# Reproducible Research - Week 1 Project

*John Koegel*

## Assignment

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

### 1. Code for reading in the dataset and/or processing the data

```
# Check Installed Packages, Install Necessary Packages, and Load Packages
list.of.packages <- c("dplyr",'ggplot2','gridExtra','lubridate')
new.packages <- list.of.packages[!(list.of.packages %in% installed.packages()[,"Package"])]
if(length(new.packages)) install.packages(new.packages)

lapply(list.of.packages,library, character.only = TRUE)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine

##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

```
## [[1]]
## [1] "dplyr"     "stats"     "graphics"  "grDevices" "utils"     "datasets"
## [7] "methods"   "base"
##
## [[2]]
## [1] "ggplot2"   "dplyr"     "stats"     "graphics"  "grDevices" "utils"
## [7] "datasets"  "methods"   "base"
##
## [[3]]
##  [1] "gridExtra" "ggplot2"   "dplyr"     "stats"     "graphics"
##  [6] "grDevices" "utils"     "datasets"  "methods"   "base"
##
## [[4]]
##  [1] "lubridate" "gridExtra" "ggplot2"   "dplyr"     "stats"
##  [6] "graphics"  "grDevices" "utils"     "datasets"  "methods"
## [11] "base"
```

```
rm('list.of.packages','new.packages')

# Set Working Directory
setwd("~/Google Drive/Development/R/Coursera/Hopkins Data Science Specialization/Reproducible Research/W
dataDir <- paste(getwd(),"/data/",sep = "")
# Download Data
if (!file.exists(".data")) {
        dir.create(dataDir)
}
```

```
## Warning in dir.create(dataDir): '/Users/johnkoegel/Google Drive/
## Development/R/Coursera/Hopkins Data Science Specialization/Reproducible
## Research/Week 1 Project/data' already exists
```

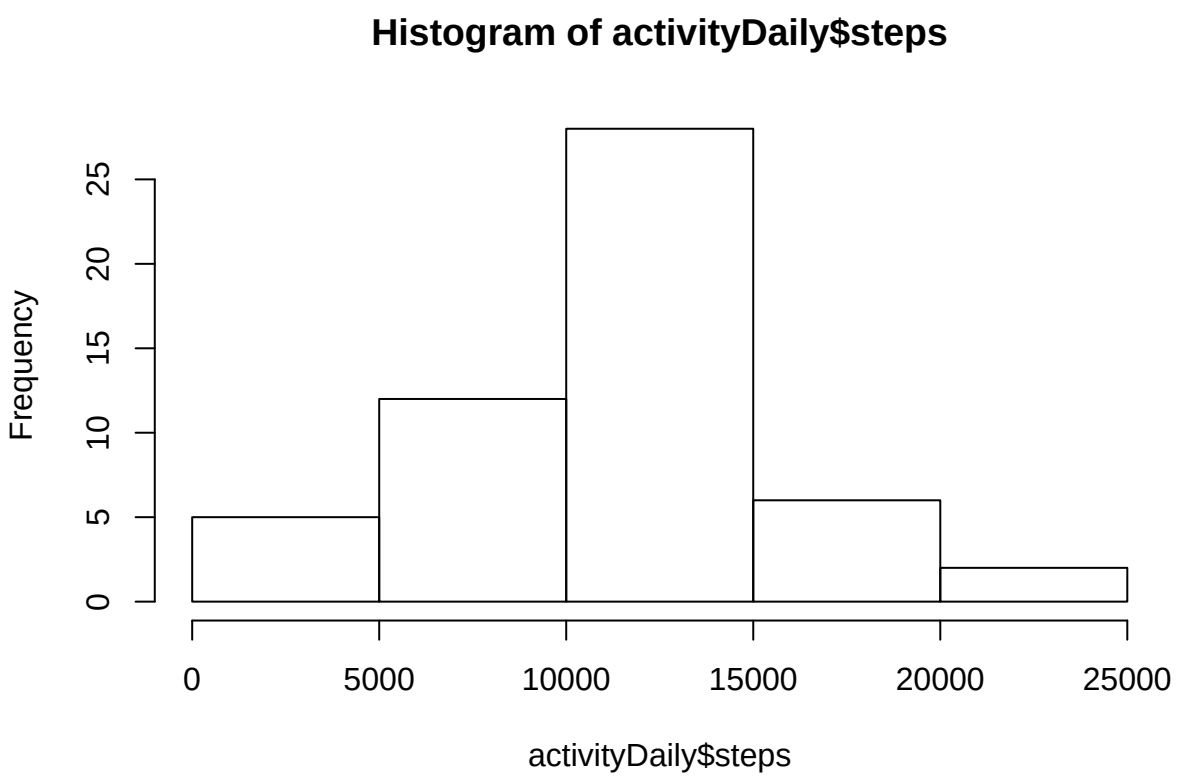
```
rawDataURL <- "https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip"
if (!file.exists(paste(dataDir,"/data.zip",sep=""))) {
        download.file(rawDataURL, destfile = paste(dataDir,"/data.zip",sep=""), method= "auto")
        unzip(zipfile=paste(dataDir,"/data.zip",sep=""),exdir = "data")
}

