Scatter Plot

Ryan Dean, Andrew Slavetskas

A scatter plot or scattergraph is a type of mathematical diagram using Cartesian coordinates to display values for two variables for a set of data.

The data is displayed as a collection of points, each having the value of one variable determining the position on the horizontal axis and the value of the other variable determining the position on the vertical axis.

The following is a brief tutorial regarding scatter-plots in R

The generic form used for scatter plots is *plot(x,y…)*

```
 # Scatter Plot Tutorial

attach(iris)
```

We attached the iris data set(already stored in R) so that we call up its values.

```
# Plot Petal length with regards to width
plot(Petal.Length, Petal.Width, pch = 16, xlab = "P Lenth", ylab = "P Width", main = "TITLE")
```

In this case we plotted Petal Length versus Petal Width in the data set of iris as the x/y of our plot. "pch" refers to specific characters/symbols that can be used in the plot. It is a 1x25 matrix of plot symbols. For example, pch=19 is a solid circle whereas pch=23 is a filled diamond. In addition, labels can be applied to the plot. For example:
xlab="" **names the x-axis**
ylab="" **names the y-axis**
main="" **names the overall plot**
sub="" **gives a subtitle to the plot**
xlim = c(x,y) gives the limitations on the x axis (sets the range of the x axis)
ylim = c(x,y) gives the limitations on the y axis (sets the range of the y axis)

```
# Add regression
abline(lm(Petal.Width ~ Petal.Length))
```

In many cases, it is useful to add a regression line to see the best fit line for the particular plot.

```
# Plot points of different colors based on value
# pch is 1x25 matrix of plot symbols
plot(Petal.Length, Petal.Width, pch = 16, col = ifelse(Petal.Length < 2, "red", "blue"))
plot(Petal.Length, Petal.Width, pch = 16, col = ifelse(Petal.Length < 2, "red", ifelse(Petal.Length < 5,
"blue", "yellow")),  xlab = "P Lenth", ylab = "P Width", main = "TITLE")
```

Color is another useful option to distinguish different points on a plot. The function *col=””* gives particular points a specific color. For example, *col=”blue”* would give the plot points the color blue. One can explicitly name a color of their choosing (one that is available in the color chart of R).

To further distinguish specific plot points, one can use the *ifelse* function. The generic form of this function is:
ifelse(test, yes, no)
where the test, an object can be made into a logical mode, “yes” returns values for true elements of the test and “no” returns values for false elements of the test. Using this, we can combine this with other functions, such as with color. We can say if the Petal Length is less than 2, make the plot points red, otherwise, make them a different color. We can also combine the *ifelse* function with the *ifelse* function to further distinguish plot points.

```
plot(Petal.Length, Petal.Width, pch = 16, col = ifelse(Species == "setosa", "red", ifelse(Species == "virginica", "yellow", "blue")),  xlab = "P Lenth", ylab = "P Width", main = "TITLE")
```
Returning to the *ifelse* function we can, instead of distinguishing based on numerical values, we can distinguish based on different species.

```
# Add regression line line to fit through cloud of points
abline(lm(Petal.Width ~ Petal.Length), col = "red")
```
In many cases, it is useful to add a regression line to see the best fit line for the particular plot. Again, it may be useful to add color to the regression line and we can do so by using the *col=””* function
The function to add a straight line through a set of points is:
Abline() **and where *lm* is used to fit linear models.**

