Assignment1

A. Loading and preprocessing the data

1.Load the data (i.e. read.csv())

Load ggplot2 and plyr library

```
library(ggplot2)
library(plyr)
```

Read data into a new data frame called data

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

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

```
#Format the date column into date class
data$date <- as.Date(data$date)
#Sort out the NA steps value for further calculation
na <- is.na(data$steps)
cleanData <- data[!na,]
```

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

1.Calculate the total number of steps taken per day

Calculate total number of steps based on the clean data and store in the variable totalSPD

```
totalSPD <- aggregate(cleanData$steps, by=list(cleanData$date),sum)
#review the newly created dataframe
head(totalSPD)
```

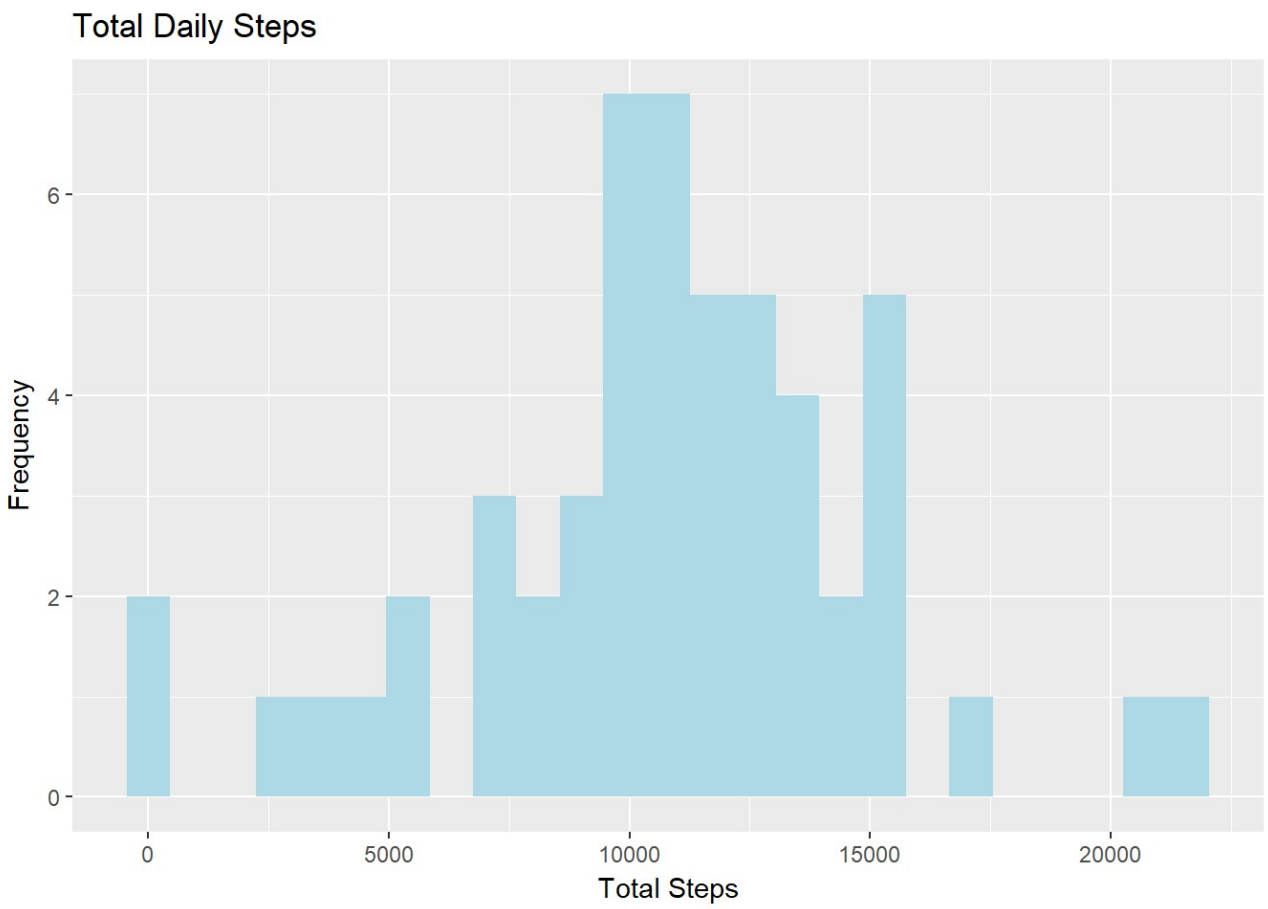
```
##      Group.1     x
## 1 2012-10-02   126
## 2 2012-10-03 11352
## 3 2012-10-04 12116
## 4 2012-10-05 13294
## 5 2012-10-06 15420
## 6 2012-10-07 11015
```

```
#rename the column
totalSPD <- rename(totalSPD, c("Group.1"="Date","x"="totalsteps"))
#review once again
head(totalSPD)
```

```
##         Date totalsteps
## 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.Make a histogram of the total number of steps taken each day

