Introduction to TTE modeling: Workbook 5

Parametric TTE models

2023-07-11

Contents

Preliminaries for R examples

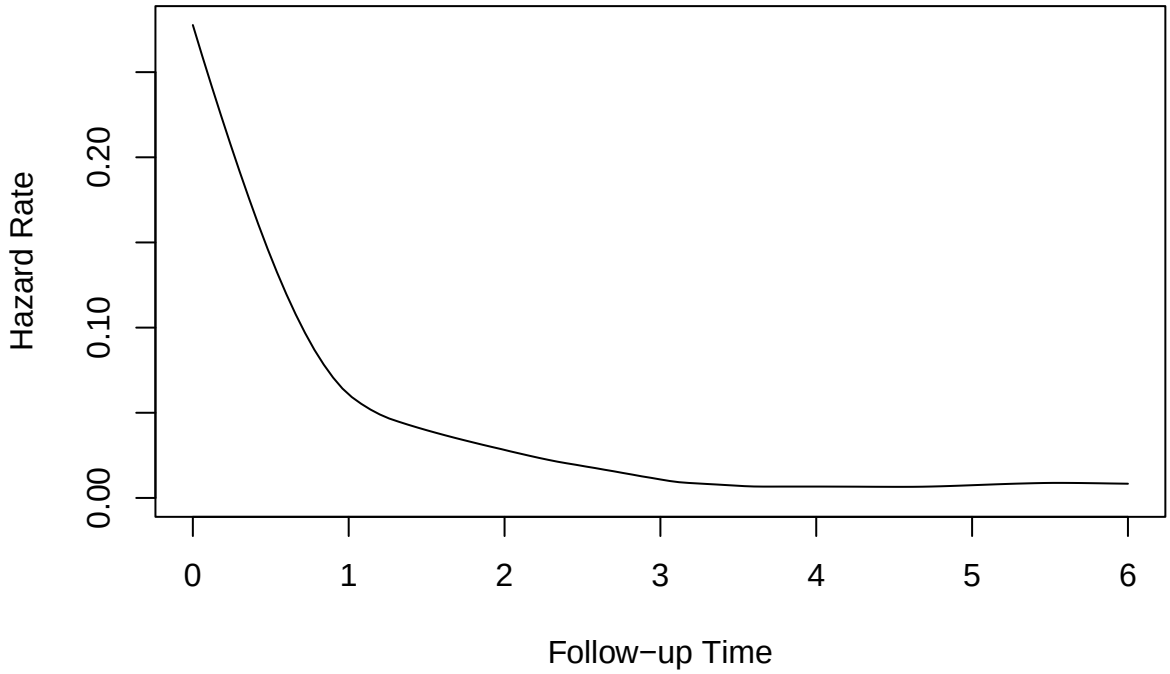
```
library(tidyverse)
library(stringr)
library(survival)
library(survminer)
library(texreg)
library(mgcv)
library(flexsurv)
library(muhaz)
library(Hmisc)
library(furrr)

theme_set(theme_bw())

plan(multisession, workers = 2)
```

```
load('../data/aedat.RDS')

aedat <-
  aedat %>%
  mutate(AETOXGR = factor(aedat$AETOXGR, 0:3, labels=c("None","Mild","Moderate","Severe")),
         ae_any = AETOXGR != 'None') %>%
  group_by(USUBJID) %>%
  # End of study for patients without a severe event
  mutate(TTE_SEVERE = case_when(
    STUDYID=="PROTA" ~ 2,
    STUDYID=="PROTB" ~ 6
  ),
  # Time of severe event for those that had one
  TTE_SEVERE = ifelse(AETOXGR=="Severe", TTE, TTE_SEVERE)
  )

# Both for EDA and for model-checking, it's generally helpful to have quartiles of exposure:
dat_use <-
  aedat %>% arrange(USUBJID, TTE_SEVERE) %>% slice(1) %>%
```

```
  group_by(PBO) %>%
  mutate(Quartile = ifelse(PBO == "PBO", "PBO",
                           paste0("Q", ntile(CAVGSS, n = 4)))) %>%
  ungroup() %>%
  mutate(rowid = 1:n())
```

