Reproducible Research: Peer Assessment 1

Simon Coburg

26/02/2021

This is my submission for the Reproducible Research Course Project 1. To obtain more information about this project see the ReadMe on GitHub.

About

This assignment makes use of data from a personal activity monitoring device. The device collected data at 5 minute intervals through out the day. The data consists of two months of data from an anonymous individual collected during the months of October and November, 2012, and include the number of steps taken in 5 minute intervals each day.

The data for this assignment can be downloaded from the course web site:

- **Dataset**: Activity monitoring data [52K]

The variables included in this dataset are:

- **steps**: Number of steps taking in a 5-minute interval (missing values are coded as NA).
- **date**: The date on which the measurement was taken in YYYY-MM-DD format.
- **interval**: Identifier for the 5-minute interval in which measurement was taken.

The dataset is stored in a *comma-separated-value (CSV) file* and there are a total of *17,568 observations* in this dataset.

Loading and preprocessing the data

The following steps were applied to load, import and check the dataset.

```
#Libraries
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
#Load data
path <- getwd()
download.file(url = "https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip"
              , destfile = paste(path, "dataFiles.zip", sep = "/"))
unzip(zipfile = "dataFiles.zip")
```

```r
#Check if data present
dir()
```

```
##  [1] "activity.csv"
##  [2] "Courseproject1.R"
##  [3] "dataFiles.zip"
##  [4] "doc"
##  [5] "instructions_fig"
##  [6] "PA1_template.html"
##  [7] "PA1_template.Rmd"
##  [8] "Plot1_Total number of steps taken each day.png"
##  [9] "Plot2_Average daily activity pattern.png"
## [10] "Plot3_Adjusted total number of steps taken each day.png"
## [11] "Plot4_Weekday vs weekend activity pattern.png"
## [12] "README.md"
## [13] "RepData_PeerAssessment1.Rproj"
```

```r
#Remove zip file
file.remove("dataFiles.zip")
```

```
## [1] TRUE
```

```r
#Import dataset
activity <- read.csv("activity.csv")

#Check dataset
head(activity)
```

```
##   steps       date interval
## 1    NA 2012-10-01        0
## 2    NA 2012-10-01        5
## 3    NA 2012-10-01       10
## 4    NA 2012-10-01       15
## 5    NA 2012-10-01       20
## 6    NA 2012-10-01       25
```

```r
str(activity)
```

```
## 'data.frame':    17568 obs. of  3 variables:
##  $ steps   : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ date    : chr  "2012-10-01" "2012-10-01" "2012-10-01" "2012-10-01" ...
##  $ interval: int  0 5 10 15 20 25 30 35 40 45 ...
```

What is mean total number of steps taken per day?

For this part of the assignment, missing values in the dataset could be ignored.

1. Calculate the total number of steps taken per day.
2. Make a histogram of the total number of steps taken each day.
3. Calculate and report the mean and median of the total number of steps taken per day.

```r
steps_total <- activity %>%
  group_by(date) %>%
  summarise(steps_daily = sum(steps, na.rm = TRUE))
```

