

Additional Exercise 5

Jonas Moss

September 20, 2018

First we load the data:

```
beetle = read.table(file = "http://www.stat.ufl.edu/~aa/glm/data/Beetles2.dat",  
                    header = TRUE)
```

A quadratic logit model

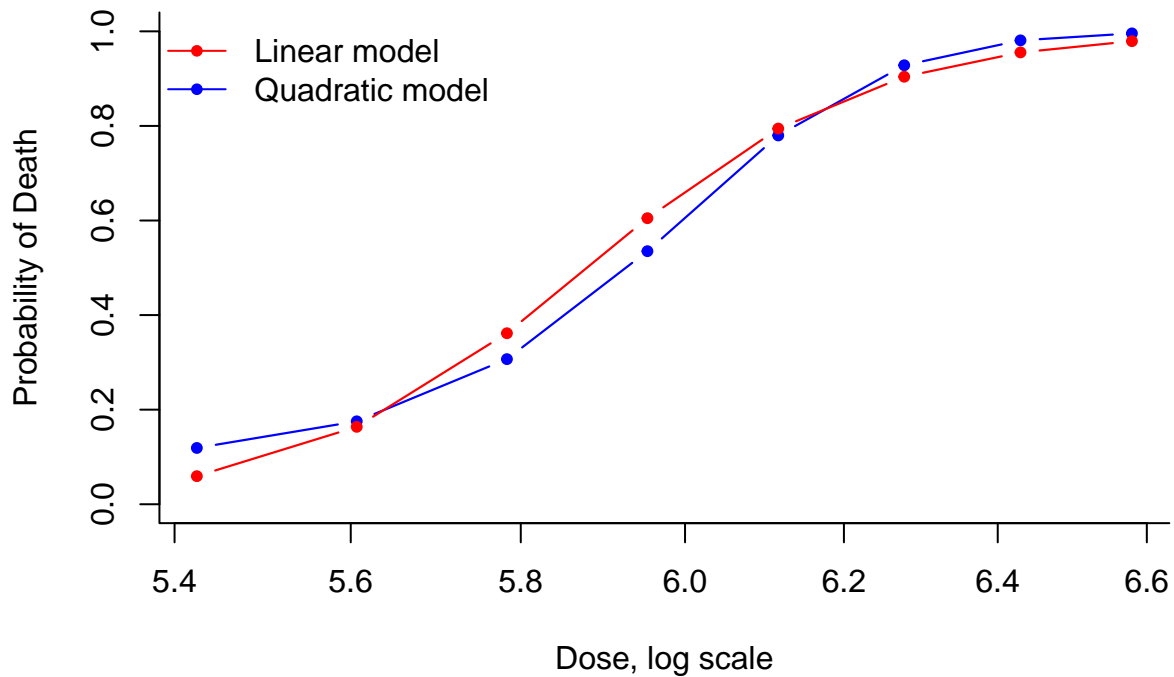
Linear and quadratic logistic regressions can be fitted by doing this.

```
glm(formula = cbind(dead, n - dead) ~ logdose,  
    family = binomial(link = logit),  
    data = beetle) ->  
linear_logit  
  
glm(formula = cbind(dead, n - dead) ~ logdose + I(logdose^2),  
    family = binomial(link = logit),  
    data = beetle) ->  
quadratic_logit
```

These can be plotted with the predict function:

```
plot(x = exp(beetle$logdose),  
     y = exp(predict(quadratic_logit))/(1 + exp(predict(quadratic_logit))),  
     type = "b",  
     col = "blue",  
     bty = "n",  
     ylim = c(0, 1),  
     pch = c(20),  
     lwd = 1.1,  
     xlab = "Dose, log scale",  
     ylab = "Probability of Death",  
     main = "Two logistic models",  
     log = "x")  
  
lines(x = exp(beetle$logdose),  
      y = exp(predict(linear_logit))/(1 + exp(predict(linear_logit))),  
      type = "b",  
      lwd = 1.1,  
      pch = c(20),  
      col = "red")  
  
legend(x = "topleft",  
       legend = c("Linear model", "Quadratic model"),  
       lty = c(1, 1),  
       pch = c(20, 20),  
       col = c("red", "blue"),  
       bty = "n")
```

Two logistic models



We can make use of the `summary` function to assess significance:

```
summary(quadratic_logit, corr = TRUE)
```

```
##
## Call:
## glm(formula = cbind(dead, n - dead) ~ logdose + I(logdose^2),
##      family = binomial(link = logit), data = beetle)
##
## Deviance Residuals:
##      1       2       3       4       5       6       7       8
## -0.4208  0.8177 -0.2843 -0.5269  0.8966 -0.8433  0.1716  0.7229
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   426.39     181.22   2.353   0.0186 *
## logdose      -515.30     205.15  -2.512   0.0120 *
## I(logdose^2)   154.92      58.04   2.669   0.0076 **
## ---
## 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:   3.2711  on 5  degrees of freedom
## AIC: 35.469
##
```

```
## Number of Fisher Scoring iterations: 4
##
## Correlation of Coefficients:
##           (Intercept) logdose
## logdose      -1.00
## I(logdose^2)  1.00      -1.00
```

Hence the correlations between the coefficients are almost as strong as they can get. This happens because the correlation between `logdose` and `logdose^2` is very high:

```
cor(beetle$logdose, beetle$logdose^2)
```

```
## [1] 0.9998808
```

This correlation can be partly removed by centering `logdose`:

```
newdose = (beetle$logdose - mean(beetle$logdose))
cor(newdose, newdose^2)
```

```
## [1] -0.180101
```

