Reproducible Research Assignment 1

Geons

Sunday, January 10, 2016

## Preparation :

# Clean up before start / tabula rasa (optional)  
rm(list=ls())  
  
# Load needed libraries - can be done later as well   
library(knitr)  
library(plyr)  
library(ggplot2)  
library(lattice)  
  
# check the workdir and reset if needed  
getwd()

## [1] "E:/reproducible\_research/assignment\_1"

setwd("e:/reproducible\_research/assignment\_1")

## Loading and preprocessing the data

We will load the data to initiate processing ##### 1. Unzip if necessary and read the data

if(!file.exists('activity.csv')){  
 unzip('activity.zip')  
}  
  
data <- read.csv("activity.csv", colClasses = c("integer", "Date", "factor"))

##### 2. Reformat the Date column & eliminate NA's

data$month <- as.numeric(format(data$date, "%m"))  
  
cleanData <- na.omit(data)

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

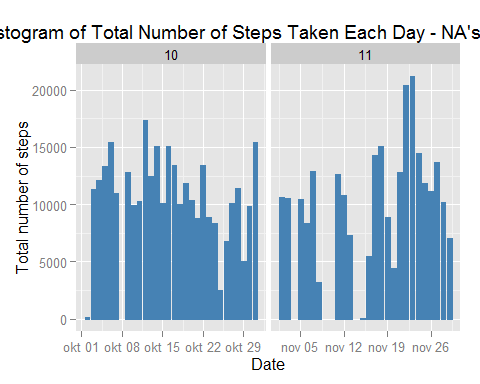
For this part of the assignment, you can ignore the missing values in the dataset.

##### 1. Make a histogram of the total number of steps taken each day

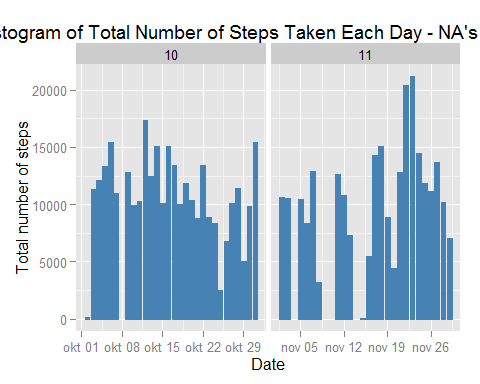
# First plot with NA's  
ggplot(data, aes(date, steps)) +   
 geom\_bar(stat = "identity", colour = "steelblue", fill = "steelblue", width = 0.8) +   
 facet\_grid(. ~ month, scales = "free") +   
 labs(title = "Histogram of Total Number of Steps Taken Each Day - NA's included", x = "Date", y = "Total number of steps")

## Warning: Removed 576 rows containing missing values (position\_stack).

## Warning: Removed 1728 rows containing missing values (position\_stack).



# Second plot without NA's - out of curiosity ...  
ggplot(cleanData, aes(date, steps)) +   
 geom\_bar(stat = "identity", colour = "steelblue", fill = "steelblue", width = 0.8) +   
 facet\_grid(. ~ month, scales = "free") +   
 labs(title = "Histogram of Total Number of Steps Taken Each Day - NA's excluded", x = "Date", y = "Total number of steps")



##### 2. Calculate and report the mean and median total number of steps taken per day

The mean total number of steps taken each day is :

dailytotalSteps <- aggregate(cleanData$steps, list(Date = cleanData$date), FUN = "sum")$x  
mean(dailytotalSteps)

## [1] 10766.19

Median total number of steps taken per day:

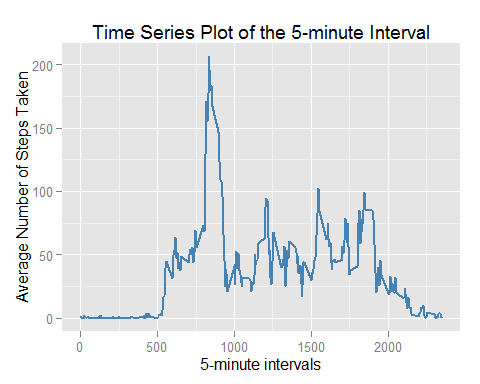
median(dailytotalSteps)

## [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)

avgSteps <- aggregate(cleanData$steps, list(interval = as.numeric(as.character(cleanData$interval))), FUN = "mean")  
names(avgSteps)[2] <- "meanOfSteps"  
  
ggplot(avgSteps, aes(interval, meanOfSteps)) +   
 geom\_line(color = "steelblue", size = 0.8) +   
 labs(title = "Time Series Plot of the 5-minute Interval", x = "5-minute intervals", y = "Average Number of Steps Taken")



##### 2. Which 5-minute interval, on average across all the days in the dataset, contains the maximum number of steps?

The 5-minute interval that contains on average the most steps is :

avgSteps[avgSteps$meanOfSteps == max(avgSteps$meanOfSteps), ]

## interval meanOfSteps  
## 104 835 206.1698

## Imputing missing values

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) The total number of rows with NAs:

sum(is.na(data)) # or sum(!complete.cases(data)) would also work

## [1] 2304

##### 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.

I chose to use the median for that 5-minute interval to fill each NA value in the steps column. So first i need to create an overview of the median steps

