Project 1

In this project we have carried out the analysis of data of personal activity monitoring device of an individual. The data from this device was taken from the month of October to November 2012 at an interval of every 5 minutes. The data is in the link https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip (https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip). Various steps for the analysis are as following.

Loading and pre-processing data

Part 1.

The zip file was downloaded from the link and the csv file was extracted into the R working directory. The code for loading the file into R is as follows:

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

Part 2.

In order to make the analysis on the data easier the raw data was then segregated upon the dates, using the split command. The code is as follows.

```
activity_data <- (split(raw_data$steps, raw_data$date, drop = FALSE))
```

Number of steps taken per day

Part 1.

The sum of the number of steps for various days was found out by sapply command the code is as follows.

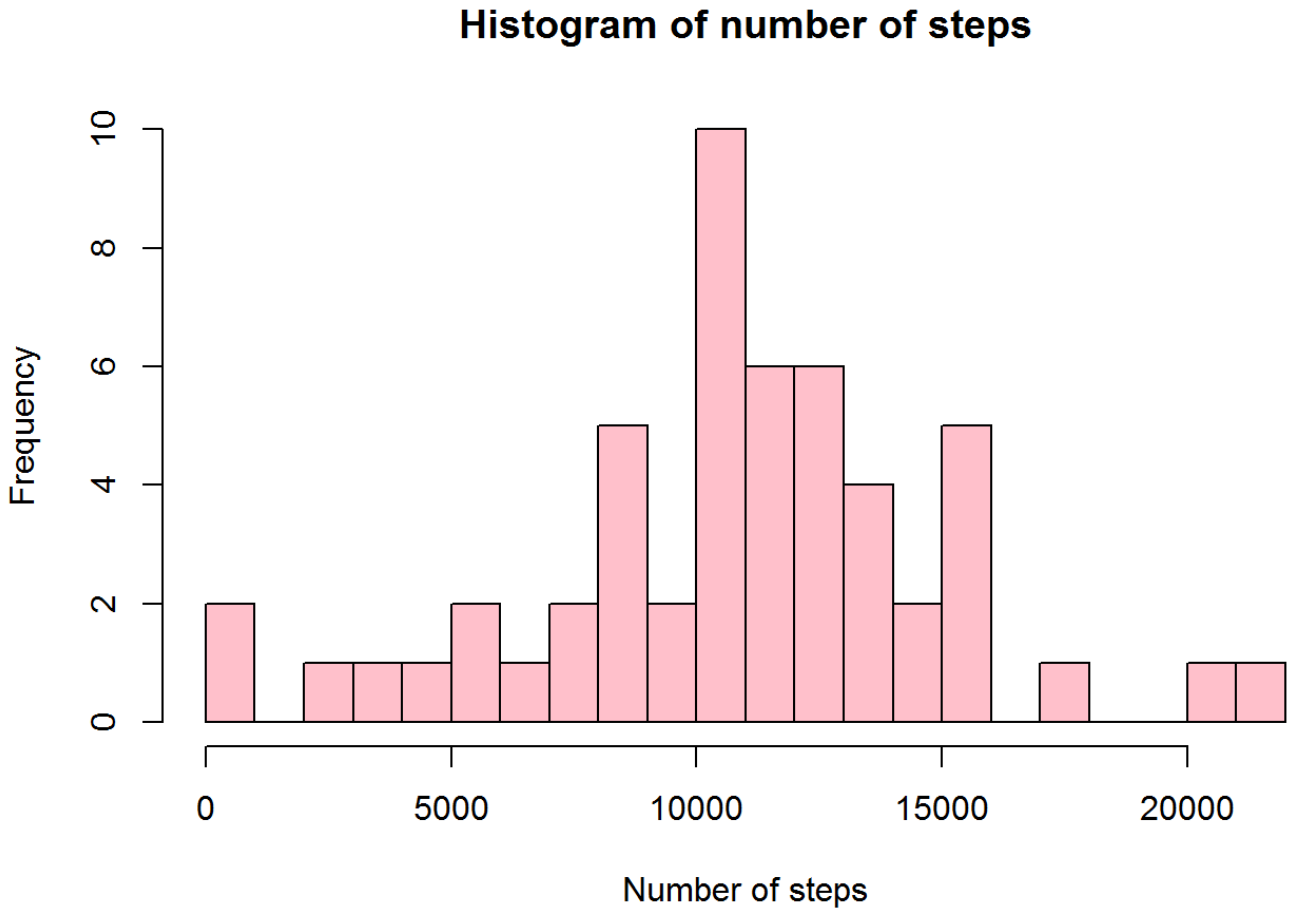
```
sapply(activity_data[1:61], sum)
```

```
