

RLDM_Report

Larissa Weyler

2024-06-14

```
setwd("~/University/Master Cognitive Neuroscience/Reinforcement Learning and
Decision-Making - Computational and Neural Mechanisms/Report/Code")

library(ggplot2)
library(tidyverse)
library(gridExtra)
source("helper_functions.r")
options(scipen = 999) # no scientific notation

rawdata <- read.csv("dataset13.csv", sep = ",")

# colors for the plots
color1 <- "maroon4"
color2 <- "paleturquoise4"
mycolors <- c(color1, color2)
```

Raw data exploration

Quick overview

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

Data summary


Name	rawdata
Number of rows	9600
Number of columns	4

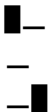

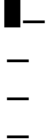
Column type frequency:

numeric	4
---------	---

Group variables	None
-----------------	------

Variable type: numeric

skim_variab	le	n_missi	ng	complete_r	ate	mea	n	sd	p0	p25	p50	p75	p100	hist
ID		0		1		6.50		3.45	1.0	3.75	6.50	9.25	12.0	

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
condition	0	1	1.50	0.50	1.0	1.00	1.50	2.00	2.0	
correct	0	1	0.85	0.36	0.0	1.00	1.00	1.00	1.0	
rt	0	1	670.49	2230.91	325.7	493.77	602.55	732.29	217104.5	

```
summary(rawdata)
```

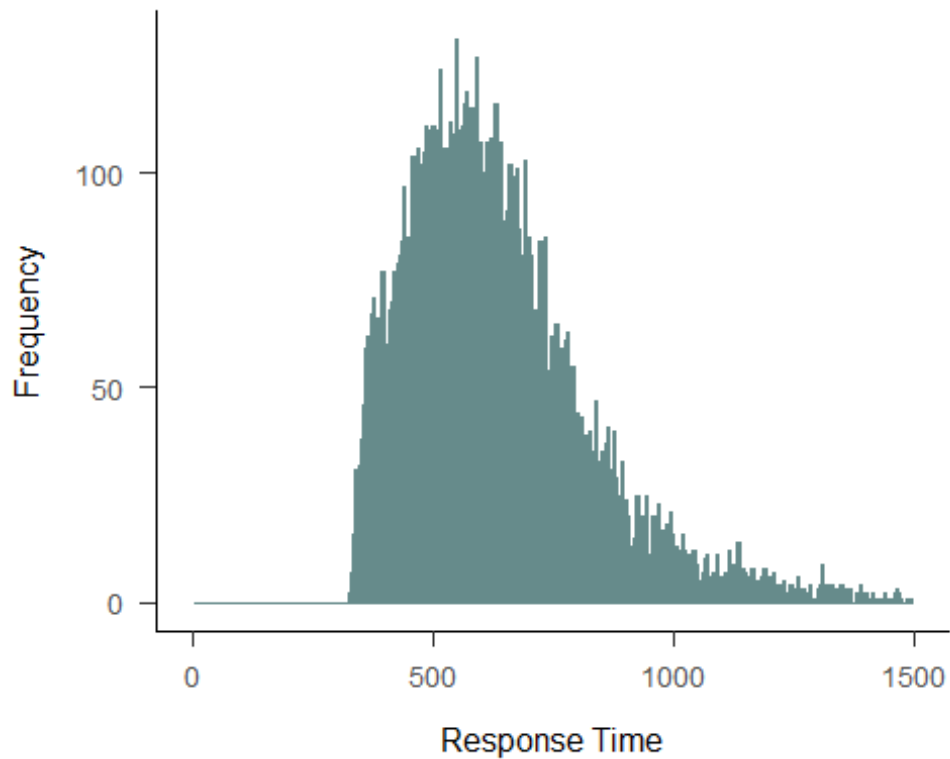
```
##      ID      condition      correct      rt
##  Min.   : 1.00   Min.   :1.0   Min.   :0.0000   Min.   : 325.7
## 1st Qu.: 3.75   1st Qu.:1.0   1st Qu.:1.0000   1st Qu.: 493.8
## Median : 6.50   Median :1.5   Median :1.0000   Median : 602.5
## Mean   : 6.50   Mean   :1.5   Mean   :0.8491   Mean   : 670.5
## 3rd Qu.: 9.25   3rd Qu.:2.0   3rd Qu.:1.0000   3rd Qu.: 732.3
## Max.   :12.00   Max.   :2.0   Max.   :1.0000   Max.   :217104.5
```

Sample size

```
N <- length(unique(rawdata$ID))
N
## [1] 12
```

Histogram of overall RT distribution

```
ggplot(rawdata, aes(x = rt)) +
  geom_histogram(binwidth = 5, fill = color2, color = color2) +
  papaja::theme_apac() +
  labs(x = "Response Time",
       y = "Frequency") +
  xlim(c(0,1500))
```



Outliers

Outlier removal

```
dataNO <- subset(rawdata,
  rawdata$rt > (quantile(rawdata$rt, 0.25) - 1.5 *
IQR(rawdata$rt))
  & rawdata$rt < (quantile(rawdata$rt, 0.75) + 1.5 *
IQR(rawdata$rt)))
```

Outlier inspection

```
outliers <- anti_join(rawdata, dataNO)

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

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

Data summary




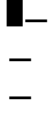
Name	outliers
Number of rows	375
Number of columns	4

Column type frequency:

numeric	4
---------	---

Group variables None

Variable type: numeric

skim_vari able	n_miss ing	complete_ rate	mea n	sd	p0	p25	p50	p75	p100	his t
ID	0	1	6.31	3.48	1.00	3.00	6.00	9.00	12.0	
condition	0	1	1.75	0.44	1.00	1.00	2.00	2.00	2.0	
correct	0	1	0.77	0.42	0.00	1.00	1.00	1.00	1.0	
rt	0	1	2088 .88	1118 0.13	1090 .22	1157 .28	1258 .58	1466 .86	2171 04.5	