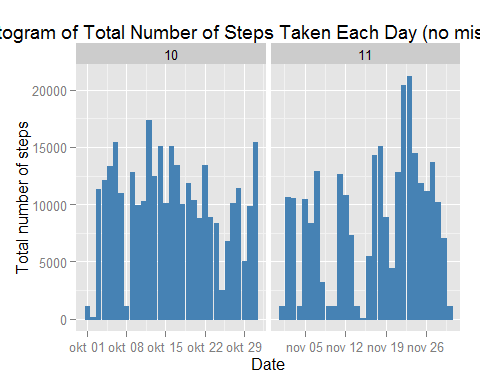
medianSteps <- aggregate(cleanData$steps, list(interval = as.numeric(as.character(cleanData$interval))), FUN = "median")  
names(medianSteps)[2] <- "medianOfSteps"

##### 3. Create a new dataset that is equal to the original dataset but with the missing data filled in.

fakeData <- data   
for (i in 1:nrow(fakeData)) {  
 if (is.na(fakeData$steps[i])) {  
 fakeData$steps[i] <- medianSteps[which(fakeData$interval[i] == medianSteps$interval), ]$medianOfSteps  
 }  
}

##### 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.

ggplot(fakeData, aes(date, steps)) +   
 geom\_bar(stat = "identity", colour = "steelblue", fill = "steelblue", width = 0.8) +   
 facet\_grid(. ~ month, scales = "free") +   
 labs(title = "Histogram of Total Number of Steps Taken Each Day (no missing data)", x = "Date", y = "Total number of steps")

  
 \* Do these values differ from the estimates from the first part of the assignment?

The Mean and median total number of steps taken per day are :

fakeTotalSteps <- aggregate(fakeData$steps, list(Date = fakeData$date), FUN = "sum")$x  
fakeMean <- mean(fakeTotalSteps)  
fakeMedian <- median(fakeTotalSteps)  
realMean <- mean(dailytotalSteps)   
realMedian <- median(dailytotalSteps)

Compare them with the two before imputing missing data:

# I started this code block with {r, echo=TRUE, results='show', warning=FALSE, message=TRUE}  
# I would put echo = FALSE if it weren't for the assignment ... but hey it's an exercise!   
message("Substracting the real mean (", round(realMean), ") from the fake mean (", round(fakeMean), ") gives : ", round(fakeMean - realMean) , " !)")

## Substracting the real mean (10766) from the fake mean (9504) gives : -1262 !)

message("Substracting the real median (", realMedian, ") from the fake median (", fakeMedian, ") gives : ", fakeMedian - realMedian , " !)")

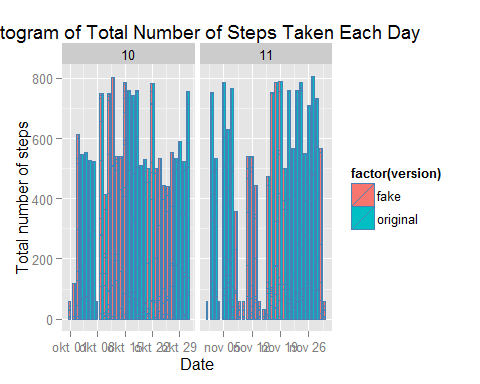
## Substracting the real median (10765) from the fake median (10395) gives : -370 !)

* What is the impact of imputing missing data on the estimates of the total daily number of steps?

After replacing the missing data by the median 5-min interval value, the fake mean of total steps taken per day is 1262 steps less than that of the mean of the recorded data; the new (fake) median of total steps taken per day is 370 steps less than that of the real (recorded) median.

We can also look at the difference in a graph :

#I first add a label (factor) to each dataset  
data$version <- 'original'  
fakeData$version <- 'fake'  
  
#and bind the datasets into a new data frame allData  
allData <- rbind(data, fakeData)  
  
ggplot(allData, aes(date, steps, fill = factor(version))) +   
 geom\_bar(stat = "identity", colour = "steelblue", width = 0.8, position="dodge") +   
 facet\_grid(. ~ month, scales = "free") +   
 labs(title = "Histogram of Total Number of Steps Taken Each Day", x = "Date", y = "Total number of steps")

  
 As such the differences do not look as big. Which goes to show that graphs do not always reveal all (or that I should have chosen another graph for this illustration).

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

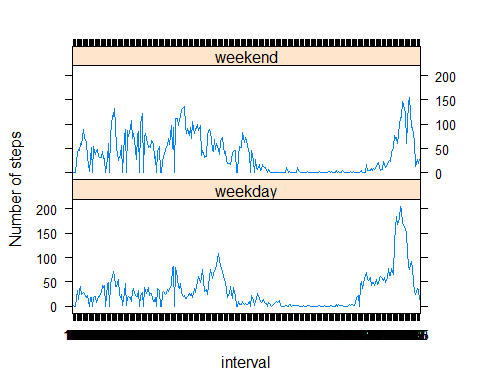
For this part the dataset with the filled-in missing values was used as instructed.

##### 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.

fakeData$dagvdweek <- weekdays(fakeData$date)  
fakeData$dagvdweek[fakeData$dagvdweek =='zaterdag'| fakeData$dagvdweek =='zondag'] <- 'weekend'  
fakeData$dagvdweek[fakeData$dagvdweek !='weekend'] <- 'weekday'  
fakeData$dagvdweek<- as.factor(fakeData$dagvdweek)

##### 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).

ppData <- ddply(fakeData, .(interval,dagvdweek), summarise, avesteps = mean(steps))  
xyplot(avesteps ~ interval | dagvdweek, data=ppData,  
 type='l',  
 lwd=1,  
 layout=c(1,2),  
 ylab = 'Number of steps')

  
 There seems to be a slight difference between weekdays and weekends. Weekdays are characterised by a longer period of low activity probably corresponding to time spent at work (desk). Also sleep seems to be longer during the weekend (the really flat part of the graph).