Reproducible Research: Peer Assessment 1

Synopsis

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 underutilized 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.

Here we make 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.

What follows is a brief exploratory data visualization analysis including the code used to generate the presented output.

Load any needed packages

```r
# Package names
packages <- c("ggplot2", "dplyr")

# Install packages not yet installed
installed_packages <- packages %in% rownames(installed.packages())
if (any(installed_packages == FALSE)) {
  invisible(install.packages(packages[!installed_packages]))
}

# Packages, library loading
invisible(lapply(packages, library, character.only = TRUE,quietly = TRUE))
```

1) Code for reading and preprocessing the data

Download the Activity monitoring data from the file url. View a summary of the data.

```r
#set/save the directory where files will be saved
WD <- getwd()
if (!is.null(WD)) setwd(WD)

fileUrl <- "https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip"
download.file(fileUrl,destfile = "activity.zip",quiet = TRUE,method = "curl")
dateDownloaded <- date()
unzip("activity.zip")
```

```r
activity <- read.csv("activity.csv")
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
```

```r
summary(activity)
```

```
##      steps                date          interval
##  Min.   :  0.00   2012-10-01:  288   Min.   :   0.0
##  1st Qu.:  0.00   2012-10-02:  288   1st Qu.: 588.8
##  Median :  0.00   2012-10-03:  288   Median :1177.5
##  Mean   : 37.38   2012-10-04:  288   Mean   :1177.5
##  3rd Qu.: 12.00   2012-10-05:  288   3rd Qu.:1766.2
##  Max.   :806.00   2012-10-06:  288   Max.   :2355.0
##  NA's   :2304     (Other)   :15840
```

Format the date variable to proper class
Add a column indicating the 5 min interval within a hour
Add a column indicating the hour of the 5 min interval

```r
activity$date <- as.Date(as.character(activity$date),"%Y-%m-%d")
activity$hour <- activity$interval %/% 100 #integer division
activity$mins <- activity$interval %% 100 #returns remainder
head(activity)
```

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

```r
summary(activity)
```

```
##      steps             date               interval           hour
##  Min.   :  0.00   Min.   :2012-10-01   Min.   :   0.0   Min.   : 0.00
##  1st Qu.:  0.00   1st Qu.:2012-10-16   1st Qu.: 588.8   1st Qu.: 5.75
##  Median :  0.00   Median :2012-10-31   Median :1177.5   Median :11.50
##  Mean   : 37.38   Mean   :2012-10-31   Mean   :1177.5   Mean   :11.50
##  3rd Qu.: 12.00   3rd Qu.:2012-11-15   3rd Qu.:1766.2   3rd Qu.:17.25
##  Max.   :806.00   Max.   :2012-11-30   Max.   :2355.0   Max.   :23.00
##  NA's   :2304
##       mins
##  Min.   : 0.00
##  1st Qu.:13.75
##  Median :27.50
##  Mean   :27.50
##  3rd Qu.:41.25
##  Max.   :55.00
##
```

What is the mean/median of number of steps taken day?

Ignoring the missing values in the dataset,

We first subset the data, then group the data by days, and finally summarize the resulting groups by sum, mean, and median.

```
steps_day <- subset(activity,select=c(steps,date))
steps_day <- steps_day %>% group_by(date) %>%
  summarise_all(list(dayTotal=sum,dayMean=mean,dayMedian=median),na.rm=TRUE)
```

Show the resulting dataframe indicating the sum, mean, and median for each of the days of the dataset

```
head(steps_day,10)     # total,mean,median steps taken each day
```

```
## # A tibble: 10 x 4
##    date       dayTotal dayMean dayMedian
##    <date>        <int>   <dbl>     <dbl>
##  1 2012-10-01        0 NaN            NA
##  2 2012-10-02      126   0.438         0
##  3 2012-10-03    11352  39.4           0
##  4 2012-10-04    12116  42.1           0
##  5 2012-10-05    13294  46.2           0
##  6 2012-10-06    15420  53.5           0
##  7 2012-10-07    11015  38.2           0
##  8 2012-10-08        0 NaN            NA
##  9 2012-10-09    12811  44.5           0
## 10 2012-10-10     9900  34.4           0
```

```
summary(steps_day)
```

```
##       date               dayTotal        dayMean          dayMedian
##  Min.   :2012-10-01   Min.   :    0   Min.   : 0.1424   Min.   :0  
##  1st Qu.:2012-10-16   1st Qu.: 6778   1st Qu.:30.6979   1st Qu.:0  
##  Median :2012-10-31   Median :10395   Median :37.3785   Median :0  
##  Mean   :2012-10-31   Mean   : 9354   Mean   :37.3826   Mean   :0  
##  3rd Qu.:2012-11-15   3rd Qu.:12811   3rd Qu.:46.1597   3rd Qu.:0  
##  Max.   :2012-11-30   Max.   :21194   Max.   :73.5903   Max.   :0  
##                                       NA's   :8         NA's   :8
```

The resulting dataframe, steps_day, can be used to answer the following questions.

Q2) calculate the total number of steps taken per day.

