Data analysis

MD

1/24/23

First import, tidy, ...

```
library(tidyverse)
```

```
-- Attaching packages --------------------------------------- tidyverse 1.3.2 --
v ggplot2 3.4.0      v purrr   1.0.1
v tibble  3.1.8      v dplyr   1.0.10
v tidyr   1.2.1      v stringr 1.5.0
v readr   2.1.3      v forcats 0.5.2
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
```

```
filename1 <- "exp_april_2022_RESULTS_2022-05-02-0622.csv"
filename2 <- "exp_april_2022_RESULTS_2022-05-09-0636.csv"

data1 <- read_csv2(filename1)
```

```
i Using "',' as decimal and "'.'" as grouping mark. Use `read_delim()` for more control.
Rows: 2273 Columns: 9-- Column specification --------------------------------------------
Delimiter: ";"
chr (4): materials, is_test_trial, condition, content
dbl (5): participant, item, position, question order, rating1
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.
```

```
data2 <- read_csv2(filename2)
```

```
i Using "',' as decimal and "'.'" as grouping mark. Use `read_delim()` for more control.
Rows: 4162 Columns: 9-- Column specification --------------------------------------------
Delimiter: ";"
chr (4): materials, is_test_trial, condition, content
dbl (5): participant, item, position, question order, rating1
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.
```

```
summary(data1)
```

```
 materials          participant    is_test_trial           item       
Length:2273        Min.   : 4.00   Length:2273        Min.   : 1.000  
Class :character   1st Qu.:15.00   Class :character   1st Qu.: 3.000  
Mode  :character   Median :27.00   Mode  :character   Median : 5.000  
                   Mean   :26.65                      Mean   : 6.101  
                   3rd Qu.:39.00                      3rd Qu.: 8.000  
                   Max.   :48.00                      Max.   :18.000  
 condition             position     question order    rating1      
Length:2273        Min.   : 1.00   Min.   :1      Min.   :1.000  
Class :character   1st Qu.:16.00   1st Qu.:1      1st Qu.:1.000  
Mode  :character   Median :32.00   Median :1      Median :4.000  
                   Mean   :32.11   Mean   :1      Mean   :3.996  
                   3rd Qu.:48.00   3rd Qu.:1      3rd Qu.:7.000  
                   Max.   :64.00   Max.   :1      Max.   :7.000  
  content         
Length:2273        
Class :character   
Mode  :character   
```

```
max(data1$participant)
```

```
[1] 48
```

```
summary(data2)
```

```
 materials          participant    is_test_trial           item
Length:4162        Min.   : 1.00   Length:4162        Min.   : 1.000
Class :character   1st Qu.:22.00   Class :character   1st Qu.: 3.000
Mode  :character   Median :40.00   Mode  :character   Median : 5.000
                   Mean   :41.85                      Mean   : 6.107
                   3rd Qu.:62.00                      3rd Qu.: 8.000
                   Max.   :84.00                      Max.   :18.000
 condition             position     question order    rating1
Length:4162        Min.   : 1.00   Min.   :1      Min.   :1.000
Class :character   1st Qu.:16.00   1st Qu.:1      1st Qu.:1.000
Mode  :character   Median :32.00   Median :1      Median :3.000
                   Mean   :32.49   Mean   :1      Mean   :3.862
                   3rd Qu.:48.00   3rd Qu.:1      3rd Qu.:7.000
                   Max.   :64.00   Max.   :1      Max.   :7.000
  content
Length:4162
Class :character
Mode  :character
```

```
data2.tmp <- data2 %>%
  mutate(data2, participant = participant + max(data1$participant))

data2 <- data2.tmp

summary(data1)
```

```
 materials          participant    is_test_trial           item
Length:2273        Min.   : 4.00   Length:2273        Min.   : 1.000
Class :character   1st Qu.:15.00   Class :character   1st Qu.: 3.000
Mode  :character   Median :27.00   Mode  :character   Median : 5.000
                   Mean   :26.65                      Mean   : 6.101
                   3rd Qu.:39.00                      3rd Qu.: 8.000
                   Max.   :48.00                      Max.   :18.000
 condition             position     question order    rating1
Length:2273        Min.   : 1.00   Min.   :1      Min.   :1.000
```

```
Class :character   1st Qu.:16.00   1st Qu.:1      1st Qu.:1.000
Mode  :character   Median :32.00   Median :1      Median :4.000
                   Mean   :32.11   Mean   :1      Mean   :3.996
                   3rd Qu.:48.00   3rd Qu.:1      3rd Qu.:7.000
                   Max.   :64.00   Max.   :1      Max.   :7.000
  content
Length:2273
Class :character
Mode  :character
```

```
summary(data2)
```

```
 materials          participant     is_test_trial           item
Length:4162        Min.   : 49.00   Length:4162        Min.   : 1.000
Class :character   1st Qu.: 70.00   Class :character   1st Qu.: 3.000
Mode  :character   Median : 88.00   Mode  :character   Median : 5.000
                   Mean   : 89.85                      Mean   : 6.107
                   3rd Qu.:110.00                      3rd Qu.: 8.000
                   Max.   :132.00                      Max.   :18.000
 condition            position     question order    rating1
Length:4162        Min.   : 1.00   Min.   :1      Min.   :1.000
Class :character   1st Qu.:16.00   1st Qu.:1      1st Qu.:1.000
Mode  :character   Median :32.00   Median :1      Median :3.000
                   Mean   :32.49   Mean   :1      Mean   :3.862
                   3rd Qu.:48.00   3rd Qu.:1      3rd Qu.:7.000
                   Max.   :64.00   Max.   :1      Max.   :7.000
  content
Length:4162
Class :character
Mode  :character
```

```
# data2$demographic1 <- as.numeric(data2$demographic1)

data3 <- bind_rows(data1, data2)
```

```r
data3.tmp <- data3

data3.tmp$participant <- as.factor(data3.tmp$participant)

levels(data3.tmp$participant) <- paste(1:length(levels(data3.tmp$participant)), sep="")

data3 <- data3.tmp

summary(data3)
```

```
 materials          participant   is_test_trial           item
Length:6435        1      :  64   Length:6435        Min.   : 1.000
Class :character   3      :  64   Class :character   1st Qu.: 3.000
Mode  :character   4      :  64   Mode  :character   Median : 5.000
                   5      :  64                      Mean   : 6.105
                   6      :  64                      3rd Qu.: 8.000
                   7      :  64                      Max.   :18.000
                   (Other):6051
 condition            position     question order    rating1
Length:6435        Min.   : 1.00   Min.   :1      Min.   :1.00
Class :character   1st Qu.:16.00   1st Qu.:1      1st Qu.:1.00
Mode  :character   Median :32.00   Median :1      Median :4.00
                   Mean   :32.35   Mean   :1      Mean   :3.91
                   3rd Qu.:48.00   3rd Qu.:1      3rd Qu.:7.00
                   Max.   :64.00   Max.   :1      Max.   :7.00

  content
Length:6435
Class :character
Mode  :character
```

```r
write_csv2(data3, "merged_results.csv")

data <- data3
```

