rstanarm feature/survival branch

Kazuki Yoshida

2018-11-12

References

- rstanarm source files
  - rstanarm/R/stan_surv.R (https://github.com/stan-dev/rstanarm/blob/feature/survival/R/stan_surv.R)
  - rstanarm/R/posterior_survfit.R (https://github.com/stan-dev/rstanarm/blob/feature/survival/R/posterior_survfit.R)

Load packages

```
library(tidyverse)
## Install
## devtools::install_github("stan-dev/rstanarm", ref = "feature/survival", build_vignettes = FALSE)
library(rstanarm)
library(tidybayes)
library(bayesplot)
library(survival)

set.seed(167268372)
```

Load dataset

```
aml                  package:survival                  R Documentation
Acute Myelogenous Leukemia survival data
Description:
     Survival in patients with Acute Myelogenous Leukemia.  The
     question at the time was whether the standard course of
     chemotherapy should be extended ('maintainance') for additional
     cycles.
Usage:
     aml
     leukemia
Format:
       time:    survival or censoring time
       status:  censoring status
       x:       maintenance chemotherapy given? (factor)
Source:
     Rupert G. Miller (1997), _Survival Analysis_.  John Wiley & Sons.
     ISBN: 0-471-25218-2.
```

```
data(leukemia, package = "survival")
leukemia <- as_data_frame(leukemia) %>%
    mutate(id = seq_len(n())) %>%
    select(id, everything())
leukemia
```

```
## # A tibble: 23 x 4
##       id  time status x
##    <int> <dbl>  <dbl> <fct>
##  1     1     9      1 Maintained
##  2     2    13      1 Maintained
##  3     3    13      0 Maintained
##  4     4    18      1 Maintained
##  5     5    23      1 Maintained
##  6     6    28      0 Maintained
##  7     7    31      1 Maintained
##  8     8    34      1 Maintained
##  9     9    45      0 Maintained
## 10    10    48      1 Maintained
## # ... with 13 more rows
```

stan_surv

Explanation

The CRAN version of rstanarm currently lacks capabilities for survival models. However, there is currently a feature branch on survival analyses, which we will demonstrate here.

What is most special about survival time modeling is the existence of the baseline hazard function. Frequentist proportional hazards regression (Cox regression) omits estimation of the baseline hazard function using a partial likelihood, from which the baseline hazard function drops out.

In the Bayesian paradigm, we need a full likelihood, resulting in a need to model it one way or another. This survival feature branch of rstanarm supports the following options for the baseline hazard function modeling.

- cubic M-spline
- cubic B-spline
- exponential
- weibull
- gomperz

```
#' @param basehaz A character string indicating which baseline hazard to use
#'   for the event submodel. Current options are:
#'   \itemize{
#'     \item \code{"ms"}: a flexible parametric model using cubic M-splines to
#'     model the baseline hazard. The default locations for the internal knots,
#'     as well as the basis terms for the splines, are calculated with respect
#'     to time. If the model does \emph{not} include any time-dependendent
#'     effects then a closed form solution is available for both the hazard
#'     and cumulative hazard and so this approach should be relatively fast.
#'     On the other hand, if the model does include time-dependent effects then
#'     quadrature is used to evaluate the cumulative hazard at each MCMC
#'     iteration and, therefore, estimation of the model will be slower.
#'     \item \code{"bs"}: a flexible parametric model using cubic B-splines to
#'     model the \emph{log} baseline hazard. The default locations for the
#'     internal knots, as well as the basis terms for the splines, are calculated
#'     with respect to time. A closed form solution for the cumulative hazard
#'     is \strong{not} available regardless of whether or not the model includes
#'     time-dependent effects; instead, quadrature is used to evaluate
#'     the cumulative hazard at each MCMC iteration. Therefore, if your model
#'     does not include any time-dependent effects, then estimation using the
#'     \code{"ms"} baseline hazard will be faster.
#'     \item \code{"exp"}: an exponential distribution for the event times.
#'     (i.e. a constant baseline hazard)
#'     \item \code{"weibull"}: a Weibull distribution for the event times.
#'     \item \code{"gompertz"}: a Gompertz distribution for the event times.
#'   }
```

Exponential model

```
stan_surv_exponential <- stan_surv(formula = Surv(time, status) ~ x,
                                   data = leukemia,
                                   basehaz = "exp")
prior_summary(stan_surv_exponential)
```

```
## Priors for model 'stan_surv_exponential'
## ------
## Intercept
##  ~ normal(location = 0, scale = 20)
##
## Coefficients
##  ~ normal(location = 0, scale = 2.5)
##
## Auxiliary (NA)
##  ~ normal(location = , scale = )
## ------
## See help('prior_summary.stanreg') for more details
```

