Lecture 7 Graphical Data Analysis

The Great Courses

Today's topic: Graphical Data analysis

Statistical graphs are useful in helping us visualize data. Through graphs, we:

- Understand data properties
- Find patterns in data
- Suggest modeling strategies
- "Debug" our analyses
- Communicate results

Learning Objectives for today

- Define and identify basic numerical and graphical summeries of data
- Use R for calculating descriptive statistics, making graphs, and writing functions

Iris Data

The Iris dataset is widely used throughout statistical science for illustrating various problems in statistical graphics, multivariate statistics and machine learning.

- It's a small, but non-trivial dataset.
- The data values are real (as opposed to simulated) and are of high quality (collected with minimal error).
- The data were used by the celebrated British statistician Ronald Fisher in 1936. (Later he was knighted and became Sir Ronald.)
- Using a few famous datasets is one of the traditions we hand down in statistics! (Also, when comparing old and new methods, or in evaluating any method, it's helpful to try them out on known datasets, thus maintaining continuity in how we assess methods.)

Iris Data

The Iris dataset is most commonly used for on pattern recognition in statistics. The dataset contains 3 classes of 50 instances each, where each class refers to a type of iris plant, with the following attributes:

- Sepal Length
- Sepal Width
- Petal Length
- Petal Width
- class: Iris setosa, Iris versicolor, Iris virginica

Load Data

The Iris data is in the datasets library in R. Type the following commands:

```
library(datasets)
library(RColorBrewer)
attach(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
```

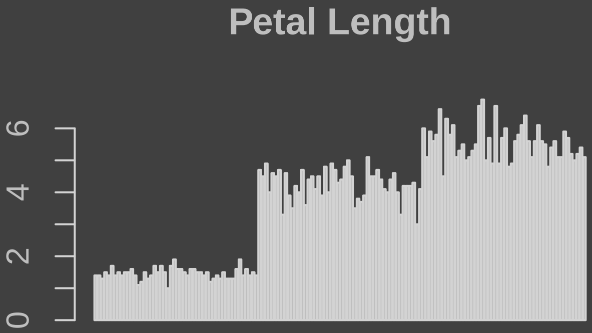
Bar Plots

Let's begin our analysis.

- **Bar Plots** are useful for showing comparisons across several groups. Although it looks like a histogram, a bar plot is plotted over a label that represents a category (e.g., Iris type).
- One indication of the difference between a bar plot and histogram: It's always appropriate to talk about the skewness of a histogram; that is, the tendency of the observations to fall more on the low end or the high end of the X axis.
- However, on bar plots, the X axis can sometimes be categorical - (i.e. not quantitative.)

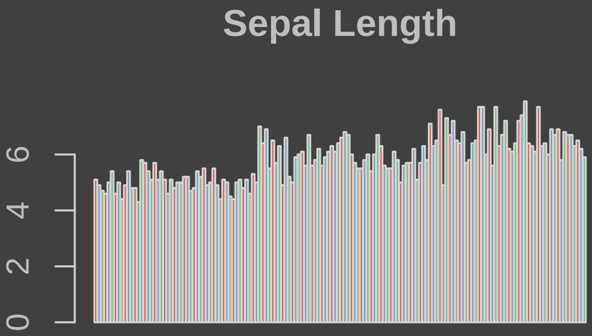
Bar Plots

```
barplot(iris$Petal.Length, main = "Petal Length")
```



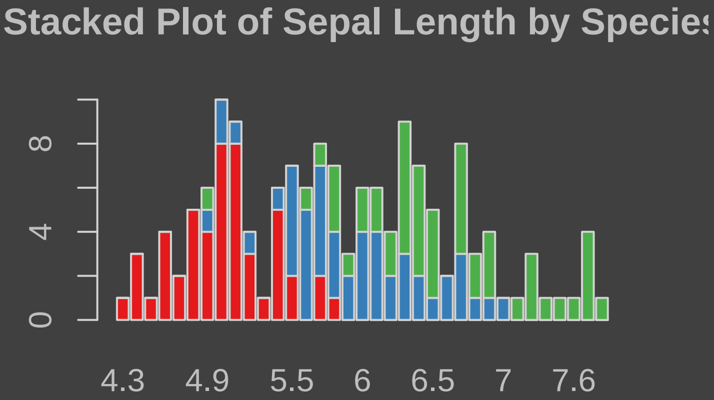
Bar Plot

```
barplot(iris$Sepal.Length,
        col= brewer.pal(3,"Set1"), main = "Sepal Length")
```



