SAS-intro

Bendix Carstensen

Steno Diabetes Center
& Department of Biostatistics, University of Copenhagen

bxc@steno.dk
www.biostat.ku.dk/~bxc

PhD-course in Epidemiology,
Department of Biostatistics,
Tuesday 12 March, 2011

SAS

- Display manager (programming):
  - program, log, output windows
  - reproducible
  - easy to document
- SAS ANALYST
  - menu-oriented interface
  - writes and runs programs for you
  - no learning by heart, no syntax errors
  - not every thing is included
  - it is heavy to use in the long run

Data set example:

Blood pressure and obesity

OBESE: weight/ideal weight
BP: systolic blood pressure

```
OBS SEX OBESE BP
1 male 1.31 130
2 male 1.31 148
3 male 1.19 146
4 male 1.11 122
. . . .
. . . .
. . . .
101 female 1.64 136
102 female 1.73 208
```

Data

Data are in the text file `BP.TXT` located at `www.biostat.ku.dk/~pka/epidata` and contains the following variables:

- `SEX`: Character variable ($)
- `OBESE`: weight/ideal weight
- `BP`: systolic blood pressure

3 variables and 102 observations

Printing in SAS

We read the file `bp.txt` directly from `www` and skip the first line containing variable names (`firstobs=2`).

```
data bp;
  filename bpfile url ''http://www.biostat.ku.dk/~pka/epidata/bp
  infile bpfile firstobs=2;
  input sex $ obese bp;
run;

proc print data=bp;
  var sex obese bp;
run;
```

A temporary data set `bp` which only exists within the current program. (Permanent data sets may be saved but we will not use this feature in this course.)

SAS programming

- data-step:

```
data bp;
  ( reading ) ;
  ( data manipulations ) ;
run;
```

- proc-step:

```
proc xx data=bp ;
 ( procedure statments ) ;
run;
```

- **NB**: No data manipulations after `run;`
  — only if we make a new data-step.
  — better to revise the first data-step.

Example

```
data bp;
  filename bpfile url ''http://www.biostat.ku.dk/~pka/epidata/bp
  infile bpfile firstobs=2;
  input sex obese bp;
run;

data bp;
  set bp;
  if bp<125 then highbp=0;
  if bp>=125 then highbp=1;
  /* an alternative way of creating the new variable highbp is:
     highbp = (bp>=125); */
run;

proc freq data=bp;
  tables sex * highbp ;
run;
```

Example, simplfied

```
data bp;
  filename bpfile url ''http://www.biostat.ku.dk/~pka/epidata/bp
  infile bpfile firstobs=2;
  input sex obese bp;
  if bp <  125 then highbp=0;
  if bp >= 125 then highbp=1;
  /* an alternative way of creating the new variable highbp is:
     highbp = (bp>=125); */
run;

proc freq data=bp;
  tables sex * highbp ;
run;
```

Typing of programs is done in the

- Program Editor window:
  - Works like all other text editors: arrow keys, backspace, delete etc.
  - When the program is submitted (click on Submit or press F3), the results are in the
- Log-window:
  - Here you can see how things went:
  - how many observations you have,
  - how many variables you have
  - if there were any errors
  - which pages were written by which procedures

- Output-window (perhaps):
  - In this window you will find the results (if there are any)
- Graph-window (which we won't use on this course)
  - Here plots are stored in order

Making life simpler

- You can move between the windows by clicking Windows in the command bar, or use that:
  - F5 is editor window,
  - F6 is log window,
  - F7 is output window.

Modifications in the program

When the program has been executed and you want to make changes:

- Go back to the Program-window
- The Log- Output- and Graph-windows cumulate, that is output is stored consecutively
- Clear by choosing Clear under Edit (or press Ctrl-E - for "erase")
- Don't print!
- Remember to save the the program from time to time before SAS crashes!

