R Tutorial for STAT 350 Lab 2

Author: Leonore Findsen, Chunyan Sun, Sarah H. Sellke, Jeremy Troisi

1. Numerical Summaries and Graphs

Example 1. Time to Start a Business (**Data: eg01-23time24.txt**)
An entrepreneur faces many bureaucratic and legal hurdles when starting a new business. The World Bank collects information about starting businesses throughout the world. It has determined the time, in days, to complete all of the procedures required to start a business. Data for 195 countries are included in the data set. For this section we will examine data for a sample of 24 of these countries. Here are the data:

13	66	36	12	8	27	6	7	5	7	52	48
15	7	12	94	28	5	13	60	5	5	18	18

a) Find the mean and the standard deviation of the times it took to start a new business among all countries in the data set.
b) Find the five-number summary of the times it took to start a new business.
c) Create a histogram of the times it took to start a new business.
d) Do you think the median is close to the mean?
e) Create a boxplot (modified) of the times it took to start the new businesses.

Solution:

In your solution, please list the complete code at the beginning. In this tutorial, the code will be repeated in each of the sections so that it can be explained more fully. Please ONLY repeat code if it is right before an answer.

```
#Reading in the data
#You may either write down the procedure if you use RStudio,

# Read in data using the interface
# Import Dataset --> From CSV --> browse to the file --> set delimiter
#   to tab --> Change name to TimeStart --> Import

# or you may include the code
TimeStart <- read.table("eg01-23time24.txt", header = T, sep = "\t")
#In this data file, there are both spaces and tabs. The
# sep"" argument tells R that you only want to use tabs (\t) as
# separators. This occurs in some of the data sets so if you
# notice that there are spaces in either the variable names or
# the data itself, you need to add this additional argument in
# the command.
#
#a) Only use one of the following three methods
# Method 1
mean(TimeStart$TimeToStart)
sd(TimeStart$TimeToStart)

#Method 2
attach(TimeStart)
mean(TimeToStart)
sd(TimeToStart)
```

Author: Leonore Findsen, Chunyan Sun, Sarah H. Sellke, Jeremy Troisi

```
#Method 3 (do NOT use this method if you have a Mac)
mean(TimeStart[,2])
sd(TimeStart[,2])

#b)
fivenum(TimeStart$TimeToStart)

#c)
install.packages("ggplot2") #Only needs to be run once
library(ggplot2) #Needs to be run for each R session

#In the following, TimeStart is the data set name and TimeToStart is
#the variable that you want to make the histogram of.
#  aes(y=..density..) makes the histogram a density plot;
#  The "bins" argument specifies the number of bins that you want
#   (the default colors for the rectangles are also modified);
#  geom_density() generates the estimated kernel density curve in red;
#  stat_function() generates the normal approximation in blue
#  ggtitle() generates the title of the plot.
windows()
xbar <- mean(TimeStart$TimeToStart)
s <- sd(TimeStart$TimeToStart)
ggplot(TimeStart, aes(TimeToStart)) +
  geom_histogram(aes(y =..density..),
            bins=sqrt(nrow(TimeStart))+2, fill="grey", col="black") +
  geom_density(col="red", lwd=1) +
  stat_function(fun=dnorm, args = list(mean = xbar, sd = s),
                col = "blue", lwd = 1) +
  ggtitle("Histogram of Time To Start")

#e)
windows()
ggplot(TimeStart, aes(x = "", y = TimeToStart)) +
  stat_boxplot(geom="errorbar") +
  geom_boxplot() +
  ggtitle("Boxplot of Time To Start") +
  stat_summary(fun.y = mean, col = "black", geom = "point", size = 3)
```

Now, we will answer the questions presented above and explain the commands in more detail.

Reading In the Data (assuming the data file is in current working directory):

```
TimeStart <- read.table("eg01-23time24.txt", header = T, sep = "\t")
#In this data file, there are both spaces and tabs. The
# sep command tells R that you only want to use tabs (\t) as
# separators. This occurs in some of the data sets so if you
# notice that there are spaces in either the variable names or
# the data itself, you need to add this addition keyword in
# the command.
```

If you use the RStudio interface, be sure that the Data Preview looks correct before loading in the file. If necessary, set the delimiter to tab.

R Tutorial for STAT 350 Lab 2

Author: Leonore Findsen, Chunyan Sun, Sarah H. Sellke, Jeremy Troisi

To access a variable (e.g., to compute its mean), you need to tell R which table the variable is contained in. There are three ways to do this (also, see the examples below):

1) Use `tableName$variableName`. This is the method that we used in Lab 1. This is the easiest method to understand.
2) First, `attach(TableName)` (attach the variable names in the table to the R search path), as in `attach(TimeStart)`.
   Then you can just type the variable name, rather than having to type the table name and a "$" sign before the variable name. Note when using the function `attach()`, the name of the R table has to be different than the name of any of the variables in the table.
   This method is easier to use if you are an experienced programmer. If you are not, it can lead to issues that are harder to detect.
3) Insert the column number of the variable in square brackets after the R table name, as in `tableName[, columnNumber]`. Notice that there is a comma before the column number.

