Notes for laboratory session 5

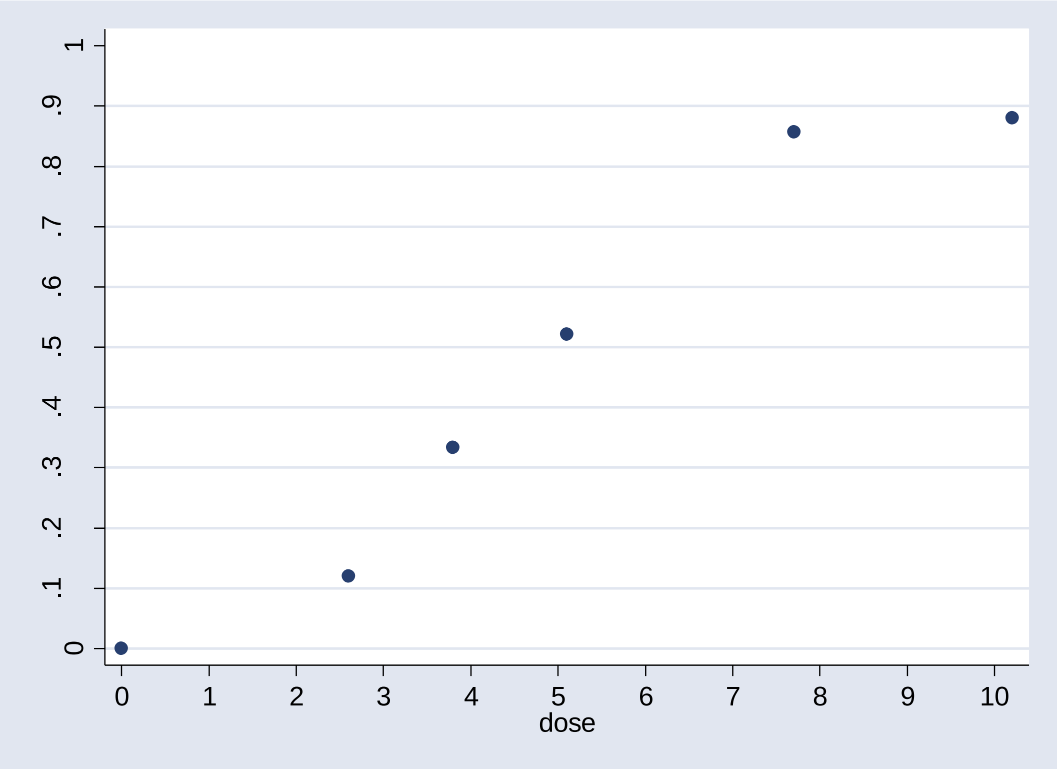
Logistic regression with grouped data

Open and list Finney's dataset

```
. use finney,clear

. list

          dose      noexp     deaths
  1.         0         49          0
  2.       2.6         50          6
  3.       3.8         48         16
  4.       5.1         46         24
  5.       7.7         49         42
  6.      10.2         50         44
```

