

Modelling Rates

Mark Lunt

Arthritis Research UK Epidemiology Unit
University of Manchester



22/11/2011

Modelling Rates

- Can model prevalence (proportion) with logistic regression
- Cannot model incidence in this way
- Need to allow for time at risk (exposure)
- Exposure often measured in person-years
- Model a rate (incidents per unit time)

Assumptions

- There is a rate at which events occur
- This rate may depend on covariates
- Rate must be ≥ 0
- Expected number of events = rate \times exposure
- Events are independent
- Then the number of events observed will follow a Poisson distribution

Poisson Regression

- Negative numbers of events are meaningless
- Model $\log(\text{rate})$, so that rate can range from $0 \rightarrow \infty$

rate = r (events per unit exposure)

Count = C (Number of events)

Exposure = E

$C \sim \text{poisson}(rE)$

$E[C] = rE$

The Poisson Regression Model

$$\begin{aligned}\log(\hat{r}) &= \beta_0 + \beta_1 x_1 + \dots + \beta_p x_p \\ \hat{r} &= e^{\beta_0 + \beta_1 x_1 + \dots + \beta_p x_p}\end{aligned}$$

$$\begin{aligned}\text{Expected Number of Events, } E[C] &= E r \\ &= E e^{\beta_0 + \beta_1 x_1 + \dots + \beta_p x_p} \\ &= e^{\log(E) + \beta_0 + \beta_1 x_1 + \dots + \beta_p x_p}\end{aligned}$$

$$\text{Regression model: } \log(E[C]) = \log(E) + \beta_0 + \beta_1 x_1 + \dots + \beta_p x_p$$

Parameter Interpretation

- When x_i increases by 1, $\log(r)$ increases by β_i
- Therefore, r is multiplied by e^{β_i}
- As with logistic regression, coefficients are less interesting than their exponents
- e^{β} is the Incidence Rate Ratio

Poisson Regression in Stata

- Command `poisson` will do Poisson regression
- Enter the exposure with the option `exposure (varname)`
- Can also use `offset (lvarname)`, where `lvarname` is the log of the exposure
- To obtain Incidence Rate Ratios, use the option `irr`

Poisson Regression Example: Doctor's Study

Age	Smokers		Non-smokers	
	Deaths	Person-Years	Deaths	Person-Years
35–44	32	52,407	2	18,790
45–54	104	43,248	12	10,673
55–64	206	28,612	28	5,710
65–74	186	12,663	28	2,585
75–84	102	5,317	31	1,462

```
. xi: poisson deaths i.agecat i.smokes, exp(pyyears) irr
```

```
Poisson regression                                Number of obs =          10
                                                  LR chi2(5)           =       922.93
                                                  Prob > chi2          =        0.0000
Log likelihood = -33.600153                    Pseudo R2           =        0.9321
```

deaths	IRR	Std. Err.	z	P> z	[95% Conf. Interval]	
_Iagecat_2	4.410584	.8605197	7.61	0.000	3.009011	6.464997
_Iagecat_3	13.8392	2.542638	14.30	0.000	9.654328	19.83809
_Iagecat_4	28.51678	5.269878	18.13	0.000	19.85177	40.96395
_Iagecat_5	40.45121	7.775511	19.25	0.000	27.75326	58.95885
_ismokes_1	1.425519	.1530638	3.30	0.001	1.154984	1.759421
pyyears (exposure)						

Using `predict` after `poisson`

Options available:

<code>n</code>	(default)	expected number of events (rate \times duration of exposure)
<code>ir</code>		incidence rate
<code>xb</code>		linear predictor

Example: predict

```
predict pred_n
```

Age	Smokers		Non-smokers	
	Deaths	pred_n	Deaths	pred_n
35-44	32	27.2	2	6.8
45-54	104	98.9	12	17.1
55-64	206	205.3	28	28.7
65-74	186	187.2	28	26.8
75-84	102	111.5	31	21.5

