Reproducible Research: Assignment: Course Project 1

Abhay Prasad

February 20, 2016

Loading necessary R packges and readying the R Studio environment

The code chunk below is to clear all pre-existing variables and to set or load packages that will be required. These packages have already been installed [using "intall.packages()"].

```
rm(list=ls(all=TRUE)) # To remove all variables from the environment at the onset. Helps in faster proc
ls()
```

```
## character(0)
```

```
options(rpubs.upload.method = "internal")
echo = TRUE  # To make the code chunks visible in the Knitted output
options(scipen = 1)  # To turn off scientific notations for numbers
library(lattice)
library(R.utils) # Primarily needed to process the Bunzip2 raw data file
```

```
## Warning: package 'R.utils' was built under R version 3.1.1

## Loading required package: R.oo

## Warning: package 'R.oo' was built under R version 3.1.1

## Loading required package: R.methodsS3

## Warning: package 'R.methodsS3' was built under R version 3.1.1

## R.methodsS3 v1.6.1 (2014-01-04) successfully loaded. See ?R.methodsS3 for help.

## R.oo v1.18.0 (2014-02-22) successfully loaded. See ?R.oo for help.

##
## Attaching package: 'R.oo'

## The following objects are masked from 'package:methods':
##
##     getClasses, getMethods

## The following objects are masked from 'package:base':
##
##     attach, detach, gc, load, save

## R.utils v1.32.4 (2014-05-14) successfully loaded. See ?R.utils for help.
```

```
##
## Attaching package: 'R.utils'

## The following object is masked from 'package:utils':
##
##     timestamp

## The following objects are masked from 'package:base':
##
##     cat, commandArgs, getOption, inherits, isOpen, parse, warnings
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.1.1
```

```
library(stringr)
library(plyr)
require(gridExtra)
```

```
## Loading required package: gridExtra

## Warning: package 'gridExtra' was built under R version 3.1.1

## Loading required package: grid
```

```
setwd("~/RepResPA1Feb16") # Setting working directory for this assignment
```

Loading and processing the data

The data for this assignment can be downloaded from the course web site URL: https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip

The raw data file was downloaded using the following code. This code is shown as text rather than as an executable code chunk to avoid the time-consuming steps of downloading and unzipping the file running repeatedly:

fileUrl <- "http://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip"

download.file(fileUrl, destfile="~/RepResPA1ActivityData.zip")

unzip("RepResPA1ActivityData.zip")

Data Loading

```
rawdata <- read.csv("~/RepResPA1Feb16/activity.csv", colClasses = c("integer", "Date", "factor"))
dim(rawdata) #This should be 17568 rows and 3 columns
```

```
## [1] 17568     3
```

```
rawdata$month <- as.numeric(format(rawdata$date, "%m"))
```

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.

```
dataexNA <- na.omit(rawdata) #Omits rows with missing values in the dataset
rownames(dataexNA) <- 1:nrow(dataexNA)
head(dataexNA)
```

```
##   steps       date interval month
## 1     0 2012-10-02        0    10
## 2     0 2012-10-02        5    10
## 3     0 2012-10-02       10    10
## 4     0 2012-10-02       15    10
## 5     0 2012-10-02       20    10
## 6     0 2012-10-02       25    10
```

