OptiVisT grasping task analysis (Blind)

Florian Pätzold, Ramon Zacharias

3/21/2023

Setup

Import all necessary packages.

```
library(dplyr)
library(tidyr)
library(ggplot2)
library(sjPlot)
library(gridExtra)
library(EnvStats)
library(outliers)
library(lme4)
library(lmerTest) # significance testing for linear mixed models
```

Data preprocessing

Set working directory, load all the grasping data, clean it and combine it into one data frame.

```
# Set working directory to the folder containing the CSV files
SAVE <- paste0(getwd(), "/Plots/")
setwd("../Data/Blind")
# Get list of all CSV files for the grapsing task in the folder
file_list <- list.files(getwd(), pattern = "*grasping*")
# Delete testing data file
file_list <- file_list[file_list != "1111_grasping.csv"]
# Create an empty data frame to store the combined data
combined_data <- data.frame()
# Create inverted %in% function
`%ni%` <- Negate(`%in%`)
# Loop through each CSV file
for (file in file_list) {
  # Read the CSV file into a data frame
  file_data <- read.csv(file, header = TRUE, sep = ",")

  # Extract the number from the filename and add as a new column
  number <- as.numeric(gsub("[^0-9]+", "", file))
  file_data$participant_id <- number

  # More than 3 rep trials for each block is unfeasible and must be wrong data saving
  # Then we only take the last 8 blocks
```

```
  if (length(file_data$location) > 8 * (9+3)) {
    file_data <- file_data[tail(which(file_data$time == "time"), n=1) + 1 : length(file_data$time), ]
    file_data <- file_data[complete.cases(file_data), ]
  }

  # Cast column types from factor to numeric/char
  file_data$time <- as.numeric(as.character(file_data$time))
  file_data$num_instructions <- as.numeric(as.character(file_data$num_instructions))
  file_data$location <- as.character(file_data$location)
  file_data$block <- as.numeric(as.character(file_data$block))

  # Assign the correct block number for repetition trials in the tactile condition of P 1-6
  transform(file_data, block = as.numeric(block))
  if (number <= 6 & file_data$condition[1] == "tactile") {
    for (i in 1:length(file_data$location)) {
      if (file_data$location[i] %ni% list(1,2,3,4,5,6,7,8,9,"location")) {
        file_data$block[i] <- file_data$block[i-1] # dtype of col is factor when it should be numeric
      }
    }
  }

  # Append the data to the combined data frame
  combined_data <- rbind(combined_data, file_data)
}

# Save the combined data frame as a CSV file
row.names(combined_data) <- NULL
write.csv(combined_data, "combined_blind.csv")
blindfolded <- read.csv("../combined.csv") %>% filter(block != 1)
```

Outlier detection

```
# first block in each condition is training block
clean_data <- combined_data %>% filter(block != 1)
clean_data$num_instructions <- ifelse(clean_data$location %in% c(5), 1,
                                      ifelse(clean_data$location %in% c(2, 4, 6, 8), 2, 3))

# Grubbs' test by condition
grubbs_results <- by(clean_data$time, clean_data$condition, grubbs.test)
print(grubbs_results)
```

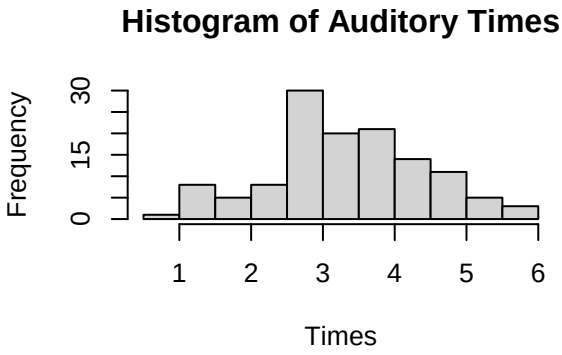
```
## clean_data$condition: auditory
##
##  Grubbs test for one outlier
##
## data:  dd[x, ]
## G = 7.49982, U = 0.56059, p-value < 2.2e-16
## alternative hypothesis: highest value 14.0887547000002 is an outlier
##
## ------------------------------------------------------------
## clean_data$condition: tactile
##
```

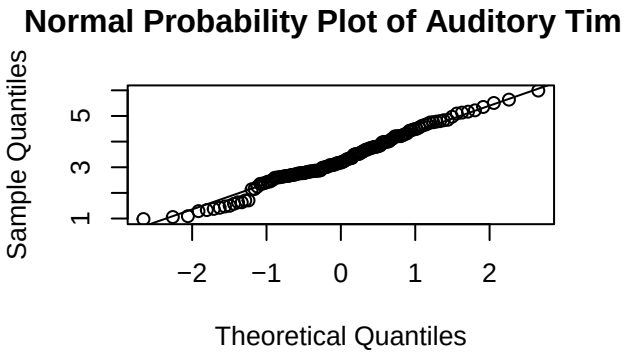
```
##  Grubbs test for one outlier
##
## data:  dd[x, ]
## G = 2.88073, U = 0.93567, p-value = 0.2271
## alternative hypothesis: highest value 8.13365479999993 is an outlier
```

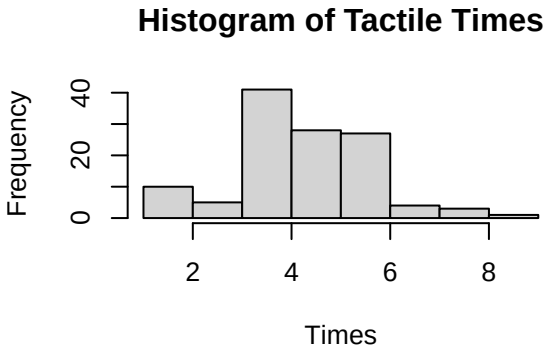
```
# clean_data <- clean_data %>% filter(time < 10.0)
```

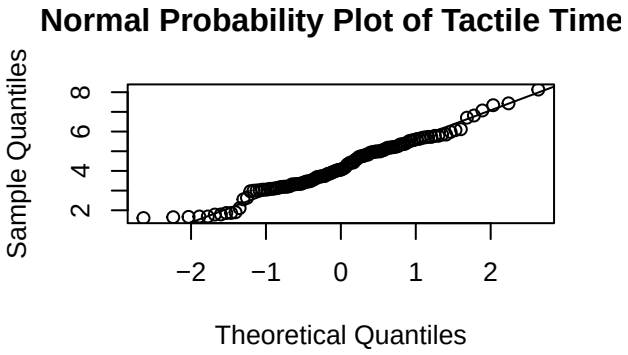
Normality assumption check

```
# Split the data by condition and drop fails
auditory_data <- clean_data %>% filter(condition == "auditory", success == "success")
tactile_data <- clean_data %>% filter(condition == "tactile", success == "success")

# Check for normality using histogram and normal probability plot for each condition
par(mfrow=c(2,2)) # create 2x2 plot grid

# Histogram and normal probability plot for auditory condition
hist(auditory_data$time, main="Histogram of Auditory Times", xlab="Times")
qqnorm(auditory_data$time, main="Normal Probability Plot of Auditory Times")
qqline(auditory_data$time)

# Histogram and normal probability plot for tactile condition
hist(tactile_data$time, main="Histogram of Tactile Times", xlab="Times")
qqnorm(tactile_data$time, main="Normal Probability Plot of Tactile Times")
qqline(tactile_data$time)
```