Goodness of Fit

- Command `estat gof` compares observed and expected (from model) counts
- Can detect whether the Poisson model is reasonable
- If not could be due to
 - Systematic part of model poorly specified
 - Random variation not really Poisson
- Degrees of freedom for test = number of categories of observations - number of coefficients in model (including `_cons`)

Goodness of Fit Example

```
. estat gof
```

```
Goodness-of-fit chi2 = 12.13244  
Prob > chi2(4)      = 0.0164
```

Example: Improving fit of the model

```
. xi: poisson deaths i.agecat*i.smokes, exp(pyyears) irr
```

deaths	IRR	Std. Err.	z	P> z	[95% Conf. Interval]	
__Iagecat_2	10.5631	8.067701	3.09	0.002	2.364153	47.19623
__Iagecat_3	46.07004	33.71981	5.23	0.000	10.97496	193.3901
__Iagecat_4	101.764	74.48361	6.32	0.000	24.24256	427.1789
__Iagecat_5	199.2099	145.3356	7.26	0.000	47.67693	832.3648
__ismokes_1	5.736637	4.181256	2.40	0.017	1.374811	23.93711
__IageXsm~2_1	.3728337	.2945619	-1.25	0.212	.0792525	1.753951
__IageXsm~3_1	.2559409	.1935392	-1.80	0.072	.0581396	1.126697
__IageXsm~4_1	.2363859	.1788334	-1.91	0.057	.0536612	1.041316
__IageXsm~5_1	.1577109	.1194146	-2.44	0.015	.0357565	.6956154
pyyears (exposure)						

```
. testparm _IageX*
```

```
      chi2( 4) =    10.20  
Prob > chi2 =     0.0372
```

```
. lincom _Ismokes_1 + _IageXsmo_5_1, eform
```

```
-----  
deaths |      exp(b)   Std. Err.      z    P>|z|    [95% Conf. Interval]  
-----+-----  
      (1) |   .9047304   .1855513   -0.49   0.625   .6052658   1.35236  
-----
```

```
. estat gof
```

```
Goodness-of-fit chi2 = .0000694  
Prob > chi2(0)      = .
```

Constraints

- Can force parameters to be equal to each other or specified value
- Can be useful in reducing the number of parameters in a model
- Simplifies description of model
- Enables goodness of fit test
- **Syntax:** `constraint define n varname = expression`

Constraint Example

```
. constraint define 1 _IageXsmo_3_1 = _IageXsmo_4_1
. xi: poisson deaths i.agecat*i.smokes, exp(pyyears) irr constr(1)
-----
      deaths |          IRR   Std. Err.      z    P>|z|      [95% Conf. Interval]
-----+-----
  _Iagecat_2 |    10.5631   8.067701     3.09   0.002     2.364153     47.19623
  _Iagecat_3 |     47.671  34.37409     5.36   0.000    11.60056    195.8978
  _Iagecat_4 |    98.22765  70.85012     6.36   0.000    23.89324    403.8244
  _Iagecat_5 |   199.2099  145.3356     7.26   0.000    47.67693    832.3648
  _Ismokes_1 |   5.736637   4.181256     2.40   0.017     1.374811    23.93711
  _IageXsm~2_1 |  .3728337   .2945619    -1.25   0.212     .0792525     1.753951
  _IageXsm~3_1 |  .2461772   .182845    -1.89   0.059     .0574155     1.055521
  _IageXsm~4_1 |  .2461772   .182845    -1.89   0.059     .0574155     1.055521
  _IageXsm~5_1 |  .1577109   .1194146    -2.44   0.015     .0357565     .6956154
  pyyears | (exposure)
-----
```

```
. estat gof

      Goodness-of-fit chi2 =  .0774185
      Prob > chi2(1)      =  0.7808
```

