Article

Using R for data analysis and research

David Lillis*
New Zealand Institute of Sport, PO Box 1260, Wellington

Introduction

More than twenty years after its inception, the R statistics language and environment for scientific and statistical computing and graphics continues to be a New Zealand success story. In 2012, the total number of R users worldwide was estimated at about two million (ORACLE, 2012). The precise number of users around the world right now is unknown, but most probably the total number is considerably more than that of 2012, and growing rapidly.

Recent developments

Four years ago I wrote an article on R for *New Zealand Science Review* (Lillis, 2011), and mentioned several useful contributed packages. In recent years, more packages have become available and several platforms have emerged that make it easy to run R scripts (e.g. R Studio), and in recent years a commercial version of R has grown in popularity – Revolution R.

As a direct response to the growing popularity of R, recently SAS[1] launched SAS University Edition, a free version of SAS, targeted at new learners. Initial reviews of SAS University Edition have been quite positive (Analytics Vidhya, 2015), but possibly the range and flexibility of the analytic tools and graphics of this particular product are not yet at the level of those of R.

Other developments of interest for researchers in New Zealand include the emergence of R-based consultancies, both in New Zealand and in other countries, the formation of R-Users Groups in Auckland, Wellington and Christchurch, and the introduction of several very fine contributed packages. I will describe some of these developments in this article.

R in statistical consulting

At present there is a significant international market for statistical consultancy, and this market extends to both teaching R and coding in R. Several New Zealand consultancies teach and use R as one of their primary analytic tools. For example, one New Zealand consultancy delivers R-based webinars and workshops, not only to New Zealand, but also to the USA, Asia, Australia, Great Britain and other parts of the world. These workshops and courses cover diverse topics, including introductory R, statistical modelling in R, introductory biomedical statistics in R, linear regression in R, generalised linear models in R, graphing with ggplot (a powerful R package for advanced graphics), structural equation modelling in R, and time series analysis in R. Many of the workshop attendees are already experienced professional researchers and statisticians, and quite a few are medical researchers. Such people can save a considerable amount of time in achieving mastery of R by learning directly from experienced practitioners.

R-users groups in New Zealand

Auckland has had an R users group for some years now, and last year Christchurch formed its own R users group. On 30 January 2014 Ian Westbroke, a public sector statistician, gave the first presentation to the Christchurch group, discussing the adoption of R by government departments and, in particular, the adoption of R Commander, a Graphical User Interface that makes it easy to learn R.

The Auckland R Users group has been active since 2013, and held several meetings during 2013 and 2014. Its first meeting for 2015 involved a presentation on 8 April from Professor Bernhard Pfahringer, of the University of Waikato, on the topics of Machine Learning and Data Mining.

In September 2013 a group of Wellington-based professionals and students formed the Wellington R-Users Group (see reference list for user group URLs). This group has held several sessions and hosted talks on topics such as using R for analysis of very large data sets and packages such as knitr and Sweave. These products enable you to produce integrated output files using R and LaTex commands (whose outputs contain the code for your analysis in which the output is woven through the code). Where possible, the Wellington R Users Group records its presentations and makes them available on YouTube.

All New Zealand R-Users Groups are intended to reflect the community of R-Users within their regions, and at their meet-

* Correspondence: sigma@outlook.co.nz

1 SAS Institute is an American developer of analytics software based in Cary, North Carolina: www.sas.com

David Lillis is a Senior Academic Manager (Research and Statistics) at the New Zealand Institute of Sport. Previously, Dr Lillis was a senior statistician at the New Zealand Qualifications Authority. A former secondary school teacher, he has also worked as a Senior Research Evaluator for the Foundation of Research, Science and Technology. David was born in Dublin, and obtained his PhD at Curtin University, Western Australia. He has a particular interest in research that requires quantitative methods.

ings they try to promote R and assist each other with various R methods and syntax.

Revolution R

Produced by Revolution Analytics, Revolution R Enterprise (see reference list for URL) is now firmly established, and makes it possible to analyse large data sets very efficiently. It seems to get around the well-known constraints on memory that characterises the standard versions of R. Revolution Analytics is a software company that develops new versions of R for specific applications, including additions for parallel processing. The core product, Revolution R, is provided free to academic users at no cost, while their commercial software focuses on the analysis of large data sets and large scale multiprocessor computing.

RStudio

RStudio is an environment for R that runs on Windows, Mac or Linux. It includes its own console, an editor for highlighting R syntax and graphics software. Its debugging tools and workspace management tools are especially useful for people new to R. RStudio is open source, though commercial editions are available. The URL for downloading RStudio is given in the references to this article.

Some great R packages

At this point I want to tell you about several R contributed packages that I recommend for your own research.

The ggplot2 package

The ggplot2 (grammar of graphics) package is well known, but nevertheless it is worth mentioning here. It is superb for creating informative and attractive graphics, and is particularly good for graphs that involve categorical data because you can map symbol colour, size and shape to the levels of a categorical variable quite easily. The syntax of ggplot2 is quite different to that of Base R, incurring a learning curve for both the newcomer and experienced R users. However, once mastered, ggplot2 enables practitioners to create beautiful graphics, ready for publication.

In the following example I have taken a medical data set on 45 patients within a randomised controlled trial, in which patients receive one of three treatments (A, B and C). Figure 1 presents a graph of patient body height (cm) against body mass (kg) before medical treatment, mapping symbol size to gender and mapping symbol colour to the binary categorical variable Exercise (i.e. whether or not the patients underwent an exercise regime during their course of treatment).

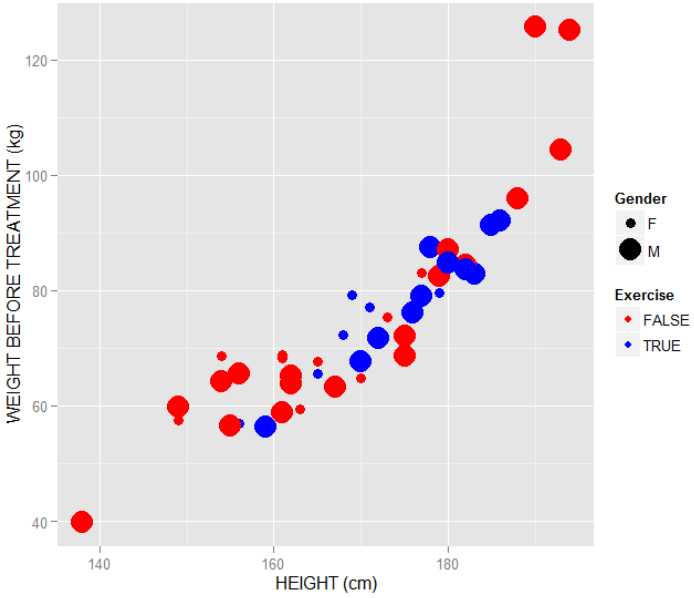

Figure 1: Graph of patient body height (cm) against body mass (kg) before medical treatment.

We see that mapping symbol size and colour to a variable can provide valuable additional insight into the relationships that exist within data.

Figure 2 presents a graph of patient body height (cm) against body mass (kg), mapping symbol shape to gender and mapping symbol colour to the categorical variable Treatment.

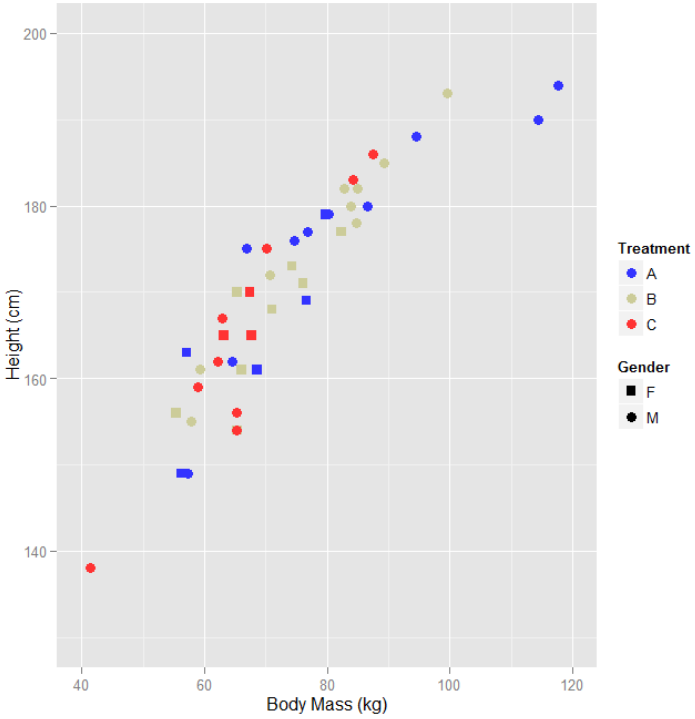

Figure 2: Graph of patient body height (cm) against body mass (kg), partitioned by treatment.

The combination of colour and shape has added valuable information to the graph of Figure 2. Next, the graph of Figure 3 gives a histogram of patients' body heights in cm, this time partitioning by ethnicity (a three-level categorical variable) and using a particular colour palette that is available through ggplot.

In this histogram, the bin width is in fact 10 cm, but we have three bars within each bin – one for each ethnicity. We have created an effective and attractive histogram in which ethnic subgroups are identified by colour. In this histogram I have created my own tick labels, including the descriptors Short, Average and Tall, at appropriate locations on the height axis.

