Reproducicle Research Course Project 1

Cesar Franca

July 9 2017

R Markdown

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 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.

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). Note: ggplot2 is the chosen plot system.

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.

Library:

```
library(ggplot2)
```

Download the activity.csv data from the provided link, Load and pre-process the data and show any code needed:

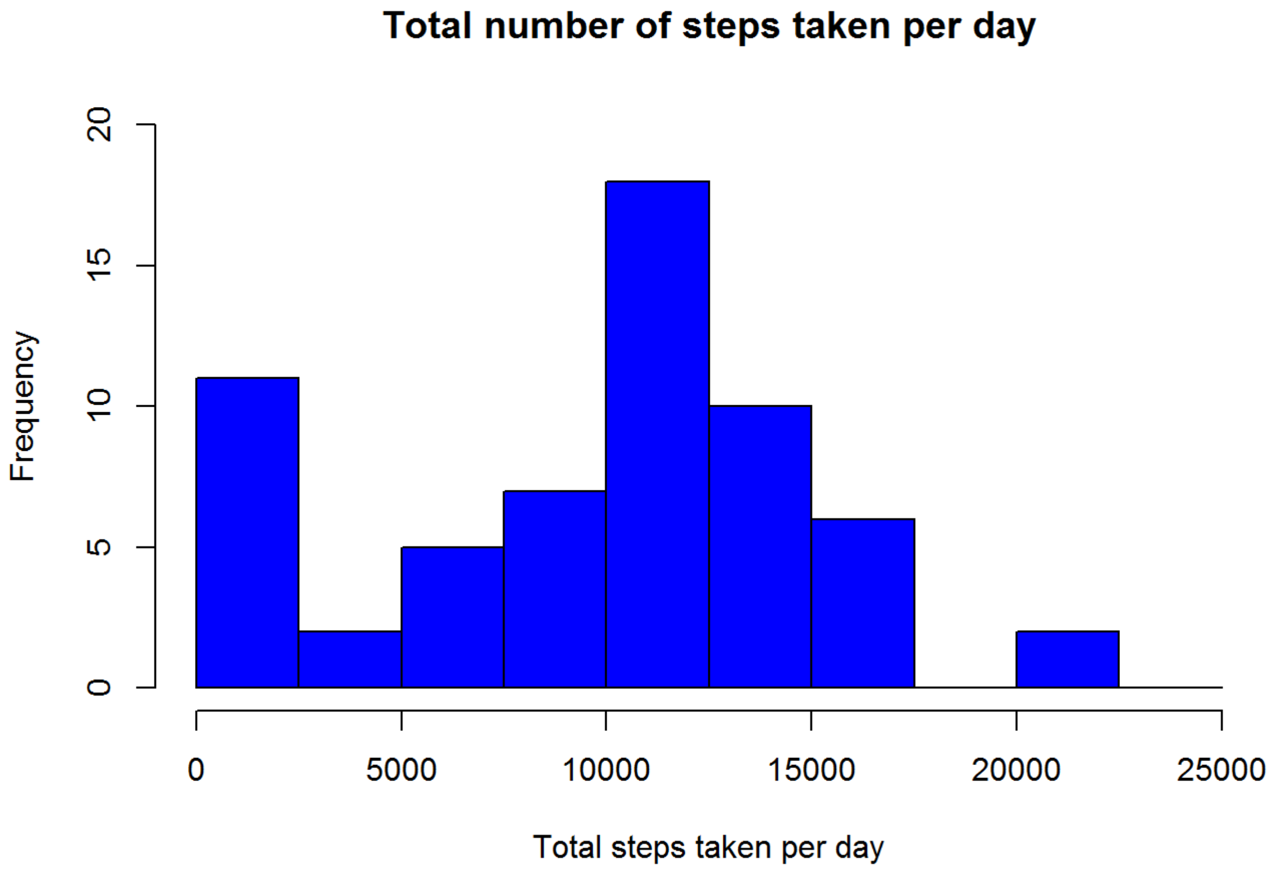
```
download <- download.file("https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip",destfile = "temporary")
unzip("temporary")
unlink("temporary")
activity <- read.csv("activity.csv")
activity$date <- as.POSIXct(activity$date, "%Y-%m-%d")
weekday <- weekdays(activity$date)
activity <- cbind(activity,weekday)
summary(activity)
```

```
##      steps             date               interval     
##  Min.   :  0.00   Min.   :2012-10-01   Min.   :   0.0  
##  1st Qu.:  0.00   1st Qu.:2012-10-16   1st Qu.: 588.8  
##  Median :  0.00   Median :2012-10-31   Median :1177.5  
##  Mean   : 37.38   Mean   :2012-10-31   Mean   :1177.5  
##  3rd Qu.: 12.00   3rd Qu.:2012-11-15   3rd Qu.:1766.2  
##  Max.   :806.00   Max.   :2012-11-30   Max.   :2355.0  
##  NA's   :2304                                          
##           weekday    
##  domingo      :2304  
##  quarta-feira :2592  
##  quinta-feira :2592  
##  sábado       :2304  
##  segunda-feira:2592  
##  sexta-feira  :2592  
##  terça-feira  :2592  
```

