Reproducible Research: Peer Assessment 1

hadeer mahmoud

September 20, 2018

Introduction

It is now possible to collect a large amount of data about personal movement using activity monitoring devices such as a Fitbit, Nike Fuelband, or Jawbone Up. These type of devices are part of the â€œquantified selfâ€ movement â€“ 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.

Brief Overview of Data

The data for this assignment can be downloaded from the course web site:

Dataset: **Activity monitoring data** The variables included in this dataset are:

. steps: Number of steps taking in a 5-minute interval (missing values are coded as NA)

. date: The date on which the measurement was taken in YYYY-MM-DD format

. interval: Identifier for the 5-minute interval in which measurement was taken ##Loading and preprocessing the data

```
library(ggplot2)

library(dplyr)

##
## Attaching package: 'dplyr'

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

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

```
library(chron)   #Used for is.weekend() function
```

1. Load the data (read.csv())

```
setwd("C:/Users/compuearth/Documents")
a <- read.csv("activity.csv", header = T , sep = ",")
```

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

```
head(a)
```

```
##   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
```

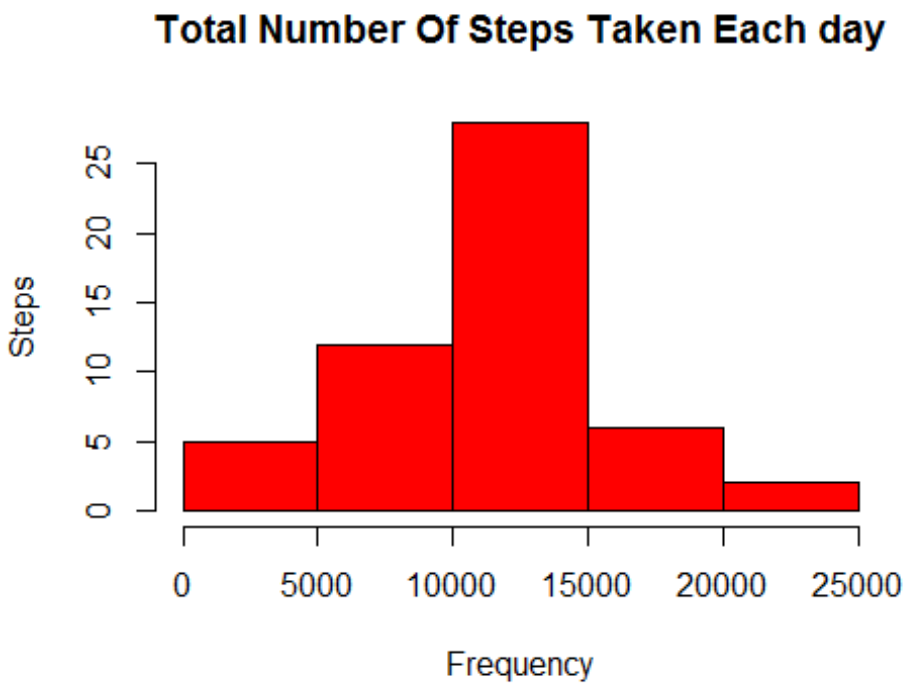
What is mean total number of steps taken per day?

For this part of the assignment, you can ignore the missing values in the dataset. **1. Make a histogram of the total number of steps taken each day**

```
aggsteps<- aggregate(steps ~ date, a, FUN=sum)
head(aggsteps)
```

```
##         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
```

```
#Ploting histogram using hist() from Base Plotting
hist(aggsteps$steps,
     col="red",
     xlab = "Frequency",
     ylab = "Steps",
     main = "Total Number Of Steps Taken Each day")
```



2. Calculate and report the mean and median total number of steps taken per day

```
amean <- mean(aggsteps$steps)
amedian <- median(aggsteps$steps)
amean

## [1] 10766.19
```

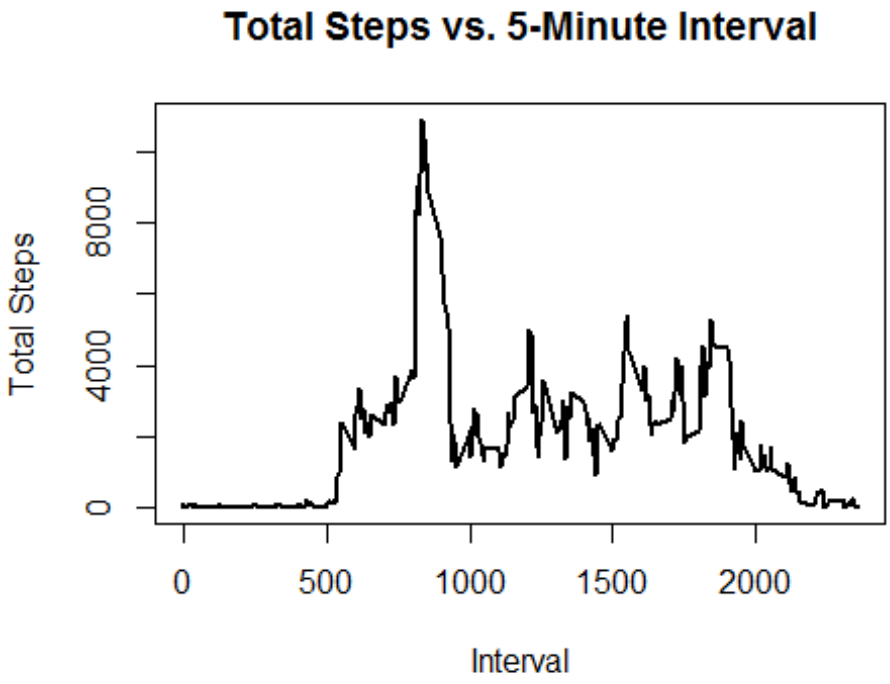
Mean and Median total number of steps taken per day are 10766.19 and 10765 respectively.

What is the average daily activity pattern?

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