In this example, we are interested in the variable "TimeToStart" which is the second variable in the TimeStart table. Therefore, you could indicate it by

1) `TimeStart$TimeToStart`,
2) `attach(TimeStart)` then just use `TimeToStart`, or
3) `TimeStart[, 2]`,

respectively. The first method will be used in all future labs. To see the variable names, look at the data table.

For question (a), I will show you how to use all of the methods. For the rest of this tutorial, I will only be using Method 1.

a) Find the mean and the standard deviation of the times it took to start a new business among all countries in the data set.

Solution:

Since the answers are just numbers, when you present your output, please also provide the command that generates it.

Method 1

```
> mean(TimeStart$TimeToStart)
[1] 23.625
> sd(TimeStart$TimeToStart)
[1] 23.82876
```

Method 2

```
> attach(TimeStart)
> mean(TimeToStart)
[1] 23.625
> sd(TimeToStart)
[1] 23.82876
```

Method 3

```
> mean(TimeStart[,2])
[1] 23.625
> sd(TimeStart[,2])
[1] 23.82876
```

From any of the methods above,
Mean = 23.625, Standard deviation = 23.82876.

As long as you include the command before the output, you do not need to retype the numbers.

b) Find the five-number summary of the times it took to start a new business.

Solution:

```
> fivenum(TimeStart$TimeToStart)
[1]  5  7 13 32 94
```

From the R output above,
Min = 5, Q_1 = 7, Median = 13, Q_3 = 32, Max = 94.

Be sure to always state what each of the values refer to. You will not receive full credit if you just include the output above.

Note: There are other ways of computing the five-number summary. However, this method will generate the values that are obtained using the method in our textbook.

2. Creating Histograms

c) Create a histogram of the times it took to start a new business.

Solution:

Remember, for small to medium data sets, the number of bins should be
$number\ of\ bins \approx \sqrt{number\ of\ data\ points}$

For large data sets, you should start with 20 bins to see if that looks appropriate.

We will be utilizing the "ggplot2" graphics package that can be installed from within an R session for creating histograms:

```
install.packages("ggplot2") #Only needs to be run once
```

R might ask you from which mirror to install the package. Please choose the one that is closest to your location. After you install the package you need to be sure that it is active.

R Tutorial for STAT 350 Lab 2

Author: Leonore Findsen, Chunyan Sun, Sarah H. Sellke, Jeremy Troisi

```
library(ggplot2) #Needs to be run for each R session
```

Please ignore any warning messages.

All ggplot2 function calls work as follows:

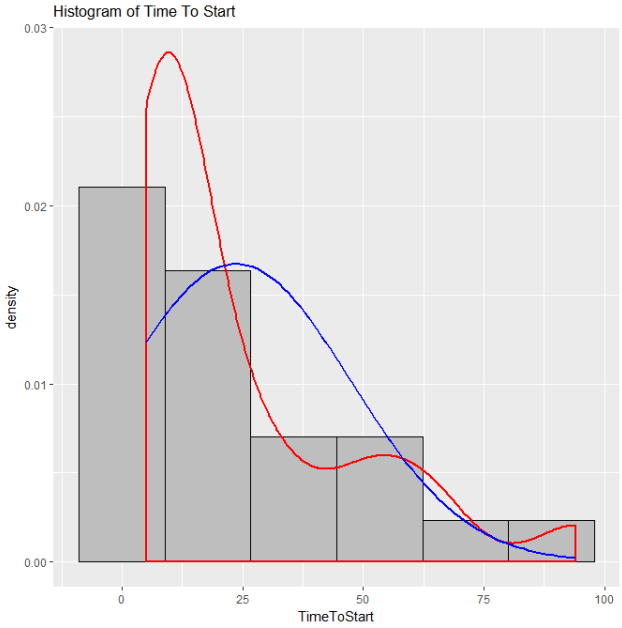
1. First, the function `ggplot()` is called, followed by a `+` sign
2. Then we specify which type of plot we want, such as a histogram (`geom_histogram()`), followed by a `+` sign
3. Then we can specify dozens of other options such as the title, x-label, and y-label. Each must be followed by a `+` sign, except the last option.

