GSandow_Fit_DDM

Greta Sandow

2024-06-11

Reinforcement Learning and Decision Making: Computational and Neural Mechanisms

---

Drift Diffusion Model - Model Fitting

Model fitting is a technique used to identify the set of parameters that best explain the observed data from our experimental task. By finding these optimal parameters, we can gain insights into the underlying decision-making processes.

> Let's start by loading the necessary packages and the data.

```
setwd("~/Documents/Cognitive Neuroscience/Reinforcement Learning & Decision Making")

library(tidyverse)
library(readr)
library(ggplot2)
library(RColorBrewer)
library(papaja)
library(reshape2)
source('helper_functions.r')

data <- read_csv("dataset9.csv")
```

The data contains the correctness and speed of the responses of 12 participants recorded at two time points.

```
names(data)
```

```
## [1] "ID"        "condition" "correct"   "rt"
```

```
length(unique(data$ID))
```

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

```
length(unique(data$condition))
```

```
## [1] 2
```

Removing Outliers

> When looking at the range and distribution of the response times, we see that there seem to be some extreme values.

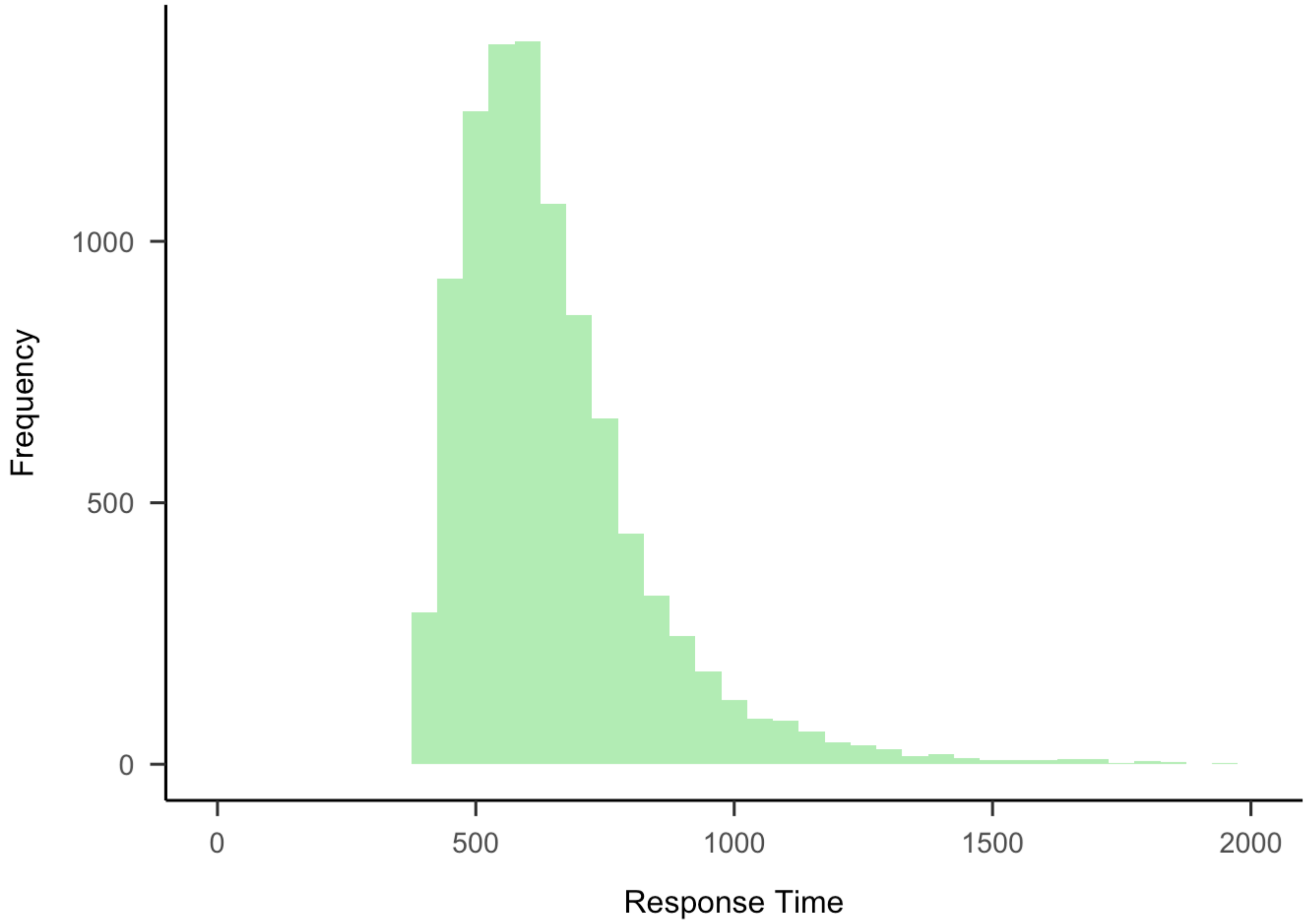
```
print(paste("Old Minimum Response Time:", round(min(data$rt), digits = 2)))
```

```
## [1] "Old Minimum Response Time: 379.01"
```

```
print(paste("Old Maximum Response Time:", round(max(data$rt), digits = 2)))
```

```
## [1] "Old Maximum Response Time: 20217.57"
```

```
ggplot(data, aes(x = rt)) + # Old Response Time Distribution
  geom_histogram(binwidth = 50, fill = "darkseagreen2") +
  xlim(c(0,2000))+
  labs(x = "Response Time",
       y = "Frequency")+
  theme_apa()
```



The extreme values, or outliers, can be removed using the Interquartile Range Method.

```
new_data <- data
Q1 <- quantile(data$rt, 0.25) # Removing Outliers with Interquartile Range Method
Q3 <- quantile(data$rt, 0.75)
IQR <- Q3 - Q1
lower_bound <- Q1 - 1.5 * IQR
upper_bound <- Q3 + 1.5 * IQR
new_data <- new_data[new_data$rt > lower_bound & new_data$rt < upper_bound, ]
data_fit <- new_data # duplicating the data to avoid problems with as.factor later wi
th fit_data()
```

Looking at the range and distribution of the cleaned response time data, we see that the most extreme values are no longer in the new data set.

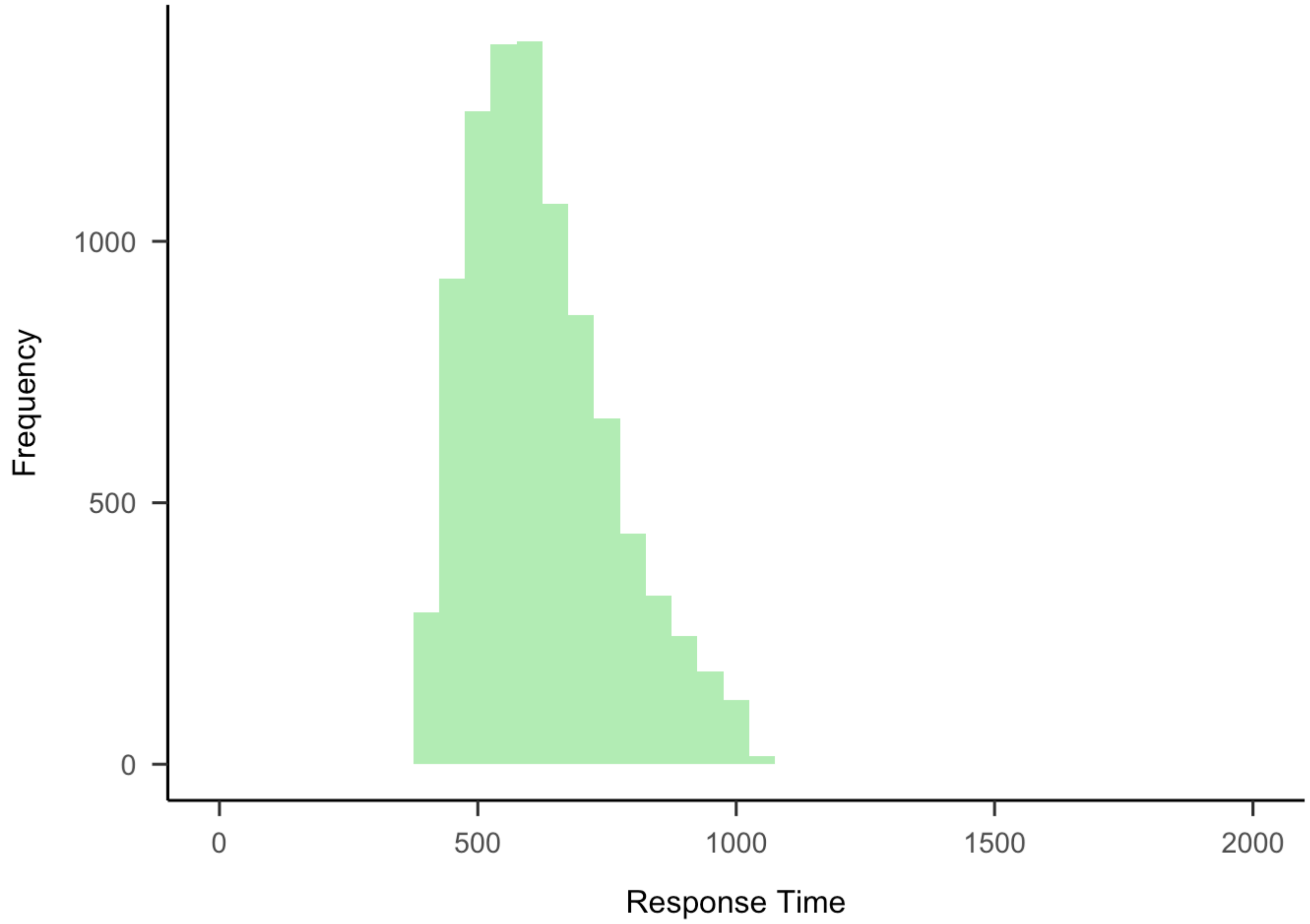
```
print(paste("New Minimum Response Time:", round(min(new_data$rt), digits = 2)))
```

```
## [1] "New Minimum Response Time: 379.01"
```

```
print(paste("New Maximum Response Time:", round(max(new_data$rt), digits = 2)))
```

```
## [1] "New Maximum Response Time: 1035.62"
```

```
ggplot(new_data, aes(x = rt)) + # New Response Time Distribution
  geom_histogram(binwidth = 50, fill = "darkseagreen2") +
  xlim(c(0,2000))+
  labs(x = "Response Time",
       y = "Frequency")+
  theme_apa()
```



