Arc, Version 1.05 July, 2004

Sanford Weisberg

Department of Applied Statistics, University of Minnesota, St. Paul, MN 55108-6042.

Supported in part by National Science Foundation Grants DUE 96-52887 and DMS 01-03983

July 8, 2004

Abstract

Arc is a computer program for the analysis of regression problems. It is described in Cook and Weisberg (1999) *Applied Regression Including Computing and Graphics*; see `www.stat.umn.edu/arc`. This document describes additions to the program since the publication of the book, and in particular describes the changes from Version 1.04 to Version 1.05.

The *Arc* computer package for regression problems is described in Cook and Weisberg (1999). A new Version 1.05 was released in summer 2004. Three new features will be visible to all users:

- The *regression dialog* for linear and generalized linear models has changed.
- A new class of *generalized nonlinear models* have been added to the program.
- A new call of *partial one-dimensional models* have been added to the program.

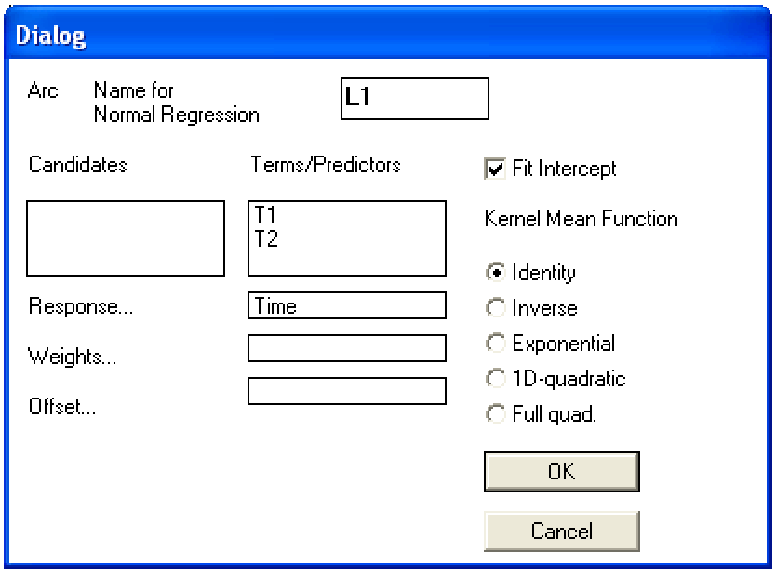
We describe these changes here.

1 The regression dialog

The Graph&Fit menu in *Arc* includes separate menu items for fitting linear LS regression, or fitting with a binomial, Poisson or Gamma distributed response. All these allow you to specify the response, weights, and terms to be put in the mean function. By selecting the appropriate item in the Graph&Fit menu, you specify the error distribution to be used. Buttons at the right of the dialog allow setting the *kernel mean function*, or kernel mean function; most often the default kernel mean function is used, so the buttons at the right can be ignored. It is the buttons at the right that have changed.

For an example of the regression dialog, we will use the data file `transact.lsp`. Figure 1. The new version of the dialog is shown in Figure 2

The only part of the dialog that has changed is the list of options at the right of the dialog. For fitting "usual" linear models (or generalized linear models with canonical links) these buttons are not used. They are used for more complicated models.


Dialog

Arc Name for Normal Regression: L1

Candidates:

Terms/Predictors: T1, T2

Response...: Time

Weights...:

Offset...:

[✓] Fit Intercept

Kernel Mean Function: [✓ Identity] Inverse Exponential 1D-quadratic Full quad.

Figure 1: The regression dialog from Version 1.04.

1.1 Components of a generalized linear model

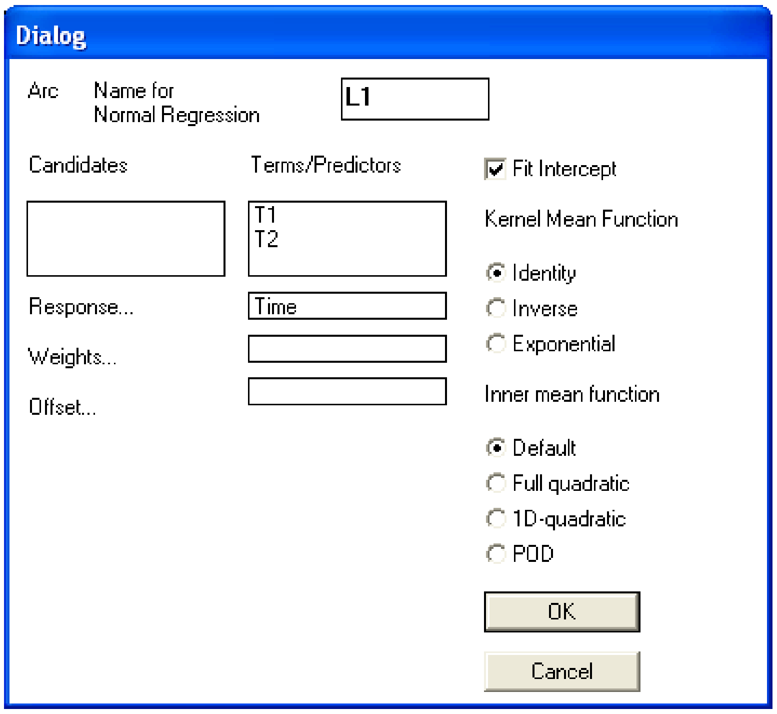
We begin with a brief recap of the main features of a generalized linear model. Suppose we have data $(x_i, y_i), i = 1, \ldots, n$, where x_i is a $p \times 1$ vector of predictors, and y_i is a univariate response. Using somewhat non-standard notation that is consistent with Cook and Weisberg (1999, Chapter 23) but not with most other references, the basic structure of a generalized linear model, or GLM, is as follows:

1. **Random component** The conditional random variables $y_i|x_i$ are independent observations, distributed according to an exponential family distribution. The common members of this family are the normal, binomial, Poisson, and gamma distributions. In *Arc*, you select a different fitting method from the Graph&Fit dialog for each of these distributions.

2. **Linear predictor** The conditional distributions depend on the value of x_i only through the linear combination $g(\theta, x_i) = \theta_0 + \theta' x_i$ for a $p \times 1$ parameter θ, and an intercept term θ_0 (with $\theta_0 = 0$ in some applications). The function $g(\theta, x_i)$ is usually called the *linear predictor*. In *Arc*, you specify the linear predictor and the response by moving items from the left-list to the right-list in the regression dialog.