```
summary(outliers)
```

```
##           ID           condition           correct           rt
## Min.      : 1.000   Min.      :1.000   Min.      :0.000   Min.      : 1090
## 1st Qu.: 3.000   1st Qu.:1.000   1st Qu.:1.000   1st Qu.: 1157
## Median : 6.000   Median :2.000   Median :1.000   Median : 1259
## Mean    : 6.307   Mean    :1.747   Mean    :0.768   Mean    : 2089
## 3rd Qu.: 9.000   3rd Qu.:2.000   3rd Qu.:1.000   3rd Qu.: 1467
## Max.    :12.000   Max.    :2.000   Max.    :1.000   Max.    :217105
```

```
# descriptives per participant
```

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

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

```
outlierDesc
```

```
##           1           2           3           4
5
## N          36.0000000   38.0000000   26.0000000   33.0000000
```

```
29.0000000
## M rt      1455.9409128 1490.3784749 1300.3821682 1569.0327943
1960.5585288
## MD rt      1231.5174287 1235.4291807 1217.4597779 1284.6138196
1258.4802631
## M accuracy  0.7222222    0.7894737    0.7307692    0.8181818
0.8275862
##           6           7           8           9          10
## N           29.0000000    38.0000000    32.0000    26.0000000    31.0000000
## M rt      1250.9126059 1605.5681537 1314.3643 10120.3095314 1546.8375707
## MD rt      1190.3715318 1270.2581882 1237.3997 1309.5031438 1343.1782343
## M accuracy  0.7586207    0.7368421    0.6875    0.6153846    0.9032258
##           11           12
## N           26.0000000    31.0000000
## M rt      1282.6852502 1550.2934722
## MD rt      1200.4882060 1309.5745501
## M accuracy  0.7307692    0.8709677
```

Cleaned data inspection

```
# overview
skimr::skim(dataNO)
```

Data summary

Name dataNO
Number of rows 9225
Number of columns 4


Column type frequency:

numeric 4

Group variables None

Variable type: numeric

skim_vari ble	n_missi ng	complete_r ate	mea n	sd	p0	p25	p50	p75	p100	hist
ID	0	1	6.51	3.45	1.0	4.00	7.00	10.0 0	12.00	
condition	0	1	1.49	0.50	1.0	1.00	1.00	2.00	2.00	
correct	0	1	0.85	0.35	0.0	1.00	1.00	1.00	1.00	

skim_vari ble	n_missi ng	complete_r ate	mea n	sd	p0	p25	p50	p75	p100	hist
rt	0	1	612. 83	161. 42	325 .7	488. 97	593. 87	713. 79	1089. 55	

```
summary(dataNO)
```

```
##          ID          condition          correct          rt
##  Min.   : 1.000   Min.   :1.00   Min.   :0.0000   Min.   : 325.7
## 1st Qu.: 4.000   1st Qu.:1.00   1st Qu.:1.0000   1st Qu.: 489.0
##  Median : 7.000   Median :1.00   Median :1.0000   Median : 593.9
##  Mean   : 6.508   Mean   :1.49   Mean   :0.8524   Mean   : 612.8
## 3rd Qu.:10.000   3rd Qu.:2.00   3rd Qu.:1.0000   3rd Qu.: 713.8
##  Max.   :12.000   Max.   :2.00   Max.   :1.0000   Max.   :1089.6
```

```
# descriptives per participant
```

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

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

```
##          1          2          3          4          5
## N      764.0000000 762.0000000 774.0000000 767.0000000 771.0000000
## M rt    611.2135682 605.2133874 623.1877421 608.4431545 612.9221308
## MD rt    591.2154341 583.3580433 606.7147658 590.1263983 590.9818274
## M accuracy 0.8429319 0.8687664 0.8643411 0.8474576 0.8560311
##          6          7          8          9         10
## N      771.0000000 762.0000000 768.0000000 774.0000000 769.0000000
## M rt    616.8082063 616.5006765 613.5793017 611.7486947 621.2088737
## MD rt    592.7699539 597.5054846 594.0063780 597.2620260 611.0536794
## M accuracy 0.8677043 0.8215223 0.8411458 0.8320413 0.8491547
##          11         12
## N      774.0000000 769.0000000
## M rt    614.8993577 598.1144511
## MD rt    589.0037335 581.4576903
## M accuracy 0.8514212 0.8855657
```

