02week_course_project_1

Fernando Villalba

16 de marzo de 2017

WEEK 2 Peer-graded Assignment: Course Project 1

In this document we show the answer to the assignment questions.

First of all, i have some dauts about the data origin,

Code for reading in the dataset and/or processing the data

Read data from website:

```
    library(httr)

    url <- "https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip"
    file <- "repdata_data_activity.zip"

#donwload file in PC
    download.file(url, file, method = "wininet")

# Unzip file
    unzip(file, list = FALSE, overwrite = TRUE)
```

Read .csv file into data frame call *datos*, and clean NA in *datos_limpios*:

```
# read csv
    datos <-read.csv("activity.csv")
    datos_limpios<- na.omit(datos)
# summary
    str(datos)
```

```
## '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 ...
```

```
    range(datos$interval)
```

```
## [1]    0 2355
```

```
    #unique(datos$date)
```

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 o hour for the 5-minute interval in which measurement was taken.
```

Histogram of the total number of steps taken each day

To show the frecuency of the number of steps in a day, aggregate data by date (days) and sum

```
# calculate num steps by day
res_day<-aggregate(steps~date,datos_limpios,sum)
png('imag/plothist.png')
    hist(res_day$steps,col="tomato",
         main= "Total frecuency. Steps in a day",
         xlab="number of steps in a day" )
    rug(res_day$steps)
dev.off()
```

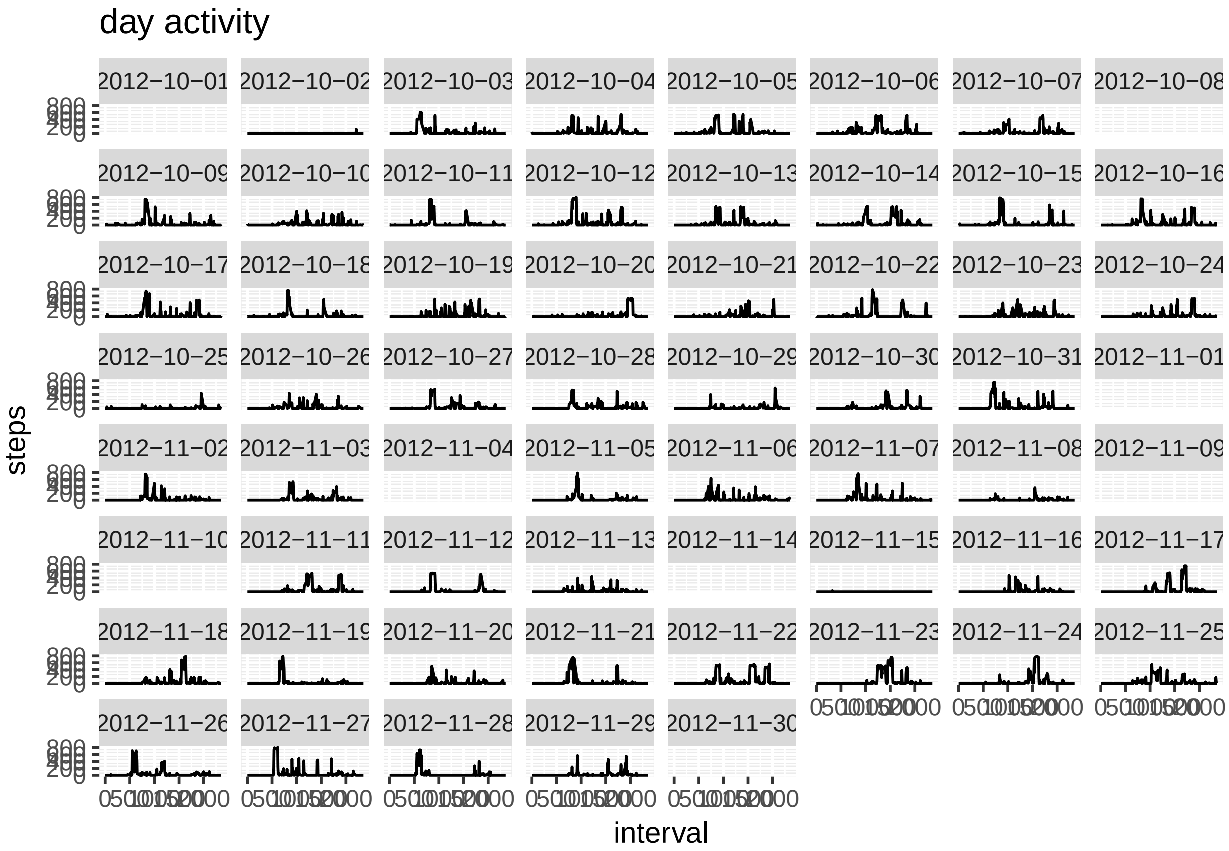
```
## pdf
##   2
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.3.2
```

