



---

# *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  
Carnegie Mellon University

---

### Abstract

The abstract of the article.

*Keywords:* keywords, not capitalized, Java.

---

## 1. Introduction

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

### 1.1. Code formatting

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 2 for example.

## 2. 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
```

### 3. 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 REF??. We show a full data analysis from beginning to end that can be accomplished in a streamlined and standardized manner.

#### 3.1. 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).

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

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, `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 <- haggelloch_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 Haggelloch 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. 1).

```
R> cif_prodromes <- haggelloch_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"))
```

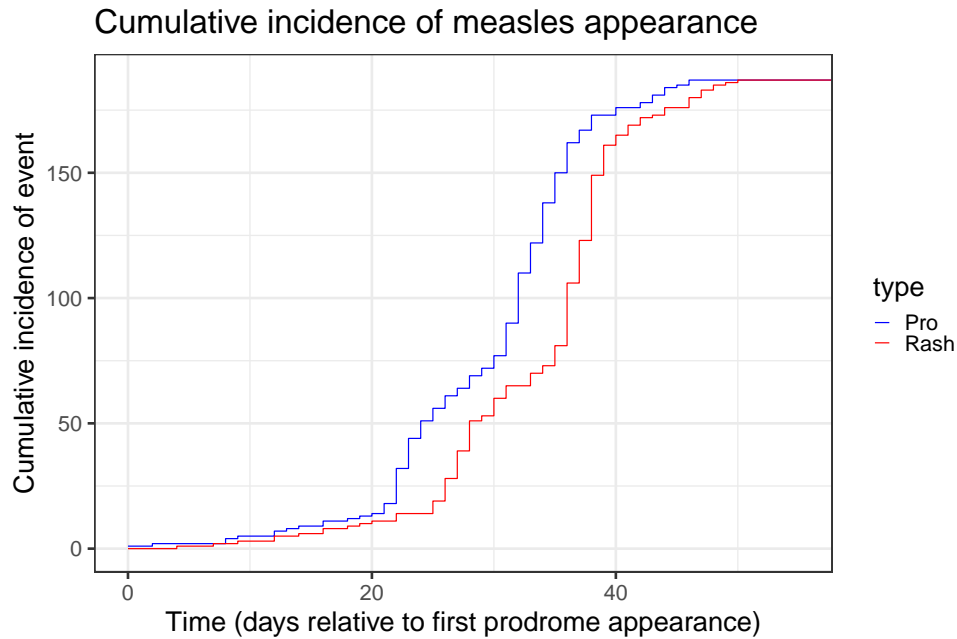


Figure 1: Empirical cumulative incidence functions of prodrome (symptom) onset and measles rash appearance. We see that there is approximately a constant lag between the two curves.

```
R> with(hagelloch_raw, sum(PRO > ERU, na.rm = TRUE))
```

```
[1] 0
```

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` functions or with our custom `geom`, `geom_aggregate`, which takes the original agent data as input (Fig. 2).

```
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)
```

## Time invariant view of Hagelloch measles outbreak

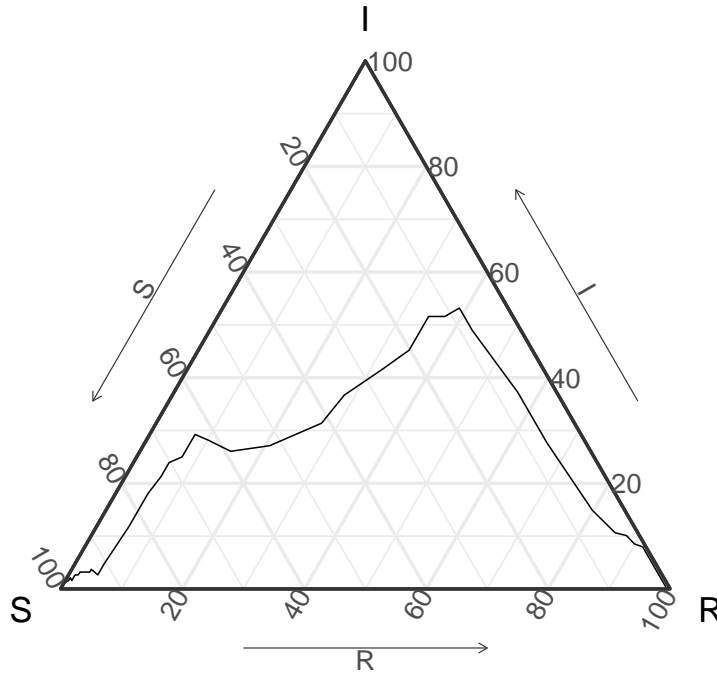


Figure 2: Time invariant view of the Hagelloch epidemic where we view the individuals in Susceptible, Infectious, or Recovered states. We see there are two peaks of infection (the vertical axis).

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 3. 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)
```

