Reproducible Research: Peer Assessment 1

J Gross

2016-02-21

Loading and preprocessing the data

The raw data is fetched, as needed, from this URL and cached on the local filesystem of the development laptop. Subsequently the string representations of the observed dates that are contained in the data are converted to Date format:

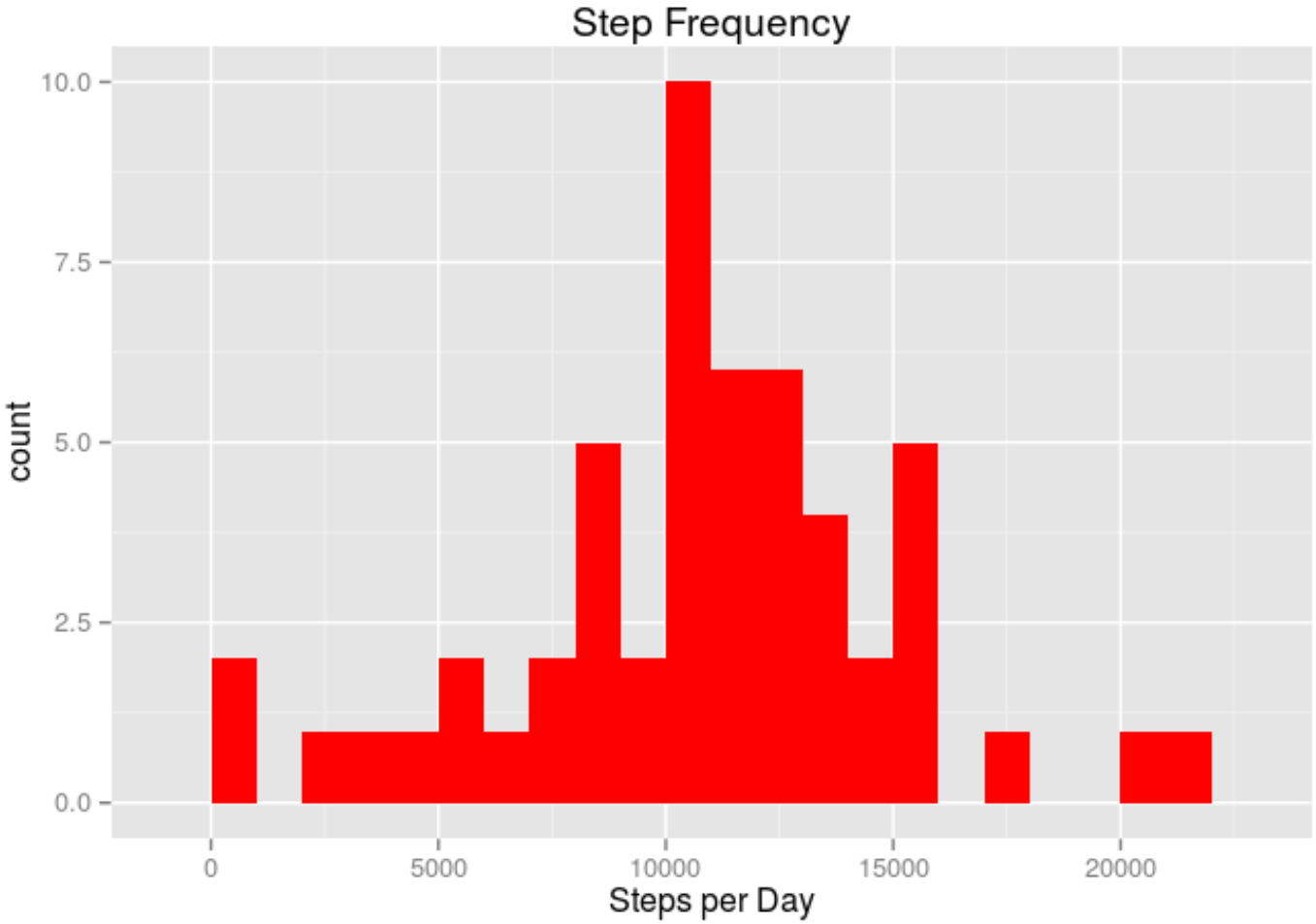
```
setwd("~/RepData_PeerAssessment1")
if ( !file.exists("activity.zip") ) {
   download.file("https://github.com/jef144/RepData_PeerAssessment1\activity.zip",
destfile="activity.zip", method="auto")
 }
activityDat <-  read.csv(unz(  "~/RepData_PeerAssessment1/activity.zip", "activit
y.csv" ) )

#Convert to date format
activityDat$date <- as.Date(activityDat$date, "%Y-%m-%d")
str(activityDat)
```

```
## 'data.frame':    17568 obs. of  3 variables:
##  $ steps   : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ date    : Date, format: "2012-10-01" "2012-10-01" ...
##  $ interval: int  0 5 10 15 20 25 30 35 40 45 ...
```

