Introduction to Bayesian Data Analysis
Computational Bayesian Analysis (Chapter 4 of book)

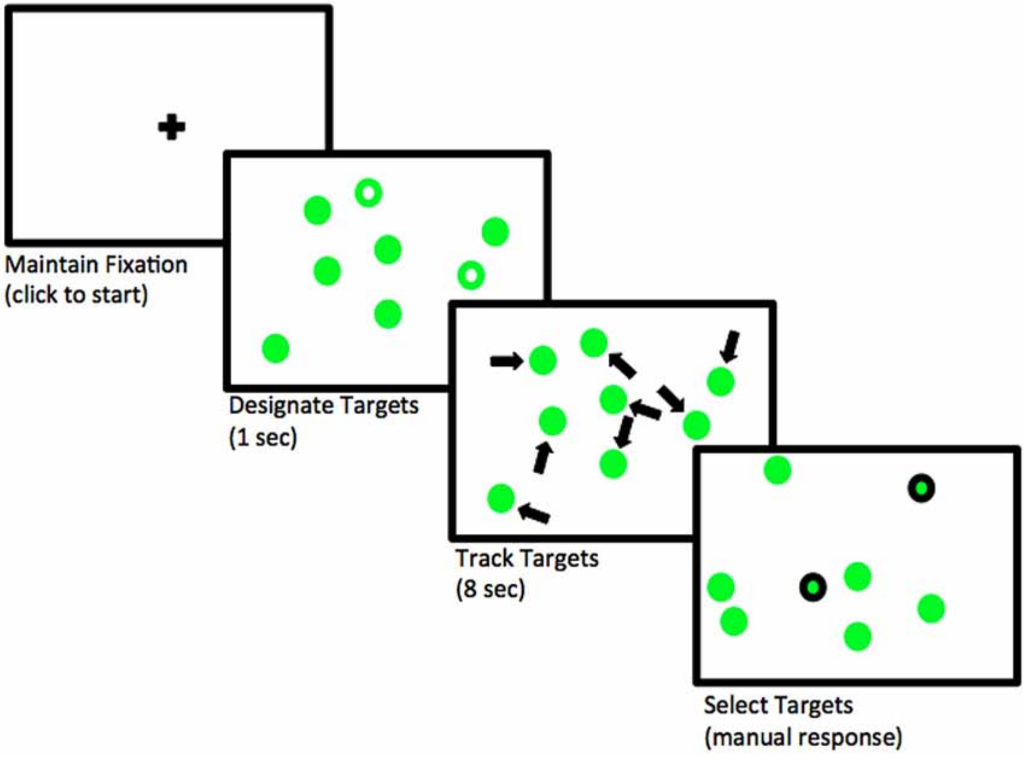
Prof. Dr. Shravan Vasishth
Professor, Linguistics
Cognitive Science / Linguistics, Uni Potsdam, Germany

Example: Multiple object tracking

- The subject covertly tracks between zero and five objects among several randomly moving objects on a computer screen.
- First, several objects appear on the screen, and a subset of them are indicated as "targets" at the beginning.
- Then, the objects start moving randomly across the screen and become indistinguishable.
- After several seconds, the objects stop moving and the subject need to indicate which objects were the targets.

Our research goal is to examine **how the attentional load affects pupil size**.

Example: Multiple object tracking



Example: Multiple object tracking

A model for this experiment design:

$$p\_size_n \sim Normal(\alpha + c\_load_n \cdot \beta, \sigma) \tag{1}$$

- n indicates the observation number with $n = 1, \ldots, N$
- c_load refers to centered load.
- Every data point is assumed to be independent (in frequentist terms: iid).

Example: Multiple object tracking: Prior specification

Some pilot data helps us work out priors:

```
data("df_pupil_pilot")
df_pupil_pilot$p_size %>% summary()

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     852     856     862     861     866     868
```

This suggests we can use the following regularizing prior for α:

$$\alpha \sim Normal(1000, 500) \tag{2}$$

What we are expressing with this prior:

```
qnorm(c(.025, .975), mean = 1000, sd = 500)

## [1]   20 1980
```

Example: Multiple object tracking: Prior specification

For σ, we use an uninformative prior:

$$\sigma \sim Normal_{+}(0, 1000) \tag{3}$$

```
extraDistr::qtnorm(c(.025,0.975), mean = 0, sd = 1000, a = 0)

## [1]   31 2241
```

Example: Multiple object tracking: Prior specification

$$\beta \sim Normal(0, 100) \tag{4}$$

```r
qnorm(c(.025, .975), mean = 0, sd = 100)

## [1] -196  196
```

Example: Multiple object tracking: Fit model

First, center the predictor:

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

```
## # A tibble: 41 x 5
##      subj trial  load p_size c_load
##     <int> <int> <int>  <dbl>  <dbl>
##  1   701     1     2  1021. -0.439
##  2   701     2     1   951. -1.44
##  3   701     3     5  1064.  2.56
##  4   701     4     4   913.  1.56
##  5   701     5     0   603. -2.44
##  6   701     6     3   826.  0.561
##  7   701     7     0   464. -2.44
```

Example: Multiple object tracking: Fit model

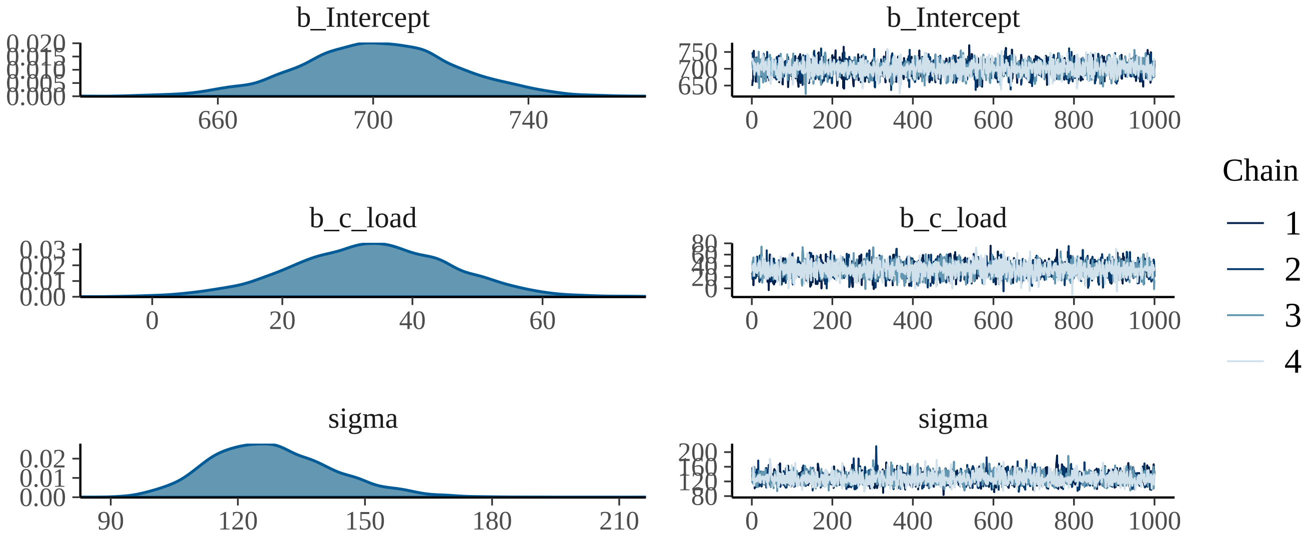
```
fit_pupil <- brm(p_size ~ 1 + c_load,
  data = df_pupil,
  family = gaussian(),
  prior = c(
    prior(normal(1000, 500), class = Intercept),
    prior(normal(0, 1000), class = sigma),
    prior(normal(0, 100), class = b, coef = c_load)
  )
)
```

The next steps

Next, we will plot the posterior distributions of the parameters, and the posterior predictive distributions for the different load levels.

