A Little Generalized Linear Models and a Lot of Logistic Regression

1 Some References

Around 1972, the analysis of categorical data was revolutionized by Nelder and Wedderburn who recognized that many disparate models for discrete data could be unified as special cases of the *generalized linear model (GLM)*. A big advantage of GLM's is that they make the analysis of discrete categorical data (which had been a bag of mostly seemingly-unrelated tricks) look much more like linear regression and ANOVA. For example, we can use our intutions about fitting models with `lm()` and `aov()` to inform our analysis using `glm()`[1] in SPLUS.

You can learn about log-linear models and logistic regression models from the following sources (among others!).

Agresti, A. (1990). *Categorical data analysis.* New York: Wiley.

A fully-detailed textbook on categorical data, generalized linear models, etc.

Chambers, J. M. and Hastie, T. J. (1992). *Statistical models in S.* Pacific Grove, CA: Wadsworth & Brooks/Cole. *[Especially Chapter 6.]*

A nice summary of what `glm()` can do.

Fox, J. (1997). *Applied regression analysis, linear models, and related methods.* SAGE publications, Thousand Oaks CA.

Chapter 15 contains a nice introduction to logit models for dichotomous and polytomous data, and GLM's.

Hosmer, D. W., Jr. and Lemeshow, S. (1989). *Applied logistic regression.* New York: Wiley.

From a customer review at `amazon.com`: "This book is widely considered the 'bible' of logistic regression analysis. It provides an accessible introduction to the theory of logistic modeling, and gives in-depth coverage of the proper use of the method, including interpretation, diagnostics, and practical considerations." *An updated edition appeared in 2000.*

McCullagh, P. and Nelder, J. A. (1989). *Generalized linear models, 2^{nd} edition.* London: Chapman and Hall.

This book really codified the revolution in discrete data analysis, by two of the main developers of the generalized linear model approach.

Rodrígues, G. (Fall 2001). *WWS509 Generalized Linear Models.* `http://data.princeton.edu/`

Online notes for a course in GLM's from a research demographer at Princeton. Very nice.

Venables, W. N. and Ripley, B. D. (1994). *Modern applied statistics with Splus.* New York: Springer-Verlag. *[Especially Chapter 7.]*

As usual, Venables and Ripley is better after you sort of know what is going on. But then it is a great reference and reminder of what is possible. *Current edition is 3^rd, some page number and chapter number references have changed...*

[1] Not to be confused with PROC GLM and the "general linear model" in SAS. PROC GLM simply fits linear regression and ANOVA models. The SAS procedure that corresponds to `glm()` in SPLUS is called PROC GENMOD.

2 The Challenger Data

You are probably all familiar with the Challenger Space Shuttle accident of January 20, 1986. I will leave out all the interesting substantive details about the accident and focus instead on the data about O-rings that was available at the time of the accident.

Well, I do need a little background, so here goes. The space shuttle uses two booster rockets and each rocket is made of several pieces whose joints are sealed by rubber O-rings. The O-rings are supposed to stop the hot gases that are inside the booster from escaping. Each booster contains 3 primary O-rings, for a total of six O-rings per flight. There was data about the number of O-rings out of six that failed on each of 23 previous space shuttle flights. Here is that data.

```
Temp Fail          Temp Fail
53   2             70   1
57   1             70   1
58   1             72   0
63   1             73   0
66   0             75   0
67   0             75   2
67   0             76   0
67   0             76   0
68   0             78   0
69   0             79   0
70   0             81   0
70   0
```

On the day of the fated flight, the forecast launch temperature was 31°F. The O-rings were known to be sensitive to temperature. The previous coldest launch temperature was $53°$. There was discussion about the cold temperature before launch, and the engineers did look at the data, but *only* the data for the failures[2]. They produced a plot that looked something like Figure 1.

```
> chal <- read.table("chal.dat", head=T)
> attach(chal)
> names(chal)
[1] "Temp" "Fail"

> plot(Temp[Fail >0], Fail[Fail>0])
```

Based on the erroneous assumption that the flights with no failures did not add any information, they noted that there did not appear to be any relationship between failures and temperature. Had they focused on a plot of all the data, as in Figure 2, they might have behaved differently.

```
> plot(Temp, Fail)
```

[2]Some engineers did look at more complete data, or at least saw that the likelihood of failure at 31°F was high; and implored management to delay the flight. The reasons that they were not listened to, and instead the plot of the failure-only data was used to justify liftoff, makes a fascinating study in (the failure of) scientific writing.

