

data_exploration

Jinghan Cui

12/2/2022

```
MFPS_PNAS_SURVIVAL_DATA <- read_dta("dataverse_files/MFPS_PNAS_SURVIVAL_DATA.dta")
summary(MFPS_PNAS_SURVIVAL_DATA)
```

```
##      caseid_n      year      treatment      preg_bl
## Min.      : 1.0    Min.    :2016    Min.    :0.0000    Min.    :0.0000
## 1st Qu.: 540.8    1st Qu.:2016    1st Qu.:0.0000    1st Qu.:0.0000
## Median :1068.5    Median :2016    Median :0.0000    Median :0.0000
## Mean   :1069.4    Mean     :2016    Mean    :0.4757    Mean    :0.4904
## 3rd Qu.:1601.5    3rd Qu.:2016    3rd Qu.:1.0000    3rd Qu.:1.0000
## Max.   :2139.0    Max.     :2016    Max.    :1.0000    Max.    :1.0000
##
## total_alive_bl  age_group_bl  sex_age_bl  edu_primary_bl
## Min.    :0.000    Min.    :0.000    Min.    :11.00    Min.    :0.0000
## 1st Qu.:1.000    1st Qu.:1.000    1st Qu.:17.00    1st Qu.:0.0000
## Median :2.000    Median :2.000    Median :19.00    Median :1.0000
## Mean   :2.281    Mean    :1.739    Mean    :18.91    Mean    :0.5615
## 3rd Qu.:3.000    3rd Qu.:2.000    3rd Qu.:20.00    3rd Qu.:1.0000
## Max.   :9.000    Max.    :3.000    Max.    :32.00    Max.    :1.0000
## NA's    :1      NA's      :5
## ever_use_bl     work_bl     religion_r_bl  ethnicity_r_bl
## Min.    :0.0000    Min.    :0.0000    Min.    :0.0000    Min.    :0.0000
## 1st Qu.:1.0000    1st Qu.:0.0000    1st Qu.:1.0000    1st Qu.:0.0000
## Median :1.0000    Median :0.0000    Median :1.0000    Median :0.0000
## Mean   :0.7698    Mean    :0.1021    Mean    :0.8313    Mean    :0.4187
## 3rd Qu.:1.0000    3rd Qu.:0.0000    3rd Qu.:1.0000    3rd Qu.:1.0000
## Max.   :1.0000    Max.    :1.0000    Max.    :1.0000    Max.    :1.0000
##
## area_bl     event_next_preg_24 month_next_preg  _st
## Min.    : 7.00    Min.    :0.00000    Min.    : 0.60    Min.    :1
## 1st Qu.:50.00    1st Qu.:0.00000    1st Qu.:18.30    1st Qu.:1
## Median :50.00    Median :0.00000    Median :22.07    Median :1
## Mean   :43.17    Mean    :0.06772    Mean    :20.27    Mean    :1
## 3rd Qu.:56.00    3rd Qu.:0.00000    3rd Qu.:24.00    3rd Qu.:1
## Max.   :56.00    Max.    :1.00000    Max.    :24.00    Max.    :1
##
##      _d      _t      _t0      _est_m1      _est_m2
## Min.    :0.00000    Min.    : 0.60    Min.    :0    Min.    :1    Min.    :0.0000
## 1st Qu.:0.00000    1st Qu.:18.30    1st Qu.:0    1st Qu.:1    1st Qu.:1.0000
## Median :0.00000    Median :22.07    Median :0    Median :1    Median :1.0000
## Mean   :0.06772    Mean    :20.27    Mean    :0    Mean    :1    Mean    :0.9972
## 3rd Qu.:0.00000    3rd Qu.:24.00    3rd Qu.:0    3rd Qu.:1    3rd Qu.:1.0000
```

```
## Max. :1.00000 Max. :24.00 Max. :0 Max. :1 Max. :1.0000
##
```

Data exploration

In survival analysis, the outcome or dependent variable is the *time to event* where some event times are not observed (IE they are censored).

Here we consider the more common scenario of *right-censoring*. This is the case where the terminating event is not observed. Observations are instead censored at time t .

Our first analysis will treat **overall survival** as the event of interest, as opposed to progression-free survival. In this cohort, the overall survival is described by two variables: `os_status` & `os_months`.

We will start by inspecting these data.

```
which(is.na(MFPS_PNAS_SURVIVAL_DATA))
```

```
## [1] 8679 10697 10938 11360 12223 12278
```

```
sum(is.na(MFPS_PNAS_SURVIVAL_DATA))
```

```
## [1] 6
```

```
sapply(MFPS_PNAS_SURVIVAL_DATA, function(x) sum(is.na (x)))
```

```
##      caseid_n      year      treatment      preg_bl
##           0           0           0           0
## total_alive_bl age_group_bl sex_age_bl edu_primary_bl
##           1           0           5           0
## ever_use_bl    work_bl    religion_r_bl ethnicity_r_bl
##           0           0           0           0
##      area_bl event_next_preg_24 month_next_preg      _st
##           0           0           0           0
##           _d           _t           _t0      _est_m1
##           0           0           0           0
##      _est_m2
##           0
```

```
###MALAWI FAMILY PLANNING STUDY
```

```
###R-SCRIPT 5: SURVIVAL PLOT FILE
```

```
###MAHESH KARRA
```

```
###MAY 10, 2022
```

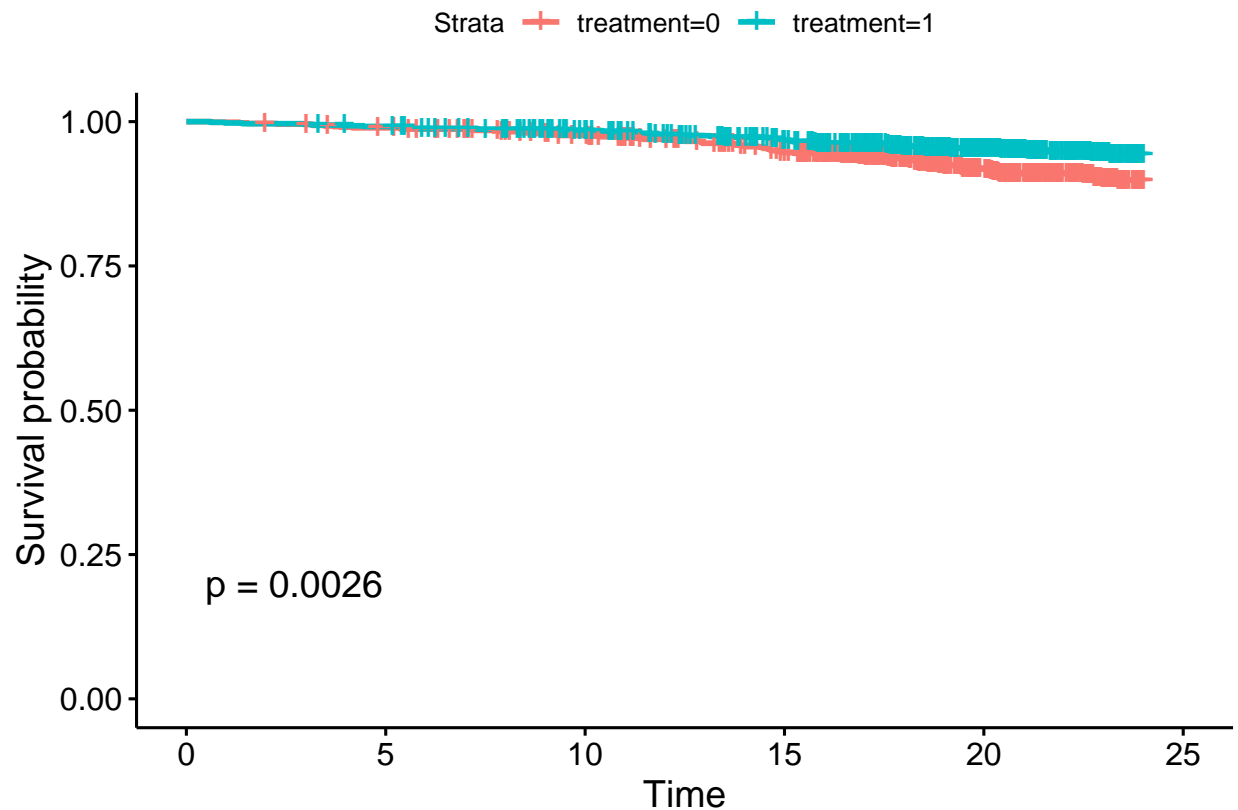
```
###STEP 0: SETTING WORKING DIRECTORY
```

```
###To use the correct file directory, replace "C:/Users/mvkarra/Documents/Harvard 2015-2016/POP CENTER/
```

```
###Make sure that the Results subfolder is created to store the results from this analysis.
```