Descriptive Statistics

> Before getting started with model fitting, let's look at the data in more detail.

```
new_data$condition <- as.factor(new_data$condition) # Change into factor
new_data$correct <- as.factor(new_data$correct) # Change into factor
print(paste("Average Response Time:", round(mean(new_data$rt), digits = 2)))
```

```
## [1] "Average Response Time: 622.75"
```

```
print(paste("SD Response Time:", round(sd(new_data$rt), digits = 2)))
```

```
## [1] "SD Response Time: 137.01"
```

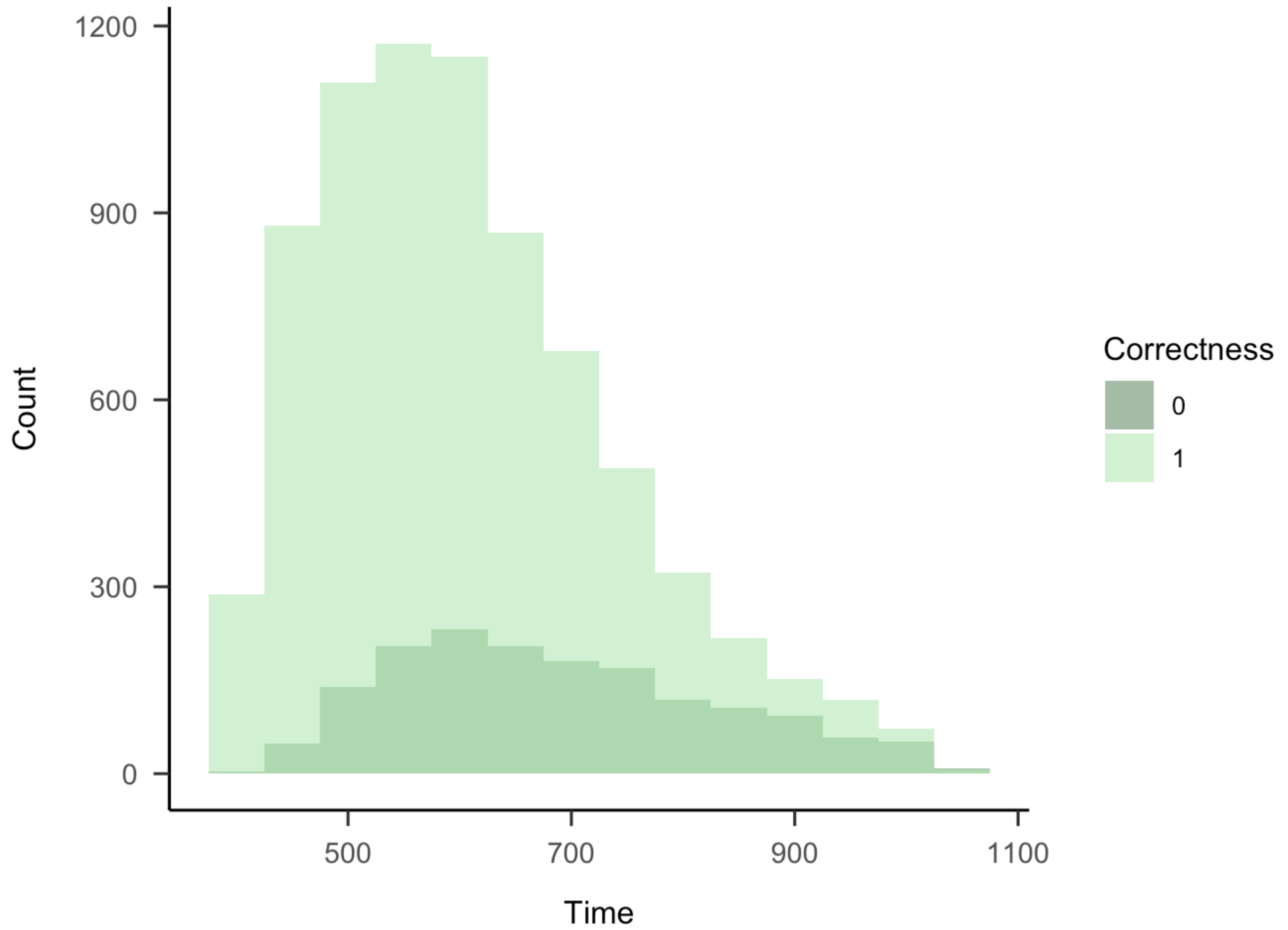
```
print(paste("Median Response Time:", round(median(new_data$rt), digits = 2)))
```

```
## [1] "Median Response Time: 600.61"
```

```
decision_counts_fit <- table(new_data$correct)
correct_decisions_fit <- decision_counts_fit["1"]
accuracy_fit <-correct_decisions_fit / sum(decision_counts_fit)*100
print(paste("Accuracy:", round(accuracy_fit, digits = 2), "%"))
```

```
## [1] "Accuracy: 82.31 %"
```

```
ggplot(new_data, aes(x = rt, fill = correct)) +
  geom_histogram(binwidth = 50, position = "identity", alpha = 0.6) +
  scale_fill_manual(values = c("darkseagreen4", "darkseagreen2")) +
  theme_apa() +
  labs(x = "Time", y = "Count", fill = "Correctness")
```



Developmental Differences in Response Time and Accuracy

> We will now take a closer look at the differences between the children and adolescents.

```
# Aggregate Data for Analysis
new_data_correct <- new_data[new_data$correct == 1, ]
df_median <- aggregate(rt ~ ID + condition, data = new_data_correct, FUN = median)
df_acc <- aggregate(correct ~ ID + condition, data = data_fit, FUN = mean)
df_aggregated <- cbind(df_median["ID"], df_median["condition"], df_median["rt"], df_a
cc["correct"])
# T-test Response Times
rt_means <- aggregate(rt ~ condition, data = df_aggregated, FUN = mean)
rt_sds <- aggregate(rt ~ condition, data = df_aggregated, FUN = sd)
paired_t_test_rt <- t.test(df_aggregated$rt ~ df_aggregated$condition, paired = TRUE)
print(paired_t_test_rt)
```

```
##
##  Paired t-test
##
## data:  df_aggregated$rt by df_aggregated$condition
## t = 23.558, df = 11, p-value = 9.169e-11
## alternative hypothesis: true mean difference is not equal to 0
## 95 percent confidence interval:
##  79.88026 96.34504
## sample estimates:
## mean difference
##        88.11265
```

```
# T-test Accuracy
acc_means <- aggregate(correct ~ condition, data = df_aggregated, FUN = mean)
acc_sds <- aggregate(correct ~ condition, data = df_aggregated, FUN = sd)
paired_t_test_acc <- t.test(df_aggregated$correct ~ df_aggregated$condition, paired =
TRUE)
print(paired_t_test_acc)
```

```
##
##  Paired t-test
##
## data:  df_aggregated$correct by df_aggregated$condition
## t = -75.902, df = 11, p-value = 2.582e-16
## alternative hypothesis: true mean difference is not equal to 0
## 95 percent confidence interval:
##  -0.3541201 -0.3341616
## sample estimates:
## mean difference
##      -0.3441408
```

These results show that there are significant differences in response time and accuracy. To help with interpretation, let's look at the respective values for each condition. Additionally, we will look at the response time distributions.

```
# Filtering data per condition
child_data <- filter(new_data, condition == 1)
adolescent_data <- filter(new_data, condition == 2)
# Response Times
print(paste("Average Response Time Children:", round(median(child_data$rt), digits =
2)))
```

```
## [1] "Average Response Time Children: 654.68"
```

```
print(paste("Average Response Time Adolescents:", round(median(adolescent_data$rt), d
igits = 2)))
```

```
## [1] "Average Response Time Adolescents: 559.75"
```

