



Economic and Social Data Service

STATA for the LFS

ESDS Government

Author: Dr Y.Li
Version: 4.0
Date: 13 October 2003

Acknowledgement

I am grateful to Professor Angela Dale for advice, to Dr Joanne Lindley for preparing the data, and to Vanessa Higgins for descriptions of the Labour Force Survey (LFS). This manual is not sponsored or approved by Stata. I am alone responsible for the errors and mistakes. Many thanks.

Stata software is developed and distributed by
Stata Corporation, 4905, Lakeway Drive, College Station, Texas 77845, USA.
Tel. +1-979-696-4601.
Email: stata@stata.com

Some useful websites:

Stata website: <http://www.stata.com>
Stata journal: <http://www.stata.com/support/faqs/res/sj.html>
Stata library: <http://www.ats.ucla.edu/stat/stata/library/>
Statalist archive: <http://www.hsph.harvard.edu/cgi-bin/lwgate/STATALIST/archives/>
Stata FAQs: <http://www.stata.com/support/faqs/>
Stata statistics FAQs: <http://www.stata.com/support/faqs/stat/>
Stata listserv: <http://www.stata.com/support/statalist/>
Stata discussion list: statalist@hsphsun2.harvard.edu

CONTENTS

| | |
|--|----|
| Chapter 1 Overview of the LFS and Stata | 4 |
| 1.1 Brief introduction to LFS..... | 4 |
| 1.2 Basic features of Stata..... | 6 |
| 1.2.1 <u>Setting</u> memory..... | 6 |
| 1.2.2 Getting <u>help</u> with known commands and <u>search</u> themes in Stata..... | 6 |
| 1.3 Starting and exiting Stata..... | 7 |
| 1.3.1 Opening a file: <u>use</u> data and systax files..... | 8 |
| 1.3.2 <u>Logging</u> the results..... | 10 |
| 1.3.3 <u>Exiting</u> Stata and <u>saving</u> data..... | 10 |
| 1.3.4 Reserved names in Stata and <u>Stata operators</u> | 11 |
| 1.4 Interactive use of Command window..... | 12 |
| 1.4.1 Using Stata as a calculator: <u>displaying</u> data..... | 12 |
| 1.4.2 Using Stata Command Window for <u>immediate commands</u> | 13 |
| 1.5 Exploring data..... | 14 |
| 1.5.1 <u>browsing</u> and <u>listing</u> data..... | 14 |
| 1.5.2 <u>Editing</u> data..... | 16 |
| 1.5.3 <u>Codebook</u> | 17 |
| 1.5.4 <u>Describing</u> data..... | 17 |
| 1.5.5 <u>Looking for</u> information in variable and value labels..... | 19 |
| 1.5.6 <u>Counting</u> data and checking data consistency..... | 20 |
| 1.6 <u>do files</u> : an example..... | 20 |
| Exercises and suggested answers..... | 22 |
| Chapter 2 Exploratory analysis with Stata | 24 |
| 2.1 The Stata command structure..... | 24 |
| 2.2 <u>Summarising</u> and <u>tabulating</u> data..... | 25 |
| 2.2.1 <u>Summarising</u> data..... | 25 |
| 2.2.2 <u>Tabulating</u> data..... | 25 |
| 2.3 <u>Sorting</u> data and the <u>by</u> prefix..... | 33 |
| 2.4 Missing values..... | 35 |
| 2.5 Creating new variables..... | 39 |
| 2.6 Recoding, labelling, dropping, keeping and renaming..... | 42 |
| Exercises and suggested answers..... | 44 |
| Chapter 3 Stata graphs | 46 |
| 3.1 Histogram..... | 46 |
| 3.2 Two-way graph..... | 47 |
| 3.3 Box..... | 49 |
| 3.4 Bar chart..... | 50 |
| 3.5 Pie chart..... | 50 |
| Exercises and suggested answers..... | 52 |
| Chapter 4 Statistical modelling: a brief introduction | 53 |
| 4.1 Statistical models: a typology..... | 53 |
| 4.1 An example of multiple linear regression..... | 53 |
| Appendix 1 Stata syntax used in the text | 58 |
| Appendix 2 Data entry into Stata | 63 |
| Appendix 2.1 Entry the data directly in Stata..... | 63 |
| Appendix 2.2 Import files of other formats into Stata..... | 64 |
| Appendix 2.3 Using StatTransfer..... | 65 |

Chapter 1 Overview of the LFS and Stata

1.1 Brief introduction to LFS

This is an introductory course to Stata using the latest Labour Force Survey (LFS) of 2002. The data set contains data from all four quarters of the 2002/3 LFS for respondents aged 16-65 and resident in the UK (n=65,943). This is only a subset of the original data files. The data set includes 60 variables. Most of the variables included within the data set are individual variables. However, there are two household variables: 'ten96' and 'house'. These, and many other key variables in the data set, form the basis of this course.

The LFS is, other than the census, the only comprehensive source of information about all aspects of the labour market. It assists many government departments in the framing and monitoring of social and economic policy. Table 1 shows major developments in the LFS over the last 30 years.

Table 1: Major developments in the LFS

| Years | Description | Other information |
|---------------|---------------------------------------|---|
| 1973-1983 | Biennial survey | Achieved sample: c.85,000 and 3,500 households in GB and Northern Ireland respectively. Sample design: cross-sectional survey (no panel element) with one responsible adult in each household answering questions on behalf of other household members. |
| 1984 – 1991 | Annual survey | Consisted of two elements: 1) a quarterly survey conducted in Great Britain throughout the year, in which each sampled address is called on five times at quarterly intervals, and which yields about 15,000 responding households in every quarter; 2) a 'boost' survey in the quarter March to May, which produces interviews at over 44,000 households in Great Britain and over 4,000 households in Northern Ireland. However, only the data from the spring quarter and the boost survey were included in the annual datasets for public release. Includes all individuals age 16+ in the household take part. ILO definition of employment introduced. |
| 1992 (Spring) | Quarterly survey | Sample size: c.60,000 and 5,000 households in GB and Northern Ireland respectively. Sample design: unclustered sample of addresses introduced. Overlapping panel design introduced with 5 waves throughout the year and c.12,000 responding households per quarter. Inclusion of those living in NHS accommodation and those resident in student halls. All individuals age 16+ in the household take part. |
| 2000 (Spring) | Local LFS (annual enhancement to LFS) | All LEAs target a minimum sample size of 875 (smaller in Rutland and LFS2002 boroughs). This is combined with annual LFS survey data. |

Since 1992, the LFS has had a simple, stratified random sample (unclustered) drawn from the Postcode Address File (PAF). Each sampled address is interviewed for five waves at 3 monthly-intervals (the first interview is face-to-face and subsequent interviews are by telephone). Interviewers can accept proxy information for household members who are unavailable when the interview takes place. Further information about the methodology of the LFS can be found in the LFS User Guide on the ESDS website.

The LFS questionnaire comprises a set of core questions that are included in every survey and cover household, family structure, basic housing information and demographic details of individuals in the households. Some questions in the core are only asked at the first interview e.g. sex, ethnic group. The survey also asks non-core questions which change from quarter to quarter. These non-core questions provide information which is only needed once or twice a year.

The survey asks very detailed questions on the labour market and employment. Information is available for people living in NHS accommodation as well as those living in private households. A sampling frame for NHS accommodation was specifically developed for the LFS. Information is available for young people aged 16 to 24 years because the LFS sample includes people living away from the parental home in a student hall of residence or a similar institution during term time. Interviews are conducted North of the Caledonian Canal.

Measurement is available over time: annually from 1984 but biennial before then (1973-1983). However, the biennial survey did not use ILO definition of unemployment. In 1992, the survey design changed considerably and it is advisable to use LFS datasets from 1992 onwards only when measuring over time.

Longitudinal datasets are available which link the quarters e.g. June 2001 to August 2002. The survey can be used for the analyses of ethnic minorities and other small samples. In order to obtain adequate sample sizes it is necessary to combine a number of years of data together. A grossing factor is available on the dataset for population estimates to be produced. The survey has small sampling errors for main population sub-groups because of the large sample size and stratified random sample with no clustering. The sample design also allows representative results to be published for any thirteen-week period.

With regard to disadvantages, the LFS has a high proportion of proxy interviews (c.30%) as compared with other surveys such as the General Household Surveys (c.5%). Proxy interviews are carried out with another member of the household if the respondent is unavailable. Like most other large-scale government surveys, the LFS excludes people living in communal establishments (except for those in NHS accommodation and students in halls of residence). There are some discontinuities over time, mainly prior to 1984 when the survey was biennial and in 1992 when the sample design changed. As with most government surveys the response rates have dropped in recent years. In 1999/2000 the response rate for the LFS was 63%.

The LFS has high research potential for secondary analysis of employment and the labour market due to its large sample size and detailed questions. The Employment & Labour Market Introductory Guide on the ESDS Government website gives recent examples of publications/articles resulting from secondary analyses of the employment and labour market questions.

1.2 Basic features of Stata

We shall learn some basic techniques in this course such as data exploration and data analysis. The package used is Intercooled Stata 7. Stata is a statistical package for managing, analysing and graphing data. Stata is very fast, partly because it keeps the data in memory. A dataset is copied from disk into memory where it is worked on, analysed, changes made and then, if necessary, saved back on to disk.

1.2.1 Setting memory

Having the data in memory means that the dataset size is limited by the amount of memory available. When Stata is started, the default memory size is set at about one megabyte. It is possible, and indeed necessary on most occasions, to change the default memory size. For example, the data used have around 2.5Mb but we can set memory to 20 Mb as we have done it. (Full syntax for this course is attached in Appendix 1.) Stata has a facility to use virtual memory, although using virtual memory slows down performance. Since Stata works with data in memory, it is really fast when there is sufficient memory but can be rather slow if the memory available is scarcely or just enough. Experienced users have suggested that, as a rule of thumb, it is good practice to set at least 20% more memory than required by the size of the dataset.

A memory size of 16 Mb will be enough for most purposes. We will set the memory to 20 Mb, which will be sufficient for our purposes in the course. The memory size can be set using the command

```
set memory 20m
```

1.2.2 Getting help with known commands and search themes in Stata

The Stata web site at <http://www.stata.com> provides user support in the form of

- Frequently Asked Questions (FAQ).
- Additions to Stata in the form of new commands. Stata is programmable. Most of Stata is implemented in Stata, meaning that most of the statistical functions and operational facilities are installed in Stata in the form of **ado files** which come with the package or can be updated from time to time from the Stata web site whilst some specialist commands (such as `coranal`, `gllamm` or `glcurve7`) can be installed from the Stata Technical Bulletins. This means that even if you never write a Stata program, you make use of that feature. The web site is an important source of new commands supplied in the form of **ado files** (automatic do files).
- Details of Stata Technical Bulletins (STB).
- Instructions for joining the very active Stata discussion list where you receive about 20 messages every day. The discussions cover a wide range of topics. Contributions come from Stata staff, statisticians, experienced Stata users or beginners. Some of the things in this course material are obtained from the discussions.
- And many other things.

Stata has a command line user interface. It is easy to save the commands you have just used in a **do file**, edit them if necessary, and then run them again. Stata also allows you to browse and edit data in a spreadsheet-like window. Results are displayed in a Results window and can also be saved in a **log file**.

When you are in Stata, you can also type `help` or `search` for on-line instructions. `help` should be followed with specific commands whilst `search` can be followed by topic names, keywords, author, manual, stb etc. (You may have noticed that we follow the Stata convention and put command names in typewriter-style typeface that looks like this. A lower-case letter is used in the previous sentence purposefully because `help` is a Stata command and Stata commands must be typed in lower-case letters, even when placed at the beginning of a sentence). Note that prior to Stata 7, `lookup` was used which has the function of `search`. The function of `lookup` is no longer documented in detail in Stata 7 although we can still use it. If what you wish to find is not in your computer, you could use `net search` or `findit` followed by what you wish to find out from the Stata web site. For example,

```
. help logistic
      (output omitted)

. search correspondence analysis
      (output omitted)

. search William Gould
      (No output because William Gould is not a command.)

. net search William Gould
(contacting http://www.stata.com)
53 packages found (STB listed first)
-----
      (output omitted)

. findit Andrew Pickles
1 package found (STB omitted)
-----
gllamm from http://www.stata.com/meetings/5uk
  Generalised linear latent and mixed models / Sophia Rabe-Hesketh Andrew
  Pickles / Institute of Psychiatry University of Manchester /
  spaksrh@iop.bpmf.ac.uk andrew.pickles@man.ac.uk / Colin Taylor /
  Addiction Research Unit / / After installation, see help gllamm.
```

Many parts of the `search` results are highlighted so that you could click on that and go directly to the topics of interest.

1.3 Starting and exiting Stata

To start running Stata, double click on the Stata icon on your Windows desktop (or double click on any dataset in Stata format, provided it is small enough, namely, within the default memory limit).

The Stata window will appear, displaying

- A menu and row of 13 icons (buttons) across the top
- A Stata Command window (bottom right)
- A Stata Results window (top right)
- A Variables window (bottom left)
- A Review window which displays previous commands (top left)

The menu is much like other systems such as Word or SPSS. The 13 buttons are:



In case you forget what a button can do, you can place the mouse pointer over it for a moment, and a box will appear with a description of that button.

1.3.1 Opening a file: use data and systax files

There are different ways to open a data file:

1: In the Command window, type

```
use filename, clear
```

where *filename* is the name of the file you wish to open. The Stata files are in the format of *filename.dta* but you do not need to type the suffix. Be sure that you are in right directory. The default is *C:\Data* but you can organise the data files in any way you wish, just like Word. Suppose the file we wish to open is **LFS2002.dta** and is placed in the directory **C:\Data_Stata\Course4_Stata_for_LFS**, the following commands would serve the same purpose:

```
clear
set more off
set mem 20m
use "C:\Data_Stata\Course4_Stata_for_LFS\LFS2002.dta", clear
```

where *clear* is an option. If there are spaces in the file name, you **must** put the directory and the file name in brackets. It is good practice to put brackets anyway. For instance, suppose we have a dataset in the same directory called **my newdata.dta** and wish to open the file:

```
. use my newdata.dta, clear
invalid 'my'
r(198);
```

It is good practice **NOT** to have blanks in the file names. For instance, as you become more experienced in your analysis, you may be involved in several projects at the same time. You may find it convenient to organise your syntax and data directories separately so that you can save your syntax files from time to time (the data files may be too big to save onto disks or even CDs). The syntax and data directories can have many levels or subdirectories, but using syntax files, it makes no difference how many levels your syntax or data directories are. This manual will show you how to write syntax files.

The *clear* is an option in Stata terminology. Options follow commas. The options are implemented in *ado* (automatic do) files that come with Stata and can be updated from time to time. There is already an *ado* file for *clear* installed. If there are data in memory, we type *clear* to clear the data in which case we invoke the *ado* file.

2: In the folder where a data file is placed, double click on the file (provided that the file is small enough)

3: Use the open button

4: From the menu, select **File**, then **Open**

The *Use New Data* dialogue box allows you to select the required folder and file. Select and open the file **LFS2002.dta**. The Variables window will now display a list of variables in the data file and their variable labels. Since Stata 7 allows for very long variable names (32 characters) and variable labels (80 characters) [it also allows 32 characters for value labels], you cannot see the variable label because some space is left between variable names and labels. You can pull the Variable window to the right to see the variable labels, which is time-consuming and less aesthetically appealing. We can use

```
set varlabelpos 8
```


to change that. The default is 32 and we use the minimum. We can see several active Stata windows open at the same time.

The Review window

This window displays the commands typed in the Command window. We can place the mouse on a command line in the window and click once to put the command back to the Command window where it can be edited and executed again; to double click it will re-execute the command. Clicking on the button left on the Review window will allow you to save the command into a do file where one can later edit and execute the whole series of commands.



The Results window

This window displays the results. The results, if long, are displayed one screenful at a time and Stata will prompt us with `--more--` where pressing the space bar or any other key will present the next page, and pressing enter will display the next line. If you accidentally or unknowingly give a command such as `list age`, which will send an annoyingly large number of pages to the Results window, you can stop the display by pressing `control & break` together or by clicking on the  button on the toolbar. You can copy the commands or results in the Results window to other places like the Command window or Word. You can, in the Command window, type

```
set more off
```

to prevent Stata from displaying the `--more--` message. But it is unwise to do so.

The Variables window

This window shows the variables and their labels in the data set we have opened. Again, placing the mouse on a variable and clicking once will send it to the command window to be edited and executed.

The Command window

This window is for issuing commands (interactive use of Stata). The following features prove very useful to users:

| <u>Press</u> | <u>Result</u> |
|--------------|--|
| <i>PgUp</i> | Steps back through commands. |
| <i>PgDn</i> | Steps forward through commands.. |
| <i>Esc</i> | Clears Command window. |
| <i>Home</i> | Move the cursor to the beginning of the command line. |
| <i>End</i> | Moves the cursor to the end of the command line. |
| #review 10 | Review the last 10 commands (the default is to display the last 5 commands. You could change to 15 or 20 if you like). |
| #review 20 2 | Reviews the 20 th and the 19 th previous command lines. |
| * | Tells Stata that what follows * is a comment rather than a command, hence not to be executed. |

Note that `/*` `*/` which can be placed anywhere in the syntax except in between the variable list as a comment in the do file, is *not* available for interactive use in the Command window.

1.3.2 Logging the results

One may save the results by typing, in the Command window, something like:

```
log using logname
```

where `logname` is any name you wish to give.

The options `, replace` or `, append` are available, and logs can be closed by

```
log close
```

Logs can also be temporarily switched on and off by typing

```
log off  
log on
```

1.3.3 Exiting Stata and saving data

To exit from Stata, simply type `exit` or click on the X at the top right corner. If you have not changed the data, Stata will allow you to exit without complaint. If you have changed the data but still intend to exit without saving the data, Stata will issue a warning. If you are sure that you do not want to save, you can ignore the warning and exit by typing `exit, clear`. If you do intend to save the changes, you could

`save newname` to save as a new file, or
`save existingname, replace` to overwrite the existing file.