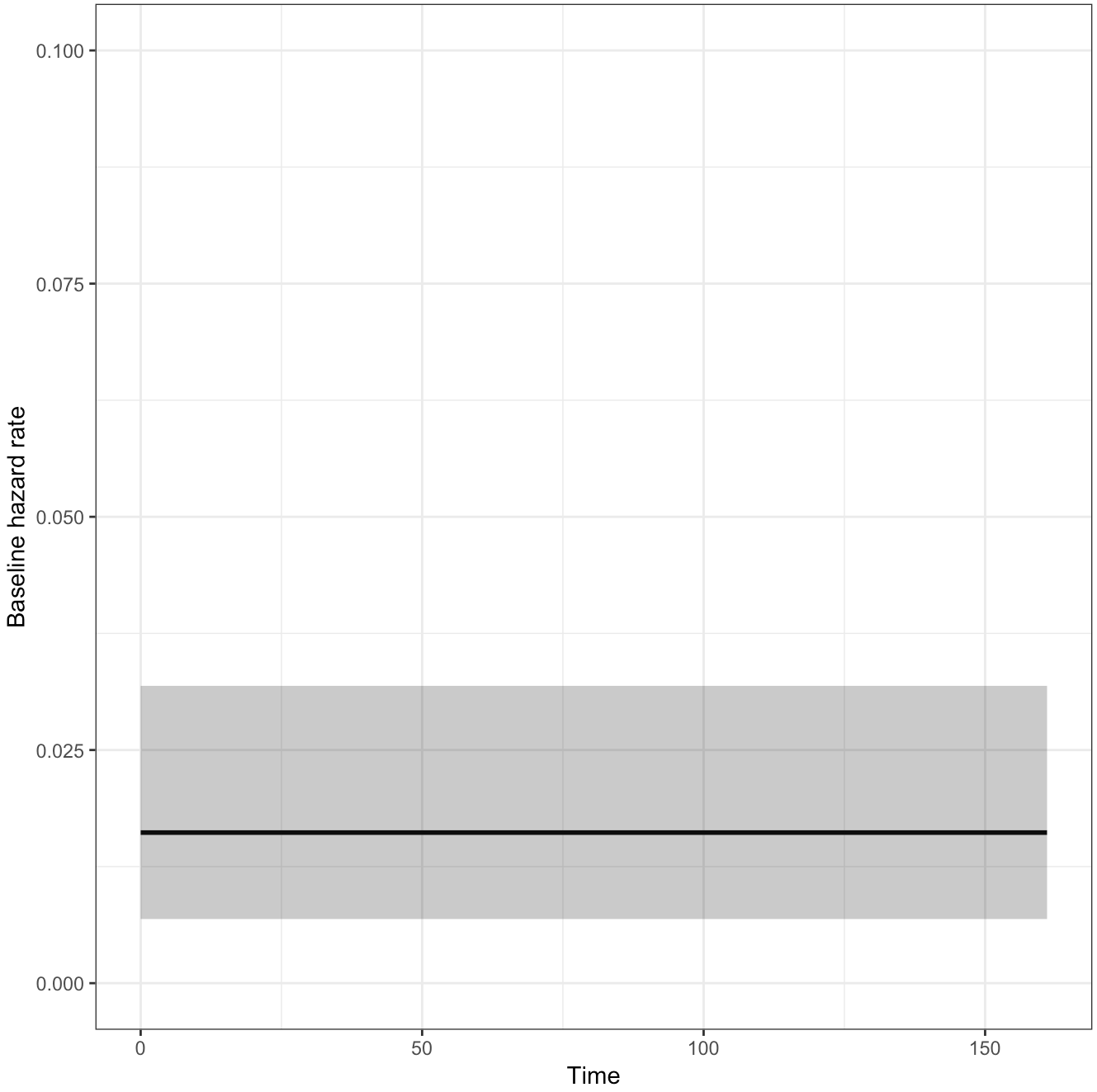
```
summary(stan_surv_exponential)
```

```
##
## Model Info:
##
##  function:        stan_surv
##  baseline hazard: exponential
##  formula:         Surv(time, status) ~ x
##  algorithm:       sampling
##  priors:          see help('prior_summary')
##  sample:          4000 (posterior sample size)
##  observations:    23
##  events:          18 (78.3%)
##  right censored:  5 (21.7%)
##  delayed entry:   no
##
## Estimates:
##                  mean   sd    2.5%   25%   50%   75%   97.5%
## (Intercept)     -4.1    0.4  -5.0   -4.4  -4.1  -3.9  -3.4
## xNonmaintained   0.9    0.5   0.0    0.6   0.9   1.3   1.9
## log-posterior  -87.3    1.0 -90.0  -87.7 -87.0 -86.5 -86.2
##
## Diagnostics:
##                mcse Rhat n_eff
## (Intercept)    0.0  1.0   856
## xNonmaintained 0.0  1.0   830
## log-posterior  0.0  1.0  1202
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample size, an
d Rhat is the potential scale reduction factor on split chains (at convergence Rhat=1).
```

It looks like the log baseline hazard appears as the “(Intercept)” since the results are comparable to my previous attempt using the Poisson trick: Piecewise constant hazard Cox (http://rpubs.com/kaz_yos/surv_stan_piecewise1).

The plot method is designed to give the baseline hazard function

```
plot(stan_surv_exponential) + coord_cartesian(ylim = c(0, 0.1))
```


0.100
0.075
0.050
0.025
0.000
Baseline hazard rate
0
50
100
150
Time

It is a constant in an exponential model.

