Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II. *doi: 10.18637/jss.v000.i00*

Time invariant analysis of epidemics with EpiCompare

Shannon K. Gallagher
Biostatistics Research Branch
National Institute of Allergy
and Infectious Diseases

Benjamin LeRoy
Dept. of Statistics & Data Science
Carnegie Mellon University

Abstract

We present **EpiCompare**, an R package that suppliments and enhances current infectious disease analysis pipelines and encourages comparisons across models and epidemics. A major contribution of this work is the set of novel *time-invariate* tools for model and epidemic comparisons - including time-invariate prediction bands. **EpiCompare** embraces R's *tidy* coding style to make adoption of the package easier and analysis faster. This paper provides an overview of both the tools in and intuition behind **EpiCompare** and a thorough demonstrating of the tools through a detailed example of a full data analysis pipeline.

Keywords: keywords, not capitalized, Java.

1. Introduction

The recent (and on-going) COVID-19 global pandemic has galvanized public interest in understanding more about infectious disease modeling and has highlighted the usefulness of research in the area of infectious disease epidemiology. Infectious diseases inflict enormous burdens on the world: millions of lives lost and trillions of dollars spent yearly. Infectious disease models typically attempt to do one or more of the following: 1) predict the spread of current and future epidemics (e.g. flu prediction Biggerstaff *et al.* 2016), 2) analyze past and current epidemics to increase scientific knowledge (e.g. historical measle outbreaks Neal and Roberts 2004), and 3) forecast or project epidemic scenarios under pre-specified parameters (e.g. Ferguson *et al.* 2020). At the same time, descriptive statistics and visualizations from universities, many branches and levels of government, and news organizations are an important first step of the process, as has been seen in the current COVID-19 pandemic (Dong

et al. 2020; CDC 2021; The Washington Post 2021).

With many visualization and exploratory tools, models and modeling paradigms, and reviews and comparisons in the literature and through the MIDAS (Models of Infectious Disease Agent Study) network (MIDAS Network 2021) available, this field has a number of devices to ~~aid~~ help[1] an individual practitioner decide the correct approach. For example, R packages such as **surveillance**, **EpiModel**, and **pomp** have all made significant steps in standardizing the flow of the data analysis pipeline for epidemic modeling through digitizing data sets, making accessible statistical models, and providing a plethora of educational material for both coding novices and experts alike (Meyer *et al.* 2017; Jenness *et al.* 2018; King *et al.* 2016).

At the same time, analysis packages often only address a specific portion of the analysis pipeline. These modeling tools usually require learning package-specific syntax and often don't provide easy ways to compare and assess their models on new data. Moreover, exploring modeling and comparing epidemics require transforming and *tidying* data in different ways. To fill these gaps, we present our R package **EpiCompare**. Our package's primary focus is to aid and advance research in the area of comparison and assessment of epidemic and epidemiological models. In Figure 1, we illustrate the data analysis pipeline of infectious diseases as 1) data pre-processing, 2) exploratory data analysis (EDA), 3) modeling and simulating, 4) post-processing, and 5) comparison and assessment; where each previous part of the pipeline influences the next. **EpiCompare** provides tools to aids practitioners in all areas of this pipeline.

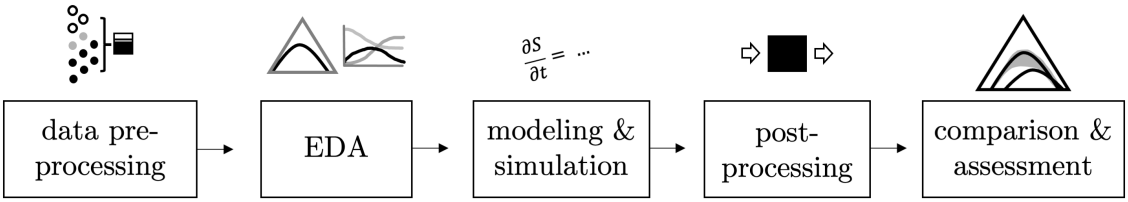

Figure 1: An idealized epidemiological data analysis pipeline.

One of **EpiCompare**'s main contribution to comparison and assessment of epidemics is through tools that provide *time-invariant* assessments. Epidemics, despite being defined as a process that evolves over time, often need to be compared in a way not constrained to initial times or time scales in order to understand the processes at play. With time-invariant analysis, comparing decades-long outbreaks of HIV in the US to a 10 day outbreak of norovirus on a cruise ship is possible. Compared to time-dependent comparison tools for state-space modeling, time-invariant analysis can make it easier to compare state-space epidemic representations in a more global, holistic fashion. Many time-dependent comparison tools only examine the proportion of individuals in each state (at a given time) in a piece-wise / marginal fashion. These time-dependent approaches can reduce the amount of connections that can be seen and insights that can be drawn}, similar to examining projections of a multidimensional distribution onto a single axis, one at a time. Tools in **EpiCompare** extend the user toolkit to evaluate

[1] [Ben says: I'm not sure this an improvement]

epidemics within a time-invariant lens. At the same time, the goal of **EpiCompare** is not to supplant existing infectious disease modeling tools and software but, rather, is a concerted effort to create standard and fair comparisons among models developed for disease outbreaks and outbreak data.

This paper is broken up into the following sections; section 2 motivates and showcases tools of time-invariant analysis, section 3 presents an outline of how **EpiCompare** aids a practitioner in every step of the pipeline and section 4 provides a demonstration of the tools through a detailed example ~~of a full data analysis pipeline.~~ from start to finish of the data analysis pipeline[2]

2. Time-invariant analysis

EpiCompare emphasizes the value of analyzing epidemics in a *time-invariant* way - approaches that remove some or all of the impact of start/end times and recording time scales when preforming the analysis. In this section we highlight some weaknesses of time-*de*pendent analysis and define the mathematical underpinning of the time-invariant approach we take. To accomplish this goal we first demonstrate the inability of time-dependent tools to adequately quantify a classic epidemic parameter, the reproduction number R_0. Then we motivate new time-invariant approaches for more complex situations where R_0 does not capture the complexities of outbreaks. This leads to the final subsection that defines ways to view epidemics in a time-invariant lens and discusses natural properties of these representations that allow for clearer ways to compare models and epidemics.

2.1. Motivation of time-invariant analysis through the reproduction number R_0

Epidemiological research is often interested in understanding intrinsic properties of the epidemic, independent from when the epidemic occurred or how frequently data was collected. One of the most famous numerical summaries of an epidemic is the reproduction number - R_0. This numerical summary is a time-invariant value that is defined as the expected number of infections caused by a single infect or who is added to a completely susceptible population (Anderson and May 1992). Gallagher *et al.* (2020) showed that the estimation of R_0 can be sensitive to time-dependent parameters like estimation of the beginning and end of an epidemic. This time-dependent sensitivity in estimating R_0 reflects a general problem in epidemiology surrounding how to transform time-dependent information into insights into desirable, intrinsic properties of an epidemic.

In many situations, epidemiologists want a much deeper understand of an epidemic than just the reproduction number. One common time-dependent tool used to better understand epidemics is a series of time series line plots that track the proportion of the population in a given state (e.g. infected) at a given time. Figure 2 visualizes two different simulated epidemics with population states (S)usceptible, (I)nfectious and (R)ecovered. These simulations have been generated under a discrete approximation of Kermack and McKendrick (1927)'s SIR model. The Kermack and McKendrick model captures the transitions from one state to the next as a system of ordinary differential equations

[2] maybe clearer? up to you The current suggestion isn't clearer - feel free to try again.

$$
\begin{aligned}
S'(t) &= -\frac{\beta S(t) I(t)}{N} \\
I'(t) &= \frac{\beta S(t) I(t)}{N} - \gamma I(t) \\
R'(t) &= \gamma I(t),
\end{aligned}
\tag{1}
$$

where N is the total number of individuals, β is the rate of infection, and γ is the rate of recovery.

From this model, the reproduction number is the ratio of the infection rate to the recovery rate, $R_0 = \beta/\gamma$.

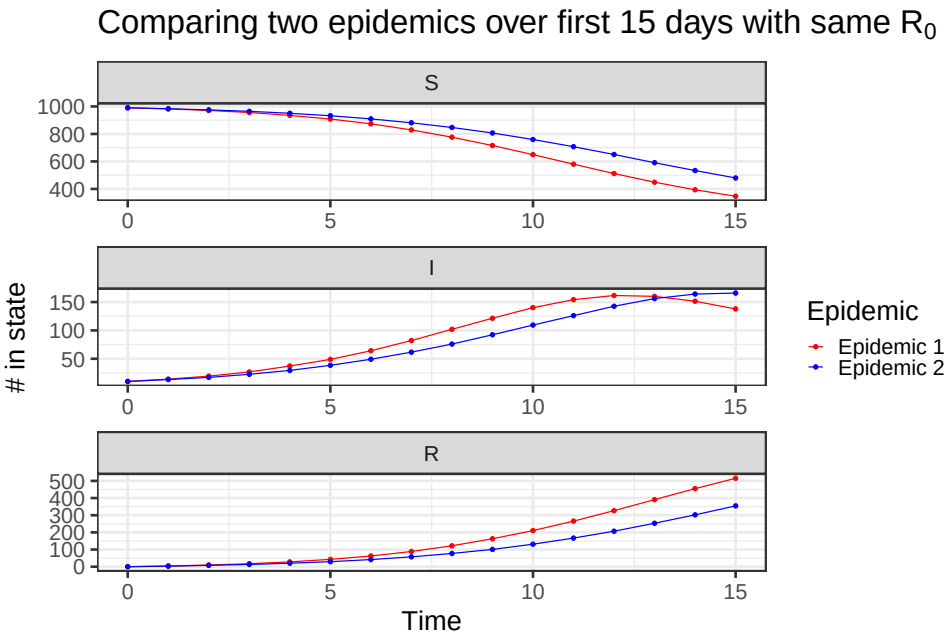

Figure 2: Example of two epidemics with different β and γ parameters but the same initial reproduction number $R_0 = 2$. Both epidemics are generated from models with $N = 1000$ individuals with $S(0) = 990$ and $I(0) = 10$.

Even though Figure 2's visual analysis might be able to present more complex properties of the epidemic, one might wish to understand how these two simulated epidemics' R_0 values compare. Even though these two simulated epidemics appear different in these figures (including having different infection peaks), there is no real intuitive way to compare the epidemics' R_0 values. In fact, both of these simulated epidemics started with the same population (1000 total people with 10 infected) and have the same R_0 value, just with slightly different infection and recovery rates ($\beta_1, \gamma_1 = 0.8, 0.4$, and $\beta_2, \gamma_2 = .64, .32$). Even for simple generative models, this time-dependent visualization cannot be used to compare a very simple but powerful numerical summary, R_0.