As the summary above shows,
the total number of steps taken is 570608
the total daily steps taken is 9354.2295082
the average daily steps taken is 37.3825996
the median daily steps taken is 0

```
sum(steps_day$dayTotal,na.rm=TRUE)
```

```
## [1] 570608
```

```
mean(steps_day$dayTotal,na.rm=TRUE)
```

```
## [1] 9354.23
```

```
mean(steps_day$dayMean,na.rm=TRUE)
```

```
## [1] 37.3826
```

```
median(steps_day$dayMedian,na.rm=TRUE)
```

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

Q3) calculate the total steps taken across all days: 570608
calculate the mean of steps taken per day: 9354.2295082

```
sum(steps_day$dayTotal,na.rm=TRUE)
```

```
## [1] 570608
```

```
mean(steps_day$dayTotal,na.rm=TRUE)
```

```
## [1] 9354.23
```

```
head(steps_day[,1:2])
```

```
## # A tibble: 6 x 2
##   date       dayTotal
##   <date>        <int>
## 1 2012-10-01        0
## 2 2012-10-02      126
## 3 2012-10-03    11352
## 4 2012-10-04    12116
## 5 2012-10-05    13294
## 6 2012-10-06    15420
```

```r
mean(steps_day$dayMean,na.rm=TRUE)
```

Q3) calculate the mean of steps taken per day: 37.3825996

```
## [1] 37.3826
```

```r
head(steps_day[,c(1,3)])
```

```
## # A tibble: 6 x 2
##   date       dayMean
##   <date>       <dbl>
## 1 2012-10-01 NaN
## 2 2012-10-02   0.438
## 3 2012-10-03  39.4
## 4 2012-10-04  42.1
## 5 2012-10-05  46.2
## 6 2012-10-06  53.5
```

```r
median(steps_day$dayMedian,na.rm=TRUE)
```

Q3) calculate the median of steps taken per day: 0

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

```r
head(steps_day[,c(1,4)])
```

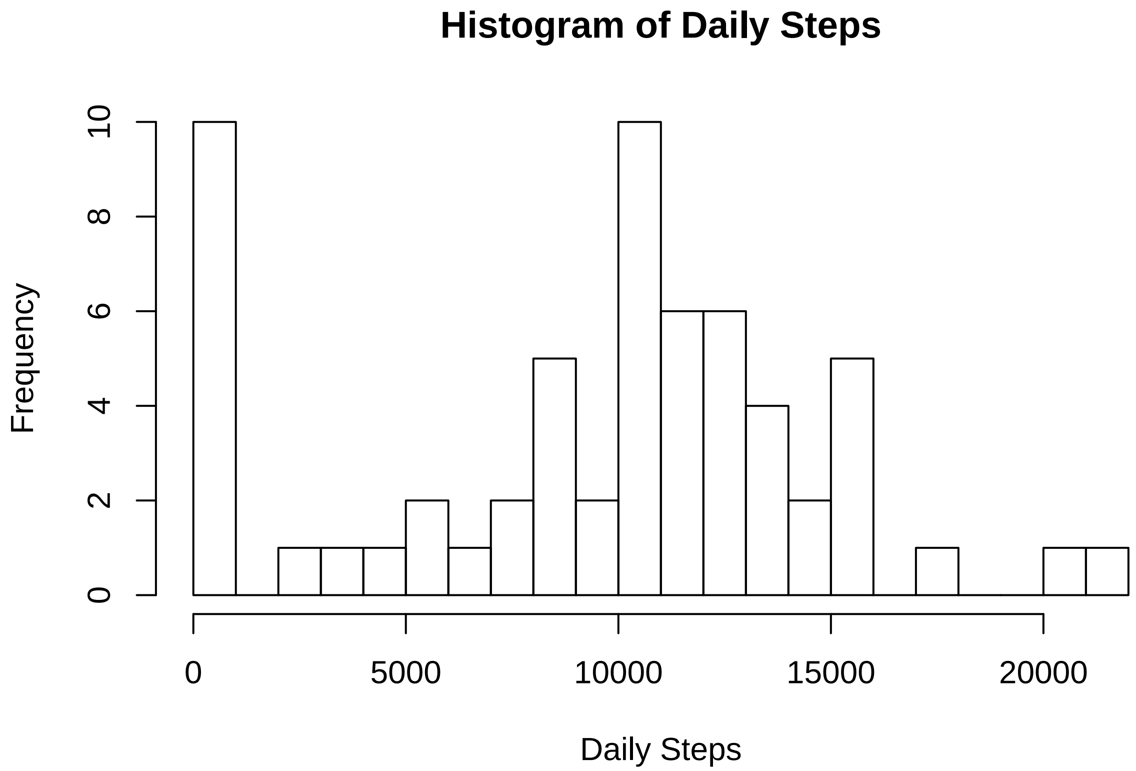
```
## # A tibble: 6 x 2
##   date       dayMedian
##   <date>         <dbl>
## 1 2012-10-01        NA
## 2 2012-10-02         0
## 3 2012-10-03         0
## 4 2012-10-04         0
## 5 2012-10-05         0
## 6 2012-10-06         0
```

Q2) plot a histogram of the total number of steps taken each day

A histogram represents the frequency distribution of *continuous* variables. Conversely, a bar graph is a comparison of *discrete* variables. Histogram presents *numerical* data whereas bar graph shows *categorical* data. The histogram is drawn in such a way that there is no gap between the bars.