Example: Multiple object tracking: Summarize posteriors

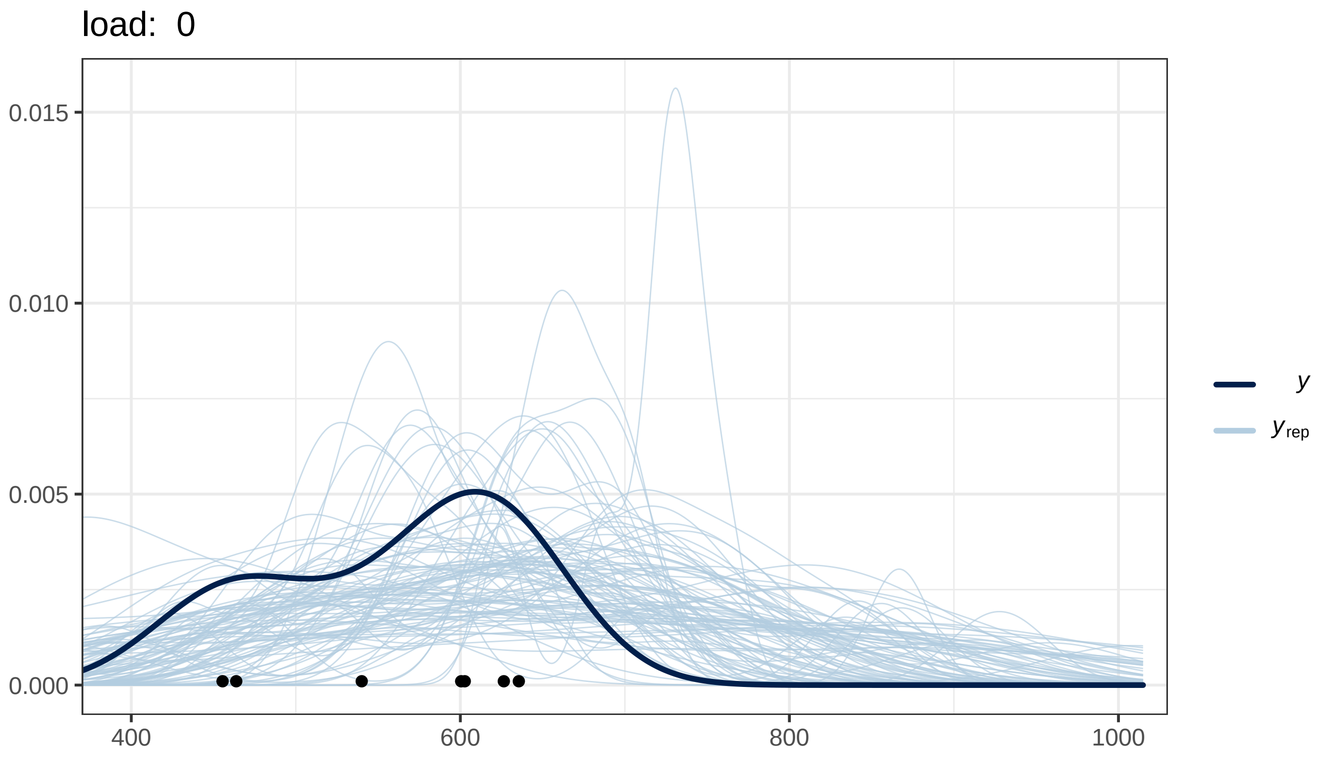
```r
plot(fit_pupil)
```



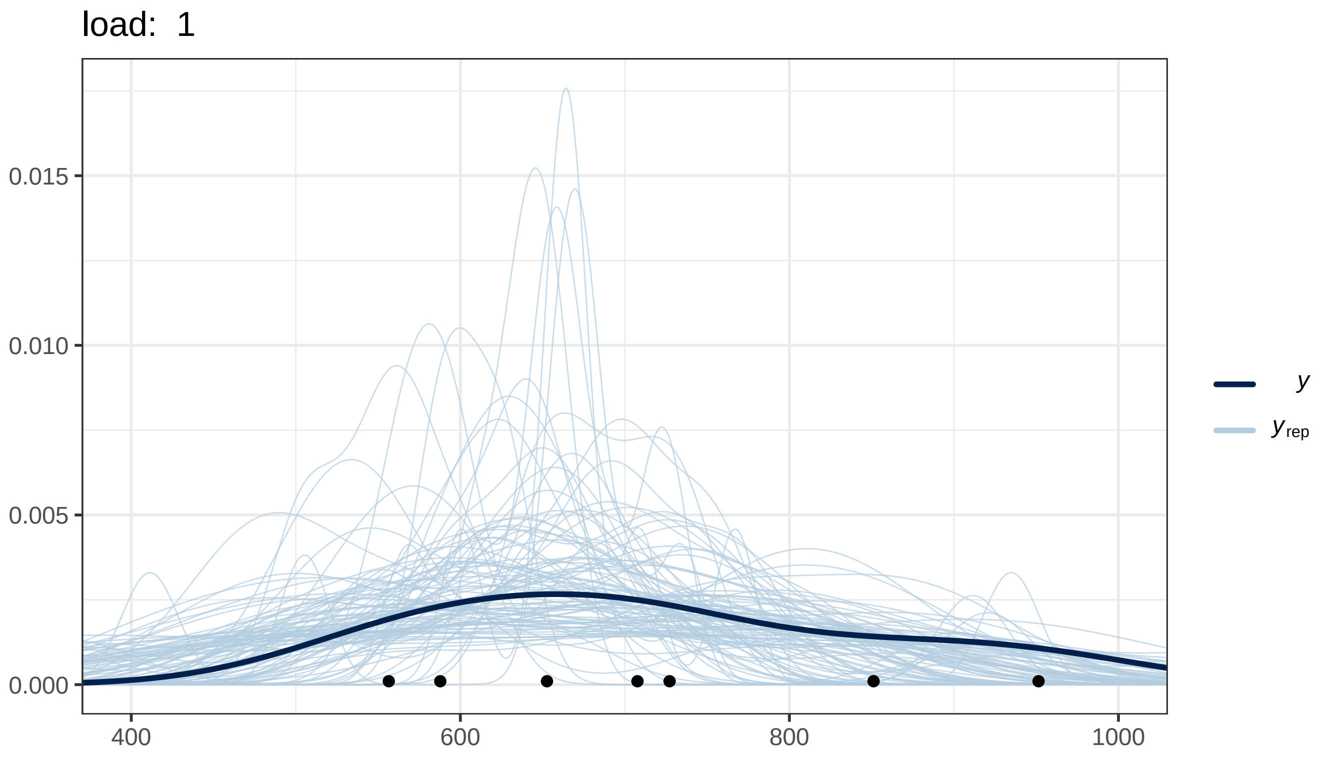
Example: Multiple object tracking: Summarize posteriors

```
## Note: short_summary is a function we wrote
short_summary(fit_pupil)

## ...
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept   701.06     20.00   660.47   739.65 1.00     3761     2844
## c_load       33.85     12.06     9.78    57.08 1.00     3639     2859
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma   128.19     14.64   102.94   159.66 1.00     3751     2937
##
## ...
```

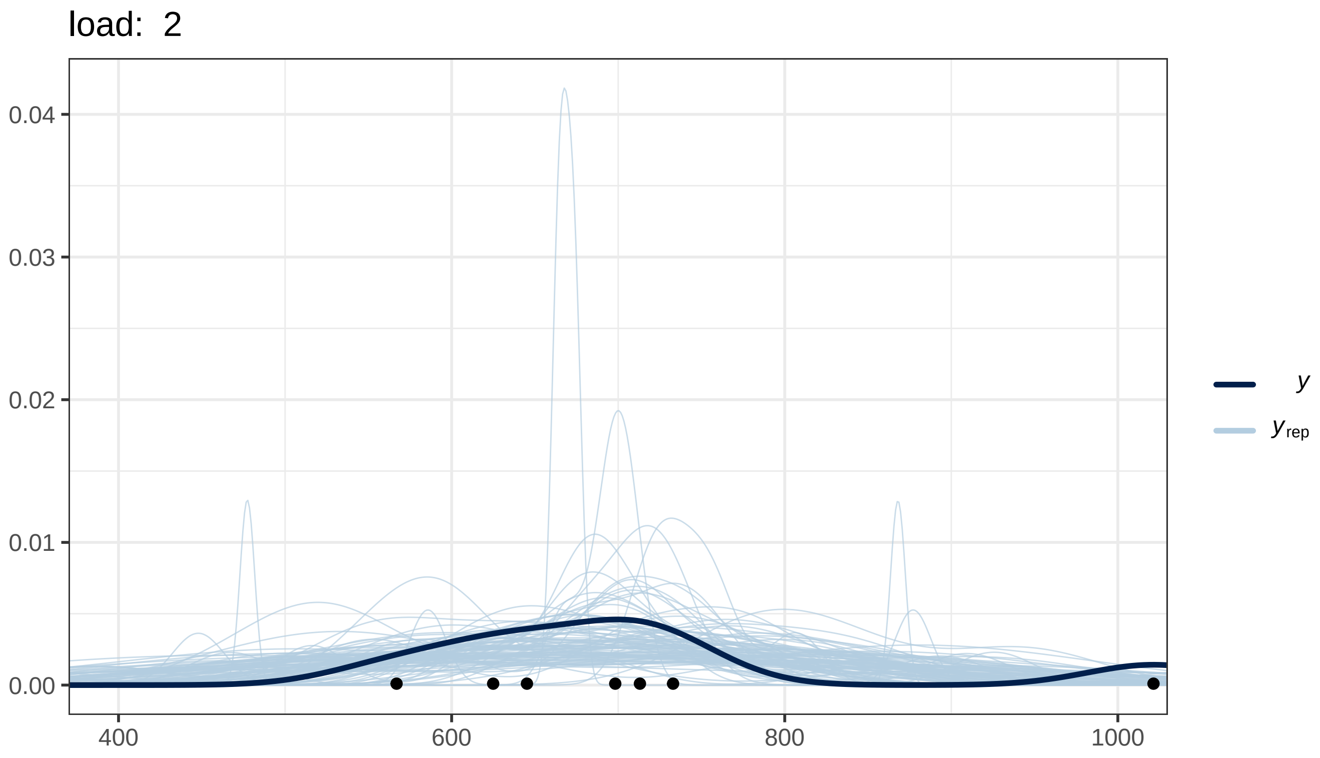
Example: Multiple object tracking: Posterior predictive check



