Monday am

10.45 – 12.00

A brief introduction to Stata

1 The births data

To introduce you to Stata we use data from births to 500 mothers. The variables in the dataset are shown in Table 1.

Variable	Units or Coding	Type	Name
Subject number	–	categorical	`id`
Birth weight	grams	metric	`bweight`
Birth weight < 2500 g	1=yes, 0=no	categorical	`lowbw`
Gestational age	weeks	metric	`gestwks`
Gestational age < 37 weeks	1=yes, 0=no	categorical	`preterm`
Maternal age	years	metric	`matage`
Maternal hypertension	1=hypertensive, 0=normal	categorical	`hyp`
Sex of baby (numeric)	1=male, 2=female	categorical	`sex`
Sex of baby (alphabetic)	"male", "female"	categorical	`sexalph`

Table 1: Variables in the births dataset

To follow the practical type in the commands which start with the stata prompt. Don't type the `.` prompt. Remember that stata distinguishes between upper and lower case letters, and that it accepts abbreviations for both commands and variable names. Think carefully about what is happening after each command.

2 A first look at the data

The file `births.dta` contains the variables names and values for the 641 records. To read the data, type

```
. use births, clear
. describe
```

A good way to start the analysis is to ask for a summary of the data by typing

```
. summarize
```

This will produce the mean, standard deviation, and range, for each variable in turn. In most datasets there will be some missing values. These are coded using the symbol `.` in place of the value which is missing. Stata can recognize other codes for missing values, but this is the one which is recommended. The `summarize` command is useful for seeing whether there are missing values. For a more detailed summary of the variable `gestwks` try

```
. codebook gestwks
```

or

```
. summarize gestwks, detail
```

The list command is used to list the values in the data file. Try out the following and see their consequences:

```
. list in 1/5
. list matage in 1/10
. list matage
. list matage bweight in 1/20
```

Stata stops after each screenful of output. Click on more to get another screenful, or press ENTER to continue line by line. The command `list` on its own would list all of the data. You can cancel this command (and any other Stata command) by clicking on Break.

When starting to look at any new data the first step is to check that the values of the variables make sense and correspond to the codes defined in the coding schedule. For categorical variables this can be done by looking at one-way frequency tables and checking that only the specified codes occur. For metric variables we need to look at ranges.

This first look at the data will also indicate whether all values are present or whether there are some missing values on some variables. Let us begin by looking at the categorical variables. The distribution of the categorical variables `hyp` and `sex` can be viewed by typing

```
. tabulate hyp
. tab sex
```

Their cross-tabulation is obtained by typing

```
. tab hyp sex
```

Cross tabulations are useful when checking for consistency. The basic output from a cross tabulation reports frequencies only; to include row and/or column percentages add the options `row`, `col`, `cell`, or any combination, as in

```
. tab hyp sex, col
```

The command `table` is used for preparing tables of summary statistics by one, two, or even more categorical variables. For example, to obtain the means and standard deviations of `bweight` separately by `sex`, type

```
. table sex, contents(freq mean bweight sd bweight)
```

To make a table of the median and interquartile range for birthweight, by sex, try

```
. table sex, contents(freq med bweight iqr bweight)
```

Note that `tab` is an abbreviation for `tabulate`, NOT for `table`, which must be typed in full.

3 Restricting commands

Stata commands can be restricted to records $1, 2, \ldots, 10$ (for example), by adding `in 1/10` to the command. The letters `f` and `l` can be used as abbreviations for first and last, so `20/l` refers to the records from 20 onwards. Commands can also be restricted to operate only on records which satisfy given conditions. The conditions are added to the command using `if` followed by a logical expression which takes the values true or false. For example, to restrict the command `list` to records with birthweight less than or equal to 2000g, type

```
. list id bweight if bweight <= 2000
```

If the logical expression `bweight <= 2000` is true the record is listed, but not otherwise. Other useful logical expressions are `==` for equal, and `!=` for not equal. A common error is to use `=` in a logical expression instead of `==` .

A useful command when exploring data is `count` which counts the number of records which satisfy some logical expression. For example

```
. count if bweight <= 2000
. count if bweight <= 2000 & sex==1
```

> *Note the use of* `&` *to link two conditions both of which must be satisfied.*

4 Generating and recoding variables

New variables are generated using the command `generate`, and variables can be recoded using `recode`. For example, to create a new variable `sex2` which is the same as `sex` but coded 1 for male and 0 for female, try

```
. gen sex2=sex
. recode sex2 2=0
. tab sex2
```

It is possible to use `recode` to group the values of a metric variable such as `matage`, but it is more conveniently done with `egen` and `cut`. For example

```
. egen agegrp=cut(matage), at(20,30(5)45)
. tab agegrp
```

creates a new variable `agegrp` which takes the value 20 when `matage` is between 20 and 30, 30 when `matage` is between 30 and 35, and so on. If you prefer the integer codes 0, 1, 2, ..., try

```
. drop agegrp
. egen agegrp=cut(matage), at(20,30(5)45) icodes
. tab agegrp
```

If you are too lazy to choose break points try

```
. drop agegrp
. egen agegrp=cut(matage), group(5)
. tab agegrp
```

which will produce 5 roughly equal–frequency groups coded 0, 1, 2, 3, 4.

5 Sorting

The records in a dataset can be sorted according to the values of one or more variables. The `births` dataset is currently sorted by `id` but for some purposes it might be better to have it sorted by `bweight`. Try

```
. list id bweight in 1/10
. sort bweight
. list id bweight in 1/10
```

The records are now in order of `bweight` and the `id` numbers and all other variables have also been sorted in this order.

Stata commands which use the option `by()` usually require the data to be first sorted by the variable in the `by()` option. The sort is not done automatically because you should always be aware of how your data are sorted.

6 Using Stata as a calculator

The `display` command can be used to carry out simple calculations. For example, the command

```
. display 2+2
```

will display the answer 4, while

```
. display ln(10)
```

will display the answer 2.3026, the natural log of 2. Standard probability functions can also be displayed, as in

```
. display chiprob(1,3.84)
```

which will return the probability of exceeding 3.84 in a χ^2 distribution on 1 degree of freedom (0.05).

7 Graphical displays

The `graph` command has many options. Bar charts are used to display the distributions of categorical variables, while histograms and box plots are used to display the distributions of metric variables.

To obtain a histogram of `bweight`, type

```
. graph bweight
```

To improve the visual display, add `xlabel ylabel` to this command, anywhere after the comma. You can vary the number of rectangles in the histogram (called bins) by adding `bin(8)`, etc. To superimpose the histogram with a normal curve which has the same mean and standard deviation as the data, add the option `normal`. Try, for example,

```
. graph bweight, xlabel ylabel bin(8) normal
```

Categorical variables do not require grouping and the histogram is a natural choice for displaying the frequency distribution for a variable such as `agegrp`. In the `graph` command Stata chooses the widths by dividing the range by the number of bins, and this can produce bins which are not centered on the variable values. An alternative command, which avoids this difficulty, is

```
. hist agegrp
```

Cumulative probability plots (also called cumulative distributions) are better than histograms for continuous metric data because they don't require you to choose grouping intervals. Try

```
. cdf bweight
. cdf bweight, by(preterm)
. cdf bweight, by(preterm) normal
. cdf bweight, by(preterm) normal same
```

to see how cumulative distributions are used.

Scatter plots can be used to evaluate the association between `bweight` and the metric variable `matage` by typing

```
. graph bweight matage, xlab ylab
```

To change the plotting symbol, try

```
. graph bweight matage, xlab ylab s(.)
```

To plot `bweight` against `gestwks`, try

```
. graph bweight gestwks, xlabel ylabel
```

8 Missing values

The missing value symbol in stata is . and is treated as plus infinity in logical comparisons. Stata commands automatically exclude missing values when they are coded in this way.

9 Saving data files

The stata data currently in memory can be saved in a file called `mydata.dta` with the command `save mydata`. For example, to read the births data from the dictionary file `births.dct`, and then to save it as a stata file called `mydata`, try

```
. infile using births.dct, clear
. save mydata
```

If the file `mydata.dta` already exists you must give permission to overwrite it by using

```
. save mydata, replace
```

The file `mydata.dta` is a binary file, and cannot be read by anything except stata.

It is good practice to lock up your data file at an early stage, and *never* to overwrite it.

10 Dates

Dates often start as string variables which contain the date in the form (say) dd/mm/yy, or to be 2000 aware, dd/mm/yyyy. A simple example is shown below and is in the file `dates`:

1. Read in the data and describe with

   ```
   . use dates, clear
   . desc
   . list
   . codebook start
   ```

   You will see that `start` is a string variable.

2. Generate a variable `datein` in which the dates are in stata form with

   ```
   . gen datein = date(start, "dmy")
   . list
   ```

The function `date()` can work out what is there from the string "dmy" which gives the order. Note that `datein` contains days since 1/1/1960, and is numeric.

3. Format the variable `datein` so that it prints as a date, using

```
. format datein %d
. list
```

11 Some practice with basic commands

1. Load the `births` data, and list the variables `bweight` and `hyp` for records 20–25 inclusive.
2. Summarize all variables
3. Summaraize `matage` in detail
4. Use `count` to find how many hypertensive women have babies with birthweight less than 2000g.
5. Create a new variable, `gest4`, by grouping the values of `gestwks` using the cut-points 20,35,37,39,45.
6. Obtain a frequency table for the categorical variable `gest4` using `tabulate`.
7. Obtain a table of mean birth weight by categories of `gest4`, using `table`.

Monday pm

14.30 – 16.00

Making tables

1 Response and explanatory variables

Most questions in statistical analysis take the form of asking whether the value which one variable takes for a given subject depends on the value taken by another variable. For example, in the `births` data we might be interested in whether the birth weight of a baby depends on whether it is a boy. The variable which is of primary interest is called the *response* variable; the variable on which the response variable may depend is called the *explanatory* variable. In the example just quoted, the response variable is birth weight and the explanatory variable is whether the baby is a boy.

In biostatistics four types of response variables are particularly common:

1. Binary
2. Metric
3. Failure
4. Count

A binary response has just two values which should be coded 0 and 1. A metric response (also called a quantitative response) measures some quantity and usually has many possible values. A failure response indicates whether or not a subject fails at the end of a period of observation, and is used with survival data. Finally, a count response records a number of events, and often arises with aggregated failure data. The type of response determines how it will be summarized. For example, a binary response is usually summarized using the proportion of 1's, and a metric response is usually summarized using its mean or median.

