Reproducible Research Assessment 1

Colette Apelian

1/28/2022

## Introduction

It is now possible to collect a large amount of data about personal movement using activity monitoring devices such as a Fitbit, Nike Fuelband, or Jawbone Up. These type of devices are part of the “quantified self” movement – a group of enthusiasts who take measurements about themselves regularly to improve their health, to find patterns in their behavior, or because they are tech geeks. But these data remain under-used because the raw data is difficult to obtain and there is a lack of statistical methods and software for processing and interpreting it.

This assignment makes use of data from a personal activity monitoring device. The device collects 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 was downloaded from this web site <https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip> and the Dataset is called “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

Download the .csv file Activity monitoring data to my desktop.

Use Excel to format the date column into dates with the month, day, and year

I load the data into R

data <- read.csv("C:\\Users\\CA\\Desktop\\activity.csv", header = TRUE, sep = ",")

## What is mean total number of steps taken per day (ignoring the missing values in the dataset)?

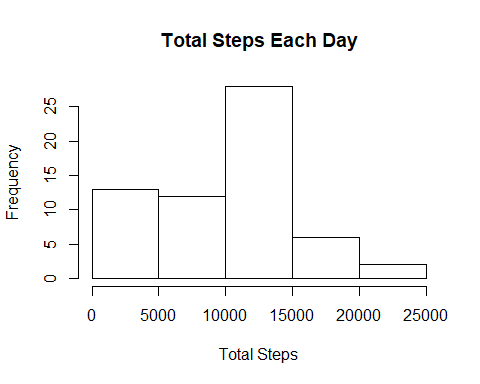
Calculate the total number of steps taken per day

Aggregate steps by date and convert from array to a dataframe

totSteps <- tapply(data$steps, data$date, FUN = sum, na.rm = TRUE)  
  
totSteps <- as.data.frame(totSteps)

Make a histogram of the total number of steps taken each day

hist(totSteps$totSteps, main = "Total Steps Each Day", xlab = "Total Steps")



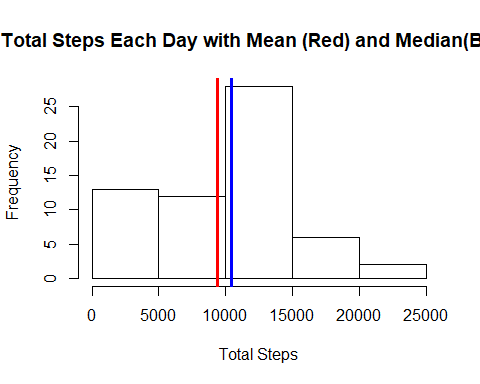
Calculate and report the mean and median of the total number of steps taken per day

First, calculate the mean and median of the total number of steps taken per day

x <- mean(totSteps$totSteps)  
  
y <- median(totSteps$totSteps)

Add Mean & Median lines to the histogram

hist(totSteps$totSteps, main = "Total Steps Each Day with Mean (Red) and Median(Blue)", xlab = "Total Steps")  
  
abline(v = mean(totSteps$totSteps),col = "red", lwd = 3)  
  
abline(v = median(totSteps$totSteps), col = "blue", lwd = 3)



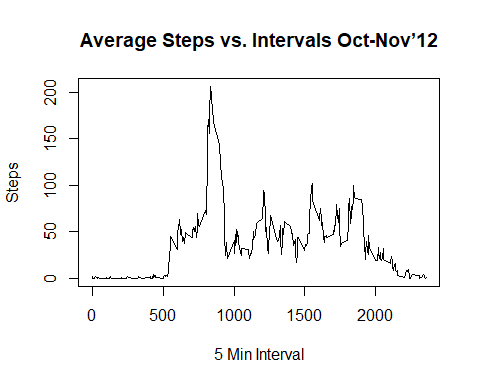
## What is the average daily activity pattern?

