```
ACT        4 453  28.42   4.69     29   28.63   4.45  15  36    21 0.22
SATV       5 453 610.66 112.31    620  617.91 103.78 200 800   600 5.28
SATQ       6 442 596.00 113.07    600  602.21 133.43 200 800   600 5.38
```

6 Day 2: Graphical data displays and Exploratory Data Analysis

A compelling reason to use R is for its graphics capabilities. There are at least three different graphics options available, only one of which, base graphics will be discussed here. The others are *lattice* graphics and *ggobi* (based upon the grammar of graphics Wilkinson (2005).

Although not all threats to inference can be detected graphically, one of the most powerful statistical tests for non-linearity and outliers is the well known but not often used "inter-occular trauma test". A classic example of the need to examine one's data for the effect of non-linearity and the effect of outliers is the data set of Anscombe (1973) which is included as the `data(anscombe)` data set. The data set is striking for it shows four patterns of results, with equal regressions and equal descriptive statistics. The graphs differ drastically in appearance for one actually has a curvilinear relationship, two have one extreme score, and one shows the expected pattern. Anscombe's discussion of the importance of graphs is just as timely now as it was 35 years ago:

> Graphs can have various purposes, such as (i) to help us perceive and appreciate some broad features of the data, (ii) to let us look behind these broad features and see what else is there. Most kinds of statistical calculaton rest on assumptions about the behavior of the data. Those assumptions may be false, and the calculations may be misleading. We ought always to try to check whether the assumptions are reasonably correct; and if they are wrong we ought to be able to perceive in what ways the are wrong. Graphs are very valuable for these purposes. (Anscombe, 1973, p 17).

6.1 The Scatter Plot Matrix (SPLOM)

The problem with suggesting looking at scatter plots of the data is the number of such plots grows by the square of the number of variables. A solution is the scatter plot matrix (SPLOM) available in the `pairs.panels` function which is based upon the `pairs` function. `pairs.panels` show the all the pairwise relationships, as well as histograms of the individual variables. Additional output includes the *Pearson Product Moment Correlation Coefficient*, the locally weighted polynomial regression (LOWESS), and a density curve for

each variable (Figure 5). This kind of graph is particularly useful for less than about 10 variables. Students in an introductory methods course do not seem to realize that this is unusual way of plotting data.

```
> data(sat.act)
> pairs.panels(sat.act)
```

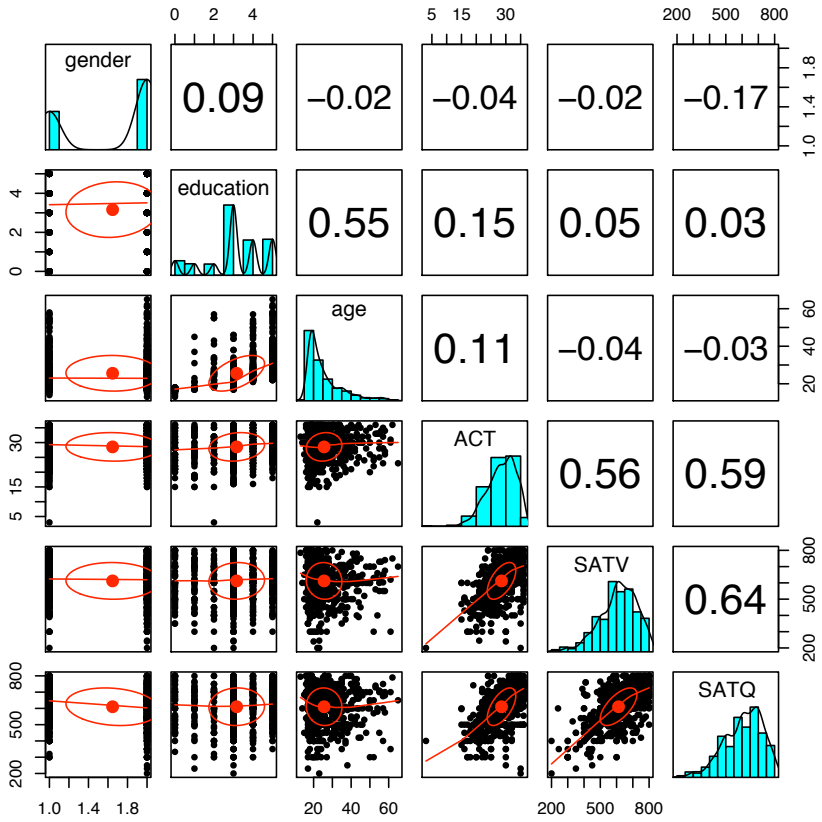

Figure 5: A SPLOM plot is a fast way of detecting non-linearities in the pair wise correlations as well as problems with distributions. For each cell below the diagonal, the x axis reflects the column variable, the y axis, the row variable.