In Figure 4 we see a box plot of the heights of female patients, partitioned by ethnicity, with different colours for each ethnic group.

In Figure 5 I have used ggplot to create a graph of counts of atomic disintegrations per second in a short-lived radioactive compound, and then used a ggplot function called `stat_smooth()` to fit a quadratic function, along with a standard error confidence band. Normally, an exponential function is used to model atomic decay, but here a quadratic in fact provided a better fit to the observed data.

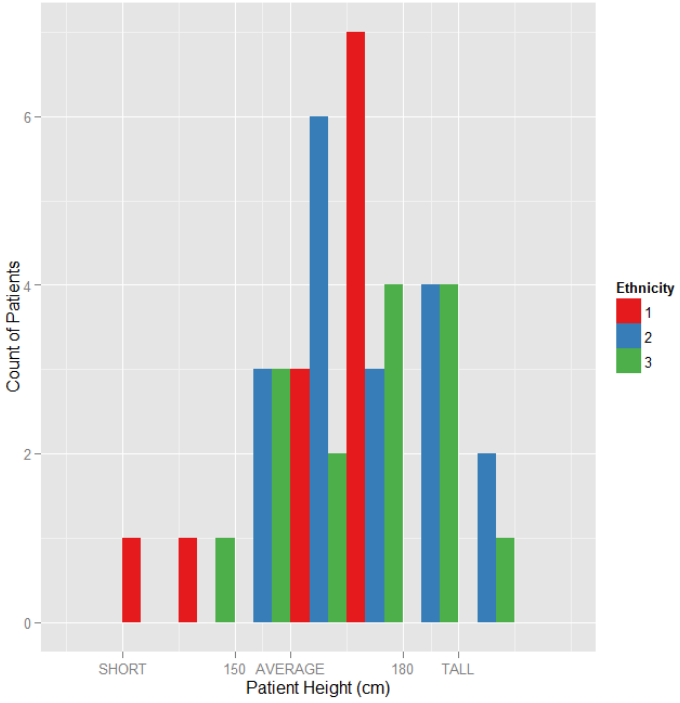

Figure 3: Histogram of patient body height (cm), partitioned by ethnicity.

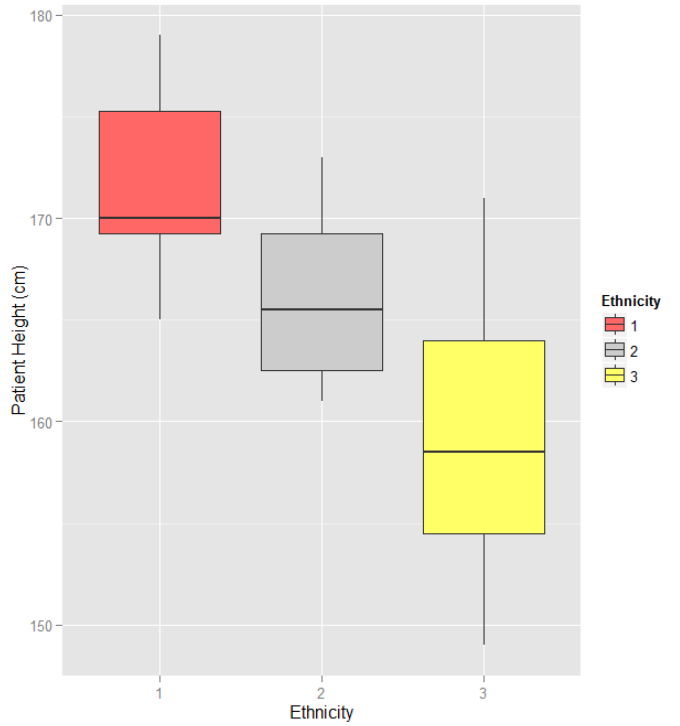

Figure 4: A box plot of the heights of female patients, partitioned by ethnicity.

The fitted curve of Figure 5 includes a standard error confidence band in light blue. The fitted curve could be presented in any colour, with your own desired line width, and either with or without the confidence band.

As a more complex example Figure 6 presents a faceted bar chart of the numbers of patients receiving each treatment (A, B or C), partitioned by gender and stacked according to whether or not the patient recovered.

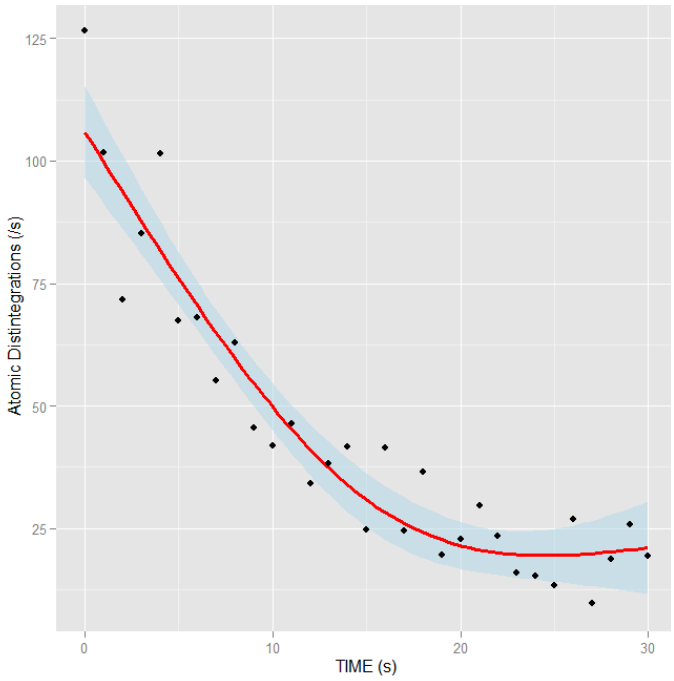

Figure 5: Counts of atomic disintegrations per second in a short-lived radio-active compound.

Figure 6 presents a lot of useful information at once. Partitioning by the three categorical variables allows us to compare patient recovery within and across the two genders, and also within and across treatment levels.

Finally, Figure 7 presents four other graphs, placed together using the `grid.arrange()` function, available within the grid library.

These examples give only a hint of the wonderful capability that R provides for creating superb graphics.

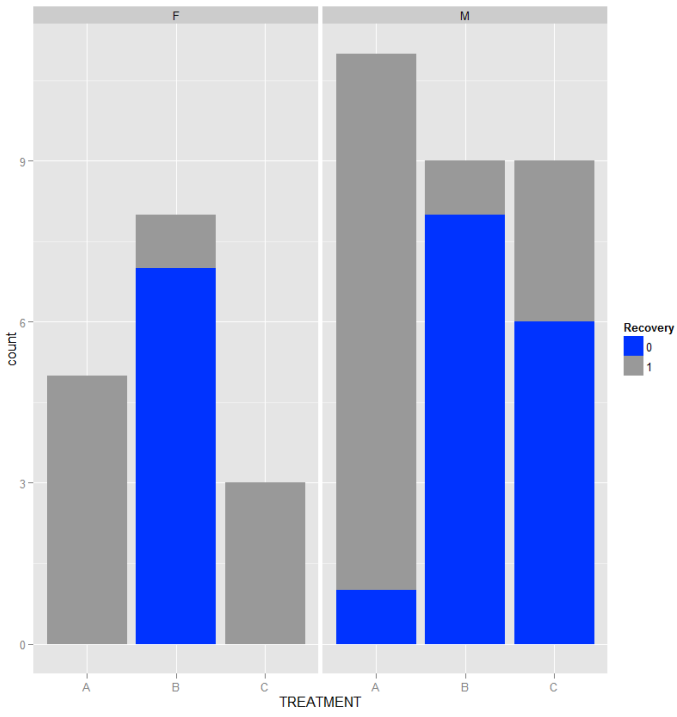

Figure 6: Numbers of patients receiving each medical treatment.