```
# an other way of seen activity
# total activity day by day
    qplot(interval,steps,data=datos,
          geom="line",main= "day activity",
          facets = ~ date)
```

```
## Warning: Removed 576 rows containing missing values (geom_path).
```



```
#datos2<-subset(datos, steps>0)
#hist(datos2$steps)
```

Mean and median number of steps taken each day

With summary we obtein this numbers. The mean number of steps taken each day is Mean :10766 and the median number of steps is Median :10765

Time series plot of the average number of steps taken

First we plot a time series of the sum of steps taken each day during the test days.

```
    #plot(res_day$steps,type = "h", main= "Total steps by day")

    adias<- as.character((levels(res_day$date)))
    adias<-strptime(adias, "%Y-%m-%d")

    plot(res_day$steps,type = "l",col="tomato1",axes=FALSE,cex.lab = 0.8,
         main= "Time series of steps by day",xlab="day/month",ylab="total steps/day"
         )
    axis(1,labels=format(adias, "%d/%m"), at=1:length(levels(res_day$date)),col="gray",cex.axis = 0.6)
    axis(2,las=1,col="gray",cex.axis = 0.7)
```

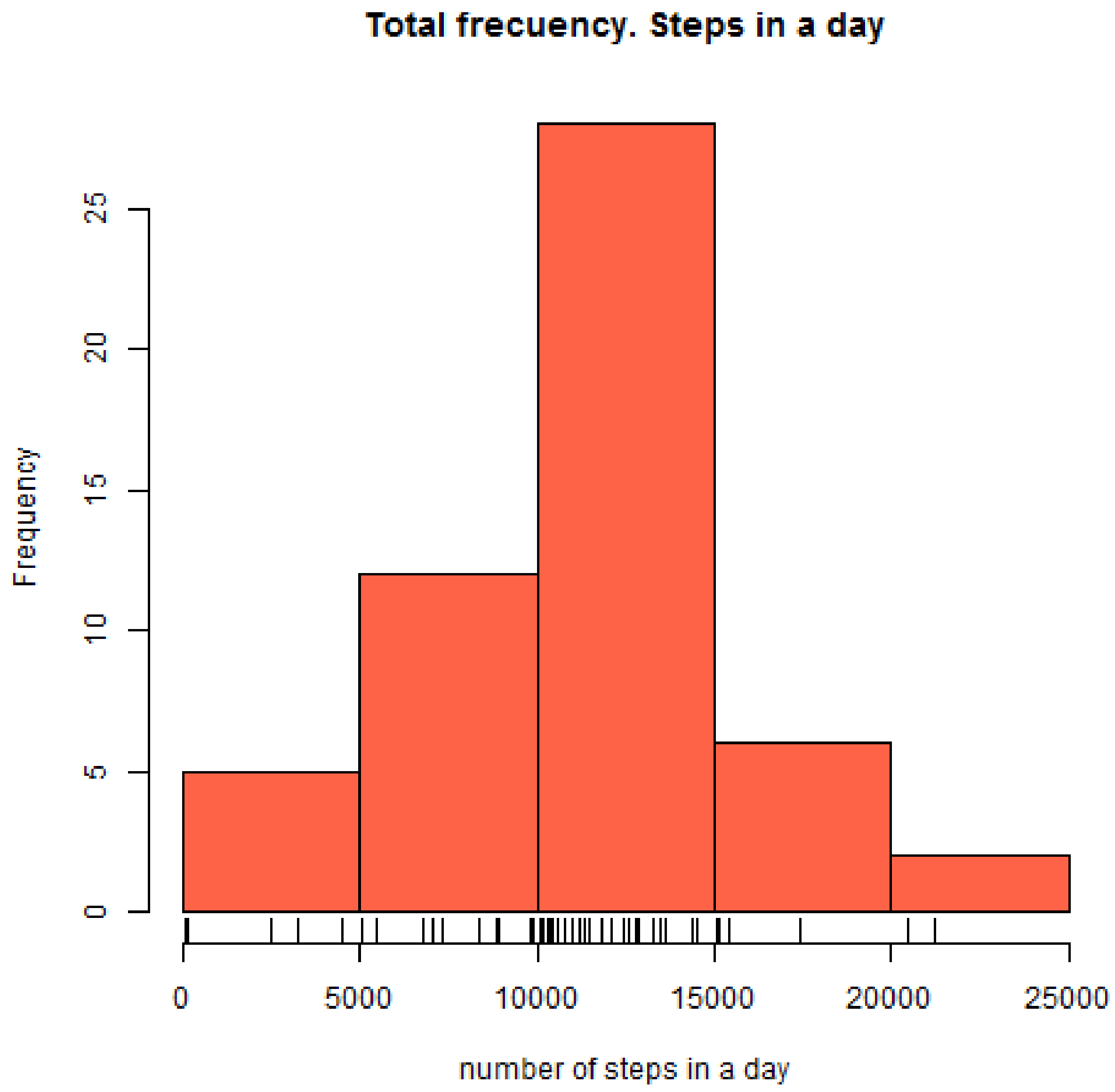
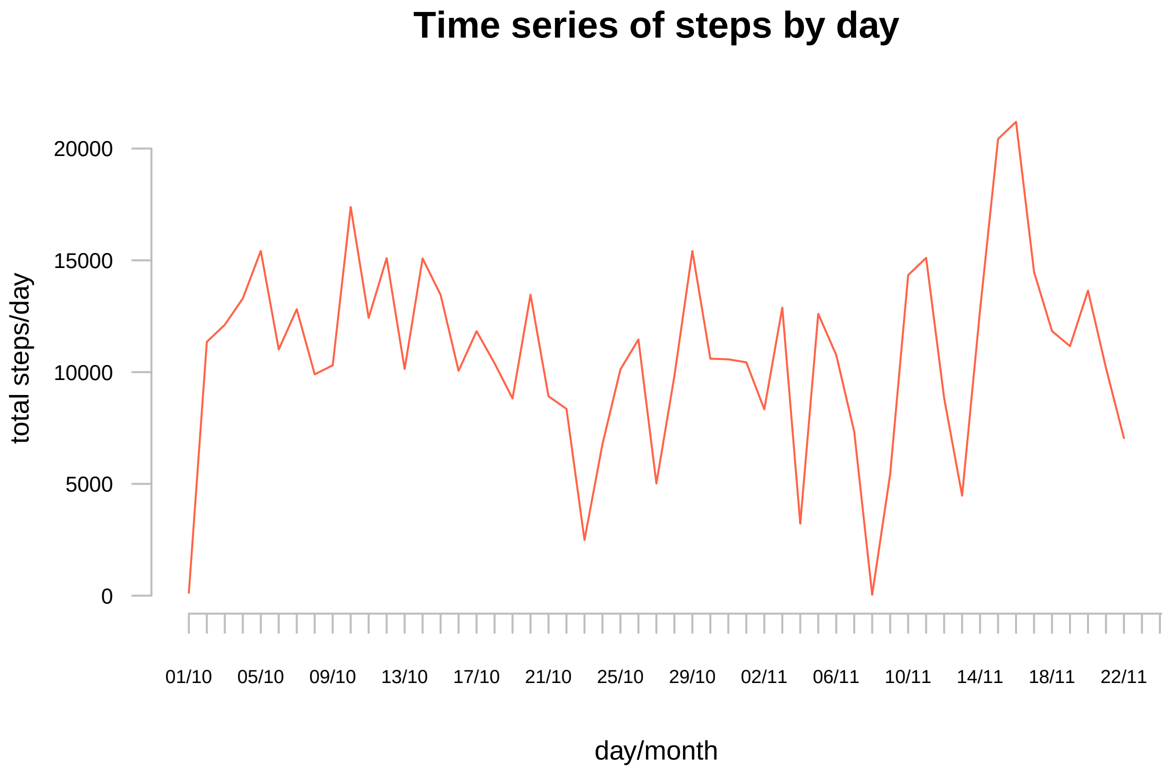


Figure 1: Total frecuency. Steps in a day



But we want to resume a normal day, so we take the mean of steps each interval and plot the result:

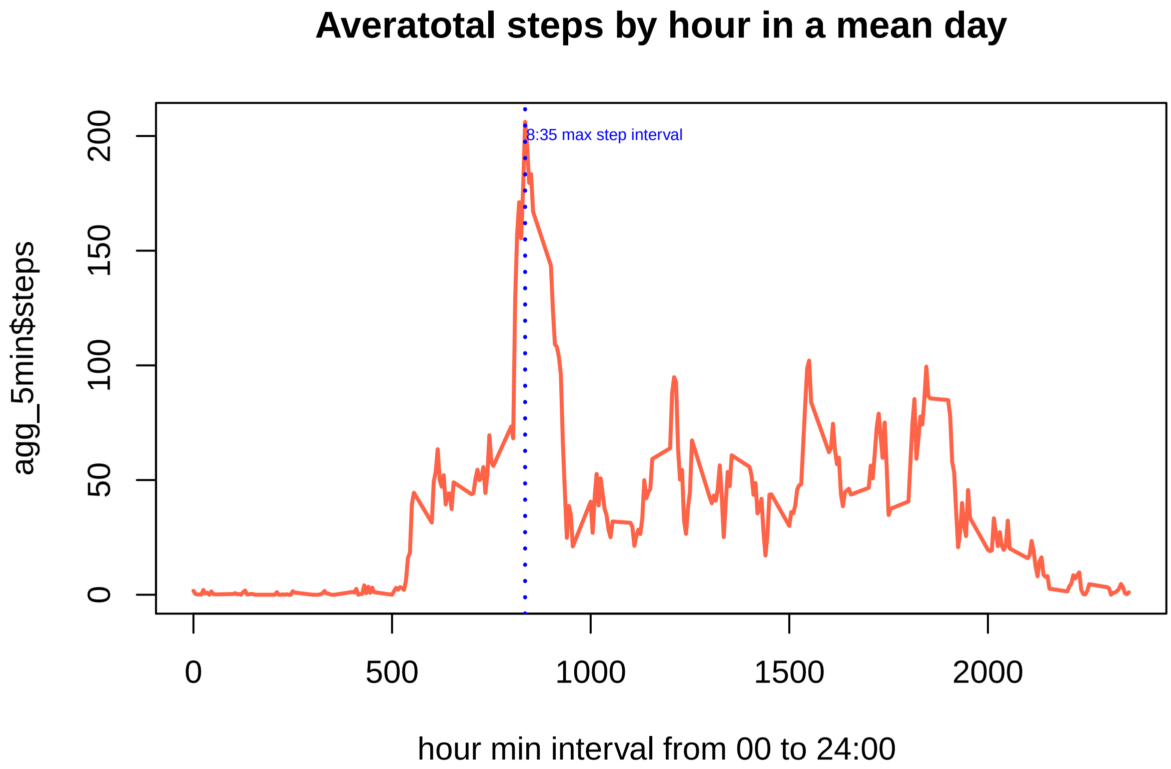
```
# calculate aggregate by 5 min interval
agg_5min<-aggregate(steps~interval,datos_limpios,mean)

