Reproducible Research: Peer Assessment 1

Sherri Verdugo

September 12, 2014

Contents

Loading and processing the data

The following libraries are needed for this project:

```
library(ggplot2)#needed for plots
library(xtable)#needed for pretty table
```

1: Load in the data

```
df.1 <- read.csv("activity.csv")
```

Check the first 5 rows of the data before processing the data for the project.

This step allows us to evaluate the first five initial rows of data in the df.1 data set.

```
h.noproc <- head(df.1, 5)
h.noproc <- xtable(h.noproc, caption="First 5 rows: non-processed", label="Head Xtable", digits=1)
print(h.noproc, include.rownames = TRUE, caption.placement="top")
```

% latex table generated in R 3.1.0 by xtable 1.7-3 package % Sun Sep 14 00:19:06 2014

Initial Process and Removing the NA's in the data.

This step allows us to evalute the first five initial rows of data in the df data set after removing na. values.

Table 1: First 5 rows: non-processed

	steps	date	interval
1		2012-10-01	0
2		2012-10-01	5
3		2012-10-01	10
4		2012-10-01	15
5		2012-10-01	20

```
df <- na.omit(df.1)
h.proc <- head(df, 5)
h.proc <- xtable(h.proc, caption="First 5 rows: processed", label="HeadP Xtable", digits=1)
print(h.proc, include.rownames = TRUE, caption.placement="top")
```

% latex table generated in R 3.1.0 by xtable 1.7-3 package % Sun Sep 14 00:19:06 2014

Table 2: First 5 rows: processed

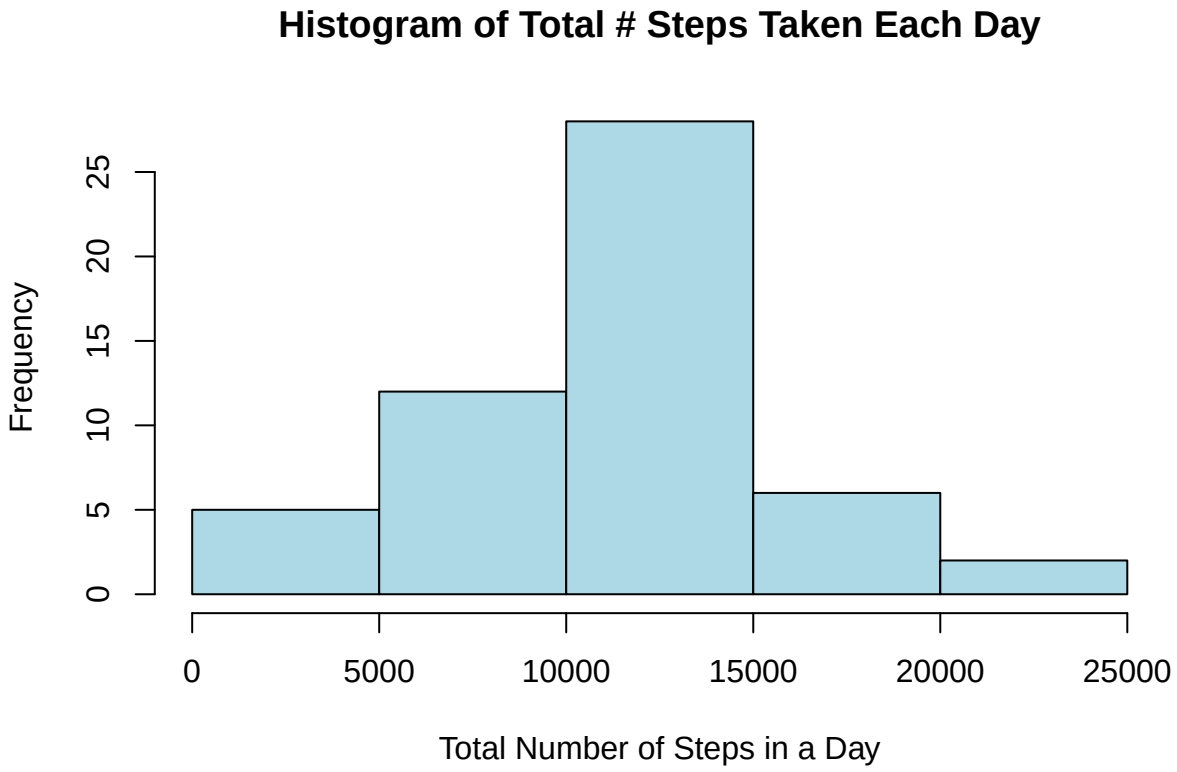
	steps	date	interval
289	0	2012-10-02	0
290	0	2012-10-02	5
291	0	2012-10-02	10
292	0	2012-10-02	15
293	0	2012-10-02	20