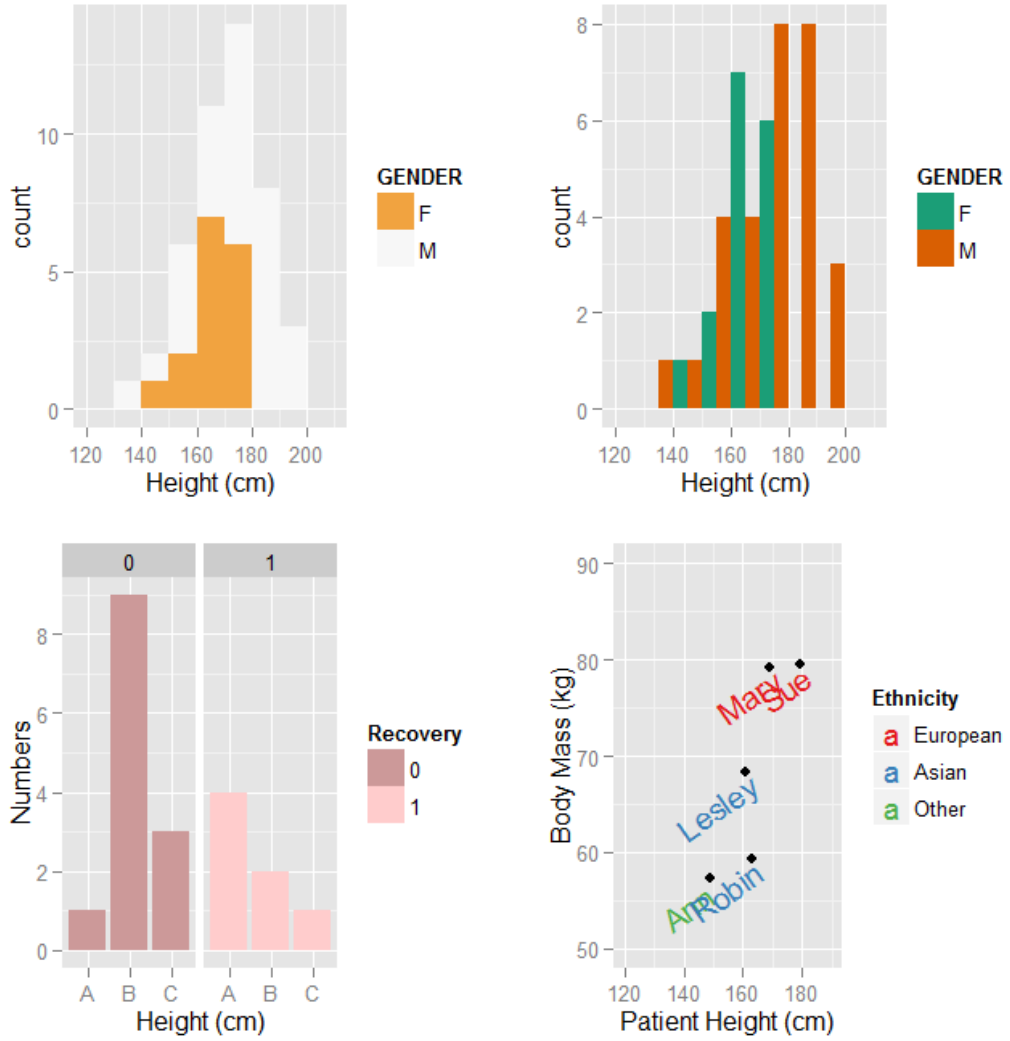

Figure 7: Four graphs, placed together.

The car package

The car package provides a wide range of functions for regression modelling. Here we look at a few of its functions. For a linear regression model the `residualPlots()` function plots the residuals against the fitted values, enabling a visual check for randomness. In addition, it performs a curvature test for each of the plots by adding a quadratic term and testing whether the quadratic is zero (Tukey's test for additivity when plotting against fitted values). Figure 8 shows regression residual plots, for a regression model with a single independent variable, that I created using the `residualPlots()` function.

The smooth, fitted curves in red suggest some curvature in the residuals. However, these curves are centred on zero (approximately), and the `residualPlots()` function tells us that we have non-significant p-values. Thus, our residuals are random enough for our purposes, and one of the critical the assumptions of linear modelling (i.e. that the residuals are distributed randomly) is upheld.

The car package also allows us to examine influential variables using the `qqPlot()` function. Figure 9 shows a plot that I created with this function.

Figure 9 suggests that observation ten is an outlier. Another function within the car package (the `outlierTest()` function) confirms that it has the characteristics of an outlier because the

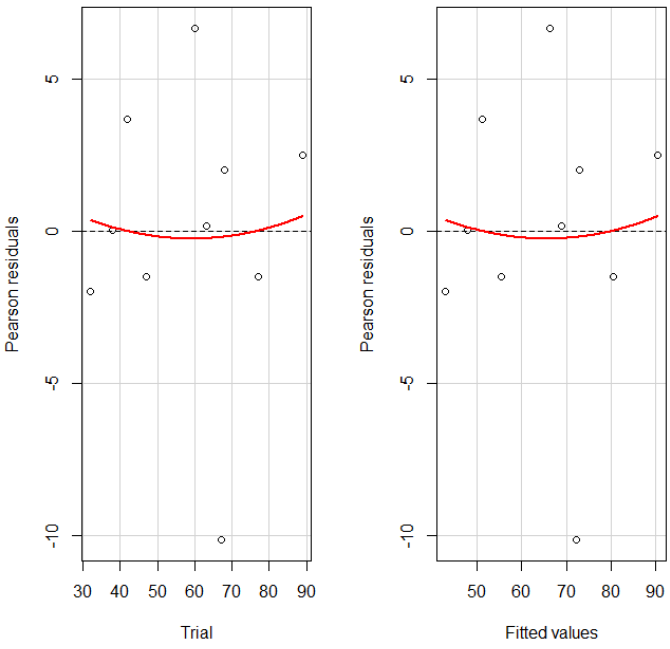

Figure 8: Residual plots for a regression model with one independent variable.

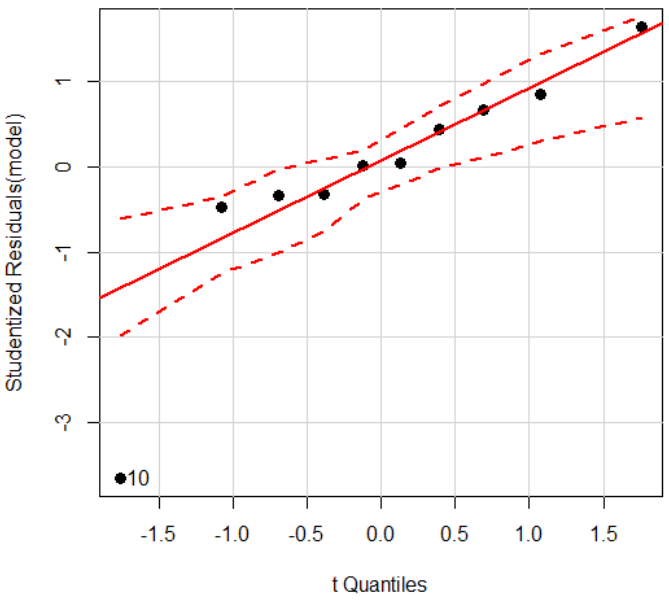

Figure 9: Testing for influential points using qqplot().

p-value for the Bonferonni Adjusted Outlier test is greater than 0.05. (The null hypothesis for the Bonferonni Adjusted Outlier test is that the observation is an outlier).

The lavaan package for latent variable modelling

The lavaan package (Rosseel, 2012) provides various methods for implementing structural equation modelling (SEM), confirmatory factor analysis (CFA), path analysis, growth mixture modelling, and other analytic models. SEM is a variant of multiple regression that is used in the social sciences and economics to complement standard techniques such as ordinary least squares multiple regression, factor analysis, and analysis of covariance. Up to now, possibly the most popular package for conducting SEM in R has been the sem package (Fox, 2006). However, the lavaan package gives researchers and statisticians a high quality and easy-to-use system for latent variable modelling.

Lavaan provides all of the expected diagnostic tests and information on both estimated coefficients and quality of fit. For CFA, the package provides the relevant diagnostics, including the comparative fit index (CFI), the Tucker–Lewis index (TLI), the root mean square error of approximation (RMSEA), and both the Akaike information criterion and the Bayesian information criterion. I have used lavaan for both CFA and SEM, and found it to be very reliable and easier to use than other packages.

The cusp package for catastrophe modelling

Catastrophe theory models evolution in the behaviour of a dynamic system under changes in environmental factors (behavioural and control variables) that determine the state of the system. It can explain how rapid changes in the system state can result from small changes in controlling factors, taking into account past states of the system. It is used in the physical sciences (involving mainly deterministic systems), but also the medical sciences, biological and social sciences, and in psychology (often involving stochastic systems). Cusp catastrophe models are used extensively to model critical economic systems, such as exchange market crashes, which can be modelled as endogenous events driven by speculative money. One body of opinion sees the endogeneity of market crashes as originating in conformity of investors with their peers and a degree of heterogeneity of the investor population (Levy 2008). Such factors can give rise to multiple equilibria in the market, sometimes leading to a market crash which can be modelled using cusp catastrophe theory.

In medical and health research patients' outcomes can be modelled using catastrophe theory. In clinical practice, certain physical and mental health conditions (e.g. strokes, heart attacks, seizures, depression) exhibit two modes: normal or abnormal, with low probability beyond the two modes – the inaccessible region. We may see a jump from one mode to the other if, for example, the diagnosis is based on the severity of the condition. Small changes in factors such as a patient's emotional state may produce sudden changes in health. These sudden changes are known as divergence. The timing and direction of such factors control the severity of the overall health outcome (hysteresis). Here, the term hysteresis refers to the notion that changes in outcomes, as we move from one mode to the other, cannot be determined uniquely by particular values of the control factors, because the sudden jumps do not always occur at the same values of the control factors.

Other scenarios in which catastrophe modelling has proved successful include analysis of the onset of hostile behaviour between nations, medical studies in which health outcomes are bimodal (e.g. normal or abnormal), animal aggression, failure (buckling) of building materials such as elastic beams, the development of anorexia nervosa, territoriality among animals such as reef fish, population dynamics, and collective bargaining. In all of these scenarios we may observe abrupt transitions, and other models cannot provide an equivalently comprehensive description.

Raoul Grasman, Han van der Maas and Eric Wagenmakers, of the University of Amsterdam, created the cusp package (Grasman et al, 2009) for catastrophe modelling, based on a special form of the maximum likelihood method. It is now relatively straightforward to implement different cusp catastrophe models and compare them with other models such as multiple regression and logistic regression. Of course, cusp catastrophe models produce a range of coefficients and diagnostic goodness of fit statistics that I will not go into here.

