Reproducible Research: Peer Assessment 1

Juan C. López Tavera

Introduction

This is the peer-graded assignment for the Reproducible Research Course by Johns Hopkins University at Coursera, which is the 5th out of 10 courses in the Data Science Specialization.

The objective of this assignment is to make a reproducible report of an individual's activity over a two month period, measured in number of steps. The report will be generated using knitr.

As provided in Professor Peng's original repository, this repository is self-contained — all data and assignment instructions necessary to reproduce this work are available in the same place.

Setting up the report

First, we need to setup the report options using, and install —if necessary— and load all the required packages to successfully complete this assignment.

```
## Loading the necessary package for reproducing the assingment
if (!require(knitr)) {
    install.packages("knitr")
}
library(knitr)

if (!require(tidyverse)) {
    install.packages("tidyverse")
}
library(tidyverse)

## Setting up all code chunks according to the assignment specs
knitr::opts_chunk$set(
    eval = TRUE,
    echo = TRUE,
    tidy = TRUE,
    results = "markup",
    include = TRUE,
    message = FALSE,
    warning = FALSE,
    knitr.table.format = "markdown",
    tidy.opts = list(width.cutoff = 80),
    fig.align = "center",
    fig.path = "figure/",
    highlight = TRUE
)
```

Loading and preprocessing the data

```
Instructions:
Show any code that is needed to
1. Load the data (i.e. read.csv())
```

```
2. Process/transform the data (if necessary) into a format suitable for your analysis
```

In the original repository, the data set is compressed in a ZIP file, which we unzip —if necessary— and load the resulting CSV file into the working environment.

No pre-processing was necessary for this data set, as it is already tidy.

```
## If necessary, unzipping the data file, and loading it
if (!file.exists("activity.csv") & file.exists("activity.zip")) {
    unzip("activity.zip")
    activity.data <- read_csv(file = "activity.csv")
} else if (file.exists("activity.csv")) {
    activity.data <- read_csv(file = "activity.csv")
} else {
    message("Activity Monitoring Data (default) from Rep Research course was not found")
}
```

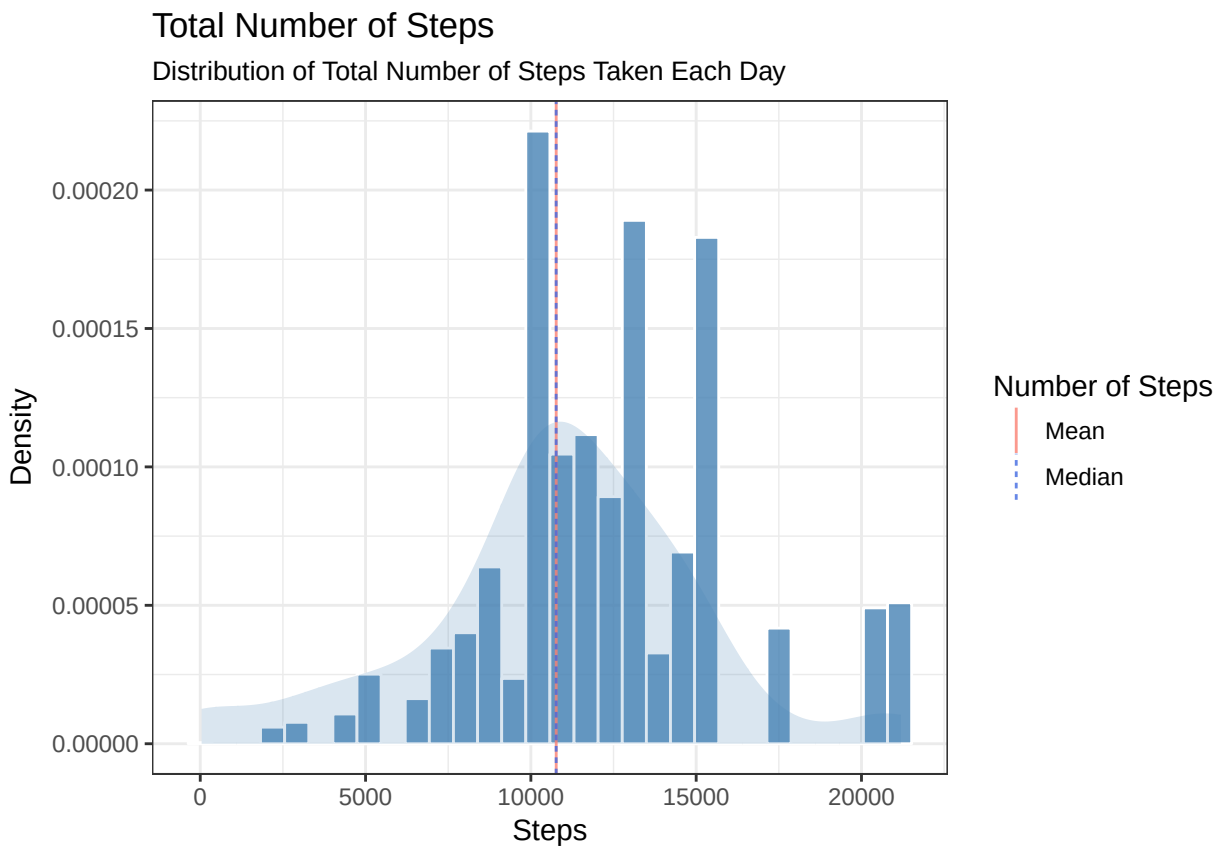
What is mean total number of steps taken per day?

```
Instrunctions:
For this part of the assignment, you can ignore the missing values in the dataset.
1. Make a histogram of the total number of steps taken each day.
2. Calculate and report the **mean** and **median** total number of steps taken per day.
```

