Reproducible Research: Peer Assessment 1

Loading and preprocessing the data

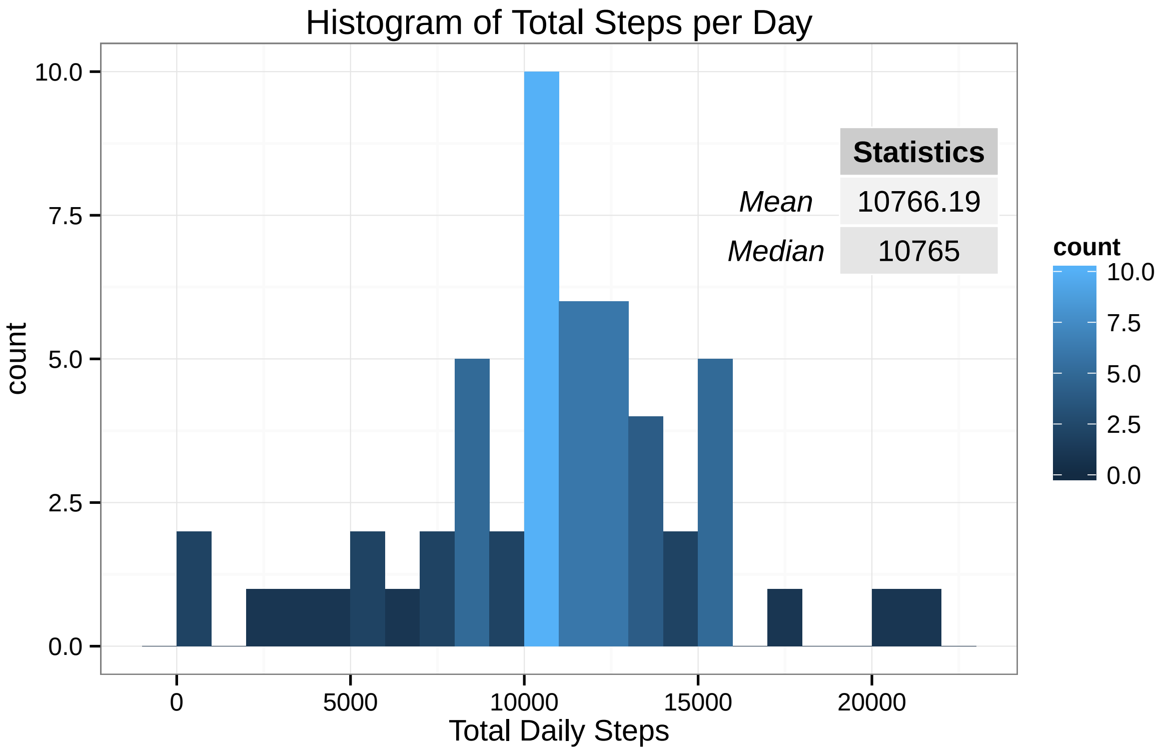
```
# Load Data from Zip File
activityData <- read.csv((unz("activity.zip", "activity.csv")), header = TRUE)
# Load required libraries
library(ggplot2)
library(gridExtra)
```

What is mean total number of steps taken per day?

```
# Aggregate function (Stats Package) was used to sum steps by date
dailySteps <- aggregate(steps ~  date, data  = activityData,
                        FUN = "sum", na.rm = TRUE )

#Plot Histogram
statsTable <- data.frame("Statistics" = c(mean(dailySteps$steps),
                                          median(dailySteps$steps)))
rownames(statsTable) <- c("Mean", "Median")

ggplot(data = dailySteps, aes(steps)) +  theme_bw() +
  geom_histogram(aes(fill = ..count..), binwidth = 1000) +
  ggtitle("Histogram of Total Steps per Day") +
  labs(x = "Total Daily Steps") +
  annotation_custom(tableGrob(signif(statsTable, digits = 7)),
                    xmin = 21100, xmax = 18000,
                    ymin = 7.5, ymax = 8)
```



```
# Calculate Mean and Median
meanNum <- mean(dailySteps$steps, na.rm = TRUE)
meanNum
```