The cusp model consists of two stable regions and two thresholds characterised by sudden changes – these are the upper and lower regions (see Figure 10 below). The cusp model enables both the forward and reverse progression for different paths in health outcomes to be modelled together. It includes both a discrete component (normal vs. abnormal) and a continuous component (severity of the condition), whereas a linear model provides for the continuous component only. A catastrophe model can be compared with multiple linear regression and logistic regression models in order to provide a basis for identifying a catastrophic event.

I have experimented with the cusp package by fitting a cusp catastrophe model to time series of the logarithm of the net present value of Malasian companies over the 32-year period from 1980 to 2012. Over that period of time these companies were affected to a greater or lesser extent by a range of macroeconomic variables (Government Budget Deficit, Real Government Gross Rate, Interest Rate, Overvaluation, Corporate Tax, Corporate Credit and Money Multiplier). Some companies prove robust enough to withstand the accumulation of small economic shocks, while others go out of business, either gradually or rapidly. My intent was to model the performance

of these companies under the accumulation of changes in the control factors (i.e. the macroeconomic variables). The model was fitted using syntax of the following form:

```
fit <- cusp(y ~ V1 + V2 + V3 + . . . + Vn ,
alpha ~ GBD + RGDPGR + IN.RATE + OVERVAL + CORPT.R + PR.CREDIT + M2,
beta ~ GBD + RGDPGR + IN.RATE + OVERVAL + CORPT.R + PR.CREDIT + M2)
```

Here, V_1 to V_N are the time series of the logarithm of the net present value of each company, while alpha and beta represent time series linear combinations of the independent variables (in this case the seven macroeconomic independent variables). I generated the model output using the `summary()` command, as follows:

```
summary(fit, logist = T)
```

The output includes p-values for the fitted coefficients, and R-Square values for the overall cusp model, the multiple linear regression model, and the logistic regression model. Figure 10 gives a three-dimensional plot of the model (obtained through the `cusp3d()` command) for these Malaysian companies.

The graph shows the cusp catastrophe model for the outcome z in the equilibrium plane. The continuous component covers the linear and gradual process (Path A), while the discrete component characterises the sudden and nonlinear process (Paths B and C).

The variable X is the asymmetry control variable and Y is the bifurcation control variable. Dynamic changes in Z have two stable regions (attractors), which consist of the lower area at front left (the lower stable region) and the upper area at the front right (the upper stable region). Outside of these stable regions the outcome variable Z is very sensitive to small changes in X and Y. The region of instability is projected on to the control plane (X, Y), forming the cusp region (shaded in grey). This cusp region is delineated by the line O'–Q' (the ascending threshold) and the line O'–R' (the descending threshold) of the equilibrium surface. Within the cusp region, the outcome Z becomes highly unstable in response to changes in X and Y, jumping between the two stable regions when the control plane (X, Y) is close to the two lines O'–Q' and O'–R'. In our graph, Paths A, B, and C

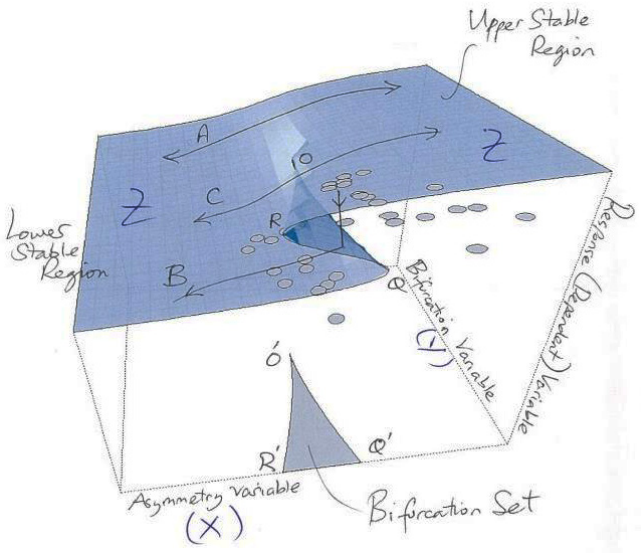

Figure 10: Three-dimensional plot of a catastrophe model.

represent possible pathways of change in the outcome. In path A we have $Y < 0$, and we see a smooth relationship between Z and X. However, path B shows that for cases where $Y > 0$, if X increases sufficiently so as to touch and pass the ascending threshold, the outcome Z will jump suddenly from the lower stable region to the upper stable region of the equilibrium plane. Finally, path C involves a sudden drop in Z as X declines sufficiently to touch and pass the descending threshold.

Cusp catastrophe modelling is now being used increasingly across many areas of research, and ultimately may become almost as widespread as regression models for particular applications. Further details on cusp catastrophe modelling can be found in many on-line sources and in the text: *Critical Transitions in Nature and Society*, by Sheffer (2009).

The ltm package for item response theory

Item response theory (IRT) refers to a family of statistical models that are designed to assess the quality of psychometric tests and assessments, and measure abilities, attitudes and other latent traits. IRT is used to underpin the design, analysis and scoring of tests and questionnaires, and is used in many countries to inform the development and analysis of educational assessments. Hambleton *et al.* (1991) provide a very good introduction to IRT, both for first-time readers and experts.

Developed by Dimitris Rizopoulos, of the Catholic University of Leuven, the ltm package (latent trait model, Rizopoulos 2013) makes it relatively easy to perform analysis of multivariate dichotomous and polytomous data using latent variable models. Latent variables provide a method of quantifying unobserved variables such as attitudes, intelligence, mathematical ability and verbal ability. Modelling them accurately provides a basis for applications such as aptitude and ability testing, educational testing, social sciences, psychology and other fields.

The ltm package provides estimates of the ability, a parameter that measures performance on the test as a whole. The ability measures the magnitude of the latent trait of the person or, more generally, the capacity or attribute measured by the test. The ability could measure a cognitive or physical ability, a skill, knowledge or attitude etc. Ability is a multi-dimensional concept, and cannot be measured uniquely for any person. In fact, the constructs we wish to measure, such as cognitive, numeric or linguistic abilities, are actually a synthesis of many related abilities and skills. Abilities are calculated for each candidate on the basis of the entire complement of item grades. Actually, abilities estimated from IRT can provide better measures of performance than aggregates of marks or raw grade point averages, because the ability estimate takes explicit account of the discrimination and difficulty properties of each item. In IRT we use an ability scale which may be thought of as representing the set of skills, abilities and knowledge that contribute to performance. This scale is calibrated to have a mean of zero, and ranges (theoretically) from negative to positive infinity. The unit of ability is the logit, a unit that is well known from logistic regression.

The ltm package also estimates the difficulty of each item at each available category or grade. The difficulty of an item reflects the proportion of test-takers who are successful in that item (i.e. either providing a correct answer to a dichotomous (two-category) item, or obtaining one of the passing grades in a polytomous item), taking account of the abilities of the candidates. For a dichotomous item (yes or no; right or wrong

etc.), item difficulty is defined as the point on the measurement scale at which the probability of success is 0.5. For a polytomous item that carries several possible grades, we estimate a difficulty parameter for each available grade, except the lowest.

In addition to these parameters, ltm estimates overall model fit (goodness of fit that is estimated through a chi-square value) and provides standard errors and other diagnostics for each of the estimated parameters. In addition, you can use ltm to plot item characteristic curves (curves created from IRT that describe the performance of the item), item information functions and test information functions.

For dichotomous data the ltm package provides the Rasch model (itself based on the logistic function – see the next section on generalised linear models), the two-parameter logistic model and Birnbaum's three-parameter models. For polytomous data the graded response model of Samejima is available. I have used ltm extensively, and find it easy to use and faster than many other packages and other equivalent software. The necessary syntax is simple. For example, to fit the graded response model and the Rasch model to a dataset (actually to an R data object that here we call object), you enter the following syntax at the command line:

```
grm(object)
```

and

```
rasch(object)
```

Figure 11 gives item characteristic curves for five test items, created under the Rasch model and obtained through the ltm package:

A description of item characteristic curves is given in Johnston and Lillis (2011). However, each of the curves of Figure 11 has the general form of a logistic function. The further an item characteristic curve is located to the right, the more difficult is the item. Thus, items 2 and 3 are more difficult than the other items, reflecting the necessity for a higher level of ability for success in items 2 and 3.

Many new packages have become available in recent years, and cannot all be discussed here. However, I conclude this discussion of R packages by mentioning the Bioconductor Project, which provides a repository of R packages of high throughput data, with particular application to genomics. The Bioconductor Project uses R as its statistical programming language. Like R it is open source and open development. Currently, it includes nearly one thousand R-based software packages and has two releases each year.

