Course_Project_1

Bigyan

October 18, 2017

For the course project, the following library has been loaded to plot and make the html document.

The activity data was loaded and it was seen as shown in the Table below:

```
#setwd("J:/SDSU/Coursera/Reproducible research")

require("knitr")
opts_knit$set(root.dir = "J:/SDSU/Coursera/Reproducible research")
activity<-read.csv("activity.csv")
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
```

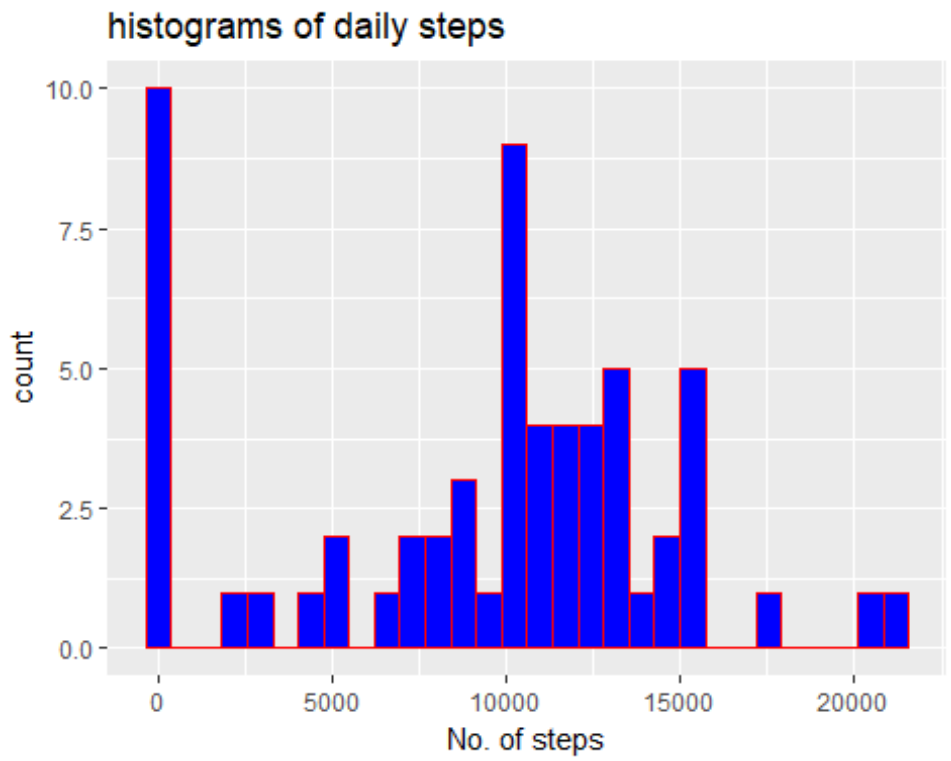
Now, to calculate the mean of the data based on the date on the use of the device from 10-01-2012 to 11-30-2012. The aggregate function was used based on activity steps with activity date. The averaged value for some of the days are listed as:

```
steps_per_date <- aggregate(x=list(steps=activity$steps),
by=list(interval=activity$date),FUN=sum, na.rm=TRUE)
head(steps_per_date)
```

```
##     interval steps
## 1 2012-10-01     0
## 2 2012-10-02   126
## 3 2012-10-03 11352
## 4 2012-10-04 12116
## 5 2012-10-05 13294
## 6 2012-10-06 15420
```

Now, the histogram has been plotted for each day using the qplot function:

```
#Histogram for each day
qplot(steps_per_date$steps,col=I("red"),fill=I("blue"), xlab = 'No. of
steps', ylab = 'count', main='histograms of daily steps')
```

