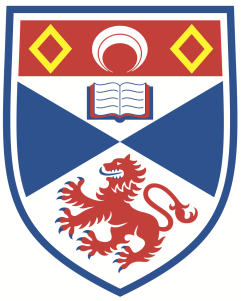
A Web Application for Compartmental Network Models Illustrated Using COVID-19

Leopold Jakob Pfeiffer

190026921



University of
St Andrews

This thesis is submitted in partial fulfilment for the degree of

MSc Data-Intensive Analysis

at the University of St Andrews, School of Computer Science

17 August 2021

Abstract

Epidemic models are a powerful tool to analyse the spread of a disease in a population and the effectiveness of different countermeasures. In this thesis, we develop a human contact network based on empirical mobility data and implement two compartmental epidemic models to run on these networks demonstrated using the COVID-19 pandemic. Given an appropriate parameterisation, the models are found to produce representative results when validated against empirical data from the pandemic in the United States.

In the second part of the thesis, a web application is developed that presents the modelling results in an aggregated form and provides visualisations to compare different models and configurations. This allows practitioners to use and work with epidemic models without the need to write any code. The extendable nature of the models and the application facilitates the integration of other diseases, which is demonstrated in this thesis by providing an alternative parameterisation for influenza.

Acknowledgements

First and foremost, I would like to thank my supervisor, Prof Simon Dobson, for his valuable advice and help in steering me in the right direction throughout this project.

I would also like to thank my partner and best friend, Kristina, for all her encouragement and emotional support during my studies and especially this dissertation.

Finally, I would like to thank my parents, without whom none of this would have been possible.

Declaration

I declare that the material submitted for assessment is my own work except where credit is explicitly given to others by citation or acknowledgement. This work was performed during the current academic year except where otherwise stated.

The main text of this project report is 14,701 words long, including project specification and plan.

In submitting this project report to the University of St Andrews, I give permission for it to be made available for use in accordance with the regulations of the University Library. I also give permission for the title and abstract to be published and for copies of the report to be made and supplied at cost to any bona fide library or research worker, and to be made available on the World Wide Web. I retain the copyright in this work.

17 August 2020

Leopold Jakob Pfeiffer

Contents

List of Figures

List of Tables

Listings

1 Introduction

1.1 Background

After first reports of cases of a "viral pneumonia" in Wuhan, China, on December 31, 2019 [WHO, 2020], it was not long until the world found itself deep in an unprecedented pandemic caused by the SARS-CoV-2 virus. Governments and health officials around the globe were faced with the challenge of implementing countermeasures against the spread of the disease under enormous uncertainty about the characteristics of COVID-19 and the effectiveness of interventions while also managing economic, social, and political costs. The events since the start of the pandemic have demonstrated the importance of objective means to guide decisions in the face of an epidemic and to predict the future spread of the disease with reasonable accuracy.

This is where epidemic modelling tries to step in to replicate, analyse, and help understand the characteristics of the epidemic processes as well as the effects of interventions. Many approaches combining epidemiology with mathematical modelling have been implemented for the task, and it remains up to the researcher to pick and choose from a myriad of techniques and frameworks. The use of these models usually requires some degree of technical expertise either of the mathematics involved or the programmatic implementation, which puts a barrier on the usability by practitioners.

1.2 Aims and objectives

This project makes an attempt at bringing together a realistic modelling approach and broad usability by developing a model of COVID-19 and making it available through an easily accessible web application that does not require programmatic implementation on the side of the user. Further, the model is readily extendable to other infectious diseases, which could then be integrated into the web app.

The primary objectives of the project are the following:

1. Develop a human contact network based on mobility data

2. Develop a compartmental model for COVID-19 using the epidemic modelling framework *epydemic*

3. Implement a web application allowing the use of the epidemic model in a user-friendly, yet sufficiently flexible manner

These primary goals are crucial for the success of the project and therefore take priority. Beyond this, we set the following secondary objectives for the project, which would benefit the project outcome but are less essential.

1. Survey of existing models of COVID-19

2. Validate the human contact network and compartmental model by comparing its results to empirical data

3. Extend the web application to other infectious diseases

1.3 Structure

We start off by conducting an extensive context survey of epidemic modelling with a focus on compartmental models, highlighting the mathematical background and existing implementations using a network approach. Further, we give an overview of models developed for COVID-19 and their different approaches, scopes, and goals. In Section 3, we develop the human contact networks based on mobility data, followed by the implementation of the epidemic models and the derivation of model parameters for COVID-19 (Section 4). We also discuss the adaptation of the model to other infectious diseases using influenza as an example. The simulation results are validated against empirical data from the United States in Section 5. The implementation of the web application is described in Section 6. In the last two sections, we evaluate the outcome and results of the project with regards to the objectives defined in the previous section and conclude the thesis.

2 Context Survey

Epidemiologists, mathematicians, and more recently computer scientists have developed a wide range of approaches, techniques, and frameworks for epidemic modelling. In the following sections, we give an overview of the most important literature and provide insights into the different approaches to epidemic modelling. Further, we give an overview of some of the models that have been developed specifically in response to the COVID-19 pandemic.

2.1 Epidemic modelling

In the broad field of epidemic modelling, we can identify two classes of approaches: compartmental models and agent-based models. *Compartmental models* split the population into separate compartments based on the stage of the disease. The flow between compartments is tracked to simulate the spread of the disease [Hethcote, 2000, pp. 600]. *Agent-based models* are a more recent development that approach epidemic modelling from the bottom up, that is with a subject level focus [Gordon, 2003, p. 398]. Individuals are assigned certain characteristics and their interactions with individuals in their environment are tracked. While this allows for significantly more choice about the parameters of the epidemic and could therefore create a more realistic model, the vast number of degrees of freedom and the inherent uncertainty remains a disadvantage. In this thesis, we solely focus on compartmental models and explore them in more detail in Section 2.1.1

Compartmental models can be described mathematically using differential equations, allowing certain behaviours of the disease spread in the population to be deduced. More recently, network models have been developed that are built on these mathematical models and attempt to provide more appropriate modelling of the population. Both approaches are described in the following sections.

2.1.1 Model description

Compartmental models get their name from their classification of the individuals of the population into different categories or *compartments* according to the current stage of the disease, for instance, whether an individual is susceptible or infected. Certain assumptions are then made about the flow between these compartments [Hethcote, 2000, pp. 600].

The most basic compartmental model is the SIS model, which splits the population into susceptible (S), i.e., those individuals in danger of contracting the disease, and infected (I), i.e. the population who currently carries the disease and can infect susceptible individuals. The model makes room for two state transitions. Firstly, susceptible individuals move to the infected state when they contract the disease, which requires the susceptible individual to be in contact with an infected individual. Secondly, an individual can move from infected to susceptible after they no longer carry the disease [Brauer and Castillo-Chavez, 2012, pp. 411].

SIS assumes that upon recovery from the disease, individuals move back into the susceptible compartment. This is different in the SIR model, which introduces the removed (R) compartment denoting those individuals who had the disease but are no longer infected and no longer susceptible - either because they are now immune or because they died from the disease [Brauer and Castillo-Chavez, 2012, p. 350].

The following paragraph establishes some notation for the SIR model. We denote the number of individuals in each compartment in a given time step t by S_t, I_t, and R_t respectively. Note that $S_t + I_t + R_t = N$ at all times, since the population size (N) is assumed to be constant over time. Since we are dealing with epidemics that occur within a limited and usually fairly short time frame (as opposed to an endemic), ignoring the regular birth-death process and assuming a constant population size is reasonable [Anderson and May, 1992, p. 123]. We can also express these quantities as fractions of the population size: $s_t = S_t/N$, $i_t = I_t/N$, and $r_t = R_t/N$. We further introduce two parameters, β and α. β is the transmission rate of the disease in contacts between infected and susceptible individuals. α denotes the probability of recovery, i.e. on average, an individual recovers α^{-1} time units after infection [Hethcote, 2000, pp. 602].

In mathematical terms, we can define the transitions between compartments as differential equations (omitting the time index for readability):

$$\dot{S} = -\beta IS/N \tag{2.1}$$

$$\dot{I} = \beta IS/N - \alpha I \tag{2.2}$$

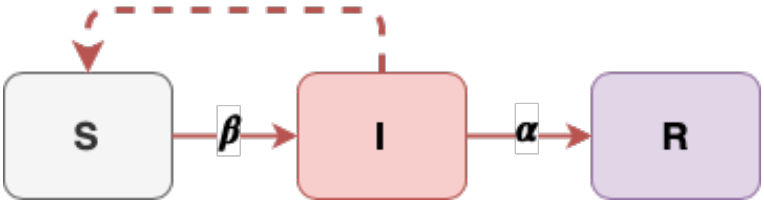
$$\dot{R} = \alpha I \tag{2.3}$$

Here, $\dot{S}$, $\dot{I}$, and $\dot{R}$ denote the partial time derivative of the respective compartment. It is also given that $S_0 \geq 0$, $I_0 \geq 0$, and $R_0 \geq 0$ [Hethcote, 2000, pp. 604].

Since $\beta I/N$ is the number of infectious contacts per susceptible individual, each time step the

total number of susceptibles decreases by this quantity multiplied by S, as it adds up for each susceptible (see Equation 2.1). Equation 2.3 gives the increase in recovered individuals per time step by multiplying the probability of recovery with the number of infected individuals. We can combine these quantities (see Equation 2.2) by subtracting the change in recovered individuals from the change in susceptibles. The transition parameters and their point of action are depicted in Figure 2.1.

Figure 2.1: SIR compartments and transition parameters. Dashed arrows represent disease transmission; solid arrows represent compartment transitions. The colour of the arrow corresponds to the colour of the compartment that is responsible for the transition.



In practice, one can extend SIS and SIR in whichever way is required by the disease at hand. A commonly used adaptation is given by SEIR, which adds the exposed (E) compartment containing all those individuals who have contracted the disease but are not infectious yet. The differential equations for the SEIR model are defined as follows [Brauer and Castillo-Chavez, 2012, p. 373]:

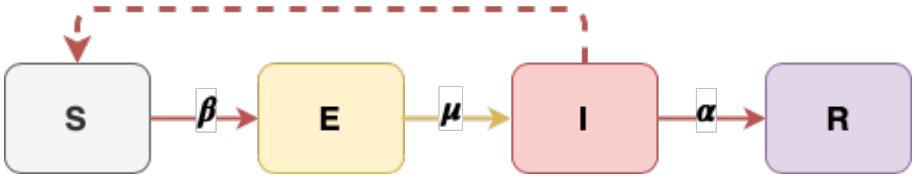
$$\dot{S} = -\beta IS/N \tag{2.4}$$
$$\dot{E} = \beta IS/N - \mu E \tag{2.5}$$
$$\dot{I} = \mu E - \alpha I \tag{2.6}$$
$$\dot{R} = \alpha I \tag{2.7}$$

Notice the additional parameter, μ, which is the probability that an exposed individual becomes infectious in a given time step. μ^{-1} is the expected duration of an individual being in the exposed compartment. The transition parameters of the SEIR model are depicted in Figure 2.2.

2.1.2 Basic reproduction number $\mathscr{R}_0$

When we defined the equations of SIR (2.1 - 2.3) we already encountered the parameters β and α as the rate of infection and rate of recovery, respectively. Beyond this, the parameters play

Figure 2.2: SEIR compartments and transition parameters. Dashed arrows represent disease transmission; solid arrows represent compartment transitions. The colour of the arrow corresponds to the colour of the compartment that is responsible for the transition.



an important role in epidemiological terms as they directly relate to the frequently quoted *basic reproduction number* $\mathscr{R}_0$. $\mathscr{R}_0$ is the expected number of secondary infections resulting from a single infected individual being introduced into a population of susceptibles [Hethcote, 2000, p. 603]. This can be expressed in terms of β and α:

$$\mathscr{R}_0 = \frac{\beta}{\alpha}. \tag{2.8}$$

$\mathscr{R}_0$ is central to determine the potential spread of a disease through a population. Specifically, one can show that $\mathscr{R}_0 = 1$ is the *epidemic threshold* and that an epidemic outbreak occurs whenever $\mathscr{R}_0 > 1$ while the spread dies out if $\mathscr{R}_0 \leq 1$ [Hethcote, 2000, p. 601]. To see why this is, think about β and α in terms of frequencies rather than probabilities: β^{-1} is the average number of time steps required for an infected individual to infect one susceptible individual; α^{-1} is the average number of time steps required from infection to recovery. Clearly, if (on average) it takes longer to recover than it takes an infected individual to infect someone susceptible, the disease will spread. That is, we get an epidemic if and only if $\beta^{-1} > \alpha^{-1} \Leftrightarrow \beta > \alpha \Leftrightarrow \frac{\beta}{\alpha} > 1 \Leftrightarrow \mathscr{R}_0 > 1$.

2.1.3 Mathematical solutions

In this section we summarise some important solutions to the SIR compartmental model that can be derived mathematically. Recall the differential equations (2.1 - 2.3) of the SIR model. We can divide these by the constant population N and get

$$\frac{ds}{dt} = -\beta is \tag{2.9}$$

$$\frac{di}{dt} = \beta is - \alpha i. \tag{2.10}$$

Equation 2.3 for the R compartment can be omitted since $r_t = 1 - s_t - i_t$ [Hethcote, 2000, p. 605]. The most important results of the mathematical model regarding the limits and the change in compartment sizes over time are summarised by Hethcote [2000, pp. 607]:

1. The fraction of the infected compartment always eventually decreases to zero as t goes to infinity:

$$\lim_{t \to \infty} i_t = 0. \tag{2.11}$$

2. If the basic reproduction number multiplied by the initial fraction of susceptibles is less than or equal one then the fraction of the infected compartment decreases to zero as t goes to infinity:

$$\mathscr{R}_0 s_0 \leq 1 \Rightarrow \frac{di_t}{dt} \leq 0. \tag{2.12}$$

   Note that the if the epidemic starts with very few initially infected individuals (as we would expect), $s_0 \approx 1$ so in this case we could replace the left side of the equation with $\mathscr{R}_0 \leq 1$. This is the same result as we saw at the end of Section 2.1.2.

3. On the other hand, if the basic reproduction number multiplied by the initial fraction of susceptibles is larger than one, then the fraction of infected individuals first increases up to a maximum i_{max} and then decreases to zero as t approaches infinity.

   Given $\mathscr{R}_0 s_0 > 1$ then

$$\frac{di_t}{dt} \begin{cases} > 0 & \text{if } t < t^* \\ = 0 & \text{if } t = t^* \\ < 0 & \text{if } t > t^* \end{cases} \tag{2.13}$$

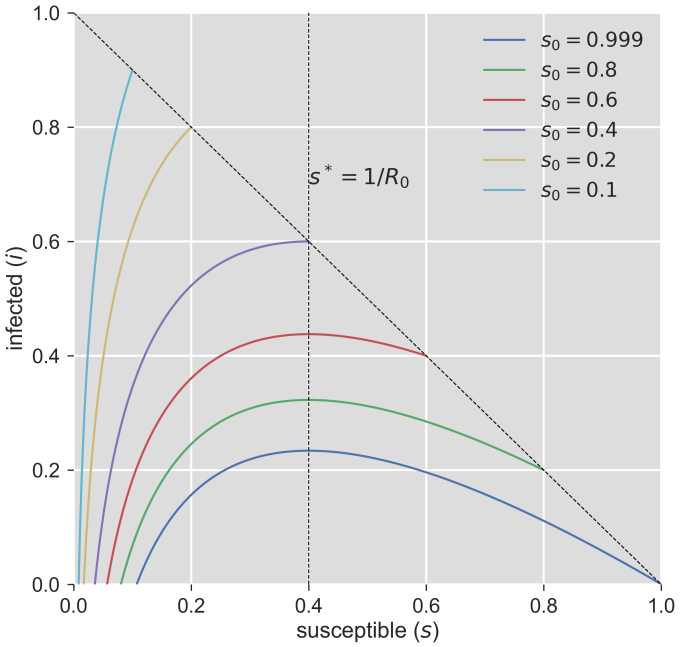
   with t^* being the time with the maximum infected fraction $i_{max} = i_0 + s_0 - \frac{1}{\mathscr{R}_0} - \frac{\ln(\mathscr{R}_0 s_0)}{\mathscr{R}_0}$.

4. The fraction of susceptibles at the end of the epidemic (s_∞) is given as the unique root in $(0, 1/\mathscr{R}_0)$ of the equation

$$i_0 + s_0 - s_\infty + \frac{\ln(s_\infty / s_0)}{\mathscr{R}_0} = 0. \tag{2.14}$$

The phase plane portrait depicted in Figure 2.3 visualises some of these results for an SIR model with $\beta = 0.25$ and $\alpha = 0.1$. Six trajectories are shown with different values for s_0 as well as the critical value of $s^* = 1/\mathscr{R}_0$. On each of the trajectories, we move from right to left as the disease progresses (since s always decreases). As is clearly visible, only on the trajectories where the most eastern point of s is larger than $1/\mathscr{R}_0$ do we see an increase in the fraction of infected individuals, that is have an epidemic (as described in Equation 2.13). In the Figure this is the case for trajectories with $s_0 = 0.999$, $s_0 = 0.8$, and $s_0 = 0.6$. For the other trajectories, $s_0 \leq s^*$, we find ourselves in the situation described by equation 2.12 without an epidemic outbreak.

Figure 2.3: Phase plane portrait for an SIR model with the parameters $\beta = 0.25$ and $\alpha = 0.1$ (adapted from Hethcote [2000, p. 605]).

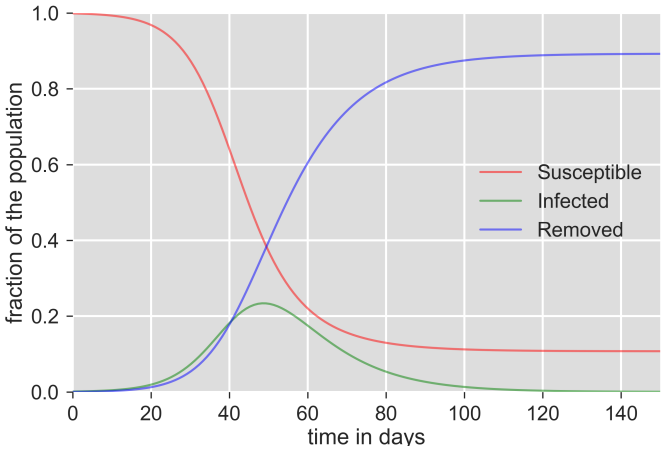


We can also plot the fraction of the three compartments in an imagined epidemic based on the mathematical model. The parameters are again set to $\beta = 0.25$ and $\alpha = 0.1$, which equates to setting $\mathscr{R}_0 = 2.5$. In period $t = 0$, we start with a very small fraction of infected individuals $i_0 = 0.001$. Figure 2.4 shows the fractions of the compartments over time.

Note that we can confirm equation 2.14 by plugging in the values as given by the model: $s_0 = 0.9999$, $i_0 = 0.0001$, $s_\infty \approx s_{150} = 0.1073$, and $\mathscr{R}_0 = 2.5$. This gives approximately zero as required.