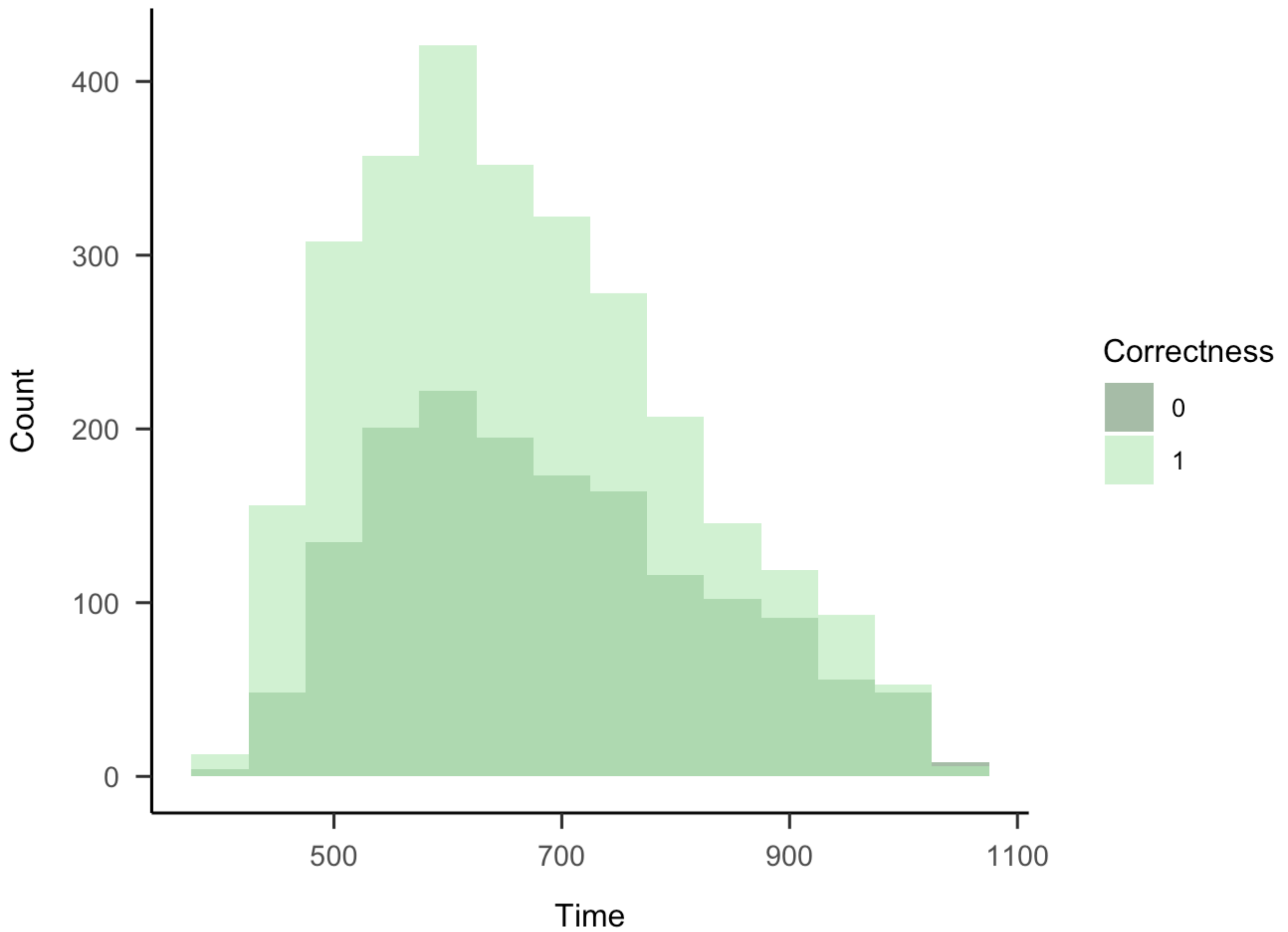
```
# Accuracy
decision_counts_child <- table(child_data$correct)
correct_decisions_child <- decision_counts_child["1"]
accuracy_child <-correct_decisions_child / sum(decision_counts_child)*100
print(paste("Accuracy Children:", round(accuracy_child, digits = 2), "%"))
```

```
## [1] "Accuracy Children: 64.43 %"
```

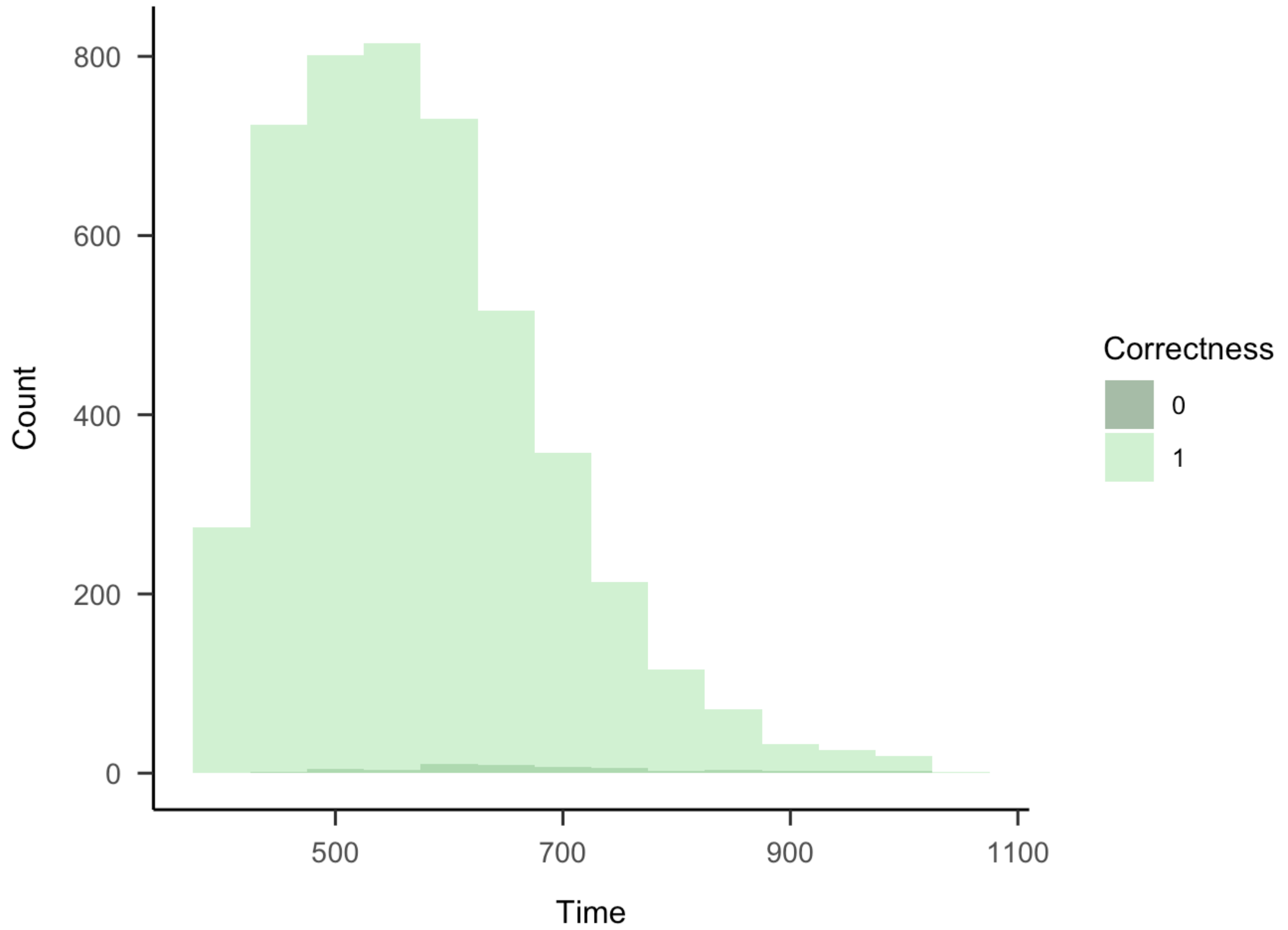
```
decision_counts_adol <- table(adolescent_data$correct)
correct_decisions_adol <- decision_counts_adol["1"]
accuracy_adol <-correct_decisions_adol / sum(decision_counts_adol)*100
print(paste("Accuracy Adolescents:", round(accuracy_adol, digits = 2), "%"))
```

```
## [1] "Accuracy Adolescents: 98.84 %"
```

```
# Correct / Incorrect Response Time Distribution
ggplot(child_data, aes(x = rt, fill = correct)) +
  geom_histogram(binwidth = 50, position = "identity", alpha = 0.6) +
  scale_fill_manual(values = c("darkseagreen4", "darkseagreen2")) +
  theme_apa() +
  labs(x = "Time", y = "Count", fill = "Correctness")
```

