HW 3 Solutions

Shravan Vasishth

Exercise 1

We need to create a new column first (there are other ways of doing this):

```
data("df_powerpose")
head(df_powerpose)
```

```
##   id hptreat female age testm1 testm2
## 2 29    High   Male  19   38.7   62.4
## 3 30     Low Female  20   32.8   29.2
## 4 31    High Female  20   32.3   27.5
## 5 32     Low Female  18   18.0   28.7
## 7 34     Low Female  21   73.6   44.7
## 8 35    High Female  20   80.7  105.5
```

```
df_powerpose <- mutate(df_powerpose, change = testm2 - testm1)
```

```
## use default priors
fit_powerpose <- brm(change ~ hptreat,data =  df_powerpose)
```

```
fit_powerpose
```

```
##  Family: gaussian
##   Links: mu = identity; sigma = identity
## Formula: change ~ hptreat
##    Data: df_powerpose (Number of observations: 39)
##   Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup draws = 4000
##
## Regression Coefficients:
##            Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
## Intercept      4.30      4.62    -4.81    13.33 1.00     3505
## hptreatLow    -8.68      6.68   -21.73     4.47 1.00     3183
##            Tail_ESS
## Intercept      2616
## hptreatLow     2668
##
## Further Distributional Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma    20.55      2.46    16.44    25.80 1.00     3407     2946
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

There does seem to be a lower testosterone level in the low power pose condition compared to the high power

pose condition; this is consistent with the hypothesis. However, the credible intervals are very wide, and it would be hard to interpret the average effect as confirming the hypothesis. A Bayes factor analysis would be needed to claim evidence for an effect, but it is unlikely that such evidence would be forthcoming (the reader should check this after reading the Bayes factor chapter).

If we want to know which priors `brms` used by default, we do the following:

```
prior_summary(fit_powerpose)
```

```
##                       prior     class       coef group resp dpar nlpar
##                     (flat)         b
##                     (flat)         b hptreatLow
##  student_t(3, -1.9, 18.7) Intercept
##     student_t(3, 0, 18.7)     sigma
##  lb ub       source
##             default
##        (vectorized)
##             default
##   0         default
```

We can inspect this even before running the model in the following way.

