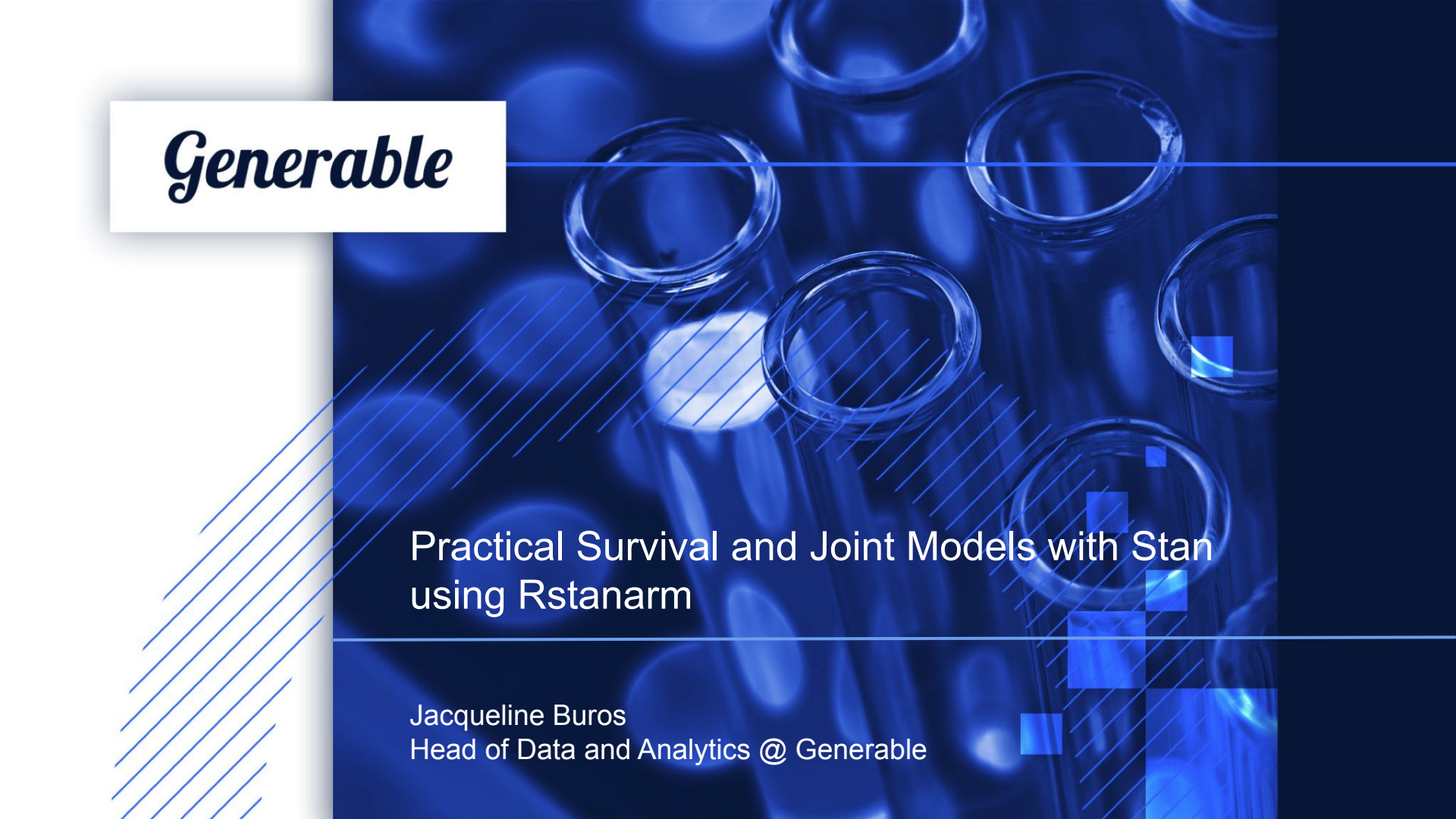

Generable
Practical Survival and Joint Models with Stan using Rstanarm
Jacqueline Buros
Head of Data and Analytics @ Generable

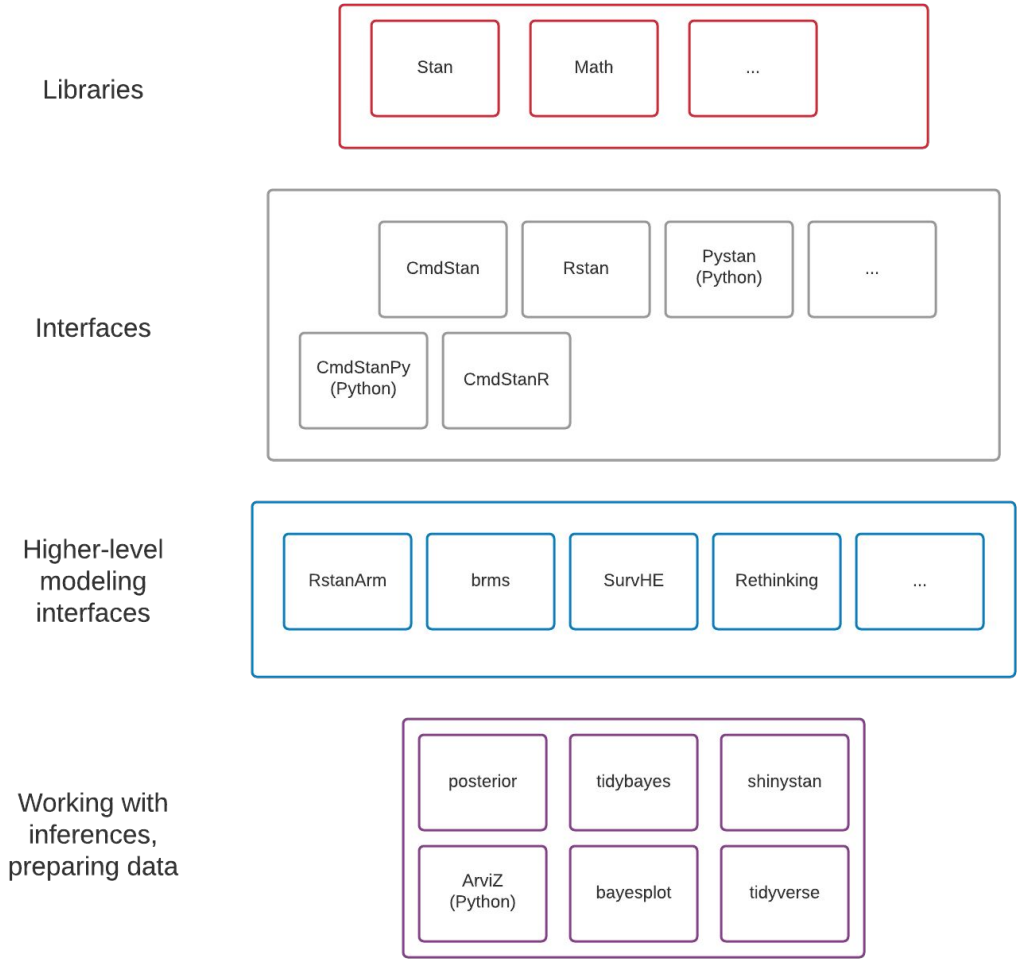
Course Materials

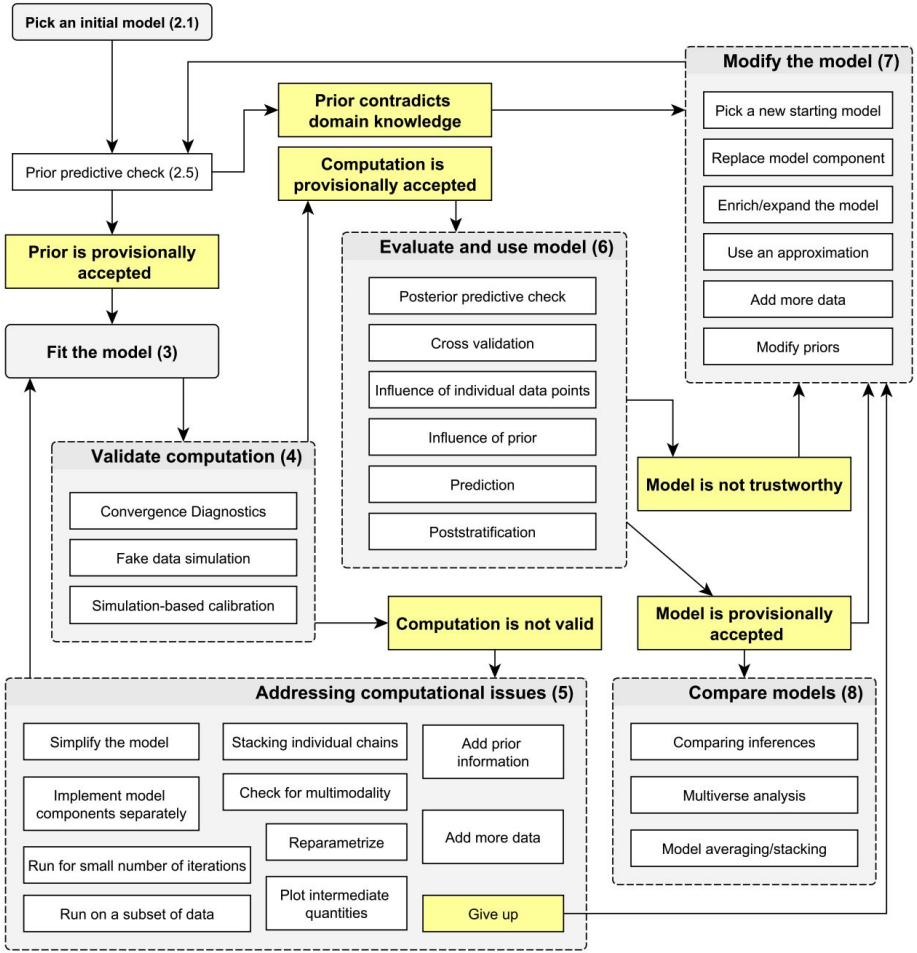
- Link to slides
- Github repository: https://github.com/generable/ncb2021
- Instructions for setting up Rstudio + RstanArm + Rstan. Test your setup.

Agenda

- Part I: Basics
  - Bayesian workflow
  - Using Stan
- Part II: Survival analysis
  - Introduction to survival models
  - Investigating priors
  - Survival analysis using stan_surv
  - Model extensions
- Part III: Joint Model analysis
  - Introduction to Joint Models
  - Investigating priors
  - Fitting a JM using stan_jm
  - Model extensions
- Conclusions

Part I: Basics


Libraries
Stan
Math
...
Interfaces
CmdStan
Rstan
Pystan
(Python)
...
CmdStanPy
(Python)
CmdStanR
Higher-level
modeling
interfaces
RstanArm
brms
SurvHE
Rethinking
...
Working with
inferences,
preparing data
posterior
tidybayes
shinystan
ArviZ
(Python)
bayesplot
tidyverse



Source: https://arxiv.org/pdf/2011.01808.pdf

Bayesian Machinery: Notation

y: measured observations

$\tilde{y}$: future observations, predictions

x: covariates, everything is $p(\cdot|x)$

θ: unknown, unobservable parameters

d: decisions or actions

Bayesian Machinery: Solving the Inverse Problem

- The joint probability of data **y** and unknown parameter vector theta:

$$p(y, \theta) = p(y|\theta) * p(\theta)$$

$$p(y, \theta) = p(\theta|y) * p(y)$$

- The conditional probability of the parameter given data:

Likelihood Prior

$$p(\theta|y) = \frac{p(y|\theta) * p(\theta)}{p(y)} = \frac{p(y|\theta) * p(\theta)}{\int p(y, \theta) d\theta} = \frac{p(y|\theta) * p(\theta)}{\int p(y|\theta) * p(\theta) d\theta}$$

Marginal Likelihood

$$\propto p(y|\theta) * p(\theta)$$

Bayesian Machinery: Prior Predictive Distribution

- For a given prior, **p(theta)** and the data model **p(y|theta)**, we can construct the prior predictive distribution as follows:

$$p(y) = \int p(\theta)p(y|\theta)d\theta$$

- Observe that we are not conditioning on the observed data y-tilde
- We do this, to assess that our model and prior generate reasonable observations
- This often helps us to tune our priors (e.g. restrict them to something reasonable)

Bayesian Machinery: Posterior Predictive Distribution

- Once we observe y-tilde and obtain the posterior p(theta|y), we can obtain the posterior predictive distribution:

$$p(\tilde{y}|y) = \int p(\tilde{y}|\theta)p(\theta|y)d\theta$$

- This allows us to assess if the model agrees with observations and make probabilistic predictions for new data.

Bayesian Machinery in Stan

Knowns: X, y

Unknowns: θ

Model: $\log(p(y|X,\theta)) + \log(p(\theta))$

Predictions: $p(\tilde{y}|y)$

```
data {
  int<lower=1>  N;
  int<lower=1>  K;
  matrix[N, K] X;
  vector[N] y;
}
parameters {
  real alpha;
  vector[K] beta;
  real<lower=0> sigma;
}
model {
  y ~ normal(X * beta + alpha, sigma);
  alpha ~ normal(0,10);
  beta ~ normal(0,10);
  sigma ~ cauchy(0,10);
}
generated quantities {
  vector[N] y_rep;
  for (n in 1:N)
    y_rep[n] = normal_rng(X[n,] * beta +
                          alpha, sigma);
}
```

Further resources

Stan

- Stan Development Team. 2020. “RStan: The R Interface to Stan.” July 26, 2020. https://cran.r-project.org/web/packages/rstan/vignettes/rstan.html.

Workflow

- Gelman, Andrew, Aki Vehtari, Daniel Simpson, Charles C. Margossian, Bob Carpenter, Yuling Yao, Lauren Kennedy, Jonah Gabry, Paul-Christian Bürkner, and Martin Modrák. 2020. “Bayesian Workflow.” *arXiv [stat.ME]*. arXiv. http://arxiv.org/abs/2011.01808.

Survival Analysis

- Paul Lambert, 2018. "Standardized survival curves and related measures from flexible survival parametric models," London Stata Conference 2018 14, Stata Users Group.

Part II: Survival Models

The data

Typically, we are interested in analyzing data for a set of subjects, where each is followed from a defined entry time t_0 until either an event is observed or the event information is censored.

