Exp 2 of 4

August 13, 2016

Contents

1 Get the data

```
# if the file data_processed.Rda already exists then load it, else do data wrangling
if (file.exists("data_processed.Rda")) {
    load("data_processed.Rda")
} else {
    # declare local variables
    number_of_valid_subjects <- 30  # = 30
    number_of_rows <- 7680  # 7680
    number_of_trials_per_subject <- number_of_rows/number_of_valid_subjects  # = 256
    # call functions
    dat <- gatherData(number_of_valid_subjects)  # = 30
    dat <- classifyResponses(dat)  # classify the response as expected, near, or far
    dat <- processData(dat)  # remove impossible trials and re-do previous rt measures
    dd <- postProcessData(dat)
}  # end else do data wrangling
```

```
names(dd)
```

```
 [1] "id"                "Subject"           "Trial"             "Condition"
 [5] "Order"             "Quantity"          "Vagueness"         "Number"
 [9] "Item"              "discriminability"  "c_Trl"             "s_Trl"
[13] "c_Itm"             "c_Vag"             "c_Num"             "f_Cnd"
[17] "c_Ord"             "c_Qty"             "RT"                "RT_log"
[21] "RT_raw"            "RTprev"            "RTprev_log"        "RTprev_raw"
[25] "Exp_Num"           "Bline_Num"         "Extr_Num"          "Exp_side"
[29] "Bline_side"        "Extr_side"         "response_num"      "response_side"
[33] "response_category" "Left"              "Mid"               "Right"
[37] "Instruction"       "nchar_instr"
```

```
head(dd)
```

```
        id Subject Trial Condition Order Quantity Vagueness  Number     Item discriminability
1 s01:t001     s01     1         2  RtoL    Small     Crisp Numeric 06:15:24        0.4875000
2 s01:t002     s01     2         1  LtoR    Large     Vague Numeric 16:25:34        0.3123529
3 s01:t003     s01     3         4  LtoR    Small     Crisp  Verbal 26:35:44        0.2308442
4 s01:t004     s01     4         1  RtoL    Large     Vague Numeric 36:45:54        0.1833333
5 s01:t005     s01     5         1  RtoL    Small     Vague Numeric 06:15:24        0.4875000
6 s01:t006     s01     6         3  LtoR    Large     Vague  Verbal 16:25:34        0.3123529
   c_Trl     s_Trl c_Itm c_Vag c_Num f_Cnd c_Ord c_Qty   RT   RT_log RT_raw RTprev RTprev_log
1 -127.5 -1.725186 -0.75  -0.5  -0.5 Cr:Nm   0.5  -0.5 1517 7.324490   1517   1517   7.324490
2 -126.5 -1.711655 -0.25   0.5  -0.5 Vg:Nm  -0.5   0.5 1920 7.560080   1920   1517   7.324490
3 -125.5 -1.698124  0.25  -0.5   0.5 Cr:Vb  -0.5  -0.5 2346 7.760467   2346   1920   7.560080
4 -124.5 -1.684593  0.75   0.5  -0.5 Vg:Nm   0.5   0.5 1773 7.480428   1773   2346   7.760467
5 -123.5 -1.671062 -0.75   0.5  -0.5 Vg:Nm   0.5  -0.5 2556 7.846199   2556   1773   7.480428
6 -122.5 -1.657531 -0.25   0.5   0.5 Vg:Vb  -0.5   0.5 2043 7.622175   2043   2556   7.846199
  RTprev_raw Exp_Num Bline_Num Extr_Num Exp_side Bline_side Extr_side response_num response_side
1       1517       6        15       24    right        mid      left            6          right
2       1517      34        25       16    right        mid      left           25            mid
3       1920      26        35       44     left        mid     right           26           left
4       2346      54        45       36     left        mid     right           45            mid
5       1773       6        15       24    right        mid      left            6          right
6       2556      34        25       16    right        mid      left           34          right
  response_category Left Mid Right                          Instruction nchar_instr
1          expected   24  15     6          Choose the square with 6 dots          29
2        borderline   16  25    34     Choose a square with about 30 dots          34
3          expected   26  35    44 Choose the square with the fewest dots          38
4        borderline   54  45    36     Choose a square with about 50 dots          34
5          expected   24  15     6     Choose a square with about 10 dots          34
6          expected   16  25    34         Choose a square with many dots          30
```

```
summary(dd)
```

```
       id          Subject         Trial          Condition    Order       Quantity    Vagueness
 s01:t001:   1   s01    : 256   Min.   :  1.0   Min.   :1.0   LtoR:3838   Small:3840   Crisp:3840
 s01:t002:   1   s02    : 256   1st Qu.: 65.0   1st Qu.:2.0   RtoL:3839   Large:3837   Vague:3837
 s01:t003:   1   s03    : 256   Median :129.0   Median :3.0
 s01:t004:   1   s04    : 256   Mean   :128.5   Mean   :2.5
 s01:t005:   1   s05    : 256   3rd Qu.:193.0   3rd Qu.:4.0
 s01:t006:   1   s06    : 256   Max.   :256.0   Max.   :4.0
 (Other) :7671   (Other):6141
    Number           Item      discriminability     c_Trl                s_Trl
 Numeric:3838   06:15:24:1919   Min.   :0.1833   Min.   :-127.50000   Min.   :-1.7251858
 Verbal :3839   16:25:34:1919   1st Qu.:0.2308   1st Qu.: -63.50000   1st Qu.:-0.8592102
                26:35:44:1920   Median :0.2308   Median :   0.50000   Median : 0.0067654
                36:45:54:1919   Mean   :0.3035   Mean   :   0.02377   Mean   : 0.0003217
                                3rd Qu.:0.3124   3rd Qu.:  64.50000   3rd Qu.: 0.8727411
                                Max.   :0.4875   Max.   : 127.50000   Max.   : 1.7251858

    c_Itm               c_Vag                c_Num             f_Cnd          c_Ord
 Min.   :-7.50e-01   Min.   :-0.5000000   Min.   :-5.00e-01   Vg:Nm:1918   Min.   :-5.00e-01
 1st Qu.:-2.50e-01   1st Qu.:-0.5000000   1st Qu.:-5.00e-01   Cr:Nm:1920   1st Qu.:-5.00e-01
 Median : 2.50e-01   Median :-0.5000000   Median : 5.00e-01   Vg:Vb:1919   Median : 5.00e-01
 Mean   : 3.26e-05   Mean   :-0.0001954   Mean   : 6.51e-05   Cr:Vb:1920   Mean   : 6.51e-05
 3rd Qu.: 2.50e-01   3rd Qu.: 0.5000000   3rd Qu.: 5.00e-01                3rd Qu.: 5.00e-01
 Max.   : 7.50e-01   Max.   : 0.5000000   Max.   : 5.00e-01                Max.   : 5.00e-01

    c_Qty                RT            RT_log           RT_raw          RTprev
 Min.   :-0.5000000   Min.   :  445   Min.   : 6.098   Min.   :  445   Min.   :  445
 1st Qu.:-0.5000000   1st Qu.: 1240   1st Qu.: 7.123   1st Qu.: 1240   1st Qu.: 1240
 Median :-0.5000000   Median : 1727   Median : 7.454   Median : 1727   Median : 1727
 Mean   :-0.0001954   Mean   : 2840   Mean   : 7.595   Mean   : 2840   Mean   : 2835
 3rd Qu.: 0.5000000   3rd Qu.: 2699   3rd Qu.: 7.901   3rd Qu.: 2699   3rd Qu.: 2697
 Max.   : 0.5000000   Max.   :42685   Max.   :10.662   Max.   :42685   Max.   :42685

  RTprev_log       RTprev_raw       Exp_Num     Bline_Num     Extr_Num     Exp_side
 Min.   : 6.098   Min.   :  445   Min.   : 6   Min.   :15   Min.   : 6   Length:7677
 1st Qu.: 7.123   1st Qu.: 1240   1st Qu.:16   1st Qu.:25   1st Qu.:24   Class :character
 Median : 7.454   Median : 1727   Median :26   Median :35   Median :34   Mode  :character
 Mean   : 7.594   Mean   : 2835   Mean   :30   Mean   :30   Mean   :30
 3rd Qu.: 7.900   3rd Qu.: 2697   3rd Qu.:36   3rd Qu.:35   3rd Qu.:44
 Max.   :10.662   Max.   :42685   Max.   :54   Max.   :45   Max.   :54

 Bline_side         Extr_side          response_num   response_side  response_category
 Length:7677        Length:7677        Min.   : 6.00   left :3215    borderline:1274
 Class :character   Class :character   1st Qu.:24.00   mid  :1274    expected  :6108
 Mode  :character   Mode  :character   Median :34.00   right:3188    extreme   : 295
                                       Mean   :30.87
                                       3rd Qu.:44.00
                                       Max.   :54.00

     Left         Mid         Right                                      Instruction
 Min.   : 6   Min.   :15   Min.   : 6   Choose a square with few dots         : 960
 1st Qu.:24   1st Qu.:25   1st Qu.:24   Choose the square with the fewest dots: 960
 Median :34   Median :35   Median :26   Choose the square with the most dots  : 960
 Mean   :30   Mean   :30   Mean   :30   Choose a square with many dots        : 959
 3rd Qu.:44   3rd Qu.:35   3rd Qu.:36   Choose a square with about 30 dots    : 480
 Max.   :54   Max.   :45   Max.   :54   Choose a square with about 40 dots    : 480
                                        (Other)                              :2878
 nchar_instr
 Min.   :29.00
 1st Qu.:30.00
 Median :30.00
 Mean   :32.59
 3rd Qu.:36.00
 Max.   :38.00
```

```
str(dd)

'data.frame': 7677 obs. of  38 variables:
 $ id                : Factor w/ 7680 levels "s01:t001","s01:t002",..: 1 2 3 4 5 6 7 8 9 10 ...
 $ Subject           : Factor w/ 30 levels "s01","s02","s03",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ Trial             : int  1 2 3 4 5 6 7 8 9 10 ...
 $ Condition         : int  2 1 4 1 1 3 3 4 4 2 ...
 $ Order             : Factor w/ 2 levels "LtoR","RtoL": 2 1 1 2 2 1 2 2 1 2 ...
 $ Quantity          : Factor w/ 2 levels "Small","Large": 1 2 1 2 1 2 1 1 2 2 ...
 $ Vagueness         : Factor w/ 2 levels "Crisp","Vague": 1 2 1 2 2 2 2 1 1 1 ...
 $ Number            : Factor w/ 2 levels "Numeric","Verbal": 1 1 2 1 1 2 2 2 2 1 ...
 $ Item              : Factor w/ 4 levels "06:15:24","16:25:34",..: 1 2 3 4 1 2 3 4 1 2 ...
 $ discriminability : num  0.487 0.312 0.231 0.183 0.487 ...
 $ c_Trl             : num  -128 -126 -126 -124 -124 ...
 $ s_Trl             : num  -1.73 -1.71 -1.7 -1.68 -1.67 ...
 $ c_Itm             : num  -0.75 -0.25 0.25 0.75 -0.75 -0.25 0.25 0.75 -0.75 -0.25 ...
 $ c_Vag             : num  -0.5 0.5 -0.5 0.5 0.5 0.5 0.5 -0.5 -0.5 -0.5 ...
 $ c_Num             : num  -0.5 -0.5 0.5 -0.5 -0.5 0.5 0.5 0.5 0.5 -0.5 ...
 $ f_Cnd             : Factor w/ 4 levels "Vg:Nm","Cr:Nm",..: 2 1 4 1 1 3 3 4 4 2 ...
 $ c_Ord             : num  0.5 -0.5 -0.5 0.5 0.5 -0.5 0.5 0.5 -0.5 0.5 ...
 $ c_Qty             : num  -0.5 0.5 -0.5 0.5 -0.5 0.5 -0.5 -0.5 0.5 0.5 ...
 $ RT                : int  1517 1920 2346 1773 2556 2043 2384 3078 1760 2218 ...
 $ RT_log            : num  7.32 7.56 7.76 7.48 7.85 ...
 $ RT_raw            : int  1517 1920 2346 1773 2556 2043 2384 3078 1760 2218 ...
 $ RTprev            : int  1517 1517 1920 2346 1773 2556 2043 2384 3078 1760 ...
 $ RTprev_log        : num  7.32 7.32 7.56 7.76 7.48 ...
 $ RTprev_raw        : int  1517 1517 1920 2346 1773 2556 2043 2384 3078 1760 ...
 $ Exp_Num           : num  6 34 26 54 6 34 26 36 24 34 ...
 $ Bline_Num         : num  15 25 35 45 15 25 35 45 15 25 ...
 $ Extr_Num          : num  24 16 44 36 24 16 44 54 6 16 ...
 $ Exp_side          : chr  "right" "right" "left" "left" ...
 $ Bline_side        : chr  "mid" "mid" "mid" "mid" ...
 $ Extr_side         : chr  "left" "left" "right" "right" ...
 $ response_num      : int  6 25 26 45 6 34 26 36 24 25 ...
 $ response_side     : Factor w/ 3 levels "left","mid","right": 3 2 1 2 3 3 3 3 3 2 ...
 $ response_category: Factor w/ 3 levels "borderline","expected",..: 2 1 2 1 2 2 2 2 2 1 ...
 $ Left              : int  24 16 26 54 24 16 44 54 6 34 ...
 $ Mid               : int  15 25 35 45 15 25 35 45 15 25 ...
 $ Right             : int  6 34 44 36 6 34 26 36 24 16 ...
 $ Instruction       : Factor w/ 17 levels "Choose a square with about 10 dots",..: 15 3 16 5 1 7 6 16 17 11 ...
 $ nchar_instr       : int  29 34 38 34 34 30 29 38 36 30 ...
```