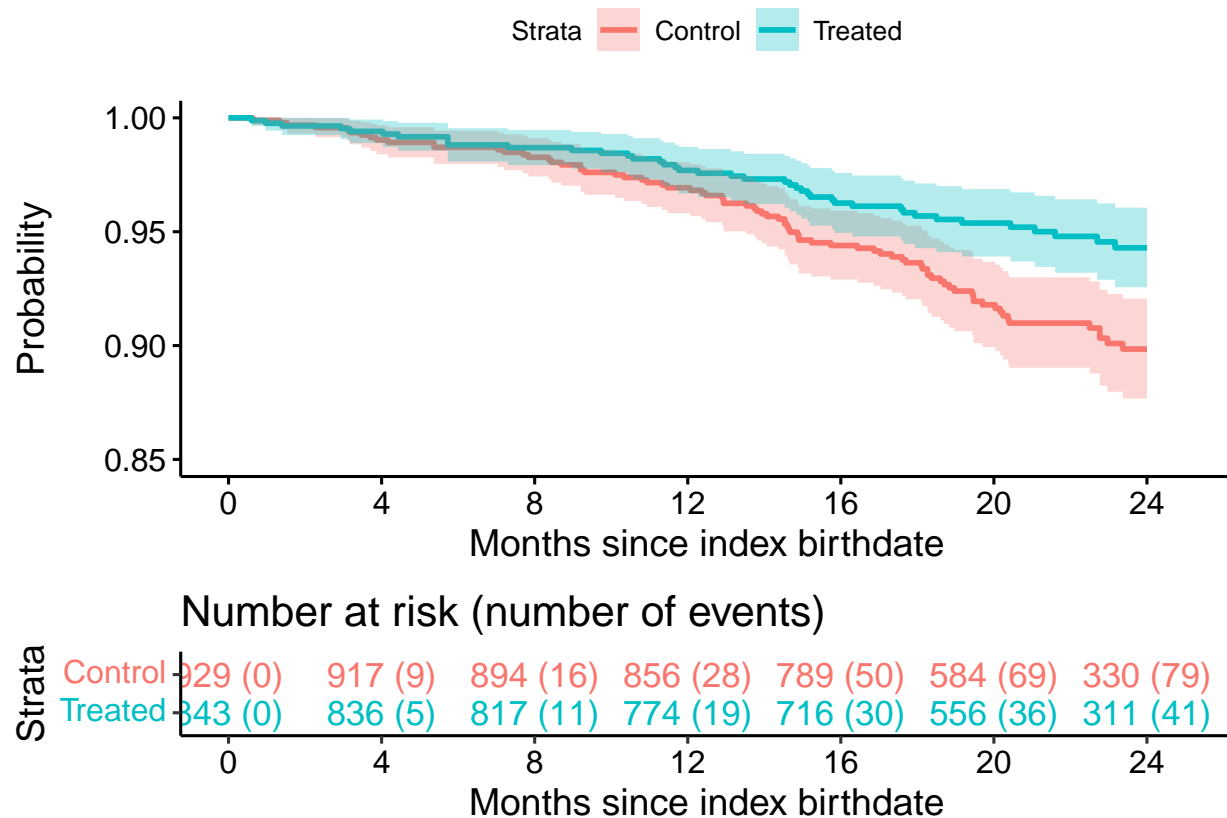
```
###SET WORKING DIRECTORY
```

```
fit <- survfit(Surv(month_next_preg,event_next_preg_24) ~ treatment, data = MFPS_PNAS_SURVIVAL_DATA)
ggsurvplot(fit, data = MFPS_PNAS_SURVIVAL_DATA, pval = TRUE)
```



```
fit_cox <- coxph(Surv(month_next_preg,event_next_preg_24) ~ treatment, data = MFPS_PNAS_SURVIVAL_DATA)%>%
  broom::tidy(exp = TRUE)
aft.fit <- survreg(Surv(month_next_preg,event_next_preg_24) ~ treatment, dist="weibull", data = MFPS_PNAS_SURVIVAL_DATA)%>%
  broom::tidy(exp = TRUE)
###K-M PLOT
ggsurvplot(
  fit,
  data = MFPS_PNAS_SURVIVAL_DATA,
  censor=FALSE,
  conf.int = TRUE,
  risk.table = "nrisk_cumevents",
  risk.table.col = "strata", # Risk table color by groups
  risk.table.height = 0.3, # Useful to change when you have multiple groups
  break.time.by = 4,
  ylim = c(0.85, 1),
  xlab = "Months since index birthdate",
  ylab = "Probability",
  legend.labs =
    c("Control", "Treated"),
  pval = TRUE
)
```

```
## Warning: Removed 1 rows containing missing values ('geom_text()').
## Removed 1 rows containing missing values ('geom_text()').
```



prior predictive checks

```
CHAINS <- 4
CORES <- 2
ITER <- 2000
SEED <- 1234

#exponential proportional hazard function that has a constant baseline hazard.
#Setting the prior_PD argument equal to TRUE ensures that we do not condition on the outcome data and i
prior.stan.const <- stan_surv(
  Surv(month_next_preg, event_next_preg_24) ~ treatment,
  data = MFPS_PNAS_SURVIVAL_DATA,
  basehaz = "exp",
  prior_PD = TRUE,
  chains = CHAINS,
  cores = CORES,
  iter = ITER,
  seed = SEED)
prior.stan.const
```

```
## stan_surv
## baseline hazard: exponential
## formula: Surv(month_next_preg, event_next_preg_24) ~ treatment
## observations: 1772
## events: 120 (6.8%)
## right censored: 1652 (93.2%)
## delayed entry: no
```

```
## -----
##           Median MAD_SD exp(Median)
## (Intercept) -5.4   20.4    NA
## treatment    0.0    2.6    1.0
##
## -----
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

```
prior_summary(prior.stan.const)
```

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

```
#marginal prior distributions for log HR
library(bayesplot)
```

```
## Warning: package 'bayesplot' was built under R version 4.1.2

## This is bayesplot version 1.10.0

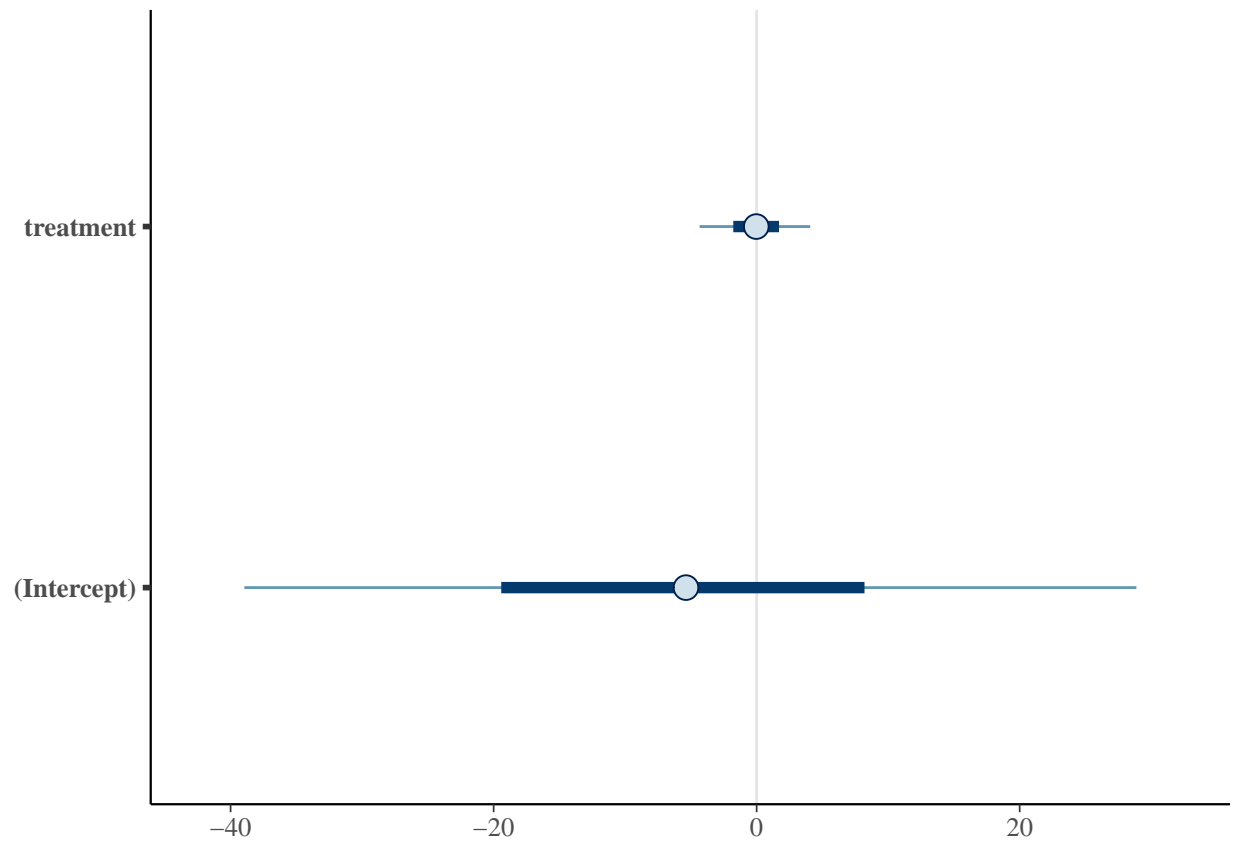
## - Online documentation and vignettes at mc-stan.org/bayesplot

