Reproducible Research-Project 1

Bikram Bhusal
28 July, 2019

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 “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.

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

The dataset is stored in a comma-separated-value (CSV) file and there are a total of 17,568 observations in this dataset.

Loading and preprocessing the data

```
activity<-read.csv("activity.csv",header=TRUE)

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 ...
```

What is mean total number of steps taken per day?

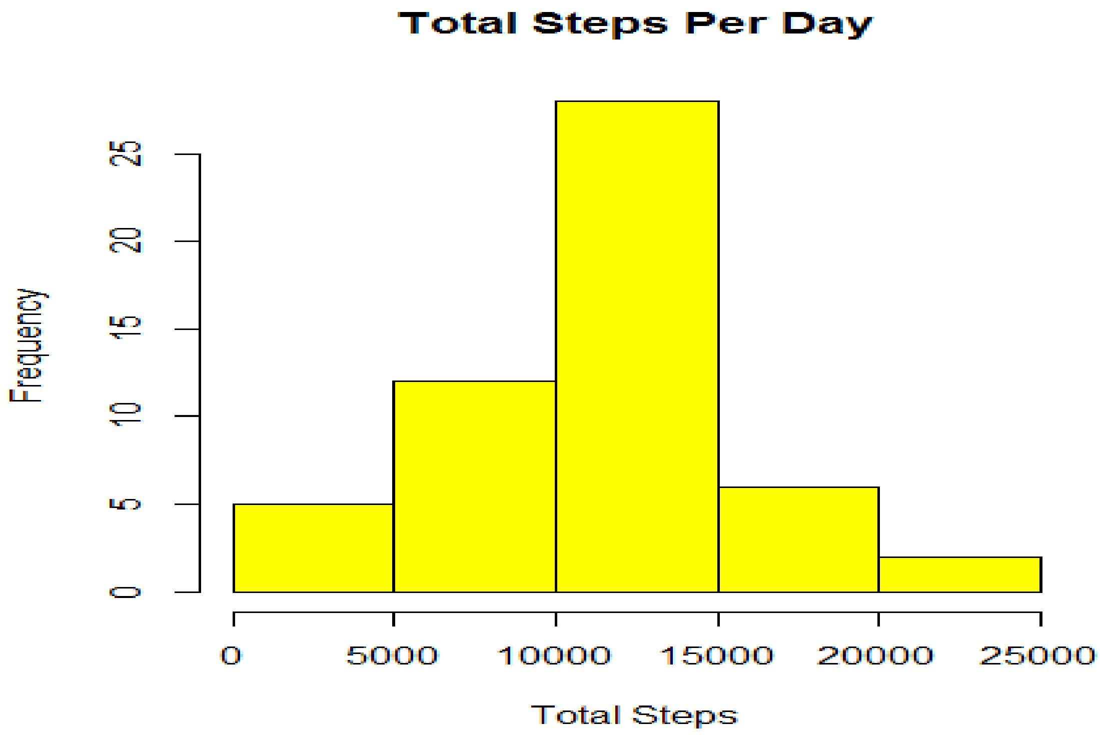
Calculating the total number of steps taken per day

```
head(steps_perday)
```

```
## 2012-10-01 2012-10-02 2012-10-03 2012-10-04 2012-10-05 2012-10-06

##         NA        126      11352      12116      13294      15420
```

Making the histogram of total steps per day

```
hist(steps_perday,col = "yellow",main="Total Steps Per Day",  xlab="Total Steps",ylab = "Frequency")
```