```
agginterval <- aggregate(steps ~ interval, a, FUN=sum)

plot(agginterval$interval, agginterval$steps,
     type = "l", lwd = 2,
     xlab = "Interval",
     ylab = "Total Steps",
     main = "Total Steps vs. 5-Minute Interval")
```



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

```
filter(agginterval, steps==max(steps))
```

```
##   interval steps
## 1      835 10927
```

Maximum number of steps (10927 steps) happened in 835th 5-min interval

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)

```
table(is.na(a))
```

```
##
## FALSE  TRUE
## 50400  2304
```

The total number of rows with **NAs are 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.

```
meaninterval<- aggregate(steps ~ interval, a, FUN=mean)

anew <- merge(x=a, y=meaninterval, by="interval")

anew$steps <- ifelse(is.na(anew$steps.x), anew$steps.y, anew$steps.x)

head(anew)
```

```
##   interval steps.x       date  steps.y    steps
## 1        0      NA 2012-10-01 1.716981 1.716981
## 2        0       0 2012-11-23 1.716981 0.000000
## 3        0       0 2012-10-28 1.716981 0.000000
## 4        0       0 2012-11-06 1.716981 0.000000
## 5        0       0 2012-11-24 1.716981 0.000000
## 6        0       0 2012-11-15 1.716981 0.000000
```

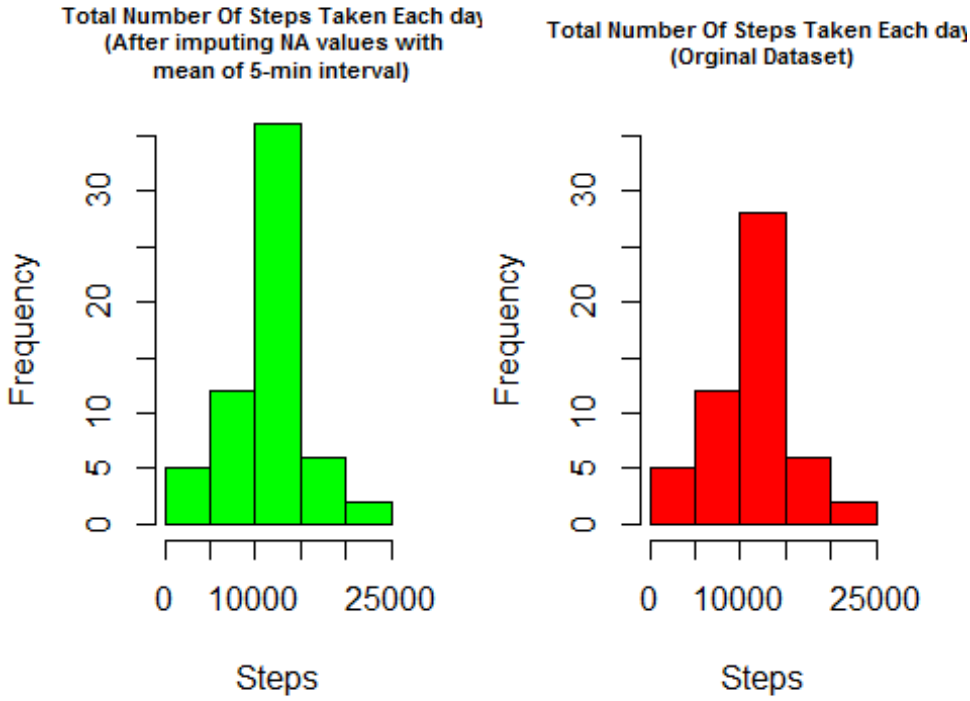
3. Create a new dataset that is equal to the original dataset but with the missing data filled in.

```
anew <- select(anew, steps, date, interval)
head(anew)
```

```
##      steps       date interval
## 1 1.716981 2012-10-01        0
## 2 0.000000 2012-11-23        0
## 3 0.000000 2012-10-28        0
## 4 0.000000 2012-11-06        0
## 5 0.000000 2012-11-24        0
## 6 0.000000 2012-11-15        0
```

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. 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?




