Step1 - Code for reading in the dataset and/or processing the data

Assuming the data is in the current project

```
library(knitr)
library(ggplot2)
library(dplyr)

Sys.setlocale(category = "LC_ALL", locale = "US")
```

```
## [1] "LC_COLLATE=English_United States.1252;LC_CTYPE=English_United States.1252;LC_MONETARY=English_United S
```

```
activity <- read.csv("./activity.csv", header = TRUE)
# convert date to convenient format
activity$date <- as.POSIXct(activity$date, format = "%Y-%m-%d", tz="EST")
weekday <- weekdays(activity$date)
activity <- cbind(activity,weekday)
summary(activity)
```

```
##      steps              date                interval            weekday
##  Min.   :  0.00   Min.   :2012-10-01   Min.   :   0.0   Friday   :2592
##  1st Qu.:  0.00   1st Qu.:2012-10-16   1st Qu.: 588.8   Monday   :2592
##  Median :  0.00   Median :2012-10-31   Median :1177.5   Saturday :2304
##  Mean   : 37.38   Mean   :2012-10-31   Mean   :1177.5   Sunday   :2304
##  3rd Qu.: 12.00   3rd Qu.:2012-11-15   3rd Qu.:1766.2   Thursday :2592
##  Max.   :806.00   Max.   :2012-11-30   Max.   :2355.0   Tuesday  :2592
##  NA's   :2304                                           Wednesday:2592
```

```
head(activity)
```

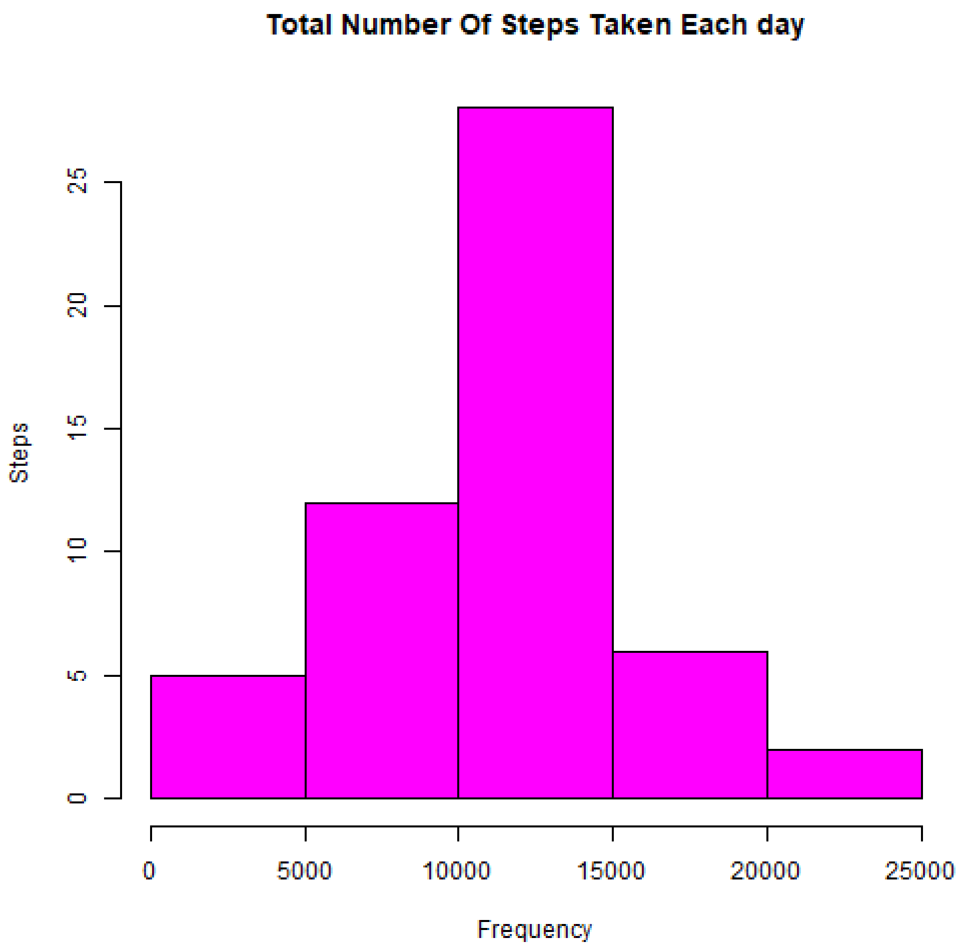
```
##   steps       date interval weekday
## 1    NA 2012-10-01        0  Monday
## 2    NA 2012-10-01        5  Monday
## 3    NA 2012-10-01       10  Monday
## 4    NA 2012-10-01       15  Monday
## 5    NA 2012-10-01       20  Monday
## 6    NA 2012-10-01       25  Monday
```