Let's start by getting some idea about the shape of the hazard function. To do this, we can use the package `muhaz` to get a non-parametric estimate.

```
np_est = muhaz(times = dat_use$TTE_SEVERE,
               delta = dat_use$AE01,
               min.time=0,
               max.time=6)

plot(np_est)
```



Using all of the data, the hazard appears to be monotonic - high early and declining sharply. Based on this, we can rule out the exponential model. Which models might be good candidates to fit this shape?

Let's try the Weibull and Gompertz

```
fit01_weibull <- flexsurvreg(Surv(TTE_SEVERE, AE01) ~ CAVGSS,
                             data = dat_use,
                             dist = 'weibull')

fit01_gompertz <- flexsurvreg(Surv(TTE_SEVERE, AE01) ~ CAVGSS,
                             data = dat_use,
                             dist = 'gompertz')
```

Comparing the two models by AIC, it looks like there isn't much difference between the two models

```
AIC(fit01_weibull, fit01_gompertz)
```

```
.                df      AIC
. fit01_weibull   3 140.2161
. fit01_gompertz  3 152.0211
```

Let's compare based on VPCs. Unfortunately, there is no simple way to simulate from these models.

So, we'll write a short function to do this. The input will be a dataset with predicted survival function. The default number of simulations will be 100. For now, we'll assume only administrative censoring.

```
simulate_TTE <- function(.data, nsims=100) {
  times <- c(0,.data$time)
  surv <- c(1,.data$est)
  surv_fun <- approxfun(x=surv, y=times, yright=max(times), rule=2)
  u <- runif(nsims)
  sim_times <- surv_fun(u)
  delta <- sim_times < max(times)
  return(data.frame(time=sim_times, event=delta, sim=1:nsims))
}
```

Let's extract the predicted survival curve from the Weibull model for each subject.

```
summary_weibull <- summary(fit01_weibull,
                           newdata=dat_use,
                           ci=FALSE,
                           tidy=TRUE) %>%
  arrange(time) %>%
  mutate(rowid = rep(1:nrow(dat_use), times=length(unique(.$time)))) %>%
  arrange(rowid,time)
```

Now, we'll simulate data according to this model.

```
sims_weibull <- summary_weibull %>%
  select(rowid,time,est) %>%
  nest(data = c("time","est")) %>%
  mutate(sims = map(data, simulate_TTE)) %>%
  unnest(cols=sims) %>%
  select(-data) %>%
  left_join(dat_use)
```

Our summary statistic is the Kaplan-Meier estimator. So, we'll calculate the K-M in each group, in this case, exposure quartile.

```
vpc_stat <- function(.data, pred_times=NULL) {
  fit <- survfit(Surv(time,event)~Quartile, data=.data)
  if (is.null(pred_times)) {
    pred_times <- c(0,sort(fit$time))
  }
  preds = summary(fit, times=pred_times)

  data.frame(pred_times=pred_times, preds = preds$surv, group=preds$strata)
}
```

