

# STAT 656: Bayesian Data Analysis

## Fall 2024

### Homework 3

Juanwu Lu\*

## Gibbs Sampling for Normal Hierarchical Models

---

### The Stroop Effect

Consider a psychological task where subjects are presented with a word at the center of a computer screen ('red', 'blue', or 'green'). Further, the word is colored either red, blue or green. In some trials, the word matches the color of the text ('congruent' condition); otherwise they do not match ('incongruent' condition, *e.g.*, the word is 'red' but it is colored blue). Subjects are told to focus only on the color that the word is written in, and press 1 if the color is red, 2 if it is blue, and 3 if it is green. In each case, the experimenter measures the reaction time (*i.e.*, how long it takes them to press the correct button). The Stroop effect is a robust effect in psychology where the reaction time in the incongruent condition is on average **larger** than in the congruent condition.

Your task is to use the data in `stroop_data.csv` to verify if this is the case. The data measures multiple reactions times of different subjects in congruent and incongruent settings. You will model this with a hierarchical Bayesian model, with the goal of determining:

- how much longer reaction times are for each color in incongruent vs congruent cases, and whether this difference is significant.
- how different the effect is for each color, and whether these differences are significant.
- how different individuals in the study are from each other.

Your model should account for the fact that

- each response of each individual involves random fluctuations
- reaction times and the magnitude of the Stroop effect can be different for different individuals
- reaction times and the magnitude of the Stroop effect can be different for different colors (*e.g.*, it might be smaller for red where you have to press 1 vs others)

Your hierarchical model should allow statistical sharing among individuals and possibly among different colors. Justify your model and prior choices, implement it in **Stan** and discuss your findings, being sure to include visualizations and predictive checks of model fit. You must present your final results in a form that can be readily understood by a general audience.

**Solution:**

```
# Read the data
if (file.exists("data/stroop_dataset.csv")) {
  data <- read.csv("data/stroop_dataset.csv", header = TRUE, row.names = 1)
```

---

\*College of Engineering, Purdue University, West Lafayette, IN, USA

```

} else {
  stop("FileNotFound: data file not found at 'data/stroop_dataset.csv'.")
}
head(data)

```

```

##   subj trial  condition word color  RT
## 1    1     0  Congruent  red   red 1484
## 2    1     1 Incongruent green blue 1316
## 3    1     2 Incongruent blue  green 628
## 4    1     3  Congruent green green 511
## 5    1     4  Congruent blue  blue 509
## 6    1     5 Incongruent red   blue 903

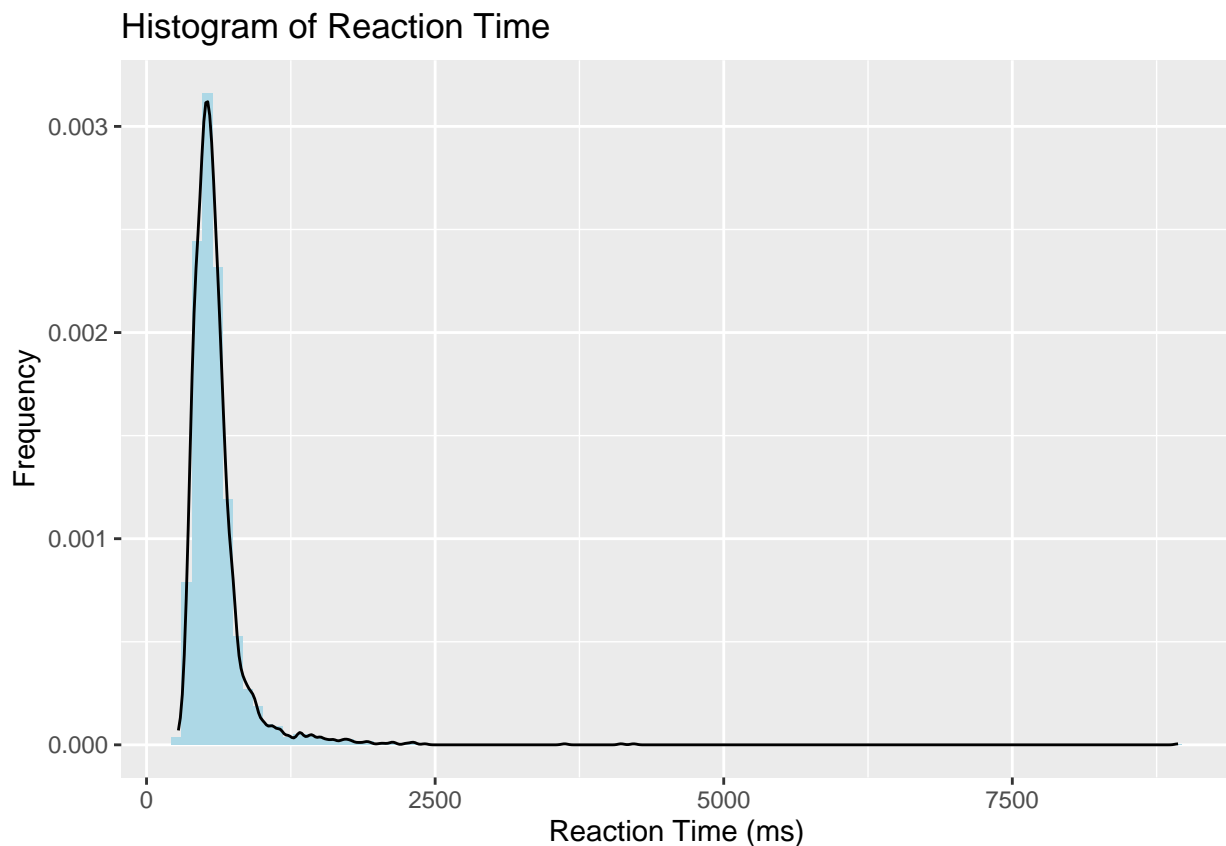
```