What is defined as change of data in Stata? According to William Gould from StataCorp:

Stata defines a changed dataset as

1. An existing variable changes its value
2. A new variable is created
3. A previously existing variable is dropped
4. Observations are added
5. Observations are dropped

Changing the following does not count as ‘changing’ the dataset:

1. Placing or changing a variable label on a variable
2. Adding or removing a value label on a variable
3. Changing, adding, or dropping a value label
4. Changing, adding, or dropping a characteristic
5. Changing a display format
6. Changing the variable types using `-compress-`
7. Changing the order of variables using `-order-` or `-move-`

The thinking is to raise the red flag when a ‘substantive’ change has been made to the data. Items 1-7, were they to change, would not actually change a statistical result, although they would change how the results might appear.

1.3.4 Reserved names in Stata and Stata operators

There are some system names in Stata to which we need to pay special attention. System names begin with the underscore `_` as seen in the following examples:

| | |
|---------------------|---|
| <code>_n</code> | running number of the current observation |
| <code>_N</code> | total number of observations |
| <code>_all</code> | all variables |
| <code>_b</code> | vector of regression coefficients |
| <code>_se</code> | vector of standard errors of regression coefficients |
| <code>_merge</code> | variable created after merging files which tells us the source of resulting observation |

The following are the reserved names that should not be used as our variable names:

| | | | |
|--------------------|---------------------|--------------------|--------------------|
| <code>_all</code> | <code>double</code> | <code>long</code> | <code>_rc</code> |
| <code>_b</code> | <code>float</code> | <code>_n</code> | <code>_se</code> |
| <code>byte</code> | <code>if</code> | <code>_N</code> | <code>_skip</code> |
| <code>_coef</code> | <code>in</code> | <code>_pi</code> | <code>using</code> |
| <code>_cons</code> | <code>int</code> | <code>_pred</code> | <code>with</code> |

We will soon come to the analysis in Stata. Before we go into details, let us have a look at the operators used in Stata. Since some of you may have used SPSS, I thought that it would be a good idea to see the differences between SPSS and Stata in terms of the operators, which may be better appreciated if they are placed side by side.

| | SPSS | Stata | Description |
|---|---------|----------|---|
| Arithmetic operators | + | + | Addition |
| | - | - | Subtraction |
| | * | * | Multiplication |
| | / | / | Division |
| | ** | ^ | Raised to the power of (Exponentiation) |
| Relational operators | = (EQ) | == | Equal |
| | ^= (NE) | ~= or != | Not equal |
| | < (LT) | < | Less than |
| | > (GT) | > | Greater than |
| | <= (LE) | <= | Less than or equal |
| | >= (GE) | >= | Greater than or equal |
| Logical operators | & (AND) | & | And |
| | (OR) | | Or |
| | ~ | ~ or ! | Not |
| Notes: 1: The symbol = in Stata is for assignment and == for equality. 2: Letters such as eq ne lt are operative in SPSS but not allowed in Stata. | | | |

1.4 Interactive use of Command window

For most of our purposes, we do not need to import data but simply to use Stata files to do analysis. There are two general ways of carrying out analysis using Stata. The first is to use the Stata Command window for simple tasks. The second, used by most experienced analysts, is to use syntax (`.do`) files. We shall now have a brief look at how to use the Command window. It is highly recommended that commands for data management and data analysis be written and executed via `.do` files. Nevertheless, there are occasions when jobs may be better (that is, more easily) done interactively in the Command window, such as using Stata as a calculator or for certain types of data exploration.

1.4.1 Using Stata as a calculator: displaying data

Stata can be used as a very convenient and effective calculator with the `display` command (once you are familiar with this, you will no longer need a calculator if you have access to Stata). Note that the first two letters in the command `display` are underlined to denote that the minimum number of letters to type for this command is 2. Stata commands follows the minimalist principle, meaning that only the minimum number of letters is required for commands, options and variables as long as they are not confusing: that is, as long as they are unique. Some commands are destructive in nature like `clear` or `replace`. For these options, Stata is highly protective and will not allow you to use abbreviated versions. The following are some examples of using Stata as a calculator.

```

. display (57.23-3.21)/(12.8+4.56)
3.1117512

. displa 1.5e+3
1500

. displ 1.5e-3
.0015

. disp log(250)
5.5214609

. dis ln(250)
5.5214609

. di log10(250)
2.39794

. display exp(3.6)
36.598234

. display sqrt(2*log(100))/(3^2-7)
1.5174271

. display chiprob(2, 6.45)
.03975578

. display _N
63538

```

Most of the commands above are straightforward. Two of them may need explanation. First, the command `display chiprob(2, 6.45)` tests the chi square probability for the degree of freedom equal to 2 and the change of deviance equal to 6.45. Suppose we have fitted a regression model and then wish to fit another model with more explanatory variables. The added variable uses two degrees of freedom and explains an additional variance of 6.45. We can use the command to see whether the added variables constitute a significant change. The result `.03975578` shows that it is significant at the 0.05 level. The second point, as earlier noted, is that `_N` is the Stata internal variable for the total number in the data set (`_n` is that for the current (record) number).

1.4.2 Using Stata Command Window for immediate commands

Sometimes, you may wish to use the interactive facility for the so-called immediate commands in Stata, commands that end with `i` such as `cii` (for testing confidence intervals) or `ttesti` (for doing one-sample or two-sample t-tests). For example, if you know the sample size, the mean and the standard deviation, you can, using immediate commands, find out the standard error and the confidence intervals without real data. Using immediate commands will not harm the data in memory.

```

. cii 97 24 6

```

| Variable | Obs | Mean | Std. Err. | [95% Conf. Interval] | |
|----------|-----|------|-----------|----------------------|----------|
| | 97 | 24 | .6092077 | 22.79073 | 25.20927 |

For a sample of 97 observations, with a mean value of 24 and standard deviation of 6, the standard error is .61 and the confidence intervals range from 22.8 to 25.2. As a matter of fact, whether or not we know the mean does not really matter if we know how to get standard error with standard deviation and the sample size, i.e.,

Standard error = standard deviation/(square root of the sample size)

We can use the display function to get the same result:

```
. di 6/sqrt(97)
.6092077
```

(Do you remember how to get the 95%, the 99% and the 99.9% confidence intervals?)

Similarly, we could test the hypothesis that the true mean equals 22:

```
. ttesti 97 24 6 22
```

One-sample t test

```
-----+-----
      |      Obs      Mean   Std. Err.   Std. Dev.   [95% Conf. Interval]
-----+-----
      x |         97         24   .6092077         6   22.79073   25.20927
-----+-----
```

Degrees of freedom: 96

Ho: mean(x) = 22

Ha: mean < 22
t = 3.2830
P < t = 0.9993

Ha: mean ~= 22
t = 3.2830
P > |t| = 0.0014

Ha: mean > 22
t = 3.2830
P > t = 0.0007

This shows that we can reject the null hypothesis at the 0.001 level. Suppose we have two samples of the same size, but with different means and standard deviations, and we wish to test the hypothesis that there is no difference between two sample means:

```
. ttesti 97 24 6 97 28 8
```

Two-sample t test with equal variances

```
-----+-----
      |      Obs      Mean   Std. Err.   Std. Dev.   [95% Conf. Interval]
-----+-----
      x |         97         24   .6092077         6   22.79073   25.20927
      y |         97         28   .8122769         8   26.38764   29.61236
-----+-----
combined |        194         26   .5264237   7.332234   24.96172   27.03828
-----+-----
      diff |             -4   1.015346             -6.002665   -1.997335
-----+-----
```

Degrees of freedom: 192

Ho: mean(x) - mean(y) = diff = 0

Ha: diff < 0
t = -3.9395
P < t = 0.0001

Ha: diff ~= 0
t = -3.9395
P > |t| = 0.0001

Ha: diff > 0
t = -3.9395
P > t = 0.9999

We can say, with confidence, that there is a difference between the two means.

1.5 Exploring data

There are various ways to explore the data. The most frequently used are commands such as `browse`, `list`, `codebook`, `describe`, `ds`, `lookfor`, `summarize`, `tabulate`, `count`, `edit` etc.

1.5.1 browsing and listing data

To browse the dataset does not (allow you to) change the values. You could browse the whole data set, or a particular variable, or a particular observation for a particular variable, or

particular range of observations for a variable or a variable list, with or without the label. For example,

```
brose
```

This will allow you to browse the whole dataset. Note that the minimum command is br in this case, as underlined. This command is equal to

```
brose _all
```

We can choose a range of variables to look at if we do not want to see all variables.

```
br
```

This will browse variables from sex to nation. The – in Stata means the same as TO in SPSS for variable list. The command is equivalent to

```
br
```

Another very useful feature of Stata is the use of * which means ‘zero or more characters go here’. For instance, if you suffix * to a partial variable name, you are referring to all variable names that start with that letter combination. For example, if we want to know what variables in our file begin with s, we can find out by

```
. ds s*  
sex      status      soc2km      sc2kmmj      schm99      secjob      start
```

We can look at the s variables by

```
br
```

Which is the same as issuing the following command:

```
br
```

If, for instance, we wish to know the values of certain variables for certain observations, we can:

```
br
```

And we find that the respondent is aged 21. Compare this with this:

```
br
```

Note that while the ‘age status’ refers to variable list, the ‘in 30005/30008’ to a list of cases. In the Stata jargon, in 30005/30008 is called a range (ranging from observation 30005 to observation 30008) and ,nolabel is called an option. We will discuss it in greater detail as we go along. One could also look at the results in Results window by typing

```
list age in 30005  
list age status in f/5  
list age status in -4/l  
list age status in 12570/12575
```

In the commands above, f/5 means from the first to the fifth observations, and –4/l (letter l) means from the fourth observation from the bottom of the file to the last observation. Note that, as earlier mentioned, Stata recognises the abbreviated variable names such as *statu* and


stat, but had we typed `num`, Stata would have complained saying that ‘num ambiguous abbreviation’. This is because there are two variables starting with *num* (`numchild numchil1`) which can be abbreviated as such. The `-` in `sex - ages` tells Stata to list all variables from `sex` to `ages`, but in this case there is only one variable in between. Another point to note in this case is that we had sorted the data by ethnicity before we used the `list` command. We will discuss the `sort` command in greater detail below. The options for `list` are `display`, `nolabel`, `noobs` and `doublespace`.

Sometimes it may be appropriate to change the default if there are many variables or observations you wish to list or if you expect to have a very long line in the command line. You could (blocked). You can block any command by placing an `*` at the beginning of a command like this:

```
*set log linesize 140
*set linesize 140
*set display linesize 140
```

Generally, however, there is no need to change these defaults. In order to know what the default is for Stata 7, you could type `query` (or just `q`). For instance, on my machine,

```
. q
----- Status
      type | float          linesize | 108
  virtual | off            pagesize  | 31
      more | on                dp    | period
      rmsg | off              trace   | off
  matsize | 40                level   | 95
  adosize | 128              logtype  | smcl
  textsize | 100             linegap  | 1
  graphics | on                |
----- Files
      log  | (closed)
  cmdlog  | (closed)
httpproxy- |
      host | wwwcache.mcc.ac.uk
      port | 3128
      auth | off
-----
```

In the Stata Browser window (the  button on the toolbar), the *Hide* button allows you to hide the currently selected column. The buttons with left and right arrows allow you to change the display by moving columns to the far left or far right. This is particularly useful after using the `egen` command so that we can check whether we have made any mistake. We can put all the relevant variables to the left or the right. Close the Browser window to return to the Command window.

1.5.2 Editing data

You could `edit` data in Command window or in Data Editor window. For example, you could invoke the Editor and change the values as you wish:

```
edit age sex in 1/5, nolabel
```

You could also make changes in the Commands window using the `replace` command.

```
replace age=120 in 1
```

would change the original value of age in the first observation to 120!

1.5.3 Codebook

Typing `codebook` without variable names will display information for all variables in the dataset, which is the same as to type `codebook _all.` This is not a very useful way of exploring the data set, especially when we have many variables in the data set. Instead, one may wish to look at the information associated with one variable, for example.

```
. codebook status

status ----- economic status
      type: numeric (float)
      label: status

      range: [1,4]                units: 1
unique values: 4                  coded missing: 0 / 65943

      tabulation: Freq.   Numeric   Label
                  40617      1   employed/scheme
                  5172      2   self-employed/unpaid fam
                  2409      3   ilo unemployed
                  17745      4   not in labour force
```

and we are told that there are 4 ‘unique’ (i.e., different) values for this variable which we frequently call ‘categories’, and we are also told the number of missing values. There are no missing values in this variable

1.5.4 Describing data

`describe` can be followed by variable names. Typing `describe` alone without variable names gives information on all variables. `ds,` on the other hand, is much like ‘display index’ in SPSS and gives a list of variables in the data set, which is very useful for refreshing our memory.

```
. d

Contains data from C:\Documents and Settings\mscssvah\My Documents\Teaching datasets\LFS\All
cases\version 2\Final files\LFS2002.dta.dta
  obs:      65,943
  vars:      60                19 Aug 2003 15:47
  size: 16,090,092 (23.0% of memory free)

-----
variable name  storage  display  value  variable label
              type    format   label
-----
ten96         float   %9.0g   ten96   accommodation details
house         float   %9.0g   house   accommodation details (grouped)
sex           float   %9.0g   sex     sex
age           float   %9.0g   age     age last birthday
ages         float   %9.0g   ages    age groups in 5 yearly intervals
nation        float   %9.0g   nation  nationality
cry01         float   %9.0g   cry01   country of birth
region        float   %9.0g   region  region of usual residence
numchild      float   %9.0g   numchild number of children in the
              household aged 0-4
numchill      float   %9.0g   numchill number of children in the
              household aged 5-16
ayfl19        float   %9.0g   ayfl19  age of youngest dependent child
              in family aged <19
ethnic        float   %9.0g   ethnic  ethnicity
```

| | | | | |
|----------|-------|-------|----------|--|
| fb | float | %9.0g | fb | whether born outside uk |
| arrival | float | %9.0g | arrival | year of arrival in uk |
| marstt | float | %9.0g | marstt | marital status |
| livtog | float | %9.0g | livtog | whether living together as a couple |
| married | float | %9.0g | married | whether married/cohabiting |
| inecaca | float | %9.0g | inecaca | economic activity |
| status | float | %9.0g | status | economic status |
| ilodefa | float | %9.0g | ilodefa | basic economic activity (ilo definition) |
| grsswk | float | %9.0g | grsswk | gross weekly pay in main job (£) |
| hourpay | float | %9.0g | hourpay | gross hourly pay (£) |
| soc2km | float | %9.0g | soc2km | occupation (main job) |
| sc2kmmj | float | %9.0g | sc2kmmj | occupation in main job (grouped) |
| nsecm | float | %9.0g | nsecm | ns-sec category (main job) |
| jobtyp | float | %9.0g | jobtyp | permanent or temporary job |
| conmpy | float | %9.0g | conmpy | year started working for current employer |
| ptime | float | %9.0g | ptime | whether part-time (self-reported) |
| ptimehrs | float | %9.0g | ptimehrs | whether part-time (work <31 hours per week) |
| ttushr | float | %9.0g | ttushr | total usual hours in main job per week (including overtime) |
| public | float | %9.0g | public | whether working in public or private sector |
| appren | float | %9.0g | appren | recognised trade apprenticeship |
| schm99 | float | %9.0g | schm99 | type of government employment or training scheme |
| manage | float | %9.0g | manage | managerial duties or supervising |
| secjob | float | %9.0g | secjob | whether had 2nd job in reference week |
| teclec | float | %9.0g | teclec | whether on scheme run by a tec or lec |
| newdeal | float | %9.0g | newdeal | new deal option |
| ytetmp | float | %9.0g | ytetmp | yt, et, tec schemes |
| ytetjb | float | %9.0g | ytetjb | whether had paid job in addition to scheme |
| wrking | float | %9.0g | wrking | whether had paid job in addition to scheme |
| jbaway | float | %9.0g | jbaway | whether temporarily away from paid work |
| ownbus | float | %9.0g | ownbus | whether doing unpaid work for own business |
| relbus | float | %9.0g | relbus | whether doing unpaid work for relatives business |
| nstat | float | %9.0g | nstat | employment status in main job |
| look4 | float | %9.0g | look4 | whether looking for paid work in last 4 weeks |
| lkyt4 | float | %9.0g | lkyt4 | whether looking for a place on a government scheme in the last 4 weeks |
| start | float | %9.0g | start | whether could start work within the next 2 weeks |
| wait | float | %9.0g | wait | whether waiting to take up a job |
| likewk | float | %9.0g | likewk | whether would like to work |
| ystart | float | %9.0g | ystart | reason why could not start work within 2 weeks |
| nolook | float | %9.0g | nolook | reason not looking for work (if likewk=1) |
| nowant | float | %9.0g | nowant | reason not looking for work (if likewk=2) |
| hiqual | float | %9.0g | hiqual | highest qualification |
| hiquald | float | %9.0g | hiquald | highest qualification (grouped) |
| levqual | float | %9.0g | levqual | level of highest qualification |
| edage | float | %9.0g | edage | age completed continuous full-time education |
| bhealth | float | %9.0g | bhealth | bad health that limits paid work |
| quart | float | %9.0g | quart | quarter taken from |

Sorted by:

This tells us at a glance the key information in the data set: the number of observations and variables, variable names, storage types, variable labels etc. Note that if we had put in any

notes for the data or for any of the variables, Stata would tell us. We shall come to learn how to make notes later. It is the case that note-taking is especially important to remind ourselves what we do in our analysis. We can, of course, make notes in the do files without actually putting them in the data set.

Another very important feature, one which I personally use very often whenever I come to a new data set, is `ds`:

```
. ds
. ds
ten96      house      sex      age      ages      nation    cry01     region
numchild   numchill  ayfl19   ethnic   fb        arrival   marstt    livtog
married    inecaca   status   ilodefa  grsswk    hourpay   soc2km    sc2kmmj
nsecm      jobtyp    conmpy   ptime    ptimehrs  ttushr    public    appren
schm99     manage    secjob   teclec   newdeal   ytetmp    ytetjb    wrking
jbaway     ownbus    relbus   nstat    look4     lkyl4     start     wait
likewk     ystart    nolook   nowant   hiqual    hiquald   levqual   edage
bhealth    quart
```

You may not have got the function on your machine. If that is the case, you can update by typing

```
update all
```

and then follow the instructions. It is good practice to update every one or two months to keep up with the rapid developments in Stata.

