Reproducible Research - Project 1

NK

Tuesday, May 12, 2015

The following document is a code for Project 1 of reproducible research. It describes an exploratory data analysis on a step counter data.

First set global options to always display code chunks:

```
library(knitr)
```

```
## Warning: package 'knitr' was built under R version 3.1.3
```

```
opts_chunk$set(echo=TRUE)
```

Loading and Preprocessing the Data

First, set the working directory

```
#Set working directory
setwd("C:\\Users\\N\\Dropbox\\Coursera\\05 - Reproducible Research\\RepData_PeerAssessment1")
```

Read accelerometer data that is located in the same folder

```
#read in data
dat=read.csv(file="activity.csv",
             header=TRUE, sep=",",
             stringsAsFactors=FALSE,
             na.strings="NA")
dat$date=as.factor(dat$date)
```

What is the average dailty activity pattern?

Calculate the total number of steps taken per day

First extract the step data and the sum the steps via the aggregate function.

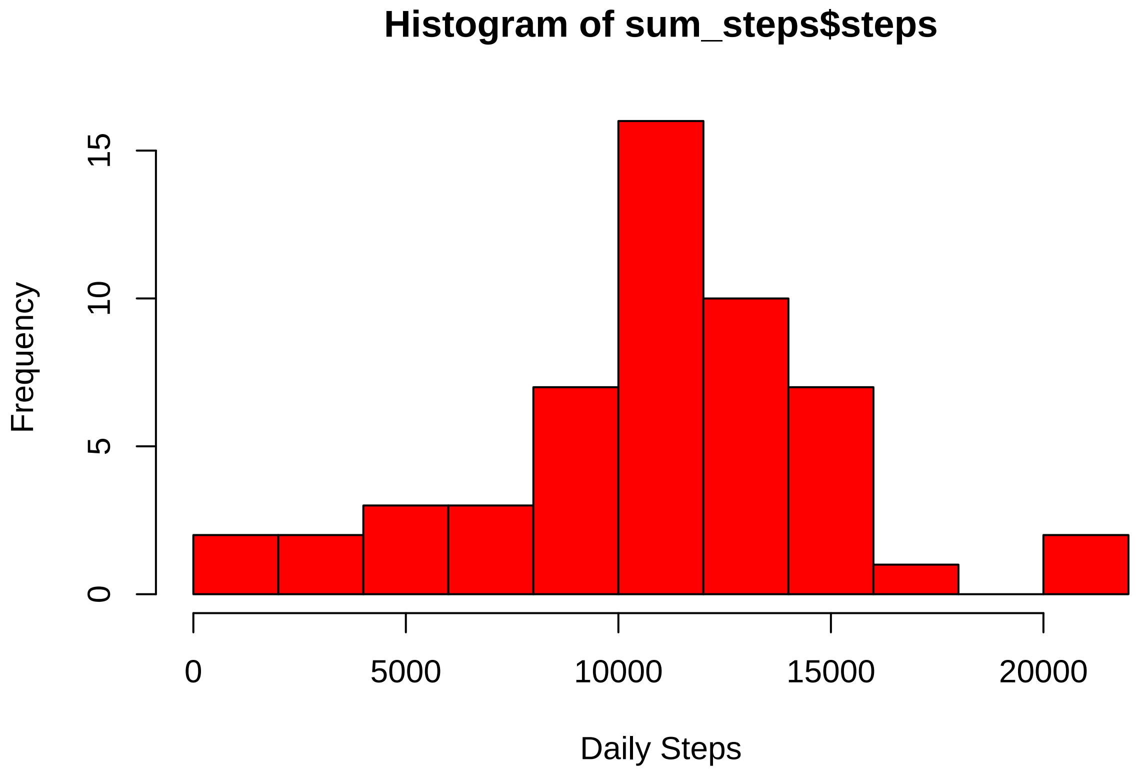
```
#Take the step data and remove the NA values
mean_step_data=dat[!is.na(dat$steps),1:3]
#Calculate the sum
sum_steps=aggregate(mean_step_data[1],
                    by=list(mean_step_data$date),
                    FUN=sum)
sum_steps
```

```
##       Group.1 steps
## 1  2012-10-02   126
## 2  2012-10-03 11352
## 3  2012-10-04 12116
## 4  2012-10-05 13294
## 5  2012-10-06 15420
## 6  2012-10-07 11015
## 7  2012-10-09 12811
## 8  2012-10-10  9900
## 9  2012-10-11 10304
## 10 2012-10-12 17382
## 11 2012-10-13 12426
## 12 2012-10-14 15098
## 13 2012-10-15 10139
## 14 2012-10-16 15084
## 15 2012-10-17 13452
## 16 2012-10-18 10056
## 17 2012-10-19 11829
## 18 2012-10-20 10395
## 19 2012-10-21  8821
## 20 2012-10-22 13460
## 21 2012-10-23  8918
## 22 2012-10-24  8355
## 23 2012-10-25  2492
## 24 2012-10-26  6778
## 25 2012-10-27 10119
## 26 2012-10-28 11458
## 27 2012-10-29  5018
## 28 2012-10-30  9819
## 29 2012-10-31 15414
## 30 2012-11-02 10600
## 31 2012-11-03 10571
## 32 2012-11-05 10439
## 33 2012-11-06  8334
## 34 2012-11-07 12883
## 35 2012-11-08  3219
## 36 2012-11-11 12608
## 37 2012-11-12 10765
## 38 2012-11-13  7336
## 39 2012-11-15    41
## 40 2012-11-16  5441
## 41 2012-11-17 14339
## 42 2012-11-18 15110
## 43 2012-11-19  8841
## 44 2012-11-20  4472
## 45 2012-11-21 12787
## 46 2012-11-22 20427
## 47 2012-11-23 21194
## 48 2012-11-24 14478
## 49 2012-11-25 11834
## 50 2012-11-26 11162
## 51 2012-11-27 13646
## 52 2012-11-28 10183
## 53 2012-11-29  7047
```

