Reproducible Research: Peer Assessment 1

Loading and preprocessing the data

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

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.4.3
```

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

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

```
activity$date <- as.POSIXct(activity$date, "Asia/Singapore","%Y-%m-%d")
```

What is mean total number of steps taken per day?

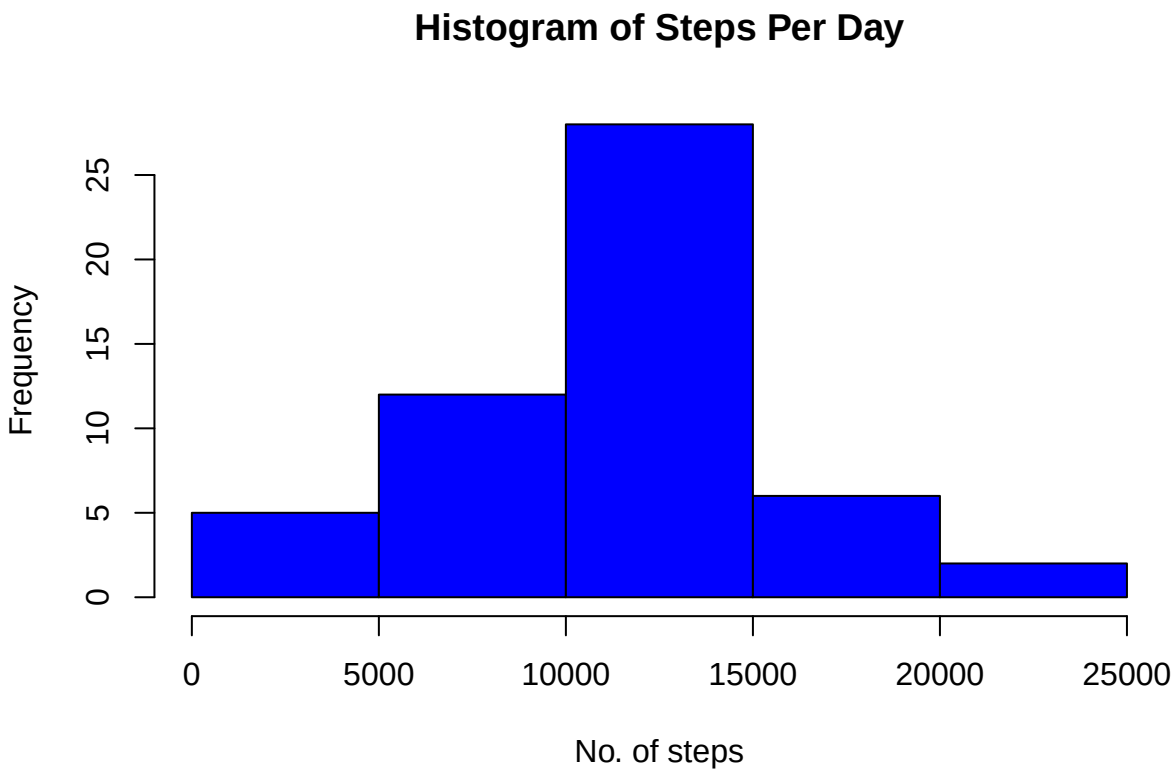
For this part of the assignment, you can ignore the missing values in the dataset.

1. Calculate the total number of steps taken per day

```
steps_perday <- aggregate(steps ~ date,data=activity, FUN=sum)
```

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

```
hist(steps_perday$steps, main = "Histogram of Steps Per Day", xlab = "No. of steps", col = "blue")
```



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

```
## Mean of the total number of steps taken per day
mean(steps_perday$steps)
```

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