Bar Plot

```
barplot(table(iris$Species,iris$Sepal.Length),
        col  = brewer.pal(3,"Set1"),
        main = "Stacked Plot of Sepal Length by Species")
```



Summary Statistics

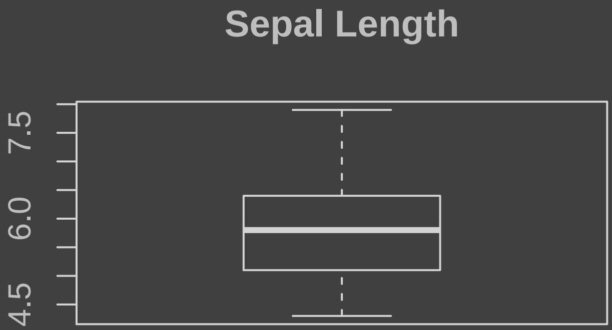
The `summary` function is a quick and easy way to assess the statistical properties of each attribute. These values are displayed graphically in a box plot.

```
summary(iris)
  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width   
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100  
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300  
 Median :5.800   Median :3.000   Median :4.350   Median :1.300  
 Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199  
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800  
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500  
       Species  
 setosa    :50  
 versicolor:50  
 virginica :50  
```

Box Plots

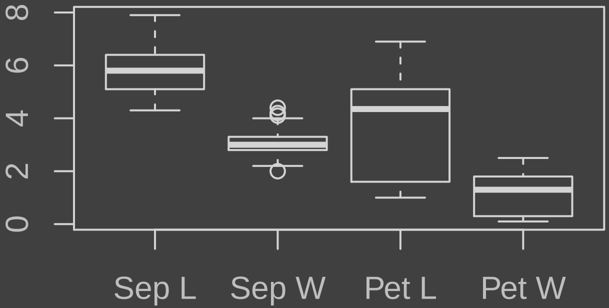
- Box plots are used to compactly show many pieces of information about a variables distribution and is useful for visualizing the spread of the data.
- Box plots show **five statistically important numbers** - the minimum, the 25th percentile, the median, the 75th percentile and the maximum.

```
boxplot(iris$Sepal.Length, main = "Sepal Length")
```



Box Plots

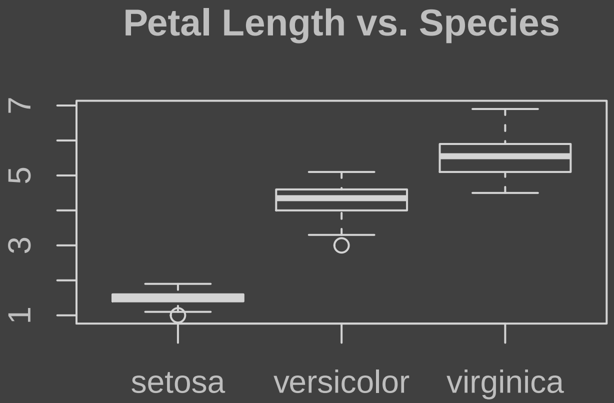
```
boxplot(iris[,1:4],
        names=c("Sep L", "Sep W", "Pet L", "Pet W"))
```



Box Plots

A box plot can also be used to show how one attribute `petal.length` varies with another attribute `iris.type`.

```
boxplot(iris$Petal.Length~iris$Species,
        main = "Petal Length vs. Species")
```

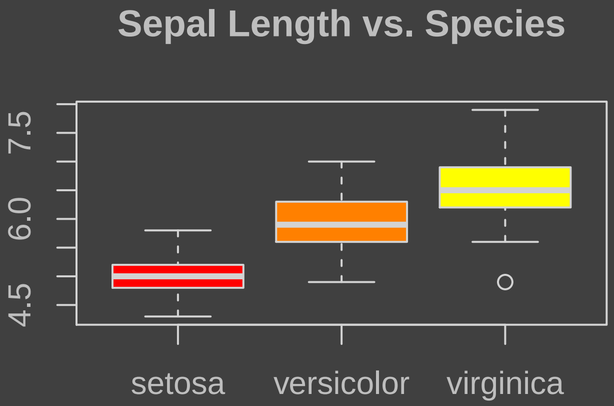


Box Plot

We can visualize how the spread of `Sepal Length` changes across various categories of `Species`. A color palette is a group of colors that is used to make the graph more appealing and help create visual distinctions in the data.

