Because I am not able to install the markdown document add-in for R Studio on my computer, I have created this word document that I hope will suffice.

Thank you for understanding.

Coursera Reproducible Research – Week 2 project 1

Susan Dunham

Loading the Activity Monitoring Data

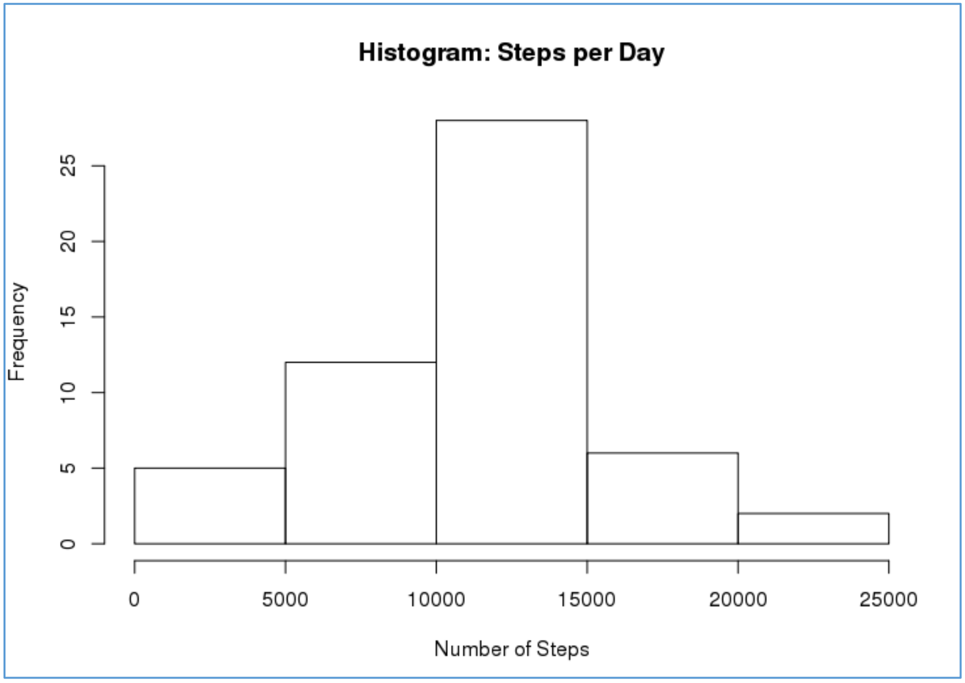
```
> mydata <- read.csv("./ReproducibleResearch/activity.csv", header = TRUE)
```

Q1: What is the mean total number of steps taken per day?
Calculate total number of steps taken per day:

```
> dailysteps <- tapply(mydata$steps, mydata$date, sum)
```

Histogram of total number of steps taken each day

```
> hist(dailysteps,xlab = "Number of Steps", main = "Histogram: Steps per Day")
```



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

```
> MeanDaily <- mean(dailysteps, na.rm = TRUE)
> MedianDaily <- median(dailysteps, na.rm = TRUE)
```

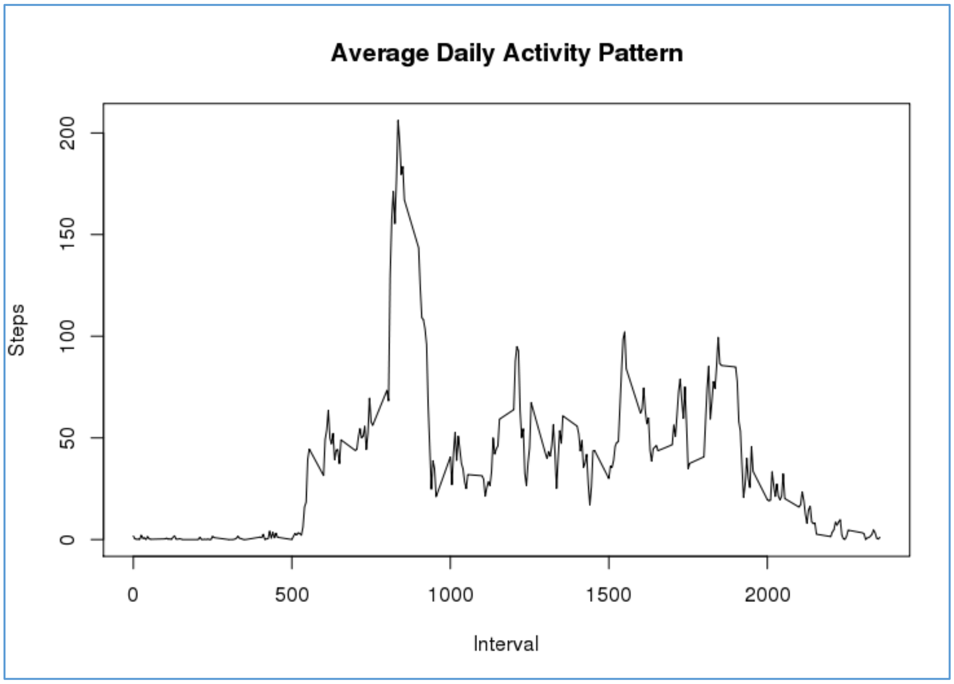
MeanDaily = 10766.1887

MedianDaily = 10765

Q2: What is the average daily activity pattern?

