Assignment 1 RepResearch

Filip NovotnÄ‚Ëť

2017-10-27

## Loading and Preprocessing the Data

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.

The data is a comma delimited file, it includes 17,568 observations of 3 variables:

steps: Number of steps taken in a 5 min interval date: The date when the measurement was taken in YYY-MM-DD format interval: Identifier for the 5-min interval in which the measurement was taken

Firstly, you should set the right working directory: setwd() function

1. Load the data (the date column to character)

# Clear the workspace  
rm(list=ls())  
  
# Load the data  
activityData <- read.csv ("activity.csv", header = T, sep = ",", stringsAsFactors = F)

1. Convert the date column to the appropriate format

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

## 'data.frame': 17568 obs. of 3 variables:  
## $ steps : int NA NA NA NA NA NA NA NA NA NA ...  
## $ date : Date, format: "2012-10-01" "2012-10-01" ...  
## $ interval: int 0 5 10 15 20 25 30 35 40 45 ...

# Check the dimensions and a few rows of the newly created data frame  
dim(activityData)

## [1] 17568 3

head(activityData)

## 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

During the first day of data collection we have several intervals with missing values that we will need to deal later with.

## The mean of the total number of steps taken per day

The following lines calculate the total number of steps per day and the mean number of daily steps.

library (dplyr)

##   
## Attaching package: 'dplyr'

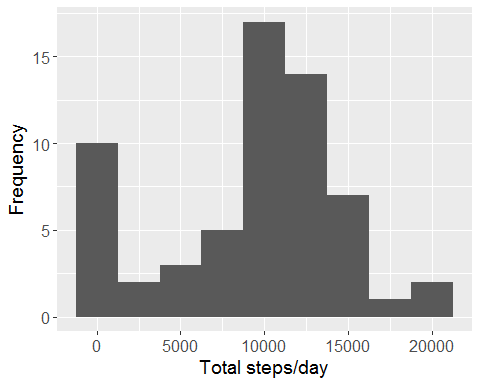
## The following objects are masked from 'package:stats':  
##   
## filter, lag

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

AvgDay <- activityData %>% group\_by(date) %>%  
 summarize(total.steps = sum(steps, na.rm = T),   
 mean.steps = mean(steps, na.rm = T))

The histogram of the total steps

library(ggplot2)  
g <- ggplot(AvgDay, aes(x=total.steps))  
g + geom\_histogram(binwidth = 2500) + theme(axis.text = element\_text(size = 12),   
 axis.title = element\_text(size = 14)) + labs(y = "Frequency") + labs(x = "Total steps/day")



dev.copy(png,'totalstepsday.png')

## png   
## 3

dev.off()

## png   
## 2

The data is symmetrically distributed around the center of the distribution, except for one class at the extreme left.

Calculate the mean and median of the total number of steps taken per day

summary(AvgDay$total.steps)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0 6778 10395 9354 12811 21194

summary(AvgDay$mean.steps)

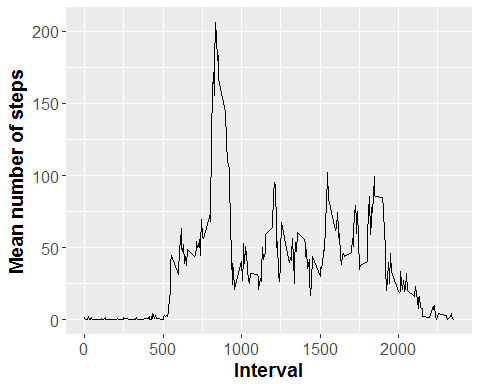
## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 0.1424 30.6979 37.3785 37.3826 46.1597 73.5903 8

The mean and the median of the total steps are close in value. There are also 8 missing values.

## The Average Daily Activity Pattern

Make a time series plot of the 5-minute interval (x-axis) and the average number of steps taken, average across all days (y-axis)

AvgInterval <- activityData %>% group\_by(interval) %>%  
 summarize(mean.steps = mean(steps, na.rm = T))  
g <- ggplot(AvgInterval, aes(x = interval, y = mean.steps))  
g + geom\_line() + theme(axis.text = element\_text(size = 12),   
 axis.title = element\_text(size = 14, face = "bold")) +   
 labs(y = "Mean number of steps") + labs(x = "Interval")



dev.copy(png,'mean.png')

## png   
## 3

dev.off()

## png   
## 2

The largest amount of steps occurs between time intervals 500 and 1000. The maximum average number of steps is 206 and occurs in time interval #835.