Box Plot

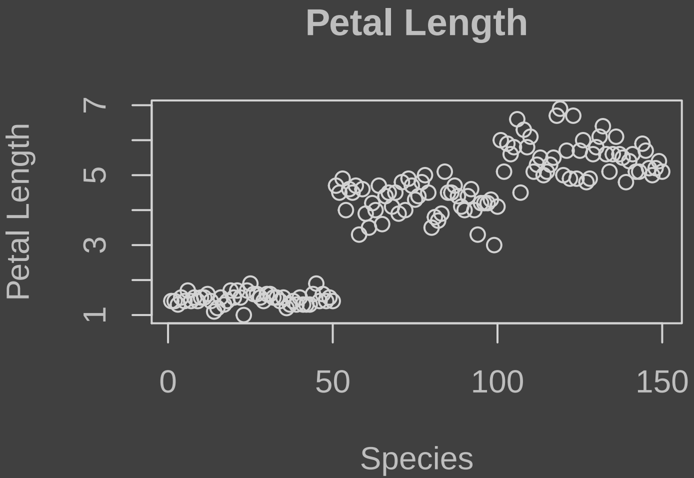
```
boxplot(iris$Sepal.Length~iris$Species,
        col=heat.colors(3),
        main = "Sepal Length vs. Species")
```



Scatter Plot

Scatter plots help in visualizing data easily and for simple data inspection. Try the following code.

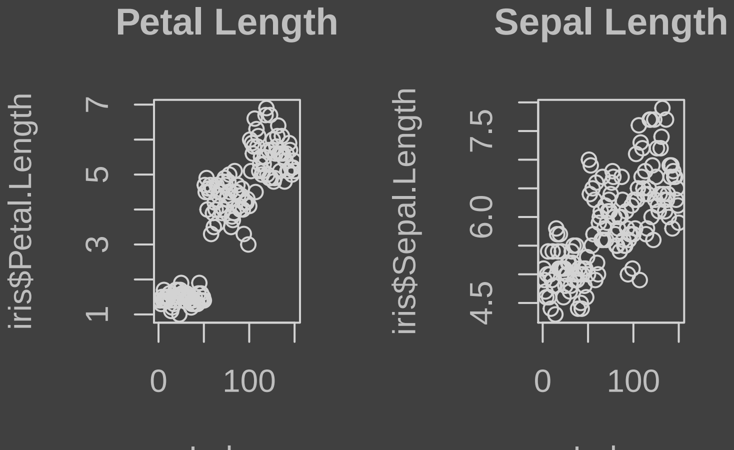
```
plot(iris$Petal.Length, main="Petal Length",
     ylab = "Petal Length", xlab = "Species")
```



Scatter Plot

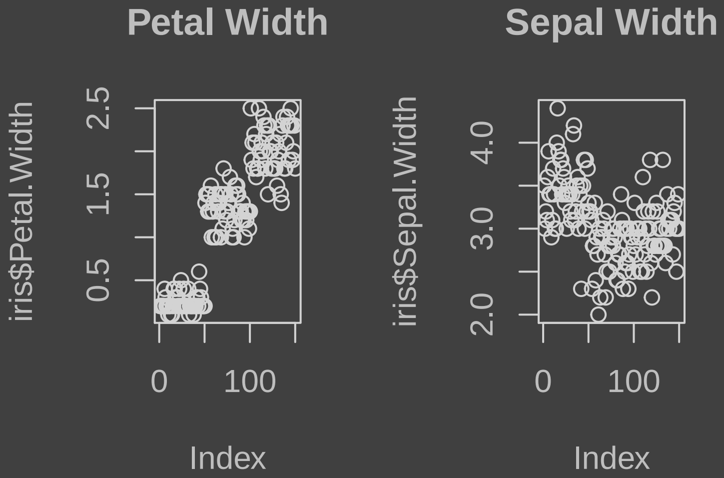
Let's generate corresponding scatterplots for Petal.Width, Sepal.Length and Sepal.Width.

```
par(mfrow=c(1,2))
plot(iris$Petal.Length, main="Petal Length")
plot(iris$Sepal.Length, main="Sepal Length")
```



