Coursera Data Science Courses Projects

Reproducible Research - Week 2 Peer Graded Project -

Author: Mohamed Osama

Date: May 17, 2020

1- Loading and preprocessing the data

Load the data Process/transform the data (if necessary) into a format suitable for your analysis

```
#Get Your Current Working Directory to Download Data In it
getwd()
```

```
## [1] "D:/Data science/Self Projects/R/Reproducible Research/1/RepData_PeerAssessment1"
```

```
#Make Sure that the Dataset is Downloaded In your Current Working Directory
dir()
```

```
##  [1] "activity.csv"                  "activity.zip"
##  [3] "doc"                           "instructions_fig"
##  [5] "PA1_template.html"             "PA1_template.md"
##  [7] "PA1_template.Rmd"              "PA1_template.tex"
##  [9] "README.md"                     "RepData_PeerAssessment1.Rproj"
```

```
#Load Your Data To Your Environment To Work On It
df<-read.csv("activity.csv")
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
```

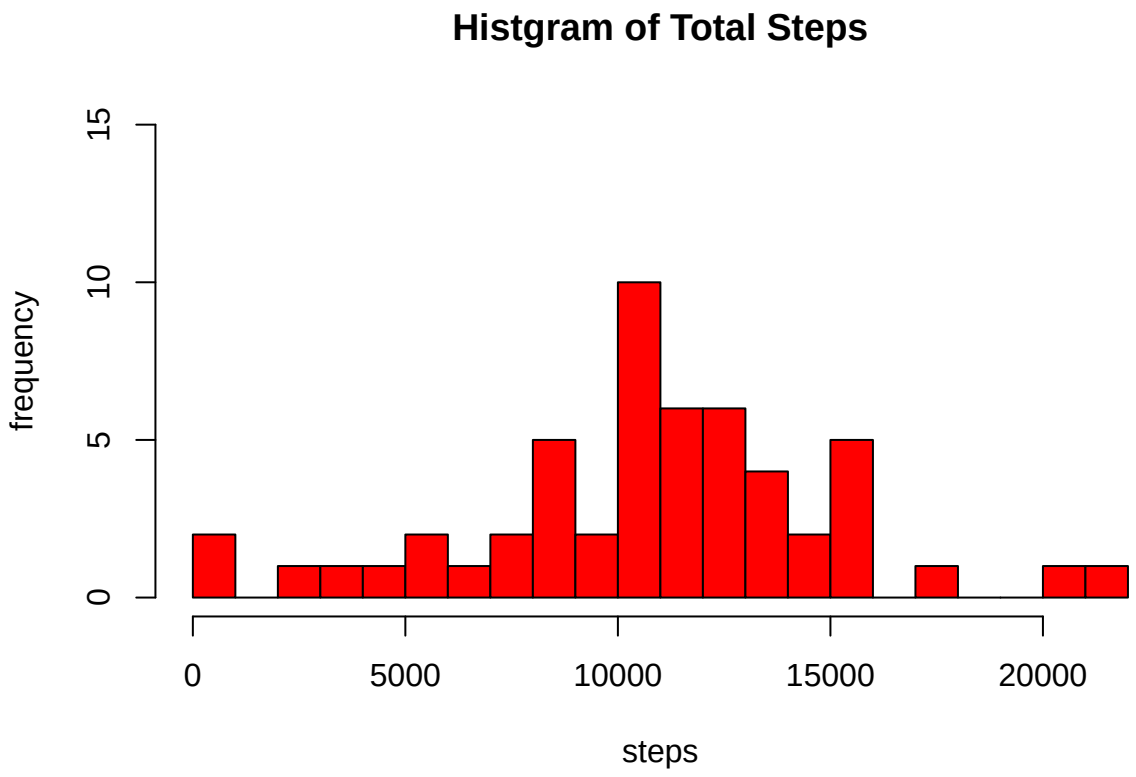
2- Calculate Total Number Of Steps Taken Each Day

Showing The Histogram Corresponding to that Calculation

```
#Calculating Total Steps For Each Day
dfused<-aggregate(steps ~ date,df,sum)
head(dfused)
```

```
##         date steps
## 1 2012-10-02   126
## 2 2012-10-03 11352
## 3 2012-10-04 12116
## 4 2012-10-05 13294
## 5 2012-10-06 15420
## 6 2012-10-07 11015
```

```r
# Plotting The Histogram
hist(dfused$steps,breaks = 25 ,xlab = "steps",ylab = "frequency", main = "Histgram of Total Steps" , yli
```