2 Plots

2.1 Discriminability

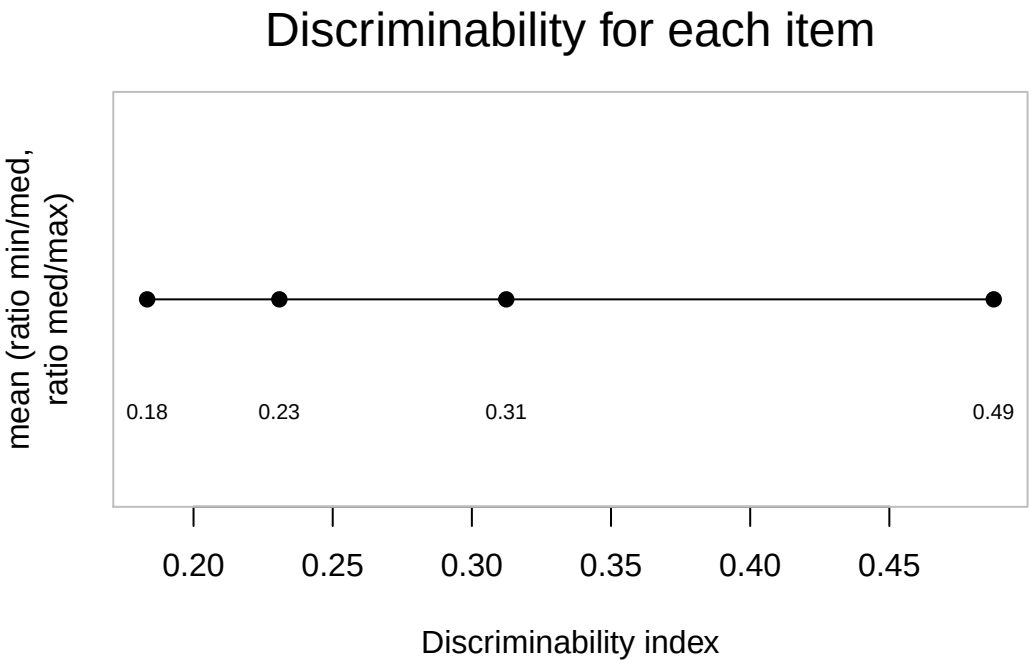


Figure 1: Ratios for different numbers of dots in the arrays: smaller values are more discriminable. Blue is for the ratio between the smallest number in the array and the largest number in an array. Red is for the mean of two ratios, one for the smallest number to the middle number, the other for the middle number to the largest number in the array

2.2 Consider using log RT

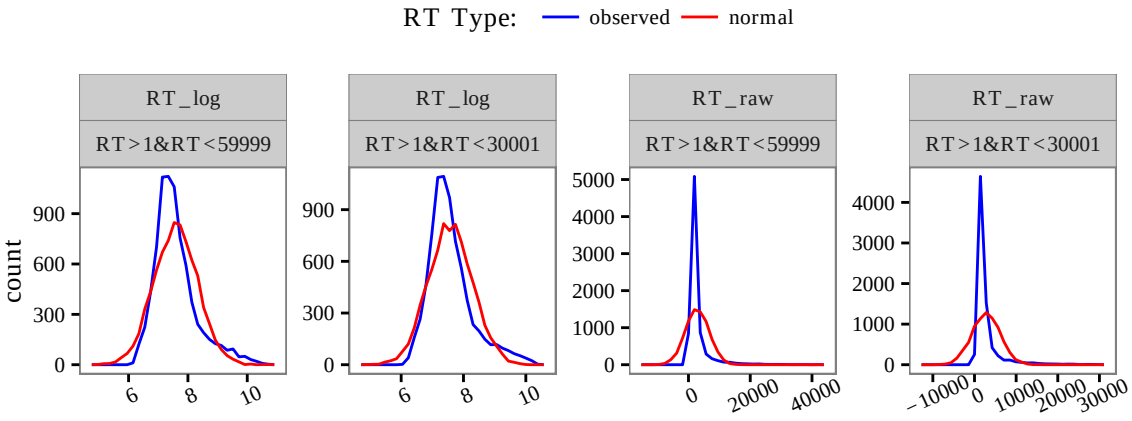


Figure 2: Compare distributions of the various transformations of RT against random samples from normal distributions with the same mean and sd to see which transformations best approximate normal distributions

2.3 How logging RT affects the distribution

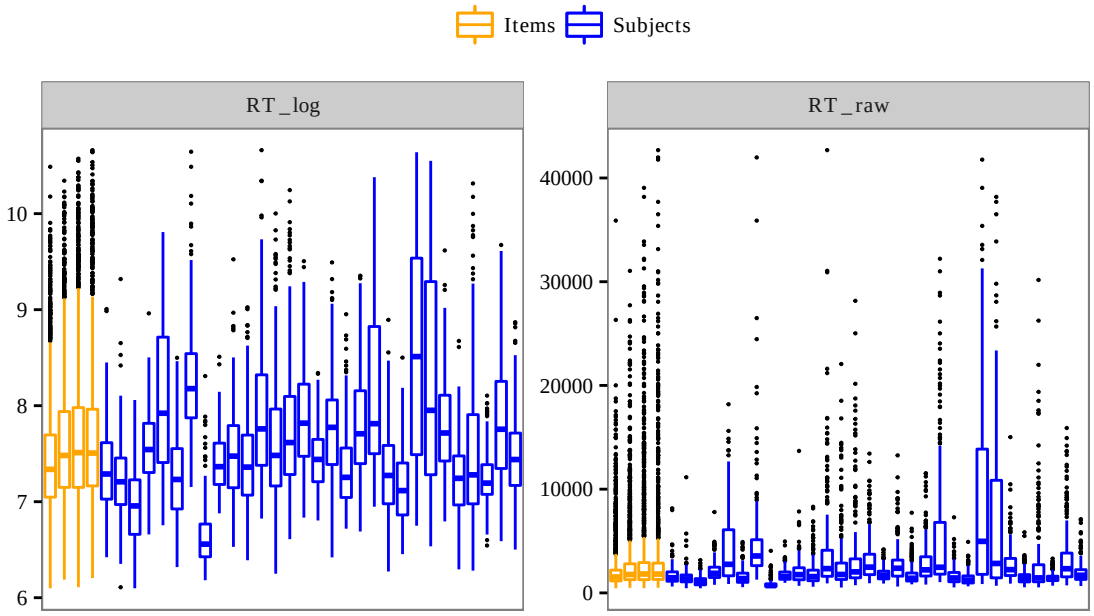


Figure 3: Show how transformations of RT affect distribution of times, and how they affect which times are outliers.

