Code practice. From t-tests to data modeling

Linguistic data: Quantitative analysis and vizualization

Universal linguistic hierarchies: a case of Modern Greek (Standard and Cypriot dialects)

Data (responces, quesionnaire) adapted from the survey: Leivada, Evelina; Westergaard, Marit, 2019, Universal linguistic hierarchies are not innately wired. PeerJ, v.7.

Source of data: TROLLing repository: Leivada, Evelina; Westergaard, Marit, 2019, "Replication Data for: Universal linguistic hierarchies are not innately wired", https://doi.org/10.18710/NTLLUF, DataverseNO, V1

Disclaimer Tables and figures produced by your code can look slightly different from what you see in the article. This concerns absolute numbers, the size of columns, the color of lines, themes, etc. Still, reproduce the visualization type (e.g. barplot, violing plot) and the order of elements/groups just as it was plotted by Leivada and Westergaard.

Constructions with two adjectives In English, the order of two adjectives in phrases like:

```
 a big black bag # ok
*a black big bag # unacceptable, ungrammatically ill-formed, or semantically anomalous
```

is powered by the semantic class of adjective (e.g. the `color` adjective closer to the noun than the `size` adjective).

A syntactic hierarchy of closeness to the noun in Chomsky's Universal Grammar suggests the following order and is claimed to be innate and universal (= valid for all languages).

```
Subjective Comment > Evidential > Size > Length
> Height > Speed > Depth > Width > Temperature > Wetness > Age
> Shape > Color > Nationality/Origin > Material
# (adapted from Scott, 2002: 114)
```

The goal of Leivada & Westergaard research is identify what happens when people process orderings that either comply with the hierrarchy or violate it.

Method In the first experiment, 140 neurotypical, adult speakers completed a timed forced choice task that featured stimuli showing a combination of two adjectives and a concrete noun (e.g., *I bought a square black table*). Two types of responses were collected:

(i) acceptability judgments on a 3-point Likert scale that featured the options
   1. wrong,
   2. neither correct nor wrong,
   3. correct;

(ii) reaction times (RT).

The task featured three conditions: 1. size adjective > nationality adjective, 2. color adjective > shape adjective, 3. subjective comment adjective > material adjective. Each condition had two orders. In the congruent order, the adjective pair was ordered in agreement with what is traditionally accepted as dictated

by the universal hierarchy. In the incongruent order, the ordering was reversed, thus the hierarchy was violated.

In the second experiment, 30 bidialectals (native speakers of Standard and Cypriot Greek) were tested in both language varieties, 36 observations per participant, 18 for each variety.

Two kinds of fillers were used in both experiments, FillerAcceptable and FillerUnacceptable – sentences that included well-formed and ungrammatical structures, respectively. In both tasks the ratio of fillers to actual test structures was 2:1.

```
library(tidyverse)
```

Data

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
## v ggplot2 3.3.4     v purrr   0.3.4
## v tibble  3.1.2     v dplyr   1.0.6
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
mono_socio <- read_csv2("https://raw.githubusercontent.com/LingData2019/LingData2020/master/data/greek-
```

```
## i Using '\',\'' as decimal and '\'.\'' as grouping mark. Use `read_delim()` for more control.
##
## -- Column specification --------------------------------------------------------
## cols(
##   ParticipantID = col_character(),
##   TypeOfQuestion = col_character(),
##   QuestionCategory = col_character(),
##   Responce = col_character()
## )
```

```
mono <- read_csv2("https://raw.githubusercontent.com/LingData2019/LingData2020/master/data/greek-word-o
```

```
## i Using '\',\'' as decimal and '\'.\'' as grouping mark. Use `read_delim()` for more control.
##
## -- Column specification --------------------------------------------------------
## cols(
##   ParticipantID = col_character(),
##   TypeOfQuestion = col_character(),
##   TypeOfStimuli = col_character(),
##   WordOrder = col_character(),
##   ResponseAcceptabilityJudgement = col_character(),
##   RT = col_double()
## )
```

see also reading key for the data

```
mono_socio
```

```
## # A tibble: 180 x 4
##    ParticipantID                TypeOfQuestion QuestionCategory Responce
```

```
##    <chr>                          <chr>          <chr>           <chr>
##  1 2bc93481d25426f259a0f1a3d5d3e~ Form           age             58
##  2 2bc93481d25426f259a0f1a3d5d3e~ Form           country1        Nicosia
##  3 2bc93481d25426f259a0f1a3d5d3e~ Form           country2        Kaimakli,Nico~
##  4 2bc93481d25426f259a0f1a3d5d3e~ Form           education       College
##  5 2bc93481d25426f259a0f1a3d5d3e~ Form           handedness      right
##  6 2bc93481d25426f259a0f1a3d5d3e~ Form           sex             female
##  7 4b4c9debd4aedeb1fd3dfc9f1cab3~ Form           age             24
##  8 4b4c9debd4aedeb1fd3dfc9f1cab3~ Form           country1        Nicosia
##  9 4b4c9debd4aedeb1fd3dfc9f1cab3~ Form           country2        Nicosia
## 10 4b4c9debd4aedeb1fd3dfc9f1cab3~ Form           education       UniversityDeg~
## # ... with 170 more rows
```

1. Data overview

1.1

Use `mono_socio` dataframe to answer the following questions:

1. How many participants are mentioned in this dataframe?
2. How many of them are males and females?
3. Which education levels are mentioned in the dataframe?
4. How many participants of each education levels are present?
5. How many left- and right-randed participants are present?

The following functions from tidyverse can be usefult for this problem: `filter`, `group_by`, `count` and `distinct`. (Another approach is to use `pivot_wider`.)

```
length(unique(mono_socio$ParticipantID))
```

```
## [1] 30
```

```
sexes <- mono_socio %>% filter(QuestionCategory=='sex')
table(sexes$Responce)
```

```
##
## female   male
##     19     11
```

```
educations <- mono_socio %>% filter(QuestionCategory=='education')
unique(educations$Responce)
```

```
## [1] "College"          "UniversityDegree" "Postgraduate"     "PhDongoing"
```

```
table(educations$Responce)
```

```
##
##          College       PhDongoing     Postgraduate UniversityDegree
##                3                1               14               12
```

```
handedness <- mono_socio %>% filter(QuestionCategory == 'handedness')
table(handedness$Responce)
```

