Assignment 1

Vladimir Urrutia

Monday, Sep 20, 2021

This assignment is for class Reproducible Research(Peer Assessment 1) Coursera provided by JOHNS HOPKINS. This paper contains the process of the analysis of the data.

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 quantified 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.

Analysis

Load and preprocess the data

```
setwd("L:/R/Reproducible Research")
data <- read.csv("./RepData_PeerAssessment1/activity/activity.csv", header = TRUE, sep = ",")
index <- complete.cases(data)
data1 <- data[index,]
```

The row data is like :

```
head(data)
```

```
##   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
```

The processed data :

```
head(data1)
```

```
##     steps       date interval
## 289     0 2012-10-02        0
## 290     0 2012-10-02        5
## 291     0 2012-10-02       10
## 292     0 2012-10-02       15
## 293     0 2012-10-02       20
## 294     0 2012-10-02       25
```

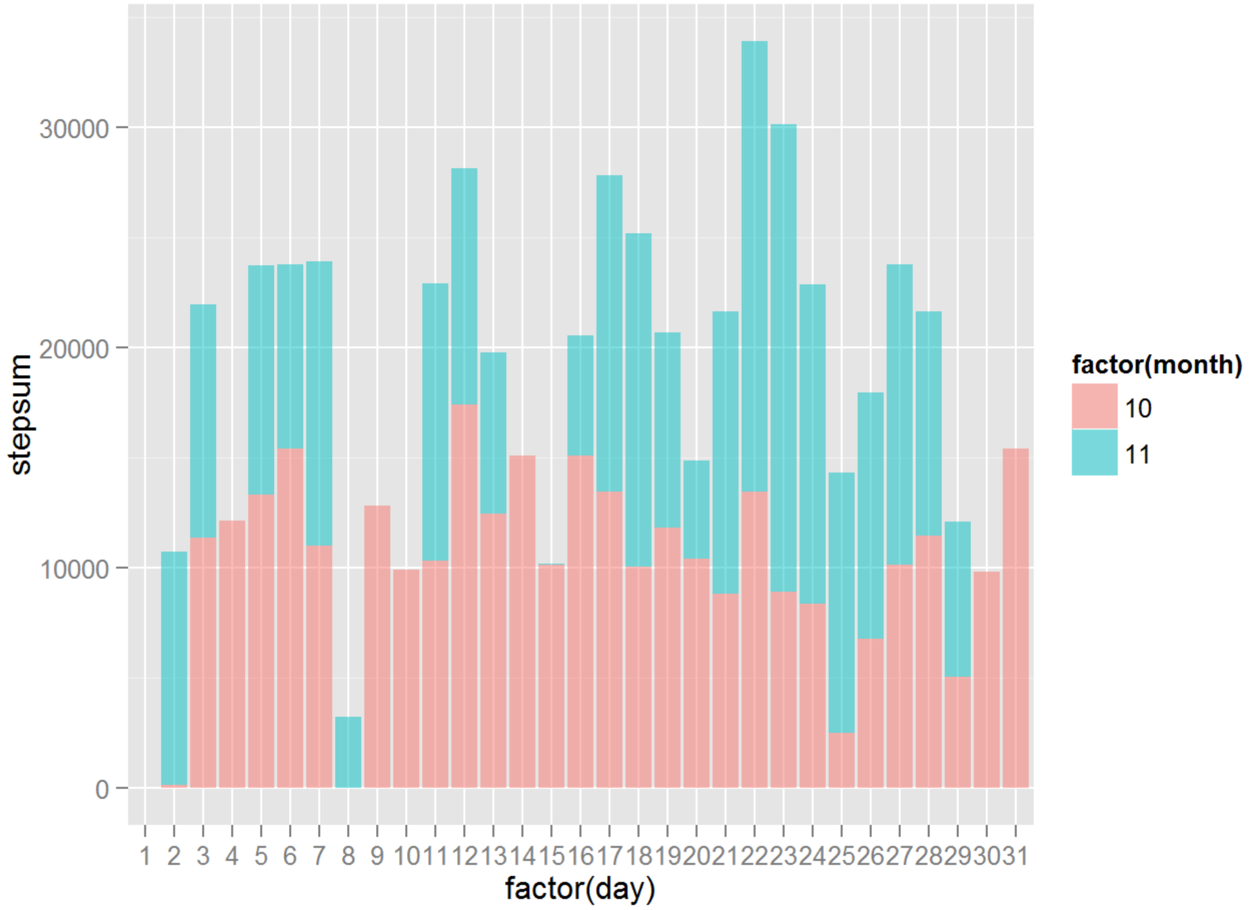
Calculate mean total number of steps taken per day

