instructions

mt

Saturday, September 19, 2015

Introduction

Activity recognition consists of broad range of application in security systems, gaming platforms, health care systems and etc. The first step in the case of working with evaluated datasets is changing raw data into prcessed data. Removing Not Assigned values and changing the name of columns in this dataset is the prerequisite of processed 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.

Loading and preprocessing the data

the dataset could download from here:

https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip

1- Load the data (i.e. read.csv())

```
setwd(file.path("D:", "sbu", "RLearning", "reproducible research","programming1"))
activity <- read.csv("activity.csv")
head(activity,4)
```

```
##   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
```

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

```
Ractivity <-tbl_df(activity)
rm(activity)
```

What is mean total number of steps taken per day?

For this part of the assignment, you can ignore the missing values in the dataset.

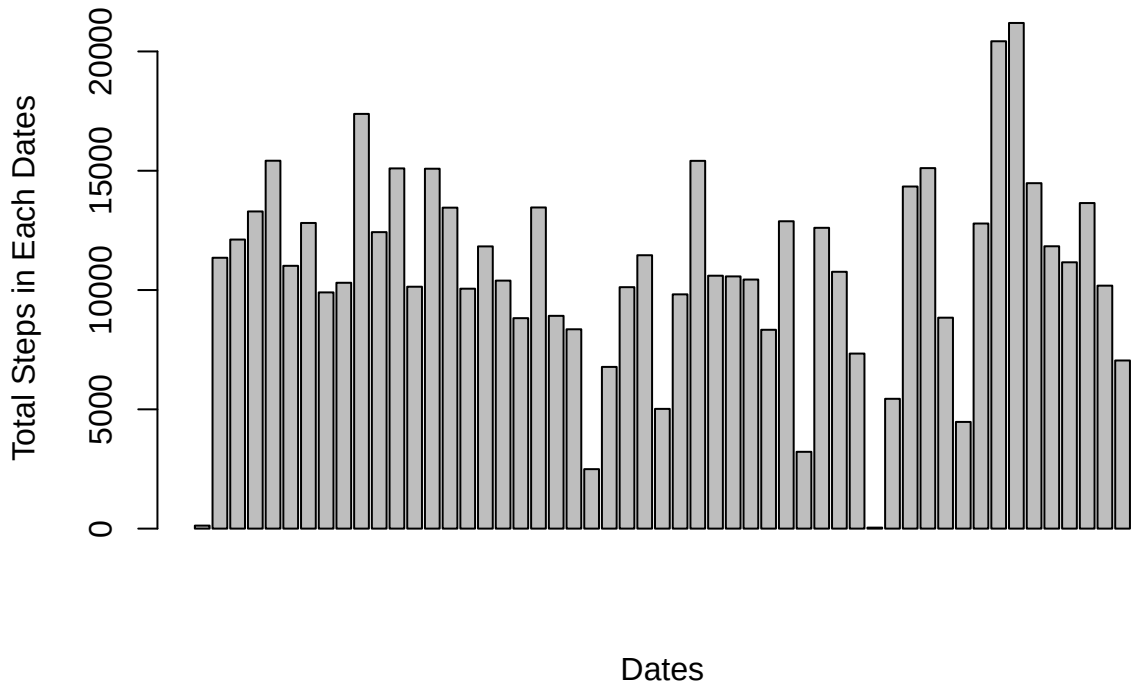
1- Calculate the total number of steps taken per day