What is mean total number of steps taken per day?

We have a few steps to take here. First, I like to plot the data. This time we are using qplot from the ggplot2 library. Make sure you have that installed. If you do not have it, be sure to use install.packages("ggplot2"). Out of curiosity, the histogram plot was tried two times to find the better plotting function as shown below.

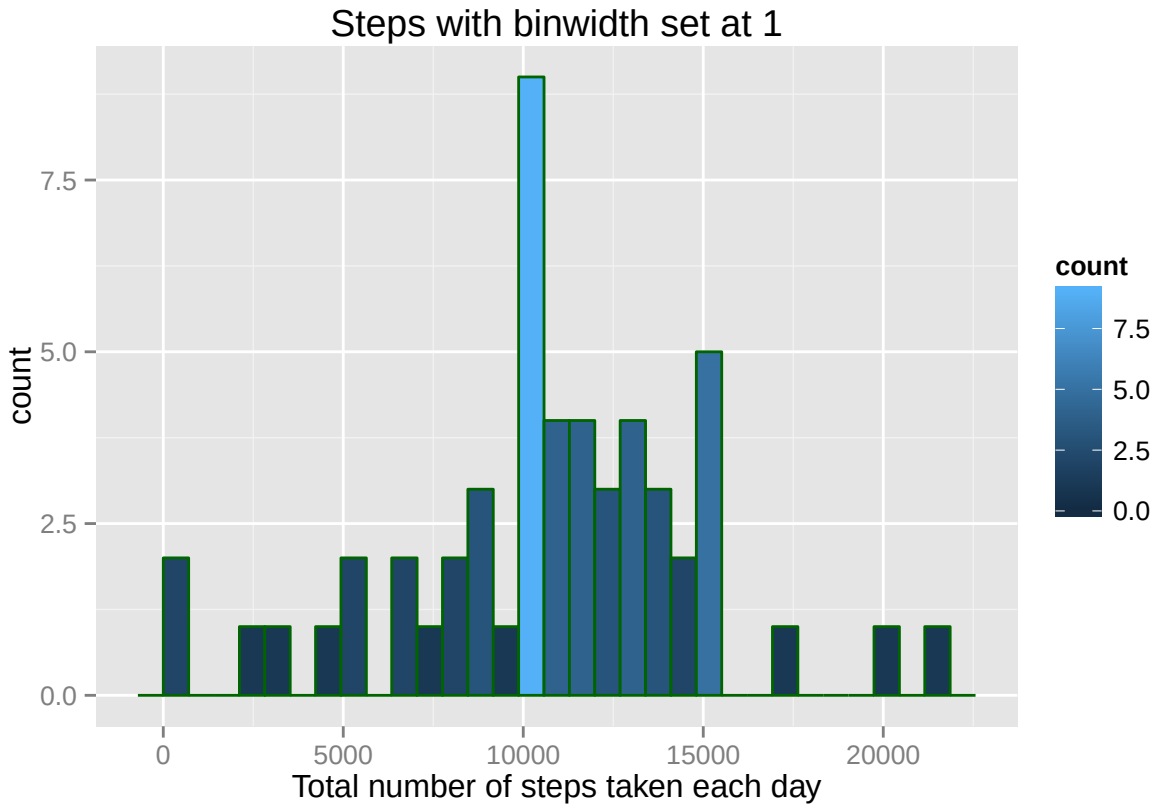
Histogram of the total number of steps taken each day

```
df.steps <- aggregate(steps ~ date, df, sum)
##Raw Histogram
hist(df.steps$steps, col="lightblue", main = "Histogram of Total # Steps Taken Each Day",
     xlab="Total Number of Steps in a Day")
```




```
##New Bins set
qplot(steps, data=df.steps, binwidth = "1", xlab = "Total number of steps taken each day",
      main = "Steps with binwidth set at 1", na.rm=TRUE) +
      geom_histogram(colour="darkgreen", aes(fill = ..count..))
```

```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```



The overall shape of the histogram slightly changes when the binwidth is set at 1.

Mean total number of steps per day

The next step is to find two measures of central tendency: the mean total number of steps per day and the median number of steps per day.

