Analyzing Activity Monitoring Device Data

Matteo Gambera

11 marzo 2020

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 underutilized 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.

```r
library(ggplot2)
setwd("/home/matteo/Scrivania/datasciencecoursera")
url <- "https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip"
path <- paste0(getwd() , "/Reproducible_Research")
download.file(url, file.path(path, "dataFiles.zip"))
unzip(file.path(path, zipfile = "dataFiles.zip"))
```

Data

The data for this assignment can be downloaded from the course web site: Dataset: Activity monitoring data [52K] 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.

```r
activity <- read.csv("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 ...
```

```r
head(activity)
```

```
##   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
```

Clean Data

```
# problem!!! the problem ask before to do this... bha seems stupid
activity_na <- activity # build a new dataframe for this problem

n_na_step <- sum(is.na(activity$steps))
n_na_date <- sum(is.na(as.character(activity$date)))

n_na_step
```

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

```
n_na_date
```

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

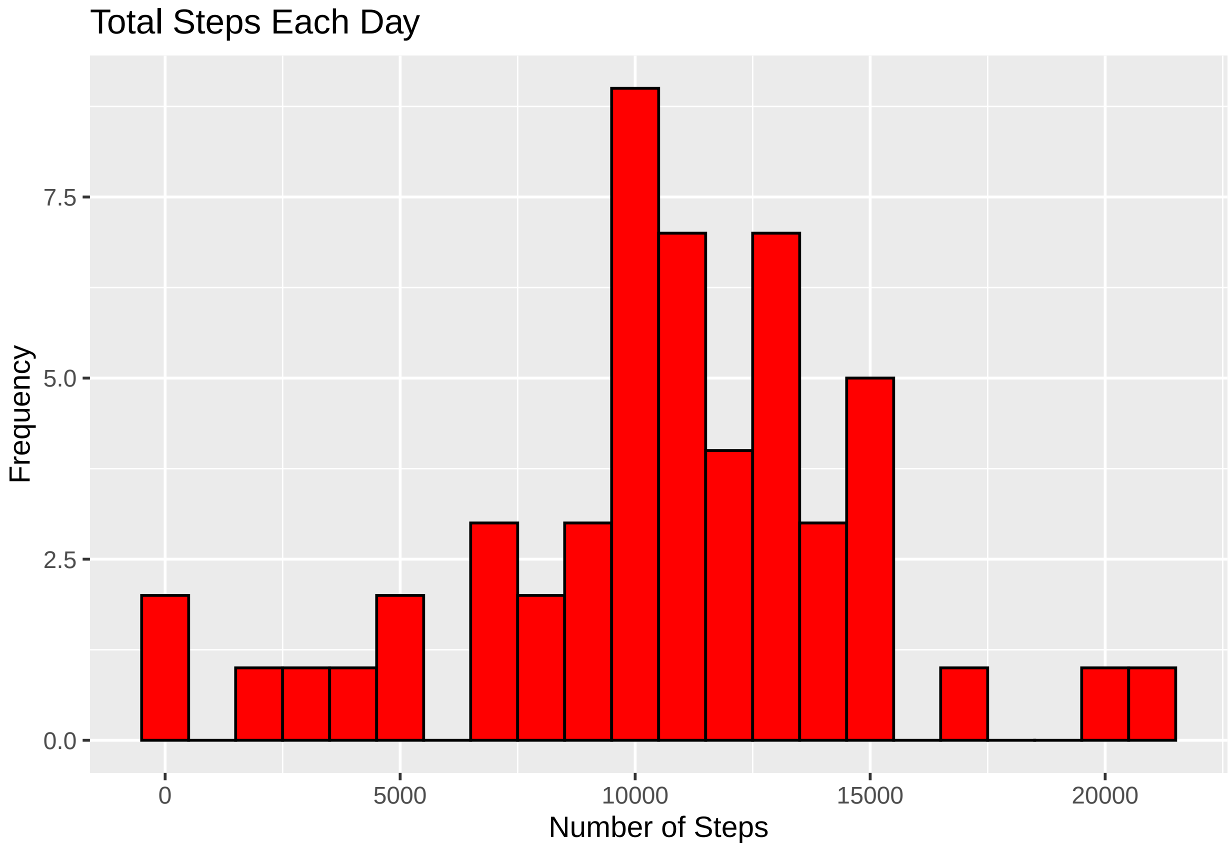
```
na <- (is.na(activity$steps))
activity <- activity[!na,]
str(activity)
```

```
## 'data.frame':    15264 obs. of  3 variables:
##  $ steps   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ date    : Factor w/ 61 levels "2012-10-01","2012-10-02",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ interval: int  0 5 10 15 20 25 30 35 40 45 ...
```

