Reproducible research – Project 1

Gururaj

5/29/2020

Overview

As part of this assignment, analysing the data about personal movement using activity monitoring devices such as a Fitbit, Nike Fuelband, or Jawbone Up.

Loading relevant libraries

```
library("data.table")
library(ggplot2)
```

Preparing data for analysis

```
activitydata = read.csv("activity.csv", header=TRUE, sep=",")
str(activitydata)
```

```
## '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 ...
```

Formatting activity data

```
activitydata$date = as.Date(activitydata$date, format="%Y-%m-%d")
activitydata$interval = as.factor(activitydata$interval)
str(activitydata)
```

```
## 'data.frame':    17568 obs. of  3 variables:
##  $ steps   : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ date    : Date, format: "2012-10-01" "2012-10-01" ...
##  $ interval: Factor w/ 288 levels "0","5","10","15",..: 1 2 3 4 5 6 7 8 9 10 ...
```

```
head(activitydata,10)
```

```
##    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
## 7     NA 2012-10-01       30
## 8     NA 2012-10-01       35
## 9     NA 2012-10-01       40
## 10    NA 2012-10-01       45
```

Part-1

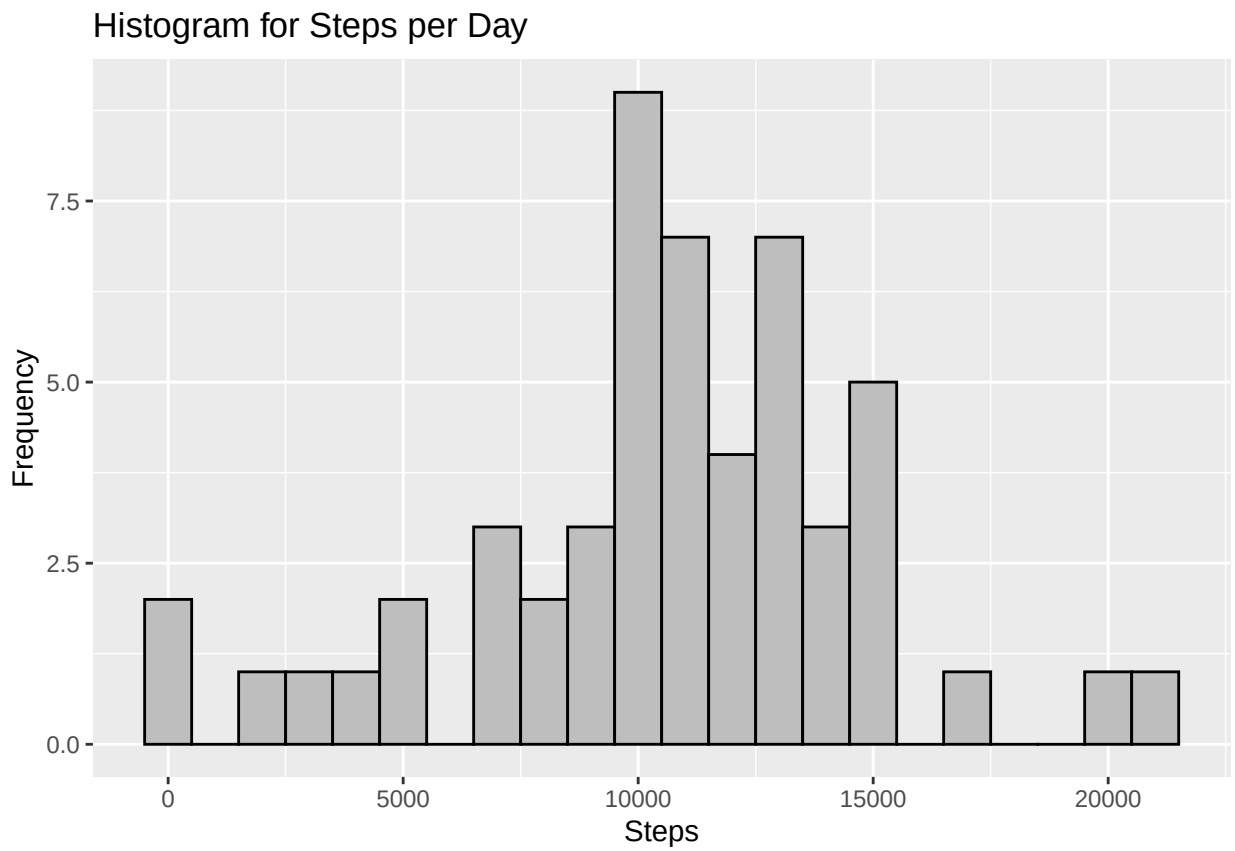
1. Total number of steps taken per day

```
totalStepsPerDay = aggregate(steps ~ date, data=activitydata, FUN=sum)
colnames(totalStepsPerDay) = c("date", "steps")
head(totalStepsPerDay,10)
```

```
##          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
## 7  2012-10-09 12811
## 8  2012-10-10  9900
## 9  2012-10-11 10304
## 10 2012-10-12 17382
```

