RMarkdownAssignment1

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 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.

Load all required library

```
library(dplyr)
library(lubridate)
library(ggplot2)
```

Read-in dataset from link

```
temp <-tempfile()
download.file("https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip", temp)
activity <- read.csv(unzip(temp,"activity.csv"), header = TRUE, sep ="," )
unlink(temp)
```

Converting data items into correct format

```
activity$date <- as.Date(activity$date, format="%Y-%m-%d")
activity$weekday <- weekdays(activity$date)
activity$weekday <- as.factor(activity$weekday)
activity$interval <- as.factor(activity$interval)
```

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

1. Calculate the total number of steps taken per day

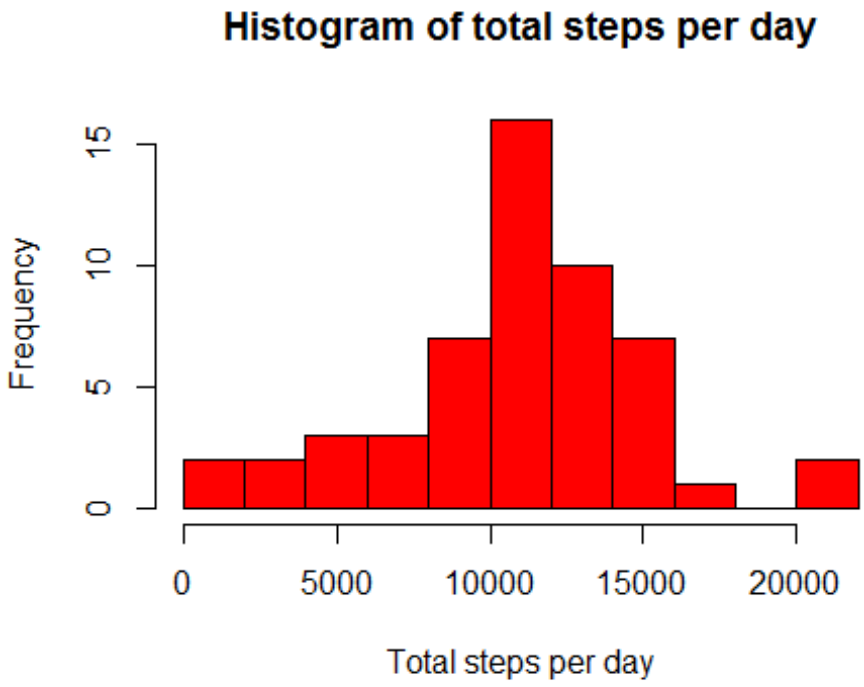
```
totalsteps <- aggregate(steps ~ date, activity, sum)
names(totalsteps)[names(totalsteps)=="steps"] <- "totalsteps"
totalsteps
```

```
##          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
## 7  2012-10-09      12811
## 8  2012-10-10       9900
## 9  2012-10-11      10304
## 10 2012-10-12      17382
## 11 2012-10-13      12426
## 12 2012-10-14      15098
## 13 2012-10-15      10139
## 14 2012-10-16      15084
## 15 2012-10-17      13452
## 16 2012-10-18      10056
## 17 2012-10-19      11829
## 18 2012-10-20      10395
## 19 2012-10-21       8821
## 20 2012-10-22      13460
## 21 2012-10-23       8918
## 22 2012-10-24       8355
## 23 2012-10-25       2492
## 24 2012-10-26       6778
## 25 2012-10-27      10119
## 26 2012-10-28      11458
## 27 2012-10-29       5018
## 28 2012-10-30       9819
## 29 2012-10-31      15414
## 30 2012-11-02      10600
## 31 2012-11-03      10571
## 32 2012-11-05      10439
## 33 2012-11-06       8334
## 34 2012-11-07      12883
## 35 2012-11-08       3219
## 36 2012-11-11      12608
## 37 2012-11-12      10765
## 38 2012-11-13       7336
## 39 2012-11-15         41
## 40 2012-11-16       5441
## 41 2012-11-17      14339
## 42 2012-11-18      15110
## 43 2012-11-19       8841
## 44 2012-11-20       4472
## 45 2012-11-21      12787
## 46 2012-11-22      20427
## 47 2012-11-23      21194
## 48 2012-11-24      14478
## 49 2012-11-25      11834
## 50 2012-11-26      11162
## 51 2012-11-27      13646
## 52 2012-11-28      10183
## 53 2012-11-29       7047
```

2. If you do not understand the difference between a histogram and a barplot, research the difference between them. Make a histogram of the total number of steps taken each day

```
hist(totalsteps$totalsteps, main= "Histogram of total steps per day", xlab= "Total steps per day", col="red", breaks = 10)
```