Histogram of the total number of steps taken each day

build new dataframe to work with step aggragated

```
steps_by_day <- aggregate(steps ~ date, data = activity , sum)
#Adding column names to the created data frame
colnames(steps_by_day) <- c("date", "steps")
# date is a factor, so i sum every step by days
ggplot(steps_by_day, aes(x = steps)) +
    geom_histogram(fill = "red", binwidth = 1000, color="black") +
    labs(title = "Total Steps Each Day", x = "Number of Steps", y = "Frequency")
```



# Mean and median number of steps taken each day

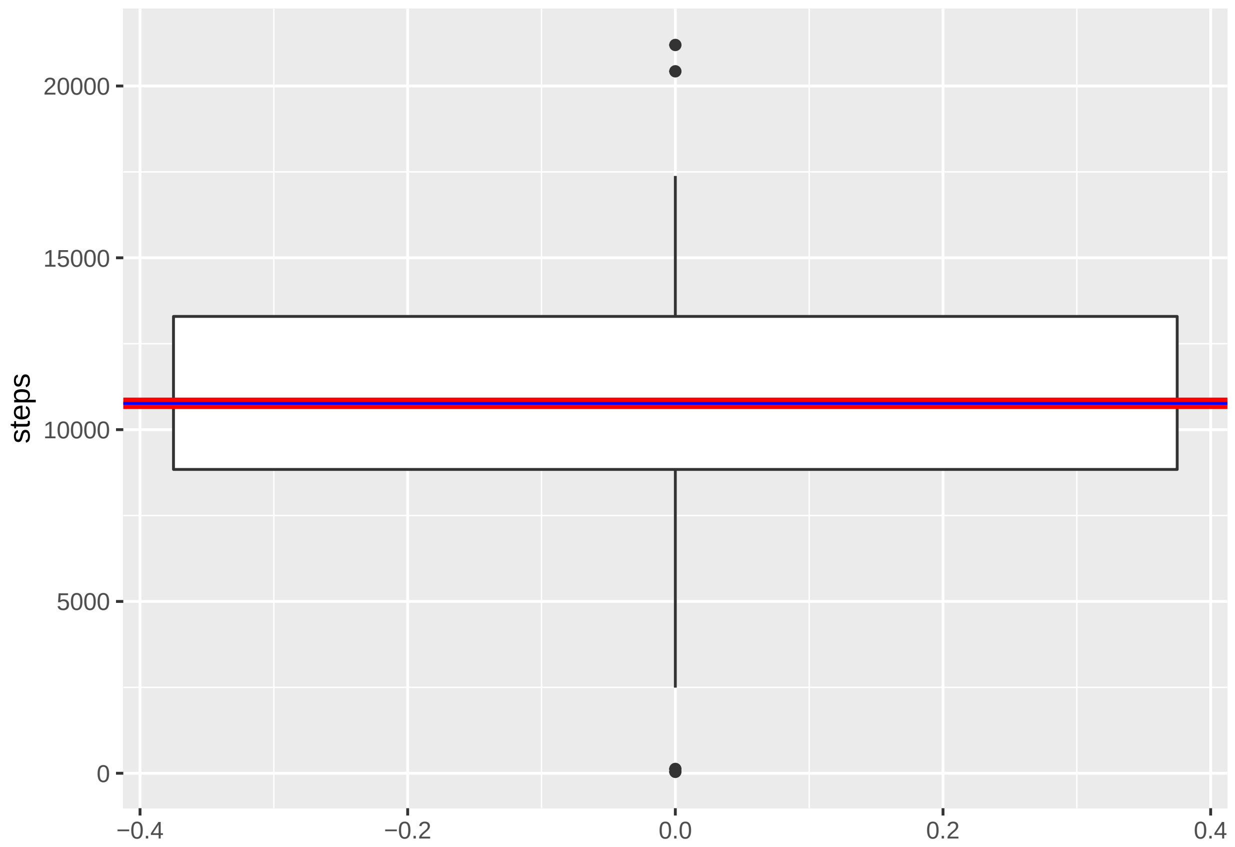
```
mean <- mean(steps_by_day$steps)
median <- median(steps_by_day$steps)
mean
```

```
## [1] 10766.19
```

```
median
```

```
## [1] 10765
```

```
ggplot(steps_by_day, aes(y = steps)) +
    geom_boxplot() +
    geom_hline(yintercept = mean, color="red", size=2) +
    geom_hline(yintercept = median, color="blue")
```