1. Number of steps taken per day

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
head(steps_total)
```

```
## # A tibble: 6 x 2
##   date       steps_daily
##   <chr>            <int>
## 1 2012-10-01           0
## 2 2012-10-02         126
## 3 2012-10-03       11352
## 4 2012-10-04       12116
## 5 2012-10-05       13294
## 6 2012-10-06       15420
```

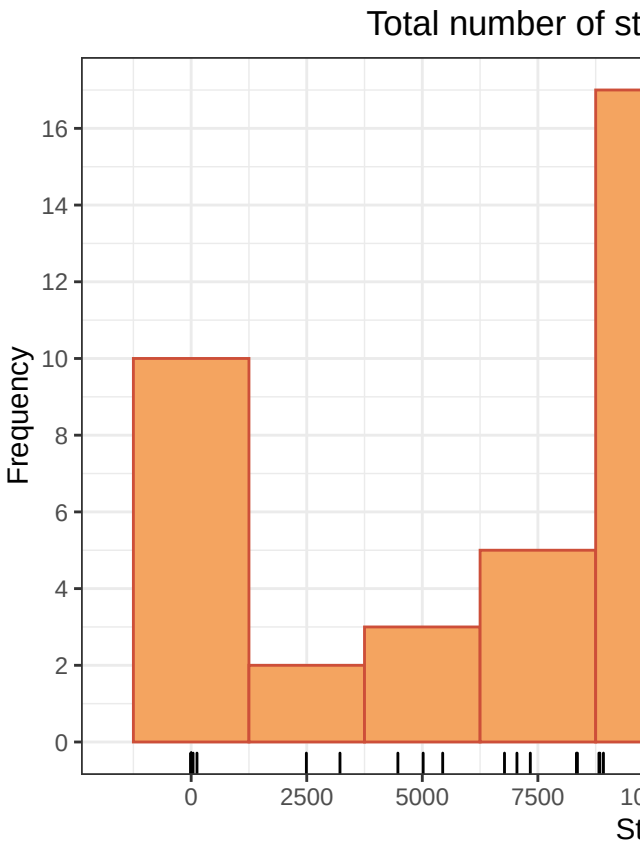
```
#Calculate and report sum
sum(steps_total$steps_daily, na.rm = TRUE)
```

```
## [1] 570608
```

Total number of steps taken are *570,608*.

```
#Create barplot
p1 <- ggplot(steps_total, aes(steps_daily)) +
      geom_histogram(binwidth = 2500, col = "tomato3", fill = "sandybrown") +
      geom_rug(aes(steps_daily)) +
      ggtitle("Total number of steps taken each day") +
      xlab("Steps") +
      ylab("Frequency") +
      scale_y_continuous(breaks=seq(0,18,2)) +
      scale_x_continuous(breaks=seq(0,25000,2500)) +
      theme_bw() +
      theme(plot.title = element_text(hjust = 0.5))

p1
```



2. Create histogram of total number of steps taken each day

```
#Safe file in working directory
ggsave(filename = "Plot1_Total number of steps taken each day.png", p1, dpi = 600, limitsize = TRUE)
```

```
## Saving 6.5 x 4.5 in image
```

```
summary(steps_total$steps_daily, na.rm = TRUE)
```

3. Calculate mean and median of total number of steps taken per day

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0    6778   10395    9354   12811   21194
```

What is the average daily activity pattern?