6.2 Bars vs. Boxes

Many psychological graphs report means by using "bar graphs". These are particularly uninformative, for they carry no information about the amount of variability. Some then

```
> data(sat.act)
> pairs.panels(sat.act, scale = TRUE)
```

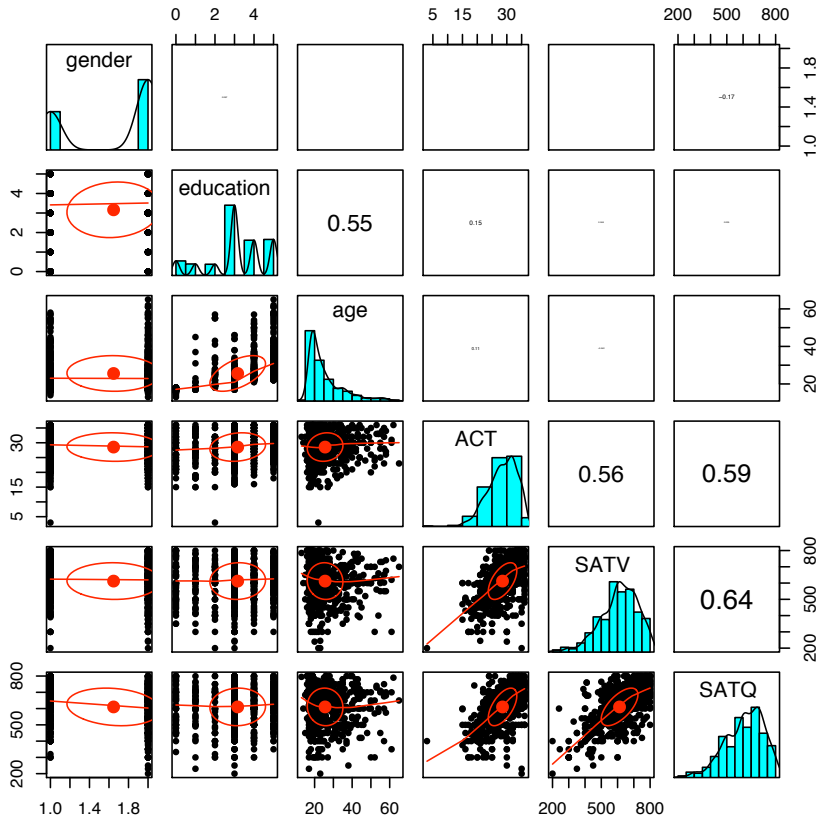

Figure 6: A probably not useful option in pairs.panels, is to scale the font size of the correlations to reflect their magnitude. More a demonstration of the range of possibilities in R graphics rather than a useful option.

add error bars to form "dynamite plots", which are slightly more informative. A much more useful graphic that displays the median, interquartile range, and the 99% confidence intervals is the `boxplot`. For small data sets, showing the actual data points, with or without error bars is easy to do using the `stripchart` function.

Consider the following four data sets. What is the best way to describe their differences?

```
> set.seed(42)
> x1 <- c(sample(5, 20, replace = TRUE) + 3, rep(NA, 30))
> x2 <- c(sample(10, 20, replace = TRUE), rep(NA, 30))
> x3 <- c(sample(5, 10, replace = TRUE), sample(5, 10, replace = TRUE) + 10, rep(NA, 30))
> x4 <- sample(10, 50, replace = TRUE)
> X.df <- data.frame(x1, x2, x3, x4)
```

6.3 Graph basics

There are many options available for graphing, including the number of graphs to present per page, the x and y limits of the graph, the x and y labels, the point size, the color, the type of line, etc. These are all specified in the `help` for `plot`, and the associated links. Here are just a few of the high points.

par Graphic options are stored in an object, `op`. These can be changed by using the `par` function which will set the graphics options to particular values. A typical use is to set the number of plots per page. This is done before calling a specific plot function. op <- par(mfrow=c(3,2)) #will put 3 rows of 2 columns of graphs on a page

x(y)lim The ranges of the x (y) variable. These are set inside the particular graphics call. xlim =c(0,10) #will make the axis range from 0 to 10. I

type Choose between p,l,b (points, lines, both)

pch What plotting character to use.

lty What line type (solid, dashed, dotted, etc.) item[col] What color to use.

6.4 Regression plots with fits

A more typical graphics problem is to plot regression lines, perhaps with the underlying data.