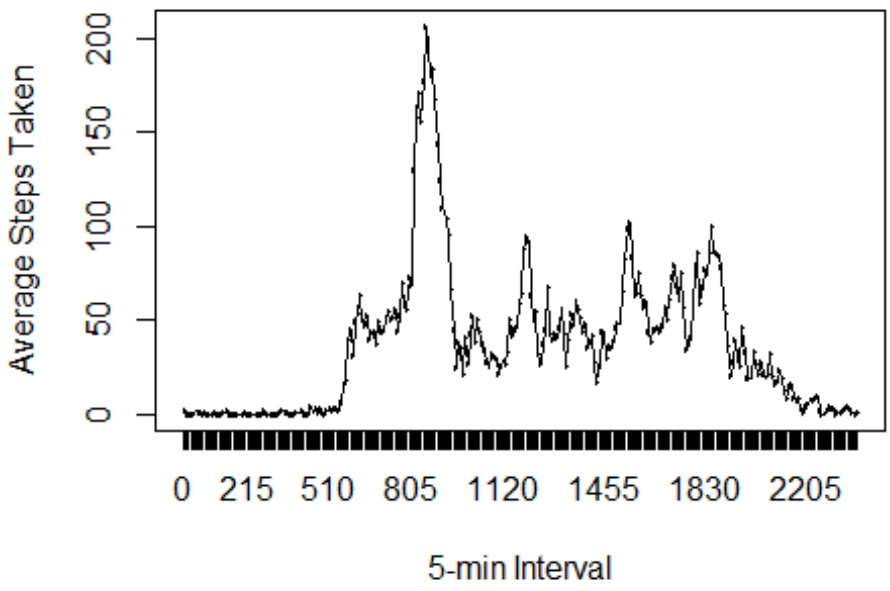
3.Calculate and report the mean and median of the total number of steps taken per day

```
mean(totalsteps$totalsteps)

## [1] 10766.19

median(totalsteps$totalsteps)

## [1] 10765
```

Question2: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)

```
meansteps<- aggregate(steps ~ interval, activity, mean, na.rm=TRUE)
plot(meansteps, type="l", xlab="5-min Interval", ylab="Average Steps Taken")
lines(meansteps)
```



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

```
meansteps$interval[meansteps$steps==max(meansteps$steps)]
```

```
## [1] 835
## 288 Levels: 0 5 10 15 20 25 30 35 40 45 50 55 100 105 110 115 120 ... 2355
```

Question3: Impute missing values

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

```
sum(is.na(activity$steps))
```

```
## [1] 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.

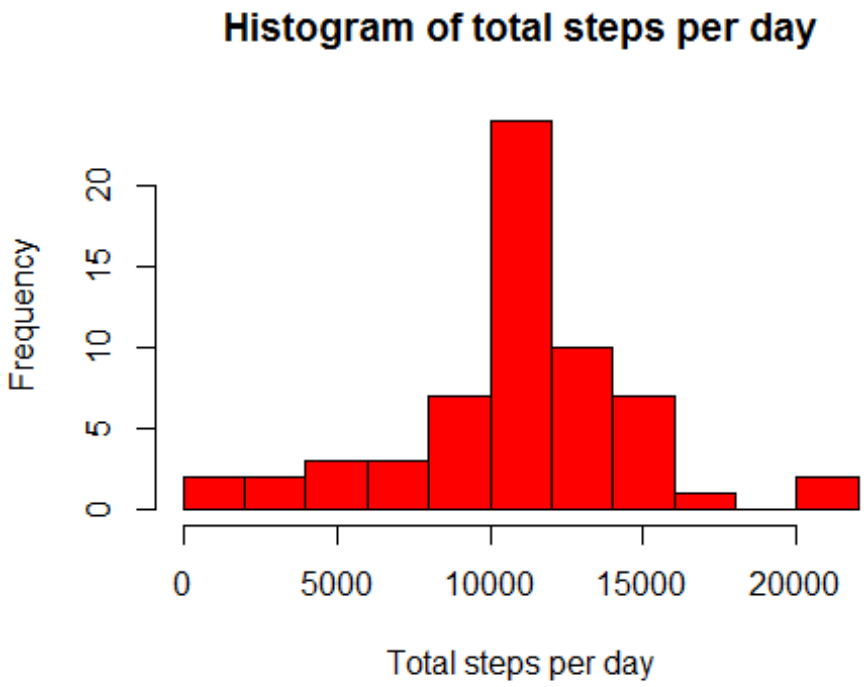
Answer: The missing values of the dataset could be imputed based on the mean steps of the particular day and time interval.

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

```
imputed_activity <- activity
imputed_activity$steps <- ifelse(is.na(activity$steps), mean(activity$steps,
na.rm = TRUE), activity$steps)
```

4.Make a histogram of the total number of steps taken each day

```
imputed_stepbyday <- tapply(imputed_activity$steps, imputed_activity$date,
sum)
hist(imputed_stepbyday, main= "Histogram of total steps per day", xlab=
"Total steps per day", col="red", breaks = 10)
```



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

```
mean(imputed_activity$steps)

## [1] 37.3826

median(imputed_activity$steps)