## - bayesplot theme set to bayesplot::theme_default()

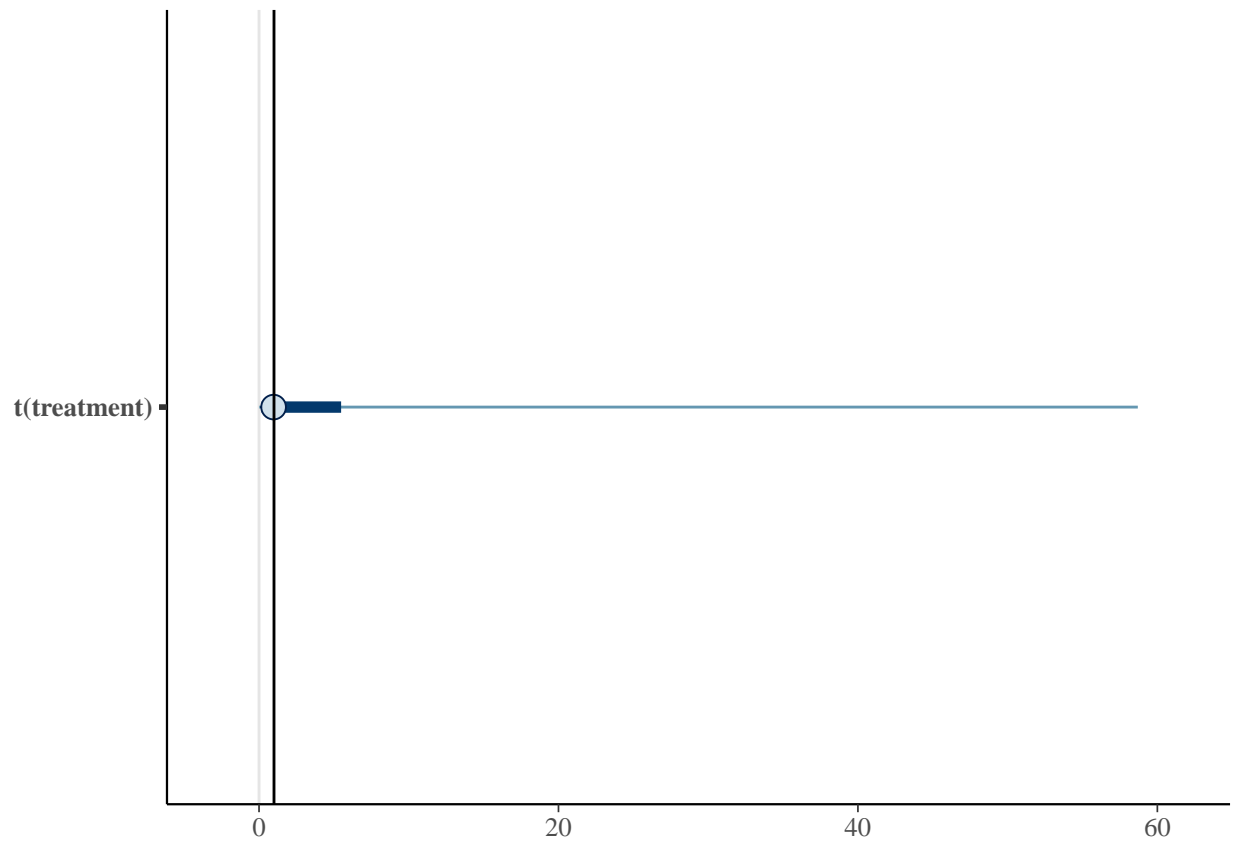
##   * Does _not_ affect other ggplot2 plots

##   * See ?bayesplot_theme_set for details on theme setting
```

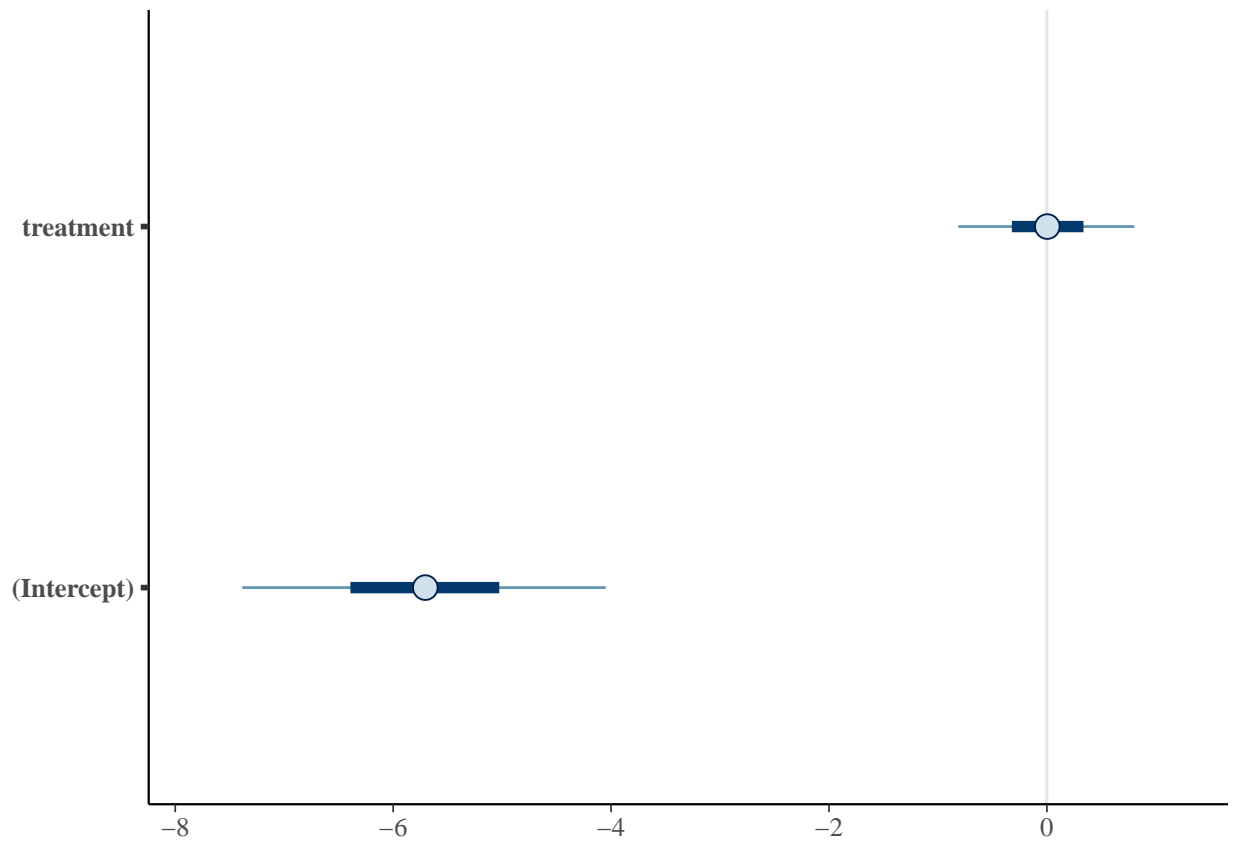
```
mcmc_intervals(prior.stan.const, pars = c("treatment", "(Intercept)"))
```



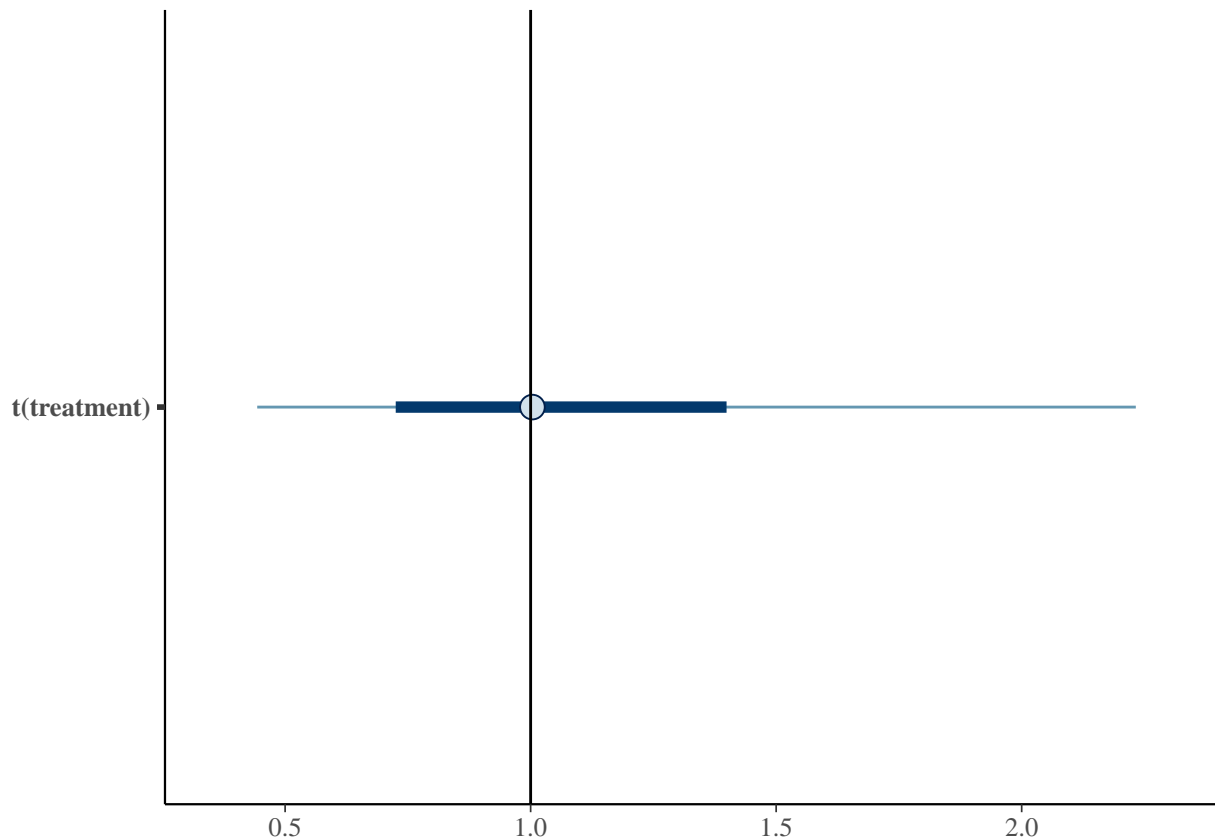
```
#marginal prior distribution for the HR for treatment  
mcmc_intervals(prior.stan.const, pars = c("treatment"),  
transformations = exp) + vline_at(1)
```