2. Histogram of the total number of steps taken each day

```
g = ggplot(totalStepsPerDay, aes(x = steps)) +
  geom_histogram(color="black", fill="grey", binwidth = 1000) +
  labs(title = "Histogram for Steps per Day", x = "Steps", y = "Frequency")
print(g)
```



3. Mean and median for number of steps taken each day

```
meanOfStepsPerDay =  mean(totalStepsPerDay$steps)
paste("Mean of Steps per Day: " , meanOfStepsPerDay)
```

```
## [1] "Mean of Steps per Day:  10766.1886792453"
```

```
mediaOfStepsPerDay = median(totalStepsPerDay$steps)
paste("Median of Steps per Day: " , mediaOfStepsPerDay)
```

```
## [1] "Median of Steps per Day:  10765"
```

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

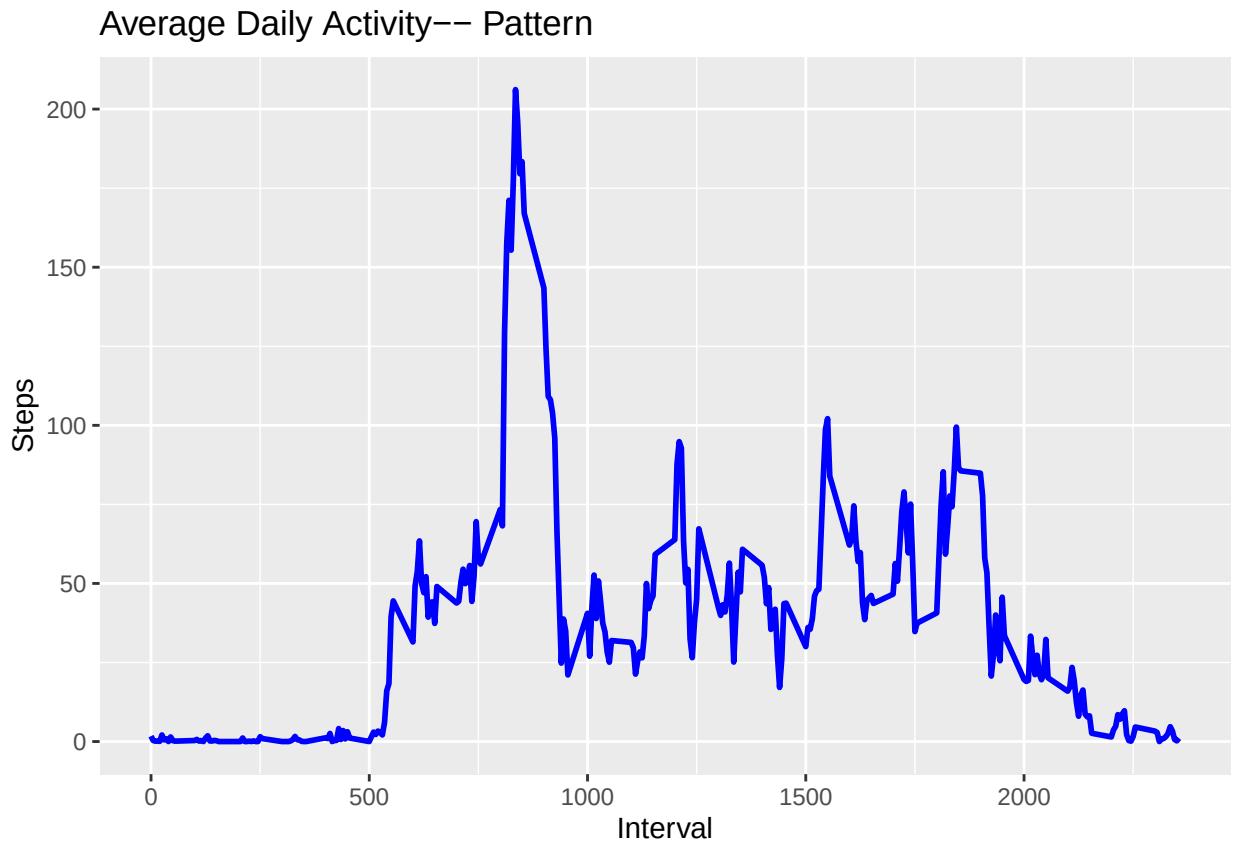
```
stepsPerInterval = aggregate(steps ~ interval, data = activitydata, FUN = mean, na.rm = TRUE)
stepsPerInterval$interval = as.integer(levels(stepsPerInterval$interval)[stepsPerInterval$interval])
colnames(stepsPerInterval) = c("interval", "steps")
head(stepsPerInterval,5)
```

```
##   interval     steps
```

```
## 1        0 1.7169811
## 2        5 0.3396226
## 3       10 0.1320755
## 4       15 0.1509434
## 5       20 0.0754717
```

```
g = ggplot(stepsPerInterval, aes(x = interval, y = steps)) +
            geom_line(col = "blue", size = 1)+
            labs(title = "Average Daily Activity-- Pattern", x = "Interval", y = "Steps")

print(g)
```



