Conceptual metaphor and graphical convention influence the interpretation of line graphs

Greg Woodin, Bodo Winter, and Lace Padilla

10/12/2019

Contents

Main analyses

This is the code used for the analysis reported in Experiment 2 of 'Conceptual metaphor and graphical convention influence the interpretation of line graphs'.

Data wrangling

Load packages used, load datasets and give them shorter names for easier coding:

```
library(plyr)           # Data processing
library(tidyverse)      # Data processing
library(brms)           # Bayesian mixed models
library(ggmcmc)         # Data visualisation
library(tidybayes)      # Data visualisation
```

Get citation information for R and for the packages we use:

```
# R:
R.Version()
```

```
## $platform
## [1] "x86_64-apple-darwin17.0"
##
## $arch
## [1] "x86_64"
##
## $os
## [1] "darwin17.0"
##
```

```
## $system
## [1] "x86_64, darwin17.0"
##
## $status
## [1] ""
##
## $major
## [1] "4"
##
## $minor
## [1] "0.3"
##
## $year
## [1] "2020"
##
## $month
## [1] "10"
##
## $day
## [1] "10"
##
## $`svn rev`
## [1] "79318"
##
## $language
## [1] "R"
##
## $version.string
## [1] "R version 4.0.3 (2020-10-10)"
##
## $nickname
## [1] "Bunny-Wunnies Freak Out"
```

```
citation()
```

```
##
## To cite R in publications use:
##
##   R Core Team (2020). R: A language and environment for statistical
##   computing. R Foundation for Statistical Computing, Vienna, Austria.
##   URL https://www.R-project.org/.
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {R: A Language and Environment for Statistical Computing},
##     author = {{R Core Team}},
##     organization = {R Foundation for Statistical Computing},
##     address = {Vienna, Austria},
##     year = {2020},
##     url = {https://www.R-project.org/},
##   }
##
## We have invested a lot of time and effort in creating R, please cite it
## when using it for data analysis. See also 'citation("pkgname")' for
```

```
## citing R packages.
```

```
# RStudio:
#RStudio.Version()

# plyr:
citation('plyr')
```

```
##
## To cite plyr in publications use:
##
##   Hadley Wickham (2011). The Split-Apply-Combine Strategy for Data
##   Analysis. Journal of Statistical Software, 40(1), 1-29. URL
##   http://www.jstatsoft.org/v40/i01/.
##
## A BibTeX entry for LaTeX users is
##
##   @Article{,
##     title = {The Split-Apply-Combine Strategy for Data Analysis},
##     author = {Hadley Wickham},
##     journal = {Journal of Statistical Software},
##     year = {2011},
##     volume = {40},
##     number = {1},
##     pages = {1--29},
##     url = {http://www.jstatsoft.org/v40/i01/},
##   }
```

```
packageVersion('plyr')
```

```
## [1] '1.8.6'
```

```
# tidyverse:
citation('tidyverse')
```

```
##
##   Wickham et al., (2019). Welcome to the tidyverse. Journal of Open
##   Source Software, 4(43), 1686, https://doi.org/10.21105/joss.01686
##
## A BibTeX entry for LaTeX users is
##
##   @Article{,
##     title = {Welcome to the {tidyverse}},
##     author = {Hadley Wickham and Mara Averick and Jennifer Bryan and Winston Chang and Lucy D'Agostin
##     year = {2019},
##     journal = {Journal of Open Source Software},
##     volume = {4},
##     number = {43},
##     pages = {1686},
##     doi = {10.21105/joss.01686},
##   }
```

```
packageVersion('tidyverse')
```

```
## [1] '1.3.0'
```

```r
# brms:
citation('brms')
```

```
##
## To cite brms in publications use:
##
##   Paul-Christian Bürkner (2017). brms: An R Package for Bayesian
##   Multilevel Models Using Stan. Journal of Statistical Software, 80(1),
##   1-28. doi:10.18637/jss.v080.i01
##
##   Paul-Christian Bürkner (2018). Advanced Bayesian Multilevel Modeling
##   with the R Package brms. The R Journal, 10(1), 395-411.
##   doi:10.32614/RJ-2018-017
##
## To see these entries in BibTeX format, use 'print(<citation>,
## bibtex=TRUE)', 'toBibtex(.)', or set
## 'options(citation.bibtex.max=999)'.
```

```r
toBibtex(citation('brms'))
```

```
## @Article{,
##   title = {{brms}: An {R} Package for {Bayesian} Multilevel Models Using {Stan}},
##   author = {Paul-Christian Bürkner},
##   journal = {Journal of Statistical Software},
##   year = {2017},
##   volume = {80},
##   number = {1},
##   pages = {1--28},
##   doi = {10.18637/jss.v080.i01},
##   encoding = {UTF-8},
## }
##
## @Article{,
##   title = {Advanced {Bayesian} Multilevel Modeling with the {R} Package {brms}},
##   author = {Paul-Christian Bürkner},
##   journal = {The R Journal},
##   year = {2018},
##   volume = {10},
##   number = {1},
##   pages = {395--411},
##   doi = {10.32614/RJ-2018-017},
##   encoding = {UTF-8},
## }
```

```r
packageVersion('brms')
```

```
## [1] '2.14.4'
```

```r
# ggpubr:
citation('ggpubr')
```

```
##
## To cite package 'ggpubr' in publications use:
##
##   Alboukadel Kassambara (2020). ggpubr: 'ggplot2' Based Publication
##   Ready Plots. R package version 0.4.0.
```

```
##   https://CRAN.R-project.org/package=ggpubr
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {ggpubr: 'ggplot2' Based Publication Ready Plots},
##     author = {Alboukadel Kassambara},
##     year = {2020},
##     note = {R package version 0.4.0},
##     url = {https://CRAN.R-project.org/package=ggpubr},
##   }
```

```
packageVersion('ggpubr')
```

```
## [1] '0.4.0'
```

```
# ggmcmc:
citation('ggmcmc')
```

```
##
## To cite ggmcmc in publications use:
##
##   Xavier Fernández i Marín (2016). ggmcmc: Analysis of MCMC Samples and
##   Bayesian Inference. Journal of Statistical Software, 70(9), 1-20.
##   doi:10.18637/jss.v070.i09
##
## A BibTeX entry for LaTeX users is
##
##   @Article{,
##     title = {{ggmcmc}: Analysis of {MCMC} Samples and {B}ayesian Inference},
##     author = {Xavier Fern{\'a}ndez-i-Mar{\'i}n},
##     journal = {Journal of Statistical Software},
##     year = {2016},
##     volume = {70},
##     number = {9},
##     pages = {1--20},
##     doi = {10.18637/jss.v070.i09},
##   }
```

```
packageVersion('ggmcmc')
```

```
## [1] '1.5.0'
```

```
# tidybayes:
citation('tidybayes')
```

```
##
## Kay M (2020). _tidybayes: Tidy Data and Geoms for Bayesian Models_.
## doi: 10.5281/zenodo.1308151 (URL:
## https://doi.org/10.5281/zenodo.1308151), R package version 2.3.1, <URL:
## http://mjskay.github.io/tidybayes/>.
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {{tidybayes}: Tidy Data and Geoms for {Bayesian} Models},
##     author = {Matthew Kay},
```

```
##     year = {2020},
##     note = {R package version 2.3.1},
##     url = {http://mjskay.github.io/tidybayes/},
##     doi = {10.5281/zenodo.1308151},
##   }
```

```r
packageVersion('tidybayes')
```

```
## [1] '2.3.1'
```

Load data:

```r
df1 <- read_csv('../data/data_viz_1.csv')
df2 <- read_csv('../data/data_viz_2.csv')
df3 <- read_csv('../data/data_viz_3.csv')
df4 <- read_csv('../data/data_viz_4.csv')
df5 <- read_csv('../data/data_viz_5.csv')
df6 <- read_csv('../data/data_viz_6.csv')
```

Disable scientific notation:

```r
options("scipen" = 999)
```

Create new column in each dataset denoting experiment version:

```r
df1$Version <- 1
df2$Version <- 2
df3$Version <- 3
df4$Version <- 4
df5$Version <- 5
df6$Version <- 6
```

Change column names:

```r
# Create function to change V1_r columns:
col_names <- function(df, col1, col2, col3, col4){
  df <- rename(df, V1_r = col1)
  df <- rename(df, V2_r = col2)
  df <- rename(df, V3_r = col3)
  df <- rename(df, V4_r = col4)
  }

# Implement function to change V1_r columns:
df1 <- col_names(df1, 'G1_V1_r', 'G1_V2_r', 'G1_V3_r', 'G1_V4_r')
```

```
## Note: Using an external vector in selections is ambiguous.
## i Use `all_of(col1)` instead of `col1` to silence this message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
```

```
## Note: Using an external vector in selections is ambiguous.
## i Use `all_of(col2)` instead of `col2` to silence this message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
```

```
## Note: Using an external vector in selections is ambiguous.
## i Use `all_of(col3)` instead of `col3` to silence this message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
```

```
## Note: Using an external vector in selections is ambiguous.
## i Use `all_of(col4)` instead of `col4` to silence this message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
```

```
df2 <- col_names(df2, 'G2_V1_r', 'G2_V2_r', 'G2_V3_r', 'G2_V4_r')
df3 <- col_names(df3, 'G3_V1_r', 'G3_V2_r', 'G3_V3_r', 'G3_V4_r')
df4 <- col_names(df4, 'G4_V1_r', 'G4_V2_r', 'G4_V3_r', 'G4_V4_r')
df5 <- col_names(df5, 'G5_V1_r', 'G5_V2_r', 'G5_V3_r', 'G5_V4_r')
df6 <- col_names(df6, 'G6_V1_r', 'G6_V2_r', 'G6_V3_r', 'G6_V4_r')

# Create function to change FirstClick columns:
col_names <- function(df, col1, col2, col3, col4){
  df <- rename(df, V1_RT = col1)
  df <- rename(df, V2_RT = col2)
  df <- rename(df, V3_RT = col3)
  df <- rename(df, V4_RT = col4)
  }

# Implement function to change FirstClick columns:
df1 <- col_names(df1, 'G1_V1_time_First Click', 'G1_V2_time_First Click', 'G1_V3_time_First Click', 'G1_
df2 <- col_names(df2, 'G2_V1_time_First Click', 'G2_V2_time_First Click', 'G2_V3_time_First Click', 'G2_
df3 <- col_names(df3, 'G3_V1_time_First Click', 'G3_V2_time_First Click', 'G3_V3_time_First Click', 'G3_
df4 <- col_names(df4, 'G4_V1_time_First Click', 'G4_V2_time_First Click', 'G4_V3_time_First Click', 'G4_
df5 <- col_names(df5, 'G5_V1_time_First Click', 'G5_V2_time_First Click', 'G5_V3_time_First Click', 'G5_
df6 <- col_names(df6, 'G6_V1_time_First Click', 'G6_V2_time_First Click', 'G6_V3_time_First Click', 'G6_
```

Join datasets together:

```
df <- rbind.fill(df1, df2, df3, df4, df5, df6)
```

Add Subject column:

```
df$Subject <- 1:nrow(df)
```

Exclude participants who got the trick question incorrect:

```
# Original number of participants:
(old_len <- length(df$Subject))
```

```
## [1] 302
```

```
# Original number of participants in each condition:
aggregate(cbind(count = Subject) ~ Version,
          data = df,
          length)
```

```
##   Version count
## 1       1    50
## 2       2    50
## 3       3    50
## 4       4    50
## 5       5    52
## 6       6    50
```

```
# Exclude participants who got trick question wrong:
df <- filter(df, Trick == 'city')
```

```r
# Number of participants remaining:
(new_len <- length(df$Subject))
```

```
## [1] 294
```

```r
# Number of participants excluded:
old_len - new_len
```

```
## [1] 8
```

Exclude rows with response latencies more than 2 standard deviations above mean:

```r
# Preliminaries:
cols <- c(df$V1_RT, df$V2_RT, df$V3_RT, df$V4_RT)   # Combine values of columns
cols <- as.numeric(cols)                            # Make numeric
up_lim <- (mean(cols) + (sd(cols) * 2))             # Upper limit

# Upper limit:
round(up_lim, 1)
```

```
## [1] 25.9
```

```r
# Exclude:

    # First column:
    df$V1_RT <- as.numeric(df$V1_RT)    # Make numeric
    df <- filter(df, V1_RT < up_lim)    # Filter

    # Second column:
    df$V2_RT <- as.numeric(df$V2_RT)    # Make numeric
    df <- filter(df, V2_RT < up_lim)    # Filter

    # Third column:
    df$V3_RT <- as.numeric(df$V3_RT)    # Make numeric
    df <- filter(df, V3_RT < up_lim)    # Filter

    # Fourth column:
    df$V4_RT <- as.numeric(df$V4_RT)    # Make numeric
    df <- filter(df, V4_RT < up_lim)    # Filter

# Number of participants after exclusion:
(newer_len <- length(df$Subject))
```

```
## [1] 287
```

```r
# Number of participants excluded:
new_len - newer_len
```

```
## [1] 7
```

Find out info about participants:

```r
# Age
df$Age <- as.numeric(df$Age)    # Make numeric
range(df$Age)   # Range
```

```
## [1] 23 72
```

```r
round(mean(df$Age), 0)    # Mean
```

```
## [1] 39
```

```r
round(sd(df$Age), 0)    # Mean
```

```
## [1] 10
```

```r
# Gender
(xtab <- table(df$Gender))    # Raw stats
```

```
## 
##                  Female                    Male Non-binary/third gender 
##                     108                     175                       2 
##       Prefer not to say 
##                       2
```

```r
round(prop.table(xtab), 3) * 100    # Proportions (in order)
```

```
## 
##                  Female                    Male Non-binary/third gender 
##                    37.6                    61.0                     0.7 
##       Prefer not to say 
##                     0.7
```

```r
# Handedness
(xtab <- table(df$Handed))    # Raw stats
```

```
## 
##  left right 
##   142   145
```

```r
round(prop.table(xtab), 3) * 100    # Proportions (in order)
```

```
## 
##  left right 
##  49.5  50.5
```

```r
# Number of participants remaining in each condition:
(pps <- aggregate(cbind(count = Subject) ~ Version,
          data = df,
          length))
```

```
##   Version count
## 1       1    49
## 2       2    47
## 3       3    47
## 4       4    48
## 5       5    49
## 6       6    47
```

```r
# Proportions
(pps$count <- round(prop.table(pps$count), 3) * 100)
```

```
## [1] 17.1 16.4 16.4 16.7 17.1 16.4
```

Remove extraneous columns:

```r
# Columns:
df <- select(df, Subject, V1_RT, V1_r, V2_RT, V2_r, V3_RT, V3_r, V4_RT, V4_r, Handed, Version, Ed)
```

Create AxisInversion column:

```
df <- mutate(df, AxisInversion = ifelse(df$Version %in% c(1, 2), 'normal', 'inverted'))
```

Create Orientation column:

```
df <- mutate(df, Orientation = ifelse(Version %in% c('1', '3', '5'), 'quant_y', 'quant_x'))
```

Make data long and make valence column:

```
# Make long format:
df <- gather(df, Response, Measurement, c('V1_r', 'V2_r', 'V3_r', 'V4_r', 'V1_RT', 'V2_RT', 'V3_RT', 'V4

# Order data frame by subject column:
df <- arrange(df, Subject)

# Create column:
df <- mutate(df, Valence = ifelse(Response %in% c('V1_r', 'V2_r'), 'positive', 'negative'))
```

Make Accuracy column:

```
# Create column:
df <-
  mutate(df, Accuracy = case_when(
  Response == 'V1_r' & Measurement == 'Declining' ~ 'wrong',
  Response == 'V2_r' & Measurement == 'Improving' ~ 'wrong',
  Response == 'V3_r' & Measurement == 'Improving' ~ 'wrong',
  Response == 'V4_r' & Measurement == 'Declining' ~ 'wrong',
  Response == 'V1_r' & Measurement == 'Improving' ~ 'right',
  Response == 'V2_r' & Measurement == 'Declining' ~ 'right',
  Response == 'V3_r' & Measurement == 'Declining' ~ 'right',
  Response == 'V4_r' & Measurement == 'Improving' ~ 'right',
  Response == 'V1_r' & Measurement == 'Neither declining or improving' ~ 'wrong',
  Response == 'V2_r' & Measurement == 'Neither declining or improving' ~ 'wrong',
  Response == 'V3_r' & Measurement == 'Neither declining or improving' ~ 'wrong',
  Response == 'V4_r' & Measurement == 'Neither declining or improving' ~ 'wrong'))

# Order data frame by subject column:
df <- arrange(df, Subject)
```

Make column showing the trend depicted in each graph:

```
# Create column:
df <- mutate(df, Trend = ifelse(Response %in% c('V1_r', 'V3_r'), 'rising', 'falling'))
```