Apply the summary statistic to the original data

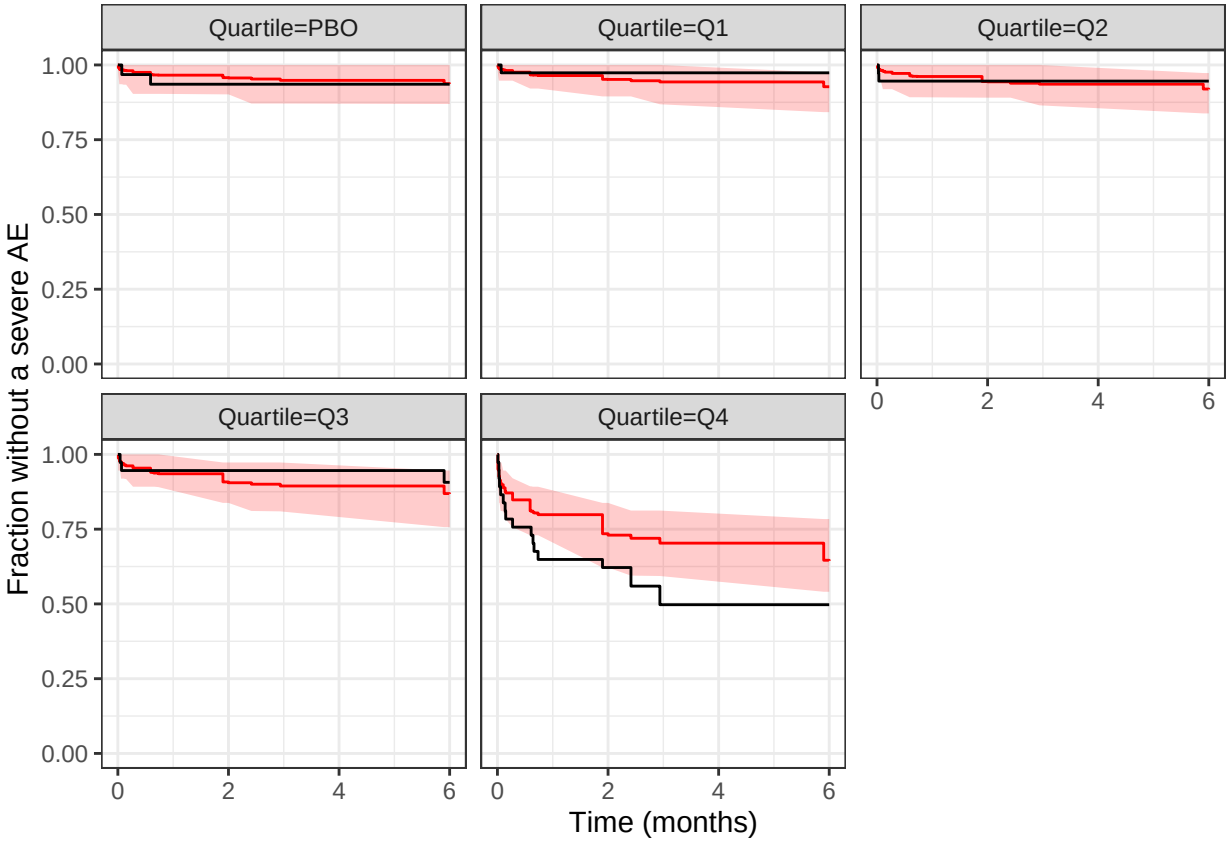
```
obs_surv = vpc_stat(dat_use %>% mutate(time=TTE_SEVERE, event=AE01))
```

Apply the summary statistic to each simulated dataset

```
sim_surv = sims_weibull %>%
  nest(data = -sim) %>%
  mutate(km_est = map(data, ~vpc_stat(., pred_times=sort(unique(obs_surv$pred_times))))) %>%
  select(-data) %>%
  unnest(cols=km_est)
```

Aggregate across simulations and plot

```
sim_surv %>% group_by(pred_times, group) %>%
  summarise(med=mean(preds),
            lcl = quantile(preds,probs = 0.05),
            ucl = quantile(preds, probs=0.95)) %>%
  ggplot(aes(x=pred_times)) +
  geom_step(aes(y=med), color='red') +
  geom_ribbon(aes(ymin=lcl, ymax=ucl), fill='red', alpha=0.2) +
  geom_step(data=obs_surv,aes(y=preds)) +
  facet_wrap(~group) +
  ylim(0,1) +
  labs(x='Time (months)', y='Fraction without a severe AE')
```



Exercise:

- How does this look? How does it compare to the Cox model with a linear effect of exposure?

Answer:

Like the Cox model predictions, the Weibull model with a linear effect of exposure predicts reasonably well at low exposures, but it underpredicts the risk at higher exposures.

- Make a similar VPC using the Gompertz model. Does this look any better?

