Coursera-Reproducible Research: Class Project 1

## Executive Summary

The assignment below summarizes the data collected from a personal activity monitoring device, collecting information in 5 minute intervals from an anonymous source over a 2 month period. Comparing weekday with weekend data identified small but noticable differences in movement throughout specific time intervals recorded. However, statistical analysis beyond basic descriptive analysis was not conducted due to the large number of missing data.Further analysis with complete data sets and multiple users are suggested.

## Analysis

The data was examined in a 2 step process. Basic descriptive analysis such as histograms and calculation of midpoints (mean and medians) of the observed data was conducted using  
- the raw data provided on the Coursera website (*Reproducible Research course*)  
- a data set imputing the missing values using the statistical mean of the corresponding time interval.

After reading the raw data into a dataframe, the R packages dplyr and tidyr were used to summarize the data for the required plots. The package ggplot was utilized for creating histograms and time series charts. Package lubridate allowed for assigning weekdays to the raw data, enabling the comparison of weekday and weekend data pattern.

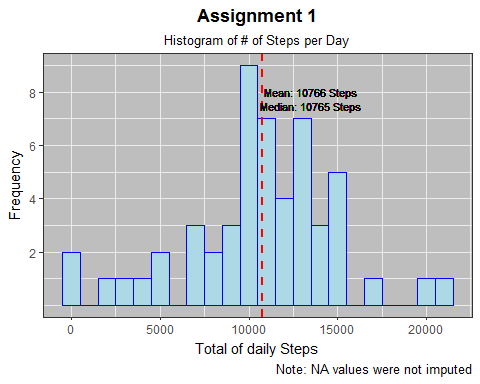
## Plot 1

The code below computed the data required for the first histogram. Missing data were estimated at 2,400 data points, about 15% of the raw data set.

library(dplyr)  
library(tidyr)  
library(ggplot2)  
library(readxl)  
library(lubridate)  
  
myDir <- "~/Z\_R/5) Reproducible Research/Assignment1"  
setwd(myDir)  
  
Activity.Data.File<- c("activity.xlsx")  
Raw.Activity.Data<- read\_excel(Activity.Data.File, sheet = "activity",   
 col\_names = TRUE,col\_types = NULL, skip = 0)  
Raw.Activity.Data$steps<- as.numeric((Raw.Activity.Data$steps))  
  
Daily.Total.Steps<- select(Raw.Activity.Data,-interval) %>% filter(!is.na(Raw.Activity.Data$steps))   
Daily.Total.Steps<- Daily.Total.Steps %>% group\_by(date) %>% summarise(steps = sum(steps))  
  
Mean.Steps<- mean(Daily.Total.Steps$steps, na.rm = TRUE)  
Median.Steps<- median(Daily.Total.Steps$steps, na.rm = TRUE)

.

The distribution of the raw data (minus incomplete data points), which summarizes the frequency of number of total steps per interval, indicates a fairly symmetric distribution. Even though the calculated midpoints (median and mean) are nearly identical, the plot indicates the data is slightly positively skewed.



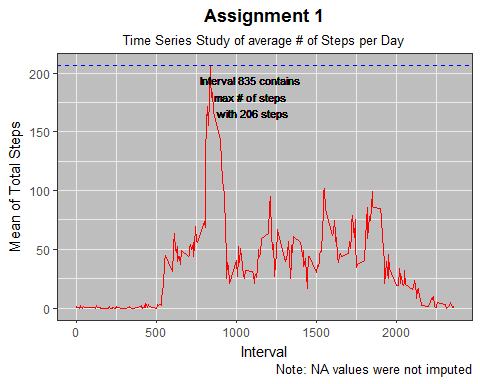
Please note that the calcuated midpoint data (mean and median) are displayed in the frequency chart.

## Plot 2

A time series analysis, plotting the mean of total steps by time interval indicates a rather erratic data pattern, peaking around time interval 835.The line graph is flat for the first 500 data points, largely due to the number of missing data points.

##  
## Time series Plot of Average Steps per Day  
##  
Average.Total.Steps<- select(Raw.Activity.Data,-date) %>% filter(!is.na(Raw.Activity.Data$steps))   
Average.Total.Steps<- Average.Total.Steps %>% group\_by(interval) %>% summarise(steps = mean(steps))  
  
## for annotation below the graph  
Max.Total.Steps.Interval<-Average.Total.Steps[Max.Interval<- which.max(Average.Total.Steps$steps),]  
Max.Interval<- as.integer(Max.Total.Steps.Interval[1,1])  
Max.Steps<- as.integer(Max.Total.Steps.Interval[1,2])  
  
## for annotation in the graph  
Max.Total.Steps<- as.integer(max(Average.Total.Steps$steps))

.