# max interval
max(agg_5min$steps)
```

```
## [1] 206.1698
```

```
maxInterval <- agg_5min[ agg_5min$step == max( agg_5min$step ), 'interval']

# Plot
plot(agg_5min$interval,agg_5min$steps,type = "l",col="tomato1", lwd=2,
     main= "Averatotal steps by hour in a mean day", xlab="hour min interval from 00 to 24:00 ")
#add vertical line in max interval
abline(v=maxInterval, col="blue",lwd=2, lty=3)
text(x=maxInterval+200, y=200, label="8:35 max step interval", cex = 0.5, col="blue")
```



The 5-minute interval that, on average, contains the maximum number of steps

As put on the last plot, the max interval is at 835 (08:35)

Code to describe and show a strategy for imputing missing data

First calculate the number of missing data

```
num_NA <- sum(is.na(datos))
#Total number o f NA in data frame
num_NA
```

```
## [1] 2304
```

```
# Total number of NA by day
num_NA<-split(datos$steps,datos$date)
na_count <-sapply(num_NA, function(y) sum(length(which(is.na(y)))))
na_count <- data.frame(na_count)

library(xtable)
print(xtable(na_count), type="latex", comment=FALSE)
```

```
## \begin{table}[ht]
## \centering
## \begin{tabular}{rr}
##   \hline
##  & na\_count \\
##   \hline
## 2012-10-01 & 288 \\
##   2012-10-02 &   0 \\
##   2012-10-03 &   0 \\
##   2012-10-04 &   0 \\
##   2012-10-05 &   0 \\
##   2012-10-06 &   0 \\
##   2012-10-07 &   0 \\
##   2012-10-08 & 288 \\
##   2012-10-09 &   0 \\
##   2012-10-10 &   0 \\
##   2012-10-11 &   0 \\
##   2012-10-12 &   0 \\
##   2012-10-13 &   0 \\
##   2012-10-14 &   0 \\
##   2012-10-15 &   0 \\
##   2012-10-16 &   0 \\
##   2012-10-17 &   0 \\
##   2012-10-18 &   0 \\
##   2012-10-19 &   0 \\
##   2012-10-20 &   0 \\
##   2012-10-21 &   0 \\
##   2012-10-22 &   0 \\
##   2012-10-23 &   0 \\
##   2012-10-24 &   0 \\
##   2012-10-25 &   0 \\
##   2012-10-26 &   0 \\
##   2012-10-27 &   0 \\
##   2012-10-28 &   0 \\
##   2012-10-29 &   0 \\
##   2012-10-30 &   0 \\
##   2012-10-31 &   0 \\
##   2012-11-01 & 288 \\
##   2012-11-02 &   0 \\
##   2012-11-03 &   0 \\
##   2012-11-04 & 288 \\
##   2012-11-05 &   0 \\
##   2012-11-06 &   0 \\
##   2012-11-07 &   0 \\
##   2012-11-08 &   0 \\
##   2012-11-09 & 288 \\
##   2012-11-10 & 288 \\
##   2012-11-11 &   0 \\
##   2012-11-12 &   0 \\
##   2012-11-13 &   0 \\
##   2012-11-14 & 288 \\
##   2012-11-15 &   0 \\
##   2012-11-16 &   0 \\
##   2012-11-17 &   0 \\
##   2012-11-18 &   0 \\
##   2012-11-19 &   0 \\
##   2012-11-20 &   0 \\
##   2012-11-21 &   0 \\
##   2012-11-22 &   0 \\
##   2012-11-23 &   0 \\
##   2012-11-24 &   0 \\
##   2012-11-25 &   0 \\
##   2012-11-26 &   0 \\
##   2012-11-27 &   0 \\
##   2012-11-28 &   0 \\
##   2012-11-29 &   0 \\
##   2012-11-30 & 288 \\
##    \hline
## \end{tabular}
## \end{table}
```

As we see in last table, the problem is that we found completed missing days of data.

One simple option to complete this missing days is assing the average histogram for this lost data days.