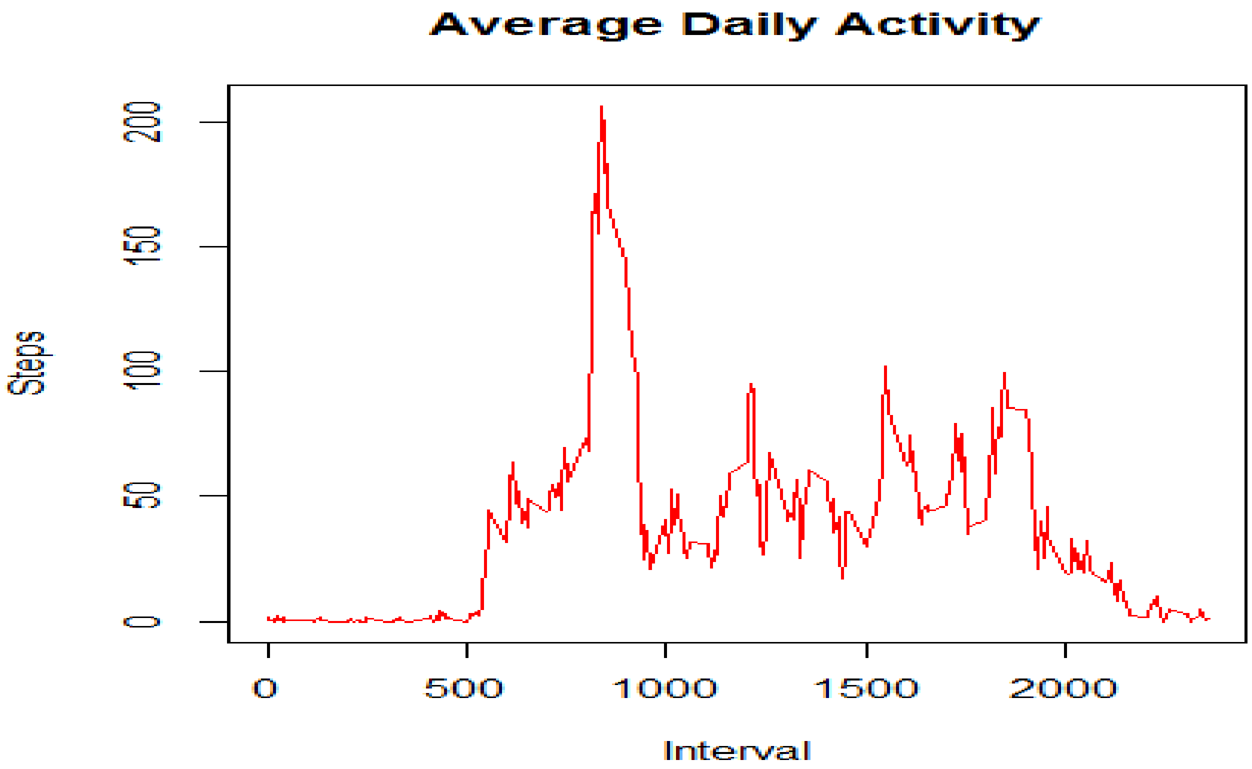
Calculating the mean and median of the total number of steps taken per day

```
Mean_PerDay<-mean(steps_perday,na.rm = T)

Median_PerDay<-median(steps_perday,na.rm = T)

Mean_PerDay

## [1] 10766.19

Median_PerDay

## [1] 10765
```

What is the average daily activity pattern?

Making a time series plot of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all days (y-axis)

```
steps_PerInt<-tapply(activity$steps,activity$interval,mean,na.rm=TRUE)

plot(as.numeric(names(steps_PerInt)),steps_PerInt,main="Average Daily Activity",xlab="Interval",ylab="Steps",type="l",col="red")
```



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

```
maxA_Interval<-names(which.max(steps_PerInt))

maxA_steps<-max(steps_PerInt)

maxA_Interval
```

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

```
maxA_steps
```

```
## [1] 206.1698
```

Thus 835th interval contains the maximum number of steps(206.1698).

Imputing missing values

Calculating the total number of missing values in the dataset (i.e. the total number of rows with NAs)

```
sum(is.na(activity$steps))
## [1] 2304
```

Here,calculating the mean of average steps per day across all the data(without NA’s) and replace all NAs by this:

```
avg_steps<-tapply(activity$steps,activity$date,mean,na.rm=TRUE)
ma<-mean(avg_steps,na.rm = TRUE)
nas<-which(is.na(activity$steps))
l<-length(nas)
## Replacing NAs by mean
for(i in 1:l){
  activity[nas[i],1]=ma
}
```