```
#create object g as layout for the histogram
g <- ggplot(totalSPD, aes(x=totalsteps))
#choose color, dimension and create labels for the histpgram
g + geom_histogram(fill="lightblue", binwidth = 900) + labs(title = "Total Daily Step
s", x="Total Steps", y="Frequency")
```



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

```
#Mean of steps
mean_og <-mean(totalSPD$totalsteps)
#Median of steps
median_og <-median(totalSPD$totalsteps)
```

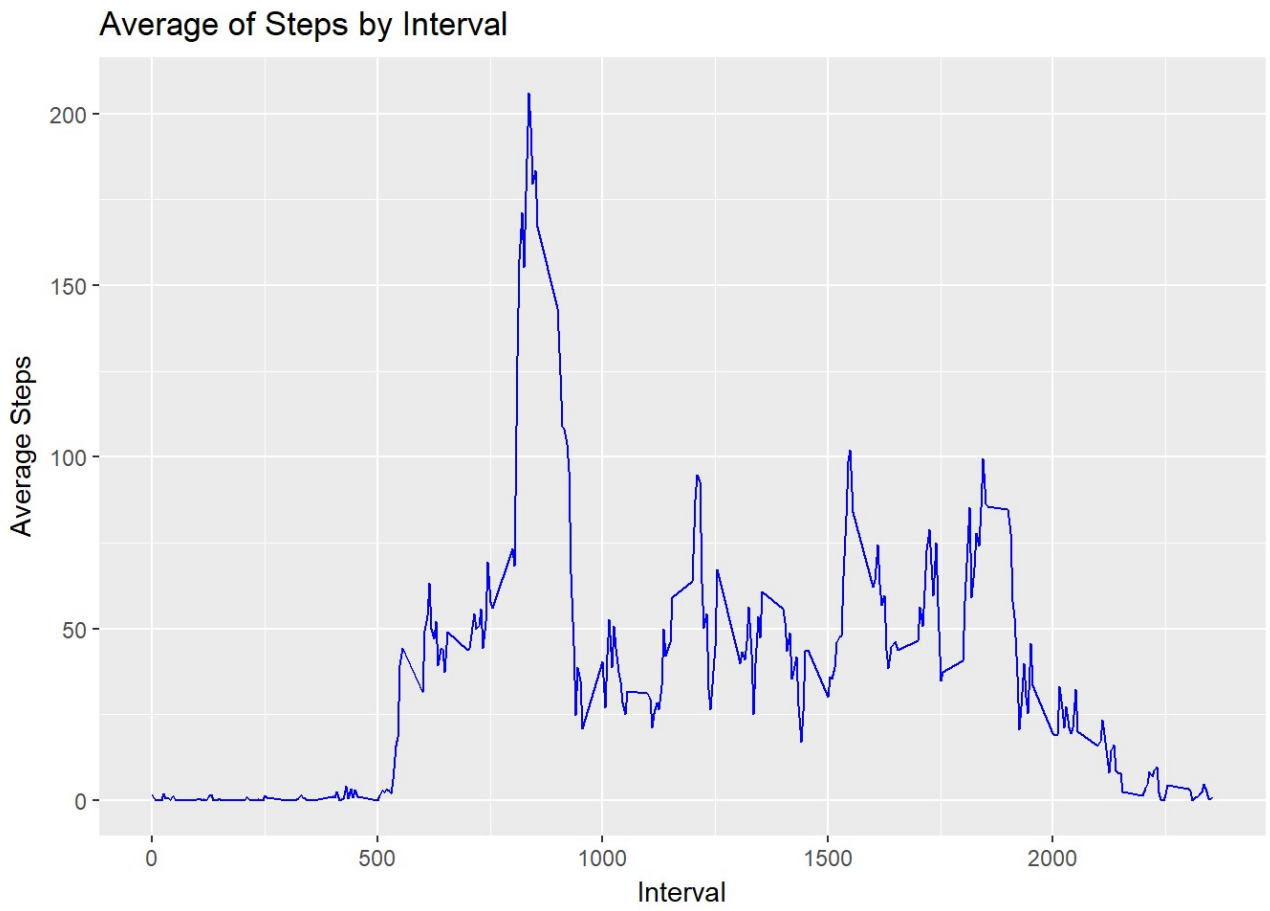
C. What is the average daily activity pattern?

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)