Weibull model

```
stan_surv_weibull <- stan_surv(formula = Surv(time, status) ~ x,
                               data = leukemia,
                               basehaz = "weibull")
prior_summary(stan_surv_weibull)
```

```
## Priors for model 'stan_surv_weibull'
## ------
## Intercept
##  ~ normal(location = 0, scale = 20)
##
## Coefficients
##  ~ normal(location = 0, scale = 2.5)
##
## Auxiliary (weibull-shape)
##  ~ half-normal(location = 0, scale = 2)
## ------
## See help('prior_summary.stanreg') for more details
```

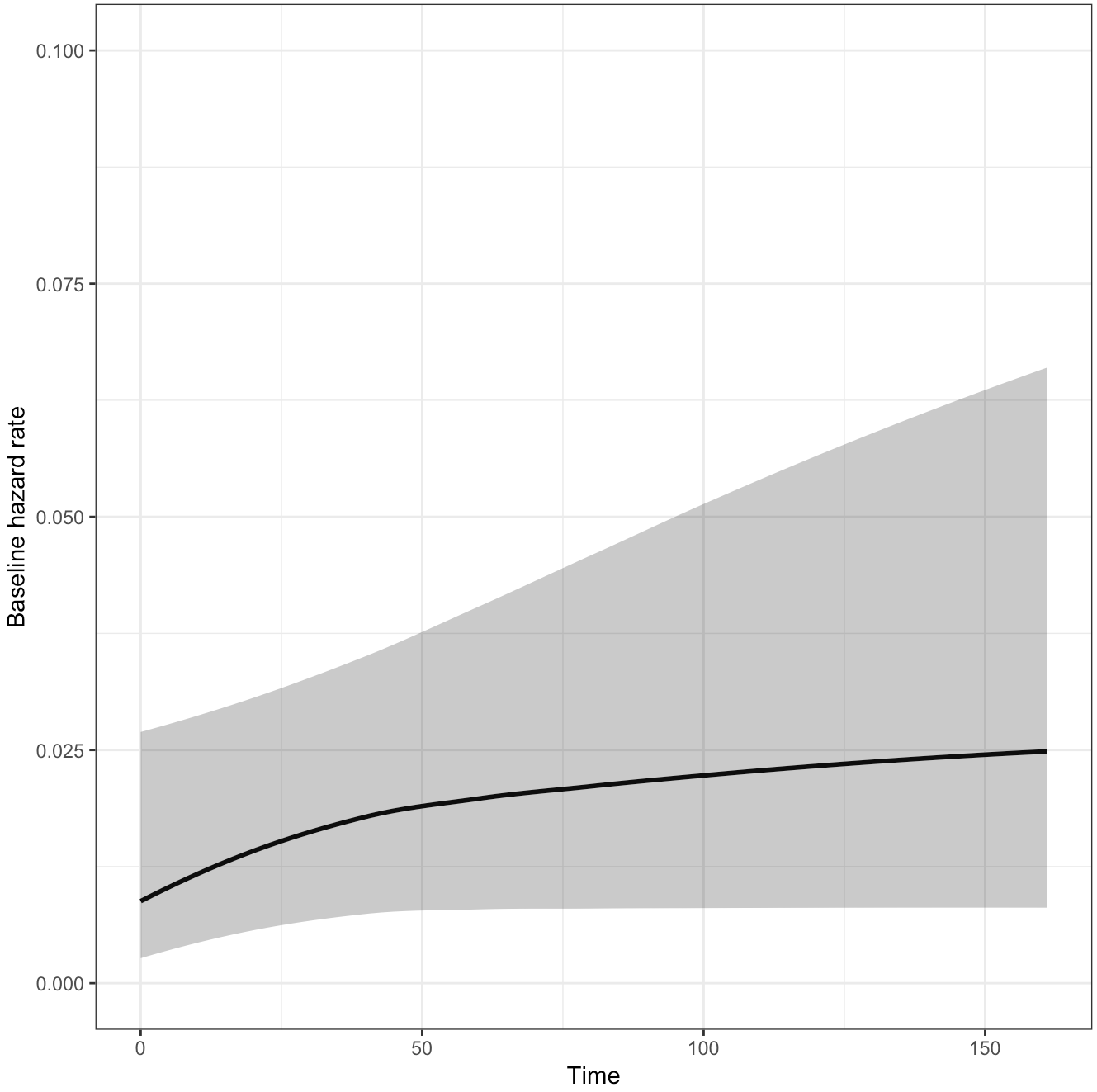
```
summary(stan_surv_weibull)
```

```
##
## Model Info:
##
##  function:        stan_surv
##  baseline hazard: weibull
##  formula:         Surv(time, status) ~ x
##  algorithm:       sampling
##  priors:          see help('prior_summary')
##  sample:          4000 (posterior sample size)
##  observations:    23
##  events:          18 (78.3%)
##  right censored:  5 (21.7%)
##  delayed entry:   no
##
## Estimates:
##                  mean   sd    2.5%   25%   50%   75%   97.5%
## (Intercept)     -5.2    1.0  -7.2   -5.9  -5.1  -4.5  -3.3
## xNonmaintained   1.1    0.5   0.1    0.8   1.1   1.5   2.2
## weibull-shape    1.2    0.2   0.8    1.1   1.2   1.4   1.7
## log-posterior  -88.6    1.3 -91.8  -89.2 -88.3 -87.7 -87.2
##
## Diagnostics:
##                mcse Rhat n_eff
## (Intercept)    0.0  1.0  1220
## xNonmaintained 0.0  1.0  1459
## weibull-shape  0.0  1.0  1252
## log-posterior  0.0  1.0  1172
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample size, an
d Rhat is the potential scale reduction factor on split chains (at convergence Rhat=1).
```

It looks like the Weibull rate parameter appears as the “(Intercept)” and the Weibull shape parameter is handled as an auxiliary parameter with its own prior.

```
plot(stan_surv_weibull) + coord_cartesian(ylim = c(0, 0.1))
```

```
## Warning: Removed 1 rows containing non-finite values (stat_smooth).
```


