Peer Assesement 1

#### This is the document that describes my work for Coursera's Reproductible Research, Assesement 1

Autor:FSC

#### \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#### 

#### 1.- Project creation and data reading and processing

The data file for the assignment has been downloaded to my download folder. The code segment below will use ProjectTemplate library to create a set if folders for this project, copy the file to the data folder and unzip it, then it reads the data into 'repdata' object

library(ProjectTemplate)  
library(plyr)  
library(dplyr)  
library(tcltk)  
library(ggplot2)  
library(lattice)  
library(lubridate)

WorkDir <- "/Volumes/LION-DATA/COURSERA /DATA SCIENTIST/REPRODUCIBLE RESEARCH" # work dir has to be modified to fit your system. The rest pends from it, so it should work  
setwd(WorkDir)  
  
# check for Peer\_Assessement\_1 folder and create ir if it doesn´t exist.  
if(!file.exists("Peer\_Assessment\_1")) create.project("Peer\_Assessment\_1")  
setwd("Peer\_Assessment\_1")  
  
# Copying a file from a dir to another dir in case ir hasn't been copied before  
from <-"/users/Fran/Downloads/repdata-data-activity.zip"  
to <- "./data/repdata-data-activity.zip"  
if(!file.exists(to)) file.copy(from, to)  
   
setwd("./data") # changing to data folder and unzip if neccesary  
if(!file.exists("activity.csv")) unzip("repdata-data-activity.zip")  
activity <- tbl\_df(read.csv("activity.csv"))

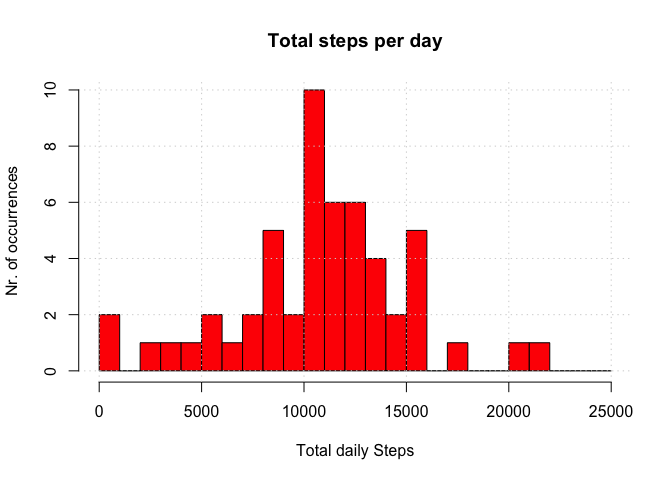
#### \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#### 

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

The data frame is cleaned by filtering NA's out, then grouped by date and summarize( sum) the total steps into the steps object. A histogram with bins=1000 steps is shown in wich we can see that the mean value of steps in a day is around 11,000 steps.

steps <- activity %>%  
 filter(steps != "NA") %>% # filtering out rows with NA's on steps  
 group\_by(date) %>% # grouping by date  
 summarise( daySteps = sum(steps)) # and summing the steps by groupped date.  
# histogram   
hist(steps$daySteps,col="red", breaks =seq(0,25000, by = 1000), main = "Total steps per day",, xlab= ("Total daily Steps"), ylab= ("Nr. of occurrences"))  
 grid() # adding a grid



#### Mean and median

options(scipen=999)  
options(digits = 2)  
mean <- summarize(steps, mean(daySteps))  
median <- summarize(steps, median(daySteps))  
mean

## Source: local data frame [1 x 1]  
##   
## mean(daySteps)  
## 1 10766

median

## Source: local data frame [1 x 1]  
##   
## median(daySteps)  
## 1 10765

The calculated mean and median ( code above) is:

- Mean: 10766.19   
- Median: 10765

So we see that the distribution shows mean symetry ( mean = median).