```
##
##  left right
##     2    28
```

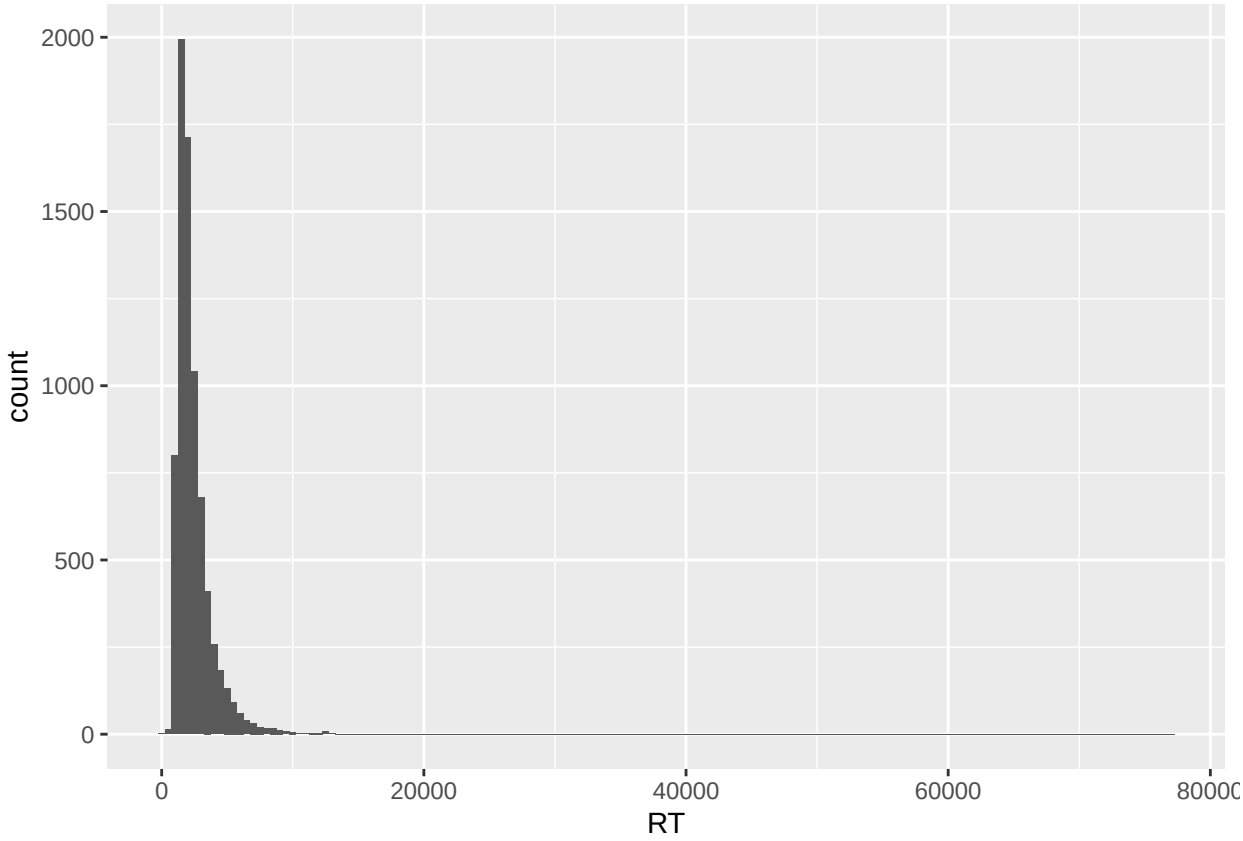
Compare you overview with that reported in Table 1 of the article. Sometimes replication data provided by authors does not allow one to reproduce their results. Let's look at another dataframe, `mono`, that contains results of experiment 1.

Answer: Data from 'mono_socio' dataframe contradicts Table 1 from the artcile. There are only 30 participants present (instead of 140 mentioned), 11 of them are male and 19 are female (vs. 66/74 reported). Also Table1 reports presence of 18 participants with only Secondary education, but in the dataframe we see no presence of respodents without a higher education degree.

1.2

Create a plot that shows the RT distribution in experiment 1 (all participants and conditions taken together). What kind of plot would you choose? Use ggplot() for this problem.

```
library(ggplot2)
c <- ggplot(mono, aes(RT))
c + geom_histogram(binwidth=500)
```



Can we say that RT approximately follows normal distribution? Which features of RT distribution contradicts this assumption? (E.g. long left tail, long right tail, outliers, skewness, etc.)

Answer: We cannot say that RT approximately follows normal distribution, because it has a long left tail.

1.3

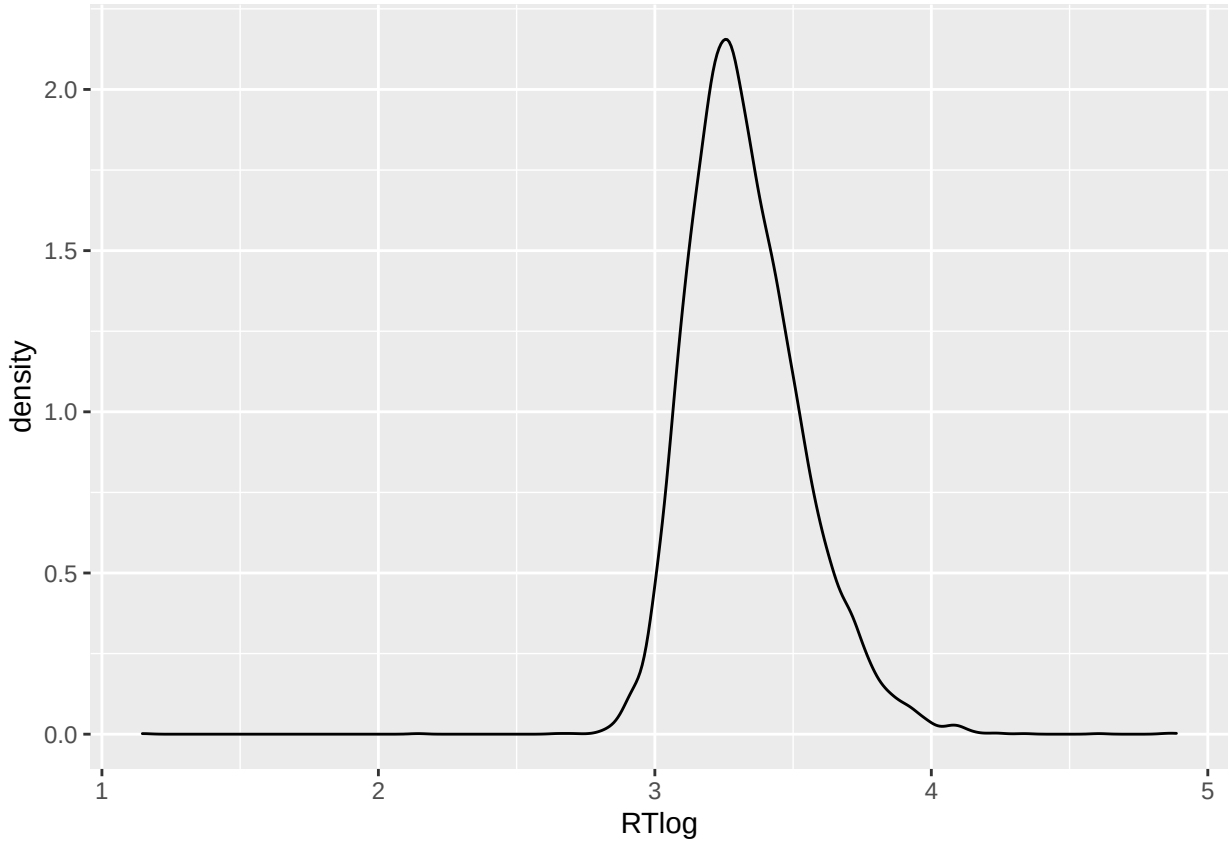
Normalise data applying the logarithm with base 10 (RTlog = log10(RT)). Use `mutate`.