Questions

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

```
activity_steps <- with(activity, aggregate(steps, by = list(date), FUN = sum, na.rm = TRUE))
names(activity_steps) <- c("date", "steps")
hist(activity_steps$steps, main = "Total number of steps taken per day", xlab = "Total steps taken per day", col = "blue", y
lim = c(0,20), breaks = seq(0,25000, by=2500))
```



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

Calculate the total number of steps taken per day

If you do not understand the difference between a histogram and a barplot, research the difference between them. Make a histogram of the total number of steps taken each day.

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

```
mean(activity_steps$steps)
```

```
## [1] 9354.23
```

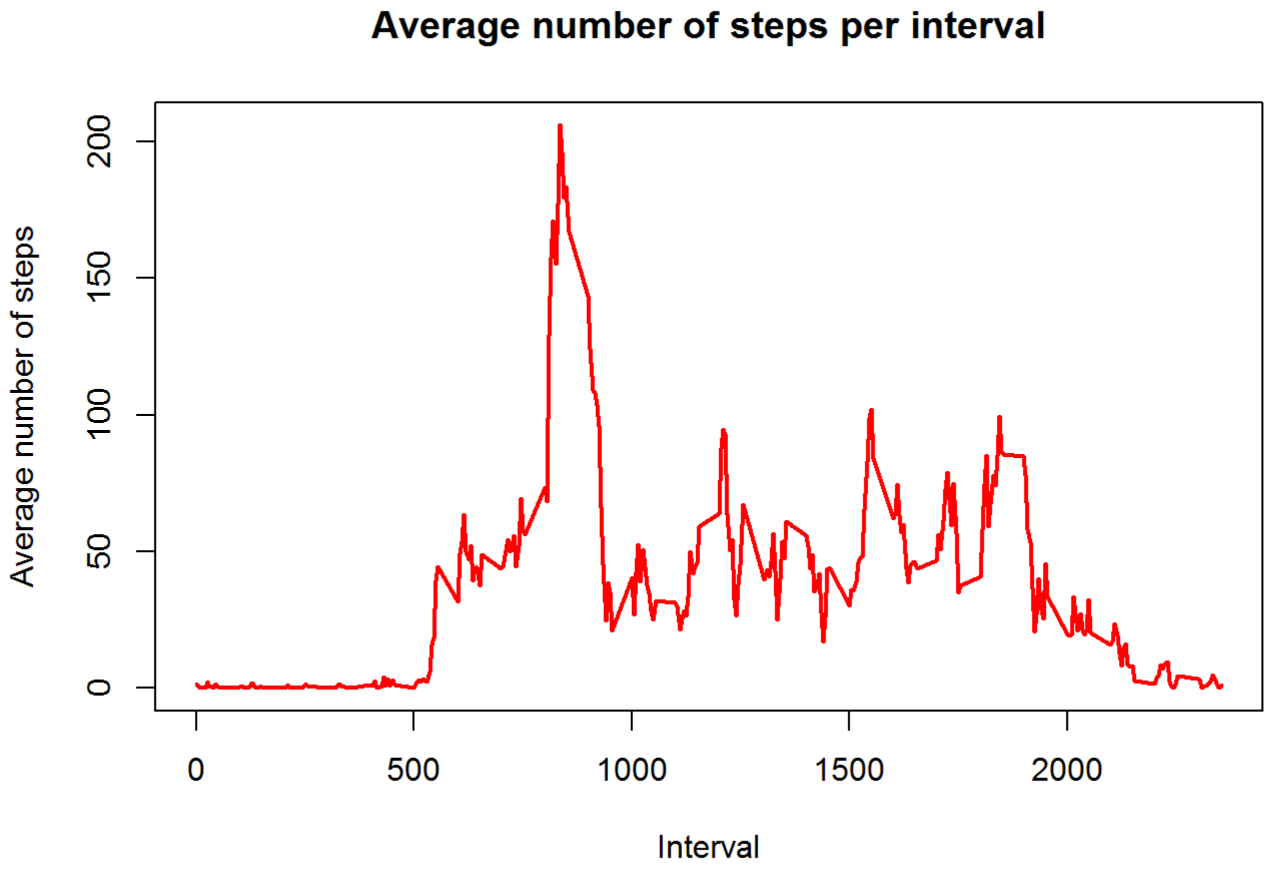
```
median(activity_steps$steps)
```

```
## [1] 10395
```