3. **Kernel mean function** The dependence of the distributions $y_i|x_i$ on $g(\theta, x_i)$ is through the mean. In particular, there is a kernel mean function m such that

$$\mathrm{E}(y_i|x_i) = \mathsf{m}(g(\theta, x_i))$$


Dialog

Arc Name for Normal Regression: L1

Candidates:

Terms/Predictors: T1, T2

[✓ Fit Intercept]

Kernel Mean Function: [✓ Identity] Inverse Exponential

Response...: Time

Weights...:

Offset...:

Inner mean function: [✓ Default] Full quadratic 1D-quadratic POD

Figure 2: The regression dialog from Version 1.05.

For example, if the conditional distributions are Poisson distributed, a common choice of m is the exponential function, so

$$\mathrm{E}(y_i|x_i) = \exp(g(\theta, x_i))$$

Taking logarithms of both sides of this equation gives

$$\log(\mathrm{E}(y_i|x_i)) = g(\theta, x_i)$$

In the case of generalized linear models, where $g(\theta, x_i) = \theta_0 + \theta' x_i$, this last equation is

$$\theta_0 + \theta' x_i = \log(\mathrm{E}(y_i|x_i))$$

Generalized linear models are usually specified in terms of a *link function*, which is just the inverse of the kernel mean function. This comes from the last equation, where we see that the linear combination of the predictors is related to the mean via the link function, which in this case is the logarithmic function. In *Arc*, the kernel mean function is selected from the buttons at the right of the regression dialog. The *canonical* kernel mean function, the one most commonly used, depends on the error distribution, as given in Table 1.

Generalized linear models are briefly introduced in Cook and Weisberg (1999, Chapter 23), and more completely in McCullagh and Nelder (1989). In the latter reference,

Table 1: Canonical kernel mean functions.

Error distribution	Canonical kernel mean function
Normal	Identity
Binomial	Logistic
Gamma	Inverse
Poisson	Exponential

the *link* function is discussed, which is just the inverse of the kernel mean function. For example, in Poisson models, the link function corresponding to the exponential kernel mean function is the log-link, while in binomial models the link function corresponding to the logistic function, given by $\mathrm{E}(y|x) = (1+\exp(g))^{-1}$ is the logit, $g = \log(\mathrm{E}(y|x)/(1-\mathrm{E}(y|x)))$.

1.2 New feature of the regression dialog

You can completely ignore the new features in the regression dialog and still fit linear and generalized linear regression models as in previous versions of Arc. The right-side of the regression dialog in Figure 2 now has *two* lists, one for selecting a kernel mean function, and one for selecting a new quantity that we call the *inner mean function*. Under the kernel mean function list, all the relevant choices for the kernel mean function are given. The inner mean function includes four choices. We describe them in the context of the transactions data with two terms T_1 and T_2:

Default Fit the specified model with terms $\eta_1 T_1 + \eta_2 T_2$. This is the usual choice.

Full quadratic Fit the model including all main effects, interactions and quadratic terms, so for the transactions data this will fit $\eta_1 T_1 + \eta_2 T_2 + \eta_{11} T_1^2 + \eta_{22} T_2^2 + \eta_{12} T_1 T_2$.

1D quadratic Using the linear LS item and the identity kernel mean function, this will fit a nonlinear model with mean function

$$\eta_0 + \eta_1 T_1 + \eta_2 T_2 + \theta_2(\eta_1 T_1 + \eta_2 T_2)^2$$

For other fitting methods, this is a generalized nonlinear model, Section 2.

POD Fit a *partial one-dimensional model*. This item is used only if there is a grouping variable available. In the transactions data, suppose we had a variable G which was equal to one for urban bank branches, and zero for suburban or rural branches. The POD model fits the following *nonlinear* combination of terms:

$$\eta_0 + \eta_1 T_1 + \eta_2 T_2 + G \times (\theta_0 + \theta_1(\eta_1 T_1 + \eta_2 T_2))$$

These models are described more completely in Section 3.

2 Generalized nonlinear models

A *generalized nonlinear model*, or GNM, allows nonlinear combinations of parameters in a statistical model that would otherwise have the form of a generalized linear model. The name was used by Lane (1996), although models in this class were considered earlier (e.g., Baker and Thompson, 1981, for a particular case, and Jorgensen, 1983, for theoretical properties).

A generalized nonlinear model is of exactly the same structure as a generalized linear model described in the last section, except that the *linear predictor* in the inner mean function. Rather than requiring the dependence of y_i on x_i to be completely determined by a linear predictor $\theta' x_i$, we generalize to allow $g = g(\theta, x_i)$ to be any smooth function of the vector valued parameter θ.

2.1 Examples

2.1.1 Logistic regression

Suppose we have a single predictor x and a 0-1 response y such that the distribution of $y|x$ is Bernoulli with probability $\Pr(y = 1|x) = \mathrm{E}(y|x)$ that is a function of x. If the standard logistic regression model were to hold, we would have that (Cook and Weisberg, 1999, Chapter 21)

$$\mathrm{E}(y|x) = \frac{1}{1 + \exp(\theta_0 + \theta_1 x)} \tag{1}$$

The right side of this equation is the logistic function, with argument given by the linear predictor $\theta_0 + \theta_1 x$. This equation can be rewritten as

$$\log\left[\frac{\mathrm{E}(y|x)}{1 - \mathrm{E}(y|x)}\right] = \theta_0 + \theta_1 x \tag{2}$$

showing that the log of the odds is a linear function of the linear predictor. Such a model would be fit using a GLM, with predictor x, with an intercept, using binomial errors and the logistic kernel mean function.

Now suppose that we believe that the logistic model is correct, but x must be transformed via an unknown power transformation for the model to hold. We would then have that:

$$\log\left[\frac{\mathrm{E}(y|x)}{1 - \mathrm{E}(y|x)}\right] = \theta_0 + \theta_1 x^{\theta_2} \tag{3}$$

The right side of this equation is no longer linear in the parameters, and so this is not a GLM. However, it can be treated as a GNM, with binomial errors, logistic kernel mean function, and nonlinear predictor given by the right side of (3).

In another problem, we might believe that

$$\mathrm{E}(y|x) = \theta_2 + \frac{1}{1 + \exp(\theta_0 + \theta_1 x)} \tag{4}$$

where perhaps θ_2 is a small positive value. This model asserts that there is a minimum threshold for $\mathrm{E}(y|x) = \Pr(y = 1|x)$ is θ_2 rather than the value zero given by the logistic function. Model (4) cannot be viewed as logistic regression, but it can be fit as a GNM, with binomial errors, the identity kernel mean function, and nonlinear predictor given by the right side of (4).

