Assignment1

Duc Nguyen

November 18, 2016

Table of Contents

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 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. Review criterialess

Repo

Valid GitHub URL At least one commit beyond the original fork Valid SHA-1 SHA-1 corresponds to a specific commit

Commit containing full submission

Code for reading in the dataset and/or processing the data Histogram of the total number of steps taken each day Mean and median number of steps taken each day Time series plot of the average number of steps taken The 5-minute interval that, on average, contains the maximum number of steps Code to describe and show a strategy for imputing missing data Histogram of the total number of steps taken each day after missing values are imputed Panel plot comparing the average number of steps taken per 5-minute interval across weekdays and weekends All of the R code needed to reproduce the results (numbers, plots, etc.) in the report

Assignment

This assignment will be described in multiple parts. You will need to write a report that answers the questions detailed below. Ultimately, you will need to complete the entire assignment in a single R markdown document that can be processed by knitr and be transformed into an HTML file.

Throughout your report make sure you always include the code that you used to generate the output you present. When writing code chunks in the R markdown document, always use echo = TRUE so that someone else will be able to read the code. This assignment will be evaluated via peer assessment so it is essential that your peer evaluators be able to review the code for your analysis.

For the plotting aspects of this assignment, feel free to use any plotting system in R (i.e., base, lattice, ggplot2)

Fork/clone the GitHub repository created for this assignment. You will submit this assignment by pushing your completed files into your forked repository on GitHub. The assignment submission will consist of the URL to your GitHub repository and the SHA-1 commit ID for your repository state.

NOTE: The GitHub repository also contains the dataset for the assignment so you do not have to download the data separately.

Solution and results

I. Loading and preprocessing the data

Show any code that is needed to Load the data (i.e. read.csv()). Process/transform the data (if necessary) into a format suitable for your analysis

```
# Loading and preprocessing the data
dt <- read.csv("activity.csv", header = T)
# Checking the data
dim(dt)

## [1] 17568     3

str(dt)

## 'data.frame':    17568 obs. of  3 variables:
##  $ steps   : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ date    : Factor w/ 61 levels "10/1/2012","10/10/2012",..: 1 1 1 1 1 1
1 1 1 1 ...
##  $ interval: int  0 5 10 15 20 25 30 35 40 45 ...

head(dt)

##   steps      date interval
## 1    NA 10/1/2012        0
## 2    NA 10/1/2012        5
## 3    NA 10/1/2012       10
## 4    NA 10/1/2012       15
## 5    NA 10/1/2012       20
## 6    NA 10/1/2012       25

tail(dt)

##       steps       date interval
## 17563    NA 11/30/2012     2330
## 17564    NA 11/30/2012     2335
## 17565    NA 11/30/2012     2340
## 17566    NA 11/30/2012     2345
## 17567    NA 11/30/2012     2350
## 17568    NA 11/30/2012     2355

# checking the missing values
missing_dt <- dt[is.na(dt$steps),]
dim(missing_dt)

## [1] 2304    3
```