```
# descriptives per condition x correctness
```

```
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))
)

```

```

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

```

```

## $correct_0
##      ID      condition      correct      rt
## Min.   : 1.000   Min.   :1.00   Min.   :0   Min.   : 343.3
## 1st Qu.: 4.000   1st Qu.:1.00   1st Qu.:0   1st Qu.: 473.0
## Median : 7.000   Median :1.00   Median :0   Median : 600.1
## Mean   : 6.497   Mean   :1.38   Mean   :0   Mean   : 614.3
## 3rd Qu.: 9.000   3rd Qu.:2.00   3rd Qu.:0   3rd Qu.: 723.0
## Max.   :12.000   Max.   :2.00   Max.   :0   Max.   :1089.1
##
## $correct_1
##      ID      condition      correct      rt
## Min.   : 1.00   Min.   :1.000   Min.   :1   Min.   : 325.7
## 1st Qu.: 3.00   1st Qu.:1.000   1st Qu.:1   1st Qu.: 491.4
## Median : 6.00   Median :2.000   Median :1   Median : 593.2
## Mean   : 6.51   Mean   :1.509   Mean   :1   Mean   : 612.6
## 3rd Qu.:10.00   3rd Qu.:2.000   3rd Qu.:1   3rd Qu.: 712.2
## Max.   :12.00   Max.   :2.000   Max.   :1   Max.   :1089.6
##
## $condition_1
##      ID      condition      correct      rt
## Min.   : 1.000   Min.   :1   Min.   :0.0000   Min.   : 325.7
## 1st Qu.: 4.000   1st Qu.:1   1st Qu.:1.0000   1st Qu.: 433.4
## Median : 6.000   Median :1   Median :1.0000   Median : 521.5
## Mean   : 6.507   Mean   :1   Mean   :0.8206   Mean   : 548.1
## 3rd Qu.:10.000   3rd Qu.:1   3rd Qu.:1.0000   3rd Qu.: 635.3
## Max.   :12.000   Max.   :1   Max.   :1.0000   Max.   :1088.6
##
## $condition_2
##      ID      condition      correct      rt
## Min.   : 1.000   Min.   :2   Min.   :0.0000   Min.   : 396.4
## 1st Qu.: 4.000   1st Qu.:2   1st Qu.:1.0000   1st Qu.: 565.8
## Median : 7.000   Median :2   Median :1.0000   Median : 661.1
## Mean   : 6.509   Mean   :2   Mean   :0.8854   Mean   : 680.2

```

```
## 3rd Qu.:10.000 3rd Qu.:2 3rd Qu.:1.0000 3rd Qu.: 772.9
## Max. :12.000 Max. :2 Max. :1.0000 Max. :1089.6
##
## $condition_1_correct_0
##      ID      condition      correct      rt
## Min.   : 1.000   Min.   :1   Min.   :0   Min.   : 343.3
## 1st Qu.: 4.000   1st Qu.:1   1st Qu.:0   1st Qu.: 434.2
## Median : 7.000   Median :1   Median :0   Median : 512.1
## Mean   : 6.506   Mean   :1   Mean   :0   Mean   : 544.7
## 3rd Qu.: 9.000   3rd Qu.:1   3rd Qu.:0   3rd Qu.: 627.7
## Max.   :12.000   Max.   :1   Max.   :0   Max.   :1036.4
##
## $condition_1_correct_1
##      ID      condition      correct      rt
## Min.   : 1.000   Min.   :1   Min.   :1   Min.   : 325.7
## 1st Qu.: 3.000   1st Qu.:1   1st Qu.:1   1st Qu.: 433.2
## Median : 6.000   Median :1   Median :1   Median : 524.4
## Mean   : 6.507   Mean   :1   Mean   :1   Mean   : 548.8
## 3rd Qu.:10.000   3rd Qu.:1   3rd Qu.:1   3rd Qu.: 637.0
## Max.   :12.000   Max.   :1   Max.   :1   Max.   :1088.6
##
## $condition_2_correct_0
##      ID      condition      correct      rt
## Min.   : 1.000   Min.   :2   Min.   :0   Min.   : 446.2
## 1st Qu.: 3.250   1st Qu.:2   1st Qu.:0   1st Qu.: 624.6
## Median : 7.000   Median :2   Median :0   Median : 709.3
## Mean   : 6.483   Mean   :2   Mean   :0   Mean   : 727.8
## 3rd Qu.: 9.000   3rd Qu.:2   3rd Qu.:0   3rd Qu.: 817.9
## Max.   :12.000   Max.   :2   Max.   :0   Max.   :1089.1
##
## $condition_2_correct_1
##      ID      condition      correct      rt
## Min.   : 1.000   Min.   :2   Min.   :1   Min.   : 396.4
## 1st Qu.: 4.000   1st Qu.:2   1st Qu.:1   1st Qu.: 557.9
## Median : 6.500   Median :2   Median :1   Median : 654.0
## Mean   : 6.512   Mean   :2   Mean   :1   Mean   : 674.1
## 3rd Qu.:10.000   3rd Qu.:2   3rd Qu.:1   3rd Qu.: 765.7
## Max.   :12.000   Max.   :2   Max.   :1   Max.   :1089.6
```

RT and accuracy differences between conditions

Histograms of RTs by condition/correctness

histogram for RTs of correct/incorrect responses

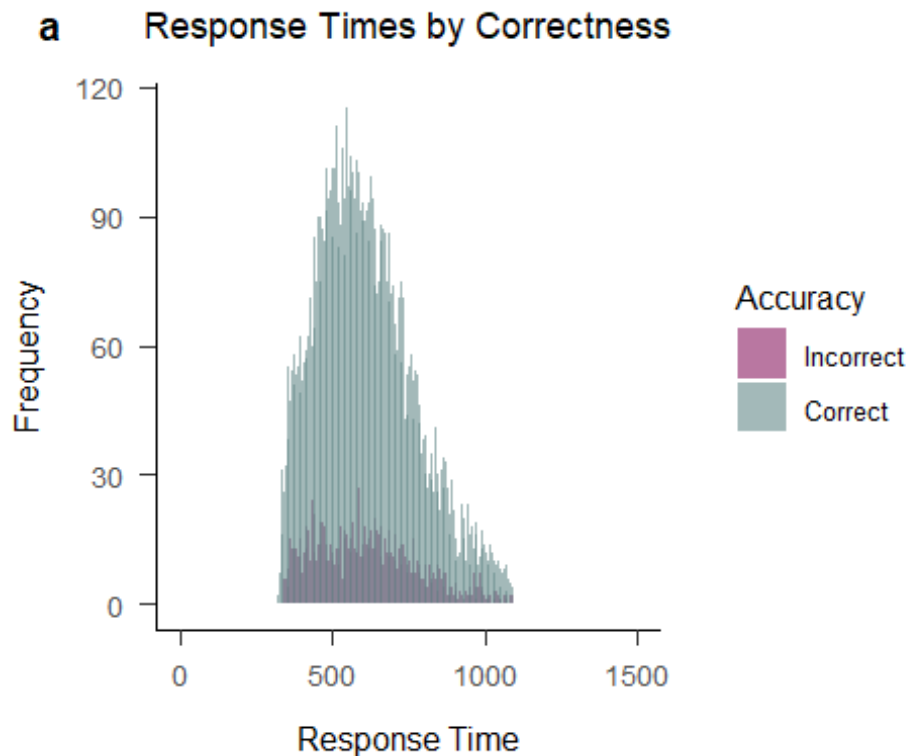
```
histCorrect <- ggplot(dataNO, aes(x = rt, group = factor(correct),
                                fill = factor(correct))) +
  geom_histogram(binwidth = 5, alpha = .6, position = "identity") +
  papaja::theme_apa() +
  labs(title = "Response Times by Correctness",
       x = "Response Time",
       y = "Frequency",
```



