Statistical Modelling With Stata

Mark Lunt

Contents

1. Essentials of the Stata Language

1.1. Introduction

There are currently over 2000 commands in the stata language. Fortunately, nearly all of them relate to particular kinds of data analysis, and any given individual only needs to know a tiny fraction of them. However, there are a small number of commands and basic concepts that relate to data manipulation and management, which it is essential for all users to be familiar with. This document gives a basic introduction to these essential concepts and commands.

The most important aspect of stata to come to terms with is the fact that it is command-based. There are graphical shortcuts to almost all of the available commands, but you can only use stata really efficiently if you think of a stata session as a series of commands. There are three main reasons why this is important:

1. You must be able to reproduce your analysis. If your analysis consists of a series of commands, they can be stored as a script (called a *do-file* in stata) and rerun any time with a single command. Reproducing an analysis by pointing and clicking is at least as much work as performing the analysis in the first place: if you forget which options you selected and which variables you adjusted for you cannot reproduce your original results.

2. It is easier to produce an analysis that is nearly, but not quite the same. You may want to repeat your analysis after excluding certain subjects, or adjusting for an additional confounder that you forgot about in your original analysis. Again, with a do-file this is very straightforward: using point-and-click software it is not.

3. Finally, using a command line is much quicker *once you are familiar with it*. You can type `reg ht wt` far quicker than you can search through menus to select linear regression of ht on wt, as long as you know that `reg ht wt` is the command you need. However, since most of your analysis will be done using do-files, rather than interactively, the total time saved is not great.

Having said that, there are times when the point-and-click interface can be useful. Most obviously, it can be quicker to use a dialog box to select a file to read in than to type out the full name of the file, particularly if it is nested in a large number of folders.

1.2. Getting Help

1.2.1. Help

Stata has a very comprehensive help system. There is online help for every stata command, which can be accessed by typing `help` *`cmdname`*. This will outline the syntax of the command, the meaning all of the available options, and often there will be a number of examples of its use. Unfortunately, this option is only useful if you know the exact name of the command you are interested in, but are unsure of the precise use or syntax.

1.2.2. Manuals

There is a very comprehensive set of manuals available online. Each entry in the manual begins with a section more-or-less identical to the help-file, but then goes into more detail. References and explanations of the mathematics are normally given, as well as more detailed worked examples. Clicking on the blue command name at the top of the help page will take you directly to the appropriate page in the PDF version of the manual.

1.2.3. Search

The main problem with the help system is that you need to know the exact name of a command in order to use it. Obviously, this is a drawback: how do you find out what command to use if you can't find out about it until you know its name ? The solution is to use `search` *`string`*, which looks through all of the helpfiles on the computer and materials available via stata's `net` command to find *string*, and produce a list of resources that mention it, along with a brief description of the resource.

There are a number of options to search to control where the search takes place. By default, it searches keyword databases on your computer and the internet: you can restrict the search to your local files only with the option `[`local]. For other possible options, see `help search`. Results are ordered according to likely relevance: first local material, then material from the stata web-site, then user-written material.

1.2.4. Website

There is a stata website at http://www.stata.com. There are literally hundreds of FAQ's online dealing with both common, simple problems and more complicated ways of using stata to perform analyses that it may appear at first sight that stata cannot perform.

1.2.5. Statalist

There is also a mailing list devoted to stata. Instruction for joining (& leaving) are given on the stata website. This is a fairly busy list (around 50 messages a day), consisting mainly of people saying "How do I do . . . " with stata", and getting replies that are usually very good. Stata support staff are also involved in the list and will jump in if they think that more detailed technical explanations are required. Old messages are archived, so you may find that searching the archives turns up the answer to your problem.

1.2.6. Stata Journal

The Stata Journal volumes 1-17 are available online at `"N:\Unit\The Stata Journal"`. Many of the articles are highly specialised, but there is a series entitled “Speaking Stata” which is aimed at beginners and gives a good introduction to how to use stata efficiently. These issues of the Journal are also available online, as are issues 18-20. Issues since 2021 are subscription only.

1.2.7. Me

If all else fails, try me. I’ll probably get the answer from one of the above sources (if I get it at all), but I will probably have a good idea of where to look, at least.

1.3. Basic Concepts

1.3.1. Main Stata Windows

On opening stata, you should see 4 windows: the command, results, variables and review windows. The results window has a white background, with black text[a]. The small window at the bottom is the command window, in which you enter commands. The review window contains a list of all of the commands that you have entered in the current stata session, and the variables window contains a list of all the variables in the current dataset.

Command Window

Command Syntax All stata commands follow a common syntax. The full syntax is explained in the User’s Guide. A very simplified version, sufficient to start with, is:

```
command [varlist] [,options]
```

What does that all mean ? Well, the square brackets `[]` surround optional parts, so that the only compulsory part is the actual command name. For example

```
describe
```

produces a list of the variables in the dataset, and

[a] by default: you can change these colours if you wish, but if you have found out how to do that, you certainly know what the results window is.

```
exit
```

ends the session.

However, most of the time, the command needs the name of one or more variables. For example,

```
summarize age
```

will produce the mean, standard deviation and some other statistics of the age variable, and

```
regress height age gender
```

will perform regression to predict height as a function of age and gender.

Most commands also have options, which always given at the end of the command, following a comma. For example,

```
summarize age, detail
```

provides a more detailed summary than

```
summarize age
```

If an option needs to be given a value, it is given in backets after the option name. For example, 95% confidence intervals are calculated by default, but for many commands this can be changed using the option `level`. If you want to get 99% confidence intervals in your regression output, you would use the command

```
regress varnames, level(99)
```

There are other clauses that can be slotted into this syntax, some of which we will see next week.

You do need to be careful with the syntax of a command, since simple errors like a misplaced space or a missing comma will give an error message. However, it does not take long to get used to it, and once you understand the logic of the syntax it is quite difficult to get it wrong.

You must remember that the language is case-sensitive: `height` and `HEIGHT` are two different variables. You could have both of these variables in the same dataset without confusing stata, but you will confuse yourself if you do. So I recommend that all your variable names contain only lower case letters.

Many commands and options have abbreviations, but I do not recommend using them until you are very familiar with the commands. It only takes a few seconds to type in the additional letters, and when you look at your work again in six months time, it will make it far easier to remember what you did. Also, you need to know the full name of a command in order to get help on that command. For this reason, I do not use any abbreviations in this document[b].

Repeating previous commands The command that has just been entered can be brought back to the command window using the `PageUp` key. Pressing this key repeatedly will bring back all of the previous commands. If you go too far back with the `PageUp` key, you can use the `PageDown` key to get more recent commands.

Variable name completion If you enter the beginning of a variable name and then press the `tab` key, Stata will complete the variable name as much as possible. If there are several possible completions, it will complete as far as possible and wait for further input. For example, if your data contains variables `height0` and `height1`, and you enter `hei` then press the `tab` key, the variable name will be completed to `height` and wait for you to type either 0 or 1. If you are used to filename completion in the bash shell, this works in the same way.

Variables Window

This contains a list of variables in the data set, together with their labels, although the labels may not be visible if the window is not sufficiently wide. Clicking on a variable name copies that name to the command window.

Review Window

This contains a list of commands that have been entered previously. Clicking the left mouse button on a command will copy it to the command window, where it can be edited and re-run if required. Double-clicking on a command in this window runs it directly. The entire list of commands in the review window can be saved as a do file by clicking on the upper left corner of the window: this gives a menu, one option being `"Save Review Contents"`.

Results Window

The results of any commands you type will appear in the results window. There is a scroll bar to return to previous output. However, the results window only retains a limited amount of your analysis, so you need to start a logfile (explained below) to ensure that you do not lose any output.

The results window is interactive, to the extent that any blue text represents a hyperlink: clicking on the blue text will generally open a viewer window with more information.

Normally, data is presented in the results window one screenful at a time. When paging through data in this way, pressing the `Return` advances by one line, the `Space` key advances by one page, and pressing `q` stops the output. The button containing a red circle with a white cross can also be used to stop the output.

[b] well, I try not to, but `gen` may slip instead of `generate` and `tab` instead of `tabulate`

It is possible to turn this paging off with the command

```
set more off
```

This is very useful in a do-file: it will then produce all of the output you need without you needing to press space all of the time. To turn paging back on again, type

```
set more on
```

1.3.2. Do-Files

A do-file is simply a list of stata commands. Giving the command

```
do do-file.do
```

causes stata to run all of the commands in the do-file. This is one of the greatest strengths of stata, the ability to perform exactly the same actions repeatedly, and it is vital to get to grips with this concept as soon as possible.

For example, if you wish to create a new variable, you should always use a do-file to do so, rather than just entering a command and saving the resulting dataset. This has two advantages:

1. You can see exactly how the variable was calculated by looking at the do file. If two people try to calculate the same variable and get different answers, it is possible to check exactly what each did and why they differ.
2. If you need to add subjects to the dataset later, or calculate the same variable in a different dataset, it can be done very simply and quickly.

It is also a good idea to keep a do-file containing your analyses. You can be certain that any analysis that you do will need to be performed many times, with very slight changes, before your write-up of the data-analysis is published. It is far easier to edit a do-file to make these minor changes than try to remember exactly how you performed an analysis originally and then do it again slightly differently.

You can also collect together a group of files needed for a particular analysis in a “Project”. These files can be do-files or datasets, and can make it easier to find all of the files that you need if they are spread across multiple directories. For example, most analyses involve some do-files used by everyone working on a given project, and some do-files specific each individual analysis, which would be stored in different places. Type “help Project Manager” for more information.

Profile.do

Every time Stata is run, it searches for a file called `profile.do` in certain places, including `C:\ado\personal\profile.do`, which is where I would recommending creating your own (for details, type `help profilew` into a Stata command window). This file contains commands which you wish to have run every time that you start Stata. Possible uses of `profile.do` is to define your own entries in the Stata menu system, or set up logging of commands as you enter them.

1.3.3. Log Files

Stata does not log your results by default. There is a “Stata Results” window which contains the results of each command, but this is of a limited size. It may not even contain the complete output of a single command if it is sufficiently complicated. To preserve your results for posterity, you need to open a log file. This can be done using

```
log using filename
```

All of your output will be stored in *filename* until you close the log with

```
log close
```

Stata can keep logs in two formats: SMCL or text. SMCL is a text markup language that is only understood by stata, so such logs must be viewed in stata. Text logs can be viewed in any text editor. The default format is SMCL: if you want the log to be in text format, the command to use is

```
log using filename, text
```

You should always open a log file as one of the first things that happens in a do file. The syntax to use is

```
capture log close
log using myfile.log, options
```

The reason for the "`log close`" command is that it is impossible to open a new log-file if one is already open. However, if a log file were not open, then the command "`log close`" would generate an error, and the do-file would halt. The command "`capture`" means “do not stop, even if there is an error”, so that the "`log using`" command will be run regardless.

1.3.4. Interaction with Operating System

Stata can only find files if either:

1. it is in the current working directory

2. you give the full path name to the file

Full names can be long and tedious to type in (not to mention error-prone), so it is useful to be able to change the current working directory. There are a number of commands to help with this.

pwd displays the name of the current working directory

cd "*dirname*" changes the current working directory to *dirname*. The inverted commas are optional, unless there is a space in *dirname*, but are a good habit to get into.

mkdir ”*dirname*" creates a new directory called *dirname*.

dir lists all of the files in the current working directory

shell *command_name* runs the command *command_name* in a command prompt window

Windows uses the symbol "\" to separate directories on a path, whilst Unix uses "/". Stata on windows can understand either, so either can be used in giving file and directory names. However, "\" will cause problems if it is followed by a macro (see below), and we now have a Linux workstation that requires Unix filenames, so it is a good idea to get into that habit.

1.3.5. Macros

A macro is a way of using a short name to represent a longer piece of text. For example, if you store your data in `N:\projects\A_Major_project\My_subproject\Data`, it can be a nuisance having to type that directory name in whenever you wish to read in a dataset. To make life easier, you can define a macro by typing

```
global mydir N:\projects\A_Major_project\My_subproject\Data
```

This creates a global macro called `mydir` containing the text `N:\projects\A_Major_project\My_subpro`
Then, whenever you type `$mydir` in stata, it will be replaced by the text `N:\projects\A_Major_project\My_subproject\Data`. So, if you type

```
use $mydir/mydata
```

stata will read in the dataset `N:\projects\A_Major_project\My_subproject\Data\mydata.dta`

A major advantage of macros is the ease with which do-files can be made portable. All directories that are used in a file should be referenced using macros. Then, when the do-file and data-files are moved to a different directory (as they will be when they are archived, if not before), it is only necessary to change the macro definitions and all of the references in the do-file will be changed.

Global vs Local Macros

The macros we have seen so far (beginning with “$”) are *global* macros, meaning that once they are set, they keep their value until stata finishes. There are also *local* macros, that only retain their value until the end of the do-file in which they are defined. Local macros are also used by the commands `foreach` and `forvalues`, which we will see next week. Local macros are defined using the command `local`, and used by putting them in single inverted commas: ‘ and ’ (the opening inverted comma is found at the top left of a standard UK keyboard, the closing one at the right of the middle row). So the “mydir” example above, rewritten to use local macros, would be

```
local mydir N:\projects\A_Major_project\My_subproject\Data
use "‘mydir’\mydata"
```

1.3.6. Variable lists

Often you need to present a list of variables to stata. Rather than listing the name of every variable individually, there are a number of shortcuts that can be used. For example, if you wish to summarize every variable which begins with the letters “age”, you would use the command

```
summarize age*
```

You can also give a list of variables which are consecutive in the dataset as `firstvar-lastvar`. For detailed information about specifying variable lists (or *varlists*), type

```
help varlist
```

1.3.7. Number Lists

There is also a shorthand for entering lists of numbers. For fuller details, type

```
help numlist
```

.

Symbol	Meaning	Example	Expansion
	list of numbers	1 2 3	1 2 3
x/y	whole numbers from x to y inclusive	1/5	1 2 3 4 5
x y to n	numbers from x to n, increasing by $y - x$	5 10 to 20	5 10 15 20
x y : n	same as x y to n	5 10:20	5 10 15 20
$x(y)n$	numbers from x to n, increasing by y	10(10)50	10 20 30 40 50
$x[y]n$	same as $x(y)n$	10[10]50	10 20 30 40 50

Table 1.1.: Number Lists

1.4. Manipulating Variables

1.4.1. Creating and modifying variables

Generate and Replace

The simplest command to create a new variable is `generate`. For example, if the date of birth is stored in the variable `date_of_birth` and the date the questionnaire was filled in is stored in `date_of_quest`, then the subject's age at the time the questionnaire was completed can be calculated as

```
generate age = (date_of_quest - date_of_birth) / 365.25
```

The above command would not have worked if a variable called `age` already existed in the dataset. In this case, it would have been necessary to either drop the variable `age` before generating the new variable:

```
drop age
generate age = (date_of_quest - date_of_birth) / 365.25
```

or to use the `replace` command:

```
replace age = (date_of_quest - date_of_birth) / 365.25
```

Either of the above will end up with exactly the same dataset.

By default, `generate` creates variables of type float (i.e. they contain numerical values, and can contain a decimal point). If you wish to generate a variable of a different type, you need to explicitly state that in your command. The available types are listed below:

type	size (bytes)	min	max	precision	missing values
byte	1	-127	126	whole numbers	.
int	2	-32,767	32,766	whole numbers	.
long	4	-2,147,483,647	2,147,483,646	whole numbers	.
float	4	-10^{36}	10^{36}	7 digits	.
double	8	-10^{308}	10^{308}	15 digits	.
str*n*	*n*				" "
strL	varies				" "

Table 1.2.: Available data types

For example, the command

```
gen str6 name = "MyName"
```

creates a variable called `name`, containing "MyName" in every observation. Variables of the type `strn` can contain up to *n* characters, where n has a maximum value of 2,045. Variables of type `strL` can contain up to 2,000,000,000 characters.

Missing Values

Missing values in string variables are always represented by `""`, but numerical variables can take a number of different missing values. The default missing value is simply a dot, but you can also use ".a" ".b" . . . ".z" if you wish to distinguish between different types of missing data: that subject was not asked the question, the question was not answered, the answer was illegible etc. However, all missing values are larger than the largest number that stata can represent with that data type, so you can always exclude *all* missing values with

```
if variable < .
```

Egen

A more powerful way of generating new variables is with `egen` (short for *Extended GENerate*. The syntax is very similar to that of `generate`:

```
egen newvar = fcn(varlist)
```

However, there are a large number of functions which can be used for *fcn(varlist)* which can perform calculations that `generate` cannot. For example, there are functions to calculate sums, means, medians, variances, z-scores, etc. Type `help egen` for a full list of available functions.

1.4.2. Labelling variables

Variable names in Stata are limited to 32 characters. This enables reasonably descriptive names to be used, but it may be that a fuller description of a variable is required. For example, it is often useful to know the exact wording used for a question. In this case, the command

```
label variable varname "label"
```

can be used. For example, if you wish to assign the label "How many days in the week do you drink alcohol ?" to the variable `alcohol`, the command used would be

```
label variable alcohol "How many days in the week do you drink alcohol
?"
```

Equally important is the ability to label the values of a variable. Usually, categorical variables are stored as numbers, since this requires less storage and is more efficient. However, it is important to know which values these numbers refer to. This can be done using value labels.

Assigning labels to values is done in two stages. First the labels are assigned to numbers. This is done with a `label define` command, such as:

```
label define yesno 1 "Yes" 0 "No"
```

which assigns the value “Yes” to the number 1 and the value “No” to the value 0.

The second step is to assign the label to a variable, with the `label values` command. This command takes the form

```
label values varname labelname
```

For example, to use the label `yesno` defined above with the variable `back_pain`, the command to use would be

```
label values back_pain yesno
```

Only one variable can be given in the `label values` command, yet there may be several variables which have the same label applied to them. We will see an efficient way of doing this next week.

1.4.3. Selecting variables

Often there are more variables in your dataset than you are interested in, and you may wish to use only a subset of the available variables. There are two commands to facilitate this: `drop` and `keep`. They work as you might expect: `drop` *`varlist`* removes all of the variables in *`varlist`* from the dataset, and `keep` *`varlist`* removes all of the variables that are not in *`varlist`* from the dataset.

1.4.4. Formatting variables

The command `format` can be used to change how stata presents data to you. It is most commonly used for dates. Dates are stored as the number of days since January 1, 1960. So January 2, 1960 would be stored as 1, and February 10, 2005 as 16477. This has the advantage that you can do arithmetic with dates, to calculate the time between two dates etc. However, it means that if you list a series of dates, you will see a list of numbers, and converting from the number to the date is not trivial.

But you can use the `format` command to do this for you. The command

```
format %dD/N/CY date
```

means that the variable `date` will be presented in the form Day/Month/Year, rather than as numbers. Alternatively,

```
format %dCY-N-D date
```

will present the dates as Year-Month-Day instead. For a full list of all of the characters that can be used in a date format, along with their meanings, type

```
help dfmt
```

The format command can also be used to determine the formatting of other numbers: how many decimal places to use etc. Full details are given by

```
help format
```

1.5. Manipulating Datasets

1.5.1. Filenames

Filenames that contain spaces can cause problems in stata, unless the filename is put in inverted commas. For this reason, I strongly suggest that whenever you use a filename, you put the name of the file in inverted commas, even if it does not contain a space: you may move it to a different directory later and wonder why your do-file no longer works. This is especially important when using a macro as part of the filename, since the macro may expand to something containing a space.

1.5.2. Reading and Saving data

The most important commands for manipulating datasets are `use`, which reads a stata dataset into stata, and `save`, which saves a stata dataset. These commands have a very simple syntax, although there are a few safeguards built in to them to stop you destroying your own data by mistake.

Use

The simplest way to read a file into stata is to issue the command

```
use "filename"
```

This will read the file *`filename`* into stata. However, if you have data already in stata, reading new data in will replace it and it may be lost forever. For this reason, stata will not read in the new dataset if there is data in memory that has been changed since it was read from the disk. If you want to keep the data you are currently working on, you need to enter

```
save "old_filename"
use "filename"
```

If you don't mind losing the existing data, you can type either

```
clear
use "filename"
```

or

```
use "filename", clear
```

either of which have the same effect.

Typing in the full name of a file can be tedious, especially when there are may levels of directories to go through. When working interactively, this is one of the commands which are much easier to use from the menus or button-bar. Once you have opened the dataset once, however, the command can be stored in a do-file for subsequent use. If you need to reopen the dataset (if you have made an error in manipulating the dataset, for example), the command can be rerun from the review window. Be warned, however, that using the menu always implies the option `clear`, so the safeguards built into `use` do not work.

Save

A stata dataset is saved using the command

```
save "filename"
```

However, this might not be what you really want: if a file of that name already exists, it will be replaced with the new file. If you want to have copies of both files available, you must use different names for them. If you want to replace the original file, you can with the command

```
save "filename", replace
```

Saveold

It may be that you need to save a dataset in an old format: you may be collaborating with someone who only has access to an old version of stata, or you may want to use StatTransfer, which does not recognise the last stata file format. This can be done using the command

```
saveold filename
```

You may specify `,replace` if you wish, but it would be a bad idea to replace a file in a more recent format with one in an older format: you may be losing information.

1.5.3. Combining Datasets

Sometimes you may want to combine data from 2 or more data sets. There are two possible situations:

1. The datasets contain the same information about two different groups of people
2. The datasets contain different information about the same group of people.

In the first situation (e.g. you have received datasets from two different centres, each recording the same information about different populations, and you want to combine the datasets), the command to use is `append`. In the second case (e.g. data from x-rays are stored in one file, DNA data in another, and you want to look at genetic risk factors for x-ray outcomes), the command is `merge`.

append

The command `append` adds new observations to the end of an existing dataset. It is essential that corresponding variables in the two datasets have exactly the same names, otherwise they will be treated as different variables. If variables have the same meaning, but different names, the command `rename` can be used to change the name in one of the files.

A simple example of appending data is shown below:

ID	common_1	common_2	file1_1	file1_2
1	a_1	b_1	c_1	d_1
2	a_2	b_2	c_2	d_2
3	a_3	b_3	c_3	d_3

Table 1.3.: Appending Data: File 1

ID	common_1	common_2	file2_1	file2_2
4	a_4	b_4	e_4	f_4
5	a_5	b_5	e_5	f_5
6	a_6	b_6	e_6	f_6

Table 1.4.: Appending Data: File 2

The variables `ID, common_1 & common_2` exist in both files, whilst `file1_1 & file1_2` exist only in file1 and `file2_1 & file2_2` exist only in file2. Thus there will be missing data for these variables in the combined file.

It will almost always be useful to create a variable which will tell you from which file a given subject was taken. A simple way of doing this is shown below:

ID	common_1	common_2	file1_1	file1_2	file2_1	file2_2
1	a_1	b_1	c_1	d_1	.	.
2	a_2	b_2	c_2	d_2	.	.
3	a_3	b_3	c_3	d_3	.	.
4	a_4	b_4	.	.	e_4	f_4
5	a_5	b_5	.	.	e_5	f_5
6	a_6	b_6	.	.	e_6	f_6

Table 1.5.: Appending Data: Combined Files

```
use "filename1"
gen fromfile = 1
append using "filename2"
replace fromfile = 2 if fromfile == .
```

Here, the file *filename1* is read into memory. Then a new variable, `fromfile` is created, which takes the value 1 for all observations in *filename1*. Next, addition observations are added to the dataset from *filename2*. Finally, `fromfile` is given the value 2 for any observations for which `fromfile` is currently missing. Since `fromfile` == 1 for all observations in *filename1*, this means that `fromfile` is set to 2 for all observations in *filename2*.

merge

Merging data is used more commonly than appending data. In order to merge two datasets, you need to specify which record(s) in one dataset correspond to which record(s) in the other. Therefore, before merging two files, you must ensure that there is a variable or group of variables that are identical in each dataset.

For example, suppose that you wish to merge the files `file1` and `file2`, and that the variable `idno` exists in both datasets and is a unique identifier for each subject. Then before merging, the datasets would look like this:

idno	var1	var2
1	a_1	b_1
2	a_2	b_2
3	a_3	b_3

Table 1.6.: Merging Data: File 1

In order to merge these two files, you need to make sure that they are both sorted by `idno`:

```
use file2
sort idno
save, replace
```

idno	var3	var4
1	c_1	d_1
3	c_3	d_3
4	c_4	d_4

Table 1.7.: Merging Data: File 2

```
use file1
sort idno
```

Then you can give the command to merge the files:

```
merge 1:1 idno using file2
```

Note the syntax of the command:

```
merge [1:1]|[1:m]|[m:1]|[1:m] variable_name using filename
```

After merging, the dataset would look like this:

idno	var1	var2	var3	var4	_merge
1	a_1	b_1	c_1	d_1	3
2	a_2	b_2	.	.	1
3	a_3	b_3	c_3	d_3	3
4	.	.	c_4	d_4	2

Table 1.8.: Merging Data: Combined Files

Things to note:

- All subjects who appear in either of the files will appear in the merged file.
- Subjects who do not appear in one of the files will have missing values for those variables.
- Stata has created a new variable called `_merge`, which contains the values

  1 if the value of idno exists in file1, but not in file2

  2 if the value of idno exists in file2, but not in file1

  3 if the value of idno exists in both files

The most common error in merging data is that one or other datasets is not sorted. In this case, you will get the error message

```
master data not sorted
```

or

```
using data not sorted
```

If it is the master data not sorted, you can simply enter the command

```
sort matching_variable
```

and try again. If it is the using data that is not sorted, save the current master file in its sorted state, then `use` and `sort` the second dataset.

The above example would be slightly more complex if there was no unique identifier for each subject. Imagine that there are a number of different centres participating in the study, and each numbered their subjects 1–n. You cannot merge be `idno`, since the same `idno` can exist in each centre, and correspond to different subjects. For this reason, you can use a number of variables to match on: in this case, the command to use is

```
merge 1:1 centre idno using file2
```

Of course, you would need to run the command

```
sort centre idno
```

on both files first.

Ensuring Uniqueness Very often, when merging, you will want to match a single record in one file to a single record in the other. If this is what you want, the command `merge 1:1` will verify that this is true, and produce and error if not. However, there are times that you want to match several observations in one file to a single observation in the other: maybe one file contains data on individuals, and the other contains data on the households to which they belong. Multiple household members would need to be matched to the same household. There are commands `merge 1:m` for the case where a single record from the data in memory should be merged to several records in the "using" file, and `merge m:1` if several records from the data in memory should be matched to a single record in the "using" file.

However, the `merge` command does not require this. If a subject appears twice in one of the files, there will be two records for that subject in the final dataset. This can be avoided by using the option `unique` to the `merge` command:

```
merge idno using file2, unique
```

This will produce an error message if either file contains more than one entry with the same `idno`.

1.6. Other Dataset Manipulation Commands

1.6.1. `browse` and `edit`

The command `browse` opens a data editor window, in which the dataset is presented in spreadsheet form. The data can be examined, but not changed, following a `browse` command. If you wish to examine only a subset of the data, you can list the variables you want to see after the `browse` command.

The command `edit` is similar to `browse` but does allow changes. **Do not use it**. Any changes you make to your data must be documented, so the best way to manage it is with a do-file.

1.6.2. `preserve` and `restore`

You may wish to change your data temporarily. For example, there is a command `collapse` which creates a new dataset, consisting of the means (or other statistics) of the variables in the current dataset, calculated for a number of subgroups. You may want to do some analysis of these means, then return to your original dataset. The command `preserve` saves a copy of your dataset to disk, so that you can get it back easily after analysing the `collapse`d dataset, and `restore` restores the previously saved dataset.

1.7. Practical on the Essentials of the Stata Language

1.7.1. Preliminaries

To start Stata, click on Start Button ⇒ All Programs ⇒ Stata 14 ⇒ StataIC 14 (64 bit)

This should work for most of you, but depending on your faculty and whether you are staff or a student, the exact route may vary. Let me know if you have any difficulty.

Solutions for all practicals can be found at `http://personalpages.manchester.ac.uk/staff/mark.lunt/stats_course.html`

Choose a directory to hold the work you are going to do on this course. I suggest that `P:` is the best place, maybe `P:\statacourse`. Set the global macro to the name of your chosen directory:

```
global mydir my chosen directory
```

(Remember text in italics is not typed in as it is, but replaced with the name of the directory you are using). This way, you can all chose different directories, but if I give my instructions in terms of the macro `$mydir`, they will work for all of you.

Of course, the directory you are going to use must exist, before you can save any files in it. You can use

```
mkdir "$mydir"
```

provided that the new directory is only one level below an already existing directory. To make sure that it exists, type

```
cd "$mydir"
```

to change to it, and make sure you don't get an error message in response.

It is a good idea to start a log now.

1.7.2. Reading and Saving Files

Type the command

```
sysuse auto
```

This will search for a dataset called `auto` that is installed with stata. You now want to save this dataset to your own directory. You can type

```
save "$mydir/auto"
```

to save this dataset in your own working directory, or if you have already changed to `$mydir`, you can just type

```
save auto
```

Make sure that you are in you own working directory (use `pwd` to find out, and `cd` to change if necessary). Now type

```
dir
```

to ensure that you really do have a copy of the dataset saved in this directory. If you have, type

```
clear
```

to remove the dataset from memory, then

```
use auto
```

to read the version in your own directory back in.

Now save the dataset using a different name, say `myauto`. This is because we are going to modify the dataset, and you should always make sure that your original data cannot be lost by mistake: save the original and work on a copy.

```
save myauto
```

1.7.3. Creating and Modifying Variables

Using `generate`

We are now going to create a new variable, `wtkg`, to contain the weight of each vehicle in kg. Since 1kg is approximately 2.2046lbs, the command to do this is

```
generate wtkg = weight/2.2046
```

It is good practice to label every variable as soon as you have created it, before you have time to forget what it is, so let's do that:

```
label variable wtkg "Weight (kg)"
```

Creating indicator variables

Very often, you want to generate a variable that can only take two values, representing "true" and "false". As an example, we will create a variable `short`, which takes the value 1 for all cars less than 190 in long and 0 for longer cars. The conceptually simplest way to do this is

```
generate short = 0
replace short = 1 if length < 190
```

However, there is a more efficient way: in stata, any logical expression (such as `(length < 190)`) has a value 0 if the expression is not true and 1 if the expression is true. So we could identify short cars with the single command

```
generate short2 = (length < 190)
```

If you type `tab short short2`, you should see that both commands have had exactly the same effect.

Using `egen`

Suppose that you wish to divide the vehicles into tertiles. There are a number of ways to do it: the simplest is to use `egen`. The function `cut` can be used to categorize a continuous variable, and you can either choose the thresholds yourself, or divide the observations into a given number of (more-or-less) equal sized groups. The code for doing this is

```
egen wtt = cut(weight), group(3)
```

As always, we now need to label this variable

```
label variable wtt "Tertiles of weight"
```

If you type `tab wtt`, you will see that `wtt` takes three values, 0, 1 or 2. The lowest tertile contains 24 cars, the others 25.

So that you don't get confused as to which tertile is which, it is best to label them. First, you need to define a label for them:

```
label def tertiles 0 "Lowest tertile" 1 "Middle Tertile" 2 "Highest Tertile"
```

Then you can assign that label to the variable

```
label values wtt tertiles
```

If you now repeat the command

```
tab wtt
```

you should see more meaningful labels for the categories of `wtt`.

Creating a string variable

If you type

```
tab make
```

you will see that of the 74 cars in the dataset, several are made by the same manufacturer: 3 Toyotas, 4 VW's etc. However, there is no variable to identify the manufacturer, so we had better create one. We will do this by taking the first word of the variable `make`, i.e. all of the characters up to the first space. This requires two functions: `strpos(`*`string1, string2`*`)`, which gives the position within *`string1`* at which *`string2`* first occurs, and `substr(`*`string, num1, num2`*`)`, which returns the substring of *`string`* that starts at position *`num1`* and ends at position *`num2`*. So, to extract the manufacturer of each vehicle, we use the command

```
gen str20 company = substr(make, 1, strpos(make, " "))
```

if you now type

```
tab company
```

you should see the following table

```
    company |      Freq.     Percent        Cum.
------------+-----------------------------------
        AMC |          3        4.11        4.11
       Audi |          2        2.74        6.85
        BMW |          1        1.37        8.22
      Buick |          7        9.59       17.81
       Cad. |          3        4.11       21.92
      Chev. |          6        8.22       30.14
     Datsun |          4        5.48       35.62
      Dodge |          4        5.48       41.10
       Fiat |          1        1.37       42.47
       Ford |          2        2.74       45.21
      Honda |          2        2.74       47.95
      Linc. |          3        4.11       52.05
      Mazda |          1        1.37       53.42
      Merc. |          6        8.22       61.64
       Olds |          7        9.59       71.23
    Peugeot |          1        1.37       72.60
      Plym. |          5        6.85       79.45
      Pont. |          6        8.22       87.67
    Renault |          1        1.37       89.04
     Toyota |          3        4.11       93.15
         VW |          4        5.48       98.63
      Volvo |          1        1.37      100.00
------------+-----------------------------------
      Total |         73      100.00
```

Notice that we have lost a car: there are only 73 values of `company`. This is because the the Subaru has no model name, and hence there is no space in `make` for this car. There are a number of solutions for this: the simplest is

```
replace company = make if company == ""
```

You now need to save this dataset, as we may use it again later:

```
save, replace
```

1.7.4. Manipulating Datasets

In order practice combining datasets, we need to create two datasets to combine. We will use a simulated dataset of blood pressure measurements called `bplong`. This file contains two records of blood pressure for each subject: one made before some intervention, the second taken after the intervention. The variable `when` takes the value 1 for the “before” measurement and 2 for the “after” measurement. This dataset will be split into separate “before” and “after” datasets.

This is done with the following commands:

```
sysuse bplong
save "$mydir/bplong"
preserve
keep if when == 1
save "$mydir/bpbefore"
restore
keep if when == 2
save "$mydir/bpafter"
```

Now, all of the records for the first visit are in `bpbefore` and all those for the second visit in `bpafter`.

Append

First of all, we will see how to append the two datasets, to reproduce the structure of `bplong`. The code to do this is:

```
use bpbefore, clear
gen fromfile = 1
append using bpafter
replace fromfile = 2 if fromfile == .
```

Now you can check that `fromfile` agrees with `when` by typing

```
tab fromfile when
```

If you wish to save this file, make sure that you give a suitable label to `fromfile` first. You may also wish to label the values that `fromfile` takes.

Merge

Now we are going to merge the files, so that we end up with a single record for each subject, and two separate variables for the "before" and "after" measurements. Since we want two variables in the merged dataset, we will need to change the name of the variable `bp` in at least one of the files. In fact, it is easier to change the variable name in *both* files.

```
use bpbefore, clear
rename bp bp_before
save, replace
use bpafter
rename bp bp_after
save, replace
```

Note that you need the replace option to save, since the file already exists, but we want to change it. Now, you need to ensure that both files are sorted by patient, so that they can be merged:

```
use bpbefore
sort patient
save, replace
use bpafter
sort patient
save, replace
```

Yes, it would have been more efficient to do the sorting and renaming at the same time, rather than have to read the files in twice. You can do that in future. Of course, if you remember to use PageUp and PageDown to get older commands, or took your commands from the review window, there was very little typing to do anyway.

Now, we can do the actual merge:

```
merge 1:1 patient using bpbefore
```

If you have followed the previous instructions, bpafter is still loaded in stata, so you just need to merge in bpbefore. You can make sure that the merge was successful by typing in tab _merge: you should see that _merge takes the value 3 for all 120 patients, showing that they had data in both files.

1.7.5. Further Exercises

1. Calculate the lengths of each car in metres, using the fact that 1 inch is 0.0254 metres.
2. Create a variable heavy, which takes the value 0 for cars weighing less than 3000 lbs and 1 for cars weighing more than 3000 lbs.
3. Create tertiles of wtkg from the auto dataset, and check that it produces exactly the same results as creating tertiles of weight.
4. A simpler way of creating the company variable would be to use egen with the ends function. Use help egen to find out how this function works, then create a new variable called comp2, containing the first word of make.
5. Using the dataset that you created in Section 1.7.4, calculate the change in blood pressure between the “before” and “after” visit.
6. Using the same dataset, create a variable that takes 6 different values, depending on which age group and sex the patient belongs to. (*Hint: the group option to egen will help: check it out*) Label the values so that you can tell which group is which.

2. *Summarizing Data*

Having collected some data, you need to be able to describe it. Simply presenting all of the data to others is impractical, so there needs to be some way of summarising the data. This may be done graphically or numerically: both approaches have advantages and disadvantages. However, different types of data require different methods, so we must first consider the types of data we may be dealing with.

2.1. Types of Data

2.1.1. Quantitative data

Quantitative data is any kind of data that is meaured on an interval scale. It makes sense to do mathematics with the values that the data takes. For example, height and weight are both quantitative variables. In particular, it makes sense to use subtraction on these measures: for example if you weigh 75kg today and last week you weighed 76kg, you could say you have lost 1kg.

We further subdivide quantitative data into two types: discrete and continuous. Discrete data can only take a fixed number of distinct values, whereas continuous data can take any value within a given range. So height and weight are continuous, whereas number of children or number of visits to the G.P. this year are discrete.

2.1.2. Qualitative data

Qualitative, or categorical, variables are used when it is necessary to classify each observation into one of a number of groups. For example sex, ethnicity and marital status are all qualitative variables.

It may, in some cases, be possible to impose an ordering on a categorical variable. For example, a patient may be asked to describe their pain as none, mild, moderate or severe. Clearly, in this case, severe pain is always worse than any of the other categories. Such data is referred to as *ordinal*

2.1.3. Caveats

It is not always straightforward to decide what type of data you are dealing with. Indeed, it is often the case that the same variable may be appropriatedly treated as a different type in different circumstances.

Nominal vs ordinal

Whilst it may be possible to impose an ordering on a categorical variable, and thus treat it as ordinal, it may not necessarily be helpful. For example, hair colour is a nominal variable, but it could be ordered according to how dark it is, and this may be of use if the question being investigated relates to melanin.

Ordinal vs Discrete

If a person is asked to rate their pain on a scale of 1 to 10, that looks very like a discrete (or possibly even continuous, if people give fractional scores). However, treating it as discrete makes an assumption that a change in pain from 1 to 2 is somehow the same as a change in pain from 9 to 10. It is unlikely that anyone is capable of being that precise and rational about their own pain, so it would probably make more sense to treat it as ordinal.

Sometimes quantitative data may be grouped before you receive it, and thus become ordinal. For example, the number of years of education a person has received is a discrete variable. However, if education is simply classified as primary, secondary or university, so much information about the actual number of years has been lost that it would make more sense to treat that variable as ordinal.

Continuous vs Discrete

Technically, there is no such thing as a continuous variable in a computer analysis. There will always be a smallest possible increment in the variable that can be recorded by the computer, which makes the variable discrete. However, it makes sense to treat the variable as continuous since that smallest possible increment may be so small as to be negligible.

However, it may be that the smallest measurable increment is substantial. For example, I once worked with a study that involved measuring the diameter of the brachial artery on a computer screen. The resolution was not great, and the artery would be either 2, 3 or 4 pixels wide. Although the true diameter would be a continuous variable, the data we had to work with was discrete.

2.2. Summarising Qualitative Data

Qualitative data is easier to summarise simply because we cannot do maths with it. All we can do is say how many observations belong to each category. Those numbers may then be presented either numerically or graphically.

2.2.1. Numerical Summaries

When describing qualitative data numerically, it is best to give both the number and the proportion (or percentage) in each category. Proportions are useful since they are not affected by sample size: the proportion in each category will be roughly the same for a study of 1,000 people and a study of 100,000 people, but the numbers in each category will be completely different.

The stata command you would use to produce a numerical summary of a categorical variable is `tabulate`. The example below shows the results of typing `tabulate region` on a dataset taken from the

```
region |      Freq.     Percent        Cum.
------------+-----------------------------------
     Canada |        422       22.84       22.84
        USA |        541       29.27       52.11
     Mexico |        223       12.07       64.18
     Europe |        493       26.68       90.85
       Asia |        169        9.15      100.00
------------+-----------------------------------
      Total |      1,848      100.00
```

This shows the number of subjects recruited in each of 5 regions, in the column labelled "Freq.". The percentage from each region is in the column labelled "Percent". The column labelled "Cum." for "cumulative percentage" is completely meaningless for a nominal variable like region, but stata always produces it anyway. It only has any meaning for an ordinal variable: the categories of a nominal variable can be presented in any order, and changing the order would change the cumulative percentage columm.

2.2.2. Graphical Summaries

In general, a graphical summary of a qualitative variable will take up more space than a numeric summary, but for many people it will be easier to absorb the information in this form. When writing a paper, there would normally not be space to produce graphical summaries of every qualitative variable in the dataset (although they could always be included as supplementary material if necessary), but it may be worth including one or two for the most important variables, or if they clearly show an important feature of the data.

The best graphical summary of a qualitative variable is a bar chart. In this representation, there is a bar for each category, and the length of the bar is proportional to the number of observations in that category. The lengths of the bars may be labelled with either the number or the proportion in each category: this has no effect on the shape of the chart at all.

The vertical axis may be labelled with the number of observations that belong to that group, or the proportion. This does not change the shape of the graph in any way. Figure 2.1 is an example of a bar chart: it shows the number of patients recruited by region in the SLICC study.

2.3. Summarising Quantitative Data

One simple approach to summarising numerical data is to divide the data into groups and treat the groups as a categorical variable. It may not even be necessary to create your own groups if the numerical variable is discrete with a reasonable small number of possible values. This approach is commonly used for age: the sample is divided into 5-year or 10-year age-bands, depending on the range of ages in the sample, the number and proportion of people in each age band is given.

However, there a also a number of other ways to summarise numerical data. These summaries use the fact that it is possible to do mathemetics with the values of the variable, for example to calculate a mean.

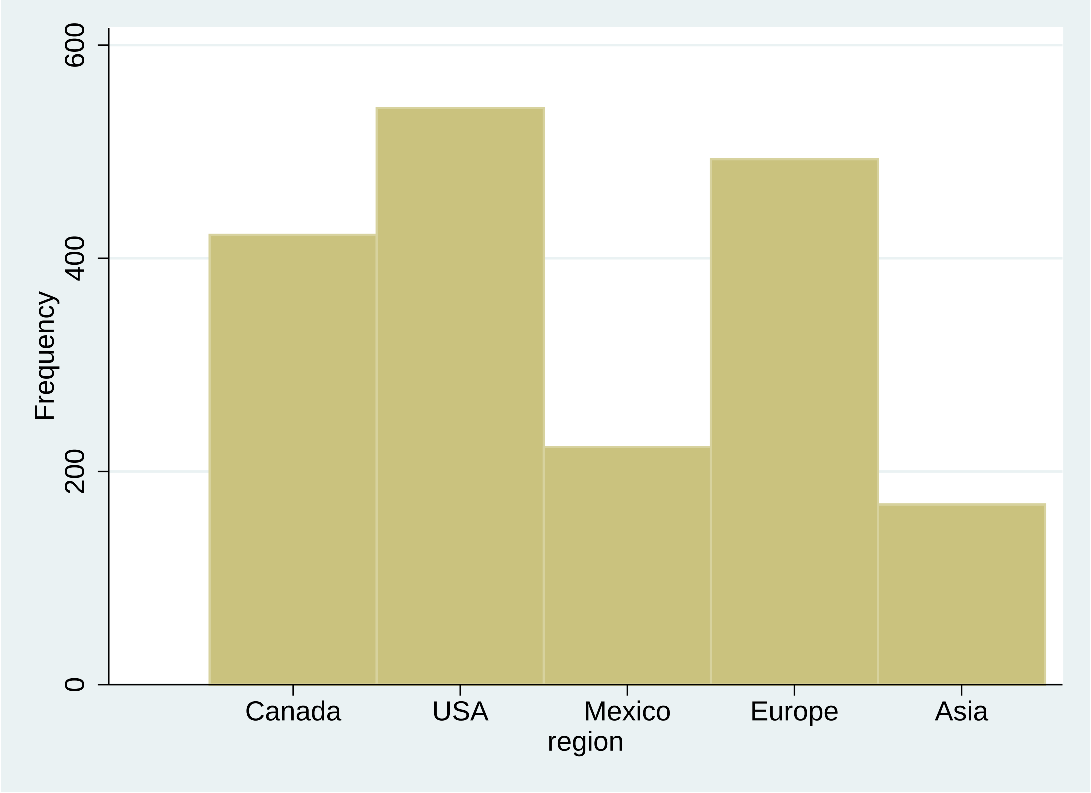


Figure 2.1.: Bar Chart for Recruitment by Region

2.3.1. Numerical Summaries

With numerical or quantiative data, there are a number of characteristics of the data that can be described numerically. The most important of these answering the questions “What is a typical value ?” (measure of location)and “How much do the values vary ?” (measure of scale). Some other descriptive statistics are discussed in section 2.3.1, but they are not widely used. This is partly because they are unnecessary for normally distributed data: given measure of location and a measure of scale, a normal distribution is completely determined.

Measures of Location

The most commonly used measures of location are the mean and the median. The mean, often written with a bar over the variable name ($\bar{x}$, pronounced “x bar”), is calculated by adding all of the values, then dividing by the number of values. In other words

$$\begin{aligned} \bar{x} &= \frac{x_1 + x_2 + \ldots + x_n}{n} \\ &= (\Sigma_{i=1}^{n} x_i)/n \end{aligned}$$

The median does not have a special notation, and is calculated by arranging all of the values in order, and picking out the middle one. If there are an even number of values, take the mean of the middle two.

Changing a single value will always change the mean (albeit very slightly in a large sample), but it is far less likely to change the median. The median can only change if the value that is changed goes from below the median to above it (or vice versa), or it is the middle observation (or one of the two middle observations if there is an even number. This makes the median far less sensitive to outliers, and it can be a better measure of location if the data is highly skewed.

For example, the list of numbers below are the number of days of absence from work due to sickness:

1,1,2,2,3,3,3,4,4,4,5,6,6,6,7,7,8,10,10,38,80

The mean value is 10, despite the fact that almost all of the observations are smaller than 10. This is not a good answer to the question "What is a typical value ?". The reason for this is the fact that there are two very large observations (38 and 80) which increase the mean considerably. The median for this data is 5, which gives a much better idea of a typical value: 9 of the 21 observations are within 1 day of this value, and 12 (more than half) are within 2 days. This is why the median should always be preferred to the mean when giving a measure of location for skewed data.

Measures of Scale

There are a number of statistics that can be used as a measure of how much a numerical variable varies.

Range You will occasionaly see the range of the data presented, that is the smallest value and the largest value. This is not a good idea, since it is determined completely by the two most extreme measurements. This means that as the sample size increases, the range can never decrease. If the new observation lies within the range of existing observations, the range is unchanged. If it lies outside this range, the range increases.

Inter-Quartile Range A better, yet still simple, measure of scale is the inter-quartile range. This is calculated by finding the quartiles of the data, those values that lie one quarter of the way through the list of observations in order and three quarters of the way through. For the sickness days data shown above, there are 21 observations, so the lower quartile would be at observation $\frac{21}{4} = 5.25$, i.e. between the 5th and 6th observation. Since both of these observations take the value 3, the lower quartile is 3. The upper quartile would be at observation $\frac{3\times21}{4} = 15.75$, i.e. between the 15th and the 16th observation. Since both of these observations are 7, the upper quartile is 7 and and the inter-quartile range is (3, 7). This means that the central half of the data lies between 3 and 7.

Standard Deviation A third possible measure of scale is the standard deviation. This can be thought of as the average distance of an observation from the mean. This is not quite true: thanks to the definition of the mean, the mean distance the observations in a sample from the sample mean is 0.

Since the average distance from the mean is not helpful, we sometimes use the average of the *squared* distance from the mean. This quantity, called the variance, can be calculated as

$$\text{Variance} \quad = \quad \Sigma(x_i - \bar{x})^2/n$$

The disadvantage of this statistic is that it is not in the same units as x, but in the square of that unit. For example, for the sickness days data, the variance is 316.2 days^2.

Variance is a particularly unintuitive measure of variability, as it is difficult to conceptualise what a squared day is. If we take the square root of the the variance, we get the standard deviation:

$$\text{Standard Deviation} \quad = \quad \sqrt{\Sigma(x_i - \bar{x})^2/n}$$

This statistic is in the same units as the original variable, which makes it easier to interpret. For example, the standard deviation of the sickness days data is 17.8 days. This seems quite high for a measure of how far the observations are from a typical value on average, but his is because it is strongly influenced by the outlying values of 38 and 80.

Other Descriptive Statistics

Quantiles and Percentiles We have seen that the median is a value such that half of the data is less than or equal to it, and half of the data is greater than or equal to it. However, there is nothing magical about half of the data: we could have a value such that one third of the data is less than or equal to it, and two thirds of the data is greater than or equal to it: this is the lower tertile.

Tertiles and quintiles are commonly used in data analysis: they can be used to divide your data up into a small number of equally sized groups. However, I would not recommend this common approach, since another study would have different tertiles and quintiles and generated different boundaries for their groups, thus making it harder to compare studies. Far better to use meaningful values for the variable in question: for example, if you wish to categorise BMI, use $<$ 18; 18 – 25; 25 – 30; and $>$ 30, since these are widely accepted thresholds, even if you do end up with groups of differing sizes.

It is possible to define any quantile (dividing the data into fractions, such as thirds or quarters) or percentile (selecting the bottom $x\%$ of the data, provided that there are enough observations).