We can also confirm Equation 2.13 by calculating i_{max}, which gives approximately 0.2335, as is

Figure 2.4: Progress of an imagined epidemic with the SIR model with the parameters $\beta = 0.25$, $\alpha = 0.1$ and an initially infected fraction of 0.001.



visible in the figure around time step 50.

2.1.4 Network approaches

The results from the last sections can already answer many questions about the epidemic processes. However, this approach fails to incorporate the characteristics of human interaction and instead assumes that everyone is equally likely to interact with everyone else - clearly a strong simplification. Network models try to address this by modelling populations as graphs. Conclusions about the epidemic on the network are then drawn either through theoretical analysis of the graph or by means of simulation. Before we look at both these approaches more closely, let us define the notation and terminology adapted from Boccaletti et al. [2006, pp. 180] and Pastor-Satorras et al. [2015, pp. 932], who provide detailed explanations of the topic.

A network or *graph* G consists of a set of *nodes* (or vertices) $V = \{n_1, n_2, ..., n_N\}$, which are connected by *edges* in the set $E = \{\{n_i, n_j\} \mid n_i, n_j \in V \text{ and } i \neq j\}$, so $G = (V, E)$. The edge connecting nodes i and j is denoted $e_{ij} = \{n_i, n_j\}$. We can distinguish between *undirected* graphs where an edge e_{ij} denotes a symmetrical connection from n_i to n_j and vice versa (for example, a network representing friendships), and *directed* graphs where e_{ij} denotes a connection only in the direction from n_i to n_j (for example, a network of websites with links from one website to another).

Each node n_i has a *degree* denoted k_{n_i} which is the number of edges attached to it. The *degree distribution* is denoted $P(k)$ and is defined as the probability of any given node having a degree of

k. We can further calculate the *mean degree* as $\langle k \rangle = \sum_{k=0}^{\infty} P(k)$ or simply $\langle k \rangle = \frac{\sum_i k_{n_i}}{N}$ assuming we have N nodes in the network.

When generating suitable networks for an epidemic model, we are faced with the decision of how to appropriately model the connections of the individuals, i.e. the edges between the nodes. One naive approach would be to simply connect all nodes with each other. This is called a *complete graph* [Boccaletti et al., 2006, p. 181] and an example for a network with $N = 20$ nodes is depicted in the left column of Figure 2.5. In a complete graph of size N, each node has the same degree $k_{n_i} = N - 1 \; \forall i$ and hence $\langle k \rangle = N - 1$. This is clearly not a very realistic representation of most social networks.

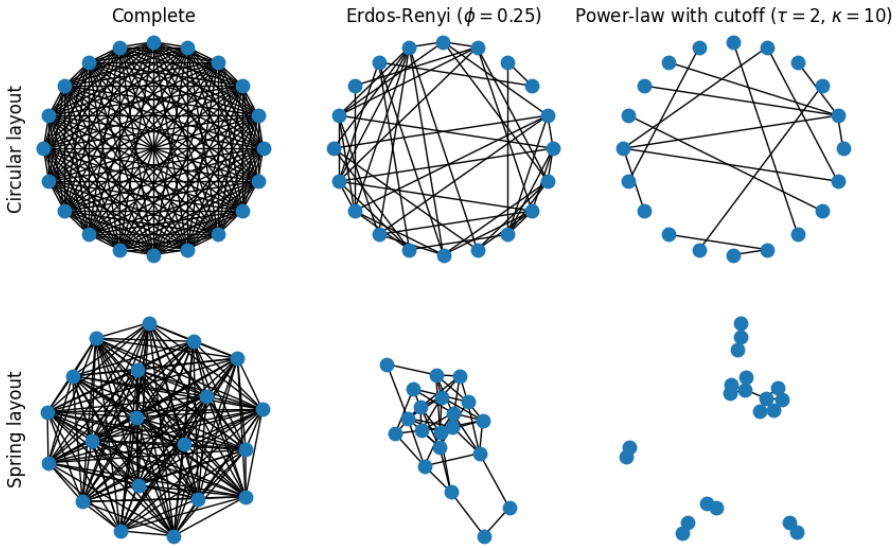
A potentially more promising approach would be to use an *Erdős-Rényi* (ER) graph, which starts out with N disconnected nodes from which each possible edge is then randomly included with a probability of ϕ [Boccaletti et al., 2006, p. 191]. For ER graphs, we can deduce that $\langle k \rangle = \phi N$ as $N \to \infty$. Figure 2.5 (middle column) gives an example of an ER graph with $N = 20$ and $\phi = 0.25$. While this approach can generate significantly less connected graphs (by setting a small ϕ), the resulting network still does not really resemble the clustered structures that we find in real social networks.

In reality, social networks tend to be much less homogeneous than ER graphs in the sense that very few nodes tend to be highly connected while the vast majority of nodes have much smaller degrees. Several graph generation algorithms have been proposed to model this via a power law distribution of degrees [Pastor-Satorras et al., 2015, p. 935]. However, since power law models tend to produce a few nodes with degrees that are implausibly high for social networks, Newman et al. [2001] propose a modified approach which introduces an upper cutoff for the power-law generated degrees preventing implausibly high values. In this model the probability for a given edge to be included in the graph is given by

$$P(k) = \frac{k^{-\tau} e^{-k/\kappa}}{Li_{\tau}\left(e^{-1/\kappa}\right)} \tag{2.15}$$

where τ is the exponent of the power-law distribution, κ the cutoff value, and Li the polylogarithm [Newman et al., 2001, p. 4]. A graph generated with this approach ($N = 20$, $\tau = 2$, $\kappa = 10$) is depicted in Figure 2.5 (right column). In the spring layout (bottom row) it is clearly visible how the graph now includes some nodes that are highly connected while the majority of nodes is far less connected resembling what we observe in the real world.

Figure 2.5: A complete graph, an Erdős-Rényi graph ($\phi = 0.25$), and a graph created with degrees from a power law with cutoff distribution ($\alpha = 2$, $\kappa = 10$), all with $N = 20$ nodes. The first row shows a circular layout and the second row shows the same graphs in a spring layout.



In Section 2.1.2 we defined the basic reproduction number $\mathscr{R}_0$ and its calculation as well as its role as a threshold. When using networks to model epidemics, the characteristics of the network influence the way the disease spreads and hence also the relation between $\mathscr{R}_0$ and the infection and removal rate. One approach to deduce the epidemic threshold and $\mathscr{R}_0$ in the network case is the *degree-based mean field* approach, which assumes that all nodes of degree k are statistically equivalent [Pastor-Satorras et al., 2015, p. 937]. Defining the transmissibility as $T = \frac{\beta}{\alpha}$, the epidemic threshold is

$$T_c = \frac{\langle k \rangle}{\langle k^2 \rangle - \langle k \rangle} \tag{2.16}$$

where $\langle k^2 \rangle$ is the second moment (variance) of the degree distribution. An epidemic outbreak occurs if and only if $T > T_c$ [Pastor-Satorras et al., 2015, p. 947]. Given that in terms of $\mathscr{R}_0$ the threshold value is 1, we can establish that at the threshold

$$\mathscr{R}_0 = \frac{T}{T_c} = T \frac{\langle k^2 \rangle - \langle k \rangle}{\langle k \rangle}. \tag{2.17}$$

This is an important result and will be used later on to derive the infection probability β from the more commonly reported $\mathscr{R}_0$.

Having reviewed this background on network approaches to compartmental epidemic models, we build on these findings during the development of the human contact network in Section 3.

2.2 Models of COVID-19

The largely unprecedented events of the COVID-19 pandemic generated significant research interest in the topic of epidemic modelling, which lead to numerous models aimed at understanding the spread of the disease. Meehan et al. [2020] give a comprehensive overview over the role of mathematical modelling of the disease. The authors note that after the original outbreak in Wuhan, China, simple models based on air travel data helped understand the potential of the disease spreading globally. Based on data from the first outbreak, researchers and public health officials were able to give a first estimate of $\mathscr{R}_0$. Compartmental models were developed early on in the pandemic as they allow for the calculation of rough, but important results (such as the epidemic size) even in the absence of detailed epidemiological data. As more detailed data became available, models were adapted to represent the disease more realistically through the inclusion of elements such as age dependency, however, uncertainty about the data and inconsistent study results remain issues.

In the following section we provide an overview over models that have been developed for COVID-19. Note that this review is not exhaustive as the number of models is simply too large and continued interest in the topic leads to new models appearing to this date. Instead, we aim to provide an overview over the different dimensions of the models: their goals, their scope, as well as the modelling approach used.

2.2.1 Model goals

Generally, we can identify two rough categories of model goals, although there is not always a clear cut between the two.

On the one hand, a model can aim to replicate the pandemic situation holistically, either to estimate epidemiological parameters or to provide forecasts of the real-world development of COVID-19 in a certain context (e.g. a country). These models strive to provide guidance for governments and health care providers to gauge the future spread of the pandemic, for

example, with regards to the stress put on hospitals. On the other hand, a model can strive to be more theoretical in the sense that it demonstrates the results of a certain strategy or specific interventions, such as a certain quarantine strategy in a more isolated scenario. These types of models tend to deviate more from the real world since the evaluation of a newly proposed strategy should first be undertaken without the interference and possible interaction of other methods. In particular, the implementation of several interventions in quick succession by governments make it hard to empirically validate the effect of any single intervention [Meehan et al., 2020, p. 66], which justifies the use of more theoretical approaches.

We first present notable examples of the first of the two goals. Liu et al. [2020] review twelve early studies that estimate $\mathscr{R}_0$ that use different approaches to estimate the basic reproduction number, among which three use compartmental models, four rely on statistical exponential growth models, and seven on various other mathematical/statistical approaches.

Anastassopoulou et al. [2020] develop a Susceptible-Infected-Recovered-Dead compartmental model based on data from Wuhan, China. Beyond estimating $\mathscr{R}_0$, the per day infection, mortality and recovery rates, the model is used to forecast the outbreak in Wuhan for three weeks and it is found that the official data released for the predicted period fell within the predicted bounds.

An extensive compartmental model with eight compartments (Susceptible, Infected, Diagnosed, Ailing, Recognised, Threatened, Healed, Extinct) was developed by Giordano et al. [2020] with the stated goal of predicting the course of the pandemic and to provide guidance for the planning of interventions. The model was developed and tested on data from Italy, where a large COVID-19 outbreak occurred in the spring of 2020. Using results from the developed model, the authors assert that a combination of lockdowns and large-scale testing is crucial to ending the COVID-19 pandemic.

Overall, Friedman et al. [2021] identify $n = 386$ different predictive models of COVID-19 at the end of the year 2020. Among these, seven were identified as covering more than five countries, regularly updated, publicly released and sufficiently verifiable[1]. Evaluating their predictive power, it was found that the models generally performed reasonably despite the difficulty of modelling both human behaviour as well as government interventions.

On the side of the second of the two goals, the isolated modelling of certain intervention strategies, the following are notable examples. Meidan et al. [2021] investigate the results of a so-called alternating quarantine strategy in which at any given time during the pandemic, half the population remains quarantined while the other half maintains their daily routines, with the groups switching weekly. The strategy is evaluated on a social network model of

[1] For the exact five criteria, see Friedman et al. [2021, p. 10]

10^4 nodes where nodes are clustered into households and their interaction depends on whether they are currently in the quarantined or the active part of the population. The disease model incorporates a presymptomatic stage, three levels of symptomatic stages as well as compartments for hospitalised and ventilated individuals. The model is parameterised with estimates of β and α based on data from several countries. The results of the study suggest that an alternating quarantine strategy can provide an economically and socially sustainable, yet very effective strategy in terms of containing COVID-19.

In a different approach, Plazas et al. [2021] develop a multiplex network with a household, work, and social layer to more accurately model different types of contact environments. The study aims to find a strategy to partition the population in a way that minimises interactions between members from different partitions to control the spread of the pandemic while reducing the economic cost of a full lockdown. The effectiveness of the partitioning strategies in terms of disease control is evaluated by simulating an SIR process on the network. They find that *random aggregation* provides the best strategy for containing the disease but also leads to significant job loss. A compromise is proposed with the *minimal aggregation* strategy that creates sufficient segregation in the network while also minimising link removal on the work and social layer.

An SEIR model is used by de Vlas and Coffeng [2021] to evaluate an exit strategy of a phased lift of control that attempts to achieve herd immunity in a small sub-population while keeping the remainder of the population under lockdown before moving to the next sub-population until full herd immunity is achieved. Simultaneously, the authors propose that since the health care system would only be under pressure in one sub-population at a time, surplus resources from other regions could be moved as required. The authors run the SEIR on a stochastically simulated population with sub-populations and find that the average time spent under lockdown would add up to about 14 months before full herd immunity.

2.2.2 Model scopes

A second dimension of model type is given by the scope of the model. Generally, we can distinguish between models that only represent the situation in a limited region (e.g. a city, a country, few neighbouring countries) versus models that take a more global approach. It is up for discussion, how many countries a model should include to count as global, for example, Friedman et al. [2021] classify models as global if they include more than five countries. The authors also find that among $n = 386$ screened predictive models, the majority ($n = 327$) takes a local approach [Friedman et al., 2021, Supplementary Information].

A local model for the COVID-19 pandemic in France was developed by Hoertel et al. [2020] who implement an agent-based micro-simulation model. As is typical for these types of models, the authors rely on a large battery of parameters ($n = 194$) to model the French population, the social network, and disease characteristics. Based on the simulation results, they conclude that a combination of physical distancing, mask wearing, and shielding of vulnerable individuals, could prevent the need for a second lockdown in France.

The SIDARTHE model by Giordano et al. [2020] that was described in Section 2.2.1 provides another example of a locally scoped model.

A notable globally scoped model of $n = 164$ countries was developed by the IHME [Institute for Health Metrics and Evaluation, 2020], which is essentially a group of models that is being adapted over time: while the initial model was a statistical curve fit model, it evolved into a hybrid model of statistical curve fit and SEIR, until the statistical curve fit was replaced by a spline fit of the relationship between log cumulative deaths and log cumulative cases [Friedman et al., 2021, p. 11].

In Section 2.2.3 we describe models developed by the LANL Covid-19 Team [Los Alamos National Laboratory COVID-19 Team, 2020] and the MRC [MRC Centre for Global Infectious Disease Analysis, 2020] that are further examples of a global modelling scope.

2.2.3 Model methodologies

Lastly, we can distinguish models for COVID-19 based on the modelling approach used. Of course, hybrid models that rely on a combination of modelling approaches are also common, as is the case with the previously mentioned IHME model [Institute for Health Metrics and Evaluation, 2020]. For this review, we stick to the most widely used approaches and give a few notable examples from the literature.

A prime example of a compartmental model is the LMIC model which uses an age-structured SEIRD model that further differentiates infections by severity [Walker et al., 2020]. The model targets low- or middle income countries, many of whom are still in earlier stages of the pandemic. The model is still being updated and available online [MRC Centre for Global Infectious Disease Analysis, 2020].

Chang et al. [2021] take a network model approach, devising a mobility network of 98 million individuals derived from mobile phone data including the hourly movements of the individuals. Predicting the case trajectories using an SEIR model on the network provides a good fit to the

empirical case trajectory. The authors conclude that the majority of infections stems from a small number of points of interest, indicating that the restrictions of large gatherings at points of interest is more effective in terms of containment than a uniform reduction in mobility. The model also suggests that the infection rates among racially or socioeconomically disadvantaged groups exceed those of the rest of the population merely as a result of a difference in mobility patterns.

As described in Section 2.2.2, the model developed by Hoertel et al. [2020] uses agent-based micro-simulation to model the pandemic in France, providing an example of another widely used approach that allows for very detailed modelling of individual behaviour at the cost of a vast number of required parameters.

Yet another approach is taken in the model developed by the LANL Covid-19 Team [Los Alamos National Laboratory COVID-19 Team, 2020], which forecasts the number of COVID-19 infections using a dynamic growth parameter which is updated based on trends in the number of reported cases. The mortality estimates are then calculated using the new case estimate and the (dynamically estimated) case fatality rate.

3 Development of a Human Contact Network

A promising approach to recreating realistic social networks is to make use of real-world mobility data that records the movements of individuals using location tracking of their mobile devices. Based on this data, it is possible to identify patterns in the movements around public places as well as in private settings. For privacy purposes, the data is usually anonymised through aggregation and the addition of noise. To help understand mobility patterns of the population to analyse the effects of countermeasures against COVID-19, several providers have made parts of this data available to the public. While the data varies by country and level of detail, the most widely used data in research is provided by Google [Google LLC, 2021], Facebook [Facebook Inc., 2021], and SafeGraph [SafeGraph Inc., 2021b]. Generally, the data sets provided by Google and Facebook are more aggregated but available for most countries, while the SafeGraph data is more granular, albeit only for the US.

We adapt an approach proposed by Klise et al. [2021] to create a social network from mobility data of the Patterns data set from SafeGraph [SafeGraph Inc., 2021b], which contains data on approximately 4.5 million points of interest (POI) in the US. Among else, the data contains counts of the number of daily visitors grouped by the census block group (CBG) of the visitor. Using the POIs as points of interaction between individuals from different CBGs, we attempt to create a social network that captures the characteristics of the real population more realistically.

The approach taken in this thesis differs from that of Klise et al. [2021] in several ways. Most importantly and quite unfortunately, SafeGraph has taken additional data used by Klise et al. [2021] offline, which would have contained data similar to the Patterns data set but on a private level rather than based on POIs. This allows for more detailed modelling of social interactions, but we attempt to compensate for this lack of data through a few reasonable assumptions. Secondly, Klise et al. [2021] use a temporal network that models connections between nodes on an hourly basis, while our network is assumed to be static. The temporal edges are further weighted by the strength of the connection between nodes which is in turn based on the number of interactions found in the mobility data. In contrast, the edges in our network are not weighted and we instead account for the difference in connectivity between different CBGs by probabilistically drawing edges from a distribution that favours highly frequented connections. This is approach is arguably less accurate but the lack of data on private interactions made this the more appropriate choice. Lastly, we had to limit our network to a small sample of the full data set to keep computational

times within reason. We followed Klise et al. [2021] by creating a network of 10,000 individuals from a single US county, in our case New Haven County, CT, with a population of around 850,000. The reason for this seemingly small network lies in the prohibitively high computation time required for building the network from the mobility data as well as the high time complexity of many graph algorithms.

Two separate networks are devised, the first replicating a situation before any official guidance for social distancing was issued and a second network after the issuance of stay-at-home orders. The state government of Connecticut started issuing several such orders over the course of March 2020 [ct.gov, 2021]. Hence, the baseline network (without stay-at-home orders) is based on data from February 2020, while the restricted network (under stay-at-home orders) is based on April 2020 data.

We further make use of the most recent Open Census Data provided by SafeGraph that contains a wide range of demographic indicators on a CBG level from the year 2019 [SafeGraph Inc., 2021a]. To more appropriately model household sizes in our network, the household cluster sizes in a given CBG are drawn from a distribution around the mean household size as reported in the Open Census Data. To validate the reduction in mobility in the networks, data for New Haven County, CT, from the Google COVID-19 Mobility Report is used [Google LLC, 2021].

