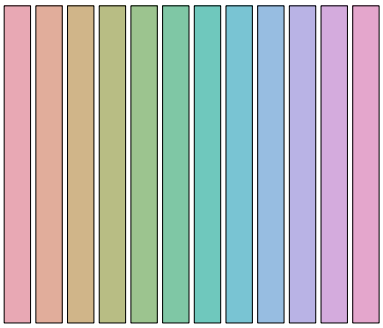
Presentation Graphics



Presentation Graphics

S has a large number of plots which can be used to present information. Usually such plots are banished to lower level statistics courses, but we'll revisit them, because they are the most commonly used graphical displays.

The R variants of these plots are typically more flexible (particularly when it comes to color) and in this lecture, I'll stick entirely to using R.

Pie Charts

Pie charts show visually how a total can be broken down into its component parts. They do this by dividing a circle into segments, with angle of the segment proportional to contribution of the corresponding value to the total.

A typical use of a pie chart is to show data such as the following; which shows the division of New Zealand's expenditure on income support in the 1994–1995 year.

Superannuation	53%
Unemployment	14%
Domestic Purposes	14%
Sickness	8%
Others	12%

Color

Most simple statistical displays can be greatly enhanced by using color. Choosing colors which work together well is a hard task.

The following set of colors do work together well. (This is hot-off-the press research). The colors are 12 hues are equally spaced around a color wheel. The colors are constructed to have equal visual impact.

```
> pal <- c("#E8A8B4", "#E1AD9B", "#D0B589",
+          "#B8BD84", "#9CC48F", "#7FC7A5",
+          "#6FC8BD", "#79C4D3", "#97BDE1",
+          "#B9B3E5", "#D4ABDD", "#E4A6CC")
```

Drawing A Pie Chart With R

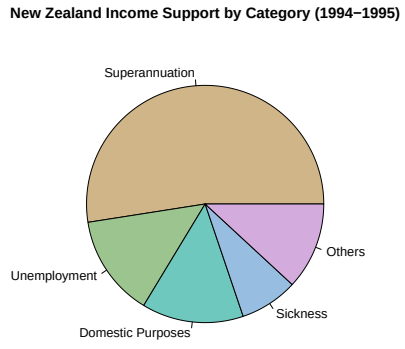
In R the function **`pie`** (or in older versions **`piechart`**) can be used to draw pie charts.

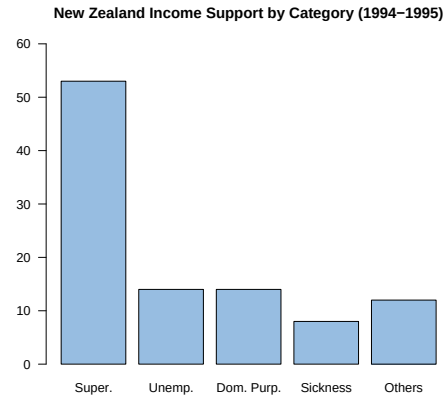
The example shows how a pie chart is produced for a named vector of values. It is also possible to supply the names in a separate argument.

```
> expend <- c(53, 14, 14, 8, 12)
> names(expend) <- c("Superannuation",
+                    "Unemployment",
+                    "Domestic Purposes",
+                    "Sickness", "Others")

> pie(expend, col = pal[c(3,5,7,9,11)],
```

The colors all have equal *luminance* (brightness) and *chroma* (colorfulness).

```
+   main = paste("New Zealand Income Support",
+                "by Category (1994-1995)"))
```





Bar Charts

Bar charts are a better way of displaying data than pie charts.

Barcharts encode their values as the heights of bars. This makes it easier to decode and compare the values presented.

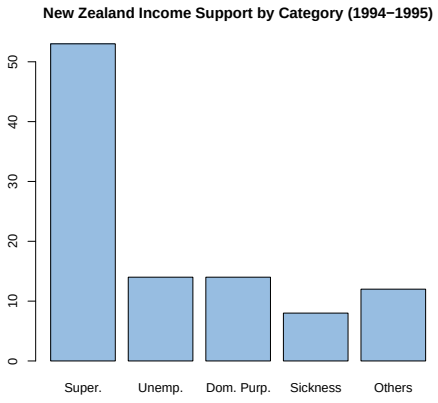
```
> expend <- c(53, 14, 14, 8, 12)
> names(expend) <- c("Super.", "Unemp.",
+                    "Dom. Purp.", "Sickness",
+                    "Others")

> barplot(expend, col = pal[9],
+   main = paste("New Zealand Income Support",
+                  "by Category (1994-1995)"))
```

