Workbook for Introduction to TTE modeling

Repeated time to event analysis in Stan

2022-08-31

Contents

Preliminaries for R examples

```
library(tidyverse)
library(stringr)
library(survival)
library(survminer)
library(texreg)
library(mgcv)
library(muhaz)
library(rstan)
library(bayesplot)
library(tidybayes)
library(survminer)

theme_set(theme_bw())
bayesplot::color_scheme_set("viridis")
```

Data exploration

The data are simulated from a PK-PD model for time to bleeding events [@Abrantes2020-yd].

- PK model for a full-length recombinant human FVIII product
    - Dosed at levels of 20 - 50 IU/kg
- Repeated time to bleeding event model
    - Gompertz TTE model with an inhibitory Emax effect of FVIII(t)

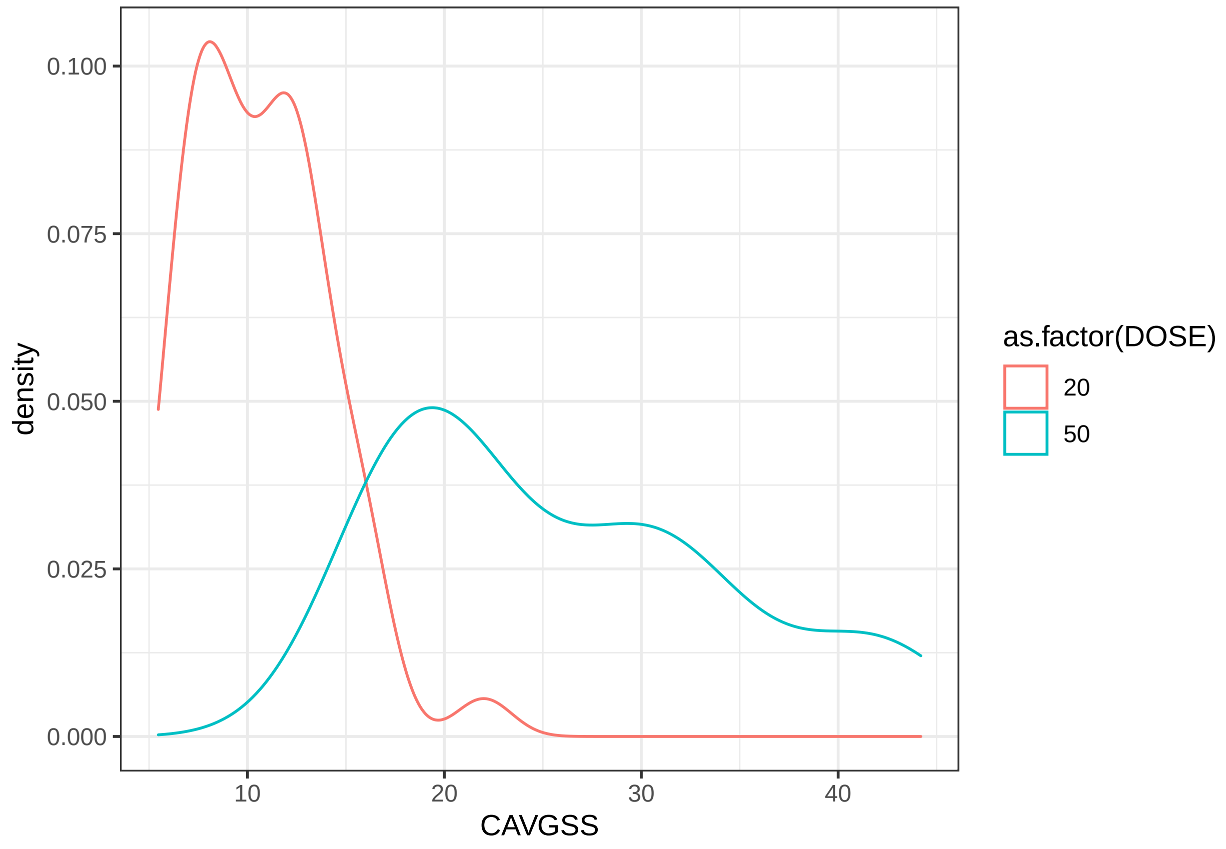
- The simulated data are from a hypothetical trial in 100 patients
  - Randomly allocated to doses of 20 and 50 IU/kg
  - Follow-up for 90 days
- We will investigate the relationship of steady-state Caverage on repeated time to bleeding
  - Predict effect on bleeding events at a 100 IU/kg dose
  - Not considering other potential side effects or personalized dosing

```
# Time to bleeding data
d <- read_csv('../data/derived/rtte_example.csv', na = c('.'))

d <- d %>%
  group_by(ID) %>%
  mutate(TIME = TIME/24,  # Convert from hours to days
         PREVTIME = lag(TIME, default = 0))

# Extract one record per subject.  The last record will indicate the number of bleeding events.
d_ind = d %>% group_by(ID) %>% slice_tail() %>% ungroup() %>% arrange(ID)
```

Exposures

To get a sense of the range of exposures, let's plot them.

```
d_ind %>% ggplot(aes(x=CAVGSS, group=DOSE, col=as.factor(DOSE))) + geom_density()
```



Numerically, it looks like there is minimal overlap between the FX VIII levels at the 20 and 50 IU/kg dose levels.

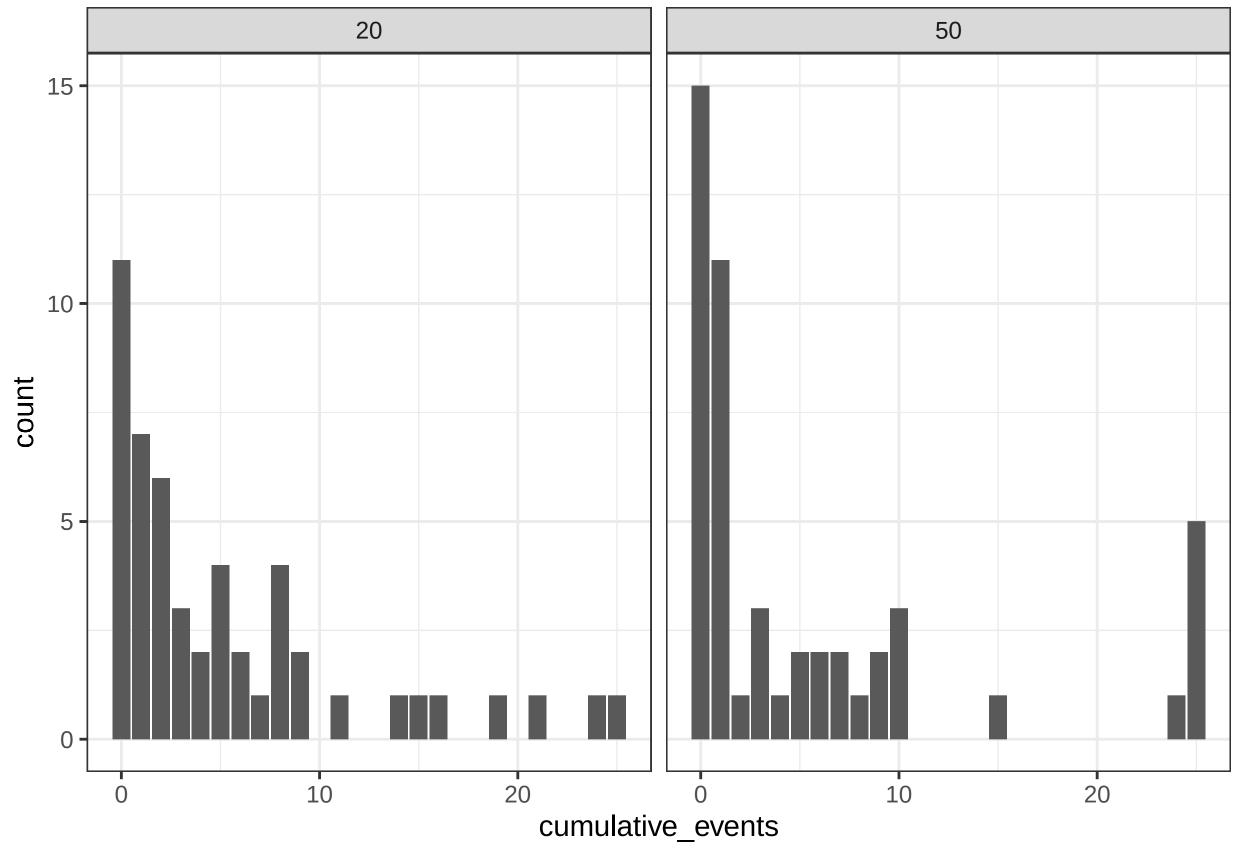
```
d_ind %>% group_by(DOSE) %>%
  summarise(q05 = quantile(CAVGSS,probs=0.05),
            median = quantile(CAVGSS,probs=0.5),
            q95 = quantile(CAVGSS,probs=0.95))
```

```
. # A tibble: 2 x 4
.    DOSE   q05 median   q95
.   <dbl> <dbl>  <dbl> <dbl>
. 1    20  6.39   10.3  16.1
. 2    50 16.5    23.2  42.2
```

Bleeding events

To understand how many events subjects experience, let's start with a barplot of the number of events stratified by dose. While the modal value is 0 events, most patients experience at least one bleeding event. There is a long tail, with some patients experiencing ovet 10 events.

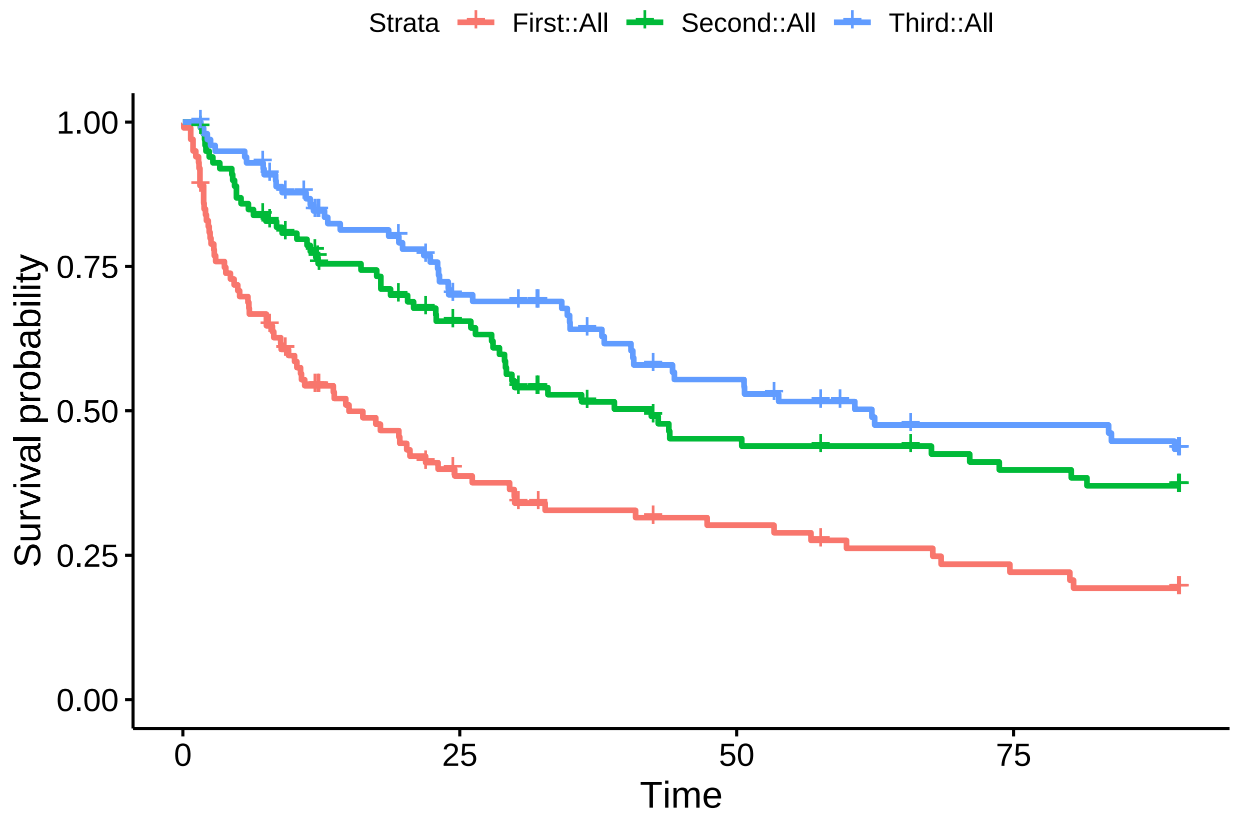
```
d_ind %>%
  ggplot(aes(x=cumulative_events)) +
  geom_bar() +
  facet_grid(~DOSE)
```



Next, we'll plot time to first, second, and third bleeding events. This is similar to Figure 1 in [@Abrantes2020-yd].