```
get_prior(change ~ hptreat, df_powerpose)
```

```
##                       prior     class       coef group resp dpar nlpar
##                     (flat)         b
##                     (flat)         b hptreatLow
##  student_t(3, -1.9, 18.7) Intercept
##     student_t(3, 0, 18.7)     sigma
##  lb ub       source
##             default
##        (vectorized)
##             default
##   0         default
```

Exercise 2

(a) Our priors for this experiment were quite arbitrary. How do the prior predictive distributions look like? Do they make sense?

We first create a centered version of load:

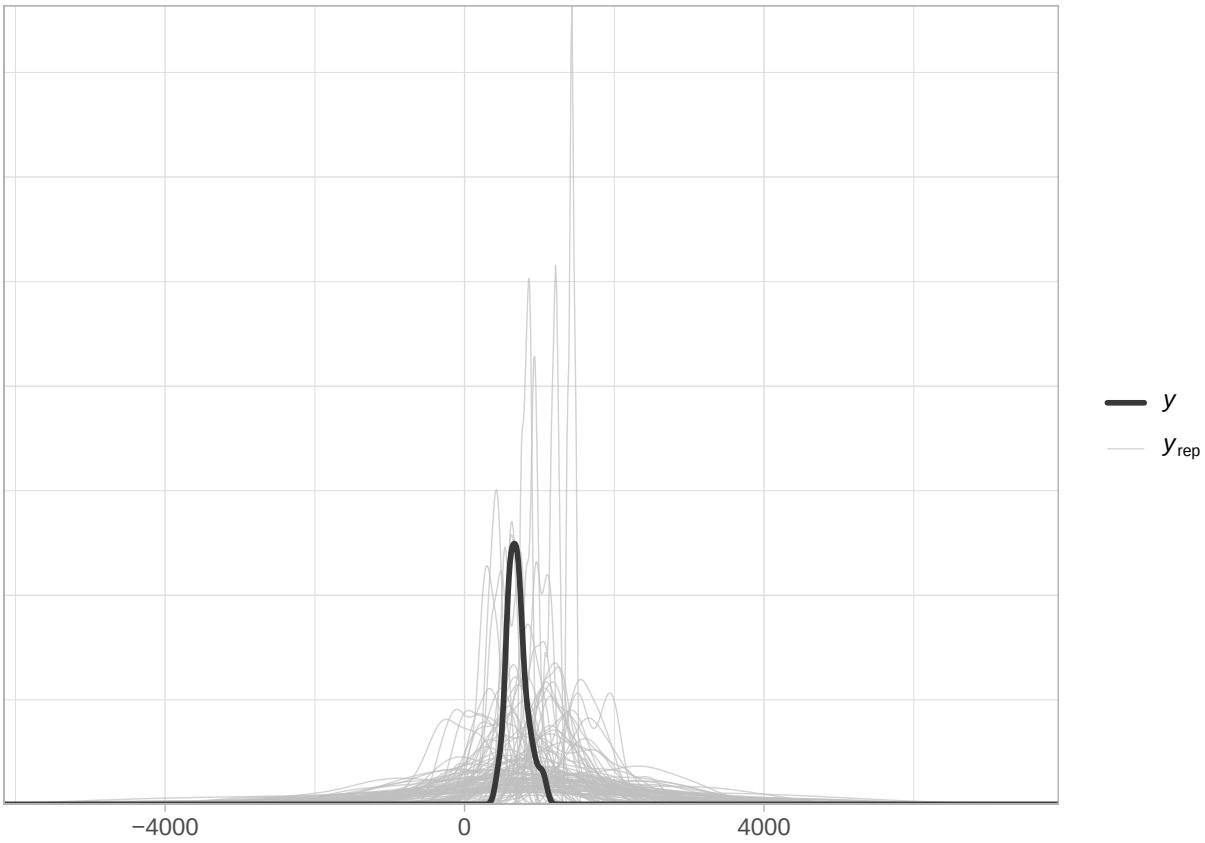
```
data("df_pupil")
df_pupil <- df_pupil %>%
  mutate(c_load = load - mean(load))
```

We first sample from the priors only:

```
fit_pupil_prior <- brm(p_size ~ 1 + c_load,
  data = df_pupil,
  family = gaussian(),
  prior = c(
    prior(normal(1000, 500), class = Intercept),
    prior(normal(0, 1000), class = sigma),
    prior(normal(0, 100), class = b, coef = c_load)
  ),
  sample_prior = "only",
```

