REPRODUCIBLE RESEARCH:Course Project 1

July 18, 2017

```
knitr::opts_chunk$set(echo = TRUE)
```

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.

Loading and preprocessing the data The first step is to read data from the downloaded csv file. The file is already downloaded, decompressed and copied into the working directory.

```
setwd("C:/Users/brian/Desktop/Activity Monitoring Data")
df<-read.csv("activity.csv")
df2<-df

head(df)
```

```
##   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
```

Looking at the data and their class

```
names(df)
```

```
## [1] "steps"    "date"     "interval"
```

```
str(df)
```

```
## 'data.frame':    17568 obs. of  3 variables:
##  $ steps   : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ date    : Factor w/ 61 levels "2012-10-01","2012-10-02",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ interval: int  0 5 10 15 20 25 30 35 40 45 ...
```

As “date” is presented as a ‘Factor’ variable, Convert it to a variable of “Date” class.

```
df$date<-as.Date(df$date)
head(df$date)
```

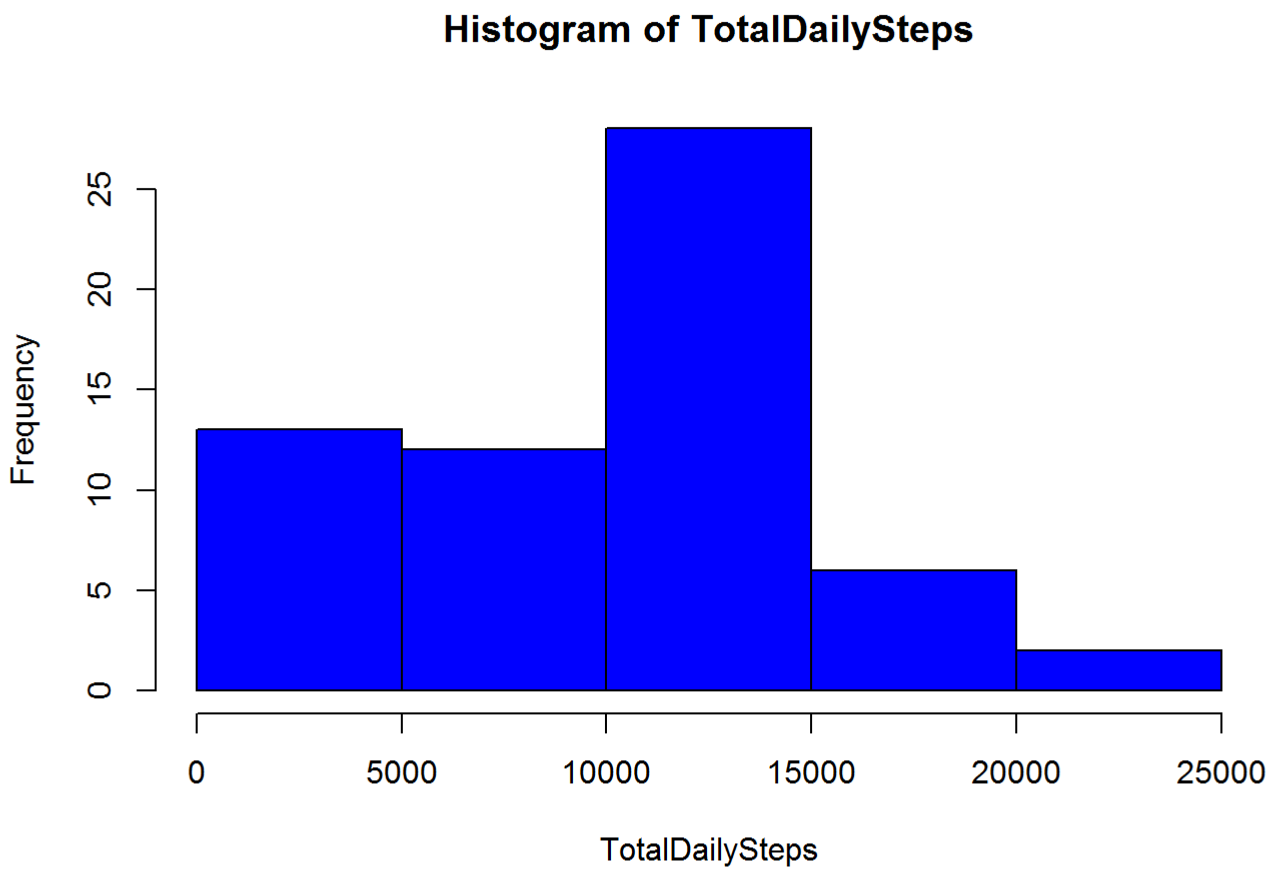
```
## [1] "2012-10-01" "2012-10-01" "2012-10-01" "2012-10-01" "2012-10-01"
## [6] "2012-10-01"
```