```
mean(df.steps$steps)
```

[1] 10766

```
median(df.steps$steps)
```

[1] 10765

```
desc <- summary(df.steps) #from the psych library
desc <- xtable(desc, caption="Summary Statistics for Data",
               label="Description Xtable", digits=1)
print(desc, include.rownames = TRUE, caption.placement="top")
```

% latex table generated in R 3.1.0 by xtable 1.7-3 package % Sun Sep 14 00:19:10 2014

The summary function was ran to double check the two values of central tendency that we are interested in: the median and the mean.

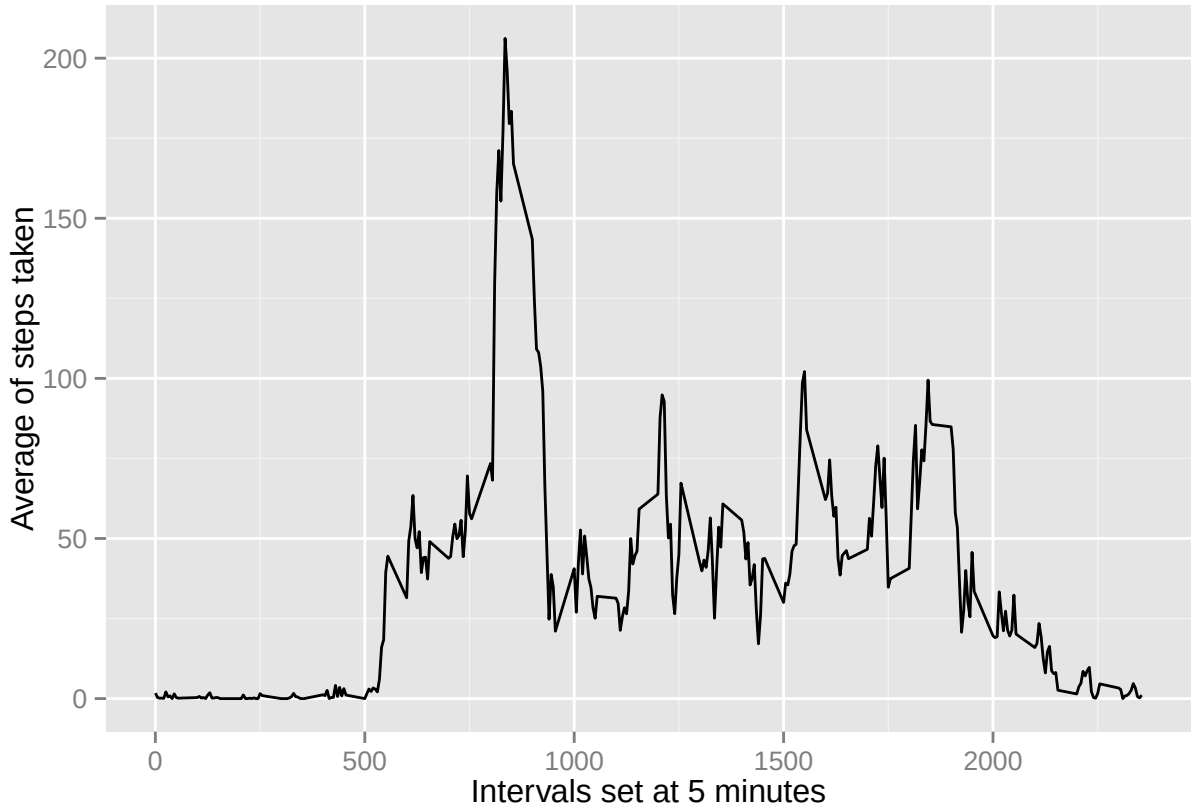
Table 3: Summary Statistics for Data

	date	steps
1	2012-10-02: 1	Min. : 41
2	2012-10-03: 1	1st Qu.: 8841
3	2012-10-04: 1	Median :10765
4	2012-10-05: 1	Mean :10766
5	2012-10-06: 1	3rd Qu.:13294
6	2012-10-07: 1	Max. :21194
7	(Other) :47	

What is the average daily activity pattern?

This time, we are looking at the average daily activity pattern. This means that we have to aggregate and then plot. Again, we are using the library (ggplot2)... make sure you have that installed.

