Descriptive Statistics and Graphics for Exploratory Data Analysis

Dr. Christopher Brown

December 22, 2018

Introduction

In the early 1960's, John Tukey of Bell Labs began to promote a concept he called *exploratory data analysis*, or EDA. At the time, the field of statistics placed an enormous focus on *confirmatory data analysis*, which is the formal numerical testing of hypotheses. But how do investigators formulate those hypotheses in the first place? Tukey noted that with the right tools, investigators might formulate hypotheses from data; this could complement hypothesis formulation based on experience, which might be ill-informed or biased.

Exploratory data analysis includes descriptive (or summary) statistics for data and useful graphics. Because exploratory data analysis is, well, exploratory, one might need to create many summaries and graphics. In practice this requires computational power. Tukey's call for better EDA led directly to the development of the **S** statistical language at Bell Labs, which in turn led to **R**. So if it seems as though R has a great toolset for EDA, that's baked in its design!

For more information, visit the Wikipedia page on exploratory data analysis.

In this document we briefly review some of R's functions for creating summary statistics and graphics related to EDA. For graphics I'll present versions from both the base R graphics package and from the *ggplot2* package.

Univariate Summary Statistics: Measures of Center

One of the first descriptive statements we would like to make about numerical univariate (one variable) data is about its location or center. That is, we would like to say "Values in data set X are mostly located at point x on the number line". For example, suppose our data set is

```
## X = {  0 1 1 1 1 1 2  }
```

Then we would certainly want to say that our data is "mostly" located at $x = 1$. But as always, fuzzy words like "mostly" tend to have different interpretations in different situations, so we need to be a little more particular. For example, suppose our data set is

```
## Y = {  0 1 1 1 1 1 2 2 2 2 100  }
```

Then at what location is our data "mostly" located? If we take into account distance along the number line, then we have a lot of data $\approx$ 1.4ish and we have one data point at 100, so does this mean the center of the data is around 15 or 20? Or do we just line up the data points in order (there are 11 of them) and pick the middle one (the 6th one)? This leads us to two different concepts answering our question.

The **mean** $\overline{x}$ of the data is the classical average of the data:$$/overline%7Bx%7D=/frac%7Bx_1+x_2+/cdots +x_n}{n}$$The mean takes into account distance along the number line, and so the mean will be sensitive to data that is located very far from other data. In R, we can compute the mean with the *mean* command. So for our data sets X and Y above we have

```
mean(X)
```

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

and

```
mean(Y)
```

```
## [1] 10.27273
```

The **median** of the data is the "middle" data point when the data points have been lined up in order; if there is no middle data point, then we take the mean of the two data points closest to the middle (this happens when the data set has an even number of data points). In R we can compute the median with the *median* command. So for our data sets X and Y above we have

```
median(X)
```

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

and

```
median(Y)
```

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

The mean requires the data to be of a numerical type, and the median requires the data type have an order (of ordinal type). But what if the data is purely categorical in nature? For example, suppose I survey a group of people about their favorite color? Here is one example data set resulting from a multiple choice survey question:

```
## Z = {  red red red blue blue blue blue blue blue blue green green other other other other other  }
```

In this case we might say that "most" of the data is lcoated at "blue", because it is the most frequently occurring category. The **mode** of the data is the most frequently occurring data point or points. In R, we do NOT use the *mode* command to compute the mode (look up the *mode* command to see what it actually does!). Oddly, there is no built-in command for computing the mode in R; we'll see why this is so when we look at measures of variation for categorical data in the next section. When the data has only a few categories it is often easiest to pick out the mode using the *table* and *sort* commands:

```
sort(table(Z))
```

```
## Z
## green   red other  blue
##     2     3     5     7
```

Here, it is clear that the mode is "blue". If the data has a few hundred categories we may have to adjust this approach. We can write a custom fuction if needed, but we rarely need to compute a mode in these scenarios anyway.

Univariate Summary Statistics: Measures of Variation

Now that we can get some sense of the center of the data, we can also ask how much the data tends to vary about the center.