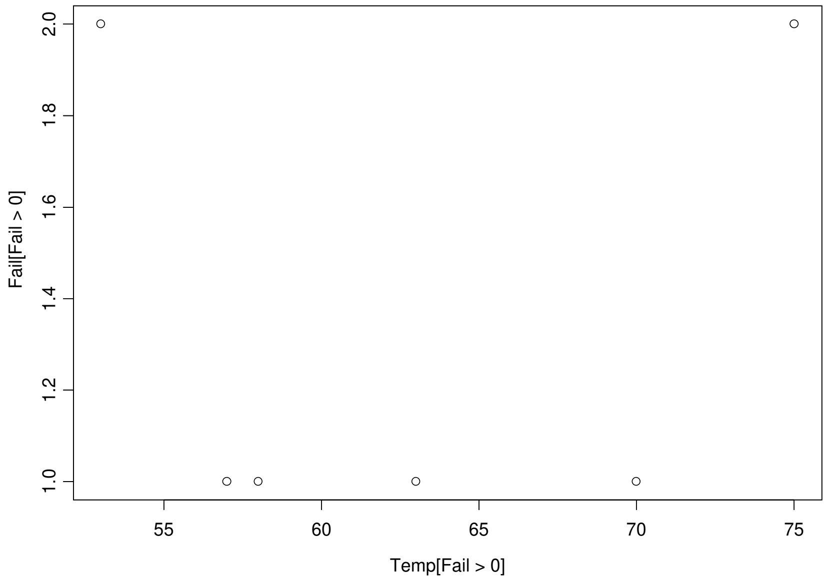

Figure 1: Plot of the number of O-ring failures including only space shuttle flights prior to Challenger in which failures occured.

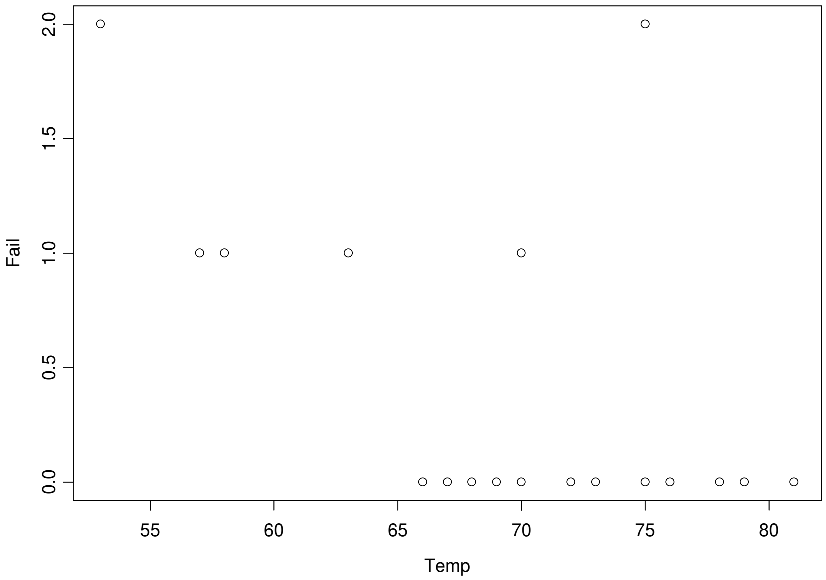

Figure 2: Plot of the number of O-ring failures including all space shuttle flights prior to Challenger, whether or not failures occured.

In addition before each flight the O-rings were given a pressure test. The pressure at which that test was conducted increased over time, and it is possible that the higher pressure test could have caused more failures. That data, and a lot more discussion, is available in the paper *Risk Analysis of the Space Shuttle: Pre-*Challenger *Prediction of Failure*, by S.R Dalal, E.B. Fowlkes, and B. Hoadley, in V. 84, pp.945–957, of the Journal of the American Statistical Association (JASA). **You** should be able to read most of that article. It is well written and easy to follow.

2.1 The logistic model

Obviously, we are most interested how probability of failure relates to temperature, and other quantities known before a flight. Given our data, it is not immediately obvious what response variable we should be looking at. For example, should we look at the number of failures of O-rings in a flight, or just the fact that an O-ring failed? These might lead to different results (but fortunately for us they can be looked at using the same analyses).

Instead of modelling $\underline{y}$ directly, we will model $p(t)$, the probability of failure given the temperature t. If one wanted to be really careful, one might write this as

$$p(F|t) = \text{probability of failure given temperature was } t$$

but the shorthand notation will probably suffice.

A first approach is to consider that the failures are a binomial process with the probability of failure dependent on t, but with the events themselves independent. Thus using the first form of the data (where we explicitly model the number of O-rings that failed) we are assuming that

- The 6 O-rings on any flight are fail independently (but with a common probability), and
- The 23 flights are independent.

The second assumption is more tenable than the first, but even the first seems quite reasonable. In order to fret about the first assumption, we will also analyze a version of the data that just records whether there was a failure or not.

We believe (at least I am suggesting) that the data can be modelled as follows. Let n be the number of trials that are done for each run of the experiment (in the Challenger case this is either 6 if each O-ring is considered or 1 if just overall success/failure is considered) and let r be the number of failures in those n trials. Let R be the corresponding random variable. Then we believe that

$$\begin{aligned} P(R = r|t) &= \begin{pmatrix} n \\ r \end{pmatrix} p(t)^r (1 - p(t))^{n-r} \\ &= \begin{pmatrix} n \\ r \end{pmatrix} \left(\frac{p(t)}{1 - p(t)}\right)^r (1 - p(t))^n \\ &= (\text{Fcn of } r) \times (\text{Fcn of } p \text{ and } r) \times (\text{Fcn of only } p) \end{aligned}$$

Likelihood theory (and common sense) suggests that the only thing of any importance here is the part that depends on both r and p. In other words, if you want to maximize the likelihood only the middle bit counts.