In addition to showing the regression analysis for presentations, one can also examine the residuals and errors of the regression.

```
> op <- par(mfrow = c(3, 2))
> barplot(colMeans(na.omit(X.df)), ylim = c(0, 14), main = "A particularly uninformative gr
> box()
> error.bars(X.df, bars = TRUE, ylim = c(0, 14), main = "Somewhat more informative")
> boxplot(X.df, main = "Better yet")
> stripchart(X.df, method = "stack", vertical = TRUE, main = "Perhaps better")
> stripchart(X.df, method = "stack", vertical = TRUE, main = "Add error bars")
> error.bars(X.df, add = TRUE)
> error.bars(X.df, main = "Just error bars")
> op <- par(mfrow = c(1, 1))
```

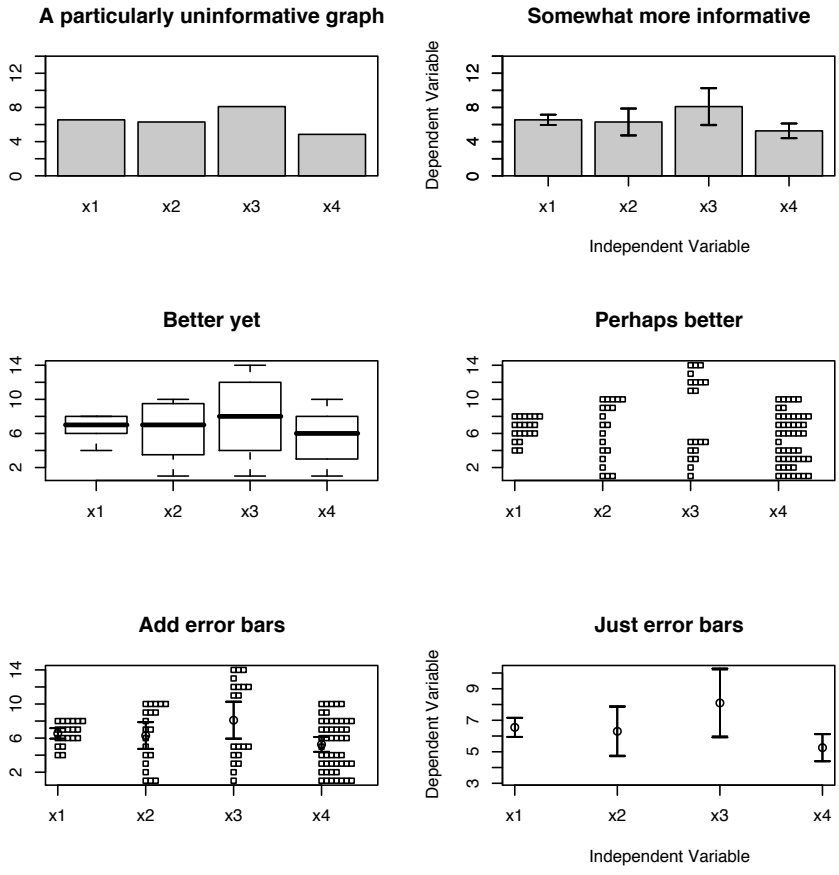

Figure 7: Six different ways of presenting the differences between four groups.

```
> op <- par(mfrow = c(3, 2))
> plot(1:10)
> plot(1:10, xlab = "Label the x axis", ylab = "label the y axis", main = "And add a title
+     pch = 21, col = "blue")
> points(1:4, 3:6, bg = "red", pch = 22)
> plot(1:10, xlab = "x is oversized", ylab = "y axis label", main = "Change the axis sizes"
+     pch = 23, bg = "blue", xlim = c(-5, 15), ylim = c(0, 20))
> points(1:4, 13:16, bg = "red", pch = 24)
> plot(1:10, ylab = "y axis label", main = "Line graph", pch = 23, bg = "blue", type = "l")
> plot(1:10, 2:11, xlab = "X axis", ylab = "y axis label", main = "Line graphs with and wit
+     pch = 23, bg = "blue", type = "b", ylim = c(0, 15))
> points(1:10, 12:3, type = "l", lty = "dotted")
> curve(cos(x), -2 * pi, 2 * pi, main = "Show a curve for a function")
> op <- par(mfrow = c(1, 1))
```