Simple statistical models
Proportions and rates

Bendix Carstensen

Steno Diabetes Center
& Department of Biostatistics, University of Copenhagen

bxc@steno.dk
www.biostat.ku.dk/~bxc

PhD-course in Epidemiology,
Department of Biostatistics,
Tuesday 12 March, 2011

A single proportion

The log-likelihood for π, the proportion dead, if we observe 4 deaths out of 10:

$$\ell(\pi) = 4\log(\pi) + 6\log(1-\pi)$$

The log-likelihood for ω, the odds of dying, if we observe 4 deaths and 6 non-deaths:

$$\ell(\pi) = 4\log(\omega) - 10\log(1+\omega)$$

Programs

General purpose programs for estimating in the binomial and Poisson distribution:

- **SAS**: `proc genmod`
- **R**: `glm`
- **Stata**: `glm`

Here we primarily look at SAS.

Estimating odds: genmod

```
data p ;
  input x n ;
  datalines ;
  4 10
  ;
run ;

proc genmod  data= p ;
  model x/n = / dist=bin link=logit ;
  estimate "4 versus 6" intercept 1 / exp ;
run ;
```

Parameter	DF	Estimate	Standard Error	Wald 95% Confidence Limits	
Intercept	1	-0.4055	0.6455	-1.6706	0.8597
Scale	0	1.0000	0.0000	1.0000	1.0000

Label	L'Beta Estimate	Standard Error	Contrast Estimate Re L'Beta Confidence Limits		C Squ
4 versus 6	-0.4055	0.6455	-1.6706	0.8597	0
Exp(4 versus 6)	0.6667	0.4303	0.1881	2.3624	

Estimating a proportion: genmod

The only difference from estimation of odds is the `link=` argument, which is changed to `log` (instead of `logit`):

```
proc genmod  data= p ;
  model x/n = / dist=bin link=log ;
  estimate "4 out of 10" intercept 1 / exp ;
run ;
```

Parameter	DF	Estimate	Standard Error	Wald 95% Confidence Limits	
Intercept	1	-0.9163	0.3873	-1.6754	-0.1572
Scale	0	1.0000	0.0000	1.0000	1.0000

Label	L'Beta Estimate	Standard Error	Contrast Estimate Re L'Beta Confidence Limits		C Squ
4 out of 10	-0.9163	0.3873	-1.6754	-0.1572	5
Exp(4 out of 10)	0.4000	0.1549	0.1872	0.8545	

A single proportion: R: `glm`

So simple that we do odds and proportion in one slide:

```
> library( Epi )

> ci.exp( glm( cbind(4,6) ~ 1, family=binomial(link=log) ) )
             exp(Est.)      2.5%     97.5%
(Intercept)        0.4 0.1872367 0.8545332

> ci.exp( glm( cbind(4,6) ~ 1, family=binomial ) )
             exp(Est.)      2.5%    97.5%
(Intercept) 0.6666667 0.1881311 2.362419
```

A single proportion: individual records

```
data bissau;
  filename bisfile url "http://www.biostat.ku.dk/~pka/epidata/bi
  infile bisfile firstobs=2;
  input id fuptime dead bcg dtp age agem;
run;

title "Estimate odds - Bissau" ;
proc genmod  data=bissau  descending ;
  model dead = / dist=bin link=logit ;
  estimate "odds of dying" intercept 1 / exp ;
run ;
```

Contrast Estimate Results

Label	L'Beta Estimate	Standard Error	L'Beta Confidence Limits		S
odds of dying	-3.1249	0.0686	-3.2593	-2.9905	2
Exp(odds of dying)	0.0439	0.0030	0.0384	0.0503	

A single proportion: individual records

```
title "Estimate proportion - Bissau" ;
proc genmod  data=bissau  descending ;
  model dead = / dist=bin link=log ;
  estimate "prob of dying" intercept 1 / exp ;
run ;
```

Label	L'Beta Estimate	Standard Error	Contrast Estimate L'Beta Confidence Limits		R S
prob of dying	-3.1679	0.0657	-3.2966	-3.0391	2
Exp(prob of dying)	0.0421	0.0028	0.0370	0.0479	

Likelihood for a single rate

Recall the log-likelihood for a single rate, λ based on D events during Y person years:

$$D\log(\lambda) - \lambda Y$$

This is also the log-likelihood for a Poisson variate D with mean $\mu = \lambda Y$.