The time series analysis, plotting the mean of total steps by time interval indicates a rather erratic data pattern, peaking around time interval 835.The line graph is flat for the first 500 data points, largely due to the number of missing data points.  


Please note that the calculated data, determining the interval with the largest number of steps was included in the plot above.

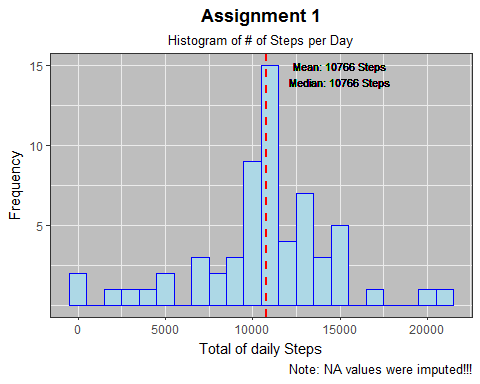
## Plot 3

The approach to imputing the missing data points utilized the mutate function of the package dplyr combined with the ifelse function, replacing every NA value with a mean value for the interval.

##  
## Imputing missing values  
Missing.Values<- sum(is.na(Raw.Activity.Data$steps))  
Missing.Values  
Raw.Activity.Data.Imputed<- Raw.Activity.Data %>%   
 group\_by(interval) %>%  
 mutate(steps= ifelse(is.na(steps),mean(steps,   
 na.rm=TRUE), steps))  
##  
## Check for missing values  
Missing.Values.Imputed<- sum(is.na(Raw.Activity.Data.Imputed$steps))  
##  
## Creating updated Data Sets for Plotting  
Daily.Total.Steps.Imputed<- Raw.Activity.Data.Imputed %>% group\_by(date) %>% summarise(steps = sum(steps))  
##  
## Computation of results  
Mean.Steps.Imputed<- mean(Daily.Total.Steps.Imputed$steps, na.rm = TRUE)  
Median.Steps.Imputed<- median(Daily.Total.Steps.Imputed$steps, na.rm = TRUE)

.

The chart below depicts the revised time series analysis using the imputed data (mean number of steps per interval).



Please note that the identification of the interval with the maximum mean number of steps was included in the time series chart above.

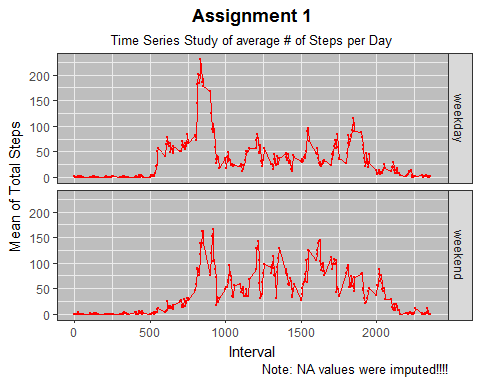
## Plot 4

The final chart compares data collected on weekends versus regular week days. The lubridate package identifies the day of the week based on the date given in the data set. A variable 'flag' was calculated to separate weekdays and weekends.The package dplyr was used to group the data by said flag.

Raw.Activity.Data.Imputed$day<- wday(Raw.Activity.Data.Imputed$date, label=TRUE)  
Raw.Activity.Data.Imputed$flag<- Raw.Activity.Data.Imputed$day  
levels(Raw.Activity.Data.Imputed$flag) <- list(  
 weekday = c("Mon", "Tues", "Wed", "Thurs", "Fri"),  
 weekend = c("Sun", "Sat"))  
  
Activity.Data.Imputed.Flag<- Raw.Activity.Data.Imputed %>%  
 group\_by(interval,flag) %>%  
 summarize(sum.steps=sum(steps, na.rm = TRUE),mean.steps= mean(steps,na.rm=TRUE))

The panel plot, comparing time series data of weekdays with weekends, was created using the facet\_grid function of ggplot.

The plot indicates similar patterns for both categories, weekdays and weekends. However, weekday patterns are more pronounced during intervals 500 to 1000. This is most likely caused by different time schedules and bahavioral patterns on weekends, potentially allowing for different rest periods during certain parts of the days. To fully evaluate the different in the data, a complete data set would be required.



## Appendix

In order to maintain the flow of the document, sample code used to generate the charts is included in the appdix.

