Peer-graded Assignment: Course Project 1

Maria Lyasheva

11 February 2019

Background

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.

Information about the data for the assessment

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.

Loading and preprocessing the data

Loading required libraries

```
library(ggplot2) # this will be used for the graphs
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

library(chron) # this will be used for the function weekdays()
library(Hmisc) # this will be used for substituting NA values

## Loading required package: lattice

## Loading required package: survival

## Loading required package: Formula

##
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:dplyr':
##
##     src, summarize

## The following objects are masked from 'package:base':
##
##     format.pval, units
```

Loading and reading the dataset

```
activity <- read.csv("activity.csv", header = TRUE)
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

tail(activity)

##       steps       date interval
## 17563    NA 2012-11-30     2330
## 17564    NA 2012-11-30     2335
## 17565    NA 2012-11-30     2340
```

```
## 17566    NA 2012-11-30     2345
## 17567    NA 2012-11-30     2350
## 17568    NA 2012-11-30     2355
```

What is mean total number of steps taken per day?

Calculate the total number of steps taken per day

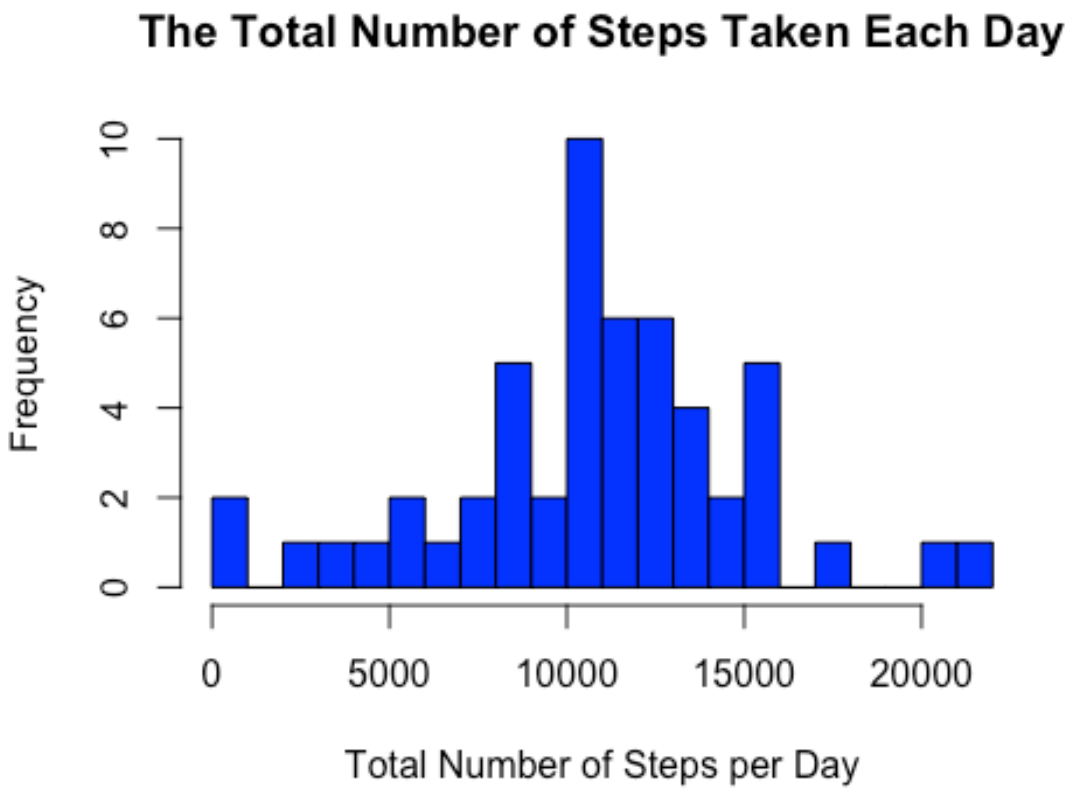
Summary of all steps that were taken daily.

```
steps_per_day <- aggregate(steps ~ date, activity, FUN = sum)
head(steps_per_day)

##         date steps
## 1 2012-10-02   126
## 2 2012-10-03 11352
## 3 2012-10-04 12116
## 4 2012-10-05 13294
## 5 2012-10-06 15420
## 6 2012-10-07 11015

tail(steps_per_day)

##          date steps
## 48 2012-11-24 14478
## 49 2012-11-25 11834
## 50 2012-11-26 11162
## 51 2012-11-27 13646
## 52 2012-11-28 10183
## 53 2012-11-29  7047
```

Make a histogram of the total number of steps taken each day

The histogram, representing the total number of steps taken daily.

```
hist (steps_per_day$steps,
      col = "blue",
      breaks = 20,
      xlab = "Total Number of Steps per Day",
      ylab = "Frequency",
      main = "The Total Number of Steps Taken Each Day")
```



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

```
mean_steps_per_day <- mean (steps_per_day$steps)
mean_steps_per_day

## [1] 10766.19

median_steps_per_day <- median (steps_per_day$steps)
median_steps_per_day

## [1] 10765
```

The mean total number of steps taken per day was 10766.19. The median total number of steps taken per day was 10765.