# Load activity data into table
activity <- read.csv(paste(dataDir,"activity.csv", sep=""), header = TRUE)
activityDfRAW <- tbl_df(activity)
rm(activity)
activityDf <- filter(activityDfRAW, !is.na(steps))

activityDaily <- group_by(activityDf, date) %>% summarise(steps = sum(steps))
```

## 2. Histogram of the total number of steps taken each day

```
hist(activityDaily$steps)
```



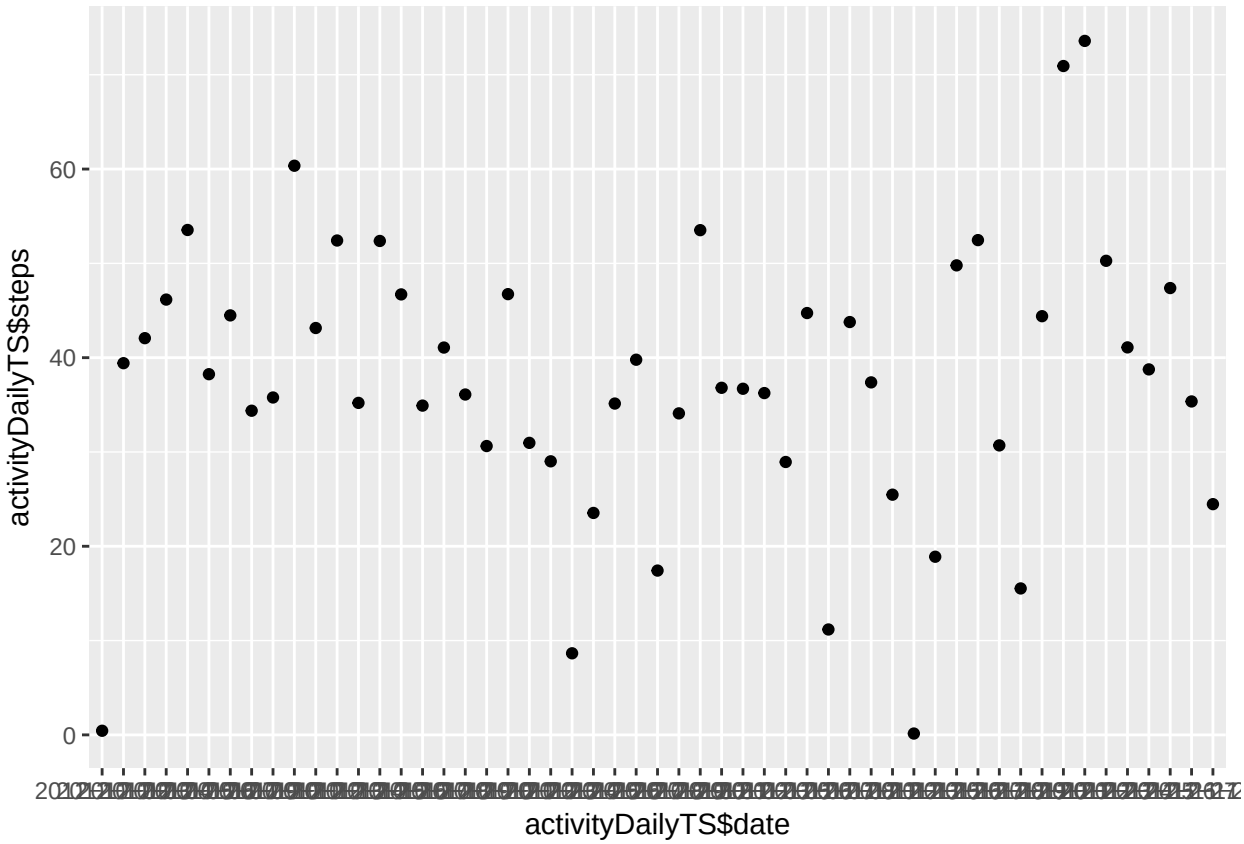

### 3. Mean and median number of steps taken each day