Pay attention to the "bins" argument. The default number of the bins in `geom_histogram()` is not correct for small to medium data sets. This option lets you specify the number of bins, such as the square root of the number of rows in the data set. You can modify this value as desired to improve the visualization. Remember, for large data sets, the default value might or might not be appropriate. Below, I have added two extra bins to the recommended value for better visualization.

```
#In the following, TimeStart is the data set name and TimeToStart is
#the variable that you want to make the histogram of.
#  aes(y=..density..) makes the histogram a density plot;
#  The "bins" argument specifies the number of bins that you want
#   (the default colors for the rectangles are also modified);
#  geom_density() generates the estimated kernel density curve in red;
#  stat_function() generates the normal approximation in blue
#  ggtitle() generates the title of the plot.
xbar <- mean(TimeStart$TimeToStart)
s <- sd(TimeStart$TimeToStart)
ggplot(TimeStart, aes(TimeToStart)) +
  geom_histogram(aes(y =..density..),
                 bins=sqrt(nrow(TimeStart))+2, fill="grey", col="black") +
  geom_density(col="red", lwd=1) +
  stat_function(fun=dnorm, args = list(mean = xbar, sd = s),
                col = "blue", lwd = 1) +
  ggtitle("Histogram of Time To Start")
```

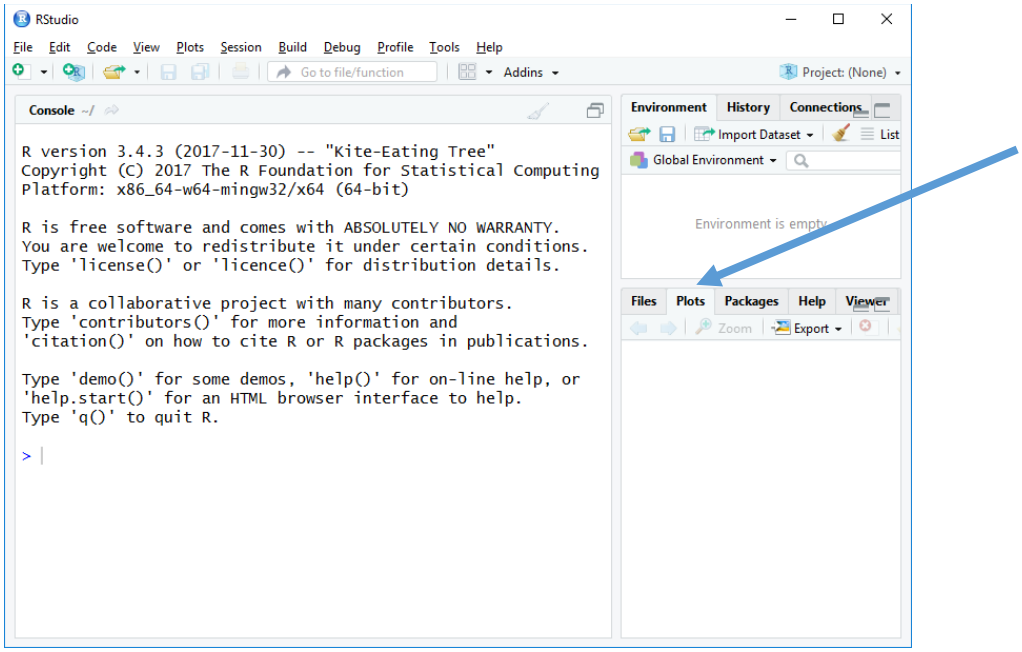
R Tutorial for STAT 350 Lab 2

Author: Leonore Findsen, Chunyan Sun, Sarah H. Sellke, Jeremy Troisi



The red line is a smoothed line representing the histogram which is called a Kernel. This is generated from the function `geom_density()`. The blue line is the normal approximation to the histogram using the mean and standard deviation from the data. This is generated from the function `stat_function()`.We will revisit these lines in future labs.

The above method will place the graph in the lower right window shown below (shown by the blue arrow).



Author: Leonore Findsen, Chunyan Sun, Sarah H. Sellke, Jeremy Troisi

You may either use the Snipping Tool or the Export → Copy to Clipboard and then paste the graph into your Word document.

The following is an alternate method to display the graphs.

If you are using the Windows operating system, you can run the command:

```
windows()
```

And then run your graphing command. The `windows()` function will create a separate window in which your graph will appear. Then you can right click to get the copy commands. This is optional.

If you are using a Mac, the following command does the same thing:

```
quartz()
```

I will be using the `windows()` command in the code; so if you have a Mac, be sure that you change the command.

d) Do you think the median is close to the mean?

