhard Kirchgässner and Jürgen Wolters. *Introduction to modern time series analysis*. Springer Science & Business Media, 2007 (cited on page 81).

[38] Michel Tibayrenc. *Genetics and evolution of infectious diseases*. Elsevier, 2017 (cited on pages 27, 117).

Webpages

[4] *Bevölkerung - Zahl der männlichen Einwohner in Deutschland nach Altersgruppen am 31. Dezember 2018*. https://de.statista.com/statistik/daten/studie/1112607/umfrage/maennliche-bevoelkerung-in-deutschland-nach-altersgruppen/ (cited on page 56).

[5] *Bevölkerung - Zahl der weiblichen Einwohner in Deutschland nach Altersgruppen am 31. Dezember 2018*. https://de.statista.com/statistik/daten/studie/1112611/umfrage/weibliche-bevoelkerung-in-deutschland-nach-altersgruppen/ (cited on page 56).

[8] Simon Burgermeister. *covid_sequence*. https://github.com/simonjuleseric2/covid_sequence. 2020 (cited on pages 27, 119).

[9] *Coronavirus (COVID-19) death rate in Italy as of May 20, 2020, by age group*. https://www.statista.com/statistics/1106372/coronavirus-death-rate-by-age-group-italy/. Accessed: 2020-05-25 (cited on page 55).

[10] *Coronavirus Disease 2019 (COVID-19) Daily Situation Report of the Robert Koch Institute*. https://www.rki.de/DE/Content/InfAZ/N/Neuartiges_Coronavirus/Situationsberichte/2020-04-29-en.pdf?__blob=publicationFile (cited on page 56).

[12] *Diabetes Germany*. https://www.diabetes-news.de/nachrichten/diabetes-daten-2020-das-sind-die-zahlen. Accessed: 2020-05-24 (cited on page 68).

[14] *Hamburg in Zahlen*. https://www.hamburg.de/info/3277402/hamburg-in-zahlen/ (cited on page 70).

[16] *Hypertension RKI*. https://edoc.rki.de/handle/176904/2663. Accessed: 2020-05-24 (cited on page 68).

[27] *Obesity RKI*. https://www.rki.de/DE/Content/Gesundheitsmonitoring/Gesundheitsberichterstattung/GesundAZ/Content/H/Hypertonie/Inhalt/Blutdruck_DZHK.pdf?__blob=publicationFile. Accessed: 2020-05-24 (cited on page 68).

[30] *Phylogeography*. https://justinbagley.org/pages/phylogeog.html. Accessed: 2020-06-29 (cited on page 123).

[31] Robert-Koch-Institute. *COVID-19 case count in Germany state-by-state, over time*. https://github.com/jgehrcke/covid-19-germany-gae. 2020 (cited on pages 82, 95, 103, 104).

[32] *SARS-CoV-2 Steckbrief zur Coronavirus-Krankheit-2019 (COVID-19)*. https://www.rki.de/DE/Content/InfAZ/N/Neuartiges_Coronavirus/Steckbrief.html (cited on pages 56, 57).

[33] Leonardo Setti et al. *Airborne Transmission Route of COVID-19: Why 2 Meters/6 Feet of Inter-Personal Distance Could Not Be Enough*. 2020 (cited on page 70).

[34] *SIR flow diagramm*. https://www.lewuathe.com/covid-19-dynamics-with-sir-model.html/. Accessed: 2020-05-24 (cited on page 53).

[40] *Verkaufsfläche im Einzelhandel je 1.000 Einwohner in Deutschland im Jahr 2014 nach Bundesländern*. https://www.handelsdaten.de/deutschsprachiger-einzelhandel/verkaufsflaeche-einzelhandel-je-1000-einwohner-deutschland (cited on page 69).

[41] *Worldometers Kernel Description*. https://www.worldometers.info/coronavirus/countries-where-coronavirus-has-spread/. Accessed: 2020-05-11 (cited on page 11).

Index