```
mono <- mono %>% mutate(RTlog = log10(RT))
```

1.4

Create a density plot that shows the RTlog distribution.

```
c <- ggplot(mono, aes(RTlog))
c + geom_density(kernel='gaussian')
```



Can we say that RTlog approximately follows normal distribution? What features of RTlog distribution contradicts this assumption? (E.g. long left tail, long right tail, outliers, skewness, etc.)

Answer: Now we can say that RTlog approximately follows normal distribution, however its peak is skewed to the right.

1.5

Give a summary of `RTlog` distribution (min, max, mean, median, standard deviation)

```
# hint: sd
summary(mono$RTlog)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.146   3.185   3.302   3.329   3.449   4.886
```

```
print("Std.: ")
```

```
## [1] "Std.: "
```

```
sd(mono$RTlog)
```

```
## [1] 0.2076783
```

1.6

To filter out outliers, remove from the table the following observations: * responses RT of which is below 600 ms (i.e., when a button is pressed too fast, without allowing enough time for actual consideration of the presented stimuli)
* responses RTlog of which deviates from the mean value of RTlog for more than 3 standard deviations
* fillers (both acceptable and unacceptable)
Convert relevant variables to factors and save fitered data as `mono1`.

```
#  mono %>%
# filter here

mono1 <- mutate(mono, RTlogAbsDev = abs((RTlog-mean(RTlog))/sd(RTlog)))

mono1 <- mono1 %>% filter(RT>=600,
                          RTlogAbsDev<=3,
                          TypeOfStimuli!='FillerAcceptable',
                          TypeOfStimuli!='FillerUnacceptable')

# select(ParticipantID, TypeOfStimuli, WordOrder, AcceptabilityJ =
# ResponseAcceptabilityJudgement, RTlog) %>%
#  mutate(ParticipantID = as.factor(ParticipantID),
# do more convertion here
```

1.7

Calculate the number of observations in `mono1`.

