Reproducible research Project 1

Glauco Stradiotto

11/12/2020

##Setting global echo

```
knitr::opts_chunk$set(echo = TRUE)
```

Library

```
library(ggplot2)
library(dplyr)
library(lubridate)
```

Data acquisition

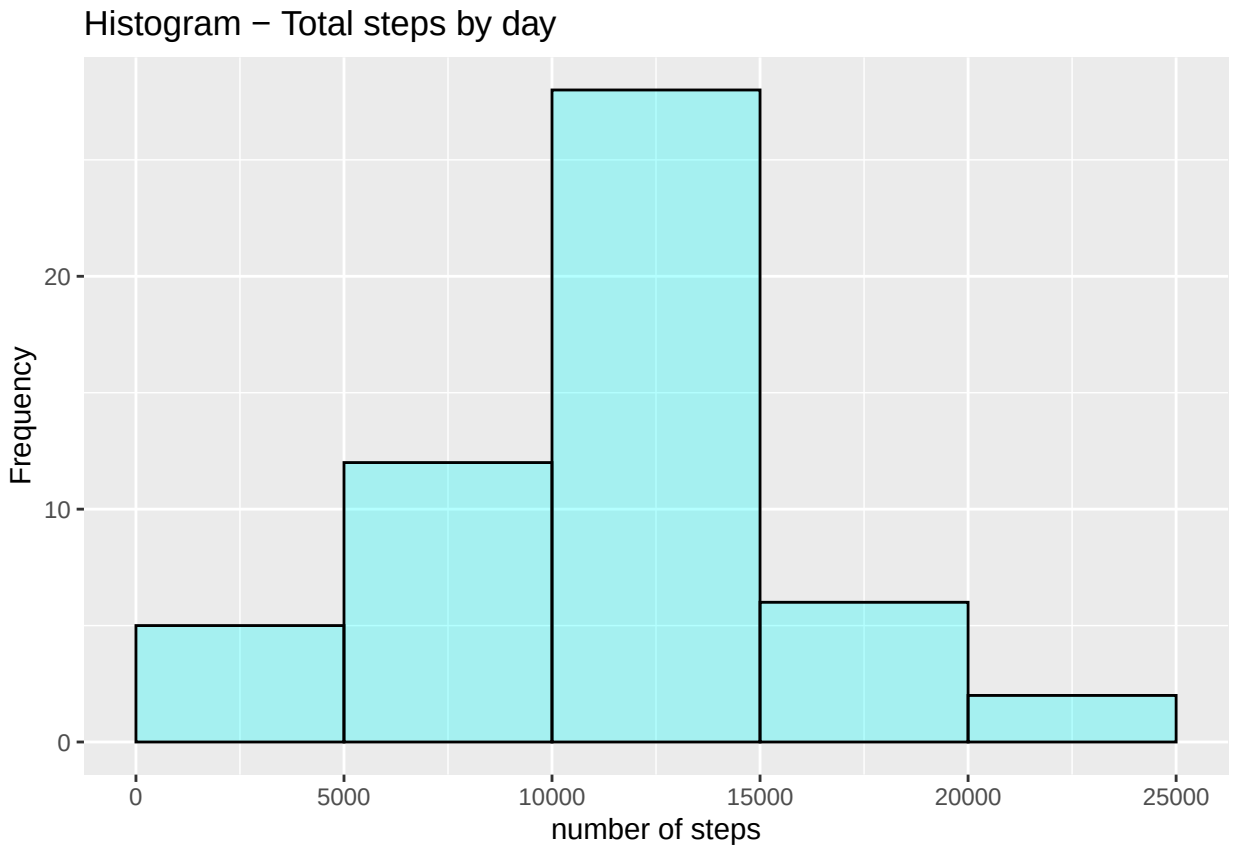
1. Code for reading in the dataset and/or processing the data

```
original_wd <- getwd()
setwd(paste0(original_wd,'/R/local/'))

file_URL <- 'https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip'

file_dest <- './dataset/repdata_data_activity.zip'

download.file(url = file_URL, file_dest, method = 'curl')

unzip('./dataset/repdata_data_activity.zip', exdir = './dataset/')

activity <- read.csv("./dataset/activity.csv")
```

