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 enhance current infectious disease modeling analysis pipelines as well as to encourage comparisons across these pipelines. 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** encorporates R's *tidy* coding style to aid it rapid and easy use. 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. ...)[1]. At the same time, descriptive statistics and visualizations (from universities like

[1] @Shannon: CITE

John Hopkins ()[2], many branches and levels of government, and new organizations ()[3]) are an important first step of the process.

With the many visualization and exploratory tools, models and modeling paradigms, and reviews and comparisons in the literature and through MIDAS (Models of Infectious Disease Agent Study) Control Center comparison guides ()[4], this field has a lot of devices to aid an individual practitioner decide the correct approach. At the same time, analysis packages often only address a portion of the analysis pipeline. Modeling tools often don't provide easy ways to compare and assess their models on new data. Moreover, exploring and modeling 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 & 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.

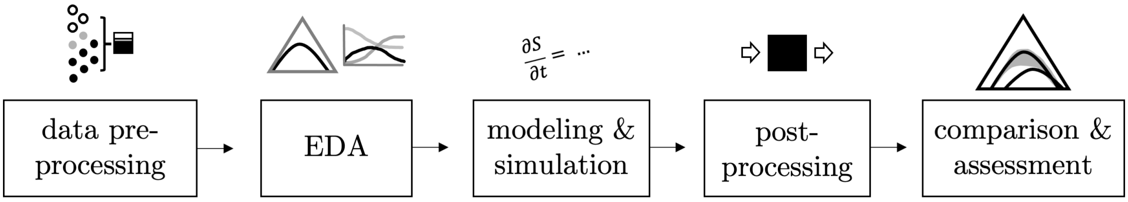

Figure 1: An idealized epidemiological data analysis pipeline.

This package also emphasizes the value of approach epidemics in a *time-invariant* way, not constrained to things like a specific time scale, or initial date of the first infection. Although epidemics are by definition a process that evolves over time, epidemics, including the recent COVID-19, often need to be compared in a time-invariant way to understand the processes at play. Additionally, many tools to examine the quantity of the population in each state along the infection process (for example: quantity of suspectible vs infected vs recovered individuals) don't always as intelligently capture the natural connections between the proportion of individuals in these states. Tools in **EpiCompare** attempt to give the user the ability to extend their toolkit to evaluate epidemics to also include time-invariant approaches. 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. flesh out time invariance introduction to better explain to non-experts what we mean. perhaps mention R0

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

[2] @Shannon: CITE
[3] @Shannon: CITE
[4] @Shannon: CITE

in every step of the pipeline and section 4 provides a thorough demonstrating of the tools through a detailed example of a full data analysis pipeline.

SHANNON: I THINK JUST INCLUDING LOTS OF LITERATURE IN INTRO + ETC. INSTEAD OF HAVING A LITERATURE REVIEW IS BETTER.

2. Motivation and tools for time-invariant analysis

Epidemics can be difficult to compare to one another due to differences in diseases, locations, or population behaviors, or times. We focus on comparisons between epidemics adjusting for time, both the relative unit of rate of infection (e.g. days, months, years) and the relative beginning and end of an epidemic. By adjusting for the unit of infection rate, we can focus on the "lifetime" of an epidemic, a view that is concerned more with the total number of lives affected than due to a time constraint. By adjusting for the relative beginning/end of an infection we can compare, for example the first part of infection from one city to another where this is some time gap in spread.

One way to adjust for both facets of time may seem obvious – scale the time between zero and one, based on the initial time of infection t_0 and the final time t_F. Upon inspection, this method of adjustment requires clear definition of what a beginning and end of an epidemic constitute which requires making one or more choices. We propose the use of time-invariant analysis as it appears in **EpiCompare**, which avoids the need of making such a choice.

2.1. What R_0 has to do with time-invariant analysis

By definition, R_0 is the number of expected secondary infections when a primary infection is introduced to a susceptible population. R_0 is also, maybe, the most famous *time-invariant* numerical summary of an epidemic, which allows epidemics to be compared to one another in both different time and geographic scales. For example, R_0 for Covid-19 is estimated to be between 2-3, seasonal influenza between 1.2-2, and modern measles outbreaks between 5-6.