When our data is quantitative and our measure of center is the mean $\overline{x}$, we can answer this question with the **standard deviation**

$$s = \sqrt{\frac{(x_1 - \overline{x})^2 + \cdots + (x_n - \overline{x})^2}{n-1}}$$

In R, we can compute standard deviation with the *sd* command. FOr our data sets X and Y above we have

```
sd(X)
```

```
## [1] 0.5773503
```

and

```
sd(Y)
```

```
## [1] 29.76606
```

There are a number of other measures of variation for quantitative data that are used regularly. The *range* of a quantitative data set is simply the difference between the maximum and minimum values of the data:

```
max(X)-min(X)
```

```
## [1] 2
```

The *range* command simply returns the min and max of the data.

```
range(X)
```

```
## [1] 0 2
```

We can also compute the percentiles of the data for various percentile levels. The five number summary of quantitative data is a common and useful tool:

```
quantile(X)
```

```
##   0%  25%  50%  75% 100%
##    0    1    1    1    2
```

We'll see a couple of graphics based on the five-number summary later.

When our data is ordinal and our measure of center is the median, we have fewer choices for measuring variation. We can still order the data and compute percentiles for the data points, and that can lead to some useful comparisons of variation.

Graphics for Univariate Quantitative Data

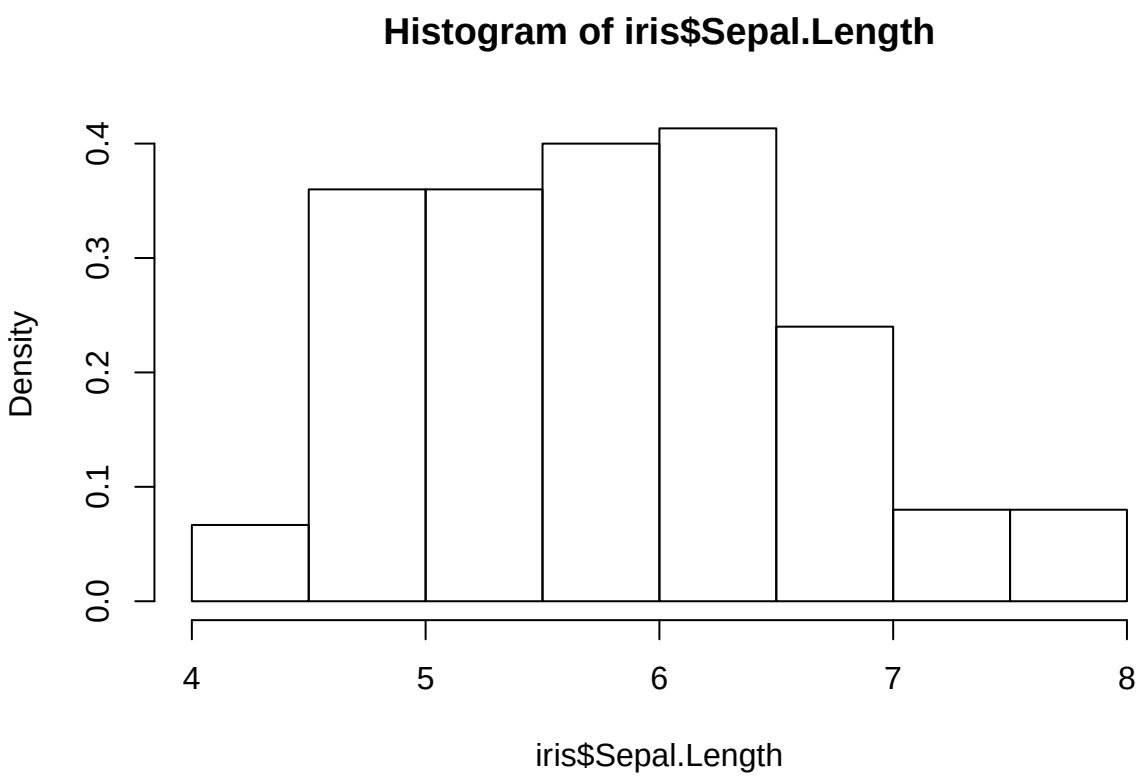
By far our most useful graphical representations for univariate data are the **histogram** and **density** plots. The histogram bins the data and then plots the frequency or relative frequency of bins versus the bin values. The density smooths the histogram and plots the relative frequency of data versus the data values. Let's take a look at the *iris* data set in R:

```
data('iris')
head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

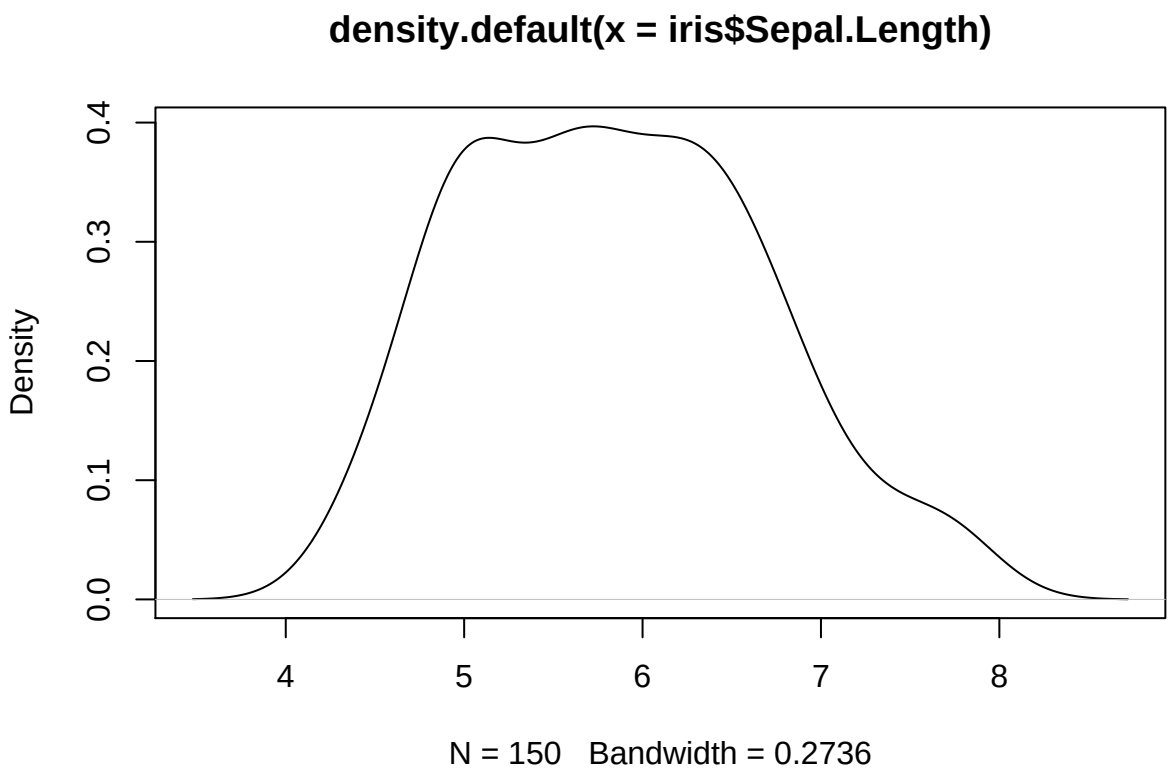
Let's build a histogram for the *Sepal.Length* variable. Here we use the *freq=FALSE* option to plot the relative frequencies for bins (check the labels on the vertical axis).