```
dim(mono1)[1]
```

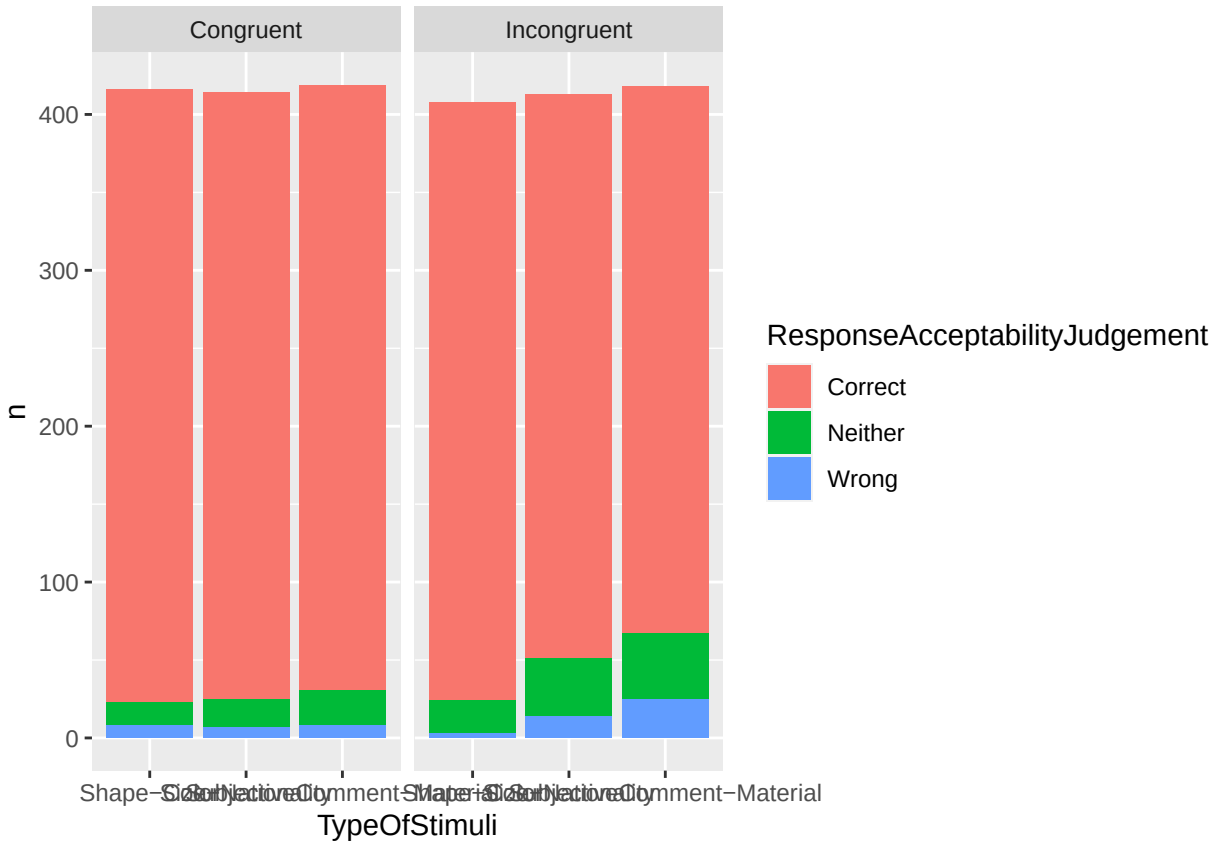
```
## [1] 2488
```

1.8

Reproduce Figure 1 from the article using `ggplot`.

Hint: You can make a summary and use `geom_col()` (see example here). Use either facet_wrap or facet_grid to make six plots. Note that we figures created in 1.8-1.10 may look different from what plotted in the article.

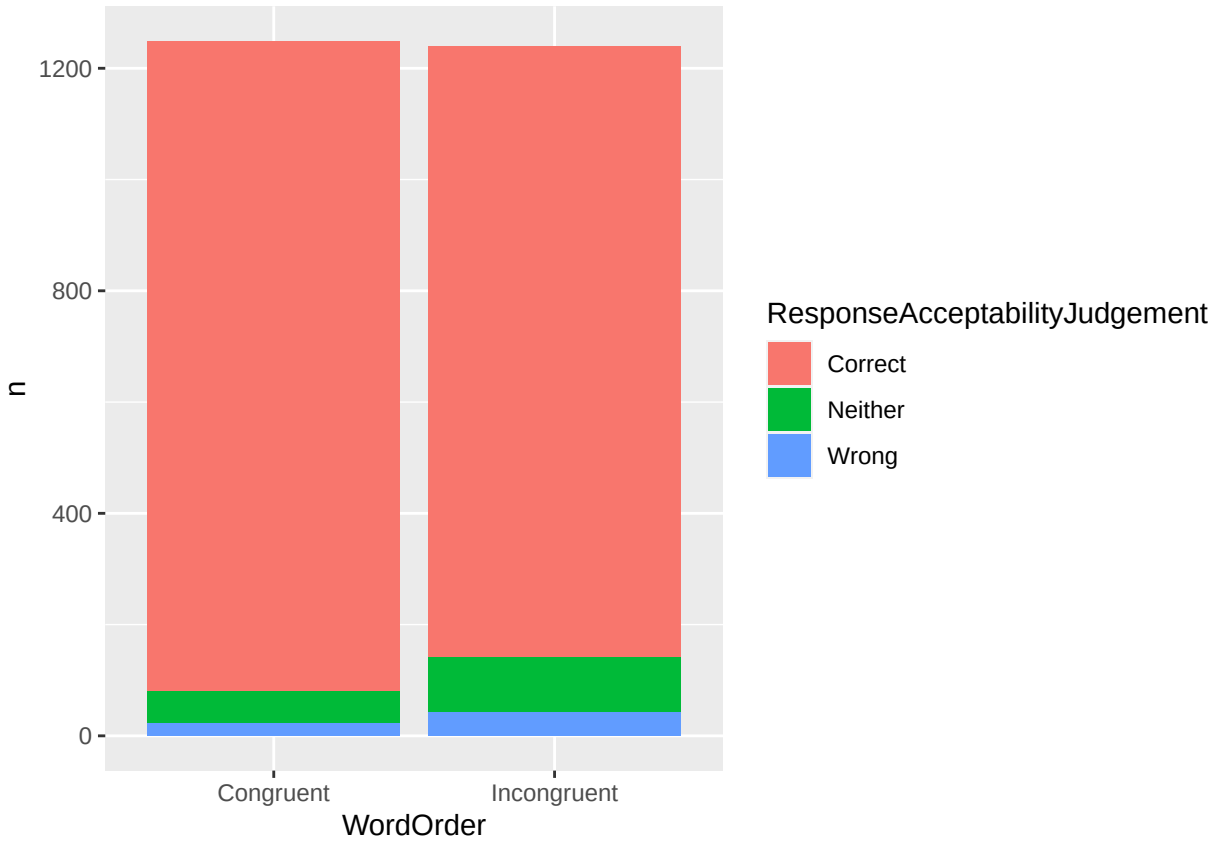
```
counts <- mono1 %>% group_by(TypeOfStimuli, WordOrder, ResponseAcceptabilityJudgement)
counts <- count(counts)
c <- ggplot(counts,
            aes(x=TypeOfStimuli, y=n,
                fill=ResponseAcceptabilityJudgement,))
c + geom_col() + facet_grid(~WordOrder)
```



1.9

Reproduce Figure 2 from the article using ggplot.

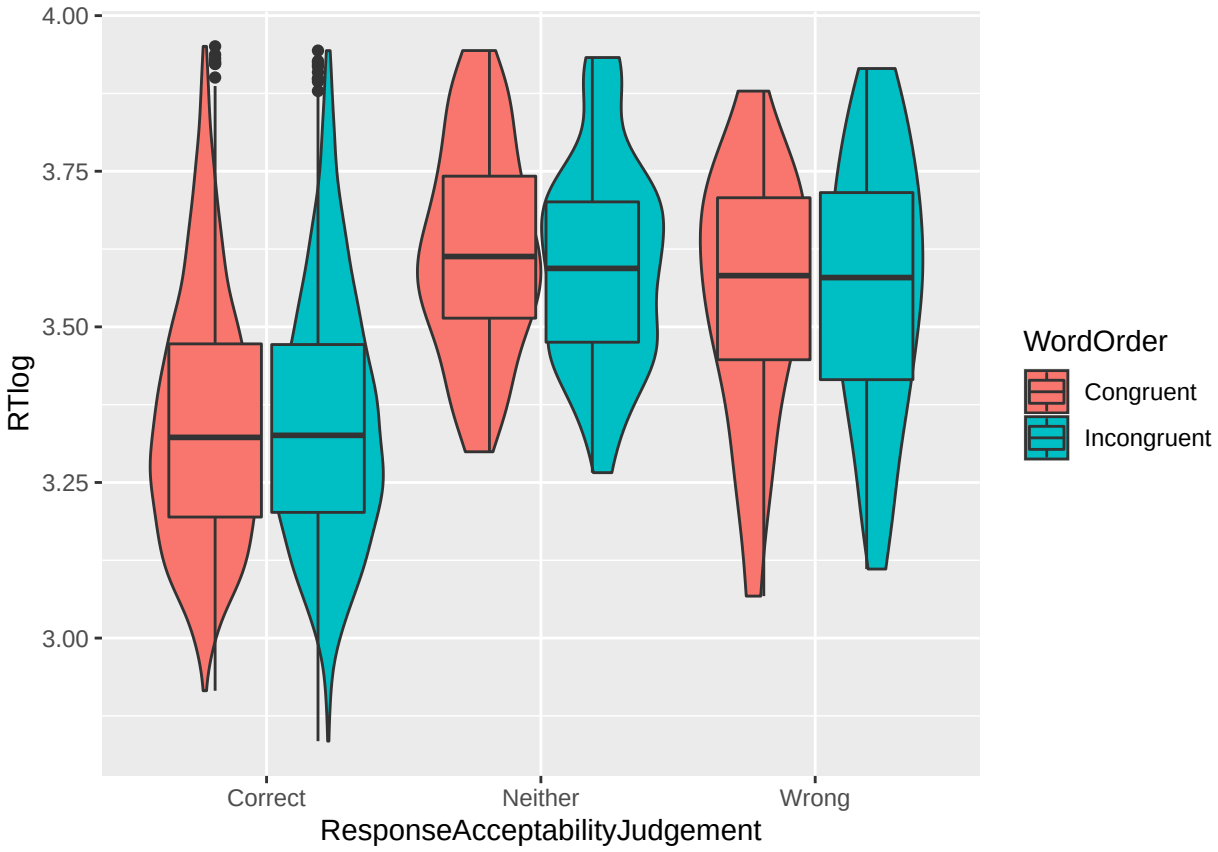
```
counts <- mono1 %>% group_by(WordOrder, ResponseAcceptabilityJudgement)
counts <- count(counts)
c <- ggplot(counts, aes(x=WordOrder, y=n, fill=ResponseAcceptabilityJudgement))
c + geom_col()
```



1.10

Reproduce Figure 7 from the article using ggplot.

