CourseProject1

Francisco Guzman

10/02/2020

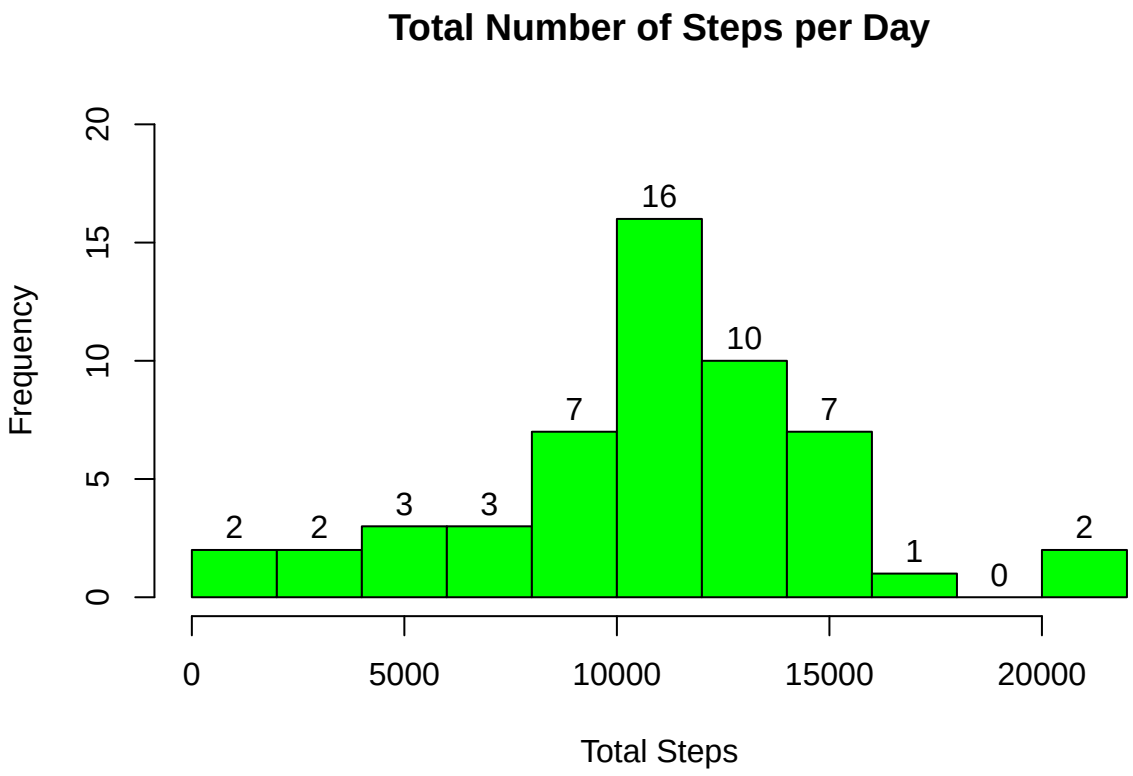
Global options and libraries

Loading and preprocessing the data

```
setwd("~/Learning_R/ReprodResearch/Project1")

zipDataFile <- "repdata_data_activity.zip"
dataFile <- "activity.csv"

if (!file.exists(dataFile) & file.exists(zipDataFile)) {
    unzip (zipfile = zipDataFile)
}

# 1. Load the data (i.e. read.csv())
activity <- read.csv("activity.csv", na.strings = "NA")

# 2. Process/transform the data (if necessary) into a format suitable for your analysis
activity <- transform(activity, date = as.Date(date))
```

What is mean total number of steps taken per day?

```
# 1. Calculate the total number of steps taken per day
total_steps <- aggregate(steps ~ date, activity, sum, na.rm = TRUE)

# 2. Make a histogram of the total number of steps taken each day
hist(total_steps$steps, breaks = 15, col = "green", labels = TRUE, ylim = c(0, 20), xlab = "Total Steps
```




```
# 3. Calculate and report the mean and median of the total number of steps taken per day
# 3.1 Steps mean
mean(total_steps$steps)
```