## 2012-10-01 2012-10-02 2012-10-03 2012-10-04 2012-10-05 2012-10-06
##         NA        126      11352      12116      13294      15420
## 2012-10-07 2012-10-08 2012-10-09 2012-10-10 2012-10-11 2012-10-12
##      11015         NA      12811       9900      10304      17382
## 2012-10-13 2012-10-14 2012-10-15 2012-10-16 2012-10-17 2012-10-18
##      12426      15098      10139      15084      13452      10056
## 2012-10-19 2012-10-20 2012-10-21 2012-10-22 2012-10-23 2012-10-24
##      11829      10395       8821      13460       8918       8355
## 2012-10-25 2012-10-26 2012-10-27 2012-10-28 2012-10-29 2012-10-30
##       2492       6778      10119      11458       5018       9819
## 2012-10-31 2012-11-01 2012-11-02 2012-11-03 2012-11-04 2012-11-05
##      15414         NA      10600      10571         NA      10439
## 2012-11-06 2012-11-07 2012-11-08 2012-11-09 2012-11-10 2012-11-11
##       8334      12883       3219         NA         NA      12608
## 2012-11-12 2012-11-13 2012-11-14 2012-11-15 2012-11-16 2012-11-17
##      10765       7336         NA         41       5441      14339
## 2012-11-18 2012-11-19 2012-11-20 2012-11-21 2012-11-22 2012-11-23
##      15110       8841       4472      12787      20427      21194
## 2012-11-24 2012-11-25 2012-11-26 2012-11-27 2012-11-28 2012-11-29
##      14478      11834      11162      13646      10183       7047
## 2012-11-30
##         NA
```