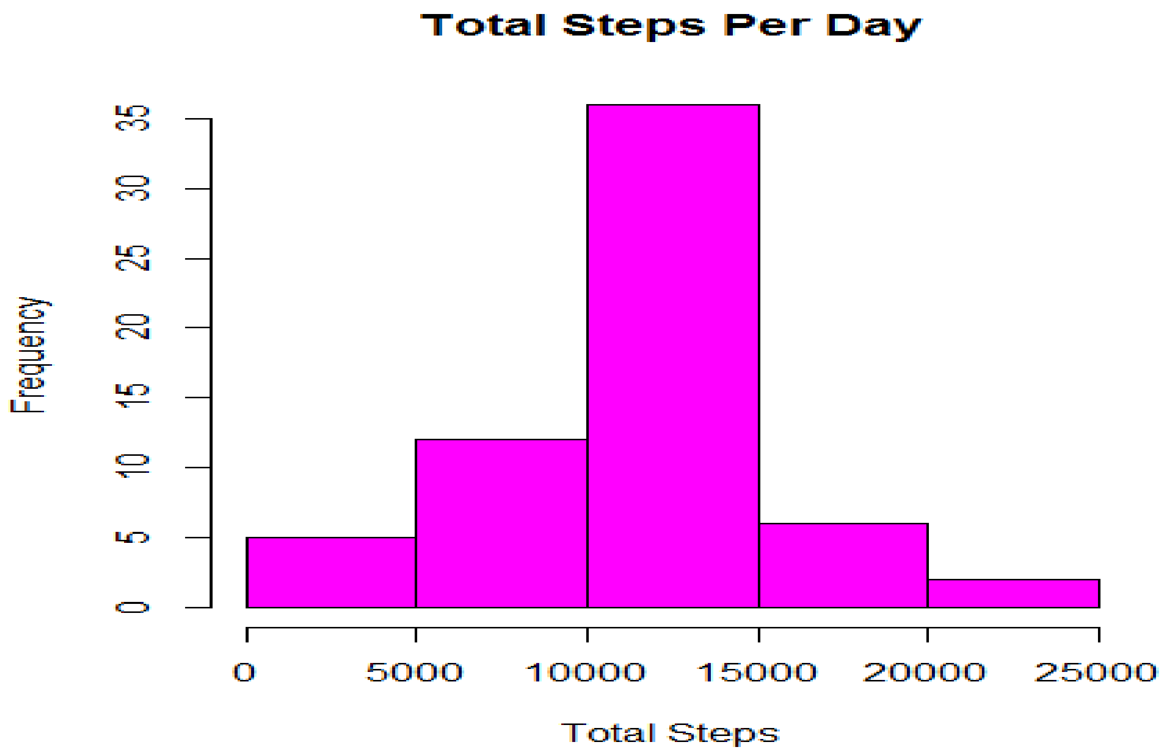
Checking weather NAs are successfully replaced? Also see how our new dataset look like.

```
sum(is.na(activity$steps))
## [1] 0
str(activity)
## 'data.frame':    17568 obs. of  3 variables:
##  $ steps   : num  37.4 37.4 37.4 37.4 37.4 ...
##  $ 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 ...
```

Yes NAs are successfully replaced my mean

Now Making a histogram of the total number of steps taken each day:

```
steps_2perday<-tapply(activity$steps,activity$date, FUN=sum)
hist(steps_2perday,col = "6",main="Total Steps Per Day",  xlab="Total Steps",
ylab = "Frequency")
```



And Calculating the mean and median total number of steps taken per day.

```
Mean_2PerDay<-mean(steps_2perday,na.rm = T)
Median_2PerDay<-median(steps_2perday,na.rm = T)
Mean_2PerDay
## [1] 10766.19
Median_2PerDay
## [1] 10766.19
```

Thus mean and median of the new dataset are 107.66 and 10766.19 respectivelly.

Are there differences in activity patterns between weekdays and weekends?

In this section, we will use dplyr package .

```
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
```

Now creating a new variable in the dataset named “day” that shows the day of the week in terms of weekday or weekend