Make column for whether graphs with quantity on the y-axis aligned with vertical valence metaphors:

```
# Create column and fill in each row as NA by default:
df$ValAl_y <- NA

# Code whether graph did or did not align with valence metaphors for quant-y graphs:
df <-
  mutate(df, ValAl_y = case_when(
    Version == 1 & Valence == 'positive' ~ 'yes',
    Version == 3 & Valence == 'negative' ~ 'yes',
    Version == 5 & Valence == 'positive' ~ 'yes',
    Version == 1 & Valence == 'negative' ~ 'no',
    Version == 3 & Valence == 'positive' ~ 'no',
```

```r
    Version == 5 & Valence == 'negative' ~ 'no'))
```

Make column for whether graphs with quantity on the x-axis aligned with horizontal valence metaphors, irrespective of handedness:

```r
# Create column and fill in each row as NA by default:
df$ValAl_x <- NA

# Code whether graph did or did not align with valence metaphors for quant-y graphs:
df <-
  mutate(df, ValAl_x = case_when(
    Version == 2 & Valence == 'positive' ~ 'yes',
    Version == 4 & Valence == 'positive' ~ 'yes',
    Version == 6 & Valence == 'positive' ~ 'no',
    Version == 2 & Valence == 'negative' ~ 'no',
    Version == 4 & Valence == 'negative' ~ 'no',
    Version == 6 & Valence == 'negative' ~ 'yes'))

  # Check it's worked:
  sample_n(df, 10) %>%
    select(Version, Valence, ValAl_x)
```

```
##    Version  Valence ValAl_x
## 1        1 negative    <NA>
## 2        2 negative      no
## 3        5 positive    <NA>
## 4        2 negative      no
## 5        1 negative    <NA>
## 6        2 positive     yes
## 7        5 positive    <NA>
## 8        4 negative      no
## 9        6 negative     yes
## 10       3 negative    <NA>
```

Make column for whether graphs with quantity on the x-axis aligned with horizontal valence metaphors, factoring in participants' handedness:

```r
# Create column and fill in each row as NA by default:
df$ValAlHand_x <- NA

# Code whether graph did or did not align with valence metaphors for quant-y graphs:
df <-
  mutate(df, ValAlHand_x = case_when(
    Version == 2 & Valence == 'positive' & Handed == 'right' ~ 'yes',
    Version == 4 & Valence == 'positive' & Handed == 'right' ~ 'yes',
    Version == 6 & Valence == 'positive' & Handed == 'right' ~ 'no',
    Version == 2 & Valence == 'negative' & Handed == 'right' ~ 'no',
    Version == 4 & Valence == 'negative' & Handed == 'right' ~ 'no',
    Version == 6 & Valence == 'negative' & Handed == 'right' ~ 'yes',
    Version == 2 & Valence == 'positive' & Handed == 'left' ~ 'no',
    Version == 4 & Valence == 'positive' & Handed == 'left' ~ 'no',
    Version == 6 & Valence == 'positive' & Handed == 'left' ~ 'yes',
    Version == 2 & Valence == 'negative' & Handed == 'left' ~ 'yes',
    Version == 4 & Valence == 'negative' & Handed == 'left' ~ 'yes',
    Version == 6 & Valence == 'negative' & Handed == 'left' ~ 'no'))
```

Create column for x-inverted versus y-inverted graphs:

```
# Create column and fill in each row as NA by default:
df$InvertXY <-  NA

# Code whether x-axis or y-axis was inverted
df <-
  mutate(df, InvertXY = case_when(
  Version == 3 ~ 'y',
  Version == 4 ~ 'y',
  Version == 5 ~ 'x',
  Version == 6 ~ 'x'))
```

Create two separate datasets for looking at accuracy and response latency information respectively:

```
# Reduce to response latencies for use later in exploratory analysis:
df_RT <- df %>% filter(Response %in% c('V1_RT', 'V2_RT', 'V3_RT', 'V4_RT')) %>%
  mutate(Valence = case_when(
    Response == 'V1_RT' ~ 'positive',
    Response == 'V2_RT' ~ 'positive',
    Response == 'V3_RT' ~ 'negative',
    Response == 'V4_RT' ~ 'negative'
  ))

# Reduce to accuracy information for use now:
df <- df %>% filter(Response %in% c('V1_r', 'V2_r', 'V3_r', 'V4_r'))
```

Analyses

We now perform the main analyses of our study.

Descriptive statistics

Look at Accuracy overall:

```
(xtab <- table(df$Accuracy))
```

```
##
## right wrong
##   787   361
```

```
round(prop.table(xtab), 3) * 100
```

```
##
## right wrong
##  68.6  31.4
```

People were more likely to answer correctly overall, although there were a fair number of incorrect responses.

Look at descriptive stats for Accuracy as a function of AxisInversion:

```
(xtab <- table(df$AxisInversion, df$Accuracy))
```

```
##
##            right wrong
##   inverted   470   294
##   normal     317    67
```

```
round(prop.table(xtab, 1), 3) * 100
```

```
##
##            right wrong
##   inverted  61.5  38.5
##   normal    82.6  17.4
```

People were more likely to answer incorrectly for graphs with an inverted axis.

Look at descriptive stats for Accuracy as a function of Orientation:

```
(xtab <- table(df$Orientation, df$Accuracy))
```

```
##
##           right wrong
##   quant_x   441   127
##   quant_y   346   234
```

```
round(prop.table(xtab, 1), 3) * 100
```

```
##
##           right wrong
##   quant_x  77.6  22.4
##   quant_y  59.7  40.3
```

People were more likely to answer correctly when quantity was on the x-axis, maybe because they recognised that the graph was unusual, and so took longer to respond. In contrast, people may not have noticed that the axis was inverted because this requires reading of the axes.

Look at descriptive stats for Accuracy as a function of Valence:

```
(xtab <- table(df$Valence, df$Accuracy))
```

```
##
##            right wrong
##   negative   376   198
##   positive   411   163
```

```
round(prop.table(xtab, 1), 3) * 100
```

```
##
##            right wrong
##   negative  65.5  34.5
##   positive  71.6  28.4
```

People were slightly more likely to answer correctly when responding to positively valenced quantities.

Look at descriptive stats for Accuracy as a function of ValAl_y:

```
(xtab <- table(df$ValAl_y, df$Accuracy))
```

```
##
##       right wrong
##   no    159   131
##   yes   187   103
```

```
round(prop.table(xtab, 1), 3) * 100
```

```
##
##       right wrong
##   no   54.8  45.2
```

```
##   yes  64.5  35.5
```

When valence aligns with vertical spatial associations, people are more likely to interpret the resultant graphs correctly.

Look at descriptive stats for Accuracy as a function of ValAl_x:

```
(xtab <- table(df$ValAl_x, df$Accuracy))
```

```
##
##        right wrong
##   no     200    84
##   yes    241    43
```

```
round(prop.table(xtab, 1), 3) * 100
```

```
##
##        right wrong
##   no    70.4  29.6
##   yes   84.9  15.1
```

When valence aligns with horizontal spatial associations, irrespective of handedness, people are more likely to interpret the resultant graphs correctly.

Look at descriptive stats for Accuracy as a function of ValAlHand_x:

```
(xtab <- table(df$ValAlHand_x, df$Accuracy))
```

```
##
##        right wrong
##   no     227    57
##   yes    214    70
```

```
round(prop.table(xtab, 1), 3) * 100
```

```
##
##        right wrong
##   no    79.9  20.1
##   yes   75.4  24.6
```

We see the opposite (albeit weak) trend when we factor handedness in, indicating that this trend may be absolute rather than relative to participants' handedness.

Get accuracy information for each graph type that was relevant to our hypotheses (Trend was not considered here):

```
# Positive valence:
positive <- df %>% filter(Valence == 'positive')           # Filter to positive valence
(positive <- table(positive$Accuracy, positive$Version))    # Get raw N
```

```
##
##           1  2  3  4  5  6
##   right 97 85 43 87 34 65
##   wrong  1  9 51  9 64 29
```

```
round(prop.table(positive, 2) * 100, 1)                     # Proportions
```

```
##
##            1    2    3    4    5    6
##   right 99.0 90.4 45.7 90.6 34.7 69.1
##   wrong  1.0  9.6 54.3  9.4 65.3 30.9
```

```r
# Negative valence:
negative <- df %>% filter(Valence == 'negative')            # Filter to negative valence
(negative <- table(negative$Accuracy, negative$Version))     # Get raw N
```

```
##
##          1  2  3  4  5  6
##   right 73 62 56 73 43 69
##   wrong 25 32 38 23 55 25
```

```r
round(prop.table(negative, 2) * 100, 1)                      # Proportions
```

```
##
##            1    2    3    4    5    6
##   right 74.5 66.0 59.6 76.0 43.9 73.4
##   wrong 25.5 34.0 40.4 24.0 56.1 26.6
```

Inferential statistics

Model 1, which tests the following hypotheses: 1) Normal, non-inverted graphs will elicit more accurate responses than graphs with inverted axes, 2) Graphs mapping quantity onto the y-axis will elicit more/fewer accurate responses than graphs mapping quantity onto the x-axis, and 3) Graphs depicting positively-valenced quantities will elicit more accurate responses than graphs depicting negatively-valenced quantities

```r
# Run chains in parallel:
options(mc.cores = parallel::detectCores())

# Turn variables into factors:
df$Accuracy <- factor(df$Accuracy, levels = c('wrong', 'right'))
df$AxisInversion <- as.factor(df$AxisInversion)
df$Orientation <- as.factor(df$Orientation)
df$Valence <- as.factor(df$Valence)

# Set prior:
my_priors <- c(prior(normal(0, 2), class = b),
               prior(normal(0, 2), class = 'sd'))

# Set controls:
my_controls <- list(adapt_delta = 0.99,
                    max_treedepth = 13)

# Run model:
xmdl <- brm(Accuracy ~ AxisInversion + Orientation + Valence +
                (1 + Valence|Subject),
                data = df,
                family = bernoulli,
                init = 0,
                chains = 4,
                warmup = 2000,
                iter = 4000,
                prior = my_priors,
                control = my_controls,
                seed = 13)

# Summary of model:
summary(xmdl)
```

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: Accuracy ~ AxisInversion + Orientation + Valence + (1 + Valence | Subject)
##    Data: df (Number of observations: 1148)
## Samples: 4 chains, each with iter = 4000; warmup = 2000; thin = 1;
##          total post-warmup samples = 8000
##
## Group-Level Effects:
## ~Subject (Number of levels: 287)
##                                  Estimate Est.Error l-95% CI u-95% CI Rhat
## sd(Intercept)                        3.72      0.49     2.83     4.74 1.00
## sd(Valencepositive)                  3.95      0.52     2.99     5.05 1.00
## cor(Intercept,Valencepositive)      -0.57      0.13    -0.78    -0.28 1.00
##                                  Bulk_ESS Tail_ESS
## sd(Intercept)                        1885     3825
## sd(Valencepositive)                  2197     3937
## cor(Intercept,Valencepositive)       1432     2835
##
## Population-Level Effects:
##                       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept                 1.79      0.41     1.02     2.62 1.00     2674     4667
## AxisInversionnormal       2.37      0.57     1.26     3.50 1.00     1598     2977
## Orientationquant_y       -1.84      0.46    -2.78    -0.96 1.00     2268     3965
## Valencepositive           0.76      0.47    -0.10     1.72 1.00     1865     2951
##
## Samples were drawn 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).
```

```
# Get odds:
round(exp(summary(xmdl)$fixed[2, 1]), 2)  # AxisInversion
```

```
## [1] 10.73
```

```
round(exp(summary(xmdl)$fixed[3, 1]), 2)  # Orientation
```

```
## [1] 0.16
```

```
round(exp(summary(xmdl)$fixed[4, 1]), 2)  # Valence
```

```
## [1] 2.14
```

```
# Posterior predictive checks:
# pp_check(xmdl)
```

Create table summary of this model:

```
# Make table of fixed effects:
summary1 <- tibble(
  "Predictors" =  c('Axis Orientation',
                    'Quantity Mapping',
                    'Valence'),
  "Estimate" =    c(round(summary(xmdl)$fixed[2, 1], 2),
                    round(summary(xmdl)$fixed[3, 1], 2),
                    round(summary(xmdl)$fixed[4, 1], 2)),
  "Std. Error" =  c(round(summary(xmdl)$fixed[2, 2], 2),
                    round(summary(xmdl)$fixed[3, 2], 2),
```

```r
                    round(summary(xmdl)$fixed[4, 2], 2)),
  "Lower" =       c(round(summary(xmdl)$fixed[2, 3], 2),
                    round(summary(xmdl)$fixed[3, 3], 2),
                    round(summary(xmdl)$fixed[4, 3], 2)),
  "Upper" =       c(round(summary(xmdl)$fixed[2, 4], 2),
                    round(summary(xmdl)$fixed[3, 4], 2),
                    round(summary(xmdl)$fixed[4, 4], 2)))

# Factorise predictor column and re-order levels:
summary1$Predictors <- factor(summary1$Predictors, levels = c('Valence', 'Quantity Mapping', 'Axis Orien
```

Wrangle outputs from model 1 for plotting:

```r
# Convert output of model 1 into tibble:
xtrans <- ggs(xmdl)
```

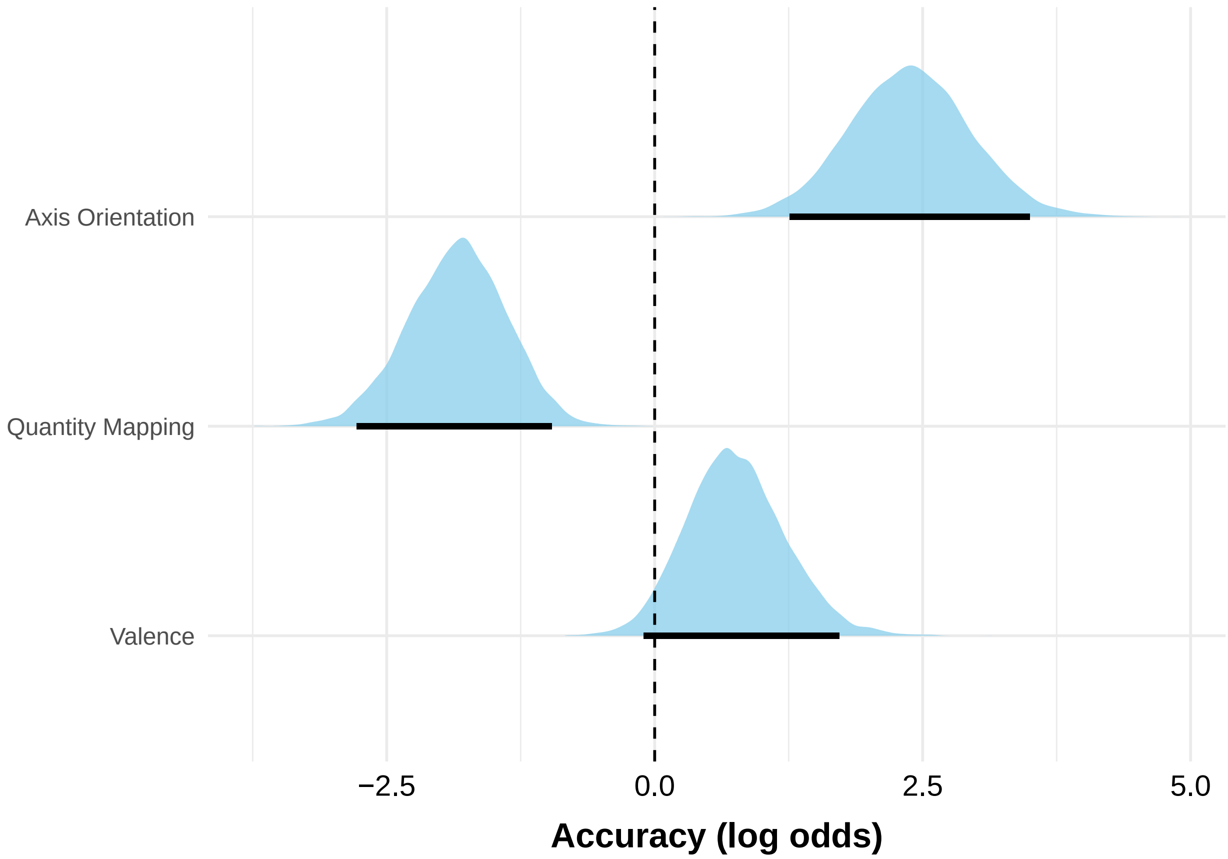
```
## Warning in custom.sort(D$Parameter): NAs introduced by coercion
```

```r
# Filter xmdl_trans to parameter rows and change name of Parameter column to match table summary (above
xmdl_trans <- xtrans %>%
  filter(Parameter %in% c('b_AxisInversionnormal', 'b_Orientationquant_y', 'b_Valencepositive')) %>%
  rename(Predictors = Parameter)

# Change name of predictor levels:
xmdl_trans$Predictors <- revalue(xmdl_trans$Predictors, c("b_AxisInversionnormal" = "Axis Orientation",
                                                          "b_Orientationquant_y" = "Quantity Mapping",
                                                          "b_Valencepositive" = "Valence"))