Predicted Numbers from Poisson Regression Model

Age	Smokers			Non-smokers		
	Observed	Pred 1	Pred 2	Observed	Pred 1	Pred 2
35-44	32	27.2	32.0	2	6.8	2.0
45-54	104	98.9	104.0	12	17.1	12.0
55-64	206	205.3	205.0	28	28.7	29.0
65-74	186	187.2	187.0	28	26.8	27.0
75-84	102	111.5	102.0	31	21.5	31.0

Pred 1 No Interaction

Pred 2 Interaction & Constraints

Zeros

- May be structural (Count *had* to be 0)
- Don't count towards DOF
- Lead to problems in estimation
 - IRR is huge or tiny
 - SE is huge
 - Confidence interval is undefined
- Stata may be unable to produce a confidence interval

Overdispersion

- Adding predictors to model may not lead to an adequate fit
- There may be variation between individuals in rate not included in model
- Variance is equal to mean for a Poisson distribution
- The variation between individuals means there is more variation than expected: overdispersion
- If there is overdispersion, standard errors will be too small

Negative Binomial Regression

- Allows for extra variation
- Assumes a mixture of Poisson variables, with the means having a given distribution
- Two possible models:
 - $\text{Var}(Y) = \mu(1 + \delta)$
 - $\text{Var}(Y) = \mu(1 + \alpha\mu)$
- α or δ is the overdispersion parameter
- $\alpha = 0$ or $\delta = 0$ gives the Poisson model.

Negative Binomial Regression in Stata

- Command `nbreg`
- Syntax similar to `poisson`
- Default gives $\text{Var}(Y) = \mu(1 + \alpha\mu)$
- Option `dispersion(constant)` gives $\text{Var}(Y) = \mu(1 + \delta)$

Negative Binomial Regression Example

```
. xi: poisson deaths i.cohort, exposure(exposure) irr
```

```
Poisson regression                               Number of obs   =          21
                                                LR chi2(2)      =          49.16
                                                Prob > chi2     =          0.0000
Log likelihood = -2159.5158                    Pseudo R2       =          0.0113
```

```
-----+-----
```

deaths	IRR	Std. Err.	z	P> z	[95% Conf. Interval]	
-----+-----						
_Icohort_2	.7393079	.0423859	-5.27	0.000	.6607305	.82723
_Icohort_3	1.077037	.0635156	1.26	0.208	.959474	1.209005
exposure (exposure)						

```
-----+-----
```

```
. estat gof
```

```
Goodness-of-fit chi2 = 4190.689
Prob > chi2(18)     = 0.0000
```

```
. xi: nbreg deaths i.cohort, exposure(exposure) irr
```

```
Negative binomial regression          Number of obs   =          21
                                      LR chi2(2)       =           0.40
                                      Prob > chi2      =           0.8171
Log likelihood = -131.3799            Pseudo R2       =           0.0015
```

deaths	IRR	Std. Err.	z	P> z	[95% Conf. Interval]	
+_Icohort_2	.7651995	.5537904	-0.37	0.712	.1852434	3.160869
+_Icohort_3	.6329298	.4580292	-0.63	0.527	.1532395	2.614209
exposure (exposure)						
+/lnalpha	.5939963	.2583615			.087617	1.100376
alpha	1.811212	.4679475			1.09157	3.005294

```
Likelihood-ratio test of alpha=0:  chibar2(01) = 4056.27 Prob>=chibar2 = 0.000
```

Log-Linear Models

- An $R \times C$ table is simply a series of counts
- The counts have two predictor variables (rows and columns)
- Can fit a Poisson model to such a table
- Association between two variables is given by the interaction between the variables
- Model: $\log(p) = \beta_0 + \beta_r x_r + \beta_c x_c + \beta_{rc} x_r x_c$
- For a 2×2 table, such a model is *exactly* equivalent to logistic regression.