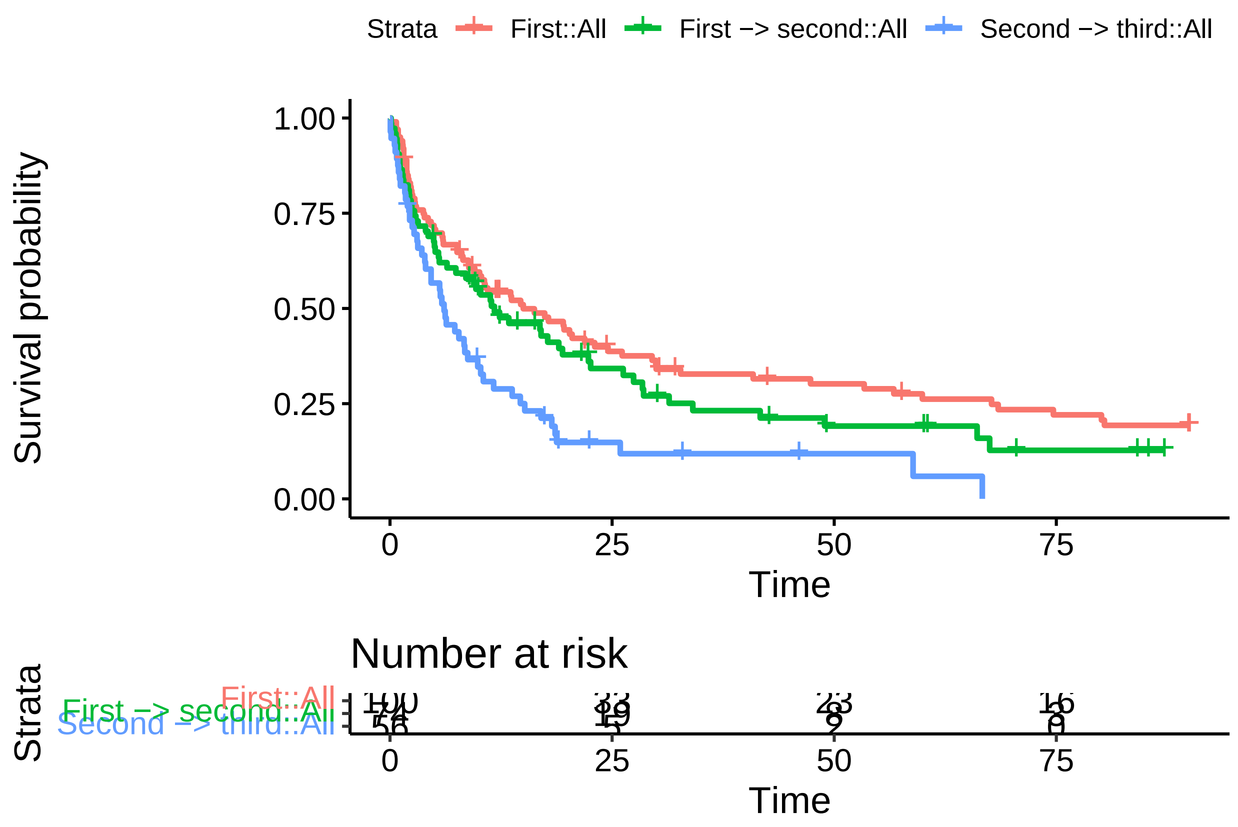
```
d_first = d %>% group_by(ID) %>% filter(TIME == min(TIME)) %>% ungroup()
```

```
d_second = d %>%
  group_by(ID) %>%
  filter((cumulative_events==2 & event==1) | (cumulative_events < 2 & TIME == max(TIME) & event==0)) %>%
  ungroup()

d_third = d %>%
  group_by(ID) %>%
  filter((cumulative_events==3 & event==1) | (cumulative_events < 3 & TIME == max(TIME) & event==0)) %>%
  ungroup()

fit1 <- survfit(Surv(TIME,event)~1, data = d_first)
fit2 <- survfit(Surv(TIME,event)~1, data = d_second)
fit3 <- survfit(Surv(TIME,event)~1, data = d_third)

ggsurvplot_combine(list(First=fit1, Second=fit2, Third=fit3))
```



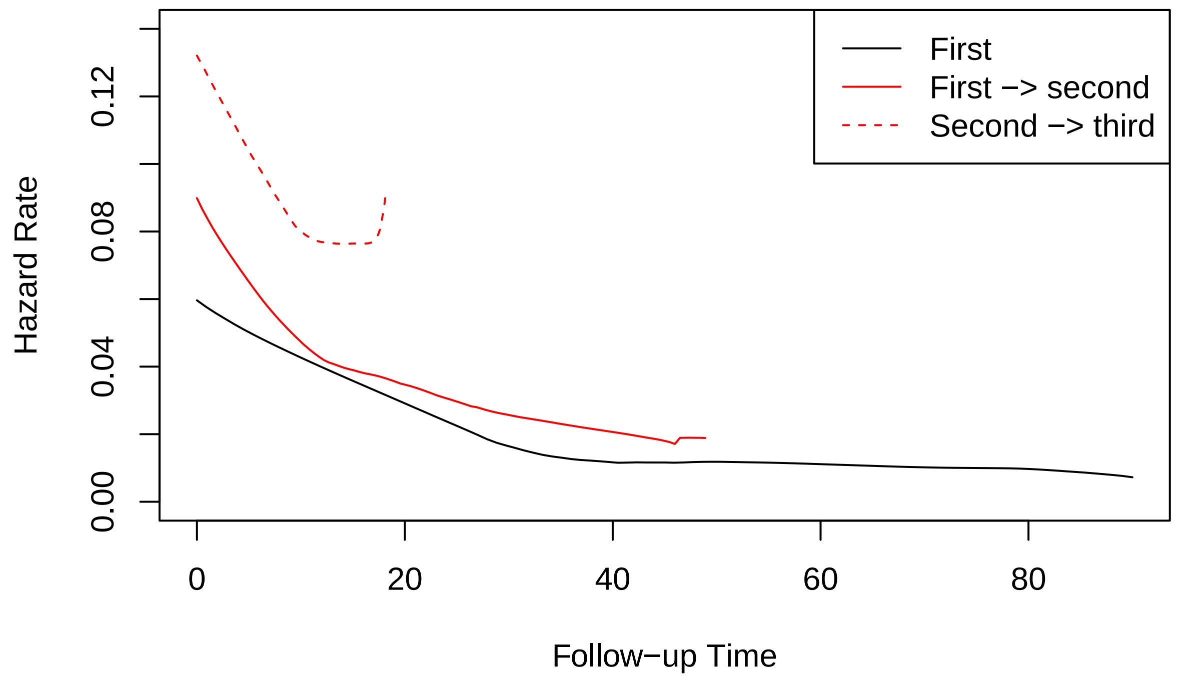
When we're modeling the hazard of repeated events, we don't usually model the marginal hazard. Instead, we're modeling the inter-event hazard. So, let's plot the time to the first event, the time between first and second events (among subjects with one or more events), and the time between second and third events (among subjects with two or more events).

```
d_first = d %>% group_by(ID) %>% filter(TIME == min(TIME)) %>% ungroup()

d_second = d %>%
  group_by(ID) %>%
  filter((cumulative_events==2 & event==1) | (cumulative_events == 1 & TIME == max(TIME) & event==0)) %
  ungroup()

d_third = d %>%
  group_by(ID) %>%
  filter((cumulative_events==3 & event==1) | (cumulative_events == 2 & TIME == max(TIME) & event==0)) %
  ungroup()

fit1 <- survfit(Surv(TIME,event)~1, data = d_first)
fit2 <- survfit(Surv(TIME-PREVTIME,event)~1, data = d_second)
fit3 <- survfit(Surv(TIME-PREVTIME,event)~1, data = d_third)

ggsurvplot_combine(list(First=fit1, 'First -> second'=fit2, 'Second -> third'=fit3), risk.table = TRUE)
```



Now, let's plot the non-parametric estimates of the hazards for these same events.

```
hazard1 = with(d_first, muhaz(TIME,event, min.time=0, max.time=90))
hazard2 = with(d_second, muhaz(TIME-PREVTIME, event, min.time = 0))
hazard3 = with(d_third, muhaz(TIME-PREVTIME, event, min.time = 0))

plot(hazard1, ylim=range(pretty(c(hazard1$haz.est, hazard2$haz.est,hazard3$haz.est))))
lines(hazard2, col='red')
lines(hazard3, col='red', lty=2)
legend(x='topright',
       legend = c('First','First -> second','Second -> third'),
       col=c('black','red','red'), lty=c(1,1,2))
```



The shape generally looks similar, with a shift upward for subsequent events. What do you think might be causing this upward shift? Hazard increasing with subsequent events? Subjects with higher hazard are likeliy to have more events and hence be in the 'second -> third' risk set?

Fitting an RTTE model

Given this shape for the hazard function, what basic hazard functions might be good places to investigate (or rule out)?

Both the Weibull ($h(t) = \alpha\gamma^{\alpha}t^{\alpha-1}$) and Gompertz ($h(t) = \alpha e^{\gamma t}$) can give us this shape.

Let's start with the Gompertz model to get our grounding with RTTE models, then we'll compare to the Weibull model.

First, we'll set-up the data for modeling.

```
stan_data = list(Nobs=nrow(d),
                 Nsubj = nrow(d_ind),
                 delta = d$event,
                 time = d$TIME,
                 prev_time = d$PREVTIME,
                 ID = d$ID,
                 CAVGSS = d$CAVGSS)
```

Base model: No IIV, no exposure

```r
fit_base_noiiv <- stan(file = '../model/stan/Day4/rtte_gompertz_noiiv.stan',
            data = stan_data,
            chains = 4,
            iter = 2000,
            cores = 4)
```

Check convergence of the Markov chains

```r
print(fit_base_noiiv, pars=c('log_alpha_pop','gamma'))
```

```
. Inference for Stan model: rtte_gompertz_noiiv.
. 4 chains, each with iter=2000; warmup=1000; thin=1;
. post-warmup draws per chain=1000, total post-warmup draws=4000.
.
.                 mean se_mean   sd  2.5%   25%   50%   75% 97.5% n_eff Rhat
. log_alpha_pop -2.24       0 0.07 -2.38 -2.29 -2.24 -2.19 -2.09  1144    1
. gamma         -0.80       0 0.15 -1.10 -0.90 -0.80 -0.70 -0.50  1155    1
.
. Samples were drawn using NUTS(diag_e) at Wed Aug 31 22:41:41 2022.
. For each parameter, n_eff is a crude measure of effective sample size,
. and Rhat is the potential scale reduction factor on split chains (at
. convergence, Rhat=1).
```

Let's simulate from this model to see if we get back a similar distribution of number of events and event times.

```r
sim_base = function(.ingredients, .max_time=90) {
  # Gompertz cumulative hazard
  alpha = exp(.ingredients$log_alpha_pop)
  gamma = .ingredients$gamma

  ev_times = numeric()
  t1 = 0
  while(t1 < .max_time) {
    u = runif(1)
    x = exp(gamma*t1/90)+(-gamma/90/alpha)*log(u)
    if (x > 0) {
      new_t1 = t1 + (90/gamma)*log(x)
      if (new_t1 > .max_time) new_t1 = .max_time
    } else {
      new_t1 = .max_time
    }
    ev_times = c(ev_times,new_t1)
    t1 = new_t1
  }
  data.frame(ev_times=ev_times, event=as.numeric(ev_times < .max_time))
}
```

```r
draws <- spread_draws(fit_base_noiiv, log_alpha_pop, gamma) %>%
  slice_sample(n=500)

dind = d %>% filter(!duplicated(ID))

simulations = draws %>%
  nest(samples=-.draw) %>%
  mutate(sims = map(samples, function(.draws, .data=dind) {
    bind_cols(.data,.draws) %>%
      nest(.tmp = -ID) %>%
      mutate(sims = map(.tmp, sim_base)) %>%
      select(ID,sims) %>%
      unnest(sims)
  }))
```

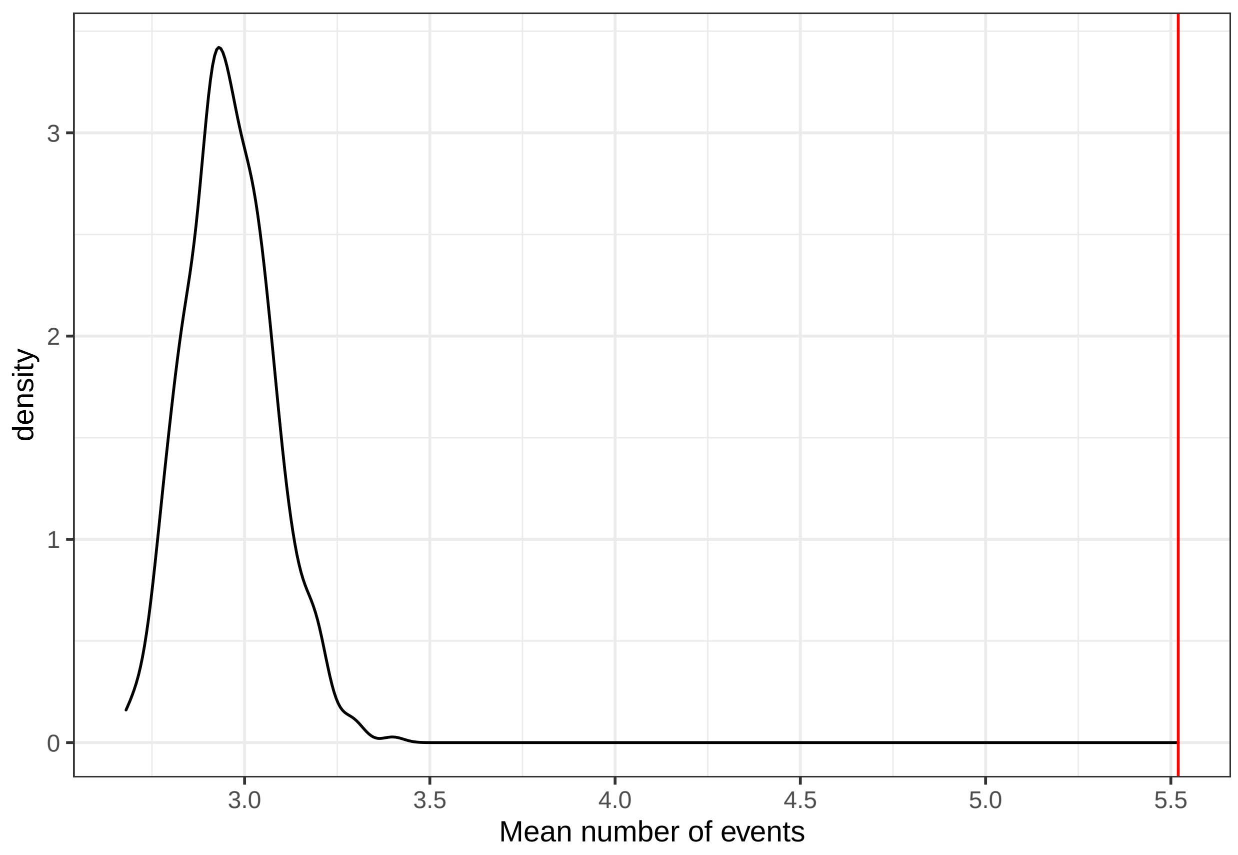
Predictive check for number of events

```r
foo=simulations %>%
  mutate(event_table = map(sims, ~.x %>% group_by(ID) %>% summarise(n=sum(event)) %>% count(num=n))) %>
  select(.draw,event_table) %>%
  unnest(event_table) %>%
  complete(num,nesting(.draw),fill=list(n=0)) %>%
  arrange(.draw,num)
```