## [1] 0
```

4.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?

Answer: The frequency of the 2nd histogram is higher as compared to the 1st frequency for values of total steps per day. Imputation of missing data does not change the overall distribution and the value intervals of the total daily number of steps.

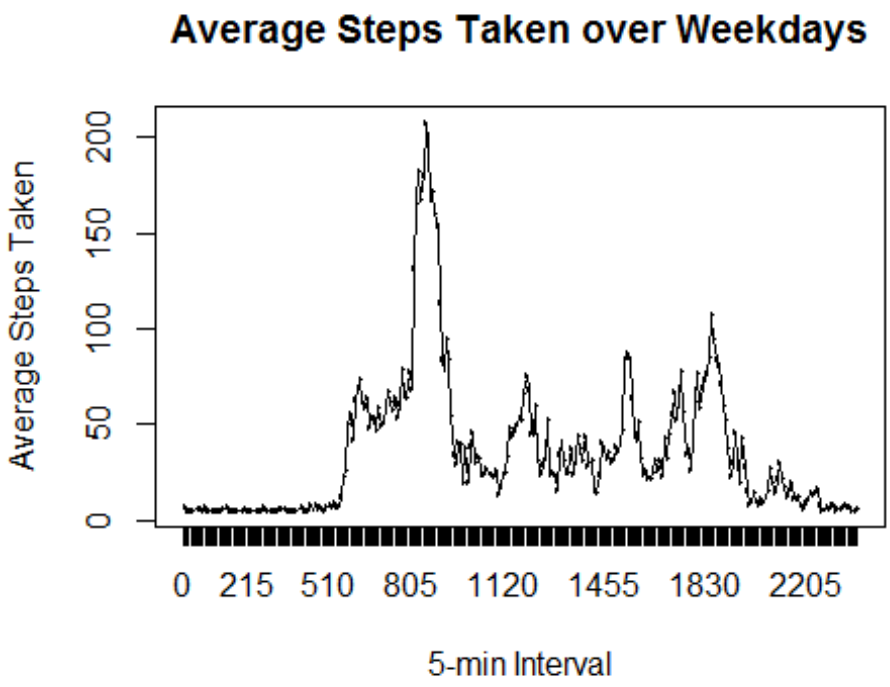
Are there differences in activity patterns between weekdays and weekends?

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.

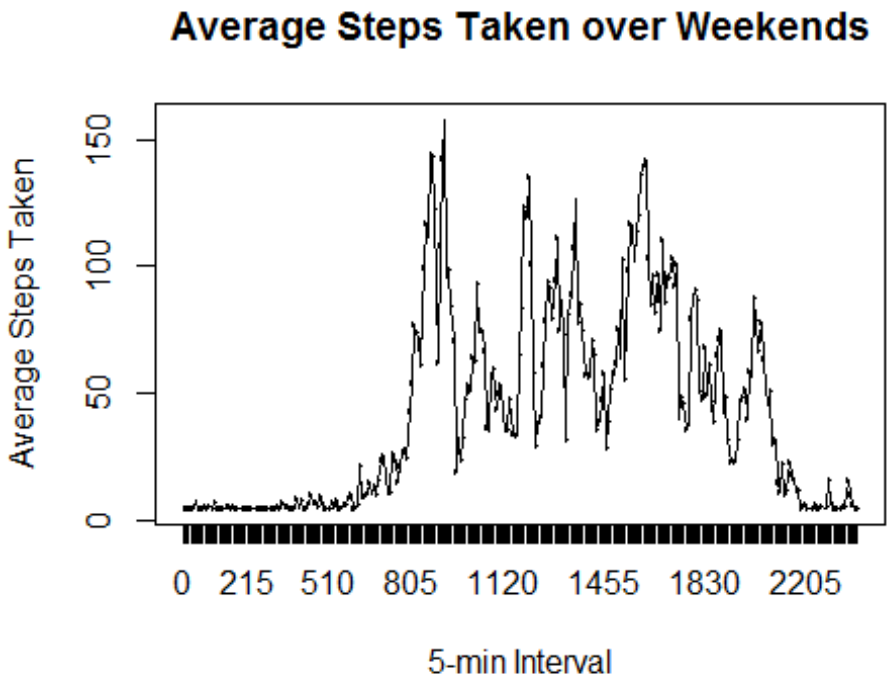
```
imputed_activity$dateType <-
ifelse(imputed_activity$weekday=="Monday"|imputed_activity$weekday=="Tuesday"
|imputed_activity$weekday=="Wednesday"|imputed_activity$weekday=="Thursday"|i
mputed_activity$weekday=="Friday","weekday","weekend")
```

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).

```
imputed_activity_weekday <- subset(imputed_activity, dateType=="weekday")
imputed_activity_weekend <- subset(imputed_activity, dateType=="weekend")

meansteps_weekday <- aggregate(steps ~ interval, imputed_activity_weekday,
mean, na.rm=TRUE)
meansteps_weekend <- aggregate(steps ~ interval, imputed_activity_weekend,
mean, na.rm=TRUE)

plot(meansteps_weekday, type="l", xlab="5-min Interval", ylab="Average Steps
Taken", main="Average Steps Taken over Weekdays")
lines(meansteps_weekday)
```




```
plot(meansteps_weekend, type="l", xlab="5-min Interval", ylab="Average Steps
Taken", main="Average Steps Taken over Weekends")
lines(meansteps_weekend)
```



Answer: The average steps taken over weekends are higher for the later part of the time intervals as compared to weekdays.