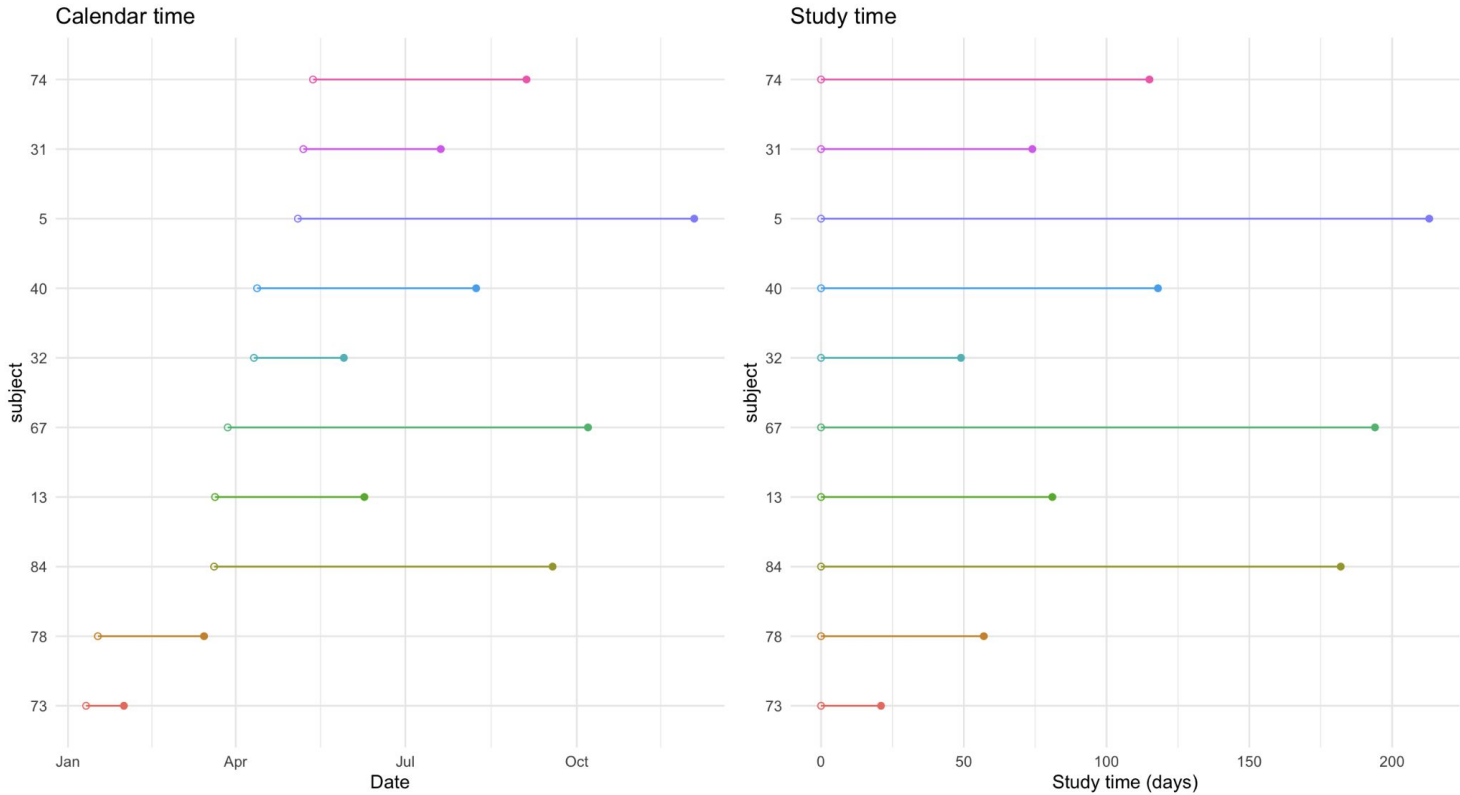
The event data are often represented as a tuple:

- Event time: $t_i = \min[t_{event}, t_{censor}]$
- Event status: $d_i \in \{0, 1\}$ denoting an event indicator
  - taking value 0 if individual *i* was right censored (i.e. $t_{event} > t_i$), and
  - taking value 1 if individual *i* was uncensored (i.e. $t_{event} = t_i$).

However, the entry time can vary across subjects, and events can be left censored or interval-censored. There can be gaps during which it is known that the subject is not at risk.

Nonetheless, at each study-time, the event rate is the portion of subjects at risk with an event observed.

The data


Calendar time
74
31
5
40
32
67
13
84
78
73
subject
Jan
Apr
Jul
Oct
Date
Study time
74
31
5
40
32
67
13
84
78
73
subject
0
50
100
150
200
Study time (days)

Terminology

In this course, I will use the language of “survival analysis”, but the terms are flexible.

Subjects are the primary unit of analysis, and defines the unit at which events are observed and counted. The name is derived from medical applications of survival analysis, but a subject can be anything: machines, customers, patients, etc.

Failure/survival can be any binary terminal event. Time to machine failure, time to conception, or time to purchase are examples of survival events. Failure doesn’t always have to be bad, and survival doesn’t always have to be good. But the failure event *should* be the event of interest.

Time doesn’t have to be measured in calendar time, but often is. It should be the unit of measurement over which the risk of failure events accumulates. For example, we will be working with data where time is measured in machine cycles.

Basic model

There are three components to the survival model:

- Hazard (h): instantaneous hazard at time t (events / time)
- Cumulative hazard (H): cumulative hazard up to time t (events)
- Survival probability (S): probability of surviving up to time t (probability)

Hazard rate for a subject i to have an event at time t

$$h_i(t) = \lim_{\Delta t \to 0} \frac{P(t \leq T_i^* < t + \Delta t | T_i^* > t)}{\Delta t}$$

Hazard rate for a subject i to have an event at time t

$$h_i(t) = \lim_{\Delta t \to 0} \frac{P(t \leq T_i^* < t + \Delta t | T_i^* > t)}{\Delta t}$$

True event time T_i^* within interval $[t, t+\Delta t)$

Hazard rate for a subject i to have an event at time t

$$h_i(t) = \lim_{\Delta t \to 0} \frac{P(t \leq T_i^* < t + \Delta t \mid T_i^* > t)}{\Delta t}$$

True event time T_i^* within interval $[t, t+\Delta t)$

Condition on true event time T_i^* being after t

Hazard rate for a subject i to have an event at time t

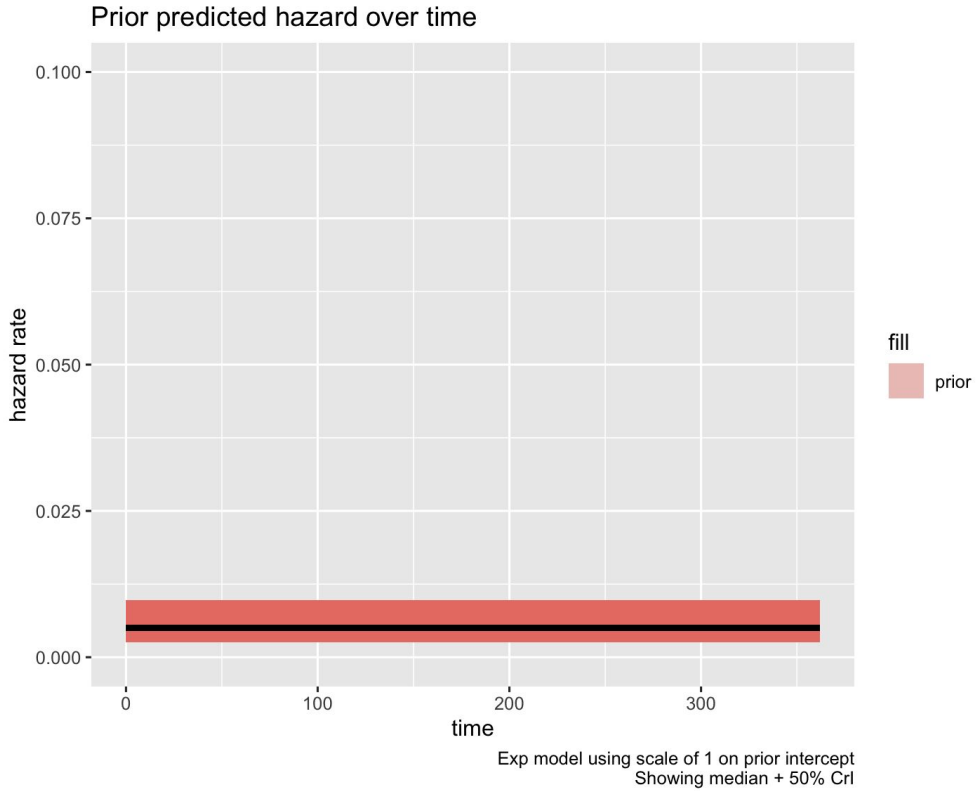
$$h_i(t) = \lim_{\Delta t \to 0} \frac{P(t \leq T_i^* < t + \Delta t \mid T_i^* > t)}{\Delta t}$$

True event time T_i^* within interval $[t, t+\Delta t)$

Condition on true event time T_i^* being after t

Convert to a rate: events per unit time

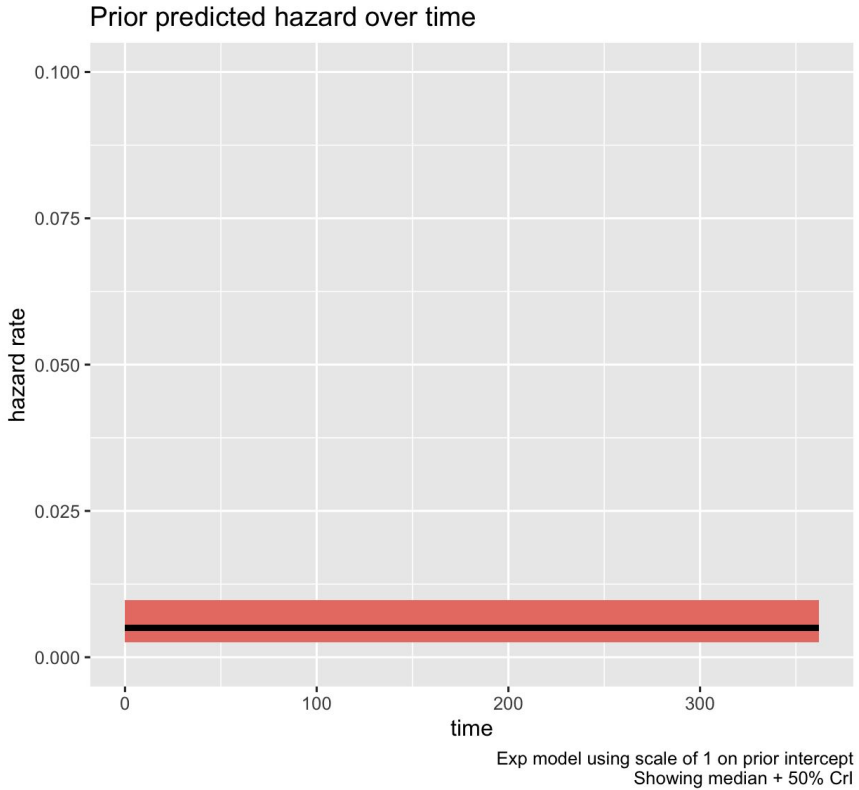
For example: a constant hazard

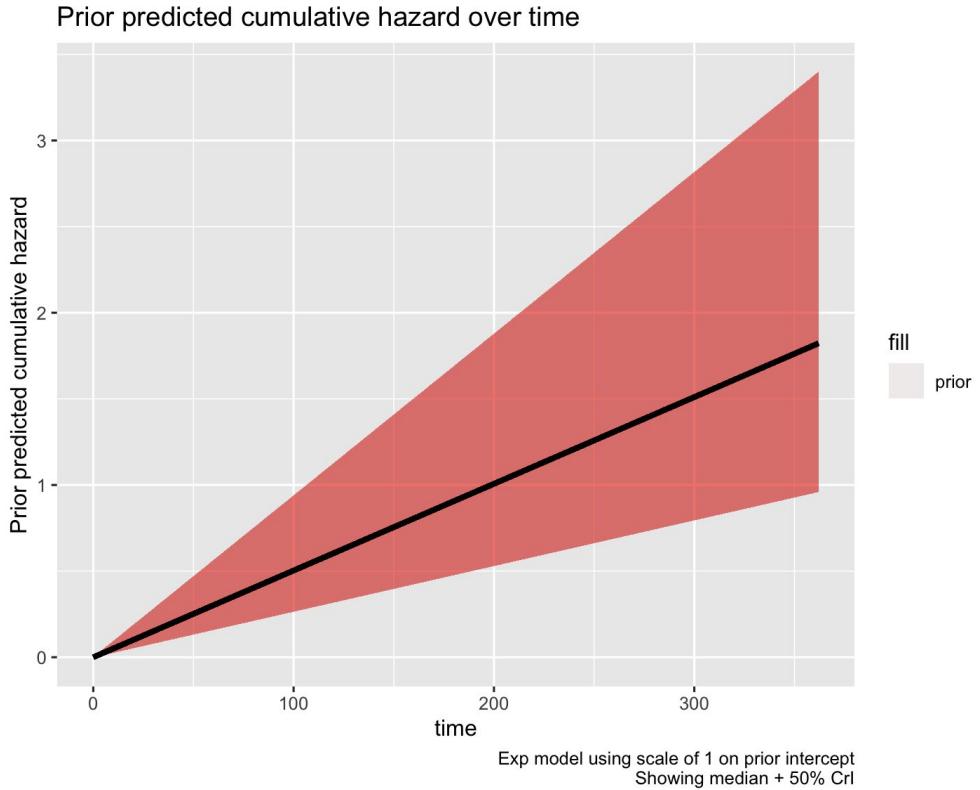


The cumulative hazard is defined as:

$$H_i(t) = \int_{u=0}^{t} h_i(u)du$$

For example: a constant hazard





Linking hazard to survival

Given a function for Survival ***S***, which is the probability of surviving to time *t*:

$$S(t) = Pr(T_i^* > t)$$

And a function for the instantaneous *hazard* ***h***, i.e. the probability of an event occurring in the interval [*t*, *t*+Δ*t*), given that a patient has survived to time *t*:

$$h(t) = \lim_{\Delta t \to 0} \frac{Pr(t \leq T_i^* < t + \Delta t \mid T_i^* > t)}{\Delta t}$$

We can equivalently write

$$h(t) = \frac{-S'(t)}{S(t)}$$

Linking hazard to survival

Given a function for Survival ***S***, which is the probability of surviving to time *t*:

$$S(t) = Pr(T_i^* > t)$$

And a function for the instantaneous *hazard* ***h***, i.e. the probability of an event occurring in the interval [*t*, *t*+Δ*t*), given that a patient has survived to time *t*:

$$h(t) = \lim_{\Delta t \to 0} \frac{Pr(t \leq T_i^* < t + \Delta t \mid T_i^* > t)}{\Delta t}$$

We can equivalently write

$$h(t) = \frac{-S'(t)}{S(t)}$$

Change in survival probability at time t

Probability of surviving to time t

Linking hazard to survival

Solving this yields:

$$S(t) = \exp\left(-\int_0^t h(z)\,dz\right)$$

Linking hazard to survival

Solving this yields:

$$S(t) = \exp\left(-\int_0^t h(z)\,dz\right)$$

Cumulative hazard up to time t

Linking hazard to survival

Solving this yields:

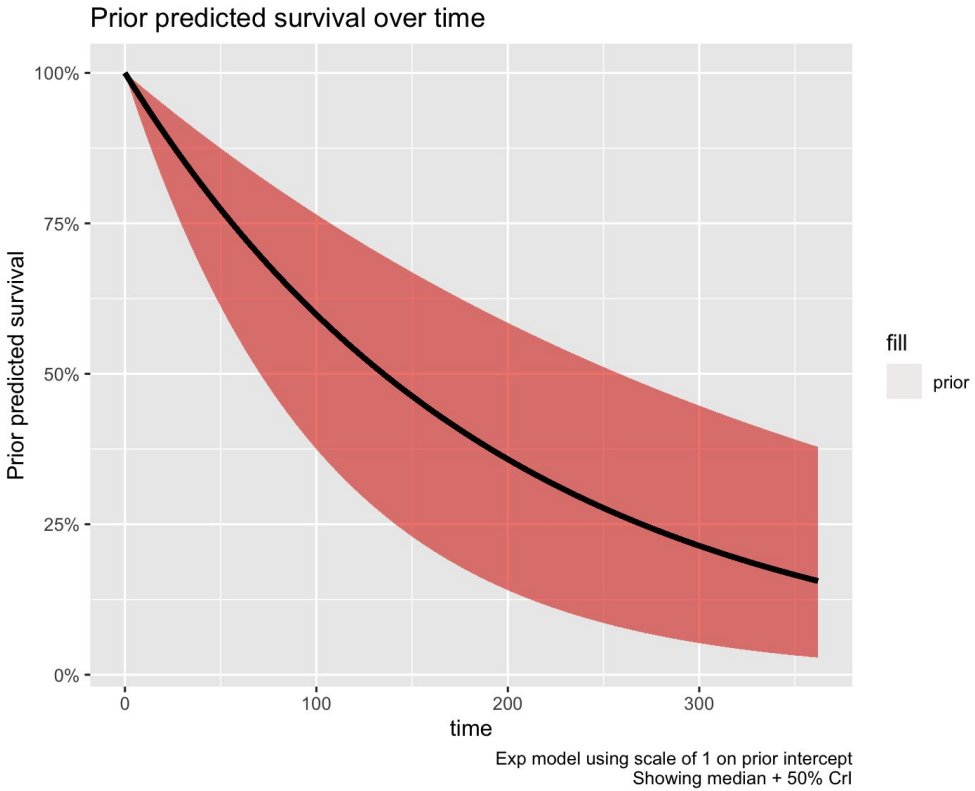
$$S(t) = \exp\left(-\int_0^t h(z)\,dz\right)$$

Which is equivalent to

$$S(t) = \exp(-H(t))$$

Cumulative hazard up to time t

$$S_i(t) = \exp\left(-H_i(t)\right) = \exp\left(-\int_{u=0}^{t} h_i(u)du\right)$$



Typical strategy for modeling survival

- Define model for $h_i(t)$: hazard for subject *i* at each survival time *t*
- Compute $H_i(t_i)$: cumulative hazard for subject *i* up to observed time t_i
  - Integration can be done analytically (fast) or by numerical integration (slower)
- Compute $S_i(t_i)$: probability of subject *i* surviving up to time t_i
- Compute $p(y_i|\theta_i,X_i)$: probability of observing our data given θ_i and X_i
  - If event was censored: $S_i(t_i)$
  - If event occurred: $S_i(t_i) \times h_i(t_i)$

Typical strategy for modeling survival

- Define model for $h_i(t)$: hazard for subject *i* at each survival time *t*

- Compute $H_i(t_i)$: cumulative hazard for subject *i* up to observed time t_i
  - Integration can be done analytically (fast) or by numerical integration (slower)
- Compute $S_i(t_i)$: probability of subject *i* surviving up to time t_i
- Compute $p(y_i|\theta_i,X_i)$: probability of observing our data given θ_i and X_i
  - If event was censored: $S_i(t_i)$
  - If event occurred: $S_i(t_i) \times h_i(t_i)$

Thank you Stan/Sam!

A few comments

We prefer models for hazard that are easy to integrate. We can largely ignore this integration as an implementation detail.

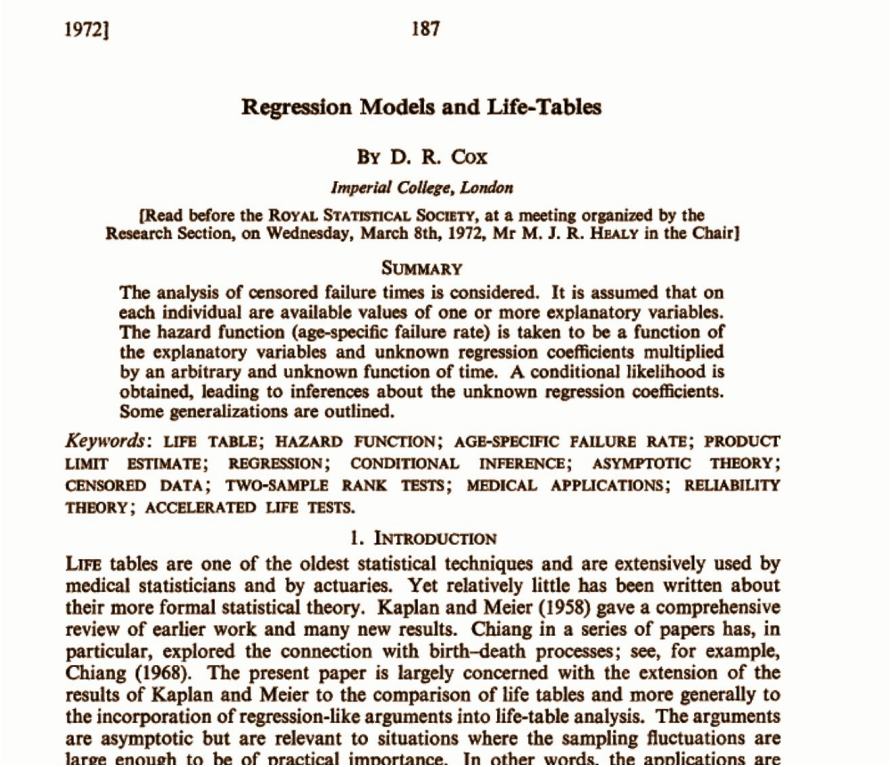
BUT

1. Know when you are working with numerical integration. The models will sample more slowly. An equivalent model that samples faster will be better.

2. You can control the number of integration points, etc. Tune these settings according to your data and modeling goals.

The Cox Model

$$h_i(t \mid x_i) = h_0(t)\, exp(x_i \beta)$$



1972] 187

Regression Models and Life-Tables

By D. R. Cox

Imperial College, London

[Read before the Royal Statistical Society, at a meeting organized by the Research Section, on Wednesday, March 8th, 1972, Mr M. J. R. Healy in the Chair]

Summary

The analysis of censored failure times is considered. It is assumed that on each individual are available values of one or more explanatory variables. The hazard function (age-specific failure rate) is taken to be a function of the explanatory variables and unknown regression coefficients multiplied by an arbitrary and unknown function of time. A conditional likelihood is obtained, leading to inferences about the unknown regression coefficients. Some generalizations are outlined.

Keywords: LIFE TABLE; HAZARD FUNCTION; AGE-SPECIFIC FAILURE RATE; PRODUCT LIMIT ESTIMATE; REGRESSION; CONDITIONAL INFERENCE; ASYMPTOTIC THEORY; CENSORED DATA; TWO-SAMPLE RANK TESTS; MEDICAL APPLICATIONS; RELIABILITY THEORY; ACCELERATED LIFE TESTS.

1. Introduction

Life tables are one of the oldest statistical techniques and are extensively used by medical statisticians and by actuaries. Yet relatively little has been written about their more formal statistical theory. Kaplan and Meier (1958) gave a comprehensive review of earlier work and many new results. Chiang in a series of papers has, in particular, explored the connection with birth–death processes; see, for example, Chiang (1968). The present paper is largely concerned with the extension of the results of Kaplan and Meier to the comparison of life tables and more generally to the incorporation of regression-like arguments into life-table analysis. The arguments are asymptotic but are relevant to situations where the sampling fluctuations are large enough to be of practical importance. In other words, the applications are

The Cox Model

$$h_i(t \mid x_i) = h_0(t)\, exp(x_i \beta)$$

The Cox Model

$$h_i(t \mid x_i) = h_0(t)\, exp(x_i \beta)$$

"Baseline hazard"

- Describes variation in hazard over time
- Typically shared among subjects

"Relative hazard"

- Describes variation according to covariates
- Typically constant over time

The Cox Model

$$\log\left(h_i(t \mid x_i) \right) = \log\left(h_0(t) \right) + x_i\beta$$

"Baseline hazard"

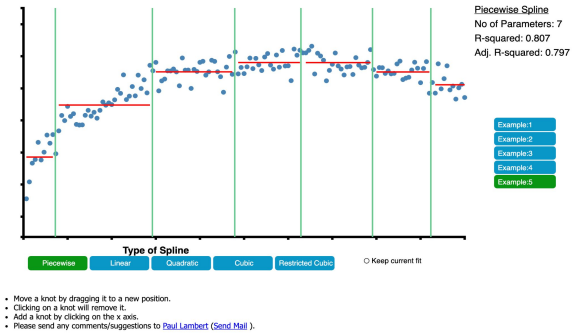
- Describes variation in hazard over time
- Typically shared among subjects

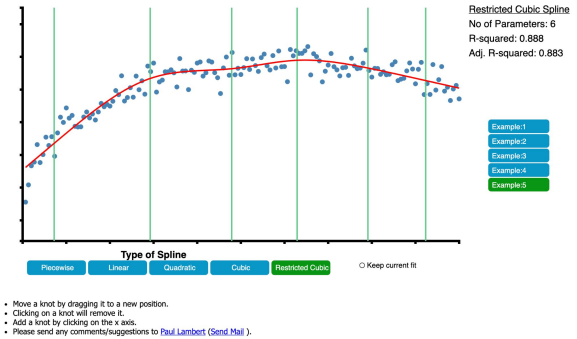
"Relative hazard"

- Describes variation in hazard according to covariates
- Typically constant over time

Parametric and semi-parametric baseline hazards

- Constant baseline hazard (exponential survival)
- Parametric model for survival times, like weibull, gompertz, etc.
- Piecewise constant baseline hazard
- Smoothed spline over time
- Penalized spline over time
- …

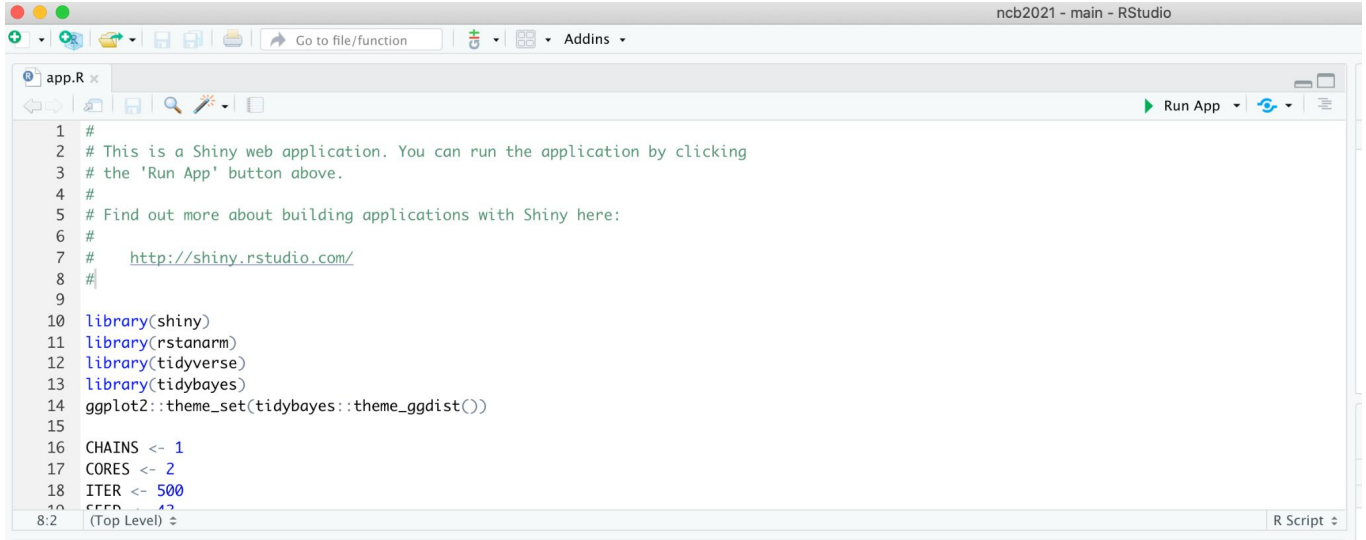




Source: https://pclambert.net/interactivegraphs/spline_eg/spline_eg

Demo

1. Open Rstudio
2. Clone the github repository: https://github.com/generable/ncb2021
3. Open the file: `hazard_simulator/app.R`
4. Click on `Run App`



Likelihood

An individual contribution to the likelihood depends on whether the subject is censored at time t_i or is observed.

- If the event is observed at time t_i: $L_i = S(t_i)\, h(t_i)$
- If the event is censored at time t_i: $L_i = S(t_i)$

Example Survival Analysis

The scenario

We will investigate data from NASA, tracking time to turbofan failure

Here is a link to the data: here.

Steps:

1. Open Rstudio
2. Clone the github repository: https://github.com/generable/ncb2021
3. Open the `01-EDA.R`

The training data

- 100 machines followed until failure event
- We are told that time is in cycles.
- 3 settings tracked at each cycle. Presumably these are exogenous.
- 21 sensors recorded at each cycle. Presumably these are endogenous.

Download the data if you want to follow along.