2.2. Ternary plots: a time-invariant visualization tool

The faceted time series plot, like that seen in Figure 2, not only fails to allow the practitioner to compare simulated epidemics' R_0 values but also presented epidemics' trajectory data so that only the marginal information for each state can be examined at a given time. A more holistic approach to visualizing the overall path of our simulated epidemics is to examine how each epidemic traverses the three-dimensional $(S(t), I(t), R(t))$ space. Figure 3's left plot does just that, presenting the trajectories of the simulated epidemics in this three-dimensional space. For state space models like in our example, given the constraint that $S(t)+I(t)+R(t)$ is always equal to N (the total population size) we can visual these point in a a two-dimensional *ternary* plot, as seen in Figure 3's right plot. In both of Figure 3's plots we observe that the two epidemics are on the same trajectory. In this simple generative example setting, being on the same trajectory indicates that the two epidemics have the same R_0 value. This can be proven mathematically.

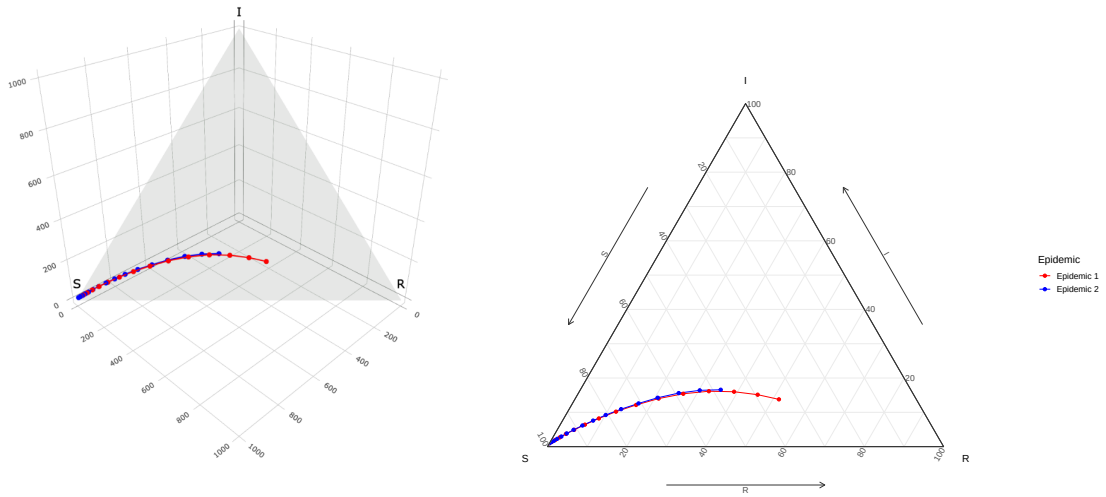

Figure 3: Left: trajectory of epidemic in three-dimensional space, plotting $(S(t), I(t), R(t))$. Right: the gray-shaded region and epidemic trajectories shown from (left) now shown in two-dimensional space. This is more commonly known as a ternary plot.

Theorem 1. *Let there be two Kermack and McKendrick (1927)'s SIR models* $((S_1(t), I_1(t), R_1(t))_{t\geq 0}$ *and* $(S_1(s), I_1(s), R_1(s))_{s\geq 0})$*, with* $(S_1(0), I_1(0), R_1(0)) = S_2(0), I_2(0), R_2(0))$ *Let both models have the same* R_0 *(i.e.* $R_0 = \frac{\beta_1}{\gamma_1} = \frac{\beta_2}{\gamma_2}$*) and define* $a > 0$ *such that* $\beta_2 = a\beta_1$*. Then for all* $t > 0$ *there exists* $s > 0$ *such that* $(S_1(t), I_1(t), R_1(t)) = (S_2(s), I_2(s), R_2(s))$*. Moreover,* $s = \frac{1}{a}t$*.*

The proof of Theorem 1 relies on a fairly recent result from Harko *et al.* (2014) and is shown in detail in Proof 4.7. The consequence of Theorem 1 is that for two SIR models that have the same initial percent of individuals in each state and R_0, then every point on the epidemic path of the first SIR model can be mapped to a point on the epidemic path of the second SIR model. In other words, the two epidemics form the same filamental trajectory.

2.3. Time-invariance beyond SIR models: Trajectories and Filaments

Through the R_0 example, we see that treating epidemics like filamental trajectories embedded in a lower dimensional space allows us to better compare the overall structure of the epidemic and see how the population is directly impacted. In this section we present time-invariant tools that can be applied to complex epidemics where the epidemic's generative process is unknown and can have more than three states. These tools leverage the idea that an epidemic can be viewed as a trajectory and that many properties of the epidemic are well captured when we do so. This approach is useful when the epidemic of interest has only gone through a single realization of its outbreak (before the population of individuals become susceptible again).

The first set of tools allows a practitioner to define distances between epidemics whose time features do not align. For completed epidemics, one way to better examine their properties is to represent their filamental trajectories as a finite sequence of equally spaced points. This representation induces a natural distance between epidemics, specifically:

$$d_{\text{equi-distance}}(\psi_1, \psi_2) = \int_{s\in[0,1]} (\tilde{\psi}_1(s) - \tilde{\psi}_2(s))^2 ds,$$

where $\tilde{\psi}_i(s)$ is the point along ψ_i that is $s \cdot |\psi_i|$ distance away from the start of ψ_i (with $|\psi_i|$ is the length of ψ_i). This distance is naturally time-invariant, and can be plugged into multiple distance-based assessment tools to examine the overall "extremeness" of points, including pseudo-density estimators and depth/local depth functions (for examples see Ciollaro *et al.* 2016; Geenens and Nieto-Reyes 2017). These extremeness estimators can be useful when comparing the true epidemic to a set of simulated epidemics, and practitioners can interpret the epidemic's extremeness score relative to the extremeness scores of the simulations very easily.

In addition to being used to better understand completed epidemics, time-invariant tools can be used to aid prediction of future epidemics by providing a state-space regions in which we expect the true epidemic to traverse. In settings where the epidemic only generates a single outbreak, these regions can be very telling if simulation models capture the epidemic's structure. In **EpiCompare** we create geometric prediction regions around all but the α proportion of most extreme simulated trajectories. These geometric regions can also be used to compared simulation models that have different time scales, parameters and even different statistical philosophies, through set different distances. Although visualization is the easiest when the epidemic has three states, prediction regions can be useful to assess and compare simulation models where the epidemics have multiple states.

Overall, there are many tools to aid in the assessment and comparison of epidemics and models that avoid being affected by time-based parameters. We believe the time-invariant analysis provides many insights and should be in the toolkit of many epidemiologists. **EpiCompare** provides a strong starting point to do just that.

3. Overview of EpiCompare

In this section, we present the tools implemented in **EpiCompare** and explain how they aid in the data analysis pipeline. In Figure 4, we show how our package's functions fit into the data analysis pipeline introduced in Figure 1. All front-facing functions in **EpiCompare** are aimed to be as user-friendly as possible. We also focus on providing the user "tidyverse" style functions, that encourage piping objects from one function to the next and follow clear "verb"

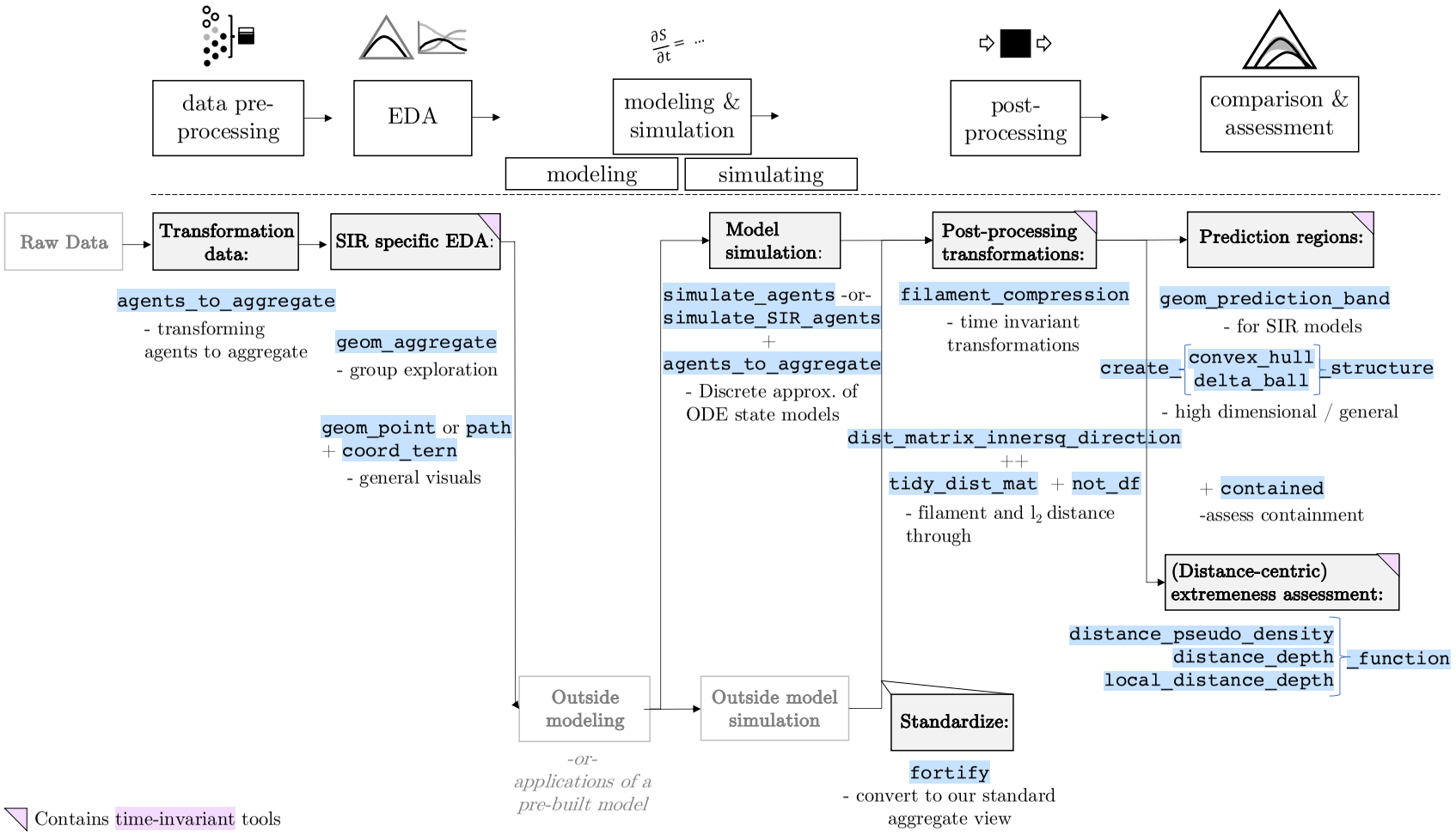

Figure 4: How **EpiCompare** supplements and aids in the epidemiological data analysis pipeline.

naming schemes (Wickham *et al.* 2019). Although users can incorporate **EpiCompare** into any step in the data analysis pipeline, there are two primary points of entry. The first point of entry is the very beginning with pre-processing and visualizing raw data, and the second point of entry is after modeling and simulation. Figure 4 captures these different paths, and we highlight how to leverage **EpiCompare** functionalities in the subsections below.