Step2 - Histogram of the total number of steps taken each day

```
summed_steps<- aggregate(steps ~ date, activity, FUN=sum)
head(summed_steps)
```

```
##         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
```

```
#Use the base plotting to show a histogram
hist(summed_steps$steps,
     col="magenta",
     xlab = "Frequency",
     ylab = "Steps",
     main = "Total Number Of Steps Taken Each day")
```



Step3 - Mean and median number of steps taken each day

```
Mean <- mean(summed_steps$steps)
Median <- median(summed_steps$steps)
#Print results of mean and median
Mean
```

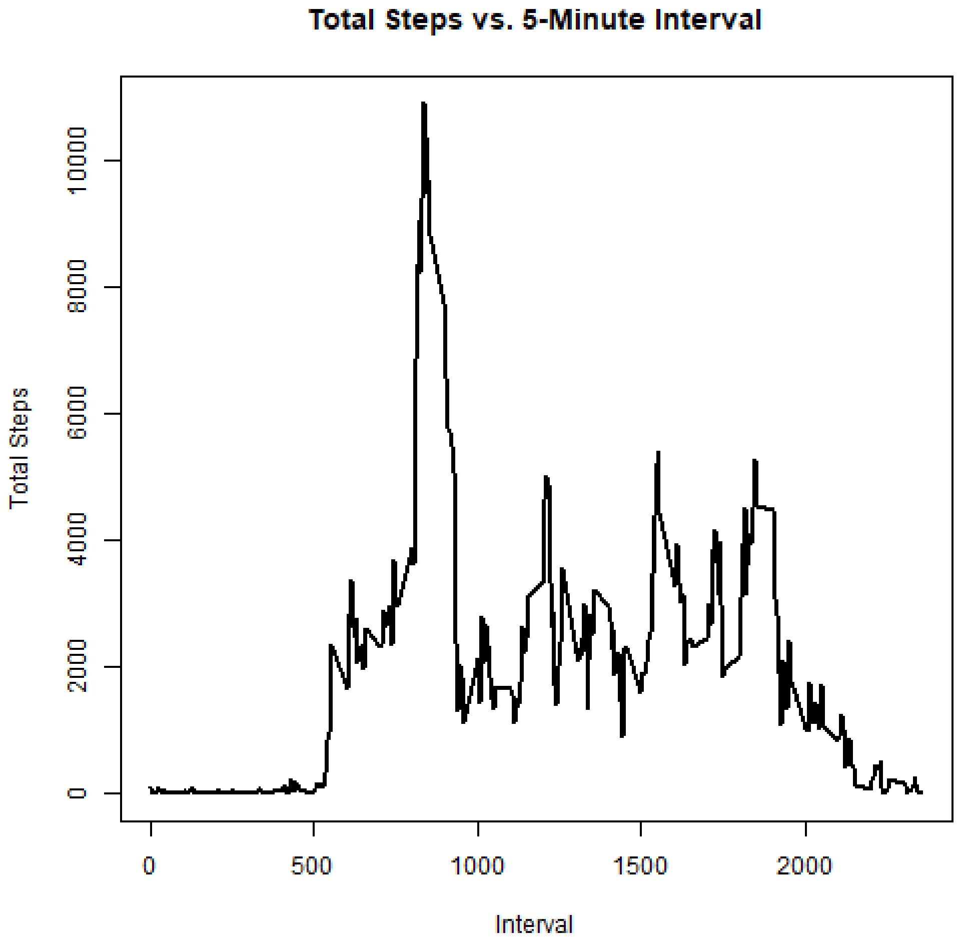
```
## [1] 10766.19
```

```
Median
```

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

Step4 - Time series plot of the average number of steps taken: What is the average daily activity pattern?

```
# Make a time series plot (i.e. type = "l") of the 5-minute interval (x-axis) and the average number of steps
#aggregation of steps over time interval (of 5 min)
agginterval <- aggregate(steps ~ interval, activity, FUN=sum)
#Plotting line graph using plot() from Base Plotting for Total Steps vs 5-Minute Interval
plot(agginterval$interval, agginterval$steps,
     type = "l", lwd = 2,
     xlab = "Interval",
     ylab = "Total Steps",
     main = "Total Steps vs. 5-Minute Interval")
```



Step5 - Which 5-minute interval, on average across all the days in the dataset, contains the maximum number of steps?

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

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

Imputing missing values

Step6 - Code to describe and show a strategy for imputing missing data

```
# 1.Calculate and report the total number of missing values in the dataset (i.e. the total number of rows with
table(is.na(activity))
```

```
## 
## FALSE  TRUE 
## 67968  2304
```

```
# 2.Devise a strategy for filling in all of the missing values in the dataset. The strategy does not need to b
mean_interval<- aggregate(steps ~ interval, activity, FUN=mean)
newMerged <- merge(x=activity, y=mean_interval, by="interval")
#Replace the NA values with the mean for that 5-minute interval
newMerged$steps <- ifelse(is.na(newMerged$steps.x), newMerged$steps.y, newMerged$steps.x)
#Here is the merged dataset which will be subsetted in the next step by removing not required columns
head(newMerged)
```