```
c <- ggplot(mono1, aes(x=ResponseAcceptabilityJudgement, y=RTlog,
                       fill=WordOrder))
c + geom_violin() + geom_boxplot()
```



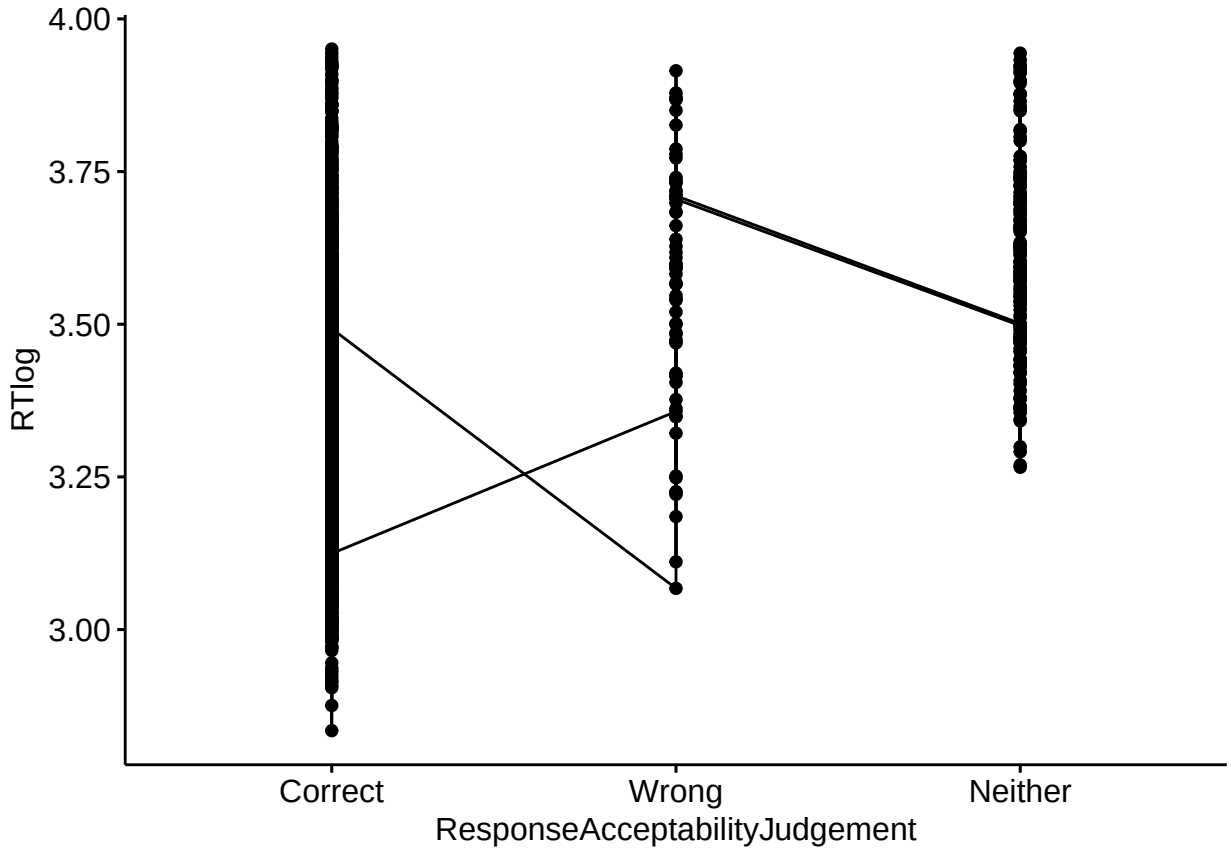
1.11

For the same data, draw a lineplot for group means and standard errors using `ggline()`:

```
install.packages("ggpubr")
```

```
## Installing package into '/home/rstudio-user/R/x86_64-pc-linux-gnu-library/4.0'
## (as 'lib' is unspecified)
```

```
library("ggpubr")
ggline(mono1, x="ResponseAcceptabilityJudgement",
       y="RTlog", group="WordOrder")
```



2. Difference in reaction time

Let us test are there any difference in the reaction time between congruent and incongruent orders. Reaction time is a numeric variable so we can use t-test to compare means. One option is to use two-sample t-test. However, as we have data for congruent and incongruent orders for *the same participants*, it is better to use *paired t-test* here. In paired t-test, for each participant, we will find difference of their reaction time in congruent and incongruent orders, and compare these differences with 0 using 1-sample t-test. To make sure that our data satisfy assumptions of t-test (values that we compare are independent samples from some approximately normal distributions), we will find mean logarithm of reaction time for each participant (across ovservations in all conditions), and consider them as our new sample.

2.1 Summarising

Use `group_by` and `summarise` to find mean logarithm of reaction time for each participant and each word order. Put this dataframe to `mean_rtlog_long` variable. It should be like

```
# A tibble: 280 x 3
   ParticipantID                    WordOrder   RTlog
   <fct>                            <fct>       <dbl>
 1 00e0b159cf5b9abcc73b92506d8b1c38 Congruent    3.24
 2 00e0b159cf5b9abcc73b92506d8b1c38 Incongruent  3.47
 3 021a49cde484f8fa18439f026ec99459 Congruent    3.22
 4 021a49cde484f8fa18439f026ec99459 Incongruent  3.21
 ...
```

```
mean_rtlog_long <- mono1 %>%
  group_by(ParticipantID, WordOrder) %>%
  summarise(RTlog=mean(RTlog))
```

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

```
mean_rtlog_long
```

```
## # A tibble: 280 x 3
## # Groups:   ParticipantID [140]
##    ParticipantID                    WordOrder   RTlog
##    <chr>                            <chr>       <dbl>
##  1 00e0b159cf5b9abcc73b92506d8b1c38 Congruent    3.24
##  2 00e0b159cf5b9abcc73b92506d8b1c38 Incongruent  3.47
##  3 021a49cde484f8fa18439f026ec99459 Congruent    3.22
##  4 021a49cde484f8fa18439f026ec99459 Incongruent  3.21
##  5 02810ff2a65eae2b3e54ac57d906309d Congruent    3.46
##  6 02810ff2a65eae2b3e54ac57d906309d Incongruent  3.36
##  7 037c50e80288a89b13f4c77e0b825789 Congruent    3.55
##  8 037c50e80288a89b13f4c77e0b825789 Incongruent  3.55
##  9 054635a48143f6d4dd711cbb00b9f6fd Congruent    3.22
## 10 054635a48143f6d4dd711cbb00b9f6fd Incongruent  3.18
## # ... with 270 more rows
```

2.2. Pivoting

Use `pivot_wider` to spread values of `RTlog` in `mean_rtlog_long` into two columns: `Congruent` and `Incongruent`. Put new dataframe in variable `mean_rtlog`. It should look like

```
# A tibble: 140 x 3
   ParticipantID                    Congruent Incongruent
   <fct>                                <dbl>       <dbl>
 1 00e0b159cf5b9abcc73b92506d8b1c38      3.24        3.47
 2 021a49cde484f8fa18439f026ec99459      3.22        3.21
 3 02810ff2a65eae2b3e54ac57d906309d      3.46        3.36
```