```r
# function for calculate mean average by interval
getmedia<-function(interval){
    agg_5min[agg_5min$interval==interval,]$steps
}

#getmedia(835)


# clon dataframe
impute_datos<-datos
# select NA steps --> interval values
impute_datos<-impute_datos[is.na(impute_datos$steps),c('interval')]
# pass function to this select data
impute_datos1<-sapply(impute_datos,getmedia)
#change data NA with average in impute data frame
impute_datos<-datos
impute_datos[is.na(impute_datos$steps),c('steps')]<-impute_datos1


## for(i in 1:nrow(impute_datos)){
##    if(is.na(impute_datos[i,]$steps)){
##        impute_datos[i,]$steps <- getmedia(impute_datos[i,]$interval)
##    }
## }
```

Histogram of the total number of steps taken each day after missing values are imputed

```r
    res_dayNoNA<-aggregate(steps~date,impute_datos,sum)
    png('imag/plothistnoNA.png')
        hist(res_dayNoNA$steps,col="wheat",
             main= "Total frecuency. Steps in a day",
             xlab="number of steps in a day" )
        rug(res_dayNoNA$steps)
    dev.off()
```

```
## pdf
##   2
```

```
summary(res_dayNoNA)
```

```
##          date        steps
##  2012-10-01: 1   Min.   :   41
##  2012-10-02: 1   1st Qu.: 9819
##  2012-10-03: 1   Median :10766
##  2012-10-04: 1   Mean   :10766
##  2012-10-05: 1   3rd Qu.:12811
##  2012-10-06: 1   Max.   :21194
##  (Other)   :55
```

it seems no significant changes appear.

Panel plot comparing the average number of steps taken per 5-minute interval across weekdays and weekends

```
    datos$diasem<-c("weekday")
    datos[weekdays(as.Date(datos[, 2])) %in% c("sábado", "domingo"), ]$diasem <- c("weekend")
    # to count how many weekend days
    table(datos$diasem == "weekend")
```

```
##
## FALSE  TRUE
## 12960  4608
```

We decided not to chose impute data.

```
    # calculate aggregate by 5 min interval
    agg_5min<-aggregate(steps~interval+diasem,datos,mean)
    png('imag/weekend.png')
        qplot(interval,steps,data = agg_5min,
              geom=c("area","line"),xlab="hour/interval",
              facets = diasem ~.,
              fill=diasem)
    dev.off()
```

```
## pdf
##   2
```

it seem in weekend we start later to walk

All of the R code needed to reproduce the results (numbers, plots, etc.) in the report

```
library(dplyr)

activityData <- read.csv("activity.csv")
cleanData <- na.omit(activityData)


dailyTotals <- cleanData %>%
               group_by(date) %>%
               summarize(number_of_steps_per_day = sum(steps)) %>%
               na.omit()

hist(dailyTotals$number_of_steps_per_day)

report <- dailyTotals %>% summarize(mean   = mean(number_of_steps_per_day),
                                    median = median(number_of_steps_per_day))

print(report)

#What is the average daily activity pattern?

timeSeries <- cleanData %>%
              group_by(interval) %>%
              summarize(average = mean(steps))

p <- ggplot(timeSeries, aes(interval, average)) +
     geom_line() +
     ylab("Steps Average") +
     xlab("Daily Intervals")

print(p)
```