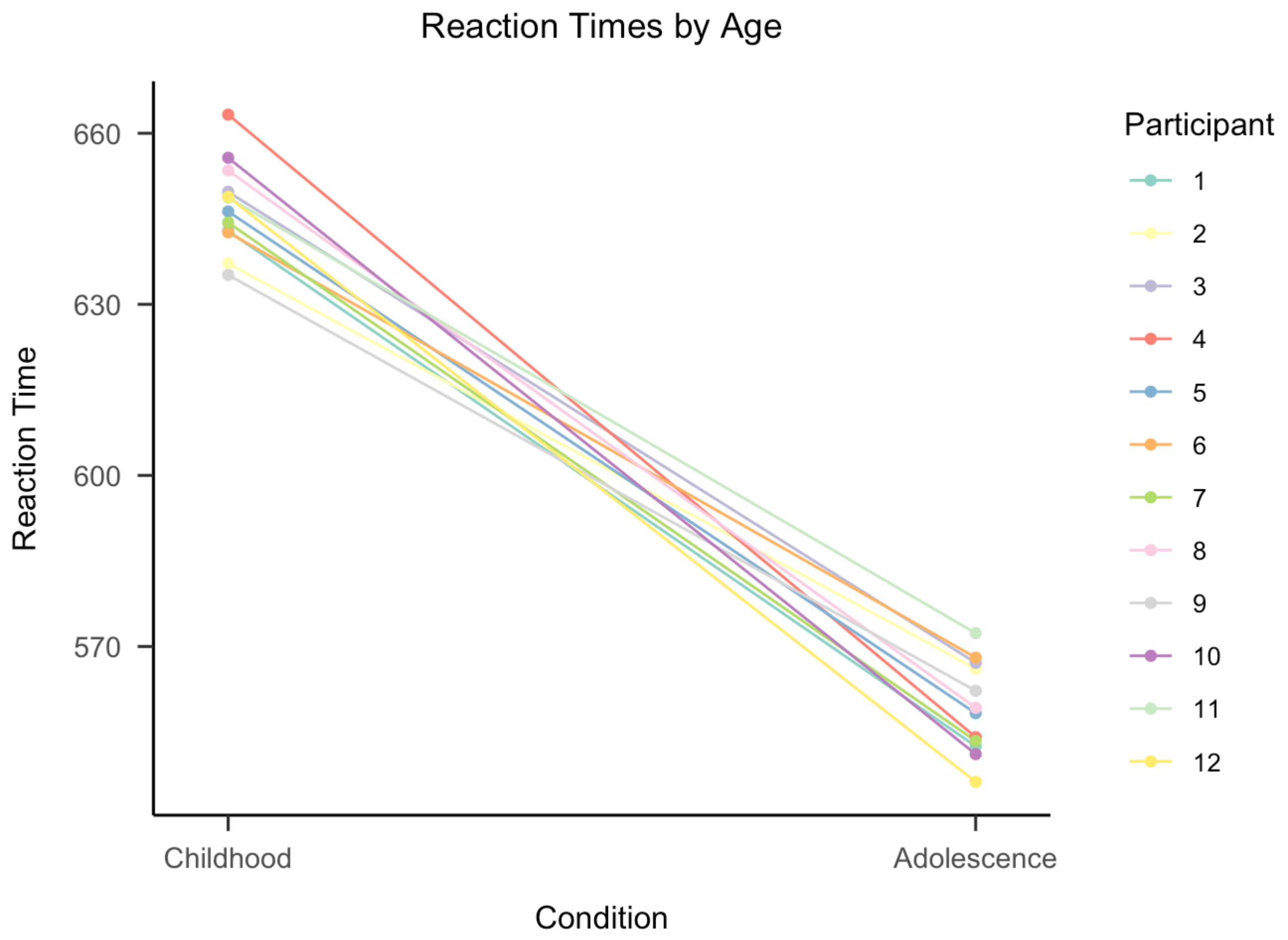


```
ggplot(adolescent_data, aes(x = rt, fill = correct)) +
  geom_histogram(binwidth = 50, position = "identity", alpha = 0.6) +
  scale_fill_manual(values = c("darkseagreen4", "darkseagreen2")) +
  theme_apa() +
  labs(x = "Time", y = "Count", fill = "Correctness")
```

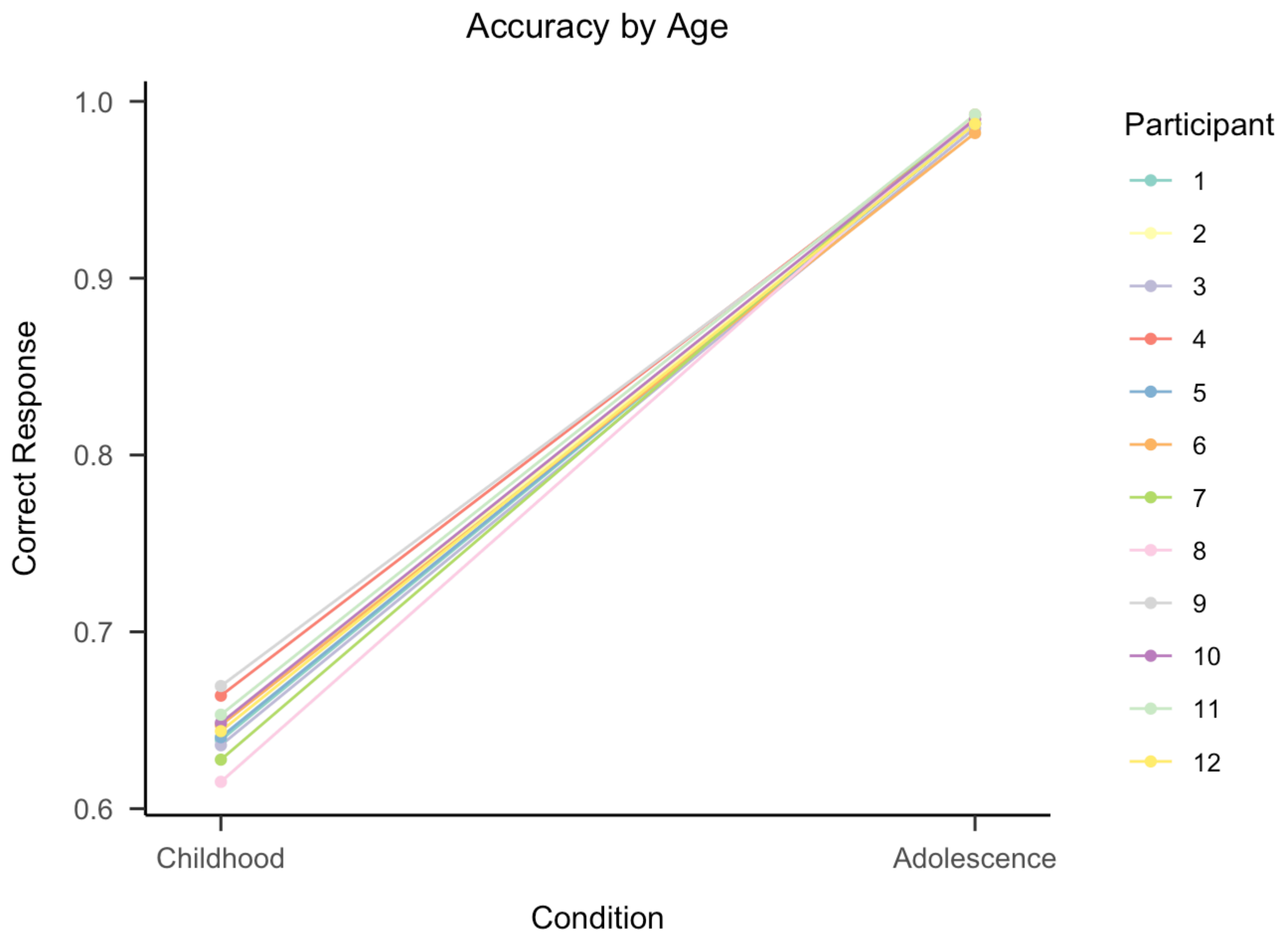


The following plots will visualize the development of the response times and accuracy values over age for each participant.

```
# Plot for Median RT
levels(df_aggregated$condition) <- c("Childhood", "Adolescence") # Name conditions fo
r plot
ggplot(df_aggregated, aes(x = condition, y = rt, group = ID, color = as.factor(ID)))
+
  geom_line() +
  geom_point() +
  labs(title = "Reaction Times by Age",
       x = "Condition",
       y = "Reaction Time",
       color = "Participant") +
  scale_x_discrete(expand = c(0, 0.1)) +
  scale_color_brewer(palette = "Set3") +
  theme_apa()
```




```
# Plot for Accuracy
ggplot(df_aggregated, aes(x = condition, y = correct, group = ID, color = as.factor(I
D))) +
  geom_line() +
  geom_point() +
  labs(title = "Accuracy by Age",
       x = "Condition",
       y = "Correct Response",
       color = "Participant") +
  scale_x_discrete(expand = c(0, 0.1)) +
  scale_color_brewer(palette = "Set3") +
  theme_apa()
```



