RLDM Code S2229889

Sanne van Dijk

2024-07-02

Inspecting the raw data

Overview of rawdata

```
skimr::skim(rawdata)
```

Table 1: Data summary

Name	rawdata
Number of rows	9600
Number of columns	4
Column type frequency:	
numeric	4
Group variables	None

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
ID	0	1	6.50	3.45	1.00	3.75	6.50	9.25	12.0	
condition	0	1	1.50	0.50	1.00	1.00	1.50	2.00	2.0	
correct	0	1	0.95	0.22	0.00	1.00	1.00	1.00	1.0	
rt	0	1	694.66	217.96	384.48	567.78	658.11	768.87	6374.9	

```
summary(rawdata)
```

```
##        ID          condition      correct            rt        
##  Min.   : 1.00   Min.   :1.0   Min.   :0.0000   Min.   : 384.5  
##  1st Qu.: 3.75   1st Qu.:1.0   1st Qu.:1.0000   1st Qu.: 567.8  
##  Median : 6.50   Median :1.5   Median :1.0000   Median : 658.1  
##  Mean   : 6.50   Mean   :1.5   Mean   :0.9494   Mean   : 694.7  
##  3rd Qu.: 9.25   3rd Qu.:2.0   3rd Qu.:1.0000   3rd Qu.: 768.9  
##  Max.   :12.00   Max.   :2.0   Max.   :1.0000   Max.   :6374.9  
```

Sample size

```
describe(rawdata)
```

```
##           vars    n   mean     sd median trimmed    mad    min    max   range
## ID           1 9600   6.50   3.45   6.50    6.50   4.45   1.00   12.0   11.00
## condition    2 9600   1.50   0.50   1.50    1.50   0.74   1.00    2.0    1.00
## correct      3 9600   0.95   0.22   1.00    1.00   0.00   0.00    1.0    1.00
## rt           4 9600 694.66 217.96 658.11  668.95 144.59 384.48 6374.9 5990.42
##            skew kurtosis   se
## ID         0.00    -1.22 0.04
## condition  0.00    -2.00 0.01
## correct   -4.10    14.80 0.00
## rt         5.55    83.01 2.22
```

```
N <- length(unique(rawdata$ID))
N # N = 12, and there is a total of 9600 observations
```

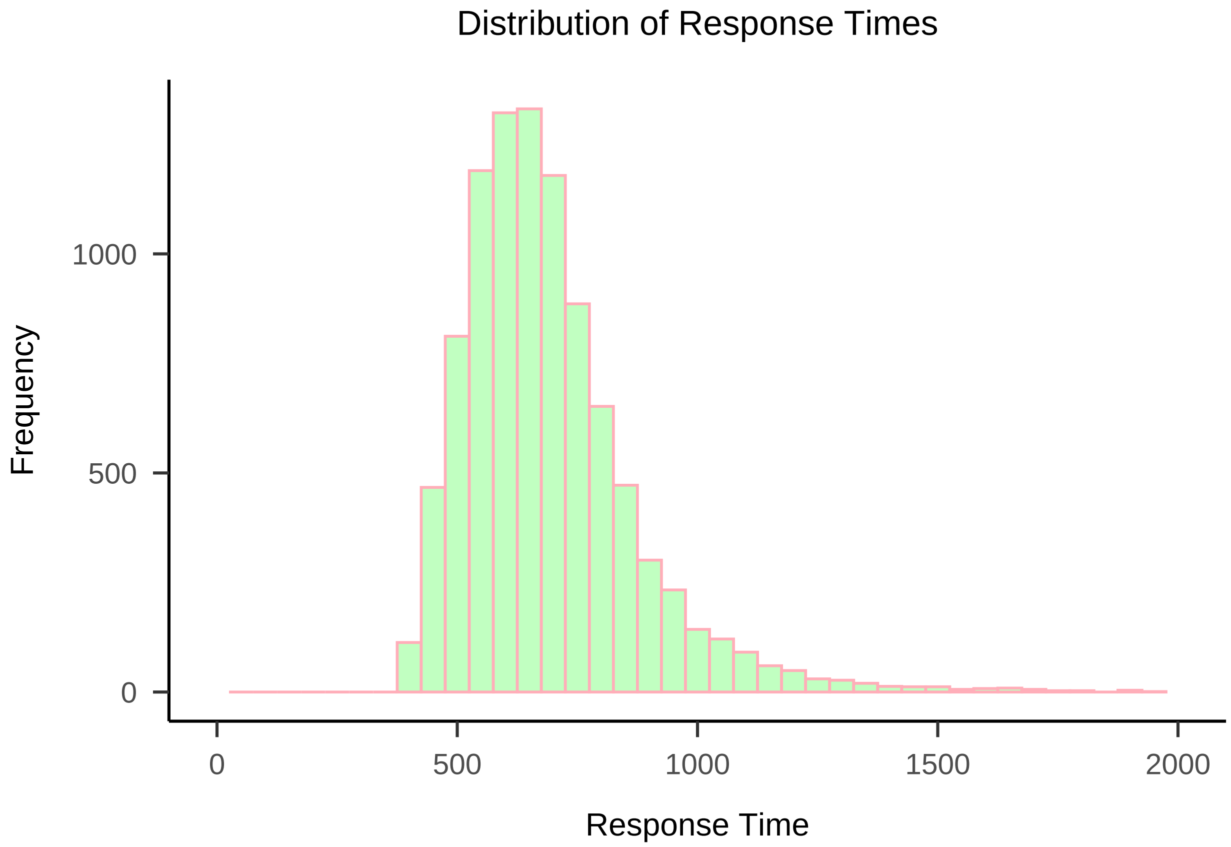
```
## [1] 12
```

Visual inspection

```
# For ggplot, the condition and correct columns should be coded as factors
rawdata_ggplot <- as.data.frame(dataset4) # In order to not get confused later

# Change condition and correct into factors
rawdata_ggplot$condition <- as.factor(rawdata_ggplot$condition)
rawdata_ggplot$correct <- as.factor(rawdata_ggplot$correct)

# Histogram showing the distribution of response times
rt_hist <- ggplot(rawdata_ggplot, aes(x = rt)) +
  geom_histogram(binwidth = 50, fill = color1, color=color2) +
  labs(title = "Distribution of Response Times",
       x = "Response Time",
       y = "Frequency") +
  theme_apa() +
  xlim(0,2000)
print(rt_hist)
```

```
## Warning: Removed 23 rows containing non-finite outside the scale range
## (`stat_bin()`).
```