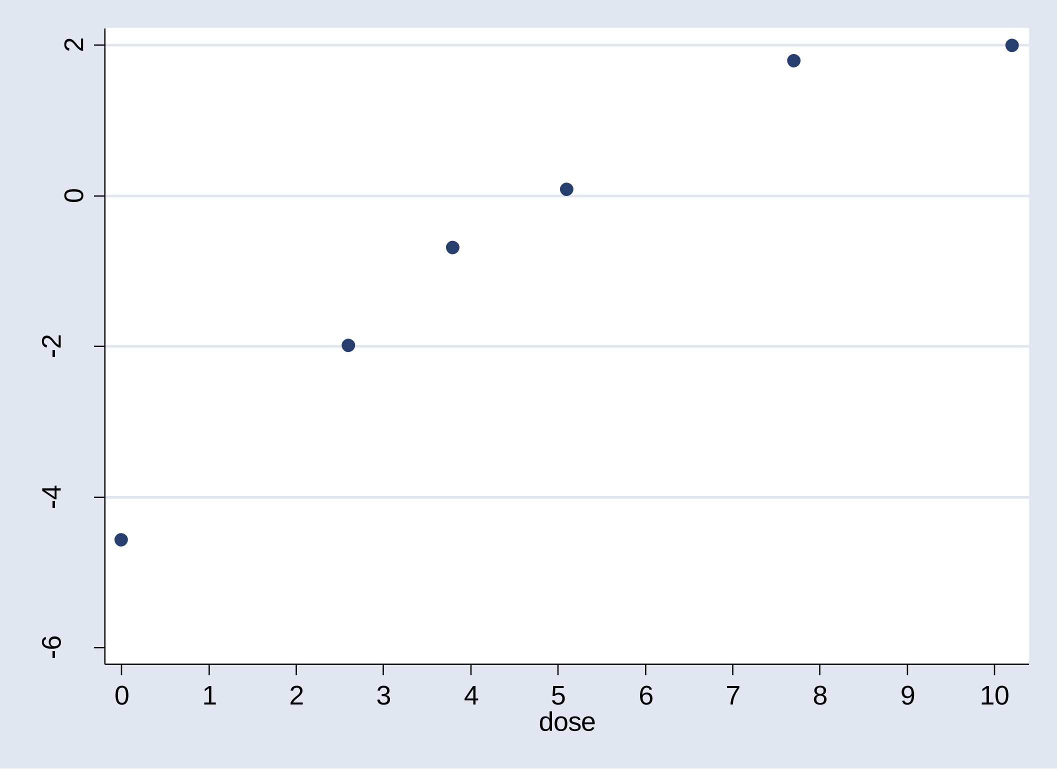
Now create a new variable for the rate of deaths in each dose class and then produce a graph showing the dose-death rate relation

```
. gen death_r= deaths/ noexp

. label var  death_r "Death rate"

. sc  death_r dose, xlab(0(1)10) ylab(0(0.1)1)
```



It appears to follow some sort of sigmoid curve. The logistic transformation (log(p/1-p)) may straighten the relationship. However the zero proportion will cause problems. We can duck this problem by pretending half an insect died at dose 0.

```
. replace death_r= (deaths+0.5)/ noexp if deaths==0

. gen logit_dr=log( death_r/(1- death_r))

. label var  logit_dr "Logit(Death rate)"

. sc   logit_dr dose,xlab(0(1)10) ylab()
```



This is getting straighter. Now fit a logistic regression model for the probability of death using as independent variable the dose

```
. blogit  deaths noexp dose

Logit estimates                                   Number of obs   =        292
                                                  LR chi2(1)      =     153.49
                                                  Prob > chi2     =     0.0000
Log likelihood = -124.31132                       Pseudo R2       =     0.3817

------------------------------------------------------------------------------
_outcome |      Coef.   Std. Err.       z     P>|z|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
    dose |   .6051256   .0678099      8.924   0.000       .4722207    .7380304
   _cons |  -3.225663   .3699052     -8.720   0.000      -3.950664   -2.500662
------------------------------------------------------------------------------
```

Notice that the binomial denominator (`noexp`) is required in the blogit command syntax. We can tell Stata to report Odds Ratios instead of betas in the blogit output using the `or` option.