2.4 Identify fast and slow subjects and items

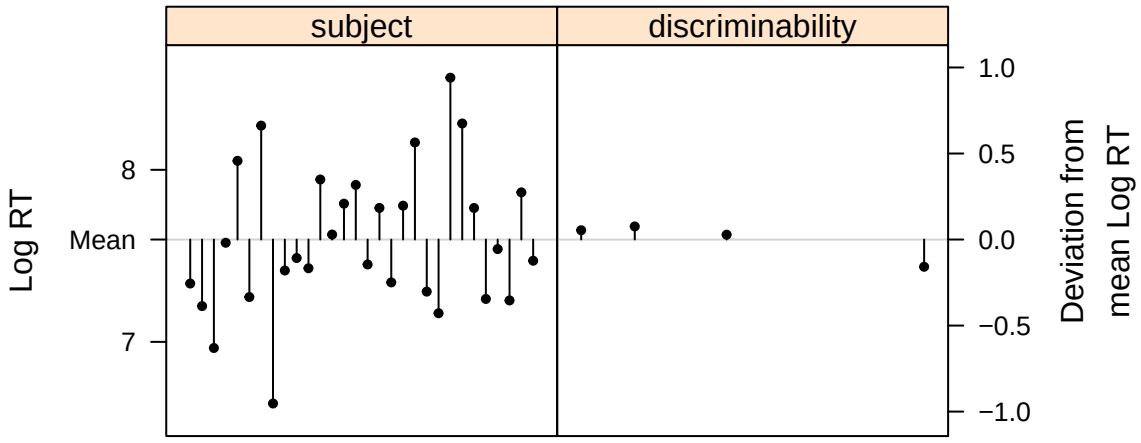


Figure 4: Show how mean times for individual subjects and items vary with respect to the grand mean Log RT.