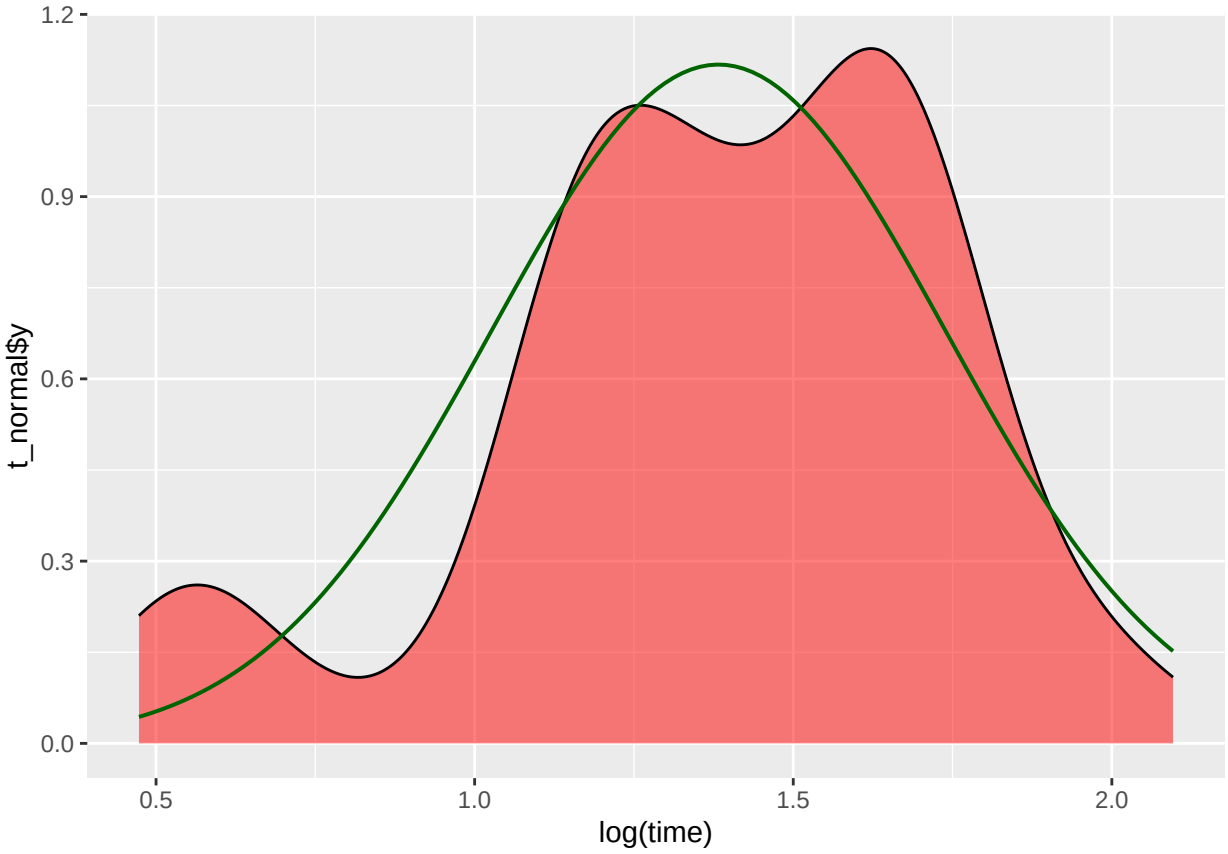




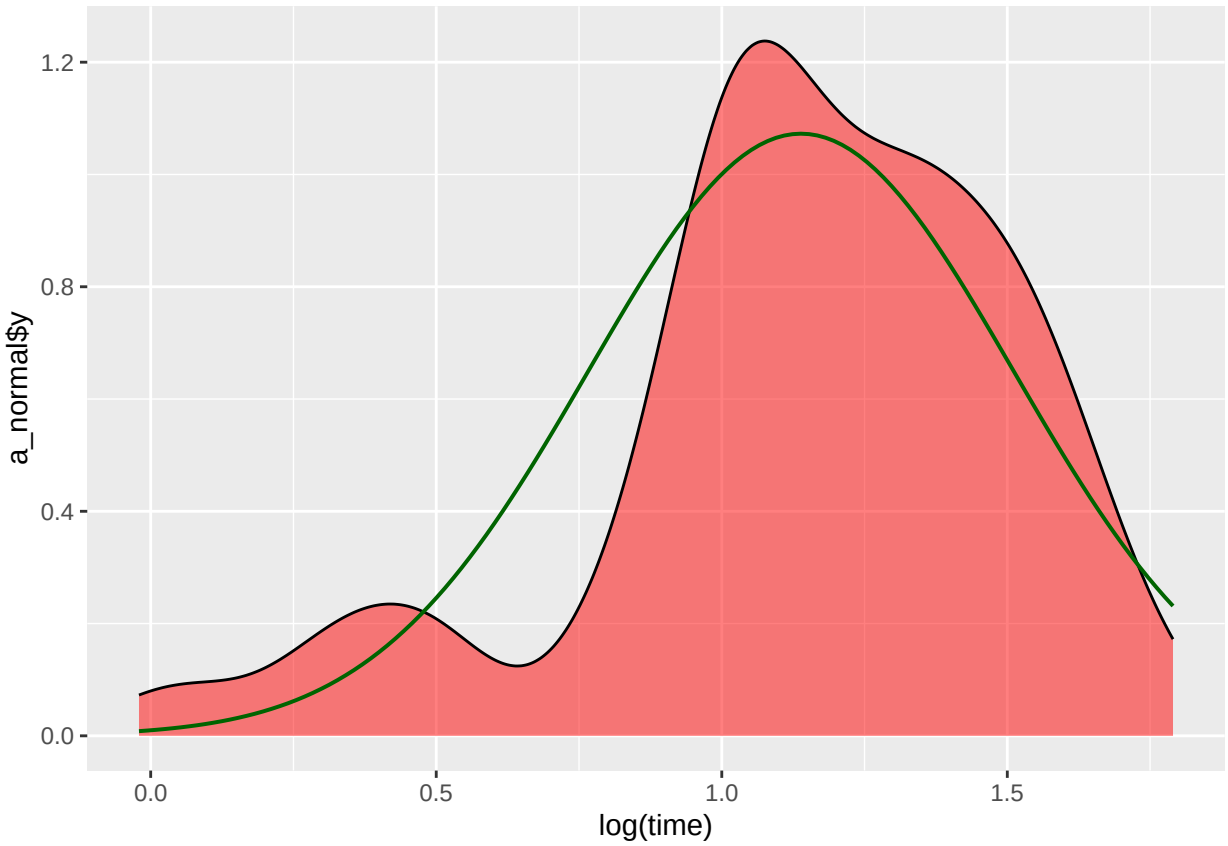




```r
# normal distribution
x <- seq(min(log(tactile_data$time)), max(log(tactile_data$time)), length.out = length(tactile_data$time
y <- dnorm(x, mean = mean(log(tactile_data$time)), sd = sd(log(tactile_data$time)))
t_normal <- data.frame(x = x, y = y)

x <- seq(min(log(auditory_data$time)), max(log(auditory_data$time)), length.out = length(auditory_data$
y <- dnorm(x, mean = mean(log(auditory_data$time)), sd = sd(log(auditory_data$time)))
a_normal <- data.frame(x = x, y = y)

# plot density comparisons for t and a (justification for parametric LMM)
ggplot(tactile_data) +
  geom_density(aes(x = log(time)), fill = "red", alpha = 0.5) +
  geom_line(aes(x = t_normal$x, y = t_normal$y), size = 0.7, color = "darkgreen")
```



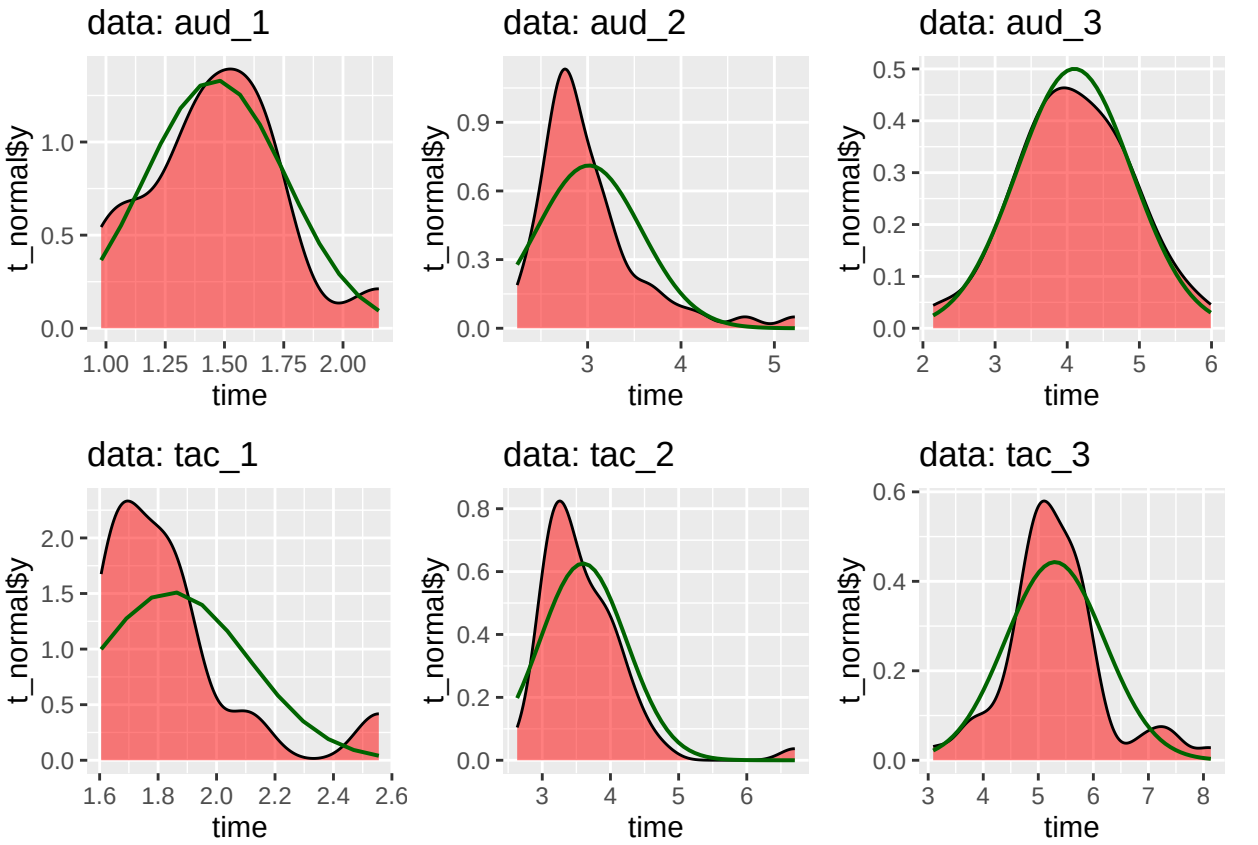

```r
ggplot(auditory_data) +
  geom_density(aes(x = log(time)), fill = "red", alpha = 0.5) +
  geom_line(aes(x = a_normal$x, y = a_normal$y), size = 0.7, color = "darkgreen")
```



Looks multi-modal because of different number of instructions.

```r
# Assumption: tri-modality of the data for different number of commands
aud_1 <- clean_data %>% filter(condition == "auditory", success == "success", num_instructions == 1)
aud_2 <- clean_data %>% filter(condition == "auditory", success == "success", num_instructions == 2)
aud_3 <- clean_data %>% filter(condition == "auditory", success == "success", num_instructions == 3)

tac_1 <- clean_data %>% filter(condition == "tactile", success == "success", num_instructions == 1)
tac_2 <- clean_data %>% filter(condition == "tactile", success == "success", num_instructions == 2)
tac_3 <- clean_data %>% filter(condition == "tactile", success == "success", num_instructions == 3)

# plot density comparisons filtered by number of commands
compare_to_normal <- function(data) {
  x <- seq(min(data$time), max(data$time), length.out = length(data$time))
  y <- dnorm(x, mean = mean(data$time), sd = sd(data$time))
  t_normal <- data.frame(x = x, y = y)

  p <- ggplot(data) +
    geom_density(aes(x = time), fill = "red", alpha = 0.5) +
    geom_line(aes(x = t_normal$x, y = t_normal$y), size = 0.7, color = "darkgreen") +
    ggtitle(sprintf("data: %s", deparse(substitute(data))))

  return(p)
}

aud1 <- compare_to_normal(aud_1)
aud2 <- compare_to_normal(aud_2)
```

```
aud3 <- compare_to_normal(aud_3)

tac1 <- compare_to_normal(tac_1)
tac2 <- compare_to_normal(tac_2)
tac3 <- compare_to_normal(tac_3)

grid.arrange(aud1,aud2,aud3,tac1,tac2,tac3, ncol = 3)
```




```
#ggsave(paste0(SAVE,"normality_blind.jpeg"), plot = grid, dpi = 600)
```

Statistical tests suggest non-normality, visually they are normally distributed. The t-test and Wilcoxon rank sum test yield the same results in all comparisons, so we go with the student's t-test and assume that the data stems from a normal distribution.