```
library(ggplot2)
df.averages <- aggregate(x=list(steps=df$steps), by=list(interval=df$interval), FUN=mean)
ggplot(data=df.averages, aes(x=interval, y=steps)) + geom_line() +
  xlab("Intervals set at 5 minutes") + ylab("Average of steps taken")
```



Further, on average for all days in the dataset df, the 5 minute intervals contains the following maximum number of steps:

```
df.averages[which.max(df.averages$steps),]
```

```
##     interval steps
## 104      835 206.2
```

Imputing missing values

This dataset has many missing values that are coded as NA. The very presence of the missing data may introduce what is known as bias into the data analysis process. We need to take care to address this and carefully impute the data using r. First we identify the number of missing items from the dataframe. Finally, we generate a table to identify the number of missing items in this dataset.

```
df.missing <- is.na(df$steps)
num.missing <- sum(df.missing)
table(df.missing)
```

```
## df.missing
## FALSE
## 15264
```

```
table (num.missing)
```

```
## num.missing
## 0
## 1
```

We can replace the missing values with the mean value of the 5-minute intervals by using a function that is conditional on the is.na and number of steps. This was tricky as it took more time to run through various options of how to do this.

```
nafiller <- function(steps, interval){
  filler <- NA
  if (!is.na(steps))
    filler <- c(steps)
  else
    filler <- (df.averages[df.averages$interval==interval, "steps"])
  return(filler)
}
myfill.df <- df
myfill.df$steps <- mapply(nafiller, myfill.df$steps, myfill.df$interval)
```

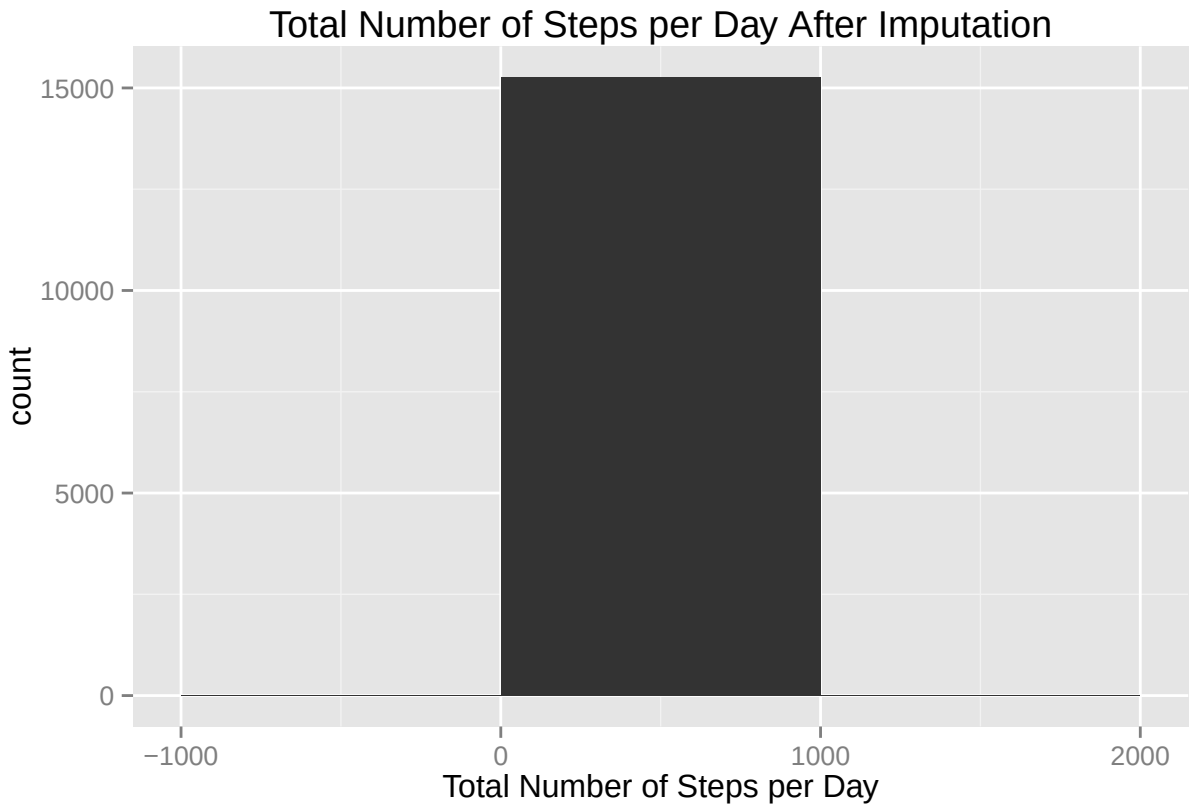
Now we can look at what we have done so far by calling the object.

```
head(myfill.df)
```

```