The training data

```
> head(d)
# A tibble: 6 x 54
     id  time setting1 setting2 setting3 sensor1 sensor2 sensor3 sensor4 sensor5 sensor6 sensor7 sensor8 sensor9
  <dbl> <dbl>    <dbl>    <dbl>    <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
1     1     1  -0.0007  -0.0004      100    519.    642.   1590.   1401.    14.6    21.6    554.   2388.   9046.
2     1     2   0.0019  -0.0003      100    519.    642.   1592.   1403.    14.6    21.6    554.   2388.   9044.
3     1     3  -0.0043   0.0003      100    519.    642.   1588.   1404.    14.6    21.6    554.   2388.   9053.
4     1     4   0.0007   0           100    519.    642.   1583.   1402.    14.6    21.6    554.   2388.   9049.
5     1     5  -0.0019  -0.0002      100    519.    642.   1583.   1406.    14.6    21.6    554    2388.   9055.
6     1     6  -0.0043  -0.0001      100    519.    642.   1584.   1398.    14.6    21.6    555.   2388.   9050.
# … with 40 more variables: sensor10 <dbl>, sensor11 <dbl>, sensor12 <dbl>, sensor13 <dbl>, sensor14 <dbl>,
#   sensor15 <dbl>, sensor16 <dbl>, sensor17 <dbl>, sensor18 <dbl>, sensor19 <dbl>, sensor20 <dbl>, sensor21 <dbl>,
#   setting1_normalized <dbl>, setting2_normalized <dbl>, setting3_normalized <dbl>, sensor1_normalized <dbl>,
#   sensor2_normalized <dbl>, sensor3_normalized <dbl>, sensor4_normalized <dbl>, sensor5_normalized <dbl>,
#   sensor6_normalized <dbl>, sensor7_normalized <dbl>, sensor8_normalized <dbl>, sensor9_normalized <dbl>,
#   sensor10_normalized <dbl>, sensor11_normalized <dbl>, sensor12_normalized <dbl>, sensor13_normalized <dbl>,
#   sensor14_normalized <dbl>, sensor15_normalized <dbl>, sensor16_normalized <dbl>, sensor17_normalized <dbl>,
#   sensor18_normalized <dbl>, sensor19_normalized <dbl>, sensor20_normalized <dbl>, sensor21_normalized <dbl>,
#   start <dbl>, end <dbl>, event <lgl>, group <chr>
```

Training data transformed into durations

```
> head(events)
# A tibble: 6 x 5
     id group    start   end event
  <dbl> <chr>    <dbl> <dbl> <lgl>
1     1 training     0   192 TRUE
2     2 training     0   287 TRUE
3     3 training     0   179 TRUE
4     4 training     0   189 TRUE
5     5 training     0   269 TRUE
6     6 training     0   188 TRUE
```

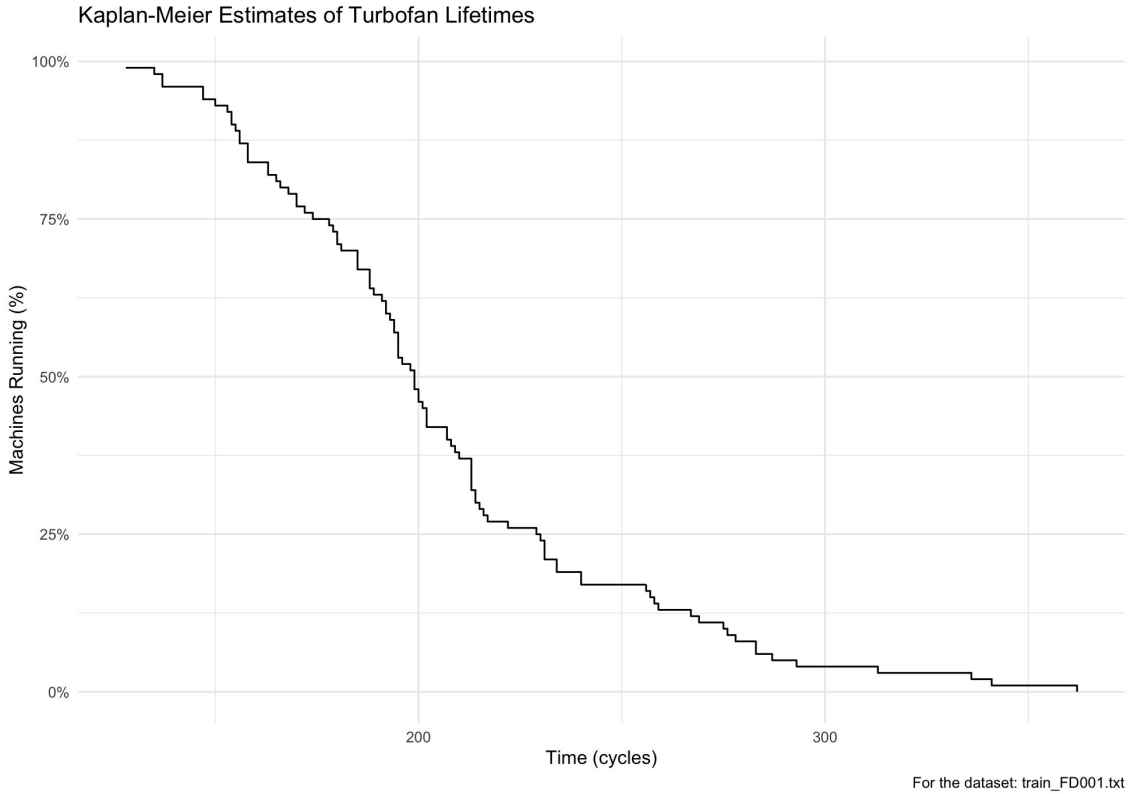
Kaplan-Meier estimates from training data

- Median survival: ~200 cycles
- First failure at ~120 cycles
- Last failure at ~350 cycles

This is a non-parametric estimate of the Survival probability over time.

It is a summary of the event rate as a percentage of the remaining, at-risk subjects at each time.

See this article for a nice explanation of K-M curves



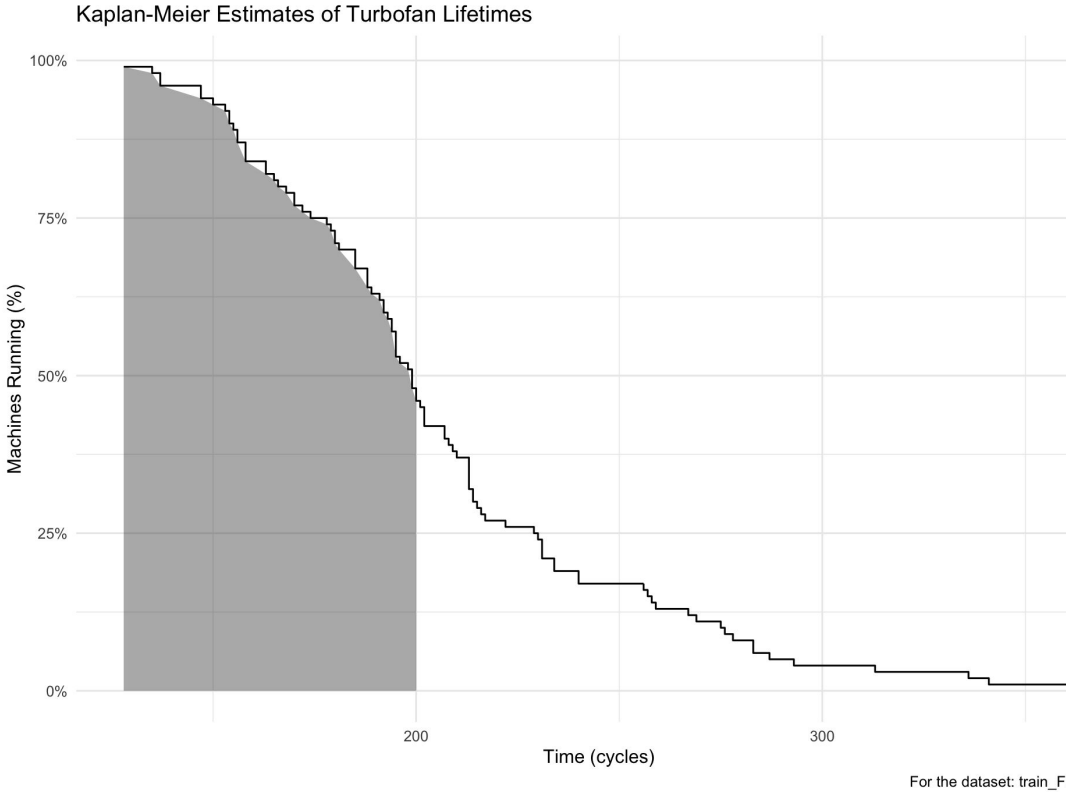
Restricted Mean Survival Time (RMST)

Expected survival time up to a cutoff time tau

$$RMST(\tau) = E[min(T, \tau)]$$

Estimated by the area under the survival curve up to time t = tau

$$RMST_s(\tau \mid X, \theta) = \int_0^{\tau} E[S(u \mid X, \theta)]du$$



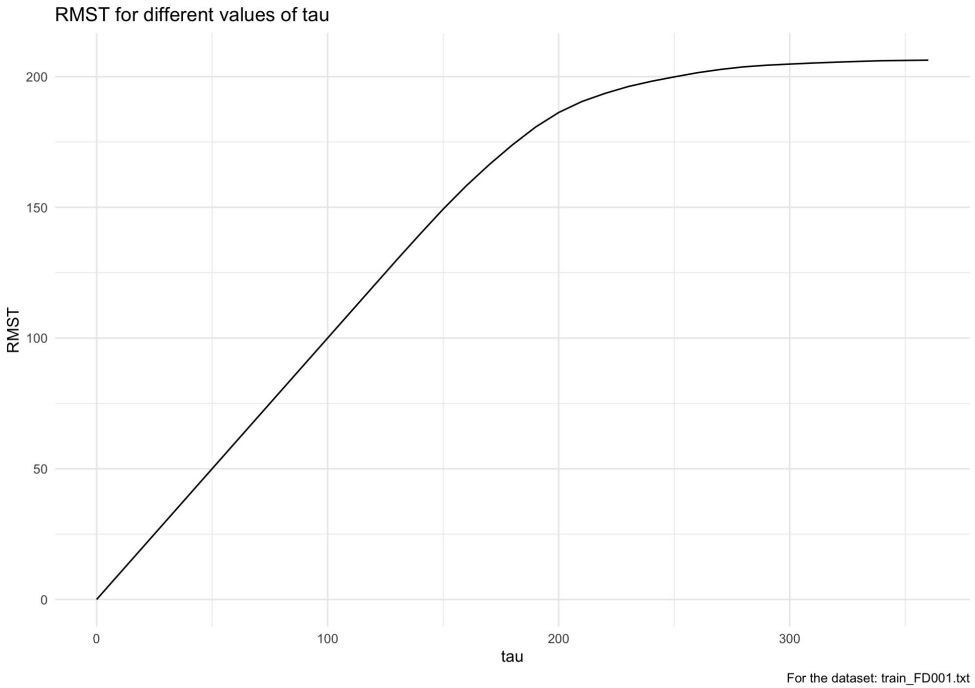
Restricted Mean Survival Time (RMST)