```

    fill = "Accuracy") +
  scale_fill_manual(values = mycolors,
                    labels = c("Incorrect", "Correct")) +
  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

```

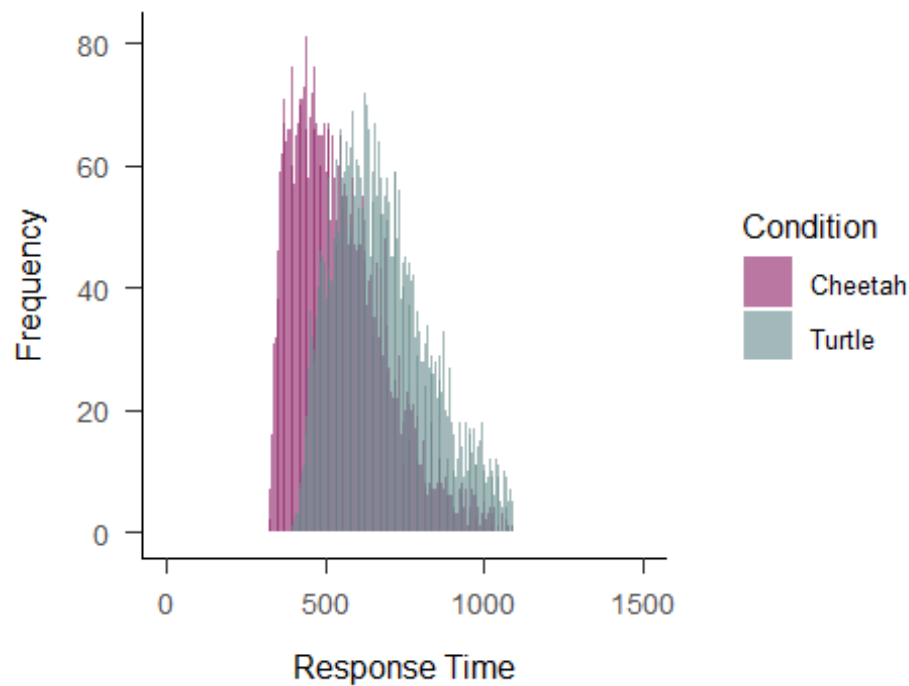


```

# histogram for RTs of the responsees of each condition
histCond <- ggplot(dataNO, aes(x = rt, group = factor(condition),
                              fill = factor(condition))) +
  geom_histogram(binwidth = 5, alpha = .6, position = "identity") +
  papaja::theme_apapa() +
  labs(title = "Response Times by Condition",
        x = "Response Time",
        y = "Frequency",
        fill = "Condition") +
  scale_fill_manual(values = mycolors,
                    labels = c("Cheetah", "Turtle")) +
  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

```

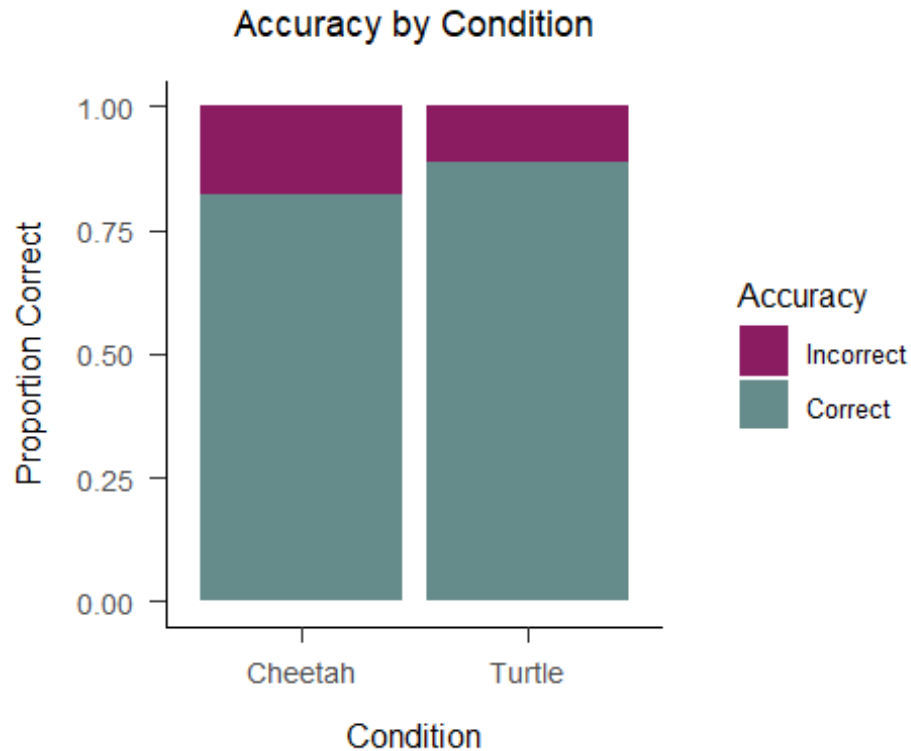
b Response Times by Condition