Estimator(s) for R_0 are dependent on the epidemic modeling framework, which consists of which states an individual can occupy (e.g. susceptible, infectious, recovered) and a description of how individuals move from one state to the next over time. A common epidemic modeling framework is the SIR model, originally introduced by Kermack and McKendrick (1927). Transitions from one state to the next are defined by a series of ordinary differential equations, where N is the (fixed) total number individuals, β is the rate of infection, and γ is rate of recovery,

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

From this, $\hat{R}_0 = \frac{\hat{\beta}}{\hat{\gamma}}$, the ratio of the estimated infection rate compared to the estimated recovery rate.

With regards to traditional epidemic *state* vs. *time* plots, R_0 is difficult to visualize, especially with respect from one epidemic to another. For example, consider the scenarios where the first epidemic is generated from a SIR model with $(S(0) = 990, I(0) = 10)$, $\beta_1 = 0.3$ and $\gamma_1 = 0.15$, and the second epidemic is generated from a SIR model with $(S(0) = 990, I(0) = 10)$, $\beta_2 = 0.24$ and $\gamma_1 = 0.12$ over 40 days. Both epidemics have the same value of $R_0 = \beta_1/\gamma_1 = \beta_2/\gamma_2 = 2$. The epidemic trajectories are shown in the *state* vs. time plots in Figure **??**. At a glance, we may assume that Model 1 has a larger R_0 than Model 2 because the peak of infection occurs more quickly than in Model 2. On the other hand, we may think Model 2 has a larger R_0 because it has a slightly larger peak of infection than Model 1.

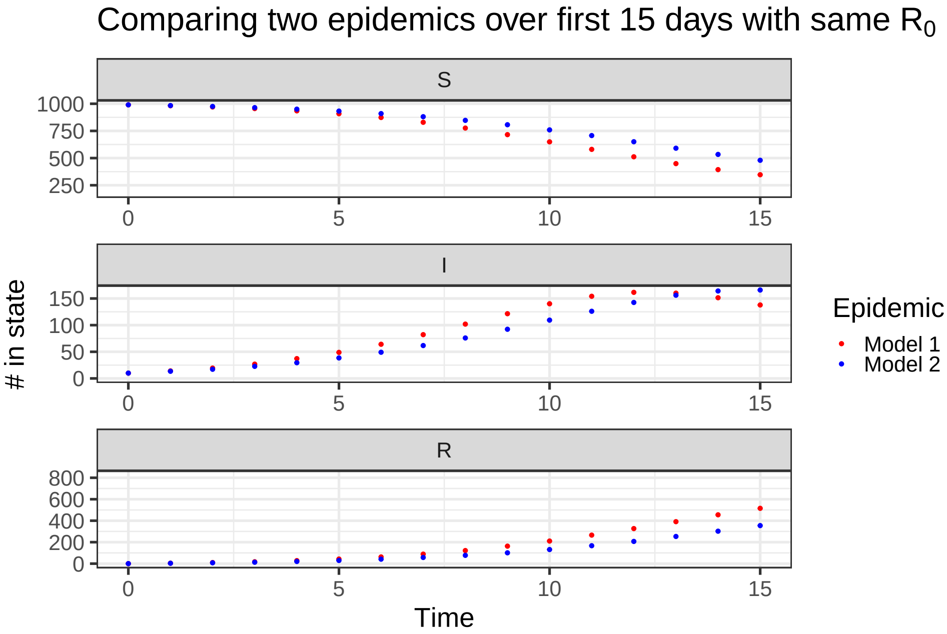

Figure 2: Example of two epidemics with different β and γ paremeters but the same initial reproduction number $R_0 = 2$. Both plots are generated from models with $N = 1000$ individuals with $S(0) = 990$ and $I(0) = 10$.

However, when we plot the trajectories as a single curve using the ternary plot in the time-invariant view, we immediately see a different story. In this time-invariant view in Fig. **??**, the points seem to overlap and form the same trajectory. In other words, we can see there is something fundamentally linking these two different epidemics, and this fundamental link turns out to be R_0.