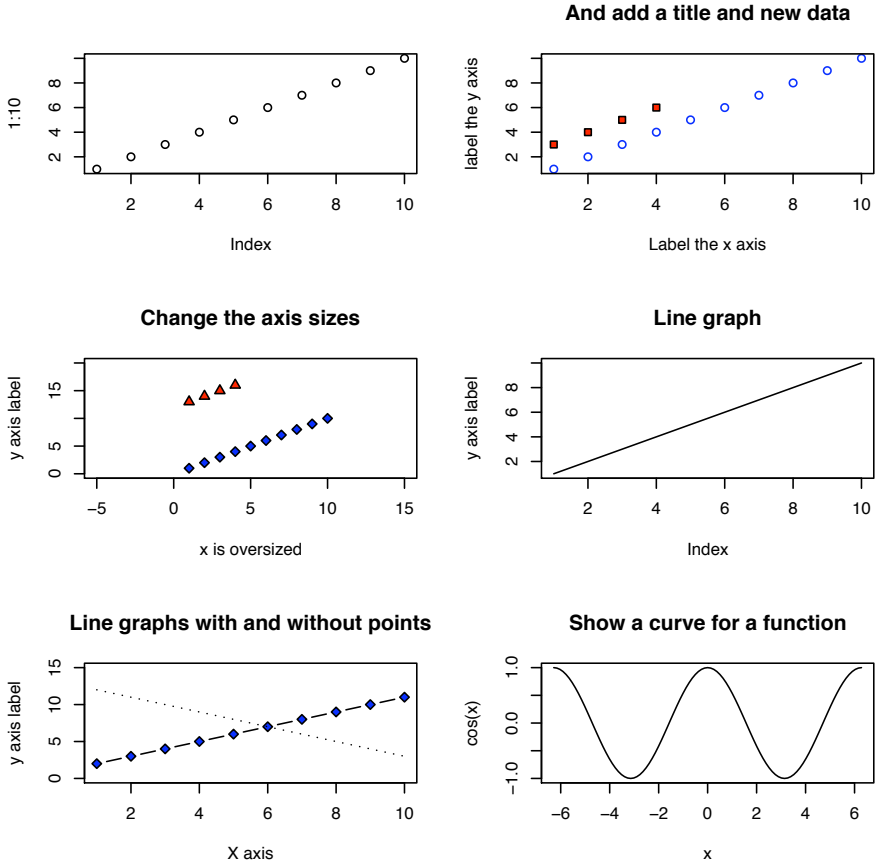

Figure 8: Various options of base graphics. Panel 1 is the basic call to plot. Panel 2 is the same, but with labels and titles. Panel 3 shows how to add more data, panel 4 how to do a line graph, panel 5 adds a second line, panel 6 is just an example the curve function.

```
> data(sat.act)
> with(sat.act, plot(SATQ ~ SATV, main = "SAT Quantitative varies with SAT Verbal"))
> model = lm(SATQ ~ SATV, data = sat.act)
> abline(model)
> lab <- paste("SATQ = ", round(model$coef[1]), "+", round(model$coef[2], 2), "* SATV")
> text(600, 200, lab)
```

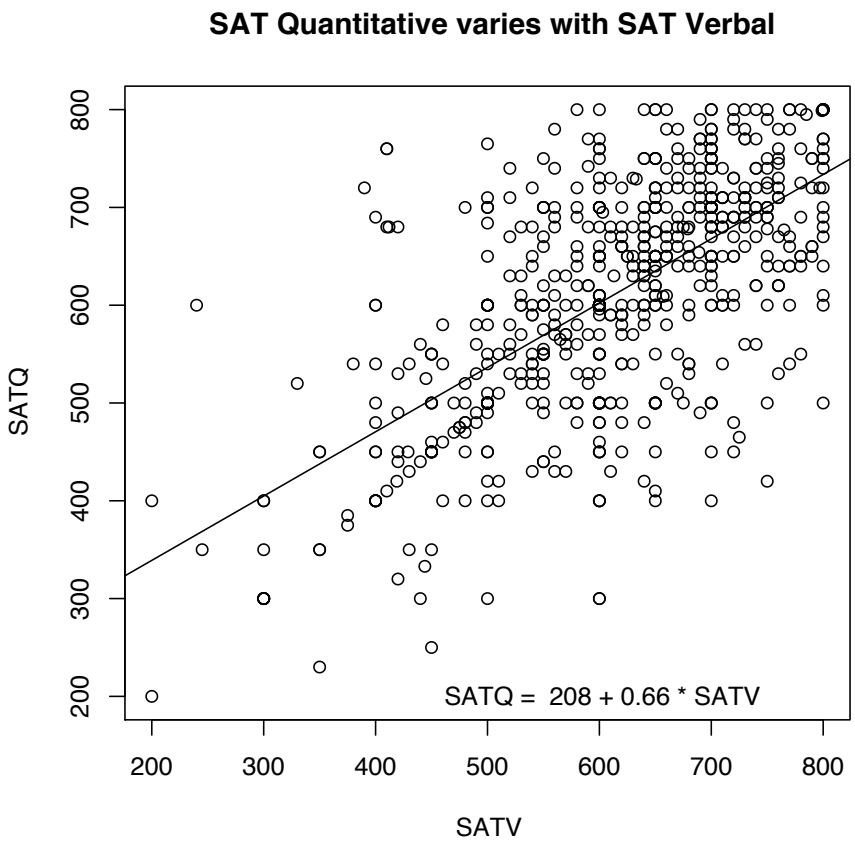

Figure 9: A regression data set with a regression line

```
> data(sat.act)
> color <- c("blue", "red")
> with(sat.act, plot(SATQ ~ SATV, col = color[gender], main = "SATQ varies by SATV and gend
> by(sat.act, sat.act$gender, function(x) abline(lm(SATQ ~ SATV, data = x)))
sat.act$gender: 1
NULL
------------------------------------------------------------------------------------------
sat.act$gender: 2
NULL
```

