Reproducible Research Course Project 1

Henrique dos Santos Almeida
January 13, 2017

Course Project 1

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.

Commit containing full submission

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

Loading data

```
#set working directory
#getwd()
#setwd('Documents/R/Coursera/RepResearch/')
#read data
data <- read.csv('activity.csv')
head(data)
```

```
##   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
```

Histogram, mean and median of total Steps taken each day

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

Mean and median of total steps taken each day

```
# Loading ggplot2
library(ggplot2)

# Aggregate steps as interval to get average number of steps in an interval across all days
dataSteps <- aggregate(steps ~ date, data = data, sum, na.rm = TRUE)
head(dataSteps)

##        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

mean(dataSteps$steps)

## [1] 10766.19
```

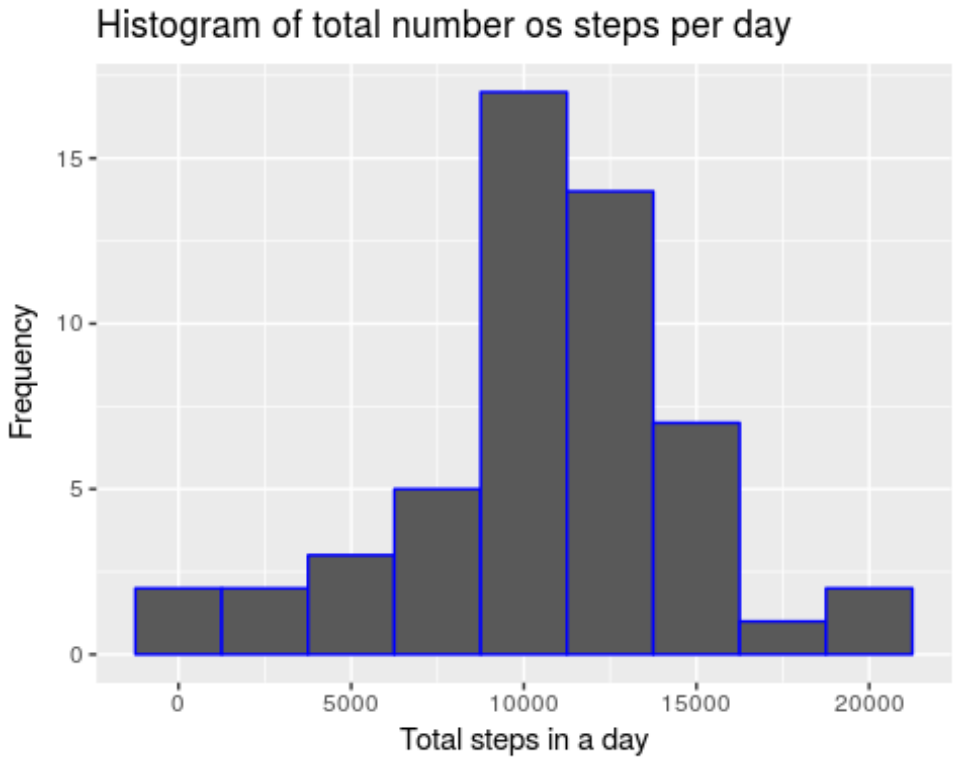
The mean of total steps per day is 10766.19

```
median(dataSteps$steps)

## [1] 10765
```