Part 2.

The histogram for the data was then drawn using the following code.

```
hist(sapply(activity_data[1:61], sum), breaks = 20, col = "Pink", main = "Histogram of nu
mber of steps", xlab = "Number of steps")
```



Part 3.

The mean number of steps in a day was calculated using the following code,

```
mean(sapply(activity_data[1:61], sum), na.rm = TRUE)
```

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

Median was calculated by the code,

```
median(sapply(activity_data[1:61], sum), na.rm = TRUE)
```

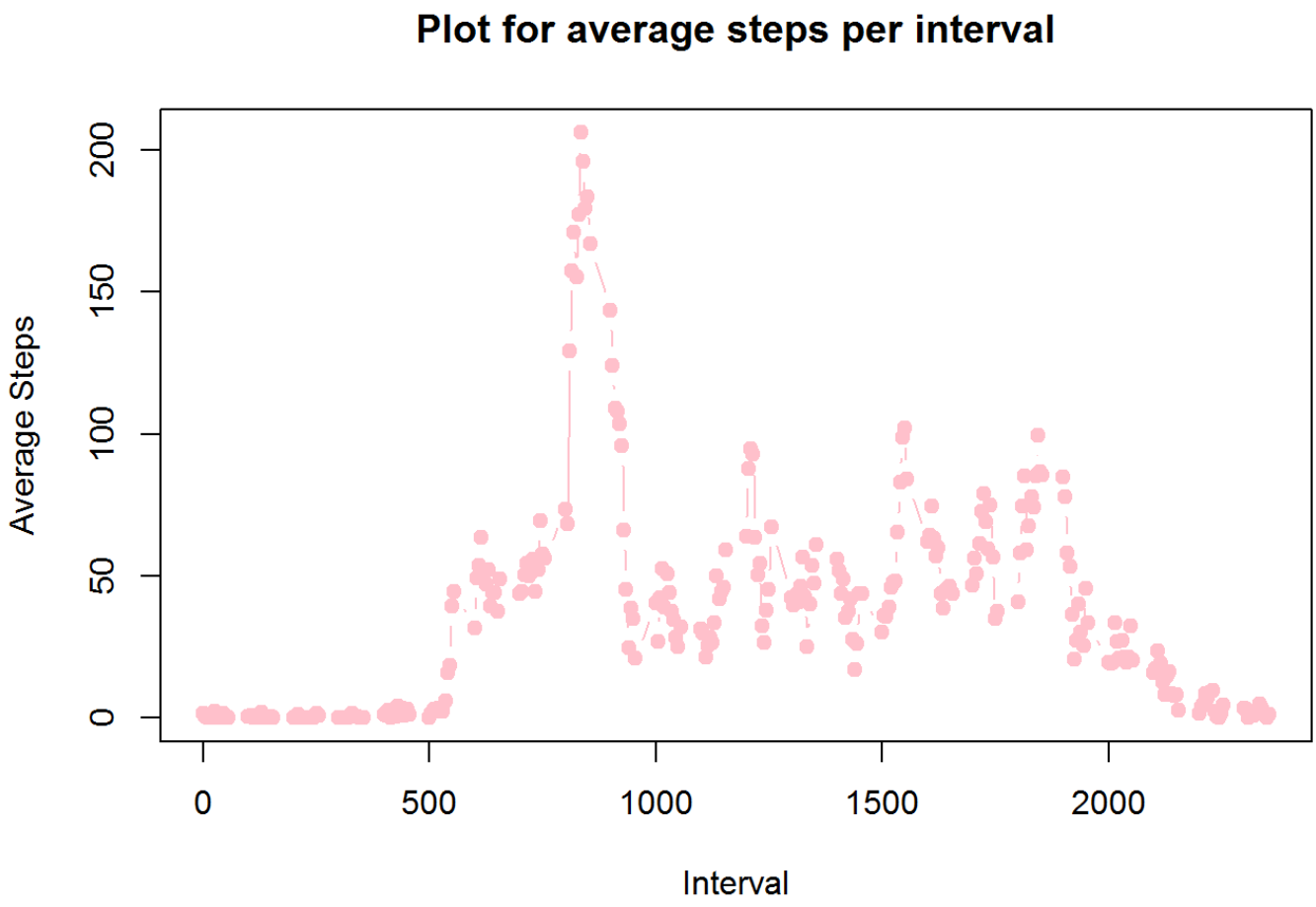
```
## [1] 10765
```

Average daily activity pattern

Part 1.

The graph for the average number of steps taken per interval was plotted using the following code

```
interval_data<- aggregate(raw_data$steps~raw_data$interval, FUN = mean)
plot(interval_data$`raw_data$interval`, interval_data$`raw_data$steps`,
     pch = 19, type = "b", col = "Pink", xlab = "Interval", ylab = "Average Steps", main
= "Plot for average steps per interval")
```



Part 2.

The 5 minutes interval in which the maximum numbers of steps were taken is given by,

```
interval_data[interval_data$`raw_data$steps`==  max(interval_data$`raw_data$steps`),]
```

```
##     raw_data$interval raw_data$steps
## 104               835       206.1698
```

Imputing missing values

Part 1.

The number of the missing values for the dataset will be given by the code,

```
nrow(raw_data[is.na(raw_data$steps),])
```

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

Part 2.

We have approximated the NA values to be equal to the mean of the number of steps taken in an interval. Therefore, NA's will be equal to,

```
mean(raw_data$steps, na.rm = TRUE)
```

```
## [1] 37.3826
```

Part 3.