We want to know the distribution of the number of steps taken each day, which we can observe in a histogram generated using ggplot2.

In the figure below, we can observe the histogram of the number of steps taken each day, with an overlayed density plot.

```
steps.day.df <- activity.data %>% group_by(date) %>% summarise(steps = sum(steps))

mean.median.steps <- steps.day.df %>% summarise(Mean = round(mean(steps, na.rm = TRUE)),
    Median = round(median(steps, na.rm = TRUE))) %>% gather()

ggplot(data = steps.day.df, mapping = aes(x = steps, y = ..density..)) + geom_histogram(aes(weight = st
    fill = "steelblue", colour = "white", alpha = 0.8) + geom_density(fill = "steelblue",
    colour = NA, alpha = 0.2) + ggtitle(list(title = "Total Number of Steps", subtitle = "Distribution
    x = "Steps", y = "Density")) + geom_vline(data = mean.median.steps, aes(xintercept = value,
    linetype = key), size = 0.5, alpha = 0.8, colour = c("salmon", "royalblue")) +
    scale_linetype_discrete(name = "Number of Steps") + theme_bw()
```



We want to know basic statistics about the number of steps, which are also depicted in the following table:

```
steps.day.df %>% summarise(`Mean number of steps taken per day` = round(mean(steps,
    na.rm = TRUE)), `Median number of steps taken per day` = round(median(steps,
    na.rm = TRUE))) %>% kable(align = "c")
```

Mean number of steps taken per day	Median number of steps taken per day
10766	10765

What is the average daily activity pattern?