2.1.2 Michaelis-Menten formula

Suppose that y is the speed of a chemical reaction, and x is the concentration of an input agent in the reaction. The Michaelis-Menten formula for enzyme kinetics suggests that

$$\mathrm{E}(y_i|x_i) = \frac{\theta_1 x_i}{x_i + \theta_2} \tag{5}$$

In (5) we recognize a nonlinear mean function, since (5) is a nonlinear function of the parameters. Assuming normal errors, we can fit this model using the identity kernel mean function and nonlinear predictor given by the right side of (5).

Equation (5) can be rewritten by dividing both the top and the bottom of the right hand side by $\theta_1 x_i$:

$$\mathrm{E}(y_i|x_i) = \frac{1}{1/\theta_1 + \theta_2/(\theta_1 x_i)} \tag{6}$$

Fitting either mean function (5) or (6) is equivalent, as they are simply restatements of the same function. In (6), we can usefully think of the mean function as the inverse kernel mean function applied to the nonlinear predictor $1/\theta_1 + \theta_2/(\theta_1 x_i)$, so we can fit this model using normal errors, the inverse kernel mean function, and nonlinear predictor $1/\theta_1 + \theta_2/(\theta_1 x_i)$.

We could also fit (6) with normal errors, the identity kernel mean function, and using the right side of (6) as the nonlinear predictor. This will once again give the same solution, and this illustrates that there is not a unique way of defining the nonlinear predictor and the kernel mean function.

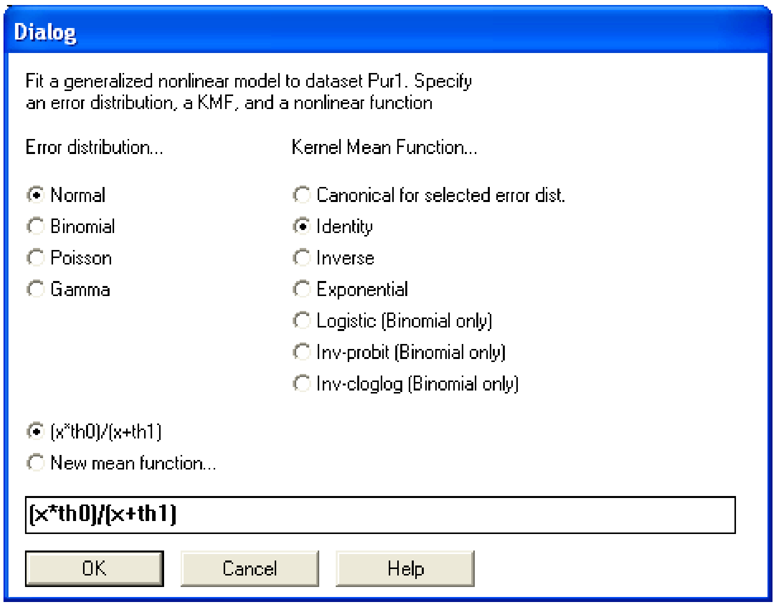
Carrying this further, define $\theta_1 = 1/\theta_1, \theta_2 = \theta_2/\theta_1$ and $u_i = 1/x_i$. Then (6) can be rewritten as

$$\mathrm{E}(y_i|x_i) = \frac{1}{\theta_1 + \theta_2 u_i} \tag{7}$$

This is once again the same as model (5), except for the reparameterization from the θs to the θs. Estimates of the θs can be obtained by substituting estimates for the θs in the equations that relate the two sets of parameters. Model (7) can be fit using using a GNM with normal errors, the inverse kernel mean function, and the linear predictors $\theta_0 + \theta_1 x_i$. However, we recognize (7) as a generalized *linear* model because the predictor is linear in the parameters, so GLM software can be used. The advantage of GLM software is that choosing starting values for the iterative procedure is not required, and selecting starting values is required for GNMs.

2.2 Using *Arc* for generalized nonlinear models

With Version 1.05, generalized nonlinear models are a standard part of *Arc*. As an example of the methodology, we use the file `pur.lsp` that is included with the *Arc*


Dialog

Fit a generalized nonlinear model to dataset Pur1. Specify
an error distribution, a KMF, and a nonlinear function

Error distribution...

[✓ Normal]
Binomial
Poisson
Gamma

Kernel Mean Function...

Canonical for selected error dist.
[✓ Identity]
Inverse
Exponential
Logistic (Binomial only)
Inv-probit (Binomial only)
Inv-cloglog (Binomial only)

[✓ (x*th0)/(x+th1)]
New mean function...

(x*th0)/(x+th1)

Figure 3: The first dialog for generalized nonlinear models.

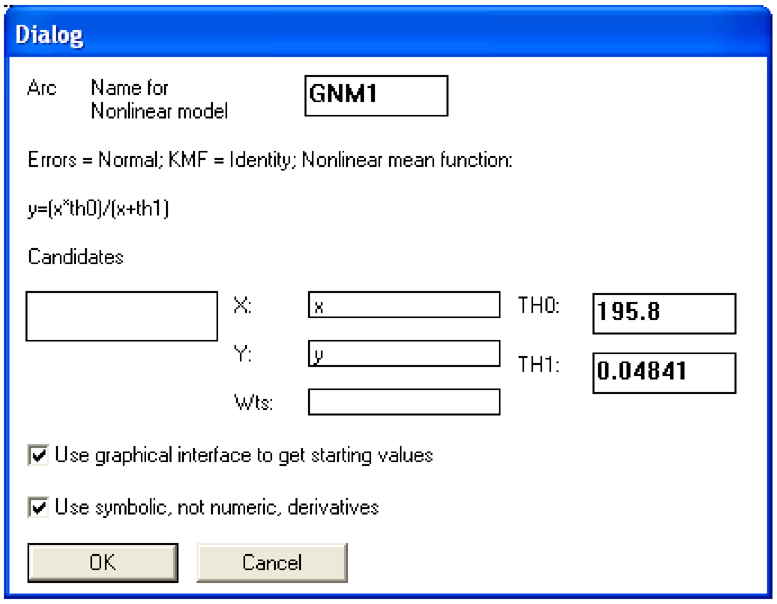
ddistribution. This is a small data set for which the Michaelis-Menten model is appropriate. To use the GNM methods, you need to have the following bits of information available to you:

1. The error distribution, which for these data is the assumption of normal errors.
2. The kernel mean function and the nonlinear predictor.
3. Starting values for the parameters to be estimated in the nonlinear function.

One of the reasons that generalized linear models have proven to be so popular is that reliable computing algorithms are available that do not require the user to provide starting values for parameter estimates. In GNMs, however, starting values are generally required, and good starting values are needed for the computing method to converge to the correct estimates. *Arc* provides numerous tools to help find starting values for normal nonlinear models, and these same tools can be used for GNMs as well. In addition, the GNM code provides a graphical tool for finding starting values when there is only one variable (but an arbitrary number of parameters).