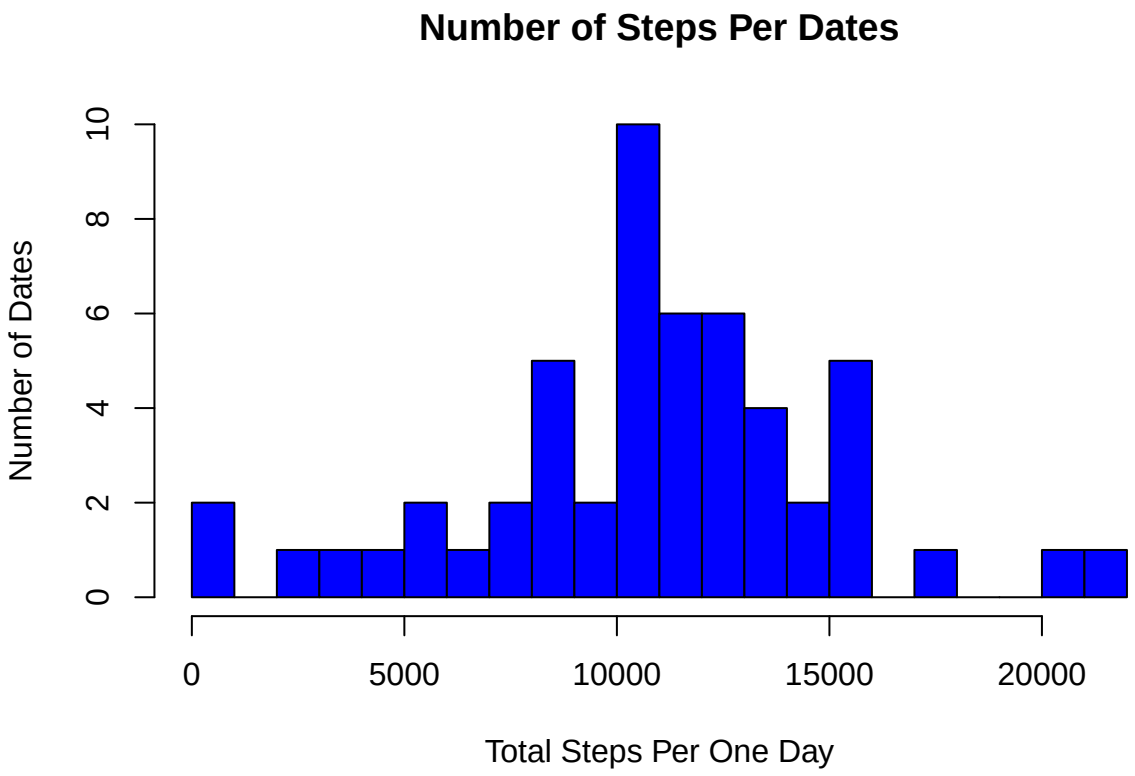
```
by_date<-group_by(Ractivity[!is.na(Ractivity$steps),],date)
total_steps<-summarize(by_date, TotalStepsPerDay=sum(steps))
head(total_steps,7)
```

```
## Source: local data frame [7 x 2]
##
##         date TotalStepsPerDay
## 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
```

2- If you do not understand the difference between a histogram and a barplot, research the difference between them.

Make a histogram of the total number of steps taken each day.





- In BarPlot each dates has its own frequency where Histogram shows number of dates with close frequency.