Create a time series plot of the 5-minute interval and the average number of steps taken, averaged across all days

```
> StepsperInterval <- tapply(mydata$steps, mydata$interval, mean, na.rm=TRUE)
> plot(as.numeric(names(StepsperInterval)),
+      StepsperInterval,
+      xlab = "Interval",
+      ylab = "Steps",
+      main = "Average Daily Activity Pattern",
+      type = "l")
```



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

```
> maxInterval <- names(sort(StepsperInterval, decreasing = TRUE)[1])
> maxSteps <- sort(StepsperInterval, decreasing = TRUE)[1]
```

The interval associated with maximum activity is interval 835, at 206 steps

Q3: Inputing missing values

Calculate and report the total number of missing values in the dataset

```
> TotNAs <- sum(is.na(mydata$steps))
```

There are 2304 missing values in the dataset

Strategy for filling in the missing values in the dataset.

I will be using the mean steps per interval for missing data values.

Create a new dataset that has the missing values filled in.

Splitting up the data by interval

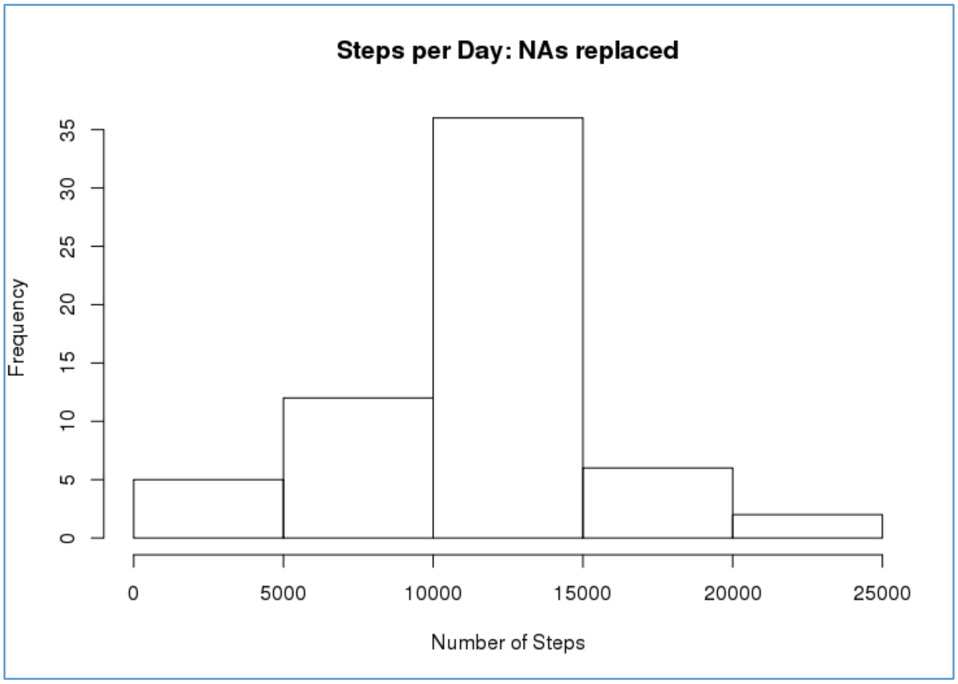
```
> activity_split <- split(mydata, mydata$interval)
```

Filling in the missing data in each interval

```
> for(i in 1:length(activity_split)){
+     activity_split[[i]]$steps[is.na(activity_split[[i]]$steps)] <- StepsperInterval[i]
+ }
> activity_added <- do.call("rbind",activity_split)
> activity_added <- activity_added[order(activity_added$date),]
```

Histogram of total number of steps taken each day.

```
> StepsPerDay_added <- tapply(activity_added$steps,activity_added$date,sum)
> hist(StepsPerDay_added, xlab ="Number of Steps", main ="Steps per Day: NAs replaced")
```



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

```
> MeanDaily_added <- mean(StepsPerDay_added, na.rm =TRUE)
> MedianDaily_added <- median(StepsPerDay_added, na.rm =TRUE)
```

The mean number of steps per day including replaced data is 10766.1887

The median value is 10766.1887

The mean value stayed the same, the median value increased.

Q4: Are there differences in activity patterns between weekdays and weekends?

Create a new factor variable in the dataset with two levels “weekday” and “weekend”

```
> activity_added$day <- ifelse(weekdays(as.Date(activity_added$date))=="Saturday" | weekdays(
as.Date(activity_added$date))=="Sunday", "weekend", "weekday")
```

Panel plot containing a time series plot of the 5-minute interval and the average number of steps taken, averaged across all weekday days and weekend days.