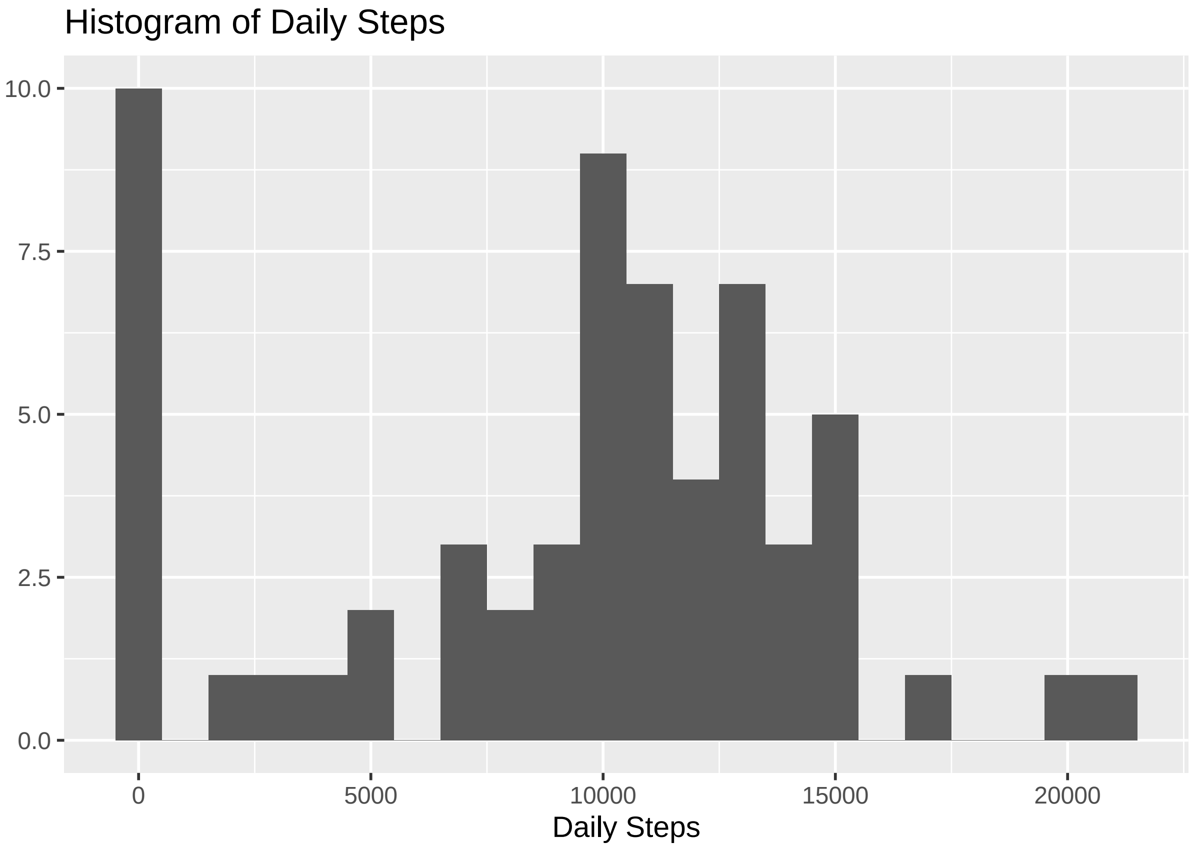
BasePlot

```r
hist(steps_day$dayTotal,breaks=20,main="Histogram of Daily Steps",xlab="Daily Steps")
```



ggplot2

```
qplot(steps_day$dayTotal, geom="histogram",binwidth=1000,main="Histogram of Daily Steps",xlab="Daily St
```



What is the average daily activity pattern?