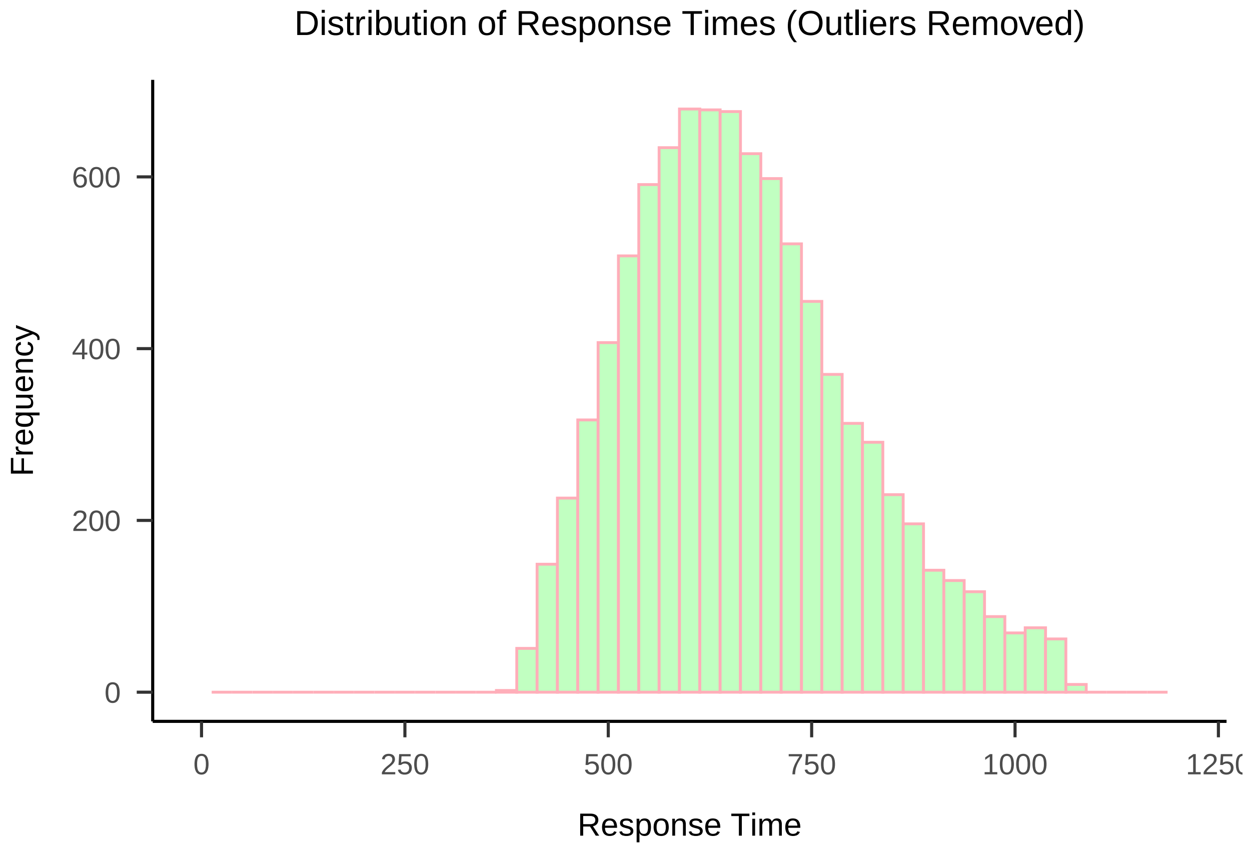
```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_bar()`).
```



Remove extreme variables

```
# Remove outliers using IQR
  # Calculate Q1 (25th percentile) and Q3 (75th percentile)
Q1 <- quantile(rawdata$rt, 0.25, na.rm = TRUE)
Q3 <- quantile(rawdata$rt, 0.75, na.rm = TRUE)

  # Calculate IQR
IQR <- Q3 - Q1

 # Determine bounds for outliers
lower_bound <- Q1 - 1.5 * IQR
upper_bound <- Q3 + 1.5 * IQR

  # Make rawdata numeric
num_rawdata <- as.data.frame(rawdata)
num_rawdata$correct <- as.numeric(num_rawdata$correct, levels = c("1", "0"))

    # Filter the data to remove outliers
Datanooutl <- num_rawdata[num_rawdata$rt >= lower_bound & num_rawdata$rt <= upper_bound, ]

  # Show histogram of the cleaned distribution
DataNO_hist <- ggplot(Datanooutl, aes(x = rt)) +
  geom_histogram(binwidth = 25, fill = color1, color=color2) +
  labs(title = "Distribution of Response Times (Outliers Removed)",
       x = "Response Time",
       y = "Frequency") +
  theme_apa() +
  xlim(0,1200)
print(DataNO_hist)
```

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_bar()`).
```



### Outlier inspection

```
outliers <- anti_join(num_rawdata,Datanooutl)
```

```
## Joining with `by = join_by(ID, condition, correct, rt)`
```

```
# Overview
skimr::skim(outliers)
```

Table 3: Data summary

Name	outliers
Number of rows	388

Number of columns	4
Column type frequency:	
numeric	4
Group variables	None

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
ID	0	1	6.35	3.49	1.00	3.00	6.00	9.00	12.0	
condition	0	1	1.34	0.47	1.00	1.00	1.00	2.00	2.0	
correct	0	1	0.87	0.33	0.00	1.00	1.00	1.00	1.0	
rt	0	1	1365.10	506.76	1071.31	1123.43	1202.52	1386.36	6374.9	

```
summary(outliers)
```

```
##        ID           condition        correct             rt      
##  Min.   : 1.000   Min.   :1.000   Min.   :0.0000   Min.   :1071  
##  1st Qu.: 3.000   1st Qu.:1.000   1st Qu.:1.0000   1st Qu.:1123  
##  Median : 6.000   Median :1.000   Median :1.0000   Median :1203  
##  Mean   : 6.353   Mean   :1.338   Mean   :0.8737   Mean   :1365  
##  3rd Qu.: 9.000   3rd Qu.:2.000   3rd Qu.:1.0000   3rd Qu.:1386  
##  Max.   :12.000   Max.   :2.000   Max.   :1.0000   Max.   :6375  
```

```
# Descriptives per participant
outlierspp <- matrix(nrow = 4, ncol = 12,
                     dimnames = list(c("N", "M rt", "MD rt", "M accuracy"), 1:12))

for (i in 1:12) {
  outlierspp[1,i] <- sum(outliers$ID == i)
  outlierspp[2,i] <- mean(outliers$rt[outliers$ID == i])
  outlierspp[3,i] <- median(outliers$rt[outliers$ID == i])
  outlierspp[4,i] <- mean(outliers$correct[outliers$ID == i])
}
outlierspp
```

```
##                       1            2            3        4            5
## N            38.0000000   33.0000000   37.0000000   30.000   24.0000000
## M rt       1345.2584303 1507.3912307 1283.2423594 1443.240 1377.4705976
## MD rt      1180.4948976 1199.5001225 1153.4337985 1215.207 1155.9505094
## M accuracy    0.8947368    0.8787879    0.8378378    0.800    0.8333333
##                       6          7            8          9           10
## N            38.0000000   32.00000   31.0000000   32.00000   30.0000000
## M rt       1463.6195101 1338.40379 1297.4553610 1341.95982 1313.8247436
## MD rt      1234.4407687 1251.65237 1245.0619232 1187.10940 1210.7108449
## M accuracy    0.9736842    0.84375    0.9354839    0.90625    0.8666667
##                      11          12
## N            34.0000000   29.000000
## M rt       1245.7302359 1433.700965
## MD rt      1171.8232752 1271.290473
## M accuracy    0.8235294    0.862069
```

```
# Table of participant's descriptives after removing outliers
  # Group data by participant (ID) and calculate descriptives
descrip_perpart <- Datanooutl %>%
  group_by(ID) %>%
  summarise(
    mean_rt = mean(rt, na.rm = TRUE),
    sd_rt = sd(rt, na.rm = TRUE),
    median_rt = median(rt, na.rm = TRUE),
    min_rt = min(rt, na.rm = TRUE),
    max_rt = max(rt, na.rm = TRUE),
    mean_accuracy = mean(correct, na.rm = TRUE),
    trials = n()  # Count the number of trials per participant
  )
  # Print the table of descriptives
print(descrip_perpart)
```

```
## # A tibble: 12 x 8
##       ID mean_rt sd_rt median_rt min_rt max_rt mean_accuracy trials
##    <int>   <dbl> <dbl>     <dbl>  <dbl>  <dbl>         <dbl>  <int>
##  1     1    671.  144.      660.   403.  1069.         0.954    762
##  2     2    662.  133.      648.   393.  1055.         0.962    767
##  3     3    673.  147.      652.   384.  1065.         0.949    763
##  4     4    673.  138.      660.   396.  1061.         0.958    770
##  5     5    659.  133.      644.   400.  1070.         0.950    776
##  6     6    662.  137.      644.   402.  1065.         0.949    762
##  7     7    671.  139.      655.   407.  1069.         0.944    768
##  8     8    666.  137.      652.   389.  1070.         0.956    769
##  9     9    670.  139.      654.   392.  1046.         0.948    768
## 10    10    666.  133.      652.   396.  1061.         0.956    770
## 11    11    660.  137.      642.   385.  1066.         0.958    766
## 12    12    662.  136.      642.   394.  1052.         0.947    771
```

Data Inspection Cleaned Dataset

