Reproducible Research: Peer Assessment 1

20/7/2021

Loading and preprocessing the data

Here we unzip the file and load the dataset into R. As well as take a look into the data.

```
setwd("C:/Users/Daniela/Documents/R Coursera")
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.0.5
```

```
## Registered S3 methods overwritten by 'tibble':
##   method     from
##   format.tbl pillar
##   print.tbl  pillar
```

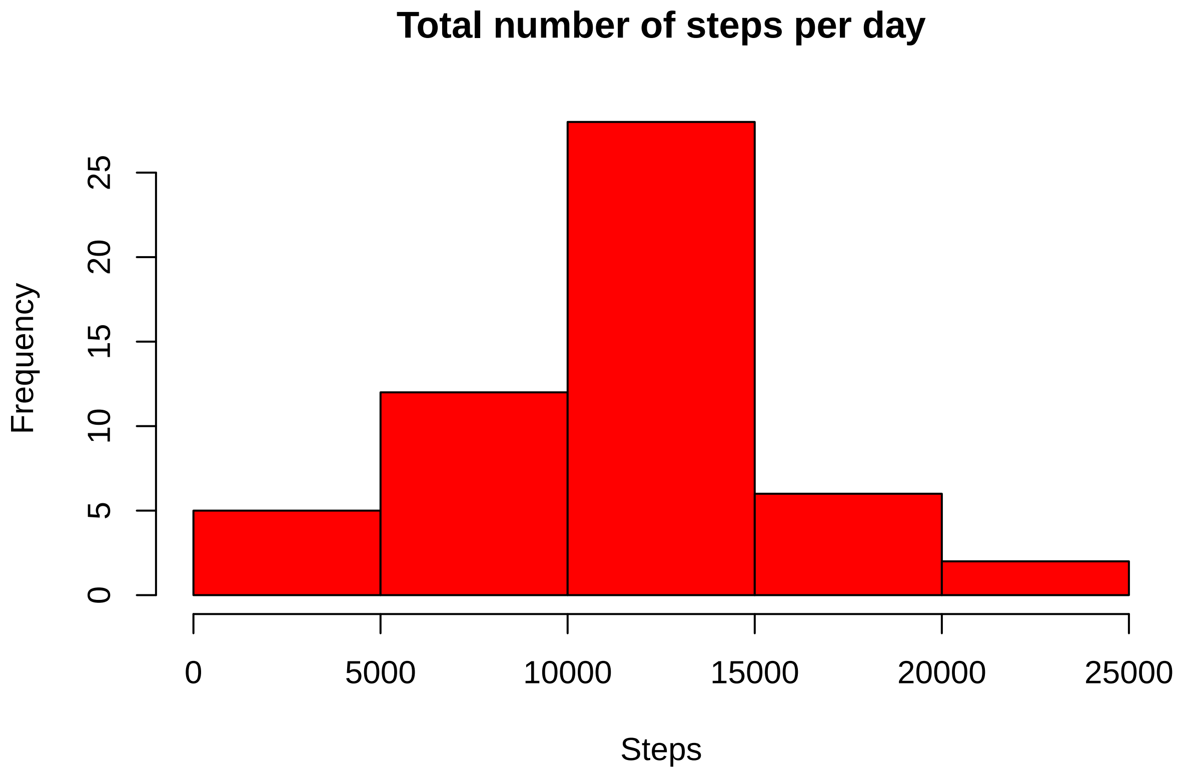
```
unzip(zipfile = "repdata_data_activity.zip")
RepData <- read.csv(file= "activity.csv", header = TRUE)
head(RepData)
```

```
##   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
```

What is mean total number of steps taken per day?