# Time series plot of the average number of steps taken

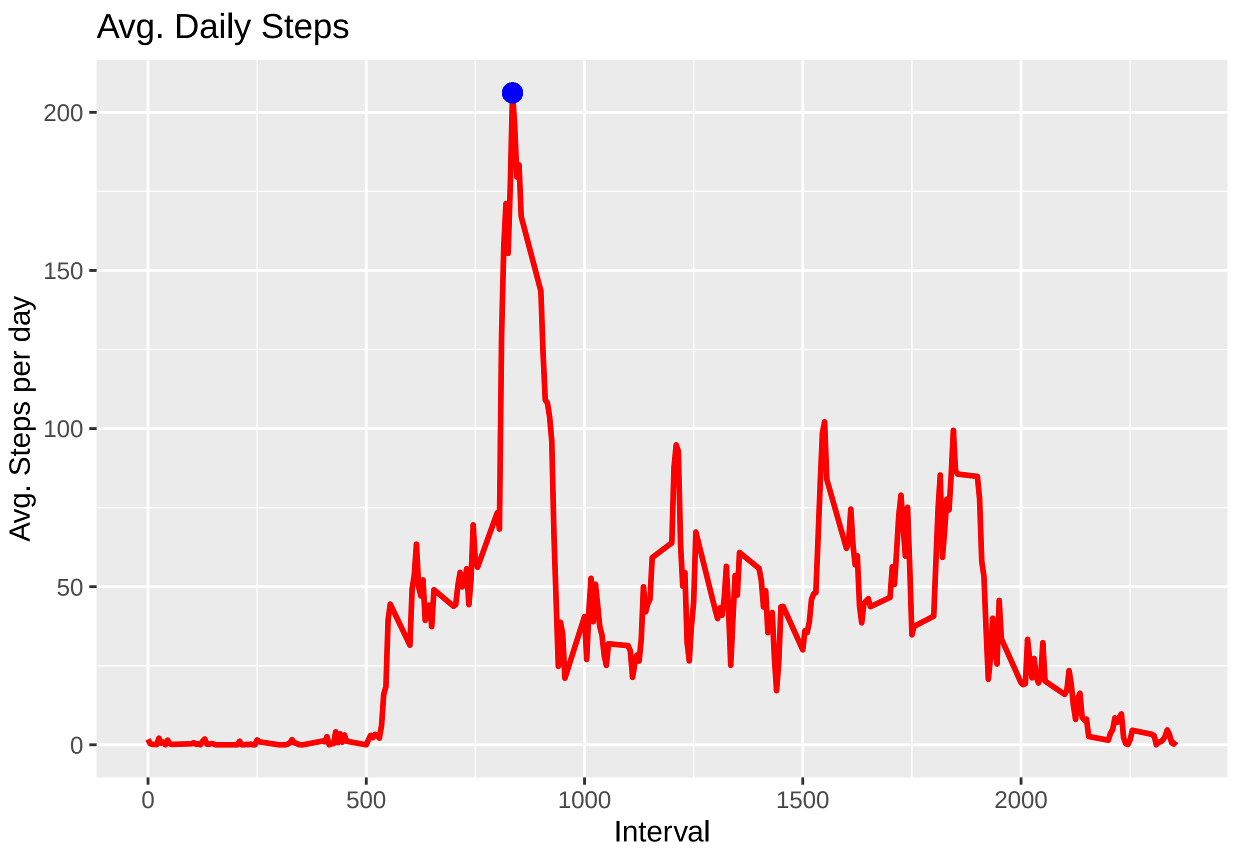
```
steps <- aggregate(activity$steps, by=list(interval=activity$interval), FUN=mean)
colnames(steps) <- c("interval", "average_steps")
max_steps <- max(steps$average_steps)
max_steps
```