Solution

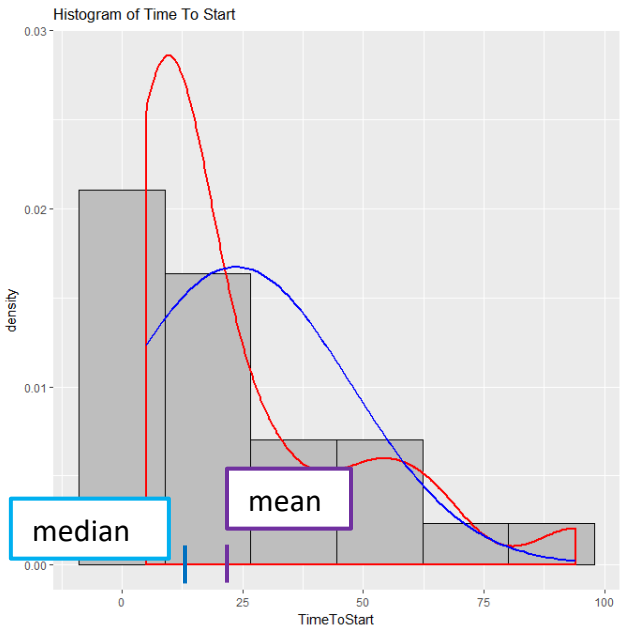
There are a number of different ways that this can be accomplished.

1) Compare the numbers

From parts a) and b), Mean = 23.625, Median = 13.
I would say that these numbers are not close in this data set because their absolute difference is 10.625 which is large compare to the range of data (approximately 100).

Author: Leonore Findsen, Chunyan Sun, Sarah H. Sellke, Jeremy Troisi

2) Visually look at the numbers when they are plotted on the histogram.



To me, these numbers do not look close to each other.

3) Compare the difference of the numbers to the total spread of the numbers

$$\frac{mean - median}{maximum - minimum} = \frac{23.625 - 13}{94 - 5} = 0.119$$

This is ~12% which is fairly large.

4) Compare the difference of the numbers to the standard deviation

$$\frac{mean - median}{standard\ deviation} = \frac{23.625 - 13}{23.82876} = 0.446$$

Therefore the difference is about half the standard deviation which is fairly large.

In this case, I would say that the two numbers are not close to each other. However, these four methods will not always provide the same answer. When answering this question, you need to look at all of the information to make a decision. In some cases, it is possible for different people to have different answers.

R Tutorial for STAT 350 Lab 2

Author: Leonore Findsen, Chunyan Sun, Sarah H. Sellke, Jeremy Troisi

3. Boxplots

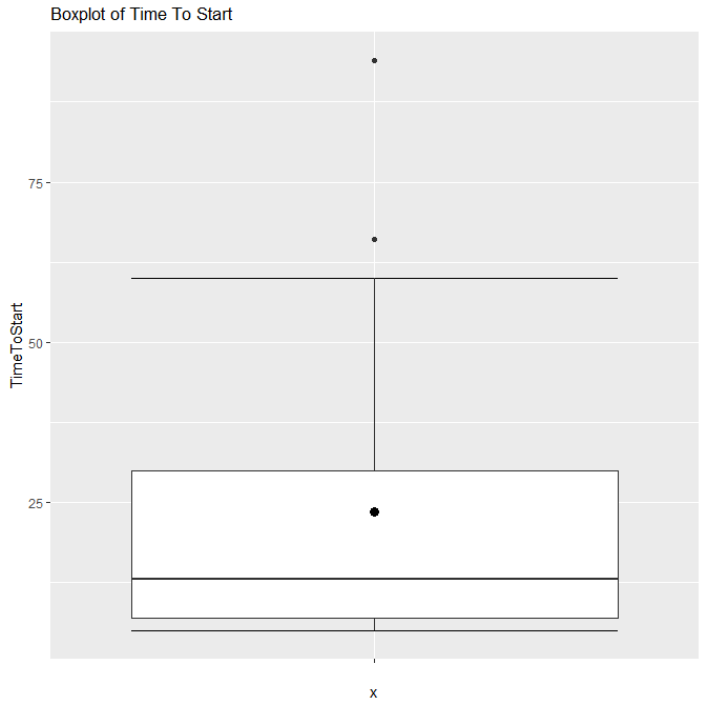
e) Create a boxplot (modified) of the times it took to start the new businesses.

Solution:

The following is the procedure for generating a modified boxplot. In a modified boxplot, the outliers are explicitly plotted. The procedure is more complicated if you want to generate boxplots across multiple groups for comparison (side-by-side boxplot). That procedure will be shown later in the semester.

Note the `stat_summary()` function will put the mean on the boxplot. You may modify the color and size.

```
windows()
ggplot(TimeStart, aes(x = "", y = TimeToStart)) +
  stat_boxplot(geom="errorbar") +
  geom_boxplot() +
  ggtitle("Boxplot of Time To Start") +
  stat_summary(fun.y = mean, col = "black", geom = "point", size = 3)
```



The circle on the boxplot is the location of the mean.

Note: Please resize your graphics so that they fit on the page especially when you have more than one plot.