2.5 Plot main effects in both transformations

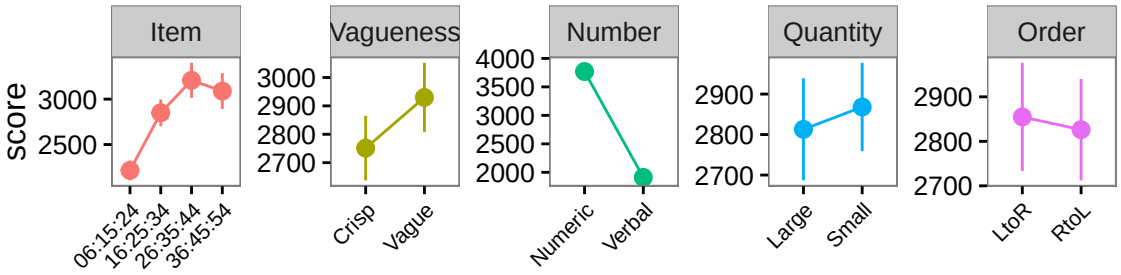


Figure 5: Plot main effects in raw RT

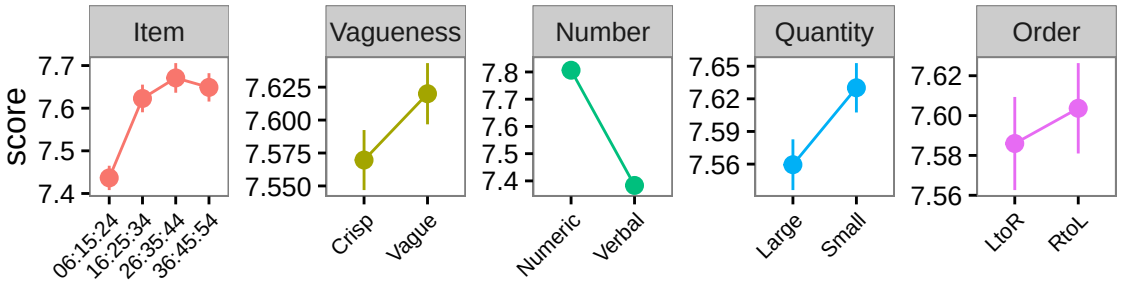


Figure 6: Plot main effects in log RT

2.6 Main effects in log RT over discriminability

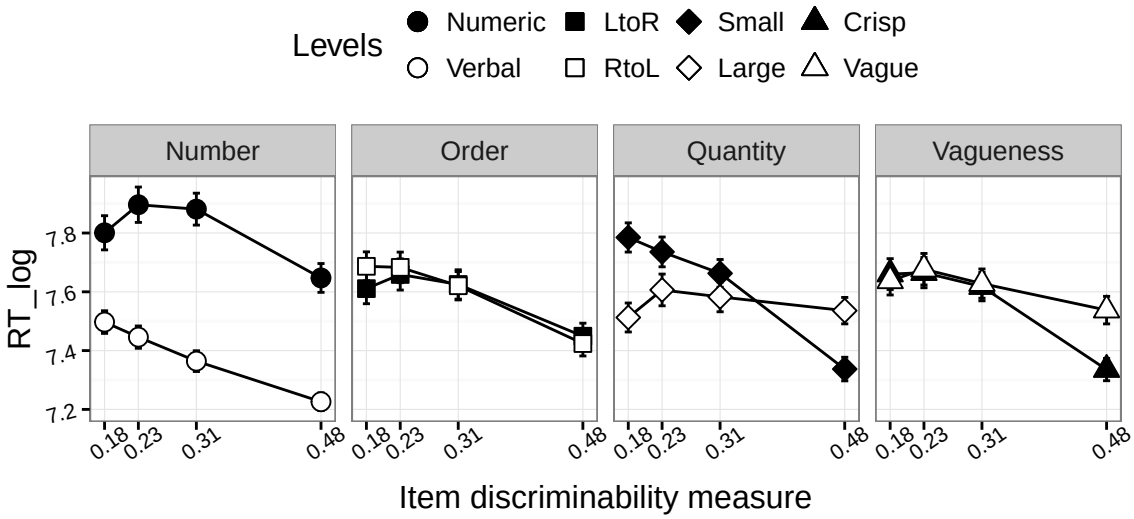


Figure 7: Main effects in log RT over discriminability

2.7 2-Way interactions over discriminability

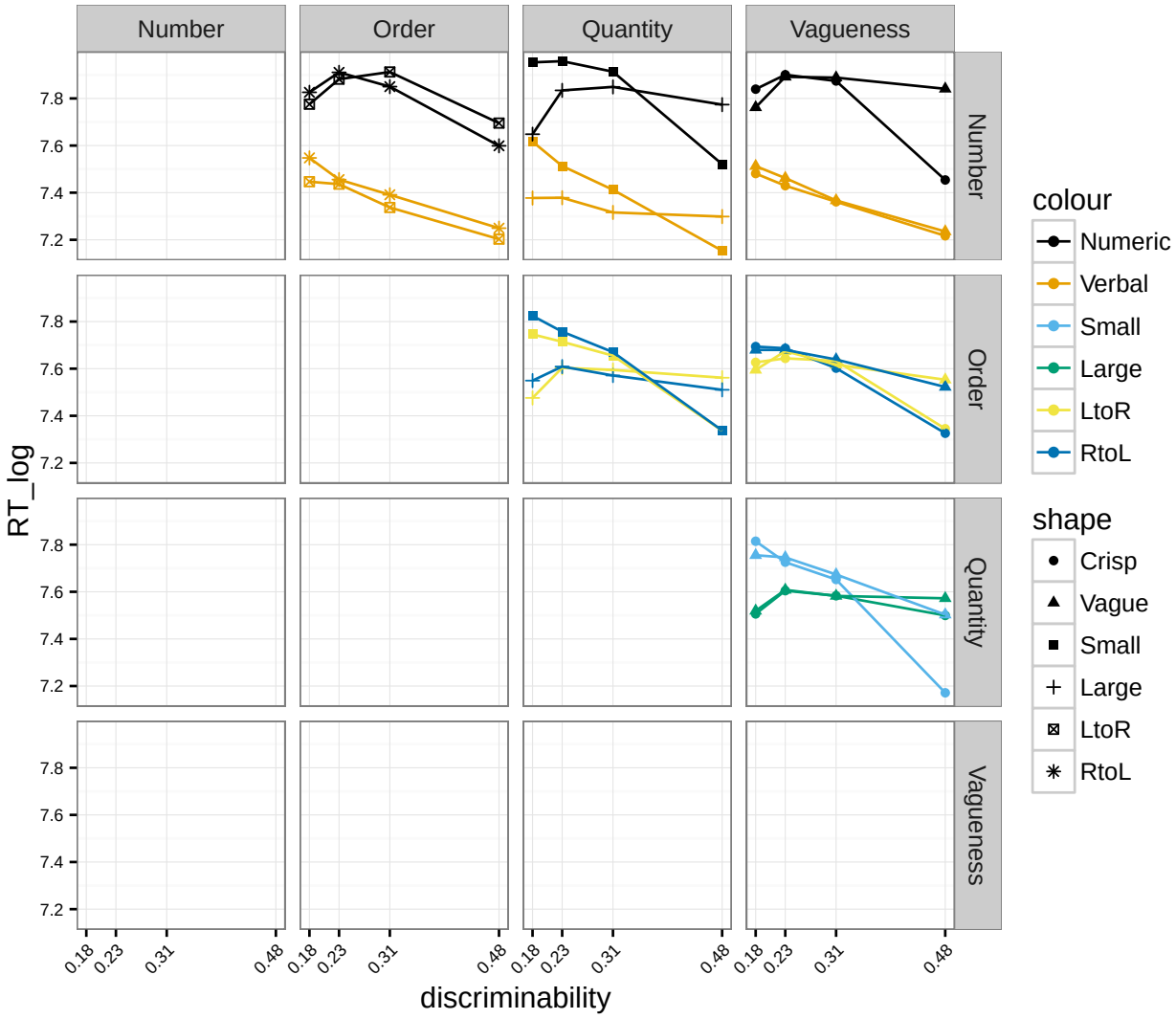


Figure 8: 2-way interactions over discriminability

2.8 Vagueness by number interaction over items

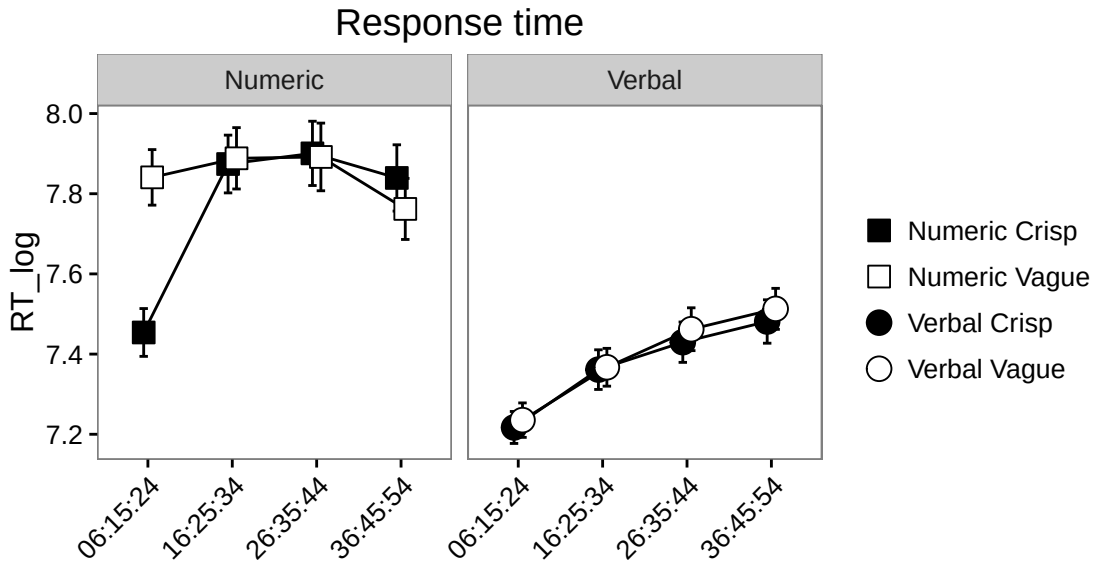


Figure 9: vagueness by number interaction over items

2.9 Vagueness by number interaction over discriminability

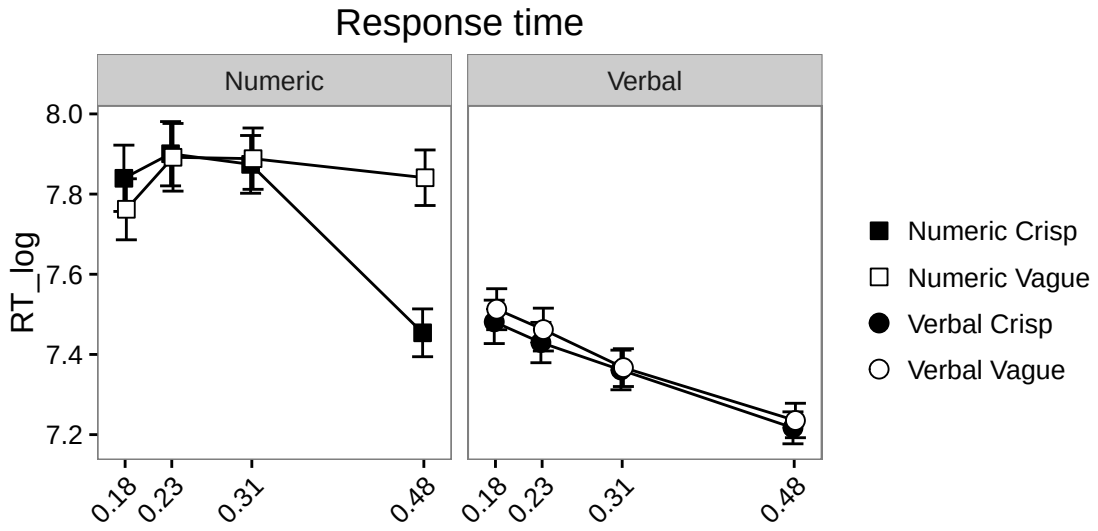


Figure 10: vagueness by number interaction over discriminability

2.10 3-Way interactions

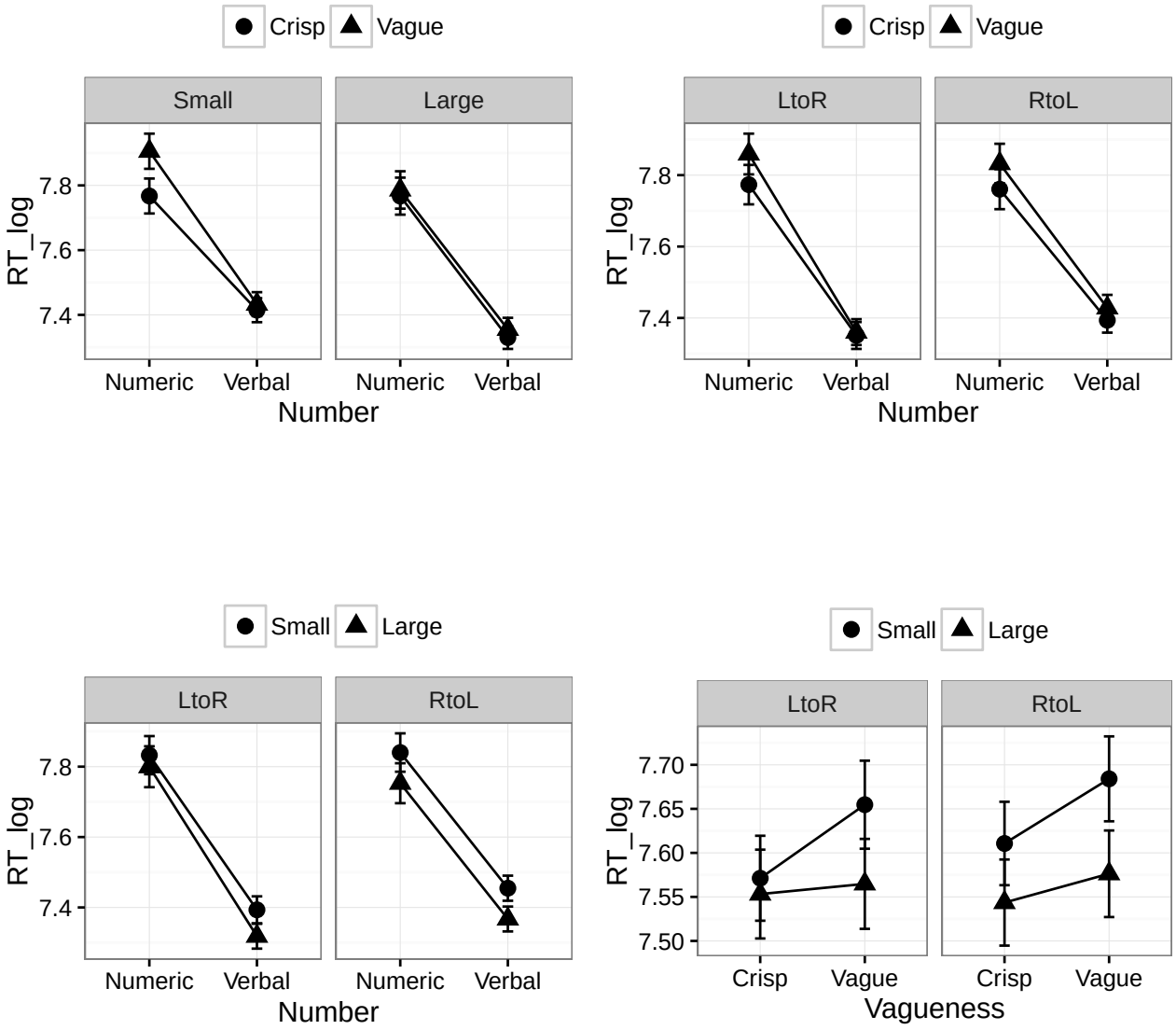


Figure 11: 3-Way interactions

2.11 Vagueness by number by quantity over discriminability

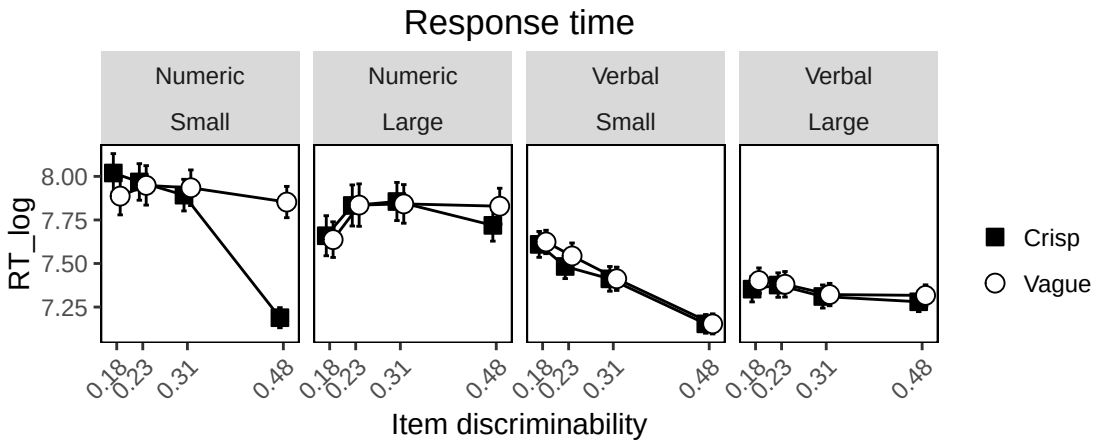


Figure 12: Vagueness by number by quantity over discriminability

2.12 4-Way interaction

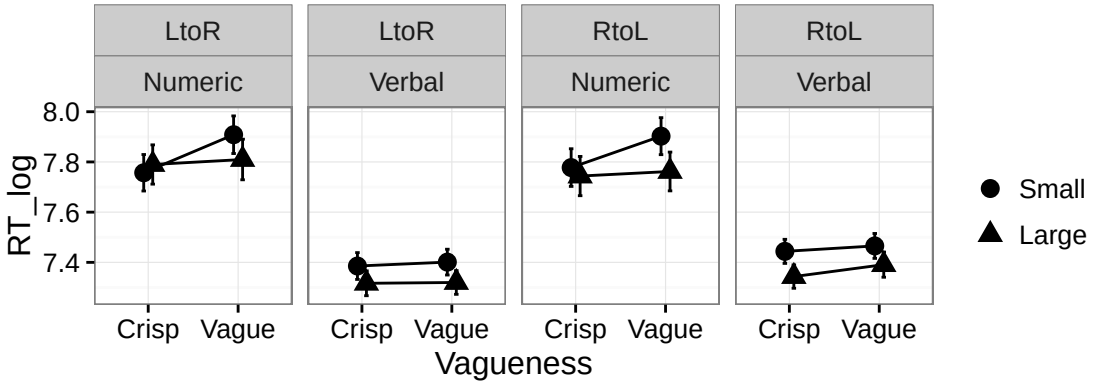


Figure 13: 4-Way interaction

2.13 4-Way interaction split over discriminability

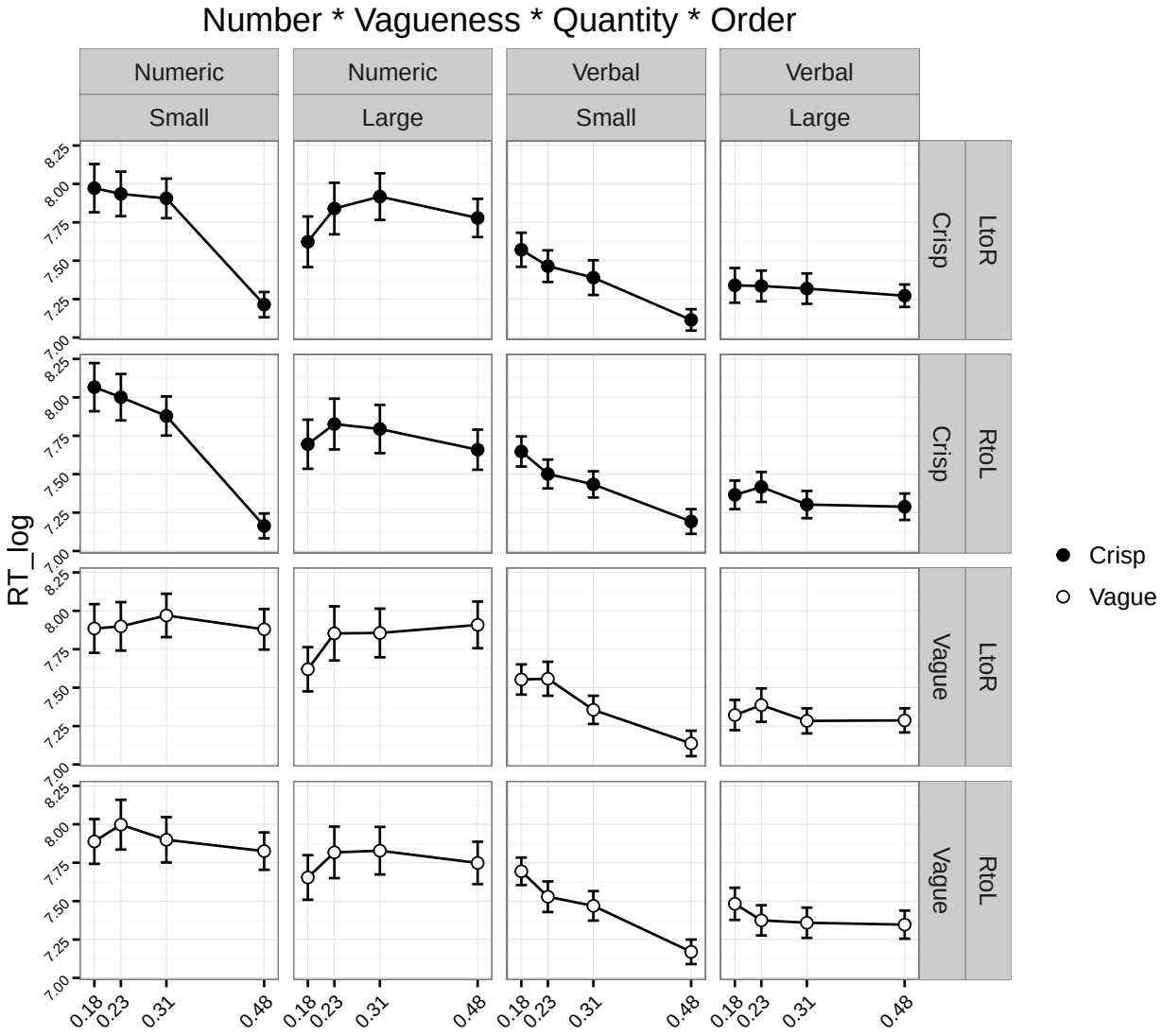


Figure 14: 4-Way interaction split over discriminability

3 Lmer model: before outlier removal

```
load("data_processed.Rda")
```

```
v5 <- lme4::lmer(data=dd,
                 RT_log ~
                   c_Vag + c_Num + c_Qty + c_Ord +
                   c_Num:c_Vag:c_Qty +
                   discriminability +
                   s_Trl +
                   RTprev_log +
                   nchar_instr +
                   (1+c_Vag + c_Num + c_Qty + c_Ord|Subject))
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: RT_log ~ c_Vag + c_Num + c_Qty + c_Ord + c_Num:c_Vag:c_Qty +
    discriminability + s_Trl + RTprev_log + nchar_instr + (1 +
    c_Vag + c_Num + c_Qty + c_Ord | Subject)
   Data: dd

REML criterion at convergence: 11474.8

Scaled residuals:
    Min      1Q  Median      3Q     Max
-4.5470 -0.6351 -0.0955  0.5372  5.0914

Random effects:
 Groups   Name        Variance Std.Dev. Corr
 Subject  (Intercept) 0.153949 0.39236