More formally, let two Kermack and McKendrick (see Eq. (1)) SIR models be denoted $(S_1(t), I_1(t), R_1(t))$ and $(S_2(t), I_2(t), R_2(t))$, respectively, for $t > 0$. Assume both models have initial values $(S(0), I(0), R(0))$. Let $R_0 = \frac{\beta_1}{\gamma_1} = \frac{\beta_2}{\gamma_2}$ where β_i and γ_i are the average infection rate and recovery rate, respectively, for SIR model $i = 1, 2$. Equivalently, $\beta_2 = a\beta_1$ if and only if $\gamma_2 = a\gamma_1$ for some $a > 0$.

Theorem 1 *Let there be two SIR models as described above. Then for all $t > 0$ there exists an s 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 Appendix **??**. 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 for every point on the epidemic path of the first SIR model is also a point on the epidemic path of the second SIR model. Taking the sample simulations from Fig. 2, Fig. 3 presents these two models in a ternary plot. This means, that at a glance, we can tell if two epidemics have different values of R_0.

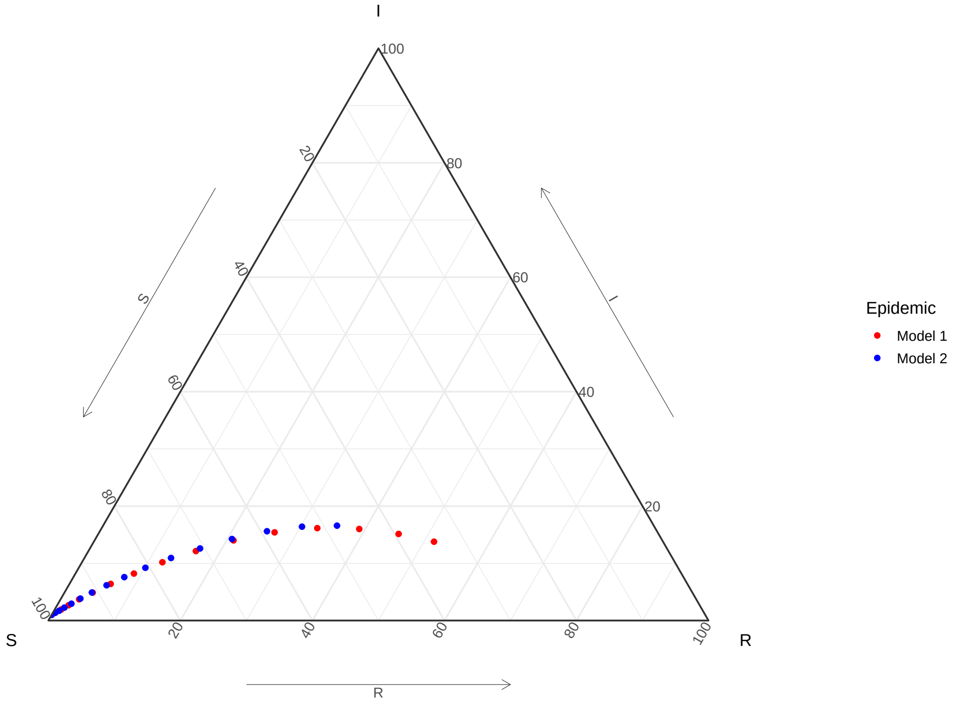

Figure 3: Example of two epidemics with different β and γ paremeters but the same initial reproduction number $R_0 = 2$. Both plots are generated from models with $N = 1000$ individuals with $S(0) = 990$ and $I(0) = 10$. These are plotted in the time-invariant view, where we can see the number of susceptible, infectious, and recovered.

2.2. Beyond the Kermack and McKendrick SIR models

Although the result of Theorem 1 allows for easy visual comparison of R_0 in Kermack SIR models, it does require stringent assumptions of homogeneity of behavior in populations. The use of visualizing epidemics in a time-invariant lens via ternary plots extends beyond those of models that follow the ODEs in the Kermack-McKendrick equations. Any model with S, I, and R states can be visualized with ternary plots, regardless of birth and death dynamics and regardless of homogeneity of individual behavior. We can use ternary plots to compare the spread of a disease for groups within a population without time as a confounding factor. We show an example of this in a later section.

