

The Probabilistic Mind: A Bayesian Cognitive Modeling Tutorial

Julia Fischer

Introduction

Technical objective

Learn the basics of Bayesian cognitive modeling, a method for simulating the mind's psychological processes as functions of a Bayesian inference machine

Substantive research question

What internal model do people use to infer others' preferences based on their response times when making decisions?

Bayesian cognitive modeling

Bayesian cognitive models are a class of computational models that aim to simulate human cognition by representing one's understanding of the world as probabilistic (or Bayesian) inference using abstract world knowledge and evidence (Tenenbaum et al., 2011). In such models, we first specify a prior distribution over possible states of the world (Lee & Wagenmakers, 2013). We then consider observed evidence, which is often noisy, and use this to update our posterior distribution over possible states of the world. As more observations are made, the model can be sequentially updated to reflect this new knowledge in the posterior distribution.

The present study: Inferring preferences from response times

We will illustrate Bayesian cognitive modeling by examining how we reason about others' decision-making processes. Reasoning about others' minds is a common topic in cognitive modeling, as Bayesian inference often provides good explanations for the systematicity and errors in humans' lay theories of others' thinking.

We make use of data from the first experiment run in Gates et al. (2021). The authors' data and analysis code are publicly available and can be downloaded from OSF (<https://osf.io/pczb3/>).

In the experiment we will be looking at, participants watched eight videos of a person making a decision between an item in one bowl and an item in another bowl. In four of the videos, the person chooses the item in bowl A after three, five, seven, or nine seconds. In the other four videos, the person chooses the item in bowl B after three, five, seven, or nine seconds. Each participant saw all eight possible versions of the video. After each video, they were asked to rate the person's preference on a sliding scale from "Strongly Prefers A" to "Strongly Prefers B."

We (and the authors) hypothesize that when a person watches someone else make a decision, they use the other person's response time as an indicator of the strength of their preference towards the item they select. Specifically, we propose that if the person takes a longer time to make their decision, the observer will rate them as having a weaker preference for the item they ultimately select. In contrast, if the person makes their

decision very quickly, the observer will rate them as having a stronger preference for their selected item. We now construct two models to test this hypothesis.

Preliminaries

Load libraries and data

```
library(brms)
library(tidyverse)
```

Read in data from experiment 1

```
# set filepath for data file
filepath <- "results_Expt1.csv"
# read in the .csv file using the read_csv() function
experiment_data <- read_csv(filepath, header = TRUE)
# remove extra header row
experiment_data <- experiment_data[-c(1:2),]
# replace missing data with NAs
experiment_data[experiment_data == ""] <- NA
```