3. 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)

```
average_daily_activity <- aggregate(activity$steps, by=list(activity$interval), FUN=mean, na.rm=TRUE)
names(average_daily_activity) <- c("interval", "mean")
plot(average_daily_activity$interval, average_daily_activity$mean, type = "l", col="red", lwd = 2, xlab="Interval", ylab="Average number of steps", main="Average number of steps per interval")
```



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

```
average_daily_activity[which.max(average_daily_activity$mean), ]
```

```
##     interval     mean
## 104      835 206.1698
```

Answer: The 5-minute interval “835”, on average across all the days in the dataset, contains the maximum “206.1698” number of steps

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.

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

```
sum(is.na(activity$steps))
```

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

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.

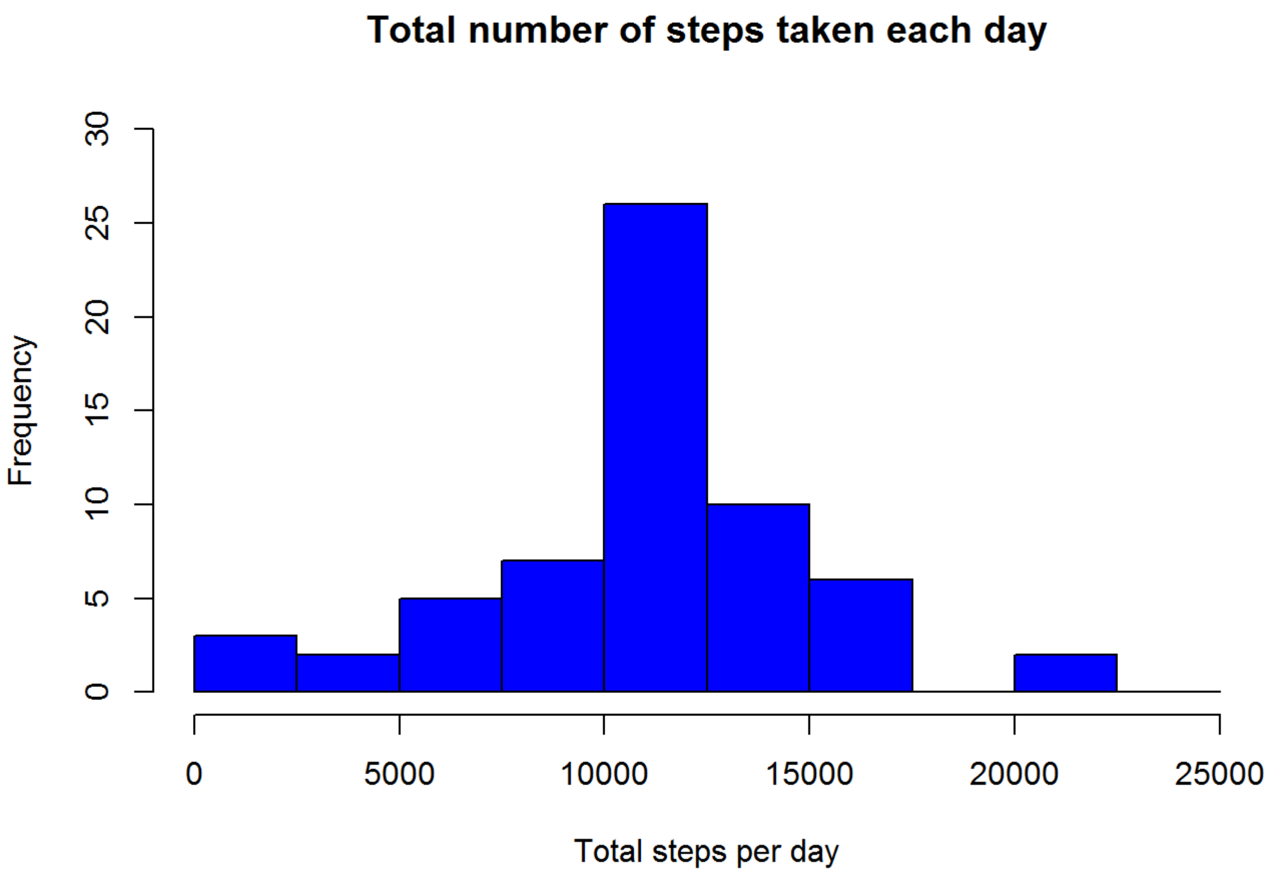
```
imputed_steps <- average_daily_activity$mean[match(activity$interval, average_daily_activity$interval)]
```

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

```
activity_imputed <- transform(activity, steps = ifelse(is.na(activity$steps), yes = imputed_steps, no = activity$steps))
total_steps_imputed <- aggregate(steps ~ date, activity_imputed, sum)
names(total_steps_imputed) <- c("date", "daily_steps")
```

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.

```
hist(total_steps_imputed$daily_steps, col = "blue", xlab = "Total steps per day", ylim = c(0,30), main = "Total number of st
eps taken each day", breaks = seq(0,25000,by=2500))
```



```
mean(total_steps_imputed$daily_steps)
```

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

```
median(total_steps_imputed$daily_steps)
```

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

Do these values differ from the estimates from the first part of the assignment?

Answer: Yes the values are different, in other words, higher with the missing data filled in.

What is the impact of imputing missing data on the estimates of the total daily number of steps?

Answer: The profile of the second histogram is about the same in comparison to the first one but the values are different.

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.