Scatter Plot

```
par(mfrow=c(1,2))
plot(iris$Petal.Width, main="Petal Width")
plot(iris$Sepal.Width, main="Sepal Width")
```



What are our observations? Which plots help us distinguish between

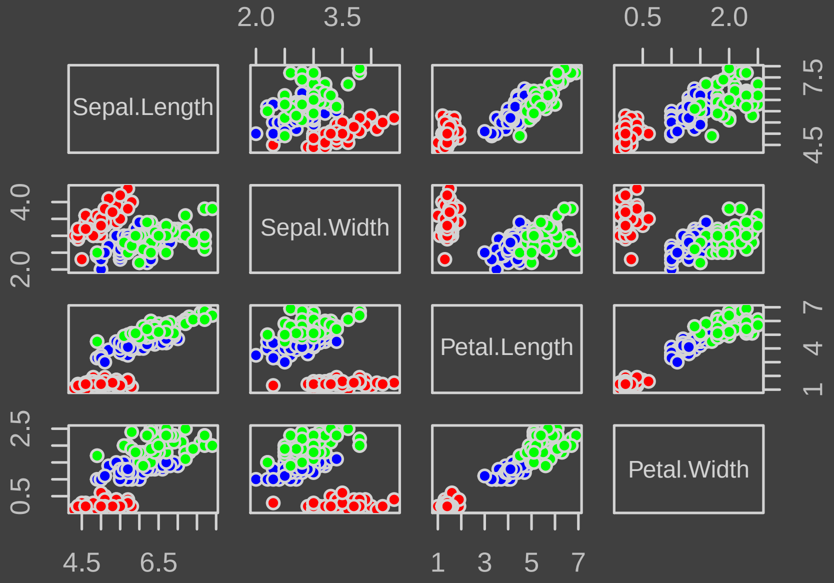
Scatter Plots

- Scatter plots are used to plot two variables against each other. We can add a third dimension by coloring the data values according to their Species.
- For datasets with only a few attributes, we can construct and view all the pairwise scatter plots.

Pairwise Plots

```
pairs(as.matrix(iris[,-5]), pch=21, bg=c("red", "blue", "green")
```

Pairwise Plots



Pairwise Plots

- Since the upper and lower graphs are duplicates of each other, we can augment our code to display the **correlation** between our variables in the upper level.
- The correlation measures the strength of the relationship between two random variables.
- Correlations range from -1 to 1, where:

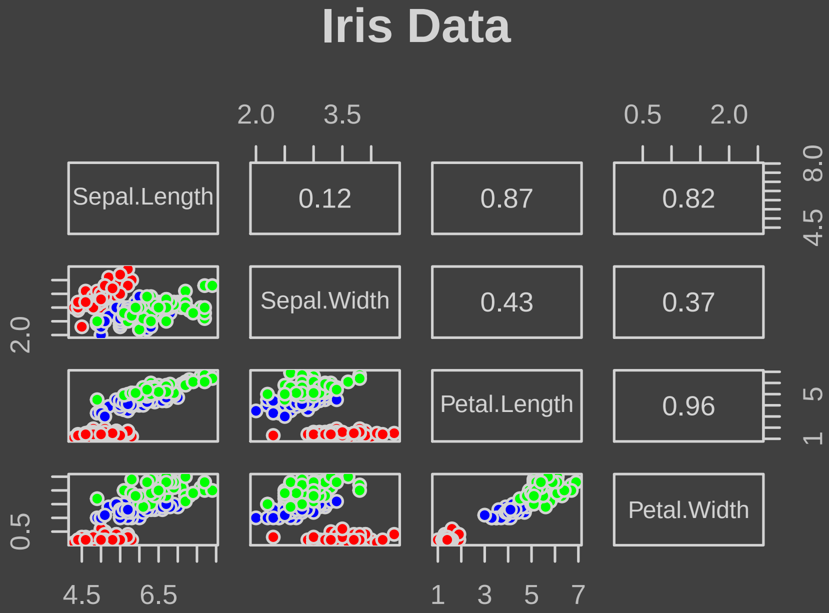
– Values near 1 indicate a strong positive relationship
– Values near -1 indicate a strong negative relationship
– Values near 0 indicate no relationship.