```
  control = list(adapt_delta = .99)
)
```

```
pp_check(fit_pupil_prior, ndraws = 100)
```

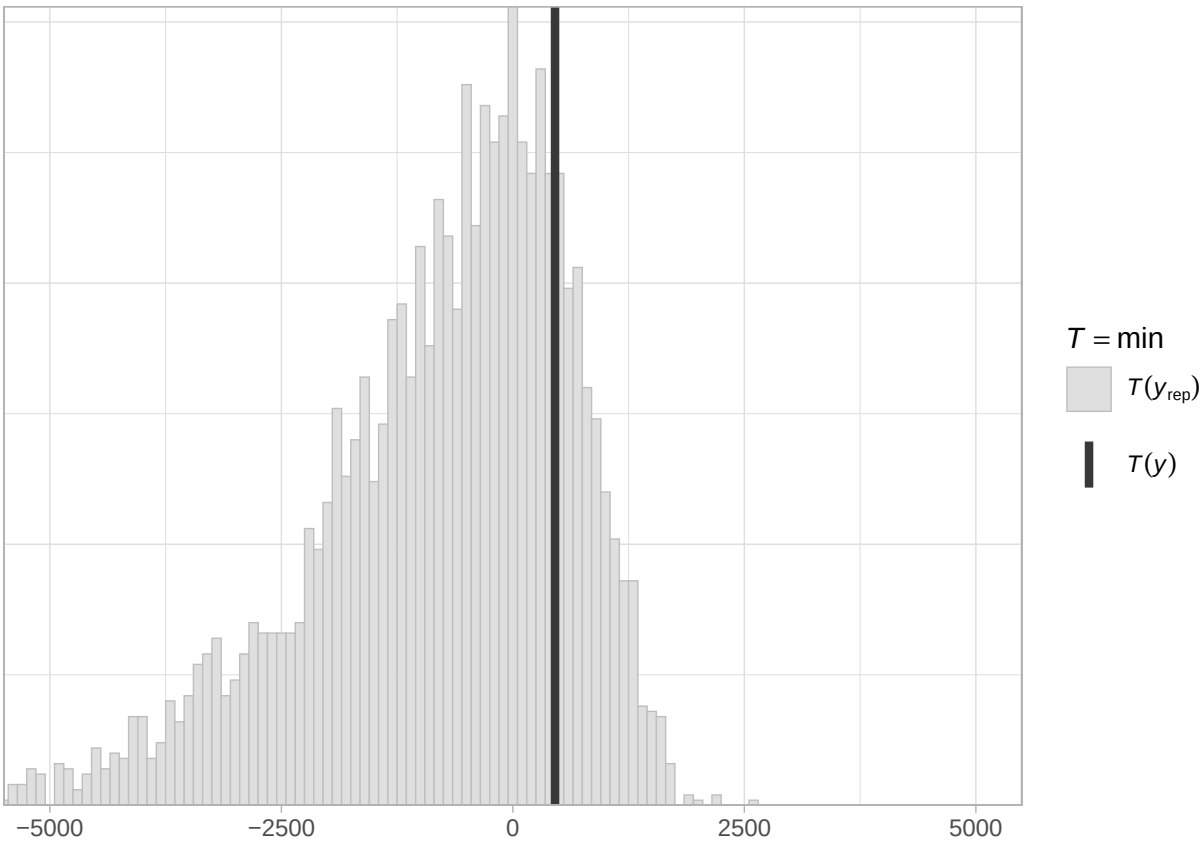


They don't look too good, we are assuming that pupil sizes can be even negative!

In fact the prior distribution of minimum values shows that we are putting too much prior probability mass on negative values of pupil size:

```
pp_check(fit_pupil_prior,
  type = "stat",
  stat = "min",
  binwidth = 100
) +
  coord_cartesian(xlim = c(-5000, 5000))
```