Data visualisation and hypothesis testing

Calculate summary statistics, visualize aspects of the data and test the corresponding hypotheses.

Trial times per participant

```
# time standard devs
clean_data %>% group_by(condition) %>% summarize(mean = mean(time), sd = sd(time))
```

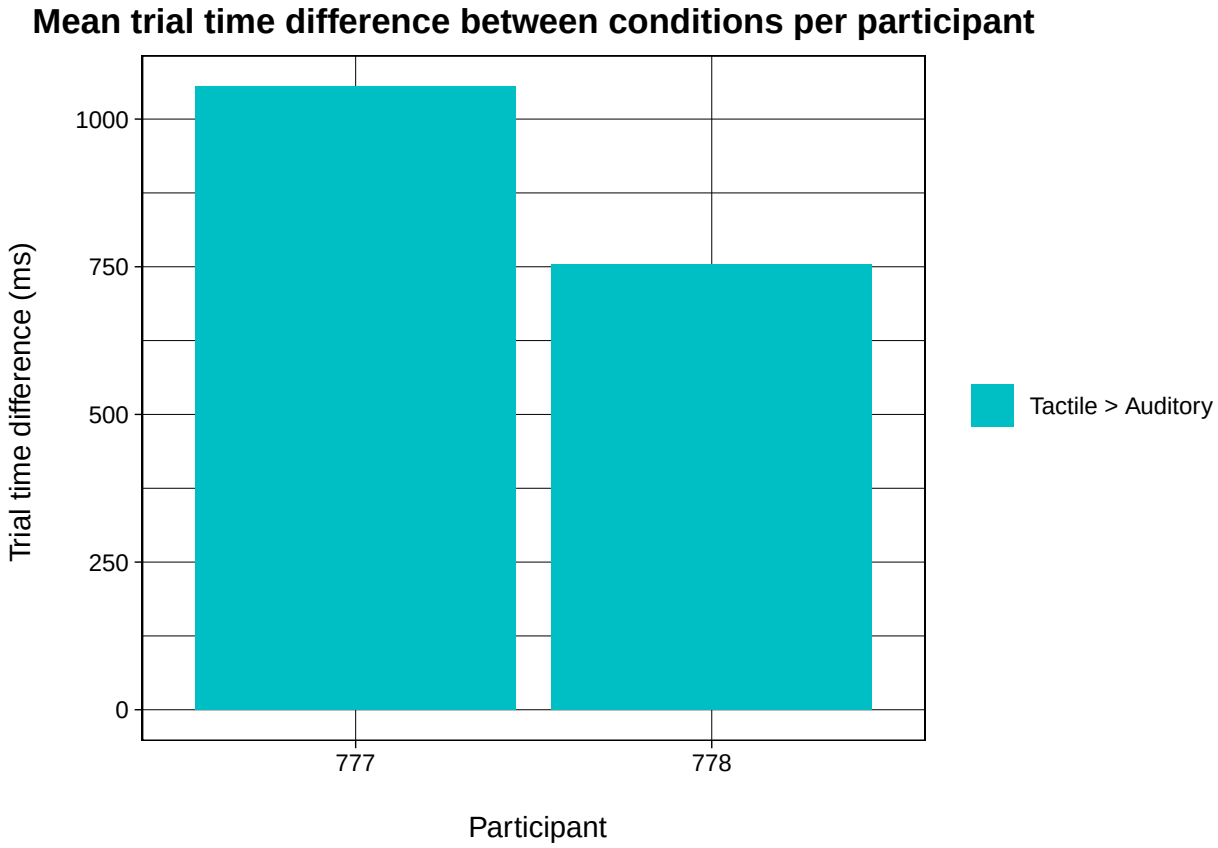
```
## # A tibble: 2 x 3
##   condition  mean    sd
```

```
##   <chr>     <dbl> <dbl>
## 1 auditory   3.41  1.42
## 2 tactile    4.21  1.36
```

```
combined_data %>% group_by(condition, block) %>% summarize(mean = mean(time), sd = sd(time))
```

```
## # A tibble: 16 x 4
## # Groups:   condition [2]
##    condition block  mean    sd
##    <chr>     <dbl> <dbl> <dbl>
##  1 auditory      1  5.81 3.08
##  2 auditory      2  3.20 1.50
##  3 auditory      3  2.94 1.13
##  4 auditory      4  3.50 0.963
##  5 auditory      5  3.90 2.61
##  6 auditory      6  3.53 0.994
##  7 auditory      7  3.61 1.15
##  8 auditory      8  3.18 0.777
##  9 tactile       1  3.99 1.58
## 10 tactile       2  4.28 1.54
## 11 tactile       3  4.36 1.44
## 12 tactile       4  4.26 1.09
## 13 tactile       5  4.44 1.71
## 14 tactile       6  4.13 1.26
## 15 tactile       7  4.12 0.980
## 16 tactile       8  3.90 1.52
```

```
times_per_participant <- clean_data %>%
  filter(success == "success") %>%
  group_by(participant_id, condition) %>%
  summarize(mean_time = mean(time) * 1000) %>%
  spread(condition, mean_time) %>%
  mutate(diff = tactile - auditory, color = ifelse(diff > 0, "#00BFC4", "#F8766D")) %>%
  ggplot(aes(x = factor(participant_id), y = diff, fill = color)) +
  geom_bar(stat = "identity", position = "identity") +
  labs(
    title = "Mean trial time difference between conditions per participant",
    x = "\n Participant",
    y = "Trial time difference (ms) \n") +
  theme_linedraw() +
  theme(
    plot.title = element_text(face = "bold", hjust = 0.5),
    plot.subtitle = element_text(hjust = 0.5),
    legend.position = "right") +
  scale_fill_manual(name=NULL, values = c("#00BFC4", "#F8766D"), labels = c("Tactile > Auditory", "Audi

times_per_participant
```



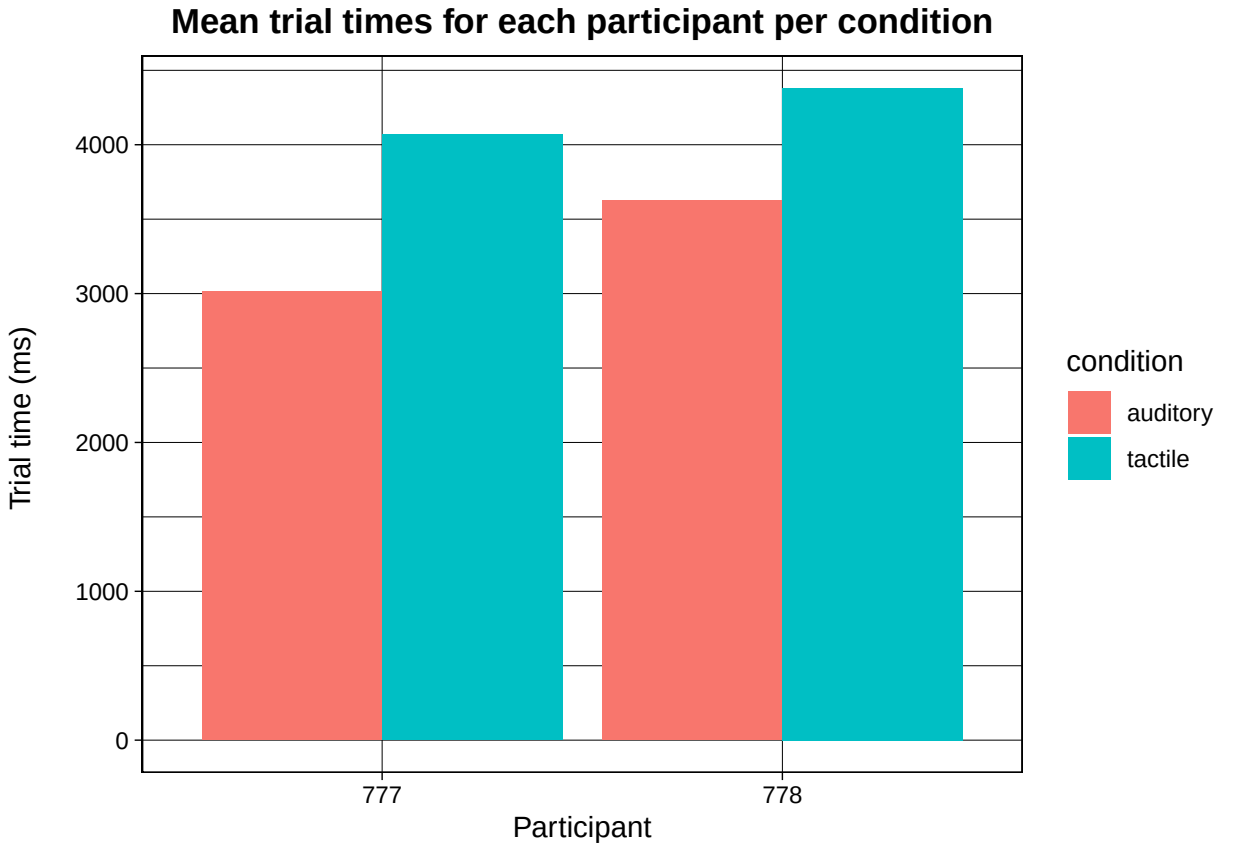

```
#ggsave(paste0(SAVE,"times_diff_per_participant_blind.jpeg"), plot = times_per_participant, dpi = 600)
```

```
# Barplot: Mean trial times for each condition per participant
clean_data %>% filter(success == "success") %>% group_by(participant_id, condition) %>% summarize(mean_
```

```
## # A tibble: 4 x 3
## # Groups:   participant_id [2]
##   participant_id condition mean_time
##            <dbl> <chr>         <dbl>
## 1            777 auditory      3015.
## 2            777 tactile       4070.
## 3            778 auditory      3626.
## 4            778 tactile       4380.
```

```
times_per_participant <- clean_data %>%
  filter(success == "success") %>%
  group_by(participant_id, condition) %>%
  summarize(mean_time = mean(time)*1000) %>%
  ggplot(aes(x = factor(participant_id), y = mean_time, fill = condition)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(
    title = "Mean trial times for each participant per condition",
    x = "Participant",
    y = "Trial time (ms) \n",
    fill = "condition"
    ) +
  theme_linedraw() +
  theme(plot.title = element_text(face = "bold", hjust = 0.5),
```

```
    plot.subtitle = element_text(hjust = 0.5),
    legend.position = "right")

times_per_participant
```




```
#ggsave(paste0(SAVE,"times_per_participant_blind.jpeg"), plot = times_per_participant, dpi = 600)
```