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

```
maxInterval= stepsPerInterval[which.max(stepsPerInterval$steps),]
maxInterval
```

```
##     interval    steps
## 104      835 206.1698
```

Part-2 : Imputing missing values

1. Number of missing values in the activity data set

```
missingValues = sum(is.na(activitydata$steps))
paste(" Missing values in the activity data set are: " , missingValues)
```

```
## [1] " Missing values in the activity data set are:  2304"
```

2. Strategy for filling in all of the missing values in the data set

For populating missing values, suggesting to replace missing values with the mean values at the same intervals accross days!

3. New data set by replacing missing values

```
newData = activitydata
indexOfNA = which(is.na(newData$steps))
  for (i in indexOfNA)
    {
      newData$steps[i] <- with(stepsPerInterval, steps[interval = newData$interval[i]])
    }
head(newData,10)
```

```
##        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
## 7  0.5283019 2012-10-01       30
## 8  0.8679245 2012-10-01       35
## 9  0.0000000 2012-10-01       40
## 10 1.4716981 2012-10-01       45
```

```
# Check if there are any missing values in the new data set
missingValues = sum(is.na(newData$steps))
paste(" Missing values in the new activity data set are: " , missingValues)
```

```
## [1] " Missing values in the new activity data set are:  0"
```

4. Analyse New data after replacing the missing values

Analysing the new data set by looking at the histogram, mean/median values below:

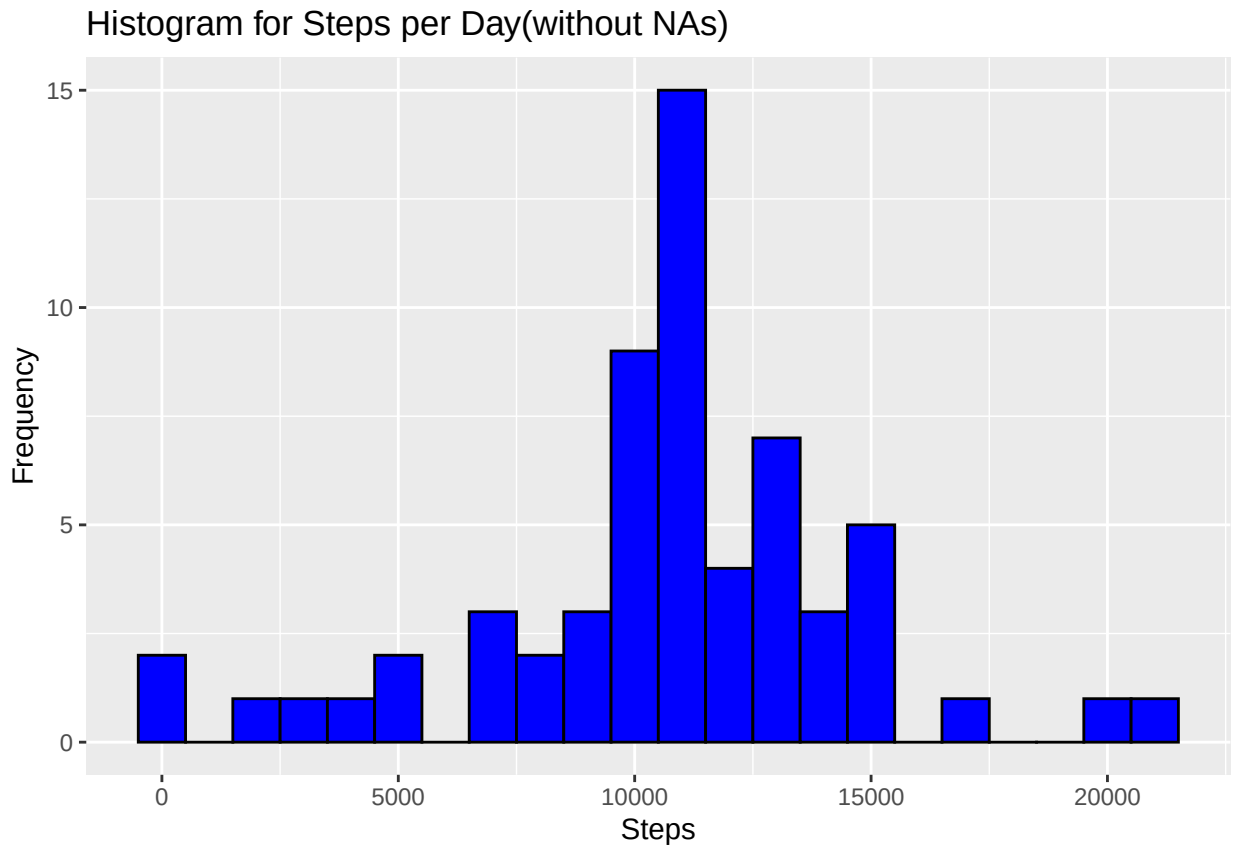
4.1. Total number of steps taken per day

```r
totalStepsPerDay_1 = aggregate(steps ~ date, data=newData, FUN=sum)
colnames(totalStepsPerDay_1) = c("date", "steps")
head(totalStepsPerDay_1,10)
```

```
##          date    steps
## 1  2012-10-01 10766.19
## 2  2012-10-02   126.00
## 3  2012-10-03 11352.00
## 4  2012-10-04 12116.00
## 5  2012-10-05 13294.00
## 6  2012-10-06 15420.00
## 7  2012-10-07 11015.00
## 8  2012-10-08 10766.19
## 9  2012-10-09 12811.00
## 10 2012-10-10  9900.00
```

4.2. Histogram of the total number of steps taken each day

```r
g1 = ggplot(totalStepsPerDay_1, aes(x = steps)) +
  geom_histogram(color="black", fill="blue", binwidth = 1000) +
  labs(title = "Histogram for Steps per Day(without NAs)", x = "Steps", y = "Frequency")
print(g1)
```



4.3. Mean and median for number of steps taken each day

```
meanOfStepsPerDay_1 =  mean(totalStepsPerDay_1$steps)
paste("Mean of Steps per Day: " , meanOfStepsPerDay_1)
```

```
## [1] "Mean of Steps per Day:  10766.1886792453"
```

```
mediaOfStepsPerDay_1 = median(totalStepsPerDay_1$steps)
paste("Median of Steps per Day: " , mediaOfStepsPerDay_1)
```

```
## [1] "Median of Steps per Day:  10766.1886792453"
```

```
## [1] "Old Mean:  10766.1886792453 Old Median:  10765 Difference is 1.1886792452824"
```

```
## [1] "New Mean:  10766.1886792453 New Median:  10766.1886792453 Difference is 0"
```

After replacing the missing values, there is no difference between mean and median values!

5. Review of activity patterns between weekdays and weekends

Adding factor variable to denote a particular day is a weekday or weekend

```
data1 = data.table(newData)
data1[, weekday := ifelse(weekdays(date) %in% c("Saturday", "Sunday"), "Weekend", "Weekday")]
data1$weekday <- as.factor(data1$weekday)
data1$interval <- as.integer(levels(data1$interval)[data1$interval])
head(data1, 5)
```