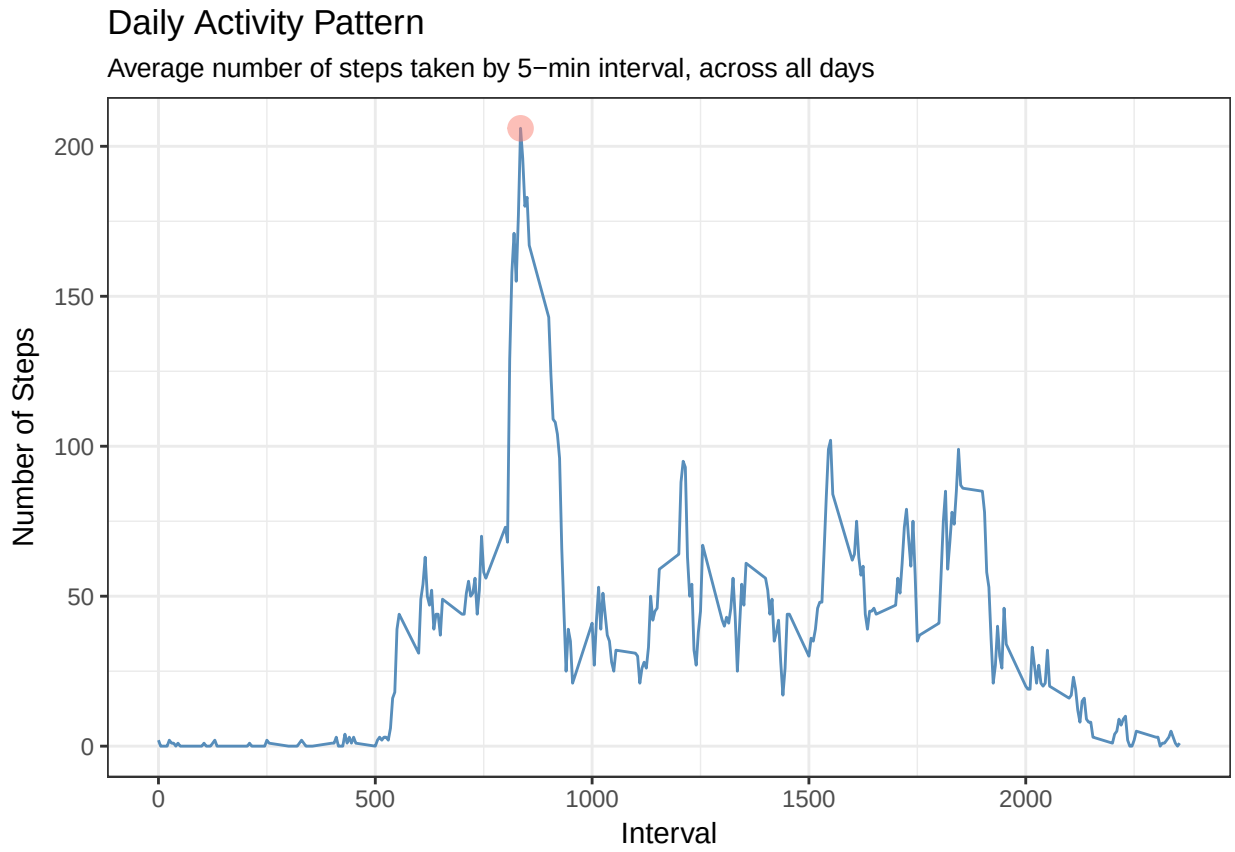
```
Instructions:
1. Make a time series plot (i.e. `type = "l"`) of the 5-minute interval (x-axis) and the average number
2. Which 5-minute interval, on average across all the days in the dataset, contains the maximum number
```

We also want to know the daily activity pattern, which is how many steps —averaged across all days— are taken during the day.

For this purpose, we create a line plot, using ggplot2, shown below.

```
avg.steps.day <- activity.data %>% group_by(interval) %>% summarise(steps = round(mean(steps,
    na.rm = TRUE))) %>% filter(!is.nan(steps))

max.steps <- avg.steps.day[which.max(avg.steps.day$steps), ]

ggplot(data = avg.steps.day, mapping = aes(x = interval, y = steps)) + geom_line(size = 0.5,
    colour = "steelblue", alpha = 0.9) + ggtitle(list(title = "Daily Activity Pattern",
```

```
    subtitle = "Average number of steps taken by 5-min interval, across all days",
    y = "Number of Steps", x = "Interval")) + geom_point(data = max.steps, mapping = aes(x = interval,
    y = steps), size = 4, alpha = 0.5, colour = "salmon") + theme_bw()
```



In the figure, we can observe a peak (maximum value) in the number of steps taken, 206, in the interval 835.