Trial times per condition

```
blindfolded_violin <- blindfolded %>% filter(success == "success") %>% select(-1)
blindfolded_violin$blindedness <- "Blindfolded"
blindfolded_violin %>% summarize(mean_time = mean(time)*1000)
```

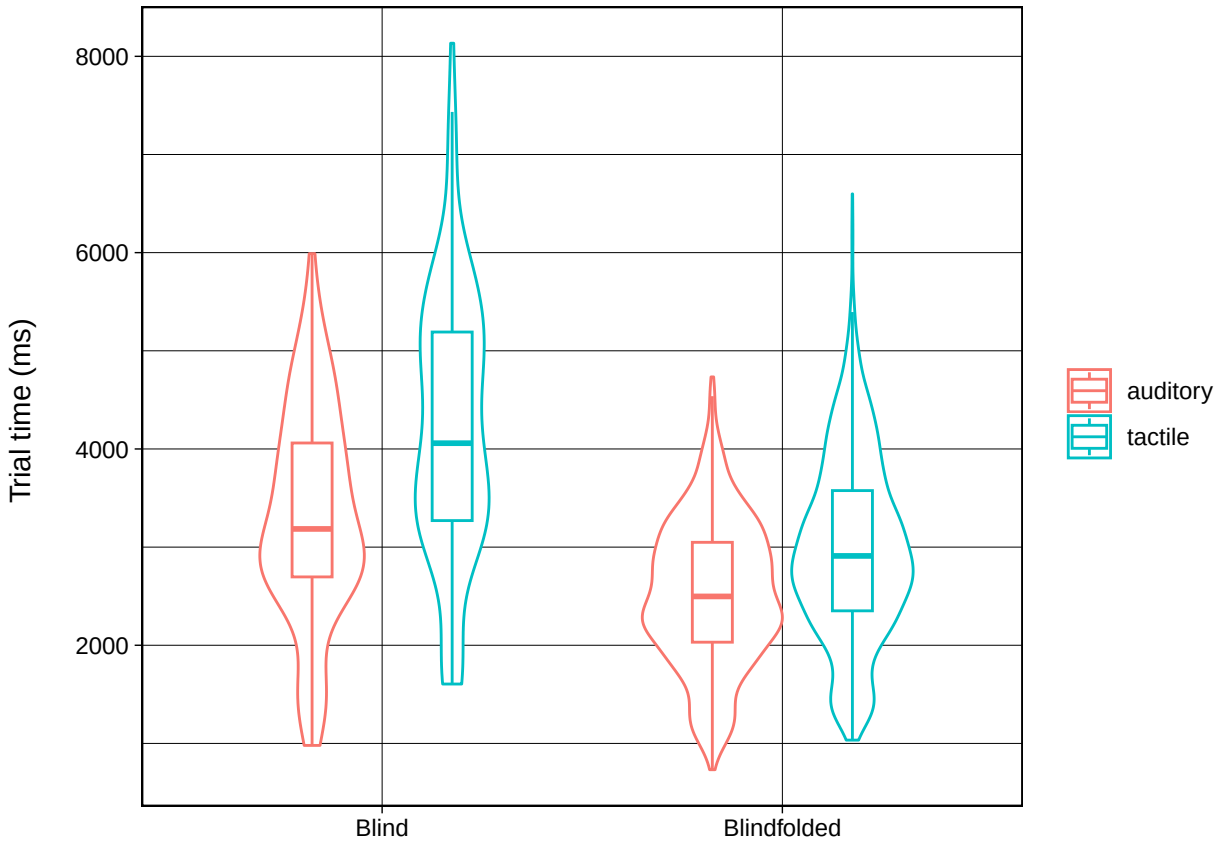
```
##   mean_time
## 1  2743.768
```

```
blind_violin <- clean_data %>% filter(success == "success")
blind_violin$blindedness <- "Blind"
blind_violin %>% summarize(mean_time = mean(time)*1000)
```

```
##   mean_time
## 1  3758.982
```

```
violins <- rbind(blindfolded_violin, blind_violin)

# Violin plot: x = condition, y = RT
times_condition_violin <- violins %>%
```

```
  ggplot(aes(x = blindedness, y = time*1000, color=condition)) +
  geom_violin(width=0.7) +
  geom_boxplot(outlier.shape = NA, width=0.2, position = position_dodge(width = 0.7)) +
  labs(
    #title = "Distribution of trial times per condition",
    x = NULL,
    y = "Trial time (ms) \n"
    ) +
  theme_linedraw() +
  theme(plot.title = element_text(face = "bold"), legend.position = "right", legend.title = element_bla
  #scale_y_continuous(limits = c(0, 6000)) # remove outliers from plot

times_condition_violin
```




```
#ggsave(paste0(SAVE,"times_condition_violin_blind.jpeg"), plot = times_condition_violin, dpi = 600)


# Perform t-test (with unpaired data as fail trials are excluded) test to compare group means
t.test(auditory_data$time, tactile_data$time, paired = FALSE)
```

