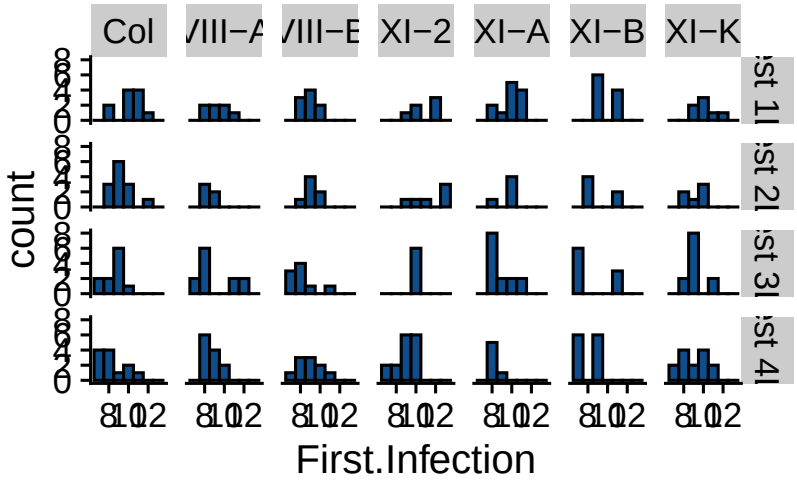
Classical Survival Analysis

Kevin Schoelz

August 2019

We have an experiment where we are looking at the behavior of different ecotypes of a certain species of plant. In the experiment, the plant are inoculated by a virus. They are then observed once a day for several days to see when virus symptoms become apparent. There is reason to believe that maybe some of the ecotypes can slow the infection of the plant.

We can start by loading up the data. In the grid of plots below, we are looking at the number of new infections observed on a given day. The data has been divided up so that different columns of plots show the behavior of a given ecotype. The control plants are in the left-most column. The different rows correspond to different trials. Each trial consists of plants that were inoculated at the same time. We expect that the different trials are probably not directly comparable due to uncontrolled environmental effects.



Typically, to characterize a dataset like this, we should do a survival analysis.

Terminology of a Survival analysis

In a survival analysis, we have some object, (people, plants, machine etc. . .) that can undergo some type of event. In a survival analysis, our goal is to estimate the time until the event happens. The canonical example would be to analyze mortality curves, i.e. to estimate time until death for people who undergo some type of treatment. But the mathematical framework is more general than that. We can

use survival analysis to analyze time until a disease shows up (as we will do here), or even time until a given part fails on a machine. It is a very general and powerful framework. In survival analysis, we are trying to estimate a series of functions. We have the probability distribution function $f(y)$ which gives the probability of a new infection occuring on day y. There is the cumulative distribution function, $F(y)$ which gives the probability that the event has occured by day y. Finally we have the survival function $S(y) = 1 - F(y)$ which gives the probability of a given individual surviving until day y. Classically, the goal of a survival analysis is to estimate the survival function $S(y)$.

Estimating the survival function

How do we estimate the survival function? This is a big question, and there are a lot of different approaches. Let's generate some fake survival data to try and get a better feel for how to answer this question. Suppose we have some time until event data. This data is drawn from the following true survival function, as given below.

We want to estimate the survival function from our data. We can use the `rweibull` function to draw samples from a Weibull distribution. Results are stored in the table below.

ysim	new_infections	n_plants	n_infected_total	n_healthy	surv_prob	cum_surv_prob	proportion
7	1	20	1	20	0.9500000	0.95	0.95
8	2	20	3	19	0.8947368	0.85	0.85
9	8	20	11	17	0.5294118	0.45	0.45
10	8	20	19	9	0.1111111	0.05	0.05
11	1	20	20	1	0.0000000	0.00	0.00

```
## Warning: Removed 60 rows containing missing
## values (geom_path).
```

A plot of the Survival function is shown in the margin. The dashed line shows the "true" survival function, while the solid line shows the Kaplan-Meier estimate as generated from the observed data.

Fitting the Plant data using the `survival` package.

Using the Kaplan-Meier estimate of the survival function

We want to fit the plant data. We need to fit separately by ecotype and trial. Data has been stored in in the dataframe `df`.