Create a dataframe with the average daily activity pattern

data2 <- aggregate(x = list(steps =data$steps), by = list(interval = data$interval), FUN = mean, na.rm = TRUE)

Make a time series plot of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all days (y-axis)

plot(data2$interval, data2$steps, type = "l", xlab = "5 Min Interval", ylab = "Steps", main = "Average Steps vs. Intervals Oct-Nov’12")



## Which 5-minute interval, on average across all the days in the dataset, contains the maximum number of steps?

Find the maximum number of steps

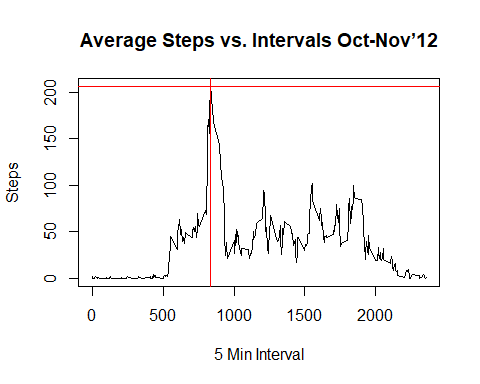
summary(data2)

## interval steps   
## Min. : 0.0 Min. : 0.000   
## 1st Qu.: 588.8 1st Qu.: 2.486   
## Median :1177.5 Median : 34.113   
## Mean :1177.5 Mean : 37.383   
## 3rd Qu.:1766.2 3rd Qu.: 52.835   
## Max. :2355.0 Max. :206.170

View the dataset by clicking on the table icon in the Environment pane to manually research the corresponding interval to the maximum number of steps by inputting 206 into the find window and finding interval 835.

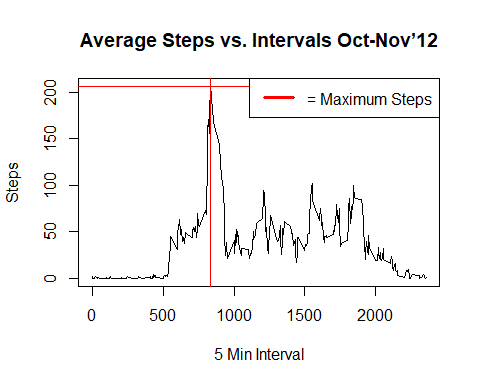
Plot a horizontal and vertical line indicating the maximum number of steps

plot(data2$interval, data2$steps, type = "l", xlab = "5 Min Interval", ylab = "Steps", main = "Average Steps vs. Intervals Oct-Nov’12")  
  
abline(h = 206.170, col = "red")  
  
abline(v = 835, col = "red")



Add a legend explaining the lines

plot(data2$interval, data2$steps, type = "l", xlab = "5 Min Interval", ylab = "Steps", main = "Average Steps vs. Intervals Oct-Nov’12")  
  
abline(h = 206.170, col = "red")  
  
abline(v = 835, col = "red")  
  
legend("topright", legend = "= Maximum Steps", lwd = 3, col = "red")



## Impute missing values (days/intervals where there are missing values coded with NA)

Why impute the missing values with a mean, in my choice? Because the presence of missing days may introduce bias into some calculations or summaries of the data.

Calculate and report the total number of missing values in the dataset (i.e. the total number of rows with NA)

colSums(is.na(data))

## steps date interval   
## 2304 0 0

Create a new dataset based on daily mean

data3 <- aggregate(x = list(steps =data$steps), by = list(date = data$date), FUN = mean, na.rm = FALSE)

Calculate a mean for the steps column ignoring the NA

x <- mean(data3$steps, na.rm = TRUE)  
  
print(x)

## [1] 37.3826

Replace the NA in the steps column with the mean (37.3826)

data3$steps[which(is.na(data3$steps))] <- mean(data3$steps, na.rm = TRUE)

Confirm new dataset has no NA values

colSums(is.na(data3))

## date steps   
## 0 0

Make a histogram of the total number of steps taken each day.

Create a new dataset from the original

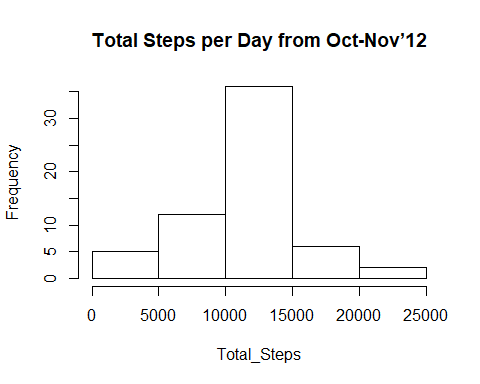
data4 <- data

Change NA to mean

data4$steps[which(is.na(data4$steps))] <- mean(data3$steps, na.rm = TRUE)

Create another dataset of sum values

data5 <- aggregate(x = list(steps =data4$steps), by = list(date = data4$date), FUN = sum, na.rm = FALSE)  
  
hist(data5$steps, xlab = "Total\_Steps", ylab = "Frequency", main = "Total Steps per Day from Oct-Nov’12")



Calculate and report the mean and median total number of steps taken per day with purple (red+blue=purple) indicating them on the histogram, and add a legend