Expected survival time up to a cutoff time tau

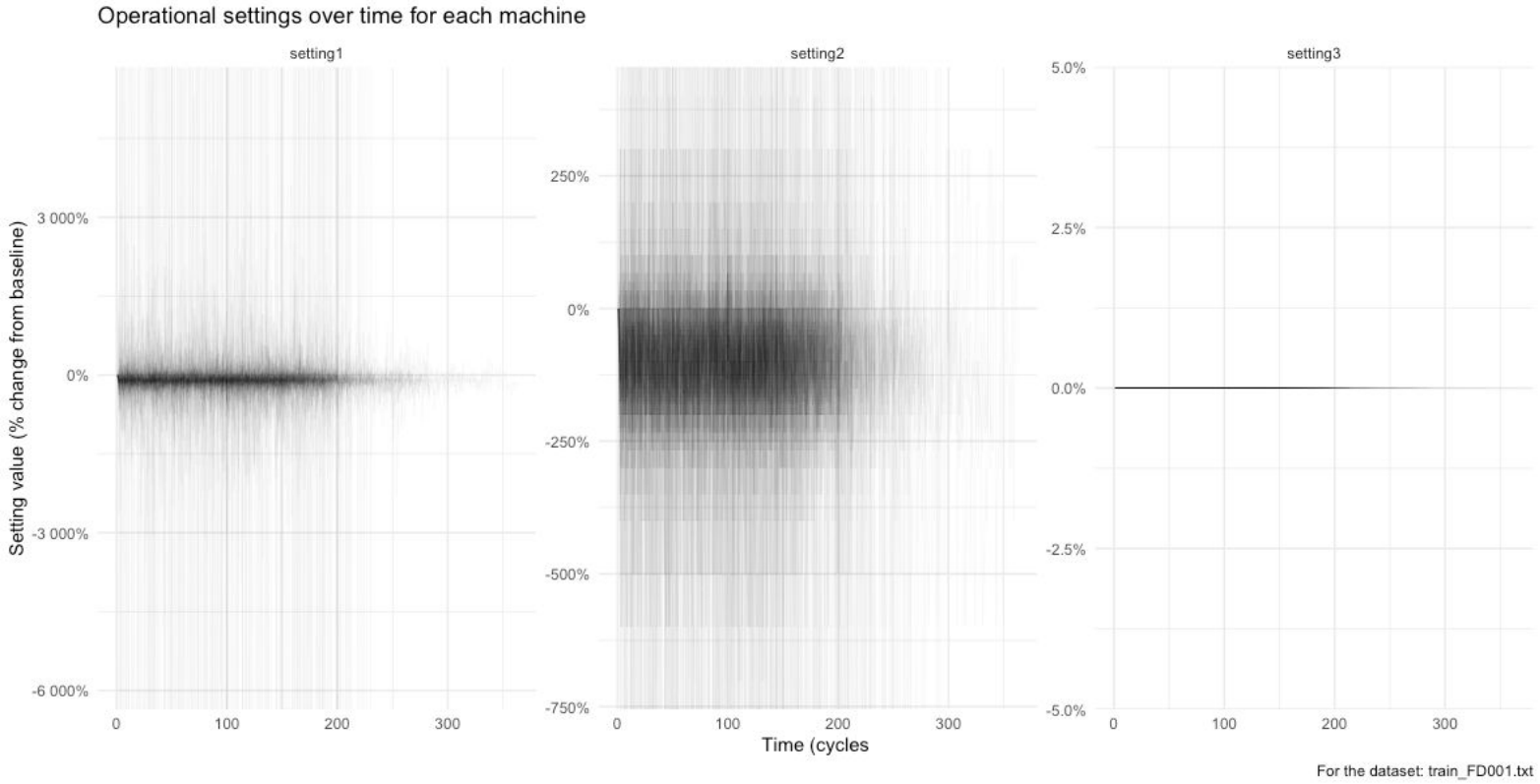
$$RMST(\tau) = E[min(T, \tau)]$$

Estimated by the area under the survival curve up to time t = tau

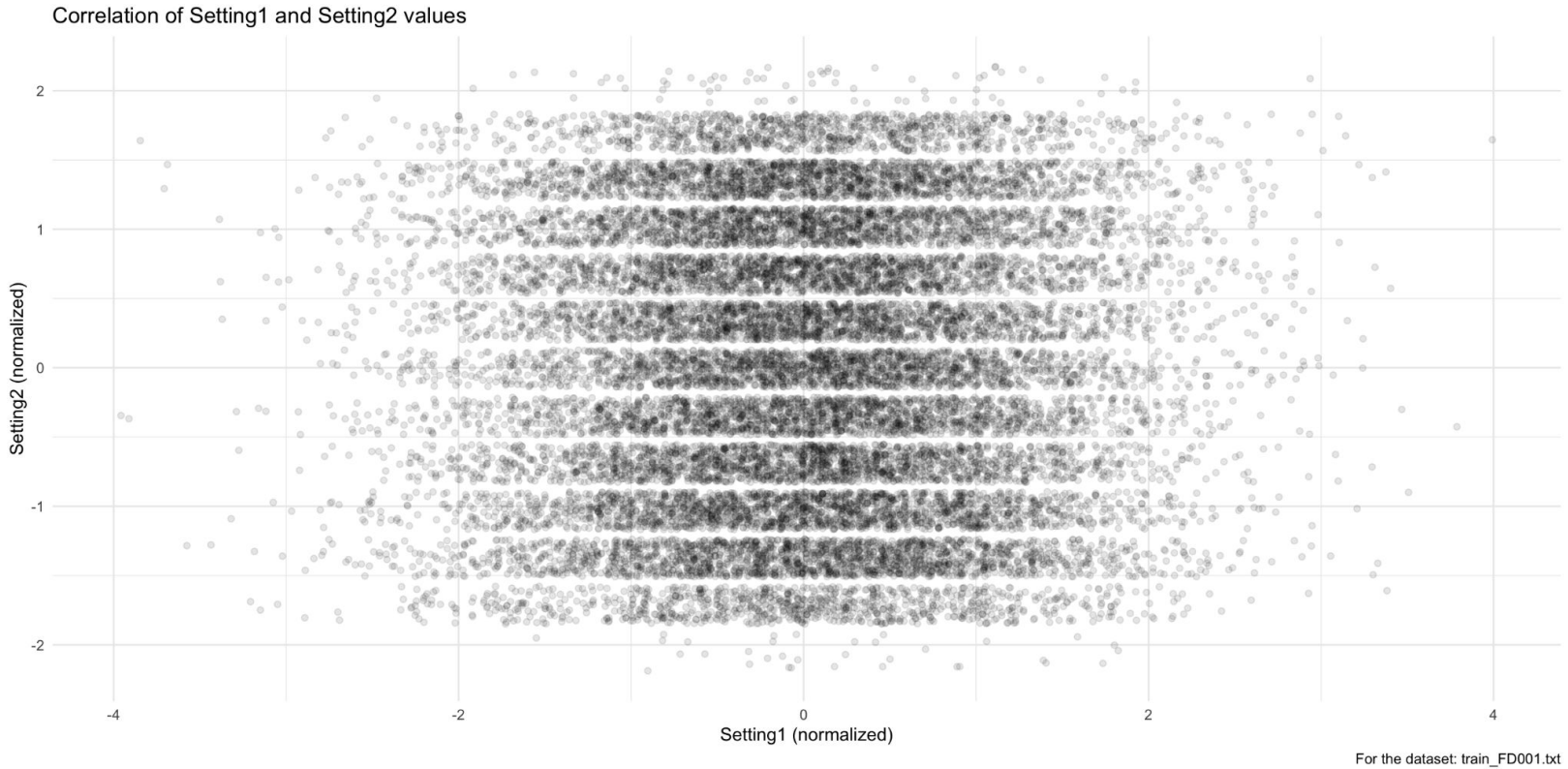
$$RMST_s(\tau \mid X, \theta) = \int_0^{\tau} E[S(u \mid X, \theta)]du$$



EDA of settings data



EDA of settings data



Pick an initial model

1. Survival-type model
2. Start with intercept-only (population model)
3. Consider adding time-varying covariates

Explore priors

Using default priors

```
library(survival)

library(rstanarm)


prior_exp_hazard <- stan_surv(
  formula = Surv(time = end, event = event) ~ 1,
  data = events,
  basehaz = "exp",
  prior_PD = TRUE
)
```

(data transformed into subject durations)

constant baseline hazard (exponential Survival)

Return prior draws

Inspect default priors

```
> prior_summary(prior_exp_hazard)

Priors for model 'prior_exp_hazard'

------

Intercept

 ~ normal(location = 0, scale = 20)

Auxiliary (NA)

 ~ flat

------

See help('prior_summary.stanreg') for more details
```

Prior on the (constant) log baseline hazard

Bayesian Machinery: Prior Predictive Distribution

- For a given prior, **p(theta)** and the data model **p(y|theta)**, we can construct the prior predictive distribution as follows:

$$p(y) = \int p(\theta)p(y|\theta)d\theta$$

- Observe that we are not conditioning on the observed data y-tilde

In practice

- Priors on parameters are specific to the model in question
- Priors on parameters often interact with one another
- Priors have a greater impact where information (data) are scarce:
  - Hyper-parameters
  - Scale / variance parameters
  - Parameters uninformed by data (example: a beta term where covariates are invariant)

Generate predicted quantities from a prior fit

```
pphaz <- posterior_survfit(
  prior_exp_hazard,
  type = 'haz',
  newdata = events,
  prob = 0.5
)
```

Return predicted hazard (haz)

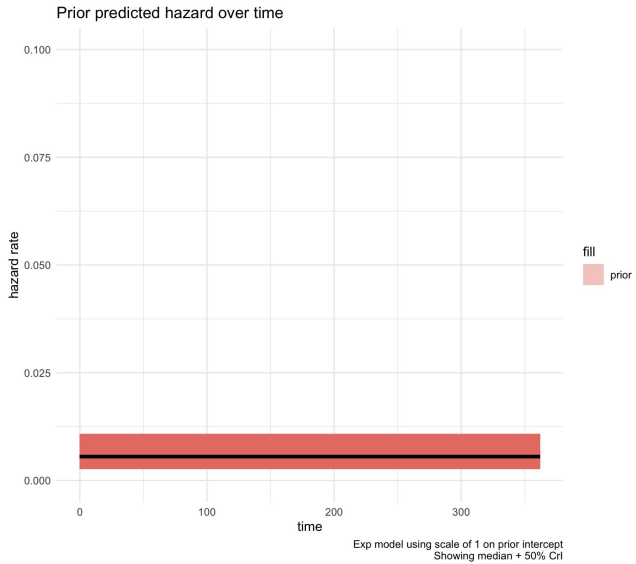
Width of uncertainty (aka credible) interval

Prior predicted hazard (default priors)

```
> head(pphaz)
stan_surv predictions
 num. individuals: 100
 prediction type:  hazard rate
 standardised?:    no
 conditional?:     no

  id cond_time    time median  ci_lb     ci_ub
1  1        NA  0.0000  9e-04 0.0000 2102.5023
2  1        NA  3.6566  9e-04 0.0000 2102.5023
3  1        NA  7.3131  9e-04 0.0000 2102.5023
4  1        NA 10.9697  9e-04 0.0000 2102.5023
5  1        NA 14.6263  9e-04 0.0000 2102.5023
6  1        NA 18.2828  9e-04 0.0000 2102.5023
```

Update the priors

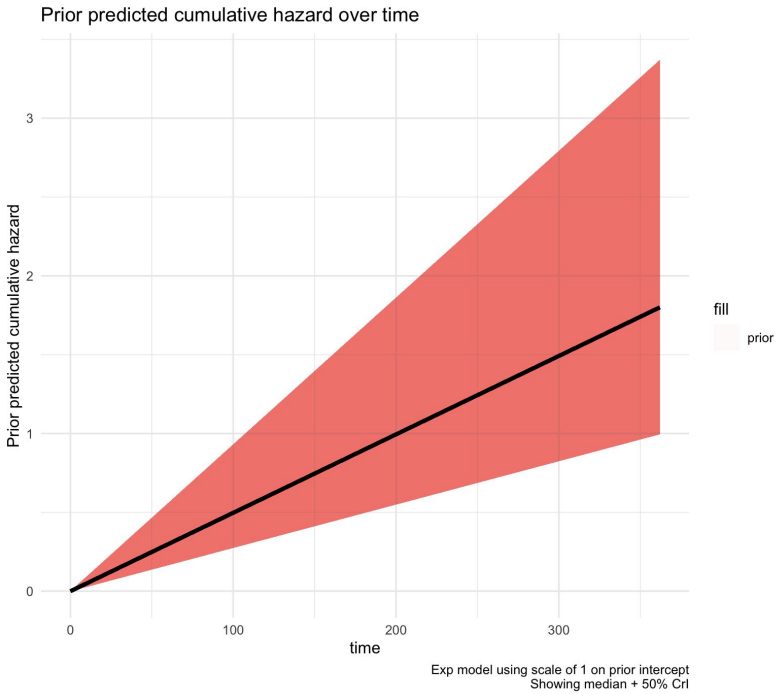
```
prior_exp_hazard2 <- update(prior_exp_hazard,
                            prior_intercept = normal(0, 1))
```




```
> head(pphaz2)
stan_surv predictions
 num. individuals: 100
 prediction type:  hazard rate
 standardised?:    no
 conditional?:     no

  id cond_time    time median  ci_lb  ci_ub
1  1        NA  0.0000 0.0056 0.0027 0.0108
2  1        NA  3.6566 0.0056 0.0027 0.0108
3  1        NA  7.3131 0.0056 0.0027 0.0108
4  1        NA 10.9697 0.0056 0.0027 0.0108
5  1        NA 14.6263 0.0056 0.0027 0.0108
6  1        NA 18.2828 0.0056 0.0027 0.0108
```

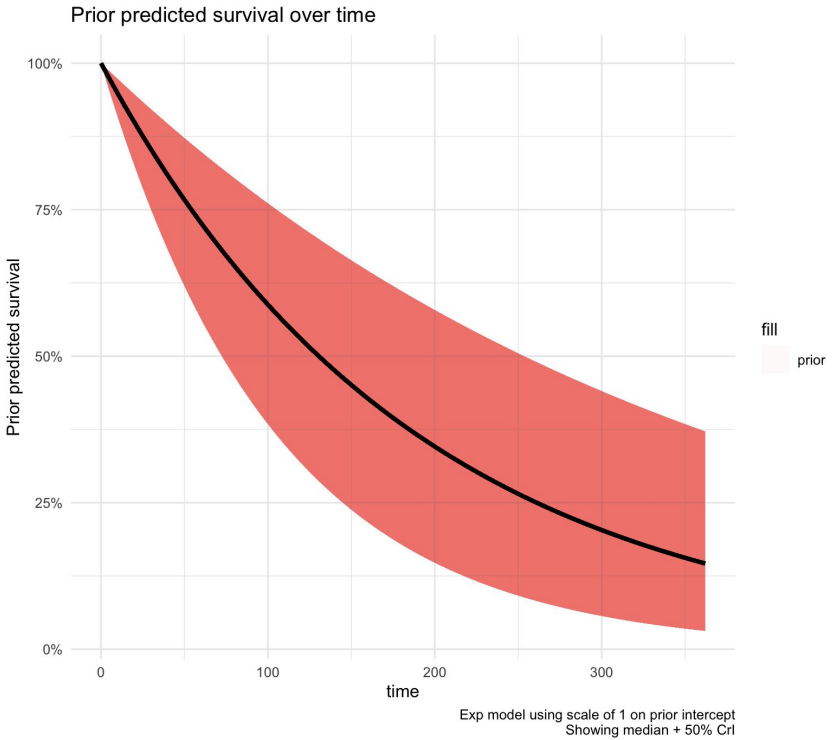
Prior predicted cumulative hazard

```
ppcumhaz2 <- posterior_survfit(
  prior_exp_hazard2,
  type = 'cumhaz',
  newdata = events,
  prob = 0.5)
```



Prior predicted survival

```
ppsurv2 <- posterior_survfit(
  prior_exp_hazard2,
  type = 'surv',
  newdata = events,
  prob = 0.5)
```



Standardized estimates

All of these functions return predictions for the provided newdata argument. That is, they predict for each subject-level (and possibly time) provided in the data.

However, we can instead return a standardized survival estimate:

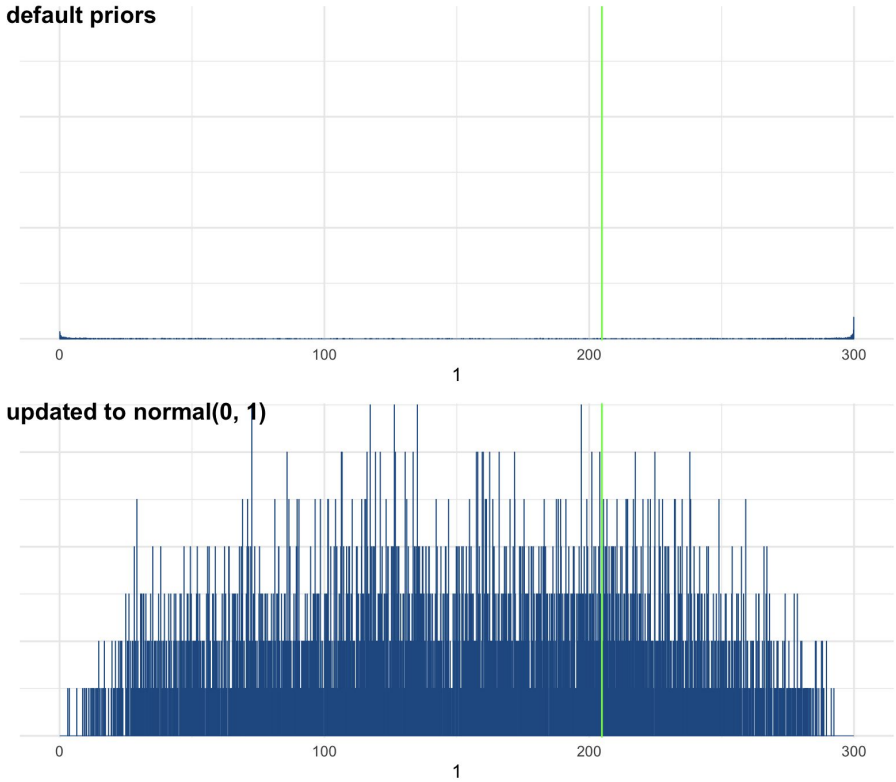
$$\widehat{S}^{P}(t) = \frac{1}{N}\sum_{i=1}^{N} S_i(t)$$

Standardized estimates

```
ppsurv2_standardized <- seq(from = 0, to = 300, by = 10) %>%
  set_names() %>%
  map_dfr(
    ~ posterior_survfit(prior_exp_hazard2, newdata = events,
                        type = 'surv', times = .x,
                        standardise = TRUE, prob = 0.5),
    .id = 'time'
  ) %>%
  mutate(time = as.integer(time))
```