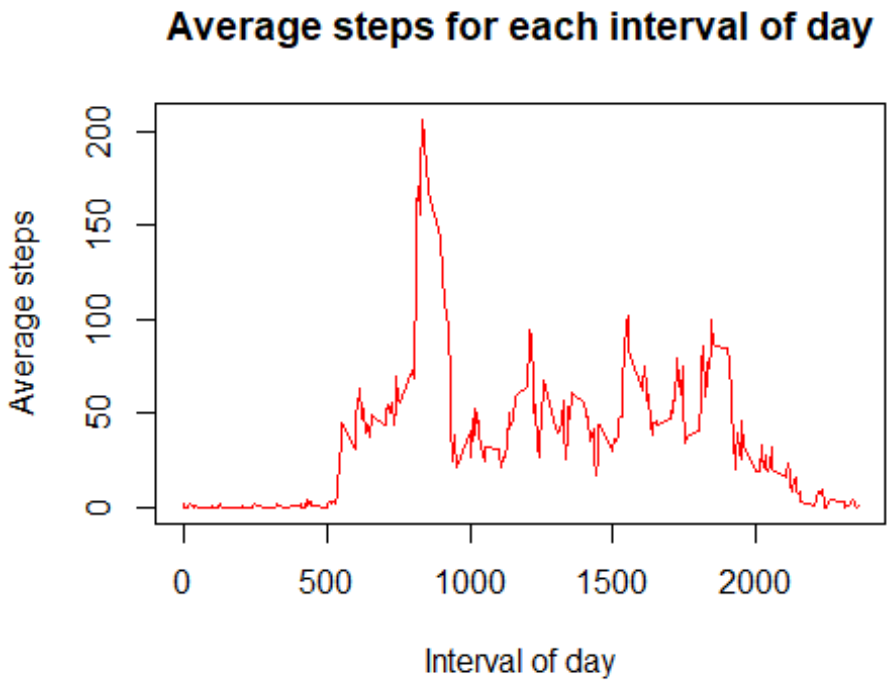


The mean and the median of the daily steps data was obtained respectively as:

```
#Mean and median value
mean(steps_per_date$steps)

## [1] 9354.23

median(steps_per_date$steps)

## [1] 10395
```

For the average daily activity pattern, using the aggregate function, the average number of steps on particular activity for all the given date was calculated. In this case, the missing values were removed. Then the time series graph was plotted as shown in the figure below:

```
average_step <- aggregate(x=list(steps=activity$steps),
by=list(interval=activity$interval),FUN=mean, na.rm=TRUE)
plot(average_step$interval,average_step$steps, type = 'l', col='red',xlab =
'Interval of day', ylab='Average steps', main= 'Average steps for each
interval of day')
```



Now, to find for which 5-minute interval, on average across all the days in the dataset, contains the maximum number of steps, we use the which.max function. The value was obtained as

```
#Finding which 5 minute interval has the maximum steps
average_step$interval[which.max(average_step$steps)]

## [1] 835
```

To calculate the total number of NA's in the data set, following code has been implemented and the value was obtained as:

```
#total number of NA's in the dataset
sum(is.na(activity$steps))

## [1] 2304
```

For the imputation of the NA values in the steps, what we have done next is that, claculate the average of that particular interval and fill with that value for each NA for that interval. The question suggests, we can also do by average of Date, however for 1st Oct, the total steps is zero so, taking average by interval would be more wise option. For the imputation, the fill value function has been created, which fill the average value in steps if there is NA, else leave as it was. Using the mapply function, this function wa scarried for all activity.

```
# Replace each missing value with the mean value of its 5-minute interval
# Now the new dataframe is fill activity
fill.value <- function(steps, interval) {
```

```
  filled <- NA
  if (!is.na(steps))
    filled <- c(steps)
  else
    filled <- (average_step[average_step$interval==interval, "steps"])
  return(filled)
}
fill_activity <- activity
fill_activity$steps <- mapply(fill.value, fill_activity$steps,
fill_activity$interval)
```

Now, the NA values are gone, which is obvious from the some data as:

```
head(fill_activity)
```

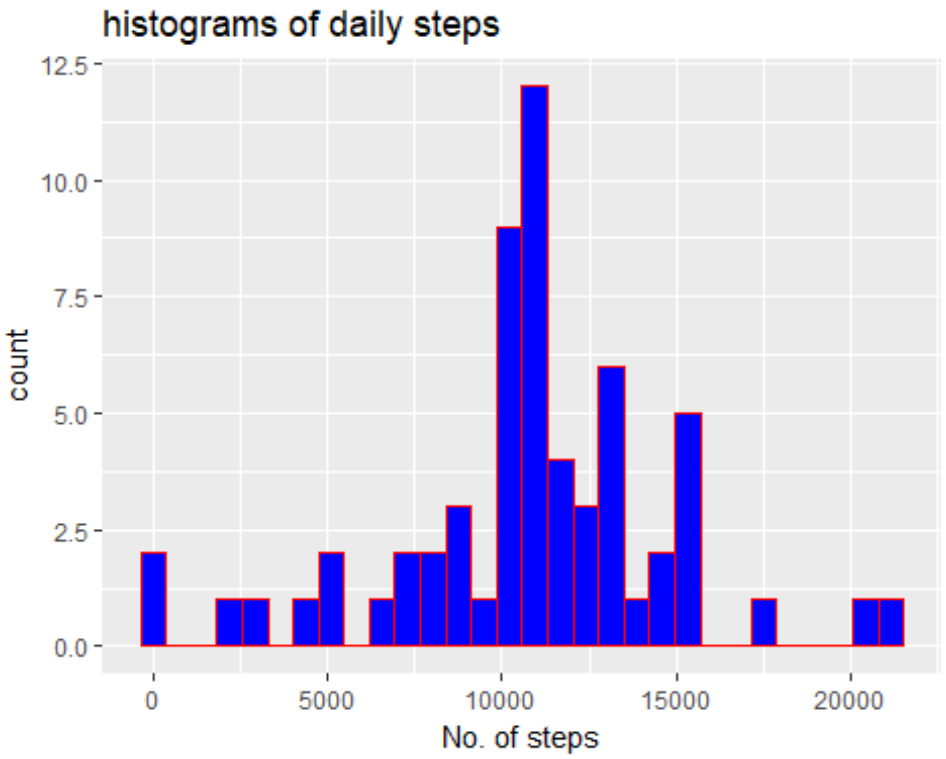
```
##       steps       date interval
## 1 1.7169811 2012-10-01        0
## 2 0.3396226 2012-10-01        5
## 3 0.1320755 2012-10-01       10
## 4 0.1509434 2012-10-01       15
## 5 0.0754717 2012-10-01       20
## 6 2.0943396 2012-10-01       25
```

