Activity Monitoring: Peer Assignment 1

December 15, 2015

## 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 activityCSV 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.

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 Processing the Data

Presuming the reader as set their working directory, the first step is to load the data:

activityCSV\_raw <- read.csv("C:/Users/Iso/Documents/activity.csv")

To process/transform the data into a more suitable format we will start by changing the date format, distinguishing the weekdays from the weekends, and then creating the "cleaned" data version.

activityCSV\_raw$date <- as.POSIXct(activityCSV\_raw$date, format="%Y-%m-%d")  
  
  
activityCSV\_raw <- data.frame(date=activityCSV\_raw$date,   
 weekday=tolower(weekdays(activityCSV\_raw$date)),   
 steps=activityCSV\_raw$steps,   
 interval=activityCSV\_raw$interval)  
  
  
activityCSV\_raw <- cbind(activityCSV\_raw,   
 daytype=ifelse(activityCSV\_raw$weekday == "saturday" |   
 activityCSV\_raw$weekday == "sunday", "weekend",   
 "weekday"))  
  
  
activityCSV <- data.frame(date=activityCSV\_raw$date,   
 weekday=activityCSV\_raw$weekday,   
 daytype=activityCSV\_raw$daytype,   
 interval=activityCSV\_raw$interval,  
 steps=activityCSV\_raw$steps)

To view the raw data we can type:

head(activityCSV\_raw)

## date weekday steps interval daytype  
## 1 2012-10-01 monday NA 0 weekday  
## 2 2012-10-01 monday NA 5 weekday  
## 3 2012-10-01 monday NA 10 weekday  
## 4 2012-10-01 monday NA 15 weekday  
## 5 2012-10-01 monday NA 20 weekday  
## 6 2012-10-01 monday NA 25 weekday

To view the processed data we type:

head(activityCSV)

## date weekday daytype interval steps  
## 1 2012-10-01 monday weekday 0 NA  
## 2 2012-10-01 monday weekday 5 NA  
## 3 2012-10-01 monday weekday 10 NA  
## 4 2012-10-01 monday weekday 15 NA  
## 5 2012-10-01 monday weekday 20 NA  
## 6 2012-10-01 monday weekday 25 NA

## What is Mean Total Number of Steps Taken Per Day?

First, we are asked to total the number of steps taken each day (and to ignore the missing values):

sumData <- aggregate(activityCSV$steps, by=list(activityCSV$date), FUN=sum, na.rm=TRUE)

It would be better if we could rename the attributes for convenience:

names(sumData) <- c("date", "total")

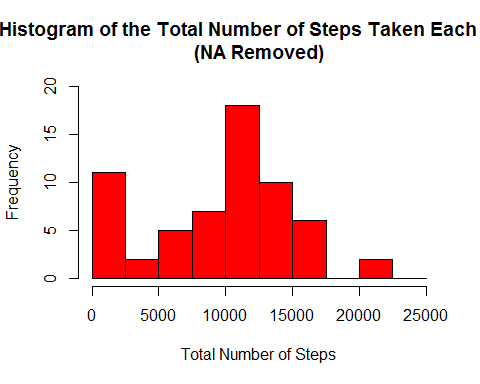
To display the first few rows:

head(sumData)

## date total  
## 1 2012-10-01 0  
## 2 2012-10-02 126  
## 3 2012-10-03 11352  
## 4 2012-10-04 12116  
## 5 2012-10-05 13294  
## 6 2012-10-06 15420

Next, we are asked to produce a histogram:

hist(sumData$total,   
 breaks=seq(from=0, to=25000, by=2500),  
 col="red",   
 xlab="Total Number of Steps",   
 ylim=c(0, 20),   
 main="Histogram of the Total Number of Steps Taken Each Day\n(NA Removed)")



The mean and median of the total number of steps taken per day can be found by:

mean(sumData$total)

## [1] 9354.23

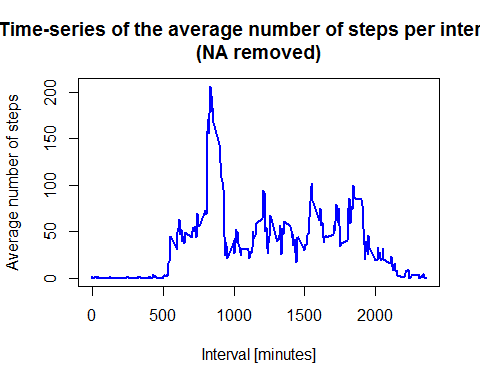
median(sumData$total)

