Logistic regression and Dose Response Data

Many thanks to Thaddeus Tarpey at Wright University Check out his cite for this and more http://www.wright.edu/~thaddeus.tarpey/

These data are a reproduction of data from C.I. Bliss (1935). The calculation of the dosage-mortality curve. *Annals of Applied Biology*, vol 22, Issue 1, 134-167.

Data

Beetles were exposed to carbon disulphide at varying concentrations for 5 hours.

- dose = mf/L concentration of CS_2
- nexp = number of beetles exposed
- ndied = number of beetles killed
- prop = proportion of dead to exposed beetles

```
exp.dat = matrix(c(49.1,53,56.9,60.8,64.8,68.7,72.6,76.5,
                   59,60,62,56,63,59,62,60,
                   6,13,18,28,52,53,61,60,
                   .102,.217,.29,.5,.825,.898,.984,1),ncol=4)
colnames(exp.dat) = c('dose','nexp','ndied','prop')
exp.dat = as.data.frame(exp.dat)
# compute how many lived
exp.dat$nalive = exp.dat$nexp - exp.dat$ndied
```

Logistic Model

Run a logistic regression of the proportion of dead to living beetles as a function of the dose of CS_2 gas.

```
summary(glm(cbind(ndied,nalive) ~ dose,
            family=binomial,
            data=exp.dat)->exp.glm)
```

```
Call:
glm(formula = cbind(ndied, nalive) ~ dose, family = binomial,
    data = exp.dat)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.2746  -0.4668   0.7688   0.9544   1.2990

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -14.82300    1.28959  -11.49   <2e-16 ***
dose          0.24942    0.02139   11.66   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 284.2024  on 7  degrees of freedom
Residual deviance:   7.3849  on 6  degrees of freedom
AIC: 37.583

Number of Fisher Scoring iterations: 4
```

we may be interested in finding the concentration of CS_2 gas that is lethal 50% of the time, the LD_{50} below is the function for this computation.

```
dose4prob = function(b0,b1,prob){
  d = (-b0+log(-prob/(prob-1)))/b1
  return(d)
}
dose4prob(coef(exp.glm)[[1]],coef(exp.glm)[[2]],.5)
```

```
[1] 59.43092
```

Note that if we have a function with multiple predictors we

can solve for each variable using something similar. For example if

$$y \sim b_0 + b_1(x_1) + b_2(x_2) + b_3(x_3)$$

is the model. Then to find a specific value for one of the predictors (x_1, x_2, x_3) that corresponds to a desired probability (y).

- $x_1 = (-b_0 - b_2 - b_3 + log\left(\frac{-y}{(y-1)}\right))/b1$
- $x_2 = (-b_0 - b_1 - b_3 + log\left(\frac{-y}{(y-1)}\right))/b2$
- $x_3 = (-b_0 - b_1 - b_2 + log\left(\frac{-y}{(y-1)}\right))/b3$

```
# what is the range of doses
range(exp.dat$dose)
```

```
[1] 49.1 76.5
```

```
# let's create our own range to predict with
drange = seq(30,90,length=100)
# predict probability of beetle death based on the model
exp.pred = predict(exp.glm, newdata=data.frame(dose=drange))
# Now we plot
# first the predicted values
plot(drange,
     exp(exp.pred)/(1+exp(exp.pred)),
     type='l',
     xlab='dose',
     ylab='probability')
```

```r
# add in the observed points
points(prop ~ dose, data = exp.dat, col='red', pch=19)
# compute our LD values
ld50 = dose4prob(coef(exp.glm)[[1]],coef(exp.glm)[[2]],.50)
ld25 = dose4prob(coef(exp.glm)[[1]],coef(exp.glm)[[2]],.25)
# let's add these to our graph for visualization purposes
abline(h=.5,lty='dashed',col='red')
abline(h=.25,lty='dotted',col='blue')
legend('topleft',
       c(expression(LD[50]),
         expression(LD[25])),
       col=c('red','blue'),lty=c('dashed','dotted'),
       inset=.01)
text(ld50,.52,sprintf("%.3f",ld50),pos=c(2),col='red')
text(ld25,.27,sprintf("%.3f",ld25),pos=c(2),col='blue')
```