# Filter to above the 1000th iteration:
xmdl_trans <- xmdl_trans %>% filter(Iteration > 1000)
```

Make plot showing posterior distributions for model 1 (inspired by https://osf.io/atr57/):

```r
# Combine point estimates with posterior samples:
posterior <- merge(summary1, xmdl_trans, by = 'Predictors')

# Re-order levels:
posterior$Predictors <- factor(posterior$Predictors, levels = c("Valence", "Quantity Mapping", "Axis Ori

# Make plot:
posterior %>%
  ggplot(aes(x = value, y = Predictors, fill = Predictors, xmin = Lower, xmax = Upper)) +
  stat_slab(alpha = 0.75) +
  geom_linerange(size = 1) +
  theme_minimal() +
  geom_vline(xintercept = 0,
             color = "black",
             linetype = 2) +
    theme(axis.text.x = element_text(size = 10.5,
                                     colour = 'black'),
          axis.title.x = element_text(size = 13,
                                      face = "bold",
                                      vjust = -0.7),
          axis.title.y = element_blank(),
          legend.position = "none") +
  scale_fill_manual(values = c("skyblue", "skyblue", "skyblue")) +
```

```
  scale_x_continuous(name = "Accuracy (log odds)",
                     breaks = seq(-5, 10, 2.5))
```




```
# Save plot as PDF:
ggsave('../../table_creation/E2_model1.pdf', width = 6, height = 5)
```

Save table of model coefficients:

```
# Remove lower and upper credible intervals:
summary1 <- summary1 %>% select(-Lower, -Upper)

# Save as CSV:
write_csv(summary1, '../../table_creation/E2_model1.csv')
```

Run LOO-CV on Model 1:

```
# Run models to compare:

  # Run intercept-only model:
  #xmdl_null <- brm(Accuracy ~ 1 +
  #               (1 + Valence|Subject),
  #               data = df,
  #               family = bernoulli,
  #               init = 0,
  #               chains = 4,
  #               warmup = 2000,
  #               iter = 4000,
  #               sample_prior = "yes",
```

```r
  #               control = my_controls,
  #               seed = 13)

  # Run AxisInversion-only model:
  #xmdl_axis <- brm(Accuracy ~ AxisInversion +
  #               (1 + Valence|Subject),
  #               data = df,
  #               family = bernoulli,
  #               init = 0,
  #               chains = 4,
  #               warmup = 2000,
  #               iter = 4000,
  #               prior = my_priors,
  #               control = my_controls,
  #               seed = 13)

  # Run Orientation-only model:
  #xmdl_orient <- brm(Accuracy ~ Orientation +
  #               (1 + Valence|Subject),
  #               data = df,
  #               family = bernoulli,
  #               init = 0,
  #               chains = 4,
  #               warmup = 2000,
  #               iter = 4000,
  #               prior = my_priors,
  #               control = my_controls,
  #               seed = 13)

  # Run Valence-only model:
  #xmdl_val <- brm(Accuracy ~ Valence +
  #               (1 + Valence|Subject),
  #               data = df,
  #               family = bernoulli,
  #               init = 0,
  #               chains = 4,
  #               warmup = 2000,
  #               iter = 4000,
  #               control = my_controls,
  #               prior = my_priors,
  #               seed = 13)

# Calculate LOO for each model:
#xmdl_axis <- loo(xmdl_axis)
#xmdl_null <- loo(xmdl_null)
#xmdl_orient <- loo(xmdl_orient)
#xmdl_val <- loo(xmdl_val)

# Run LOO comparing null model with AxisInversion model:
#loo_compare(xmdl_null, xmdl_axis)

# Run LOO comparing null model with Orientation model:
#loo_compare(xmdl_null, xmdl_orient)
```

```r
# Run LOO comparing null model with Valence model:
#loo_compare(xmdl_null, xmdl_val)
```

Run model 2, which tests the hypothesis that graphs aligning with vertical valence associations will elicit more accurate responses than graphs not aligning with these associations:

```r
# Filter to graph depicting quantity on y-axis:
df_y <- df %>%
  filter(Orientation == 'quant_y')

# Create copies of relevant predictors:
df_y$AxisInversion_c <- factor(df_y$AxisInversion, levels = c('normal', 'inverted'))
df_y$Valence_c <- factor(df_y$Valence, levels = c('positive', 'negative'))

# Deviation code these predictors:
contrasts(df_y$AxisInversion_c) <- contr.sum(2) / 2
contrasts(df_y$Valence_c) <- contr.sum(2) / 2

# Run model:
y_mdl <- brm(Accuracy ~ AxisInversion_c * Valence_c +
               (1 + Valence_c|Subject),
             data = df_y,
             family = bernoulli,
             init = 0,
             chains = 4,
             warmup = 2000,
             iter = 4000,
             prior = my_priors,
             control = my_controls,
             seed = 13)
```

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

```
## Start sampling
```

```r
# Summary of model:
summary(y_mdl)
```

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: Accuracy ~ AxisInversion_c * Valence_c + (1 + Valence_c | Subject)
##    Data: df_y (Number of observations: 580)
## Samples: 4 chains, each with iter = 4000; warmup = 2000; thin = 1;
##          total post-warmup samples = 8000
##
## Group-Level Effects:
## ~Subject (Number of levels: 145)
##                             Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
## sd(Intercept)                   3.48      0.50     2.60     4.55 1.00     3046
## sd(Valence_c1)                  4.04      0.70     2.77     5.48 1.00     3052
## cor(Intercept,Valence_c1)      -0.08      0.25    -0.54     0.41 1.00     2248
##                             Tail_ESS
## sd(Intercept)                   5152
## sd(Valence_c1)                  4627
## cor(Intercept,Valence_c1)       4117
```

```
##
## Population-Level Effects:
##                              Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
## Intercept                        2.19      0.44     1.36     3.10 1.00     2848
## AxisInversion_c1                 4.94      0.82     3.42     6.65 1.00     3897
## Valence_c1                       1.23      0.64    -0.00     2.51 1.00     4645
## AxisInversion_c1:Valence_c1      4.39      1.19     2.03     6.77 1.00     4207
##                              Tail_ESS
## Intercept                        4098
## AxisInversion_c1                 5323
## Valence_c1                       5591
## AxisInversion_c1:Valence_c1      6023
##
## Samples were drawn 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).
```

```
# Posterior predictive checks:
# pp_check(y_mdl)

# Get odds for interaction:
round(exp(summary(y_mdl)$fixed[4, 1]), 2)
```

```
## [1] 80.33
```

```
# Get posterior samples:
myposts <- posterior_samples(y_mdl) %>%
  select(b_Intercept, b_AxisInversion_c1, b_Valence_c1, `b_AxisInversion_c1:Valence_c1`)

# Save samples for different columns:
intercept <- myposts$b_Intercept
axis_coef <- myposts$b_AxisInversion_c1
val_coef <- myposts$b_Valence_c1
interaction_coef <- myposts$`b_AxisInversion_c1:Valence_c1`

# Normal, positive graphs:
normal_positive <- (intercept +
                      (+0.5) * axis_coef +
                      (+0.5) * val_coef +
                      (+0.5) * (+0.5) * interaction_coef)
round(quantile(normal_positive, 0.025), 2)
```

```
## 2.5%
## 4.53
```

```
round(quantile(normal_positive, 0.975), 2)
```

```
## 97.5%
##  8.44
```

```
# Normal, negative graphs:
normal_negative <- (intercept +
                      (+0.5) * axis_coef +
                      (-0.5) * val_coef +
                      (+0.5) * (-0.5) * interaction_coef)
round(quantile(normal_negative, 0.025), 2)
```

```
## 2.5%
## 1.47
```

```
round(quantile(normal_negative, 0.975), 2)
```

```
## 97.5%
##  4.71
```

```
# Inverted, positive graphs:
inverted_positive <- (intercept +
                      (-0.5) * axis_coef +
                      (+0.5) * val_coef +
                      (-0.5) * (+0.5) * interaction_coef)
round(quantile(inverted_positive, 0.025), 2)
```

```
##  2.5%
## -1.73
```

```
round(quantile(inverted_positive, 0.975), 2)
```

```
## 97.5%
##  0.22
```

```
# Inverted, negative graphs:
inverted_negative <- (intercept +
                      (-0.5) * axis_coef +
                      (-0.5) * val_coef +
                      (-0.5) * (-0.5) * interaction_coef)
round(quantile(inverted_negative, 0.025), 2)
```

```
##  2.5%
## -0.83
```

```
round(quantile(inverted_negative, 0.975), 2)
```

```
## 97.5%
##  1.27
```

Get accuracy proportions for each graph type:

```
# Normal graphs:
(xtab <- df_y %>%
  filter(AxisInversion == 'normal') %>%
  with(table(Accuracy, Valence)))
```

```
##         Valence
## Accuracy negative positive
##    wrong       25        1
##    right       73       97
```

```
round(prop.table(xtab, 2) * 100, 1)
```

```
##         Valence
## Accuracy negative positive
##    wrong     25.5      1.0
##    right     74.5     99.0
```

```
# Inverted graphs:
(xtab <- df_y %>%
  filter(AxisInversion == 'inverted') %>%
```

```
  with(table(Accuracy, Valence)))
```

```
##         Valence
## Accuracy negative positive
##    wrong       93      115
##    right       99       77
```

```
round(prop.table(xtab, 2) * 100, 1)
```

```
##         Valence
## Accuracy negative positive
##    wrong     48.4     59.9
##    right     51.6     40.1
```

Create table summary of this model:

```
# Make table of fixed effects:
summary2 <- tibble(
  "Predictors" =  c("Axis Orientation",
                    "Valence",
                    "Axis Orientation x Valence"),
  "Estimate"   =  c(round(summary(y_mdl)$fixed[2, 1], 2),
                    round(summary(y_mdl)$fixed[3, 1], 2),
                    round(summary(y_mdl)$fixed[4, 1], 2)),
  "Std. Error" =  c(round(summary(y_mdl)$fixed[2, 2], 2),
                    round(summary(y_mdl)$fixed[3, 2], 2),
                    round(summary(y_mdl)$fixed[4, 2], 2)),
  "Lower"      =  c(round(summary(y_mdl)$fixed[2, 3], 2),
                    round(summary(y_mdl)$fixed[3, 3], 2),
                    round(summary(y_mdl)$fixed[4, 3], 2)),
  "Upper"      =  c(round(summary(y_mdl)$fixed[2, 4], 2),
                    round(summary(y_mdl)$fixed[3, 4], 2),
                    round(summary(y_mdl)$fixed[4, 4], 2)))

# Factorise predictor column and re-order levels:
summary2$Predictors <- factor(summary2$Predictors, levels = c("Axis Orientation x Valence", "Valence",
```

Wrangle outputs from model 2 for plotting:

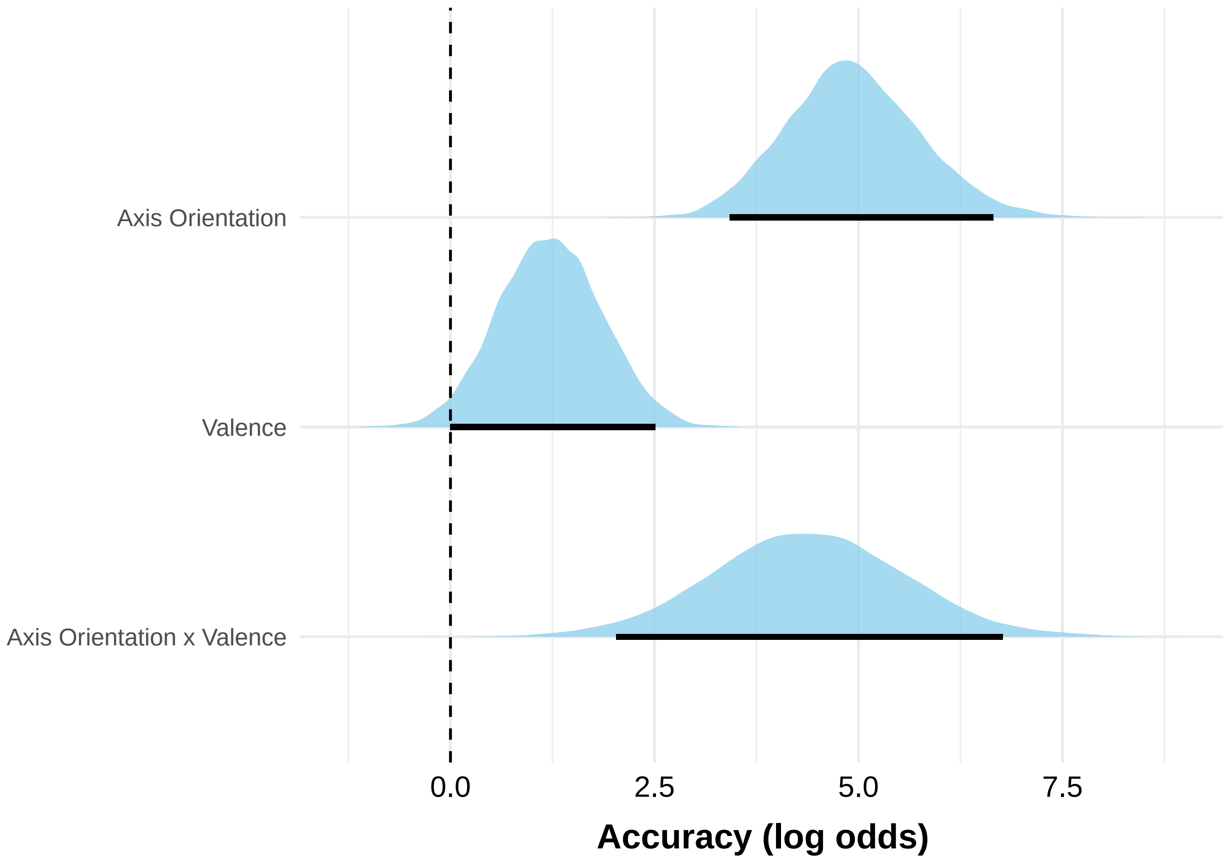
```
# Convert output of model 1 into tibble:
ytrans <- ggs(y_mdl)
```

```
## Warning in custom.sort(D$Parameter): NAs introduced by coercion
```

```
# Filter xmdl_trans to parameter rows and change name of Parameter column to match table summary (above
xmdl_trans_2 <- ytrans %>%
  filter(Parameter %in% c('b_AxisInversion_c1', 'b_Valence_c1', 'b_AxisInversion_c1:Valence_c1')) %>%
  rename(Predictors = Parameter)

# Change name of predictor levels:
xmdl_trans_2$Predictors <- revalue(xmdl_trans_2$Predictors, c('b_AxisInversion_c1' = "Axis Orientation"
                                                              "b_Valence_c1" = "Valence",
                                                              "b_AxisInversion_c1:Valence_c1" = "Axis O

# Filter to above the 1000th iteration:
xmdl_trans_2 <- xmdl_trans_2 %>% filter(Iteration > 1000)
```

Make plot showing posterior distributions for model 2 (inspired by https://osf.io/atr57/):

```
# Combine point estimates with posterior samples:
posterior <- merge(summary2, xmdl_trans_2, by = 'Predictors')

# Re-order levels:
posterior$Predictors <- as.factor(posterior$Predictors)

# Make plot:
posterior %>%
  ggplot(aes(x = value, y = Predictors, fill = Predictors, xmin = Lower, xmax = Upper)) +
  stat_slab(alpha = 0.75) +
  geom_linerange(size = 1) +
  theme_minimal() +
  geom_vline(xintercept = 0,
             color = "black",
             linetype = 2) +
    theme(axis.text.x = element_text(size = 10.5,
                                     colour = 'black'),
          axis.title.x = element_text(size = 13,
                                      face = "bold",
                                      vjust = -0.7),
          axis.title.y = element_blank(),
          legend.position = "none") +
  scale_fill_manual(values = c("skyblue", "skyblue", "skyblue")) +
  scale_x_continuous(name = "Accuracy (log odds)",
                     breaks = seq(-5, 15, 2.5))
```



```r
# Save plot as PDF:
ggsave('../../table_creation/E2_model2.pdf', width = 6, height = 5)
```

Save table summary for model 2:

```r
# Remove lower and upper 95% credible interval information:
summary2 <- summary2 %>% select(-Lower, -Upper)

# Save as CSV:
write_csv(summary2, '../../table_creation/E2_model2.csv')
```

Run LOO-CV on Model 2:

```r
# Run models to compare:

  # Run intercept-only model:
  #y_mdl_null <- brm(Accuracy ~ 1 +
                #(1 + Valence_c|Subject),
                #data = df_y,
                #family = bernoulli,
                #init = 0,
                #chains = 4,
                #warmup = 2000,
                #iter = 4000,
                #sample_prior = "yes",
                #control = my_controls,
                #seed = 13)

  # Run AxisInversion-Valence interaction model:
  #y_mdl <- brm(Accuracy ~ AxisInversion_c * Valence_c +
             #(1 + Valence_c|Subject),
             #data = df_y,
             #family = bernoulli,
             #init = 0,
             #chains = 4,
             #warmup = 2000,
             #iter = 4000,
             #control = my_controls,
             #prior = my_priors,
             #seed = 13)

# Calculate LOO scores for each mode:
#y_mdl_null <- loo(y_mdl_null)
#y_mdl <- loo(y_mdl)

# Run LOO comparing null model with interaction model:
#loo(y_mdl_null, y_mdl)
```

Run Model 3, which tests the following hypotheses: 1) Graphs aligning with horizontal valence associations (relative to handedness) will elicit more accurate responses than graphs not aligning with these associations, and 2) Graphs aligning with horizontal valence associations (where GOOD IS RIGHT, irrespective of handedness) will elicit more accurate responses than graphs not aligning with these associations.