## Inputing the missing values

Calculate and report the total number of missing values in the dataset

mean(is.na(activityData$steps))

## [1] 0.1311475

sum(is.na(activityData$steps))

## [1] 2304

The number of NA’s is 2304 (13%)

Check for missing values in the interval column.

sum(is.na(AvgInterval$mean.steps))

## [1] 0

Since there are no missing values in this variable we will use it to fill in for NAs. Next we create a duplicate of the original data named newData and we will draw the appropriate values AvgInterval.

newData <- activityData

for (i in 1:nrow(newData)) {  
 if (is.na(newData$steps[i])) {  
 index <- newData$interval[i]  
 value <- subset(AvgInterval, interval==index)  
 newData$steps[i] <- value$mean.steps  
 }  
}  
head(newData)

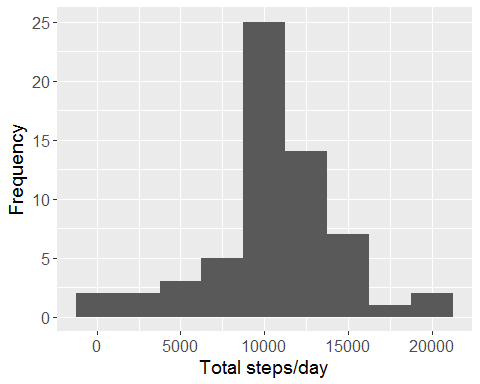
## steps date interval  
## 1 1.7169811 2012-10-01 0  
## 2 0.3396226 2012-10-01 5  
## 3 0.1320755 2012-10-01 10  
## 4 0.1509434 2012-10-01 15  
## 5 0.0754717 2012-10-01 20  
## 6 2.0943396 2012-10-01 25

There are numeric values in the first rows of the dataset now. Group the data by date and calculate daily totals.

newAvg <- newData %>% group\_by(date) %>%  
 summarize(total.steps = sum(steps, na.rm = T))

Construct the histogram.

g <- ggplot(newAvg, aes(x=total.steps))  
g + geom\_histogram(binwidth = 2500) + theme(axis.text = element\_text(size = 12),  
 axis.title = element\_text(size = 14)) + labs(y = "Frequency") + labs(x = "Total steps/day")



dev.copy(png,'totalstepsdaynew.png')

## png   
## 3

dev.off()

## png   
## 2

Similarly to the first histogram, symmetrically distributed data around the maximum without the column in the extreme left (which contained the days with missing data).

summary (AvgDay$total.steps)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0 6778 10395 9354 12811 21194

sd(AvgDay$total.steps, na.rm=T)

## [1] 5405.895

summary (newAvg$total.steps)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 41 9819 10766 10766 12811 21194

sd(newAvg$total.steps, na.rm=T)

## [1] 3974.391

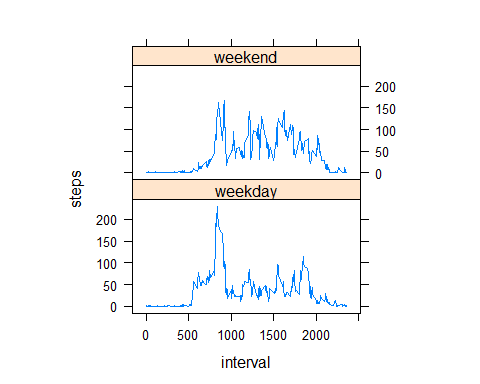
The new distribution is more concentrated around the center of gravity.

## Differences in activity patterns between weekdays and weekends

newData$day <- ifelse(as.POSIXlt(as.Date(newData$date))$wday%%6 ==   
 0, "weekend", "weekday")  
newData$day <- factor(newData$day, levels = c("weekday", "weekend"))

Create the plots

steps.interval= aggregate(steps ~ interval + day, newData, mean)  
library(lattice)  
xyplot(steps ~ interval | factor(day), data = steps.interval, aspect = 1/2,   
 type = "l")



dev.copy(png,'comparison.png')

## png   
## 3

dev.off()

## png   
## 2

The activity profiles between weekdays and weekends greatly differ. Weekend data is overall more evenly distributed throughout the day.