If we go one step further, and consider the log-likelihood

$$\log \begin{pmatrix} n \\ r \end{pmatrix} + r \times \log \left(\frac{p(t)}{1 - p(t)} \right) + n \times \log \left(1 - p(t)\right)$$

then we see that modelling might be easy and natural (and the asymptotic statistics might work) if we form models that are linear in

$$\log \left(\frac{p(t)}{1 - p(t)} \right)$$

That is exactly what we will do.

The function

$$\text{logit}\,(p) = \log \left(\frac{p}{1 - p} \right)$$

is called the logistic transformation. If

$$x = \text{logit}\,(p) = \log \left(\frac{p}{1 - p} \right)$$

then the inverse function is

$$p = \text{expit}\,(x) = \frac{\exp(x)}{1 + \exp(x)}$$

We will (obviously, from the name) model the logit of the probabilities as a function of the predictors.

2.2 Back to the Challenger Data

How do we want to model the challenger data? We (with the current data) only have temperature as a predictor, so we are stuck with models of the form,

$$\text{logit}(p(t)) = \text{some function of } t$$

The obvious first step is to plot the data (how can we do that?), then fit some simple models, then think about transformations. In this example we are particularly interested in extrapolations (dangerous though that is!) to temperatures that are less than we have seen.

2.3 Plotting the data

This is harder than one might imagine. One (mostly) can not just plot the logits, because much of the data is at the extremes where $p = 0$ or $(1 - p) = 0$. Often the best we can do is the plots that we did earlier, perhaps adding some sort of smooth line to the data. (Next example, we will try that). We will be able to do better once we have a model and can consider residuals—but be careful which residuals we think about.

2.4 The model

The S-Plus command to fit logistic regressions is `glm`, which also fits members of the more general class of models known as *generalized linear models*. To fit a logistic regression, `glm` needs to know both the number of successes and the number of failures, so we need to start out by creating such a variable,

```
> FN <- cbind(Fail, n-Fail)
```

Now, we just use the `glm` command, with the `family=binomial` option, to fit the model, viz.

```
> mymod <- glm( FN ~ Temp, family=binomial)
> mymod <- glm( cbind(Fail, n-Fail) ~ Temp, family=binomial)
```

Either of the above forms will work, but the first is easier in the long run.

The `family` option tells Splus what distribution the individual y's should have, and we have chosen `binomial` since that is the way our model was developed above. There are other `family` values for other kinds of data (poisson, Gaussian, inverse Gaussian, Gamma, etc.). By default the `binomial` family uses the `logit link function`, i.e. it builds a linear model for logit $p(y = 1|x)$. If you wanted to fit a *probit* model, for example, you would say `family=binomial(link = probit)` instead of just `family=binomial`.

Now let's look at the results of the fit,

```
> summary(mymod)

Call: glm(formula = FN ~ Temp, family = binomial)
Deviance Residuals:
       Min         1Q     Median          3Q      Max
 -0.9522773 -0.7829968 -0.5411832 -0.04378972 2.651492

Coefficients:
                 Value Std. Error   t value
(Intercept)  5.0848620  3.0477855  1.668379
       Temp -0.1155992  0.0469317 -2.463136

(Dispersion Parameter for Binomial family taken to be 1 )

    Null Deviance: 24.23036 on 22 degrees of freedom

Residual Deviance: 18.08633 on 21 degrees of freedom

Number of Fisher Scoring Iterations: 4

Correlation of Coefficients:
     (Intercept)
Temp -0.9932182
> anova(mymod)
Analysis of Deviance Table

Binomial model

Response: FN

Terms added sequentially (first to last)
```

```
     Df Deviance Resid. Df Resid. Dev
NULL                      22   24.23036
Temp  1 6.144035          21   18.08633
```

So what does all that mean? And what's a "deviance"??

2.5 Maximum likelihood, standard errors, IRLS

Likelihood equations. `glm` is not just doing a least-squares fit, but actually maximizing the likelihood. Let's re-cap the model, and see how the fit should work. Let

$$X = \begin{bmatrix} 1 & t_1 \\ 1 & t_2 \\ \vdots & \vdots \\ 1 & t_{23} \end{bmatrix} = \begin{bmatrix} \underline{x}_1^T \\ \underline{x}_2^T \\ \vdots \\ \underline{x}_{23}^T \end{bmatrix}$$

and recall that our model has the form

$$p_i \equiv P[\mathit{failure}|\underline{x}_i^T] = p(\underline{x}_i^T\underline{\beta}) = \frac{\exp[\underline{x}_i^T\underline{\beta}]}{1+\exp[\underline{x}_i^T\underline{\beta}]}$$

and therefore if we experience y_i failures in n_i attempts ($n_i \equiv 6$ for us), $i = 1, \ldots, N = 23$, we have

$$\begin{aligned} P(R = y_i|\underline{x}_i^T) &= \binom{n_i}{y_i} p_i^{y_i}(1-p_i))^{n_i-y_i} \\ &= \binom{n_i}{y_i}\left(\frac{p_i}{1-p_i}\right)^{y_i}(1-p_i)^{n_i} \\ &= \binom{n_i}{y_i}\exp[y_i \cdot \underline{x}_i^T\underline{\beta}](1+\exp[\underline{x}_i^T\underline{\beta}])^{-n_i} \end{aligned}$$

where in the last line I have applied the useful identity $1/(1-p_i) = 1 + p_i/(1-p_i)$. It follows that the log-likelihood for all $N = 23$ shuttle flights is

$$\begin{aligned} \log L(\underline{y}|X) &= \log\prod_{i=1}^{N} P(R = y_i|\underline{x}_i^T) \\ &= \sum_i \log\binom{n_i}{y_i} + \sum_i y_i \cdot \underline{x}_i^T\underline{\beta} - \sum_i n_i\log(1+\exp[\underline{x}_i^T\underline{\beta}]) \end{aligned}$$