Log-Linear Modelling Example

Outcome	Exposure	
	Exposed	Unexposed
Cases	20	10
Non-cases	10	20

OR = 4

Log-linear modelling example: stata output

```

+-----+
| outcome  exposure  freq |
+-----+
1. |      0           0   20 |
2. |      1           0   10 |
3. |      0           1   10 |
4. |      1           1   20 |
+-----+
    
```

```
. xi: poisson freq i.exp*i.out, irr
```

```

Poisson regression                Number of obs   -      4
                                LR chi2(3)      -      6.80
                                Prob > chi2     -      0.0787
Log likelihood = -8.9990653       Pseudo R2     -      0.2741
    
```

```

-----+-----
      freq |          IRR   Std. Err.      z    P>|z|     [95% Conf. Interval]
-----+-----
_1exposure_1 |          .5   .1936492   -1.79   0.074   .2340459   1.068166
_1outcome_1 |          .5   .1936492   -1.79   0.074   .2340459   1.068166
_1expXout_1 |          4   2.19089    2.53   0.011   1.367218   11.7026
    
```

```
. logistic outcome exposure [fw=freq]
```

```

Logistic regression                Number of obs   -      60
                                LR chi2(1)      -      6.80
                                Prob > chi2     -      0.0091
Log likelihood = -38.19085       Pseudo R2     -      0.0817
    
```

```

-----+-----
      outcome | Odds Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-----+-----
      exposure |          4   2.19089    2.53   0.011   1.367218   11.7026
    
```

Direct & Indirect Standardisation

- Used for comparing rates between populations
- Assumes covariates differ
- What would rates be if the covariates were the same ?

Direct Standardisation

- Calculate rate in each stratum
- Standardised rate = weighted mean of these rates
- Weights = proportions of subjects in each stratum of standard population.
- Standardised rate = what rate would be in standard population
- Different standard = different standardised rate
- Can compare directly adjusted rates

Indirect Standardisation

- Per stratum rates are unavailable/unreliable
- Use known rates from a standard population
- Weight known rates according to stratum size our population
- Produce expected number of events if standard rates apply
- Ratio $\frac{\text{Observed}}{\text{Expected}} = \text{SMR}$

Standardisation vs. Adjustment

- Direct standardisation
 - Can use poisson model to get d.s. results
 - Poisson assumes same RR in each stratum
 - D.S. assumes different RR in each stratum
- Indirect Standardisation
 - Good comparison with standard population
 - Can be useful in e.g. observational study of treatment effect.
 - Do not compare SMR's
 - They tell you what happened in observed group.
 - Do not tell you what might happen in a different group.

Generalized Linear Models

- We have met a number of regression models
- All have the form:

$$\begin{aligned}g(\mu) &= \beta_0 + \beta_1 x_1 + \dots + \beta_p x_p \\ Y &= \mu + \varepsilon\end{aligned}$$

where μ is the expected value of Y
 ε has a known distribution (normal, binomial etc)
 $g()$ is called the link function

Components of a GLM

- You can choose the link function for yourself
- It should:
 - Map $-\infty$ to ∞ onto reasonable values for μ
 - Have parameters that are easy to interpret
- Error distribution is determined by the data
- Only certain distributions are allowed

Examples of GLM's

Model	Range of μ	Link	Error Distribution
Linear Regression	$-\infty$ to ∞	$g(\mu) = \mu$	Normal
Logistic Regression	0 to 1	$g(\mu) = \log\left(\frac{\mu}{1-\mu}\right)$	Binomial
Poisson Regression	0 to ∞	$g(\mu) = \log(\mu)$	Poisson

GLM's in Stata

- Command `glm`
- Option `family()` sets the error distribution
- Option `link()` sets the link function
- There are more options to `predict` after `glm`

E.g. `glm yvar xvars, family(binomial) link(logit)`
is equivalent to `logistic yvar xvars`

Setting Reference Category for x_i

```
char variable[omit] #  
char      Characteristic  
variable Name of variable to set reference category for  
#         Value of reference category
```