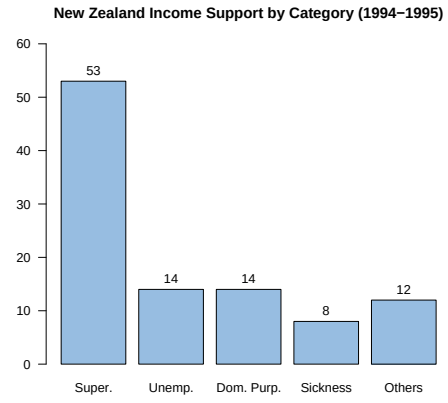
Annotating Bar Heights

Barplot returns a vector which gives the *x* coordinates of the midpoints of each bar. This can be used to place values at the top of each bar.

```
> xmid <-
+  barplot(
+   expend, col=pal[9],
+   las=1, ylim=c(0,60),
+   main = paste("New Zealand Income Support",
+                "by Category (1994-1995)")
+ )
> text(xmid, expend+2, expend)
```

A similar technique can be used to draw “error bars” at the top of each bar (to indicate variability).





Improved Bar Charts

The standard bar charts produced by R have a number of shortcomings.

- They have rotated tick mark labels on the *y* axis.
- The *y* axis can be shorter than the tallest bar in the plot.

The standard plots can be improved by making all tick mark labels horizontal and (sometimes) by extending the *y* axis.

```
> barplot(expend, col=pal[9],
+   las=1, ylim=c(0,60),
+   main = paste("New Zealand Income Support",
+                  "by Category (1994-1995)"))
```

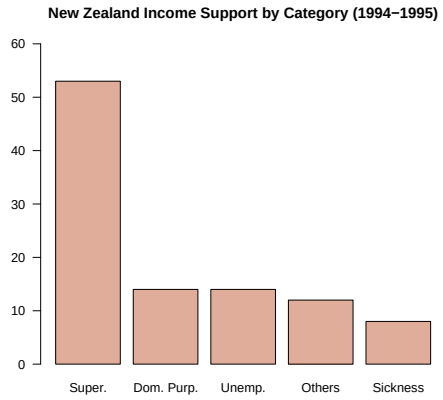
[I’ve shortened the labels here so that they will fit.]

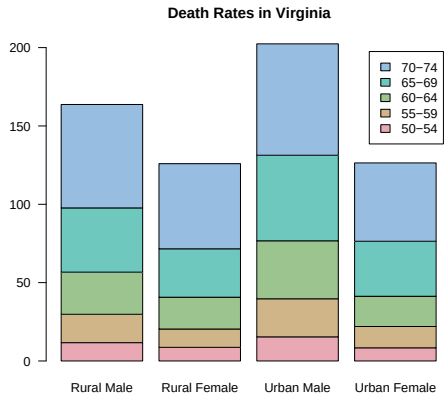
Pareto Charts

A Pareto chart is a variant of bar charts where the bars are drawn in descending order of height.

This kind of plot is used to order the faults ocurring in a manufacturing process in order of “importance.”

```
> barplot(rev(sort(expend)),
+   col=pal[2], las = 1, ylim=c(0, 60),
+   main = paste("New Zealand Income Support",
+                "by Category (1994-1995)")
+ )
```





Two-Way Barcharts

Bar charts can be used to plot grouped data.

The data set **`VADeaths`** gives death rates per 100,000 for the state of Virginia in the USA.

The values are broken down by gender/location and age goup.

	Rural		*Urban*	
	Male	*Female*	*Male*	*Female*
50–54	11.7	8.7	15.4	8.4
55–59	18.1	11.7	24.3	13.6
65–69	41.0	30.9	54.6	35.1
70–74	66.0	54.3	71.1	50.0

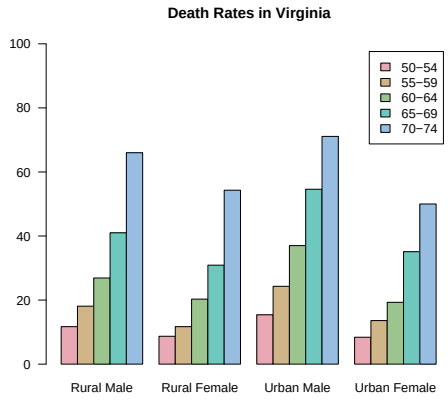
Side-By-Side Bars

The optional argument **`beside=TRUE`** causes the values in the columns to appear in an unstacked form.