Data pre-processing

The first step of most data analysis is "cleaning" the raw data so it can be explored. Before data can be explored, they must be collected. Sometimes individual records are collected, with times of different states of the epidemic (infection, recovery, etc.) as well as individual information like network structure, location, and sub-population information. Other data collections focus on aggregate counts of individuals in each epidemic state. In fact, many times only the number of new infections at each time step (e.g. weekly case counts) is observed. In this setting, compartment totals (amounts of individuals in each state) are then imputed from those case counts and using other information about the disease and the population of interest. In **EpiCompare**, we focus on understanding the overall impact of an outbreak at the aggregate/population level, which allows for streamlined examination of overall trends of an epidemic.

To help the practitioner examine epidemics from an aggregate/population lens, we provide a function called `agents_to_aggregate()`. This function transforms data about individual/agents' initial entry into each state (e.g. start of infection, start of recovery, etc.) to an aggregate view of how many individuals were in a state at a given time. Researchers, including Rvachev and Longini (1985); Anderson and May (1992); Worby *et al.* (2015), often are interesting in more granular trends that can be detected by aggregation, conditional on subpopulations (e.g. subpopulations defined by age or sex). By combining the function **dplyr**::`group_by()` and `agents_to_aggregate()`, **EpiCompare** provides group level aggregation.

Besides aiding subpopulation analysis, `agents_to_aggregate()` can accommodate a wide range of information about each individual. In fact, this function can account for infinitely many states. This functionality allows the practitioner to aggregate information relative to common states (e.g. "Susceptible", "Infectious", and "Recovered") as well as more complex states (e.g. "Exposed", "iMmune", "Hospitalized"). Additionally, `agents_to_aggregate()` permits indicators for death/exit and birth/entry dates. Overall, this function is a powerful tool for pre-processing data.

Exploratory data analysis (EDA)

In the early stages of a project, familiarizing oneself with the data usually means figuring out useful combinations of visualizations and numerical summaries of the data both at population and subpopulation level. An expert coder can start with `agents_to_aggregate()` to successfully accomplish exploratory data analysis (EDA) in many ways. **EpiCompare** also includes tools that allow a novice coder to rapidly explore data, provided there are three unique epidemiological states (like in the SIR model). Building on **ggplot2** and **ggtern** packages, **EpiCompare**'s `geom_aggregate()` provides a way to explore how different subpopulations experience of an epidemic (Wickham 2016; Hamilton and Ferry 2018). The function `geom_aggregate()` provides a visualization tool to holistically examine aggregate level information across different subpopulations by visualizing each subpopulation's epidemic trajectory in the three-dimensional state space. Visualization tools for three-state models were

developed because SIR models are some of the most common and basic epidemic state-based models and our three-dimensional simplex representation of these epidemics emphasizes a time-invariant representation of the data (for a refresher see Section 2).

Model fitting and simulations

After getting a sense of what a past or current epidemic looks like with EDA, the next step in the data analysis pipeline is often model fitting and/or simulation. While **EpiCompare** does not focus on fitting models to data, we do provide some flexible functions for simulation of basic discrete-time epidemic-state models. These functions simulate individual-level information based on practitioner estimated transition rates between states and can be combined with `agents_to_aggregate()` to view these simulations through an aggregate lens. The function `simulate_SIR_agents()` simulates a basic SIR epidemic with user inputs for the number of simulations, the initial number in each state, the infection and recovery parameters (β, γ), and the total number of discrete time steps. Beyond SIR models, the function `simulate_agents()` takes as input a user-specified state-transition matrix and other epidemic parameters to allow the user to create simulations for an outbreak with *any* number of states and any number of transitions among them. This flexibility in states can be used to also reflect group-based dynamics. Both of these functions allow users to explore the space of models in an intuitive way without getting bogged down by too much mathematical detail. For consistency, we have made output from `simulate_agents()` and `simulate_SIR_agents()` compatible with `agents_to_aggregate()` so aggregate information may easily be accessed.

Post-processing

If practitioners wish to compare models-to-observations or even models-to-models, they need to post-process their models and simulations to disseminate the results in an easily digestible format. In **EpiCompare**, we provide (1) functions to standardize simulation and model output from external packages and (2) a function to transform standardized simulation and model output into a format amenable to time-invariant analysis.

Modeling and simulation output can be very complex objects, and as a result, a number of epidemic modeling R packages return a special class. The special classes often contain a plethora of information about residuals, model diagnostics, input parameters, and more. While incredibly useful, these special classes can be difficult for novice coders to handle. To this end, **EpiCompare** provides a series of fortify-style methods, called `fortify_aggregate()` which transform output from infectious disease modeling and simulation packages like **pomp** and **EpiModel** into tidy-styled data frames which contain information about the total number of individuals in each state at a given time, for a given simulation. These fortify functions have output that is consistent with that of `agents_to_aggregate()`. These standardized outputs can then be piped to summaries, tables, and plots.

Because epidemic data is stored in a temporal way, we provide the function, `filament_compression()`, to transform temporally defined epidemics to their filamental representations. These filaments can then be fairly compared to one another or passed to further time-invariant analysis tools described below.

Comparisons and assessment

The last step of the data analysis pipeline often ends with plots, tables, and summary statistics that are used to assess model performance and compare across models or simulations. In **EpiCompare** we provide a set of comparison and assessment tools for model and simulation results that extend beyond the standard performance metrics (e.g. mean squared error or AIC)

and into the lens of time-invariant analysis. We have found that these tools are specifically applicable for situations where only one season or cycle of an epidemic has occurred or is the object of interest.

The first set of of tools surround the creation of prediction regions. We can create a prediction regions from model simulations to examine if our model simulations capture the true epidemic trajectory. We do so in a time-invariant way and utilizing filamental representations of the model simulations and the true epidemic. For three-state epidemic models, we provide the `ggplot/ggtern` extension `geom_prediction_band()` which creates a prediction region around the top $1 - \alpha$ proportion of the simulations. In this visual setting, comparing this prediction region to the true epidemic trajectory can be done by eye. In **EpiCompare**, we also provide these prediction regions for epidemic models with with more than three states. The functions `create_convex_hull_structure()` and `create_delta_ball_structure()` create different geometric representations of prediction regions for any dimensional state-based model. For both of these geometric structures, we provide functions to check if a path is contained (`contained()`)]. We can also use these prediction regions to visually or mathematically compare how similar two sets of simulations are. In **EpiCompare** we provide the `hausdorff_dist()` function to calculate the Hausdorff distance between multiple prediction regions, when visual comparison is not possible.

We also provide functions to calculate the "extremeness" of a true epidemic trajectory compared to simulated epidemics via the equi-distance filamental trajectory representation as mentioned in Section **??**.] We provide implementations of a few distance-based score functions that capture how "reasonable" an epidemic is relative to other epidemics, and these scores can be turned into an extremeness measure with `mean(sim_scores > truth_score)`. [Specifically, functions like `distance_pseudo_density_function()` can calculate a pseudo-density estimate of the true epidemic relative to simulated ones. Functions `distance_depth_function()` and `local_distance_depth_function()` provide depth scores that suggest how geometrically central an epidemic is to simulations.

4. A tour of EpiCompare

I have substantially overhauled section 4. The older versions are kept after the entirety of this section but note that much of the text (but not the structure) of this section is different. I've tried to better motivate why we use the epicompare functions to answer our questions of interest.

UPDATE 6/11/2021. I am now adding overall goals of section and each subheading. Additionally there are a few updates from last draft (unmarked since you didn't comment on last draft).

SECTION GOALS:

The main goal of this section is to show how a person can use EpiCompare in a real data analysis. With that said my primary focus is on EpiCompare and not the data analysis – a point I think you may be having an issue with. I've given this some thought and believe that the data analysis is a means for an end to us – showing off EpiCompare. That said again, I've made some changes in the first paragraph below.

To accomplish this main goal, I shadow the data analysis pipeline we have introduced and show how epicompare can be used in the steps.

The secondary goals are figuring out useful information about this measles outbreak.

[[NEWEST TEXT

[[sub-section goal: outline goal of section – how to use epicompare in a real situation

To conclude our paper, we demonstrate how **EpiCompare** streamlines the data analysis process with a case study of a measles outbreak in 1861-1862 Germany. With the help of **EpiCompare**, we can answer important questions such as is a SIR model a good fit for the data, does the outbreak spread differently within the different school classes of the children, and can incorporating an underlying network structure enhance model fit? After introducing the data available in this outbreak, we show how **EpiCompare** can aid in each step of the data analysis pipeline (see Fig. 4) to answer these questions motivated by this outbreak.]]

4.1. Background for 1861-1862 measles outbreak

[[Goal of this section isto provide context for our case study.]]

We analyze an outbreak of measles in the town of Hagelloch, Germany from 1861-1862, a data set organized by Pfeilsticker (1863). The data was later made visible by Oesterle (1992) and made available in an R by Meyer *et al.* (2017). In this outbreak, 188 children were infected with measles over the course of three months. This data set includes a rich collection of features including day of measles rash, age, sex, school class, household and household location, and alleged infector of each child. We show a subset of the data in Table 1. We are particularly interested if the SIR model is a good fit to our data, whether we can identify subgroups with differing infection behavior, and whether incorporating those subgroups into a network-based SIR model enhances model fit compared to the baseline SIR.

Table 1: Subset of Hagelloch infection data. Features include the person ID, household ID (HH ID), age, sex, class level (Pre-K/1st/2nd), date of first symptoms, date of the appearance of the measles rash, and the alleged infector ID of the individual.

ID	HH ID	Name	Age	Sex	Class	Symp. Start	Rash Date	Infector ID
1	61	Mueller	7	female	1st class	1861-11-21	1861-11-25	45
2	61	Mueller	6	female	1st class	1861-11-23	1861-11-27	45
3	61	Mueller	4	female	preschool	1861-11-28	1861-12-02	172
4	62	Seibold	13	male	2nd class	1861-11-27	1861-11-28	180
5	63	Motzer	8	female	1st class	1861-11-22	1861-11-27	45
45	51	Goehring	7	male	1st class	1861-11-11	1861-11-13	184

4.2. Pre-processing and EDA

[[The goal of this is to show how EpiCompare can be used to transform the data for basic tablaes and plotting, especially with regards to aggregating the data for the SIR model.]]

