Peer-graded Assignment: Course Project 1

Andrea Ballestero

6/27/2020

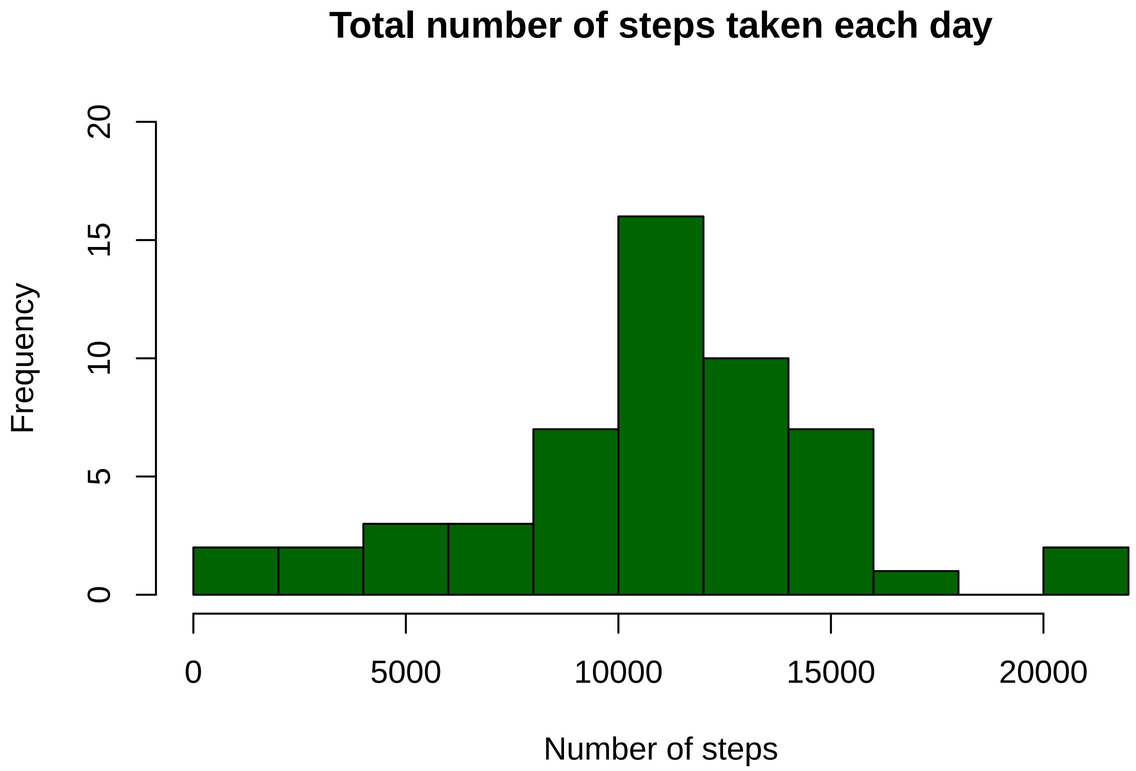
Analysis of activity monitoring data

Loading and preprocessing the data

```
activity <- read.csv("~/R/Coursera/ProgAssign5_1/activity.csv")
```

What is the mean total number of steps taken per day?

```
library(dplyr)
total_steps <- summarise(group_by(activity, date), total = sum(steps, na.rm = FALSE))

hist(total_steps$total, breaks = 8,
     xlim = c(0, 22000), col = "dark green",
     main = "Total number of steps taken each day",
     xlab = "Number of steps",
     ylim = c(0, 20))
```