The aggregate function was used based on activity steps with activity date for the filled dataset.

```
#Mean number of steps taken per day
fillsteps_per_date <- aggregate(x=list(steps=fill_activity$steps),
by=list(interval=fill_activity$date),FUN=sum, na.rm=TRUE)
```

Similarly as previous, the histogram has been plotted for each day using the qplot function for imputed data:

```
#Histogram for each day
qplot(fillsteps_per_date$steps,col=I("red"),fill=I("blue"), xlab = 'No. of
steps', ylab = 'count', main='histograms of daily steps')
```



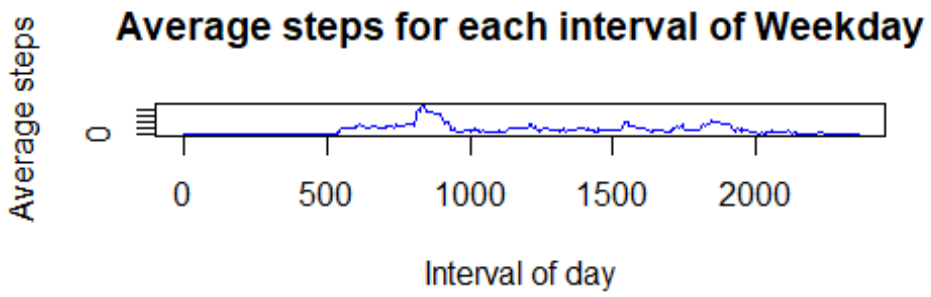
The mean and the median of the daily steps data was obtained respectively as:

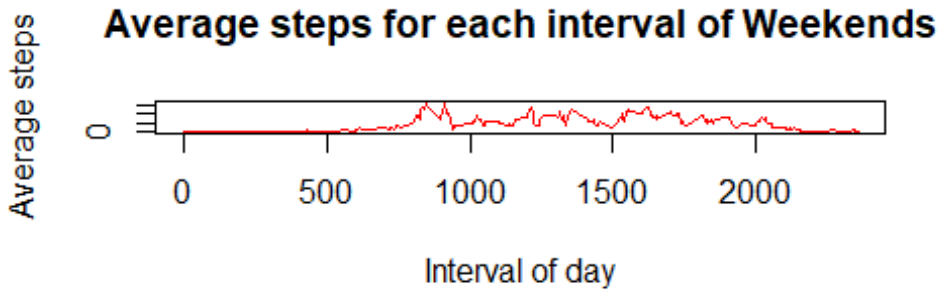
```
#Mean and median value
mean(fillsteps_per_date$steps)

## [1] 10766.19

median(fillsteps_per_date$steps)

## [1] 10766.19
```

Now, to observe if there are any differences in activity patterns between weekdays and weekends or not, the weekdays function was used to determine the days. If the days was Saturday or Sunday, the dataframe was subsetted to another dataframe for weekends and the rest was classified for weekdays. So, after the classification was done, the average for the interval was carried out using aggregate fuction as previously. So, two different plots were created for the weekends interval and weekdays interval. The plot is obtained as:

```
fill_activity$Days<- ifelse(weekdays(as.Date(fill_activity$date)) %in%
c('Saturday','Sunday'), "Weekends","Weekdays")

Weekdays_data<- subset(fill_activity, Days == "Weekdays")
Weekends_data<- subset(fill_activity, Days == "Weekends")

average_step_wd <- aggregate(x=list(steps=Weekdays_data$steps),
by=list(interval=Weekdays_data$interval), FUN=mean, na.rm=TRUE)
```

```
average_step_we <- aggregate(x=list(steps=Weekends_data$steps),
by=list(interval=Weekends_data$interval), FUN=mean, na.rm=TRUE)

par(mfrow=c(2,1))
plot(average_step_wd$interval,average_step_wd$steps,col='blue',type = 'l',
     xlab = 'Interval of day', ylab='Average steps',
     main= 'Average steps for each interval of Weekday')
plot(average_step_we$interval,average_step_we$steps,col='red', type = 'l',
     xlab = 'Interval of day', ylab='Average steps',
     main= 'Average steps for each interval of Weekends')
```





Now, looking at the above graph we can notice that there is differences between the activity in the weekends and weekdays. There is difference between average steps seen in the two graphs as there is less number of steps in weekends than weekdays.