Create a histogram of the step summaries

Create and show a histogram plot:

```r
#Make historgram
hist(sum_steps$steps,
     breaks=10,
     xlab="Daily Steps",
     col="red")
```



Find the mean and median

The median of the number of steps is:

```r
median(sum_steps$steps)
```

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

The mean number of steps is:

```r
mean(sum_steps$steps)
```

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

What is the average daily activity pattern?

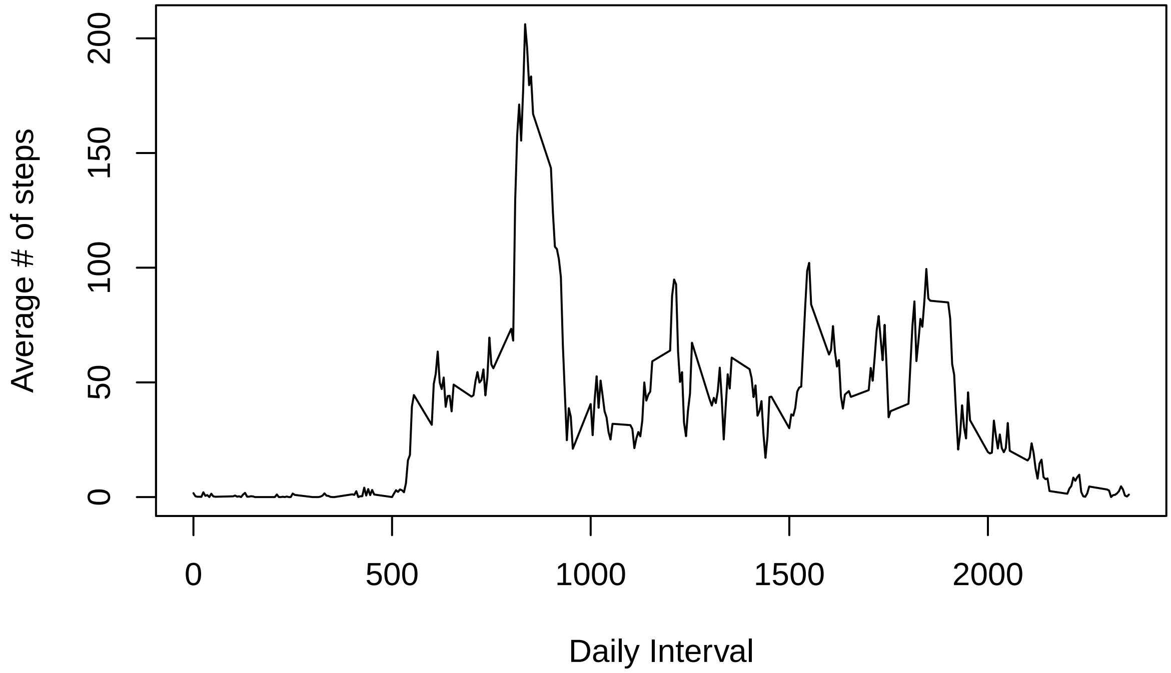
Make a time series plot:

Find the average interval number of steps:

```
#interval data mean
id=aggregate(mean_step_data[1],by=list(mean_step_data$interval),FUN=mean)
```