1.5.5 Looking for information in variable and value labels

Sometimes one may wish to use a variable with a particular feature in either variable names or value labels. Suppose we do not know whether there is any information on social class, or we do know that there is a variable on class but have forgot the names for the variable. How do we quickly find out, especially if there are hundreds or thousands of variables in a data set? In SPSS, we have to go through a complicated route, use display dictionary, copy the output into Word and then use Find. In Stata, it is much more direct: we simply use `lookfor` which will help us find out the variables by searching for strings among all variable names and labels that contain whatever we wish to find. We type

```
. lookfor class
```

```
.
```

Alas! The class variable is not included in this file. What about employment status?

```
. lookfor employment status
```

| variable name | storage type | display format | value label | variable label |
|---------------|--------------|----------------|-------------|---|
| marstt | float | %9.0g | marstt | marital status |
| status | float | %9.0g | status | economic status |
| schm99 | float | %9.0g | schm99 | type of government employment or training scheme |
| nstat | float | %9.0g | nstat | employment status in main job |

Note that the words ‘employment’ and/or ‘status’ are in the variable or value labels of four variables within the file. When you prepare data, even if for your own use, it is always good advice to make full documentation in terms of variable and value labels so that you can check

for future use. (Your supervisor or collaborator or referees may demand what you exactly did and why!).

The `ds` and `lookfor` are two of the most often used commands in finding what is available for particular research purposes when we have only a vague memory of what is there. Do not get confused with `lookup` which was used in older versions of Stata and can still be used in Stata 7 but it is not listed in the manuals. It is now replaced by `search`.

```
search logit
lookup logit
```

1.5.6 Counting data and checking data consistency

It is sometimes necessary to check the number of observations available for a particular analysis. For instance, how many Pakistani women in the age band 25-40 in **LFS2002.dta** are foreign born? We can find out the information easily. But we need to check the variables to know what is what.

For instance, Pakistani = 6 in `ethnic`; foreign born is in the variable `fb` = 1, women is coded 1 in `sex`. We have to be sure about the variables and categories. The way to do this is to use `tabulate variablename, with` and then without the option `, nolabel`.

```
tab          ethnic
tab          ethnic, nol
tab          fb
tab          fb, nolabel
tab          sex
tab          sex, nolabel
. count          if ethnic==6&sex==1&fb==1&age>=25&age<=40
  85
```

This shows that there are 85 Pakistani women born outside the UK who are in the age group.

1.6 do files: an example

Commands written in the form of `do` files can be saved, edited and checked in future. It is always preferable to use `do` files, that is, to write your own syntax. The importance cannot be over-emphasised: when you write a report or paper, it takes a long time and you may forget what you did two months or even two weeks ago. The command window is basically for very simple stuff but `do` files are really the gems of your analysis: you can make it in as good an order as you wish and build up as complicated models as you wish. Also, since we can copy and paste and make whatever changes we wish, it is really much quicker and safer to use `do` files than to use menus. Finally, when you submit a report or a paper, you may find that the referees (or your collaborators) may come to you three months later and ask you to change or re-do some analysis. And, in the meantime, you may have been working on another project or paper and have forgotten what you did!

To write your own syntax is not at all difficult, as you will find all the syntax for this course in the Appendix. As a starting point, you may, if you so wish, copy the syntax and save it in your Stata as an example. If you do not know what syntax to use, you can, apart from the help and search facilities mentioned above, set the menu on. Stata 8 already has more

comprehensive menus on as a default but I do not have Stata 8 at the moment. It is, however, no difficulty, as you can set it on by typing

quest on

Actually, you may, as you become more experienced, wish not to have the menus on. You can set it off by

quest off

Here is an example of the format where only one job is carried out. But you could write whatever commands for whatever jobs are needed for the analysis. The file is called **“C:\Data\Couse4_Stata_for_LFS\Stata_for_LFS_Example.do”**.

```
version 7.0                /*To make the programme usable for later versions */
set memory 32m             /*To give enough space for the file */
log using Stata_for_LFS_Example,replace /*To replace existing log if any */
set more off              /*To display all results without manual manipulation */
use LFS2002 ,clear        /*To clear data in memory if any */
tab sex ptime,r          /*Command 1: gender differences in parttime work */
tab eth marr if age>=35&age<=50,r /*Command 2: who are likely to be married
                           other commands
                           A
                           B
                           ...
                           Z
                           */
log close                 /*To close the log */
*How to view the log?
/* The default in Stata 7 is a smcl file (Stata Markup and Control Language)
   To look at results in the smcl file, you could, via Stata menu:
       (1) go to File->Log->View
           (File or URL->Browse for "using Stata_for_LFS_Example")
           or in Command window, type
       (2) view "C:\DATA\Course4_Stata_for_LFS\Stata_for_for_LFS_Example.smcl"

   To turn the smcl into log file viewable in notepad or word, via Stata menu:
       (3) go to File->Log->Translate
           (input filename="using Stata_for_LFS_Example"
            output filename="new_name")
           or, by issuing the following commands in Command window:
       (4) translate "C:\DATA\Course4_Stata_for_LFS\Stata_for_LFS_Example.smcl" /*
           */ "C:\DATA\Course4_Stata_for_LFS\Stata_for_LFS_Example.log", replace */
exit, clear
```

We can execute the file in different ways. In the folder “C:\Data\Course4_Stata_for_LFS” where the do file is placed, we could double click the file and it will be executed automatically. In the Do-File Editor, we could select the part you wish to implement (in this case the whole file) and choose Do Selection under Tools menu. The third way is to execute it in Command window where we could type

```
do Stata_for_LFS_Example
```

Or type

```
run Stata_for_LFS_Example
```

To run the file will execute it without presenting the results, which is the same as typing

```
quietly do Stata_for_LFS_Example
```

Exercises and suggested answers

Questions:

Using **LFS2002.dta**

- 1.1 Find out socio-demographic information (age, gender, marital status, employment status, housing tenure, long-term limiting illness) for observations 10020 to 10025 with labels and without labels [*age sex married status house bhealth*].
- 1.2 What is the ethnic group for observation 12006 [ethnic]? What is the proportion of the respondents in the sample who are in that category?
- 1.3 How many Bangladeshi women are unemployed?
- 1.4 How many Pakistani women are in part-time work?
- 1.5 What proportion of Indians are owner-occupiers?
- 1.6 Which variable contains information concerning hour pay?

Suggested answers:

*Exercise 1.1

```
list age sex married status house bhealth in 10020/10025
list age sex married status house bhealth in 10020/10025, nolabel
```

*Exercise 1.2

```
list ethnic in 10026, nolabel
tab ethnic
```

*Exercise 1.3

```
tab status if sex==1&ethn==7
```

*Exercise 1.4

```
tab ptime if sex==1&ethn==6
```

*Exercise 1.5

```
tab ethn house,r
```

*Exercise 1.6

```
lookfor hourpay
```

Chapter 2 Exploratory analysis with Stata

2.1 The Stata command structure

Most Stata procedures conform to a common command structure like this:

```
[by varlist:] command [varlist] [=exp] [if exp]] [in range] [weight] [,options]
```

[] are optional. Each command is written on one line terminated by a hard return (pressing the enter key). To use several lines, we can use the `/* */` structure in do files, but this is not available in Command window. One can, in a do file, change the default [note that this is not available in the Command window]:

```
#delimiter ;
```

so that only when a semicolon is met will Stata treat it as the end of a command. One could change the delimiter back to carriage return by typing

```
#delimiter cr
```

Two other points to note about the Stata commands:

- (1) They must be entered in lower case (almost without exception).
- (2) Stata allows abbreviations for commands and variable names as long as they meet the minimum requirements.

In the following, we will give some examples of the command structure.

```
[by varlist]
```

tells Stata to execute the command for each group in the `by varlist`. The variable (list) needs to be sorted first using `sort`, which can be done in different ways (all blocked here as there are many examples below).

```
/* Option 1: to use sort and by in two command lines */
*sort sex
*by sex: summ age

/* Option 2: to combine sort and by in one command line */
*by sex, sort: summarize age

/* Option 3: to use the bysort structure */
*bys sex: summarize age
```

```
[if exp]] [in range] [,options]
```

The noticeable feature about this is that this is the recommended order in which segments of a command be entered but, on the other hand, there is no particular ordering required. For example, we can put the `if exp` before or after the `in range`, and put the `,option` before or after the `if` condition (admittedly, this is not good practice). Again, these commands are all blocked as there are many examples below.

```
*summ hourpay if sex==1 & age>=20 &age<=64 in -100/1
*summ hourpay in -100/1 if sex==1 & age>=20 &age<=64
*summ hourpay, detail, in -100/1 if sex==1 & age>=20 &age<=64
*summ hourpay in -100/1 if sex==1 & age>=20 &age<=64 , detail
```

2.2 Summarising and tabulating data

2.2.1 Summarising data

`summarize` for continuous variables and `tabulate` for categorical variables are convenient ways of summing up the data (notice the American usage). One could look at the `details` in the summarized data by using the option `detail` after `summarize`. The two ways can be combined (`tabsum`).

```
. summ hourpay if stat==1&hourpay>0/*for employees with no missing data on pay only*/
```

| Variable | Obs | Mean | Std. Dev. | Min | Max |
|----------|-------|----------|-----------|-----|-------|
| hourpay | 31410 | 9.700548 | 6.99723 | .05 | 204.8 |

```
. su          hour if status==1, detail
```

```
          gross hourly pay (£)
-----
Percentiles      Smallest
1%              -9          -9
5%              -9          -9
10%             -9          -9      Obs          40617
25%             3.18        -9      Sum of Wgt.  40617

50%             6.38
                    Largest      Mean          5.461536
75%             10.3        138.46      Std. Dev.    9.958244
90%             15.66         150      Variance    99.16663
95%             19.76        163.5     Skewness    1.147692
99%             32.05        204.8     Kurtosis    15.93524
```

This shows that even among employees (`status==1`), there are many people whose hourpay is not reported (-9). Can you find out how many employees reported missing hourpay from what we discussed in the last chapter? If you cannot remember, here is the command:

```
. count if stat==1&hourpay==-9
9207
```

Note that whilst summarizing hours without `details` gives results as would be obtained by using descriptive hours in SPSS, to do it with the `detail` option gives a lot of statistics such as percentiles, mode and other things which are especially useful if we wish to turn the continuous variable into quartiles.

2.2.2 Tabulating data

`tabulate` is used both for frequency and for crosstabulation (frequency and `crosstab` in SPSS). If you use one variable, then `tab` gives you the frequency. If you give two variables, then it is crosstabulation. Three-way tables need the `sort` or `bysort` and then `tab`. If you give three variables A B C with `tab2`, then it is A by B, A by C and B by C. If you have four variables A B C D and wish to have A B C crosstabulated with D respectively, you need to use the `for` structure. If you just wish to list frequencies of several variables, you can use

tab1. If you update, you can use `tab1` `tab2` and `tabm` (these may, however, NOT be available on university clusters). You can even combine `tabulate` and `sum` or use `table` with the `content()` options. Here are examples:

For frequency:

```
. tab sex, m
```

| sex | Freq. | Percent | Cum. |
|--------|-------|---------|--------|
| male | 31817 | 48.25 | 48.25 |
| female | 34126 | 51.75 | 100.00 |
| Total | 65943 | 100.00 | |

For two way crosstabulation:

```
. tab eth sex, r m
```

| ethnicity | sex | | Total |
|-----------|----------------|----------------|-----------------|
| | male | female | |
| white | 28391 47.90 | 30880 52.10 | 59271 100.00 |
| caribbean | 263 45.03 | 321 54.97 | 584 100.00 |
| african | 189 44.26 | 238 55.74 | 427 100.00 |
| black oth | 87 39.91 | 131 60.09 | 218 100.00 |
| indian | 486 47.09 | 546 52.91 | 1032 100.00 |
| pstani | 269 46.30 | 312 53.70 | 581 100.00 |
| bdeshi | 116 47.54 | 128 52.46 | 244 100.00 |
| chinese | 92 50.83 | 89 49.17 | 181 100.00 |
| other | 491 49.10 | 509 50.90 | 1000 100.00 |
| . | 1433 59.58 | 972 40.42 | 2405 100.00 |
| Total | 31817 48.25 | 34126 51.75 | 65943 100.00 |

```
. tab2 sex married fb, r m
```

| sex | whether married/cohabiting | | Total |
|--------|----------------------------|----------------|-----------------|
| | no | yes | |
| male | 13915 43.73 | 17902 56.27 | 31817 100.00 |
| female | 14629 42.87 | 19497 57.13 | 34126 100.00 |

| | | | | | |
|-------|--|-------|-------|--|--------|
| Total | | 28544 | 37399 | | 65943 |
| | | 43.29 | 56.71 | | 100.00 |

-> tabulation of sex by fb

| sex | whether born outside uk | | Total |
|--------|----------------------------|------|--------|
| | no | yes | |
| male | 29237 | 2580 | 31817 |
| | 91.89 | 8.11 | 100.00 |
| female | 31101 | 3025 | 34126 |
| | 91.14 | 8.86 | 100.00 |
| Total | 60338 | 5605 | 65943 |
| | 91.50 | 8.50 | 100.00 |

-> tabulation of married by fb

| whether married/co habiting | whether born outside uk | | Total |
|-----------------------------------|----------------------------|------|--------|
| | no | yes | |
| no | 26566 | 1978 | 28544 |
| | 93.07 | 6.93 | 100.00 |
| yes | 33772 | 3627 | 37399 |
| | 90.30 | 9.70 | 100.00 |
| Total | 60338 | 5605 | 65943 |
| | 91.50 | 8.50 | 100.00 |

For three-way crosstabulation:

. bys sex: tab ethn fb,r

-> sex = male

| ethnicity | whether born outside uk | | Total |
|-----------|----------------------------|-------|--------|
| | no | yes | |
| white | 27135 | 1256 | 28391 |
| | 95.58 | 4.42 | 100.00 |
| caribbean | 144 | 119 | 263 |
| | 54.75 | 45.25 | 100.00 |
| african | 30 | 159 | 189 |
| | 15.87 | 84.13 | 100.00 |
| black oth | 67 | 20 | 87 |
| | 77.01 | 22.99 | 100.00 |
| indian | 166 | 320 | 486 |
| | 34.16 | 65.84 | 100.00 |
| pstani | 105 | 164 | 269 |
| | 39.03 | 60.97 | 100.00 |

| | | | |
|---------|-------|-------|--------|
| bdeshi | 20 | 96 | 116 |
| | 17.24 | 82.76 | 100.00 |
| chinese | 20 | 72 | 92 |
| | 21.74 | 78.26 | 100.00 |
| other | 118 | 373 | 491 |
| | 24.03 | 75.97 | 100.00 |
| Total | 27805 | 2579 | 30384 |
| | 91.51 | 8.49 | 100.00 |

-> sex = female

| ethnicity | whether born outside uk | | Total |
|-----------|----------------------------|-------|--------|
| | no | yes | |
| white | 29320 | 1560 | 30880 |
| | 94.95 | 5.05 | 100.00 |
| caribbean | 180 | 141 | 321 |
| | 56.07 | 43.93 | 100.00 |
| african | 32 | 206 | 238 |
| | 13.45 | 86.55 | 100.00 |
| black oth | 107 | 24 | 131 |
| | 81.68 | 18.32 | 100.00 |
| indian | 198 | 348 | 546 |
| | 36.26 | 63.74 | 100.00 |
| pstani | 112 | 200 | 312 |
| | 35.90 | 64.10 | 100.00 |
| bdeshi | 31 | 97 | 128 |
| | 24.22 | 75.78 | 100.00 |
| chinese | 20 | 69 | 89 |
| | 22.47 | 77.53 | 100.00 |
| other | 129 | 380 | 509 |
| | 25.34 | 74.66 | 100.00 |
| Total | 30129 | 3025 | 33154 |
| | 90.88 | 9.12 | 100.00 |

For separate listing of frequencies:

```
. tab1          eth sex fb, m
```

-> tabulation of ethnic

| ethnicity | Freq. | Percent | Cum. |
|-----------|-------|---------|--------|
| white | 59271 | 89.88 | 89.88 |
| caribbean | 584 | 0.89 | 90.77 |
| african | 427 | 0.65 | 91.42 |
| black oth | 218 | 0.33 | 91.75 |
| indian | 1032 | 1.56 | 93.31 |
| pstani | 581 | 0.88 | 94.19 |
| bdeshi | 244 | 0.37 | 94.56 |
| chinese | 181 | 0.27 | 94.84 |
| other | 1000 | 1.52 | 96.35 |
| . | 2405 | 3.65 | 100.00 |

```
-----+-----
      Total |      65943      100.00
```

-> tabulation of sex

```
-----+-----
      sex |      Freq.      Percent      Cum.
-----+-----
      male |      31817      48.25      48.25
      female |      34126      51.75      100.00
-----+-----
      Total |      65943      100.00
```

-> tabulation of fb

```
-----+-----
      whether |
      born |
      outside uk |      Freq.      Percent      Cum.
-----+-----
      no |      60338      91.50      91.50
      yes |      5605      8.50      100.00
-----+-----
      Total |      65943      100.00
```

For separate two-way crosstabulations with the for structures:

```
. for var eth status house bhealth: tab X sex, col
* for var eth status house bhealth: tab sex X, row /*same as above*/
```

-> tab ethnic sex, col

```
-----+-----
      ethnicity |      sex
      male      female |      Total
-----+-----
      white |      28391      30880 |      59271
      |      93.44      93.14 |      93.28
-----+-----
      caribbean |      263      321 |      584
      |      0.87      0.97 |      0.92
-----+-----
      african |      189      238 |      427
      |      0.62      0.72 |      0.67
-----+-----
      black oth |      87      131 |      218
      |      0.29      0.40 |      0.34
-----+-----
      indian |      486      546 |      1032
      |      1.60      1.65 |      1.62
-----+-----
      pstani |      269      312 |      581
      |      0.89      0.94 |      0.91
-----+-----
      bdeshi |      116      128 |      244
      |      0.38      0.39 |      0.38
-----+-----
      chinese |      92      89 |      181
      |      0.30      0.27 |      0.28
-----+-----
      other |      491      509 |      1000
      |      1.62      1.54 |      1.57
-----+-----
      Total |      30384      33154 |      63538
      |      100.00      100.00 |      100.00
```