Parameter Estimation

> Now we will turn to the actual model fitting to find the optimal DDM parameters. First, we will run the fit_data() function to find the optimal parameters. Using paired t-tests, we can find any significant differences in these parameters between the two conditons (children and adolescents).

```
N <- length(unique(data_fit$ID))
final_data <- data.frame(ID = numeric(),
                         condition = numeric(),
                         s = numeric(),
                         A = numeric(),
                         ter = numeric(),
                         b = numeric(),
                         v1 = numeric())

for (participant in 1:N) { # Loop through all participants to estimate best fitting p
arameters
  for (cond in 1:2) {
    pars <- fit_data(data_fit[data_fit$ID == participant & data_fit$condition == con
d, ])
    final_data[nrow(final_data) + 1, ] <- unlist(c(participant, cond, pars))
  }
}
```

```
# V1 - Drift Rate
v1_means <- aggregate(v1 ~ condition, data = final_data, FUN = mean)
v1_sds <- aggregate(v1 ~ condition, data = final_data, FUN = sd)
paired_t_test_v1 <- t.test(final_data$v1 ~ final_data$condition, paired = TRUE)
print(paired_t_test_v1)
```

```
##
##  Paired t-test
##
## data:  final_data$v1 by final_data$condition
## t = -5.8781, df = 11, p-value = 0.0001064
## alternative hypothesis: true mean difference is not equal to 0
## 95 percent confidence interval:
##  -0.4702261 -0.2140200
## sample estimates:
## mean difference
##      -0.3421231
```

```
# S - Drift Rate Variability
s_means <- aggregate(s ~ condition, data = final_data, FUN = mean)
s_sds <- aggregate(s ~ condition, data = final_data, FUN = sd)
paired_t_test_s <- t.test(final_data$s ~ final_data$condition, paired = TRUE)
print(paired_t_test_s)
```

```
##
##  Paired t-test
##
## data:  final_data$s by final_data$condition
## t = -2.1415, df = 11, p-value = 0.05546
## alternative hypothesis: true mean difference is not equal to 0
## 95 percent confidence interval:
##  -0.117534909  0.001610654
## sample estimates:
## mean difference
##     -0.05796213
```

```
# B - Decision Boundary
b_means <- aggregate(b ~ condition, data = final_data, FUN = mean)
b_sds <- aggregate(b ~ condition, data = final_data, FUN = sd)
paired_t_test_b <- t.test(final_data$b ~ final_data$condition, paired = TRUE)
print(paired_t_test_b)
```

```
##
##  Paired t-test
##
## data:  final_data$b by final_data$condition
## t = 0.1438, df = 11, p-value = 0.8883
## alternative hypothesis: true mean difference is not equal to 0
## 95 percent confidence interval:
##  -41.59742  47.41279
## sample estimates:
## mean difference
##        2.907687
```

```
# A - Response Bias
A_means <- aggregate(A ~ condition, data = final_data, FUN = mean)
A_sds <- aggregate(A ~ condition, data = final_data, FUN = sd)
paired_t_test_A <- t.test(final_data$A ~ final_data$condition, paired = TRUE)
print(paired_t_test_A)
```

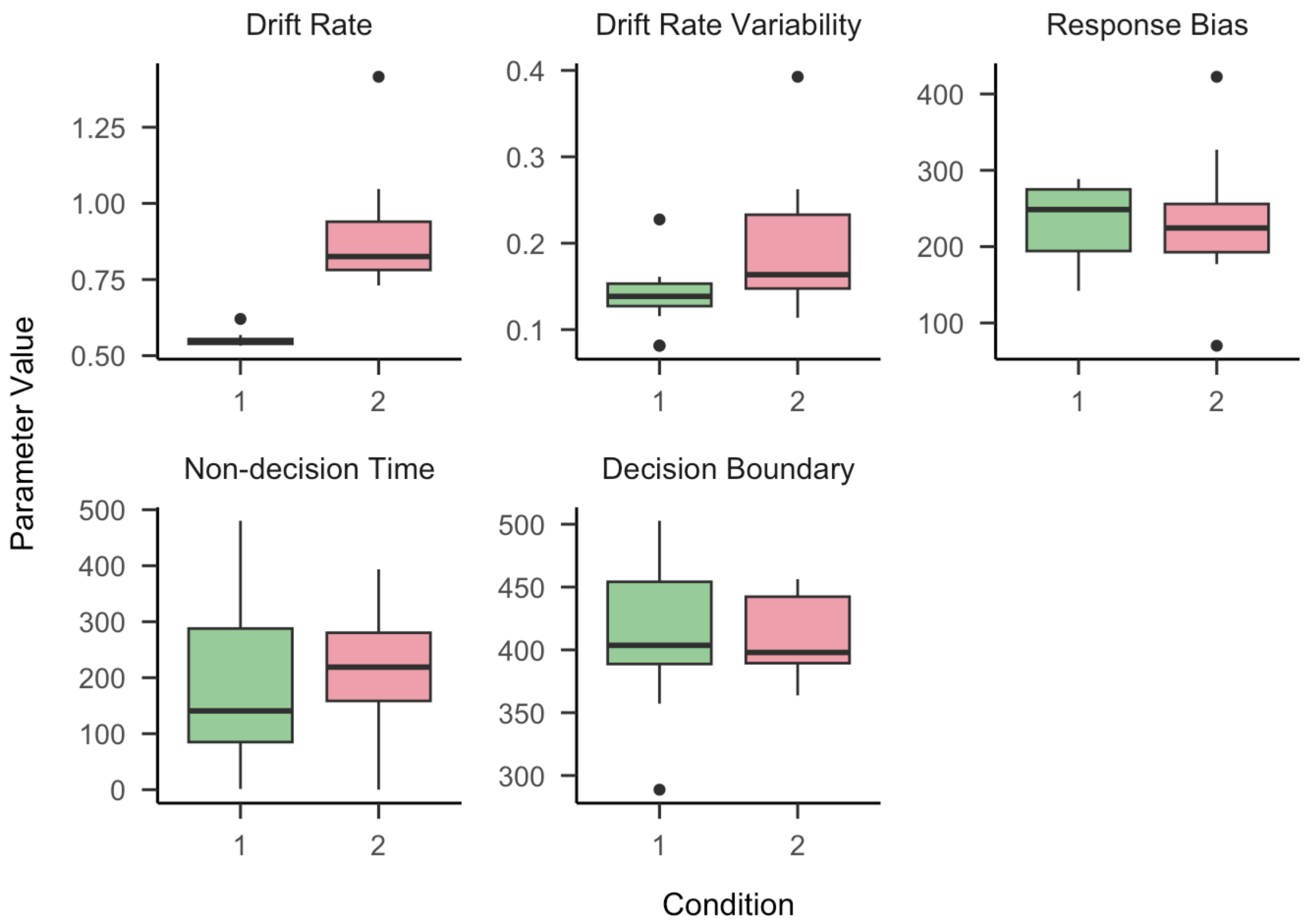
```
##
##  Paired t-test
##
## data:  final_data$A by final_data$condition
## t = -0.053195, df = 11, p-value = 0.9585
## alternative hypothesis: true mean difference is not equal to 0
## 95 percent confidence interval:
##  -69.81964  66.52437
## sample estimates:
## mean difference
##       -1.647631
```

```
# Ter - Non-decision Time
ter_means <- aggregate(ter ~ condition, data = final_data, FUN = mean)
ter_sds <- aggregate(ter ~ condition, data = final_data, FUN = sd)
paired_t_test_ter <- t.test(final_data$ter ~ final_data$condition, paired = TRUE)
print(paired_t_test_ter)
```

```
##
##  Paired t-test
##
## data:  final_data$ter by final_data$condition
## t = -0.58811, df = 11, p-value = 0.5683
## alternative hypothesis: true mean difference is not equal to 0
## 95 percent confidence interval:
##  -170.29574   98.47881
## sample estimates:
## mean difference
##       -35.90847
```