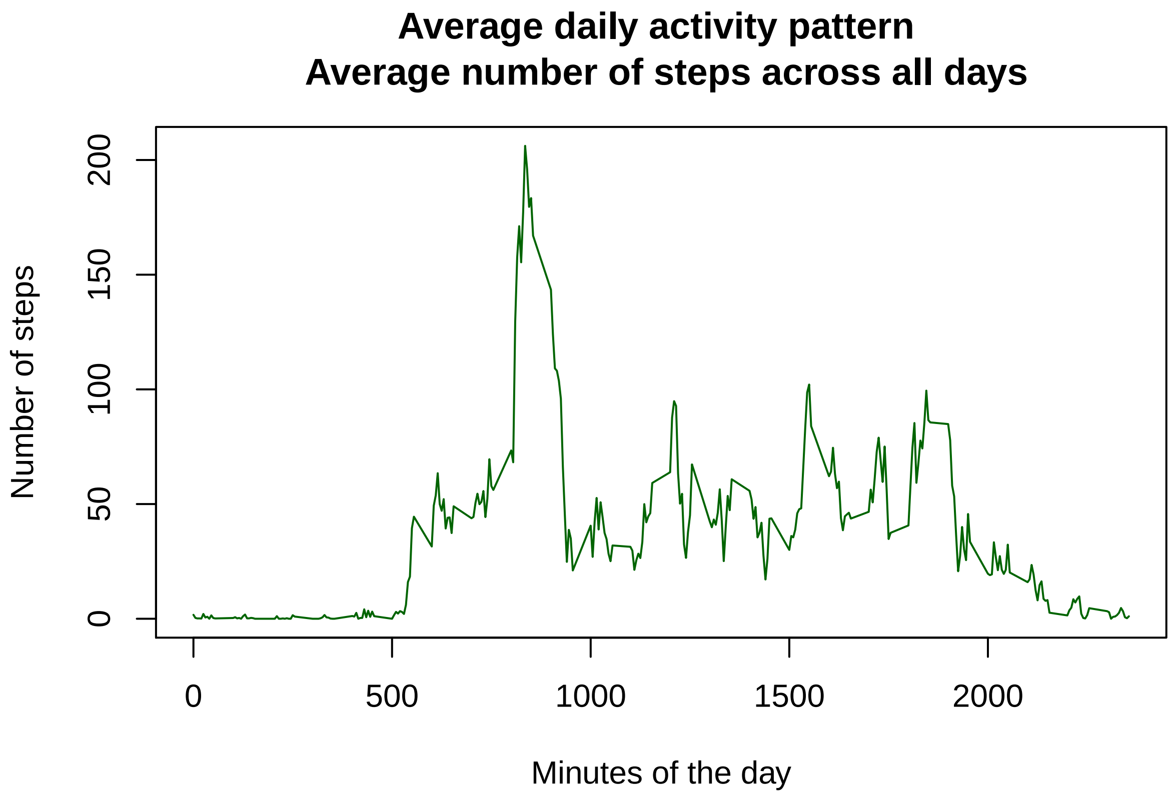


```r
library(formattable)
mean_steps <- comma(mean(total_steps$total, na.rm = TRUE), digit = 2)
median_steps <- comma(median(total_steps$total, na.rm = TRUE), digit = 2)
```

The mean of the total number of steps taken per day is 10,766.19 and the median of the total number of steps taken per day is 10,765.00.

What is the average daily activity pattern?

```r
library(lubridate)
activity <- mutate(activity, date = date(date))
five_min <- summarise(group_by(activity, interval), total = mean(steps, na.rm = TRUE))
five_min <- mutate(five_min, mean_min = comma(total, digits = 2))

plot(five_min$interval, five_min$mean_min, type = "l",
     col = "dark green",
     main = "Average daily activity pattern \n Average number of steps across all days",
     xlab = "Minutes of the day",
     ylab = "Number of steps")
```



```
max_steps <- max(five_min$mean_min)
interv_max <- five_min[five_min$mean_min == max_steps, 1]
interv_max2 <- interv_max + 5
```

The 5-minute interval that contains the maximum number of steps is 835 - 840.

Imputing missing values

```
na <- activity[is.na(activity) == TRUE,]
dim_na <- dim(na)
rows_na <- comma(dim_na[1], digits = 0)
```

The total number of missing values in the dataset is 2,304.

```
days_na <- unique(na$date)
```

The days which have missing values are: 2012-10-01, 2012-10-08, 2012-11-01, 2012-11-04, 2012-11-09, 2012-11-10, 2012-11-14, 2012-11-30.

The following procedure fills the empty data by completing it with the mean for the 5-minute interval and creates a new database called new_data:

```
days_na <- data.frame(rep(unique(na$date), 472))
colnames(days_na) <- "days"
days_na <- arrange(days_na, days)
days_na <- mutate(days_na, interval = rep(seq(0, 2355, 5), 8))
days_na <- right_join(days_na, five_min, "interval")
colnames(days_na) <- c("date", "interval", "steps2", "steps")
```