Plot the mean interval data:

```
plot(unique(mean_step_data$interval),
     id$steps,pch=16,
     type="l",
     xlab="Daily Interval",
     ylab="Average # of steps"
     )
```



Find the Interval with maximum steps

The values are stored in the id variable. Find the maximum index:

```
which.max(id$steps)
```

```
## [1] 104
```

with a maximum value of

```r
max(id$steps)
```

```
## [1] 206.1698
```

Imputing missing Values

Find total missing entries

First create a vector full of missing steps entries and then report its length

```r
#create empty vector indicator
na_vec=is.na(dat$steps)
#find total entires that are missing
sum(na_vec)
```

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

Fill missing entries

There are two strategies to fully get rid of NA values: 1. First replace them by the mean of the day

The code first appends a column full of means. The code then replaces the missing values with those means.

```r
#append means
dat$means<-ave(dat$steps,dat$date,rm.na=TRUE)
#remove NAs
dat[na_vec,"steps"]=dat[na_vec,"means"]
```

Assign to new Tidy Data set

label the tidy data set as tdat

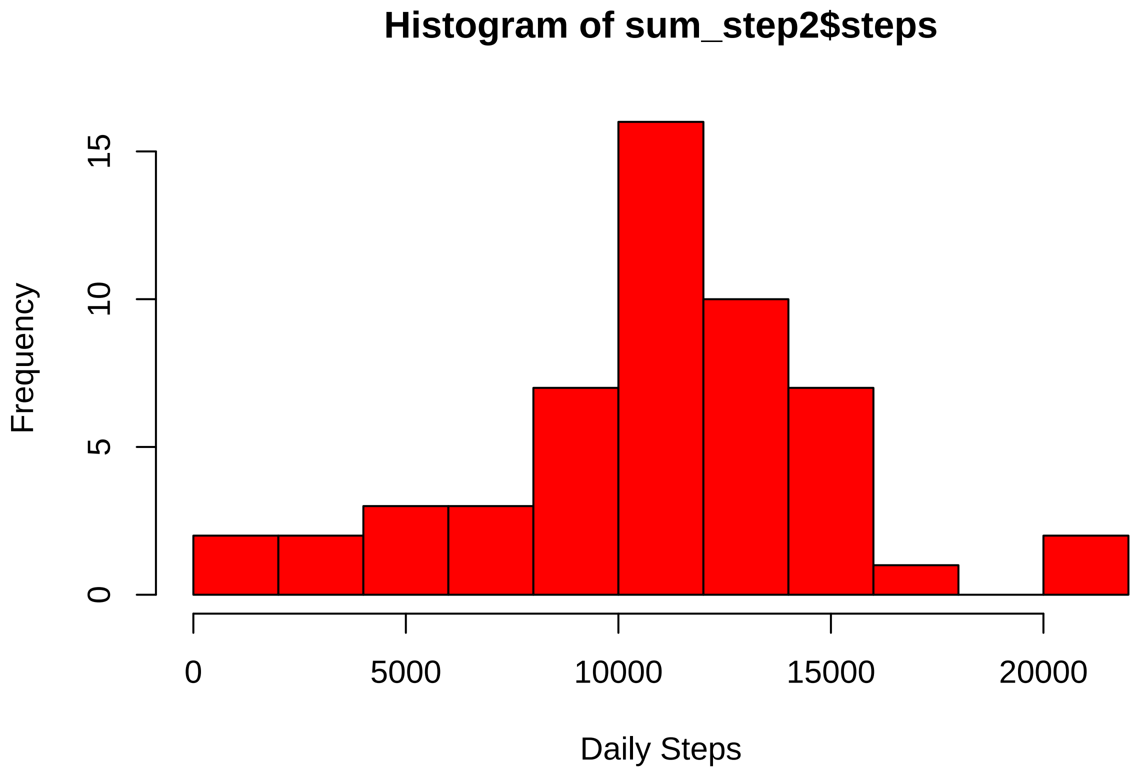
```r
#find values that are still NA
still_na=is.na(dat$means)
#Assign tidy data set
tdat=dat[!still_na,]
#verify that there are no empties:
sum(is.na(tdat))
```

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

Make a histogram of the total number of steps

Calculate sum and draw histogram:

```r
#Calculate the sum
sum_step2=aggregate(tdat[1],
                    by=list(tdat$date),
                    FUN=sum)
#Make histogram
hist(sum_step2$steps,
     breaks=10,
     xlab="Daily Steps",
     col="red")
```



The median of the tidy number of steps is:

```
median(sum_step2$steps)
```

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

The mean tudy number of steps is:

```
mean(sum_step2$steps)
```

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

The mean and median has not changed by replacing NA values with the mean step values.

