RepData\_PeerAssessment

Razib

January 3, 2019

## Activity monitoring devices: Peer Assessment 1

## 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.

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.

## Data

The data for this assignment can be downloaded from the course web site using below link.

[Activity monitoring dataset](https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip)

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

Load the data:

echo = TRUE  
setwd("../RepData\_PeerAssessment")  
activity <- NULL  
activity <- read.csv("activity.csv", header = T, sep = ",")

The variable created during by this code are set to NULL.

echo = TRUE  
df\_summary <- NULL  
su2 <- NULL  
su <- NULL  
mn\_int <- NULL  
activity2 <- NULL  
mean\_su2 <- NULL  
median\_su2 <- NULL  
activity2\_weekend <- NULL  
activity2\_weekday <- NULL  
mean\_activity2\_weekday <- NULL  
mean\_activity2\_weekend <- NULL

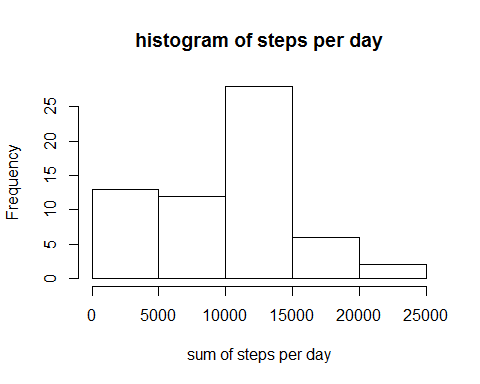
## What is mean total number of steps taken per day?

First, the total (sum) of steps is determined for every single date.

echo = TRUE  
su <- tapply(activity$steps, activity$date, sum, na.rm=T)

Hereafter is presented a histogram of the total number of steps taken each day.

echo = TRUE  
hist(su, xlab = "sum of steps per day", main = "histogram of steps per day")



The mean and the median total number of steps taken per day are reported :

echo = TRUE  
mean\_su <- round(mean(su))  
median\_su <- round(median(su))  
  
print(c("The mean is",mean\_su))

## [1] "The mean is" "9354"

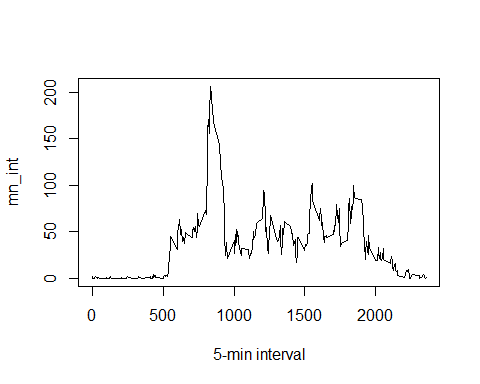
print(c("The median is",median\_su))

## [1] "The median is" "10395"

## what is the average daily activity pattern ?

A time series plot of the 5-minute interval and the average number of steps taken (averaged across all days) is shown below:

echo = TRUE  
mn\_int <- tapply(activity$steps, activity$interval, mean, na.rm=T)  
plot(mn\_int ~ unique(activity$interval), type="l", xlab = "5-min interval")



The 5-minute interval (on average across all the days in the dataset) that contains the maximum number of steps is the following (below are shown the interval showing the max. number of steps and the value of the max. number of steps):

echo = TRUE  
mn\_int[which.max(mn\_int)]

## 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.

First, in order to visualize in which variable the NAs are:

echo = TRUE  
table(is.na(activity) == TRUE)

##   
## FALSE TRUE   
## 50400 2304

All of the NA’s are in the steps variable. There are 2304 NA’s.

## Strategy for filling in all of the missing values in the dataset

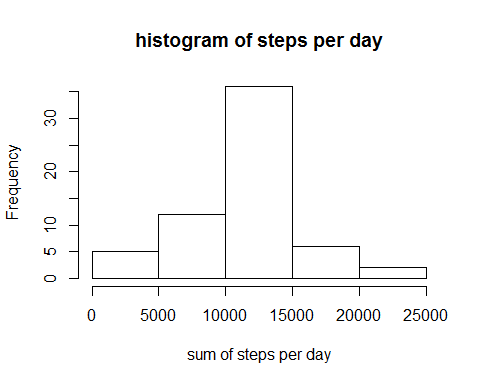
The following strategy is chosen: for any NA is the step variable, the mean (of steps) of the corresponding interval is taken as the replacing value.

