Reproducible Research Project 1

Terry Jones

10/19/2018

This assignment makes use of data from a personal activity monitoring device. This device collects data at 5 minute intervals through out the day. The data consists of two months of data from an anonymous individual collected during the months of October and November, 2012 and include the number of steps taken in 5 minute intervals each day.

Overview of Data

The data for this assignment can be downloaded from the course web site:

Dataset: (https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip)

The variables included in this dataset are:

- steps: Number of steps taking in a 5-minute interval (missing values are coded as NA)
- date: The date on which the measurement was taken in YYYY-MM-DD format
- interval: Identifier for the 5-minute interval in which measurement was taken

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

1. Load and Preprocess the data

```
## Download project file to working directory -
## "https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip"
## Load packages to support analysis

library(ggplot2)
library(base)
library(chron)
library(dplyr)

##
## 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

## set working directory & load file
setwd("C:/Users/tljon/datasciencecoursera")
activity <- read.csv("activity.csv", header=TRUE)
```

2. Process/transform the data(if necessary) into a format suitable for analysis

```
head(activity)
```

```
##   steps       date interval
## 1    NA 2012-10-01        0
## 2    NA 2012-10-01        5
## 3    NA 2012-10-01       10
## 4    NA 2012-10-01       15
## 5    NA 2012-10-01       20
## 6    NA 2012-10-01       25
```

What is mean total number of steps taken per day?

For this part of the assignment, you can ignore the missing values in the dataset. ####1. Calculate the total number of steps taken per day

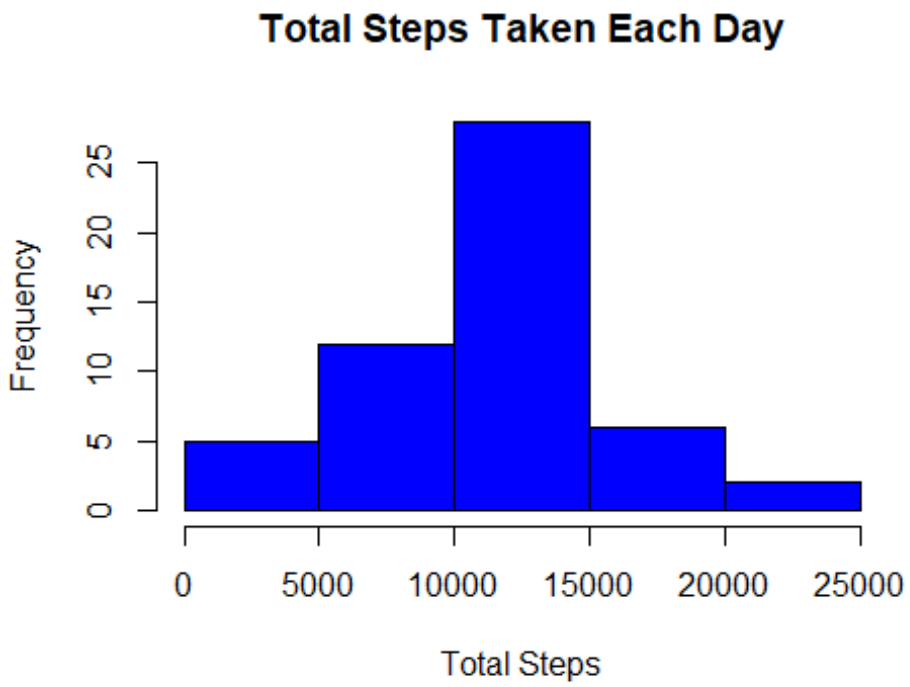
```
##Computing the number of steps
actSteps <-aggregate(steps ~ date, activity, FUN=sum)

##View the head data for number of steps
head(actSteps)
```

```
##         date steps
## 1 2012-10-02   126
## 2 2012-10-03 11352
## 3 2012-10-04 12116
## 4 2012-10-05 13294
## 5 2012-10-06 15420
## 6 2012-10-07 11015
```