All data processing and network implementations were done in Python. The creation of the networks relies heavily on the Python libraries *networkx* [NetworkX developers, 2021] and *epydemic* [Dobson, 2017b].

3.1 Constructing the network

To create the human contact networks from mobility data, we devise a workflow to extract the data, transform it and aggregate it into more suitable data structures before finally building the network.

3.1.1 Data extraction and transformation

The workflow starts by extracting the mobility data sets for February and April 2020 from the raw CSV files before aggregating them to show how often individuals visited the different CBGs over the observed period (we will refer to this as *trip count* data). We further extract information about the average household size and population of each CBG from the census data set as well as

mobility indices for the two months from the Google Mobility Report data set[1]. The workflow was devised as an ETL-like process that would allow for straightforward use of other data such as that of another county or month.

Using the trip count data, we instantiate an object of the *NetworkData* data class, which provides methods to access and handle the mobility data. Firstly, we can create an adjacency list that maps the transition probability from a given CBG to all other CBGs (in a fixed order) to each CBG. The transition probability from CBG i to CBG j is given by

$$p_{ij} = \frac{comb\_counts_{(i,j)}}{trip\_counts_i}. \tag{3.1}$$

Consequently, the resulting adjacency list for n CBGs is a Python dictionary with all n CBG codes as keys and the corresponding n transition probabilities for each CBG in a list as values. From these lists we can create the cumulative sum of probabilities, which we will use as the basis for a probability distribution later on. Secondly, we compute for each CBG the change in the total number of trips leaving that CBG from February 2020 to April 2020.

3.1.2 Network creation

To create the network we adapt an algorithm proposed by Dobson [2020, pp. 157] to build a physically distanced human contact network. We adapt the behaviour of the algorithm in several ways to integrate the mobility data into the network.

The algorithm can be split into five steps starting with the creation of the households. For each CBG, we create a number of small complete graphs that represent households. The household size is drawn from a truncated normal distribution with a mean that corresponds to the average household size of the CBG found in the census data (the standard deviation defaults to half the mean). The number of nodes per CBG is approximately proportional to the CBG population. These household networks are then integrated into a single graph.

In the second step, each node is assigned a degree from a power law with cutoff distribution with exponent $\tau = 2$, a common setting for real-world networks [Newman et al., 2002, p.6] and $\kappa = 40$ cutoff[2]. For each degree of a node, we create a copy of that node not connected to any

[1] Two of the 628 CBGs have missing values for the household and population size. Since this is a very small number, we impute them with the respective mean value of all other CBGs.

[2] This distribution determining the number of extra-household connections of the nodes, is the only distribution

nodes yet (called a *stub*). In case the sum of all degrees is odd, an additional random node is copied, since we will require pairs of nodes in the next step.

Iterating over every other stub ($stub_i$), the stub next in line ($stub_{i+1}$) is swapped with a random other stub ($stub_j$). Recall that each node belongs to a CBG and the CBG of $stub_i$ is CBG_i. Hence, $stub_j$ is determined as follows. First, draw a random CBG where CBG_k is chosen with a probability of p_{ik}, corresponding to the transition probabilities defined in Equation 3.1. Second, pick a random stub from CBG_k, which is $stub_j$. This results in a reordered list of stubs that represent connections occurring with a relative frequency proportional to empirical connections between CBGs.

Steps four and five are straightforward. Any stub pairs that connect two nodes of the same household are broken up by swapping them with a random other node. Lastly, each stub pair is connected by an edge which is added to the graph.

We implemented this algorithm in the *create* method of the *MobilityNetwork* class and implemented a corresponding *MNGenerator* class, which inherits from the *epydemic* class *NetworkGenerator*. This allows us to pass the the generator with the desired parameterisation to the epidemic models of the *epydemic* framework, as discussed in Section 4.1.

3.2 Characterising the network

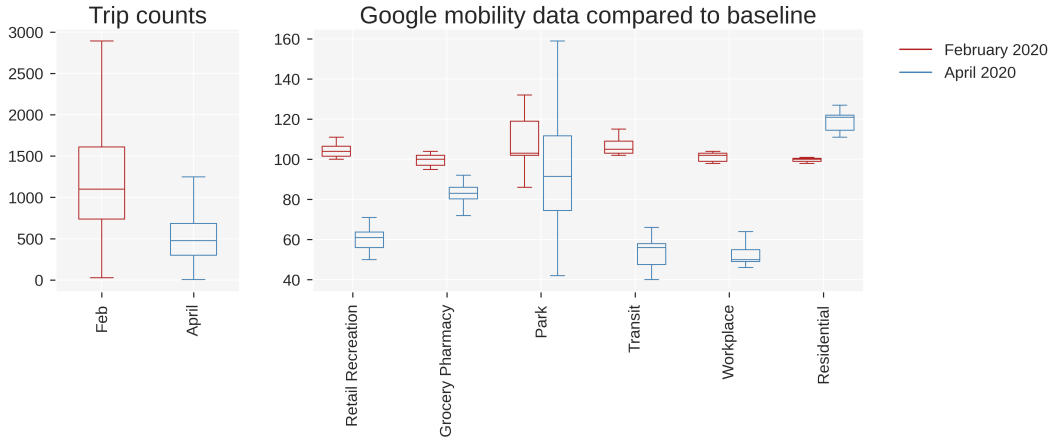
With the networks for February and April 2020 constructed, we can now take a closer look at some of their characteristics and critically assess to what extent they reflect the change in mobility after stay-at-home orders were issued in March 2020.

Before looking at the networks directly, we can observe a change in the mobility patterns by aggregating the total number of outgoing trips for each CBG for both months. As expected, the average number of trips per CBG reduces from ≈ 1240 in February to ≈ 521 in April, indicating a reduction in mobility. (Note that due to the large differences between CBGs, the standard error of these estimates are very large.) This is visible in the left plot of Figure 3.1. We can compare these numbers to data from the Google COVID-19 Mobility Report for New Haven County, which provides daily estimates of the change in mobility related to six categories compared to a baseline value from January 2020. The mobility estimates are based on location services of mobile devices. This data is aggregated for February and April 2020 and depicted in the right

of the algorithm whose parameters are not based on empirical data. This is an unfortunate result of the aggregation of the data and lack of detail on subject-level mobility.

plot of Figure 3.1. Mobility decreased across all categories except for residential mobility, which confirms the trend we saw in the SafeGraph Patterns data set, the basis of the networks.

Figure 3.1: Mobility metrics of the sampled SafeGraph data [SafeGraph Inc., 2021b] and Google COVID-19 mobility report for New Haven County [Google LLC, 2021]. Red boxes correspond to data from February 2020, blue boxes to April 2020. Plot on the left displays the number of trips leaving each CBG. Plot on the right depicts the mobility in different categories in relation to a baseline value of 100 (January 2020).



To assess whether our networks capture this change in mobility, we use four different network metrics following the approach proposed by Klise et al. [2021]:

- **Node degrees**: As described in Section 2.1.4, the degree k of a node is the number of its edges. In our case, this corresponds to the number of contacts with other individuals.

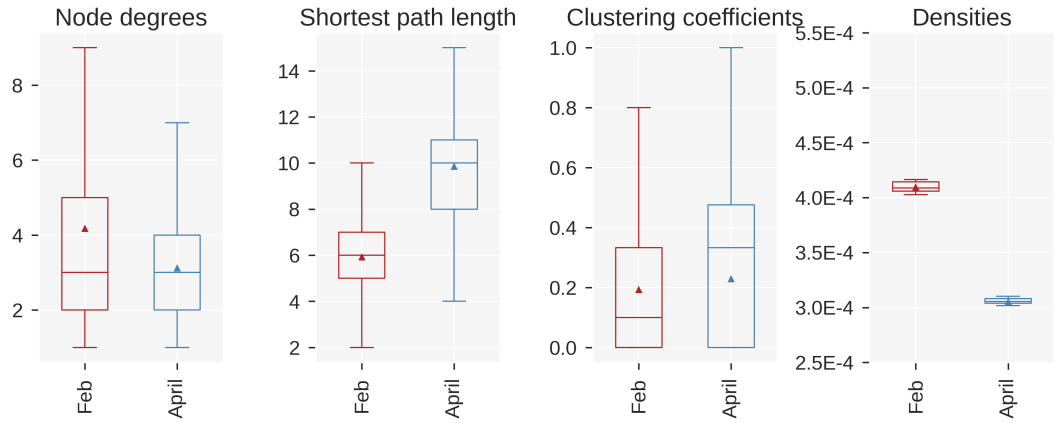
- **Shortest path length**: The shortest path length between a pair of nodes describes the minimum number of edges that have to be traversed to travel between the two. A large average shortest path length signifies that individuals are further apart from each other, something we would expect from a socially distanced network.

- **Clustering coefficient**: The clustering coefficient quantifies the tendency of nodes in a network to build clusters. It is calculated as the fraction of possible triangles through a given node, with values from 0 to 1 and a higher coefficient signifying more clustering.

- **Network density**: The network density compares a a given graph G to a complete graph K of the same number of nodes by dividing the number of edges of G by the number of edges of K. This quantifies the overall connectedness of individuals in the network.

To compare these measures between the February and April networks, each metric was evaluated on ten instances per network and the distribution of the metric results is given in Figure 3.2.

Figure 3.2: Network metrics of the networks created for February 2020 (red) and April 2020 (blue) from the sampled data. From left to right the plots depict: (1) node degrees, (2) shortest path length for all pairs of nodes, (3) clustering coefficient, and (4) node density. Triangle markers show the mean value.



Since the calculation of the shortest path length is very computationally expensive, this measure is based on a single instance per network, however, manual tests have shown that the results are fairly robust.

The average node degrees of the networks are depicted in the boxplot to the left of Figure 3.2. We can see that node degrees are smaller in the mean (however not in the median) in April than in February, indicating that the networks capture the reduction in the number of connections between individuals after the stay-at-home orders were issued.

A large difference between the two networks is visible in the second graph of Figure 3.2 which depicts the shortest path lengths of the networks, showing a clear increase in shortest path length from February to April. This can be interpreted as nodes of the April network being further away from each other on average than in the February network, capturing a trend we would expect in light of the stay-at-home orders.

Clustering coefficients of the networks are depicted in the third boxplot of Figure 3.2. While the difference in means is marginal, the coefficient still increases in the median from February to April. In the context of social distancing as expected in the April network, a higher clustering coefficient indicates that individuals are more likely to visit a few highly connected locations (e.g. grocery stores) as well as interactions within their own household.

Lastly, the plot to the right of Figure 3.2 depicts the network density of both nodes. From February to April the density decreases by about 25%, indicating an overall decrease in the number of contacts individuals interacted with in April.

In summary, the measures suggest that the networks mostly capture the change in behaviour of

the individuals before and after the stay-at-home orders were issued. As described before, we modified to approach to create the networks proposed by Klise et al. [2021] due to a lack of data. From our analysis of the network, these simplifications do not appear to significantly change the characteristics of the networks. In particular, all four trends observed in the networks are also found to a similar extent in the networks devised by Klise et al. [2021, pp. 9].

3.3 Non-mobility based alternative networks

The mobility based network model devised in the last sections attempts to capture more of the characteristics of interactions between individuals of a population. However, this is highly dependent on the availability of data with a sufficient level of detail for the target population. Building the model from mobility data is also more complicated since the procedure is dependent on the type of available mobility data. Therefore, human contact networks often rely on more theoretical algorithms that are known to produce networks resembling the real world, some of which were discussed in Section 2.1.4.

In this project, we incorporate two alternatives to mobility based networks; a power-law with cutoff network (*PLC*) and a network built with the algorithm used by Dobson [2020, pp. 156] for building networks that imitate social bubbles (households) and connecting them according to some distribution (*Distanced network*). The reason we chose these two networks is that the mobility network is a combination of both. It uses the same procedure as the distanced network for building households (albeit using empirical demographic data), adapts the procedure for connecting nodes (although biased by the inter-CBG trip counts) but uses a PLC distribution for the number of out-of-household connections per individual. For both networks, we develop a baseline network and a restricted network that assumes less interaction between the nodes.

The networks are parameterised in a way that attempts to make the average node degree similar to that of the corresponding mobility network. Recall that the average node degree of the baseline and restricted mobility network is $\langle k_{baseline} \rangle \approx 4.19$ and $\langle k_{restricted} \rangle \approx 3.13$, respectively.

For the PLC network we achieve a similar average node degree by fixing the cutoff to $\kappa = 40$ and solving the following equation (the first derivative of the probability generating function of the PLC distribution evaluated at 1 [Newman et al., 2001, p.5]) for the exponent $\tau_{baseline}$

$$\langle k_{baseline} \rangle = \frac{Li_{\tau_{baseline}-1}\left(e^{-1/\kappa}\right)}{Li_{\tau_{baseline}}\left(e^{-1/\kappa}\right)} \Longleftrightarrow \tau_{baseline} \approx 1.57 \tag{3.2}$$

and similarly $\tau_{restricted} \approx 1.78$.

For the distanced network the node degree distribution is determined by an exponential distribution from which is drawn to determine the number of out-of-household connections of contact nodes. The expected node degree can hence be set by passing the mobility network $\langle k_{baseline} \rangle$ and $\langle k_{restricted} \rangle$, respectively, as the scale parameter to the exponential distribution. In the case of the distanced network we can further make the networks comparable by setting the average household size to the empirical value observed in the demographics data, approximately 2.52.

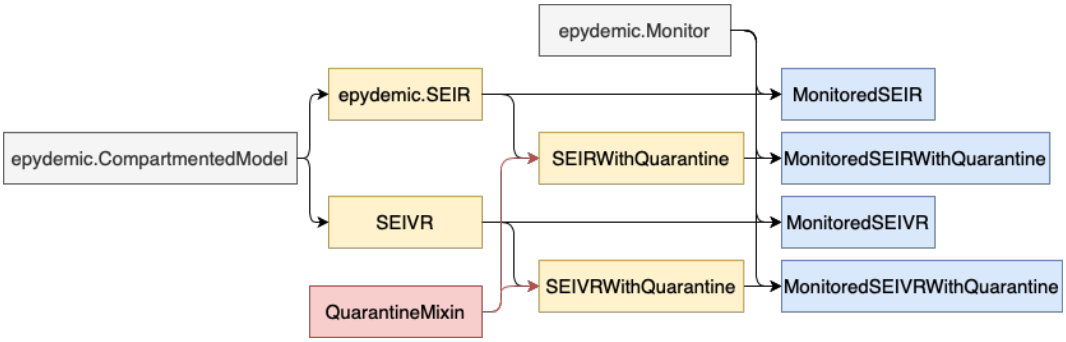
The distanced network was implemented in the *DistancedNetwork* class and a corresponding *DNGenerator* network generator class was developed for use with the epidemic models. A generator for the PLC network is already implemented in *epydemic* out of the box.

4 Implementation of the Compartmental Models

In this section, we describe the development and implementation of two compartmental models for COVID-19, which are then run on the networks devised in the previous sections. As was extensively reviewed in Section 2, most models developed so far include a removed compartment that individuals enter after they have recovered from infection. At this point, it appears that individuals infected with SARS-CoV-2 acquire protection from the virus for some time after recovery, however, the length of immunity is still debated [Edridge et al., 2020]. In the long run, a model allowing reinfection is certainly appropriate especially given the emergence of different virus variants. However, as the topic of this project is the modelling of epidemics as opposed to endemics with recurring waves of infections, we stick to the most commonly used modelling approach that assumes recovery from infection provides immunity. We first devise our base model, an SEIR model as described in Section 2.1.1. This is then extended to an SEIVR model which includes an additional compartment to keep track of vaccinated individuals. Lastly, we implement a quarantine extension that can be added to either model to explore the effects of this policy on the epidemic.

We use the epidemic modelling framework *epydemic* for all our models, which was developed with the goal of providing a comprehensive and extendable toolkit for simulating various epidemic processes [Dobson, 2017b]. Figure 4.1 gives an overview of the classes used for our models, which are described in more detail in the following sections.

Figure 4.1: Class structure of the models. Grey classes are parent classes only; yellow classes represent the model classes; blue classes are monitored models used for simulation; red classes are mixins.



4.1 Background on *epydemic*

In this section, we provide a brief overview on how to implement an epidemic model using *epydemic*. We strictly focus on the parts that are relevant for the implementation at hand and refer to the official documentation for more detailed explanations [Dobson, 2017a].

The basis for the implementation of any compartmental model in *epydemic* is the *CompartmentedModel* class, which defines the disease as a process on a network where each node is in exactly one compartment at a time. As an abstract class, we are required to implement a child class that inherits from *CompartmentedModel* and overrides the *build* method. This method takes a dictionary containing the required model parameters as argument and is responsible for building the process. That is, it adds the compartments with their respective starting occupancy, defines the possible transitions between compartments by adding so-called *loci* to the edges and setting the probabilities of events on these edges.

Beyond overriding the *build* method, the model class must implement the methods that define the events that can occur on nodes or edges, such as the transmission of the disease or the transition between compartments. These methods all take two arguments: the time step of the simulation t and either the node n for which the event occurs or the edge e on which the event occurs. The model classes are highlighted in yellow in Figure 4.1.

Once the setup of the model class is completed, we could start running experiments with it. However, to keep track of the size of each compartment over time, we need to create a monitored model class that inherits from the model as well as the *epydemic* class *Monitor* using multiple inheritance. The *Monitor* automatically calls an *observe* method in a regular time interval that tracks the size of the *loci* and makes them available at the end of the simulation as a time series. The monitored classes are highlighted in blue in Figure 4.1.

At this point, we can create an instance of the monitored model class as well as a dictionary with the model parameters. Passing both to the *SochasticDynamics* class of *epydemic*, we can run the simulation and collect a dictionary with the results.

4.2 SEIR model

epydemic comes with a standard implementation of an SEIR model, which is fully appropriate for our case. The implementation deviates slightly from the standard SEIR formulation we encountered in Section 2.1.1 in that the exposed compartment contains infected, asymptomatic,

infectious individuals (as opposed to them not being infectious in the standard model). The reason for this adaptation is that it better describes the disease progression of COVID-19 where individuals usually go through a short period of time after infection in which they are not yet symptomatic but still infectious. The adapted model formulation is given as

$$\dot{S} = -\frac{\beta IS}{N} - \frac{\gamma ES}{N} \tag{4.1}$$

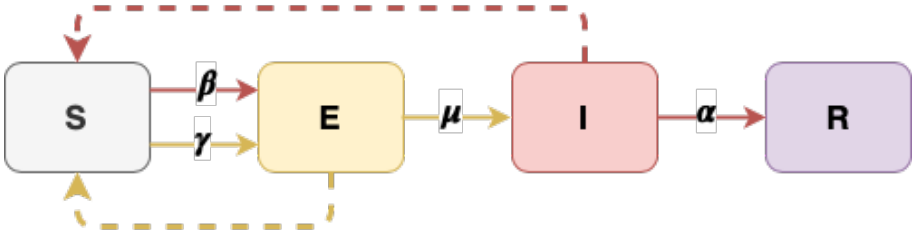
$$\dot{E} = \frac{\beta IS}{N} + \frac{\gamma ES}{N} - \mu E \tag{4.2}$$

$$\dot{I} = \mu E - \alpha I \tag{4.3}$$

$$\dot{R} = \alpha I \tag{4.4}$$

using the same notation as in Section 2.1.1 in addition to the parameter γ, which stands for the rate of infection during contacts between exposed and susceptible individuals. The transitions and corresponding parameters are visualised in Figure 4.2 where the dashed lines represent the transmission of the disease and solid lines the transition of individuals between compartments.