```
#use a different prior distribution
prior.stan.const <- update(prior.stan.const,
prior_intercept = normal(0, 1),
prior = normal(0, 0.5))
#marginal prior distributions for log HR
mcmc_intervals(prior.stan.const, pars = c("treatment", "(Intercept)"))
```



```
#marginal prior distribution for the HR for treatment  
mcmc_intervals(prior.stan.const, pars = c("treatment"),  
transformations = exp) + vline_at(1)
```

#a HR around 1 would be plausible for a clinical trial

Model estimation

```
fit.stan.const <- update(prior.stan.const, prior_PD = FALSE)
print(fit.stan.const,digits=3)

## stan_surv
## baseline hazard: exponential
## formula:      Surv(month_next_preg, event_next_preg_24) ~ treatment
## observations: 1772
## events:      120 (6.8%)
## right censored: 1652 (93.2%)
## delayed entry: no
## -----
##               Median MAD_SD exp(Median)
## (Intercept) -5.498  0.108    NA
## treatment   -0.497  0.191  0.609
##
## -----
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

```
#MAD_SD: a measure of the standard deviation of the marginal posterior distribution of the treatment ef
```

```
#Prior vs posterior checks
```

```
require("cowplot")
```

```
## Loading required package: cowplot
```

```
##
```

```
## Attaching package: 'cowplot'
```

```
## The following object is masked from 'package:ggpubr':
```

```
##
```

```
##      get_legend
```

```
#intercept and log hazard ratio
```

```
plot_grid(  
  bayesplot_grid(mcmc_intervals(prior.stan.const),  
    mcmc_intervals(fit.stan.const),  
    titles = c("Prior", "Posterior"),  
    xlim = c(-8, 1),  
    grid_args = list(nrow = 2)),  
  bayesplot_grid(mcmc_hist(prior.stan.const),  
    mcmc_hist(fit.stan.const),  
    titles = c("Prior", "Posterior"),  
    grid_args = list(nrow = 2)),  
  ncol = 2  
)
```

```
## Scale for x is already present.
```

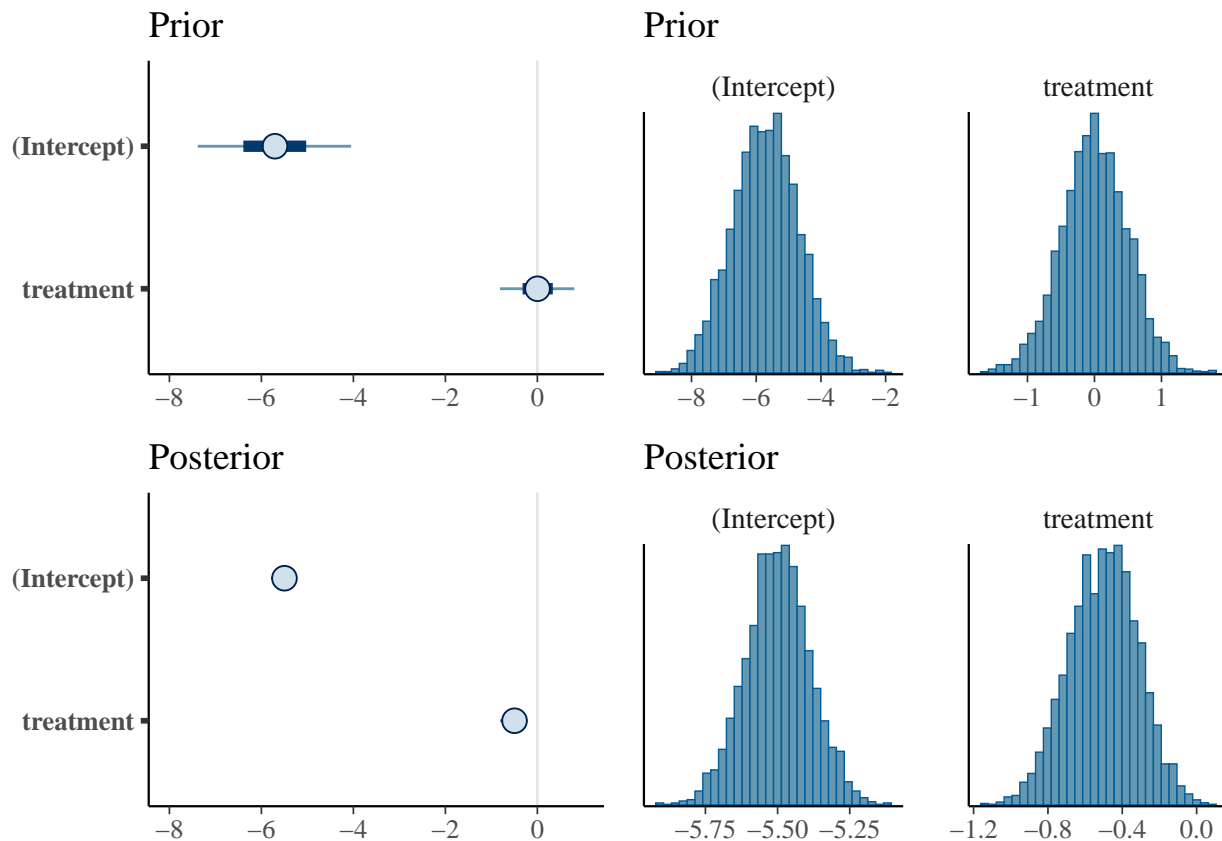
```
## Adding another scale for x, which will replace the existing scale.
```

```
## Scale for x is already present.
```

```
## Adding another scale for x, which will replace the existing scale.
```

```
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
```

```
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
```



```
#fit a cox model
```

```
fit.coxph <- coxph(Surv(month_next_preg, event_next_preg_24) ~ treatment, data = MFPS_PNAS_SURVIVAL_DATA)
fit.coxph
```

```
## Call:
```

```
## coxph(formula = Surv(month_next_preg, event_next_preg_24) ~ treatment,
##       data = MFPS_PNAS_SURVIVAL_DATA, x = TRUE)
```

```
##
```

```
##           coef exp(coef) se(coef)      z      p
## treatment -0.5710    0.5649  0.1925 -2.967 0.00301
```

```
##
```

```
## Likelihood ratio test=9.23 on 1 df, p=0.002376
```

```
## n= 1772, number of events= 120
```

```
#hazard ratio
```

```
add_cox_hr <- vline_at(exp(coef(fit.coxph)), color = "green")
```

```
bayesplot_grid(
```

```
mcmc_hist(prior.stan.const,
```

```
  pars = c("treatment"),
```

```
  transformations = exp,
```

```
  binwidth = 0.05) + add_cox_hr,
```

```
mcmc_hist(fit.stan.const,
```

```
  pars = c("treatment"),
```

```
  transformations = exp,
```

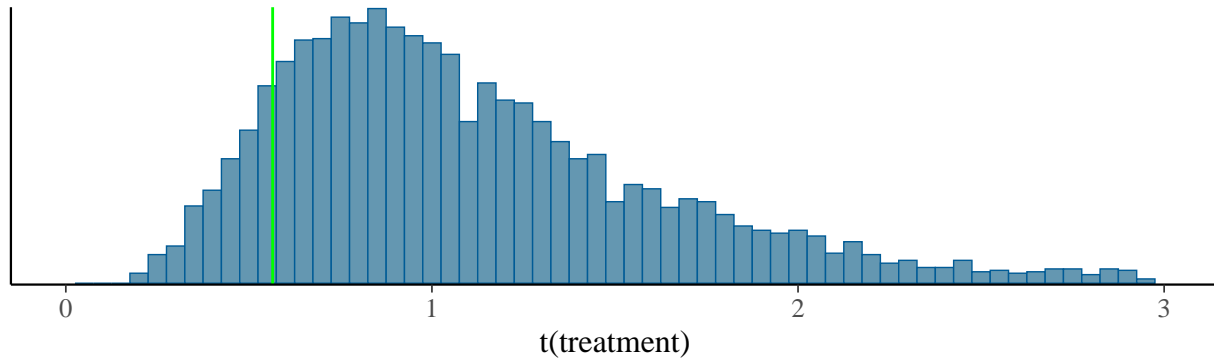
```
  binwidth = 0.05) + add_cox_hr,
```

```

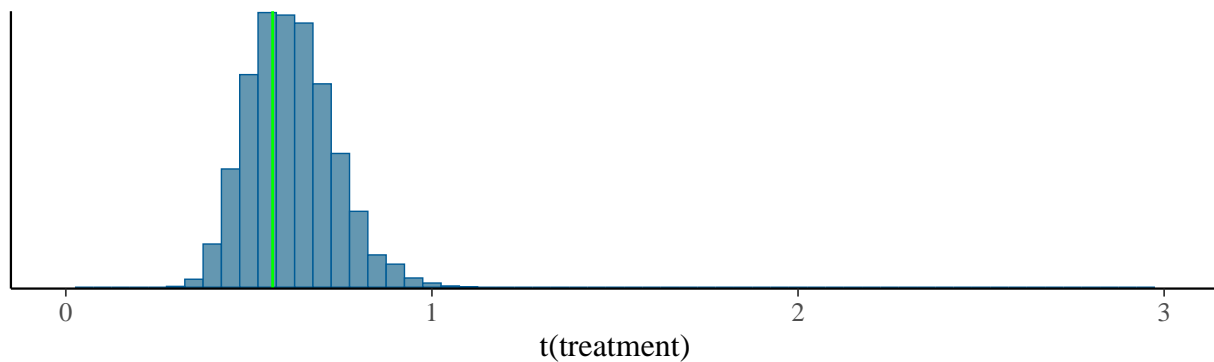
titles = c("Prior", "Posterior"),
xlim = c(0, 3),
grid_args = list(nrow = 2)
)

