Reproducible research

Sourav

30 November 2017

This is the markdown document fot the reproducible reasearch week 2

Background

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.

Getting the data

```r
setwd("D:/Git/Repres")
unzip("repdata.zip")
basedata <- read.csv("activity.csv")

# Check data

head(basedata)
```

```
##   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
# checking for volume of data availalble

dim(basedata)
```

```
## [1] 17568     3
```

```r
compl <- complete.cases(basedata) # creating logical matrix of NA

base <- basedata[compl,] # dropping the NA values and creating anew dataframe

total <- aggregate(base$steps, by = list(base$date), sum) # taking total daily steps

head(total) # checking data
```

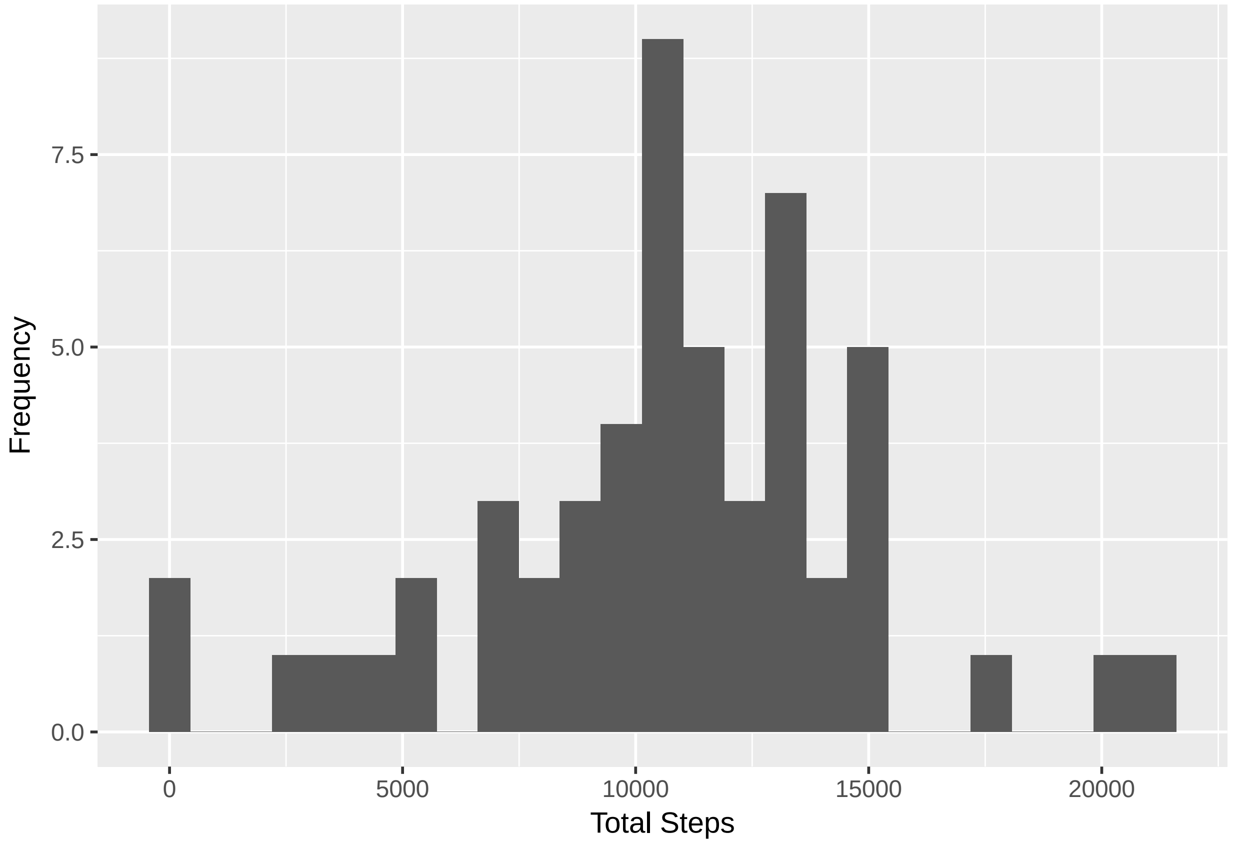
```
##       Group.1     x
## 1 2012-10-02   126
## 2 2012-10-03 11352
## 3 2012-10-04 12116
## 4 2012-10-05 13294
## 5 2012-10-06 15420
## 6 2012-10-07 11015
```

```
names(total)[1] = "Date" # renaming the date column
names(total)[2] = "Total Steps" # renaming the total steps column

library(ggplot2) # loading ggplot
```

```
## Warning: package 'ggplot2' was built under R version 3.4.2
```

```
qplot(total$`Total Steps`, bins = 25, xlab = "Total Steps", ylab = "Frequency")
```