We begin our analysis by first examining and transforming the raw data (`hagelloch_raw`) to explore whether the SIR model is a good fit to the data. The raw data is in the format of individual-level data and first needs to be aggregated. By specifying the time of infection (`tI`) and the time of recovery (`tR`), the function `agents_to_aggregate()` calculates the number

susceptible, infectious, and recovered individuals at each time step. Once aggregated, we can plot the SIR values through a time-invariant lens using **ggplot2** and **ggtern** functions (as shown in Fig. 9) or with our custom `geom`, `geom_aggregate()`, which takes the raw agent data as input. This is show in the below code.

```
R> hagelloch_sir <- hagelloch_raw %>%
+   agents_to_aggregate(states = c(tI, tR),
+                       min_max_time = c(0, 55)) %>%
+   rename(time = t, S = X0, I = X1, R = X2)
R>
R>
R> ggplot(hagelloch_sir, aes(x = S, y = I, z = R))+
+   coord_tern() +
+   geom_path() +
+   labs(x = "S", y = "I", z = "R",
+        title = "Time invariant view of Hagelloch measles outbreak") +
+   theme_sir(base_size = 24)
```

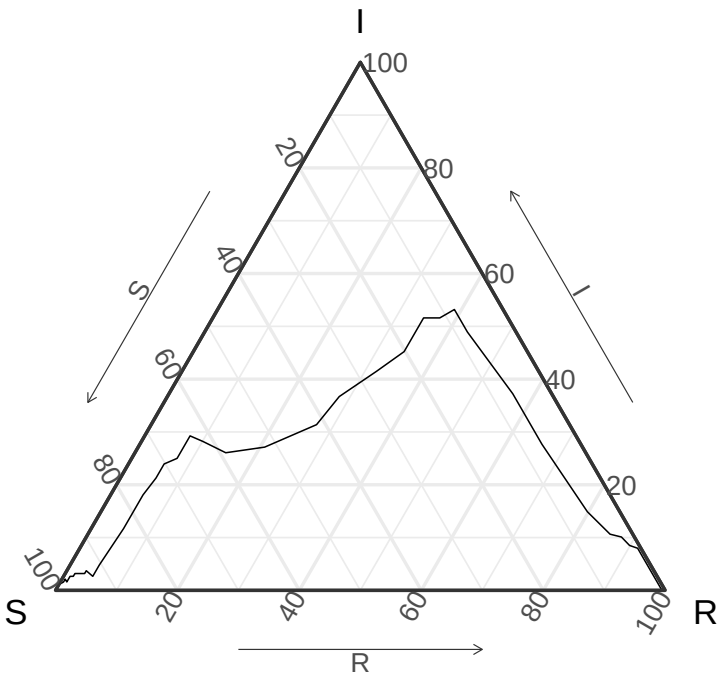

Figure 5: Time invariant view of the Hagelloch epidemic where we view the individuals in Susceptible, Infectious, or Recovered states. We see there are two peaks of infection (the vertical axis).

In Figure 9, we can focus on the infections over time by analyzing the vertical axis. Specifically, we see two peaks of infection. This is interesting because the SIR model, which is sometimes

used to model the spread of measles, generally has one defined peak of infection. We may wonder if the two peaks in the observed data may be due to random noise or if a model more complex than the simple SIR is needed to adequately capture these two peaks.

With regards to our goal of identifying interesting subgroups with different infection behavior, previous study tells us that measles outbreaks are often associated with children within the same grade level, and we examine if this is the case here. By combining the **dplyr** function `facet_wrap()` with `geom_aggregate()` we can easily analyze this scenario,

```
R> hagelloch_raw %>%
+   ggplot(aes(y = tI, z = tR, color = CL)) +
+   geom_aggregate(size = 2) + coord_tern() +
+   labs(x = "S", y = "I", z = "R",
+        color = "Class") +
+   scale_color_brewer(palette = "Dark2") +
+   facet_wrap(~CL)
```

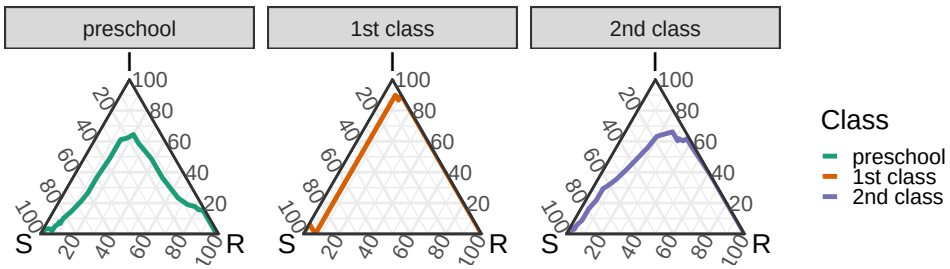

Figure 6: Time invariant outbreak curves for the three class groups. The pre-school class has a distinct peak of infection whereas the peak infection point for the other two classes are less well defined.

Immediately in Fig. 10, we see that time invariant infection curve is different for the pre-school class compared to the 1st class. In the 1st class, we see about 95% of the class become infected and less than 10% of them having recovered, which may be indicative of a super-spreading event. This suspicion is further supported in that 26 of the 30 1st class students have been reportedly infected by the same individual. We now have some evidence that class structure may play a role in the spread of infection. We can further analyze this claim with modeling and simulation.

4.3. Modeling and Simulation

[[The goal is to 'test' (although not formally) whether our observations from EDA are correct. We want to see whether SIR model is a good fit and whether including these found sub-groups helps model fit]]

We now use modeling and simulation to informally test whether the baseline SIR model is a good fit for the data and whether incorporating a network-based structure dependent on the class of the children improves model fit.

We first try to model the Hagelloch data with a baseline stochastic SIR model, which we refer to as the 'simple SIR.' In our full vignette (available online), we show how to fit this simple

SIR model via maximum likelihood, a common approach used to fit parameters, and simulate from the model with those best fit parameters. Our function `simulate_agents()` (or `simulate_SIR_agents()`) generates individual level data according to discrete-time multinomial draws, which depend on the number of individuals in each state at the previous time step and a matrix of transition probabilities. For example, the below code generates 100 simulations of an outbreak of a disease with one initial infector in a population of $n = 188$ individuals, a scenario analogous to the actual outbreak.

```
R> trans_mat <- matrix(c("X0 * (1 - X1 * par1 / N)", "X0 * X1  * par1 / N", "0",
+                   "0", "X1 * (1 - par2)", "par2 * X1",
+                   "0", "0", "X2"), byrow = TRUE, nrow = 3)

R> set.seed(2020)
R>
R> best_params <- c("beta" = .36, "gamma" = .13)
R> ## This is the SIR representation
R>
R> rownames(trans_mat) <- c("S", "I", "R")
R> init_vals <- c(187, 1, 0)
R> par_vals <- c(par1 = best_params[1], par2 = best_params[2])
R> max_T <- 55
R> n_sims <- 100
R>
R> agents <- simulate_agents(trans_mat,
+                         init_vals,
+                         par_vals,
+                         max_T,
+                         n_sims,
+                         verbose = FALSE)

R> agg_model <- agents %>% group_by(sim) %>%
+   agents_to_aggregate(states = c(I, R)) %>%
+   mutate(Type = "Simple SIR")
```

The result of our simulation is the object `agents` which is a 18800×5 data frame, which details the time of entry into the S, I, and R states for a given simulation.

To fit a more complex SIR model with a network structure, we use the package **EpiModel** (Jenness *et al.* 2018). The below code sets up a network of individuals (which includes class as a variable) and then simulates infection and recovery over this network.

```
R> library(EpiModel)
R> ## WARNING:  Will take a minute or two
R>
R> set.seed(42)
R> nw <- network.initialize(n = 188, directed = FALSE)
R> nw <- set.vertex.attribute(nw, "group", rep(0:2, each = 90, 30, 68))
```

```
R> formation <- ~edges + nodematch("group") + concurrent
R> target.stats <- c(200, 300, 200)
R> coef.diss <- dissolution_coefs(dissolution = ~offset(edges),  duration = 5)
R> est1 <- netest(nw, formation, target.stats, coef.diss, edapprox = TRUE)
R>
R> param <- param.net(inf.prob = 0.1, act.rate = 5, rec.rate = 0.1)
R> status.vector <- c(rep(0, 90), rep(0, 30), rep(0, 67), 1)
R> status.vector <- ifelse(status.vector == 1, "i", "s")
R> init <- init.net(status.vector = status.vector)
R> control <- control.net(type = "SIR", nsteps = 55,
+                         nsims = 100, epi.by = "group")
R> epimodel_sir <- netsim(est1, param, init, control)
```

The output of this network model is `epimodel_sir`, an object of class `netsim`, which contains a plethora of modeling information. In the following section, we will use the capabilities of **EpiCompare** to streamline the process of comparing the two models contained in the objects `agg_model` (the simple SIR) and `epimodel_sir` (the network SIR model).

4.4. Post-processing and comparison

[[Finally we want to answer the questions in our analysis by comparing the models to the data and the models to each other]].