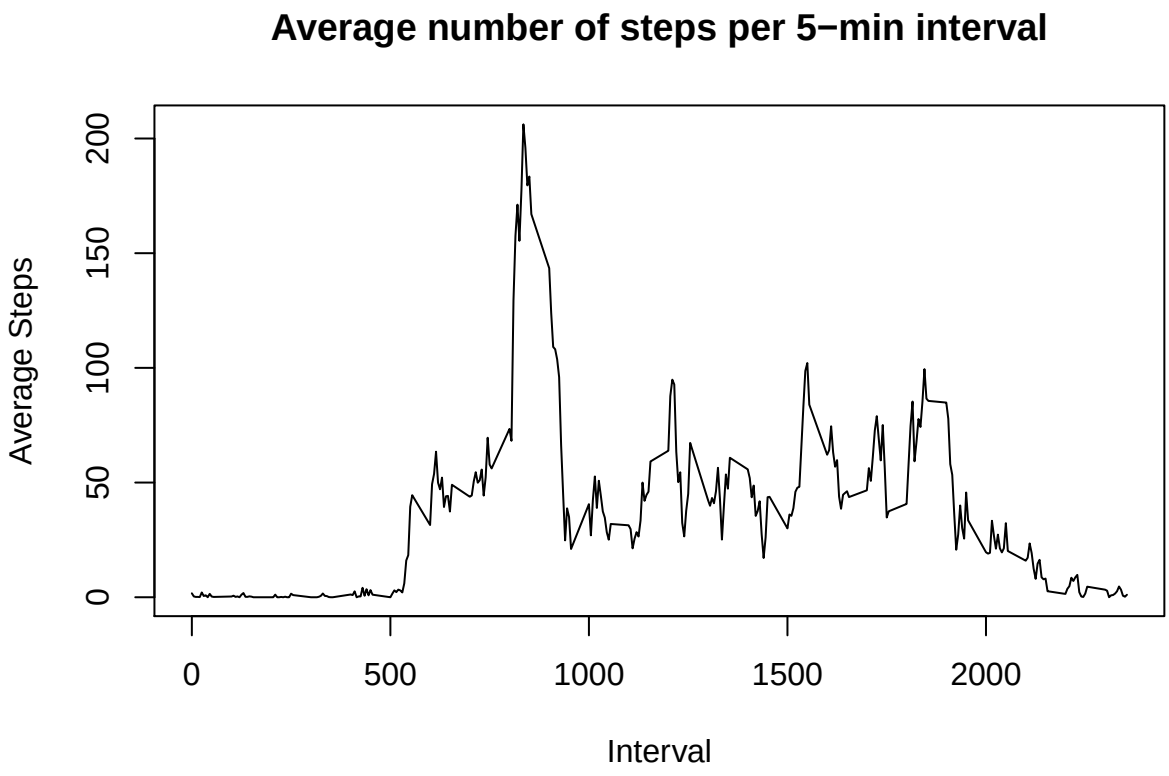
```
## [1] 10766.19
```

```
# 3.2 Steps median
median(total_steps$steps)
```

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

What is the average daily activity pattern?

```
avgDailyActivity <- aggregate(steps ~ interval, activity, mean, na.rm = TRUE)

# 1. Make a time series plot (i.e. type = "l") of the 5-minute interval (x-axis) and the average number
plot(avgDailyActivity$interval, avgDailyActivity$steps, type = "l", main = "Average number of steps per
```




```
# 2. Which 5-minute interval, on average across all the days in the dataset, contains the maximum numbe
print(avgDailyActivity$interval[which(avgDailyActivity$steps == max(avgDailyActivity$steps))])
```

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

Imputing missing values

```
# 1. Calculate and report the total number of missing values in the dataset (i.e. the total number of r
NAIndex <- is.na(activity$steps)
sum(NAIndex)
```

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

```
# 2. Devise a strategy for filling in all of the missing values in the dataset. Strategy: replace NA's
```