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

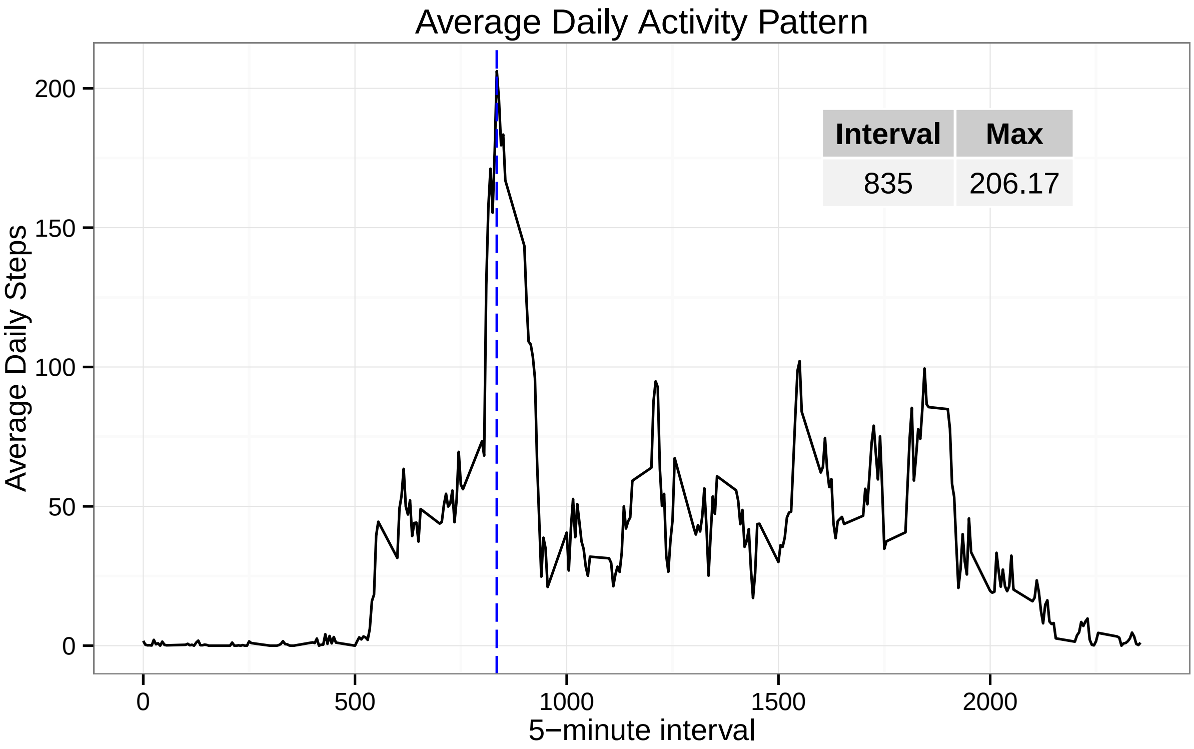
```
# Calculate Median
medNum <- median(dailySteps$steps, na.rm = TRUE)
medNum
```

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

The mean and median are given above by both the script, and the graph.

What is the average daily activity pattern?

```
# Aggregate function (Stats Package) was used to average steps by interval
dailyAverage <- aggregate(steps ~ interval, data = activityData,
                      FUN = mean, na.rm = TRUE)
# Plot
maxLocation <- dailyAverage[dailyAverage$steps == max(dailyAverage$steps),
                            "interval"]
maxTable <- data.frame("Interval" = maxLocation,
                       "Max" = max(dailyAverage$steps))
ggplot(data = dailyAverage, aes(x = interval, y = steps)) + theme_bw() +
  geom_line() +
  ggtitle("Average Daily Activity Pattern") +
  labs(x = "5-minute interval", y = "Average Daily Steps") +
  geom_vline(xintercept = maxLocation,
               colour="blue", linetype = "longdash") +
  annotation_custom(tableGrob(signif(maxTable, digits = 6), rows = NULL),
                    xmin = 2000, xmax = 1800,
                    ymin = 150, ymax = 200)
```



Interval	Max
835	206.17

```
# Max
max(dailyAverage$steps)
```

```
## [1] 206.1698
```

The plot describes the average daily activity pattern, while the blue vertical line indicates the max of the series.

