Reproducible Research Course Project 1

Mlibazisi Ndlovu

19 July 2020

Reproducible Research Project 1

The starting point is to load the fitness data

```
#create the working directory

if (!file.exists("data")) {
    dir.create("data")
}

# Fork/clone via git bash the following repsoitory which contains the data:
https://github.com/rdpeng/RepData_PeerAssessment1

#read the data files into a data frame.
unzip(zipfile = "activity.zip")
activity = read.csv("activity.csv")
```

The next step is a quick overview of the data

```
summary(activity)

##      steps                date          interval
##  Min.   :  0.00   2012-10-01:  288   Min.   :   0.0
##  1st Qu.:  0.00   2012-10-02:  288   1st Qu.: 588.8
##  Median :  0.00   2012-10-03:  288   Median :1177.5
##  Mean   : 37.38   2012-10-04:  288   Mean   :1177.5
##  3rd Qu.: 12.00   2012-10-05:  288   3rd Qu.:1766.2
##  Max.   :806.00   2012-10-06:  288   Max.   :2355.0
##  NA's   :2304     (Other)   :15840

dim(activity)

## [1] 17568     3

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
```

```
tail(activity)
```

```
##       steps       date interval
## 17563    NA 2012-11-30     2330
## 17564    NA 2012-11-30     2335
## 17565    NA 2012-11-30     2340
## 17566    NA 2012-11-30     2345
## 17567    NA 2012-11-30     2350
## 17568    NA 2012-11-30     2355
```

Then clean and process the data

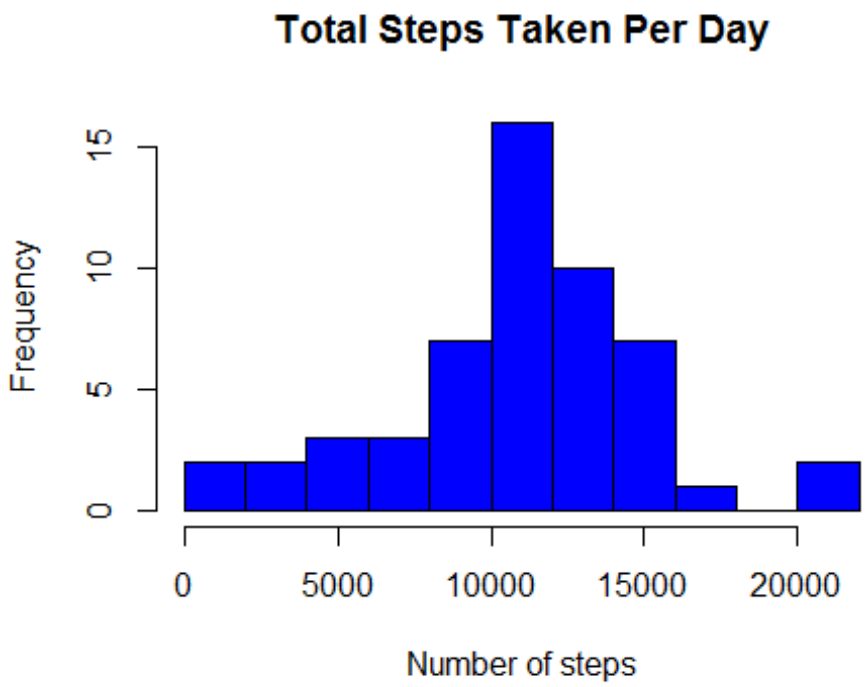
```
# Remove the rows with missing data

activity_clean <- na.omit(activity)

# create dataframes grouped by day and interval

activity_clean2 <- tbl_df(activity_clean)
act_date <- group_by(activity_clean2, date)
act_interval <- group_by(activity_clean2, interval)
```

Calcualte the average total number of steps per day

```
daily_steps <- summarise(act_date, sum(steps))
```