The following questions illustrate response and explanatory variables and refer to the `births` data.

Does the birth weight of a baby depend on whether the mother was hypertensive?

The response variable is `bweight` which is metric and the explanatory variable is `hyp`. Because the response is metric the distribution of response can be summarized using either the mean or the median. Such a table of means, for example, can be produced with the commands

```
. use births, clear
. table hyp, contents(freq mean bweight)
```

Does low birth weight depend on the sex of the baby?

The response variable is now `lowbw` which is binary and the explanatory variable is `sex`. The distribution of `lowbw` by `sex` is given by the relative frequencies of its values, obtained with

```
. tabulate lowbw sex, col
```

which shows that 10% of male babies are of low birth weight, while 14% of female babies are of low birth weight. The `table` command is meant primarily for summarizing a metric response, but it can be used with a binary response provided that this is coded 0/1. Thus

```
. table sex, contents(freq mean lowbw)
```

again shows that 10% of male babies are of low birth weight, while 14% of female babies are of low birth weight. This works because the mean of a variable coded 0/1 is equal to the relative frequency of the 1's.

2 Declaring the type of response

To facilitate the preparation of tables for a variety of responses a Stata menu invoked by the command `tabmenu1`[1] invites you to specify the type of response. Consider again the question of whether `bweight` varies with `hyp` and try

```
. tabmenu1, clear
```

The different parts of the menu are used as follows:

- Select the response variable – choose `bweight`.
- Select the type of response – choose `metric`.
- Select follow-up time variable if appropriate. This is needed with survival data – ignore it for now.
- Select the explanatory variable which will be used for the rows of the table – choose `hyp`.
- Select the explanatory variable which will be used for the columns of the table – ignore it for now.

From here on we shall abbreviate these instructions as follows:

```
. tabmenu1, clear
---> select bweight as response
---> select metric as type
---> select hyp as rows
```

Note that only the first line is typed – all the rest are instructions to follow within the menu. The option `clear` is used with `tabmenu1` to remove any previous selections, so the menu will start off with blanks.

[1] The command `tabmenu1` is not part of official Stata

3 Producing tables

Clicking on the Tables button in the first menu calls up another menu. Because the response which was chosen in `tabmenu1` is metric you are offered three possibilities for summarizing the response - the mean, the geometric mean, and the median. Select Mean, and press OK. You should see something like this:

```
Response variable is: bweight which is metric
Row variable is: hyp
Number of records used:  500

Summary using means
----------------------
hypertens |    bweight
----------+-----------
        0 |    3198.90
        1 |    2768.21
----------------------
```

The figures in the table are the mean values of the babies' birth weights for mothers who were normal (`hyp=0`) or hypertensive (`hyp=1`). The default is to provide minimum information, because it is often easier to start with this, but to get more you could check (tick) the boxes for frequencies and confidence intervals that appear in the second menu. Try

```
. tabmenu1
---> click on Tables
     ---> select Mean
     ---> check Frequencies
     ---> check Confidence intervals
     ---> click on OK
```

The second level of indentation above refers to the second menu.

With a binary response, such as `lowbw`, the choice of summary is different. Try

```
. tabmenu1, clear
---> select lowbw as response
---> select binary as type
---> select hyp as rows
---> click on Tables
```

Because the response is binary you will be offered a choice between Proportions and Odds. Select Proportions and click on OK. You should see something like this

```
Response variable is: lowbw which is binary
Row variable is: hyp
Number of records used:  500
```

```
Summary using proportions per 100
----------------------
hypertens |      lowbw
----------+-----------
        0 |       9.35
        1 |      27.78
----------------------
```

The figures in the table are the percentages of low birth weight babies for mothers who were normal or hypertensive. To produce 90% confidence intervals for these proportions, try

```
. tabmenu1
---> click on Tables
     ---> check Confidence intervals
     ---> enter 90 in the Level of confidence box
     ---> click on OK
```

The command `tabmenu1` works entirely through menus, but there is an equivalent command, `tabmenu2`, which does the same thing without menus. Using the option `display` with `tabmenu1` will show the equivalent `tabmenu2` command, and this can then be cut and pasted into the Command window or a do file. For example, try

```
. tabmenu1, display
```

and repeat the table you have just created. You should now see the line

tabmenu2, res(lowbw) typ(binary) row(hyp) summ(prop) ci level(90)

displayed in the Results window. Cut and paste this into the Commands window using Edit/Copy Text and Edit/Paste, press return, and you will see the same table again. The command can be edited, so small changes can be made without going through the menus again, but the main use of this facility is to include tables produced by the `tabmenu1` command in a do file, without having to fill out the menus. There is more information about `tabmenu1` and `tabmenu2` in the help files.

4 A second explanatory variable

We found a strong relationship between birth weight and whether the mother was hypertensive, but is this relationship the same for both male and female babies, i.e. is it *modified* by `sex`? To study this we need to produce a table of mean birth weight by both `hyp` and `sex`. Using

```
. use births, clear
. table hyp sex, contents(freq mean bweight)
```

we find that the birth weight of both male and female babies is lower when the mother is hypertensive than when the mother is normal – about 500 g lower for males babies and about 400 g for female babies.

To produce the same table using `tabmenu1`, try

```
. tabmenu1, clear
---> select bweight as response
---> select type as metric
---> select hyp as rows
---> select sex as columns
---> click on Tables
     ---> select Mean
     ---> click on OK
```

To reverse the rows and columns, check the box marked Reverse variables in the second menu that appears after clicking on Tables.

5 Odds

Odds are less familiar than proportions, but they measure the same thing. When 60 babies out of 500 are low birth weight the proportion is $60/500 = 0.12$, while the odds of being low birth weight are $60/440 = 0.1364$.

To make a table of the odds of the baby being low birth weight for normal and hypertensive mothers, try

```
. tabmenu1, clear
---> select lowbw as response
---> select binary as type
---> select hyp as rows
---> click on Tables
     ---> select Odds
     ---> click on OK
```

To obtain frequencies and confidence intervals for the odds you can check the appropriate boxes. The Stata command `tabodds` will also tabulate odds, but only by one explanatory variable:

```
. tabodds lowbw hyp
```

Tables with too many rows are not particularly useful, and for this reason an upper limit of 10 values for the row and column variables has been set in `tabmenu1`. This can be increased if required. Try

```
. tabmenu1, clear
---> select lowbw as response
---> select binary as type
```

```
---> select matage as rows
---> click on Tables
     ---> select Odds
     ---> click on OK
```

and you will get a message telling you that there are too many values in the row variable (`matage`). Try again, and set the maximum number of values that a row or column variable can have to 25, select Odds as the summary, and you will see a table of the odds of being low birth weight by maternal age. This is not very useful, and it would be better to group the values of `matage` using `egen` with `cut`.

6 Case-control studies

Odds are important in case-control studies where, instead of sampling all subjects equally, a different sampling fraction is used for subjects who have a disease (the cases) than for those who do not (the controls). This is called *outcome-based sampling* in econometrics. As an example, we shall look at a study of physical activity at work and tuberculosis (TB), one of the first case-control studies to be carried out (see Table 2). The cases were cases of TB among outpatients at a hospital, and the controls were chosen without matching from outpatients at the same hospital, who were not suffering from TB[6]. Start by listing the data in the file `guy.dta` with

```
. use guy, clear
. list
. list, nolabel
```

The variable `level` is coded 1, 2, 3, 4 for the four levels of activity, and the variable `d` is coded 1 for a case and 0 for a control. The data are aggregated, and the variable `N` contains the frequency with which each combination of `level` and `d` occurs. One way of dealing with data of this kind is to make it into individual records with the command `expand`:

```
. expand N
```

Table 2: Physical activity at work for 1659 outpatients

Level of physical activity	Tuberculosis (Cases)	Other diseases (Controls)
Little (1)	125	385
Varied (2)	41	136
More (3)	142	630
Great (4)	33	167
Total	341	1318

```
. drop N
. summarize
```

There are now two possible approaches to the analysis (Clayton & Hills, 1993[4]). In the *retrospective* approach we argue from disease back to exposure so the response is `level`, and the explanatory variable is `d`. The values of `level` are measuring physical activity on some sort of metric scale, so we shall treat `level` as metric, and try

```
. tabmenu1, clear
---> select level as response
---> select metric as type
---> select d as rows
---> click on Tables
     ---> click on OK
```

You should see something like

```
Response variable is: level which is metric
Row variable is: d

Summary using means

        d |      level
----------+-----------
        0 |       2.44
        1 |       2.24
```

which shows that cases of TB were, on average, less physically active than controls. In the *prospective* approach we argue from exposure forward to disease so the response is `d`, which is binary, and the explanatory variable is `level`. Try

```
. tabmenu1, clear
---> select d as response
---> select binary as type
---> select level as rows
---> click on Tables
     ---> select Odds
     ---> click on OK
```

and you should see something like

```
Response variable is: d which is binary
Row variable is: level

Summary using odds

activity  |          d
----------+-----------
```

```
   little |     0.3247
   varied |     0.3015
     more |     0.2254
    great |     0.1976
----------------------
```

which shows that the odds of being a case decreases with the level of physical activity.

Note that the odds which are being tabulated refer to the odds of being a case in the study, not the population. However, it can be shown that

$$\text{Odds in study} = K \times \text{Odds in population},$$

where K is the ratio between the sampling fractions for cases and controls, so provided the sampling fractions do not depend on `level`, it follows that if the study odds are going down with `level`, then the population odds are also going down with `level`. See Clayton & Hills (1993), p153[4] for a more detailed discussion of this point.

Now you can see why odds are chosen for the analysis of case-control studies: suppose that, instead of 1318 controls, there had been 10 times as many; the odds would now be (apart from random variation),

```
Summary using odds

activity  |          d
----------+-----------
   little |     0.03247
   varied |     0.03015
     more |     0.02254
    great |     0.01976
```

So although the odds have changed drastically because of the change in sampling fraction of the controls, the trend in the odds with `level` is unchanged. Note that this simple relationship between the odds and the sampling fractions does not hold for proportions.

Both retrospective and prospective analyses are useful, but on the whole the prospective one is more informative.