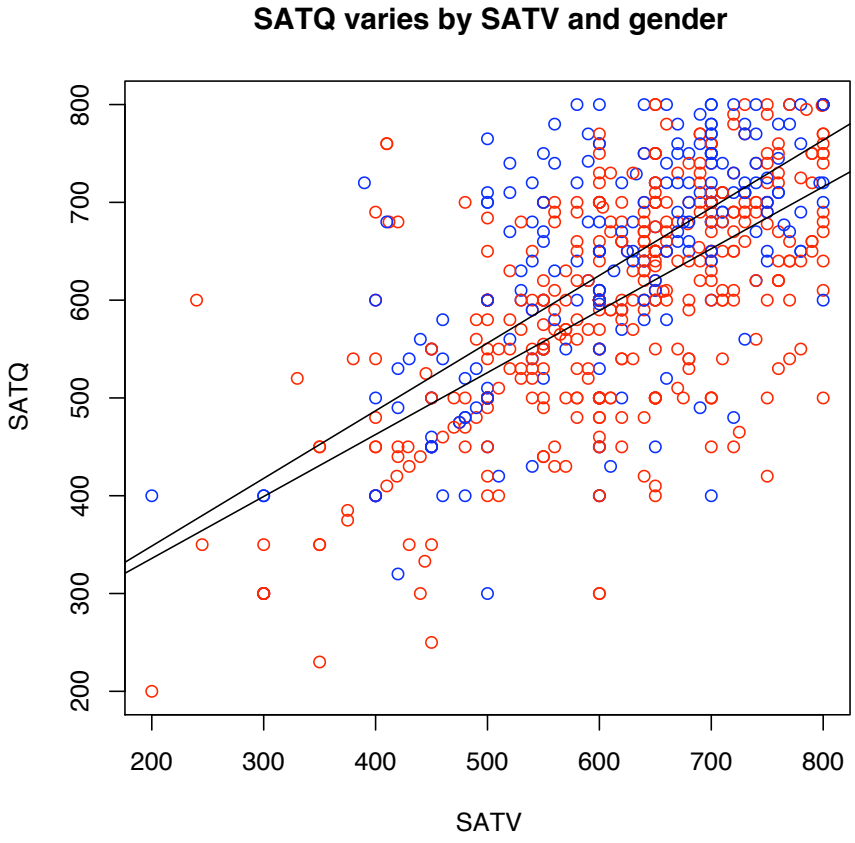

Figure 10: A regression data set with two regression lines. The higher regression line is for the women.

```
> op <- par(mfrow = c(2, 2))
> plot(lm(SATQ ~ SATV, data = sat.act))
> op <- par(mfrow = c(1, 1))
```