Pairwise Plots

```
panel.pearson <- function(x, y, ...) {
  horizontal <- (par("usr")[1] + par("usr")[2]) / 2;
  vertical <- (par("usr")[3] + par("usr")[4]) / 2;
  text(horizontal, vertical, format(abs(cor(x,y)), digits=2))}

pairs(as.matrix(iris[1:4]), main = "Iris Data", pch = 21,
      bg = c("red","blue", "green")[unclass(iris$Species)],
      upper.panel=panel.pearson)
```

Pairwise Plots



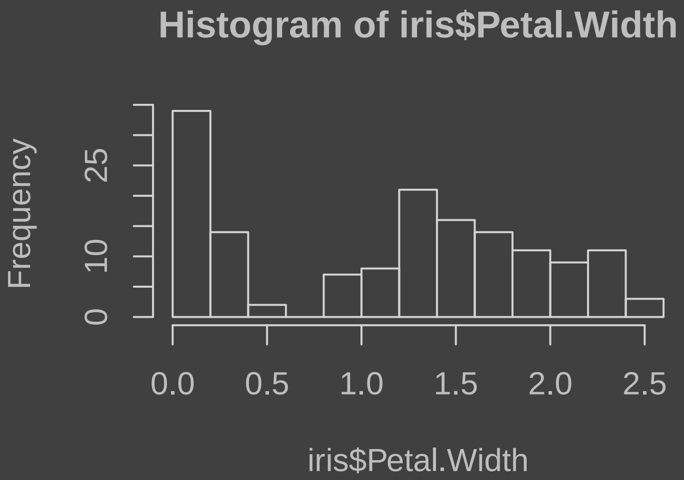
Histograms

- A **Histogram** is a plot that breaks the data into bins (or breaks) and shows the frequency distribution of those bins.
- We can change the breaks to see the effect it has data visualization.

Histograms

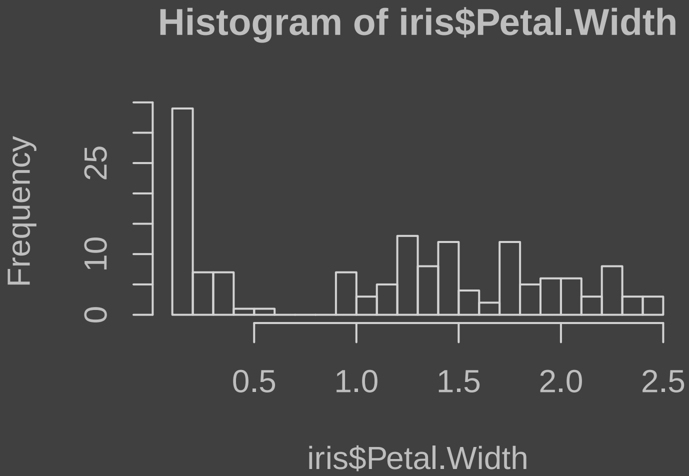
Let's create some histograms of our Iris data. The number of bins in the histogram is variable.

```
hist(iris$Petal.Width, breaks=13)
```



Histograms

```
hist(iris$Petal.Width, breaks=25)
```

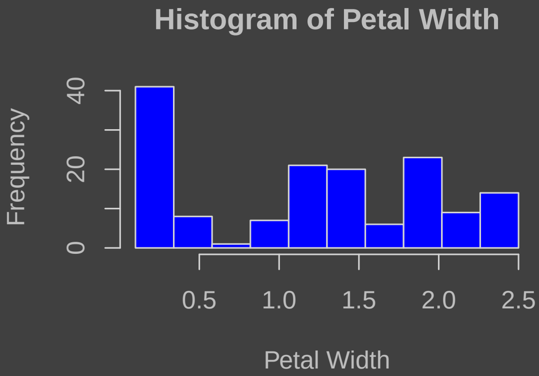


Histograms

We can create custom break points by defining a sequence vector, `b`, that ranges from `min(iris$Petal.Width)` to the `max(iris$Petal.Width)` with a specified number of breaks.

```
b <- seq(min(iris$Petal.Width),
         max(iris$Petal.Width), length=11)
b
hist(iris$Petal.Width, breaks=b,
     xlab="Petal Width", main="Histogram of Petal Width")
```