##     steps       date interval
## 289     0 2012-10-02        0
## 290     0 2012-10-02        5
## 291     0 2012-10-02       10
## 292     0 2012-10-02       15
## 293     0 2012-10-02       20
## 294     0 2012-10-02       25
```

The next thing we can do is utilize the histogram for visualization with the filled in data set.

```
myts <- tapply(myfill.df$steps, myfill.df$date)
qplot(myts, binwidth=1000, xlab="Total Number of Steps per Day",
      main="Total Number of Steps per Day After Imputation")
```




```
library(psych)
```

```
## Warning: package 'psych' was built under R version 3.1.1
```

```
##
## Attaching package: 'psych'
##
## The following object is masked from 'package:ggplot2':
##
##     %+%
```

```
describe(myts)
```

```
##   vars     n  mean    sd median trimmed   mad min max range skew kurtosis
## 1    1 15264 30.72 17.47     29   30.65 22.24   2  60    58 0.08    -1.26
##     se
## 1 0.14
```

```
mean(myts)
```

```
## [1] 30.72
```

```
median(myts)
```

```
## [1] 29
```

```
summary(myts, na.rm=TRUE)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     2.0    16.0    29.0    30.7    47.0    60.0
```

From the imputation process, we notice that the mean and median values are higher. One explanation is that in the original data with some days that have 'steps' with the value of NA for any 'interval'. That means that the number of steps would have 0 values that are removed in the original histogram. After the imputation, the values of the mean and median increase.

Are there differences in activity patterns between weekdays and weekends?

To do this step, we have to look at the day of the week for every single measurement in the data that we are analyzing. We will continue using our filled data (myfill.df) for the next portion of this assignment.