(NB: This model is slightly different to the one preregistered because the original model was a fully saturated model and didn't show overall effect of interaction).

```r
# Filter to graphs with quantity on the y-axis:
df_x <- df %>% filter(Orientation == 'quant_x')

# Create copies of relevant predictors:
df_x$AxisInversion_c <- factor(df_x$AxisInversion, levels = c('normal', 'inverted'))
df_x$Valence_c <- factor(df_x$Valence, levels = c('positive', 'negative'))
df_x$Handed_c <- factor(df_x$Handed, levels = c('right', 'left'))

# Deviation code these predictors:
contrasts(df_x$AxisInversion_c) <- contr.sum(2) / 2
contrasts(df_x$Valence_c) <- contr.sum(2) / 2
contrasts(df_x$Handed_c) <- contr.sum(2) / 2

# Run model:
x_mdl <- brm(Accuracy ~ (AxisInversion_c * Valence_c) +
                        (AxisInversion_c * Valence_c * Handed_c) +
                        (1 + Valence_c|Subject),
                data = df_x,
                family = bernoulli,
                init = 0,
                chains = 4,
                warmup = 2000,
                iter = 4000,
                prior = my_priors,
                control = my_controls,
                seed = 13)

# Summary of model:
summary(x_mdl)
```

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: Accuracy ~ (AxisInversion_c * Valence_c) + (AxisInversion_c * Valence_c * Handed_c) + (1 + V
##    Data: df_x (Number of observations: 568)
## Samples: 4 chains, each with iter = 4000; warmup = 2000; thin = 1;
##          total post-warmup samples = 8000
##
## Group-Level Effects:
## ~Subject (Number of levels: 142)
##                          Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
## sd(Intercept)                2.06      0.36     1.43     2.85 1.00     2540
## sd(Valence_c1)               2.80      0.63     1.66     4.13 1.00     2246
## cor(Intercept,Valence_c1)    0.30      0.27    -0.29     0.75 1.00     2301
##                          Tail_ESS
## sd(Intercept)                4168
## sd(Valence_c1)               3318
## cor(Intercept,Valence_c1)    3649
##
## Population-Level Effects:
##                                      Estimate Est.Error l-95% CI u-95% CI Rhat
## Intercept                                2.46      0.36     1.84     3.25 1.00
## AxisInversion_c1                         0.26      0.50    -0.74     1.25 1.00
## Valence_c1                               1.94      0.62     0.81     3.25 1.00
## Handed_c1                               -0.20      0.49    -1.16     0.76 1.00
```

```
## AxisInversion_c1:Valence_c1              1.69      0.81     0.09     3.31 1.00
## AxisInversion_c1:Handed_c1              -1.19      0.88    -2.93     0.52 1.00
## Valence_c1:Handed_c1                    -1.03      0.78    -2.57     0.49 1.00
## AxisInversion_c1:Valence_c1:Handed_c1   -0.48      1.29    -3.04     2.01 1.00
##                                       Bulk_ESS Tail_ESS
## Intercept                                 3392     4622
## AxisInversion_c1                          5227     5650
## Valence_c1                                2960     4579
## Handed_c1                                 4990     5012
## AxisInversion_c1:Valence_c1               7451     6686
## AxisInversion_c1:Handed_c1                6052     6684
## Valence_c1:Handed_c1                      7137     6467
## AxisInversion_c1:Valence_c1:Handed_c1     7885     6225
##
## Samples were drawn 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).
```

```r
# Posterior predictive checks:
# pp_check(x_mdl)

# Get odds for axis-valence interaction:
round(exp(summary(x_mdl)$fixed[5, 1]), 2)
```

```
## [1] 5.43
```

```r
# Get odds for axis-valence-handedness interaction:
round(exp(summary(x_mdl)$fixed[8, 1]), 2)
```

```
## [1] 0.62
```

```r
# Get posterior samples:
myposts <- posterior_samples(x_mdl) %>%
  select(b_Intercept:`b_AxisInversion_c1:Valence_c1:Handed_c1`)

# Save samples for different columns:
intercept <- myposts$b_Intercept
axis_coef <- myposts$b_AxisInversion_c1
val_coef <- myposts$b_Valence_c1
interaction_coef <- myposts$`b_AxisInversion_c1:Valence_c1`

# Normal, positive graphs:
normal_positive <- (intercept +
                      (+0.5) * axis_coef +
                      (+0.5) * val_coef +
                      (+0.5) * (+0.5) * interaction_coef)
round(quantile(normal_positive, 0.025), 2)
```

```
## 2.5%
## 2.64
```

```r
round(quantile(normal_positive, 0.975), 2)
```

```
## 97.5%
##  5.68
```

```r
# Normal, negative graphs:
normal_negative <- (intercept +
```

```r
                        (+0.5) * axis_coef +
                        (-0.5) * val_coef +
                        (+0.5) * (-0.5) * interaction_coef)
round(quantile(normal_negative, 0.025), 2)
```

```
## 2.5%
## 0.39
```

```r
round(quantile(normal_negative, 0.975), 2)
```

```
## 97.5%
##  2.15
```

```r
# Inverted, positive graphs:
inverted_positive <- (intercept +
                        (-0.5) * axis_coef +
                        (+0.5) * val_coef +
                        (-0.5) * (+0.5) * interaction_coef)
round(quantile(inverted_positive, 0.025), 2)
```

```
## 2.5%
## 1.85
```

```r
round(quantile(inverted_positive, 0.975), 2)
```

```
## 97.5%
##  4.23
```

```r
# Inverted, negative graphs:
inverted_negative <- (intercept +
                        (-0.5) * axis_coef +
                        (-0.5) * val_coef +
                        (-0.5) * (-0.5) * interaction_coef)
round(quantile(inverted_negative, 0.025), 2)
```

```
## 2.5%
## 1.11
```

```r
round(quantile(inverted_negative, 0.975), 2)
```

```
## 97.5%
##  2.61
```

Get accuracy proportions for each graph type:

```r
# Normal graphs:
(xtab <- df_x %>%
  filter(AxisInversion == 'normal') %>%
  with(table(Accuracy, Valence)))
```

```
##          Valence
## Accuracy negative positive
##    wrong       32        9
##    right       62       85
```

```r
round(prop.table(xtab, 2) * 100, 1)
```

```
##          Valence
## Accuracy negative positive
```

```
##    wrong     34.0      9.6
##    right     66.0     90.4
```

```r
# Inverted graphs:
(xtab <- df_x %>%
  filter(AxisInversion == 'inverted') %>%
  with(table(Accuracy, Valence)))
```

```
##         Valence
## Accuracy negative positive
##    wrong       48       38
##    right      142      152
```

```r
round(prop.table(xtab, 2) * 100, 1)
```

```
##         Valence
## Accuracy negative positive
##    wrong     25.3     20.0
##    right     74.7     80.0
```

Create table summary of this model:

```r
# Make table of fixed effects:
summary3 <- tibble(
  "Predictors" =  c('Axis Orientation',
                    'Valence',
                    'Hand',
                    'Axis Orientation x Valence',
                    'Axis Orientation x Hand',
                    'Valence x Hand',
                    'Axis Orientation x Valence x Hand'),
  "Estimate"   =  c(round(summary(x_mdl)$fixed[2, 1], 2),
                    round(summary(x_mdl)$fixed[3, 1], 2),
                    round(summary(x_mdl)$fixed[4, 1], 2),
                    round(summary(x_mdl)$fixed[5, 1], 2),
                    round(summary(x_mdl)$fixed[6, 1], 2),
                    round(summary(x_mdl)$fixed[7, 1], 2),
                    round(summary(x_mdl)$fixed[8, 1], 2)),
  "Std. Error" =  c(round(summary(x_mdl)$fixed[2, 2], 2),
                    round(summary(x_mdl)$fixed[3, 2], 2),
                    round(summary(x_mdl)$fixed[4, 2], 2),
                    round(summary(x_mdl)$fixed[5, 2], 2),
                    round(summary(x_mdl)$fixed[6, 2], 2),
                    round(summary(x_mdl)$fixed[7, 2], 2),
                    round(summary(x_mdl)$fixed[8, 2], 2)),
  "Lower"      =  c(round(summary(x_mdl)$fixed[2, 3], 2),
                    round(summary(x_mdl)$fixed[3, 3], 2),
                    round(summary(x_mdl)$fixed[4, 3], 2),
                    round(summary(x_mdl)$fixed[5, 3], 2),
                    round(summary(x_mdl)$fixed[6, 3], 2),
                    round(summary(x_mdl)$fixed[7, 3], 2),
                    round(summary(x_mdl)$fixed[8, 3], 2)),
  "Upper"      =  c(round(summary(x_mdl)$fixed[2, 4], 2),
                    round(summary(x_mdl)$fixed[3, 4], 2),
                    round(summary(x_mdl)$fixed[4, 4], 2),
                    round(summary(x_mdl)$fixed[5, 4], 2),
```

```r
                    round(summary(x_mdl)$fixed[6, 4], 2),
                    round(summary(x_mdl)$fixed[7, 4], 2),
                    round(summary(x_mdl)$fixed[8, 4], 2)))

# Factorise predictor column and re-order levels:
summary3$Predictors <- factor(summary3$Predictors, levels = c('Axis Orientation x Valence x Hand', 'Vale
```

Wrangle outputs from model 3 for plotting:

```r
# Convert output of model 2 into tibble:
x_trans <- ggs(x_mdl)
```

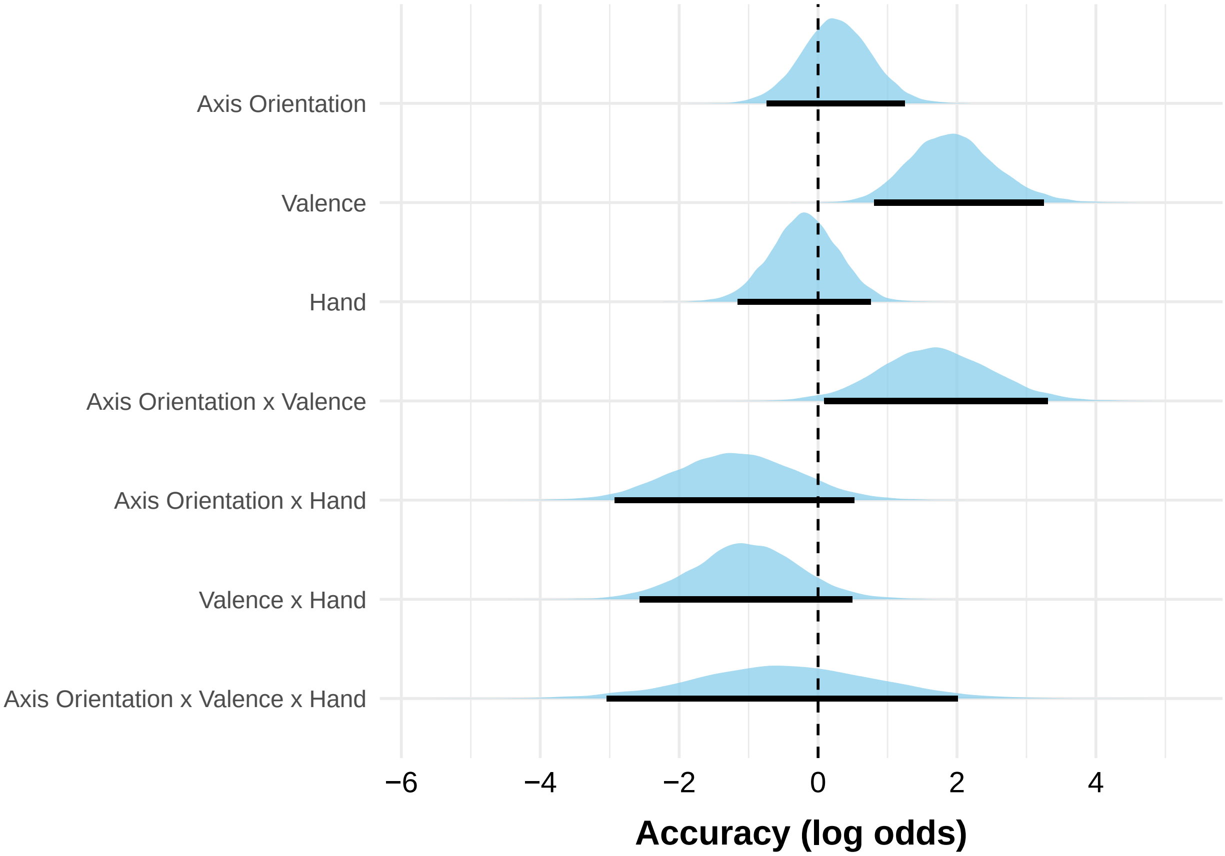
```
## Warning in custom.sort(D$Parameter): NAs introduced by coercion
```

```r
# Filter xmdl_trans_2 to interaction row:
xmdl_trans_3 <- x_trans %>%
  filter(Parameter %in% c("b_AxisInversion_c1", "b_Valence_c1", "b_Handed_c1", 'b_AxisInversion_c1:Valen
  rename(Predictors = Parameter)

# Change name of predictor levels:
xmdl_trans_3$Predictors <- revalue(xmdl_trans_3$Predictors, c("b_AxisInversion_c1" = "Axis Orientation"
                                                              "b_Valence_c1" = "Valence",
                                                              "b_Handed_c1" = "Hand",
                                                              "b_AxisInversion_c1:Valence_c1" = "Axis Or
                                                              "b_AxisInversion_c1:Handed_c1" = "Axis Ori
                                                              "b_Valence_c1:Handed_c1" = "Valence x Hand
                                                              "b_AxisInversion_c1:Valence_c1:Handed_c1"

# Filter to above the 1000th iteration:
xmdl_trans_3 <- xmdl_trans_3 %>% filter(Iteration > 1000)
```

Make plot showing posterior distributions for model 3 (inspired by https://osf.io/atr57/):

```r
# Combine point estimates with posterior samples:
posterior <- merge(xmdl_trans_3, summary3, by = 'Predictors')

# Re-order levels:
posterior$Predictors <- factor(posterior$Predictors, levels = c("Axis Orientation x Valence x Hand", "Va

# Make plot:
posterior %>%
  ggplot(aes(x = value, y = Predictors, fill = Predictors, xmin = Lower, xmax = Upper)) +
  stat_slab(alpha = 0.75) +
  geom_linerange(size = 1) +
  theme_minimal() +
  geom_vline(xintercept = 0,
             color = "black",
             linetype = 2) +
    theme(axis.text.x = element_text(size = 10.5,
                                     colour = 'black'),
          axis.title.x = element_text(size = 13,
                                      face = "bold",
                                      vjust = -0.8),
          axis.title.y = element_blank(),
          legend.position = "none") +
  scale_fill_manual(values = c("skyblue", "skyblue", "skyblue", "skyblue", "skyblue", "skyblue", "skybl
```

```
  scale_x_continuous(name = "Accuracy (log odds)",
                     breaks = seq(-8, 8, 2))
```




```
# Save plot as PDF:
ggsave('../../table_creation/E2_model3.pdf', width = 6, height = 5)
```

Save table summary:

```
# Remove lower and upper credible interval columns:
summary3 <- summary3 %>% select(-Lower, -Upper)

# Save as CSV:
write_csv(summary3, '../../table_creation/E2_model3.csv')
```

Run LOO-CV on Model 3:

```
# Run models to compare:

  # Run intercept-only model:
  #xmdl_null <- brm(Accuracy ~ 1 +
  #               (1 + Valence_c|Subject),
  #               data = df_x,
  #               family = bernoulli,
  #               init = 0,
  #               chains = 4,
  #               warmup = 2000,
  #               iter = 4000,
  #               sample_prior = "yes",
```

```
  #               control = my_controls,
  #               seed = 13)

  # Run AxisInversion-Valence interaction model:
  #xmdl_inter <- brm(Accuracy ~ AxisInversion:Valence +
  #               (1 + Valence_c|Subject),
  #               data = df_x,
  #               family = bernoulli,
  #               init = 0,
  #               chains = 4,
  #               warmup = 2000,
  #               iter = 4000,
  #               prior = my_priors,
  #               control = my_controls,
  #               seed = 13)