But time-invariant analysis is also useful for epidemic models with more than three states. The constraints in three dimensions that are met with the SIR model (that is $\sum_{i=1}^{3}(\text{number in state}(i)) = N(t)$) actually represents a space of 3d simplices, and the ternary plot specifically represents these simplices, after scaling to examine such values as proportions between 0 and 1 (ternary plots are known as a 3d unit simplex due to it's scaling). This same scaling for larger models (i.e. with more states) can be done onto different simplexes. In this package we present tools to help compare models (mostly through simulations). s tool compares these objects after projecting them into a one-dimension-fewer space through the simplexical structure of the data.

In **EpiCompare** we allow for the comparison epidemics in these higher dimensional spaces by first projecting onto these simplexes. Even though higher dimensional models may not be able to visualized, we provide multiple tools to aid in the comparison of models and epidemics. The first of which uses multiple simulations under specific model parameters to assess the bairabilty of the model fit. In **EpiCompare** we provide ways to create prediction regions for a true epidemic under the model assumptions using these simulations. These regions require representing multi-dimensional structures for functions to completely contain epidemics, and treat these simulations and epidmeics as filamental objects. We extend off of papers like Dalmasso *et al.* (2019) to create these bands. These high dimensional bands allow the user to assess if the true epidemic is within the band (there-by assessing the model's representation of the epidemic), and also compare who different models are from each other through distances that compare sets. We recommend using the Hausdorff distance to compare such sets as it captures how much bigger the sets would have to expand to cover each other, and is defined mathematically as

$$d_{\text{Hausdorff}}(S_1, S_2) = \max \left\{ \sup_{x \in S_1} \inf_{y \in S_2} d(x,y) \ , \ \sup_{y \in S_2} \inf_{x \in S_1} d(x,y) \right\} .$$

3. Package tool overview

The goals of this section is to present parts of the data science pipeline, introduce how we help, why they are useful and the ideas behind it.

In this section we will present the tools in this package and how they aid in the data analysis pipeline. We present a modification of Figure 1 from the Section 1 with Figure 4 with uses of our package. We recommend the package user keep this image around as a reference. All user functions are aimed to be as friendly as possible, and we focus on providing the user "tidyverse" style functions, that encourage piping and also follow clear verb naming schemes (Wickham *et al.* 2019). There are 2 different ways **EpiCompare** can be incorporated in the data analysis pipeline for epidemics, either at the very beginning when pre-processing data and visualizing raw data, or after modeling has been done and a generative model is fit. Figure 4 captures these different paths, and we will highlight both approaches and how to leverage **EpiCompare** in the subsections below.

3.1. Data Preprocessing

The first step of most data analysis is cleaning up the data to be explored. There are multiple ways to collect the epidemiological data. Sometimes individual records with times of different states of the epidemic (infection, recovery, etc.) as well as individual information like network structure, location, and-sub population information will be collected, whereas other data collections will focus population/sub-population counts of individuals in each epidemic state[5].

