Notes for Tuesday March 17, 1998

Logistic Regression

The Challenger Data

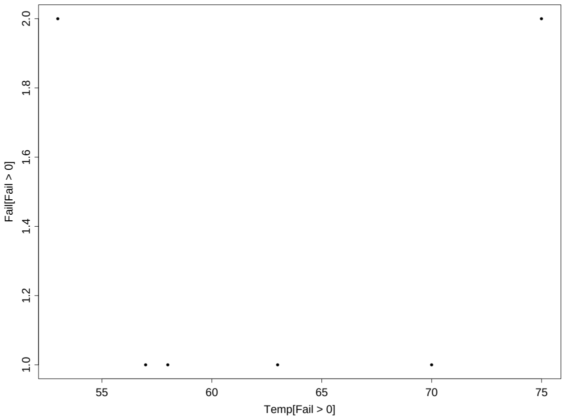
You are probably all familiar with the Challenger Space Shuttle accident of January 20, 1986. I will leave out all the interesting substantive details about the accident and focus instead on the data about O-rings that was available at the time of the accident.

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. There was data about the O-rings on 23 previous flights of space shuttles. 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. They produced a plot that looked something like this.

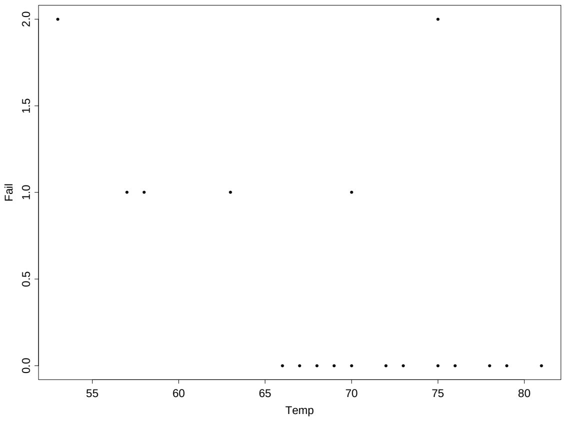
```
402 > chal <- read.table("chal.dat", head=T)
402 > attach(chal, pos=1)
402 > ls()
[1] ".Last.value" "Fail"        "Temp"
402 > n <- rep(6, length(Fail))

402 > 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 produced a plot of the full data, which follows, they might have behaved differently.

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



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.

How to look at these data

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).

We have in the past looked at models for count data that were of the form

$$\log m = \text{some combination of factor effects}$$

Although these loglinear models did not require discrete (categorical) predictors, the only examples we saw were of that form. The current data can also be modelled using the standard loglinear model, but an equivalent and more natural approach is to set up the model as a logistic regression.

In this case, allow $p(t)$ to be 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.

The Logistic model

Clearly, we can not get away with modelling

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

unless we are very careful about the arbitrary function. One would quickly find oneself in the position where the models produced negative probabilities, or probabilities greater than 1.

One might think of the model (based on our loglinear experience)

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

but it suffers from the same problems.

Lets go back to first principles and think about the data we are trying to model.

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

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

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

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

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

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

is called the logistic transformation. If

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

then the inverse function is

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

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

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,

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

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.

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.

The model

The S-Plus logistic regression command needs to know both the number of successes and the number of failures, so we need to start out by creating such a variable,

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

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

```
402 > mymod <- glm( FN ~ Temp, family=binomial)
402 > 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. Now look at the results of the fit,

```
402 > 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
402 > 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
```

The model fits well (although, just like R^2 is regression, one should be cautious about this), and the temperature coefficient is certainly significant.

At this point we need to

- Get predictions for the low-temperature extrapolations, and
- Evaluate the fit of the model

Extrapolating Lets work on that, in that order (which is really back-to-front, but useful for teaching purposes).

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

```
402 > 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
```

The predicted values (on the logit scale) are,

```
402 > 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
402 > 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),

```
402 > 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.,

```
402 > 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:
```