First we calculate the total number of steps and make a histogram of the total number of steps taken each day.

```
TotSteps <- aggregate(steps ~ date, RepData, FUN = sum)
hist(TotSteps$steps, main= "Total number of steps per day", xlab = "Steps", col = "red")
```



Then we calculate the mean and median number of steps taken each day

```
MeanSteps <- mean(TotSteps$steps, na.rm = TRUE)
MedianSteps <- median(TotSteps$steps, na.rm = TRUE)
MeanSteps
```

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

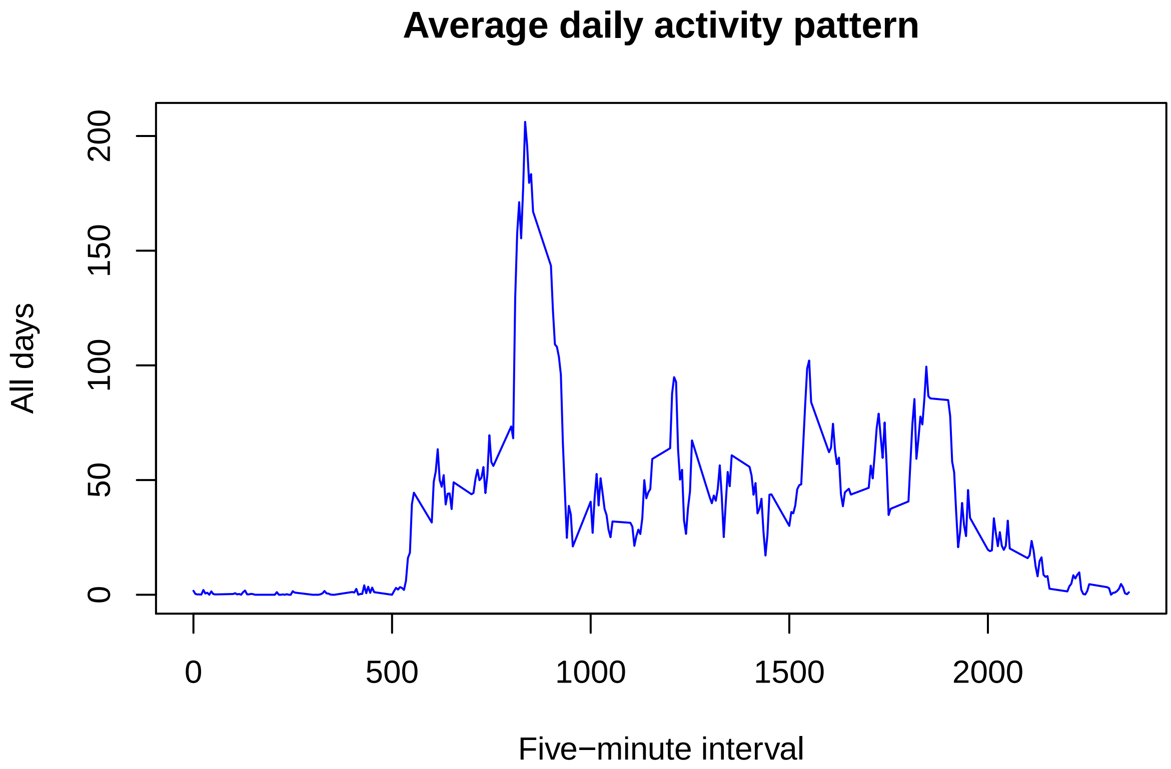
```
MedianSteps
```

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

What is the average daily activity pattern?

Then we calculate a time series plot of the 5-minute interval and average number of steps taken to understand which 5-minute interval conatins the maximum number of steps

```
FMinInt <- aggregate(steps ~ interval, RepData, mean)
plot(x = FMinInt$interval, y = FMinInt$steps, type = "l", xlab = "Five-minute interval", ylab = "All da
```



```
# Finding the 5-minute interval that contains the maximum number of steps
MaxInt <-
MaxSteps <- max(FMinInt$steps)
for(i in 1:288)
{
  if(FMinInt$steps [i] == MaxSteps)
    FMinIntMax <- FMinInt$interval[i]
}
FMinIntMax <- FMinInt[which.max(FMinInt$steps),]
```