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

2. If you do not understand the difference between a histogram and a barplot, research the difference between them. Make a histogram of the total number of steps taken each day

```
##plot a histogram using base plotting
hist(actSteps$steps, col="blue",
                     xlab = "Total Steps",
                     ylab = "Frequency",
                     main = "Total Steps Taken Each Day")
```



3. Mean and median number of steps taken each day

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

```
## Calculate the mean
actMean <- mean(actSteps$steps)

##Mean for the number of steps taken per day
actMean

## [1] 10766.19

## Calculate the median
actMedian <- median(actSteps$steps)

##Median for the number of steps taken per day
actMedian

## [1] 10765
```

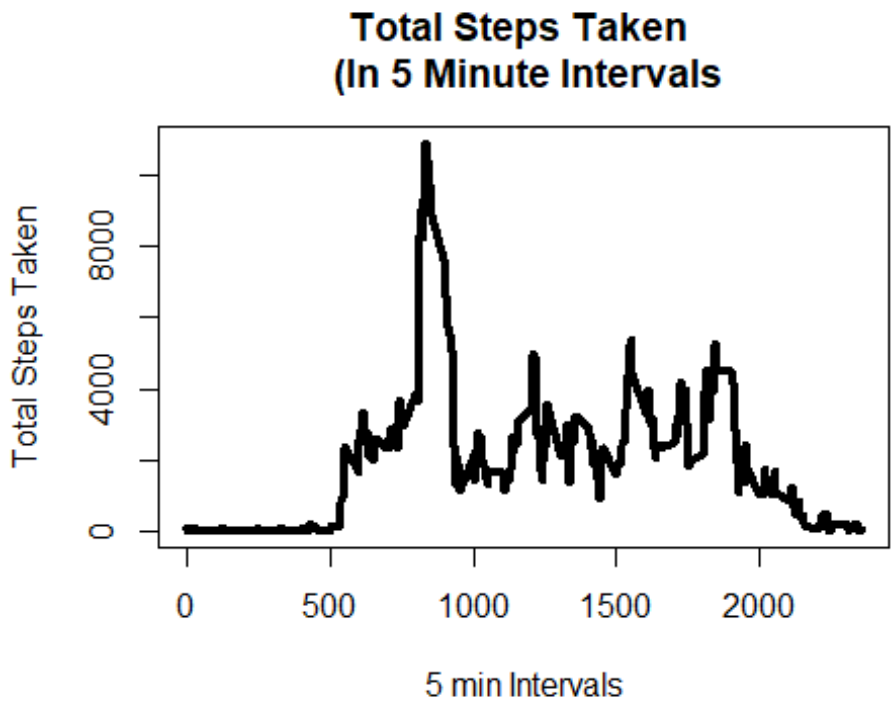
The MEAN is 10766.19 and the MEDIAN is 10765

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

What is the average daily activity pattern?

1. Make a times series plot (i.e., type="l") of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all days (y-axis)

```
##Aggregating the number of steps over a 5-minute time interval
intAggr <- aggregate(steps ~ interval, activity, FUN=sum)

##Applying base plotting to build a line graph
plot(intAggr$interval, intAggr$steps,
                        type = "l",
                        lwd = 4, xlab="5 min Intervals",
                        ylab="Total Steps Taken",
                        main="Total Steps Taken \n (In 5 Minute Intervals")
```



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

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

```
##Identify the 5-minute interval that contains the maximum number of steps
filter(intAggr, steps==max(steps))
```

```
##   interval steps
## 1      835 10927
```

The 5-minute interval that contains the maximum number of steps is 835, with a maximum number of steps equal to 10927

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

Imputing missing values

Note there are a number of days/intervals where there are missing values (coded as NA). The presence of missing days may introduce bias into some calculations or summaries of the data.