Figure 4.2: SEIR compartments and transition parameters as implemented in *epydemic*. Dashed arrows represent the transmission of the disease; solid arrows represent compartment transitions. The colour of the arrow corresponds to the colour of the compartment that is responsible for the event.



Since the SEIR model is already implemented in *epydemic*, we only need to sub-class *SEIR* together with the *Monitor* class to get the *MonitoredSEIR* class. The parameters required are listed in Table 4.1 together with their description and the respective symbol used in the mathematical formulation of the model (where applicable).

4.3 SEIVR model

The approval of the first vaccines for COVID-19 in late 2020 has given hope for a way out of the COVID-19 pandemic and an end to socially and economically strenuous lockdowns and

Table 4.1: Parameters required by the SEIR model.

Parameter	Mathematical symbol	Description
p_exposed	n/a	Pr. of initially exposed
$p\_infect\_symptomatic$	β	Pr. of infection (I infects S)
$p\_infect\_asymptomatic$	γ	Pr. of infection (E infects S)
p_symptoms	μ	Pr. of exposed developing symptoms
p_remove	α	Pr. of removal

other containment measures. As vaccines move into the focus of attention, an updated model is required to explore the effect that vaccines could have on the progression of epidemics.

We, therefore, extended the SEIR further by adding a V (vaccinated) compartment that contains all those individuals who have received a vaccination against the disease. We assume that only susceptible individuals can be vaccinated since vaccinations are usually not administered while one is infected and the model assumes immunity after recovery from the disease.

Further, no vaccine provides complete protection but rather reduces the risk of infection by some percentage. This is generally referred to as the vaccine efficacy, called relative risk reduction (*RRR*) in clinical trials, which is one minus the rate of infection amongst vaccinated individuals divided by the rate of infection amongst unvaccinated individuals. For example, if the rate of infection in the unvaccinated population is 50% and the *RRR* of the vaccine is 90%, then the rate of infection for a vaccinated individual is reduced to $50\% \times (100\% - 90\%) = 5\%$. As a result, vaccinated individuals can still be infected and move back into the exposed compartment albeit with a lower probability than susceptible individuals.

There are two things to consider regarding the rate at which individuals are vaccinated. Firstly, the fraction of the population that is initially vaccinated is relevant for diseases for which a vaccine already exists before the outbreak of the epidemic (e.g. influenza). However, in cases such as COVID-19, no vaccine is available initially and the epidemic starts in a fully unvaccinated population. Secondly, the rate at which individuals receive their vaccination determines how quickly individuals move from the susceptible into the vaccinated compartment. The expected time for one individual until receiving a vaccination is the inverse of the vaccination rate.

The other transitions between compartments remain the same as in the SEIR model. The mathematical formulation as a system of ODEs is given as follows:

$$\dot{S} = -\frac{\beta IS}{N} - \frac{\gamma ES}{N} - \nu S \tag{4.5}$$

$$\dot{E} = \left(\frac{\beta IS}{N} + \frac{\gamma ES}{N}\right) + \left((1-\rho)\frac{\beta IV}{N} + (1-\rho)\frac{\gamma EV}{N}\right) - \mu E \tag{4.6}$$

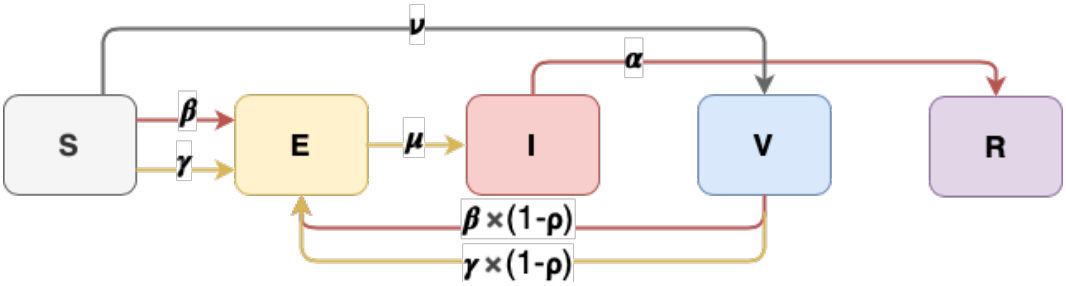
$$\dot{I} = \mu E - \alpha I \tag{4.7}$$

$$\dot{V} = \nu S - \left((1-\rho)\frac{\beta IV}{N} + (1-\rho)\frac{\gamma EV}{N}\right) \tag{4.8}$$

$$\dot{R} = \alpha I \tag{4.9}$$

Figure 4.3 visualises the transitions between compartments. As described above, disease transmissions occur between susceptible individuals and the compartments E and I, as well as between vaccinated individuals and the compartments E and I.

Figure 4.3: SEIVR compartments and transition parameters. The colour of the arrow corresponds to the colour of the compartment that is responsible for the transition.



The implementation of the SEIVR model using *epydemic* is very close to what was described for the SEIR model and we refer to the accompanying code for details. A summary of the parameters used by the model including the corresponding mathematical symbols and descriptions is provided in Table 4.2.

Table 4.2: Parameters required by the SEIR model.

Parameter	Mathematical symbol	Description
p_exposed	n/a	Pr. of initially exposed
p_infect_symptomatic	β	Pr. of infection (I infects S)
p_infect_asymptomatic	γ	Pr. of infection (E infects S)
p_symptoms	μ	Pr. of exposed developing symptoms
p_remove	α	Pr. of removal
p_vaccinated	ν	Pr. of getting vaccinated
p_vaccinated_initial	n/a	Pr. of initial vaccination
RRR	ρ	Vaccine efficacy

4.4 Quarantine

One of the most commonly used containment measures during the COVID-19 pandemic is the use of quarantine to isolate infectious or potentially infectious individuals by temporarily shielding them from interactions with other susceptible individuals to prevent the spread of the disease. In compartmental models, such a policy could be implemented in at least two ways, either by creating an additional compartment that holds currently quarantined individuals or by changing the structure of the network itself by removing edges to and from the quarantined nodes with some probability. We choose to do the latter as it keeps the model simple and with careful implementation would facilitate the integration of quarantine in any other compartmental model.

Hence, we implement the *quarantine* event as a Python mixin, which may be included in the definition of the epidemic model class using multiple inheritance (highlighted red in Figure 4.1). The mixin defines the *quarantine* method which was taken from Dobson [2020, pp. 138] and takes the node *n* to be quarantined as input and detaches each of its edges with a probability *p_quarantine* substituting the endpoint *n* with a random other susceptible node thereby leaving the node *n* with only a fraction of contacts to infect. The code for the *QuarantineMixin* is provided in Listing 4.1.

```
1  class QuarantineMixin:
2
3      def quarantine(self, n: Node):
4
5          g = self.network()
6          rng = np.random.default_rng()
7          neighbors = list(g.neighbors(n))
8
9          for neighbor in neighbors:
10
11             # Keep going with probability `_p_quarantine`
12             if rng.random() > self._p_quarantine:
13                 continue
14
15             # Only remove susceptible neighbors
16             if self.getCompartment(neighbor) == self.SUSCEPTIBLE:
17                 self.removeEdge(n, neighbor)
18
19                 neighbor_prime = self.locus(self.SUSCEPTIBLE).draw()
20                 self.addEdge(neighbor, neighbor_prime)
```

Listing 4.1: *QuarantineMixin* for *epydemic* compartmental models.

The *QuarantineMixin* may be included in any model class that inherits from the *epydemic* class *CompartmentedModel*. It further requires the probability *_p_quarantine* to be defined as a class or instance variable and at least a susceptible compartment. Given this, the *quarantine* method of the mixin can be called from the model class from any event method. For example, if we assume individuals are quarantined right after they become infected, the method would be called from the *infect* method. Alternatively, if we assume quarantining happens after individuals become symptomatic, the method would be called from the *symptoms* method. The identification of infected individuals and subsequent quarantining of individuals at the time of infection requires the existence of a very thorough asymptomatic testing scheme. We chose to implement the more feasible scenario of quarantining individuals after they become symptomatic.

4.5 Parameterisation for COVID-19

The majority of parameters required for the epidemic models can be derived from empirical data. While not explicitly a parameter in the models, the basic reproduction number is used to determine the probability of infection (p_infect) and probability of removal (p_remove). Estimates for $\mathscr{R}_0$ of COVID-19 vary greatly between countries and regions. As such, it seems crucial to select and estimate that corresponds to the region being modelled. Sy et al. [2021] provide highly detailed estimates of $\mathscr{R}_0$ across the United States down to the county level. Since our model is based on mobility data from New Haven County, CT, we chose the corresponding county-level estimate of 2.85 [Sy et al., 2021, Supplemental Data].

The duration of infectiousness after the onset of symptoms is generally assumed to be around 10 days [Cevik et al., 2020, p. 17], [Gallo et al., 2020, p. 4]. Using days as time units, this equates to setting p_remove to 0.1.

p_infect (β) is estimated by plugging the mean and variance of the PLC distribution (which drives the node degree distribution of the mobility networks) into Equation 2.17 together with the estimate for $\mathscr{R}_0$ and p_remove (α). For the baseline network (February) this yields a value for p_infect of 0.0393. We could do the same calculation for the April network, however, this would imply that we assume that the epidemic started in the mobility-reduced network, which is not the case. Instead, we use the same value for p_infect for both networks. For the PLC and distanced network we use Equation 2.17 again plugging in the expected mean and variance of the respective network, giving an infection rate of 0.0182 (PLC) and 0.0895 (distanced).

As was noted by McEvoy et al. [2021, p. 6], the infectiousness of asymptomatic cases is approximately half the infectiousness of symptomatic cases. This translates into setting the value

Table 4.3: COVID-19 specific model parameters used in the compartmental models.

Variable	Value	Description	Reference
$\mathscr{R}_0$	2.85	Basic reproduction number	Sy et al. [2021]
p_infect	0.0393 (Mobility) 0.0182 (PLC) 0.0895 (Distanced)	Pr. of infection	calculated
$p\_infect\_asympt.$	0.0197 (Mobility) 0.0091 (PLC) 0.0448 (Distanced)	Pr. of asymptomatic infection	calculated McEvoy et al. [2021]
p_remove	0.1	Pr. of removal	Cevik et al. [2020] Gallo et al. [2020]
$p\_symptoms$	0.2	Pr. of becoming symptomatic	Gallo et al. [2020]
$p\_exposed$	0.01	Pr. of initially exposed	n/a
RRR	0.5 to 0.95	Vaccine efficacy	Polack et al. [2020] Baden et al. [2021] Voysey et al. [2021] FDA [2020]
p_vacc	0.001 to 0.01	Pr. of getting vaccinated	n/a
p_vacc_init	0 to 0.8	Pr. of initial vaccination	n/a
$p\_quarantine$	0 to 1	Pr. of rewiring	n/a

for $p\_infect\_asympyomatic$ to $0.0393 \times \frac{1}{2} = 0.01965$ for the mobility networks, 0.0091 for the PLC networks, and 0.0448 for the distanced networks.

Estimates for the duration between infection and onset of symptoms vary around a value of approximately 5 days [Gallo et al., 2020, p. 4]. In terms of $p\_symptoms$, this equates to a parameter value of 0.2.

In the case of the SEIVR model we also require the relative risk reduction (RRR), commonly referred to as the vaccine efficacy. Trial data for the currently approved vaccines provides an estimate of this measure with values ranging from about 65% to 95%.[1] Consequently, we run simulations with values of this parameter in the range from 0.5 to 0.95.

Lastly, the probability of being exposed at the start of the simulation, $p\_exposed$, is set to 0.01. There is no special meaning behind this parameter, it is simply set to seed a small number of individuals with the disease to start the epidemic.

On the other hand, the parameters p_vacc, $p\_vacc\_initial$, and $p\_quarantine$ are policy

[1]The RRR of the vaccines currently approved in the US and/or Europe are 95% for Pfizer-BioNTech [Polack et al., 2020], 94% for Moderna-NIH [Baden et al., 2021], 67% for AstraZeneza-Oxford [Voysey et al., 2021], and 67% for Johnson & Johnson [FDA, 2020].

parameters that are highly dependent on the specific model application and the vaccination and quarantine policies in the target population. As a result, we run simulations with different settings for the parameters in a reasonable range. For *p_vacc* these values lie between 0.001 and 0.01, i.e. on average, it takes between 100 and 1000 days for an individual to get vaccinated. The parameters for $p\_vacc\_initial$ are set in the range from 0 to 0.8, and $p\_quarantine$ in the range from 0 to 1.

All model parameters, their values, description and references (where applicable) are depicted in Table 4.3.

4.6 Extension to Influenza

While the research focus of this paper lies in modelling the COVID-19 pandemic, the developed approaches readily allow for an extension to other infectious diseases in a straightforward manner. To demonstrate this, we provide an alternative model parameterisation for the pandemic influenza using the same mobility networks.[2]

The influenza virus breaks out around the world on a seasonal basis with small changes to the virus between seasons. Pandemic influenza describes outbreaks of the disease in which a major change in the virus causes the vast majority of the population to be susceptible to infection despite previous infection with other strains of the virus. This phenomenon is by no means a rarity, with four influenza pandemics having occurred since the start of the 20th century [Biggerstaff et al., 2014, p. 1]. A review of estimates of the basic reproduction number of the most recent pandemic influenza outbreak in 2009 found a median estimate of $\mathscr{R}_0 = 1.46$ with an interquartile range from 1.30 to 1.70 [Biggerstaff et al., 2014, p. 6]. Based on data from the Ontario province in Canada, Tuite et al. [2010, p. 131] estimate the median incubation period of the same outbreak to be 4 days and the duration of symptoms to be 7 days.

This provides us with enough data to set the parameters of our model, similar to Section 4.5. The incubation period of 4 days translates to setting $p\_symptoms$ to 0.25, and the duration of symptoms of 7 days to setting *p_remove* to $\frac{1}{7} \approx 0.1429$. As before, we calculate p_infect for the baseline mobility network using Equation 2.17, which gives 0.0288. Research about the transmissibility of influenza from asymptomatic infected individuals is inconclusive [Patrozou and Mermel, 2009, p. 196], however, frequently cited mathematical models of the disease often assume a reduction in transmissibility compared to symptomatic cases of 50% [Longini Jr et al.,

[2]The simulations for influenza are run using the mobility networks only since these are the focus of this project and to save on simulation time (see Section 4.7).

2004, p. 632], [Ferguson et al., 2006, Supplementary Notes]. Hence, we assume the same reduction, which coincides with the reduction assumed for COVID-19.

At the start of the epidemic, we assume that 1% of the population is exposed, that is we set *p_exposed* to 0.01 as before. Since all other parameters in the model are policy-specific rather than disease-specific, we can use the same values as for COVID-19. A summary of all model parameters for influenza is depicted in Table 4.4.

Table 4.4: Influenza specific model parameters used in the compartmental models.

Variable	Value	Description	Reference
$\mathscr{R}_0$	1.46	Basic reproduction number	Tuite et al. [2010]
p_infect	0.0288	Pr. of infection	calculated
$p\_infect\_asymptom.$	0.0144	Pr. of asymptom. infect.	calculated Longini Jr et al. [2004] Ferguson et al. [2006]
p_remove	0.1429	Pr. of removal	Biggerstaff et al. [2014]
p_symptoms	0.25	Pr. of becoming sympt.	Biggerstaff et al. [2014]
p_exposed	0.01	Pr. of initially exposed	n/a
RRR	0.5 to 0.95	Vaccine efficacy	n/a
p_vacc	0.001 to 0.01	Pr. of getting vaccinated	n/a
p_vacc_init	0 to 0.8	Pr. of initial vaccination	n/a
p_quarantine	0 to 1	Pr. of rewiring	n/a

4.7 Running the models

Having set up the networks and models including a parameterisation for COVID-19 and influenza, the simulations are ready to be run. *epydemic* provides a straightforward API to run any of the compartmental models with a single parameterisation. However, since the simulations are stochastic, each run gives slightly different results, which is why it is best practice to run several repetitions of the experiments to gauge the range of possible outcomes.

Given the number of parameters and resulting possible parameter combinations, we rely on *epyc*, an experiment management framework written in Python [Dobson, 2016]. *epydemic* is actually built on top of *epyc*, which facilitates the integration. The reason for using a more sophisticated experiment management system rather than merely setting up the simulations manually is threefold. Firstly, *epyc* takes care of collecting the simulation results in a unified way in the same format for all model configurations. Secondly, since the library implements parallel computing (and cluster computing, if required), we can speed up the time-intensive simulations drastically, without having to implement parallelism ourselves. Lastly, *epyc* makes

sure to produce all possible combinations of parameter settings, which reduces the amount of boilerplate code and therefore reduces the potential for implementation mistakes.

To keep the computation times within reason while still allowing several experiment repetitions to gauge the uncertainty of the model results, we set the number of repetitions per experiment to 10. For the COVID-19 simulations, we run all possible combinations, that is all six networks (baseline and restricted for Mobility, PLC, and Distanced), all models (SEIR, SEIR_Q, SEIVR, and SEIVR_Q) and all parameter combinations. For influenza, we reduce the load by two thirds by only simulating on the two mobility networks. For COVID-19, this leaves us with $30,360$ total simulation runs and $10,120$ runs for influenza. Break-downs of these numbers are provided in Tables 4.5 and 4.6.

Table 4.5: Number of simulation runs for COVID-19.

Model	Networks	Parameter combinations	Repetitions	Total
SEIR	6	1	10	60
SEIR_Q	6	5	10	300
SEIVR	6	100	10	6000
SEIVR_Q	6	400	10	24000

Table 4.6: Number of simulation runs for influenza.

Model	Networks	Parameter combinations	Repetitions	Total
SEIR	2	1	10	20
SEIR_Q	2	5	10	100
SEIVR	2	100	10	2000
SEIVR_Q	2	400	10	8000

5 Validation of the Simulation Results

This section focuses on validating the results of the simulations against empirical data of the COVID-19 pandemic. All validation is performed using the two mobility networks.

Unless otherwise stated, the case numbers mentioned in the following sections are obtained from the CDC [Centers for Disease Control and Prevention, 2021]. The population numbers for each US state were obtained from the US Census Bureau [US Census Bureau, 2021].

5.1 SEIR and SEIR with Quarantine

In the following sections, we propose our process for validating the SEIR and SEIR_Q models and present the results.

5.1.1 Process

For the validation process, we ought to consider several aspects.

1. **Regions:** Since the network model and $\mathscr{R}_0$ in the epidemic models are based on data from New Haven County, CT, USA, we use case data from Connecticut as one point of comparison. However, since the models are fairly general and the parameterisation should apply reasonably well to other regions (at least within the US), we evaluate how the simulation results compare to the case data of all 50 states of the US.

