

Your Turn!

We will use a simple Stroop data set throughout for the practical part. You can load it using the following code in R:

```
install.packages('curl') #You need to load the R package "curl" to use this cleaning code
library(curl)

# See https://github.com/mdnunez/encodingN200 for more information about the data
filename <- curl("https://raw.githubusercontent.com/mdnunez/encodingN200
                 /master/Data/N200_rt_window_150_275.csv")
pdm <- read.csv(filename)

colnames(pdm) <- c('N200_latencies', 'N200_amplitudes'
                  , 'RT', 'accuracy', 'condition', 'EEG_session'
                  , 'experiment', 'session', 'subject')

# Take on the data from the first experiment
pdm <- pdm[pdm$experiment == 1, ]

# Remove irrelevant columns
pdm <- pdm[, -c(1:2)]

# Convert from milliseconds to seconds
pdm$RT <- pdm$RT/1000

ntrial <- dim(pdm)[1]
nsub <- length(unique(pdm$subject))
```

1. Familiarize yourself with the data. Maybe use a plot or two, see which variables might be relevant.

RT	accuracy	condition	EEG_session	experiment	session	subject
0.945	1	1	0	1	1	1
0.773	1	1	0	1	1	1
1.505	1	1	0	1	1	1
0.672	0	1	0	1	1	1
1.079	1	0	0	1	1	1
0.989	1	2	0	1	1	1

2. Suppose we wanted to model the accuracy of the responses using `brms`. As a first step, we will not use any predictors, but just model the responses provided and across incongruent and congruent trials. Here is the code (*warning*: this may take a while):

```
library(brms)
priors <- c(prior_string("normal(0, 1)", class = "Intercept"))
model_fit <- brm(RT ~ 1
  , data = pdm
  , family = gaussian
  , prior = priors
  , chains = 2
  , iter = 1500
  , warmup = 750)
```

You can see an overview over the estimated parameters using

```
summary(model_fit)

## Family: gaussian
## Links: mu = identity; sigma = identity
## Formula: RT ~ 1
## Data: pdm (Number of observations: 5532)
## Draws: 2 chains, each with iter = 1500; warmup = 750; thin = 1;
## total post-warmup draws = 1500
##
## Population-Level Effects:
##      Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept      0.78      0.00   0.77    0.78 1.00    1526    1126
##
## Family Specific Parameters:
##      Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma      0.24      0.00   0.23    0.24 1.00     759     950
##
## 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).
```

3. Let's make sense of the results. First, what does the line represent? What do the columns represent? Then, what do the estimates and do they make sense?
4. Is this a good prior for response time? What would be better?

Bonus. Suppose we wanted to model the accuracy of the responses using `brms`. As a first step, we will not use any predictors, but just model the responses provided and across incongruent and congruent trials. Here is the code (*warning*: this may take a while):

```
library(brms)
priors <- c(prior_string("normal(0, 1)", class = "Intercept"))
model_fit <- brm(accuracy ~ 1
  , data = pdm
  , family = bernoulli(link = "logit")
  , prior = priors
  , chains = 2
  , iter = 1500
  , warmup = 750)
```

You can see an overview over the estimated parameters using

```
summary(model_fit)

## Family: bernoulli
## Links: mu = logit
## Formula: accuracy ~ 1
## Data: pdm (Number of observations: 5532)
## Draws: 2 chains, each with iter = 1500; warmup = 750; thin = 1;
## total post-warmup draws = 1500
##
## Population-Level Effects:
##      Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept      0.78      0.03   0.72   0.84 1.00      533      577
##
## 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).
```

Let's again make sense of the results. Be careful to properly understand what the estimate of .78 represents. It is not the probability of responding accurately (the observed proportion of accurate responses is 0.69. (*hint*: try the `inv.logit` from the `gtools` package.)