```
summary(Datanooutl)
```

```
##        ID           condition        correct             rt        
##  Min.   : 1.000   Min.   :1.000   Min.   :0.0000   Min.   : 384.5  
##  1st Qu.: 4.000   1st Qu.:1.000   1st Qu.:1.0000   1st Qu.: 564.5  
##  Median : 7.000   Median :2.000   Median :1.0000   Median : 651.1  
##  Mean   : 6.506   Mean   :1.507   Mean   :0.9526   Mean   : 666.4  
##  3rd Qu.:10.000   3rd Qu.:2.000   3rd Qu.:1.0000   3rd Qu.: 750.7  
##  Max.   :12.000   Max.   :2.000   Max.   :1.0000   Max.   :1069.7  
```

```
describe(Datanooutl)
```

```
##           vars    n   mean     sd median trimmed   mad    min     max  range
## ID            1 9212   6.51   3.45   7.00    6.51   4.45   1.00   12.00  11.00
## condition     2 9212   1.51   0.50   2.00    1.51   0.00   1.00    2.00   1.00
## correct       3 9212   0.95   0.21   1.00    1.00   0.00   0.00    1.00   1.00
## rt            4 9212 666.42 137.79 651.12  657.99 136.30 384.48 1069.75 685.27
##            skew kurtosis   se
## ID          0.00    -1.22 0.04
## condition -0.03    -2.00 0.01
## correct   -4.26    16.13 0.00
## rt          0.54    -0.11 1.44
```

```
skimr::skim(Datanooutl)
```

Table 5: Data summary

Name	Datanooutl
Number of rows	9212
Number of columns	4
Column type frequency:	
numeric	4
Group variables	None

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
ID	0	1	6.51	3.45	1.00	4.00	7.00	10.00	12.00	
condition	0	1	1.51	0.50	1.00	1.00	2.00	2.00	2.00	
correct	0	1	0.95	0.21	0.00	1.00	1.00	1.00	1.00	
rt	0	1	666.42	137.79	384.48	564.51	651.12	750.65	1069.75	

Descriptives per participant

```
dataWOdesc <- matrix(nrow = 4, ncol = 12,
                     dimnames = list(c("N", "M rt", "MD rt", "M accuracy"), 1:12))

for (i in 1:12) {
  dataWOdesc[1,i] <- sum(Datanooutl$ID == i)
  dataWOdesc[2,i] <- mean(Datanooutl$rt[Datanooutl$ID == i])
  dataWOdesc[3,i] <- median(Datanooutl$rt[Datanooutl$ID == i])
  dataWOdesc[4,i] <- mean(Datanooutl$correct[Datanooutl$ID == i])
}
dataWOdesc
```

```
##                      1           2          3           4           5
## N          762.0000000 767.0000000 763.000000 770.0000000 776.0000000
## M rt       671.4582259 662.2119375 673.295254 673.3971590 659.1154963
## MD rt      659.7190479 647.5310680 652.488828 659.7024003 643.7369905
## M accuracy   0.9540682   0.9621904   0.948886   0.9584416   0.9497423
##                    6           7           8           9          10
## N          762.0000000 768.0000000 769.0000000 768.0000000 770.0000000
## M rt       661.8461381 670.8367310 666.1880566 670.2817032 666.3345478
## MD rt      644.3338561 654.9488365 652.2071608 653.7054790 651.9766268
## M accuracy   0.9488189   0.9440104   0.9557867   0.9479167   0.9558442
##                   11          12
## N          766.0000000 771.0000000
## M rt       659.8506609 662.3729455
## MD rt      641.7510358 641.7037902
## M accuracy   0.9582245   0.9468223
```

Descriptives per conditions x corrections

```
conditions <- list(
  list(name = "correct_0", filter = quo(correct == 0)),
  list(name = "correct_1", filter = quo(correct == 1)),
  list(name = "condition_1", filter = quo(condition == 1)),
  list(name = "condition_2", filter = quo(condition == 2)),
  list(name = "condition_1_correct_0", filter = quo(condition == 1
                                                    & correct == 0)),
  list(name = "condition_1_correct_1", filter = quo(condition == 1
                                                    & correct == 1)),
  list(name = "condition_2_correct_0", filter = quo(condition == 2
                                                    & correct == 0)),
  list(name = "condition_2_correct_1", filter = quo(condition == 2
                                                    & correct == 1))
)

summarycond <- list()
for (cond in conditions) {
  filtered_data <- Datanooutl %>% filter(!!cond$filter)
  summarycond[[cond$name]] <- summary(filtered_data)
}
summarycond
```

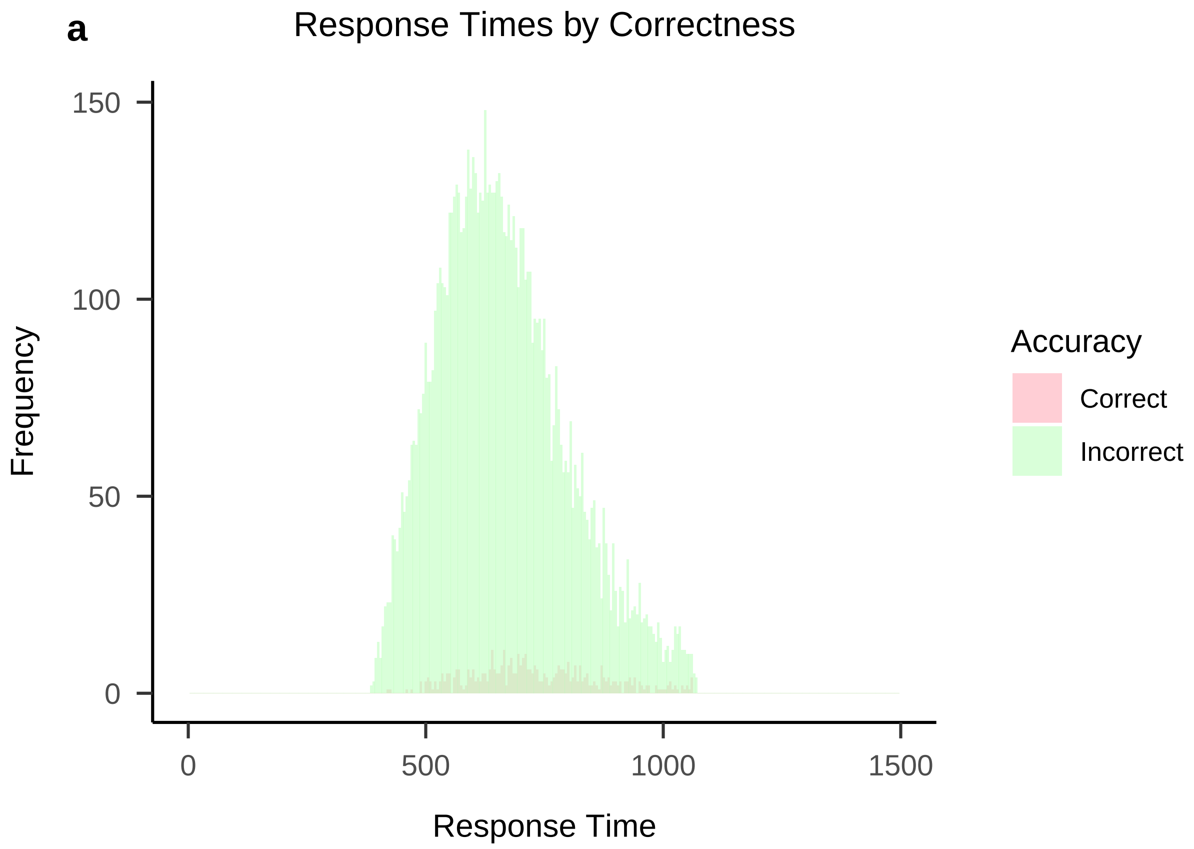
```
## $correct_0
##        ID         condition        correct        rt
##  Min.   : 1.0   Min.   :1.000   Min.   :0   Min.   : 422.3
##  1st Qu.: 4.0   1st Qu.:1.000   1st Qu.:0   1st Qu.: 637.5
##  Median : 7.0   Median :2.000   Median :0   Median : 712.9
##  Mean   : 6.6   Mean   :1.643   Mean   :0   Mean   : 732.9
##  3rd Qu.: 9.0   3rd Qu.:2.000   3rd Qu.:0   3rd Qu.: 822.8
##  Max.   :12.0   Max.   :2.000   Max.   :0   Max.   :1061.5
##
## $correct_1
##        ID           condition      correct        rt
##  Min.   : 1.000   Min.   :1.0   Min.   :1   Min.   : 384.5
##  1st Qu.: 4.000   1st Qu.:1.0   1st Qu.:1   1st Qu.: 562.2
##  Median : 7.000   Median :2.0   Median :1   Median : 647.5
##  Mean   : 6.502   Mean   :1.5   Mean   :1   Mean   : 663.1
##  3rd Qu.:10.000   3rd Qu.:2.0   3rd Qu.:1   3rd Qu.: 746.5
##  Max.   :12.000   Max.   :2.0   Max.   :1   Max.   :1069.7
```