Ultimately we want to compare the models to the data and the models to one another to answer our questions of interest. The **EpiCompare** function `fortify_aggregate()` takes in an object from specialized classes of modeling output (like those made by `netsim()`) and transforms it into a tidy-style data frame.

```
R> fortified_net <- fortify_aggregate(epimodel_sir,
+                                     states = c("s.num", "i.num", "r.num")) %>%
+   mutate(Type = "EpiModel SIR",
+          sim = as.numeric(gsub("sim", "", sim)))
```

With the two modeling objects both in the same format, we can then compare the models side-by-side. The results are shown in Figure 11, where a 90% prediction region is estimated for the two models and the actual data are plotted as black dots. For the Simple SIR model, we see that while the prediction region covers the data fairly well, the prediction region clearly misses the second peak of infection. This indicates that the simple SIR is not a good fit to our data, even after incorporating noise. We also see that the prediction region is very large, covering up a large area of the ternary plot. Together, this indicates that the simple SIR model produces a biased model with a large amount of variance. On the other hand, for the EpiModel network model, we see that the prediction region covers the data quite well and takes up less area compared to the simple SIR. With **EpiCompare**, we see that the model using the class structure is a better fit to the outbreak than the simple SIR model.

```
R> both_models <- bind_rows(agg_model, fortified_net)
R>
R>
```

```
R> g <- ggplot() + geom_prediction_band(data = both_models %>% filter(t != 0) %>%
+                   mutate(Type = factor(Type, levels = c("Simple SIR",
+                                                         "EpiModel SIR"))),
+          aes(x = X0, y = X1, z = X2,
+              sim_group = sim, fill = Type),
+          alpha = .5,
+          conf_level = .90)

R> g +   geom_path(data = both_models %>% filter(t != 0) %>%
+                   mutate(Type = factor(Type, levels = c("Simple SIR",
+                                                         "EpiModel SIR"))),
+             aes(x = X0, y = X1, z = X2, group = paste(Type, sim)),
+             alpha = .3, col = "gray40") +
+     coord_tern() + theme_sir(base_size = 24) +
+   geom_point(data = hagelloch_sir,
+              aes(x = S, y = I, z =R), col = "black") +
+   labs(title = "Simple SIR model",
+        subtitle = "90% Prediction band and original data",
+        x = "S", y = "I", z = "R") +
+   scale_fill_manual(values = c("#006677", "#AA6600")) +
+   facet_wrap(~Type) +
+   theme(legend.position = "bottom")
```

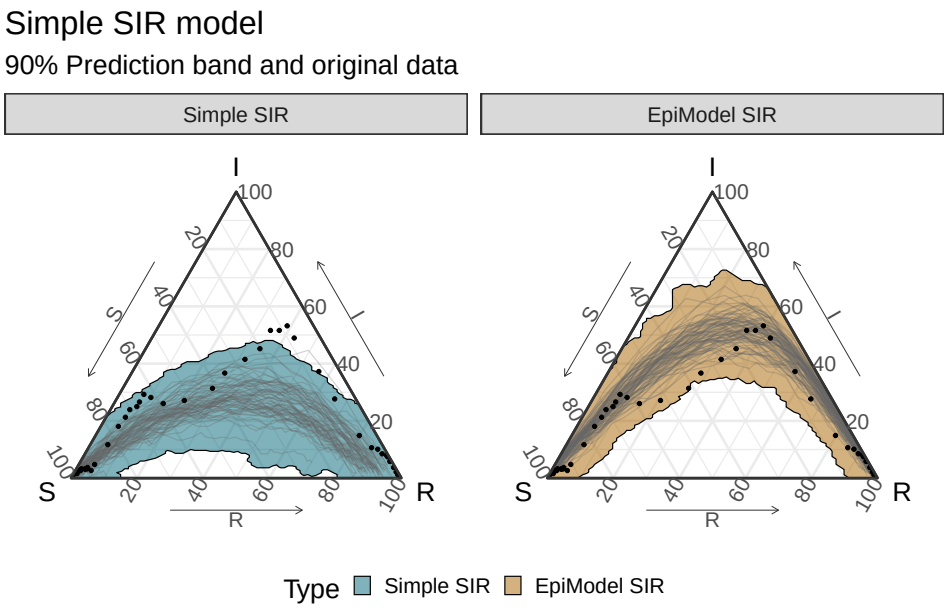

Figure 7: Original Hagelloch SIR data (black) along with 90% prediction band and actual simulation paths from the Simple SIR and the EpiModel SIR models.

Although the prediction region generated by the network model covers the observed data well visually, that does not mean that individual filaments generated from the network model

are good fits to the observed data. We can further examine the model fits by incorporating these filaments into our visualization. In Fig. 11 we show the individual filaments generated from the two sets of models as gray lines. Examining these, we see that the individual lines typically only have one defined peak, whereas the data certainly looks like it has two distinct peaks, a feature possibly caused by our speculated super-spreader event.

We can also examine these filaments quantitatively by using functions that take the distance between the filaments with the observed data. In the below code, we transform the simulations to a more computationally-friendly format with the function `filament_compression()`. Following that, we calculate the distance between the simulated filaments and the observed filament with the function `dist_matrix_innersq_direction()` and calculate the probability of the truth with respect to those simulations with the function `compare_new_to_rest_via_distance()`. The estimated pseudo-density of the observed epidemic (relative to the simulations from either model) is much less likely then **any** of the simulations (reported in Table 6). This indicates that neither of two SIR models are good fits to the data at the filament level.

```
R> simple_sir <- both_models %>% filter(Type == "Simple SIR") %>%
+   rename(S = "X0", I = "X1", R = "X2") %>%
+   select(Type, sim, t, S, I, R)
R>
R> hagelloch_sir2 <- hagelloch_sir %>%
+   rename(t = "time") %>%
+   mutate(Type = "true observation",
+          sim = 0) %>%
+   select(Type, sim, t, S, I, R)

R> #-- after cleaning up and combining --
R> all_together_df <- rbind(simple_sir,
+                           hagelloch_sir2)
```

Table 2: Top and bottom 2 rows of `all_together_df`, combining both simulated epidemics and the true epidemic.

Type	sim	t	S	I	R
Simple SIR	1	0	188	0	0
Simple SIR	1	1	187	1	0
true observation	0	54	1	0	187
true observation	0	55	1	0	187

```
R> compression_df <- all_together_df %>% group_by(Type, sim) %>%
+   filament_compression(data_columns = c("S","I","R"),
+                        number_points = 20)

R> tdmat <- compression_df %>%
+   dist_matrix_innersq_direction(
```

```
+     position = c(1:length(compression_df))[
+       names(compression_df) %in% c("S","I", "R")],
+     tdm_out = T)
R>
R> simple_sir_true_obs_info <- tdmat %>%
+   compare_new_to_rest_via_distance(
+     new_name_id = data.frame(Type = "true observation", sim = 0),
+     distance_func = distance_psuedo_density_function,
+     sigma = "20%")
```

Table 3: The extremeness of the true simulations based on comparing pseudo-density estimates between true vs simulated curves

Type	simulations-based estimated pseudo-density	proportion of simulations with lower estimated pseudo-density
Simple SIR	0.0036733	0
EpiModel SIR	0.0028813	0

In conclusion, **EpiCompare** allows us to fully examine this outbreak at every step in the data analysis pipeline (see Fig. 4) in a streamlined fashion. With EDA, we saw evidence that class structure may be important in the spread of of measles. We then compared a baseline simple SIR model to a more complicated SIR model which incorporated a network structure which included the class structure. Based on the prediction regions generated from these models, we saw that the network model fit the data better than the simple SIR model. However, when we examined the individual filaments generated by the network model, we found that the data are unlikely to be generated from such a model. For further analysis, we would recommend looking into models that can more accurately capture super-spreading events based on the observation that one child was allegedly responsible for nearly all of his classmates' infections. Overall, this analysis demonstrates how **EpiCompare** aids in the data analysis pipeline for both novice and expert practitioners and coders alike.

]]

[[NEWER TEXT

[3] To conclude our paper, we demonstrate the capabilities of **EpiCompare** with a complete data analysis of a measles outbreak in 1861-1862 Germany. Specifically, we demonstrate how tools in **EpiCompare** can be used in each step of the data analysis pipeline [4] ~~(see Figure 1)~~[5]. Additionally, we highlight how time-invariant analysis (see Section 2) can be used to enhance understanding of an outbreak[6].

[3] [Ben says: Generally I found this whole section to be pretty passive and not well motivated on why we'd actually do the analysis.]

[4] [Ben says: based on an earlier read of this section, I might suggest something like "and streamline the analysis process" - is that a selling point you want to highlight?] check out new re-write. new goal is to motivate question of hagelloch and then show how epicompare helps

[5] [Ben says: It's unclear why you want to highlight the figure again - could you be clearere on that? / what does it add to the conv/ why should they reference it?]

[6] [Ben says: I'm not sure this is necessary to state. Additionally - is it really true do we highlight this?]

[7] Before demonstrating **EpiCompare**, we provide some context for the measles outbreak presented here.[8] The data was originally organized by Pfeilsticker (1863), later made visible by Oesterle (1992), and made available in an R by Meyer *et al.* (2017). This data set includes a rich collection of features including household location, class level, and allged infector ID, and is an ideal testing ground for methodology in infectious disease epidemiology Neal and Roberts (2004); Britton *et al.* (2011); Groendyke *et al.* (2012); Becker *et al.* (2016)[9]. In this data set, there are 188 children who became infected with the measles over the course approximately 90 days.

]]

[[LESS NEW TEXT

Finally, in this section we show how tools from **EpiCompare** can be used in each step of the data analysis pipeline shown in Fig. 1. We analyze an outbreak of measles in the town of Hagelloch, Germany from 1861-1862, a data set organized by Pfeilsticker (1863). The data was later made visible by Oesterle (1992) and made available in an R by Meyer *et al.* (2017). This data set includes a rich collection of features and is an ideal testing ground for methodology in infectious disease epidemiology Neal and Roberts (2004); Britton *et al.* (2011); Groendyke *et al.* (2012); Becker *et al.* (2016).[10]

]]

[[OLD TEXT

[11]In this section, we highlight many of the tools available in **EpiCompare**. As previously discussed, these tools include data cleaning; visualization; modeling and simulation; post-processing; and comparison and model assessment, in accordance with the data analysis pipeline (Fig. 1). We show a full data analysis from beginning to end that can be accomplished in a streamlined and standardized manner via **EpiCompare**.]]

4.5. Pre-processing and EDA

[12]The Hagelloch data include a rich set of features at the individual level, and the tools in **EpiCompare** help with pre-processsing and EDA. Recorded features include household members, school level, household locations, date of first symptoms (prodromes), date of measles rash, and even the alleged infector. A subset of the data is shown in Table 1. For example,[13] with **EpiCompare**, we can easily pre-process the data to obtain the empirical cumulative incidence function with respect to the measles rash appearance (variable `ERU`) with the following tidy-style function, `agents_to_aggregate()`. The function `agents_to_aggregate()` is a key

[7] [Ben says:The data introduction could be another subsection. Why does it fit better here?. Other comment: This paragraph is pretty rambly. It is unclear what you want someone to take away from it. I might suggest highlight the fact that the data is "an ideal testing group for methodology".]

[8] [Ben says: this first sentence is pretty sign-post-y yet it only relates to the next few sentences. Update so it's less sign-post-y.]

[9] [Ben says: This citation doesn't make sense - should it be "citep"?]

[10] The old first paragraph from data and exploratory analysis paragraph was combined with the intro as a better lead-in to what's going on.

[11] [Ben says: Shannon, would you mind reading this whole section over again once we've finished edits for section 2 and 3? This initial paragraph seems to be stating section 3's story.]

[12] [Ben says: The first two sentences is very similar to the data paragraph in the section above. Given it's not really connecting the 2 sections I suggest a rewrite - could move some of this stuff above.]