Mean number of events

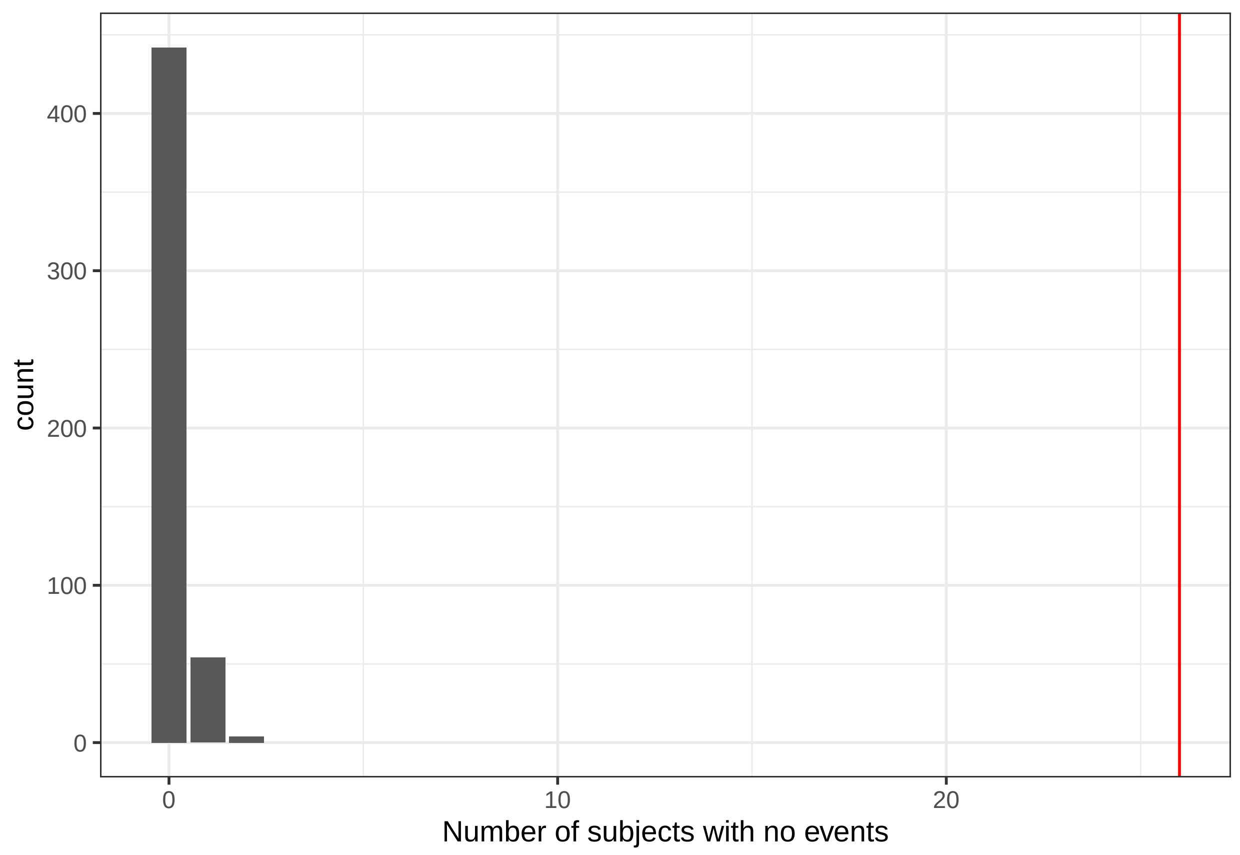
```r
summ_stats = foo %>%
  group_by(.draw) %>%
  summarise(wt_mean = sum(n*num)/sum(n),
         wt_sd = sqrt(sum(n*num^2)/sum(n) - wt_mean^2))
```

```r
summ_stats %>%
  ggplot(aes(x=wt_mean)) +
  geom_density() +
  geom_vline(xintercept = mean(d_ind$cumulative_events), col='red') +
  labs(x='Mean number of events')
```



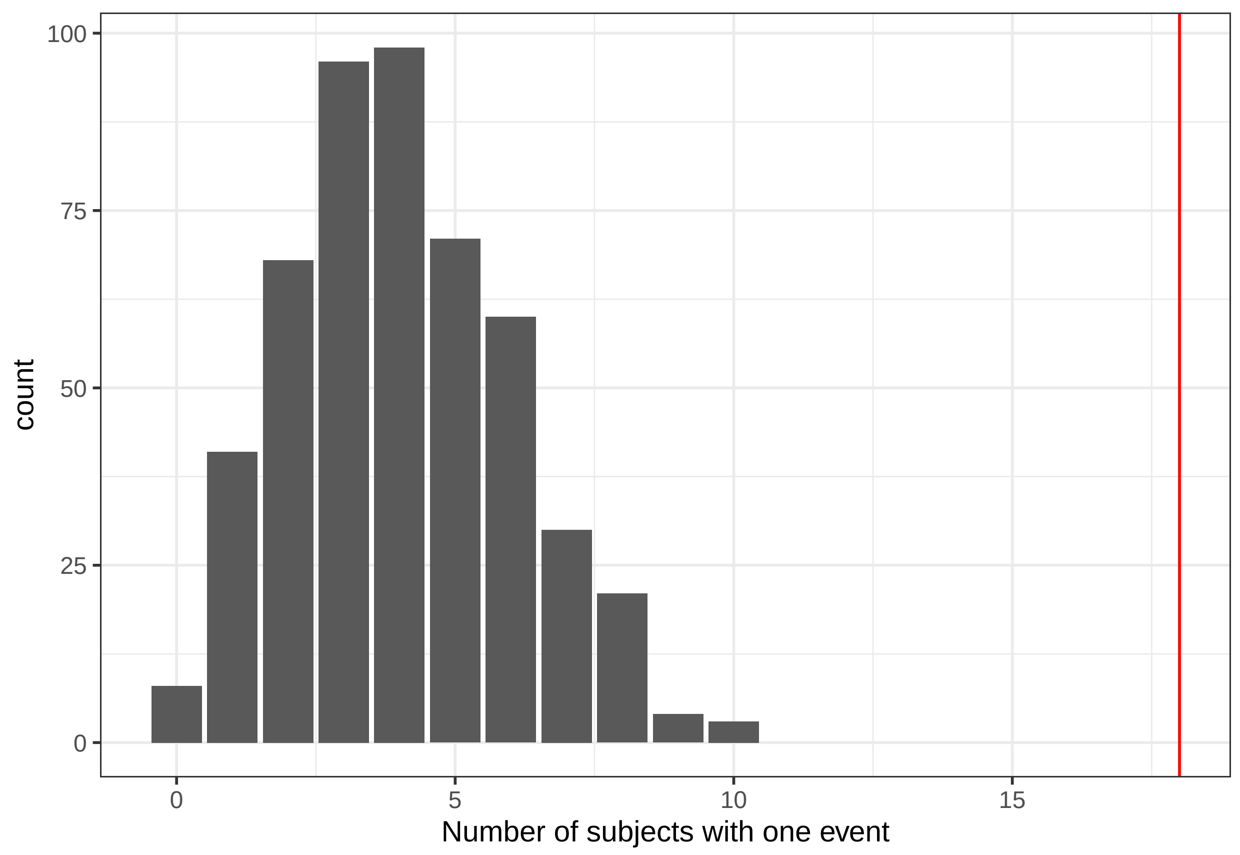
Distribution of number of subjects with no events

```
foo %>%
  filter(num==0) %>%
  ggplot(aes(n)) +
  geom_bar() +
  geom_vline(data=d_ind %>% summarise(n=sum(cumulative_events==0)),
             aes(xintercept=n), col='red') +
  labs(x='Number of subjects with no events')
```



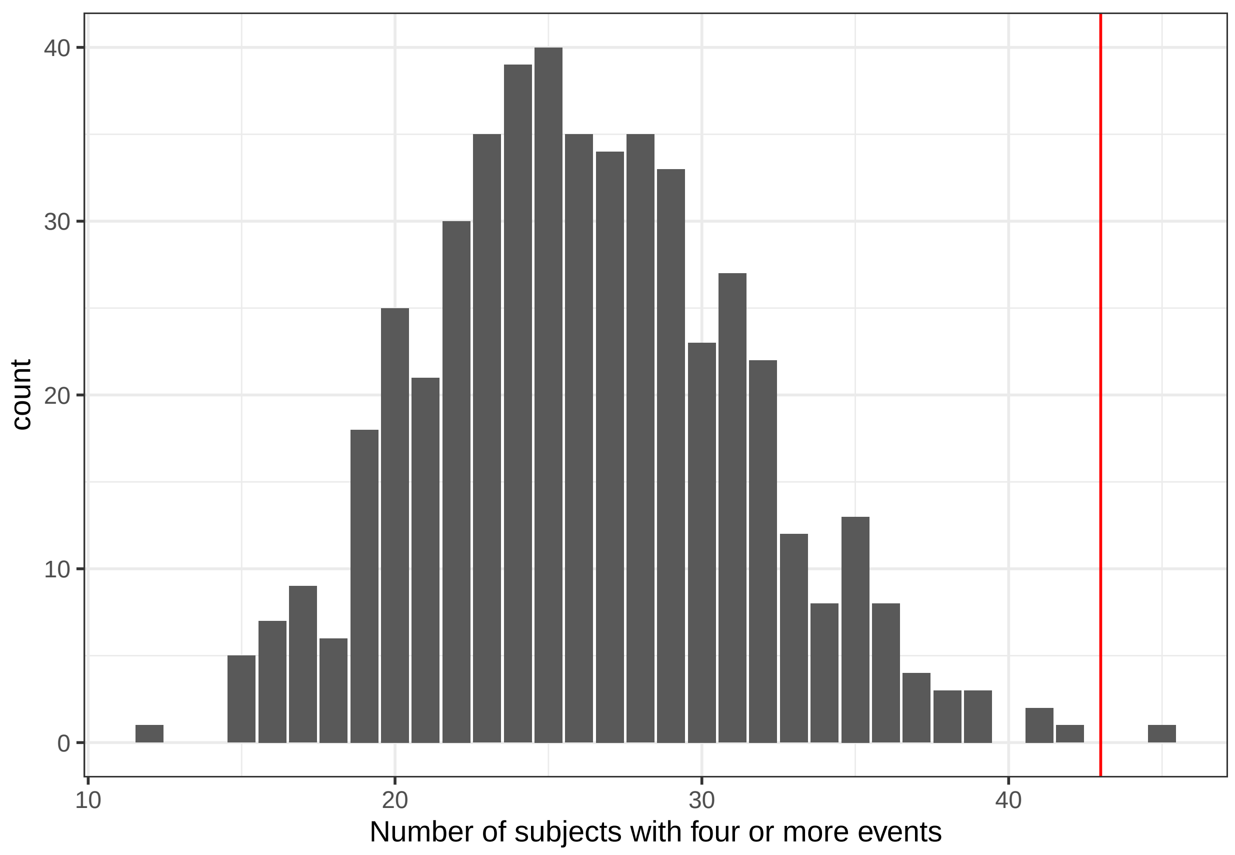
Distribution of number of subjects with one events

```
foo %>%
  filter(num==1) %>%
  ggplot(aes(n)) +
  geom_bar() +
  geom_vline(data=d_ind %>% summarise(n=sum(cumulative_events==1)),
             aes(xintercept=n), col='red') +
    labs(x='Number of subjects with one event')
```



Distribution of number of subjects with for or more events

```r
foo %>%
  filter(num>=4) %>%
  group_by(.draw) %>%
  summarise(n=sum(n)) %>%
  ggplot(aes(n)) +
  geom_bar() +
  geom_vline(data=d_ind %>% summarise(n=sum(cumulative_events>=4)),
             aes(xintercept=n), col='red') +
    labs(x='Number of subjects with four or more events')
```



Base model 2: With IIV, no exposure

Centered vs non-centered parameterization for hierarchical models

Centered parameterization:

$$\log \alpha_i \sim \text{normal}(\mu_\alpha, \omega)$$

Efficient when there are many groups and the data are very informative about the parameters (low shrinkage).

Non-centered parameterization:

$$\log \alpha_i = \mu_\alpha + \omega \times \eta_i \tag{1}$$
$$\eta_i \sim \text{normal}(0, 1) \tag{2}$$

More efficient when there are not many groups, when the inter-group variation is high, or when there is high shrinkage.

Let's compare formulations in this model. First, we'll start with the centered parameterization.

