OptiVisT grasping task analysis

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")
# 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.csv")
```

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)
```

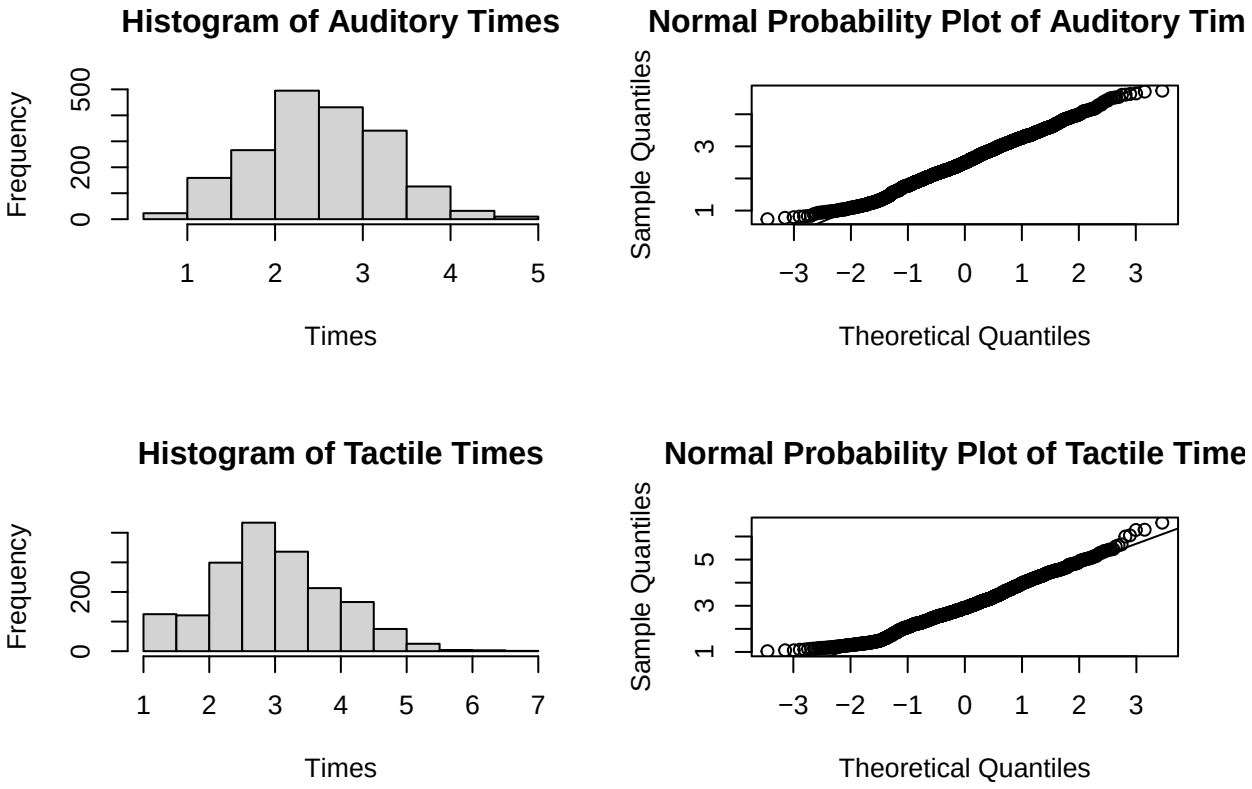
```
## clean_data$condition: auditory
##
##  Grubbs test for one outlier
##
## data:  dd[x, ]
## G = 4.5325, U = 0.9894, p-value = 0.005356
## alternative hypothesis: highest value 5.9696434 is an outlier
##
## ------------------------------------------------------------
## clean_data$condition: tactile
##
##  Grubbs test for one outlier
```

```
##
## data:  dd[x, ]
## G = 6.85857, U = 0.97578, p-value = 5.063e-09
## alternative hypothesis: highest value 9.86988169999995 is an outlier
```

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

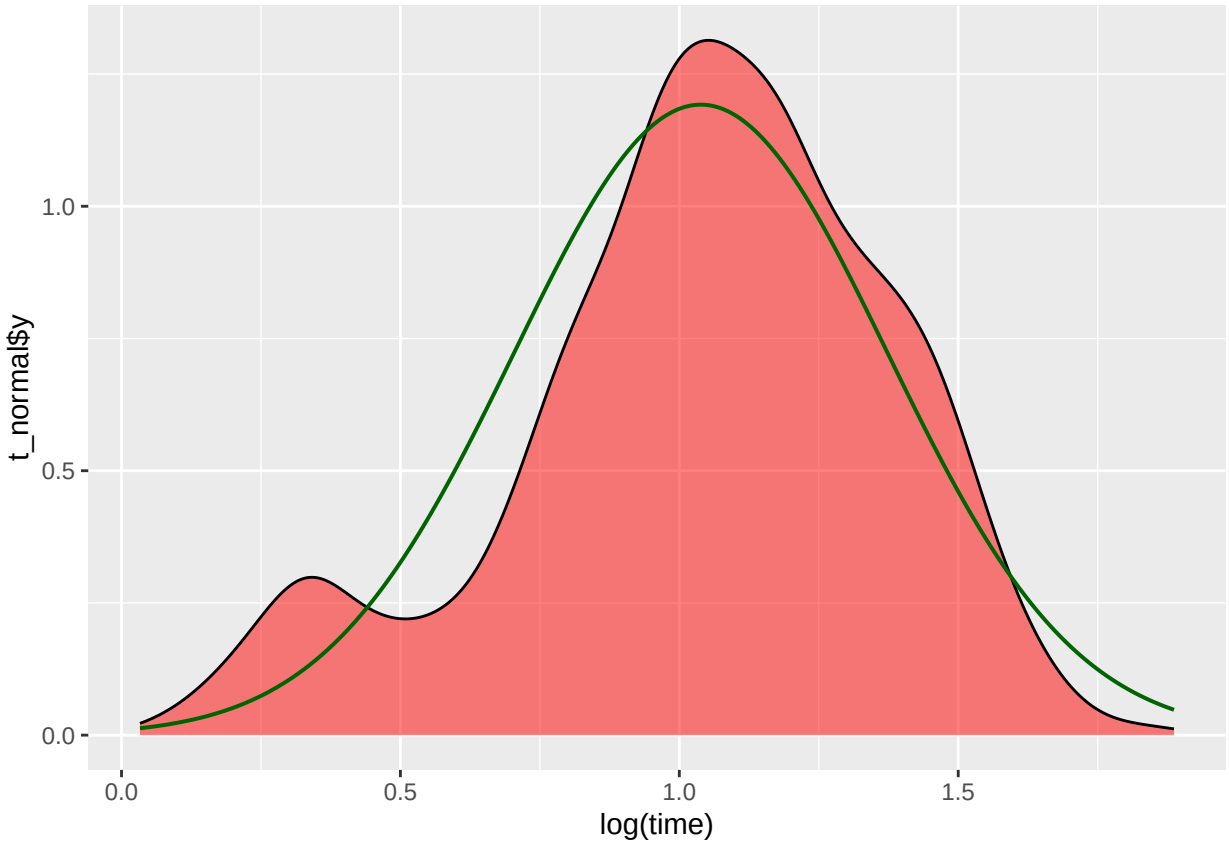
Normality assumption check

```
# 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)
```

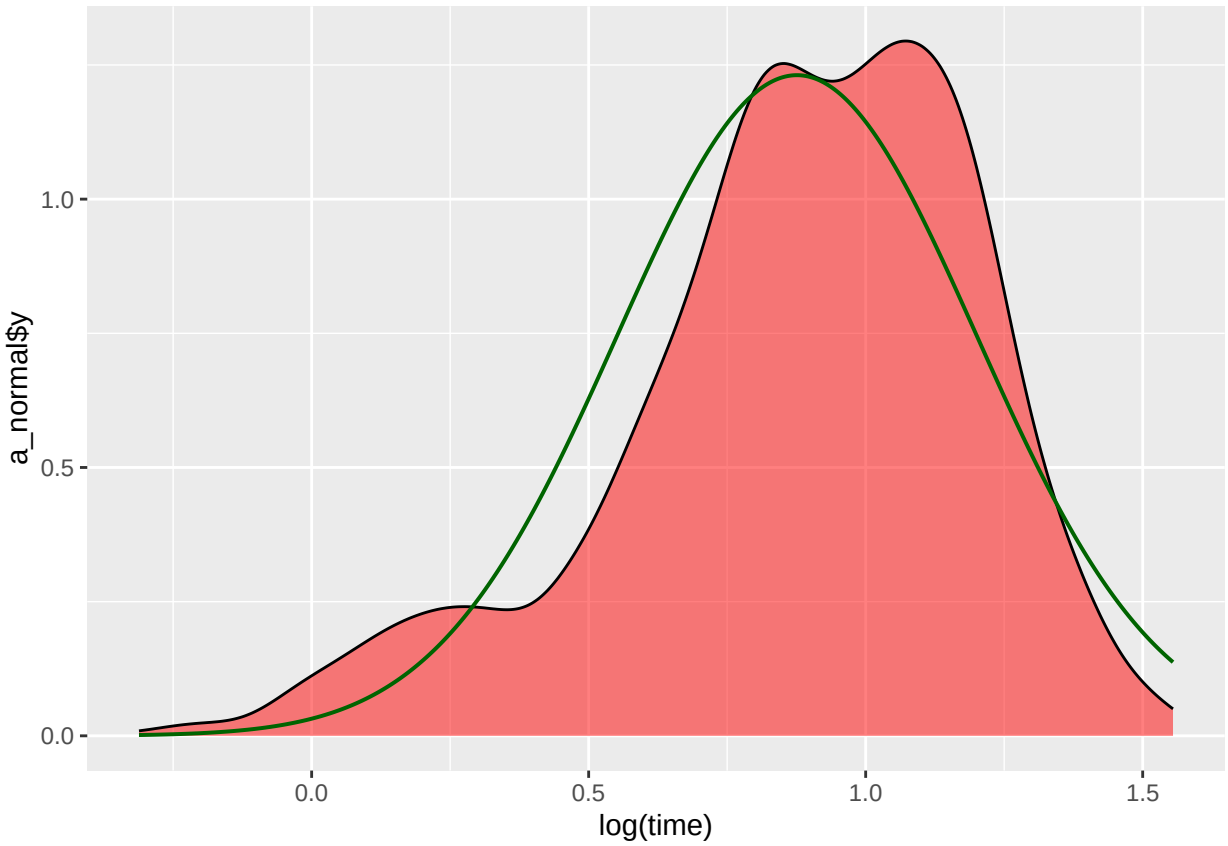



```
# normal distribution
x <- seq(min(log(tactile_data$time)), max(log(tactile_data$time)), length.out = length(tactile_data$tim
```

```r
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")
```