This Histogram can be called gaussian - hence the mean and the median would be close to each other

Calculating the mean and the median

```
mean(total$`Total Steps`); median(total$`Total Steps`)
```

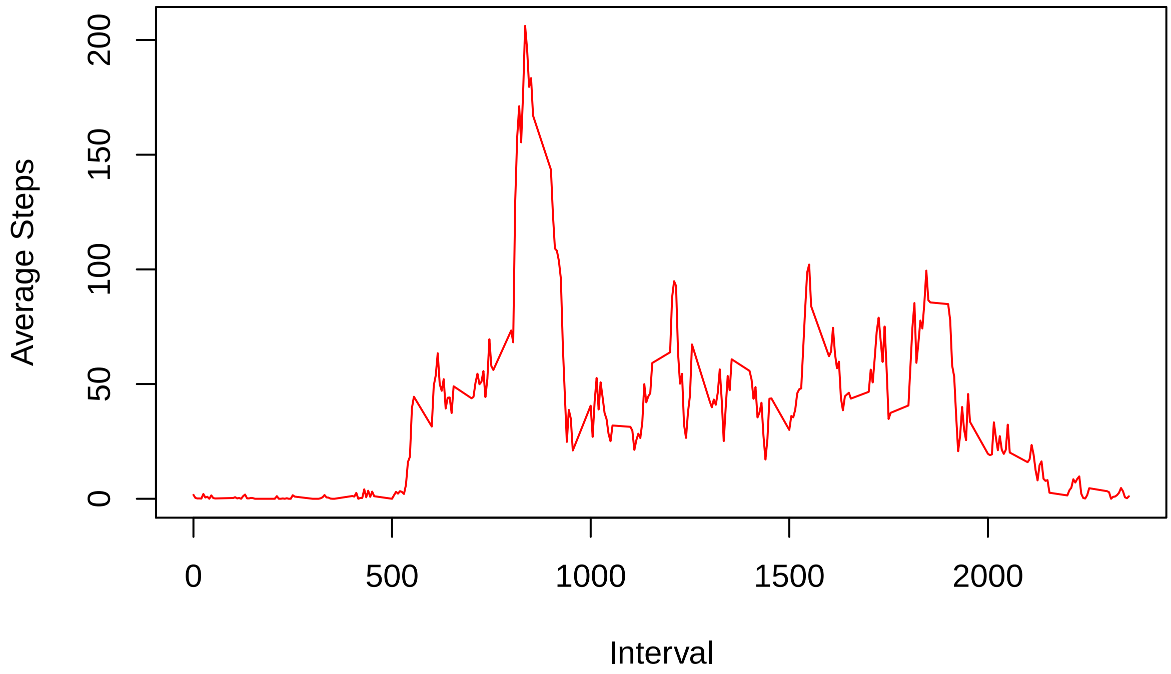
```
## [1] 10766.19
```

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

Making a time series graph on the 5 min interval and average no of steps

```
avg <- aggregate(base$steps, by = list(base$interval), mean)
names(avg)[1] = "Interval"
names(avg)[2] = "Average Steps"

plot(avg$Interval, avg$`Average Steps`, type = "l",col = "red", xlab = "Interval", ylab = "Average Step
```



Answering the question

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

```
avg[which.max(avg$`Average Steps`),]
```

```
##     Interval Average Steps
## 104      835      206.1698
```

hence the interval 835 contains the maximum number of average steps

Inputing missing Values

1st Step : Finding the missing values in the base data

```
# checking for NA values in all three columns
sum(is.na(basedata$steps) == TRUE); sum(is.na(basedata$date) ==TRUE); sum(is.na(basedata$interval) == T
## [1] 2304
```

```
## [1] 0
```

```
## [1] 0
```

So, there are 2304 NA/Missing values in the steps column

Strategy for replacing the Missing values

The missing values can be replaces be either the mean or the median of the steps dataset

for the purpose of this excercise, I would be using the mean.