The ‘mn\_int’ contains the mean for each single interval calculated over the 61 days. The right value coming from ‘mn\_int’ is going to be used to replace the NA at the same interval.

echo = TRUE  
activity2 <- activity # creation of the dataset that will have no more NAs  
for (i in 1:nrow(activity)){  
 if(is.na(activity$steps[i])){  
 activity2$steps[i]<- mn\_int[[as.character(activity[i, "interval"])]]  
 }  
}

Below is a histogram of the total number of steps taken each day. The mean and median total number of steps taken per day are reported.

echo = TRUE  
su2 <- tapply(activity2$steps, activity2$date, sum, na.rm=T)  
hist(su2, xlab = "sum of steps per day", main = "histogram of steps per day")



mean\_su2 <- round(mean(su2))  
median\_su2 <- round(median(su2))

The new values are :

echo = TRUE  
print(c("The mean is",mean\_su2))

## [1] "The mean is" "10766"

print(c("The median is",median\_su2))

## [1] "The median is" "10766"

In order to compare the new values with the “old” values:

echo = TRUE  
df\_summary <- rbind(df\_summary, data.frame(mean = c(mean\_su, mean\_su2), median = c(median\_su, median\_su2)))  
rownames(df\_summary) <- c("with NA's", "without NA's")  
print(df\_summary)

## mean median  
## with NA's 9354 10395  
## without NA's 10766 10766

For comparison with NA’s and without (see earlier):

echo = TRUE  
summary(activity2)

## steps date interval   
## Min. : 0.00 2012-10-01: 288 Min. : 0.0   
## 1st Qu.: 0.00 2012-10-02: 288 1st Qu.: 588.8   
## Median : 0.00 2012-10-03: 288 Median :1177.5   
## Mean : 37.38 2012-10-04: 288 Mean :1177.5   
## 3rd Qu.: 27.00 2012-10-05: 288 3rd Qu.:1766.2   
## Max. :806.00 2012-10-06: 288 Max. :2355.0   
## (Other) :15840

It confirms there is no more NAs in the steps variable.

## Are there differences in activity patterns between weekdays and weekends?

A new column is added to the dataframe, this column will contain the factor “weekday days”" or “weekend days”.

echo = TRUE  
activity2$weekday <- c("weekday")  
activity2[weekdays(as.Date(activity2[, 2])) %in% c("Saturday", "Sunday", "samedi", "dimanche", "saturday", "sunday", "Samedi", "Dimanche"), ][4] <- c("weekend")  
table(activity2$weekday == "weekend")

##   
## FALSE TRUE   
## 12960 4608

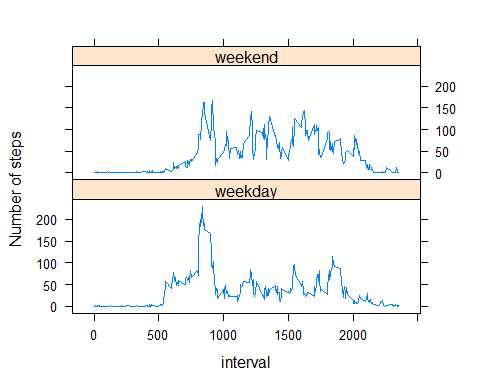
activity2$weekday <- factor(activity2$weekday)

In order to visualize the difference bewteen weekends and days of the week, a new dataframe is created to be usable by the lattice package. First, the data are calculated:

echo = TRUE  
activity2\_weekend <- subset(activity2, activity2$weekday == "weekend")  
activity2\_weekday <- subset(activity2, activity2$weekday == "weekday")  
  
mean\_activity2\_weekday <- tapply(activity2\_weekday$steps, activity2\_weekday$interval, mean)  
mean\_activity2\_weekend <- tapply(activity2\_weekend$steps, activity2\_weekend$interval, mean)

Then the dataframe is prepared and the plot is. plotted !

echo = TRUE  
library(lattice)  
df\_weekday <- NULL  
df\_weekend <- NULL  
df\_final <- NULL  
df\_weekday <- data.frame(interval = unique(activity2\_weekday$interval), avg = as.numeric(mean\_activity2\_weekday), day = rep("weekday", length(mean\_activity2\_weekday)))  
df\_weekend <- data.frame(interval = unique(activity2\_weekend$interval), avg = as.numeric(mean\_activity2\_weekend), day = rep("weekend", length(mean\_activity2\_weekend)))  
df\_final <- rbind(df\_weekday, df\_weekend)  
  
xyplot(avg ~ interval | day, data = df\_final, layout = c(1, 2),   
 type = "l", ylab = "Number of steps")



It can be observed that there is a small difference between the period.