- now we add demographics
- TODO: some graphs, etc.

```r
data_demographics <- read_csv2("merged_demographics_cleaned.csv")
```

```
i Using "','" as decimal and "'.'" as grouping mark. Use `read_delim()` for more control.

Rows: 88 Columns: 14
-- Column specification --------------------------------------------------------
Delimiter: ";"
chr  (4): demographic2, demographic4, demographic5, demographic6
dbl  (4): participant, time taken sec, demographic1, demographic3
lgl  (4): id, region_Morava, age_under_27, reading_time_over_60_minutes
dttm (2): trial start utc, trial end utc

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
summary(data_demographics)
```

```
  participant       id          trial start utc
 Min.   : 1.00   Mode:logical   Min.   :2022-04-28 11:52:48
 1st Qu.:22.75   NA's:88        1st Qu.:2022-04-29 15:05:01
 Median :44.50                  Median :2022-05-02 08:47:37
 Mean   :44.50                  Mean   :2022-05-01 12:19:56
 3rd Qu.:66.25                  3rd Qu.:2022-05-02 18:32:05
 Max.   :88.00                  Max.   :2022-05-04 07:28:26

 trial end utc                 time taken sec   demographic1
 Min.   :2022-04-28 12:00:07   Min.   :  269   Min.   :19.00
 1st Qu.:2022-04-29 15:21:27   1st Qu.:  780   1st Qu.:21.00
 Median :2022-05-02 09:00:52   Median : 1005   Median :23.00
 Mean   :2022-05-01 13:00:20   Mean   : 2423   Mean   :25.59
 3rd Qu.:2022-05-02 18:50:27   3rd Qu.: 1614   3rd Qu.:25.00
 Max.   :2022-05-04 07:41:37   Max.   :72552   Max.   :71.00
                                               NA's   :1
 demographic2        demographic3    demographic4       demographic5
 Length:88          Min.   : 0.000   Length:88          Length:88
 Class :character   1st Qu.: 1.000   Class :character   Class :character
 Mode  :character   Median : 1.000   Mode  :character   Mode  :character
                    Mean   : 1.425
                    3rd Qu.: 2.000
                    Max.   :10.000
```

```
demographic6       region_Morava   age_under_27
Length:88          Mode :logical   Mode :logical
Class :character   FALSE:59        FALSE:16
Mode  :character   TRUE :29        TRUE :72
```

```
reading_time_over_60_minutes
Mode :logical
FALSE:46
TRUE :42
```

```
data_demographics$participant <- as.factor(data_demographics$participant)

data_demographics2 <- data_demographics %>%
  select(participant, region_Morava, age_under_27, reading_time_over_60_minutes, demograph

data.tmp <- data %>%
  left_join(data_demographics2, by = "participant")

summary(data.tmp)
```

```
 materials          participant   is_test_trial           item
Length:6435        1      :  64   Length:6435        Min.   : 1.000
Class :character   3      :  64   Class :character   1st Qu.: 3.000
Mode  :character   4      :  64   Mode  :character   Median : 5.000
                   5      :  64                      Mean   : 6.105
                   6      :  64                      3rd Qu.: 8.000
                   7      :  64                      Max.   :18.000
                   (Other):6051
 condition             position     question order    rating1
Length:6435        Min.   : 1.00   Min.   :1      Min.   :1.00
Class :character   1st Qu.:16.00   1st Qu.:1      1st Qu.:1.00
Mode  :character   Median :32.00   Median :1      Median :4.00
                   Mean   :32.35   Mean   :1      Mean   :3.91
```

```
                  3rd Qu.:48.00   3rd Qu.:1      3rd Qu.:7.00
                  Max.   :64.00   Max.   :1      Max.   :7.00

  content          region_Morava   age_under_27
Length:6435        Mode :logical   Mode :logical
Class :character   FALSE:3714      FALSE:972
Mode  :character   TRUE :1633      TRUE :4375
                   NA's :1088      NA's :1088



reading_time_over_60_minutes  demographic1
Mode :logical                 Min.   :19.00
FALSE:2831                    1st Qu.:21.00
TRUE :2516                    Median :22.00
NA's :1088                    Mean   :25.34
                              3rd Qu.:25.00
                              Max.   :71.00
                              NA's   :1152
```

```
data <- data.tmp
```

- here we continue with filtering

```
items <- filter(data, grepl("^Accept", materials))
practice <- filter(data, grepl("^Practice", materials))

fillers_good <- filter(data, grepl("^Filler.*Good", materials))
fillers_bad <- filter(data, grepl("^Filler.*Bad", materials))

fillers <- filter(data, grepl("^Filler.*Bad", materials) | grepl("^Filler.*Good", material

nrow(fillers_good) + nrow (fillers_bad) == nrow (fillers)
```

```
[1] TRUE
```

```
nrow(items) + nrow(practice) + nrow(fillers) == nrow(data)
```

```
[1] TRUE
```