What is the average daily activity pattern?

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)

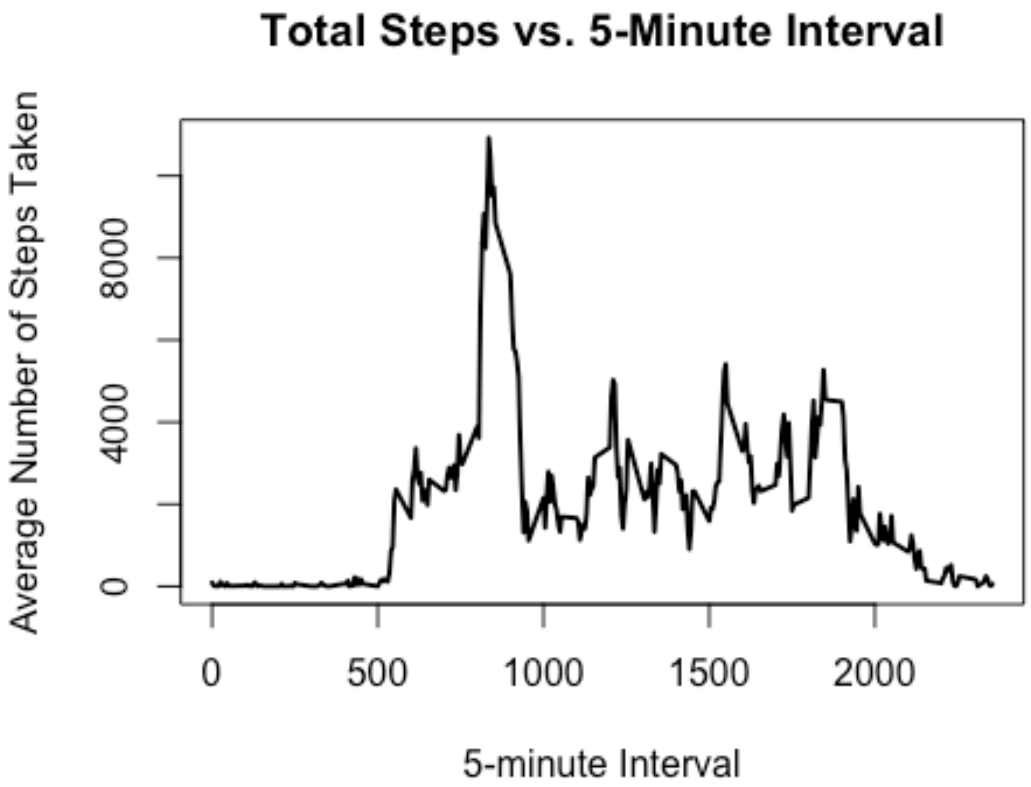
```
steps_per_day_interval <- aggregate(steps ~ interval, activity, FUN=sum)
head(steps_per_day_interval)
```

```
##   interval steps
## 1        0    91
## 2        5    18
## 3       10     7
## 4       15     8
## 5       20     4
## 6       25   111
```

```r
tail(steps_per_day_interval)
```

```
##     interval steps
## 283     2330   138
## 284     2335   249
## 285     2340   175
## 286     2345    34
## 287     2350    12
## 288     2355    57
```

```r
plot(steps_per_day_interval$interval, steps_per_day_interval$steps,
     type = "l", lwd = 2,
     xlab = "5-minute Interval",
     ylab = "Average Number of Steps Taken",
     main = "Total Steps vs. 5-Minute Interval")
```



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

```
maximum_number_of_steps <-
steps_per_day_interval[which.max(steps_per_day_interval$steps),1]
maximum_number_of_steps

## [1] 835
```

The maximum number of steps are contained in the 835th 5-min interval.