```
hist(iris$Sepal.Length,freq=FALSE)
```



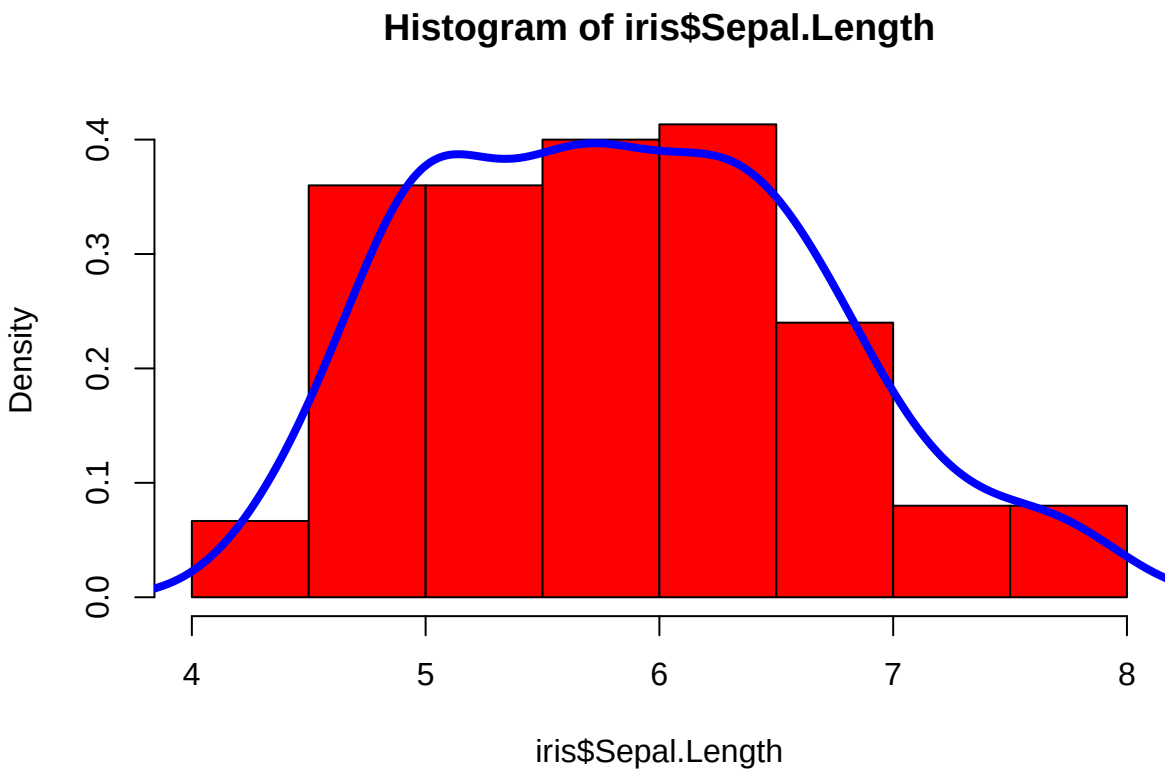
We can also plot the density function for this variable.

