Reproducible Research - Course project 1

David Wagner

Date: 2/26/2020

Introduction

It is now possible to collect a large amount of data about personal movement using activity monitoring devices such as a Fitbit, Nike Fuelband, or Jawbone Up. These type of devices are part of the “quantified self” movement – a group of enthusiasts who take measurements about themselves regularly to improve their health, to find patterns in their behavior, or because they are tech geeks. But these data remain under-utilized both because the raw data are hard to obtain and there is a lack of statistical methods and software for processing and interpreting the data.

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.

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

Dataset: Activity monitoring data [52K] 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.Setup the libraries & working directories

```
knitr::opts_chunk$set(echo = TRUE)
setwd("~/GitHub/Repo Research Proj 1")
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
```

```
library(ggplot2)
options(scipen = 999, digits=2)
```

2.Download, unzip data file,read.csv into variable activ and check sample data

```
knitr::opts_chunk$set(echo = TRUE)
download.file("https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip",
"activity.zip")
unzip("activity.zip")
activ <- read.csv("activity.csv")
head(activ)
```

```
##   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
```

Project Questions

1. What is mean total number of steps taken per day?

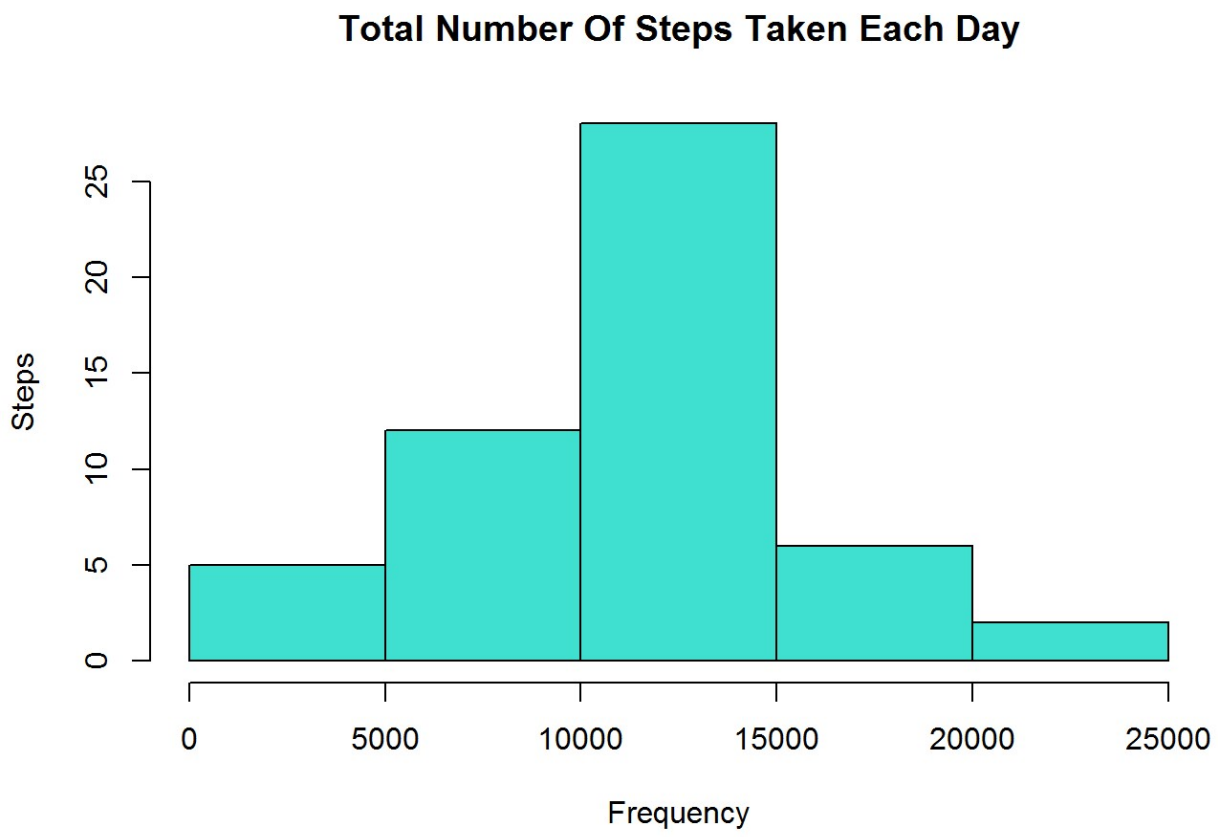
For this part of the assignment, you can ignore the missing values in the dataset.

```
#Aggregating(summation) of steps over date
aggsteps<- aggregate(steps ~ date, activ, FUN=sum)
#Aggregated Data Sample (all steps added for a particular date)
head(aggsteps)
```

```
##         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
```