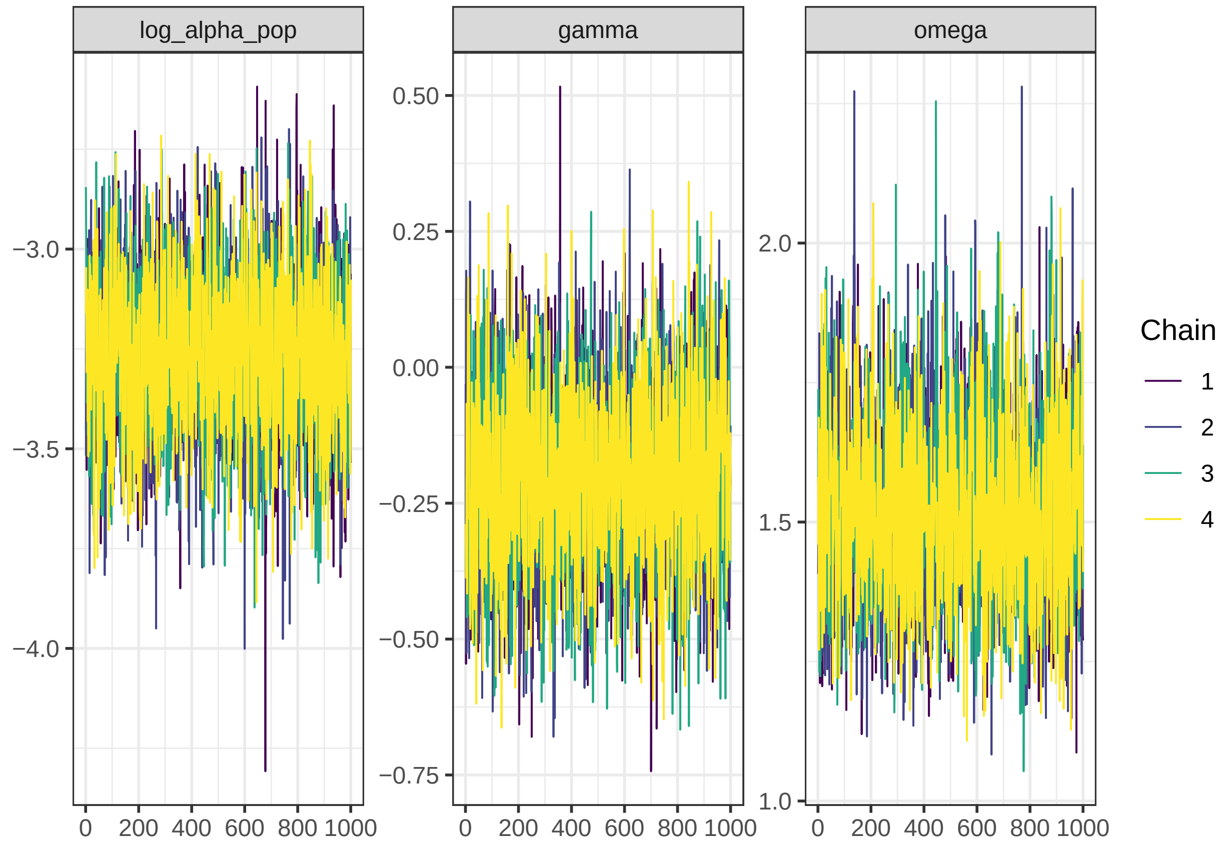
```
fit_base_centered_iiv <- stan(file = '../model/stan/Day4/rtte_gompertz_withiiv_centered.stan',
            data = stan_data,
            chains = 4,
            iter = 2000,
            cores = 4)
```

You may get one or two divergent transitions, but we can often fix a small numver of divergent transitions by increasing the `adapt_delta` parameter. A larger number of divergent transitions indicates the geometry of the problem is problematic and you should reparameterize the model.

```
print(fit_base_centered_iiv, pars=c('log_alpha_pop','gamma','omega'))
```

```
. Inference for Stan model: rtte_gompertz_withiiv_centered.
. 4 chains, each with iter=2000; warmup=1000; thin=1;
. post-warmup draws per chain=1000, total post-warmup draws=4000.
.
.                mean se_mean   sd  2.5%   25%   50%   75% 97.5% n_eff Rhat
. log_alpha_pop -3.26       0 0.20 -3.65 -3.38 -3.25 -3.12 -2.89  2974    1
. gamma         -0.20       0 0.16 -0.51 -0.31 -0.20 -0.09  0.11  2650    1
. omega          1.52       0 0.16  1.23  1.41  1.51  1.62  1.84  2100    1
.
. Samples were drawn using NUTS(diag_e) at Wed Aug 31 22:43:12 2022.
. For each parameter, n_eff is a crude measure of effective sample size,
. and Rhat is the potential scale reduction factor on split chains (at
. convergence, Rhat=1).
```

```
mcmc_trace(fit_base_centered_iiv, pars=c('log_alpha_pop', 'gamma', 'omega'))
```



Next, we'll fit the model with the non-centered parameterization

```
fit_base_noncentered_iiv <- stan(file = '../model/stan/Day4/rtte_gompertz_withiiv_noncentered.stan',
              data = stan_data,
              chains = 4,
              iter = 2000,
            cores = 4, control = list(max_treedepth=11))
```

```
print(fit_base_noncentered_iiv, pars=c('log_alpha_pop','gamma','omega'))
```

```
. Inference for Stan model: rtte_gompertz_withiiv_noncentered.
. 4 chains, each with iter=2000; warmup=1000; thin=1;
. post-warmup draws per chain=1000, total post-warmup draws=4000.
.
.                mean se_mean   sd  2.5%   25%   50%   75% 97.5% n_eff Rhat
. log_alpha_pop -3.27    0.01 0.20 -3.66 -3.40 -3.26 -3.14 -2.90   994    1
. gamma         -0.19    0.00 0.17 -0.52 -0.31 -0.19 -0.08  0.14  7404    1
. omega          1.52    0.01 0.16  1.23  1.41  1.51  1.62  1.88   724    1
.
. Samples were drawn using NUTS(diag_e) at Wed Aug 31 22:44:27 2022.
. For each parameter, n_eff is a crude measure of effective sample size,
. and Rhat is the potential scale reduction factor on split chains (at
. convergence, Rhat=1).
```

Note that for this model, the centered parameterization is more efficient in terms of effective sample sizes.

Let's simulate from this model to see if we get back a similar distribution of number of events and event times.

```
sim_base_iiv = function(.ingredients, .max_time=90) {
  # Gompertz cumulative hazard
  alpha = exp(.ingredients$log_alpha)
  gamma = .ingredients$gamma

  ev_times = numeric()
  t1 = 0
  while(t1 < .max_time) {
    u = runif(1)
    x = exp(gamma*t1/90)+(-gamma/90/alpha)*log(u)
    if (x > 0) {
      new_t1 = t1 + (90/gamma)*log(x)
      if (new_t1 > .max_time) new_t1 = .max_time
    } else {
      new_t1 = .max_time
    }
    ev_times = c(ev_times,new_t1)
    t1 = new_t1
  }
  data.frame(ev_times=ev_times, event=as.numeric(ev_times < .max_time))
}
```

```
draws <- spread_draws(fit_base_centered_iiv, log_alpha_pop, omega, gamma) %>%
  filter(.draw <= 500)

simulations = draws %>%
  nest(samples=-.draw) %>%
  mutate(sims = map(samples, function(.draws, .data=dind) {
    bind_cols(.data,.draws) %>%
      ungroup() %>%
      mutate(log_alpha = rnorm(n(),log_alpha_pop, omega)) %>%
      nest(.tmp = -ID) %>%
      mutate(sims = map(.tmp, sim_base_iiv)) %>%
      select(ID,sims) %>%
      unnest(sims)
  }))
```

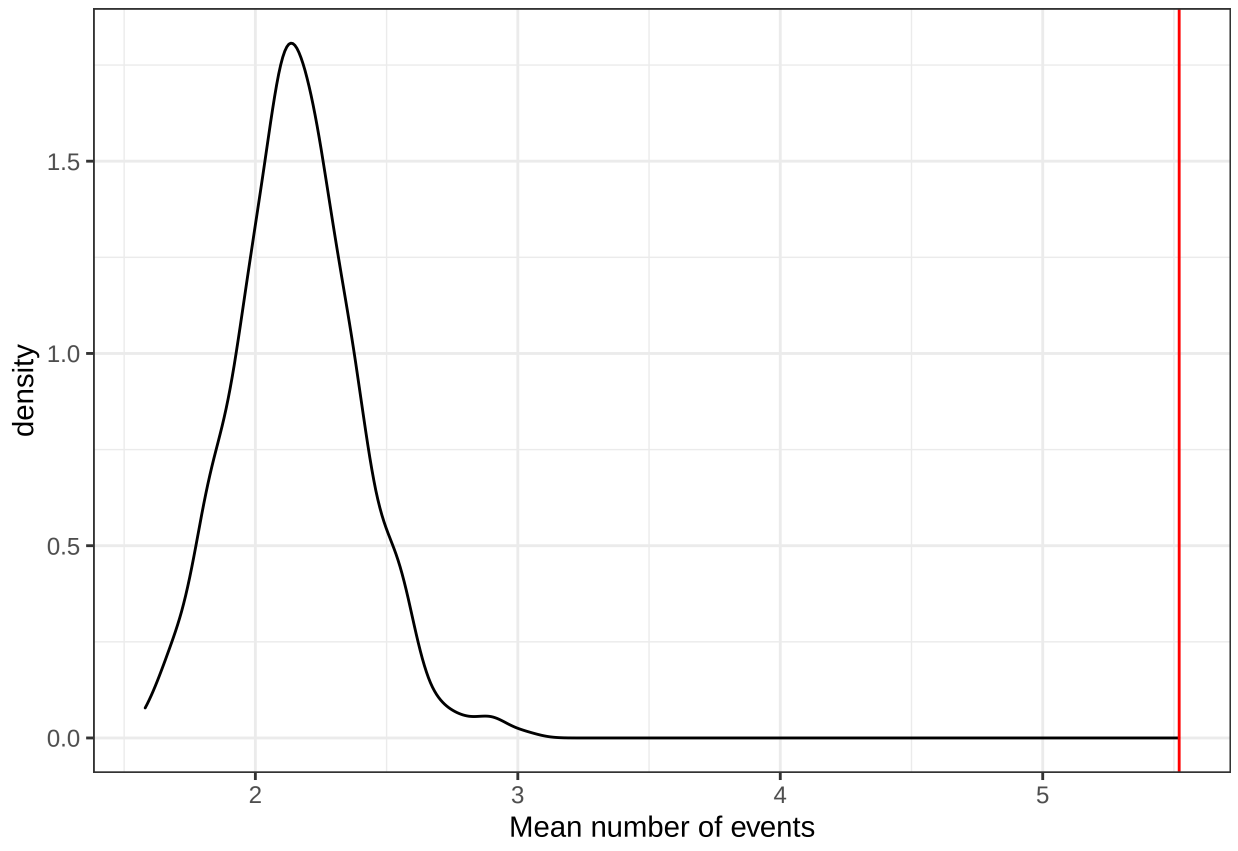
Predictive check for number of events

```
num_event_summary = simulations %>%
  mutate(event_table = map(sims, ~.x %>% group_by(ID) %>% summarise(n=sum(event)) %>% count(num=n))) %>%
  select(.draw,event_table) %>%
  unnest(event_table) %>%
  complete(num,nesting(.draw),fill=list(n=0)) %>%
  arrange(.draw,num)
```

Mean number of events

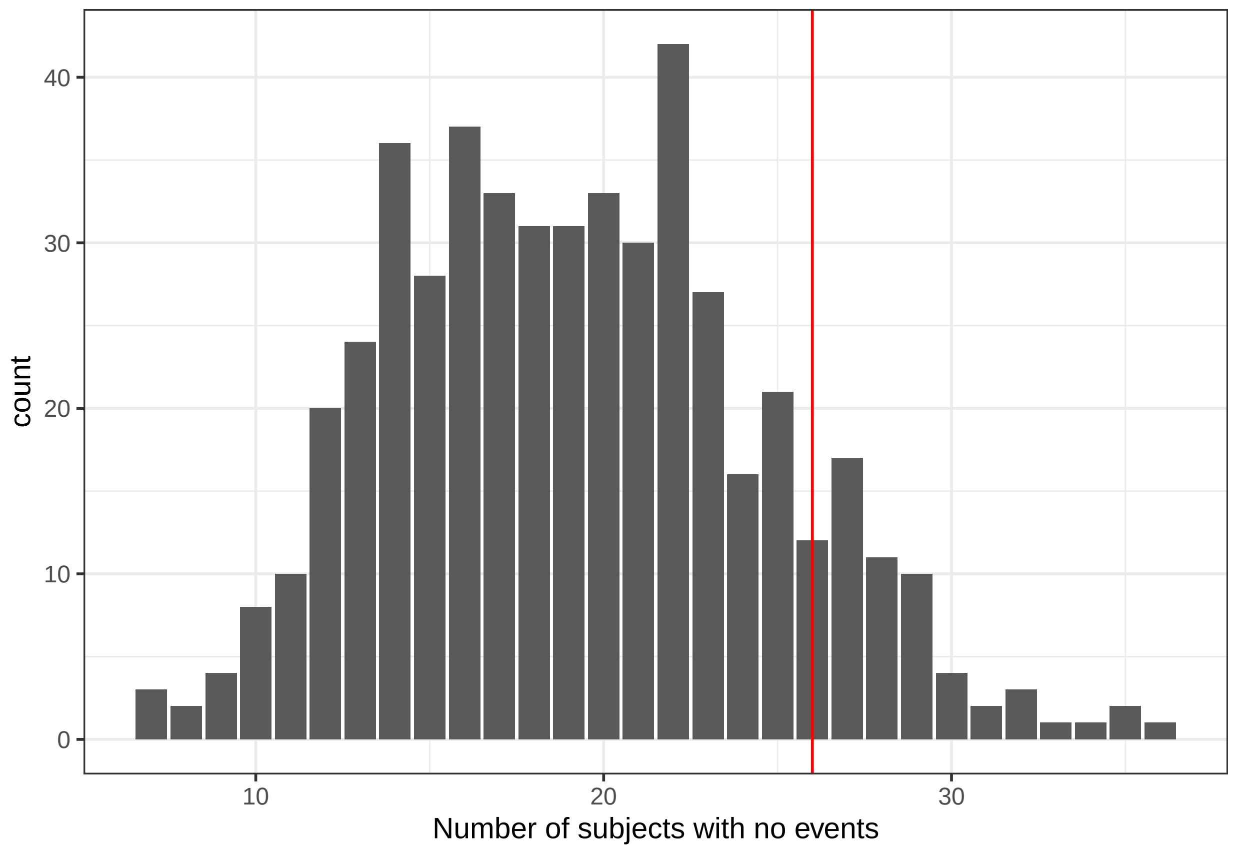
```
summ_stats = num_event_summary %>%
  group_by(.draw) %>%
  summarise(wt_mean = sum(n*num)/sum(n),
        wt_sd = sqrt(sum(n*num^2)/sum(n) - wt_mean^2))
```

```
summ_stats %>%
  ggplot(aes(x=wt_mean)) +
  geom_density() +
  geom_vline(xintercept = mean(d_ind$cumulative_events), col='red') +
  labs(x='Mean number of events')
```