```
activityDailySummary <- summarise(activityDaily,mean(steps), median(steps))
activityDailySummary
```

```
## # A tibble: 1 × 2
##   `mean(steps)` `median(steps)`
##           <dbl>           <int>
## 1      10766.19           10765
```

### 4. Time series plot of the average number of steps taken

```
activityDailyTS <- group_by(activityDf, date) %>% summarise(steps = mean(steps))
ggplot(activityDailyTS) + geom_point(aes(activityDailyTS$date,activityDailyTS$steps))
```




## 5. The 5-minute interval that, on average, contains the maximum number of steps

```
activity5Min <- group_by(activityDf, interval) %>% summarise(steps = mean(steps))
filter(activity5Min, steps == max(activity5Min$steps))
```

```
## # A tibble: 1 × 2
##   interval    steps
##      <int>    <dbl>
## 1      835 206.1698
```

## 6. Code to describe and show a strategy for imputing missing data

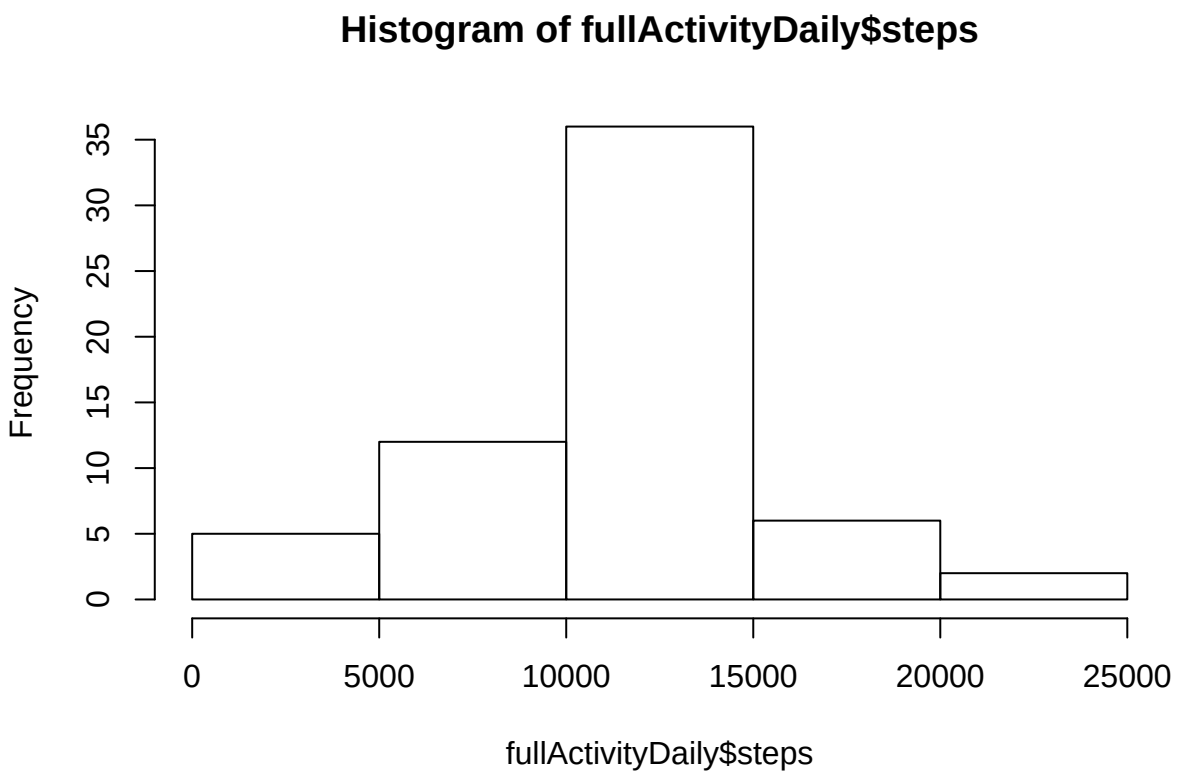
Missing values should be filled in with the average values for that timeframe.