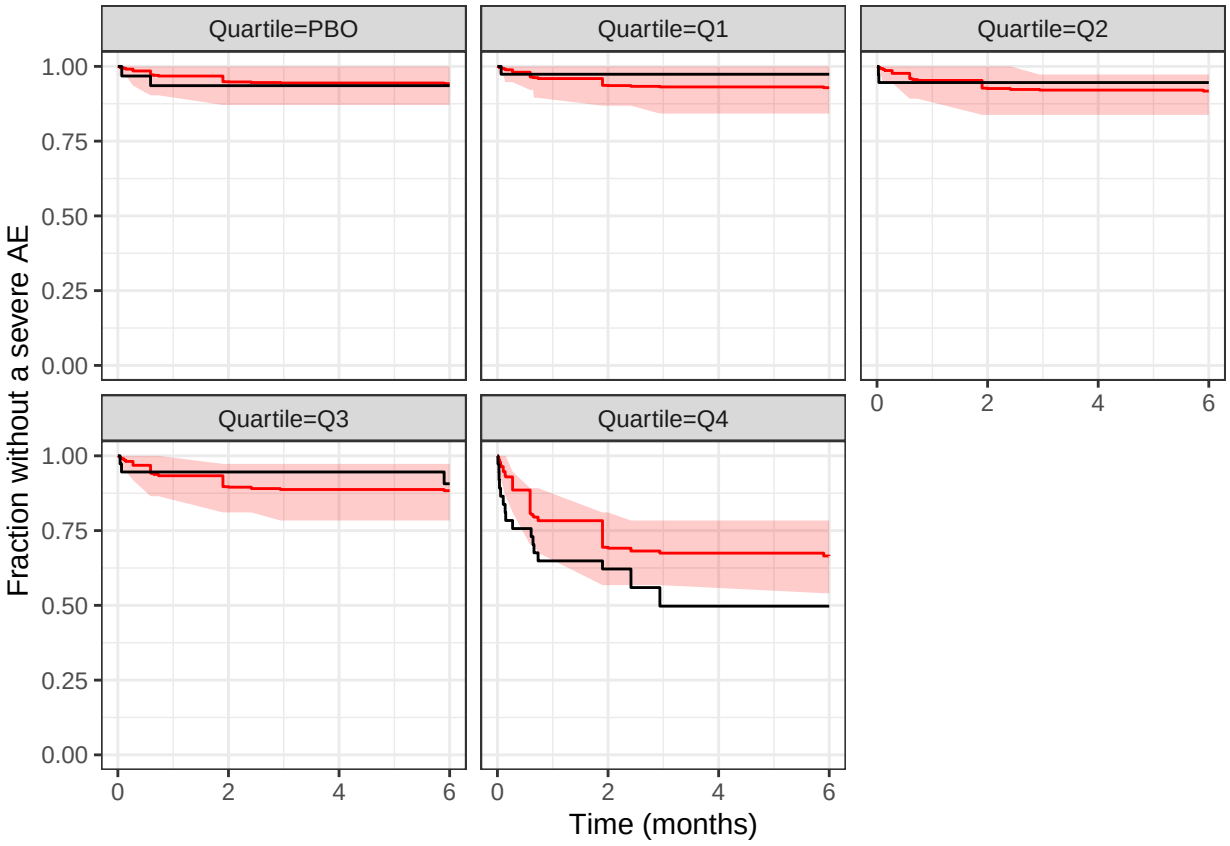
Answer:

Building on the code above, we write a function to perform the VPC steps. As an input, the function will take a survfit object and number of simulations

```
flexsurv_vpc <- function(.object, .data, .nsims=1000) {

  model_summary <- summary(.object,
                           newdata=.data,
                           ci=FALSE,
                           tidy=TRUE)
  nobs = nrow(.data)

  model_summary <- model_summary %>%
  arrange(time) %>%
  mutate(rowid = rep(1:nobs, times=length(unique(.$time)))) %>%
  arrange(rowid,time)

sims <- model_summary %>%
  select(rowid,time,est) %>%
  nest(data = c("time","est")) %>%
  mutate(sims = map(data, ~simulate_TTE(.x, nsims=.nsims))) %>%
  unnest(cols=sims) %>%
  select(-data) %>%
  left_join(dat_use)

sim_surv = sims %>%
  nest(data = -sim) %>%
  mutate(km_est = map(data, ~vpc_stat(., pred_times=sort(unique(obs_surv$pred_times))))) %>%
  select(-data) %>%
  unnest(cols=km_est)

}
```

```
sim_surv_gompertz <- flexsurv_vpc(fit01_gompertz, dat_use)

sim_surv_gompertz %>%
  group_by(pred_times, group) %>%
  summarise(med=mean(preds),
            lcl = quantile(preds,probs = 0.05),
            ucl = quantile(preds, probs=0.95)) %>%
  ggplot(aes(x=pred_times)) +
```

```
  geom_step(aes(y=med), color='red') +
  geom_ribbon(aes(ymin=lcl, ymax=ucl), fill='red', alpha=0.2) +
  geom_step(data=obs_surv,aes(y=preds)) +
  facet_wrap(~group) +
  ylim(0,1) +
  labs(x='Time (months)', y='Fraction without a severe AE')
```



- Try fitting a parametric model using a smoothing spline for the effect of CAVGSS and plotting the VPC. Is this better?

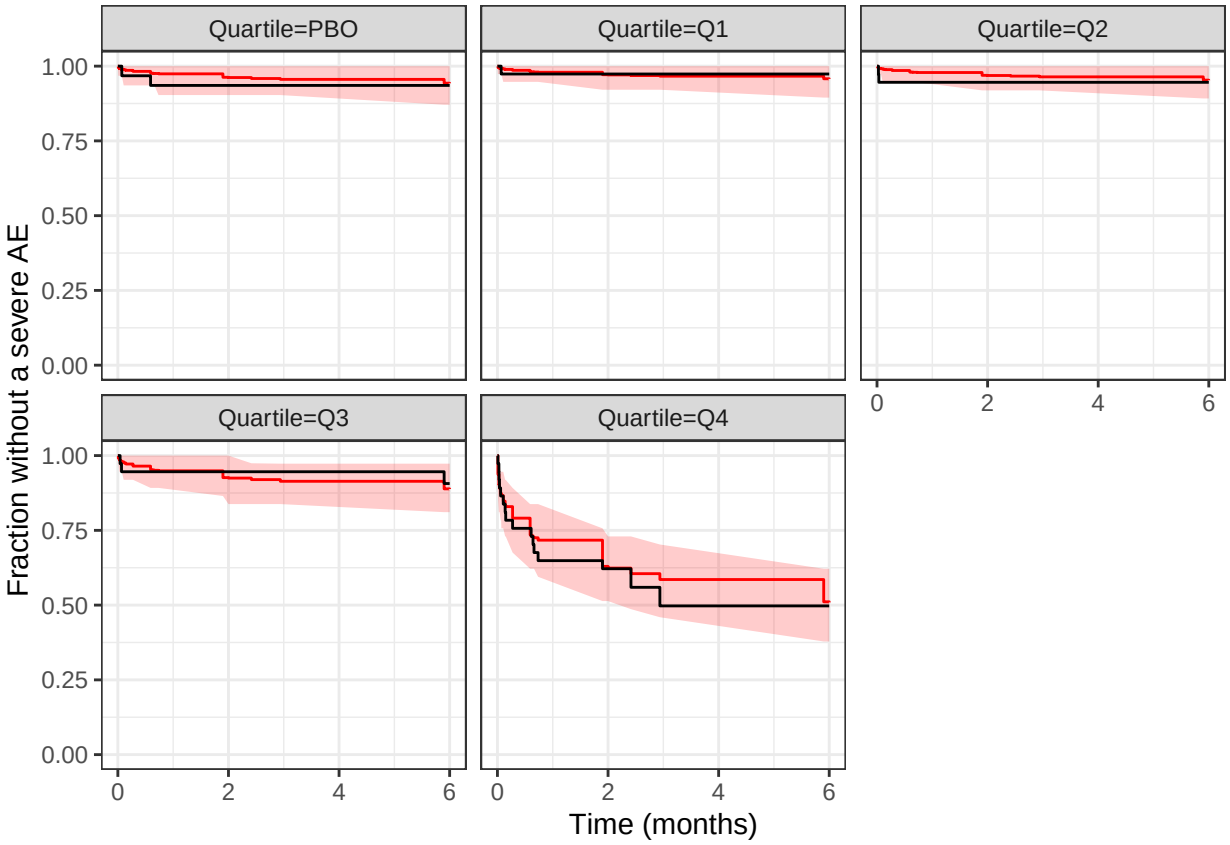
Answer:

The code for fitting the model and generating the VPC is below. Like the penalized spline fit under the Cox model, this model fits notably better than the Weibull model with a linear effect of exposure.

```
fit_smooth <- flexsurvreg(Surv(TTE_SEVERE, AE01) ~ pspline(CAVGSS),
                          data = dat_use,
                          dist = 'weibull')
```

```
sim_surv_weib_nonlin <- flexsurv_vpc(fit_smooth, dat_use)

sim_surv_weib_nonlin %>%
  group_by(pred_times, group) %>%
  summarise(med=mean(preds),
            lcl = quantile(preds,probs = 0.05),
```

```
          ucl = quantile(preds, probs=0.95)) %>%
ggplot(aes(x=pred_times)) +
geom_step(aes(y=med), color='red') +
geom_ribbon(aes(ymin=lcl, ymax=ucl), fill='red', alpha=0.2) +
geom_step(data=obs_surv,aes(y=preds)) +
facet_wrap(~group) +
ylim(0,1) +
labs(x='Time (months)', y='Fraction without a severe AE')
```



KMMC by-hand

1. Write a function to calculate the running mean among subjects at risk

```
running_mean <- function(.localdata, .cov_names, .time_var, .time_grid) {
  .localdata[,'time_var'] <- .localdata[,.time_var]

 subsets = tibble(time = .time_grid) %>%
   mutate(subset = map(time, function(.x,.dtmp=.localdata) {
          .dtmp %>%
            filter(time_var >= .x) %>%
            select(all_of(.cov_names))
       }))

subsets %>%
  mutate(means = map(subset, ~.x %>% pivot_longer(cols=everything()) %>%
                       group_by(name) %>%
                       summarise(mean_cov=mean(value)))) %>%
  select(-subset) %>%
  unnest(means)
}
```

2. Calculate the grid of times at which to evaluate the means. It's recommended to use the observed event times.

```
times <- dat_use %>% filter(AE01==1) %>%
  distinct(TTE) %>%
  pull(TTE) %>%
  as.numeric() %>%
  sort()
```

3. Evaluate the covariate means (CAVGss and body weight) in the observed data. Since we want to stratify the plot by treatment group, we group by `PBO`.

```
obs_means <- dat_use %>%
  group_by(PBO) %>%
  nest() %>%
  mutate(obs_means = map(data, ~running_mean(.x, c('CAVGSS','BWT'), 'TTE_SEVERE',times))) %>%
  select(-data) %>%
  unnest(obs_means)
```

4. Evaluate the covariate means in each of the simulated trials. Note that we group by both `sim` and `PBO` to achieve this.

