explanation_data_analysis

```r
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.6
v forcats   1.0.1     v stringr   1.6.0
v ggplot2   4.0.1     v tibble    3.3.0
v lubridate 1.9.4     v tidyr     1.3.1
v purrr     1.2.0
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
```

```r
library(readr)
library(ggplot2)
library(dplyr)
library(brms)
```

```
Loading required package: Rcpp
Loading 'brms' package (version 2.23.0). Useful instructions
can be found by typing help('brms'). A more detailed introduction
to the package is available through vignette('brms_overview').

Attaching package: 'brms'

The following object is masked from 'package:stats':

    ar
```

```r
library(lme4)
```

```
Loading required package: Matrix

Attaching package: 'Matrix'

The following objects are masked from 'package:tidyr':

    expand, pack, unpack


Attaching package: 'lme4'

The following object is masked from 'package:brms':

    ngrps
```

```r
library(lmerTest)
```

```

Attaching package: 'lmerTest'

The following object is masked from 'package:lme4':

    lmer

The following object is masked from 'package:stats':

    step
```

```r
library(ggpubr)
```

```r
df_full <- read_csv("/home/pulapura/Documents/MSc Speech and Language Processing/diss_data_e
```

```
Rows: 64 Columns: 103
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (103): StartDate, EndDate, Status, IPAddress, Progress, Duration (in sec...

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
#I manually coded to itemname-itemtype-condition-question format in excel
# so I renamed the headers where appropriate (qualtrics output was the full question text)
old_names <- names(df_full)
new_names <- df_full[2, ] |> as.character()
new_names[is.na(new_names) | new_names == ""] <- old_names[is.na(new_names) | new_names == "
names(df_full) <- new_names
df_clean <- df_full[-c(1:4), ]
```

```r
#Create a new row per participant per item per question
#and replace likert labels with numbers

df_pivot <- df_clean %>%
  pivot_longer(
    cols = matches("^[A-Za-z]+-(exp|unexp)-(r|nr)-[1-3]$"),
    names_to = c("item", "type", "condition", "question"),
    names_pattern = "([A-Za-z]+)-(exp|unexp)-(r|nr)-([1-3])",
    values_to = "rating"
  )

df_pivot <- df_pivot |>
  mutate(
    type = toupper(type),
    condition = toupper(condition),
    question = recode(question,"1" = "Q1","2" = "Q2", "3"="Q3")
  )

likert_map <- c(
  "Very unfavorable" = 1,
  "Unfavorable" = 2,
  "Somewhat unfavorable" = 3,
  "Neither unfavorable nor favorable" = 4,
  "Somewhat favorable" = 5,
  "Favorable" = 6,
  "Very favorable" = 7
)

df_numeric <- df_pivot %>%
  mutate(rating_num = likert_map[rating])

df_numeric$type <- factor(df_numeric$type, levels = c("UNEXP", "EXP"))
```

```
#quick summary stats

df_summary <- df_numeric %>%
  filter(question %in% c("Q1","Q2")) %>%
  group_by(type, condition, question) %>%
  summarise(
    mean = mean(rating_num, na.rm = TRUE),
    sd = sd(rating_num, na.rm = TRUE),
  )
```

```
`summarise()` has grouped output by 'type', 'condition'. You can override using
the `.groups` argument.
```

```
df_summary
```

```
# A tibble: 8 x 5
# Groups:   type, condition [4]
  type  condition question  mean    sd
  <fct> <chr>     <chr>    <dbl> <dbl>
1 UNEXP NR        Q1        4.95  1.38
2 UNEXP NR        Q2        5.08  1.42
3 UNEXP R         Q1        4.38  1.62
4 UNEXP R         Q2        5.09  1.39
5 EXP   NR        Q1        5.06  1.28
6 EXP   NR        Q2        5.18  1.22
7 EXP   R         Q1        5.05  1.60
8 EXP   R         Q2        5.60  1.26
```

```
#violin plots for Q1 (Did Alice expect Bob to react favorably?)