```
activityDfNA <- filter(activityDfRAW, is.na(steps))
activityDfNA <- full_join(activityDfNA,activity5Min, by = c("interval" = "interval")) %>% select(-steps
```

## 7. Histogram of the total number of steps taken each day after missing values are imputed

Imputed values merged back with clean data set, grouped, and then a histogram is created

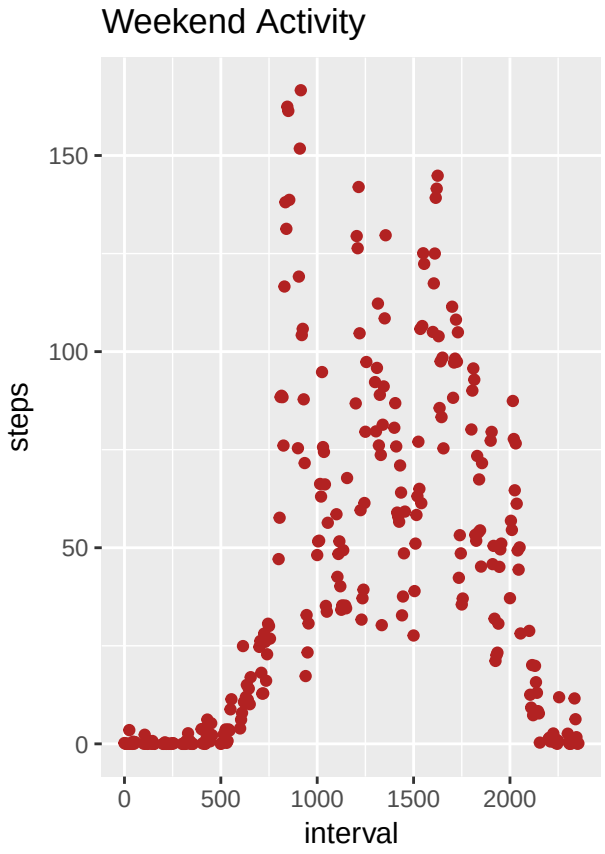
```
full <- rbind(activityDfNA, activityDf)
fullActivityDaily <- group_by(full, date) %>% summarise(steps = sum(steps))
hist(fullActivityDaily$steps)
```

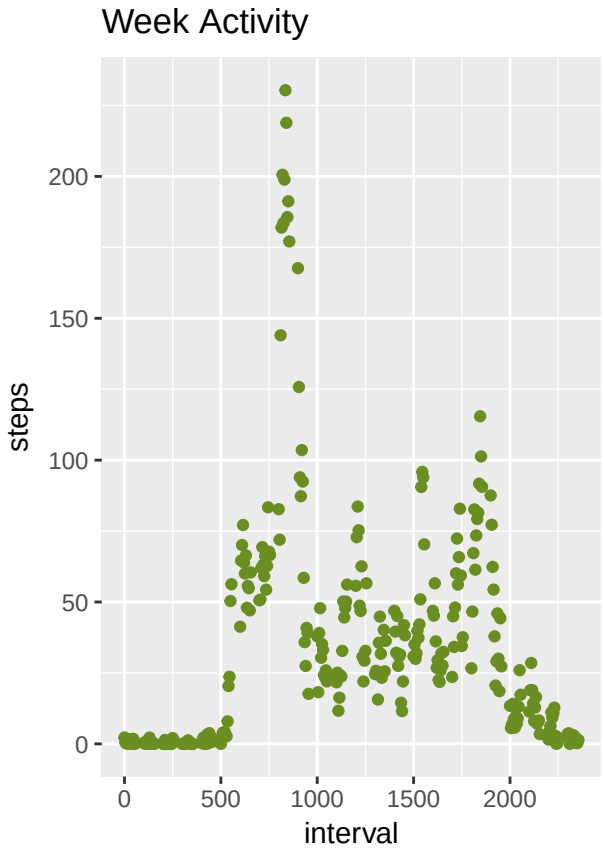



### 8. Panel plot comparing the average number of steps taken per 5-minute interval across weekdays and weekends

```
# Determine Weekdays
fullActivity5Min <- mutate(full, wDay = wday(date))
weekendActivity5Min <- fullActivity5Min %>% filter(wDay ==1 | wDay == 7) %>% select(-wDay) %>% group_by
wdayActivity5Min <- fullActivity5Min %>% filter(wDay != 1 & wDay != 7) %>% select(-wDay) %>% group_by(in

myplot1<-ggplot(weekendActivity5Min, aes(interval, steps))+geom_point(color="firebrick") + ggtitle('Week
myplot2<-ggplot(wdayActivity5Min, aes(interval, steps))+geom_point(color="olivedrab") + ggtitle('Week A
grid.arrange(myplot1, myplot2, ncol=2)
```


Weekend Activity
steps
interval
0
50
100
150
0
500
1000
1500
2000


Week Activity
steps
interval
0
50
100
150
200
0
500
1000
1500
2000