```

Prior



Posterior



extend to different baseline ditribution

```

# weibull model
fit.stan.weib <- update(fit.stan.const,
basehaz = "weibull")
# gompertz model
fit.stan.gomp <- update(fit.stan.const,
basehaz = "gompertz")
# cubic m-spline model (with df = 5)
fit.stan.ms5 <- update(fit.stan.const, basehaz = "ms", basehaz_ops = list(df = 5))
# cubic m-spline model (with df = 10)
fit.stan.ms10 <- update(fit.stan.const,
basehaz = "ms",
basehaz_ops = list(df = 10))
# cubic b-spline model (with df = 5)
fit.stan.bs5 <- update(fit.stan.const,
basehaz = "bs",
basehaz_ops = list(df = 5))

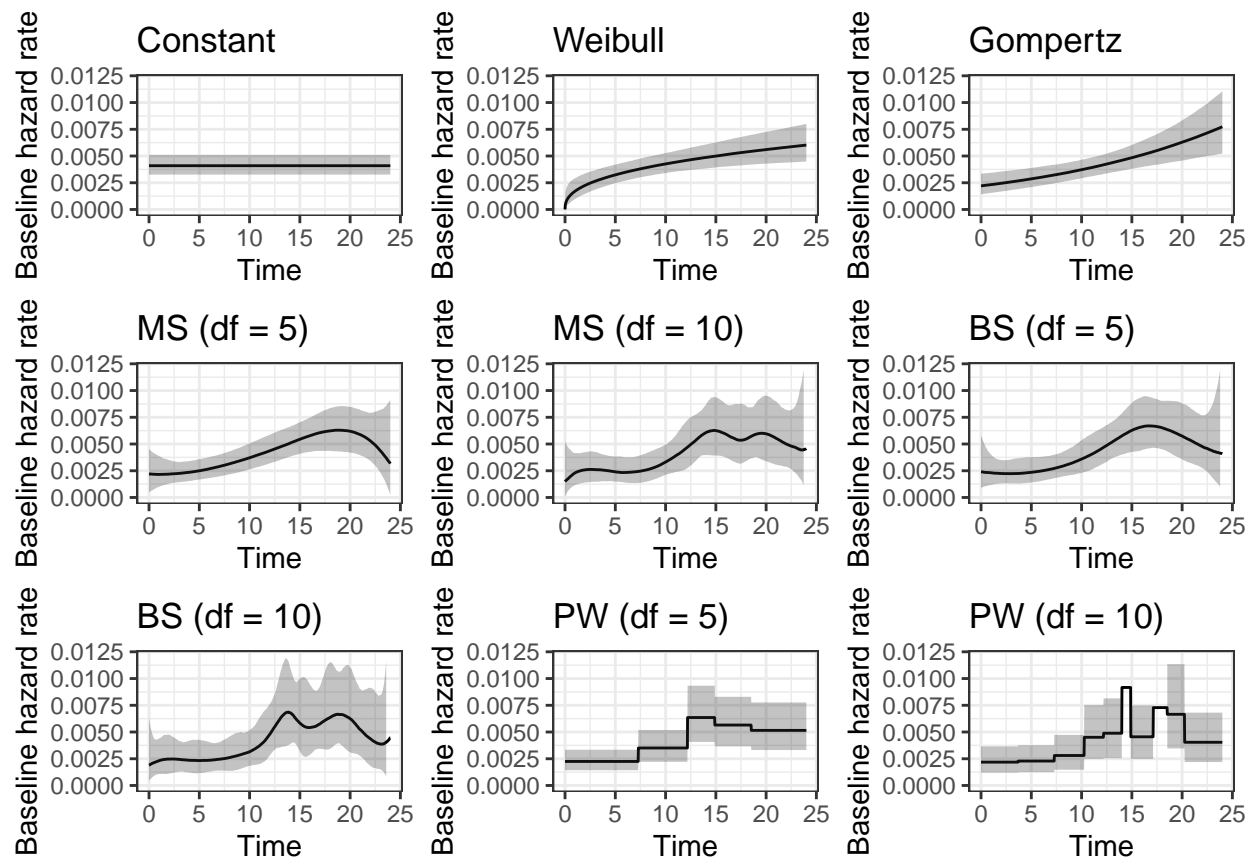
```

```

# cubic s-spline model (with df = 10)
fit.stan.bs10 <- update(fit.stan.const,
  basehaz = "bs",
  basehaz_ops = list(df = 10))
# piecewise constant model (with df = 5)
fit.stan.pw5 <- update(fit.stan.const,
  basehaz = "ms",
  basehaz_ops = list(degree = 0, df = 5))
# piecewise constant model (with df = 10)
fit.stan.pw10 <- update(fit.stan.const,
  basehaz = "ms",
  basehaz_ops = list(degree = 0, df = 10))
fits_stan <- list(
  "Constant" = fit.stan.const,
  "Weibull" = fit.stan.weib,
  "Gompertz" = fit.stan.gomp,
  "MS (df = 5)" = fit.stan.ms5,
  "MS (df = 10)" = fit.stan.ms10,
  "BS (df = 5)" = fit.stan.bs5,
  "BS (df = 10)" = fit.stan.bs10,
  "PW (df = 5)" = fit.stan.pw5,
  "PW (df = 10)" = fit.stan.pw10
)

# Comparison of estimated baseline hazards
require("purrr")
plots <- map(fits_stan, plot)
bayesplot_grid(
  plots = plots,
  ylim = c(0, 0.012),
  titles = names(fits_stan),
  grid_args = list(ncol = 3))

```



#posterior distribution

```
nd <- data.frame(treatment = c("Treatment", "Control"))
head(nd)
```

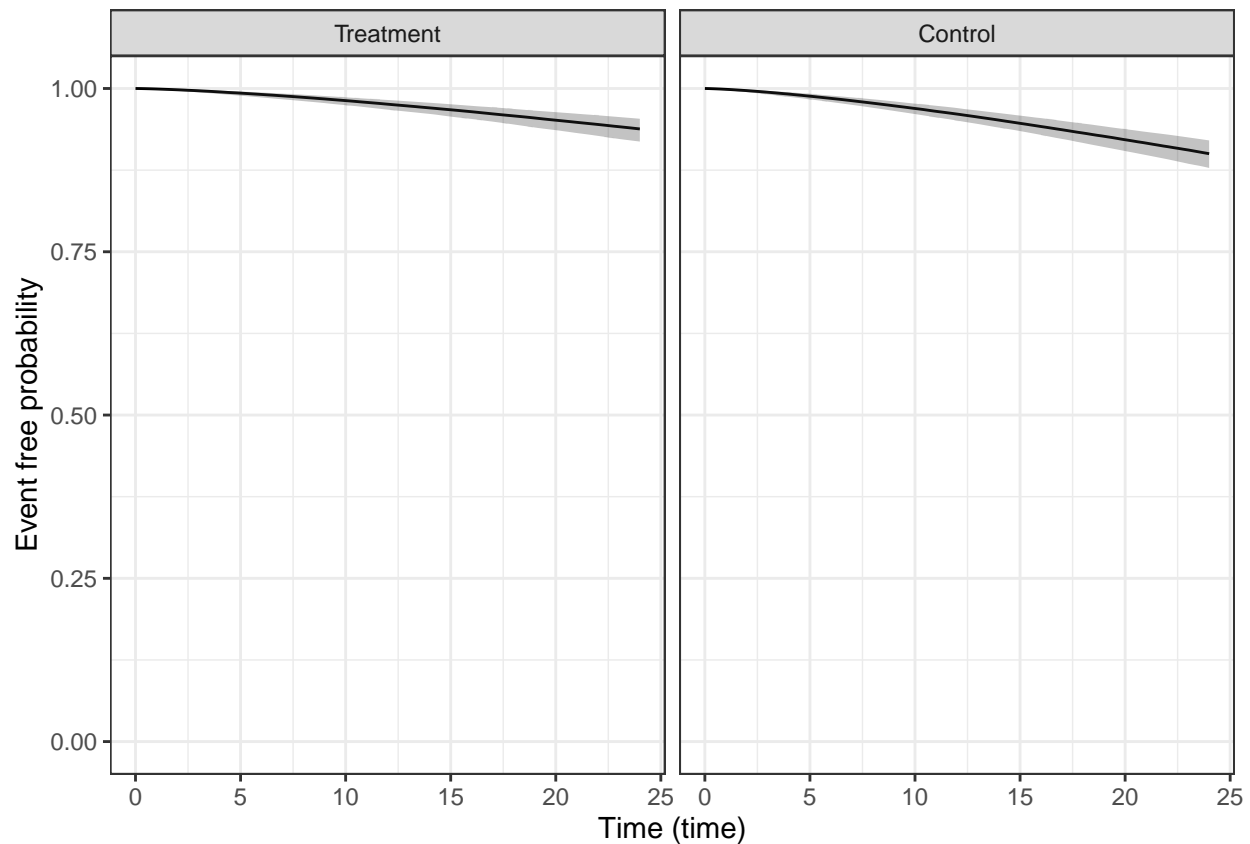
```
## treatment
## 1 Treatment
## 2 Control
```

```
ps <- posterior_survfit(fit.stan.weib,
newdata = nd,
times = 0,
extrapolate = TRUE)
head(ps)
```

```
## stan_surv predictions
## num. individuals: 2
## prediction type: event free probability
## standardised?: no
## conditional?: no
##
## id cond_time time median ci_lb ci_ub
## 1 1 NA 0.0000 1.0000 1.0000 1.0000
## 2 1 NA 0.2424 0.9999 0.9997 1.0000
## 3 1 NA 0.4848 0.9997 0.9993 0.9999
```

```
## 4 1      NA 0.7273 0.9995 0.9989 0.9998
## 5 1      NA 0.9697 0.9993 0.9985 0.9996
## 6 1      NA 1.2121 0.9990 0.9980 0.9995
```

```
panel_labels <- c('1' = "Treatment", '2' = "Control")
pps <- plot(psi) +
  facet_wrap(~ id, labeller = labeller(id = panel_labels))
pps
```