Data manipulation

```
# create new dataframe
cleaned_data <- data.frame(matrix(data = NA, nrow = nrow(experiment_data), ncol = 9))
colnames(cleaned_data) <- c("id", "A3", "A5", "A7", "A9", "B3", "B5", "B7", "B9")

# transfer relevant data; various labels used in experiment all get mapped to A and B
for (i in 1:nrow(experiment_data)) {
  cleaned_data[i, "id"] <- i
  if (!is.na(experiment_data[i, 12])) {
    cleaned_data[i, c(2:9)] <- experiment_data[i, c(12:19)]

  } else if (!is.na(experiment_data[i, 20])) {
    cleaned_data[i, c(2:5)] <- experiment_data[i, c(24:27)]
    cleaned_data[i, c(6:9)] <- experiment_data[i, c(20:23)]

  } else if (!is.na(experiment_data[i, 28])) {
    cleaned_data[i, c(2:9)] <- experiment_data[i, c(28:35)]

  } else if (!is.na(experiment_data[i, 36])) {
    cleaned_data[i, c(2:5)] <- experiment_data[i, c(40:43)]
    cleaned_data[i, c(6:9)] <- experiment_data[i, c(36:39)]

  } else if (!is.na(experiment_data[i, 44])) {
    cleaned_data[i, c(2:9)] <- experiment_data[i, c(44:51)]
```

```

} else if (!is.na(experiment_data[i, 52])) {
  cleaned_data[i, c(2:5)] <- experiment_data[i, c(56:59)]
  cleaned_data[i, c(6:9)] <- experiment_data[i, c(52:55)]

} else if (!is.na(experiment_data[i, 60])) {
  cleaned_data[i, c(2:9)] <- experiment_data[i, c(60:67)]

} else if (!is.na(experiment_data[i, 68])) {
  cleaned_data[i, c(2:5)] <- experiment_data[i, c(72:75)]
  cleaned_data[i, c(6:9)] <- experiment_data[i, c(68:71)]

} else if (!is.na(experiment_data[i, 76])) {
  cleaned_data[i, c(2:9)] <- experiment_data[i, c(76:83)]

} else if (!is.na(experiment_data[i, 84])) {
  cleaned_data[i, c(2:5)] <- experiment_data[i, c(88:91)]
  cleaned_data[i, c(6:9)] <- experiment_data[i, c(84:87)]

} else if (!is.na(experiment_data[i, 92])) {
  cleaned_data[i, c(2:9)] <- experiment_data[i, c(92:99)]

} else if (!is.na(experiment_data[i, 100])) {
  cleaned_data[i, c(2:5)] <- experiment_data[i, c(104:107)]
  cleaned_data[i, c(6:9)] <- experiment_data[i, c(100:103)]
}
}

# pivot longer
cleaned_data_long <- cleaned_data %>% pivot_longer(cols = -id,
                                                    names_to = c("stimulus", "seconds"),
                                                    names_sep = 1,
                                                    values_to = "preference")

# add stimulus values back in
for (i in 1:nrow(cleaned_data)) {
  cleaned_data_long[8 * i - 7, "stimulus"] <- "A"
  cleaned_data_long[8 * i - 6, "stimulus"] <- "A"
  cleaned_data_long[8 * i - 5, "stimulus"] <- "A"
  cleaned_data_long[8 * i - 4, "stimulus"] <- "A"
  cleaned_data_long[8 * i - 3, "stimulus"] <- "B"
  cleaned_data_long[8 * i - 2, "stimulus"] <- "B"
  cleaned_data_long[8 * i - 1, "stimulus"] <- "B"
  cleaned_data_long[8 * i, "stimulus"] <- "B"
}

# convert values to numeric
cleaned_data_long$seconds <- as.numeric(cleaned_data_long$seconds)
cleaned_data_long$preference <- as.numeric(cleaned_data_long$preference)

# subtract "B" values from 100 for consistency
for (i in 1:nrow(cleaned_data_long)) {
  if (cleaned_data_long[i, "stimulus"] == "B") {
    cleaned_data_long[i, "preference"] <- 100 - cleaned_data_long[i, "preference"]
  }
}

```

```

    }
  }

head(cleaned_data_long)

```

```

## # A tibble: 6 x 4
##       id stimulus seconds preference
##   <int> <chr>      <dbl>      <dbl>
## 1     1 1 A          3         100
## 2     2 1 A          5         100
## 3     3 1 A          7          42
## 4     4 1 A          9          67
## 5     5 1 B          3          95
## 6     6 1 B          5          73

```

Set a seed for modeling

```
set.seed(8)
```

Model 1: Bayesian linear regression model

We will now fit our first Bayesian model, making use of the `brms` (Bayesian regression models using ‘Stan’) package for probabilistic modeling (Bürkner, 2017). This model assumes that a participant infers the person’s preference for the selected item as a linear function of how long they take to make their decision. This model is not inherently Bayesian, but we fit it in the Bayesian framework to get practice modeling cognitive processes using probability distributions to represent beliefs about the world.

Like in a standard Bayesian analysis model, we select prior distributions over the parameters we are estimating. If we have reason to believe that certain parameters should have values in a specified range, we can give these parameters informative priors. Informative priors are generally centered at a value of particular interest and deviate only slightly from that value. However, since our current model is relatively theory-agnostic, we will set uninformative priors and allow the data to find appropriate parameter values.

Fit model

```

lm.fit <- brm(preference ~ 1 + seconds,
  data = cleaned_data_long, family = gaussian(),
  prior = c(prior(normal(0, 10), class = "Intercept"),
    prior(normal(0, 10), class = "b", coef = "seconds")),
  iter = 1000, chains = 4, cores = 4)

```

```
## Compiling Stan program...
```

```
## Trying to compile a simple C file
```

```
## Start sampling
```

```
summary(lm.fit)
```