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

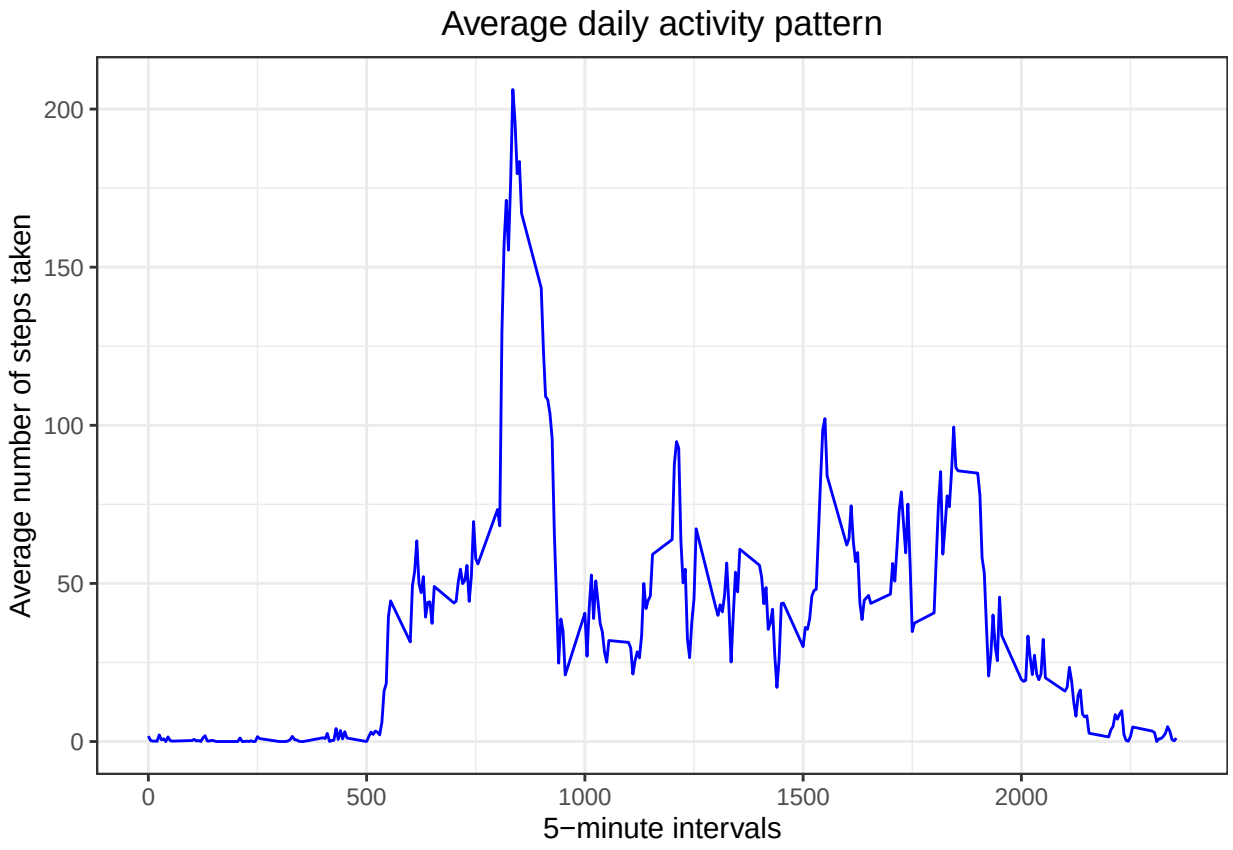
```
#Create line plot for average steps per 5-minute intervals
steps_interval <- activity %>%
  group_by(interval) %>%
  summarise(steps = mean(steps, na.rm =TRUE))
```

1. Time series of the averaged number of steps taken by 5-minute intervals

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
p2 <- ggplot(steps_interval, aes(interval, steps)) +
      geom_line(col="blue") +
      ggtitle("Average daily activity pattern") +
      xlab("5-minute intervals") +
      ylab("Average number of steps taken") +
      theme_bw() +
      theme(plot.title = element_text(hjust = 0.5))

p2
```




```
#Safe file in working directory
ggsave(filename = "Plot2_Average daily activity pattern.png", p2, dpi = 600, limitsize = TRUE)
```

```
## Saving 6.5 x 4.5 in image
```

```
#Report max 5-minute interval
which.max(steps_interval$steps)
```

2. Which 5-minute interval (averaged across all days) contains max. number of steps?

```
## [1] 104
```

```
max_interval = round(steps_interval[104,])
max_interval
```

```
## # A tibble: 1 x 2
##   interval steps
```

```
##      <dbl> <dbl>
## 1      835   206
```

The 5-minute interval that contains the maximum number of steps is the *835th interval*.

Imputing missing values

There are a number of days/intervals where there are missing values (coded as NANA). The presence of missing days may introduce bias into some calculations or summaries of the data.

1. Calculate and report the total number of missing values (NA) in the dataset.
2. Replace missing values (NA) in the dataset.
3. Create a new dataset that is equal to the original dataset but with the missing data filled in.
4. Make a histogram of the total number of steps taken each day.
5. 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?