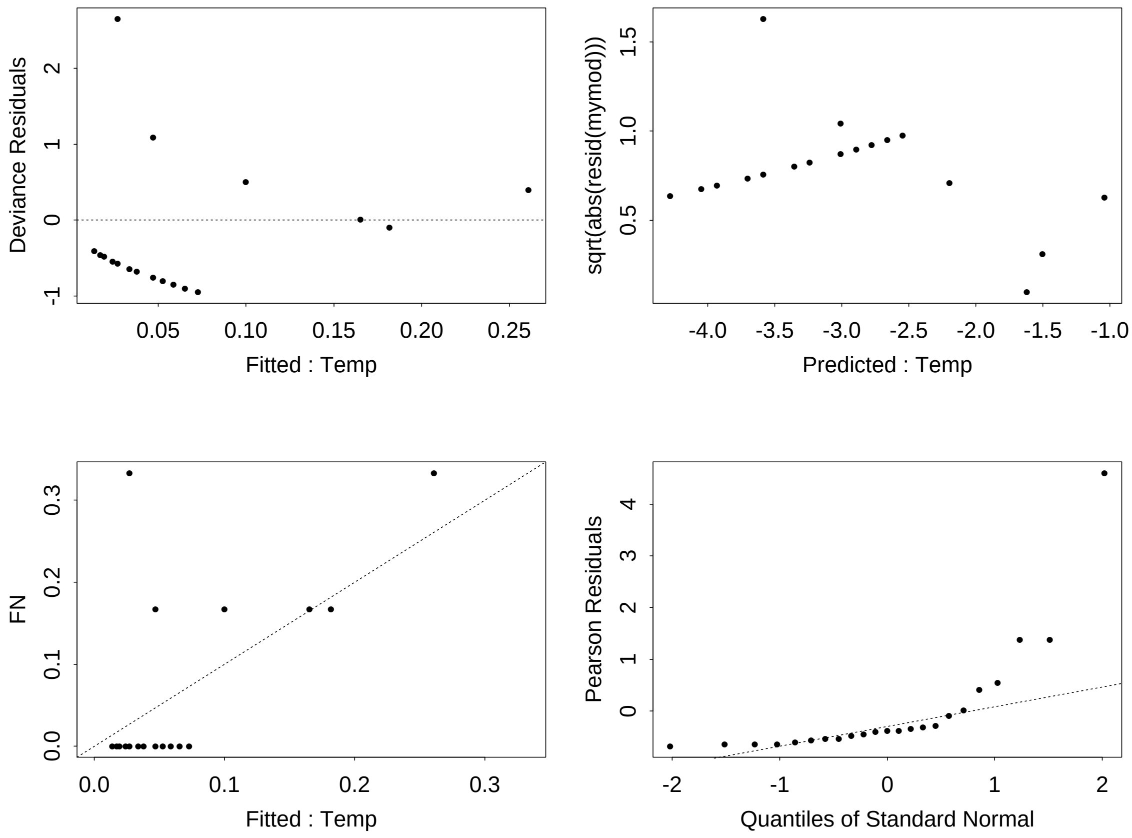
```
        1        2        3       4
 1.972255 1.917739 1.863302 1.80895
```

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

Evaluating the Model

```
402 > plot(mymod)
```



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 don't be fooled. The places where there are big residuals are for temperatures where there were both no failures, and failures. I tried a number of additional functions of temperature, and nothing much worked.

The Zero-One version of the model

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

```
402 > Bn <- cbind(sign(Fail), 1 - sign(Fail))
402 > 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

```
402 > mymod <- glm(Bn ~ Temp, family=binomial)
```

```
402 > summary(mymod)

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
402 > anova(mymod)
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
```

So far things look very similar, including the residual plot (believe me).

Budworms

This data set is analyzed by Venables and Ripley also, pp. 230 ff.

The data is the result of an experiment on the toxicity of the tobacco budworm to doses of particular drug, that the moths are beginning to show resistance to. Batches of 20 moths of each sex were exposed for 3 days to the chemical and the number of dead (or disabled) moths in each group were recorded. The data is,

```
Dose Response Gender
1       1       M
2       4       M
4       9       M
8       13      M
16      18      M
32      20      M
1       0       F
2       2       F
4       6       F
8       10      F
16      12      F
32      16      F
```

The doses were in micro-grams, and the dose level when up by powers of two. It is a judgement call as to whether you will try to model with dose as the response or $\log_2$(dose). We will try both.