```
##
## $condition_1
##        ID           condition    correct             rt
##  Min.   : 1.000   Min.   :1   Min.   :0.0000   Min.   : 445.3
##  1st Qu.: 4.000   1st Qu.:1   1st Qu.:1.0000   1st Qu.: 598.4
##  Median : 7.000   Median :1   Median :1.0000   Median : 677.6
##  Mean   : 6.516   Mean   :1   Mean   :0.9657   Mean   : 695.4
##  3rd Qu.:10.000   3rd Qu.:1   3rd Qu.:1.0000   3rd Qu.: 773.4
##  Max.   :12.000   Max.   :1   Max.   :1.0000   Max.   :1069.7
##
## $condition_2
##        ID           condition    correct             rt
##  Min.   : 1.000   Min.   :2   Min.   :0.0000   Min.   : 384.5
##  1st Qu.: 4.000   1st Qu.:2   1st Qu.:1.0000   1st Qu.: 527.4
##  Median : 6.000   Median :2   Median :1.0000   Median : 620.6
##  Mean   : 6.497   Mean   :2   Mean   :0.9398   Mean   : 638.2
##  3rd Qu.: 9.000   3rd Qu.:2   3rd Qu.:1.0000   3rd Qu.: 723.6
##  Max.   :12.000   Max.   :2   Max.   :1.0000   Max.   :1069.5
##
## $condition_1_correct_0
##        ID           condition    correct        rt
##  Min.   : 1.000   Min.   :1   Min.   :0   Min.   : 535.1
##  1st Qu.: 4.000   1st Qu.:1   1st Qu.:0   1st Qu.: 704.6
##  Median : 6.500   Median :1   Median :0   Median : 785.6
##  Mean   : 6.808   Mean   :1   Mean   :0   Mean   : 795.6
##  3rd Qu.:10.000   3rd Qu.:1   3rd Qu.:0   3rd Qu.: 877.0
##  Max.   :12.000   Max.   :1   Max.   :0   Max.   :1060.0
##
## $condition_1_correct_1
##        ID           condition    correct        rt
##  Min.   : 1.000   Min.   :1   Min.   :1   Min.   : 445.3
##  1st Qu.: 4.000   1st Qu.:1   1st Qu.:1   1st Qu.: 595.8
##  Median : 7.000   Median :1   Median :1   Median : 673.9
##  Mean   : 6.506   Mean   :1   Mean   :1   Mean   : 691.9
##  3rd Qu.:10.000   3rd Qu.:1   3rd Qu.:1   3rd Qu.: 765.6
##  Max.   :12.000   Max.   :1   Max.   :1   Max.   :1069.7
##
## $condition_2_correct_0
##        ID           condition    correct        rt
##  Min.   : 1.000   Min.   :2   Min.   :0   Min.   : 422.3
##  1st Qu.: 3.000   1st Qu.:2   1st Qu.:0   1st Qu.: 598.8
##  Median : 7.000   Median :2   Median :0   Median : 677.8
##  Mean   : 6.484   Mean   :2   Mean   :0   Mean   : 698.1
##  3rd Qu.: 9.000   3rd Qu.:2   3rd Qu.:0   3rd Qu.: 790.5
##  Max.   :12.000   Max.   :2   Max.   :0   Max.   :1061.5
##
## $condition_2_correct_1
##        ID           condition    correct        rt
##  Min.   : 1.000   Min.   :2   Min.   :1   Min.   : 384.5
##  1st Qu.: 4.000   1st Qu.:2   1st Qu.:1   1st Qu.: 523.5
##  Median : 6.000   Median :2   Median :1   Median : 615.9
##  Mean   : 6.497   Mean   :2   Mean   :1   Mean   : 634.4
##  3rd Qu.:10.000   3rd Qu.:2   3rd Qu.:1   3rd Qu.: 720.4
##  Max.   :12.000   Max.   :2   Max.   :1   Max.   :1069.5
```

RT and accuracy differences between conditions

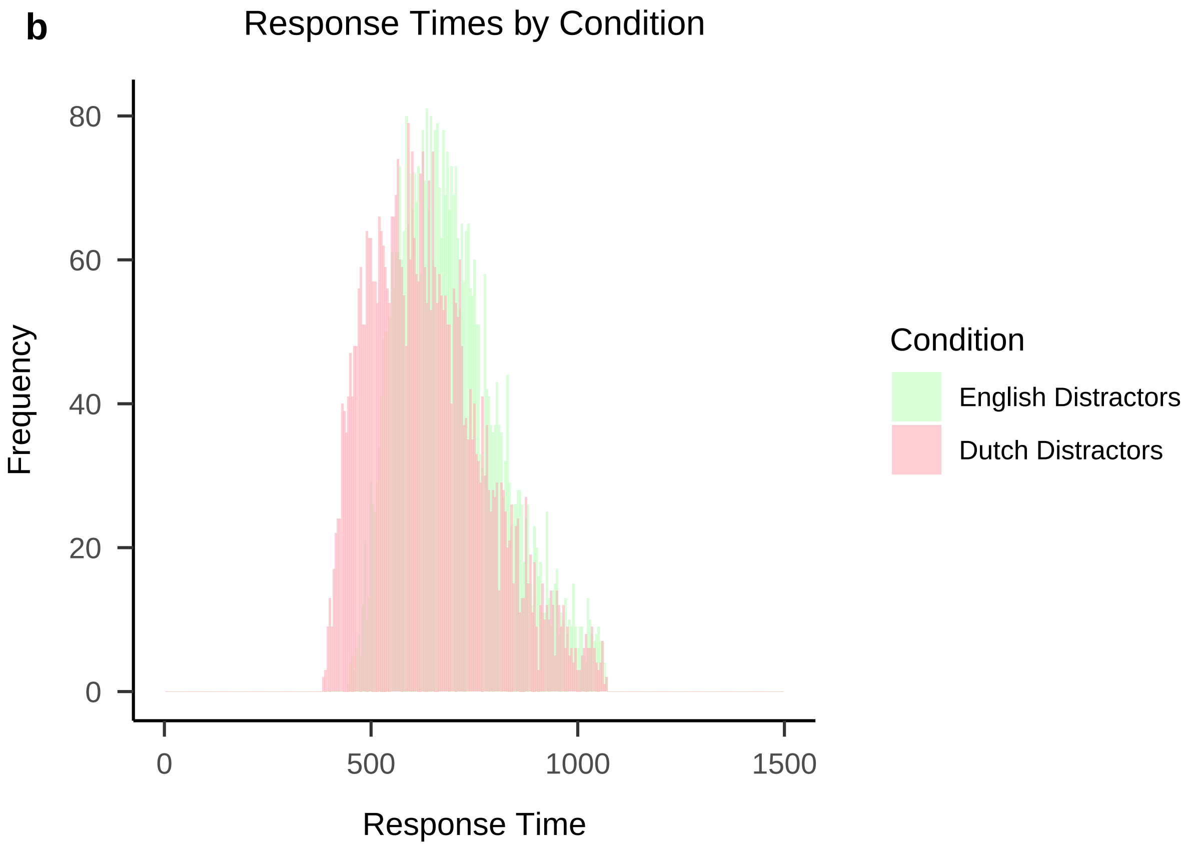
```r
## Histogram showing RTs for both correct and incorrect as overlaying distributions
histCorrect <- ggplot(Datanooutl, aes(x = rt, group = factor(correct),
                                  fill = factor(correct))) +
  geom_histogram(binwidth = 5, alpha = .6, position = "identity") +
  papaja::theme_apa() +
  labs(title = "Response Times by Correctness",
       x = "Response Time",
       y = "Frequency",
       fill = "Accuracy") +
  scale_fill_manual(values = mycolors2,
                    labels = c("Correct", "Incorrect")) +
  xlim(c(0,1500)) +
  annotate("text", x = 50, y = Inf, label = "a", size = 5,
           fontface = "bold", vjust = -1.5, hjust = 7) +
  coord_cartesian(clip = "off")
histCorrect
```