  # Run AxisInversion-Valence-Handed interaction model:
  #xmdl_inter_hand <- brm(Accuracy ~ AxisInversion:Valence:Handed +
  #               (1 + Valence_c|Subject),
  #               data = df_x,
  #               family = bernoulli,
  #               init = 0,
  #               chains = 4,
  #               warmup = 2000,
  #               iter = 4000,
  #               prior = my_priors,
  #               control = my_controls,
  #               seed = 13)

# Calculate LOO for these models:
#xmdl_null <- loo(xmdl_null)
#xmdl_inter <- loo(xmdl_inter)
#xmdl_inter_hand <- loo(xmdl_inter_hand)

# Run LOO comparing null model with VagueVsSpecific and WordVsNumber models:
#loo(xmdl_null, xmdl_inter, xmdl_inter_hand)
```

Exploratory analysis

First, check whether axis inversion effect was stronger for y-axis graphs than x-axis graphs:

```
(xtab <- table(df$Accuracy, df$AxisInversion, df$Orientation))
```

```
## , ,  = quant_x
##
##
##          inverted normal
##   wrong        86     41
##   right       294    147
##
## , ,  = quant_y
##
##
##          inverted normal
```

```
##   wrong      208     26
##   right      176    170
```

```
round(prop.table(xtab, c(2, 3)), 3) * 100
```

```
## , ,  = quant_x
##
##
##         inverted normal
##   wrong     22.6   21.8
##   right     77.4   78.2
##
## , ,  = quant_y
##
##
##         inverted normal
##   wrong     54.2   13.3
##   right     45.8   86.7
```

For inverted graphs, check effects of time axis versus quantity axis being subverted:

```
# Filter dataset to inverted graphs and add column to mark whether quantity or time is subverted:
df %>%
  filter(AxisInversion == 'inverted') %>%
  mutate(WhichSubvert = case_when(
    Orientation == 'quant_y' & InvertXY == 'y' ~ 'quant',
    Orientation == 'quant_x' & InvertXY == 'x' ~ 'quant',
    Orientation == 'quant_y' & InvertXY == 'x' ~ 'time',
    Orientation == 'quant_x' & InvertXY == 'y' ~ 'time')) %>%
  with(print(table(Accuracy, WhichSubvert))) %>%
  prop.table(2) %>%
  round(3) * 100
```

```
##         WhichSubvert
## Accuracy quant time
##    wrong   143  151
##    right   233  237
```

```
##         WhichSubvert
## Accuracy quant time
##    wrong  38.0 38.9
##    right  62.0 61.1
```

Reviewer-requested additional analysis

Educational background We now look at the effect of educational background on responses.

First, look at demographic information:

```
(xtab <- table(df$Ed))
```

```
##
##                                     Associate degree in college (2-year)
##                                                                      124
##                                    Bachelor's degree in college (4-year)
##                                                                      524
##                                                          Doctoral degree
##                                                                        8
```

```
## High school graduate (high school diploma or equivalent including GED)
##                                                                    148
##                                              Less than high school degree
##                                                                      4
##                                                           Master's degree
##                                                                    108
##                                              Professional degree (JD, MD)
##                                                                     24
##                                                Some college but no degree
##                                                                    208
```

```
round(prop.table(xtab) * 100, 1)
```

```
##
##                                      Associate degree in college (2-year)
##                                                                   10.8
##                                     Bachelor's degree in college (4-year)
##                                                                   45.6
##                                                           Doctoral degree
##                                                                    0.7
## High school graduate (high school diploma or equivalent including GED)
##                                                                   12.9
##                                              Less than high school degree
##                                                                    0.3
##                                                           Master's degree
##                                                                    9.4
##                                              Professional degree (JD, MD)
##                                                                    2.1
##                                                Some college but no degree
##                                                                   18.1
```

Look at how accuracy varies according to education level:

```
(xtab <- table(df$Ed, df$Accuracy))                     # Raw stats
```

```
##
##                                                                          wrong
##   Associate degree in college (2-year)                                     37
##   Bachelor's degree in college (4-year)                                   177
##   Doctoral degree                                                           4
##   High school graduate (high school diploma or equivalent including GED)   51
##   Less than high school degree                                              0
##   Master's degree                                                          24
##   Professional degree (JD, MD)                                              7
##   Some college but no degree                                               61
##
##                                                                          right
##   Associate degree in college (2-year)                                     87
##   Bachelor's degree in college (4-year)                                   347
##   Doctoral degree                                                           4
##   High school graduate (high school diploma or equivalent including GED)   97
##   Less than high school degree                                              4
##   Master's degree                                                          84
##   Professional degree (JD, MD)                                             17
##   Some college but no degree                                              147
```

```r
(xtab <- round(prop.table(xtab, 1), 3) * 100)           # Proportions
```

```
##
##                                                                    wrong
##   Associate degree in college (2-year)                              29.8
##   Bachelor's degree in college (4-year)                             33.8
##   Doctoral degree                                                   50.0
##   High school graduate (high school diploma or equivalent including GED)  34.5
##   Less than high school degree                                       0.0
##   Master's degree                                                   22.2
##   Professional degree (JD, MD)                                      29.2
##   Some college but no degree                                        29.3
##
##                                                                    right
##   Associate degree in college (2-year)                              70.2
##   Bachelor's degree in college (4-year)                             66.2
##   Doctoral degree                                                   50.0
##   High school graduate (high school diploma or equivalent including GED)  65.5
##   Less than high school degree                                     100.0
##   Master's degree                                                   77.8
##   Professional degree (JD, MD)                                      70.8
##   Some college but no degree                                        70.7
```

Look at how response time varied according to education level:

```r
df_RT %>%
  group_by(Ed) %>%
  summarise(mean(as.numeric(Measurement))) %>%
  arrange(desc(`mean(as.numeric(Measurement))`))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 8 x 2
##   Ed                                                   `mean(as.numeric(Measurem~
##   <chr>                                                                     <dbl>
## 1 Doctoral degree                                                            5.93
## 2 Less than high school degree                                               5.40
## 3 Some college but no degree                                                 4.77
## 4 Master's degree                                                            4.71
## 5 Associate degree in college (2-year)                                       4.71
## 6 Professional degree (JD, MD)                                               4.34
## 7 Bachelor's degree in college (4-year)                                      4.25
## 8 High school graduate (high school diploma or equiv~                        4.16
```

Run Model 1 but with an interaction with Ed entered for each of the predictors, to see if Education modulates any of the effects:

```r
# Turn variables into factors:
df$Ed <- factor(df$Ed)

# Run model:
xmdl <- brm(Accuracy ~ (AxisInversion * Ed) +
                       (Orientation * Ed) +
                       (Valence * Ed) +
                (1 + Valence|Subject),
                data = df,
```

```r
                 family = bernoulli,
                 init = 0,
                 chains = 4,
                 warmup = 2000,
                 iter = 4000,
                 prior = my_priors,
                 control = my_controls,
                 seed = 13)

# Summary of model:
summary(xmdl)
```

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: Accuracy ~ (AxisInversion * Ed) + (Orientation * Ed) + (Valence * Ed) + (1 + Valence | Subje
##    Data: df (Number of observations: 1148)
## Samples: 4 chains, each with iter = 4000; warmup = 2000; thin = 1;
##          total post-warmup samples = 8000
##
## Group-Level Effects:
## ~Subject (Number of levels: 287)
##                                 Estimate Est.Error l-95% CI u-95% CI Rhat
## sd(Intercept)                       3.92      0.52     2.95     5.01 1.00
## sd(Valencepositive)                 4.34      0.56     3.34     5.51 1.00
## cor(Intercept,Valencepositive)     -0.50      0.14    -0.73    -0.19 1.00
##                                 Bulk_ESS Tail_ESS
## sd(Intercept)                       2232     3377
## sd(Valencepositive)                 2910     4599
## cor(Intercept,Valencepositive)      1986     3059
##
## Population-Level Effects:
##                                                                                    Estimate
## Intercept                                                                              2.20
## AxisInversionnormal                                                                    1.75
## EdBachelorsdegreeincollege4Myear                                                      -0.26
## EdDoctoraldegree                                                                      -0.47
## EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED                         -0.51
## EdLessthanhighschooldegree                                                             0.76
## EdMastersdegree                                                                        0.17
## EdProfessionaldegreeJDMD                                                               0.68
## EdSomecollegebutnodegree                                                              -0.58
## Orientationquant_y                                                                    -1.72
## Valencepositive                                                                        0.92
## AxisInversionnormal:EdBachelorsdegreeincollege4Myear                                   1.06
## AxisInversionnormal:EdDoctoraldegree                                                   1.00
## AxisInversionnormal:EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED     -0.59
## AxisInversionnormal:EdLessthanhighschooldegree                                        -0.01
## AxisInversionnormal:EdMastersdegree                                                    0.53
## AxisInversionnormal:EdProfessionaldegreeJDMD                                           1.36
## AxisInversionnormal:EdSomecollegebutnodegree                                           0.95
## EdBachelorsdegreeincollege4Myear:Orientationquant_y                                   -0.63
## EdDoctoraldegree:Orientationquant_y                                                   -0.47
## EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED:Orientationquant_y       0.58
## EdLessthanhighschooldegree:Orientationquant_y                                         -0.02
```

```
## EdMastersdegree:Orientationquant_y                                                     0.55
## EdProfessionaldegreeJDMD:Orientationquant_y                                           -0.16
## EdSomecollegebutnodegree:Orientationquant_y                                           -0.94
## EdBachelorsdegreeincollege4Myear:Valencepositive                                      -0.48
## EdDoctoraldegree:Valencepositive                                                      -0.30
## EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED:Valencepositive         -0.02
## EdLessthanhighschooldegree:Valencepositive                                             0.35
## EdMastersdegree:Valencepositive                                                        0.35
## EdProfessionaldegreeJDMD:Valencepositive                                              -0.48
## EdSomecollegebutnodegree:Valencepositive                                               1.54
##                                                                                   Est.Error
## Intercept                                                                              0.75
## AxisInversionnormal                                                                    0.92
## EdBachelorsdegreeincollege4Myear                                                       0.86
## EdDoctoraldegree                                                                       1.79
## EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED                          1.06
## EdLessthanhighschooldegree                                                             1.85
## EdMastersdegree                                                                        1.21
## EdProfessionaldegreeJDMD                                                               1.58
## EdSomecollegebutnodegree                                                               0.99
## Orientationquant_y                                                                     0.82
## Valencepositive                                                                        0.77
## AxisInversionnormal:EdBachelorsdegreeincollege4Myear                                   1.06
## AxisInversionnormal:EdDoctoraldegree                                                   1.91
## AxisInversionnormal:EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED      1.32
## AxisInversionnormal:EdLessthanhighschooldegree                                         1.97
## AxisInversionnormal:EdMastersdegree                                                    1.36
## AxisInversionnormal:EdProfessionaldegreeJDMD                                           1.79
## AxisInversionnormal:EdSomecollegebutnodegree                                           1.19
## EdBachelorsdegreeincollege4Myear:Orientationquant_y                                    0.98
## EdDoctoraldegree:Orientationquant_y                                                    1.81
## EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED:Orientationquant_y       1.20
## EdLessthanhighschooldegree:Orientationquant_y                                          1.98
## EdMastersdegree:Orientationquant_y                                                     1.30
## EdProfessionaldegreeJDMD:Orientationquant_y                                            1.67
## EdSomecollegebutnodegree:Orientationquant_y                                            1.17
## EdBachelorsdegreeincollege4Myear:Valencepositive                                       0.85
## EdDoctoraldegree:Valencepositive                                                       1.82
## EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED:Valencepositive          1.04
## EdLessthanhighschooldegree:Valencepositive                                             1.89
## EdMastersdegree:Valencepositive                                                        1.18
## EdProfessionaldegreeJDMD:Valencepositive                                               1.59
## EdSomecollegebutnodegree:Valencepositive                                               1.05
##                                                                                   l-95% CI
## Intercept                                                                              0.74
## AxisInversionnormal                                                                   -0.04
## EdBachelorsdegreeincollege4Myear                                                      -1.96
## EdDoctoraldegree                                                                      -3.93
## EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED                         -2.61
## EdLessthanhighschooldegree                                                            -2.87
## EdMastersdegree                                                                       -2.18
## EdProfessionaldegreeJDMD                                                              -2.49
## EdSomecollegebutnodegree                                                              -2.52
## Orientationquant_y                                                                    -3.34
```

```
## Valencepositive                                                                        -0.55
## AxisInversionnormal:EdBachelorsdegreeincollege4Myear                                   -1.00
## AxisInversionnormal:EdDoctoraldegree                                                   -2.73
## AxisInversionnormal:EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED     -3.21
## AxisInversionnormal:EdLessthanhighschooldegree                                         -3.88
## AxisInversionnormal:EdMastersdegree                                                    -2.15
## AxisInversionnormal:EdProfessionaldegreeJDMD                                           -2.20
## AxisInversionnormal:EdSomecollegebutnodegree                                           -1.38
## EdBachelorsdegreeincollege4Myear:Orientationquant_y                                    -2.53
## EdDoctoraldegree:Orientationquant_y                                                    -3.96
## EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED:Orientationquant_y      -1.74
## EdLessthanhighschooldegree:Orientationquant_y                                          -3.88
## EdMastersdegree:Orientationquant_y                                                     -2.02
## EdProfessionaldegreeJDMD:Orientationquant_y                                            -3.51
## EdSomecollegebutnodegree:Orientationquant_y                                            -3.20
## EdBachelorsdegreeincollege4Myear:Valencepositive                                       -2.12
## EdDoctoraldegree:Valencepositive                                                       -3.81
## EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED:Valencepositive         -2.05
## EdLessthanhighschooldegree:Valencepositive                                             -3.35
## EdMastersdegree:Valencepositive                                                        -1.95
## EdProfessionaldegreeJDMD:Valencepositive                                               -3.61
## EdSomecollegebutnodegree:Valencepositive                                               -0.49
##                                                                                     u-95% CI
## Intercept                                                                               3.70
## AxisInversionnormal                                                                     3.54
## EdBachelorsdegreeincollege4Myear                                                        1.48
## EdDoctoraldegree                                                                        2.96
## EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED                           1.63
## EdLessthanhighschooldegree                                                              4.53
## EdMastersdegree                                                                         2.54
## EdProfessionaldegreeJDMD                                                                3.73
## EdSomecollegebutnodegree                                                                1.40
## Orientationquant_y                                                                     -0.15
## Valencepositive                                                                         2.46
## AxisInversionnormal:EdBachelorsdegreeincollege4Myear                                    3.13
## AxisInversionnormal:EdDoctoraldegree                                                    4.75
## AxisInversionnormal:EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED      2.02
## AxisInversionnormal:EdLessthanhighschooldegree                                          3.90
## AxisInversionnormal:EdMastersdegree                                                     3.22
## AxisInversionnormal:EdProfessionaldegreeJDMD                                            4.90
## AxisInversionnormal:EdSomecollegebutnodegree                                            3.31
## EdBachelorsdegreeincollege4Myear:Orientationquant_y                                     1.32
## EdDoctoraldegree:Orientationquant_y                                                     3.08
## EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED:Orientationquant_y       2.95
## EdLessthanhighschooldegree:Orientationquant_y                                           3.79
## EdMastersdegree:Orientationquant_y                                                      3.11
## EdProfessionaldegreeJDMD:Orientationquant_y                                             3.14
## EdSomecollegebutnodegree:Orientationquant_y                                             1.37
## EdBachelorsdegreeincollege4Myear:Valencepositive                                        1.20
## EdDoctoraldegree:Valencepositive                                                        3.29
## EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED:Valencepositive          2.09
## EdLessthanhighschooldegree:Valencepositive                                              4.11
## EdMastersdegree:Valencepositive                                                         2.70
## EdProfessionaldegreeJDMD:Valencepositive                                                2.64
```

```
## EdSomecollegebutnodegree:Valencepositive                                  3.64
##                                                                         Rhat
## Intercept                                                               1.00
## AxisInversionnormal                                                     1.00
## EdBachelorsdegreeincollege4Myear                                        1.00
## EdDoctoraldegree                                                        1.00
## EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED           1.00
## EdLessthanhighschooldegree                                              1.00
## EdMastersdegree                                                         1.00
## EdProfessionaldegreeJDMD                                                1.00
## EdSomecollegebutnodegree                                                1.00
## Orientationquant_y                                                      1.00
## Valencepositive                                                         1.00
## AxisInversionnormal:EdBachelorsdegreeincollege4Myear                    1.00
## AxisInversionnormal:EdDoctoraldegree                                    1.00
## AxisInversionnormal:EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED 1.00
## AxisInversionnormal:EdLessthanhighschooldegree                          1.00
## AxisInversionnormal:EdMastersdegree                                     1.00
## AxisInversionnormal:EdProfessionaldegreeJDMD                            1.00
## AxisInversionnormal:EdSomecollegebutnodegree                            1.00
## EdBachelorsdegreeincollege4Myear:Orientationquant_y                     1.00
## EdDoctoraldegree:Orientationquant_y                                     1.00
## EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED:Orientationquant_y  1.00
## EdLessthanhighschooldegree:Orientationquant_y                           1.00
## EdMastersdegree:Orientationquant_y                                      1.00
## EdProfessionaldegreeJDMD:Orientationquant_y                             1.00
## EdSomecollegebutnodegree:Orientationquant_y                             1.00
## EdBachelorsdegreeincollege4Myear:Valencepositive                        1.00
## EdDoctoraldegree:Valencepositive                                        1.00
## EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED:Valencepositive     1.00
## EdLessthanhighschooldegree:Valencepositive                              1.00
## EdMastersdegree:Valencepositive                                         1.00
## EdProfessionaldegreeJDMD:Valencepositive                                1.00
## EdSomecollegebutnodegree:Valencepositive                                1.00
##                                                                         Bulk_ESS
## Intercept                                                                   4043
## AxisInversionnormal                                                         3464
## EdBachelorsdegreeincollege4Myear                                            3923
## EdDoctoraldegree                                                           13071
## EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED               4227
## EdLessthanhighschooldegree                                                 13763
## EdMastersdegree                                                             5074
## EdProfessionaldegreeJDMD                                                   10114
## EdSomecollegebutnodegree                                                    4335
## Orientationquant_y                                                          3825
## Valencepositive                                                             3281
## AxisInversionnormal:EdBachelorsdegreeincollege4Myear                        4571
## AxisInversionnormal:EdDoctoraldegree                                       15130
## AxisInversionnormal:EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED     5434
## AxisInversionnormal:EdLessthanhighschooldegree                             16321
## AxisInversionnormal:EdMastersdegree                                         6020
## AxisInversionnormal:EdProfessionaldegreeJDMD                               11621
## AxisInversionnormal:EdSomecollegebutnodegree                                5096
## EdBachelorsdegreeincollege4Myear:Orientationquant_y                         3870
```

```
## EdDoctoraldegree:Orientationquant_y                                           14469
## EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED:Orientationquant_y  4998
## EdLessthanhighschooldegree:Orientationquant_y                                 17760
## EdMastersdegree:Orientationquant_y                                             5696
## EdProfessionaldegreeJDMD:Orientationquant_y                                    9660
## EdSomecollegebutnodegree:Orientationquant_y                                    4778
## EdBachelorsdegreeincollege4Myear:Valencepositive                               3932
## EdDoctoraldegree:Valencepositive                                              13556
## EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED:Valencepositive  5164
## EdLessthanhighschooldegree:Valencepositive                                    17188
## EdMastersdegree:Valencepositive                                                5574
## EdProfessionaldegreeJDMD:Valencepositive                                       9540
## EdSomecollegebutnodegree:Valencepositive                                       5037
##                                                                             Tail_ESS
## Intercept                                                                      5646
## AxisInversionnormal                                                            5294
## EdBachelorsdegreeincollege4Myear                                               4889
## EdDoctoraldegree                                                               6252
## EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED                  4807
## EdLessthanhighschooldegree                                                     6052
## EdMastersdegree                                                                5311
## EdProfessionaldegreeJDMD                                                       6574
## EdSomecollegebutnodegree                                                       5567
## Orientationquant_y                                                             5190
## Valencepositive                                                                5170
## AxisInversionnormal:EdBachelorsdegreeincollege4Myear                           5582
## AxisInversionnormal:EdDoctoraldegree                                           6261
## AxisInversionnormal:EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED  5820
## AxisInversionnormal:EdLessthanhighschooldegree                                 5523
## AxisInversionnormal:EdMastersdegree                                            6163
## AxisInversionnormal:EdProfessionaldegreeJDMD                                   6021
## AxisInversionnormal:EdSomecollegebutnodegree                                   6038
## EdBachelorsdegreeincollege4Myear:Orientationquant_y                            5197
## EdDoctoraldegree:Orientationquant_y                                            5460
## EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED:Orientationquant_y  6128
## EdLessthanhighschooldegree:Orientationquant_y                                  5505
## EdMastersdegree:Orientationquant_y                                             6195
## EdProfessionaldegreeJDMD:Orientationquant_y                                    6237
## EdSomecollegebutnodegree:Orientationquant_y                                    5571
## EdBachelorsdegreeincollege4Myear:Valencepositive                               5210
## EdDoctoraldegree:Valencepositive                                               6493
## EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED:Valencepositive  5754
## EdLessthanhighschooldegree:Valencepositive                                     5888
## EdMastersdegree:Valencepositive                                                5919
## EdProfessionaldegreeJDMD:Valencepositive                                       6713
## EdSomecollegebutnodegree:Valencepositive                                       5843
##
## Samples were drawn 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).
```

```
# Posterior predictive checks:
# pp_check(xmdl)
```