Q4) Time Series Plot

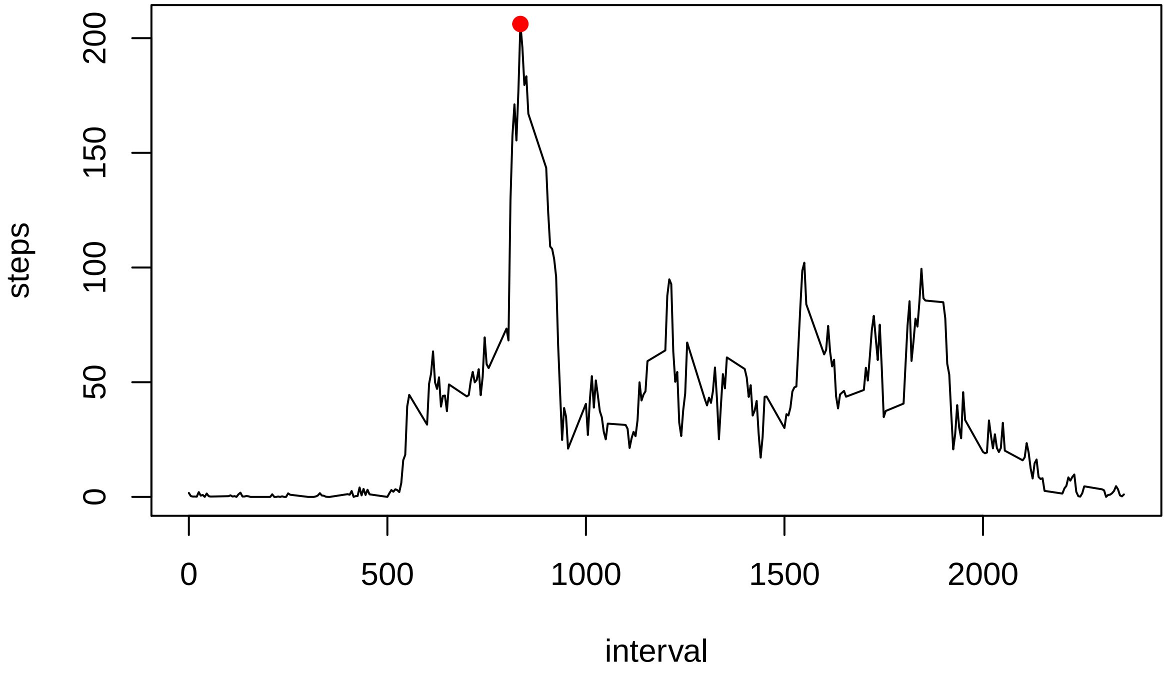
Here, a time series plot of the 5-minute interval (x-axis) and the average number of steps taken per 5 minute interval, averaged across all days (y-axis)

```
stepsPerint <- subset(activity,select=c(steps,interval))
stepsPerint <- stepsPerint %>% group_by(interval) %>%
  summarise_all(list(mean),na.rm=TRUE)
head(stepsPerint)
```

```
## # A tibble: 6 x 2
##   interval  steps
##      <int>  <dbl>
## 1        0 1.72
## 2        5 0.340
## 3       10 0.132
## 4       15 0.151
## 5       20 0.0755
## 6       25 2.09
```

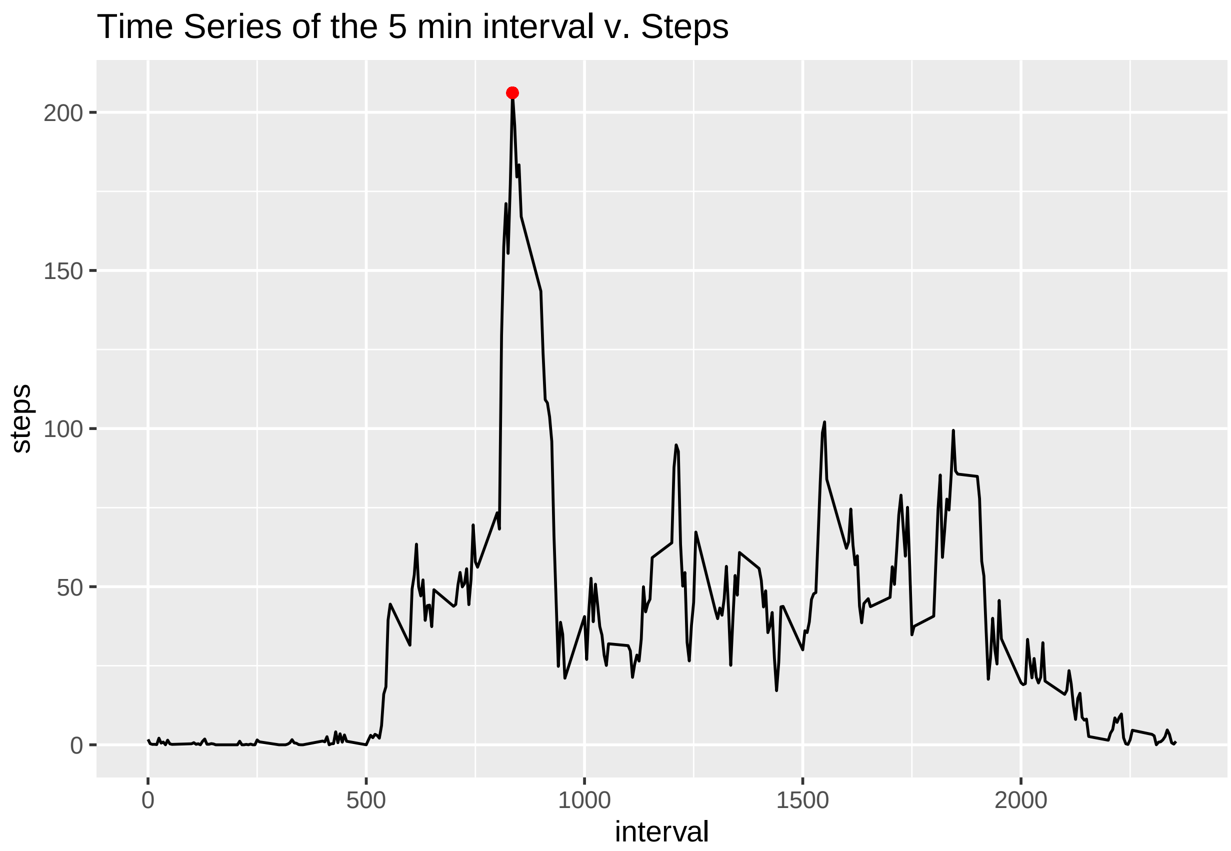
BasePlot

```
plot(stepsPerint,type="l")
points(stepsPerint$interval[which.max(stepsPerint$steps)],stepsPerint$steps[which.max(stepsPerint$steps
```



ggplot2

```
pointx<-stepsPerint$interval[which.max(stepsPerint$steps)]
pointy<-stepsPerint$steps[which.max(stepsPerint$steps)]
ggplot(stepsPerint, aes(interval, steps)) +
  geom_line() +
  geom_point(aes(pointx,pointy),col="red",pch = 19) +
  labs(title=("Time Series of the 5 min interval v. Steps"))
```



