Peer-graded Assignment-1

wassim

May 19, 2018

Introduction

It is now possible to collect a large amount of data about personal movement using activity monitoring devices such as a Fit bit, Nike Fuel band, or Jawbone Up. These type of devices are part of the "quantified self" movement - a group of enthusiasts who take measurements about themselves regularly to improve their health, to find patterns in their behavior, or because they are tech geeks. But these data remain under-utilized both because the raw data are hard to obtain and there is a lack of statistical methods and software for processing and interpreting the data.

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.

1. in this step we set the working directory where data file exist and we load the data and we get details information about data frame.

```
   setwd("C:/Users/Administrator/Desktop/coursera")
    activity <- read.csv("activity.csv", header=TRUE, sep=",")
str(activity)
```

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

```
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
```

```
tail(activity)
```

```
##       steps       date interval
## 17563    NA 2012-11-30     2330
## 17564    NA 2012-11-30     2335
## 17565    NA 2012-11-30     2340
## 17566    NA 2012-11-30     2345
## 17567    NA 2012-11-30     2350
## 17568    NA 2012-11-30     2355
```

1.1 Process/transform the data into a format suitable for analysis

```
library(lubridate)
```

```
## Warning: package 'lubridate' was built under R version 3.4.4
```

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

```
activity$date<-ymd(activity$date)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:lubridate':
##
##     intersect, setdiff, union
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

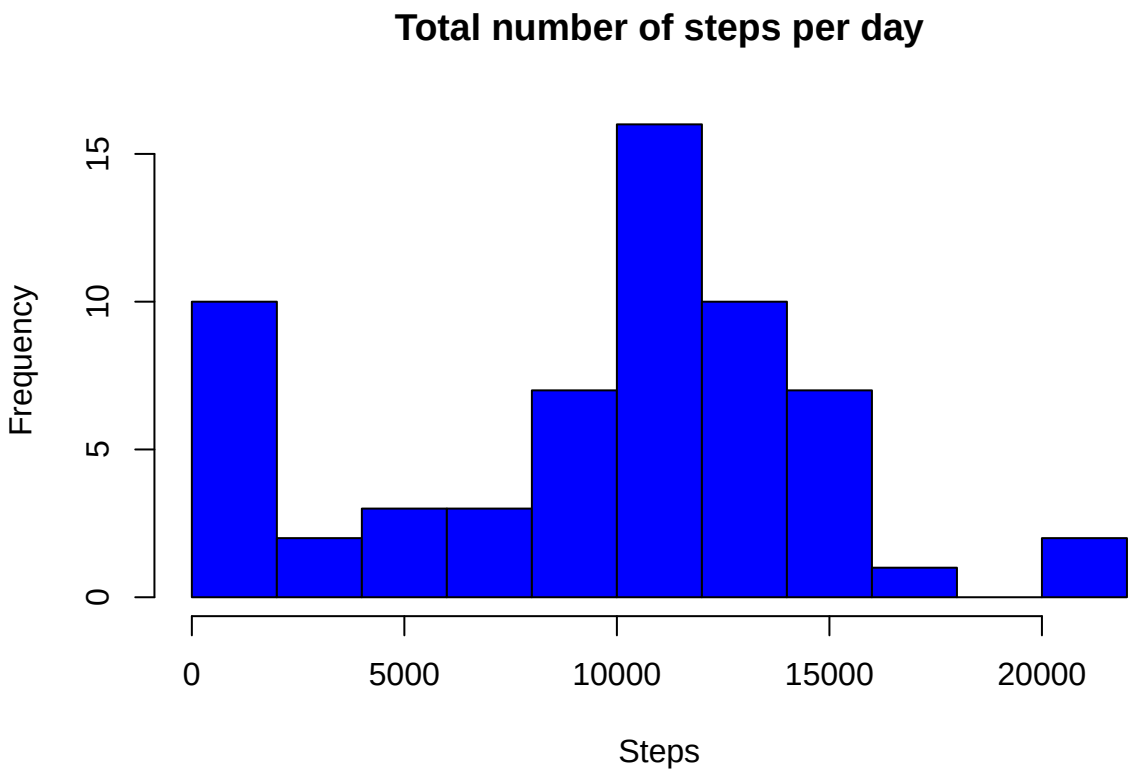
```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
groupActivity<-group_by(activity, date)
newActivity<-summarize(groupActivity, steps= sum(steps, na.rm = TRUE ))
newActivity
```