Plot a histogram of the number average total number of steps per day

```
hist(daily_steps$`sum(steps)`,col = "blue",breaks = 9 ,main = "Total Steps
Taken Per Day", xlab = "Number of steps", ylab = "Frequency")
```

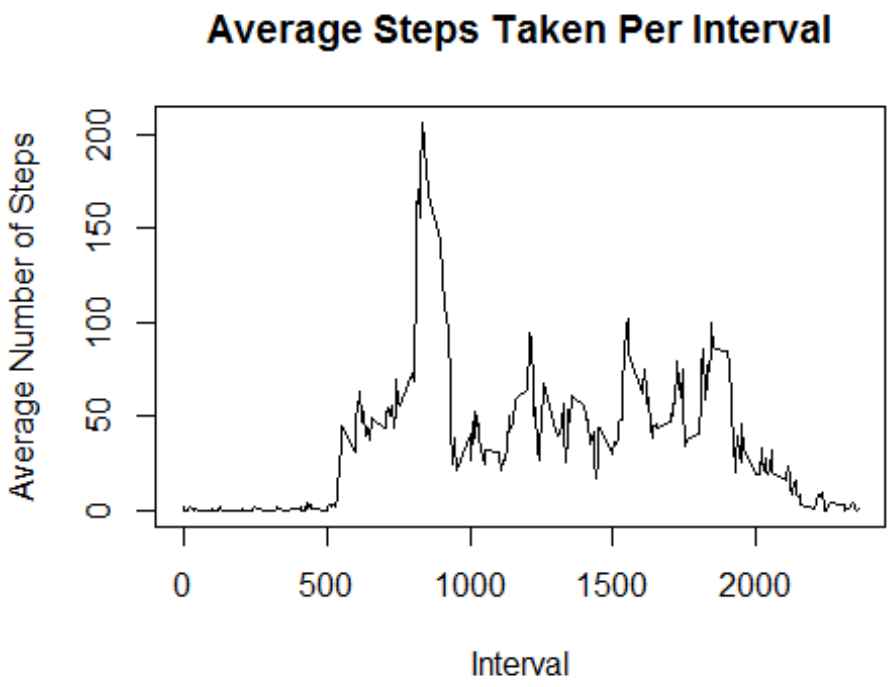


Calculate the mean and median for the daily total number of steps

```
mean(daily_steps$`sum(steps)`)

## [1] 10766.19

median(daily_steps$`sum(steps)`)

## [1] 10765
```

Average daily activity pattern analysis

```
#Obtain the average number of steps per interval acoss all the days
pattern <- summarise(act_interval, mean(steps))

# Plot the average total number of steps by interval
plot(pattern$interval,pattern$`mean(steps)`, type = "l",main = "Average Steps
Taken Per Interval", xlab = "Interval", ylab = "Average Number of Steps" )
```




```
# Determine and report the interval with the most number of steps on average
w = as.character(pattern[which.max(pattern$`mean(steps)`),][1])
x = "The five minute interval containing the maximum steps on average is: "
print(c(x,w))

## [1] "The five minute interval containing the maximum steps on average is:
"
## [2] "835"
```