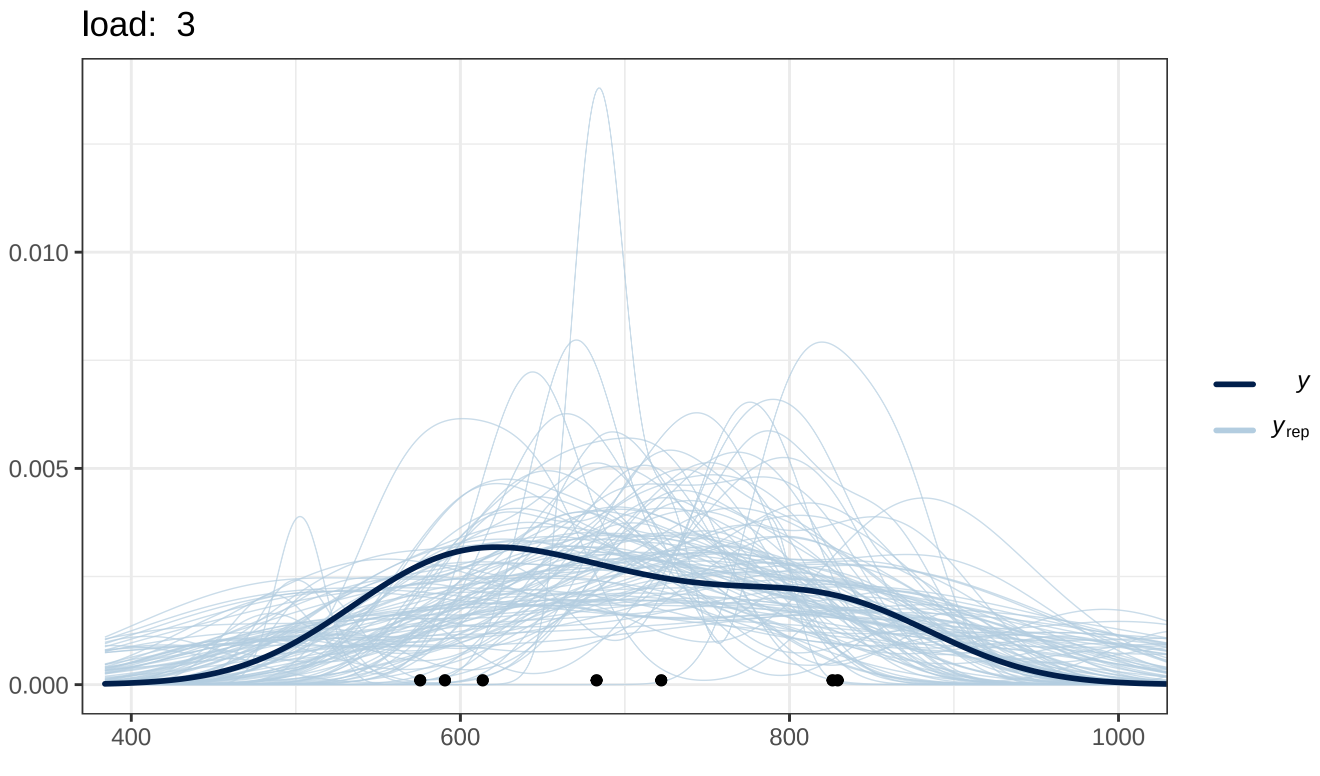
Example: Multiple object tracking: Posterior predictive check



Example: Multiple object tracking: Posterior predictive check



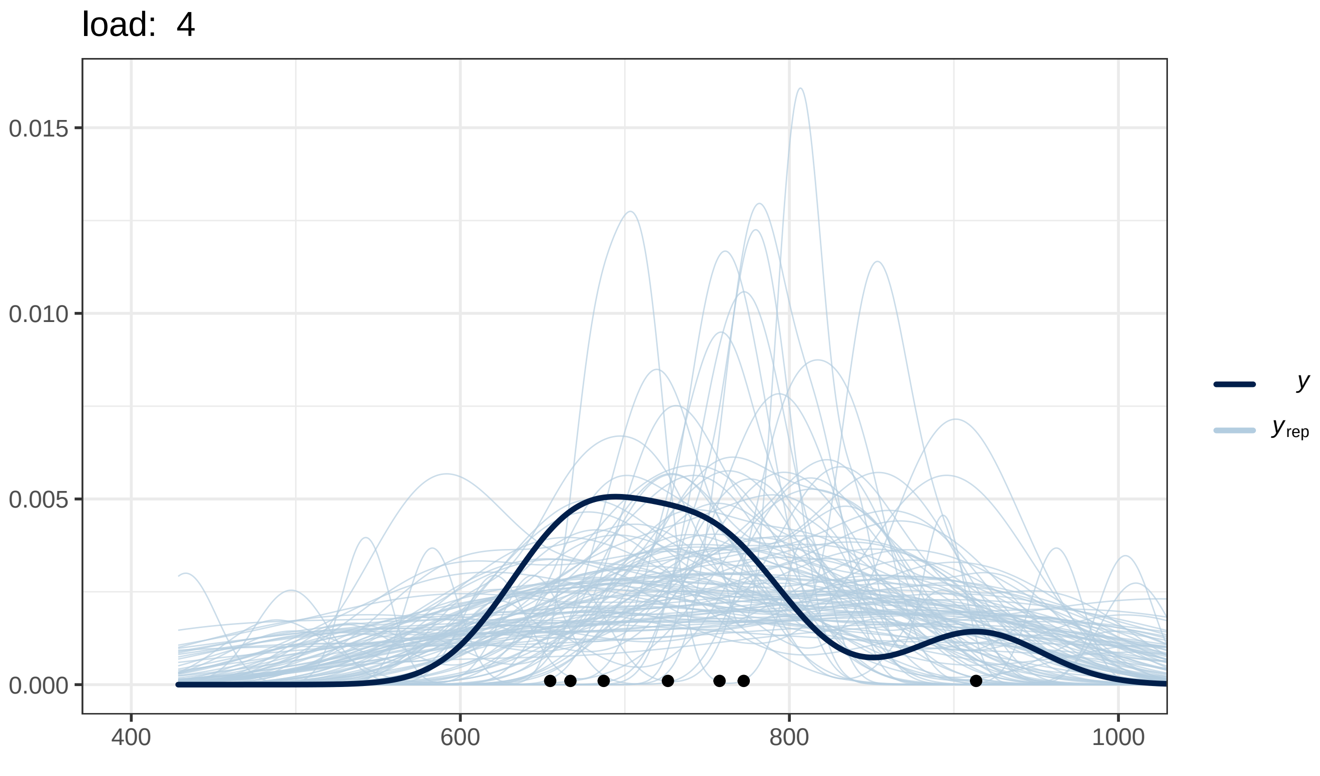
Example: Multiple object tracking: Posterior predictive check



Bayesian Data Analysis

Shravan Vasishth
vasishth.github.io

Example: Multiple object tracking: Posterior predictive check



The next steps

Next, we will look at another example: the effect of trial id on button-pressing times.
This time, we will use the log-normal likelihood.