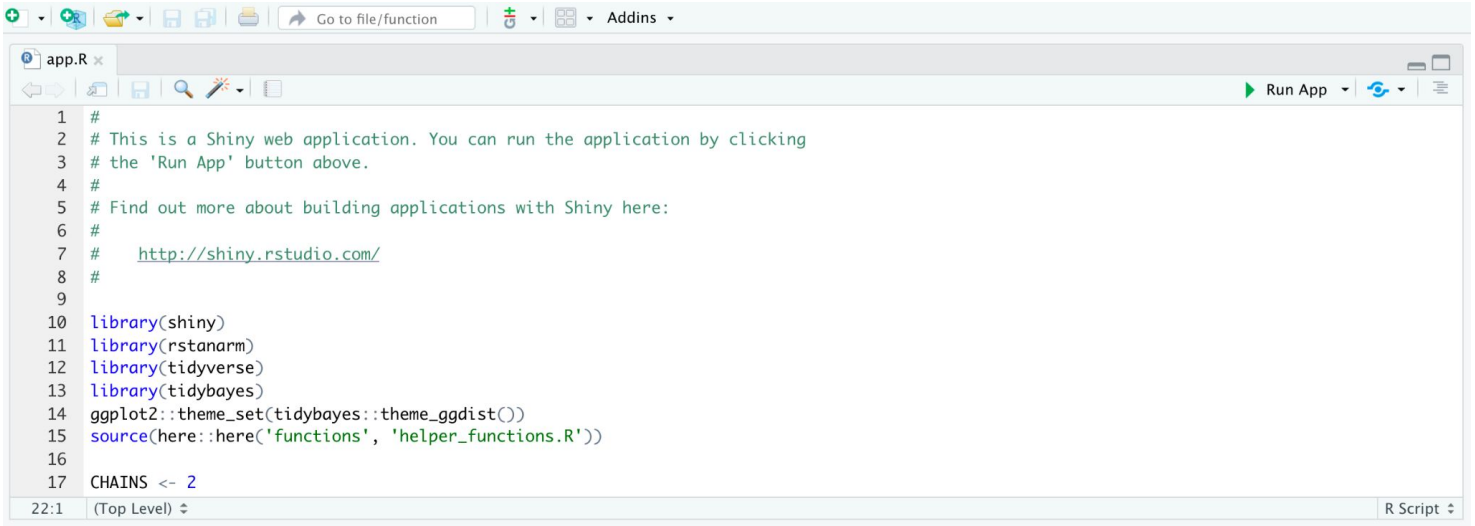
RMST

- We integrate the predicted survival curve numerically to approximate RMST, using Gauss-Kronrod quadrature.
- This is not yet integrated into rstanarm
- See the helper-functions.R code for details.



Your turn

1. Open Rstudio
2. Create a new project from github repo: https://github.com/generable/ncb2021
3. Open hazard_simulator2/app.R
4. Click Run App



What did you learn?

1. Prior scales can have a big impact on observed quantities

2. There are some unexpected patterns in prior distributions, depending on parameters and options.

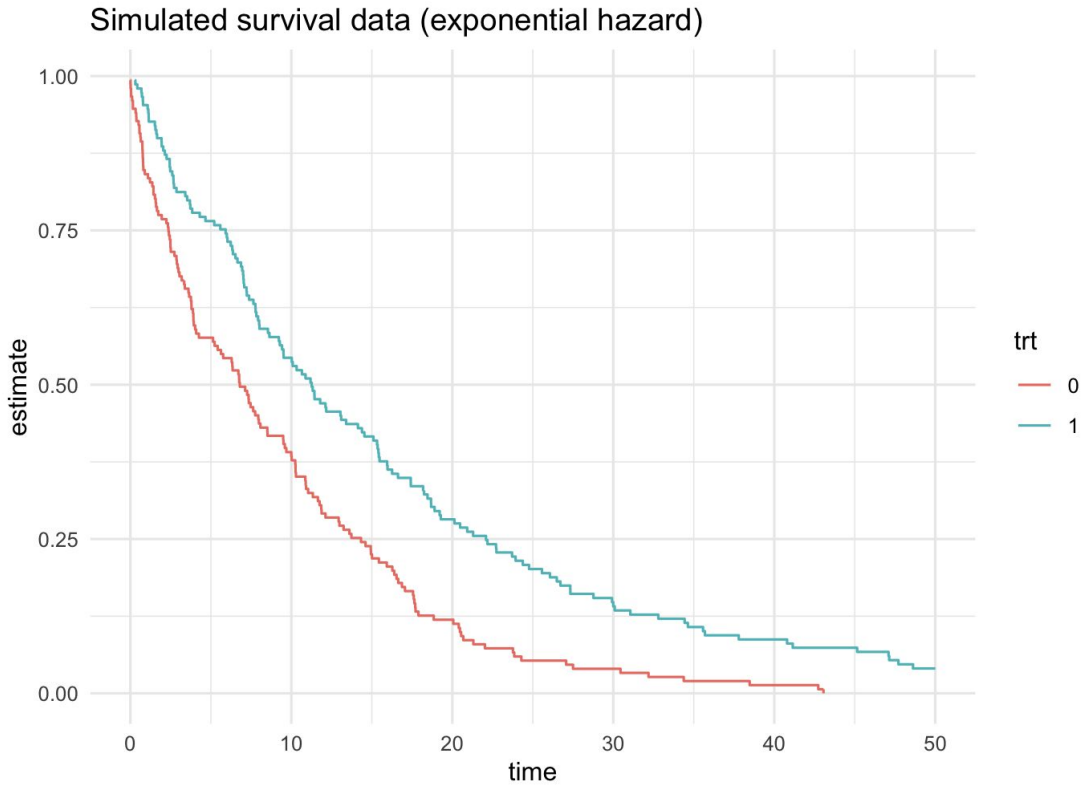
3. What priors would you use, and which models would you fit to these data?

Simulating data

Simulating data using simsurv

```
library(simsurv)

library(tidyverse)


set.seed(1234)

covs <- data.frame(id = 1:300, trt = stats::rbinom(300, 1L, 0.5))

s1 <- simsurv(lambdas = 0.1, betas = c(trt = -0.5),

              dist = 'exponential',

              x = covs, maxt = 50) %>%

  mutate(true_lambda = 0.1, true_beta = -0.5, dist = 'exponential')

sim_exp <- left_join(covs, s1, by = 'id')
```

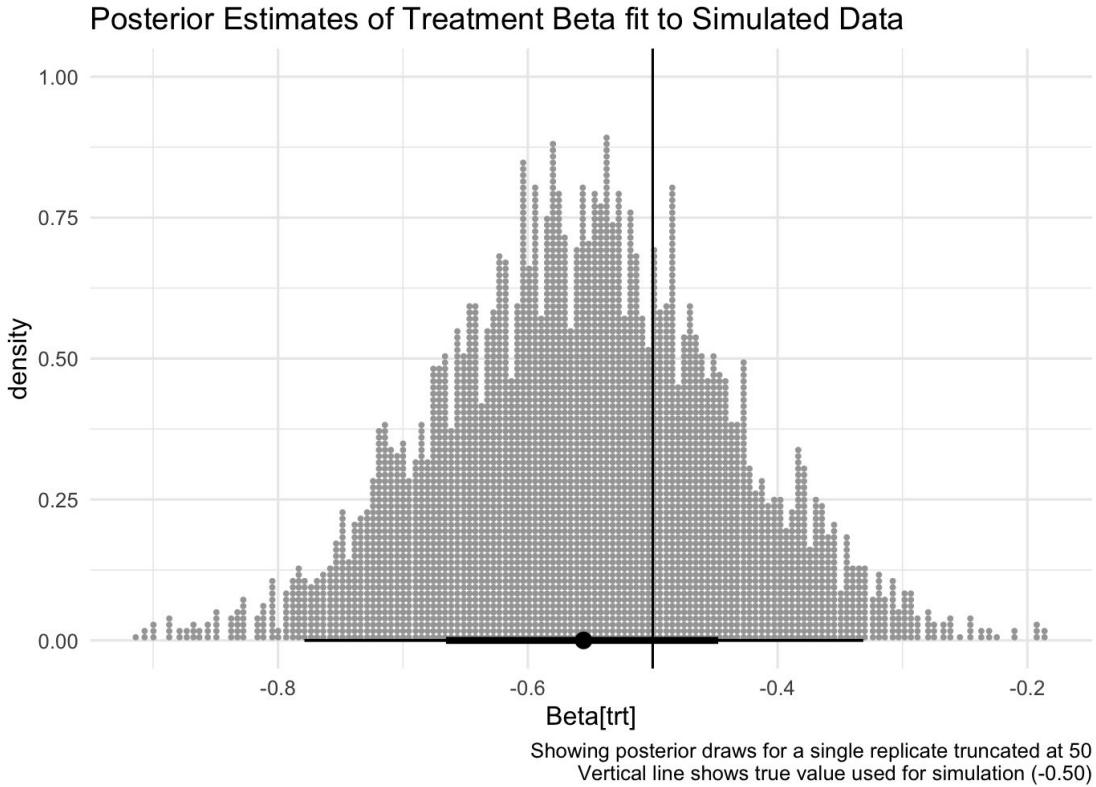
Plot simulated data



Fit model to simulated data

```
sim_fit <- stan_surv(
  formula = Surv(time = eventtime, event = status) ~ trt,
  data = sim_exp,
  basehaz = 'exp',
  prior_intercept = normal(0, 1))
```

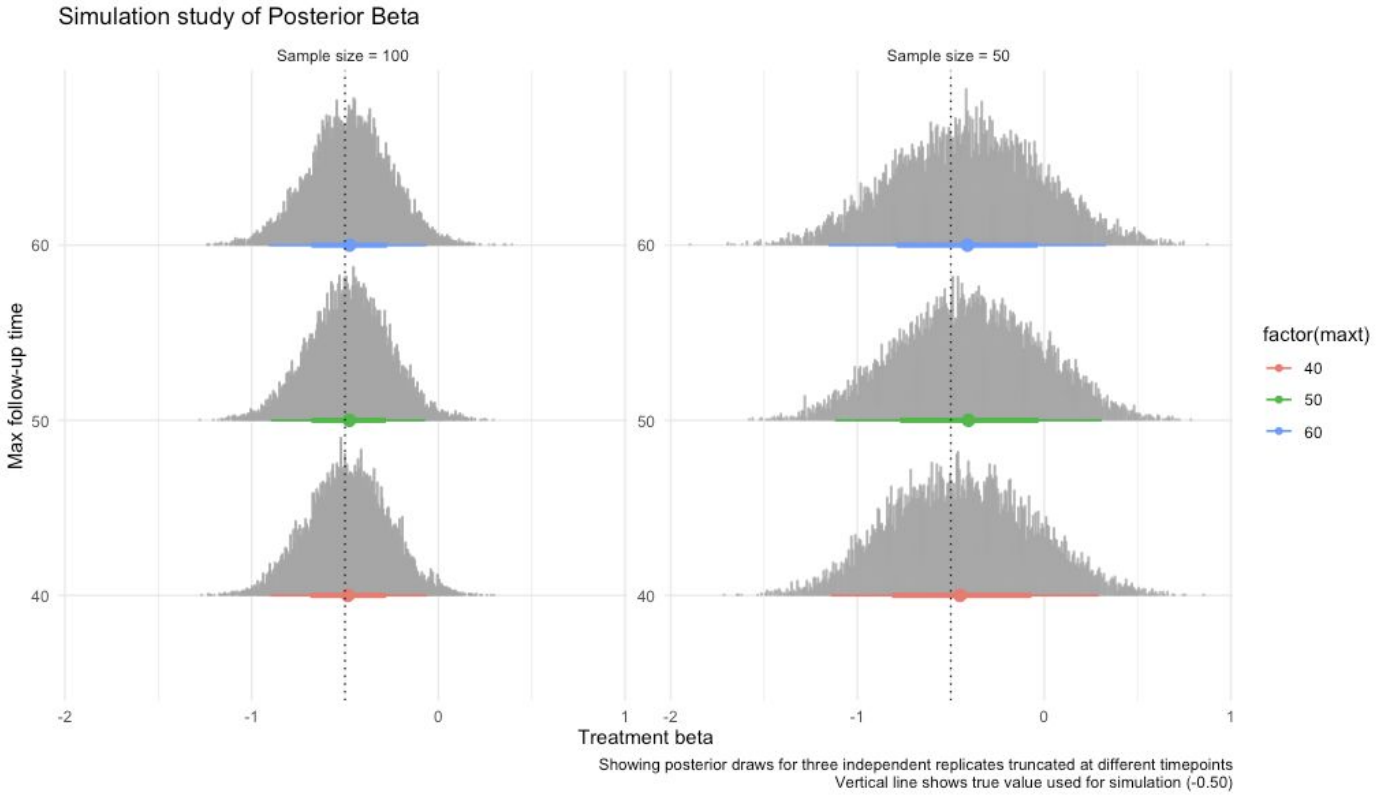
Compare posterior estimates to true values



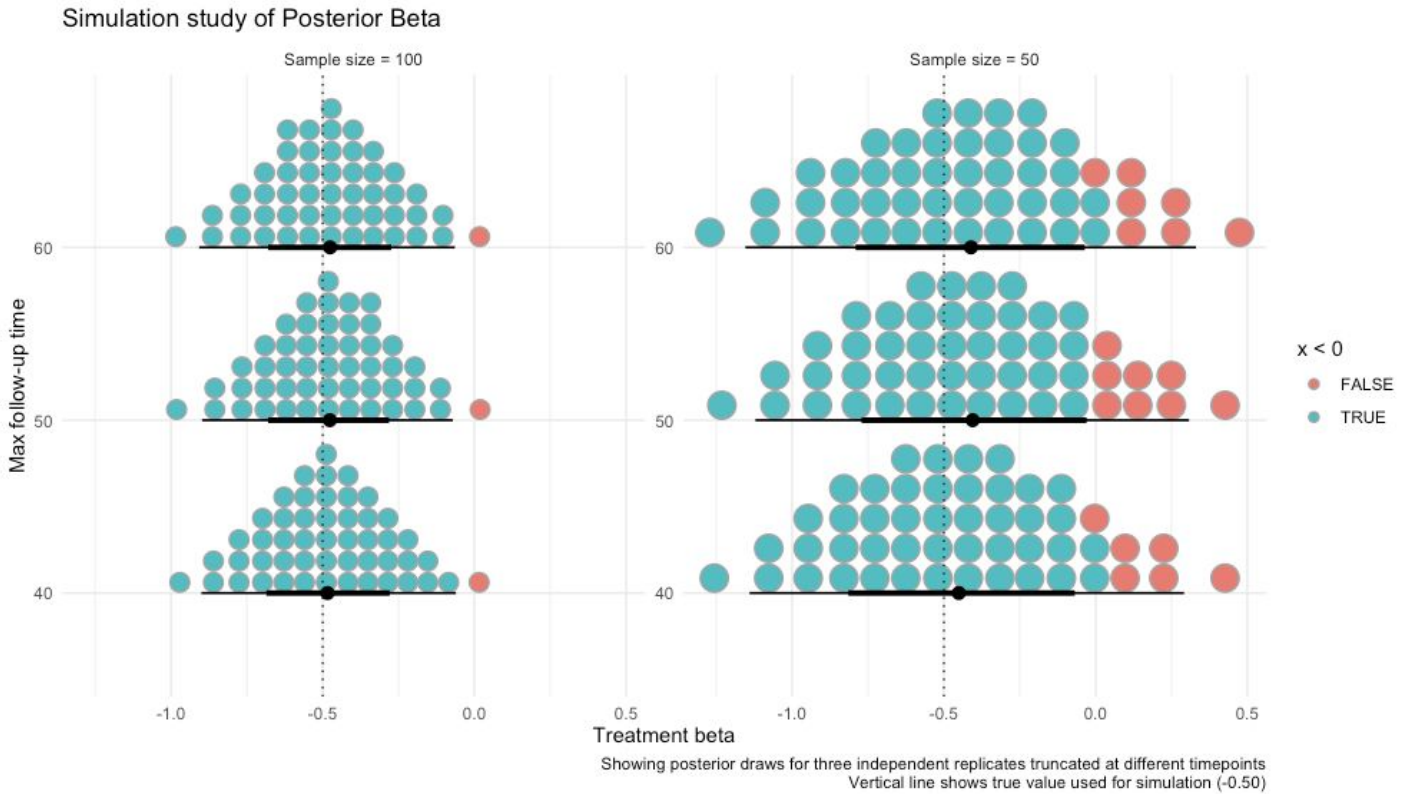
Test sensitivity to sample size & follow-up

```
scenarios <- expand_grid(
  n = seq(from = 50, to = 100, by = 50),
  maxt = seq(from = 40, to = 60, by = 10),
  seed = seq_len(3))

simulations <- purrr::pmap(scenarios, simulate_data)
```