[5] Shannon include citations

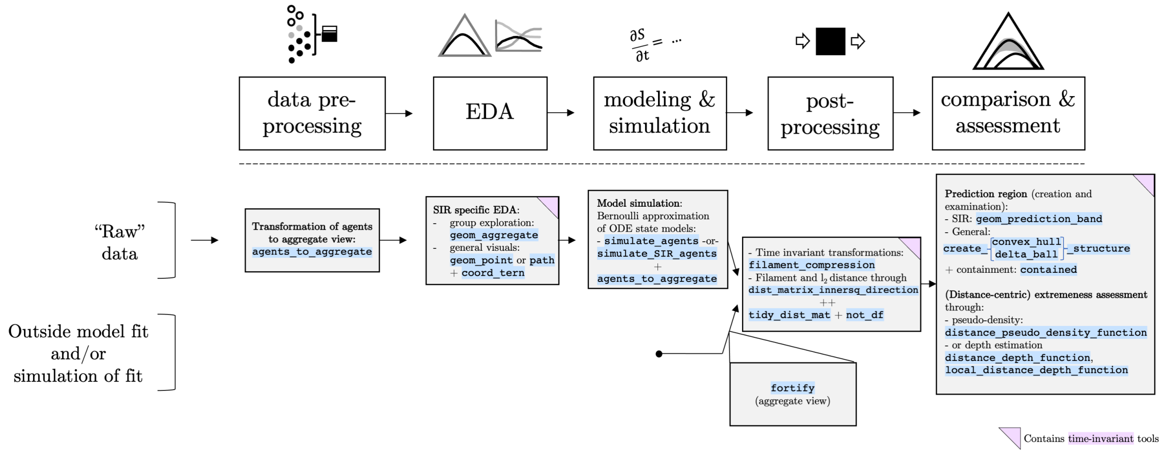
data pre-processing
EDA
modeling & simulation
post-processing
comparison & assessment
$\frac{\partial S}{\partial t} = \cdots$
"Raw" data
Outside model fit and/or simulation of fit
Transformation of agents to aggregate view: agents_to_aggregate
SIR specific EDA:
- group exploration: geom_aggregate
- general visuals: geom_point or path + coord_tern
Model simulation: Bernoulli approximation of ODE state models:
- simulate_agents -or- simulate_SIR_agents + agents_to_aggregate
- Time invariant transformations: filament_compression
- Filament and l_2 distance through dist_matrix_innersq_direction ++ tidy_dist_mat + not_df
fortify (aggregate view)
Prediction region (creation and examination):
- SIR: geom_prediction_band
- General: create_{convex_hull / delta_ball}_structure
+ containment: contained
(Distance-centric) extremeness assessment through:
- pseudo-density: distance_pseudo_density_function
- or depth estimation distance_depth_function, local_distance_depth_function
Contains time-invariant tools

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

We focus on understanding the overall impact on the population/sub-populations we provide a function to transform information about each agent's start time of each state (e.g. start of infection, etc). This transformation between from agent information to aggregate information is very useful for seeing the overall trend of the epidemic (and how it impacts different subpopulations). Our tools aim to allow the user to easily group agents and define new subpopulations to explore. This is important as many case studies have highlighted the usefulness to identify differing subpopulations[6] and many state based models provide for subpopulation based states in their analysis[7].

We provide a "tidyverse"-styled function, `agents_to_aggregate` to transform agent information into aggregate state information, As a "tidy" function, our function `agents_to_aggregate` allows the user to identify any subpopulations by first doing `group_by` from `dplyr`, and also allows for infinite epidemic states - which can be for models with assumptions of sub population based states (e.g. youth infections), as well as indicators for death and birth dates. Our function, `agents_to_aggregate` allows for states to be skipped - captured with `NA` values and multiple starts of a state - (e.g. multiple infections) as long as the original data is stored in columns that are constrained to ordering ideas[8]. Currently this function is constrained to integer time steps (for example days), but transformations (linear or otherwise) of the time columns can make the time steps more (or less) . `agents_to_aggregate` then returns a standardized count for each state for each time point. We see `agents_to_aggregate` as a really powerful tool, but also very useful to quickly transform agents information to be used in aggregate state-based models and compare state-based models to observed data.

3.2. EDA

With raw data, getting to know our data now-a-days very frequently means figuring out

[6] Shannon cite

[7] Shannon cite

[8] make this more clear?

good combinations of visualizations, numerical summaries and subsetting. An expert coder can start from `agents_to_aggregate` to successfully do this in many ways, but we've also developed tools to rapidly explore your data if your have a 3 unique states models (like the SIR model). Our `geom_aggregate` provides a rapid way to explore different subpopulations experience of the epidemic, and combines the ideas behind agents_to_aggregate for the SIR case, with `geom_path` and `coord_term` to visualize any number of groups epidemic trajectory in 3d simplex space using `ggplot2` and `ggtern` (Wickham 2016; Hamilton and Ferry 2018). Visualization tools for SIR models were developed because (1) SIR models are the most common and basic state-based models[9] and (2) our simplex representation of these epidemics emphasizes a "time-invarance" representation of the data (for a refresher see Section 2).