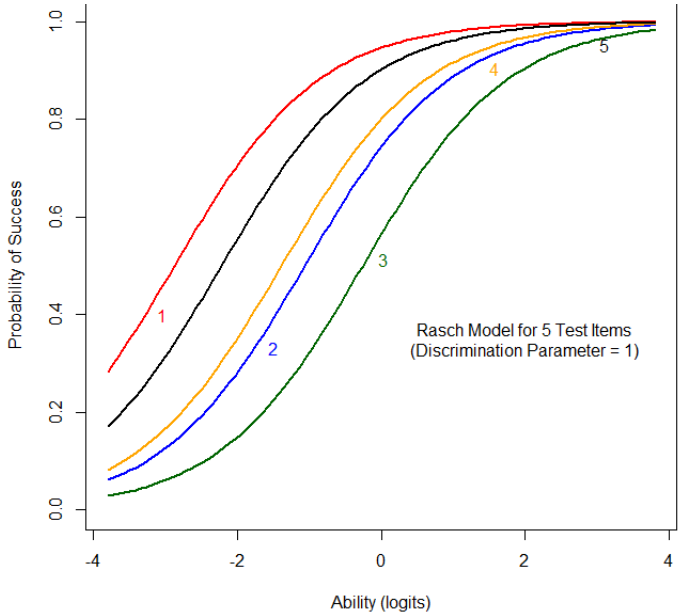

Figure 11: Item Characteristic Curves for five test items under the Rasch model.

R for generalised linear models

Ordinary least squares regression provides linear models of continuous response variables where we have one or more continuous independent variables (or predictors). However, much data of interest to researchers is not continuous, and so that other methods must be used to create useful predictive models. These other method include generalised linear models (GLMs), which can be implemented in R using the `glm()` command.

The `glm()` command is a core R function, so that it is not necessary to download contributed packages to implement GLMs. It was designed to perform GLMs on binary outcome data, count data, probability data, proportion data, and other data types. GLMs are designed to model response variables that are not distributed normally by relating the linear model to the response variable through a link function. GLMs are introduced in many senior degree-level statistics courses, psychology and in the bio-medical sciences, but in my opinion are under-utilised in the social sciences and education. Here I will give a few examples in quite some detail, some of which are taken from education research. An excellent text on GLMs is *Generalised Linear Models*, by McCullagh & Nelder (1989).

Generalised linear models are relatively easy to fit in R. The `glm()` command incorporates various arguments, as follows:

```
glm(formula, family, data, subset, ...)
```

The `family` argument specifies the variance model and your choice of link function. For binomial variance the link functions include the logit, probit or complementary log-log functions. For the default links, only the family is specified. For non-default links, you must supply the link argument. For example:

```
glm(formula, family=binomial(link=probit))
```

Table 1 gives a summary of the main error families and the relevant link functions.

Table 1. Main families of errors and their relevant link functions.

Error family	Default link	Inverse of link	Data type
Binomial	logit	1/ [1 + 1 / exp(x)]	Proportions or binary data
Poisson	log	exp(x)	Count data
Gaussian	identity	1	Normal errors
Gamma	inverse	1/x	Non-constant errors

Modelling with logistic regression

Let's take a look at a practical example. Recently I was able to access data on a study of elementary school students in the US who undertook additional instruction to assist them to pass a competency test. Subsets of them were tested at various stages during the year, after undergoing differing amounts of additional instruction. I wanted to create a logistic model that explains the numbers passing and failing for various total hours of instruction.

	Hours	Fail	Pass
1	1	3	1
2	25	3	1
3	50	9	6
4	100	18	4
5	150	25	36
6	200	46	76
7	250	61	98
8	300	67	204
9	350	74	287
10	400	89	305

Note that this data set is arranged as frequencies of students gaining either a pass or a fail. Essentially, it is a summarised version of a larger data set of zeroes and ones, where the result for each child is recorded in a separate row. Essentially we have a binary (dichotomous) outcome variable (e.g. pass or fail, yes or no, right or wrong, success or failure, presence or absence). We use a logistic regression model with binomial errors to explain the proportions of students passing and failing. Usually, such variables are given as vectors of zeroes and ones, and we treat them as deriving from a binomial trial with a sample size of one. Here, we cannot use ordinary least squares regression because the:

1. variance of the response variable changes across the range of values of the predictors
2. error terms are not distributed normally
3. predicted probabilities may exceed 1 or be less than zero.

Let's assume that the variable y is a random binary variable, with mean p and variance $p(1-p)$. The probability of success is p, and the probability of obtaining the outcome y is P(y) where:

$$P(y) = p^y (1-p)^{1-y}$$

This formula is a form of the binomial distribution in which we have a binomial trial with a sample size of 1. For the outcome $y=1$ this expression reduces to $P(1)=p$, while for outcome $y=0$ we get $P(0) = 1 - p$.

We use logistic regression to fit a predictive model for binary outcome data, proportions and probabilities. Figure 12, which I created in R, gives the general shape of the logistic function.

The logistic function passes through 0.5, and tends asymptotically to zero for negative values of x. It tends to +1 for positive values of x and is bounded below by zero and above by one. Fitting to a logistic function ensures that probabilities and proportions are bounded. Thus, we cannot predict negative probabilities or proportions, nor can we predict probabilities or proportions greater than one. The logistic function is as follows:

$$P = e^{a+bx} / 1 + e^{a+bx}$$

Using this expression, it is easy to show that the logit reduces to a linear function of x:

$$\ln(P / 1 - P) = a + bx$$

This is the logit transformation of P (also called the log-odds), the link function that provides a linear model in the variable x. Since we need a linear model, we regress the logit against x. We see that the logit is actually the quantity: a + bx. Thus it is the logit, rather than the original variables, that provides the linear model. Thus, logistic regression is in effect ordinary least squares regression in which the logit acts as the response variable.

Returning now to the problem of modelling the proportions of students passing and failing the test – after some experimentation I found that the following syntax provided a good model for the observed data:

```
model <- glm(Y ~ hours, binomial)
```

This model produces the graph of Figure 13 (obtained after writing some further R syntax).

Our model reflects the fact that the proportions of students passing the assessment tend to increase with the total number of hours of tuition. The curve looks like a straight line, but in fact it is curved, in accordance with a logistic model.

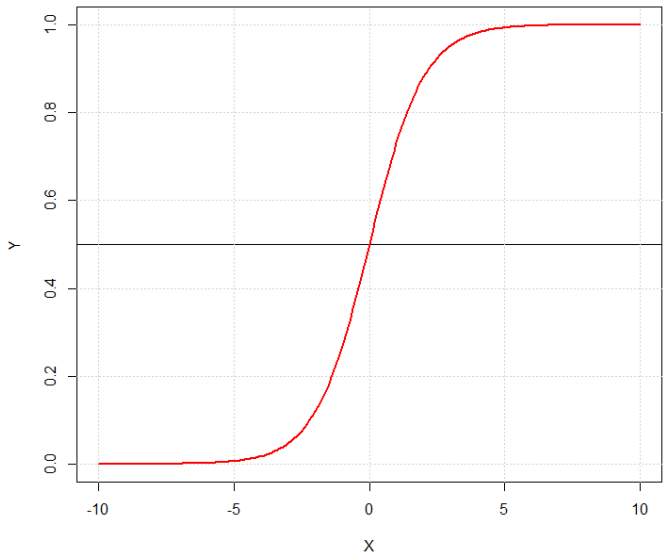

Figure 12: The logistic function.

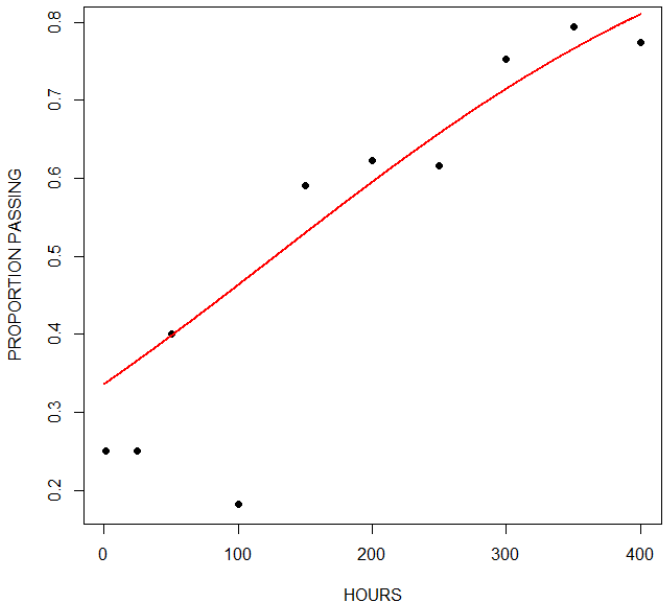

Figure 13: Raw data and fitted binomial model for students passing the assessment.

A GLM on count data

We can also use R to implement GLMs on count data. In such data the errors may well be distributed non-normally and the variance may increase with the mean values. As with binary data, we use the glm() command, but this time we specify a Poisson error distribution and the logarithm as the link function (the default link function for the Poisson error distribution). The

Poisson error distribution assumes that the variance is equal to the mean. Therefore, specifying a Poisson error distribution accounts for integer data whose variance is equal to their mean, while specifying a logarithm as the link function forces all of the predicted values to be positive.

In the following example we fit a GLM to count data using a Poisson error structure. The data set consists of counts of patients diagnosed with an infectious disease within a period of days from an initial outbreak. Let's look at the first three rows and last three rows:

Days	Number
1	0
2	0
3	0
.	.
.	.
97	0
98	0
100	1

Now we fit the GLM in R, specifying the Poisson distribution by including it as the second argument. After experimentation, I found that the best model was as follows:

```
model <- glm(Number ~ Days, poisson)
```

This model gives rise to the graph of Figure 14 (again obtained after writing some R syntax).