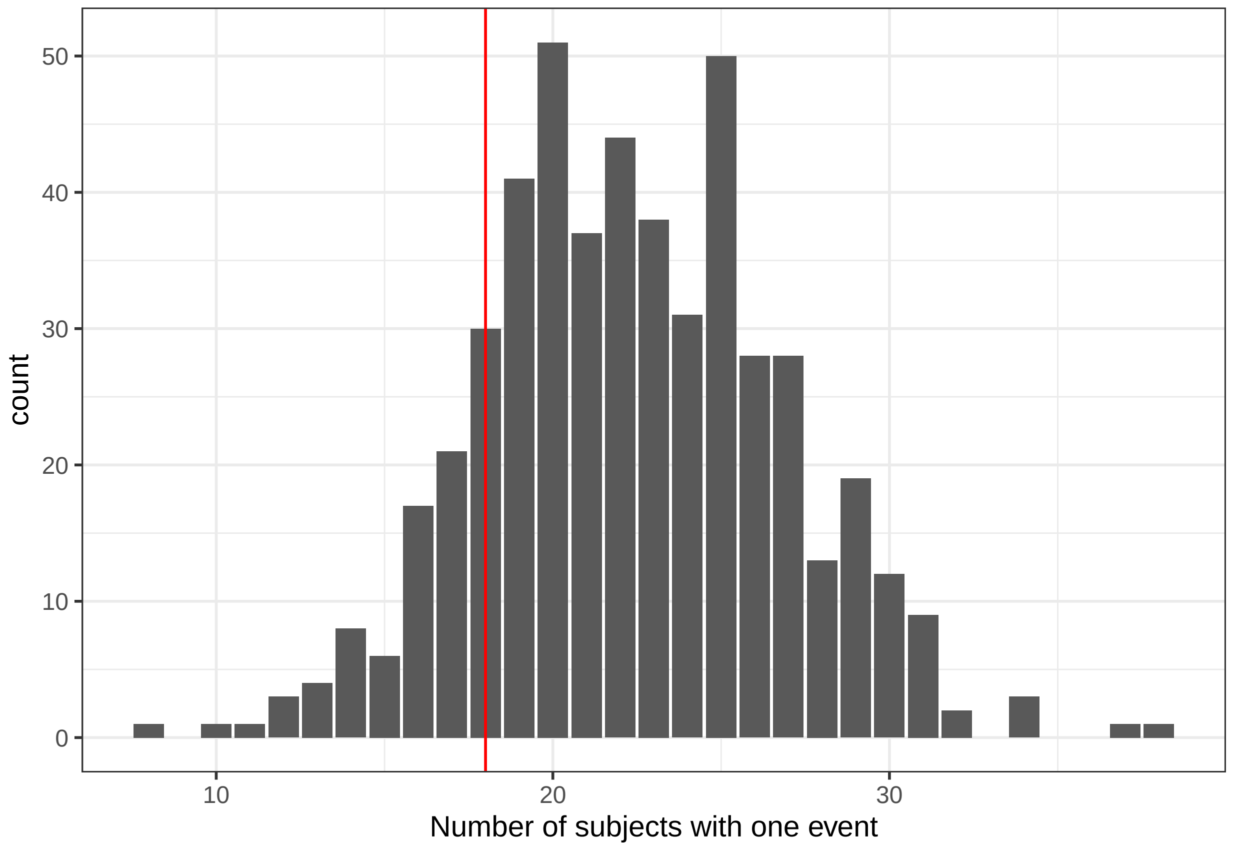
Distribution of number of subjects with no events

```
num_event_summary %>%
  filter(num==0) %>%
  ggplot(aes(n)) +
  geom_bar() +
  geom_vline(data=d_ind %>% summarise(n=sum(cumulative_events==0)),
             aes(xintercept=n), col='red') +
  labs(x='Number of subjects with no events')
```



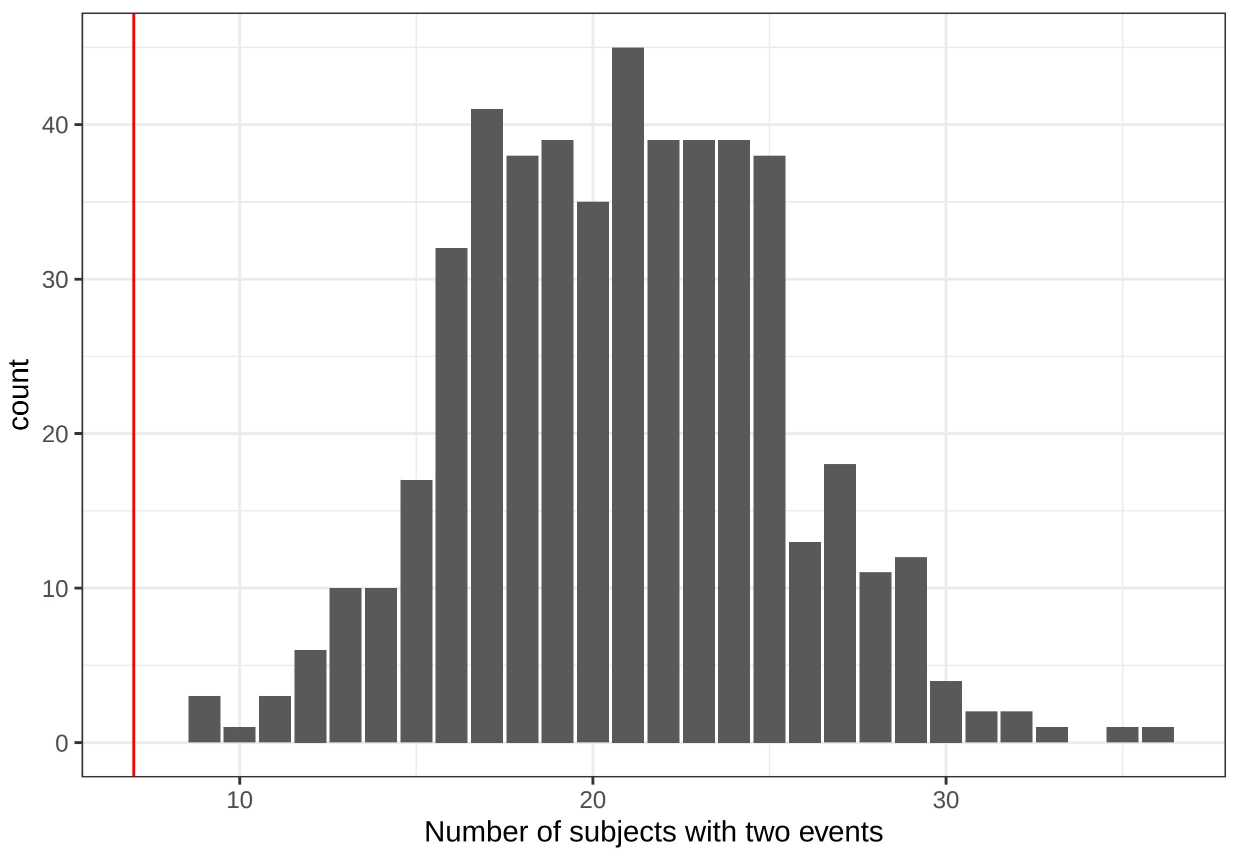
Distribution of number of subjects with one event

```
num_event_summary %>%
  filter(num==1) %>%
  ggplot(aes(n)) +
  geom_bar() +
  geom_vline(data=d_ind %>% summarise(n=sum(cumulative_events==1)),
             aes(xintercept=n), col='red') +
    labs(x='Number of subjects with one event')
```



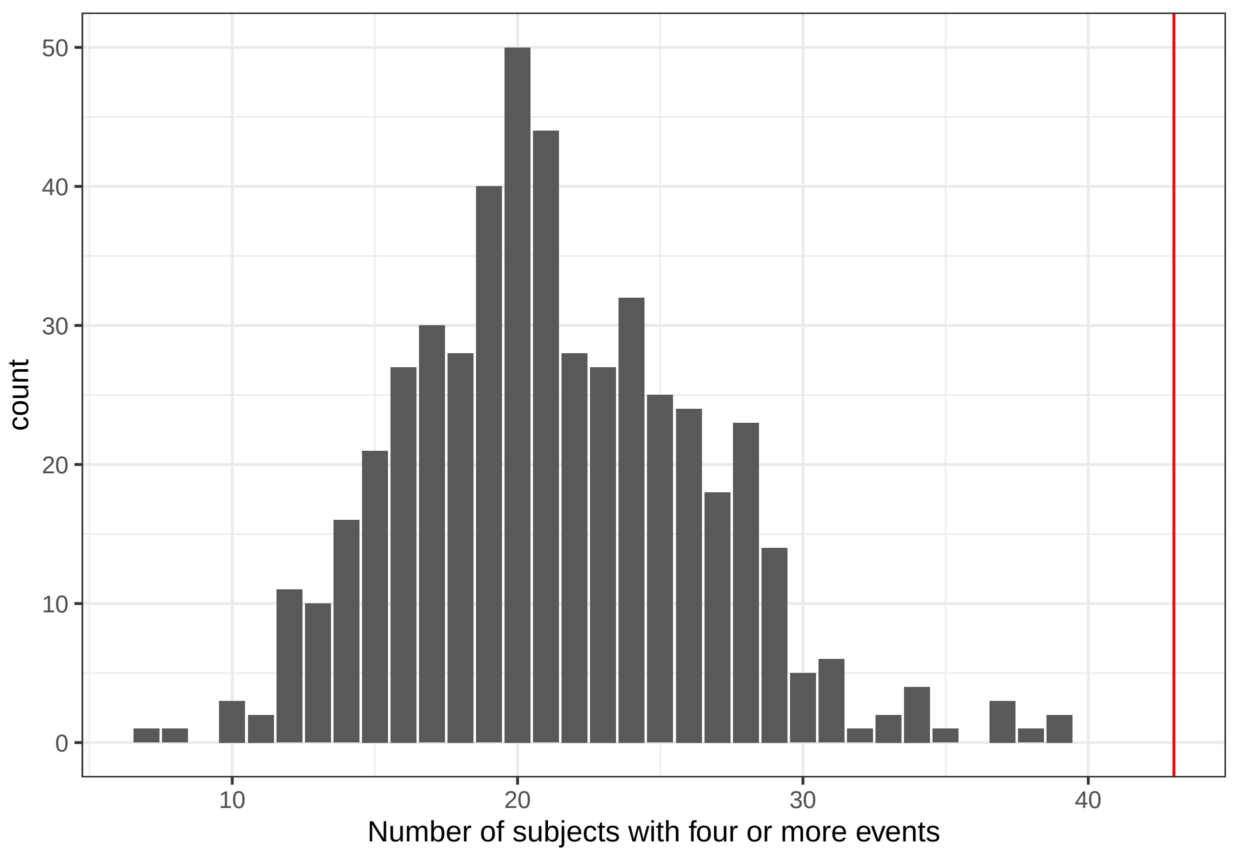
Distribution of number of subjects with two events

```r
num_event_summary %>%
  filter(num==2) %>%
  ggplot(aes(n)) +
  geom_bar() +
  geom_vline(data=d_ind %>% summarise(n=sum(cumulative_events==2)),
             aes(xintercept=n), col='red') +
    labs(x='Number of subjects with two events')
```



Distribution of number of subjects with for or more events

```
num_event_summary %>%
  filter(num>=4) %>%
  group_by(.draw) %>%
  summarise(n=sum(n)) %>%
  ggplot(aes(n)) +
  geom_bar() +
  geom_vline(data=d_ind %>% summarise(n=sum(cumulative_events>=4)),
             aes(xintercept=n), col='red') +
    labs(x='Number of subjects with four or more events')
```

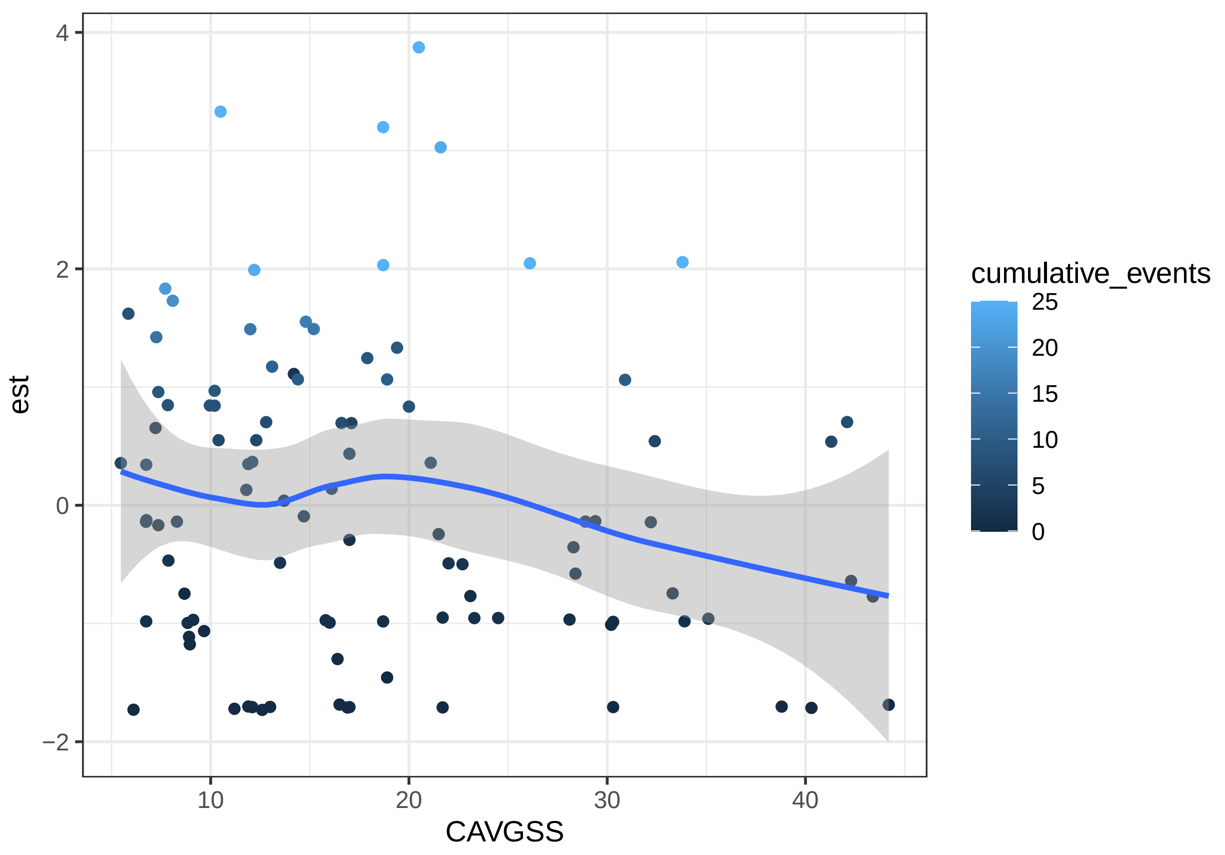


The model isn't perfect, but let's look at incorporating exposure.