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

```
intervale_max_steps <- steps[which.max(steps$average_steps),]$interval
intervale_max_steps
```

```
## [1] 835
```

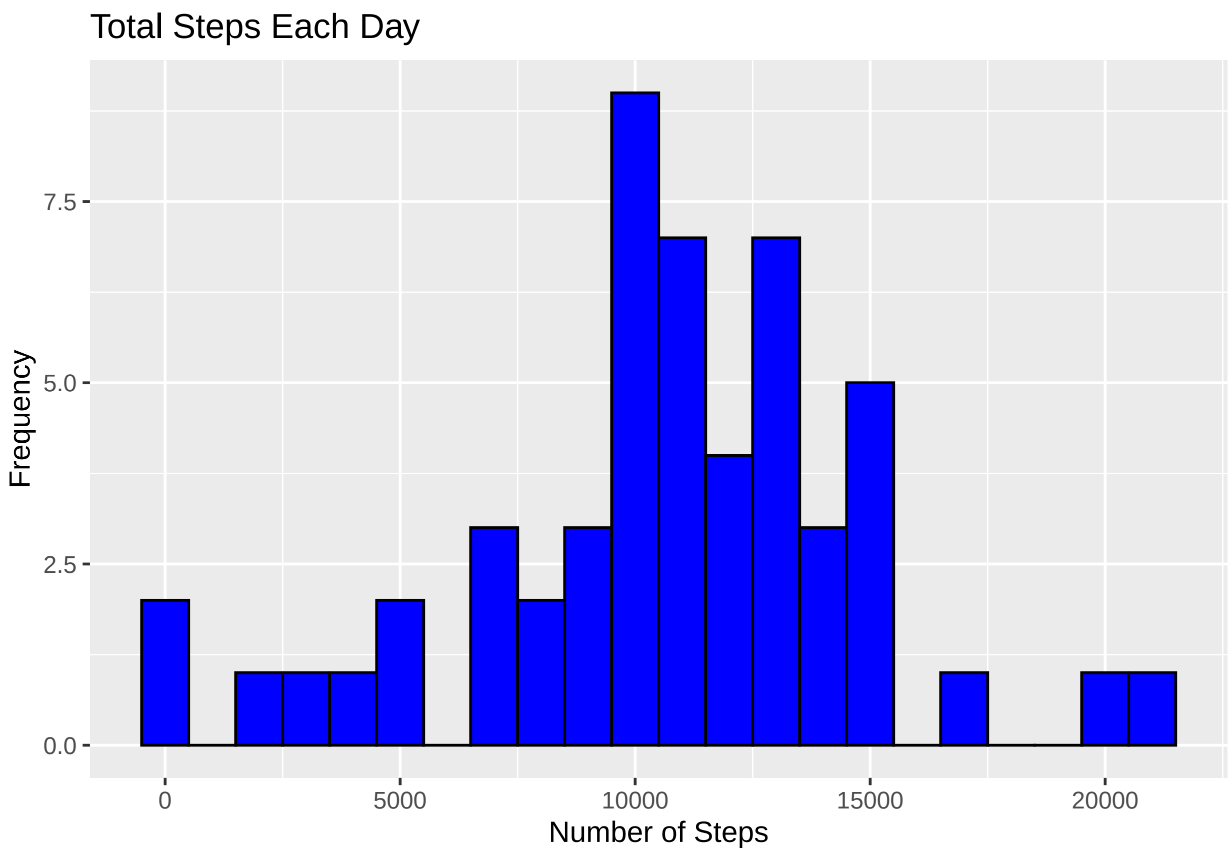
```
ggplot(steps, aes(x = interval , y = average_steps)) +
        geom_line(color="red", size=1) +
        labs(title = "Avg. Daily Steps", x = "Interval", y = "Avg. Steps per day") +
        geom_point(x = intervale_max_steps , y = max_steps , colour="blue", size = 3)
```