```
#create a new data frame avgSBI(average steps by interval) based on the cleanData
avgSBI <- aggregate(cleanData$steps, by=list(cleanData$interval),mean)
#rename the column
avgSBI <- rename(avgSBI, c("Group.1"="interval","x"="averagesteps"))
#review the data frame
head(avgSBI)
```

```
##   interval averagesteps
## 1        0    1.7169811
## 2        5    0.3396226
## 3       10    0.1320755
## 4       15    0.1509434
## 5       20    0.0754717
## 6       25    2.0943396
```

```
#create the new object "p" to make the plot as required
p <- ggplot(avgSBI, aes(x=interval,y=averagesteps), type="l")
# create the time series plot
p + geom_line(color="blue") + labs(title="Average of Steps by Interval", x="Interva
l", y="Average Steps")
```



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

```
# Find the max Average steps value and store it in the new variable "maxSteps"
maxSteps <- avgSBI[which.max(avgSBI$averagesteps),]
# Review the value
maxSteps
```

```
##     interval averagesteps
## 104      835     206.1698
```

D. Imputing missing values

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

```
# Count the total number of missing value by using the variable "na" defined in the A part of the assignment
sum(na)
```

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

2.Devise a strategy for filling in all of the missing values in the dataset

```
#The average steps by Interval will be used to replace the na value
#Create a new variable meanSI to store the average steps by interval based on the original data frame
meanSI <- aggregate(steps~interval,data,FUN=mean)
#Create a new dataframe that merges the original data frame with the means of steps by interval
newData <- merge(x=data,y=meanSI,by="interval")
#Take a sneakpeak at the new dataframe:
str(newData)
```

```
## 'data.frame':    17568 obs. of  4 variables:
##  $ interval: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ steps.x : int  NA 0 0 0 0 0 0 0 0 0 ...
##  $ date    : Date, format: "2012-10-01" "2012-11-23" ...
##  $ steps.y : num  1.72 1.72 1.72 1.72 1.72 ...
```