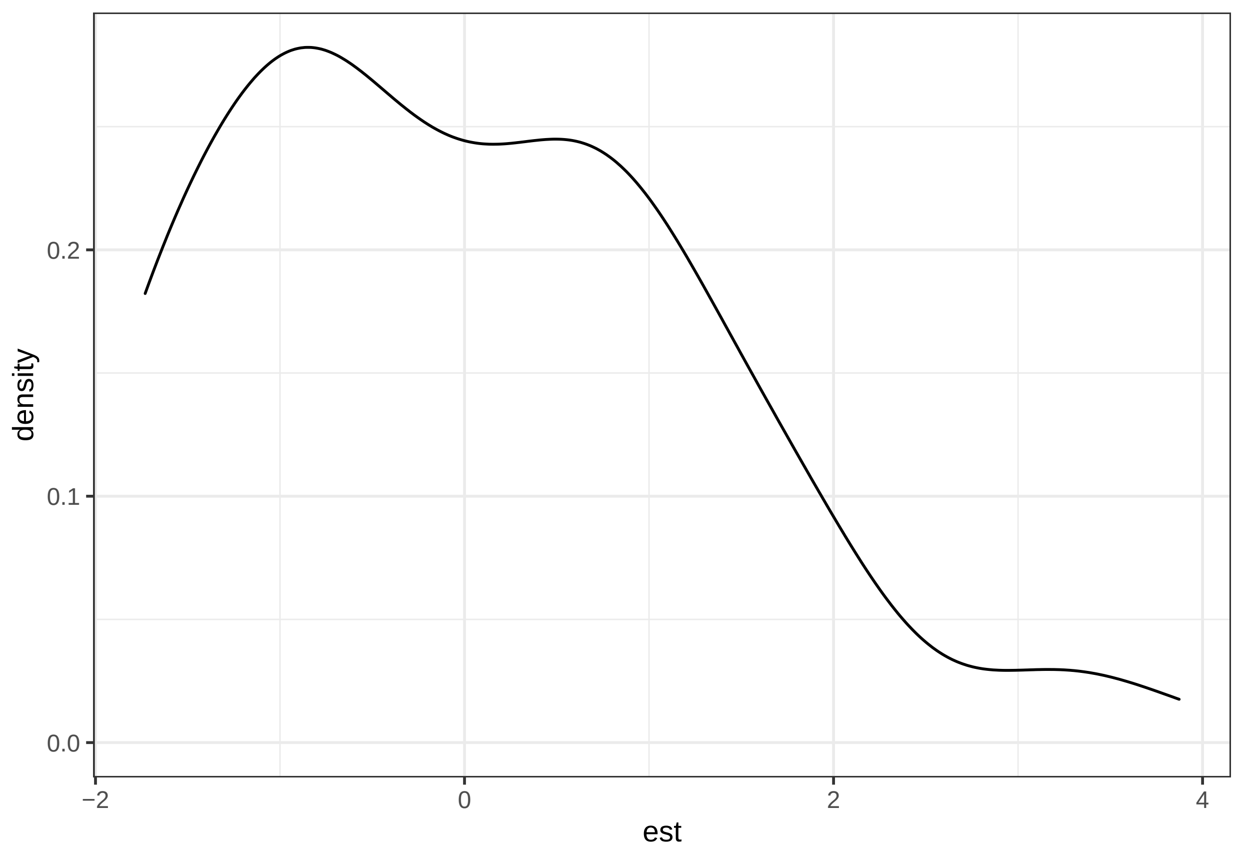
Plot etas vs CAVG

```
eta_sample <- spread_draws(fit_base_centered_iiv,log_alpha[ID], log_alpha_pop,omega)
```

```
d_post = eta_sample %>%
  group_by(ID) %>%
  summarise(est = median(log_alpha-log_alpha_pop)) %>%
  left_join(d_ind)

d_post %>% ggplot(aes(x=CAVGSS,y=est, col=cumulative_events))+geom_point() + geom_smooth()
```



```
d_post %>%
  ggplot(aes(x=est)) +
  geom_density()
```



Exposure-response model

Let's try fitting a log-linear model for the effect of steady-state Cavg on the scale parameter (α).

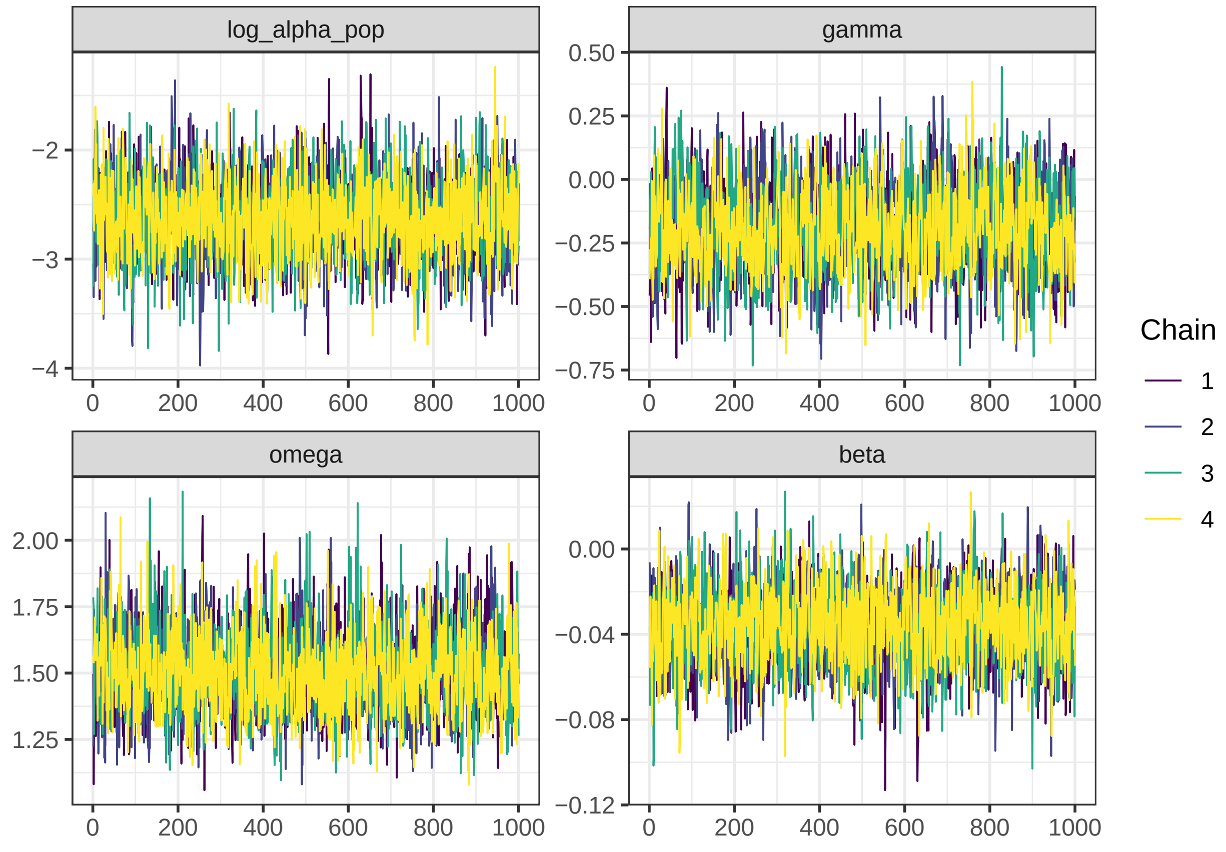
$$h_i(t) = \alpha_i \exp(-\gamma \cdot t) \tag{3}$$
$$\alpha_i = \alpha_{pop} \exp(\beta \cdot \text{CAVGSS}_i) \tag{4}$$

```
stan_data = list(Nobs=nrow(d),
                 Nsubj = nrow(d_ind),
                 delta = d$event,
                 time = d$TIME,
                 prev_time = d$PREVTIME,
                 ID = d$ID ,
                 CAVGSS = d_ind$CAVGSS)

fit_expresp <- stan(file = '../model/stan/Day4/rtte_gompertz_expresp_linear.stan',
            data = stan_data,
            chains = 4,
            iter = 2000,
            cores = 4,
            control = list(adapt_delta=0.90))
```

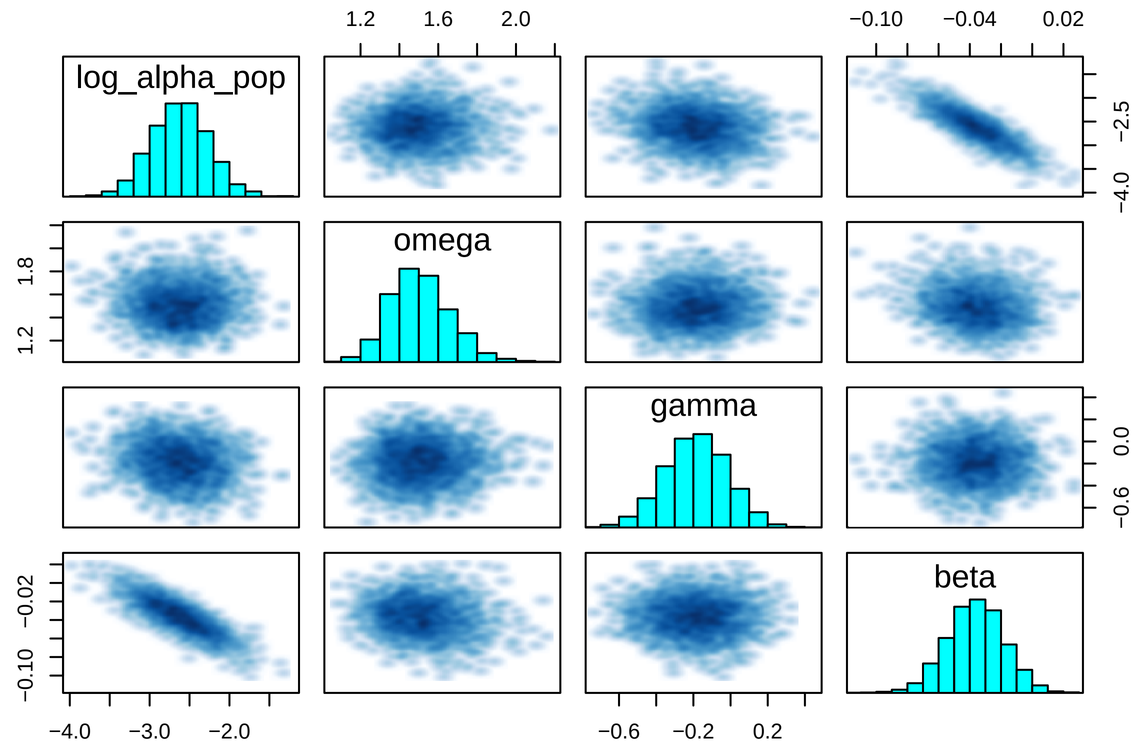
```
print(fit_expresp, pars=c('log_alpha_pop','gamma','omega','beta'))
```

```
. Inference for Stan model: rtte_gompertz_expresp_linear.
. 4 chains, each with iter=2000; warmup=1000; thin=1;
. post-warmup draws per chain=1000, total post-warmup draws=4000.
.
.                mean se_mean   sd  2.5%   25%   50%   75% 97.5% n_eff Rhat
. log_alpha_pop -2.62    0.01 0.36 -3.33 -2.86 -2.62 -2.38 -1.91  2363    1
. gamma         -0.19    0.00 0.17 -0.52 -0.30 -0.18 -0.07  0.14  1320    1
. omega          1.51    0.00 0.16  1.23  1.39  1.50  1.60  1.83  1533    1
. beta          -0.04    0.00 0.02 -0.07 -0.05 -0.04 -0.02  0.00  2391    1
.
. Samples were drawn using NUTS(diag_e) at Wed Aug 31 22:46:13 2022.
. For each parameter, n_eff is a crude measure of effective sample size,
. and Rhat is the potential scale reduction factor on split chains (at
. convergence, Rhat=1).
```

```
bayesplot::mcmc_trace(fit_expresp,pars=c('log_alpha_pop','gamma','omega','beta'))
```



```
pairs(fit_expresp, pars=c('log_alpha_pop','omega','gamma','beta'))
```



Convergence looks good, but there is a high posterior correlation between `log_alpha_pop` and `beta`. How might you address this?

Let's look at some predictive checks

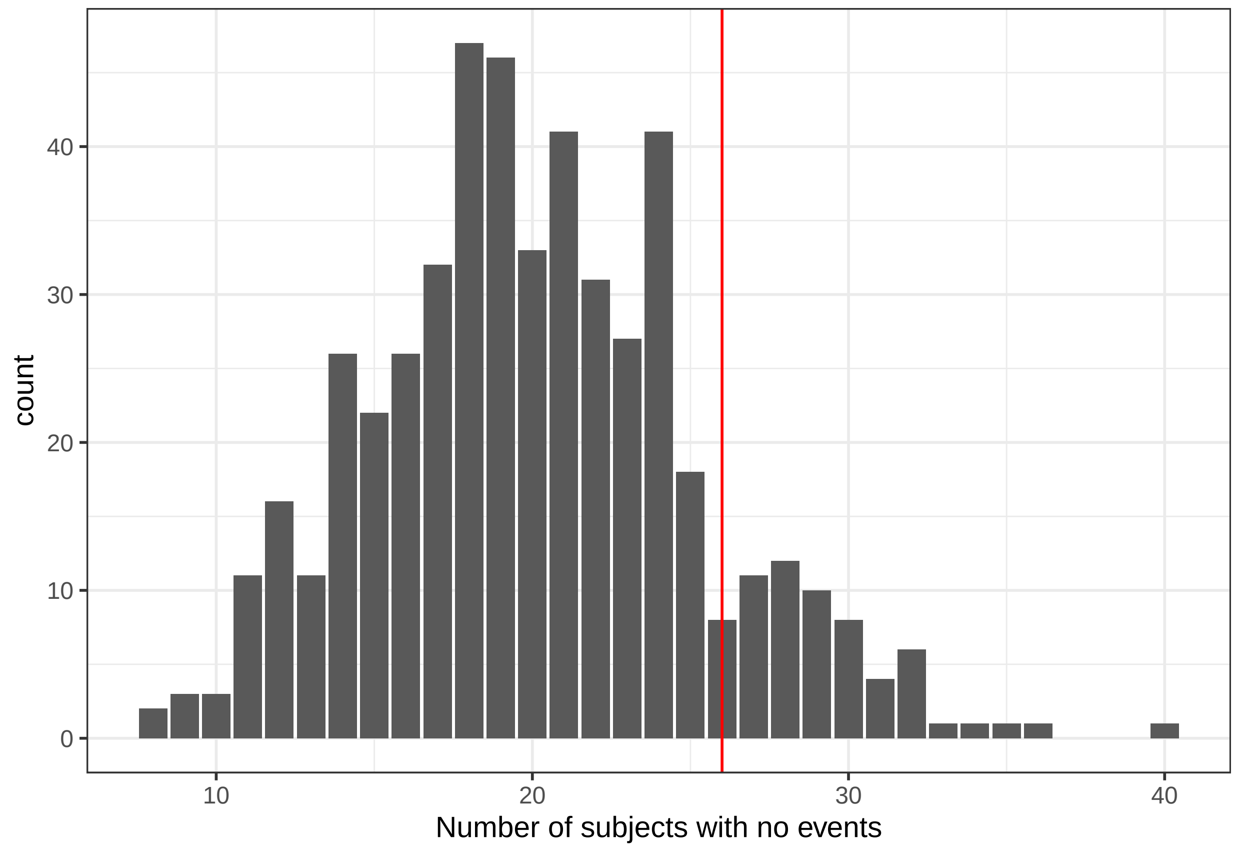
```
draws <- spread_draws(fit_expresp, log_alpha_pop, beta, omega, gamma) %>%
  filter(.draw <= 500)

simulations = draws %>%
  nest(samples=-.draw) %>%
  mutate(sims = map(samples, function(.draws, .data=d_ind) {
    bind_cols(.data,.draws) %>%
      ungroup() %>%
      mutate(log_alpha = rnorm(n(),log_alpha_pop + beta*CAVGSS, omega)) %>%
      nest(.tmp = -ID) %>%
      mutate(sims = map(.tmp, sim_base_iiv)) %>%
      select(ID,sims) %>%
      unnest(sims)
  }))
```

