Reproducible Research: Peer Assessment 1

Joseph R. Taylor

Sunday, October 19, 2014

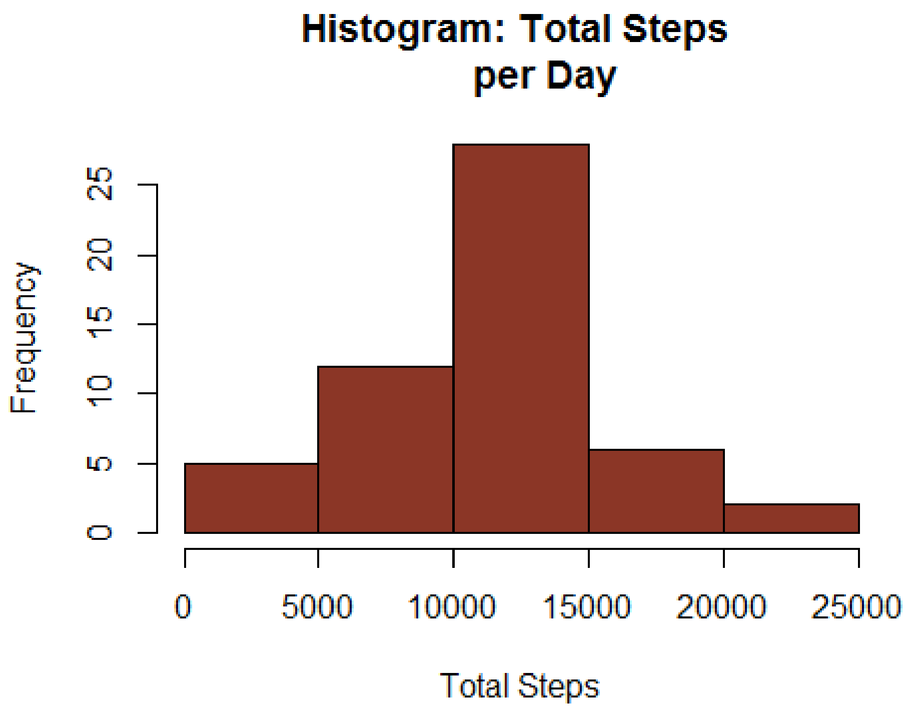
Load and Pre-Process the data

```
# load the data file
thedata <- read.csv("activity.csv")
# process data to convert the date column to a date class
thedata$date <- as.Date(thedata$date)
```

What is mean total number of steps taken per day?

Histogram the total number of steps taken each day

```
# make a new dataframe without NA values
cleanData <- thedata[!is.na(thedata$steps), ]
# load plyr library
library(plyr)
# group values by date
groupedByDate <- ddply(cleanData, ~date, summarise, sum = sum(steps))
# construct the histogram
hist(groupedByDate$sum, xlab = "Total Steps", main = "Histogram: Total Steps
     per Day", col = "tomato4")
```



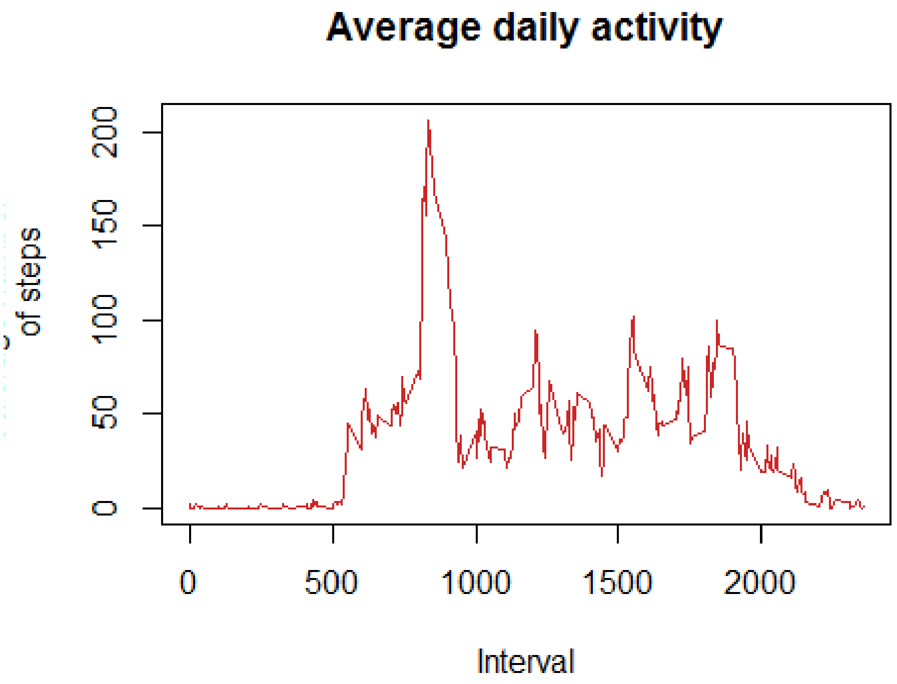
Calculate and display the mean and median total number of steps taken per day

```
# Mean
mean(groupedByDate$sum)

## [1] 10766

# Median
median(groupedByDate$sum)

## [1] 10765
```