Therefor we can use a program for the Posson distribution to estimate rates, except we must "remove" the Y from the mean.

Poisson distribution usually use the log-mean:

$$\log(\mu) = \log(\lambda) + \log(Y)$$

$\log(Y)$ extracted via the OFFSET argument.

A single rate

```
data r ;
  input d y ;
  ly = log(y) ;
  my = log(y/1000) ;
  datalines ;
  30 261.9
  ;
run ;

title "Estimate a rate per 1 year" ;
proc genmod  data= r ;
  model d = / dist=poisson link=log offset=ly ;
  estimate "30 during 261.9 - per 1 year" intercept 1 / exp ;
run ;
```

	Contrast Estimate Resul		
	L'Beta	Standard	L'
Label	Estimate	Error	Confide
30 during 261.9 - per 1 year	-2.1668	0.1826	-2.5246
Exp(30 during 261.9 - per 1 year)	0.1145	0.0209	0.0801

A single rate: Scaling

Remember the `data step` statement: `my = log(y/1000) ;`

```
title "Estimate a rate per 1000 year" ;
proc genmod  data= r ;
  model d = / dist=poisson link=log offset=my ;
  estimate "30 during 261.9 - per 1000 years" intercept 1 / exp
run ;
```

Label	Contrast L'Beta Estimate	Estimate Standard Error	Result Alph
30 during 261.9 - per 1000 years	4.7410	0.1826	0.0
Exp(30 during 261.9 - per 1000 years)	114.5475	20.9134	0.0

A single rate: individual records

```
data bissau ;
  set bissau ;
  ld = log(fuptime) ;
  ly = log(fuptime/36525) ;
run ;

title "Estimate a rate per 1 day" ;
proc genmod  data=bissau ;
  model dead = / dist=poisson link=log offset=ld ;
  estimate "mortality rate - per 1 day" intercept 1 / exp ;
run ;
```

Label	L'Beta Estimate	Contrast Estimate Standard Error	Alpha	C
mortality rate - per 1 day	-8.2852	0.0671	0.05	-
Exp(mortality rate - per 1 day)	0.0003	0.0000	0.05	

Single rate individual records, scaling

Remember the `data step` statement: `ly = log(fuptime/36525) ;`

```
title "Estimate a rate per 1 year" ;
proc genmod  data=bissau ;
  model dead = / dist=poisson link=log offset=ly ;
  estimate "mortality rate - per 100 years" intercept 1 / exp ;
run ;
```

Label	L'Beta Estimate	Contrast Estimate Standard Error	Re Alpha
mortality rate - per 100 years	2.2205	0.0671	0.05
Exp(mortality rate - per 100 years)	9.2123	0.6183	0.05

A single rate: R

```
> library( Epi )

> D <- 30 ; Y <- 261.9

> ci.exp( glm( D ~ 1, offset=log(Y     ), family=poisson ) )
             exp(Est.)       2.5%     97.5%
(Intercept) 0.1145475 0.08009009 0.1638297

> ci.exp( glm( D ~ 1, offset=log(Y/1000), family=poisson ) )
             exp(Est.)     2.5%    97.5%
(Intercept)  114.5475 80.09009 163.8297
```

A single rate: R, individual records

```
> bis <- read.table("../data/bissau.txt", header=TRUE )

> ci.exp( glm( dead ~ 1, offset=log(fuptime)       , family=poiss
                exp(Est.)        2.5%        97.5%
(Intercept) 0.0002522191 0.000221131 0.0002876779

> ci.exp( glm( dead ~ 1, offset=log(fuptime/36525), family=poiss
              exp(Est.)     2.5%    97.5%
(Intercept)  9.212304 8.076808 10.50744
```