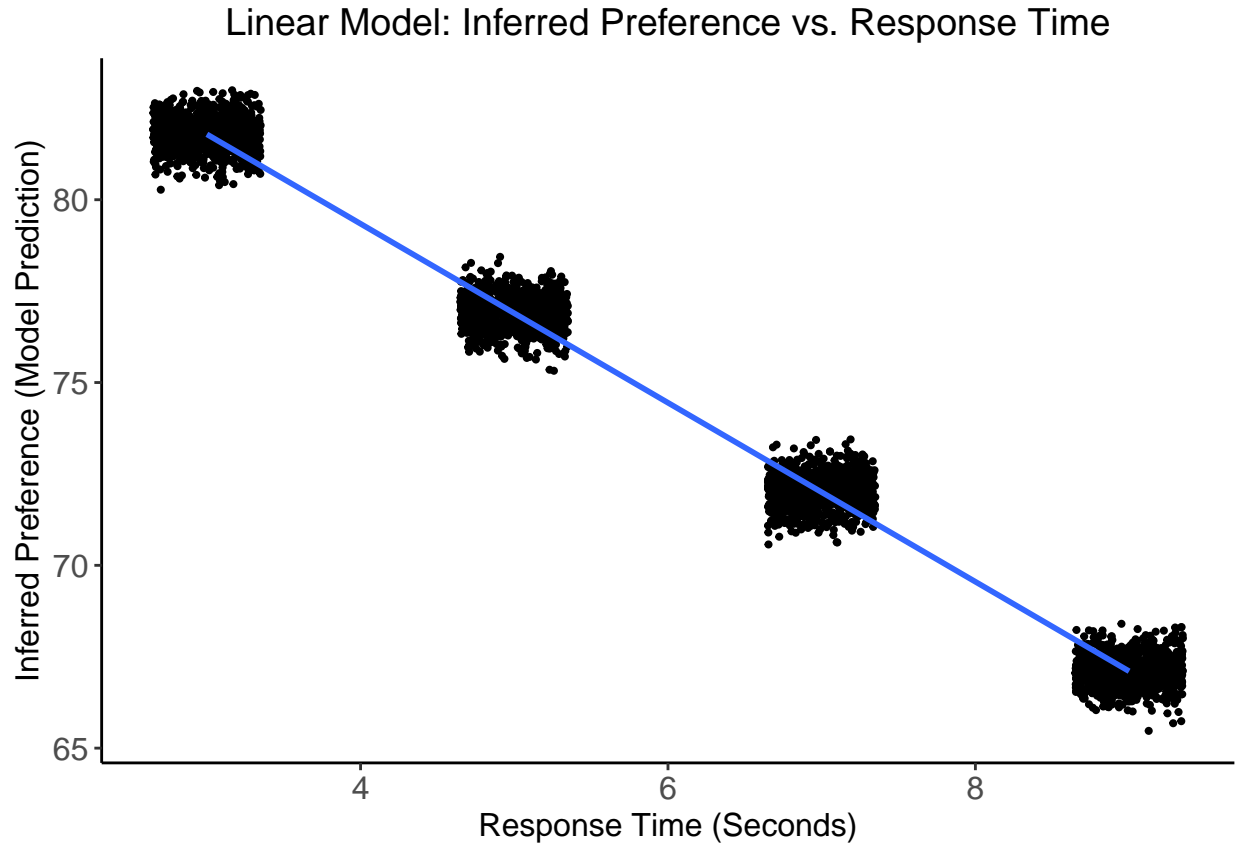
```
## Family: gaussian
## Links: mu = identity; sigma = identity
## Formula: preference ~ 1 + seconds
## Data: cleaned_data_long (Number of observations: 3848)
## Draws: 4 chains, each with iter = 1000; warmup = 500; thin = 1;
## total post-warmup draws = 2000
##
## Population-Level Effects:
##      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept    89.15     0.90   87.37   90.89 1.00    1965    1617
## seconds     -2.45     0.14   -2.72   -2.17 1.00    1855    1587
##
## Family Specific Parameters:
##      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma     20.43     0.22   19.99   20.87 1.00    2556    1645
##
## 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).
```

Plot model

```
cleaned_data_long$pred_preference <- predict(lm.fit,
                                             newdata=cleaned_data_long,
                                             allow_new_levels=TRUE)

ggplot(data = cleaned_data_long, aes(x = seconds, y = pred_preference[, "Estimate"])) +
  geom_jitter(size = .75, width = 0.35) +
  geom_smooth(method = lm) +
  xlab("Response Time (Seconds)") +
  ylab("Inferred Preference (Model Prediction)") +
  theme_classic() +
  theme(axis.title=element_text(size=12),
        axis.text=element_text(size=12),
        plot.title=element_text(size=14, hjust=.5)) +
  #theme(text = element_text(family = "Times New Roman")) +
  ggtitle("Linear Model: Inferred Preference vs. Response Time")
```

```
## 'geom_smooth()' using formula = 'y ~ x'
```



Interpret model

We see from the model summary and the plot that for each second the person's response time increases, the participant's rating of their preference decreases by -2.45 points on a 100-point scale. This is in line with our hypothesis that when observing someone else make a decision, people infer that a longer response time indicates a weaker preference for the selected object.