What is the average daily activity pattern?

Make a time series plot of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all days (y-axis)

```
groupedByInterval <- ddply(cleanData, ~interval, summarise, mean =
mean(steps))
with(groupedByInterval, plot(interval, mean, type = "l", ylab = "Average
number
        of steps", xlab = "Interval", main = "Average daily activity",
        col = "firebrick3"))
```



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

```
# Find the maximum value
maxVal <- max(groupedByInterval$mean)
print(maxVal)

## [1] 206.2

# Find the line containing the maximum value
maxLine <- groupedByInterval[groupedByInterval$mean == maxVal, ]
# Find the corresponding interval
maxInterval <- maxLine$interval
print(maxInterval)

## [1] 835
```

The maximum number of steps (on average across all the days) is 206.1698. It is contained in interval 835.

Imputing Missing Values

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

```
# Calculate the sum of missing values
sum(is.na(thedata$steps))
```

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

Devise a strategy for filling in all of the missing values in the dataset.

Strategy: Five-minute interval means

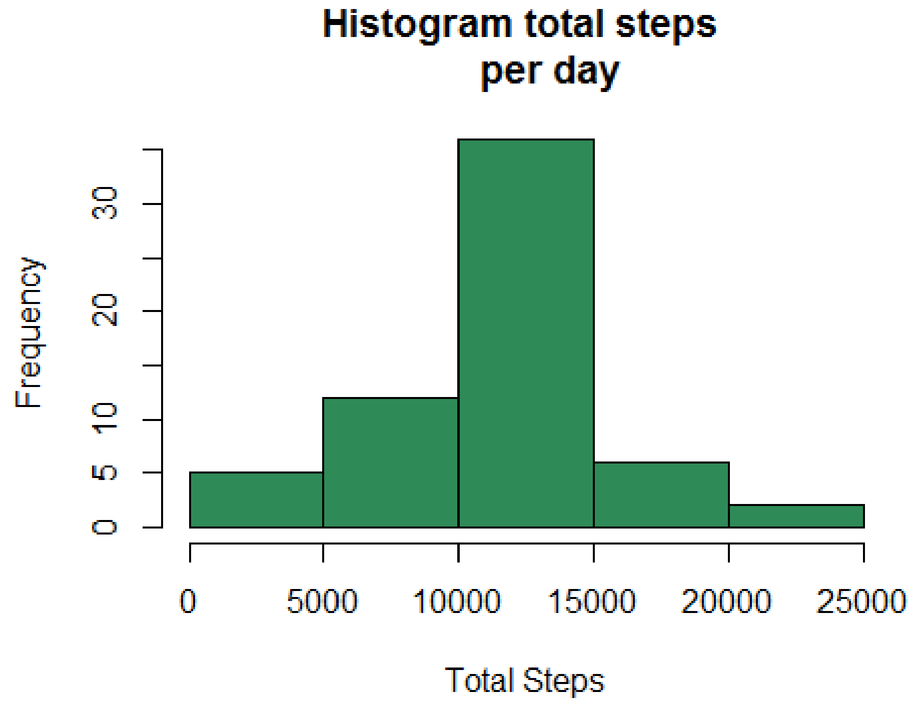
Create a new dataset that is equal to the original dataset but with the missing data filled in

```r
# Create a new dataset
newdata <- thedata
# Locate tha NAs
missingsteps <- is.na(newdata$steps)
# Convert interval(s) to factor(s)
newdata$interval <- factor(newdata$interval)
groupedByInterval$interval <- factor(groupedByInterval$interval)

# Fill newdata on missing values wherever a missing value appears.
# Fill missing data from groupedByInterval$mean column (steps).
newdata[missingsteps, "steps"] <- groupedByInterval[newdata[missingsteps,
        "interval"], "mean"]
```

Make a histogram of the total number of steps taken each day.