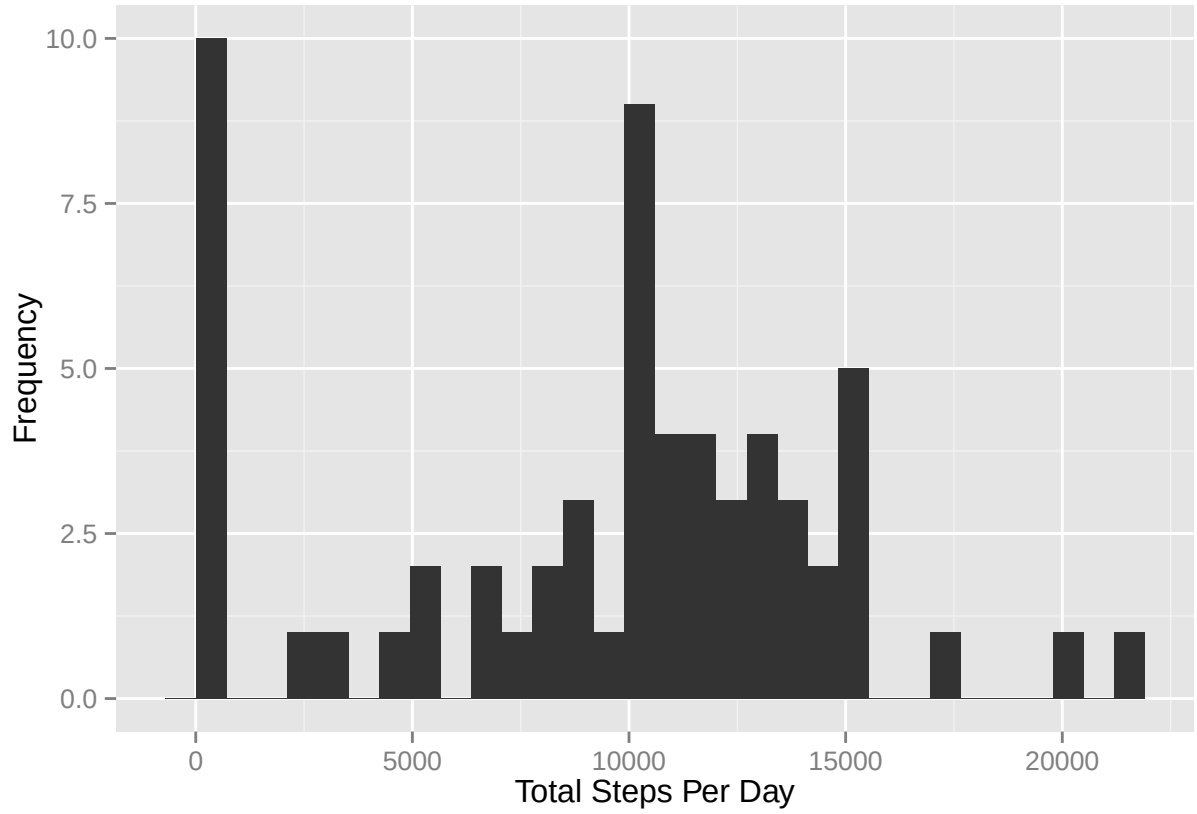
```
dim(dataexNA)
```

```
## [1] 15264     4
```

Histogram of the total number of steps taken each day

```
numSteps <- tapply(rawdata$steps, rawdata$date, sum, na.rm=TRUE)
qplot(numSteps, xlab='Total Steps Per Day', ylab='Frequency')
```

```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```



Mean and median number of steps taken each day

```
totalSteps <- aggregate(dataexNA$steps, list(Date = dataexNA$date), FUN = "sum")$x
mean(totalSteps)
```

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

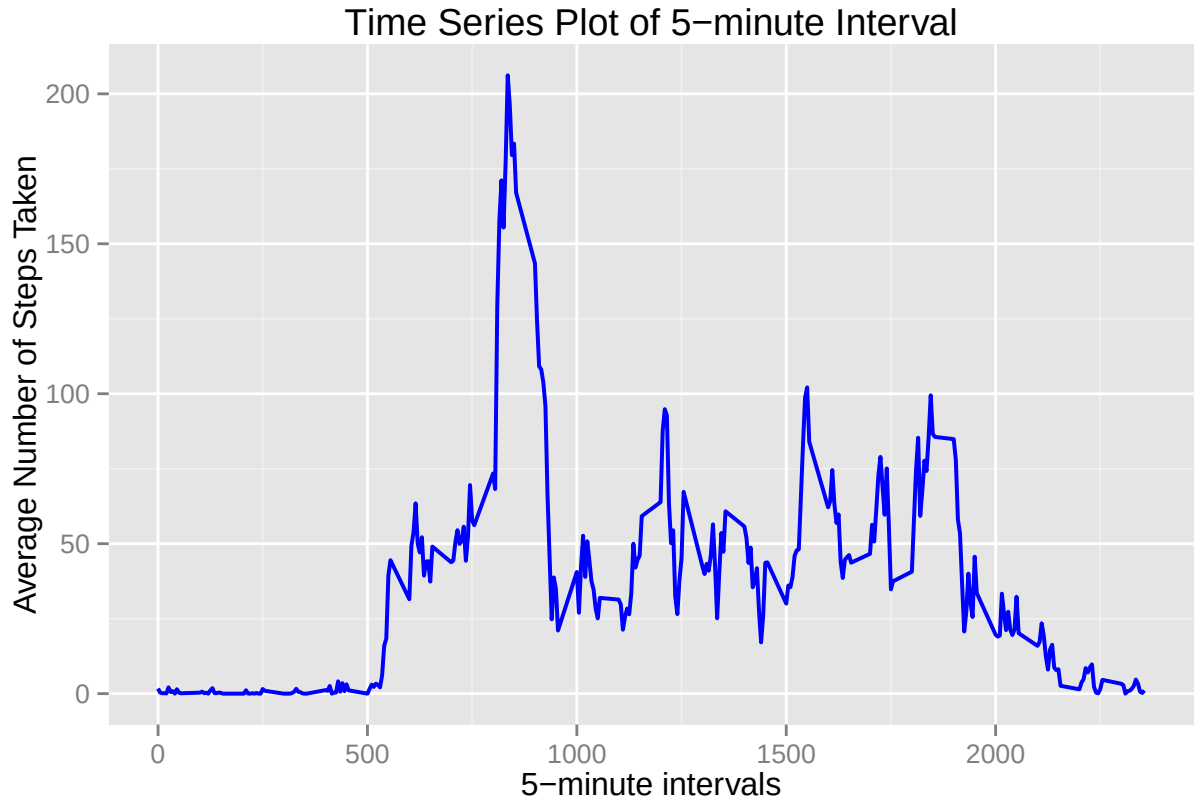
```
median(totalSteps)
```

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

What is the average daily activity pattern?

Time series plot of the average number of steps taken