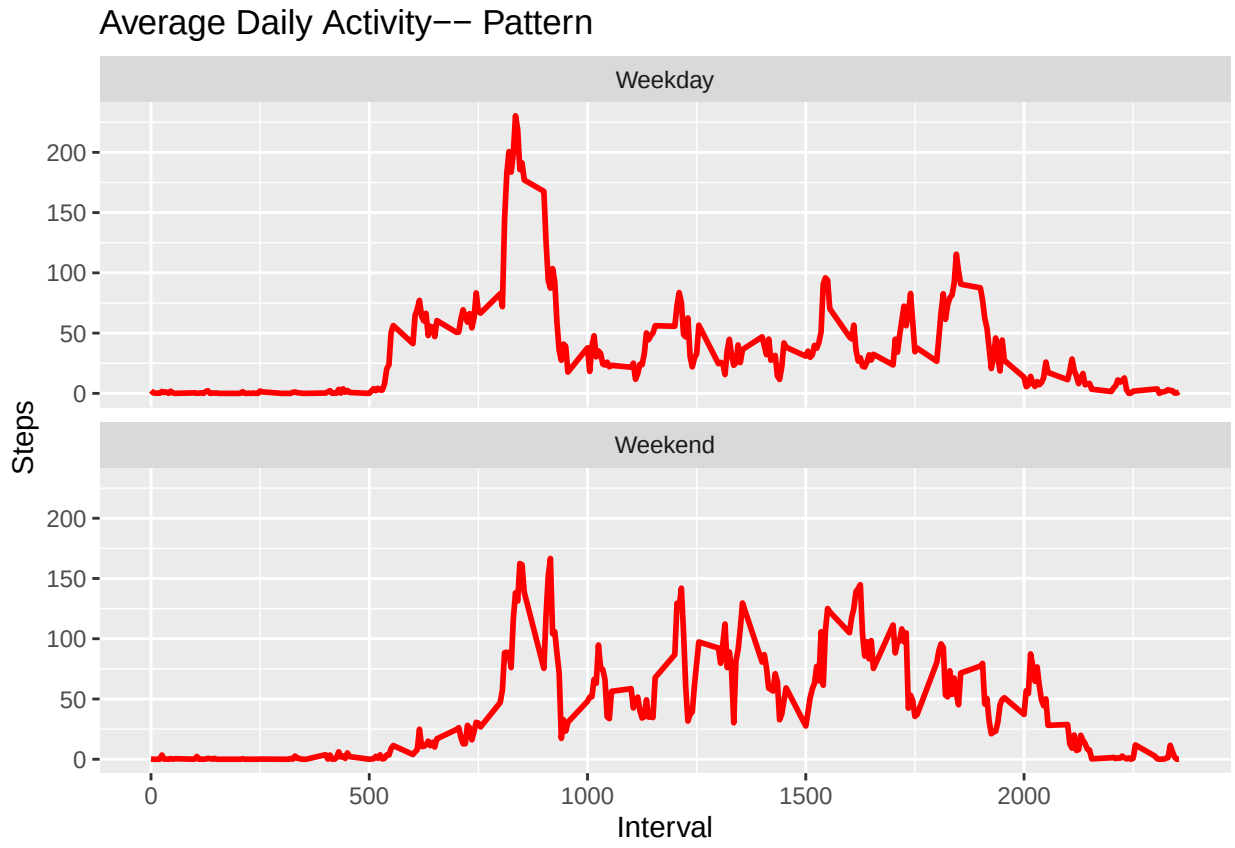
```
##        steps       date interval weekday
## 1: 1.7169811 2012-10-01        0 Weekday
## 2: 0.3396226 2012-10-01        5 Weekday
## 3: 0.1320755 2012-10-01       10 Weekday
## 4: 0.1509434 2012-10-01       15 Weekday
## 5: 0.0754717 2012-10-01       20 Weekday
```

```
tail(data1, 5)
```

```
##        steps       date interval weekday
## 1: 4.6981132 2012-11-30     2335 Weekday
## 2: 3.3018868 2012-11-30     2340 Weekday
## 3: 0.6415094 2012-11-30     2345 Weekday
## 4: 0.2264151 2012-11-30     2350 Weekday
## 5: 1.0754717 2012-11-30     2355 Weekday
```

6. Time series plot

```
stepsPerWeekday = aggregate(steps ~ interval+weekday, data = data1, FUN = mean)
ggplot(stepsPerWeekday, aes(x = interval, y = steps)) +
  geom_line(col = "red", size = 1) +
  facet_wrap(~ weekday, nrow = 2, ncol = 1) +
  labs(title = "Average Daily Activity-- Pattern", x = "Interval", y = "Steps")
```



Weekend activities have reported more number of steps than week-day activities though one of the week-day peeked to maximum number of steps.