The strategy for Missing values

```
complete_data <- activity_na
complete_data <- aggregate(steps ~ date, data = activity , sum)
NA_index <- which(is.na(as.character(complete_data$steps)))
#Imputing missing values using the mean for that 5-minute interval
complete_data[NA_index, ]$steps<-unlist(lapply(NA_index, FUN=function(NA_index){
                steps_per_interval[data[NA_index,]$interval==steps_per_interval$interval,]$average_steps
                }))
steps_each_day_complete <- aggregate(steps ~ date, data = complete_data, sum)
#Adding column names to the created data frame
colnames(steps_each_day_complete) <- c("date", "steps")

ggplot(steps_each_day_complete, aes(x = steps)) +
    geom_histogram(fill = "blue", binwidth = 1000, color="black") +
    labs(title = "Total Steps Each Day", x = "Number of Steps", y = "Frequency")
```



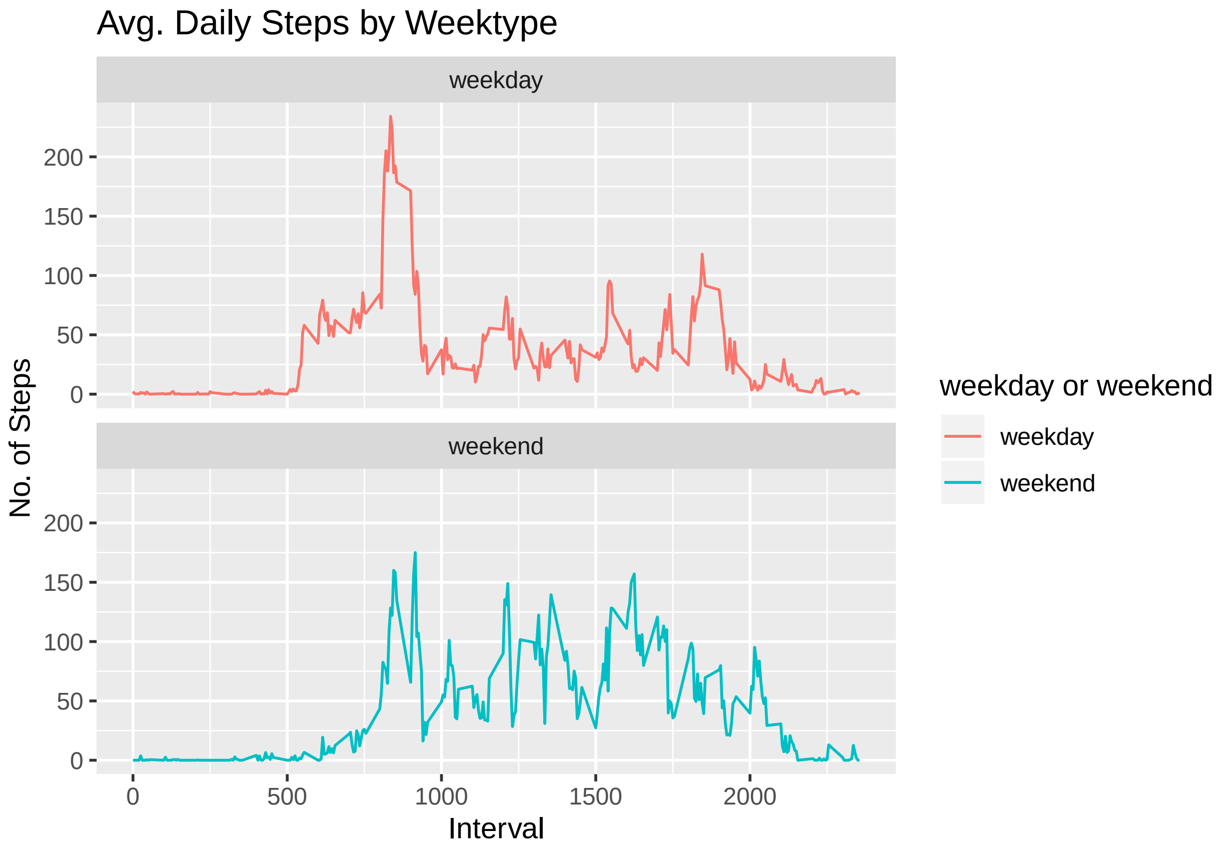
```
activity2 <- data.table::fread(input = "activity.csv")
activity2[, date := as.POSIXct(date, format = "%Y-%m-%d")]
na <- (is.na(activity2$steps))
activity2 <- activity2[!na,]
activity2[, `Day of Week`:= weekdays(x = date)]
activity2[grepl(pattern = "lunedì|martedì|mercoledì|giovedì|venerdì", x = `Day of Week`), "weekday or we
activity2[grepl(pattern = "sabato|domenica", x = `Day of Week`), "weekday or weekend"] <- "weekend"
activity2[, `weekday or weekend` := as.factor(`weekday or weekend`)]