3.3. Model Fitting and Simulations

Although this package does not focus on estimating a model for the data, we do provide some power functions for simulation of basic state models with Bernoulli approximation of ODE state models encoded in simulate_agents and simulate_SIR_agents (which can naturally be combined with agents_to_aggregate). **Description of these models, reference to Shannon R0 paper, etc...**

3.4. Post-processing

...

3.5. Comparions and Assessment

...

4. A tour of EpiCompare

In this section, we highlight a number of the functionalities available in **EpiCompare**. These functionalities include data cleaning, visualization, simulation, and comparison, in accordance with the data analysis pipeline 1. We show a full data analysis from beginning to end that can be accomplished in a streamlined and standardized manner.

4.1. Data and exploratory analysis

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). The Hagelloch data includes a rich set of features including 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. Because of these rich features, this data set has been an ideal testing ground methodology in infectious disease epidemiology and is used in work by Neal and Roberts (2004); Britton *et al.* (2011); Groendyke *et al.* (2012); Becker *et al.* (2016).

With **EpiCompare**, we can easily obtain the empirical cumulative incidence function with respect to the measles rash appearance (variable `ERU`) with the following tidy-style function,

[9] Shannon: cite

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

`agents_to_aggregate`. The function `agents_to_aggregate` is a key component of **EpiCompare**, allowing the user to easily switch from an individual-level (i.e. an agent) view of a disease to an aggregate level. For example, the below code shows how we can convert the agent data to a cumulative incidence 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 symptoms. We do this with the below code, and a part of the cumulative incidence data output are shown in Table 2. 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 2: 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 question of interest is the duration between initial onset of prodromes or symptoms and the appearance of the measles rash. Since `agent_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. 5).

```
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")

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"))
```

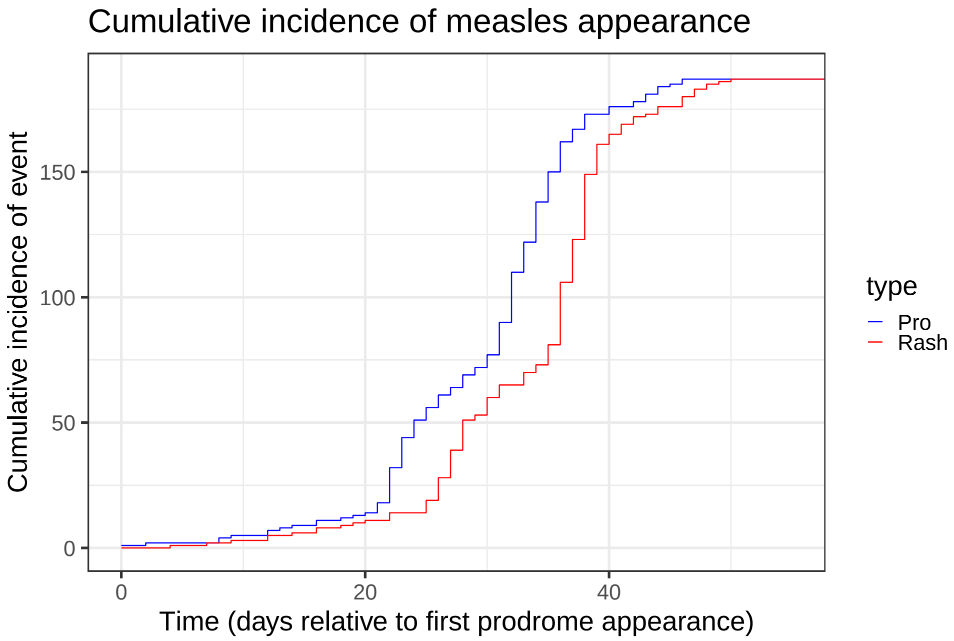

Figure 5: Empirical cumulative incidence functions of prodrome (symptom) onset and measles rash appearance. We see that there is approximately a a constant lag between the two curves.