Replacing the value NA with the mean steps in the original dataset, we get the new data set as,

```
new_data<- raw_data
new_data$steps[is.na(new_data$steps)] <- mean(new_data$steps, na.rm = TRUE)
head(new_data)
```

```
##     steps       date interval
## 1 37.3826 2012-10-01        0
## 2 37.3826 2012-10-01        5
## 3 37.3826 2012-10-01       10
## 4 37.3826 2012-10-01       15
## 5 37.3826 2012-10-01       20
## 6 37.3826 2012-10-01       25
```

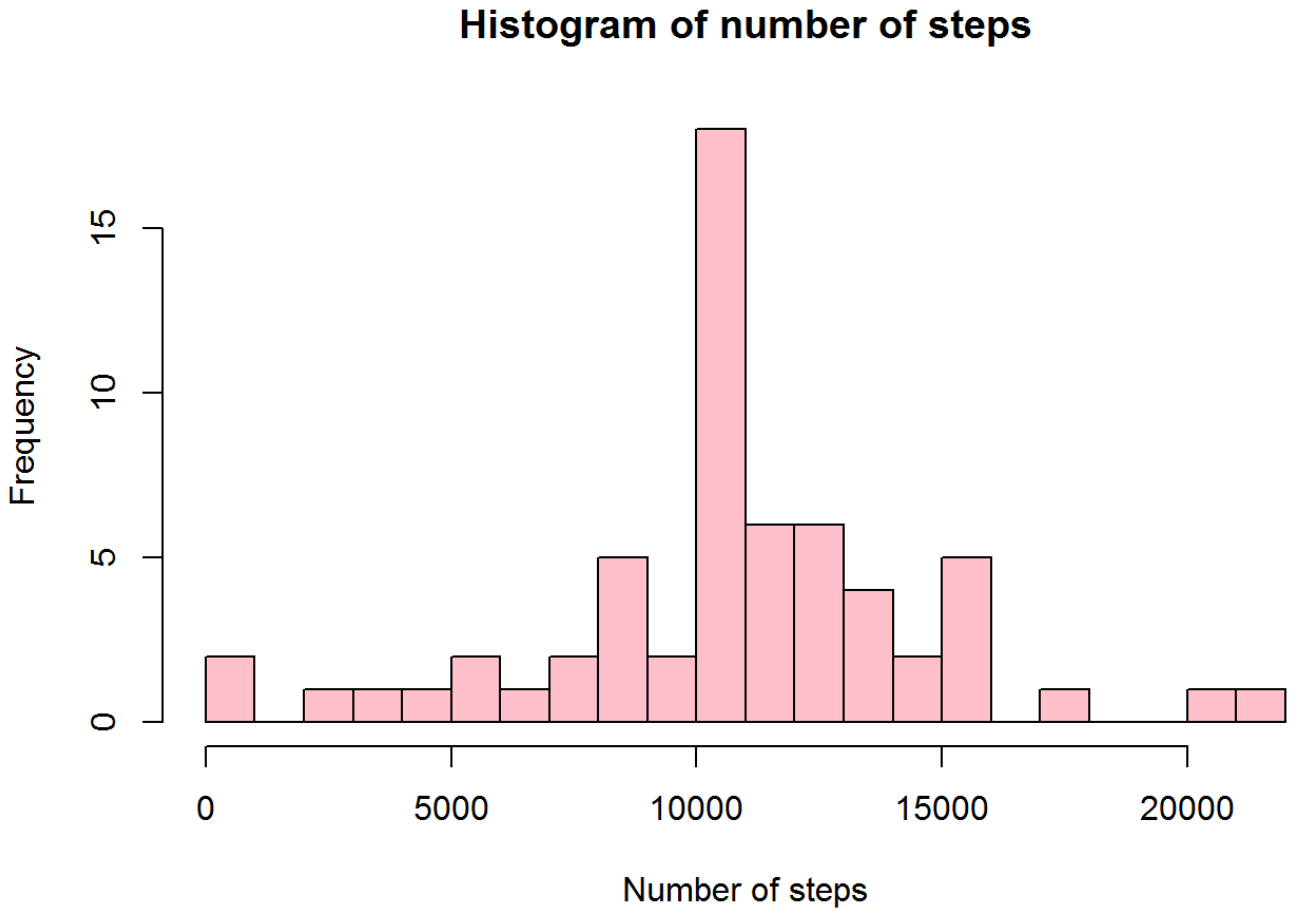
Part 4.