2. **Time frame:** Our model assumes that a given population is initially infected at the start of the epidemic. Given the size of the network of $N = 10000$, this initial ratio generally should be at least about 0.001 to prevent the disease from dying out immediately. To compare the empirical case data with our simulation results, we define the start of the epidemic for each comparison data set as the day at which the total case count relative to the population of the region is larger or equal to 0.001 for the first time. For most US states, this is the case somewhere between the end of March 2020 and the end of April 2020. Since the United States experienced further epidemic waves starting in late 2020, our model is expected to break down at the start of this second wave. Hence, we set a time frame of 120 days after the outbreak of the first wave.

3. **Measure:** Our primary measure by which we compare simulation and real-world data is the cumulative case count. In our models, this equates to the occupancy of the exposed, infected, and removed compartments at any given time step. Secondly, we also compare the new cases per day, which in SEIR based models is equal to the change in the susceptible compartment. The daily new cases are much harder to compare exactly to real-world data due to their fluctuation. As a result, the range of the daily cases matters more than the value on any particular day.

For the start of the pandemic, the SEIR and SEIR with Quarantine models are most appropriate. Over our validation time frame of 120 days, no vaccinations were available rendering the SEIVR models inappropriate in this case. For the network, we consider both the baseline as well as the restricted mobility network.

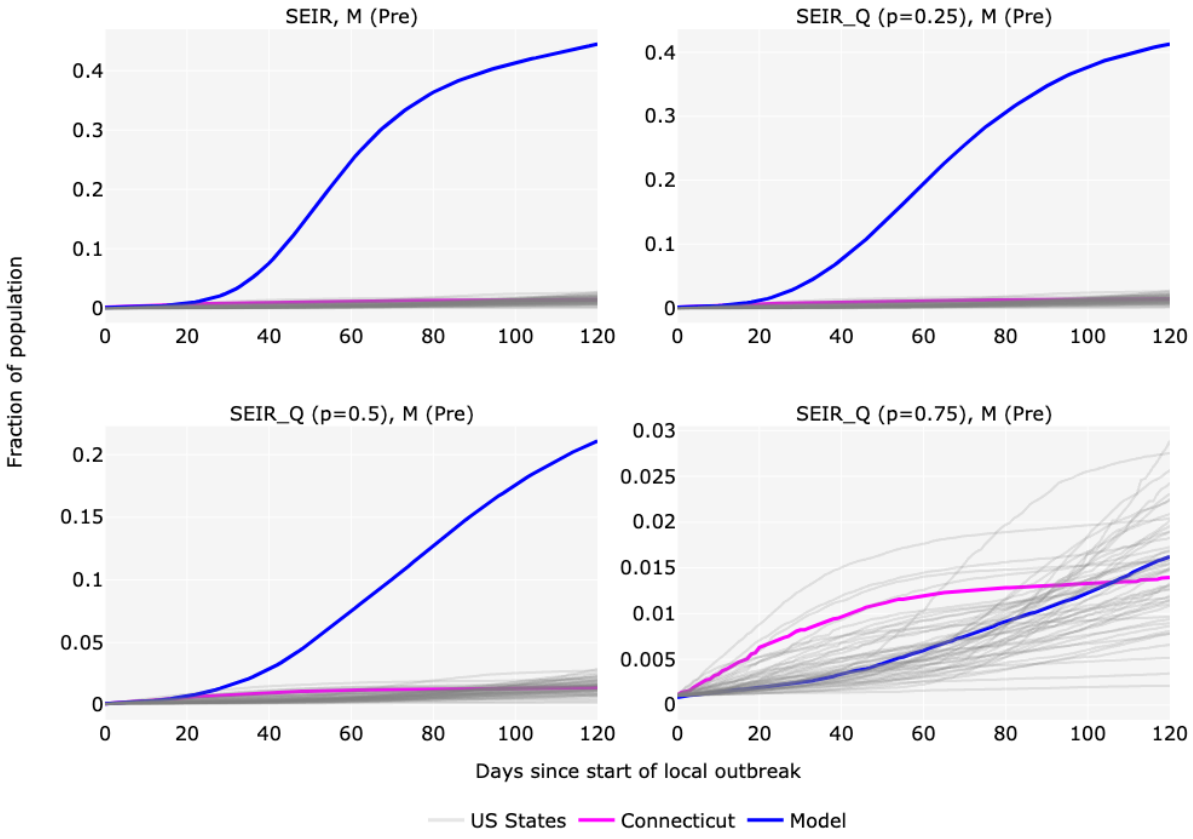
5.1.2 Results

Figure 5.1 contains four plots of the cumulative total case count of all 50 states (grey opaque lines; Connecticut is the magenta line) as well as the simulation result (blue line). The simulations use the baseline mobility network combined with the SEIR model, as well as three configurations of the SEIR with Quarantine ($p\_quarantine$ of 0.25, 0.5, and 0.75, respectively). The SEIR model as well as the two SEIR_Q models with the $p\_quarantine$ of 0.25 and 0.5 drastically overestimate the epidemic outbreak by one to two orders of magnitude. Only if we assume $p\_quarantine$ of 0.75 do we get a result that corresponds approximately to the mean empirical epidemic curve. This result would correspond to a situation where individuals largely maintained their pre-pandemic mobility patterns but, once symptomatic, were shielded from three quarters of their contacts.

A similar result is visible in Appendix B.1: for the SEIR with Quarantine and $p\_quarantine = 0.75$, the simulated daily new case counts approximately track the mean case counts in all states, while all other models drastically overestimate them.

Looking at the total case counts of the same models but run on the reduced mobility network gives different results (see Figure 5.2). Here, the SEIR model is actually within a reasonable range of the empirical data, albeit still comparatively high. Setting $p\_quarantine = 0.25$ in the SEIR_Q model now yields the best fit and the simulation results approximately follow the mean empirical case counts. Further increasing the quarantine parameter is too restrictive in that it suppresses the epidemic significantly more than was empirically the case (with the exception of a few lucky states with relatively low case counts during the first wave).

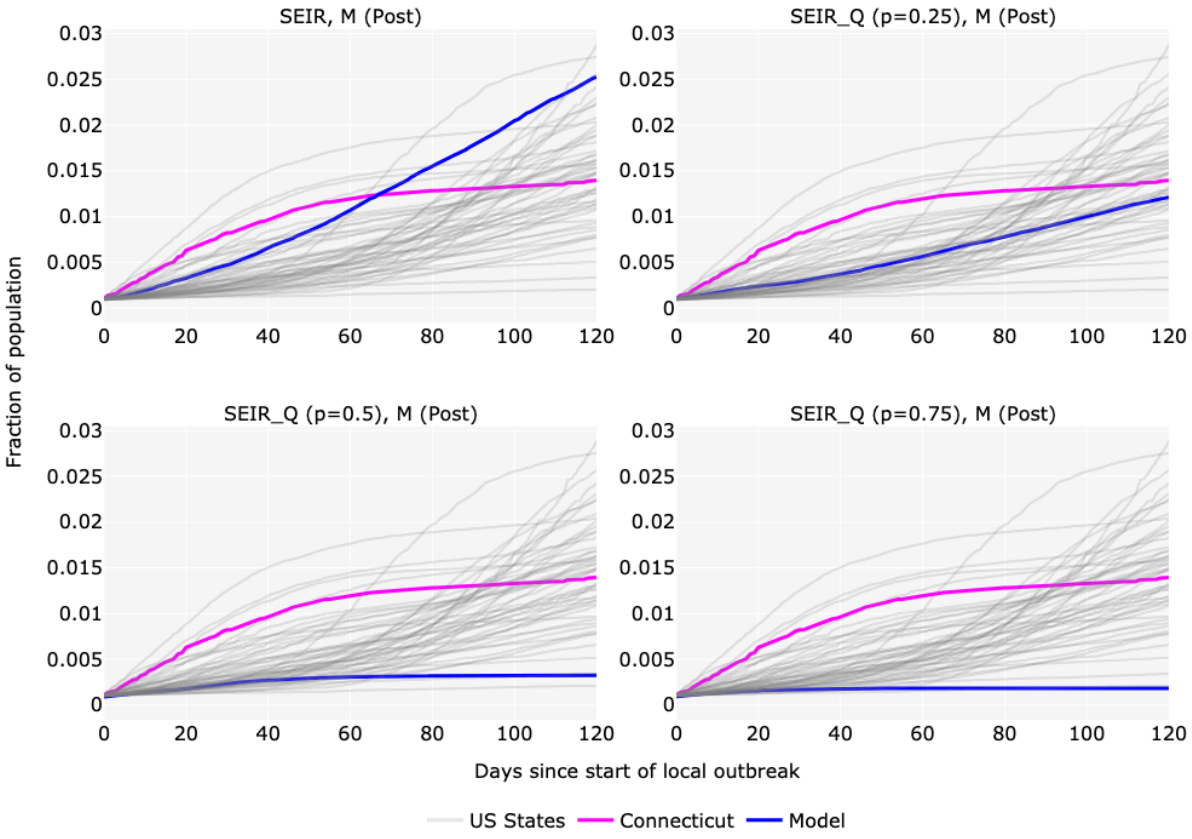
Figure 5.1: Cumulative total case counts with SEIR / SEIR_Q models run on the **baseline** mobility network ("Pre"). Start of local outbreak defined as first day where total case count $\geq 0.1\%$ of the population.



The daily new cases depicted in Appendix B.2 underline these results. While the daily cases of the SEIR model are within a reasonable range, since they are constantly on the higher end, the total case counts quickly accumulate. The contrary happens for the SEIR_Q with $p\_quarantine$ of 0.5 and 0.75, where a constantly low daily case count accumulates much slower than is the case for the average state. In other terms, this scenario would correspond to a population where the mobility is reduced compared to the baseline, in addition to infected individuals shielding from one quarter of their contacts.

Alas, which is the correct assumption: baseline mobility with strict quarantine or reduced mobility with lax quarantine? A universal answer to this question is not possible, since each state had drastically different responses to the outbreak of the pandemic and implemented countermeasures that varied greatly in their severity. Merely looking at the empirical data and the range of the total case counts after 120 days supports this. While in Montana, close to 2.9% of the population had been infected at that point, Vermont reported only about 0.21%. This also gives an explanation

Figure 5.2: Cumulative total case counts with SEIR / SEIR_Q models run on the **restricted** mobility network ("Post"). Start of local outbreak defined as first day where total case count $\geq 0.1\%$ of the population.



as to why, even though our mobility network and model parameterisation was based on data from New Haven County, CT, the model fit for that state was actually comparatively poor. (Consider the highlighted magenta line in the figures, which represents Connecticut). While unfortunate, this is hard to avoid with general models such as the one at hand and underlines that all of the simulation results produced by the models are more suitable for comparing epidemic outbreaks under different scenarios in general, rather than predicting exact case numbers.

5.2 SEIVR and SEIVR with Quarantine

In the following sections, we propose an adapted process for validating the SEIVR and SEIVR_Q models and present the results.

5.2.1 Process

Validating models that implement vaccination comes with several caveats. Firstly, the speed with which vaccinations were administered varies greatly between states, for reasons of infrastructure, policy, vaccine hesitancy, and many others. This issue is aggravated by the fact that the different states had diverged significantly in their case numbers by the time vaccination started (see for example the large spread of case counts across states in Figure 5.2). This would require to simulate a separate model with an entirely different parameterisation for each state, which is very time-intensive, both computationally as well manually. Secondly, even if we were to model each state individually, the vaccination rate within any given state was most likely not constant over time, something that is not allowed for in our model. Lastly, since a significant proportion of the population had already had an infection or was currently infected, starting a simulation in the middle of the pandemic would require us to move these individuals directly into the removed and infected compartments, which is is possible but not standard in the models.

Despite all these reservations, we developed a process that attempts to cope with these issues to still give us some idea of how valid the model results are. The following assumptions were made.

1. **Region:** Since validating the results for several states is prohibitively time intensive, we validate our models only on data from Connecticut, as this is the state the mobility data of the networks is from.

2. **Time frame:** The vaccination programme in Connecticut started in late December 2020, with very few vaccinations given in that month. We chose January 1, 2021, as the start of our validation time frame. As before, the length of the time frame is 120 days.

3. **Vaccinations:** Our model assumes that individuals only receive one vaccination dose, while in reality, most COVID-19 vaccinations require two doses. To handle this, we make the plausible assumption that one dose still provides a reduction in risk, albeit smaller. We, therefore, treat single doses as full vaccinations but choose a relatively low efficacy of 50%. From the start of the vaccination program in Connecticut to the day when half the population had received at least one vaccine, it took 110 days [Johns Hopkins University, 2021]. Therefore, we estimate the vaccination rate to be $1/110 \approx 0.0091$.

4. **Initial compartments:** Since our validation time frame starts in the middle of the pandemic, we need to consider the previous disease progression in our model. Consequently, we assume that as of January 1, 2021, 1.54% of the population had already received at least one dose [Johns Hopkins University, 2021]. On the same day, 2.05% of the population was classified

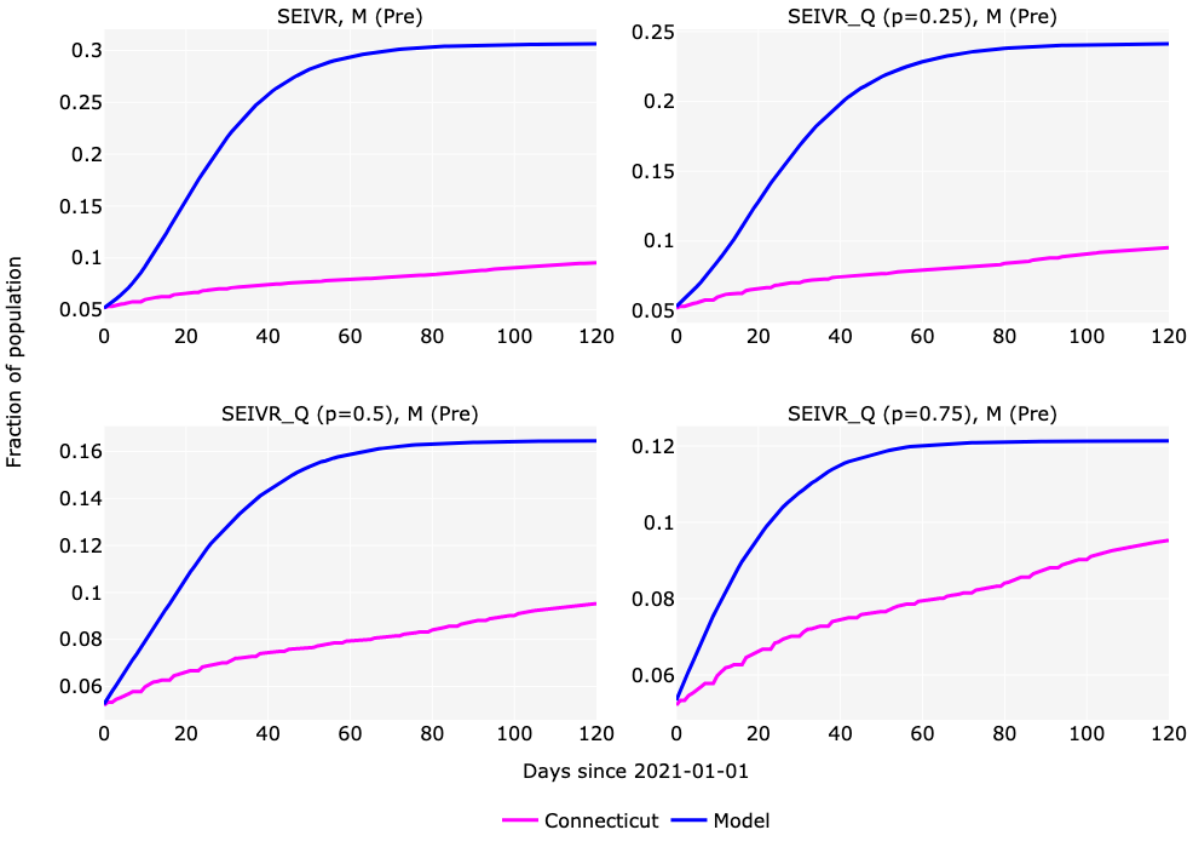
as active cases in the state [Worldometer, 2021]. We assume that one third of these were still asymptomatic and two thirds symptomatic, therefore setting the initial occupancy of the exposed compartment to 0.0068 and the infected compartment to 0.0137. Considering the total cumulative case count at this point minus the active cases, we set the initial occupancy of the removed compartment to 0.0316.

5. **Measure:** The cumulative total case counts are used for validation.

5.2.2 Results

Figure 5.3 depicts the simulation results using the baseline mobility network. Similar to what we observed for the models without vaccinations, all models appear to overestimate the number of cases quite drastically. Again, the model assuming a quarantine probability of 0.75 provides the best fit, although the case growth is still clearly too steep in the first 30 days of the simulation.

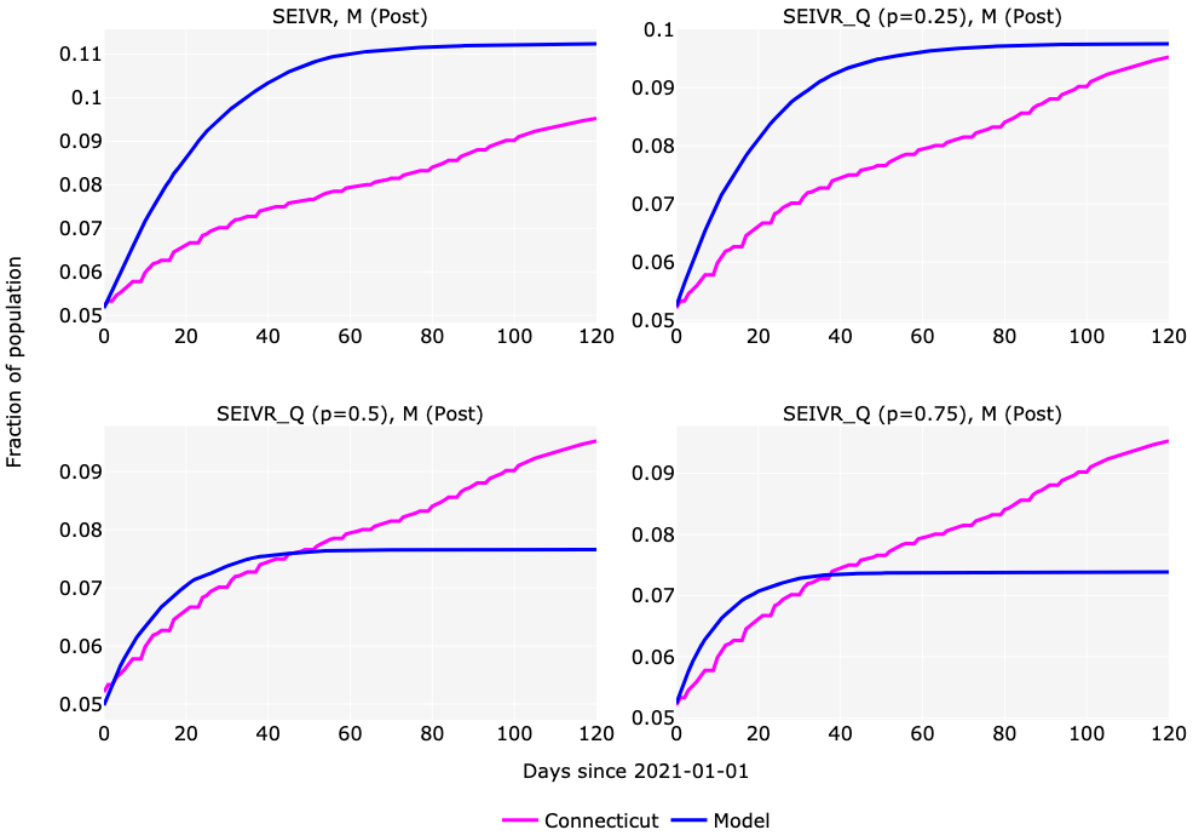
Figure 5.3: Cumulative total case counts with SEIVR / SEIVR_Q models run on the **baseline** mobility network ("Pre").