```
## Median of the total number of steps taken per day
median(steps_perday$steps)
```

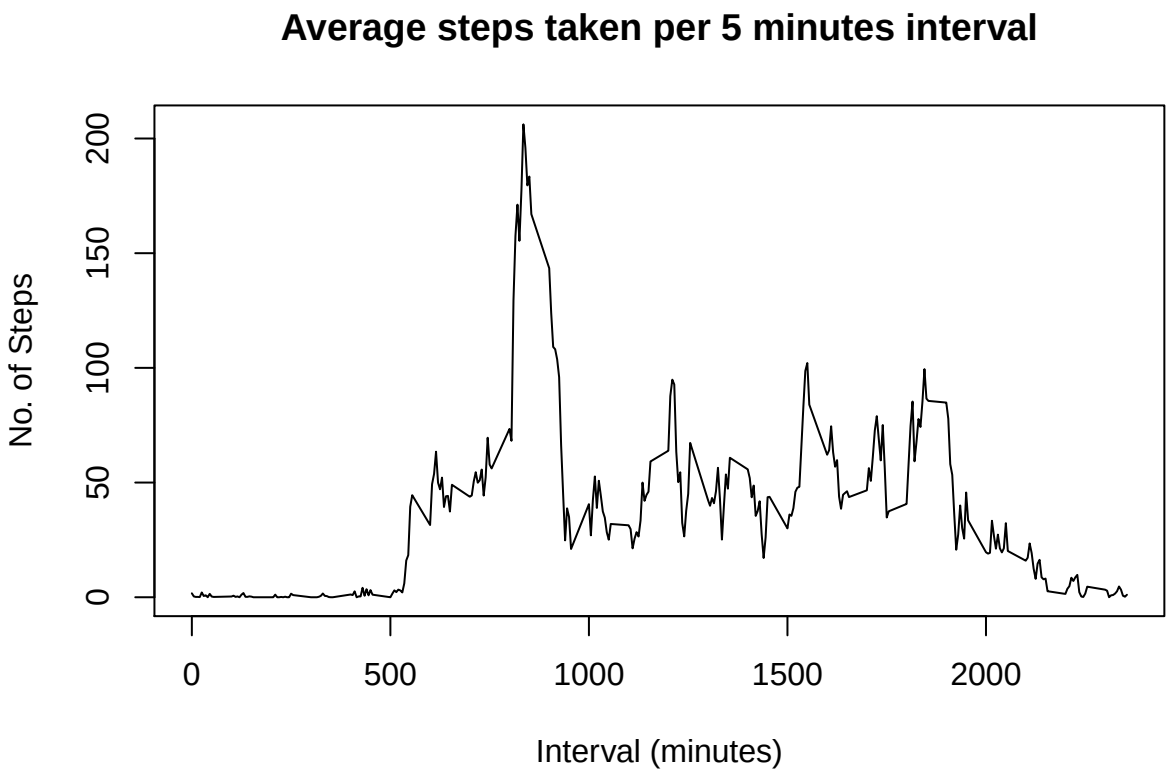
```
## [1] 10765
```

What is the average daily activity pattern?

```
avg_steps_int <- aggregate(steps~interval, activity, mean)
```

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)

```
plot(avg_steps_int$interval, avg_steps_int$steps, type = "l", main = "Average steps taken per 5 minutes
```



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

```
## Get the max value index and print interval value
cat("Interval ", avg_steps_int$interval[which.max(avg_steps_int$steps)], " with max average of", max(avg
```

```
## Interval  835  with max average of 206.1698  steps
```

Imputing missing values

Note that 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.

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

```
library(data.table)

activity <- as.data.table(activity)

nrow(activity[which(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.

```
avg_steps_interval <- setnames(data.table(unique(activity$date)),c("date"))
avg_steps_interval <- as.data.table(aggregate(steps ~ interval,na.rm = TRUE, data=activity, FUN=mean))
```