```
## Warning: Removed 4 rows containing missing values or values outside the scale range
## (`geom_bar()`).
```



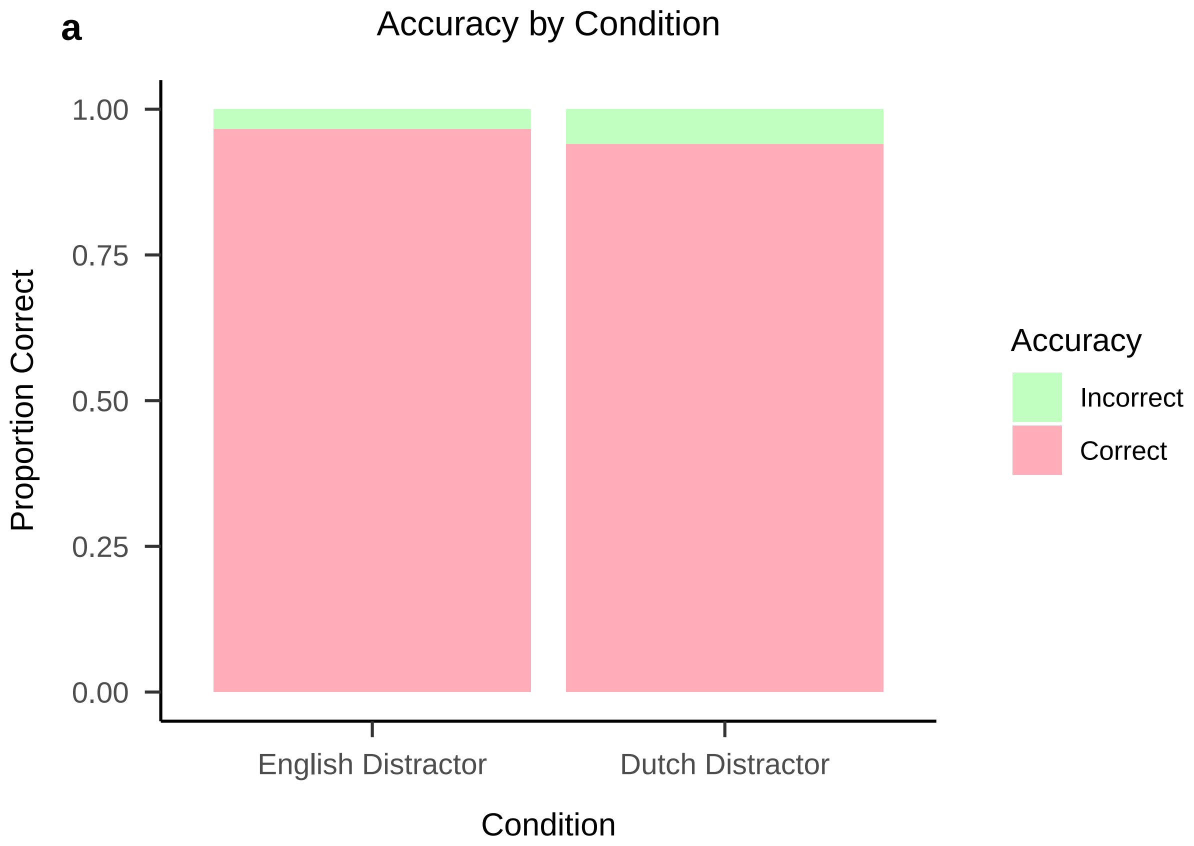
```r
## Histogram showing RTs for both condition 1 and condition 2 as overlaying distributions
histCond <- ggplot(Datanooutl, aes(x = rt, group = factor(condition),
                               fill = factor(condition)) )+
  geom_histogram(binwidth = 5, alpha = .6, position = "identity") +
  papaja::theme_apa() +
  labs(title = "Response Times by Condition",
       x = "Response Time",
       y = "Frequency",
       fill = "Condition") +
  scale_fill_manual(values = mycolors1,
                    labels = c("English Distractors", "Dutch Distractors")) +
  xlim(c(0,1500)) +
  annotate("text", x = 50, y = Inf, label = "b", size = 5,
           fontface = "bold", vjust = -1.5, hjust = 7)+
  coord_cartesian(clip = "off")
histCond
```

```
## Warning: Removed 4 rows containing missing values or values outside the scale range
## (`geom_bar()`).
```



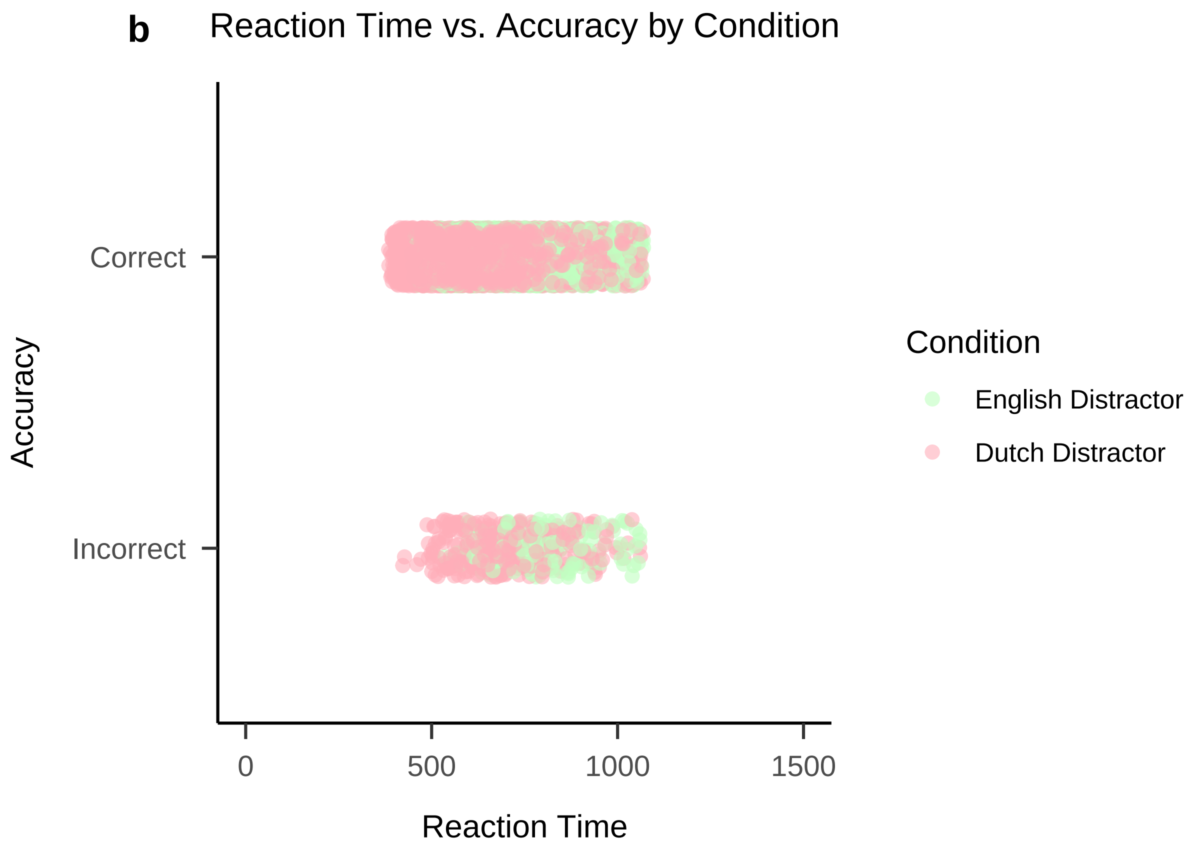
### Plots of accuracy by condition