The histogram of the new data set is as follows,

```
new_activity_data <- (split(new_data$steps, new_data$date, drop = FALSE))
sapply(new_activity_data[1:61], sum)
```

```
## 2012-10-01 2012-10-02 2012-10-03 2012-10-04 2012-10-05 2012-10-06
##   10766.19     126.00   11352.00   12116.00   13294.00   15420.00
## 2012-10-07 2012-10-08 2012-10-09 2012-10-10 2012-10-11 2012-10-12
##   11015.00   10766.19   12811.00    9900.00   10304.00   17382.00
## 2012-10-13 2012-10-14 2012-10-15 2012-10-16 2012-10-17 2012-10-18
##   12426.00   15098.00   10139.00   15084.00   13452.00   10056.00
## 2012-10-19 2012-10-20 2012-10-21 2012-10-22 2012-10-23 2012-10-24
##   11829.00   10395.00    8821.00   13460.00    8918.00    8355.00
## 2012-10-25 2012-10-26 2012-10-27 2012-10-28 2012-10-29 2012-10-30
##    2492.00    6778.00   10119.00   11458.00    5018.00    9819.00
## 2012-10-31 2012-11-01 2012-11-02 2012-11-03 2012-11-04 2012-11-05
##   15414.00   10766.19   10600.00   10571.00   10766.19   10439.00
## 2012-11-06 2012-11-07 2012-11-08 2012-11-09 2012-11-10 2012-11-11
##    8334.00   12883.00    3219.00   10766.19   10766.19   12608.00
## 2012-11-12 2012-11-13 2012-11-14 2012-11-15 2012-11-16 2012-11-17
##   10765.00    7336.00   10766.19      41.00    5441.00   14339.00
## 2012-11-18 2012-11-19 2012-11-20 2012-11-21 2012-11-22 2012-11-23
##   15110.00    8841.00    4472.00   12787.00   20427.00   21194.00
## 2012-11-24 2012-11-25 2012-11-26 2012-11-27 2012-11-28 2012-11-29
##   14478.00   11834.00   11162.00   13646.00   10183.00    7047.00
## 2012-11-30
##   10766.19
```

```
hist(sapply(new_activity_data[1:61], sum), breaks = 20, col = "Pink", main = "Histogram o
f number of steps", xlab = "Number of steps")
```



The mean and median for the new data set will be given by,

```
mean(sapply(new_activity_data[1:61], sum), na.rm = TRUE)
```

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

```
median(sapply(new_activity_data[1:61], sum), na.rm = TRUE)
```

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

Difference between weekends and weekdays

Part 1.

Whether the day is weekday or weekend was determined by the following code.

```
daytype <- function(date) {
  if (weekdays(as.Date(date)) %in% c("Saturday", "Sunday")) {
    "weekend"
  } else {
    "weekday"
  }
}
new_data$daytype <- as.factor(sapply(new_data$date, daytype))

head(new_data)
```

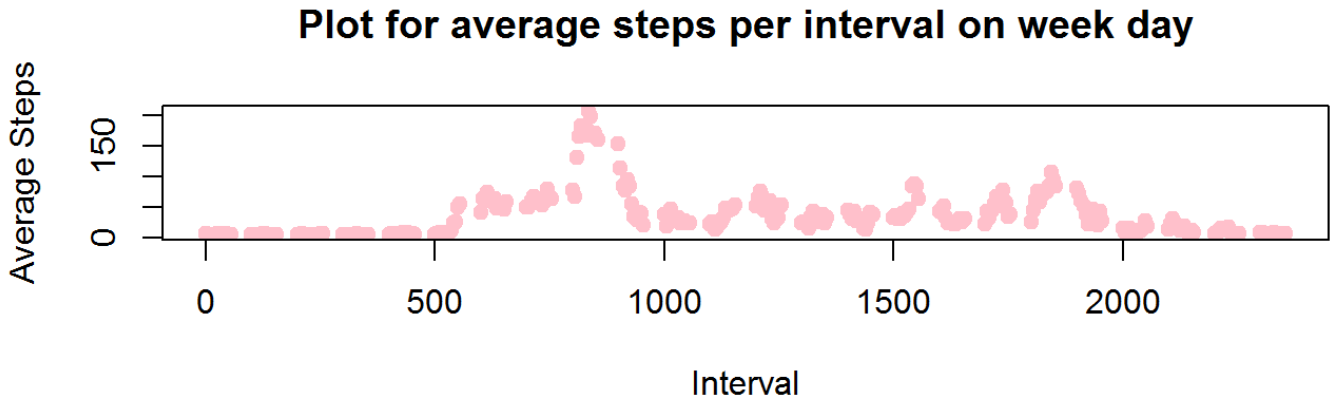
```
##     steps       date interval daytype
## 1 37.3826 2012-10-01        0 weekday
## 2 37.3826 2012-10-01        5 weekday
## 3 37.3826 2012-10-01       10 weekday
## 4 37.3826 2012-10-01       15 weekday
## 5 37.3826 2012-10-01       20 weekday
## 6 37.3826 2012-10-01       25 weekday
```