3- Calculate Mean And Median Of steps Each Day

Mean :

```r
mean(dfused$steps)
```

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

Median:

```r
median(dfused$steps)
```

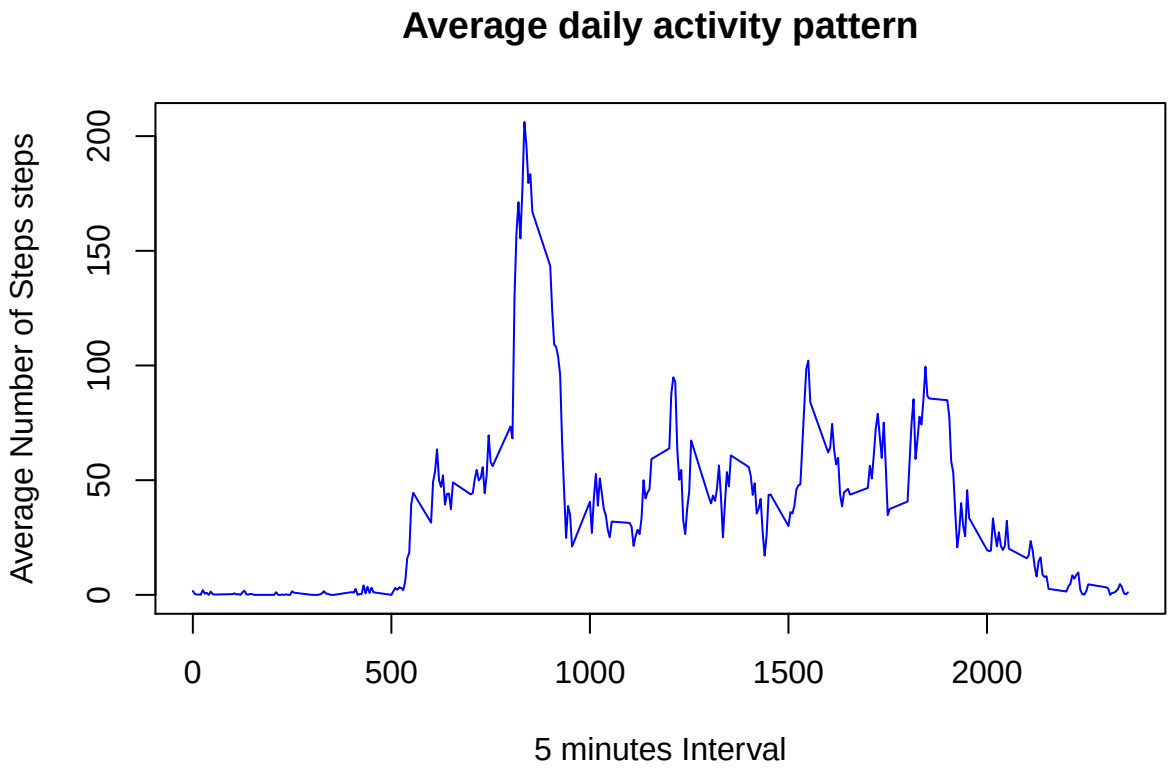
```
## [1] 10765
```

4- Time series plot of the average number of steps taken

```r
avaraged_day <- aggregate(steps~interval , df ,mean)
head(avaraged_day)
```

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

```
plot(avaraged_day$interval,avaraged_day$steps,type = "l",col="blue", xlab = "5 minutes Interval" , ylab
```



5-The(5)minute interval that on average, contains the maximum number of steps

At first, we can look at the plot of the number of steps taken averaged across all days, along all 5-min intervals

```
max_steps<-which.max(avaraged_day$steps)
max_interval <- avaraged_day[max_steps,1]
max_interval
```

```
## [1] 835
```

6-Code to describe and show a strategy for imputing missing data

- 1st I calculated Number Of Missing Values :

```
n_missing<- sum(is.na(df))
n_missing
```

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

- Then I Used The Median Function - For Steps of Each Interval To Replace The Missing Values :

- I installed The Package (“Hmisc”), Then I used It To Impute The Missing Values

```
# install.packages("Hmisc")
library(Hmisc)
```

```
## Loading required package: lattice
```

```
## Loading required package: survival
```

```
## Loading required package: Formula
```

```
## Loading required package: ggplot2
```