          c_Vag       0.001546 0.03932   0.69
          c_Num       0.165314 0.40659  -0.67 -0.64
          c_Qty       0.008148 0.09027   0.16  0.26 -0.34
          c_Ord       0.001559 0.03949  -0.13  0.02 -0.39 -0.52
 Residual             0.249734 0.49973
Number of obs: 7677, groups:  Subject, 30

Fixed effects:
                   Estimate Std. Error t value
(Intercept)        7.171685   0.122236   58.67
c_Vag              0.060938   0.013873    4.39
c_Num             -0.433614   0.075148   -5.77
c_Qty             -0.069067   0.020047   -3.45
c_Ord              0.017959   0.013497    1.33
discriminability  -0.771402   0.049267  -15.66
s_Trl             -0.106972   0.005807  -18.42
RTprev_log         0.060469   0.009692    6.24
nchar_instr        0.006086   0.001944    3.13
c_Vag:c_Num:c_Qty  0.104949   0.046066    2.28

Correlation of Fixed Effects:
            (Intr) c_Vag  c_Num  c_Qty  c_Ord  dscrmn s_Trl  RTprv_ nchr_n
c_Vag        0.083
c_Num       -0.369 -0.337
c_Qty        0.069  0.117 -0.278
c_Ord       -0.052  0.007 -0.206 -0.228
discrmnblty -0.140  0.004 -0.001  0.000  0.001
s_Trl       -0.114  0.003 -0.001 -0.002  0.006  0.017
RTprev_log  -0.607  0.006 -0.004 -0.003  0.019  0.015  0.185
nchar_instr -0.524  0.237 -0.034  0.018  0.000  0.017  0.000  0.006
c_Vg:c_N:_Q  0.073 -0.033  0.005 -0.003  0.000 -0.002 -0.001 -0.002 -0.138
```

	Estimate	Std. Error	t value
(Intercept)	7.17	0.12	58.67
c_Vag	0.06	0.01	4.39
c_Num	-0.43	0.08	-5.77
c_Qty	-0.07	0.02	-3.45
c_Ord	0.02	0.01	1.33
discriminability	-0.77	0.05	-15.66
s_Trl	-0.11	0.01	-18.42
RTprev_log	0.06	0.01	6.24
nchar_instr	0.01	0.00	3.13
c_Vag:c_Num:c_Qty	0.10	0.05	2.28

Table 1: xtable v5

```
R^2
[1] 0.533722
```

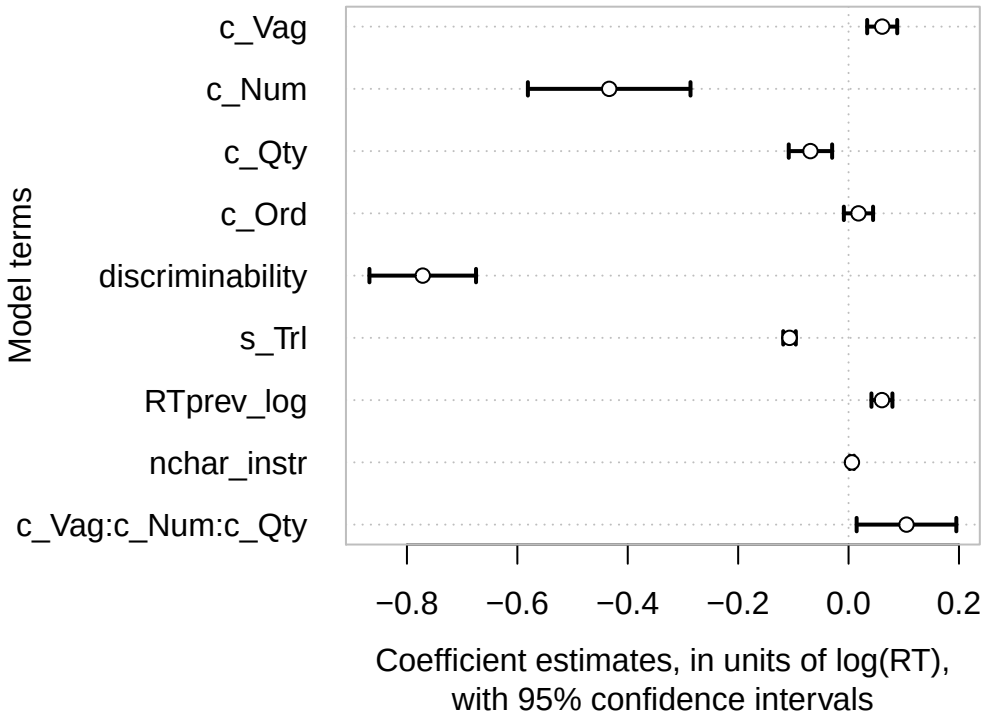


Figure 15: Coefficient estimates and their (Wald) 95 per cent confidence intervals

```
par(mfrow = c(2, 4))
plotLMER.fnc(v5)

effect size (range) for  c_Vag is  0.03470056
effect size (range) for  c_Num is  0.4073765
effect size (range) for  c_Qty is  0.09530422
effect size (range) for  c_Ord is  0.0179595
effect size (range) for  discriminability is  0.2346348
effect size (range) for  s_Trl is  0.369093
effect size (range) for  RTprev_log is  0.2759499
effect size (range) for  nchar_instr is  0.05477539
```