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 plot the SIR values through a time-invariant lens using **ggplot2** and **ggtern** functions (as shown in Fig. 6) 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)
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)
```

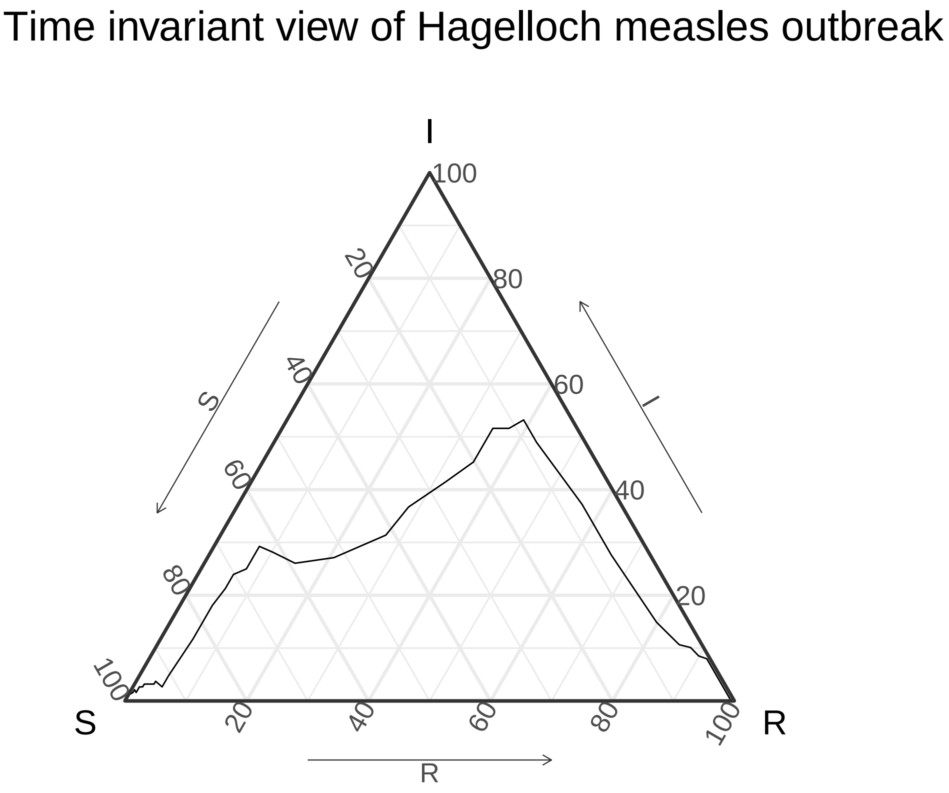

Figure 6: Time invariant view of the Hagelloch epidemic where we view the individuals in Susceptible, Infectious, or Recovered states. We see there are two peaks of infection (the vertical axis).

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 7. 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 is 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.

```
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)
```

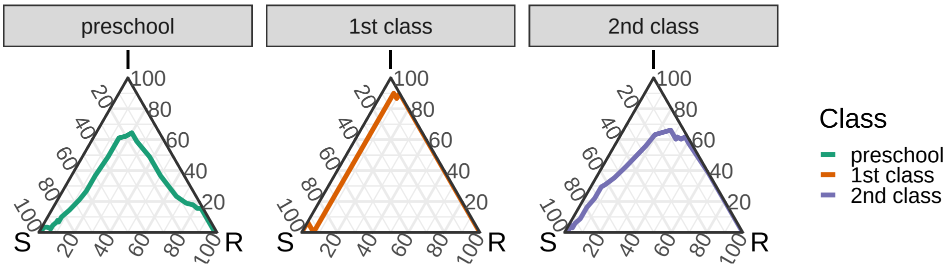

Figure 7: Time invariant outbreak curves for the three class groups. The pre-school class has a distinct peak of infection whereas the peak infection point for the other two classes are less well defined.

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.

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

We first try to model the Hagelloch data with an SIR model (see Eq. **??**). In our vignette, we show how to fit a stochastic model via maximum likelihood and simulate from the model with those best fit parameters. Our function `simulate_agents()` generates individual level data according to discrete 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 diseease 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 which details the time of entry into the S, I, and R states for a given simulation. 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**. Briefly, this model first fits a contact network to the set of indivduals, where the class of the student is a covariate. 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
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. We provide the function `fortify_aggregate()`, which can take objects from specialized classes of modeling output 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 epidemic curves. The results are shown in Figure 8, where a 90% prediction band is estimated from the delta ball 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. 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** 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),
+          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),
+             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")
```

