Reproducible Research: Peer Assessment 1

## Introduction

It is now possible to collect a large amount of data about personal movement using activity monitoring devices such as a Fitbit, Nike Fuelband, or Jawbone Up. These type of devices are part of the "quantified self" movement - a group of enthusiasts who take measurements about themselves regularly to improve their health, to find patterns in their behavior, or because they are tech geeks. But these data remain under-utilized both because the raw data are hard to obtain and there is a lack of statistical methods and software for processing and interpreting 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 dataset was downloaded in August 23rd, 2017 from [the course web site](https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip).

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 for the 5-minute interval in which measurement was taken

The dataset is stored in a comma-separated-value (CSV) file and there are a total of 17,568 observations in this dataset.

## Loading Necessary packages

This section lists (and loads) all the packages used during the analysis.

PS.: In order for chron to work properly, the timezone was changed to UTC, it will be returned to the current at the end of the analysis.

## Changin system timezone  
curTZ <- Sys.getenv("TZ")  
Sys.setenv(TZ = "UTC")  
  
## Loading packages  
library(chron)  
library(dplyr)  
library(ggplot2)

## Loading and preprocessing the data

In order to read the data without extracting the content of the .zip file the unz() function was used to create open and close a connection with the .zip file allowing the read.csv function to read the dataset directly from the zipped directory.

activity <- read.csv(unz("activity.zip", "activity.csv"))  
str(activity)

## '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 ...

By the output of the str() command above, it is possible to verify 2 possible problems for our analysis in the dataset, the format of the variables date and interval, in order to simplify the further steps of the analysis the code bellow was used to convert the format of the variable date to Date and the variable interval to time.

activity$date <- as.POSIXct(activity$date, format = "%Y-%m-%d")  
activity$interval <- strptime(formatC(activity$interval, width = 4, flag = "0"),  
 format = "%H%M", tz = "UTC")  
activity$interval <- times(format(activity$interval, "%H:%M:%S"))  
str(activity)

## 'data.frame': 17568 obs. of 3 variables:  
## $ steps : int NA NA NA NA NA NA NA NA NA NA ...  
## $ date : POSIXct, format: "2012-10-01" "2012-10-01" ...  
## $ interval:Class 'times' atomic [1:17568] 0 0.00347 0.00694 0.01042 0.01389 ...  
## .. ..- attr(\*, "format")= chr "h:m:s"

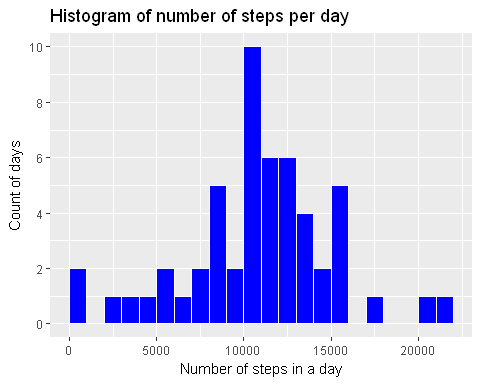
It would be possible to work with the variable interval as intergers but when plotting it in a time series the x axis would be out of scale making the visualization not as clear as it could be.So, the package chron was used in order to extrac the time at which each interval was acquire.

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

To answer this question the dplyr package was used to summarize the number of steps taken each day. A histogram was made using the ggplot2 system to show the results.

It was decided that during the summarization the NAs would not be removed as it would cause the day with no available data acquired to be considered as zero which is misleading.

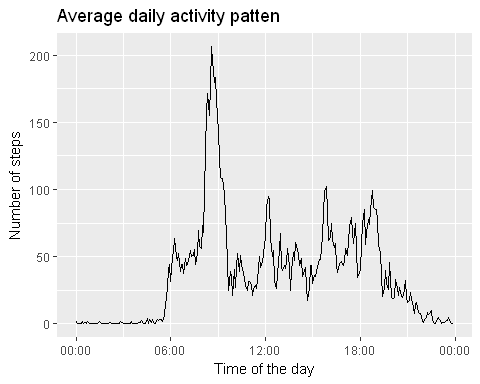
totalByDay <- activity %>%  
 group\_by(date) %>%  
 summarize(steps = sum(steps))  
  
ggplot(totalByDay, aes(x = steps)) +   
 geom\_histogram(col = "white", fill = "blue", na.rm = TRUE,   
 binwidth = 1000, center = 500) +   
 scale\_x\_continuous(name = "Number of steps in a day") +  
 scale\_y\_continuous(name = "Count of days", breaks = seq(0, 10, by = 2)) +  
 ggtitle("Histogram of number of steps per day")



## What is the average daily activity pattern?

In this section, a summarization was made to extract the average number of steps in each interval. The results are shown in the time series plot below.

avgByInt <- activity %>%  
 group\_by(interval) %>%  
 summarize(steps = mean(steps, na.rm = TRUE))  
  
ggplot(avgByInt, aes(x = interval, y = steps)) +   
 geom\_line() +  
 scale\_x\_chron(name = "Time of the day", format = "%H:%M") +  
 ylab("Number of steps") +  
 ggtitle("Average daily activity patten")



## Imputing missing values

There are a number of days/intervals where there are missing values (coded as NA). The presence of missing days may introduce bias into some calculations or summaries of the data.

sum(is.na(activity$steps))

## [1] 2304

To define a good strategy for filling missing values, find out where the missing values occour and try to find some pattern.

table(activity$date[is.na(activity$steps)])

##   
## 2012-10-01 2012-10-08 2012-11-01 2012-11-04 2012-11-09 2012-11-10   
## 288 288 288 288 288 288   
## 2012-11-14 2012-11-30   
## 288 288

The command above shows that every time a missing data appear it means that that whole day is missing. In order to fill this values, a good strategy would be to use the average pattern calculated in the previous step for the missing days.

fillActivity <- activity  
fillActivity $steps[is.na(activity$steps)] <- avgByInt$steps

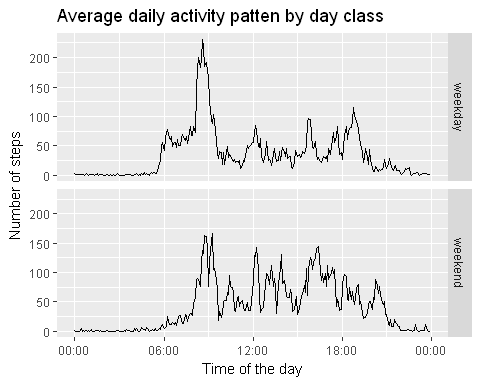
## Are there differences in activity patterns between weekdays and weekends?

To answer this question a new factor variable was created in the dataset with two levels - "weekday" and "weekend" indicating whether a given date is a weekday or weekend day.

fillActivity$dayClass <- is.weekend(fillActivity$date)  
fillActivity$dayClass <- factor(fillActivity$dayClass,   
 labels = c("weekday", "weekend"))

To extract the patterns of the weekdays and weekends, the data was grouped by intervals and day class and then summarized. The results are shown in the plot below.

avgByIntDayClass <- fillActivity %>%  
 group\_by(interval, dayClass) %>%  
 summarize(steps = mean(steps)) %>%  
 arrange(dayClass, interval)  
  
ggplot(avgByIntDayClass, aes(x = interval, y = steps)) +  
 geom\_line() +  
 facet\_grid(dayClass ~ .) +  
 scale\_x\_chron(name = "Time of the day", format = "%H:%M") +  
 ylab("Number of steps") +  
 ggtitle("Average daily activity patten by day class")



## Returning to the previous timezone

Sys.setenv(TZ = curTZ)