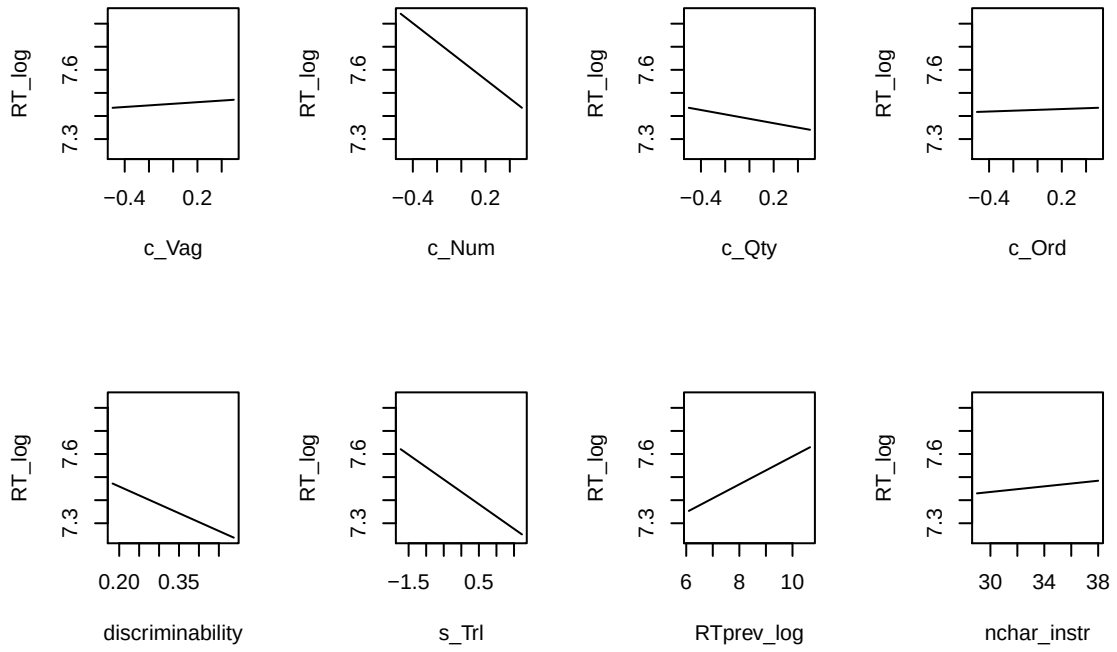


Figure 16: plotMLERfnc

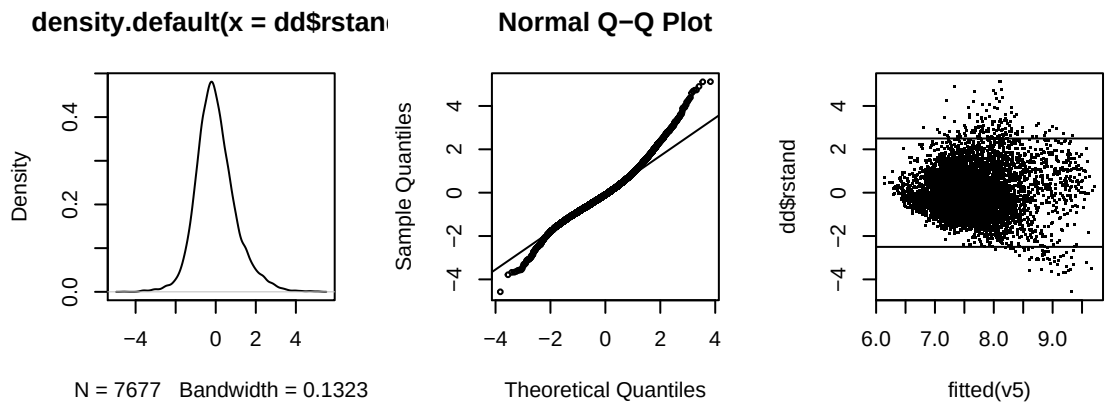


Figure 17: Baayen Model Criticism Plots

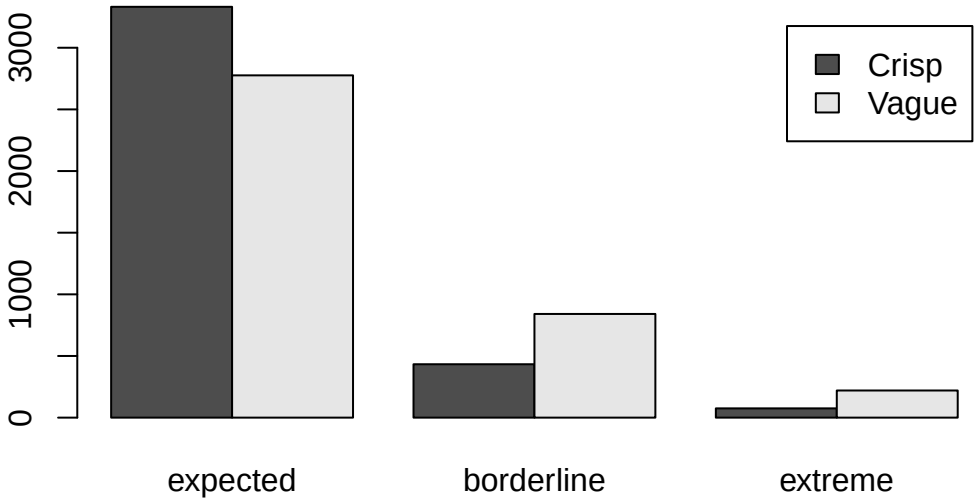
4 lmerTest Version

```
v6 <- lmerTest::lmer(data=dd,
                     RT_log ~
                       c_Vag + c_Num + c_Qty + c_Ord +
                       c_Num:c_Vag:c_Qty +
                       discriminability +
                       s_Trl +
                       RTprev_log +
                       nchar_instr +
                       (1+c_Vag + c_Num + c_Qty + c_Ord|Subject))
```

```
summary(v6)

Linear mixed model fit by REML t-tests use Satterthwaite approximations to degrees of freedom [
lmerMod]
Formula: RT_log ~ c_Vag + c_Num + c_Qty + c_Ord + c_Num:c_Vag:c_Qty +
    discriminability + s_Trl + RTprev_log + nchar_instr + (1 +
    c_Vag + c_Num + c_Qty + c_Ord | Subject)
   Data: dd

REML criterion at convergence: 11474.8

Scaled residuals:
    Min      1Q  Median      3Q     Max
-4.5470 -0.6351 -0.0955  0.5372  5.0914

Random effects:
 Groups   Name        Variance Std.Dev. Corr
 Subject  (Intercept) 0.153949 0.39236
          c_Vag       0.001546 0.03932   0.69
          c_Num       0.165314 0.40659  -0.67 -0.64
          c_Qty       0.008148 0.09027   0.16  0.26 -0.34
          c_Ord       0.001559 0.03949  -0.13  0.02 -0.39 -0.52
 Residual             0.249734 0.49973
Number of obs: 7677, groups:  Subject, 30

Fixed effects:
                    Estimate Std. Error         df t value Pr(>|t|)
(Intercept)        7.172e+00  1.222e-01  2.370e+02  58.671  < 2e-16 ***
c_Vag              6.094e-02  1.387e-02  3.300e+01   4.393 0.000112 ***
c_Num             -4.336e-01  7.515e-02  2.900e+01  -5.770 2.97e-06 ***
c_Qty             -6.907e-02  2.005e-02  2.900e+01  -3.445 0.001743 **
c_Ord              1.796e-02  1.350e-02  5.100e+01   1.331 0.189164
discriminability  -7.714e-01  4.927e-02  7.551e+03 -15.658  < 2e-16 ***
s_Trl             -1.070e-01  5.807e-03  7.558e+03 -18.421  < 2e-16 ***
RTprev_log         6.047e-02  9.692e-03  7.594e+03   6.239 4.63e-10 ***
nchar_instr        6.086e-03  1.944e-03  7.551e+03   3.131 0.001749 **
c_Vag:c_Num:c_Qty  1.049e-01  4.607e-02  7.551e+03   2.278 0.022742 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
            (Intr) c_Vag  c_Num  c_Qty  c_Ord  dscrmn s_Trl  RTprv_ nchr_n
c_Vag        0.083
c_Num       -0.369 -0.337
c_Qty        0.069  0.117 -0.278
c_Ord       -0.052  0.007 -0.206 -0.228
discrmnblty -0.140  0.004 -0.001  0.000  0.001
s_Trl       -0.114  0.003 -0.001 -0.002  0.006  0.017
RTprev_log  -0.607  0.006 -0.004 -0.003  0.019  0.015  0.185
nchar_instr -0.524  0.237 -0.034  0.018  0.000  0.017  0.000  0.006
c_Vg:c_N:_Q  0.073 -0.033  0.005 -0.003  0.000 -0.002 -0.001 -0.002 -0.138
```

5 Lmer model: after outlier removal

6 Borderline responses



	Crisp	Vague
expected	3332	2776
borderline	433	841
extreme	75	220

Table 2: Borderline cases counts

A Functions listing

A.1 Gather Data

```
gatherData <- function(number_of_valid_subjects) {
    data_dir <- "../experimentCode/output/"
    column.headers.df <- head(read.table(paste(data_dir, "subject01.data", sep = ""), header = TRUE),
        0)
    gathered.data <- column.headers.df
    for (subject in 1:number_of_valid_subjects) {
        current.filename <- paste(data_dir, "subject", sprintf("%02d", subject), ".data", sep = "")
        current.data <- read.table(file = current.filename, header = TRUE, stringsAsFactors = FALSE)
        gathered.data <- rbind(gathered.data, current.data)
    }
    message("Returning gathered data")
    return(gathered.data)
}  # end of gatherData function
```