[13] [Ben says: I'm unclear of what this is actually an example of.]

component of **EpiCompare**, allowing the user to easily switch from an individual-level (i.e. an agent) ~~view~~ lens of a disease to an aggregate ~~level~~ lens. For example[14], the below code shows how we can convert the agent data to a cumulative incidence plot of the measles rash~~, in order to see how the disease spread through the population over time.~~ We can then compare the cumulative incidence of the rash to the cumulative incidence of the prodromes, i.e. the initial[15] symptoms[16]. We do this with the below code, and a part of the cumulative incidence data output is shown in Table 4. The argument `integer_time_expansion` indicates whether we should include all time points in the recorded range of the data or only when there is a change in the incidence.

```
R> cif_rash  <- hagelloch_raw %>%
+   mutate(time_of_rash = as.numeric(ERU - min(PRO, na.rm = TRUE))) %>%
+   agents_to_aggregate(states = time_of_rash,
+                       integer_time_expansion = FALSE) %>%
+   mutate(type = "Rash")
```

Table 4: Turning the individual-level information from the Hagelloch data to an aggregate view of the cumulative incidence of the measles rash in the population over time.

Time	# Susceptible	# Total rash appearances
0	188	0
4	187	1
7	186	2
9	185	3
12	183	5

One possible question of interest is the duration between initial onset of prodromes and the appearance of the measles rash[17]. Since `agents_to_aggregate()` outputs a tidy-style data frame, it is a simple task to plot the two sets of incidence curves on the same graph (Fig. 8).

```
R> cif_prodromes <- hagelloch_raw %>%
+   mutate(time_of_PRO = as.numeric(PRO - min(PRO, na.rm = TRUE))) %>%
+   agents_to_aggregate(states = time_of_PRO,
+                       integer_time_expansion = FALSE) %>%
+   mutate(type = "Pro")
```

18

[14] [Ben says: Same comment as before.]

[15] [Ben says: please be clearer on what these could be given you comparing them to start of the rash - which seems like an early symptom to me...]

[16] [Ben says: This action in the analysis pipeline is unmotivated - which naturally makes me want to ask "why would I do this?"]

[17] [Ben says: You don't give a good definition of prodromes above, and you only use the name twice. Is this a super common term in Epi? I find it a bit taxing on the reader to remember what this is referring to.]

[18] I'm confused why Figure 5 is included. What is the conclusion you'd like to take away? / Why do people create plots like this?

```
R> plot_df <- bind_rows(cif_rash, cif_prodromes)
R>
R> ggplot(data = plot_df,
+         aes(x = t, y = X1, col = type)) +
+   geom_step() +
+   labs(title = "Cumulative incidence of measles appearance",
+         x = "Time (days relative to first prodrome appearance)",
+         y = "Cumulative incidence of event") +
+   coord_cartesian(xlim = c(0, 55)) +
+   scale_color_manual(values = c("blue", "red"))
```

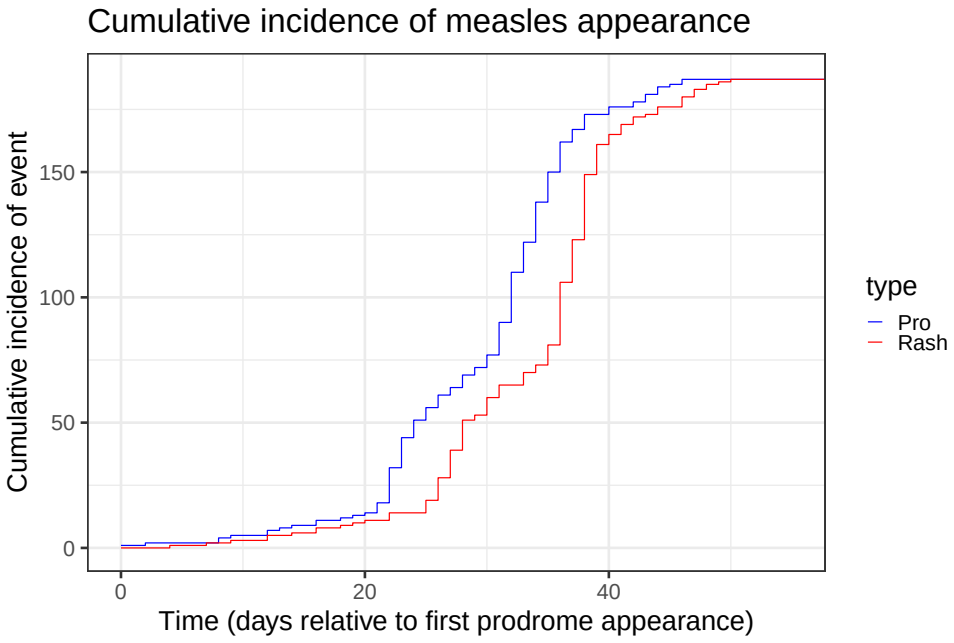

Figure 8: Empirical cumulative incidence functions of prodrome (symptom) onset and measles rash appearance. We see that there is approximately a a constant lag between the two curves.

The real power of `agents_to_aggregate()` lies in its ability to aggregate over any number of pre-specified states. For example, the Hagelloch data sets contains two columns, `tI` and `tR`, the time of infection and recovery, respectively of each individual. We can then[19] plot the SIR values through a time-invariant lens using **ggplot2** and **ggtern** functions (as shown in Fig. 9) or with our custom `geom`, `geom_aggregate`, which takes the raw agent data as input.

```
R> hagelloch_sir <- hagelloch_raw %>%
+   agents_to_aggregate(states = c(tI, tR),
+                       min_max_time = c(0, 55)) %>%
+   rename(time = t, S = X0, I = X1, R = X2)
```

[19] [Ben says: using "then" here captures a very progression step of analysis but I stopped here and asked "what is this following" - and the previous "step" occurred a paragraph back and wasn't described as a direct progression but just a possible thing to do.]

```
R>
R>
R> ggplot(hagelloch_sir, aes(x = S, y = I, z = R))+
+   coord_tern() +
+   geom_path() +
+   labs(x = "S", y = "I", z = "R",
+        title = "Time invariant view of Hagelloch measles outbreak") +
+   theme_sir(base_size = 24)
```

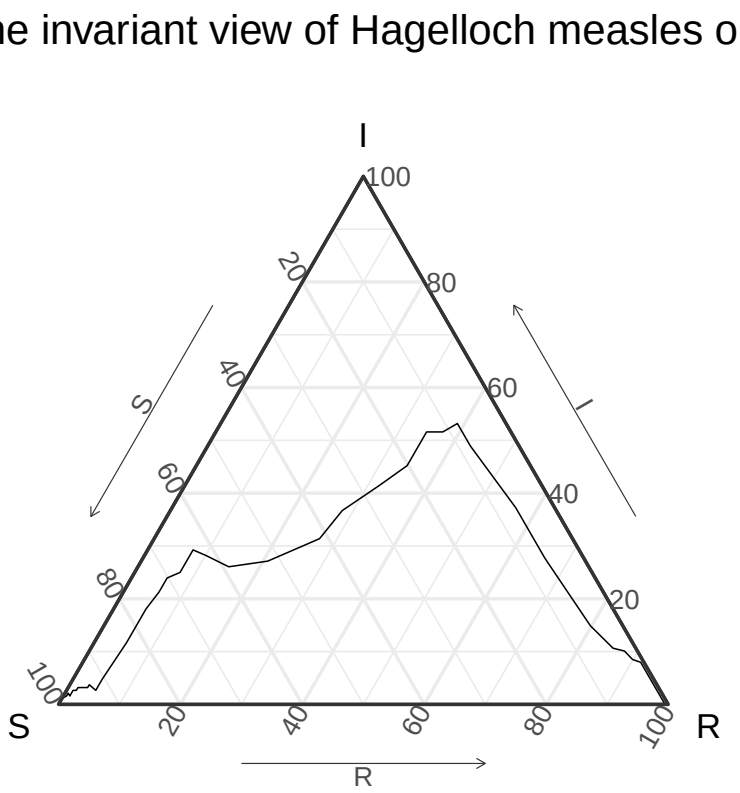

Figure 9: Time invariant view of the Hagelloch epidemic where we view the individuals in Susceptible, Infectious, or Recovered states. We see there are two peaks of infection (the vertical axis).

[20]Moreover, we can look at the outbreaks of the disease by group within `agent_to_aggregate()` or `geom_aggregate()`. This allows us to examine differences among the different groups of individuals. For example, we show the time invariant outbreak by class level in Figure 10. Immediately, we see that time invariant infection curve is different for the pre-school class compared to the 1st class. In the 1st class, we see about 95% of the class become infected and less than 10% of them having recovered, which may be indicative of a super-spreading event. This suspicion is further confirmed in that 26 of the 30 1st class students have been reportedly infected by the same individual.

[20]I found this pargraph very unmotivated. I recommend first arguing why we might a care to look into the class subpopulation grouping. And maybe comment that is is a common desire for practioners.

```
R> hagelloch_raw %>%
+   ggplot(aes(y = tI, z = tR, color = CL)) +
+   geom_aggregate(size = 2) + coord_tern() +
+   labs(x = "S", y = "I", z = "R",
+        color = "Class") +
+   scale_color_brewer(palette = "Dark2") +
+   facet_wrap(~CL)
```

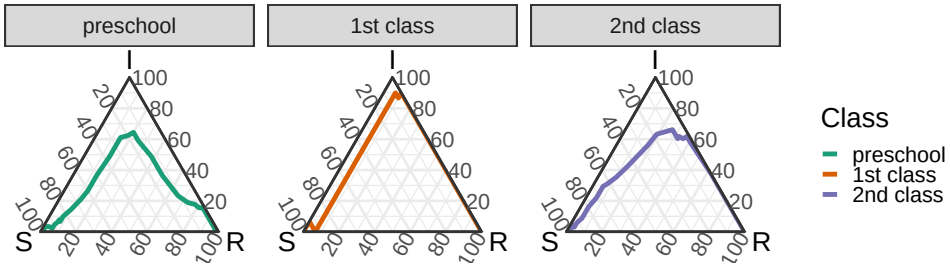

Figure 10: Time invariant outbreak curves for the three class groups. The pre-school class has a distinct peak of infection whereas the peak infection point for the other two classes are less well defined.

~~Along with multiple epidemic states, the function~~ `agents_to_aggregate()` ~~can also be extended to populations with vital dynamics (e.g. birth and death) and examples of this are shown in the package vignette. In summary,~~ `agents_to_aggregate()` ~~is a multi-purpose workhorse that may be leveraged to convert individual level records into aggregate information that may be more useful for some forms of epidemic modeling such as compartment modeling.~~[21]

4.6. Modeling and simulation

[22]

Up to this point, we have used **EpiCompare** in the context of observed data. [23]We also want to compare statistical models, and **EpiCompare** aids in that process via a simple yet flexible individual-level simulator~~, conversion tools for popular epidemic model packages, and model assessments.~~ We demonstrate an example[24] here.