Differentiating with respect to $\underline{\beta}$ we obtain

$$\frac{\partial \log L}{\partial \underline{\beta}} = \sum_i y_i\,\underline{x}_i - \sum_i n_i\frac{\exp[\underline{x}_i^T\underline{\beta}]}{1+\exp[\underline{x}_i^T\underline{\beta}]}\,\underline{x}_i$$

Setting this equal to zero to obtain the likelihood equations, and noting the definition of $p_i = p(\underline{x}_i^T\underline{\beta})$ above, we get

$$\sum_i n_i\,p_i\,\underline{x}_i = \sum_i y_i\,\underline{x}_i$$

Since $n_i p_i = E[y_i | \underline{x}_i^T]$, we have the usual maximum likelihood intepretation that this equates the observed sufficient statistics $\sum_i y_i \underline{x}_i$ with their expected values.

It is also worth noting that the likelihood equations above can be written as

$$X^T \underline{f}(X\underline{\beta}) = X^T y, \tag{1}$$

where

$$\underline{f}(X\underline{\beta}) = \begin{bmatrix} n_1 p(\underline{x}_1^T \underline{\beta}) \\ n_2 p(\underline{x}_2^T \underline{\beta}) \\ \vdots \\ n_N p(\underline{x}_N^T \underline{\beta}) \end{bmatrix}$$

is the vector of "fitted values" ($\hat{y}_i = f_i = p(\underline{x}_i^T \underline{\beta})$), and this is essentially like the normal equations of OLS regression,

$$X^T X \underline{\beta} = X^T y,$$

where $\underline{\hat{y}} = X\underline{\beta}$ is the vector of fitted values for the linear regresion.

Asymptotic standard errors. Once we have found the MLE's $\hat{\underline{\beta}}$ by solving (1), their asymptotic information matrix can be calculated as usual as

$$I(\underline{\beta}) = -E\left[\frac{\partial^2 \log L}{\partial \underline{\beta} \partial \underline{\beta}^T}\right] = \text{[many steps omitted]} = X^T V X$$

where

$$V = \begin{bmatrix} n_1 p_1(1-p_1) & 0 & \cdots & 0 \\ 0 & n_2 p_2(1-p_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & n_N p_N(1-p_N) \end{bmatrix}$$

and hence the asymptotic variance-covariance matrix of $\hat{\underline{\beta}}$ is

$$\Sigma_\beta = \text{Var}\,(\hat{\underline{\beta}}) = (X^T V X)^{-1}$$

This looks exactly like the variance-covariance matrix for $\hat{\underline{\beta}}$ in a WLS regression. It leads immediately to asymptotic confidence ellipsoids for linear functions $L\underline{\beta}$ of the parameters, based on the so-called *Wald statistics* [see Appendix D.6.2 in Fox, or section 15.3 of Rawlings et al., for example]

$$(L\underline{\beta} - L\underline{\beta}_0)^T [L \Sigma_\beta L^T]^{-1} (L\underline{\beta} - L\underline{\beta}_0),$$

and similarly for hypothesis tests, confidence intervals for individual parameters, approximate Scheffe' corrections for multiple inferences, etc., etc.

Finding the MLE's. Solving the likelihood equations (1) requires some sort of iterative method. "Newton's method" from elementary calculus for finding a root $\hat{\beta}$ of the equation $\ell(\beta) = 0$ by iterating

$$\beta_{k+1} = \beta_k - \ell(\beta_k)/\ell'(\beta_k)$$

can be generalized to the multivariate-β case, and the case in which $\ell(\underline{\beta}) = \partial \log L(\underline{\beta})/\partial\underline{\beta}$ is the derivative of the log-likelihood. In the case of logistic regression, this boils down (after much fooling around) to iterating

$$\underline{\beta}_{k+1} = \underline{\beta}_k + (X^T V_k X)^{-1} X^T (\underline{y} - f(X\underline{\beta}_k))$$

where V_k is defined as V above, but using $\underline{\beta}_k$. Note that this is a plausible thing to do, since when β_k has converged to a constant value $\underline{\beta}_k \equiv \hat{\underline{\beta}}$, we get

$$\hat{\underline{\beta}} = \hat{\underline{\beta}} + (X^T V X)^{-1} X^T (\underline{y} - \underline{f}(X\hat{\underline{\beta}})) \tag{2}$$

so that (1), $X^T f(X\hat{\underline{\beta}}) = X^T \underline{y}$, is in fact satisfied.

