Conceptual metaphor and graphical convention influence the interpretation of line graphs

Greg Woodin, Bodo Winter, and Lace Padilla

09/04/2019

Contents

Main analyses **1**
Data wrangling 1
Analyses 12
Descriptive stats 12
Inferential stats 14
Exploratory analysis 24
Reviewer-requested additional analysis 25
Educational background 25
Speed-accuracy trade-off 35

Main analyses

This is the code used for the analysis reported in Experiment 1 of 'Conceptual metaphor and graphical convention influence the interpretation of line graphs'.

Data wrangling

Load packages:

```
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.
```

```r
# 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/},
##   }
```

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

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

```r
# 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},
##   }
```

```r
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},
##   }
```

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

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

```r
# 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},
##   }
```

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

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

```r
# 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},
##   }
```

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

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

Load datasets and give them shorter names for easier coding:

```
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:

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

Create new column in each dataset denoting experiment version:

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

Join data sets together:

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

We noticed that some trials had response latencies of 0. Check how many response latencies of 0 there were per trial:

```
table(df$V1_FirstClick == 0)
```

```
##
## FALSE  TRUE
##   203    97
```

```
table(df$V2_FirstClick == 0)
```

```
##
## FALSE  TRUE
##   296     4
```

```
table(df$V3_FirstClick == 0)
```

```
##
## FALSE  TRUE
##   298     2
```

```
table(df$V4_FirstClick == 0)
```

```
##
## FALSE  TRUE
```

```
## 296 4
```

These zero response latencies seem to be mostly in the first trial, with some malfunctions in the other trials that can maybe be put down to software errors. Look to see if this error seems to disproportionately affect specific versions of the experiment:

```r
table(df$V1_FirstClick == 0, df$Version)
```

```
##
##          1  2  3  4  5  6
##   FALSE 48 49 48 50  4  4
##   TRUE   2  1  2  0 47 45
```

It seems to affect the 5th and 6th versions of the experiment mostly. See if it has anything to do with participants not answering the practice question:

```r
df2 <- filter(df, is.na(Instructions))
table(df2$V1_FirstClick == 0, df2$Version)
```

```
##
##         1 2 3 4 5 6
##   FALSE 1 1 2 6 2 2
##   TRUE  0 0 0 0 5 1
```

It doesn't seem to be anything to do with the practice question - there weren't actually many respondents that didn't answer the practice question. This is strange but there seems to have been a problem with the 5th and 6th versions of the experiment. We'll exclude these later.

Create Accuracy columns denoting whether participant got answer right to each question:

```r
df <- mutate(df, V1_a = ifelse(V1_r %in% 'Improving', 'right', 'wrong'))    # First question
df <- mutate(df, V2_a = ifelse(V2_r %in% 'Declining', 'right', 'wrong'))    # Second question
df <- mutate(df, V3_a = ifelse(V3_r %in% 'Declining', 'right', 'wrong'))    # Third question
df <- mutate(df, V4_a = ifelse(V4_r %in% 'Improving', 'right', 'wrong'))    # Fourth question
```

Exclude participants who got the trick question incorrect. Also, calculate how many participants remain after this exclusion, and how many participants were excluded:

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

```
## [1] 300
```

```r
# Original number of participants remaining 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    51
## 6       6    49
```

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

# Number of participants remaining:
```

```r
(new_len <- length(df$Subject))
```

```
## [1] 293
```

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

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

Exclude rows with response latencies more than 2 standard deviations above mean. Also, calculate how many participants remain after this exclusion, and how many participants were excluded:

```r
# Preliminaries:
cols <- c(df$V1_FirstClick, df$V2_FirstClick, df$V3_FirstClick, df$V4_FirstClick)   # Combine values of
cols <- as.numeric(cols)    # Make numeric
up_lim <- (mean(cols) + (sd(cols) * 2))   # Upper limit

# Exclude:

    # First column:
    df$V1_FirstClick <- as.numeric(df$V1_FirstClick)    # Make numeric
    df <- filter(df, V1_FirstClick < up_lim)    # Filter

    # Second column:
    df$V2_FirstClick <- as.numeric(df$V2_FirstClick)    # Make numeric
    df <- filter(df, V2_FirstClick < up_lim)    # Filter

    # Third column:
    df$V3_FirstClick <- as.numeric(df$V3_FirstClick)    # Make numeric
    df <- filter(df, V3_FirstClick < up_lim)    # Filter

    # Fourth column:
    df$V4_FirstClick <- as.numeric(df$V4_FirstClick)    # Make numeric
    df <- filter(df, V4_FirstClick < up_lim)    # Filter

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

# Number of participants excluded
new_len - newer_len
```

```
## [1] 3
```

```r
# 2 SDs above mean:
round(up_lim, 1)
```

```
## [1] 55.9
```

Find out info about participants:

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

```
## [1] 24 72
```

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

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

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

```
## [1] 11
```

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

```
##
##                  Female                     Male Non-binary/third gender
##                     130                      159                       1
```

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

```
##
##                  Female                     Male Non-binary/third gender
##                    44.8                     54.8                     0.3
```

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

```
##   Version count
## 1       1    48
## 2       2    50
## 3       3    48
## 4       4    49
## 5       5    48
## 6       6    47
```

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

```
## [1] 16.6 17.2 16.6 16.9 16.6 16.2
```

Remove extraneous columns:

```r
# Columns:
df <- select(df, Subject, V1_r, V1_RT = V1_FirstClick, V2_r, V2_RT = V2_FirstClick, V3_r, V3_RT = V3_Fi
```

Create AxisInversion column:

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

# Check to see if it's worked:
sample_n(df, 10) %>%
  select(Version, AxisInversion)
```

```
##    Version AxisInversion
## 1        6      inverted
## 2        3      inverted
## 3        4      inverted
## 4        6      inverted
## 5        1        normal
## 6        3      inverted
## 7        2        normal
## 8        1        normal
## 9        5      inverted
```