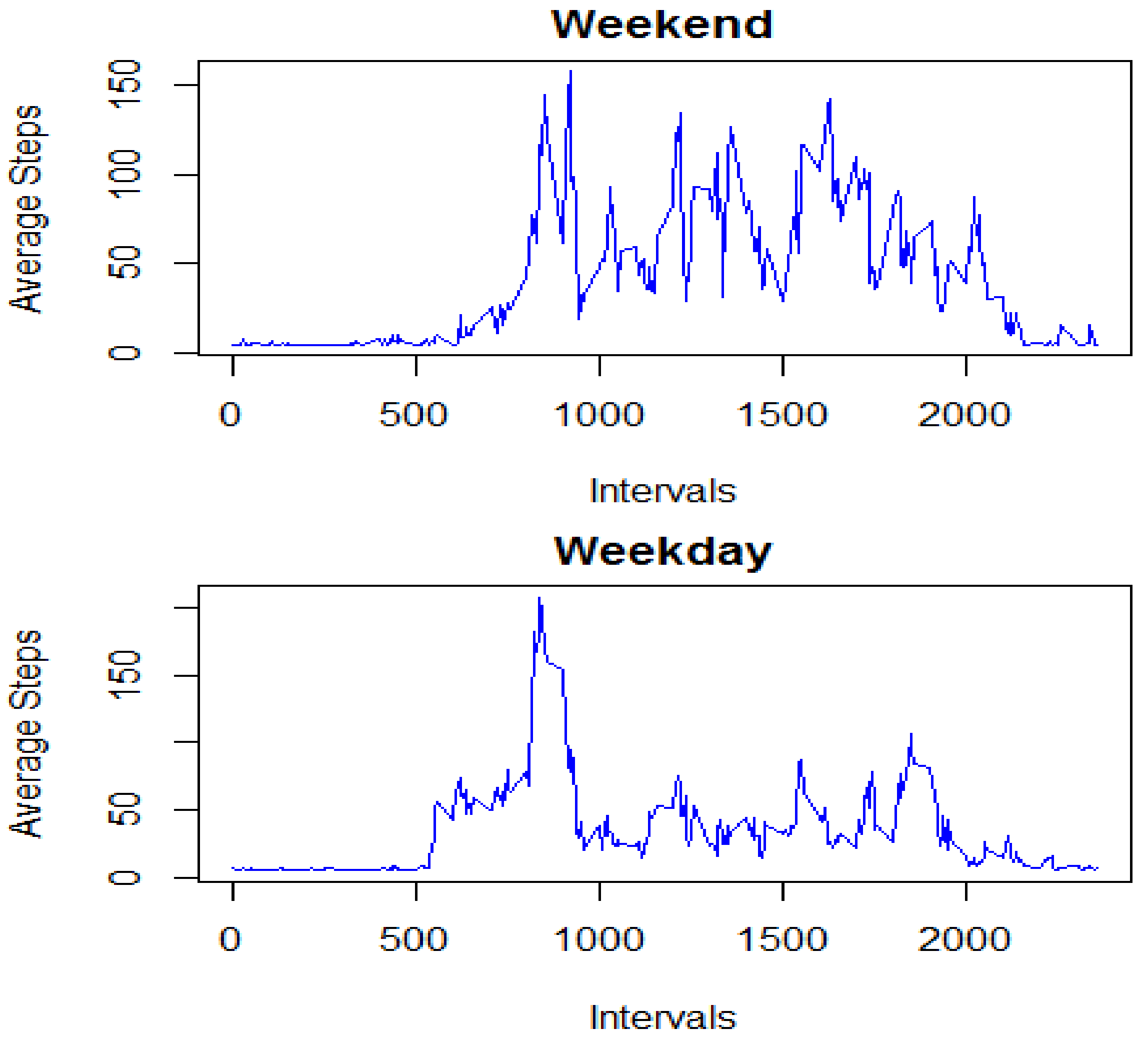
```
activity$day<-ifelse(weekdays(as.Date(activity$date))=="Saturday"|weekdays(as
.Date(activity$date))=="Sunday","weekend","weekday")
activity$day<-as.factor(activity$day)
str(activity)
```

```
## 'data.frame':    17568 obs. of  4 variables:
##  $ steps   : num  37.4 37.4 37.4 37.4 37.4 ...
##  $ 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 ...
##  $ day     : Factor w/ 2 levels "weekday","weekend": 1 1 1 1 1 1 1 1 1 1 .
..
```

Plotting the weekday and weekend data in separate graphs:

```
act_wknd<-subset(activity,as.character(activity$day)=="weekend")
act_wkdy<-subset(activity,as.character(activity$day)=="weekday")
steps_wknd<-with(act_wknd,tapply(steps,interval,mean,na.rm=T))
steps_wkdy<-with(act_wkdy,tapply(steps,interval,mean,na.rm=T))
int_wknd<-unique(act_wknd$interval)
int_wkdy<-unique(act_wkdy$interval)
new_wknd<-data.frame(cbind(steps_wknd,int_wknd))
new_wkdy<-data.frame(cbind(steps_wkdy,int_wkdy))
```

```
par(mfrow=c(2,1),mar=c(4,4,2,1))
plot(new_wknd$int_wknd,new_wknd$steps_wknd,type = "l",xlab = "Intervals",
     ylab = "Average Steps",main = "Weekend",col="4")
plot(new_wkdy$int_wkdy,new_wkdy$steps_wkdy,type = "l",xlab = "Intervals",
     ylab = "Average Steps",main = "Weekday",col="4")
```



From these it is clear that the average steps over the weekends show higher pattern than that of the weekend.