```
mean_rtlog <- mean_rtlog_long %>%
  pivot_wider(names_from = WordOrder,
              values_from=RTlog)
mean_rtlog
```

```
## # A tibble: 140 x 3
## # Groups:   ParticipantID [140]
##    ParticipantID                    Congruent Incongruent
##    <chr>                                <dbl>       <dbl>
##  1 00e0b159cf5b9abcc73b92506d8b1c38      3.24        3.47
##  2 021a49cde484f8fa18439f026ec99459      3.22        3.21
##  3 02810ff2a65eae2b3e54ac57d906309d      3.46        3.36
##  4 037c50e80288a89b13f4c77e0b825789      3.55        3.55
##  5 054635a48143f6d4dd711cbb00b9f6fd      3.22        3.18
##  6 066f8885734d3b91f905df8c23195a27      3.39        3.38
##  7 086805c265966ff2a53482f3b6c1328c      3.37        3.42
##  8 0912f84f2fb81492f8e7a4c5c02d33a9      3.18        3.21
##  9 09b1c9e98227331746533f60cc9b2c2d      3.33        3.43
## 10 0a14b7b4f101746c9cd29fbaa746ba5e      3.48        3.45
## # ... with 130 more rows
```

2.3. Two-sample t-test

Let us try to apply two-sample t-test to our data. Consider values in columns `Congruent` and `Incongruent` as two independent samples. Our null hypothesis is that these two samples are from populations with equal

means. Alternative hypothesis: population mean for incongruate word order is larger (people need more time to 'parse' it). Use `t.test` function to perform a test. Don't forget to specify `alternative`.

```
t.test(x=mean_rtlog$Incongruent,
       y=mean_rtlog$Congruent,
       alternative='greater')
```

```
##
##  Welch Two Sample t-test
##
## data:  mean_rtlog$Incongruent and mean_rtlog$Congruent
## t = 0.90314, df = 277.26, p-value = 0.1836
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  -0.01313405         Inf
## sample estimates:
## mean of x mean of y
##  3.377166  3.361292
```

Would you reject null hypothesis (under 5% significance level) according to this test?

Answer: The null hypothesis is not rejected as the p-value is higher than 0.05.

What claim about logarithms of reaction time for Congruent and Incongruent stimuli can you make according to this test?

Answer: There is no significant difference in logarithm of reaction time between Congruent and Incogruent stimuli.

2.4. Paired t-test: manually

To use paired t-test, let us find difference between logarithms of reaction time for each participant. Use `mutate` and add variable `diff` with aforementioned meaning to dataframe `mean_rtlog`. Save result as `mean_rtlog` again. Then compare mean of `diff` with 0 using 1-sample t-test. (Use appropriate alternative.)

```
mean_rtlog <- mean_rtlog %>%
  mutate(diff=Incongruent-Congruent)
t.test(mean_rtlog$diff, alternative = 'greater')
```

```
##
##  One Sample t-test
##
## data:  mean_rtlog$diff
## t = 2.3235, df = 139, p-value = 0.0108
## alternative hypothesis: true mean is greater than 0
## 95 percent confidence interval:
##  0.004561374         Inf
## sample estimates:
##  mean of x
## 0.01587439
```

Would you reject null hypothesis?

Answer: The null hypothesis is rejected as the p-value is less than 0.05

What claim about logarithms of reaction time for Congruent and Incongruent stimuli can you make now?

Average logarithm of reaction time for Incongruent stimuli is greater than for Congruent stimuli.

How can you interpret difference with the result of 2.3?

Answer: In 2.3 we tested the difference in means (unpaired t-test), whereas in 2.4 we tested the mean difference. By testing the difference in means we assumed that the variance between groups was equal (unpaired t-test) which was not the case - and it gave us false result.

2.5. Paired t-test out of the box In fact, we can avoid manual calculation of difference and perform paired t-test using `t.test` function with parameter `paired = True`. Apply this function to your data and make sure you get the same result as in 2.4.

```
t.test(x=mean_rtlog$Incongruent,
       y=mean_rtlog$Congruent,
       alternative='greater',
       paired=TRUE)
```

```
##
##  Paired t-test
##
## data:  mean_rtlog$Incongruent and mean_rtlog$Congruent
## t = 2.3235, df = 139, p-value = 0.0108
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  0.004561374         Inf
## sample estimates:
## mean of the differences
##              0.01587439
```

3. Difference between conditions

Now we will consider reaction time for Incongruent word ordering only. Let us check are there any statistically significant difference in logarithm of reaction time for different conditions (types of stimuli).

3.1 Data preparation

Filter only observation with `Incongruent` word order, then find average logarithm of reaction time for each participant and each type of stimuli. Save new dataframe as `incong_rtlog` variable. It should look like the following table:

```
# A tibble: 420 x 3
   ParticipantID                    TypeOfStimuli              RTlog
   <fct>                            <fct>                      <dbl>
 1 00e0b159cf5b9abcc73b92506d8b1c38 Shape-Color                 3.34
 2 00e0b159cf5b9abcc73b92506d8b1c38 Size-Nationality            3.20
 3 00e0b159cf5b9abcc73b92506d8b1c38 SubjectiveComment-Material  3.19
 4 021a49cde484f8fa18439f026ec99459 Shape-Color                 3.20
```

```
incong_rtlog <- mono1 %>%
  filter(WordOrder == "Incongruent") %>%
  group_by(ParticipantID, TypeOfStimuli) %>%
  summarise(RTlog=mean(RTlog))
```

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

```
incong_rtlog
```

```
## # A tibble: 420 x 3
## # Groups:   ParticipantID [140]
##    ParticipantID                    TypeOfStimuli              RTlog
##    <chr>                            <chr>                      <dbl>
```

```
##  1 00e0b159cf5b9abcc73b92506d8b1c38 Shape-Color                 3.49
##  2 00e0b159cf5b9abcc73b92506d8b1c38 Size-Nationality            3.47
##  3 00e0b159cf5b9abcc73b92506d8b1c38 SubjectiveComment-Material  3.44
##  4 021a49cde484f8fa18439f026ec99459 Shape-Color                 3.20
##  5 021a49cde484f8fa18439f026ec99459 Size-Nationality            3.23
##  6 021a49cde484f8fa18439f026ec99459 SubjectiveComment-Material  3.21
##  7 02810ff2a65eae2b3e54ac57d906309d Shape-Color                 3.48
##  8 02810ff2a65eae2b3e54ac57d906309d Size-Nationality            3.31
##  9 02810ff2a65eae2b3e54ac57d906309d SubjectiveComment-Material  3.29
## 10 037c50e80288a89b13f4c77e0b825789 Shape-Color                 3.67
## # ... with 410 more rows
```

3.2 Statistical testing

Use appropriate statistical test to answer the following question: are there any statistically significant difference in logarithm of reaction time for different conditions (types of stimuli)? Choose the test and provide justification for your choice. Provide your code, results and interpretation. What is your answer to the question?

Answer: Here we will use One-way ANOVA test as it is designed for finding significance of mean difference between multiple (more than 2) groups

```
aov_result <- aov(RTlog ~ TypeOfStimuli, data=incong_rtlog)
summary(aov_result)
```

```
##                Df Sum Sq Mean Sq F value Pr(>F)
## TypeOfStimuli   2  0.247 0.12370   4.431 0.0125 *
## Residuals     417 11.642 0.02792
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Answer: The p-value for one-way ANOVA is less than the significance level (0.05), so there is statistically signifficant difference in logarithm of reaction time for different conditions

3.3 Post-hoc analysis: which differences are significant?

If we compare means for several (more than two) groups and reject null hypothesis that corresponding population means are equal to each other, the next natural question is to find all pairs of groups which difference is statistically significant. As we discussed at the lecture, pairwise t-tests cannot be used here without appropriate corrections. Instead, one can use Tukey Honest Significant Differences. It reports adjusted confidence intervals for differences between group means for each pair of groups as well as p-values for null hypothesis 'difference is equal to zero'.