```
avgSteps <- aggregate(dataexNA$steps, list(interval = as.numeric(as.character(dataexNA$interval))), FUN
names(avgSteps)[2] <- "meanOfSteps"
ggplot(avgSteps, aes(interval, meanOfSteps)) + geom_line(color = "blue", size = 0.7) + labs(title = "Ti
```



Imputing missing values

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

```
avgSteps[avgSteps$meanOfSteps == max(avgSteps$meanOfSteps), ]
```

```
##     interval meanOfSteps
## 104      835    206.1698
```

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

```
sum(is.na(rawdata)) # To calculate and report the total number of missing values in the original datase
```

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

```
dim(rawdata) - dim(dataexNA)  # Another way to verify the  calculated total number of missing values in
```

```
## [1] 2304    0
```

Devise a strategy for filling in all of the missing values in the dataset and Create a new dataset that is equal to the original dataset but with the missing data filled in.

```
newData <- rawdata
for (i in 1:nrow(newData)) {
    if (is.na(newData$steps[i])) {
        newData$steps[i] <- avgSteps[which(newData$interval[i] == avgSteps$interval), ]$meanOfSteps
    }
}
head(newData)
```

```
##       steps       date interval month
## 1 1.7169811 2012-10-01        0    10
## 2 0.3396226 2012-10-01        5    10
## 3 0.1320755 2012-10-01       10    10
## 4 0.1509434 2012-10-01       15    10
## 5 0.0754717 2012-10-01       20    10
## 6 2.0943396 2012-10-01       25    10
```