Example: Button-pressing time and the effect of trial

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

Example: Button-pressing time and the effect of trial

If we assume that response times are log-normally distributed, we could proceed as follows:

$$rt_n \sim LogNormal(\alpha + c\_trial_n \cdot \beta, \sigma) \tag{5}$$

where

- N is the total number of (independent!) data points
- $n = 1, \ldots, N$, and
- rt is the dependent variable (response times in milliseconds).

Example: Button-pressing time and the effect of trial

The priors have to be defined on the log scale:

$$
\begin{aligned}
\alpha &\sim Normal(6, 1.5) \\
\sigma &\sim Normal_{+}(0, 1)
\end{aligned}
\tag{6}
$$

A new parameter, β, needs a prior specification:

$$
\beta \sim Normal(0, 1) \tag{7}
$$

This prior on β is very uninformative.

Example: Button-pressing time and the effect of trial

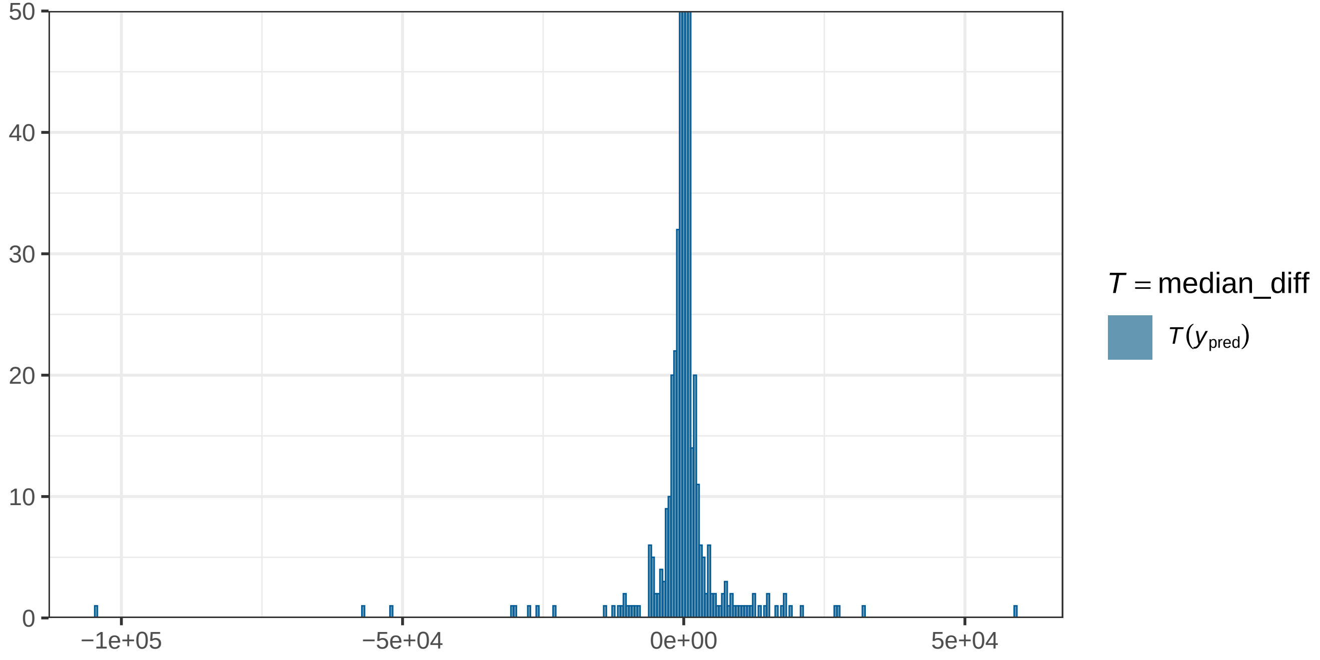
Prior predictive distribution:

```
df_spacebar_ref <- df_spacebar %>%
  mutate(rt = rep(1, n()))
fit_prior_press_trial <- brm(t ~ 1 + c_trial,
  data = df_spacebar_ref,
  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_trial)
  ),
  sample_prior = "only",
  control = list(adapt_delta = .9)
)
```

Example: Button-pressing time and the effect of trial

```
median_diff <- function(x) {
  median(x - lag(x), na.rm = TRUE)
}
pp_check(fit_prior_press_trial,
         type = "stat",
         stat = "median_diff",
  # show only prior predictive distributions
         prefix = "ppd",
  # each bin has a width of 500ms
         binwidth = 500) +
  # cut the top of the plot to improve its scale
  coord_cartesian(ylim = c(0, 50))+theme_bw()
```

Example: Button-pressing time and the effect of trial



Example: Button-pressing time and the effect of trial

What would the prior predictive distribution look like if we set the following more informative prior on β?

$$\beta \sim Normal(0, 0.01) \tag{8}$$

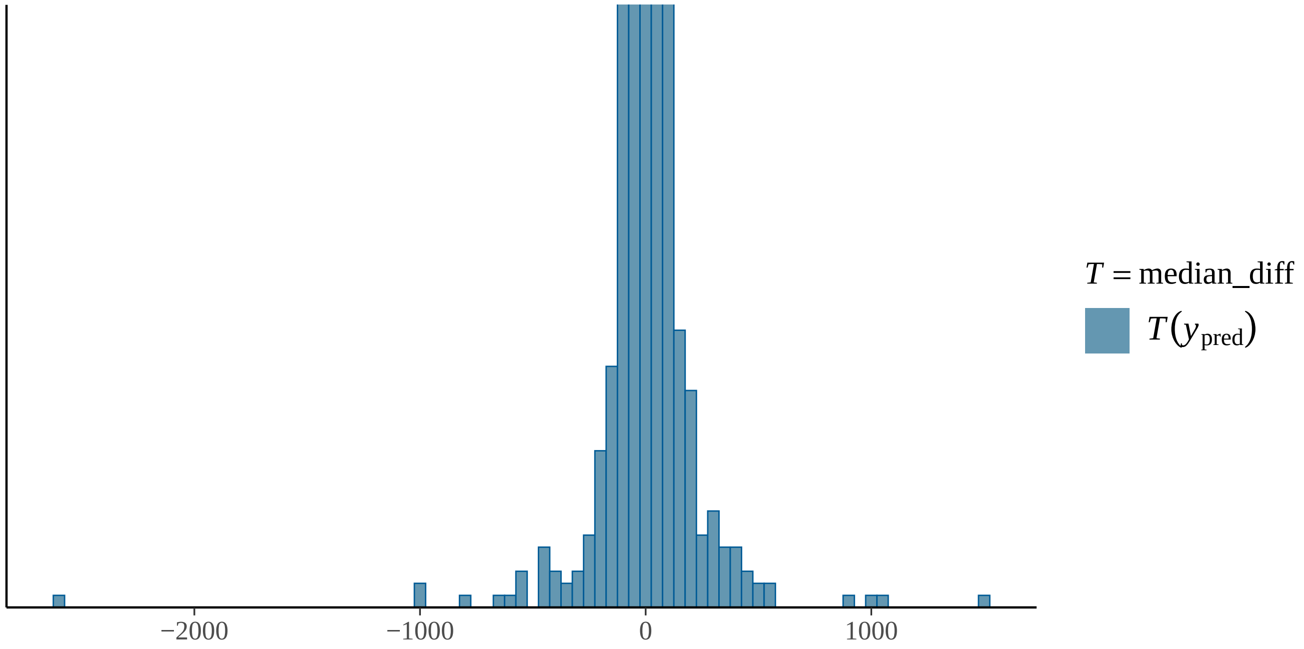
Example: Button-pressing time and the effect of trial

```r
fit_prior_press_trial <- brm(t ~ 1 + c_trial,
  data = df_spacebar_ref,
  family = lognormal(),
  prior = c(
    prior(normal(6, 1.5), class = Intercept),
    prior(normal(0, 1), class = sigma),
    prior(normal(0, .01), class = b, coef = c_trial)
  ),
  sample_prior = "only",
  control = list(adapt_delta = .9)
)
```

Example: Button-pressing time and the effect of trial

```
fit_prior_press_trial <- brm(t ~ 1 + c_trial,
  data = df_spacebar_ref,
  family = lognormal(),
  prior = c(
    prior(normal(6, 1.5), class = Intercept),
    prior(normal(0, 1), class = sigma),
    prior(normal(0, .01), class = b, coef = c_trial)
  ),
  sample_prior = "only",
  control = list(adapt_delta = .9)
)
```

Example: Button-pressing time and the effect of trial



The next steps

Now that we have decided on our priors, we fit the model.