```
## Using all posterior draws for ppc type 'stat' by default.
```



As an extra step, let's look at our choice for σ. Our choice of prior for σ might have been too wide leading to too much variation in the prior predictive distribution.

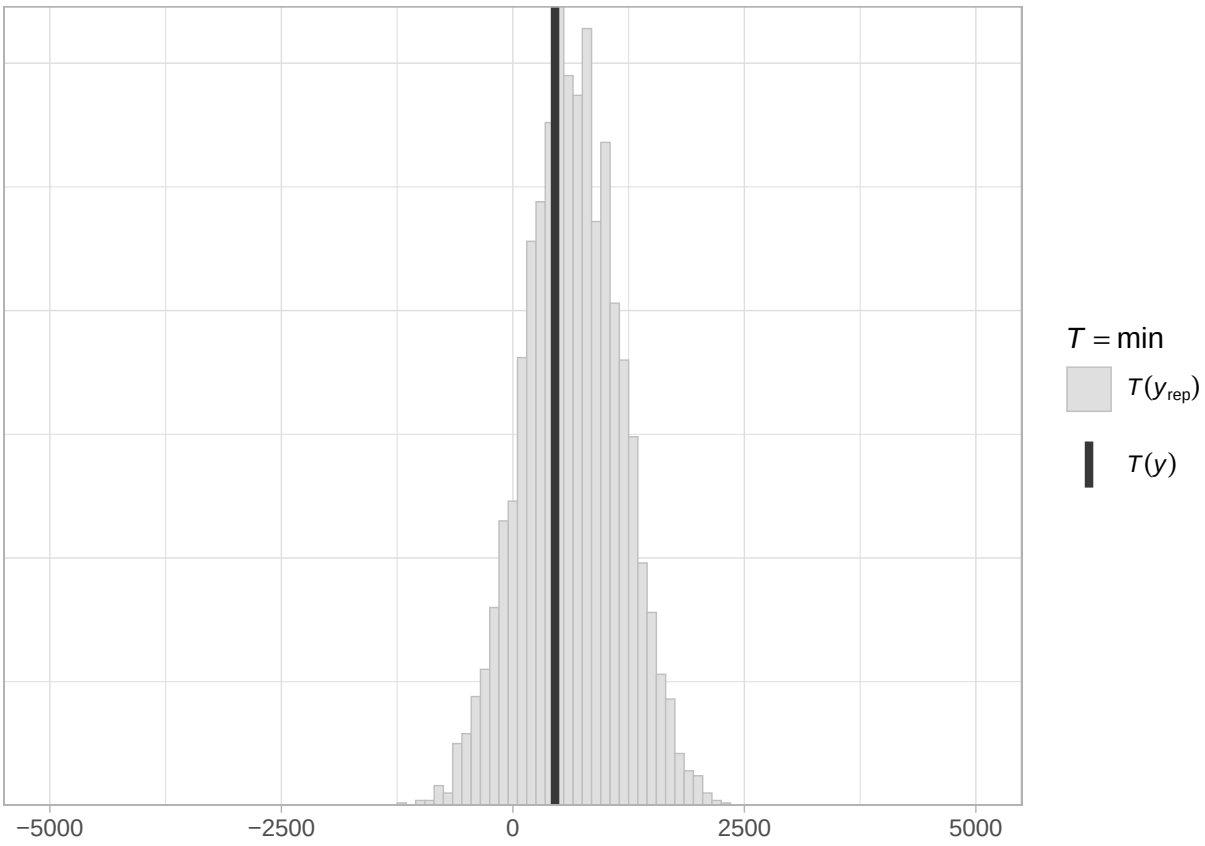
The following seems like a more reasonable prior:

$$\sigma \sim Normal_{+}(100, 50) \tag{1}$$

```
fit_pupil_sigma2_prior <- brm(p_size ~ 1 + c_load,
  data = df_pupil,
  family = gaussian(),
  prior = c(
    prior(normal(1000, 500), class = Intercept),
    prior(normal(100, 50), class = sigma),
    prior(normal(0, 100), class = b, coef = c_load)
  ),
  sample_prior = "only",
)
```

```
pp_check(fit_pupil_sigma2_prior,
  type = "stat",
  stat = "min",
  binwidth = 100
) +
  coord_cartesian(xlim = c(-5000, 5000))
```

```
## Using all posterior draws for ppc type 'stat' by default.
```



It's still not perfect, but we have reduced the probability mass of negative pupil sizes. We could continue tuning our priors and using our prior knowledge, this also might lead to a change in the likelihood. For now, we stop here.

(b) Is our posterior distribution sensitive to the priors that we selected? Perform a sensitivity analysis to find out whether the posterior is affected by our choice of prior for σ.

We'll check now if our estimates change when we select a more informative prior for σ:

$$\sigma \sim Normal_{+}(100, 50) \tag{2}$$

```
fit_pupil_sigma2 <- brm(p_size ~ 1 + c_load,
  data = df_pupil,
  family = gaussian(),
  prior = c(
    prior(normal(1000, 500), class = Intercept),
    prior(normal(100, 50), class = sigma),
    prior(normal(0, 100), class = b, coef = c_load)
  )
)
```

```
fit_pupil_sigma2
```

```
##  Family: gaussian
##   Links: mu = identity; sigma = identity
## Formula: p_size ~ 1 + c_load
```

```
##    Data: df_pupil (Number of observations: 41)
##   Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup draws = 4000
##
## Regression Coefficients:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept   701.92     19.67   663.97   739.30 1.00     3969     3116
## c_load       34.06     11.64    11.52    56.19 1.00     4130     2617
##
## Further Distributional Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma   125.66     13.46   101.85   154.74 1.00     3621     2814
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

We see that our estimates remain virtually identical. One can continue and check other potential priors for σ.