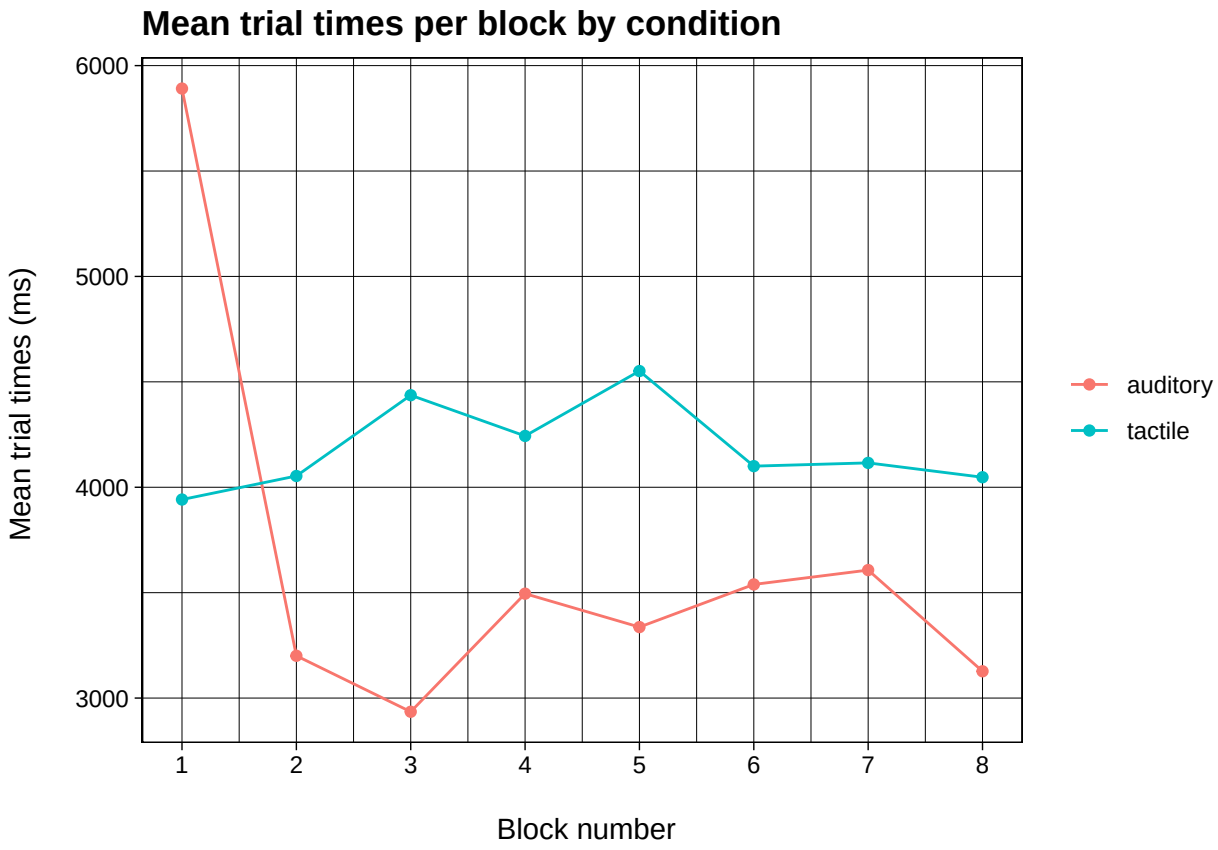
```
##
##  Welch Two Sample t-test
##
## data:  auditory_data$time and tactile_data$time
## t = -5.7412, df = 224.69, p-value = 3.028e-08
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
```

```
##  -1.2136109 -0.5933898
## sample estimates:
## mean of x mean of y
##  3.320139  4.223639
```

```
learning_data <- combined_data %>% filter(success == "success")

# Line plot: x = block (grouped by condition --> 2 lines), y = median/mean RT
learning_data %>% group_by(block, condition) %>% summarize(mean_time = mean(time)*1000)
```

```
## # A tibble: 16 x 3
## # Groups:   block [8]
##    block condition mean_time
##    <dbl> <chr>         <dbl>
##  1     1 auditory      5891.
##  2     1 tactile       3942.
##  3     2 auditory      3201.
##  4     2 tactile       4053.
##  5     3 auditory      2935.
##  6     3 tactile       4437.
##  7     4 auditory      3495.
##  8     4 tactile       4243.
##  9     5 auditory      3337.
## 10     5 tactile       4551.
## 11     6 auditory      3539.
## 12     6 tactile       4100.
## 13     7 auditory      3607.
## 14     7 tactile       4116.
## 15     8 auditory      3127.
## 16     8 tactile       4047.
```

```
times_per_block <- learning_data %>% group_by(block, condition) %>% summarize(mean_time = mean(time)*10
  ggplot(aes(x = block, y = mean_time, color=condition)) +
  geom_point() +
  geom_line() +
  labs(
    title = "Mean trial times per block by condition",
    x = "\n Block number",
    y = "Mean trial times (ms) \n"
    ) +
  theme_linedraw() +
  theme(plot.title = element_text(face = "bold"), legend.position = "right", legend.title = element_bla
  scale_x_continuous(breaks = c(1:8)) #+
  #scale_y_continuous(limits = c(2000, 6000))

times_per_block
```




```
#ggsave(paste0(SAVE,"times_per_block_blind.jpeg"), plot = times_per_block, dpi = 600)

# test diffs between blocks
# tactile
t.test(filter(learning_data, block == 1, condition == "tactile")$time,
            filter(learning_data, block == 2, condition == "tactile")$time, paired = FALSE) # ***
```

```
##
##  Welch Two Sample t-test
##
## data:  filter(learning_data, block == 1, condition == "tactile")$time and filter(learning_data, bloc
## t = -0.21256, df = 30.751, p-value = 0.8331
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.1843256  0.9608297
## sample estimates:
## mean of x mean of y
##  3.941707  4.053455
```

```
t.test(filter(learning_data, block == 3, condition == "tactile")$time,
            filter(learning_data, block == 4, condition == "tactile")$time, paired = FALSE) # *
```

```
##
##  Welch Two Sample t-test
##
## data:  filter(learning_data, block == 3, condition == "tactile")$time and filter(learning_data, bloc
## t = 0.4401, df = 30.471, p-value = 0.663
## alternative hypothesis: true difference in means is not equal to 0
```

```
## 95 percent confidence interval:
##  -0.7025944  1.0889061
## sample estimates:
## mean of x mean of y
##  4.436524  4.243368
```

```
# auditory
t.test(filter(learning_data, block == 1, condition == "auditory")$time,
            filter(learning_data, block == 2, condition == "auditory")$time, paired = FALSE) # ***
```

```
##
##  Welch Two Sample t-test
##
## data:  filter(learning_data, block == 1, condition == "auditory")$time and filter(learning_data, bloc
## t = 3.2746, df = 24.387, p-value = 0.00316
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.996170 4.384861
## sample estimates:
## mean of x mean of y
##  5.891105  3.200589
```

```
# other comparisons yielded no significant differences
t.test(filter(learning_data, block == 4, condition == "auditory")$time,
            filter(learning_data, block == 5, condition == "auditory")$time, paired = FALSE)
```

```
##
##  Welch Two Sample t-test
##
## data:  filter(learning_data, block == 4, condition == "auditory")$time and filter(learning_data, bloc
## t = 0.51917, df = 33.621, p-value = 0.607
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.4622125  0.7792232
## sample estimates:
## mean of x mean of y
##  3.495264  3.336758
```

```
t.test(filter(learning_data, block == 4, condition == "tactile")$time,
            filter(learning_data, block == 5, condition == "tactile")$time, paired = FALSE)
```

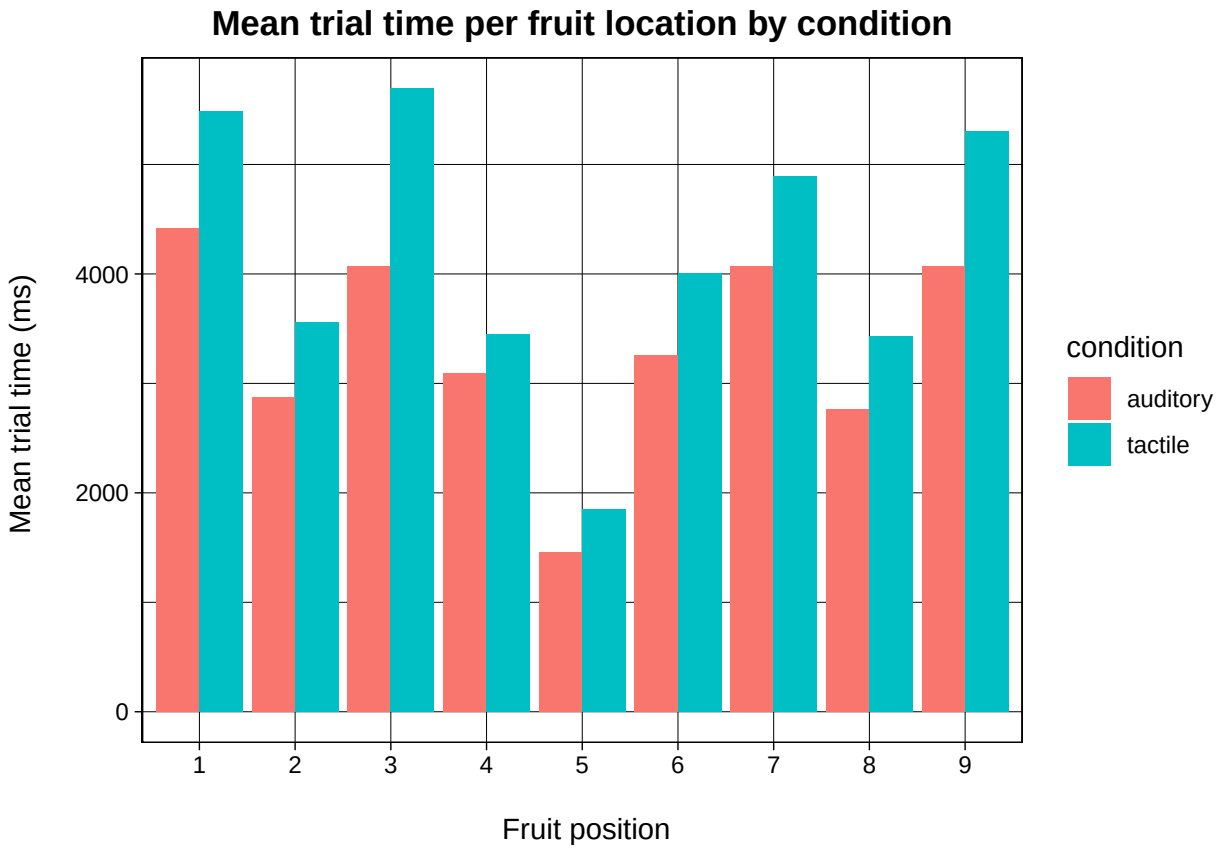
```
##
##  Welch Two Sample t-test
##
## data:  filter(learning_data, block == 4, condition == "tactile")$time and filter(learning_data, block
## t = -0.62984, df = 26.865, p-value = 0.5341
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.3109710  0.6952682
## sample estimates:
## mean of x mean of y
##  4.243368  4.551219
```

```
# Barplot: y = RT, x = fruit position
pos_data <- clean_data # dummy
pos_data$location[pos_data$location == "rep_1"] = "1"
pos_data$location[pos_data$location == "rep_2"] = "2"
```

```r
pos_data$location[pos_data$location == "rep_3"] = "3"
pos_data$location[pos_data$location == "rep_4"] = "4"
pos_data$location[pos_data$location == "rep_5"] = "5"
pos_data$location[pos_data$location == "rep_6"] = "6"
pos_data$location[pos_data$location == "rep_7"] = "7"
pos_data$location[pos_data$location == "rep_8"] = "8"
pos_data$location[pos_data$location == "rep_9"] = "9"

pos_data %>% filter(success == "success") %>% group_by(condition, location) %>% summarize(mean_time = me
```

```
## # A tibble: 18 x 3
## # Groups:   condition [2]
##    condition location mean_time
##    <chr>     <chr>        <dbl>
##  1 auditory  1            4417.
##  2 auditory  2            2874.
##  3 auditory  3            4070.
##  4 auditory  4            3093.
##  5 auditory  5            1461.
##  6 auditory  6            3260.
##  7 auditory  7            4073.
##  8 auditory  8            2759.
##  9 auditory  9            4073.
## 10 tactile   1            5487.
## 11 tactile   2            3556.
## 12 tactile   3            5694.
## 13 tactile   4            3449.
## 14 tactile   5            1846.
## 15 tactile   6            4005.
## 16 tactile   7            4892.
## 17 tactile   8            3432.
## 18 tactile   9            5305.
```

```r
pos_data %>% filter(success == "success") %>% group_by(condition, location) %>% summarize(mean_time = me
  ggplot(aes(x = location, y = mean_time, fill = condition)) +
  geom_bar(stat="identity", position = "dodge") +
  labs(
    title = "Mean trial time per fruit location by condition",
    x = "\n Fruit position",
    y = "Mean trial time (ms) \n"
    ) +
  theme_linedraw() +
  theme(plot.title = element_text(face = "bold", hjust = 0.5),
    plot.subtitle = element_text(hjust = 0.5),
    legend.position = "right")
```