To fit model (5), we notice that for very large values of x, $\mathrm{E}(y|x) \rightarrow \theta_1$, and so θ_1 is the asymptote or maximum value of the expected response. From a graph of y versus x, we can read this value to be about 210. Getting a starting value for θ_2, which is a rate parameter, is harder. We will use the graphical interface to help us choose a starting value.

From the Graph&Fit menu, select the item “Generalized nonlinear.” This will produce the dialog shown in Figure 3. This complicated dialog requires that you select an



Arc Name for Nonlinear model GNM1

Errors = Normal; KMF = Identity; Nonlinear mean function:

y=(x*th0)/(x+th1)

Candidates

X: x

Y: y

Wts:

TH0: 195.8

TH1: 0.04841

☑ Use graphical interface to get starting values

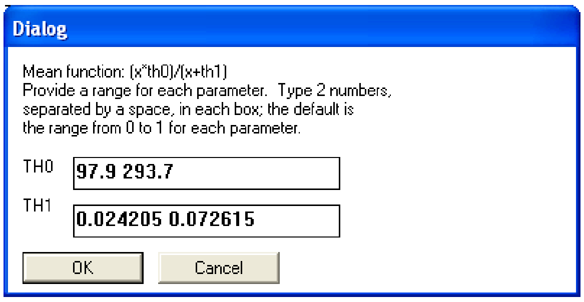
☑ Use symbolic, not numeric, derivatives

Figure 4: The second dialog for generalized nonlinear models.

error distribution from the left column, a kernel mean function from the right column, and a mathematical expression for the nonlinear predictor from the bottom of the dialog. If you have previously entered expressions for nonlinear predictors for this data set, or you have included expressions for nonlinear predictors on the data file, they will appear in a list at the bottom of the dialog, and you only need to select the one you want to use. If you want to use a new expression for the nonlinear predictor, type it in the text area. The expression should be in standard mathematical format, and needs to be an expression that could be read by the language C. The expression for a nonlinear predictor can consist of constants or numbers, parameters, and variables. Parameters are given be a prefix, optionally followed by a number. For example, for the prefix Th, names of parameters could be Th, Th1, Th2 or Th01. The allowable prefixes for parameters are Th, Theta, Gam, Gamma, Eta, A, Alpha, B, Beta, Del, Delta, Lam and Lambda. Variable names are also of the form of a prefix possibly followed by a letter, such as X, X1 or X01. The prefixes for variables are X, Z and U. In a later dialog, you will make a connection between the generic names of variables in this dialog and the names of variables in your dataset.

In Figure 3 the nonlinear predictor on the right side of (5) has been typed in the text area. This was not strictly necessary because this expression is given on the data file, and so appears as a choice in the dialog.

When you press the OK button in this dialog, you will get a second dialog shown in Figure 4. In this dialog you specify the variables in the dataset that correspond to the variables in the nonlinear predictor, and you specify starting values. There are two additional items at the bottom of the dialog. By default, *Arc* uses symbolic differentiation in computations, but you can use numerical derivatives by un-checking the



Dialog

Mean function: (x*th0)/(x+th1)
Provide a range for each parameter. Type 2 numbers, separated by a space, in each box; the default is the range from 0 to 1 for each parameter.

TH0 97.9 293.7

TH1 0.024205 0.072615

Figure 5: The third dialog for generalized nonlinear models.

box. Most users will not find this option particularly interesting. The other check-box appears only when exactly one variable appears in the nonlinear predictor. If the box is checked, then a graph is produced to help select starting values. If the box is not checked, then the starting values you provide are used to estimate parameters. With this example, the starting values were given in the data file, and are displayed automatically if available in the boxes for the parameters; we will ignore these for this example, and use the features of the program to choose starting values.

Leave the "graphical interface" button selected, and push the OK button. This produces a third dialog shown in Figure 5 to select ranges for the parameters to search over for starting values. This dialog appears only if the graphical interface button is selected. If you provided values for starting values in the second dialog, then the default ranges are centered on the values you provided; otherwise the range $(0, 1)$ is given. You can set a range here, or you can change the range later on the graph. Pushing the OK button will (finally) produce the graph shown in Figure 6. The deviance for the

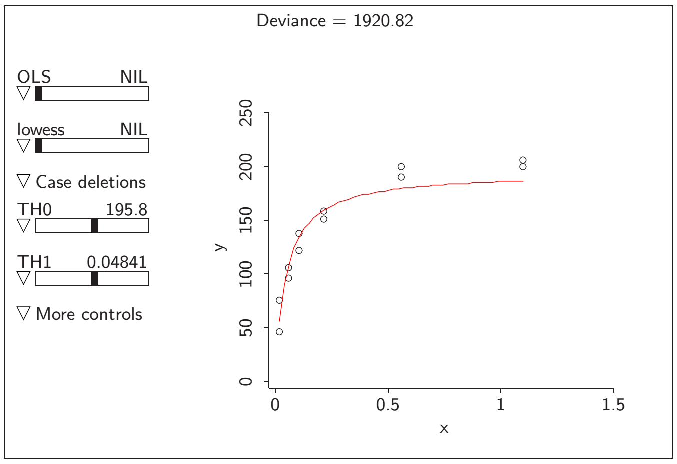


Figure 6: Figure to help select starting values. Use the sliders to change the value of the parameters.

current fit is shown at the top of the plot; the best estimates have the smallest deviance. The sliders can be used to vary the fit to the data, as shown on the graph. The pop-up menus for the sliders and for More controls can be used to work with this graph; all the items are self-explanatory. When you think you have good starting values, the “Fit model” item in the More controls pop-up can be used to fit the model.

2.3 The fitted model

The model is fit by default using the Newton-Raphson algorithm, similar to the fitting for nonlinear models in *Arc* (which is, unfortunately, not documented yet). At convergence, the fitted model is displayed. For the data in `pur.lsp`, the output is given in Table 2.

Table 2: GNM output.

```
Iteration 0, deviance = 1207.87467264
Iteration 1, deviance = 1195.60378495
Iteration 2, deviance = 1195.45030243
Iteration 3, deviance = 1195.44882830
Iteration 4, deviance = 1195.44881457
Reason for termination: Converged normally.

Data set = Pur1, Name of Fit = GNM4
Gnm-Normal Regression
Kernel mean function = Identity
Nonlinear mean function: y = (x*th0)/(x+th1)
Fit using symbolic derivatives.
Response      = y
Terms         = (x)
Coefficient Estimates
Label      Estimate        Std. Error     Est/SE    p-value
TH0        212.684         6.94715        30.615     0.0000
TH1        0.0641210       0.00828092      7.743     0.0000

Scale factor:             10.9337
Number of cases:              12
Degrees of freedom:           10
Pearson X2:             1195.449
Deviance:               1195.449
Reason for termination: Converged normally.
```

Fitting a GNM will also produce a model menu that has the same items as the model menu for nonlinear models.