ggplot(df_numeric %>% filter(question == "Q1"),
       aes(x = condition, y = rating_num, fill = condition)) +
  geom_violin(alpha = 0.6) +
  geom_jitter(width = 0.1, size = 1.5, alpha = 0.15) +
  geom_boxplot(width = 0.1, fill = "white") +
  facet_grid(cols = vars(type)) +
  stat_compare_means( #will replace this with the proper test later
    method = "wilcox.test",
    paired = FALSE,
    label = "p.signif",
  ) +
```

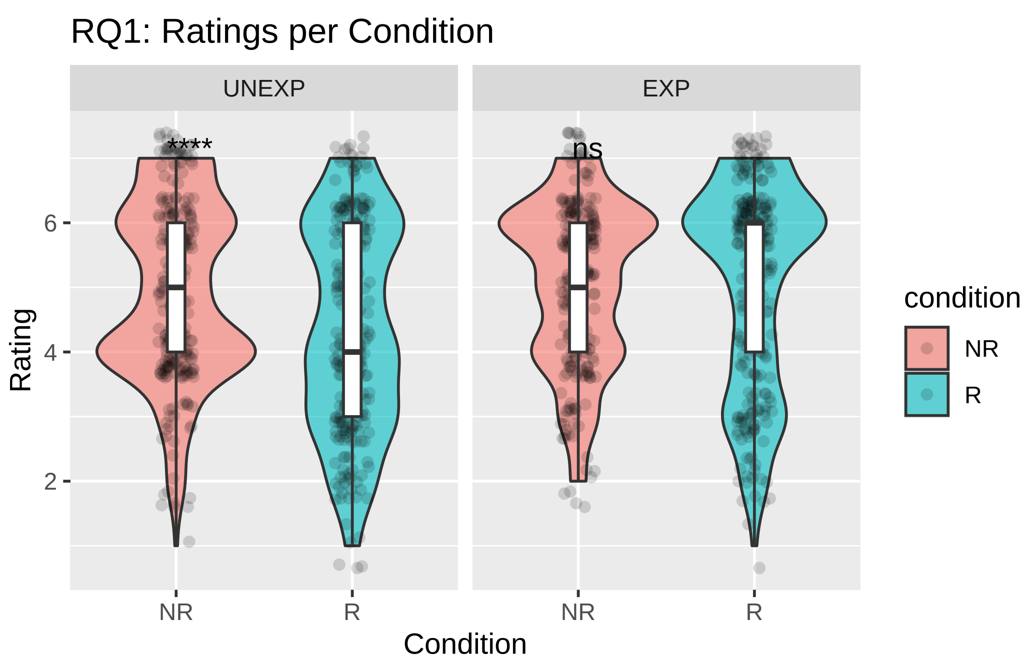
```
  labs(
    x = "Condition", y = "Rating", title = "RQ1: Ratings per Condition")
```

```
Warning: Removed 967 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 967 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 967 rows containing non-finite outside the scale range
(`stat_compare_means()`).

Warning: Removed 967 rows containing missing values or values outside the scale range
(`geom_point()`).
```




```
#Violin plots for Q2 ("How will Bob actually feel?")