*Code for first Plot*

## Plotting Histogram  
##  
p1<- ggplot(Daily.Total.Steps, aes(x = Daily.Total.Steps$steps)) +   
 geom\_histogram(binwidth = 1000, color="blue", fill = "lightblue") +  
 geom\_vline(aes(xintercept = Mean.Steps), color = "red", linetype = "dashed",  
 size = 1)  
  
## Formatting Axis  
##  
p1<- p1 + scale\_x\_continuous("Total of daily Steps") +  
 scale\_y\_continuous("Frequency", breaks = c(2,4,6,8)) +  
 theme(axis.text.x = element\_text(size=10, angle=45, hjust=1),  
 axis.text.y = element\_text(size=10))  
  
## Formatting Title & Canvas  
##  
p1<- p1 + labs(title = "Assignment 1",  
 subtitle = "Histogram of # of Steps per Day",  
 caption = "Note: NA values were not imputed") +   
 theme\_bw() + theme(panel.background = element\_rect(fill = 'grey')) +   
 theme(plot.title = element\_text(lineheight=.8, face="bold", hjust = 0.5)) +   
 theme(plot.subtitle = element\_text(lineheight=.8,hjust = 0.5))   
  
## Annotation  
##  
p1<- p1 + geom\_text(size = 3, x = 13500, y = 8,   
 aes(label = paste0("Mean: ",Mean.Steps, " Steps")))+  
 geom\_text(size = 3, x = 13500, y = 7.5,   
 aes(label = paste0("Median: ",Median.Steps, " Steps")))

*Code for second Plot*

p2<- ggplot(Average.Total.Steps, aes(x = interval, y = steps)) +   
 geom\_line(color= "red") +  
 geom\_hline(aes(yintercept = Max.Total.Steps), color = "blue", linetype = "dashed",  
 size = 0.5)  
  
## Formatting Axis  
##  
p2<- p2 + scale\_x\_continuous("Interval") + scale\_y\_continuous("Mean of Total Steps")   
   
  
## Formatting Title & Canvas  
##  
p2<- p2 + labs(title = "Assignment 1",  
 subtitle = "Time Series Study of average # of Steps per Day",  
 caption = "Note: NA values were not imputed") +   
 theme\_bw() + theme(panel.background = element\_rect(fill = 'grey')) +  
 theme(plot.title = element\_text(lineheight=.8, face="bold", hjust = 0.5)) +   
 theme(plot.subtitle = element\_text(lineheight=.8,hjust = 0.5))   
  
## Annotation  
##  
p2<- p2 + geom\_text(size = 3, x = 1100, y = 180, aes(label =   
 paste0("Interval ",Max.Interval, " contains \nmax # of steps \nwith ",Max.Total.Steps, " steps")))

*Code for third Plot*

## Plotting Histogram  
##  
p3<- ggplot(Daily.Total.Steps.Imputed, aes(x = steps)) +   
 geom\_histogram(binwidth = 1000, color="blue", fill = "lightblue") +  
 geom\_vline(aes(xintercept = Mean.Steps.Imputed), color = "red", linetype = "dashed",  
 size = 1)  
  
## Formatting Axis  
##  
p3<- p3 + scale\_x\_continuous("Total of daily Steps") +  
 scale\_y\_continuous("Frequency", breaks = c(5,10,15))  
  
## Formatting Title & Canvas  
##  
p3<- p3 + labs(title = "Assignment 1",  
 subtitle = "Histogram of # of Steps per Day",  
 caption = "Note: NA values were imputed!!!") +   
 theme\_bw() + theme(panel.background = element\_rect(fill = 'grey')) +   
 theme(plot.title = element\_text(lineheight=.8, face="bold", hjust = 0.5)) +   
 theme(plot.subtitle = element\_text(lineheight=.8,hjust = 0.5))   
  
## Annotation  
##  
p3<- p3 + geom\_text(size = 3, x = 15000, y = 15,   
 aes(label = paste0("Mean: ",as.integer(Mean.Steps.Imputed), " Steps")))+  
 geom\_text(size = 3, x = 15000, y = 14,   
 aes(label = paste0("Median: ",as.integer(Median.Steps.Imputed), " Steps")))

*Code for fourth Plot*

p4<- ggplot(Activity.Data.Imputed.Flag, aes(x = interval, y = mean.steps)) +   
 geom\_line(color= "red")+ geom\_point(size = 0.5, color = "red")  
p4<- p4 + facet\_grid(flag ~ .)  
   
## Formatting Title & Canvas  
##  
p4<- p4 + labs(title = "Assignment 1",  
 subtitle = "Time Series Study of average # of Steps per Day",  
 caption = "Note: NA values were imputed!!!!") +   
 theme\_bw() + theme(panel.background = element\_rect(fill = 'grey')) +  
 theme(plot.title = element\_text(lineheight=.8, face="bold", hjust = 0.5)) +   
 theme(plot.subtitle = element\_text(lineheight=.8,hjust = 0.5))   
  
## Formatting Axis  
##  
p4<- p4 + scale\_x\_continuous("Interval") + scale\_y\_continuous("Mean of Total Steps")