The graph of Figure 14 appears to show a decrease in numbers of cases with number of days. However, the calculated p-value for the number of days was 0.09. Thus, the apparent decline over time was non-significant at $\alpha = 0.05$, but marginally significant at $\alpha = 0.1$).

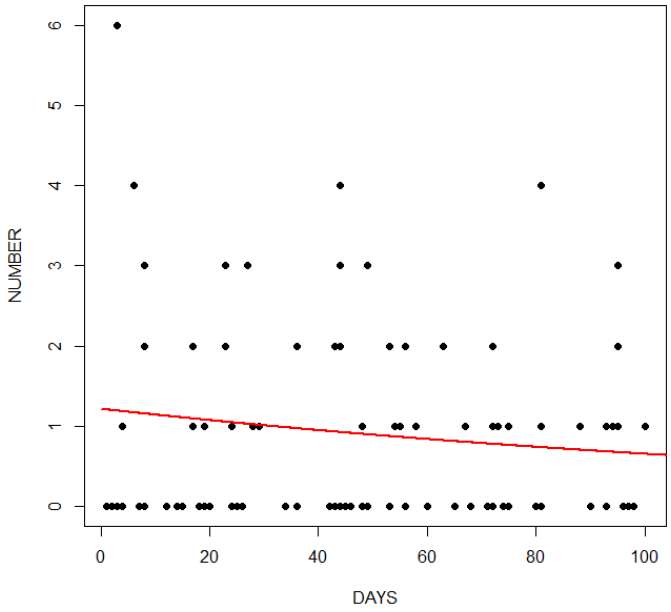

Figure 14: Raw data and fitted Poisson model for counts of people presenting with the infectious disease.

A GLM using probit regression

In addition to the logistic model that we discussed above, we can use R to implement probit regression to model dichotomous (binary) outcome variables. In probit regression, the inverse standard normal distribution of the probability is represented as a linear combination of the predictors. The probit function is the quantile function associated with the standard normal distribution. Figure 15 gives a graph of the probit function,

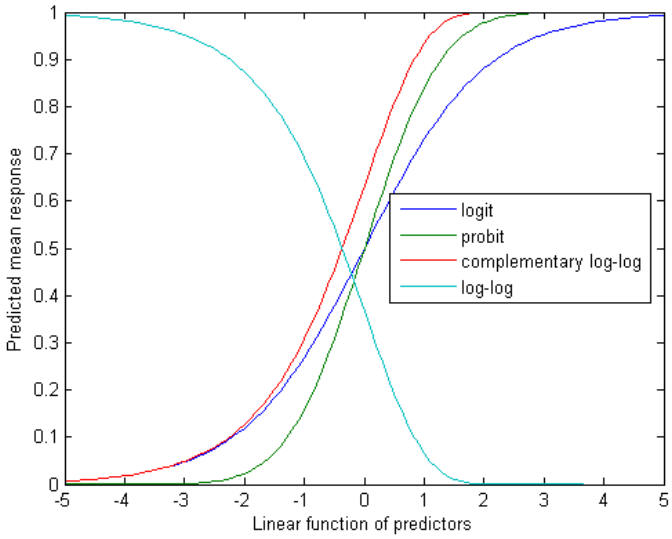

Figure 15: Graph of the logit, probit, log-log and complementary log-log functions.

which looks similar to that of the logistic function (and indeed the complementary log-log function).

The probit function passes through 0.5, and tends asymptotically to zero for negative values of x. It tends to + 1 for positive values of x. Like the logistic function, the probit function is bounded below by zero and above by one, so that fitting to a probit function ensures that all predicted probabilities and proportions are bounded by these values. As with the logistic function, we cannot predict negative probabilities or proportions, and nor can we predict probabilities and proportions greater than one.

I used a probit model to investigate how predictors such as test scores, high school grade averages and socio-economic level predict the successful completion of a Bachelor degree at a well-known North American university (for which I have the relevant data). Here we have an outcome variable (success or failure) which we can represent as a binary variable comprised of ones and zeroes.

The data set for our example includes a binary variable called Success. The three independent variables include Socioeconomic level (Socio is a categorical variable of four levels: 1, 2, 3 and 4, where 1 represents the highest and 4 the lowest level). We also have a SAT test score (Test) with a maximum of 800, and a grade average (Gd) whose maximum is 4.0. Let's see six rows from this data set:

Socio	Test	Gd	Success
3	789	3.97	1
3	728	3.83	0
3	693	3.79	0
2	693	3.71	0
2	649	3.70	1
1	609	3.65	1

In Figure 16 we have used R's `barplot()` command to graph the cross tabulation of the two categorical variables.

It seems as though the group from socio-economic level 1 has performed more strongly than the other groups, while the group from socio-economic level 3 has performed a little worse than expected. I now fitted a GLM to the data using the probit link function, as follows:

```
model <- glm(Success ~ Test + Gd + Socio, family = binomial(link = "probit"), data = dataset)
```

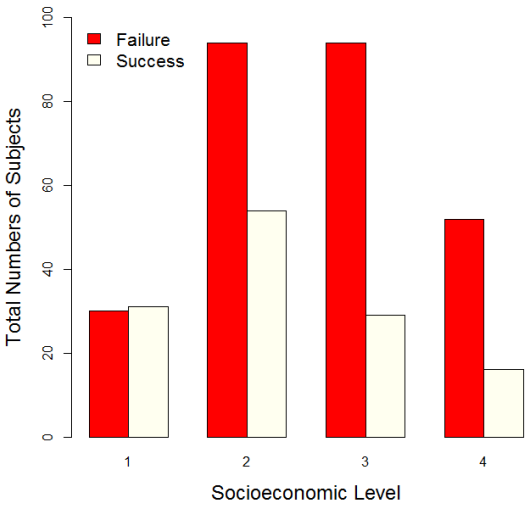

Figure 16: Bar chart of student numbers by socioeconomic level and success

The model gives rise to the following graphs (after developing some further R syntax), where I have classified grade averages using a common approach: averages up to 2.5 are classified as C, grade averages between 2.5 and 3 as B-, grade averages between 3.0 and 3.5 as B+, and grade averages between 3.5 and 4.0 as A. Figure 17 gives the probit model for each grade average.

The model of Figure 17 appears to work well, revealing the effect of socio-economic level, and indicating that both higher test scores and higher grade averages are predictive of success. Note that the students from socio-economic level 4 out-performed those from level 3 at each grade. This fact was evident from the bar chart above.

A GLM for a multinomial outcome variable

The next example involves a GLM on a multinomial outcome variable (i.e. the outcome variable is a categorical variable of several levels – three levels in the present case). In our example the levels have no intrinsic order. However, some outcome variables that you may wish to model using a GLM may embody intrinsic order (e.g. a Likert agreement scale involving increasing levels of agreement, or a rating scale in which certain responses reflect higher ratings than others). The `glm()` command provides both for variables that embody no intrinsic order and for variables that do embody intrinsic order.

For our example, we have data on 200 high school students and wish to predict whether they terminate their education at high Sshool, whether they progress to a polytechnic programme, or whether they attempt a university degree. The independent