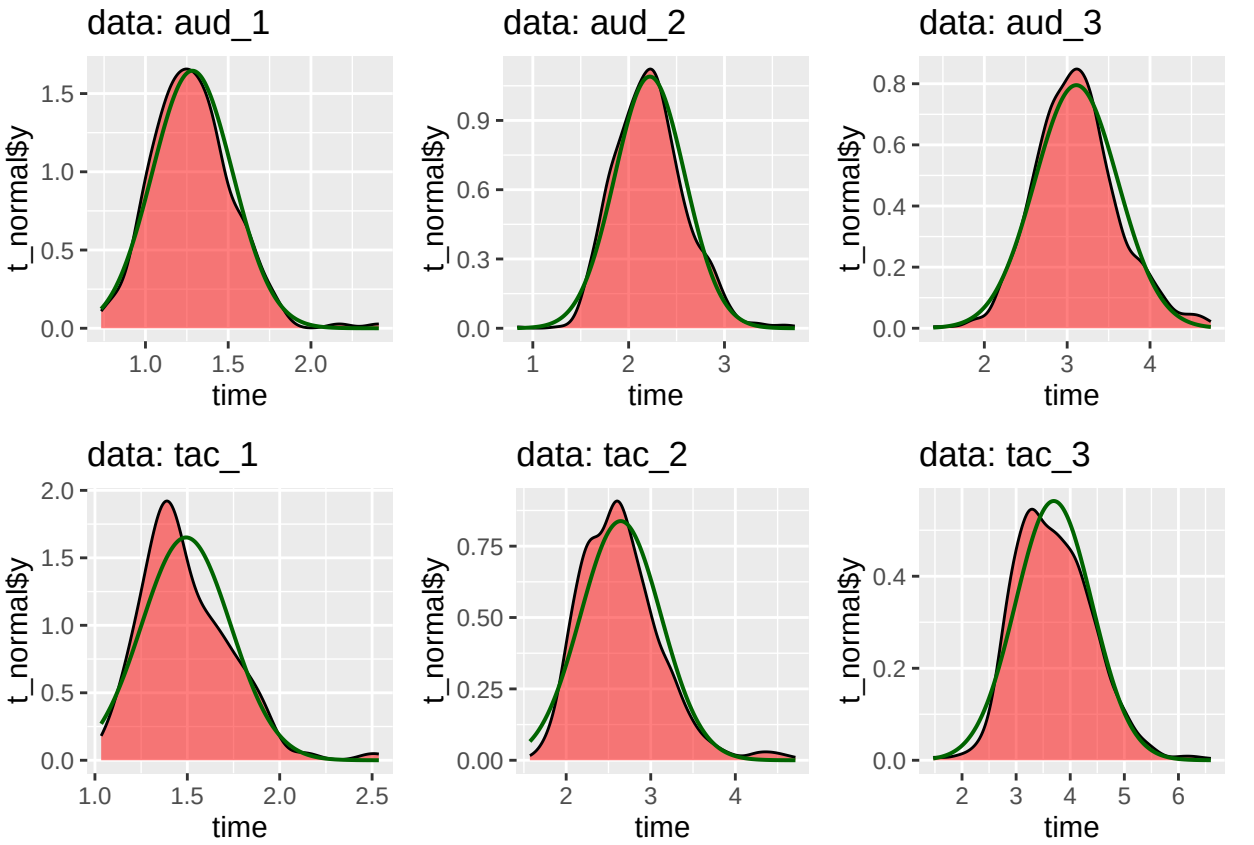

```r
ggplot(auditory_data) +
  geom_density(aes(x = log(time)), fill = "red", alpha = 0.5) +
  geom_line(aes(x = a_normal$x, y = a_normal$y), size = 0.7, color = "darkgreen")
```



Looks multi-modal because of different number of instructions.

```
# 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.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))
```

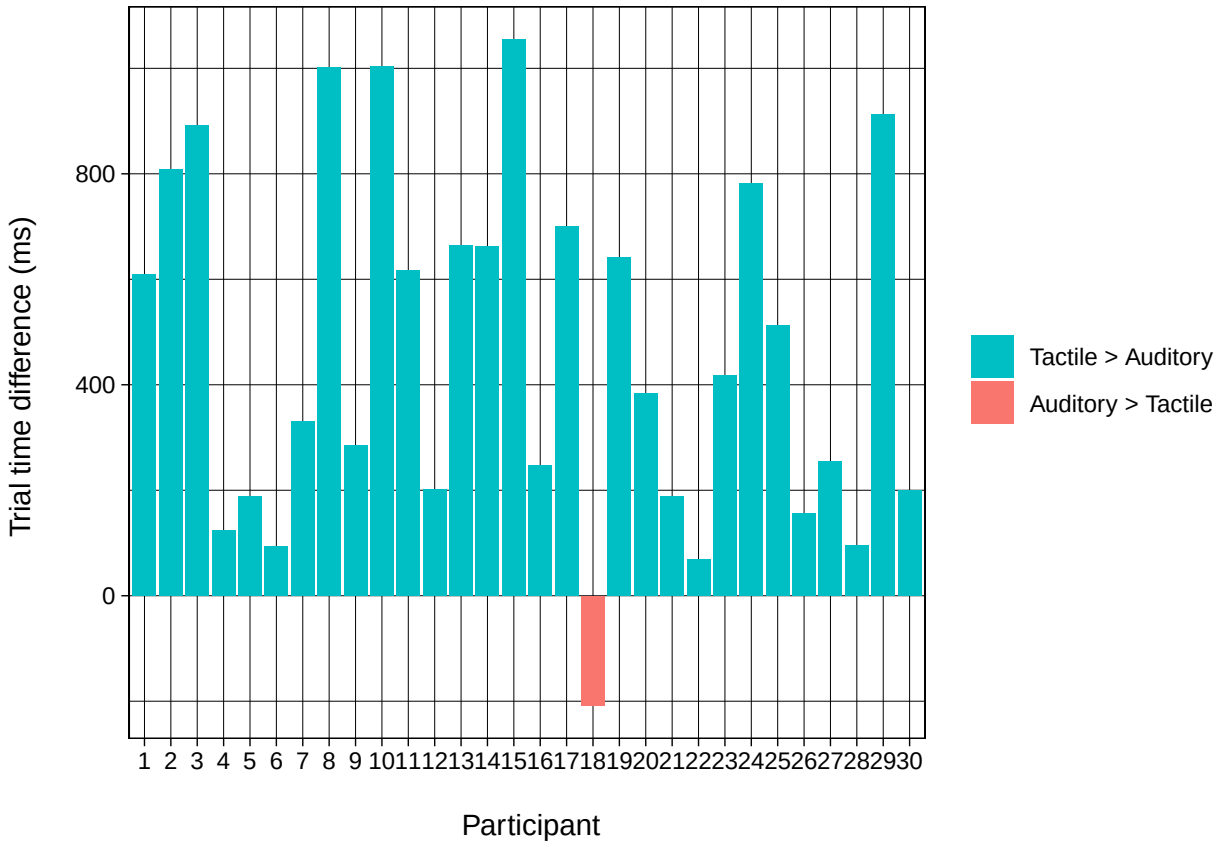
```
## # A tibble: 2 x 3
##   condition  mean    sd
```

```
##   <chr>     <dbl> <dbl>
## 1 auditory   2.54 0.757
## 2 tactile    3.02 0.999
```

```
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  3.74 1.82
##  2 auditory      2  2.69 0.798
##  3 auditory      3  2.59 0.745
##  4 auditory      4  2.47 0.762
##  5 auditory      5  2.50 0.786
##  6 auditory      6  2.50 0.723
##  7 auditory      7  2.53 0.743
##  8 auditory      8  2.48 0.723
##  9 tactile       1  4.41 1.94
## 10 tactile       2  3.17 1.13
## 11 tactile       3  3.16 1.06
## 12 tactile       4  2.98 0.959
## 13 tactile       5  3.02 0.876
## 14 tactile       6  3.00 0.976
## 15 tactile       7  2.92 0.945
## 16 tactile       8  2.89 1.00
```

```
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
```



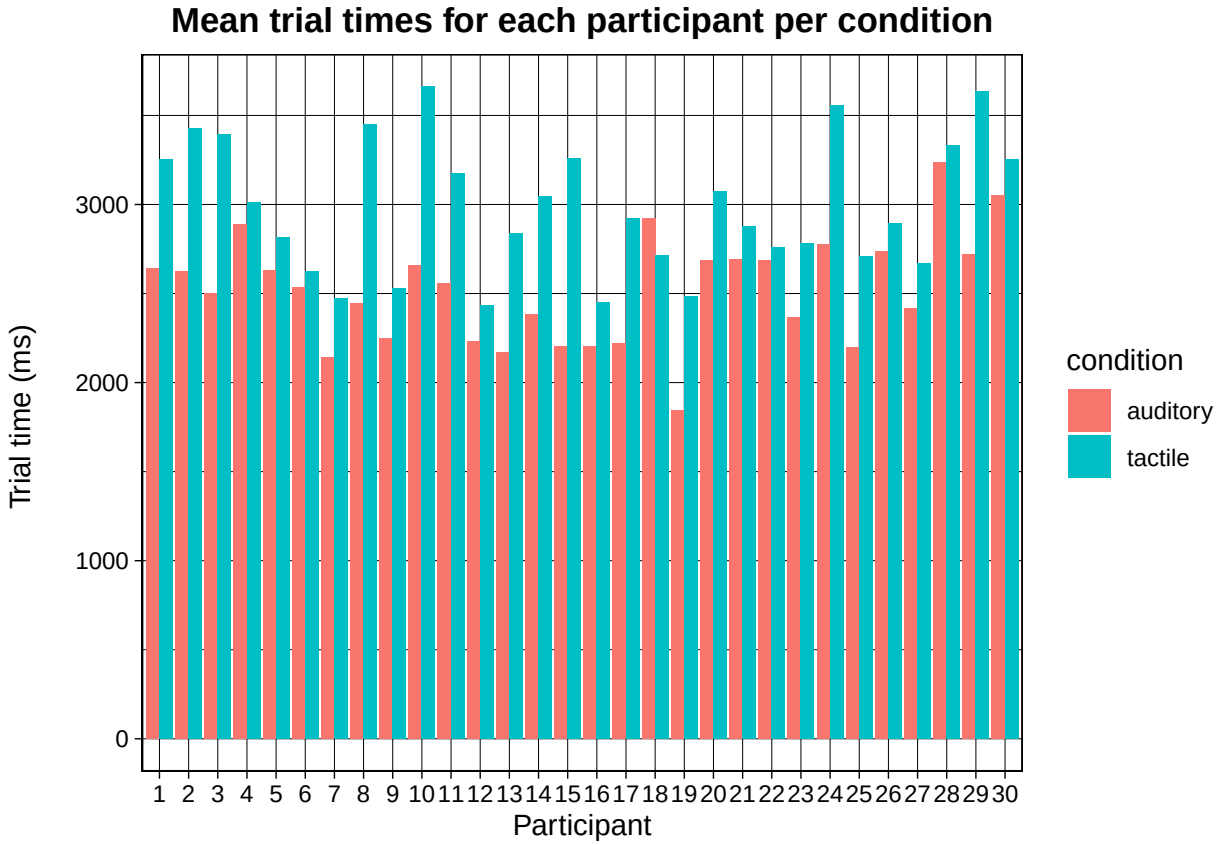

```
#ggsave(paste0(SAVE,"times_per_participant.jpeg"), plot = times_per_participant, dpi = 600)
```

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

```
## # A tibble: 60 x 3
## # Groups:   participant_id [30]
##    participant_id condition mean_time
##             <dbl> <chr>         <dbl>
##  1              1 auditory      2642.
##  2              1 tactile       3252.
##  3              2 auditory      2622.
##  4              2 tactile       3430.
##  5              3 auditory      2500.
##  6              3 tactile       3392.
##  7              4 auditory      2888.
##  8              4 tactile       3012.
##  9              5 auditory      2630.
## 10              5 tactile       2819.
## # i 50 more rows
```

```
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
```