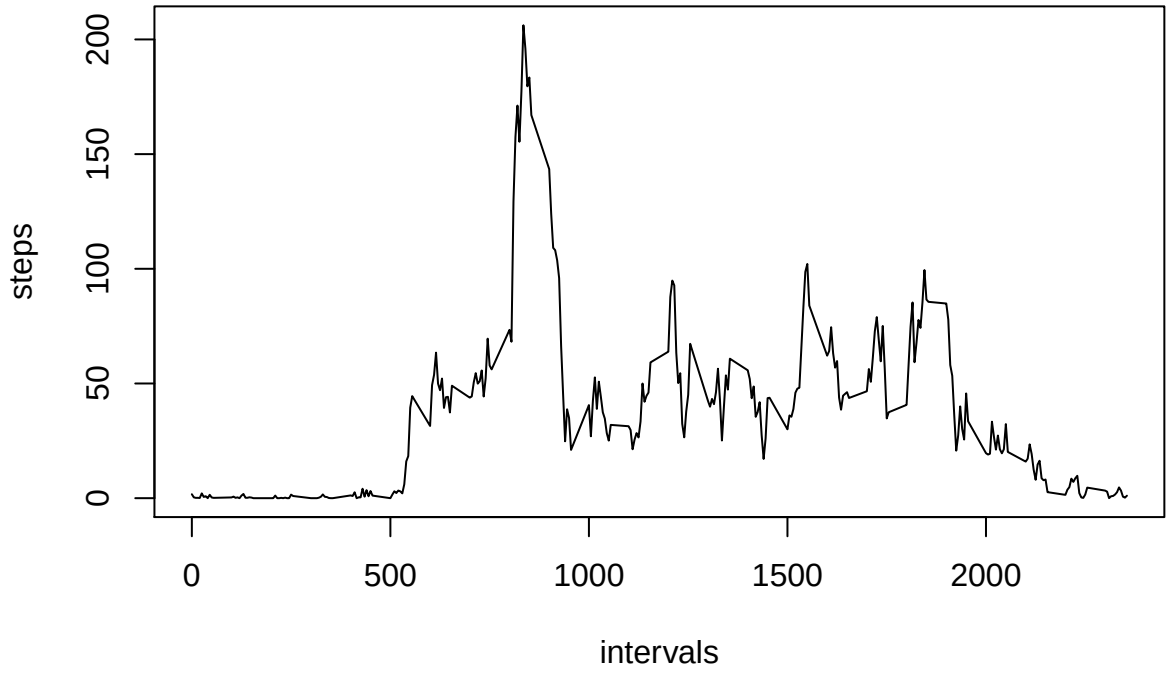
3- Calculate and Report the mean and median of the total number of steps taken per day

```
MeanMedianSteps<-summarize(by_date,MeanStepsPerDay=mean(steps),
                           MedianStepsPerDay=median(steps))
head(MeanMedianSteps,6)
```

```
## Source: local data frame [6 x 3]
##
##         date MeanStepsPerDay MedianStepsPerDay
## 1 2012-10-02         0.43750                 0
## 2 2012-10-03        39.41667                 0
## 3 2012-10-04        42.06944                 0
## 4 2012-10-05        46.15972                 0
## 5 2012-10-06        53.54167                 0
## 6 2012-10-07        38.24653                 0
```

What is the average daily activity pattern?

1- Make a time series plot (i.e. type = "l") of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all days (y-axis)



2- Which 5-minute interval, on average across all the days in the dataset, contains the maximum number of steps?

```
intervalsteps[max(intervalsteps$AverageSteps)==intervalsteps$AverageSteps,]
```

```
## Source: local data frame [1 x 2]
##
##   interval AverageSteps
## 1      835     206.1698
```

Imputing missing values

Note that there are a number of days/intervals where there are missing values (coded as NA). The presence of missing days may introduce bias into some calculations or summaries of the data.

1- Calculate and report the total number of missing values in the dataset (i.e. the total number of rows with NAs)

```
sum(is.na(Ractivity))
```

[1] 2304

2- Devise a strategy for filling in all of the missing values in the dataset.

The strategy does not need to be sophisticated. For example, you could use the mean/median for that day, or the mean for that 5-minute interval, etc.

In this question first of all Imputing takes by mean values base on Date Steps then on 5 min Intervals. Another imputing strategy is taking median of steps values first base on Dates and then by Intervals.

Imputing by Mean

```
head(menactivity,10)
```

```
## Source: local data frame [10 x 3]
##
##        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
```

Imputing by median

```
head(medactivity,10)
```

```
## Source: local data frame [10 x 3]
##
##    steps       date interval
## 1      0 2012-10-01        0
## 2      0 2012-10-01        5
## 3      0 2012-10-01       10
## 4      0 2012-10-01       15
## 5      0 2012-10-01       20
## 6      0 2012-10-01       25
## 7      0 2012-10-01       30
## 8      0 2012-10-01       35
## 9      0 2012-10-01       40
## 10     0 2012-10-01       45
```

3- Create a new dataset that is equal to the original dataset but with the missing data filled in.

```
write.table(menactivity,file="menactivity.csv", row.name=F,col.names=T)
# Data Base on mean controlling Strategy
write.table(medactivity,file="medactivity.csv", row.name=F,col.names=T)
# Data Base on median controlling Strategy
```

4- Make a histogram of the total number of steps taken each day and Calculate and report the mean and median total number of steps taken per day.