Imputing missing values

Here we are calculating all total missing values in the dataset

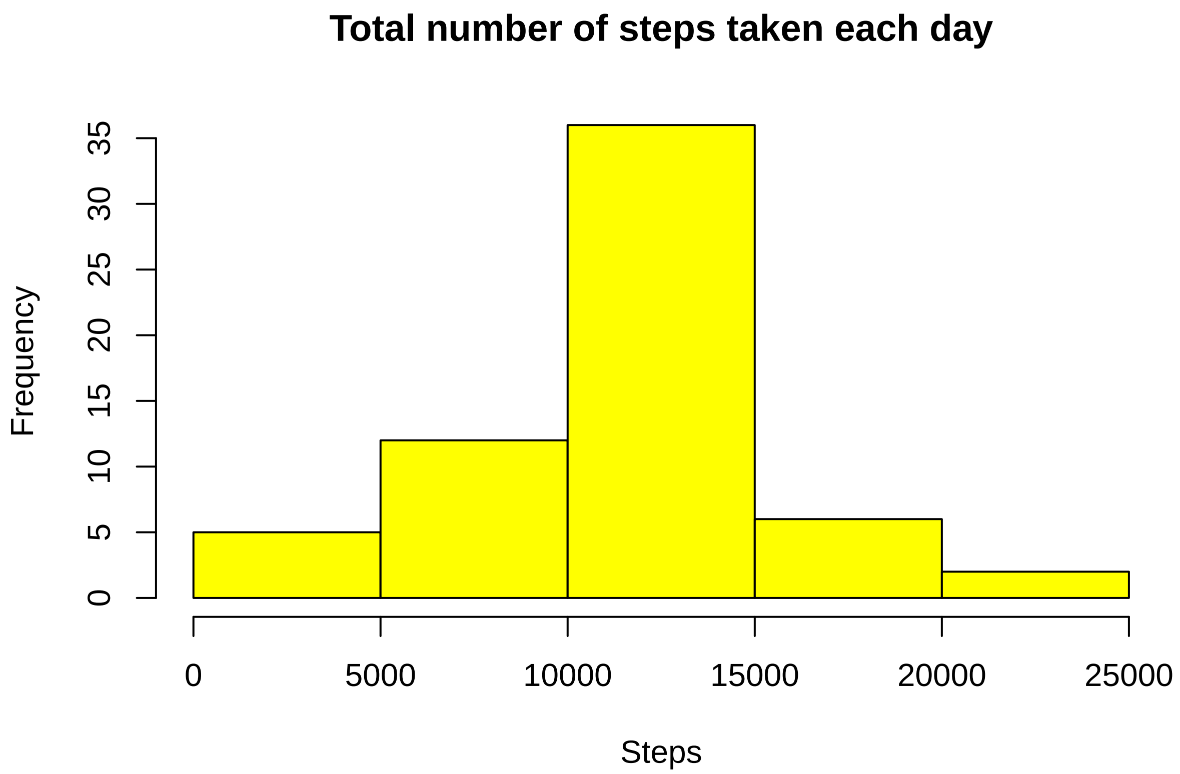
```
total_na <- is.na(RepData$steps)
summary(total_na)
```

```
##    Mode   FALSE    TRUE
## logical   15264    2304
```

Then we create a new dataset equal to the original but with no missing data, using the mean to fill the missing values and creating a histogram

```r
# Replace missing values in new dataset
 # replacing the na by the mean in that fime minute interval

NoNARepData <- transform(RepData, steps = ifelse(is.na(RepData$steps),  FMinInt$steps[match(RepData$inte
StepsByInt <- aggregate(steps ~ date, NoNARepData, FUN = sum)
# Create histogram
hist(StepsByInt$steps, main = "Total number of steps taken each day", xlab = "Steps", col = "yellow")
```



```r
# Report Mean and Median of the total number of steps taken per day
IntMeanSteps <- mean(StepsByInt$steps, na.rm = TRUE)
IntMedianSteps <- median(StepsByInt$steps, na.rm = TRUE)
DifferenceMean <- IntMeanSteps - MeanSteps
DifferenceMedian <- IntMedianSteps - MedianSteps
DifferenceMean
```