- fillers

```
  group_by(fillers, condition) %>%
      summarise(mean(rating1), median = median(rating1))
```

```
# A tibble: 2 x 3
  condition `mean(rating1)` median
  <chr>               <dbl>  <dbl>
1 bad                  2.06      1
2 good                 5.92      7
```

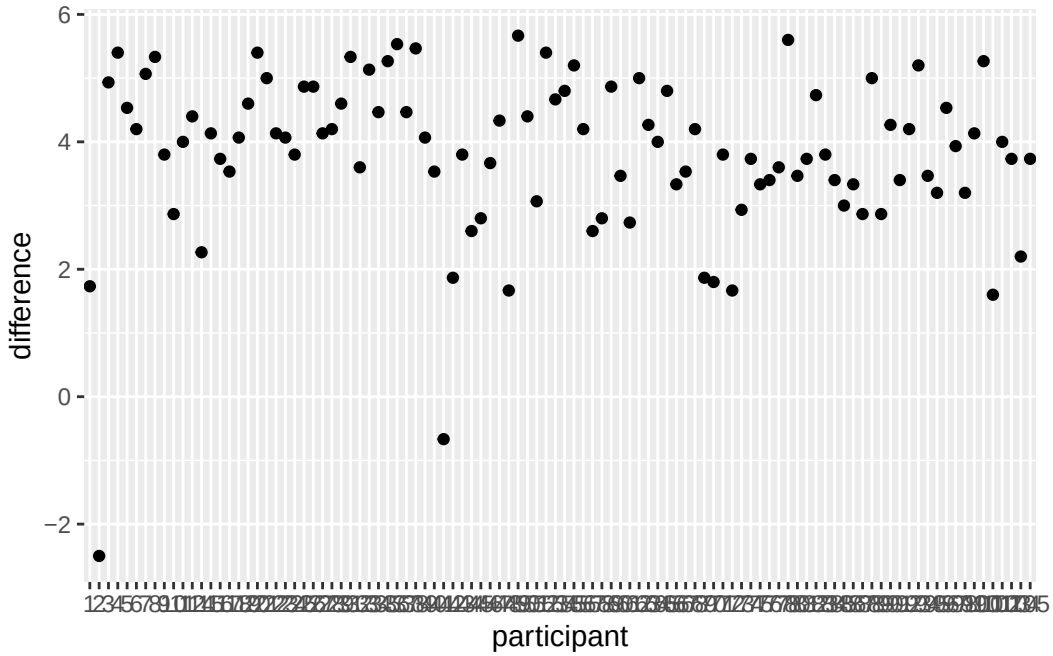
```
  by_participant_and_condition <- group_by(fillers, participant, condition)

  diff_fillers <- summarise(by_participant_and_condition, mean = mean(rating1, na.rm=TRUE),
```

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

```
  diff_fillers.tmp <- summarise(diff_fillers, difference = mean[2]-mean[1])

  ggplot(data = diff_fillers.tmp) +
    geom_point(mapping = aes(x = participant, y = difference))
```



- erase all under the difference =< 2

```
vector_for_removing <- filter(diff_fillers.tmp, difference <= 3)
vector_for_removing$participant
```

```
 [1] 1   2   10  14  41  42  44  45  48  57  58  61  69  70  72  73  85  87  89
[20] 101 104
105 Levels: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 ... 105
```

```
clean_items <- filter(items, !(participant %in% vector_for_removing$participant))
clean_fillers <- filter(fillers, !(participant %in% vector_for_removing$participant))
```

- check

```
((select(items, participant) %>%
  unique %>%
   nrow)-(select(clean_items, participant) %>%
  unique %>%
   nrow)==nrow(vector_for_removing))
```

```
[1] TRUE
```

- all clean

```
items <- clean_items
fillers <- clean_fillers

items <- group_by(items, participant, condition)
items
```

```
# A tibble: 2,406 x 13
# Groups:   participant, condition [804]
   mater~1 parti~2 is_te~3  item condi~4 posit~5 quest~6 rating1 content regio~7
   <chr>   <fct>   <chr>   <dbl> <chr>     <dbl>   <dbl>   <dbl> <chr>   <lgl>
 1 Accept~ 3       no          1 ATop         43       1       1 "*Kont~ TRUE
 2 Accept~ 3       no          2 ZBott        45       1       1 "*Kont~ TRUE
 3 Accept~ 3       no          3 ZTop         47       1       1 "*Kont~ TRUE
 4 Accept~ 3       no          4 ABott        61       1       1 "*Kont~ TRUE
 5 Accept~ 3       no          5 ATop         55       1       7 "*Kont~ TRUE
 6 Accept~ 3       no          6 ZBott        51       1       1 "*Kont~ TRUE
```

```
 7 Accept~ 3       no          7 ZTop         41       1       1 "*Kont~ TRUE
 8 Accept~ 3       no          8 ABott        53       1       7 "*Kont~ TRUE
 9 Accept~ 3       no          9 ATop         63       1       3 "*Kont~ TRUE
10 Accept~ 3       no         10 ZBott        59       1       2 "*Kont~ TRUE
# ... with 2,396 more rows, 3 more variables: age_under_27 <lgl>,
#   reading_time_over_60_minutes <lgl>, demographic1 <dbl>, and abbreviated
#   variable names 1: materials, 2: participant, 3: is_test_trial,
#   4: condition, 5: position, 6: `question order`, 7: region_Morava
```

```
  summarise(items, mean_cond_accept = mean(rating1))
```