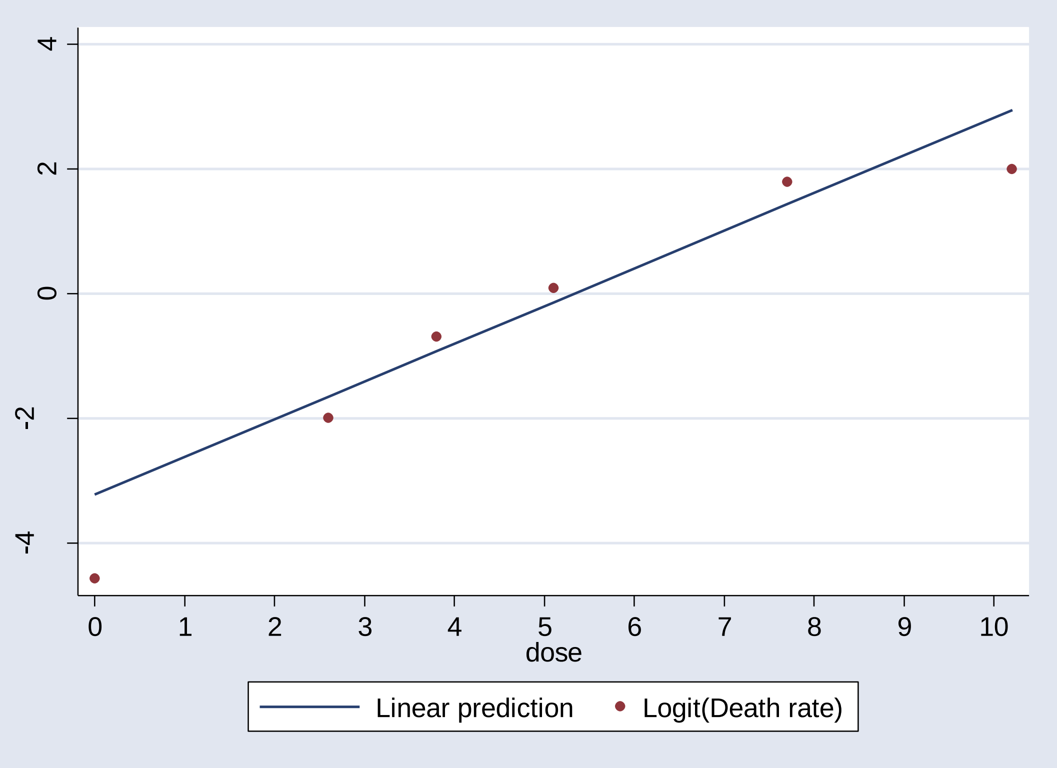
```
. blogit  deaths noexp dose,or

Logit estimates                                   Number of obs   =        292
                                                  LR chi2(1)      =     153.49
                                                  Prob > chi2     =     0.0000
Log likelihood = -124.31132                       Pseudo R2       =     0.3817

------------------------------------------------------------------------------
_outcome | Odds Ratio   Std. Err.       z     P>|z|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
    dose |   1.831482   .1241925      8.924   0.000       1.603551    2.091811
------------------------------------------------------------------------------

. est store M1
```

The `est (estimates)` command is used to save the Log likelihood of this model (we may want to use it for likelihood ratio tests later).

a) Calculate the Odds Ratio (and its 95% confidence interval) by hand and compare it with the second `blogit` output. Give a meaningful interpretation of this result. How can we calculate the Odds Ratio for 3 units increase in dose?

If we want to visualize this model's results we can obtain the linear predictions (log Odds) using the `predict` command followed by the `xb` option.

```
. predict pre1,xb

. sc  pre1 logit_dr dose,xlab(0(1)10) ylab() c(l .) ms(i o)
```



As we can see in the previous graph our data are showing some curvature thus a quadratic term of dose may be required to improve the fit

```
. gen dose2=dose^2

. blogit  deaths noexp dose dose2

Logit estimates                                   Number of obs   =        292
                                                  LR chi2(2)      =     162.67
                                                  Prob > chi2     =     0.0000
Log likelihood = -119.71879                       Pseudo R2       =     0.4045

------------------------------------------------------------------------------
_outcome |      Coef.   Std. Err.       z     P>|z|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
    dose |   1.513806    .359191      4.214   0.000       .8098049    2.217808
   dose2 |  -.0764208   .0277264     -2.756   0.006      -.1307635   -.0220782
   _cons |  -5.466344   1.023386     -5.341   0.000      -7.472143   -3.460545
------------------------------------------------------------------------------

. est store M2
```

As we can see the Log likelihood is raised by almost five units. We can check the significance of this result using the `lrtest` command