```
dim(newData)
```

```
## [1] 17568     4
```

```
sum(is.na(newData))
```

```
## [1] 0
```

```
dim(rawdata) - dim(newData)
```

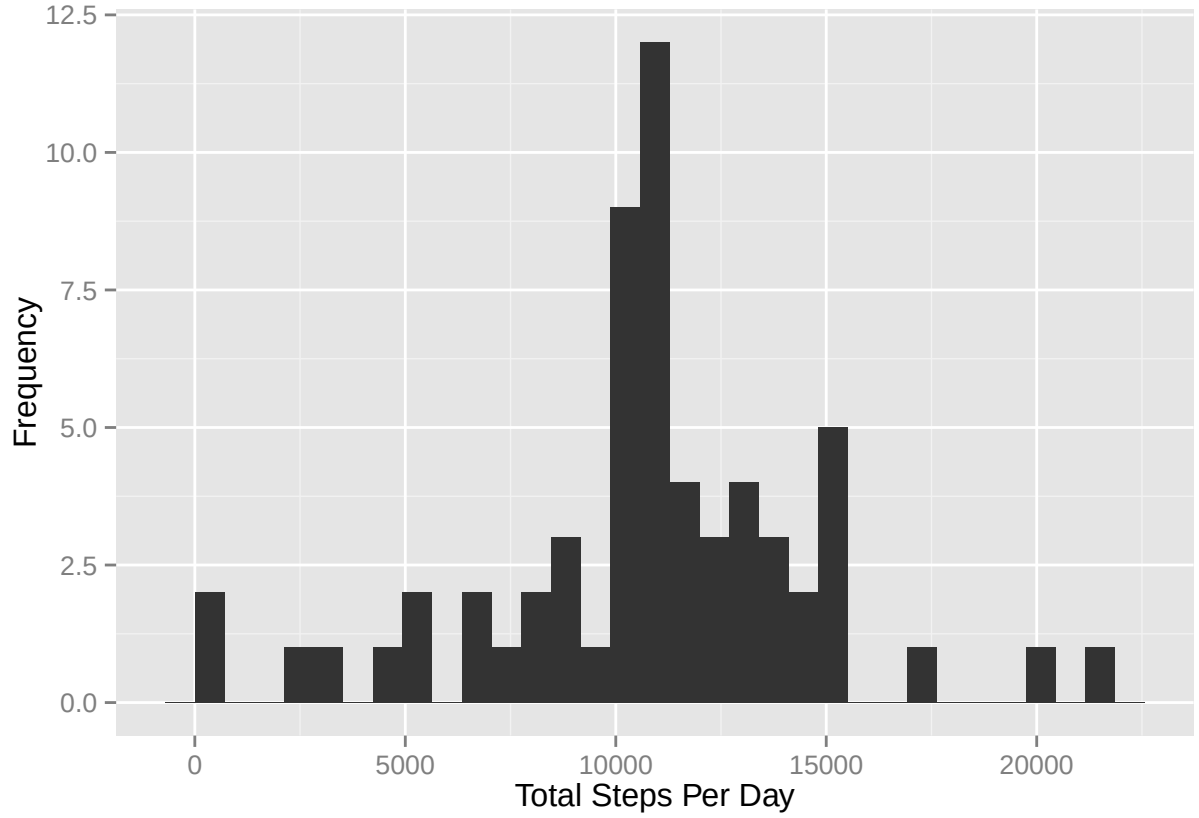
```
## [1] 0 0
```

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

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.

```
newnumSteps <- tapply(newData$steps, newData$date, sum, na.rm=TRUE)
qplot(newnumSteps, xlab='Total Steps Per Day', ylab='Frequency')
```

```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```



Do these values differ from the estimates from the first part of the assignment? What is the impact of imputing missing data on the estimates of the total daily number of steps?

```
newTotalSteps <- aggregate(newData$steps, list(Date = newData$date), FUN = "sum")$x
newMean <- mean(newTotalSteps)
newMean
```

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

```
newMedian <- median(newTotalSteps)
newMedian
```

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

```
oldMean <- mean(totalSteps)
oldMedian <- median(totalSteps)
newMean - oldMean
```

```
## [1] 0
```

```
newMedian - oldMedian
```

```
## [1] 1.188679
```

After imputing the missing data and then calculating and comparing the means it is seen that the mean of the total steps taken per day in the imputed dataset is the same as that of the mean of the total steps taken per day in the original dataset; however, the median of total steps taken per day of the imputed dataset is (slightly) greater than that of the original median.

Are there differences in activity patterns between weekdays and weekends?

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

```
newData$weekdays <- factor(format(newData$date, "%A"))
levels(newData$weekdays)
```

```
## [1] "Friday"    "Monday"    "Saturday"  "Sunday"    "Thursday"  "Tuesday"
## [7] "Wednesday"
```

```
levels(newData$weekdays) <- list(Weekday = c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday"), W
levels(newData$weekdays)
```