Predictive check for number of events

```
num_event_summary = simulations %>%
  mutate(event_table = map(sims, ~.x %>% group_by(ID) %>% summarise(n=sum(event)) %>% count(num=n))) %>
  select(.draw,event_table) %>%
  unnest(event_table) %>%
  complete(num,nesting(.draw),fill=list(n=0)) %>%
  arrange(.draw,num)
```

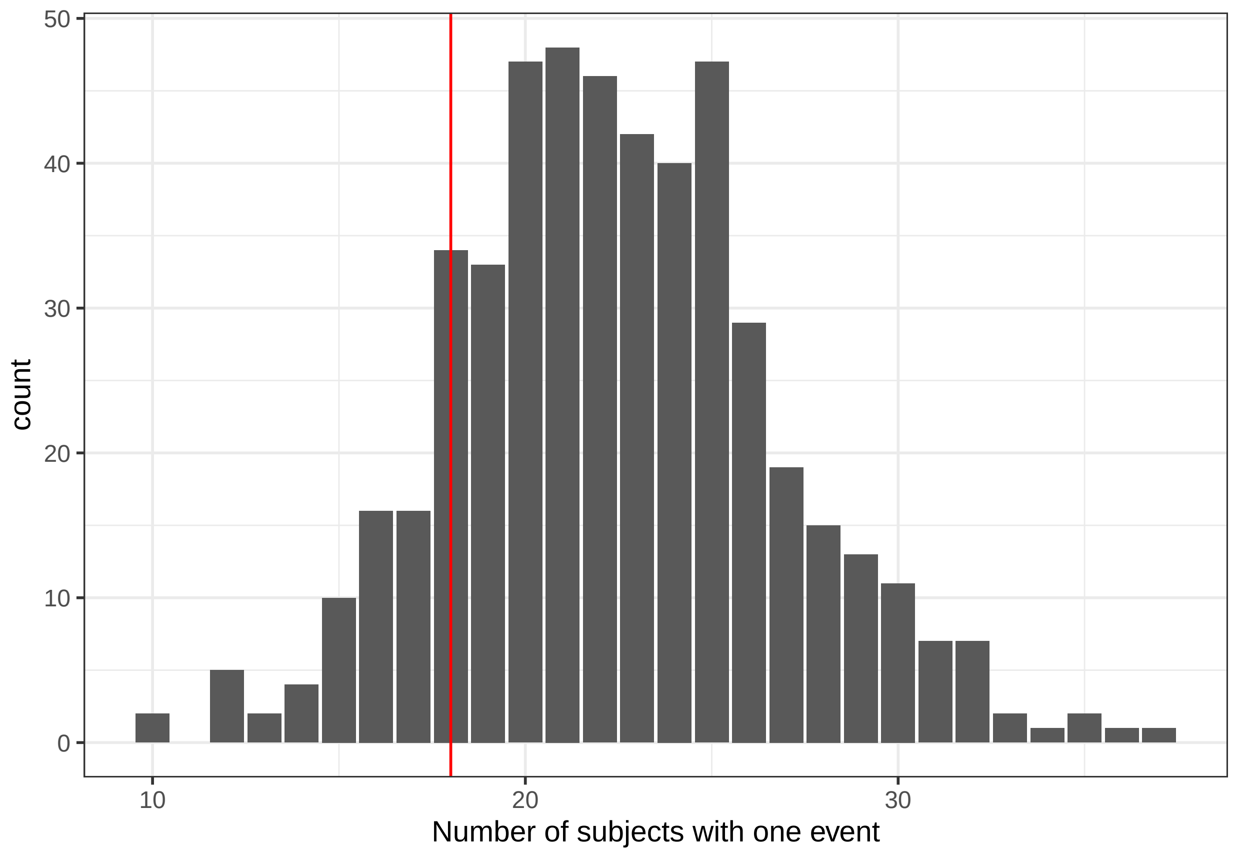
Distribution of number of subjects with no events

```
num_event_summary %>%
  filter(num==0) %>%
  ggplot(aes(n)) +
  geom_bar() +
  geom_vline(data=d_ind %>% summarise(n=sum(cumulative_events==0)),
             aes(xintercept=n), col='red') +
  labs(x='Number of subjects with no events')
```



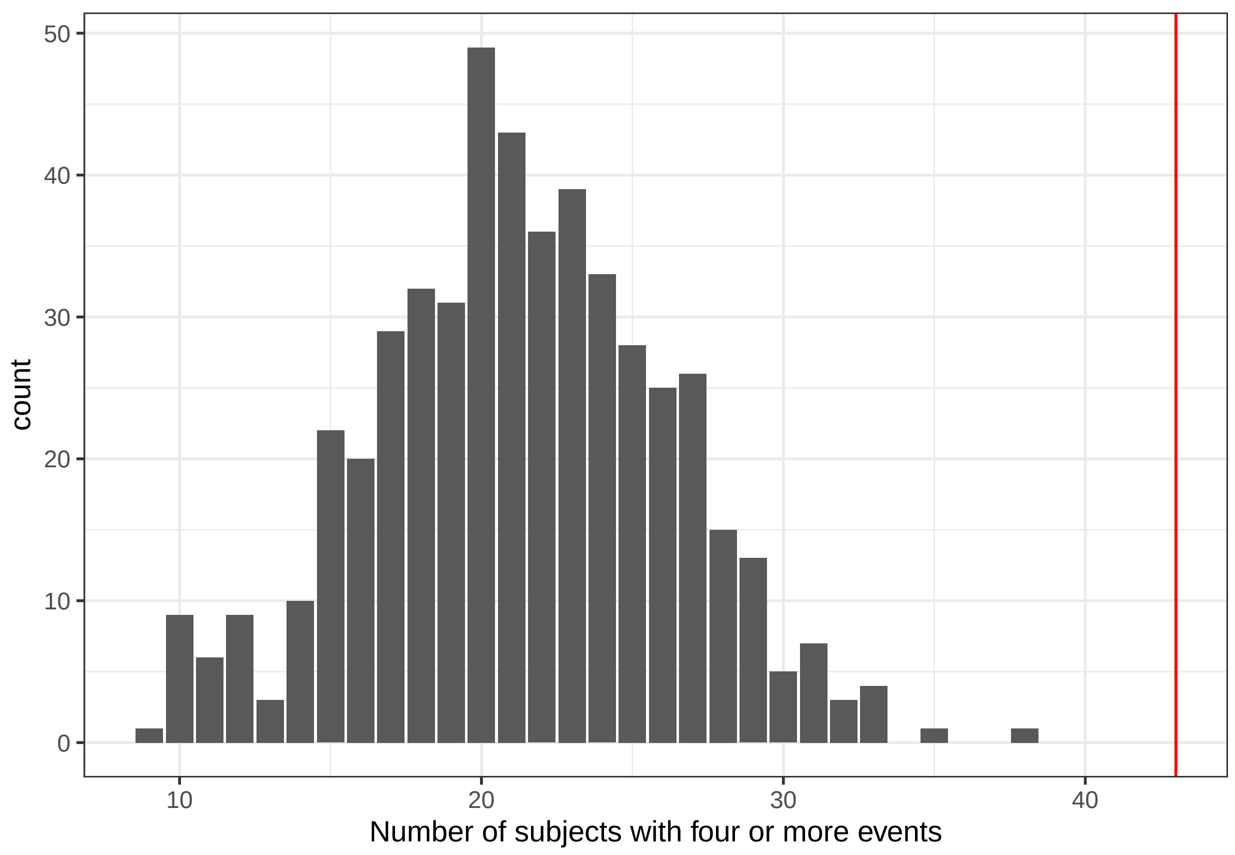
Distribution of number of subjects with one events

```
num_event_summary %>%
  filter(num==1) %>%
  ggplot(aes(n)) +
  geom_bar() +
  geom_vline(data=d_ind %>% summarise(n=sum(cumulative_events==1)),
             aes(xintercept=n), col='red') +
    labs(x='Number of subjects with one event')
```



Distribution of number of subjects with four or more events

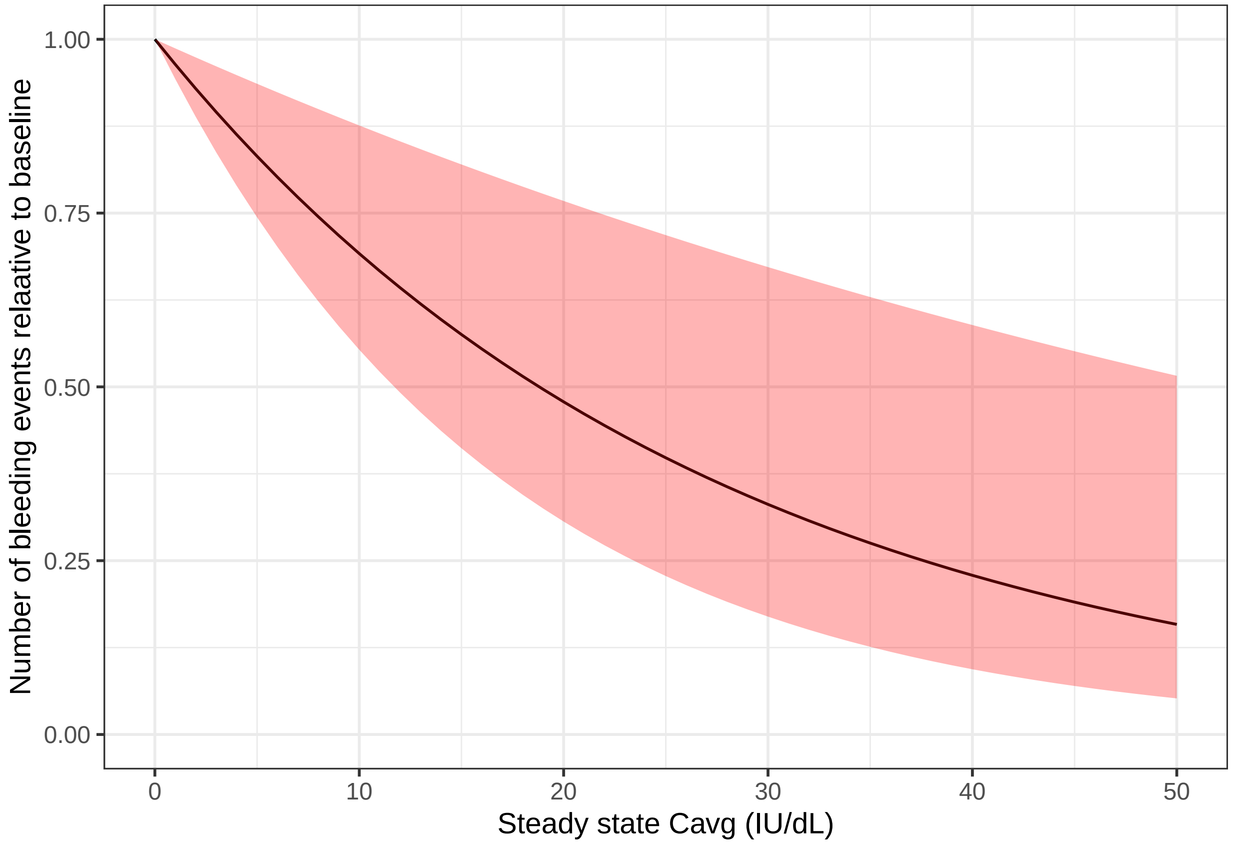
```r
num_event_summary %>%
  filter(num>=4) %>%
  group_by(.draw) %>%
  summarise(n=sum(n)) %>%
  ggplot(aes(n)) +
  geom_bar() +
  geom_vline(data=d_ind %>% summarise(n=sum(cumulative_events>=4)),
             aes(xintercept=n), col='red') +
    labs(x='Number of subjects with four or more events')
```