(c) Our data set includes also a column that indicates the trial number. Could it be that trial has also an effect on the pupil size? As in lm, we indicate another main effect with a + sign. How would you communicate the new results?

As we did with pupil size, we first create a centered version of trial:

```
df_pupil <- df_pupil %>% mutate(c_trial = trial - mean(trial))
```

We'll fit a model with the following likelihood:

$$p\_size_n \sim Normal(\alpha + c\_load_n \cdot \beta_1 + c\_trial_n \cdot \beta_2, \sigma) \tag{3}$$

The formula will look like this `1 + c_load + c_trial`. We can see the new priors that the model needs by doing the following:

```
get_prior(p_size ~ 1 + c_load + c_trial,
  data = df_pupil,
  family = gaussian()
)
```

```
##                       prior     class    coef group resp dpar nlpar
##                     (flat)         b
##                     (flat)         b  c_load
##                     (flat)         b c_trial
##  student_t(3, 687.1, 125.2) Intercept
##      student_t(3, 0, 125.2)     sigma
##  lb ub       source
##             default
##        (vectorized)
##        (vectorized)
##             default
##   0         default
```

For simplicity, we assign the same prior distribution to both β parameters.

```
fit_pupil_trial <- brm(p_size ~ 1 + c_load + c_trial,
  data = df_pupil,
```

```
  family = gaussian(),
  prior = c(
    prior(normal(1000, 500), class = Intercept),
    prior(normal(0, 1000), class = sigma),
    prior(normal(0, 100), class = b)
  )
)
```

```
fit_pupil_trial
```

```
##  Family: gaussian
##   Links: mu = identity; sigma = identity
## Formula: p_size ~ 1 + c_load + c_trial
##    Data: df_pupil (Number of observations: 41)
##   Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup draws = 4000
##
## Regression Coefficients:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept   701.57     16.88   667.84   735.61 1.00     4461     2896
## c_load       31.81     10.41    11.66    51.56 1.00     4379     2855
## c_trial      -5.54      1.50    -8.41    -2.52 1.00     4076     2753
##
## Further Distributional Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma   110.06     13.41    87.78   139.31 1.00     3706     2750
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

The summary of the posterior tells us that the most likely values of the effect of load will be around the mean of the posterior, 31.81, and we can be 95% certain that the true value of the effect of load (given the model and the data) lies between 11.66 and 51.56. As before, as load increases the pupil size increases. The mean of the posterior for the effect of trial will be -5.54, with a 95% credible interval of $[-8.41, -2.52]$; this is telling us that as the trials proceed further, the pupil size is reduced.

Exercise 3

(a) Estimate the slowdown in milliseconds between the last two times the subject pressed the space bar in the experiment.

We look at what happens between the last two trials in milliseconds in the following way:

```
alpha_samples <- as_draws_df(fit_press_trial)$b_Intercept
beta_samples <- as_draws_df(fit_press_trial)$b_c_trial
last_trial <- df_spacebar$c_trial %>% max()
effect_end_ms <- exp(alpha_samples + last_trial * beta_samples) -
  exp(alpha_samples + (last_trial - 1) * beta_samples)
c(mean = mean(effect_end_ms), quantile(effect_end_ms, c(.025, .975)))
```

```
##   mean   2.5%  97.5%
## 0.0964 0.0715 0.1230
```

(b)

We first create the new predictors:

```
df_spacebar <- df_spacebar %>%
  mutate(
    c_log_trial = log(trial) - mean(log(trial)),
    c_sqrt_trial = sqrt(trial) - mean(sqrt(trial))
  )
```

We start with log trial. The slope (β) now represents the change in log response times when we move from the centered log transformed trial (`c_log_trial`) 0 to 1, which corresponds to going from trial 1 to trial "2.718" (because $\exp(0) = 1$ and $\exp(1) = 2.718$). However the scale is not linear, and thus when we move from log transformed trials 2 to 3, it represents going from trial "7.389" to trial "20.086" (because $\exp(2) = 7.389$ and $\exp(3) = 20.086$). This means that our prior for β from section @ref(sec-trial) might be too restrictive, one unit for the new β can be much more than one trial. We change the prior for β in the next model to $Normal(0, 1)$:

```
fit_press_log_trial <- brm(t ~ 1 + c_log_trial,
  data = df_spacebar,
  family = lognormal(),
  prior = c(
    prior(normal(6, 1.5), class = Intercept),
    prior(normal(0, 1), class = sigma),
    prior(normal(0, 1), class = b, coef = c_log_trial)
  )
)
```

We will look at the effect in milliseconds for the middle of the experiment. Thus we need to know `c_log_trial` at the middle and one trial before in our experiment. (Notice that it won't be 0 and -1 as before).

```
middle_log <- df_spacebar %>%
  filter(c_trial == 0) %>%
  pull(c_log_trial)
## in base R this would be  <- df_spacebar[c_trial == 0]$c_log_trial
middle_m1_log <- df_spacebar %>%
  filter(c_trial == -1) %>%
  pull(c_log_trial)
```

```
alpha_samples <- as_draws_df(fit_press_log_trial)$b_Intercept
beta_samples <- as_draws_df(fit_press_log_trial)$b_c_log_trial
effect_middle_ms <- exp(alpha_samples + middle_log * beta_samples) -
  exp(alpha_samples + middle_m1_log * beta_samples)
c(mean = mean(effect_middle_ms), quantile(effect_middle_ms, c(.025, .975)))
```

```
##    mean    2.5%   97.5%
## 0.0589 0.0467 0.0715
```

We can do the same for square-root-transformed trials

```
fit_press_sqrt_trial <- brm(t ~ 1 + c_sqrt_trial,
  data = df_spacebar,
  family = lognormal(),
  prior = c(
    prior(normal(6, 1.5), class = Intercept),
    prior(normal(0, 1), class = sigma),
    prior(normal(0, 1), class = b, coef = c_sqrt_trial)
  )
```

```
)
```

```
middle_sqrt <- df_spacebar %>%
  filter(c_trial == 0) %>%
  pull(c_sqrt_trial)
middle_m1_sqrt <- df_spacebar %>%
  filter(c_trial == -1) %>%
  pull(c_sqrt_trial)
```

```
alpha_samples <- as_draws_df(fit_press_sqrt_trial)$b_Intercept
beta_samples <- as_draws_df(fit_press_sqrt_trial)$b_c_sqrt_trial
effect_middle_ms <- exp(alpha_samples + middle_sqrt * beta_samples) -
  exp(alpha_samples + middle_m1_sqrt * beta_samples)
c(mean = mean(effect_middle_ms), quantile(effect_middle_ms, c(.025, .975)))
```