Note that (2) may be rewritten as

$$\hat{\underline{\beta}} = (X^T V X)^{-1} X^T V \underline{y}^*$$

where

$$\underline{y}^* = X\hat{\underline{\beta}} + V^{-1}(\underline{y} - \underline{f}(X\hat{\underline{\beta}}))$$

which suggests that an alternate way to iterate a solution of the likelihood equations is

$$\begin{array}{rrcl} \text{Let} & \underline{y}_k^* & = & X\underline{\beta}_k + V_k^{-1}(\underline{y} - \underline{f}(X\underline{\beta}_k)) \\ \text{and then} & \underline{\beta}_{k+1} & = & (X^T V_k X)^{-1} X^T V_k \underline{y}_k^* \end{array}$$

Note that at each stage, $\underline{\beta}_{k+1}$ are the estimated regression coefficients from a WLS problem with weight matrix $V_k^{-1/2}$. This method is called *iteratively reweighted least squares* (IRLS) and is a standard way to calculate MLE's for generalized linear models; it is in fact exactly what `glm` is doing in Splus.

2.6 Evaluating the model

Model deviance. Since we are out of the realm of least-squares (except as a computational tool, as with IRLS above), the F-testing approach using $c \cdot (RSS_{small} - RSS_{big})/RSS_{big}$ to model fit is of limited use in logistic regression (and glm's more generally). A useful generalization of RSS (it is essentially equivalent for OLS regression with normal errors) is the *model deviance*

$$G^2 = -2 \log L$$

where $\log L$ is the log-likelihood we were working with above. Exploiting the analogy between G^2 and RSS,

$$R^2 = 1 - \frac{G^2}{G_0^2}$$

is a reasonable analogy for R^2, where G^2 is the deviance of the model we are interested in, and G_0^2 is the deviance of the intercept-only model.

Using classical MLE theory we also know that for two nested models, "big" and "small",

$$G^2_{small} - G^2_{big} = -2 \log \frac{L_{small}}{L_{big}} \sim \chi^2_{df}$$

where *df* is the difference in the number of parameters in the two models (really, the difference in ranks of the corresponding X matrices).

Residuals. A basic reference for evaluating the fit of logistic regression models is Pregibon (1981, *Annals of Statistics, 9,* 705–724).

The raw residuals

$$E_i = y_i - n_i p_i$$

are not very useful in logistic regression, since their variance $n_i p_i(1-p_i)$ varies greatly with $p_i = p(\underline{x}_i^T \hat{\underline{\beta}})$ [and n_i]. The *Pearson residuals*

$$Z_i = \frac{E_i}{\sqrt{n_i p_i (1-p_i)}}$$

have more stable variances; and the sum of their squares

$$Z^2 = \sum_i Z_i^2$$

provides a measure of goodness-of-fit analogous to G^2. Recall that

$$G^2 = -2\log L = -2\sum_i [y_i \log p_i + (n_i - y_i)\log(1-p_i)];$$

this is a measure of the goodness-of-fit of the model also, since it implicitly contrasts the fit of the model with the fit of a model that fits the data perfectly (so that $L = 1$ and $\log L = 0$). The contribution of individual observations to G^2 is sometimes written as

$$G_i = \mathtt{sign}(E_i)\sqrt{-2[y_i \log p_i + (n_i - y_i)\log(1-p_i)]};$$

these G_i are called *deviance residuals.*

Pearson residuals and deviance residuals are the most commonly used for diagnosing the fit of a logistic regression. The default diagnostic plots for logistic regression (and any glm) in Splus involve plots of both deviance residuals and Pearson residuals.

```
> par(mfrow=c(2,2))
> plot(mymod)
```

The result is plotted in Figure 3.

Notice that the residual plot has at least two recognizable components, but it couldn't help but look like this. There is perhaps some suggestion that a function of temperature might be needed, like perhaps t^2, but the evidence is less than clear. The places where there are big residuals are for temperatures where there were both no failures, and failures.

Another kind of residual that is sometimes used is the *working residuals*; these are the terms $V^{-1}(\underline{y} - \underline{f}(X\hat{\underline{\beta}}))$ in the working y vector

$$\underline{y}^* = X\hat{\underline{\beta}} + V^{-1}(\underline{y} - \underline{f}(X\hat{\underline{\beta}}))$$

from IRLS. All three kinds of residuals can be obtained in Splus using the `residuals(glm.object, type=[your choice])` command, where "`[your choice]`" should be one of `response` (for raw residuals), `pearson`, `deviance` or `working`. Fox (p. 458) calls working residuals "logit residuals".