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

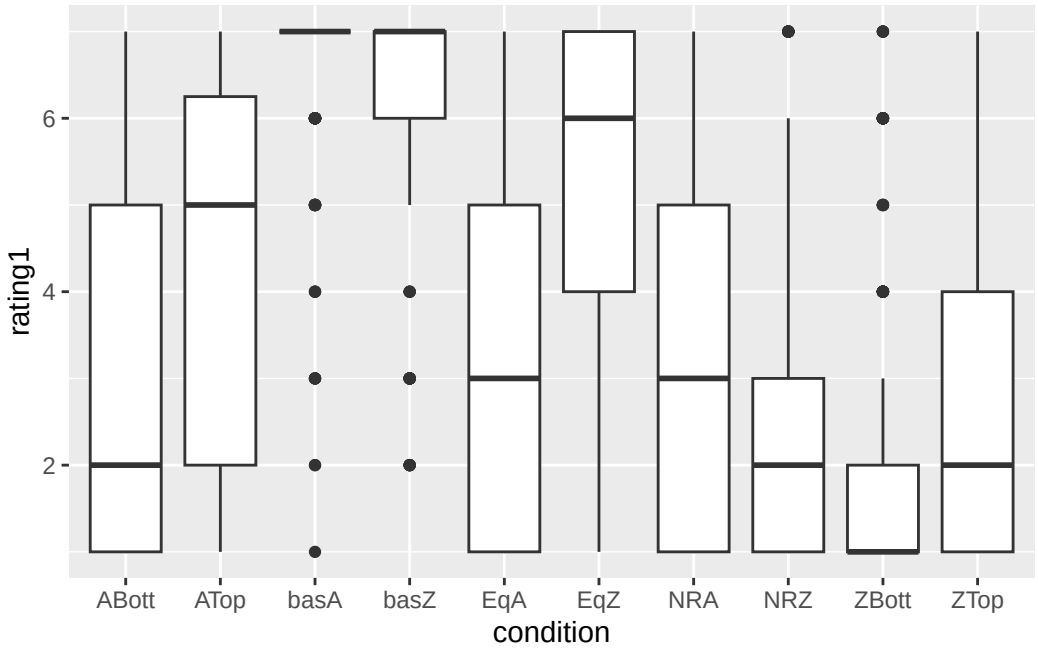
```
# A tibble: 804 x 3
# Groups:   participant [82]
   participant condition mean_cond_accept
   <fct>       <chr>               <dbl>
 1 3           ABott                4.33
 2 3           ATop                 3.67
 3 3           basA                 6.67
 4 3           basZ                 5.33
 5 3           EqA                  1
 6 3           EqZ                  7
 7 3           NRA                  3
 8 3           NRZ                  1
 9 3           ZBott                1.33
10 3           ZTop                 1.33
# ... with 794 more rows
```

```
  items <- group_by(items, condition)
  summarise(items, mean_cond_accept = mean(rating1), median_cond_accept = median(rating1), s
```

```
# A tibble: 10 x 4
   condition mean_cond_accept median_cond_accept sd_cond_accept
   <chr>               <dbl>              <dbl>          <dbl>
 1 ABott                3.12                  2          2.22
 2 ATop                 4.33                  5          2.30
 3 basA                 6.68                  7          0.931
```

```
 4 basZ      6.45      7      1.16
 5 EqA       3.39      3      2.11
 6 EqZ       5.40      6      1.89
 7 NRA       3.09      3      1.97
 8 NRZ       2.37      2      1.59
 9 ZBott     1.95      1      1.50
10 ZTop      2.7       2      1.97
```

```
ggplot(items, aes(x = condition, y = rating1)) +
geom_boxplot()
```



- better graph

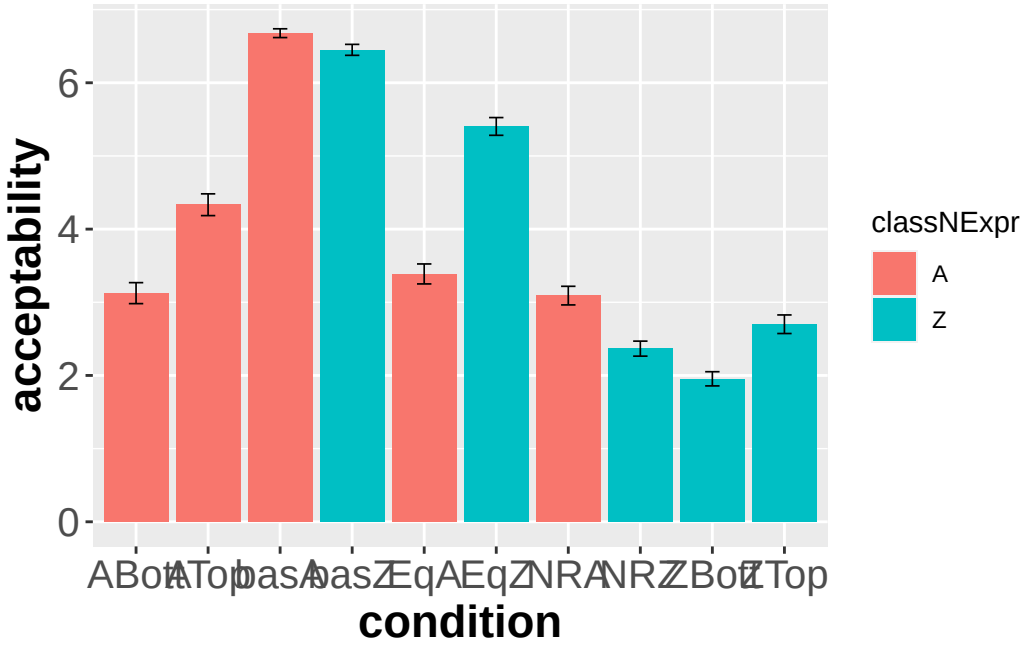
```
items$classNExpr <- "Z"
items$classNExpr[items$condition == "ABott"|items$condition == "ATop"|items$condition == "

p <- ggplot(items, aes(condition, rating1, fill = classNExpr)) +
  stat_summary(geom = "bar", fun.y = mean, position = "dodge") +
  stat_summary(geom = "errorbar", fun.data = mean_se, size=.3,
               width=.2,
               position=position_dodge(.9))
```

```
Warning: The `fun.y` argument of `stat_summary()` is deprecated as of ggplot2 3.3.0.
i Please use the `fun` argument instead.

Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
i Please use `linewidth` instead.
```

```
p  + labs(y = "acceptability") +
   theme(axis.text=element_text(size=15),
           axis.title=element_text(size=17,face="bold"))
```




```
ggsave("error_bar.png", p)
```

```
Saving 5.5 x 3.5 in image
```