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

```
## Identify the total number of "NA" values in the dataset
sum(is.na(activity$steps))

## [1] 2304
```

There are 2304 rows with NA values.

2.Devise a strategy for filling in all of the missing values in the dataset. The strategy does not need to be sophisticated. For example, you could use the mean/median for that day, or the mean for that 5-minute interval.

```
##Compute an aggregated mean - new variable (activity mean interval "ami")
ami <- aggregate(steps ~ interval, activity, FUN=mean)

##Merge
amiMrg <- merge(x=activity, y=ami, by="interval")

##Replace NA values with the overall mean
amiMrg$steps <- ifelse(is.na(amiMrg$steps.x), amiMrg$steps.y, amiMrg$steps.x)

##View new merged table
head(amiMrg)

##   interval steps.x       date  steps.y    steps
## 1        0      NA 2012-10-01 1.716981 1.716981
## 2        0       0 2012-11-23 1.716981 0.000000
## 3        0       0 2012-10-28 1.716981 0.000000
## 4        0       0 2012-11-06 1.716981 0.000000
## 5        0       0 2012-11-24 1.716981 0.000000
## 6        0       0 2012-11-15 1.716981 0.000000
```

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

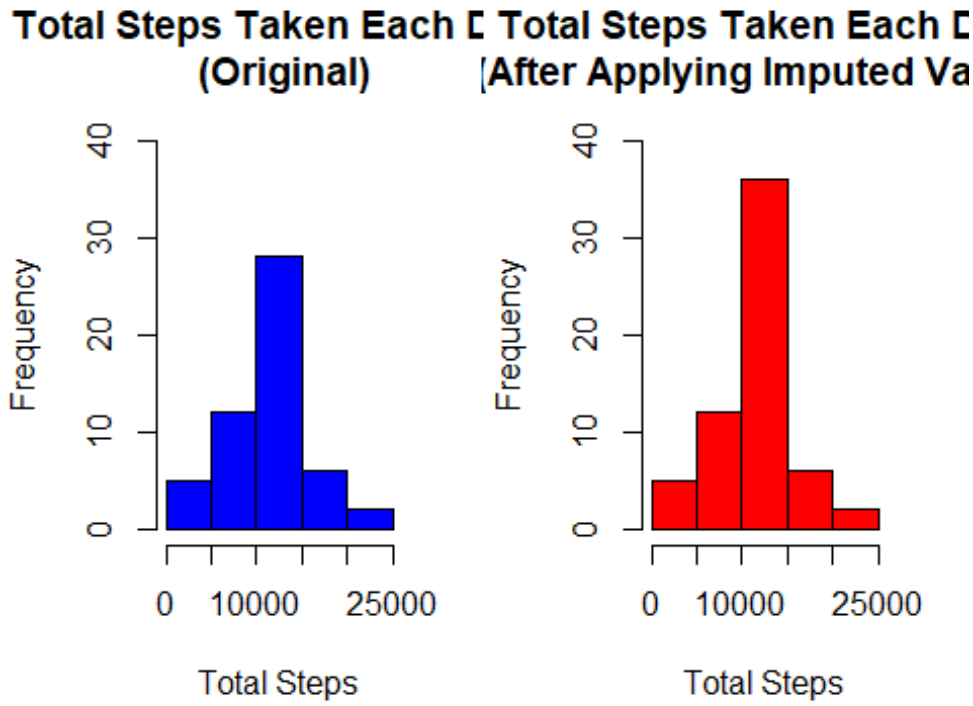
```
##Generate a new dataset conssting of the steps, date and intervals (same pattern as
original file) to create a new dataset
amiMrg <- select(amiMrg, steps, date, interval)
```

```
## View new dataset
head(amiMrg)

##      steps       date interval
## 1 1.716981 2012-10-01        0
## 2 0.000000 2012-11-23        0
## 3 0.000000 2012-10-28        0
## 4 0.000000 2012-11-06        0
## 5 0.000000 2012-11-24        0
## 6 0.000000 2012-11-15        0
```