Calculates average steps per interval for weekend days

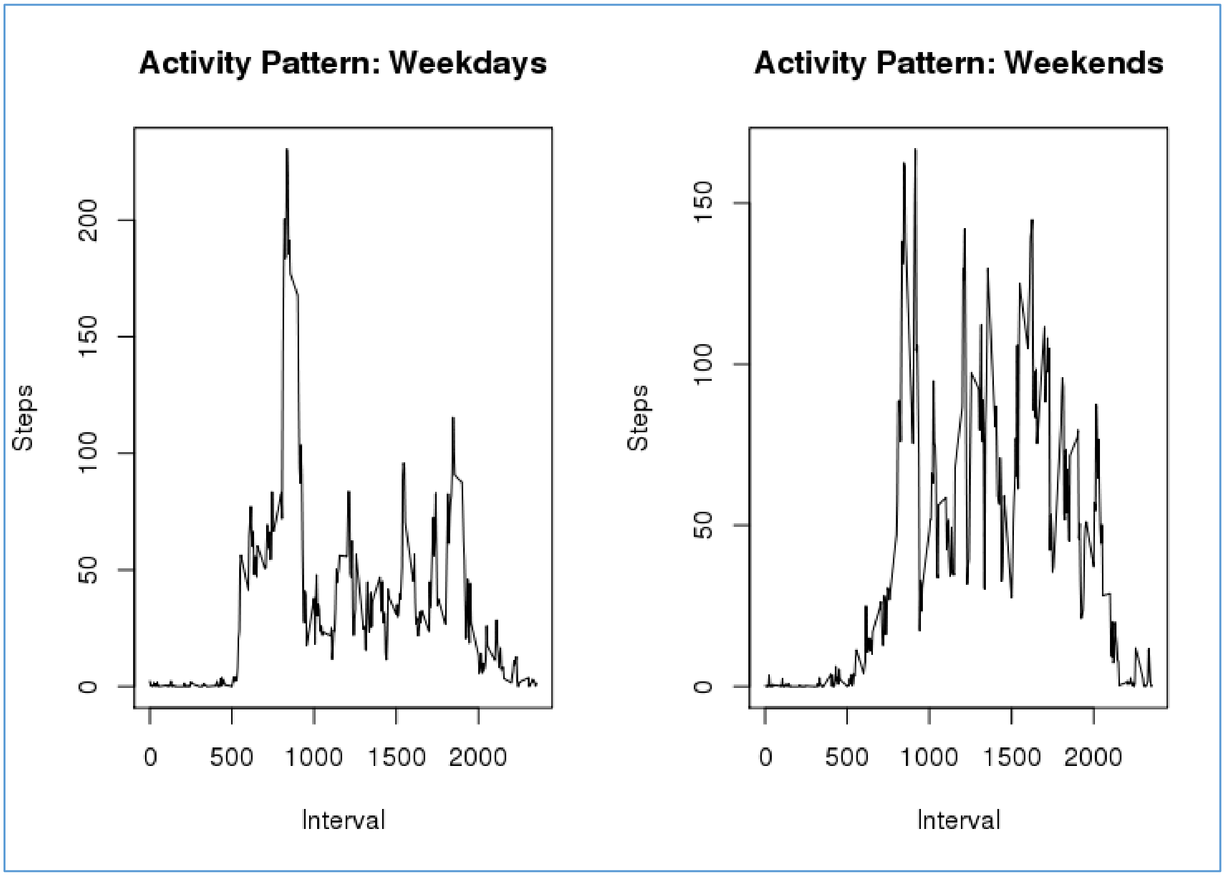
```
> StepsPerInterval_weekend <- tapply(activity_added[activity_added$day == "weekend",]$steps,
activity_added[activity_added$day == "weekend",]$interval,mean, na.rm =TRUE)
```

Calculate average steps per interval for weekday days

```
> StepsPerInterval_weekday <- tapply(activity_added[activity_added$day == "weekday",]$steps,
activity_added[activity_added$day == "weekday",]$interval, mean, na.rm =TRUE)
```

Creating a 2 panel plot

```
> par(mfrow=c(1,2))
> ##Plotting weekday activity
> plot(as.numeric(names(StepsPerInterval_weekday)),
+      StepsPerInterval_weekday,
+      xlab = "Interval",
+      ylab = "Steps",
+      main = "Activity Pattern: Weekdays",
+      type = "l")
> ##Plotting weekend activity
> plot(as.numeric(names(StepsPerInterval_weekend)),
+      StepsPerInterval_weekend,
+      xlab = "Interval",
+      ylab = "Steps",
+      main = "Activity Pattern: Weekends",
+      type = "l")
```


Activity Pattern: Weekdays
Steps
Interval
Activity Pattern: Weekends
Steps
Interval