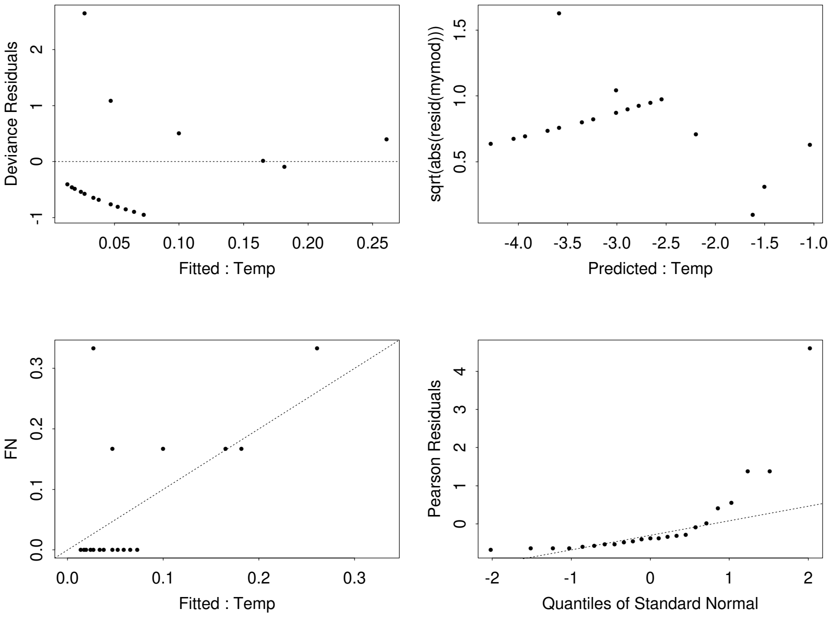

Figure 3: Splus's standard logistic regression diagnostic plots for the Challenger data.

Leverage and "delete-one" diagnostics. An analogue of the hat matrix,

$$H = V^{1/2}X(X^TVX)^{-1}V^{1/2}$$

is available (for example, $(I-H)\underline{Z} = \underline{Z}$), and the diagonal elements of it can be interpreted the same way as h_{ii} in linear regression, as measures of leverage of each observation.

Recall that in OLS we could obtain studentized residuals by fitting the usual model, but with an additional indicator variable for observation i. The same thing can be done in logistic regression: we fit the model

$$logit(\underline{p}) = X\underline{\beta} + \gamma D_{\text{obs } i}$$

and then the same model without γ. The log-likelihood ratio statistic (or deviance statistic)

$$G^2_{(-i)} = -2\log(L_\gamma/L_{\gamma=0})$$

is asymptotically χ^2_1 and its square root $G_{(-i)}$ (with sign chosen to agree with E_i is analogous to a studentized residual.

It can be expensive to refit the model N times to obtain each studentized residual in this way, but Pregibon (1981) shows that

$$G^2_{(-i)} \approx G^2_i + \frac{Z_i^2 h_{ii}}{1-h_{ii}};$$

substituting G_i for Z_i and taking the square root we obtain approximate studentized residuals

$$G^*_i = \frac{G_i}{\sqrt{1-h_{ii}}}.$$

The G_i^* are approximately normally distributed when the model holds (hence `qqnorm` plots etc. are useful for checking the model, checking for outliers, erc; though discreteness of the response can still make interpretation difficult).

Similarly $DFBETA$ plots and Cook's distances can be obtained exactly by refitting the model N times and looking at $\underline{\hat{\beta}} - \underline{\hat{\beta}}_{(-i)}$. Approximations are again provided by Pregibon (1981): For $DFBETA$,

$$\underline{\hat{\beta}} - \underline{\hat{\beta}}_{(-i)} \approx (X^T V X)^{-1} \underline{x}_i \frac{E_i}{1 - h_{ii}}$$

where, as usual, $\underline{x}_i$ is the i^{th} row of the X matrix, written as a column vector; and for Cook's D,

$$D_i \approx \frac{E_i^2}{n_i p_i (1 - p_i)} \cdot \frac{h_{ii}}{(1 - h_{ii})^2} = Z_i^2 \cdot \frac{h_{ii}}{(1 - h_{ii})^2}$$

These are all very much like the analogous formulas for OLS regression (cf. e.g. p. 290 of Fox); some more details on their derivation are given on pp. 457–462 of Fox; see also Pregibon (1981) and Landwehr et al. (1980, *Proceedings of the ASA Business and Economics Section,* 15–20).

Figure 4 illustrates some of these diagnostics.

```
> X _ model.matrix(mymod)
> V _ diag(mymod$weights)
> W _ diag(sqrt(mymod$weights))
> H _ W%*%X%*%solve(t(X)%*%V%*%X)%*%t(X)%*%W
> hats _ diag(H)
> studres _ residuals(mymod,type="deviance")/sqrt(1-hats)
> CooksD _ residuals(mymod,type="pearson")^2 * hats/(1-hats)^2
> source("../week10/qqenv.splus")
> par(mfrow=c(2,2))
> plot(predict(mymod),studres)
> abline(h=0) ; identify(predict(mymod),studres)
> qqenv(studres)
> plot(hats)
> plot(CooksD)
```

2.7 Fitted values and extrapolations

The fitted values (on the probability scale) are obtained via,

```
> fitted(mymod)
         1         2         3          4          5          6          7
 0.2607849 0.1817866 0.1652202 0.09994062 0.07278457 0.06535816 0.06535816
          8          9         10         11         12         13         14
 0.06535816 0.05864156 0.05257638 0.04710711 0.04710711 0.04710711 0.04710711
         15         16         17         18         19         20        21
 0.03775043 0.03376847 0.02698616 0.02698616 0.02411117 0.02411117 0.0192299
         22         23
 0.01716665 0.01367159
```