Apply `TukeyHSD` to the result of 3.2.

```
TukeyHSD(aov_result)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = RTlog ~ TypeOfStimuli, data = incong_rtlog)
##
## $TypeOfStimuli
##                                                   diff         lwr
## Size-Nationality-Shape-Color                -0.009997979 -0.05697105
## SubjectiveComment-Material-Shape-Color      -0.055750104 -0.10272317
## SubjectiveComment-Material-Size-Nationality -0.045752126 -0.09272519
##                                                    upr     p adj
```

```
## Size-Nationality-Shape-Color                 0.036975089 0.8710433
## SubjectiveComment-Material-Shape-Color      -0.008777036 0.0151255
## SubjectiveComment-Material-Size-Nationality  0.001220942 0.0581629
```

Interpret the results of your analysis in 3.2 and 3.3 here. Do not forget to report p-values obtained. Report which pair of conditions has statistically significant difference between logarithms of reaction time.

```
Answer: One-way ANOVA test shows that there is significant difference in logarithm of response time und
```

4. Multivariate linear regression

4.1 Using the `mono1` data, fit and compare two models that predict RTlog: * using Acceptability Judgements as predictor, and
* using Acceptability Judgements and TypeOfStimuli as predictors

```
lm1 <- lm(RTlog~ResponseAcceptabilityJudgement, data=mono1)
summary(lm1)
```

```
##
## Call:
## lm(formula = RTlog ~ ResponseAcceptabilityJudgement, data = mono1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.51017 -0.14573 -0.01805  0.12812  0.60607
##
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                        3.344590   0.004137 808.463   <2e-16 ***
## ResponseAcceptabilityJudgementNeither 0.263382   0.016304  16.154   <2e-16 ***
## ResponseAcceptabilityJudgementWrong   0.214643   0.024779   8.662   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.197 on 2485 degrees of freedom
## Multiple R-squared:  0.1156, Adjusted R-squared:  0.1149
## F-statistic: 162.4 on 2 and 2485 DF,  p-value: < 2.2e-16
```

```
lm2 <- lm(RTlog~ResponseAcceptabilityJudgement+TypeOfStimuli,
   data=mono1)
summary(lm2)
```

```
##
## Call:
## lm(formula = RTlog ~ ResponseAcceptabilityJudgement + TypeOfStimuli,
##     data = mono1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.51095 -0.14647 -0.01726  0.12509  0.61010
##
## Coefficients:
##                                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)                          3.370844   0.006867 490.881  < 2e-16
## ResponseAcceptabilityJudgementNeither  0.268881   0.016237  16.560  < 2e-16
## ResponseAcceptabilityJudgementWrong    0.224365   0.024690   9.087  < 2e-16
## TypeOfStimuliSize-Nationality         -0.025469   0.009649  -2.639  0.00835
```

```
## TypeOfStimuliSubjectiveComment-Material -0.054655   0.009647  -5.665 1.64e-08
##
## (Intercept)                              ***
## ResponseAcceptabilityJudgementNeither    ***
## ResponseAcceptabilityJudgementWrong      ***
## TypeOfStimuliSize-Nationality            **
## TypeOfStimuliSubjectiveComment-Material ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1958 on 2483 degrees of freedom
## Multiple R-squared:  0.1269, Adjusted R-squared:  0.1255
## F-statistic: 90.21 on 4 and 2483 DF,  p-value: < 2.2e-16
```

4.2 Add interaction of two predictors in the model.

```
lm3 <- lm(RTlog~ResponseAcceptabilityJudgement+TypeOfStimuli+ResponseAcceptabilityJudgement*TypeOfStimul
   data=mono1)
summary(lm3)
```

```
##
## Call:
## lm(formula = RTlog ~ ResponseAcceptabilityJudgement + TypeOfStimuli +
##     ResponseAcceptabilityJudgement * TypeOfStimuli, data = mono1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.50972 -0.14660 -0.01762  0.12376  0.60894
##
## Coefficients:
##                                                                          Estimate
## (Intercept)                                                              3.370931
## ResponseAcceptabilityJudgementNeither                                    0.252969
## ResponseAcceptabilityJudgementWrong                                      0.269874
## TypeOfStimuliSize-Nationality                                           -0.026788
## TypeOfStimuliSubjectiveComment-Material                                 -0.053584
## ResponseAcceptabilityJudgementNeither:TypeOfStimuliSize-Nationality      0.037818
## ResponseAcceptabilityJudgementWrong:TypeOfStimuliSize-Nationality       -0.054418
## ResponseAcceptabilityJudgementNeither:TypeOfStimuliSubjectiveComment-Material  0.006024
## ResponseAcceptabilityJudgementWrong:TypeOfStimuliSubjectiveComment-Material   -0.055413
##                                                                         Std. Error
## (Intercept)                                                               0.007027
## ResponseAcceptabilityJudgementNeither                                     0.033394
## ResponseAcceptabilityJudgementWrong                                       0.059476
## TypeOfStimuliSize-Nationality                                             0.010023
## TypeOfStimuliSubjectiveComment-Material                                   0.010065
## ResponseAcceptabilityJudgementNeither:TypeOfStimuliSize-Nationality       0.043172
## ResponseAcceptabilityJudgementWrong:TypeOfStimuliSize-Nationality         0.073590
## ResponseAcceptabilityJudgementNeither:TypeOfStimuliSubjectiveComment-Material   0.041921
## ResponseAcceptabilityJudgementWrong:TypeOfStimuliSubjectiveComment-Material     0.068935
##                                                                         t value
## (Intercept)                                                             479.705
## ResponseAcceptabilityJudgementNeither                                     7.575
## ResponseAcceptabilityJudgementWrong                                       4.538
## TypeOfStimuliSize-Nationality                                            -2.672
```

```
## TypeOfStimuliSubjectiveComment-Material                                    -5.324
## ResponseAcceptabilityJudgementNeither:TypeOfStimuliSize-Nationality         0.876
## ResponseAcceptabilityJudgementWrong:TypeOfStimuliSize-Nationality          -0.739
## ResponseAcceptabilityJudgementNeither:TypeOfStimuliSubjectiveComment-Material   0.144
## ResponseAcceptabilityJudgementWrong:TypeOfStimuliSubjectiveComment-Material    -0.804
##                                                                           Pr(>|t|)
## (Intercept)                                                                < 2e-16
## ResponseAcceptabilityJudgementNeither                                     5.03e-14
## ResponseAcceptabilityJudgementWrong                                       5.96e-06
## TypeOfStimuliSize-Nationality                                              0.00758
## TypeOfStimuliSubjectiveComment-Material                                   1.11e-07
## ResponseAcceptabilityJudgementNeither:TypeOfStimuliSize-Nationality        0.38113
## ResponseAcceptabilityJudgementWrong:TypeOfStimuliSize-Nationality          0.45969
## ResponseAcceptabilityJudgementNeither:TypeOfStimuliSubjectiveComment-Material  0.88574
## ResponseAcceptabilityJudgementWrong:TypeOfStimuliSubjectiveComment-Material    0.42157
##
## (Intercept)                                                               ***
## ResponseAcceptabilityJudgementNeither                                     ***
## ResponseAcceptabilityJudgementWrong                                       ***
## TypeOfStimuliSize-Nationality                                             **
## TypeOfStimuliSubjectiveComment-Material                                   ***
## ResponseAcceptabilityJudgementNeither:TypeOfStimuliSize-Nationality
## ResponseAcceptabilityJudgementWrong:TypeOfStimuliSize-Nationality
## ResponseAcceptabilityJudgementNeither:TypeOfStimuliSubjectiveComment-Material
## ResponseAcceptabilityJudgementWrong:TypeOfStimuliSubjectiveComment-Material
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1959 on 2479 degrees of freedom
## Multiple R-squared:  0.1275, Adjusted R-squared:  0.1247
## F-statistic: 45.28 on 8 and 2479 DF,  p-value: < 2.2e-16
```