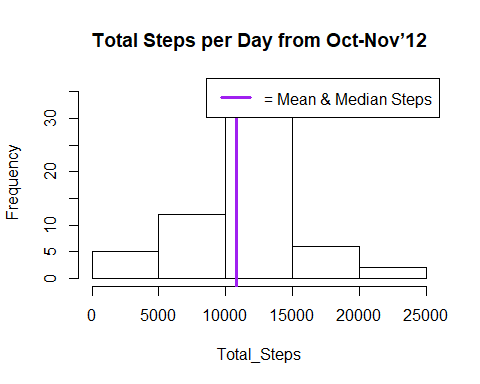
hist(data5$steps, xlab = "Total\_Steps", ylab = "Frequency", main = "Total Steps per Day from Oct-Nov’12")  
  
x <- mean(data5$steps)  
  
print(x)

## [1] 10766.19

y <- median(data5$steps)  
  
print(y)

## [1] 10766.19

abline(v = mean(data5$steps),col = "purple", lwd = 3)  
  
abline(v = median(data5$steps), col = "purple", lwd = 3)  
  
legend("topright", legend = "= Mean & Median Steps", lwd = 3, col = "purple")



Report: the values do indeed differ from the estimates from the first part of the assignment, as one might expect, with the impact of imputing missing data on the estimates of the total daily number of steps being augmentation of the total daily number of steps sums. Therefore, the histogram distribution is more a bell shaped curve. The mean and median change, as well, being equal in this context.

## Are there differences in activity patterns between weekdays and weekends?

Convert date from factor to date objects in data4 and double-check work

data4$date <- as.Date(data4$date, format = "%m/%d/%Y")  
  
colSums(is.na(data4))

## steps date interval   
## 0 0 0

class(data4$date)

## [1] "Date"

Add to data4 a new character variable / column specifying the day of the week for the particular date

data4$day <- weekdays(data4$date, abbreviate=TRUE)

Replace the day of the week with a “weekday” or “weekend” in the day by calling gsub()

data4$day <- gsub("Mon", "weekday", data4$day, ignore.case = TRUE)  
  
data4$day <- gsub("Tue", "weekday", data4$day, ignore.case = TRUE)  
  
data4$day <- gsub("Wed", "weekday", data4$day, ignore.case = TRUE)  
  
data4$day <- gsub("Thu", "weekday", data4$day, ignore.case = TRUE)  
  
data4$day <- gsub("Fri", "weekday", data4$day, ignore.case = TRUE)  
  
data4$day <- gsub("Sat", "WEEKEND", data4$day, ignore.case = TRUE)  
  
data4$day <- gsub("Sun", "WEEKEND", data4$day, ignore.case = TRUE)

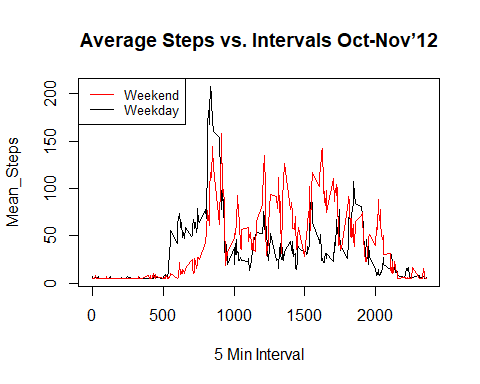
Make a panel plot containing a time series plot type = “l” of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all weekday days or weekend days (y-axis).

Create dataframes of the mean of steps taken per interval during the weekdays and weekends

data6 <- data4[data4$day == "weekday",]  
  
data7 <- data4[data4$day == "WEEKEND",]  
  
Mean\_Weekday <- aggregate(x = list(steps =data6$steps), by = list(interval = data6$interval), FUN = mean, na.rm = FALSE)  
  
Mean\_Weekend <- aggregate(x = list(steps =data7$steps), by = list(interval = data7$interval), FUN = mean, na.rm = FALSE)

Make a panel plot containing a time series plot 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)

plot(Mean\_Weekday$interval, Mean\_Weekday$steps, type = "l", xlab = "5 Min Interval", ylab = "Mean\_Steps", main = "Average Steps vs. Intervals Oct-Nov’12")  
  
lines(x = Mean\_Weekend$interval, y = Mean\_Weekend$steps, col = "red")  
  
legend("topleft", legend = c("Weekend", "Weekday"), col = c("red", "black"), lty = 1:1, cex = 0.8)



The red line on the graph suggests there is more activity on the weekends.