2.4 Computations

The basic computations are similar to those in generalized linear models, and we will follow in part the derivation of the Fisher scoring algorithm from these models given by Agresti (1990, Sec. 13.1.1). Some of the details are omitted, but can be obtained from this reference. Be warned, however, that the Greek letters used in this development do not correspond exactly to those that Agresti uses.

2.4.1 Likelihood

We first assume that the probability function for $y_i|x_i$ is of the exponential family form

$$f(y_i; \psi_i, \phi) = \exp\{[y_i\psi - b(\psi_i)]/a_i(\phi) + c(y_i, \phi)\} \tag{8}$$

The parameter ψ is called the *natural parameter*. The function $a_i(\phi)$ is a dispersion function, and in all the cases we consider it is of the form $a_i(\phi) = \phi/w_i$ for known prior weights w_i. This development includes the normal, gamma, Poisson and binomial distributions as special cases, depending on the choice of the functions $b()$ and $c()$. The specification of the error distribution given here is the same as for generalized linear models.

The next step in setting up the model is to connect y_i to x_i, through the mean function. First, the log-likelihood for a single observation is

$$\ell(\psi_i, \phi; y_i) = [y_i\psi - b(\psi_i)]/a_i(\phi) + c(y_i, \phi) \tag{9}$$

The usual likelihood results

$$\mathrm{E}\left(\frac{\partial \ell}{\partial \psi}\right) = 0 \text{ and } \mathrm{E}\left(\frac{\partial^2 \ell}{\partial \psi^2}\right) = -\mathrm{E}\left(\frac{\partial \ell}{\partial \psi}\right)^2$$

imply that the mean function and the variance function are

$$\begin{aligned} \mathrm{E}(y_i|x_i) &= \mu_i = b'(\psi) \\ \mathrm{Var}(y_i|x_i) &= b''(\psi)a(\phi) \end{aligned}$$

where the primes denote differentiation. As with a generalized linear model, the connection between y_i and x_i is through the mean function, which we write as

$$\mu_i = \mathrm{E}(y_i|x_i) = \mathsf{m}(g(\theta, x_i)) \tag{10}$$

where as before m is the kernel mean function and g is a nonlinear function of the parameters θ. In a generalized linear model, the function $g(\theta, x_i) = \theta_0 + \theta'\mathbf{x}_i$, is linear in the parameters, but we make so such a restriction here. The role of the kernel mean function in (10) is the same in generalized linear and nonlinear models.

2.4.2 Derivatives

Computational methods are based on derivatives of the likelihood and the information matrix. As with generalized linear models, derivatives are found using the chain rule,

$$\frac{\partial \ell_i}{\partial \theta_j} = \frac{\partial \ell_i}{\partial \psi_i}\frac{\partial \psi_i}{\partial \mu_i}\frac{\partial \mu_i}{\partial g_i}\frac{\partial g_i}{\partial \theta_j} \tag{11}$$

where by g_i we mean the expression $g(\theta, x_i)$. We compute each of the four terms on the right of (11) separately. First, $\partial\ell_i/\partial\psi_j = [y_i - b'(\psi)]/a_i(\phi) = [y_i - \mu_i]/a_i(\phi)$. This depends on the exponential family given in (8) through the mean function. Next, $\partial\psi_i/\partial\mu_i = (\partial\mu_i/\partial\psi_i)^{-1} = (b''(\psi_i))^{-1} = [\text{Var}(y_i|x_i)/a_i(\phi)]^{-1}$, which depends on the particular exponential family through the variance function.

For the third partial derivative, we use the representation derived by inverting (10),

$$g(\theta, x_i) = \mathsf{m}^{-1}(\mu_i)$$

The function m^{-1} is called the *link function*. The derivative $\partial g_i/\partial\mu_i$ is therefore the partial derivative of the link function with respect to its argument, evaluated at μ_i. For example, in Poisson regression with the log-link, we have $g_i = \log(\mu_i)$, and the derivative $\partial g_i/\partial\mu_i = 1/\mu_i$. Everything so far is a standard computation with generalized linear models. The remaining term is different.

The final partial derivative is just the first derivative of the mean function with respect to the parameter. This is a standard calculation with nonlinear models, and depends on the nonlinear function g, but not on the error distribution or the link function. Combining the four components, we get

$$\frac{\partial\ell_i}{\partial\theta_j} = \frac{y_i - \mu_i}{\text{Var}(y_i|x_i)} \left(\frac{\partial g_i}{\partial\mu_i}\right)^{-1} \left(\frac{\partial g_i}{\partial\theta_j}\right) \tag{12}$$

The maximum likelihood estimates solve $\partial L(\theta)/\partial\theta = \sum \partial\ell_i/\partial\theta = 0$.

We will need to have a matrix expression for the derivatives $q = \partial(L(\theta))/\partial\theta$. This will be a $p \times 1$ vector. Let $\hat{\theta}$ be the current estimate of θ, and let V be the matrix whose elements are given by $\partial g_i/\partial\theta_j$ evaluated at $\hat{\theta}$. This is a standard component of nonlinear regression. Let e be a vector of residuals whose i-th element is given by $y_i - \mu_i$, with μ_i evaluated at $\hat{\theta}$. Next, define A to be a diagonal matrix with entries $(\partial g_i/\partial\mu_i)^{-1}$, and let W be a diagonal matrix with entries $1/\text{Var}(y_i|x_i)$. Both of these matrices have elements evaluated at $\hat{\theta}$. Then q is given by are then given by

$$q = V'AWe \tag{13}$$

2.4.3 Information

We need to compute an information matrix, both for the computing algorithm, and for asymptotic standard errors. It is easiest to use expected information, although a similar development with observed information, using second derivatives is possible and potentially useful. An element of the expected information for a single observation is

$$\begin{aligned} -\text{E}\left(\frac{\partial\ell}{\partial\theta_h}\right)\left(\frac{\partial\ell}{\partial\theta_j}\right) \\ &= -\text{E}\left[\tfrac{y_i - b'(\psi_i)}{\text{Var}(y_i|x_i)}\left(\tfrac{\partial g_i}{\partial\mu_i}\right)^{-1}\left(\tfrac{\partial g_i}{\partial\theta_h}\right)\right]\left[\tfrac{y_i - b'(\psi_i)}{\text{Var}(y_i|x_i)}\left(\tfrac{\partial g_i}{\partial\mu_i}\right)^{-1}\left(\tfrac{\partial g_i}{\partial\theta_j}\right)\right] \\ &= -\left(\tfrac{\partial g_i}{\partial\theta_h}\right)\left(\tfrac{\partial g_i}{\partial\theta_j}\right)\tfrac{1}{\text{Var}(y_i|x_i)}\left(\tfrac{\partial g_i}{\partial\mu_i}\right)^{-2} \end{aligned}$$

Given this last result, we can get a matrix form for the information matrix based on all n data points. Let $L(\theta)$ be the log-likelihood for the whole sample. Let V, A and W be as defined in the last section. The expected information matrix $\mathcal{I}$ is then

$$\mathcal{I} = V'AWAV$$

The expected information has exactly the same form for generalized linear models, except that the matrix V is replaced by the $n \times p$ matrix of predictors.