0.100
0.075
0.050
0.025
0.000
Baseline hazard rate
0
50
100
150
Time

This seems to be an increasing hazard setting with the Weibull fit.

```
tidybayes::tidy_draws(stan_surv_weibull)
```

```
## # A tibble: 4,000 x 12
##    .chain .iteration .draw `(Intercept)` xNonmaintained `weibull-shape` accept_stat__ stepsize__ treedepth__
##     <int>      <int> <int>         <dbl>          <dbl>           <dbl>         <dbl>      <dbl>       <dbl>
##  1      1          1     1         -4.43          0.986            1.02         0.953      0.121           2
##  2      1          2     2         -4.50          1.08             1.07         0.999      0.121           4
##  3      1          3     3         -4.84          1.16             1.15         1.000      0.121           5
##  4      1          4     4         -5.15          1.09             1.17         0.900      0.121           2
##  5      1          5     5         -4.99          1.20             1.21         0.932      0.121           3
##  6      1          6     6         -5.13          1.45             1.26         0.964      0.121           4
##  7      1          7     7         -4.59          1.70             1.06         0.986      0.121           4
##  8      1          8     8         -4.62          1.67             1.09         0.953      0.121           2
##  9      1          9     9         -4.90          1.19             1.29         0.914      0.121           4
## 10      1         10    10         -4.41          0.779            1.07         0.899      0.121           4
## # ... with 3,990 more rows, and 3 more variables: n_leapfrog__ <dbl>, divergent__ <dbl>, energy__ <dbl>
```

Gompertz model

```
stan_surv_gompertz <- stan_surv(formula = Surv(time, status) ~ x,
                                data = leukemia,
                                basehaz = "gompertz")
```

```
## Warning: There were 2 chains where the estimated Bayesian Fraction of Missing Information was low. See
## http://mc-stan.org/misc/warnings.html#bfmi-low
```

```
## Warning: Examine the pairs() plot to diagnose sampling problems
```

```
## Warning: Markov chains did not converge! Do not analyze results!
```

```
prior_summary(stan_surv_gompertz)
```

```
## Priors for model 'stan_surv_gompertz'
## ------
## Intercept
##  ~ normal(location = 0, scale = 20)
##
## Coefficients
##  ~ normal(location = 0, scale = 2.5)
##
## Auxiliary (gompertz-scale)
##  ~ half-normal(location = 0, scale = 2)
## ------
## See help('prior_summary.stanreg') for more details
```

```
summary(stan_surv_gompertz)
```

```
##
## Model Info:
##
##  function:        stan_surv
##  baseline hazard: gompertz
##  formula:         Surv(time, status) ~ x
##  algorithm:       sampling
##  priors:          see help('prior_summary')
##  sample:          4000 (posterior sample size)
##  observations:    23
##  events:          18 (78.3%)
##  right censored:  5 (21.7%)
##  delayed entry:   no
##
## Estimates:
##                  mean              sd                 2.5%               25%                50%
## (Intercept)                -1.7              2.8              -5.2              -4.4              -1.6
## xNonmaintained              2.1              1.1               0.3               1.1               2.9
## gompertz-scale              0.1              0.1               0.0               0.0               0.0
## log-posterior   -1275542225705.0   4896239114737.5 -17505883897310.5   -186248452118.5        -1759478.6
##                  75%               97.5%
## (Intercept)                 0.5              1.9
## xNonmaintained              3.1              3.3
## gompertz-scale              0.1              0.2
## log-posterior             -94.8            -93.7
##
## Diagnostics:
##                mcse            Rhat            n_eff
## (Intercept)               1.9             9.3  2
## xNonmaintained            0.7             3.1  2
## gompertz-scale            0.0            14.5  2
## log-posterior  1456695876789.9             1.3 11
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample size, an
d Rhat is the potential scale reduction factor on split chains (at convergence Rhat=1).
```

Gompertz with its default configuration did not converge. More investigation is needed.

Cubic M-spline

```
stan_surv_mspline <- stan_surv(formula = Surv(time, status) ~ x,
                               data = leukemia,
                               basehaz = "ms")
prior_summary(stan_surv_mspline)
```

```
## Priors for model 'stan_surv_mspline'
## ------
## Coefficients
##  ~ normal(location = 0, scale = 2.5)
##
## Auxiliary (M-spline-coefficients)
##  ~ normal(location = [0,0,0,...], scale = [20,20,20,...])
## ------
## See help('prior_summary.stanreg') for more details
```

```
summary(stan_surv_mspline)
```