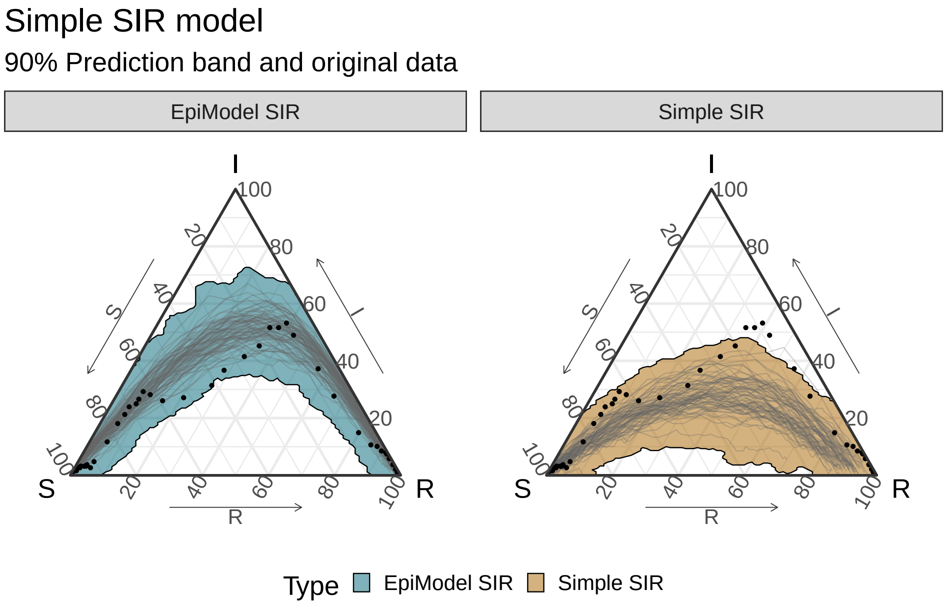

Figure 8: Original Hagelloch SIR data (black) along with 90% prediction band and actual simulation paths from the Simple SIR and the EpiModel SIR models.

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 captures with the set of simulations both models predict, 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 by the below analysis that demonstrates that the estimated psuedo-density of the observed epidemic (relative to the simulations from either model) is much less likely then **any** of the simulations (reported in Table 4. 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 individual point level as opposed to the epidemic path level.

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

Table 3: Top and bottom 2 rows of `all_together_df, combining both simulated epidemics and the tr`

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 4: The extremeness of the true simulations based on comparing psuedo-density estimates between true vs simulated curves

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

5. Helpful Rmd tricks

5.1. RMD Code formatting info

This is the Section 1. This template demonstrates some of the basic LaTeX that you need to know to create a JSS article.

In general, don't use Markdown, but use the more precise LaTeX commands instead:

- Java
- **plyr**

One exception is inline code, which can be written inside a pair of backticks (i.e., using the Markdown syntax).

If you want to use LaTeX commands in headers, you need to provide a `short-title` attribute. You can also provide a custom identifier if necessary. See the header of Section 5.2 for example.

5.2. RMD R code

Can be inserted in regular R markdown blocks.

hags hags hags Neal and Roberts (2004)

```
R> x <- 1:10
R> x
```

```
 [1]  1  2  3  4  5  6  7  8  9 10
```

References

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.

Dalmasso N, Dunn R, LeRoy B, Schafer C (2019). "A Flexible Pipeline for Prediction of Tropical Cyclone Paths." 1906.08832, URL http://arxiv.org/abs/1906.08832.

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.

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.

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.

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.

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.

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