Histograms

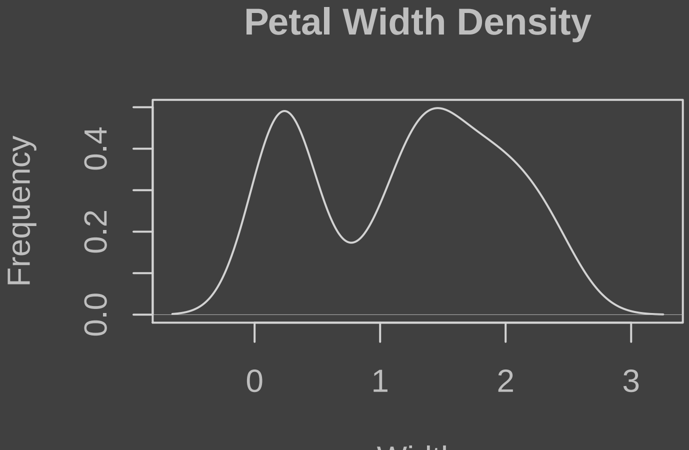


Histogram of Petal Width
Frequency
0
20
40
0.5
1.0
1.5
2.0
2.5
Petal Width

Density Plots

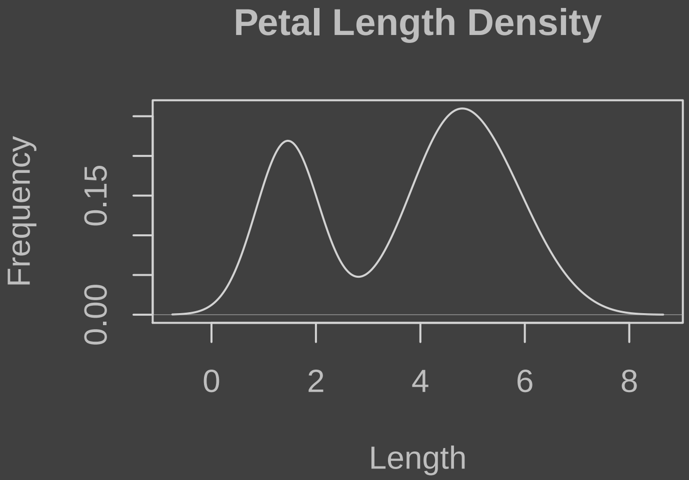
Density plots can be viewed as smoothed versions of a histogram.
We can estimate the density using R's `density` function

```
dens.pw = density(iris$Petal.Width)
plot(dens.pw, ylab = "Frequency", xlab = "Width",
     main= "Petal Width Density")
```



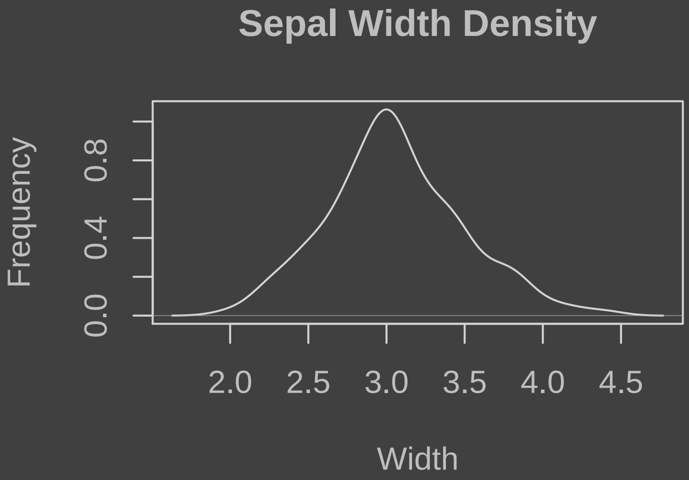
Density Plots

```
dens.pl = density(iris$Petal.Length)
plot(dens.pl, ylab = "Frequency", xlab = "Length",
     main= "Petal Length Density")
```