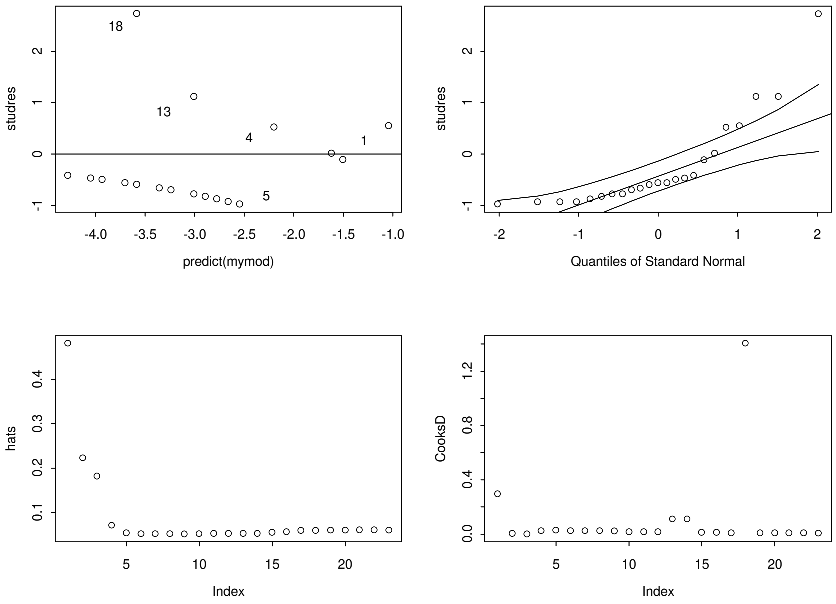

Figure 4: Some additional diagnostic plots for the Challenger data.

The predicted values (on the logit scale) are,

```
> predict(mymod)
        1         2          3          4          5          6          7
 -1.041893 -1.50429 -1.619889 -2.197885 -2.544682 -2.660281 -2.660281
        8         9         10         11         12         13         14
 -2.660281 -2.77588 -2.891479 -3.007079 -3.007079 -3.007079 -3.007079
       15        16         17         18         19         20         21
 -3.238277 -3.353876 -3.585074 -3.585074 -3.700673 -3.700673 -3.931872
       22        23
 -4.047471 -4.278669
> fitted(mymod) - unlogit(predict(mymod))
 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
```

(unlogit is the following simple function),

```
> unlogit <- function(x) { exp(x)/(1 + exp(x)) }
```

For communication purposes, it makes sense to work on the probability scale (and hence use `fitted`), but for all other purposes the predicted values are what we want. In particular, the `predict` function has an option that will give the Standard Errors of the predicted values (on the logit scale), viz.,

```
> predict(mymod,se=T)
$fit:
```

```
        1         2         3         4         5         6         7
 -1.041893 -1.50429 -1.619889 -2.197885 -2.544682 -2.660281 -2.660281
        8         9        10        11        12        13        14
 -2.660281 -2.77588 -2.891479 -3.007079 -3.007079 -3.007079 -3.007079
       15        16        17        18        19        20        21
 -3.238277 -3.353876 -3.585074 -3.585074 -3.700673 -3.700673 -3.931872
       22        23
 -4.047471 -4.278669

$se.fit:
        1         2         3         4         5         6         7
 0.7682786 0.5943294 0.556385 0.4298909 0.4298809 0.4441498 0.4441498
        8         9        10        11        12        13        14
 0.4441498 0.4647341 0.4908397 0.5216385 0.5216385 0.5216385 0.5216385
       15        16        17        18        19        20        21
 0.5942932 0.6348851 0.7222049 0.7222049 0.7682357 0.7682357 0.8637766
       22       23
 0.9129226 1.013299

$residual.scale:
[1] 1.189907

$df:
[1] 21
```

So, a 95% confidence interval for the first predicted value on the logit scale would be

$$-1.041893 \pm 2 \times 0.7682786$$

Transferred to the probability scale, this is an interval $(0.070, 0.621)$, with a center at 0.260.

But what about predictions nearer the take-off temperature?

```
 predict(mymod, data.frame(Temp=c(30,31,32,33)), se=T)
$fit:
        1        2        3       4
 1.616887 1.501288 1.385689 1.27009

$se.fit:
        1        2        3       4
 1.972255 1.917739 1.863302 1.80895
```

(That was pretty sneaky, but the `predict` function requires a data frame of predictor values).

The bottom line is that at 31°F our model extrapolates to $p = 0.83$, with a 95% confidence interval of $(0.089, 0.996)$, which is VERY wide. Even with this wide interval, the point estimate is something to worry about.