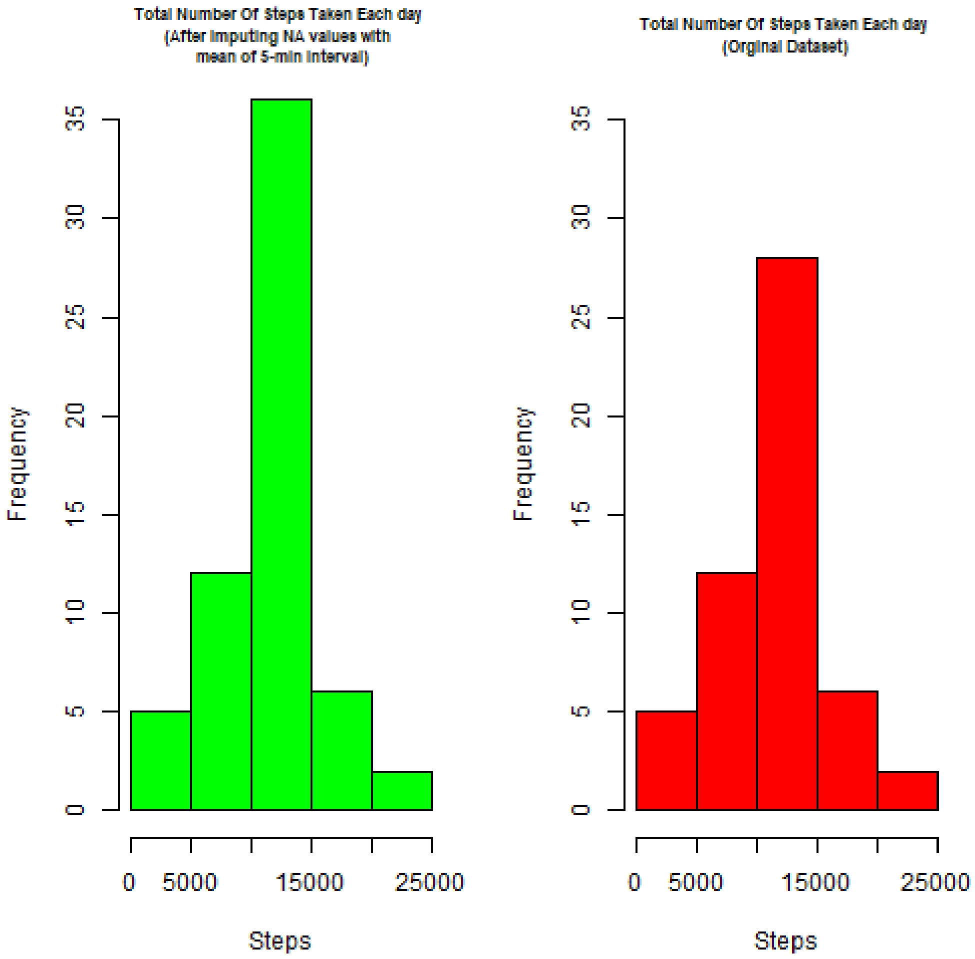
```
##   interval steps.x       date  weekday  steps.y    steps
## 1        0      NA 2012-10-01   Monday 1.716981 1.716981
## 2        0       0 2012-11-23   Friday 1.716981 0.000000
## 3        0       0 2012-10-28   Sunday 1.716981 0.000000
## 4        0       0 2012-11-06  Tuesday 1.716981 0.000000
## 5        0       0 2012-11-24 Saturday 1.716981 0.000000
## 6        0       0 2012-11-15 Thursday 1.716981 0.000000
```

```
# 3. Create a new dataset that is equal to the original dataset but with the missing data filled in.
newMerged <- select(newMerged, steps, date, interval)
head(newMerged)
```

```
##      steps       date interval
## 1 1.716981 2012-10-01        0
## 2 0.000000 2012-11-23        0
## 3 0.000000 2012-10-28        0
## 4 0.000000 2012-11-06        0
## 5 0.000000 2012-11-24        0
## 6 0.000000 2012-11-15        0
```

Step7 - 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 to
#Aggregating(summation) of steps over date
aggsteps_new<- aggregate(steps ~ date, newMerged, FUN=sum)
#Plotting : Setting 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(aggsteps_new$steps,
     col="green",
     xlab = "Steps",
     ylab = "Frequency",
     ylim = c(0,35),
     main = "Total Number Of Steps Taken Each day \n(After imputing NA values with \n mean of 5-min interval)"
     cex.main = 0.7)
#Histogram with the orginal dataset
hist(summed_steps$steps,
     col="red",
     xlab = "Steps",
     ylab = "Frequency",
     ylim = c(0,35),
     main = "Total Number Of Steps Taken Each day \n(Orginal Dataset)",
     cex.main = 0.7)
```



```
par(mfrow=c(1,1)) #Resetting the panel
Mean_new <- mean(aggsteps_new$steps)
Median_new <- median(aggsteps_new$steps)
#Comparing Means
paste("New Mean      :", round(Mean_new,2), "," ,
      " Original Mean :", round(Mean,2),"," ,
      " Difference :",round(Mean_new,2) -  round(Mean,2))
```

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

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

```
## [1] "New Median    : 10766.1886792453 ,  Original Median : 10765 ,  Difference : 1.19"
```

Step 8 -Are there differences in activity patterns between weekdays and weekends?