```
# if to be arrange next to each other (positioning of a/b labels needs to be
adjusted then):
# grid.arrange(histCorrect, histCond, ncol = 2)
```

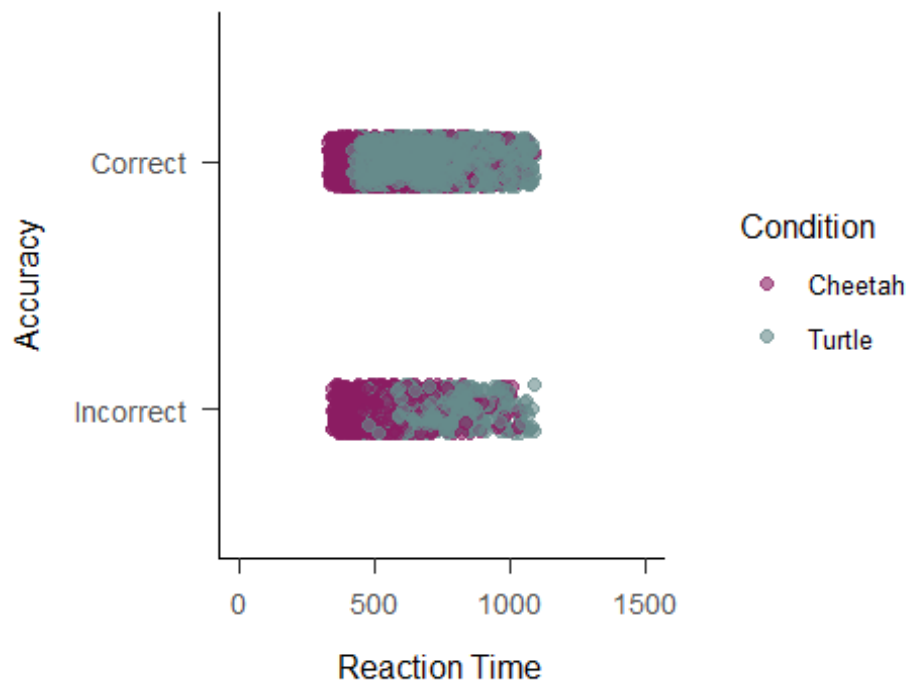
Plots of the accuracy by condition

```
# barplot of accuracy by condition
barAccuracy <- ggplot(dataNO, 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" = "Cheetah", "2" = "Turtle")) +
  scale_fill_manual(values = mycolors,
                   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")
barAccuracy
```



```
# dotplot of RT vs accuracy
dotRT <- ggplot(dataNO, 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 = mycolors,
    labels = c("Cheetah", "Turtle")) +
  papaja::theme_apapa() +
  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")
dotRT
```

Reaction Time vs. Accuracy by Condition



```
# if to be arrange next to each other (positioning of a/b labels needs to be
adjusted then):
#grid.arrange(barAccuracy, dotRT, ncol = 2, nrow = 1, widths = c(0.7,1))
```

T-tests

Paired t-tests to test whether the reaction times and accuracy differ between conditions. A paired Wilcoxon signed rank test would be an alternative that is similarly powerful and more robust against non-normality, but due to aggregating the data normality should not be a problem.

```
# difference of median rt between conditions
aggregatedDat <- dataNO %>%
  group_by(ID, condition) %>%
  summarize(median_rt = median(rt), mean_acc = mean(correct), .groups =
'drop')

cond1Rt <- aggregatedDat %>% filter(condition == 1) %>% pull(median_rt)
cond2Rt <- aggregatedDat %>% filter(condition == 2) %>% pull(median_rt)

t.test(cond1Rt, cond2Rt, paired = TRUE)

##
## Paired t-test
##
## data: cond1Rt and cond2Rt
## t = -39.056, df = 11, p-value = 0.0000000000003755
```


fitresults

##	ID	condition	s	A	ter	b	v1
## 1	1	1	0.2102226	325.6268	311.0390097	344.5674	0.7200989
## 2	1	2	0.1740852	266.0706	273.4850956	402.5512	0.6909158
## 3	2	1	0.1689041	288.5980	298.6012457	320.5288	0.7226989
## 4	2	2	0.1372623	322.1502	356.1932512	396.3158	0.7307956
## 5	3	1	0.2135962	345.5778	329.4548378	358.0326	0.7827132
## 6	3	2	0.1925881	303.9231	304.7751575	423.1455	0.7117532
## 7	4	1	0.2721179	313.8105	319.8961061	331.3865	0.7633145
## 8	4	2	0.2022639	293.1188	346.3928647	381.9415	0.7276319
## 9	5	1	0.2328482	333.3876	338.9879481	335.9998	0.7687905
## 10	5	2	0.1816300	243.5593	234.2130025	418.4254	0.6908694
## 11	6	1	0.2655652	263.5886	254.1932572	344.9116	0.7233929
## 12	6	2	0.1625401	315.1293	309.4881215	428.4046	0.7257411
## 13	7	1	0.2373699	282.2652	285.3146603	324.7594	0.6769337
## 14	7	2	0.1482446	298.4972	225.9746839	461.0681	0.6725480
## 15	8	1	0.2345589	292.7788	294.8127198	331.5225	0.7008190
## 16	8	2	0.1509050	318.8998	325.7901501	413.4079	0.7155574
## 17	9	1	0.2678973	317.1473	344.9211355	317.6376	0.7525807
## 18	9	2	0.1153810	215.6992	48.5289769	493.3914	0.6120406
## 19	10	1	0.2602065	312.0803	339.7538454	320.3202	0.7588066
## 20	10	2	0.1021500	202.5399	0.4787487	516.2175	0.6043429
## 21	11	1	0.1782137	314.5501	353.5567016	314.8382	0.7650809
## 22	11	2	0.1605162	241.0152	117.3727945	487.0423	0.6526044
## 23	12	1	0.2451656	312.8785	345.7775021	312.8797	0.8237968
## 24	12	2	0.1658102	287.8898	271.9974522	425.7611	0.7180035

Parameters:

s = *SD of drift rates* = This reflects variability in drift rates. However, as this parameter does not have an easily interpretable cognitive mapping, this parameter does not significantly differ between conditions in the provided datasets.

A = *upper limit of starting point* = This reflects the starting point of the evidence accumulation process. It reflects bias or expectations for one choice.

ter = *non-decision time* = This reflects the time necessary for processes that are not related to evidence integration. For example, the time it takes for activation of the motor cortex to result in the hand pressing the response button, or the time it takes for visual information to get from the stimulus to the visual cortex.

b = *threshold* = The distance from 0 to the threshold. It reflects cautiousness: lower thresholds lead to faster responses but a higher error rate.

$v1$ = *drift rate* = The quality of the evidence or difficulty of the problem. Higher drift rates lead to faster and more accurate responses.

Parameter comparison

Via paired Wilcoxon signed rank test because the parameters are not normally distributed.

