My assignment for reproducible research

This assignment makes use of data from a personal activity 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.

# Loading the data

First, we fetch the dataset from the web and load it into our R environment.

directory <- "figure"  
  
if(!file.exists(directory)) {  
 dir.create(directory)  
}  
  
setwd(directory)  
dir()

## [1] "activity.csv" "activity.zip"

url <- "https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip"  
  
# Keep track of download date  
download.file(url, "activity.zip")  
#date.download <- date()  
  
# Unzip the file and read it  
name.file <- unzip(zipfile = "activity.zip", list = TRUE)$Name  
unzip(zipfile = "activity.zip", name.file)  
df <- read.csv(name.file, colClasses = c("integer", "Date", "integer"))

dim(df)

## [1] 17568 3

Our dataset contains 17.568 rows and 3 columns. Once data are loaded we are interested in getting a general overview of the data.

# Observing the data

## General overview

summary(df)

## steps date interval   
## Min. : 0.00 Min. :2012-10-01 Min. : 0.0   
## 1st Qu.: 0.00 1st Qu.:2012-10-16 1st Qu.: 588.8   
## Median : 0.00 Median :2012-10-31 Median :1177.5   
## Mean : 37.38 Mean :2012-10-31 Mean :1177.5   
## 3rd Qu.: 12.00 3rd Qu.:2012-11-15 3rd Qu.:1766.2   
## Max. :806.00 Max. :2012-11-30 Max. :2355.0   
## NA's :2304

We have here 2 month of monitoring data from 2012-10-01 to 2012-11-30. The steps variable contains NA values.

## Activity overview

We want to observe the total number of steps taken each day by the user and out of this extract the mean and the median.

First, we plot an histogram with the number of steps per day.

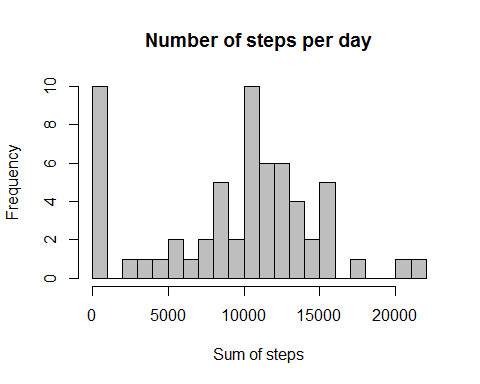
library(dplyr)

##   
## 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

# Sum the number of steps per day  
sum.df <- df %>%  
 group\_by(date) %>%  
 summarise(sum.steps = sum(steps, na.rm = TRUE))  
  
# Plot the distribution  
hist(sum.df$sum.steps, breaks = 20, col = "grey", main = "Number of steps per day", xlab = "Sum of steps")



The sample distribution seems to be bimodal with one pic at 0 and one pic at about 10000 steps. Furthermore, we can identify that there are some outliers. Then we extract the mean and the median from the distribution.

# Calculating the median and mean for the sum of steps made each day by the user  
median.df <- with(sum.df, median(sum.steps))  
mean.df <- with(sum.df, mean(sum.steps))  
  
# Printing out  
median.df

## [1] 10395

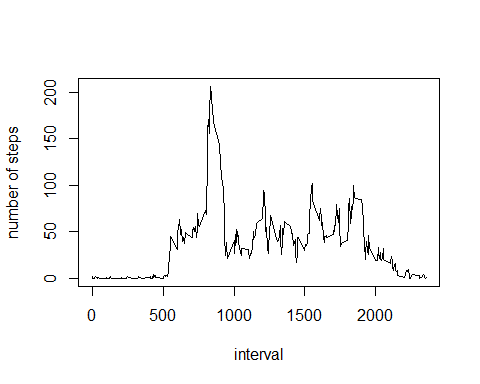
mean.df

## [1] 9354.23

Mean < Median meaning that the distribution is left skewed (as we could see in the histogram) which is certainly due to the high number of outliers with a number of steps equal to 0.

Then we want to create a Time series plot of the average number of steps taken each day. **Each day has 471 intervals of 5 minutes for a total of 2355 minutes. It is weird to me since there are 1440 total minutes in a day.**

library(dplyr)  
# stack each day and calculate the mean for each interval  
ts.df <- df %>%  
 group\_by(interval) %>%  
 summarise(mean.steps = mean(steps, na.rm = TRUE))  
  
plot(x = ts.df$interval, y = ts.df$mean.steps, xlab = "interval", ylab = "number of steps", type = 'l')



#Calculating the max number of steps  
ts.df.max <- ts.df[which(ts.df$mean.steps == max(ts.df$mean.steps)),]  
ts.df.max

## # A tibble: 1 × 2  
## interval mean.steps  
## <int> <dbl>  
## 1 835 206.1698

The maximun averaged number of steps per day is obtained for the interval 830 to 835.

## Imputing missing values

When we looked at the summary, we saw that some values were missing.