4.3 Which of the models fits data the best?

```
Answer: The model fitted on two predictors (ResponseAcceptabilityJudgement and TypeOfStimuli) fits data
```

5. Binary classification

5.1 It can happen that the some parts of data are not provided by the authors. Let us assume that WordOrder is a variable one want to predict (Data: mono1). Suggest at least one type of models to predict this dependent variable. Run the code and find the minimal optimal model (model with predictors that show the statistical significance).

Let's try logistic regression:

```
y <- as.factor(mono1$WordOrder == "Congruent")
lr_model <- glm(y~TypeOfStimuli+ResponseAcceptabilityJudgement+RT+RTlog, data=mono1, family="binomial")
lr_model
```

```
##
## Call:  glm(formula = y ~ TypeOfStimuli + ResponseAcceptabilityJudgement +
##     RT + RTlog, family = "binomial", data = mono1)
##
## Coefficients:
##                              (Intercept)
##                                0.4174441
```

```
##           TypeOfStimuliSize-Nationality
##                               0.0046606
## TypeOfStimuliSubjectiveComment-Material
##                               0.0207907
##   ResponseAcceptabilityJudgementNeither
##                              -0.6455568
##     ResponseAcceptabilityJudgementWrong
##                              -0.6699239
##                                      RT
##                               0.0000165
##                                   RTlog
##                              -0.1202426
##
## Degrees of Freedom: 2487 Total (i.e. Null);  2481 Residual
## Null Deviance:       3449
## Residual Deviance: 3428  AIC: 3442
```

```
summary(lr_model)
```

```
##
## Call:
## glm(formula = y ~ TypeOfStimuli + ResponseAcceptabilityJudgement +
##     RT + RTlog, family = "binomial", data = mono1)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.222  -1.203   1.137   1.152   1.450
##
## Coefficients:
##                                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)                              4.174e-01  1.990e+00   0.210 0.833822
## TypeOfStimuliSize-Nationality            4.661e-03  9.906e-02   0.047 0.962473
## TypeOfStimuliSubjectiveComment-Material  2.079e-02  9.956e-02   0.209 0.834585
## ResponseAcceptabilityJudgementNeither   -6.456e-01  1.819e-01  -3.550 0.000385
## ResponseAcceptabilityJudgementWrong     -6.699e-01  2.678e-01  -2.502 0.012356
## RT                                       1.650e-05  9.703e-05   0.170 0.864941
## RTlog                                   -1.202e-01  6.616e-01  -0.182 0.855776
##
## (Intercept)
## TypeOfStimuliSize-Nationality
## TypeOfStimuliSubjectiveComment-Material
## ResponseAcceptabilityJudgementNeither   ***
## ResponseAcceptabilityJudgementWrong     *
## RT
## RTlog
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3449.1  on 2487  degrees of freedom
## Residual deviance: 3428.4  on 2481  degrees of freedom
## AIC: 3442.4
##
## Number of Fisher Scoring iterations: 4
```

Let's keep only statistical statistically significant predictors (which have p-values less than 0.05)

```
lr_model_min <- glm(y~ResponseAcceptabilityJudgement, data=mono1, family="binomial")
lr_model_min
```

```
##
## Call:  glm(formula = y ~ ResponseAcceptabilityJudgement, family = "binomial",
##     data = mono1)
##
## Coefficients:
##                             (Intercept)  ResponseAcceptabilityJudgementNeither
##                                 0.06442                               -0.64424
##   ResponseAcceptabilityJudgementWrong
##                               -0.66660
##
## Degrees of Freedom: 2487 Total (i.e. Null);  2485 Residual
## Null Deviance:       3449
## Residual Deviance: 3429  AIC: 3435
```

Let's see that the predictive power of minimal model is close (or even equal) to that of full model:

```
predicted <- predict(lr_model, mono1) > 0.5
mean(y == predicted)
```

```
## [1] 0.4979904
```

```
predicted <- predict(lr_model_min, mono1) > 0.5
mean(y == predicted)
```

```
## [1] 0.4979904
```

5.2 Interpret the summary of this model. Write down your conclusions.

```
summary(lr_model_min)
```

```
##
## Call:
## glm(formula = y ~ ResponseAcceptabilityJudgement, family = "binomial",
##     data = mono1)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.205  -1.205   1.150   1.150   1.441
##
## Coefficients:
##                                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)                           0.06442    0.04203   1.533 0.125293
## ResponseAcceptabilityJudgementNeither -0.64424    0.17211  -3.743 0.000182 ***
## ResponseAcceptabilityJudgementWrong   -0.66660    0.26278  -2.537 0.011190 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3449.1  on 2487  degrees of freedom
## Residual deviance: 3428.5  on 2485  degrees of freedom
## AIC: 3434.5
```

```
##
## Number of Fisher Scoring iterations: 4
```

Answer: “Correct” Acceptability responses can predict Congruent word order, whereas “Neither” or “Wrong” Acceptability responses can predict Incongruent word order (as they have negative coefficients).