## [1] 10395

## What is the Average Daily activityCSV Pattern?

To make a time series plor, we type in:

meanData <- aggregate(activityCSV$steps,   
 by=list(activityCSV$interval),   
 FUN=mean,   
 na.rm=TRUE)  
  
names(meanData) <- c("interval", "mean")  
  
plot(meanData$interval,   
 meanData$mean,   
 type="l",   
 col="blue",   
 lwd=2,   
 xlab="Interval [minutes]",   
 ylab="Average number of steps",   
 main="Time-series of the average number of steps per intervals\n(NA removed)")



Check your work by displaying:

head(meanData)

## interval mean  
## 1 0 1.7169811  
## 2 5 0.3396226  
## 3 10 0.1320755  
## 4 15 0.1509434  
## 5 20 0.0754717  
## 6 25 2.0943396

We can find out which 5-minute interval contains the maximum number of steps byusing the following code:

max\_pos <- which(meanData$mean == max(meanData$mean))  
  
max\_interval <- meanData[max\_pos, 1]

## Inputing the 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.

We are asked to:

1. Calculate and report the total number of missing values in the dataset (i.e. the total number of rows with NAs)
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.
3. Create a new dataset that is equal to the original dataset but with the missing data filled in.
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.

We begin by counting how many NA fields are present, locating them and then replacing them with the mean:

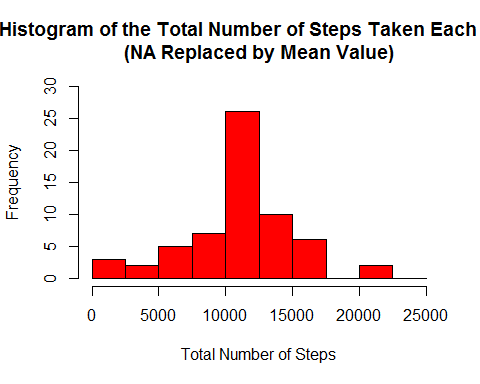
NA\_count <- sum(is.na(activityCSV$steps))  
  
na\_pos <- which(is.na(activityCSV$steps))  
  
mean\_Rep <- rep(mean(activityCSV$steps, na.rm=TRUE), times=length(na\_pos))

Next, we want to create a new dataset. We do this by using the last code we created above:

activityCSV[na\_pos, "steps"] <- mean\_Rep

Lastly, we display the histogram:

sumData <- aggregate(activityCSV$steps, by=list(activityCSV$date), FUN=sum)  
  
names(sumData) <- c("date", "total")  
  
hist(sumData$total,   
 breaks=seq(from=0, to=25000, by=2500),  
 col="red",   
 xlab="Total Number of Steps",   
 ylim=c(0, 30),   
 main="Histogram of the Total Number of Steps Taken Each Day\n(NA Replaced by Mean Value)")



## 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 - "weekday" and "weekend" indicating whether a given date is a weekday or weekend day.
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).

We actually already made the new factgor variable:

head(activityCSV)

## date weekday daytype interval steps  
## 1 2012-10-01 monday weekday 0 37.3826  
## 2 2012-10-01 monday weekday 5 37.3826  
## 3 2012-10-01 monday weekday 10 37.3826  
## 4 2012-10-01 monday weekday 15 37.3826  
## 5 2012-10-01 monday weekday 20 37.3826  
## 6 2012-10-01 monday weekday 25 37.3826

To create the panel plot, we need to install the lattice package. I find the easiest way to do this is by going to the "Packages" icon on your Files, Plots, Packages, etc helper window. Top left is an "Install" option, select that and type in "lattice".

Once you have the new package installed, load the package using the library(lattice) command (you can ignore the warning that follows).

library(lattice)

## Warning: package 'lattice' was built under R version 3.2.3

Next, we can calculate the average number of steps taken by using the code:

meanData <- aggregate(activityCSV$steps,   
 by=list(activityCSV$daytype,   
 activityCSV$weekday, activityCSV$interval), mean)  
  
names(meanData) <- c("daytype", "weekday", "interval", "mean")

We check our work:

head(meanData)

## daytype weekday interval mean  
## 1 weekday friday 0 8.307244  
## 2 weekday monday 0 9.418355  
## 3 weekend saturday 0 4.672825  
## 4 weekend sunday 0 4.672825  
## 5 weekday thursday 0 9.375844  
## 6 weekday tuesday 0 0.000000

Finally, we make the plot:

xyplot(mean ~ interval | daytype, meanData,   
 type="l",   
 lwd=1,   
 xlab="Interval",   
 ylab="Number of Steps",   
 layout=c(1,2))