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

```r
# group values by date
groupedByDate2 <- ddply(newdata, ~date, summarise, sum = sum(steps))

# construct the histogram
hist(groupedByDate2$sum, xlab = "Total Steps", main = "Histogram total steps
     per day", col = "seagreen")
```



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

```
# Calculate mean value
mean(groupedByDate2$sum)

## [1] 10766
```

The mean does not increase.

```
# Calculate median value
median(groupedByDate2$sum)

## [1] 10766
```

The median increases by one.

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

There is negligible impact.

Are There Differences in Activity Patterns Between Weekdays and Weekends?

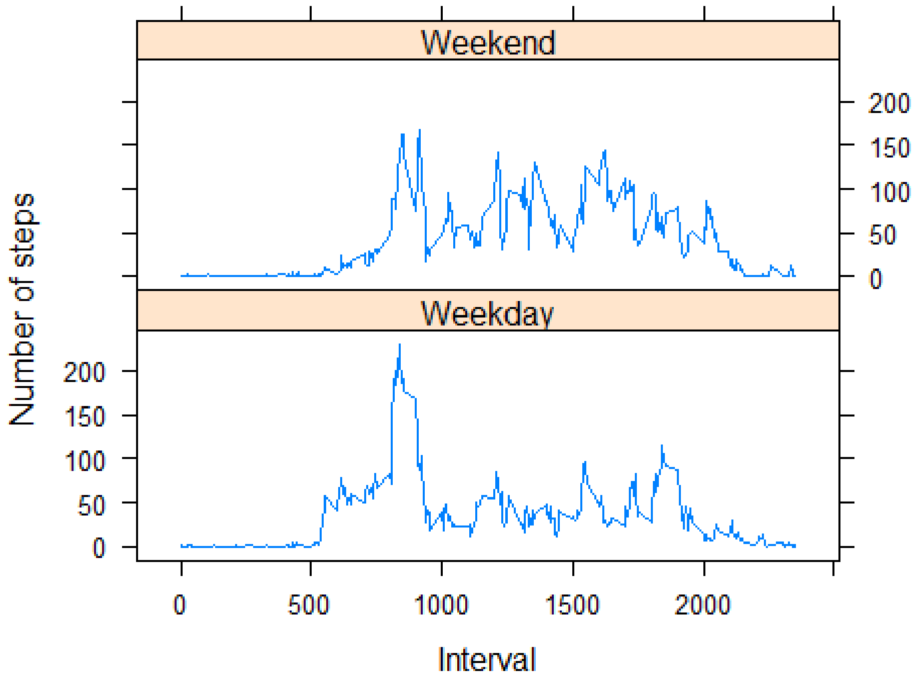
Create a new factor variable in the dataset with two levels - "weekday" and "weekend" indicating whether a given date is a weekday or weekend day.

```
# Add a new column containing the day of the week
newdata$weekday = weekdays(newdata$date)
# Add a new column containing either the Weekday OR Weekend
```

```
newdata$weekday.type <- ifelse(newdata$weekday == "Saturday" |
newdata$weekday
                                 == "Sunday", "Weekend", "Weekday")
# Convert: column to factor
newdata$weekday.type <- factor(newdata$weekday.type)
```

Make a panel plot containing a time series plot of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all weekday days or weekend days (y-axis).

```
# Make a new dataset. Grouping data by interval and weekday.type
groupedBy.Interval.WeekDay <- ddply(newdata, ~interval + weekday.type,
        summarise, mean = mean(steps))

# An intelligible plot 'unfactors' interval. Convert to characters
# first. Otherwise, the level values (1,2,3..) are produced.
groupedBy.Interval.WeekDay$interval <- as.numeric(as.character
        (groupedBy.Interval.WeekDay$interval))
library(lattice)
xyplot(mean ~ interval | weekday.type, groupedBy.Interval.WeekDay, type =
"l",
         layout = c(1, 2), xlab = "Interval", ylab = "Number of steps")
```


Weekend
Weekday
Number of steps
Interval