The median of total steps per day is 10765

Histogram

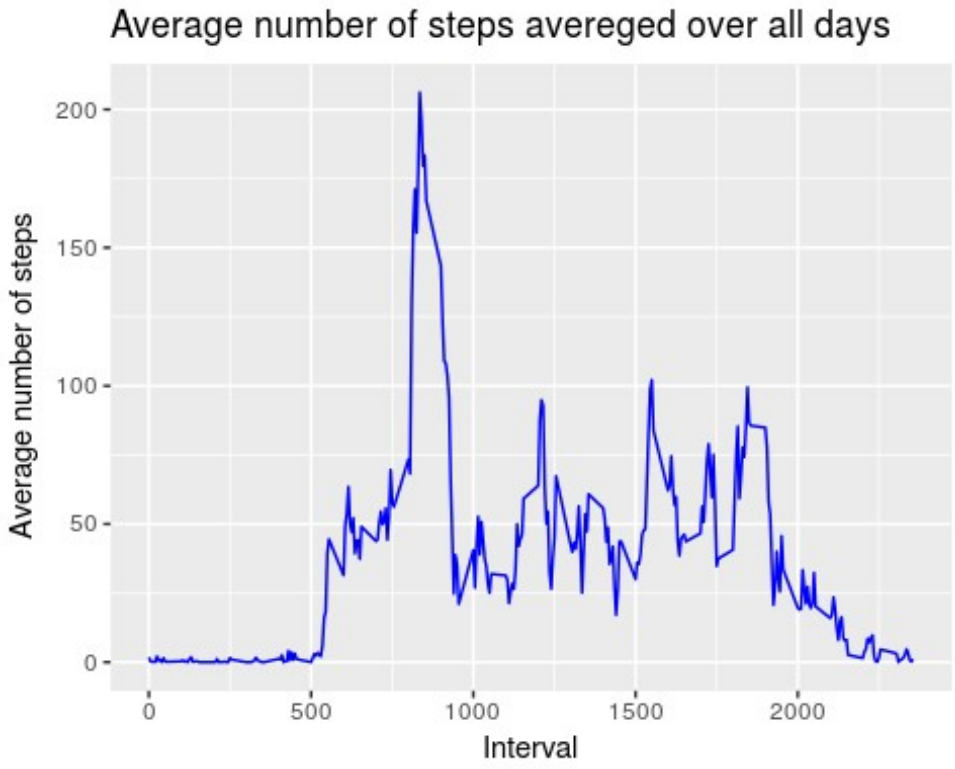
```
ggplot(data=dataSteps, aes(dataSteps$steps)) + geom_histogram(binwidth = 2500,color='blue') + xlab('Total steps in a day') + ylab('Frequency') + ggtitle('Histogram of total number os steps per day')
```



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)
2. Which 5-minute interval, on average across all the days in the dataset, contains the maximum number of steps? ### Time series plot of the average number of steps taken

```
dataInterval <- aggregate(steps ~ interval, data = data, mean, na.rm = TRUE)
head(dataInterval)

##   interval     steps
## 1        0 1.7169811
## 2        5 0.3396226
## 3       10 0.1320755
## 4       15 0.1509434
## 5       20 0.0754717
## 6       25 2.0943396

ggplot(dataInterval,aes(dataInterval$interval,dataInterval$steps)) + geom_line(color='blue') + ggtitle('Average number of steps avereged over all days') + xlab('Interval') + ylab('Average number of steps')
```



5-minutes interval with maximum average number of steps

```
# Using wich.max function to get the row with maximum average of steps and showing data
dataInterval[which.max(dataInterval$steps), ]

##     interval    steps
## 104      835 206.1698
```

The interval with maximum average number of steps is 835

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

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 of missing values in the dataset

```
# Returning number of rows with NA's
sum(is.na(data))

## [1] 2304
```

There are 2304 rows with NA's values

Strategy to fill all missing values

First the data will be copied as dataAux. After that an loop will read from the first line to last line of dataAux and if some step is NA, it will be replaced by the median of the total steps taken per day in that 5-minutes interval. This median can be found in dataInterval object.

```
#Copying data
dataAux <- data
#Performing loop
for (i in 1:nrow(dataAux)){
  #Condition
  if (is.na(dataAux$steps[i])){
    #Saving the interval of that NA step
    interval <- dataAux$interval[i]
    #Saving row id of median interval that have same interval of the NA step
    id <- which(dataInterval$interval == interval)
    #Filling missing values with median of steps taken in that 5-minute
interval
    dataAux$steps[i] <- dataInterval$steps[id]
  }
}
head(dataAux)

##       steps       date interval
## 1 1.7169811 2012-10-01        0
## 2 0.3396226 2012-10-01        5
## 3 0.1320755 2012-10-01       10
## 4 0.1509434 2012-10-01       15
## 5 0.0754717 2012-10-01       20
## 6 2.0943396 2012-10-01       25
```