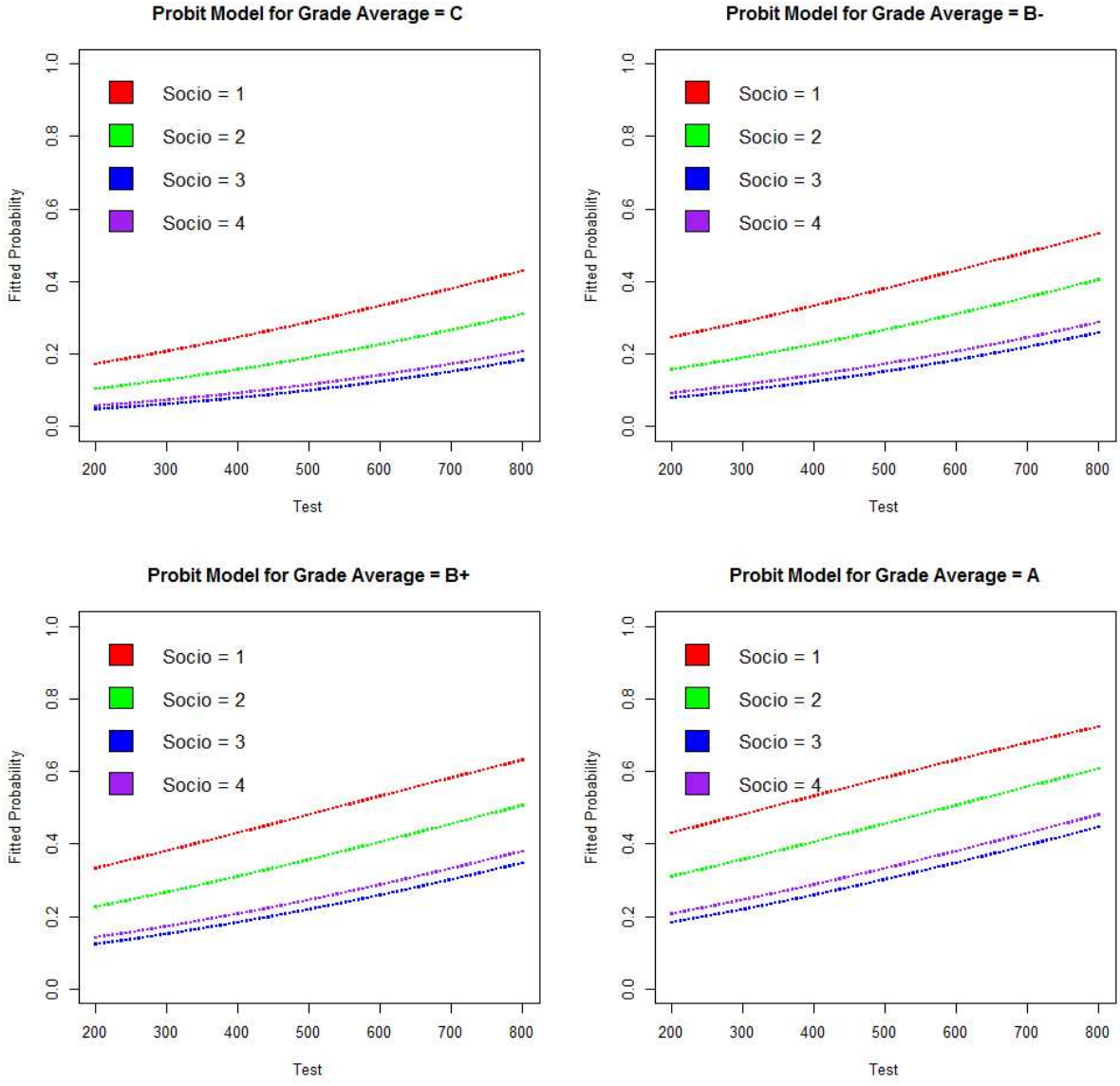

Figure 17: Probit model for each grade average.

variables are their household income (Income) and an ability test score (Score). Income is a categorical variable of three levels – Low, Medium and High. Score is a continuous variable that ranges from 0 to 100.

For this problem we have three possible outcomes (terminating at High School, proceeding to Polytechnic, or attempting a Degree). We use multinomial logistic regression, where we model the logarithm of the odds of the outcomes as a linear combination of the independent variables. To fit our model I used the `multinom()` function, which is available by installing the `nnet` contributed package. Let's take a look at the first six rows (where the data are arranged in descending order of Score).

Score	Income	Education
78	High	Degree
76	High	Degree
75	High	Degree
75	High	HighSchool
74	High	Polytechnic
73	Mid	Degree

Now we gain more insight into the data by creating a cross tabulation of the two variables together.

	Income		
Education	High	Low	Mid
Degree	37	19	47
HighSchool	8	11	30
Polytechnic	12	17	19

Let's look at the mean scores across the levels of Education using `tapply()`:

Degree	HighSchool	Polytechnic
67.0	58.2	62.5

Clearly, students aiming for a degree tend to score higher marks than others. Now we run the model using `multinom()` and store the model output as an object called multimodel. We include the Education variable and the two independent variables Income and Score.

```
multimodel <- multinom(Education ~ Income + Score, data = dataset)
```

Figure 18 gives the data, including a bar chart of student numbers, grouped by education and income level (i.e. the cross-tabulation given above) and fitted model – a separate plot for each level of the outcome variable Education.

Generally, the model of Figure 18 appears to make sense. High-income students are more likely to attempt a degree, and high scores are indeed predictive of attempting a degree. Low-income students are more likely to attempt a polytechnic qualification, while high scores are less predictive of attempting a polytechnic qualification because high-scoring students generally choose to attend university. Higher scores are not predictive of terminating at high school because high-scoring students generally pursue higher education. It appears that we have a

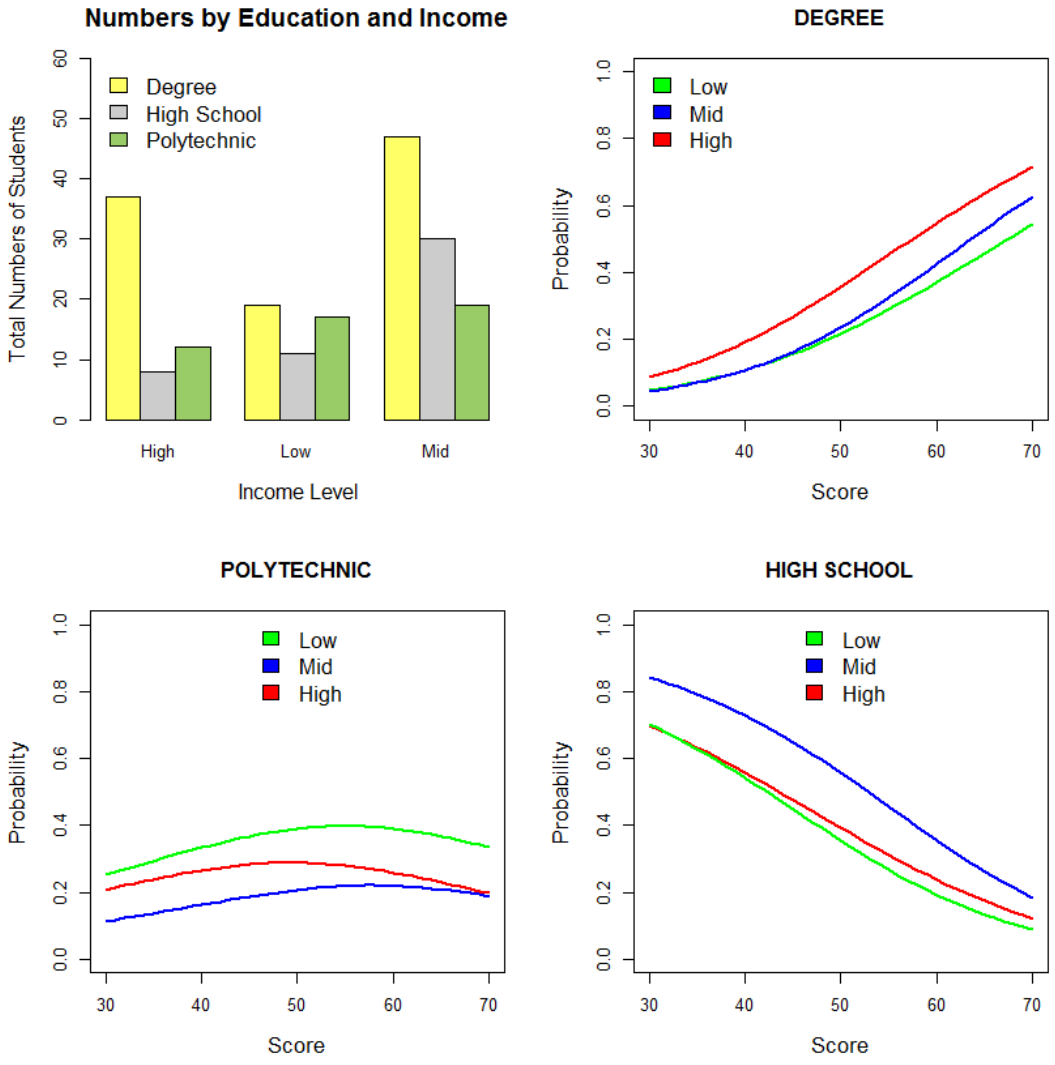

Figure 18: Multinomial model: terminating at high school, entering polytechnic, or attempting a degree.

workable predictive model that provides insight into the relative influences of income and test scores on educational outcome.

R for time series analysis

R provides a superb platform for analysing time series data (data that are measured over an extended period of time – usually, though not always, at regular intervals). Such data frequently involve economic measures, population statistics, industrial processes or business measures. Often we wish to model past data in order to understand the drivers of variability and to forecast future values of the time series. In making forecasts we assume that factors that influence the past and present will continue into the future.

R provides the `decompose()` function to isolate the trend component, seasonal component and random component of a seasonal time series (a time series that displays annual fluctuations). We now fit a time series model to an economic time series data set. In the observed data we can see an overall trend – the observed values increase over the period of interest. We also see recurring annual cycles – the seasonal variation.

In the graph of Figure 19 I have decomposed an economic time series data set (the observed data) into a trend, seasonal and random component.

The graph of Figure 19 gives the original time series (observed), the estimated trend component (the major movement that occurs over a period of more than one year), the estimated seasonal component (a recurring pattern that occurs annually), and the estimated random component (irregular and unpredictable movements that may be due to such factors as political events, industrial strikes or extreme weather).

Using R you can make short-term forecasts using various techniques. For example, in the graphs of Figure 20 I have used a technique known as Holt's exponential smoothing (available through the forecast contributed package) to model past data from another economic dataset and create a forecast.