ggplot(df_numeric %>% filter(question == "Q2"),
       aes(x = condition, y = rating_num, fill = condition)) +
  geom_violin(alpha = 0.6) +
```

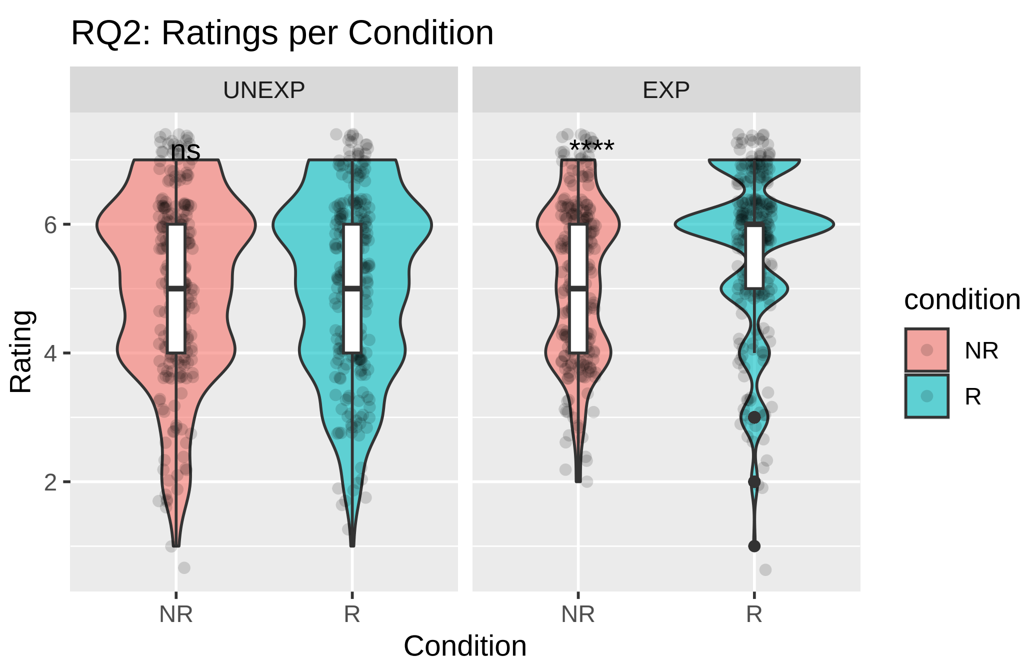
```r
  geom_jitter(width = 0.1, size = 1.5, alpha = 0.15) +
  geom_boxplot(width = 0.1, fill = "white") +
  facet_grid(cols = vars(type)) +
  stat_compare_means( #ditto Q1
    method = "wilcox.test",
    paired = FALSE,
    label = "p.signif",
  ) +
  labs(x = "Condition", y = "Rating", title = "RQ2: Ratings per Condition")
```

```
Warning: Removed 960 rows containing non-finite outside the scale range
(`stat_ydensity()`).

Warning: Removed 960 rows containing non-finite outside the scale range
(`stat_boxplot()`).

Warning: Removed 960 rows containing non-finite outside the scale range
(`stat_compare_means()`).

Warning: Removed 960 rows containing missing values or values outside the scale range
(`geom_point()`).
```



```r
df_diff <- df_numeric %>%
  filter(question %in% c("Q1", "Q2")) %>%
  select(ResponseId, item, type, condition, question, rating_num) %>%
  pivot_wider(
    names_from = question,
    values_from = rating_num
  ) %>%
  mutate(diff = Q1 - Q2) #ended up not using this yet but I kept the df name sorry
```

```r
#lme first cause im lazy

lme_Q1 <- lmer(Q1 ~ condition * type +
                   (1 | ResponseId) +
                   (1 | item),
                    data = df_diff)

summary(lme_Q1)
```

```
Linear mixed model fit by REML. t-tests use Satterthwaite's method [
lmerModLmerTest]
Formula: Q1 ~ condition * type + (1 | ResponseId) + (1 | item)
   Data: df_diff

REML criterion at convergence: 3143.3

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.2078 -0.6610  0.0688  0.6724  2.4549

Random effects:
 Groups     Name        Variance Std.Dev.
 ResponseId (Intercept) 0.3653   0.6044
 item       (Intercept) 0.5495   0.7413
 Residual               1.3499   1.1619
Number of obs: 953, groups:  ResponseId, 60; item, 16

Fixed effects:
                   Estimate Std. Error       df t value Pr(>|t|)
(Intercept)          4.9270     0.2835  17.6528  17.376 1.52e-12 ***
conditionR          -0.5248     0.1061 876.0176  -4.945 9.12e-07 ***
typeEXP              0.1456     0.3858  15.1591   0.377  0.71118
```

```
conditionR:typeEXP   0.5103     0.1507 876.2277   3.386  0.00074 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
            (Intr) cndtnR typEXP
conditionR  -0.187
typeEXP     -0.679  0.138
cndtnR:tEXP  0.132 -0.704 -0.197
```

```
lme_Q2 <- lmer(Q2 ~ condition * type +
                   (1 | ResponseId) +
                   (1 | item),
                    data = df_diff)

summary(lme_Q2)
```

```
Linear mixed model fit by REML. t-tests use Satterthwaite's method [
lmerModLmerTest]
Formula: Q2 ~ condition * type + (1 | ResponseId) + (1 | item)
   Data: df_diff

REML criterion at convergence: 2930.6

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.8830 -0.5956  0.1043  0.6490  2.8917

Random effects:
 Groups     Name        Variance Std.Dev.