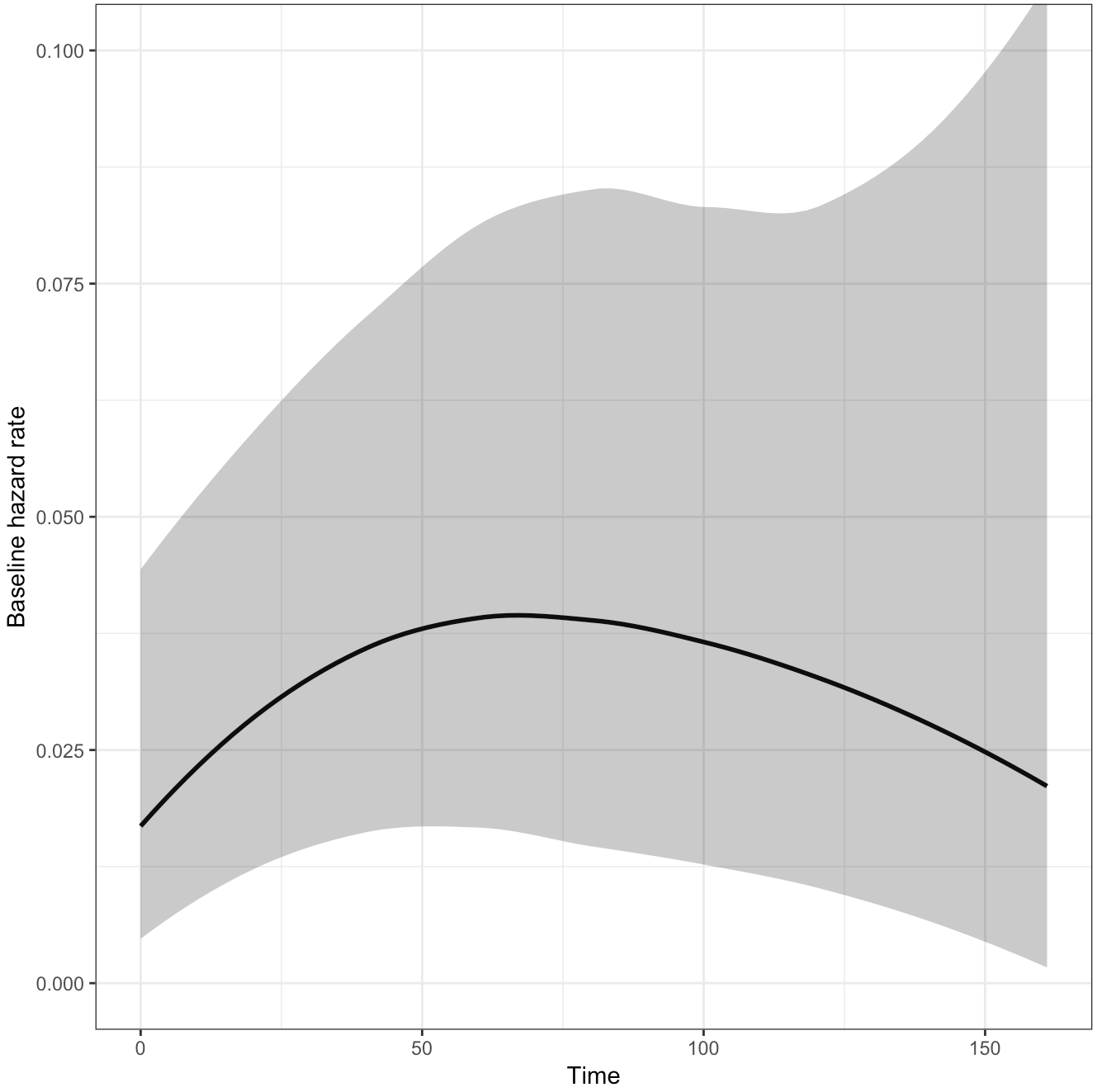
```
##
## Model Info:
##
##  function:        stan_surv
##  baseline hazard: M-splines on hazard scale
##  formula:         Surv(time, status) ~ x
##  algorithm:       sampling
##  priors:          see help('prior_summary')
##  sample:          4000 (posterior sample size)
##  observations:    23
##  events:          18 (78.3%)
##  right censored:  5 (21.7%)
##  delayed entry:   no
##
## Estimates:
##                   mean   sd     2.5%   25%    50%    75%    97.5%
## xNonmaintained     0.5    0.4   -0.4    0.2    0.5    0.8    1.4
## m-splines-coef1    0.0    0.0    0.0    0.0    0.0    0.1    0.2
## m-splines-coef2    0.2    0.1    0.0    0.1    0.2    0.3    0.5
## m-splines-coef3    1.1    0.7    0.1    0.5    1.0    1.5    2.7
## m-splines-coef4    2.3    1.6    0.1    1.1    2.1    3.2    6.0
## m-splines-coef5    1.2    1.1    0.0    0.3    0.8    1.6    4.2
## m-splines-coef6    1.0    1.0    0.0    0.3    0.7    1.4    3.8
## log-posterior   -113.8    2.1 -118.9 -115.0 -113.5 -112.3 -110.8
##
## Diagnostics:
##                 mcse Rhat n_eff
## xNonmaintained  0.0  1.0  3139
## m-splines-coef1 0.0  1.0  3977
## m-splines-coef2 0.0  1.0  2501
## m-splines-coef3 0.0  1.0  2120
## m-splines-coef4 0.0  1.0  2498
## m-splines-coef5 0.0  1.0  4497
## m-splines-coef6 0.0  1.0  4446
## log-posterior   0.1  1.0  1254
```

```
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample size, an
d Rhat is the potential scale reduction factor on split chains (at convergence Rhat=1).
```

The default configuration of the spline is explained in the basehaz_ops argument.

```
#' @param basehaz_ops A named list specifying options related to the baseline
#'   hazard. Currently this can include: \cr
#'   \itemize{
#'     \item \code{df}: a positive integer specifying the degrees of freedom
#'     for the M-splines or B-splines. An intercept is included in the spline
#'     basis and included in the count of the degrees of freedom, such that
#'     two boundary knots and \code{df - 4} internal knots are used to generate
#'     the cubic spline basis. The default is \code{df = 6}; that is, two
#'     boundary knots and two internal knots.
#'     \item \code{knots}: An optional numeric vector specifying internal
#'     knot locations for the M-splines or B-splines. Note that \code{knots}
#'     cannot be specified if \code{df} is specified. If \code{knots} are
#'     \strong{not} specified, then \code{df - 4} internal knots are placed
#'     at equally spaced percentiles of the distribution of uncensored event
#'     times.
#'   }
#'   Note that for the M-splines and B-splines - in addition to any internal
#'   \code{knots} - a lower boundary knot is placed at the earliest entry time
#'   and an upper boundary knot is placed at the latest event or censoring time.
#'   These boundary knot locations are the default and cannot be changed by the
#'   user.
```

```
plot(stan_surv_mspline) + coord_cartesian(ylim = c(0, 0.1))
```


0.100
0.075
0.050
0.025
0.000
Baseline hazard rate
0
50
100
150
Time

With more flexible modeling, the baseline hazard is increasing then decreasing.