7 Survival data and rates

Data involving survival times are often summarized using *rates*, i.e. the number of events per unit time. There are no survival time variables in the `births` data, so we need another dataset to demonstrate how to make tables of rates. Try

```
. use diet, clear
. describe
```

These data refer to a follow–up study of 337 male subjects who were asked to weigh the different components of their diet for a week[8]. They were then followed until

1. They developed, and possibly died from, coronary heart disease (CHD)
2. They died from some other cause, or were withdrawn from the study for some reason, or the study ended.

The time for which each subject is followed is the true survival time in the first case, but in the second case the true survival time has been *censored* by death from another cause, withdrawal, or the end of the study. To record data with true and censored survival times we need two variables: the time spent in the study and a variable which indicates whether the subject developed CHD or not. These are called the *time* and *failure* variables, respectively. In the absence of censoring, the failure variable takes the value 1 for all subjects, and the response variable is time, but when there is censoring, the response is a combination of the time and the failure variables.

In this example the time variable is `y`, and the failure variable is `chd`, coded 1 if the subject developed coronary heart disease (CHD) during the period of the study, and 0 otherwise. For a preliminary analysis the total energy intake per day is converted to a binary variable `hieng` coded 1 if the energy intake is > 2750 kcal, and 0 otherwise. To create a table of rates for `chd` by `hieng`, try

```
. tabmenu1, clear
---> select chd as response
---> select failure as type
---> select y as follow-up time
---> select hieng as rows
---> click on Tables
     ---> select Rates per 1000
     ---> click on OK
```

Note that with rates the failure variable is selected as response. Rather unexpectedly, eating a lot seems to prevent CHD (7.07 per 1000 years compared with 13.60 per 1000 years). This is because what you eat is largely determined by your level of physical activity, and a high level of physical activity helps prevent CHD.

The same table of rates can also be created using the `table` command by creating a rate for each subject, and then weighting these by the follow-up time for each subject. Try

```
. gen rate=chd/y
. table hieng [iw=y], content(mean rate)
```

Unfortunately `table` cannot produce confidence intervals for the rates.

8 Count data and rates

An example where the response variable is a count arises in the mortality data taken from Rothman (1986)[9]. These data refer to the total deaths and populations in 1962 in Panama and Sweden, by three age categories. Load the data with

```
. use mortality, clear
. describe
. list
```

The data are aggregated, so the response is the total number of deaths in each age x country category, which is in the variable `deaths`. Count data requires a follow-up time, and assuming each subject is followed for the whole of 1962, this is equal to the population multiplied by 1, which is the same as the variable `pop`. To find the mortality rate by nation using `tabmenu1`, try

```
. tabmenu1, clear
---> select deaths as response
---> select count as type
---> select pop as follow-up time
---> select nation as rows
---> click on Tables
     ---> select Rates per 1000
     ---> click on OK
```

Tuesday am

10.45 – 12.00

Effects

1 Comparing means using effmenu

Earlier we prepared a table of mean `bweight` by `hyp` using `tabmenu1`. This showed that hypertensive mothers had babies which were on average about 430 g smaller than those delivered to normal mothers. This difference in mean birth weight is called the *effect* of hypertension on birth weight. The command `effmenu1`, like `tabmenu1`, brings up a menu which first invites you to specify the type of response.[2] When you bring up this menu you will see that the term *exposure variable* is used instead of *explanatory variable*. The reason for this change is that when calculating effects it is important to distinguish between the different roles which the explanatory variables might have in an analysis. The explanatory variable whose effects you want to calculate is called the *exposure* variable; *control* and *modifying* variables will be introduced later. In this example the response variable is `bweight` and the exposure variable is `hyp`, so try

```
. use births, clear
. effmenu1, clear
---> select bweight as response
---> select metric as type
---> select hyp as exposure
---> click on Effects
     ---> select Difference in means
     ---> click on OK
```

The option `clear` after `effmenu1` removes any selections which might remain from a previous use of this command. You will see that the effect of `hyp` on `bweight`, measured using the difference in the mean response, is -430.7 g. The 95% confidence interval for this effect is from -276 to -586 g. The statistical test is for the null hypothesis that the true effect of hypertension is zero. It takes the form of an F statistic, because the response is metric, and is on 1 and 498 degrees of freedom (df). The 1 df is because one effect is estimated, and in this situation the F-test is equivalent to the t-test. The P-value is very low so there is strong evidence against the null hypothesis.

Now cut `gestwks` into 4 groups with

```
. egen gest4=cut(gestwks), at(20,35,37,39,45)
. tabulate gest4
```

When comparing the mean birth weight between the four different levels of `gest4` there will be three effects: the effect comparing level 2 with level 1; level 3 with level 1; and level 4 with level 1. The level with which each of the other levels is compared is called the *baseline* (level 1 in this case). To prepare a table of the three effects of `gest4`, try

```
. effmenu1
---> select gest4 as exposure
```

[2] The command `effmenu1` is not part of official Stata.

```
---> click on Effects
     ---> select Difference in means
     ---> click on OK
```

There are three effects, and the statistical test is for the null hypothesis that the true values of these effects are all zero. It takes the form of an F statistic, because the response is metric, and is on 3 df because 3 effects are tested. The second menu allows you to change the baseline from its default value of 1. Try changing the baseline to 3: each of the levels 1, 2, 4 is now compared with level 3.

Like `tabmenu1`, the command `effmenu1`, with the option `display`, will display the equivalent `effmenu2` command which works without menus. See the help on `effmenu1` and `effmenu2` for more information.

2 Comparing proportions and odds

When examining the proportion of low birth weight babies according to the length of their gestation, using gestation time in four groups, each level of `gest4` can be compared with the baseline level using the difference in proportions, or the ratio of proportions, or the ratio of odds. The preferred method is to use the ratio of odds.

Start by using `tabmenu1` to prepare a table of the odds that a baby has low birth weight, by the levels of `gest4`, and note that the odds are 4.1667 for level 1 of `gest4`, 0.6842 for level 2, 0.1208 for level 3, and 0.0117 for level 4. Using ratios of odds to measure effects we should get $0.6842/4.1667 = 0.1642$ for level 2 compared with level 1, and so on. Now try

```
. effmenu1, clear
---> select lowbw as response
---> select binary as type
---> select gest4 as exposure
---> click on Effects
     ---> select Odds ratios
     ---> click on OK
```

You should see this table of three effects comparing levels 2, 3, 4 against level 1, using odds ratios:

```
                      Effect    95% Confidence Interval

Level 2 vs level 1    0.1642   [ 0.053 , 0.512 ]
Level 3 vs level 1    0.0290   [ 0.010 , 0.080 ]
Level 4 vs level 1    0.0028   [ 0.001 , 0.012 ]
```

Compared with level 1 of `gest4`, mothers at level 2 have lower odds of a low birth weight baby by a factor of 0.1642, mothers at level 3 have lower odds by a factor of 0.0290, while mothers at level 4 have lower odds by a factor of 0.0028.

The statistical test that appears below the table is for the null hypothesis that the true values of these three effects (odds ratios) are all 1. It takes the form of a chi–squared statistic, because the response is binary, and is on 3 df because there are three effects being tested.

We have chosen to measure the effects of `gest4` as odds ratios, but they can also be measured using the ratios of proportions. Try

```
. effmenu1
---> click on Effects
     ---> select Ratio of proportions
     ---> click on OK
```

and you will see a table showing the three effects of `gest4` as ratios of proportions instead of odds ratios. You should be aware that when using the ratio or difference of proportions to measure effects, the program may fail to reach an answer for datasets where some of the proportions being compared are close to 0 or 1.

3 Case-control studies

The measurement of effects in a prospective analysis of unmatched case-control studies is carried out in the same way, but this time it is essential to select odds ratios (Section 6). Try

```
. use guy, clear
. expand N
. drop N
. effmenu1, clear
---> select d as response
---> select binary as type
---> select level as exposure
---> click on Effects
     ---> select Odds ratios
     ---> click on OK
```

and you should see a table of odds ratios comparing level 2 of physical activity with level 1, level 3 with level 1, and level 4 with level 1.

4 Comparing rates

Load the `diet` data and cut `energy` into 3 groups with

```
. use diet, clear
. egen eng3=cut(energy),at(1500,2500,3000,4500)
. tabulate eng3
```

Use `tabmenu1` to prepare a table showing the rates of CHD for different levels of `eng3`, and note that the rate goes down from 16.90 to 4.88 per 1000, with increasing level of `energy`. The two effects of `eng3` can be measured as rate differences or rate ratios. To prepare a table of effects using rate ratios, try

```
. effmenu1, clear
---> select chd as response
---> select failure as type
---> select y as follow-up time
---> select eng3 as exposure
---> click on Effects
     ---> select Rate ratios
     ---> click on OK
```

You should see something like this:

```
                    Effect   95% Confidence Interval

Level 2 vs level 1  0.6452   [ 0.339 , 1.229 ]
Level 3 vs level 1  0.2886   [ 0.124 , 0.674 ]
```

Compared with level 1 of `eng3`, subjects at level 2 have a lower rate of CHD by a factor of 0.6452, and subjects at level 3 have a lower rate of CHD by a factor of 0.2886. The statistical test that appears below the table is for the null hypothesis that the true values of both effects of `eng3` are 1. The P-value is very small, so there is strong evidence against the null hypothesis.

5 Metric exposures

When the exposure variable is metric, such as `energy` in the `diet` data, the most common way of dealing with it is to group the values and treat the grouped variable as categorical. This is what was done with `energy` in the previous section, where the values of `energy` were grouped into `eng3`. However, it is also possible to find an effect per unit of exposure, without grouping.

We shall illustrate this by finding the effect of `gestwks` on `bweight`. This can be done provided we can assume that this effect is the same throughout the range, i.e. that the effect of changing from 30 to 31 weeks of gestation is the same as changing from 31 to 32, and so on throughout the range. If this is the case then the relationship between `bweight` and `gestwks` is *linear*. To check this assumption, we shall start by grouping `gestwks` into `gest4`, and tabulating the mean birth weight by `gest4` with

```
. use births, clear
. egen gest4=cut(gestwks), at(20(5)45)
. tabmenu1, clear
---> select bweight as response
---> select metric as type
```

```
---> select gest4 as rows
---> click on Tables
     ---> select Means
     ---> click on OK
```

You will see that birth weight tends to go up with gestational age, and that the increase in birth weight per unit increase in gestation (in weeks) is roughly constant throughout its range of values. Now we can find the effect of a unit increase in `gestwks` with

```
. effmenu1, clear
---> select bweight as response
---> select metric as type
---> select gestwks as exposure
---> check metric exposure
---> click on Effects
     ---> select Difference in means
```

The menu now shows that the exposure variable is metric, and offers a choice of effects *per something*. The default is per 1 unit, so go with this and click OK to produce the table of effects. In this case there is only one effect, namely 197.0 g per unit increase in `gestwks`, i.e. per week of gestation.