 ResponseId (Intercept) 0.2176   0.4665
 item       (Intercept) 0.5373   0.7330
 Residual               1.0669   1.0329
Number of obs: 960, groups:  ResponseId, 60; item, 16

Fixed effects:
                    Estimate Std. Error        df t value Pr(>|t|)
(Intercept)          5.05756    0.27429  16.44247  18.439 2.06e-12 ***
conditionR           0.04737    0.09434 883.06257   0.502  0.61571
typeEXP              0.10980    0.37844  14.91161   0.290  0.77571
conditionR:typeEXP   0.40124    0.13342 883.06257   3.007  0.00271 **
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
            (Intr) cndtnR typEXP
conditionR  -0.172
typeEXP     -0.690  0.125
cndtnR:tEXP  0.122 -0.707 -0.176
```

```
#bayesian ordinal model cause im paranoid

fit_Q1 <- brm(Q1 ~ condition * type +
                (1|ResponseId) +
                (1|item),
                data = df_diff,
                family = cumulative())
```

```
Warning: Rows containing NAs were excluded from the model.

Compiling Stan program...

Start sampling


SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 1).
Chain 1:
Chain 1: Gradient evaluation took 0.000628 seconds
Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 6.28 seconds.
Chain 1: Adjust your expectations accordingly!
Chain 1:
Chain 1:
Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
```

```
Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
Chain 1:
Chain 1:  Elapsed Time: 33.263 seconds (Warm-up)
Chain 1:                31.442 seconds (Sampling)
Chain 1:                64.705 seconds (Total)
Chain 1:

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 2).
Chain 2:
Chain 2: Gradient evaluation took 0.000566 seconds
Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 5.66 seconds.
Chain 2: Adjust your expectations accordingly!
Chain 2:
Chain 2:
Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
Chain 2:
Chain 2:  Elapsed Time: 33.58 seconds (Warm-up)
Chain 2:                43.624 seconds (Sampling)
Chain 2:                77.204 seconds (Total)
Chain 2:

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 3).
Chain 3:
Chain 3: Gradient evaluation took 0.000627 seconds
Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 6.27 seconds.
Chain 3: Adjust your expectations accordingly!
Chain 3:
Chain 3:
Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
```

```
Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
Chain 3:
Chain 3:  Elapsed Time: 33.756 seconds (Warm-up)
Chain 3:                31.587 seconds (Sampling)
Chain 3:                65.343 seconds (Total)
Chain 3:

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 4).
Chain 4:
Chain 4: Gradient evaluation took 0.000611 seconds
Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 6.11 seconds.
Chain 4: Adjust your expectations accordingly!
Chain 4:
Chain 4:
Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
Chain 4:
Chain 4:  Elapsed Time: 33.437 seconds (Warm-up)
Chain 4:                28.017 seconds (Sampling)
Chain 4:                61.454 seconds (Total)
Chain 4:
```

```r
summary(fit_Q1)
```

```
 Family: cumulative
```

```
  Links: mu = logit
Formula: Q1 ~ condition * type + (1 | ResponseId) + (1 | item)
   Data: df_diff (Number of observations: 953)
  Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
         total post-warmup draws = 4000

Multilevel Hyperparameters:
~item (Number of levels: 16)
              Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sd(Intercept)     1.25      0.27     0.83     1.89 1.00     1151     1838

~ResponseId (Number of levels: 60)
              Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sd(Intercept)     1.13      0.14     0.89     1.43 1.00      919     1761

Regression Coefficients:
                   Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
Intercept[1]          -5.76      0.57    -6.90    -4.66 1.00     1152     1977
Intercept[2]          -3.46      0.49    -4.39    -2.49 1.00     1011     1994
Intercept[3]          -1.93      0.47    -2.84    -0.99 1.00      991     1910
Intercept[4]          -0.43      0.47    -1.34     0.49 1.00      969     1787
Intercept[5]           0.34      0.47    -0.53     1.26 1.00      991     1759
Intercept[6]           2.69      0.48     1.78     3.67 1.00     1033     1995
conditionR            -0.77      0.17    -1.09    -0.43 1.00     3489     2773
typeEXP                0.19      0.66    -1.12     1.52 1.00      883     1608
conditionR:typeEXP     0.83      0.24     0.35     1.30 1.00     3401     3085

Further Distributional Parameters:
     Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
disc     1.00      0.00     1.00     1.00   NA       NA       NA

Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
and Tail_ESS are effective sample size measures, and Rhat is the potential
scale reduction factor on split chains (at convergence, Rhat = 1).