II. What is mean total number of steps taken per day?

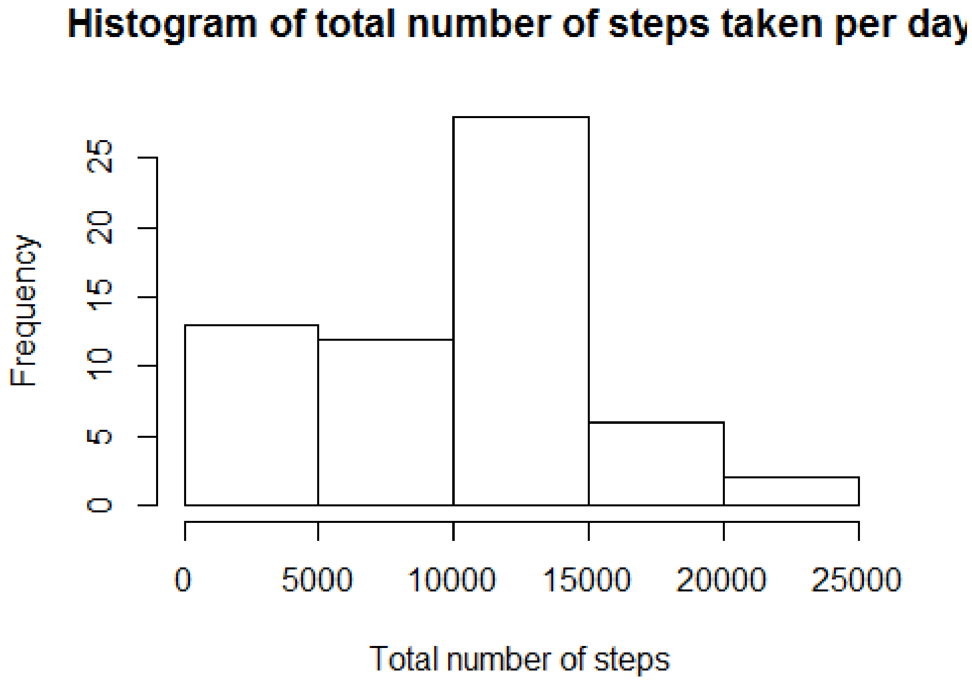
For this part of the assignment, you can ignore the missing values in the dataset.

1. Calculate the total number of steps taken per day

```
# The data without any missing values
dt1 <- dt[!is.na(dt$steps),]

# Calculate the total number of steps taken per day
total_number_steps <- with(dt, tapply(steps, as.factor(dt$date), sum, na.rm =
T))
```

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

```
hist(total_number_steps, main = "Histogram of total number of steps taken per
day", xlab = "Total number of steps")
```



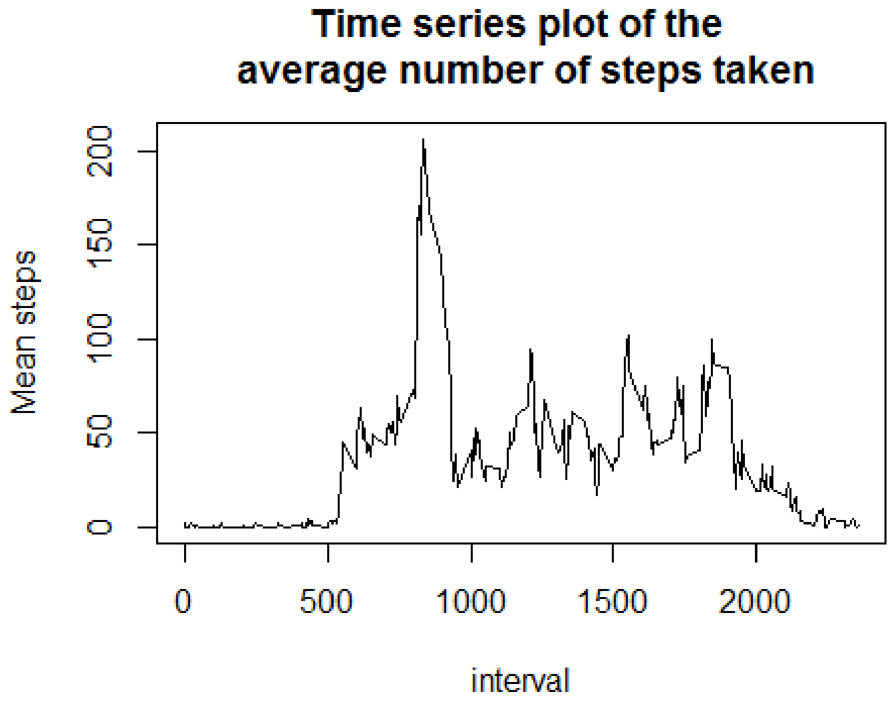
3. Calculate and report the mean and median of the total number of steps taken per day

```
summary(total_number_steps)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##       0    6778   10400    9354   12810   21190
```

III. 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)

```
mean_steps <- with(dt1, tapply(steps, dt1$interval, mean))
interval <- levels(as.factor(dt1$interval))
plot(interval, mean_steps, type = "l", main = "Time series plot of the \n
average number of steps taken", xlab = "interval", ylab = "Mean steps")
```



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

```
table <- data.frame(mean_steps, interval)
table[table$mean_steps==max(table$mean_steps),][2]

##     interval
## 835      835
```

IV. Imputing missing values

Note that 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.