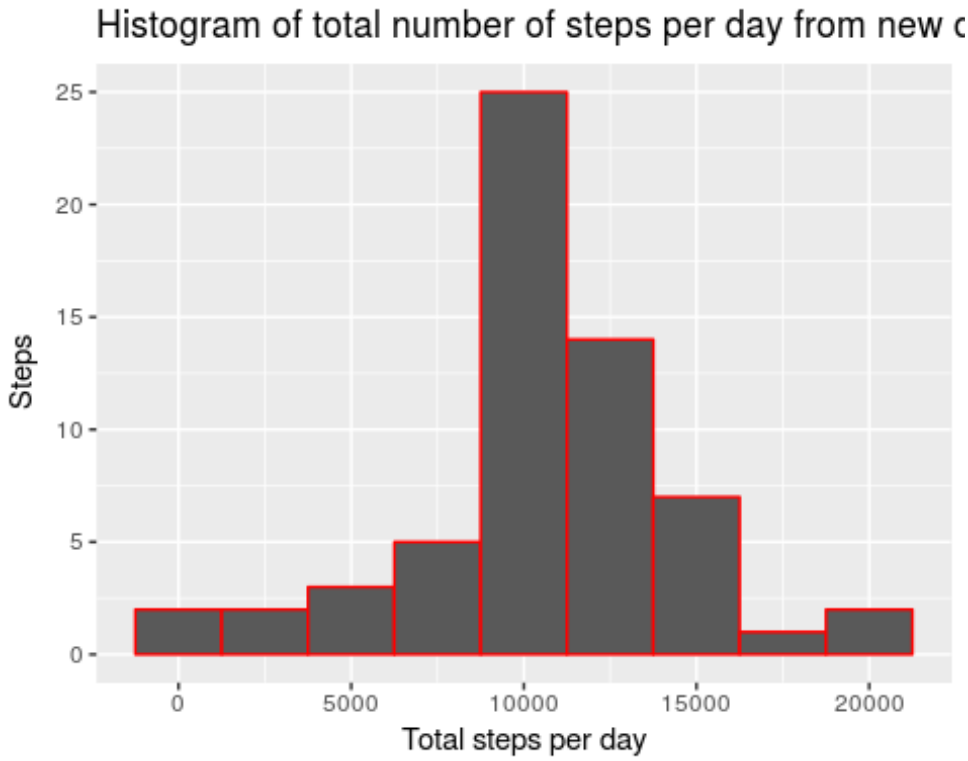
The NA steps has been filled

Histogram

```
# Copying data
dataAuxSteps <- aggregate(steps ~ date , data=dataAux, sum)
head(dataAuxSteps)
```

```
##        date    steps
## 1 2012-10-01 10766.19
## 2 2012-10-02   126.00
## 3 2012-10-03 11352.00
## 4 2012-10-04 12116.00
## 5 2012-10-05 13294.00
## 6 2012-10-06 15420.00
```

```
ggplot(data=dataAuxSteps, aes(dataAuxSteps$steps)) +
geom_histogram(binwidth = 2500,color='red') + xlab('Total steps per day') +
ylab('Steps') + ggtitle('Histogram of total number of steps per day from new
data filled')
```




```
#Mean and median of imputed Data
mean(dataAuxSteps$steps)
```

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

```
median(dataAuxSteps$steps)
```

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

The mean and the median of total number of steps per day from imputed data are 10766.19

```
#Mean and median original Data
mean(dataSteps$steps)

## [1] 10766.19

median(dataSteps$steps)

## [1] 10765
```

The mean of total number of steps per day from original data is 10766.19 and the median is 10765. Only the median has changed a little when filled missing steps values.

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.
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.