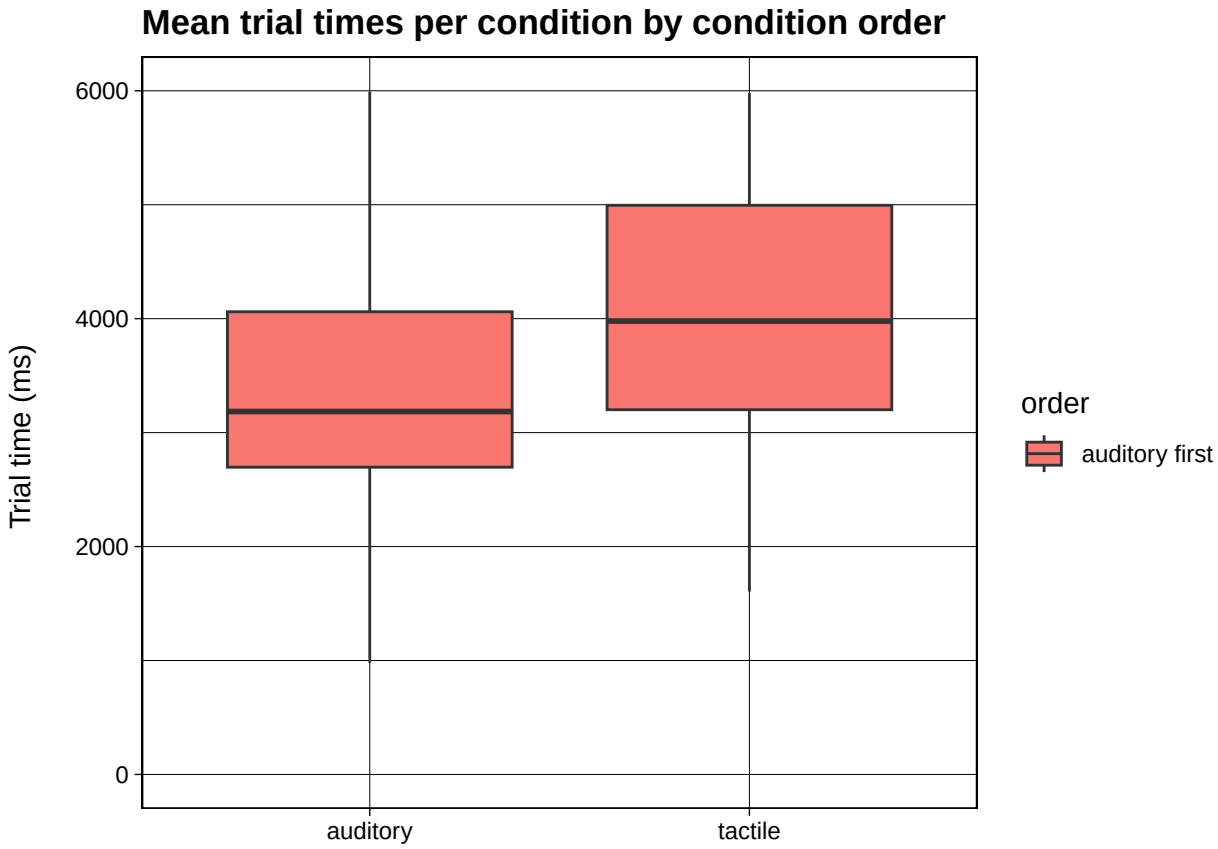



```
# mid (5) fastest as starting point is in front
# one-command positions up (2), left (4), right (6), down (8) are shorter than two-command positions
# diff between conditions for positions 1 and 3 (upper left and right) greater than for 7 and 9 (lower
# --> grasping at top shelf took longer, probably the chair was too low

# Could add test for each fruit position comparing conditions
# Could add comparison between upper left (1) and upper right (3) and/or lower left (7) and lower right
```

```
# Subset data by order
data_afirst <- clean_data[clean_data$participant_id == 777
                          | clean_data$participant_id == 778, ]


if (length(data_afirst != 0)) {
  data_afirst$order <- "auditory_tactile"
}

# Boxplot: x = condition, y = trial time, grouped by order
data_afirst %>% filter (success == "success") %>% group_by(condition, order) %>% summarize(mean_time = m
```

```
## # A tibble: 2 x 3
## # Groups:   condition [2]
##   condition order            mean_time
##   <chr>     <chr>                <dbl>
## 1 auditory  auditory_tactile     3320.
## 2 tactile   auditory_tactile     4224.
```

```r
data_afirst %>% filter (success == "success") %>%
  ggplot(aes(x = condition, y = time*1000, fill = order)) +
  geom_boxplot(outlier.shape = NA) +
  labs(
    title = "Mean trial times per condition by condition order",
    x = NULL,
    y = "Trial time (ms) \n"
    ) +
  theme_linedraw() +
  theme(plot.title = element_text(face = "bold"), legend.position = "right") +
  scale_fill_discrete(labels = c("auditory first", "tactile first")) +
  scale_y_continuous(limits = c(0, 6000)) # remove outliers from plot
```

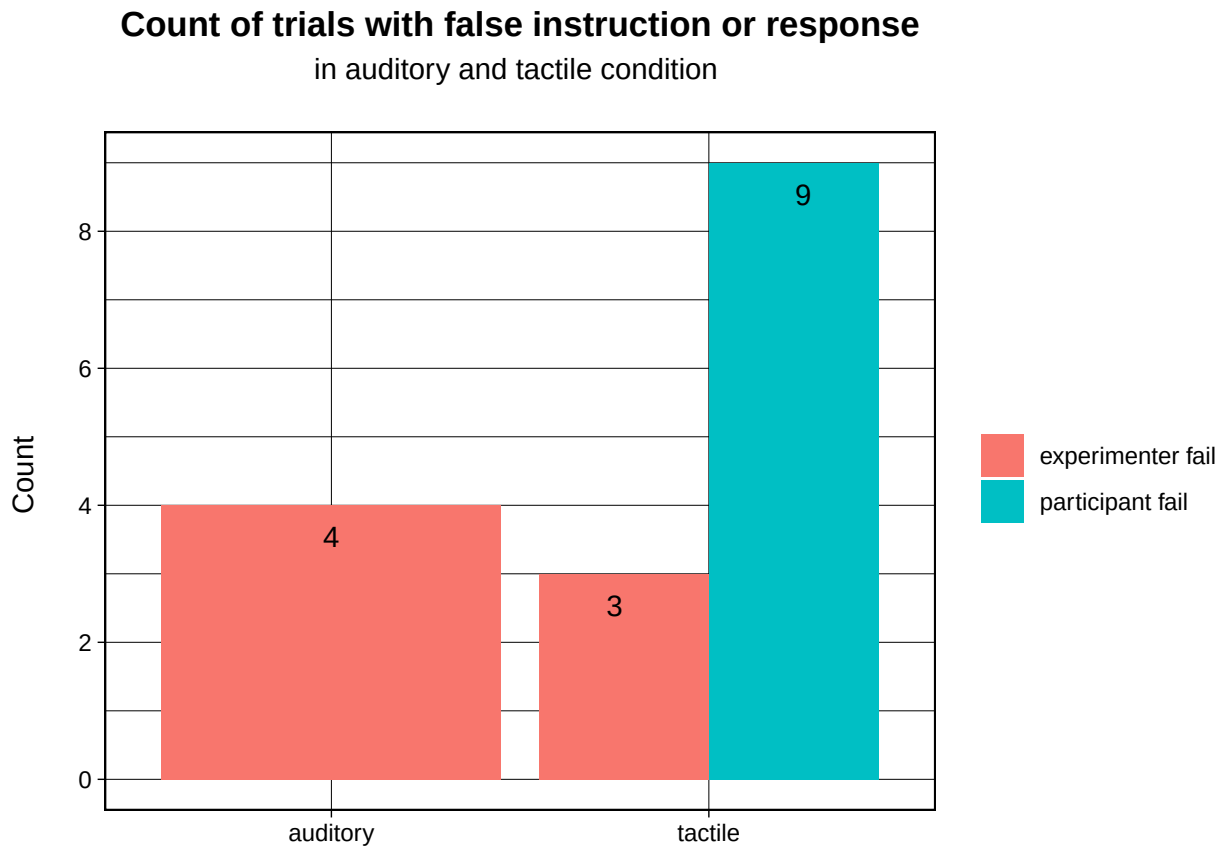


Fails by condition

```r
# Bar plot: x = fail, exfail (grouped by condition), y = counts
clean_data %>% group_by(condition, success) %>% count() %>% filter(success != "success")
```

```
## # A tibble: 3 x 3
## # Groups:   condition, success [3]
##   condition success     n
##   <chr>     <chr>   <int>
## 1 auditory  exFail      4
## 2 tactile   exFail      3
## 3 tactile   fail        9
```

```r
clean_data %>% group_by(condition, success) %>% count() %>% filter(success != "success") %>%
  ggplot(aes(x = condition, y = n, fill = success)) +
  geom_bar(stat = "identity", position = "dodge") +
  geom_text(aes(label = n), position = position_dodge(width = 1), vjust = 2, hjust=0.5) +
  scale_y_continuous(breaks = scales::pretty_breaks()) +
  labs(
    title = "Count of trials with false instruction or response",
    x = NULL,
    y = "Count \n",
    subtitle = "in auditory and tactile condition \n"
    ) +
  theme_linedraw() +
  theme(plot.title = element_text(face = "bold", hjust = 0.5),
    plot.subtitle = element_text(hjust = 0.5),
    legend.position = "right", legend.title = element_blank()) +
  scale_fill_discrete(labels = c("experimenter fail", "participant fail"))
```