1. Draw a figure of histogram of the total number of steps taken each day:

```
stepsum <- as.numeric()
for (i in 1:length(date))
{
    stepsum[i] <- sum(data1$steps[which(data1$date == date[i])])
}
month <- as.POSIXlt(date)$mon + 1
day <- as.POSIXlt(date)$mday
stepsum <- data.frame(stepsum,date,month,day)
library(ggplot2)
library(plyr)
```

So you can see the figure here as follows:

```
mm <- ddply(stepsum, "date",summarise, stepsum = sum(stepsum))
ggplot(data=mm,aes(x = factor(day),fill = factor(month), y = stepsum)) + geom_bar(stat = "identity",alpha
=0.5)
```



Now, the figure above is the statictis of total number of the steps every day. Color “Red” represent Month Oct.and “blue” stands for Nov. In this cumulative distribution figure, we can clearly the difference between two months and the distribution of the same month.

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

```
Mean <- as.numeric()
Median <- as.numeric()
for (i in 1:length(date))
{
    Mean[i] <- mean(data1$steps[which(data1$date == date[i])])
}
data2 <- data1[-which(data1$steps==0),]
for (i in 1:length(date))
{
    Median[i] <- median(data2$steps[which(data2$date == date[i])])
}
```

Some part of the result is as follows:

```
##      mean median       Date
## 1     NaN     NA 2012-10-01
## 2  0.4375   63.0 2012-10-02
## 3 39.4167   61.0 2012-10-03
## 4 42.0694   56.5 2012-10-04
## 5 46.1597   66.0 2012-10-05
## 6 53.5417   67.0 2012-10-06
```

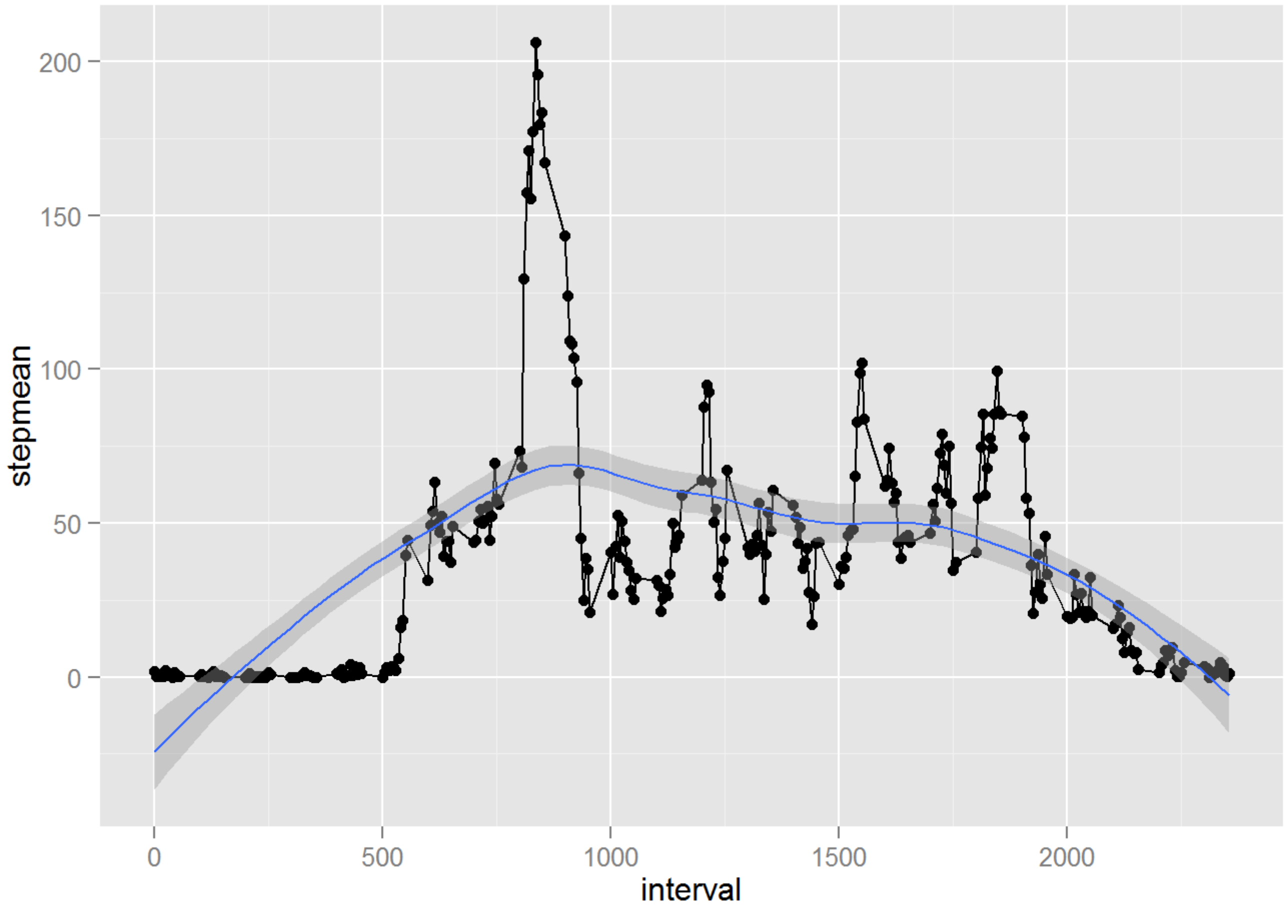
The “mean” variable represents the mean value of total number of steps and the “median” variable represents the median value of the total number of steps. Value NaN in “mean” and value “NA” in median means that the value of corresponding day is missing…