While I would be making the histogram and also reporting the mean and the median, the mean and the median would not change significantly since they were close to each other). Since I would be using the median to replace the missing values, the mean would remain the same while the median would increase a bit

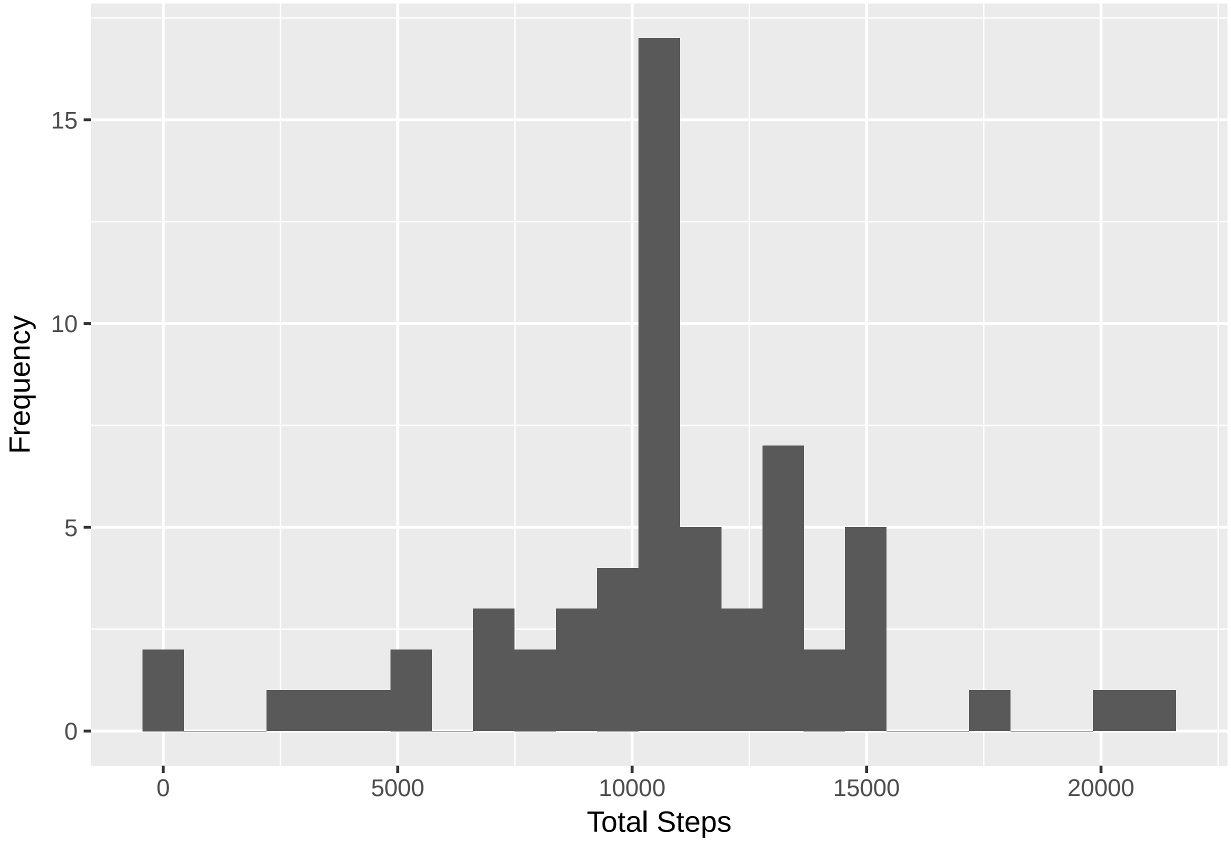
```r
# creating a new data set with missing values filled in
basenew <- basedata
basenew$steps <- ifelse(is.na(basenew$steps) == TRUE, mean(basenew$steps, na.rm = T), basenew$steps)
```

Plotting the histogram

```r
total1 <- aggregate(basenew$steps, by = list(basenew$date), sum) # taking total daily steps

names(total1)[1] = "Date" # renaming the date column
names(total1)[2] = "Total Steps" # renaming the total steps column
library(ggplot2) # adding library

qplot(total1$`Total Steps`, bins = 25, xlab = "Total Steps", ylab = "Frequency")
```



### Now would be calculating the mean and median

```
mean(total1$`Total Steps`); median(total1$`Total Steps`)
```

```
## [1] 10766.19

## [1] 10766.19
```