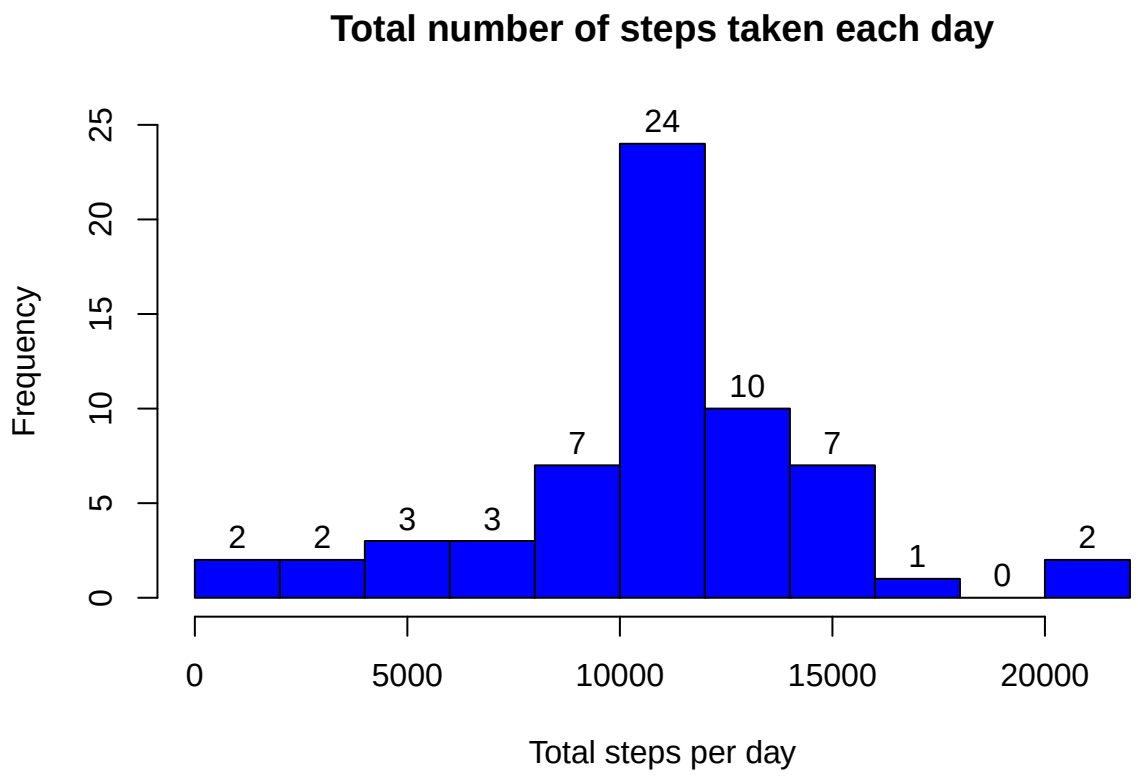
3. Create a new dataset that is equal to the original dataset but with the missing data filled in.

```
imputedData <- activity

for (i in 1:length(imputedData$steps)) {
  if (is.na(imputedData$steps[i])) {
    imputedData$steps[i] <- avgDailyActivity$steps[avgDailyActivity$interval == imputedData$interval[i]]
  }
}
```

```
totalStepsImputed <- aggregate(steps ~ date, imputedData, sum)
names(totalStepsImputed) <- c("date", "dailySteps")
```

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?

```
hist(totalStepsImputed$dailySteps, col = "blue", labels = TRUE, xlab = "Total steps per day", ylim = c(0
```




```
# 4.1 Calculate and report the mean
mean(totalStepsImputed$dailySteps)
```

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

```
# 4.2 Calculate and report the median
median(totalStepsImputed$dailySteps)
```

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

```
# A: mean and median values converge to the mean value in the 1st part of the assignment
```

Are there differences in activity patterns between weekdays and weekends?

```
# 1. Create a new factor variable in the dataset with two levels - "weekday" and "weekend" indicating w
activity$date <- as.Date(strptime(activity$date, format = "%Y-%m-%d"))
activity$datetype <- sapply(activity$date, function(x) {
        if (weekdays(x) == "Saturday" | weekdays(x) == "Sunday")
                {y <- "Weekend"}
        else
                {y <- "Weekday"}
                y
        })

activityByDate <- aggregate(steps ~ interval + datetype, activity, mean, na.rm = TRUE)

# 2. Make a panel plot containing a time series plot (i.e. type = "l") of the 5-minute interval (x-axis)

plot <- ggplot(activityByDate, 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)
```