Histogram Plotting

2. Histogram of the total number of steps taken each day

```
steps_histogram <- activity %>%
  filter(!is.na(steps)) %>%
  group_by(date) %>%
  summarize(total_steps = sum(steps))

ggplot(steps_histogram, aes(total_steps)) +
  geom_histogram(breaks = seq(0,25000, by = 5000), color = "black",fill = "cyan", alpha = 0.3) +
  labs(title = "Histogram - Total steps by day", x = "number of steps", y = "Frequency")
```



Data KPI

3. Mean and median number of steps taken each day

```
steps_mean <- mean(steps_histogram$total_steps)
print_mean <- paste0("The average number of steps is: ",steps_mean)

print_mean
```

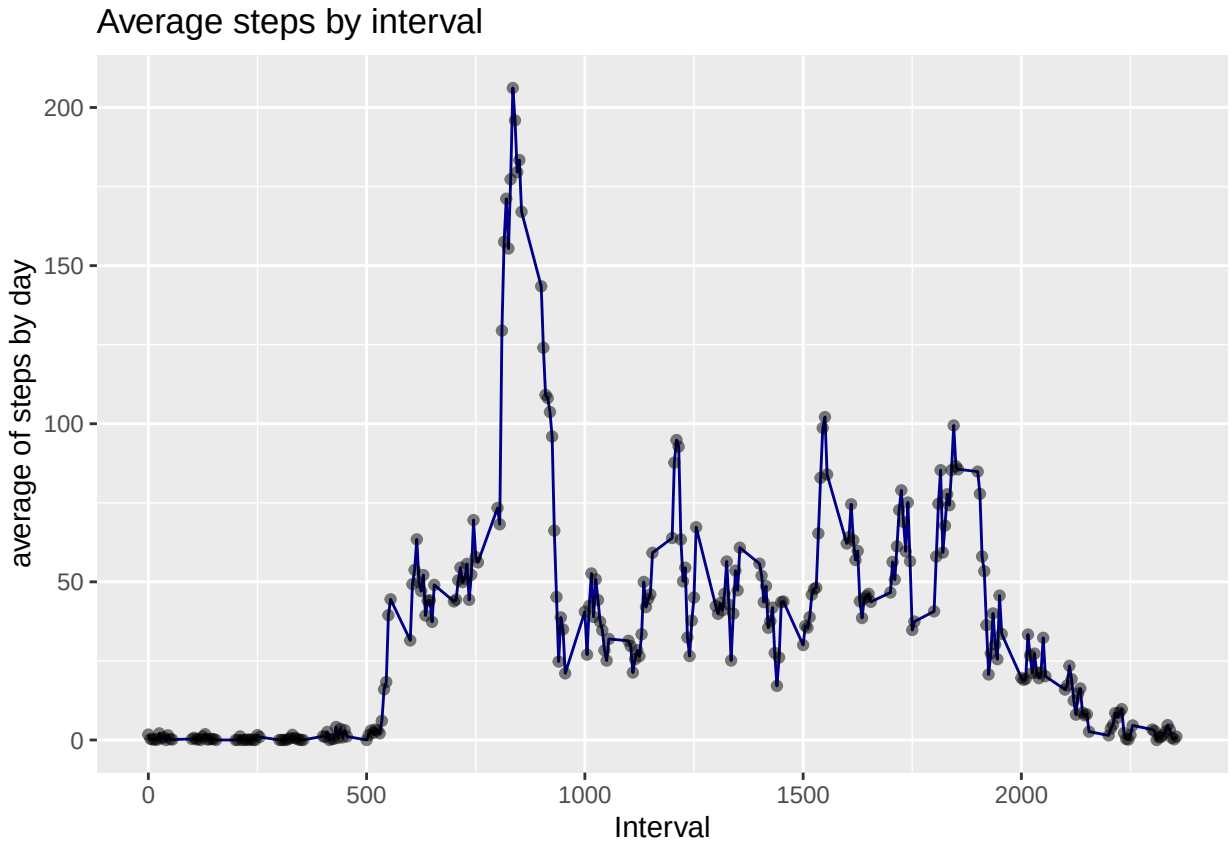
```
## [1] "The average number of steps is: 10766.1886792453"
```

```
steps_median <- median(steps_histogram$total_steps)
print_median <- paste0("The median of steps is: ", steps_median)

print_median
```

```
## [1] "The median of steps is: 10765"
```