```
##   mean   2.5%  97.5%
## 0.0824 0.0645 0.0997
```

Let's also look at the descriptive adequacy of these three models. You cannot easily combine models with bayesplot (which in turn is what `brms` uses to plot). See https://github.com/stan-dev/bayesplot/issues/232

This requires a bit of R knowledge. (Another alternative is to just have 3 plots).

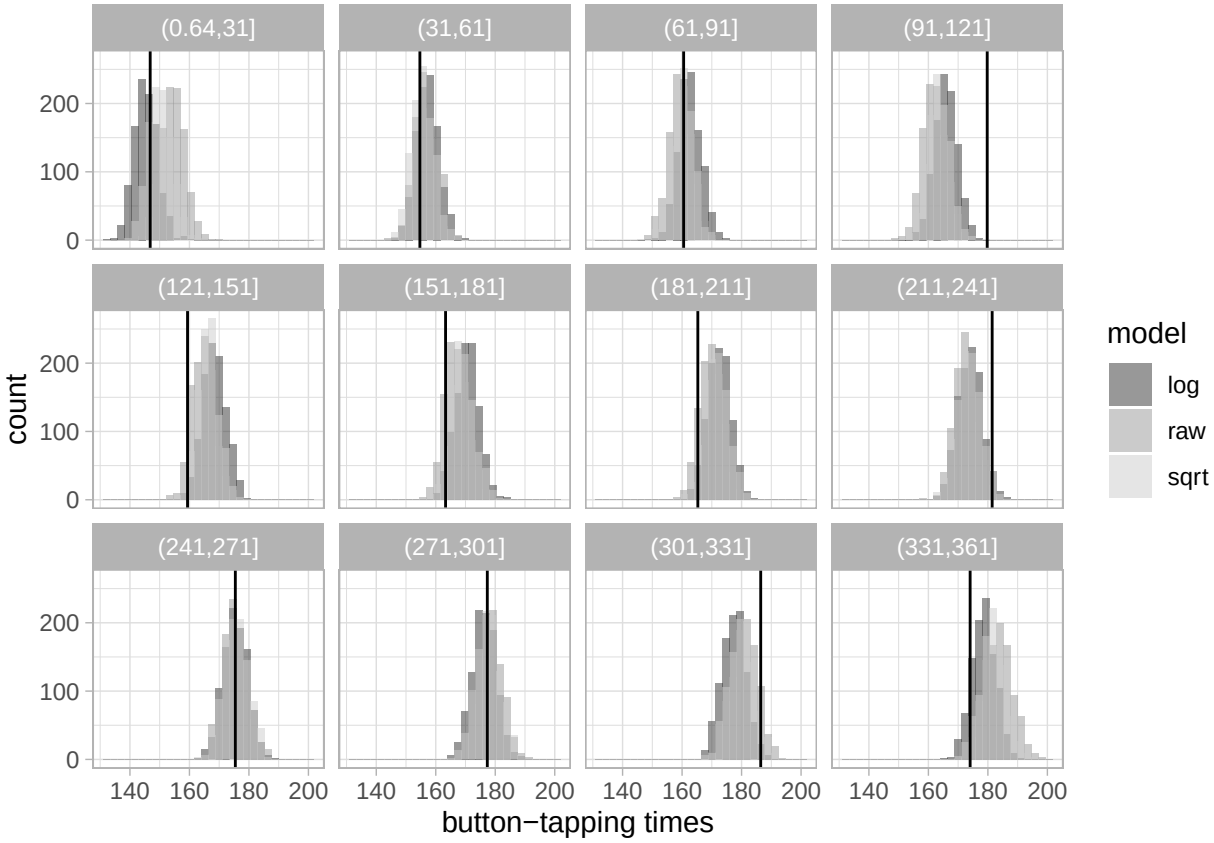
```
# I use imap to iterate across the fit of the three models,
# and create a long df with predictions of the three models
df_noreading_data_pred <-
  imap_dfr(
    list(raw = fit_press_trial, log = fit_press_log_trial, sqrt = fit_press_sqrt_trial),
    function(fit, model) {
      posterior_predict(fit, ndraws = 1000) %>%
        array_branch(margin = 1) %>%
        map_dfr(function(yrep_iter) {
          df_spacebar %>%
            mutate(
              rt = yrep_iter,
              model = model
            )
        }, .id = "iter") %>%
        mutate(iter = as.numeric(iter))
    }
  )
```

We see that for the range of trials that we have in our experiment, the three models make very similar predictions:

```
df_noreading_pred_summary <- df_noreading_data_pred %>%
  # I create 12 intervals of trials
  mutate(inter = cut(trial, breaks = 12)) %>%
  group_by(model, iter, inter) %>%
  summarize(rt = mean(rt))