```
#Replace na steps values in newData with average steps by interval, create a new column called steps
newData$steps <- ifelse(is.na(newData$steps.x),newData$steps.y,newData$steps.x)
```

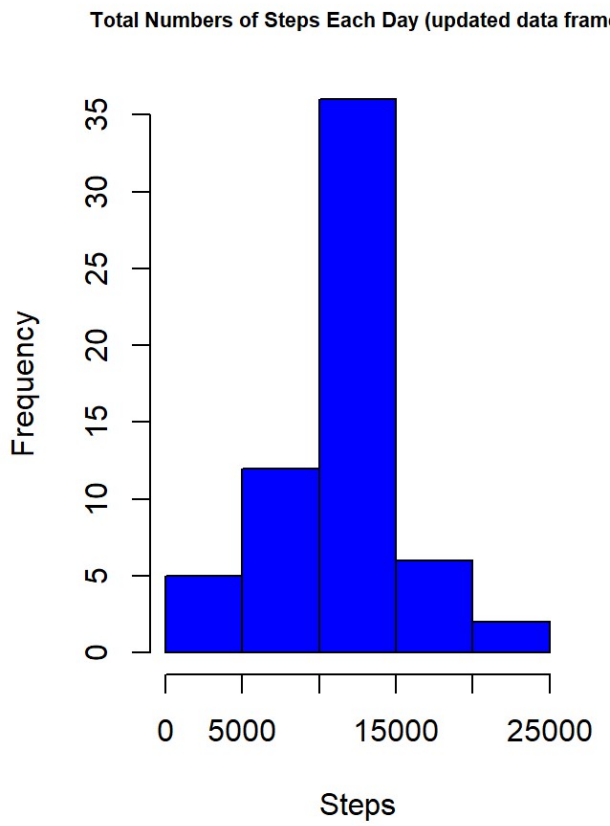
3.Create a new dataset that is equal to the original dataset but with the missing data filled in.

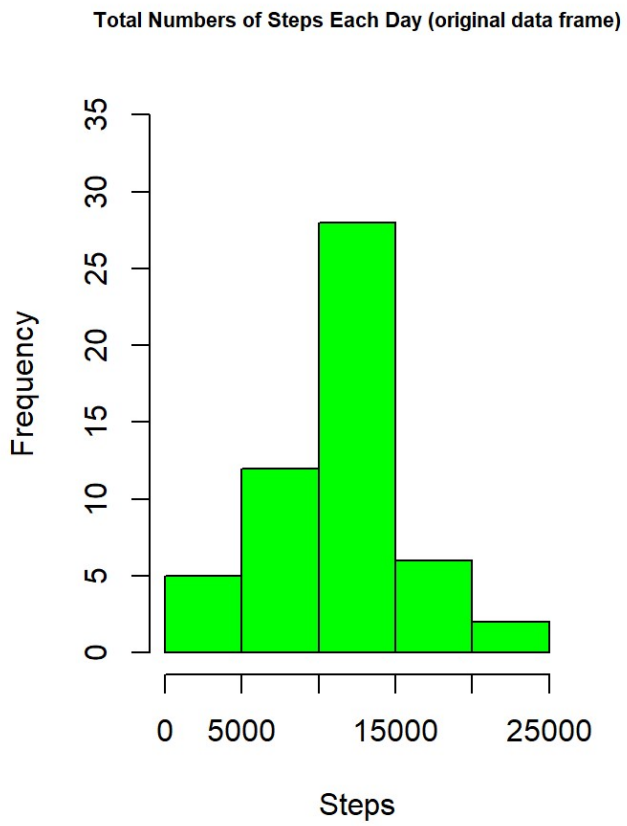
```
head(newData)
```

```
##   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
```

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

```
#Total steps by day based on new data frame
totalSPD_new <- aggregate(steps~date,newData,FUN=sum)
##Set up plot to compare old and new data frame
par(mfrow=c(1,2))
hist(totalSPD_new$steps, col="blue", ylim=c(0,35),xlab = "Steps", ylab ="Frequency",
     main= "Total Numbers of Steps Each Day (updated data frame)", cex.main=0.7)
hist(totalSPD$totalsteps, col="green", ylim=c(0,35),xlab = "Steps", ylab ="Frequenc
y",
     main= "Total Numbers of Steps Each Day (original data frame)",cex.main=0.7)
```






```
#comparing the old and new mean/median between the original and updated data frame
par(mfrow=c(2,1))
newmean <- mean(totalSPD_new$steps)
newmedian <- median(totalSPD_new$steps)
paste("New mean:  ",round(newmean,2),", ","Original mean  :",round(mean_og,2),"Differe
nce  :", round(newmean-mean_og,2))
```

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

```
paste("New median:  ",round(newmedian,2),", ","Original median  :",round(median_og,
2),"Difference  :", round(newmedian-median_og,2))
```

```
## [1] "New median:   10766.19 ,  Original median  : 10765 Difference  : 1.19"
```

E. Are there differences in activity patterns between weekdays and weekends?

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 packages "chron" to use the function is.weekend
library(chron)
#Create a new variable "day" to sort out Weekday/Weekend in the new data frame
newData$day <- ifelse(is.weekend(newData$date),"Weekend","Weekday")
head(newData)
```

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

```
#Create new table with average steps over interval and day of week
meanSI_new <- aggregate(steps~interval+day,newData,FUN=mean)
head(meanSI_new)
```

```
##   interval     day      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
```

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)

```
#Create plot to compare activity pattern between weekday and weekend
ggplot(meanSI_new, aes(x=interval, y=steps),type="l") + geom_line(color="blue", size=
0.5) +facet_grid(day~.) +labs(x="Interval", y="Average Steps", title = "Activity Patte
rn")
```