Cubic B-spline

```
stan_surv_bspline <- stan_surv(formula = Surv(time, status) ~ x,
                               data = leukemia,
                               basehaz = "bs")
```

```
## Warning: There were 1 divergent transitions after warmup. Increasing adapt_delta above 0.95 may help. See
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
```

```
## Warning: Examine the pairs() plot to diagnose sampling problems
```

```
prior_summary(stan_surv_bspline)
```

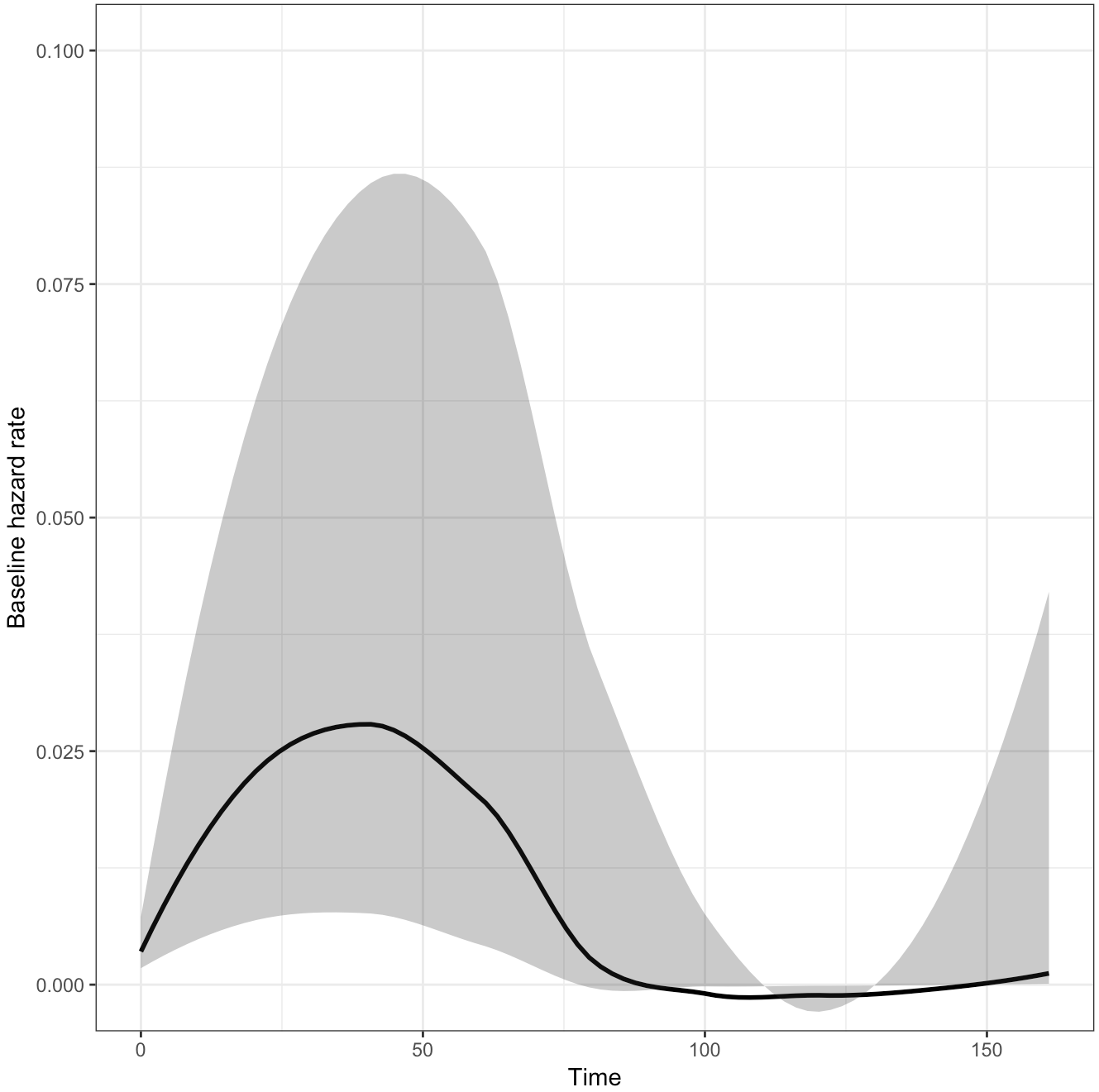
```
## Priors for model 'stan_surv_bspline'
## ------
## Coefficients
##  ~ normal(location = 0, scale = 2.5)
##
## Auxiliary (B-spline-coefficients)
##  ~ normal(location = [0,0,0,...], scale = [20,20,20,...])
## ------
## See help('prior_summary.stanreg') for more details
```

```
summary(stan_surv_bspline)
```

```
##
## Model Info:
##
##  function:        stan_surv
##  baseline hazard: B-splines on log hazard scale
##  formula:         Surv(time, status) ~ x
##  algorithm:       sampling
##  priors:          see help('prior_summary')
##  sample:          4000 (posterior sample size)
##  observations:    23
##  events:          18 (78.3%)
##  right censored:  5 (21.7%)
##  delayed entry:   no
##
## Estimates:
##                   mean   sd    2.5%   25%   50%   75%   97.5%
## xNonmaintained    0.8    0.5  -0.1    0.5   0.8   1.2   1.8
## b-splines-coef1 -11.1    3.9 -19.9  -13.2 -10.5  -8.3  -5.1
## b-splines-coef2  -2.1    1.3  -4.5   -2.9  -2.1  -1.2   0.4
## b-splines-coef3  -5.7    1.1  -7.9   -6.3  -5.6  -4.9  -3.7
## b-splines-coef4   2.8    3.2  -3.0    0.7   2.7   4.9   9.3
## b-splines-coef5 -20.7   10.9 -45.3  -27.2 -19.5 -12.8  -2.7
## b-splines-coef6 -18.7   13.2 -52.4  -25.8 -16.0  -8.5  -1.9
## log-posterior   -85.9    2.0 -90.9  -87.0 -85.5 -84.5 -83.1
##
## Diagnostics:
##                 mcse Rhat n_eff
## xNonmaintained  0.0  1.0  2379
## b-splines-coef1 0.1  1.0  1288
## b-splines-coef2 0.0  1.0  1533
## b-splines-coef3 0.0  1.0  1100
## b-splines-coef4 0.1  1.0  1082
## b-splines-coef5 0.3  1.0  1336
## b-splines-coef6 0.4  1.0  1013
## log-posterior   0.1  1.0  1158
```

```
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample size, an
d Rhat is the potential scale reduction factor on split chains (at convergence Rhat=1).
```

```
plot(stan_surv_bspline) + coord_cartesian(ylim = c(0, 0.1))
```