Find average daily activity pattern

In order to distinguish the different days and get an obvious comparation, I plot the data within the same picture and use different colors:

```
interval <- unique(data1$interval)
stepmean <- as.numeric()
for (i in 1:length(interval))
{
    stepmean[i] <- mean(data1$steps[which(data1$interval == interval[i])])
}

stepmean <- data.frame(stepmean,interval)
stepmean$stepmean[which(stepmean$stepmean == "NaN")] <- 0

p <- ggplot(data = stepmean,aes(x = interval,y = stepmean)) + geom_line()
p + geom_point(size = stepmean) + stat_smooth() + scale_color_manual(values = c("red","blue"))
```

```
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to cha
nge the smoothing method.
```



And the 5-minute interval that on average across all the days in the dataset, contains the maximum number of steps is the 615th interval on 2012-11-27:

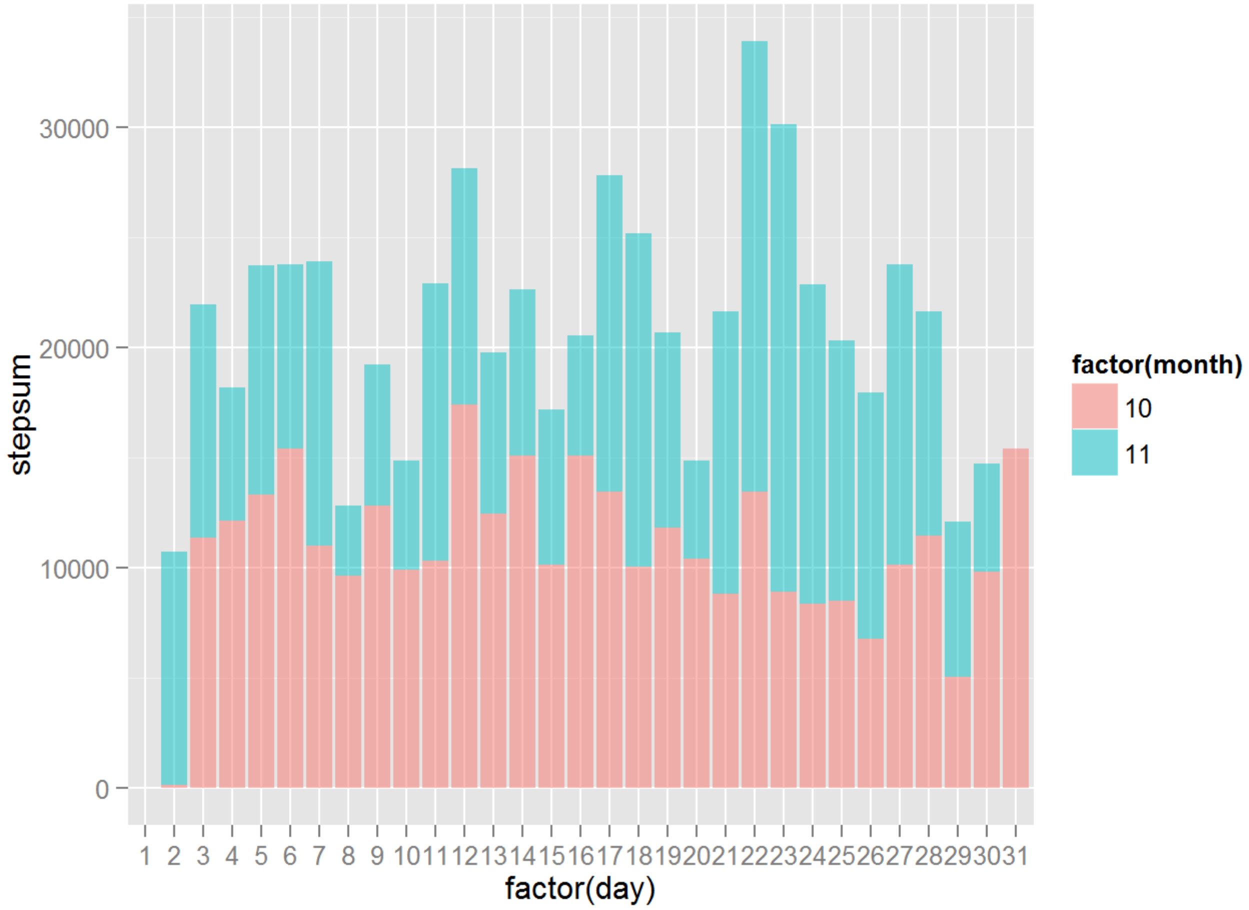
```
data2[which(data2$step == max(data2$step)),]
```

```
##       steps       date interval
## 16492   806 2012-11-27      615
```

Imputing missing values

```
number <- dim(data)[1] - sum(complete.cases(data))

Mean[which(Mean == "NaN")] <- 0
Mean[which(Mean == 0)] <- (Mean[which(Mean == 0)] + Mean[(which(Mean == 0) + 31)%%62])/2



for (i in c(1,8,32,35,40,41,45,61))
{
    data$steps[which(data$date == date[i])] <- Mean[i]
}

stepsum2 <- as.numeric()

for (i in 1:length(date))
{
  stepsum2[i] <- sum(data$steps[which(data$date == date[i])])
}



month <- as.POSIXlt(date)$mon + 1
day <- as.POSIXlt(date)$mday
stepsum2 <- data.frame(stepsum2,date,month,day)
mm <- ddply(stepsum2, "date", summarise, stepsum = sum(stepsum2))
```

