Course project 1

Biswajit Chowdhury

20/06/2019

Loading the data

```
activityData <- read.csv("activity.csv")
head(activityData)
```

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

Calculate the total number of steps taken per day

```
stepPerDay<- aggregate(steps ~ date, activityData, sum)

# Make a histogram of the total number of steps taken each day
png("plot1.png", width=480, height=480)
hist(stepPerDay$steps, col="green", xlab = "Steps taken",
     main = "Total number of steps taken per day")
```

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

```
meansteps<- mean(stepPerDay$steps, na.rm = TRUE)
mediansteps<- median(stepPerDay$steps, na.rm = TRUE)
```

#What is the average daily activity pattern?

```
stepsperinterval<- aggregate(steps ~ interval, activityData, mean)

# Time series plot of the 5-minute interval and the average number of steps taken


png("plot2.png", width=480, height=480)
plot(stepsperinterval$interval, stepsperinterval$steps, type = "l",
     xlab = "Interval", ylab = "Steps taken",
     main = "Average Activity per day", col="red")
```

the maximum number of steps?

```
head(stepsperinterval)
```

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

```
maxsteps<- stepsperinterval[which.max(stepsperinterval$steps),1]
maxsteps
```

```
## [1] 835
```

Imputing missing values

Calculate the total number of missing values in the data set

```
totalNa<- sum(!complete.cases(activityData))
totalNa
```

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

Devise a strategy for filling in all of the missing values in the dataset. Here we fill the missing values with the mean steps taken per interval.

```
new_data <- transform(activityData, steps = ifelse(is.na(activityData$steps),

head(new_data)
```

```
##       steps       date interval
## 1 1.7169811 2012-10-01        0
## 2 0.3396226 2012-10-01        5
## 3 0.1320755 2012-10-01       10
## 4 0.1509434 2012-10-01       15
## 5 0.0754717 2012-10-01       20
## 6 2.0943396 2012-10-01       25
```

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.

```
# Recalculate the total steps

newStepsPerDay<- aggregate(steps ~ date, new_data, sum)

# making a histogram of the the above information


png("plot1.png", width=480, height=480)
hist(newStepsPerDay$steps, xlab="Steps",
     main="Difference between impute and non-impute data", col="red")

#Create Histogram to show difference.


hist(stepPerDay$steps, main="Difference between impute and non-impute data",
     xlab="Steps", col="green", add=T)
legend("topleft", c("Imputed", "Non-imputed"),
       col=c("red", "green"), lwd=10)


# Calculate new mean and median for imputed data.

newmeansteps <- mean(newStepsPerDay$steps)
newmediansteps <- median(newStepsPerDay$steps)

# Difference between new and old mean, median
diff_mean<- newmeansteps-meansteps
diff_mean
```

```
## [1] 0
```

```
diff_med <- newmediansteps-mediansteps

diff_med
```

```
## [1] 1.188679
```

```
#Calculate total difference.


total_diff <- sum(newStepsPerDay$steps) - sum(stepPerDay$steps)
total_diff
```

```
## [1] 86129.51
```

Are there differences in activity patterns between weekdays and weekends?

```
weekdays <- c("Monday", "Tuesday", "Wednesday", "Thursday",
              "Friday")
new_data$dow = as.factor(ifelse(is.element(weekdays(as.Date(new_data$date)),

                                           weekdays), "Weekday", "Weekend"))

newStepsPerInterval <- aggregate(steps ~ interval + dow, new_data, mean)

library(lattice)
png("plot4.png", width=480, height=480)
xyplot(newStepsPerInterval$steps ~ newStepsPerInterval$interval|newStepsPerInterval$dow,
       main="Average Steps per Day by Interval",xlab="Interval",
       ylab="Steps",layout=c(1,2), type="l", lwd=2)
```