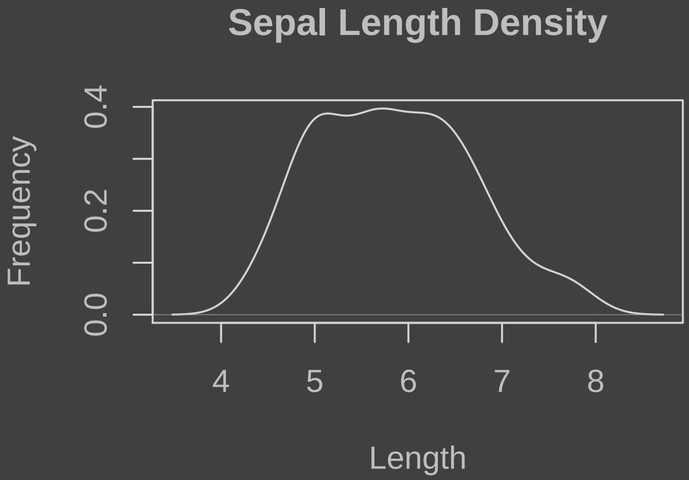
Density Plots

```
dens.sw = density(iris$Sepal.Width)
plot(dens.sw, ylab = "Frequency", xlab = "Width",
     main= "Sepal Width Density")
```



Density Plots

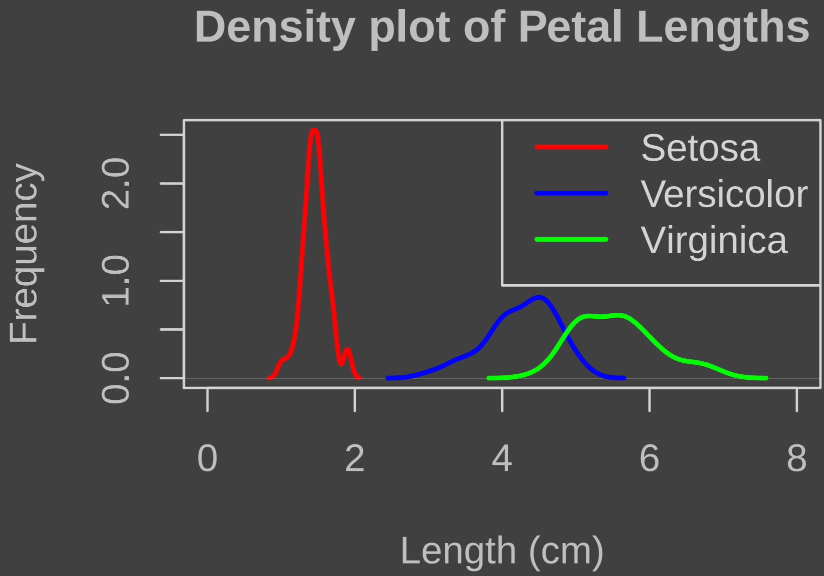
```
dens.sl = density(iris$Sepal.Length)
plot(dens.sl, ylab = "Frequency", xlab = "Length",
     main= "Sepal Length Density")
```



Density Plots

Let's also look at the density function of `Petal.Length` for each of the three classes of irises.