2.4.4 Fisher scoring algorithm

Given a current estimate $\theta^{(k)}$, we can compute the matrices V, W and A evaluated at the current estimate. By evaluating the mean function at $\theta^{(k)}$, we can also compute the residuals e. The Fisher scoring algorithm is then given by:

$$\begin{aligned}\theta^{(k+1)} &= \theta^{(k)} + (\mathcal{I})^{-1}q \\ &= \theta^{(k)} + (V'AWAV)^{-1}V'AWe \qquad (14)\end{aligned}$$

The only difference between (14) and the equivalent formulation for generalized linear models is that V is replaced by the data matrix. The generalized linear model computations are usually done using an iteratively reweighted least squares algorithm. Rewrite (14) as

$$\begin{aligned}\theta^{(k+1)} &= (V'AWAV)^{-1}V'AW(AV\theta^{(k)} + e) \\ &= (V'AWAV)^{-1}V'AWA(V\theta^{(k)} + A^{-1}e)\end{aligned}$$

The usual algorithm used for generalized linear models is based on this last equation. We recognize that the next iterate can be obtained using weighted least squares with “working weights” given by the diagonals of AWA and “working response” given by $V\theta^{(k)} + A^{-1}e$.

A somewhat different iteratively reweighted least squares algorithm is available directly from (14), and has been implemented in *Arc*. We view (14) as the new estimate is equal to the old estimate plus an increment. This is the form of the algorithm that *Arc* uses for nonlinear models. We redefine the first derivative method to return AV rather than just V. The increment for the next iterate is obtained by the weighted least squares of e on AV with weights given by the diagonals of W.

2.4.5 Illustrative example

Suppose we are fitting a Poisson regression with

$$\mathrm{E}(y|x) = \exp(\theta_0 + \theta_1 x_1/(1 + \theta_2 x_2))$$

We can view this mean function has having the exponential kernel mean function, $\mathsf{m}(g) = \exp(g)$, or equivalently as having the logarithmic link. Suppose we have a current estimate $\theta^{(k)}$ of θ. We then perform the following steps to get the next estimate of θ:

1. Obtain fitted values, given by $\mu_i = \exp(\theta_0^{(k)} + \theta_1^{(k)} x_1/(1 + \theta_2^{(k)} x_2))$.

2. Compute the matrix V. A typical row of this matrix is given by $(1, x_1/(1 + \theta_2 x_2, -\theta_1 x_1 x_2/(1 + \theta_2 x_2)^2)$, to be evaluated at $\theta^{(k)}$.

3. Compute the diagonal elements of A. For the log-link, the elements are $(1/\mu_i)^{-1} = \mu_i$.

4. Compute the elements of W, given by $1/\text{Var}(y_i|x_i)$. For the Poisson model, $\text{Var}(y_i|x_i) = \mu_i$, again evaluated at $\theta^{(k)}$.

Once these are computed, use (14) to get the updated estimate of θ. A convergence criterion is then checked, either based on the size of the increment, or based on the change in the deviance.

3 Partial one-dimensional models

Partial one-dimensional regression models were introduced by Cook and Weisberg (2004) as a simple way to describe regression mean functions that may depend on a grouping variable. For example, consider the data file `ais.txt`, which includes the lean body mass *LBM*, height *Ht*, weight *Wt* and red blood cell count *RCC* for 202 elite athletes we trained at the Australian Institute of Sport. Approximately half the athletes were male and have female. To include a *Sex* effect in a mean function could require including a term for *Sex* and terms for the interactions of *Sex* with each of the other predictors. Such a mean function complex, and difficult to interpret.

A POD model for this problem is given by:

$$\begin{aligned}\mathrm{E}(\mathit{LBM}|\mathit{Ht},,\mathit{Wt},\mathit{RCC},\mathit{Sex} &= \eta_0 + \eta_1 \mathit{Ht} + \eta_2 \mathit{Wt} + \eta_3 \mathit{RCC} + \\ &\quad \mathit{Sex} \times (\theta_{01} + \theta_{11}(\eta_1 \mathit{Ht} + \eta_2 \mathit{Wt} + \eta_3 \mathit{RCC}))\end{aligned}$$

This mean function specifies that: (1) there is one linear combination of the terms that is relevant for both sexes, and (2) in the plot of the response versus $\eta_1 \mathit{Ht} + \eta_2 \mathit{Wt} + \eta_3 \mathit{RCC}$, each sex can have its own slope and intercept.

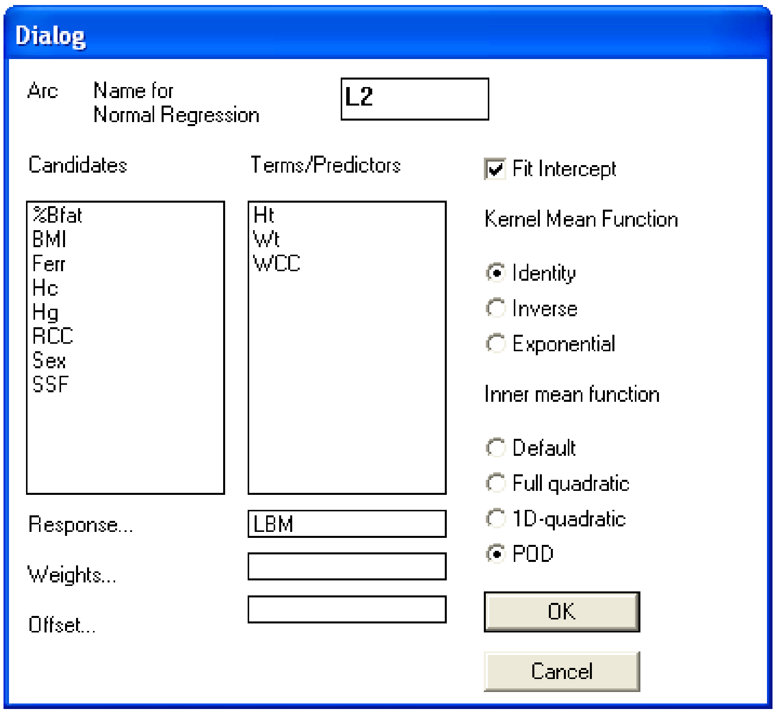
Fitting this model in *Arc* is very easy. Start with the regression dialog shown in Figure 7. Select the response and terms as usual, *except that the grouping variable Sex is excluded*. Select POD from the list at the right and push OK. You will then get the dialog in Figure 8. In this dialog, select *Sex* as the grouping variable, and push OK. The program will then fit the POD model, which requires a nonlinear fit (generalized nonlinear for other than normal errors with the identity kernel mean function). The output is shown in Table 3.