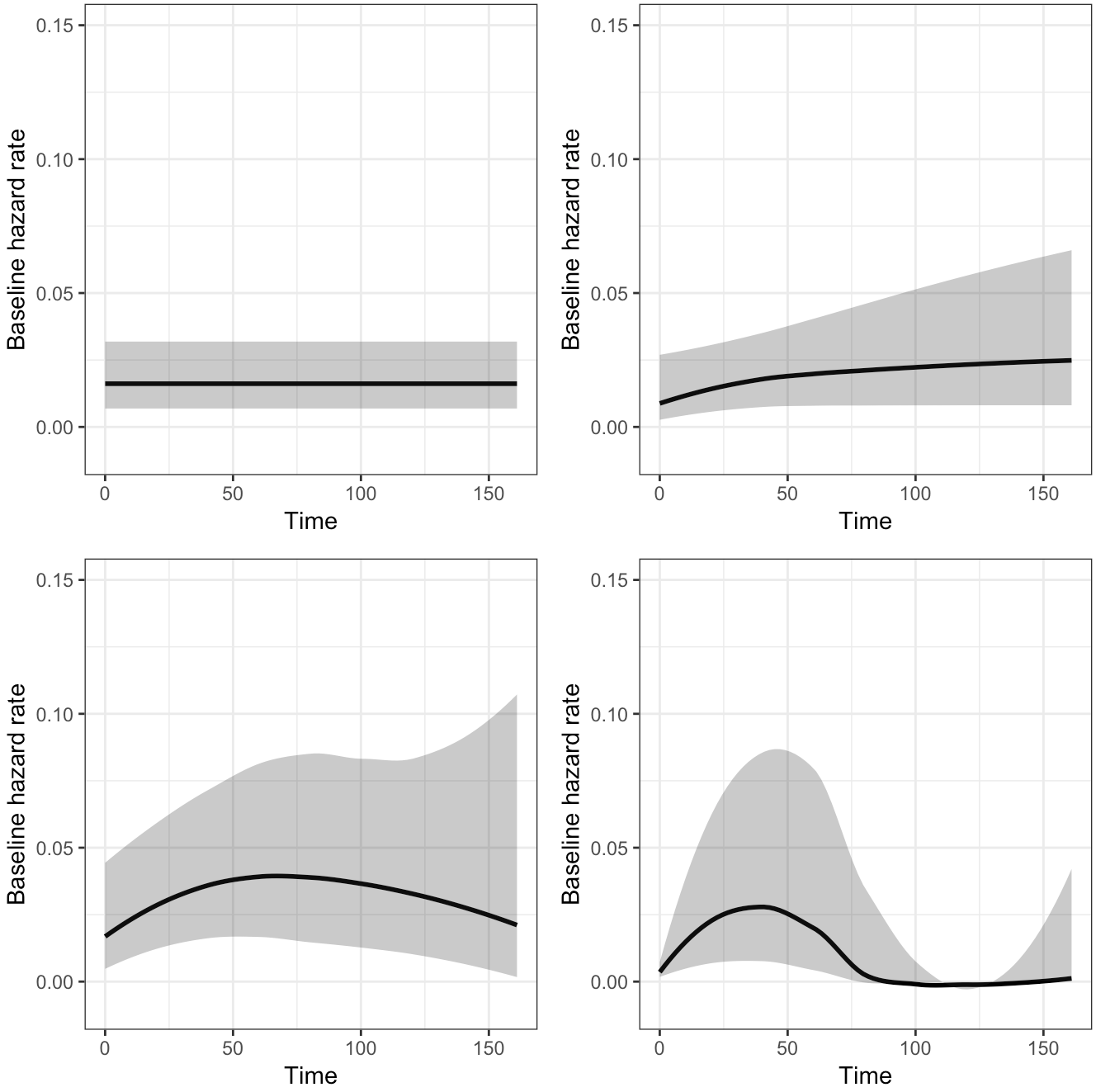

0.100
0.075
0.050
0.025
0.000
Baseline hazard rate
0
50
100
150
Time

This baseline hazard estimate with B-spline appears wired. There is a region with a negative hazard, so there is some posterior resurrection happening…