```
#Creating table
missing <- tbl_df(activity)
```

1. Calculate total number of NA in the dataset

```
## Warning: `tbl_df()` is deprecated as of dplyr 1.0.0.
## Please use `tibble::as_tibble()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

```
#Filtering for NA
missing %>% filter(is.na(steps)) %>% summarize(missing_values = n())
```

```
## # A tibble: 1 x 1
##   missing_values
##            <int>
## 1           2304
```

```
activity$steps_complete <- ifelse(is.na(activity$steps),
  round(steps_interval$steps[match(activity$interval, steps_interval$interval)],0),
  activity$steps)
```

2. Replacing NA values by the averaged 5-minute intervals

```
activity_complete <- data.frame(steps=activity$steps_complete,
  interval=activity$interval, date=activity$date)

head(activity_complete)
```

3. Create new dataset adjusted for NA values

```
##   steps interval       date
## 1     2        0 2012-10-01
## 2     0        5 2012-10-01
## 3     0       10 2012-10-01
## 4     0       15 2012-10-01
```

```
## 5     0       20 2012-10-01
## 6     2       25 2012-10-01
```

```
#Check if NA values still present
any(is.na(activity_complete))
```

```
## [1] FALSE
```

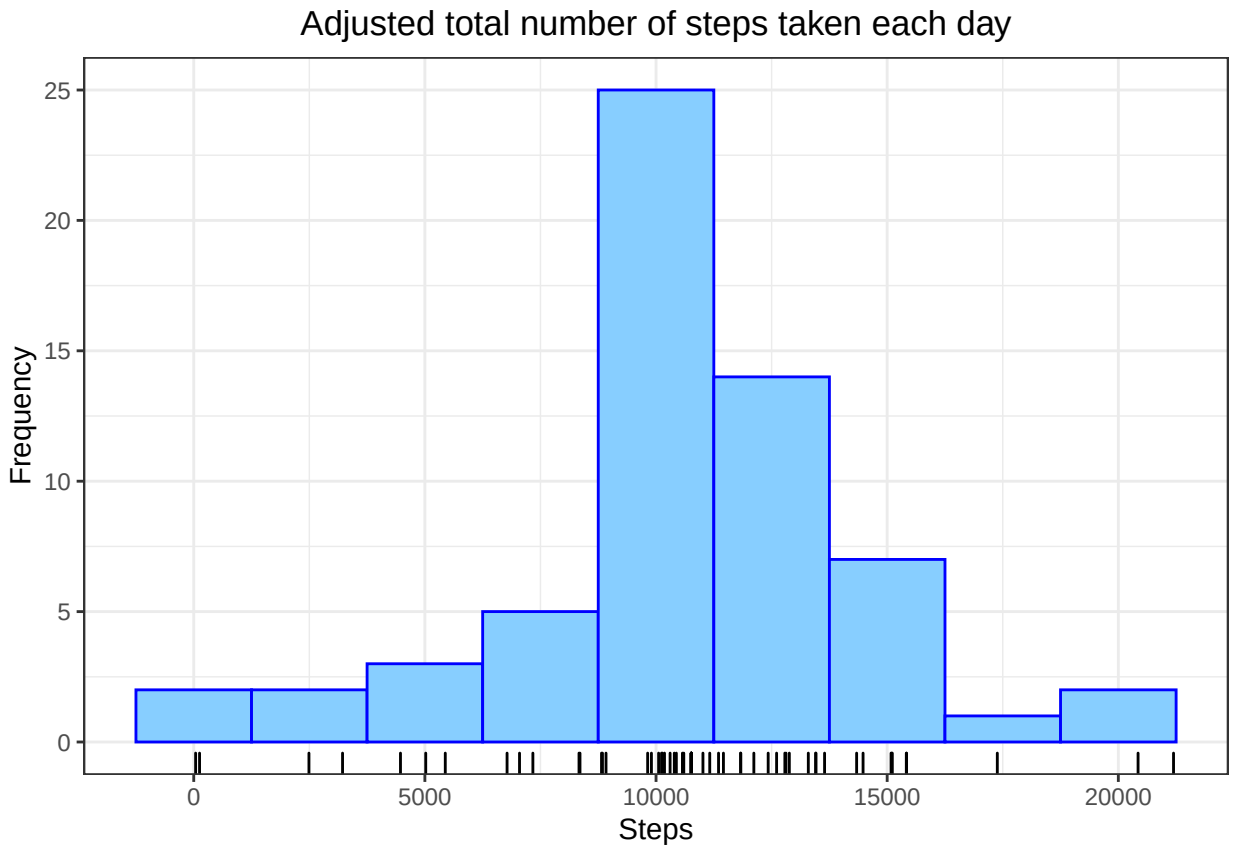
```
activity_complete_steps <- aggregate(activity_complete$steps, list(activity_complete$date), FUN=sum)
colnames(activity_complete_steps) <- c("Date", "Steps")
head(activity_complete_steps)
```