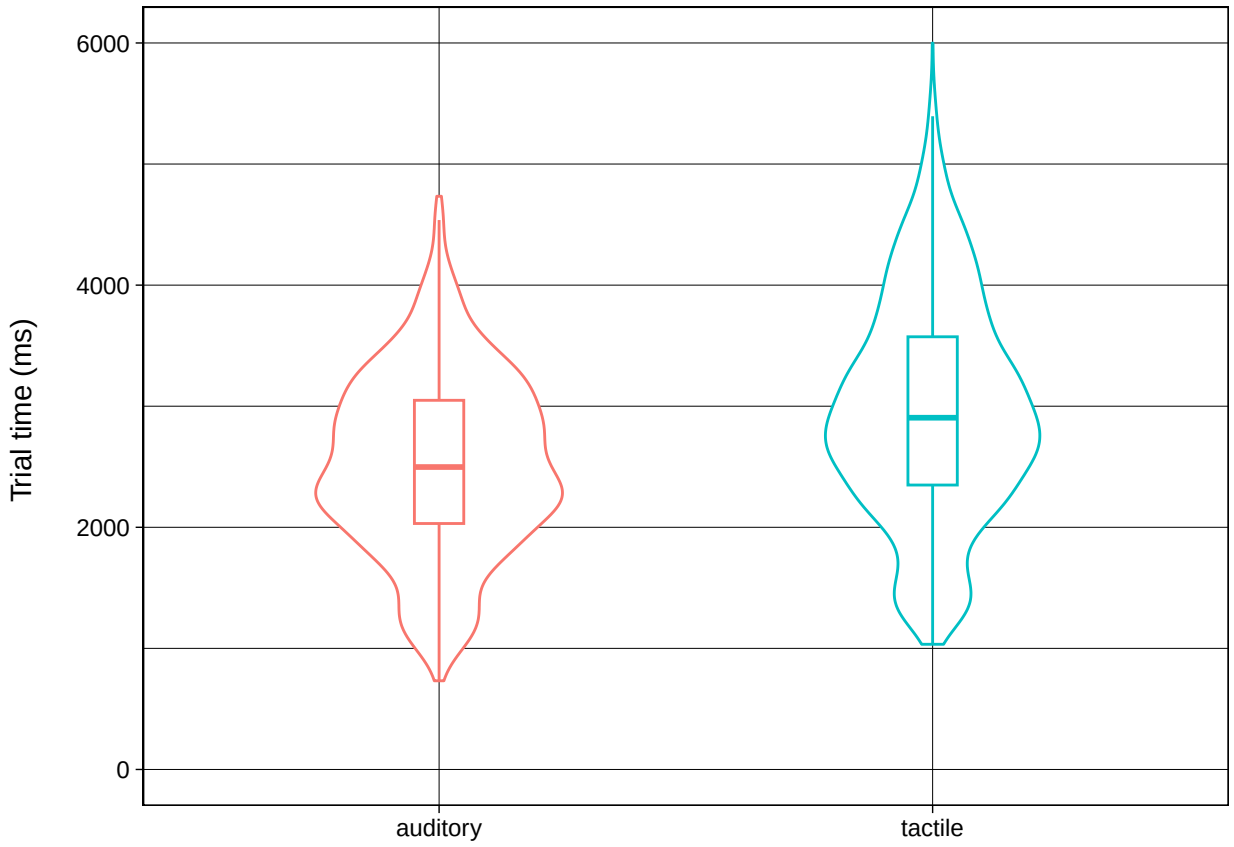

```
#ggsave(paste0(SAVE,"times_per_participant.jpeg"), plot = times_per_participant, dpi = 600)
```

Trial times per condition

```
# Violin plot: x = condition, y = RT
clean_data %>% group_by(condition) %>% filter(success == "success") %>% summarize(mean_time = mean(time
```

```
## # A tibble: 2 x 2
##   condition mean_time
##   <chr>         <dbl>
## 1 auditory      2520.
## 2 tactile       2977.
```

```r
times_condition_violin <- clean_data %>% filter(success == "success") %>%
  ggplot(aes(x = condition, y = time*1000, color=condition)) +
  geom_violin(width=0.5) +
  geom_boxplot(outlier.shape = NA, width=0.1) +
  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
  guides(color = FALSE)

times_condition_violin
```




```r
#ggsave(paste0(SAVE,"times_condition_violin.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)
```

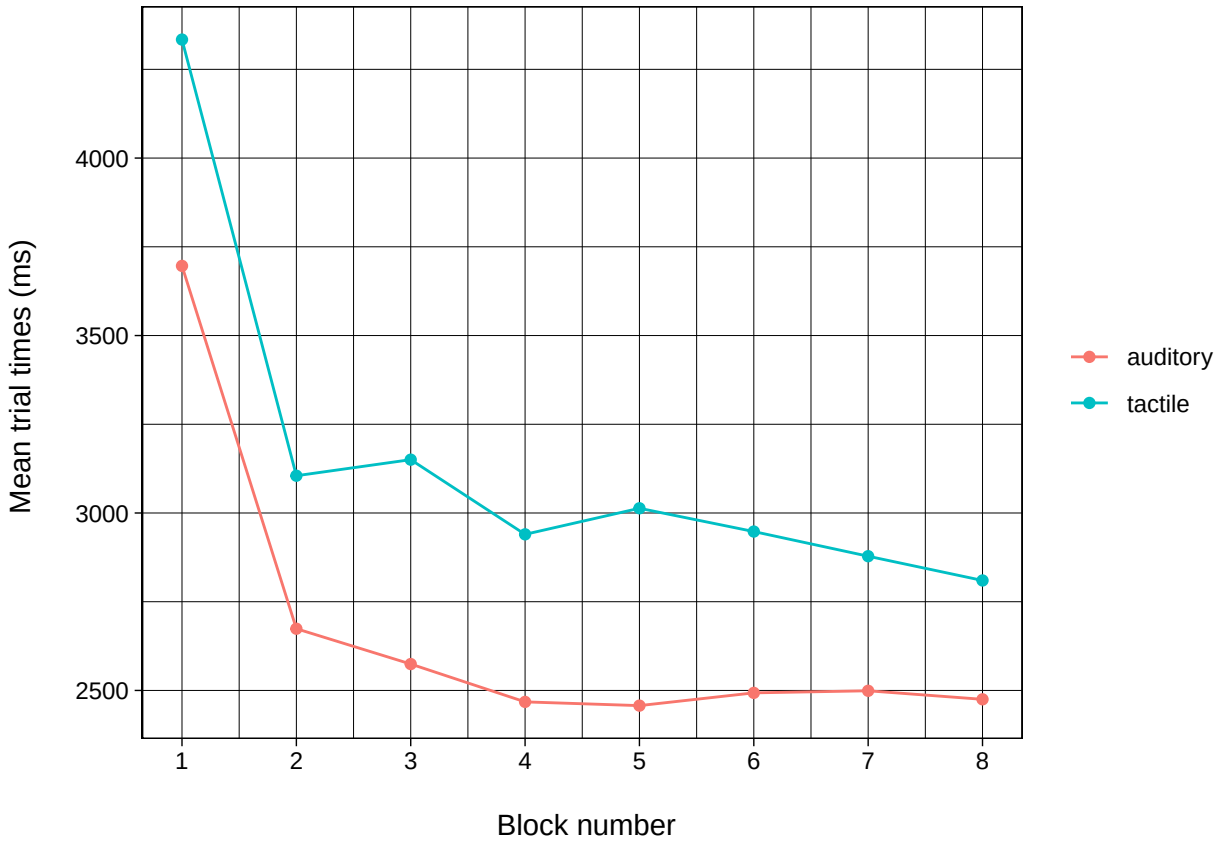
```
##
##  Welch Two Sample t-test
##
## data:  auditory_data$time and tactile_data$time
## t = -16.501, df = 3439.2, p-value < 2.2e-16
```

```
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.5115145 -0.4028665
## sample estimates:
## mean of x mean of y
##  2.520197  2.977388
```

```
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      3696.
##  2     1 tactile       4334.
##  3     2 auditory      2674.
##  4     2 tactile       3105.
##  5     3 auditory      2574.
##  6     3 tactile       3150.
##  7     4 auditory      2468.
##  8     4 tactile       2940.
##  9     5 auditory      2457.
## 10     5 tactile       3013.
## 11     6 auditory      2493.
## 12     6 tactile       2948.
## 13     7 auditory      2499.
## 14     7 tactile       2878.
## 15     8 auditory      2475.
## 16     8 tactile       2810.
```

```
times_per_block <- learning_data %>% group_by(block, condition) %>% summarize(mean_time = mean(time)*100
  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.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 = 9.3839, df = 376.33, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.9713955 1.4863942
## sample estimates:
## mean of x mean of y
##  4.333879  3.104984
```

```
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 = 2.5408, df = 512.22, p-value = 0.01136
## alternative hypothesis: true difference in means is not equal to 0
```

```
## 95 percent confidence interval:
##  0.0477178 0.3731385
## sample estimates:
## mean of x mean of y
##  3.150268  2.939840
```

```
# 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 = 8.8956, df = 371.94, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.7963129 1.2482650
## sample estimates:
## mean of x mean of y
##  3.696215  2.673926
```

```
# 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 = 0.16151, df = 535.96, p-value = 0.8718
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.1173220  0.1383423
## sample estimates:
## mean of x mean of y
##  2.467777  2.457267
```

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

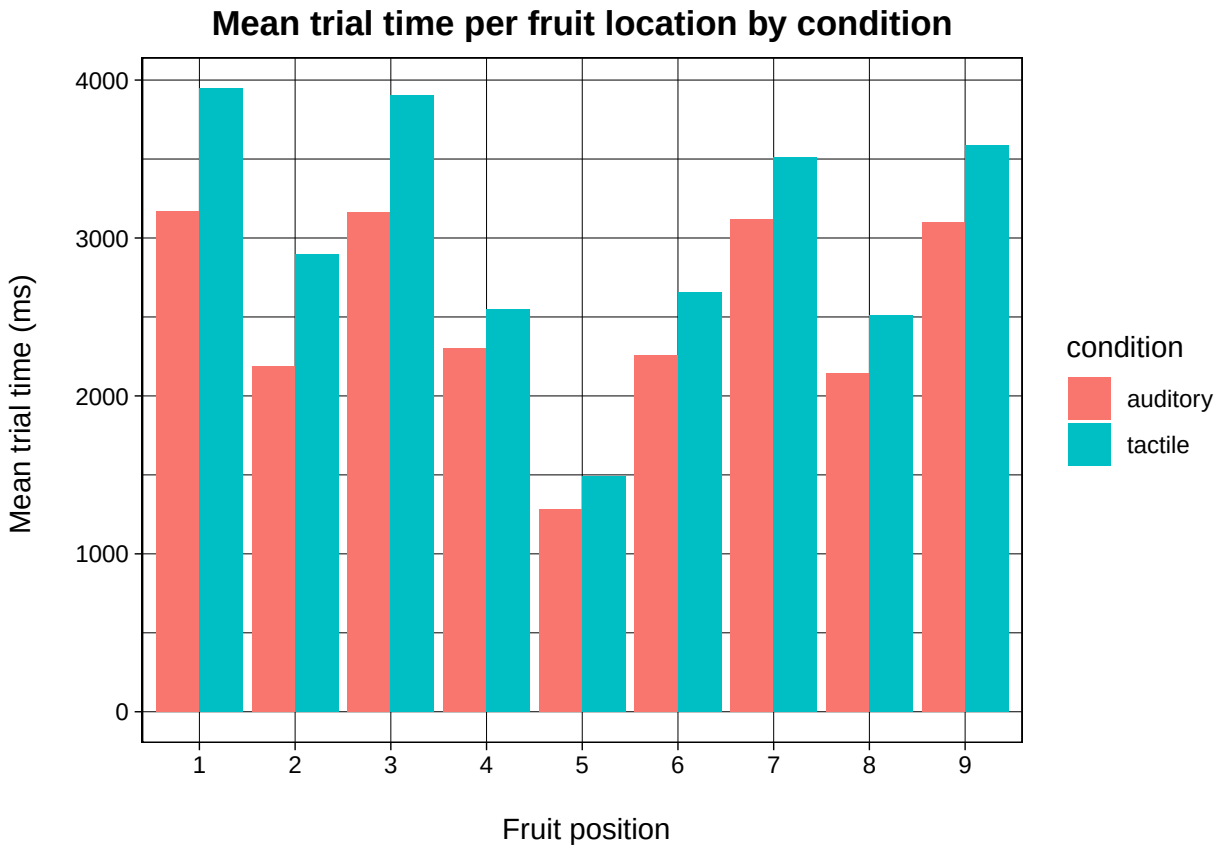
```
##
##  Welch Two Sample t-test
##
## data:  filter(learning_data, block == 4, condition == "tactile")$time and filter(learning_data, block
## t = -0.92947, df = 508.72, p-value = 0.3531
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.22868045  0.08179513
## sample estimates:
## mean of x mean of y
##  2.939840  3.013283
```

```
# 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            3167.
##  2 auditory  2            2189.
##  3 auditory  3            3164.
##  4 auditory  4            2302.
##  5 auditory  5            1284.
##  6 auditory  6            2260.
##  7 auditory  7            3117.
##  8 auditory  8            2141.
##  9 auditory  9            3102.
## 10 tactile   1            3946.
## 11 tactile   2            2896.
## 12 tactile   3            3907.
## 13 tactile   4            2551.
## 14 tactile   5            1493.
## 15 tactile   6            2657.
## 16 tactile   7            3514.
## 17 tactile   8            2508.
## 18 tactile   9            3589.
```

```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")
```