Histogram of steps taken daily

```
#Plotting histogram using hist() from Base Plotting
hist(aggsteps$steps,
     col = "turquoise",
     xlab = "Frequency",
     ylab = "Steps",
     main = "Total Number Of Steps Taken Each Day")
```



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

```
activmean <- mean(aggsteps$steps)
activmedian <- median(aggsteps$steps)

## mean total steps per day
activmean
```

```
## [1] 10766
```

```
## median total steps per day
activmedian
```

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

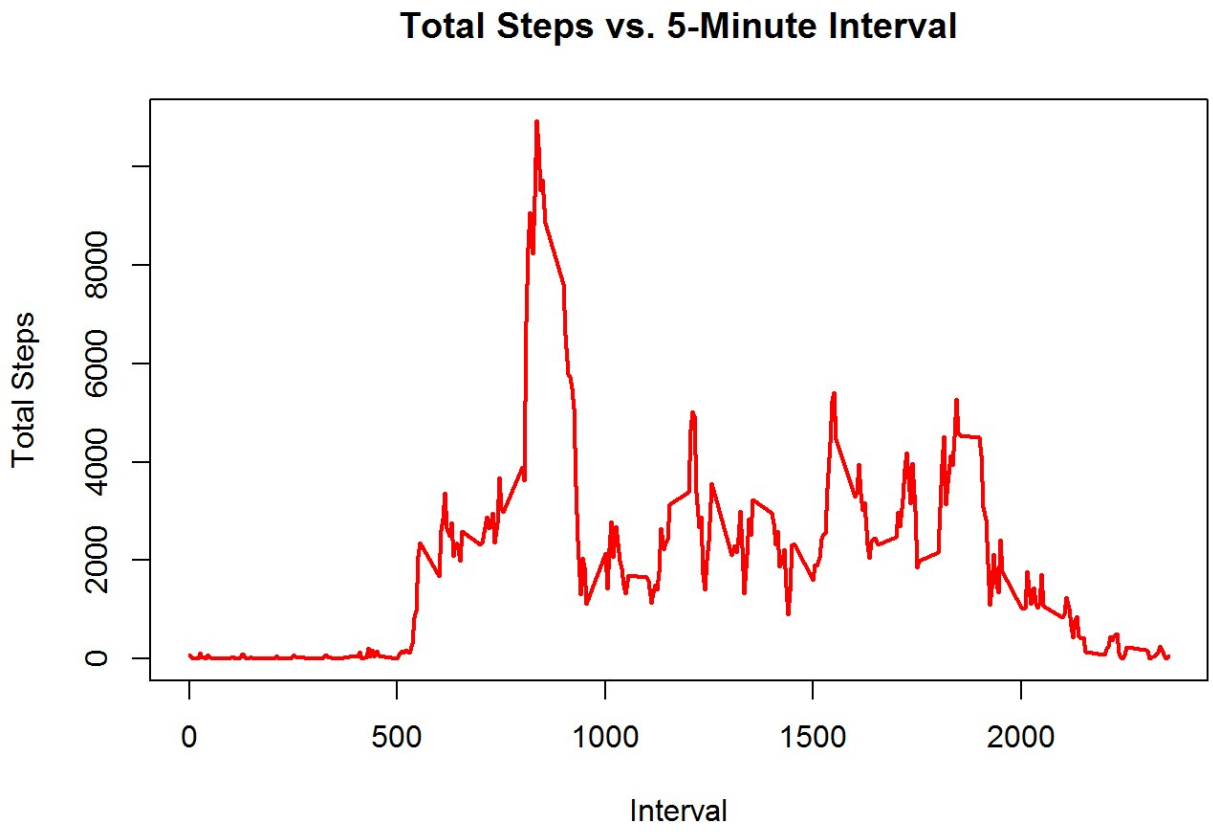
Mean step total taken is: **10766.19**.
Median step total taken is : **10765**.

What is the average daily activity pattern?