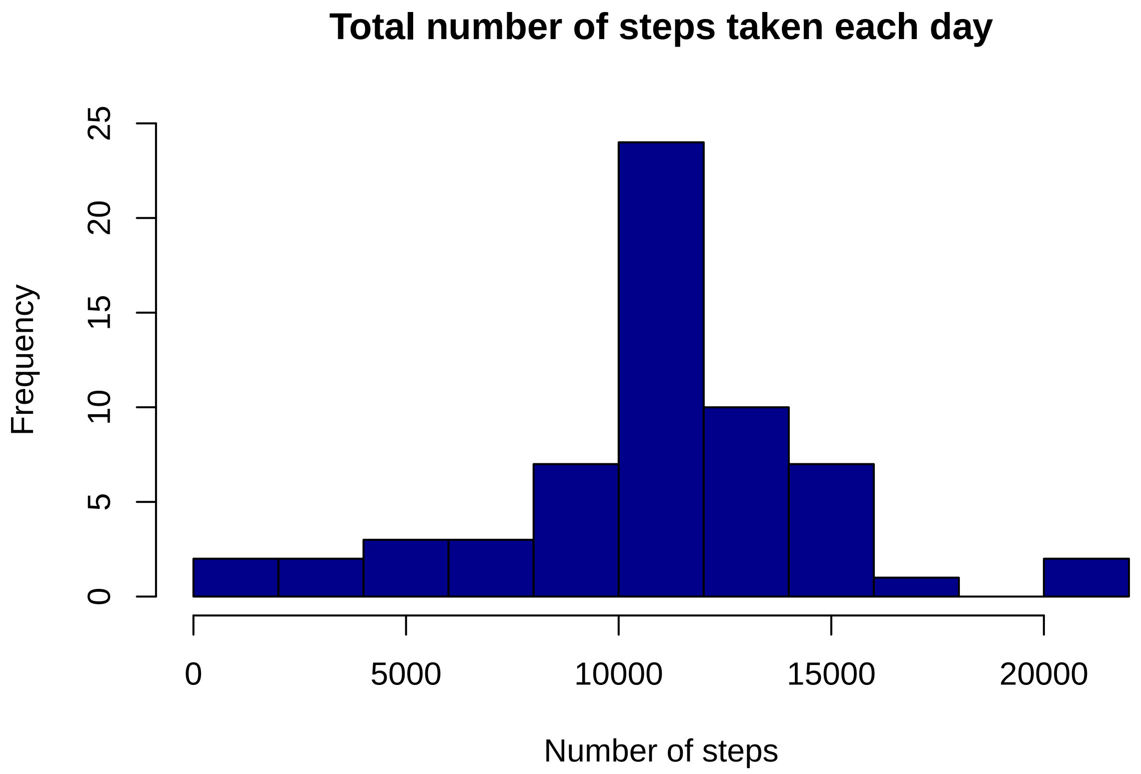
```
new_data <- bind_rows(activity, days_na)
```

```
## Warning in bind_rows_(x, .id): Vectorizing 'formattable' elements may not
## preserve their attributes
```

```
new_data <- select(new_data, c("steps", "date", "interval"))
new_data <- new_data[!is.na(new_data$steps),]
new_data <- arrange(new_data, date)
new_data <- mutate(new_data, steps = comma(steps, digits = 2))
```

The histogram of the total number of steps taken each day with the new data base is the following:

```
new_total_steps <- summarise(group_by(new_data, date), total = sum(steps))

hist(new_total_steps$total, breaks = 8,
     xlim = c(0, 22000), col = "dark blue",
     main = "Total number of steps taken each day",
     xlab = "Number of steps",
     ylim = c(0, 25))
```



```
new_mean_steps <- comma(mean(new_total_steps$total), digit = 2)
new_median_steps <- comma(median(new_total_steps$total), digit = 2)

diff <- new_median_steps - median_steps
perc <- diff/median_steps*100
```