We first try to model the Hagelloch data with a stochastic SIR model, which we refer to as the 'simple SIR.'[25] In our full vignette (available online), we show how to fit this simple SIR model via maximum likelihood and simulate from the model with those best fit parameters[26]. Our function `simulate_agents()`\footnote{[Ben Should somethings be said about the `simulation_SIR_agents()` function?]} generates individual level data according to discrete time multinomial draws, which depend on the number of individuals in each state at the previous time step and a matrix of transition probabilities. For example, the below code

[21] Is this not just a repeat of section 3?
[22] section headings to align with our pipeline
[23] [Ben says: Why?]
[24] [Ben says: to me this whole section is 1 example - as such this wording is confusing to me.]
[25] [Ben says: could / should this be thought of as a "base" model?]
[26] [Ben says: should you highlight that this is a common approach?]

generates 100 simulations of an outbreak of a disease with one initial infector in a population of $n = 188$ individuals.

```
R> trans_mat <- matrix(c("X0 * (1 - X1 * par1 / N)", "X0 * X1  * par1 / N", "0",
+                   "0", "X1 * (1 - par2)", "par2 * X1",
+                   "0", "0", "X2"), byrow = TRUE, nrow = 3)

R> set.seed(2020)
R>
R> best_params <- c("beta" = .36, "gamma" = .13)
R> ## This is the SIR representation
R>
R> rownames(trans_mat) <- c("S", "I", "R")
R> init_vals <- c(187, 1, 0)
R> par_vals <- c(par1 = best_params[1], par2 = best_params[2])
R> max_T <- 55
R> n_sims <- 100
R>
R> agents <- simulate_agents(trans_mat,
+                         init_vals,
+                         par_vals,
+                         max_T,
+                         n_sims,
+                         verbose = FALSE)

R> agg_model <- agents %>% group_by(sim) %>%
+   agents_to_aggregate(states = c(I, R)) %>%
+   mutate(Type = "Simple SIR")
```

The result of our simulation is the object `agents` which is a 18800×5 data frame, which details the time of entry into the S, I, and R states for a given simulation. [27]Before we examine the results of this simple SIR model, we will also examine another, more sophisticated SIR model, this time from the package **EpiModel** (Jenness *et al.* 2018). Briefly, this model first fits a contact network to the set of individuals, where the class of the child is a covariate[28]. The model then simulates a SIR-epidemic on that network.

```
R> library(EpiModel)
R> ## WARNING:  Will take a minute or two
R>
R> set.seed(42)
R> nw <- network.initialize(n = 188, directed = FALSE)
R> nw <- set.vertex.attribute(nw, "group", rep(0:2, each = 90, 30, 68))
R> formation <- ~edges + nodematch("group") + concurrent
```

[27] [Ben says: please motivate - through model comparison?]

[28] [Ben says: Make this connect better to the problem at hand - why do you think you should build this bigger model?]

```
R> target.stats <- c(200, 300, 200)
R> coef.diss <- dissolution_coefs(dissolution = ~offset(edges),  duration = 5)
R> est1 <- netest(nw, formation, target.stats, coef.diss, edapprox = TRUE)
R>
R> param <- param.net(inf.prob = 0.1, act.rate = 5, rec.rate = 0.1)
R> status.vector <- c(rep(0, 90), rep(0, 30), rep(0, 67), 1)
R> status.vector <- ifelse(status.vector == 1, "i", "s")
R> init <- init.net(status.vector = status.vector)
R> control <- control.net(type = "SIR", nsteps = 55,
+                         nsims = 100, epi.by = "group")
R> epimodel_sir <- netsim(est1, param, init, control)
```

The output of this model is `epimodel_sir`, an object of class `netsim`, which contains a plethora of modeling information. [29]

4.7. Post-processing and comparison

The next step is to compare the simple SIR model to the EpiModel SIR model. We provide[30] the function `fortify_aggregate()`, which can take objects from specialized classes of modeling output (like those made by `netsim()`) and transform it into a tidy-style data frame.

```
R> fortified_net <- fortify_aggregate(epimodel_sir,
+                                     states = c("s.num", "i.num", "r.num")) %>%
+   mutate(Type = "EpiModel SIR",
+          sim = as.numeric(gsub("sim", "", sim)))
```

We can then analyze the results of the two models side by side as time-invariant[31] epidemic curves. The results are shown in Figure 11, where a 90% prediction band is estimated from the delta ball[32] method for each of the two models. For the Simple SIR model, we see that the data generally covers the data fairly well but clearly misses the second peak of infection[33]. We also see that the prediction band is very large, covering up a large area of the ternary plot. On the other hand, for the EpiModel network model, we see that the prediction band covers the data quite well and takes up less area.

```
R> both_models <- bind_rows(agg_model, fortified_net)
R>
R>
R> g <- ggplot() + geom_prediction_band(data = both_models %>% filter(t != 0) %>%
+                   mutate(Type = factor(Type, levels = c("Simple SIR",
+                                                         "EpiModel SIR"))),
+          aes(x = X0, y = X1, z = X2,
```

[29] [Ben says: what's the point for this sentence. It also doesn't flow/ connect to previous and later text.]

[30] [Ben says: the phrase "We provide" is very passive / distance from the current demonstration at hand. Moreover section 3 already phrases things this way.]more of sentence 2 of ben than 1.

[31] [Ben says: this isn't a clear phrrase here - what are you trying to say?]

[32] [Ben says: this has never been discussed anyway.]

[33] [Ben says: This could be better movitated with talk of model fit...]

```
+                sim_group = sim, fill = Type),
+           alpha = .5,
+           conf_level = .90)
```

[Ben says: In figure 8 I changed the order of the facets given we talk about the simple model first and its more like the "base" model. I think the title should be changed?]

```
R> g +   geom_path(data = both_models %>% filter(t != 0) %>%
+                    mutate(Type = factor(Type, levels = c("Simple SIR",
+                                                          "EpiModel SIR"))),
+              aes(x = X0, y = X1, z = X2, group = paste(Type, sim)),
+              alpha = .3, col = "gray40") +
+     coord_tern() + theme_sir(base_size = 24) +
+   geom_point(data = hagelloch_sir,
+              aes(x = S, y = I, z =R), col = "black") +
+   labs(title = "Simple SIR model",
+        subtitle = "90% Prediction band and original data",
+        x = "S", y = "I", z = "R") +
+   scale_fill_manual(values = c("#006677", "#AA6600")) +
+   facet_wrap(~Type) +
+   theme(legend.position = "bottom")
```

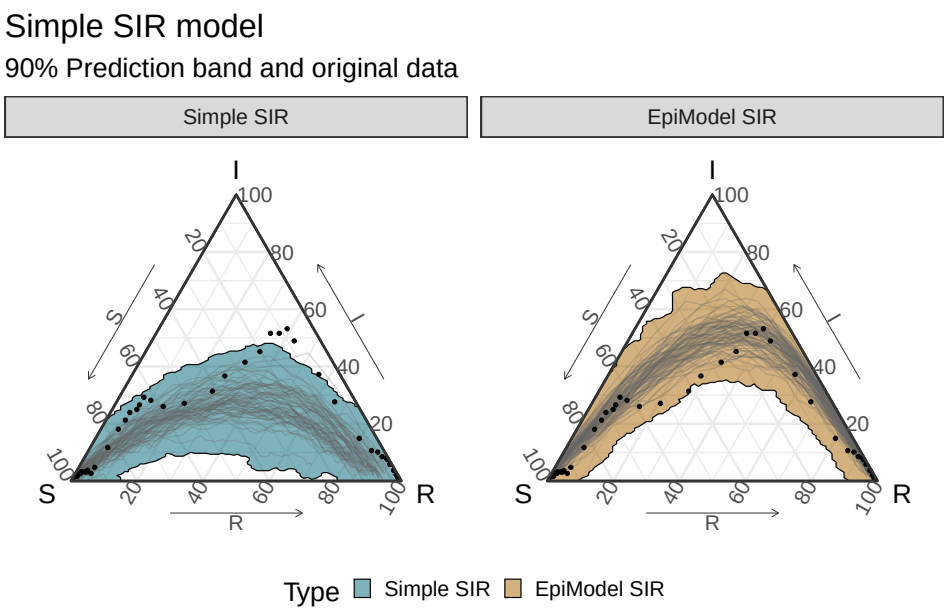

Figure 11: Original Hagelloch SIR data (black) along with 90% prediction band and actual simulation paths from the Simple SIR and the EpiModel SIR models.

However, both models are not a good fit to the filamental path as opposed to the individual points in (S, I, R)-space. This can be[34] captured with the set of simulations both models

[34] [Ben says: this is a very passive way to say such things. Try being more direct.]

predict (gray lines), which all generally have a single defined peak of infection whereas the data certainly looks like it has two distinct peaks, likely caused by our assumed super-spreader event. This observation is backed up[35] by the below analysis that demonstrates that the estimated pseudo-density of the observed epidemic (relative to the simulations from either model) is much less likely then **any** of the simulations (reported in Table 6)[36] In conclusion, **EpiCompare** makes it clear that, at a glance, 1) the EpiModel network model is a better fit than the Simple SIR model, and 2) the fit is only good at the ~~geometric filamental level as opposed to the epidemic trajectory filamental level.~~ individual point level as opposed to the geometric filamental level.[37]

```
R> #-- after cleaning up and combining --
R> all_together_df <- rbind(simple_sir,
+                           hagelloch_sir2)
```

Table 5: Top and bottom 2 rows of `all_together_df`, combining both simulated epidemics and the true epidemic.

Type	sim	t	S	I	R
Simple SIR	1	0	188	0	0
Simple SIR	1	1	187	1	0
true observation	0	54	1	0	187
true observation	0	55	1	0	187

```
R> compression_df <- all_together_df %>% group_by(Type, sim) %>%
+   filament_compression(data_columns = c("S","I","R"),
+                        number_points = 20)

R> tdmat <- compression_df %>%
+   dist_matrix_innersq_direction(
+     position = c(1:length(compression_df))[
+       names(compression_df) %in% c("S","I", "R")],
+     tdm_out = T)
R>
R> simple_sir_true_obs_info <- tdmat %>%
+   compare_new_to_rest_via_distance(
+     new_name_id = data.frame(Type = "true observation", sim = 0),
+     distance_func = distance_psuedo_density_function,
+     sigma = "20%")
```

[35] [Ben says: describe this?]

[36] Ben, do we want to add another sentence or two explaining the two columns in the table? The second one I think makes sense to me but not the first.