Imputing missing values

```
Instructions:
Note that there are a number of days/intervals where there are missing values (coded as `NA`). The prese
1. Calculate and report the total number of missing values in the dataset (i.e. the total number of rows
2. Devise a strategy for filling in all of the missing values in the dataset. The strategy does not nee
3. Create a new dataset that is equal to the original dataset but with the missing data filled in.
```

```
nas.df <- activity.data %>% sapply(is.na) %>% tbl_df() %>% summarise(Steps = sum(steps),
    Date = sum(date), Interval = sum(interval))

row.names(nas.df) <- c("Number of NAs")

nas.df %>% kable(caption = "Number of Missing Values of each Variable", align = "c")
```

Table 2: Number of Missing Values of each Variable

	Steps	Date	Interval
Number of NAs	2304	0	0

First, we want to know how many missing our data set has; in this case, there are 2304 missing values, in total. The detail of the missing values for each variable is shown in the table above

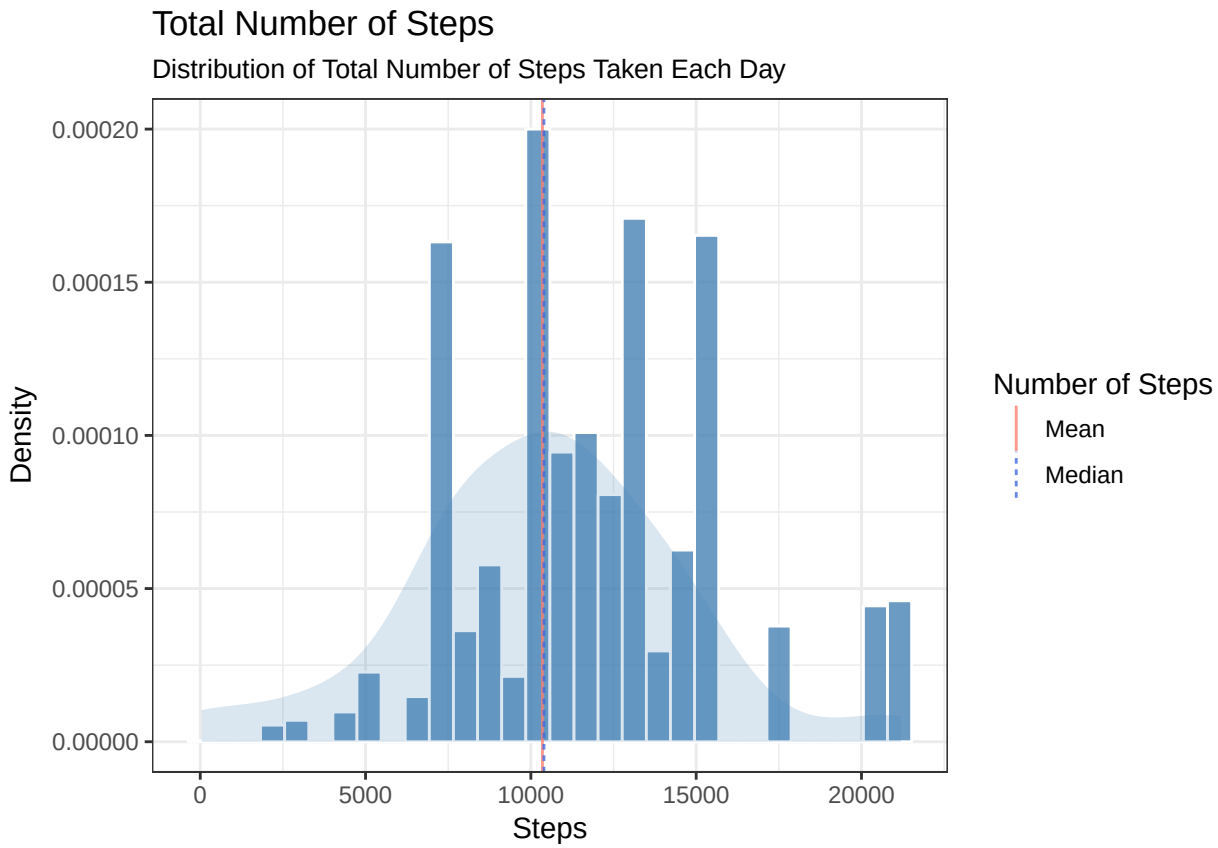
In the assignment instructions, we are required to devise a simple strategy to impute missing values. The strategy that we'll follow is, indeed, simple:

a. Get the trimmed mean of the number of steps (exclude the 10% most extreme observations) by 5-min interval.
b. Joining the original data set with NAs and the data set of mean steps by 5-min interval.
c. Substitute NAs with the corresponding mean value.
d. Cleaning the data frame.

```
non.NA <- activity.data %>% group_by(interval) %>% summarise(steps = round(mean(steps,
    na.rm = TRUE, trim = 0.05)))


imp.activity.data <- full_join(x = activity.data, y = non.NA, by = "interval") %>%
    mutate(steps = ifelse(test = is.na(steps.x), yes = steps.y, no = steps.x)) %>%
    select(steps, date, interval)
```

The histogram of the total number of steps taken per day doesn't change much.

```
steps.day.df <- imp.activity.data %>% group_by(date) %>% summarise(steps = sum(steps))

mean.median.steps <- steps.day.df %>% summarise(Mean = round(mean(steps, na.rm = TRUE)),
    Median = round(median(steps, na.rm = TRUE))) %>% gather()

ggplot(data = steps.day.df, mapping = aes(x = steps, y = ..density..)) + geom_histogram(aes(weight = st
    fill = "steelblue", colour = "white", alpha = 0.8) + geom_density(fill = "steelblue",
    colour = NA, alpha = 0.2) + ggtitle(list(title = "Total Number of Steps", subtitle = "Distribution
    x = "Steps", y = "Density")) + geom_vline(data = mean.median.steps, aes(xintercept = value,
    linetype = key), size = 0.5, alpha = 0.8, colour = c("salmon", "royalblue")) +
    scale_linetype_discrete(name = "Number of Steps") + theme_bw()
```



With the chosen imputation strategy, there were no changes in the median and mean number of the steps taken per day.

```
steps.day.df %>% summarise(`Mean number of steps taken per day` = round(mean(steps,
    na.rm = TRUE)), `Median number of steps taken per day` = round(median(steps,
    na.rm = TRUE))) %>% kable(align = "c")
```