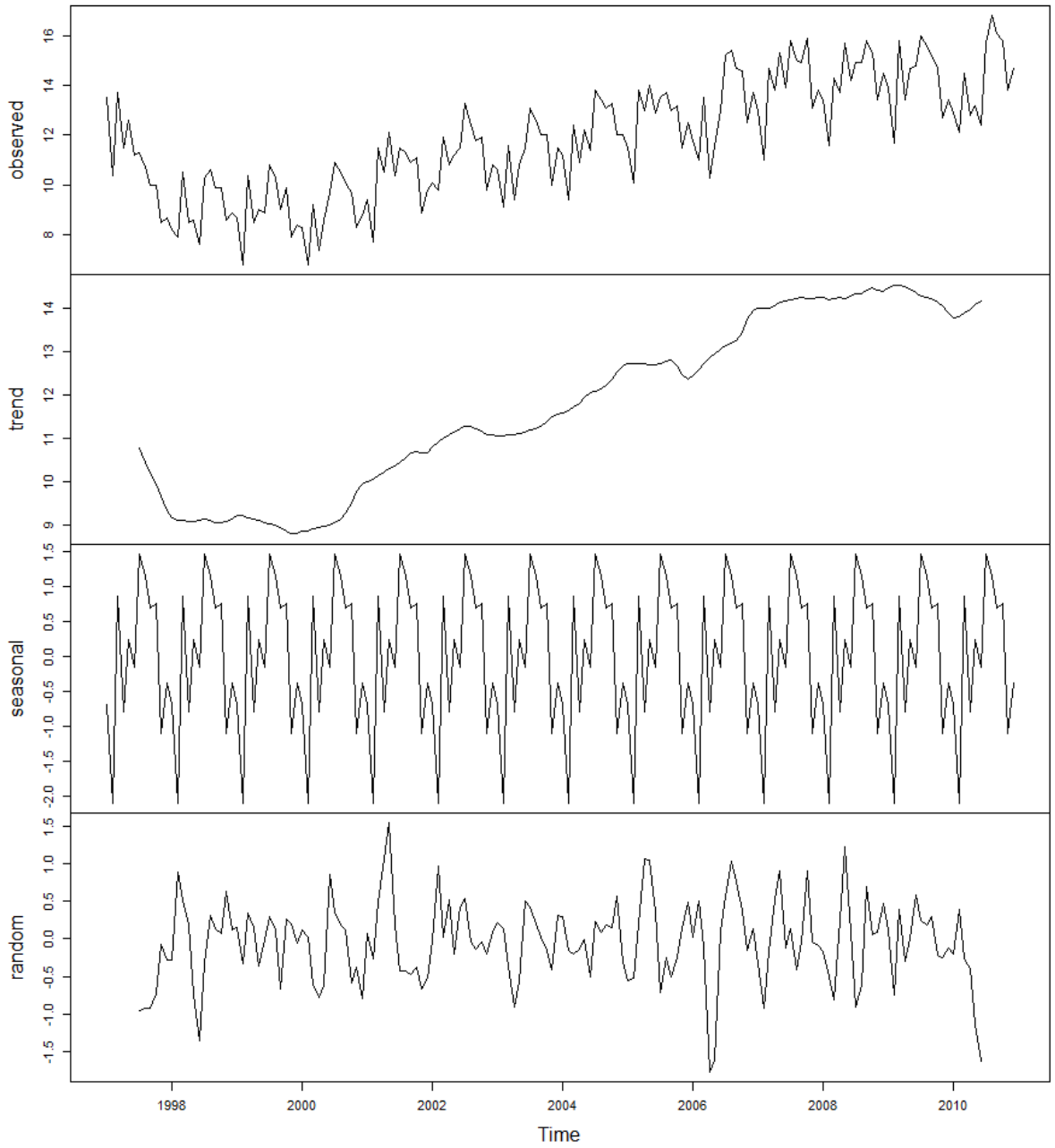

Figure 19: An economic time series and its trend, seasonal and random components.

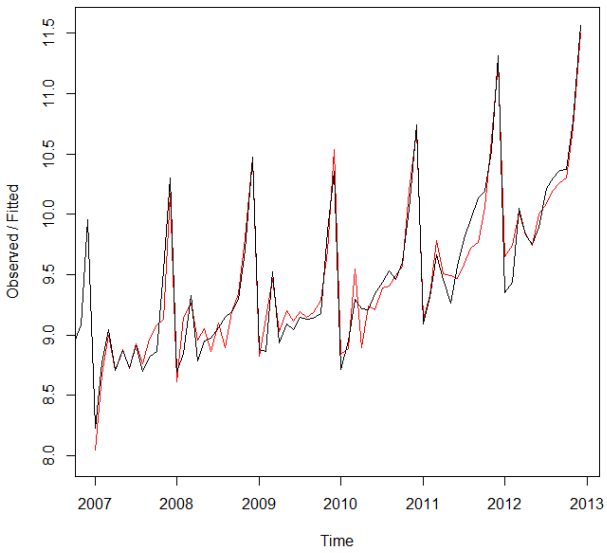

Figure 20: Modelling past data using Holt's exponential smoothing.

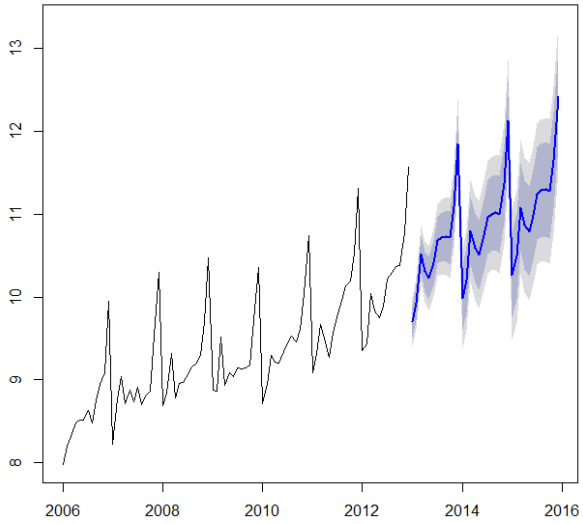

Figure 21: Forecasting using Holt's exponential smoothing.

The black curve of Figure 20 gives the original time series, while the red curve gives the fitted model values. Our model has done a pretty good, though not perfect, job of modelling the time series retrospectively. How well will it perform for forecasting future values? Figure 21 gives the forecast.

Here the forecasts are shown as a blue curve, and the shaded areas give 80% and 95% prediction intervals. The forecast appears reasonable, reflecting the overall trend and the annual fluctuations. Of course, the quality of the forecast can only be determined once the actual data for the forecasting period have been collected and other models may give a better prediction for our data. In any case Holt's exponential smoothing is only one of many useful functions available in R for modelling time series and diagnosing and improving your models.

Summary

In this article we have explored a small fraction of R's analytics and graphical capability and seen how the emergence of contributed software packages has made R so powerful and versatile. R continues to grow in popularity, and an increasingly diverse suite of contributed packages provide a unique platform for many areas of research and data analysis. It is already the application of choice for many professional statisticians around the world, and in the future it may become the application of choice for both the public and private sectors in New Zealand.

Dedication

I dedicate this article to the memory of my valued friend and colleague Telu Vaeau.

References

Analytics Vidhya, 2015. SAS launches a free version – but, is it good enough? Retrieved from: http://www.analyticsvidhya.com/blog/2014/06/sas-launches-free-version-but-good-enough/

Bioconductor Project: Bioconductor: Open Source Software for Bioinformatics. Retrieved from: http://www.bioconductor.org/

Fox, J. 2006. Structural Equation Modeling With the sem Package in R. *Structural Equation Modeling 13(3)*: 465–486. Lawrence Erlbaum Associates, Inc.

Grasman, R.P., van der Maas, H.L.J.; Wagenmakers, E.J. 2009. *Fitting the Cusp Catastrophe in R: A cusp-Package*. University of Amsterdam.

Hambleton, R.K.; Swaminathan, H.; Rogers, H.J. 1991. *Fundamentals of Item Response Theory*. Newbury Park, CA, Sage Publs.

Johnston, M.; Lillis, D. 2011. Statistical modelling and analysis of NCEA and New Zealand Scholarship assessment data. *New Zealand Science Review 68(4)*: 126–135.

Levy, M. 2008. Stock market crashes as social phase transitions. *Journal of Economic Dynamics and Control 32(1)*: 137–155.

Lillis, D. 2011. Use R for data analysis and research. *New Zealand Science Review 68(2)*: 73–79.

McCullagh, P.; Nelder, J.A. 1989. *Generalised Linear Models*. Springer – Science and Business. ISBN 10: 0412317605

ORACLE Learning R Series Session 1. 2012. Retrieved from: http://www.google.co.nz/l?sa=t&rct=j&q=&esrc=s&source=web&cd=3&ved=0CCkQFjAC&url=http%3A%2F%2Fwww.oracle.com%2Ftechnetwork%2Fdatabase%2Foptions%2Fadvanced-analytics%2Fr-enterprise%2Fore-trng1-gettingstarted-1501628.pdf&ei=hBL2VMLOMYaG8QWizIH4BQ&usg=AFQjCNGznIg-eMzSSosHNtXeDZtiuriepQ

R Foundation for Statistical Computing. Undated. *R: A language and environment for statistical computing.* R Foundation for Statistical Computing, Vienna. ISBN 3-900051-07-0. http://www.R-project.

Revolution R Enterprise: http://www.revolution-computing.com/revolution-r-enterprise

Rizoupoulos, D. 2013. Retrieved from: www.r-project.org/conferences/useR-2008/slides/Rizopoulos.pdf

Rosseel, Y. 2012. lavaan: An R Package for Structural Equation Modeling. *Journal of Statistical Software 48(2)*: 1–36. http://www.jstatsoft.org/v48/i02/

RStudio. http://www.rstudio.com

Sheffer, M. 2009. *Critical Transitions in Nature and Society.* Princeton University Press. ISBN 9780691122045.

Wellington R Users Group: http://www.meetup.com/Wellington-R-Users-Group-WRUG/