We can do the same thing to find the effect of `gestwks` on `lowbw`, using odds ratios to measure the effect. The assumption we are now making is that the odds that the baby has low birth weight is reduced by the same factor for a change in gestation from 30 to 31 weeks, 31 to 32 weeks, and so throughout the range. This is the same as saying that the relationship between the log odds and `gestwks` is linear. To tabulate the odds of being low birth weight by `gest4`, try

```
. tabmenu1, clear
---> select lowbw as response
---> select binary as type
---> select gest4 as rows
---> click on Tables
     ---> select Odds
     ---> click on OK
```

The odds are blank for the first level, 3.3333 for the next, 0.1142 for the next, and 0.0074 for the last level, so the odds are certainly going down with increasing `gestwks`. The blank refers to an odds of 4/0, or infinity, because all babies with gestation in the range 25-29 weeks have low birth weight. Now try

```
. effmenu1, clear
---> select lowbw as response
---> select binary as type
---> select gestwks as exposure
---> check metric exposure
---> click on Effects
```

```
     ---> select Odds ratios
     ---> click on OK
```

The effect of a unit increase in `gestwks` is to multiply the odds that the baby has low birth weight by 0.41, i.e. a reduction to 41% of its current level for each extra week of gestation.

We shall now return to the `diet` data and find the effect of `energy` on the rate of CHD, where `energy` is a metric exposure variable, and the effect is measured as a rate ratio per unit of energy. As before we assume that this effect is the same throughout the range, that is the effect of changing from 1700 to 1701 kcal is the same as changing from 1702 to 1703, and so on throughout the range. This is the same as saying that the relationship between the log rate and `energy` is linear. First we check this assumption by cutting energy into equally spaced groups, and looking at how the rates change from one level to the next:

```
. egen eng5=cut(energy),at(1700(500)4200)
. tabmenu1, clear
---> select chd as the response
---> select failure as type
---> select y as follow-up time
---> click on Tables
     ---> select Rates per 1000
     ---> click on OK
```

Although the rates do not go down by the same factor at each level, at least they go down, so the assumption that the rate ratio comparing each level with the previous one is constant is not unreasonable. Now try

```
. effmenu1, clear
---> select chd as the response
---> select failure as type
---> select y as follow-up time
---> select energy as exposure
---> check metric exposure
---> click on Effects
     ---> select Rate ratio
     ---> click on OK
```

The effect of a unit increase in `energy` is 0.9988 per unit of energy, i.e. the CHD rate is reduced by a factor of 0.9988 for each increase of 1 kcal in total energy. An increase of 1 kcal is a very small amount of energy, which explains why the effect is so close to 1. It would be better to measure the effect per 100 kcal, or even per 500 kcal. Go back by typing `effmenu1`, click on Effects, and then change the units for computing the rate ratio to per 100 kcal. The effect is now 0.8913 per 100 kcal.

Tuesday pm

14.30 – 16.00

Confounders and effect modifiers

1 Controlling for confounding variables

The effects calculated so far are *marginal* effects, and take no account of the possibility of confounding due to other variables. For example, in the `births` data, the marginal effect of `hyp` is −430.7 g, but the sex of the baby is associated with its birth weight, so if sex is also associated with hypertension, then part of the marginal effect could be due to differences in the sex ratio between the babies of normal and hypertensive women. To exclude this possibility we need to *control* the effect of `hyp` for `sex` by selecting `sex` as a control variable:

```
. use births, clear
. effmenu1, clear
---> select bweight as response
---> select metric as type
---> select hyp as exposure
---> click on Effects
     ---> select Difference in means
     ---> select sex as a control variable
     ---> click on OK
```

The effect of `hyp` controlled for `sex` is −448.1 g, but to understand how this is obtained we need to look at the effect of `hyp` on `bweight` separately for each sex. This is done by specifying `sex` as a modifying variable.

2 Effect modification

To specify `sex` as a modifying variable, try

```
. effmenu1
---> select sex as modifier
---> click on Effects
     ---> select Difference in means
     ---> remove sex as a control variable
     ---> click on OK
```

The result should look like this:

```
Level or value
of sex                          Effect   95% Confidence Interval

1                             -496.3513     [ -296.143 , -696.560 ]
2                             -379.7734     [ -141.606 , -617.941 ]
```

The effect of `hyp` is −496.3 g for boys and −379.8 g for girls. These two separate effects of `hyp` are really what is meant by "controlled for `sex`" because all boy babies have the same value for `sex`, and all girl babies have the same value for `sex`, so no part of these effects can be due to differences in the sex ratio.

However, because there is no evidence that the true values of these two effects differ, reporting the separate effects is unnecessary, and they are combined to give a single effect, −448.1 g. It is this combined effect that is called the effect of `hyp` controlled for `sex`. The word *control* (unfortunately) is used to cover both controlling, i.e. finding effects separately for different levels of the confounder, and combining the separate effects when they appear to be the same.

Although many people control for a potential confounder without first looking at the separate effects at different levels of the confounder, it is better to look first, because the separate effects are combined on the assumption that there is no effect modification, i.e. that their true values are the same. The statistical test for no effect modification is used to confirm this assumption, although commonsense also plays a role. When there is strong effect modification the effect of exposure should be reported separately for each level of the modifying variable.

As another example, we shall return to the `diet` data, and control the effect of `hieng` on the rate of CHD for `job`, first looking at the separate effects of `hieng` in the different jobs. Start by finding the marginal effect of `hieng` (level 2 vs level 1) on `chd` as a rate ratio, with

```
. effmenu1, clear
---> select chd as response
---> select failure as type
---> select y as follow-up time
---> select hieng as exposure
---> click on Effects
     ---> select Rate ratios
     ---> click on OK
```

The answer should be 0.5204. Now go back to `effmenu1`, select `job` as the modifying variable, click on Effects, and then OK as before. You will now see three effects of `hieng`, one for each level of `job`:

```
Level or value
of job            Effect   95% Confidence Interval

driver            0.4103   [ 0.124 , 1.362 ]
conductor         0.6551   [ 0.227 , 1.888 ]
bank              0.5177   [ 0.212 , 1.267 ]
```

All three effects are measuring the effect of `hieng`, but in subjects who have different jobs. These three effects are similar in size, so there is no evidence that `job` modifies the size of the effect of `hieng`. This is confirmed by the significance test for no effect modification which appears below the table. The chi–squared statistic is on 2 df because we are making two comparisons: 0.6551 (conductors) with 0.4103 (driver) and 0.5177 (bank) with 0.4103 (driver). The P-value is large, confirming that there is no evidence that `job` modifies the size of the effect of `hieng`.

We can now combine the three separate effects as a single effect, by removing `job` from being a modifying variable in the first menu, and selecting `job` as a

control variable in the second menu. Try this now – you should get 0.5248 for the effect of `hieng` controlled for `job`, not very different from the marginal, or uncontrolled effect of 0.5204, showing that the confounding influence of `job`, if any, is minimal.

It is also possible to control for one variable while allowing another to be a modifier. For example, try

```
. effmenu1
---> select hieng as exposure
---> select job as modifier
---> click on Effects
     ---> select month as control
     ---> select Rate ratios
     ---> click on OK
```

The output shows the effect of `hieng`, controlled for month, but separately by job. Any number of control variables can be selected with `effmenu1`, but there can be only one modifier.

3 Controlling for age

To control for true age, rather than age at entry, we need split the follow-up for each subject into parts which belong to different age groups. There are good commands in Stata for doing this, but these are not included in this course. Instead it has been done for you in the file `diet_age`, using 5 year age bands from 40 to 70 (there were no failures before 40).[3]

Start by loading this file and inspecting its contents with

```
. use diet_age, clear
. desc
. list id ageband chd y if id==34

       id    ageband        chd          y
       34         55          0   .1560575
       34         60          0          5
       34         65          1    2.55373
```

You will see that the follow-up for this subject has been split into 0.156 years in ageband 55-59, 5 years in ageband 60-64, and 2.554 years in ageband 65-69. In the first two records `chd` is coded 0, because the subject survives these agebands, but in the last record it is coded 1 because the subject fails during the last age band.

Use `tabmenu1` to find the rates per 1000 for each ageband, and also the rates by `ageband` and `hieng`. Use `effmenu1` to find the effect of `hieng` (as a rate ratio)

[3] The commands are

```
. stset dox, fail(chd) origin(dob) enter(doe) scale(365.25) id(id)
. stsplit ageband, at(40(5)70) trim
```

controlled for ageband, checking first that ageband does not modify the effect of hieng.

4 Controlling the effect of a metric exposure

The effect of a metric exposure variable can be controlled for a confounding variable, in the same way as for a categorical exposure. For example, try

```
. use diet, clear
. effmenu1
---> select energy as exposure
---> check metric exposure
---> select job as modifier
---> click on Effects
     ---> enter 100 in per unit box
     ---> select Rate ratios
     ---> click on OK
```

The three effects you will see are the effects of energy per 100 kcal at each of the levels of job. They seem similar, and this is confirmed by the test for effect modification. Combine them by moving job from being a modifier to being a control variable, to get 0.8919 as the effect of energy per 100 kcal, controlled for job.

