Reproducible research - assigment 1

Isaac Freites

December 8, 2019

Introduction of the Assigment -

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-utilized both because the raw data are hard to obtain and there is a lack of statistical methods and software for processing and interpreting the data.

This assignment makes use of data from a personal activity monitoring device. This 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 can be downloaded from the course web site:

Dataset: Activity monitoring data (https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip) [52K]

Intall packages:

- 1- ggplot2.
- 2- markdown.
- 3- dplyr.
- 4- stats.
- 5- data.table.
- 5- tidyr.

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.5.3
```

```
library(rmarkdown)
```

```
## Warning: package 'rmarkdown' was built under R version 3.5.3
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.5.3
```

```
##
## 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
```

```
library(stats)
library(data.table)
```

```
## Warning: package 'data.table' was built under R version 3.5.3
```

```
##
## Attaching package: 'data.table'
```

```
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 3.5.3
```

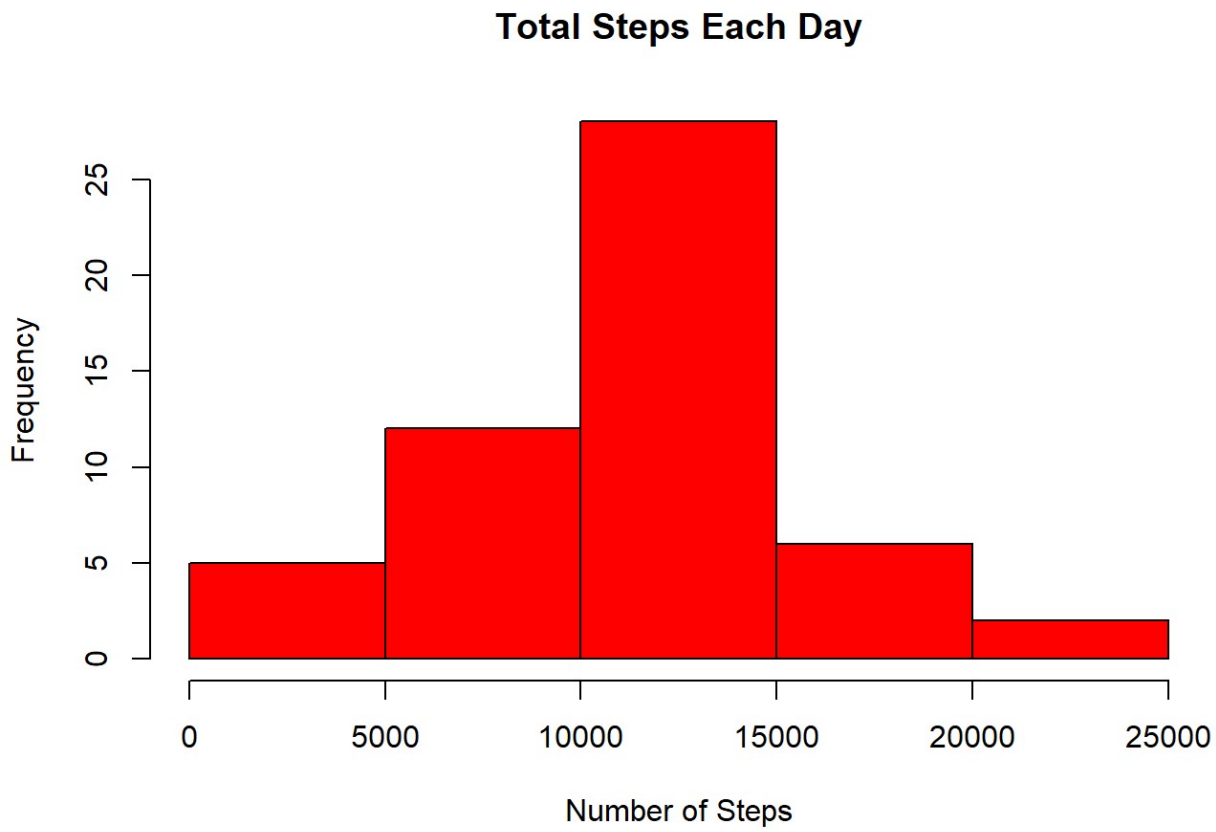
```
library(lattice)
```

```
## Warning: package 'lattice' was built under R version 3.5.3
```

Questions

a- What is mean total number of steps taken per day?