Imputing missing values

To reduce bias due to the presence of missing values, missing values are inputed using the mean value for the 5-minute intervals.

```
# Missing Values Table
missing.values <- is.na(activityData$steps)
table(missing.values)
```

```
## missing.values
## FALSE  TRUE
## 15264  2304
```

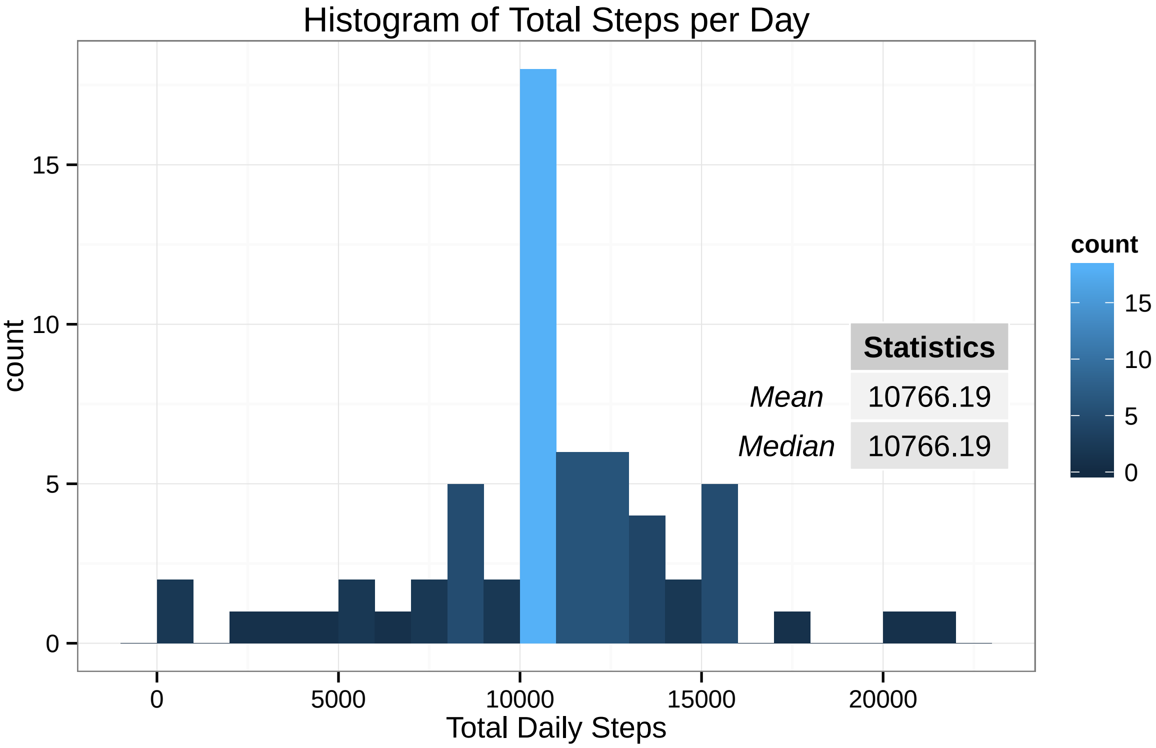
```
# Imputed values using the mean for each 5-minute interval.
# Values already calculated for previous question
impData <- activityData
impData$steps <- mapply(function(x, y){if(is.na(x)){
                            dailyAverage[dailyAverage$interval == y, "steps"]
                          }else x}, impData$steps, impData$interval)
table(is.na(impData$steps))
```

```
##
## FALSE
## 17568
```

```
# Make Histrogram, same steps as with question 1
# Aggregate function (Stats Package) was used to sum steps by date
impdailySteps <- aggregate(steps ~  date, data  = impData, FUN = "sum" )

#Plot Histogram
statsTable <- data.frame("Statistics" = c(mean(impdailySteps$steps),
                                          median(impdailySteps$steps)))
rownames(statsTable) <- c("Mean", "Median")

ggplot(data = impdailySteps, aes(steps)) +  theme_bw() +
  geom_histogram(aes(fill = ..count..), binwidth = 1000) +
  ggtitle("Histogram of Total Steps per Day") +
  labs(x = "Total Daily Steps") +
  annotation_custom(tableGrob(signif(statsTable, digits = 7)),
                    xmin = 21100, xmax = 18000,
                    ymin = 7.5, ymax = 8)
```



```
# Calculate Mean and Median
meanNum <- mean(impdailySteps$steps)
meanNum
```