```
## 10       4      inverted
```

Create Orientation column:

```r
# Create column in df:
df <- mutate(df, Orientation = ifelse(Version %in% c('1', '3', '5'), 'quant_y', 'quant_x'))
# Check to see if it's worked:
sample_n(df, 10) %>%
  select(Version, Orientation)
```

```
##    Version Orientation
## 1        5     quant_y
## 2        5     quant_y
## 3        4     quant_x
## 4        5     quant_y
## 5        2     quant_x
## 6        1     quant_y
## 7        3     quant_y
## 8        2     quant_x
## 9        6     quant_x
## 10       5     quant_y
```

Make data long and make valence column:

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

  # 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'))

  # Check to see if it's worked:
  df %>% select(Subject, Response, Measurement, Valence) %>% head()
```

```
##   Subject Response Measurement  Valence
## 1       1     V1_r   Improving positive
## 2       1     V2_r   Declining positive
## 3       1     V3_r   Declining negative
## 4       1     V4_r   Improving negative
## 5       1    V1_RT        1.43 negative
## 6       1    V2_RT       1.753 negative
```

Make column for whether 'quant_y' graphs aligned with vertical valence metaphors:

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

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

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

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

Make Accuracy 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)

# Check to see if it's worked:
select(df, Subject, Response, Measurement, Accuracy) %>% head()
```

```
##   Subject Response Measurement Accuracy
## 1       1     V1_r   Improving    right
## 2       1     V2_r   Declining    right
## 3       1     V3_r   Declining    right
## 4       1     V4_r   Improving    right
## 5       1    V1_RT        1.43     <NA>
## 6       1    V2_RT       1.753     <NA>
```

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',
```

```r
  Version == 4 ~ 'y',
  Version == 5 ~ 'x',
  Version == 6 ~ 'x'))
```

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

```r
# 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 stats

Look at Accuracy overall:

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

```
##
## right wrong
##   736   424
```

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

```
##
## right wrong
##  63.4  36.6
```

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

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

```
##
##            right wrong
##   inverted   431   337
##   normal     305    87
```

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

```
##
##            right wrong
##   inverted  56.1  43.9
##   normal    77.8  22.2
```

People were more likely to get the answers to the questions right if the graph was normal and not inverted.

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

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

```
##
##           right wrong
##   quant_x   419   165
##   quant_y   317   259
```

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

```
##
##           right wrong
##   quant_x  71.7  28.3
##   quant_y  55.0  45.0
```

Contrary to our predictions, speakers were more likely to get the answer right if quantity was on the x-axis, and time on the y-axis.

Look at Accuracy as a function of Valence:

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

```
##
##            right wrong
##   negative   313   267
##   positive   423   157
```

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

```
##
##            right wrong
##   negative  54.0  46.0
##   positive  72.9  27.1
```

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

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

```
##
##       right wrong
##   no    125   163
##   yes   192    96
```

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

```
##
##       right wrong
##   no   43.4  56.6
##   yes  66.7  33.3
```

For graphs depicting quantity on the y-axis, people were more likely to get the answer right if the graph they looked at aligned with vertical valence metaphors.

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 94 95 30 90 48 66
##   wrong  2  5 66  8 48 28
```

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

```
##
##            1    2    3    4    5    6
##   right 97.9 95.0 31.2 91.8 50.0 70.2
##   wrong  2.1  5.0 68.8  8.2 50.0 29.8
```

```
# 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 54 62 50 60 41 46
##   wrong 42 38 46 38 55 48
```

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

```
##
##            1    2    3    4    5    6
##   right 56.2 62.0 52.1 61.2 42.7 48.9
##   wrong 43.8 38.0 47.9 38.8 57.3 51.1
```

Inferential stats

Run Model 1, which tests the effect of axis inversion, quantity mapping, and valence on response accuracy.

```
# 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: 1160)
## Samples: 4 chains, each with iter = 4000; warmup = 2000; thin = 1;
##          total post-warmup samples = 8000
##
## Group-Level Effects:
## ~Subject (Number of levels: 290)
##                                 Estimate Est.Error l-95% CI u-95% CI Rhat
## sd(Intercept)                       3.80      0.46     2.97     4.76 1.00
## sd(Valencepositive)                 3.64      0.45     2.81     4.60 1.00
## cor(Intercept,Valencepositive)     -0.84      0.06    -0.93    -0.70 1.00
##                                 Bulk_ESS Tail_ESS
## sd(Intercept)                       1764     3435
## sd(Valencepositive)                 1938     3512
## cor(Intercept,Valencepositive)      1968     3423
##
## Population-Level Effects:
##                       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept                 0.44      0.36    -0.29     1.14 1.00     2370     3241
## AxisInversionnormal       2.68      0.45     1.85     3.61 1.00     2595     3906
## Orientationquant_y       -1.81      0.36    -2.52    -1.12 1.00     2882     4251
## Valencepositive           1.50      0.35     0.85     2.23 1.00     2517     3149
##
## 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] 14.63
```

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

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

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

```
## [1] 4.5
```

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

Run leave-one-out cross-validation comparing intercept-only model with models with predictors left in:

```
# Run models to compare:

  # Run intercept-only model:
  #xmdl_null <- brm(Accuracy ~ 1 +
```

```
  #               (1 + Valence|Subject),
  #               data = df,
  #               family = bernoulli,
  #               chains = 4,
  #               warmup = 2000,
  #               init = 0,
  #               iter = 4000,
  #               sample_prior = "yes",
  #               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,
  #               prior = my_priors,
  #               control = my_controls,
  #               seed = 13)


# Calculate LOO for each model:
#loo(xmdl_null)
```

```
#loo(xmdl_axis)
#loo(xmdl_orient)
#loo(xmdl_val)

# Compare null model with AxisInversion model:
#loo_compare(xmdl_null, xmdl_axis)

# Compare null model with Orientation model:
#loo_compare(xmdl_null, xmdl_orient)

# Compare null model with Valence model:
#loo_compare(xmdl_null, xmdl_val)
```