A.2 Classify Response

```
classifyResponses <- function(dat) {
    # what were they expected to respond?
    dat$crossed <- as.factor(paste("Con", dat$Condition, ":Quan", dat$Quantity, ":Item", dat$Item, sep = ""))
    dat[dat$crossed == "Con1:Quan1:Item1", "Exp_Num"] <- 6
    dat[dat$crossed == "Con1:Quan1:Item1", "Bline_Num"] <- 15
    dat[dat$crossed == "Con1:Quan1:Item1", "Extr_Num"] <- 24
    dat[dat$crossed == "Con1:Quan1:Item2", "Exp_Num"] <- 16
    dat[dat$crossed == "Con1:Quan1:Item2", "Bline_Num"] <- 25
    dat[dat$crossed == "Con1:Quan1:Item2", "Extr_Num"] <- 34
    dat[dat$crossed == "Con1:Quan1:Item3", "Exp_Num"] <- 26
    dat[dat$crossed == "Con1:Quan1:Item3", "Bline_Num"] <- 35
    dat[dat$crossed == "Con1:Quan1:Item3", "Extr_Num"] <- 44
    dat[dat$crossed == "Con1:Quan1:Item4", "Exp_Num"] <- 36
    dat[dat$crossed == "Con1:Quan1:Item4", "Bline_Num"] <- 45
    dat[dat$crossed == "Con1:Quan1:Item4", "Extr_Num"] <- 54
    dat[dat$crossed == "Con1:Quan2:Item1", "Exp_Num"] <- 24
    dat[dat$crossed == "Con1:Quan2:Item1", "Bline_Num"] <- 15
    dat[dat$crossed == "Con1:Quan2:Item1", "Extr_Num"] <- 6
    dat[dat$crossed == "Con1:Quan2:Item2", "Exp_Num"] <- 34
    dat[dat$crossed == "Con1:Quan2:Item2", "Bline_Num"] <- 25
    dat[dat$crossed == "Con1:Quan2:Item2", "Extr_Num"] <- 16
    dat[dat$crossed == "Con1:Quan2:Item3", "Exp_Num"] <- 44
    dat[dat$crossed == "Con1:Quan2:Item3", "Bline_Num"] <- 35
    dat[dat$crossed == "Con1:Quan2:Item3", "Extr_Num"] <- 26
    dat[dat$crossed == "Con1:Quan2:Item4", "Exp_Num"] <- 54
    dat[dat$crossed == "Con1:Quan2:Item4", "Bline_Num"] <- 45
    dat[dat$crossed == "Con1:Quan2:Item4", "Extr_Num"] <- 36
    dat[dat$crossed == "Con2:Quan1:Item1", "Exp_Num"] <- 6
    dat[dat$crossed == "Con2:Quan1:Item1", "Bline_Num"] <- 15
    dat[dat$crossed == "Con2:Quan1:Item1", "Extr_Num"] <- 24
    dat[dat$crossed == "Con2:Quan1:Item2", "Exp_Num"] <- 16
    dat[dat$crossed == "Con2:Quan1:Item2", "Bline_Num"] <- 25
    dat[dat$crossed == "Con2:Quan1:Item2", "Extr_Num"] <- 34
    dat[dat$crossed == "Con2:Quan1:Item3", "Exp_Num"] <- 26
    dat[dat$crossed == "Con2:Quan1:Item3", "Bline_Num"] <- 35
    dat[dat$crossed == "Con2:Quan1:Item3", "Extr_Num"] <- 44
    dat[dat$crossed == "Con2:Quan1:Item4", "Exp_Num"] <- 36
    dat[dat$crossed == "Con2:Quan1:Item4", "Bline_Num"] <- 45
    dat[dat$crossed == "Con2:Quan1:Item4", "Extr_Num"] <- 54
    dat[dat$crossed == "Con2:Quan2:Item1", "Exp_Num"] <- 24
    dat[dat$crossed == "Con2:Quan2:Item1", "Bline_Num"] <- 15
    dat[dat$crossed == "Con2:Quan2:Item1", "Extr_Num"] <- 6
    dat[dat$crossed == "Con2:Quan2:Item2", "Exp_Num"] <- 34
    dat[dat$crossed == "Con2:Quan2:Item2", "Bline_Num"] <- 25
    dat[dat$crossed == "Con2:Quan2:Item2", "Extr_Num"] <- 16
    dat[dat$crossed == "Con2:Quan2:Item3", "Exp_Num"] <- 44
    dat[dat$crossed == "Con2:Quan2:Item3", "Bline_Num"] <- 35
    dat[dat$crossed == "Con2:Quan2:Item3", "Extr_Num"] <- 26
    dat[dat$crossed == "Con2:Quan2:Item4", "Exp_Num"] <- 54
    dat[dat$crossed == "Con2:Quan2:Item4", "Bline_Num"] <- 45
    dat[dat$crossed == "Con2:Quan2:Item4", "Extr_Num"] <- 36
    dat[dat$crossed == "Con3:Quan1:Item1", "Exp_Num"] <- 6
    dat[dat$crossed == "Con3:Quan1:Item1", "Bline_Num"] <- 15
    dat[dat$crossed == "Con3:Quan1:Item1", "Extr_Num"] <- 24
    dat[dat$crossed == "Con3:Quan1:Item2", "Exp_Num"] <- 16
    dat[dat$crossed == "Con3:Quan1:Item2", "Bline_Num"] <- 25
    dat[dat$crossed == "Con3:Quan1:Item2", "Extr_Num"] <- 34
    dat[dat$crossed == "Con3:Quan1:Item3", "Exp_Num"] <- 26
    dat[dat$crossed == "Con3:Quan1:Item3", "Bline_Num"] <- 35
    dat[dat$crossed == "Con3:Quan1:Item3", "Extr_Num"] <- 44
    dat[dat$crossed == "Con3:Quan1:Item4", "Exp_Num"] <- 36
    dat[dat$crossed == "Con3:Quan1:Item4", "Bline_Num"] <- 45
    dat[dat$crossed == "Con3:Quan1:Item4", "Extr_Num"] <- 54
    dat[dat$crossed == "Con3:Quan2:Item1", "Exp_Num"] <- 24
    dat[dat$crossed == "Con3:Quan2:Item1", "Bline_Num"] <- 15
    dat[dat$crossed == "Con3:Quan2:Item1", "Extr_Num"] <- 6
    dat[dat$crossed == "Con3:Quan2:Item2", "Exp_Num"] <- 34
    dat[dat$crossed == "Con3:Quan2:Item2", "Bline_Num"] <- 25
    dat[dat$crossed == "Con3:Quan2:Item2", "Extr_Num"] <- 16
    dat[dat$crossed == "Con3:Quan2:Item3", "Exp_Num"] <- 44
    dat[dat$crossed == "Con3:Quan2:Item3", "Bline_Num"] <- 35
    dat[dat$crossed == "Con3:Quan2:Item3", "Extr_Num"] <- 26
    dat[dat$crossed == "Con3:Quan2:Item4", "Exp_Num"] <- 54
```

```
dat[dat$crossed == "Con3:Quan2:Item4", "Bline_Num"] <- 45
dat[dat$crossed == "Con3:Quan2:Item4", "Extr_Num"] <- 36
dat[dat$crossed == "Con4:Quan1:Item1", "Exp_Num"] <- 6
dat[dat$crossed == "Con4:Quan1:Item1", "Bline_Num"] <- 15
dat[dat$crossed == "Con4:Quan1:Item1", "Extr_Num"] <- 24
dat[dat$crossed == "Con4:Quan1:Item2", "Exp_Num"] <- 16
dat[dat$crossed == "Con4:Quan1:Item2", "Bline_Num"] <- 25
dat[dat$crossed == "Con4:Quan1:Item2", "Extr_Num"] <- 34
dat[dat$crossed == "Con4:Quan1:Item3", "Exp_Num"] <- 26
dat[dat$crossed == "Con4:Quan1:Item3", "Bline_Num"] <- 35
dat[dat$crossed == "Con4:Quan1:Item3", "Extr_Num"] <- 44
dat[dat$crossed == "Con4:Quan1:Item4", "Exp_Num"] <- 36
dat[dat$crossed == "Con4:Quan1:Item4", "Bline_Num"] <- 45
dat[dat$crossed == "Con4:Quan1:Item4", "Extr_Num"] <- 54
dat[dat$crossed == "Con4:Quan2:Item1", "Exp_Num"] <- 24
dat[dat$crossed == "Con4:Quan2:Item1", "Bline_Num"] <- 15
dat[dat$crossed == "Con4:Quan2:Item1", "Extr_Num"] <- 6
dat[dat$crossed == "Con4:Quan2:Item2", "Exp_Num"] <- 34
dat[dat$crossed == "Con4:Quan2:Item2", "Bline_Num"] <- 25
dat[dat$crossed == "Con4:Quan2:Item2", "Extr_Num"] <- 16
dat[dat$crossed == "Con4:Quan2:Item3", "Exp_Num"] <- 44
dat[dat$crossed == "Con4:Quan2:Item3", "Bline_Num"] <- 35
dat[dat$crossed == "Con4:Quan2:Item3", "Extr_Num"] <- 26
dat[dat$crossed == "Con4:Quan2:Item4", "Exp_Num"] <- 54
dat[dat$crossed == "Con4:Quan2:Item4", "Bline_Num"] <- 45
dat[dat$crossed == "Con4:Quan2:Item4", "Extr_Num"] <- 36
dat$crossed <- NULL
# what side LEFT, MIDDLE, RIGHT corresponds with Expected, Borderline, Extreme?
for (row in 1:nrow(dat)) {
    if (dat[row, "Exp_Num"] == dat[row, "Left"]) {
        dat[row, "Exp_side"] <- "left"
    }
    if (dat[row, "Exp_Num"] == dat[row, "Mid"]) {
        dat[row, "Exp_side"] <- "mid"
    }
    if (dat[row, "Exp_Num"] == dat[row, "Right"]) {
        dat[row, "Exp_side"] <- "right"
    }
    if (dat[row, "Bline_Num"] == dat[row, "Left"]) {
        dat[row, "Bline_side"] <- "left"
    }
    if (dat[row, "Bline_Num"] == dat[row, "Mid"]) {
        dat[row, "Bline_side"] <- "mid"
    }
    if (dat[row, "Bline_Num"] == dat[row, "Right"]) {
        dat[row, "Bline_side"] <- "right"
    }
    if (dat[row, "Extr_Num"] == dat[row, "Left"]) {
        dat[row, "Extr_side"] <- "left"
    }
    if (dat[row, "Extr_Num"] == dat[row, "Mid"]) {
        dat[row, "Extr_side"] <- "mid"
    }
    if (dat[row, "Extr_Num"] == dat[row, "Right"]) {
        dat[row, "Extr_side"] <- "right"
    }
}
# what button press did the subject actually make? LEFT, MIDDLE, RIGHT, NOANSWER?
dat$RESPONSE <- as.factor(dat$RESPONSE)
# what number of dots corresponds with the subject's button press?
for (row in 1:nrow(dat)) {
    switch(as.character(dat[row, "RESPONSE"]), LEFT = {
        dat[row, "response_num"] <- dat[row, "Left"]
    }, MIDDLE = {
        dat[row, "response_num"] <- dat[row, "Mid"]
    }, RIGHT = {
        dat[row, "response_num"] <- dat[row, "Right"]
    }, NOANSWER = {
        dat[row, "response_num"] <- NA
    })
}
# what side was the subject's button-press? Left, mid right?
dat$response_side <- tolower(dat$RESPONSE)
dat$response_side[dat$response_side == "middle"] <- "mid"
dat$response_side <- factor(dat$response_side, exclude = "noanswer")
# what category was the subject's response? Expected, Borderline, Extreme
```

```
    dat$response_category <- "nocat"
    for (row in row.names(na.omit(dat))) {
        if (dat[row, "response_num"] == dat[row, "Exp_Num"]) {
            dat[row, "response_category"] <- "expected"
        }
        if (dat[row, "response_num"] == dat[row, "Bline_Num"]) {
            dat[row, "response_category"] <- "borderline"
        }
        if (dat[row, "response_num"] == dat[row, "Extr_Num"]) {
            dat[row, "response_category"] <- "extreme"
        }
    }
    dat$response_category <- factor(dat$response_category, exclude = "nocat")
    dat$RESPONSE <- NULL
    message("Returning classified data")
    return(dat)
}  # end of classifyResponse function
```