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

```
# total number of missing values in the dataset
length(missing_dt$steps)

## [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.

```
# in this exercise I am going to replace the missing values by the the
average number of steps taken, averaged across all days.
```

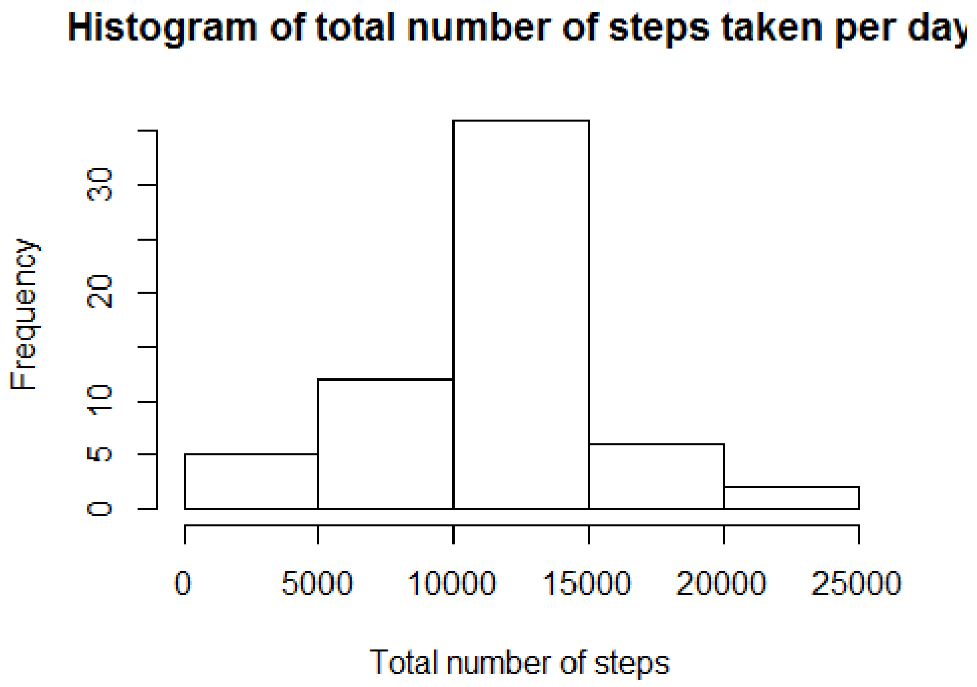
```
# Using this method we do not affect this data
mean_steps <- with(dt1, tapply(steps, dt1$interval, mean))
missing_dt$steps <- mean_steps
```

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

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

new_dt <- rbind(dt1, missing_dt)
new_dt <- new_dt[order(new_dt$date), ]
```

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. 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?

```
total_number_steps2 <- with(new_dt, tapply(steps, as.factor(new_dt$date),
sum))
#Make a histogram of the total number of steps taken each day
hist(total_number_steps2, main = "Histogram of total number of steps taken
per day", xlab = "Total number of steps")
```



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

Mean and median total number of steps taken per day WITHOUT filling in the missing values

```
summary(total_number_steps)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##       0    6778   10400    9354   12810   21190
```

Mean and median total number of steps taken per day WITH filling in the missing values

```
summary(total_number_steps2)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##      41    9819   10770   10770   12810   21190
```

Yes, mean and median total number of steps taken per day for the filled in missing values differ from these of the origional dataset.

V. Are there differences in activity patterns between weekdays and weekends?

For this part the weekdays() function may be of some help here. Use the dataset with the filled-in missing values for this part.