```
parameters <- c("s", "A", "ter", "b", "v1")
paracomparison <- data.frame(Parameter = character(), tstatistic = numeric(),
                             pvalue = numeric(), stringsAsFactors = FALSE)

for (para in parameters) {
  tresult <- wilcox.test(fitresults[fitresults$condition == 1, para],
                        fitresults[fitresults$condition == 2, para],
                        paired = TRUE)
  paracomparison <- rbind(paracomparison,
                          c(para, tresult$statistic, tresult$p.value))
}
colnames(paracomparison) <- c("parameter", "tstatistic", "pvalue")
paracomparison
```

##	parameter	tstatistic	pvalue
## 1	s	78	0.00048828125
## 2	A	61	0.0922851562500001
## 3	ter	62	0.0771484375000001
## 4	b	0	0.00048828125
## 5	v1	70	0.01220703125

Descriptives of parameters by condition

```
results1 <- fitresults %>%
  filter(condition == 1)
skimr::skim(results1)
```

Data summary



Name	results1
Number of rows	12
Number of columns	7







Column type frequency:

numeric	7
---------	---

Group variables	None
-----------------	------

Variable type: numeric

skim_variab	n_missi	complete_r	mea						p10	
le	ng	ate	n	sd	p0	p25	p50	p75	0	hist
ID	0	1	6.50	3.6	1.00	3.75	6.50	9.25	12.0	
				1					0	

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
condition	0	1	1.00	0.00	1.00	1.00	1.00	1.00	1.00	
s	0	1	0.23	0.03	0.17	0.21	0.24	0.26	0.27	
A	0	1	308.52	22.98	263.59	291.73	313.34	319.27	345.58	
ter	0	1	318.03	30.02	254.19	297.65	324.68	341.05	353.56	
b	0	1	329.78	13.97	312.88	319.65	328.07	338.14	358.03	
v1	0	1	0.75	0.04	0.68	0.72	0.76	0.77	0.82	

```
results2 <- fitresults %>%
  filter(condition == 2)
skimr::skim(results2)
```

Data summary




Name	results2
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	0.16	0.03	0.10	0.15	0.16	0.18	0.20	

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
A	0	1	275.71	41.15	202.54	242.92	290.50	306.72	322.15	
ter	0	1	234.56	117.66	0.48	198.82	272.74	313.56	356.19	
b	0	1	437.31	42.32	381.94	410.69	424.45	467.56	516.22	
v1	0	1	0.69	0.04	0.60	0.67	0.70	0.72	0.73	

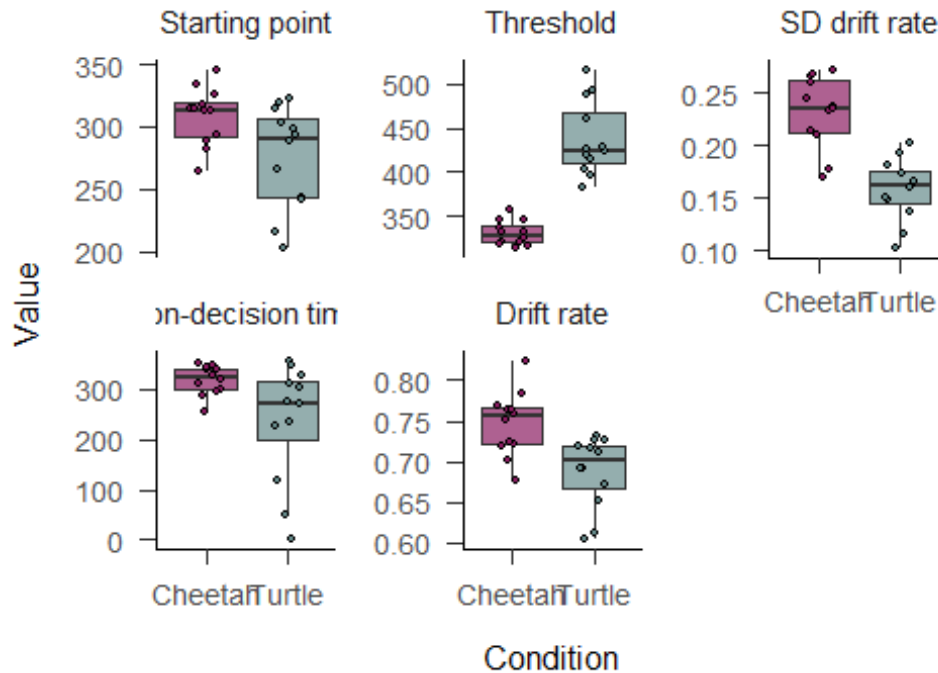
Parameter plots

Boxplots of all parameters

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

ggplot(fitresults_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("Cheetah", "Turtle")) +
  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 = mycolors) +
  papaja::theme_apa() +
  theme(legend.position = "none") +
  labs(x = "Condition", y = "Value", title = "Parameter Distribution by Condition")
```