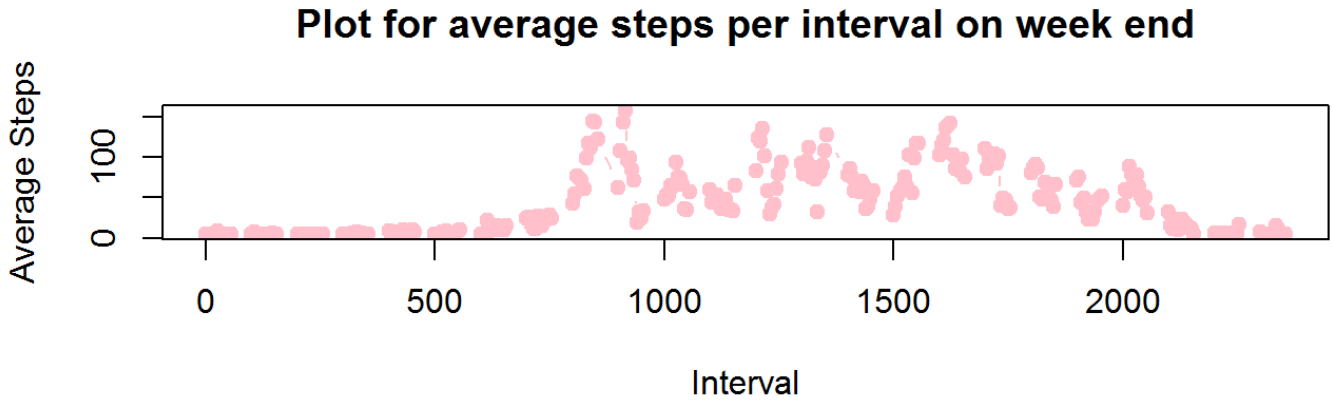
Part 2.

The plot for the number of steps taken on the weekend and weekday was drawn using the following code.

```
weekday_data <- aggregate(new_data[new_data$daytype == "weekday", ]$steps~new_data[new_da
ta$daytype == "weekday", ]$interval, FUN = mean)
weekend_data <- aggregate(new_data[new_data$daytype == "weekend", ]$steps~new_data[new_da
ta$daytype == "weekend", ]$interval, FUN = mean)

par(mfcol = c(2,1))
plot(weekday_data$`new_data[new_data$daytype == "weekday", ]$interval`, weekday_data$`ne
w_data[new_data$daytype == "weekday", ]$steps`,pch = 19, type = "b", col = "Pink", xlab =
"Interval", ylab = "Average Steps", main = "Plot for average steps per interval on week d
ay")
plot(weekend_data$`new_data[new_data$daytype == "weekend", ]$interval`, weekend_data$`ne
w_data[new_data$daytype == "weekend", ]$steps`, pch = 19, type = "b", col = "Pink", xlab
= "Interval", ylab = "Average Steps", main = "Plot for average steps per interval on week
end")
```


Plot for average steps per interval on week day
Average Steps
0
150
0
500
1000
1500
2000
Interval


Plot for average steps per interval on week end
Average Steps
0
100
0
500
1000
1500
2000
Interval