Reproducible Research: Peer Assessment 1

Abstract

Introduction

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.

Data Processing and Analysis

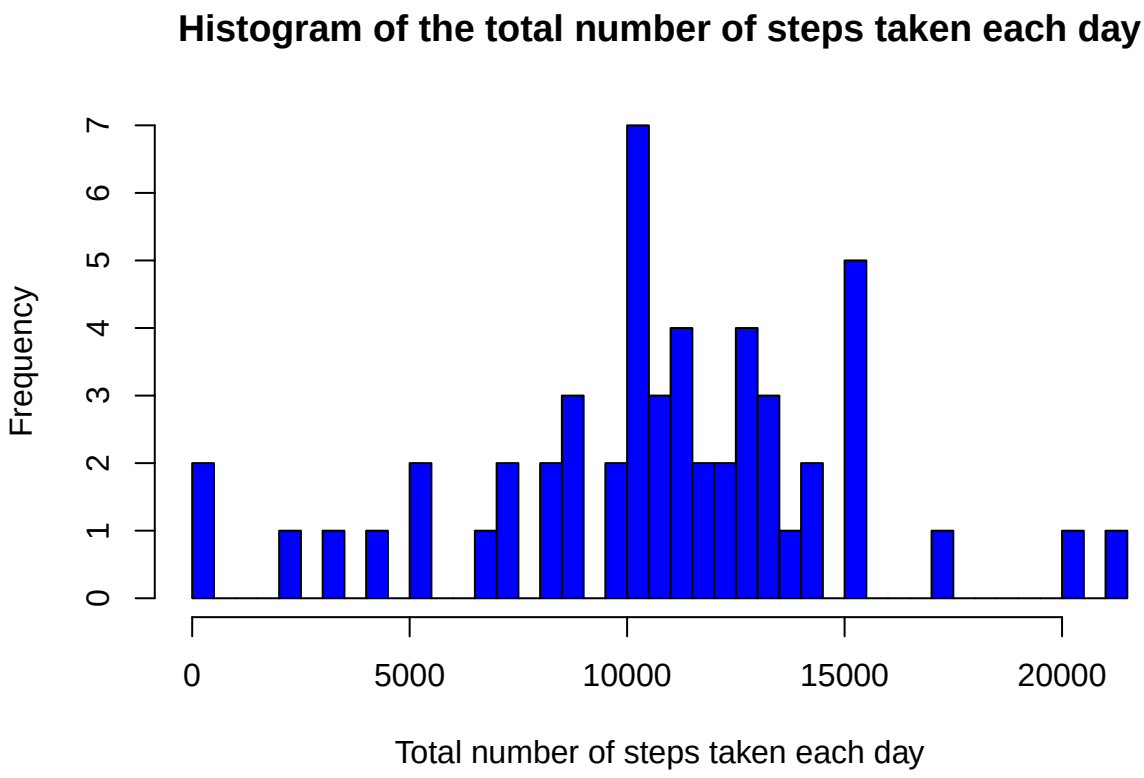
The source of the data used in this study is provided by this link [https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip]

```
        require(dplyr)
        require(ggplot2)
        require(data.table)
        require(scales)
        require(lubridate)
        activityData <- read.csv("activity.csv",
                                 header = TRUE,
                                 na.strings = "NA",
                                 stringsAsFactors=FALSE)

        activityData <- data.table(activityData)
        activityData$date <- as.Date(activityData$date, "%Y-%m-%d")
        dayOfWeek <- weekdays(activityData$date)
        activityData <- mutate(activityData, dayOfWeek)

        activityData <- activityData[which(complete.cases(activityData)),]
```

Summary of Data Analysis

The mean total number of steps taken per day

```
      sumStepsPerDay <- aggregate(activityData$steps, by = list(activityData$date), sum)
        hist(sumStepsPerDay$x, breaks = 50,
                col = "blue", border = NULL,
                main = "Histogram of the total number of steps taken each day",
                xlab = "Total number of steps taken each day")
```



The distribution of the total number of steps taken per day appear to follow normal behaviour, where the mean of the total number of steps taken per day is

```
mean(sumStepsPerDay$x, na.rm = TRUE)
```

