# **Reproducible Research Course Project 1**

## 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-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.

## Assignment

The course project has been designed in a way that we need to answer a series of questions that will be mentioned below. After answering the questions we need to make an **R markdown** file and add all the *code chunks* to it. One of the main observation during the entire project was there were a lot of warnings, that would have made the code look a tad bit ugly, so to avoid that I will amke the warnings as FALSE for the entire project, so that it doesn’t showup in the **html** file.

## Questions:

#### 1. Plot the histogram for steps as pe the day and add the mean and median to the output?

#### 2. Plot the average daily activity pattern and add the maximum average steps?

#### 3. How are we Imputing missing values?

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

## **Turning off the warnings**

knitr::opts\_chunk$set(warning = FALSE)

## **The data**

### Exploring the working directory

# First we check the files in the working directory  
dir()

## [1] "activity.csv" "repdata\_data\_activity.zip"   
## [3] "RepData\_PeerAssessment1-master.zip" "reproducible\_research.html"   
## [5] "reproducible\_research.Rmd" "reproducible\_research.tex"

We see that there is a file names “**activity.csv**” in the working directory that we will now load in a new variable named **activity**.

### Storing the data in a new variable

activity <- read.csv("activity.csv")

### Checking out the basic information about the data

# Checking the dimenssions of the activity dataset  
dim(activity)

## [1] 17568 3

# Looking at the summary of the activity dataset  
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

So we see that there are 3 variables namely:  
1. steps  
2. date  
3. interval  
There are **2304 NA’s** in *steps* variable, and many other informations are provided.

Now we need to make a histogram plot for the person based on the number of steps taken each day.

## **Question 1**

### Extracting the data

We first plot the histogram according to the steps each day. We need to keep in mind that there were a lot of NA’s in the steps variable. We create a variable “steps\_by\_date” that conatins the aggregated output of the steps of the activity dataset according to the dates, and has no NA values, and the function sum applied to it.

steps\_by\_date <- with(activity, aggregate(steps, by = list(date), FUN = sum, na.rm = TRUE))

### Checking the summary of steps\_by\_date variable

summary(steps\_by\_date)

## Group.1 x   
## 2012-10-01: 1 Min. : 0   
## 2012-10-02: 1 1st Qu.: 6778   
## 2012-10-03: 1 Median :10395   
## 2012-10-04: 1 Mean : 9354   
## 2012-10-05: 1 3rd Qu.:12811   
## 2012-10-06: 1 Max. :21194   
## (Other) :55

We see that there are 61 dates stored in the **Group.1** variable and the steps stored in the **x** variable and no **NA’s** are there. The names of the variable are confusing so we give them the names that we desire, and see the summary again to cross validate.

### Naming the coulumns

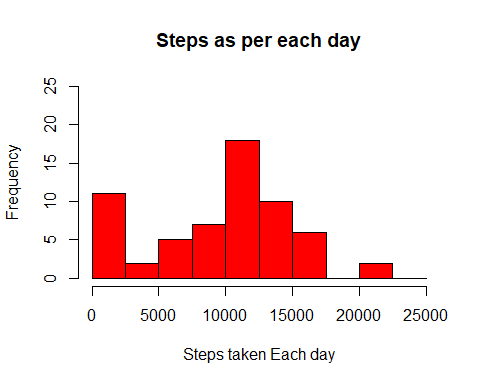
names(steps\_by\_date) <- c("dates", "steps")  
summary(steps\_by\_date)

## dates steps   
## 2012-10-01: 1 Min. : 0   
## 2012-10-02: 1 1st Qu.: 6778   
## 2012-10-03: 1 Median :10395   
## 2012-10-04: 1 Mean : 9354   
## 2012-10-05: 1 3rd Qu.:12811   
## 2012-10-06: 1 Max. :21194   
## (Other) :55

Now the data looks a lot better. We will plot the histogram.

### Ploting the histogram

hist(steps\_by\_date$steps, main = "Steps as per each day", xlab = "Steps taken Each day", ylim = c(0,25), col = "red", breaks = seq(0, 25000, by = 2500))



The summary of *“steps\_by\_date”* gave us the **mean** and the **median** of the *steps* but as it it asked in the question, we add the **mean** and **median** explicitly.

### Mean and Median

mean(steps\_by\_date$steps)

## [1] 9354.23

median(steps\_by\_date$steps)

## [1] 10395

## **Question 2**

We make a new variable “**mean\_steps\_per\_interval**” in which we store the aggregate of the steps by the interval and apply the mean function to the steps variable of the **activity** data set. Also we remove the **NA’s** from the steps variable.We then see the summary of the **mean\_steps\_per\_interval** variable.

mean\_steps\_per\_interval <- aggregate(activity$steps, by = list(activity$interval), FUN = mean, na.rm = TRUE)  
summary(mean\_steps\_per\_interval)

## Group.1 x   
## 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

We see that the columns in the **mean\_steps\_per\_interval** dataset are **Group.1** and **x**. We need to change the names of the columns as per our convnience, and check the summary again to cross validate.