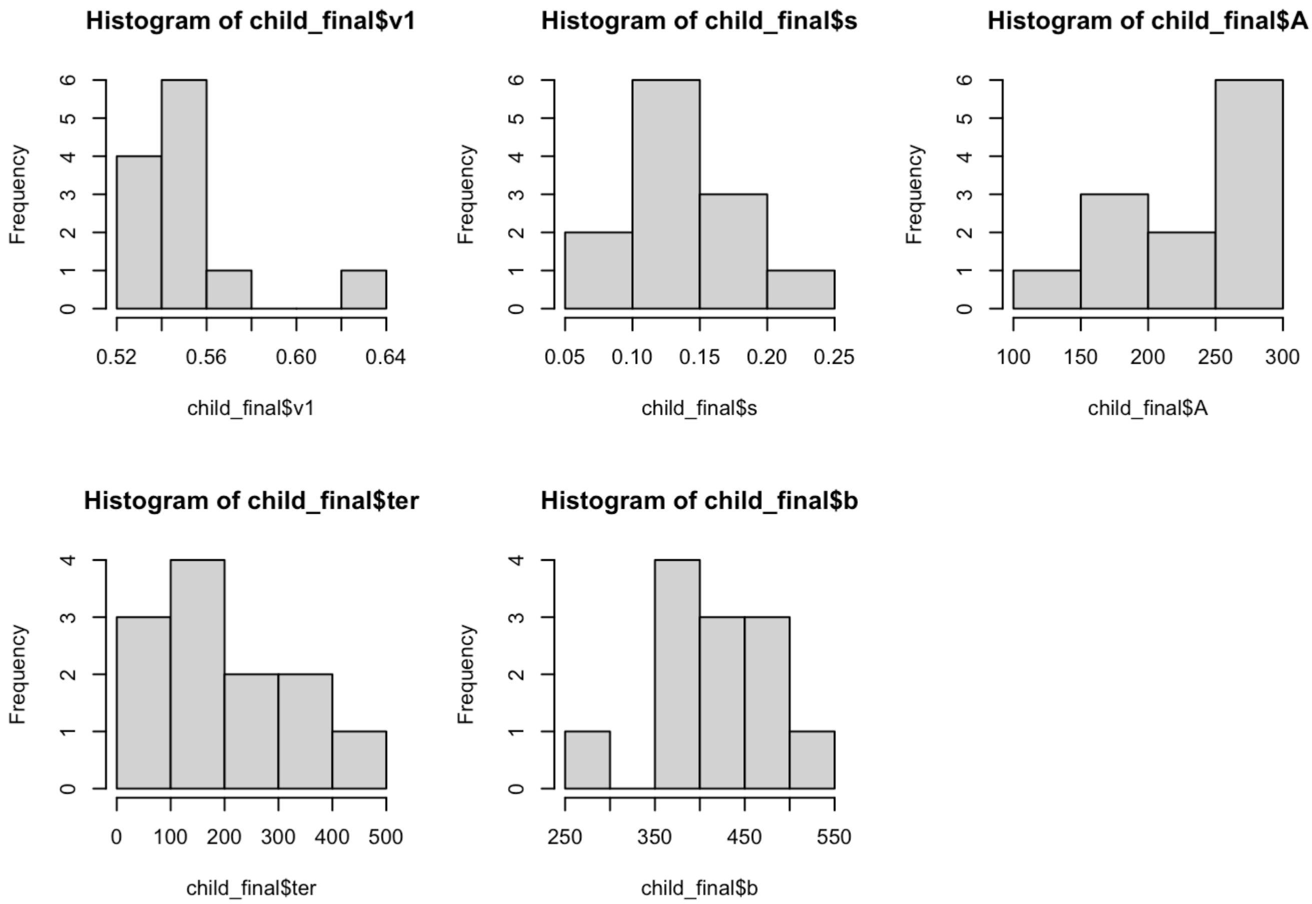
Boxplots of these Parameters

```
# Boxplot of Parameters
final_data$condition <- as.factor(final_data$condition)
final_data <- relocate(final_data, v1, .before = s) # just to show drift rates first
params_long <- melt(final_data, id.vars = c("ID", "condition"),
                    variable.name = "Parameter", value.name = "Value")

# Custom Colors and Labels
custom_colors <- c("1" = "darkseagreen3", "2" = "lightpink2")

custom_labels <- c(
  v1 = "Drift Rate",
  s = "Drift Rate Variability",
  A = "Response Bias",
  ter = "Non-decision Time",
  b = "Decision Boundary"
)
ggplot(params_long, aes(x = condition, y = Value, fill = condition)) +
  geom_boxplot() +
  facet_wrap(~ Parameter, scales = "free", labeller = as_labeller(custom_labels)) +
  scale_fill_manual(values = custom_colors) +
  labs(x = "Condition",
       y = "Parameter Value") +
  theme_apa() +
  theme(legend.position = "none")
```



In the boxplots we can see some differences and similarities between the parameters, so let's look at these values in more detail. First, let's look for any skew in the parameter distributions.

```
# Filter data per condition
child_final <- filter(final_data, condition == 1)
adolescent_final <- filter(final_data, condition == 2)
# Check the parameter value distributions
par(mfrow = c(2, 3))
hist(child_final$v1)
hist(child_final$s)
hist(child_final$A)
hist(child_final$ter)
hist(child_final$b)
```



The parameters are not normally distributed, so we will look at the median values instead of the means.

```
median(child_final$v1) # Children Drift Rate
```

```
## [1] 0.5476128
```

```
median(child_final$s) # Children Drift Rate Variability
```

```
## [1] 0.138483
```

```
median(child_final$A) # Children Bias
```

```
## [1] 248.5641
```

```
median(child_final$ter) # Children Non-response Time
```

```
## [1] 140.7549
```

```
median(child_final$b) # Children Boundary
```

```
## [1] 403.6217
```

```
median(adolescent_final$v1) # Adolescent Drift Rate
```

```
## [1] 0.8254108
```

```
median(adolescent_final$s) # Adolescent Drift Rate Variability
```

```
## [1] 0.1635729
```

```
median(adolescent_final$A) # Adolescent Bias
```

```
## [1] 224.4311
```

```
median(adolescent_final$ter) # Adolescent Non-response Time
```

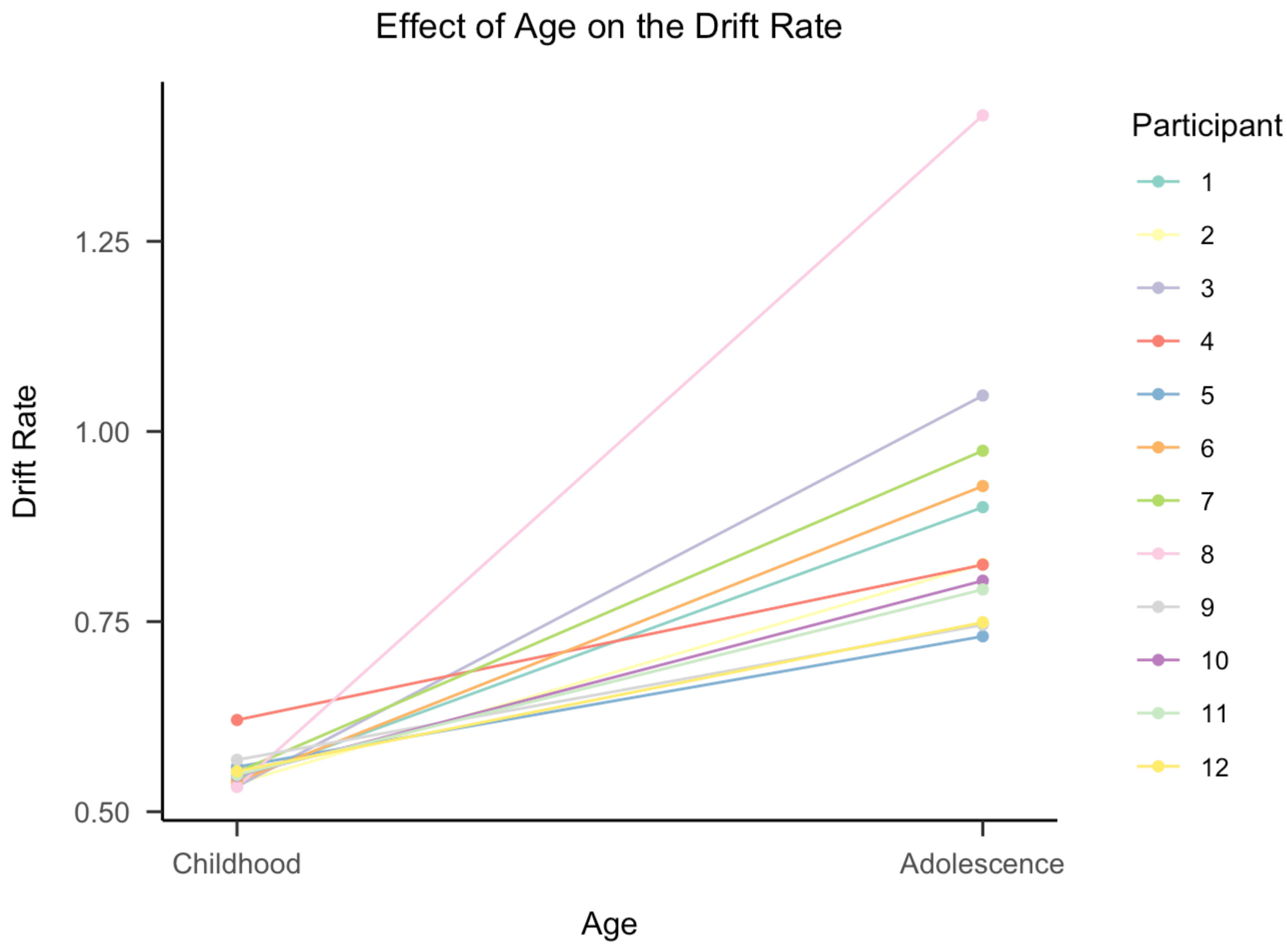
```
## [1] 218.9025
```

```
median(adolescent_final$b) # Adolescente Boundary
```