```
plot(density(iris$Sepal.Length))
```



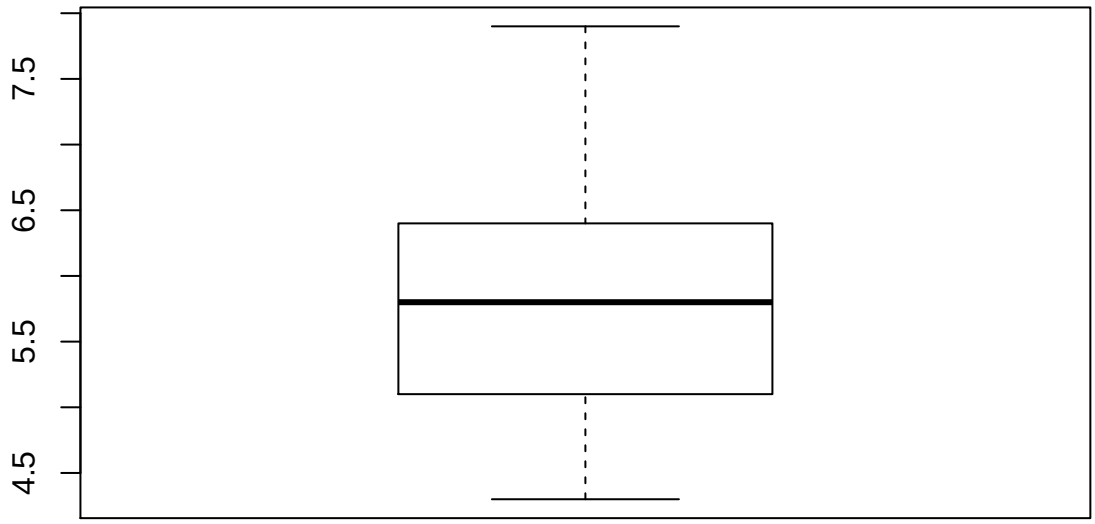
We can combine the two for a good look at the data.