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

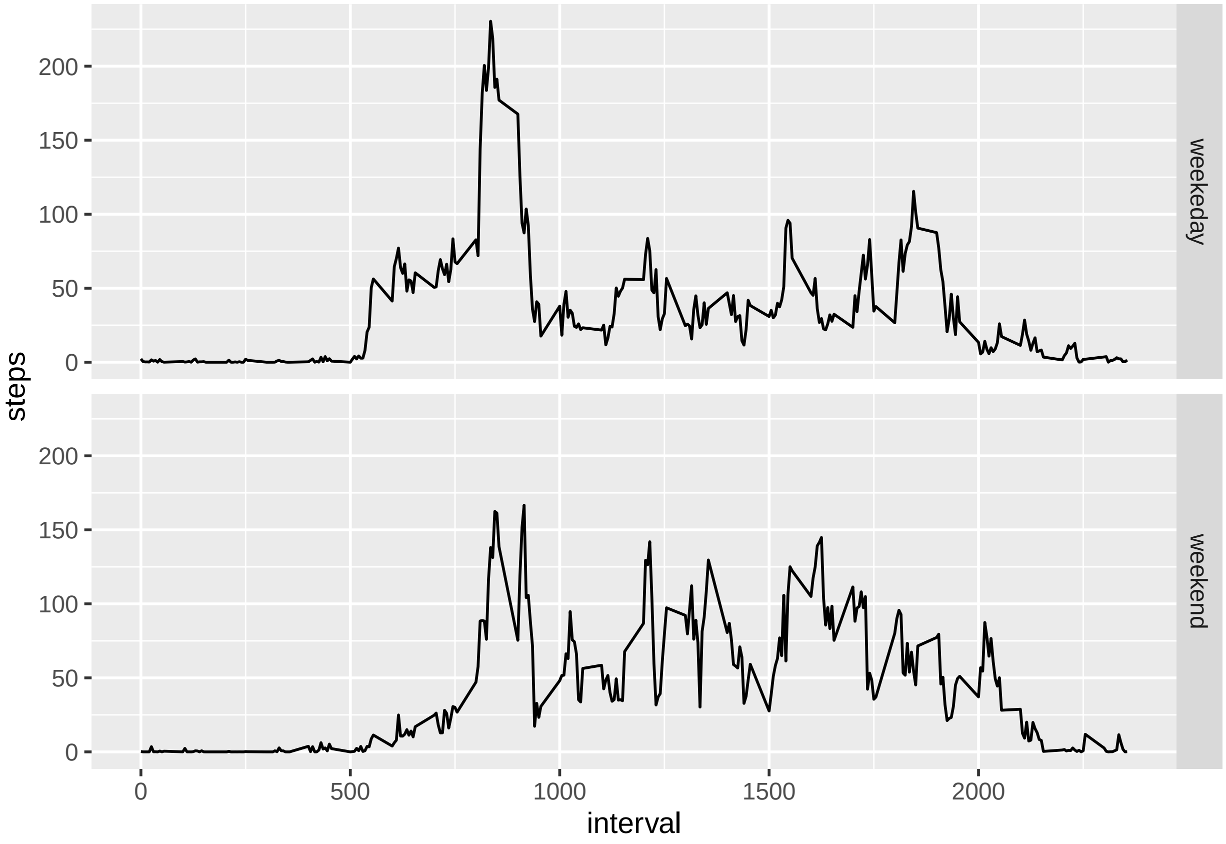
```r
DifferenceMedian
```

```
## [1] 1.188679
```

There is no difference in the mean and the median has increased. The values do not differe greatly from the estimates. However, there is impact when imputing missing data, specially when estimating the median of the total daily number of steps

Are there differences in activity patterns between weekdays and weekends?

```
DayType <- function(date) {
  day <- weekdays(date)
  if (day %in% c('lunes', 'martes', 'miércoles', 'jueves', 'viernes'))
      return ("weekeday")
  else if (day %in% c('sábado', 'domingo'))
      return ("weekend")
  else
      stop ("Invalid Date Format.")
}
NoNARepData$date <- as.Date(NoNARepData$date)
NoNARepData$day <- sapply(NoNARepData$date, FUN = DayType)
MeansStepsDay <- aggregate(steps ~ interval + day, NoNARepData, mean)
ggplot(data = MeansStepsDay, aes(x = interval, y = steps))+ geom_line() + facet_grid(day ~.)
```


weekeday
steps
weekend
interval