Q5) Calculation of interval number and maximum steps per interval:

The red dot on the above plots indicate **The Maximum Number of Steps** 206.1698113 that occurs at **Interval number:** 835

```
stepsPerint$interval[which.max(stepsPerint$steps)] #interval number
```

```
## [1] 835
```

```
stepsPerint$steps[which.max(stepsPerint$steps)] #Number of steps
```

```
## [1] 206.1698
```

Q6) Imputing missing values

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.

```

sum(is.na(activity$steps)) #Number missing
```

6a) Here we calculate and report the total number of missing step values in the dataset: Total of 2304 missing step values which is apprximately 13.1147541% of values.

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

```
mean(is.na(activity$steps)) # Percent missing
```

```
## [1] 0.1311475
```

Use the average for the 5 minute interval as our strategy for filling in all of the missing values in the dataset.

```
steps_5int <- subset(activity,select=c(steps,mins)) #35.86478
steps_5int <- steps_5int %>% group_by(mins) %>%
              summarise_all(list(avg_steps=mean),na.rm=TRUE)
steps_5int
```

6b) Calculate the average number of steps per interval.

```
## # A tibble: 12 x 2
##     mins avg_steps
##    <dbl>     <dbl>
##  1     0      34.7
##  2     5      35.9
##  3    10      39.0
##  4    15      42.1
##  5    20      37.9
##  6    25      37.2
##  7    30      37.4
##  8    35      34.9
##  9    40      35.3
## 10    45      37.5
## 11    50      38.5
## 12    55      38.2
```

From the table, we see the average steps taken during the 5 min interval is 35.8647798742138 or as integer of 35.

```
newActivity <- activity
Indx<-is.na(newActivity$steps) # index of NAs in the interval variable
sum(Indx,na.rm=TRUE) #count number of missing values
```

6c) Create a new dataset that is equal to the original dataset but with the missing data filled in with the average number of steps taken during the 5 min interval.

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

```
newActivity$steps[Indx] <- as.integer(steps_5int[2,2]) #fill the NAs
Indx<-is.na(newActivity$steps) # index of NAs in the interval variable
sum(Indx,na.rm=TRUE) #count number of missing values
```

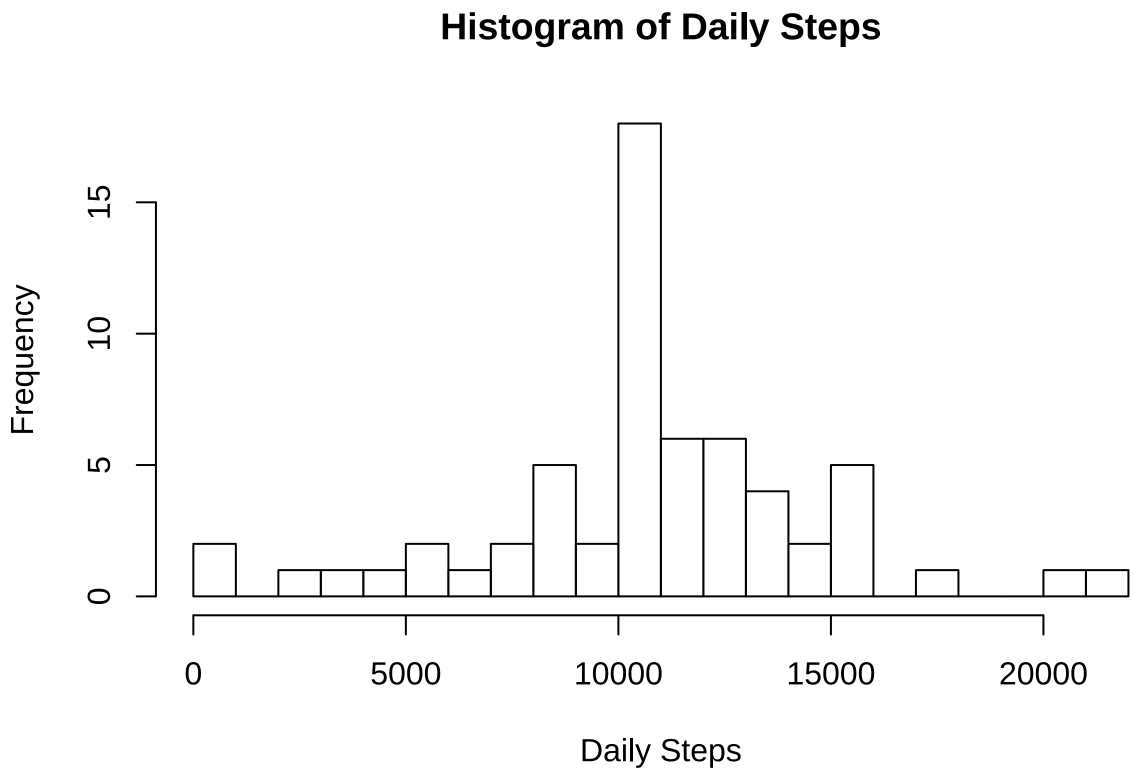
```
## [1] 0
```

7) Plot a Histogram with the new dataset after imputing missing data

```
newSteps_day <- subset(newActivity,select=c(steps,date))
newSteps_day <- newSteps_day %>% group_by(date) %>%
  summarise_all(list(dayTotal=sum,dayMean=mean,dayMedian=median),na.rm=TRUE)
```