While this linear model is a helpful start, it does not account for the noisiness that we would expect to see in real human judgments. For each response time, this model's predictions are constrained to a very small range of preference values. We wouldn't expect humans to always guess the same preference given the same response time. Our next model allows for noise in the observer's prior distribution over strength of preference and in how they incorporate response time into their judgment of preference.

Model 2: Cognitive model with noise

For this model, which incorporates more of the probabilistic component of a Bayesian cognitive model, we use the basic structure of our fitted linear model, but incorporate random noise at several points.

We use the estimated intercept from our first model as the basis for the observer's prior distribution over strength of preference. However, we assume that there is some randomness in this prior estimate, so noise is introduced to offset prior values slightly from the estimated intercept.

We then use the estimated slope to incorporate the likelihood, which is the new evidence introduced by observing the decision-maker's reaction time. We again include some random noise in this calculation. This helps account for cases in which the observer incorrectly judges how much time passes before the decision is

made, among other sources of error. This calculation gives us the observer's posterior distribution over the decision-maker's strength of preference.

Fit model

```
# extract linear model intercept and slope
fixed_effects <- fixef(lm.fit)
intercept <- fixed_effects["Intercept", "Estimate"]
slope <- fixed_effects["seconds", "Estimate"]

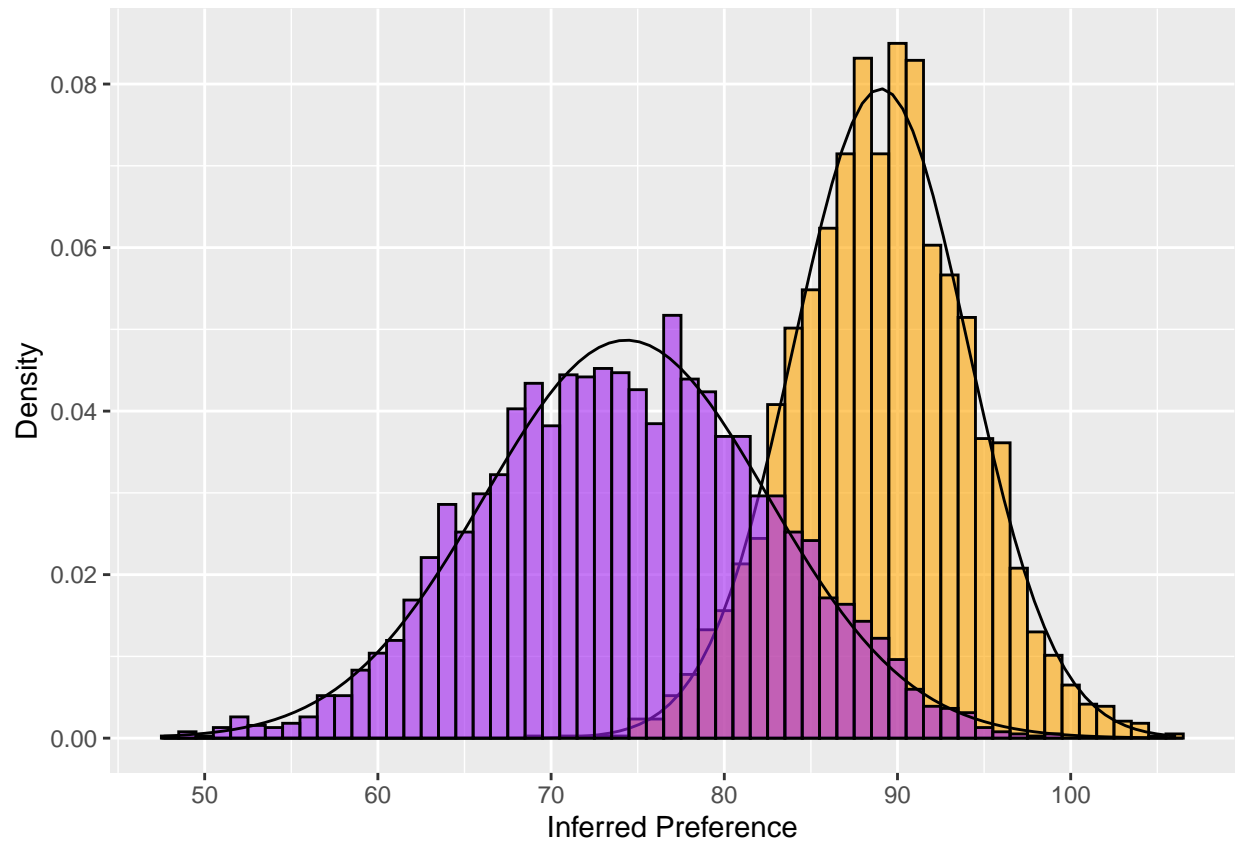
# create prior distribution that includes random noise
cleaned_data_long$prior <- rnorm(n = nrow(cleaned_data_long), mean = intercept, sd = 5)

# use linear model slope + noise as likelihood to calculate posterior
cleaned_data_long$posterior <- cleaned_data_long$prior +
  slope * cleaned_data_long$seconds +
  rnorm(n = nrow(cleaned_data_long), mean = 0, sd = 3.5)
```