None of the interactions with the Ed predictor were significant (their 95% credible intervals all contained zero).

Run Model 2, which tests the effect of vertical valence alignment on response accuracy, except this time, include an interaction with Ed to see if this modulates the effects:

```
# Create copies of relevant predictors:
df_y$Ed_c <- as.factor(df_y$Ed)
contrasts(df_y$Ed_c) <- contr.sum(7) / 2

# Run model:
y_mdl <- brm(Accuracy ~ (AxisInversion_c * Valence_c) * Ed_c +
                (1 + Valence_c|Subject),
                data = df_y,
                family = bernoulli,
                init = 0,
                chains = 4,
                warmup = 2000,
                iter = 4000,
                prior = my_priors,
                control = my_controls,
                seed = 13)
```

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

```
## Start sampling
```

```
# Summary of model:
summary(y_mdl)
```

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: Accuracy ~ (AxisInversion_c * Valence_c) * Ed_c + (1 + Valence_c | Subject)
##    Data: df_y (Number of observations: 580)
## Samples: 4 chains, each with iter = 4000; warmup = 2000; thin = 1;
##          total post-warmup samples = 8000
##
## Group-Level Effects:
## ~Subject (Number of levels: 145)
##                            Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
## sd(Intercept)                  3.71      0.52     2.81     4.84 1.00     3258
## sd(Valence_c1)                 4.23      0.72     2.91     5.72 1.00     3176
## cor(Intercept,Valence_c1)     -0.05      0.24    -0.50     0.43 1.00     2688
##                            Tail_ESS
## sd(Intercept)                  5206
## sd(Valence_c1)                 4824
## cor(Intercept,Valence_c1)      4512
##
## Population-Level Effects:
##                                    Estimate Est.Error l-95% CI u-95% CI Rhat
## Intercept                              2.44      0.53     1.45     3.51 1.00
## AxisInversion_c1                       5.05      0.90     3.35     6.90 1.00
## Valence_c1                             1.18      0.72    -0.24     2.55 1.00
## Ed_c1                                  0.34      1.44    -2.49     3.21 1.00
## Ed_c2                                 -0.93      1.14    -3.18     1.33 1.00
## Ed_c3                                 -0.23      1.82    -3.76     3.28 1.00
## Ed_c4                                  0.60      1.38    -2.08     3.24 1.00
## Ed_c5                                  1.00      1.42    -1.79     3.82 1.00
```

```
## Ed_c6                                 0.30      1.70    -3.08     3.64 1.00
## AxisInversion_c1:Valence_c1           4.27      1.22     1.94     6.71 1.00
## AxisInversion_c1:Ed_c1               -0.78      1.79    -4.25     2.76 1.00
## AxisInversion_c1:Ed_c2                0.52      1.56    -2.52     3.59 1.00
## AxisInversion_c1:Ed_c3               -0.26      1.93    -4.05     3.57 1.00
## AxisInversion_c1:Ed_c4               -0.72      1.75    -4.12     2.70 1.00
## AxisInversion_c1:Ed_c5               -0.96      1.76    -4.42     2.45 1.00
## AxisInversion_c1:Ed_c6               -0.30      1.87    -3.97     3.47 1.00
## Valence_c1:Ed_c1                     -0.80      1.60    -3.96     2.34 1.00
## Valence_c1:Ed_c2                     -0.46      1.38    -3.22     2.23 1.00
## Valence_c1:Ed_c3                     -0.74      1.87    -4.41     2.93 1.00
## Valence_c1:Ed_c4                     -0.98      1.57    -4.06     2.16 1.00
## Valence_c1:Ed_c5                     -0.63      1.61    -3.85     2.48 1.00
## Valence_c1:Ed_c6                     -0.94      1.80    -4.45     2.53 1.00
## AxisInversion_c1:Valence_c1:Ed_c1     0.01      1.88    -3.66     3.68 1.00
## AxisInversion_c1:Valence_c1:Ed_c2     0.44      1.75    -2.98     3.80 1.00
## AxisInversion_c1:Valence_c1:Ed_c3    -0.28      1.94    -4.07     3.50 1.00
## AxisInversion_c1:Valence_c1:Ed_c4     0.06      1.86    -3.56     3.70 1.00
## AxisInversion_c1:Valence_c1:Ed_c5     0.56      1.87    -3.11     4.19 1.00
## AxisInversion_c1:Valence_c1:Ed_c6    -0.21      1.91    -3.98     3.51 1.00
##                                    Bulk_ESS Tail_ESS
## Intercept                              3242     4187
## AxisInversion_c1                       3704     5172
## Valence_c1                             4522     5558
## Ed_c1                                  4889     5489
## Ed_c2                                  3636     5038
## Ed_c3                                  7431     5501
## Ed_c4                                  4426     5159
## Ed_c5                                  3955     5309
## Ed_c6                                  6464     5671
## AxisInversion_c1:Valence_c1            5088     6133
## AxisInversion_c1:Ed_c1                 7904     6529
## AxisInversion_c1:Ed_c2                 5264     6156
## AxisInversion_c1:Ed_c3                 8924     5488
## AxisInversion_c1:Ed_c4                 7915     6303
## AxisInversion_c1:Ed_c5                 6463     5713
## AxisInversion_c1:Ed_c6                 8160     5728
## Valence_c1:Ed_c1                       7297     5567
## Valence_c1:Ed_c2                       5468     5329
## Valence_c1:Ed_c3                       9478     6138
## Valence_c1:Ed_c4                       7380     6068
## Valence_c1:Ed_c5                       6650     6006
## Valence_c1:Ed_c6                       9000     5914
## AxisInversion_c1:Valence_c1:Ed_c1      9578     5927
## AxisInversion_c1:Valence_c1:Ed_c2      8092     5691
## AxisInversion_c1:Valence_c1:Ed_c3      9812     6159
## AxisInversion_c1:Valence_c1:Ed_c4      9342     6034
## AxisInversion_c1:Valence_c1:Ed_c5      9931     6302
## AxisInversion_c1:Valence_c1:Ed_c6      9536     6024
##
## Samples were drawn 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).
```

```r
# Posterior predictive checks:
# pp_check(y_mdl)
```

None of the interactions with the Ed predictor were significant (their 95% credible intervals all contained zero).

Run Model 3, which tests the effect of horizontal valence alignment on response accuracy, except this time, include an interaction with Ed to see if this modulates the effects:

```r
# Create copies of relevant predictors:
df_x$Ed_c <- as.factor(df_x$Ed)

# Deviation code these predictors:
contrasts(df_x$Ed_c) <- contr.sum(7) / 2

# Run model:
x_mdl <- brm(Accuracy ~ ((AxisInversion_c * Valence_c) * Ed) +
                        ((AxisInversion_c * Valence_c * Handed_c) * Ed) +
                        (1 + Valence_c|Subject),
                data = df_x,
                family = bernoulli,
                init = 0,
                chains = 4,
                warmup = 2000,
                iter = 4000,
                prior = my_priors,
                control = my_controls,
                seed = 13)
```

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