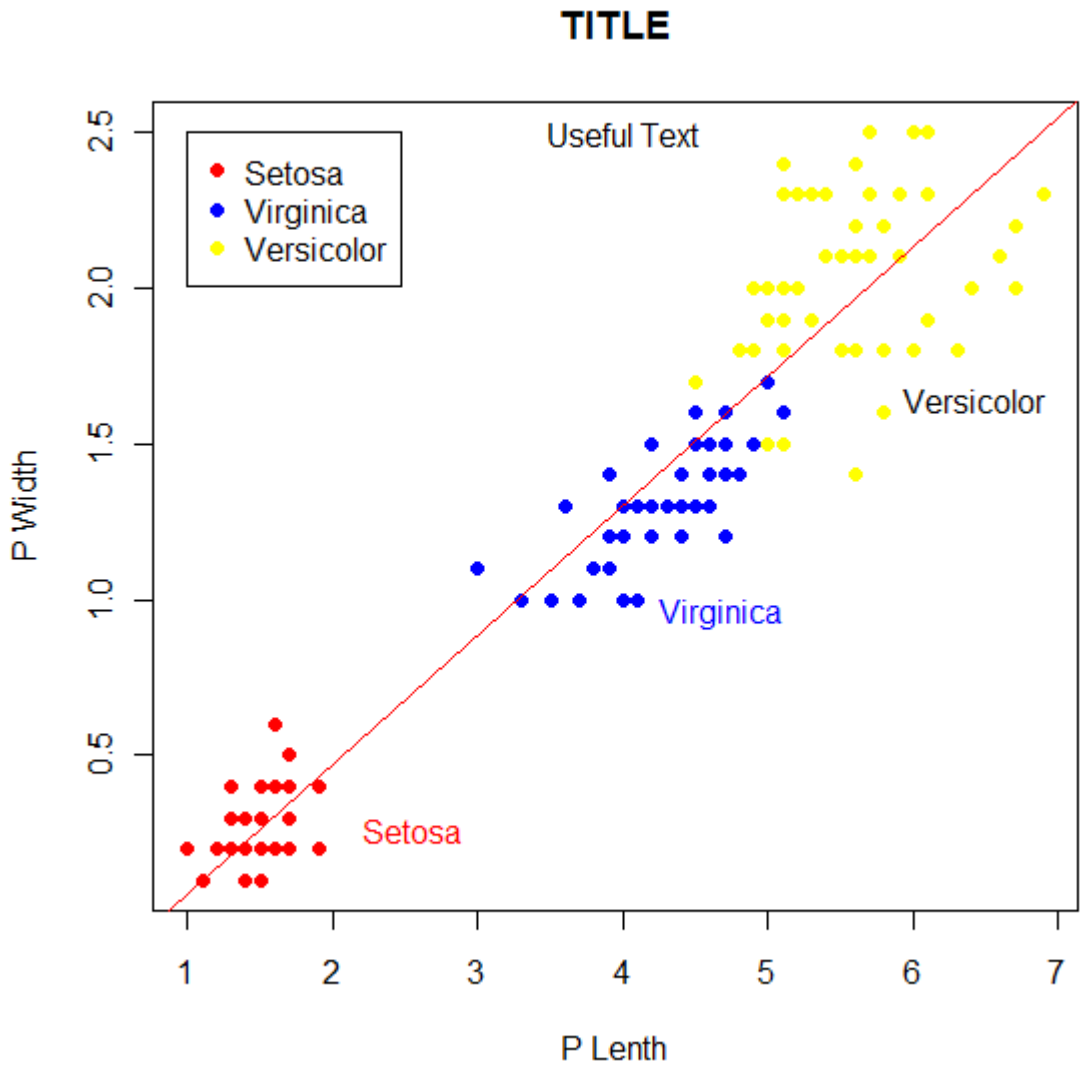
```
#Add Legend to plot
legend(1, max(Petal.Width), c("Setosa", "Virginica", "Versicolor"), col = c("red", "blue", "yellow"), pch = 16)
```
It is important to make sure that the plot is labeled correctly so that the reader knows what points relate to what data. A legend makes this possible. Legend(x, y, etc.) puts a legend at a certain point on the graph, in this case at 1 and the max of Petal Width (top left corner). The other parameters give color representations and point characteristics to match the plot.

```
#Add text to plot
text(4, 2.5, "Useful Text")
text(locator(3), c("Setosa", "Virginica", "Versicolor"), col = c("red", "blue", "black"))
```
Another way to label a graph is to add text right in the middle of the plot. This is done using the text command. Here, locator(3) is used to add text to a point where the mouse is clicked after the plot is

produced. This gives the ability for the user to add labels to certain areas of the plot (can label each species next to their respective areas on the plot).

detach(iris)

Clean up the script. Never want to keep data up in RAM if it's not necessary



Prototype in R:

 # Scatter Plot Tutorial

attach(iris)

 # Plot Petal length with regards to width

```
plot(Petal.Length, Petal.Width, pch = 16, col = ifelse(Species == "setosa", "red", ifelse(Species == "virginica", "yellow", "blue")),  xlab = "P Lenth", ylab = "P Width", main = "TITLE")


# Add regression line
abline(lm(Petal.Width ~ Petal.Length), col = "red")

#Add Legend to plot
legend(1, max(Petal.Width), c("Setosa", "Virginica", "Versicolor"), col = c("red", "blue", "yellow"), pch = 16)

#Add text to plot
text(4, 2.5, "Useful Text")
text(locator(3), c("Setosa", "Virginica", "Versicolor"), col = c("red", "blue", "black"))

detach(iris)
```