1. Make 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 days (y-axis).

```
#Aggregating(summation) of steps over time interval (of 5 min)
agginterval <- aggregate(steps ~ interval, activ, FUN = sum)

#Plotting line graph using plot() from Base Plotting for Total Steps vs 5-Minute Inter
val
plot(agginterval$interval, agginterval$steps,
     type = "l", lwd = 2,
     col = "red",
     xlab = "Interval",
     ylab = "Total Steps",
     main = "Total Steps vs. 5-Minute Interval")
```



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

locating the 5 min interval which has maximum number of steps.

```
filter(agginterval, steps==max(steps))
```

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

The maximum steps taken are **10927** which occurred in the **835th** interval.

Imputing missing values

Note that 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)

```
#In the output of the below query TRUE represents the total number of NA values
table(is.na(activ))
```

```
##
## FALSE  TRUE
## 50400  2304
```

Total number of rows with NAs is **2304**

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, etc.

```
# Aggregating mean steps over 5-minute interval in the original data set
meaninterval<- aggregate(steps ~ interval, activ, FUN=mean)

# Merging the mean of total steps for a date with the original data set
activnew <- merge(x=activ, y=meaninterval, by="interval")

# Replacing the NA values with the mean for that 5-minute interval
activnew$steps <- ifelse(is.na(activnew$steps.x), activnew$steps.y, activnew$steps.x)

# Merged dataset which will be subsetted in the next step by removing not required col
umns
head(activnew)
```

```
##   interval steps.x       date steps.y steps
## 1        0      NA 2012-10-01     1.7   1.7
## 2        0       0 2012-11-23     1.7   0.0
## 3        0       0 2012-10-28     1.7   0.0
## 4        0       0 2012-11-06     1.7   0.0
## 5        0       0 2012-11-24     1.7   0.0
## 6        0       0 2012-11-15     1.7   0.0
```

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

```
# Retrieving only the required columns (steps, date, interval) and storing in the new
data set.
activnew <- select(activnew, steps, date, interval)

#New dataset with NA imputed by mean for that 5-minute interval
head(activnew)
```

```
##   steps       date interval
## 1   1.7 2012-10-01        0
## 2   0.0 2012-11-23        0
## 3   0.0 2012-10-28        0
## 4   0.0 2012-11-06        0
## 5   0.0 2012-11-24        0
## 6   0.0 2012-11-15        0
```

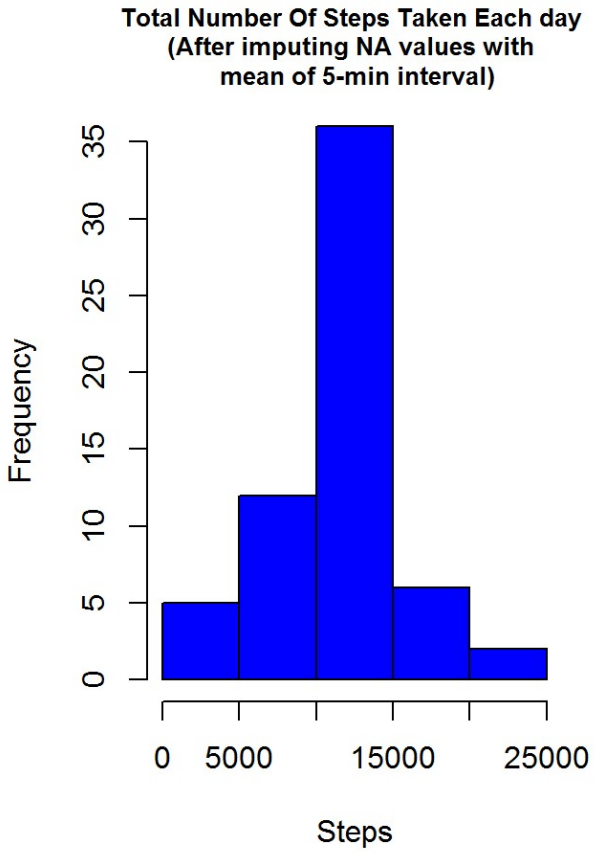
4. Make a histogram of the total number of steps taken each day and Calculate and report the mean and median total number of steps taken per day. Do these values differ from the estimates from the first part of the assignment? What is the impact of imputing missing data on the estimates of the total daily number of steps?