And some models

- first main effects

```r
items$condition <- as.factor(items$condition)

levels(items$condition)
```

```
 [1] "ABott" "ATop"  "basA"  "basZ"  "EqA"   "EqZ"   "NRA"   "NRZ"   "ZBott"
[10] "ZTop"
```

```r
library("lmerTest")
```

```
Loading required package: lme4

Loading required package: Matrix


Attaching package: 'Matrix'

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

    expand, pack, unpack


Attaching package: 'lmerTest'

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

    lmer

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

    step
```

```r
items$condition <- relevel(items$condition, ref="basZ")

m1 <- lmer(as.numeric(rating1) ~ condition + (1|participant) + (1|item), data=items)
summary(m1)
```

```
Linear mixed model fit by REML. t-tests use Satterthwaite's method [
lmerModLmerTest]
Formula: as.numeric(rating1) ~ condition + (1 | participant) + (1 | item)
   Data: items

REML criterion at convergence: 9607.9

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.0810 -0.7028 -0.0378  0.5735  3.0915

Random effects:
 Groups      Name        Variance Std.Dev.
 participant (Intercept) 0.27127  0.5208
 item        (Intercept) 0.03288  0.1813
 Residual                3.00284  1.7329
Number of obs: 2406, groups:  participant, 82; item, 18

Fixed effects:
                Estimate Std. Error        df t value Pr(>|t|)
(Intercept)       6.4574     0.1332  350.7494  48.493  < 2e-16 ***
conditionABott   -3.3131     0.1598 2278.0524 -20.732  < 2e-16 ***
conditionATop    -2.0934     0.1598 2278.3344 -13.100  < 2e-16 ***
conditionbasA     0.2222     0.1581 2308.1992   1.405     0.16
conditionEqA     -3.0587     0.1584 2305.6950 -19.313  < 2e-16 ***
conditionEqZ     -1.0522     0.1579 2309.1970  -6.662 3.37e-11 ***
conditionNRA     -3.3679     0.1581 2300.1125 -21.309  < 2e-16 ***
conditionNRZ     -4.0815     0.1584 2305.6949 -25.772  < 2e-16 ***
conditionZBott   -4.4740     0.1598 2278.0829 -27.996  < 2e-16 ***
conditionZTop    -3.7350     0.1598 2278.2993 -23.372  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
            (Intr) cndtAB cndtAT cndtnA cndtEA cndtEZ cndNRA cndNRZ cndtZB
conditnABtt -0.589
conditinATp -0.589  0.510
conditinbsA -0.597  0.496  0.496
conditinEqA -0.595  0.496  0.495  0.501
conditinEqZ -0.598  0.497  0.497  0.504  0.501
conditinNRA -0.595  0.496  0.496  0.501  0.500  0.502
conditinNRZ -0.595  0.496  0.496  0.501  0.501  0.501  0.500
conditnZBtt -0.589  0.509  0.510  0.496  0.496  0.497  0.496  0.496
```

```
conditinZTp -0.589  0.510  0.509  0.496  0.495  0.497  0.496  0.496  0.510
```

- better base model: with interactions
- let's start with the part without probability

```
summary(items)
```

```
 materials          participant   is_test_trial           item
Length:2406        3      :  30   Length:2406        Min.   : 1.000
Class :character   4      :  30   Class :character   1st Qu.: 4.000
Mode  :character   5      :  30   Mode  :character   Median : 8.000
                   6      :  30                      Mean   : 8.307
                   7      :  30                      3rd Qu.:12.000
                   8      :  30                      Max.   :18.000
                   (Other):2226
  condition      position     question order    rating1
EqZ    :243   Min.   : 5.00   Min.   :1      Min.   :1.000
basA   :242   1st Qu.:19.00   1st Qu.:1      1st Qu.:2.000
NRA    :241   Median :34.00   Median :1      Median :4.000
basZ   :240   Mean   :34.44   Mean   :1      Mean   :3.953
ABott  :240   3rd Qu.:49.00   3rd Qu.:1      3rd Qu.:7.000
ATop   :240   Max.   :64.00   Max.   :1      Max.   :7.000
(Other):960
  content          region_Morava   age_under_27
Length:2406        Mode :logical   Mode :logical
Class :character   FALSE:1290      FALSE:390
Mode  :character   TRUE :696       TRUE :1596
                   NA's :420       NA's :420



reading_time_over_60_minutes  demographic1    classNExpr
Mode :logical                 Min.   :19.00   Length:2406
FALSE:960                     1st Qu.:21.00   Class :character
TRUE :1026                    Median :23.00   Mode  :character
NA's :420                     Mean   :25.89
                              3rd Qu.:25.00
                              Max.   :71.00
                              NA's   :450
```

```
(2406/10)*6
```

```
[1] 1443.6
```

```
items_without_probability <- items %>%
    filter((condition != "ABott" & condition != "ATop" & condition != "ZBott" & condition
    group_by(participant)



items_without_probability$condition <- as.factor(items_without_probability$condition)
items_without_probability$classNExpr <- as.factor(items_without_probability$classNExpr)


levels(items_without_probability$condition)
```

```
 [1] "basZ"  "ABott" "ATop"  "basA"  "EqA"   "EqZ"   "NRA"   "NRZ"   "ZBott"
[10] "ZTop"
```

```
items_without_probability$condition <- relevel(items_without_probability$condition, ref="b

items_without_probability$condition2 <- "NA"
items_without_probability$condition2[items_without_probability$condition == "basA" | items

items_without_probability$condition2[items_without_probability$condition == "NRA" | items_

items_without_probability$condition2[items_without_probability$condition == "EqA" | items_

items_without_probability$condition <- items_without_probability$condition2

items_without_probability <- select(items_without_probability, -condition2)


m1 <- lmer(as.numeric(rating1) ~ condition * classNExpr + (1|participant) + (1|item), data
summary(m1)
```

```
Linear mixed model fit by REML. t-tests use Satterthwaite's method [
lmerModLmerTest]
Formula: as.numeric(rating1) ~ condition * classNExpr + (1 | participant) +
    (1 | item)
   Data: items_without_probability
```

```
REML criterion at convergence: 5487.5

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.8249 -0.6080  0.0416  0.5281  3.4818

Random effects:
 Groups      Name        Variance Std.Dev.
 participant (Intercept) 0.31155  0.5582
 item        (Intercept) 0.09736  0.3120
 Residual                2.37733  1.5419
Number of obs: 1446, groups:  participant, 82; item, 18

Fixed effects:
                          Estimate Std. Error        df t value Pr(>|t|)
(Intercept)                 6.6674     0.1383  109.1613  48.226  < 2e-16 ***
conditionEq                -3.2603     0.1409 1347.5412 -23.147  < 2e-16 ***
conditionNR                -3.5707     0.1407 1346.7314 -25.379  < 2e-16 ***
classNExprZ                -0.1964     0.1409 1347.6423  -1.394  0.16355
conditionEq:classNExprZ     2.1833     0.1994 1350.9627  10.949  < 2e-16 ***
conditionNR:classNExprZ    -0.5255     0.1995 1348.0250  -2.634  0.00853 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
            (Intr) cndtnE cndtNR clsNEZ cE:NEZ
conditionEq -0.507
conditionNR -0.508  0.499
classNExprZ -0.507  0.498  0.501
cndtnEq:NEZ  0.359 -0.708 -0.356 -0.709
cndtnNR:NEZ  0.358 -0.350 -0.707 -0.708  0.500
```