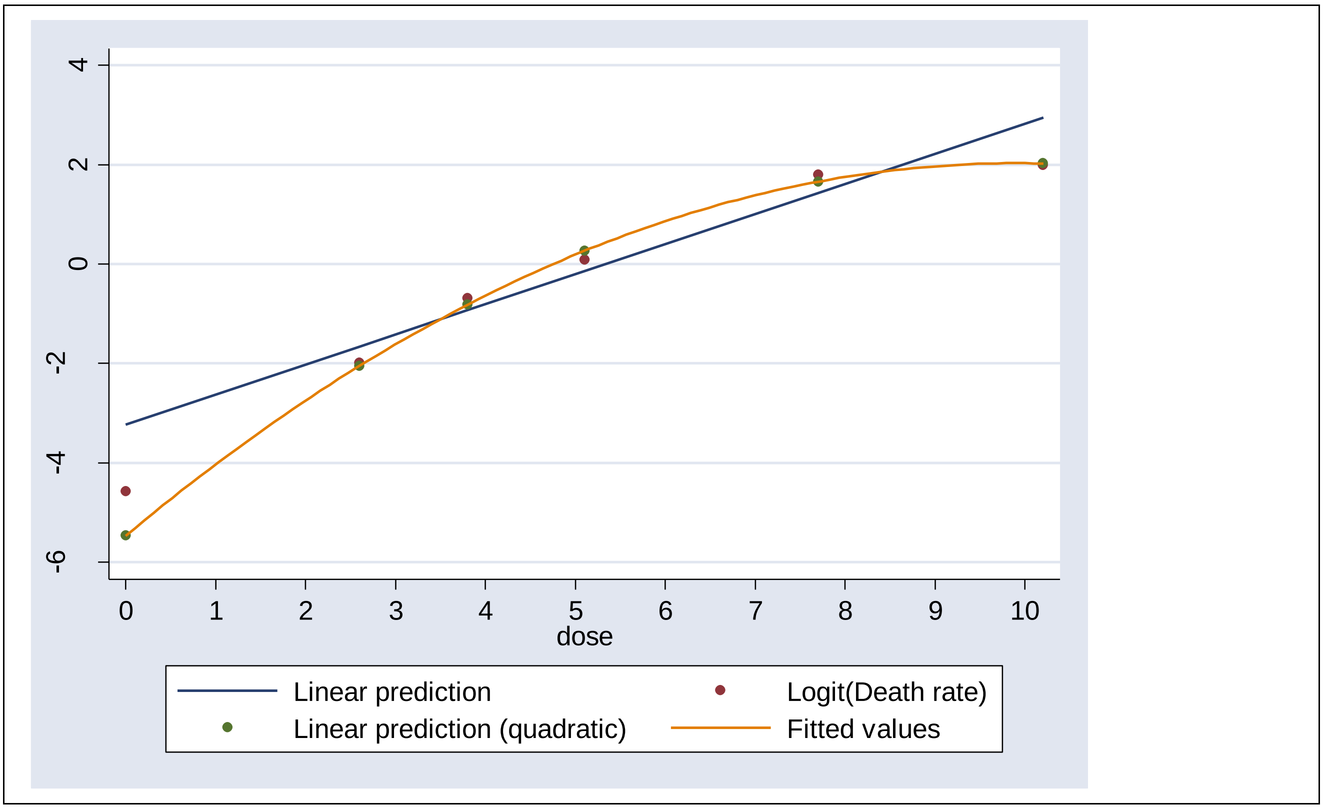
```
. lrtest M1 M2

likelihood-ratio test                                  LR chi2(1)  =      9.19
(Assumption: M1 nested in M2)                          Prob > chi2 =    0.0024
```

b) What do you think about the quadratic term? Verify the previous result using hand calculations

We can produce now a graph showing the raw data and fitted lines obtained by the “linear” and the “quadratic” model.

```
. predict pre2,xb

. label var pre2 "Linear prediction (quadratic)"

. sc  pre1  logit_dr pre2 dose,c(l . .) ms(i o o) || qfit pre2 dose ,xlab(0(1) 10) ylab()
```

We use `qfit` to connect the predictions of the quadratic model using a quadratic curve instead of straight lines. As we can see the fit now is clearly improved.