```
## # A tibble: 61 x 2
##    date       steps
##    <date>     <int>
##  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
##  7 2012-10-07 11015
##  8 2012-10-08     0
##  9 2012-10-09 12811
## 10 2012-10-10  9900
## # ... with 51 more rows
```

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

```
hist(newActivity$steps, breaks=8,main = "Total number of steps per day", xlab = "Steps", col = "blue")
```



3. Mean and median number of steps taken each day

```
mean(newActivity$steps)
```

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

```
median(newActivity$steps)
```

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

4.Time series plot of the average number of steps taken 4.1 First we fix our data set then find the average number of steps taken.

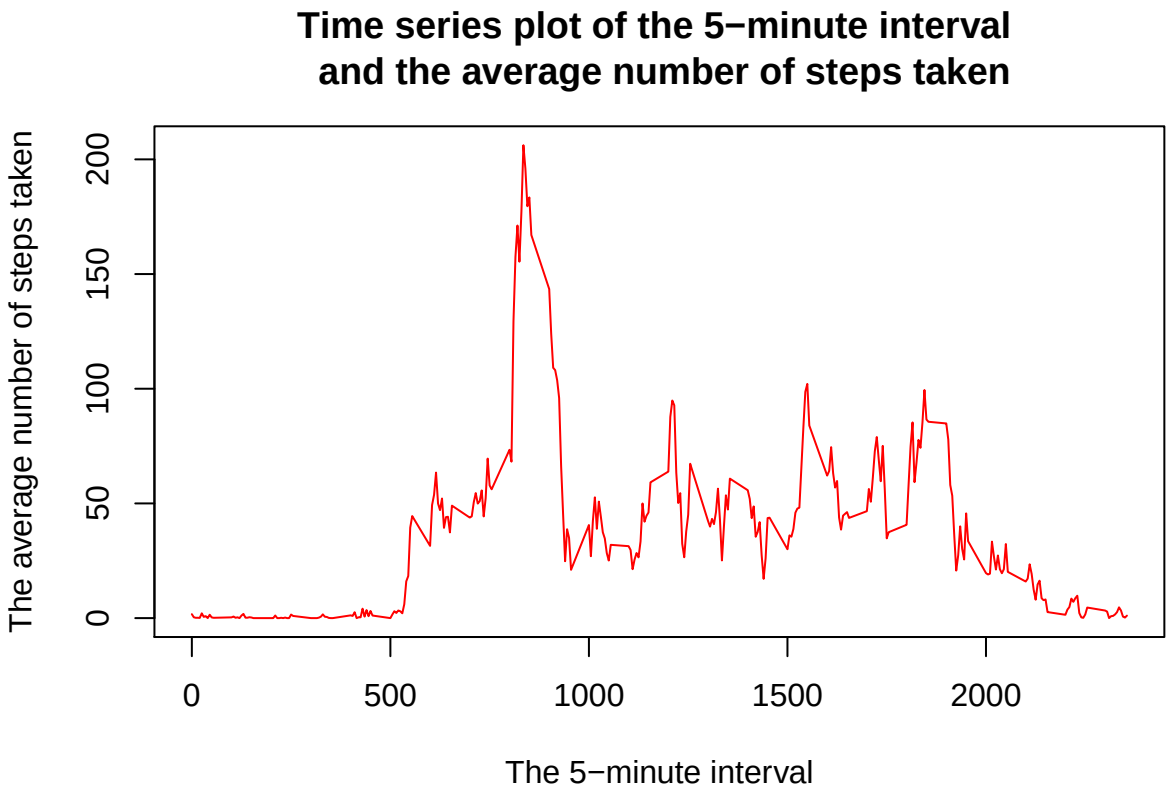
```
intervalGroupActivity <- group_by(activity, interval)
library(dplyr)
averageActivity <-summarize(intervalGroupActivity, steps= mean(steps, na.rm = TRUE ))
averageActivity
```

```
## # A tibble: 288 x 2
##    interval  steps
##       <int>  <dbl>
##  1        0 1.72
##  2        5 0.340
##  3       10 0.132
##  4       15 0.151
##  5       20 0.0755
##  6       25 2.09
##  7       30 0.528
##  8       35 0.868
```

```
##  9       40 0
## 10       45 1.47
## # ... with 278 more rows
```

4.2 plotting 5-minute interval, on average across all the days in the dataset, contains the average number of steps taken

```
plot(averageActivity$interval,averageActivity$steps, type="l", col="red",main="Time series plot of the 5
```



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