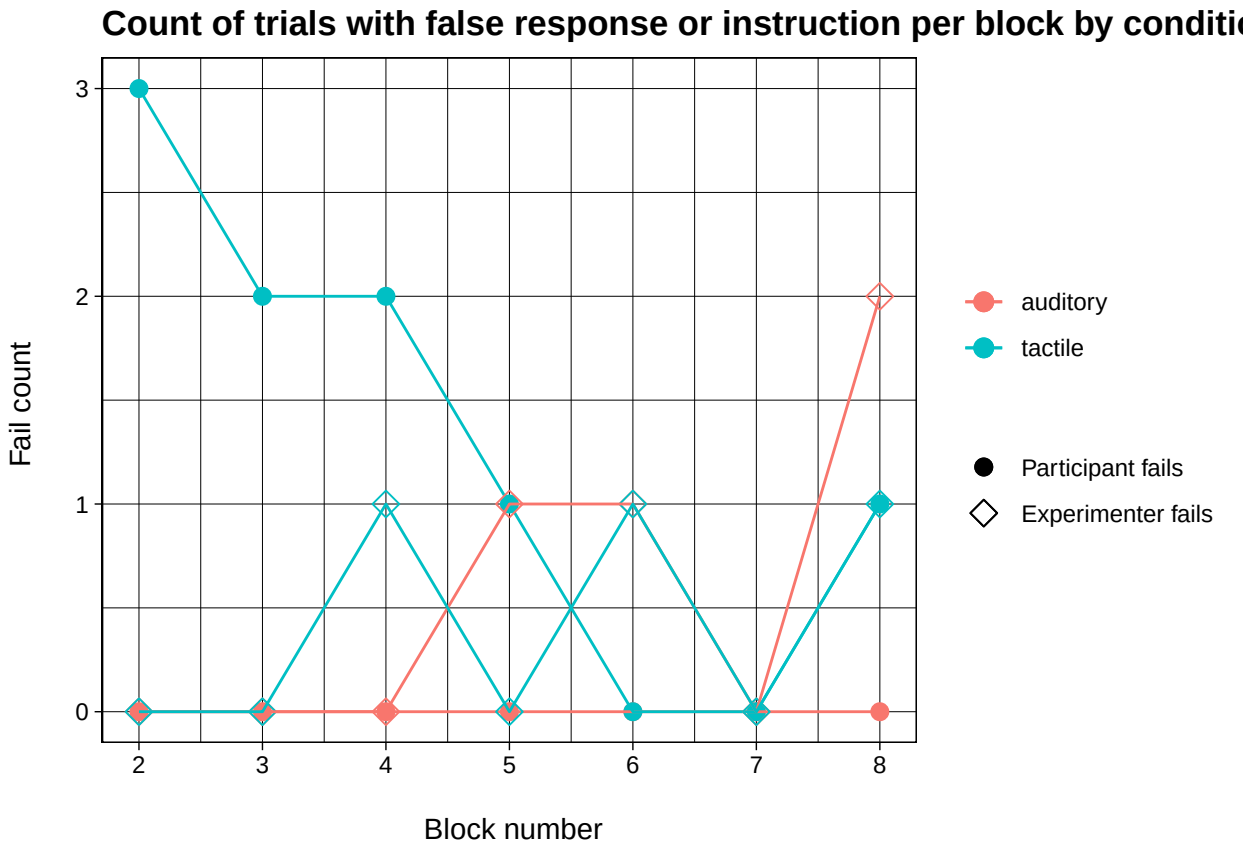



```r
# Line plot: x = block (grouped by condition --> 2 lines), y = fail counts
clean_data %>% group_by(block, condition) %>% summarize(fail_count = sum(success == "fail"), exFail_cou
```

```
## # A tibble: 14 x 4
## # Groups:   block [7]
##    block condition fail_count exFail_count
##    <dbl> <chr>          <int>        <int>
##  1     2 auditory           0            0
##  2     2 tactile            3            0
##  3     3 auditory           0            0
```

```
##  4     3 tactile            2            0
##  5     4 auditory           0            0
##  6     4 tactile            2            1
##  7     5 auditory           0            1
##  8     5 tactile            1            0
##  9     6 auditory           0            1
## 10     6 tactile            0            1
## 11     7 auditory           0            0
## 12     7 tactile            0            0
## 13     8 auditory           0            2
## 14     8 tactile            1            1
```

```r
fails_per_block <- clean_data %>% group_by(block, condition) %>% summarize(fail_count = sum(success ==
  ggplot(aes(x = block, y = fail_count, color=condition)) +
  geom_point(aes(y = fail_count, shape="Group 1"), size=3) +
  geom_line(aes(y = fail_count)) +
  geom_point(aes(y = exFail_count, shape="Group 2"), size=3) +
  geom_line(aes(y = exFail_count)) +
  labs(
    title = "Count of trials with false response or instruction per block by condition",
    x = "\n Block number",
    y = "Fail count \n"
    ) +
  theme_linedraw() +
  theme(plot.title = element_text(face = "bold"), legend.position = "right", legend.title = element_bla
  scale_x_continuous(breaks = c(1:8)) +
  scale_shape_manual(
    name = "Fail type",
    values = c("Group 1" = 16, "Group 2" = 5),  # Use 16 for a solid point
    labels = c("Participant fails", "Experimenter fails"))

fails_per_block
```




```
#ggsave(paste0(SAVE,"fails_per_block_blind.jpeg"), plot = fails_per_block, dpi = 600)

# test for significance
fails_a <- clean_data %>% group_by(block, condition) %>% summarize(fail_count = sum(success == "fail"))
fails_t <- clean_data %>% group_by(block, condition) %>% summarize(fail_count = sum(success == "fail"))

t.test(fails_a$fail_count, fails_t$fail_count)
```

```
##
##  Welch Two Sample t-test
##
## data:  fails_a$fail_count and fails_t$fail_count
## t = -3.0571, df = 6, p-value = 0.02231
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2.3147876 -0.2566409
## sample estimates:
## mean of x mean of y
##  0.000000  1.285714
```

```
fails_time <- clean_data %>% group_by(block, condition) %>% summarize(fail_count = sum(success == "fail
cor.test(fails_time$mean_time, fails_time$fail_count)
```

```
##
##  Pearson's product-moment correlation
##
## data:  fails_time$mean_time and fails_time$fail_count
## t = 1.0348, df = 5, p-value = 0.3482
```

```
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.4871302  0.8911888
## sample estimates:
##       cor
## 0.4199993
```

Fails per experimenter

```
# Subset data by experimenter
data_e2 <- clean_data[clean_data$participant_id == 778, ]
data_e2$experimenter <- "2" # p

data_e3 <- clean_data[clean_data$participant_id == 777, ]
data_e3$experimenter <- "3" # f

# bind the data together
experimenter_data <- rbind(data_e2, data_e3)

# Boxplot: x = experimenter, y = trial time
experimenter_data %>% filter(success == "success") %>% group_by(experimenter) %>% summarize(mean_times =
```

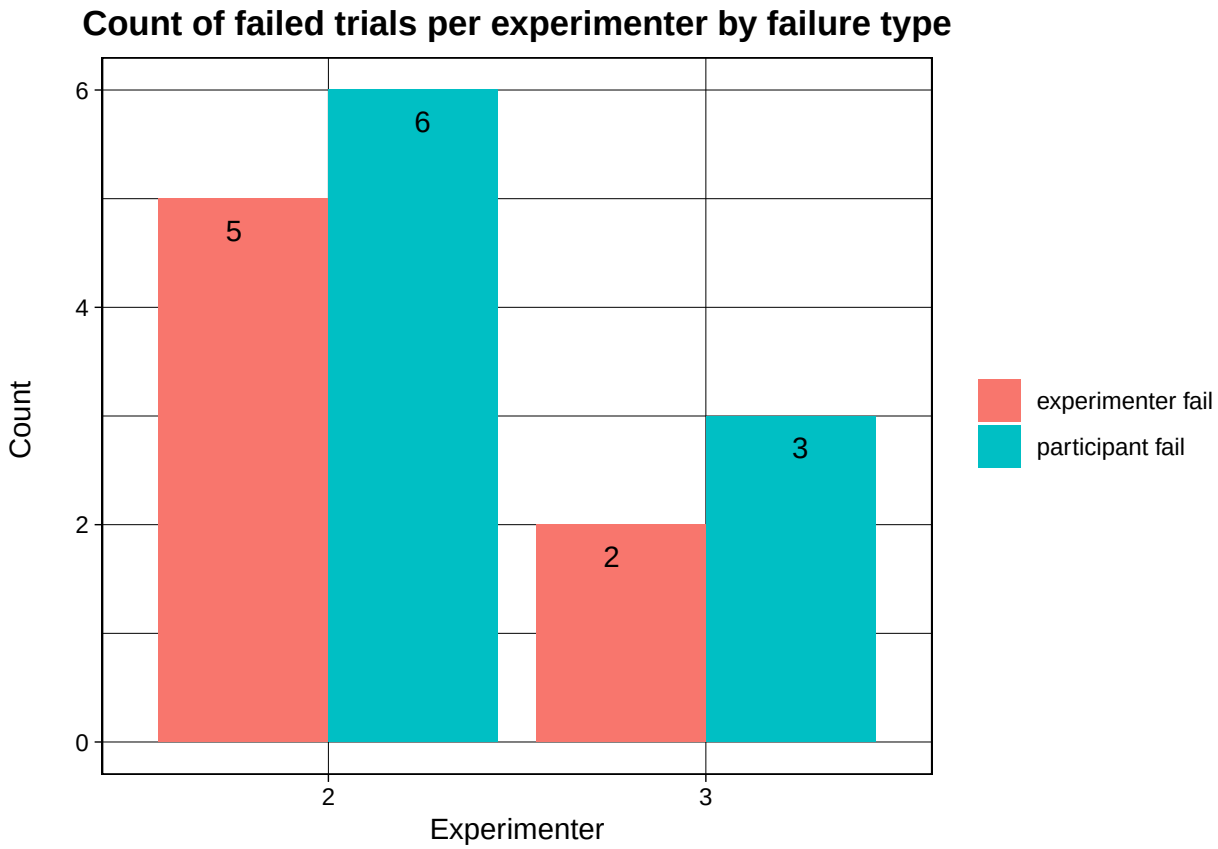
```
## # A tibble: 2 x 2
##   experimenter mean_times
##   <chr>             <dbl>
## 1 2                 3990.
## 2 3                 3529.
```

```
experimenter_data %>% filter(success == "success") %>%
  ggplot(aes(x = experimenter, y = time*1000)) +
  geom_boxplot(outlier.shape = NA) +
  labs(
    title = "Mean trial times per experimenter",
    x = "Experimenter",
    y = "Trial time (ms) \n",
    color = "experimenter"
    ) +
  theme_linedraw() +
  theme(plot.title = element_text(face = "bold"), legend.position = "right") +
  scale_fill_discrete(labels = c("auditory first", "tactile first")) +
  scale_y_continuous(limits = c(0, 6000)) # remove outliers from plot
```


```
# Barplot: x = experimenter, y = number of fails, grouped by type of fail
experimenter_data %>% group_by(experimenter, success) %>% count() %>% filter(success != "success")
```

```
## # A tibble: 4 x 3
## # Groups:   experimenter, success [4]
##   experimenter success     n
##   <chr>        <chr>   <int>
## 1 2            exFail      5
## 2 2            fail        6
## 3 3            exFail      2
## 4 4            fail        3
```

```
experimenter_data %>% group_by(experimenter, success) %>% count() %>% filter(success != "success") %>%
  ggplot(aes(x = experimenter, y = n, fill = success)) +
  geom_bar(stat = "identity", position = "dodge") +
  geom_text(aes(label = n), position = position_dodge(width = 1), vjust = 2, hjust=0.5) +
  labs(
    title = "Count of failed trials per experimenter by failure type",
    x = "Experimenter",
    y = "Count \n",
    ) +
  theme_linedraw() +
  theme(plot.title = element_text(face = "bold", hjust = 0.5),
    plot.subtitle = element_text(hjust = 0.5),
    legend.position = "right", legend.title = element_blank()) + # legend.title = element_blank()
  scale_fill_discrete(labels = c("experimenter fail", "participant fail"))
```