What is mean total number of steps taken per day?

To answer this question we will sum the steps per each of the days and then generate a histogram.

```
library(ggplot2)
aggDay <-aggregate(steps ~ date, data=activityDat, sum)
histSum <- qplot(aggDay$steps, geom="histogram", na.rm=TRUE, binwidth=1000, xla
b="Steps per Day", main="Step Frequency", fill=I("red"))
histSum
```

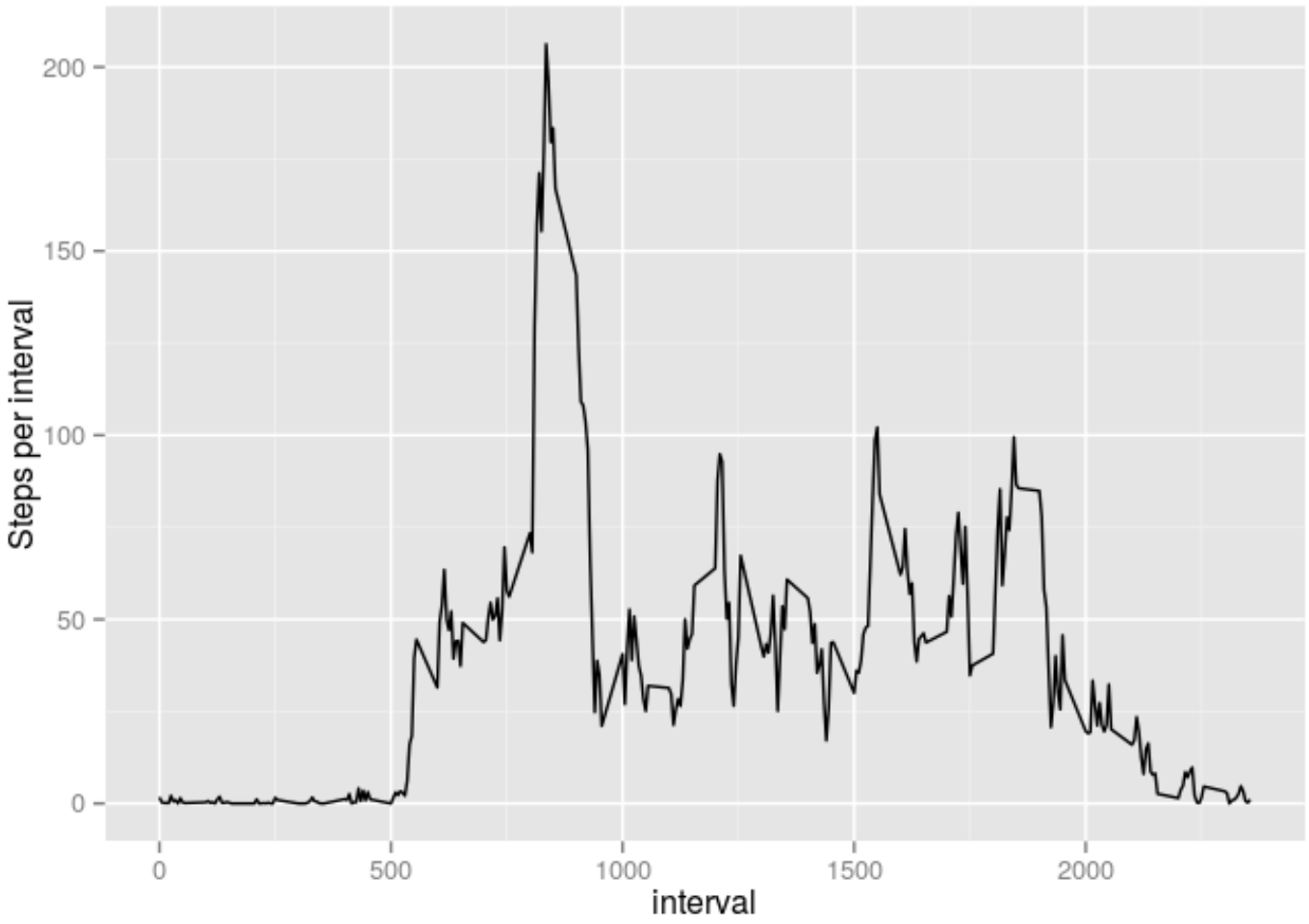


We can see that the distribution of steps per day approximates a normal distribution. The mean steps per day is 10766 and the median value is 10765. Both of these values are calculated by ignoring the missing observations.