5 Metric modifying and control variables

To complete the story we need to discuss metric modifying and control variables. For example, suppose we want to control the effect of hieng on chd for height. On the assumption that the log rate changes linearly with height, this is easily done as follows:

```
. effmenu1, clear
---> select chd as response
---> select failure as type
---> select y as follow-up time
---> select hieng as exposure
---> click on Effects
     ---> select Rate ratios
     ---> select height as control
     ---> select height as metric control
     ---> click on OK
```

The effect of hieng controlled for height (metric) is 0.6132. Of course, before doing this, you should check on the linear relationship between the log rate and height by grouping height, and making a table of rates by grouped height, as follows:

```
. egen htgrp=cut(height),at(150(10)190)
. tabmenu1, clear
---> select chd as response
---> select failure as type
---> select y as follow-up time
---> select htgrp as rows
---> click on Tables
     ---> select Rates per 1000
     ---> click on OK
```

One man has a height above 190 cm, but since we are only checking the linear assumption, he can be excluded. You will see that the rate is going down at each level, so the assumption of a linear relationship between the log rate and `height` is not unreasonable. You should also check that `height` does not modify the effect of `hieng` by specifying `height` as a modifier and producing a table of effects for different values of `height`. To do this, use `effmenu1` again, specify `height` as a modifier, and check the metric modifier box. You will see (among other things):[4]

```
Level or value
of height          Effect   95% Confidence Interval

p25                0.6409   [ 0.348 , 1.181 ]
p50                0.5673   [ 0.295 , 1.091 ]
p75                0.4916   [ 0.193 , 1.249 ]
```

What the program has done is best explained in terms of log rates. A linear relationship between the log rate and `height` is assumed for each level of `hieng` – these are the two straight lines you see in Figure 1. The 25th, 50th, and 75th percentiles of height are marked on the horizontal axis as p25, p50, and p75. Because the log of the ratio of two rates is equal to the difference between the two log-transformed rates, the log of the effect of `hieng` when height is at the 25th percentile is the vertical distance between the two lines at p25. The log of the effect when height is at the 50th percentile is the vertical distance between the two lines at p50, and similarly for height at the 75th percentile. If there is no effect modification these three effects of `hieng` at different levels of `height` will be the same, apart from random variation, i.e. the lines will be parallel. If there is substantial divergence or convergence between them then the effect of `hieng` is modified by `height`.

The `showat` box in the second menu allows you to choose the points of the metric modifier at which the effects of the exposure are calculated. For example, you might prefer to show the effects of `hieng` at heights 160, 165, 170, 175 cm, instead of at the three percentile values shown above.

[4] If you don't see this, you have probably forgotten to check the metric modifier box.

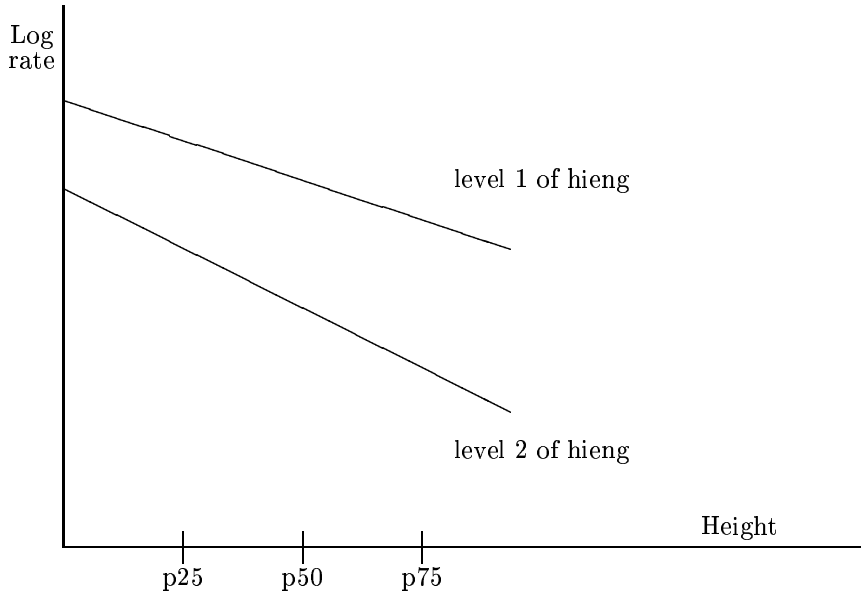

Figure 1: Displaying effects when the modifier is metric (hypothetical example)

6 Metric versus grouped

For a metric exposure it is usually best to group its values and to find the effects by comparing each level of the grouped exposure with the baseline. With small amounts of data there is some advantage in using the exposure in its metric form, and finding the effect per unit of exposure, but the problem is that with small datasets it is not easy to check the assumption of a constant effect throughout the range, unless the response is also metric.

For a metric control variable, the issue is whether to control the effects of an exposure using the control variable in its metric form, or whether to group its values and control for it in a categorical form. Very similar results are usually obtained both ways, so this is a less important decision than how to measure the effects of an exposure (metric or categorical) in the first place. In general, when you have lots of data, group the control variable; when you have only a small amount of data, treat the control variable as metric, where possible.

When checking for effect modification, there is a strong case for treating the exposure as metric, whenever the assumption of a constant effect throughout the range appears reasonable. This is because when an exposure is grouped with, say, 5 levels there will 4 effects per level of the modifier, but with a metric exposure there will only be one. Since there is usually only a small amount of data with which to estimate effects at each level of the modifier, the fewer effects to estimate the better. This advice holds for both categorical and metric modifiers, but particularly for categorical modifiers.

Wednesday am

10.45 – 12.00

Regression Commands

1 Regression commands

The table of mean `bweight` by `hyp`, using the `births` dataset, showed that hypertensive mothers have babies which are on average about 430 g smaller than those delivered to normal mothers. We shall now investigate this difference in birth weights more fully using one of Stata's estimation commands. Which one to use depends on the type of response variable, and some of the main possibilities are shown below.

Response	`effmenu1`	Regression
binary	odds ratios	`logistic`
metric	differences in means	`regress`
failure/count	rate ratios	`poisson`

The response is `bweight`, which is metric so the appropriate command to estimate the effect of `hyp` is `regress`. The explanatory variable is `hyp`, which is categorical, but because it has only two categories we can cheat and treat it as metric, which is the default for estimation commands. We show how to treat it as categorical in the next section, but for the moment, try

```
. use births, clear
. regress bweight hyp
```

Note that after the command the response `bweight` comes first, followed by the explanatory variable `hyp`. The response variable `bweight` is named at the head of the first column. The explanatory variable `hyp` is named just below. The first number in the column marked `Coef.`, -430.7, is the change in mean birth weight for a unit increase in `hyp`, i.e. from 0 to 1, or normal to hypertensive. The next row, named `_cons`, refers to the mean birth weight when `hyp` is zero, i.e. the mean birth weight for normal mothers. The last two columns show the 95% confidence intervals. To change them to 90% try

```
. regress bweight hyp, level(90)
```

or you could use

```
. set level 90
```

which changes the level to 90% until you re-set it, or close Stata.

The first half of the output, headed Source, describes the partition of the total variability in the data between that explained by the regression model and what is left (the residual). In most applications this is only of historic interest.

As an example of a genuinely metric explanatory variable we shall study how birth weight varies with gestational age. Try

```
. regress bweight gestwks
```

Ignoring the first part of the output, the row in the table that corresponds to `gestwks` shows that a unit increase in `gestkws`, anywhere on the `gestwks` scale, results in an increase of 197 g in birth weight, with confidence interval from 180 to 214 g. The `_cons` term is the result of extrapolating birth weight back to a gestational age of 0, and is not helpful.

2 Generating indicator variables

In the first example above the explanatory variable, `hyp`, was categorical with two categories, and because it had only two categories we were able to cheat and treat it as metric. To show how to deal with a categorical explanatory variable with more than two levels we shall cut `gestwks` into 4 categories with

```
. egen gest4=cut(gestwks), at(20,35,37,39,45)
. table gest4, contents(mean bweight)
```

The `table` command shows that the mean birth weight goes up from one category of gestation time to the next. To make a comparison between the mean birth weights for different categories of gestational age we have to create indicator variables for each of the levels of `gest4`. One way of doing this is with

```
. tabulate gest4, generate(ind)
```

which generates four indicator variables called `ind1` to `ind4`. To see how these are coded try

```
. sort gest4
. browse gest4 ind1-ind4
```

For subjects at the first level of `gest4` the variable `ind1` is coded 1, but the other three are coded 0. For subjects at the second level of `gest4` the variable `ind2` is coded 1, but the other three are coded 0, and so on. To set the first level of `gest4` as the baseline against which the other levels are compared, the first indicator variable is omitted from the regression. Try

```
. regress bweight ind2 ind3 ind4
```

The coefficients of the indicators are the three effects of `gest4`, using the first level as the baseline. To make the second level the baseline, omit the second indicator and try

```
. regress bweight ind1 ind3 ind4
```

It is not necessary to generate your own indicator variables. Instead the generation of indicator variables can be automated with the `xi:` command, which stands for *expand indicators*. Try

```
. xi: regress bweight i.gest4
```

The `xi:` warns Stata that there are categorical variables in the command that follows, and the `i.` states which variables are categorical. The command

```
. describe
```

shows that variables with names like `_Igest4_35` have been added to the dataset. These are the indicator variables generated by `xi`. Their names are a combination of the name of the variable identified by `i.` and the category they indicate. Since `gest4` had four categories, only three indicators are generated and included in the regression. The lowest category is the one used as baseline. You can change the baseline category with

```
. char gest4[omit] 37
```

which omits the category coded 37, i.e. makes this category the baseline. Try

```
. xi: regress bweight i.gest4
```

again, to see the effect of changing the baseline to the category coded 37. Use

```
. char gest4[omit]
```

to re–set the default baseline.

Returning to `hyp` as the explanatory variable, find the effect of `hyp` on `bweight`, treating `hyp` first as categorical, then as metric, with

```
. xi: regress bweight i.hyp
. regress bweight hyp
```

You will see that the results are the same. Leaving out the `i.` makes Stata treat the variable as metric.

3 A binary response

When the response variable is binary, like `lowbw`, the regression command `logistic` can be used to calculate effects as odds ratios. For example, and to show this in detail, the odds of the baby being low birth weight for hypertensive and normal mothers can be found first from

```
. use births, clear
. tabulate lowbw hyp
```

The odds are 40/388 for normal mothers and 20/52 for hypertensive mothers. The odds ratio is

$$\frac{20/52}{40/388} = \frac{388 \times 20}{40 \times 52} = 3.73$$

so the odds of the baby being low birth weight for hypertensive mothers are 3.73 times the odds for normal mothers. This can be obtained directly from

```
. xi: logistic lowbw i.hyp
```

To compare the odds of the baby being low birth weight between groups of gestational age, try

```
. egen gest4=cut(gestwks), at(20,35,37,39,45)
. xi: logistic lowbw i.gest4
```

The baseline level (by default) is the first, i.e. `gest4=20`. The effect of changing from the first to the second level is to decrease the odds by a factor of 0.164; from the first to the third the odds are reduced by a factor of 0.029, and from the first to the fourth by a factor of 0.003.