[37] [Ben says: how would this look with the time plots? Do we add value here?]

Table 6: The extremeness of the true simulations based on comparing pseudo-density estimates between true vs simulated curves

Type	simulations-based estimated pseudo-density	proportion of simulations with lower estimated pseudo-density
Simple SIR	0.0036733	0
EpiModel SIR	0.0028813	0

[38]Overall, **EpiCompare** aids in the data analysis pipeline for both novice and expert practitioners and coders alike. These tools encourage model and simulation exploration of many of the existing and well-supported packages that already exist, and side-by-side comparison thereof. Finally, we hope that practicioners will consider using time-invariant analysis when trying to assess and compare epidemics and epidemic models.

A. Appendix

A.1 Proof of Theorem 1

Proof. Harko *et al.* (2014) provide an analytical solution for the Kermack and McKendrick equations (Eq. (1)) by reparameterizing the ODEs so that $\mathcal{S}(u) = S(t)$, $\mathcal{I}(u) = S(t)$, and $\mathcal{R}(u) = R(t)$ for $0 < u_T < 1$ with

$$\begin{aligned} \mathcal{S}(u) &= S(0)u \\ \mathcal{I}(u) &= N - R(0) + NR_0^{-1}\log u - S(0)u \\ \mathcal{R}(u) &= R(0) - NR_0^{-1}\log u, \end{aligned} \tag{2}$$

and u and t are related by the following integral,

$$\begin{aligned} t &= \int_u^1 \frac{N}{\beta\tau(N - R(0) + R_0^{-1}\log\tau - S(0)\tau)}d\tau \\ &= \int_u^1 \frac{1}{\beta f(S(0), R(0), N, R_0, \tau)}d\tau \\ &= \int_u^1 \frac{1}{\beta f(\tau)}d\tau, \end{aligned}$$

where we have made the denominator of the integral a function of N, the initial values, R_0, and τ, which we further condense to $f(\tau)$ for brevity. Then for a given t we want to find s such that $(S_1(t), I_1(t), R_1(t)) = (S_2(s), I_2(s), R_2(s))$. Or equivalently, for a fixed u want to

[38]I think this paragraph captures some good goals, but I don't think we've done some of this. For example - we don't really highlight novice/expert usage, and we don't highlight side-by-side comparisons of models.

find v such that $\mathcal{S}_1(u) = \mathcal{S}_2(v)$ and then the corresponding t and s are given by

$$t = \int_u^1 \frac{1}{\beta_1 f(\tau)} d\tau$$
$$s = \int_v^1 \frac{1}{\beta_2 f(\tau)} d\tau.$$

Note that since the equations in Eq. (2) are functions of the initial values and R_0, then $u = v$. We then can find a relation for s,

$$\begin{aligned} s &= \int_u^1 \frac{1}{\beta_2 f(\tau)} d\tau \\ &= \int_u^1 \frac{1}{a\beta_1 f(\tau)} d\tau \\ &= \frac{1}{a} \int_u^1 \frac{1}{\beta_1 f(\tau)} d\tau \\ &= \frac{1}{a} t. \end{aligned}$$

□

References

Anderson RM, May RM (1992). *Infectious diseases of humans: dynamics and control*. Oxford university press.

Becker AD, Birger RB, Teillant A, Gastanaduy PA, Wallace GS, Grenfell BT (2016). "Estimating enhanced prevaccination measles transmission hotspots in the context of cross-scale dynamics." *Proceedings of the National Academy of Sciences*, **113**(51), 14595–14600.

Biggerstaff M, Alper D, Dredze M, Fox S, Fung ICH, Hickmann KS, Lewis B, Rosenfeld R, Shaman J, Tsou MH, Velardi P, Vespignani A, Finelli L, Chandra P, Kaup H, Krishnan R, Madhavan S, Markar A, Pashley B, Paul M, Meyers LA, Eggo R, Henderson J, Ramakrishnan A, Scott J, Singh B, Srinivasan R, Bakach I, Hao Y, Schaible BJ, Sexton JK, Del Valle SY, Deshpande A, Fairchild G, Generous N, Priedhorsky R, Hickman KS, Hyman JM, Brooks L, Farrow D, Hyun S, Tibshirani RJ, Yang W, Allen C, Aslam A, Nagel A, Stilo G, Basagni S, Zhang Q, Perra N, Chakraborty P, Butler P, Khadivi P, Ramakrishnan N, Chen J, Barrett C, Bisset K, Eubank S, Anil Kumar VS, Laskowski K, Lum K, Marathe M, Aman S, Brownstein JS, Goldstein E, Lipsitch M, Mekaru SR, Nsoesie EO, Gesualdo F, Tozzi AE, Broniatowski D, Karspeck A, Tse ZTH, Ying Y, Gambhir M, Scarpino S (2016). "Results from the centers for disease control and prevention's predict the 2013-2014 Influenza Season Challenge." *BMC Infectious Diseases*, **16**(1), 1–10. ISSN 14712334. doi:10.1186/s12879-016-1669-x. URL http://dx.doi.org/10.1186/s12879-016-1669-x.

Britton T, Kypraios T, O'Neill PD (2011). "Inference for epidemics with three levels of mixing: methodology and application to a measles outbreak." *Scandinavian Journal of Statistics*, **38**(3), 578–599.

CDC (2021). "CDC COVID Data Tracker." URL https://covid.cdc.gov/covid-data-tracker/#cases_casesper100klast7days.

Ciollaro M, Genovese CR, Wang D (2016). "Nonparametric clustering of functional data using pseudo-densities." *Electronic Journal of Statistics*, **10**(2), 2922–2972. ISSN 19357524. doi:10.1214/16-EJS1198.

Dong E, Du H, Gardner L (2020). "An interactive web-based dashboard to track COVID-19 in real time." *The Lancet infectious diseases*, **20**(5), 533–534.

Ferguson N, Laydon D, Nedjati Gilani G, Imai N, Ainslie K, Baguelin M, Bhatia S, Boonyasiri A, Cucunuba Perez Z, Cuomo-Dannenburg G, *et al.* (2020). "Report 9: Impact of non-pharmaceutical interventions (NPIs) to reduce COVID19 mortality and healthcare demand."

Gallagher S, Chang A, Eddy WF (2020). "Exploring the nuances of R0: Eight estimates and application to 2009 pandemic influenza." *arXiv preprint arXiv:2003.10442*.

Geenens G, Nieto-Reyes A (2017). "On the functional distance-based depth."

Groendyke C, Welch D, Hunter DR (2012). "A network-based analysis of the 1861 Hagelloch measles data." *Biometrics*, **68**(3), 755–765.

Hamilton NE, Ferry M (2018). "ggtern: Ternary Diagrams Using ggplot2." *Journal of Statistical Software, Code Snippets*, **87**(3), 1–17. doi:10.18637/jss.v087.c03.

Harko T, Lobo FS, Mak MK (2014). "Exact analytical solutions of the Susceptible-Infected-Recovered (SIR) epidemic model and of the SIR model with equal death and birth rates." *Applied Mathematics and Computation*, **236**, 184–194. ISSN 00963003. doi:10.1016/j.amc.2014.03.030. 1403.2160, URL http://dx.doi.org/10.1016/j.amc.2014.03.030.

Jenness SM, Goodreau SM, Morris M (2018). "EpiModel: An R Package for Mathematical Modeling of Infectious Disease over Networks." *Journal of Statistical Software*. doi:10.18637/jss.v084.i08.EpiModel.

Kermack WO, McKendrick AG (1927). "A contribution to the mathematical theory of epidemics." *Proceedings of the royal society of london. Series A, Containing papers of a mathematical and physical character*, **115**(772), 700–721.

King AA, Nguyen D, Ionides EL (2016). "Statistical inference for partially observed markov processes via the R package pomp." *Journal of Statistical Software*, **69**(12), 1–43. ISSN 15487660. doi:10.18637/jss.v069.i12. 1509.00503.

Meyer S, Held L, Höhle M (2017). "Spatio-Temporal Analysis of Epidemic Phenomena Using the R Package surveillance." *Journal of Statistical Software*, **77**(11), 1–55. doi:10.18637/jss.v077.i11.

MIDAS Network (2021). "Online Portal for COVID-19 Modeling and Research." URL https://midasnetwork.us/covid-19/.

Neal PJ, Roberts GO (2004). "Statistical inference and model selection for the 1861 Hagelloch measles epidemic." *Biostatistics*, **5**(2), 249–261. ISSN 14654644. doi:10.1093/biostatistics/5.2.249.

Oesterle H (1992). "Statistische Reanalyse einer Masernepidemie 1861 in Hagelloch."

Pfeilsticker A (1863). "Beiträge zur Pathologie der Masern mit besonderer Berücksichtigung der statistischen Verhältnisse." URL http://www.archive.org/details/beitrgezurpatho00pfeigoog.

Rvachev LA, Longini IM (1985). "A mathematical model for the global spread of influenza." *Mathematical Biosciences*, **75**(1), 3 – 22. ISSN 0025-5564. doi:http://dx.doi.org/10.1016/0025-5564(85)90064-1. URL http://www.sciencedirect.com/science/article/pii/0025556485900641.

The Washington Post (2021). "Coronavirus US Cases and." URL https://washingtonpost.com/graphics/2020/national/coronavirus-us-cases-deaths/.

Wickham H (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. ISBN 978-3-319-24277-4. URL https://ggplot2.tidyverse.org.

Wickham H, Averick M, Bryan J, Chang W, McGowan LD, François R, Grolemund G, Hayes A, Henry L, Hester J, Kuhn M, Pedersen TL, Miller E, Bache SM, Müller K, Ooms J, Robinson D, Seidel DP, Spinu V, Takahashi K, Vaughan D, Wilke C, Woo K, Yutani H (2019). "Welcome to the tidyverse." *Journal of Open Source Software*, **4**(43), 1686. doi:10.21105/joss.01686.

Worby CJ, Chaves SS, Wallinga J, Lipsitch M, Finelli L, Goldstein E (2015). "On the relative role of different age groups in influenza epidemics." *Epidemics*, **13**, 10–16.

Affiliation:

Shannon K. Gallagher
Biostatistics Research Branch
National Institute of Allergy
and Infectious Diseases
5603 Fishers Lane
Rockville, MD 20852
E-mail: shannon.gallagher@nih.gov
URL: http://skgallagher.github.io

Benjamin LeRoy
Dept. of Statistics & Data Science
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA 15213
E-mail: bpleroy@andrew.cmu.edu
URL: https://benjaminleroy.github.io/

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
MMMMMM YYYY, Volume VV, Issue II *Submitted:* yyyy-mm-dd
doi:10.18637/jss.v000.i00 *Accepted:* yyyy-mm-dd