Run Model 2, which tests the effect of vertical valence alignment on response accuracy.

```
# Filter to graphs with quantity on the 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'))
df_y$Accuracy <- factor(df_y$Accuracy, levels = c('wrong', 'right'))

# 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...

## recompiling to avoid crashing R session

## Start sampling
```

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

# 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: 576)
```

```
## 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 Bulk_ESS
## sd(Intercept)                2.51      0.38     1.85     3.31 1.00     2911
## sd(Valence_c1)               3.49      0.62     2.38     4.81 1.00     2862
## cor(Intercept,Valence_c1)   -0.18      0.24    -0.60     0.33 1.00     2529
##                          Tail_ESS
## sd(Intercept)                4343
## sd(Valence_c1)               4572
## cor(Intercept,Valence_c1)    3919
##
## Population-Level Effects:
##                            Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
## Intercept                      1.29      0.32     0.68     1.94 1.00     5229
## AxisInversion_c1               3.55      0.66     2.32     4.90 1.00     5421
## Valence_c1                     2.09      0.54     1.04     3.19 1.00     6171
## AxisInversion_c1:Valence_c1    4.78      1.05     2.78     6.90 1.00     5817
##                            Tail_ESS
## Intercept                      5420
## AxisInversion_c1               5975
## Valence_c1                     6030
## AxisInversion_c1:Valence_c1    5598
##
## 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
# Get odds:
round(exp(summary(y_mdl)$fixed[4, 1]), 2)
```

```
## [1] 118.53
```

```r
# 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%
## 3.77
```

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

```
## 97.5%
##  7.08
```

```r
# 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%
## -0.29
```

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

```
## 97.5%
##  2.08
```

```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%
## -1.39
```

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

```
## 97.5%
##  0.06
```

```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%
## -1.16
```

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

```
## 97.5%
##  0.48
```

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)

# Calculate LOO for each model:
#loo(y_mdl_null)
#loo(y_mdl)

# Compare null model with interaction model:
#loo(y_mdl_null, y_mdl)
```

Create table summary of model 1:

```
# 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)$fixed[3, 2], 2),
                    round(summary(xmdl)$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 Orie
```

Create table summary of model 2:

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

Wrangle outputs from model 1 for plotting:

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

```
## 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 <- 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"))
```

Wrangle outputs from model 2 for plotting:

```
# Convert output of model 2 into tibble:
ytrans <- ggs(y_mdl)
```

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

```
# Filter xmdl_trans_2 to interaction row:
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
```

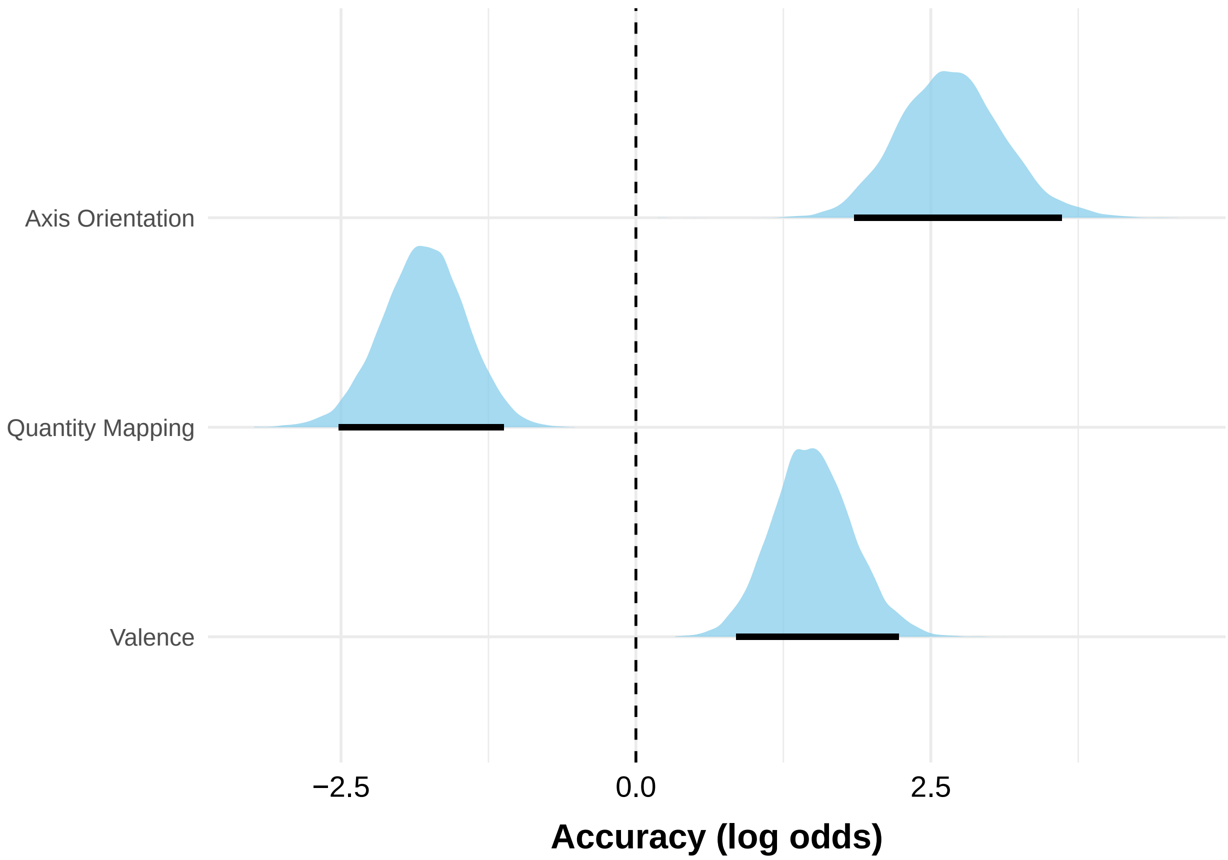
Make plot showing posterior distributions for model 1 (inspired by https://osf.io/atr57/):

```
# 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 Or

# 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/E1_model1.pdf', width = 6, height = 5)
```