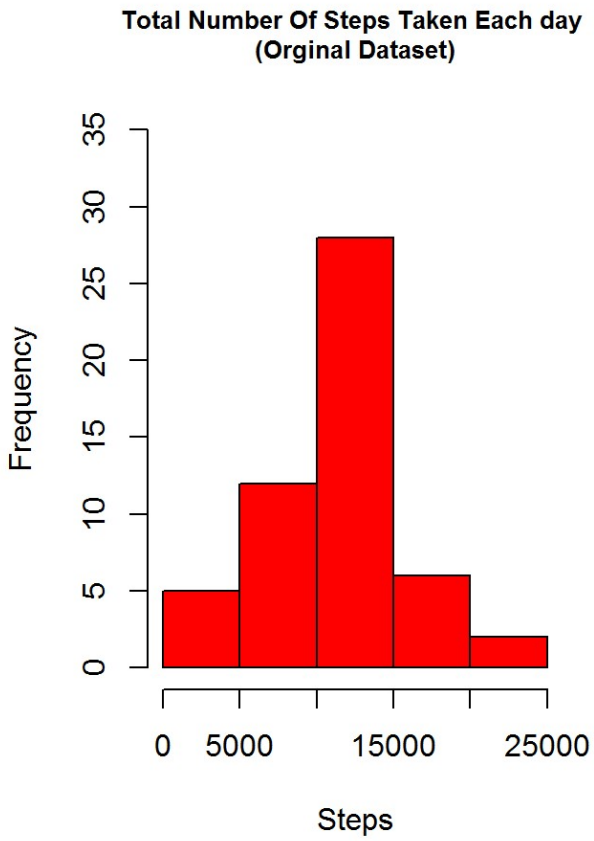
```
#Aggregating(summation) of steps over date
aggstepsnew<- aggregate(steps ~ date, activnew, FUN=sum)

#Plotting
#Set up the pannel for one row and two columns
par(mfrow=c(1,2))

#Histogram after imputing NA values with mean of 5-min interval
hist(aggstepsnew$steps,
     col="blue",
     xlab = "Steps",
     ylab = "Frequency",
     ylim = c(0,35),
     main = "Total Number Of Steps Taken Each day \n(After imputing NA values with \n
mean of 5-min interval)",
     cex.main = 0.8)

#Histogram with the orginal dataset
hist(aggsteps$steps,
     col="red",
     xlab = "Steps",
     ylab = "Frequency",
     ylim = c(0,35),
     main = "Total Number Of Steps Taken Each day \n(Orginal Dataset)",
     cex.main = 0.8)
```






```
par(mfrow=c(1,1)) #Reset the panel

activmeannew <- mean(aggstepsnew$steps)
activmediannew <- median(aggstepsnew$steps)

#Comparing Means
paste("New Mean      :", round(activmeannew,2), "," ,
      " Original Mean :", round(activmean,2),"," ,
      " Difference :",round(activmeannew,2) -  round(activmean,2))
```

```
## [1] "New Mean      : 10766.19 ,  Original Mean : 10766.19 ,  Difference : 0"
```

```
#Comparing Medians
paste("New Median    :", activmediannew, ",",
      " Original Median :", activmedian,"," ,
      " Difference :",round(activmediannew - activmedian,2))
```

```
## [1] "New Median    : 10766.1886792453 ,  Original Median : 10765 ,  Difference : 1.
19"
```

The Mean are same but New Median differs from Original Median by 1.19

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.

```
# install and load chron in case not loaded
library(chron)
```

```
## Warning: package 'chron' was built under R version 3.6.2
```

```
## NOTE: The default cutoff when expanding a 2-digit year
## to a 4-digit year will change from 30 to 69 by Aug 2020
## (as for Date and POSIXct in base R.)
```

```
#is.weekend() function considers Saturday and Sunday as weekends
#In the output of below query FALSE means weekday, TRUE means weekend
table(is.weekend(activnew$date))
```

```
##
## FALSE  TRUE
## 12960  4608
```

```
##
## FALSE  TRUE
## 12960  4608
#Adding new factor variable "dayofweek" indicating whether a given date is a weekday o
r weekend day
activnew$dayofweek <- ifelse(is.weekend(activnew$date), "weekend", "weekday")

#Number of Weekdays and Weekends
table(activnew$dayofweek)
```

```
##
## weekday weekend
##   12960    4608
```

```
##
## weekday weekend
##   12960    4608
#New Data after adding factor variable for weekday or weekend
head(activnew)
```

```
##   steps       date interval dayofweek
## 1   1.7 2012-10-01        0   weekday
## 2   0.0 2012-11-23        0   weekday
## 3   0.0 2012-10-28        0   weekend
## 4   0.0 2012-11-06        0   weekday
## 5   0.0 2012-11-24        0   weekend
## 6   0.0 2012-11-15        0   weekday
```