Example: Button-pressing time and the effect of trial

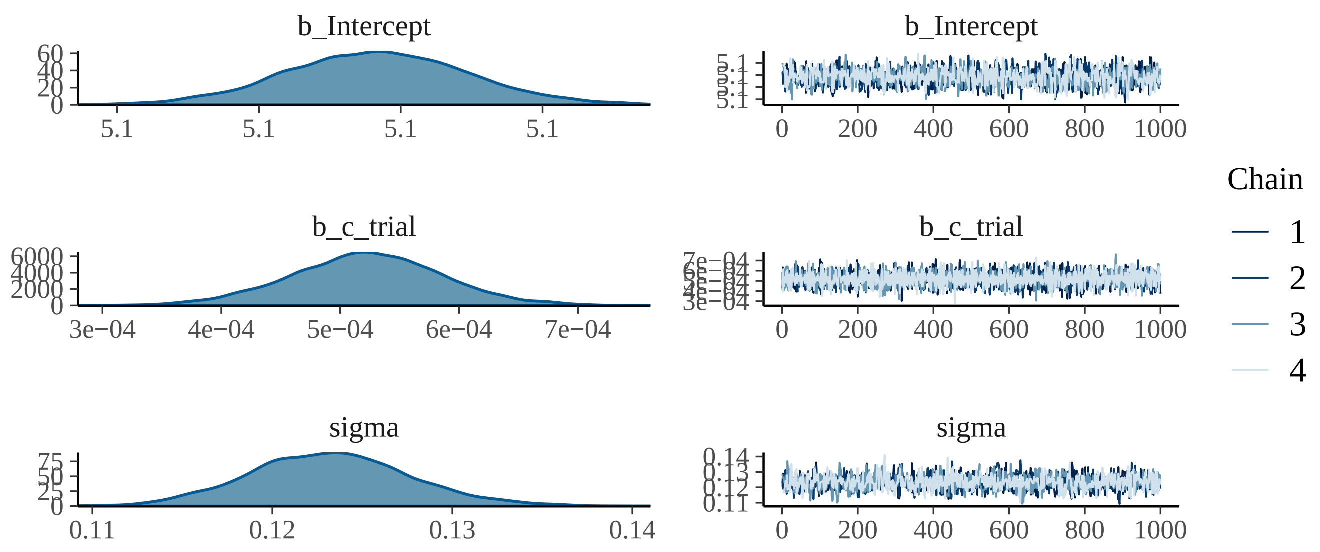
Fit the model:

```
fit_press_trial <- brm(t ~ 1 + c_trial,
  data = df_spacebar,
  family = lognormal(),
  prior = c(
    prior(normal(6, 1.5), class = Intercept),
    prior(normal(0, 1), class = sigma),
    prior(normal(0, .01), class = b, coef = c_trial)
  )
)
```

Example: Button-pressing time and the effect of trial

Summarize posteriors (graphically or in a table, or both):

```r
plot(fit_press_trial)
```



Example: Button-pressing time and the effect of trial

Summarize results on the ms scale (the effect estimate from the middle of the expt to the preceding trial):

```
alpha_samples <- as_draws_df(fit_press_trial)$b_Intercept
beta_samples <- as_draws_df(fit_press_trial)$b_c_trial

beta_ms <- exp(alpha_samples) - exp(alpha_samples - beta_samples)

beta_msmean <- round(mean(beta_ms), 5)
beta_mslow <- round(quantile(beta_ms, prob = 0.025), 5)
beta_mshigh <- round(quantile(beta_ms, prob = 0.975), 5)
c(beta_msmean , beta_mslow, beta_mshigh)
```

```
##        2.5% 97.5%
## 0.087 0.066 0.108
```

Example: Button-pressing time and the effect of trial

The effect estimate at the first vs second trial:

```
first_trial <- min(df_spacebar$c_trial)
second_trial <- min(df_spacebar$c_trial) + 1
effect_beginning_ms <-
  exp(alpha_samples + second_trial * beta_samples) -
  exp(alpha_samples + first_trial * beta_samples)
## ms effect from first to second trial:
c(mean = mean(effect_beginning_ms),
  quantile(effect_beginning_ms, c(0.025, 0.975)))
```

```
##  mean  2.5% 97.5%
## 0.079 0.062 0.096
```

Example: Button-pressing time and the effect of trial

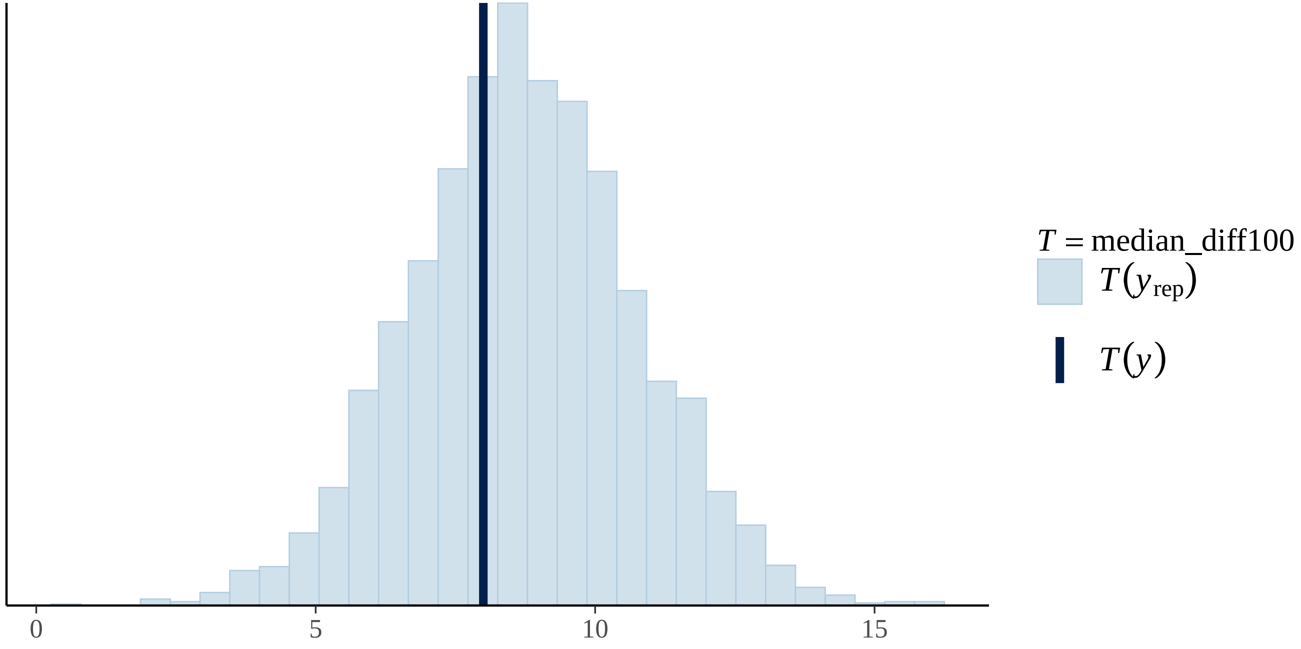
Slowdown after 100 trials from the middle of the expt:

```
effect_100 <-
  exp(alpha_samples + 100 * beta_samples) -
  exp(alpha_samples)
c(mean = mean(effect_100),
  quantile(effect_100, c(0.025, 0.975)))
```

```
##  mean  2.5% 97.5%
##   9.0   6.7  11.2
```

Example: Button-pressing time and the effect of trial

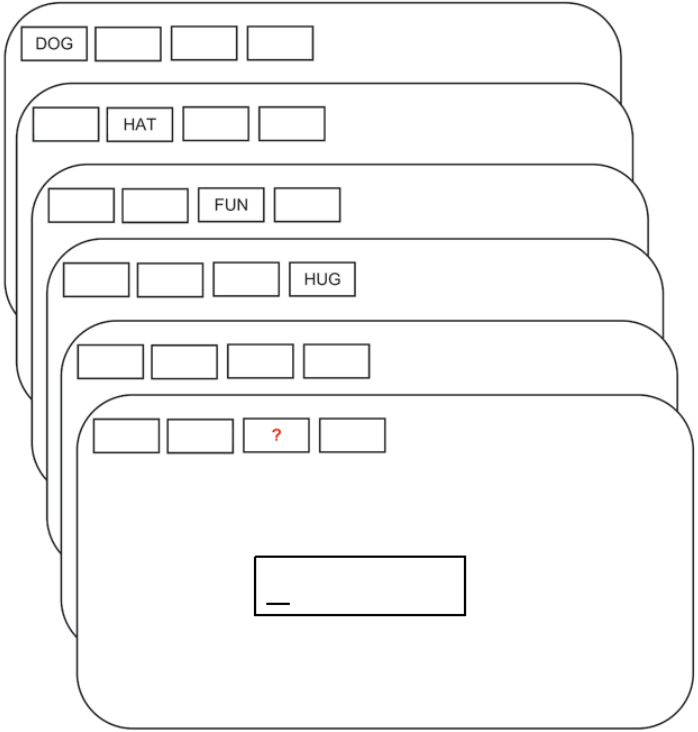
The posterior predictive distribution (distribution of predicted median differences between the n and n-100th trial):



The next steps

Next: logistic regression.