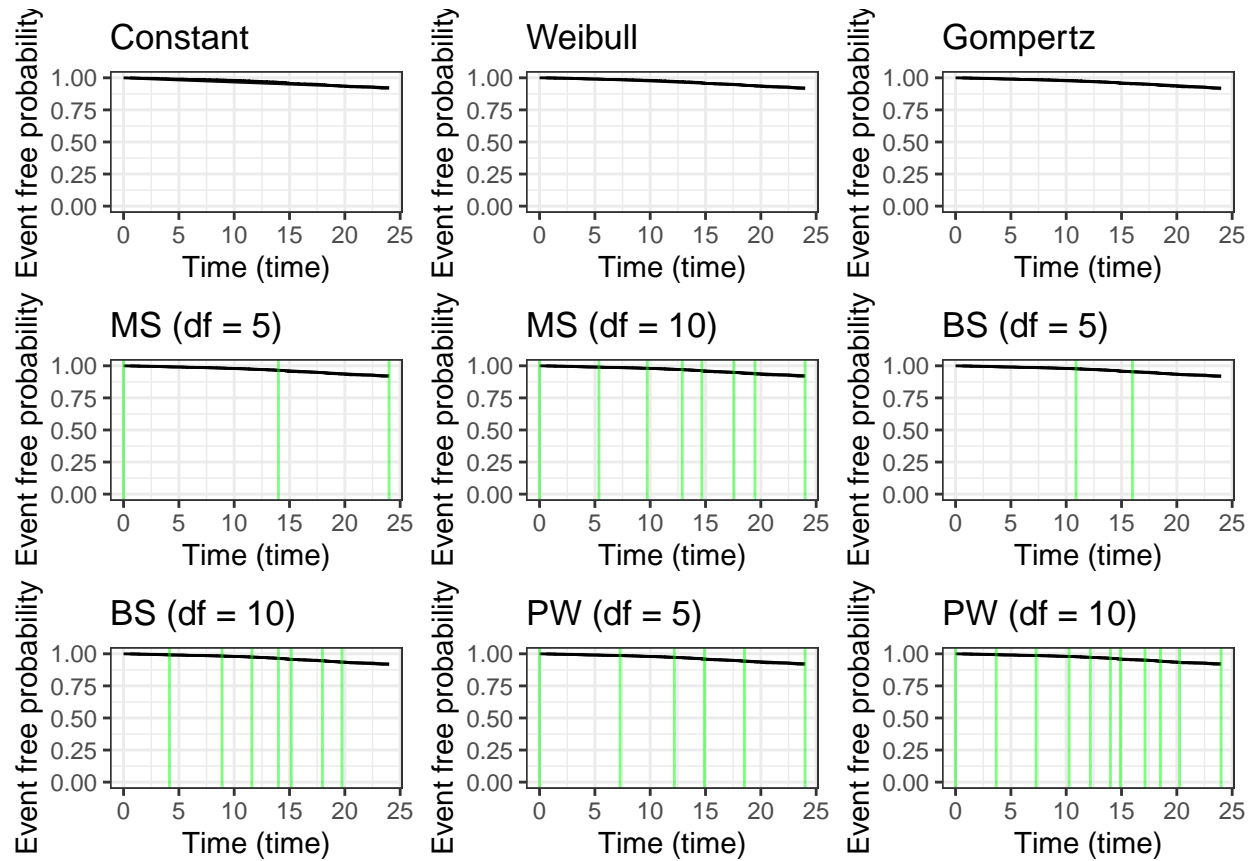


Assessing fit of the survival function

```
# define helper function to add knot locations
add_knots <- function(x) {
  knots <- x$basehaz$knots
  if (is.null(knots))
    return(NULL)
  geom_vline(xintercept = knots, color = "green", alpha = 0.5)
}

# generate the 'ps_check' plots
plots <- map(fits_stan, ~ (ps_check(.) + add_knots(.) + ggplot2::ylim(0.5,1)))
# combine the plots
bayesplot_grid(
  plots = plots,
```

```
titles = names(fits_stan),
grid_args = list(ncol = 3))
```



```
test <- posterior_survfit(fit.stan.weib, draw=1)
```

Comparison using leave-one-out cross validation

```
loos <- map(fits_stan, loo)
loo_compare(loos)
```

```
##           elpd_diff se_diff
## Weibull         0.0      0.0
## Gompertz        -0.2      1.4
## MS (df = 5)     -0.4      1.8
## PW (df = 5)     -1.3      2.4
## BS (df = 5)     -1.7      2.4
## MS (df = 10)    -2.3      2.4
## PW (df = 10)    -2.9      3.4
## BS (df = 10)    -6.0      2.9
## Constant       -6.9      3.4
```


exponential Hierchical

```
CHAINS <- 4
CORES <- 2
ITER <- 2000
SEED <- 1234

mod_randint_HC <- stan_surv(formula = Surv(month_next_preg,event_next_preg_24) ~treatment + (1 | area_b
seed = SEED,iter = ITER)
```

```
## Warning: There were 1 divergent transitions after warmup. See
## https://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
## to find out why this is a problem and how to eliminate them.
```

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

```
mod_randint_HC <- update(mod_randint_HC ,
prior_intercept = normal(0, 1),
prior = normal(0, 0.5))
summary(mod_randint_HC)
```

```
##
## Model Info:
##
## function:      stan_surv
## baseline hazard: exponential
## formula:      Surv(month_next_preg, event_next_preg_24) ~ treatment + (1 |
## area_bl)
## algorithm:     sampling
## sample:        4000 (posterior sample size)
## priors:        see help('prior_summary')
## observations:  1772
## events:        120 (6.8%)
## right censored: 1652 (93.2%)
## delayed entry: no
##
## Estimates:
##
```

| | mean | sd | 10% | 50% | 90% |
|---|------|-----|------|------|------|
| ## (Intercept) | -5.5 | 0.2 | -5.8 | -5.5 | -5.4 |
| ## treatment | -0.5 | 0.2 | -0.7 | -0.5 | -0.3 |
| ## b[(Intercept) area_bl:7] | -0.1 | 0.2 | -0.4 | -0.1 | 0.1 |
| ## b[(Intercept) area_bl:8] | -0.1 | 0.2 | -0.3 | 0.0 | 0.2 |
| ## b[(Intercept) area_bl:23] | 0.0 | 0.2 | -0.2 | 0.0 | 0.3 |
| ## b[(Intercept) area_bl:24] | 0.1 | 0.3 | -0.2 | 0.0 | 0.4 |
| ## b[(Intercept) area_bl:50] | 0.0 | 0.2 | -0.2 | 0.0 | 0.2 |
| ## b[(Intercept) area_bl:51] | -0.1 | 0.2 | -0.4 | -0.1 | 0.1 |
| ## b[(Intercept) area_bl:56] | 0.2 | 0.2 | 0.0 | 0.2 | 0.5 |
| ## Sigma[area_bl:(Intercept),(Intercept)] | 0.1 | 0.1 | 0.0 | 0.0 | 0.2 |

```
##
## MCMC diagnostics
##
```

| | mcse | Rhat | n_eff |
|--|------|------|-------|
|--|------|------|-------|

```
## (Intercept)                0.0  1.0  2085
## treatment                  0.0  1.0  4783
## b[(Intercept) area_bl:7]   0.0  1.0  3017
## b[(Intercept) area_bl:8]   0.0  1.0  3501
## b[(Intercept) area_bl:23]  0.0  1.0  4235
## b[(Intercept) area_bl:24]  0.0  1.0  3416
## b[(Intercept) area_bl:50]  0.0  1.0  2265
## b[(Intercept) area_bl:51]  0.0  1.0  3006
## b[(Intercept) area_bl:56]  0.0  1.0  1795
## Sigma[area_bl:(Intercept),(Intercept)] 0.0  1.0  2329
## log-posterior              0.1  1.0  1219
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

The model contains a baseline covariate for treatment (0 or 1) as well as a area-specific intercept to allow for correlation in the event times for patients from the same areas. We've called the model object `mod_randint` to denote the fact that it includes a area-specific (random) intercept. Let's examine the parameter estimates from the model:

```
print(mod_randint_HC, digits=2)

## stan_surv
## baseline hazard: exponential
## formula:          Surv(month_next_preg, event_next_preg_24) ~ treatment + (1 |
##   area_bl)
## observations:      1772
## events:            120 (6.8%)
## right censored:    1652 (93.2%)
## delayed entry:     no
## -----
##               Median MAD_SD exp(Median)
## (Intercept) -5.54   0.15      NA
## treatment   -0.49   0.18   0.61
##
## Error terms:
## Groups Name      Std.Dev.
## area_bl (Intercept) 0.305
## Num. levels: area_bl 7
##
## -----
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