What is mean total number of steps taken per day? In this data frame, there are 17568 observations over 61 days period of the study. We should first calculate the total number of steps on each day, so, I used a ‘tapply’ function.

```
TotalDailySteps<-with(df, tapply(steps, date, sum, na.rm=TRUE))
```

Now, I can make the histogram of total steps taken each day. As this is a simple histogram, ‘base plotting system’ is sufficient.

```
hist(TotalDailySteps, col = "blue")
```



The next step is to report the mean and median of total number of steps taken each day. This is a simple and straight forward task!

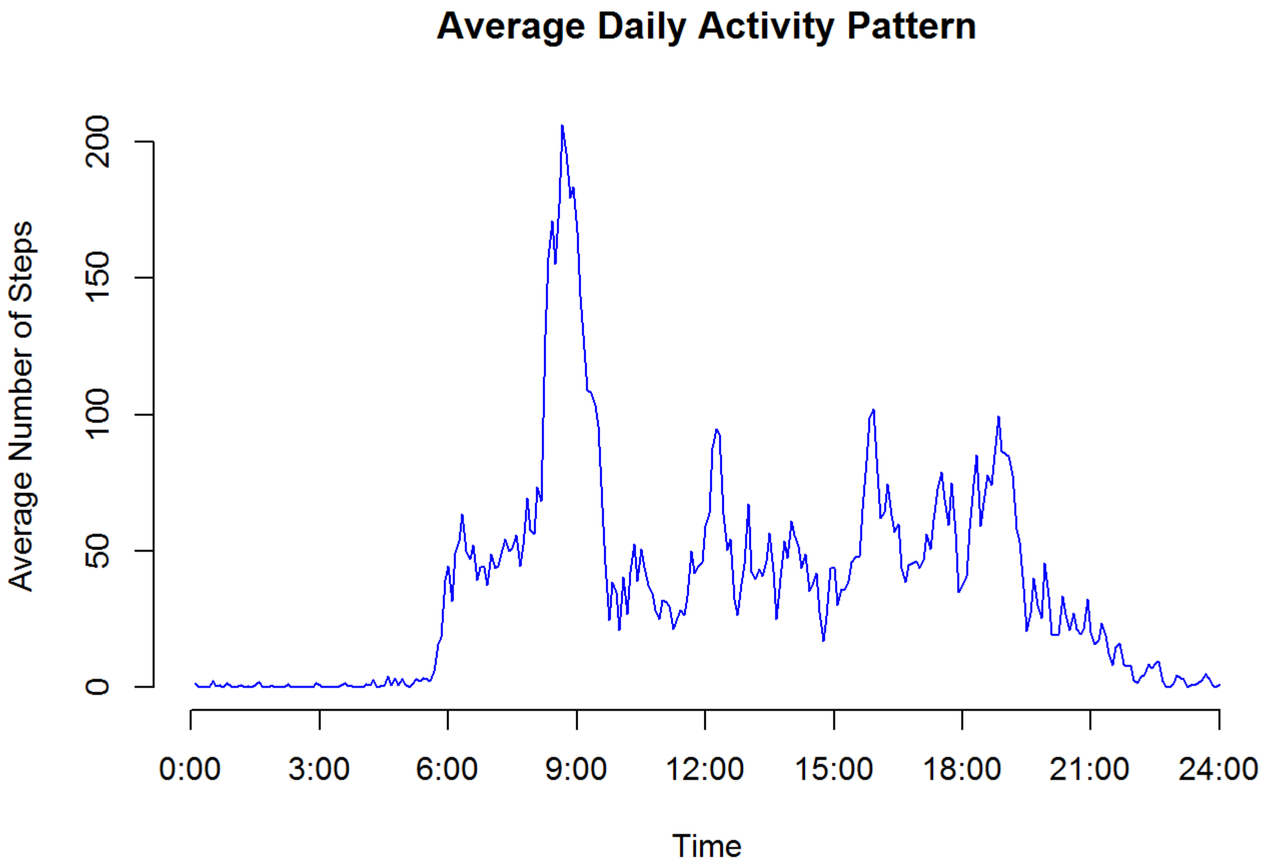
```
meanStep<-round(mean(TotalDailySteps), 2)
medianStep<-round(median(TotalDailySteps), 2)
```