Example of logistic regression: Does set size affect free recall?



Example of logistic regression: Does set size affect free recall?

```
data("df_recall")
head(df_recall)

## # A tibble: 6 x 7
##   subj  set_size correct trial session block tested
##   <chr>    <int>   <int> <int>   <int> <int>  <int>
## 1 10           4       1     1       1     1      2
## 2 10           8       0     4       1     1      8
## 3 10           2       1     9       1     1      2
## 4 10           6       1    23       1     1      2
## 5 10           4       1     5       1     2      3
## 6 10           8       0     7       1     2      5

df_recall <- df_recall %>%
  mutate(c_set_size = set_size - mean(set_size))
```

Example of logistic regression: Does set size affect free recall?

```r
# Set sizes in the data set:
df_recall$set_size %>%
  unique() %>% sort()
```

```
## [1] 2 4 6 8
```

Example of logistic regression: Does set size affect free recall?

```
# Trials by set size
df_recall %>%
  group_by(set_size) %>%
  count()
```

```
## # A tibble: 4 x 2
## # Groups:   set_size [4]
##   set_size     n
##      <int> <int>
## 1        2    23
## 2        4    23
## 3        6    23
## 4        8    23
```

Example of logistic regression: Does set size affect free recall?

$$correct_n \sim Bernoulli(\theta_n) \tag{9}$$

$$\eta_n = g(\theta_n) = \log\left(\frac{\theta_n}{1-\theta_n}\right) \tag{10}$$

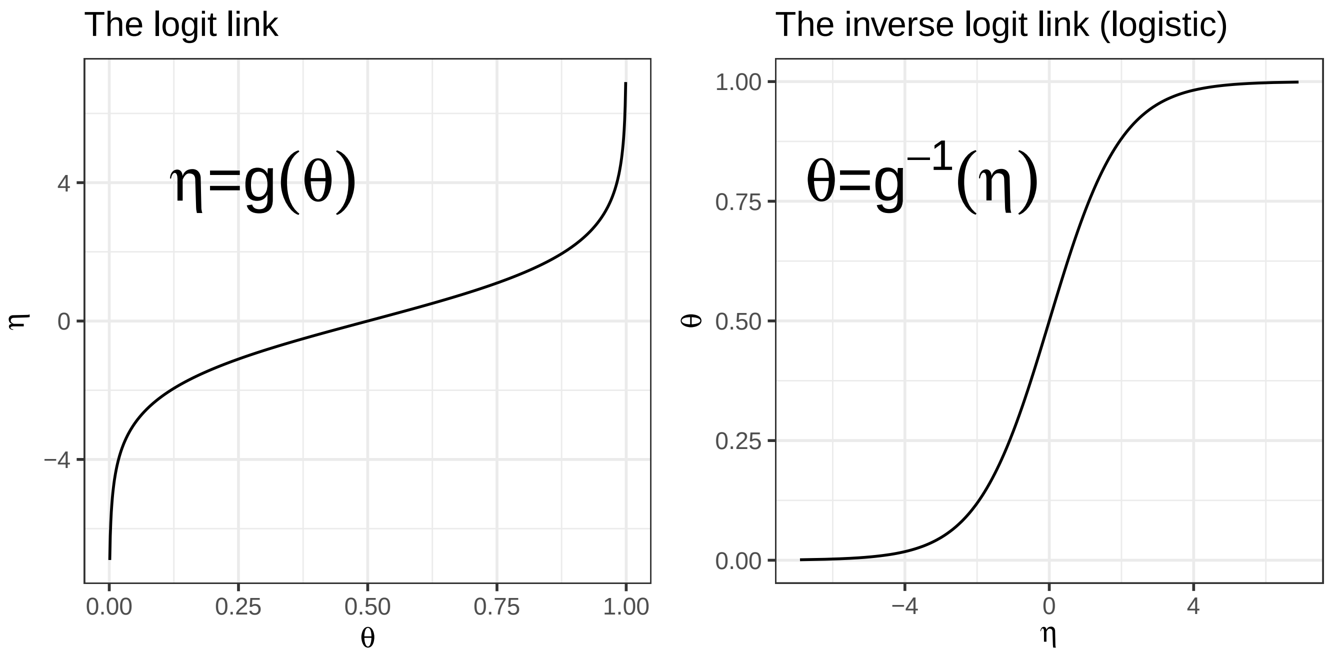
Example of logistic regression: Does set size affect free recall?

```
x <- seq(0.001, 0.999, by = 0.001)
y <- log(x / (1 - x))
logistic_dat <- data.frame(theta = x, eta = y)

p1 <- qplot(logistic_dat$theta, logistic_dat$eta, geom = "line") +
  xlab(expression(theta)) +
  ylab(expression(eta)) +
  ggtitle("The logit link") +
  annotate("text",
    x = 0.3, y = 4,
    label = expression(paste(eta, "=",
                             g(theta))), parse = TRUE,
    size = 8
  )
```

Example of logistic regression: Does set size affect free recall?

```
p2 <- qplot(logistic_dat$eta, logistic_dat$theta,
            geom = "line") + xlab(expression(eta)) +
  ylab(expression(theta)) +
  ggtitle("The inverse logit link (logistic)") +
  annotate("text",
  x = -3.5, y = 0.80,
  label = expression(paste(theta, "=", g^-1,
                           (eta))), parse = TRUE, size = 8
)

gridExtra::grid.arrange(p1, p2, ncol = 2)
```

Example of logistic regression: Does set size affect free recall?



The next steps

Next step: deciding on priors.

Example of logistic regression: Does set size affect free recall?

```
data("df_recall")
head(df_recall)
```

```
## # A tibble: 6 x 7
##   subj  set_size correct trial session block tested
##   <chr>    <int>   <int> <int>   <int> <int>  <int>
## 1 10           4       1     1       1     1      2
## 2 10           8       0     4       1     1      8
## 3 10           2       1     9       1     1      2
## 4 10           6       1    23       1     1      2
## 5 10           4       1     5       1     2      3
## 6 10           8       0     7       1     2      5
```

```
df_recall <- df_recall %>%
  mutate(c_set_size = set_size - mean(set_size))
```

Example of logistic regression: Does set size affect free recall?

The linear model is now fit not to the 0,1 responses as the dependent variable, but to η_n, i.e., log-odds, as the dependent variable:

$$\eta_n = \log\left(\frac{\theta_n}{1-\theta_n}\right) = \alpha + \beta \cdot c\_set\_size \tag{11}$$

Example of logistic regression: Does set size affect free recall?

- Unlike the linear models, the model is defined so that there is no residual error term (ε) in this model.
- Once η_n is estimated, one can solve the above equation for θ_n (in other words, we compute the inverse of the logit function and obtain the estimates on the probability scale).

This gives the above-mentioned logistic regression function:

$$\theta_n = g^{-1}(\eta_n) = \frac{\exp(\eta_n)}{1+\exp(\eta_n)} = \frac{1}{1+exp(-\eta_n)} \tag{12}$$

Example of logistic regression: Does set size affect free recall?