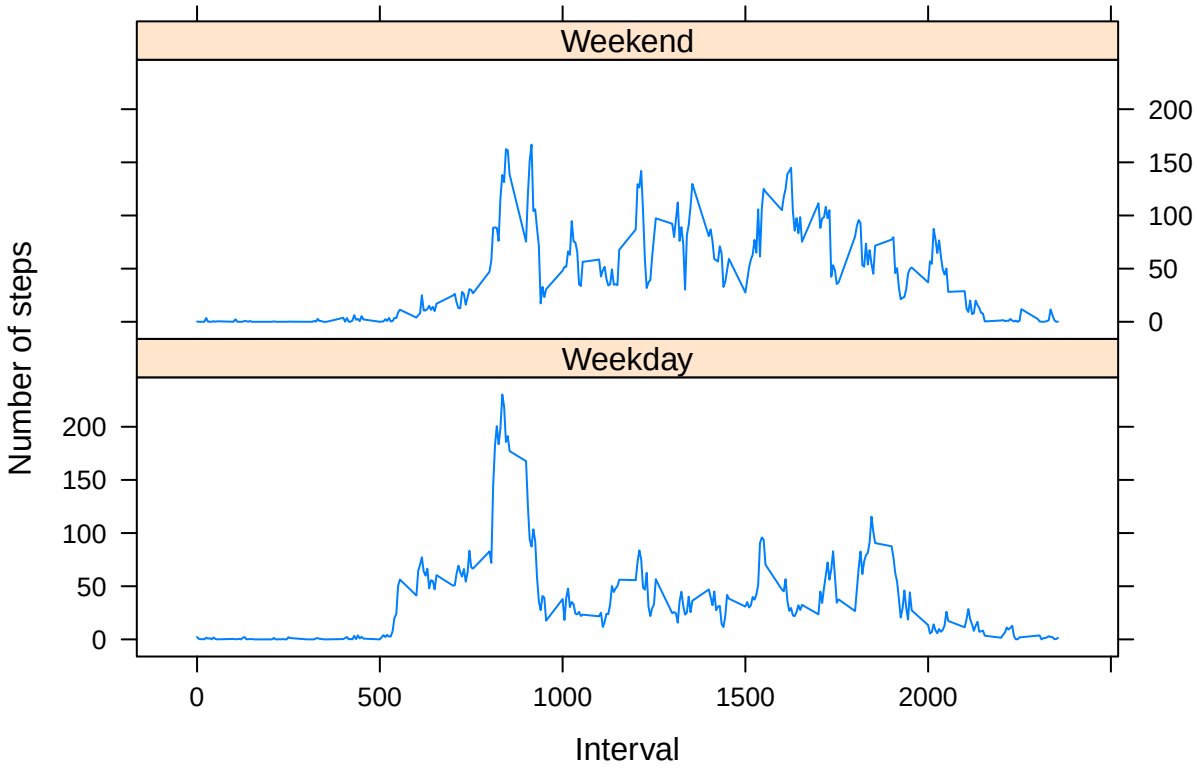
```
## [1] "Weekday" "Weekend"
```

```
table(newData$weekdays)
```

```
##
## Weekday Weekend
##   12960    4608
```

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).

```
avgSteps <- aggregate(newData$steps, list(interval = as.numeric(as.character(newData$interval)), weekda
names(avgSteps)[3] <- "meanOfSteps"
xyplot(avgSteps$meanOfSteps ~ avgSteps$interval | avgSteps$weekdays, layout = c(1, 2), type = "l", xlab
```



On the basis of the graph it seems that the patterns for weekdays and weekends are different. On weekdays, there's a higher degree of activity just after 8am, but much lesser on weekends. On weekends there seems to be more regular activity throughout the day; whereas, on weekdays the level of activity is less evenly spread out and most activity occurs in the mornings, just after noon, and in the evenings.

```
sessionInfo()
```

```
## R version 3.1.0 (2014-04-10)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] grid      stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
## [1] gridExtra_0.9.1   plyr_1.8.1        stringr_0.6.2     ggplot2_1.0.0
## [5] R.utils_1.32.4    R.oo_1.18.0       R.methodsS3_1.6.1 lattice_0.20-29
##
## loaded via a namespace (and not attached):
```

```
##  [1] colorspace_1.2-4 digest_0.6.4     evaluate_0.8     gtable_0.1.2
##  [5] htmltools_0.2.6  knitr_1.12.3     labeling_0.2     MASS_7.3-31
##  [9] munsell_0.4.2    proto_0.3-10     Rcpp_0.11.2      reshape2_1.4
## [13] rmarkdown_0.9.2  scales_0.2.4     tools_3.1.0      yaml_2.1.11
```