To use gestational age as a metric explanatory variable, try

```
. logistic lowbw gestwks
```

The odds ratio of 0.408 refers to the effect of a unit change in `gestwks`, i.e. a reduction in the odds of being low birth weight of 41% for every extra week of gestation.

4 Survival data and rates

For survival time data, when effects are measured as rate ratios, the appropriate regression command is called `poisson`. To illustrate this we need to load the `diet` data. To find the effect of `hieng` as a rate ratio try

```
. use diet, clear
. xi: poisson chd i.hieng, e(y) irr
```

The option `e(y)` states that the follow-up time (sometimes called the exposure time) is `y`, and the option `irr` asks for incidence rate ratios (i.e. rate ratios). Apart from these two options `poisson` is used in exactly the same way as `logistic`. Try

```
. egen eng3=cut(energy), at(1500,2500,3000,4500)
. tabulate eng3
. xi: poisson chd i.eng3, e(y) irr
```

The effects of `eng3` are 0.6452 and 0.2886. To find the effect of `energy` per 1 kcal, try

```
. poisson chd energy, e(y) irr
```

To find the effect of energy intake per 100 kcal, try

```
. gen energy100=energy/100
. poisson chd energy100, e(y) irr
```

5 Likelihood ratio and Wald tests

Hypotheses generally take the form that (on a log scale) one or more parameters are zero. The log likelihood when these parameters are set to zero is compared to the log likelihood when the parameters take their most likely value, using the difference in log likelihoods, which is the same as the log likelihood ratio. If the hypothesis involves m parameters then the p-value for the hypothesis is obtained by referring minus twice the log likelihood ratio to the chi-squared distribution on m degrees of freedom. Stata reports the log likelihood whenever it fits a model.

1. The command

   ```
   . xi: poisson chd i.job, e(y)
   ```

   fits a Poisson regression model to the `diet` data with a corner parameter (called `_cons` by stata) and two job parameters (on a log scale), the first comparing level 1 with level 0, the second comparing level 2 with level 0. The log likelihood for this model is -176.608.

2. To test the hypothesis that both job parameters are zero, fit the model without the job parameters

   ```
   . xi: poisson chd, e(y)
   ```

   The log likelihood is -177.411. Twice the difference between these two log likelihoods is 1.606, and the probability of exceeding this value in a chi-squared distribution with 2 degrees of freedom is 0.448, found by using

   ```
   . di chiprob(2,1.606)
   ```

3. All this can be done automatically using the `lrtest` command as follows:

   ```
   . xi: poisson chd i.job, e(y)
   . lrtest, saving(0)
   . xi: poisson chd, e(y)
   . lrtest
   ```

   The output shows the value of minus twice the log likelihood ratio, and the probability (p-value) that this is exceeded in a chi-squared distribution on 2 degrees of freedom. The p-value is high so we conclude that there is no difference between jobs as far as CHD mortality is concerned.

4. An approximate version of `lrtest`, called the Wald test, is provided by `testparm`. Try

   ```
   . xi: poisson chd i.job, e(y)
   . testparm _Ijob_2 _Ijob_3
   ```

   which tests whether the the two job parameters are zero. The parameter list can be abbreviated to `_I*`. The advantage of `testparm` is that it is only necessary to fit one model.

6 Several explanatory variables

Estimation commands can have more than one explanatory variable. As an example we shall use the `births` data to study how `bweight` varies with both `gestwks`, which is metric, and `sex`, which is categorical. Try

```
. use births, clear
. xi: regress bweight gestwks i.sex
```

The interpretation of the coefficients for `gestwks` and `sex` is as follows:

- The effect of a unit change in `gestwks` when `sex` is kept constant (i.e. when we control for `sex`) is an increase of 196 g of birth weight.
- The effect of a unit change in `sex` (from male to female) when `gestwks` is kept constant (i.e. when we control for `gestwks`) is a decrease of 190 g in birth weight.

Using `effmenu1` we could have found the effect of `gestwks` controlled for `sex` with

```
. effmenu1, clear
---> select bweight as response
---> select metric as type
---> select gestwks as exposure
---> check metric exposure
---> click on Effects
     ---> select Difference in means
     ---> select sex as control variable
     ---> click on OK
```

Similarly, the effect of `sex` controlled for `gestwks` could have been found by declaring `sex` as exposure and `gestwks` as control, as follows:

```
. effmenu1
---> select bweight as response
---> select metric as type
---> select sex as exposure
---> click on Effects
     ---> select Difference in means
     ---> select gestwks as control variable
     ---> check metric control variable
     ---> click on OK
```

While `effmenu1` treats one variable as the exposure and the other as control, and reports either the effect of `gestwks` controlled for `sex` or the effect of `sex` controlled for `gestwks`, estimation commands treat the two variables symmetrically. As another example we shall use `lowbw` as the response, and look at the effects of `gestwks` controlled for `sex` and the effects of `sex` controlled for

gestwks. We shall measure effects using odds ratios, so the appropriate estimation command is

```
. xi: logistic lowbw gestwks i.sex
```

The effect of a unit increase in gestwks, controlling for sex, is to reduce the odds of low birth weight by a factor of 0.405, and the effect of a unit increase in sex (i.e. from male to female) controlled for gestwks is to increase the odds of low birth weight by a factor of 1.510.

7 Effect modification and interactions

To see how to study effect modification using estimation commands we shall use the diet dataset. Earlier, in Section 2, we considered whether the effect of hieng is modified by job using effmenu1. Try this again with

```
. use diet, clear
. effmenu1, clear
---> select chd as response
---> select failure as type
---> select y as follow-up time
---> select hieng as exposure
---> select job as modifier
---> click on Effects
     ---> select Rate ratios
     ---> click on OK
```

You should see the following table:

```
Level
of job       Effect   95% Confidence Interval

driver       0.4103   [ 0.124 , 1.362 ]
conductor    0.6551   [ 0.227 , 1.888 ]
bank         0.5177   [ 0.212 , 1.267 ]
```

There are three effects of hieng, one for each level of job. To produce the interactions using poisson, try

```
. xi: poisson chd i.hieng*i.job, e(y) irr
```

The * between i.hieng and i.job tells Stata that interactions between these two variables are required. You should see the following table (which has been abbreviated):

```
-------------+------------
         chd |        IRR
```

```
-------------+------------
   _Ihieng_1 |   .4102648
     _Ijob_2 |   1.136857
     _Ijob_3 |    .813427
_IhieXjob_~2 |   1.596755
_IhieXjob_~3 |   1.261973
-------------+------------
```

The first row of the table shows the effect of `hieng` at the first level of `job`. The fourth and fifth rows show the two interactions, identified as `_IhieXjob_~2` and `_IhieXjob_~3`. Although Stata goes to a lot of trouble to name the interactions appropriately, the only important thing is to recognize that they are interactions between `hieng` and `job`. The terms in the second and third rows of the `poisson` results refer to the two effects of `job` when `hieng` is at its first level.

We would have obtained exactly the same results by reversing the order of `job` and `hieng`, like this:

```
. xi: poisson chd i.job*i.hieng, e(y) irr
```

When the three effects of `hieng` at the different levels of `job` are the same (i.e. there is no effect modification) the interactions will both be 1, apart from random variation. The test for no effect modification is therefore the same as the test that the interaction parameters are 1. Try

```
. testparm _IjobXhie*
```

to carry out this test. You will see that the chi-squared statistic on 2 df is 0.33, and the P-value is 0.8475, so there is no evidence of effect modification.

8 Control variables

The interactions between `hieng` and `job` did not differ significantly from 1, so we can fit the model without interactions using

```
. xi: poisson chd i.hieng i.job, e(y) irr
```

You should see the following (abbreviated) table:

```
-------------+-----------
         chd |        IRR
-------------+-----------
   _Ihieng_2 |   .5247666
     _Ijob_2 |   1.358442
     _Ijob_3 |   .8843023
-------------+-----------
```

The effect of `hieng` controlled for `job` is 0.525. The other two terms are the effects of `job` controlled for `hieng`.

The `lrtest` command can be used to obtain the likelihood ratio test for the hypothesis that the effect of high energy controlled for job is zero, using the commands

```
. xi: poisson chd hieng i.job, e(y)
. lrtest, saving(0)
. xi: poisson chd i.job, e(y)
. lrtest
```

Or alternatively

```
. xi: poisson chd hieng i.job, e(y)
. testparm hieng
```

carries out the Wald test.

Wednesday pm

14.30 – 16.00

Random effects models

1 The PNG data

The Papua New Guinea data refer to a trial concerned with the role of a vaccine in reducing the rate of occurrence of recurring infections in young children.

The best way to code recurrent event data of this kind is as two files. In the first file, which we shall call the *subject file* there is a single record for each subject, containing the data which remain unchanged throughout the follow–up. The second file, the *event file*, will contain one record for each recorded event, including any censoring event (termination of follow–up). Apart from there being more than one event per subject, the data in the event file are exactly the same as for any survival data: each record contains the time of entry, time of exit, and the event type, for that that record, and a subject identifier. For simplicity the time for which the infectious episodes last is ignored; in real life it would be necessary to subtract this time from the time at risk.

Have look at the subject file `pngsubj` in which each record refers to a subject, and the event file `pngevent` in which each record refers to a part of the follow-up for a subject, with

```
. use pngsubj, clear
. desc
. list in 1/10
. use pngevent
. desc
. list in 1/10
```

The outcome variable is called `diag` and is coded according to the type of infection. Codes 80 and 81 refer to acute lower respiratory tract (ALR) infections. We shall be interested in ALR infections, because the vaccine was specifically for pneumonia, but also in all infections, as a measure of general health.

Start by loading the `png` subject file and merging with the event file, matching on `id` with

```
. use pngsubj, clear
. merge id using pngevent
. tab _merge
. drop _merge
. sort id timein
. list id vacc timein timeout diag if id==2921
```

The last two commands show how the vaccination status (for example), which comes from the subject file, has been merged with the information about events. The system variable `_merge` is used to check the results of the merge.

Now create failure variables for ALL infections, ALR infections, plus the follow-up time:

```
. gen dalli=0 if diag != .
. replace dalli=1 if diag>0 & diag<.
```

```
. gen dalri=0 if diag != .
. replace dalri=1 if diag==80 | diag==81
. gen y=(timeout-timein)/365.25
```

It is necessary to exclude missing values of `diag` in the second line because missing is treated as an infinitely large number, and is therefore > 0. Finally aggrgate the data with[5]

```
. collapse (sum) dalli dalri y, by(id vacc)
. list id vacc dalli dalri y if id==2921
```

In the aggregated form the data record the total number of infectious episodes and the total follow–up time. The variable `dalli` refers to all infections, `dalri` refers to acute lower respiratory infections only, and `y` is the total follow-up time.