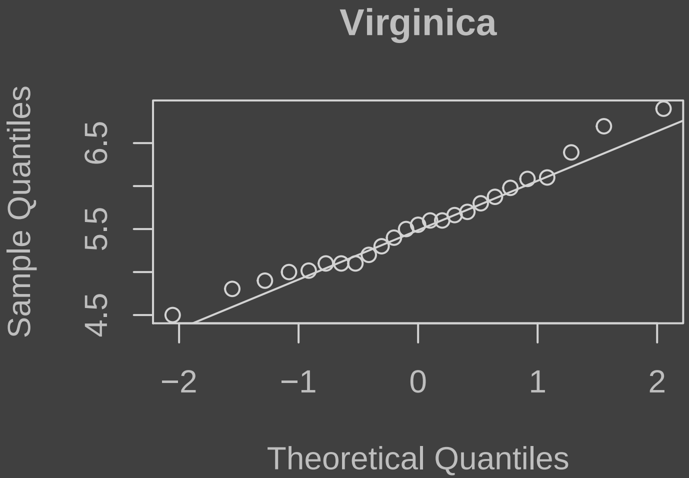
Density Plots



Quantile Plots

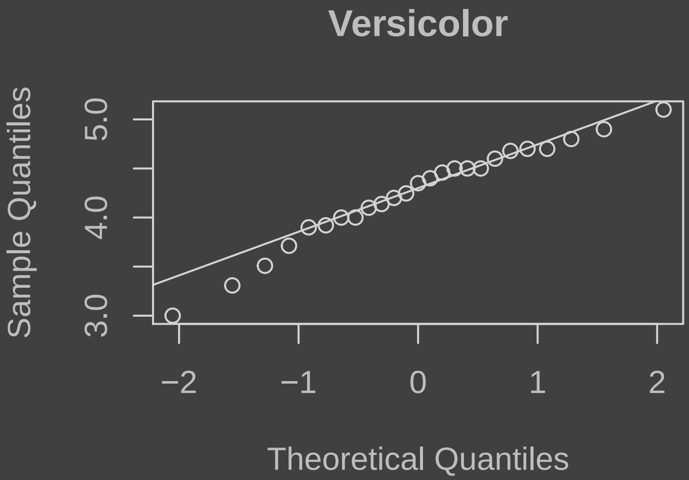
We can calculate the quantiles of the iris dataset to compare them to those of a normal distribution.

```
qqnorm(quantile.virginica, main="Virginica")
qqline(quantile.virginica)
```



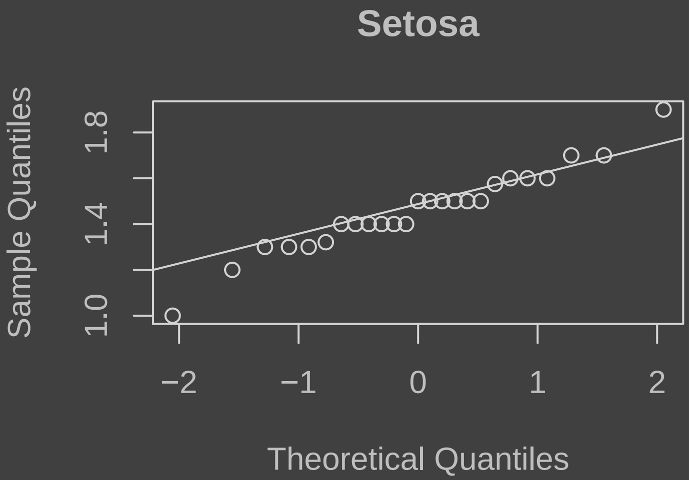
Quantile Plots

```
qqnorm(quantile.versicolor, main="Versicolor")
qqline(quantile.versicolor)
```



Quantile Plots

```
qqnorm(quantile.setosa, main="Setosa")
qqline(quantile.setosa)
```



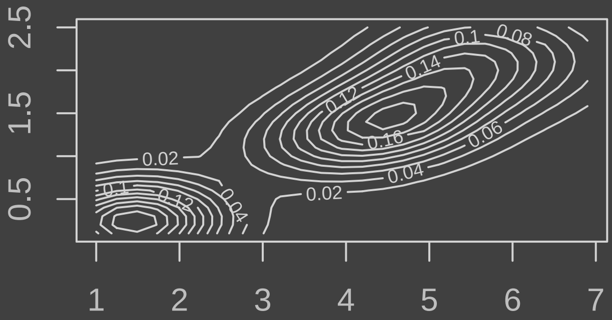
Contour Plots

Density estimation is available for higher dimensional data using Contour plots.

- A **contour plot** is a graph that explores the potential relationship among three variables.
- Contour plots display the 3-dimensional relationship in two dimensions, with x and y variables plotted on the x and y scales and the z variable represented by contours.
- A contour plot is like a topographical map in which x, y, and z values are plotted instead of longitude, latitude, and elevation.

Contour Plots

```
library(MASS)
petal.dens = kde2d(iris$Petal.Length, iris$Petal.Width)
contour(petal.dens)
```



Contour Plots

The plot may also be viewed as a heatmap, with brighter colors denoting higher values.

```
image(petal.dens)
```