```
 par(mfrow=c(1,1)) #Resetting the panel

amean_new <- mean(aggsteps_new$steps)
amedian_new <- median(aggsteps_new$steps)

#Comparing Means
paste("New Mean      :", round(amean_new,2), "," ,
      " Original Mean :", round(amean,2),"," ,
      " Difference :",round(amean_new,2) -  round(amean,2))

## [1] "New Mean      : 10766.19 ,  Original Mean : 10766.19 ,  Difference :
0"

paste("New Median    :", amedian_new, ",",
      " Original Median :", amedian,"," ,
      " Difference :",round(amedian_new-amedian,2))

## [1] "New Median    : 10766.1886792453 ,  Original Median : 10765 ,
Difference : 1.19"
```

The Mean are same but New Median differs from Original Median by 1.19 ##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 â€“ â€œweekdayâ€ and â€œweekendâ€ indicating whether a given date is a weekday or weekend day.

```
table(is.weekend(anew$date))

##
## FALSE  TRUE
## 12960  4608

anew$dayofweek <- ifelse(is.weekend(anew$date), "weekend", "weekday")
table(anew$dayofweek)

##
## weekday weekend
##   12960    4608

 head(anew)

##      steps       date interval dayofweek
## 1 1.716981 2012-10-01        0   weekday
## 2 0.000000 2012-11-23        0   weekday
## 3 0.000000 2012-10-28        0   weekend
## 4 0.000000 2012-11-06        0   weekday
## 5 0.000000 2012-11-24        0   weekend
## 6 0.000000 2012-11-15        0   weekday
```

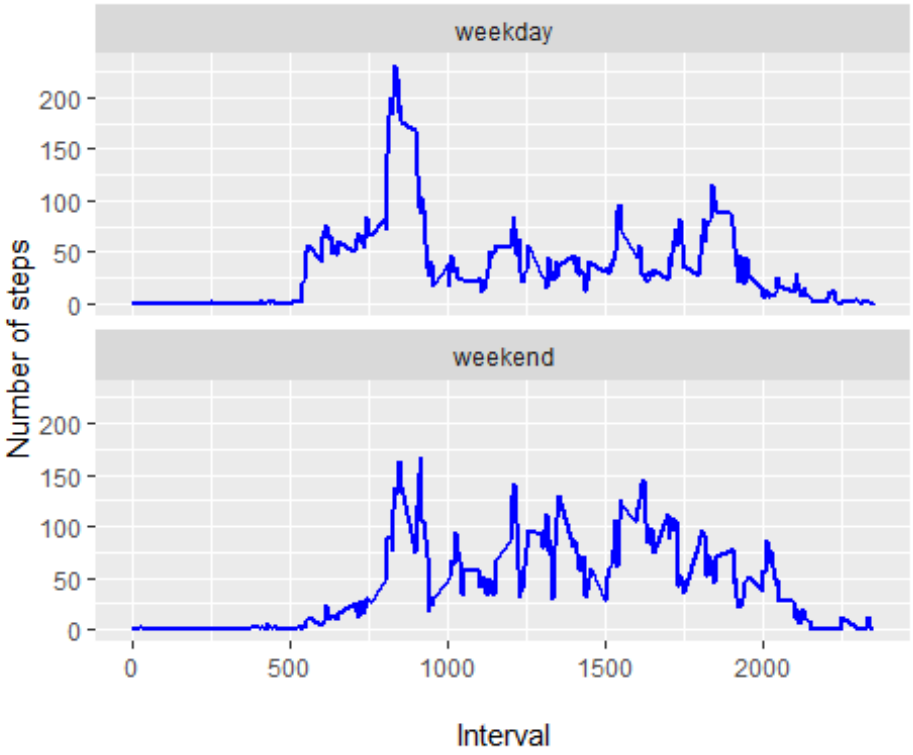
2. Make a panel plot containing a time series plot (i.e. type = “l”) of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all weekday days or weekend days (y-axis). The plot should look something like the following, which was created using simulated data:

```
 meaninterval_new<- aggregate(steps ~ interval + dayofweek, anew, FUN=mean)

head(meaninterval_new)

##   interval dayofweek      steps
## 1        0   weekday 2.25115304
## 2        5   weekday 0.44528302
## 3       10   weekday 0.17316562
## 4       15   weekday 0.19790356
## 5       20   weekday 0.09895178
## 6       25   weekday 1.59035639

 ggplot(meaninterval_new, aes(x=interval, y=steps)) +
  geom_line(color="blue", size=1) +
  facet_wrap(~dayofweek, nrow=2) +
  labs(x="\nInterval", y="\nNumber of steps")
```


weekday
weekend
Number of steps
200
150
100
50
0
0
500
1000
1500
2000
Interval