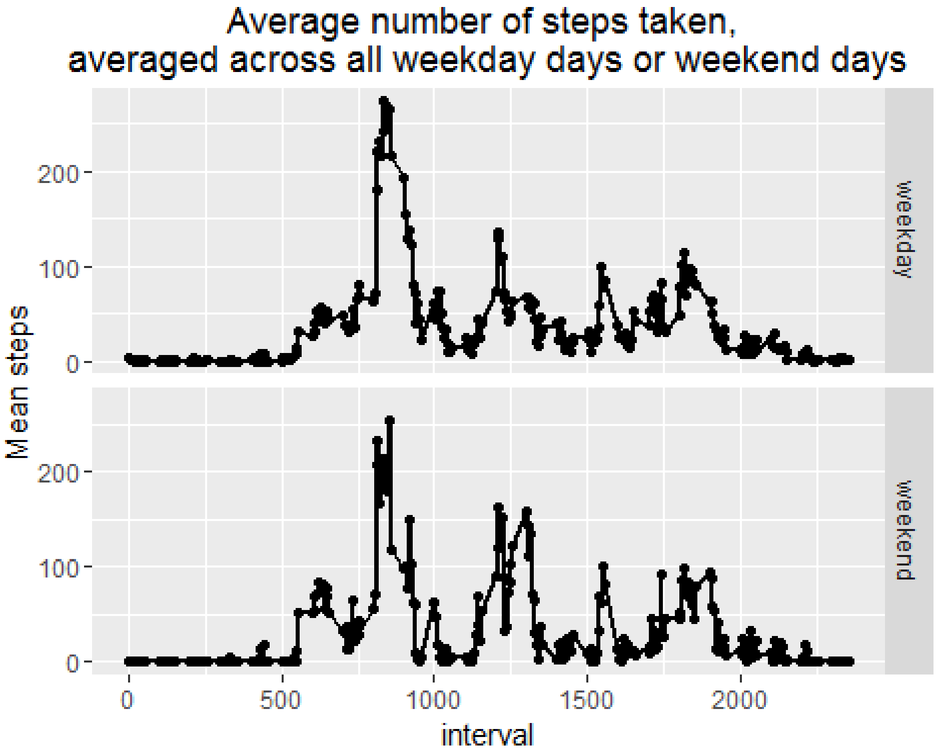
1. Create a new factor variable in the dataset with two levels - "weekday" and "weekend" indicating whether a given date is a weekday or weekend day.

```
new_dt$days <- weekdays(as.Date(new_dt$date))
# find weekend features in the dataset
weekend_feature <- grep("Saturday|Sunday", new_dt$days, ignore.case = T)
# subset data of the weekend
weekend_dt<-  new_dt[weekend_feature, ]
weekend_dt$weekday <- "weekend"

# subset data of the weekday
weekday_dt <- subset(new_dt,new_dt$days!=weekend_feature)

## Warning in new_dt$days != weekend_feature: longer object length is not a
## multiple of shorter object length

weekday_dt$weekday <- "weekday"

# create a new dataset containing 2 new variable "days" and weekday"
# - days: indicates the days in the week
# - weekday: indicate the days are at the "weekend" or "weekday"
new_dt2 <- rbind(weekday_dt, weekend_dt)
```

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). See the README file in the GitHub repository to see an example of what this plot should look like using simulated data.

```
mean_number_steps <- aggregate(steps~ interval+weekday, new_dt2, mean)
g <- qplot(interval, steps, data = mean_number_steps, facets = weekday~.)
```

```
g + geom_line(size = 1) + ylab("Mean steps") + ggtitle("Average number of
steps taken, \n averaged across all weekday days or weekend days ")
```



Discussions and Conclusions

I leave it open!

Acknowledgement

Thank you very much for your effort to review my code!

References

1. R Programming for Data Science, Roger D. Peng.
2. Exploratory Data Analyis with R, Roger D. Peng
3. Report Writing for Data Science in R, Roger D. Peng