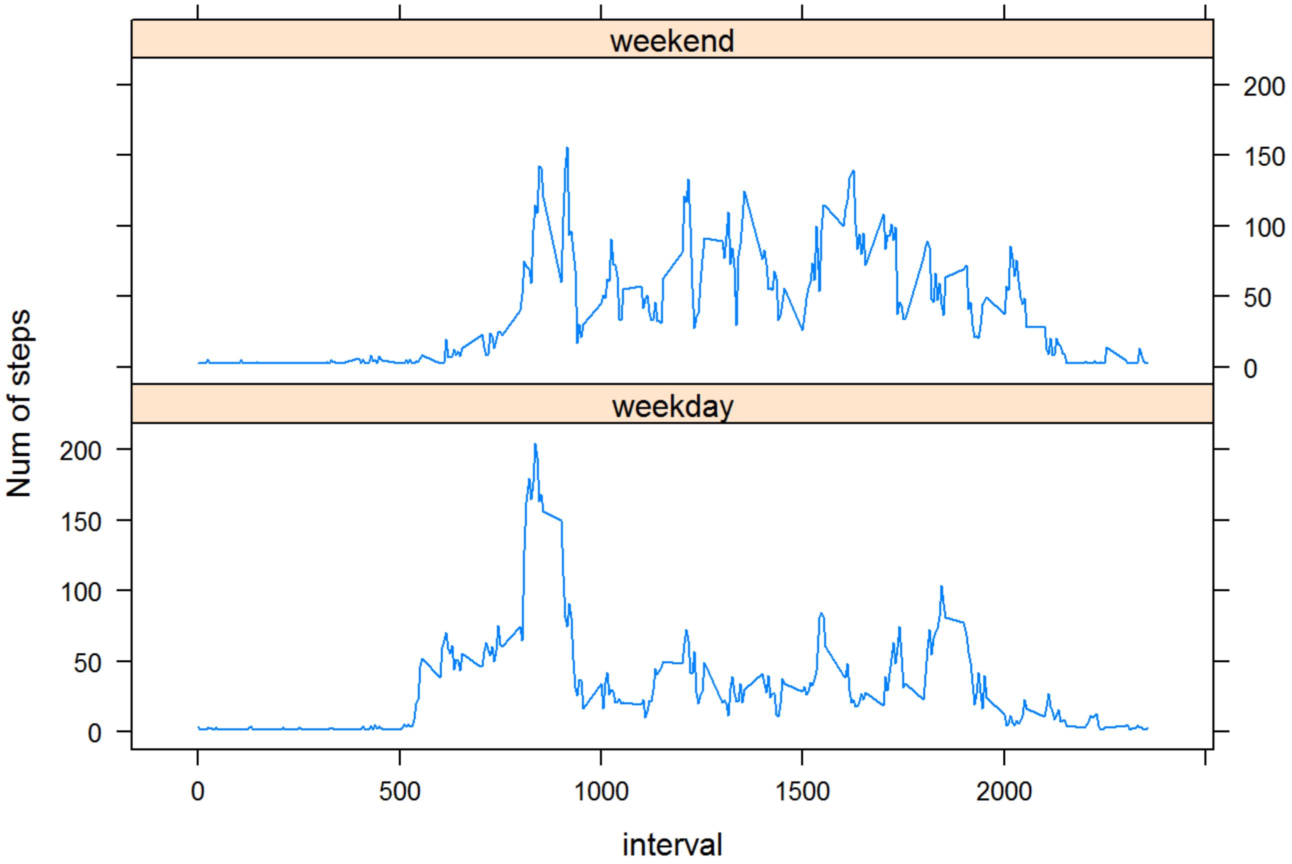
```
ggplot(data=mm, aes(x = factor(day),fill = factor(month), y = stepsum)) + geom_bar(stat = "identity",alpha=0.5)
```



I use the average data of the same day to substitute the missing value. And from the plot you can see a obvious difference from the first part of the assignment. Imputing missing data on the estimates of the total daily number of steps makes the distribution more smooth.

Find differences in activity patterns between weekdays and weekends

```
workday <- as.character()
day <- weekdays(as.Date(data$date))
workday[which(day == "Sunday" | day == "Saturday")]  <- "weekend"
workday[which(day == "Monday" | day == "Tuesday" | day == "Wednesday"| day == "Thursday" | day == "Frida
y")]  <- "weekday"
Data <- data.frame(data,workday)


mmm <- ddply(Data, c("interval","workday"),summarise, steps = mean(steps))
nnn <- mmm[which(mmm$workday == "weekday"),]
nn <- mmm[which(mmm$workday == "weekend"),]
Data2 <- t(data.frame(t(nnn),t(nn)))
Data2 <- as.data.frame(Data2)
Data2$steps <- as.numeric(as.character(Data2$steps))
Data2$interval <- as.numeric(as.character(Data2$interval))
row.names(Data2) <- c()
library(lattice)

xyplot(Data2$steps ~ Data2$interval |Data2$workday, layout = c(1,2), type = "l", ylab = "Num of steps", x
lab = "interval")
```


weekend
weekday
Num of steps
interval
0
50
100
150
200
0
500
1000
1500
2000