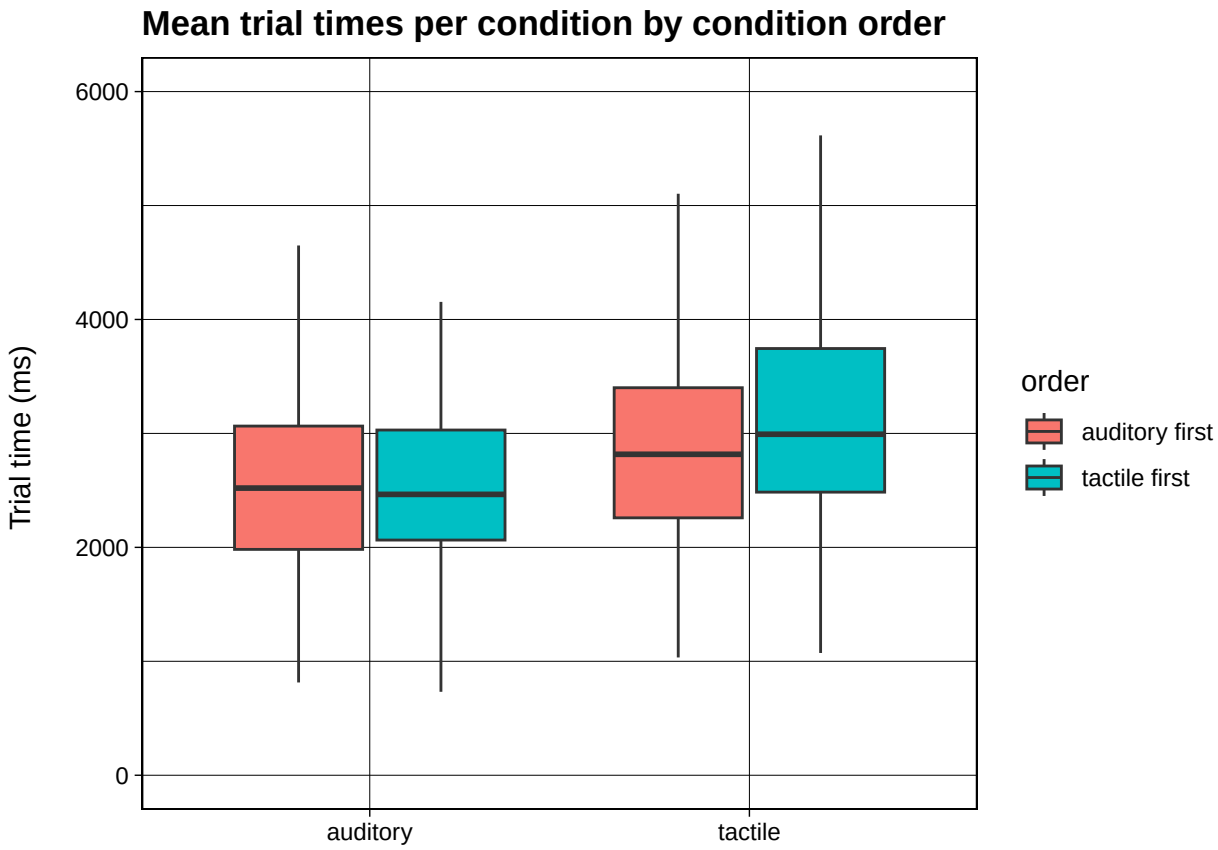

```
# mid (5) fastest as starting point is in front
# one-command positions up (2), left (4), right (6), down (8) are shorter than two-command positions
# diff between conditions for positions 1 and 3 (upper left and right) greater than for 7 and 9 (lower
# --> grasping at top shelf took longer, probably 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_tfirst <- clean_data[clean_data$participant_id == 2
                          | clean_data$participant_id == 3
                          | clean_data$participant_id == 5
                          | clean_data$participant_id == 8
                          | clean_data$participant_id == 10
                          | clean_data$participant_id == 12
                          | clean_data$participant_id == 15
                          | clean_data$participant_id == 17
                          | clean_data$participant_id == 19
                          | clean_data$participant_id == 20
                          | clean_data$participant_id == 21
                          | clean_data$participant_id == 24
                          | clean_data$participant_id == 26
                          | clean_data$participant_id == 27
                          | clean_data$participant_id == 29, ]


if (length(data_tfirst != 0)) {
```

```
  data_tfirst$order <- "tactile_auditory"
}

data_afirst <- clean_data[clean_data$participant_id == 1
                          | clean_data$participant_id == 4
                          | clean_data$participant_id == 6
                          | clean_data$participant_id == 7
                          | clean_data$participant_id == 9
                          | clean_data$participant_id == 11
                          | clean_data$participant_id == 13
                          | clean_data$participant_id == 14
                          | clean_data$participant_id == 16
                          | clean_data$participant_id == 18
                          | clean_data$participant_id == 22
                          | clean_data$participant_id == 23
                          | clean_data$participant_id == 25
                          | clean_data$participant_id == 28
                          | clean_data$participant_id == 30, ]


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

# bind the order data
order_data <- rbind(data_tfirst, data_afirst)

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

```
## # A tibble: 4 x 3
## # Groups:   condition [2]
##   condition order            mean_time
##   <chr>     <chr>                <dbl>
## 1 auditory  auditory_tactile     2549.
## 2 auditory  tactile_auditory     2492.
## 3 tactile   auditory_tactile     2862.
## 4 tactile   tactile_auditory     3092.
```

```
order_data %>% 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
```




```
# test diffs for significance
tactile_tfirst <- order_data %>% filter(success == "success", condition == "tactile", order == "tactile_
tactile_afirst <- order_data %>% filter(success == "success", condition == "tactile", order == "auditory
auditory_tfirst <- order_data %>% filter(success == "success", condition == "auditory", order == "tacti
auditory_afirst <- order_data %>% filter(success == "success", condition == "auditory", order == "audit

t.test(tactile_tfirst$time, tactile_afirst$time, paired = FALSE) # ***
```

```
##
##  Welch Two Sample t-test
##
## data:  tactile_tfirst$time and tactile_afirst$time
## t = 5.3081, df = 1790.9, p-value = 1.245e-07
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.1451528 0.3152749
## sample estimates:
## mean of x mean of y
##  3.092111  2.861897
```

```
t.test(auditory_tfirst$time, auditory_afirst$time, paired = FALSE)
```

```
##
##  Welch Two Sample t-test
##
## data:  auditory_tfirst$time and auditory_afirst$time
## t = -1.6675, df = 1859, p-value = 0.09559
## alternative hypothesis: true difference in means is not equal to 0
```

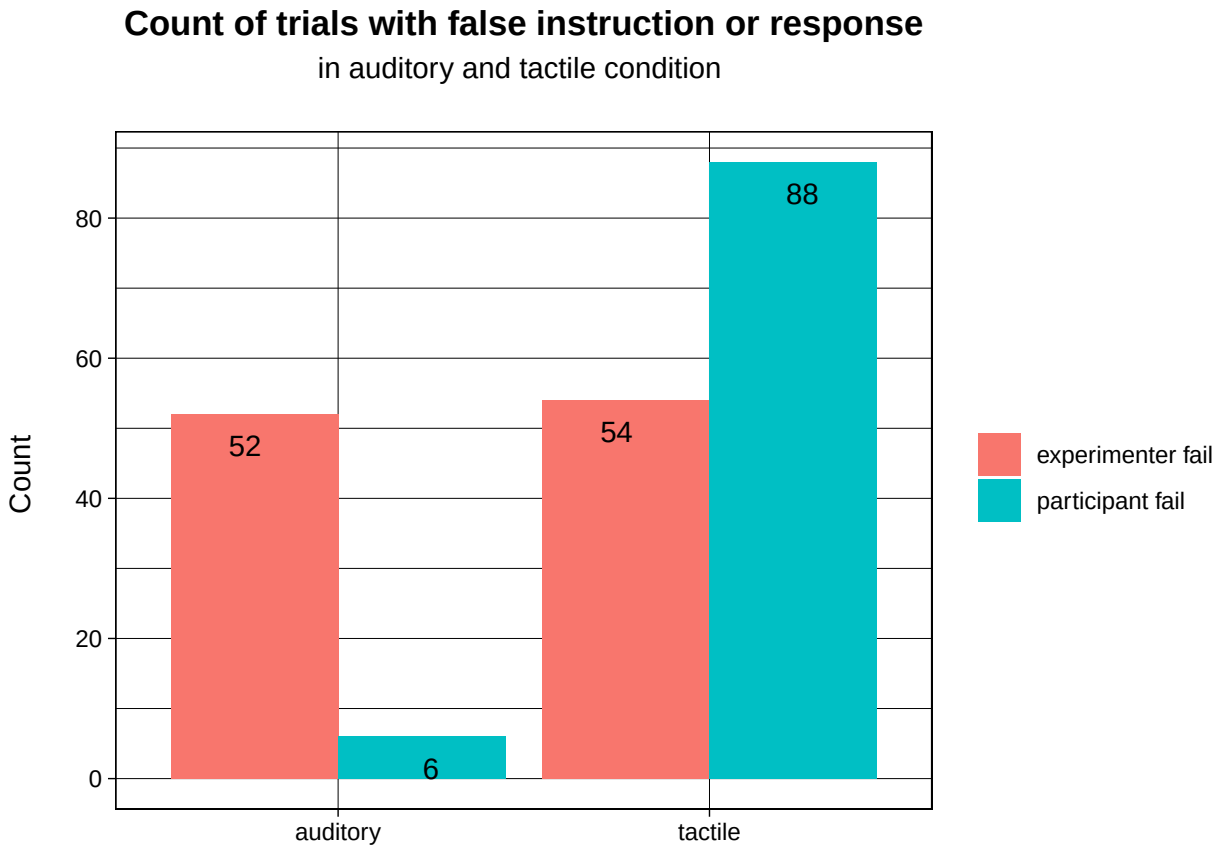
```
## 95 percent confidence interval:
##  -0.12356201  0.01000258
## sample estimates:
## mean of x mean of y
##  2.491822  2.548602
```

Fails by condition

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