In the output, we see that both θ-parameters have small p-values, suggesting that both a separate slope and intercept are needed for each *Sex*; details of testing are discussed by Cook and Weisberg (2004). A POD model is a nonlinear model, and for the model's menu is the same as for a nonlinear model. In particular, an item "Fit subset model" allows refitting the POD model by, for example, setting $\theta_{01} = 0$, forcing the two sexes to agree at the origin.

Table 3: POD fit for the Australian athletes data.

```
Data set = AIS, Name of Fit = L1
Gnm-Normal Regression
Kernel mean function = Identity
Nonlinear mean function: y = eta0+eta1*x1+eta2*x2+eta3*x3+
                  G1*(th01+th11*(eta1*x1+eta2*x2+eta3*x3))
Fit using symbolic derivatives.
Response      = LBM
Terms         = (Ht Wt RCC)
Grouping var. = Sex
Coefficient Estimates
Label      Estimate        Std. Error     Est/SE    p-value
Eta0      -14.6565         6.46448        -2.267     0.0234
Eta1       0.146264        0.0342436       4.271     0.0000
Eta2       0.709342        0.0241639      29.355     0.0000
Eta3       0.724766        0.585402        1.238     0.2157
Th01       12.8473         3.76340         3.414     0.0006
Th11      -0.258750        0.0344636      -7.508     0.0000

Scale factor:             2.45979
Number of cases:              202
Degrees of freedom:           196
Pearson X2:            1185.911
Deviance:              1185.911
Reason for termination: Converged normally.
```


Dialog

Arc Name for Normal Regression: L2

Candidates: %Bfat, BMI, Ferr, Hc, Hg, RCC, Sex, SSF

Terms/Predictors: Ht, Wt, WCC

[✓] Fit Intercept

Kernel Mean Function: [● Identity] ○ Inverse ○ Exponential

Inner mean function: ○ Default ○ Full quadratic ○ 1D-quadratic [● POD]

Response...: LBM

Weights...:

Offset...:

Figure 7: The regression dialog to select a POD model

You can get a graph of the fitted POD model by selecting “Plot of” from the Graph&Fit menu, and then selecting the “Linear part” of the fitted model for the horizontal axis, the response on the vertical axis, and using *Sex* as a marking variable, the “Fit by groups–general” item in the parametric smoother slidebar shows the fitted response for each sex, as shown in Figure 9.

4tbh

4 Other changes

Page numbers below refer to the book.

1. The slicing mouse mode (pages 32 and 47) displays the within-slice sample size, mean and SD as the mouse is moved across a plot. If you change the setting `*boxplot-show-5-num-sumry*` to `t`, then on boxplots only the displayed summary statistics include the minimum, maximum, quartiles, median and sample size. If you want this to the the default on your system, change the setting `*boxplot-show-5-num-sumry*` to `t`, and select the item “Make change and save new setting file.”

2. (Revised December 20, 1999) Several features have been added to make communication between *Arc* and other statistical packages and spreadsheets eas-

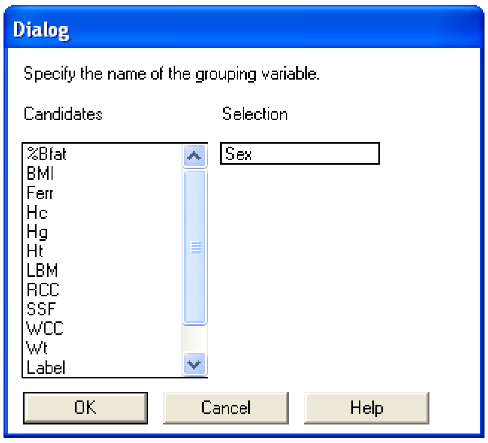


Figure 8: Select the grouping variable dialog

ier. We define an *interchange file* to be a plain file with headers (as described in Section A.5.2, page 555). Such a file can be read by many standard programs, including MS Excel, Minitab, JMP, SPSS and Splus. To create an interchange file, select "Display data" from the data set menu, then at the bottom of the resulting dialog select "Save data in an interchange file." The variables you select will be saved in the file you name in a subsequent dialog. There are four settings in the Settings menu to make this file easier to use. The setting `*interchange-header*` if set to `nil` will leave the variable labels off the file for use with programs that cannot read the first row as variable labels. The setting `*interchange-missing-indicator*` gives the missing value code. The default is `?`, as used in *Arc*, but this can be changed, for example, to `NA` for Splus, or to "." for SAS or JMP (SAS can use `?` for numeric variables but not for text variables). *However, an interchange file that uses "." for the missing value code cannot be read into Arc unless you use an editor to replace all the periods by something else, like* `?`.

The remaining two settings help with text variables. For example, suppose you have a text variable *City*, with values like "Los Angeles." By default, if *City* is written in an interchange file, the value for "Los Angeles" will appear as `Los_Angeles`, with the underline character _ substituted for any blanks, and without quotation marks around the name. If this file is read back into *Arc*, the value for Los Angeles will become `LOS_ANGELES`, since *Arc* translates all strings without quotation marks into all upper-case. Although not the best format for reading data back into *Arc*, this format is the default because it is suitable for most computer programs.

If you want the text-strings to be quoted (so *Arc* can correctly retain upper and lower case), set `*interchange-quotes*` to `t`. If you do not want blanks translated to _, set `*interchange-delete-blanks*` to `nil`.