```
averageActivity[which.max(averageActivity$steps), ]$interval
```

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

6.Code to describe and show a strategy for imputing missing data. 6.1 Calculate and report the total number of rows with NAs

```
summary(activity)
```

```
##      steps             date               interval
##  Min.   :  0.00   Min.   :2012-10-01   Min.   :   0.0
##  1st Qu.:  0.00   1st Qu.:2012-10-16   1st Qu.: 588.8
##  Median :  0.00   Median :2012-10-31   Median :1177.5
##  Mean   : 37.38   Mean   :2012-10-31   Mean   :1177.5
##  3rd Qu.: 12.00   3rd Qu.:2012-11-15   3rd Qu.:1766.2
##  Max.   :806.00   Max.   :2012-11-30   Max.   :2355.0
##  NA's   :2304
```

So we have 2304 NA's

6.2 Create a new dataset that is equal to the original dataset but with the missing data filled in. using the mean for that 5-minute interval.

```
newData<-activity
for (i in 1:nrow(newData))
{
  if (is.na(newData$steps[i])){
    newData$steps[i]<-averageActivity[which(newData$interval[i] == averageActivity$interval),]$steps
  }
}
summary(newData)
```

```
##      steps             date               interval
##  Min.   :  0.00   Min.   :2012-10-01   Min.   :   0.0
##  1st Qu.:  0.00   1st Qu.:2012-10-16   1st Qu.: 588.8
##  Median :  0.00   Median :2012-10-31   Median :1177.5
##  Mean   : 37.38   Mean   :2012-10-31   Mean   :1177.5
##  3rd Qu.: 27.00   3rd Qu.:2012-11-15   3rd Qu.:1766.2
##  Max.   :806.00   Max.   :2012-11-30   Max.   :2355.0
```

```
sum(is.na(newData))
```

```
## [1] 0
```

So the result show no NA's in the new dataset.

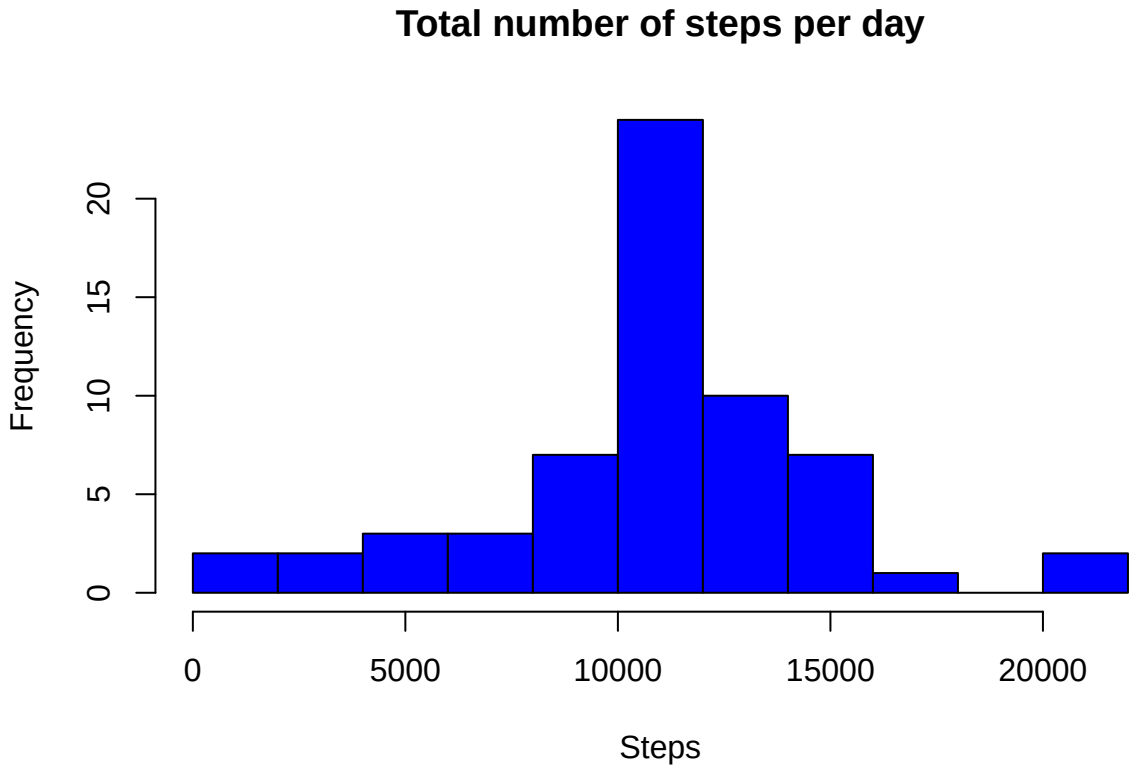
6.3 Calculate and report the mean and median total number of steps taken per day. Do these values differ from the estimates from the first part of the assignment? What is the impact of imputing missing data on the estimates of the total daily number of steps?