A legend is still helpful. In this case we must enlarge the *y* scale to fit it in.

```
> barplot(VADeaths, beside = TRUE,
+        col = pal[c(1,3,5,7,9)], las = 1,
+        legend = rownames(VADeaths),
+        ylim = c(0, 100))
> title(main = "Death Rates in Virginia")
```

There are two ways of producing bar charts from a matrix like this.



Stacked Bars

When barplot is given a matrix as an argument it produces a "stacked barplot."

In this case the observations in each column of the matrix are combined to form one bar.

It helps to provide a legend which labels the divisions within each bar.

```
> barplot(VADeaths,
+        col = pal[c(1,3,5,7,9)], las = 1,
+        legend = rownames(VADeaths))
> title(main = "Death Rates in Virginia")
```

The Legend Function

Finer control over legend placement can be obtained with the **`legend`** function. The following example shows the legend

placed at the top center of the graph.

```
> xm <- barplot(VADeaths, beside = TRUE,
+              col = pal[c(1,3,5,7,9)],
+              las = 1, ylim = c(0, 100))

> legend(mean(xm), 100,
+        legend=rev(rownames(VADeaths)),
+        xjust = 0.5,
+        fill = rev(pal[c(1,3,5,7,9)]))
```

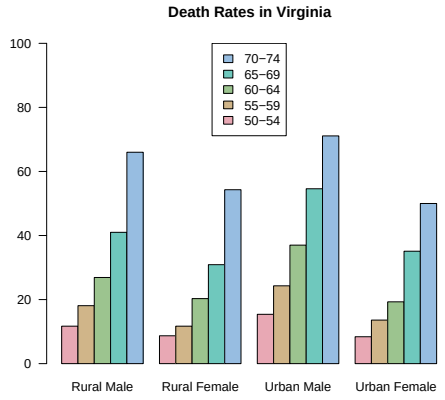
```
> title(main = "Death Rates in Virginia")
```

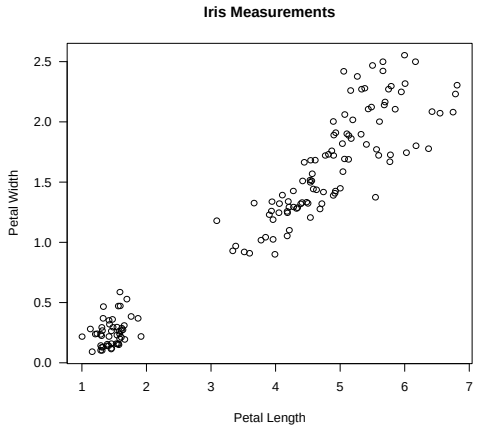
Improving Scatter Plots

The plot can be improved by rotating the *y* axis tick mark labels (there is enough space) and by jittering the values to avoid overplotting.

Here we add a random amount which is uniformly distributed on the interval $[-.1, .1]$ to separate the points.

```
> plot(jitter(PetalLength,amount=.1),
+      jitter(PetalWidth,amount=.1),
+      xlab = "Petal Length",
+      ylab = "Petal Width",
+      main = "Iris Measurements",
+      las = 1)
```





Scatter Plots

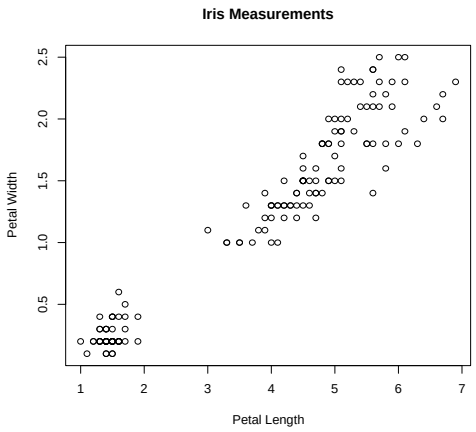
The plot function does a good job of producing scatter plots. Suppose that the variables **`PetalLength`** and **`PetalWidth`** contain two of the variables in the Anderson Iris data set, and that **`Species`** records the species.

```
> plot(PetalLength, PetalWidth,
+      xlab = "Petal Length",
+      ylab = "Petal Width",
+      main = "Iris Measurements")
```