So, the mean total daily steps is 9354.23 and the median is 10395 What is the average daily activity pattern? For this part of analysis, we should calculate average steps taken on each 15 minutes interval across all study period. Hence would be calculated into a new variable.

```
intervalSteps<-with(df, tapply(steps, interval, mean, na.rm=TRUE))
```

Now, we can plot the mean step value for each interval. Again, ‘base plotting system’ is more the sufficient for this step.

```
plot(intervalSteps,axes = F, type="l", col="blue", xlab="Time", ylab="Average Number of Steps"
, main="Average Daily Activity Pattern")
axis(1,at=c(0, 36, 72, 108, 144, 180, 216, 252, 288), label = c("0:00", "3:00","6:00", "9:00"
, "12:00","15:00","18:00","21:00","24:00"))
axis(2)
```



We also have been asked about the interval with maximum average steps taken within. We can calculate it as follow:

```
intervalSteps[which.max(intervalSteps)]
```

```
##      835
## 206.1698
```

```
MaxStepID<-which.max(intervalSteps)
Hour<-MaxStepID[[1]]%/%12
IntervalMinuteEnd<-(MaxStepID[[1]]%/%12)*5
IntervalMinuteStart<-IntervalMinuteEnd - 5
maxStep<-intervalSteps[MaxStepID]
```

In this cohort, the highest average steps were taken between 8:35 and 8:40 and had a maximum value of 206.1698113.

Imputing missing values To report the number of missing values.

```
NAcount<-sum(is.na(df$steps))
NAcount
```

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

In this data frame, there are 2304 rows with missing value for ‘Steps’ variable. For imputing missing values, I use the very nice “MICE” package.

```
install.packages("mice", repos = "http://cran.us.r-project.org")
```

```
## Installing package into 'C:/Users/brian/Documents/R/win-library/3.3'
## (as 'lib' is unspecified)
```

```
## Warning: unable to access index for repository http://cran.us.r-project.org/src/contrib:
##   Found continuation line starting '    <meta http-equiv ...' at begin of record.
```

```
## Warning: package 'mice' is not available (for R version 3.3.2)
```

```
## Warning: unable to access index for repository http://cran.us.r-project.org/bin/windows/co
ntrib/3.3:
##   Found continuation line starting '    <meta http-equiv ...' at begin of record.
```

```
library(mice)
```

```
## Warning: package 'mice' was built under R version 3.3.3
```

```
imputedValues<-mice(df2)
```

```
## 
##  iter imp variable
##   1   1  steps
##   1   2  steps
##   1   3  steps
##   1   4  steps
##   1   5  steps
##   2   1  steps
##   2   2  steps
##   2   3  steps
##   2   4  steps
##   2   5  steps
##   3   1  steps
##   3   2  steps
##   3   3  steps
##   3   4  steps
##   3   5  steps
##   4   1  steps
##   4   2  steps
##   4   3  steps
##    4    4   steps
##    4    5   steps
##    5    1   steps
##    5    2   steps
##    5    3   steps
##    5    4   steps
##    5    5   steps
```

We have imputed values and can reconstruct the new database:

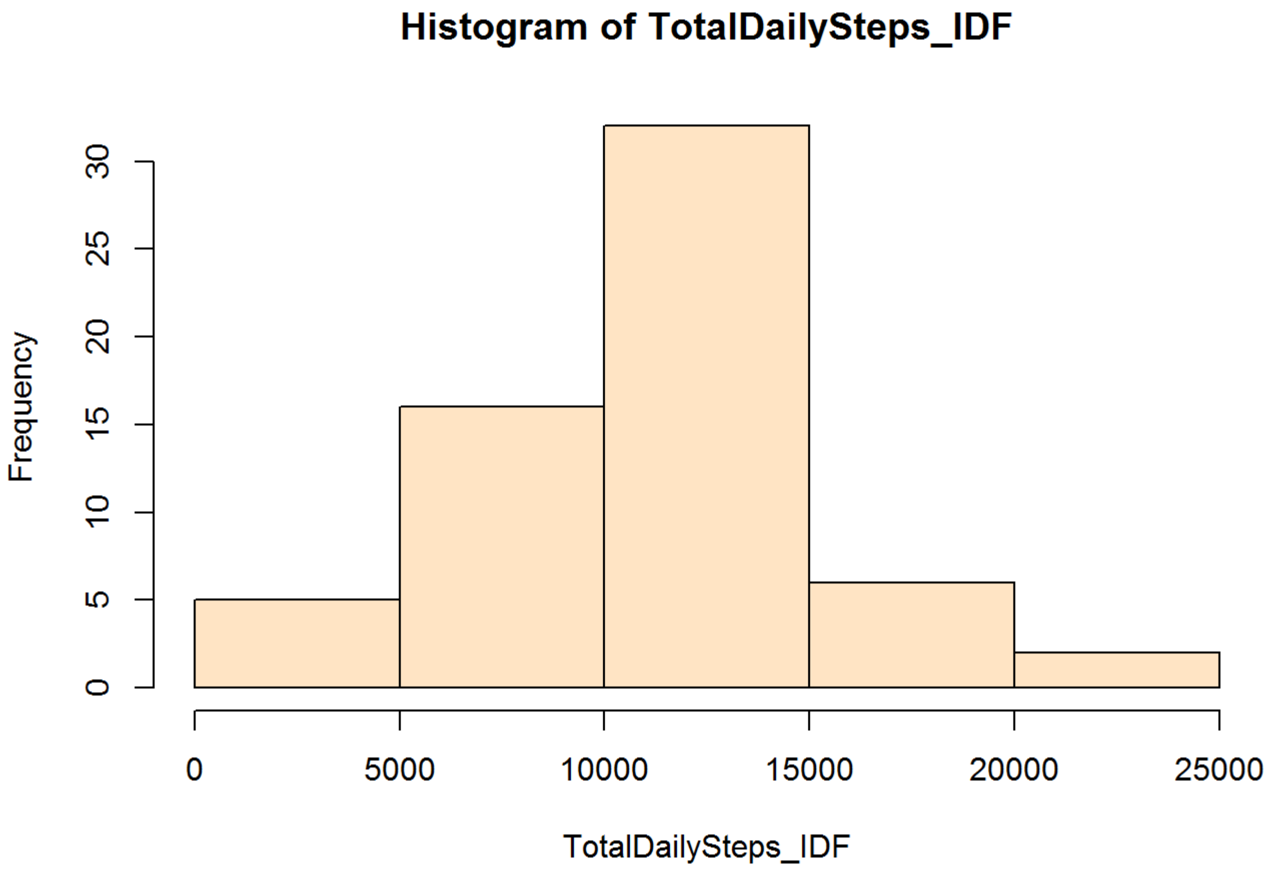
```
imputedDF<-complete(imputedValues)
```

Preprocessing of new dataframe:

```
imputedDF$date<-as.Date(imputedDF$date)
```

Now, we can use the same code as used in the first part of this assignment to produce the histogram and calculate the mean and median. 1. Calculating total daily steps: 2. Making the histograms:

```
TotalDailySteps_IDF<-with(imputedDF, tapply(steps, date, sum, na.rm=TRUE))
hist(TotalDailySteps_IDF, col = "bisque")
```



3.

Calculating the central values:

```
meanStep_IDF<-round(mean(TotalDailySteps_IDF), 2)
medianStep_IDF<-round(median(TotalDailySteps_IDF), 2)
```

After imputation of the missing values, the mean total daily steps is 1.12552610^{4} and the median is 1.127910^{4}.

Calculating the resulted change in central values:

```
meanDiff<- meanStep - meanStep_IDF
medianDiff<-medianStep - medianStep_IDF
```

After imputation, change inmean value is -1901.03 and change in median value is -884.