```
## # A tibble: 4 x 3
## # Groups:   condition, success [4]
##   condition success     n
##   <chr>     <chr>   <int>
## 1 auditory  exFail     52
## 2 auditory  fail        6
## 3 tactile   exFail     54
## 4 tactile   fail       88
```

```
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"))
```

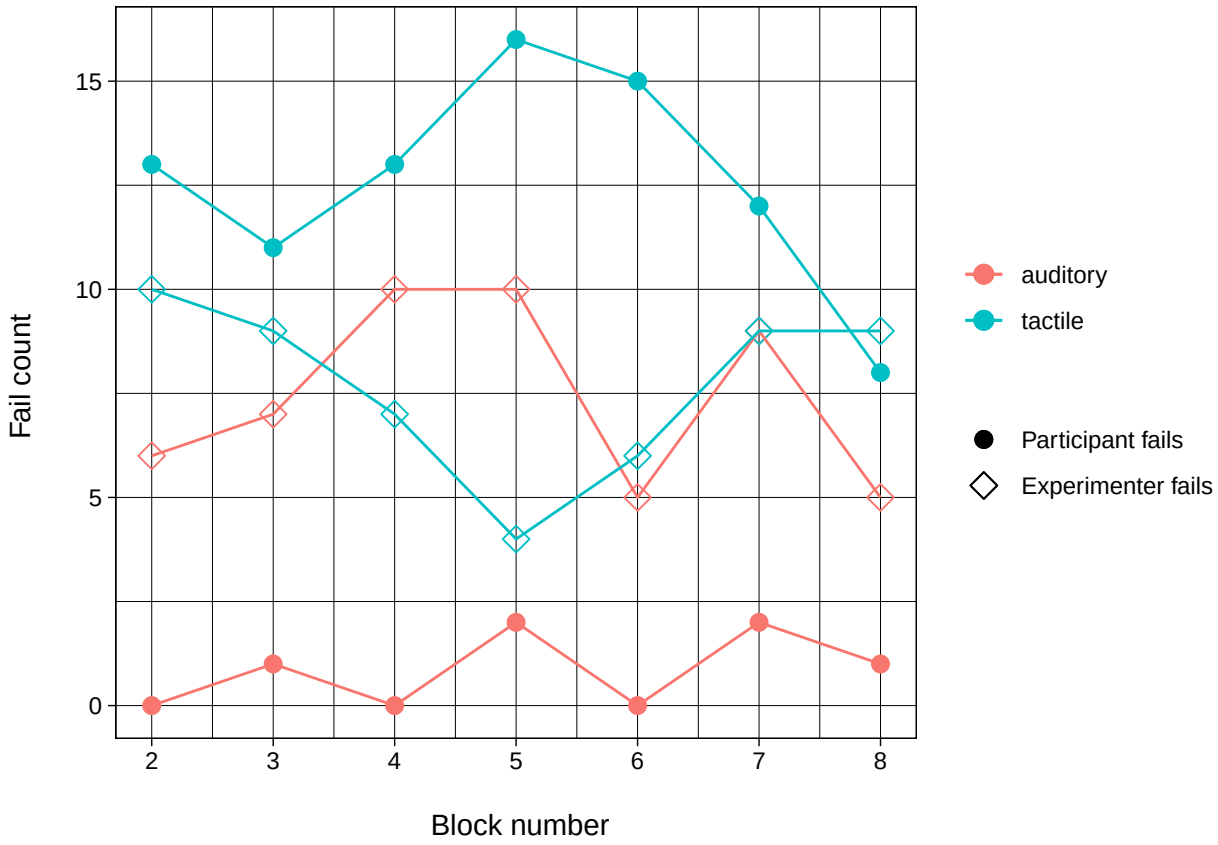



```
# Line plot: x = block (grouped by condition --> 2 lines), y = fail counts
clean_data %>% group_by(block, condition) %>% summarize(fail_count = sum(success == "fail"), exFail_cou
```

```
## # A tibble: 14 x 4
## # Groups:   block [7]
##    block condition fail_count exFail_count
##    <dbl> <chr>          <int>        <int>
##  1     2 auditory           0            6
##  2     2 tactile           13           10
##  3     3 auditory           1            7
##  4     3 tactile           11            9
##  5     4 auditory           0           10
##  6     4 tactile           13            7
##  7     5 auditory           2           10
##  8     5 tactile           16            4
##  9     6 auditory           0            5
## 10     6 tactile           15            6
## 11     7 auditory           2            9
## 12     7 tactile           12            9
## 13     8 auditory           1            5
## 14     8 tactile            8            9
```

```
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.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 = -11.125, df = 7.3786, p-value = 7.12e-06
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -14.178609  -9.249963
## sample estimates:
##  mean of x  mean of y
##  0.8571429 12.5714286
```

```
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 = 0.52941, df = 5, p-value = 0.6192
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.6323846  0.8380493
## sample estimates:
##       cor
## 0.2303908
```

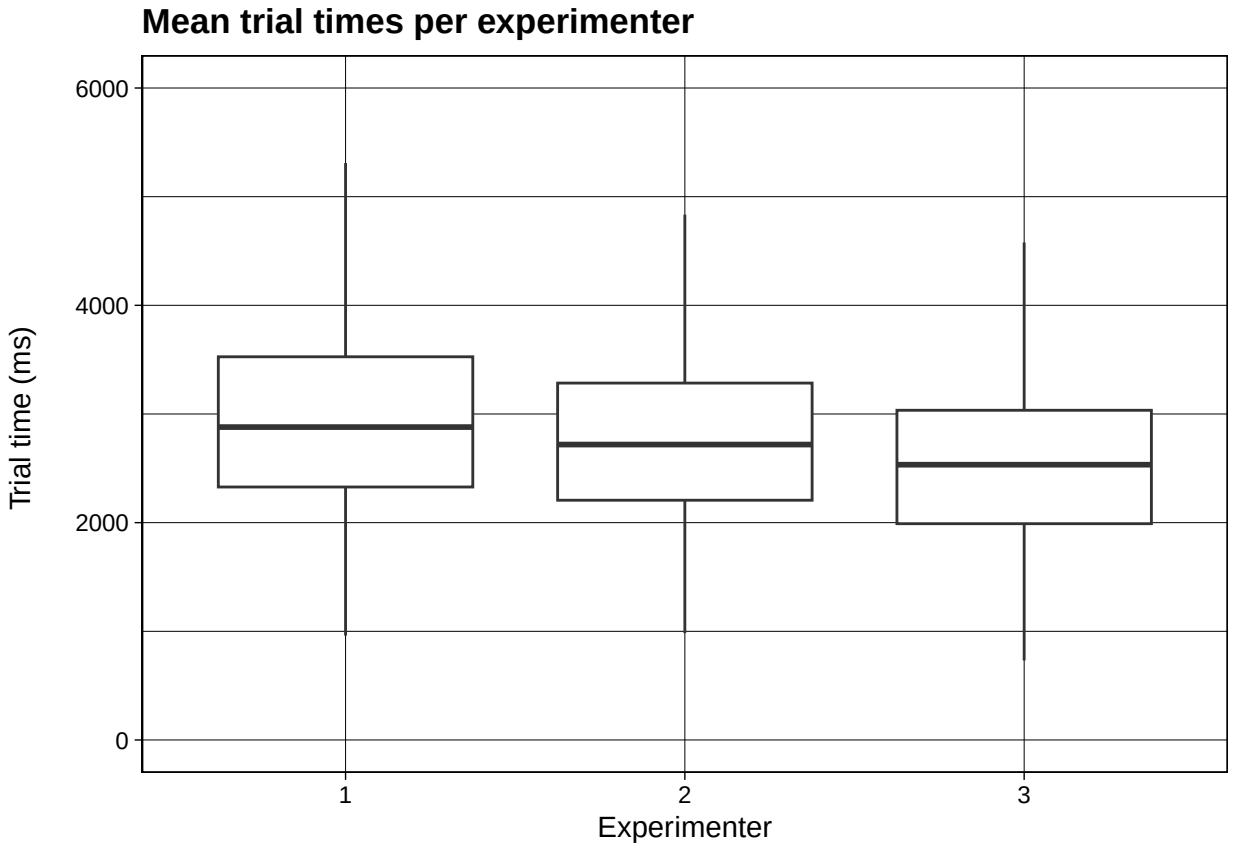
Fails per experimenter

```
# Subset data by experimenter
data_e1 <- clean_data[clean_data$participant_id == 1
                      | clean_data$participant_id == 3
                      | clean_data$participant_id == 14
                      | clean_data$participant_id == 17
                      | clean_data$participant_id == 18
                      | clean_data$participant_id == 20
                      | clean_data$participant_id == 28
                      | clean_data$participant_id == 29
                      | clean_data$participant_id == 30, ]
data_e1$experimenter <- "1" # m

data_e2 <- clean_data[clean_data$participant_id == 2
                      | clean_data$participant_id == 4
                      | clean_data$participant_id == 5
                      | clean_data$participant_id == 6
                      | clean_data$participant_id == 10
                      | clean_data$participant_id == 11
                      | clean_data$participant_id == 12
                      | clean_data$participant_id == 13
                      | clean_data$participant_id == 21
                      | clean_data$participant_id == 22
                      | clean_data$participant_id == 26, ]
data_e2$experimenter <- "2" # p

data_e3 <- clean_data[clean_data$participant_id == 7
                      | clean_data$participant_id == 8
                      | clean_data$participant_id == 9
```

```
                          | clean_data$participant_id == 15
                          | clean_data$participant_id == 16
                          | clean_data$participant_id == 19
                          | clean_data$participant_id == 23
                          | clean_data$participant_id == 24
                          | clean_data$participant_id == 25
                          | clean_data$participant_id == 27, ]
data_e3$experimenter <- "3" # f

# bind the data together
experimenter_data <- rbind(data_e1, data_e2, data_e3)

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

```
## # A tibble: 3 x 2
##   experimenter mean_times
##   <chr>             <dbl>
## 1 1                 2939.
## 2 2                 2758.
## 3 3                 2553.
```