Difference in Activity Patterns between Weekends and Weekdays

Separate data into wekedays

The following code first appends a weekdays variable. Then the code appends a variable that states whether the day is a weekend or a weekday.

```
#change factors to characters
tdat$date<-lapply(tdat[,"date"],as.character)
#change char to date
library("lubridate")
```

```
## Warning: package 'lubridate' was built under R version 3.1.3
```

```
tdat$date<-ymd(tdat$date)
tdat$weekdays<-weekdays(tdat$date)
tdat$isWeekend<-tdat$weekdays %in% c('Saturday','Sunday')
```

Determine patterns in Intervals and Weekdays

First prepare the interval data (id) to properly rename the variables and calculate the means

```
#Calculate means and rename variables and prepare values for plotting
id=aggregate(tdat[,1],by=list(tdat$interval,tdat$isWeekend),FUN=mean)
#rename variables
library(plyr)
```

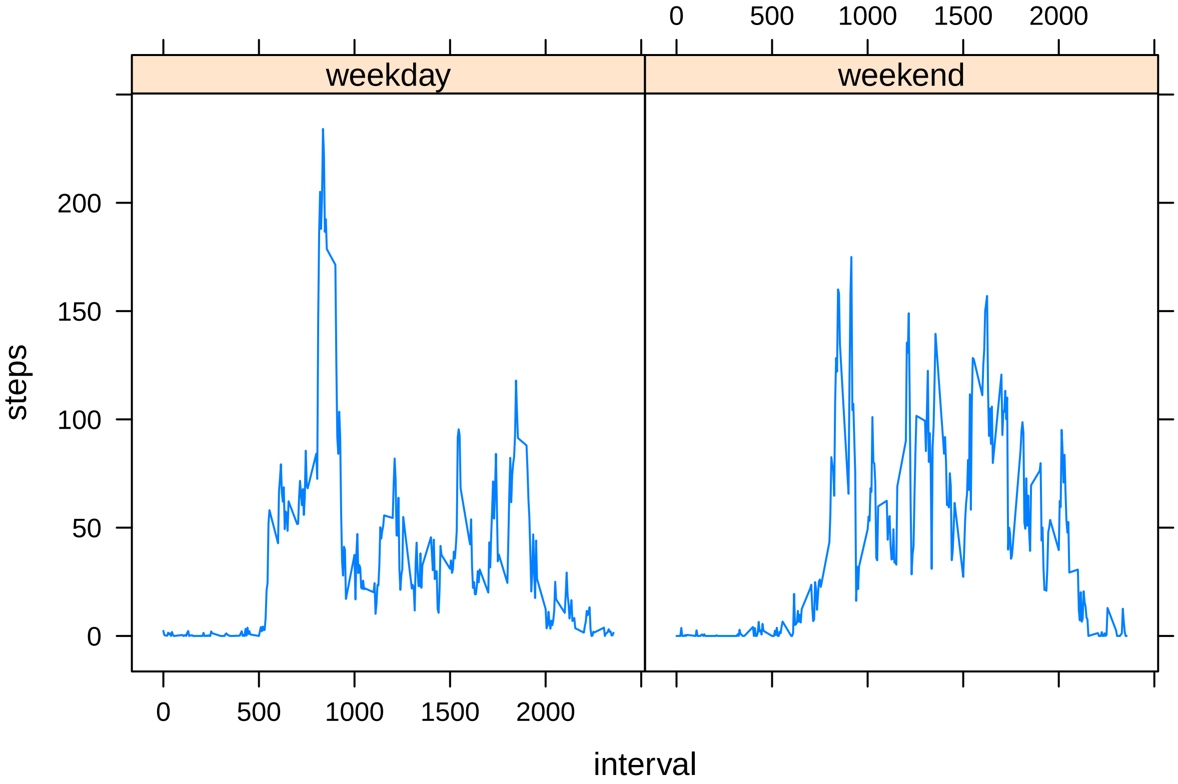
```
## Warning: package 'plyr' was built under R version 3.1.3
```

```
##
## Attaching package: 'plyr'
##
## The following object is masked from 'package:lubridate':
##
##     here
```

```
id=rename(id,c("Group.1"="interval", "Group.2"="isWeekend","x"="steps"))
id$isWeekend<-as.factor(ifelse(id$isWeekend, "weekend", "weekday"))
```

Now run some plotting code to reproduce the graph usign the lattice plot function:

```
#Plot
library(lattice)
xyplot(steps~interval| isWeekend, data=id,type="l")
```



There is a clear difference in the shapes of the weekeday steps and the weekend step patterns. There are more steps in the weekend and the peak of the weekdays occurs a early in the interval.