```
# Set Global Echo = On

# Load data
if (!file.exists("activity.csv") )
    {
     dlurl <- 'http://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.z
ip'
     download.file(dlurl,destfile='repdata%2Fdata%2Factivity.zip',mode='wb')
     unzip('repdata%2Fdata%2Factivity.zip')
    }

# Read data
data <- read.csv("activity.csv")

# Calculate total number of steps per day
steps_by_day <- aggregate(steps ~ date, data, sum)

hist(steps_by_day$steps, main = paste("Total Steps Each Day"), col="red",xlab
="Number of Steps")
```




```
rmean <- mean(steps_by_day$steps)
```

The mean steps taken per day is:

```
rmean
```

```
## [1] 10766.19
```

The median steps taken per day is:

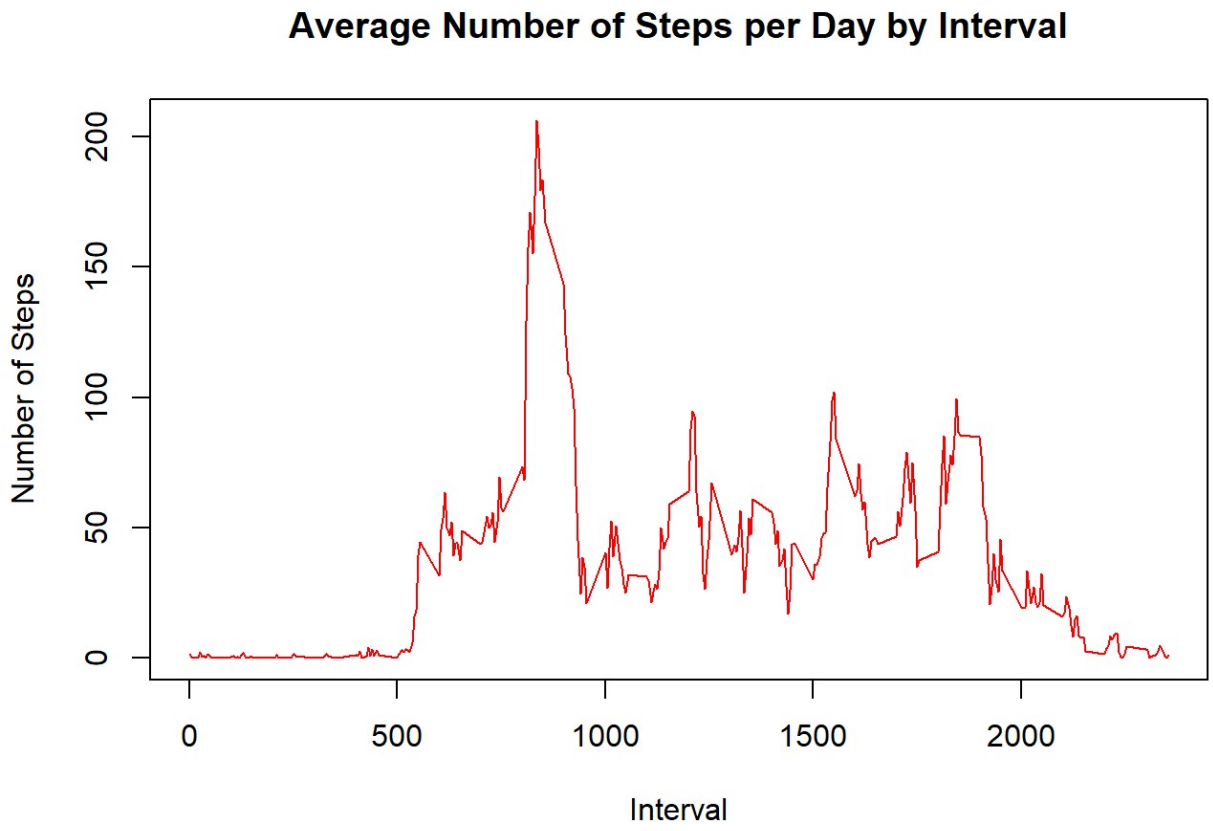
```
rmedian <- median(steps_by_day$steps)  # calculate media

rmedian # print median
```

```
## [1] 10765
```

b.What is the average daily activity pattern?

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

```
steps_by_interval <- aggregate(steps ~ interval, data, mean)

plot(steps_by_interval$interval,steps_by_interval$steps, col="red",type="l", xl
ab="Interval", ylab="Number of Steps",main="Average Number of Steps per Day by
Interval")
```



2- Which 5-minute interval, on average across all the days in the dataset, contains the maximum number of steps? The answer is:

```
max_interval <- steps_by_interval[which.max(steps_by_interval$steps),1] # calcu
late maximum

max_interval # print maximun
```

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

c.Imputing missing values

Note that there are a number of days/intervals where there are missing values (coded as NA). 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 in the dataset (i.e. the total number of rows with NAs).

Mising values is:

```
data <- data.table::fread(input = "activity.csv")
Missing_values<- data[is.na(steps), .N ]
Missing_values
```