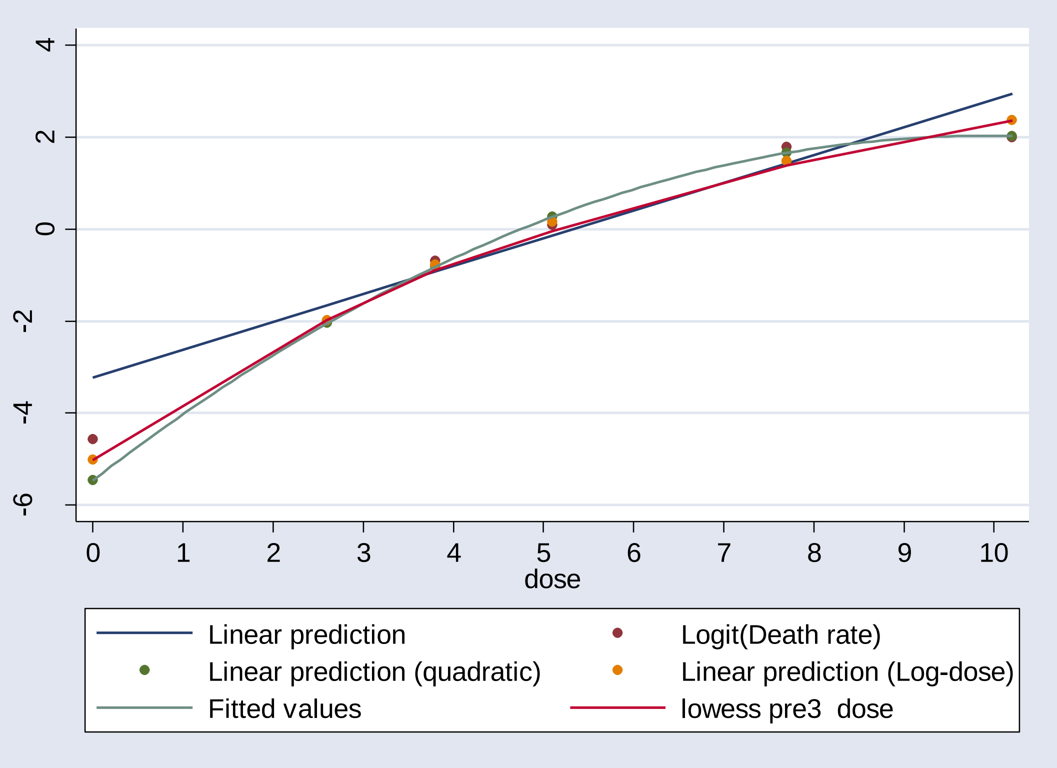


In many such relationships it is common for the response to increase linearly with the log of the dose. We can try now to use the log of the dose as independent variable instead of dose and dose2. We can also add one in dose when dose equals zero to avoid problems in the calculation of its logarithm

```
. gen logdose=log(dose+1*(dose==0))

. label var logdose "Log(dose)"

. blogit  deaths noexp logdose

Logit estimates                                   Number of obs   =        292
                                                  LR chi2(1)      =     161.63
                                                  Prob > chi2     =     0.0000
Log likelihood = -120.23794                       Pseudo R2       =     0.4020

------------------------------------------------------------------------------
_outcome |      Coef.   Std. Err.       z     P>|z|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
 logdose |   3.182329   .3713164      8.570   0.000       2.454562    3.910096
   _cons |  -5.021747   .6130903     -8.191   0.000      -6.223382   -3.820112
------------------------------------------------------------------------------
```

We cannot compare this model with the previous ones using likelihood ratio tests since this is not the “nested models” case, but is clear that the log likelihood is now higher compared with the one of the simple linear model (these two models have equal number of degrees of freedom) and slightly lower compared with the “quadratic” model’s one, while in the same time the “log-dose” model is more parsimonious. We can also check the fit of the model visually (the thicker curve corresponds to the “log-dose” model).

```
. predict pre3,xb

. label var pre3 "Linear prediction (Log-dose)"

. sc  pre1  logit_dr pre2 pre3 dose,c(l . . .) ms(i o o o) || qfit pre2 dose ||
lowess pre3 dose ,xlab(0(1) 10) ylab()
```



As we can see the fit of the “log-dose” model is quite acceptable thus given its parsimony compared with the “quadratic” model, this model may be the most preferable.