Imputing missing values

Calculate and report the total number of missing values in the dataset (i.e. the total number of rows with NAs)

```
NA_total_number <- table(is.na(activity))
NA_total_number

##
## FALSE  TRUE
## 50400  2304
```

Overall, there are 2304 missing values (NA) in the dataset.

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.

To substitute the missing values (NAs) I used the 'impute' function in 'Hmisc' package. This function replaced missing values with mean.

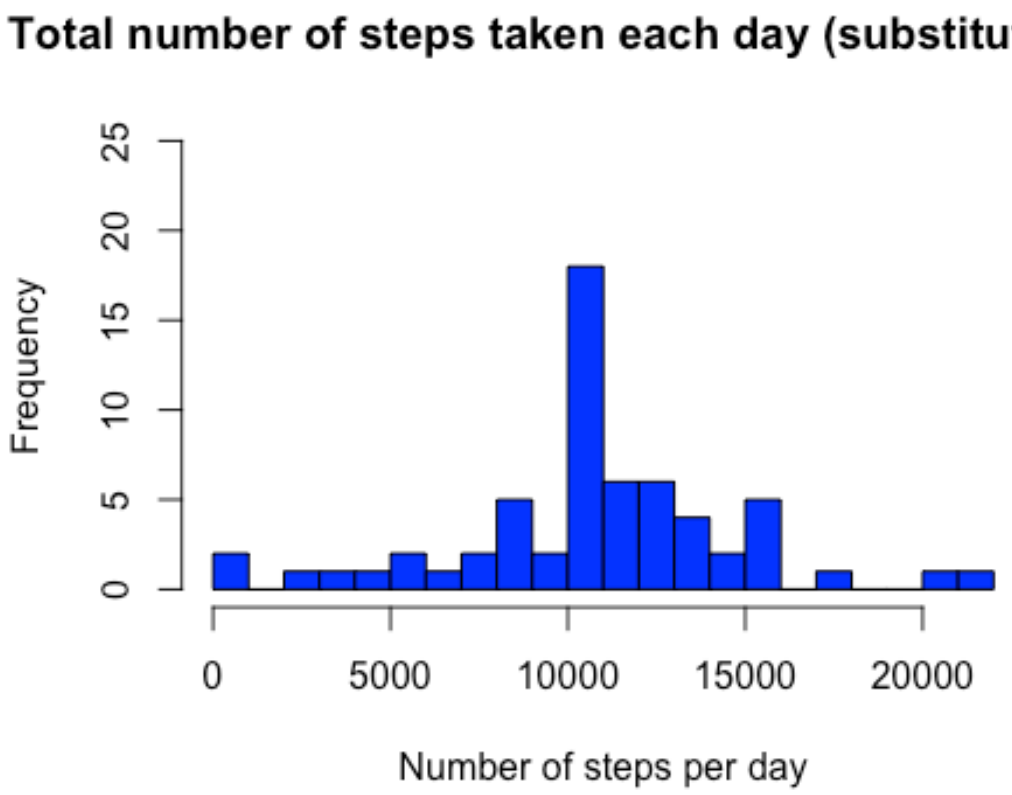
Create a new dataset that is equal to the original dataset but with the missing data filled in.

```
activity_substituted <- activity
activity_substituted$steps <- impute(activity$steps, fun=mean)
```

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. Do these values differ from the estimates from the first part of the assignment? What is the impact of imputing missing data on the estimates of the total daily number of steps?

```
steps_per_day_withoutNA <- aggregate(activity_substituted$steps,
                                     by = list(Steps.Date =
activity_substituted$date),
                                     FUN = "sum")
hist(steps_per_day_withoutNA$x, col = "blue",
     breaks = 20,
```

```
     main = "Total number of steps taken each day (substituted data)",
     xlab = "Number of steps per day",
     ylim = c(0,25))
```




```
mean_steps_per_day_withoutNA <- mean(steps_per_day_withoutNA[,2])
mean_steps_per_day_withoutNA

## [1] 10766.19
```

New mean total daily number of steps is 10766.19.

```
median_steps_per_day_withoutNA <- median(steps_per_day_withoutNA[,2])
median_steps_per_day_withoutNA

## [1] 10766.19
```

New median total daily number of steps is 10766.19.

The mean total number of steps taken per day from the original data set is 10766.19, which is the same with the mean total daily number of steps from the new dataset.

However, the median total number of steps taken per day from original dataset is 10765, whereas the median total number of steps taken per day from new dataset is slightly higher, 10766.19.

Thus, the mean are same but new median differs from original median by 1.19.

Are there differences in activity patterns between weekdays and weekends?