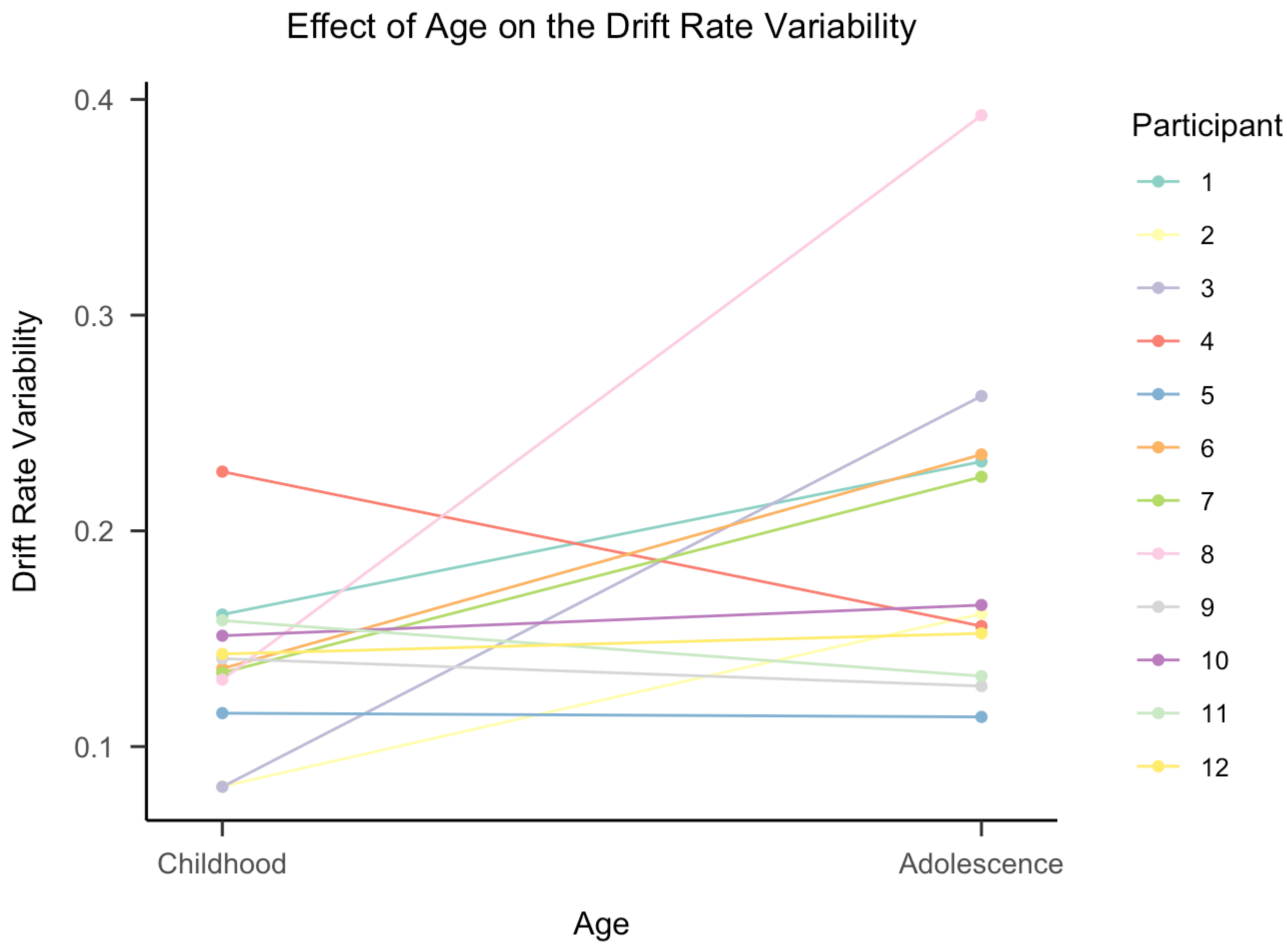
```
## [1] 397.921
```

Lastly, let's look at the slope plots of each parameter over the two conditions (childhood and adolescence) for each participant.

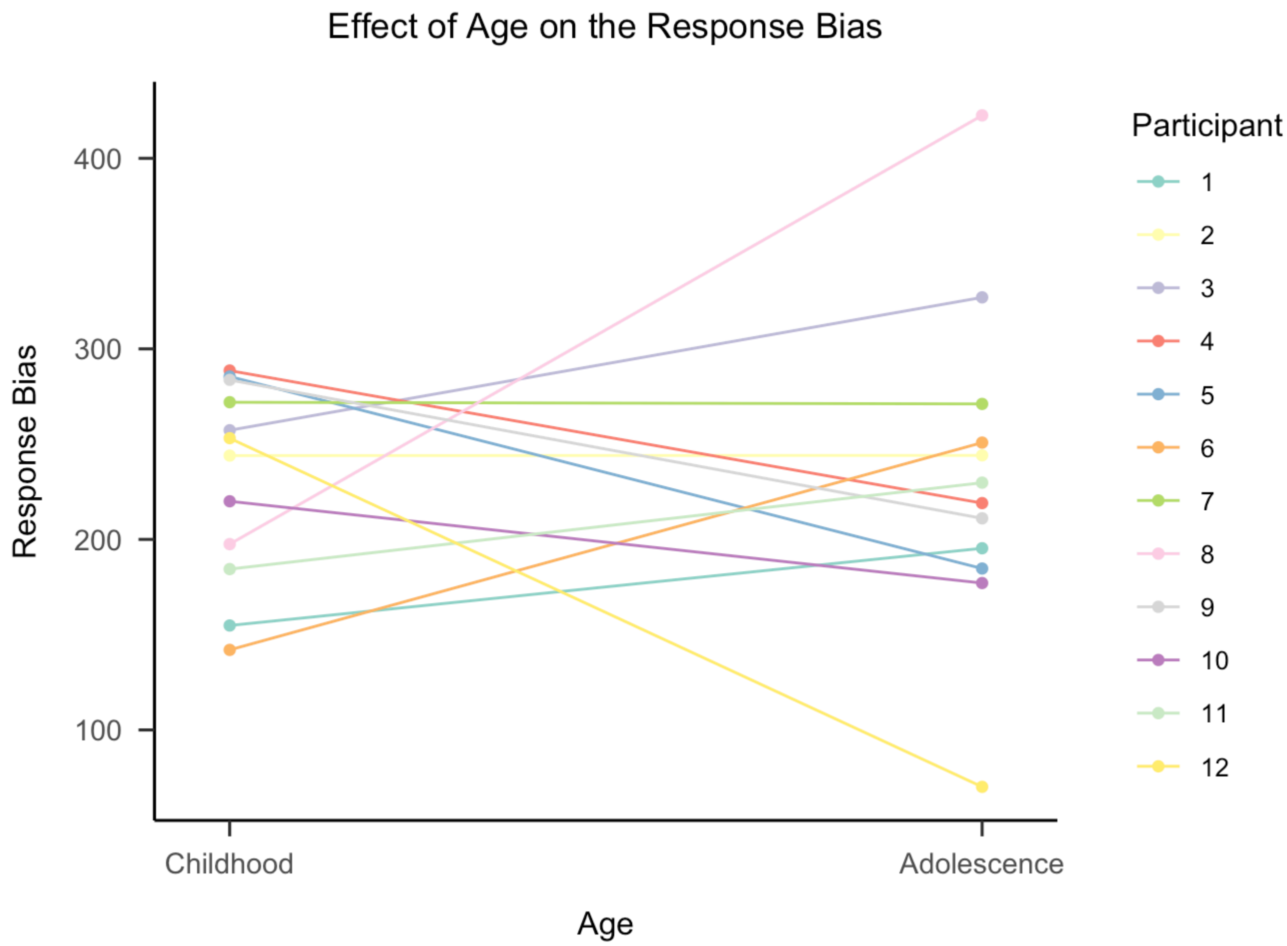
```
# Line Plot V1
levels(final_data$condition) <- c("Childhood", "Adolescence") # Name conditions for p
lot
ggplot(final_data, aes(x = condition, y = v1, group = ID, color = as.factor(ID))) +
  geom_line() +
  geom_point() +
  labs(title = "Effect of Age on the Drift Rate",
       x = "Age",
       y = "Drift Rate",
       color = "Participant") +
  scale_x_discrete(expand = c(0, 0.1)) +
  scale_color_brewer(palette = "Set3") +
  theme_apa()
```