What is the average daiy activity pattern?

This question is best answered by a buiding an average for each interval of the day, using the avaiable data:

```
library(ggplot2)
aggInt <-aggregate(steps ~ interval, data=activityDat, mean, na.rm=TRUE)
plotInt<- ggplot(data=aggInt, aes(x=interval, y=steps)) +
  geom_line() +
  ylab("Steps per interval")

plotInt
```



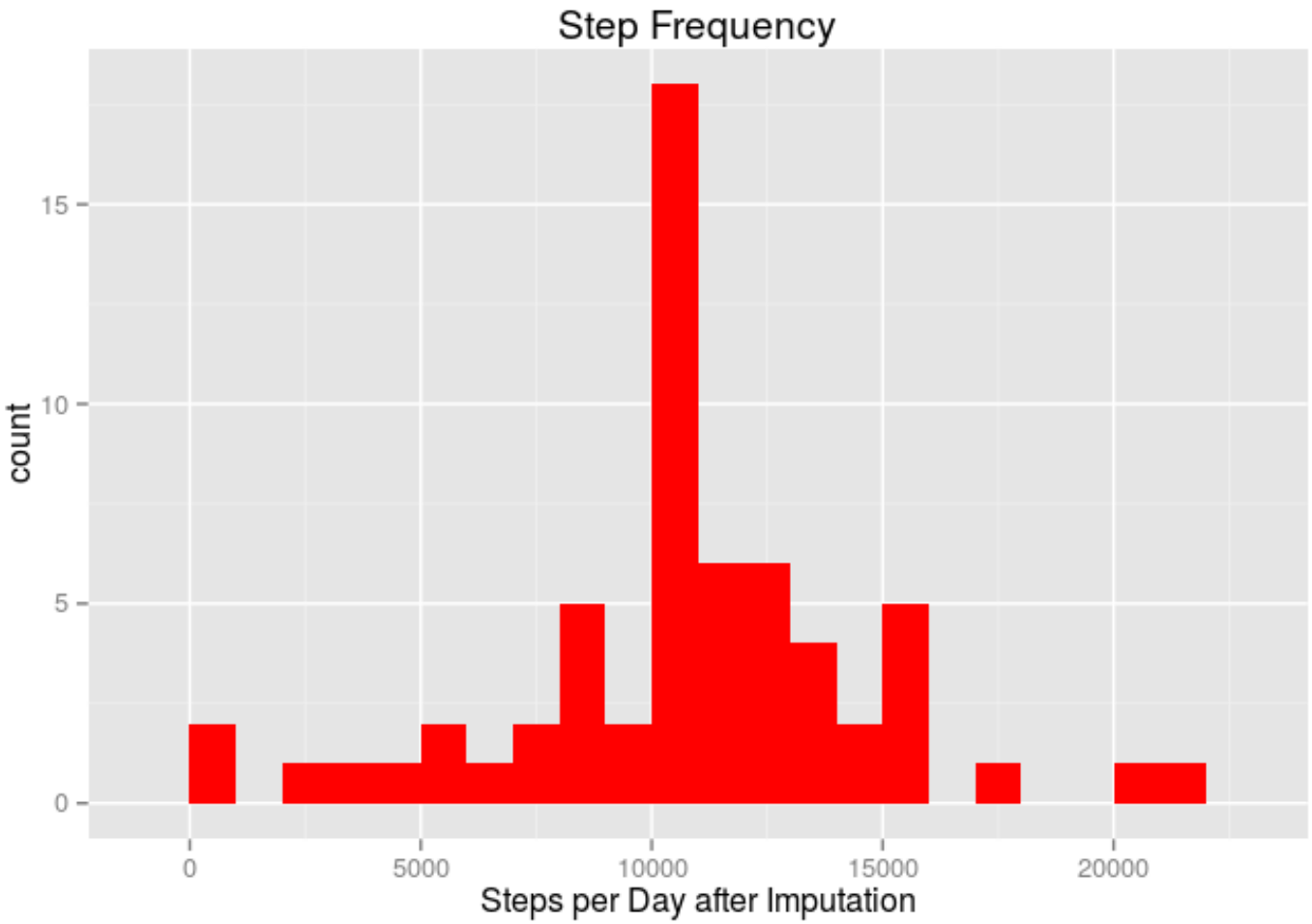

```
#Interval with the highest average count. Then sort the aggreate to get theVi high
est interval on top
library(ggplot2)
aggInt <-aggregate(steps ~ interval, data=activityDat, mean)
topaggInt <- aggInt[order(aggInt$steps, decreasing =TRUE), ][1,]
```

