```r
# R course for beginners
# Week 7
# assignment by Amir Mano, id 205779788

#### create row data ----
rm(list = ls()) #or Ctrl + Shift + F10 & Ctrl + L
library(tidyverse)

# load data
fnames <- dir("stroop_data")
df = data.frame()
for (ind in c(1:length(fnames))){
  temp_df <- read.csv(file.path('stroop_data',fnames[ind]))
  df <- rbind(df, temp_df)
}

# organize data
df <- df |> mutate(task = ifelse(grepl("ink", condition), "ink_naming", "word_reading"))
df <- df |> mutate(congureancy = ifelse(grepl("incong", condition), "incongureant", "congureant"))
df <- df |> mutate(accuracy = ifelse(correct_response==participant_response, 1, 0))
df <- df |> mutate(task = factor(task))
df <- df |> mutate(congureancy = factor(congureancy))
df <- df |> select(subject, block, trial, task, congureancy, accuracy, rt)

# make sure the contrast are in alphabetical order
contrasts(df$task) <- contr.treatment(2)
contrasts(df$congureancy) <- contr.treatment(2)

save(df, file='raw_data.RData')
```

```r
#### create filtered data ----
rm(list = ls())
load('raw_data.rdata')

# remove non-relevat data
N <- length(unique(df$subject))
remove_indexes <-is.na(df$rt) | df$rt>3000 | df$rt<300
df<- df|>filter(!remove_indexes)

# calculate ratio
count_trial <- c()
for (subj in unique(df$subject)){
  count_trial <- c(count_trial, sum(df$subject==subj))
}
ratio <- count_trial/400
cat('Average ratio:', mean(ratio), '\n', 'SD ratio:', sd(ratio), '\n')

save(df, file='filtered_data.RData')

#### statistics and plotting ----
load('filtered_data.rdata')
df <- df |> mutate(group = interaction(task, congureancy))

stats <- df |>
  group_by(group) |>
  summarize(means = mean(rt), SDs = sd(rt))

```

```r
# plotting with ggplot
library(ggplot2)
ggplot(df, aes(x = group, y = rt, color = group)) +
  geom_jitter(position = position_jitter(width = 0.15, height = 5), size = 1.5, alpha = 0.3) +
  geom_point(data = stats, aes(x = group, y = means, group = group),
             position = position_dodge(1), size = 3, color = "darkred") +
  geom_errorbar(data = stats,aes(x = group, y = means, group = group, ymin = means - SDs, ymax = means + SDs),
                position = position_dodge(1), width = 0.5, color = 'darkred') +
  labs(x = 'Condition', y = 'Reaction Time (ms)') +
  theme_minimal()  +
  theme(axis.ticks.y = element_blank(), axis.text.y = element_blank())
```