Parameter Distribution by Condition

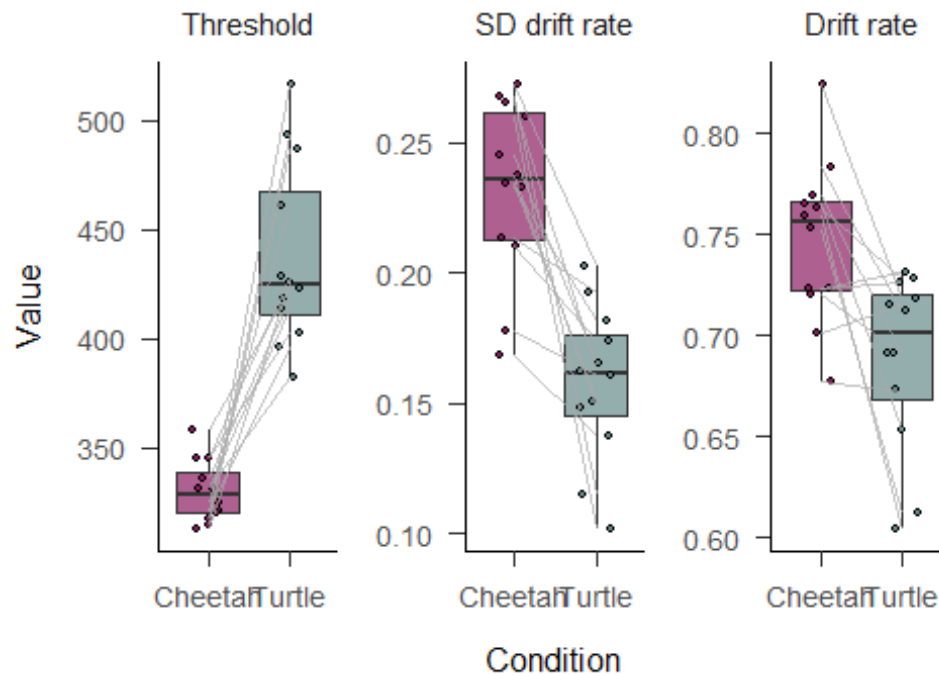


Slope + boxplots for significant parameters

```
# filter for significant parameters
significant_parameters <- fitresults_long %>%
  filter(parameter %in% c("s", "b", "v1"))

# 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 = "darkgrey", alpha = 0.7) +
  scale_x_discrete(labels = c("Cheetah", "Turtle")) +
  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 = mycolors) +
  papaja::theme_apapa() +
  theme(legend.position = "none") +
  labs(x = "Condition", y = "Value",
    title = "Significant Parameter Distribution by Condition")
```

Significant Parameter Distribution by Condition



Plot accumulation process

One trial of evidence accumulation using the average parameters per condition. Based on the DDM function developed during the workgroup, adapted to save current evidence of every time point of the accumulation process.

```
# DDM function
DDM <- function(b, v1, s, ter, A) {
  act <- A
  time <- ter
  accumulation <- data.frame(time = numeric(),
                             activation = numeric(),
                             v1 = numeric(),
                             s = numeric(),
                             ter = numeric(),
                             A = numeric(),
                             condition = character())

  i <- 1
  while ((act < b) & (act > 0)) {
    accumulation <- rbind(accumulation, c(time, act, v1, s, ter, A,
condition))
    drift <- rnorm(1, mean = v1, sd = s)
    act <- act + drift
    time <- time + 2
    i <- i + 1
  }
  colnames(accumulation) <- c("time", "activation", "v1", "s", "ter", "A",
```