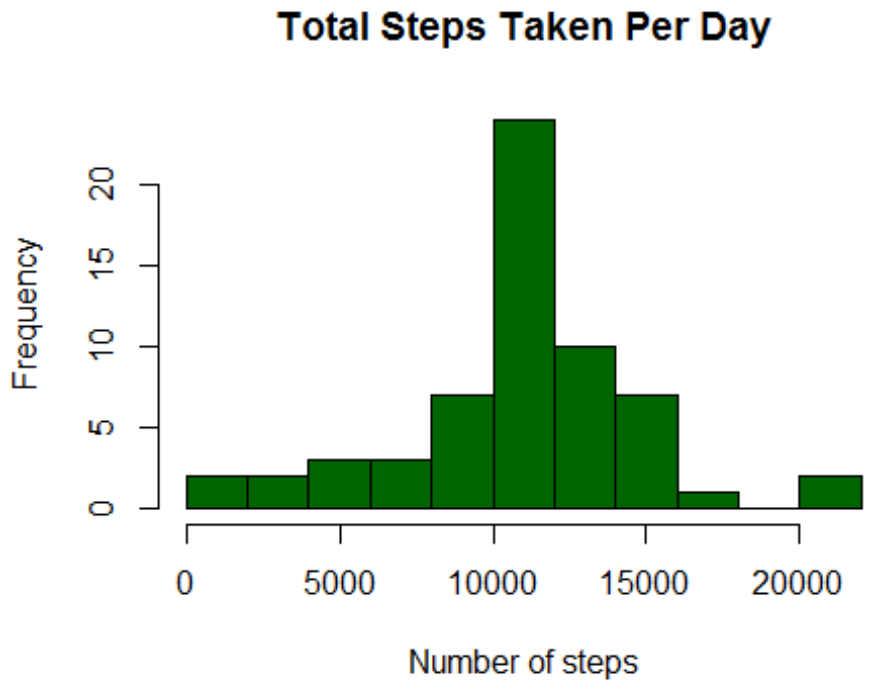
Perfoming analysis with the missing values being imputed

```
 #Obtain and report the count of the missing values
Missing_data <- sum(is.na(activity))
Missing_data

## [1] 2304

y = "The number of missing values is: "
print(c(y,Missing_data))

## [1] "The number of missing values is: " "2304"

# Impute the missing values with the mean for each interval
avgsteps <- rep(pattern$`mean(steps)`,61) #create a vector of the interval
averages
activ_avgs <- cbind(activity,avgsteps) #create a new dataframe with a column
for interval averages
```

```
activity_clean3 <- tbl_df(activ_avgs)
cleanactivity <- mutate(activity_clean3,steps_2 = ifelse(is.na(steps),
avgsteps,steps)) #The new dataframe with missing values imputed in cloumn
steps_2
```

Plot the histogram of daily total number of steps , wtih missing values imputed

```
act_date_2 <- group_by(cleanactivity, date) #group by date so as to analyse
steps per day.

daily_steps_2 <- summarise(act_date_2, sum(steps_2))

hist(daily_steps_2$`sum(steps_2)`, col = "dark green",breaks = 9 ,main =
"Total Steps Taken Per Day", xlab = "Number of steps", ylab = "Frequency")
```



Compare the means of the cleaned and uncleand data

```
means <- c(mean(daily_steps$`sum(steps)`),mean(daily_steps_2$`sum(steps_2)`))
medians <-
c(median(daily_steps$`sum(steps)`),median(daily_steps_2$`sum(steps_2)`))
comparison <- data.frame(means,medians, row.names = c("Raw Data"," Imputed
Data"))
comparison

##                   means  medians
## Raw Data      10766.19 10765.00
##  Imputed Data 10766.19 10766.19
```

Activity patterns by day of the week

```
#Add a column of days to the data

cleanactivity2 <- mutate(cleanactivity, Day =weekdays(as.Date(date)))
cleanactivity3 <- mutate(cleanactivity2, Daytype = ifelse(Day ==
"Saturday"|Day == "Sunday", "Weekend","Weekday"))

#Group data by intervals

act_interval2 <- group_by(cleanactivity3, interval)
pattern2 <- summarise(act_interval2, mean(steps_2))
interval_avg <- rep(pattern2$`mean(steps_2)`,61)

cleanctivity4 <- mutate(cleanactivity3,interval_avg)
```

Plot a graph of activity over the weekdays and weekends

```
g <- ggplot(cleanctivity4, aes(interval,interval_avg))
g + geom_line() + facet_grid(Daytype ~ .) + labs(title = "Activity Patterns
By Time of Week") + labs(x ="Five Minute Interval", y = "Total Average
Steps") + theme_light()
```