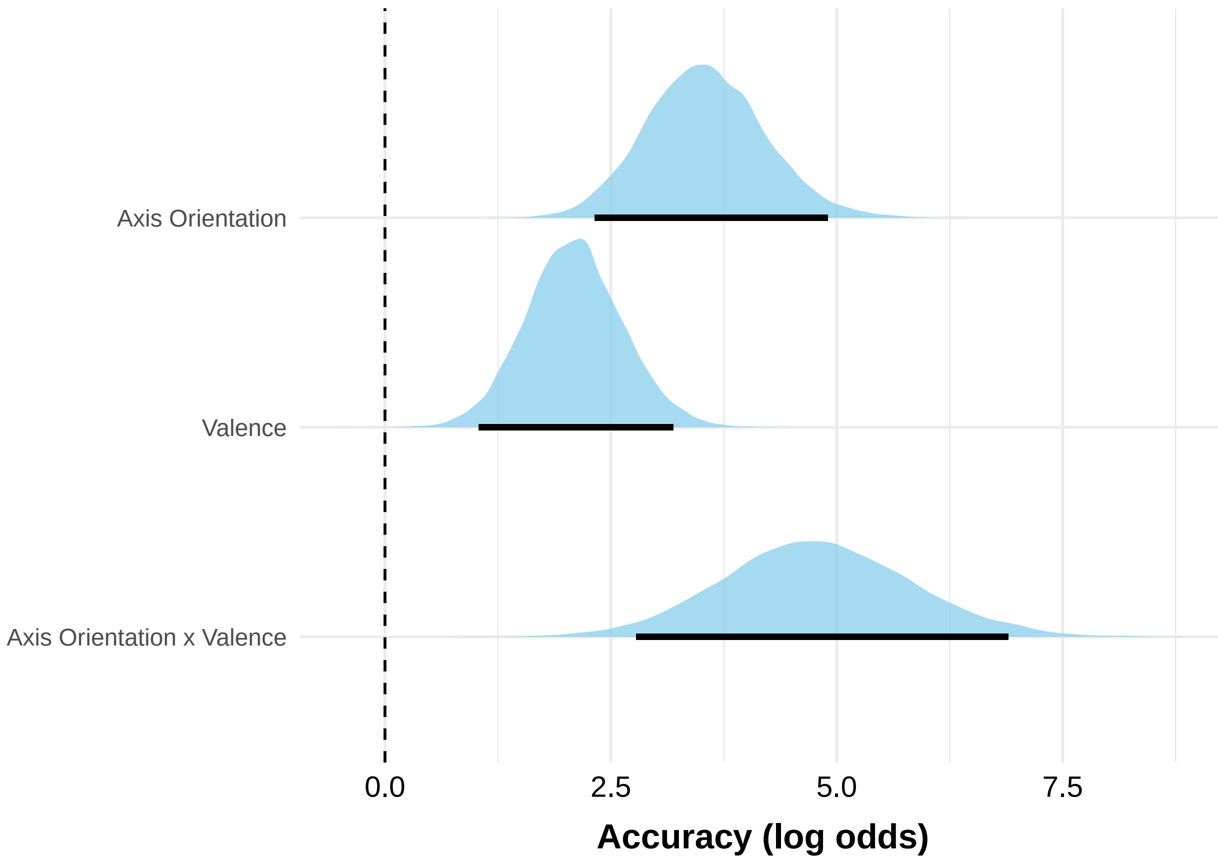
Make plot showing posterior distributions for model 2 (inspired by https://osf.io/atr57/):

```
# Combine point estimates with posterior samples:
posterior2 <- merge(summary2, xmdl_trans_2, by = 'Predictors')

# Re-order levels:
posterior2$Predictors <- factor(posterior2$Predictors, levels = c("Axis Orientation x Valence", "Valenc

# Make plot:
posterior2 %>%
  ggplot(aes(x = value, y = Predictors, fill = Predictors, xmin = Lower, xmax = Upper)) +
  stat_slab(alpha = 0.75) +
  theme_minimal() +
```

```
  geom_linerange(size = 1) +
  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))
```




```
# Save plot as PDF:
ggsave('../../table_creation/E1_model2.pdf', width = 6, height = 4)
```

Save table summaries:

```
# Remove lower and upper 95% interval values:
summary1 <- summary1 %>% select(-Lower, -Upper)          # Model 1
summary2 <- summary2 %>% select(-Lower, -Upper)          # Model 2

# Save summary of model 1 as CSV:
write_csv(summary1, '../../table_creation/E1_model1.csv')
```

```r
# Save summary of model 2 as CSV:
write_csv(summary2, '../../table_creation/E1_model2.csv')
```

Get accuracy proportions for each graph type:

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

```
##          Valence
## Accuracy negative positive
##    wrong       42        2
##    right       54       94
```

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

```
##          Valence
## Accuracy negative positive
##    wrong     43.8      2.1
##    right     56.2     97.9
```

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

```
##          Valence
## Accuracy negative positive
##    wrong      101      114
##    right       91       78
```

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

```
##          Valence
## Accuracy negative positive
##    wrong     52.6     59.4
##    right     47.4     40.6
```

Exploratory analysis

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

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

```
## , ,  = quant_x
##
##
##          inverted normal
##   wrong       122     43
##   right       262    157
##
## , ,  = quant_y
##
##
##          inverted normal
##   wrong       215     44
##   right       169    148
```