```
week.identify <- function(date){
 day <- weekdays(date)
 if (day %in% c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday"))
   return("Weekday")
 else if (day %in% c("Saturday", "Sunday"))
   return("Weekend")
 else
   stop("Invalid Date")
}
myfill.df$date <- as.Date(myfill.df$date)
myfill.df$day <- sapply(myfill.df$date, FUN=week.identify)
```

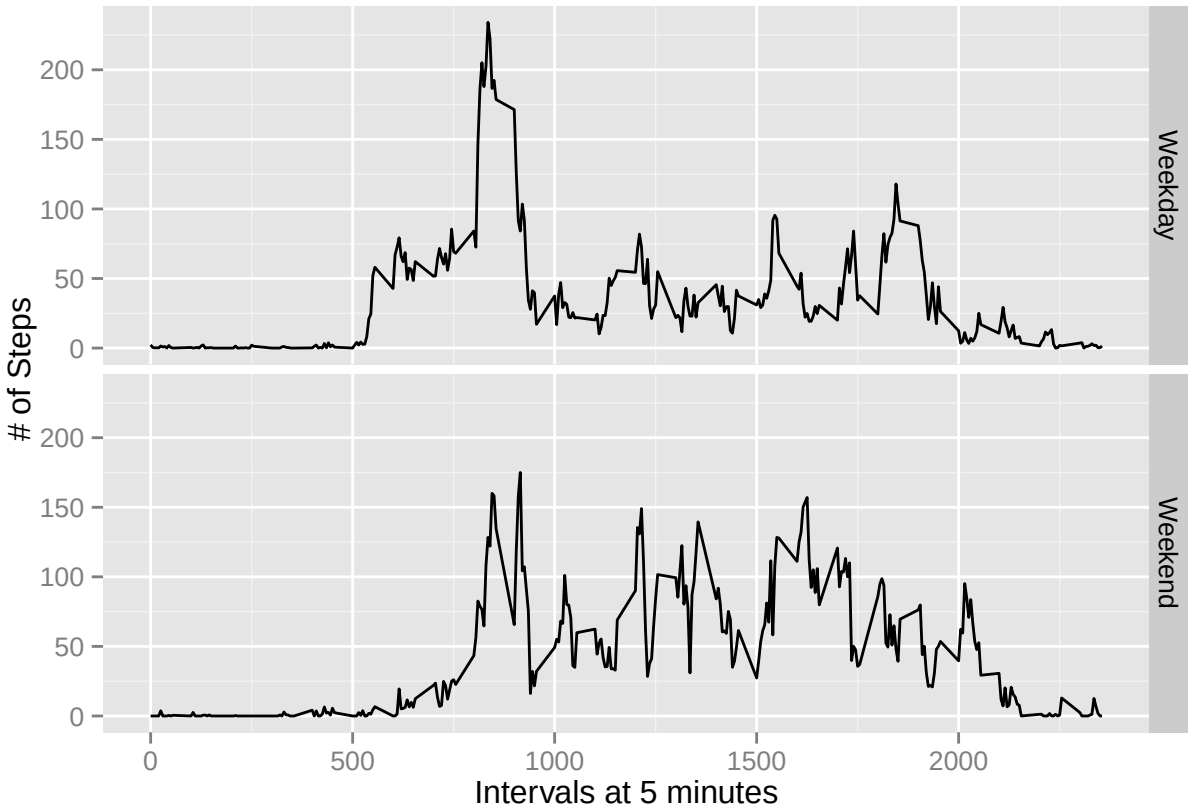
Let's look at what we have so far for identifying the day of the week as a weekend or weekday. Is R smart enough to handle that? The answer is, yes.

```
head(myfill.df$day)
```

```
## [1] "Weekday" "Weekday" "Weekday" "Weekday" "Weekday" "Weekday"
```

The next step for this is to visually explore the data that we created. The option that is used is the panel plot that contains the average number of steps taken on either weekends or weekdays. Do people take more steps on the weekends or the weekdays?

```
avg <- aggregate(steps ~ interval + day, data=myfill.df, mean)
ggplot(avg, aes(interval, steps))+geom_line()+ facet_grid(day ~ .) + xlab("Intervals at 5 minutes") + yl
```



From the graph we see that weekday steps start out similar to the weekend steps. The difference is that more regular paterns occur in the weekend steps.