Estimating the LD50

The LD50 is the dose equivalent to a proportion responding (dying in our case) of 50%. On the log(Odds) scale used by logistic regression this is zero (log(0.5/(1-0.5))=0). This means for a fitted relationship y=a+bx we need x from 0=a+bx i.e. x=-a/b or –intercept/slope. CI’s for this ratio have to be obtained using Fieller’s theorem to obtain the variance of this ratio. The formula that we will use is as follows:

$$\operatorname{var}\left(\frac{T_1}{T_2}\right) \approx \left[\frac{E(T_1)}{E(T_2)}\right]^2 \left\{\frac{\operatorname{var}(T_1)}{[E(T_1)]^2} - \frac{2\operatorname{cov}(T_1,T_2)}{E(T_1)E(T_2)} + \frac{\operatorname{var}(T_2)}{[E(T_2)]^2}\right\}$$

It is first necessary to extract the betas and their variances and covariance and store them to local macros

```
. mat li e(b)

e(b)[1,2]
       logdose       _cons
y1   3.1823292  -5.0217469

. mat li e(V)

symmetric e(V)[2,2]
            logdose       _cons
logdose    .1378759
  _cons  -.21970321   .37587971

. mat coef=e(b)

. mat varcov=e(V)

. local a=coef[1,2]

. local b=coef[1,1]

. local var_a=varcov[2,2]

. local var_b=varcov[1,1]

. local cov_ab=varcov[1,2]
```

Now we can calculate the ratio, its variance and its 95% CI

```
. local v_aOVb=(`a'/`b')^2*( `var_a'/(`a')^2 - 2*`cov_ab'/(`a'*`b') +
`var_b'/(`b')^2)

. di `v_aOVb'
.00254946

.  di -`a'/`b'
1.5780099

.  di -`a'/`b'-1.96*sqrt(`v_aOVb') , -`a'/`b'+1.96*sqrt(`v_aOVb')
1.4790452 1.6769745
```

and on the actual scale

```
. di exp(-`a'/`b')
4.8453035

.  di exp(-`a'/`b'-1.96*sqrt(`v_aOVb')) , exp(-`a'/`b'+1.96*sqrt(`v_aOVb'))
4.3887535 5.3493471
```

LD50 (95% CI) = 4.85 (4.39 – 5.35)

Alternatively, one can use the Stata `nlcom` command (Nonlinear combinations of estimators) either in the log of dose scale (and then exponentiate)

```
. nlcom -_b[_cons]/_b[logdose]

       _nl_1:  -_b[_cons]/_b[logdose]

------------------------------------------------------------------------------
             |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       _nl_1 |    1.57801   .0504922    31.25   0.000     1.479047    1.676973
------------------------------------------------------------------------------

. di exp(1.57801), exp(1.479047), exp(1.676973)
4.8453041 4.3887612 5.349339
```

or directly in the dose scale

```
. nlcom exp(-_b[_cons]/_b[logdose])

       _nl_1:  exp(-_b[_cons]/_b[logdose])

------------------------------------------------------------------------------
             |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       _nl_1 |   4.845303   .2446499    19.81   0.000     4.365799    5.324808
------------------------------------------------------------------------------
```

Finally one can use a Monte-Carlo (simulation) approach by simulating from the bivariate normal distribution of the coefficients, generating the required expressions of either log(LD50) or LD50 and calculating the median (for point estimation) and 2.5 and 97.5 percentiles (for a 95% CI) :

```
. drawnorm b a ,means(coef) cov(varcov) n(100000) clear
. gen log_ld=-a/b
. centile log_ld,centile(2.5 50 97.5)

                                                          -- Binom. Interp. --
    Variable |       Obs  Percentile    Centile        [95% Conf. Interval]
-------------+-------------------------------------------------------------
      log_ld |   100,000        2.5    1.475372        1.474592    1.476307
             |                   50    1.577745        1.577387    1.578098
             |                 97.5    1.678726        1.677725    1.679655

. gen ld=exp(log_ld)

. centile ld,centile(2.5 50 97.5)

                                                          -- Binom. Interp. --
    Variable |       Obs  Percentile    Centile        [95% Conf. Interval]
-------------+-------------------------------------------------------------
          ld |   100,000        2.5    4.372661        4.369254    4.376754
             |                   50    4.844022        4.842286    4.845732
             |                 97.5    5.358723        5.353366    5.363706
```

Compare the results from all previous approaches.