Higher Moments If a variable follows a normal a normal distribution, the mean and the standard deviation are sufficient to describe it exactly. All quantiles and percentiles can be calculated if these two numbers are known. However, there are an infinite number of differently shaped distributions that have the same mean and standard deviation. For example, it may have some extremely high values (positively skewed, see Figure 2.2) or some extremely low values (negatively skewed). It may have more than one peak (bimodality, see Figure 2.3), usually due to two (or more) distinct populations being mixed and treated as one.

There are additional statistics that can be presented to describe the shapes of these distributions. For example, *skewness* is a measure of how lop-sided the distribution is, and *kurtosis* is a measure of how much of the weight of the distribution is the tails. Just as the variance is based on the squared differences between the observations and the means, skewness is based on cubed differences and kurtosis

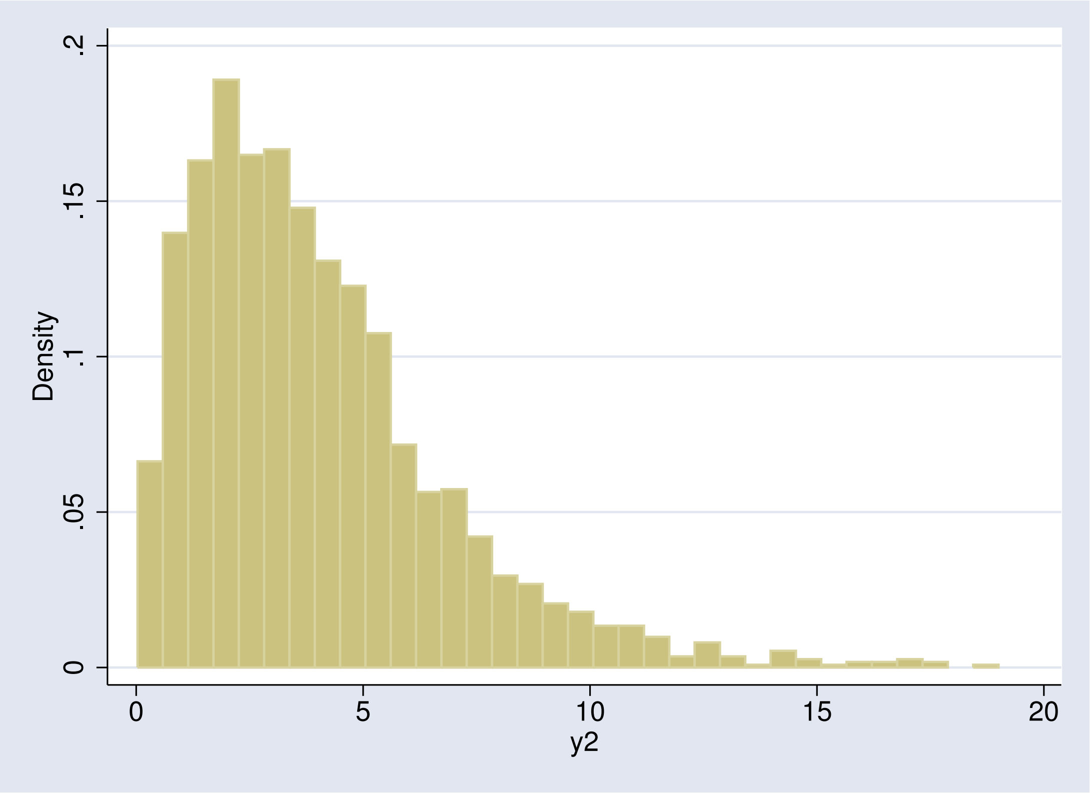


Figure 2.2.: Positively Skewed Distributionn

on differences raised to the power 4. The exact formulae are a bit more complex than the variance formula, and not of any interest, so look them up on wikipedia if you really want to see them. Skewness and kurtosis are rarely presented when giving a numerical summary of a quantitative variable, since they are not well understood. If you are presenting a distribution in which the non-normality matters, it is better and clearer to give a graphical summary.

2.3.2. Graphical Summaries

Histogram

A histogram is similar to a bar chart, but is used for quantitative, rather then qualitative variables. Rather than having pre-determined groups on the x-axis, the width of each bar can be determined individually. You can, in theory, have bars of different widths, although there is no way to achieve that in Stata. Which is a shame, since it could be useful: if you are recruiting subjects aged 18–50, it would make sense to have a group for those less than 20, then divied into 5 or 10 year age-bands. You would need the first bar to have a width of 2, the rest to have widths of 5 or 10.

The big difference between a bar chart and a histogram is that with a bar chart, the *height* of the bar is proportional to the size of te group, whereas with a histogram, it is the *area* of the bar. However, if the bars are all of equal width, those two statements are equivalent.

Setting the bar width correctly can be essential to produce a meaningful graph, and there is no single

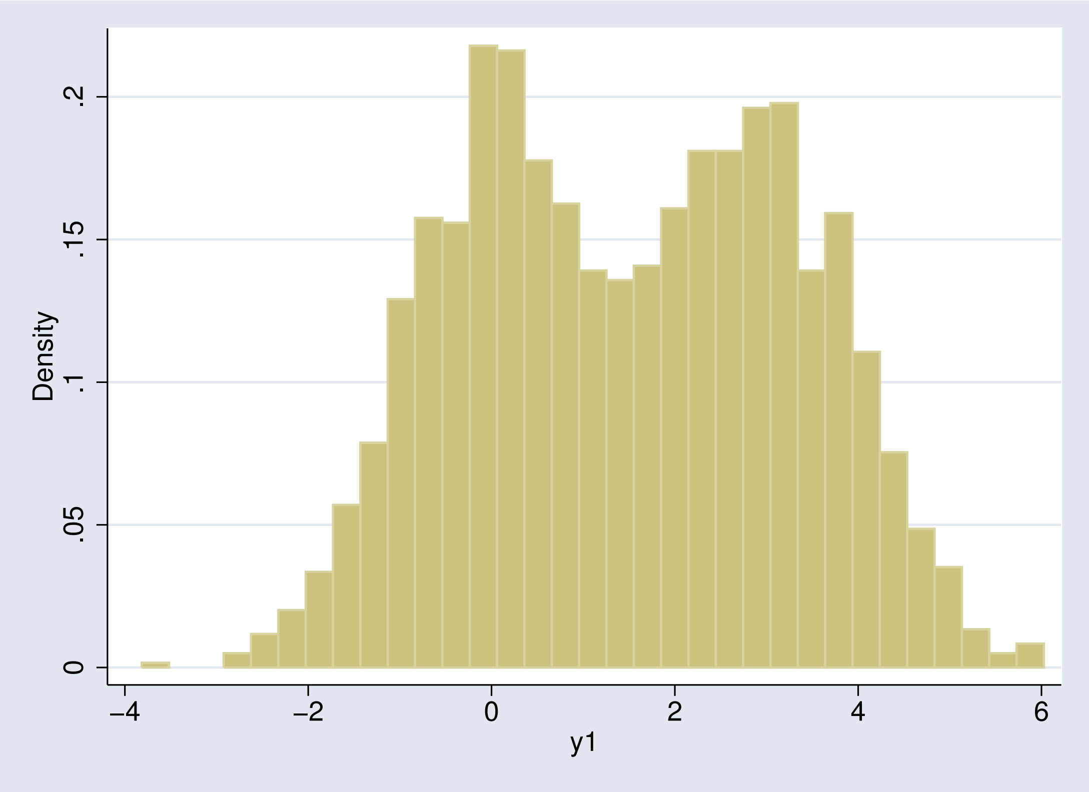
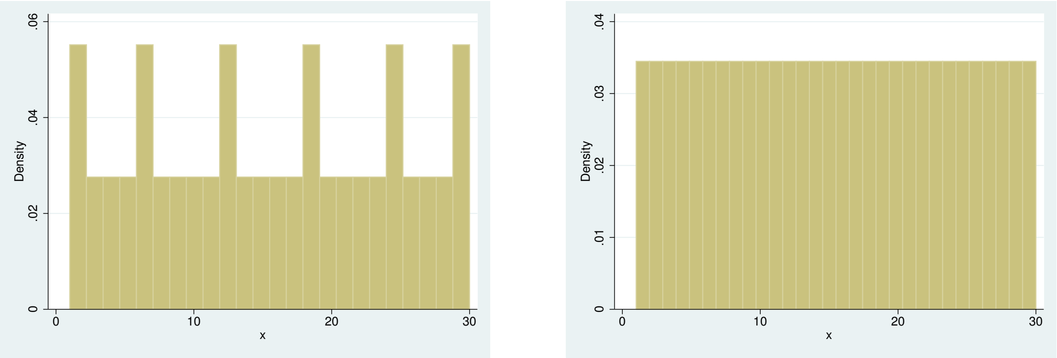


Figure 2.3.: Bimodal Distribution

formula which tells you what it should be, although there are a number of recommendations based on the sample size and the range of the data. Particularly with discrete data, it can be very important to set the bin width to an integer value. To see why, consider Figure 2.4. By default, Stata used 24 bins for 30 discrete values, meaning that some bins contained 2 values whilst most contained only 1. This gave the characteristic "comb" effect seen in Figure 2.4(a). Ensuring that each value had its own bin gave Figure 2.4(b), revealing that each value was equally likely to occur.

Bar charts and Histograms in Stata Both bar charts and histograms are produced with the command `histogram`. You can set the number of bars in the chart with the option `bin()`, or the width of each bar with the option `width()`. Alternatively, the option `discrete` tells stata to produce a bar for each value in the dataset. Stata also has the commands `graph bar` and `graph hbar` which produce graphs that look like bar charts, but they are really intended to show the association between a continuous variable and a categorical variable, rather than showing the distribution of a single continuous variable.

If you are concerned with the normality of a distribution, the option `normal` can be useful. This will overlay the histogram with a normal distribution having the same mean and standard deviation as the observed data. Any deviation from normality with then become clearer. An example is shown in Figure 2.5



(a) 24 bins (default

(b) 30 bins (correct

Figure 2.4.: Effet of Changing Number of Bins with a Discrete Variable

Box and Whisker Plot

An alternative graphical summary of a quantitative variable is a box and whisker plot. The central box shows the median and the upper and lower quartiles, and the "whiskers" show the range of "normal" values, as well as any individual "outlying" values. The definitions of "normal" and "outlying" can vary, but Stata treats any observations more than 1.5 interquartile ranges away from the nearest quartile as outlying. See `help graph box` for details. A box and whisker plot does not show the "shape" of the distribution as well as a histogram would, but it can be very useful for comparing distributions in different subgroups.

For example, consider the two box plots in Figure 2.6. The left panel shows an normal distribution, the symmetry of this distribution is made clear: the median line is in the centre of the box, and the whiskers are of very similar lenghths. However, the right panel, of a skewed distribution, is markedly asymmetrical: the median line is is the lower part of the box, the upper whisker is much longer than the lower whisker, and the outlying values are all large values, there are no small outliers.

The Stata command for producing a box and whisker plot is `graph box`.

2.4. Table 1

It is extremely common for the first table in a clinical or epidemiological research paper ("Table 1") to provide descriptive statistics for the main variables of interest in the study. When numerically summarising a quantitative variable, if the variable is normally distributed you can present the mean and standard deviation, otherwise present the median and the quartiles. It may be simpler to present the median and quartiles for all quantitative variables if some are normally distributed and others are not, but views differ on that. For qualitative variables, present the number and percentage in each category.

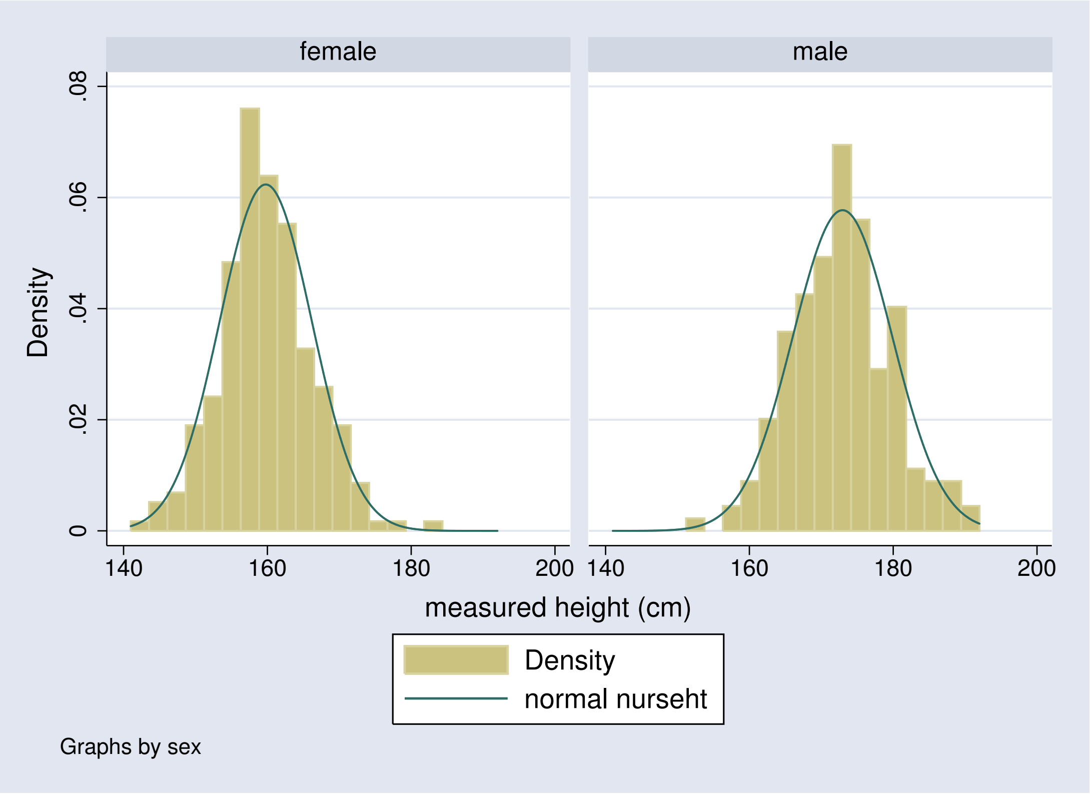
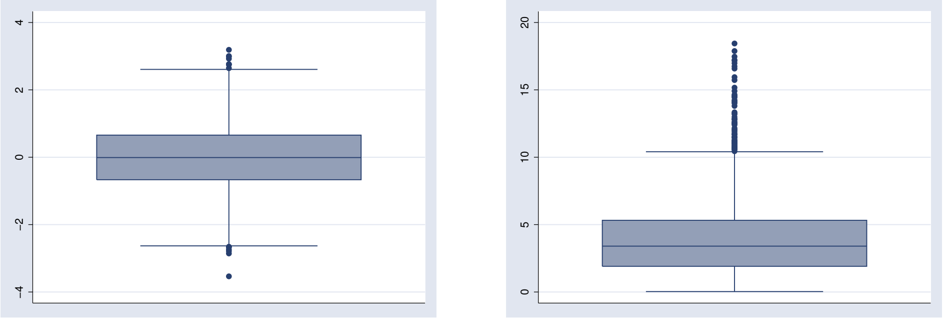


Figure 2.5.: Histogram with superimposed normal distribution



(a) Normal Distribution

(b) Skewed Distribution

Figure 2.6.: Boxplots of a normal and a skewed distribution

2.5. Further Reading

Further Reading

[1] E. Hayes-Larson, K. L. Kezios, S. J. Mooney, and G. Lovasi. Who is in this study, anyway? Guidelines for a useful Table 1. *Journal of Clinical Epidemiology*, 114, 2019.

Further Reading

2.6. Practical on Summarising Data

2.6.1. Hand Calculations

This section gives you the chance to do some calculations for yourself and see how the concepts we saw in the lecture work in real life. Once upon a time, these calculations would have been done by hand: I'm sure you could do them in your head, but getting stata to do them for you will be quicker. However, we are going to go through the steps that you would have to perform if you were calculating them by hand, so that you can see how it works. In practice, you would simply ask stata to churn the results out rather than calculating them this way.

To start Stata, click on Start Button ⇒ All Programs ⇒ Stata 14 ⇒ StataIC 14 (64 bit)

This should work for most of you, but depending on your faculty and whether you are staff or a student, the exact route may vary. Let me know if you have any difficulty.

Solutions for all practicals can be found at
`http://personalpages.manchester.ac.uk/staff/mark.lunt/stats_course.html`

Load in the PImax dataset with the commands

```
global datadir http://personalpages.manchester.ac.uk/staff/mark.lunt/stats
use $datadir/2_summarizing_data/data/pimax.dta
```

(Note that there are two separate commands above, and they need to be entered separately, each on its own line.)

This dataset contains a single variable, pimax, which is the maximal static inspiratory pressure, measured in cm H_2O, of 25 cystic fibrosis patients.

Sort the data into ascending order with the command

```
sort pimax
```

6.1 Now look at the data in the spreadsheet view using the command `browse` or the browse button, and look up the median value ......

6.2 From the same browse view, what are the 25th and 75th centiles (the values of the 7th and 18th observation respectively).

Lower quartile ......

Upper quartile ......

6.3 Now we will calculate the mean. First put the data back in order with the command

```
sort id
```

Then we can generate a running sum with the command

```
gen sum = sum(pimax)
```

If you look at the data with `browse`, you will see the variable `sum` contains a running sum, i.e. the sum of all observations from the top of the dataset to that observation inclusive. Hence, `sum` in the last observation contains the sum of `pimax` in all of the observations.
Finally, we have to divide by the number of observations. We can do this with two commands: first

```
gen n = _n
```

will put the number of the observation into a variable called `n`, then

```
gen mean = sum/n
```

will generate a running mean: the mean of all observations from the top of the dataset to the current observation. Again, you can browse the data.
What is the mean PImax in this sample ? ......

6.4 Finally, we are going to calculate the standard deviation. First, we need to create a variable containing the mean value for each observation:

```
drop mean
egen mean = mean(pimax)
```

If you `browse` the data, you will see that the variable `mean` now contains the same value for each observation, the overall mean. This is an important difference between `gen` and `egen`. (From now on, I will assume that you are `browse`ing the data each time you change it and will not explicitly mention it).
Next we calculate the difference between each observation and this mean (i.e. $x_i - \bar{x}$):

```
gen diff = pimax - mean
```

Square the difference to give $(x_i - \bar{x})^2$:

```
gen diff2 = diff * diff
```

and add them all up ($\Sigma\,(x_i - \bar{x})^2$):

```
gen diff2_sum = sum(diff2)
```

Dividing by the observation number will give a "running variance", with the value of the variance in for the entire sample in the last observation ($\frac{\Sigma(x_i-\bar{x})^2}{n}$)

```
gen variance = diff2_sum / n
```

Finally, take the square root of the variance to get the standard deviation ($\sqrt{\frac{\Sigma(x_i-\bar{x})^2}{n}}$):

```
generate sd = sqrt(variance)
```

Again, the value for the entire sample will be in the last observation.
What is the standard deviation of PImax in this sample ?[a] ......

2.6.2. Summarising Data in Stata

Read the stata file `htwt.dta` into stata by typing

```
global datadir http://personalpages.manchester.ac.uk/staff/mark.lunt/stats
use $datadir/2_summarizing_data/data/htwt.dta, clear
```

This file includes two BMI values: `bmi` which was based on measured data and `bmirep` which was based on reported data.

6.5 Examine the distribution of measured BMI scores by displaying a histogram of the data, using the command

```
histogram bmi
```

Is the data normally distributed, or does it show some skewness ?

....

2.6.3. Summarise

6.6 Calculate summary measures of the measured BMI using the command

```
summarize bmi, det
```

6.7 Write down the mean BMI. ......

6.8 How does the mean compare to the median ?

....

....

....

6.9 What are the lower and upper quartiles of the data ?

....

....

....

2.6.4. Summarise by group.

6.10 Display the data separately for each sex, using the commands

```
sort sex
by sex:  summ bmi, det
```

6.11 The distributions for the two sexes can be neatly compared graphically using boxplots. The command to do this is `graph box bmi, by(sex)`. Write down a short description of what you see.

. . . .

. . . .

. . . .

. . . .

2.6.5. ***`tabstat`***

This command can be used to produce tables of summary statistics: it is similar to `summarize`, but the output it produces is far more controllable. The basic syntax is `tabstat` *`varlist`*`, statistics(`*`statname [statname ...]`* The option `statistics` can be given one or more of the options in table 2.1.

For example,

```
tabstat bmi, statistics(mean sd)
```

would give the mean and SD of BMI. There is also a `by()` option, which enables you to obtain the statistics for different subgroups:

```
tabstat bmi, statistics(mean sd) by(sex)
```

will give the mean and SD of BMI for men and women separately.

6.12 Use `tabstat` to produce the mean and SD of height and weight, as measured by the nurse, for men and women separately (If you have forgotten the names of the variables to use for this, try typing `describe`.

. . . .

. . . .

statname	definition
mean	mean
count	count of nonmissing observations
n	same as count
sum	sum
max	maximum
min	minimum
range	range = max - min
sd	standard deviation
var	variance
cv	coefficient of variation (sd/mean)
semean	standard error of mean = sd/sqrt(n)
skewness	skewness
kurtosis	kurtosis
median	median (same as p50)
p1	1st percentile
p5	5th percentile
p10	10th percentile
p25	25th percentile
p50	50th percentile (same as median)
p75	75th percentile
p90	90th percentile
p95	95th percentile
p99	99th percentile
iqr	interquartile range = p75 - p25
q	equivalent to specifying “p25 p50 p75”

Table 2.1.: Statistics available in `tabstat`

2.6.6. `table`

An alternative to `tabstat` is `table`. This is more flexible in some ways and less flexible in others: in particular, it can only produce a maximum of 5 statistics per group. The syntax is

```
table groupvars, contents(contents_list)
```

where *contents_list* consists of pairs of statistic names and variable names. So, to get the mean and SD of BMI for men and women using `table`, you would type

```
table sex, contents(mean bmi sd bmi)
```

6.13 Use `table` to produce the mean and SD of height and weight, as measured by the nurse, for men and women separately

. . . .

. . . .

2.6.7. Further exercises

6.14 What is the average age of the subjects ? . . . . . .

6.15 Draw a histogram of the ages, using the command `histogram age`. Do the ages follow a normal distribution ?

. . . .

. . . .

6.16 How old are the youngest and oldest males and females in the study ?

	Youngest	Oldest
Males	. . . . . .	. . . . . .
Females	. . . . . .	. . . . . .

6.17 What is the mean of the self-reported BMI. Is this greater or less than the mean of the BMI as measured by the nurse ?

. . . .

. . . .

6.18 Create a variable for the difference between measured BMI and self-reported BMI:

```
gen bmidiff = bmi - bmirep
```

Write down its mean value, standard deviation and the number of subjects for whom both BMI measures are available.

Mean ......

Standard Deviation ......

Both available on .... subjects

6.19 Produce histograms of height in men and women, using the commands

```
histogram nurseht, by(sex)
```

and

```
histogram nursewt, by(sex)
```

Add a superimposed normal distribution to the graphs with the `normal` option to the `histogram` command.

6.20 Start a Word document. Select the Graph window in stata and use Ctrl+C to copy the graph. Switch to the Word document and use Ctrl+V to paste the graph.

3. Sampling and Confidence Intervals

3.1. Types of Sample

Often, it is impractical survey every subject in an entire population. Instead, we survey a much smaller sample, and see what we can infer about the population from that sample. However, for the inference to be reliable, the sample has to be carefully selected to ensure it is representative of the population.

3.1.1. Simple Random Sample

The simplest type of sample to draw inference from is the simple random sample. In this type of sampling, every individual has exactly the same probability of being selected for our sample. So if we have a population of 100,000, and want to sample 1,000 people from it, every person in the population should have a 1 in 100 chanve of being selected.

There are two big issues with simple random sampling. One is the fact that you need to have a list of everyone in your population. This may be hard to obtain. For example, GP lists are often used as a sampling frame, but not everyone is registered with a GP, so the sampling frame is incomplete. The same is true for electoral registers. This means that the population you are investigating may not be the population you wish to investigate: you are not sampling from "People living in this area", but "People registered with a G.P. in this area" or "People registered to vote in this area". Whether this is a problem or not depends on how different the population we are actually studying is from the population we intended to study.

The second problem is refusals. In a simple random sample, everyone has the same probability of being in the sample. However, some people have already assigned themselves a 0% probability of being in the sample by deciding they don't want to participate. Again, you can think of this as changing the population you are studying: "People from the population we wish to study who are happy to take part in our study". This may or may not be an issue, depending on the extend to which those who don't take part differ from those who do.

3.1.2. Stratified Sample

A stratified sample can be thought of as a number of simple random samples. The population to be studied is divided into strata, and a simple random sampling is performed within each stratum. The probability of being in the sample is the same for every individual within a given stratum, but can differ between strata.

A stratified sample can be useful if you wish to estimate something that varies widely between strata.

For example, if you are interested in the prevalence of a condition that is very common in older subjects and rare in younger ones, you may want to recruit more younger subjects in order to get a equally precise estimate of the prevalence in the young as you would get in the older subjects.

3.1.3. Cluster Sample

Another commonly used type of sampling is cluster sampling: in which groups of subjects (clusters) are sampled rather than individuals. There are a number of situations where this can be useful.

1. There is no sampling frame for individuals, but there is for clusters. For example, you may have a list of houses, but not a list of the residents of those houses. It would then be impossible to sample individuals, but it is possible to sample houses.
2. It is cheaper and easier to recruit a number of people at the same time. Various addons to the ten-yearly census uses this method: they phone a landline and ask about everybody living in that house.
3. It may be that an intervention can not be allocated to individuals, only to clusters of individuals. For example, to assess the impact of posters in G.P.'s waiting rooms, you cannot randomly allocate individuals to look at or not look at the poster, but you can randomly allocate the posters to some waiting rooms and not to others

Cluster sampling requires special methods for analysing the data collected, which we will not mention in this course.

3.1.4. Other Types of Sample

There are other types of sample that you may hear mentioned, but I do not recommend using them. One is the quota sample, beloved of market researchers, where you have a fixed number of subjects to recruit to each subgroup. This can be thought of as a stratified sample, with the groups to which you are recruiting as the strata. However, you do not know what the sampling probability is for any stratum, so you don't know what proportion of the population each stratum represents. With a stratified sample, if you want to calculate a population statistic, you can reweight each stratum according to the size it should have in the population, rather than the size it has in the sample, and hence make the sample representative of the population. With a quota sample you can't do that.

In a systematic sample, you randomly select a starting point in your sampling frame, then take every n^{th} subject until you have your desired sample size. This can work, unless there is clustering or periodicity in your sampling frame. For example, if the sampling frame is in alphabetical order by surname, then you are unlikely ever to select two subjects from the same family, whilst in a simple random sample that would happen. That means that characteristics that are shared in a family, such as diet, will vary more in a systematic sample than in a simple random sample. You could fix this problem easily if you have an electronic sampling frame, by assigning each record a random number and sorting on that number, to put the sampling frame into a random order. Then a systematic sample would be a random sample.

One last type of sample to look out for is the convenience sample. This is where you simply recruit people who are easy to find. For example, when testing a questionnaire, you may ask your colleagues to complete it for you, and see how long it takes. However, your colleagues are likely to have more education than the population average, and so complete the questionnaire quicker. Provided that you are aware of the limited population you are recruiting from, this is not a problem, but it is not possible to make inference about the general population from a convenience sample.

3.2. From Sample to Population

The purpose of collecting sample data is to see what it can tell us about the population. We can be interested in point estimates ("What is our best guess at the population value ?"), interval estimates ("Can we be fairly certain the population value lies within a given range ?") and hypothesis tests ("Can we be fairly certain the population value is not one particular value ?"). In this chapter, we are only concerned with the first two questions, the next chapter covers the third one.

In order to help keep track of whether we are talking about population statistics or sample statistics, we generally use Greek letters for population values and Roman letters for sample values. The most commonly used notation is given in Table 3.1

Parameter	Population Value	Sample Value
Mean	μ	m
Standard deviation	σ	s
Proportion	π	p

Table 3.1.: Notation for Population and Sample Parameters

3.2.1. Estimating a Mean

Suppose that we are interested in the mean value in the population (μ). The mean we calculate from the sample should be close to the population value, but will not be exactly equal to it. If we were to take a number of samples, we would expect the means to be different in each sample. However, we would expect the sample means to be close to the population mean, and to vary less in larger samples. This is illustrated in the example below.

Example of estimating a mean

For this example, the population consists of 10,000 individuals. The variable we are interested in, x, takes integer values 0, 1, . . . , 9, and 1,000 individuals have each of the 10 possible values. The population distribution of x is shown in Figure 3.1.

From this population, samples of size 5, 25 and 100 were repeatedly taken, and the mean of each sample was calculated. Figure 3.2 shows the distribution of the sample means for each sample size, each histogram based on 2,000 samples.

It is clear from 3.2 that the distibution of the sample means is centred on the same value, 4.5, irrepective of the sample size. It is also clear that the distibution varies less as the sample size increases. Finally, the distribution of the sample mean appears to be normal, particularly for the larger sample sizes, despite the fact that the distribution of the original variable (in Figure 3.1) is far from normal.

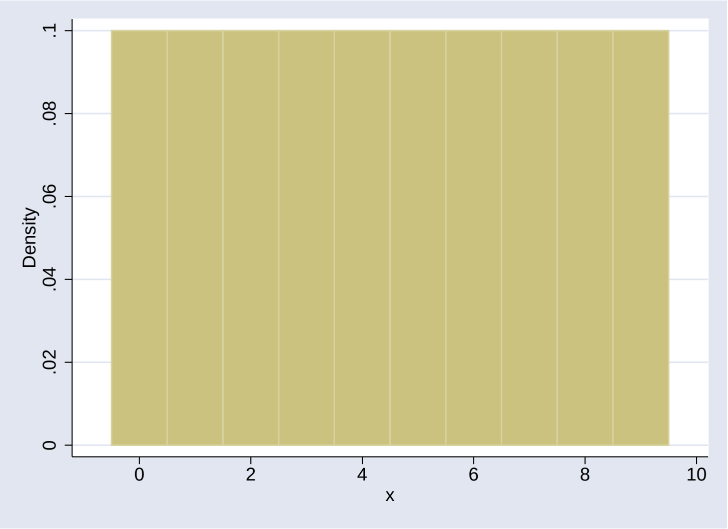
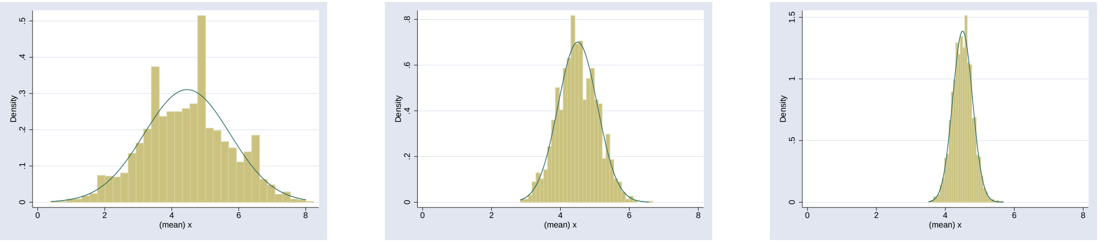


Figure 3.1.: Population Distribution for Estimating Mean Example



(a) Sample size 5

(b) Sample size 25

(c) Sample size 100

Figure 3.2.: Distributions of sample means of different sizes

The Sampling Distribution of a mean

The distribution of the values taken by the sample mean ($\bar{x}$) as samples (of a given size) are repeatedly taken from a fixed population is called the *sampling distribution* of $\bar{x}$. All statistical inference (working out what our sample tells us about the population from which it was taken) depends on this distribution. There are three important properties of the sampling distribution of the mean:

$\boldsymbol{E(\bar{x}) = \mu}$ i.e. on average, the sample mean is the same as the population mean.

Standard Deviation of $\bar{x}$ = $\frac{\sigma}{\sqrt{n}}$ i.e the uncertainty in $\bar{x}$ increases with σ, decreases with n. The standard deviation of the sampling distribution of the mean is also called the **Standard Error**

$\bar{x}$ is normally distributed This is true whether or not x is normally distributed, provided n is sufficiently large. Thanks to the *Central Limit Theorem*. This is clearly shown in Figure 3.2: the sampling distributions of the means are all normal, despite the fact that the distribution of x, shown in Figure 3.1, is clearly *not* normal.

The Standard Error

The standard error is the most important concept in statistics[a]. The standard error is the standard deviation of the sampling distribution. We will shortly see how the standard error can be used to calculate confidence intervals for population parameters, and in the next chapter how it can be used to perform hypothesis tests.

3.2.2. Estimating the Variance σ

The standard error of $\bar{x}$ is defined in terms of the population variance, σ^2. However, we do not know what this is, and need to estimate if from our sample.

In a population of size N, the variance of x is given by

$$\sigma^2 = \frac{\Sigma(x_i - \mu)^2}{N} \tag{3.1}$$

with the summation being over every observation in the population. However, if we only have a sample from this population, we do not know the population mean μ. We can estimate it as the sample mean, $\bar{x}$, but this value will depend on the actual values we observe in the sample. Since $\bar{x}$ is the number that minimises $\Sigma(x_i - \bar{x})^2$, this sum will be smaller than the value we really want, $\Sigma(x_i - \mu)^2$. Hence, $s^2 = \frac{\Sigma(x_i - \bar{x})^2}{n}$ will be smaller than σ^2. However, dividing by $n - 1$ rather than n is sufficient to correct this underestimate, and $s^2 = \frac{\Sigma(x_i - \bar{x})^2}{n-1}$ is an unbiased estimate of the population variance σ^2, and can be used to calculate the standard error of the mean without problems.

3.2.3. Estimating a Proportion

Estimating a proportion works in exactly the same way as estimating a mean. In fact, estimating a proportion *is* estimating a mean: it is estimating the mean of a variable which takes the value 0 for all observations in which the characteristic of interest is absent, and 1 for all observations in which it is present. Adding all these values gives the number of observations with the characteristic, dividing by the sample size gives the proportion. Furthermore, we have seen that the sampling distribution of $\bar{x}$ is normal (in sufficiently large samples), even if x is not.

3.2.4. Confidence Intervals

We have seen that the sampling distribution of a mean is normal, with mean μ and standard deviation $\frac{\sigma}{\sqrt{n}}$. The sample mean that we have can be thought of as a single observation drawn at random from this distribution. The $2.5th$ centile of this distribution is $\mu - 1.96\frac{\sigma}{\sqrt{n}}$, and the 97.5^{th} centile is $\mu + 1.96\frac{\sigma}{\sqrt{n}}$. Hence, there is a 95% chance that our sample value will lie in between these two values.

So, although we don't know the *exact* value of μ, we know that there is a 95% chance that it lies within $1.96 \times \frac{s}{\sqrt{n}}$, and we do know both s and n. Hence, we can calculate an interval, $\bar{x} - 1.96\frac{s}{\sqrt{n}}$ to

[a] At least in *frequentist* statistics, which is the only type of statistics this course is concerned with

$\bar{x}+1.96\frac{s}{\sqrt{n}}$, in which we are 95% certain that the population will lie. This is known as a 95% confidence interval.

In theory, we can calculate a confidence interval with any chosen probability that the population mean lies within it. The less certain we need to be, the narrower the confidence interval can be. In practice, the 95% level is always used.

Confidence Interval Example

As an example of calculating a 95% confidence interval, suppose we measured serum albumin in a sample fo 216 patients with primary biliary cirrhosis. The mean value in the sample was 34.46 g/l, and the values had a standard deviation of 5.84 g/l.

To calculate a 95% confidence interval for the mean serum albumin level in primary biliary cirrhosis patients, we first have to calculate the standard error. This is

$$\begin{aligned}
\text{Standard Error of } \bar{x} &= \frac{5.84}{\sqrt{216}} \\
&= 0.397 \\
\Rightarrow \quad 95\% \text{ Confidence Interval} &= 34.46 \pm 1.96 \times 0.397 \\
&= (33.68, 35.24)
\end{aligned}$$

So we can be 95% certain that the population value is between 33.7 and 35.2 g/l.

Confidence Interval For a Proportion

If the sample size is large enough, the sampling distribution of a proportion p will be approximately normal, and its standard error in a sample of size n will be $\sqrt{\frac{p(1-p)}{n}}$. As a rule of thumb, if both $p \times n$ and $(1-p) \times n$ are bigger than 5, the sample size is large enough to assume normality.

Confidence Intervals in Stata

Stata has a command `ci` that can be used to calculate confidence intervals around a mean or proportion. If you want to calculate a confidence interval around a proportion in a small sample, where the normal approximation may not be very accurate, there is a `binomial` option which will use the exact binomial distribution to calculate the confidence interval.

3.2.5. Sample Size Calculations

In general, the aim of any study is to estimate a statistic: a mean, proportion, relative risk, hazard ratio or some other characteristic of a population. The larger our study is, the more precisely we can estimate this statistic. We can choose the size of study we need to provide adequate precision if we know how the sampling distribution depends on the sample size, and we have a definition of adequate, i.e. what is the widest confidence interval we will accept.

Given that the confidence interval is $\bar{x} \pm \frac{1.96\sigma}{\sqrt{n}}$, its width depends only on n and σ. If we have an estimate of σ, then we can choose n to give us a confidence interval of any width.

Suppose that we want the width of our confidence interval to be $2W$, so that the interval itself is $\bar{x} \pm W$ Then

$$\begin{aligned} W &= 1.96 \times \text{Standard Error} \\ &= 1.96 \times \frac{\sigma}{\sqrt{n}} \\ \Rightarrow \quad W^2 &= \frac{1.96^2\sigma^2}{n} \\ \Rightarrow \quad n &= \left(\frac{1.96\sigma}{W}\right)^2 \end{aligned}$$

So collecting a sample of at least $\left(\frac{1.96\sigma}{W}\right)^2$ subjects will provide a 95% confidence interval that is no wider than $2W$.

Sample Size Calculation Example

Suppose that using the primary biliary cirrhosis data in Section 3.2.4, we want to be know the population mean serum albumin level to within 0.5 g/l. How many patients would we need to study (assuming a standard deviation of 5.84 g/l) ?

$$\begin{aligned} W &= 0.5 \\ \sigma &= 5.84 \\ \Rightarrow \quad n &= \left(\frac{1.96\sigma}{W}\right)^2 \\ &= \left(\frac{1.96 \times 5.84}{0.5}\right)^2 \\ &\approx 524 \end{aligned}$$

So we would need a sample size of at least 524.

3.3. Further Reading

Further Reading

[1] J. M. Bland. The tyranny of power: is there a better way to calculate sample size? *BMJ (Clinical research ed.)*, 339(november):b3985, jan 2009.

[2] D. G. Altman and J. M. Bland. Uncertainty beyond sampling error. *BMJ (Online)*, 349, 2014.

3.4. Practical on Confidence Intervals

3.4.1. Generating Random Samples

In this part of the practical, you are going to repeatedly generate random samples of varying size from a population with known mean and standard deviation. You can then see for yourselves how changing the sample size affects the variability of the sample mean. If you want to store your results in an Excel spreadsheet, double-click here to open a suitable one (Link does not work in a browser).

1. Ensure there is no data in stata's memory by entering the command `clear`
2. Set the sample size to 5 with the command `set obs 5`
3. Generate a variable `x` with a mean of 0 and a standard deviation of 1, using the command `generate x = invnorm(uniform())`
4. Obtain the mean of `x` in this sample from the command `summarize x`
5. Record the mean for this sample.
6. Repeat steps 1-5 10 times until the first column of the spreadsheet is full.
7. Now repeat the procedure a further ten times, but using `set obs 25` in step 2, to complete column 2 of the spreadsheet with the means of samples of size 25.
8. Repeat the procedure a further ten times, but using the command `set obs 100` in step 2 to produce means for samples of size 100.
9. Now calculate the mean and standard deviation of the values in each column.
   If you have used the Excel spreadsheet, it will do it for you. Otherwise, the easiest way to do this is to use the commands

   ```
   clear
   edit
   ```

   to get a spreadsheet view of an empty stata dataset, and type the values in as three columns. If you have stored them in a spreadsheet, you could cut and paste them, or use the command `import excel`: use the `help` command to find out how. Stata will call the three variables `var1`, `var2` and `var3` by default (unless you cut and paste the variable names from the spreadsheet), but you can rename them by double clicking on the name, and typing a new name in the dialog box that appears. When you have entered the data, click on the cross in the right hand top corner to close the spreadsheet view. (Once upon a time, stata would not carry out commands when a spreadsheet view was open. Current statas will, but the sheet will hide the results window).

   Now you can use the command

   ```
   summarize var1 var2 var3
   ```

   to get the mean and standard deviation of these variables.

Further Reading

3.4.2. Means

If the standard deviation of the original distribution is σ, then the standard error of the sample means is σ / $\sqrt{n}$, where n is the sample size.

4.1 If the standard deviation of measured heights is 9.31 cms, what will be the standard error of the mean in:

i a sample of size 49 ? ......

ii a sample of size 100 ? ......

4.2 Imagine we only had data on a sample of size 100, where the sample mean was 166.2cm and the sample standard deviation was 10.1cm.

i Calculate the standard error for this sample mean (using the sample standard deviation as an estimate of the population standard deviation).

....

ii Calculate the interval ranging 1.96 standard errors either side of the sample mean.

....

4.3 Imagine we only had data on a sample size of 36 where the sample mean height was 163.5 cm and the standard deviation was 10.5cm.

i Calculate the 95% confidence interval for the sample mean.

....

4.4 Figure 3.3 is a histogram of measured weight in a sample of 100 individuals.

i Would it be better to use the mean and standard deviation or the median and interquartile range to summarize this data ?

....

ii If the mean of the data is 69.69kg with a standard deviation of 12.76kg, calculate a 95% confidence interval for the mean.

....

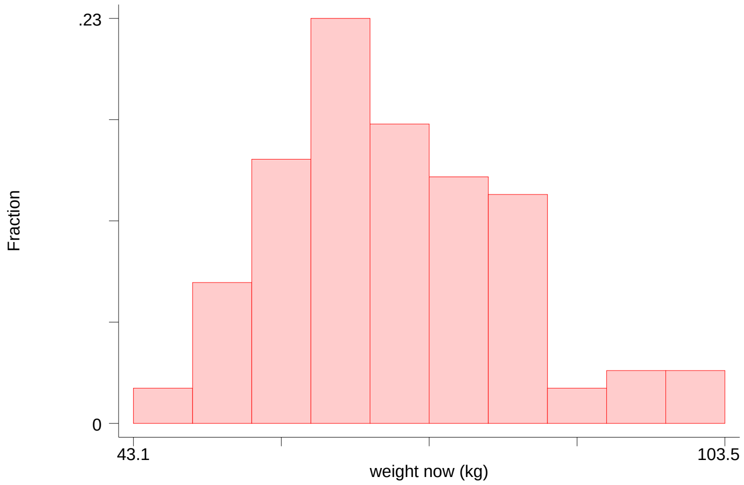


Figure 3.3.: Weights in a random sample of 100 women

3.4.3. Proportions

Again using our height and weight dataset of 412 individuals, 234 (56.8%) are women and 178 (43.2%) are men.

If we take a number of smaller samples from this population, the proportion of women will vary, although they will tend to be scattered around 57%. Figure 3.4 represents 50 samples, each of size 40.

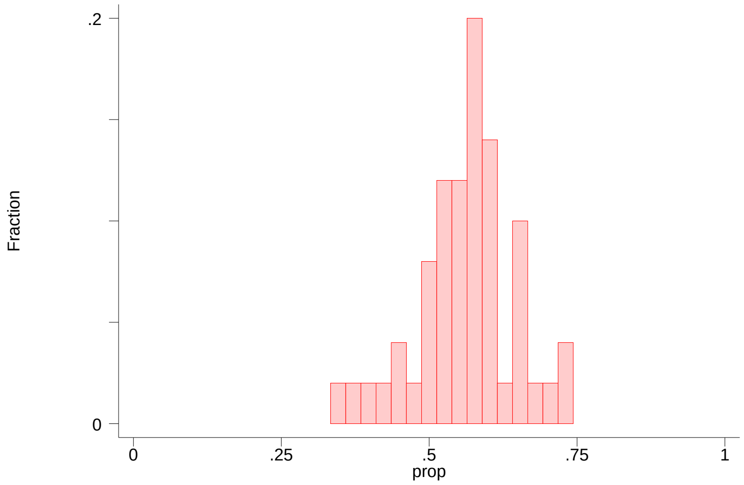


Figure 3.4.: Proportion of Women in 50 samples of size 40

4.5 What would you expect to happen if the sample sizes were bigger, say n=100 ?

. . . .

. . . .

. . . .

4.6 In a sample of 40 individuals from a larger population, 25 are women. Calculate a 95% confidence interval for the proportion of women in the population.

. . . .

. . . .

Note: When sample sizes are small the use of standard errors and the normal distribution does not work well for proportions. This is only really a problem if p (or (1-p)) is less than 5/n (i.e. there are less than 5 subjects in one of the groups).

4.7 From a random sample of 80 women who attend a general practice, 18 report a previous history of asthma.

i Estimate the proportion of women in this population with a previous history of asthma, along with a 95% confidence interval for this proportion.

. . . .

ii Is the use of the normal distribution valid in this instance ?

. . . .

. . . .

4.8 In a random sample of 150 Manchester adults it was found that 58 received or needed to receive treatment for defective vision. Estimate the proportion of adults in Manchester who receive or need to receive treatment for defective vision, a 95% confidence interval for this proportion.

i Proportion

. . . .

ii 95% Confidence interval

. . . .

3.4.4. Confidence Intervals in Stata

Load the blood pressure data in its wide form into stata with the command

```
sysuse bpwide
```

This is fictional data concerning blood pressure before and after a particular intervention.

4.9 Use the command

```
histogram bp_before
```

to see if this variable is normally distributed. What do you think ?

. . . .

4.10 Create a new variable to measure the change in blood pressure and find its mean value with the commands

```
generate bp_diff = bp_after - bp_before
summarize bp_diff
```

What is the mean change in blood pressure ? . . . . . .

4.11 Create a confidence interval for the change in blood pressure with the command

```
ci bp_diff
```

Does the intervention reduce blood pressure in general ?

. . . .

4.12 Look at the histogram of changes in blood pressure using the command

```
histogram bp_diff
```

Does this confirm your answer to the previous question ?

. . . .

4.13 Create a new variable to measure whether blood pressure went up or down in a given subject using the command

```
generate down = bp_after < bp_before
```

Use the `tabulate` command to see how many subjects, and what proportion, showed a decrease in blood pressure.

. . . .

4.14 Create a confidence interval for the proportion of subjects showing a decrease in blood pressure with the command

```
ci down, binomial
```

Does this confirm the effect of the intervention on blood pressure ?

. . . .

4. Hypothesis Testing and Power

4.1. Formulating a hypothesis test

The two main forms of inference in frequentist statistics are confidence intervals and hypothesis tests. As we saw in the previous chapter, confidence intervals give a range in which the true population value is likely to lie. A hypothesis test can be thought of as doing the opposite: it presents the strength of evidence the the true population value is not one particular value.

4.1.1. Components of Hypothesis test

To perform a hypothesis test, we begin by creating the *Null Hypothesis*. This proposes a possible value for the true population value. We then calculate a *test statistic* from our sample data, can calculate the probability of seeing that value or one more extreme (further from the the null hypothesis value) if the null hypothesis is true. This probability is called the p-value.

As originally devised by Fisher, the p-value was intended to provide an informal assessment of the strength of the evidence against a particular null hypothesis. However, some researchers were not happy with this. They wanted a decision-making tool: for example, a p-value may say "there is good evidence that this drug is effective", but is the evidence good enough to justify introducing the drug, or not. Neyman and Pearson extended the idea of hypothesis testing by introducing the notion of rejecting the null hypothesis if the p-value is sufficiently small. This introduced the concept of statistical significance: if the p-value is less than 0.05, we say that the test is significant at the 5% level.

Null Hypothesis

The null hypothesis is the hypothesis that we wish to test, and is generally that there is no association between two variables. For example, it may be that the prevalence of a disease is the same in two different groups, or that there is no difference in outcome between the two arms of a drug trial.

The alternative hypothesis is simply "The null hypothesis is untrue", and hence can cover a wide range of possibilities. For example, if the null hypothesis were "there is no difference in prevalence between these two groups", the alternative hypothesis would be "there is a difference in prevalence between these two groups"

Sometimes, people will present a one-sided alternative hypothesis. For example, a drug company way define their null hypothesis as "our drug is no better than our competitor's" and their alternative hypothesis is as "our drug is better than our competitor's". It is rarely justified to use a one-sided test: we would be interested in knowing if their drug was worse than the competitor's, even if they would not.

Generally, a one-sided test will have a smaller p-value than a two-sided test, which is often the motivation for doing it, so seeing the words "one-sided test" should ring alarm bells.

However, there are times when a one-sided test is entirely appropriate. For example, we will meet the χ^2-test for testing for an association between two categorical variables in Chapter 7. This test measures the total difference between the observed numbers in each cell of a table, and the expected numbers in each cell. If the observed numbers are unusually close to the expected numbers if the null hypothesis of no association is true, that does not provide evidence against the null hypothesis. Only if the observed numbers are unusually far from the expected values is there evidence against the null hypothesis, so a one-sided test is entirely appropriate there.