```r
# barplot of accuracy by condition
bar_acc <- ggplot(Datanooutl, aes(x = factor(condition), fill = factor(correct))) +
  geom_bar(position = "fill") +
  papaja::theme_apa() +
  labs(title = "Accuracy by Condition",
       x = "Condition",
       y = "Proportion Correct",
       fill = "Accuracy") +
  scale_x_discrete(labels = c("1" = "English Distractor", "2" = "Dutch Distractor")) +
  scale_fill_manual(values = mycolors1,
                    labels = c("Incorrect", "Correct")) +
  annotate("text", x = 1, y = Inf, label = "a", size = 5,
           fontface = "bold", vjust = -1.5, hjust = 15)+
  coord_cartesian(xlim = c(1,2), clip = "off")
bar_acc
```



```
# dotplot of RT vs accuracy
dot_RT_Acc <- ggplot(Datanooutl, aes(x = rt, y = factor(correct), color = factor(condition))) +
  geom_point(position = position_jitter(height = 0.1), size = 2, alpha = 0.6) +
  scale_color_manual(name = "Condition", values = mycolors1,
                     labels = c("English Distractor", "Dutch Distractor")) +
  papaja::theme_apa() +
  labs(x = "Reaction Time", y = "Accuracy",
       title = "Reaction Time vs. Accuracy by Condition") +
  scale_y_discrete(breaks = c(0, 1), labels = c("Incorrect", "Correct")) +
  xlim(c(0, 1500)) +
  annotate("text", x = 50, y = Inf, label = "b", size = 5, fontface = "bold",
           vjust = -1.5, hjust = 6)+
  coord_cartesian(clip = "off")
```

```
dot_RT_Acc
```



Paired t-test over RT data by condition

```
  # aggregate data by subject, using the median (due to skewed distribution of response times!)
agg <- aggregate(Datanooutl, by = list(Datanooutl$ID, Datanooutl$condition), FUN = median)
agg = subset(agg, select = c(ID, condition, correct, rt))

  # Change agg into wide format and perform a t-test
agg_wide <- agg %>% pivot_wider(names_from = condition, values_from = rt, names_prefix = 'cond_')
res <- t.test(agg_wide$cond_1, agg_wide$cond_2, paired = TRUE)
print(res)
```

```
##
##  Paired t-test
##
## data:  agg_wide$cond_1 and agg_wide$cond_2
## t = 19.753, df = 11, p-value = 6.105e-10
## alternative hypothesis: true mean difference is not equal to 0
## 95 percent confidence interval:
##  50.58915 63.27643
## sample estimates:
## mean difference
##        56.93279
```

```
# Differences of mean accuracy between conditions
  # Aggregate data by subject and condition to calculate the mean accuracy
agg_acc <- Datanooutl %>%
  group_by(ID, condition) %>%
  summarize(mean_acc = mean(correct, na.rm = TRUE), .groups = 'drop')

  # Change agg_acc into wide format for the t-test
agg_acc_wide <- agg_acc %>%
  pivot_wider(names_from = condition, values_from = mean_acc, names_prefix = 'cond_')

  # Perform the paired t-test on mean accuracy
res_acc <- t.test(agg_acc_wide$cond_1, agg_acc_wide$cond_2, paired = TRUE)
print(res_acc)
```

```
##
##  Paired t-test
##
## data:  agg_acc_wide$cond_1 and agg_acc_wide$cond_2
## t = 5.7108, df = 11, p-value = 0.000136
## alternative hypothesis: true mean difference is not equal to 0
## 95 percent confidence interval:
##  0.01588784 0.03581439
## sample estimates:
## mean difference
##      0.02585111
```

Modelfitting

Modelfit for each participant in each condition

```
# Fit model for each participant x condition
modelfit <- data.frame(ID = numeric(),
                       condition = numeric(),
                       s = numeric(),
                       A = numeric(),
                       ter = numeric(),
                       b = numeric(),
                       v1 = numeric())
  # Extract parameters and code them into final data frame per subject and condition loop
for (participant in 1:N) {
  for (condition in 1:2) {
    parameters <- fit_data(Datanooutl[Datanooutl$ID == participant & Datanooutl$condition == condition,
    modelfit[nrow(modelfit) + 1, ] <- unlist(c(participant, condition, parameters))
  }
}
```

Paramaters of our DDM from the fitting framework provided in the course: 's' = *The standard deviation of drift rates*: The variability in drift rates. This parameter does however not an easily interpretable cognitive mapping and therefore does not show significant differences between conditions in the provided datasets. 'A' = *The upper limit of the starting point*: The starting point of the evidence accumulation process, it reflects bias 'ter' = *Non-decision time*: The portion of RT occuring independently of the decision-making process. 'b' = *Threshold*: The distance from 0 to the threshold, reflects cautiousness (speed accuract trade-off). Lower thresholds cause faster responses, but reduce accuracy. 'v1' = *Drift rate*: The difficulty level of the problem, higher drift rates cause faster decision-making (easy problem), while lower drift rates cause slower decision-making (hard problem)

Parameter comparison

```r
# Compare Parameters Between Conditions
  ## BIAS
  # Aggregate bias means and SDs
A_means <- aggregate(A ~ condition, data = modelfit, FUN = mean)
A_sds <- aggregate(A ~ condition, data = modelfit, FUN = sd)
  # Bias t-test
A_t_test <- t.test(modelfit$A ~ modelfit$condition, paired = TRUE)
print(A_t_test)
```

```
##
##  Paired t-test
##
## data:  modelfit$A by modelfit$condition
## t = -2.7032, df = 11, p-value = 0.02054
## alternative hypothesis: true mean difference is not equal to 0
## 95 percent confidence interval:
##  -151.43399  -15.50735
## sample estimates:
## mean difference
##       -83.47067
```

```r
  ## BOUNDARY THRESHOLD
  # Aggregate boundary threshold means and SDs
b_means <- aggregate(b ~ condition, data = modelfit, FUN = mean)
b_sds <- aggregate(b ~ condition, data = modelfit, FUN = sd)
  # Boundary threshold t-test
b_t_test <- t.test(modelfit$b ~ modelfit$condition, paired = TRUE)
print(b_t_test)
```

```
##
##  Paired t-test
##
## data:  modelfit$b by modelfit$condition
## t = -0.86383, df = 11, p-value = 0.4061
## alternative hypothesis: true mean difference is not equal to 0
## 95 percent confidence interval:
##  -42.84929  18.69481
## sample estimates:
## mean difference
##       -12.07724
```