- now let's do the Bayesian analysis on the data without probability

```
library(rstanarm)
```

```
Loading required package: Rcpp

This is rstanarm version 2.21.3

- See https://mc-stan.org/rstanarm/articles/priors for changes to default priors!
```

- Default priors may change, so it's safest to specify priors, even if equivalent to the defa

- For execution on a local, multicore CPU with excess RAM we recommend calling

  options(mc.cores = parallel::detectCores())

```
library(bayestestR)

# m1 <- lmer(as.numeric(rating1) ~ condition * classNExpr + (1|participant) + (1|item), da

items_without_probability$item <- as.factor(items_without_probability$item)

items_without_probability$condition <- as.factor(items_without_probability$condition)

items_without_probability$condition <- relevel(items_without_probability$condition, ref="b
items_without_probability$classNExpr <- relevel(items_without_probability$classNExpr, ref=

# m1 <- lmer(as.numeric(rating1) ~ condition * classNExpr + (1|participant) + (1|item), da

# full random effects:
# model_bayes <- stan_glmer(as.numeric(rating1) ~ condition * classNExpr + (1 + participan

# partial random effects
# model_bayes <- stan_glmer(as.numeric(rating1) ~ condition * classNExpr + (1|item) + (1|p


model_bayes <- stan_glmer(as.numeric(rating1) ~ condition * classNExpr + (1|item) + (1|par
```

```
SAMPLING FOR MODEL 'continuous' NOW (CHAIN 1).
Chain 1:
Chain 1: Gradient evaluation took 0.001147 seconds
Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 11.47 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: 20.8581 seconds (Warm-up)
Chain 1:                7.1482 seconds (Sampling)
Chain 1:                28.0063 seconds (Total)
Chain 1:

SAMPLING FOR MODEL 'continuous' NOW (CHAIN 2).
Chain 2:
Chain 2: Gradient evaluation took 0.000205 seconds
Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 2.05 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: 17.8535 seconds (Warm-up)
Chain 2:                6.67252 seconds (Sampling)
Chain 2:                24.526 seconds (Total)
Chain 2:

SAMPLING FOR MODEL 'continuous' NOW (CHAIN 3).
Chain 3:
Chain 3: Gradient evaluation took 0.00021 seconds
Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 2.1 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: 21.8002 seconds (Warm-up)
Chain 3:                6.77625 seconds (Sampling)
Chain 3:                28.5765 seconds (Total)
Chain 3:

SAMPLING FOR MODEL 'continuous' NOW (CHAIN 4).
Chain 4:
Chain 4: Gradient evaluation took 0.000263 seconds
Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 2.63 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: 17.8366 seconds (Warm-up)
Chain 4:                4.43728 seconds (Sampling)
Chain 4:                22.2739 seconds (Total)
```

```
Chain 4:
```

```
print(model_bayes, digits = 3)
```

```
stan_glmer
 family:       gaussian [identity]
 formula:      as.numeric(rating1) ~ condition * classNExpr + (1 | item) + (1 |
       participant)
 observations: 1446
------
                       Median MAD_SD
(Intercept)             6.669  0.145
conditionEq            -3.266  0.137
conditionNR            -3.573  0.140
classNExprZ            -0.197  0.142
conditionEq:classNExprZ  2.184  0.199
conditionNR:classNExprZ -0.527  0.202

Auxiliary parameter(s):
      Median MAD_SD
sigma 1.543  0.029

Error terms:
 Groups      Name        Std.Dev.
 participant (Intercept) 0.5666
 item        (Intercept) 0.3454
 Residual                1.5429
Num. levels: participant 82, item 18

------
* For help interpreting the printed output see ?print.stanreg
* For info on the priors used see ?prior_summary.stanreg
```

```
describe_posterior(model_bayes)
```

```
Summary of Posterior Distribution

Parameter               | Median |         95% CI |     pd |          ROPE | % in ROPE |  Rha
-------------------------------------------------------------------------------------------
(Intercept)             |   6.67 | [ 6.38,  6.95] |   100% | [-0.10, 0.10] |        0% | 1.00
```

```
conditionEq             |  -3.27 | [-3.53, -3.00] |   100% | [-0.10, 0.10] |        0% | 1.00
conditionNR             |  -3.57 | [-3.86, -3.30] |   100% | [-0.10, 0.10] |        0% | 1.00
classNExprZ             |  -0.20 | [-0.47,  0.08] | 92.30% | [-0.10, 0.10] |    23.08% | 1.00
conditionEq:classNExprZ |   2.18 | [ 1.81,  2.58] |   100% | [-0.10, 0.10] |        0% | 1.00
conditionNR:classNExprZ |  -0.53 | [-0.91, -0.14] | 99.67% | [-0.10, 0.10] |        0% | 1.00
```

```
bfactors <- bayesfactor(model_bayes)
```

```
Sampling priors, please wait...

Warning: Bayes factors might not be precise.
For precise Bayes factors, sampling at least 40,000 posterior samples is recommended.
```

```
print(bfactors, digits=3)
```

```
Bayes Factor (Savage-Dickey density ratio)

Parameter               |       BF
--------------------------------------
(Intercept)             | 1.14e+42
conditionEq             | 1.32e+20
conditionNR             | 4.20e+21
classNExprZ             |    0.033
conditionEq:classNExprZ | 5.95e+08
conditionNR:classNExprZ |    0.540

* Evidence Against The Null: 0
```

```
library(easystats)
```

```
# Attaching packages: easystats 0.6.0
✓ correlation 0.8.3    ✓ datawizard  0.6.5
✓ effectsize  0.8.2    ✓ insight     0.18.8
✓ modelbased  0.8.6    ✓ performance 0.10.2
✓ parameters  0.20.1   ✓ report      0.5.5
✓ see         0.7.4
```

```
exp(bfactors$log_BF)
```

```
[1] 1.137489e+42 1.315427e+20 4.196189e+21 3.298754e-02 5.952184e+08
[6] 5.404608e-01
```