Are there differences in activity patterns between weekdays and weekends?

```
imputedDF$weekdays<-weekdays(imputedDF$date)
imputedDF$dayType<-ifelse(imputedDF$weekdays%in%c("Saturday", "Sunday"), "Weekend", "Weekday")
```

Now, we can calculate average steps in each interval based on type of weekday:

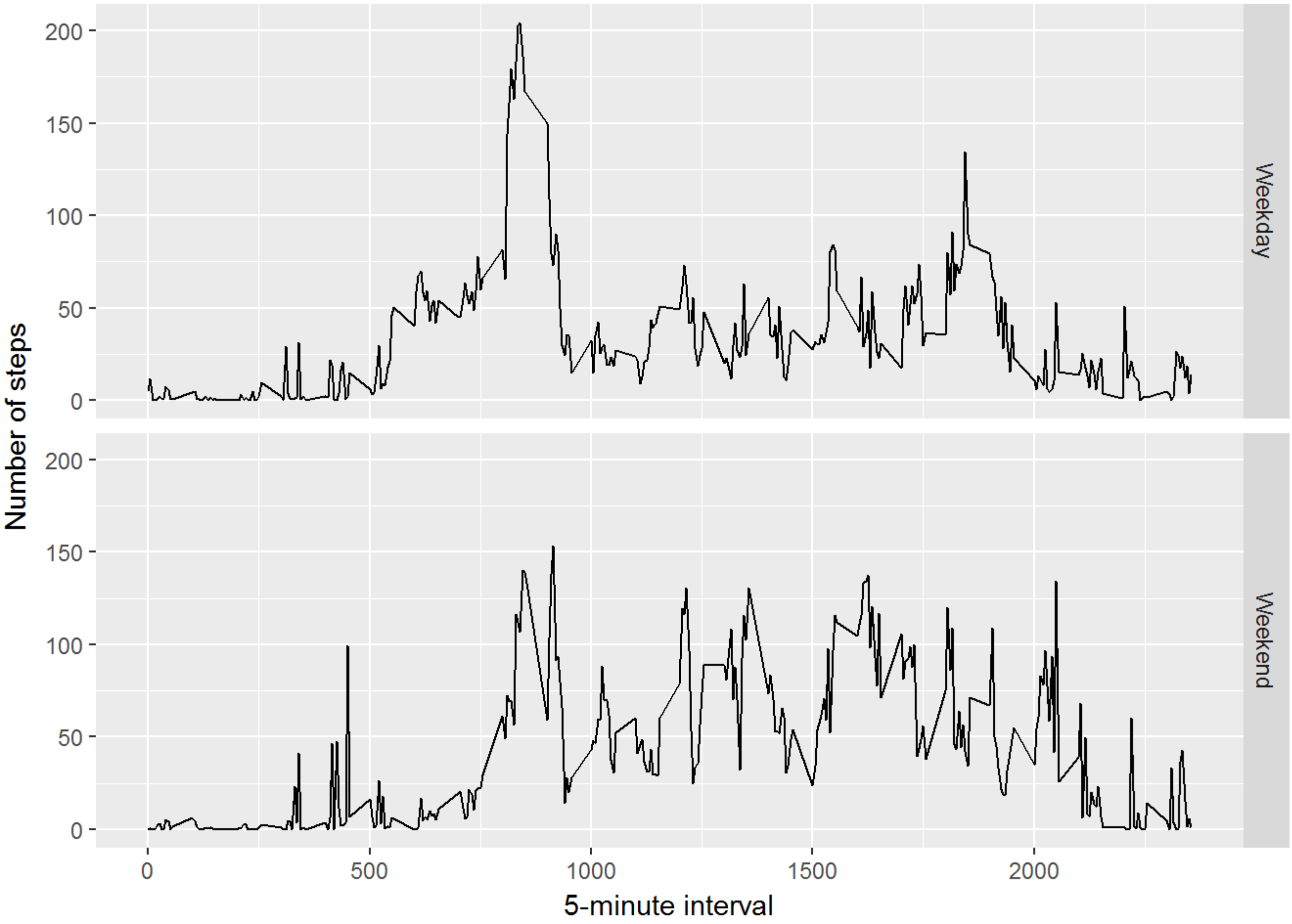
```
intervalDaySteps<-aggregate(steps~interval+dayType, data = imputedDF, mean)
```

For plotting data in two panels, I use the ggplot2 system:

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.3.3
```

```
g1<-ggplot(intervalDaySteps, aes(interval, steps))
g1 + geom_line() +
  facet_grid(dayType ~ .) +
  xlab("5-minute interval") +
  ylab("Number of steps")
```


Number of steps
200
150
100
50
0
Weekday
200
150
100
50
0
Weekend
0
500
1000
1500
2000
5-minute interval