```r
  ## DRIFT RATE
  # Aggregate drift rate means and SDs
v1_means <- aggregate(v1 ~ condition, data = modelfit, FUN = mean)
v1_sds <- aggregate(v1 ~ condition, data = modelfit, FUN = sd)
  # Drift rate t-test
v1_t_test <- t.test(modelfit$v1 ~ modelfit$condition, paired = TRUE)
print(v1_t_test)
```

```
##
##  Paired t-test
##
## data:  modelfit$v1 by modelfit$condition
## t = -0.05592, df = 11, p-value = 0.9564
## alternative hypothesis: true mean difference is not equal to 0
## 95 percent confidence interval:
##  -0.08055281  0.07656104
## sample estimates:
## mean difference
##    -0.001995884
```

```r
  ## DRIFT RATE SD
  # Aggregate drift rate SD means and SDs
s_means <- aggregate(s ~ condition, data = modelfit, FUN = mean)
s_sds <- aggregate(s ~ condition, data = modelfit, FUN = sd)
  # Drift rate SD t-test
s_t_test <- t.test(modelfit$s ~ modelfit$condition, paired = TRUE)
print(s_t_test)
```

```
##
##  Paired t-test
##
## data:  modelfit$s by modelfit$condition
## t = -0.69046, df = 11, p-value = 0.5042
## alternative hypothesis: true mean difference is not equal to 0
## 95 percent confidence interval:
##  -0.05596425  0.02923639
## sample estimates:
## mean difference
##     -0.01336393
```

```r
  ## NON-DECISION TIME
  # Aggregate non-decision time means and SDs
ter_means <- aggregate(ter ~ condition, data = modelfit, FUN = mean)
ter_sds <- aggregate(ter ~ condition, data = modelfit, FUN = sd)
  # Non-decision time t-test
ter_t_test <- t.test(modelfit$ter ~ modelfit$condition, paired = TRUE)
print(ter_t_test)
```

```
##
##  Paired t-test
##
## data:  modelfit$ter by modelfit$condition
## t = 0.18166, df = 11, p-value = 0.8592
## alternative hypothesis: true mean difference is not equal to 0
## 95 percent confidence interval:
##  -93.91023 110.80624
## sample estimates:
## mean difference
##        8.448001
```

Descriptives parameters per condition

```
resultsc1 <- modelfit %>%
  filter(condition == 1)
skimr::skim(resultsc1)
```

Table 7: Data summary

Name	resultsc1
Number of rows	12
Number of columns	7
Column type frequency:	
numeric	7
Group variables	None

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
ID	0	1	6.50	3.61	1.00	3.75	6.50	9.25	12.00	
condition	0	1	1.00	0.00	1.00	1.00	1.00	1.00	1.00	
s	0	1	0.17	0.06	0.10	0.13	0.15	0.19	0.30	
A	0	1	204.55	99.00	8.65	180.51	226.74	255.48	339.11	
ter	0	1	260.87	113.80	91.89	184.82	273.50	332.54	439.71	
b	0	1	415.48	38.56	356.28	394.91	403.28	438.02	503.50	
v1	0	1	0.77	0.10	0.66	0.70	0.73	0.81	0.96	

```
resultsc2 <- modelfit %>%
  filter(condition == 2)
skimr::skim(resultsc2)
```

Table 9: Data summary

Name	resultsc2
Number of rows	12
Number of columns	7
Column type frequency:	
numeric	7

Group variables	None

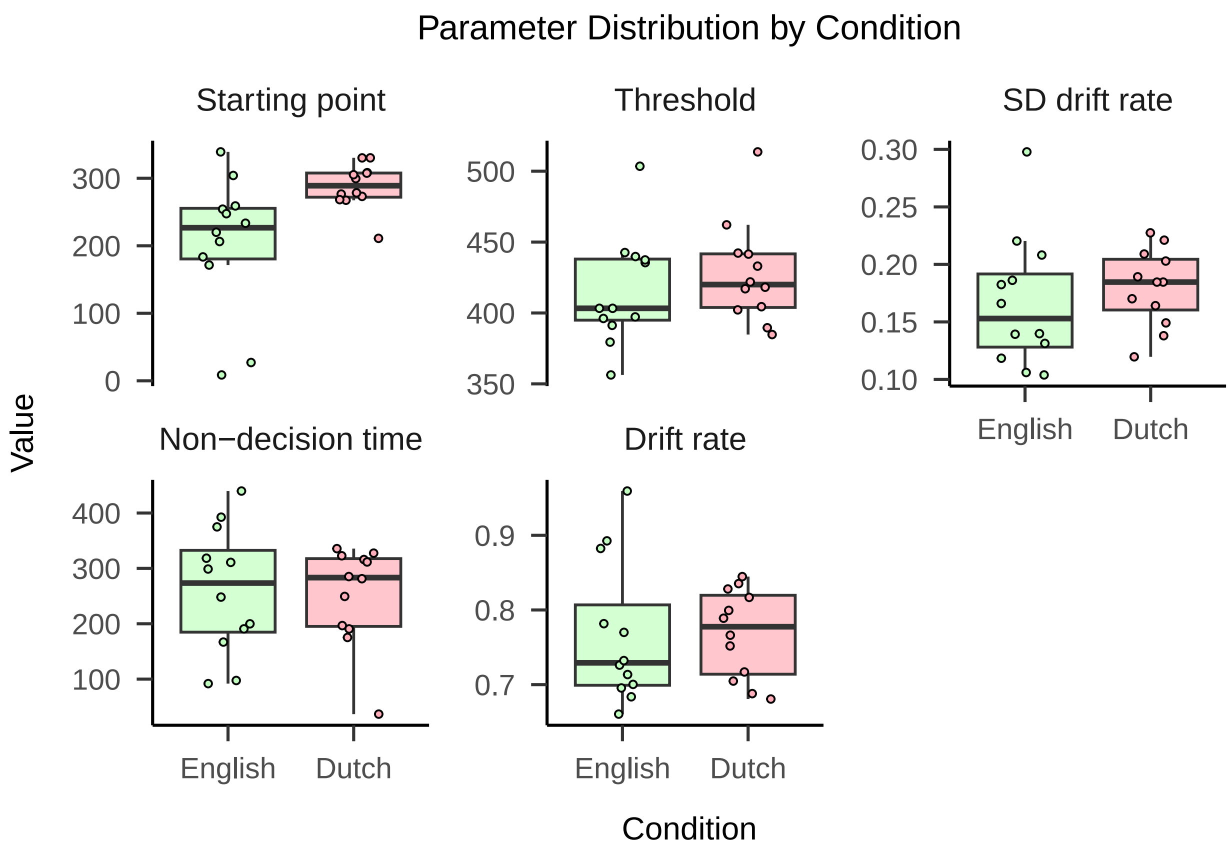
Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
ID	0	1	6.50	3.61	1.00	3.75	6.50	9.25	12.00	
condition	0	1	2.00	0.00	2.00	2.00	2.00	2.00	2.00	
s	0	1	0.18	0.03	0.12	0.16	0.18	0.20	0.23	
A	0	1	288.03	33.00	211.01	271.96	288.98	307.77	330.29	
ter	0	1	252.42	88.60	36.82	195.21	283.32	317.70	335.71	
b	0	1	427.56	35.33	384.78	403.86	420.04	441.70	513.63	
v1	0	1	0.77	0.06	0.68	0.71	0.78	0.82	0.84	

Parameter plots

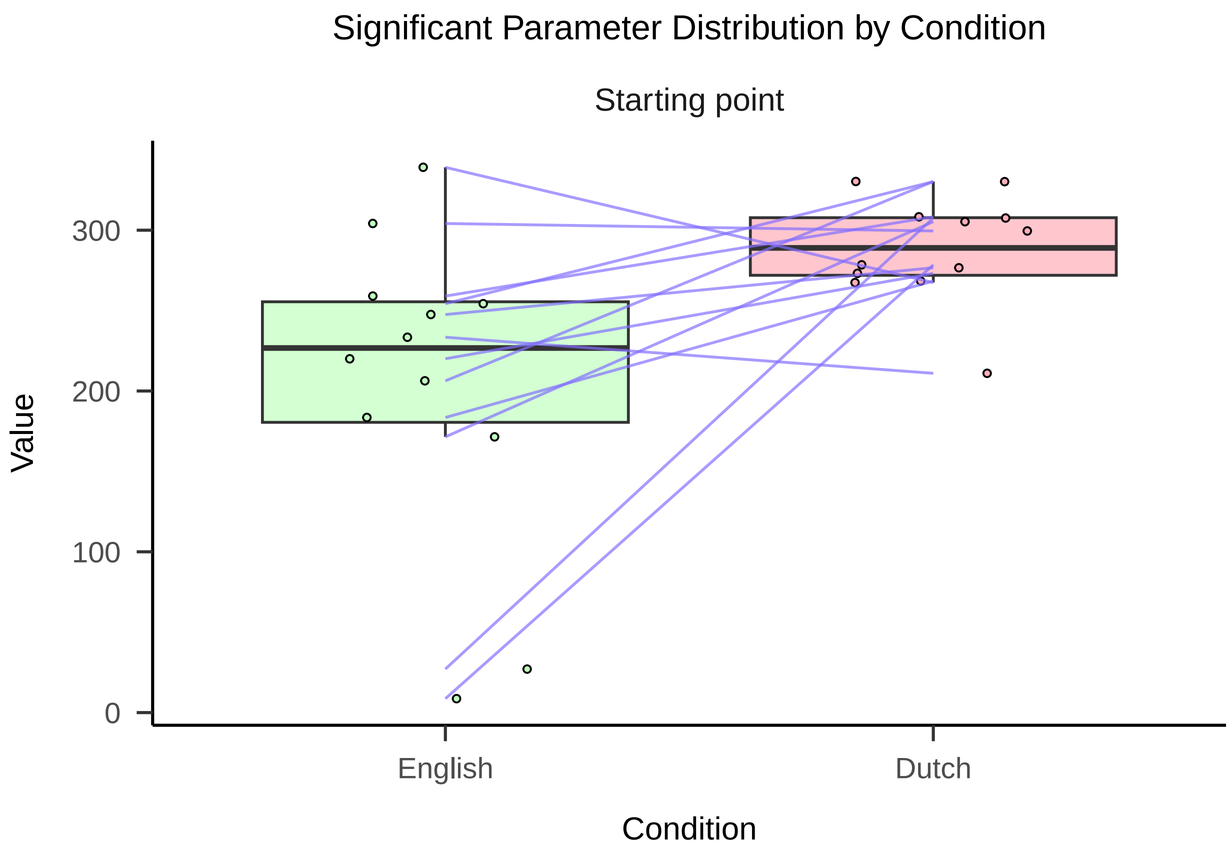
Boxplots for all parameters