The resulting plot shows several peaks during the day, notably around interval 800. Specifically, the interval with the highest number of steps is interval 835 .

Imputing missing Values

The strategy used will be to calculate the average number of steps for all intervals, across all the days. Each missing value will then be substituted using the corresponding averate for that interval. There are 2304 missing values.

```
#Then impute by substituting the average values, across all days,  into the corres
ponding missing interval numbers
aggInt <-aggregate(steps ~ interval, data=activityDat, mean)
imputedDat <- merge( activityDat, aggInt, by="interval")
#Flag the observations that have missing values. Note that the name of the column
has changed as a result of the merge operatoin
rowsNA <- is.na(imputedDat$steps.x)
#Copy over the averages as imputed values, replacing the values of steps.x
imputedDat[rowsNA,2]  = imputedDat[rowsNA,4]
aggImputedDay <-aggregate(steps.x ~ date, data=imputedDat, sum)


#7 total steps per day after imputatiojs
histImputed <- qplot(aggImputedDay$steps.x, geom="histogram",  binwidth=1000, xla
b="Steps per Day after Imputation", main="Step Frequency", fill=I("red"))
histImputed
```

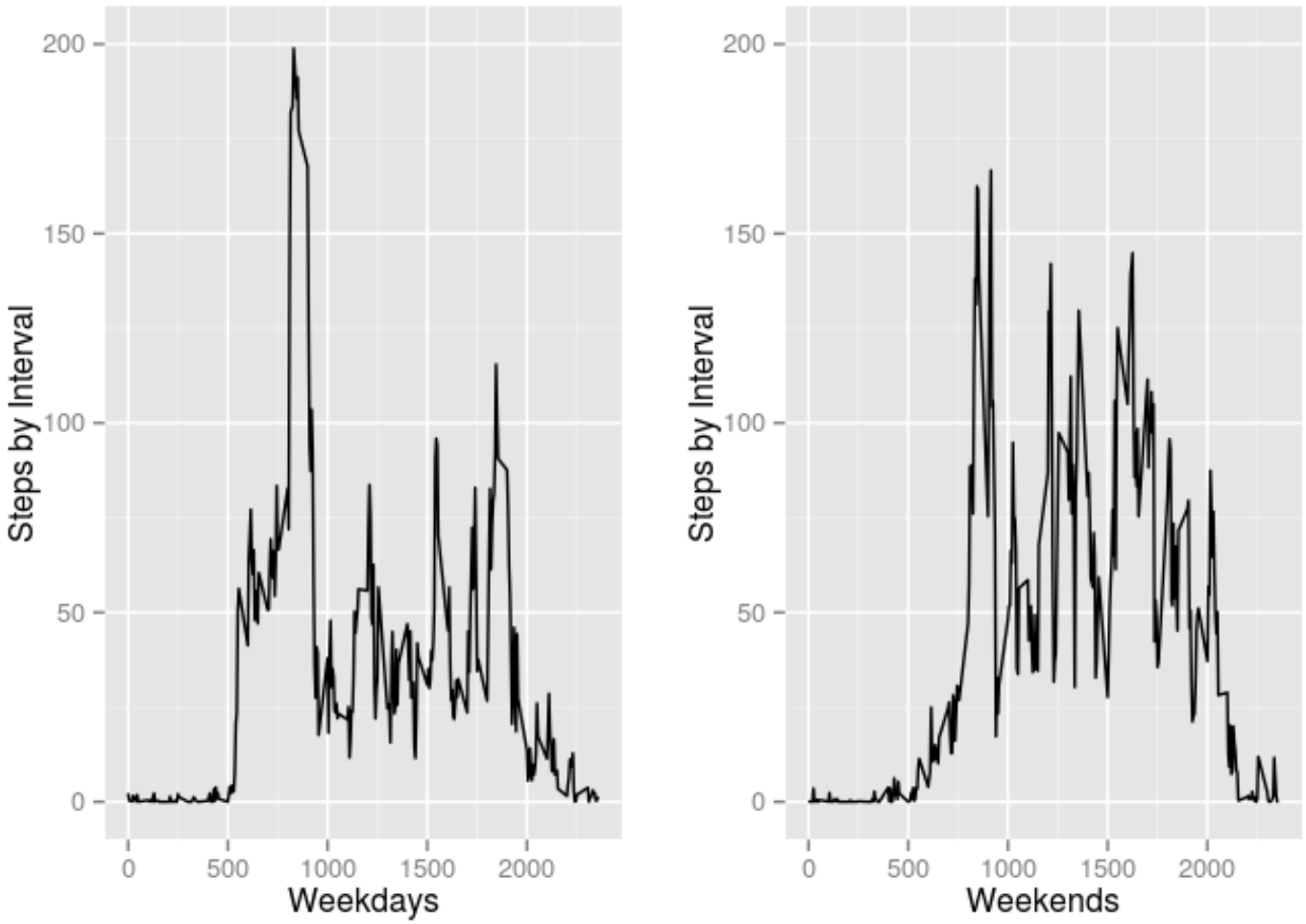


We can see that the distribution of steps per day, after imputation, still approximates a normal distribution. The mean steps per day is 10766 and the median value is 10765.

Are there differences in activity patterns between weekdays and weekends?

In order to differentiate between weekdays and weekends, I use the ‘weekdays’ function against the Date types and compare it to a list of valid weekday names. The level of granularity is at the Intervnal level so as to show patterns across the day. I then subset the imputed dataset and plot the two subsets side-by-side.