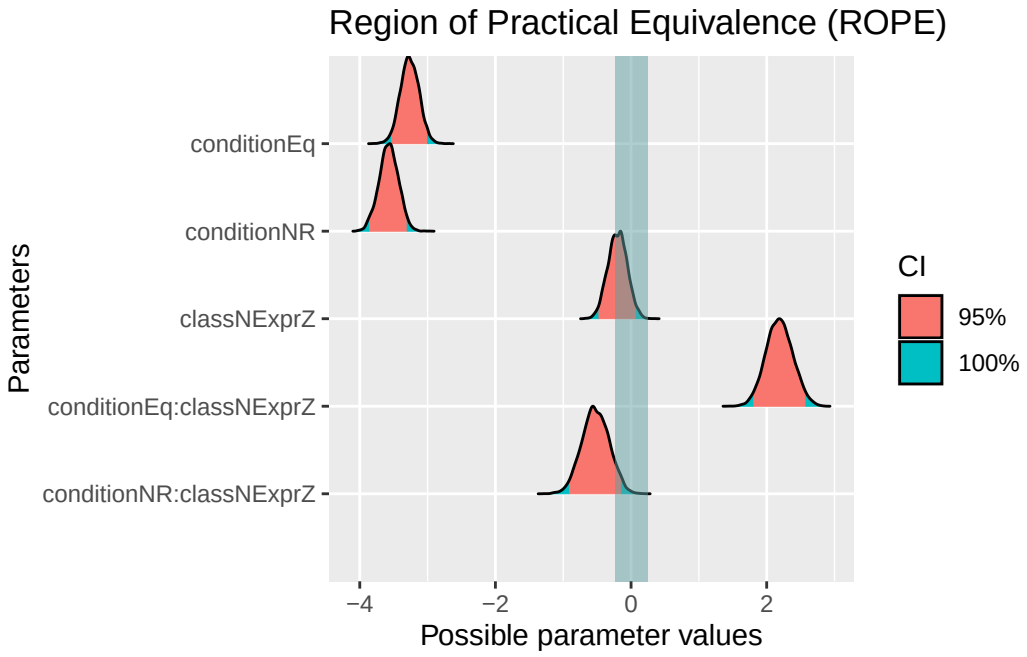
```
interpret_bf(exp(bfactors$log_BF), include_value = TRUE)
```

```
[1] "extreme evidence (BF = 1.14e+42) in favour of"
[2] "extreme evidence (BF = 1.32e+20) in favour of"
[3] "extreme evidence (BF = 4.20e+21) in favour of"
[4] "very strong evidence (BF = 1/30.31) against"
[5] "extreme evidence (BF = 5.95e+08) in favour of"
[6] "anecdotal evidence (BF = 1/1.85) against"
(Rules: jeffreys1961)
```

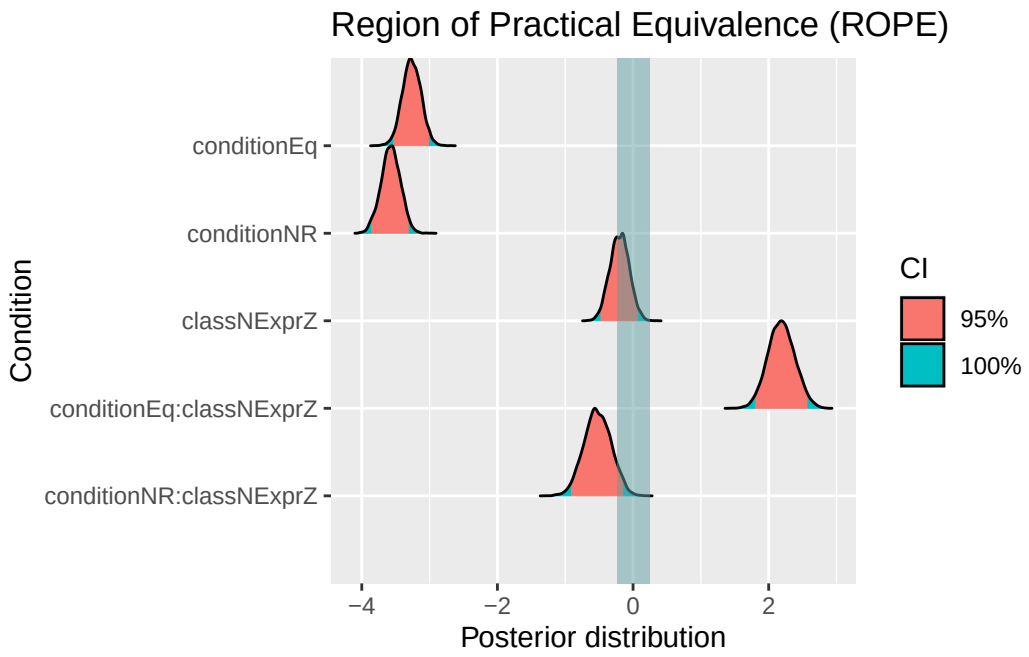
```
result <- rope(model_bayes, ci = c(0.95))
```

```
Possible multicollinearity between conditionNR:classNExprZ and conditionNR (r = 0.72), condit
```

```
plot(result)
```



```
p <- plot(result)
p <- p + ggplot2::labs(x = "Posterior distribution", y = "Condition")
p
```




```
#ggsave("rope_final_graph.png")

library(bayestestR)

equivalence_test(model_bayes)
```

```
Possible multicollinearity between conditionNR:classNExprZ and conditionNR (r = 0.72), condit

# Test for Practical Equivalence

  ROPE: [-0.24 0.24]

Parameter               |        H0 | inside ROPE |       95% HDI
-------------------------------------------------------------------
(Intercept)             |  Rejected |      0.00 % | [ 6.38  6.95]
conditionEq             |  Rejected |      0.00 % | [-3.53 -3.00]
conditionNR             |  Rejected |      0.00 % | [-3.86 -3.30]
```