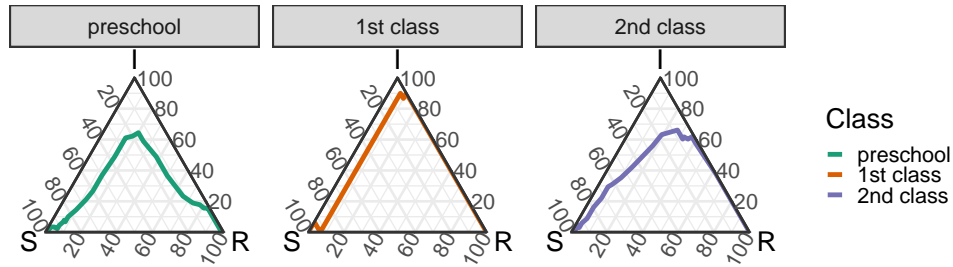


Figure 3: Time invariant outbreak curves for the three class groups. The pre-school class has a distinct peak of infection whereas the peak infection point for the other two classes are less well defined.

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 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

```

      S              I              R
S "X0 * (1 - X1 * par1 / N)" "X0 * X1 * par1 / N" "0"
I "0"                      "X1 * (1 - par2)"      "par2 * X1"
R "0"                      "0"                    "X2"

```

```

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  $18800 \times 5$  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 fits a contact network, where the class of the student is a covariate and 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 4, 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")
```

### Simple SIR model

90% Prediction band and original data

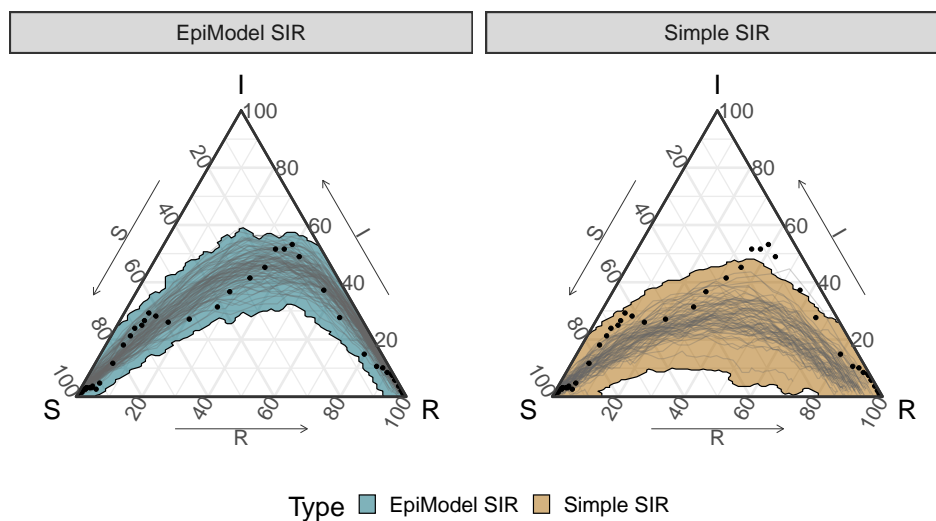


Figure 4: Original Hagelloch SIR data (black) along with 90% prediction band and actual simulation paths from the Simple SIR and the EpiModel SIR models.



However, both models are not a good fit to the filamental path as opposed to the individual points in  $(S, I, R)$ -space. What we mean by that is that both models predict epidemic paths that 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. 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.

HERE BEN IS GOING TO LOOK AT THE EPIMODEL OUTPUT DIFFERENT PREDICTION BAND THINGS.

- thing

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

**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](mailto:shannon.gallagher@nih.gov)  
URL: <http://skgallagher.github.io>

Benjamin Leroy  
Dept. of Statistics  
Carnegie Mellon University  
5000 Forbes Ave.  
Pittsburgh, PA 15213  
E-mail: [bpleroy@andrew.cmu.edu](mailto:bpleroy@andrew.cmu.edu)  
URL: <https://benjaminleroy.github.io/>