```
## [1] 2304
```

```
# or
NATotal <- sum(!complete.cases(data))
```

2- 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.

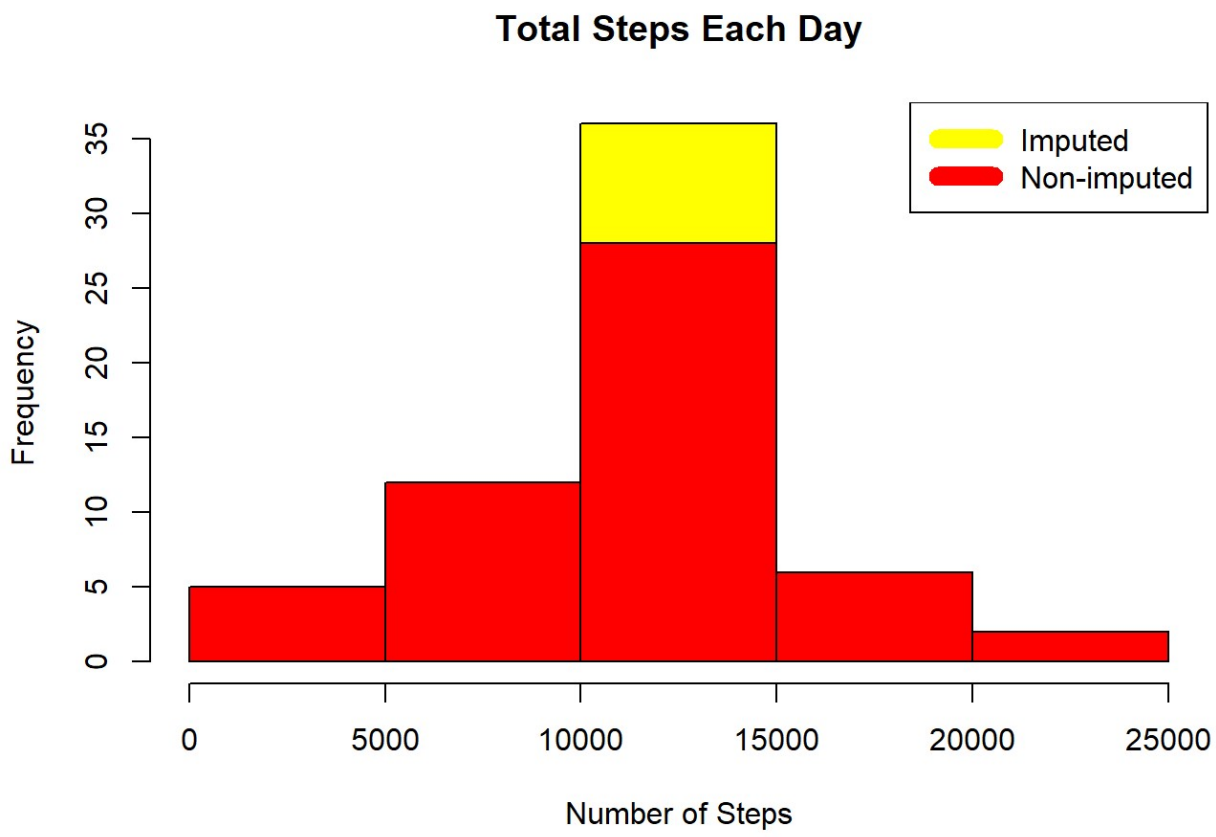
```
# Filling in missing values with median of dataset.

StepsAverage <- aggregate(steps ~ interval, data = data, FUN = mean)
fillNA <- numeric()
for (i in 1:nrow(data)) {
    obs <- data[i, ]
    if (is.na(obs$steps)) {
        steps <- subset(StepsAverage, interval == obs$interval)$steps
    } else { steps <- obs$steps}
    fillNA <- c(fillNA, steps)} # Calculating mean of missing values
```

3- Create a new dataset that is equal to the original dataset but with the missing data filled in.

```
new_activity <- data  # create new dataste
new_activity$steps <- fillNA # this database include average of missing values
```

4- 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?

```
StepsTotalUnion <- aggregate(steps ~ date, data = new_activity, sum, na.rm = TR
UE)
hist(StepsTotalUnion$steps, main = paste("Total Steps Each Day"), col="yello
w", xlab="Number of Steps") #Create Histogram to show difference
hist(steps_by_day$steps, main = paste("Total Steps Each Day"), col="red", xlab
="Number of Steps", add=T)
legend("topright", c("Imputed", "Non-imputed"), col=c("yellow", "red"), lwd=10)
```



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.