##Series Plotting 4. Time series plot of the average number of steps taken

```
steps_series <- activity %>%
  filter(!is.na(steps)) %>%
  group_by(interval) %>%
  summarize(avg_steps = mean(steps))

ggplot(steps_series, aes(x = interval,y = avg_steps)) +
  geom_line(color = "dark blue") +
  geom_point(color = "black", alpha = 0.5) +
  labs(title = "Average steps by interval", x = "Interval", y = "average of steps by day")
```



Max Interval

5. The 5-minute interval that, on average, contains the maximum number of steps

```
max_steps <- activity %>%
  filter(!is.na(steps)) %>%
  group_by(interval) %>%
  summarize(avg_step = mean(steps)) %>%
  top_n(1)

max_interval <- first(max_steps)

max_interval
```

```
## [1] 835
```

Missing Values

6. Code to describe and show a strategy for imputing missing data

```
total_na <- sum(is.na(activity$steps))

print_na <- paste0("The total number of missing value is: ",total_na)

print_na
```

```
## [1] "The total number of missing value is: 2304"
```

Devise a strategy for filling in all of the missing values in the dataset. The strategy does not need to be

sophisticated. For example, you could use the mean/median for that day, or the mean for that 5-minute interval, etc.

The strategy uses the average of that specific interval across the whole period and substitute the steps by its average when necessary.

```
max_steps_total <- activity %>%
  filter(!is.na(steps)) %>%
  group_by(interval) %>%
  summarize(avg_step = mean(steps))

activity_replaced <- activity %>%
  inner_join(max_steps_total, by = "interval") %>%
  mutate(steps = if_else(is.na(steps), as.numeric(avg_step), as.numeric(steps))) %>%
  select(steps, date, interval)

check <- sum(is.na(activity_replaced$steps))

print_replace <- paste0("The total number of missing values after the replacement is: ", check)

print_replace
```