-> tab status sex, col

```
-----+-----
      economic status |      sex
      male      female |      Total
```

| | | | |
|----------------------|--------|--------|--------|
| employed/scheme | 20189 | 20428 | 40617 |
| | 63.45 | 59.86 | 61.59 |
| self-employed/unpaid | 3684 | 1488 | 5172 |
| | 11.58 | 4.36 | 7.84 |
| ilo unemployed | 1392 | 1017 | 2409 |
| | 4.38 | 2.98 | 3.65 |
| not in labour force | 6552 | 11193 | 17745 |
| | 20.59 | 32.80 | 26.91 |
| Total | 31817 | 34126 | 65943 |
| | 100.00 | 100.00 | 100.00 |

-> tab house sex, col

| accommodation details (grouped) | sex | | Total |
|---------------------------------------|--------|--------|--------|
| | male | female | |
| owner/occupier | 24373 | 25153 | 49526 |
| | 76.63 | 73.72 | 75.12 |
| rented | 7434 | 8966 | 16400 |
| | 23.37 | 26.28 | 24.88 |
| Total | 31807 | 34119 | 65926 |
| | 100.00 | 100.00 | 100.00 |

-> tab bhealth sex, col

| bad health that limits paid work | sex | | Total |
|---|--------|--------|--------|
| | male | female | |
| no | 26597 | 29173 | 55770 |
| | 83.59 | 85.49 | 84.57 |
| yes | 5220 | 4953 | 10173 |
| | 16.41 | 14.51 | 15.43 |
| Total | 31817 | 34126 | 65943 |
| | 100.00 | 100.00 | 100.00 |

For combining categorical and continous tables:

. tabulate married sex if (status==1&hourpay>0), summ(hourpay) means nofreq

Means of gross hourly pay (£)

| whether married/co habiting | sex | | Total |
|-----------------------------------|-----------|-----------|-----------|
| | male | female | |
| no | 9.0985452 | 7.9928881 | 8.5115412 |
| yes | 12.454774 | 8.619823 | 10.514367 |
| Total | 11.13273 | 8.3577719 | 9.7005479 |

We are here concerned with employees with no missing values on hour pay. We notice here that for both men and women, married people get higher pay than the nonmarried. It seems unfair. But you may say that actually, the pattern may be confounded by age. Married people may be older and older people tend to be better paid. We thus need to control for age. We can do this as follows:

```
Tab          age,m
gen          age3=age
recode age3  min/35=1 36/50=2 51/max=3
label var   age3 "Age groups"
label def   age3 1 "16-35" 2 "36-50" 3 "51-65"
label val   age3 age3
tab         age age3, m
```

Note that we know that there is no missing data on age. If there are missing, either user missing or sysmissing (in SPSS jargons), we need to be very careful. It is always good idea to do a tab first before doing the recoding and then to check the coding afterwards.

```
. bys age3: tabulate married sex if (status==1&hourpay>0), summ(hourpay) means
nofreq
```

-> age3 = 16-35

Means of gross hourly pay (£)

| whether | sex | | Total |
|------------|-----------|-----------|-----------|
| married/co | male | female | |
| habiting | | | |
| no | 7.9627545 | 7.219154 | 7.5821269 |
| yes | 11.44572 | 8.885222 | 10.05827 |
| Total | 9.0921083 | 7.8042133 | 8.419803 |

-> age3 = 36-50

Means of gross hourly pay (£)

| whether | sex | | Total |
|------------|-----------|-----------|-----------|
| married/co | male | female | |
| habiting | | | |
| no | 11.687304 | 9.5887889 | 10.549159 |
| yes | 13.247244 | 8.7806837 | 10.974132 |
| Total | 12.858237 | 9.0031708 | 10.862447 |

-> age3 = 51-65

Means of gross hourly pay (£)

| whether | sex | | Total |
|------------|-----------|-----------|-----------|
| married/co | male | female | |
| habiting | | | |
| no | 10.323883 | 8.4640566 | 9.1777108 |
| yes | 11.874858 | 8.0882899 | 10.083455 |
| Total | 11.63026 | 8.1825651 | 9.8982134 |

It appears that for men in all three age groups, those who are married tend to be paid more. Why? Maybe married men tend to work harder, to be in better positions (they work harder for promotions because they have to feed their kids!). For women, there does not seem to be much difference.

Now you may say that, fine, but I do not want to see so many digital points after the decimal as no one is really interested in them. Can you just keep two points after the decimal? The answer is both yes and no. Yes, we can do this with the table command, no, we cannot do this with the tab command.

```
. table married sex if (status==1&hourpay>0), c(mean hourpay) format(%9.2f)
*      Note that format(%xxf) is available after the table command
-----
whether   |
married/c |      sex
ohabiting |   male  female
-----+-----
       no |    9.10    7.99
       yes |   12.45    8.62
-----

.  bys age3:  table  married  sex  if  (status==1&hourpay>0),  c(mean  hourpay)
format(%9.2f)
```

-> age3 = 16-35

```
-----
whether   |
married/c |      sex
ohabiting |   male  female
-----+-----
       no |    7.96    7.22
       yes |   11.45    8.89
-----
```

-> age3 = 36-50

```
-----
whether   |
married/c |      sex
ohabiting |   male  female
-----+-----
       no |   11.69    9.59
       yes |   13.25    8.78
-----
```

-> age3 = 51-65

```
-----
whether   |
married/c |      sex
ohabiting |   male  female
-----+-----
       no |   10.32    8.46
       yes |   11.87    8.09
-----
```

Now, you may say: wait a moment. Can you stack the tables together? I do not want to see so many sub-tables. OK. Let us do it.

```
. table married age3 sex if (status==1&hourpay>0), c(mean hourpay) format(%9.2f)
```

```

-----
whether      |                sex and Age groups
married/c  | ----- male ----- female -----
ohabiting  | 16-35 36-50 51-65   16-35 36-50 51-65
-----+-----
          no |  7.96 11.69 10.32    7.22  9.59  8.46
          yes | 11.45 13.25 11.87    8.89  8.78  8.09
-----

```

Note that for 4-way tables, we still need to use the `bysort` structure.

```
. bys fb: table married age3 sex if (status==1&hourpay>0), c(mean hourpay)
format(%9.2f)
```

```
-> fb = no
```

```

-----
whether      |                sex and Age groups
married/c  | ----- male ----- female -----
ohabiting  | 16-35 36-50 51-65   16-35 36-50 51-65
-----+-----
          no |  7.86 11.60 10.32    7.12  9.47  8.48
          yes | 11.41 13.16 11.82    8.82  8.70  8.02
-----

```

```
-> fb = yes
```

```

-----
whether      |                sex and Age groups
married/c  | ----- male ----- female -----
ohabiting  | 16-35 36-50 51-65   16-35 36-50 51-65
-----+-----
          no |  9.67 13.17 10.51    8.57 11.44  8.30
          yes | 11.71 14.18 12.74    9.63  9.88  9.15
-----

```

The results show that for both foreign and native born men and in whichever age groups, being married is associated with higher pay.

2.3 Sorting data and the by prefix

We have already covered much of this in the discussion above. Here we give a bit more discussion.

For many purposes, Stata requires us to `sort` the data before analysis can proceed. This is especially the case if we wish to use the `by` prefix such as using multiple tables with the `tabulate` or `table` command or indeed with many of the modelling approaches available in Stata (A noticeable feature about `sort` is that, as Stata's `tabulation` is restricted to two variables, using `by` prefix allows multiple tables which SPSS users might feel nostalgic about). Most Stata commands allow the `by` prefix, which repeats the command for each group of observations for which the values of the variables in the variable list are available. Up to Stata version 6, we have to `sort` the data before we can use the `by` command, but with Stata version 7, a `bysort` command is also available. The two features do the same job but the latter saves typing a line. Suppose that our data have not been `sorted` by sex and we wish to find the mean number of hours worked, Stata would complain and refuse to do the job:

```
. by sex: summ(hourpay)
not sorted
```

So we sort the data:

```
. sort sex  
. by sex: su hourpay  
*      results not shown
```

As earlier noted, in Stata version 7, a combined version is available:

```
bysort ethn: summarize hourpay  
            (output omitted here)
```

Let us look at some examples of multi-way tables. For categorical variables, we can use:

```
. bys sex:      tab ethn bhealth, row
```

-> sex = male

| ethnicity | bad health that limits paid work | | Total |
|-----------|-------------------------------------|---------------|-----------------|
| | no | yes | |
| white | 23491 82.74 | 4900 17.26 | 28391 100.00 |
| caribbean | 221 84.03 | 42 15.97 | 263 100.00 |
| african | 172 91.01 | 17 8.99 | 189 100.00 |
| black oth | 73 83.91 | 14 16.09 | 87 100.00 |
| indian | 396 81.48 | 90 18.52 | 486 100.00 |
| pstani | 207 76.95 | 62 23.05 | 269 100.00 |
| bdeshi | 88 75.86 | 28 24.14 | 116 100.00 |
| chinese | 87 94.57 | 5 5.43 | 92 100.00 |
| other | 429 87.37 | 62 12.63 | 491 100.00 |
| Total | 25164 82.82 | 5220 17.18 | 30384 100.00 |

-> sex = female

| ethnicity | bad health that limits paid work | | Total |
|-----------|-------------------------------------|---------------|-----------------|
| | no | yes | |
| white | 26287 85.13 | 4593 14.87 | 30880 100.00 |
| caribbean | 260 81.00 | 61 19.00 | 321 100.00 |
| african | 209 87.82 | 29 12.18 | 238 100.00 |
| black oth | 114 87.02 | 17 12.98 | 131 100.00 |
| indian | 469 85.90 | 77 14.10 | 546 100.00 |

| | | | |
|---------|-------|-------|--------|
| pstani | 242 | 70 | 312 |
| | 77.56 | 22.44 | 100.00 |
| bdeshi | 104 | 24 | 128 |
| | 81.25 | 18.75 | 100.00 |
| chinese | 86 | 3 | 89 |
| | 96.63 | 3.37 | 100.00 |
| other | 432 | 77 | 509 |
| | 84.87 | 15.13 | 100.00 |
| Total | 28203 | 4951 | 33154 |
| | 85.07 | 14.93 | 100.00 |

We find that for both men and women, Pakistani and Bangladeshi people were much more likely to have limiting long-term illness.

For continuous outcome variables, we can use:

```
. bys sex:      table married ethnic if grsswk>0, c(mean age mean grsswk) format(%9.2f)
```

```
-> sex = male
```

| whether married/c ohabiting | ethnicity | | | | | | | | |
|-----------------------------------|-----------|-----------|---------|--------|--------|--------|--------|--------|---------|
| | white | caribbean | african | black | oth | indian | pstani | bdeshi | chinese |
| no | 32.10 | 35.85 | 30.57 | 26.79 | 28.42 | 24.40 | 24.36 | 28.00 | 29.44 |
| | 355.43 | 383.49 | 309.17 | 206.05 | 371.02 | 247.88 | 296.91 | 429.06 | 350.30 |
| yes | 44.70 | 42.59 | 40.16 | 39.91 | 41.94 | 35.00 | 33.68 | 42.67 | 39.92 |
| | 508.36 | 400.92 | 495.57 | 370.73 | 497.88 | 381.23 | 274.05 | 411.53 | 546.07 |

```
-> sex = female
```

| whether married/c ohabiting | ethnicity | | | | | | | | |
|-----------------------------------|-----------|-----------|---------|--------|--------|--------|--------|--------|---------|
| | white | caribbean | african | black | oth | indian | pstani | bdeshi | chinese |
| no | 33.43 | 35.48 | 31.04 | 23.91 | 27.26 | 23.56 | 19.40 | 25.91 | 32.00 |
| | 264.34 | 281.77 | 223.32 | 170.97 | 262.56 | 220.81 | 134.20 | 340.09 | 300.68 |
| yes | 43.50 | 41.26 | 38.56 | 38.43 | 39.17 | 32.78 | 35.60 | 42.67 | 40.09 |
| | 250.11 | 271.65 | 316.25 | 304.71 | 298.88 | 244.95 | 115.20 | 282.06 | 312.09 |

We have included only those who have no missing data on weekly gross income (grsswk). We find that White married men, Black African married men and 'other' married men tend to have the highest incomes. There may be other factors involved but we see the limit of exploratory analysis here. We cannot effectively control for other factors which presumably affect incomes, such as country of birth, year of arrival in the UK (which accounts for language proficiency), education and class, full or part time job etc. We need to do this through multivariate regressions which we shall explore later.

2.4 Missing values

Unlike SPSS which accommodates two kinds of missing data (user missing and system missing) neither of which will interfere with results of statistical analysis, we need to be very careful with Stata in this regard. Stata has only one code for missing data denoted by a dot (.) and is internally held as the largest number for a given data type. The implication is that,

before we generate a new variable or use an existing variable for analysis, it is advisable and indeed necessary to check whether a variable has missing data and, if it does, how the missing values will impact on the results. Take the following example.

```
. tab jobtyp
```

| permanent or temporary job | Freq. | Percent | Cum. |
|-------------------------------|-------|---------|--------|
| does not apply | 25772 | 39.08 | 39.08 |
| no answer | 15 | 0.02 | 39.10 |
| permanent | 37544 | 56.93 | 96.04 |
| not permanent in some way | 2612 | 3.96 | 100.00 |
| Total | 65943 | 100.00 | |

```
. tab jobt, nol
```

| permanent or temporary job | Freq. | Percent | Cum. |
|-------------------------------------|-------|---------|--------|
| -9 | 25772 | 39.08 | 39.08 |
| -8 | 15 | 0.02 | 39.10 |
| 1 | 37544 | 56.93 | 96.04 |
| 2 | 2612 | 3.96 | 100.00 |
| Total | 65943 | 100.00 | |

If we do not use ‘missing’ we simply do not notice whether jobtyp has any user or sys missing data. This reminds us to be very careful and to check thoroughly before we do any analysis. Any serious researcher has to take full responsibility in all aspects, from data management, data analysis to interpreting results. We see this in this example. Now when we use the missing option, we find that -9 stands for ‘does not apply’ and the -8 for ‘no answer’. We need to code this as missing in Stata language, that is, to be denoted by dot. We do this as follows.

```
gen      jobperm=jobtyp
replace  jobperm=. if jobtyp<1
replace  jobperm=0 if jobtyp==2
lab var  jobperm "Whether permanent job"
lab def  jobperm 1 "Permanent" 0 "Non-perm"
lab val  jobperm jobperm
tab      jobtyp jobperm,m
```

```
. tab      jobtyp jobperm,m
```

| permanent or temporary job | Whether permanent job | | . | Total |
|-------------------------------|-----------------------|-----------|-------|-------|
| | Non-perm | Permanent | | |
| does not apply | 0 | 0 | 25772 | 25772 |
| no answer | 0 | 0 | 15 | 15 |
| permanent | 0 | 37544 | 0 | 37544 |
| not permanent in some way | 2612 | 0 | 0 | 2612 |
| Total | 2612 | 37544 | 25787 | 65943 |

We can SEE that everything is coded correctly. In case that we need to use this variable later, it would be better to put a note on the variable.

notes jobperm: Same as jobtyp but with 'Does not apply' and 'No answer' set as missing.

Now if we have a look at the file again, we see the * denoting note for the variable jobperm.