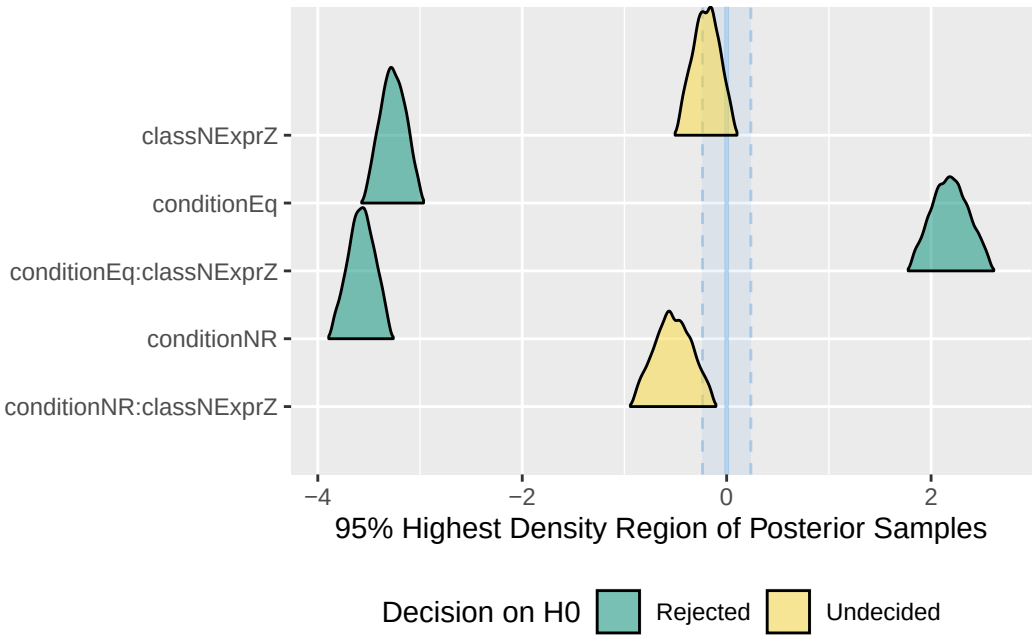
```
classNExprZ             | Undecided |     60.76 % | [-0.47  0.08]
conditionEq:classNExprZ |  Rejected |      0.00 % | [ 1.81  2.58]
conditionNR:classNExprZ | Undecided |      5.71 % | [-0.91 -0.14]
```

```
plot(equivalence_test(model_bayes))
```

```
Possible multicollinearity between conditionNR:classNExprZ and conditionNR (r = 0.72), condit
```

```
Picking joint bandwidth of 0.0247
```

```
Warning: Removed 1000 rows containing non-finite values
(`stat_density_ridges()`).
```




```
# library(emmeans)
# model_bayes.em.s <- emmeans(model_bayes, c("GrouppedCondition","Age"))
# pairs(model_bayes.em.s)
# equivalence_test(pairs(model_bayes.em.s))
```

- here we look at demographics
- first region

- and check the Bayesian version too
- nothing
- then age
- only strange interaction effect: conditionProb:age_under_27TRUE 0.84099 0.41727 1494.67832 2.015 0.044 *
- also try z-transformation and linear regression for years
- first z-transformation
- finally reading time:
- again maybe interesting observation: people with higher reading time seem to accept NR more generally (conditionNR:reading_time_over_60_minutesTRUE 0.6803 0.3317 1497.0754 2.051 0.0405 *)
- end of demographics
- in the part without probability everything works as it should
- now adding step by step probability
- first bottom (more expected)
- adding clmm package
- now top
- adding clmm package
- now let's try to remove outliers who treat *ani* as neg-word
- it seems it doesn't help
- graphs for all

Correlations

- z-transformation
- but first better descriptive stats
- NR vs. baseline
- now by subject
- graphs
- graph for all conditions

- subjects consistently rating NR high but nothing like that hapens with baseline
- no subject ranks baseline bad (consistently or not)
- great variation between speakers but only in some environments
- but it's not general acceptance of *ani*
- adding equatives (graph only for *ani*)
- because people who accept *ani* with equative are different people than those who accept *ani* with NRs
- only 3 subjects appear in both groups
- now correlations
- first z-transformation
- correlations
- first equatives with NR
- next ATop and NR
- checking against baseline

old analysis follows

- not run
- next pairwise comparison

“‘{r}{r eval=FALSE, include=FALSE} library(emmeans)

emmeans(m1, list(pairwise ~ condition), adjust = “tukey”)

“‘

- probability part