Test statistics

In order to be able to test the null hypothesis, we need to have a statistic whose sampling distribution if the null hypothesis were true is known. For example, it may be a difference in prevalence between two groups, in which case the population value under the null hypothesis would be 0. This value would therefore be the expected value of the sampling distribution.

The T Distribution Suppose that the sampling distribution of our test statistic S has a mean μ and standard deviation σ if the null hypothesis is true. Then the statistic $T = \frac{S-\mu}{\sigma}$ will have a mean of 0 and a standard deviation of 1. This is known as the standard normal distribution. We very often work with "standardised" test statistics.

Figure 4.1 shows a standard normal distribution. The shaded area consists of the lowest 2.5% of the distribution and the highest 2.5%. There is a 5% chance that the test statistic will lie in the shaded area if the null hypothesis is true. This is considered sufficiently unlikely that it provides evidence against the null hypothesis.

However, we again run into the problem that we don't know σ, and are working with a sample-based estimate of it s. So the statistic that we calculate is $T = \frac{S-\mu}{s}$, and this does not follow a standard normal distribution: the distribution it follows is called a t-distribution on $n-1$ degrees of freedom (where n is the sample size). This is where the expression "t-test" comes from.

For small values of n, the tails of the t-distribution are larger than the tails of the normal distribution, so extreme values of the statistic are less uncommon (see Figure 4.2. However, as the sample size increases, the t-distribution gets closer and closer to the normal distribution, and they are practically indistinguishable once the sample size increases beyond 100.

Some test statistics that we may be interested in do not follow a normal distribution, such as the χ^2 statistic or the Mann-Whitney U statistic. However, as long as the centiles for the distribution of these statistics are known, we can still calculate a p-value based on that statistic.

4.1.2. Examples

Here are a couple of examples of the various components of a hypothesis test

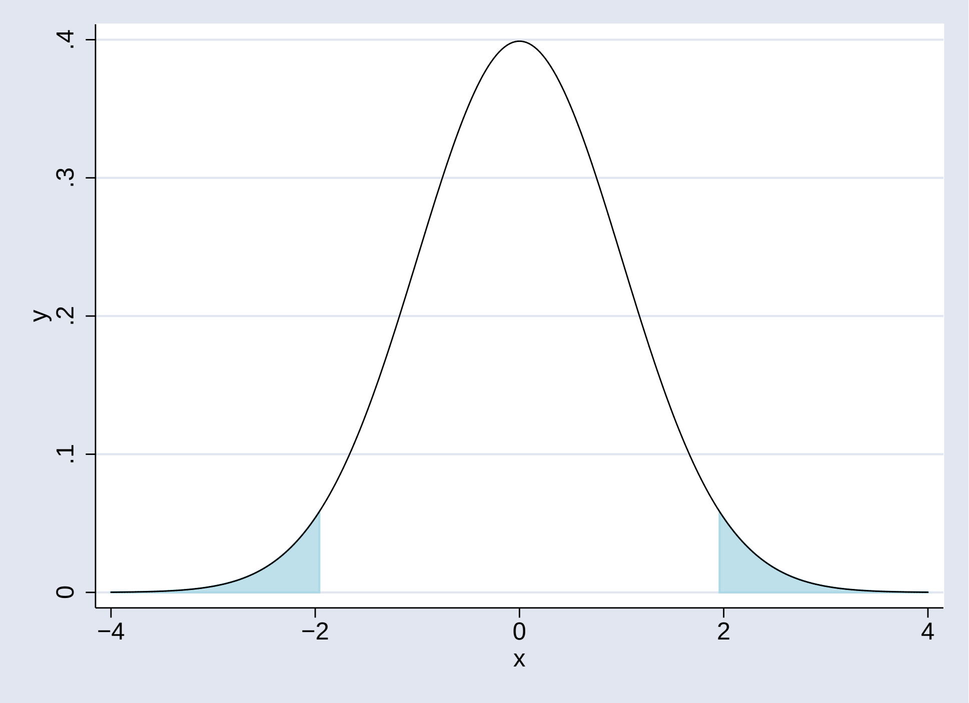


Figure 4.1.: A Standard Normal Distribution

Does height differ between men and women ?

Null hypothesis On average, men and women are the same height

Alternative Hypothesis One gender tends to be taller than the other.

Test Statistic Difference in mean height between men and women.

One-Sided Hypotheses

- Men are taller than women
- Women are taller than men

Which is more popular, Coke or Pepsi ?

Null hypothesis Equal numbers of people prefer Coke and Pepsi

Alternative Hypothesis Most people prefer one drink to the other

Test Statistic Several possibilities:

- Difference in proportions preferring each drink
- Ratio of proportions preferring each drink

One-Sided Hypotheses

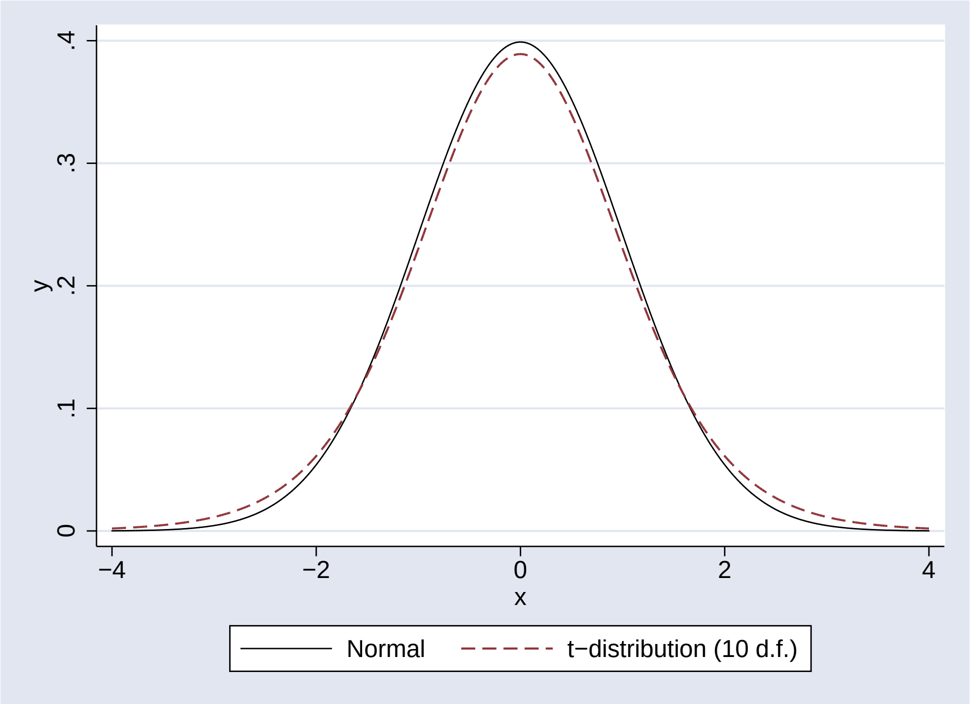


Figure 4.2.: T-Distribution compared to a Normal Distribution

- More people prefer Coke
- More people prefer Pepsi

4.2. Interpreting a hypothesis test

Interpreting a hypothsesis test is not easy. The American Statistical Association did a survey of the 25 most common errors in statistics, and 20 of them involve hypothesis testing. [1] In particular, it is not easy to grasp what the p-value means (and more importantly, what it does not mean).

4.2.1. p-values

The p-value that is produced by a hypothesis test is the probability of seeing a test statistic at least as far from the null value as was seen in our sample if the null hypothesis is true. If the p-value is small, it was unlikely that our sample data would have been produced if the null hypothesis were true. But our data *was* produced, and so it provided evidence that the null hypothesis is not true.

The p-value must lie between 0 and 1. The smaller it is, the less likely it is that the data could have been produced if the null hypothesis were true. Conventionally, we say that if the p-value is less than 0.05, there is some evidence against the null hypothesis: the effect is “statistically significant at the 5% level”. However, 0.05 is a totally arbitrary value: if the p-value is 0.055, the evidence against the null hypothesis is very nearly as strong as if the p-value were 0.045. For this reason, always present *exact* p-values (to 1 or 2 significant figures).

Large p-values cannot be thought of evidence that the null hypothesis is true. This is because the p-value is affected by both the size of the effect and the size of the study. A large p-value can happen if the null hypothesis it true, or if the study is too small to detect the actual difference between the truth and the null hypothesis.

It may help to think of a hypothesis test as a politician. You ask it a simple question, "Is the null hypothesis true ?", which only has two possible answers, "yes" or "no". And yet you never get the answer "yes" or "no". If $p < 0.05$, the answer is "probably not", and if $p > 0.05$, the answer is "no comment". Interpreting $p > 0.05$ as "no" leads to all sorts of problems

4.2.2. Errors

There are two ways that we can get a hypothesis test wrong. Either the null hypothesis is true, and we conclude that it isn't (Type 1 error), or the null hypothesis is not true, but we fail to find any evidence against it (Type II error).

Type I Error

One time in every 20 that the null hypothesis is true, we will conclude that it isn't (at the 5% significance level). The smaller the p-value, the less likely it is that we are making a type I error. If we test lots of null hypothesis at the same time, there is a good chance that at least one will produce a p-value < 0.05. This can be corrected for in a number of ways: Bonferroni's correction is the most well-known. There is some debate about when correction for multiple testing is helpful, but it essential that you inform the reader of your study how many hypothesis tests were performed, not just how many were significant.

Type II Error

It may be that the null hypothesis is not true, but that we fail to find evidence against it. This is more likely if the study is small, which it is why it is so important to ensure that a study is sifficiently large to be worth carrying out.

4.2.3. Inappropriate Hypothesis tests

Hypothesis tests are commonly performed in situations in which they are totally inappropriate. For example, in a randomised clinical trial, the null hypothesis (both arms of the trial are random samples from the same population) is true. There is no need to test it: any differences between the arms arose by chance. Even if $p < 0.05$, and you are unlikely to see such a large difference by chance, the entire process of designing a clinical trial is aimed at ensuring that the null hypothesis is true.

Even in observational studies, p-values for the differences in potential confounders between groups being compared are unhelpful. The confounding effect depends solely on the magnitude of the difference in the confounder between the two groups. However, the p-value depends on the sample size as well. You could have two studies with exactly the same confounding effect but different p-values, or very different confounding effects with the same p-value, if the sample sizes of the two studies are different. That will not stop journals asking for p-values for these differences, unfortunately.

4.3. Common types of hypothesis test

4.3.1. One-sample t-test

The simplest type of hypothesis test is comparing a sample mean to a null hypothesis value. This is often referred to as a "one-sample t-test". The test statistic in this case is

$$T = \frac{\bar{x} - \mu}{S.E.(x)}$$

and T can be compared to a t-distribution on $n - 1$ degrees of freedom.

For example, consider the following data consisting of uterine weights (in mg) for a sample of 20 rats. Previous work suggests that the mean uterine weight for the stock from which the sample was drawn was 24mg. Does this sample confirm that suggestion ?

Weights 9, 14, 15, 15, 16, 18, 18, 19, 19, 20, 21, 22, 22, 24, 24, 26, 27, 29, 30, 32

$\bar{x}$ = 21.0

S.D.(x) = 5.912

In this case, the standard error of $\bar{x}$ is $\frac{5.912}{\sqrt{20}} = 1.322$, so

$$\begin{aligned} T &= \frac{\bar{x} - 24.0}{S.E.(x)} \\ &= \frac{21.0 - 24.0}{1.322} \\ &= -2.27 \end{aligned}$$

Comparing -2.27 to a t-distribution on 19 degrees of freedom gives a p-value of 0.035 I.e if the stock had a mean uterine weight of 24mg, and we took repeated random samples, less than 4 times in 100 would a sample have such a low mean weight.

This test can be performed in stata with the command `ttest`. Assuming the data were stored in a variable called `x`, the necessary syntax and resulting output would be:

```
. ttest x = 24

One-sample t test

------------------------------------------------------------------------------
Variable |     Obs        Mean    Std. Err.   Std. Dev.   [95% Conf. Interval]
---------+--------------------------------------------------------------------
       x |      20          21    1.321881     5.91163    18.23327    23.76673
------------------------------------------------------------------------------
Degrees of freedom: 19

                              Ho: mean(x) = 24

     Ha: mean < 24             Ha: mean != 24              Ha: mean > 24
       t =  -2.2695                t =  -2.2695              t =  -2.2695
   P < t =   0.0175          P > |t| =   0.0351          P > t =   0.9825
```

Note that the one sided alternative test that the weight was lower than 24mg has a p-value equal to half of the two-sided test p-value.

4.3.2. Two-sample t-test

The two-sample t-test is used for comparing the means in two groups.

```
. ttest nurseht, by(sex)

Two-sample t test with equal variances

------------------------------------------------------------------------------
   Group |     Obs        Mean    Std. Err.   Std. Dev.   [95% Conf. Interval]
---------+--------------------------------------------------------------------
  female |     227     159.774    .4247034    6.398803    158.9371    160.6109
    male |     175    172.9571    .5224808    6.911771    171.9259    173.9884
---------+--------------------------------------------------------------------
combined |     402    165.5129    .4642267    9.307717    164.6003    166.4256
---------+--------------------------------------------------------------------
    diff |           -13.18313    .6666327               -14.49368   -11.87259
------------------------------------------------------------------------------
Degrees of freedom: 400

                  Ho: mean(female) - mean(male) = diff = 0

     Ha: diff < 0               Ha: diff != 0              Ha: diff > 0
       t = -19.7757                t = -19.7757              t = -19.7757
   P < t =   0.0000          P > |t| =   0.0000          P > t =   1.0000
```

4.3.3. Comparing proportions

We can also compare proportions between two groups. The stata command to do this is `cs`, and the output from the command is shown below. There are are a number of different ways to test for a difference in proportions: the null hypothesis could be that the risk difference is 0, or that the risk ratio is 1.

```
. cs back_p sex

                 | sex                    |
                 |   Exposed   Unexposed  |      Total
-----------------+------------------------+------------
           Cases |       637         445  |       1082
        Noncases |      1694        1739  |       3433
-----------------+------------------------+------------
           Total |      2331        2184  |       4515
                 |                        |
            Risk |  .2732733    .2037546  |   .2396456
                 |                        |
                 |      Point estimate    |    [95% Conf. Interval]
                 |------------------------+------------------------
 Risk difference |         .0695187       |    .044767    .0942704
      Risk ratio |         1.341188       |   1.206183    1.491304
 Attr. frac. ex. |         .2543926       |   .1709386     .329446
 Attr. frac. pop |         .1497672       |
                 +-------------------------------------------------
                               chi2(1) =    29.91  Pr>chi2 = 0.0000
```

4.4. Power calculations

4.4.1. How power calculations work

4.4.2. Power calculations in stata

4.5. Hypothesis tests and confidence intervals

Where possible, confidence intervals should be preferred to hypothesis tests. The confidence interval conveys more information than the hypothesis test (including whether the hypothesis test would be significant at the 5% level in many cases). It is far more useful to know a range of values within which the population value is likely to lie, than a single value that the population value is unlikely to be. There is a movement to replace p-values with confidence intervals both in the epidemiology literature and amongst statisticians.

4.6. Further Reading

Further Reading

[1] S. Greenland, S. J. Senn, K. J. Rothman, J. B. Carlin, C. Poole, S. N. Goodman, and D. G. Altman. Statistical tests, P values, confidence intervals, and power: a guide to misinterpretations. *European Journal of Epidemiology*, 31(4):337–350, 2016.

[2] R. L. Wasserstein and N. A. Lazar. The ASA's Statement on p -Values: Context, Process, and Purpose. *The American Statistician*, 70(2):129–133, 2016.

[3] R. Nuzzo. Statistical errors. *Nature*, 506:150–152, 2014.

[4] A. Gelman, J. Hill, and M. Yajima. Why We (Usually) Don't Have to Worry About Multiple Comparisons. *Journal of Research on Educational Effectiveness*, 5(2):189–211, 2012.

[5] T. L. Lash. The Harm Done to Reproducibility by the Culture of Null Hypothesis Significance Testing. *American Journal of Epidemiology*, 186(6):627–635, 2017.

[6] P. . Sprent. Some Problems of Statistical Consultancy. *Journal of the Royal Statistical Society, Series A.*, 133(2):139–165, 1970.

4.7. Practical on Hypothesis Testing and Power

4.7.1. Inference about a proportion

Out of 80 women in a random sample of women in Manchester, 13 were asthmatic; this could be used to calculate a 95% confidence interval for the proportion of women in Manchester with asthma. This confidence interval could be compared to the suggested prevalence of 20% in Northern England. An alternative approach would be to test the hypothesis that the true proportion, π, is 0.20.

7.1 What is the expected proportion of women with asthma under the null hypothesis ? ……

7.2 What is the observed proportion of women with asthma ? ……

7.3 What is the standard error of the expected proportion (remember from last week that the standard error of a proportion p is given by

$$\sqrt{\frac{p(1-p)}{n}}$$

……

7.4 The appropriate test statistic, T, is given by the formula:

$$\frac{\text{observed proportion - expected proportion}}{\text{standard error of proportion}}$$

Calculate T. ……

7.5 T should be compared to a t-distribution with how many degrees of freedom ? ……

7.6 From tables for the appropriate t-distribution, the corresponding p-value is 0.4. Is is reasonable to suppose that these women are a random sample from a population in which the prevalence of

asthma is 20% ? ……

4.7.2. More inference about a proportion

In the sample heights and weights we have looked at, there were 412 individuals of whom 234 were women. We wish to test that there are equal numbers of men and women in our population.

7.7 What is the null hypothesis proportion of women ? ......

7.8 What is the observed proportion of women ? ......

7.9 What is the null hypothesis standard error for the proportion of women ? ......

7.10 What is an appropriate statistic for testing the null hypothesis ? ......

4.7.3. Inference about a mean

Load `htwt.dta` into stata with the commands (each command needs to be entered on a separate line).

```
 global datadir http://personalpages.manchester.ac.uk/staff/mark.lunt/stats
use $datadir/2_summarizing_data/data/htwt.dta
```

We wish to test whether the mean height is the same in men and women.

7.11 What is the null hypothesis difference in height between men and women ?

....

7.12 Use the command `ttest nurseht, by(sex)` to test whether the mean height differs between men and women.

7.13 What is the mean height in men ? ......

7.14 What is the mean height in women ? ......

7.15 What is the mean difference in height between men and women, with its 95% confidence interval ?

....

7.16 Which of the three hypothesis tests is the appropriate one in this instance ?

. . . .

7.17 What is the p-value from the t-test ? . . . . . .

7.18 What would you conclude ?

. . . .

4.7.4. Two-sample t-test

Compare BMI (based on the measured values, i.e. `bmi`) between men and women in `htwt.dta`, using the command `ttest bmi, by(sex)`.

7.19 Is there a difference in BMI between men and women ? . . . . . .

7.20 What is the mean difference in BMI between men and women and its 95% confidence interval.

. . . .

7.21 Is there a difference in the standard deviation of BMI between men and women ? (This can be tested with the command `sdtest bmi, by(sex)`

. . . .

7.22 If there is, repeat the t-test you performed above, using the `unequal` option. Are your conclusions any different ?

. . . .

4.7.5. One sample t-test

Load the `bpwide` dataset into stata with the command `sysuse bpwide`. This consists of fiction blood pressure data, taken before and after an intervention. We wish to determine whether the intervention had affected the blood pressure.

7.23 Use the `summarize` command to calculate the mean blood pressure before and after the intervention. Has the blood pressure increased or decreased ?

. . . .

7.24 Generate a variable containing the change in blood pressure using the command `gen bp_diff = bp_after - bp_before`

7.25 Use the command `ttest bp_diff = 0` to test whether the change in blood pressure is statistically significant. Is it ?

. . . .

7.26 Give a 95% confidence interval for the change in blood pressure.

. . . .

4.7.6. Power Calculations

The following questions can all be answered using the `sampsi` command.

7.27 How many subjects would need to be recruited to have 90% power to detect a difference between unexposed and exposed subjects if the prevalence of the condition is 25% in the unexposed and 40% in the exposed, assuming equal numbers of exposed and unexposed subjects ?

. . . .

7.28 If the exposure was rare, so it was decided to recruit twice as many unexposed subjects as exposed subjects, how many subjects would need to be recruited ?

. . . .

7.29 Suppose it were only possible to recruit 100 subjects in each group. What power would the study then have ? ......

7.30 Suppose that we expect a variable to have a mean of 15 and an SD of 5 in group 1, and a mean of 17 and an SD of 6 in group 2. How large would two equal sized groups need to be to have 90% power to detect a difference between the groups ?

....

7.31 If we wanted 95% power, how large would the groups have to be ?

....

7.32 Suppose we could only recruit 100 subjects in group 1. How many subjects would we have to recruit from group 2 to have 90% power ?

....

Hint: the last question can only be answered by trying different numbers for the size of group 2 and seeing what power is achieved. Sensible choice of numbers will give a result fairly quickly. The `PageUp` key is your friend.

5. Introduction to Linear Models

5.1. What is a Linear Model ?

The aim of a statistical model is to predict an *outcome* variable based on one or more *predictor* variables. Outcome and predictor variables have many names: some are given table 5.1. A linear model is a particularly simple statistical model that assumes that the relationship between the outcome and the predictor(s) is a linear one, i.e. it can be described using straight lines.

Outcome	Predictor
Y-variable	x-variables
Dependent variable	Independent variables
Response variable	Regressors
Output variable	Input variables
	Explanatory variables
	Carriers
	Covariates

Table 5.1.: Names given to predictor and outcome variables

5.1.1. Linear Model Equation

Suppose that we have an outcome variable Y and p predictor variables from which we wish to predict Y, and that we have measured Y and x_1, x_2, $\ldots x_p$ on n subjects. In this case, the equation of the linear model is

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p + \varepsilon \tag{5.1}$$

The part of the equation $\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p$ is called the *Linear Predictor*. Its value is called the *predicted value of* Y and often written as $\hat{Y}$ (pronounced "Y hat"). This part of the model represents the variation in Y that can be predicted, and may be referred to as the *systematic* component of the model. The term ε is called the *error term* (also referred to as the *noise* or *random* component of the model). This represents the variation in Y that cannot be predicted. We assume that ε has a normal distribution, with mean 0 and variance σ^2_ε. The variance of Y is equal to the sum of the variance of $\hat{Y}$ plus σ^2_ε: in other words the total variability in Y is equal to the predictable variability plus the unpredictable variability. Clearly, we would like the predictable variability to be as great as possible and the unpredictable variability to be as little as possible.

5.1.2. Linear Model Assumptions

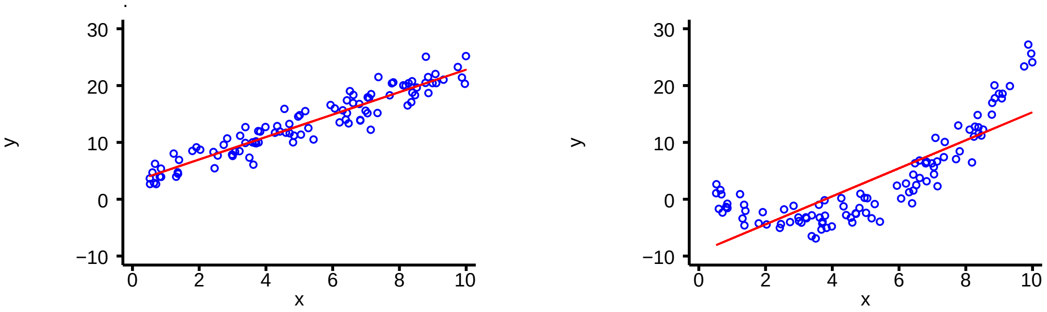
Implicit in the above equation are a number of assumptions about the data. They are

Variables Y_1, Y_2 ... Y_n are independent. Equation 5.1 assumes that only the x-variables are important for predicting Y. If Y depends also on other Y values, (for example in a time-series, when the value of Y may depend on the previous value of Y as well as the x-variables), the importance of the x variables can be over-estimated.

The variance of Y_i is constant. We assume that the error term has the same distribution irrespective of the values of x or Y.

Mean of Y is a linear function of x A linear model fits the best straight line to the data, as shown in Figure 5.1(a). If the true association between x and Y is not a straight line, the model will provide poor predictions. This can be seen in Figure 5.1(b), which illustrates a nonlinear association between x and Y, together with the fitted linear model. Due to the non-linearity, the predicted values are less than the observed values for extreme x-values and greater than the observed values for more central x values.

Distribution of Y_i is normal. This is particularly important in small samples, where outlying observations can have a large effect on the regression line.



(a) Linear Model

(b) Non-linear Model

Figure 5.1.: Comparison of Linear and Non-Linear models

Before doing any regression analysis, it is worth plotting Y against each of the x-variables to see if the above assumptions are reasonable. There are better ways to test the assumptions that we will consider later, but this first step will tell you if fitting a linear model is completely unreasonable.

5.1.3. Linear Model Parameters

The terms β_j in the linear predictor are called the *parameters* or *coefficients* of the regression model[a]. The meaning of the parameters is illustrated in figure 5.2. β_0 is the expected value of Y when all of the x-variables are 0 (the *intercept*). If x_j increases by 1, and all the other x-variables remain unchanged, then the expected value of Y increases by β_j. β_j may be referred to as the *slope* or *gradient* of Y on x_j.

[a] Avoid using the expression "beta-coefficient". Although commonly used to refer to regression coefficients, technically a beta coefficient is derived from data in which the standard deviation of all variables is 1

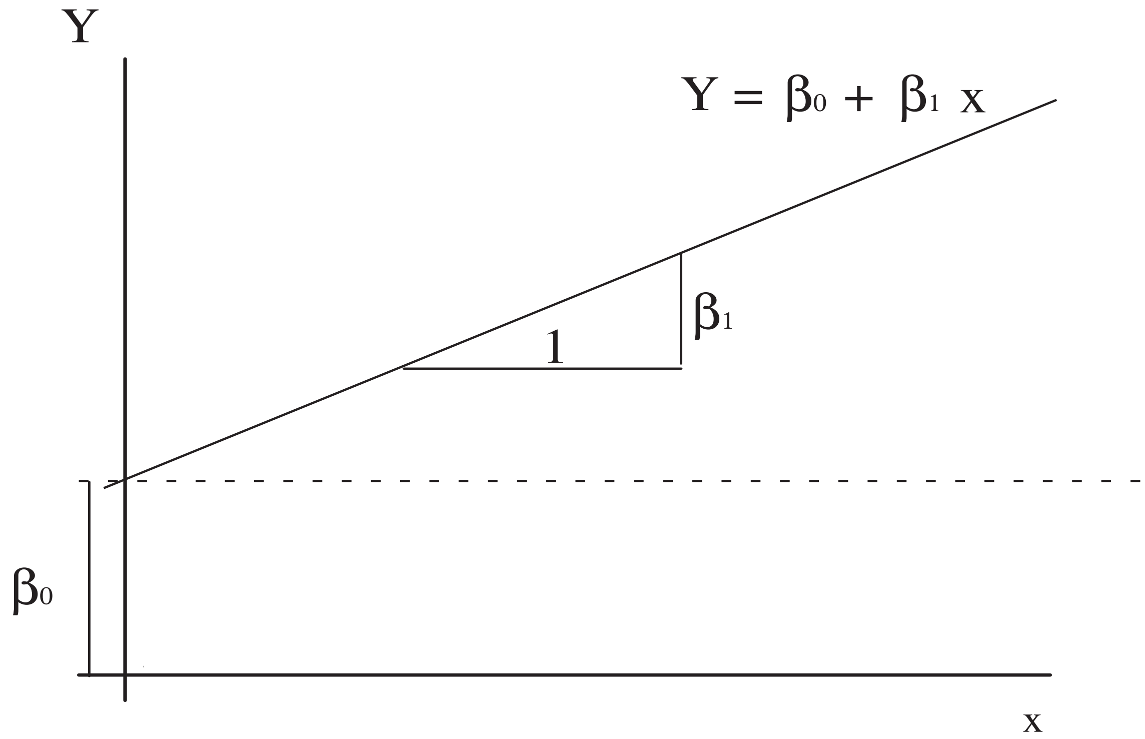


Figure 5.2.: Interpretation of Linear Model Parameters

Parameter Estimation

The β and σ parameters describe the relationship between Y and the x-variables in a population. However, we do not know these parameters, we can only estimate them based on a sample from the population. As is customary, we will use greek letters for population parameters and the equivalent roman letter for our estimate of the parameter: b_j is our estimate of the population parameter β_j and s is our estimate of the population parameter σ.

In general, the values given to b_0, b_1, etc. are those which minimise the sum of the squares of the residuals. These estimates are called the "Least Squares" estimates for this reason.

Inference on Parameters

If the assumptions of the linear model are correct, then the parameter estimates b_j will be normally distributed with mean β_j, and standard deviation

$$SD(b_j) = \sqrt{\frac{\sigma^2}{ns_x^2}}$$

where σ^2 is the variance of the error terms ε, s_x^2 is the variance of x_j and n is the number of observations (for sufficiently large n). Note that the standard deviation decreases as n increases (more observations give more precise estimates) and as s_x^2 increases (a wide range of x values gives a more precise estimate than a a narrow range).Since we know the distribution of b_j, we can perform hypothesis tests and construct confidence intervals for β_j.

Unfortunately, we do not know σ, we only have an estimate of it, s_ε, from the data. The slope estimates b_j will therefore follow a t-distribution with $n - p - 1$ degrees of freedom, with mean β_j and standard deviation

$$s_{b_j} = \sqrt{\frac{s_\varepsilon^2}{ns_x^2}}$$

This means that we can test hypotheses about β_j using t-statistics. To test the hypothesis that $\beta_j = B$, we form

$$t = \frac{(b_j - B)}{s_{b_j}}$$

This can be compared to a t-distribution on $n - p - 1$ degrees of freedom, to provide a p-value for the hypothesis. Most commonly, we wish to test the hypothesis that $\beta_j = 0$, since that would mean that Y is not associated with x_j.

We can also create a $(1 - \alpha)$ confidence interval for β_j as

$$b_j \pm t_{\alpha/2,n-p-1} \times s_{b_j}$$

Again, if 0 lies within this confidence interval, we would conclude that x_j and Y are not associated.

Note that if n is sufficiently large, the t-distribution is well approximated by a normal distribution. In this case, a 95% confidence interval can be found by

$$b_j \pm 1.96 \times s_{b_j}$$

However, for small n, the differences between the normal and t-distributions can be considerable, and the above confidence interval will be too narrow. It is therefore better to let stata calculate the confidence intervals for you whenever possible.

The Intercept, β_0 In general, inferences about the intercept, β_0 are of less interest. This is because it is the value taken when all of the x-variables are 0, and usually there is at least one x-variable for which a value of 0 is not sensible (for example height or weight). However, if required, confidence intervals and hypothesis tests can be produced in exactly the same manner as for the slope parameters β_j.

The Outcome Variable, Y The main purpose of a linear model is to be able to predict a Y-variable from one or more x-variables. Our “best estimate” of Y is given by the linear predictor, but we are also interested in how much individual values of Y will vary around that estimate. There are two sources of variation in Y:

1. The random component of the linear model.
2. Uncertainty about the parameters β_j of the linear model

Clearly, if there is considerable random variation in Y, it will not be possible to predict it with great accuracy. However, it is also clear that the more precisely we have estimated the relationship between x and Y, the more precisely we will be able to predict Y.

Thus the *prediction interval*[b] for Y has the form

$$\hat{Y} \pm t_{\alpha/2,n-p-1} \times \sqrt{s^2_{\hat{Y}} + s^2_{\varepsilon}}$$

The first term under the square root sign represents the uncertainty in the value of $\hat{Y}$, due to the uncertainty about the parameter estimates, the second term represents the random component of the linear model. The second term is assumed to be constant for all values of x, but the first term depends on x: we can estimate Y better in the centre of our data than in more outlying regions. This first term also depends on n: the larger the sample we use, the more precisely we can estimate the parameters of the linear model.

The Fitted Value, $\hat{Y}$ We may be interested in the *mean* value of Y at a given value of x in a population, rather than individual values. In this case, we need not worry about the random component of the linear model, since the mean value of the error terms is 0 by definition. Therefore, a confidence interval for $\hat{Y}$ is given by

$$\hat{Y} \pm t_{\alpha/2,n-p-1} \times \sqrt{s^2_{\hat{Y}}}$$

5.1.4. Analysis of Variance

We saw above that the total variation in Y is made up of the predictable variation and unpredictable variation. This can be formalised mathematically as

$$\sum \left(Y - \bar{Y}\right)^2 = \sum \left(\hat{Y} - \bar{Y}\right)^2 + \sum \left(Y - \hat{Y}\right)^2 \tag{5.2}$$

In equation 5.2, the first term is called the *total sum of squares*, since it represents all of the variation in Y about its mean value ($\bar{Y}$). The second term is called *the regression sum of squares*, since it represents the variation of the *predicted values* ($\hat{Y}$) about the mean, i.e. that part of the variation that is predictable. The third term is called the *residual sum of squares*, and represents the variation of the observed Y values about their predicted values. This represents the unpredictable or random variation in Y.

Each sum of squares has an associated *degrees* of freedom (d.f.). The d.f. for the total sum of squares is $n-1$, since the variance of Y is $\sum \left(Y - \bar{Y}\right)^2/(n-1)$. The d.f. for the regression sum of squares is the number of parameters in the regression model. The residual degrees of freedom is found by subtracting the regression d.f. from the total d.f. This enables us to draw up the following table, called an *Analysis of Variance* or ANOVA table:

The ratio MS_{reg}/MS_{res} is a measure of how much more of the variation in Y is explained by the x-variables than would be expected by chance. If there is no association between Y and the x-variables, this we would expect this ratio to be equal to 1. If the ratio is greater than 1, this suggests that there is an association between Y and the x-variables. MS_{reg}/MS_{res} can be compared to an F distribution to test the hypothesis that there is no association.

If the linear regression model contains only a single variable, then the p-value resulting from the hypothesis test that $\beta = 0$ will be *exactly* the same as that resulting from the hypothesis test that $F = 1$. If there are several predictors, the F-test provides a test of the overall model, whilst the t-tests provide tests of each individual predictor variable.

[b] This is *not* a confidence interval, because only parameters have confidence intervals, and Y is a random variable, not a parameter. In fact, it is a reference range for Y, conditional upon x.

Source	df	Sum of Squares	Mean Square	F
Regression	p	SS_{reg}	$MS_{reg} = \frac{SS_{reg}}{p}$	$\frac{MS_{reg}}{MS_{res}}$
Residual	n-p-1	SS_{res}	$MS_{res} = \frac{SS_{res}}{(n-p-1)}$	
Total	n-1	SS_{tot}	$MS_{tot} = \frac{SS_{tot}}{(n-1)}$	

Table 5.2.: ANOVA Table

5.1.5. Goodness of Fit

The fact that there is an association between x and Y does not necessarily imply that x is useful for predicting Y. The statistical significance of an effect depends both on the size of the effect and the size of the sample in which it is being measured. A small effect may be highly significant in a very large sample, but still provide little predictive power.

The predictive power of the linear model depends on how much of total variation in Y can be predicted. We have seen that the predictable variation is SS_{reg} and the total variation is SS_{tot}, so the proportion of the variation that we can predict is

$$R^2 = \frac{SS_{reg}}{SS_{tot}}$$

This quantity is called R^2 because it is the square of the coefficient of correlation between Y and $\hat{Y}$: since it is a proportion it can take any value from 0 to 1.

This gives an overall measure of how good a model is, but it is difficult to use to compare models. This is because adding an extra variable to a model will *always* increase R^2, whether or not the variable is related to Y. A better statistic for comparing models is the adjusted R^2, which allows for the fact that even unrelated variables will explain some of the variance of Y. The larger the sample size, the smaller the difference between R^2 and adjusted R^2.

5.1.6. Linear Models in Stata

The `regress` Command

Linear models are fitted in stata using the `regress` command. The syntax for this command is simply

```
regress yvar xvars
```

The regression can be performed in different groups using `by` or `if` clauses, and there are various other complex options which you need not worry about yet.

Fitted values from a regression model can be obtained by using the `predict` command. Typing

```
predict varname, xb
```

will create a new variable called *varname*, containing the fitted values[c].

The `predict` command can also generate other variables which may be of interest. For example, the standard error of the forecast, $\sqrt{s^2_{\hat{Y}} + s^2_{\varepsilon}}$, can be calculated by

```
predict varname, stdf
```

The standard error of $\hat{Y}$, called the standard error of the prediction in stata can be calculated by

```
predict varname, stdp
```

These enable us to construct confidence intervals and prediction intervals.

Understanding Stata Output

In this section, we are going to use stata to fit a linear model to the data in table 5.3.

x	Y
4	4.26
5	5.68
6	7.24
7	4.82
8	6.95
9	8.81
10	8.04
11	8.33
12	10.84
13	7.58
14	9.96

Table 5.3.: Data for Regression

If this data is entered into stata, and we type

```
scatter Y x
```

we get the scatterplot shown in figure 5.3. This scatterplot suggests that Y increases as x increases, and

[c] The odd name "xb" is derived from matrix algebra, since $b_1x_1 + b_2x_2 + \ldots + b_px_p$ can be written as $\mathbf{xb}$, where $\mathbf{x}$ is the $1 \times k$ matrix $x_1, x_2, \ldots, x_p$ and $\mathbf{b}$ is the $k \times 1$ matrix $b_1, b_2, \ldots, b_p$.

fitting a straight line to the data would be a reasonable thing to do. So we can do that by typing

```
regress Y x
```

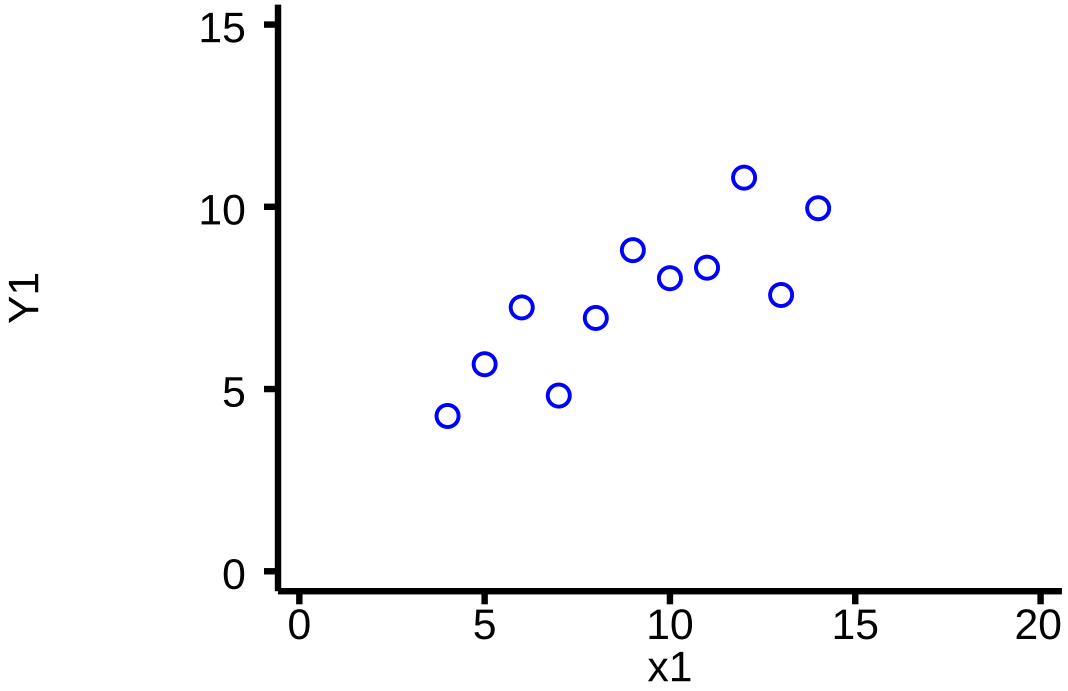


Figure 5.3.: Scatterplot of sample linear model data

The output from the `regress` command comes in two parts: first an ANOVA table and some overall tests of the model, then a table of statistics for the individual coefficients. Here is the first part of the output for the above regression.

```
  Source |       SS       df       MS              Number of obs =      11
---------+------------------------------           F(  1,     9) =   17.99
   Model |  27.5100011     1  27.5100011           Prob > F      =  0.0022
Residual |  13.7626904     9  1.52918783           R-squared     =  0.6665
---------+------------------------------           Adj R-squared =  0.6295
   Total |  41.2726916    10  4.12726916           Root MSE      =  1.2366
```

Most of the ANOVA table is on the left, with other statistics on the right. The F value is 17.99, which is considerably greater than 1. The p-value, labelled `Prob > F`, is 0.0022: in other words, if there were no true association between x and Y, only 2 times out of 1000 would random sampling produce such a strong observed association. We can therefore reject the hypothesis of no association. R^2 is 0.6665, so nearly 67% of the variation in Y can be predicted from x, and only 33% is random. Finally, the term `Root MSE` is the square root of the residual mean square MS_{res}, which is an estimator for σ.

Here is the second part of the output which deals with the individual coefficients:

```
------------------------------------------------------------------------------
       Y |      Coef.   Std. Err.       t     P>|t|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
       x |   .5000909   .1179055      4.241   0.002       .2333701    .7668117
   _cons |   3.000091   1.124747      2.667   0.026       .4557369    5.544445
------------------------------------------------------------------------------
```

The intercept parameter, β_0, is labelled _cons (for "constant") by stata. Thus our prediction equation is

$$\hat{Y} = 3.00 + 0.500 \times x$$

The table also gives standard errors for the parameters β_1 and β_0, and the corresponding t-statistics to test the hypothesis $\beta = 0$. The p-values are given in the 5^{th} column, showing that in this case both parameters are significantly different from 0[d]. Finally, a 95% confidence interval for each parameter is given.

We may now wish to look at the confidence intervals for the fitted values and the prediction intervals. To see the data along with a 95% prediction interval, we could now use the command[e]

```
twoway lfitci Y1 x1, stdf|| scatter Y1 x1, ylab(0(5)15) xlab(0(5)20)
```

Figure 5.4(a) shows the data along with a 95% prediction interval. We would expect 95% of the observations to lie within this interval: in this case, all 11 observations do. Figure 5.4(b), on the other hand shows the regression line, together with its 95% confidence interval. This figure was obtained by typing

```
twoway lfitci Y1 x1|| scatter Y1 x1, ylab(0(5)15) xlab(0(5)20)
```

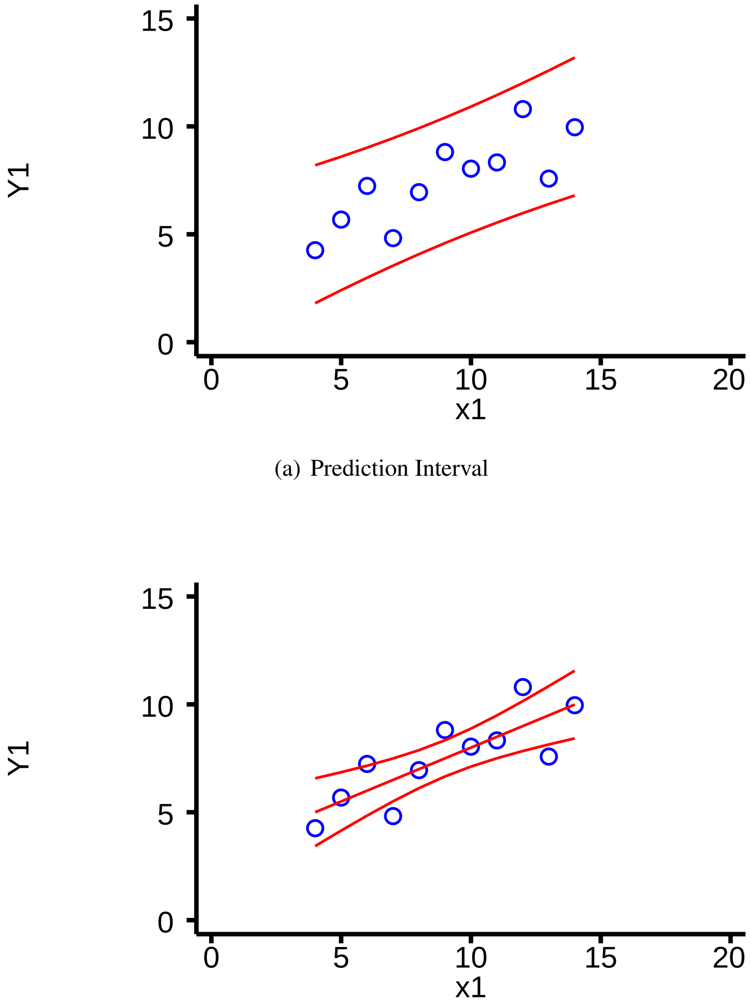
Notice that the confidence interval is much narrower than the prediction interval, since it does not need to incorporate the random element in Y. The confidence interval around the fitted value is analogous to the confidence interval around the mean of sample, whilst the prediction interval is analogous to a reference range for values in the population. Also, both the confidence interval and reference range are narrower near the centre of the range of x-values and wider further away. In particular, the confidence interval and prediction interval are very wide when x = 0, since this is well to the left of the data. In fact, the confidence interval when x = 0 is (0.46, 5.54), since this is the confidence interval of β_0.

5.2. Diagnostics

How do we know if the model is adequate ? Part of that question is answered by the R^2 statistic: if R^2 is small, the model is not good at predicting Y. However, there is another aspect to this question: do the data satisfy the assumptions of the linear model outlined in section 5.1.2. If not, any conclusions drawn may be misleading.

[d] The more observant among you may have noticed that the p-value from the t-test for x is the same as that from the F-test above. This is not a coincidence

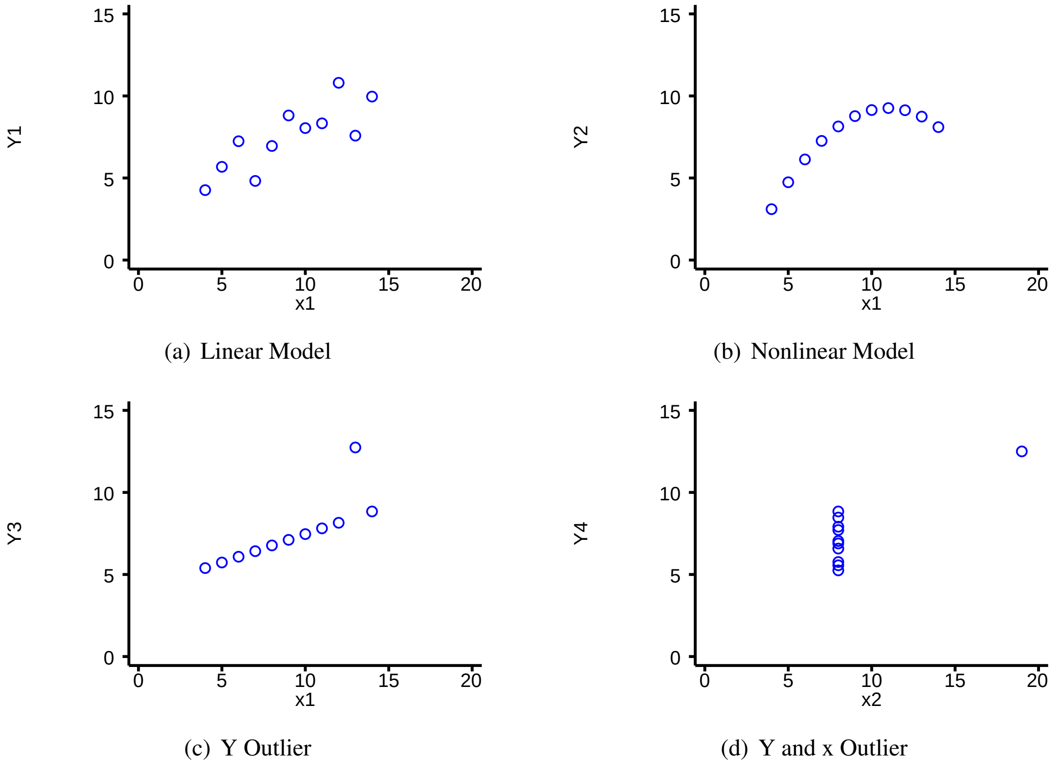
[e] (The graphics commands will be covered in session 11: for now just take my word for it that this does what it should).



(a) Prediction Interval

(b) Confidence Interval

Figure 5.4.: Prediction Interval and Confidence Interval in Linear Regression



(a) Linear Model

(b) Nonlinear Model

(c) Y Outlier

(d) Y and x Outlier

Figure 5.5.: Different Data Configurations Resulting in the Same Linear Model

Consider the scatter-plots in Figure 5.5[f]. In each case, the ANOVA table is the same as that of section 5.1.6, as are the parameter estimates and standard errors. However, only in figure (a) is a linear model an appropriate fit to the data. Using a linear model to predict values of Y from values of x in the other situations may result in very poor estimates. In the following sections, we will see why a linear model is inappropriate for the other datasets, and how we can recognise similar situations.

5.2.1. Testing Assumptions

When testing that your data satisfies the linear model assumptions, it is important to proceed in the following order:

Confirm Constant Variance If the the variance of Y varies as x varies, a linear model cannot be fitted. There are two possible solutions: either transform Y or use weighted regression, both of which are beyond the scope of this introduction.

Confirm Linearity of Association If the association between Y and x is not linear, this can be solved by either transforming one or more x variables or fitting polynomials x^2, x^3 etc.

Identify Influential Observations If one observation is unusual, it can have an inordinate influence on the regression line, as shown by Figures 5.5(c) and 5.5(d). Such points need to be identified and their influence assessed.