Plot distributions

We can plot the prior (orange) and posterior (purple) distributions over strength of preference that we believe the observer to have.

```
cleaned_data_long %>% ggplot() +
  geom_histogram(mapping = aes(x = prior, y = after_stat(density)),
    fill = "orange", alpha = 0.6, color = "black", binwidth = 1) +
  stat_function(fun = dnorm, args = list(mean = mean(cleaned_data_long$prior),
    sd = sd(cleaned_data_long$prior))) +
  geom_histogram(mapping = aes(x = posterior, y = after_stat(density)),
    fill = "purple", alpha = 0.6, color = "black", binwidth = 1) +
  stat_function(fun = dnorm, args = list(mean = mean(cleaned_data_long$posterior),
    sd = sd(cleaned_data_long$posterior))) +
  xlab("Inferred Preference") +
  ylab("Density")
```

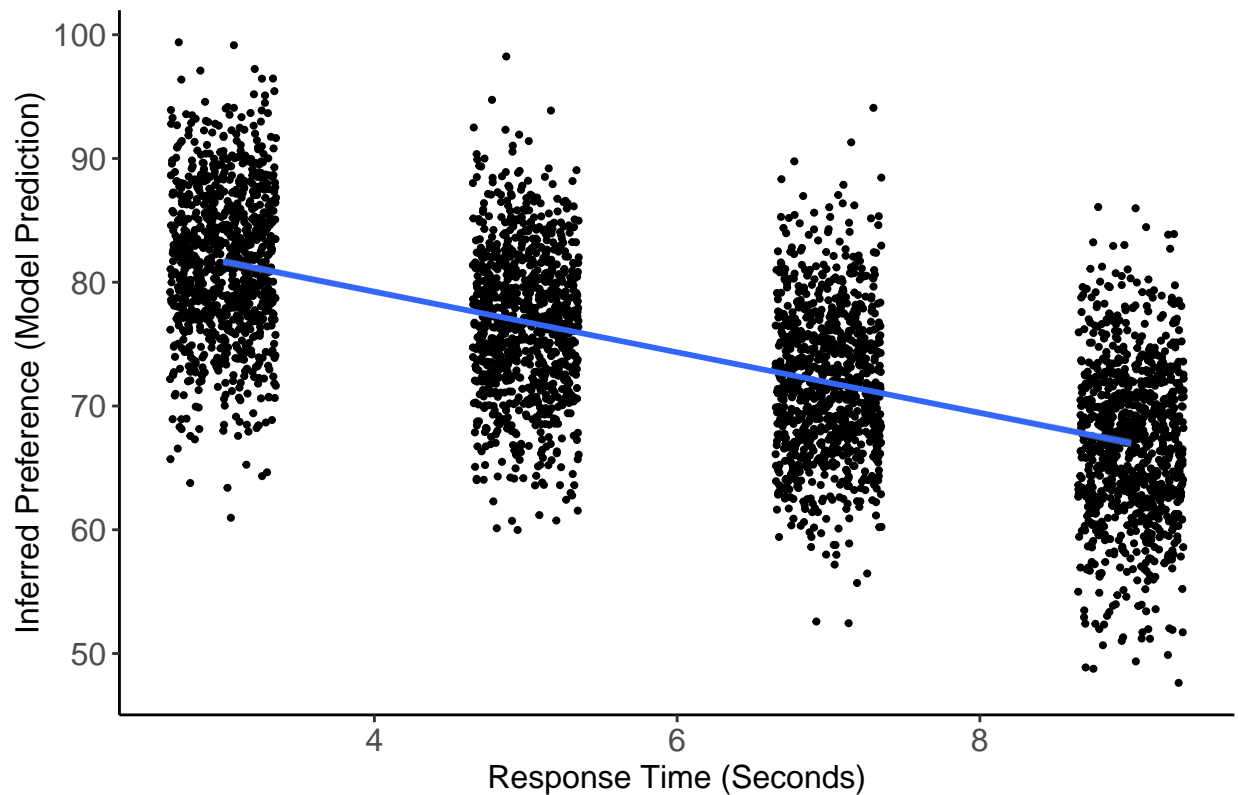


Plot model

```
ggplot(data = cleaned_data_long, aes(x = seconds, y = posterior)) +
  geom_jitter(size = .75, width = 0.35) +
  geom_smooth(method = lm) +
  xlab("Response Time (Seconds)") +
  ylab("Inferred Preference (Model Prediction)") +
  theme_classic() +
  theme(axis.title=element_text(size=12),
        axis.text=element_text(size=12),
        plot.title=element_text(size=14, hjust=.5)) +
  #theme(text = element_text(family = "Times New Roman")) +
  ggtitle("Cognitive Model with Noise: Inferred Preference vs. Response Time")
```

```
## 'geom_smooth()' using formula = 'y ~ x'
```


Cognitive Model with Noise: Inferred Preference vs. Response Time



Interpret model

We observe that compared to the initial model, this model allows for much more variation among judgments based on the same response time. This is more in line with our actual data and matches our intuitions about the imperfection of people's judgments. Nevertheless, this improved model still illustrates the same hypothesized effect of inferred preference strength decreasing with response time.

Note that we used a very simple Bayesian cognitive model here, with assumptions of linearity and normally distributed noise. We also chose noise (standard deviation) parameters by hand, as opposed to learning them from the data, which is out of the scope of this tutorial.