First we take a look at the distribution of reaction time over all subjects and trials. The following histogram shows that reaction time has a support of all positive real numbers, with a right-skewed long-tailed distribution. Therefore, we can model reaction time with a log-normal distribution.

```

ggplot(data, aes(x = RT)) +
  geom_histogram(aes(y= after_stat(density)), bins = 100, fill = "lightblue") +
  labs(title = "Histogram of Reaction Time", x = "Reaction Time (ms)", y = "Frequency") +
  geom_density()

```



According to the problem description, we have three colors and three words. Without loss of generality, we can represent them as discrete random variables. Then we obtain a binary indicator to represent the congruence of the word and the color. In this formulation, we introduce means for congruent and incongruent conditions, where both of them follow a log-normal distribution. In the distribution of the reaction time, we activate either one of them based on the binary congruence indicator. Finally, we pose standard Gamma priors on the standard deviations in the log-normal distribution of reaction time.

```

hierarchical_model_code <- "
  data {
    int<lower=0> n;                // number of responses
    int<lower=0> k;                // number of subjects
    int<lower=0> c;                // number of colors
    real<lower=0> pr_std;          // Prior standard deviations
    array[n] int<lower=1, upper=k> subjs; // Subject ID for each response
    array[n] int<lower=1, upper=c> color; // Color of the word
    array[n] int<lower=1, upper=c> word;  // Word in each trial
    array[n] real y;               // Reaction time of responses
  }

  parameters {
    vector<lower=0>[k] std;        // Emission standard deviation
    matrix[k, c] mu_congruent;    // Mean of reaction time for congruent
    matrix[k, c] mu_incongruent;  // Mean of reaction time for incongruent
  }

  transformed parameters {
    // Calculate the binary indicator for congruence
    vector[n] congruent;
    for (i in 1:n) {
      congruent[i] = (color[i] == word[i]);
    }

    // Calculate the emission mean as a Bernoulli mixture
    vector[n] mu;
    for (i in 1:n) {
      mu[i] = (
        mu_congruent[subjs[i], color[i]] * congruent[i]
        + mu_incongruent[subjs[i], color[i]] * (1 - congruent[i])
      );
    }
  }

  model {
    // Sample subject parameters from priors
    for (i in 1:k) {
      for (j in 1:c) {
        mu_congruent[i, j] ~ normal(0, pr_std);
        mu_incongruent[i, j] ~ normal(0, pr_std);
      }
    }
    std ~ gamma(1, 1);
    for (i in 1:n) {
      y[i] ~ lognormal(mu[i], std[subjs[i]]);
    }
  }

  generated quantities {
    array[n] real y_hat;
    for (i in 1:n) {
      y_hat[i] = lognormal_rng(mu[i], std[subjs[i]]);
    }
  }

```

```

    }
  }
"
hierarchical_model <- stan_model(
  model_name = "stroop_hierarchical",
  model_code = hierarchical_model_code
)

```

We sample 5,000 samples from the hierarchical model using the data as follows.

```

n <- dim(data)[1]
k <- length(unique(data$subj))
c <- length(unique(data$color))
subjs <- data$subj
color <- as.numeric(factor(data$color))
word <- as.numeric(factor(data$word))
y <- data$RT
hm_data <- list(
  n = n,
  k = k,
  c = c,
  pr_std = 10.0,
  subjs = subjs,
  color = color,
  word = word,
  y = y
)
nfit <- sampling(
  hierarchical_model,
  data = hm_data,
  chains = 1,
  iter = 10000,
  warmup = 5000,
  show_message = FALSE,
  cores = 16,
  seed = 42,
)
samples <- as.data.frame(nfit)

```