```
hist(iris$Sepal.Length,freq=FALSE,col="red",
     title="Histogram and density for Sepal Length")
points(density(iris$Sepal.Length),col='blue',lwd=4,type='l')
```



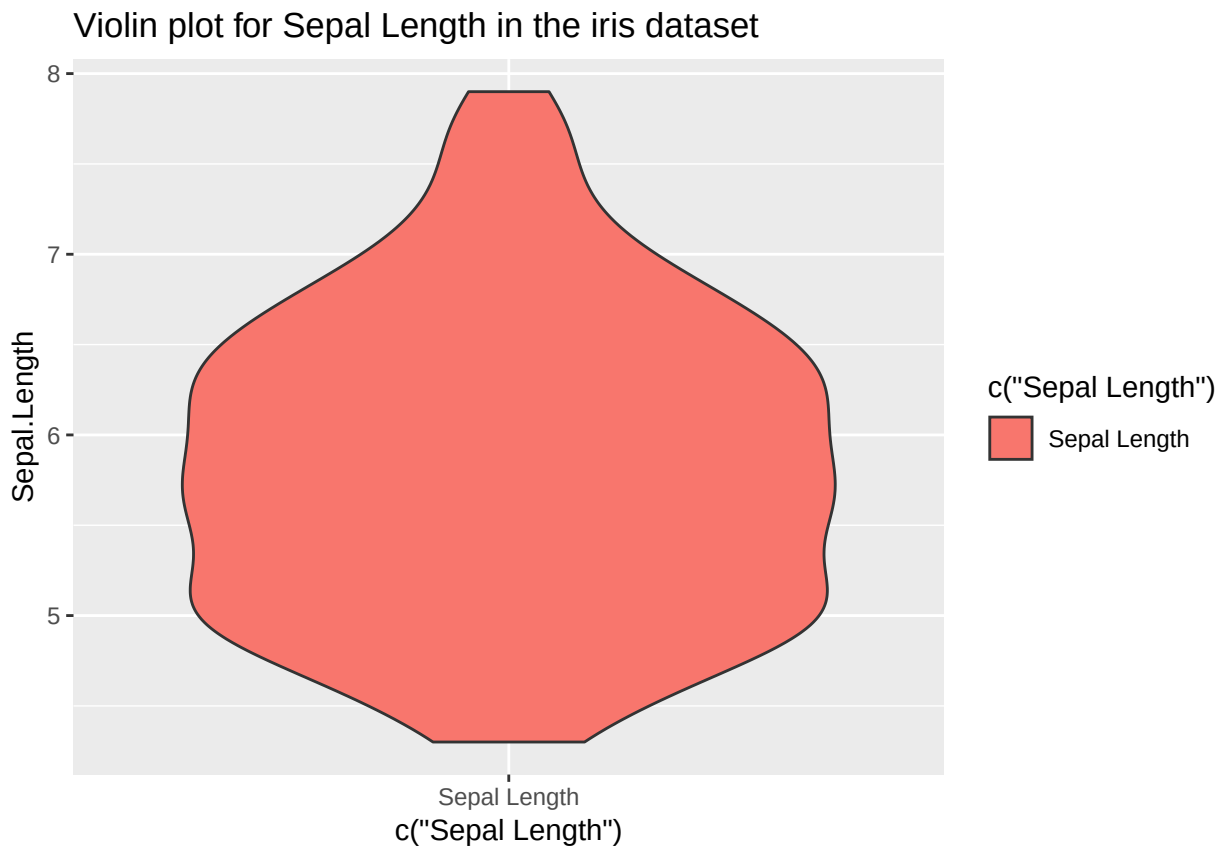
We can also create a box-whisker plot for the data, which indicates the quartiles graphically.