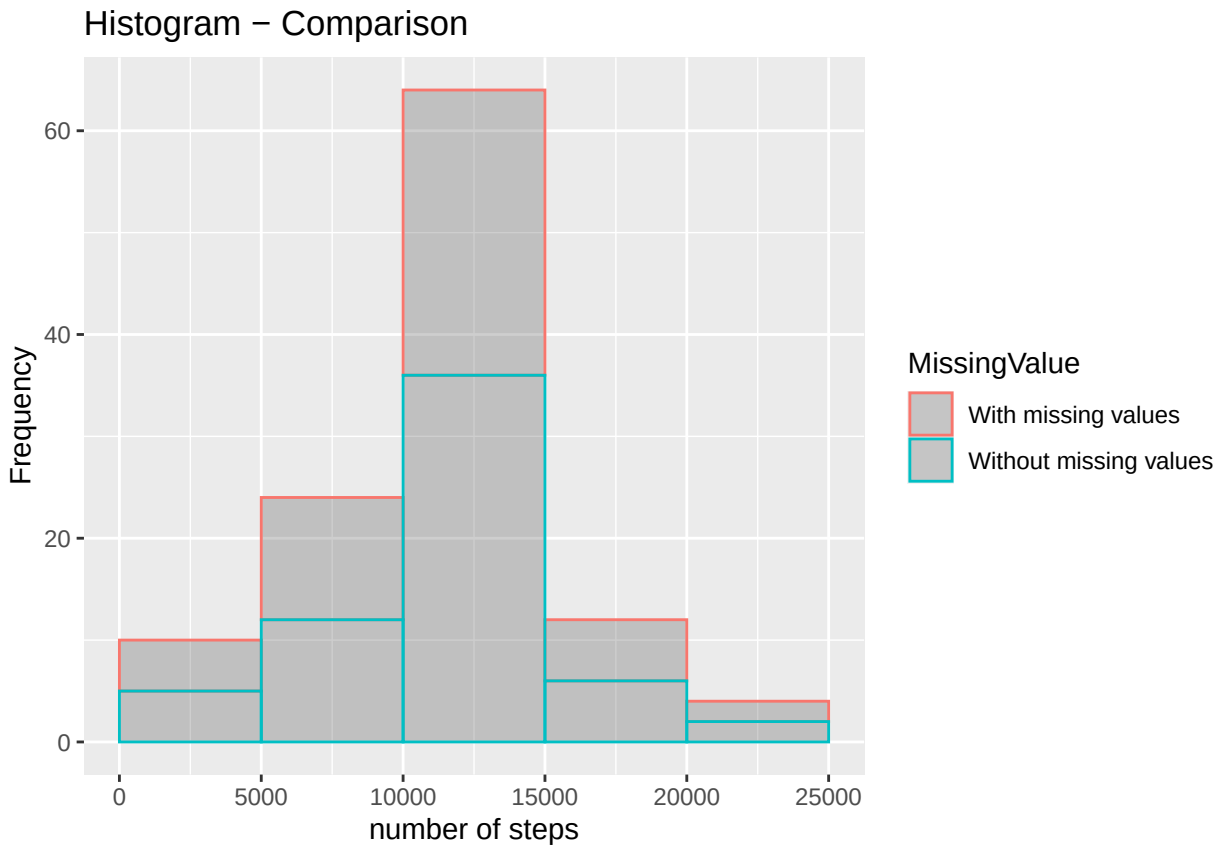
```
## [1] "The total number of missing values after the replacement is: 0"
```

Histogram Plotting 2

7. Histogram of the total number of steps taken each day after missing values are imputed

Make a histogram of the total number of steps taken each day and Calculate and report the mean and median total number of steps taken per day. Do these values differ from the estimates from the first part of the assignment? What is the impact of imputing missing data on the estimates of the total daily number of steps?

```
steps_histogram_replaced <- activity_replaced %>%
  filter(!is.na(steps)) %>%
  group_by(date) %>%
  summarize(total_steps = sum(steps)) %>%
  mutate(MissingValue = "Without missing values")

steps_histogram <- steps_histogram %>%
  mutate(MissingValue = "With missing values")

steps_hist_total <- bind_rows(steps_histogram, steps_histogram_replaced)

ggplot(steps_hist_total, aes(total_steps, color = MissingValue)) +
  geom_histogram(breaks = seq(0,25000, by = 5000), alpha = 0.3) +
  labs(title = "Histogram - Comparison", x = "number of steps", y = "Frequency")
```




```
steps_mean <- mean(steps_histogram_replaced$total_steps)
print_mean <- paste0("The average number of steps after replacing missing values is: ",steps_mean)

print_mean
```

```
## [1] "The average number of steps after replacing missing values is: 10766.1886792453"
```

```
steps_median <- median(steps_histogram_replaced$total_steps)
print_median <- paste0("The median of steps after replacing missing values is: ", steps_median)

print_median
```

```
## [1] "The median of steps after replacing missing values is: 10766.1886792453"
```

##Series Plotting 2

8. Panel plot comparing the average number of steps taken per 5-minute interval across weekdays and weekends

```
steps_series_week <- activity %>%
  filter(!is.na(steps)) %>%
  mutate(week_number = wday(ymd(date),week_start = 1),
         week_part = if_else(week_number <= 5, "Weekday","Weekend")) %>%
  group_by(interval, week_part) %>%
  summarize(avg_steps = mean(steps))


ggplot(steps_series_week, aes(x = interval,y = avg_steps, color = week_part)) +
  geom_line() +
```

```
  geom_point() +
  geom_smooth() +
  labs(title = "Average steps by interval", x = "Interval", y = "average of steps by day")
```


```
setwd(original_wd)
```