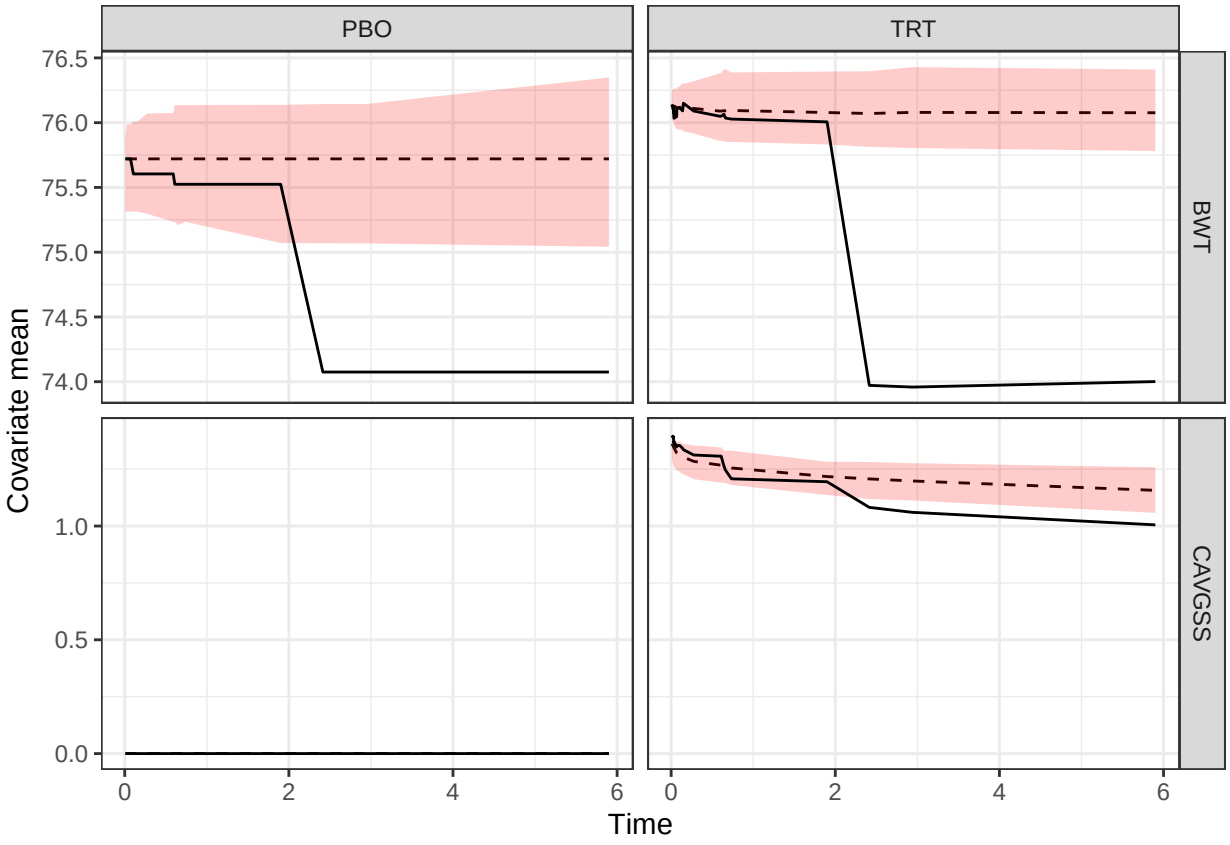
```
sim_means <- sims_weibull %>%
  group_by(sim, PBO) %>%
  nest() %>%
  mutate(sim_means = map(data, ~running_mean(.x, c('CAVGSS','BWT'),'time',times)))
```

5. Calculate summary statistics at each event time across simulated trials.

```
sim_stats <- sim_means %>%
  select(-data) %>%
  unnest(sim_means) %>%
  group_by(PBO,time,name) %>%
  summarise(med = median(mean_cov),
            qlo = quantile(mean_cov,.05),
            qhi = quantile(mean_cov,.95))
```

6. Make the plot

```
ggplot(sim_stats,
       aes(x=time, y=med)) +
  geom_line(linetype='dashed') +
  geom_ribbon(aes(ymin=qlo, ymax=qhi), fill='red', alpha=0.2) +
  geom_line(data=obs_means, aes(y=mean_cov) ) +
  facet_grid(name~PBO, scales = 'free_y') +
  labs(x='Time', y='Covariate mean')
```



For CAVGss, the observed data (solid line) shows that subjects in the risk set at later times to have lower exposures than the entire population (at the start of the study). The KMMC shows that the model captures a similar trend.