4. Create histogram with new dataset

```
##         Date Steps
## 1 2012-10-01 10762
## 2 2012-10-02   126
## 3 2012-10-03 11352
## 4 2012-10-04 12116
## 5 2012-10-05 13294
## 6 2012-10-06 15420
```

```
p3 <- ggplot(activity_complete_steps, aes(Steps)) +
      geom_histogram(binwidth = 2500, col = "blue", fill = "skyblue1") +
      geom_rug(aes(Steps)) +
      ggtitle("Adjusted total number of steps taken each day") +
      xlab("Steps") +
      ylab("Frequency") +
      theme_bw() +
      theme(plot.title = element_text(hjust = 0.5))

p3
```




```
#Safe file in working directory
ggsave(filename = "Plot3_Adjusted total number of steps taken each day.png", p3, dpi = 600, limitsize =
```

```
## Saving 6.5 x 4.5 in image
```

```
sum(activity_complete_steps$Steps, na.rm = TRUE)
```

5. Calculate and report sum, mean & median

```
## [1] 656704
```

```
summary(activity_complete_steps$Steps, na.rm = TRUE)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      41    9819   10762   10766   12811   21194
```

```
#Do sum/mean/median values differ from Q1
sum(activity_complete_steps$Steps, na.rm = TRUE) - sum(steps_total$steps_daily, na.rm = TRUE)
```

```
## [1] 86096
```

```
summary(activity_complete_steps$Steps, na.rm = TRUE) - summary(steps_total$steps_daily, na.rm = TRUE)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      41    3041     367    1411       0       0
```

The estimate of total number of steps increases by *86,096*. Mean and median differ by the amount of *1,411* and *367* respectively compared to the previous estimates that excluded the missing values.

Are there differences in activity patterns between weekdays and weekends?

1. Create a new factor variable in the dataset with two levels – "weekday" and "weekend" indicating whether a given date is a weekday or weekend day.
2. Make a panel plot containing a time series plot (i.e. `type = "l"`type = "l") of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all weekday days or weekend days (y-axis).

```
#Change date format (if necessary)
activity_complete$date <- as.Date(activity_complete$date, format = "%Y-%m-%d")

#Add variable with the according weekdays name
activity_complete$Weekday <- weekdays(activity_complete$date)

#Distinguish between weekdays and weekend
activity_complete$Type <- ifelse(activity_complete$Weekday=='Saturday' |
  activity_complete$Weekday=='Sunday', 'Weekend','Weekday')

#Check dataset
head(activity_complete)
```

1. Create a new factor variable with two levels: "weekday" and "weekend"

```
##   steps interval       date Weekday    Type
## 1     2        0 2012-10-01  Monday Weekday
## 2     0        5 2012-10-01  Monday Weekday
## 3     0       10 2012-10-01  Monday Weekday
## 4     0       15 2012-10-01  Monday Weekday
## 5     0       20 2012-10-01  Monday Weekday
## 6     2       25 2012-10-01  Monday Weekday
```