In summary, the generalized linear model with the logit link fits the following Bernoulli likelihood:

$$correct_n \sim Bernoulli(\theta_n) \tag{13}$$

- The model is fit on the log-odds scale, $\eta_n = \alpha + c\_set\_size_n \cdot \beta$.
- Once η_n has been estimated, the inverse logit or the logistic function is used to compute the probability estimates $\theta_n = \frac{\exp(\eta_n)}{1+\exp(\eta_n)}$.

Example of logistic regression: Does set size affect free recall?

There are two functions in R that implement the logit and inverse logit functions:

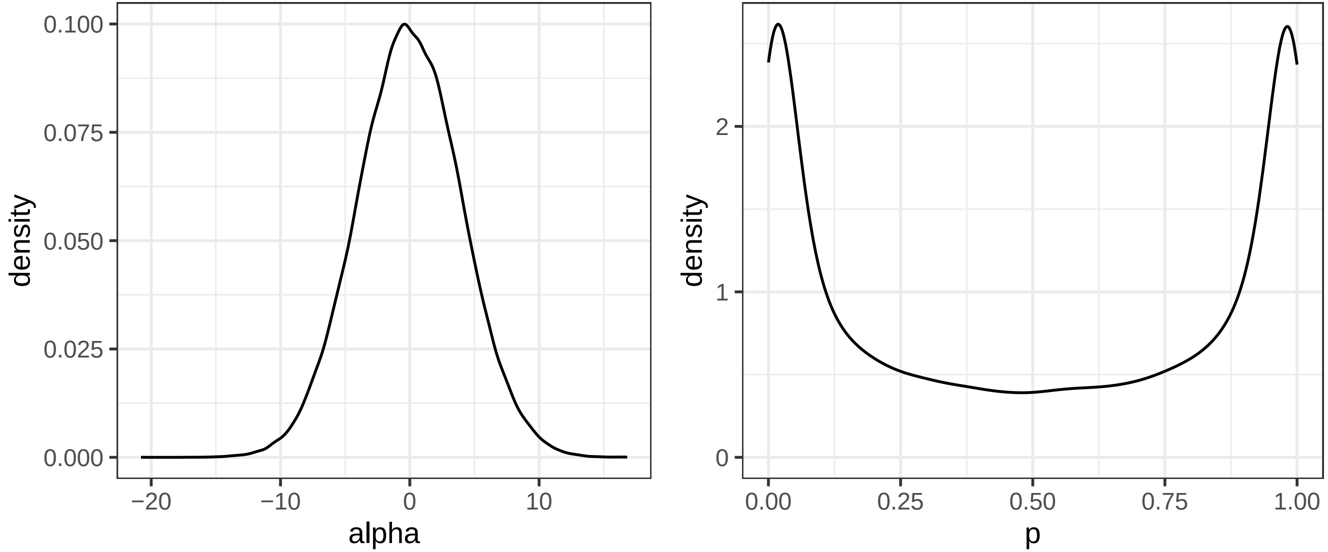
- `qlogis(p)` for the logit function and
- `plogis(x)` for the inverse logit or logistic function.

$$\alpha \sim Normal(0, 4) \tag{14}$$

Let's plot this prior in log-odds and in probability scale by drawing random samples.

Example of logistic regression: Does set size affect free recall?

Prior for $\alpha \sim Normal(0, 4)$ in log-odds and in probability space.

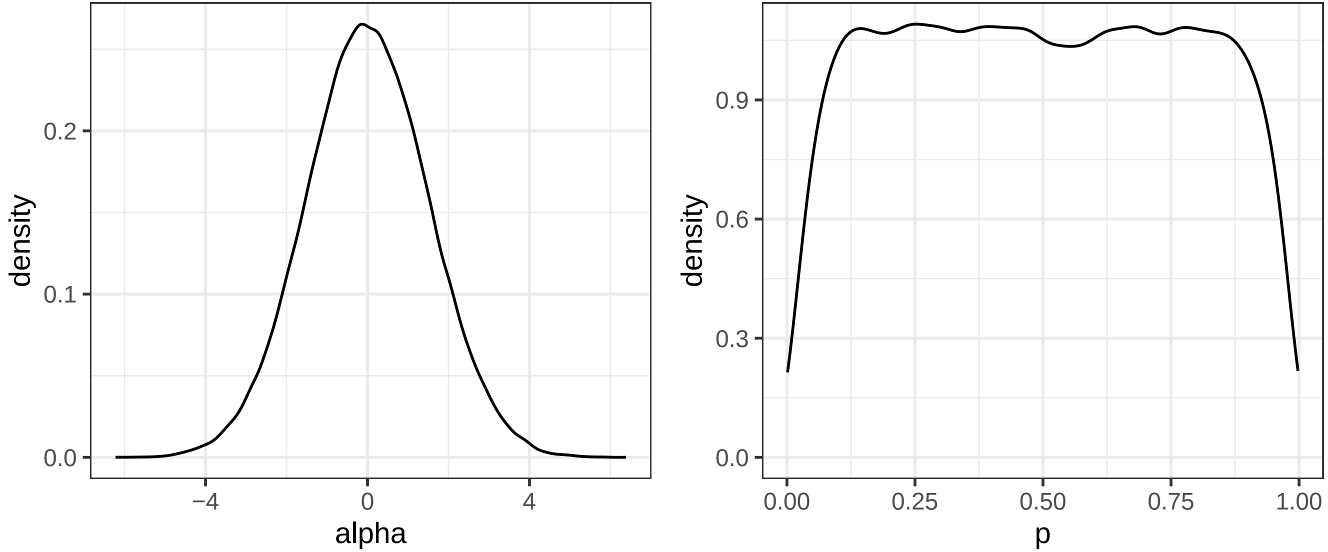


Example of logistic regression: Does set size affect free recall?

$$\alpha \sim Normal(0, 1.5) \tag{15}$$

Example of logistic regression: Does set size affect free recall?

Prior for $\alpha \sim Normal(0, 1.5)$ in log-odds and in probability space.



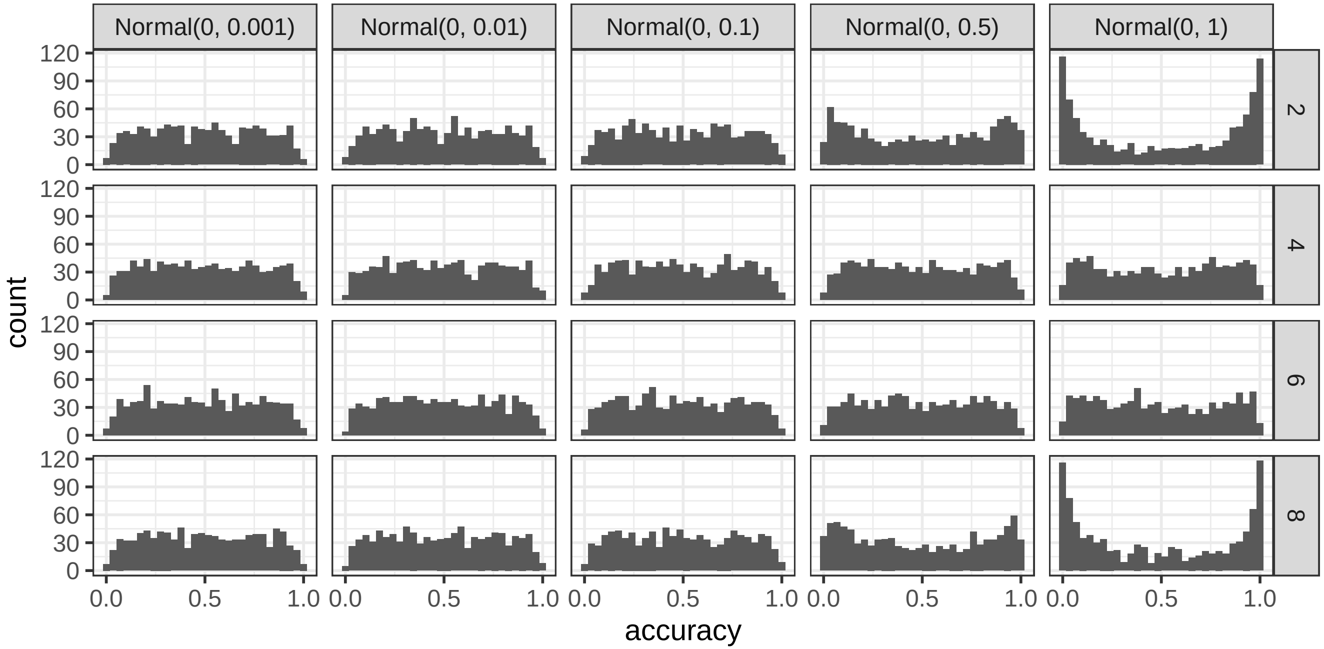
Example of logistic regression: Does set size affect free recall?