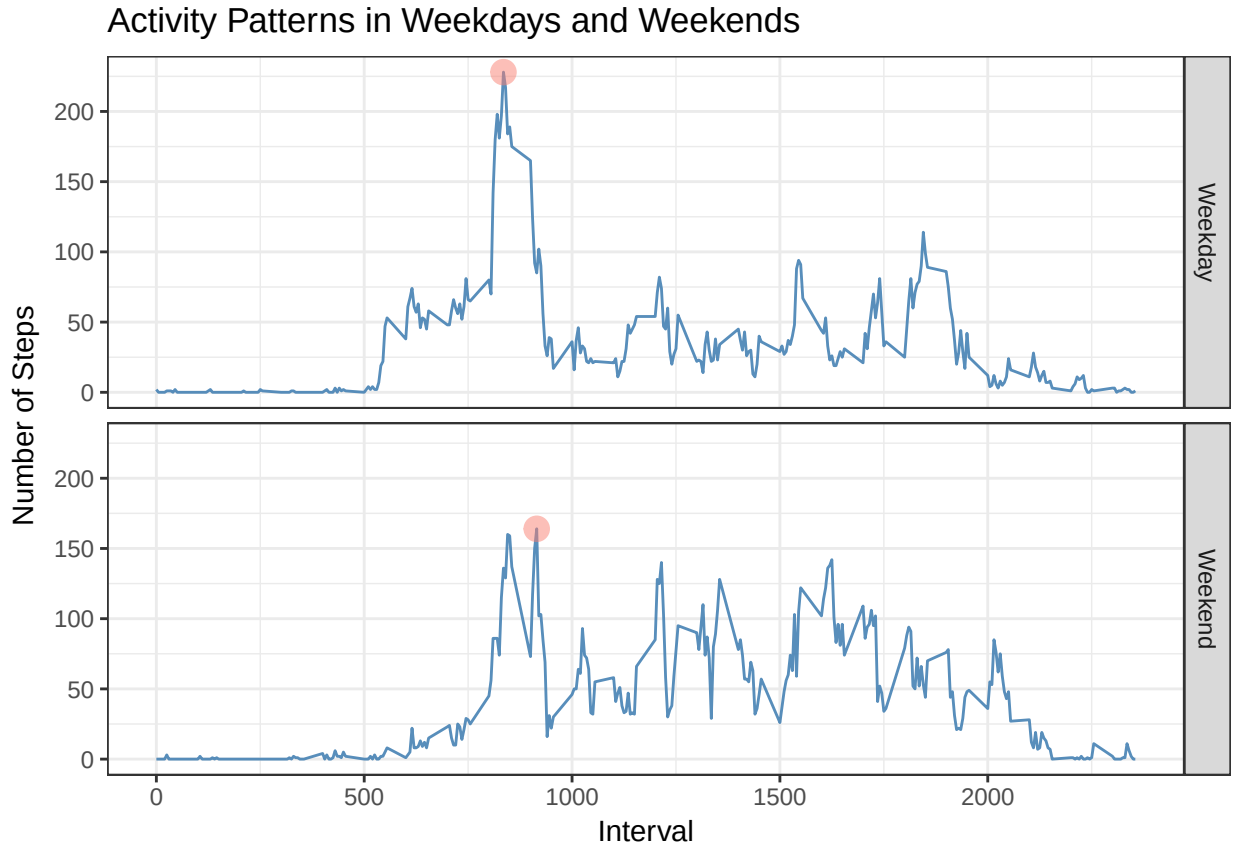
Mean number of steps taken per day	Median number of steps taken per day
10350	10395

Are there differences in activity patterns between weekdays and weekends?

```
Instructions:
For this part the `weekdays()` function may be of some help here. Use the dataset with the filled-in mis
1. Create a new factor variable in the dataset with two levels -- "weekday" and "weekend" indicating wh
2. Make a panel plot containing a time series plot (i.e. type = "l") of the 5-minute interval (x-axis) a
```

```
avg.steps.weekday <- imp.activity.data %>% mutate(weekday.end = ifelse(test = weekdays(date,
    abbreviate = TRUE) %in% c("Sat", "Sun"), yes = "Weekend", no = "Weekday")) %>%
    group_by(interval, weekday.end) %>% summarise(steps = round(mean(steps, na.rm = TRUE)))

maxima.steps <- avg.steps.weekday %>% group_by(weekday.end) %>% top_n(1, steps)

ggplot(data = avg.steps.weekday, mapping = aes(x = interval, y = steps)) + geom_line(colour = "steelblu
    alpha = 0.9) + facet_grid(weekday.end ~ .) + geom_point(data = maxima.steps,
    mapping = aes(x = interval, y = steps, group = weekday.end), alpha = 0.5, size = 4,
```

```
    colour = "salmon") + ggtitle(list(title = "Activity Patterns in Weekdays and Weekends",
    y = "Number of Steps", x = "Interval")) +
theme_bw()
```



The maxima number of steps during weekend and weekdays are notoriously different, as can be seen in the figure above. The peak of activity during the weekdays is 228 steps, which happens in the 835 interval; while the peak of activity during the weekends is 164 steps, which happens in the 915 interval.