```
weekdays <- c("Monday", "Tuesday", "Wednesday", "Thursday",
              "Friday")
new_activity$dow = as.factor(ifelse(is.element(weekdays(as.Date(new_activity$da
te)),weekdays), "Weekday", "Weekend"))
StepsTotalUnion <- aggregate(steps ~ interval + dow, new_activity, mean)

weekdays
```

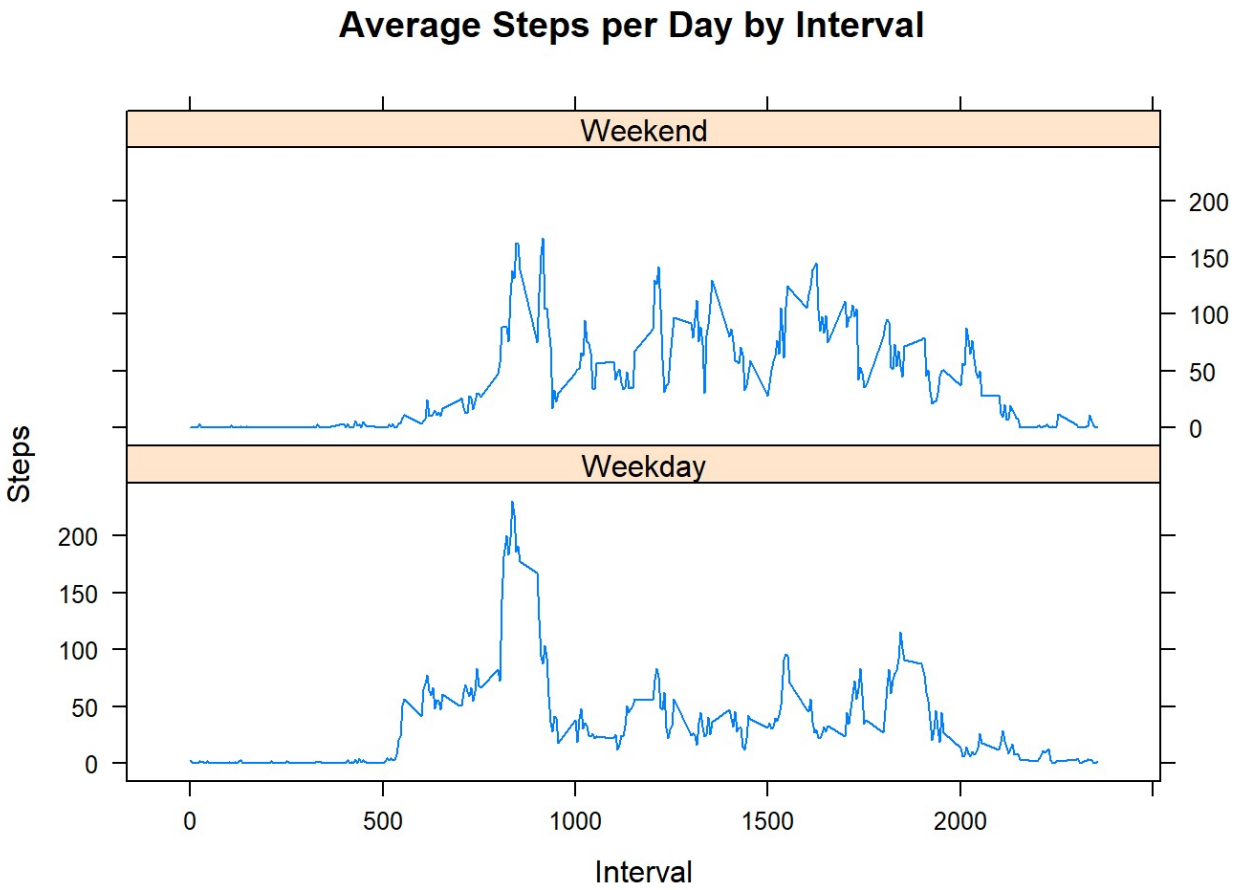
```
## [1] "Monday"    "Tuesday"   "Wednesday" "Thursday"  "Friday"
```

```
new_activity
```

```
##            steps       date interval     dow
##     1: 1.7169811 2012-10-01        0 Weekday
##     2: 0.3396226 2012-10-01        5 Weekday
##     3: 0.1320755 2012-10-01       10 Weekday
##     4: 0.1509434 2012-10-01       15 Weekday
##     5: 0.0754717 2012-10-01       20 Weekday
##    ---
## 17564: 4.6981132 2012-11-30     2335 Weekday
## 17565: 3.3018868 2012-11-30     2340 Weekday
## 17566: 0.6415094 2012-11-30     2345 Weekday
## 17567: 0.2264151 2012-11-30     2350 Weekday
## 17568: 1.0754717 2012-11-30     2355 Weekday
```

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

```
xyplot(StepsTotalUnion$steps ~ StepsTotalUnion$interval|StepsTotalUnion$dow, ma
in="Average Steps per Day by Interval",xlab="Interval", ylab="Steps",layout=c
(1,2), type="l")  # create a plot
```


Average Steps per Day by Interval
Weekend
Weekday
Steps
Interval
0
500
1000
1500
2000
50
100
150
200