```
groupNewData <- group_by(newData, date)
newFillData<-summarize(groupNewData, steps= sum(steps, na.rm = TRUE ))
newFillData
```

```
## # A tibble: 61 x 2
##    date        steps
##    <date>      <dbl>
##  1 2012-10-01 10766.
##  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
##  7 2012-10-07 11015
##  8 2012-10-08 10766.
##  9 2012-10-09 12811
## 10 2012-10-10  9900
## # ... with 51 more rows
```

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

```
hist(newFillData$steps, breaks=8,main = "Total number of steps per day", xlab = "Steps", col = "blue")
```



7.1 So, after imputing the missing data, the new mean of total steps taken per day equal that of the old mean; the new median of total steps taken per day is greater than that of the old median. We can see from the Histograms the big change is that the NA's have move from the first
class.

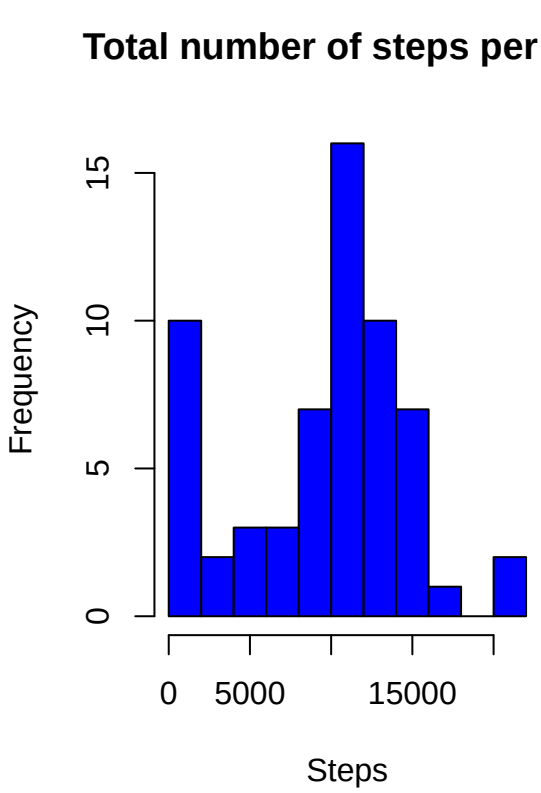
```
mean(newFillData$steps)
```

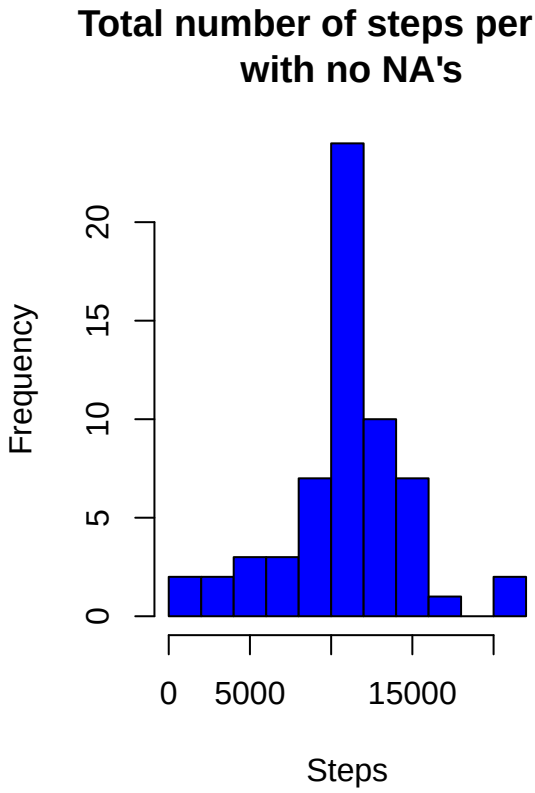
```
## [1] 10766.19
```

```
median(newFillData$steps)
```

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