Conclusion

In this tutorial, we created two preliminary Bayesian cognitive models to try to explain how people infer others' preferences based on their response times when making decisions. We found support for the hypothesis that humans use a probabilistic generative model to infer others' preferences as a function of how long they take to select an item.

What we demonstrated in this tutorial is only one (relatively basic) method for constructing a cognitive model. We hope it helped you understand the basic premises of Bayesian cognitive modeling. If you want to learn how to build more complex and expressive cognitive models, we encourage you to check out probabilistic programming languages (PPLs), such as WebPPL (<http://webppl.org/>).

Additional resources

If you would like to learn more about the concepts mentioned in this tutorial, here are a few external resources:

- Using a probabilistic programming language (PPL) to construct expressive cognitive models: <https://probmods.org/>
- In-depth philosophical and mathematical details of Bayesian models of cognition: <https://cocosci.princeton.edu/tom/papers/bayeschapter.pdf>
- A video on Bayesian cognitive modeling using the Julia programming language: <https://www.youtube.com/watch?v=NNySqCSmIo4>

References

- Bürkner, P.-C. (2017). brms: An R package for Bayesian multilevel models using Stan. *Journal of Statistical Software*, 80(1). <https://doi.org/10.18637/jss.v080.i01>
- Gates, V., Callaway, F., Ho, M. K., & Griffiths, T. L. (2021). A rational model of people’s inferences about others’ preferences based on response times. *Cognition*, 217. <https://doi.org/10.1016/j.cognition.2021.104885>
- Lee, M. D. & Wagenmakers, E.-J. (2013). *Bayesian cognitive modeling: A practical course*. Cambridge University Press. <https://doi.org/10.1017/CBO9781139087759>
- Tenenbaum, J. B., Kemp, C., Griffiths, T. L., & Goodman, N. D. (2011). How to grow a mind: Statistics, structure, and abstraction. *Science*, 331(6022), 1279–1285. <https://doi.org/10.1126/science.1192788>