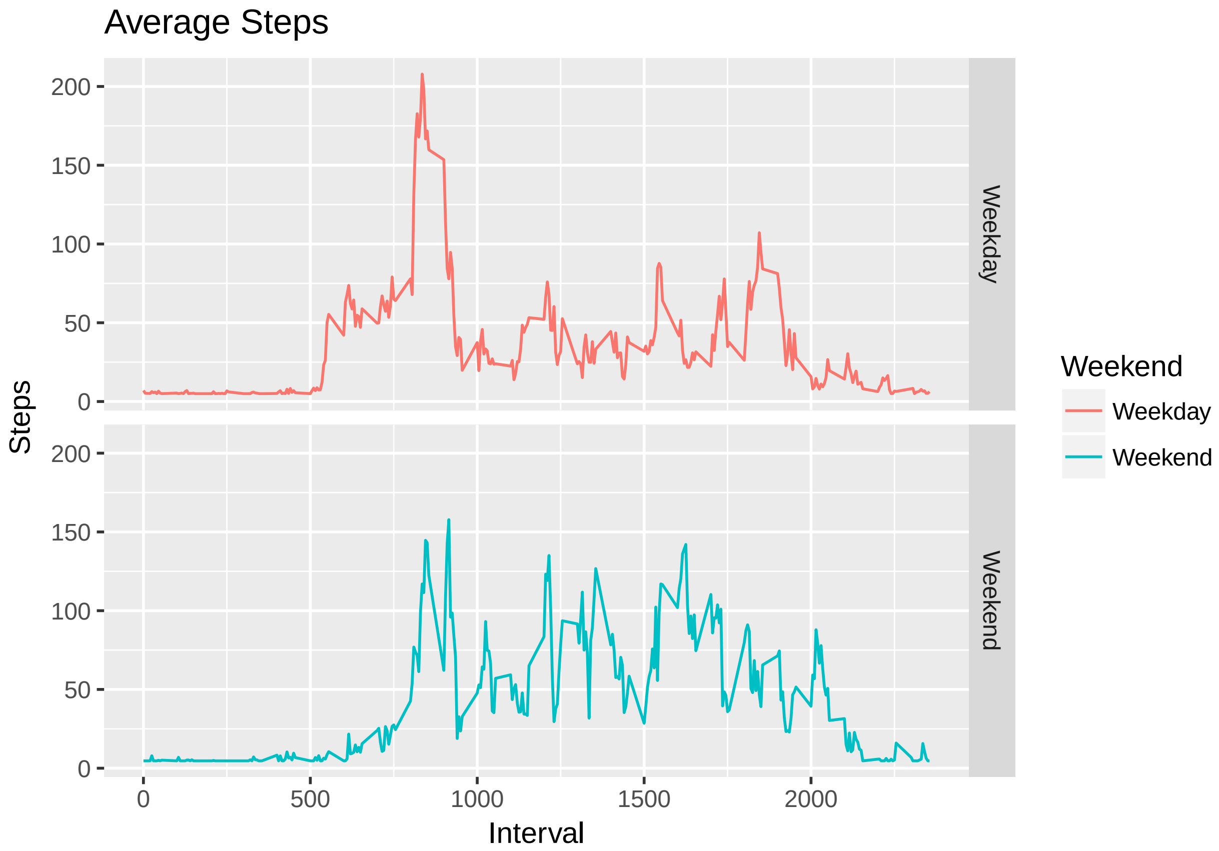
We see that the mean did not change but the median did increase

Now I would be creating the segregation basis Weekday and weekend and plotting a graph accorddingly

```
basenew$date1 <- as.Date(basenew$date) # changing to date format
basenew$day <- weekdays(basenew$date1) # getting the day of the week
basenew$weekend <- ifelse(basenew$day == "Sunday"|basenew$day == "Saturday", "Weekend", "Weekday") # Up
weekenddata <- aggregate(basenew$steps, by = list(basenew$weekend, basenew$interval), mean) # Finding a

names(weekenddata)[1] = "Weekend"
names(weekenddata)[2] = "Interval"
names(weekenddata)[3] = "Steps"

g <- ggplot(weekenddata, aes(x = Interval, y = Steps, colour = Weekend )) # base
g <- g + geom_line() # defining line type
g <- g + facet_grid(Weekend ~ .) # adding facets
g <- g + labs( title = "Average Steps", x = "Interval", y = "Steps") # addting titles
g # Graph output
```


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