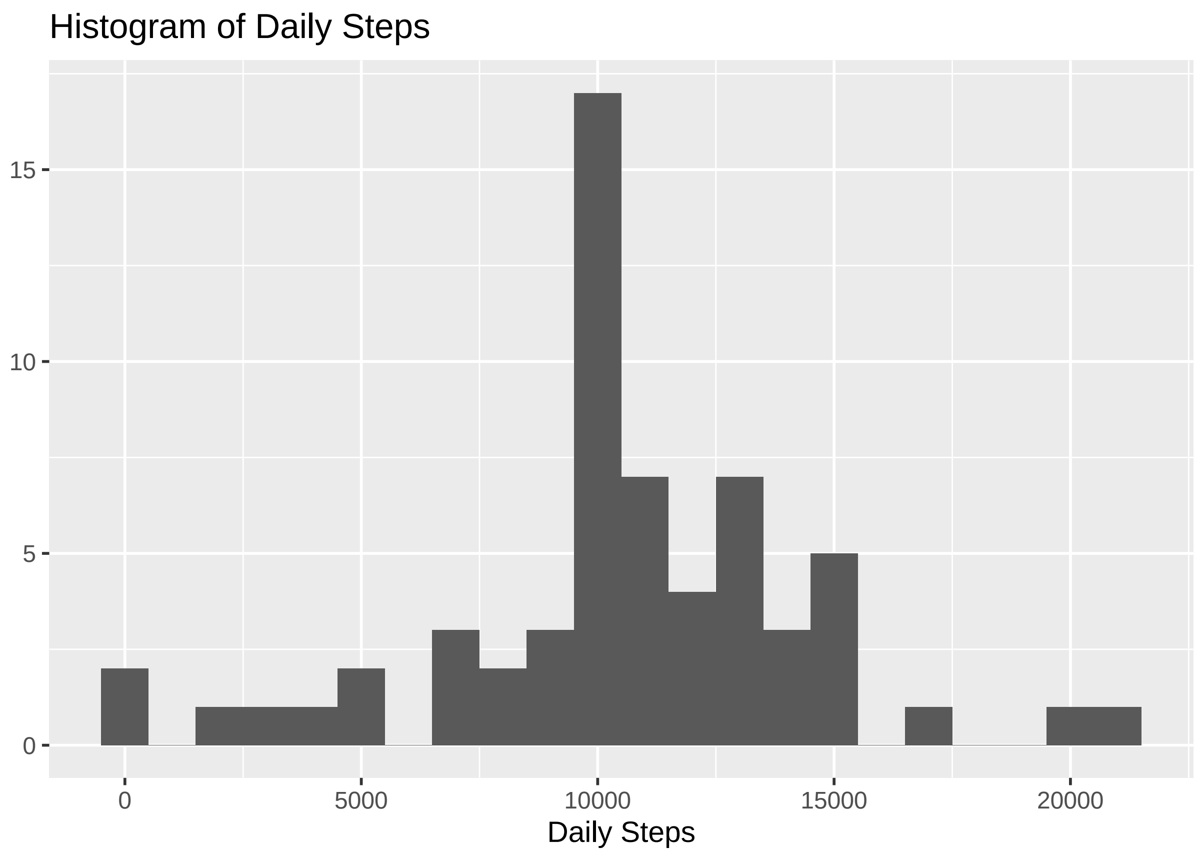
Base Plot

```
hist(newSteps_day$dayTotal,breaks=20,main="Histogram of Daily Steps",xlab="Daily Steps")
```



ggplot2

```
qplot(newSteps_day$dayTotal, geom="histogram",binwidth=1000,main="Histogram of Daily Steps",xlab="Daily
```



7a) Calculate and report the mean, median, and total number of steps taken per day

```
head(newSteps_day[,1:2]) # from dataframe above
```

total number of steps taken per day was calculated and is shown below

```
## # A tibble: 6 x 2
##   date       dayTotal
##   <date>        <int>
## 1 2012-10-01    10080
## 2 2012-10-02      126
## 3 2012-10-03    11352
## 4 2012-10-04    12116
## 5 2012-10-05    13294
## 6 2012-10-06    15420
```

```
head(newSteps_day[,c(1,3)])
```

the mean of steps taken daily was calculated as shown below

```
## # A tibble: 6 x 2
##   date       dayMean
##   <date>       <dbl>
## 1 2012-10-01  35
## 2 2012-10-02   0.438
## 3 2012-10-03  39.4
## 4 2012-10-04  42.1
## 5 2012-10-05  46.2
## 6 2012-10-06  53.5
```

```
head(newSteps_day[,c(1,4)])
```

the median of steps taken daily was calculated as shown below

```
## # A tibble: 6 x 2
##   date       dayMedian
##   <date>         <dbl>
## 1 2012-10-01        35
## 2 2012-10-02         0
## 3 2012-10-03         0
## 4 2012-10-04         0
## 5 2012-10-05         0
## 6 2012-10-06         0
```