```r
modelfit_long <- modelfit %>%
  pivot_longer(cols = c(s, A, ter, b, v1),
               names_to = "parameter", values_to = "value")

ggplot(modelfit_long,
       aes(x = factor(condition), y = value, fill = factor(condition))) +
  geom_boxplot(outlier.shape = NA, alpha = 0.7) +
  geom_jitter(shape = 21, size = 1, position = position_jitter(0.2)) +
  scale_x_discrete(labels = c("English", "Dutch")) +
  facet_wrap(~ parameter, scales = "free_y",
             labeller = as_labeller(c("s" = "SD drift rate",
                                      "A" = "Starting point",
                                      "ter" = "Non-decision time",
                                      "b" = "Threshold",
                                      "v1" = "Drift rate"))) +
  scale_fill_manual(values = mycolors1) +
  papaja::theme_apa() +
  theme(legend.position = "none") +
  labs(x = "Condition", y = "Value", title = "Parameter Distribution by Condition")
```



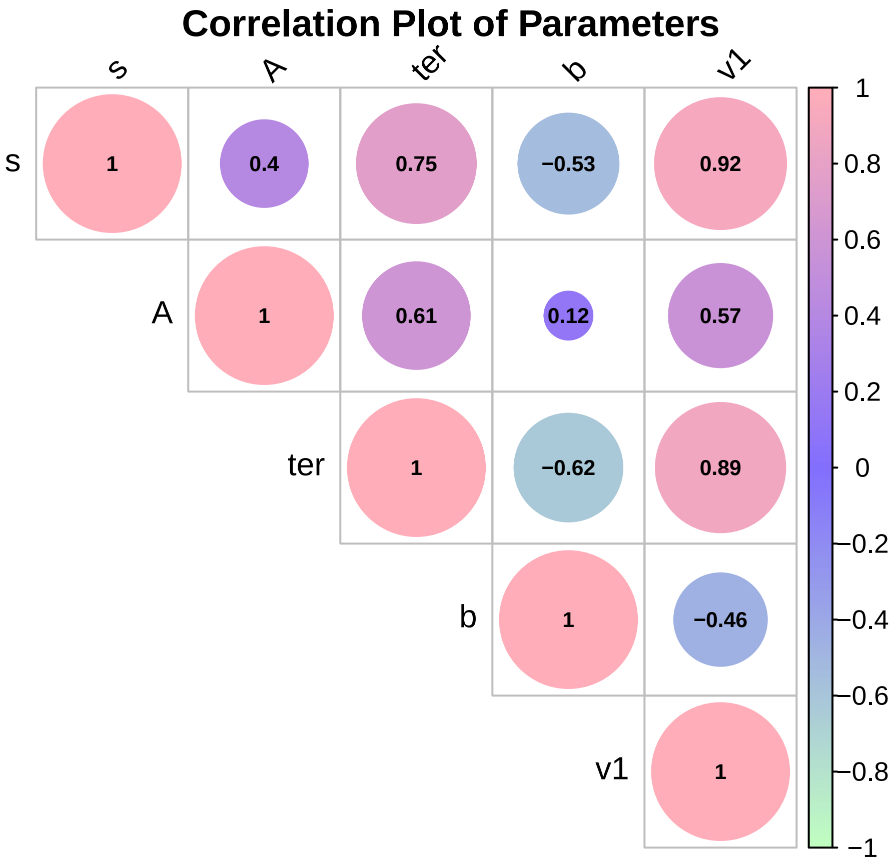
### Boxplots and slopes for significant parameters

```
# filter for significant parameters
significant_parameters <- modelfit_long %>%
  filter(parameter %in% c("A"))

# plot
ggplot(significant_parameters,
       aes(x = factor(condition), y = value, fill = factor(condition))) +
  geom_boxplot(outlier.shape = NA, alpha = 0.7) +
  geom_jitter(shape = 21, size = 1, position = position_jitter(0.2)) +
  geom_line(aes(group = ID), color = "slateblue1", alpha = 0.7) +
  scale_x_discrete(labels = c("English", "Dutch")) +
  facet_wrap(~ parameter, scales = "free_y",
             labeller = as_labeller(c("s" = "SD drift rate",
                                      "A" = "Starting point",
                                      "ter" = "Non-decision time",
                                      "b" = "Threshold",
                                      "v1" = "Drift rate"))) +
  scale_fill_manual(values = mycolors1) +
  papaja::theme_apa() +
  theme(legend.position = "none") +
  labs(x = "Condition", y = "Value",
       title = "Significant Parameter Distribution by Condition")
```



### Correlations of parameters

```
# color palette for degree of correlation
color_palette <- colorRampPalette(c(color1, "slateblue1", color2))(200)

# correlation matrix
cor_matrix <- cor(modelfit[,3:7])

# plot
corrplot::corrplot(cor_matrix, method = "circle", type = "upper",
                   tl.col = "black", tl.srt = 45,
                   addCoef.col = "black", number.cex = 0.7,
                   col = color_palette,
                   title = "Correlation Plot of Parameters",
                   mar = c(0,0,1,0))
```



Correlation Plot of Parameters