If we use `newdose`, the correlation between the coefficients estimates nearly disappears:

```
glm(formula = cbind(dead, n - dead) ~ newdose + I(newdose^2),
    family = binomial(link = logit),
    data = beetle) ->
quadratic_logit
summary(quadratic_logit, corr = TRUE)
```

```
##
## Call:
## glm(formula = cbind(dead, n - dead) ~ newdose + I(newdose^2),
##     family = binomial(link = logit), data = beetle)
##
## Deviance Residuals:
##      1      2      3      4      5      6      7      8
## -0.4208  0.8177 -0.2843 -0.5269  0.8966 -0.8433  0.1716  0.7229
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.5055     0.1661   3.043  0.00234 **
## newdose       40.3481     4.2669   9.456 < 2e-16 ***
## I(newdose^2) 154.9178    58.0375   2.669  0.00760 **
## ---
## 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:   3.2711  on 5  degrees of freedom
## AIC: 35.469
##
## Number of Fisher Scoring iterations: 4
##
## Correlation of Coefficients:
##           (Intercept) newdose
## newdose         0.00
## I(newdose^2) -0.43      0.71
```

A Probit Model

We fit both linear and quadratic probit models.

```
glm(formula = cbind(dead, n - dead) ~ logdose,
    family = binomial(link = probit),
    data = beetle) ->
linear_probit

glm(formula = cbind(dead, n - dead) ~ logdose + I(logdose^2),
    family = binomial(link = probit),
    data = beetle) ->
quadratic_probit
```

Plot

Here is a plot of all the models four models.

```
plot(x = beetle$logdose,
     y = exp(predict(quadratic_logit))/(1 + exp(predict(quadratic_logit))),
     type = "b",
     col = "blue",
     ylim = c(0, 1),
     pch = c(20),
     xlab = "Log-dose",
     ylab = "Probability of Death",
     main = "Four binary response models.")

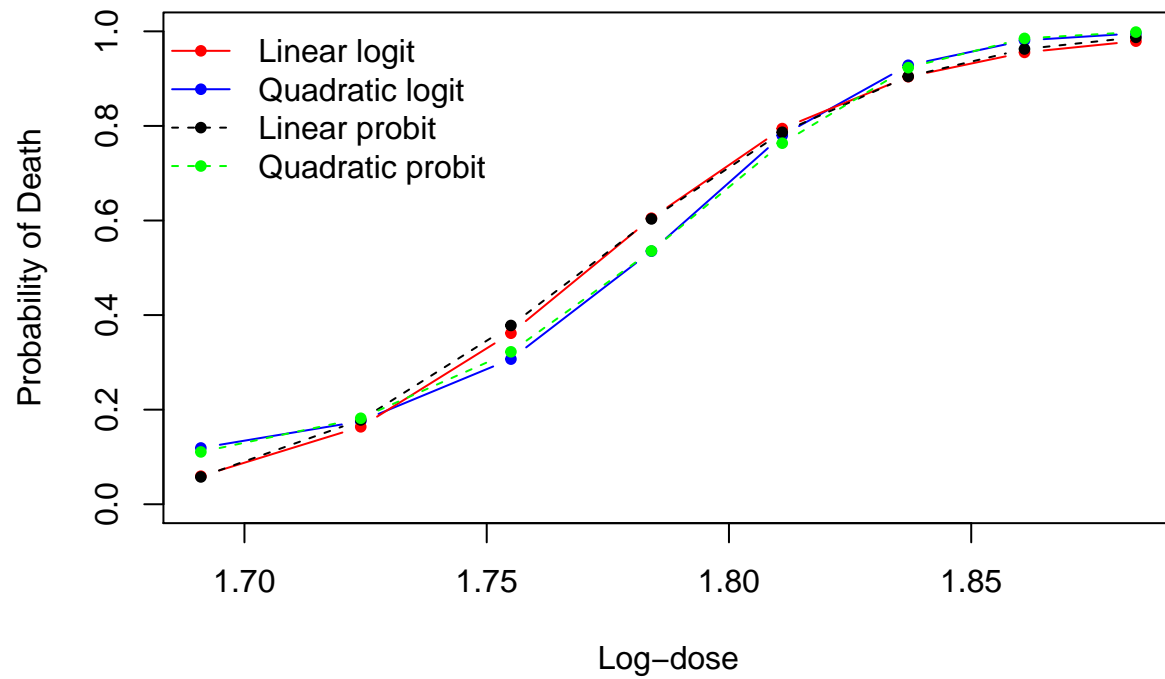
lines(x = beetle$logdose,,
      y = exp(predict(linear_logit))/(1 + exp(predict(linear_logit))),
      type = "b",
      pch = c(20),
      col = "red")

lines(x = beetle$logdose,,
      y = pnorm(predict(linear_probit)),
      type = "b",
      lty = 2,
      pch = c(20),
      col = "black")

lines(x = beetle$logdose,,
      y = pnorm(predict(quadratic_probit)),
      type = "b",
      lty = 2,
      pch = c(20),
      col = "green")

legend(x = "topleft",
       legend = c("Linear logit", "Quadratic logit",
                  "Linear probit", "Quadratic probit"),
       lty = c(1, 1, 2, 2),
       pch = c(20, 20, 20, 20),
       col = c("red", "blue", "black", "green"),
       bty = "n")
```

Four binary response models.



Log-likelihoods

The log likelihoods can be computed using the `logLik` function.

```
models = list(linear_probit, quadratic_probit, linear_logit, quadratic_logit)
unlist(lapply(models, logLik))
```

```
## [1] -18.09250 -14.69278 -18.65681 -14.73459
```