In Figure 5.4, the simulation results of running the models on the restricted network are depicted. While we see an overall increase in the fit of all model configurations using this network, it appears that the initial growth in cases is always significantly steeper than is the case empirically. Only setting the quarantine probability to 0.25 leads to a cumulative total case count that is comparable to that of the empirical data at the end of the validation period, although most of the cases in the model accumulate in the first 30 days.

The reason for this generally too steep case growth in the model initially could lie in an underestimation of the vaccine efficacy. As discussed, we assume this to be only 50% since we needed to accommodate for assuming one-dose vaccinations. Hence, increasing the vaccine efficacy would potentially be appropriate. In either case should the validation results for these models be taken with consideration due to the uncertainty about many of the assumptions we had to make during validation and the fact that we limited validation to only one state.

Figure 5.4: Cumulative total case counts with SEIVR / SEIVR_Q models run on the **restricted** mobility network ("Post").



6 Development of the Web Application

As the final part of this project, a web application is implemented that allows users to analyse the simulation results of the epidemic models. The goal of this is to open up the use of epidemic models built with *epydemic* [Dobson, 2017b] to a broader audience by taking away the need to write code on the part of the end-user. Of course, presenting the model results in a web application limits the user in the range of analysis of the models to some extent as it is restricted by the functionality provided by the interface of the application. As a result, the purpose of the web application is not to replace the programmatic framework the models are built with but rather to extend its usability by presenting the most important results in an accessible manner together with visualisations to support the analysis.

The implemented application gives the user access to all models (SEIR and SEIVR, with and without quarantine extension) run on all networks (baseline and restricted for Mobility, PLC, and Distanced). However, running these simulations takes more time than any user would deem acceptable on an interactive website. Fortunately, we can avoid this problem by pre-computing all possible parameter settings for all simulations and store them in a large data set. Then, whenever the user requests a certain setting, we can simply filter this data set and return the result almost instantly.

In the following sections, we provide an overview of the technologies used for the development of the application, the design of the front end, the functionality, and finally a discussion of the actual implementation.

6.1 Technologies

There is a range of tools and frameworks for the development of data-driven web applications but we decided to stay in the Python environment and use *Plotly Dash* as the backbone of the application [Plotly Technologies Inc., 2021a]. *Dash* was built to facilitate the development of data science dashboards and allows the developer to set up the entire web page in Python, which is then rendered using the JavaScript frameworks *Plotly.js* and *React.js*, while on the server-side, *Plotly Dash* is built around the Python micro web framework *Flask*.

Dash is quite extendable in the sense that there exist a range of libraries that allow the integration of common web tools and packages into the *Dash* application. Particularly, we use *Plotly Express* to create the figures on the dashboard [Plotly Technologies Inc., 2021b]. We also make extensive

use of the *Dash Bootstrap Components* extension for both layout and styling [Faculty AI, 2021], in addition to custom CSS stylesheets.

While *Flask* comes with a simple web server that is suitable for development and testing purposes, it is not recommended for use in production. We instead use *uWSGI* as the application server [Unbit, 2021] and *Nginx* as the web server [Nginx Inc., 2021].

Lastly, we use *Docker* to containerise both the *Dash* app as well as the *Nginx* server to allow for a simple and straightforward deployment process. *Docker Compose* is used to orchestrate the two *Docker* containers [Docker Inc., 2021].

6.2 Front end design and functionality

The web application consists of one main dashboard on which the simulation results are visualised for analysis. Further, the user can access a page that presents the results of the validation of Section 5. Lastly, we provide three mostly text-based pages that give further information about the models and data used in the app, as well as the project in general.

6.2.1 Dashboard

The dashboard includes the following functionality.

- **Control bar:** Crucially, the user requires a way of requesting specific simulation results. This includes the disease, the type of model, the network, as well as possible parameterisations. The control bar also includes a button to allow the user to download the simulation results of the current parameterisation as a CSV file.

- **Epidemic curves:** The epidemic curves showing the progress of the sizes of the model compartments over time are at the center of the application. Further, the graph allows different forms of interactivity, such as zooming and filtering, and provides tooltip information about the data.

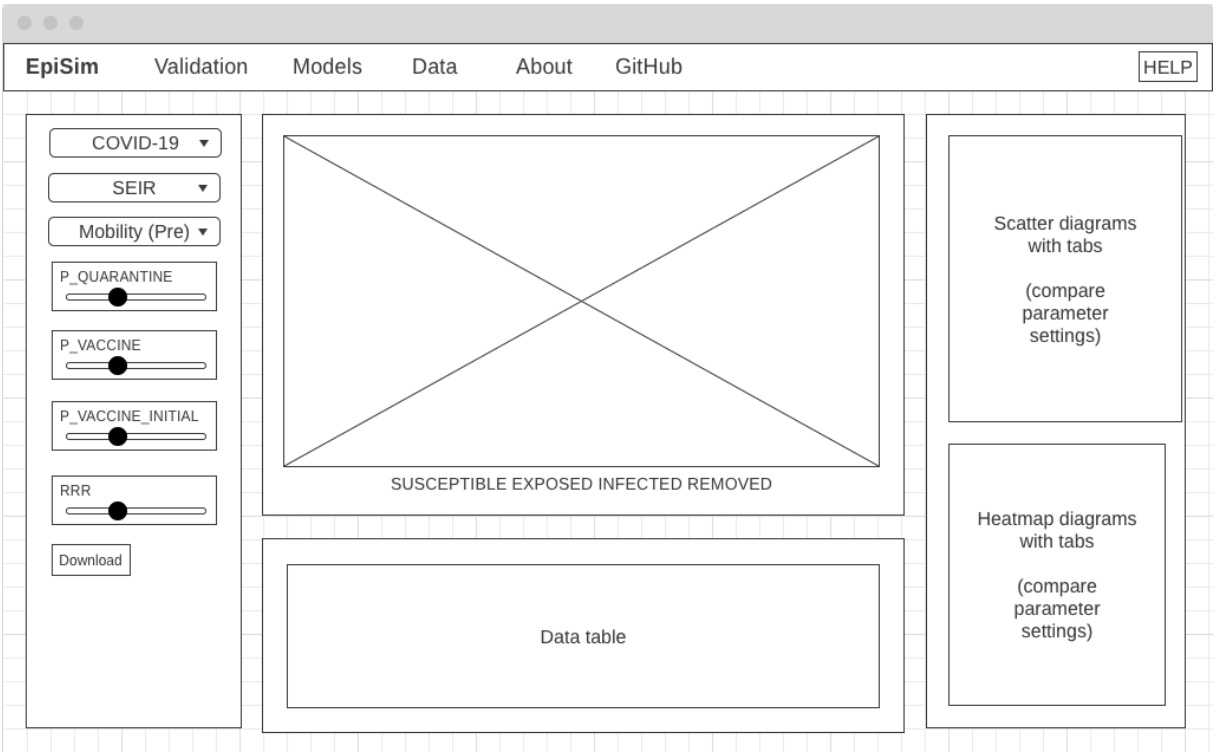
- **Comparison of parameterisations:** As the user can request different model parameterisations, it is helpful to compare the effect of different parameter settings on the epidemic size *ceteris paribus*. This is done (1) through a scatter plot that visualises the epidemic sizes per simulation run and parameter setting and (2) a heat map showing the mean epidemic size

per run and parameter setting and a comparison between different models that include the parameter.

- **Numeric output:** Some of the results of the epidemic simulation are best presented by a single number. These key results are summarised in a data table in the web application.
- **Help section:** To help the user realise all of the functionality provided by the web application, a help modal can be opened from the dashboard by clicking on a help button in the top right corner.

To provide all this functionality in a way that facilitates the analysis and allows the comparison of results without navigating around different pages and excessive scrolling, we chose a dashboard-type layout for the web application. For planning purposes, we created a layout draft which is provided in Figure 6.1. A screenshot of the dashboard is provided in Appendix C.1.

Figure 6.1: Layout draft of the web application.



6.2.2 Validation page

The validation page is included to provide the user with an easy way to gauge how the model results compare to empirical data. As such, the page is very similar to Section 5 and presents the same graphs as were discussed there. The user can select one of the provided model configurations from a control bar and can compare it to empirical case data from the United States. Next to the graph output, we further include a short descriptive text. A screenshot of the validation page is provided in Appendix C.4.

6.2.3 Text pages

Lastly, the user can access three pages with further information about the models, the networks, as well as the project in general.

The model page provides some insight into compartmental models, and the specifics of the models implemented in the application. The networks, their characteristics, and the algorithms used to create them are discussed on the networks page. We also provide references and information about the mobility data used for the creation of the networks. The about page gives a brief description of this project. Screenshots of the text pages are provided in Appendices C.2, C.3, and C.5.

6.3 Implementation

The implementation of the web application can be split into several parts, as presented in the following sections.

6.3.1 Simulations and data storage

The data available in the application corresponds to the results of the simulations described in Section 4.7. We set up a small pipeline from running the simulations to using the data in the application. The simulation results collected by *epyc* are saved in JSON files before being converted into the data format required by our application. Specifically, the occupancy numbers of each compartment are extracted and converted into a long data format data frame before being serialised as a pickled file. The reason for this is that we use *pandas* [Pandas developers, 2021] for filtering during data processing and pickled files are a fast way of reading in data frames.

While we could store all this data locally, we chose to instead store it in a public data repository on the version control hosting platform GitHub [GitHub Inc., 2021]. This gives us remote access to the files in a convenient manner, as we can simply read in the data from the public URL. Further, it facilitates the uploading process as well, since we can simply use the official GitHub REST API. The downside of this is that we can only upload files of up to 100 MB, but in our case this is sufficient.

The results presented on the validation page have a much smaller scope as we only ran a few specific parameter configurations rather than all possible ones. We refer to Section 5.1.1 for information on which data was used for validation. As with the main simulations the data was stored in the same data repository on GitHub in the same format as pickled files.

6.3.2 Data import and processing

The pickled data files stored on GitHub are downloaded and deserialised upon starting the server. While this only takes a few seconds, for performance purposes the pickled files are also saved locally after the initial download so they can be read from disk in case of a server restart. However, we provide a script to clear these cached files to force a re-download in case of an update etc.

While the server is running, the data is stored in memory as an attribute of an instance of the *SimulationData* class. The class provides all the functionality related to processing the simulation results, especially returning subsets of the data depending on which filters are applied by the user. It further provides the methods to calculate the numeric simulation results included in the data table of the application.

Since the import of the validation results is much less time-intensive than the full simulation results, the data is not stored locally, but rather downloaded from the data repository and deserialised every time the server is started. The downloaded data is stored in an attribute of instances of the classes *EmpiricalData* and *ModelledData*, respectively. During the initialisation of the instances, the data is either processed and transformed or, if a pre-processed result is available in the data repository, downloaded and deserialised from there.

6.3.3 Interactivity and graphing

The web application provides interactivity via (1) the dropdowns and sliders in the control bar that let the user select model, network, and parameterisation, as well as (2) zooming, filtering,

and tooltip functions when interacting with the graphs.

The first of the two is achieved using *Dash* callbacks, which are functions that are called whenever a certain event is triggered on the front end, e.g. the user changing the value of a slider. Whenever a user triggers one such event, the state of all elements of the control bar is queried, converted into a dictionary and passed to the *subset_data* method of the *SimulationData* function to apply the filters and return the appropriate data frame slice. The slice is then passed to several functions responsible for creating the figures visible to the user. Likewise, the numeric simulation results are calculated and the *Dash* table component is returned.

The interactivity of the figures is almost entirely provided out of the box by *Dash* (or rather *Plotly*, which is used for graphing). For the main graph, a scatter plot of the results from all simulations (i.e. ten data points per time step per compartment) is combined with a line graph that gives the mean of the observations per time step. This way, one can estimate the uncertainty involved with the modelling results as a larger horizontal scatter of the points of one time step would show that the simulation runs were very different from each other, indicating higher uncertainty about the results (see Figure C.6).

Uncertainty is communicated in a similar way in the scatter plot that depicts the epidemic size for different parameter settings. Again, more horizontal scatter suggests uncertainty. An example for the $p\_vaccinated$ parameter is given in Figure C.7.

On the validation page, the user selection of the model configuration is also handled using *Dash* callbacks. The desired configuration is retrieved from the *ModelledData* instance and together with the empirical case data of the *EmpiricalData* instance used as input for two graphs; one showing the total cumulative case counts and another depicting the daily new cases (for the SEIR based models). Both graphs allow for the same user interactions as the main line graph on the dashboard.

6.3.4 Deployment

Dash is built on top of the Python microframework *Flask* that comes with a built-in lightweight web server, which according to the *Flask* documentation is not recommended for production use. As an alternative, it is fairly straightforward to set up an application server such as *uWSGI* and a web server such as *Nginx* to provide a more robust production environment including multi-threading and more advanced security features.

To facilitate the deployment of the web application to production including the *Nginx* web server,

we chose the containerisation tool *Docker*. *Docker Compose* is used to orchestrate the two containers and run them together. This setup allows us to build the full application on another machine as long as it runs *Docker* while assuring that all dependencies are respected.

6.3.5 Testing

Testing of the application was done in several ways. Unit tests were performed to assert that critical and complicated pieces of code worked as expected. This mostly involved the data transformation process of the simulation results as well as the validation to a large extent.

Secondly, we performed manual testing of the application in three different browsers to assure that the website is displayed correctly and the functionality works as expected on different systems. The application was tested in the browsers Brave (v1.26.74), Google Chrome (v86.0.4240.198), and Safari (v14.1.2). The application worked identically on all three browsers and no issues were discovered.

Lastly, two continuous integration workflows were set up using GitHub Actions that are run automatically whenever commits are pushed to the repository hosted on GitHub. The first workflow performs semantic checks of the code, runs all unit tests and creates a test coverage report for all Python versions supported by the project. The second workflow builds the containers of the application using *Docker Compose* both in a development and a production configuration. The implementation of these automated testing tools provided a powerful way of quickly spotting errors as they arose.

7 Evaluation and critical appraisal

In Section 1.2 we provided primary and secondary objectives regarding the outcome of this project. In this section, the extent to which these objectives were reached is evaluated through self-assessment.

7.1 Primary objectives

The primary objectives represent the cornerstones of the project and fulfilling them is required for the success of the project. All three primary objectives are fully completed, although the second objective was slightly adapted to align it with the other objectives of the project:

1. **Develop a human contact network based on mobility data**
   We developed a process to extract mobility data from the SafeGraph Patterns data set and used it to build a human contact network based on empirically observed mobility patterns. We also modelled the reduction in mobility after countermeasures against the disease outbreak were taken by implementing a second mobility network that captures this reduced mobility.

2. **Develop a compartmental model for COVID-19 using the epidemic modelling framework *epydemic***
   We implemented both an SEIR and an SEIVR compartmental model, as well as a quarantine extension for each. The models were parameterised to encapsulate the characteristics of COVID-19. The reason we adapted this goal to include two more general variants of models rather than an overly COVID-19 specific one, lies in how this objective relates to the other objectives of the project. The third primary objective outlines an application that is readily extendable to other diseases, which would not be possible were the epidemic model too specific to only COVID-19. In summary, while we slightly adapted this objective, it can be considered fully complete.

3. **Implement a web application allowing the use of the epidemic model in a user-friendly, yet sufficiently flexible manner**
   The web application developed in this project fulfils the objective of giving the user a way to explore the results of epidemic simulations, including custom parameterisations and different human contact networks, making it fairly flexible. Further, the user is not required to programmatically implement any of the steps involved, which opens up the use for a broad

range of users. Hence, this objective can be considered fully complete.

7.2 Secondary objectives

The secondary objectives outline those elements of the project that take a lower priority compared to the primary objectives, but whose completion would further benefit the overall success of the project. All secondary goals and their completion status are listed below.

1. **Survey of existing models of COVID-19**
   This objective was completed in Section 2.2, where we provided an extensive review of existing models of COVID-19. This included an overview of the different types of models depending on the goal they try to achieve.

2. **Validate the human contact network and compartmental model by comparing its results to empirical data**
   In Section 3.2 the mobility networks were validated by computing several network metrics and analysed with respect to how they relate to the empirical reduction in mobility observed during the same period. We observed that this reduction was captured well, a result that was confirmed in Section 5.1, where the reduced networks appeared to provide a more appropriate fit to empirical case data from the United States under an SEIR model. In the same section, we also established that with the correct parameterisation the SEIR_Q model produces results that are comparable to empirical data. Unfortunately, the extent of validation of models incorporating vaccination was limited due to the lack of empirical data unbiased by the vast range of uncontrollable factors by the time vaccines for COVID-19 were available. An attempt at validation of the models was made nonetheless in Section 5.2.

3. **Extend the web application to other infectious diseases**
   The extension of the models and the web application to include another disease was successful. We applied a separate parameterisation for influenza to the models, ran the simulations and included the results into the application. This demonstrated the straightforward process with which such an extension is possible.

8 Conclusion

In this project, we developed different versions of simulation-based compartmental models for COVID-19 running on a mobility-based human contact network and integrated them into a web application that provides the user with an accessible way of working with the simulation results.

We successfully developed the network model from mobility data collected in New Haven County, CT, USA, in February and March 2020 on which we could run the epidemic models. These include four variations of models (SEIR, SEIVR, and quarantine extensions for each) with a parameterisation of COVID-19. All networks and models were successfully integrated into a web application.

Despite the general success of the project, the validation of the models against empirical data underlined some limitations of the models. While the models produced reasonable results given an appropriate parameterisation, there were still discrepancies from the empirical data, especially over time periods longer than 50 days after the initial outbreak. This underlines the fact that the simulation results are suitable for a comparison of different epidemic outcomes under different parameterisations, models, and networks, but not so much for accurate predictions. For this, one would likely have to implement a significantly more complicated model.

Some future work is possible to further improve on the advantages of the web application.

- **Extension to further diseases:** As illustrated by the example of influenza, extending the models to other diseases and integrating the results into the web app is straightforward. Future work could make use of this and integrate further infectious diseases of interest into the app.

- **More flexible parameterisation:** Alternatively (or in addition) to extending the application by providing more disease presets, one could allow the user to define more of the parameters, such as the infection rate etc., themselves. Of course, this would come with the downside of having to pre-compute more simulations, which was shown to be time-intensive.

