

```
# Introduction to R
# Week 6
# Assignment by Nava Peter (211311147)
```

A. Loading and Combining the Different Files

```
# 1. Using For Loop for That
files_names <- dir("stroop_data") # Creates a variable that is called "files_names" that gets its information from the data in the
computer directory
df <- data.frame() # creates an empty dataframe called df

for (f in files_names) { # Creates a variable called f that represents an individual document in the file we transported
  temp <- read.csv(paste0("stroop_data/", f)) # reads every document and saves it under the temp variable
  df <- rbind(df, temp) # adds to the empty dataframe the temp document that was read
}
```

#2. Counts the numbers of participants and conditions

```
names(df) # checks the names of columns in df
subject_nums <- length(unique(df$subject)) # Checks the numbers of subjects
subject_condition <- length(unique(df$condition)) # Checks the number of conditions in Condition
print(subject_nums) # Prints the numbers of subjects
print(subject_condition) #Prints the numbers of Conditions
```

#3. Numbers and Percentage of NaN

```
count_NA_rt <- sum(is.na(df$rt)) # Counts the numbers of NAs in rt
print(count_NA_rt) # prints the number of conditions in rt
percentage_NA_rt <- count_NA_rt/length(df$rt)*100 # Calculates the percentage of NAs in rt
print(percentage_NA_rt) # Prints out the numbers of NAs in rts
```

#4. Creates a Histogramma for Reaction Times for Congruent and Incongruent Conditions

```
unique(df$condition) # returns the names of conditions in condition
v_rt_congruent = df$rt [df$condition == "congruent"] # Creates a vector that has only the rts of people in the congruent condition
v_rt_incongruent = df$rt[df$condition== "incongruent"] # Creates a vector that has only the rts of people in the confruent condition
hist(v_rt_congruent) # Creates a histogramma of the rts in the congruent condition
hist(v_rt_incongruent) #Creates a histogramma of the rts in the incongruent condition
```

B. Cleaning Data and Identifying Outliers

```
# 1. Creates Logical Vector that Checks Extreme
extreme_rt <- function(vector_rt) { # Creates a function that receives a vector of rts and marks extreme rts
  mean_rt <- mean(vector_rt) # Creates a value that represents the mean of rts
  sd_rt <- sd(vector_rt) # Calculates the sd of all the rts
  is_extreme = vector_rt > mean_rt + 2 * sd_rt # Creates a logical vector that marks if something is extreme or not
  return(is_extreme)
}
```

```

#2. For Loop that Checks Extreme
all_subjects_rts <- logical(nrow(df)) # Logical vector filled with False for empty rts
subject_ids <- unique(df$subject) # The unique subjects

for (subject_num in subject_ids) { # Marks extreme values in reaction times, receives the subject number among the numbers of unique subjects
  rows_of_s <- df$subject == subject_num # Marks only the rows of the subject
  subject_rt <- df$rt[rows_of_s] # Takes out from df only the subject reaction times
  is_extreme_s <- extreme_rt(subject_rt) # Says which of the subject rows are extreme or not
  all_subjects_rts [rows_of_s] <- is_extreme_s # Inserts if the subject rts are extreme only in the rows of their reaction times
}

df$extreme <- all_subjects_rts # Creates a new vector in DF that marks what values are extreme and what aren't

# C. Information Presenting

#1. Counts and Prints out the number of outliers
count_outliers_s <- tapply(df$extreme, df$subject, sum) # Counts the number of outliers per subject
print(count_outliers_s) # Prints the number of outliers per subject

#2. Creates a Histogramma for the COngruent and Incongruent Conditions with No Outliers
rt_congruent_clean <- df$rt[df$condition=="congruent" & !df$extreme] # Creates a vector of reaction times for the Congruent condition and not extreme
rt_incongruent_clean <- df$rt[df$condition=="incongruent" & !df$extreme] # Creates a vector of reaction times for the Incongruent condition and not extreme

hist(rt_congruent_clean) |> # Creates a histogramma for congruent condition with no outliers
hist(rt_incongruent_clean) # Creates a histogramma for incongruent condition with no outliers

```