3. (August 8, 2001) If you use both *Arc* and either Splus or R, here is how you can

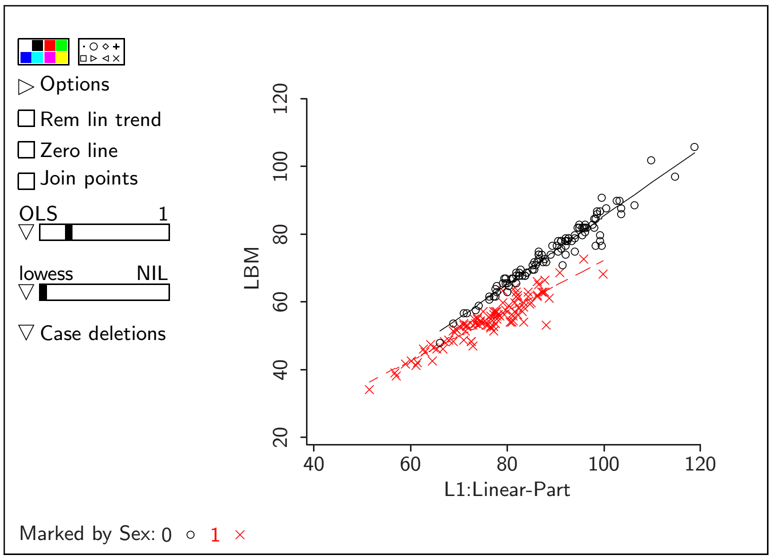


Figure 9: POD model for the Australian athletes data.

write a matrix (or a data frame) in either of these programs to a file that can be read directly by *Arc*. If you have a matrix called X, then the command in R or Splus

```
write.table(data.frame(X),"x.dat",quote=F,row.names=F,na="?")
```

will write a file called `x.dat` based on the matrix x. Strings will not be quoted, row names will not be included in the file, and the missing value character will be `?`. If X is already a data frame, then

```
write.table(X,"x.dat",quote=F,row.names=F,na="?")
```

will work.

4. The coordinates of the points in any graph can be saved in an interchange file, so the graph can then be redrawn, for example in a dedicated graphics package. If the graph is called `plot1`, simply type the command

   ```
   (send plot1 :points-to-file)
   ```

   You can avoid the typed command by adding an item to the plot's menu to save the data in the plot automatically. This can be done changing the setting `*graph-to-transport-file*` to `t`.

5. (Oct 1, 1999) A menu item on model menus allows saving some statistics like residuals and fitted values as variables. The names of the saved variables have been slightly changed so they no longer contain dashes, and can therefore be used in defining new variables.

6. (October 22, 1999) The `add-data-directory` function defined on page 548 now has an optional keyword `relative-to`. On Windows, suppose that you start *Arc* from the directory `c:\MyArc` and the library files (that is, all the files created by the installer) are in the directory `c:\Arc`. Most users will have the same name for these two directories. The following adds directories to the search path:

```
(add-data-directory "Mydata\" :relative-to 'home)
        adds c:\MyArc\Mydata\
(add-data-directory "Mydata\" :relative-to 'library)
        adds c:\Arc\MyData\
(add-data-directory "c:\" :relative-to nil)
        adds c:\
```

7. (November 10, 1999) An item called “Display regression summary” on the parametric smoother slidebar on scatterplots displays the regression summary for the graph in the text window. By default, LS fitting is used, but if a fitting method is selected on the slidebar along with a power of a polynomial, the summary for the model and method selected will be displayed. This method does not work with separate lines fit to marked groups. Only visible points are used in the fitting.

8. (May 26, 2000) An item has been added to the Case deletions plot control on each plot. This item is called “Create indicator for selected cases.” When this item is chosen, a group indicator variable (equal to 1 for each selected case and 0 for all others) is added to the data set. A setting called `*indicator-prefix*` can be set to choose the name of the new indicators.

9. (June 1, 2000) When using power curves (pages 319–320; 373–375), you can now have the power selected for you to be the least squares estimate by selecting the item “Estimate power via OLS” from the parametric smoother slide bar.

10. (August 1, 2000) A setting has been added to the Settings menu that will allow automatic saving of all **future** *Arc* text to a file. To use this option, change the setting as described below, select “Make change and save new settings file” from the settings dialog, and then restart the program. Setting this option effects future runs of the program, not the current run.

    The setting is called `*auto-save*`. If `*auto-save*` is `t`, a file called `transcript.txt` is saved in the Arc directory. If `*auto-save*` is set to `savefile.txt`, then the file of this name will be created and saved. **If you use either of these options, the current saved file will overwrite any existing file of the same name without warning!** If you want to be more careful, set `*auto-save*` to `ask`, in which case you will get a dialog at the beginning of your Arc run to name the transcript file; you will be prompted if you are overwriting an existing file. Finally, if the setting is equal to `prompt`, you will get a prompt before selecting a file name. This last setting is recommended for novice users, since getting a dialog without explanation at startup can be intimidating.

    This option is the same as the Dribble item in the Arc menu. Selecting Dribble from the file will stop saving the transcript file; see page 554.

11. (July, 2000) p-values for coefficient estimates are now reported along with the estimates and their standard errors in the Display estimates output. These are two-tailed p-values, computed from the t-distribution for linear models and from the normal distribution for all other models. The p-values can be removed by changing the setting `*display-coef-pvalues*` to `nil`.

    All remaining items require downloading the file updates.lsp and putting it in your extras directory.

12. (November 29, 2000) When using the “Display data” menu item in the dataset menu, a checkbox has been added to allow sorting the data before display. You can sort on any (or all) of the variables in the data. This effects only the display of the data, but does not permanently reorder the cases in the dataset.

13. (January 4, 2001) The function `clt-demo` can be used to reproduce the graphs in Figure 1.9 on page 19. For example, the command

    ```
    (clt-demo length 16 1000)
    ```

    will reproduce Figure 1.9c (you need to use the plot control to add the kernel density estimate). This function is available with updates.lsp dated January 4, 2001 or later.

5 References

Agresti, A. (1990). *Categorical Data Analysis*. New York: Wiley.

Baker, R. J. and Thompson, R. (1981). Composite link functions in generalized linear models, *Applied Statistics*, 30, 125–131.

Cook, R. D. and Weisberg, S. (1999). *Applied Regression Including Computing and Graphics,* New York: Wiley.

Cook, R. D. and Weisberg, S. (2004). Partial one-dimensional regression models. *American Statistician*, 58, 110–116.

Jorgensen, B. (1983). Maximum likelihood estimation and large sample inference for generalized linear and nonlinear regression models. *Biometrika*, 70, 19–28.

Lane, Peter W. (1996). “Generalized nonlinear models,”, *Compstat 1996: Proceedings in Computational Statistics, 12th symposium*, 331-336.

McCullagh, P. and Nelder, J. (1989). *Generalized Linear Models*. Boca Raton, FL: Chapman & Hall/CRC.