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.

is.weekend function divide the dataset into two groups: weekdays (Monday, Tuesday, Wednesday, Thursday or Friday) and weekends (Saturday and Sunday)

```
table(is.weekend(activity_substituted$date))

##
## FALSE  TRUE
## 12960  4608
```

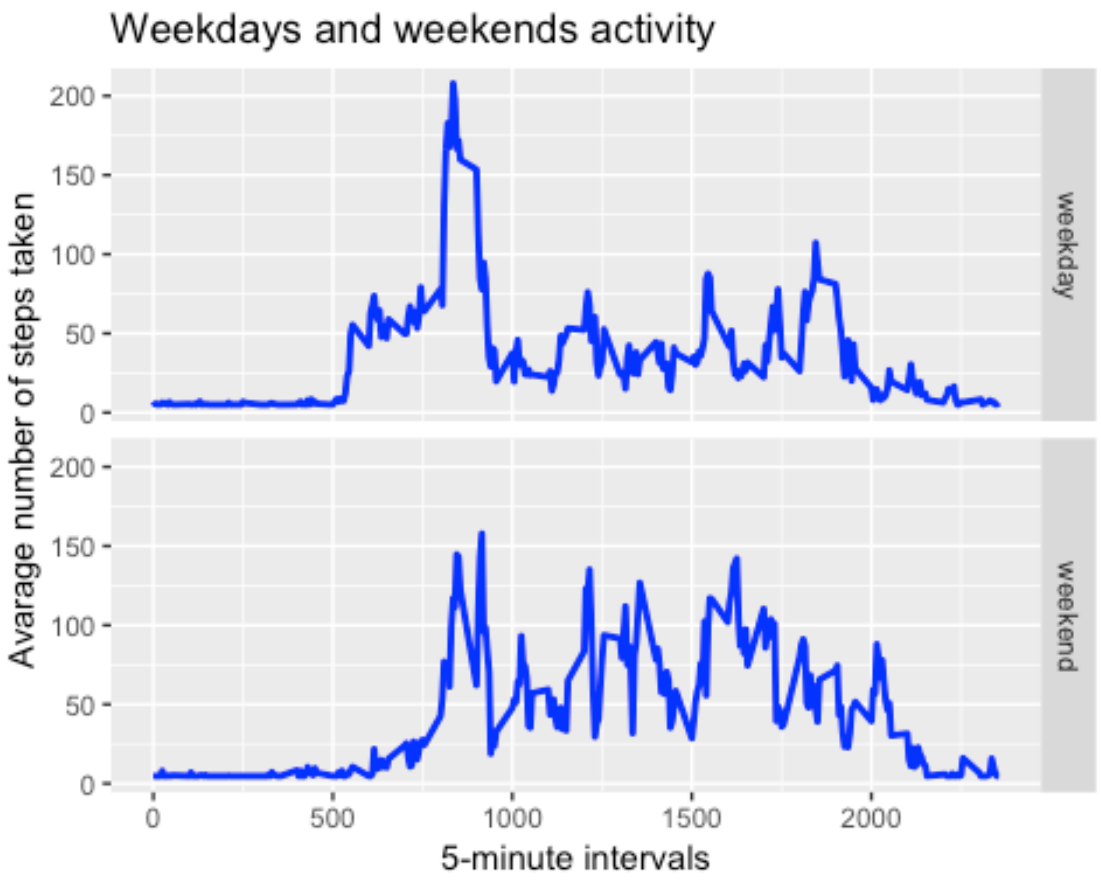
4608 - weekends (TRUE) 12960 - weekdays (FALSE)

To make it easier to work with the dataset, I will add new factor variable "dayofweek"

```
activity_substituted$dayofweek <-
ifelse(is.weekend(activity_substituted$date), "weekend", "weekday")
table(activity_substituted$dayofweek)

##
## weekday weekend
##   12960    4608
```

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).

```
activity_substituted_new<- aggregate(steps ~ interval + dayofweek,
activity_substituted, FUN=mean)

ggplot(activity_substituted_new, aes(x=interval, y=steps)) +
  geom_line(color="blue", size=1) +
  facet_wrap(~dayofweek, nrow=2) +
        facet_grid(dayofweek ~ .) +
  labs(x="5-minute intervals", y="Avarage number of steps taken") +
        ggtitle("Weekdays and weekends activity")
```



Conclusion: during the day the number of steps on weekends is generally higher than on weekdays, whereas the number of steps in the morning is higher on weekdays than on weekends.