Peer Assessment 1

Zachary Martin

4/22/2020

Introduction

This is an R Markdown document, created for the Coursera course "Reproducible Research" The data provided to be worked upon, is called "activity monitoring data".

Loading and preprocessing the data

The data must be in the user's current working directory for the code to run correctly.

```
activity <- read.csv("activity.csv", header = TRUE)
```

An initial look at the data confirms its dimensions and contents.

```
head(activity)
```

```
##   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
```

```
str(activity)
```

```
## 'data.frame':    17568 obs. of  3 variables:
##  $ steps   : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ date    : Factor w/ 61 levels "2012-10-01","2012-10-02",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ interval: int  0 5 10 15 20 25 30 35 40 45 ...
```

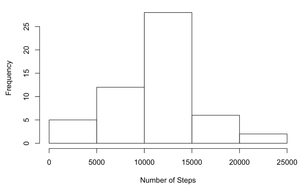
What is mean total number of steps taken per day?

The question states *any* missing values in the data set can be ignored. From using the summary functions previously, it is *already known* that there are NA values within the steps variable, so these can be removed now.

```
StepsPerDay <- tapply(activity$steps, activity$date, sum)
```

Histogram for the total number of steps taken per day

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

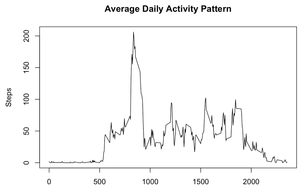


Mean and median for the total number of steps per day

```
MeanPerDay <- mean(StepsPerDay, na.rm = TRUE)
MedianPerDay <- median(StepsPerDay, na.rm = TRUE)
```

Average daily activity pattern

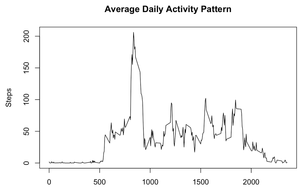
```
StepsPerInterval <- tapply(activity$steps, activity$interval, mean, na.rm = TRUE)
plot(as.numeric(names(StepsPerInterval)),
     StepsPerInterval,
     xlab = "Interval",
     ylab = "Steps",
     main = "Average Daily Activity Pattern",
     type = "l")
```



Therefore the mean value calculated is **10766.19**, and the median value **10765**.

What is the average daily activity pattern?

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



The base R plotting system is used to create a time series plot, with each interval on the x axis, and the average steps data on the y axis.

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

Imputing missing values

here are many 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.

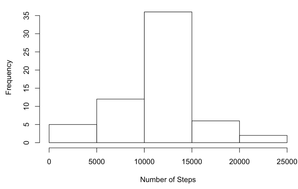
```
NA.vals <- sum(is.na(activity$steps))
```

New data set with missing values filled in.

```
StepsPerInterval <- tapply(activity$steps, activity$interval, mean, na.rm = TRUE)
# split activity data by interval
activity.split <- split(activity, activity$interval)
# fill in missing data for each interval
for(i in 1:length(activity.split)){
    activity.split[[i]]$steps[is.na(activity.split[[i]]$steps)] <- StepsPerInterval[i]
}
activity.imputed <- do.call("rbind", activity.split)
activity.imputed <- activity.imputed[order(activity.imputed$date) ,]
```

Now, using the filled data set, let's make a histogram of the total number of steps taken each day and calculate the mean and median total number of steps.

```
StepsPerDay.imputed <- tapply(activity.imputed$steps, activity.imputed$date, sum)
hist(StepsPerDay.imputed, xlab = "Number of Steps", main = "Histogram: Steps per Day (Imputed data)")
```



Mean and median values are higher after imputing missing data.

```
MeanPerDay.imputed <- mean(StepsPerDay.imputed, na.rm = TRUE)
MedianPerDay.imputed <- median(StepsPerDay.imputed, na.rm = TRUE)
```

Are there differences in activity patterns between weekdays and weekends

The question indicates that the imputed data set should be used to answer this problem.
To help in answering this question, firstly a new factor variable should be created within the data frame. This should indicate whether each day is a "weekday" or a "weekend".

To achieve this, I used the weekdays function to automatically calculate the day of the week each day resided upon, (Monday, Tuesday, etc.) Next, I wrote a for loop, which would assign the factor value "weekend" to all rows it read as having the values "Saturday" or "Sunday", and assign "weekday" to the others.

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

Calculate average steps per interval for weekends. Calculate average steps per interval for weekday. Plot weekday activity. Plot weekend activity.

Next, the average number of steps per interval is calculated, much like it has been done in previous questions.

```
# Calculate average steps per interval for weekends
StepsPerInterval.weekend <- tapply(activity.imputed[activity.imputed$day == "weekend" ,]$steps, activity.imputed[
activity.imputed$day == "weekend" ,]$interval, mean, na.rm = TRUE)

# Calculate average steps per interval for weekdays
StepsPerInterval.weekday <- tapply(activity.imputed[activity.imputed$day == "weekday" ,]$steps, activity.imputed[
activity.imputed$day == "weekday" ,]$interval, mean, na.rm = TRUE)

# Set a 2 panel plot
par(mfrow=c(1,2))

# Plot weekday activity
plot(as.numeric(names(StepsPerInterval.weekday)),
     StepsPerInterval.weekday,
     xlab = "Interval",
     ylab = "Steps",
     main = "Activity Pattern (Weekdays)",
     type = "l")

# Plot weekend activity
plot(as.numeric(names(StepsPerInterval.weekend)),
     StepsPerInterval.weekend,
     xlab = "Interval",
     ylab = "Steps",
     main = "Activity Pattern (Weekends)",
     type = "l")
```