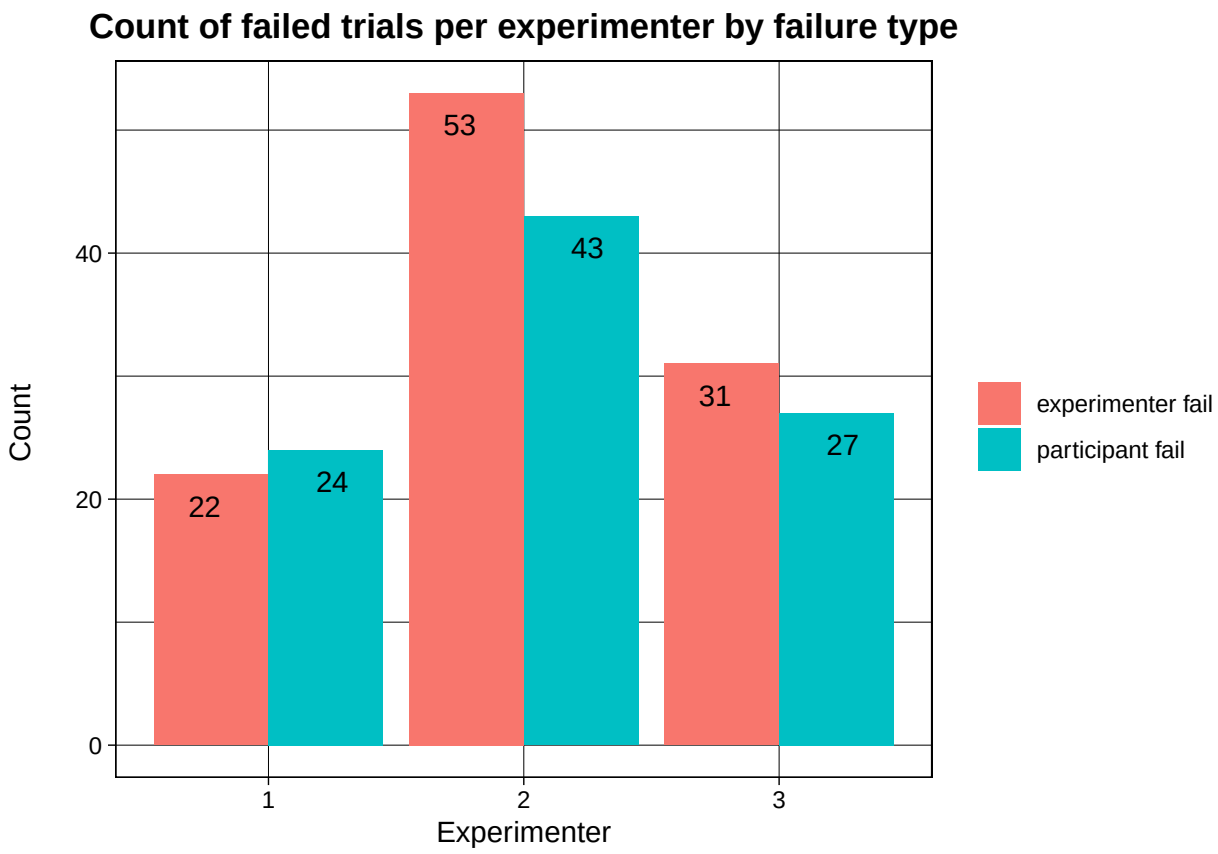
```
experimenter_data %>% filter(success == "success") %>%
  ggplot(aes(x = experimenter, y = time*1000)) +
  geom_boxplot(outlier.shape = NA) +
  labs(
    title = "Mean trial times per experimenter",
    x = "Experimenter",
    y = "Trial time (ms) \n",
    color = "experimenter"
    ) +
  theme_linedraw() +
  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: 6 x 3
## # Groups:   experimenter, success [6]
##   experimenter success     n
##   <chr>        <chr>   <int>
## 1 1            exFail     22
## 2 1            fail       24
## 3 2            exFail     53
## 4 2            fail       43
## 5 3            exFail     31
## 6 3            fail       27
```

```
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(1,3,14,17,18,20,28,29,30), "1"
                          ifelse(grasping_data$participant_id %in% c(2,4,5,6,10,11,12,13,21,22,26), "2

# add condition to df
grasping_data$order <- ifelse(grasping_data$participant_id %in% c(2,3,5,8,10,12,15,17,19,20,21,24,26,27
                              "tactile_first", "auditory_first")
grasping_data$order <- ifelse(grasping_data$participant_id %in% c(2,3,5,8,10,12,15,17,19,20,21,24,26,27
                              "tactile_first", "auditory_first")

# 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

# 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)
```

Model comparison & selection

In order to decide which variables are included in the LMM and which are excluded, we evaluate models, compare them, test for significant effects and investigate their complexity in terms of number of coefficients/parameters.

Decision train of thought: - variables: num_instructions, location, block, condition, pID, order, experimenter
- fixed effects we are interested in: condition (difference between sensory modalities), block (learning effects)
- num_instructions and location correlate (num is inferable from loc) and location gives a little more exact information (not only instructions = 2, but also whether top-left, top-right or bottom-left, bottom-right) but makes the model drastically more complex –> we would choose num_instructions as fixed –> we are not interested in it so we exclude
- crossed random effects: order and experimenter as some participants did have the order tactile_first, but not the other order, or did the experiment with "f" as first experimenter and not with the other, i.e. random effect 'order' occurs in each of the levels of random effect 'experimenter'
- we also model varying condition slopes of both random effects but not block slopes as the blocks in principle have the same structure, while the conditions are inherently different
- usually random effects should have at least 5 levels, but I want to include them
- in theory participantID could also be seen as random effect, but I think this is the variation we want to investigate in science -> check for interaction effects (correlations?) between variables and leave out variables without significant effect
- (model comparison: anova with model h0 with additional variable, h1 without additional var) –> can be replaced by lmerTest package anova testing