```
## Start sampling
```

```r
# Summary of model:
summary(x_mdl)
```

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: Accuracy ~ ((AxisInversion_c * Valence_c) * Ed) + ((AxisInversion_c * Valence_c * Handed_c)
##    Data: df_x (Number of observations: 568)
## Samples: 4 chains, each with iter = 4000; warmup = 2000; thin = 1;
##          total post-warmup samples = 8000
##
## Group-Level Effects:
## ~Subject (Number of levels: 142)
##                          Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
## sd(Intercept)                2.43      0.40     1.71     3.28 1.00     3128
## sd(Valence_c1)               3.32      0.67     2.10     4.69 1.00     3205
## cor(Intercept,Valence_c1)    0.31      0.25    -0.22     0.73 1.00     2970
##                          Tail_ESS
## sd(Intercept)                5399
## sd(Valence_c1)               5097
## cor(Intercept,Valence_c1)    4642
##
## Population-Level Effects:
##
```

```
## Intercept
## AxisInversion_c1
## Valence_c1
## EdBachelorsdegreeincollege4Myear
## EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED
## EdLessthanhighschooldegree
## EdMastersdegree
## EdProfessionaldegreeJDMD
## EdSomecollegebutnodegree
## Handed_c1
## AxisInversion_c1:Valence_c1
## AxisInversion_c1:EdBachelorsdegreeincollege4Myear
## AxisInversion_c1:EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED
## AxisInversion_c1:EdLessthanhighschooldegree
## AxisInversion_c1:EdMastersdegree
## AxisInversion_c1:EdProfessionaldegreeJDMD
## AxisInversion_c1:EdSomecollegebutnodegree
## Valence_c1:EdBachelorsdegreeincollege4Myear
## Valence_c1:EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED
## Valence_c1:EdLessthanhighschooldegree
## Valence_c1:EdMastersdegree
## Valence_c1:EdProfessionaldegreeJDMD
## Valence_c1:EdSomecollegebutnodegree
## AxisInversion_c1:Handed_c1
## Valence_c1:Handed_c1
## EdBachelorsdegreeincollege4Myear:Handed_c1
## EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED:Handed_c1
## EdLessthanhighschooldegree:Handed_c1
## EdMastersdegree:Handed_c1
## EdProfessionaldegreeJDMD:Handed_c1
## EdSomecollegebutnodegree:Handed_c1
## AxisInversion_c1:Valence_c1:EdBachelorsdegreeincollege4Myear
## AxisInversion_c1:Valence_c1:EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED
## AxisInversion_c1:Valence_c1:EdLessthanhighschooldegree
## AxisInversion_c1:Valence_c1:EdMastersdegree
## AxisInversion_c1:Valence_c1:EdProfessionaldegreeJDMD
## AxisInversion_c1:Valence_c1:EdSomecollegebutnodegree
## AxisInversion_c1:Valence_c1:Handed_c1
## AxisInversion_c1:EdBachelorsdegreeincollege4Myear:Handed_c1
## AxisInversion_c1:EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED:Handed_c1
## AxisInversion_c1:EdLessthanhighschooldegree:Handed_c1
## AxisInversion_c1:EdMastersdegree:Handed_c1
## AxisInversion_c1:EdProfessionaldegreeJDMD:Handed_c1
## AxisInversion_c1:EdSomecollegebutnodegree:Handed_c1
## Valence_c1:EdBachelorsdegreeincollege4Myear:Handed_c1
## Valence_c1:EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED:Handed_c1
## Valence_c1:EdLessthanhighschooldegree:Handed_c1
## Valence_c1:EdMastersdegree:Handed_c1
## Valence_c1:EdProfessionaldegreeJDMD:Handed_c1
## Valence_c1:EdSomecollegebutnodegree:Handed_c1
## AxisInversion_c1:Valence_c1:EdBachelorsdegreeincollege4Myear:Handed_c1
## AxisInversion_c1:Valence_c1:EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED:Handed_c1
## AxisInversion_c1:Valence_c1:EdLessthanhighschooldegree:Handed_c1
## AxisInversion_c1:Valence_c1:EdMastersdegree:Handed_c1
```

```
## AxisInversion_c1:Valence_c1:EdProfessionaldegreeJDMD:Handed_c1
## AxisInversion_c1:Valence_c1:EdSomecollegebutnodegree:Handed_c1
##                                                                                                  I
## Intercept
## AxisInversion_c1
## Valence_c1
## EdBachelorsdegreeincollege4Myear
## EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED
## EdLessthanhighschooldegree
## EdMastersdegree
## EdProfessionaldegreeJDMD
## EdSomecollegebutnodegree
## Handed_c1
## AxisInversion_c1:Valence_c1
## AxisInversion_c1:EdBachelorsdegreeincollege4Myear
## AxisInversion_c1:EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED
## AxisInversion_c1:EdLessthanhighschooldegree
## AxisInversion_c1:EdMastersdegree
## AxisInversion_c1:EdProfessionaldegreeJDMD
## AxisInversion_c1:EdSomecollegebutnodegree
## Valence_c1:EdBachelorsdegreeincollege4Myear
## Valence_c1:EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED
## Valence_c1:EdLessthanhighschooldegree
## Valence_c1:EdMastersdegree
## Valence_c1:EdProfessionaldegreeJDMD
## Valence_c1:EdSomecollegebutnodegree
## AxisInversion_c1:Handed_c1
## Valence_c1:Handed_c1
## EdBachelorsdegreeincollege4Myear:Handed_c1
## EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED:Handed_c1
## EdLessthanhighschooldegree:Handed_c1
## EdMastersdegree:Handed_c1
## EdProfessionaldegreeJDMD:Handed_c1
## EdSomecollegebutnodegree:Handed_c1
## AxisInversion_c1:Valence_c1:EdBachelorsdegreeincollege4Myear
## AxisInversion_c1:Valence_c1:EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED
## AxisInversion_c1:Valence_c1:EdLessthanhighschooldegree
## AxisInversion_c1:Valence_c1:EdMastersdegree
## AxisInversion_c1:Valence_c1:EdProfessionaldegreeJDMD
## AxisInversion_c1:Valence_c1:EdSomecollegebutnodegree
## AxisInversion_c1:Valence_c1:Handed_c1
## AxisInversion_c1:EdBachelorsdegreeincollege4Myear:Handed_c1
## AxisInversion_c1:EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED:Handed_c1
## AxisInversion_c1:EdLessthanhighschooldegree:Handed_c1
## AxisInversion_c1:EdMastersdegree:Handed_c1
## AxisInversion_c1:EdProfessionaldegreeJDMD:Handed_c1
## AxisInversion_c1:EdSomecollegebutnodegree:Handed_c1
## Valence_c1:EdBachelorsdegreeincollege4Myear:Handed_c1
## Valence_c1:EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED:Handed_c1
## Valence_c1:EdLessthanhighschooldegree:Handed_c1
## Valence_c1:EdMastersdegree:Handed_c1
## Valence_c1:EdProfessionaldegreeJDMD:Handed_c1
## Valence_c1:EdSomecollegebutnodegree:Handed_c1
## AxisInversion_c1:Valence_c1:EdBachelorsdegreeincollege4Myear:Handed_c1
```

```
## AxisInversion_c1:Valence_c1:EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED:Handed_c1
## AxisInversion_c1:Valence_c1:EdLessthanhighschooldegree:Handed_c1
## AxisInversion_c1:Valence_c1:EdMastersdegree:Handed_c1
## AxisInversion_c1:Valence_c1:EdProfessionaldegreeJDMD:Handed_c1
## AxisInversion_c1:Valence_c1:EdSomecollegebutnodegree:Handed_c1
##
## Intercept
## AxisInversion_c1
## Valence_c1
## EdBachelorsdegreeincollege4Myear
## EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED
## EdLessthanhighschooldegree
## EdMastersdegree
## EdProfessionaldegreeJDMD
## EdSomecollegebutnodegree
## Handed_c1
## AxisInversion_c1:Valence_c1
## AxisInversion_c1:EdBachelorsdegreeincollege4Myear
## AxisInversion_c1:EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED
## AxisInversion_c1:EdLessthanhighschooldegree
## AxisInversion_c1:EdMastersdegree
## AxisInversion_c1:EdProfessionaldegreeJDMD
## AxisInversion_c1:EdSomecollegebutnodegree
## Valence_c1:EdBachelorsdegreeincollege4Myear
## Valence_c1:EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED
## Valence_c1:EdLessthanhighschooldegree
## Valence_c1:EdMastersdegree
## Valence_c1:EdProfessionaldegreeJDMD
## Valence_c1:EdSomecollegebutnodegree
## AxisInversion_c1:Handed_c1
## Valence_c1:Handed_c1
## EdBachelorsdegreeincollege4Myear:Handed_c1
## EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED:Handed_c1
## EdLessthanhighschooldegree:Handed_c1
## EdMastersdegree:Handed_c1
## EdProfessionaldegreeJDMD:Handed_c1
## EdSomecollegebutnodegree:Handed_c1
## AxisInversion_c1:Valence_c1:EdBachelorsdegreeincollege4Myear
## AxisInversion_c1:Valence_c1:EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED
## AxisInversion_c1:Valence_c1:EdLessthanhighschooldegree
## AxisInversion_c1:Valence_c1:EdMastersdegree
## AxisInversion_c1:Valence_c1:EdProfessionaldegreeJDMD
## AxisInversion_c1:Valence_c1:EdSomecollegebutnodegree
## AxisInversion_c1:Valence_c1:Handed_c1
## AxisInversion_c1:EdBachelorsdegreeincollege4Myear:Handed_c1
## AxisInversion_c1:EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED:Handed_c1
## AxisInversion_c1:EdLessthanhighschooldegree:Handed_c1
## AxisInversion_c1:EdMastersdegree:Handed_c1
## AxisInversion_c1:EdProfessionaldegreeJDMD:Handed_c1
## AxisInversion_c1:EdSomecollegebutnodegree:Handed_c1
## Valence_c1:EdBachelorsdegreeincollege4Myear:Handed_c1
## Valence_c1:EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED:Handed_c1
## Valence_c1:EdLessthanhighschooldegree:Handed_c1
## Valence_c1:EdMastersdegree:Handed_c1
```

```
## Valence_c1:EdProfessionaldegreeJDMD:Handed_c1
## Valence_c1:EdSomecollegebutnodegree:Handed_c1
## AxisInversion_c1:Valence_c1:EdBachelorsdegreeincollege4Myear:Handed_c1
## AxisInversion_c1:Valence_c1:EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED:Handed_c1
## AxisInversion_c1:Valence_c1:EdLessthanhighschooldegree:Handed_c1
## AxisInversion_c1:Valence_c1:EdMastersdegree:Handed_c1
## AxisInversion_c1:Valence_c1:EdProfessionaldegreeJDMD:Handed_c1
## AxisInversion_c1:Valence_c1:EdSomecollegebutnodegree:Handed_c1
##
## Intercept
## AxisInversion_c1
## Valence_c1
## EdBachelorsdegreeincollege4Myear
## EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED
## EdLessthanhighschooldegree
## EdMastersdegree
## EdProfessionaldegreeJDMD
## EdSomecollegebutnodegree
## Handed_c1
## AxisInversion_c1:Valence_c1
## AxisInversion_c1:EdBachelorsdegreeincollege4Myear
## AxisInversion_c1:EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED
## AxisInversion_c1:EdLessthanhighschooldegree
## AxisInversion_c1:EdMastersdegree
## AxisInversion_c1:EdProfessionaldegreeJDMD
## AxisInversion_c1:EdSomecollegebutnodegree
## Valence_c1:EdBachelorsdegreeincollege4Myear
## Valence_c1:EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED
## Valence_c1:EdLessthanhighschooldegree
## Valence_c1:EdMastersdegree
## Valence_c1:EdProfessionaldegreeJDMD
## Valence_c1:EdSomecollegebutnodegree
## AxisInversion_c1:Handed_c1
## Valence_c1:Handed_c1
## EdBachelorsdegreeincollege4Myear:Handed_c1
## EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED:Handed_c1
## EdLessthanhighschooldegree:Handed_c1
## EdMastersdegree:Handed_c1
## EdProfessionaldegreeJDMD:Handed_c1
## EdSomecollegebutnodegree:Handed_c1
## AxisInversion_c1:Valence_c1:EdBachelorsdegreeincollege4Myear
## AxisInversion_c1:Valence_c1:EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED
## AxisInversion_c1:Valence_c1:EdLessthanhighschooldegree
## AxisInversion_c1:Valence_c1:EdMastersdegree
## AxisInversion_c1:Valence_c1:EdProfessionaldegreeJDMD
## AxisInversion_c1:Valence_c1:EdSomecollegebutnodegree
## AxisInversion_c1:Valence_c1:Handed_c1
## AxisInversion_c1:EdBachelorsdegreeincollege4Myear:Handed_c1
## AxisInversion_c1:EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED:Handed_c1
## AxisInversion_c1:EdLessthanhighschooldegree:Handed_c1
## AxisInversion_c1:EdMastersdegree:Handed_c1
## AxisInversion_c1:EdProfessionaldegreeJDMD:Handed_c1
## AxisInversion_c1:EdSomecollegebutnodegree:Handed_c1
## Valence_c1:EdBachelorsdegreeincollege4Myear:Handed_c1
```

```
## Valence_c1:EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED:Handed_c1
## Valence_c1:EdLessthanhighschooldegree:Handed_c1
## Valence_c1:EdMastersdegree:Handed_c1
## Valence_c1:EdProfessionaldegreeJDMD:Handed_c1
## Valence_c1:EdSomecollegebutnodegree:Handed_c1
## AxisInversion_c1:Valence_c1:EdBachelorsdegreeincollege4Myear:Handed_c1
## AxisInversion_c1:Valence_c1:EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED:Handed_c1
## AxisInversion_c1:Valence_c1:EdLessthanhighschooldegree:Handed_c1
## AxisInversion_c1:Valence_c1:EdMastersdegree:Handed_c1
## AxisInversion_c1:Valence_c1:EdProfessionaldegreeJDMD:Handed_c1
## AxisInversion_c1:Valence_c1:EdSomecollegebutnodegree:Handed_c1
##
## Intercept
## AxisInversion_c1
## Valence_c1
## EdBachelorsdegreeincollege4Myear
## EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED
## EdLessthanhighschooldegree
## EdMastersdegree
## EdProfessionaldegreeJDMD
## EdSomecollegebutnodegree
## Handed_c1
## AxisInversion_c1:Valence_c1
## AxisInversion_c1:EdBachelorsdegreeincollege4Myear
## AxisInversion_c1:EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED
## AxisInversion_c1:EdLessthanhighschooldegree
## AxisInversion_c1:EdMastersdegree
## AxisInversion_c1:EdProfessionaldegreeJDMD
## AxisInversion_c1:EdSomecollegebutnodegree
## Valence_c1:EdBachelorsdegreeincollege4Myear
## Valence_c1:EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED
## Valence_c1:EdLessthanhighschooldegree
## Valence_c1:EdMastersdegree
## Valence_c1:EdProfessionaldegreeJDMD
## Valence_c1:EdSomecollegebutnodegree
## AxisInversion_c1:Handed_c1
## Valence_c1:Handed_c1
## EdBachelorsdegreeincollege4Myear:Handed_c1
## EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED:Handed_c1
## EdLessthanhighschooldegree:Handed_c1
## EdMastersdegree:Handed_c1
## EdProfessionaldegreeJDMD:Handed_c1
## EdSomecollegebutnodegree:Handed_c1
## AxisInversion_c1:Valence_c1:EdBachelorsdegreeincollege4Myear
## AxisInversion_c1:Valence_c1:EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED
## AxisInversion_c1:Valence_c1:EdLessthanhighschooldegree
## AxisInversion_c1:Valence_c1:EdMastersdegree
## AxisInversion_c1:Valence_c1:EdProfessionaldegreeJDMD
## AxisInversion_c1:Valence_c1:EdSomecollegebutnodegree
## AxisInversion_c1:Valence_c1:Handed_c1
## AxisInversion_c1:EdBachelorsdegreeincollege4Myear:Handed_c1
## AxisInversion_c1:EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED:Handed_c1
## AxisInversion_c1:EdLessthanhighschooldegree:Handed_c1
## AxisInversion_c1:EdMastersdegree:Handed_c1
```

```
## AxisInversion_c1:EdProfessionaldegreeJDMD:Handed_c1
## AxisInversion_c1:EdSomecollegebutnodegree:Handed_c1
## Valence_c1:EdBachelorsdegreeincollege4Myear:Handed_c1
## Valence_c1:EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED:Handed_c1
## Valence_c1:EdLessthanhighschooldegree:Handed_c1
## Valence_c1:EdMastersdegree:Handed_c1
## Valence_c1:EdProfessionaldegreeJDMD:Handed_c1
## Valence_c1:EdSomecollegebutnodegree:Handed_c1
## AxisInversion_c1:Valence_c1:EdBachelorsdegreeincollege4Myear:Handed_c1
## AxisInversion_c1:Valence_c1:EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED:Handed_c1
## AxisInversion_c1:Valence_c1:EdLessthanhighschooldegree:Handed_c1
## AxisInversion_c1:Valence_c1:EdMastersdegree:Handed_c1
## AxisInversion_c1:Valence_c1:EdProfessionaldegreeJDMD:Handed_c1
## AxisInversion_c1:Valence_c1:EdSomecollegebutnodegree:Handed_c1
##
## Intercept
## AxisInversion_c1
## Valence_c1
## EdBachelorsdegreeincollege4Myear
## EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED
## EdLessthanhighschooldegree
## EdMastersdegree
## EdProfessionaldegreeJDMD
## EdSomecollegebutnodegree
## Handed_c1
## AxisInversion_c1:Valence_c1
## AxisInversion_c1:EdBachelorsdegreeincollege4Myear
## AxisInversion_c1:EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED
## AxisInversion_c1:EdLessthanhighschooldegree
## AxisInversion_c1:EdMastersdegree
## AxisInversion_c1:EdProfessionaldegreeJDMD
## AxisInversion_c1:EdSomecollegebutnodegree
## Valence_c1:EdBachelorsdegreeincollege4Myear
## Valence_c1:EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED
## Valence_c1:EdLessthanhighschooldegree
## Valence_c1:EdMastersdegree
## Valence_c1:EdProfessionaldegreeJDMD
## Valence_c1:EdSomecollegebutnodegree
## AxisInversion_c1:Handed_c1
## Valence_c1:Handed_c1
## EdBachelorsdegreeincollege4Myear:Handed_c1
## EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED:Handed_c1
## EdLessthanhighschooldegree:Handed_c1
## EdMastersdegree:Handed_c1
## EdProfessionaldegreeJDMD:Handed_c1
## EdSomecollegebutnodegree:Handed_c1
## AxisInversion_c1:Valence_c1:EdBachelorsdegreeincollege4Myear
## AxisInversion_c1:Valence_c1:EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED
## AxisInversion_c1:Valence_c1:EdLessthanhighschooldegree
## AxisInversion_c1:Valence_c1:EdMastersdegree
## AxisInversion_c1:Valence_c1:EdProfessionaldegreeJDMD
## AxisInversion_c1:Valence_c1:EdSomecollegebutnodegree
## AxisInversion_c1:Valence_c1:Handed_c1
## AxisInversion_c1:EdBachelorsdegreeincollege4Myear:Handed_c1
```

```
## AxisInversion_c1:EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED:Handed_c1
## AxisInversion_c1:EdLessthanhighschooldegree:Handed_c1
## AxisInversion_c1:EdMastersdegree:Handed_c1
## AxisInversion_c1:EdProfessionaldegreeJDMD:Handed_c1
## AxisInversion_c1:EdSomecollegebutnodegree:Handed_c1
## Valence_c1:EdBachelorsdegreeincollege4Myear:Handed_c1
## Valence_c1:EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED:Handed_c1
## Valence_c1:EdLessthanhighschooldegree:Handed_c1
## Valence_c1:EdMastersdegree:Handed_c1
## Valence_c1:EdProfessionaldegreeJDMD:Handed_c1
## Valence_c1:EdSomecollegebutnodegree:Handed_c1
## AxisInversion_c1:Valence_c1:EdBachelorsdegreeincollege4Myear:Handed_c1
## AxisInversion_c1:Valence_c1:EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED:Handed_c1
## AxisInversion_c1:Valence_c1:EdLessthanhighschooldegree:Handed_c1
## AxisInversion_c1:Valence_c1:EdMastersdegree:Handed_c1
## AxisInversion_c1:Valence_c1:EdProfessionaldegreeJDMD:Handed_c1
## AxisInversion_c1:Valence_c1:EdSomecollegebutnodegree:Handed_c1
##
## Intercept
## AxisInversion_c1
## Valence_c1
## EdBachelorsdegreeincollege4Myear
## EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED
## EdLessthanhighschooldegree
## EdMastersdegree
## EdProfessionaldegreeJDMD
## EdSomecollegebutnodegree
## Handed_c1
## AxisInversion_c1:Valence_c1
## AxisInversion_c1:EdBachelorsdegreeincollege4Myear
## AxisInversion_c1:EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED
## AxisInversion_c1:EdLessthanhighschooldegree
## AxisInversion_c1:EdMastersdegree
## AxisInversion_c1:EdProfessionaldegreeJDMD
## AxisInversion_c1:EdSomecollegebutnodegree
## Valence_c1:EdBachelorsdegreeincollege4Myear
## Valence_c1:EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED
## Valence_c1:EdLessthanhighschooldegree
## Valence_c1:EdMastersdegree
## Valence_c1:EdProfessionaldegreeJDMD
## Valence_c1:EdSomecollegebutnodegree
## AxisInversion_c1:Handed_c1
## Valence_c1:Handed_c1
## EdBachelorsdegreeincollege4Myear:Handed_c1
## EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED:Handed_c1
## EdLessthanhighschooldegree:Handed_c1
## EdMastersdegree:Handed_c1
## EdProfessionaldegreeJDMD:Handed_c1
## EdSomecollegebutnodegree:Handed_c1
## AxisInversion_c1:Valence_c1:EdBachelorsdegreeincollege4Myear
## AxisInversion_c1:Valence_c1:EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED
## AxisInversion_c1:Valence_c1:EdLessthanhighschooldegree
## AxisInversion_c1:Valence_c1:EdMastersdegree
## AxisInversion_c1:Valence_c1:EdProfessionaldegreeJDMD
```

```
## AxisInversion_c1:Valence_c1:EdSomecollegebutnodegree
## AxisInversion_c1:Valence_c1:Handed_c1
## AxisInversion_c1:EdBachelorsdegreeincollege4Myear:Handed_c1
## AxisInversion_c1:EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED:Handed_c1
## AxisInversion_c1:EdLessthanhighschooldegree:Handed_c1
## AxisInversion_c1:EdMastersdegree:Handed_c1
## AxisInversion_c1:EdProfessionaldegreeJDMD:Handed_c1
## AxisInversion_c1:EdSomecollegebutnodegree:Handed_c1
## Valence_c1:EdBachelorsdegreeincollege4Myear:Handed_c1
## Valence_c1:EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED:Handed_c1
## Valence_c1:EdLessthanhighschooldegree:Handed_c1
## Valence_c1:EdMastersdegree:Handed_c1
## Valence_c1:EdProfessionaldegreeJDMD:Handed_c1
## Valence_c1:EdSomecollegebutnodegree:Handed_c1
## AxisInversion_c1:Valence_c1:EdBachelorsdegreeincollege4Myear:Handed_c1
## AxisInversion_c1:Valence_c1:EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED:Handed_c1
## AxisInversion_c1:Valence_c1:EdLessthanhighschooldegree:Handed_c1
## AxisInversion_c1:Valence_c1:EdMastersdegree:Handed_c1
## AxisInversion_c1:Valence_c1:EdProfessionaldegreeJDMD:Handed_c1
## AxisInversion_c1:Valence_c1:EdSomecollegebutnodegree:Handed_c1
##
## Samples were drawn 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).
```

None of the interactions with the Ed predictor were significant (their 95% credible intervals all contained zero).