- **Addition of other mobility networks:** The currently used mobility network is based on New Haven County, CT, USA, which is a mostly urban area. As mobility patterns differ vastly between regions but also between rural, urban, and metropolitan areas, it would be interesting to add mobility networks for different areas. If also based on SafeGraph Patterns data, this would be fairly straightforward with the currently provided workflow.

- **More advanced data storage and processing:** For the amount of data that can be accessed

by the user in the current version of the web application, storing the data in a data repository and processing it by slicing fairly small data frames is sufficient. However, were we to include more data with more parameterisations etc., a more advanced data storage would be required to maintain the performance of the application and would aid maintainability. Storing the data in a database and working with the results of database queries seems most appropriate.

A Repositories and Deployment

The code of this project is available on GitHub:

`https://github.com/leo-pfeiffer/epi-sim`

The data repository can be found here:

`https://github.com/leo-pfeiffer/epi-sim-data`

The EpiSim web application is deployed here:

`http://epi-sim.live`

B Simulation Validation

Figure B.1: Daily new case counts with SEIR / SEIR_Q models run on the **baseline** mobility network ("Pre"). Start of local outbreak defined as first day where total case count $\geq 0.1\%$ of the population.

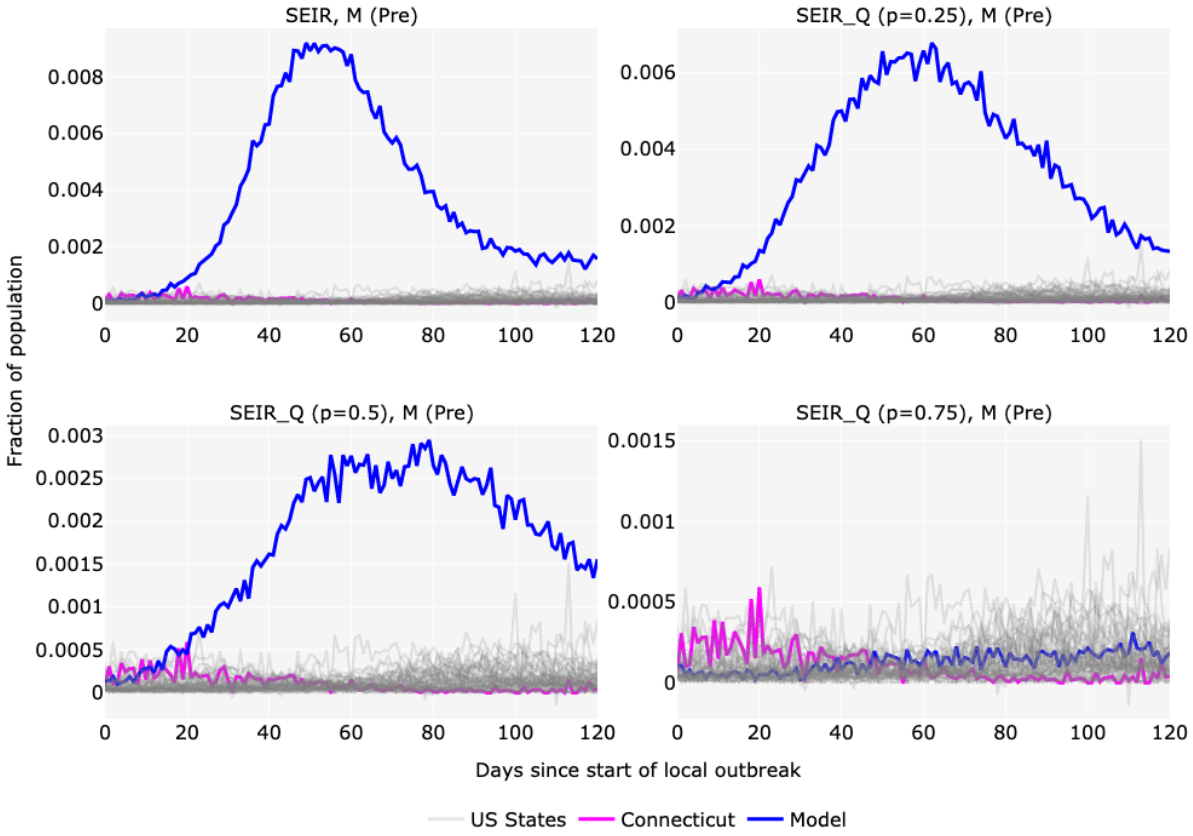
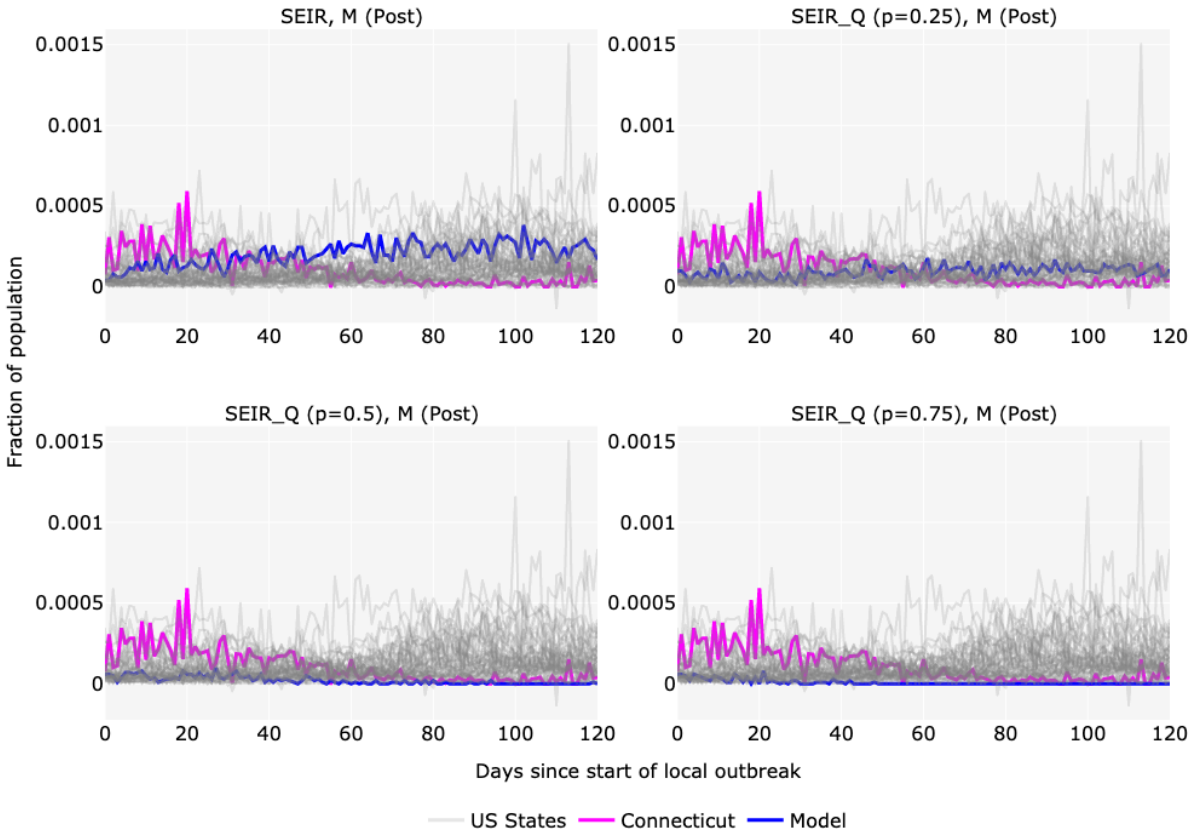


Figure B.2: Daily new case counts with SEIR / SEIR_Q models run on the **restricted** mobility network ("Post"). Start of local outbreak defined as first day where total case count $\geq 0.1\%$ of the population.



C Screenshots of the Web Application

Figure C.1: Screenshot of main page of *EpiSim*.

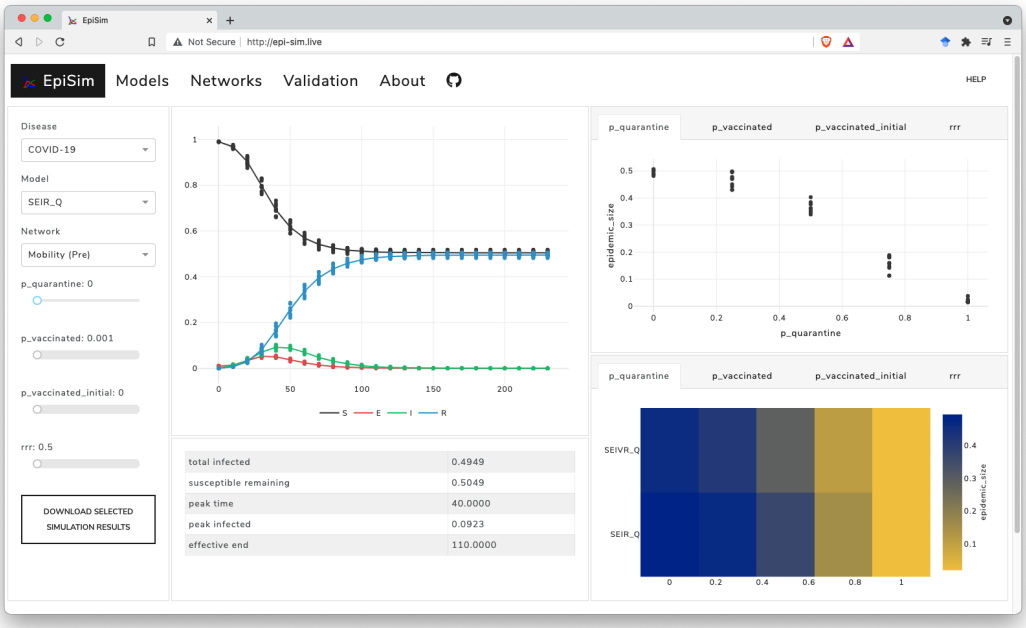


Figure C.2: Screenshot of models page of *EpiSim*.

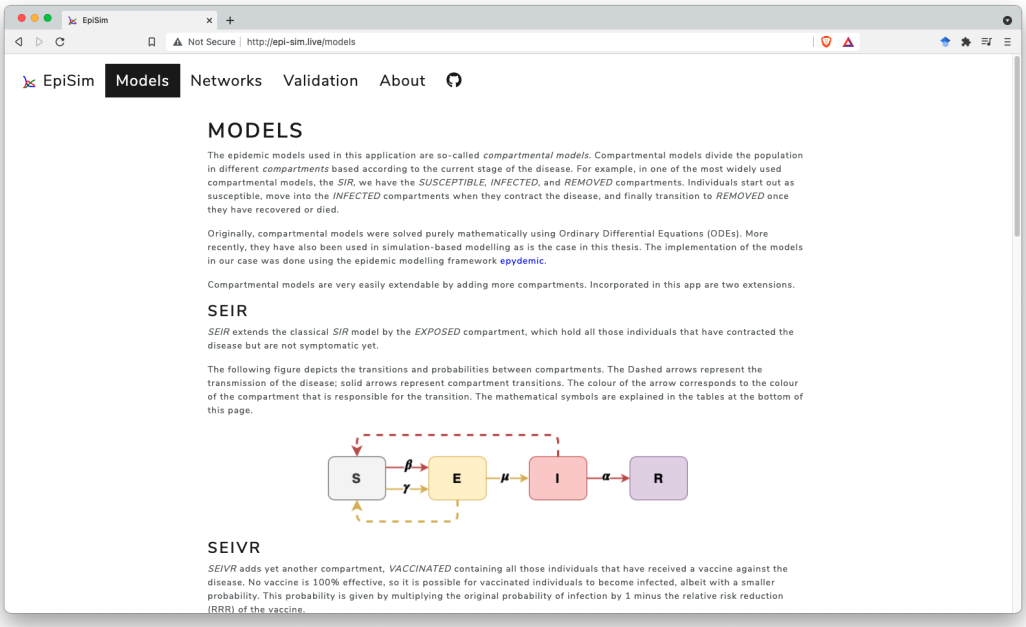


Figure C.3: Screenshot of networks page of *EpiSim*.

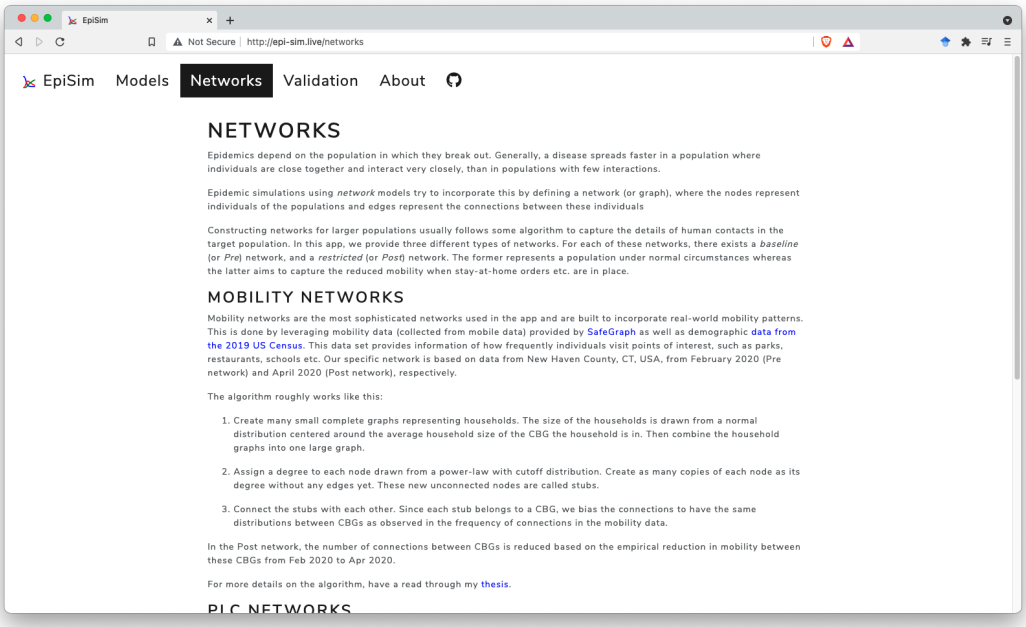


Figure C.4: Screenshot of validation page of *EpiSim*.

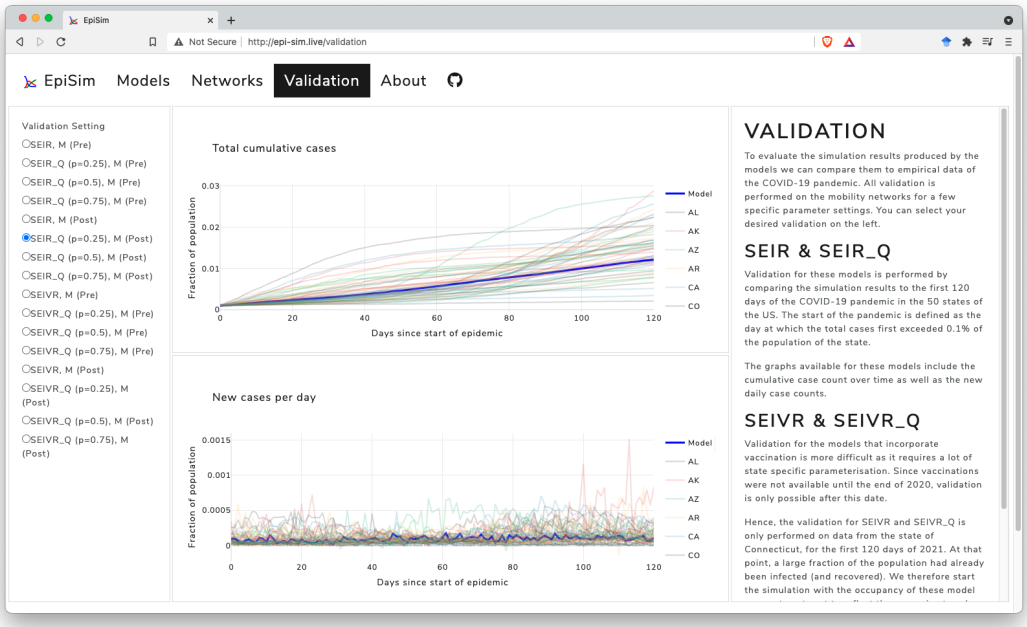


Figure C.5: Screenshot of about page of *EpiSim*.

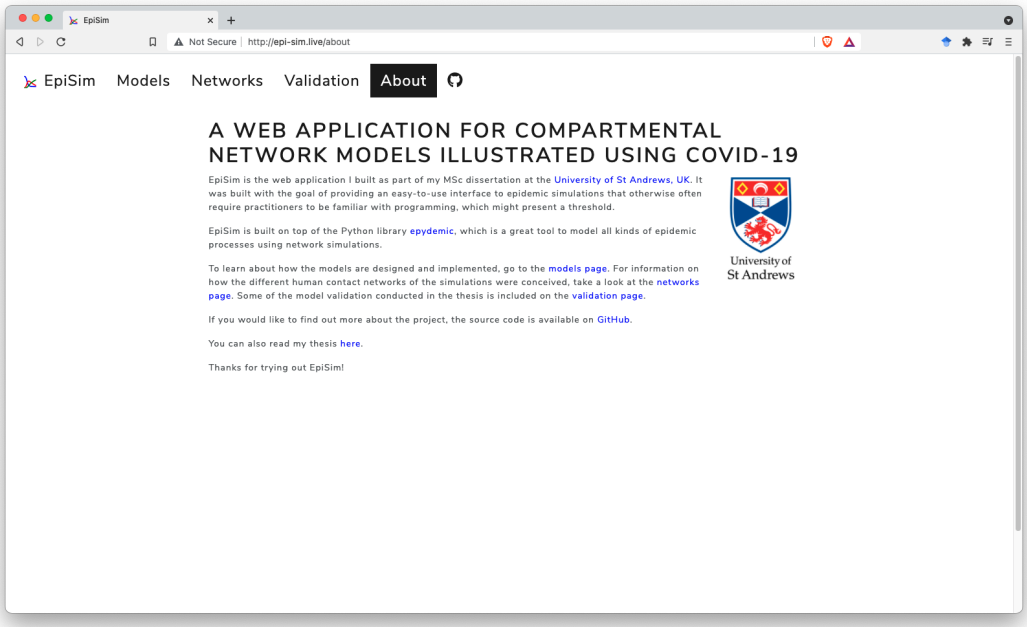


Figure C.6: Screenshot of the main graph of *EpiSim*.