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

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

```
## Once again use the aggregation function, this time to prepare the data set for the
histogram.  This will consiste of two histograms, one for the original dataset and
another for the new imputed dataset

amiMrgsteps <- aggregate(steps ~ date, amiMrg, FUN=sum)

##Code for the panel that will consist of 1 row and two columns (two histograms side
by side)
par(mfrow=c(1,2))

##The histogram for the original dataset

hist(actSteps$steps, col="blue",
                     xlab = "Total Steps",
                     ylab = "Frequency",
                     ylim = c(0,40),
                     cex = 1.0,
                     main = "Total Steps Taken Each Day \n (Original)")

#The new histogram base on the new dataset
hist(amiMrgsteps$steps, col="red",
                     xlab = "Total Steps",
                     ylab = "Frequency",
                     ylim = c(0,40),
                     cex = 1.0,
                     main = "Total Steps Taken Each Day \n (After Applying Imputed
Values)")
```




```
##New mean
act_mean <- mean(amiMrgsteps$steps)

##Compare the difference between the old and new means
paste("New Mean     :", round(act_mean, 2), ",",
      "Old Mean     :", round(actMean, 2), ",",
      "Variation    :", round(act_mean - actMean, 2))

## [1] "New Mean     : 10766.19 , Old Mean     : 10766.19 , Variation    : 0"

##New median
act_median <- median(amiMrgsteps$steps)

##Compare the difference between the old and new means
paste("New Median   :", round(act_median, 2), ",",
      "Old Median   :", round(actMedian, 2), ",",
      "Variation    :", round(act_median - actMedian, 2))

## [1] "New Median   : 10766.19 , Old Median   : 10765 , Variation    : 1.19"
```

There is a difference of 1.19 betwen the old and new MEDIA. However, the MEANs are both the same, no change in value, and both the old and new MEANS are equal in value.

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

Are there differences in activity patterns between weekdays and weekends?

For this part the weekdays() function may be of some help here. Use the

dataset with the filled-in missing values for this part.

1. 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.

```
## The"chron" package was installed during Step 1 at the top of the file "1. Code for reading in the dataset and/or processing the data"  This file will support converting Saturnday and Sunday to "weekends"

##This code breaks down and classifies each day as either a weekend day or weekdayy.

amiMrg$dayofwk <- ifelse(is.weekend(amiMrg$date),"Weekend", " Weekday")
table(amiMrg$dayofwk)

##
##  Weekday  Weekend
##    12960     4608

##Check the file
head(amiMrg)

##      steps       date interval  dayofwk
## 1 1.716981 2012-10-01        0  Weekday
## 2 0.000000 2012-11-23        0  Weekday
## 3 0.000000 2012-10-28        0  Weekend
## 4 0.000000 2012-11-06        0  Weekday
## 5 0.000000 2012-11-24        0  Weekend
## 6 0.000000 2012-11-15        0  Weekday
```

2. Make a panel plot containing a time series plot (i.e.type=“l”) of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all weekday days or weekend days (y-axis). See the README file in the GitHub repository to see an example of what this plot should look like using simulated data.

```
##Begin preparing for a panel plot containing a Time Series Plot
newamiMrg <- aggregate(steps ~ interval + dayofwk, amiMrg, FUN=mean)

##Check the data
head(newamiMrg)

##   interval  dayofwk      steps
## 1        0  Weekday 2.25115304
## 2        5  Weekday 0.44528302
```

```
## 3       10  Weekday 0.17316562
## 4       15  Weekday 0.19790356
## 5       20  Weekday 0.09895178
## 6       25  Weekday 1.59035639

##Build the Time Series plot using ggplot2

ggplot(newamiMrg, aes(x=interval, y=steps)) + geom_line(color="green", size=2) +
                  facet_wrap(~dayofwk, nrow=2) + labs(x="\n Interval", y="\n Total
Steps")
```