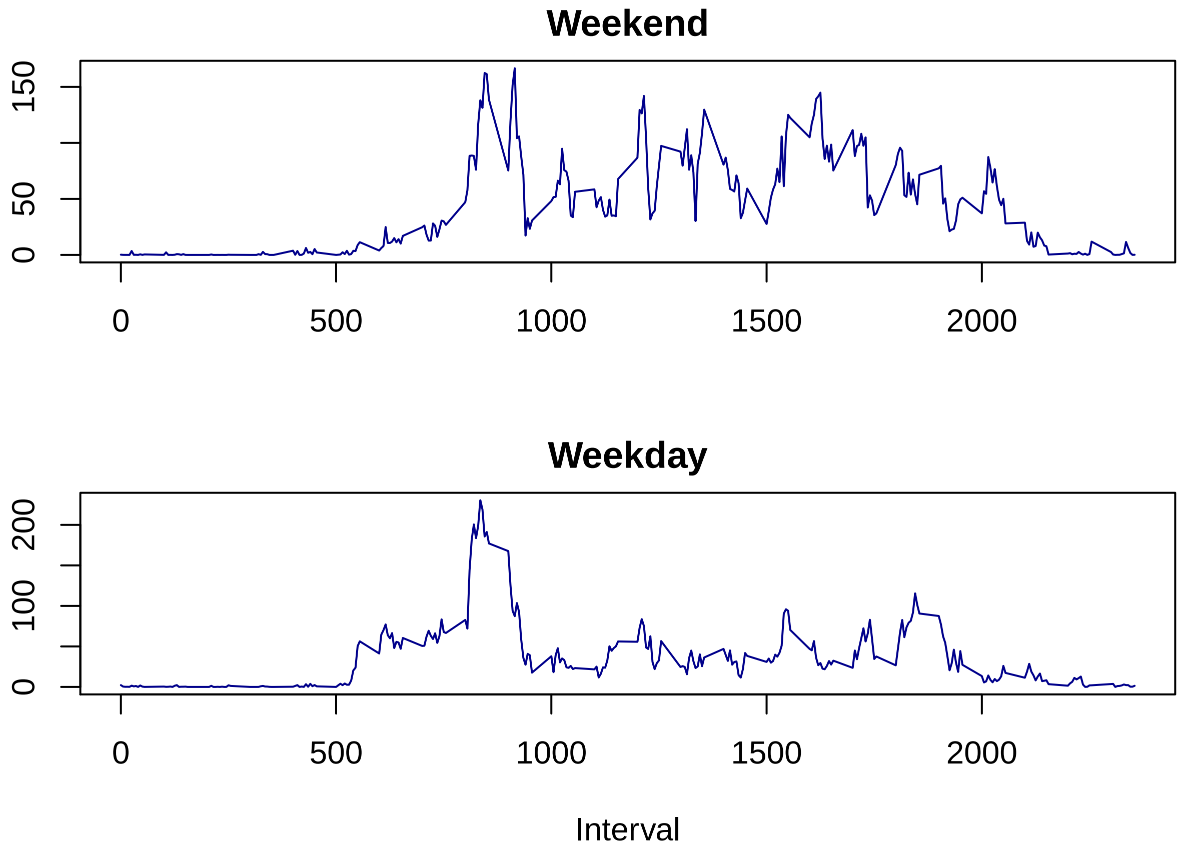
Now, the mean of the total number of steps taken per day is 10,766.19 and the median of the total number of steps taken per day is 10,766.19.

The mean is the same as the median now, whereby before the mean was a little smaller than the median. The difference between the median now minus the median before is: 1.19 steps, which accounts for 0.01% of the initial median.

When using the imputing missing data on the estimates of the total daily number of steps, the frequency of the steps between 10.000 and 12.000 is greater (over 20%, whereby before it was under 20%).

Are there differences in activity patterns between weekdays and weekends?

```r
new_data <- mutate(new_data, day = weekdays(date))
new_data <- mutate(new_data, factor_date = if_else(day %in% c("Saturday", "Sunday"), "weekend", "weekday

weekday_df <- new_data[new_data$factor_date == "weekday",]
weekend_df <- new_data[new_data$factor_date == "weekend",]

weekday_five_min <- summarise(group_by(weekday_df, interval), total = mean(steps))
weekday_five_min <- mutate(weekday_five_min, mean_min = comma(total, digits = 2))

weekend_five_min <- summarise(group_by(weekend_df, interval), total = mean(steps))
weekend_five_min <- mutate(weekend_five_min, mean_min = comma(total, digits = 2))

par(mar = c(4, 2, 2, 2))
par(mfrow = c(2, 1))

plot(weekend_five_min$interval, weekend_five_min$mean_min, type = "l",
     col = "dark blue",
     main = "Weekend",
     xlab = "",
     ylab = "Number of steps")

plot(weekday_five_min$interval, weekday_five_min$mean_min, type = "l",
     col = "dark blue",
     main = "Weekday",
     xlab = "Interval",
     ylab = "Number of steps")
```


Weekend

Weekday

As seen on the panel plot, there's more movement on weekends than on weekdays.