Confirm Normality of ε_i Finally, we should check that the residuals are normally distributed.

5.2.2. Confirm Constant Variance

The first assumption that must be tested is the constant variance of error terms. This is because if this assumption does not hold, the solution may be to transform Y. If this is necessary, it will change all of the properties of the error terms, so that if they satisfied the assumptions of the linear model before the transformation, they are unlikely to satisfy them afterwards.

To test that the variance of the error terms, $\varepsilon_i = Y_i - \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \ldots + \beta_p x_{pi}$, is constant, we need to calculate the ε_i. We cannot calculate them directly, because we do not know the values of the β parameters, we only have estimates for them. The obvious estimate for the error term ε_i is $e_i = Y_i - b_0 + b_1 x_{1i} + b_2 x_{2i} + \ldots + b_p x_{pi}$. The e_i are referred to as *residuals*.

However, these residuals are not suitable for testing the assumption of constant variance, since even if the ε_i have constant variance the e_i do not. This is because $\hat{Y} = b_0 + b_1 x_{1i} + b_2 x_{2i} + \ldots + b_p x_{pi}$ varies more away from the centre of the data, but Y has constant variance. Therefore, since

$$var(Y_i) = var(\hat{Y_i}) + var(e_i)$$

the variance of the residuals must be *less* at the extremes of the data.

[f] These data were devised by F. J. Anscombe, and first explored in "Graphs in Statistical Analysis", *The American Statistician*, vol 27, pages 17-21.

Since we can calculate the expected variance of e_i, we can adjust for it and produce residuals which do have constant variance provided that the ε_i have constant variance. These residuals are called *standardised* residuals (s_i)[g]. I will spare you the mathematical formula: the way to calculate them is to type

```
predict varname, rstandard
```

after a `regress` command.

A plot of s_i against $\hat{Y_i}$ should show no pattern. The plot should consist of a rectangular cloud of points, showing that the mean standardised residual is 0 for all values of x, and that the variance of the standardised residuals are the same for all values of x. A typical result of such a plot, is shown in Figure 5.6(a)

There are two patterns that are commonly seen in plots of s_i against $\hat{Y_i}$ which suggest that a linear model is inappropriate. Firstly, the spread of the data may increase (or more rarely decrease) as $\hat{Y_i}$ increases. This is called *heteroskedasticity*. It may be possible to remove this by transforming Y. Alternatively, a weighted regression can be used, but this is beyond the scope of this work. This is illustrated in Figure 5.6(b).

Alternatively, the plot may show curvature. This is an indication that the association between Y and at least one of the x-variables is not linear. However, it does not indicate which x-variable is the problem: a different kind of plot (a partial residual plot, described in section 5.2.3) is required for that. Curvature in the residual vs fitted value plot is shown in Figure 5.6(c).

Stata can produce standardised residuals for you, using the command

```
predict varname, rstandard
```

These can then be plotted against the fitted values produced by

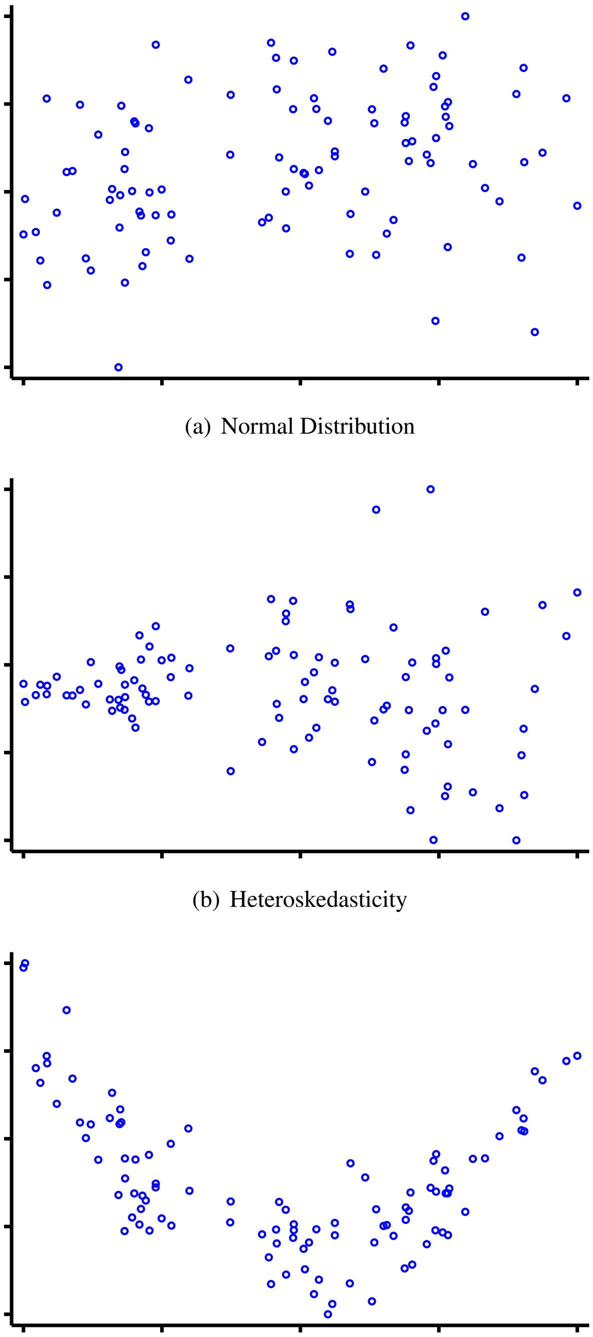
```
predict varname, xb
```

There is a builtin command `rvfplot`, which plots residuals against fitted values, but since it does not use studentised residuals it may be misleading in small datasets. In this case, it is better to to use `predict` to calculate studentised residuals and create the plot yourself. For large datasets, the difference between the residual and the standardised residual is small, and `rvfplot` can safely be used.

If you are unsure whether the plot reveals non-constant variance, there is a command `hettest` to formally test this. Simply typing `hettest` tests whether the variance is a function of $\hat{Y}$, whilst `hettest varname` tests if the variance is a function of the predictor variable *varname*. It is also possible to formally test whether there is curvature in the plot, using the command `ovtest`. However, if there are several x-variables, it is better to consider partial residual plots, outlined in section 5.2.3

Interpreting the results of the formal tests requires care. If the dataset is very large, it is possible for a

[g] Some statisticians call these residuals *studentised* residuals, but others, including those who wrote stata, use the term studentised residuals for something completely different, which we will come to soon



(a) Normal Distribution

(b) Heteroskedasticity

(c) Non-Linearity

Figure 5.6.: Plots of Residuals vs Fitted Values

very small, unimportant effect to be statistically significant. On the other hand, in a small data set a large effect, which does invalidate the model, may not be statistically significant. Therefore, the plots should always be considered, as well as the results of the formal tests.

5.2.3. Confirm Linearity of Association

One way to confirm the linearity of the association would be to plot the residuals against each of the predictor variables in turn. However, if there are several predictor variables, it can be more useful to generate the partial residuals

$$p_j = e + b_j x_j = Y - \beta_0 - \sum_{l \neq j} b_l x_l$$

These partial residuals are formed by subtracting from the observed value of Y that part of the predicted value that does not depend on x_j. Thus, a plot of p_j against x_j will reveal the association between Y and x_j after adjusting for the other predictors.

It has been suggested that it is easier to interpret a partial residual plot if $b_j x_{ij}$, is plotted along with it. The term $b_j x_{ij}$ is called the *component*, since it is the component of Y that can be predicted from x_j. This shows the presumed linear association, to make departures from linearity easier to spot. Such plots are known as *component-plus-residual* plots (CPR plots) or *component and component-plus-residual* plots , since the partial residual, p_j is the sum of the component of Y that is due to x ($b_j x_j$) and the residual e.

These plots can be obtained from stata using the command `cprplot` *`varname`*, after having run a regression. They are unnecessary if there is only a single predictor variable (a plot of Y against x provides the same information in this case), but can be very useful if there are a number of correlated predictor variables.

Figure 5.7 shows a CPR plot for the data in figure 5.5(b). It clearly illustrates the curvature in the relationship between Y and x.

5.2.4. Identify Influential Observations

We have seen from figures 5.5(c) and 5.5(d) that a single unusual observation can have a marked effect on the regression equation. Such points are called *influential* points. Identifying influential points can be surprisingly difficult, particularly if there are a number of them and several predictor variables.

The influence a point has on the regression model depends on two factors: how unusual the values of its x-variables are and how unusual the value of its y-variable is. For example, in Figure 5.5(d), the single point with an x value different from the other points completely determines the regression equation. Points with unusual x values are said to have high *leverage*, and are potentially influential points. If, in a addition, the Y value is also unusual, the point will be influential.

Identifying Influential Observations Graphically

One way to detect influential points is to produce a plot of the leverage against the squared residual, since the squared residual measures how unusual the Y value is. This can be produced simply in stata by

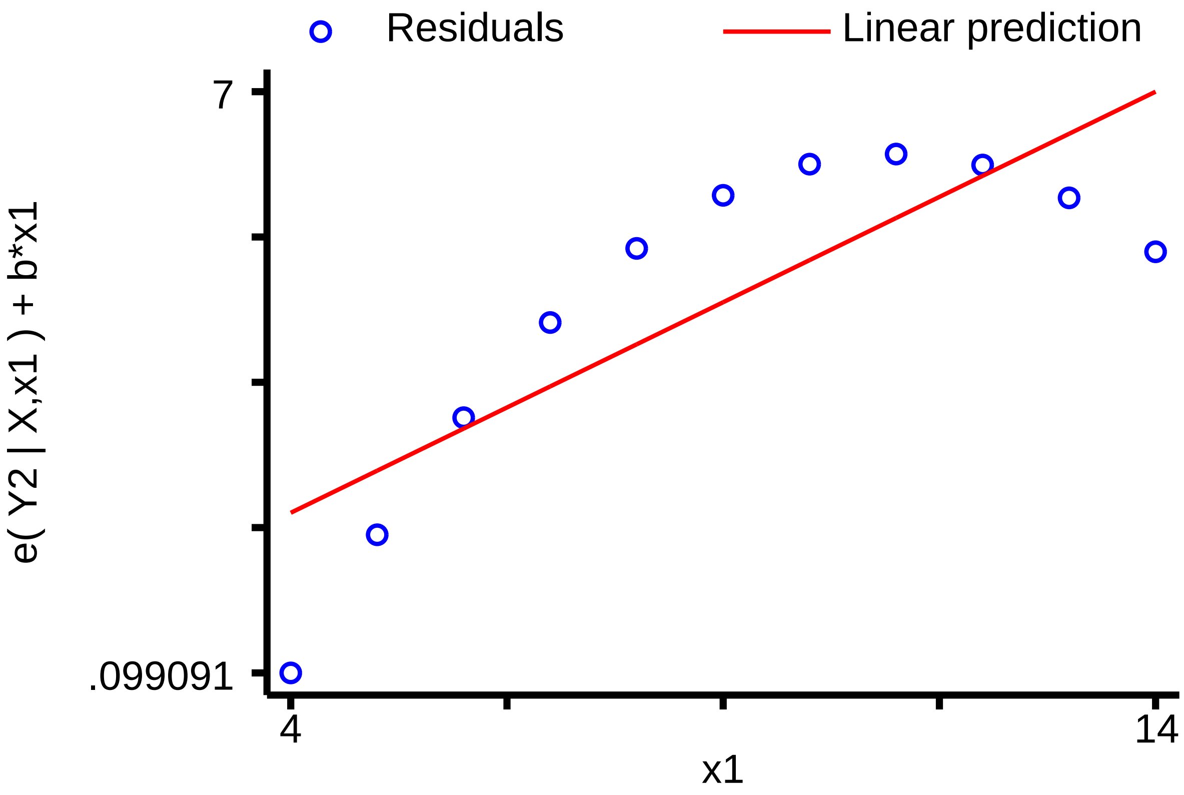


Figure 5.7.: Component and component-plus-residual plot for data from Figure 5.5(b)

typing `lvr2plot`. Points near the top of this diagram have high leverage, and are therefore potentially influential. Points to the right of this plot have large residuals. However, influential points, by definition, pull the regression line towards themselves and hence reduce their residuals. For example, in Figure 5.5(d) the point on the right is highly influential because of its x value, but has a residual of 0, precisely because it is so influential. It may be best to re-run the regression excluding any points with high leverage, to see if the conclusions change. Alternatively, there are some statistics that can be used to assess influence in section 5.2.4.

To detect influential points, *studentised* residuals are often used. These are similar to standardised residuals, the regression line from which the residual of a particular point is measured is fitted *excluding that point*. This is because an influential point will attract the regression line towards it, and may therefore have a small standardised residual, but since it is far from the pattern of the rest of the data will have a large studentised residual.

Identifying Influential Observations Formally

There are a number of statistics that can be calculated to determine how influential a point is, depending on what the point may have an influence on. For example, if you are concerned that an observation is having an undue influence on the regression coefficients, you may use the DFBETA statistic. DFBETA$_{ij}$ is a measure of how much β_j is changed if observation i is not included in the regression.

However, in general we are more interested in the predicted values. If there are a number of correlated predictors, an observation may have considerable influence over several coefficients without influencing the predicted value greatly (if the changes "cancel out"). In this case, it may be better to consider DFFITS$_i$, which gives a measure of the change in the predicted value of observation i if observation i is not used to calculate the regression equation. Another statistic related to DFFITS is Cook's distance. This can be thought of as a measure of the change in *all* the fitted values if an individual observation is omitted.

There is no theoretical basis for deciding when DFBETAS, DFFITS, or Cook's distance are large, since the expected distributions when the data are normally distributed are not known. However, some commonly used cut-offs are given in table 5.4. These should only be taken as suggestions, however, since different authors would choose different cut-offs. An alternative idea is to plot Cook's distance against the predicted value, and look for outlying observations.

Statistic	Cutoff
DFBETAS	$2/\sqrt{n}, 1.5/\sqrt{n}$
DFFITS	$2\sqrt{(p+1)/n}$
Cook's Distance	4/n

Table 5.4.: Commonly Used Cutoffs for Influence Statistics

If the dataset does contain one or more influential observations, what should be done ? Firstly, you should check that the unusual observations are not due to an error. If no error can be found, the sensible thing to do is to present the results of fitting the regression both to all points and to the non-influential points, to illustrate the effect of the influential points.

It should also be pointed out that although these statistics are good for detecting individual outliers, if there are several outliers they may not work. It is therefore essential to look at plots of the data to see if there are outliers.

All of the statistics mentioned above can be calculated directly by stata, using the `predict` command after a regression. For details, see the stata manual or type `help regress` into stata.

Y-outliers

Sometimes, there are points which do not have particularly high leverage, but which are influential, due to the observed Y value being very different from the expected value. An example of this is in Figure 5.5(c), where the single outlier pulls the regression line away from the other points, despite not having particularly high leverage.

In this case, there is an alternative to simply deleting the outlying point: robust regression. This involves repeated fitting a regression model, with the weighting of each observation being determined the magnitude of its residual from the previous regression model: the larger the residual, the smaller the weighting (deleting outliers corresponds to giving them a weight of 0, and all the other points a weight of 1). It thus provides a better fit to the bulk of the data, but does not completely ignore outliers.

Figure 5.8 illustrates the use of robust regression on the data in Figure 5.5(c). In this case, the robust regression line passes through 10 of the 11 points in the dataset, and is unaffected by the outlier.

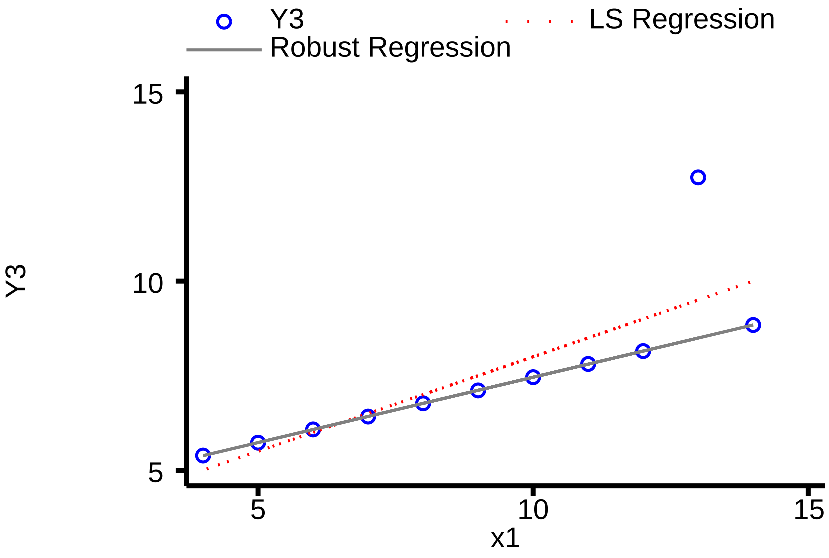


Figure 5.8.: Comparison of Least Squares and Robust Regression

5.2.5. *Confirm Normality of Residuals*

Again we use the standardised residuals. A plot of the quantiles of the standardised residuals against the quantiles of a normal distribution should give a straight line: and deviation from a straight line suggests that the residuals are not normally distributed. Such a plot can easily be achieved in stata with the command `qnorm` *`varname`*, where *`varname`* is the name given to the standardised residuals. A formal test of the normality of the residuals is provided by `swilk` *`varname`*.

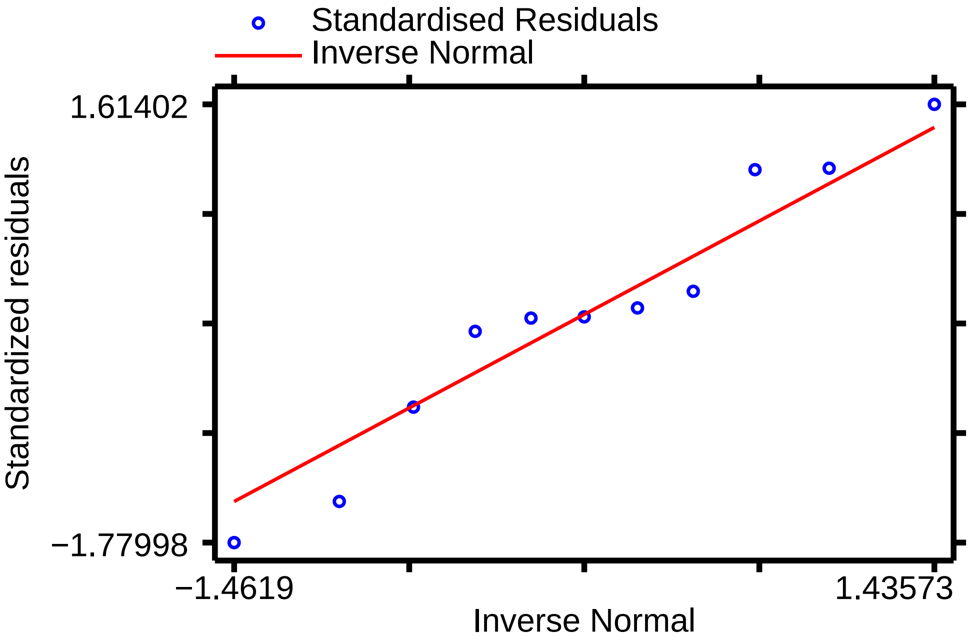
Figure 5.9 below shows normal plots for the residuals in Figures 5.5(a) and 5.5(c). The residuals from the linear model are reasonably normally distributed, but the outlier in the second dataset is clearly visible.

5.2.6. Dangers of Extrapolation

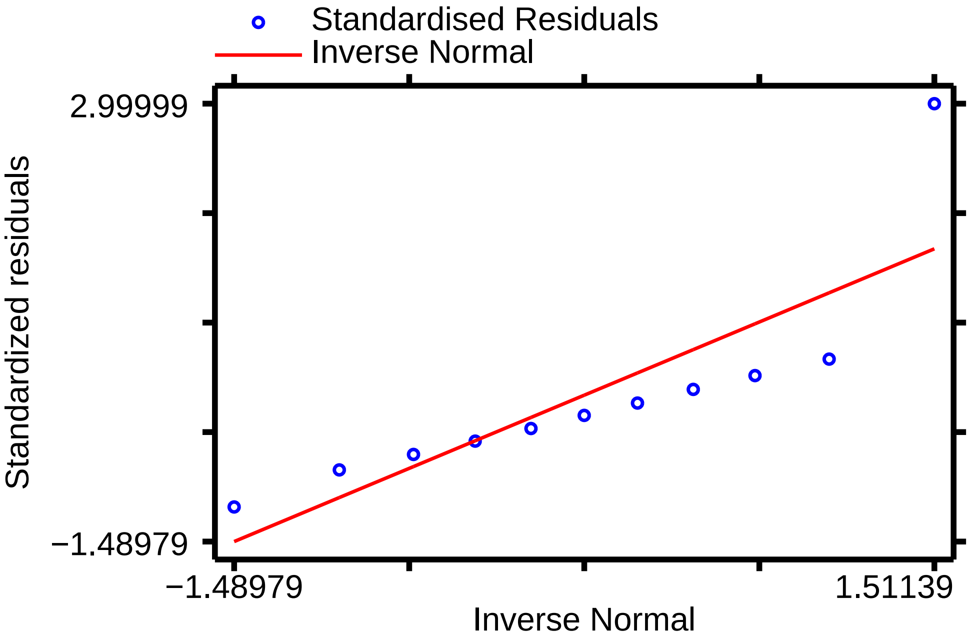
We have seen that our predictions become less reliable as we move away from the centre of our data. In addition, when we test the assumptions of the linear model, we can only test them within the range of x-values that we have observed. For example, Y may be a linear function of x within the range of x-values measured, but not for more extreme values. For these reasons it is a mistake to predict Y from values of x which lie outside the range of observed x-values.

5.3. References and Further Reading

Further Reading



(a) Linear model



(b) Y and x Outlier

Figure 5.9.: Normal plots for the residuals from data in Figures 5.5(a) and 5.5(c).

5.4. Linear Models Practical

5.4.1. Datasets

All but one of the datasets that you will use in this practical can be accessed via http from within stata. However, the directory in which they are residing has a very long name, so you can save yourself some typing if you create a global macro for this directory. You can do this by entering

```
global basedir http://personalpages.manchester.ac.uk/staff/mark.lunt
global datadir $basedir/stats/5_LinearModels1/data
```

(That could be entered as a single line, but fitting it on the page would have been tricky). If you wish to run the practical on a computer without internet access, you would need to:

1. Obtain copies of the necessary datasets
2. Place them in a directory on your computer
3. Define the global macro `$datadir` to point to this directory.

The only dataset not stored in `$datadir` is the `auto` dataset, which is distributed with stata. This can be loaded with the command

```
sysuse auto, clear
```

5.4.2. Fitting and Interpreting a Linear Model

The Anscombe Data

Load the data illustrated in figure 5.5 by entering the command

```
use "$datadir/anscombe.dta"
```

The scatter-plots in figure 5.5 can then be reproduced by entering

```
scatter Y1 x1, xlab(0 (5) 20) ylab(0 (5) 15)
scatter Y2 x1, xlab(0 (5) 20) ylab(0 (5) 15)
scatter Y3 x1, xlab(0 (5) 20) ylab(0 (5) 15)
scatter Y4 x2, xlab(0 (5) 20) ylab(0 (5) 15)
```

The `xlab` and `ylab` options are to ensure that the same axes are used for all graphs. The x-axes go from 0 to 20 with tick-marks every 5, whilst the Y-axes go from 0 to 15 with tick-marks every 5. Note that `Y1`, `Y2` and `Y3` are all plotted against `x1`, but `Y4` is plotted against `x2`.

Further Reading

You should satisfy yourself that these datasets really do produce the same linear models by entering

```
regress Y1 x1
regress Y2 x1
regress Y3 x1
regress Y4 x2
```

The Automobile Data

Load the `auto` dataset which is distributed with stata (the command `sysuse auto, clear` will achieve this). Fit a linear model to predict fuel consumption (`mpg`, miles per gallon) from the weight of the car (`weight`, measured in lbs), using the command `regress mpg weight`

4.1 Is fuel consumption associated with weight ?

4.2 What proportion of the variance in `mpg` can be explained by variations in weight ?

4.3 What change in `mpg` would be expected for a one pound increase in weight ?

4.4 What fuel consumption would you expect, based on this data, for a car weighing 3000 lbs ? (Hint: the command `lincom _cons + 3000*weight` will give you the answer, along with a 95% confidence interval around the expected value. I'll explain how it works next week)

4.5 Would it be reasonable to use this regression equation to calculate the expected fuel consumption of a car weighing 1000 lbs ?

5.4.3. Diagnostics

Constancy of Variance

Use the data `$datadir/constvar`, which is simulated data generated for this practical.

4.6 Perform a regression of `y` on `x`, using the command `regress y x`. Is there a statistically significant association between `y` and `x` ?

Create standardised residuals using the command `predict rstand, rstand`. Create predicted values using the command `predict yhat`. Now produce a graph of standardised residuals against predicted values with the command `scatter rstand yhat`.

4.7 Would you conclude that the variance is constant for all values of `yhat`, or is there any evidence of a pattern in the residuals ?

4.8 Confirm (or disprove) your answer to the previous question by using the command `hettest`.

4.9 Produce a residual vs fitted value plot with the command `rvfplot`. Would this plot give the same conclusion that you reached in the previous question ?

Use the command `gen ly = ln(y)` to generate a new variable equal to the log of `y`. Perform a regression of `ly` on `x` with the command `regress ly x`. Generate new standardised residuals and predicted values with the commands `predict rstand2, rstand` and `predict yhat2`. Produce a plot of the standardised residuals against the fitted values with `scatter rstand2 yhat2`.

4.10 Is the variance of the residuals constant following this transformation ?

4.11 Confirm your answer to the previous question with the command `hettest`.

Confirming Linearity

Use the data `$datadir/wood73`. This is simulated data to illustrate the use of the CPR plot.

4.12 Plot graphs of `Y` against `x1` and `Y` against `x2` with the commands `scatter Y x1` and `scatter Y x2`. Do these graphs suggest a nonlinear association between `Y` and either `x1` or `x2` ?

4.13 Perform a regression of `Y` on `x1` and `x2` with the command `regress Y x1 x2`.

4.14 Produce CPR plots for `x1` and `x2` with the commands `cprplot x1` and `cprplot x2`. Do either of these plots suggest a non-linear relationship between `Y` and either `x1` or `x2` ?

4.15 Generate a new variable, `x3`, equal to the square of `x1`, with the command `gen x3 = x1^2`. Include `x3` in the regression equation with the command `regress Y x1 x2 x3`. Is `x3` a statistically significant predictor of `Y`l ?

4.16 Produce a CPR plot for `x1`, `x2` and `x3`. Is there still evidence of non-linearity ?

4.17 Use the command `predict Yhat, xb` to generate predicted values for `Y`. Plot `Y` against `Yhat` with the command `scatter Y Yhat`. How good are `x1, x2` and `x3` at predicting `Y` ? Is this what you expected from the value of R^2 from the regression ?

Outlier Detection

Use the data `$datadir/lifeline`. This data was collected to test the hypothesis that the age to which a person will live is governed by the length of the crease across the palm known as the "lifeline" in palmistry. The age at which each subject died is given by `age`, and the length of their lifeline (normalised for body size) is given by `lifeline`.

4.18 Perform a regression of `age` on `lifeline`, using the command `regress age lifeline`. Is there a significant association between age at death and the length of the lifeline in this dataset ?

4.19 Produce a plot of `age` on `lifeline`, using the command `scatter age lifeline`. Are there any points that lie away from the bulk of the data ? If there are, are they outliers in `age`, `lifeline` or both ?

4.20 Are there any points in the above graph that you would expect to have undue influence on the regression equation ?

4.21 Calculate Cook's distance for each observation with the command `predict cooksd, cooksd`. Calculate the predicted age at death for each observation with the command `predict predage`. Plot Cook's distance against predicted age at death with `scatter cooksd predage`. Do any observations have an unusually large Cook's distance ?

4.22 Use `summarize cooksd, det` to identify the value of the largest Cook's distance. Rerun the regression excluding the point with the largest Cook's distance using the command `regress age lifeline if cooksd <` *x*, for some value of *x* of your choice. How does removing this point affect the regression ?

4.23 Repeat the above analysis removing the two most influential points. Does this change your conclusions about the association between age at death and length of lifeline in this dataset ?

4.24 What is your conclusion about the association between age at death and length of lifeline in this dataset ?

Confirming Normality

We will continue to use the data in `$datadir/lifeline`. Redo the regression including all observations with the command `regress age lifeline`, then use the command `predict rstand, rstand` to produce standardised residuals.

4.25 Draw a normal plot of the standardised residuals with the command `qnorm rstand`. Do the plotted points lie on a straight line ? Are there any observations that do not appear to fit with the rest of the the data ?

4.26 Confirm your answer to the previous question by formally testing for normality of the residuals with the command `swilk rstand`. Do the residuals follow a normal distribution ?

5.4.4. Complete Example

This example uses `hsng.dta`, a dataset consisting of data on housing in each state in the USA taken from the 1980 census. The variables we are particularly interested in are `rent`, the median monthly rent in dollars; `faminc`, the median annual family income in dollars; `hsng`, the number of housing units; `hsngval`, the median value of a housing unit; and `hsnggrow`, the percentage growth in housing. We are going to see if we can predict the median rent in a state from the data we have on income and housing provision.

Enter the data into stata using the command `use "$datadir/hsng.dta", clear`.

Initial Regression

Use the command `regress rent hsngval hsnggrow hsng faminc` to fit a linear model which predicts `rent` from `faminc`, `hsng`, `hsngval`, and `hsnggrow`.

4.27 How many observations are used in fitting this regression model ?

4.28 How many of the predictor variables are statistically significant ?

4.29 What is the coefficient for `hsnggrow` and its 95% confidence interval ?

4.30 How would you interpret this coefficient and confidence interval ?

4.31 What is the value of R^2 for this regression ? What does this mean ?

Diagnostics: Constancy of Variance

Create standardised residuals using the command `predict rstand, rstand`. Create predicted values using the command `predict pred_val`. Now produce a graph of standardised residuals against predicted values with the command `scatter rstand pred_val`.

4.32 Would you conclude that the variance of the residuals is the same for all predicted values of `rent` ?

4.33 Compare the plot you have just produced to the plot produced by `rvfplot`. Would you have come to the same conclusion about the constancy of variance using this plot ?

Diagnostics: Linearity

Produce a CPR plot for each of the variables `faminc`, `hsng`, `hsngval`, and `hsnggrow`, using the command `cprplot` *`varname`*.

4.34 Is there any evidence of non-linearity in the association between the four predictor variables and the outcome variable ?

Diagnostics: Influence

Calculate Cook's distance for each observation, using the command `predict cooksd, cooksd`. Produce a graph of Cook's distance against the predicted values with `scatter cooksd pred_val`.

4.35 Are there any observations with an unusually large Cook's distance ?

4.36 If so, which state or states ? (You can use the command `list state if cooksd >` *`x`* to find out, by putting in a suitable value for *x*.)

Rerun the regression analysis excluding any states with a Cook's distance of greater than 0.5. Use the command

```
regress rent hsngval hsnggrow hsng faminc if cooksd < 0.5
```

4.37 Compare the coefficients and confidence intervals for each of the 4 predictors. Have any of them changed substantially ? Are any of them no longer significantly associated with the outcome ?

Now generate new predicted values with the command `predict pred2`. Compare the new predicted values with the old ones using `scatter pred_val pred2`.

4.38 Is it important to exclude the influential observation(s) from the regression analysis ?

Diagnostics: Normality

4.39 Produce a normal plot of the standardised residuals with `qnorm rstand`. Do the plotted points lie reasonably close to the expected straight line ?

4.40 Use `swilk rstand` to test whether the residuals are normally distributed. Does the result of the test confirm your answer to the previous question ?

6. More about Linear Models

6.1. Categorical Predictors in Linear Models

So far we have considered models in which both the outcome and predictor variables are continuous. However, none of the assumptions of the linear model impose any conditions on the x-variables. The predictor variables in a linear model can have any distribution. This makes it possible to include categorical predictors (sometimes referred to as *factors*) in a linear model.

6.1.1. A Dichotomous Variable

Suppose that we are involved in a clinical trial, in which subjects are given either an active treatment or placebo. We have an outcome measure, Y, that we wish to compare between the two treatment groups. We can create a variable, x, which has the value 0 for the subjects on placebo and 1 for the subjects on active treatment. We can then form the linear model

$$Y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$$

In this model, β_0 is the expected value of Y if $x = 0$, i.e. in the placebo group. β_1 is the expected increase in Yg as x increases by 1, i.e. the expected difference between the placebo and active treatment groups.

Does the above model satisfy the assumptions of the linear model ? Well, the mean of Y is certainly a linear function of x. Whether the individual observations of Y are independent depends on the experimental design, but if they all represent different individuals they should be independent. Whether the variance of Y is constant remains to be seen, but since x can only take two values in this case, it amounts to the variance of Y in the placebo group being the same as the variance of Y in the active treatment group. Finally, a linear model would only be appropriate if Y followed a normal distribution. Again, this needs to be tested, but the assumptions of the linear model can be met as easily for a dichotomous predictor as for a continuous one.

Here is an example of using a linear model in the above scenario. The data used for this example was simulated, using the commands

```
set obs 40
gen x = _n > 20
gen Y = 3 + x + invnorm(uniform())
```

This creates a set of 40 observations, 20 with x=0 and 20 with x=1. Y is normally distributed with variance 1, and has a mean of 3 if x=0 and 4 if x=1.

When I analysed this data, using `regress Y x`, I got the following output[a]:

```
. regress Y x

      Source |       SS       df       MS              Number of obs =      40
-------------+------------------------------           F(  1,    38) =   10.97
       Model |  9.86319435     1  9.86319435           Prob > F      =  0.0020
    Residual |  34.1679607    38   .89915686           R-squared     =  0.2240
-------------+------------------------------           Adj R-squared =  0.2036
       Total |   44.031155    39  1.12900398           Root MSE      =  .94824

------------------------------------------------------------------------------
           Y |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           x |   .9931362   .2998594     3.31   0.002     .3861025     1.60017
       _cons |     3.0325   .2120326    14.30   0.000     2.603262    3.461737
------------------------------------------------------------------------------
```

What do we conclude from the above output ? Firstly, we can say that x is a significant predictor of Y ($p = 0.002$), which in this case means that the outcome differs significantly between the placebo and active treatment groups. The estimated difference between the two groups is 0.993, with a 95% confidence interval of (0.39, 1.60), which is close to the true value of 1. The mean value among the placebo group is 3.03, with a confidence interval of (2.60, 3.46), which is again close to the true value of 3.

Note that the values that x takes need not be 0 and 1: it can take any two values. However, the interpretation of the coefficients depends on the values that it takes. β_0 is the expected value of Y when $x = 0$, so if the two groups were given x-values of 1 and 2, β_0 would not correspond the mean value of Y in either of the groups. For this reason, dichotomous variables are always best coded as 0 and 1.

T-Test

We saw previously that the way to test for a difference in a normally distributed outcome between two groups is to use a t-test. In fact, the above analysis is *exactly* equivalent to a t-test, as the following stata output shows.

Note that the difference between the two groups and its standard error are *exactly* the same as in the linear model above, and consequently the p-value from the test the the difference is not equal to 0 is also exactly the same. Hence we can say that the t-test is a special case of a linear model. It requires the same assumptions (Y is normally distributed with the same variance in both groups) and leads to the same conclusions. However, the linear model has the advantage that it can incorporate adjustment for other variables.

[a] Note that the random number generator is used in generating Y through the function `uniform`, so if you repeat this analysis you will not get exactly the same results, but they should be similar.

```
. ttest Y, by(x)

Two-sample t test with equal variances
------------------------------------------------------------------------------
   Group |     Obs        Mean    Std. Err.   Std. Dev.   [95% Conf. Interval]
---------+--------------------------------------------------------------------
       0 |      20      3.0325    .2467866    1.103663    2.515969     3.54903
       1 |      20    4.025636    .1703292    .7617355    3.669133    4.382139
---------+--------------------------------------------------------------------
combined |      40    3.529068    .1680033    1.062546    3.189249    3.868886
---------+--------------------------------------------------------------------
    diff |           -.9931362    .2998594               -1.60017   -.3861025
------------------------------------------------------------------------------
    diff = mean(0) - mean(1)                                      t =  -3.3120
Ho: diff = 0                                     degrees of freedom =       38

    Ha: diff < 0                 Ha: diff != 0                 Ha: diff > 0
 Pr(T < t) = 0.0010         Pr(|T| > |t|) = 0.0020          Pr(T > t) = 0.9990
```

6.1.2. A Categorical Variable with Several Categories

.

The above method works if there only two categories, but what happens if there are more than two ? Clearly, we cannot use $x = 1, 2, \ldots k$ to represent the k categories, since this is treating x as a continuous variable: the expected value of Y would be $\beta_0 + \beta_1$ in group 1, $\beta_0 + 2\beta_1$ in group 2 etc.

Instead, we use a series of “dummy” or “indicator” variables. Indicator variables take the values 0 or 1, and we need to have enough variables that each group has a different combination, which requires $k - 1$ variables if we have k groups. We have already seen that we require a single variable if we have two groups, and table 6.1 below shows how we can use two variables to determine which one of three groups an observation belongs to.

Group	x_1	x_2
A	0	0
B	1	0
C	0	1

Table 6.1.: Use of indicator variables to identify several groups

Here we have 3 groups, so we need 2 indicator variables. The linear model is therefore

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \varepsilon$$

In this model, β_0 is the expected value of Y when $x_1 = 0$ and $x_2 = 0$, i.e. the expected value of Y in group A. β_1 represents the change in Y as x_1 increases by 1, i.e. the difference between group A and group B. β_2 represents the change in Y as x_2 increases by 1, i.e. the difference between group A and group C.

Here is an example to illustrate the use of indicator variables, again using simulated data. There are three groups, with indicator variables `x1` and `x2` defined as in table 6.1. `Y` is normally distributed with variance 1 and mean 3 in group A, 5 in group B and 4 in group C. The results of analysing this data in stata are given below

```
. regress Y x1 x2

      Source |       SS       df       MS              Number of obs =      60
-------------+------------------------------           F(  2,    57) =   16.82
       Model |  37.1174969     2  18.5587485           Prob > F      =  0.0000
    Residual |  62.8970695    57  1.10345736           R-squared     =  0.3711
-------------+------------------------------           Adj R-squared =  0.3491
       Total |  100.014566    59  1.69516214           Root MSE      =  1.0505

------------------------------------------------------------------------------
           Y |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          x1 |   1.924713   .3321833     5.79   0.000     1.259528    2.589899
          x2 |   1.035985   .3321833     3.12   0.003     .3707994    1.701171
       _cons |   3.075665   .2348891    13.09   0.000     2.605308    3.546022
------------------------------------------------------------------------------
```

From this output we can conclude that there are highly significant differences in Y between the three groups (from the F-statistic, 16.82, which gives a p-value of 0.0000). The mean value in group A (i.e. when `x1` and `x2` are both 0) is 3.08, with a 95% confidence interval of (2.61, 3.55), close to the true value of 3. The difference between group A and group B is estimated by the coefficient of `x1`, which is 1.92 (95% CI; 1.26, 2.59), compared to the true value of 2, and the difference between group A and group C is estimated by the coefficient of `x2`, which is 1.04 (95% CI; 0.37, 1.70), compared to the true value of 1.

In the above analysis, groups B and C are both compared to group A. This is how such analyses are generally performed: one group is chosen as a baseline or reference group (all of the indicator variables are set to 0 in this group), and the other groups are compared to it. However, it may be that we are also interested in the difference between group B and group C, which is not given directly in the above output. We know that the expected value in group B is $\beta_0 + \beta_1$, whilst the expected value in group C is $\beta_0 + \beta_2$. Hence the expected difference between the groups is $(\beta_0 + \beta_1) - (\beta_0 + \beta_2) = \beta_1 - \beta_2$. This difference can be calculated from the regression output as 1.92 - 1.04 = 0.88, but testing whether this is significantly different from 0, or putting a confidence interval around it, is less straightforward. Fortunately, stata can do it for us using the command `lincom`, short for “Linear Combination”, since $\beta_1 - \beta_2$ is a linear combination of the parameters β_1 and β_2. How this is done is illustrated below

```
. lincom x1 - x2

 ( 1)  x1 - x2 = 0

------------------------------------------------------------------------------
           Y |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         (1) |   .8887284   .3321833     2.68   0.010     .2235428    1.553914
------------------------------------------------------------------------------
```

Thus we can conclude that the difference between groups B and C is significant (p = 0.01), with a 95% confidence interval of (0.22, 1.55), compared to a true value of 1.

Other linear combinations may be of interest. For example, the mean value of Y in group B is given by $\beta_0 + \beta_1$. Calculating this using `lincom` gives the following output

```
. lincom _cons + x1

 ( 1)  x1 + _cons = 0

------------------------------------------------------------------------------
           Y |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         (1) |   5.000378   .2348891    21.29   0.000     4.530021    5.470736
------------------------------------------------------------------------------
```

In other words, the mean value of Y in group B is 5.00, with a 95% confidence interval (4.53, 5.47). This is close to the true value of 5.

Generating indicator variables yourself can be tedious and error-prone, particularly if there are a large number of categories. However, Stata can do it for you. Suppose that in the above example, we did not have x_1 and x_2, but only a variable `group`, which took the values "1", "2" and "3". We can tell stata to treat `group` as a categorical variable by adding `i.` as a prefix to its name.

The regression model can be fitted with the command

```
regress Y i.group
```

which produces the following output:

```
. regress Y i.group

      Source |       SS       df       MS              Number of obs =      60
-------------+------------------------------           F(  2,    57) =   16.82
       Model |  37.1174969     2  18.5587485           Prob > F      =  0.0000
    Residual |  62.8970695    57  1.10345736           R-squared     =  0.3711
-------------+------------------------------           Adj R-squared =  0.3491
       Total |  100.014566    59  1.69516214           Root MSE      =  1.0505

------------------------------------------------------------------------------
           Y |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       group |
          2  |   1.924713   .3321833     5.79   0.000     1.259528    2.589899
          3  |   1.035985   .3321833     3.12   0.003     .3707994    1.701171
             |
       _cons |   3.075665   .2348891    13.09   0.000     2.605308    3.546022
------------------------------------------------------------------------------
```

This can be seen to be identical to the previous analysis, except for the labelling of the groups.

In order to use the coefficients for group 2 and group 3 in `lincom`, we can no longer use the variable names, since the variable name is `group` for both of them. Instead, we use `2.group` for group 2 and `3.group` for group 3.

ANOVA

This section can safely be skipped over if you are unfamiliar with the technique of Analysis of Variance, or ANOVA

Just as linear regression with a single indicator variable is equivalent to a t-test, linear regression with several indicator variables is equivalent to an ANOVA test. This is illustrated by the stata output below, analysing the same data as in section 6.1.2.

```
. oneway Y group

                        Analysis of Variance
    Source              SS         df      MS            F     Prob > F
------------------------------------------------------------------------
Between groups      37.1174969      2   18.5587485     16.82      0.0000
 Within groups      62.8970695     57   1.10345736
------------------------------------------------------------------------
    Total           100.014566     59   1.69516214

Bartlett's test for equal variances:  chi2(2) =   0.3023  Prob>chi2 = 0.860
```

The anova table above is identical to that produced by the linear regression analysis. Again, the assumptions required for performing ANOVA are the same as those required for performing linear regression with several indicator variables: the `regress` command is simply a convenient way of producing exactly the same analysis. Bartlett's test, given at the bottom of the above printout, is a test that the variance of Y is the same in all three groups. Since x only takes three values, this is testing that the variance is the same for all values of x, which is one of the assumptions of the linear model.

6.1.3. Mixing Categorical & Continuous Variables

In the preceding sections we have only dealt with situations in which there was only a single categorical variable in the linear model. However, it is also possible to mix categorical and continuous predictors in the same model. For example, suppose that in the clinical trial discussed in section 6.1.1, we expect the outcome variable Y to vary with age, as well as with treatment. In such a case, we would want to fit both age and treatment in our linear model, for two reasons

1. It may be that the two treatment groups differ in age. If this is the case, a difference in Y between the two groups may be because of the age difference, rather than a treatment effect. This is known as confounding, and will be discussed in detail in section 6.2. Here, I will just say that if we fit both age and treatment, the treatment coefficient measures what the difference between the two groups would have been if the groups had not differed in age.

2. If age is an important predictor of Y, then by including age in our model, we are reducing the amount of unexplained variation, and with therefore achieve narrower confidence intervals around our estimated effect of treatment.

Again, it is easiest to illustrate this using simulated data. Suppose that we have recruited subjects aged 20 – 40, and that in the placebo group, Y is normally distributed with mean 20 - age/2 and variance

1, and that in the active treatment group, the mean of Y is 21 - age/2. A scatter-plot of this simulated data is given in Figure 6.1.

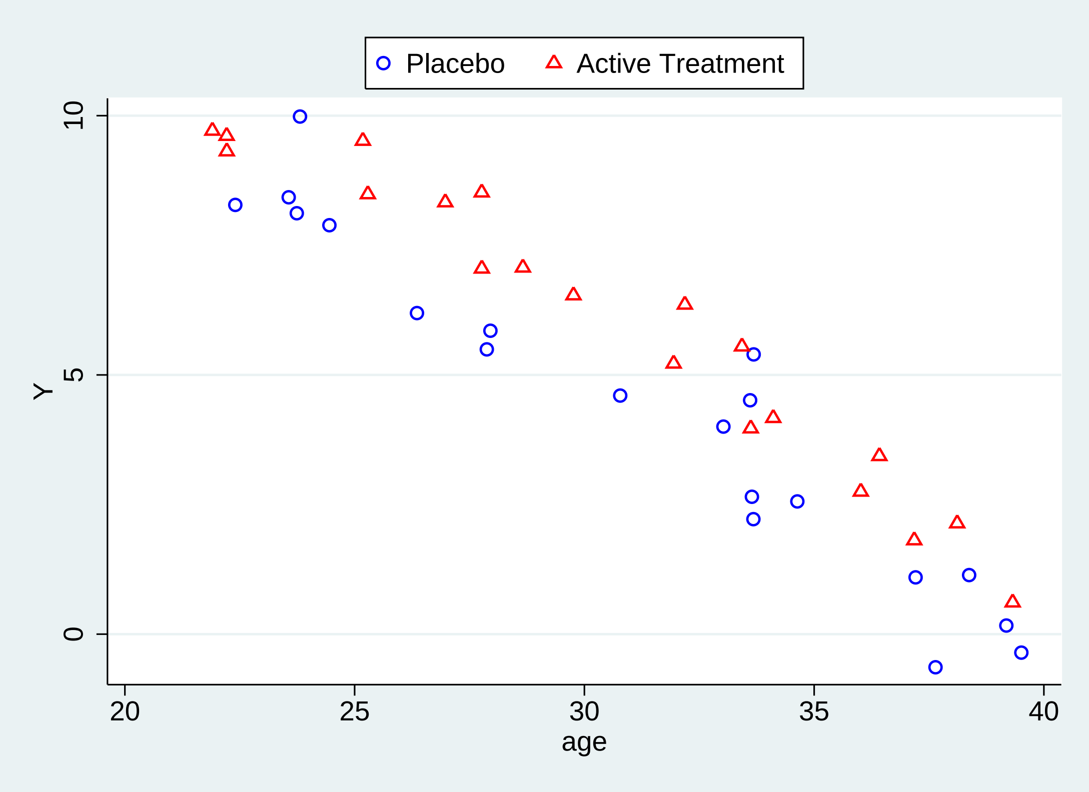


Figure 6.1.: Variation in outcome with age in simulated trial data

In this case, if we simply fit treatment as our predictor variable, we find the difference between the two groups is not significant.

```
. regress Y treat

      Source |       SS       df       MS              Number of obs =      40
-------------+------------------------------           F(  1,    38) =    2.86
       Model |  26.5431819     1  26.5431819           Prob > F      =  0.0989
    Residual |  352.500943    38  9.27634061           R-squared     =  0.0700
-------------+------------------------------           Adj R-squared =  0.0456
       Total |  379.044125    39  9.71908013           Root MSE      =  3.0457

------------------------------------------------------------------------------
           Y |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       treat |   1.629208   .9631376     1.69   0.099    -.3205623    3.578978
       _cons |   4.379165   .6810411     6.43   0.000      3.00047    5.757861
------------------------------------------------------------------------------
```

Here, although the observed effect of treatment (1.63) is close to its hypothetical value (1.00), the confidence interval is very wide, because of the variation with age which is being treated as random variation in this model. The predicted values of Y in the treated and untreated groups from this model are shown in Figure 6.2. Although the predicted values are higher in the treated arm, these predictions are not close to the observed values for those less than 25 or more than 35.