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

and median number of steps taken each day is

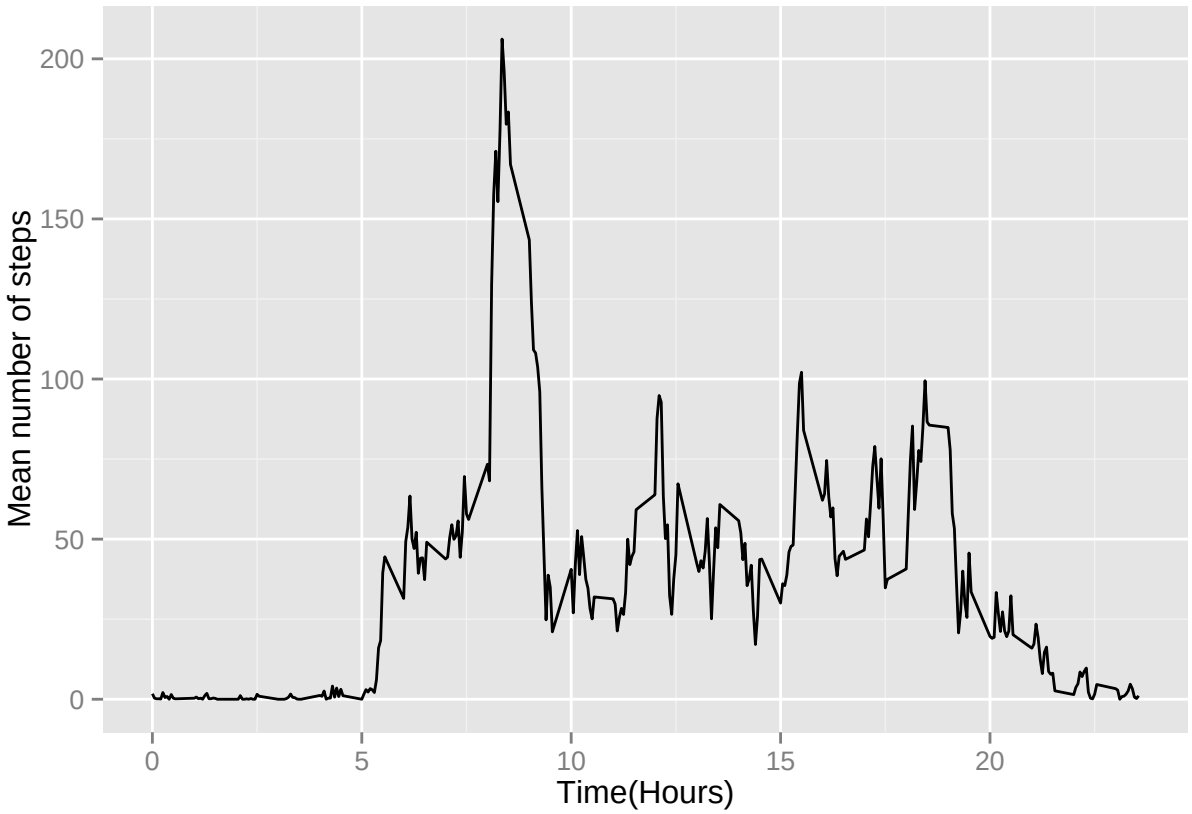
```
median(sumStepsPerDay$x, na.rm = TRUE)
```

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

The average daily activity pattern

In order to depict the average daily activity pattern, a time series graph of the average steps through the time-span of a day, that is, the mean number of steps for each of the 5 minute time intervals over the 24 hour period, is presented. Here the summary variable is the number of steps and the grouping variable is the interval.

```
activityTimeSeries <- aggregate(activityData$steps, by = list(activityData$interval),
                    mean)
colnames(activityTimeSeries) <- c("Time", "MeanNumberOfSteps")
activityTimeSeries$Time <- activityTimeSeries$Time/100 # to convert the time interval to hours
ggplot(activityTimeSeries, aes(x=Time, y=MeanNumberOfSteps)) + geom_line() + xlab("Time(Hours)"
```



The 5-minute interval, on average across all the days in the dataset, which contains the maximum number of steps is at 8.35 in the morning, as given by

```
        print(activityTimeSeries[which.max(activityTimeSeries$MeanNumberOfSteps),])
```