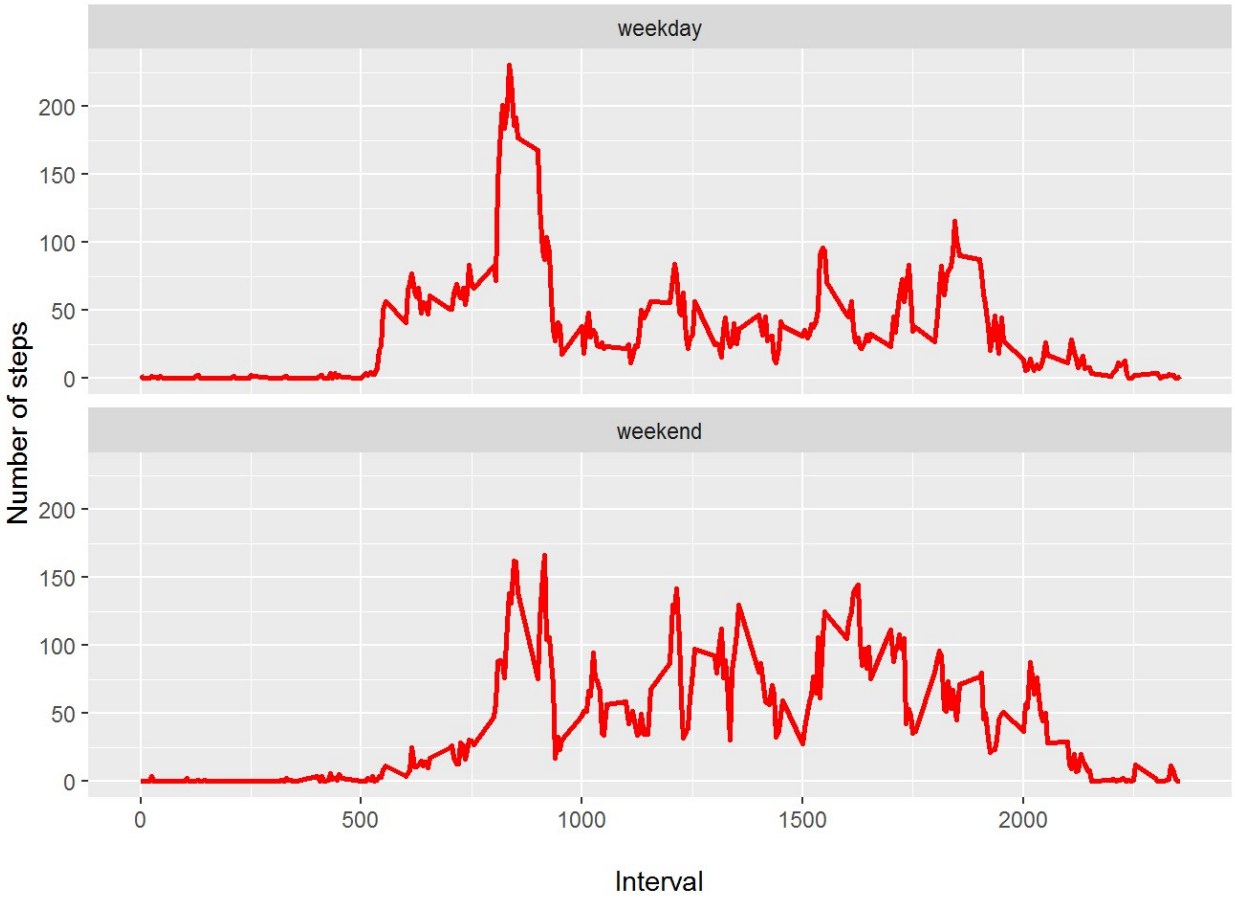
2. Make a panel plot containing a time series plot (i.e. type = “l”) of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all weekday days or weekend days (y-axis). The plot should look something like the following, which was created using simulated data:

```
#Aggregating(mean) steps over interval and day of week
meanintervalnew<- aggregate(steps ~ interval + dayofweek, activnew, FUN=mean)

#Aggregated Data
head(meanintervalnew)
```

```
##   interval dayofweek steps
## 1        0   weekday 2.251
## 2        5   weekday 0.445
## 3       10   weekday 0.173
## 4       15   weekday 0.198
## 5       20   weekday 0.099
## 6       25   weekday 1.590
```

```
#Time Series plot using ggplot
ggplot(meanintervalnew, aes(x=interval, y=steps)) +
  geom_line(color="red", size=1) +
  facet_wrap(~dayofweek, nrow=2) +
  labs(x="\nInterval", y="\nNumber of steps")
```



#library(knitr) #knit("PA1_template.Rmd")