```
#8 Compare data with inputed values between weekdays and weekends
#Subset imputedDat into dataframes based on day of week

imputedDat$date <- as.Date(imputedDat$date, "%Y-%m-%d")
weekdayDat <- imputedDat[weekdays(imputedDat$date)  %in%  c("Monday", "Tuesday",
"Wednesday", "Thursday", "Friday"),]
weekendDat <- imputedDat[!weekdays(imputedDat$date) %in%  c("Monday", "Tuesday",
"Wednesday", "Thursday", "Friday"),]
#str(weekdayDat)

#Average by Interval
imputedWeekdayInt <-aggregate(steps.x ~ interval, data=weekdayDat, mean)
imputedWeekendInt <-aggregate(steps.x ~ interval, data=weekendDat, mean)
#str(imputedWeekdayInt)

avgWeekday <- ggplot(data=imputedWeekdayInt, aes(interval, steps.x)) +
  stat_summary(fun.y=mean,   geom="line", fill ="blue", na.rm=TRUE) +
  stat_summary(fun.y=median, geom="line", fill ="purple", na.rm=TRUE ) +
   ylim(0, 200) +
  ylab("Steps by Interval") + xlab("Weekdays")

avgWeekend <- ggplot(data=imputedWeekendInt, aes(interval, steps.x)) +
  stat_summary(fun.y=mean,   geom="line", fill ="blue" , na.rm=TRUE) +
  stat_summary(fun.y=median, geom="line", fill ="purple" , na.rm=TRUE) +
   ylim(0, 200) +
  ylab("Steps by Interval") + xlab("Weekends")

library(gridExtra)
grid.arrange(avgWeekday, avgWeekend, ncol=2)
```



From these results it is apparent that, during the week, the subjects in these observations tend to concentrate their exercise in the early morning or evening hours. On the other hand their activity is more evenly spread across the daytime hours on weekends.