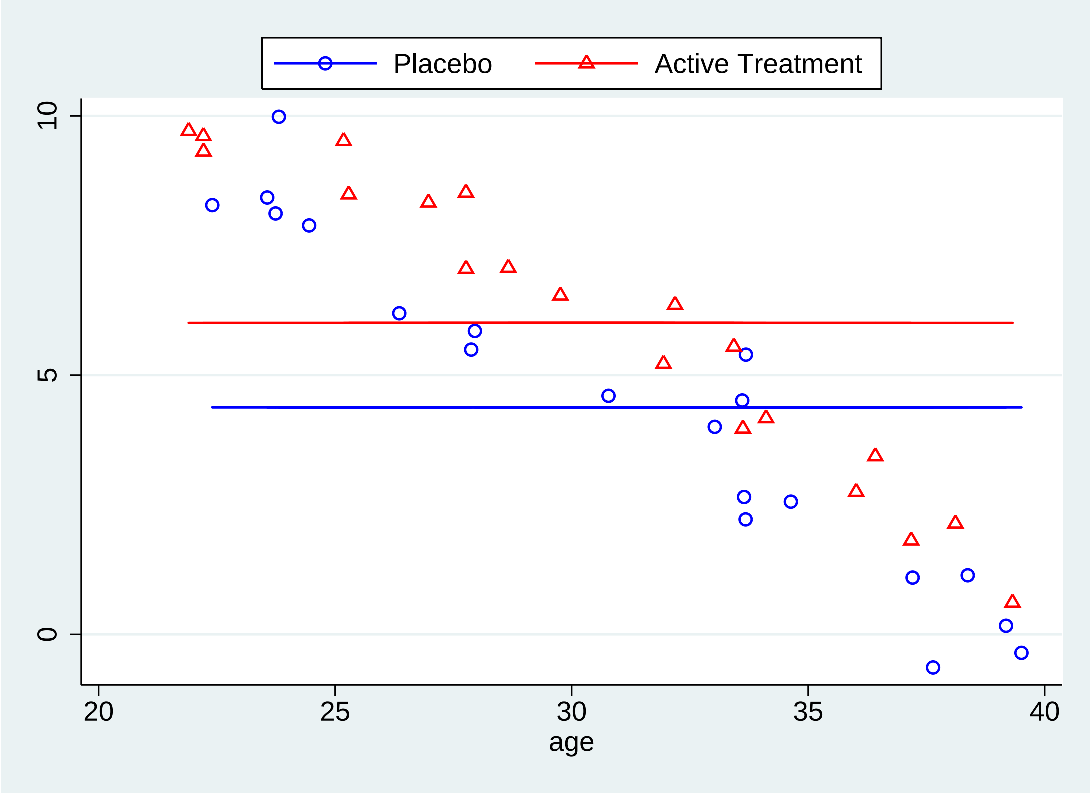


Figure 6.2.: Variation in outcome with age in simulated trial data, and predicted values

If we fit age as well as treatment, the observed effect changes very little (to 1.24), but it is now very highly significant, because its standard error is reduced greatly. We have seen that the standard error of a coefficient is equal to $\frac{\sigma}{\sqrt{n}s_x}$: $\sqrt{n}$ and s_x are unchanged, but by including age in the model, we have decreased σ greatly, and consequently decreased the standard error.

```
. regress Y treat age

      Source |       SS       df       MS              Number of obs =      40
-------------+------------------------------           F(  2,    37) =  262.58
       Model |  354.096059     2  177.04803            Prob > F      =  0.0000
    Residual |  24.9480658    37  .674272049           R-squared     =  0.9342
-------------+------------------------------           Adj R-squared =  0.9306
       Total |  379.044125    39  9.71908013           Root MSE      =  .82114

------------------------------------------------------------------------------
           Y |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       treat |   1.238752   .2602711     4.76   0.000     .7113924    1.766111
         age |  -.5186644   .0235322   -22.04   0.000    -.5663453   -.4709836
       _cons |   20.59089   .7581107    27.16   0.000     19.05481    22.12696
------------------------------------------------------------------------------
```

One way to think of this is that in the first analysis, we are selecting a subject at random from each of the two treatment groups, and measuring the difference in Y between them. This varies enormously, depending on the ages of the two subjects. However, if we select two subjects *of the same age* and measure the difference in Y between them, there will be much less variation: this is what we do when we include age in the regression equation.

From the second analysis, the expected value of Y in the placebo group is given by $20.59 - 0.52 \times age$, whilst the expected value in the active treatment group is given by $20.59 - (0.52 \times age) + 1.24$. These predictions are shown in Figure 6.3, along with the data from Figure 6.1 above.

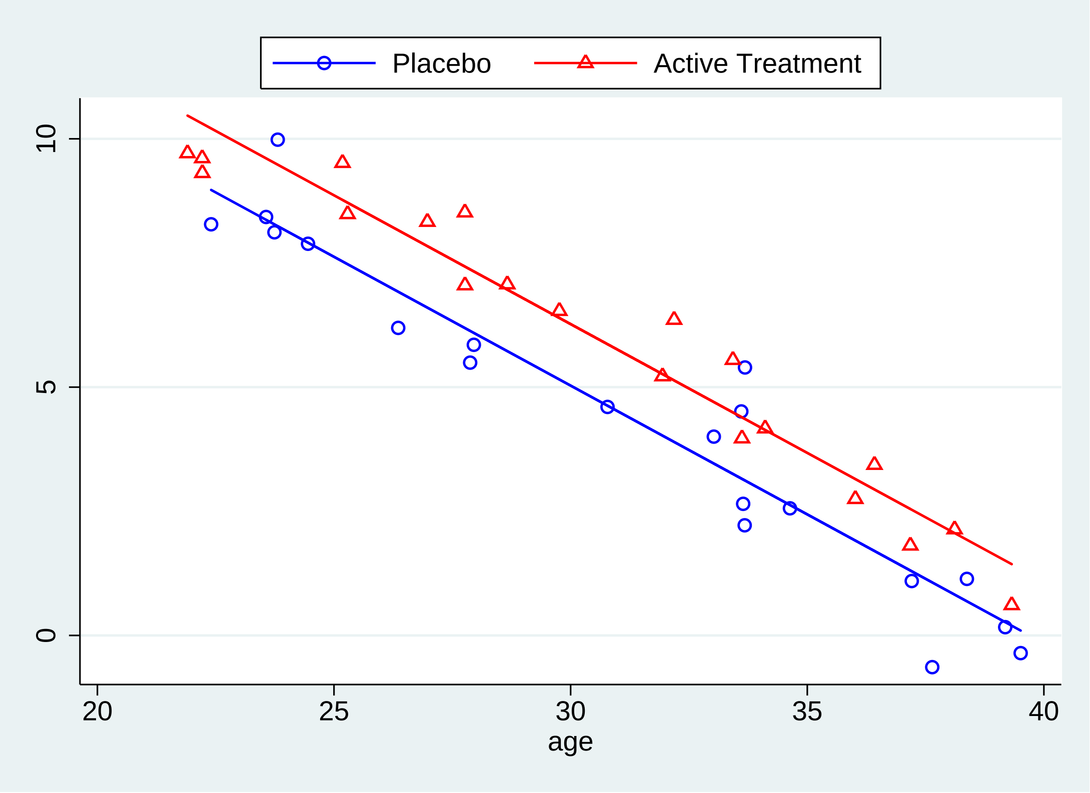


Figure 6.3.: Variation in outcome with age in simulated trial data, and predicted values

6.1.4. Interactions

In the above analysis, we have assumed that the effect of treatment is the same at all ages. Geometrically, this means that we have fitted parallel lines to the plot of Y against age, as seen in Figure 6.3. However, this assumption may not be true: it may be that the treatment is more effective in the older patients than in the younger ones, so the fitted lines should be further apart in the older subjects. A situation like this, in which the effect of one variable on the outcome depends on the value of another variable, is called an "interaction": we say that there is an interaction between age and treatment.

For example, consider the data illustrated in Figure 6.4. In this example, the effect of treatment is to reverse the effect of age, so that the outcome in the treated group is normally distributed with mean 10 and variance 1, whilst the outcome in the untreated group again has a mean of 20 - age/2 and a variance of 1. Thus the predicted values in the treated and untreated groups get further apart with increasing age.

Fitting parallel lines to these two groups does not provide a good fit to the data, as Figure 6.4 shows.

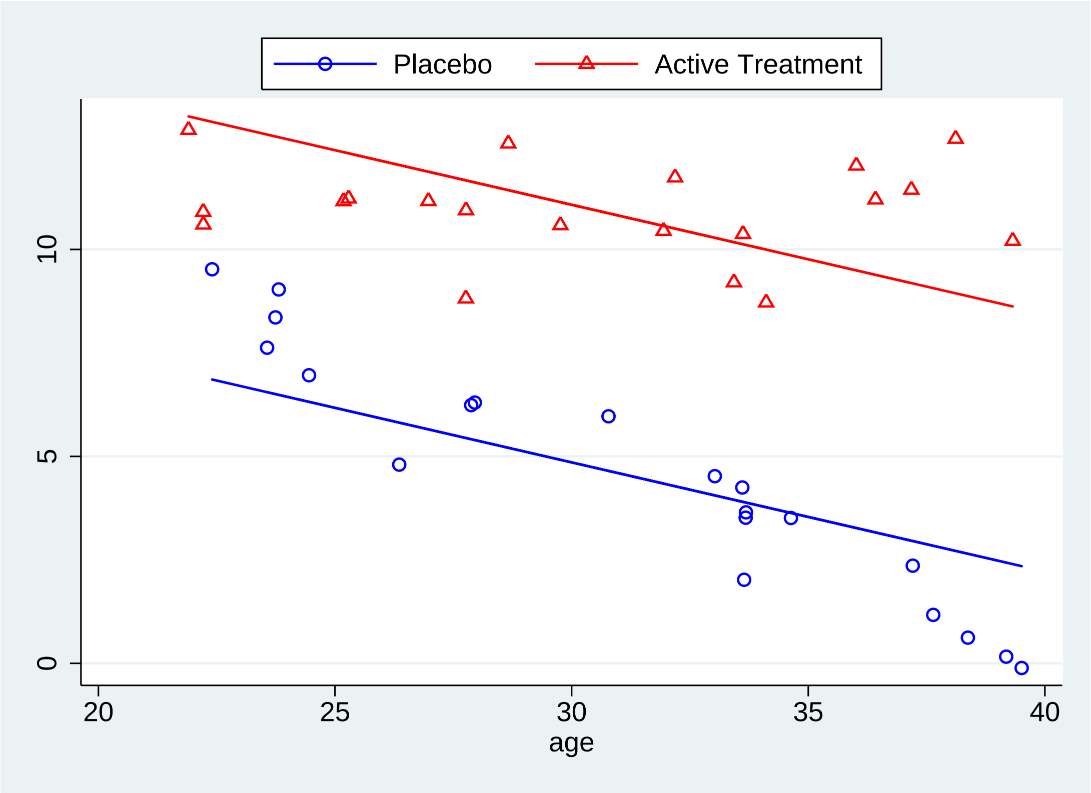


Figure 6.4.: Variation in outcome with age in simulated trial data, in which age interacts with treatment, with predicted values from regression model excluding interaction.

Fitting an interaction term can be thought of as fitting the two separate equations

$$Y = \begin{cases} \beta_{00} + \beta_{10} \times \texttt{age} + \varepsilon & \text{if } \texttt{treat} = 0 \\ \beta_{01} + \beta_{11} \times \texttt{age} + \varepsilon & \text{if } \texttt{treat} = 1 \end{cases} \tag{6.1}$$

The two groups each have different intercepts (β_{00} and β_{01}) and slopes with age (β_{10} and β_{11}). However, it is possible to combine the two above equations into the single equation

$$Y = \beta_{00} + \beta_{10} \times \text{age} + (\beta_{01} - \beta_{00}) \times \text{treat} + (\beta_{11} - \beta_{10}) \times \text{age} \times \text{treat} + \varepsilon. \tag{6.2}$$

Thus fitting the linear model with an interaction term amounts to including as predictors the variables `age`, `treat`, and a new variable formed by multiplying `age` and `treat` together. The intercept in this model is the intercept in placebo group (β_{00}), the coefficient of `age` is the slope with age in the placebo group (β_{10}), the coefficient of `treat` is the difference between the intercept in the active group and the intercept in the placebo group ($\beta_{01} - \beta_{00}$), and the coefficient of the interaction term is the difference in the slopes between the active and treatment groups ($\beta_{11} - \beta_{10}$).

Just as Stata can automatically generate indicator variables, it can also automatically generate interaction terms. To request an interaction between to variables, you include the symbol # between them. So if we had two categorical variables and wanted to consider the interaction between them, we would add `i.var1#i.var2` to the regression model. There is also a shorthand we can use: `i.var1##i.var2` is equivalent to `i.var1 i.var2 i.var1#i.var2`. If we wish to include a continuous variable in an interaction, we precede the variable name with `c.` rather than `i.`

So, to fit a linear model containing age, treatment and their interaction to the data in Figure 6.4, we would enter

```
regress Y i.treat##c.age
```

and obtain the following output.

```
. regress Y i.treat##c.age

      Source |       SS       df       MS              Number of obs =      40
-------------+------------------------------           F(  3,    36) =  173.38
       Model |  563.762012     3  187.920671           Prob > F      =  0.0000
    Residual |  39.0189256    36  1.08385904           R-squared     =  0.9353
-------------+------------------------------           Adj R-squared =  0.9299
       Total |  602.780938    39  15.4559215           Root MSE      =  1.0411

------------------------------------------------------------------------------------
                Y |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
------------------+-----------------------------------------------------------------
            treat |
Active Treatment  |  -8.226356   1.872952    -4.39   0.000    -12.02488   -4.427833
              age |  -.4866572   .0412295   -11.80   0.000    -.5702744     -.40304
                  |
      treat#c.age |
Active Treatment  |   .4682374   .0597378     7.84   0.000     .3470836    .5893912
                  |
            _cons |   19.73531   1.309553    15.07   0.000     17.07942    22.39121
------------------------------------------------------------------------------------
```

The interaction term (written `treat#c.age`) is highly significant (p = 0.000), so we have good evidence that the rate of change of Y with age is different between the two groups. The coefficient for `age` is still the slope of the graph of Y against `age` in the placebo group (i.e. when `treat` = 0). However, the slope of Y against age in the treated group is now given by the sum of the coefficients `age + treat#c.age` This slope can be calculated using `lincom` as before, to give

```
. lincom age + 1.treat#c.age

 ( 1)  age + 1.treat#c.age = 0

------------------------------------------------------------------------------
           Y |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         (1) |  -.0184198   .0432288    -0.43   0.673    -.1060919    .0692523
------------------------------------------------------------------------------
```

This is very nearly 0, suggesting that Y does not change with age in the treated group, although it does reduce with age in the placebo group. Figure 6.5 shows the same data as Figure 6.4, but the predicted values in this case include the interaction term. This shows clearly that the slopes in the treatment and placebo groups are quite different: no change with age in the treated group, reduction with age in the placebo group.

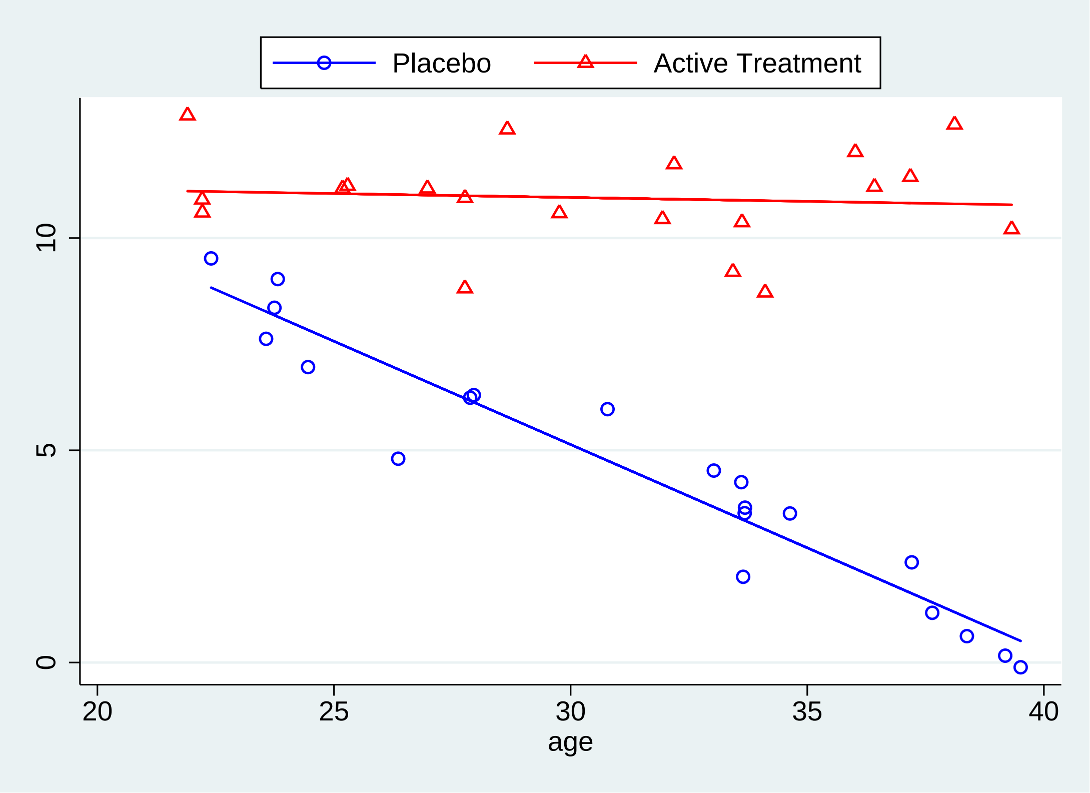


Figure 6.5.: Variation in outcome with age in simulated trial data, in which age interacts with treatment, with predicted values

As in the previous examples, the intercept (`_cons`) is the expected value of Y when all the other variables are equal to 0: i.e. the expected value at age 0 in the placebo group. The coefficient for `treat` measures the difference between the treated and untreated groups *at age 0*. It cannot be interpreted as the difference between the placebo and treated groups in this case, since the treatment effect varies with age, as seen in Figure 6.5. In fact, it is of little intrinsic meaning in this case, since

1. It only applies to subjects of age 0, and we are unlikely to be interested in them

2. The youngest subjects in our sample were of age 20, so we are extrapolating a long way beyond our data.

The effect of treatment at age a can be calculated as `treat` $+ a \times$ `treat#c.age`. Again, `lincom` can be used to perform the calculations, so the treatment effect at age 20 would be given by:

```
. lincom 1.treat + 20*1.treat#c.age

 ( 1)  1.treat + 20*1.treat#c.age = 0

------------------------------------------------------------------------------
           Y |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         (1) |   1.138392   .7279832     1.56   0.127    -.3380261     2.61481
------------------------------------------------------------------------------
```

The treatment effect at age 40 would be given by:

```
 ( 1)  1.treat + 40*1.treat#c.age = 0

------------------------------------------------------------------------------
           Y |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         (1) |   10.50314   .6378479    16.47   0.000     9.209524    11.79676
------------------------------------------------------------------------------
```

In other words, as shown in Figure 6.5, there is little difference between the treated & placebo groups at age 20, but considerable difference at age 40.

6.1.5. *Testing Several Parameters With* `testparm`

Often when dealing with categorical variables, we wish to test whether several parameters are signficantly different from 0 at the same time. The command that enables us to do so is `testparm`, which is a contraction of "test parameters". The syntax for testparm is simply `testparm` *`varlist`*, where *varlist* is a list of variables which all appeared in the last regression model. `testparm` tests the hypothesis that $\beta = 0$ for all variables in *varlist*.

6.2. Confounding

6.2.1. *What is Confounding ?*

A linear model can be used to show that two variables are associated, i.e. if one increases, the other also tends to increase (or tends to decrease, if the association is negative). It may be that changes in one cause the changes in the other, but it may be that there is a third factor which is associated with both.

As an example of confounding, consider the data in `auto.dta` which we looked at in a previous practical. Imagine that we are interested in whether US-built cars differ from non-US-built cars in their fuel consumption. We have already seen that the fuel consumption, in miles per gallon, is stored in the variable `mpg`. There is also a variable `foreign`, which is 0 for US vehicles and 1 for non-US vehicles. So we can test for a difference using a linear model to predict `mpg` from `foreign`. The output is shown below.

```
. regress mpg foreign

  Source |       SS       df       MS              Number of obs =      74
---------+------------------------------           F(  1,     72) =   13.18
   Model |  378.153515     1  378.153515           Prob > F      =  0.0005
Residual |  2065.30594    72  28.6848048           R-squared     =  0.1548
---------+------------------------------           Adj R-squared =  0.1430
   Total |  2443.45946    73  33.4720474           Root MSE      =  5.3558

------------------------------------------------------------------------------
     mpg |      Coef.   Std. Err.       t     P>|t|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
 foreign |   4.945804   1.362162      3.631   0.001       2.230384    7.661225
   _cons |   19.82692   .7427186     26.695   0.000       18.34634    21.30751
------------------------------------------------------------------------------
```

This shows a significant difference in fuel consumption between US and non-US vehicles, with foreign vehicles averaging nearly 5 miles per gallon more than US vehicles. However, remember that weight is a very important determinant of fuel consumption, and consider Figure 6.6. This shows a plot of fuel consumption against weight for both US and non-US vehicles. Clearly, the number of miles travelled per gallon decreases as the weight of the vehicle increases. However, notice that most of the lightest vehicles are non-US, whilst all of the heaviest vehicles are US-built. Could the advantage in fuel consumption of non-US vehicles be due to their lighter weight ?

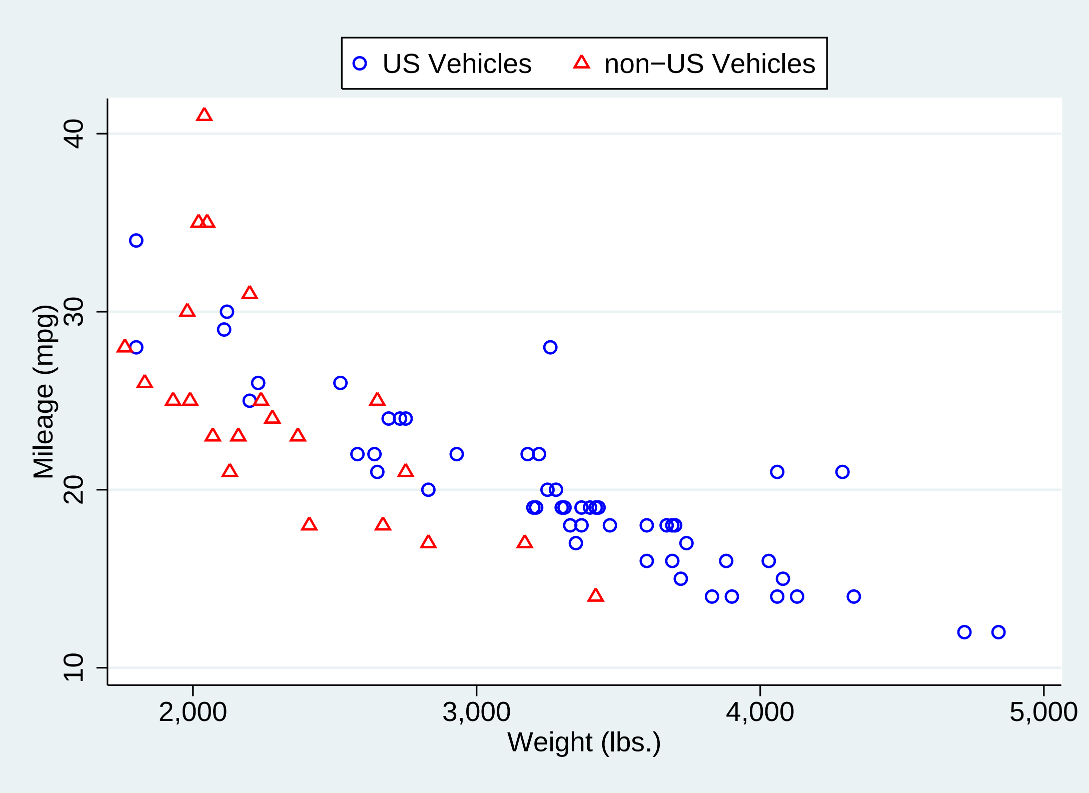


Figure 6.6.: Plot of m.p.g. against weight for US and non-US vehicles

We can test this by fitting both `weight` and `foreign` as predictors in our linear model. Then, the coefficient of `foreign` measures the difference in fuel consumption between a US and a non-US vehicle *of the same weight*, rather than just a randomly selected US and non-US vehicle as we did before. The results of fitting this regression are given below.

```
. regress mpg foreign weight

  Source |       SS       df       MS              Number of obs =      74
---------+------------------------------           F(  2,    71) =   69.75
   Model |   1619.2877     2  809.643849           Prob > F      =  0.0000
Residual |  824.171761    71   11.608053           R-squared     =  0.6627
---------+------------------------------           Adj R-squared =  0.6532
   Total |  2443.45946    73  33.4720474           Root MSE      =  3.4071

------------------------------------------------------------------------------
     mpg |      Coef.   Std. Err.       t     P>|t|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
 foreign |  -1.650029   1.075994     -1.533   0.130        -3.7955    .4954421
  weight |  -.0065879   .0006371    -10.340   0.000      -.0078583   -.0053175
   _cons |    41.6797   2.165547     19.247   0.000       37.36172    45.99768
------------------------------------------------------------------------------
```

This model fits much better than the previous model (R^2 = 66%, as opposed to 15% previously). However, the effect of `foreign` is no longer statistically significant (p = 0.13), suggesting that the difference between US and non-US vehicles we saw before could be explained by differences in weight. We can conclude that the apparent difference in fuel consumption between US and non-US cars is due to differences in weight between US and non-US cars.

6.2.2. How do we recognise a confounder ?

In order to confound the association between a predictor variable and an outcome, a variable has to satisfy the following conditions:

1. The confounder must be associated with the outcome.

2. The association between outcome and confounder must be independent of the predictor variables.

3. The confounder cannot be on the causal path linking the predictor to the outcome.

Determining whether a variable satisfies the above conditions, particularly the last, can be tricky. It is not possible without some understanding of the mechanism by which the predictor is associated with the outcome.

For example, consider the difference in fuel consumption between US-made and non-US-made cars considered earlier. We have seen that the non-US made cars have better fuel consumption because they are lighter, but why are they lighter ? If they are lighter because non-US car designers have clever tricks to reduce the weight of a car without affecting its capacity, comfort etc, then weight is a path variable, not a confounder, and non-US car designers produce cars with better fuel consumption. However, suppose the US and non-US designers were given different briefs: the US designers were asked to produce larger cars since that is what sells in the US. In this case, then weight is not a path variable but a confounder. In this case, the second analysis above is appropriate and suggests that the US designers are every bit as good at producing cars with low fuel consumption as non-US designers, given their different briefs.

6.2.3. How do we allow for confounding ?

Dealing with a confounder once it has been identified is very straightforward. Simply adding a confounding variable as a predictor in the linear model will allow for confounding. Remember that a coefficient in the linear model measures the effect of a variable *when all other variables do not change*, so the coefficient for the predictor we are interested in will be a measure of its effect on the outcome when the confounding variable does not change.

It should be noted that adjusting for confounding in this way assumes that

1. The confounder has been measured perfectly
2. The association between confounder and outcome is perfectly linear.

If either of these assumptions are not true, the linear model will remove *some*, but not *all*, of the effect of confounding. The *residual* confounding may still be enough to affect the parameter estimates.

Ultimately, confounding is a property of the real-world system we are attempting to model, not of the model itself. There is no way of determining from the data whether a variable is a confounder and should be adjusted for, or whether it is on the causal pathway and should not be adjusted for. The correct course of action depends on the mechanism by which the predictors and putative confounders affect the outcome, which requires a thorough knowledge of the subject area. Statistical methods for allowing for confounding are simple and straightforward to apply, the difficult problem is to know whether they are appropriate.

Confounding is a major problem for observational studies. When a significant association is shown in an observational study, it could be due to a causal effect (i.e. one variable is directly related to the other) or to confounding (i.e. one variable is correlated with another variable which directly affects the outcome).

6.3. Other Considerations

6.3.1. Variable Selection

Suppose that we have a number of predictor variables, from which to predict the outcome. Should we include all of the variables in our linear model, or only a subset of them ? And if we are to use a subset, how do we select that subset ?

If we are only interested in the predictions from the linear model, we can use all of the variables (provided we have enough data to make that reasonable). However, if we are interested in the mechanisms by which the predictors affect the outcome, it is of use to remove those “predictors” which are only predictive because they are correlated with other variables which genuinely affect Y. In addition, if we wish to use the model on further samples, it may be that there is a financial or logistical advantage to having a smaller number of variables to measure.

There are a number of automated ways which have been suggested for selecting variables in this context. I will explain them, as they are widely used. However, I do not recommend their use: they assume that you know nothing at all about the area you are researching, and in an ideal world that would not be true. Selecting variables on grounds other than statistical significance is preferable, be that choosing the cheapest variables to collect first, least invasive, most likely to be predictive on theoretical grounds etc.

Forward Selection

Suppose we are selecting from k predictors. First, we chose a significance level to use to determine whether a variable is a significant predictor or not. We will call this p_e, the significance level at which a predictor enters the regression model. Then we use each of the k predictors in turn to predict Y, and the one which explains the greatest proportion of the variance is our candidate for inclusion at this stage. We then look at the significance level of the t-test for $\beta = 0$ for this variable: if it is less than p_e, the variable is selected.

Once a variable has been selected in this way, each of the remaining variables are added to the model in turn. If any of these variables are significant at the p_e level after adjusting for all the variables selected so far, the most significant variable is added to the selection. This process repeats until either all of the variables have been selected, or none of them are significant at the p_e level after adjusting for the selected variables.

Backward Elimination

An alternative to forward selection is backward elimination. In this procedure, we chose a significance level at which a variable will be *removed* from the model, p_r. Then we produce a single linear model containing all k predictor variables, and compare the significance level of the least significant model to p_r. If it is greater than p_r, that variable is removed from the model and a linear model formed with the remaining $k - 1$ variables. The process is repeated until all remaining variables have significance levels less than p_r.

Backward selection can be better than forward selection if there are variables that are jointly good predictors, but not individually (as seen in the dataset `wood73`). However, if there are a number of strongly correlated variables, the results can be unreliable.

Stepwise Selection

Stepwise selection is a combination of the two preceding methods. The procedure starts as forward selection, but each time a variable is added to the model, all of the variables are tested to see if any can now be removed. Obviously, when using stepwise selection, p_r must be greater than or equal to p_e, otherwise a variable added in the "forward" part of the procedure could be removed in the "backward" part.

All Subsets

All three of the above methods have the drawback that not every possible combination of predictors is considered. In some cases it may be possible to consider every possible combination, and choose the one with the highest adjusted R^2. However, this can be very time consuming: if we have 10 predictors, there are 1023 possible subsets to be fitted, compared with at most 10 using forwards or backwards selection. Since all subsets regression is not implemented in stata, it will not be discussed further.

Some Caveats Concerning Variable Selection

Significance Levels in Variable Selection The main problem with variable selection is that the significance levels of the various parameters are no longer correct. The hypotheses we are testing to include or exclude variables are not independent of each other, and they are not randomly selected (we always test either the most significant variable or the least significant). How greatly the p-values differ from their nominal levels depends on the ratio of the sample size to the number of predictor variables: if there are many times as many observation as predictor variables, this problem will not be of great importance.

Differences in Models Selected It is not uncommon for the model chosen using forward selection to differ from that chosen using backward elimination. A measure of common sense is needed to determine which model to use. For example, if two variables are highly correlated, there will be little difference between a model containing one and a model containing the other.

Inappropriate variable choice Statistical significance should not be the only criterion for selecting predictor variables in a model. It may be that you know a particular variable is a predictor of your outcome, even though your sample is too small for the effect to be statistically significant. You should therefore have no qualms about forcing certain variables to be in the regression model even if the automatic variable selection methods would exclude them[b].

Variable selection in Stata

In stata, the `sw` command can be used to give either forward, backward or stepwise selection. The syntax for this command is

```
sw regress yvar xvars
```

The significance levels p_e and p_r are set using the options `pe()` and `pr()`: at least one of these must be set. If only `pe()` is set, forward selection is performed, whilst if only `pr()` is set, backwards elimination is performed. If both are set, stepwise selection is used.

[b] However, you will have to explain which variables were forced into the model and your reasons for doing so in the methods section of any resulting publication

For example, consider the `auto` dataset distributed with stata. Imagine we wish to determine which variables contribute to the weight of the vehicle. To use forwards selection, we would type

```
sw regress weight price headroom trunk length turn displ gear_ratio, pe(0.05)
```

to get the output

```
p = 0.0000 <  0.0500  adding   length
p = 0.0000 <  0.0500  adding   displ
p = 0.0015 <  0.0500  adding   price
p = 0.0288 <  0.0500  adding   turn

  Source |       SS       df       MS              Number of obs =      74
---------+------------------------------           F(  4,    69) =  293.75
   Model |  41648450.8     4  10412112.7           Prob > F      =  0.0000
Residual |  2445727.56    69  35445.3269           R-squared     =  0.9445
---------+------------------------------           Adj R-squared =  0.9413
   Total |  44094178.4    73  604029.841           Root MSE      =  188.27

------------------------------------------------------------------------------
  weight |      Coef.   Std. Err.       t     P>|t|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
  length |   19.38601   2.328203      8.327   0.000       14.74137    24.03064
   displ |   2.257083    .467792      4.825   0.000       1.323863    3.190302
   price |   .0332386   .0087921      3.781   0.000       .0156989    .0507783
    turn |   23.17863   10.38128      2.233   0.029       2.468546    43.88872
   _cons |  -2193.042   298.0756     -7.357   0.000      -2787.687   -1598.398
------------------------------------------------------------------------------
```

Using forward selection, first `length`, then `displ`, then `price` and finally `turn` are selected as significant predictors. The other variables, `headroom`, `trunk` and `gear_ratio` are not selected as significant.

To use backward elimination, we would type

```
sw regress weight price headroom trunk length turn displ gear_ratio, ///
pr(0.05)
```

and would get the output

```
p = 0.6348 >= 0.0500  removing headroom
p = 0.5218 >= 0.0500  removing trunk
p = 0.1371 >= 0.0500  removing gear_ratio

  Source |       SS       df       MS              Number of obs =      74
---------+------------------------------           F(  4,    69) =  293.75
   Model |  41648450.8     4  10412112.7           Prob > F      =  0.0000
Residual |  2445727.56    69  35445.3269           R-squared     =  0.9445
---------+------------------------------           Adj R-squared =  0.9413
   Total |  44094178.4    73  604029.841           Root MSE      =  188.27

------------------------------------------------------------------------------
  weight |      Coef.   Std. Err.       t     P>|t|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
   price |   .0332386   .0087921      3.781   0.000        .0156989    .0507783
    turn |   23.17863   10.38128      2.233   0.029        2.468546    43.88872
   displ |   2.257083    .467792      4.825   0.000        1.323863    3.190302
  length |   19.38601   2.328203      8.327   0.000        14.74137    24.03064
   _cons |  -2193.042   298.0756     -7.357   0.000       -2787.687   -1598.398
------------------------------------------------------------------------------
```

The final model here is exactly the same as we obtained from forward selection. First all variables are fitted, then headroom, trunk and gear_ratio are eliminated (in that order).

One important feature of sw is the ability to treat several variables as a single term. For example, if there is a categorical variable with 4 levels, it will be fitted using three indicator variables, as we have seen. Normally, these three variables should either all be included or all excluded. This can be achieved by putting parentheses around the variables to be treated together. For example, suppose we wish to predict the urban population of various American states from the total population and the region. If we are using forward selection, we would type[c]

```
xi:  sw regress popurban pop (i.region), pe(0.05)
```

and obtain the output

[c] Note that the older stata syntax for generating indicator variables needs to be used with sw, since this command does not understand factor variables. More information about the old syntax is given in Appendix A

```
i.region          _Iregion_1-4        (naturally coded; _Iregion_1 omitted)
                      begin with empty model
p = 0.0000 <  0.0500  adding   pop
p = 0.0003 <  0.0500  adding   _Iregion_2 _Iregion_3 _Iregion_4

      Source |       SS       df       MS              Number of obs =      50
-------------+------------------------------           F(  4,    45) =  794.57
       Model |  8.0830e+14     4  2.0208e+14           Prob > F      =  0.0000
    Residual |  1.1444e+13    45  2.5432e+11           R-squared     =  0.9860
-------------+------------------------------           Adj R-squared =  0.9848
       Total |  8.1975e+14    49  1.6730e+13           Root MSE      =  5.0e+05

------------------------------------------------------------------------------
    popurban |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         pop |   .8655805   .0154854    55.90   0.000     .8343914    .8967696
  _Iregion_2 |    -383325   222541.4    -1.72   0.092    -831546.4    64896.47
  _Iregion_3 |  -529247.1   210480.1    -2.51   0.016    -953175.8   -105318.3
  _Iregion_4 |   313714.2     221173     1.42   0.163      -131751    759179.4
       _cons |  -402777.4   188162.4    -2.14   0.038      -781756   -23798.79
------------------------------------------------------------------------------
```

Here, `pop` enters the model first, then `_Iregion_2`, `_Iregion_3` and `_Iregion_4`. This is despite the fact that `_Iregion_2` and `_Iregion_4` are not individually statistically significant (p = 0.09 & 0.16 respectively). The three variables taken as a whole are highly significant, as can be shown using `testparm`:

```
. testparm _I*

 ( 1)  _Iregion_2 = 0.0
 ( 2)  _Iregion_3 = 0.0
 ( 3)  _Iregion_4 = 0.0

       F(  3,    45) =    7.67
            Prob > F =    0.0003
```

If the parentheses were not used around `i.region`, only `_Iregion_3` would have been added to the model.

Finally, it may be that there are certain variables that are known to be important within the process, and we may want to ensure they are included in the model irrespective of their statistical significance. This can be achieved using the `lockterm1` option, which ensures that the predictor variable is retained in the model irrespective of its significance. Again, parentheses can be put around several variable names to make `sw` treat them as a single unit.

6.3.2. Multicollinearity

Strictly, multicollinearity occurs when one predictor variable can be calculated from one or more other predictors variables. For example, if the linear model is $Y = x_1 - 3x_2 + x_3$, but $x_3 = x_1 + 2x_2$, then we can replace the regression equation with $Y = 2x_1 - x_2$ to get exactly the same predicted values. Thus the β parameters are no longer uniquely defined. If multicollinear predictors are used with the `regress` command, stata will usually drop one or more predictors until the β parameters are uniquely determined. However, it may be unable to determine which variables are involved, in which case it will be unable to calculate standard errors for some of the variables.

In practice, such exact multicollinearity is rare. However, if two or more variables are highly correlated,[d] this can still cause problems with estimating the β parameters. A common consequence is that the standard errors of the β parameters becomes very large. It can also happen that the β parameters themselves take very unusual values, and may even have the opposite sign to that expected.

The only easy way around this problem is to use less correlated variables as predictors. For example, rather than using both diastolic blood pressure and systolic blood pressure as predictors (which are highly correlated), use systolic and (diastolic - systolic), which will be much less strongly correlated.

If it is not possible to remove the multicollinearity in this way, more complex solutions are available, such as ridge regression and principal components regression. However, these techniques are beyond the scope of this course.

6.3.3. Polynomial Regression

If the relationship between Y and x is non-linear, polynomial terms (i.e. x^2, x^3 etc.) can be added to the model. This enables curved lines defined by quadratic or cubic equations to be fitted to the data, which may improve the fit. Polynomial terms should be fitted until one of them is not significant (rather like a forward selection procedure) at a chosen significance level. The power of the highest polynomial is referred to as the *degree* of the polynomial model.

It should be remembered that the individual coefficients are meaningless in polynomial regression. In other words, if $Y = \beta_0 + \beta_1 x + \beta_2 x^2$, β_1 and β_2 do not have a simple interpretation. The change in Y for a change of 1 in x is no longer constant, but depends on the initial value of x. The only way to understand the association between Y and x is to draw the function $Y = \beta_0 + \beta_1 x + \beta_2 x^2$.

It should be noted that, depending on the range of values of x, there may be strong correlations between the various powers of x. This can lead to the problem of multicollinearity outlined in section 6.3.2. This can be avoided by using orthogonal polynomials, i.e. rather than fit x^2, fit $(x - c)^2$, where c is chosen so that the correlation between x and $(x - c)^2$ is 0. Orthogonal polynomials can be calculated by stata: see the command `orthpoly`.

An alternative, suggested by Altman, is to use fractional polynomials. For details of how to use fractional polynomials in stata, see the help for the command `fracpoly`. Another alternative approach is to use splines: details can be found in the help for the command `mkspline`.

6.3.4. Transformations

If the variance of Y is not constant, the only solution is to transform Y. A transformation may also be called for if the distribution of Y is not normal. Transformations may also be used on the x-variables, but this is less common: usually it is simpler to use polynomials if the association between Y and x is non-linear but there is no other reason to transform Y.

If Y has a positively skewed distribution (i.e. there are some unusually large values), it may be worth

[d] strictly speaking, if there are two linear combinations of variables that are highly correlated, for example if $x_1 + 3x_2$ is highly correlated with $x_3 + 4x_4$

transforming Y by taking logs and fitting the model

$$\log(Y) = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p$$

This transformation is commonly used with measurements of biochemical parameters, which tend to be skewed. This transformation is also appropriate if the variance of Y increases as Y increases.

The major drawback with transforming the data is that the parameters are harder to interpret. If we fit the linear model $f(Y) = \beta_0 + \beta_1 x_1$, where $f(Y)$ is some transformation of Y, then if x increases by 1, it is $f(Y)$, not Y which will increase by β_1. The amount by which Y increases will depend on its initial value, which makes it difficult to simply summarise the effect of x on Y.

For this reason, transformations other than the log transformation are largely avoided. The reason that the log transformation is used is that if $\log(Y)$ increases by β, that is equivalent to multiplying Y by e^{β}. Hence the parameters from a linear model for $\log(Y)$ are easy to interpret on the original log scale. For example, if the regression equation is $\log(Y) = 0.5x$, then for each unit increase in x, Y is multiplied by $e^{0.5} = 1.65$. In other words, when x increased by 1, Y increases by 65%.

6.3.5. Regression Through the Origin

It may be that there is an *a priori* reason to assume that $Y = 0$ when all of the predictor variables are 0. In this case, it is possible to fit a linear model in which β_0 is forced to be 0, i.e. the regression line is forced through the origin. However, this should only be used if data is available for x-values close to 0. If we have no data from near the origin, we should not be concerned with trying to predict Y near the origin: we can only be sure our linear model holds in the region in which we have collected data.

If we are certain regression through the origin is justified, we can force stata to force β_0 to be equal to zero with the option `nocons` to the `regress` command. However, it should be noted that R^2 is calculated differently when performing regression through the origin, and cannot be compared to the R^2 value obtained normally.

6.4. Further Reading

Further Reading

[1] S. Nieuwenhuis, B. U. Forstmann, and E.-J. Wagenmakers. Erroneous analyses of interactions in neuroscience: a problem of significance. *Nature Neuroscience*, 14(9):1105–1107, sep 2011.

[2] J. N. S. Matthews and D. G. Altman. Interaction 2: Compare effect sizes not P-values. *British Medical Journal*, 313:808, 1996.

[3] A. Gelman and H. Stern. The Difference Between “Significant” and “Not Significant” is not Itself Statistically Significant. *The American Statistician*, 60(4):328–331, 2006.

6.5. Practical

6.5.1. Datasets

The datasets that you will use in this practical can be accessed via http from within stata. However, the directory in which they are residing has a very long name, so you can save yourself some typing if you create a global macro for this directory. You can do this by entering

```
global basedir http://personalpages.manchester.ac.uk/staff/mark.lunt
global datadir $basedir/stats/6_LinearModels2/data
```

(In theory, the global variable `datadir` could have been set with a single command, but fitting the necessary command on the page would have been tricky. Far easier to use two separate commands as shown above). If you wish to run the practical on a computer without internet access, you would need to:

1. Obtain copies of the necessary datasets
2. Place them in a directory on your computer
3. Define the global macro `$datadir` to point to this directory.

6.5.2. Categorical Variables

Dichotomous Variables and T-Tests

Load the `auto` dataset distributed with stata using the command `sysuse auto, clear`. We are going to confirm that US-made vehicles tend to be heavier than non-US made vehicles.

5.1 Fit a linear model with the command `regress weight foreign`. Are foreign vehicles heavier or lighter than US vehicles ? Is the difference signficant ?

5.2 Fit the above model again, but this time use `regress weight i.foreign`
Does this make any difference ?

5.3 Check that `regress` gives the same results as `ttest` by entering the command
`ttest weight, by(foreign)` Look at the difference between the means and its standard error: are they the same as those obtained by `regress`

5.4 Make box-and-whisker plots for the weight of both US and non-US vehicles with the commands

```
sort foreign
graph box weight, by(foreign)
```

Does the variance of `weight` look the same for US and non-US vehicles ?

5.5 You may wish to confirm your ideas in the last question by typing `by foreign: summ weight`. Are the standard deviations similar between US and non-US vehicles ?

5.6 Finally, use `hettest` to confirm whether or not the variances are the same. Was a linear model an appropriate way of analysing this data ?

Multiple Categories and ANOVA

The dataset `$datadir/soap` gives information on the scores given to 90 bars of soap for their appearance. The scores are on a scale of 1-10, the higher the score the better. Three operators each produced 30 bars of soap. We wish to determine if the operators are all equally proficient.

5.7 Sort the data by operator (type `sort operator`), and produce box-and-whisker plots of appearance for each operator by typing `graph box appearance, by(operator)`. Which operator appears to have the highest scores ?

5.8 What are the mean scores for each of the 3 operators ? The command to use is `by operator: summ appearance`

5.9 Fit a linear model to the data with `regress appearance i.operator`. Are there significant differences between the three operators ?

5.10 What is the p-value for the differences between the operators ?

5.11 Which operator was used as the baseline for the linear model ? (Check that the constant term in the model is the same as this operator's mean score you calculated earlier)

5.12 Use `lincom` to calculate the mean score for operator 2. Is this the same as the score you calculated earlier ?

5.13 Use `lincom` to calculate the difference in mean score between operators 2 and 3. Is this difference statistically significant ?

6.5.3. Interactions and Confounding

The dataset `$datadir/cadmium` give data on the ages, vital capacities and durations of exposure to cadmium (in three categories) in 88 workers. We wish to see if exposure to cadmium has a detrimental effect on vital capacity. However, we know that vital capacity decreases with age, and that the subjects with the longest exposures will tend to be older than those with shorter exposures. Thus, age could confound the relationship between exposure to cadmium and vital capacity.

5.14 Plot a graph of vital capacity against age, to satisfy yourself that vital capacity decreases with increasing age ? (use `scatter capacity age`)

5.15 In case you are not satisfied, fit a linear model to predict vital capacity from age, with `regress capacity age`

It would be nice to be able to tell to which exposure group each point on the plot of vital capacity against age belonged. This can be done by using the commands

```
gen cap1 = capacity if exposure == 1
gen cap2 = capacity if exposure == 2
gen cap3 = capacity if exposure == 3
scatter cap1 cap2 cap3 age
```

This graph clearly shows that the group with the highest exposure tend to be older (they are towards the right of the graph).

5.16 Is there a difference between the three exposure groups in vital capacity ? (use `regress capacity i.exposure`)

5.17 We have seen that a lower vital capacity in the most exposed may be due to their age, rather than their exposure. Adjust the previous example for age, using `regress capacity age i.exposure` Now use `testparm i.exposure` to test whether there are significant differences between groups.

To get a visual idea of the meaning of the previous regression model, type

```
predict ppred, xb
gen ppred1 = ppred if exposure == 1
gen ppred2 = ppred if exposure == 2
gen ppred3 = ppred if exposure == 3
sc cap1 cap2 cap3 age || line ppred1 age || line ppred2 age || line ppred3
age
```

Note that the final command (`scatter ...`) must all be entered on one line. This will show the same plot of capacity against age we saw before, but with three parallel regression lines, one for each group.

5.18 Finally, we wish to test the hypothesis that subjects with high exposure lose their vital capacity quicker as they age, i.e. that there is an interaction between age and vital capacity. We can do this with the command `regress capacity i.exposure##c.age` followed by `testparm i.exposure#c.age` Is there a significant difference between the slopes with age in the three groups ?

To see the regression lines in this case, type

```
predict ipred, xb
gen ipred1 = ipred if exposure == 1
gen ipred2 = ipred if exposure == 2
gen ipred3 = ipred if exposure == 3
sc cap1 cap2 cap3 age || line ipred1 age || line ipred2 age || line ipred3
age
```

Again, the final command (`scatter ...`) must all be entered on one line.

5.19 From the regression output, which group has the steepest slope with age and which group the least steep ?

5.20 Use `lincom age + 3.exposure#c.age` to calculate the decrease in vital capacity per year increase in age in the highest exposure group.

6.5.4. Variable Selection

Use the dataset `$datadir/hald`. This contains data on the amount of heat evolved as cement hardens, as a function of the proportions of 4 chemicals in the composition of the cement.

5.21 Use forward selection to chose a model for predicting the amount of heat evolved. (Use `sw regress y x1 x2 x3 x4, pe(0.05)`) Which variables are included.

5.22 Now use backward elimination, using the command `sw regress y x1 x2 x3 x4, pr(0.05)` Does this select the same model ?

5.23 Choose a model using stepwise selection, with the command `sw regress y x1 x2 x3 x4, pe(0.05) pr(0.0500005)` Is this model the same as any or all of the previous models ?

5.24 Produce a correlation matrix for the x-variables using the command `corr x1 x2 x3 x4` What is the correlation between `x2` and `x4` ?

5.25 Does this help to explain why the different methods of variable selection produced different models ?

5.26 Fit all 4 predictors in a single model with `regress y x1 x2 x3 x4` Look at the F-statistic: is the fit of the model statistically significant ?