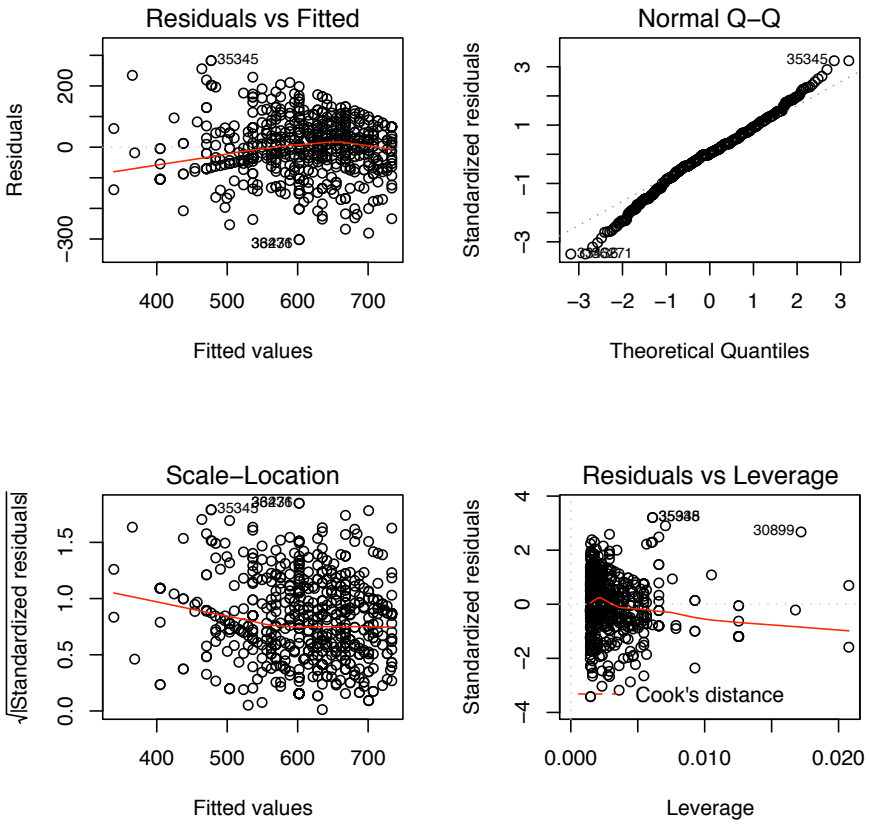

Figure 11: default

6.5 ANOVA plots

7 More complex graphics

In addition to the standard ways of displaying data, R includes packages meant for more graphical displays. These including mapping functions related to GIS files in the *map* package, density plots, and in particular graphs using such packages as *Rraghviz*.

7.1 Examples of Rgraphviz

Several of the psychometric functions in *psych* make use of *Rgraphviz* and are described in more detail in the vignette psych for sem. The next two figures take advantage of built in data sets, `Harman74.cor` and `Thurstone`.

This next figure, produced by `structure.graph` shows a symbolic structural equation path model.

```
> fxs <- structure.list(9, list(X1 = c(1, 2, 3), X2 = c(4, 5, 6), X3 = c(7, 8, 9)))
> phi <- phi.list(4, list(F1 = c(4), F2 = c(4), F3 = c(4), F4 = c(1, 2, 3)))
> fyx <- structure.list(3, list(Y = c(1, 2, 3)), "Y")
```

7.2 Using the maps package to process GIS data files

A great deal of geographic data is stored in GIS files on servers around the world. These GIS description files include all kinds of information, including the geographic coordinates of geographical regions (cities, states, countries, rivers, harbours, etc.) that can then be plotted using the *maps* package and its alternatives. The next figure is just a demonstration of what can be done (Figure 15).

7.3 Using graphics to teach sampling theory

Most students recognize that increasing the sample size will reduce the standard error of the measures and increase the ability to detect an effect if it is there. Unfortunately, some do not realize that the probability of a Type I error does not change as a function of sample size. A simple demonstration of the effect of increasing sample size on the width of the confidence intervals and also the probability of that confidence interval containing the population value is seen in Figure 16)

```
> data(Harman74.cor)
> ic <- ICLUST(Harman74.cor$cov, title = "The Holzinger-Harman 24 mental measurement proble
```

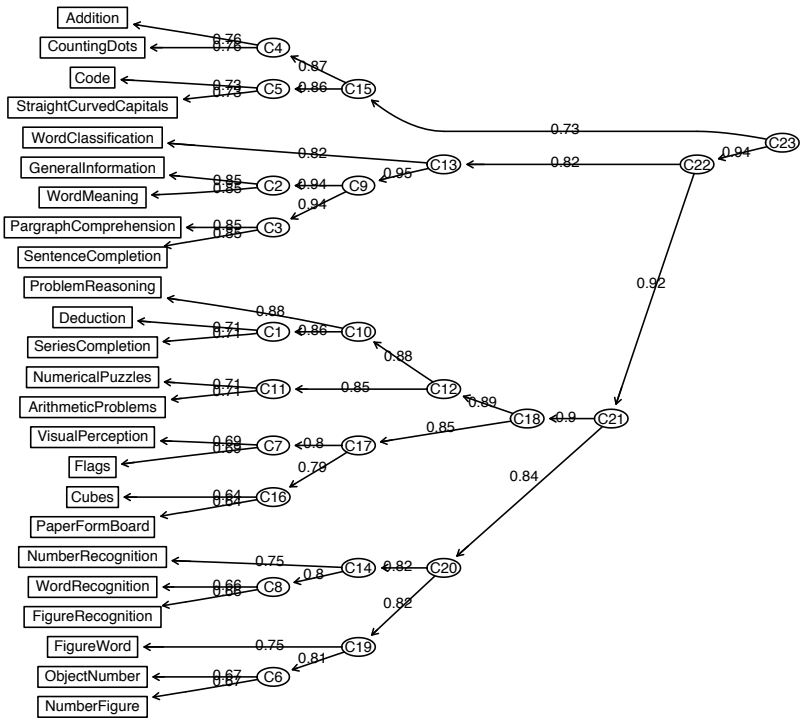

Figure 12: An example of tree diagram produced by the hierarchical cluster algorithm, ICLUST and drawn using Rgraphviz. The data set is 24 mental measurements used by Holzinger and Harman as an example factor analysis problem.

```
> data(bifactor)
> om <- omega(Thurstone, main = "A bifactor solution to a Thurstone data set")
```