```
boxplot(iris$Sepal.Length)
```



Similarly, a *violin plot* gives a sense of the smoothed density. Base R does not support violin plots, so we will use the *ggplot2* package.

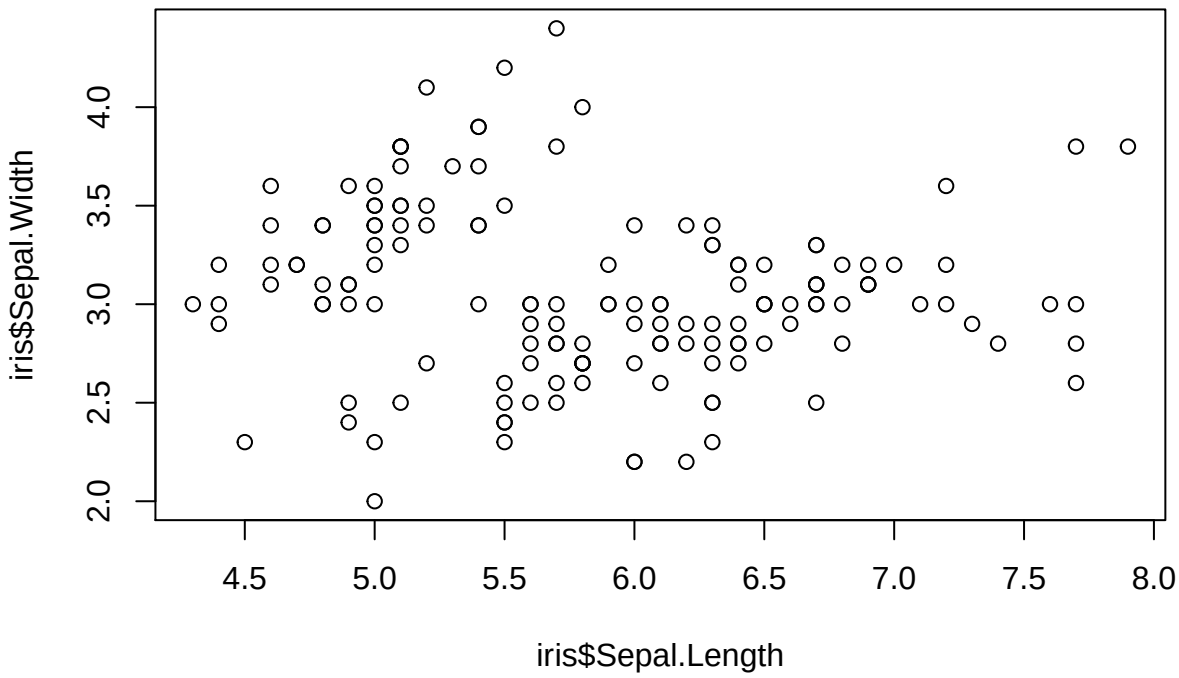
```
library(ggplot2)
ggplot(data=iris,aes(x=c("Sepal Length"),y=Sepal.Length,fill=c("Sepal Length"))) +
  geom_violin() +
  ggtitle("Violin plot for Sepal Length in the iris dataset")
```



Bivariate Graphics

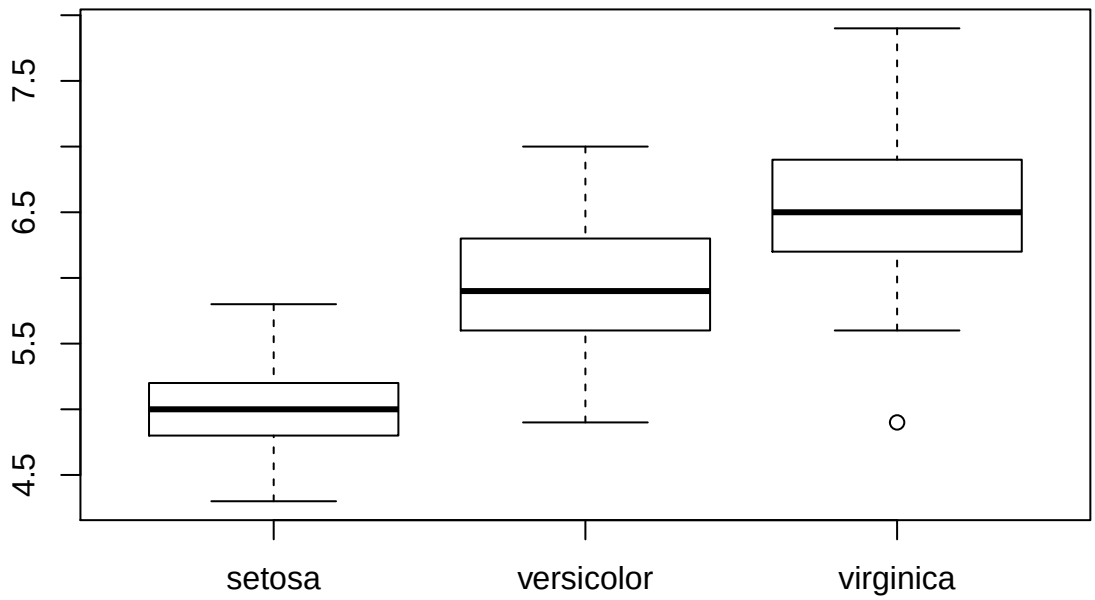
Using the *iris* data again, we can create a scatterplot.