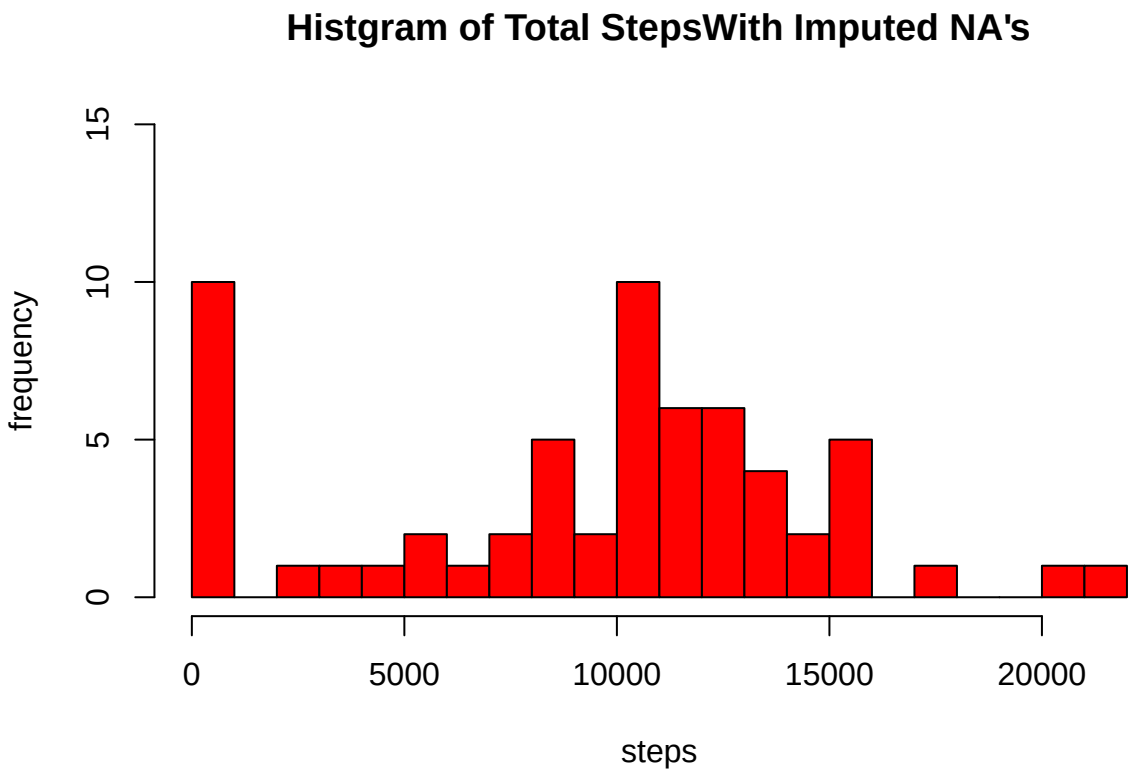
```
##
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
df_filled <- df
df_filled$steps <- impute(df$steps, fun=median)
dfImputed<-aggregate(steps ~ date,df_filled,sum)
```

7-Histogram of the total number of steps taken each day after missing values are imputed

```
hist(dfImputed$steps,breaks = 25, ylim = c(0,15) ,xlab = "steps",ylab = "frequency", main = "Histgram o
```



The effect Of Imputing Can Be shown By Comparing The Values Of Mean & Median Before & After The Imputation

Mean Of Imputed Data:

```
mean(dfImputed$steps)
```

```
## [1] 9354.23
```

Median Of Imputed Data:

```
median(dfImputed$steps)
```

```
## [1] 10395
```

Mean Of Original Data:

```
mean(dfused$steps)
```

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

Median Of Original Data:

```
median(dfused$steps)
```

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

8-Panel plot comparing the average number of steps taken per 5-minute interval across weekdays and weekends

```
df_filled$date <- as.Date(df_filled$date)
df_filled$weekday <- weekdays(df_filled$date)
df_filled$weekend <- ifelse(df_filled$weekday=="Saturday" | df_filled$weekday=="Sunday", "Weekend", "We
meandataweekendweekday <- aggregate(df_filled$steps , by= list(df_filled$weekend,df_filled$interval), na
names(meandataweekendweekday) <- c("weekend", "interval", "steps")
ggplot(meandataweekendweekday, aes(x=interval, y=steps, color=weekend)) + geom_line()+
facet_grid(weekend ~.) + xlab("Interval") + ylab("Mean of Steps") +
ggtitle("Comparison of Average Number of Steps in Each Interval")
```