. desc

```
Contains data from C:\Documents and Settings\mscssvah\My Documents\Teaching datasets\LFS\All
cases\version 2\Final files\LFS2002.dta.dta
  obs:      65,943
  vars:      62
  size:    16,617,636 (20.3% of memory free)
  19 Aug 2003 15:47
```

| variable name | storage type | display format | value label | variable label |
|---------------|--------------|----------------|-------------|---|
| ten96 | float | %9.0g | ten96 | accommodation details |
| house | float | %9.0g | house | accommodation details (grouped) |
| sex | float | %9.0g | sex | sex |
| age | float | %9.0g | age | age last birthday |
| ages | float | %9.0g | ages | age groups in 5 yearly intervals |
| nation | float | %9.0g | nation | nationality |
| cry01 | float | %9.0g | cry01 | country of birth |
| region | float | %9.0g | region | region of usual residence |
| numchild | float | %9.0g | | number of children in the household aged 0-4 |
| numchill | float | %9.0g | | number of children in the household aged 5-16 |
| ayfl19 | float | %9.0g | ayfl19 | age of youngest dependent child in family aged <19 |
| ethnic | float | %9.0g | ethnic | ethnicity |
| fb | float | %9.0g | fb | whether born outside uk |
| arrival | float | %9.0g | arrival | year of arrival in uk |
| marstt | float | %9.0g | marstt | marital status |
| livtog | float | %9.0g | livtog | whether living together as a couple |
| married | float | %9.0g | married | whether married/cohabiting |
| inecaca | float | %9.0g | inecaca | economic activity |
| status | float | %9.0g | status | economic status |
| ilodefa | float | %9.0g | ilodefa | basic economic activity (ilo definition) |
| grsswk | float | %9.0g | grsswk | gross weekly pay in main job (£) |
| hourpay | float | %9.0g | hourpay | gross hourly pay (£) |
| soc2km | float | %9.0g | soc2km | occupation (main job) |
| sc2kmmj | float | %9.0g | sc2kmmj | occupation in main job (grouped) |
| nsecm | float | %9.0g | nsecm | ns-sec category (main job) |
| jobtyp | float | %9.0g | jobtyp | permanent or temporary job |
| conmpy | float | %9.0g | conmpy | year started working for current employer |
| ptime | float | %9.0g | ptime | whether part-time (self-reported) |
| ptimehrs | float | %9.0g | ptimehrs | whether part-time (work <31 hours per week) |
| ttushr | float | %9.0g | ttushr | total usual hours in main job per week (including overtime) |
| public | float | %9.0g | public | whether working in public or private sector |
| appren | float | %9.0g | appren | recognised trade apprenticeship |
| schm99 | float | %9.0g | schm99 | type of government employment or training scheme |
| manage | float | %9.0g | manage | managerial duties or supervising |
| secjob | float | %9.0g | secjob | whether had 2nd job in reference week |
| teclec | float | %9.0g | teclec | whether on scheme run by a tec or lec |
| newdeal | float | %9.0g | newdeal | new deal option |
| ytetmp | float | %9.0g | ytetmp | yt, et, tec schemes |
| ytetjb | float | %9.0g | ytetjb | whether had paid job in addition to scheme |
| wrking | float | %9.0g | wrking | whether had paid job in addition to scheme |
| jbaway | float | %9.0g | jbaway | whether temporarily away from paid work |
| ownbus | float | %9.0g | ownbus | whether doing unpaid work for own business |

| | | | | |
|---------|-------|-------|---------|--|
| relbus | float | %9.0g | relbus | whether doing unpaid work for relatives business |
| nstat | float | %9.0g | nstat | employment status in main job |
| look4 | float | %9.0g | look4 | whether looking for paid work in last 4 weeks |
| lkyt4 | float | %9.0g | lkyt4 | whether looking for a place on a government scheme in the last 4 weeks |
| start | float | %9.0g | start | whether could start work within the next 2 weeks |
| wait | float | %9.0g | wait | whether waiting to take up a job |
| likewk | float | %9.0g | likewk | whether would like to work |
| ystart | float | %9.0g | ystart | reason why could not start work within 2 weeks |
| nolook | float | %9.0g | nolook | reason not looking for work (if likewk=1) |
| nowant | float | %9.0g | nowant | reason not looking for work (if likewk=2) |
| hiqual | float | %9.0g | hiqual | highest qualification |
| hiquald | float | %9.0g | hiquald | highest qualification (grouped) |
| levqual | float | %9.0g | levqual | level of highest qualification |
| edage | float | %9.0g | edage | age completed continuous full-time education |
| bhealth | float | %9.0g | bhealth | bad health that limits paid work |
| quart | float | %9.0g | quart | quarter taken from |
| age3 | float | %9.0g | age3 | Age groups |
| jobperm | float | %9.0g | jobperm | * Whether permanent job |

* indicated variables have notes

Sorted by: sex
Note: dataset has changed since last saved

. notes jobperm

jobperm:

1. Same as jobtyp but with 'Does not apply' and 'No answer' set as missing.

If we save the dataset now, the new variables including the notes will be saved.

Stata has functions for the manipulation of missing values. We can turn the `-8` (or any other such value) into missing by using `mvdecode` (as we do not wish to change the data, the command is blocked by `*`):

```
*mvdecode hourpay, mv(-8)
```

If all the variables in the data set had the `-8` and we wished to put them as missing, we could do it by:

```
*mvdecode _all, mv(-8)
```

where `_all` is the Stata internal variable for all the variables in the data set.

To turn missing values into coded values, we could use `mvencode`:

```
*mvencode hourpay, mv(-8)
```

```
*mvencode _all, mv(-8)
```

We might save the changes effected via the `mvdecode` or `mvencode` for the ease of future analysis. You may have to update Stata in order to use these ado files for `mvdecode` or `mvencode`.

2.5 Creating new variables

There are many commands for data management in Stata and the following are some of the most useful. The most frequently used, as already touched upon above, are `generate`, `replace`, `recode`, `egen`, `tab`, `gen()` etc. Some of these can be used in combination with useful effects as shown below.

If, for instance, we find a curvilinear relationship between age and income, we might fit a quadratic model. In such a case, we may need a variable for age squared. We can:

```
generate agesquared=age^2
replace agesquared=. if age <=20 /*assuming we are not interested in the under20s*/
```

The most notable, and useful, feature of `generate` is its ability, especially used in combination with other commands, to create indicator or dummy variables for categorical data. We use educational qualifications as an example. However, this is also a very tough problem as there are 43 categories in the variable among which two are user missing which we first of all put as Stata missing.

```
gen      hieduc=hiqual
replace  hieduc=. if hiqual<1
tab      hieduc,m
```

We then use the most up-to-date version of the educational framework adopted for the 2001 Census as follows (for simplicity, we just code higher degree=1, others=0):

The national qualifications framework

| Level of qualification | General qualification | Vocationally-related qualification | Occupational qualification | Age | |
|------------------------|--|---------------------------------------|----------------------------|------------|----------------------------------|
| 5 | Higher education (e.g. BA, BSc, MA, Ph.D) | | Level 5 NVQ | 19+ | Post- compulsory education |
| 4 | | | Level 4 NVQ | | |
| 3 advanced level | A/AS level | Vocational A level (Advanced GNVQ) | Level 3 NVQ | 17-18 | |
| 2 intermediate level | O level, GCSE grade A-C | Intermediate GNVQ | Level 2 NVQ | 15-16 | Compulsory education |
| 1 foundation level | GCSE grade D-G | Foundations GNVQ | Level 1 NVQ | 15-16 | |
| Entry level | Certificate of (educational) achievement | | | 14 or less | |

We shall, in this case, code those above A/AS levels as higher education (codes 1-16: I may be wrong here!)

```

replace hieduc=1 if hiqual>=1&hiqual<=16
replace hieduc=0 if hiqual>=17&hiqual<=41
lab var hieduc "Higher educational qualifications"
lab def hieduc 1 "Higher" 0 "A level or below"
lab val hieduc hieduc
tab hiqual hieduc, m

```

| highest qualification | Higher educational qualifications | | Total |
|-----------------------|-----------------------------------|--------|-------|
| | A level or | Higher | |
| does not apply | 0 | 0 | 5097 |
| no answer | 0 | 0 | 47 |
| higher degree | 0 | 2719 | 0 |
| nvq level 5 | 0 | 64 | 0 |
| first degree | 0 | 5983 | 0 |
| other degree | 0 | 772 | 0 |
| nvq level 4 | 0 | 270 | 0 |
| diploma in higher edu | 0 | 630 | 0 |
| hnc,hnd,btec etc high | 0 | 2236 | 0 |
| teaching, further edu | 0 | 173 | 0 |
| teaching, secondary e | 0 | 177 | 0 |
| teaching, primary edu | 0 | 263 | 0 |
| teaching, level not s | 0 | 8 | 0 |
| nursing etc | 0 | 1102 | 0 |
| rsa higher diploma | 0 | 45 | 0 |
| other higher educatio | 0 | 344 | 0 |
| nvq level 3 | 0 | 1455 | 0 |
| gnvq advanced | 0 | 428 | 0 |
| a level or equivalent | 4267 | 0 | 0 |
| rsa advanced diploma | 89 | 0 | 0 |
| ond,onc,btec etc, nat | 1238 | 0 | 0 |
| city & guilds advance | 1990 | 0 | 0 |
| scottish csys | 70 | 0 | 0 |
| sce higher or equival | 646 | 0 | 0 |
| a,s level or equivale | 451 | 0 | 0 |
| trade apprenticeship | 4535 | 0 | 0 |
| nvq level 2 or equiva | 1688 | 0 | 0 |
| gnvq intermediate | 329 | 0 | 0 |
| rsa diploma | 96 | 0 | 0 |
| city & guilds craft | 497 | 0 | 0 |

| | | | | |
|-----------------------|-------|-------|------|-------|
| btec,scotvec first/ge | 115 | 0 | 0 | 115 |
| o level, gcse grade a | 11037 | 0 | 0 | 11037 |
| nvq level 1 or equiva | 317 | 0 | 0 | 317 |
| gnvq,gsvq foundation | 47 | 0 | 0 | 47 |
| cse below gradel,gcse | 2188 | 0 | 0 | 2188 |
| btec,scotvec first/ge | 24 | 0 | 0 | 24 |
| scotvec modules | 72 | 0 | 0 | 72 |
| rsa other | 595 | 0 | 0 | 595 |
| city & guilds other | 202 | 0 | 0 | 202 |
| yt,ytp certificate | 75 | 0 | 0 | 75 |
| other qualification | 4771 | 0 | 0 | 4771 |
| no qualifications | 8627 | 0 | 0 | 8627 |
| don't know | 164 | 0 | 0 | 164 |
| ----- | | | | |
| Total | 44130 | 16669 | 5144 | 65943 |

In the file, there is a variable on the year in which the respondent joined in the present organisation (conmpy). We need to first turn it into 'length of service' as it is an important indicator on the level of income. We can then recode the variable in whatever way we wish, for instance, into indicator or dummy variables.

```
gen      lengthcompany=conmpy
replace  lengthcompany=. if conmpy<1
/*to exclude 'does not apply' and 'no answer'*/
gen      lengthcom=2003 - lengthcompany
/*How along the Rs have worked in present company*/
tab      lengthcom,m /*to check*/
tab      lengthcom
```

```
. tab lengthcom
```

| lengthcom | Freq. | Percent | Cum. |
|---|-------|---------|--------|
| 0 | 132 | 0.33 | 0.33 |
| 1 | 5435 | 13.47 | 13.80 |
| 2 | 6348 | 15.74 | 29.54 |
| 3 | 3926 | 9.73 | 39.27 |
| 4 | 3081 | 7.64 | 46.91 |
| 5 | 2478 | 6.14 | 53.05 |
| 6 | 1957 | 4.85 | 57.90 |
| 7 | 1705 | 4.23 | 62.13 |
| 8 | 1379 | 3.42 | 65.55 |
| 9 | 1104 | 2.74 | 68.28 |
| <i>(values in between omitted here)</i> | | | |
| 47 | 6 | 0.01 | 99.96 |
| 48 | 4 | 0.01 | 99.97 |
| 49 | 7 | 0.02 | 99.99 |
| 50 | 6 | 0.01 | 100.00 |
| ----- | | | |
| Total | 40339 | 100.00 | |

Assuming that we wish to have a cut-off point at 10 years. Here are the two ways (you can think of other ways):

```
. gen      lengthcom_a=cond(lengthcom <=10, 1,0) if lengthcom ~=.
(25604 missing values generated)
```

```
. gen      lengthcom_b=lengthcom <=10 if lengthcom ~=.
(25604 missing values generated)
```

```
. tab                lengthcom_a lengthcom_b

lengthcom_ |      lengthcom_b
           a |          0          1 |      Total
-----+-----+-----+-----
           0 |      11963          0 |      11963
           1 |          0      28376 |      28376
-----+-----+-----+-----
           Total |      11963      28376 |      40339
```

In the two commands above, we are asking Stata to `generate` new variables `lengthcom_a` and `lengthcom_b` (similar to the `if` command in SPSS). If the condition is true (namely, `lengthcom <=10`), it is coded by Stata as 1, otherwise as 0 for all observations except those covered in the `if` expression. Note that in the first command, the condition is specifically applied whilst in the second, it is implied. Stata knows the conditional nature of the second command and assigns value 1 for those with values less than or equal to 10 on `lengthcom` and 0 for those with other valid values in the variable. We can also use `generate` followed by `replace` for the same purpose:

```
gen          lengthcom_c=1 if lengthcom <=10
replace      lengthcom_c=0 if lengthcom_c ~=1 & lengthcom ~=.
/*not to include the missing; and we can check them*/
```

We could check whether the coding is correctly done by using the `tabm` command

```
. tabm                lengthcom_*,m
```

When the variables assume the same range of values, it is very economical to use `tabm`, as we do here. It may not be possible to use it on University clusters but you can always download it by using `update all` on your machine.

2.6 Recoding, labelling, dropping, keeping and renaming

For people familiar with SPSS, `recode` in Stata might at first appear inflexible and stubborn, unlike most of the other features we have seen thus far. In SPSS, we have `low`, `hi`, `through`, `else` etc and we have `into`, and we can recode several variables in one go. In Stata, some of these are changed and others not allowed. Let us look at the following example:

Suppose we wish to recode `lengthcom` into four categories. We know already the longest is 50 years and the shortest is 0, namely those who join the organisation in 2003:

```
gen    length4=lengthcom
recode length4 min/5=1 6/10=2 11/20=3 21/50=4 *=.
*      Note: / means through in SPSS; * means else in SPSS
label variable    length4 "Length in present company"
label define      length4 1 "0-5" 2 "6-10" 3 "11-20" 4 "21-50"
label values      length4 length4
```

In actual work, I would simplify the matters so that I would write:

```
gen          length4=lengthcom
recode      length4 min/5=1 6/10=2 11/20=3 21/50=4 *=.
lab var     length4 "Length in present company"
lab def     length4 1 "0-5" 2 "6-10" 3 "11-20" 4 "21-50"
lab val     length4 length4
```

We need to `generate` a new variable to take values from an existing variable and use `recode` on the newly created variable. Stata understands `/` to mean 'through', `*` to mean 'remaining' or 'all others' (similar to 'else' in SPSS). Note the variable and value labelling in Stata. Note also here that we have used the `min` because we know that the smallest value in the variable `lengthcom` is 0 and that we have not used `max` because we also know that the largest value is the missing value which is automatically the largest in Stata usage.

Sometimes it is easier and more efficient to work with a smaller data set containing only the most essential variables needed for the analytical tasks. Stata allows this. One could, for example, do the following (blocked here):

```
* drop age-cry01      /*      to drop variables age to cry01          */
* drop p*            /*      to drop all variables beginnig with p          */
* drop _all          /*      to drop all variables, same as clear          */
```

We could use `keep` instead of `drop` for appropriate purposes.

```
*keep if quart==1|quart==2
*keep if _n>=5000&_n<=5999
*keep sex status married bhealth house hourpay
```

`rename` works the same as SPSS for those familiar with SPSS. That is, `rename x y` would rename `x` into `y`. The variable and value labels in the original variable are kept intact.

```
. rename sex gender
```

```
. tab                gender
```

| sex | Freq. | Percent | Cum. |
|--------|-------|---------|--------|
| male | 31817 | 48.25 | 48.25 |
| female | 34126 | 51.75 | 100.00 |
| Total | 65943 | 100.00 | |

We could use `renvars` which is more powerful but we may have less opportunity to use it.

Compare:

```
* help renvars      /*for a list of variables*/
* help rename       /*for one variable*/
* help renpfix      /*for all variables*/
```

Exercises and suggested answers

- 2.1 Do male and female respondents in the data set have a similar age profile? [*sex age3*]
- 2.2 Do different ethnic men and women have similar hourpay and gross weekly pay? And among the highly educated? Do the data conform to the stereotypes of ethnic advantages and disadvantages of ethnic groups? (for employees only) [*sex ethnic hourpay grsswk hiqual*]
- 2.3 Which ethnic group is most and which is least likely to be foreign born? [*ethnic fb*]
- 2.4 For men aged 30-65 (assuming they have finished education and can, in theory, be fully 'occupied' in work), which ethnic groups get the highest pay? [*age sex ethnic grsswk*]
- 2.5 Which ethnic group are least likely to be owner-occupiers? [*ethnic house*]
- 2.6 Coding whites as one group and all non-whites as the other group, which group is more likely to have long-term limiting illness? And that after controlling for sex? Is the difference in long term limiting illness greater among men or among women? [*sex, bhealth ethn*]

Suggested answers:

*Exercise 2.1

```
sort sex
tab sex, summ(age)
bysort sex: summ age
by sex, sort: su age
table sex, content(mean age)
```

*Exercise 2.2

```
tab ethnic sex if status==1&hourpay>0, summ(hourpay) nost nofr
tab ethnic sex if status==1&grsswk>0, summ(grsswk) nost nofr
table ethnic sex if status==1&hourpay>0, c(mean hourpay) f(%5.2f)
table ethnic sex if status==1&grsswk>0, c(mean grsswk) f(%5.2f)
table ethnic sex if status==1&grsswk>0&hiqual>=1&hiqual<=5, c(mean grsswk) f(%5.2f)
```

*Exercise 2.3

```
tab ethnic fb, row
```

*Exercise 2.4

```
tab ethn, summ(hourpay) ,if (age>=35&age<=65&sex==0&status==1&hourpay>0)
tab ethn, summ(grsswk) ,if (age>=35&age<=65&sex==0&status==1&grsswk>0)
```

*Exercise 2.5

```
tab ethn house if house >0, row
```

*Exercise 2.6

```
gen white=ethn==1
bys sex: tab white bhealth, row
```

Chapter 3 Stata graphs

Stata 7 has extensive facilities for making graphs. In this part, we will only introduce some basic techniques for graph-making (for more sophisticated techniques, please consult the Stata Manual on Graphics).

The syntax for graph is:

```
graph [varlist] [weight] [if exp] [in range] [, options]
graph using filename [filename ...] [,options]
```

The letters underlined means the minimum to type for the command. The `by` prefix may be used with `graph`. There are eight `graph` styles specified as `options` in the first syntax and there are five patterns for line styles.

| <u>Option</u> | <u>Meaning</u> | <u>Comment</u> |
|-------------------------|----------------------------|--|
| <u>h</u> istogram | Histogram | default when one variable is specified |
| <u>t</u> wo <u>w</u> ay | Two-way scatterplot | default when more than one variable is specified |
| <u>m</u> atrix | Two-way scatterplot matrix | up to 30 variables may be specified |
| <u>o</u> ne <u>w</u> ay | One-way scatterplot | may be combined with <code>box</code> |
| <u>b</u> ox | Box-and-whisker plot | up to 6 variables may be specified |
| <u>s</u> tar | Star chart | up to 16 variables may be specified |
| <u>b</u> ar | Bar chart | plot sums or means of variables |
| <u>p</u> ie | Pie chart | plot sums of variables |

In the following, we will focus on five of them: `h`istogram, `t`woway, `b`ox, `b`ar and `p`ie as the other three styles do not seem particularly useful for categorical data as we have for the SARs. We continue to use **LFS2002.dta**.