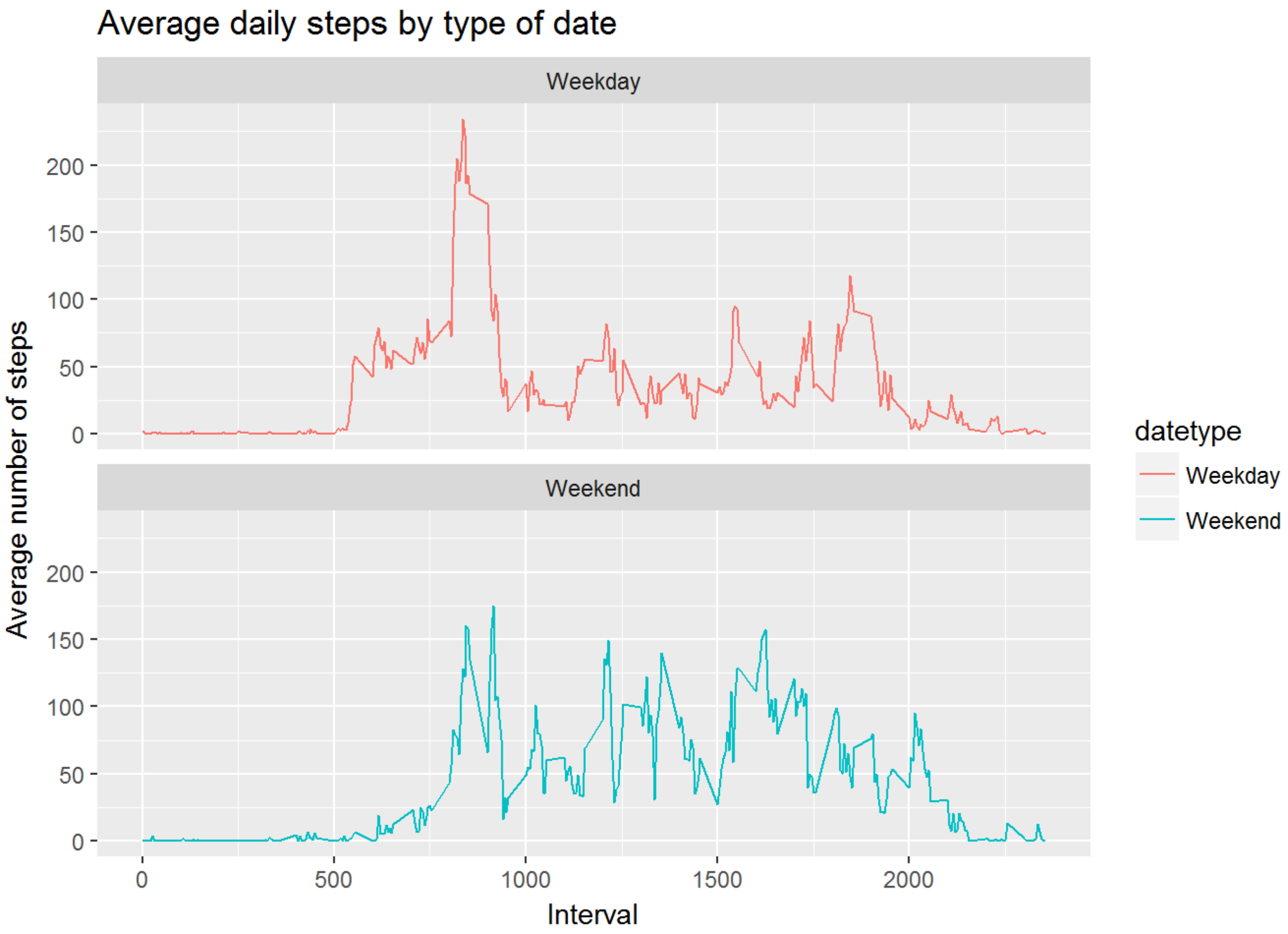
Create a new factor variable in the dataset with two levels - “weekday” and “weekend” indicating whether a given date is a weekday or weekend day.

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.

```
activity$date <- as.Date(strptime(activity$date, format="%Y-%m-%d"))
activity$datetype <- sapply(activity$date, function(x) {
        if (weekdays(x) == "sábado" | weekdays(x) =="domingo")
                {y <- "Weekend"} else
                {y <- "Weekday"}
                y
        })
```

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.

```
activity_by_date <- aggregate(steps~interval + datetype, activity, mean, na.rm = TRUE)
plot<- ggplot(activity_by_date, aes(x = interval , y = steps, color = datetype)) +
       geom_line() +
       labs(title = "Average daily steps by type of date", x = "Interval", y = "Average number of steps") +
       facet_wrap(~datetype, ncol = 1, nrow=2)
print(plot)
```



As can be seen from the above plot during a small period in the weekdays a peak value is reached which does not happen during weekend days. Nevertheless, it also seems that during the weekend days the individual had more intense activities probably due to a completely diffrenent type of activity, more related to entertainment of sport, i.e., maybe during the weekdays the individual was walking or running during a short period of time but seated for most of the time due to work activities and during the weekend days he or she was running or doing another type of a more intense activity.

Note that the `echo = TRUE` parameter was added to the code chunks to show all the R codes.