```
par(mfrow=c(1,2))
hist(newActivity$steps, breaks=8,main = "Total number of steps per day", xlab = "Steps", col = "blue")
hist(newFillData$steps, breaks=8,main = "Total number of steps per day \n with no NA's", xlab = "Steps"
```





8. First we have to make our new dataset.

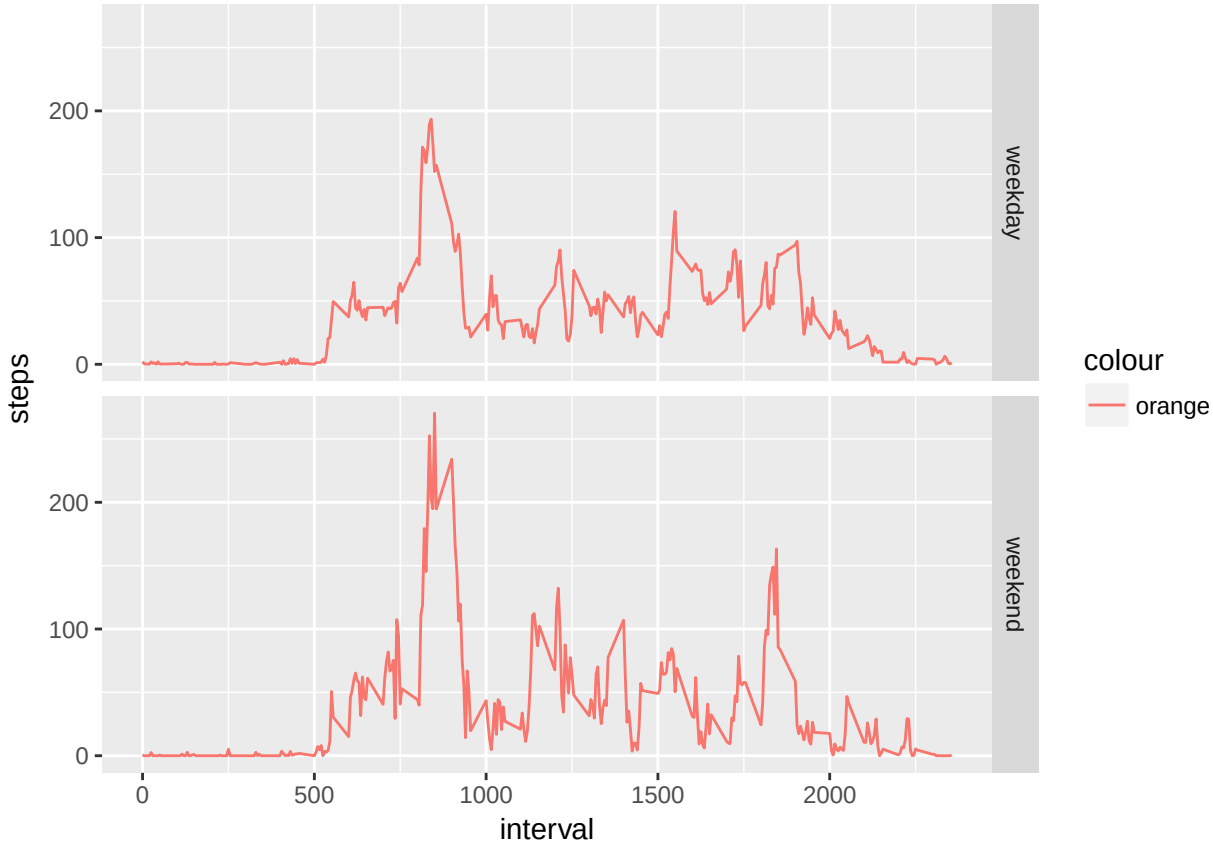
```
for (i in 1:nrow(activity))
{
  if(weekdays(activity$date[i])=="Monday" |weekdays(activity$date[i])=="Friday"){
    activity$weekdays[i]<-"weekend"
  }else{activity$weekdays[i]<-"weekday"}
}

intervalGroupActivityNA <- group_by(activity, interval,weekdays)
dataNA<-summarize(intervalGroupActivityNA, steps= mean(steps, na.rm = TRUE ))
```

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

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.4.4
```

```
qplot(interval,steps, data = dataNA,facets = weekdays~., geom="line",col="orange")
```


steps
200
100
0
weekday
200
100
0
weekend
0
500
1000
1500
2000
interval
colour
orange