Speed-accuracy trade-off We now test the possibility that there was a speed-accuracy trade-off in responses. First, we need to do some wrangling to ensure the reaction time data are in the same dataframe as the accuracy data:

```
# Create new dataframe called `df_acc` with relevant columns from default dataframe `df`:
df_acc <- df %>% select(Subject, Version, Response, Accuracy, Handed)

# Change values in Response column so they match values in `df_RT` (reaction time) dataframe:
df_acc$Response[df_acc$Response == "V1_r"] <- "V1_RT"
df_acc$Response[df_acc$Response == "V2_r"] <- "V2_RT"
df_acc$Response[df_acc$Response == "V3_r"] <- "V3_RT"
df_acc$Response[df_acc$Response == "V4_r"] <- "V4_RT"

# Merge `df_acc` and `df_RT` dataframes, arrange by Subject column, and select relevant columns:
df_acc <- merge(df_acc, df_RT, by = c('Subject', 'Version', 'Response', 'Handed')) %>%
  arrange(Subject) %>%
  select(Subject, AxisInversion, Orientation, Valence, Accuracy = Accuracy.x, Measurement, Handed)
```

Look at the mean reaction times for incorrect and correct responses:

```
df_acc %>% group_by(Accuracy) %>% summarise(mean(as.numeric(Measurement)))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 2 x 2
##   Accuracy `mean(as.numeric(Measurement))`
##   <fct>                              <dbl>
## 1 wrong                               3.98
```

```
## 2 right                                  4.65
```

In this experiment, we did find that incorrect responses were quicker than correct responses, indicative of a speed-accuracy trade-off.

Let's now test this with statistical models. First, log-transform reaction times:

```
# Log-transform reaction times:
df_acc <- df_acc %>% mutate(LogMeasurement = log(as.numeric(Measurement)))
```

Turn Accuracy into a factor variable and re-order levels:

```
df_acc$Accuracy <- factor(df_acc$Accuracy, levels = c("wrong", "right"))
```

Exclude rows in dataframe where reaction time Measurement is 0, which are due to software errors:

```
df_acc <- df_acc %>% filter(!Measurement == 0)
```

Run model:

```
# Run model:
xmdl <- brm(Accuracy ~ LogMeasurement +
              (1 + LogMeasurement|Subject),
            data = df_acc,
            family = bernoulli,
            init = 0,
            chains = 4,
            warmup = 2000,
            iter = 4000,
            prior = my_priors,
            control = my_controls,
            seed = 13)
```

```
## Compiling Stan program...

## Start sampling

## Warning: Bulk Effective Samples Size (ESS) is too low, indicating posterior means and medians may be
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#bulk-ess
```

```
# Summary of model:
summary(xmdl)
```

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: Accuracy ~ LogMeasurement + (1 + LogMeasurement | Subject)
##    Data: df_acc (Number of observations: 1137)
## Samples: 4 chains, each with iter = 4000; warmup = 2000; thin = 1;
##          total post-warmup samples = 8000
##
## Group-Level Effects:
## ~Subject (Number of levels: 285)
##                              Estimate Est.Error l-95% CI u-95% CI Rhat
## sd(Intercept)                    1.89      0.45     1.05     2.82 1.01
## sd(LogMeasurement)               0.72      0.40     0.05     1.55 1.03
## cor(Intercept,LogMeasurement)   -0.03      0.47    -0.79     0.90 1.00
##                              Bulk_ESS Tail_ESS
## sd(Intercept)                    1436     1758
## sd(LogMeasurement)                137      537
```

```
## cor(Intercept,LogMeasurement)     1017     1666
##
## Population-Level Effects:
##                Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept          0.67      0.33     0.05     1.35 1.00     1430     2468
## LogMeasurement     0.51      0.24     0.06     1.00 1.00      870     1604
##
## Samples were drawn 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).
```

```
# Posterior predictive checks:
# pp_check(xmdl)
```

This model marginally supports the idea that there was a speed-accuracy trade-off, with a credible interval that does not contain zero, although the model warns that the estimates may be unreliable.

Run model 1 with reaction time (logarithmically transformed) included to see if reaction time modulates the effects we observed:

```
# Turn variables into factors:
df_acc$AxisInversion <- as.factor(df_acc$AxisInversion)
df_acc$Orientation <- as.factor(df_acc$Orientation)
df_acc$Valence <- as.factor(df_acc$Valence)

# Run model:
xmdl1 <- brm(Accuracy ~ AxisInversion + Orientation + Valence + LogMeasurement +
                (1 + Valence + LogMeasurement|Subject),
                data = df_acc,
                family = bernoulli,
                init = 0,
                chains = 4,
                warmup = 2000,
                iter = 4000,
                prior = my_priors,
                control = my_controls,
                seed = 13)
```

```
## Compiling Stan program...

## Start sampling

## Warning: Bulk Effective Samples Size (ESS) is too low, indicating posterior means and medians may be
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#bulk-ess

## Warning: Tail Effective Samples Size (ESS) is too low, indicating posterior variances and tail quant
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#tail-ess
```

```
# Summary of model:
summary(xmdl1)
```

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: Accuracy ~ AxisInversion + Orientation + Valence + LogMeasurement + (1 + Valence + LogMeasu
##    Data: df_acc (Number of observations: 1137)
## Samples: 4 chains, each with iter = 4000; warmup = 2000; thin = 1;
```

```
##          total post-warmup samples = 8000
##
## Group-Level Effects:
## ~Subject (Number of levels: 285)
##                                    Estimate Est.Error l-95% CI u-95% CI Rhat
## sd(Intercept)                          3.21      0.60     2.13     4.46 1.00
## sd(Valencepositive)                    3.95      0.53     2.98     5.03 1.00
## sd(LogMeasurement)                     1.36      0.50     0.35     2.41 1.02
## cor(Intercept,Valencepositive)        -0.75      0.14    -0.95    -0.43 1.00
## cor(Intercept,LogMeasurement)          0.29      0.32    -0.35     0.88 1.00
## cor(Valencepositive,LogMeasurement)   -0.22      0.28    -0.76     0.35 1.00
##                                    Bulk_ESS Tail_ESS
## sd(Intercept)                          1646     3918
## sd(Valencepositive)                    2054     4525
## sd(LogMeasurement)                      202      362
## cor(Intercept,Valencepositive)          498      900
## cor(Intercept,LogMeasurement)           521     1215
## cor(Valencepositive,LogMeasurement)     870     1045
##
## Population-Level Effects:
##                     Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept               0.40      0.65    -0.84     1.72 1.00      752     1580
## AxisInversionnormal     2.93      0.59     1.79     4.10 1.00     1171     2420
## Orientationquant_y     -1.59      0.45    -2.51    -0.73 1.00     3049     4683
## Valencepositive         0.52      0.46    -0.31     1.48 1.00     1850     3461
## LogMeasurement          0.97      0.36     0.27     1.70 1.01      556     1607
##
## Samples were drawn 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).
```

```
# Posterior predictive checks:
# pp_check(xmdl)
```

There does seem to be some evidence for a speed-accuracy trade-off here, with the credible interval for Log Measurment not including zero, although again the model warns that the estimates may be unreliable. The results for Axis Inversion and Orientation are not greatly affected. The Valence predictor in the original model already marginally contained zero, so the result here is similar, but more of the interval spans the negative numbers now. It is therefore possible that the speed-accuracy trade-off modulates the Valence effect.

Do the same thing with model 2:

```
# Filter to graph depicting quantity on y-axis:
df_y <- df_acc %>%
  filter(Orientation == 'quant_y')

# Create copies of relevant predictors:
df_y$AxisInversion_c <- factor(df_y$AxisInversion, levels = c('normal', 'inverted'))
df_y$Valence_c <- factor(df_y$Valence, levels = c('positive', 'negative'))

# Deviation code these predictors:
contrasts(df_y$AxisInversion_c) <- contr.sum(2) / 2
contrasts(df_y$Valence_c) <- contr.sum(2) / 2

# Run model:
y_mdl <- brm(Accuracy ~ (AxisInversion_c * Valence_c) + LogMeasurement +
```

```
              (1 + Valence_c + LogMeasurement|Subject),
            data = df_y,
            family = bernoulli,
            init = 0,
            chains = 4,
            warmup = 2000,
            iter = 4000,
            prior = my_priors,
            control = my_controls,
            seed = 13)
```

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

```
## recompiling to avoid crashing R session
```

```
## Start sampling
```

```
## Warning: Bulk Effective Samples Size (ESS) is too low, indicating posterior means and medians may be
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#bulk-ess
```

```
# Summary of model:
summary(y_mdl)
```

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: Accuracy ~ (AxisInversion_c * Valence_c) + LogMeasurement + (1 + Valence_c + LogMeasurement
##    Data: df_y (Number of observations: 574)
## Samples: 4 chains, each with iter = 4000; warmup = 2000; thin = 1;
##          total post-warmup samples = 8000
##
## Group-Level Effects:
## ~Subject (Number of levels: 144)
##                                 Estimate Est.Error l-95% CI u-95% CI Rhat
## sd(Intercept)                       2.37      0.73     0.96     3.87 1.00
## sd(Valence_c1)                      4.24      0.74     2.94     5.81 1.00
## sd(LogMeasurement)                  1.68      0.71     0.30     3.09 1.01
## cor(Intercept,Valence_c1)          -0.45      0.29    -0.90     0.19 1.01
## cor(Intercept,LogMeasurement)       0.26      0.35    -0.43     0.88 1.01
## cor(Valence_c1,LogMeasurement)      0.27      0.32    -0.41     0.84 1.00
##                                 Bulk_ESS Tail_ESS
## sd(Intercept)                        840     1517
## sd(Valence_c1)                      3281     5636
## sd(LogMeasurement)                   227      651
## cor(Intercept,Valence_c1)            302      933
## cor(Intercept,LogMeasurement)        394     1776
## cor(Valence_c1,LogMeasurement)      1395     2103
##
## Population-Level Effects:
##                              Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
## Intercept                        1.34      0.62     0.21     2.65 1.00     1287
## AxisInversion_c1                 4.95      0.82     3.44     6.69 1.00     4351
## Valence_c1                       1.11      0.65    -0.17     2.39 1.00     3993
## LogMeasurement                   0.76      0.47    -0.15     1.71 1.00     1157
## AxisInversion_c1:Valence_c1      4.16      1.18     1.85     6.52 1.00     3874
##                              Tail_ESS
```

```
## Intercept                        3881
## AxisInversion_c1                 4990
## Valence_c1                       4816
## LogMeasurement                   2791
## AxisInversion_c1:Valence_c1      4955
##
## Samples were drawn 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).
```

```
# Posterior predictive checks:
# pp_check(y_mdl)
```

We do not get any warning this time about unreliable estimates, and the credible interval for LogMeasurement contains zero, indicating that this predictor did not reliably affect the results. The credible interval for Axis Inversion x Valence also spans only positive numbers, as it did in the original model.

Check this for model 3:

```
# Filter to graphs with quantity on the y-axis:
df_x <- df_acc %>% filter(Orientation == 'quant_x')

# Create copies of relevant predictors:
df_x$AxisInversion_c <- factor(df_x$AxisInversion, levels = c('normal', 'inverted'))
df_x$Valence_c <- factor(df_x$Valence, levels = c('positive', 'negative'))
df_x$Handed_c <- factor(df_x$Handed, levels = c('right', 'left'))

# Deviation code these predictors:
contrasts(df_x$AxisInversion_c) <- contr.sum(2) / 2
contrasts(df_x$Valence_c) <- contr.sum(2) / 2
contrasts(df_x$Handed_c) <- contr.sum(2) / 2

# Run model:
x_mdl <- brm(Accuracy ~ (AxisInversion_c * Valence_c) +
                        (AxisInversion_c * Valence_c * Handed_c) +
                        LogMeasurement +
                        (1 + Valence_c + LogMeasurement|Subject),
                data = df_x,
                family = bernoulli,
                init = 0,
                chains = 4,
                warmup = 2000,
                iter = 4000,
                prior = my_priors,
                control = my_controls,
                seed = 13)
```

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

```
## recompiling to avoid crashing R session
```

```
## Start sampling
```

```
# Summary of model:
summary(x_mdl)
```

```
##  Family: bernoulli
##   Links: mu = logit
```

```
## Formula: Accuracy ~ (AxisInversion_c * Valence_c) + (AxisInversion_c * Valence_c * Handed_c) + LogMea
##    Data: df_x (Number of observations: 563)
## Samples: 4 chains, each with iter = 4000; warmup = 2000; thin = 1;
##          total post-warmup samples = 8000
##
## Group-Level Effects:
## ~Subject (Number of levels: 141)
##                                Estimate Est.Error l-95% CI u-95% CI Rhat
## sd(Intercept)                      2.30      0.73     0.97     3.88 1.00
## sd(Valence_c1)                     2.75      0.61     1.63     4.02 1.00
## sd(LogMeasurement)                 1.11      0.55     0.10     2.26 1.01
## cor(Intercept,Valence_c1)          0.33      0.30    -0.29     0.85 1.00
## cor(Intercept,LogMeasurement)     -0.47      0.39    -0.92     0.52 1.00
## cor(Valence_c1,LogMeasurement)    -0.50      0.33    -0.95     0.31 1.00
##                                Bulk_ESS Tail_ESS
## sd(Intercept)                      2075     2561
## sd(Valence_c1)                     2549     4170
## sd(LogMeasurement)                  580      943
## cor(Intercept,Valence_c1)          1169     2252
## cor(Intercept,LogMeasurement)      1368     2233
## cor(Valence_c1,LogMeasurement)     1129     2367
##
## Population-Level Effects:
##                                       Estimate Est.Error l-95% CI u-95% CI Rhat
## Intercept                                 1.75      0.70     0.47     3.25 1.00
## AxisInversion_c1                          0.25      0.53    -0.77     1.30 1.00
## Valence_c1                                1.47      0.64     0.29     2.80 1.00
## Handed_c1                                -0.09      0.48    -1.05     0.88 1.00
## LogMeasurement                            0.52      0.43    -0.31     1.38 1.00
## AxisInversion_c1:Valence_c1               1.33      0.80    -0.25     2.91 1.00
## AxisInversion_c1:Handed_c1               -1.20      0.88    -2.98     0.52 1.00
## Valence_c1:Handed_c1                     -0.99      0.77    -2.49     0.53 1.00
## AxisInversion_c1:Valence_c1:Handed_c1    -0.56      1.27    -3.01     1.94 1.00
##                                       Bulk_ESS Tail_ESS
## Intercept                                 2503     3921
## AxisInversion_c1                          4326     5157
## Valence_c1                                1964     3618
## Handed_c1                                 6627     5831
## LogMeasurement                            2571     3217
## AxisInversion_c1:Valence_c1               5365     5774
## AxisInversion_c1:Handed_c1                7304     5414
## Valence_c1:Handed_c1                      7853     6981
## AxisInversion_c1:Valence_c1:Handed_c1     9275     6984
##
## Samples were drawn 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 do not get any warning this time about unreliable estimates, and the LogMeasurement predictor spans zero, indicating no strong evidence for a speed-accuracy trade-off.