1. Find the effect of vaccination on all infections in a random effects model using

   ```
   . nbreg dalli vacc, e(y) irr
   ```

2. To carry out a test for the effect of vaccination you can use either a Likelihood Ratio test

   ```
   . nbreg dalli vacc, e(y) irr
   . lrtest, saving(0)
   . nbreg dalli, e(y) irr
   . lrtest
   ```

   or a Wald test

   ```
   . nbreg dalli vacc, e(y) irr
   . testparm vacc
   ```

3. Compare the estimated effect of vaccination, with its SE, with what you get from poisson regression, ignoring heterogeneity.

   ```
   . poisson dalli vacc, e(y) irr
   ```

4. Repeat the analysis for ALRI infections.

2 The epilepsy data

A well known example in the literature concerns a trial of an active drug against placebo (`trt` = 1 and 0 respectively) in the treatment of epilepsy. Epileptic seizures were counted in four two-week periods following treatment. A baseline count of seizures in the eight–week period prior to treatment was also recorded.

The file `epilep1` refers to the aggregated data – `d` is the total number of seizures for each subject, and `y` is set to 8 weeks for all subjects.

[5] These commands for aggregating the data are stored in the file `png_agg.do`.

1. Load the `epilep1` data and have a look at it.
2. Find the effect of treatment (`trt`) in a random effects model
3. Find the effect of treatment (`trt`) controlled for the log of baseline reading.
4. Why has the variance of the frailties gone down compared to the model without the basline?

Thursday am

10.45 – 12.00

Quasi-likelihood and robust SE's

1 Quasi-likelihood

1. Load the PNG data with

   ```
   . use png_agg, clear
   ```

2. Find the effect of `vacc` on all infections using quasi-likelihood with a negative binomial variance function in which $\kappa = 1$

   ```
   . glm dalli vacc, fam(nbinomial 1) link(log) lnoff(y)
   ```

3. Try some other values of κ until the Pearson chi-squared is approximately equal to its degrees of freedom.

2 Robust SE's

1. The effect of `vacc`, ignoring heterogeneity, can be found with

   ```
   . glm dalli vacc, fam(poisson) link(log) lnoff(y)
   ```

   The Pearson chi-squared is about twice its degrees of freedom, indicating appreciable subject heterogeneity.

2. One way of taking account of subject heterogeneity is to use the estimates from Poisson regression, but with robust standard errors.

   ```
   . glm dalli vacc, fam(poisson) link(log) lnoff(y) robust
   ```

   Note that the standard error (log scale) has increased by about 50% compared to the standard error without the robust option.

3 The epilepsy data

1. First load the data and look at the coding with

   ```
   . use epilep1, clear
   . desc
   ```

2. Carry out a Poisson regression to estimate the effect of treatment with

   ```
   . glm d trt, fam(pois) link(log) lnoff(y)
   ```

   The Pearson chi-squared is 64 times its df, indicating massive subject heterogeneity.

3. Including the log of the baseline count (`lbase`) for the subject in the model reduces the heterogeneity considerably

```
. glm d trt lbase, fam(pois) link(log) lnoff(y)
```

The The Pearson chi-squared is now 11 times its df,, still large, but much reduced from 64. It makes sense to include the log baseline count in the model, because the response being compared between treatments is then the percentage change from baseline.

4. The standard error of the effect of `trt` from the `glm` command ignores the heterogeneity, and will therefore be too small. To take account of heterogeneity use robust standard errors using

```
. glm d trt, fam(pois) link(log) lnoff(y) robust
```

4 The DPS data

These data refer to a study of the prevalence of depression. The aggregated data are in the dps_agg, where d is the number of occasions out of a maximum of 7 when the subject was depressed.

1. Load the data and look at it with

```
. use dps_agg, clear
. desc
```

Stata does not include an overdispersed variance function for the binomial family, but we can use `glm` with the binomial family (n=7) and robust SE's to find the effect of `sex`.

Thursday pm

14.30 – 16.00

Longitudinal (panel) studies

1 Cross-sectional time series

Data in which several measurements are made on each subject, at different time points, are often called *panel* data, from the panel of subjects who provide the data. An alternative name is cross-sectional time-series data. The commands in stata for analysing such data are the `xt` commands where `xt` is short for cross-sectional time series.

2 The PNG data – subdividing the follow–up into age bands

For the PNG data, the follow–up is from age 0 until about 8 and the rate of infections varies markedly with age.

To calculate the age–specific infection rates per year we must first break the follow–up into age bands and then aggregate the events by ageband. The commands for doing this are in the file `png_age`[6], which you can run with the command

```
. do png_age
. desc
```

You will see that a new variable `ageband` has been created. Now try

```
. table ageband vacc, c(sum dalli sum dalri sum y)
```

to see the total events and person years by `ageband` and `y`. You can get the rates of alli and alri by `ageband` and `vacc` using `tabmenu1`.

3 Random effect models

In a random effects model each subject has a frailty which remains constant from one age band to the next. The model can be fitted with `xtpois` which, unlike `nbreg`, can deal with data in which records cluster by subject:

1. `. xi: xtpois d vacc i.ageband, e(y) i(id)`

   The `i(id)` indicates that records cluster by subject.

2. To test the effect of vaccination controlled for ageband using a Wald test, try

   ```
   . testparm vacc
   ```

   The effect of vaccination controlled for `ageband` is not significant.

[6]This file inlcudes `st` commands which are not part of this course

4 Poisson regression with robust SE's

1. The Poisson regression command to look at the effect of vaccination controlled for `ageband` is

   ```
   . xi: poisson dalli vacc i.ageband, e(y)
   ```

2. To obtain robust estimates of standard error it is necessary to specify that the records are clustered in groups coming from the same subject.

   ```
   . xi: poisson dalli vacc i.ageband, e(y) clus(id)
   ```

   It is not necessary to specify `robust` with `clus` as this is automatically assumed.

3. To test the effect of vaccination controlled for ageband using a Wald test, try

   ```
   . testparm vacc
   ```

   The effect of vaccination controlled for ageband is not significant.

5 Generalized estimating equations

Although the estimated parameters from the Poisson model are unbiased under the more realistic model in which there is subject heterogeneity, and therefore correlation between the records from the same subject, they may be rather inefficient. Generalized estimating equations offer the opportunity to take account of the correlations between records in the estimating equation.

In Stata GEE is implemented in the command `xtgee`. This is one of a group of commands for longitudinal data with the common convention that the subject identifier is in the option `i()` and the time variable is in `t()`.

1. When the independence correlation structure is specified with `xtgee` we get the same results as from `poisson` with `clus(id)`.

   ```
   . xi: poisson d vacc i.ageband, e(y) clus(id)
   . xi: xtgee d vacc i.ageband, e(y) i(id) fam(pois) corr(ind) robust
   ```

2. A more common choice for the correlation structure is `corr(exc)`, which stands for exchangeable. This correlation structure is implied by the model with constant frailty within a subject. Try this option

   ```
   . xi: xtgee d vacc i.ageband, i(id) fam(pois) robust corr(exc) e(y)
   ```

   Examine the estimated correlations by:

   ```
   . xtcorr
   ```

3. To test the effect of vaccination controlled for ageband using GEE with exchangeable correlation structure, try

```
. xi: xtgee d vacc i.ageband, e(y) i(id) fam(pois) corr(exch) robust
. testparm vacc
```

   The effect of vaccination controlled for ageband is not significant.

4. The remaining correlation structures require us to specify a variable which indicates to which time point a record refers. In our case this is done by adding the option `t(ageband)`. Try the `uns` (unstructured) option and examine the estimated correlations with `xtcorr`. Does the choice of correlation structure matter?

6 Use of the robust option for model criticism

When the robust standard errors are very different from the model based standard errors it suggests that the model is a poor fit, and that the estimated effects from the model may be inefficient. This will be ilustrated with the epilepsy data in its original form where epileptic seizures were counted in four two–week periods following treatment. These data are in `epilep2`.

1. First read in the data and study the coding with

```
. use epilep2, clear
. desc
. help epilep2
```

2. Now fit a standard Poisson regression, then one with robust standard errors.

```
. use epilep2, clear
. xi: poisson d trt lbase i.period, e(y)
. xi: poisson d trt lbase i.period, e(y) clus(id)
```

   The robust standard error of the `trt` effect is much larger than the model–based standard error using the Poisson model. This suggests that the estimated effect of vaccination from the Poisson model will be inefficient.

3. A better approach is to fit a random effects model with

```
. xi: xtpois d trt lbase i.period, e(y) i(id)
```

   Notice that the estimated effect of vaccination is quite a lot higher in this model than it was in the Poisson model.

4. Alternatively we can use a generalized estimating equation with the *exchangeable* correlation structure.

```
. xi: xtgee d trt lbase i.period, e(y) i(id) fam(pois) corr(exc)
```

Adding a robust option now changes the SE's, suggesting that this model is also a poor fit.

```
. xi: xtgee d trt lbase i.period,
            e(y) i(id) fam(pois) corr(exc) robust
```

5. Now try the negative binomial variance function in `xtgee` and show that the fit is much better.

7 The DPS data

1. Load the data with

   ```
   . use dps, clear
   . desc
   ```

   The variable `wave` refers to the year in which prevalence of depression was assessed. The variable `sex` is constant over time, but others, like `care` change with time.

2. Find the effect of `care` using a random effects model. Is it the same for both sexes?

3. Repeat using GEE.

Friday am

10.45 – 12.00

The bootstrap

Data missing by design

1 The bootstrap

1. Load the haemoglobin data with

```
. use haem, clear
. desc
```

   The variable `hb` contains the haemoglobin value .

2. Try

```
. summarize hb
. return list
```

   The return list shows which results are stored after the command `summarize`, and where they are stored. For example

```
. display r(N)
. display r(mean)
. display r(sd)
```

3. To bootstrap the mean haemoglobin try

```
. set seed 36758971
. bs "summarize hb" "r(mean)", reps(1000) dots
```

   The command which produces the thing to be bootstrapped is in the first set of quotes. The thing itself is in the second. In the options we have asked for 1000 bootstrap samples and for some dots to be displayed while it is working.