# summarise missing values  
any(is.na(df))

## [1] TRUE

colSums(is.na(df))

## steps date interval   
## 2304 0 0

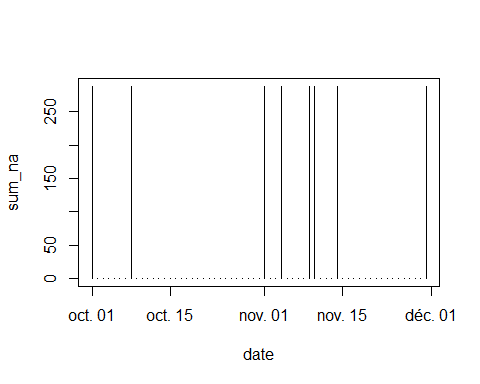
We want to see when those NAs appear by plotting the sum of NAs over time.

library(lubridate)

##   
## Attaching package: 'lubridate'

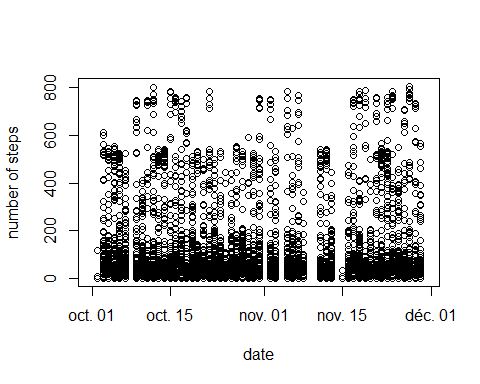
## The following object is masked from 'package:base':  
##   
## date

library(dplyr)  
  
# Summarizing the sum of NAs for the steps variable  
nas <- df %>%   
 group\_by(date) %>%  
 summarise(sum\_na = sum(is.na(steps)))  
  
with(nas, plot(date, sum\_na, type = "h"))



NA's don't seem to be generated over a particular pattern. Missing values appear toguether and seem to correspond to an entire day. We do a confirmation by printing the number of steps over time.

plot(x = df$date, y = df$steps, xlab = "date", ylab = "number of steps")

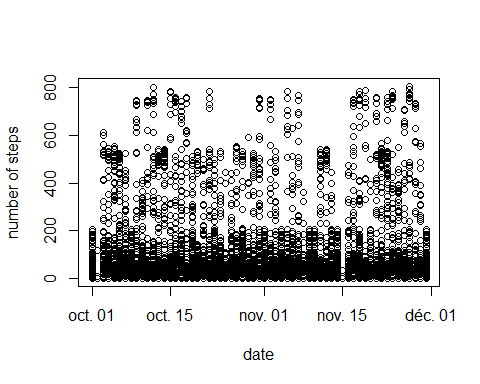


As we can see some days aren't monitored at all and some others only have a few values. Nevertheless it doesn't seem to be due to a seasonality pattern within the data. It seems to appear randomly.

We decide to impute missing values using the mean value that better fits the distribution. Indeed, since the distribution is right skewed, imputing by the median in that case would underestimate the standard deviation and distord the distribution for this particular variable.

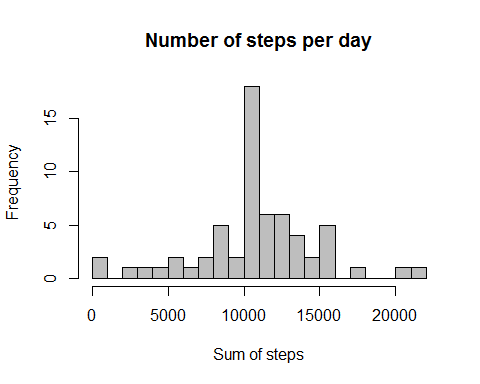
Since the variable is continuous, we keep track of where missing values appeared in the dataset by adding a logical column.

library(dplyr)  
  
# Add a column containing information about the presence of missing values  
df$was.missing <- is.na(df$steps)  
  
# Replace NA values by the mean  
impute.mean <- function(x) {   
 replace(x, is.na(x), mean(x, na.rm = TRUE))  
}  
  
# replace each missing value by the mean observed for the same interval  
df.imputed <- df %>%  
 group\_by(interval) %>%  
 mutate(  
 steps = impute.mean(steps)  
 )  
  
plot(x = df.imputed$date, y = df.imputed$steps, xlab = "date", ylab = "number of steps")



We confirm that missing days are filled in and so that missing values have been imputed.

library(dplyr)  
  
# Summarizing and plotting the sum of steps per days  
sum.df.imputed <- df.imputed %>%  
 group\_by(date) %>%  
 summarise(sum.steps = sum(steps, na.rm = TRUE))  
  
hist(sum.df.imputed$sum.steps, breaks = 20, col = "grey", main = "Number of steps per day", xlab = "Sum of steps")



As expected, the number of values at 0 drops and individuals move towards the center of mass of the histogram. We confirm this information by calculating the mean and median of the distribution.

# Calculating the median and mean for the sum of steps made each day by the user  
median.df.imputed <- with(sum.df.imputed, median(sum.steps))  
mean.df.imputed <- with(sum.df.imputed, mean(sum.steps))  
  
# Printing out  
median.df.imputed

## [1] 10766.19

mean.df.imputed

## [1] 10766.19

Imputing missing values with the mean centers the distribution. Now the median and the mean are equal so that the individuals are equally distributed.

## Seasonality analysis

In this part, we want to see whether there is a difference in activity between weekdays and the weekends.

# Add information on the day of week  
df.imputed$day.of.week <- weekdays(df.imputed$date)  
df.imputed$is.weekend <- with(df.imputed, as.factor(ifelse(day.of.week == "samedi" | day.of.week == "dimanche", "weekend", "weekday")))

Then, we want to observe whether users walk more over weekends or not

library(ggplot2)  
library(dplyr)  
  
# Summarizing the average number of steps  
ts.df.imputed <- df.imputed %>%  
 group\_by(is.weekend, interval) %>%  
 summarise(mean.steps = mean(steps, na.rm = TRUE))  
  
# Plotting the number of steps along the days depending on the nature of the week's days (week or weekends)  
ggplot(ts.df.imputed, aes(x = interval, y = mean.steps, group = is.weekend, colour = is.weekend)) +  
 geom\_line() +  
 geom\_point() +  
 ylab("Average steps in one day") +  
 xlab("One day (5 minutes interval)") +  
 ggtitle("Differences in walks between week and weekends")