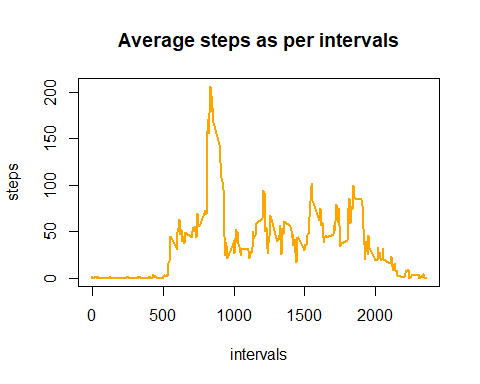
names(mean\_steps\_per\_interval) <- c("interval", "steps")  
summary(mean\_steps\_per\_interval)

## 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

### Making the plot

The plot between the intervals and the steps will be made now.

plot(mean\_steps\_per\_interval$interval, mean\_steps\_per\_interval$steps, type = "l", col = "orange", lwd = 2, xlab = "intervals", ylab = "steps", main = "Average steps as per intervals")



The interval with the maximum number of steps in the entire dataset is given in the following code chunk.

mean\_steps\_per\_interval[which.max(mean\_steps\_per\_interval$steps), ]$interval

## [1] 835

So the **835th** interval has the maximum number of average steps.

## **Question 3**

First we see how many missing values are there..

sum(is.na(activity$steps))

## [1] 2304

So there are **2,304** missing values in the *steps* variable of the **activity** dataset

Now we impute the data. The places where **NA** is there, we insert the mean instead and look at the summary of “**imputed\_data**” variable.

imputed\_data <- mean\_steps\_per\_interval$steps[match(activity$interval, mean\_steps\_per\_interval$interval)]  
summary(imputed\_data)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.000 2.486 34.113 37.383 52.835 206.170

Now we make a new dataset that is same as original but with filled in **NA’s**

activity\_imputed <- transform(activity, steps = ifelse(!is.na(activity$steps), no = imputed\_data, yes = activity$steps))  
total\_steps\_imputed <- aggregate(steps ~ date, activity\_imputed, sum)  
names(total\_steps\_imputed) <- c("date", "steps")

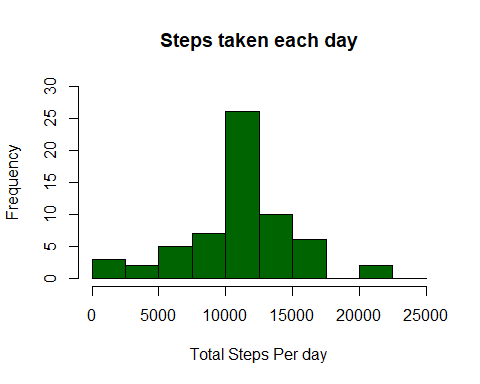
Now we look at the summary of the new data

summary(total\_steps\_imputed)

## date steps   
## 2012-10-01: 1 Min. : 41   
## 2012-10-02: 1 1st Qu.: 9819   
## 2012-10-03: 1 Median :10766   
## 2012-10-04: 1 Mean :10766   
## 2012-10-05: 1 3rd Qu.:12811   
## 2012-10-06: 1 Max. :21194   
## (Other) :55

Now we make the Histogram with this new imputed data

hist(total\_steps\_imputed$steps, col = "darkgreen", ylim = c(0,30), xlab = "Total Steps Per day", main = "Steps taken each day", breaks = seq(0, 25000, 2500))



Now we calculate the mean and median

### Mean and Median

# The mean is  
mean(total\_steps\_imputed$steps)

## [1] 10766.19

# The median is  
median(total\_steps\_imputed$steps)

## [1] 10766.19

## **Question 4**

In this part we need to see the activities during the weekdays and weekends differently. So, we first need to change the actual dataset and add the different days into it.

Now modify the activity dataset.

After changing the date column of the activity dataset we replace **Sunday** or **Saturday** with **weekend** and the rest with **weekday**.

activity$date <- as.Date(strptime(activity$date, format="%Y-%m-%d"))  
activity$datetype <- sapply(activity$date, function(x) {  
 if (weekdays(x) == "Saturday" | weekdays(x) =="Sunday")   
 {y <- "Weekend"} else   
 {y <- "Weekday"}  
 y  
 })

### Now we make the plot with this new data

new\_data <- aggregate(steps ~ interval + datetype, data = activity, mean, na.rm = TRUE)  
library(ggplot2)  
ggplot(new\_data, aes(x = interval, y = steps, color = datetype)) + geom\_line() +labs(title = "Average daily steps by type of date", x = "Interval", y = "Number of steps") + facet\_wrap(~datetype, nrow = 2, ncol = 1)