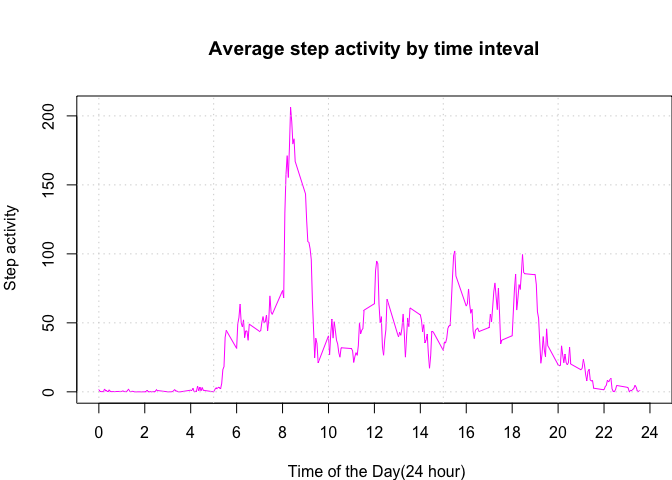
#### \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#### 

#### 3.- What is the average daily activity pattern?

Defining trim\_activity as the activity trimmed from NA's, groupped by the interval of the day and summarize the mean of each interval, then plotting it.

trim\_activity <- activity %>%  
 filter(steps != "NA") %>% # filtering out rows with NA's on steps  
 group\_by(interval) %>%  
 summarise( intSteps = mean(steps))  
  
# plot code. The x axis( interval) is divided by 100 to set hourly numbers on the x-axis.  
with(trim\_activity, {  
 par(mar = c(4,4,5,0))  
 plot( interval/100, intSteps,type="l", xlab=" Time of the Day(24 hour)", ylab=" Step activity", ylim = range(intSteps), col ="magenta", xlim = range(0:24), xaxt="n")   
 axis(1, at=seq(0,24, by =2))   
 title(main = "Average step activity by time inteval")  
 grid()  
})



Then search for the max of the day...

maxActivity <- summarise(trim\_activity, maxSteps = max(intSteps), interval = substr(hm(interval[intSteps ==maxSteps]/100),1,6))  
maxActivity

## Source: local data frame [1 x 2]  
##   
## maxSteps interval  
## 1 206 8H 35M

which is an average of 206.17 steps at 8H 35M time interval.

#### \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#### 

#### 4.- Imputing missing values

a.- Calculating the number of missing values...

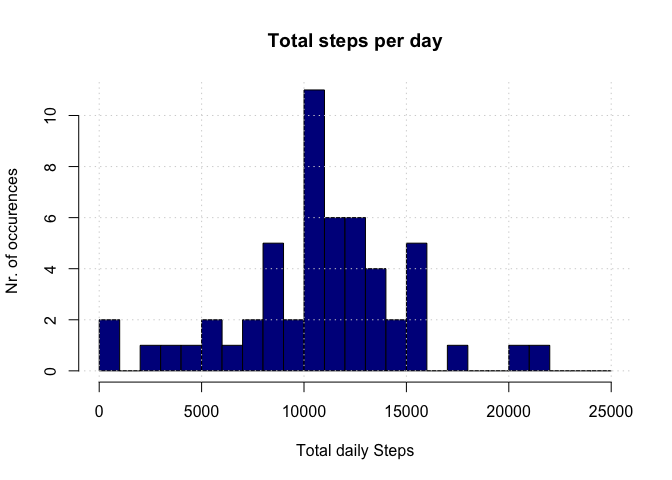
dimTot <-dim(activity)  
  
act\_nas <-activity %>%  
 filter(is.na(steps))  
  
dimNas<-dim(act\_nas)  
dimNas[1]

## [1] 2304

... and, there are 2304 NA's on a total of 17568 records; or about 13%

Now creating a dataset with the NA's filled with the average value for that interval. We'll take the average from trim\_activity that is groupped by interval and sumarized by it. So trim\_activity holds the average value for each 5 minute period of the day.

corrected <- activity %>%  
 mutate(steps = ifelse(is.na(steps), trim\_activity[trim\_activity$interval == interval,]$intSteps, steps))%>% # ifelse() inserts on 'steps' the average value if the original value is NA  
 filter(steps != "NA") # filtering out rows with NA's on steps ( that are redundant since we inseted the averages)  
  
newSteps <- corrected %>% # now 'corrected' holds valid values, so we do the same as in the 1st histogram case   
 group\_by(date) %>% # grouping by date  
 summarise( daySteps = sum(steps)) # and summing the steps by groupped date.  
  