We see that the estimated log hazard ratio for treatment $\hat{\beta}_{trt} = -0.49$ is about the same as the “true” log hazard ratio used in the simple model $\hat{\beta}_{trt} = -0.50$). The estimated baseline hazard rate is $\exp(-5.54) = 0.0039$, which is almost equal to the baseline hazard rate used in simple model.

```
mod_fixed <- update(mod_randint_HC, formula = Surv(month_next_preg,event_next_preg_24) ~treatment)
loo_fixed <- loo(mod_fixed)
loo_randint <- loo(mod_randint_HC)
loo_compare(loo_fixed, loo_randint)

##               elpd_diff se_diff
```

```
## mod_randint_HC 0.0      0.0
## mod_fixed      -0.2     1.4
```

weibull Hierchical

```
CHAINS <- 4
CORES <- 2
ITER <- 2000
SEED <- 1234

mod_randint_HC_weibull <- stan_surv(formula = Surv(month_next_preg,event_next_preg_24) ~treatment + (1
seed = SEED,iter = ITER)
mod_randint_HC_weibull <- update(mod_randint_HC_weibull ,
prior_intercept = normal(0, 1),
prior = normal(0, 0.5))
summary(mod_randint_HC_weibull)
```

```
##
## Model Info:
##
## function:      stan_surv
## baseline hazard: weibull
## formula:      Surv(month_next_preg, event_next_preg_24) ~ treatment + (1 |
##   area_bl)
## algorithm:     sampling
## sample:        4000 (posterior sample size)
## priors:        see help('prior_summary')
## observations:  1772
## events:        120 (6.8%)
## right censored: 1652 (93.2%)
## delayed entry: no
##
## Estimates:
##               mean    sd  10%   50%   90%
## (Intercept)   -6.7    0.4 -7.2  -6.7  -6.2
## treatment     -0.5    0.2 -0.7  -0.5  -0.3
## weibull-shape   1.4    0.1  1.3   1.4   1.5
## b[(Intercept) area_bl:7]   -0.1    0.2 -0.5  -0.1   0.1
## b[(Intercept) area_bl:8]   -0.1    0.2 -0.4   0.0   0.2
## b[(Intercept) area_bl:23]    0.0    0.2 -0.3   0.0   0.3
## b[(Intercept) area_bl:24]    0.1    0.3 -0.2   0.0   0.4
## b[(Intercept) area_bl:50]    0.0    0.2 -0.2   0.0   0.2
## b[(Intercept) area_bl:51]   -0.2    0.2 -0.5  -0.1   0.1
## b[(Intercept) area_bl:56]    0.2    0.2  0.0   0.2   0.4
## Sigma[area_bl:(Intercept),(Intercept)] 0.1    0.2  0.0   0.1   0.2
##
## MCMC diagnostics
##               mcse Rhat n_eff
## (Intercept)   0.0  1.0  3069
## treatment     0.0  1.0  4728
## weibull-shape 0.0  1.0  3272
```

```
## b[(Intercept) area_bl:7] 0.0 1.0 3326
## b[(Intercept) area_bl:8] 0.0 1.0 4086
## b[(Intercept) area_bl:23] 0.0 1.0 4392
## b[(Intercept) area_bl:24] 0.0 1.0 4722
## b[(Intercept) area_bl:50] 0.0 1.0 4037
## b[(Intercept) area_bl:51] 0.0 1.0 2898
## b[(Intercept) area_bl:56] 0.0 1.0 3093
## Sigma[area_bl:(Intercept),(Intercept)] 0.0 1.0 2041
## log-posterior 0.1 1.0 1213
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

```
print(mod_randint_HC_weibull, digits=2)
```

```
## stan_surv
## baseline hazard: weibull
## formula:      Surv(month_next_preg, event_next_preg_24) ~ treatment + (1 |
##   area_bl)
## observations: 1772
## events:      120 (6.8%)
## right censored: 1652 (93.2%)
## delayed entry: no
## -----
##               Median MAD_SD exp(Median)
## (Intercept) -6.69 0.38 NA
## treatment -0.49 0.17 0.61
## weibull-shape 1.38 0.11 NA
##
## Error terms:
## Groups Name Std.Dev.
## area_bl (Intercept) 0.319
## Num. levels: area_bl 7
##
## -----
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

```
mod_fixed_wei <- update(mod_randint_HC_weibull, basehaz = "weibull", formula = Surv(month_next_preg, event_next_preg_24) ~ treatment + (1 | area_bl))
loo_fixed_wei <- loo(mod_fixed_wei)
loo_randint_HC_weibull <- loo(mod_randint_HC_weibull)
loo_compare(loo_fixed_wei, loo_randint_HC_weibull)
```

```
##               elpd_diff se_diff
## mod_randint_HC_weibull 0.0 0.0
## mod_fixed_wei -0.2 1.4
```

gompertz Hierchical

```
CHAINS <- 4
CORES <- 2
```

```

ITER <- 2000
SEED <- 1234

mod_randint_HC_gom <- stan_surv(formula = Surv(month_next_preg,event_next_preg_24) ~treatment + (1 | are
seed = SEED,iter = ITER)
mod_randint_HC_gom <- update(mod_randint_HC_gom ,
prior_intercept = normal(0, 1),
prior = normal(0, 0.5))
summary(mod_randint_HC_gom)

```

```

##
## Model Info:
##
## function:      stan_surv
## baseline hazard: gompertz
## formula:      Surv(month_next_preg, event_next_preg_24) ~ treatment + (1 |
## area_bl)
## algorithm:     sampling
## sample:        4000 (posterior sample size)
## priors:        see help('prior_summary')
## observations:   1772
## events:        120 (6.8%)
## right censored: 1652 (93.2%)
## delayed entry: no
##
## Estimates:
##
##              mean    sd  10%   50%   90%
## (Intercept)    -6.2   0.2  -6.5  -6.1  -5.8
## treatment      -0.5   0.2  -0.7  -0.5  -0.3
## gompertz-scale   0.1   0.0   0.0   0.1   0.1
## b[(Intercept) area_bl:7]    -0.1   0.2  -0.4  -0.1   0.1
## b[(Intercept) area_bl:8]    -0.1   0.2  -0.4   0.0   0.1
## b[(Intercept) area_bl:23]    0.0   0.2  -0.2   0.0   0.3
## b[(Intercept) area_bl:24]    0.1   0.3  -0.2   0.0   0.4
## b[(Intercept) area_bl:50]    0.0   0.2  -0.2   0.0   0.2
## b[(Intercept) area_bl:51]   -0.2   0.2  -0.5  -0.1   0.1
## b[(Intercept) area_bl:56]    0.2   0.2   0.0   0.2   0.5
## Sigma[area_bl:(Intercept),(Intercept)] 0.1   0.2   0.0   0.0   0.2
##
## MCMC diagnostics
##
##              mcse Rhat n_eff
## (Intercept)    0.0  1.0  2400
## treatment      0.0  1.0  4764
## gompertz-scale  0.0  1.0  3084
## b[(Intercept) area_bl:7]    0.0  1.0  3380
## b[(Intercept) area_bl:8]    0.0  1.0  3682
## b[(Intercept) area_bl:23]   0.0  1.0  4174
## b[(Intercept) area_bl:24]   0.0  1.0  4132
## b[(Intercept) area_bl:50]   0.0  1.0  2982
## b[(Intercept) area_bl:51]   0.0  1.0  2921
## b[(Intercept) area_bl:56]   0.0  1.0  2420
## Sigma[area_bl:(Intercept),(Intercept)] 0.0  1.0  2628
## log-posterior    0.1  1.0  1133