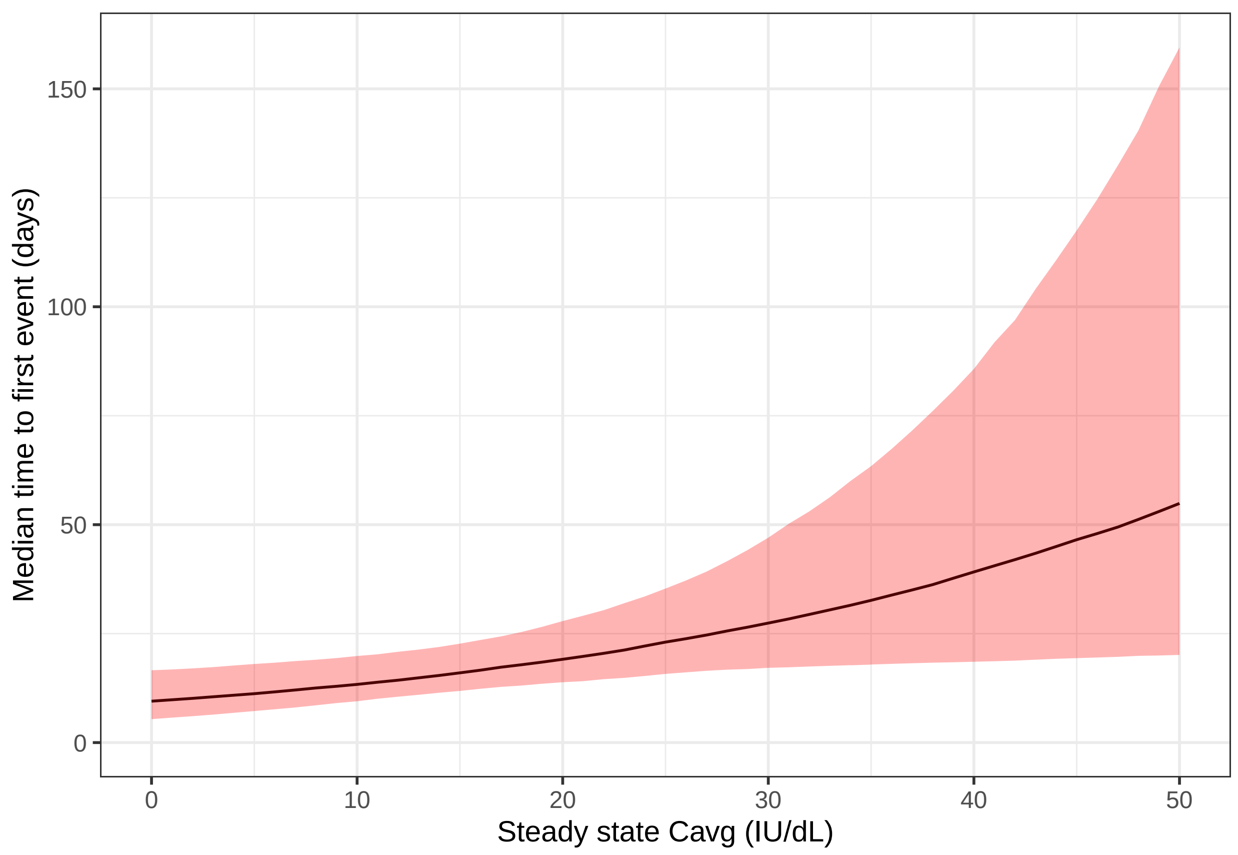
Let's plot the estimated exposure-response:

```
draws <- spread_draws(fit_expresp, log_alpha_pop, beta, gamma, omega) %>%
  filter(.draw <= 1000) %>%
  crossing(CAVGSS = seq(from=0, to=50, by=1)) %>%
  ungroup() %>%
  mutate(log_alpha_i = log_alpha_pop + beta*CAVGSS, #rnorm(n(),log_alpha_pop + beta*CAVGSS, omega ),
         alpha_i = exp(log_alpha_i),
         median_tte = -90/gamma * log(1-(gamma/90)/alpha_i * log(2)),
         expected_events = alpha_i / (gamma/90) * exp(-gamma * 90/90))

draws_pbo = draws %>%
  filter(CAVGSS==0) %>%
  select(.draw,median_tte_pbo=median_tte, expected_events_pbo=expected_events)

draws = draws %>% left_join(draws_pbo) %>%
  mutate(rel_median_tte = median_tte/median_tte_pbo,
         rel_expected_events = expected_events/expected_events_pbo)
```

```
draws %>%
  group_by(CAVGSS) %>%
  summarise(median = median(rel_expected_events),
            lcl = quantile(rel_expected_events,probs = 0.10),
            ucl = quantile(rel_expected_events, probs = 0.90)) %>%
  ggplot(aes(x=CAVGSS, y=median)) +
  geom_line() +
  geom_ribbon(aes(ymin=lcl, ymax=ucl), fill='red', alpha=0.3) +
  ylim(c(0,NA)) +
  labs(x='Steady state Cavg (IU/dL)', y='Number of bleeding events relaative to baseline')
```



```
draws %>%
  group_by(CAVGSS) %>%
  summarise(median = median(median_tte, na.rm = TRUE),
            lcl = quantile(median_tte,probs = 0.05, na.rm = TRUE),
            ucl = quantile(median_tte, probs = 0.95, na.rm = TRUE)) %>%
  ggplot(aes(x=CAVGSS, y=median)) +
  geom_line() +
  geom_ribbon(aes(ymin=lcl, ymax=ucl), fill='red', alpha=0.3) +
  ylim(c(0,NA)) +
  labs(x='Steady state Cavg (IU/dL)', y='Median time to first event (days)')
```



Exercise: Code a model which replaces the log-linear effect of CAVGSS with an inhibitory Emax model.

Predicting to a 100 IU/kg dose

1. Simulate exposures at the 100 IU/g dose

a. Assume similar population -> similar distribution of CL

2. Given exposure distribution, compute

a. Expected number of events over 90 days by dose (20, 50 100 IU/kg)
b. Time to first event by dose

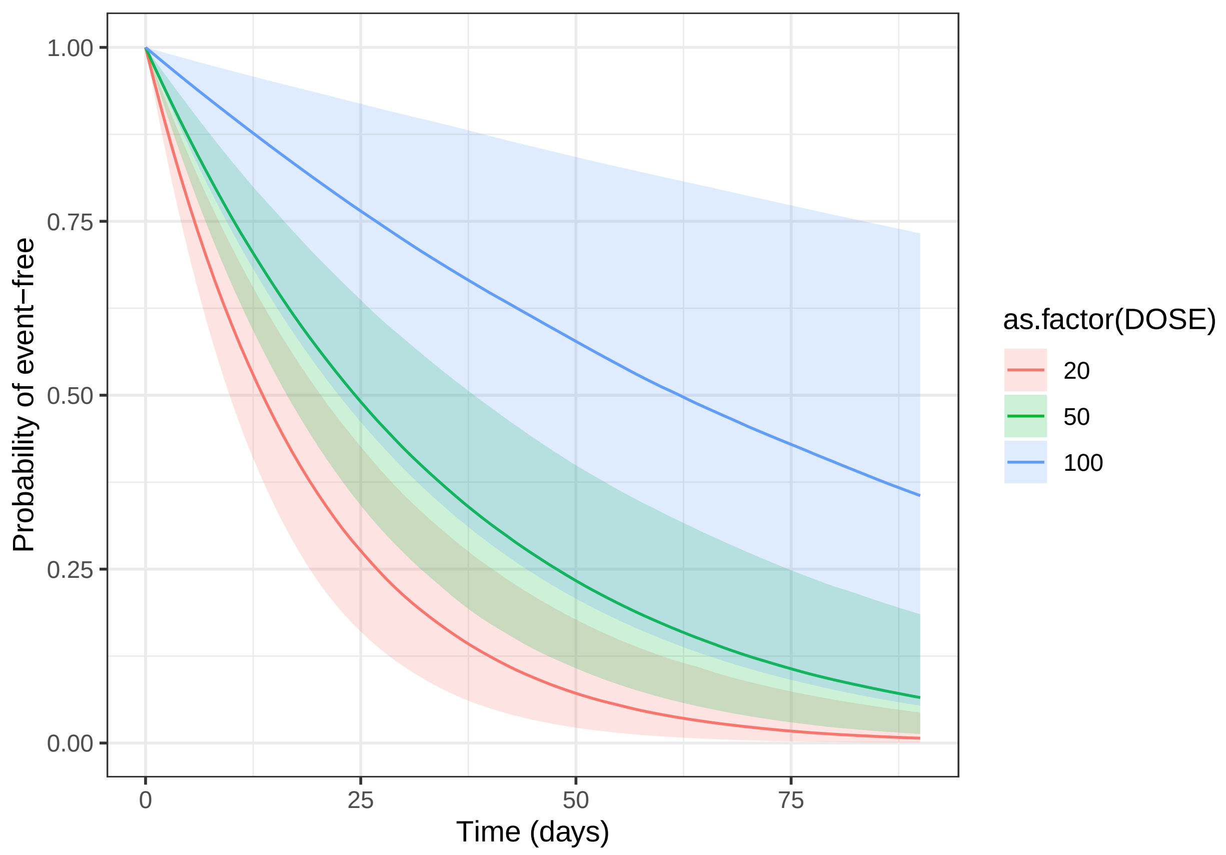
Simulate exposures

```
nsim = 500

d_sims = d_ind %>% select(BWT,CL) %>% slice_sample(n=nsim, replace = TRUE) %>%
  mutate(ID=1:n()) %>%
  crossing(DOSE=c(20,50,100)) %>%
  mutate(CAVGSS = DOSE*BWT/CL/(24*3))
```

```
num_events_function <- function(.omega, .log_alpha_pop, .beta, .gamma, .data=d_sims) {
  .data %>%
    mutate(eta = rnorm(n(), 0, .omega),
           alpha = exp(.log_alpha_pop + .beta*CAVGSS + eta),
           cumulative_hazard = alpha/(.gamma/90)*(1-exp(-.gamma*90/90))) %>%
    group_by(DOSE) %>%
    summarise(median_events = median(cumulative_hazard))
}

survival_function <- function(.omega,.log_alpha_pop, .beta, .gamma, .data=d_sims) {
  .data %>%
    mutate(eta = rnorm(n(), 0, .omega),
           alpha = exp(.log_alpha_pop + .beta*CAVGSS + eta)) %>%
    crossing(time = 0:90) %>%
    mutate(cumulative_hazard = alpha/(.gamma/90)*(1-exp(-.gamma*time/90)),
           surv = exp(-cumulative_hazard)) %>%
    group_by(DOSE,time) %>%
    summarise(median_surv = median(surv))
}


draws = spread_draws(fit_expresp, omega, log_alpha_pop, beta, gamma) %>%
  filter(.draw <= 500) %>%
  mutate(mean_num_events = pmap(list(omega,log_alpha_pop,beta,gamma), .f=num_events_function),
         survival = pmap(list(omega,log_alpha_pop,beta,gamma), .f=survival_function)
  )
```

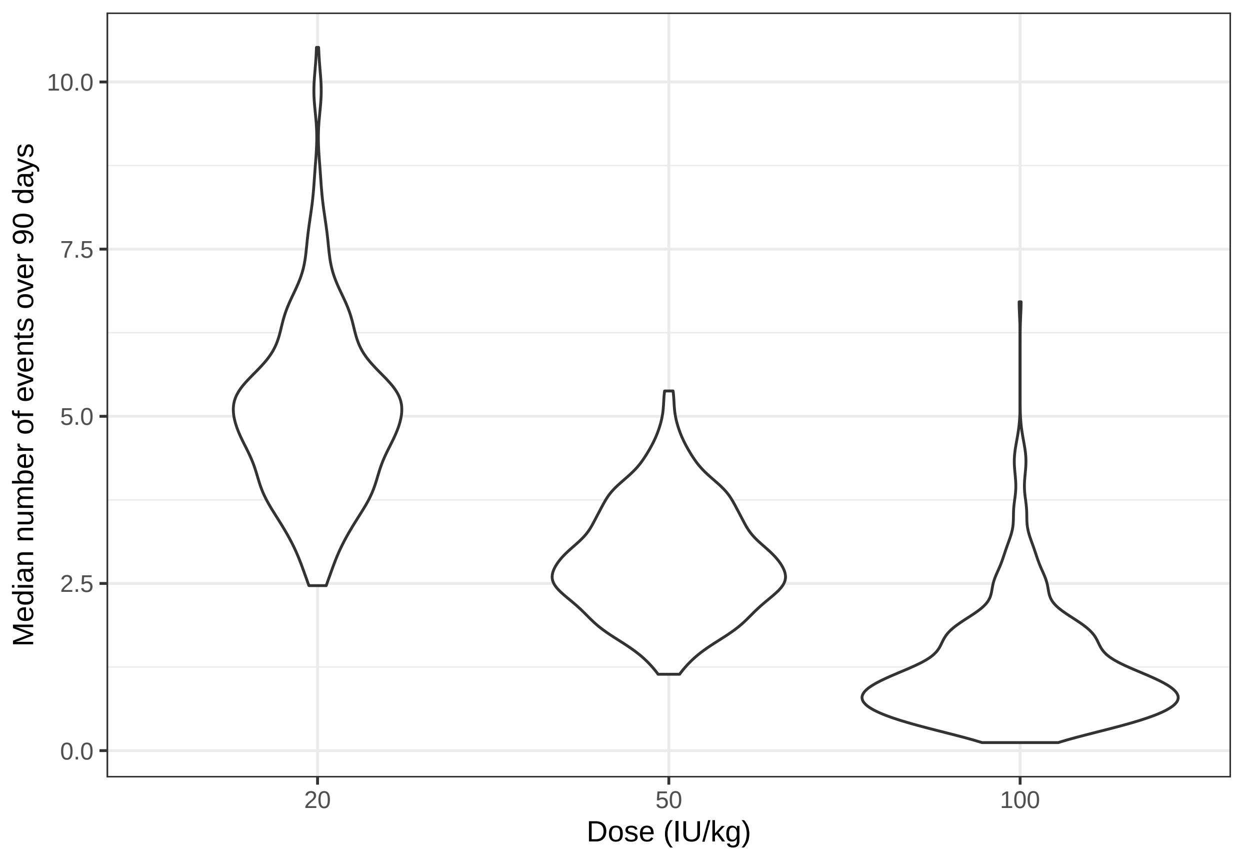
Plot distribution of time to first event

```
summary = draws %>%
  select(.draw,survival) %>%
  unnest(survival) %>%
  group_by(DOSE,time) %>%
  summarise(lcl = quantile(median_surv,probs = 0.05),
            pred = quantile(median_surv,probs = 0.50),
            ucl = quantile(median_surv,probs = 0.95))

summary %>%
  ggplot(aes(x=time, group=DOSE)) +
  geom_line(aes(y=pred, col=as.factor(DOSE))) +
  geom_ribbon(aes(ymin=lcl, ymax=ucl,fill=as.factor(DOSE)), alpha=0.2) +
  labs(x='Time (days)', y='Probability of event-free')
```



Plot distributions of expected number of events

```
summary = draws %>%
  select(.draw,mean_num_events) %>%
  unnest(mean_num_events)

summary %>%
  ggplot(aes(x=as.factor(DOSE), y=median_events)) +
  geom_violin() +
  labs(y='Median number of events over 90 days', x='Dose (IU/kg)')
```



Median number of events over 90 days
10.0
7.5
5.0
2.5
0.0
20
50
100
Dose (IU/kg)