7b) Do these values differ from the estimates from the first part of the assignment? In the first instance, we recognize that the total number of steps over the course of the dataset has increased from 570608 to 651248.

```
sum(steps_day$dayTotal)
```

```
## [1] 570608
```

```
sum(newSteps_day$dayTotal)
```

```
## [1] 651248
```

Combining the pre and post-imputing dataframes just for a simple comparison, we see the mean daily steps from the original dateset w/missing data decreased from 37.3825996 to 37.0701275. The median daily steps have also decresaed from 37.3784722 to 36.09375.

```
head(cbind(steps_day,newSteps_day[,2:4]),10)
```

```
##          date dayTotal  dayMean dayMedian dayTotal  dayMean dayMedian
## 1  2012-10-01        0      NaN        NA    10080 35.00000        35
## 2  2012-10-02      126  0.43750         0      126  0.43750         0
## 3  2012-10-03    11352 39.41667         0    11352 39.41667         0
## 4  2012-10-04    12116 42.06944         0    12116 42.06944         0
## 5  2012-10-05    13294 46.15972         0    13294 46.15972         0
## 6  2012-10-06    15420 53.54167         0    15420 53.54167         0
## 7  2012-10-07    11015 38.24653         0    11015 38.24653         0
## 8  2012-10-08        0      NaN        NA    10080 35.00000        35
## 9  2012-10-09    12811 44.48264         0    12811 44.48264         0
## 10 2012-10-10     9900 34.37500         0     9900 34.37500         0
```

```
summary(steps_day[,2:4]) #dataset with missing values
```

```
##     dayTotal        dayMean          dayMedian
##  Min.   :    0   Min.   : 0.1424   Min.   :0
##  1st Qu.: 6778   1st Qu.:30.6979   1st Qu.:0
##  Median :10395   Median :37.3785   Median :0
##  Mean   : 9354   Mean   :37.3826   Mean   :0
##  3rd Qu.:12811   3rd Qu.:46.1597   3rd Qu.:0
##  Max.   :21194   Max.   :73.5903   Max.   :0
##                  NA's   :8         NA's   :8
```

```
summary(newSteps_day[,2:4]) #dataset with imputed missing values filled in
```

```
##     dayTotal        dayMean          dayMedian
##  Min.   :   41   Min.   : 0.1424   Min.   : 0.00
##  1st Qu.: 9819   1st Qu.:34.0938   1st Qu.: 0.00
##  Median :10395   Median :36.0938   Median : 0.00
##  Mean   :10676   Mean   :37.0701   Mean   : 4.59
##  3rd Qu.:12811   3rd Qu.:44.4826   3rd Qu.: 0.00
##  Max.   :21194   Max.   :73.5903   Max.   :35.00
```

7c) What is the impact of imputing missing data on the estimates of the total daily number of steps? Imputing values for the missing data has decreased the mean daily steps from 37.3825996 to 37.0701275. The median daily steps have also decresaed from 37.3784722 to 36.09375.

Are there differences in activity patterns between weekdays and weekends?

The patterns are remarkably similar with the exception of the number of steps which we could have expected.

Here we 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 for more visualizations.

```
newSteps_day$week<-weekdays(newSteps_day$date)
newSteps_day$week<-ifelse(newSteps_day$week=="Saturday" | newSteps_day$week=="Sunday","weekend","weekda
head(newSteps_day)
```

```
## # A tibble: 6 x 5
##   date       dayTotal dayMean dayMedian week
##   <date>        <int>   <dbl>     <dbl> <chr>
## 1 2012-10-01    10080  35            35 weekday
## 2 2012-10-02      126   0.438         0 weekday
## 3 2012-10-03    11352  39.4           0 weekday
## 4 2012-10-04    12116  42.1           0 weekday
## 5 2012-10-05    13294  46.2           0 weekday
## 6 2012-10-06    15420  53.5           0 weekend
```

```
summary(newSteps_day) #dataframe for panel plot
```

```
##       date               dayTotal        dayMean          dayMedian    
##  Min.   :2012-10-01   Min.   :   41   Min.   : 0.1424   Min.   : 0.00  
##  1st Qu.:2012-10-16   1st Qu.: 9819   1st Qu.:34.0938   1st Qu.: 0.00  
##  Median :2012-10-31   Median :10395   Median :36.0938   Median : 0.00  
##  Mean   :2012-10-31   Mean   :10676   Mean   :37.0701   Mean   : 4.59  
##  3rd Qu.:2012-11-15   3rd Qu.:12811   3rd Qu.:44.4826   3rd Qu.: 0.00  
##  Max.   :2012-11-30   Max.   :21194   Max.   :73.5903   Max.   :35.00  
##      week          
##  Length:61         
##  Class :character  
##  Mode  :character  
##                    
##                    
##                    
```

Q8) 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).

```
newActivity$week<-weekdays(newActivity$date)
newActivity$week<-ifelse(newActivity$week=="Saturday" |   newActivity$week=="Sunday","weekend","weekday

stepsPerint <- subset(newActivity,select=c(steps,interval,week))
stepsPerint <- stepsPerint %>% group_by(week,interval) %>%
  summarise_all(list(avg_steps=mean,sumSteps=sum),na.rm=TRUE)
head(stepsPerint)
```