Resources: - LMM Tutorial: https://www.youtube.com/watch?v=QCqF-2E86r0 !!!
- Interpretation explanation: https://www.youtube.com/watch?v=yJnHmCMb1q4
- An LMM example (interaction explanation): https://www.youtube.com/watch?v=W8txfclM16U

```
# model testing: decide which variables to include/exclude (with lmerTest)
anova(model_complex <- lmer(time ~ condition * block * (condition | order) + (condition | experimenter)
                            data = grasping_data))
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##                   Sum Sq Mean Sq NumDF  DenDF F value    Pr(>F)
## condition        5842030 5842030     1    1.3  8.8341    0.1629
## block           26956404 4492734     6 3666.0  6.7937 3.536e-07 ***
## condition:block  5922529  987088     6 3666.0  1.4926    0.1764
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(model_complex <- lmer(time ~ condition * block * location * (condition | order) + (condition | ex
                            data = grasping_data))
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##                              Sum Sq   Mean Sq NumDF  DenDF   F value    Pr(>F)
```

```
## condition                   1754422   1754422     1    1.3    9.4735    0.1554
## block                      24170824   4028471     6 3554.0   21.7529 < 2.2e-16
## location                 1691309757 211413720     8 3554.0 1141.5920 < 2.2e-16
## condition:block             6662576   1110429     6 3554.0    5.9961 2.993e-06
## condition:location         37610151   4701269     8 3554.1   25.3859 < 2.2e-16
## block:location              9991222    208150    48 3554.3    1.1240    0.2592
## condition:block:location    8670787    180641    48 3554.3    0.9754    0.5214
##
## condition
## block                     ***
## location                  ***
## condition:block           ***
## condition:location        ***
## block:location
## condition:block:location
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# --> introduction of location makes interaction effect significant, but we are not interested in locat

# linear mixed model: complex (with significant interactions)
model_complex <- lmer(time ~
                condition * # include or exclude interaction?
                block +
                location +
                condition:location + # include or exclude interaction?
                # show correlated varying condition slopes and intercepts for each random effect
                (condition | order) +
                (condition | experimenter),
              data = grasping_data)

anova(model_complex) # with lmerTest package
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##                         Sum Sq   Mean Sq NumDF  DenDF   F value    Pr(>F)
## condition              1691700   1691700     1    1.3    9.1312    0.1567
## block                 23477427   3912905     6 3650.0   21.1204 < 2.2e-16 ***
## location            1717629460 214703682     8 3650.0 1158.8888 < 2.2e-16 ***
## condition:block        5886241    981040     6 3650.0    5.2953 1.903e-05 ***
## condition:location    37357034   4669629     8 3650.0   25.2049 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(model_complex)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula:
## time ~ condition * block + location + condition:location + (condition |
##     order) + (condition | experimenter)
##    Data: grasping_data
##
## REML criterion at convergence: 54886.1
##
## Scaled residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -3.1966 -0.6469 -0.0610  0.5338  5.8141
##
## Random effects:
##  Groups       Name             Variance Std.Dev. Corr
##  experimenter (Intercept)       44830   211.7
##               conditiontactile   9604    98.0    -0.51
##  order        (Intercept)        1657    40.7
##               conditiontactile  44928   212.0    -1.00
##  Residual                      185267   430.4
## Number of obs: 3685, groups:  experimenter, 3; order, 2
##
## Fixed effects:
##                              Estimate Std. Error        df t value Pr(>|t|)
## (Intercept)                  3289.807    131.299     2.609  25.056 0.000350 ***
## conditiontactile              880.044    169.814     1.599   5.182 0.055122 .
## block3                       -104.586     37.154  3650.002  -2.815 0.004905 **
## block4                       -129.158     37.119  3650.001  -3.480 0.000508 ***
## block5                       -158.439     37.162  3650.001  -4.263 2.06e-05 ***
## block6                       -134.672     37.099  3650.001  -3.630 0.000287 ***
## block7                       -125.487     37.155  3650.001  -3.377 0.000740 ***
## block8                       -145.995     37.199  3650.003  -3.925 8.84e-05 ***
## location2                    -988.569     41.486  3650.017 -23.829  < 2e-16 ***
## location3                     -12.833     42.316  3650.046  -0.303 0.761707
## location4                    -866.209     41.979  3650.017 -20.634  < 2e-16 ***
## location5                   -1880.719     41.917  3650.003 -44.867  < 2e-16 ***
## location6                    -915.852     42.505  3650.007 -21.547  < 2e-16 ***
## location7                     -60.571     41.731  3650.046  -1.451 0.146735
## location8                   -1029.788     41.768  3650.015 -24.655  < 2e-16 ***
## location9                     -75.153     42.423  3650.059  -1.772 0.076559 .
## conditiontactile:block3        -8.547     53.171  3650.016  -0.161 0.872300
## conditiontactile:block4       -49.802     53.119  3650.008  -0.938 0.348537
## conditiontactile:block5       -94.382     53.294  3650.002  -1.771 0.076651 .
## conditiontactile:block6      -158.456     53.251  3650.004  -2.976 0.002943 **
## conditiontactile:block7      -211.104     53.147  3650.011  -3.972 7.26e-05 ***
## conditiontactile:block8      -190.269     53.073  3650.013  -3.585 0.000341 ***
## conditiontactile:location2    -75.490     61.413  3650.062  -1.229 0.219071
## conditiontactile:location3    -20.313     61.698  3650.049  -0.329 0.742003
## conditiontactile:location4   -537.428     60.954  3650.025  -8.817  < 2e-16 ***
## conditiontactile:location5   -570.244     60.743  3650.024  -9.388  < 2e-16 ***
## conditiontactile:location6   -378.633     61.142  3650.018  -6.193 6.57e-10 ***
## conditiontactile:location7   -362.583     61.028  3650.017  -5.941 3.09e-09 ***
## conditiontactile:location8   -412.256     60.537  3650.084  -6.810 1.14e-11 ***
## conditiontactile:location9   -273.235     61.235  3650.021  -4.462 8.36e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
```

```
coef(model_complex)
```

```
## $experimenter
##   (Intercept) conditiontactile    block3   block4    block5   block6    block7
## 1    3455.792         910.3807 -104.5865 -129.158 -158.4392 -134.672 -125.4866
## 2    3361.424         773.5237 -104.5865 -129.158 -158.4392 -134.672 -125.4866
```

```
## 3    3052.204         956.2262 -104.5865 -129.158 -158.4392 -134.672 -125.4866
##      block8 location2 location3 location4 location5 location6 location7
## 1 -145.9951 -988.5691 -12.83269 -866.2095 -1880.719 -915.8525 -60.57133
## 2 -145.9951 -988.5691 -12.83269 -866.2095 -1880.719 -915.8525 -60.57133
## 3 -145.9951 -988.5691 -12.83269 -866.2095 -1880.719 -915.8525 -60.57133
##   location8 location9 conditiontactile:block3 conditiontactile:block4
## 1 -1029.788 -75.15288              -8.547207              -49.80208
## 2 -1029.788 -75.15288              -8.547207              -49.80208
## 3 -1029.788 -75.15288              -8.547207              -49.80208
##   conditiontactile:block5 conditiontactile:block6 conditiontactile:block7
## 1               -94.38159              -158.4559              -211.1041
## 2               -94.38159              -158.4559              -211.1041
## 3               -94.38159              -158.4559              -211.1041
##   conditiontactile:block8 conditiontactile:location2 conditiontactile:location3
## 1               -190.2689                  -75.49027                  -20.31267
## 2               -190.2689                  -75.49027                  -20.31267
## 3               -190.2689                  -75.49027                  -20.31267
##   conditiontactile:location4 conditiontactile:location5
## 1                  -537.4278                  -570.2437
## 2                  -537.4278                  -570.2437
## 3                  -537.4278                  -570.2437
##   conditiontactile:location6 conditiontactile:location7
## 1                  -378.6326                  -362.5826
## 2                  -378.6326                  -362.5826
## 3                  -378.6326                  -362.5826
##   conditiontactile:location8 conditiontactile:location9
## 1                  -412.2556                  -273.2352
## 2                  -412.2556                  -273.2352
## 3                  -412.2556                  -273.2352
##
## $order
##                (Intercept) conditiontactile    block3   block4    block5
## auditory_first    3318.487         730.6865 -104.5865 -129.158 -158.4392
## tactile_first     3261.126        1029.4005 -104.5865 -129.158 -158.4392
##                   block6    block7    block8 location2 location3 location4
## auditory_first -134.672 -125.4866 -145.9951 -988.5691 -12.83269 -866.2095
## tactile_first  -134.672 -125.4866 -145.9951 -988.5691 -12.83269 -866.2095
##                location5 location6 location7 location8 location9
## auditory_first -1880.719 -915.8525 -60.57133 -1029.788 -75.15288
## tactile_first  -1880.719 -915.8525 -60.57133 -1029.788 -75.15288
##                conditiontactile:block3 conditiontactile:block4
## auditory_first               -8.547207              -49.80208
## tactile_first                -8.547207              -49.80208
##                conditiontactile:block5 conditiontactile:block6
## auditory_first               -94.38159              -158.4559
## tactile_first                -94.38159              -158.4559
##                conditiontactile:block7 conditiontactile:block8
## auditory_first               -211.1041              -190.2689
## tactile_first                -211.1041              -190.2689
##                conditiontactile:location2 conditiontactile:location3
## auditory_first                  -75.49027                  -20.31267
## tactile_first                   -75.49027                  -20.31267
##                conditiontactile:location4 conditiontactile:location5
## auditory_first                  -537.4278                  -570.2437
```

```
## tactile_first                   -537.4278                   -570.2437
##                conditiontactile:location6 conditiontactile:location7
## auditory_first                  -378.6326                   -362.5826
## tactile_first                   -378.6326                   -362.5826
##                conditiontactile:location8 conditiontactile:location9
## auditory_first                  -412.2556                   -273.2352
## tactile_first                   -412.2556                   -273.2352
##
## attr(,"class")
## [1] "coef.mer"
```

```
# linear mixed model: simple (without interactions)
model_occam <- lmer(time ~
                condition +
                block +
                location +
                (condition | order) +
                (condition | experimenter),
              data = grasping_data)

anova(model_occam)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##               Sum Sq   Mean Sq NumDF  DenDF   F value Pr(>F)
## condition    1753998   1753998     1    1.3    8.9317 0.1577
## block       21765581   3627597     6 3664.0   18.4724 <2e-16 ***
## location  1710743391 213842924     8 3664.0 1088.9279 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(model_occam)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula:
## time ~ condition + block + location + (condition | order) + (condition |
##     experimenter)
##    Data: grasping_data
##
## REML criterion at convergence: 55247.1
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.5200 -0.6564 -0.0697  0.5430  6.1102
##
## Random effects:
##  Groups       Name             Variance Std.Dev. Corr
##  experimenter (Intercept)       44890   211.87
##               conditiontactile   9898    99.49   -0.50
##  order        (Intercept)        1586    39.82
##               conditiontactile  44809   211.68   -1.00
##  Residual                      196379   443.15
## Number of obs: 3685, groups:  experimenter, 3; order, 2
##
## Fixed effects:
```

```
##                   Estimate Std. Error        df t value Pr(>|t|)
## (Intercept)       3467.113    128.975     2.420  26.882 0.000467 ***
## conditiontactile   481.141    160.993     1.295   2.989 0.157676
## block3             -97.058     27.329  3664.013  -3.551 0.000388 ***
## block4            -148.325     27.309  3664.008  -5.431 5.95e-08 ***
## block5            -188.673     27.378  3664.003  -6.891 6.48e-12 ***
## block6            -204.099     27.368  3664.005  -7.458 1.09e-13 ***
## block7            -218.549     27.319  3664.014  -8.000 1.65e-15 ***
## block8            -234.394     27.285  3664.005  -8.591  < 2e-16 ***
## location2        -1019.515     31.474  3664.027 -32.392  < 2e-16 ***
## location3          -16.714     31.667  3664.025  -0.528 0.597658
## location4        -1117.535     31.276  3664.020 -35.731  < 2e-16 ***
## location5        -2152.326     31.196  3664.013 -68.995  < 2e-16 ***
## location6        -1093.197     31.342  3664.014 -34.879  < 2e-16 ***
## location7         -231.026     31.326  3664.087  -7.375 2.02e-13 ***
## location8        -1223.266     31.084  3664.028 -39.353  < 2e-16 ***
## location9         -198.279     31.446  3664.123  -6.305 3.22e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
```

```
coef(model_occam)
```

```
## $experimenter
##   (Intercept) conditiontactile    block3    block4    block5    block6
## 1    3632.433         513.5184 -97.05838 -148.3255 -188.6725 -204.0995
## 2    3539.654         372.7177 -97.05838 -148.3255 -188.6725 -204.0995
## 3    3229.252         557.1870 -97.05838 -148.3255 -188.6725 -204.0995
##      block7    block8 location2 location3 location4 location5 location6
## 1 -218.5488 -234.3938 -1019.515  -16.7143 -1117.535 -2152.326 -1093.197
## 2 -218.5488 -234.3938 -1019.515  -16.7143 -1117.535 -2152.326 -1093.197
## 3 -218.5488 -234.3938 -1019.515  -16.7143 -1117.535 -2152.326 -1093.197
##   location7 location8 location9
## 1 -231.0262 -1223.266 -198.2789
## 2 -231.0262 -1223.266 -198.2789
## 3 -231.0262 -1223.266 -198.2789
##
## $order
##                (Intercept) conditiontactile    block3    block4    block5
## auditory_first    3495.163         332.0411 -97.05838 -148.3255 -188.6725
## tactile_first     3439.063         630.2410 -97.05838 -148.3255 -188.6725
##                    block6    block7    block8 location2 location3 location4
## auditory_first -204.0995 -218.5488 -234.3938 -1019.515  -16.7143 -1117.535
## tactile_first  -204.0995 -218.5488 -234.3938 -1019.515  -16.7143 -1117.535
##                location5 location6 location7 location8 location9
## auditory_first -2152.326 -1093.197 -231.0262 -1223.266 -198.2789
## tactile_first  -2152.326 -1093.197 -231.0262 -1223.266 -198.2789
##
## attr(,"class")
## [1] "coef.mer"
```

```
# alternative (simpler) model: 'number of instructions' instead 'location' for less parameters
model_alt1 <- lmer(time ~
                condition +
```

```
                block +
                num_instructions +
                (condition | order) +
                (condition | experimenter),
              data = grasping_data)
summary(model_alt1)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: time ~ condition + block + num_instructions + (condition | order) +
##     (condition | experimenter)
##    Data: grasping_data
##
## REML criterion at convergence: 55697.3
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.7570 -0.6370 -0.0581  0.5324  6.0592
##
## Random effects:
##  Groups       Name             Variance Std.Dev. Corr
##  experimenter (Intercept)       44156   210.13
##               conditiontactile  11438   106.95   -0.47
##  order        (Intercept)        1436    37.89
##               conditiontactile  40913   202.27   -1.00
##  Residual                      218583   467.53
## Number of obs: 3685, groups:  experimenter, 3; order, 2
##
## Fixed effects:
##                   Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)       1322.143    127.964    2.425  10.332  0.00461 **
## conditiontactile   476.141    156.550    1.377   3.041  0.14441
## block3            -105.293     28.802 3670.012  -3.656  0.00026 ***
## block4            -153.716     28.806 3670.006  -5.336 1.01e-07 ***
## block5            -186.239     28.873 3670.002  -6.450 1.26e-10 ***
## block6            -213.347     28.837 3670.004  -7.398 1.70e-13 ***
## block7            -217.262     28.817 3670.012  -7.539 5.91e-14 ***
## block8            -243.306     28.764 3670.004  -8.459  < 2e-16 ***
## num_instructions2 1036.995     25.670 3670.008  40.398  < 2e-16 ***
## num_instructions3 2006.747     25.628 3670.025  78.303  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) cndtnt block3 block4 block5 block6 block7 block8 nm_ns2
## conditntctl -0.373
## block3      -0.109  0.000
## block4      -0.115  0.000  0.500
## block5      -0.111  0.000  0.499  0.499
## block6      -0.112  0.000  0.500  0.499  0.498
## block7      -0.112  0.000  0.500  0.500  0.499  0.499
## block8      -0.114  0.000  0.501  0.501  0.499  0.500  0.500
## nm_nstrctn2 -0.160  0.000 -0.011  0.014 -0.002  0.005  0.001  0.013
## nm_nstrctn3 -0.160  0.001 -0.026  0.016 -0.009 -0.010 -0.004  0.008  0.796
```

```
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
```

```
# alternative (simplest) model: disregard 'block' entirely
model_alt2 <- lmer(time ~
                condition +
                num_instructions +
                (condition | order) +
                (condition | experimenter),
              data = grasping_data)
summary(model_alt2)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula:
## time ~ condition + num_instructions + (condition | order) + (condition |
##     experimenter)
##    Data: grasping_data
##
## REML criterion at convergence: 55847.4
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.6821 -0.6393 -0.0560  0.5280  6.3192
##
## Random effects:
##  Groups       Name             Variance Std.Dev. Corr
##  experimenter (Intercept)       44153   210.13
##               conditiontactile  11401   106.78   -0.47
##  order        (Intercept)        1432    37.84
##               conditiontactile  40927   202.31   -1.00
##  Residual                      224305   473.61
## Number of obs: 3685, groups:  experimenter, 3; order, 2
##
## Fixed effects:
##                   Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)       1160.221    126.615    2.320   9.163  0.00716 **
## conditiontactile   475.854    156.554    1.375   3.040  0.14477
## num_instructions2 1040.246     25.992 3676.009  40.022  < 2e-16 ***
## num_instructions3 2008.475     25.933 3676.025  77.449  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) cndtnt nm_ns2
## conditntctl -0.378
## nm_nstrctn2 -0.163  0.000
## nm_nstrctn3 -0.164  0.001  0.796
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
```