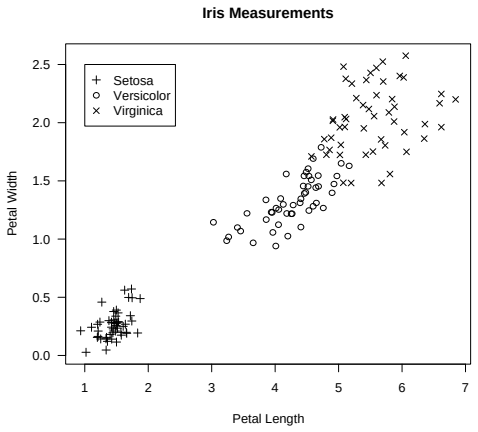
Plotting Symbols

Using different plotting symbols can help to separate species. There are 25 symbols which you can choose from.

```
> plot(jitter(PetalLength,amount=.1),
+      jitter(PetalWidth,amount=.1),
+      pch = c(3,1,4)[Species],
+      xlab = "Petal Length",
+      ylab = "Petal Width",
+      main = "Iris Measurements",
+      las = 1)

> legend(1, 2.5, pch = c(3,1,4),
+        legend=c("Setosa",
+                 "Versicolor",
```




```
+                 "Virginica"))
```



Exotic Plotting Symbols

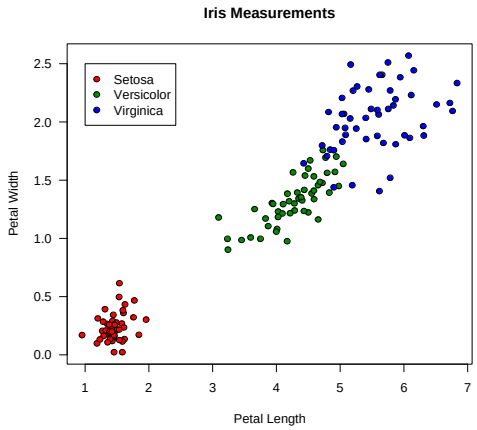
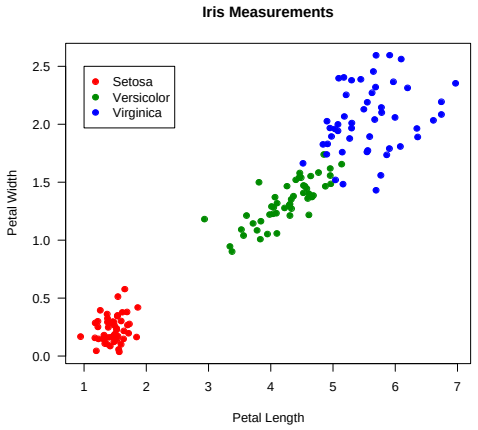
In R, the plotting symbols from 21–25 can have different border and fill colors. This can be help make the points look crisper.

```
> plot(jitter(PetalLength,amount=.1),
+      jitter(PetalWidth,amount=.1),
+      pch = 21, las = 1,
+      bg = c("red","green4","blue")[Species],
+      xlab = "Petal Length",
+      ylab = "Petal Width",
+      main = "Iris Measurements")

> legend(1, 2.5, pch = 21,
+        pt.bg = c("red","green4","blue"),
```

Plotting Symbols And Colour

Colour can be a big help in scatter plots. The plotting symbol numbered 19 is a filled circle, and makes a good coloured symbol.

```
> plot(jitter(PetalLength,amount=.1),
+      jitter(PetalWidth,amount=.1),
+      pch = 19, las = 1,
+      col = c("red","green4","blue")[Species],
+      xlab = "Petal Length",
+      ylab = "Petal Width",
+      main = "Iris Measurements")

> legend(1, 2.5, pch = 19,
+        col = c("red","green4","blue"),
```

```
+        legend=c("Setosa", "Versicolor",
+                 "Virginica"))
```

```
+        legend=c("Setosa", "Versicolor",
+                 "Virginica"))
```