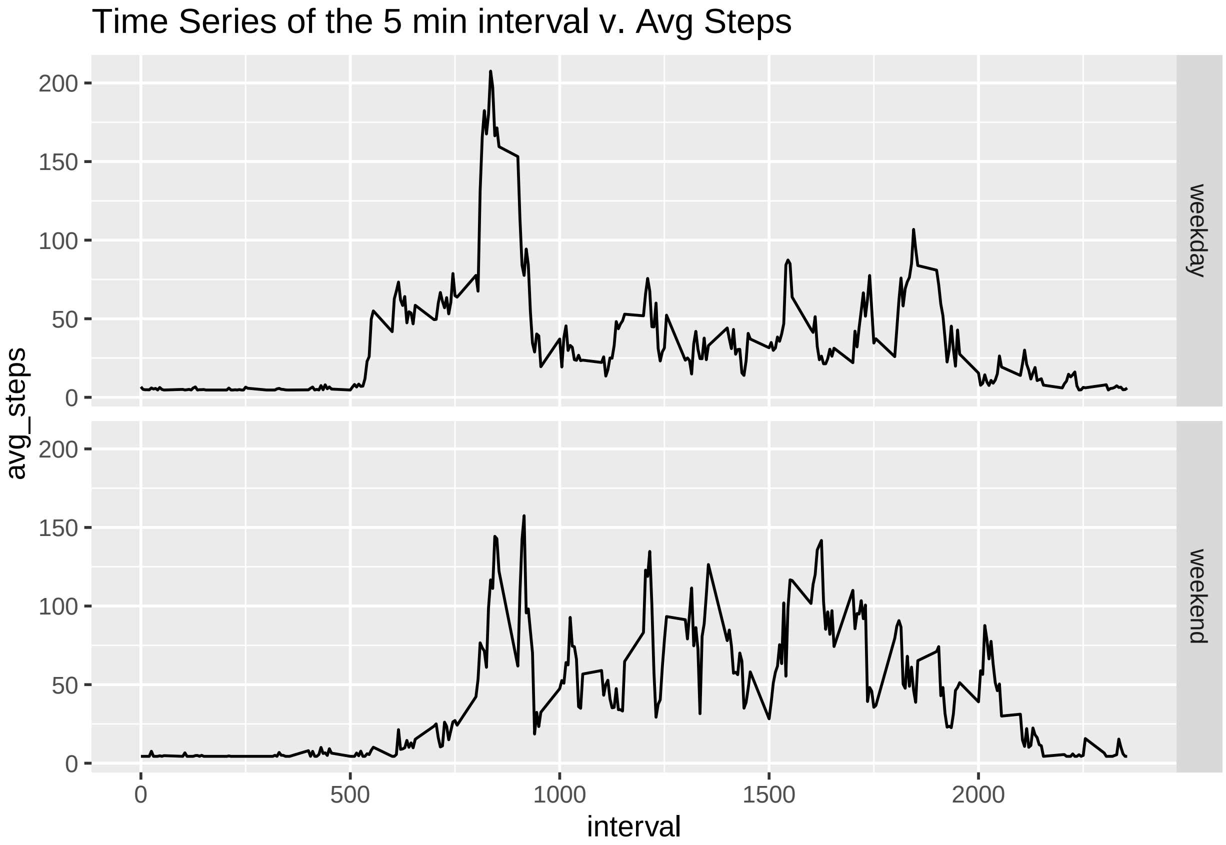
```
## # A tibble: 6 x 4
## # Groups:   week [1]
##   week    interval avg_steps sumSteps
##   <chr>      <int>     <dbl>    <int>
## 1 weekday        0      6.69      301
## 2 weekday        5      5.07      228
## 3 weekday       10      4.82      217
## 4 weekday       15      4.84      218
## 5 weekday       20      4.76      214
## 6 weekday       25      5.98      269
```

View a quick side by side comparison of weekday and weekend summary statistics

```
summary(cbind(subset(stepsPerint,week=="weekday",3:4),
              subset(stepsPerint,week=="weekend",3:4)))
```

```
##    avg_steps          sumSteps        avg_steps          sumSteps     
##  Min.   :  4.667   Min.   : 210.0   Min.   :  4.375   Min.   :  70.0  
##  1st Qu.:  6.589   1st Qu.: 296.5   1st Qu.:  5.344   1st Qu.:  85.5  
##  Median : 25.444   Median :1145.0   Median : 32.406   Median : 518.5  
##  Mean   : 35.293   Mean   :1588.2   Mean   : 42.069   Mean   : 673.1  
##  3rd Qu.: 49.622   3rd Qu.:2233.0   3rd Qu.: 70.500   3rd Qu.:1128.0  
##  Max.   :207.556   Max.   :9340.0   Max.   :157.500   Max.   :2520.0  
```

```
ggplot(stepsPerint, aes(interval, avg_steps)) +
  geom_line() +
  facet_grid(week~.) +
  labs(title=("Time Series of the 5 min interval v. Avg Steps"))
```


Time Series of the 5 min interval v. Avg Steps
avg_steps
weekday
weekend
interval