2.8 The Zero-One version of the model

Very quickly, lets look at the model when we just look at a failure or not.

```
> Bn <- cbind(sign(Fail), 1 - sign(Fail))
> Bn
   [,1] [,2]
 1    1    0
 2    1    0
 3    1    0
 4    1    0
 5    0    1
 6    0    1
 7    0    1
 8    0    1
 9    0    1
10    0    1
11    0    1
12    0    1
13    1    0
14    1    0
15    0    1
16    0    1
17    0    1
18    1    0
19    0    1
20    0    1
21    0    1
22    0    1
23    0    1
```

then

```
> newmod <- glm(Bn ~ Temp, family=binomial)
> summary(newmod)

Call: glm(formula = Bn ~ Temp, family = binomial)
Deviance Residuals:
       Min        1Q     Median         3Q       Max
 -1.061112 -0.76128 -0.3782822 0.4523828 2.217506

Coefficients:
                 Value Std. Error   t value
(Intercept)  15.0422911  7.3366528  2.050293
       Temp  -0.2321537  0.1076141 -2.157279

(Dispersion Parameter for Binomial family taken to be 1 )
```

```
    Null Deviance: 28.26715 on 22 degrees of freedom

Residual Deviance: 20.31519 on 21 degrees of freedom

Number of Fisher Scoring Iterations: 4

Correlation of Coefficients:
     (Intercept)
Temp -0.99716
> anova(newmod)
Analysis of Deviance Table

Binomial model

Response: Bn

Terms added sequentially (first to last)
     Df Deviance Resid. Df Resid. Dev
NULL                    22   28.26715
Temp  1  7.95196        21   20.31519

> plot(newmod)

> X _ model.matrix(newmod)
> V _ diag(newmod$weights)
> W _ diag(sqrt(newmod$weights))
> H _ W%*%X%*%solve(t(X)%*%V%*%X)%*%t(X)%*%W
> hats _ diag(H)
> studres _ residuals(newmod,type="deviance")/sqrt(1-hats)
> CooksD _ residuals(newmod,type="pearson")^2 * hats/(1-hats)^2
> par(mfrow=c(2,2))
> plot(predict(newmod),studres)
> abline(h=0) ; identify(predict(newmod),studres)
> qqenv(studres)
> plot(hats)
> plot(CooksD)
```

Figure 5 gives the diagnostic plots.

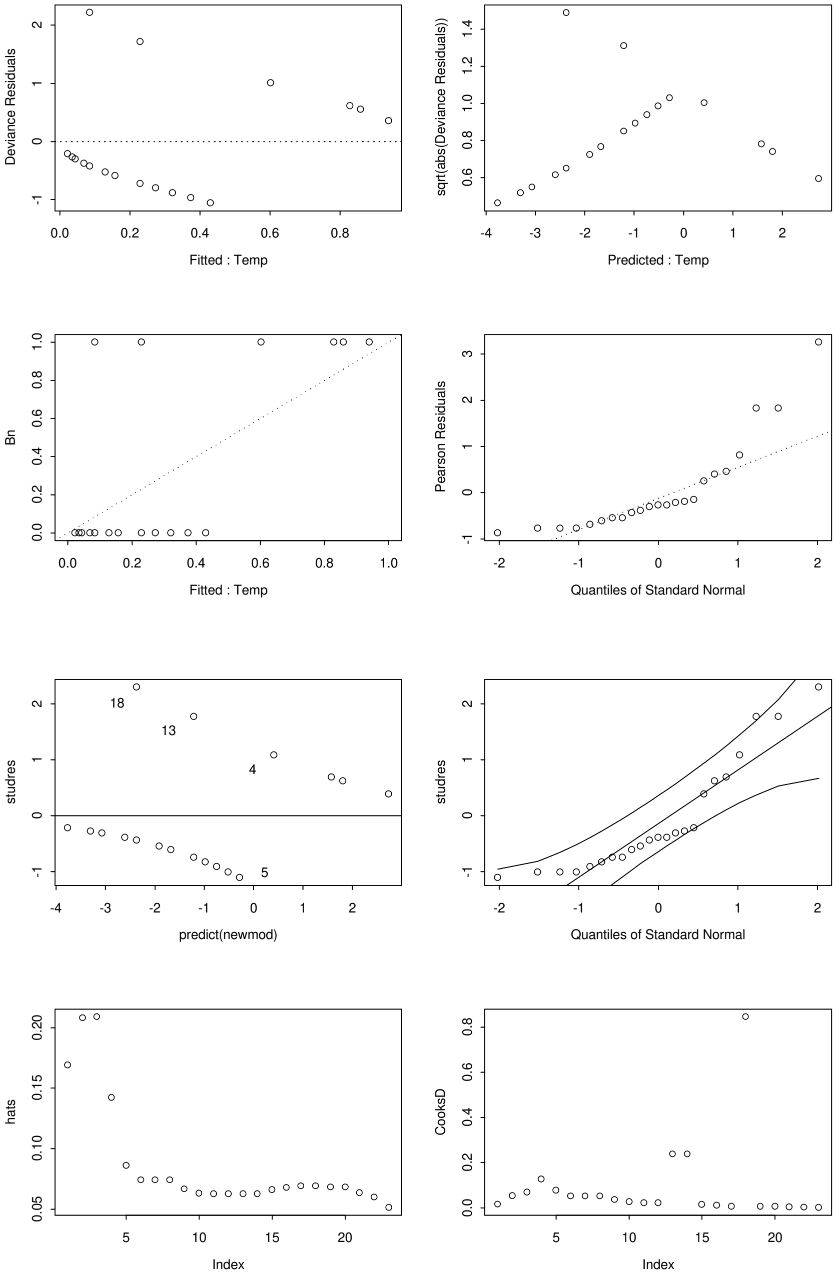

Figure 5: Standard and additional diagnostic plots for the 0/1 model for the Challenger data.