5.27 Look at the R^2 statistic: is this model good at predicting how much heat will be evolved ?

5.28 Look at the table of coefficients: how many of them are significant at the p= 0.05 level ?

6.5.5. Polynomial Regression

Use the data in `$datadir/growth`. This dataset gives the weight, in ounces, of a baby weighed weekly for the first 20 weeks of its life.

5.29 Plot the data with `scatter weight week` Does the plot appear to be linear, or is there evidence of curvature ?

5.30 Fit a straight line to the data with `regress weight week` Produce a partial residual plot with `cprplot week` Does this confirm what you thought previously ?

5.31 Generate an week^2 term with the command `gen week2 = week*week` Add this term to the regression model, with `regress weight week week2` Does this improve the fit of the model ? (i.e. is `week2` a significant predictor ?)

5.32 Generate predicted values from this model with `predict pred2, xb`. Produce a graph of the observed and predicted values with `twoway scatter weight week || line pred2 week`

5.33 Continue to generate polynomial terms and add them to the regression model until the highest order term is no longer significant. What order of polynomial is required to fit this data ?

5.34 Produce a correlation matrix for the polynomial terms with `corr week*`. What is the correlation between `week` and `week2` ?

7. Modelling Binary Outcomes

7.1. Cross-tabulation

If we are interested in the association between two binary variables, for example the presence or absence of a given disease and the presence or absence of a given exposure. Then we can simply count the number of subjects with the exposure and the disease; those with the exposure but not the disease, those without the exposure who have the disease and those without the exposure who do not have the disease. We can than put these numbers into a 2 $\times$ 2 table, as shown in Table 7.1.

	Exposed	Unexposed	Total
Cases	a	b	a + b
Controls	c	d	c + d
Total	a + c	b + d	a + b + c + d

Table 7.1.: Presentation of data in a two-by-two table

If our sample has been randomly selected from a given population, then the proportions in each cell (i.e. $a/(a+b+c+d)$ etc.) will differ from the proportions in the population only by random variation. However, there are two other widely used sampling schemes for which this is not the case. Both sampling schemes can be thought of as stratified samples, in which subjects are sampled from two different strata.

The first scheme is exposure-based sampling, in which a fixed number of exposed subjects and a fixed number of unexposed subjects are sampled. In this case, the prevalence of the disease in the exposed and unexposed subjects ($a/(a+c)$ and $b/(b+d)$) are unaffected, but the proportion of exposed subjects is fixed by the sampling scheme, and need not reflect the proportion in the population[a].

The alternative is outcome-based sampling, often referred to as a case-control study, in which we sample a fixed number of cases ($a+b$) and a fixed number of controls ($c+d$). In this case, the prevalence of the disease in our sample ($(a+b)/(a+b+c+d)$ is fixed by the design, and does not reflect the prevalence in the population.

If there is no association between the exposure and disease, we would expect the prevalence of the disease to be the same in the exposed as it is in the unexposed. Since $a+b$ subjects have the disease, the overall prevalence of disease is $\frac{a+b}{a+b+c+d}$. There are $a+c$ exposed subjects, so we would expect $\frac{(a+b)\times(a+c)}{a+b+c+d}$ subjects who are exposed and ill. This is the expected value of a, under the null hypothesis that the prevalence does not vary with the exposure, and this works for all sampling schemes.

We can calculate expected values for b, c and d in a similar fashion. If the observed values are sufficiently far from their expected values under the null hypothesis, we can conclude that the null hypothesis

[a] In fact, it generally won't, because the reason for sampling in this way is usually that the exposure is rare, and we need to artificially increase the number of exposed subjects in our sample in order to increase our power.

is unlikely to be true. This can be done by calculating $\frac{(O-E)^2}{E}$ for each cell of the table (where O is the observed value and E is the expected value) and summing them for the four cells in the table. This sum will follow a χ^2 distribution on 2 degree of freedom if the null hypothesis is true, so a P-value can be obtained as the probablity of the observing a higher value from this distribution. This hypothesis test is referred to as the χ^2-test.

To perform a χ^2-test in stata, the command to use is `tabulate`, with the `chi2` option. So if the variable `exposure` contains the exposure data and `disease` contains the disease information, the full command for a χ^2-test is

```
tabulate exposure disease, chi2
```

7.1.1. Measures of Effect

Rather than simply testing the null hypothesis that the exposure does not affect the outcome, we may wish to quantify the effect. Most commonly, we are interested in the relative risk and its confidence interval, but this is not the only possibility. For example, if we are particularly interested in the absolute risk of a particular outcome, we can use the risk difference (i.e. the difference between the prevalence in the unexposed and the prevalence in the exposed.

However, if we have outcome-based sampling, the relative risk and the risk difference are meaningless, because the overall prevalence of the disease in our sample is fixed by the design, and does not reflect the prevalence in the population. However, the odds ratio will take the same value for any of the sampling schemes, and hence it can still be used with outcome-based sampling.

In stata, the relative risk and risk difference are most easily obtained using the command

```
cs disease_var exposure_var
```

The odds ratio can be obtained from the command by adding the option `or` at the end of the command.

$$\begin{aligned}
\text{Relative Risk} &= \frac{\frac{a}{a+c}}{\frac{b}{b+d}} = \frac{a(c+d)}{c(a+b)} \\
\text{Risk Difference} &= \frac{a}{a+c} - \frac{b}{b+d} \\
\text{Odds Ratio} &= \frac{\frac{a}{c}}{\frac{b}{d}} = \frac{ad}{cb}
\end{aligned}$$

7.1.2. Limitations of Tabulation

Whilst tabulation can be useful when initially exploring data, it is not sufficient for complex analysis. This is primarily because continuous variables cannot be included in the analysis, only categorical variables. In addition, only a single variable can be analysed at a time. There are more complex tabulations that can be used to allow for more than one variable, but these also have drawbacks: the more variables are included in the model, the more "sparse" the tables become (i.e. the fewer observations in each cell of the table), and the inference from the table becomes less robust.

7.2. Modelling Approaches

7.2.1. Linear Regression and dichotomous outcomes

To understand what we are doing with logistic regression, consider an example taken from [1]. In this example, we wish to predict the probability that a person has coronary heart disease (CHD) from their age. A sccatter plot of CHD against age is, at first sight, uninformative: see Figure 7.1

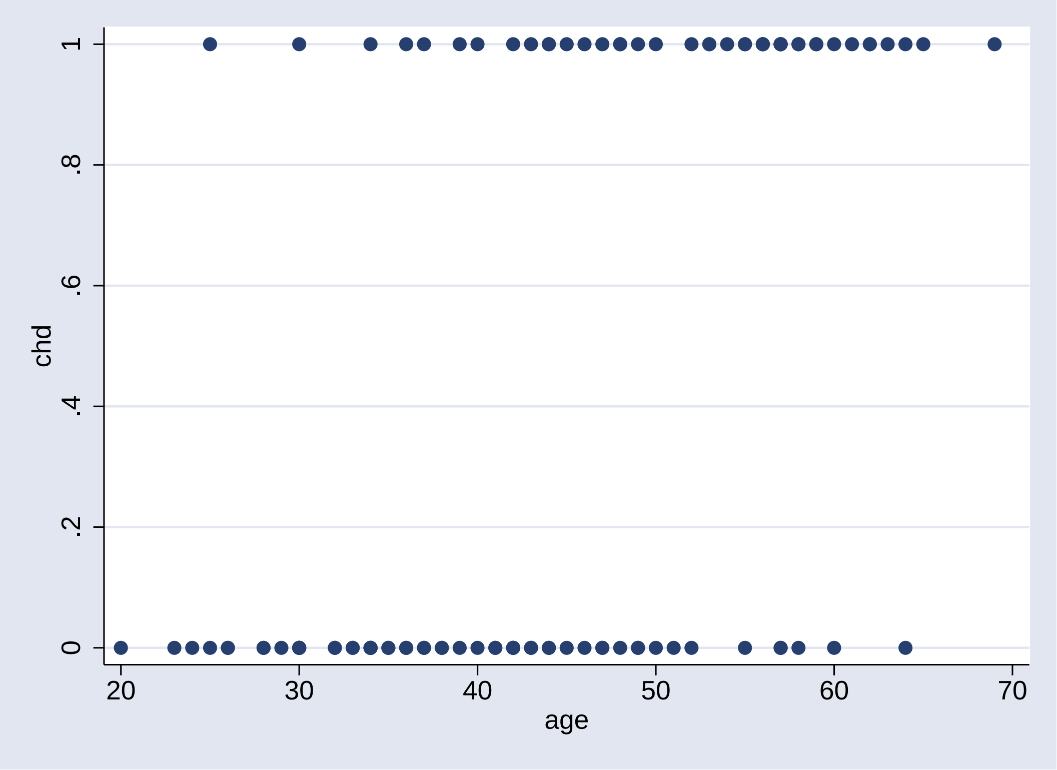


Figure 7.1.: Scatter plot of CHD against against age

This illustrates one of the problems with using a linear model for a dichotomous outcome: the outcome variable Y is clearly not normally distributed. It can only take one of two values: 0 or 1. Hence, to model this data, we will need to use a different error distribution.

However, the scatter plot does suggest that those with CHD tend to be older than those without. We can confirm this by dividing subjects up into age groups, and plotting the proportion of subjects with CHD in each group against the mean age in that group. The results are illustrated in Figure 7.2, which confirms that the probability of CHD increases as age increases.

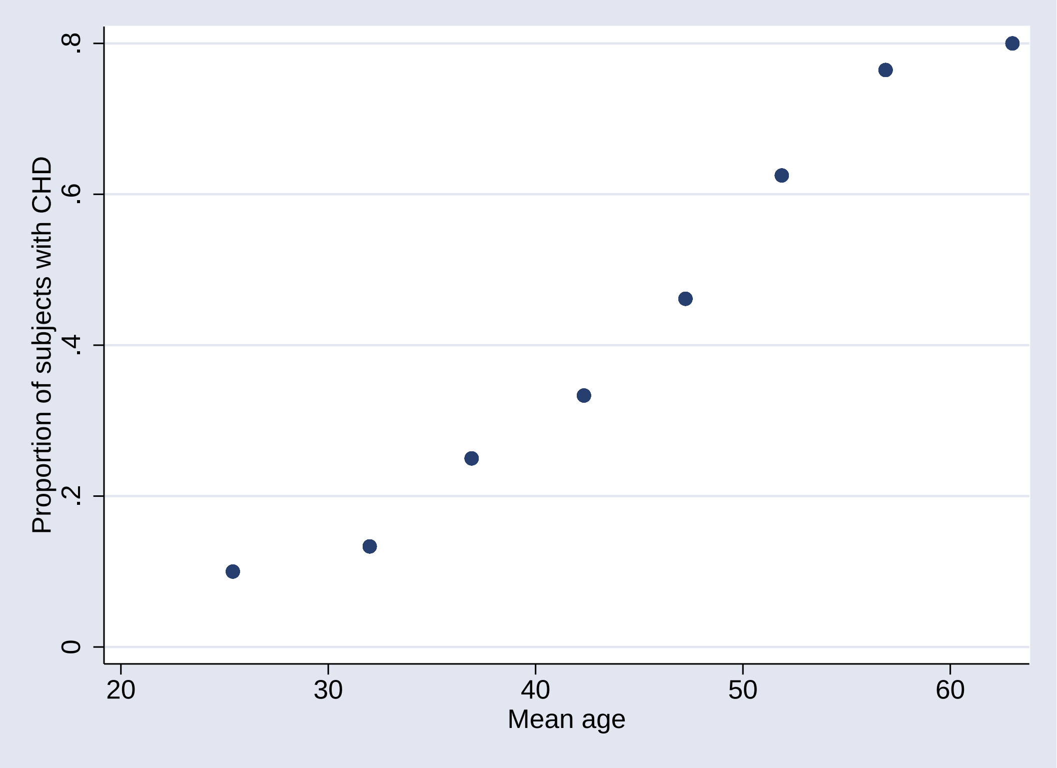


Figure 7.2.: Scatter plot of proportion of CHD against against mean age

Notice that although the observed values of Y are either 0 or 1, the predicted outcome $(\hat{Y})$ can take any value between 0 and 1. For example, there are 10 subjects between 20 and 30, of whom 1 had CHD. Therefore, the proportion of subjects with CHD is 0.1, although there is nobody in the sample with a value of 0.1 for the CHD variable. $\hat{Y}$ can also be thought of as the mean of Y in the subgroup: since 1 subject has $Y = 1$ and 9 have $Y = 0$, the mean of $Y = (9 \times 0 + 1 \times 1)/10 = 0.1$.

Notice also that the estimates for each age group do not lie on a straight line: the line becomes flatter as the proportion of subjects with CHD approaches 0 or 1. This is common when there is a strong association between a continuous risk factor and a dichotomous outcome: the strong association means that the slope is steep when the risk is around 0.5, but it has to flatten out as it approaches 0 and 1 so that it does not exceed these values. If we were to fit a straight line to this data, the predicted probability of CHD would be less than 0 for the very youngest subjects in this dataset, as shown in Figure 7.3.

Clearly, we cannot use the linear regression model for this data, since this would give predicted values ranging from $-\infty$ to ∞, and even within the age range we are considering it would produce meaningless predicted values. In order to find a suitable model, we need to consider the relationship between probabilities and odds.

7.2.2. Probabilities and Odds

A probability is a number between 0 and 1 (inclusive): 0 means the event in question never happens, 1 means it always happens, and 0.5 means it happens half of the time. Another scale that is useful

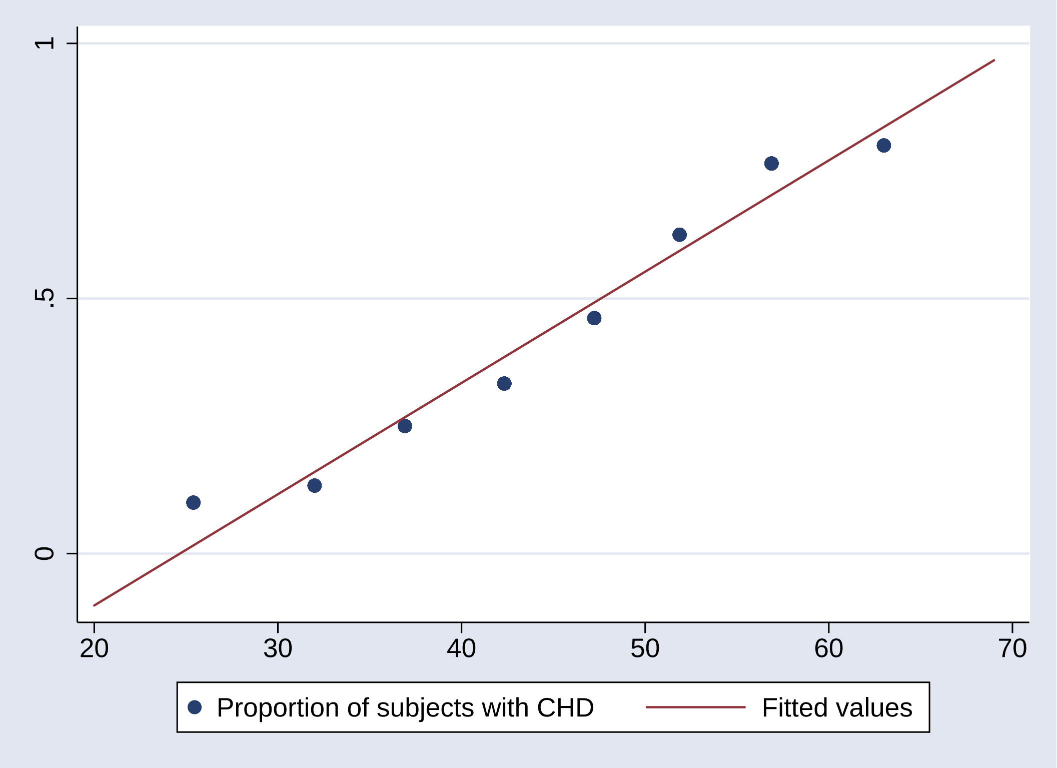


Figure 7.3.: Scatter plot of proportion of CHD against against mean age with overlaid linear model

for measuring probabilities is the odds scale, familiar to those who like betting on the horses. If the probability of an event occuring in p, then the odds (Ω) of it occuring are $p : 1 - p$, which is often written as a fraction $\Omega = p/(1 - p)$. Hence if the probability of an event is 0.5, the odds are 1:1, whilst if the probability is 0.1, the odds are 1:9.

Why is the odds scale useful ? Primarily because it can take any value from 0 to ∞. Both p and $1 - p$ have to be positive, so $p/(1 - p)$ must be positive. As $p \to 0$, $p/(1 - p) \to 0$, whilst as $p \to 1$, $1 - p$ gets extremely small so $p/(1 - p)$ gets extremely large: for example, if $p = 0.99, \Omega = 0.99/0.01 = 99$, whilst if $p = 0.999, \Omega = 0.999/0.001 = 999$.

So, if p ranges from 0 to 1, $p/(1 - p)$ ranges from 0 to ∞. If we now take logs, then $\log\left(p/(1 - p)\right)$ ranges from $-\infty$ to ∞. Hence, we can model

$$g(p) = \log\left(\frac{p}{(1 - p)}\right) \tag{7.1}$$

rather than p: as $g(p)$ goes from $-\infty$ to ∞, p goes from 0 to 1. Figure 7.4 shows the relationship between p and $g(p)$: you will see that the shape of this curve is very similar to that in Figure 7.2, suggesting that using this link would give a good fit to that data. It is also symmetric, since if the odds of an event happening are Ω, the odds of it not happening are $1/\Omega$.

It is important to remember that there is a one-to-one correspondance between odds and probabilities: if you know the probability (p) of an event occurring, you can calculate the odds ($\Omega = p/(1 - p)$), and if you know the odds you can calculated the probability ($p = \Omega/(1 + \Omega)$). So the odds scale is just a different scale that can be used to measure probabilities.

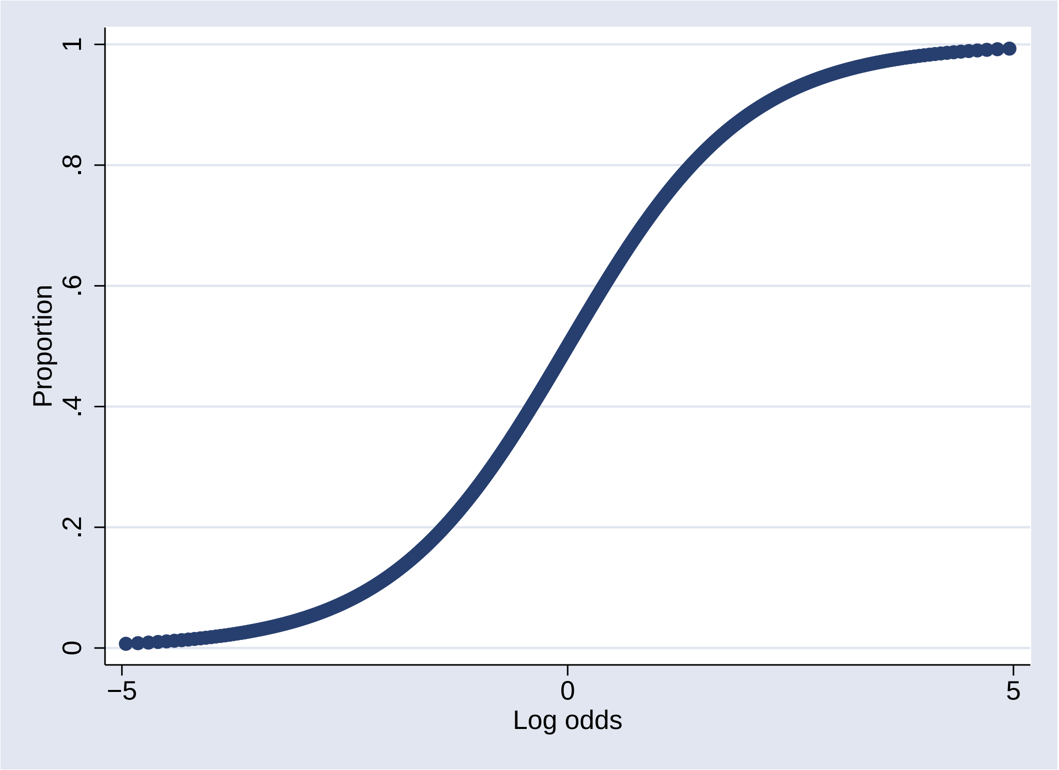


Figure 7.4.: Relationship between proportion and log-odds

7.2.3. The Binomial Distribution

Suppose you toss a coin 10 times: what is the probability that you get a) 10 heads, b) 5 heads, c) no heads. Intuitively, you think that getting 5 heads should happen more often than getting 10 or none. Getting 10 heads would be seen as being unusual, but not impossible. However, getting 4 or 6 heads would be seen as less unusual.

The binomial distribution can be used to determine how unusual getting 10 heads would be in this case. It requires two parameters: the probability of a success (in this case 0.5, assuming that the coin we use is fair), and the number of trials that we carry out (in this case 10). The number of heads we get has to be a whole number from 0 to 10 inclusive, and from the binomial distribution we can calculate how like each of these outcomes is.

The binomial distribution is appropriate whenever:

1. the outcome we are interested in is dichotomous (we can think of it as a success or a failure); and

2. we are considering a number of independent trials.

Therefore, the binomial distribution is appropriate to use as an error distribution in logistic regression.

7.2.4. The Logistic Regression Model

So, the logistic regression model is

$$\begin{aligned}\log\left(\frac{\hat{\pi}}{(1-\hat{\pi})}\right) &= \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p \\ Y &\sim \text{Binomial}(\hat{\pi})\end{aligned}$$

where $\hat{\pi}$ is the predicted probability that $Y = 1$, given the values of $x_1 \ldots x_p$.

Parameter Interpretation

In a simple linear model $Y = \beta_0 + \beta_1 x$, if x increases by 1, Y increases by β_1. In a logistic regression model, it is $\log(\hat{\pi}/(1-\hat{\pi})$ which increases by β_1. What does this mean in the real world ?

Suppose that the predicted probability of the event of interest is p_0 when $x = 0$ and p_1 when $x = 1$, Then we have

$$\log\left(\frac{\hat{p_0}}{1-\hat{p_0}}\right) = \beta_0$$

and

$$\log\left(\frac{\hat{p_1}}{1-\hat{p_1}}\right) = \beta_0 + \beta_1$$

So

$$\log\left(\frac{\hat{p_1}}{1-\hat{p_1}}\right) = \log\left(\frac{\hat{p_0}}{1-\hat{p_0}}\right) + \beta_1$$

If we exponentiate both sides of this equation we get

$$e^{\log\left(\frac{\hat{p_1}}{1-\hat{p_1}}\right)} = e^{\log\left(\frac{\hat{p_0}}{1-\hat{p_0}}\right)+\beta_1}$$

which simplifies to

$$\frac{\hat{p_1}}{1-\hat{p_1}} = \frac{\hat{p_0}}{1-\hat{p_0}} \times e^{\beta_1} \tag{7.2}$$

In other words, when x increases by 1, the odds of a positive outcome increase by a factor of e^{β}_1. Hence e^{β}_1 is called the *odds ratio for a unit increase in* x.

The interpretation of β_0 is slightly different, since there is no variable associated with this coefficient. However, we can say that if $x = 0$, $\log\left(\frac{\hat{\pi}}{(1-\hat{\pi})}\right) = \beta_0$, so $\frac{\hat{\pi}}{(1-\hat{\pi})} = e^{\beta_0}$. In other words, β_0 is the log of the odds of a positive outcome when all of the predictor variables take the value 0 (it is analogous to the intercept in a linear model, which is the value taken by Y when all the predictor variables are 0).

Odds Ratios and Relative risks If p is small, then $\Omega \approx p$, since $1 - p \approx 1$. Thus, odds are very similar to probabilities provided that p is small, and odds ratios are then very similar to relative risks. However, if the outcome of interest is more common, then the difference is greater. Figure 7.5shows a plot of Ω against p, showing clearly how similar they are for small p but how they diverge increasingly as p increases.

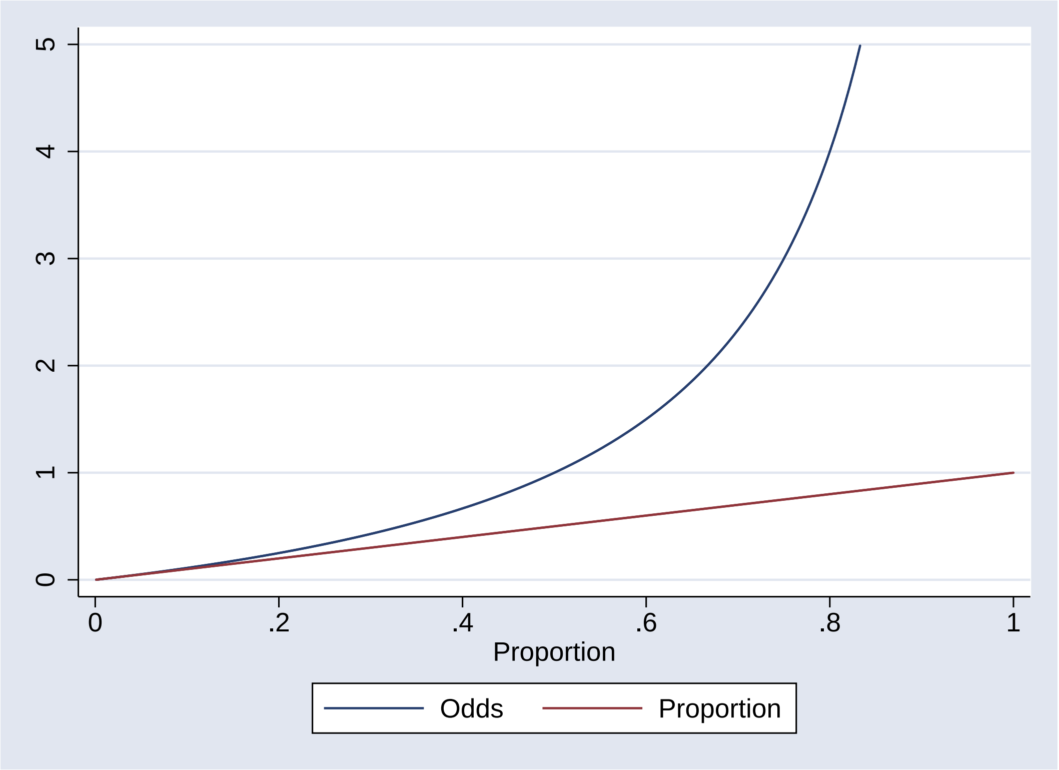


Figure 7.5.: Comparison of Odds and Probability

7.2.5. Logistic Regression in Stata

There are two commands for performing logistic regression in stata, `logistic` and `logit`. They are almost identical: the only difference is that by default, `logistic` produces a table of odds ratios whilst `logit` produces a table of coefficients. However, either can be used, and there are options to get coefficients from `logistic` and odds ratios from `logit`.

The basic syntax for both commands is the same:

`logistic` *depvar* [*varlist*]

where *`depvar`* is the outcome variable and *`varlist`* are the predictors. Stata assumes that *depvar* takes the value 0 if the event of interest did not take place and 1 if it did[b]. The variables in *varlist* can be categorical (if you use the construction `logistic depvar i.indepvar`), continuous or a mixture of the two.

Here is an example of performing logistic regression using stata. We will look at the same example as at the beginning of this chapter, using age to predict the presence of coronary heart disease, with the data taken from [1]. The results of running the command `logistic chd age` are given below:

[b] In fact, it will take any value not equal to 0 or missing as meaning the event took place, but that is not usually a sensible way of coding your data

```
. logistic chd age
Logistic regression                               Number of obs   =        100
                                                  LR chi2(1)      =      29.31
                                                  Prob > chi2     =     0.0000
Log likelihood = -53.676546                       Pseudo R2       =     0.2145
------------------------------------------------------------------------------
         chd | Odds Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         age |   1.117307   .0268822     4.61   0.000     1.065842    1.171257
------------------------------------------------------------------------------
```

This is very similar to the output from linear regression wich you are familiar with. There is no ANOVA table, since that is not meaningful in logistic regression. There is an overall model likelihood ratio χ^2-statistic (29.31) and its corresponding p-value at the top. R^2 cannot be calculated for a logistic regression model, but there are a number of surrogates that can be used, one of which is given by stata as the "Pseudo R2". More details can be found in section 7.3.2

Then there is a table of odds ratios. In this case, it only contains a single entry, since we only have a single predictor variable. The odds ratio is given as 1.12, with a 95% confidence interval of (1.07, 1.17). This means that for each year increase in age, the odds of CHD increase by a factor of 1.12. There is also a z-statistic for testing the null hypothesis that the odds ratio is 1, and the result of that significance test ($p = 0.000$, the effect of age is highly significant).

If we were to use the `coef` option of the logistic command, we would get the coefficients of the logistic regression model rather than odds ratios:

```
. logistic chd age, coef
Logistic regression                               Number of obs   =        100
                                                  LR chi2(1)      =      29.31
                                                  Prob > chi2     =     0.0000
Log likelihood = -53.676546                       Pseudo R2       =     0.2145
------------------------------------------------------------------------------
         chd |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         age |   .1109211   .0240598     4.61   0.000     .0637647    .1580776
       _cons |  -5.309453   1.133655    -4.68   0.000    -7.531376   -3.087531
------------------------------------------------------------------------------
```

The top part of the output is identical: only the table of coefficients is different. We now have two coefficients: `age` and `_cons`. The reason that we did not have `_cons` in the previous is example is that this coefficient corresponds to β_0, and, as we have seen, e^{β_0} is not an odds ratio, it is simply the odds of having CHD when age is 0, and it was therefore not appropriate to put it in a column labelled "Odds Ratios". (As with linear regression, the value of β_0 is often not of interest, since it lies outside the range of the data, and corresponds to a situation in which we are not interested (in this case, the prevalence of CHD in newborn babies)).

Using `predict` after `logistic`

Having fitted our logistic regression model, we can use the `predict` command to obtain additional variables in our dataset. There are a variety of diagnostic statistics which can be obtained: we will meet some of them in Section 7.3.3. For now, we will only consider two options: the predicted probability and the linear predictor.

The predicted probability can be generated using the command `predict it varname, p`. Lets do this with the CHD data. If we enter the commands

```
logistic chd age
predict pred, p
graph twoway scatter pred age
```

we will get a scatter plot of the predicted probability of CHD at each age, which should look something like Figure 7.6 (I have overlaid the proportion of subjects in each age-band, as shown in figure 7.2)

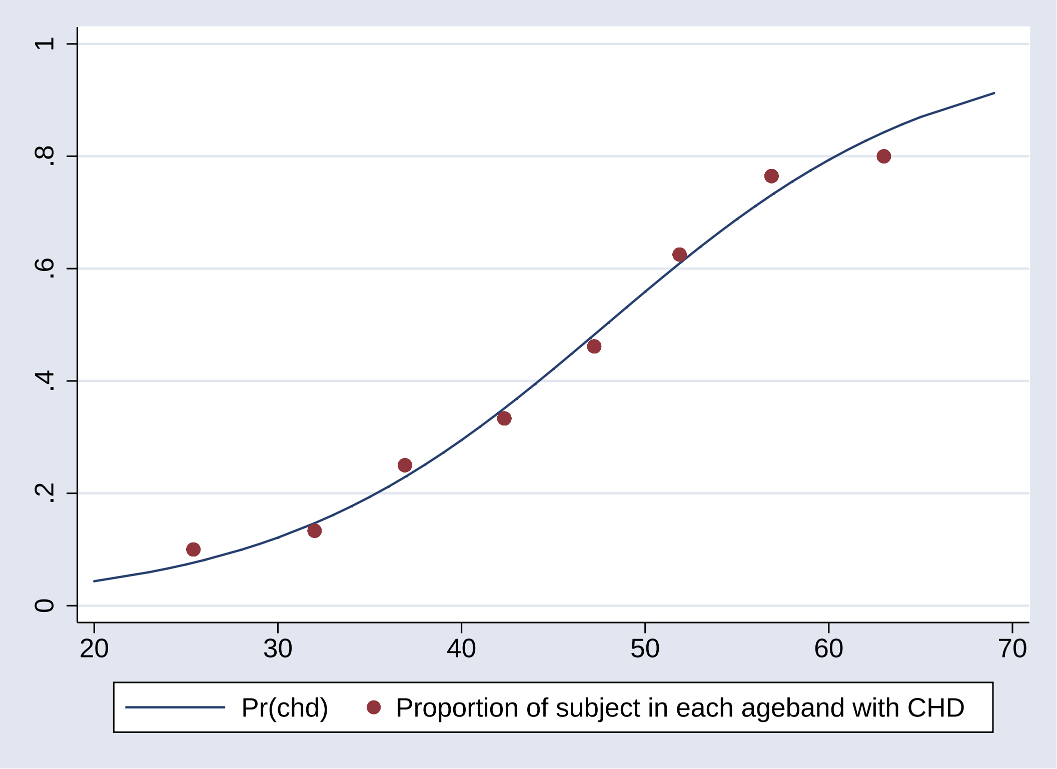


Figure 7.6.: Scatter plot of predicted probability of CHD against age

The change in the probability of having CHD as age increases is clearly non-linear: it increases slowly at young ages, more rapidly arounnd ages 40–50 and slowly again thereafter. This echoes what we see when we look at the prevalence by ageband.

However, this non-linearity is a consequence of the link function that we have chosen. By using the logit link, we are assuming that the log of the odds of CHD *does* increase linearly with age. We can see that if we obtain the linear predictor from the predict command using the option `xb`, and plot that against age, as seen in Figure 7.7:

```
predict lp, xb
graph twoway scatter lp age
```

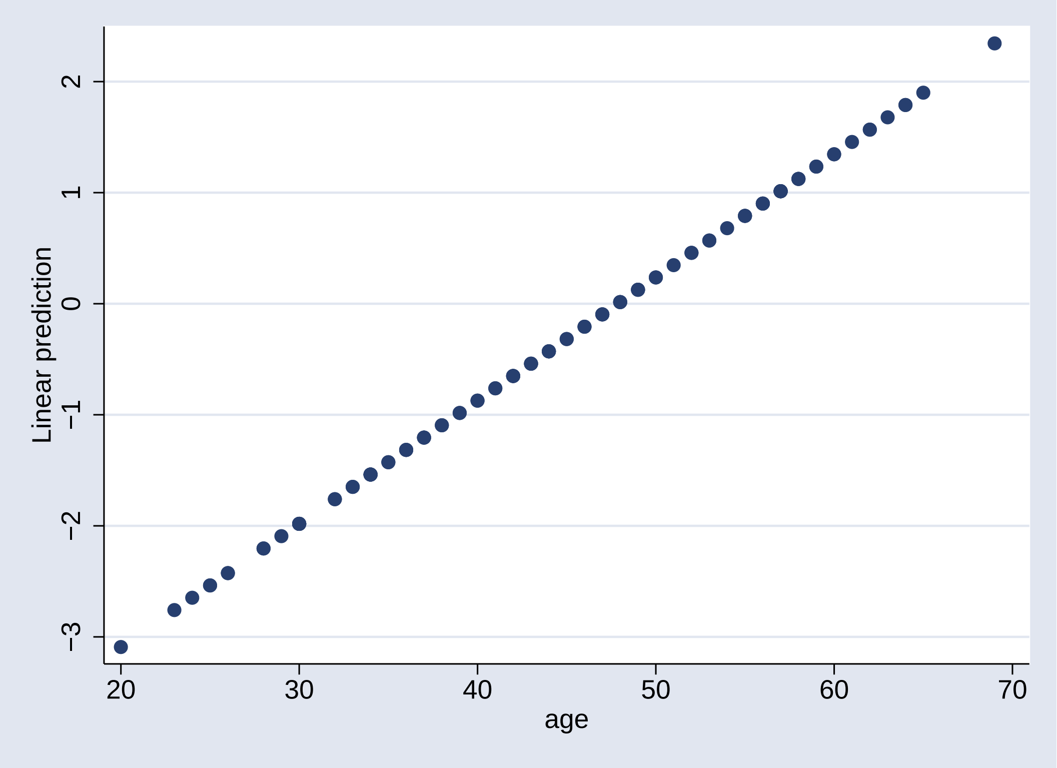


Figure 7.7.: Scatter plot of linear predictor from logistic regression model for CHD against age

7.2.6. Other Possible Models for Proportions

There are many advantages to using the logit link for modelling proportions, but it is not the only possible link function.

7.2.7. Log-binomial

One perceived drawback of the logit link is that the model is linear in the log of the odds of the outcome, and the model coefficients can be transformed into odds ratios. People are more familiar with using probabilities than odds, and would prefer to have relative risks than odds ratios. Unfortunately, the logit link can only provide odds ratios (although, as we have seen, these are numerically close to relative risks if the prevalence is low).

In order to obtain relative risks, we need to use a log link, i.e. model

$$\log \hat{\pi} = \beta_0 + \beta_1 x$$

When x increases by one, we add β to $\log \hat{\pi}$, which is equivalent to multiplying $\hat{\pi}$ by e^{β}.

This can be done, and such a model, with a log link and and a binomial error distribution, is called a log-binomial model. Such models are gaining in popularity in epidemiology, but there are a number of concerns which should be borne in mind:

1. If $\log \hat{\pi}$ can take any value from $-\infty$ to ∞, $\hat{\pi}$ can take any value from 0 to ∞. So a log-binomial model can produce predicted values that are greater than 1 (but not less than 0)
2. A logistic regression model assumes that the probability of a positive outcome increases linearly on a logit scale, whilst a log-binomial model assumes a linear increase on a log scale. If all of the predicted probabilities are small, then there will be little difference between the logistic regression model and the log-binomial model. However, if the outcome is common, the can be a considerable difference in the predicted probabilities from the two models, and it will be an empirical decision which model fits better.
3. The log-binomial model is not symmetric. That is, if q is the probability that the event in question does not take place, and we estimate q using a log-binomial model, it is not true that $\hat{q} = 1 - \hat{\pi}$, which would be the case if we used logistic regression. If the outcome in question is common, it may be necessary to model q, since modelling p could produce predicted probabilities greater than 1.

The `glm` command can be used in stata to fit a log-binomial model (in fact, this command can be used to fit any Generalised Linear Model, hence the name). The stata command, and resulting output, for fitting this model are shown below

```
. glm chd age, link(log) fam(binom) eform
Iteration 0:   log likelihood = -101.27916
Iteration 1:   log likelihood = -57.187516
  (output omitted)
Iteration 50:  log likelihood = -54.740594  (backed up)
convergence not achieved
Generalized linear models                          No. of obs      =       100
Optimization     : ML: Newton-Raphson              Residual df     =        98
                                                   Scale parameter =         1
Deviance         =  109.4811885                    (1/df) Deviance =  1.117155
Pearson          =  95.42977676                    (1/df) Pearson  =  .9737732
Variance function: V(u) = u*(1-u)                  [Bernoulli]
Link function    : g(u) = ln(u)                    [Log]
Standard errors  : OIM
Log likelihood   = -54.74059424                    AIC             =  1.134812
BIC              = -341.8254897
------------------------------------------------------------------------------
         chd | Risk Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         age |   1.047654     .00875     5.57   0.000     1.030644    1.064945
------------------------------------------------------------------------------
Warning: parameter estimates produce inadmissible mean estimates in one or
         more observations.
```

This produces a relative risk estimate of 1.05 per year increase in age, with a 95% confidence interval (1.03, 1.06). Notice the warning at the bottom of the output that some predicted values are inadmissable. This means that we have predicted probabilities greater than 1. We can see this by plotting the predicted probabilities against age, as seen in Figure 7.8

In this case, we clearly get a better fit from the logistic regression model: the assumption made by the log-binomial model that the probability of CHD increases exponentially with age is not borne out by the data. However, this needs to be checked for each individual model.

In particular, the problem in this example arises from the fact that age is a continuous variable. If all

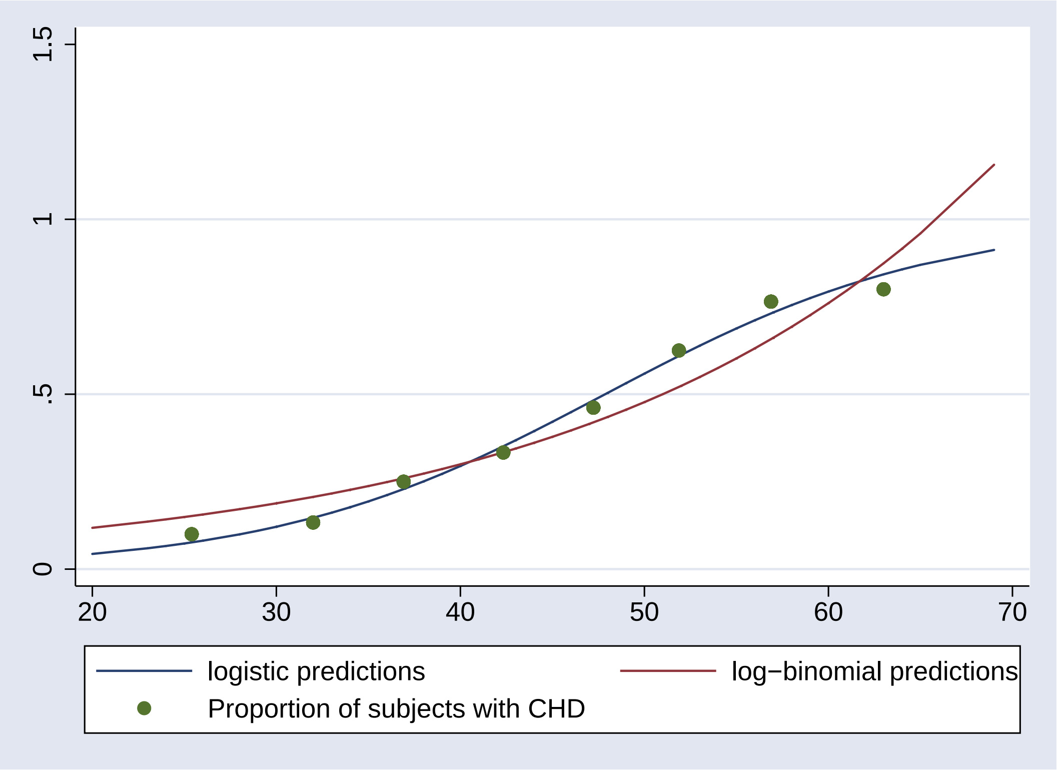


Figure 7.8.: Comparison of predictions from logistic regression and log-binomial models

of the predictors are categorical, the issue of the choice of link is far less important (since all variables can only take the values either 0 or 1), and the difference between the logistic and log-binomial models will be less.

7.2.8. Other Link Functions

The log link function can be used because it produces coefficients that are more easily interpretable than the odds ratios that are produced by logistic regression. There are also other link functions that have been used historically, such as the probit function and the complementary log-log link. Both of these link functions produce predicted values in the range 0-1, but the coefficients from these models do not have a straightforward interpretation, and they are therefore not as useful as the logistic regression models. The link functions that can be used in `glm` can be found using the command `help glm`.

7.3. Logistic Regression Diagnostics

Having produced a logistic regression model, we need to check how good it is. There are two components to this: how well it fits the data overall (discussed in Section 7.3.2) and whether there are any individual observations with either a poor fit or an undue influence on the overall fit of the model (discussed in Section 7.3.3).

7.3.1. Discrimination and Calibration

Two terms often used when talking about how good a logistic regression model is are *calibration* and *discrimination*. Discrimination refers to how well the model distinguishes between subjects at high and low risk of the outcome. Calibration refers to how closely the predicted probabilities are to the true probabilities. It is possible to have good discrimination without good calibration: if a model developed in a population with high prevalence is applied to a population with low prevalence, the calibration will be poor: the predicted probabilities will be higher than the observed probabilities. However, the discrimination may still be as good: subjects with high predicted probabilities may be more likely to have the event than those with low probabilities, even if the actual predicted probabilities are all too high.

7.3.2. Goodness of Fit

R^2

Whilst R^2 is a very valuable measure of how well a linear regression model fits, it is far less useful with logistic regression. For one thing, there are a number of different ways in which it can be calculated, all of which give the same answer for linear regression but different answers for logistic regression. A comparison of 12 different statistics was made by Mittlböck and Schemper [2], who recommended using either the square of the Pearson correlation coefficient between the observed outcomes and the predicted probabilities, or an alternative measure based on the sum of the squares of the differences between the observed outcomes and the predicted probabilities. The pseudo-R^2 produced by stata is based on the log-likelihood, and was not recommended as it lacks an easily understood interpretation.

It should also be noted that R^2 values tend to be very low for logistic regression models, much lower than for linear regression models. This is because we are trying to predict the outcome, whereas the model only gives us the *probability* of the outcome. So, for example, suppose we have two groups of subjects, a high risk group and a low risk group, and we know which group each subject belongs to. If the prevalence is 0.45 in the low risk group and 0.55 in the high risk group, then the pseudo-R^2 value produced by stata for a logistic regression of group predicting outcome is less than 0.01, despite identifying the high risk and low risk groups perfectly. If the prevalences were 0.1 and 0.9 respectively, the pseudo-R^2 value would be 0.53. Hence, R^2 is a poor choice for an overall assessment of the fit of the model.

Hosmer-Lemeshow test

A better way of assessing the fit of a logistic regression model is compare the expected and observed numbers of positives for different subgroups of the data. If the observed and expected numbers are sufficiently close, then we can assume that we have an adequate model. This test assesses calibration.

How do we select these subgroups ? If all the predictor variables are categorical, then there are only a limited number of covariate patterns that are possible. For example, if we considered age in 5 age-bands and sex as our only predictors, then there are 10 possible covariate patterns, no matter how many subjects in our dataset. We could therefore compare the observed and expected numbers of positives in each of these 10 subgroups.

However, if we treated age as continous variable, it is quite possible that there are no two subjects in the dataset with exactly the same age, and therefore there are as many different covariate patterns as there are subjects. We therefore need another way to group the data.

One suggestion, now widely used, was made by Hosmer & Lemeshow [3]. They rank the subjects according to their predicted probability of a positive outcome, and then divide them into a number of equally sized groups. A χ^2-statistic can be calculated from the expected and observed numbers of positive outcomes in each group. The number of groups to use is arbitrary, but 10 is common.

If this statistic is unusually large, then the differences between the observed and expected values are greater than we would expect by chance, and our model is not adequate. This may be becasue we have missed out an important predictor variable, misspecified the association between one or more predictors and the outcome (e.g. assumed a linear association between the predictor and the logit of the outcome when this is not in fact the case) or omitted one or more important interactions between the variables in the model. We would have to go back to our original logistic regression model and see if changin the model improved the fit to an acceptable level.

This Hosmer-Lemeshow test is implemented in stata using the command `estat gof`. Without any options, this treats each covariate pattern as a distinct group, which is often not useful if there are continuous predictors. The option `group` enables you to choose the number of groups yourself. The results of applying this test to the CHD data is given below:

```
. estat gof, group(10)
Logistic model for chd, goodness-of-fit test
  (Table collapsed on quantiles of estimated probabilities)
       number of observations =       100
             number of groups =        10
      Hosmer-Lemeshow chi2(8) =         2.22
                  Prob > chi2 =         0.9734
```

In the above example, the fit of the model would be considered to be adequate, since a χ^2 value of 2.22 on 8 degrees of freedom is not large: $p = 0.97$. So our model fits the data well.

ROC Curves

Another intuitively appealing way to assess the fit of a logistic regression model is to see what proportion of true positives it classifies as being positive (the *sensitivity*) and what proportion of true negatives it classifies as being negative (the *specificity*). Unfortunately, the output of a logistic regression is not a classification as positive or negative, but a predicted probability of being positive or negative.

Of course, we can choose one particular probability as our threshold, and count how many positives and negatives are above and below the threshold, but the choice of threshold will always be arbitrary. The stata command `estat classification` will produce sensitivities and specificities for you. By default, it uses a probability of 0.5 as the threshold, but this can be changed with the option `cutoff`.

A better idea is to measure the sensitivity and specificity at *every* possible threshold. We can then plot the sensitivity and specificity against the chosen threshold, as shown in Figure 7.9 (which was produced using the stata command `lsens`.