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

```
## , ,  = quant_x
##
##
##          inverted normal
##   wrong      31.8   21.5
##   right      68.2   78.5
##
## , ,  = quant_y
##
##
##          inverted normal
##   wrong      56.0   22.9
##   right      44.0   77.1
```

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   188  149
##    right   192  239
```

```
##         WhichSubvert
## Accuracy quant time
##    wrong  49.5 38.4
##    right  50.5 61.6
```

Check response latencies for participants responding to graphs mapping quantity information onto the x-axis versus the y-axis:

```
df_RT %>%
  group_by(Orientation) %>%
  summarise(mean(as.numeric(Measurement)))
```

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

```
## # A tibble: 2 x 2
##   Orientation `mean(as.numeric(Measurement))`
##   <chr>                                 <dbl>
## 1 quant_x                                5.48
## 2 quant_y                                3.75
```

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)
##                                                                        144
##                                      Bachelor's degree in college (4-year)
##                                                                        540
##                                                            Doctoral degree
##                                                                         20
## High school graduate (high school diploma or equivalent including GED)
##                                                                        120
##                                                  Less than high school degree
##                                                                         12
##                                                            Master's degree
##                                                                        140
##                                                Professional degree (JD, MD)
##                                                                         12
##                                                  Some college but no degree
##                                                                        172
```

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

```
##
##                                       Associate degree in college (2-year)
##                                                                       12.4
##                                      Bachelor's degree in college (4-year)
##                                                                       46.6
##                                                            Doctoral degree
##                                                                        1.7
## High school graduate (high school diploma or equivalent including GED)
##                                                                       10.3
##                                                  Less than high school degree
##                                                                        1.0
##                                                            Master's degree
##                                                                       12.1
##                                                Professional degree (JD, MD)
##                                                                        1.0
##                                                  Some college but no degree
##                                                                       14.8
```

Look at how accuracy varies according to education level:

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

```
##
##                                                                          wrong
##   Associate degree in college (2-year)                                      46
##   Bachelor's degree in college (4-year)                                    187
##   Doctoral degree                                                            8
##   High school graduate (high school diploma or equivalent including GED)    45
##   Less than high school degree                                               8
##   Master's degree                                                           48
##   Professional degree (JD, MD)                                               6
##   Some college but no degree                                                76
##
```

```
##                                                                              right
##   Associate degree in college (2-year)                                          98
##   Bachelor's degree in college (4-year)                                        353
##   Doctoral degree                                                               12
##   High school graduate (high school diploma or equivalent including GED)        75
##   Less than high school degree                                                   4
##   Master's degree                                                               92
##   Professional degree (JD, MD)                                                   6
##   Some college but no degree                                                    96
```

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

```
##
##                                                                              wrong
##   Associate degree in college (2-year)                                        31.9
##   Bachelor's degree in college (4-year)                                       34.6
##   Doctoral degree                                                             40.0
##   High school graduate (high school diploma or equivalent including GED)      37.5
##   Less than high school degree                                                66.7
##   Master's degree                                                             34.3
##   Professional degree (JD, MD)                                                50.0
##   Some college but no degree                                                  44.2
##
##                                                                              right
##   Associate degree in college (2-year)                                        68.1
##   Bachelor's degree in college (4-year)                                       65.4
##   Doctoral degree                                                             60.0
##   High school graduate (high school diploma or equivalent including GED)      62.5
##   Less than high school degree                                                33.3
##   Master's degree                                                             65.7
##   Professional degree (JD, MD)                                                50.0
##   Some college but no degree                                                  55.8
```

Look at how response time varied according to education level:

```
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                                                             6.45
## 2 Associate degree in college (2-year)                                        6.05
## 3 Bachelor's degree in college (4-year)                                       4.49
## 4 Less than high school degree                                                4.45
## 5 High school graduate (high school diploma or equiv~                         4.39
## 6 Some college but no degree                                                  4.37
## 7 Master's degree                                                             4.10
## 8 Professional degree (JD, MD)                                                1.92
```

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,
                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: 1160)
## Samples: 4 chains, each with iter = 4000; warmup = 2000; thin = 1;
##          total post-warmup samples = 8000
##
## Group-Level Effects:
## ~Subject (Number of levels: 290)
##                                  Estimate Est.Error l-95% CI u-95% CI Rhat
## sd(Intercept)                        4.13      0.48     3.26     5.15 1.00
## sd(Valencepositive)                  3.94      0.47     3.08     4.90 1.00
## cor(Intercept,Valencepositive)      -0.82      0.06    -0.92    -0.68 1.00
##                                  Bulk_ESS Tail_ESS
## sd(Intercept)                        2338     4117
## sd(Valencepositive)                  2598     4558
## cor(Intercept,Valencepositive)       2084     3727
##
## Population-Level Effects:
##                                                                                   Estimate
## Intercept                                                                             0.54
## AxisInversionnormal                                                                   2.79
## EdBachelorsdegreeincollege4Myear                                                     -0.02
## EdDoctoraldegree                                                                      0.27
## EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED                        -0.94
## EdLessthanhighschooldegree                                                           -1.00
## EdMastersdegree                                                                       0.18
## EdProfessionaldegreeJDMD                                                             -0.60
## EdSomecollegebutnodegree                                                             -0.21
## Orientationquant_y                                                                   -2.16
## Valencepositive                                                                       1.81
## AxisInversionnormal:EdBachelorsdegreeincollege4Myear                                  0.14
## AxisInversionnormal:EdDoctoraldegree                                                 -0.00
```