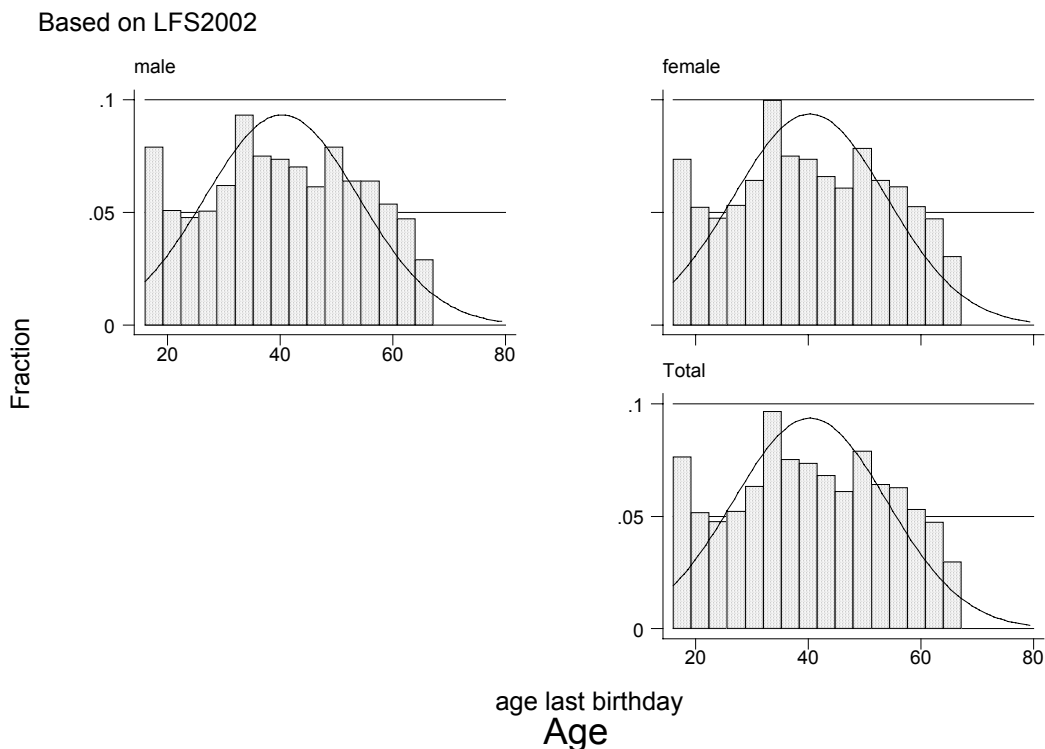
3.1 Histogram

By default, Stata `graph` draws a `h`istogram if there is only one variable specified. In addition to standard options, `h`istogram provides three unique options: `bin()`, `freq`, and `normal`. The `bin()` option sets the number of bins or intervals used to accumulate the frequencies. The default is 5. The `freq` option requests that the y-axis be labelled in frequency units rather than the default fractional units. The `normal` option superimposes a normal curve on the histogram. We can also add titles to the graphs. Let us assume that we want to find out the age profile in the **LFS2002.dta** and whether there are clear sex differences?

```
sort sex
graph age, bin(20) normal xlabel ylabel yline ylabel t1title("Based on LFS2002") /*
*/      title("Age") by(sex) total
```

In the command, we have changed the `bin` to 20 intervals and asked for a normal line. We have also asked the `xlabel`, `ylabel` and `yline` to be shown. Two titles, one at the top (`t1title`) and the other at the bottom (`title`, which is the same as using `b1title`), are added. And we have asked for gender differences to be shown together with an overall picture. Since we

need one graph for males and another for females, we have to `sort` the data first if not sorted already.



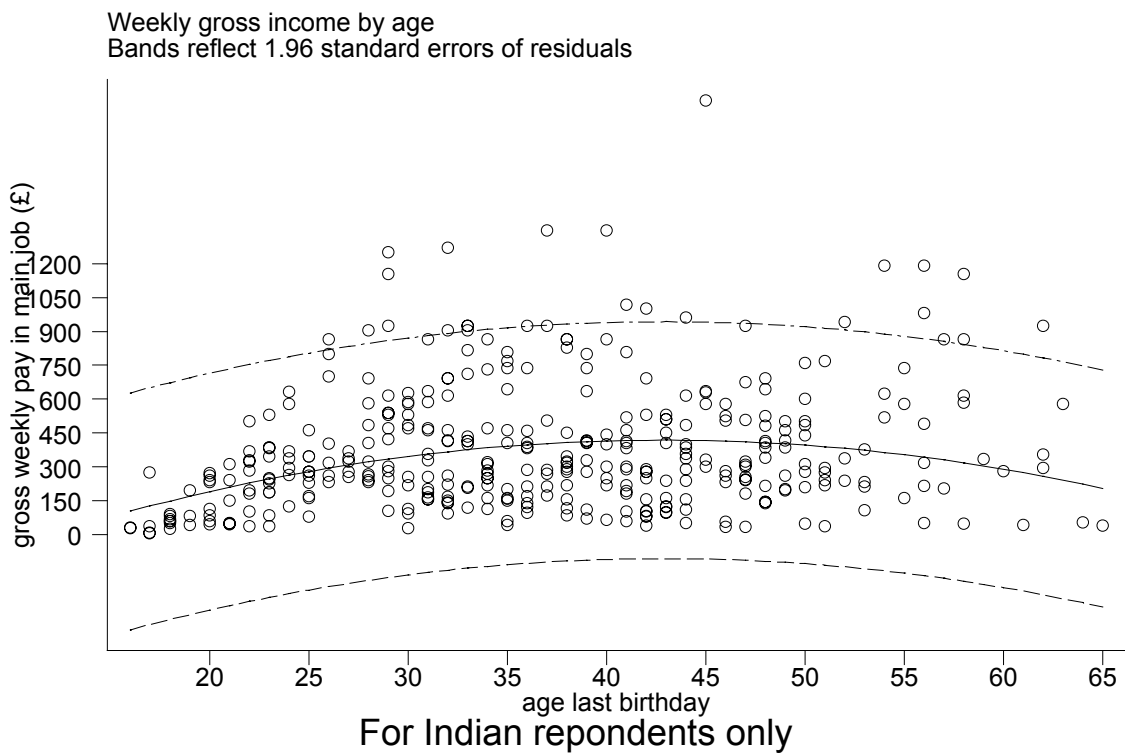
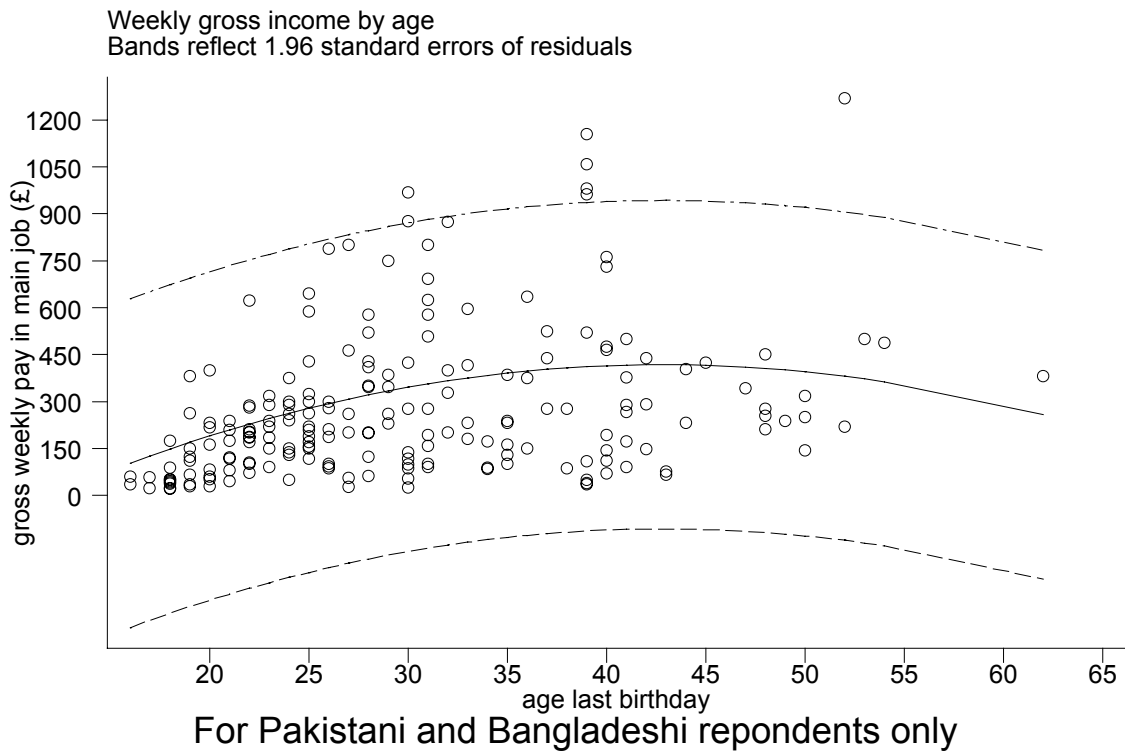
3.2 Two-way graph

If you have two or more continuous variables, you can plot $y_1 \text{ v } x$, $y_2 \text{ v } x$, $y_3 \text{ v } x$, etc. With social surveys it is rather difficult to find continuous variables that form meaningful patterns. The following example serves as an illustration only.

Let us assume that we wish to find out graphically the relationship between weekly gross income (`grsswk`) and age and then decide to do a regression. We can then use the predicted values from the regression model in the graph, values outside the 95% confidence intervals. The following are the commands.

```
gen      agesq=age*age
reg      grsswk age agesq if status==1&grss>0
*income is only recorded for employees
predict  hat          /*predicted values          */
predict  s, stdr      /*standard error of the residuals          */
gen      low = hat-1.96*s
gen      high= hat+1.96*s
graph grsswk hat hig low age if status==1&grsswk>0&(ethni==6|ethn==7) /*
    /*      , c(.11[-.-]1[--]) s(Oiii)          /*
    /*      sort ylab(0 150 to 1250) xlab(20 25 to 65) gap(3)          /*
    /*      t1(Weekly gross income by age)          /*
    /*      t2(Bands reflect 1.96 standard errors of residuals)          /*
    /*      title(For Pakistani and Bangladeshi repondents only)          /*

graph grsswk hat hig low age if status==1&grsswk>0&ethni==5          /*
    /*      , c(.11[-.-]1[--]) s(Oiii)          /*
    /*      sort ylab(0 150 to 1250) xlab(20 25 to 65) gap(3)          /*
    /*      t1(Weekly gross income by age)          /*
    /*      t2(Bands reflect 1.96 standard errors of residuals)          /*
    /*      title(For Indian repondents only)          /*
```



In this graph, we requested Stata to plot *grsswk hat high* and *low* against *age* respectively, and to have two titles at the top of the graph. `c(.11[-.-]1[--])` asked Stata not to connect *grsswk* and *age*, but to connect *hat* with *age* by straight line, to connect *high* and *age* by straight line with `-.-` pattern, and to connect *low* and *age* with `--` pattern. Furthermore, we asked Stata to plot *grsswk* versus *age* with large circles but to plot *hat*, *low* and *high* with age with invisible (`.`). `sort` requests the data be sorted into x-order before graphing. `ylabel(0 100 to 1400)` asks ylabel to start from 0 and end at 1400 with an interval of 100. Similarly, `xlab(20 25 to 65)` asks Stata to start at 20 and stop at 65 with an interval of 5. Finally, with `gap(5)`,

we asked Stata to leave a space of 5 between the left title and the values along the axis. The default is 8.

Now what do we make of the comparison of the two graphs? It seems that the Pakistani and Bangladeshi people's income peak at around 30 and then level off and fall at the age of 40. The Indians' income seems to peak at around age 35 and continue to be high until about age 55. The Indians are doing very well, in terms of education, occupation, income and social integration as compared with Pakistanis and Bangladeshis.

Now follow the example and do it for the other groups, including the Whites.

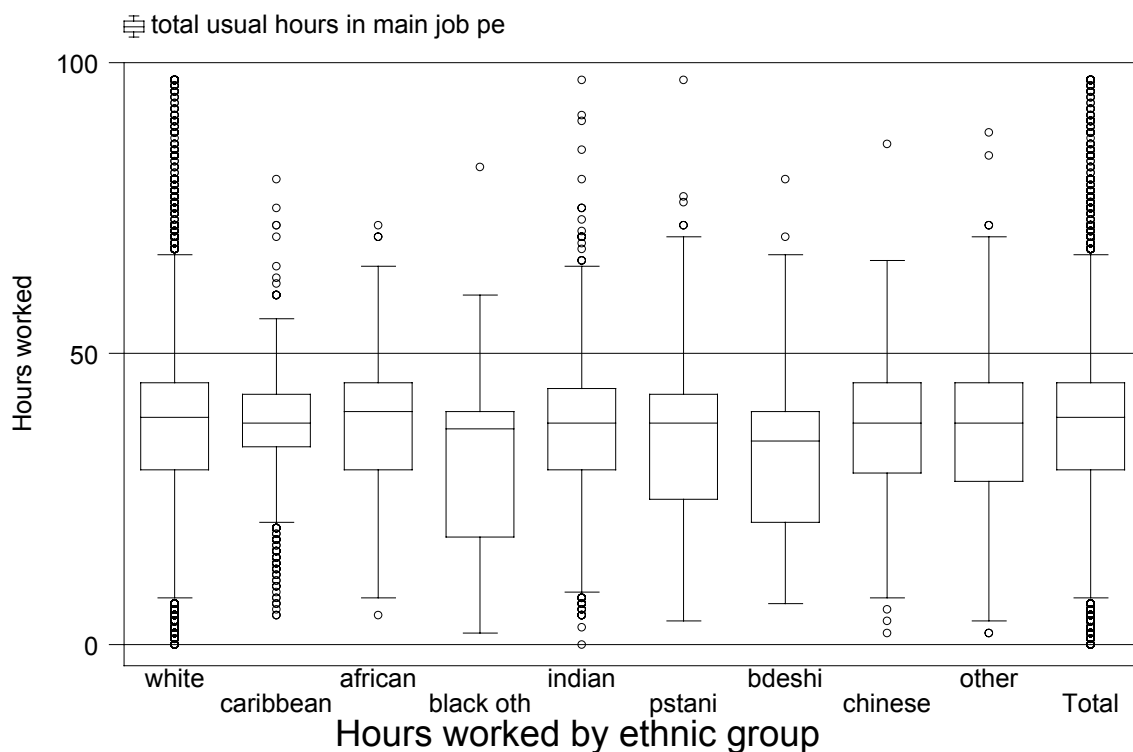
3.3 Box

Let us draw a box plot comparing the number of hours worked between ethnic groups. We need to sort the data by *ethnic* first.

```

sort eth
gr ttushr if ttushr>=0, box by(eth) total yline ylab s(o) /*
                */ti(Hours worked by ethnic group) lltitle(Hours worked) /*
                */gap(4)

```

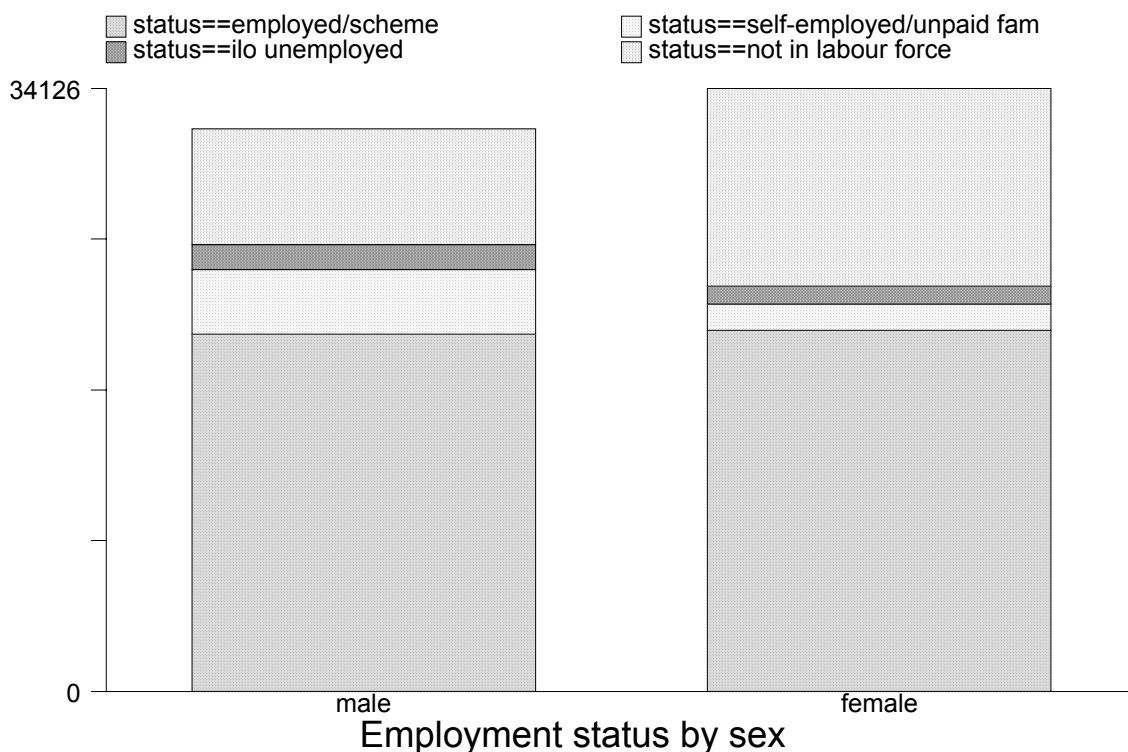


The graph is for people who reported usual hours worked (the -8 -9 omitted). Within each box, the line in the middle represents the median or the 50th percent of the data. The box covers the so-called inter-quartile range (*IQR*) ranging from the 25th percentile ($x_{[25]}$) to the 75th percentile ($x_{[75]}$). The lines emerging from the box are called the whiskers and they extend to the upper and lower adjacent values, defined as $(x_{[75]}) + 1.5 \times IQR$ and $(x_{[25]}) - 1.5 \times IQR$ respectively.