```
402 > bud <- read.table("budworm.dat", head=T)
402 > bud
   Dose Response Gender
 1    1        1      M
 2    2        4      M
 3    4        9      M
 4    8       13      M
 5   16       18      M
 6   32       20      M
```

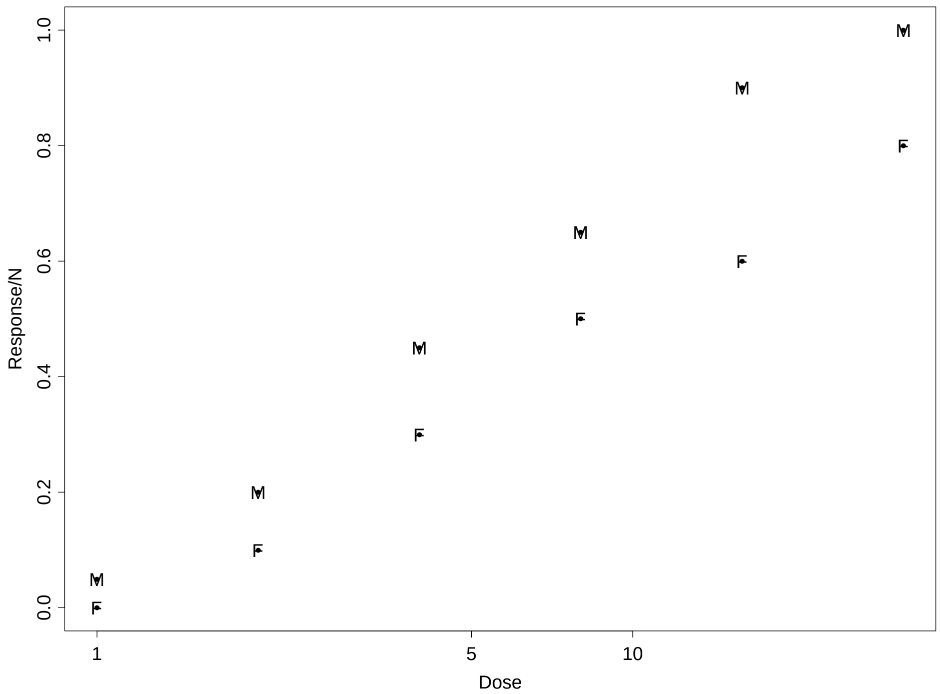
```
 7    1        0      F
 8    2        2      F
 9    4        6      F
10    8       10      F
11   16       12      F
12   32       16      F
402 > attach(bud,pos=1)
402 > N <- 20
```

First, lets plot the data, and see if anything obvious shows up.

```
402 > plot( Dose, Response/N)
402 > points( Dose[Gender=="F"], (Response/N)[Gender=="F"], pch="F")
402 > points( Dose[Gender=="M"], (Response/N)[Gender=="M"], pch="M")
```

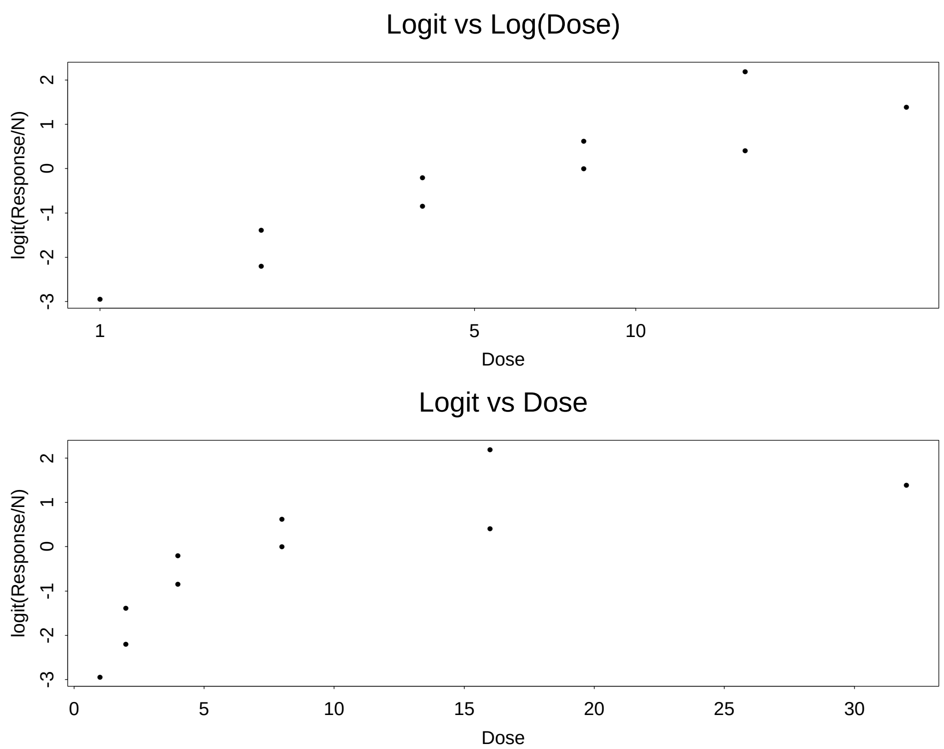
The same plot on the log(Dose) scale, is

```
402 > plot( Dose, Response/N, log="x")
402 > points( Dose[Gender=="F"], (Response/N)[Gender=="F"], pch="F")
402 > points( Dose[Gender=="M"], (Response/N)[Gender=="M"], pch="M")
```



Given that we have more than 0-1 data, it makes a lot of sense to at least try to plot the (observed) logits,

```
402 > par(mfrow=c(2,1))
402 > plot( Dose, logit(Response/N), log="x")
402 > title("Logit vs Log(Dose)")
402 > plot( Dose, logit(Response/N))
402 > title("Logit vs Dose")
```



This plot makes it quite clear, that the log scale has a much better chance of working.

So, proceed to fit the model. We surely want a model with different slopes for the two sexes

```
402 > ldose <- log(Dose)
402 > SF <- cbind(Response, N - Response)
402 > mymod <- glm(SF ~ Gender * ldose, family=binomial)
402 > summary(mymod)
```

....