```
## AxisInversionnormal:EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED    -0.00
## AxisInversionnormal:EdLessthanhighschooldegree                                       -0.32
## AxisInversionnormal:EdMastersdegree                                                   0.06
## AxisInversionnormal:EdProfessionaldegreeJDMD                                          1.00
## AxisInversionnormal:EdSomecollegebutnodegree                                          0.36
## EdBachelorsdegreeincollege4Myear:Orientationquant_y                                   0.55
## EdDoctoraldegree:Orientationquant_y                                                   0.28
## EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED:Orientationquant_y      0.34
## EdLessthanhighschooldegree:Orientationquant_y                                        -0.73
## EdMastersdegree:Orientationquant_y                                                   -0.17
## EdProfessionaldegreeJDMD:Orientationquant_y                                          -0.61
## EdSomecollegebutnodegree:Orientationquant_y                                           0.33
## EdBachelorsdegreeincollege4Myear:Valencepositive                                     -0.34
## EdDoctoraldegree:Valencepositive                                                      0.07
## EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED:Valencepositive         1.20
## EdLessthanhighschooldegree:Valencepositive                                           -0.86
## EdMastersdegree:Valencepositive                                                       0.23
## EdProfessionaldegreeJDMD:Valencepositive                                              0.24
## EdSomecollegebutnodegree:Valencepositive                                             -1.00
##                                                                                    Est.Error
## Intercept                                                                               0.64
## AxisInversionnormal                                                                     0.77
## EdBachelorsdegreeincollege4Myear                                                        0.78
## EdDoctoraldegree                                                                        1.47
## EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED                           1.04
## EdLessthanhighschooldegree                                                              1.61
## EdMastersdegree                                                                         0.99
## EdProfessionaldegreeJDMD                                                                1.77
## EdSomecollegebutnodegree                                                                1.00
## Orientationquant_y                                                                      0.71
## Valencepositive                                                                         0.65
## AxisInversionnormal:EdBachelorsdegreeincollege4Myear                                    0.90
## AxisInversionnormal:EdDoctoraldegree                                                    2.03
## AxisInversionnormal:EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED       1.20
## AxisInversionnormal:EdLessthanhighschooldegree                                          1.76
## AxisInversionnormal:EdMastersdegree                                                     1.19
## AxisInversionnormal:EdProfessionaldegreeJDMD                                            1.69
## AxisInversionnormal:EdSomecollegebutnodegree                                            1.10
## EdBachelorsdegreeincollege4Myear:Orientationquant_y                                     0.84
## EdDoctoraldegree:Orientationquant_y                                                     1.55
## EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED:Orientationquant_y        1.14
## EdLessthanhighschooldegree:Orientationquant_y                                           1.68
## EdMastersdegree:Orientationquant_y                                                      1.09
## EdProfessionaldegreeJDMD:Orientationquant_y                                             1.73
## EdSomecollegebutnodegree:Orientationquant_y                                             1.02
## EdBachelorsdegreeincollege4Myear:Valencepositive                                        0.76
## EdDoctoraldegree:Valencepositive                                                        1.47
## EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED:Valencepositive           1.01
## EdLessthanhighschooldegree:Valencepositive                                              1.62
## EdMastersdegree:Valencepositive                                                         0.96
## EdProfessionaldegreeJDMD:Valencepositive                                                1.67
## EdSomecollegebutnodegree:Valencepositive                                                0.92
##                                                                                    l-95% CI
## Intercept                                                                              -0.70
```

```
## AxisInversionnormal                                                              1.29
## EdBachelorsdegreeincollege4Myear                                                -1.52
## EdDoctoraldegree                                                                -2.60
## EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED                   -2.93
## EdLessthanhighschooldegree                                                      -4.23
## EdMastersdegree                                                                 -1.75
## EdProfessionaldegreeJDMD                                                        -4.01
## EdSomecollegebutnodegree                                                        -2.17
## Orientationquant_y                                                              -3.56
## Valencepositive                                                                  0.56
## AxisInversionnormal:EdBachelorsdegreeincollege4Myear                            -1.62
## AxisInversionnormal:EdDoctoraldegree                                            -4.02
## AxisInversionnormal:EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED   -2.35
## AxisInversionnormal:EdLessthanhighschooldegree                                  -3.79
## AxisInversionnormal:EdMastersdegree                                             -2.24
## AxisInversionnormal:EdProfessionaldegreeJDMD                                    -2.29
## AxisInversionnormal:EdSomecollegebutnodegree                                    -1.81
## EdBachelorsdegreeincollege4Myear:Orientationquant_y                             -1.07
## EdDoctoraldegree:Orientationquant_y                                             -2.71
## EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED:Orientationquant_y    -1.91
## EdLessthanhighschooldegree:Orientationquant_y                                   -4.03
## EdMastersdegree:Orientationquant_y                                              -2.33
## EdProfessionaldegreeJDMD:Orientationquant_y                                     -4.02
## EdSomecollegebutnodegree:Orientationquant_y                                     -1.70
## EdBachelorsdegreeincollege4Myear:Valencepositive                                -1.88
## EdDoctoraldegree:Valencepositive                                                -2.80
## EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED:Valencepositive      -0.77
## EdLessthanhighschooldegree:Valencepositive                                      -4.04
## EdMastersdegree:Valencepositive                                                 -1.66
## EdProfessionaldegreeJDMD:Valencepositive                                        -2.99
## EdSomecollegebutnodegree:Valencepositive                                        -2.79
##                                                                             u-95% CI
## Intercept                                                                        1.79
## AxisInversionnormal                                                              4.32
## EdBachelorsdegreeincollege4Myear                                                 1.52
## EdDoctoraldegree                                                                 3.17
## EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED                    1.11
## EdLessthanhighschooldegree                                                       2.18
## EdMastersdegree                                                                  2.15
## EdProfessionaldegreeJDMD                                                         2.87
## EdSomecollegebutnodegree                                                         1.79
## Orientationquant_y                                                              -0.80
## Valencepositive                                                                  3.12
## AxisInversionnormal:EdBachelorsdegreeincollege4Myear                             1.90
## AxisInversionnormal:EdDoctoraldegree                                             3.99
## AxisInversionnormal:EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED    2.36
## AxisInversionnormal:EdLessthanhighschooldegree                                   3.10
## AxisInversionnormal:EdMastersdegree                                              2.42
## AxisInversionnormal:EdProfessionaldegreeJDMD                                     4.32
## AxisInversionnormal:EdSomecollegebutnodegree                                     2.60
## EdBachelorsdegreeincollege4Myear:Orientationquant_y                              2.18
## EdDoctoraldegree:Orientationquant_y                                              3.32
## EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED:Orientationquant_y     2.55
## EdLessthanhighschooldegree:Orientationquant_y                                    2.52
```

