Peer Assignmnet - Activity Monitoring Device

Niall Graham

16 December 2017

# Overview

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.

# Data

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

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

Loading the Data

echo=TRUE   
setwd("C:/Users/Niall Graham/Desktop/Data Science/Reproducible Data/Peer Grade assignment V2")  
activity <- NULL  
activity <- read.csv("activity.csv", header = T , sep = ",")

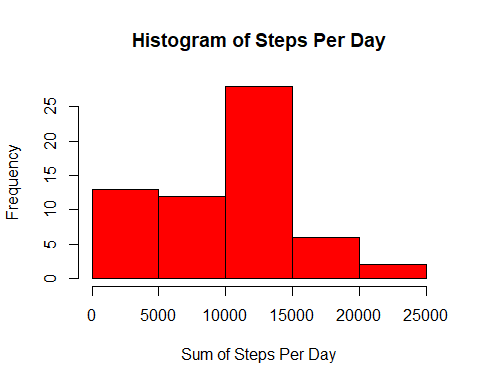
echo = TRUE  
df\_summary <- NULL  
su2 <- NULL  
su <- NULL  
mn\_int <- NULL  
activity2 <- NULL  
mean\_su2 <- NULL  
median\_su2 <- NULL  
activity2\_weekend <- NULL  
activity2\_weekday <- NULL  
mean\_activity2\_weekday <- NULL  
mean\_activity2\_weekend <- NULL

### Mean of Total Number of Steps taken per day.

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

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", col = "red")



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"

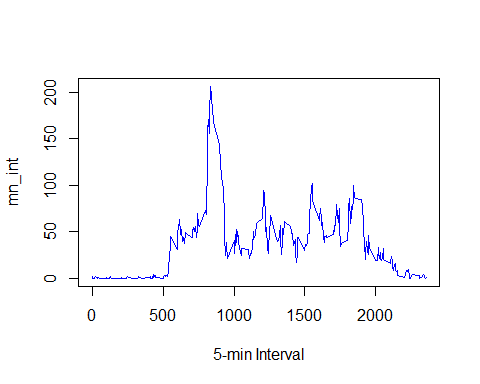
print(c("The median is",median\_su))

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

### The Average Daily Activity Pattern

Plot of the 5-minute interval and the average number of steps taken.

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", col = "blue")



The 5-minute interval (on average across all the days in the dataset) that contains the maximum number of steps is the following:

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

## 835   
## 206.1698

## Imputing missing values

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

summary(activity)

## 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.: 12.00 2012-10-05: 288 3rd Qu.:1766.2   
## Max. :806.00 2012-10-06: 288 Max. :2355.0   
## NA's :2304 (Other) :15840

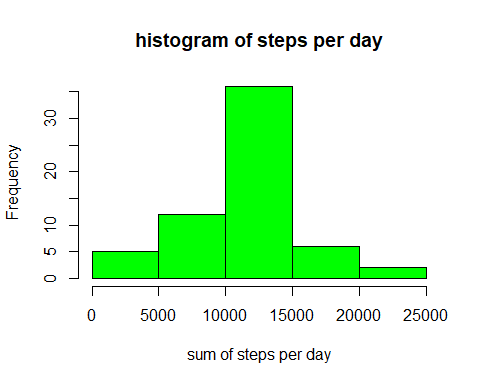
There are 2304 NA's.

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

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"])]]  
 }  
}

Histogram of the Total Number of Steps Taken Each Day.

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", col = "green")



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

### 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)

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")