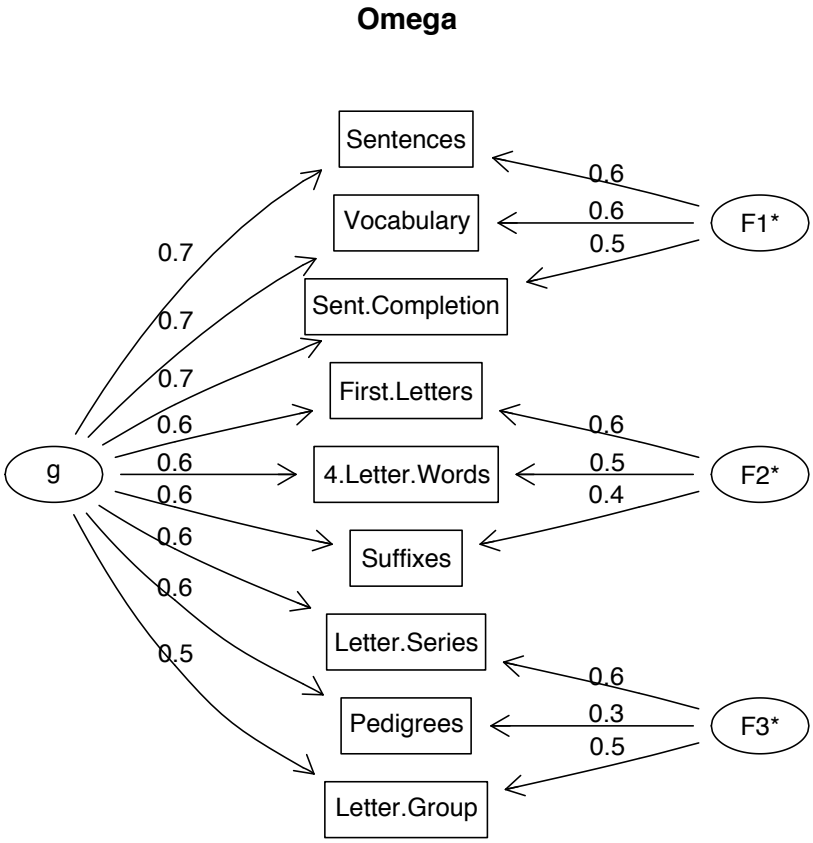

Figure 13: An example of bifactor model using Rgrapviz

```
> if (require(Rgraphviz)) {
+     sg3 <- structure.graph(fxs, phi, fyx)
+ } else {
+     plot(1:4, main = "Rgraphviz is not available")
+ }
```

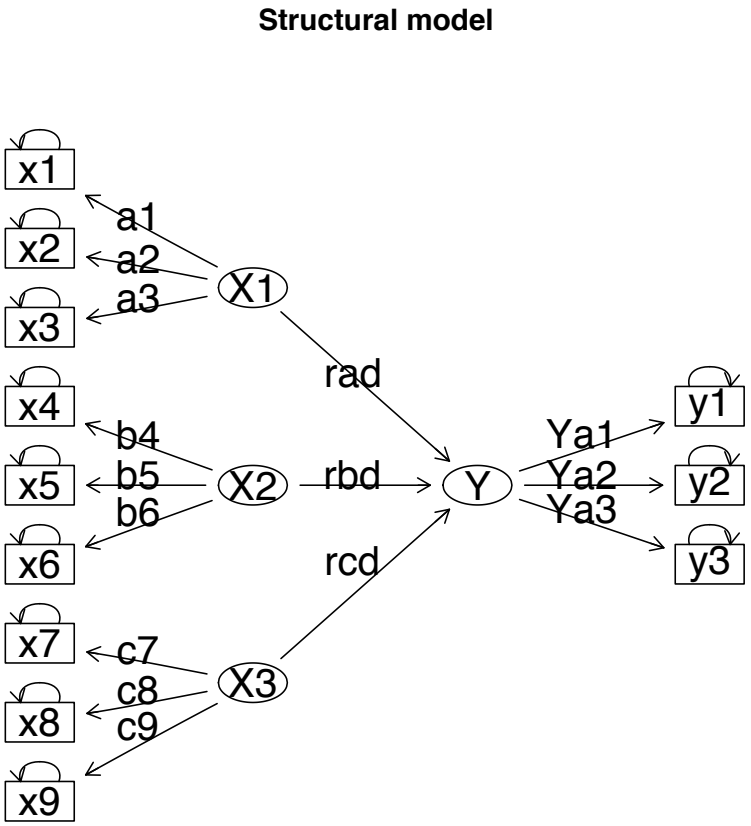

Figure 14: A symbolic structural model. Three independent latent variables are regressed on a latent Y.

```
> library(maps)
> map("county")
```