```
Coefficients:
                 Value Std. Error    t value
 (Intercept) -2.9935414  0.5525295 -5.4178852
      Gender  0.1749865  0.7781556  0.2248733
       ldose  1.3071341  0.2410133  5.4234939
Gender:ldose  0.5091459  0.3894474  1.3073547


    Null Deviance: 124.8756 on 11 degrees of freedom

Residual Deviance: 4.993727 on 8 degrees of freedom

402 > anova(mymod,test="Chisq")
Analysis of Deviance Table

Binomial model

Response: SF

Terms added sequentially (first to last)
             Df Deviance Resid. Df Resid. Dev   Pr(Chi)
        NULL                    11   124.8756
      Gender  1   6.0770        10   118.7986 0.0136955
       ldose  1 112.0415         9     6.7571 0.0000000
Gender:ldose  1   1.7633         8     4.9937 0.1842088
402 >
```

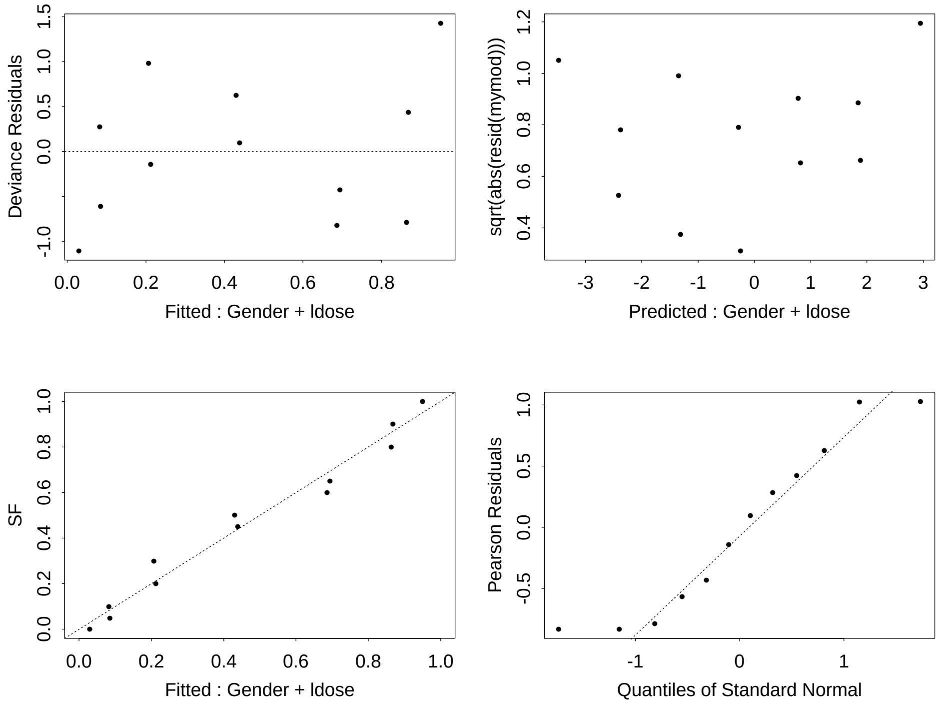
Now, what is going on here? `Gender` seems to have no effect (based on the t-value), but the deviance is quite large? Even more, our plots seem to imply what looks like a gender effect. The best way to find out is to distrust the parameter stuff (it can get easily confused), and try fitting a model without the interaction. This makes sense anyway, because there is little/no evidence of an interaction.

```
402 > anova(mymod <- glm(SF ~ Gender + ldose, family=binomial))
```

```
Terms added sequentially (first to last)
       Df Deviance Resid. Df Resid. Dev
  NULL                    11   124.8756
Gender  1   6.0770        10   118.7986
 ldose  1 112.0415         9     6.7571
402 > summary(mymod)
Coefficients:
                Value Std. Error   t value
(Intercept) -3.473154  0.4682939 -7.416612
     Gender  1.100743  0.3557226  3.094385
      ldose  1.535336  0.1890119  8.122959

    Null Deviance: 124.8756 on 11 degrees of freedom

Residual Deviance: 6.757064 on 9 degrees of freedom
```

That is a little more like it, and seems to be a VERY good fitting model. We should at least look at some diagnostics, and some other plots, but unless they show something very odd, we are done.

```
402 > plot(mymod)
```



That looks very good, we should finish by plotting the data and fitted curves on the same graph, and that is left as an exercise for you (or look it up in the book). The book also goes on to talk about LD50, and how to estimate it.