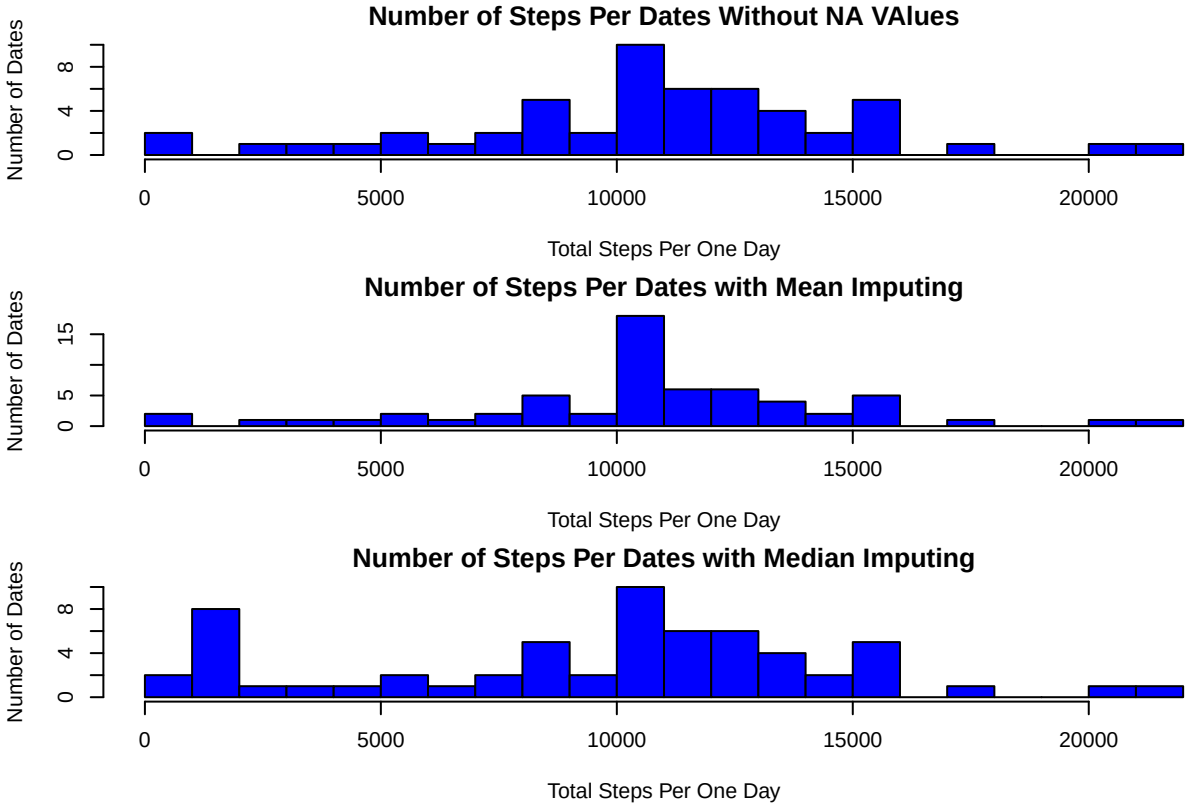
- Survey By Mean

```
mean_by_date<-group_by(menactivity,date)
mean_total_steps<-summarize(mean_by_date, TotalStepsPerDay=sum(steps))
mean_by_interval<-group_by(menactivity[!is.na(menactivity$steps),],interval)
mean_intervalsteps<-summarize(mean_by_interval,AverageSteps=mean(steps))
```