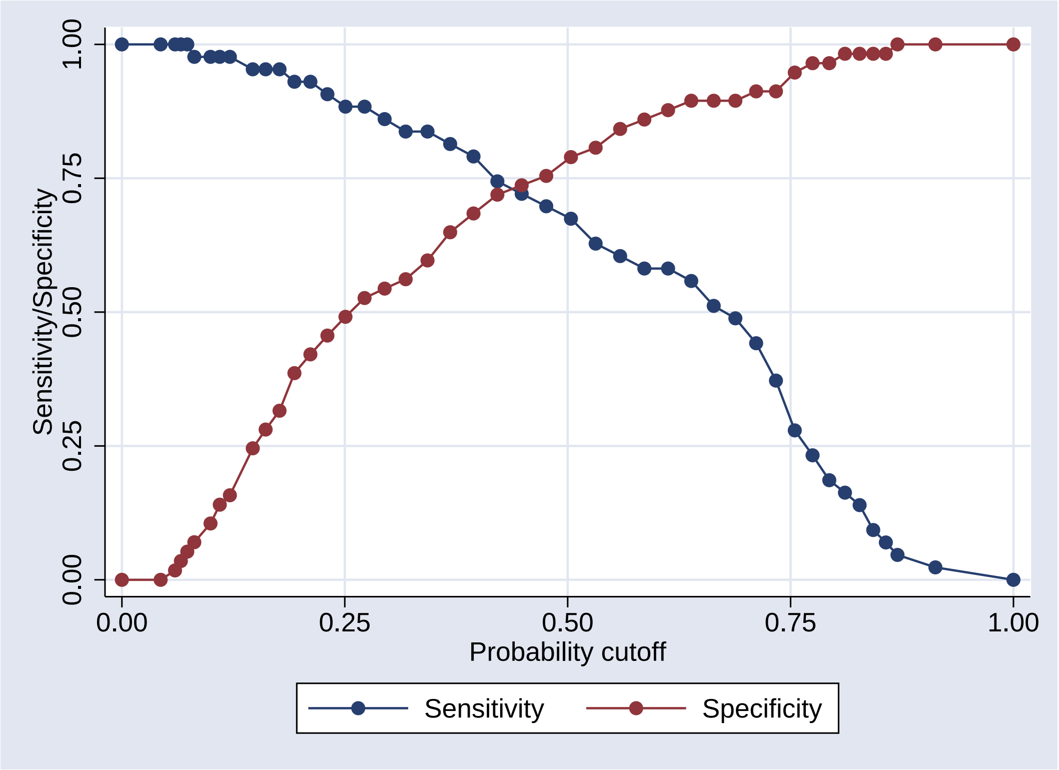


Figure 7.9.: Sensitivity and specificity by predicted probability

If we say that everybody with a probability more than 0.1 counts as a positive, we will have a very sensitive test, but it will not be very specific. As the threshold increases, the sensitivity decreases and the specificity increases, until nobody at all is classed as positive (sensitivity = 0%, specificity = 100%).

However, this is not a very efficient summary of the fit of the model. We can improve matters by plotting the sensitivity against (1 - the specificity) to give a receiver operating characteristic (ROC) curve as shown in Figure 7.10 (which was produced using the stata command `lroc`.

The bottom left corner of this graph corresponds to the right hand side of the previous graph (sensitivity is 0%, specificity 100%), whilst the top right corner corresponds to the left hand side (sensitivity 100%, specificity 0%). Ideally, we would like to reach the top left corner, since there sensitivity is 100% and specificity is 100%. The closer we approach this point, the better the model.

The results of the goodness of fit can therefore be summed up in a single number: the area under the ROC curve (often abbreviated to AUC). If this area reaches 1, then the curve must pass through the top left corner, and the model is perfect. It can be shown that the area under the curve is the probability that, if one one positive subject and one negative subject are selected at random, the positive subject has a higher predicted probability than the negative subject. It is thus a measure of discrimination. The diagonal line represents an area under the curve of 0.5, which is the discrimination that you would expect if you tossed a coin to identify positive subjects, rather than use a logistic regression model.

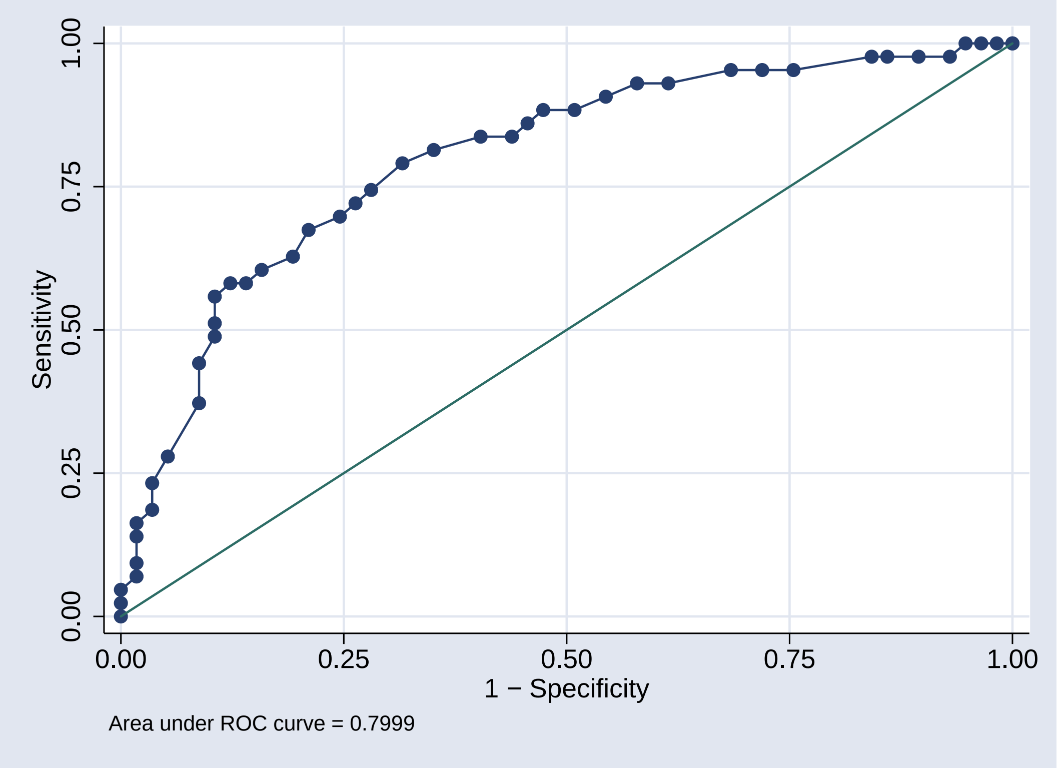


Figure 7.10.: ROC curve

7.3.3. Assessing Fit of Individual Points

The residuals from a linear model are vital in determining whether or not the data satisfies the assumptions of a linear model. However, if the logistic regression model is based on data from individual subjects (which is what we have considered so far), the residuals from a logistic regression model are far less useful. This is because the outcome variable can only take the values 0 and 1, so the residual can only take the values $1 - \hat{\pi}$ or $-\hat{\pi}$.

However, there are some diagnostics available. We can assess how great an influence an individual point has on the coefficients of the logistic regression model and on any lack of overall fit in the model. Any points with unduly high influence should be investigated to ensure that the data has been recorded correctly. The effect of removing the influential point on the model should also be investigated, and if necessary the results presented both with and without the influential point(s).

The concept of *leverage* in a linear model does not translate directly to a logistic model. However, there is a quantity $\Delta\hat{\boldsymbol{\beta}}_i$ which measures the amount that the logistic regression model parameters change when the i^{th} observation is omitted from the model. This can be obtained from stata using the `dbeta` option to the `predict` command after a logistic regression model has been fitted. A plot of $\Delta\hat{\boldsymbol{\beta}}_i$ against $\hat{\pi}$ for each observation (or each covariate pattern) will reveal observations which have a large effect on the regression parameters. It is not possible to give a numerical value to determine which points are influential: you need to identify outliers by eye.

As an example, Figure 7.11 is a plot of $\Delta\hat{\boldsymbol{\beta}}_i$ against $\hat{\pi}$ for each observation in the CHD dataset.

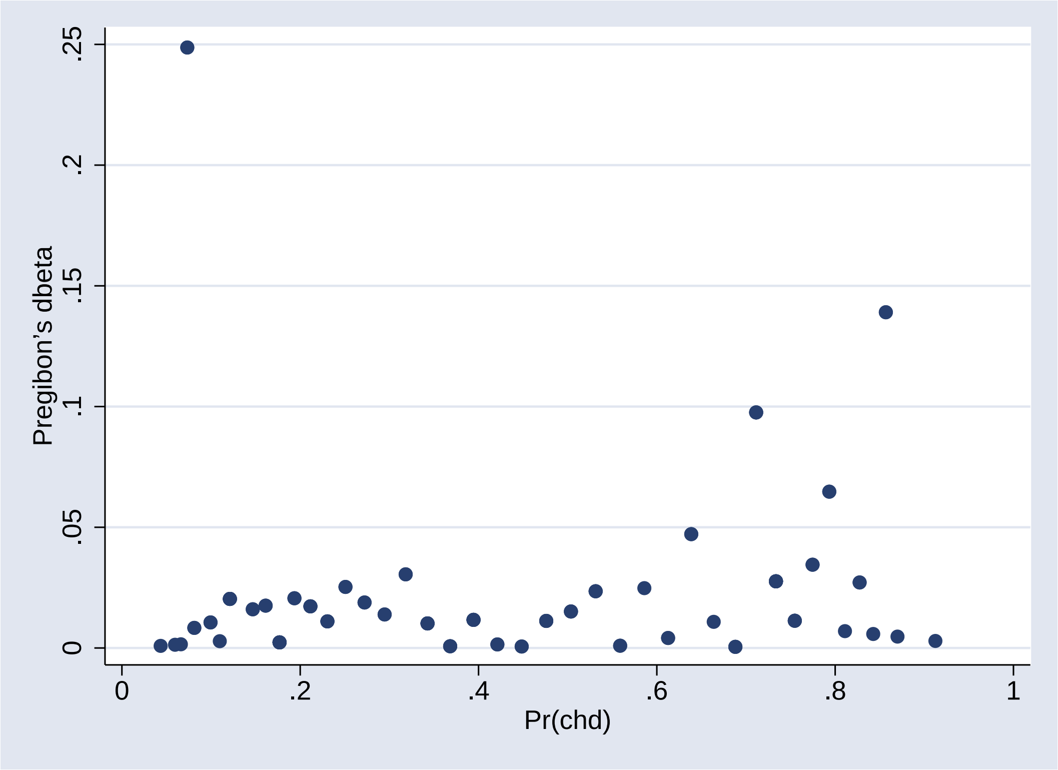


Figure 7.11.: Plot of $\hat{\boldsymbol{\beta}}_{\boldsymbol{i}}$ against $\hat{\pi}$

You will notice that there is one point which has far more influence than the others. This corresponds to subjects aged 25, 1 of whom had CHD and one did not. This gives an observed prevalence of 0.5, compared to a predicted prevalence of 0.07, which will tend to reduce the slope of the logistic regression model by pulling the regression line upwards. However, excluding these two subjects has very little effect on the logistic regression model: the odds ratio increased from 1.12 per year to 1.13 per year.

```
. logistic chd age if dbeta < 0.2
Logistic regression                                 Number of obs   =         98
                                                    LR chi2(1)      =      32.12
                                                    Prob > chi2     =     0.0000
Log likelihood = -50.863658                         Pseudo R2       =     0.2400
------------------------------------------------------------------------------
         chd | Odds Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         age |   1.130329   .0293066     4.73   0.000     1.074324    1.189254
------------------------------------------------------------------------------
```

In addition, we can use residuals to identify individual points for which the model fits poorly. Stata provides two different kinds of residuals for this: ΔX_i^2, obtained by using the `dx2` option to `predict`; and ΔD_i, obtained by using the `ddeviance` option. Again, these statistics can be plotted against $\hat{\pi}$, and outlying observations identified by eye. Such observations do not need to be omitted from the analysis, since they are not having an undue influence on the regression parameters, but it may be important to point out areas in which the predictions from the model are poor.

7.3.4. Problems of separation

It can happen that one of the predictor variables predicts the outcome perfectly. For example, suppose that we are interested in the effect of gender on our outcome, but we only have a single woman in our sample. The odds ratio for women compared to men will be either 0 (if the one woman does not have the outcome of interest), or ∞ if she does. Unfortunately, odds of 0 and ∞ correspond to a linear predictor of $-\infty$ or ∞, and stata cannot handle infinite numbers (that should not be taken as criticism, nor can any other stats package, or indeed computer program). It will therefore report that “gender determines outcome exactly” and drop the woman from the regression model.

This problem arises when there is a particular combination of indicator variables for which there are no cases (or every subject is a case). The problem is most common when modelling interactions between predictors with several categories, since this involves dividing subjects into a large number of subgroups. The only solution is to collapse categories together until there is at least one case and one control for each combination of the indicator variables. If one of the categorical variables can be fitted as a continuous variable, this might also help.

7.4. References and Further Reading

Further Reading

[1] D. W. Hosmer and S. Lemeshow. *Applied Logistic Regression.* John Wiley & Sons, Inc., 2nd edition, 2000.

[2] M. Mittlböck and M. Schemper. Explained variation for logistic regression. *Statistics in Medicine*, 15(19):1987–1997, 1996.

[3] D. W. Hosmer, S. Lemesbow, S. Lemeshow, and S. Lemesbow. Goodness of fit tests for the multiple logistic regression model. *Communications in Statistics - Theory and Methods*, 9(10):1043–1069, 1980.

7.5. Logistic Regression Practical

7.5.1. Datasets

The datasets that you will use in this practical can be accessed via http from within stata. However, the directory in which they are residing has a very long name, so you can save yourself some typing if you create a global macro for this directory. You can do this by entering

```
global basedir http://personalpages.manchester.ac.uk/staff/mark.lunt
global datadir $basedir/stats/7_binary/data
```

(In theory, the global variable `datadir` could have been set with a single command, but fitting the necessary command on the page would have been tricky. Far easier to use two separate commands as shown above). If you wish to run the practical on a computer without internet access, you would need to:

1. Obtain copies of the necessary datasets
2. Place them in a directory on your computer
3. Define the global macro `$datadir` to point to this directory.

7.5.2. Cross-tabulation and Logistic Regression

Use the file `$datadir/epicourse.dta`

i Produce a tabulation of hip pain against gender with the command `tab hip_p sex, column`. What is the prevalence of hip pain in men and in women ?

ii Rerun the previous command using the `chi2` option. Is the difference in prevalence between men and women statistically significant ?

iii Use the command `cs hip_p sex, or` to obtain the odds ratio for being female compared to being male. What is the confidence interval for this odds ratio ?

iv How does the odds ratio compare to the relative risk ? (The relative risk is called ”Risk Ratio” by stata).

v Does the confidence interval for the risk difference suggest that hip pain is more common in one of the genders ?

vi Now use logistic regression to calculate the odds ratio for being female, using the command `logistic hip_p sex`. How does it compare to that produced by `cs` ?

vii How does the confidence interval compare to that produced by `cs` ?

7.5.3. Introducing Continuous Variables

Age may well affect the prevalence of hip pain, as well as gender. To test this, create an `agegp` variable with the following commands:

```
egen agegp = cut(age), at(0 30(10)100)
label define age 0 "<30" 30 "30-39" 40 "40-49" 50 "50-59"
label define age 60 "60-69" 70 "70-79" 80 "80-89" 90 "90+", modify
label values agegp age
```

viii Produce a tabulation of hip pain against agegroup with the command `tab agegp hip_p, chi2`. Is there evidence that the prevalence of hip pain increases with age ?

ix Add `age` to the regression model for hip pain with the command `logistic hip_p age sex`. Is age a significant predictor of hip pain ?

x How do the odds of having hip pain change when age increases by 1 year ?

xi To test whether the effect of age differs between men and women, fit the model `logistic hip_p i.sex##c.age`. Is there evidence that the change in prevalence with age differs between men and women

xii Rather than fitting age as a continuous variable, it is possible to fit it as a categorical variable, using the command `logistic hip_p sex i.agegp`. What are the odds of having hip pain for a man aged 55, compared to a man aged 20 ?

7.5.4. Goodness of Fit

Refit the logistic regression model that was linear in age with the command `logistic hip_p age sex`.

5.1 Use the command `estat gof, group(10)` to perform the Hosmer-Lemeshow test. Is the model adequate, according to this test ?

5.2 Use the command `lroc` to obtain the area under the ROC curve.

5.3 Now rerun the logistic regression model that used age as a categorical variable (`logistic hip_p sex i.agegp`). Is this model adequate according to the Hosmer-Lemeshow test ?

5.4 What is the area under the ROC curve for this model ?

5.5 Create an age^2 term with the command `gen age2 = age*age`. Add this to the model with age as a continuous variable (`logistic hip_p sex age age2`). Does adding this term improve the fit of the model ?

5.6 Obtain the area under the ROC curve for this model. How does it compare to those for the previous models you considered ?

7.5.5. Diagnostics

5.7 Obtain the predicted values from the previous regression model with `predict p`. Obtain $\Delta\hat{\boldsymbol{\beta}}_i$ with `predict db, dbeta`. Plot $\Delta\hat{\boldsymbol{\beta}}_i$ against $\hat{\pi}$ with `scatter db p`. Is there any evidence of an influential point ?

5.8 Obtain the deviance residuals with the command `predict d, ddeviance` and plot them against $\hat{\pi}$ with `scatter d p`. Is there evidence of a poor fit for any particular range of predicted values ?

5.9 Plot $\hat{\pi}$ against age with `graph twoway scatter p age`. Why are there two lines ?

5.10 It is possible to compare the logistic regression model predictions to a non-parametric (i.e. making no assumptions about the form of the association) smoothed curve with

```
 graph twoway scatter p age || ///
lowess hip_p age if sex == 1 || ///
lowess hip_p age if sex == 0
```

Does this suggest that the quadratic model is reasonable ? (Note that the three lines of code make up a single command. Thus they can be entered into the do-file editor exactly as shown and run from there. However, if they are to be run from the command window, they need to be entered as a single line, and the comment-start symbols ("///") need to be removed).

7.5.6. The CHD Data

This section of the practical gives you the opportunity to work through the CHD example that was used in the notes. The data is in the file `$datadir/chd.dta`.

The following commands with reproduce Figure 7.2

```
 sort agegrp
by agegrp:  egen agemean = mean(age)
by agegrp:  egen chdprop = mean(chd)
label var agemean "Mean age"
label var chdprop "Proportion of subjects with CHD"
scatter chdprop agemean
```

5.11 Fit the basic logistic regression model with `logistic chd age`. What is the odds ratio for a 1 year increase in age ?

5.12 Save the predicted values from this model with `predict p`, the deviance residuals with `predict d, ddeviance` and $\Delta\hat{\beta}_i$ with `predict db, dbeta`

5.13 Plot $\Delta\hat{\beta}_i$ against $\hat{\pi}$ with `scatter db p`. Are any points particularly influential ?

5.14 Identify the most influential point (using `summ db, detail`). Rerun the regression excluding this point (with `logistic chd age if db <` x, where x is a number of your choice. Does this affect the odds ratio ?

8. Modelling Categorical Outcomes

If the outcome that you wish to model is a categorical variable with more than 2 categories, a more complex model than logistic regression is required. The exact model will depend on whether the categorical variable is nominal or ordinal.

8.1. Nominal Outcomes

8.1.1. Cross-Tabulation

We have seen previously that a 2 × 2 table could be used to examine an association between two dichotomous variables. In fact, the same approach can be used for any two categorical variables, irrespective of the number of categories. If the variable used to define the rows of the table has R categories, and the variable used to define the columns has C categories, you end up with an $R \times C$ table. The expected number of observations in each cell which lies at the intersection of row r and column c can be calculated as

$$E_{rc} = P_r \times P_c \times N = \frac{N_r \times N_c}{N} \tag{8.1}$$

where

P_r Proportion of the sample that is in row r.

P_c Proportion of the sample that is in column c.

N Total sample size.

N_r Number of observations that are in row r.

N_c Number of observations that are in column c.

Just as in the dichotomous case, we can calculated a test statistic by summing $\frac{(O_{rc}-E_{rc})^2}{E_{rc}}$ for each cell in the table. However, we will now have $R \times C$ terms to add, and this statistic will follow a χ^2 distribution on $(R-1) \times (C-1)$ degrees of freedom if the null hypothesis is true. In stata, this test can be performed in exactly the same way as the χ^2-test for 2 × 2 tables, the only difference being the number of categories in the two variables passed to the command.

For example, consider the table below, drawn up to see if males and females tend to have different preferences for their medical insurance (data taken from stata's built in Health Insurance dataset, and can be loaded into stata with the command `webuse sysdsn1`).

	Females		Males		Total	
Indemnity	234	(50.7%)	60	(39.0%)	294	(47.7%)
Prepaid	196	(42.4%)	81	(52.6%)	277	(45.0%)
No Insurance	32	(6.9%)	13	(8.4%)	45	(7.3%)
Total	462	(100%)	154	(100%)	616	(100%)

Table 8.1.: A 3 by 2 table

A smaller proportion of men than women have indemnity insurance, whereas a larger proportion of men than women have prepaid or no insurance. We can quantify these differences with what stata calls the “Relative Risk Ratio”. The relative risk of having prepaid rather than indemnity insurance in males is $\frac{0.526}{0.390} = 1.35$, whereas in females it is $\frac{0.42}{0.51} = 0.84$. The relative risk ratio is therefore $\frac{1.35}{0.84} = 1.61$. Similarly, the relative risk ratio for no insurance rather than indemnity is $\frac{0.08/0.39}{0.07/0.51} = 1.58$

8.1.2. Multinomial Logistic Regression

Whilst it is possible to use a tabulation to get some information about the magnitude of the association between a categorical predictor and a categorical outcome. However, there are occasions when we want to include multiple predictors at the same time, some of which may be quantitative. It is possible to extend the notion of logistic regression to the case where the outcome has more than two categories: this is known as multinomial logistic regression.

Multiple Logistic Regressions

It is easiest to think of multinomial logistic regression as a series of dichotomous logistic regressions. If our outcome variable has R possible categories, we choose one of the categories as our reference or baseline category, and perform $R - 1$ binary logistic regressions, with the outcome taking the value 0 for the reference category in each case, and the value 1 for one particular outcome category (and is considered missing for the other possible outcome categories.

So in the data presented above, we could do a logistic regression of prepaid vs indemnity, and a second logistic regression of no insurance vs indemnity. In this case, indemnity is our reference category. The output of performing these two logistic regressions in stata is shown below:

```
. logistic insure1 male

------------------------------------------------------------------------------
     insure1 | Odds Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        male |   1.611735   .3157844     2.44   0.015      1.09779     2.36629
       _cons |   .8376068   .0811033    -1.83   0.067     .6928203    1.012651
------------------------------------------------------------------------------

. logistic insure2 male

------------------------------------------------------------------------------
     insure2 | Odds Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        male |   1.584375   .5693029     1.28   0.200     .7834322    3.204163
       _cons |   .1367521   .0257746   -10.56   0.000     .0945154    .1978636
------------------------------------------------------------------------------
```

You can see that the odds ratios calculated by the two logistic regressions correspond to the relative risk ratios we calculated from Table 8.1. However, I suggest that you think of them as relative risk ratios rather than odds ratios, since they are only odds ratios if you ignore any possible outcome other than the reference outcome and the one being predicted. In that case, $P(\text{reference outcome}) = 1 - P(\text{regression outcome})$, so that the relative risk $\frac{P(\text{regression outcome})}{P(\text{reference outcome})} = \frac{P(\text{regression outcome})}{1-P(\text{regression outcome})}$, which is the odds of the regression outcome. The ratio of the relative risks is then a ratio of odds.

Combining Multiple Logistic Regressions

Rather then perform multiple logistic regressions in this way, it is possible to fit a single model covering all possible outcomes. Rather than a single linear predictor, as in binary logistic regression, there will need to be $R - 1$, corresponding to the separate logistic regressions that could be fitted. If we represent the linear predictor for the i^{th} observation when comparing the j^{th} outcome to the reference outcome (which we will take to be R) as LP_{ij}, then probability the the outcome for the i^{th} observation takes the value j can be calculated as

$$P(Y_i = j|\boldsymbol{X_i}) = \begin{cases} \frac{exp(LP_{ij})}{1+\sum_{m=2}^{R} exp(LP_{im})} & \text{if } j < R \\ \frac{1}{1+\sum_{m=2}^{R} exp(LP_{im})} & \text{if } j = R \end{cases} \tag{8.2}$$

Multinomial Logistic Regression in Stata

The stata command for fitting this model is `mlogit`. To fit a multinomial logistic regression model for the above data, the command would be

```
mlogit insure male
```

The output from the above command would be

```
Multinomial logistic regression                   Number of obs     =        616
                                                  LR chi2(2)        =       6.38
                                                  Prob > chi2       =     0.0413
Log likelihood = -553.40712                       Pseudo R2         =     0.0057

------------------------------------------------------------------------------
      insure |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
Indemnity    |  (base outcome)
-------------+----------------------------------------------------------------
Prepaid      |
        male |    .477311   .1959283     2.44   0.015     .0932987    .8613234
       _cons |  -.1772065   .0968274    -1.83   0.067    -.3669847    .0125718
-------------+----------------------------------------------------------------
Uninsure     |
        male |     .46019   .3593233     1.28   0.200    -.2440708    1.164451
       _cons |  -1.989585   .1884768   -10.56   0.000    -2.358993   -1.620177
------------------------------------------------------------------------------
```

All of this output has exactly the same interpretation as the output we have seen previously, except that since we have three possible outcomes, so we have 2 linear predictors. Stata chose indemnity as the reference outcome (we will see how to change that shortly), and produced two linear predictors, one for comparing prepaid to indemnity, the other comparing no insurance to indemnity. In each case, the linear predictor takes the form

$$LP = \beta_0 + \beta_1 \times \text{male}$$

with β_0 taking the values -0.177 for the prepaid linear predictor and -1.990 for the uninsured linear predictor. The coefficients β_1 take the values 0.477 and 0.460 respectively.

More often than not, these coefficients are not particularly meaningful, and we would prefer to see relative risk ratios. This can be achieved using the `rrr` option. We can also change the outcome used as the reference with the `baseoutcome()` option. For example, the command

```
mlogit insure male, rrr baseoutcome(3)
```

produces the following output:

```
. mlogit insure male, baseoutcome(3) rrr

Iteration 0:   log likelihood = -556.59502
Iteration 1:   log likelihood = -553.40794
Iteration 2:   log likelihood = -553.40712
Iteration 3:   log likelihood = -553.40712

Multinomial logistic regression                   Number of obs     =        616
                                                  LR chi2(2)        =       6.38
                                                  Prob > chi2       =     0.0413
Log likelihood = -553.40712                       Pseudo R2         =     0.0057

------------------------------------------------------------------------------
      insure |        RRR   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
Indemnity    |
        male |   .6311637   .2267918    -1.28   0.200      .312094    1.276435
       _cons |     7.3125   1.378237    10.56   0.000     5.053987    10.58029
-------------+----------------------------------------------------------------
Prepaid      |
        male |   1.017268   .3605507     0.05   0.961       .50786    2.037639
       _cons |      6.125   1.167805     9.51   0.000     4.215161    8.900163
-------------+----------------------------------------------------------------
Uninsure     |  (base outcome)
------------------------------------------------------------------------------
```

The likelihood ratio χ^2 in this case is 6.38 on 2 degrees of freedom, suggesting that there is a significant association between sex and insurance type. Howeve, the individual RRRs are not statistically significantly different from 1. That is because we have chosen to use Uninsured as our reference outcome, and this has the smallest numbers, and hence lowest power to detect differences. There is, in fact, a statistically significant difference between the RRRs for Indemnity and Prepaid, as we saw in the previous model, but our choice of reference category means that it is not automatically presented as part of the output. We can use lincom to recover it, as we will see in section 8.1.2.

predict Using predict after mlogit is slightly complicated by the fact that there are multiple linear predictors. You therefore need to either provide multiple variables for the predictions to be put into, or specify which one particular prediction you want (if you provide a single variable name, stata assumes you want outcome 1, which may or may not be useful). The most useful options to predict after mlogit are p, which gives the predicted probability for each outcome, and xb, which gives the linear predictor for each outcome (or 0 for the baseline outcome). For example, we the command

```
predict prob*, p
```

would create 3 new variables, called prob1, prob2 and prob3, containing the predicted probabilities of belonging having each of the 3 types of insurance.

lincom The command lincom can also be uaed after the mlogit command, but again the multiple linear predictors make it more complicated. You cannot simple use a variable name (or _cons) to identify a coefficient, you also need to specify which linear predictor the coeffiencet belongs to. We do this by putting the value of the relevant outcome in square brackets before the variable name. For example, suppose that we want to test whether there is a statistically significant difference between the coefficients for male in the prepaid and uninsured linear predictors. We can do this with the command

```
lincom [Prepaid]male - [Uninsure]male
```

and get the following output:

```
 ( 1)  [Prepaid]male - [Uninsure]male = 0

------------------------------------------------------------------------------
      insure |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         (1) |    .017121   .3544302     0.05   0.961    -.6775495    .7117916
------------------------------------------------------------------------------
```

As you might have expected, the difference between the relative risk ratios is not statistically significant.

8.2. Ordinal Variables

If the variable that you want to predict is ordinal, rather than nominal, there are a number of possible approaches.

1. Ignore the ordinal nature of the variable, treat it as nominal

2. Ignore the ordinal nature of the variable, treat it as interval

3. Respect the ordinal nature of the variable.

Options 1 and 2 may seem inappropriate if the data is ordinal, but both can be useful in certain circumstances.

Option 1 may be the best approach if different predictors are important at different levels of the outcome variable. For example, with an ordinal pain rating, rated as none, mild, moderate, severe, it may be that a particular variable reduces the probability of reporting “none”, but has no impact on the level of pain reported. In this case, the variable may be ordinal, but the association between predictors and outcome is not, and an ordinal regression model may not fit well.

Option 2 can be useful for ordinal data with lots of possible values, such as a visual analog scale. Technically, this is an ordinal variable, but it is usually appropriate to treat it as interval.

Option 2 is also commonly used when the ordinal variable is a *predictor*, rather than the outcome. It does make the assumption that each time the predictor goes up 1 category, the outcome goes up by an equal amount, which is quite a strong assumption. Ways of testing this assumption and including the predictor in the most appropriate way are outlined in section 8.2.2. Option 2 for predictor variables is also the basis of the “test for trend”, discussed in section 8.2.1.

8.2.1. Trend Test

We have seen that the χ^2-test can be used to test for an association betwen two categorical variables. However, it treats any deviation of the observed values from the expected values in the same way. It does not specifically test if observations in higher categories of one variable tend to be in higher (or lower) categories of the second variable. However, the χ^2 statistic can be broken down into two components, one of which measures the linear trend, and the other deviations around that trend.

As an example, consider the data in Table 8.2. This is looking to see if there is an association between the reading score (dichotomised as "High" or "Low") and the writing score (categorised into 4 ordinal levels, labelled 0, 1, 2, 3 for ease of interpretation when we get round to doing some regression).

Reading Score	Writing Score			
	Low		High	
0	18	(82%)	4	(28%)
1	42	(53%)	37	(47%)
2	10	(19%)	42	(81%)
3	4	(9%)	43	(91%)

Table 8.2.: Association between reading and writing scores

The proportion of children with high writing scores increases as the reading score increases. The χ^2 statistic is 51.2 on 3 degrees of freedom, giving $p < 0.001$ and suggesting there is an association, but saying nothing about how the association works.

This χ^2 statistic can be decomposed into two parts, one testing the trend for the proportion to increase as the ordinal predictor increases and on testing for variation around this trend. The test is sometimes referred to as the Cochran-Armitage test, and it can be performed in stata with a user-written command `ptrend`:

```
  trend test]
. ptrendi 4 18 0 \ 37 42 1 \ 42 10 2 \ 43 4 3

     +------------------------+
     |  r   nr   _prop      x |
     |------------------------|
  1. |  4   18   0.182   0.00 |
  2. | 37   42   0.468   1.00 |
  3. | 42   10   0.808   2.00 |
  4. | 43    4   0.915   3.00 |
     +------------------------+

Trend analysis for proportions
------------------------------

Regression of p = r/(r+nr) on x:

Slope =  .24784, std. error =  .03548, Z =   6.984

Overall chi2(3) =          51.222,  pr>chi2 = 0.0000
Chi2(1) for trend =        48.781,  pr>chi2 = 0.0000
Chi2(2) for departure =     2.441,  pr>chi2 = 0.2951
```

Here, the χ^2 test for trend is highly significant, the test for departures from a linear trend is non-significant.

This trend test appears regularly in the literature, and was developed by two highly respected statisticians. However, I would suggest that it is *never* the best analysis available. The reasoning behind it is perfectly sound, but if you look at the mathematics, it is exactly equivalent to performing a linear regression, and we have seen that linear regression with dichotomous outcomes is not a good idea. However, we can combine the idea of the Cochran-Armitage test with logistic regression as outlined in section 8.2.2.

8.2.2. Ordinal Predictors

We have seen that we can include a categorical predictor in a logistic regression model by putting an `i.` before the variable's name. If we create a variable called `oread` containing the ordinal reading score and a variable called `gwrite` containing the dichotomous writing score, we can perform a logistic regression.

```
. logistic gwrite i.oread

Logistic regression                               Number of obs     =        200
                                                  LR chi2(3)        =      55.25
                                                  Prob > chi2       =     0.0000
Log likelihood = -104.16821                       Pseudo R2         =     0.2096

------------------------------------------------------------------------------
      gwrite | Odds Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       oread |
          1  |   3.964286   2.366622     2.31   0.021     1.230296     12.7738
          2  |       18.9   12.38441     4.49   0.000     5.232435     68.2684
          3  |     48.375   36.80353     5.10   0.000     10.89005     214.888
             |
       _cons |   .2222222    .122838    -2.72   0.007     .0752087     .656609
------------------------------------------------------------------------------

. testparm i.oread

 ( 1)  [gwrite]1.oread = 0
 ( 2)  [gwrite]2.oread = 0
 ( 3)  [gwrite]3.oread = 0

           chi2(  3) =   40.22
         Prob > chi2 =    0.0000
```

Performing `testparm i.oread` will test whether there is any association between `oread` and `gwrite`. It is conceptually, but not mathematically, comparable to the overall χ^2-test.

To get a test for trend, we add `oread` as a continuous variable a to the model. We can still only have 3 coefficients for `oread` altogether, so one coefficient will need to be dropped. For this reason, it is important to add `oread` *before* `i.oread`, otherwise it will be dropped as the unidentifiable 4^{th} coefficient.

```
. logistic gwrite oread i.oread
note: 3.oread omitted because of collinearity

Logistic regression                               Number of obs     =        200
                                                  LR chi2(3)        =      55.25
                                                  Prob > chi2       =     0.0000
Log likelihood = -104.16821                       Pseudo R2         =     0.2096

------------------------------------------------------------------------------
      gwrite | Odds Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       oread |   3.643681   .9240332     5.10   0.000     2.216545    5.989686
             |
       oread |
         1   |   1.087989   .5068218     0.18   0.856     .4366229    2.711083
         2   |   1.423578   .7522192     0.67   0.504     .5053657    4.010112
         3   |          1  (omitted)
             |
       _cons |   .2222222    .122838    -2.72   0.007     .0752087     .656609
------------------------------------------------------------------------------

. testparm oread

 ( 1)  [gwrite]oread = 0

           chi2(  1) =   26.00
         Prob > chi2 =    0.0000

. testparm i.oread

 ( 1)  [gwrite]1.oread = 0
 ( 2)  [gwrite]2.oread = 0

           chi2(  2) =    0.49
         Prob > chi2 =    0.7844
```

Now, the `testparm oread` tests the linear trend effect of `oread` whilst the `testparm i.oread` tests the departures around the linear trend. In this case, the latter test is not significant, whilst the trend test is, so the best way to include `oread` in our model is as a continuous predictor.

It should be noted that the test for trend using logistic regression is testing linearity on the log-odds scale, where as `ptrend` tests for linearity on the probability scale. However, the important thing is whether the effect is linear *in the model you are using*, and for dichotomous outcomes, the appropriate model is logistic.

8.2.3. Ordinal Outcomes

There are a number of possible approaches when the outcome variable is ordinal. A simple crosstabulation can be used to calculate and ordinal odds ratio. This approach can then be extended to produce a model predicting the probaility of being in each outcome category, respecting the ordinal nature of the data in a number of different ways.

We will explore these methods by applying them to the data in Table 8.3. This compares two treatments, A and B, with the outcome being an ordinal variable with 4 levels: "Healed", "Improved", "No change" and "Worse". On treatment A, most subjects are in the "Healed" and "Improved" categories, while on treatment B, most subjects are in the “No change” or “worse” categories, sugggesting that treatment A is better.

	Treatment A		Treatment B		Total	
Healed	12	(38%)	5	(16%)	17	(27%)
Improved	10	(31%)	8	(25%)	18	(28%)
No Change	4	(13%)	8	(25%)	12	(19%)
Worse	6	(19%)	11	(34%)	17	(27%)
Total	32	(100%)	32	(100%)	34	(100%)

Table 8.3.: Ordinal Outcome Example

Crosstabulation with Ordinal outcomes

From this Table 8.3, we could calculate three odds ratios, by dichotomising the outcome in 3 ways: “Healed” vs “Improved”, “No Change” and “Worse”; “Healed” and “Improved” vs “No Change” and “Worse”; or “Healed” “Improved”, “No Change” vs “Worse”. These odds ratios can be calculated as in we saw for 2 by two tables:

$$OR_1 = \tfrac{(12)\times(8+8+11)}{5\times(10+4+6)} = 3.2 \tag{8.3}$$

$$OR_2 = \tfrac{(12+10)\times(8+11)}{(5+8)\times(4+6)} = 3.2 \tag{8.4}$$

$$OR_3 = \tfrac{(12+10+4)\times 11}{(5+8+8)\times 6} = 2.3 \tag{8.5}$$

This is as much as we can do with cross-tabulation. However, the 3 odds ratios above are not too dissimilar. Would it be reasonable to assume they are all estimates of the same population parameter ? And if so, what is our best estimate of the value of that parameter ?

Ordinal Regression

We can get an estimate of this parameter using ordinal logistic regression, sometimes referred to as ordered polytomous logistic regression. I’ll show how this works in Stata first, then explain what is happening in the background.

Suppose we have a variable `treatmentA` containing 1 for those on treatment A and 0 for those on treatment B, and a variable y containing 1 for “Worse”, 2 for “No Change”, 3 for “Improved” and 4 for “Healed”. We could fit an ordinal logistic regression with the command

```
ologit y treatment, or
```

(the `or` says that we want to see odds ratios rather then coefficients, since we don’t know what the coefficients might mean yet). The output we get is

```
Iteration 0:   log likelihood = -87.993692
Iteration 1:   log likelihood = -85.260015
Iteration 2:   log likelihood = -85.249205
Iteration 3:   log likelihood =   -85.2492

Ordered logistic regression                        Number of obs
                                                   LR chi2(1)
                                                   Prob > chi2
Log likelihood =   -85.2492                        Pseudo R2

------------------------------------------------------------------------------
           y | Odds Ratio   Std. Err.      z    P>|z|     [95%
-------------+----------------------------------------------------------------
  treatmentA |   2.932027   1.367426     2.31   0.021     1.17
-------------+----------------------------------------------------------------
       /cut1 |  -.5635603   .3435512                      -1.23
       /cut2 |   .3179999   .3363157                      -.341
       /cut3 |   1.616945    .396272                       .840
------------------------------------------------------------------------------
```

predict By default, the command `predict` after `ologit` gives predicted probabilities for each outcome, exactly the same as it does after `mlogit`. There is now only a single linear predictor, so if you use the `xb` option, you only need to give a single new variable. However, if you want the probabilities of each possible outcome, you need to give as many variables as there are outcomes.

8.2.4. Alternatives

There are a number of alternatives to the ordered polytomous regression model for ordinal data[**?**, **?**]. One approach is to assume that there is a normally distributed latent variable underlying the ordinal outcome, and that there are thresholds in this latent variable which define which category is manifest. This is the ordinal probit model, which can be fitted with `oprobit`.

Another approach to ordinal data is the Stereotype Regression model. This can be thought of as lying between the ordered polytomous model and the multinomial model, in that it allows variables to affect different transitions in different ways. If a variable has an effect on the transition from level 1 to level 2, but not on the transition from level 2 to level 3, a stereotype regression model is a useful way to model this. Stereotype regression models can be fitted with the command `slogit`.

8.3. References and Further Reading

Further Reading

8.4. Practical For Session 8: Categorical Outcomes

Datasets

The datasets that you will use in this practical can be accessed via http from within stata. However, the directory in which they are residing has a very long name, so you can save yourself some typing if you create a global macro for this directory. You can do this by entering

```
global basedir http://personalpages.manchester.ac.uk/staff/mark.lunt
global datadir $basedir/stats/8_categorical/data
```

(In theory, the global variable `datadir` could have been set with a single command, but fitting the necessary command on the page would have been tricky. Far easier to use two separate commands as shown above). If you wish to run the practical on a computer without internet access, you would need to:

1. Obtain copies of the necessary datasets

2. Place them in a directory on your computer

3. Define the global macro `$datadir` to point to this directory.

8.4.1. Binomial & Multinomial Logistic Regression

The data used for this section was collected as part of a survey of alligator food choices in 4 lakes in Florida. The largest contributor to the volume of the stomach contents was used as the outcome variable `food`, and the charactertics of the alligators are their length (dichotomised as $\leq$ 2.3m and $>$ 2.3m), their gender and which of the four lakes they were caught in.

4.1 Load the alligators data into stata with the command `use $datadir/alligators`, and familiarise yourself with the values used for each of the variables and their meanings with the command `label list`

4.2 Create a new variable `invertebrate` which takes the value 0 if the main food was fish, 1 if the main food was invertebrates and missing if the main food was anything else. This can be done with the command `gen invertebrate = food - 1 if food < 3`

4.3 Produce a cross-tabulation of food against length, with the command

```
tabulate invertebrate size, co
```

You should see that whilst fish and invertebrates are equally common in the smaller alligators, the larger ones are more likely to eat fish than invertebrates.

4.4 Obtain an odds ratio for the effect of size on the probability that the main food is either fish or invertebrates with

```
logistic invertebrate size
```

Is size a significant predictor of food choice ?

4.5 Now create another outcome variable which compares the probability that the main food is reptiles to the probability that the main food is fish with

```
gen reptile = (food == 3) if (food == 1) | (food == 3)
```

4.6 Obtain an odds ratio for the effect of size on the probability that the main food is either fish or reptiles with

```
logistic reptile size
```

Is size a significant predictor of this food choice ?

4.7 Now use `mlogit food size, rrr` to get the odds ratios for the effect of size on all food choices. Which food category is the comparison group ?

4.8 Check that the odds ratios for the invertebrate vs. fish and reptile vs. fish comparisons are the same as before.

4.9 Are larger alligators more likely to choose reptiles rather than invertebrates ? You can test this with

```
lincom [Reptile]size - [Invertebrate]size, eform
```

What is the odds ratio for size in this food choice ?

4.10 Generate a new variable to enable you to check this result using a single logistic regression model (`gen rep_inv = food == 3 if food == 3 | food == 2`). Perform the logistic regression with

```
logistic rep_inv size
```

Are the results the same as you got with `lincom` ?

4.11 Now we are going to look at the influence of the lakes on the food choices. Produce a table of main food choice against lake with

```
tabulate food lake, co chi2
```

Does the primary food differ between the 4 lakes ?

4.12 What proportion of alligators from Lake Hancock had invertebrates as their main food choice ?

4.13 How does this proportion compare to the other three lakes ?

4.14 Now fit a multinomial logistic regression model with

```
mlogit food i.lake, rrr
```

Look at the LR χ^2 statistic at the top: does this suggest that the primary food differs between the lakes ?

4.15 What is the odds ratio for preferring invertebrates to fish in lake Oklawaha compared to Lake Hancock ? Does this agree with what you saw in the table ?

4.16 Confirm your answer to the previous question by using the command `logistic invertebrate i.lake`

8.4.2. Using ***`mlogit`***

This section uses the dataset `$datadir/politics`, which contains information on the effect of gender and race on political party identification.

4.17 Use `label list` to find out the meanings of the variables

4.18 Use `mlogit party race, rrr` to determine the effect of race on party affiliation. How does being black affect the odds of being a republican rather than a democrat ?

4.19 How does being black affect the odds of being an independent rather than a democrat ?

4.20 Use `tabulate party race, co` to confirm that your answers to the previous questions are sensible.

4.21 What is the odds ratio for being a republican rather than a democrat for women compared to men (use `mlogit party gender, rrr` to find out).

4.22 Fit a multinomial model in which party identification is predicted from both race and gender (`mlogit party race gender, rrr`).

4.23 Add the interaction between race and gender, to see if the race influence differs between men and women. Is this difference statistically significant ?

8.4.3. Ordinal Models

This section uses the data in `$datadir/housing`. This data concerns levels of satisfaction among tenants of different types of housing, according how much contact they have with other residents and how much influence they feel they have over the management of their housing.

4.24 Use `label list` to find out the meanings of the variables.

4.25 Does the degree of satisfaction depend on which type of housing the tenant lives in ? (Use `ologit satisfaction i.housing` to find out).

4.26 Of which type of housing are the tenants most satisfied ?

4.27 Test whether `influence` and `contact` are significant predictors of satisfaction

4.28 Create a multivariate model for predicting satisfaction from all of the variables that were significant univariately. Are these predictors all independently significant ? (You may need to use `testparm` for categorical predictors).

4.29 Does the effect of influence depend on which type of housing a subject lives in ? (Fit an interaction term and use `testparm` to test its significance).

9. Modelling Count Variables

9.1. Introduction

We can use logistic regression to model the prevalence of a condition, i.e. the proportion of people who have that condition. However, we have not yet a way to model *incidence*, i.e. the *rate* at which new cases are occurring.

Incidence is not measured as a proportion, but as a rate: the number of events that happen over a fixed amount of time. If events are happening at a fixed rate λ over a time T, then the expected number of events to occur is λT. The *observed* number of events will follow a Poisson distribution with parameter λT.

9.2. Poisson Regression

9.2.1. Introduction

Poisson regression models the rate at which events occur as a function of the covariates. A rate can never be negative (that is, the number of events that have occurred can never decrease, events can't "unhappen". We commonly model the logarithm of the rate, since as this value goes from $-\infty$ to ∞, the rate goes from 0 to ∞.

So the expected number of events, C for a given observation is

$$E[C] = \lambda T$$

where

C is the number of events

λ is the rate at which events happen

T is the duration of followup for that observation.

So if we model $log(\lambda)$ as a linear function of our covariates, we get

$$\begin{aligned}
\log(\hat{\lambda}) &= \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p \\
\hat{\lambda} &= e^{\beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p} \\
E[C] &= T\lambda \\
&= T \times e^{\beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p} \\
&= e^{\log(T) + \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p} \\
log(E[C]) &= \log(T) + \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p
\end{aligned}$$

So if we are going to model the rate, but what we observe is the number of events, we need to include the log of the exposure time, *with a coefficient fixed as 1*, in our linear predictor. This is referred to as the *offset*.

Since we are modelling the log of the rate, an increase of 1 in a x_p corresponds to and increase of β_p in the log rate. This in turn corresponds to *multiplying* the rate by e^{β_p}. So just like logistic regression, where e^{β} is more meaningful than β itself, so with Poisson regression. In this case, e^{β} is a *Rate Ratio*.

For each observation in our dataset, we have an observed number of events C, and an expected number of events $e^{\log(T)+\beta_0+\beta_1 x_1+\ldots+\beta_p x_p}$. We can calculated a χ squared statistic to test whethere the observed values are further from the expected values that we would expect by chance if the expected values are modelled correctly. This statistic will follow a χ^2 distribution on $N - p - 1$ degrees of freedom, where N is our sample size and p is the number of covariates in our model.

Poisson Regression in Stata The basic command for performing Poisson regression in stata is `poisson`. The first variable after the command will be the outcome variable, any subsequent variables will be predictors. You will also almost always want to include an `exposure(`*`varname`*`)` option, where *`varname`* is the name of a variable giving te duration of exposure for each observation. In order to get Rate Ratios rather than coefficients in the output, use the option `irr` (short for Incidence Rate Ratio).

If you use the `predict` command after a Poisson regression, the following options are available:

`n`	(default)	expected number of events (rate × duration of exposure)
`ir`		incidence rate
`xb`		linear predictor, log of the incidence rate

9.2.2. Example

The data in Table 9.1 shows the mortality by age-group and smoking status for a cohort of British male doctors. The study was set up in 1951, which explains why nearly 80% of the exposure time was in the smokers.

If we had a stata dataset containing 10 observations, with a variable `agecat` containing the age group and `smokes` containing the smoking status for that observation, `pyears` containing the Person-Years of followup and `deaths` containing the number of deaths, then we could model that data with the command

```
poisson deaths i.agecat i.smokes, exp(pyears) irr
```

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

Table 9.1.: Mortality by Age-Group and Smoking Status among Male British Doctors

and get the following output:

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

------------------------------------------------------------------------------
      deaths |        IRR   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      agecat |
      45-54  |   4.410584   .8605197     7.61   0.000     3.009011    6.464997
      55-64  |    13.8392   2.542638    14.30   0.000     9.654328    19.83809
      65-74  |   28.51678   5.269878    18.13   0.000     19.85177    40.96395
      75-84  |   40.45121   7.775511    19.25   0.000     27.75326    58.95885
             |
      smokes |
        Yes  |   1.425519   .1530638     3.30   0.001     1.154984    1.759421
       _cons |   .0003636   .0000697   -41.30   0.000     .0002497    .0005296
  ln(pyears) |          1  (exposure)
------------------------------------------------------------------------------
```

This output shows that mortality increases with increasing age, and is nearly 43% higher in smokers than it is in non-smokers. However, if we check the goodness of fit using the command `estat gof`, we find that the fit is poor: the observed values are significantly further from the expected values than we would expect if the model were correct.