```

```
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

```
print(mod_randint_HC_gom, digits=2)
```

```
## stan_surv
## baseline hazard: gompertz
## formula:      Surv(month_next_preg, event_next_preg_24) ~ treatment + (1 |
##   area_bl)
## observations: 1772
## events:      120 (6.8%)
## right censored: 1652 (93.2%)
## delayed entry: no
## -----
##               Median MAD_SD exp(Median)
## (Intercept)  -6.15  0.23    NA
## treatment    -0.50  0.18  0.61
## gompertz-scale 0.05  0.01    NA
##
## Error terms:
## Groups Name      Std.Dev.
## area_bl (Intercept) 0.313
## Num. levels: area_bl 7
##
## -----
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

```
mod_fixed_gom <- update(mod_randint_HC_gom, formula = Surv(month_next_preg,event_next_preg_24) ~treatment
loo_fixed_gom <- loo(mod_fixed_gom)
loo_randint_HC_gom <- loo(mod_randint_HC_gom)
loo_compare(loo_fixed, loo_randint,loo_fixed_wei, loo_randint_HC_weibull,loo_fixed_gom, loo_randint_HC_gom)
```

```
##               elpd_diff se_diff
## mod_randint_HC_weibull  0.0      0.0
## mod_fixed_wei          -0.2      1.4
## mod_randint_HC_gom     -0.3      1.4
## mod_fixed_gom          -0.5      2.0
## mod_randint_HC         -6.9      3.4
## mod_fixed              -7.1      3.7
```

```
#adjusted hierarchical model
```

```
aj_HC_weibull <- stan_surv(formula = Surv(month_next_preg,event_next_preg_24) ~treatment+age_group_bl+e
seed = SEED,iter = ITER)
aj_HC_weibull <- update(aj_HC_weibull ,
prior_intercept = normal(0, 1),
prior = normal(0, 0.5))
summary(aj_HC_weibull)
```

```
##
## Model Info:
```

```

##
## function:      stan_surv
## baseline hazard: weibull
## formula:      Surv(month_next_preg, event_next_preg_24) ~ treatment + age_group_bl +
##               edu_primary_bl + ever_use_bl + religion_r_bl + ethnicity_r_bl +
##               (1 | area_bl)
## algorithm:     sampling
## sample:        4000 (posterior sample size)
## priors:        see help('prior_summary')
## observations:  1772
## events:        120 (6.8%)
## right censored: 1652 (93.2%)
## delayed entry: no
##
## Estimates:
##               mean    sd   10%   50%   90%
## (Intercept)    -6.4    0.5  -7.0  -6.4  -5.8
## treatment       -0.5    0.2  -0.7  -0.5  -0.2
## age_group_bl    -0.1    0.1  -0.2  -0.1   0.1
## edu_primary_bl   0.0    0.2  -0.2   0.0   0.2
## ever_use_bl     -0.2    0.2  -0.5  -0.2   0.0
## religion_r_bl   -0.1    0.2  -0.4  -0.1   0.2
## ethnicity_r_bl   0.0    0.2  -0.2   0.0   0.2
## weibull-shape    1.4    0.1   1.3   1.4   1.5
## b[(Intercept) area_bl:7] -0.1    0.2  -0.5  -0.1   0.1
## b[(Intercept) area_bl:8] -0.1    0.2  -0.4   0.0   0.2
## b[(Intercept) area_bl:23] 0.0    0.2  -0.3   0.0   0.3
## b[(Intercept) area_bl:24] 0.1    0.3  -0.2   0.0   0.4
## b[(Intercept) area_bl:50] 0.0    0.2  -0.2   0.0   0.2
## b[(Intercept) area_bl:51] -0.2    0.2  -0.5  -0.1   0.1
## b[(Intercept) area_bl:56] 0.2    0.2   0.0   0.2   0.4
## Sigma[area_bl:(Intercept),(Intercept)] 0.1    0.2   0.0   0.0   0.2
##
## MCMC diagnostics
##               mcse Rhat n_eff
## (Intercept)    0.0  1.0  3030
## treatment       0.0  1.0  5216
## age_group_bl    0.0  1.0  5795
## edu_primary_bl   0.0  1.0  6253
## ever_use_bl     0.0  1.0  5671
## religion_r_bl   0.0  1.0  6394
## ethnicity_r_bl   0.0  1.0  5475
## weibull-shape    0.0  1.0  2740
## b[(Intercept) area_bl:7] 0.0  1.0  3364
## b[(Intercept) area_bl:8] 0.0  1.0  3989
## b[(Intercept) area_bl:23] 0.0  1.0  4834
## b[(Intercept) area_bl:24] 0.0  1.0  4628
## b[(Intercept) area_bl:50] 0.0  1.0  3337
## b[(Intercept) area_bl:51] 0.0  1.0  3026
## b[(Intercept) area_bl:56] 0.0  1.0  2695
## Sigma[area_bl:(Intercept),(Intercept)] 0.0  1.0  2705
## log-posterior    0.1  1.0  1152
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample

```

```
print(aj_HC_weibull, decimal=3)
```

```
## stan_surv
## baseline hazard: weibull
## formula:      Surv(month_next_preg, event_next_preg_24) ~ treatment + age_group_bl +
##               edu_primary_bl + ever_use_bl + religion_r_bl + ethnicity_r_bl +
##               (1 | area_bl)
## observations: 1772
## events:      120 (6.8%)
## right censored: 1652 (93.2%)
## delayed entry: no
## -----
##               Median MAD_SD exp(Median)
## (Intercept)   -6.4    0.5    NA
## treatment      -0.5    0.2    0.6
## age_group_bl   -0.1    0.1    0.9
## edu_primary_bl  0.0    0.2    1.0
## ever_use_bl    -0.2    0.2    0.8
## religion_r_bl  -0.1    0.2    0.9
## ethnicity_r_bl  0.0    0.2    1.0
## weibull-shape  1.4    0.1    NA
##
## Error terms:
## Groups Name      Std.Dev.
## area_bl (Intercept) 0.32
## Num. levels: area_bl 7
##
## -----
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

```
loo_randint_HC_weibull <- loo(mod_randint_HC_weibull)
loo_aj_HC_weibull <- loo(aj_HC_weibull)
loo_compare(loo_randint_HC_weibull, loo_aj_HC_weibull)
```

```
##               elpd_diff se_diff
## mod_randint_HC_weibull  0.0      0.0
## aj_HC_weibull          -3.1      1.5
```

```
aj_HC_gompertz <- stan_surv(formula = Surv(month_next_preg, event_next_preg_24) ~ treatment + age_group_bl +
seed = SEED, iter = ITER)
aj_HC_weibull <- update(aj_HC_gompertz,
prior_intercept = normal(0, 1),
prior = normal(0, 0.5))
summary(aj_HC_gompertz)
```

```
##
## Model Info:
##
## function:      stan_surv
## baseline hazard: gompertz
```



```

## formula:      Surv(month_next_preg, event_next_preg_24) ~ treatment + age_group_bl +
##      edu_primary_bl + ever_use_bl + religion_r_bl + ethnicity_r_bl +
##      (1 | area_bl)
## algorithm:    sampling
## sample:       4000 (posterior sample size)
## priors:       see help('prior_summary')
## observations: 1772
## events:       120 (6.8%)
## right censored: 1652 (93.2%)
## delayed entry: no
##
## Estimates:
##              mean    sd  10%   50%   90%
## (Intercept)   -5.8    0.4 -6.3  -5.8  -5.3
## treatment     -0.5    0.2 -0.8  -0.5  -0.3
## age_group_bl  -0.1    0.1 -0.2  -0.1   0.1
## edu_primary_bl  0.0    0.2 -0.2   0.0   0.2
## ever_use_bl   -0.3    0.2 -0.6  -0.3   0.0
## religion_r_bl  -0.2    0.2 -0.5  -0.2   0.1
## ethnicity_r_bl  0.0    0.2 -0.2   0.0   0.2
## gompertz-scale  0.1    0.0  0.0   0.1   0.1
## b[(Intercept) area_bl:7] -0.1  0.2 -0.4  -0.1   0.1
## b[(Intercept) area_bl:8] -0.1  0.2 -0.4   0.0   0.2
## b[(Intercept) area_bl:23]  0.0  0.2 -0.3   0.0   0.3
## b[(Intercept) area_bl:24]  0.1  0.3 -0.2   0.0   0.4
## b[(Intercept) area_bl:50]  0.0  0.2 -0.2   0.0   0.2
## b[(Intercept) area_bl:51] -0.2  0.2 -0.5  -0.1   0.1
## b[(Intercept) area_bl:56]  0.2  0.2  0.0   0.2   0.5
## Sigma[area_bl:(Intercept),(Intercept)] 0.1  0.2  0.0   0.1   0.2
##
## MCMC diagnostics
##              mcse Rhat n_eff
## (Intercept)   0.0  1.0  4387
## treatment     0.0  1.0  5505
## age_group_bl  0.0  1.0  5135
## edu_primary_bl 0.0  1.0  5236
## ever_use_bl   0.0  1.0  5611
## religion_r_bl 0.0  1.0  6136
## ethnicity_r_bl 0.0  1.0  5816
## gompertz-scale 0.0  1.0  3194
## b[(Intercept) area_bl:7] 0.0  1.0  3908
## b[(Intercept) area_bl:8] 0.0  1.0  4411
## b[(Intercept) area_bl:23] 0.0  1.0  5090
## b[(Intercept) area_bl:24] 0.0  1.0  3754
## b[(Intercept) area_bl:50] 0.0  1.0  3196
## b[(Intercept) area_bl:51] 0.0  1.0  3343
## b[(Intercept) area_bl:56] 0.0  1.0  2233
## Sigma[area_bl:(Intercept),(Intercept)] 0.0  1.0  2056
## log-posterior 0.1  1.0  1234
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
print(aj_HC_gompertz, decimal=3)

```

```

## stan_surv
## baseline hazard: gompertz
## formula:      Surv(month_next_preg, event_next_preg_24) ~ treatment + age_group_bl +
##      edu_primary_bl + ever_use_bl + religion_r_bl + ethnicity_r_bl +
##      (1 | area_bl)
## observations: 1772
## events:      120 (6.8%)
## right censored: 1652 (93.2%)
## delayed entry: no
## -----
##              Median MAD_SD exp(Median)
## (Intercept)  -5.8    0.4    NA
## treatment     -0.5    0.2    0.6
## age_group_bl  -0.1    0.1    0.9
## edu_primary_bl 0.0    0.2    1.0
## ever_use_bl   -0.3    0.2    0.8
## religion_r_bl -0.2    0.2    0.9
## ethnicity_r_bl 0.0    0.2    1.0
## gompertz-scale 0.1    0.0    NA
##
## Error terms:
## Groups Name      Std.Dev.
## area_bl (Intercept) 0.32
## Num. levels: area_bl 7
##
## -----
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg

loo_randint_HC_gompertz <- loo(mod_randint_HC_gom)
loo_aj_HC_gompertz <- loo(aj_HC_gompertz)
loo_compare(loo_randint_HC_weibull, loo_aj_HC_weibull, loo_randint_HC_gompertz, loo_aj_HC_gompertz)

##              elpd_diff se_diff
## mod_randint_HC_weibull 0.0      0.0
## mod_randint_HC_gom     -0.3     1.4
## aj_HC_weibull          -3.1     1.5
## aj_HC_gompertz         -4.0     2.1

```

As we can see in the output the adjusted model is worse