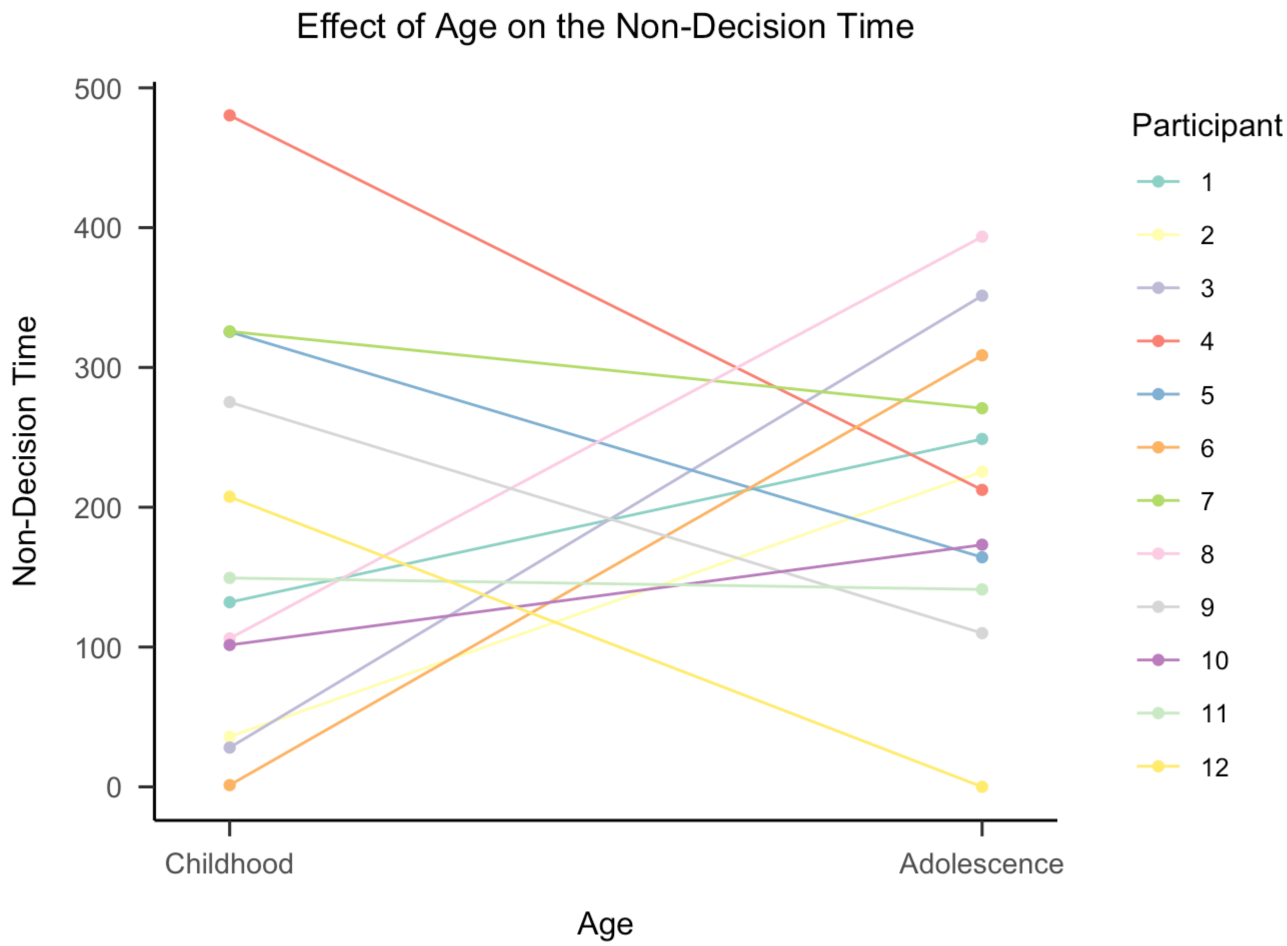

```
# Line Plot s
ggplot(final_data, aes(x = condition, y = s, group = ID, color = as.factor(ID))) +
  geom_line() +
  geom_point() +
  labs(title = "Effect of Age on the Drift Rate Variability",
       x = "Age",
       y = "Drift Rate Variability",
       color = "Participant") +
  scale_x_discrete(expand = c(0, 0.1)) +
  scale_color_brewer(palette = "Set3") +
  theme_apa()
```



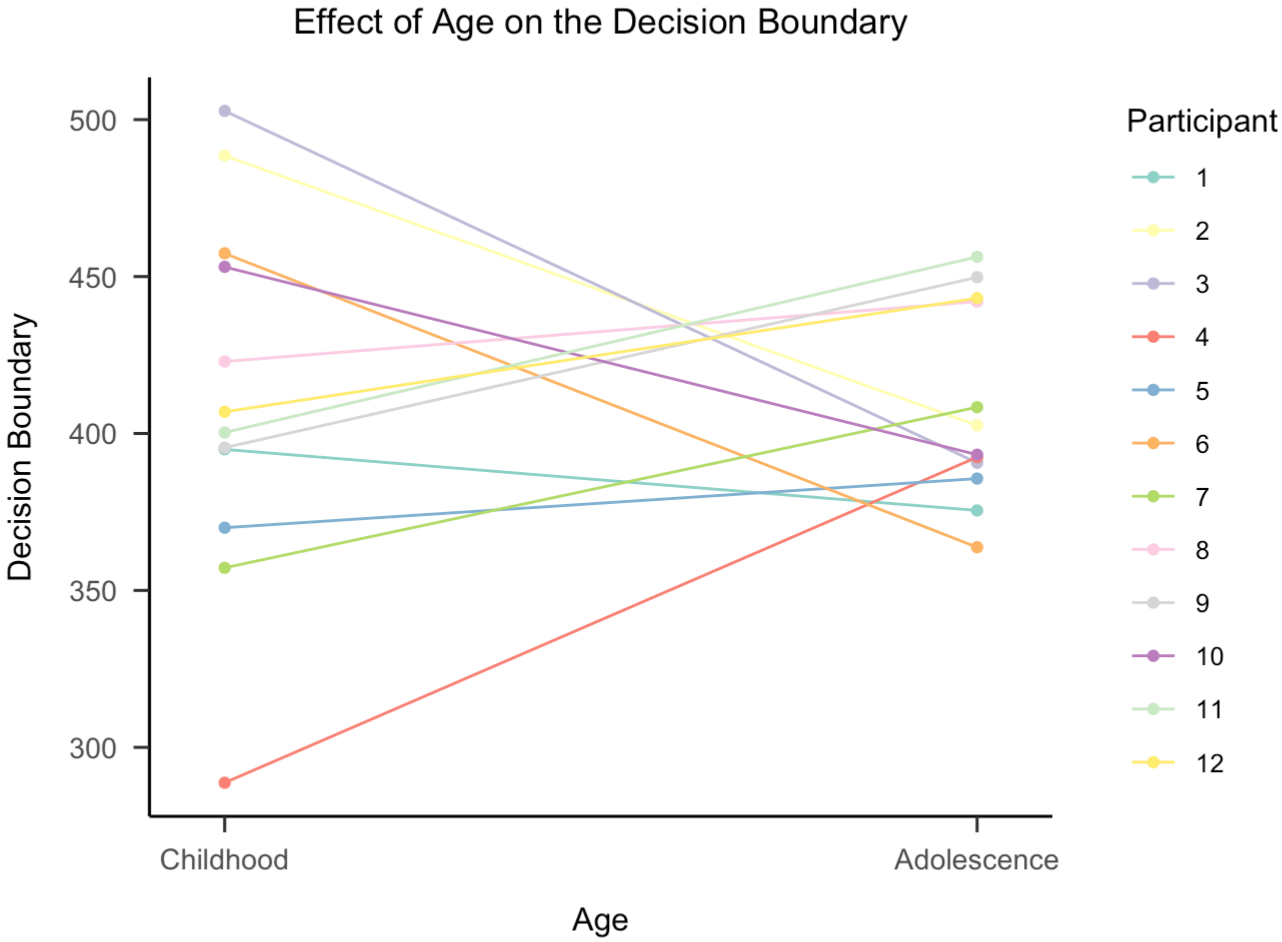

```
# Line Plot A
ggplot(final_data, aes(x = condition, y = A, group = ID, color = as.factor(ID))) +
  geom_line() +
  geom_point() +
  labs(title = "Effect of Age on the Response Bias",
       x = "Age",
       y = "Response Bias",
       color = "Participant") +
  scale_x_discrete(expand = c(0, 0.1)) +
  scale_color_brewer(palette = "Set3") +
  theme_apa()
```




```
# Line Plot ter
ggplot(final_data, aes(x = condition, y = ter, group = ID, color = as.factor(ID))) +
  geom_line() +
  geom_point() +
  labs(title = "Effect of Age on the Non-Decision Time",
       x = "Age",
       y = "Non-Decision Time",
       color = "Participant") +
  scale_x_discrete(expand = c(0, 0.1)) +
  scale_color_brewer(palette = "Set3") +
  theme_apa()
```




```
# Line Plot b
ggplot(final_data, aes(x = condition, y = b, group = ID, color = as.factor(ID))) +
  geom_line() +
  geom_point() +
  labs(title = "Effect of Age on the Decision Boundary",
       x = "Age",
       y = "Decision Boundary",
       color = "Participant") +
  scale_x_discrete(expand = c(0, 0.1)) +
  scale_color_brewer(palette = "Set3") +
  theme_apa()
```



This marks the end of the model fitting code.