# observed means:
df_noreading_summary <- df_spacebar %>%
  mutate(inter = cut(trial, breaks = 12)) %>%
  group_by(inter) %>%
  summarize(rt = mean(t))
```

```
ggplot(df_noreading_pred_summary, aes(rt)) +
  geom_histogram(alpha = 0.5, aes(fill = model), position = "identity") +
  geom_vline(aes(xintercept = rt), data = df_noreading_summary) +
  xlab("button-tapping times") +
  facet_wrap(inter ~ .)
```



Exercise 4

```
data("df_recall")
df_recall <- df_recall %>%
  mutate(c_set_size = set_size - mean(set_size),
         c_tested = tested - mean(tested))

fit_recall <- brm(correct ~ 1 + c_set_size + c_tested,
  data = df_recall,
  family = bernoulli(link = logit),
  prior = c(
    prior(normal(0, 1.5), class = Intercept),
    #same prior for both parameters
    prior(normal(0, .1), class = b)
  )
)
```

```
fit_recall
```

```
##  Family: bernoulli
```

```
##   Links: mu = logit
## Formula: correct ~ 1 + c_set_size + c_tested
##    Data: df_recall (Number of observations: 92)
##   Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup draws = 4000
##
## Regression Coefficients:
##            Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
## Intercept      1.93      0.31     1.37     2.57 1.00     3800
## c_set_size    -0.18      0.08    -0.34    -0.01 1.00     3134
## c_tested      -0.03      0.08    -0.20     0.13 1.00     3065
##            Tail_ESS
## Intercept      3088
## c_set_size     2724
## c_tested       3005
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

Exercise 5

```
data("df_red")
head(df_red)
```

```
##    risk age red pink redorpink
## 8     0  19   0    0         0
## 9     0  25   0    0         0
## 10    0  20   0    0         0
## 11    0  20   0    0         0
## 14    0  20   0    0         0
## 15    0  18   0    0         0
```

We fit three different models. We use similar priors as before.

```
fit_red <- brm(red ~ 1 + risk,
  data = df_red,
  family = bernoulli(link = logit),
  prior = c(
    prior(normal(0, 1.5), class = Intercept),
    prior(normal(0, 0.1), class = b, coef= risk)
  )
)
```

```
fit_red
```

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: red ~ 1 + risk
##    Data: df_red (Number of observations: 124)
##   Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup draws = 4000
##
## Regression Coefficients:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
```

```
## Intercept    -2.60      0.34    -3.30    -1.98 1.00     2941     2344
## risk          0.02      0.10    -0.18     0.22 1.00     2857     2614
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
fit_pink <- brm(pink ~ 1 + risk,
  data = df_red,
  family = bernoulli(link = logit),
  prior = c(
    prior(normal(0, 1.5), class = Intercept),
    prior(normal(0, 0.1), class = b)
  )
)
```

```
fit_pink
```

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: pink ~ 1 + risk
##    Data: df_red (Number of observations: 124)
##   Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup draws = 4000
##
## Regression Coefficients:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept    -2.04      0.28    -2.60    -1.53 1.00     3765     2517
## risk          0.04      0.10    -0.15     0.23 1.00     4052     2917
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
fit_redorpink <- brm(redorpink ~ 1 + risk,
  data = df_red,
  family = bernoulli(link = logit),
  prior = c(
    prior(normal(0, 1.5), class = Intercept),
    prior(normal(0, .1), class = b)
  )
)
```

```
fit_redorpink
```

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: redorpink ~ 1 + risk
##    Data: df_red (Number of observations: 124)
##   Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup draws = 4000
##
## Regression Coefficients:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept    -1.54      0.24    -2.02    -1.08 1.00     3500     2693
## risk          0.06      0.10    -0.14     0.25 1.00     3777     2691
```

```
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

The slope parameter's posterior in the models doesn't seem to be consistent with the theory. However, to establish whether the effect "exists" or not, one would need to do a formal hypothesis test (Bayes factor).