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

```
# Calculate Median
medNum <- median(impdailySteps$steps)
medNum
```

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

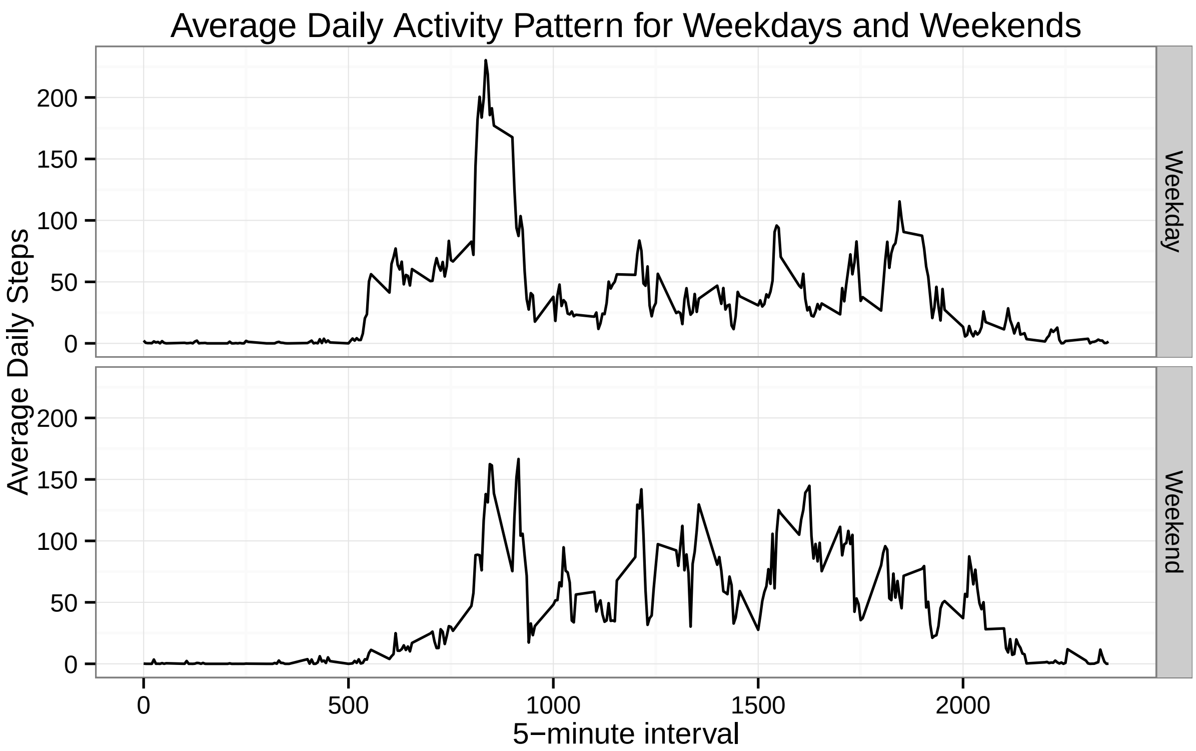
Using this imputation method, the mean was the same, but the median increased. The median is the same as the mean after the imputation. The change is due to the imputation method used.

Are there differences in activity patterns between weekdays and weekends?

```r
# Missing Values Table
impData$weekday <- weekdays(as.Date(as.character(impData$date)))
impData$identDay <- ifelse(impData$weekday %in% c("Saturday", "Sunday"),
                           "Weekend", "Weekday")
weekAverage <- aggregate(steps ~ interval + identDay, data = impData, mean)

# Plot
maxLocation <- dailyAverage[dailyAverage$steps == max(dailyAverage$steps),
                            "interval"]
maxTable <- data.frame("Interval" = maxLocation,
                       "Max" = max(dailyAverage$steps))

ggplot(weekAverage, aes(interval, steps)) +  theme_bw()+
  geom_line() +
  facet_grid(identDay ~ .) +
  ggtitle("Average Daily Activity Pattern for Weekdays and Weekends") +
  labs(x = "5-minute interval", y = "Average Daily Steps")
```



The distribution is clearly different for the weekend and weekday. The max of the series is higher in weekdays, than in the weekends. Although for many of the right handed intervals, the average seems to be higher in the weekends than in the weekdays.