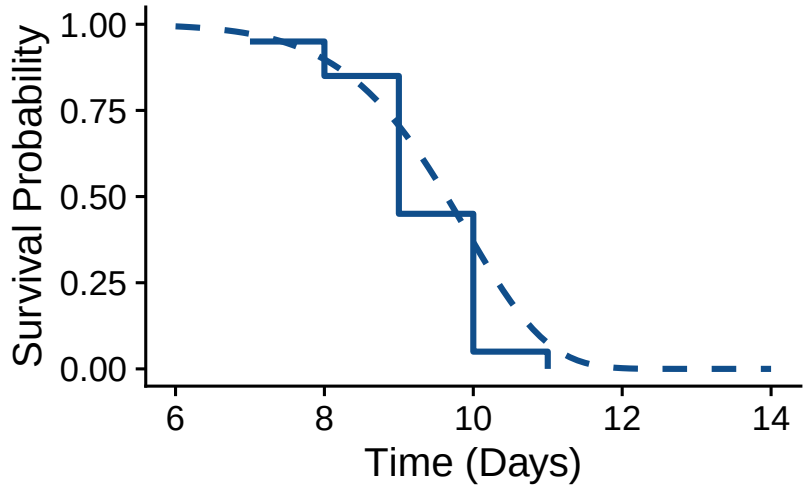
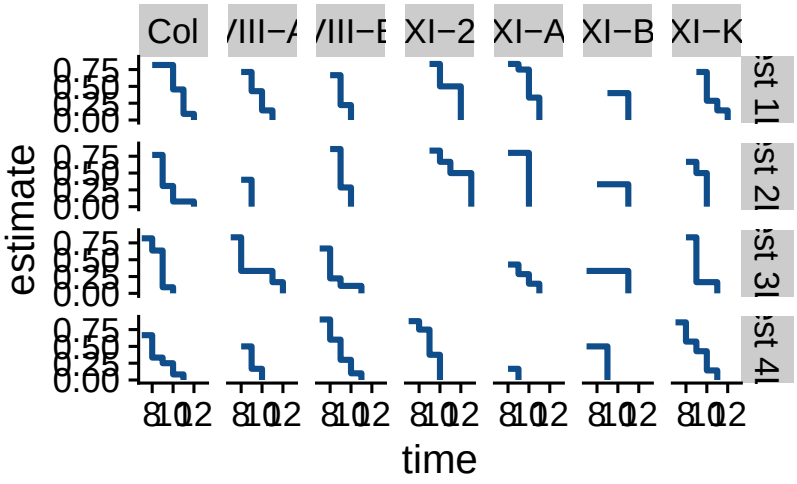


Figure 1: Solid line indicates the Kaplan-Meier estimate of the survival curve. Dashed line indicates 'true' survival curve.

The following plot shows the Kaplan-Meier estimate of the survival function for each ecotype in each trial. Control plants for each trial have been included as well. We calculate p-values from a chi-squared test in

```
# Define kaplan meier fitting function using
# survival package
kaplan_meier_model <- function(df) {
    surv_obj <- Surv(time = df$First.Infection)
    surv_fit <- survfit(surv_obj ~ 1, data = df)
    surv_fit <- tidy(surv_fit)
}

kaplan_meier_model_stats <- function(df) {
    surv_obj <- Surv(time = df$First.Infection)
    surv_fit <- survfit(surv_obj ~ 1, data = df)
    surv_fit <- glance(surv_fit)
}

# Apply to trials in original dataframe
nested_km_df <- df %>% group_by(Trial.Name, Ecotype) %>%
    nest() %>% mutate(model = map(data, kaplan_meier_model),
    model_stats = map(data, kaplan_meier_model_stats))

km_models_df <- nested_km_df %>% unnest(model)

# models_df
```

```
ggplot(data = km_models_df, aes(x = time, y = estimate,
    color = Ecotype)) + geom_step(size = 1, color = "DodgerBlue4") +
    facet_grid(Trial.Name ~ Ecotype)

## geom_path: Each group consists of only one
## observation. Do you need to adjust the
## group aesthetic?
```




```
# ggsurvplot_list(fit = nested_df$model, data
# <- )
```

Using the Cox proportional hazards model

Another technique we can use to fit the data uses the Cox proportional hazards model. The Kaplan-Meier estimator is a non-parametric estimate of the survival function. It is extraordinarily flexible method, however, it does not allow for the use of covariates.

Another approach is the Cox proportional hazards method. In the Cox proportional hazards method, we can estimate a Hazard function

$$h(y) = h_0 \exp(\beta_0 + X\beta)$$

The Survival function is then given by

$$S(y) = \exp\left(\int_0^t h(u)du\right)$$

The `survival` package can be used estimate the coefficients for the Cox proportional hazard models.

```
coxph_model <- function(df) {
    res.cox <- coxph(Surv(time = First.Infection) ~
```

```
        Ecotype, data = df)
    cox_summary <- summary(res.cox)
}

nested_coxph_df <- df %>% group_by(Trial.Name) %>%
    nest() %>% mutate(coxph_models = map(data,
    coxph_model))

# str(nested_coxph_df$coxph_models[1])
# nested_coxph_df$coxph_models[1]
```