```
## EdMastersdegree:Orientationquant_y                                                         1.98
## EdProfessionaldegreeJDMD:Orientationquant_y                                                2.82
## EdSomecollegebutnodegree:Orientationquant_y                                                2.31
## EdBachelorsdegreeincollege4Myear:Valencepositive                                           1.12
## EdDoctoraldegree:Valencepositive                                                           2.93
## EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED:Valencepositive              3.15
## EdLessthanhighschooldegree:Valencepositive                                                 2.30
## EdMastersdegree:Valencepositive                                                            2.12
## EdProfessionaldegreeJDMD:Valencepositive                                                   3.55
## EdSomecollegebutnodegree:Valencepositive                                                   0.78
##                                                                                        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                                                                                  3532
## AxisInversionnormal                                                                        3934
## EdBachelorsdegreeincollege4Myear                                                           3280
## EdDoctoraldegree                                                                           6921
## EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED                              4287
## EdLessthanhighschooldegree                                                                10210
## EdMastersdegree                                                                            4006
## EdProfessionaldegreeJDMD                                                                  11182
## EdSomecollegebutnodegree                                                                   4031
## Orientationquant_y                                                                         3749
```

```
## Valencepositive                                                                   3633
## AxisInversionnormal:EdBachelorsdegreeincollege4Myear                              4221
## AxisInversionnormal:EdDoctoraldegree                                             14748
## AxisInversionnormal:EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED  5103
## AxisInversionnormal:EdLessthanhighschooldegree                                   11036
## AxisInversionnormal:EdMastersdegree                                               5514
## AxisInversionnormal:EdProfessionaldegreeJDMD                                     11289
## AxisInversionnormal:EdSomecollegebutnodegree                                      5113
## EdBachelorsdegreeincollege4Myear:Orientationquant_y                               3533
## EdDoctoraldegree:Orientationquant_y                                               8191
## EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED:Orientationquant_y  4486
## EdLessthanhighschooldegree:Orientationquant_y                                     9975
## EdMastersdegree:Orientationquant_y                                                4149
## EdProfessionaldegreeJDMD:Orientationquant_y                                      11538
## EdSomecollegebutnodegree:Orientationquant_y                                       4426
## EdBachelorsdegreeincollege4Myear:Valencepositive                                  3698
## EdDoctoraldegree:Valencepositive                                                  7934
## EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED:Valencepositive     4769
## EdLessthanhighschooldegree:Valencepositive                                       10399
## EdMastersdegree:Valencepositive                                                   4465
## EdProfessionaldegreeJDMD:Valencepositive                                         11708
## EdSomecollegebutnodegree:Valencepositive                                          3943
##                                                                               Tail_ESS
## Intercept                                                                         5491
## AxisInversionnormal                                                               5709
## EdBachelorsdegreeincollege4Myear                                                  4482
## EdDoctoraldegree                                                                  6001
## EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED                     5771
## EdLessthanhighschooldegree                                                        6103
## EdMastersdegree                                                                   4896
## EdProfessionaldegreeJDMD                                                          5635
## EdSomecollegebutnodegree                                                          5308
## Orientationquant_y                                                                5221
## Valencepositive                                                                   4761
## AxisInversionnormal:EdBachelorsdegreeincollege4Myear                              5443
## AxisInversionnormal:EdDoctoraldegree                                              5028
## AxisInversionnormal:EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED  6300
## AxisInversionnormal:EdLessthanhighschooldegree                                    6608
## AxisInversionnormal:EdMastersdegree                                               5879
## AxisInversionnormal:EdProfessionaldegreeJDMD                                      6415
## AxisInversionnormal:EdSomecollegebutnodegree                                      6120
## EdBachelorsdegreeincollege4Myear:Orientationquant_y                               5484
## EdDoctoraldegree:Orientationquant_y                                               6621
## EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED:Orientationquant_y  5409
## EdLessthanhighschooldegree:Orientationquant_y                                     5899
## EdMastersdegree:Orientationquant_y                                                5378
## EdProfessionaldegreeJDMD:Orientationquant_y                                       6518
## EdSomecollegebutnodegree:Orientationquant_y                                       4797
## EdBachelorsdegreeincollege4Myear:Valencepositive                                  4906
## EdDoctoraldegree:Valencepositive                                                  5735
## EdHighschoolgraduatehighschooldiplomaorequivalentincludingGED:Valencepositive     5979
## EdLessthanhighschooldegree:Valencepositive                                        6405
## EdMastersdegree:Valencepositive                                                   5677
## EdProfessionaldegreeJDMD:Valencepositive                                          6395
```