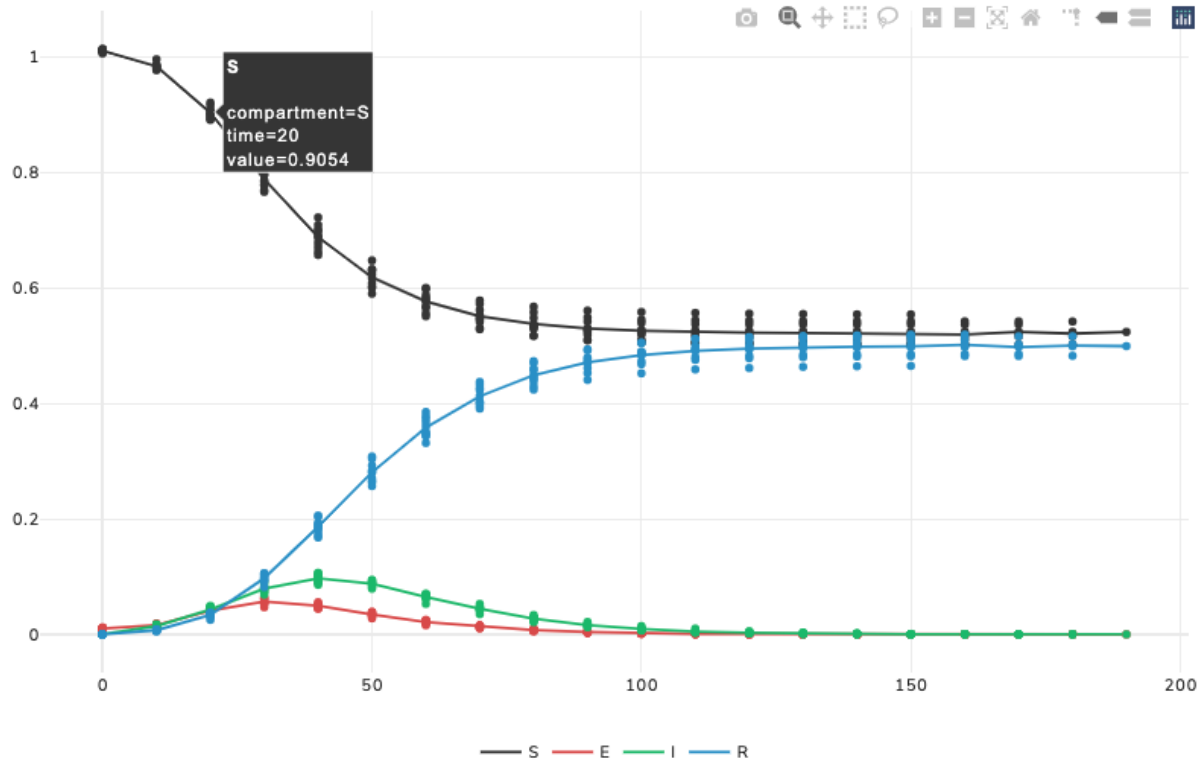


Figure C.7: Screenshot of the scatter plot showing the epidemic size for different settings of the P_-VACCINATED parameter in *EpiSim*.

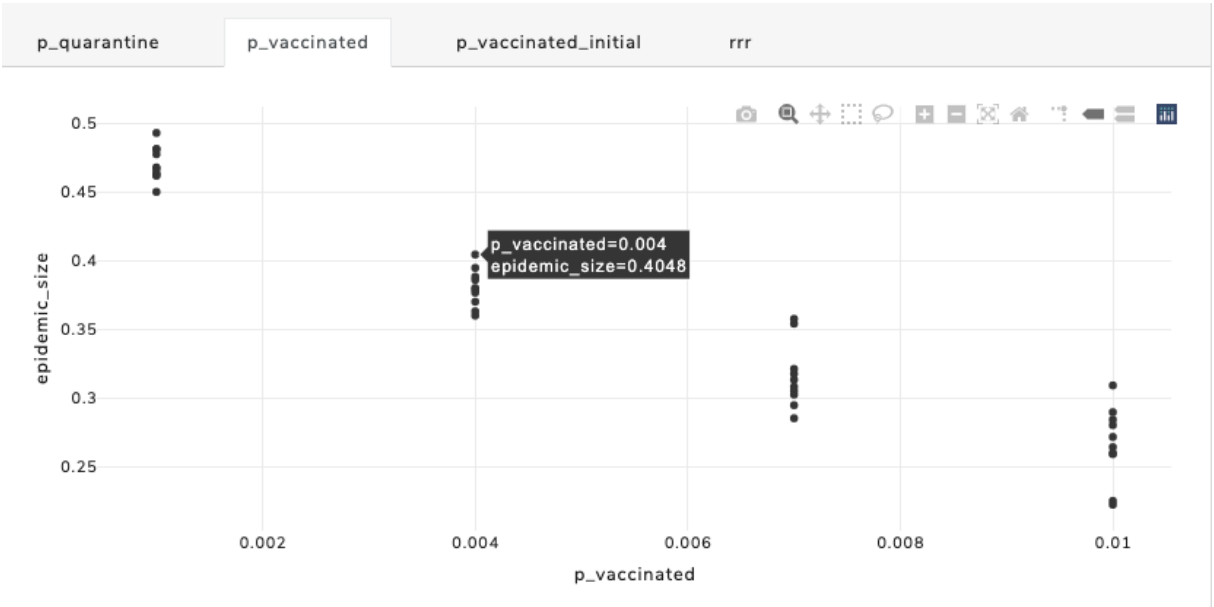
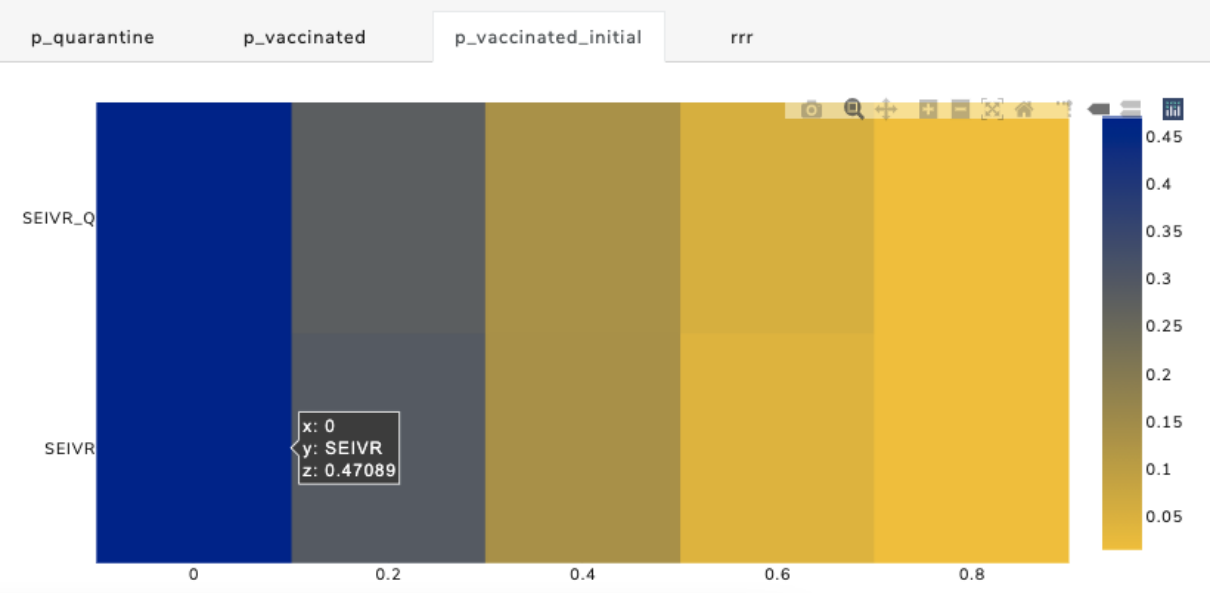


Figure C.8: Screenshot of the heat map showing the epidemic size for different settings of the P_-VACCINATED_INITIAL parameter for different models in *EpiSim*.



References

C. Anastassopoulou, L. Russo, A. Tsakris, and C. Siettos. Data-based analysis, modelling and forecasting of the covid-19 outbreak. *PloS one*, 15(3):e0230405, 2020.

R. M. Anderson and R. M. May. *Infectious diseases of humans: dynamics and control*. Oxford university press, 1992.

L. R. Baden, H. M. El Sahly, B. Essink, K. Kotloff, S. Frey, R. Novak, D. Diemert, S. A. Spector, N. Rouphael, C. B. Creech, et al. Efficacy and safety of the mrna-1273 sars-cov-2 vaccine. *New England Journal of Medicine*, 384(5):403–416, 2021.

M. Biggerstaff, S. Cauchemez, C. Reed, M. Gambhir, and L. Finelli. Estimates of the reproduction number for seasonal, pandemic, and zoonotic influenza: a systematic review of the literature. *BMC infectious diseases*, 14(1):1–20, 2014.

S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang. Complex networks: Structure and dynamics. *Physics reports*, 424(4-5):175–308, 2006.

F. Brauer and C. Castillo-Chavez. *Mathematical models in population biology and epidemiology*, volume 2. Springer, 2012.

Centers for Disease Control and Prevention. United States COVID-19 Cases and Deaths by State over Time. `https://data.cdc.gov/Case-Surveillance/United-States-COVID-19-Cases-and-Deaths-by-State-o/9mfq-cb36`, 2021. Accessed: 2021-07-30.

M. Cevik, M. Tate, O. Lloyd, A. E. Maraolo, J. Schafers, and A. Ho. Sars-cov-2, sars-cov, and mers-cov viral load dynamics, duration of viral shedding, and infectiousness: a systematic review and meta-analysis. *The Lancet Microbe*, 2020.

S. Chang, E. Pierson, P. W. Koh, J. Gerardin, B. Redbird, D. Grusky, and J. Leskovec. Mobility network models of covid-19 explain inequities and inform reopening. *Nature*, 589(7840): 82–87, 2021.

ct.gov. Emergency Orders issued by the Governor and State Agencies. `https://portal.ct.gov/Coronavirus/Pages/Emergency-Orders-issued-by-the-Governor-and-State-Agencies`, 2021. Accessed: 2021-06-15.

S. J. de Vlas and L. E. Coffeng. Achieving herd immunity against covid-19 at the country level by the exit strategy of a phased lift of control. *Scientific reports*, 11(1):1–7, 2021.

S. Dobson. epyc: Python computational experiment management. `https://github.com/simoninireland/epyc`, 2016.

S. Dobson. Documentation of epydemic: Epidemic simulations on networks in Python. `https://pyepydemic.readthedocs.io/en/latest/`, 2017a.

S. Dobson. epydemic: Epidemic simulations on networks in Python. `https://github.com/simoninireland/epydemic`, 2017b.

S. Dobson. *Epidemic Modelling: Some Notes, Maths, and Code*. Independent Publishing Network, 2020.

Docker Inc. Docker. `https://docs.docker.com/`, 2021.

A. W. Edridge, J. Kaczorowska, A. C. Hoste, M. Bakker, M. Klein, K. Loens, M. F. Jebbink, A. Matser, C. M. Kinsella, P. Rueda, et al. Seasonal coronavirus protective immunity is short-lasting. *Nature medicine*, 26(11):1691–1693, 2020.

Facebook Inc. Facebook Data for Good. Our work on COVID-19. `https://dataforgood.fb.com/docs/covid19/`, 2021. Accessed: 2021-06-16.

Faculty AI. Dash Bootstrap Components. `https://dash-bootstrap-components.opensource.faculty.ai/`, 2021.

U. FDA. Vaccines and Related Biological Products Advisory Committee meeting: FDA briefing document. `https://bit.ly/3xv05zA`, 2020. Accessed: 2021-07-05.

N. M. Ferguson, D. A. Cummings, C. Fraser, J. C. Cajka, P. C. Cooley, and D. S. Burke. Strategies for mitigating an influenza pandemic. *Nature*, 442(7101):448–452, 2006.

J. Friedman, P. Liu, C. E. Troeger, A. Carter, R. C. Reiner, R. M. Barber, J. Collins, S. S. Lim, D. M. Pigott, T. Vos, et al. Predictive performance of international covid-19 mortality forecasting models. *Nature communications*, 12(1):1–13, 2021.

L. G. Gallo, A. F. d. M. Oliveira, A. A. Abrahão, L. A. M. Sandoval, Y. R. A. Martins, M. Almirón, F. S. G. Dos Santos, W. N. Araújo, M. R. F. de Oliveira, and H. M. Peixoto. Ten epidemiological parameters of covid-19: use of rapid literature review to inform predictive models during the pandemic. *Frontiers in Public Health*, 8:830, 2020.

G. Giordano, F. Blanchini, R. Bruno, P. Colaneri, A. Di Filippo, A. Di Matteo, and M. Colaneri. Modelling the covid-19 epidemic and implementation of population-wide interventions in italy. *Nature medicine*, 26(6):855–860, 2020.

GitHub Inc. GitHub. `https://github.com`, 2021. Accessed: 2021-07-24.

Google LLC. Google COVID-19 Community Mobility Reports. `https://www.google.com/covid19/mobility`, 2021. Accessed: 2021-06-16.

T. J. Gordon. A simple agent model of an epidemic. *Technological Forecasting and Social Change*, 70(5):397–417, 2003.

H. W. Hethcote. The mathematics of infectious diseases. *SIAM review*, 42(4):599–653, 2000.

N. Hoertel, M. Blachier, C. Blanco, M. Olfson, M. Massetti, M. S. Rico, F. Limosin, and H. Leleu. A stochastic agent-based model of the sars-cov-2 epidemic in france. *Nature medicine*, 26(9): 1417–1421, 2020.

Institute for Health Metrics and Evaluation. COVID-19 estimation updates. `http://www.healthdata.org/covid/updates`, 2020.

Johns Hopkins University. Coronavirus Resource Center, United States, Connecticut. `https://coronavirus.jhu.edu/region/us/connecticut`, 2021. Accessed: 2021-07-30.

K. Klise, W. Beyeler, P. Finley, and M. Makvandi. Analysis of mobility data to build contact networks for covid-19. *Plos one*, 16(4):e0249726, 2021.

Y. Liu, A. A. Gayle, A. Wilder-Smith, and J. Rocklöv. The reproductive number of covid-19 is higher compared to sars coronavirus. *Journal of travel medicine*, 2020.

I. M. Longini Jr, M. E. Halloran, A. Nizam, and Y. Yang. Containing pandemic influenza with antiviral agents. *American journal of epidemiology*, 159(7):623–633, 2004.

Los Alamos National Laboratory COVID-19 Team. LANL COVID-19 Cases and Deaths Forecasts. `https://covid-19.bsvgateway.org/`, 2020.

D. McEvoy, C. McAloon, A. Collins, K. Hunt, F. Butler, A. Byrne, M. Casey-Bryars, A. Barber, J. Griffin, E. A. Lane, et al. Relative infectiousness of asymptomatic sars-cov-2 infected persons compared with symptomatic individuals: a rapid scoping review. *BMJ open*, 11(5): e042354, 2021.

M. T. Meehan, D. P. Rojas, A. I. Adekunle, O. A. Adegboye, J. M. Caldwell, E. Turek, B. Williams, J. M. Trauer, and E. S. McBryde. Modelling insights into the covid-19 pandemic. *Paediatric respiratory reviews*, 2020.

D. Meidan, N. Schulmann, R. Cohen, S. Haber, E. Yaniv, R. Sarid, and B. Barzel. Alternating quarantine for sustainable epidemic mitigation. *Nature communications*, 12(1):1–12, 2021.

MRC Centre for Global Infectious Disease Analysis. Imperial College COVID-19 LMIC Reports. `https://mrc-ide.github.io/global-lmic-reports/`, 2020.

NetworkX developers. NetworkX. `https://networkx.org/`, 2021.

M. E. Newman, S. H. Strogatz, and D. J. Watts. Random graphs with arbitrary degree distributions and their applications. *Physical review E*, 64(2):026118, 2001.

M. E. Newman, D. J. Watts, and S. H. Strogatz. Random graph models of social networks. *Proceedings of the national academy of sciences*, 99(suppl 1):2566–2572, 2002.

Nginx Inc. Nginx. `https://nginx.org/en/docs/`, 2021.

Pandas developers. pandas. `https://pandas.pydata.org/docs/index.html`, 2021.

R. Pastor-Satorras, C. Castellano, P. Van Mieghem, and A. Vespignani. Epidemic processes in complex networks. *Reviews of modern physics*, 87(3):925, 2015.

E. Patrozou and L. A. Mermel. Does influenza transmission occur from asymptomatic infection or prior to symptom onset? *Public health reports*, 124(2):193–196, 2009.

A. Plazas, I. Malvestio, M. Starnini, and A. Díaz-Guilera. Modeling partial lockdowns in multiplex networks using partition strategies. *Applied network science*, 6(1):1–15, 2021.

Plotly Technologies Inc. Plotly Dash. `https://dash.plotly.com/`, 2021a.

Plotly Technologies Inc. Plotly Express. `https://plotly.com/python/plotly-express/`, 2021b.

F. P. Polack, S. J. Thomas, N. Kitchin, J. Absalon, A. Gurtman, S. Lockhart, J. L. Perez, G. P. Marc, E. D. Moreira, C. Zerbini, et al. Safety and efficacy of the bnt162b2 mrna covid-19 vaccine. *New England Journal of Medicine*, 2020.

SafeGraph Inc. SafeGraph Open Census Data. `https://www.safegraph.com/open-census-data`, 2021a. Accessed: 2021-06-15.

SafeGraph Inc. SafeGraph COVID-19 Data Consortium. `https://www.safegraph.com/covid-19-data-consortium`, 2021b. Accessed: 2021-06-15.

K. T. L. Sy, L. F. White, and B. E. Nichols. Population density and basic reproductive number of covid-19 across united states counties. *PloS one*, 16(4):e0249271, 2021.

A. R. Tuite, A. L. Greer, M. Whelan, A.-L. Winter, B. Lee, P. Yan, J. Wu, S. Moghadas, D. Buckeridge, B. Pourbohloul, et al. Estimated epidemiologic parameters and morbidity associated with pandemic h1n1 influenza. *Cmaj*, 182(2):131–136, 2010.

Unbit. uWSGI. `https://uwsgi-docs.readthedocs.io/`, 2021.

US Census Bureau. State Population Totals and Components of Change: 2010-2019. `https://www.census.gov/data/tables/time-series/demo/popest/2010s-state-total.html`, 2021. Accessed: 2021-07-30.

M. Voysey, S. A. C. Clemens, S. A. Madhi, L. Y. Weckx, P. M. Folegatti, P. K. Aley, B. Angus, V. L. Baillie, S. L. Barnabas, Q. E. Bhorat, et al. Safety and efficacy of the chadox1 ncov-19 vaccine (azd1222) against sars-cov-2: an interim analysis of four randomised controlled trials in brazil, south africa, and the uk. *The Lancet*, 397(10269):99–111, 2021.

P. G. Walker, C. Whittaker, O. J. Watson, M. Baguelin, P. Winskill, A. Hamlet, B. A. Djafaara, Z. Cucunubá, D. O. Mesa, W. Green, et al. The impact of covid-19 and strategies for mitigation and suppression in low-and middle-income countries. *Science*, 369(6502):413–422, 2020.

WHO. Listings of WHO's response to COVID-19. `https://www.who.int/news/item/29-06-2020-covidtimeline`, 2020. Accessed: 2021-06-10.

Worldometer. Coronavirus, Active Cases in Connecticut. `https://www.worldometers.info/coronavirus/usa/connecticut/`, 2021. Accessed: 2021-07-30.