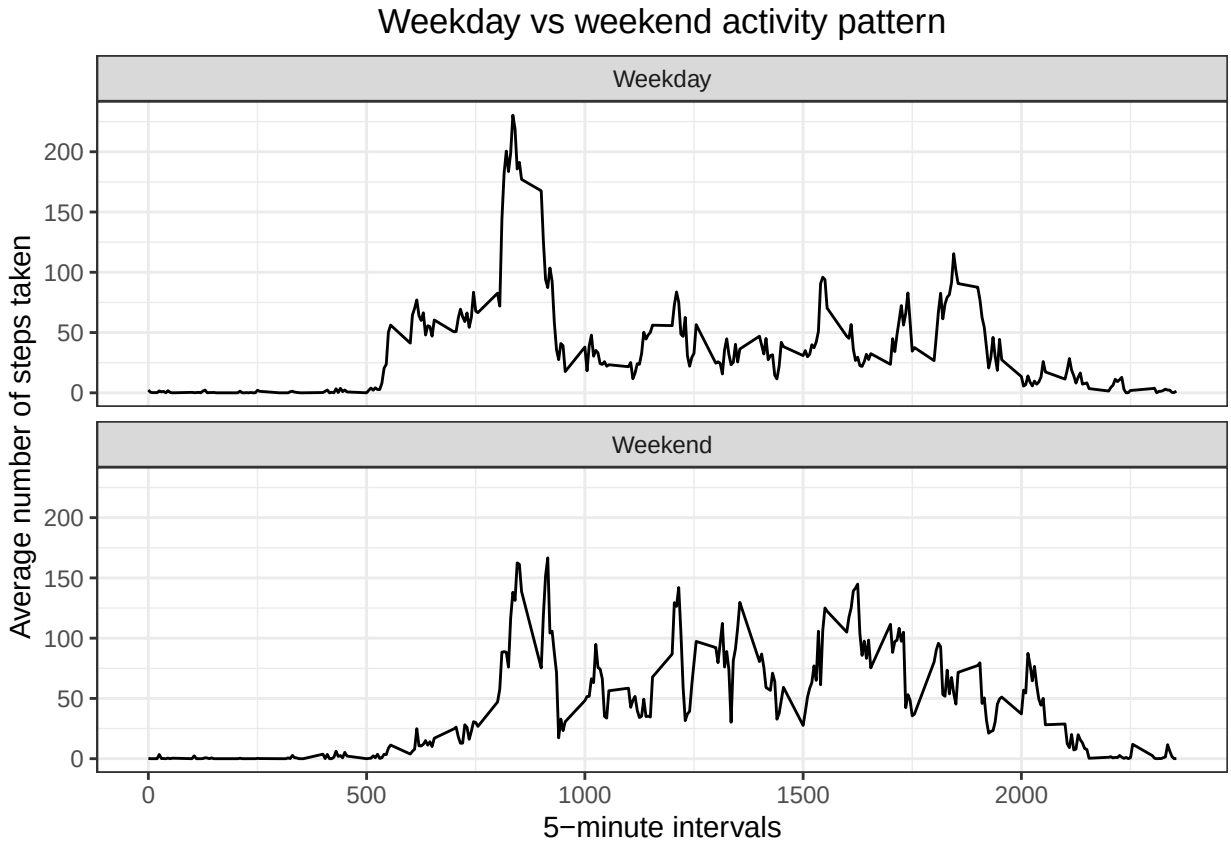
```
steps_week <- activity_complete %>%
  group_by(interval, Type) %>%
  summarise(steps = mean(steps, na.rm =TRUE))
```

2. Plot two time series for weekdays & weekend

```
## `summarise()` regrouping output by 'interval' (override with `.groups` argument)
```

```
p4 <- ggplot(steps_week, aes(interval, steps)) +
      geom_line(col="black") +
      facet_wrap(~Type, nrow=2) +
      ggtitle("Weekday vs weekend activity pattern") +
      xlab("5-minute intervals") +
      ylab("Average number of steps taken") +
      theme_bw() +
      theme(plot.title = element_text(hjust = 0.5))

p4
```




```
#Safe file in working directory
ggsave(filename = "Plot4_Weekday vs weekend activity pattern.png", p4, dpi = 600, limitsize = TRUE)
```

```
## Saving 6.5 x 4.5 in image
```

This concludes the assignment. *This document, the R-script and graphs can be found under my GitHub domain.*