

```

# R course for beginners
# Week 7
# assignment by Shira Zadok, id 318958311

#### Creating Raw_Data ----
library(dplyr)
file_names <- dir("stroop_data")

df <- NULL
for (file in file_names) {
  temp_df <- read.csv(file.path("stroop_data", file))
  df <- rbind(df, temp_df)
}

#saving only relevant columns:
df <- df |>
  mutate(
    task = ifelse(grepl("word", condition), "word_reading", "ink_naming"),
    congruency = ifelse(grepl("incong", condition), "Incongruent", "Congruent"),
    acc = (correct_response == participant_response)*1,

    #making sure we have the right type:
    subject = as.factor(subject),
    task = factor(task, levels = c("word_reading", "ink_naming")),
    congruency = factor(congruency, levels = c("Congruent", "Incongruent")),
    block = as.numeric(block),
    trial = as.numeric(trial),
    acc = as.numeric(acc),
    rt = as.numeric(rt)
  ) |>
  select(subject, task, congruency, block, trial, acc, rt)

summary(df)

#checking contrasts:
contrasts(df$task) <- c(0,1)
contrasts(df$task)
contrasts(df$congruency)

#save
save(df, file = "./raw_data.rdata")

#Question 2:
#### Creating Filtered Data: ----
load("./raw_data.rdata")
library(tidyverse)
library(dplyr)

cat("The number of participants it:", length(unique(df$subject)))

df <- df |>
  na.omit(df) |>
  filter(rt >= 300 & rt <= 3000)

#calculating total trials number:
per_sum <- df |>
  group_by(subject) |>
  summarise(percentage = 1 - (n()/400))

#mean and std for total sample:
cat("The mean percent of trial removal is:", mean(per_sum$percentage),
    "and the sd is:", sd(per_sum$percentage))

#save:
save(df, file = "./filtered_data.rdata")

```

#Question 3:

Descriptive Statistics: ----

```
load("./filtered_data.rdata")
```

```
df_sum <- df |>
  group_by(task, congruency) |>
  summarise(number = n(),
            mean_acc = mean(acc),
            mean_rt = mean(rt),
            sd_acc = sd(acc),
            sd_rt = sd(rt))
```

#plotting:

```
library(ggplot2)
library(patchwork)
```

```
p1 = ggplot(df_sum, aes(x = task, y = mean_acc, fill = congruency)) +
  geom_bar(stat = "identity", position = position_dodge(0.9), width = 0.7) +
  geom_errorbar(aes(ymin = mean_acc - sd_acc, ymax = mean_acc + sd_acc),
               position = position_dodge(0.9),
               width = 0.3) +
  labs(title = "Mean Accuracy per Task and Congruency",
       x = "Task", y = "Accuracy (%)") +
  theme_minimal()
```

```
p2 = ggplot(df, aes(x = task, y = rt, color = congruency)) +
  geom_jitter(position = position_jitter(width = 0.3),
             size = 2, alpha = 0.2) +
  geom_point(data = df_sum, aes(x = task, y = mean_rt, group = congruency,
                               color = congruency),
            position = position_dodge(1), size = 3) +
  geom_errorbar(data = df_sum, aes(x = task, y = mean_rt,
                                   ymin = mean_rt - sd_rt, ymax = mean_rt + sd_rt,
                                   color = congruency), position = position_dodge(1),
               width = 0.2) +
  labs(title = "RT per Task", x = "Task", y = "RT (msec)") +
  theme_minimal()
```

Regression ----

```
library(lme4)
model <- lmer(rt ~ congruency + task + (1|subject), data = df)
summary(model)
```