3.4 Bar chart

bar charts for categorical variables can be most easily done by first converting the variables into dummy variables and then asking Stata to graph the dummy variables. The bars can be stacked together, though. For example, what is the relationship between employment status and gender? The data shows that there are many more women than men who are not in labour force.

```
tab status, gen(newstatus) /*to create 5 dummies for employment status */
sort sex
gra news*, bar by(sex) stack ti(Employment status by sex) gap(5)
```



Now that we have used `tab varname, gen(newvar)` to create the dummies. This is a very clever way of getting indicator variables. We know that there are four categories in *status* and four indicator variables will be created, called *newstatus1* *newstatus2* *newstatus3* and *newstatus4*. If you do not believe, we can check it for you:

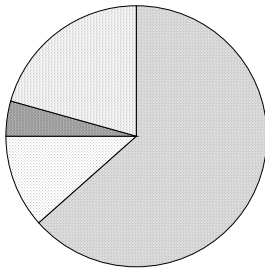
```
. ds
ten96   house   sex     age     ages   nation  cry01   region
numchild numchill ayfl19 ethnic fb     arrival marstt  livtog
married  inecaca  status  ilodefa grsswk hourpay soc2km  sc2kmmj
nsecm    jobtyp   conmpy  ptime  ptimehrs ttushr  public  appren
schm99   manage  secjob  teclec  newdeal ytetmp  ytetjb  wrking
jbaway   ownbus   relbus  nstat  look4   lkyt4   start   wait
likewk   ystart  nolook  nowant  hiqual  hiquald levqual  edage
bhealth  quart   agesq   hat     s       low     high    newsta~1
newsta~2 newsta~3 newsta~4
```

3.5 Pie chart

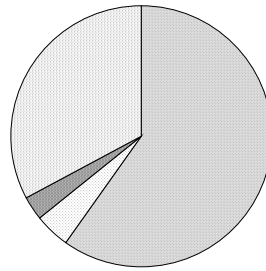
Once we know how to do bar charts, doing pie charts is easy. The following is the same information arranged as pie chart.

gra news*, pie by(sex) total ti(Employment status by sex)

male

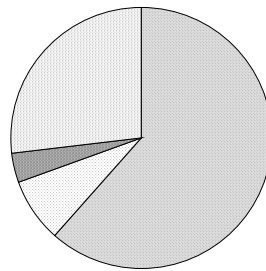


female



- 62% status==employed/schem
- 8% status==self-employed/unemployed
- 4% status==ILO unemployed
- 27% status==not in labour force

Total



Employment status by sex

Exercises and suggested answers

Questions

- 3.1: Show the mean income by sex [sex grsswk].
- 3.2: Show the income and age relationship for whites [grsswk age].

Suggested answers

*Exercise 3.1

```
sort sex
graph grsswk, bin(20) normal xlabel yline ylabel          /*
    */t1title("Based on LFS2002")                       /*
    */title("Income")                                    /*
    */by(sex) total, if (grsswk >=0)
```

*Exercise 3.2

```
graph grsswk hat hig low age if status==1&grsswk>0&ethni==1 /*
    */ , c(.11[-.-]1[--]) s(Oiii)                          /*
    */ sort ylab(0 100 to 1400) xlab(20 25 to 65) gap(5)   /*
    */ t1(Weekly gross income by age)                       /*
    */ t2(Bands reflect 1.96 standard errors of residuals) /*
    */ title(For White repondents only)
```

Chapter 4 Statistical modelling: a brief introduction

4.1 Statistical models: a typology

Some techniques are more frequently used in social science research (sociology in particular) than others. For instance, when we read sociological books or papers, we often see logistic regressions but seldom see topological (or levels matrix) models. The techniques used are related both to the research questions at hand and to the nature of the dependent variables available. The following are some of the most frequently used techniques and we shall give a very brief introduction here.

| Types of dependent variables | | | | |
|-------------------------------|-------------------|---------------------|------------------------------|--------------------------|
| | Continuous | Binary | Multinomial | Ordinal |
| Examples | Income | Long-term illness | Employment status | Levels of schooling |
| Modelling techniques involved | Linear regression | Logistic regression | Multinomial logit regression | Ordinal logit regression |
| Stata techniques | regress | logistic/logit | mlogit | ologit |

Type of independent variables: any

4.1 An example of multiple linear regression

When we have a continuous variable as outcome (also called dependent) variable, it is preferable to use the linear regression technique. We can include many explanatory (also called independent) variables, both categorical and continuous.

The most basic structure of regression is:

```
regress depvar [varlist] [weight] [if exp] [in range] [, level(#) beta robust
cluster(varname) noconstant noheader]
```

by ... : may be used with regress; see help by.
 aweights, fweights, iweights, and pweights are allowed; see help weights.

Since we do not have weighting variables in our dataset we can not use any weights. Now let us consider some research questions:

- 1: Which ethnic groups are more likely to have higher incomes?
- 2: Is it related to gender and marital status?
- 3: Do foreign-born people suffer from nativity penalty?

We have used income data quite a lot in descriptive analysis in the above but did not put it to any 'formal' test because we could not include many theoretically meaningful explanatory variables. Suppose we wish to analyse this again, this time as a function of some socio-

demographic attributes such as those mentioned above. Note that we can only do this for the employees with no missing income data. Our three questions are translated into three models:

Model 1

$$\text{income} = \beta_0 + \beta_1 \text{ethnic} + \varepsilon$$

Model 2

$$\text{income} = \beta_0 + \beta_1 \text{ethnic} + \beta_2 \text{sex} + \beta_3 \text{married} + \varepsilon$$

Model 3

$$\text{income} = \beta_0 + \beta_1 \text{ethnic} + \beta_2 \text{sex} + \beta_3 \text{married} + \beta_4 \text{foreignborn} + \varepsilon$$

As with any data analysis, the first step is data preparation. We know that sex, married and fb are already coded as indicator (dummy) variables: sex (female = 1, male = 0); married (married = 1, non-married = 0) and fb (foreign born = 1 and native born = 0). But we need to create indicator variables for ethnic groups. We have already learned a few ways, such as `generate` and `replace`, `tab var`, `gen(newvar)`. In this case, we need to create the dependent variable, and to turn the independent variables into dummies. Here we introduce another way which is most useful in modelling techniques. Suppose that we wish to use White as the reference groups.

```
char ethn1[omit] 1
```

The `char variablename[omit] #` function turns categorical variables into indicator (dummy) variables. The `#` is the category which we theoretically wish to use as the base (also called comparison or omitted or reference) group. We can decide to omit whichever group which we prefer. There are two points here: 1: the omitted category is usually the sociologically meaningful category. For instance, to use the ‘Other’ category may not be sociologically meaningful because we do not have a clear idea who the ‘Other’ group are and what sociological theories are associated with the group. 2: The reference group should have a fairly large size, otherwise, it would affect the stability of the models. How large is large? It depends. Anyway, if the size is too small, Stata will complain and we need to reconsider choosing another category as the reference category.

When we have indicator variables in linear or other kinds of regression, we need to use `xi:` at the beginning of the programme. The `xi:` function tells Stata that there are categorical variables in the model and when it sees a variable preceded by `i.` it will treat it as a categorical variable depending on which category we have declared as the base group.

```
. *
. xi: regress grsswk i.ethnic if status==1&grsswk>=0
i.ethnic      _Iethnic_1-9      (naturally coded; _Iethnic_1 omitted)
```

| Source | SS | df | MS | Number of obs = 31568 | | |
|----------|------------|-------|------------|-----------------------|---|--------|
| Model | 3068172.75 | 8 | 383521.594 | F(8, 31559) | = | 4.94 |
| Residual | 2.4508e+09 | 31559 | 77656.2294 | Prob > F | = | 0.0000 |
| | | | | R-squared | = | 0.0013 |
| | | | | Adj R-squared | = | 0.0010 |
| Total | 2.4538e+09 | 31567 | 77733.7446 | Root MSE | = | 278.67 |

| grsswk | Coef. | Std. Err. | t | P> t | [95% Conf. Interval] | |
|------------|-----------|-----------|-------|-------|----------------------|-----------|
| _Iethnic_2 | -23.72905 | 17.73292 | -1.34 | 0.181 | -58.48626 | 11.02816 |
| _Iethnic_3 | -5.981017 | 21.49624 | -0.28 | 0.781 | -48.11448 | 36.15245 |
| _Iethnic_4 | -118.1102 | 31.39386 | -3.76 | 0.000 | -179.6434 | -56.57704 |
| _Iethnic_5 | 25.67318 | 14.46086 | 1.78 | 0.076 | -2.670672 | 54.01702 |

| | | | | | | | |
|------------|--|-----------|----------|--------|-------|-----------|-----------|
| _Iethnic_6 | | -56.42616 | 22.96276 | -2.46 | 0.014 | -101.4341 | -11.41826 |
| _Iethnic_7 | | -115.4301 | 40.25447 | -2.87 | 0.004 | -194.3304 | -36.52978 |
| _Iethnic_8 | | 15.2324 | 36.01186 | 0.42 | 0.672 | -55.35227 | 85.81706 |
| _Iethnic_9 | | 34.13716 | 14.29295 | 2.39 | 0.017 | 6.122419 | 62.1519 |
| _cons | | 349.0343 | 1.607448 | 217.14 | 0.000 | 345.8836 | 352.1849 |

We find that, when only ethnic groups are considered in the model, then Black Others, Pakistanis and Bangladeshis get significantly less income than Whites and that the ‘Other’ group get significantly higher incomes.

```

*           Model 2
xi: regress grsswk i.ethni sex married if status==1&grsswk>=0
xi: regress grsswk i.ethni i.sex i.married if status==1&grsswk>=0

. xi: regress grsswk i.ethni i.sex i.married if status==1&grsswk>=0
i.ethnic      _Iethnic_1-9      (naturally coded; _Iethnic_1 omitted)
i.sex         _Isex_0-1        (naturally coded; _Isex_0 omitted)
i.married     _Imarried_0-1    (naturally coded; _Imarried_0 omitted)

-----+-----
Source |           SS          df           MS              Number of obs =   31568
-----+-----+-----+-----
Model | 320155308          10    32015530.8          F( 10, 31557) =   473.51
Residual | 2.1337e+09    31557    67613.0751          Prob > F      =   0.0000
-----+-----+-----+-----
Total | 2.4538e+09    31567    77733.7446          R-squared     =   0.1305
                                           Adj R-squared =   0.1302
                                           Root MSE     =   260.03

-----+-----
grsswk |           Coef.      Std. Err.      t    P>|t|     [95% Conf. Interval]
-----+-----+-----+-----+-----
_Iethnic_2 | -1.831729      16.55295     -0.11   0.912    -34.27617     30.61271
_Iethnic_3 | -5.290509      20.05968     -0.26   0.792    -44.60826     34.02725
_Iethnic_4 | -79.97105      29.30674     -2.73   0.006    -137.4134    -22.5287
_Iethnic_5 | 17.44311       13.49736      1.29   0.196     -9.012241    43.89846
_Iethnic_6 | -71.83893      21.42786     -3.35   0.001    -113.8384    -29.8395
_Iethnic_7 | -151.5194      37.56624     -4.03   0.000    -225.1507    -77.8881
_Iethnic_8 | 12.12303       33.60301      0.36   0.718    -53.74018     77.98624
_Iethnic_9 | 33.56038       13.3376      2.52   0.012      7.418158     59.7026
_Isex_1 | -188.2034      2.930134    -64.23   0.000    -193.9466    -182.4602
_Imarried_1 | 66.28188       2.984302     22.21   0.000      60.43253     72.13123
_cons | 406.6536       2.795909    145.45   0.000      401.1735     412.1337
-----+-----

```

Note that since sex and married are coded as dummies, using i. or not using i. before the two variables produce exactly the same results. Now, in Model 2, our reference group has changed: White women who are not married. We find that women get significantly lower incomes and married get significantly higher incomes holding constant the ethnic factor. Furthermore, compared with the results in Model 1, we find that the relative advantage or disadvantage of the ethnic groups makes little impact.

One important question in this regard is: why do we need to include these additional variables into the model? The reason is two fold: theoretical and statistical. Theoretically, we can expect the patterns to be in the way they appear. Statistically, how do we know whether the additional terms offer a significant improvement in fit? Well, we can.

```

. testparm      _Is* _Ima*

( 1)  _Isex_1 = 0.0
( 2)  _Imarried_1 = 0.0

```

```

F( 2, 31557) = 2344.87
Prob > F = 0.0000

```

The command `testparm` is to test the parameters after fitting the model. It is very useful for testing any of the terms in the model which are significant. In this conjunction, we need to know whether sex and marital status make a significant contribution and we include them. We see that they are highly significant. You can use it after `logit`, `logistic`, `mlogit` and `ologit`. Note that by inspecting the output, we know `_Isex_1` stands for sex and `_Imarried_1` for married and that there are no other terms beginning with s or m so that we can simply use `_Is*` and `_Im*` to stand for the two terms respectively.

```

. *
. xi: regress grsswk i.ethni sex married fb if status==1&grsswk>=0
i.ethnic          _Iethnic_1-9          (naturally coded; _Iethnic_1 omitted)

-----+-----
Source |           SS          df           MS              Number of obs =   31568
-----+-----
Model |  327352435          11   29759312.3          F( 11, 31556) =   441.62
Residual | 2.1265e+09   31556   67387.1429          Prob > F      =    0.0000
-----+-----
Total | 2.4538e+09   31567   77733.7446          R-squared     =    0.1334
                                           Adj R-squared =    0.1331
                                           Root MSE    =   259.59

-----+-----
grsswk |           Coef.      Std. Err.      t    P>|t|     [95% Conf. Interval]
-----+-----
_Iethnic_2 | -26.30005      16.69402     -1.58  0.115    -59.02099     6.420884
_Iethnic_3 | -56.19266      20.62295     -2.72  0.006    -96.61445    -15.77087
_Iethnic_4 | -87.46918      29.26673     -2.99  0.003   -144.8331    -30.10525
_Iethnic_5 | -22.66187      14.02247     -1.62  0.106    -50.14646     4.822708
_Iethnic_6 | -102.5066      21.59687     -4.75  0.000   -144.8373    -60.17588
_Iethnic_7 | -187.8878      37.66817     -4.99  0.000   -261.7189   -114.0567
_Iethnic_8 | -33.2174       33.83249     -0.98  0.326    -99.5304     33.0956
_Iethnic_9 | -11.09539      13.99887     -0.79  0.428   -38.53372     16.34295
sex | -188.3054      2.925251    -64.37  0.000   -194.039    -182.5718
married |  65.41331      2.980497     21.95  0.000    59.57142     71.2552
fb |  67.33948      6.515964     10.33  0.000    54.56794     80.11103
_cons |  404.2611      2.800818     144.34  0.000    398.7714     409.7508
-----+-----

```

We find, surprisingly, that foreign born people (`fb==1`) have significantly higher incomes than native born and that the sex marital parameters are similar to those in Model 2. Once sex, marital and nativity factors are controlled for, all ethnic groups have significantly lower incomes than Whites. Does the inclusion of `fb` make a statistically significant contribution to the terms already included in Model 2?

```

. testparm          fb

( 1)  fb = 0.0

F( 1, 31556) = 106.80
Prob > F = 0.0000

```

The answer is that yes, it does. Now what if you say: well, all this is very good, but I am not really interested in this; what I really want to know is whether, in this model, there are statistically significant differences between particular groups (not between each group and the Whites!). Thus we may wish to see whether there are significant differences between Black Caribbeans and Black Others, between Indians and Pakistanis and between Pakistanis and Bangladeshis, for example.

Well, we can do that. For this, we need to use the function test (the same can be used for logit, logistic, mlogit and ologit, which are most useful sociological models).

```
. test          _Iethnic_2=_Iethnic_4
( 1)  _Iethnic_2 - _Iethnic_4 = 0.0
      F( 1, 31556) =    3.32
      Prob > F =    0.0684

. test          _Iethnic_5=_Iethnic_6
( 1)  _Iethnic_5 - _Iethnic_6 = 0.0
      F( 1, 31556) =   10.03
      Prob > F =    0.0015

. test          _Iethnic_6=_Iethnic_7
( 1)  _Iethnic_6 - _Iethnic_7 = 0.0
      F( 1, 31556) =    3.92
      Prob > F =    0.0477
```

By now, I hope that you know which ethnic groups they are referring to. If not, use `tab ethn` to find out. Interestingly, controlling for gender, marital status and country of birth, there are no significant differences between Black Caribbeans and Black Others, but Indians get significantly more than Pakistanis and Pakistanis get significantly more than Bangladeshis. Note that Stata is also very clever. When we tell it to compare `_Iethnic_2=_Iethnic_4`, it automatically re-orders the equation so that it becomes `_Iethnic_2 - _Iethnic_4 = 0.0`.

We have, in this chapter, given a brief introduction of linear regression. We hope that the techniques involved go a long way, including statistical modelling with binary, polynomial and ordinal data.

Task: suppose the three models above are your own research and you wish to present it in one table in a paper to a journal. How are you going to do it, including all the supplementary statistics that we have put into the text above?