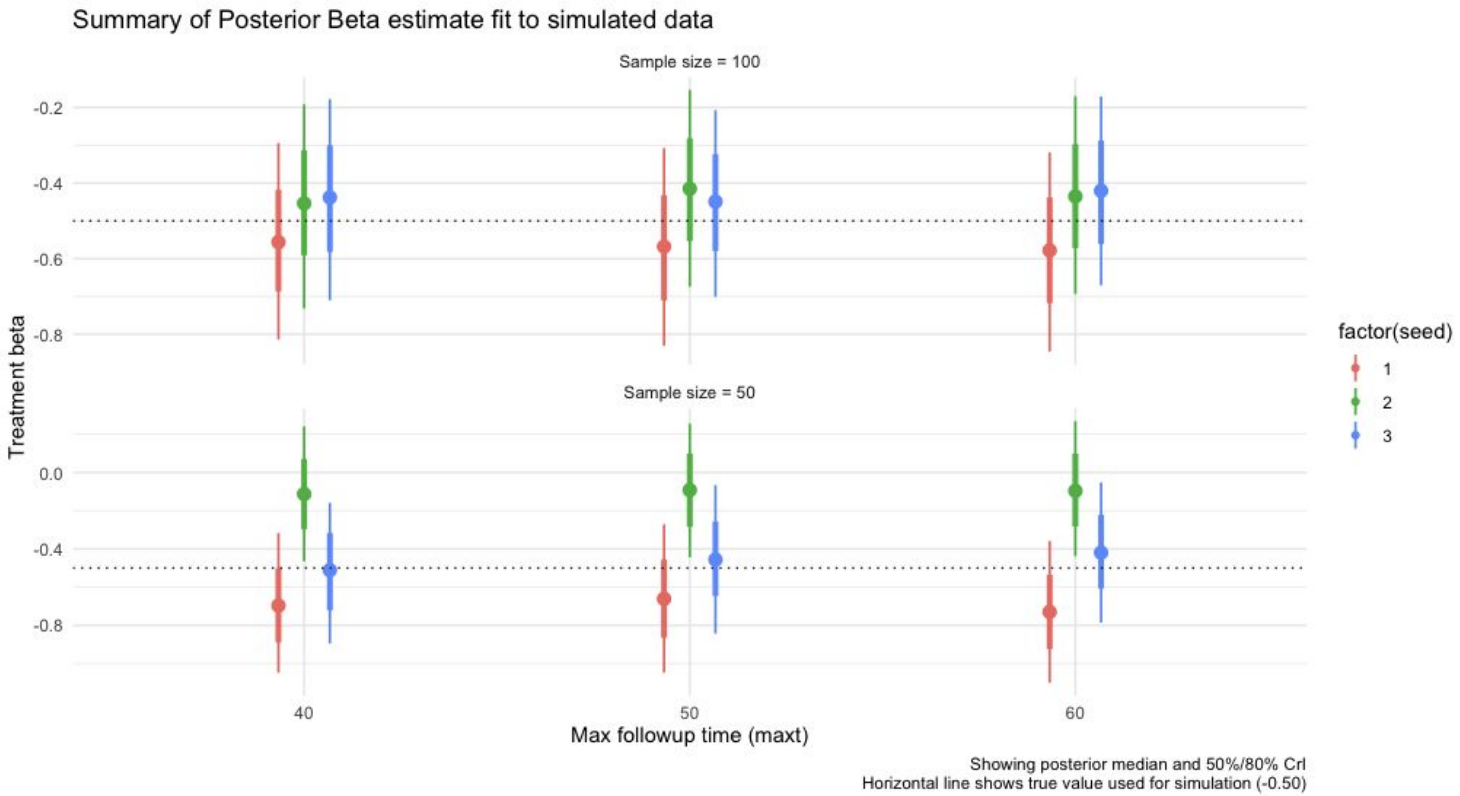
Simulation study: Sensitivity to sample size & follow-up



Simulation study: Sensitivity to sample size & follow-up



Simulation study: Sensitivity to sample size & follow-up



Model comparison

LOO (leave-one-out) comparisons

1. LOO-CV: Use cross-validation to compute leave-one-out posterior predictions

$$\sum_{n=1}^{N} \log p(y_n \mid y_1, \ldots, y_{n-1}, y_{n+1}, \ldots, y_N)$$

2. LOO-PSIS: Approximates the LOO-CV by using the log-likelihood for each observation to approximate the likelihood if that observation were removed

Running LOO

```
> sim_loo <- loo(sim_fit)
```

```
> sim_loo

Computed from 4000 by 300 log-likelihood matrix

         Estimate   SE
elpd_loo  -1027.9 15.3
p_loo         1.8  0.2
looic      2055.8 30.5
------
Monte Carlo SE of elpd_loo is 0.0.

All Pareto k estimates are good (k < 0.5).
See help('pareto-k-diagnostic') for details.
```

LOO for model comparison

1. Fit an alternative model, just for demonstration purposes
2. Compare models using ELPD estimates from LOO-PSIS

```
> loo_compare(sim_loo, sim_loo_alt)
             elpd_diff se_diff
sim_fit       0.0       0.0
sim_fit_alt -1.6        2.1
```

Best performing model listed first

In this case, the difference is small relative to the `se_diff`

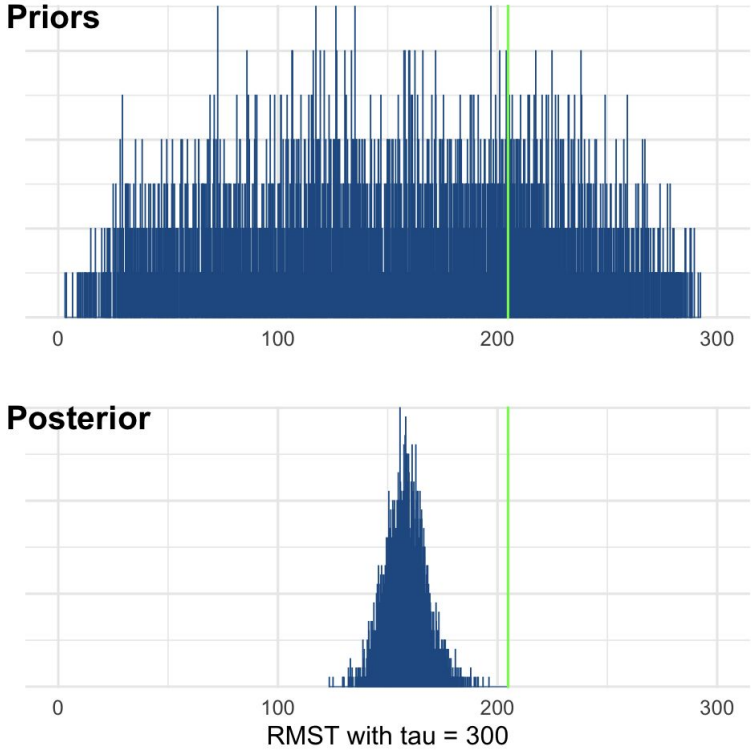
Further resources on LOO and survival models

- Efficient Leave-One-Out Brier Score for time-dependent evaluation of Bayesian Survival Models (Eren M. Elçi)
-
-

Fit the model to data

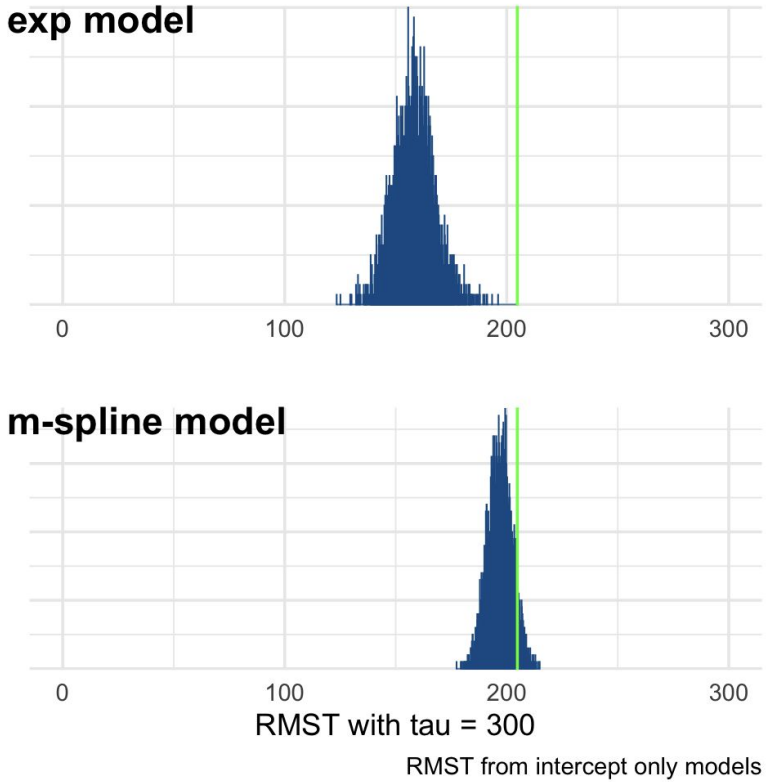
Fitting the model to our turbofan dataset

```r
# fit posterior model
post_exp_hazard2 <- stan_surv(
  formula = Surv(time = end, event = event) ~ 1,
  data = events,
  basehaz = "exp",
  prior_PD = FALSE,
  chains = CHAINS,
  cores = CORES,
  iter = ITER,
  seed = SEED,
  prior_intercept = normal(0, 1),
  prior = normal(0, 0.5))
```



RMST from intercept only model with exp baseline hazard (model 2)

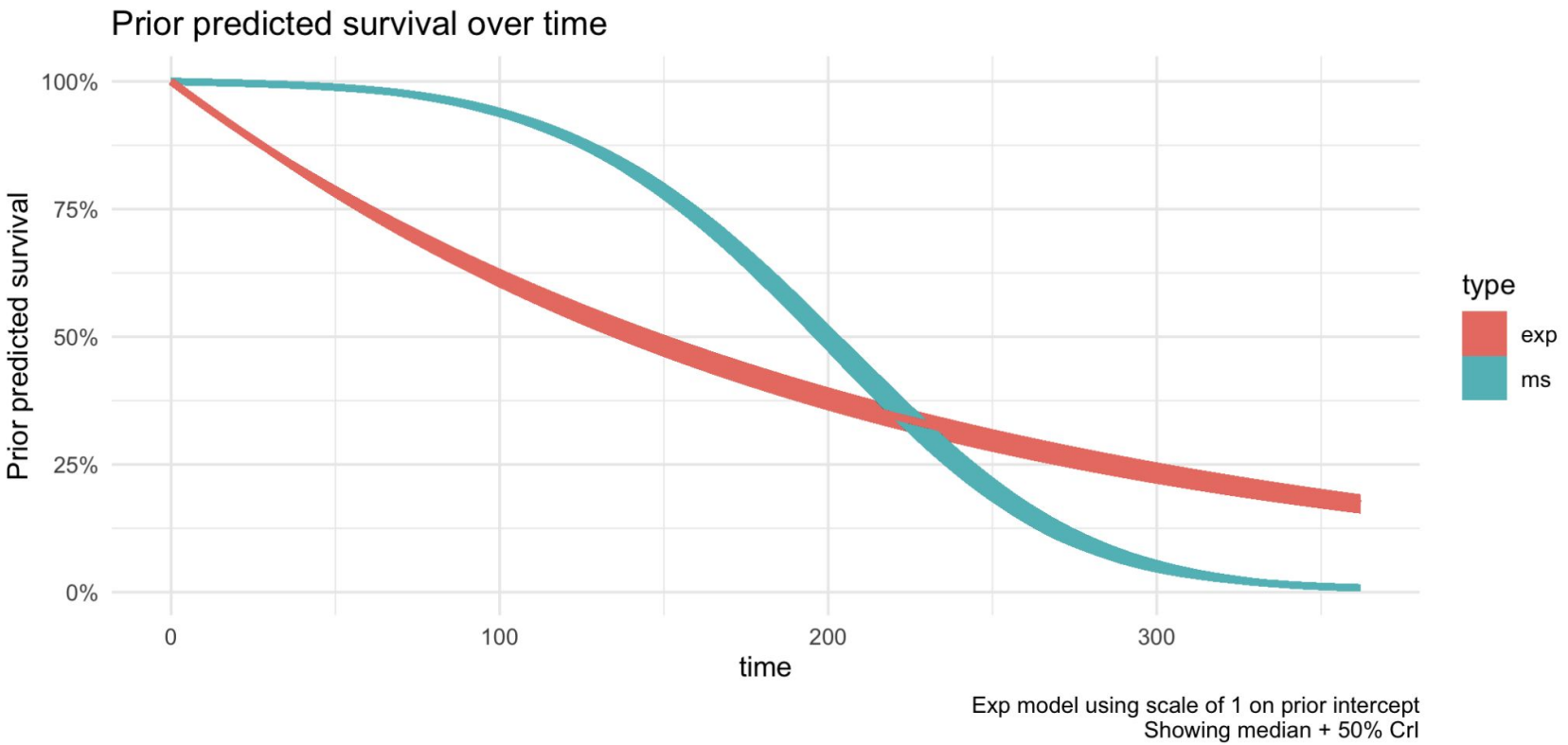
Comparing exp model to m-Spline fit



Comparison of ELPD according to LOO

```
> loo_compare(loo(post_exp_hazard2), loo(post_ms_hazard2))
                  elpd_diff se_diff
post_ms_hazard2     0.0       0.0
post_exp_hazard2 -98.3        4.9
```