Linear Mixed-Effects Model (LMM)

Data preparation

```
grasping_data <- clean_data %>% filter(success == "success") %>% select(-success)

# add experimenter to df
grasping_data$experimenter <- ifelse(grasping_data$participant_id %in% c(778), "2", "3") # p, f

# add condition to df
grasping_data$order <- ifelse(grasping_data$participant_id %in% c(777,778), "auditory_first", "tactile_

# rename repetition locations in df
grasping_data$location[grasping_data$location == "rep_1"] = "1"
grasping_data$location[grasping_data$location == "rep_2"] = "2"
grasping_data$location[grasping_data$location == "rep_3"] = "3"
grasping_data$location[grasping_data$location == "rep_4"] = "4"
grasping_data$location[grasping_data$location == "rep_5"] = "5"
grasping_data$location[grasping_data$location == "rep_6"] = "6"
grasping_data$location[grasping_data$location == "rep_7"] = "7"
grasping_data$location[grasping_data$location == "rep_8"] = "8"
grasping_data$location[grasping_data$location == "rep_9"] = "9"
grasping_data$time <- grasping_data$time*1000 # s -> ms
```

```r
# cast block number and number of instructions to characters
grasping_data$block <- as.character(grasping_data$block) # numeric to character for linear mixed model
grasping_data$num_instructions <- as.character(grasping_data$num_instructions)
```

Final LMM

```r
model <- lmer(time ~
                condition +
                block +
                (condition | experimenter),
              data = grasping_data)

summary(model)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: time ~ condition + block + (condition | experimenter)
##    Data: grasping_data
##
## REML criterion at convergence: 4068.2
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.1916 -0.6163 -0.1075  0.7091  2.9764
##
## Random effects:
##  Groups       Name             Variance Std.Dev. Corr
##  experimenter (Intercept)       167926   409.8
##               conditiontactile   40453   201.1   -1.00
##  Residual                       1461912  1209.1
## Number of obs: 245, groups:  experimenter, 2
##
## Fixed effects:
##                   Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)       3182.113    364.973    1.952   8.719   0.0139 *
## conditiontactile   902.845    210.091    1.408   4.297   0.0895 .
## block3              44.494    291.504  236.008   0.153   0.8788
## block4             240.332    295.476  236.001   0.813   0.4168
## block5             310.001    293.415  236.000   1.057   0.2918
## block6             186.030    291.480  236.001   0.638   0.5239
## block7             227.804    291.480  236.001   0.782   0.4353
## block8             -42.480    293.415  236.000  -0.145   0.8850
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) cndtnt block3 block4 block5 block6 block7
## conditntctl -0.679
## block3      -0.412 -0.018
## block4      -0.409 -0.006  0.515
## block5      -0.411 -0.012  0.518  0.511
## block6      -0.412 -0.018  0.522  0.515  0.518
```

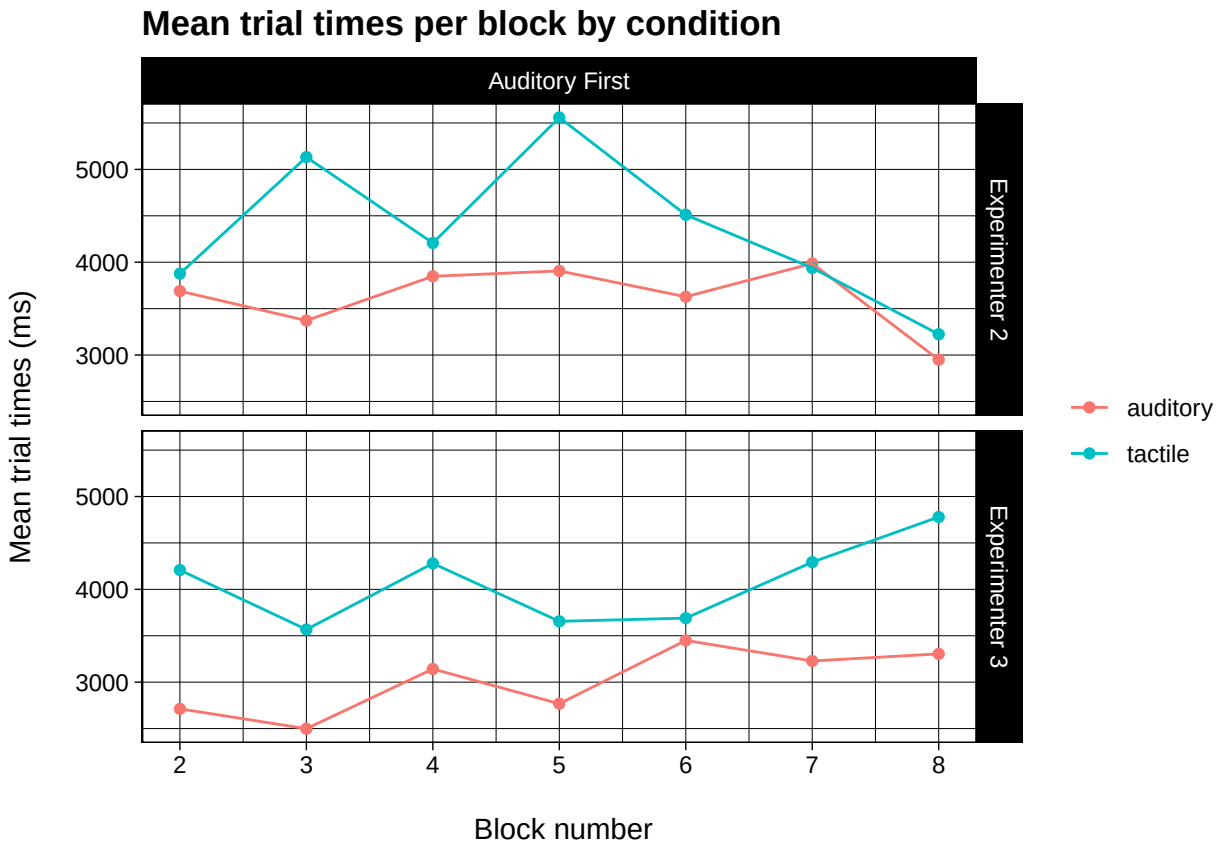
```
## block7      -0.412 -0.018  0.522  0.515  0.518  0.522
## block8      -0.411 -0.012  0.518  0.511  0.515  0.518  0.518
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
```

```
#anova(model) # --> no significant difference in trial times between conditions in this model
```

Random effects cannot be plotted against time since experimenter, order (and block) are all discrete.

LMM Visualizations

```
grasping_data$block <- as.double(grasping_data$block)

# change label names
labels <- c(`1` = "Experimenter 1",
            `2` = "Experimenter 2",
            `3` = "Experimenter 3",
            `auditory_first` = "Auditory First",
            `tactile_first` = "Tactile First")

lmm_plot <- grasping_data %>% group_by(block, condition, experimenter, order) %>% summarize(mean_time =
  ggplot(aes(x = block, y = mean_time, color=condition)) +
  geom_point() +
  geom_line() +
  labs(
    title = "Mean trial times per block by condition",
    x = "\n Block number",
    y = "Mean trial times (ms) \n"
    ) +
  theme_linedraw() +
  theme(plot.title = element_text(face = "bold"), legend.position = "right", legend.title = element_bla
  scale_x_continuous(breaks = c(2:8)) +
  #scale_y_continuous(limits = c(2000, 4000)) +
  facet_grid(experimenter ~ order,
             labeller = as_labeller(labels))
lmm_plot
```




```
#ggsave(paste0(SAVE,"lmm_plot_blind.jpeg"), plot = lmm_plot, dpi = 600)
```