# histogram   
hist(newSteps$daySteps,col="darkblue", breaks =seq(0,25000, by = 1000), main = "Total steps per day",, xlab= ("Total daily Steps"), ylab= ("Nr. of occurences"))  
 grid()



Then, for the mean and median, we take the percent difference between the old value and the new one.

newmean <- summarize(newSteps, mean(daySteps))  
newmedian <- summarize(newSteps, median(daySteps))  
  
diff\_median <- 100\*abs(median-newmedian)/median # multiplied by 100 to get a percent difference  
diff\_mean <- 100\*abs(mean-newmean)/mean  
diff\_median

## median(daySteps)  
## 1 0.0055

diff\_mean

## mean(daySteps)  
## 1 0

We can see that inserting the average value of the interval into NA values it virtually doesn't modify the chacracteristics of the distribution, since the mean is unaffected (obvious, since we have introduced mean values) and the median is affected by only 0.01%, a negligible difference.

The only concern is about stationality ( weekends vs non weekends), but it should be negligible too, since we deal wih a small change on a 0.01% difference.

#### \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

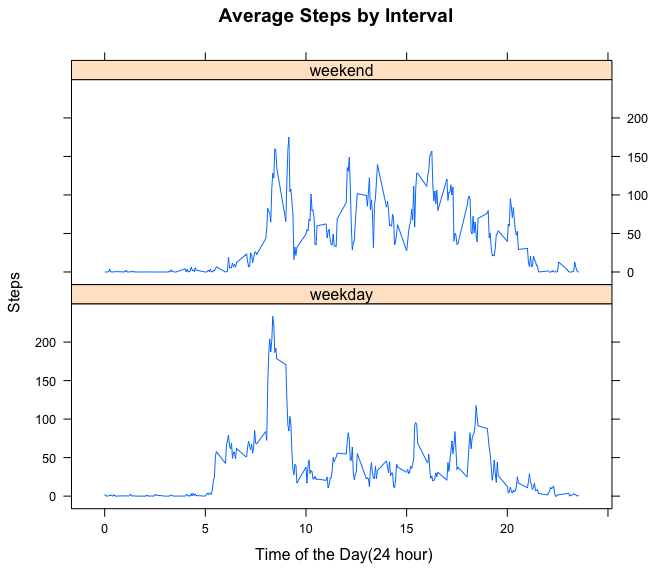
#### 

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

Below the code that try to answer the question. We add a new category 'day', with values 'weekday' or 'weekend', deppending on the day of the week.

Then we group and summarise by interval as before.  
The plotting code define a dual pannel that plots the subset group 'weekend' on the upper plot and the subset group 'weekdays' on the lower one.

weekactivity <- corrected %>%  
 mutate( day = ifelse(wday(ymd(date))==7 | wday(ymd(date))==1, "weekend", "weekday" )) %>% ## adding a day category, then grouping and summarizing  
 group\_by(day, interval) %>%  
 summarise( intSteps = mean(steps))  
   
par(mfrow = c(2, 1)) # define a dual pannel, 2 rows, by 1.  
par(mar = c(4,4,2,0)) # defining margins.  
  
xyplot(intSteps ~ interval/100 | day, data = weekactivity,  
 layout = c(1, 2),xlab=" Time of the Day(24 hour)", ylab=" Steps", type = "l",  
 main = "Average Steps by Interval")



So the short answer is YES; there are different patterns.

We can observe a different pattern on weekends from weekdays, as might be expected. Where on weekdays we see an peak activity from 8 to 9:30 and then a plateau, on weekend days we see with roughly evenly distributed peaks from 8 to 20, so it seems the subject has a sedentary job but seems to enjoy sporty weekend days.