A.3 Processing

```
processData <- function(dat) {
    # SUBJECT
    dat$Subject <- factor(paste("s", sprintf("%02d", dat$Subject), sep = ""))
    # TRIAL
    dat$Trial <- rep(x = 1:number_of_trials_per_subject, times = number_of_valid_subjects)
    # make a centred Trial for modeling
    dat$c_Trl <- dat$Trial - mean(dat$Trial)
    # make a scaled Trial for modelling
    dat$s_Trl <- as.numeric(scale(dat$Trial))
    # ID id is a unique identifier for the 7680 row data
    dat$id <- factor(paste(paste(dat$Subject), paste("t", sprintf("%03d", dat$Trial), sep = ""), sep = ":"))
    # ITEM create a centred numeric item variable for modeling
    dat$c_Itm <- ifelse(dat$Item == 1, -0.75, ifelse(dat$Item == 2, -0.25, ifelse(dat$Item == 3, 0.25,
        0.75)))
    # make Item be a factor and assign labels
    dat$Item <- factor(dat$Item, levels = c(1, 2, 3, 4), labels = c("06:15:24", "16:25:34", "26:35:44",
        "36:45:54"))
    # VAGUENESS Create a factor coding for Vagueness
    dat[dat$Condition == 1, "Vagueness"] <- "Vague"
    dat[dat$Condition == 2, "Vagueness"] <- "Crisp"
    dat[dat$Condition == 3, "Vagueness"] <- "Vague"
    dat[dat$Condition == 4, "Vagueness"] <- "Crisp"
    dat$Vagueness <- as.factor(dat$Vagueness)
    # manually center Vagueness
    dat$c_Vag <- ifelse(dat$Vagueness == "Crisp", -0.5, 0.5)
    # NUMBER Create a factor coding for Number use
    dat[dat$Condition == 1, "Number"] <- "Numeric"
    dat[dat$Condition == 2, "Number"] <- "Numeric"
    dat[dat$Condition == 3, "Number"] <- "Verbal"
    dat[dat$Condition == 4, "Number"] <- "Verbal"
    dat$Number <- as.factor(dat$Number)
    # manually center Number
    dat$c_Num <- ifelse(dat$Number == "Numeric", -0.5, 0.5)
    # CONDITION make a factor out of Condition, as f_Cnd
    dat$f_Cnd <- factor(dat$Condition, levels = c(1, 2, 3, 4), labels = c("Vg:Nm", "Cr:Nm", "Vg:Vb",
        "Cr:Vb"))
    # ORDER give the levels of Order meaningful names
    dat$Order <- factor(dat$Order, levels = c(1, 2), labels = c("LtoR", "RtoL"))
    # make a manually centred Order
    dat$c_Ord <- ifelse(dat$Order == "LtoR", -0.5, 0.5)
    # QUANTITY give the levels of Quantity meaningful names
    dat$Quantity <- factor(dat$Quantity, levels = c(1, 2), labels = c("Small", "Large"))
    # make a manually centred Quantity
    dat$c_Qty <- ifelse(dat$Quantity == "Small", -0.5, 0.5)
    # INSTRUCTION add number of characters in the instruction # 29 30 34 36 38
    dat$nchar_instr <- nchar(dat$Instruction)
    dat$nchar_instr_scaled <- as.vector(scale(nchar(dat$Instruction), scale = TRUE))
    # make Instruction be a factor (17 levels)
    dat$Instruction <- as.factor(dat$Instruction)
    # RT add transformations of RT
    dat$RT_log <- log(dat$RT)
    dat$RT_raw <- dat$RT
    # print to file a table with information about the design
    design_info <- unique(subset(dat, select = c(Item, Condition, Vagueness, Number, Quantity, Order,
        Left, Mid, Right, Exp_Num, Bline_Num, Extr_Num, Exp_side, Bline_side, Extr_side, Instruction)))
    design.info <- design_info[order(design_info$Item, design_info$Condition, design_info$Quantity, design_info$Order),
        ]
    row.names(design_info) <- NULL
    capture.output(print.data.frame(design_info, row.names = F, print.gap = 3, quote = F, right = F),
        file = "design_info-table.txt")
    # Add discriminability metric with reference to item
    discriminability_range <- c(0.75, 0.5294118, 0.4090909, 0.3333333)
    discriminability_range_scaled <- c(1.3441995, 0.1316642, -0.5297187, -0.946145)
    discriminability <- c(0.4875, 0.3123529, 0.2308442, 0.1833333)
    discriminability_scaled <- c(1.37582241, 0.06614191, -0.54334858, -0.89861574)
    dat[dat$Item == "06:15:24", "discriminability"] <- 0.4875
    dat[dat$Item == "16:25:34", "discriminability"] <- 0.3123529
    dat[dat$Item == "26:35:44", "discriminability"] <- 0.2308442
    dat[dat$Item == "36:45:54", "discriminability"] <- 0.1833333
    # put dat in better column order
    dat <- subset(dat, select = c(id, Subject, Trial, Condition, Order, Quantity, Vagueness, Number,
        Item, discriminability, c_Trl, s_Trl, c_Itm, c_Vag, c_Num, f_Cnd, c_Ord, c_Qty, RT, RT_log, RT_raw,
        Exp_Num, Bline_Num, Extr_Num, Exp_side, Bline_side, Extr_side, response_num, response_side, response_category,
```

```
        Left, Mid, Right, Instruction, nchar_instr))
    # This data set (dat) contains *all* trials 7680 including impossible trials and is mainly for graphs
    # comparing different removals
    save(dat, file = "data_raw.Rda")
    message("Returning processed data")
    return(dat)
}  # end of function processData
```

A.4 Post-processing

```
postProcessData <- function(dat) {
    # dd removes impossible trials from dat Throw out RT = 1 and RT = 59998, and RTprev = 1 and RTprev =
    # 59998 i.e., throw out sticky fingers and timeouts, and the trials that followed sticky fingers and
    # timeouts since they were likely affected by unusual previous trials. Also lose impossible trials
    dd <- dat
    dd$RT[dd$RT == 1] <- NA
    dd$RT[dd$RT == 59998] <- NA
    dd <- dd[complete.cases(dd), ]
    row.names(dd) <- NULL
    # add preceding RT: because we removed impossible trials, the value for preceding RT for a trial
    # following an impossible trial is the value of the trial that preceded the impossible trial.
    dd$RTprev <- NA
    for (s in levels(dd$Subject)) {
        nrows <- nrow(dd[dd$Subject == s, ])
        for (i in 1:nrows) {
            if (i == 1) {
                dd[dd$Subject == s, "RTprev"][i] <- dd[dd$Subject == s, "RT"][i]
            } else dd[dd$Subject == s, "RTprev"][i] <- dd[dd$Subject == s, "RT"][i - 1]
        }
    }
    # add transformations of previous RT
    dd$RTprev_log <- log(dd$RTprev)
    dd$RTprev_raw <- dd$RTprev
    # put dd in better column order
    dd <- subset(dd, select = c(id, Subject, Trial, Condition, Order, Quantity, Vagueness, Number, Item,
        discriminability, c_Trl, s_Trl, c_Itm, c_Vag, c_Num, f_Cnd, c_Ord, c_Qty, RT, RT_log, RT_raw,
        RTprev, RTprev_log, RTprev_raw, Exp_Num, Bline_Num, Extr_Num, Exp_side, Bline_side, Extr_side,
        response_num, response_side, response_category, Left, Mid, Right, Instruction, nchar_instr))
    save(dd, file = "data_processed.Rda")
    message("Returning post processed data")
    return(dd)
}  # end of function postProcessResponses
```