Compare predicted survival curves



Joint Models

Motivations

1. Interested in how longitudinal marker influences survival
    a. Adjusted for measurement error, missing observations
    b. When longitudinal marker is mediating a treatment response on survival

2. Interested in evolution of clinical biomarker adjusted for informative dropout
    a. Example: a covid trial where patients are discharged when sufficiently healthy
    b. Or, a cancer trial where treatments are changed following tumor regrowth

In general

1. Define a (hierarchical, longitudinal) model to the clinical biomarker data
2. Define a hazard model describing survival outcomes
3. Include parameters from the longitudinal model as predictors in the survival model

Longitudinal submodel(s)

Observations $y_{ijm}(t) = y_{im}(t_{ij})$ follow a distribution in the exponential family

with expected value $\mu_{ijm}(t)$.

$$\eta_{ijm}(t) = g_m(\mu_{ijm}(t)) = \boldsymbol{x}_{ijm}^T(t)\boldsymbol{\beta}_m + \boldsymbol{z}_{ijm}^T(t)\boldsymbol{b}_{im}$$

$$\begin{pmatrix} \boldsymbol{b}_{i1} \\ \vdots \\ \boldsymbol{b}_{iM} \end{pmatrix} = \boldsymbol{b}_i \sim \text{Normal}\,(0, \boldsymbol{\Sigma})$$

Event submodel

$$h_i(t) = h_0(t; \boldsymbol{\omega})\exp\left(\boldsymbol{w}_i^T(t)\boldsymbol{\gamma} + \sum_{m=1}^{M} \alpha_m \mu_{im}(t)\right)$$

Longitudinal submodel(s)

Observations $y_{ijm}(t) = y_{im}(t_{ij})$ follow a distribution in the exponential family

with expected value $\mu_{ijm}(t)$.

$$\eta_{ijm}(t) = g_m(\mu_{ijm}(t)) = \boldsymbol{x}_{ijm}^T(t)\boldsymbol{\beta}_m + \boldsymbol{z}_{ijm}^T(t)\boldsymbol{b}_{im}$$

Population-level effects

Subject-level effects

$$\begin{pmatrix} \boldsymbol{b}_{i1} \\ \vdots \\ \boldsymbol{b}_{iM} \end{pmatrix} = \boldsymbol{b}_i \sim \text{Normal}\,(0, \boldsymbol{\Sigma})$$

Event submodel

$$h_i(t) = h_0(t; \boldsymbol{\omega})\exp\left(\boldsymbol{w}_i^T(t)\boldsymbol{\gamma} + \sum_{m=1}^{M} \alpha_m \mu_{im}(t)\right)$$

Longitudinal submodel(s)

Observations $y_{ijm}(t) = y_{im}(t_{ij})$ follow a distribution in the exponential family

with expected value $\mu_{ijm}(t)$.

$$\eta_{ijm}(t) = g_m(\mu_{ijm}(t)) = \boldsymbol{x}_{ijm}^T(t)\boldsymbol{\beta}_m + \boldsymbol{z}_{ijm}^T(t)\boldsymbol{b}_{im}$$

"Current value" included in hazard model

$$\begin{pmatrix} \boldsymbol{b}_{i1} \\ \vdots \\ \boldsymbol{b}_{iM} \end{pmatrix} = \boldsymbol{b}_i \sim \text{Normal}\,(0, \boldsymbol{\Sigma})$$

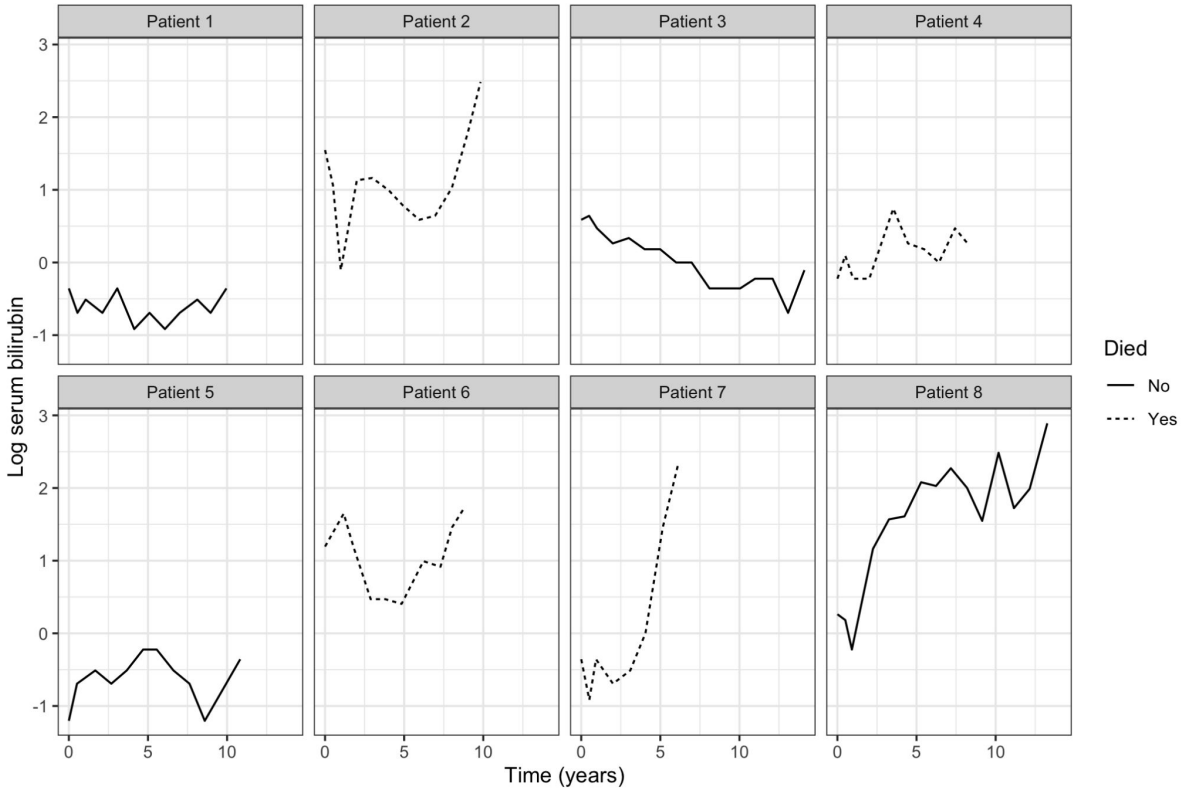
Event submodel

$$h_i(t) = h_0(t; \boldsymbol{\omega})\exp\left(\boldsymbol{w}_i^T(t)\boldsymbol{\gamma} + \sum_{m=1}^{M} \alpha_m \mu_{im}(t)\right)$$

There is a lot of flexibility in the association structure

$$h_i(t) = h_0(t;\omega)\exp\left(w_i^T(t)\,\gamma + \sum_{m=1}^{M}\sum_{q=1}^{Q_m} f_{mq}(\beta, b_i, \alpha_{mq}; t)\right)$$

Longitudinal marker: log serum bilirubin

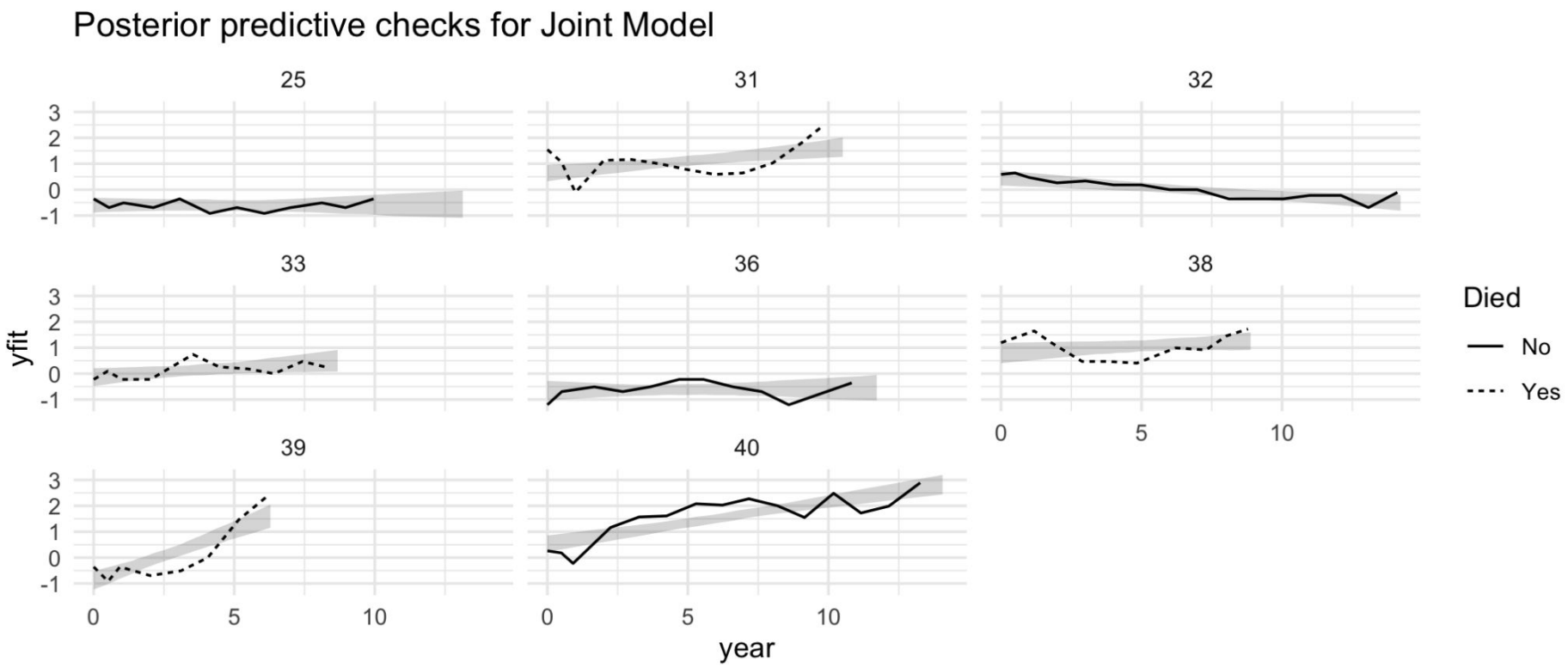


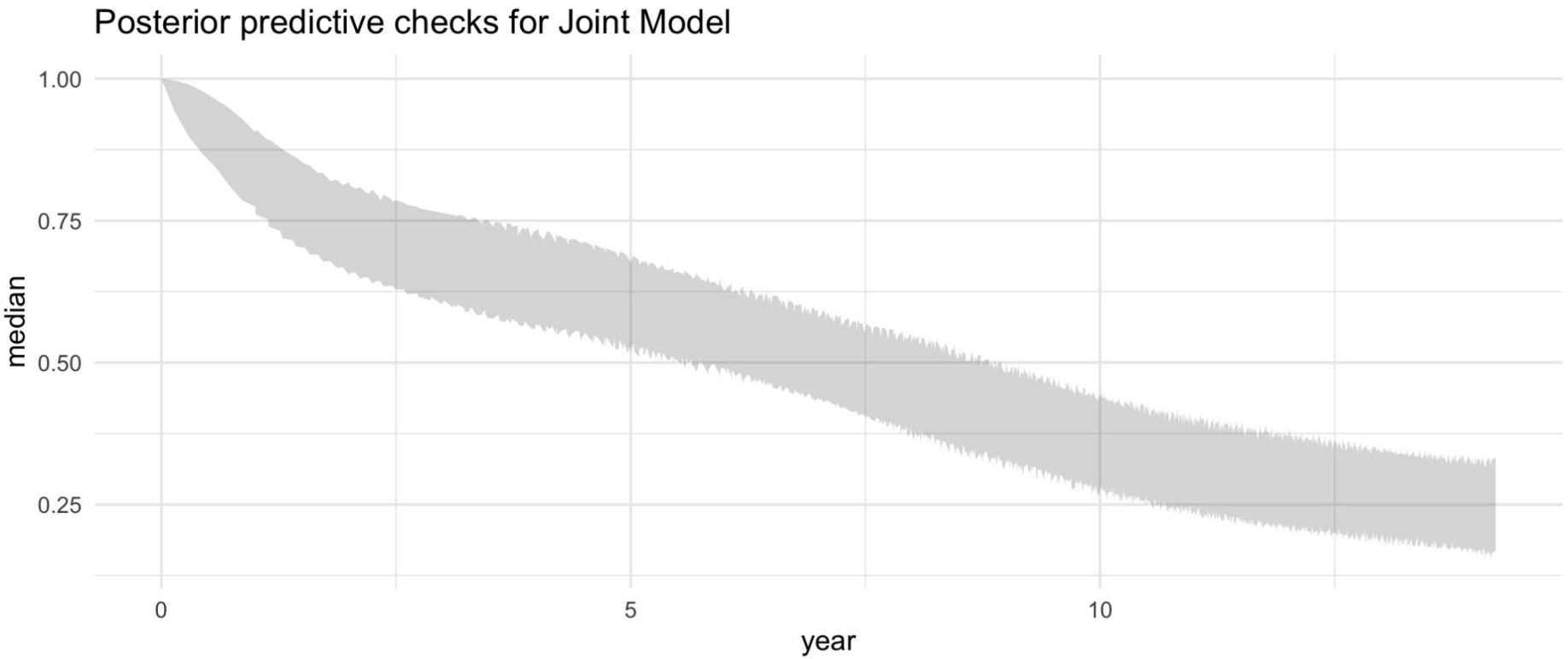
Fitting the model

```
mod1 <- stan_jm(
  formulaLong = logBili ~ sex + trt + year + (year | id),
  dataLong = pbcLong,
  formulaEvent = survival::Surv(futimeYears, death) ~ sex + trt,
  dataEvent = pbcSurv,
  time_var = "year",
  chains = 1, refresh = 2000, seed = 12345)
```

Fitting individual components

```
long1 <- stan_glmer(
  formula = logBili ~ sex + trt + year + (year | id),
  data = pbcLong,
  chains = 1, refresh = 2000, seed = 12345)

surv1 <- stan_surv(
  formula = survival::Surv(futimeYears, death) ~ sex + trt,
  data = pbcSurv,
  chains = 1, refresh = 2000, seed = 12345)
```


Posterior predictive checks for Joint Model
25
31
32
33
36
38
39
40
yfit
year
Died
No
Yes


Posterior predictive checks for Joint Model
1.00
0.75
0.50
0.25
median
0
5
10
year