Final LMM

```
model <- lmer(time ~
                condition +
                block +
                (condition | order) +
                (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 | order) + (condition |
##     experimenter)
##    Data: grasping_data
##
## REML criterion at convergence: 59782.7
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.5994 -0.6379 -0.0392  0.6879  4.3137
##
## Random effects:
##  Groups       Name             Variance Std.Dev. Corr
##  experimenter (Intercept)       46460   215.55
##               conditiontactile   8421    91.76   -0.56
##  order        (Intercept)        1107    33.27
##               conditiontactile  41233   203.06   -1.00
##  Residual                      661833   813.53
## Number of obs: 3685, groups:  experimenter, 3; order, 2
##
## Fixed effects:
##                  Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)      2666.940    132.171    2.412  20.178 0.000951 ***
## conditiontactile  459.863    155.388    1.262   2.959 0.163987
## block3            -30.725     50.094 3672.050  -0.613 0.539687
## block4           -188.168     50.118 3672.034  -3.755 0.000176 ***
## block5           -156.901     50.237 3672.015  -3.123 0.001803 **
## block6           -169.393     50.165 3672.023  -3.377 0.000741 ***
## block7           -201.635     50.141 3672.053  -4.021 5.90e-05 ***
## block8           -249.406     50.047 3672.024  -4.983 6.53e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) cndtnt block3 block4 block5 block6 block7
## conditntctl -0.363
## block3      -0.190  0.000
## block4      -0.190  0.000  0.500
## block5      -0.189  0.000  0.499  0.499
## block6      -0.190  0.000  0.500  0.500  0.498
## block7      -0.189  0.000  0.500  0.500  0.499  0.499
## block8      -0.190 -0.001  0.501  0.501  0.499  0.500  0.500
```

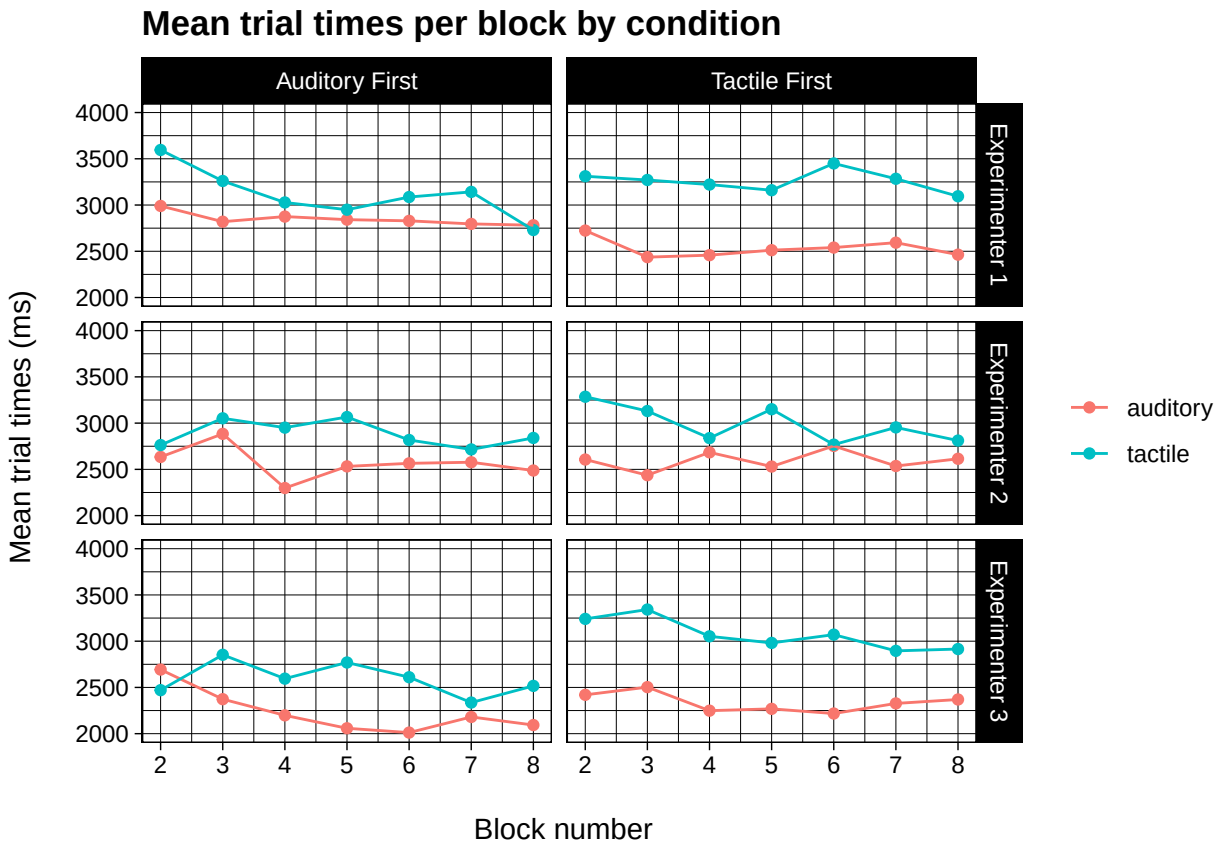
```
## optimizer (nloptwrap) convergence code: 0 (OK)
## Model failed to converge with max|grad| = 0.0021096 (tol = 0.002, component 1)
```

```
#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

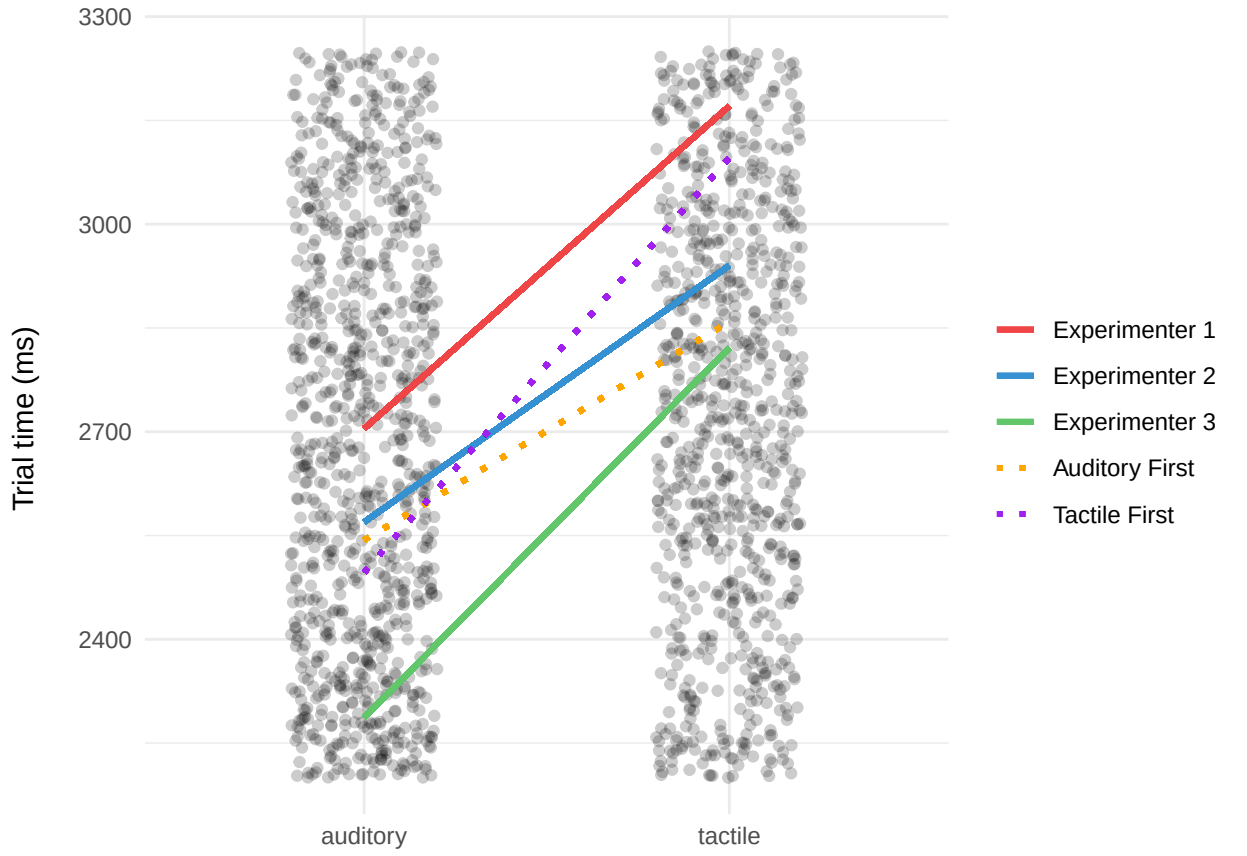
```
grasping_data$block <- as.double(grasping_data$block)

# 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.jpeg"), plot = lmm_plot, dpi = 600)
```

```
exp <- ranef(model)$experimenter
ord <- ranef(model)$order

mean_aud <- grasping_data %>% filter(condition == "auditory") %>% summarize(mean = mean(time))
mean_tac <- grasping_data %>% filter(condition == "tactile") %>% summarize(mean = mean(time))

colors <- c("Experimenter 1"="#f04546","Experimenter 2"="#3591d1","Experimenter 3"="#62c76b", "Auditory
linetypes <- c("Experimenter 1"="solid","Experimenter 2"="solid","Experimenter 3"="solid", "Auditory Fi
legend_order <- c("Experimenter 1", "Experimenter 2", "Experimenter 3", "Auditory First", "Tactile Firs

lmm_ranef <- grasping_data %>% ggplot(aes(x = condition, y = time)) +
  geom_point(position = position_jitter(width=0.2), alpha=0.2) +
  geom_segment(aes(x = "auditory", xend = "tactile",
                   y = exp$`(Intercept)`[1] + mean_aud$mean,
                   yend = exp$`(Intercept)`[1] + mean_tac$mean + exp$conditiontactile[1], color = "Expe
  geom_segment(aes(x = "auditory", xend = "tactile",
                   y = exp$`(Intercept)`[2] + mean_aud$mean,
                   yend = exp$`(Intercept)`[2] + mean_tac$mean + exp$conditiontactile[2], color = "Expe
  geom_segment(aes(x = "auditory", xend = "tactile",
                   y = exp$`(Intercept)`[3] + mean_aud$mean,
                   yend = exp$`(Intercept)`[3] + mean_tac$mean + exp$conditiontactile[3], color = "Expe
  geom_segment(aes(x = "auditory", xend = "tactile",
                   y = ord$`(Intercept)`[1] + mean_aud$mean,
                   yend = ord$`(Intercept)`[1] + mean_tac$mean + ord$conditiontactile[1], color = "Audi
  geom_segment(aes(x = "auditory", xend = "tactile",
```

```
                  y = ord$`(Intercept)`[2] + mean_aud$mean,
                  yend = ord$`(Intercept)`[2] + mean_tac$mean + ord$conditiontactile[2], color = "Tacti
  scale_color_manual(name = NULL, values = colors, breaks = legend_order) +
  scale_linetype_manual(name = NULL, values = linetypes, breaks = legend_order) +
  labs(
    #title = "Random effects experimenter and order on trial times",
    x = NULL,
    y = "Trial time (ms) \n"
    ) +
    scale_y_continuous(limits = c(2200, 3250)) +
    theme_minimal()

lmm_ranef
```




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