Creating a new factor variable in the dataset with two levels – “weekday” and “weekend”

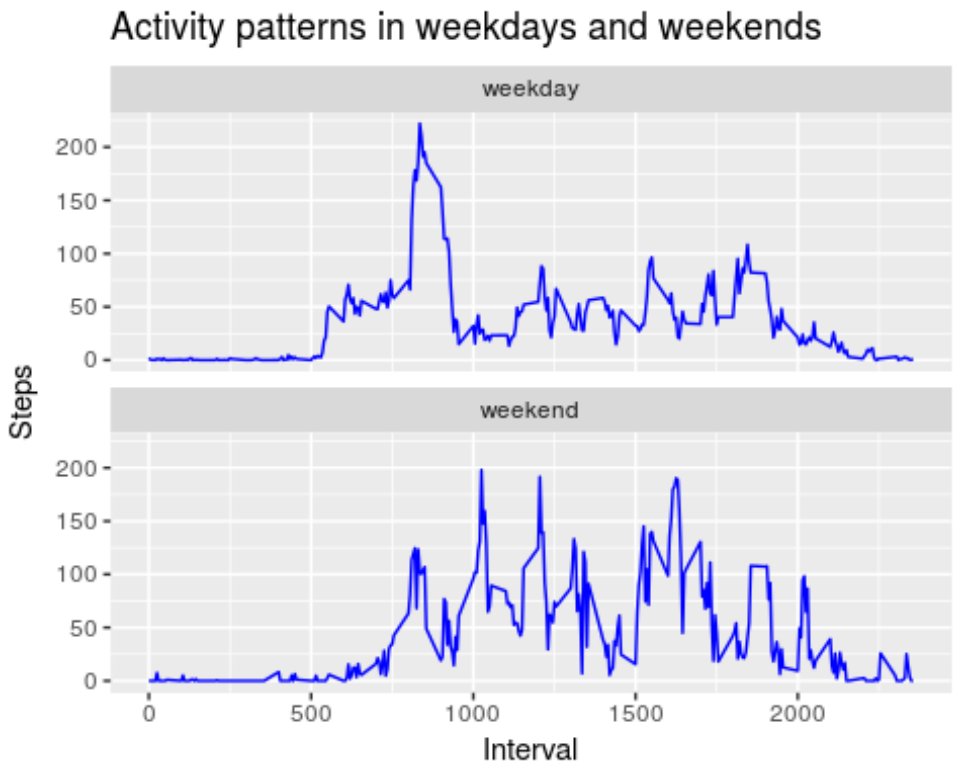
```
#Create new column of the weekday using column date with as.Date function
data$day <- weekdays(as.Date(data$date))
#Create new column of the weekday Type
data$dayType <- "weekday"
#Vizualizing new data
head(data)

##   steps       date interval     day dayType
## 1    NA 2012-10-01        0 segunda weekday
## 2    NA 2012-10-01        5 segunda weekday
## 3    NA 2012-10-01       10 segunda weekday
## 4    NA 2012-10-01       15 segunda weekday
## 5    NA 2012-10-01       20 segunda weekday
## 6    NA 2012-10-01       25 segunda weekday
```

```
# Loop to read all data lines
for (i in 1:nrow(data)){
  # Make daytype as weekend if = Saturday(sabado) or Sunday(domingo)
  if (data$day[i] == "sabado" || data$day[i] == "domingo"){
    data$dayType[i] <- "weekend"
  }
}
# Changing dayType to factor
data$dayType <- as.factor(data$dayType)
# Aggregate steps as interval to get average number of steps in an interval
across all days
dataAux2Steps <- aggregate(steps ~ interval+dayType, data, mean)
head(dataAux2Steps)
```

```
##   interval dayType      steps
## 1        0 weekday 1.97826087
## 2        5 weekday 0.39130435
## 3       10 weekday 0.15217391
## 4       15 weekday 0.17391304
## 5       20 weekday 0.08695652
## 6       25 weekday 1.28260870
```

Analysing differences between weekends and weekdays activity pattern

```
# Making graphic
ggplot(data=dataAux2Steps,aes(interval, steps)) + geom_line(color='blue') +
facet_wrap(~ dayType, ncol=1) + xlab('Interval') + ylab('Steps') +
ggtitle('Activity patterns in weekdays and weekends')
```



Now it is possible to analyse differences of activity patterns between weekdays and weekends. On weekdays there are more steps in 500 to 1000 minutes interval while weekends the maximuns are arround 1000 to 2000 minutes interval.