Baseline hazard function comparison

```
bayesplot::bayesplot_grid(plot(stan_surv_exponential),
                          plot(stan_surv_weibull),
                          plot(stan_surv_mspline),
                          plot(stan_surv_bspline),
                          ylim = c(-0.01, 0.15))
```

```
## Warning: Removed 1 rows containing non-finite values (stat_smooth).
```


Baseline hazard rate
0.15
0.10
0.05
0.00
0
50
100
150
Time

- Top Page: http://rpubs.com/kaz_yos/ (http://rpubs.com/kaz_yos/)
- Github: https://github.com/kaz-yos (https://github.com/kaz-yos)

```
print(sessionInfo())
```

```
## R version 3.5.1 (2018-07-02)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS  10.14
##
## Matrix products: default
## BLAS: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] parallel  stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] bindrcpp_0.2.2    survival_2.43-1   bayesplot_1.6.0   tidybayes_1.0.3   rstanarm_2.18.1   Rcpp_0.12.19
##  [7] forcats_0.3.0     stringr_1.3.1     dplyr_0.7.7       purrr_0.2.5       readr_1.1.1       tidyr_0.8.2
## [13] tibble_1.4.2      ggplot2_3.1.0     tidyverse_1.2.1   doRNG_1.7.1       rngtools_1.3.1    pkgmaker_0.27
## [19] registry_0.5      doParallel_1.0.14 iterators_1.0.10  foreach_1.4.4     knitr_1.20
##
## loaded via a namespace (and not attached):
##  [1] minqa_1.2.4               colorspace_1.3-2          ggridges_0.5.1            rsconnect_0.8.8
##  [5] rprojroot_1.3-2           ggstance_0.3.1            markdown_0.8              base64enc_0.1-3
##  [9] rstudioapi_0.8            rstan_2.18.1              svUnit_0.7-12             DT_0.4
## [13] fansi_0.4.0               lubridate_1.7.4           xml2_1.2.0                codetools_0.2-15
## [17] splines_3.5.1             shinythemes_1.1.1         jsonlite_1.5              nloptr_1.2.1
## [21] broom_0.5.0               shiny_1.1.0               compiler_3.5.1            httr_1.3.1
## [25] backports_1.1.2           assertthat_0.2.0          Matrix_1.2-14             lazyeval_0.2.1
## [29] cli_1.0.1                 later_0.7.5               htmltools_0.3.6           prettyunits_1.0.2
## [33] tools_3.5.1               igraph_1.2.2              coda_0.19-2               gtable_0.2.0
## [37] glue_1.3.0                reshape2_1.4.3            cellranger_1.1.0          nlme_3.1-137
## [41] crosstalk_1.0.0           ps_1.2.0                  lme4_1.1-18-1             rvest_0.3.2
## [45] mime_0.6                  miniUI_0.1.1.1            gtools_3.8.1              MASS_7.3-51
## [49] zoo_1.8-4                 scales_1.0.0              colourpicker_1.0          hms_0.4.2
## [53] promises_1.0.1            inline_0.3.15             shinystan_2.5.0           yaml_2.2.0
```

```
## [57] gridExtra_2.3             loo_2.0.0               StanHeaders_2.18.0      stringi_1.2.4
## [61] dygraphs_1.1.1.6          pkgbuild_1.0.2          bibtex_0.4.2            rlang_0.3.0.1
## [65] pkgconfig_2.0.2           matrixStats_0.54.0      evaluate_0.12           lattice_0.20-35
## [69] bindr_0.1.1               splines2_0.2.8          labeling_0.3            rstantools_1.5.1
## [73] htmlwidgets_1.3           tidyselect_0.2.5        processx_3.2.0          plyr_1.8.4
## [77] magrittr_1.5              R6_2.3.0                pillar_1.3.0            haven_1.1.2
## [81] withr_2.1.2               xts_0.11-1              modelr_0.1.2            crayon_1.3.4
## [85] arrayhelpers_1.0-20160527 utf8_1.1.4              rmarkdown_1.10          grid_3.5.1
## [89] readxl_1.1.0              callr_3.0.0             threejs_0.3.1           digest_0.6.18
## [93] xtable_1.8-3              httpuv_1.4.5            stats4_3.5.1            munsell_0.5.0
## [97] shinyjs_1.0
```

```
## Record execution time and multicore use
end_time <- Sys.time()
diff_time <- difftime(end_time, start_time, units = "auto")
cat("Started  ", as.character(start_time), "\n",
    "Finished ", as.character(end_time), "\n",
    "Time difference of ", diff_time, " ", attr(diff_time, "units"), "\n",
    "Used ", foreach::getDoParWorkers(), " cores\n",
    "Used ", foreach::getDoParName(), " as backend\n",
    sep = "")
```

```
## Started  2018-11-12 06:23:05
## Finished 2018-11-12 06:23:21
## Time difference of 16.06129 secs
## Used 12 cores
## Used doParallelMC as backend
```