```
## EdSomecollegebutnodegree:Valencepositive                                   5286
##
## 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(8) / 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
```

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

# 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: 576)
## 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 Bulk_ESS
## sd(Intercept)                   2.75      0.39     2.06     3.57 1.00     3240
## sd(Valence_c1)                  3.73      0.63     2.61     5.06 1.00     3153
## cor(Intercept,Valence_c1)      -0.21      0.23    -0.62     0.27 1.00     2229
##                             Tail_ESS
## sd(Intercept)                   5273
```

```
## sd(Valence_c1)                4628
## cor(Intercept,Valence_c1)     4072
##
## Population-Level Effects:
##                                   Estimate Est.Error l-95% CI u-95% CI Rhat
## Intercept                             1.21      0.43     0.40     2.08 1.00
## AxisInversion_c1                      3.77      0.75     2.39     5.28 1.00
## Valence_c1                            2.10      0.63     0.89     3.36 1.00
## Ed_c1                                -0.37      1.38    -3.10     2.33 1.00
## Ed_c2                                 0.72      0.99    -1.23     2.65 1.00
## Ed_c3                                 0.37      1.69    -2.89     3.71 1.00
## Ed_c4                                 0.31      1.28    -2.21     2.81 1.00
## Ed_c5                                -0.62      1.76    -4.11     2.80 1.00
## Ed_c6                                 0.55      1.27    -1.93     3.02 1.00
## Ed_c7                                -0.61      1.68    -3.89     2.62 1.00
## AxisInversion_c1:Valence_c1           4.86      1.11     2.75     7.03 1.00
## AxisInversion_c1:Ed_c1                0.55      1.76    -2.93     3.94 1.00
## AxisInversion_c1:Ed_c2               -0.05      1.45    -2.92     2.83 1.00
## AxisInversion_c1:Ed_c3               -0.37      1.86    -4.01     3.38 1.00
## AxisInversion_c1:Ed_c4               -0.27      1.65    -3.53     3.04 1.00
## AxisInversion_c1:Ed_c5                0.09      1.88    -3.57     3.76 1.00
## AxisInversion_c1:Ed_c6               -0.09      1.65    -3.28     3.18 1.00
## AxisInversion_c1:Ed_c7                0.23      1.88    -3.48     4.01 1.00
## Valence_c1:Ed_c1                     -0.18      1.64    -3.38     3.04 1.00
## Valence_c1:Ed_c2                      0.66      1.30    -1.83     3.23 1.00
## Valence_c1:Ed_c3                      0.58      1.78    -2.95     4.00 1.00
## Valence_c1:Ed_c4                      0.40      1.55    -2.57     3.46 1.00
## Valence_c1:Ed_c5                     -0.80      1.83    -4.35     2.75 1.00
## Valence_c1:Ed_c6                     -0.68      1.52    -3.64     2.29 1.00
## Valence_c1:Ed_c7                      0.10      1.82    -3.46     3.65 1.00
## AxisInversion_c1:Valence_c1:Ed_c1    -0.60      1.84    -4.15     2.93 1.00
## AxisInversion_c1:Valence_c1:Ed_c2    -1.01      1.72    -4.36     2.41 1.00
## AxisInversion_c1:Valence_c1:Ed_c3    -1.17      1.89    -4.92     2.48 1.00
## AxisInversion_c1:Valence_c1:Ed_c4    -0.28      1.82    -3.79     3.33 1.00
## AxisInversion_c1:Valence_c1:Ed_c5    -0.49      1.91    -4.20     3.21 1.00
## AxisInversion_c1:Valence_c1:Ed_c6     0.00      1.85    -3.61     3.58 1.00
## AxisInversion_c1:Valence_c1:Ed_c7    -0.72      1.94    -4.54     3.09 1.00
##                                   Bulk_ESS Tail_ESS
## Intercept                             5431     5727
## AxisInversion_c1                      5544     6084
## Valence_c1                            5968     6233
## Ed_c1                                 6863     5542
## Ed_c2                                 5084     5944
## Ed_c3                                 9866     6080
## Ed_c4                                 6734     6495
## Ed_c5                                10334     6394
## Ed_c6                                 6828     5850
## Ed_c7                                 9538     6090
## AxisInversion_c1:Valence_c1           6660     6199
## AxisInversion_c1:Ed_c1                9799     5811
## AxisInversion_c1:Ed_c2                6957     6424
## AxisInversion_c1:Ed_c3               12949     6718
## AxisInversion_c1:Ed_c4                9485     5650
## AxisInversion_c1:Ed_c5               13421     6507
```

```
## AxisInversion_c1:Ed_c6                 9004     6629
## AxisInversion_c1:Ed_c7                12187     5824
## Valence_c1:Ed_c1                      11271     6110
## Valence_c1:Ed_c2                       7103     6025
## Valence_c1:Ed_c3                      13088     5969
## Valence_c1:Ed_c4                       9439     6498
## Valence_c1:Ed_c5                      11815     6635
## Valence_c1:Ed_c6                       9483     6651
## Valence_c1:Ed_c7                      12665     6635
## AxisInversion_c1:Valence_c1:Ed_c1    13421     6723
## AxisInversion_c1:Valence_c1:Ed_c2     9483     6718
## AxisInversion_c1:Valence_c1:Ed_c3    15087     5296
## AxisInversion_c1:Valence_c1:Ed_c4    13741     6566
## AxisInversion_c1:Valence_c1:Ed_c5    14307     6366
## AxisInversion_c1:Valence_c1:Ed_c6    13595     6596
## AxisInversion_c1:Valence_c1:Ed_c7    13965     5555
##
## 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)

# 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')) %>%
  arrange(Subject) %>%
  select(Subject, AxisInversion, Orientation, Valence, Accuracy = Accuracy.x, Measurement)
```

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                               4.77
## 2 right                               4.53
```

Incorrect responses were more likely to be slower, which is the opposite of what we'd expect from a speed-accuracy trade-off.

Now, look at this solely for graphs that plotted quantity on the y-axis:

```
df_x <- df_acc %>% filter(Orientation == 'quant_x')
df_x %>% 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                               6.08
## 2 right                               5.24
```

Even for these graphs, correct responses were more likely to be quicker than incorrect responses.