```
#install.packages("chron")
library(chron)
# Create a new factor variable in the dataset with two levels – “weekday” and “weekend” indicating whether a g
table(is.weekend(newMerged$date))
```

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

```
newMerged$dayofweek <- ifelse(is.weekend(newMerged$date), "weekend", "weekday")
table(newMerged$dayofweek)
```

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

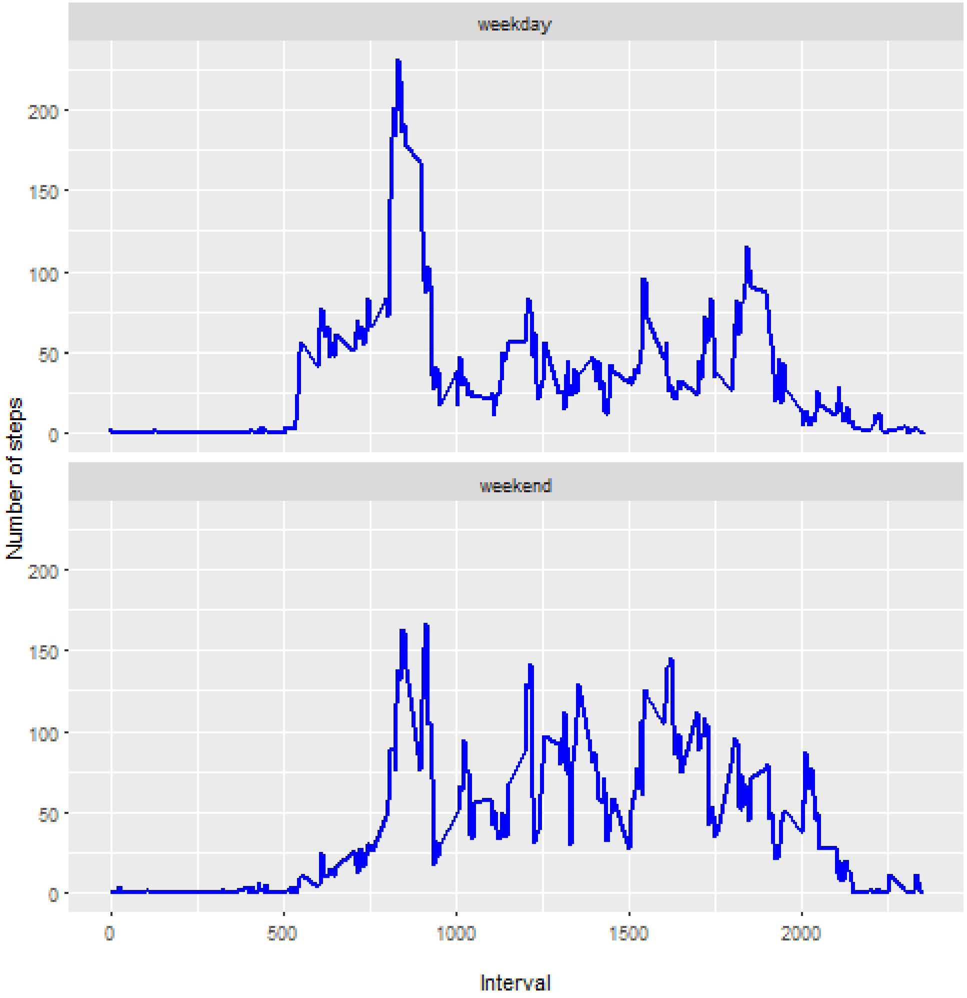
```
head(newMerged)
```

```
##      steps       date interval dayofweek
## 1 1.716981 2012-10-01        0   weekday
## 2 0.000000 2012-11-23        0   weekday
## 3 0.000000 2012-10-28        0   weekend
## 4 0.000000 2012-11-06        0   weekday
## 5 0.000000 2012-11-24        0   weekend
## 6 0.000000 2012-11-15        0   weekday
```

```
# Make a panel plot containing a time series plot (i.e. type = "l") of the 5-minute interval (x-axis) and the
meaninterval_new<- aggregate(steps ~ interval + dayofweek, newMerged, FUN=mean)
head(meaninterval_new)
```

```
##   interval dayofweek      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
```

```
ggplot(meaninterval_new, aes(x=interval, y=steps)) +
  geom_line(color="blue", size=1) +
  facet_wrap(~dayofweek, nrow=2) +
  labs(x="\nInterval", y="\nNumber of steps")
```


weekday
weekend
Number of steps
Interval