```

```r
fit_Q2 <- brm(Q2 ~ condition * type +
                (1|ResponseId) +
                (1|item),
                data = df_diff,
                family = cumulative())
```

```
Warning: Rows containing NAs were excluded from the model.
```

```
Compiling Stan program...

Start sampling


SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 1).
Chain 1:
Chain 1: Gradient evaluation took 0.000647 seconds
Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 6.47 seconds.
Chain 1: Adjust your expectations accordingly!
Chain 1:
Chain 1:
Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
Chain 1:
Chain 1:  Elapsed Time: 46.089 seconds (Warm-up)
Chain 1:                43.231 seconds (Sampling)
Chain 1:                89.32 seconds (Total)
Chain 1:

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 2).
Chain 2:
Chain 2: Gradient evaluation took 0.000649 seconds
Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 6.49 seconds.
Chain 2: Adjust your expectations accordingly!
Chain 2:
Chain 2:
Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
```

```
Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
Chain 2:
Chain 2:  Elapsed Time: 44.36 seconds (Warm-up)
Chain 2:                45.368 seconds (Sampling)
Chain 2:                89.728 seconds (Total)
Chain 2:

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 3).
Chain 3:
Chain 3: Gradient evaluation took 0.000657 seconds
Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 6.57 seconds.
Chain 3: Adjust your expectations accordingly!
Chain 3:
Chain 3:
Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
Chain 3:
Chain 3:  Elapsed Time: 46.069 seconds (Warm-up)
Chain 3:                44.79 seconds (Sampling)
Chain 3:                90.859 seconds (Total)
Chain 3:

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 4).
Chain 4:
Chain 4: Gradient evaluation took 0.000603 seconds
Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 6.03 seconds.
Chain 4: Adjust your expectations accordingly!
```

```
Chain 4:
Chain 4:
Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
Chain 4:
Chain 4:  Elapsed Time: 41.748 seconds (Warm-up)
Chain 4:                37.875 seconds (Sampling)
Chain 4:                79.623 seconds (Total)
Chain 4:
```

```r
summary(fit_Q2)
```

```
 Family: cumulative
  Links: mu = logit
Formula: Q2 ~ condition * type + (1 | ResponseId) + (1 | item)
   Data: df_diff (Number of observations: 960)
  Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
         total post-warmup draws = 4000

Multilevel Hyperparameters:
~item (Number of levels: 16)
              Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sd(Intercept)     1.40      0.31     0.94     2.12 1.00     1325     2387

~ResponseId (Number of levels: 60)
              Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sd(Intercept)     1.03      0.13     0.80     1.32 1.00     1316     2262

Regression Coefficients:
                   Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
Intercept[1]          -6.23      0.74    -7.69    -4.80 1.00     1629     2298
Intercept[2]          -3.88      0.57    -4.94    -2.72 1.00     1212     1991
```

```
Intercept[3]          -2.40      0.54    -3.43    -1.31 1.00     1119     1675
Intercept[4]          -0.65      0.54    -1.68     0.43 1.00     1115     1899
Intercept[5]           0.50      0.54    -0.52     1.57 1.00     1111     1810
Intercept[6]           2.79      0.55     1.73     3.90 1.00     1135     1890
conditionR             0.15      0.17    -0.20     0.50 1.00     4484     3216
typeEXP                0.25      0.76    -1.25     1.79 1.00      947     1346
conditionR:typeEXP     0.75      0.24     0.28     1.22 1.00     4706     3127

Further Distributional Parameters:
     Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
disc     1.00      0.00     1.00     1.00   NA       NA       NA

Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
and Tail_ESS are effective sample size measures, and Rhat is the potential
scale reduction factor on split chains (at convergence, Rhat = 1).
```

the end :)