head(activity2, 10)
```

```
##     steps       date interval Day of Week weekday or weekend
##  1:     0 2012-10-02        0     martedì            weekday
##  2:     0 2012-10-02        5     martedì            weekday
##  3:     0 2012-10-02       10     martedì            weekday
##  4:     0 2012-10-02       15     martedì            weekday
##  5:     0 2012-10-02       20     martedì            weekday
##  6:     0 2012-10-02       25     martedì            weekday
##  7:     0 2012-10-02       30     martedì            weekday
##  8:     0 2012-10-02       35     martedì            weekday
##  9:     0 2012-10-02       40     martedì            weekday
## 10:     0 2012-10-02       45     martedì            weekday
```

```
library(ggplot2)
```

```
activity2[is.na(steps), "steps"] <- activity2[, c(lapply(.SD, median, na.rm = TRUE)), .SDcols = c("steps
Interval <- activity2[, c(lapply(.SD, mean, na.rm = TRUE)), .SDcols = c("steps"), by = .(interval, `weel
```

```
ggplot(Interval , aes(x = interval , y = steps, color=`weekday or weekend`)) +
    geom_line() +
    labs(title = "Avg. Daily Steps by Weektype", x = "Interval", y = "No. of Steps") +
    facet_wrap(~`weekday or weekend` , ncol = 1, nrow=2)
```


Avg. Daily Steps by Weektype
weekday
weekend
No. of Steps
Interval
weekday or weekend
weekday
weekend