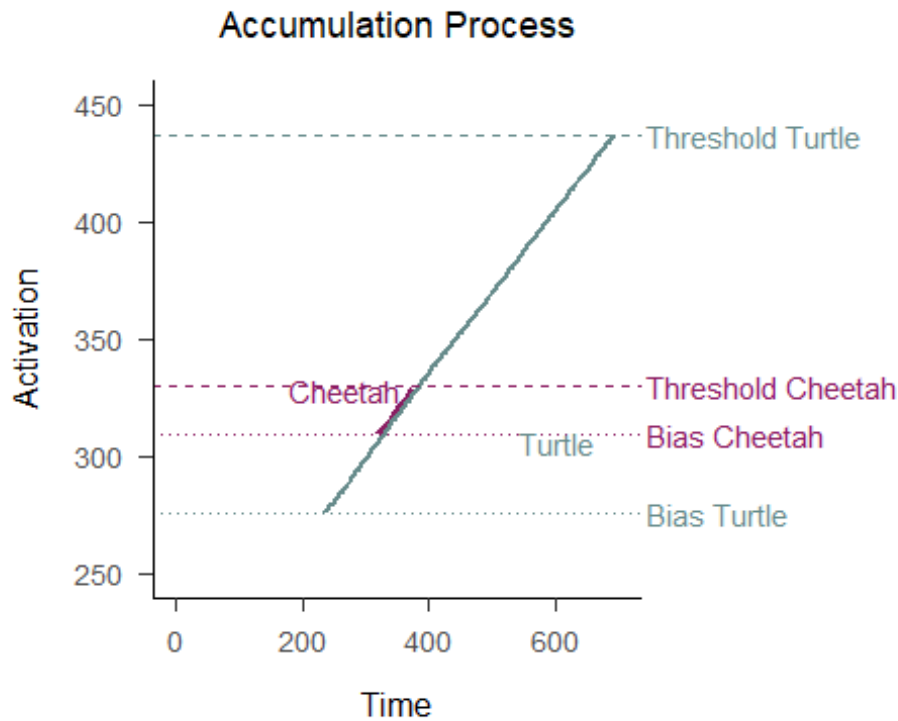
```

"condition")
  return(accumulation)
}

# fit DDM with average parameters
accumulationCheetah <- DDM(b = 330, v1 = 0.747, s = 0.232, ter = 318, A =
309)
accumulationTurtle <- DDM(b = 437, v1 = 0.688, s = 0.158, ter = 235, A =
276)

# plot evidence accumulation
ggplot() +
  geom_line(data = accumulationCheetah,
            aes(x = time, y = activation), color = color1, linewidth = 0.8) +
  geom_line(data = accumulationTurtle,
            aes(x = time, y = activation), color = color2, linewidth = 0.8) +
  geom_hline(yintercept = c(309, 276), linetype = "dotted", color = mycolors)
+
  geom_hline(yintercept = c(330, 437), linetype = "dashed", color = mycolors)
+
  geom_text(aes(x = max(accumulationTurtle$time) + 50, y = c(309, 330),
                label = c("Bias Cheetah", "Threshold Cheetah"),
                color = color1), hjust = 0) +
  geom_text(aes(x = max(accumulationTurtle$time) + 50, y = c(276, 437),
                label = c("Bias Turtle", "Threshold Turtle"),
                color = color2), hjust = 0) +
  geom_text(data = accumulationCheetah[1,],
            aes(x = time, y = activation, label = "Cheetah"),
            color = color1, vjust = -1.4, hjust = 0.8) +
  geom_text(data = accumulationTurtle[1,],
            aes(x = time, y = activation, label = "Turtle"),
            color = color2, vjust = -2.4, hjust = -2.6) +
  labs(title = "Accumulation Process",
        x = "Time",
        y = "Activation",
        color = "Condition") +
  scale_color_manual(name = "Condition", values = mycolors,
                     labels = c("Cheetah", "Turtle")) +
  papaja::theme_apo() +
  theme(legend.position = "none", plot.margin = unit(c(0.4, 4, 0.4, 0.4),
"cm")) +
  coord_cartesian(xlim = c(0, max(accumulationTurtle$time) + 10), clip =
"off") +
  scale_x_continuous(n.breaks = 6) +
  scale_y_continuous(limits = c(250, 450))

```



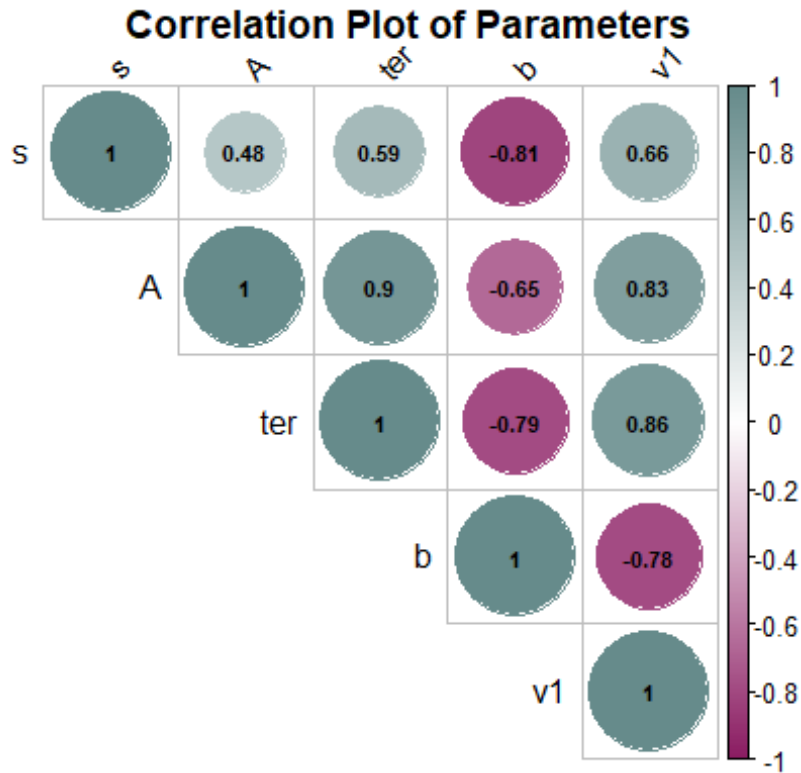
Correlation plot of the parameters

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

# correlation matrix
cor_matrix <- cor(fitresults[,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))
mtext("a", side = 3, line = 2.9, adj = 0, cex = 1.5, font = 2)
```

a



Correlation of just v1 and b

```
ggplot(fitresults, aes(x = v1, y = b, color = factor(condition))) +
  geom_point(size = 4) +
  scale_color_manual(name = "Condition", values = mycolors,
    labels = c("Cheetah", "Turtle")) +
  papaja::theme_apapa() +
  labs(x = "Drift rate", y = "Threshold",
    title = "Correlation Drift Rate and Threshold") +
  theme(plot.margin = unit(c(0.4, 0, 0.4, 0.6), "cm")) +
  annotate("text", x = 0.6, y = Inf, label = "b", size = 5,
    fontface = "bold", vjust = -1.5, hjust = 7) +
  coord_cartesian(xlim = c(0.6, 0.85), clip = "off")
```

b Correlation Drift Rate and Threshold