Appendix 1 Stata syntax used in the text

```
*          Stata for LFS2002 (July 2003)
*          Dr Yaojun Li
*          CCSR, Manchester University

*****
*          Chapter 1: Introduction to LFS and Stata          *
*****

help          logistic

search        correspondence analysis

search        William Gould

net search    William Gould

findit        Andrew Pickles

quest on
quest off

*set          varlabelpos 8
* this may not be available in university clusters but you can download it

clear
set           more off
set           mem 20m
use           "C:\DATA_Stata\Course4_Stata_for_LFS\LFS2002.dta", clear

display      (57.23-3.21)/(12.8+4.56)
displa       1.5e+3
displ        1.5e-3
disp         log(250)
dis          ln(250)
di           log10(250)
display      exp(3.6)
display      sqrt(2*log(100))/(3^2-7)
display      chiprob(2, 6.45)
display      _N

/*
browse _all

browse sex - nation
browse sex age - nation

desc
ds           s*
browse s*
bro sex status soc2km sc2kmmj schm99 secjob start
browse age in 30005
browse age status in 30005/30008, nolabel
*/

list         age in 30005
list         age status in f/5
list         age status in -4/1
list         age status in 12570/12575

*codebook
*codebook    _all

codebook     status
```

```

*describe
describe
d

ds

lookfor      class
lookfor      employment status

tab          ethnic
tab          ethnic, nol
tab          fb
tab          fb,nolabel
tab          sex
tab          sex,nolabel
count       if ethnic==6&sex==1&fb==1&age>=25&age<=40

*Exercise 1.1
list        age sex married status house bhealth in 10020/10025
list        age sex married status house bhealth in 10020/10025, nolabel

*Exercise 1.2
list        ethnic in 10026, nolabel
tab         ethnic

*Exercise 1.3
tab        status if sex==1&ethni==7

*Exercise 1.4
tab        ptime if sex==1&ethni==6

*Exercise 1.5
tab        ethn house,r

*Exercise 1.6
lookfor    hourpay

*****
*          Chapter 2: Exploration of LFS          *
*****

summarize   hourpay if status==1&hourpay>=0
su          hour if status==1, detail

count      if stat==1&hourpay==9

tab         sex, m
tab         eth sex, r m
tab2        sex married fb, r m
bys sex:    tab ethn fb,r

tab1        eth sex fb, m
*tabm       quart eth status house bhealth
for var eth status house bhealth: tab X sex, col

tabulate    married sex if (status==1&hourpay>0), summ(hourpay) means nofreq

gen         age3=age
recode age3 min/35=1 36/50=2 51/max=3
label var   age3 "Age groups"
label def   age3 1 "16-35" 2 "36-50" 3 "51-65"
label val   age3 age3
tab         age age3, m

bys age3:   tabulate married sex if (status==1&hourpay>0), summ(hourpay) means nofreq

```

```

table married sex if (status==1&hourpay>0), c(mean hourpay) format(%9.2f)
bys age3: table married sex if (status==1&hourpay>0), c(mean hourpay) format(%9.2f)
table married age3 sex if (status==1&hourpay>0), c(mean hourpay) format(%9.2f)
bys fb: table married age3 sex if (status==1&hourpay>0), c(mean hourpay)
format(%9.2f)

/*
by sex:      summ(hourpay)
*/

bys sex:      summ(hourpay)

bys sex:      tab ethn bhealth, row

bys sex:      table married ethnic if grsswk>0, c(mean age mean grsswk) format(%9.2f)

tab          jobtyp
tab          jobtyp, m

gen          jobperm=jobtyp
replace     jobperm=. if jobtyp<1
replace     jobperm=0 if jobtyp==2
lab var     jobperm "Whether permanent job"
lab def     jobperm 1 "Permanent" 0 "Non-perm"
lab val     jobperm jobperm
tab         jobtyp jobperm,m
notes jobperm:      Same as jobtyp but with 'Does not apply' and 'No answer' set as
missing.

desc
notes jobperm

gen          hieduc=hiqual
replace     hieduc=. if hiqual<1
tab         hieduc,m
replace     hieduc=1 if hiqual>=1&hiqual<=16
replace     hieduc=0 if hiqual>=17&hiqual<=41
lab var     hieduc "Higher educational qualifications"
lab def     hieduc 1 "Higher" 0 "A lever or below"
lab val     hieduc hieduc
tab         hiqual hieduc, m

gen          lengthcompany=conmpy
replace     lengthcompany=. if conmpy<1
/*to exclude 'does not apply' and 'no answer'*/
gen         lengthcom=2003 - lengthcompany
/*How along the Rs have worked in present company*/
tab         lengthcom,m
tab         lengthcom
gen         lengthcom_a=cond(lengthcom <=10, 1,0) if lengthcom ~=.
gen         lengthcom_b=lengthcom <=10 if lengthcom ~=.
tab         lengthcom_a lengthcom_b

gen         lengthcom_c=1 if lengthcom <=10
replace     lengthcom_c=0 if lengthcom_c ~=1 & lengthcom ~=.
/*not to include the missing; and we can check them*/
tabm       lengthcom_*,m

gen         length4=lengthcom
recode     length4 min/5=1 6/10=2 11/20=3 21/50=4 *=.
lab var     length4 "Length in present company"
lab def     length4 1 "0-5" 2 "6-10" 3 "11-20" 4 "21-50"
lab val     length4 length4
tab         lengthcom length4,m

rename sex gender

```

```

tab          gender

rename gender sex

*Exercise 2.1
sort        sex
tab         sex, summ(age)
bysort     sex: summ age
by sex, sort: su age
table      sex, content(mean age)

*Exercise 2.2
tab ethnic  sex if status==1&hourpay>0, summ(hourpay) nost nofr
tab ethnic  sex if status==1&grsswk>0, summ(grsswk) nost nofr
table ethnic sex if status==1&hourpay>0, c(mean hourpay) f(%5.2f)
table ethnic sex if status==1&grsswk>0, c(mean grsswk) f(%5.2f)
table ethnic sex if status==1&grsswk>0&hiqual>=1&hiqual<=5, c(mean grsswk) f(%5.2f)

*Exercise 2.3
tab ethnic  fb, row

*Exercise 2.4
tab ethn,   summ(hourpay) ,if (age>=35&age<=65&sex==0&status==1&hourpay>0)
tab ethn,   summ(grsswk) ,if (age>=35&age<=65&sex==0&status==1&grsswk>0)

*Exercise 2.5
tab ethni house if house >0, row

*Exercise 2.6
gen         white=ethn==1
bys sex: tab white bhealth,row

```

```

*****
*                               *
*           Chapter 3: Stata Graphics           *
*****

```

```

sort sex
graph age, bin(20) normal xlabel yline ylabel t1title("Based on LFS2002")
/*
*/ title("Age") by(sex) total

gen      agesq=age*age
reg      grsswk age agesq if status==1&grss>0
/*income is only recorded for employees */
predict  hat                               /*predicted values                */
predict  s, stdr                            /*standard error of the residuals      */
gen      low = hat-1.96*s
gen      high= hat+1.96*s
graph grsswk hat hig low age if status==1&grsswk>0&(ethni==6|ethn==7) /*
*/      , c(.11[-.-]11[--]) s(Oiii) /*
*/      sort ylab(0 100 to 1400) xlabel(20 25 to 65) gap(5) /*
*/      t1(Weekly gross income by age) /*
*/      t2(Bands reflect 1.96 standard errors of residuals) /*
*/      title(For Pakistani and Bangladeshi repondents only)

graph grsswk hat hig low age if status==1&grsswk>0&ethni==5 /*
*/      , c(.11[-.-]11[--]) s(Oiii) /*
*/      sort ylab(0 100 to 1400) xlabel(20 25 to 65) gap(5) /*
*/      t1(Weekly gross income by age) /*
*/      t2(Bands reflect 1.96 standard errors of residuals) /*
*/      title(For Indian repondents only)

```

```

sort eth
gr ttushr if ttushr>=0, box by(eth) total yline ylab s(o) /*
                */ti(Hours worked by ethnic group) l1title(Hours worked) /*
                */gap(4)

/*          collapse the ethnic groups
gen          eth5=ethn
recode eth5 1=1 2/4=2 5=3 6 7=4 8 9=5
lab var      eth5 "Collapsed ethnic groups"
lab def      eth5 1 "White" 2 "Black" 3 "Indian" 4 "Pak-Bang" 5 "ChnsOth"
lab val      eth5 eth5
tab eth5,    gen(neth)          /*to create 5 dummies for ethnic groups */
*/

tab status,  gen(newstatus)     /*to create 5 dummies for employment status
                */
sort sex
gra news*,   bar by(sex) stack ti(Employment status by sex) gap(5)

gra news*,   pie by(sex) total ti(Employment status by sex)

*Exercise 3.1
sort sex
graph grsswk, bin(20) normal xlabel yline ylabel /*
                */t1title("Based on LFS2002") /*
                */title("Income") /*
                */by(sex) total, if (grsswk >=0)

*Exercise 3.2
graph grsswk hat hig low age if status==1&grsswk>0&ethni==1 /*
                */ , c(.ll[-.-]l[--]) s(Oiii) /*
                */ sort ylab(0 100 to 1400) xlab(20 25 to 65) gap(5) /*
                */ t1(Weekly gross income by age) /*
                */ t2(Bands reflect 1.96 standard errors of residuals) /*
                */ title(For White repondents only)

*****
*          Chapter 4: Statistical modelling: a brief introduction          *
*****

char ethnic[omi] 1
*          Model 1
xi: regress grsswk i.ethni if status==1&grsswk>=0

*          Model 2
xi: regress grsswk i.ethni sex married if status==1&grsswk>=0
xi: regress grsswk i.ethni i.sex i.married if status==1&grsswk>=0

testparm    _Is* _Ima*

*          Model 3
xi: regress grsswk i.ethni sex married fb if status==1&grsswk>=0
tab hiq fb if hiq>=1,col
table fb if school>=0,c(mean school)

testparm    fb

test        _Iethnic_2=_Iethnic_4
test        _Iethnic_5=_Iethnic_6
test        _Iethnic_6=_Iethnic_7

```

Appendix 2 Data entry into Stata

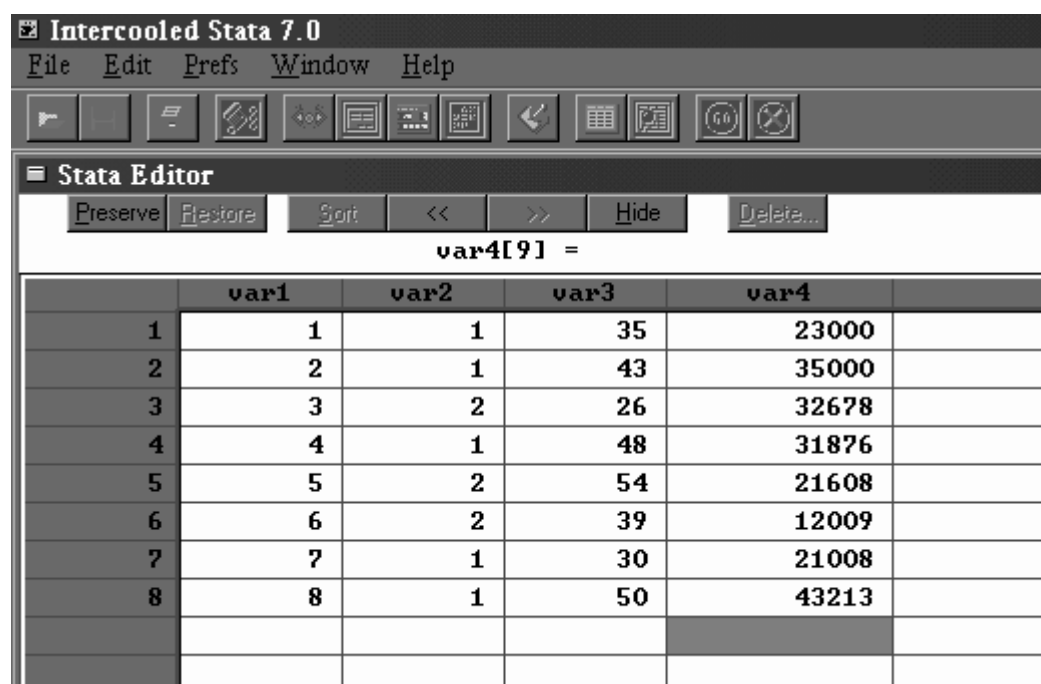
Sometimes it may be necessary to put data from other sources into Stata. This happens when we have new data from our own work or data from other formats. This can involve:

- (1) how to input the data directly in Stata;
- (2) how to get the data of other formats into Stata format and use them;
- (3) how to use StatTransfer

Appendix 2.1 Entry the data directly in Stata

Suppose we have raw data from a survey sample. We could enter them into SPSS format or Excel format. Or we can enter them directly in Stata.

Go to Data Editor (not browse editor). Put the cursor in row 1 and column 1 and begin to enter data. In the following, I have entered a hypothetical dataset composed only of four variables and eight records. Stata will insert the variables for us:



The screenshot shows the Stata 7.0 Data Editor window. The title bar reads "Intercooled Stata 7.0". The menu bar includes "File", "Edit", "Prefs", "Window", and "Help". Below the menu bar is a toolbar with various icons. The main window is titled "Stata Editor" and contains a table with the following data:

| | var1 | var2 | var3 | var4 | |
|---|------|------|------|-------|--|
| 1 | 1 | 1 | 35 | 23000 | |
| 2 | 2 | 1 | 43 | 35000 | |
| 3 | 3 | 2 | 26 | 32678 | |
| 4 | 4 | 1 | 48 | 31876 | |
| 5 | 5 | 2 | 54 | 21608 | |
| 6 | 6 | 2 | 39 | 12009 | |
| 7 | 7 | 1 | 30 | 21008 | |
| 8 | 8 | 1 | 50 | 43213 | |
| | | | | | |
| | | | | | |

We can see that, by default, the variables are called var1 var2 var3 var4. In real situations, there would be many more variables and records, of course.

We can rename the variables all in one go (suppose we have intended the variables to be so):

```
. renvars v*\id sex age income
```

And the data now look like this:

The screenshot shows the Stata 7.0 interface. The main window displays a data browser for a file named 'var5[1]'. The data is presented in a table with the following columns: 'id', 'sex', 'age', 'income', and an empty column. The rows contain the following data:

| | id | sex | age | income | |
|---|----|-----|-----|--------|--|
| 1 | 1 | 1 | 35 | 23000 | |
| 2 | 2 | 1 | 43 | 35000 | |
| 3 | 3 | 2 | 26 | 32678 | |
| 4 | 4 | 1 | 48 | 31876 | |
| 5 | 5 | 2 | 54 | 21608 | |
| 6 | 6 | 2 | 39 | 12009 | |
| 7 | 7 | 1 | 30 | 21008 | |
| 8 | 8 | 1 | 50 | 43213 | |

After inputting the data, we could do the analysis or save it as a file in Stata format. It is always good practice to `compress` the data within Stata before saving.

```
. compress
. save "C:\Data\Input_data_example", replace
(note: file C:\Data\Input_data_example.dta not found)
file C:\Data\Input_data_example.dta saved
```

Appendix 2.2 Import files of other formats into Stata.

Suppose that we have a csv file called `CVS_Example` in the directory “C:\Data\”, we can, in Stata, use the following command to import the data and then analyse them:

```
clear
set mem 30m
set more off
* to have double rather than the float default to save memory
insheet using "C:\DATA\CSV_Example.csv", double
```

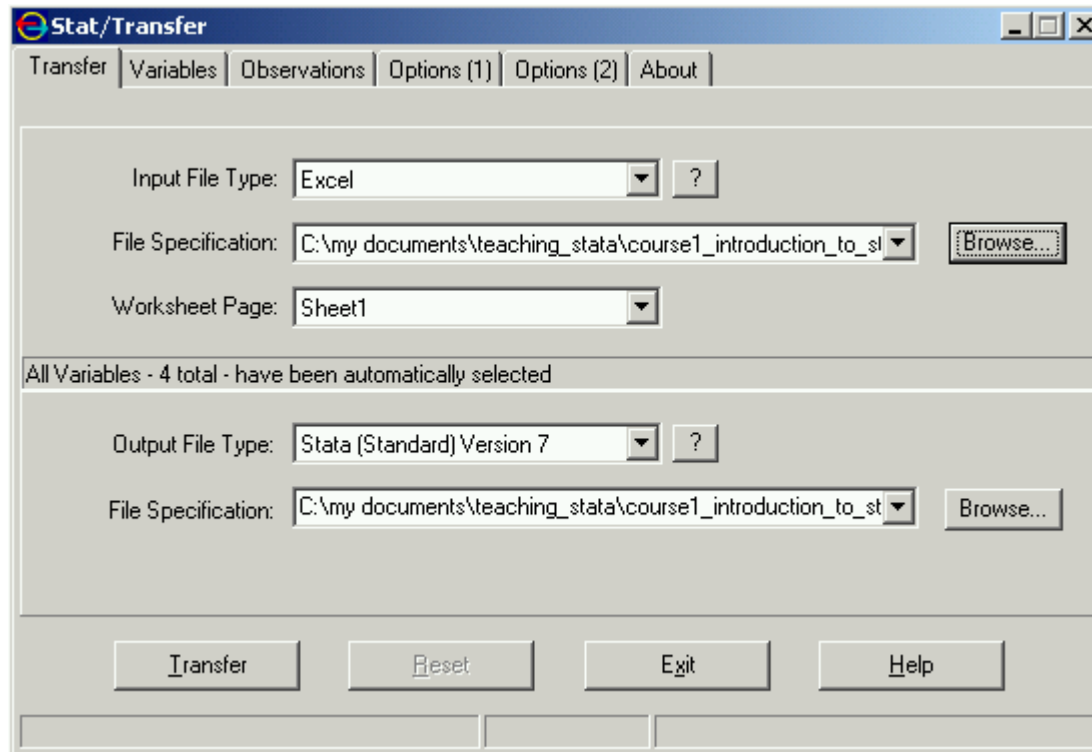
Suppose, again, that we have a file in `dat` or `raw` format stored in a different directory, we can use

```
cd c:\data\sophia\Chapter5\
* suppose that we only wish to get four variables from slim.dat and
* the first is a string variable which we allow to have 20 characters
infile str20 name cond status resp using slim.dat
```

Appendix 2.3 Using StatTransfer

The current version is StataTransfer7 which I do not have. However, the basic principle is the same. We can convert many kinds of file into Stata format (around 26 formats using StatTransfer6). It is, therefore, best to have StatTransfer on the machine. StatTransfer6 can transfer Stata 7 dta files onto SPSS and vice versa and StatTransfer7 can transfer Stata 8 files onto SPSS and vice versa.

Now I use StatTransfer 6 to convert the same data saved as xls format into Stata.



The most frequent use of StatTransfer may be to import SPSS portable files into Stata. We can do it using StatTransfer just as we do other kinds of file. StatTransfer will give the same directory and file name but will change the xls or por file to dta file format.



Economic and Social Data Service

ESDS Government
Economic and Social Data Service
Cathie Marsh Centre for Census and Survey Research
University of Manchester
Manchester M13 9PL

Email: govsurveys@esds.ac.uk
Tel: +44 (0)161 275 1980
Fax: 0161 275 4722
www.esds.ac.uk/government