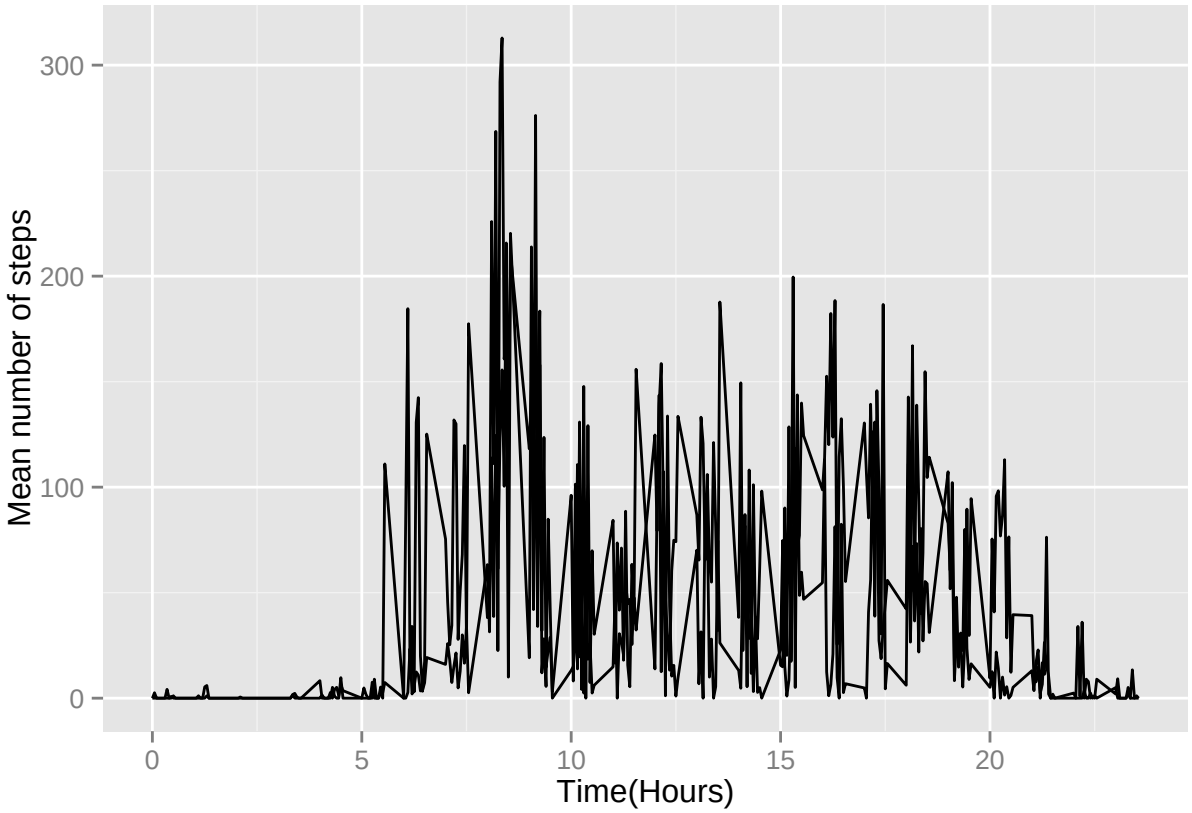
```
##     Time MeanNumberOfSteps
## 104 8.35          206.1698
```

The difference in activity patterns between weekdays and weekends

```
        weekend <- c("Sunday","Saturday")
        activityDTWE <- activityData[which(dayOfWeek == weekend),]
        activityTimeSeriesWE <- aggregate(activityDTWE$steps, by = list(activityDTWE$interval),
                        mean)
        colnames(activityTimeSeriesWE) <- c("Time", "MeanNumberOfStepsWE")
        activityTimeSeriesWE$Time <- activityTimeSeriesWE$Time/100 # to convert the time interval to ho

        weekday <- c("Monday","Tuesday", "Wednesday","Thursday","Friday")
        activityDTWD <- activityData[which(dayOfWeek == weekday),]
```

```
## Warning in dayOfWeek == weekday: longer object length is not a multiple of
## shorter object length
```

```
activityTimeSeriesWD <- aggregate(activityDTWD$steps, by = list(activityDTWD$interval),
                  mean)
colnames(activityTimeSeriesWD) <- c("Time", "MeanNumberOfStepsWD")
activityTimeSeriesWD$Time <- activityTimeSeriesWD$Time/100 # to convert the time interval to ho

activityTimeSeries <- merge(activityTimeSeriesWD, activityTimeSeriesWE, "Time")

ggplot(activityTimeSeries, aes(x=Time, y=MeanNumberOfStepsWD)) + geom_line() + xlab("Time(Hours)
```



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

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.

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

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?

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.

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.