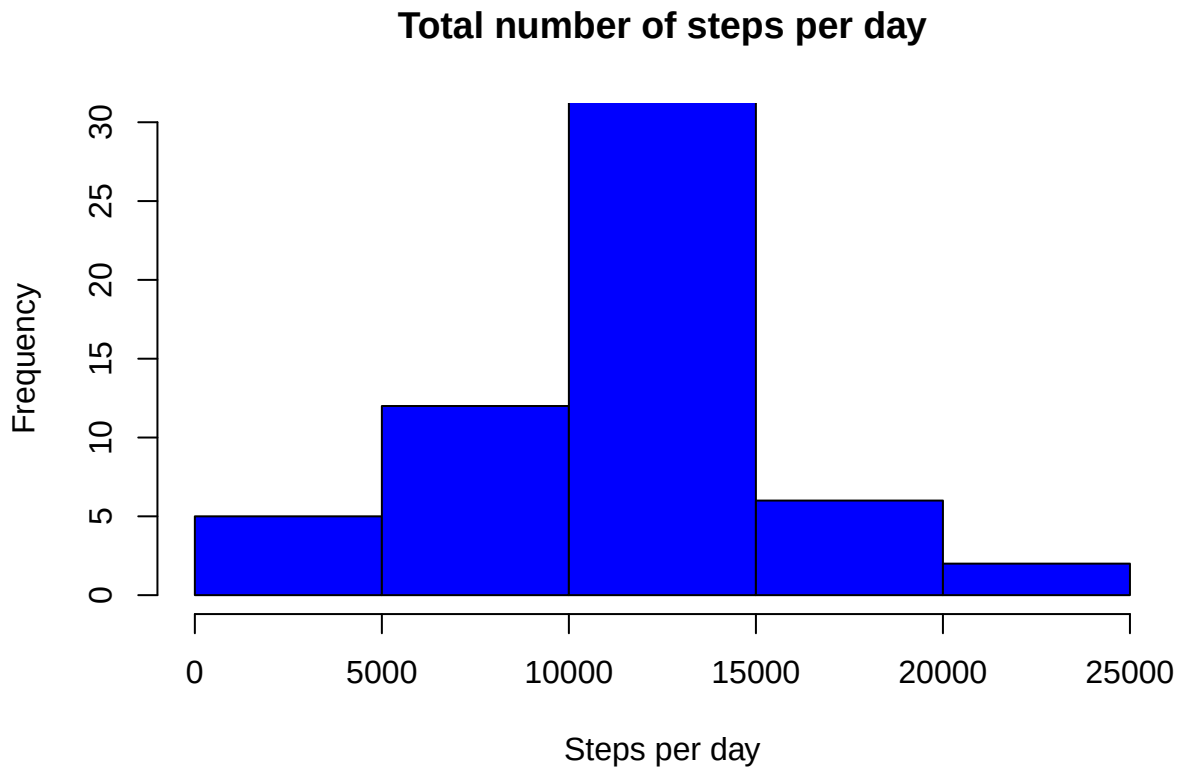
3. Create a new dataset that is equal to the original dataset but with the missing data filled in.

```
activity_clean <- transform(activity, steps = ifelse(is.na(activity$steps), yes = avg_steps_interval$st
```

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. 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?

```
steps_perday_clean <- as.data.table(aggregate(steps ~ date, data=activity_clean, FUN=sum))

hist(steps_perday_clean$steps, col = "blue", xlab = "Steps per day", ylim = c(0,30), main = "Total numb
```



Are there differences in activity patterns between weekdays and weekends?

For this part the weekdays() function may be of some help here. Use the dataset with the filled-in missing values for this part.

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.

```
library(chron)
```

```
## Warning: package 'chron' was built under R version 3.4.3
```

```
activity_clean[is.weekend(date),dayofweek:="Weekend"][!is.weekend(date),dayofweek:="Weekday"]
```

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). See the README file in the GitHub repository to see an example of what this plot should look like using simulated data.

```
activity_avg_clean <- aggregate(steps~interval + dayofweek, activity_clean, mean, na.rm = TRUE)
plot<- ggplot(activity_avg_clean, aes(x = interval , y = steps, color = dayofweek)) +
       geom_line() +
       labs(title = "Average Steps taken by Day of Week", x = "Interval", y = "Average No. steps") +
       facet_wrap(~dayofweek, ncol = 1, nrow=2)
print(plot)
```