Reproducible Research

Fernando Roque

6 de mayo de 2016

## Introduction

Nowdays get and collect data is easier. Today, we are going to talk about personal movement using activity monitoring devices such as a Fitbit, Nike Fuelband, or Jawbone Up. *We are being monitored!* Whatever, let's see some statistical discover about someone (maybe you).

## Data

The data for this assignment can be downloaded from the course web site [Data source](https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip).

The variables included in this dataset are:

1. **steps**: Number of steps taking in a 5-minute interval (missing values are coded as NA)
2. **date**: The date on which the measurement was taken in YYYY-MM-DD format
3. **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

Once we have dowloaded data from the web, the first step is loading data into workspace. We are using read.csv() function.

setwd("C:/Users/Fernando/Documents/Material/Coursera/DataScience/Reproducible\_Research")  
act <- read.csv("activity.csv", header=TRUE)

Let's see the first observations:

head(act)

## 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

Explore a little more.

str(act)

## '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 ...

For further analysis we need to transfor date variable in date format as follow.

act$date <- as.Date(act$date,"%Y-%m-%d")

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

*For this part of the assignment, you can ignore the missing values in the dataset.*

1. *Calculate the total number of steps taken per day*
2. *If you do not understand the difference between a histogram and a barplot, research the difference between them. Make a histogram of the total number of steps taken each day*
3. *Calculate and report the mean and median of the total number of steps taken per day*

We ignore nulls using is.na() function as follows.

rmna <- !is.na(act$steps)  
act1 <- act[rmna,]

To answer above questions, we manipulate data using dplyr library. First, we calculate the total number of steps taken per day using summarize(), and store the results in a new data frame called act\_stat

library(dplyr)

##   
## Attaching package: 'dplyr'

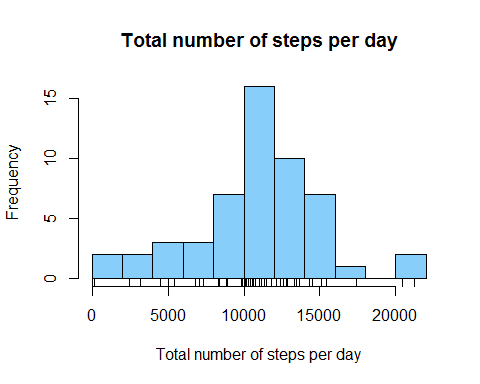
## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

act\_stat <- act1 %>% group\_by(date) %>% summarize(sum=sum(steps))

Whit this new data, we generate the histogram

hist(act\_stat$sum, nclass=15, col="lightskyblue", main="Total number of steps per day", xlab="Total number of steps per day")  
rug(act\_stat$sum)



Now, we calculate the mean and median of the total number of steps per day.

mean(act\_stat$sum)

## [1] 10766.19

median(act\_stat$sum)

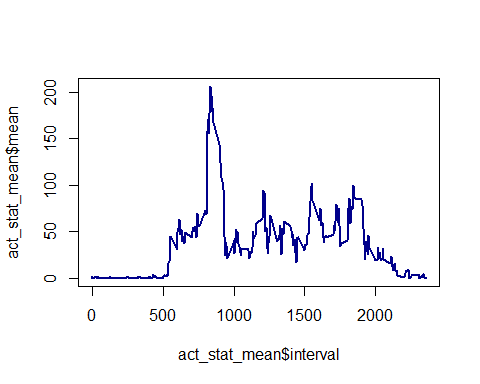
## [1] 10765

## What is the average daily activity pattern?

1. *Make 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 days (y-axis)*
2. *Which 5-minute interval, on average across all the days in the dataset, contains the maximum number of steps?*

Firts question will be aswered by calculate a new summary table

act\_stat\_mean <- act1 %>% group\_by(interval) %>% summarize(mean=mean(steps))  
plot(act\_stat\_mean$interval, act\_stat\_mean$mean, type="l", col="darkblue", lwd=2)



The maximun number of step is calculated as follow:

act\_stat\_mean[which.max(act\_stat\_mean$mean),]

## Source: local data frame [1 x 2]  
##   
## interval mean  
## (int) (dbl)  
## 1 835 206.1698

## 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.*

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. Do these values differ from the estimates from the first part of the assignment? What is the impact of imputing missing data on the estimates of the total daily number of steps?*

We return using act data frame. The number of missing values are:

sum(is.na(act$steps))

## [1] 2304

Impute data will be made using the mean for each 5-minute interval. I did this by creating a summary table that contains the mean and merge this results with the original data.

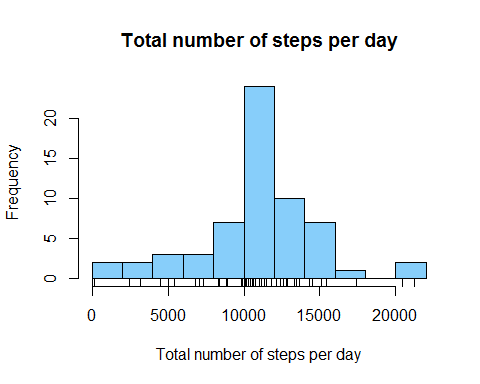
act\_stat1 <- act1 %>% group\_by(interval) %>% summarize(imput=round(mean(steps)))  
act\_mer <- merge(act, act\_stat1, by="interval")  
act\_mer <- act\_mer[order(act\_mer$date,act\_mer$interval),]

Finally we impute the data by replacing those for which data is NA according to the interval variable. The second line reorder the data.

act\_imp <- data.frame(date=act\_mer$date, interval=act\_mer$interval, steps=ifelse(is.na(act$steps), act\_mer$imput, act\_mer$steps ))

The new data act\_imp contains no more NA's! Using this data, we plot a new histogram and calculate the mean and median.

act\_imp\_sum <- act\_imp %>% group\_by(date) %>% summarize(sum=sum(steps))  
hist(act\_imp\_sum$sum, nclass=15, col="lightskyblue", main="Total number of steps per day", xlab="Total number of steps per day")  
rug(act\_imp\_sum$sum)



Finally, we calculate the mean and median of the data with imputed data

mean(act\_imp\_sum$sum)

## [1] 10765.64

median(act\_imp\_sum$sum)

## [1] 10762

Threre is not a significal change in the mean of the totla step per day, just a little change in the median.

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

1. *For this part the weekdays() function may be of some help here. Use the dataset with the filled-in missing values for this part.*
2. *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.* *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). See the README file in the GitHub repository to see an example of what this plot should look like using simulated data.*

First, we create the variable wd, which represents the level required.

act\_wd <- ifelse(weekdays(act\_imp$date)=="sábado" | weekdays(act\_imp$date)=="domingo" ,"Weekend","Weekday")  
act\_imp\_wd <- data.frame(date=act\_imp$date, interval=act\_imp$interval, steps=act\_imp$steps, wd=act\_wd)  
act\_imp\_wd$wd <- factor(act\_imp\_wd$wd)

Using this new variable, we creat a summary data.

act\_wd\_sm <- act\_imp\_wd %>% group\_by(interval, wd) %>% summarize(sum=sum(steps))

In the last step, create the plot using lattice package.

library(lattice)  
xyplot(sum ~ interval | wd, act\_wd\_sm, type = "l", layout = c(1, 2), xlab = "Interval", ylab = "Number of steps")