4. The command

```
. summarize hb, detail
. return list
```

   shows where the median is kept. Find the bootstrap SE of the median haemoglobin.

5. Load the epilep1 data and try

```
. use epilep1, clear
. poisson d trt, e(y)
. display _b[trt]
```

   You will see that the effect of `trt` is stored in `_b[trt]`. Use this information to find the bootstrap SE of the effect of `trt` (follow question 3 but with `poisson d trt, e(y)` in place of `summarize, d` and `_b[trt]` in place of `r(mean)`. How does it compare with the Poisson SE?

2 Case–cohort studies

In this practical we shall be using weights, and since `tabmneu1` does not yet allow weights we shall use `table` instead.

1. Start by reminding yourself of the rates for `heing` in the `diet` data with

   ```
   . use diet, clear
   . generate rate= chd/y*1000
   . table hieng [iw=y], c(mean rate)
   ```

2. Now we shall artificially create a case–cohort study from the cohort study of diet and coronary heart disease by drawing a 25% sample of the cohort and then adding all the cases of CHD.

   ```
   . use diet, clear
   . gen u = uniform()
   . keep if u <= 0.25 | chd==1
   . tab chd
   ```

3. The rates by `hieng` for this case-cohort study, found using

   ```
   . generate rate= chd/y*1000
   . table hieng [iw=y], c(mean rate)
   ```

   are quite different from those in the original cohort study because we have kept all the cases but only a 25% sample of the rest.

4. The probability of a subject being selected is 1 for a case and 0.25 for a non–case, so the inverse probability weights are either 1 or 4. These can be created with the `cond` command

   ```
   . gen invpw = cond(chd==1, 1, 4)
   ```

   This asks that `invpw` be set to 1 if the condition `chd==1` is true, and to 4 otherwise.

5. We can now analyse the case–cohort study exactly as if it were a full cohort study, by using these weights. For example, to look at the variation of rates (per 1000 person–years) with `hieng`, using `table`, try

   ```
   . gen w=invpw*y
   . table hieng [iw=w], c(mean rate)
   ```

   Now the rates by `hieng` are much closer to those from the original cohort.

6. The rate ratio can be found with

   ```
   . poisson chd hieng [iw=invpw], e(y) robust
   ```

Note that weighting by `y` as well `invpw` was only necessary with `table`, but the `robust` option is needed with `poisson` to ensure that the standard errors take account of the weighting.

7. We shall need to return to these data, and re–execution of the above commands would create a *different* case–cohort study, so we shall save it with

```
. label data ''Diet and CHD case-cohort study''
. save cscht, replace
```

8. More complex analyses can be carried out equally easily. For example, to control for `job`

```
. xi: poisson hieng i.job, e(y)
. poisson, irr
```

9. To control for age is a little more complicated as the follow-up needs to split between age bands first, as in the file `diet_age`. This has to be done for the case-cohort study you have generated by making a note of which subjects are in the case-cohort study, and then dropping the rest from the file `diet_age`.

```
. keep id invpw
. sort id
. save cscht_id, replace
. use diet, clear
. sort id
. merge id using cscht_id
. keep if _merge==3
. drop _merge
```

   The last line keeps those records where a match was found in both of the merged files. Now the effect of `hieng` can be controlled for `ageband` with

```
. xi: poisson chd i.hieng i.ageband [iw=invpw], e(y)
. poisson, irr
```

10. Before leaving this example we should note that we have used *theoretical weights* based upon the fact that we know that the sampling probability for a non–case is 0.25. Alternatively we could have used *empirical weights* obtained by dividing the total number of non–cases in the entire cohort (291) by the number of non–cases in our case–cohort study. These two approaches give the same *expected* answers, although the precise results may differ for small studies. The consensus seems to be that it is better to use weights based on what actually happened (i.e. empirical weights) rather on what was expected to happen.

3 A 2–phase case–control study

In a 2–phase case–control study the first phase is a case–control study with complete data on the cases but incomplete data on the controls. In the second phase a sample of those controls already selected is taken, and complete data are obtained for this sample. As before, the cases and controls with complete data are analysed, but using inverse probability weights obtained from the first stage.

The study we shall use to demonstrate this is not strictly a 2–phase case–control study, although its analysis is exactly the same. It is a case control study of perinatal mortality in which, for logit reasons, controls (live births) were matched to cases (perinatal deaths) with respect to place and date of delivery. The reasons for matching are not the usual ones and for most analyses the conventional matched case–control methods, which would estimate the effects of risk factors *controlled for date and hospital*, are not what are required. Instead we might wish to un-do the matching and to estimate the effects of risk factors in the entire population of births from which the cases and controls were drawn. Since statistics of total births by date and place of delivery are available from routine sources, we can use these to calculate inverse probability weights. The case–control study is then analysed using these weights in exactly the same way as if the weights had been obtained form the first of two phases in a two–phase case–control study.

1. The data for the case–control study are stored in the file `pnleics` and the data for the inverse probability weights in the file `pnbirths`. You can use the help system to find out more about these files. To read in and merge the data from the two files,

   ```
   . use pnleics, clear
   . sort year stratum
   . merge year stratum using pnbirths
   ```

   The inverse probability weights take the value 1 for cases and, for controls, the ratio of the total number of live births in each sampling stratum to the number of controls:

   ```
   . gen invpw = cond(d==1, 1, births/cntrls)
   ```

2. Because of the sampling design, which matched on place of delivery, simple case/control ratios do not reflect the gradient of risk with place of delivery. Inverse probability weighting reveals the true picture:

   ```
   . xi: logit d i.deliv, or
   . xi: logit d i.deliv [pw = invpw], or
   ```

   Hospitals coded 1 and 2 are the teaching hospitals, They have the highest perinatal mortality rates because they deal with the high risk births.

3. It is advisable to use the inverse probability weighting to investigate the effect of all variables, even those not used in the matching, because these

may be associated with the matching variables. For example, the distribution of birth parity varies between places of delivery

```
. tab parity deliv, col
```

and so inverse probability weighting is necessary to get an idea of the effect of parity in the population of births

```
. xi: logit d i.parity [pw = invpw], or
. testparm _Ipar*
```

Robust SE's are given automatically, and the second command carries out a Wald test of significance of the parity effect:

4. To get results on a log scale use the `logit` command

   ```
   . xi: logit d i.parity [pw = invpw], or
   ```

5. The variable `asian` identifies mothers whose ethnicity is Asian (Indian subcontinent). Estimate the risk associated with this variable and investigate whether it is explained by the confounding effects of other risk factors (parity, social class etc.).

4 A 2–phase prevalence study

We demonstrate the analysis of these studies using a study of mental health status carried out in Cantabria, Northern Spain. The screening measurements were (a) reporting by GP, and (b) a short questionnaire (GHQ). A subsample of subjects where interviewed by a senior psychiatrist using a standard schedule (SCAN) to estabish the definitive diagnosis of mental state. Of those classified as positive either by the GP or by the GHQ, a rather large proportion were selected for the second phase of the study, while a much smaller proportion of the remaining subjects were investigated further.

The data file `wcantab1` contains the data for those subjects with complete records, including the definitive diagnosis (SCAN). The file also contains, for each combination of GP and GHQ, the number of cases in the full study, `N`, and the number of complete cases, `n`. The inverse probability weights are simply `N/n`.

1. Load the file `cantab`, sort on `gp` and `ghq`, and then merge it with the weights in `wcantab1`, matching on `gp` and `ghq`.

2. Use `table` to tabulate the prevalence positives on the definitive (SCAN) diagnosis, according to the sex of the subject, using inverse probability weights.

3. Regression models for the (log) prevalence odds may be fitted using standard logistic regression, with inverse probability weights and robust standard errors for the regression coefficients. Estimate the odds ratio for sex with a robust standard error.

4. Finally, we should note that although there was no intention to sample differentially from different subgroups, other than by screen positive or negative, inevitably different sampling fractions were achieved in practice. The file `wcantab2` reworks the inverse probability weight calculations using the finer cross–classification of `gp` by `ghq` by `sex`. We have the option of using these probability weights rather than those in `wcantab1`, which are calculated in full knowledge that there was no systematic dependence of sampling upon sex. In fact theoretical studies have indicated that it is better to use empirically calculated weights. Repeat the previous analyses using empirical weights and compare your results.

Bibliography

Good general reference books are those by Clayton and Hills (1993)[4], Breslow and Day, volumes I [1] and II[2], and Collett (1994)[5]. The book by Hills and De Stavola (2002)[7] provides a short introduction to Stata. An attempt is made below to suggest possible further reading for each of the sections in the course.

1. Introduction to Stata: Hills and De Stavola (2002)[7]
2. Likelihood: Clayton and Hills (1993)[4], chapters 1–3 and 8–12.
3. Rates: Clayton and Hills (1993)[4], chapters 4–6 and 13–15.
4. Regression models: Clayton and Hills (1993)[4], chapters 22–26.
5. Multiple events and random effects: Clayton (1994)[3].
6. Robust standard errors: Stata manual.

References

[1] N.B. Breslow and N.B. Day. *Statistical Methods in Cancer Research. Vol. I – The analysis of case-control studies.* Number 32 in IARC Scientific Publications. International Agency for Research on Cancer, Lyon, France, 1980.

[2] N.B. Breslow and N.B. Day. *Statistical Methods in Cancer Research. Vol. II – The design and analysis of cohort studies.* Number 82 in IARC Scientific Publications. International Agency for Research on Cancer, Lyon, France, 1987.

[3] D. Clayton. Some approaches to the analysis of recurrent event data. *Statistical Methods in Medical Research*, 3:244–262, 1994.

[4] D.G. Clayton and M. Hills. *Statistical Models in Epidemiology.* Oxford University Press, Oxford, 1993.

[5] D. Collett. *Modelling survival data in medical research.* Chapman and Hall, London, 1994.

[6] W.A. Guy. *Journal of the Royal Statistical Society*, 6:197–211, 1843.

[7] M. Hills and B.L. De Stavola. *A short Introduction to Stata for Biostatistics.* Timberlake Consultants, www.timberlake.co.uk, 2002.

[8] J.N. Morris, J.W. Marr, and D.G. Clayton. Diet and heart: a postscript. *British Medical Journal*, 19 November(2):1307–14, 1977.

[9] K.J. Rothman. *Modern Epidemiology.* Little, Brown, and Company, Boston, 1986.