We can examine the consequences of each of the following prior specifications:

1. $\beta \sim Normal(0, 1)$
2. $\beta \sim Normal(0, .5)$
3. $\beta \sim Normal(0, .1)$
4. $\beta \sim Normal(0, .01)$
5. $\beta \sim Normal(0, .001)$

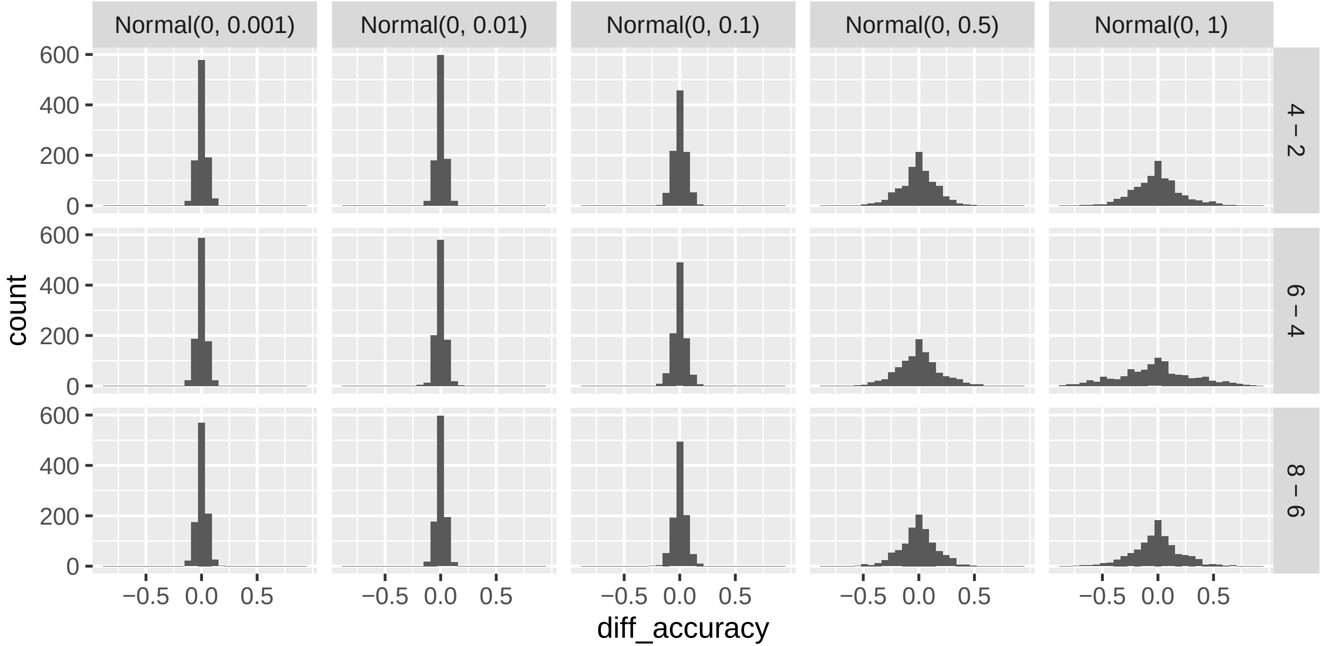
(The R code is in the textbook, chapter 4!)

Example of logistic regression: Does set size affect free recall?



Example of logistic regression: Does set size affect free recall?

It's usually more useful to look at the predicted differences in accuracy between set sizes.



Bayesian Data Analysis

Shravan Vasishth
vasishth.github.io

Example of logistic regression: Does set size affect free recall?

These priors seem reasonable:

$$\begin{aligned} \alpha &\sim Normal(0, 1.5) \\ \beta &\sim Normal(0, 0.1) \end{aligned} \tag{16}$$

The next steps

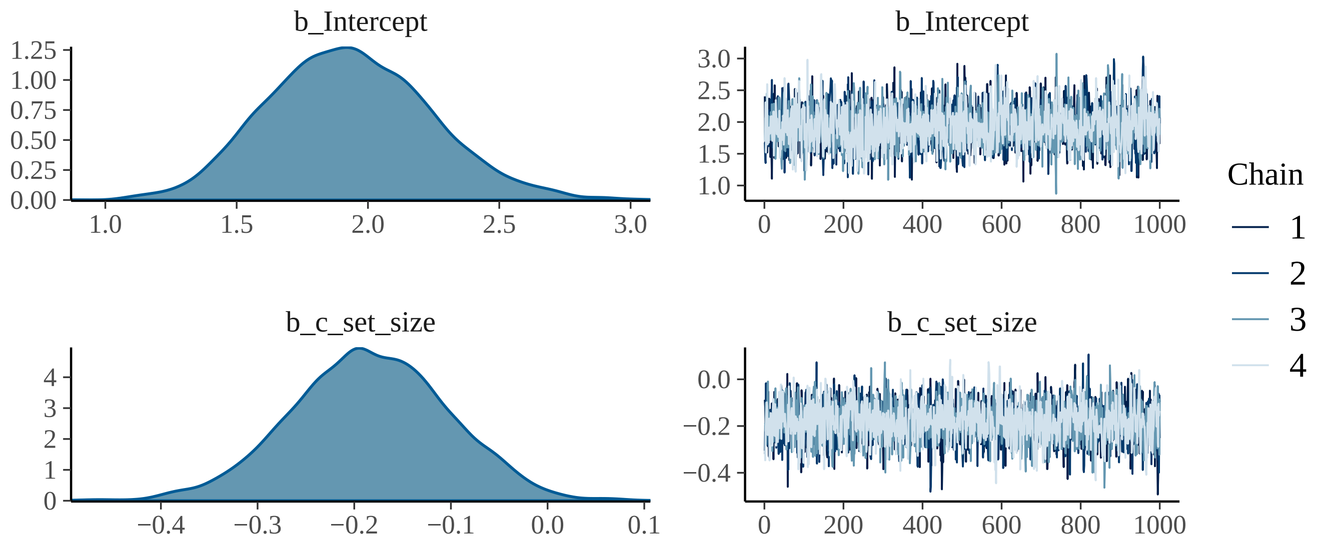
Next: fit the model and examine the posterior distributions of the parameters.

Example of logistic regression: Does set size affect free recall?

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

posterior_summary(fit_recall,
                  variable = c("b_Intercept", "b_c_set_size"))
```

Example of logistic regression: Does set size affect free recall?

```
plot(fit_recall)
```



Example of logistic regression: Does set size affect free recall?

```
alpha_samples <- as_draws_df(fit_recall)$b_Intercept
beta_samples <- as_draws_df(fit_recall)$b_c_set_size
beta_mean <- round(mean(beta_samples), 5)
beta_low <- round(quantile(beta_samples, prob = 0.025), 5)
beta_high <- round(quantile(beta_samples, prob = 0.975), 5)

alpha_samples <- as_draws_df(fit_recall)$b_Intercept
av_accuracy <- plogis(alpha_samples)
c(mean = mean(av_accuracy), quantile(av_accuracy, c(0.025, 0.975)))
```

```
##  mean  2.5% 97.5%
##  0.87  0.80  0.93
```

Example of logistic regression: Does set size affect free recall?

Find out the decrease in accuracy in proportions or probability scale:

```
beta_samples <- as_draws_df(fit_recall)$b_c_set_size
effect_middle <- plogis(alpha_samples) -
  plogis(alpha_samples - beta_samples)
c(mean = mean(effect_middle),
  quantile(effect_middle, c(0.025, 0.975)))
```

```
##    mean    2.5%   97.5%
## -0.0188 -0.0370 -0.0033
```

Example of logistic regression: Does set size affect free recall?

```
four <- 4 - mean(df_recall$set_size)
two <- 2 - mean(df_recall$set_size)
effect_4m2 <-
  plogis(alpha_samples + four * beta_samples) -
  plogis(alpha_samples + two * beta_samples)
c(mean = mean(effect_4m2),
  quantile(effect_4m2, c(0.025, 0.975)))
```

```
##    mean    2.5%   97.5%
## -0.0294 -0.0535 -0.0063
```

The next steps

Next: Hierarchical models.