```
data('iris')
plot(iris$Sepal.Length,iris$Sepal.Width)
```

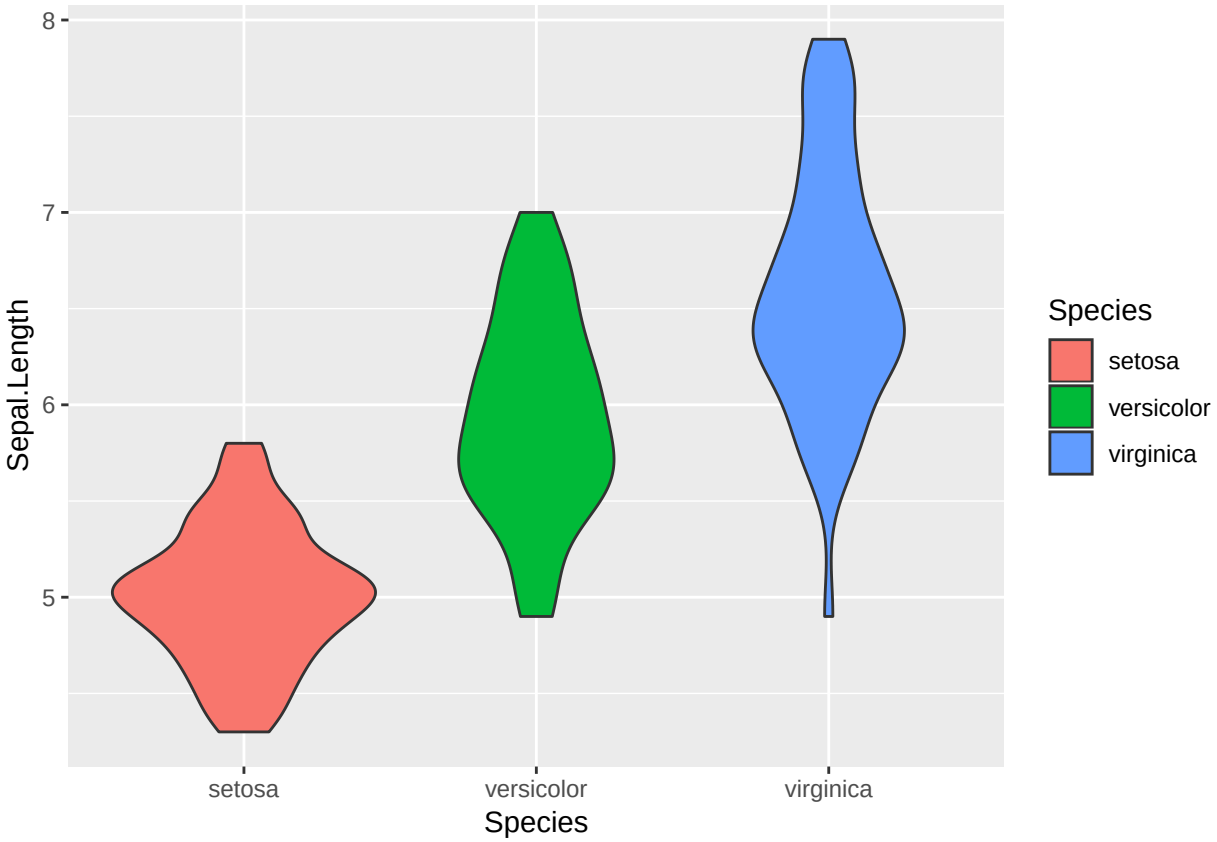


When we are interested in how a measurement varies across categories in a factor, we can use multiple boxplots or violin plots, one for each category.

```
boxplot(Sepal.Length ~ Species,data=iris)
```




```
library(ggplot2)
ggplot(data=iris,aes(x=Species,y=Sepal.Length,fill=Species)) +
  geom_violin()
```



For some examples of using *ggplot2* to create a variety of boxplots and violin plots (and weird hybrids!) visit https://www.data-to-viz.com/caveat/boxplot.html.

And Also. . .

EDA is a rich field and a developing one! You can read more in the following R-oriented books:

- Peter Bruce and Andrew Bruce. **Practical Statistics for Data Science: 50 Essential Concepts**. O'Reilly. Great info, especially if you have less familiarity with statistics.
- Hadley Wickham and Garrett Grolemund. **R for Data Science**. O'Reilly. A great guide to the *tidyverse*, Wickham's system of R packages for data import, manipulation, and representation.