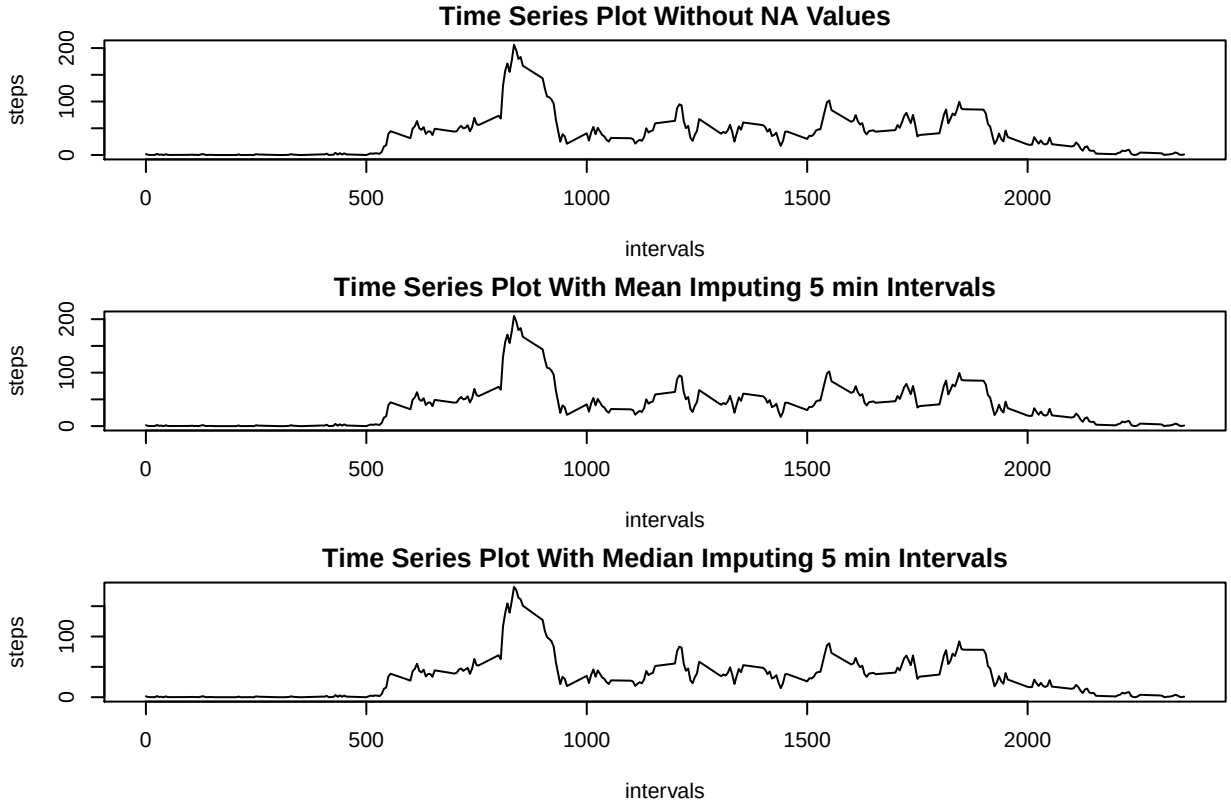
- Survey By Median

```
median_by_date<-group_by(medactivity,date)
median_total_steps<-summarize(median_by_date, TotalStepsPerDay=sum(steps))
median_by_interval<-group_by(medactivity[!is.na(medactivity$steps),],interval)
median_intervalsteps<-summarize(median_by_interval,AverageSteps=mean(steps))
```

- Histogram Plot



- Time Series Plot



Do these values differ from the estimates from the first part of the assignment?

In Histogram Plot mean and median imputing cause non zero values in Not Assigned data. Mean and median imputing cause increase in similar steps of histogram plot.

What is the impact of imputing missing data on the estimates of the total daily number of steps?

Time series plot shows high similarity between imputed NA values and without NA values structure.

- as have been shown in resaults means values changed and median of repaired result shifted so hisogram have smoother characteristics than before

Are there differences in activity patterns between weekdays and weekends?

For this part the weekdays() function may be of some help here.

Use the dataset with the filled-in missing values for this part.

1- Create a new factor variable in the dataset with two levels - "weekday" and "weekend" indicating whether a given date is a weekday or weekend day.

```
head(newactivity,6)
```

```
## Source: local data frame [6 x 4]
##
##       steps       date interval weekdays
## 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
## 6 2.0943396 2012-10-01       25  weekday
```

2- Make a panel plot containing a time series plot (i.e. type = "l") of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all weekday days or weekend days (y-axis).

See the README file in the GitHub repository to see an example of what this plot should look like using simulated data.