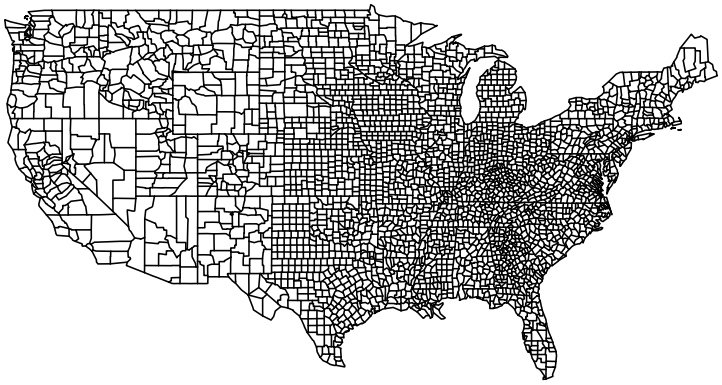

Figure 15: The counties of the US may be combined with demographic data to display income, voting records, education or any other data set organized by county.

```
> op <- par(mfrow = c(3, 1))
> set.seed(42)
> x <- matrix(rnorm(500), ncol = 20)
> error.bars(x, ylim = c(-1, 1), main = "N= 25")
> abline(h = 0)
> x <- matrix(rnorm(2000), ncol = 20)
> error.bars(x, ylim = c(-1, 1), main = "N = 100")
> abline(h = 0)
> x <- matrix(rnorm(8000), ncol = 20)
> error.bars(x, ylim = c(-1, 1), main = "N = 400")
> abline(h = 0)
> op <- par(mfrow = c(1, 1))
```

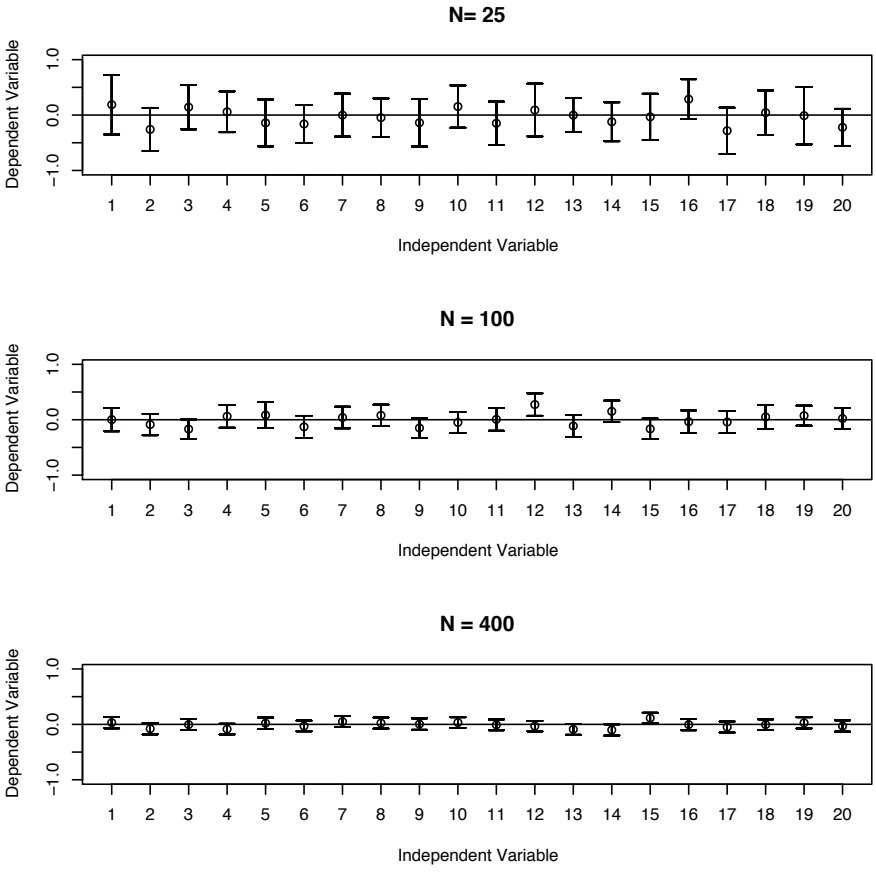

Figure 16: Increasing the sample size reduces the width of the confidence interval, but does not change the probability of including the population value.