```
estat gof

     Deviance goodness-of-fit =  12.13244
     Prob > chi2(4)           =    0.0164

     Pearson goodness-of-fit  =  11.15533
     Prob > chi2(4)           =    0.0249
```

This lack of fit can happen if we have not specified the linear predictor correctly. It could be because we have modelled continuous variables incorrectly, for example assuming that the log of the rate increases linearly with the variable, when the increase is really quadratic. However, that cannot be the explanation in this instance, since we have not continuous variables in our model. With categorical variables, this happens when we are missing interaction terms. In our example, we are assuming that the rate ratio is the same for all age groups: if this is not the case, our model will not fit well.

We can use `predict` with the `n` option to get the expected number of events, and see how the expected and predicted numbers differ: these are shown in Table 9.2.

The expected numbers of deaths are lower than the observed numbers in smokers in the lowest age groups and in non-smokers in the highest age group. This suggests that the rate ratio is changing with age, and we need to incorporate that into our model. Including the interaction between age and smoking

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

Table 9.2.: Expected and Observed Numbers of Deaths in Doctors Study

in our model gives the following output:

```
. poisson deaths i.agecat##i.smokes, exp(pyears) irr

Poisson regression                                Number of obs   =         10
                                                  LR chi2(9)      =     935.07
                                                  Prob > chi2     =     0.0000
Log likelihood =  -27.53397                       Pseudo R2       =     0.9444

-------------------------------------------------------------------------------
       deaths |        IRR   Std. Err.      z    P>|z|     [95% Conf. Interval]
--------------+----------------------------------------------------------------
       agecat |
       45-54  |    10.5631   8.067701     3.09   0.002     2.364153    47.19623
       55-64  |   46.07004   33.71981     5.23   0.000     10.97496    193.3901
       65-74  |    101.764   74.48361     6.32   0.000     24.24256    427.1789
       75-84  |   199.2099   145.3356     7.26   0.000     47.67693    832.3648
              |
       smokes |
         Yes  |   5.736637   4.181256     2.40   0.017     1.374811    23.93711
              |
agecat#smokes |
   45-54#Yes  |   .3728337   .2945619    -1.25   0.212     .0792525    1.753951
   55-64#Yes  |   .2559409   .1935392    -1.80   0.072     .0581396    1.126697
   65-74#Yes  |   .2363859   .1788334    -1.91   0.057     .0536612    1.041316
   75-84#Yes  |   .1577109   .1194146    -2.44   0.015     .0357565    .6956154
              |
        _cons |   .0001064   .0000753   -12.94   0.000     .0000266    .0004256
   ln(pyears) |          1  (exposure)
-------------------------------------------------------------------------------
```

The rate ratio in the baseline 35-44 category is 5.74, much higher than the overall estimate of 1.43 we got previously. The rate ratios in the other age categories are much lower, although the actual values are not given directly in this output. The estimate in the 45-54 is $5.74 \times 0.373 = 2.14$, lower than the estimate in the youngest age group, but still higher than the overall estimate.

Rather than work out the rate ratios in the various age groups by hand, we can use the `lincom` command: that way we get confidence intervals and hypothesis tests as well. To get the rate ratio in the 75-84 age group, the stata command would be

```
lincom 1.smokes + 5.age#1.smokes, eform
```

and the corresponding outcome would be

```
( 1)  [deaths]1.smokes + [deaths]5.agecat#1.smokes = 0

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

In this age-group, the rate ratio for smoking is slightly, but not significantly, less than 1.

9.2.3. Constraints

The last Poisson model we fitted, with different rate ratios for each age group, is called a "staturated" model. This means that there are as many parameters in the model (1 for smoking, 4 for age group, 4 for interactions between smoking and age group and 1 for the constant term) as there are observations in the dataset. This means that it is possible to fit the data perfectly, and the observed numbers of deaths in each group will be exactly equal to the expected numbers of deaths.

It also means that it is not possible to perform a goodness of fit test. The χ^2 statistic is 0, since the observed and expected values are all equal. And the number of degrees of freedom for the test is also 0, since we have 10 observations and 9 variables in the model.

However, the interaction terms for the 55–64 age group and the 65–74 age group look very similar. What would happen if we were to force them to be exactly the same ? That would reduce the number of parameters in our model and enable us to perform a goodness of fit test. It would also simplify the presentation of our model: we would only need to give 4 rate ratios for smoking, not 5. Simplifying the presentation of a model is a much more importand reason for using constraints than enabling a goodness of fit test.

Parameters may be constrained to either equal other paramaters, or to equal a particular value. The stata command to define a constraint is `constraint define` *`n parameter`* `=` *`expression`*, where `n` is an integer that will be used later to identify the constraint, *`parameter`* can be the name of a variable, or a way of identifying a particular level or combination of levels for categorical variables, and *`expression`* can be either another parameter, or a numerical value.

For example, the command to force the rate ratio for age 55–64 to be equal to the rate ratio for age 65–74 would be

```
constraint define 1 3.agecat#1.smokes = 4.agecat#1.smokes
```

We can than fit this constrained model to the data by using the `constraint()` option of the `poisson` command:

```
. poisson deaths i.agecat##i.smokes, exp(pyears) irr constr(1)

Poisson regression                                Number of obs   =         10
                                                  Wald chi2(8)    =     632.14
Log likelihood = -27.572645                       Prob > chi2     =     0.0000

 ( 1)  [deaths]3.agecat#1.smokes - [deaths]4.agecat#1.smokes = 0
------------------------------------------------------------------------------
       deaths |        IRR   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       agecat |
       45-54  |    10.5631   8.067701     3.09   0.002     2.364153    47.19623
       55-64  |     47.671   34.37409     5.36   0.000     11.60056    195.8978
       65-74  |   98.22765   70.85012     6.36   0.000     23.89324    403.8244
       75-84  |   199.2099   145.3356     7.26   0.000     47.67693    832.3648
              |
       smokes |
         Yes  |   5.736637   4.181256     2.40   0.017     1.374811    23.93711
              |
agecat#smokes |
    45-54#Yes |   .3728337   .2945619    -1.25   0.212     .0792525    1.753951
    55-64#Yes |   .2461772    .182845    -1.89   0.059     .0574155    1.055521
    65-74#Yes |   .2461772    .182845    -1.89   0.059     .0574155    1.055521
```

```
     75-84#Yes |   .1577109   .1194146    -2.44   0.015     .0357565    .6956154
               |
         _cons |   .0001064   .0000753   -12.94   0.000     .0000266    .0004256
    ln(pyears) |          1  (exposure)
------------------------------------------------------------------------------
```

You will see that the table of coefficients now has two identical lines: the interaction term between age group and smoking is identical for the 55–64 and 65–74 age groups.

9.2.4. Other considerations

9.3. Negative Binomial Regression

Although Poisson regression can be very useful for modelling count variables, I would not recommend it's use in general. This is because the variance of the Poisson distribution is equal to its mean, but this is not the only kind of distribution that a count variable can follow. If you are modelling a count variable for which the variance is greater than its mean, the variable is said to be "overdispersed".

If you use Poisson regression for a variable that is overdispersed, the standard errors for the model parameters will be too small. This means that hypothesis tests will produce statistically significant results more than 5% of the time that the null hypothesis is true, and confidence intervals will be narrower than they should be. It is therefore essential to test for overdispersion before fitting a Poisson regression model.

Life is made easier by the fact that there is an alternative model for count data which specifically models the overdispersion. This is the negative binomial regression model. There are in fact two types of negative binomial regression model, which differ in the way that they model the overdispersion. They model the variance of the outcome variable Y as either Var(Y) = $\mu(1+\delta)$ or Var(Y) = $\mu(1+\alpha\mu)$. I.e. the overdispersion is either constant (first model) or proportional to the mean of Y (model 2). Both models reduce to the Poisson model if α or δ are 0. So by fitting one of these models, you not only test whether fitting a Poisson model would be appropriate, but you also fit it if it is.

The command for fitting negative binomial models in stata is `nbreg`. Almost all of the options, and commands that can be run after `nbreg` are the same as for the command `poisson`. The only difference is that the `nbreg`command has an `overdispersion()` option: by default it uses Var(Y) = $\mu(1+\alpha\mu)$, but with the option `overdispersion(constant)` it uses Var(Y) = $\mu(1+\delta)$.

9.3.1. Overdispersion Example

To see the difference between Poisson regression and negative binomial regression, consider the output below.

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

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

------------------------------------------------------------------------------
      deaths |        IRR   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
```

```
      cohort |
  1960-1967  |   .7393079   .0423859    -5.27   0.000     .6607305      .82723
  1968-1976  |   1.077037   .0635156     1.26   0.208      .959474    1.209005
             |
       _cons |   .0202523   .0008331   -94.80   0.000     .0186836    .0219527
ln(exposure) |          1  (exposure)
------------------------------------------------------------------------------

. nbreg deaths i.cohort, exposure(exposure) irr

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

------------------------------------------------------------------------------
      deaths |        IRR   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      cohort |
  1960-1967  |   .7651995   .5537904    -0.37   0.712     .1852434    3.160869
  1968-1976  |   .6329298   .4580292    -0.63   0.527     .1532395    2.614209
             |
       _cons |   .1240922   .0635173    -4.08   0.000     .0455042    .3384052
ln(exposure) |          1  (exposure)
-------------+----------------------------------------------------------------
    /lnalpha |   .5939963   .2583615                       .087617    1.100376
-------------+----------------------------------------------------------------
       alpha |   1.811212   .4679475                       1.09157    3.005294
------------------------------------------------------------------------------
Likelihood-ratio test of alpha=0:  chibar2(01) = 4056.27 Prob>=chibar2 = 0.000
```

The Poisson model suggests that the rate is significantly lower in the 1960–1967 cohort than in the baseline cohort. However, the negative binomial model shows that there is highly significant overdispersion (LR test of alpha = 0 has a P-value given as 0.000). Furthermore, there a no longer any significant differences between the cohorts once overdispersion is taken into account.

9.4. Practical For Session 9: Counts

Datasets

The datasets that you will use in this practical can be accessed via http from within stata. However, the directory in which they are residing has a very long name, so you can save yourself some typing if you create a global macro for this directory. You can do this by entering

```
global basedir http://personalpages.manchester.ac.uk/staff/mark.lunt
global datadir $basedir/stats/9_Counts/data
```

(In theory, the global variable `datadir` could have been set with a single command, but fitting the necessary command on the page would have been tricky. Far easier to use two separate commands as shown above). If you wish to run the practical on a computer without internet access, you would need to:

1. Obtain copies of the necessary datasets

2. Place them in a directory on your computer

3. Define the global macro `$datadir` to point to this directory.

9.4.1. Poisson Regression

In this section you will be analysing the dataset `$datadir/ships`. This is data from Lloyds of London concerning the rate at which damage occured at different times to different types of ship. There are 5 types of ship (labelled “A” to “E”), which could have been built in any one of 4 time periods, and sailed during one of two time periods. The aggregate duration of operation of each type of ship is given by `months`, and the number of incidents of damage is given by `damage`.

4.1 Familiarise yourself with the the meanings of each of the variables with the command

```
label list
```

Set the reference categories for type and time built to E and 1975-1979 respectively with the commands

```
fvset base 5 type
fvset base 4 built
```

4.2 Are there any differences in the rates at which damage occurs according to the type of ship ? The command to test this is

```
poisson damage i.type, exposure(months) irr
```

4.3 Are there any differences in the rates at which damage occurs according to the time at which the ship was built ? The command to test this is

```
poisson damage i.built, exposure(months) irr
```

4.4 Are there any differences in the rates at which damage occurs accoding to the time in which the ship was operated ? (You can work out this command for yourself).

4.5 Now add all three variables into a multivariate poisson model. Use

```
testparm i.type
```

to test if type is still significant after adjusting for the other predictors.

4.6 Use

```
predict pred_n
```

to obtain predicted numbers of damage incidents. Compare the observed and predicted numbers of incidents with

```
list type built sailed damage pred_n
```

For which type of ship and which time periods are predicted values furthest from the observed values ?

4.7 Use `estat gof` to test whether the model is adequate.

4.8 Add a term for the interaction between ship type and year of construction (`i.type#i.built`). Use `testparm` to determine whether this term is statistically significant.

4.9 Does this term affect the adequacy of the model as determined by `estat gof` ?

9.4.2. Negative Binomial Regression

This section used data concerning childhood mortality in three cohorts, from the dataset `$datadir/nbreg`. The children were divided into 7 age-bands, and the number of deaths, and the persons-months of exposure are recorded in `deaths` and `exposure` respectively. For some reason, one model that converged perfectly well using `xi:` to define indicators failed when `xi:` was not used, which is why `ltol(0.000001)` has been added to one command below: the model converges with a less severe tolerance criterion.

4.10 Fit a poisson regression model using only cohort as a predictor:

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

Are there differences in mortality rate between the cohorts ?

4.11 Use `estat gof` to test whether the poisson model was appropriate

4.12 Fit a negative binomial regression model to test the same hypothesis:

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

Do you reach the same conclusion about the role of `cohort` ?

4.13 What is the value of the parameter α, and its 95% confidence interval ?

4.14 Fit a constant dispersion negative binomial regression model with

```
nbreg deaths i.cohort, exposure(exposure) dispersion(constant)
irr
```

Is δ significantly greater than 0 in this model ?

4.15 Does this model suggest any different conclusions as to whether the mortality rate differs between cohorts ?

4.16 One possible source of the extra variation is a change in mortality with age. Fit a model to test whether mortality varies with age with

```
nbreg deaths i.age_gp, exposure(exposure) irr
```

Is age a significant predictor of mortality ?

4.17 Would it be appropriate to use Poisson regression to fit this model ?

4.18 Now fit a negative binomial regression model with both age and cohort as predictors (you will need to add the option `ltol(0.000001)` to get this model to converge). Use `testparm` to determine whether both age and cohort are independently significant predictors of mortality.

4.19 Is α significantly greater than 0 in this model ?

4.20 Fit the same model using `poisson`. Does this model agree with the negative binomial model ?

4.21 Use `estat gof` to test the adequacy of this model. Is using a Poisson regression model appropriate in this case ?

9.4.3. Using constraints

This section uses the data on damage to ships from the dataset `$datadir/ships` again.

4.22 Refit the final Poisson regression model we considered with

```
poisson damage i.type i.built i.sailed, irr exposure(months)
```

Which of the incidence rate ratios are not significantly different from 1 ?

4.23 Create predicted numbers of damage incidents with the command

```
predict pred_n
```

4.24 Define a constraint to force the incidence rate ratio for ships of type D to be equal to 1 with

```
constraint define 1 4.type = 0
```

(Note that the constraints are defined on the *coefficients* of the model, rather than the incidence rate ratios. If the coefficient is 0, the incidence rate ratio is 1.)

4.25 Fit this model with the command

```
poisson damage i.type i.built i.sailed, irr exposure(months)
constr(1)
```

How does the output of this command differ from that of the previous Poisson regression command ?

4.26 Use `estat gof` to test the adequacy of this model. How does the constrained model compare to the unconstrained model ?

4.27 Define a second constraint to force the incidence rate ratio for ships of type E to be equal to 1 with

```
constraint define 2 5.type = 0
```

4.28 Fit a Poisson regression model with both of these constraints using the command

```
poisson damage i.type i.built i.sailed, irr exposure(months)
constr(1 2)
```

(The above command should be entered on one line.)

4.29 How does the adequacy of this model compare to that of the previous one ?

4.30 It appears that the incidence rate ratio for being built in 1965-1969 is very similar to the incidence rate ratio for being built in 1970-1974. Define a new constraint to force these parameters to be equal with

```
constraint define 3 2.built = 3.built
```

Fit a Poisson regression model with all three constraints using the command

```
poisson damage i.type i.built i.sailed, irr exposure(months)
constr(1 2 3)
```

(The above command should be entered on one line.) Notice that the lines for 2.built and 3.built are now identical. In what way do these two lines differ from the lines for the other constrained values ?

4.31 What do you think is the reason for the difference you have just observed ?

4.32 Use `estat gof` to test the adequacy of this constrained model. Have the constraints that you have applied to the model had a serious detrimental effect on the fit of the model.

4.33 Obtain predicted counts from this constrained model with the command

```
predict pred_cn
```

4.34 Compare the predictions from the constrained model and the unconstrained model to each other and to the observed values with

```
corr damage pred_n pred_cn
```

How has the fit of the model been affected by the constraints ?

4.35 If you wish, you can examine the observed and predicted values directly with

```
list type built sailed damage pred_n pred_cn
```

Does this list confirm your answer to the previous question ?

9.4.4. Constraints in Multinomial Logistic Regression

Constraints can be applied to many different types of regression model. However, applying constraints when using `mlogit` can be tricky because there are several equations. The syntax is then similar to the syntax we saw last week for `lincom`. For this part of the practical, we are using the same

`$datadir/alligators` dataset that we saw last week.

4.36 Use

```
label list
```

to remind yourself of what the variables mean.

4.37 Fit a multinomial logistic regression model to predict food choice from lake with the command

```
mlogit food i.lake, rrr
```

Are there significant differences between lakes in the primary food choice ?

4.38 What are the odds ratios for preferring invertebrates to fish in Lakes Oklawaha, Trafford and George ?

4.39 It appears that for the choice of invertebrates rather than fish, there is no significant difference between Lake Oklawaha and Lake Trafford. Define the constraint that corresponds to this with

```
constraint define 1 [Invertebrate]2.lake = [Invertebrate]3.lake
```

Fit the model again with this constraint using

```
mlogit food i.lake, rrr const(1)
```

4.40 Even Lake George does not appear to be significantly different from Lake Oklawaha and Lake Trafford. Define a new constraint with

```
constraint define 2 [Invertebrate]4.lake = [Invertebrate]3.lake
```

Fit a multinomial logistic regression model with both of these constraints with

```
mlogit food i.lake, rrr const(1 2)
```

How does the common odds ratio for all three lakes compare to the 3 separate odds ratios you calculated previously ?

10. Modelling Time to Event Data

10.1. Introduction

10.2. Censoring and Survival

10.3. Describing Survival

10.3.1. Survivor function

10.3.2. Stata Commands

```
stset stjoin stsplit sts list sts graph
```

10.4. Comparing Survival Between Groups

10.5. Modelling Survival

10.5.1. The hazard function

10.5.2. Cox Regression

```
stcox
```

10.5.3. Proportional Hazards Assumption

```
stcoxkm stphplot stphtest
```

10.6. Practical 10: Survival Analysis

Datasets

The datasets that you will use in this practical can be accessed via http from within stata. However, the directory in which they are residing has a very long name, so you can save yourself some typing if you create a global macro for this directory. You can do this by entering

```
global basedir http://personalpages.manchester.ac.uk/staff/mark.lunt
global datadir $basedir/stats/10_survival/data
```

(In theory, the global variable `datadir` could have been set with a single command, but fitting the necessary command on the page would have been tricky. Far easier to use two separate commands as shown above). If you wish to run the practical on a computer without internet access, you would need to:

1. Obtain copies of the necessary datasets
2. Place them in a directory on your computer
3. Define the global macro `$datadir` to point to this directory.

10.6.1. Life tables and Survival Curves

This section uses the dataset "`leukaemia`".

6.1 First, set up the data for survival analysis. The time variable is `weeks`, the number of weeks to relapse. The outcome variable is `relapse`, which is 1 if the subject had a relapse at that time and 0 if they did not. Hence the command to set the data up for survival analysis is `stset weeks, fail(relapse)`

6.2 Obtain a life table for the subjects on Drug A with the command `sts list if treatment1 == 1`. What is the median survival in this group (at what time does the survivor function reach 0.5) ?

6.3 How many subjects were lost to followup in this treatment arm ?

6.4 Obtain a life table for the subjects on standard treatment with the command `sts list if treatment1 == 0`. What is the median survival in this group ?

6.5 How many subjects were lost to followup in this treatment arm ?

6.6 Do the answers to your previous questions suggest that Drug A is better, worse, or the same as standard treatment ?

6.7 Produce a Kaplan-Meier curve for each of the treatments with the command `sts graph, by(treatment1)`. Does this confirm your answer to the previous question ?

6.8 Add a horizontal line to the graph by adding the option `yline(0.5)` to the previous command. This line represents half of the group surviving and half having a relapse: the point where it crosses the two survival curves should give you the median survival times you calculated in earlier questions.

6.9 Add the option `lost` to the previous command. This will show how many subjects were censored at each time point. How many subjects were lost to followup in the two treatment arms ? Does this agree with the results you got from `sts list` ?

6.10 Add the option `gwood` to the previous command to obtain confidence bands for the survival curve ? (The odd name for this option is because the formulae used to calculate the confidence bands were developed by a Major Greenwood). Why do the confidence bands get wider over time ?

6.11 Perform a logrank test to compare the survival on Drug A to that on standard treatment, with the command `sts test treatment1`. Is the difference between Drug A and standard treatment statistically significant ?

6.12 Would have had the same answer to the previous question if you had used a Wilcoxon test in place of a logrank test ? (You can do this by adding the option `wilcoxon` to the previous command.)

10.6.2. Cox Regression

6.13 Have a look at the survival curves by white blood cell count using `sts graph, by(wbc3cat)`. Does the white blood cell count affect survival ?

6.14 Do a cross-tabulation of `treatment1` against `wbc3cat` with `tab wbc3cat treatment1, co` Are the proportions of subjects in each of the white blood cell counts categories the same in the two treatment arms ?

6.15 Given that proportion of subjects in the "High" cell count group is greater in the standard treatment arm than in the Drug A arm, would you expect this to have increased or decreased survival in this arm of the trial ?

6.16 White blood cell count is a potential confounder, so we need to adjust for it. First, we will perform an unadjusted Cox regression to obtain the hazard ratio before adjusting. This is done with the command `stcox treatment1`. What is the hazard ratio for Drug A, and its 95% confidence interval ?

6.17 Now obtain the adjusted hazard ratio with the command `stcox treatment1 i.wbc3cat`. What is the adjusted hazard ratio and its 95% confidence interval ?

6.18 How did the confounding by white blood cell count affect the apparent effect of Drug A ? Is this what you expected from the earlier questions ?

6.19 Now we need to test the proportional hazards assumption. First for treatment: produce a plot of the observed and predicted Kaplan Meier plots with `stcoxkm, by(treatment1)`. Are the observed and predicted curves close to each other ?

6.20 Now we can test the same assumption for the effect of white blood cell count, with `stcoxkm, by(wbc3cat)`. Are the observed and predicted curves close to each other ?

6.21 To obtain a formal test, we need to store the scaled and unscaled Schoenfeld residuals by running the command `stcox treatment1 i.wbc3cat, sca(sca*) sch(sch*)` Now enter the command `stphtest` to get an overall test of proportionality. Is the regression model valid ?

6.22 Use the command `stphtest, detail` to obtain tests of proportionality for each individual variable. Is there any evidence of non-proportional hazards ?

10.6.3. Non-Proportional Hazards

6.23 There is a second drug used in this trial, stored in `treatment2`. Compare the survival curves for Drug B and standard treatment with the command `sts graph, by(treatment2)` How does the survival on Drug B compare to that on standard treatment during the first 10 weeks ?

6.24 How does the survival on Drug B compare to that on standard treatment after the first 10 weeks ?

6.25 Superimpose the predicted survival curves from the Cox regression model with `stcoxkm, by(treatment2)`. How do the predicted and observed curves differ ?

6.26 Perform a Cox regression of `treatment2` and `wbc3cat` with `stcox treatment2 i.wbc3cat` Does Drug B have a significant effect on survival ?

6.27 To test the proportional hazards assumption, we need to store the Schoenfeld residuals again. First drop the residual from the previous model with `drop sca* sch*` Then rerun the `stcox` command with the options `sca(sca*) sch(sch*).` Perform the overall test with `stphtest`: is the model appropriate ?

6.28 Test the proportional hazards assumption for each variable separately with `stphtest, detail`. Which variable does not satisfy the assumption ?

6.29 The Kaplan-Meier curves suggest that Drug B has a negative effect on survival initially, then becomes positive. So we will test for different effects before and after 10 weeks. First produce a life-table with `sts list`

6.30 To be able to split the data, you need to have an id for each subject. We can do this with `generate id = _n`. Now each observation has its record number as an identifier.

6.31 Now, for each subject followed for more than 10 weeks, we will split the data into 2 observations, one for the time up to 10 weeks and one for the time after. First, we must include the id information in the stset command with `stset weeks, fail(relapse) id(id)` Then we can split the data with `stsplit split_time, at(10)`

6.32 Check that the life-table remains unchanged by entering the command `sts list` Is it the same as before ?

6.33 Examine the data with `list id weeks relapse split_time _t0 _t`. You should see that for subjects who were followed up for less than ten weeks, there is still a single record. However, for those followed up for more than 10 weeks, there are two records, one with `split_time` == 0, the other with `split_time` == 10. The start of the interval is given by `_t0`, the end by `_t`

6.34 Now we can generate separate treatment variables for the treatment effect before and after 10 weeks. The commands to use are `gen t1 = treatment2*(split_time == 0)` and `gen t2 = treatment2*(split_time == 10).`

6.35 Now fit the Cox regression model with both `t1` and `t2` as predictors with the command `stcox t1 t2 i.wbc3cat` What is the hazard ratio for t1, with its 95% confidence interval ?

6.36 What is the hazard ratio for `t2` ?

6.37 Do these hazard ratios confirm what you were expecting ?

6.38 Drop the residuals from the previous model with `drop sca* sch*` then create new residuals with `stcox t1 t2 i.wbc3cat, sca(sca*) sch(sch*)` Now test the proportional hazards assumption with `stphtest`. Is the model now appropriate ?

6.39 Test the proportional hazards assumptions for each of the variables separately with `stphtest, detail`. Do any of the predictors show non-proportionality ?

11. More about the Stata Language

11.1. Graphs

The graphical capabilities of stata were massively improved for version 8.0. However, the additional power means that the graphics system is slightly more complicated to use than previously, and also slower: it can take several seconds for the first graph in a session to appear. Subsequent commands of the same type are less slow.

All graph commands start with the word `graph`. In some cases, this can be omitted, but not always.

11.1.1. A simple scatterplot

Very commonly, you will want to plot one variable against another. This is achieved using the command `graph twoway`, or simply `twoway`. There are a number of subcommands to `twoway`, to give different types of plot. We will meet many of them later, but the simplest is `scatter`, to give a scatter plot. The effect of `twoway scatter mpg weight` using the `auto` data is given in Figure 11.1. Note that the horizontal axis (x-axis) is the second variable, and the first variable is plotted on the vertical (y) axis.

11.1.2. Labelling Graphs

Stata makes sensible guesses as to what titles and labels you want on your graph, but they can be changed easily if required. The options most commonly used relate to giving the graph a title[a] or labelling and setting tick-marks on the axes.

The title options most commonly used are:

`title` The overall title, which by default appears centred at the top of the graph

`subtitle` A subtitle appearing centred above the graph below the overall title by default

`note` A note about the graph appearing left-justified below the graph

`caption` A caption for the graph appearing left-justified below the note, if any.

Each of these options needs to be given one or more strings (i.e. contained in inverted commas). If more than one string is given, the two strings appear on separate lines. So, for example, `title("Life Expectancy" "1900-2000")` would create the title

[a] or several: there is scope for 12 differently placed titles altogether, although many of them are used very rarely

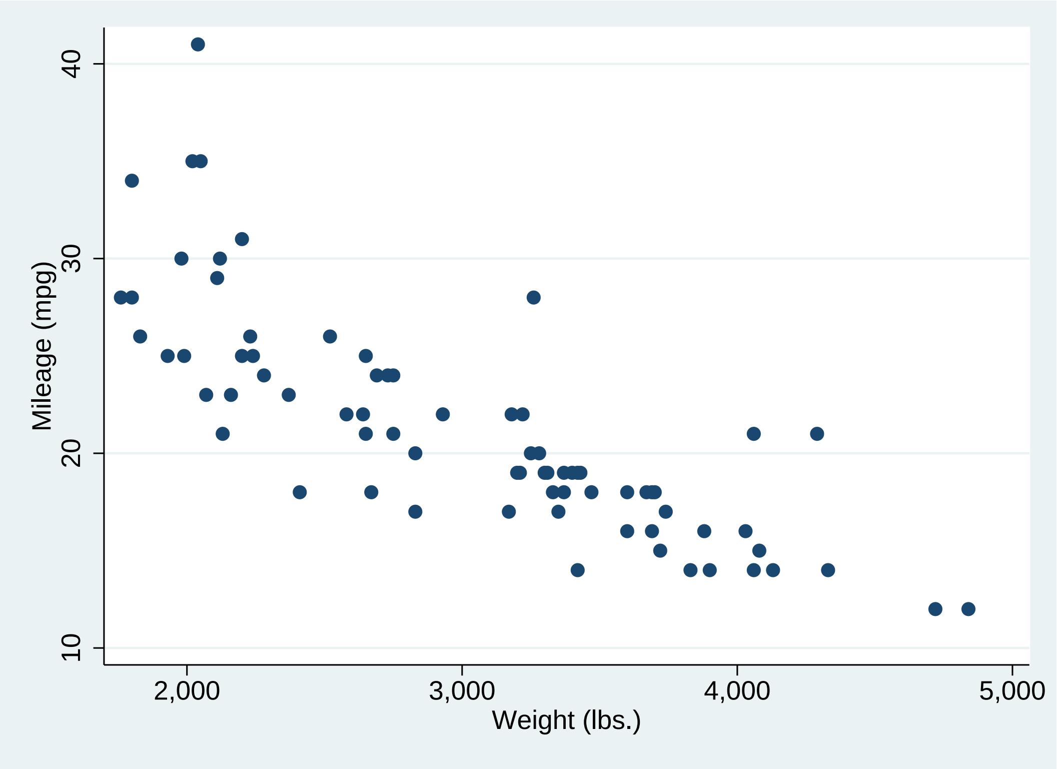


Figure 11.1.: Simple Scatterplot

Life Expectancy
1900-2000

The titles for the axes are generally taken from the corresponding variable's label, or if it has no label, its name. Hence the labels "Weight (lbs)" and "Mileage (mpg)" in the Figure 11.1. However, they can be changed using the options `xtitle()` and `ytitle()`.

There are a vast number of options to control the scale, tick-marks and values labelled on the axes: see `help axis_scale_options` and `help axis_label_options`. The most important ones are `[x|y]scale(log)` to produce a log-scale for the x or y-axis, and `[x|y]label` to define the values of the axis that are to be labelled. You can use a `numlist` (which we met in session 1) to define the values, or you can use the form `xlabel(#`n`)`, where n is an integer, which means "create about n labels at reasonable values for me".

11.1.3. Overlaying Graphs

It is very common to want to overlay graphs. For example, you may wish to add a regression line to a scatter plot. Producing graphs of parameter estimates with confidence intervals also requires overlaying graphs.

There are two different syntaxes for combining graphs: you can either list each plot within parentheses, or separate them with two "pipe" symbols `"||"`. For example, suppose we want to plot both `mpg` and `length` against `weight`: we could use either

```
twoway (scatter mpg weight) (scatter length weight, yaxis(2))
```

or

```
twoway scatter mpg weight || scatter length weight, yaxis(2)
```

Both would produce the graph in Figure 11.2. The `yaxis(2)` option tell stata to use a different Y-axis for the two variables: as you can see, they cover very different ranges.

Figure 11.2.: Overlaid Scatterplots

Although overlaying graphs in that way is possible, it is more useful to be able to overlay graphs of different types. For example, to overlay a regression line on the scatter-plot of mpg against weight, together with a 95% confidence region, the command would be

```
twoway lfitci mpg weight || scatter mpg weight
```

The results are shown in Figure 11.3. Note that plots are overlayed from left to right, so we give the `lfitci` (Linear FIT with Confidence Interval) plot first. If it were given second, the points lying within the confidence region would be hidden by the shading of the confidence region.

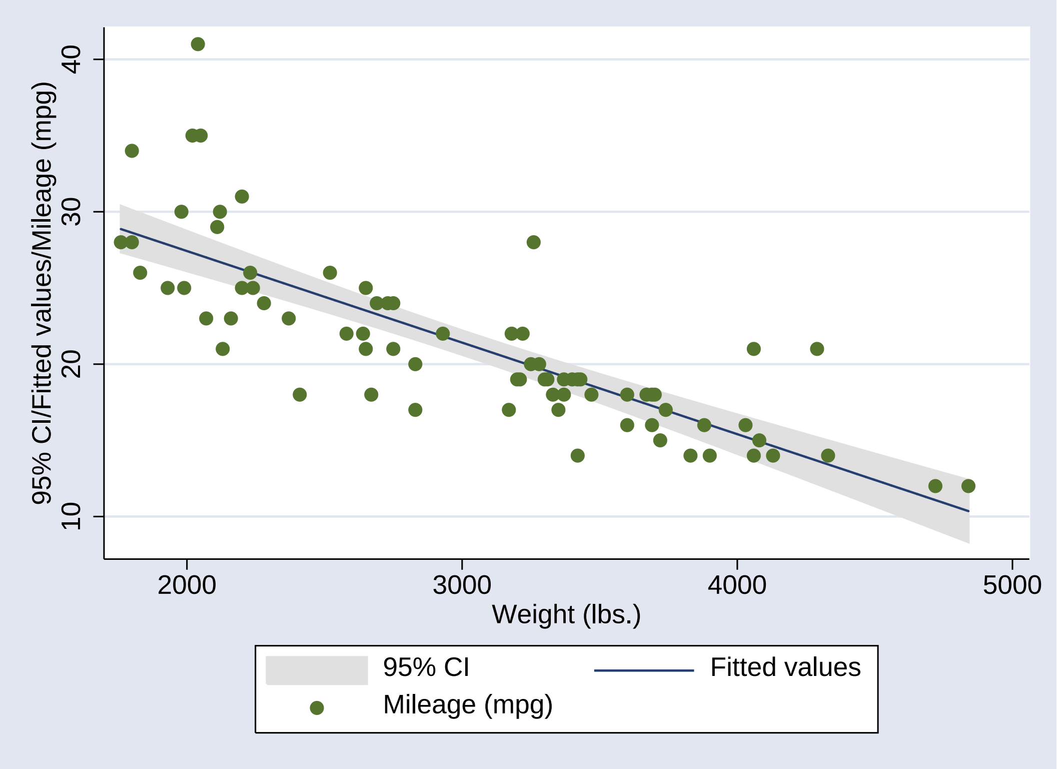


Figure 11.3.: Scatterplot with overlaid regression line and confidence interval

11.1.4. Graph Schemes

You often want to present exactly the same data in exactly the same way, but with slight differences in presentation. For example, a simple black graph on a white background would be suitable for a journal, but if you were presenting the same data in a Powerpoint presentation, you may wish to use a coloured background, make the lines bolder, increase the size of the lettering relative to the overall size of the graph etc. This can be achieved very simply with schemes.

There are a number of schemes delivered with stata, and it is also possible to define your own[b]. Some of the available schemes are listed in Table 11.1. In general, the `s1` family are simpler than the `s2` family.

To set a scheme for the rest of your stata session, the command is

```
set scheme scheme_name
```

If you want that scheme as your default scheme for all your stata sessions, use

[b] Basically, copy an existing scheme to your own do-file directory (C:/ado/personal in most cases), save it with a new name, and edit it to achieve the effects you want.

Family	Scheme	Foreground	Background	Description
s1	s1color	coloured	white	colour on white
	s1rcolor	coloured	black	colour on black
	s1mono	monochrome	white	grey on white
	s1manual	monochrome	white	`s1mono`, but smaller
s2	s2color	coloured	white	default
	s2mono	monochrome	white	grey on white
	s2manual	monochrome	white	`s2mono`, but smaller
others	sj	monochrome	white	Used in Stata Journal
	economist	colour	white	copied from *The Economist*

Table 11.1.: Available Graphics Schemes

```
set scheme scheme_name, permanently
```

Alternatively, you set set a scheme for a single graph by adding the scheme name as the `scheme` option, e.g.

```
twoway scatter mpg weight, schemesj
```

To give an example of how the scheme can affect the appearance of a graph, Figure 11.4 was created using the commands

```
set scheme economist
twoway lfitci mpg weight || scatter mpg weight
```

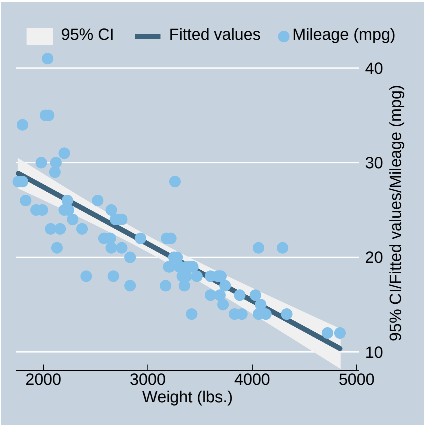


Figure 11.4.: Simple scatterplot in *Economist* scheme

11.1.5. Saving Graphs

Graphs can be saved in stata's own graphical format with the command `graph save`. Graphs can be saved either "live", which is the default, or "as-is" using the option `asis`. A live graph can be recalled to stata and its appearance altered by using a different scheme, whilst an as-is graph will always look exactly as it did when it was saved.

It is also possible to save graphs in other formats using the command `graph export`. The format of the graph is determined by the file suffix, but this can be overriden by using an option to the command if an unusual file suffix is required. The file formats available are listed in Table 11.2.

Output format	file suffix	option
PostScript	.ps	as(ps)
Encapsulated PostScript	.eps	as(eps)
Windows Metafile	.wmf	as(wmf)
Windows Enhanced Metafile	.emf	as(emf)
PDF	.pdf	as(pdf)
Macintosh `PICT` Format	.pict	as(pict)

Table 11.2.: Available `graph export` options

For inclusion in any sensible software, the format of choice is Encapsulated PostScript. However, Powerpoint can't handle PostScript, so you need to use Metafiles or Enhanced Metafiles.

11.1.6. Naming Graphs

As well as storing a graph as a file, it is possible to store it in memory, so that stata can recall it quickly, rather than having to replot it. This is done by giving the graph a name. This can be done by using the `name` option in the graph command:

```
twoway scatter mpg weight, name(scat)
```

The graph can then be reviewed using the command

```
graph display scat
```

By default, every graph is given the name `Graph`. It is also possible to change the name of a graph using the `graph rename` command. So

```
graph rename Graph scat
```

will change the name of the currently displayed graph to `scat`, so that it is not overwritten by the next graph to be displayed.

11.1.7. Other Graph Types

This is not meant to be a complete list of all graph types in stata, but an introduction to the most commonly used ones. See `help graph` for (lots) more information

`graph bar` Bar charts

`graph box` Box and whisker plots

`graph matrix` Given n variables, creates an n by n matrix of scatterplots, plotting every variable against every other variable.

`twoway histogram` Histograms

`twoway lfit` Linear regression fit to a scatter plot

`twoway qfit` Quadratic regression fit to a scatter plot

`twoway fpfit` Fractional polynomial fit to a scatter plot

`twoway lowess` Non-parametric smoothed fit to a scatter plot

`twoway rcap` Given two y-values for each x-value, plots a line between the two y-values, with "caps" at each end. Useful for showing confidence intervals if overlaid.

`kdensity` Can be thought of as a smoothed histogram. Very useful for comparing 2 or more distributions: see Figure 11.5

11.2. Summarizing Data

11.2.1. `describe`

To find out what is in a dataset, you can use the command `describe`. This gives some information as to the size of the dataset, and the number of variables and observations. It also gives, for every variable, the name, type, display format[c] and any labels assigned to the variable. If you are only interested in a subset of the variables, you can use `describe` *`varlist`*.

11.2.2. `codebook`

The `codebook` command gives a little more detail about each variable. This gives, for each variable, the type, range, number of unique values, number of missing values and the units used (the minimum distance between values of the variable). In addition, it gives the mean, standard deviation and various percentiles for continous variables, and a freqency table for categorical variables (or typical values if there are too many for a full table). This command is particularly useful when data-cleaning: it makes implausible values obvious.

[c] this controls how many decimal places are shown for each variable, for details type `help format`

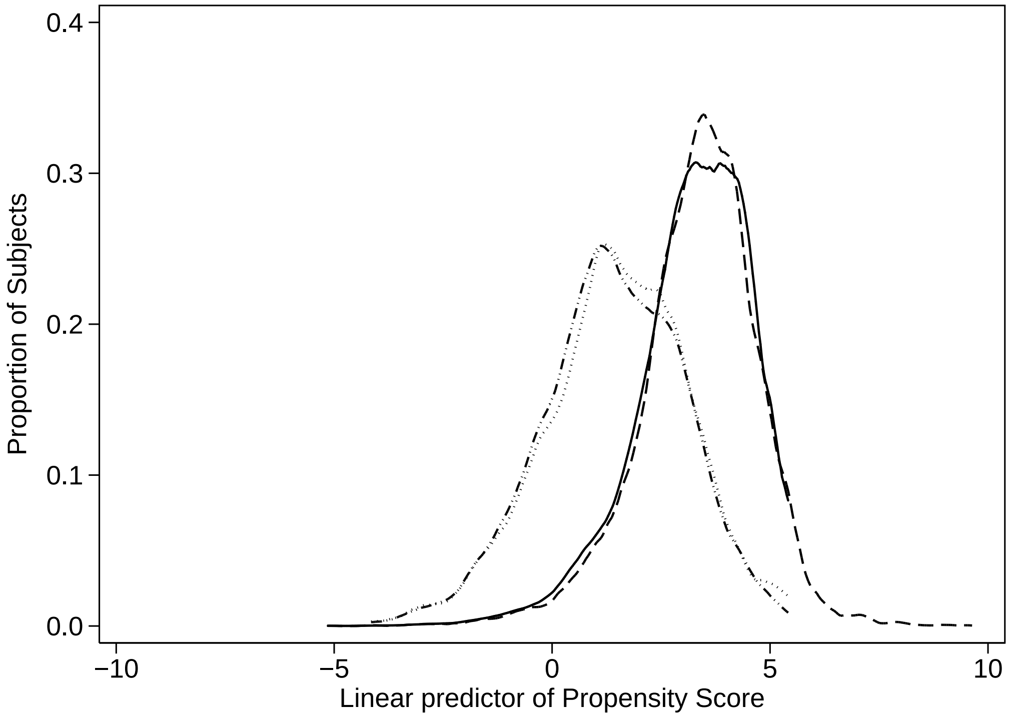


Figure 11.5.: Example kernel density plot

11.2.3. `summarize`

Another command that can be useful when inspecting data is `summarize`. In its simple form, `summarize` *`varlist`* gives the mean, standard deviation, minimum, maximum and number of non-missing values for each variable in *`varlist`*, in a very compact form. The option *`detail`* gives a fuller summary, including various percentiles, the 5 largest and 5 smallest values, the skewness and kurtosis.

11.2.4. `tabulate`

Whilst `summarize` is useful for numerical variables, the command for finding out about categorical variables is `tabulate`. The command `tabulate` *`varname`*, with a single variable, produces a freqency table for that variable. Beware: if the variable is continuous, the frequency table can be extremely long !

Another use for tabulate is to produce cross-tabulations of two variables. The syntax for this is

```
tabulate varname1 varname2
```

11.3. More Command Syntax

In session 1 we saw a simplified version of the syntax that (almost) all stata commands follow. However, there are some additional optional parts that we will consider this week. The full syntax we will consider is

```
[by varlist]:  command varlist [if expression][, options]
```

11.3.1. The `if` clause

It is possible to restrict a command to run on only a subset of your data using an “if” clause. For example, in the `auto` data

```
summarize weight if foreign == 1
```

will produce a summary of weight only for the foreign cars, and

```
summarize weight if foreign == 0
```

will produce a summary only for the domestic cars.

Note that *two* equals signs are required: this is a hangover from the “C” programming language (used by the programmers who wrote stata), in which "=" is used to assign a value and "==" is used when comparing two values. This is probably the most irritating “feature” of stata, and one that causes a huge amount of wasted time. If you put a single equals sign by mistake, the error message is simply `invalid syntax` which is not a great help.

The operators that can be used in logical expressions are listed in Table 11.3

Operator	Meaning
&	and
|	or
==	equal
˜=	not equal
!=	not equal
<	less than
<=	less than or equal
>	greater than
>=	greater than or equal

Table 11.3.: Operators for logical expressions

For example

```
if (foreign==0) | (rep78 != 3)
```

will select any cars that are either U.S. made or have a repair history score other than 3.

A very important thing to remember when using logical expressions is that the missing value is larger than any non-missing value. Hence the expression

```
if price > 15000
```

would pick out any cars which had missing prices, as well as cars with price over $15000. If you only want to select cars with prices known to be over $15000, you must use

```
if (price > 15000) & (price != .)
```

This odd behaviour of the missing value is particularly important when dividing a continuous variable into categories. For example, in the first practical, you divided cars into long and short using the code

```
generate short = 0
replace short = 1 if length < 190
```

This was fine because there was no missing data in `length`. However, any cars who did have missing lengths would be given a 0 for `short`, which is not what we want. It would be better to use the code

```
generate short = 0 if length  = .
replace short = 1 if length < 190
```

11.3.2. The `by:` construct

You may want to process different subgroups of your data differently. For example, you may want to calculate the mean value of a variable in men and women separately, or in cases and controls separately. This can be done using the `by:` part of the command.

The important thing to remember with the `by:` construct is that the data needs to be sorted before it is processed. Suppose that we wish to know the mean weight of foreign and domestic cars separately. The commands to use would be

```
sort foreign
by foreign:  summarize weight
```

However, there is a command `bysort` which will, if used in place of `by` both sort the data and produce output for each subgroup separately.

These commands should produce the same results as

```
summarize weight if foreign == 0
summarize weight if foreign == 1
```

Which method is preferable depends on

1. How many groups there are: `by` is better than `if` if there are large numbers of groups
2. Whether you need results for all groups or not
3. Whether the command is “byable”: although `by` can be used with most official stata commands, there are a few, and some user-written add-ons, which do not accept it.
4. If you need a complex expression to identify the group you are interested in, only an `if` clause will do.

11.3.3. Subscripting

A finer subdivision of your data is possible using subscripting, which effectively treats each observation as a separate group. Subscripting is achieved by putting the number of the observation that you want in square brackets after the variable name: e.g. weight[7] means the weight of the 7^{th} car in the dataset.

Two macros that are commonly used in subscripting are `"_n"` and `"_N"`. The macro `"_n"` represents the number of the current observation, whilst `"_N"` represents the total number of observations.

Two particular uses of subscripting are to create “lagged” differences and to find the number of subjects in a particular subgroup.

Lagged Differences

You may wish to calculate the difference between one measurement and the next. For example, consider the blood pressure data we have seen previously. With the data in wide form (one observation and two variables per subject, see section 11.5) this can be done easily. However, if there are a large number of observations per subject, it may be easier an more efficient to leave the data in long form (several observations per subject). In this case, you want to subtract from each observation, the value of the previous observation.

patient	sex	agegroup	when	bp
1	0	1	1	143
1	0	1	2	153
2	0	1	1	163
2	0	1	2	170
3	0	1	1	153
3	0	1	2	168

Table 11.4.: Extract from `bplong`

Consider the data in Table 11.4. The change in blood pressure is given by subtracting the `bp` when `when` is 1 from the value of `bp` when `when` is 2. This can be calculated as follows:

```
sort patient when
by patient:  gen bpdiff = bp - bp[_n-1]
```

Subgroup size

The size of a subgroup is given by `_N` directly, with `by:` being used to define the subgroups. So, if we consider the `bpwide` dataset, we can generate a variable containing the the number of men and women with the code

```
sort sex
by sex:  gen count = _N
```

11.4. Looping

You often want to perform the same command repeatedly, with only slight changes. The command `foreach` can help to do this. In its simplest form, the syntax is

```
foreach macname in list {
list of stata commands
}
```

For example, using the `bplong`

```
foreach visit in 1 2 {
summarize bp if when == 'visit'
}
```

will produce

```
    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
          bp |       120      156.45    11.38985        138        185

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
          bp |       120    151.3583    14.17762        125        185
```

There are more specialised forms of `foreach`, but they are all very similar. For example,

```
foreach varname of varlist list
```

will check that each *varname* in *list* is the name of an existing variable before performing the commands. It can also expand a *varlist*, which foreach ... in ... cannot. Notice that all of the specialised forms use foreach ...of *listtype* ..., rather than foreach ...in ...

The varlist form of foreach can be particularly useful when labelling variables. Suppose that there are a series of variables, all called *site*_pain, and all taking the value 0 for no and 1 for yes. Rather than label each variable individually, you can simplify with the following code:

```
label define yesno 0 "No" 1 "Yes"
foreach x of varlist *_pain {
label values `x' yesno
}
```

Thus four lines of code can label as many variables as necessary.

11.5. Wide Form vs. Long Form

We saw in session 1 that when the data consists of several observations on each subject, the data can be stored either as several observations per subject (long form) or as a single observation with several variables per subject (wide form). Either form may be needed for analysis, so it is important to be able to change between them freely. This can be done with the reshape command.

There are two forms of the reshape command: reshape wide converts from long form to wide form, and reshape long converts from wide form to long form.

11.5.1. Converting from long to wide

First, we will see how to convert from long form to wide form. We will use the data listed in Table 11.5. This data consists of ID number and gender for each of three subjects, and their x-ray score at 1 year, 2 years and possibly 5 years after registration in the study.

ID	Gender	Anniversary	Score
900108	1	1	7
900108	1	2	15
900108	1	5	19
900113	2	1	0
900113	2	2	18
900114	1	1	0
900114	1	2	0

Table 11.5.: X-Ray data in long form

In order to be able to reshape the data, stata needs to know:

- How to recognise which observations belong to the same subject

- How to recognise which visit a particular observation relates to
- Which variables should change between visits

In this case, we have `ID` to show which observations belong together, and `anniversary` to show which visit the observation belongs to. The only variable that should change between anniversaries is the x-ray score.

The basic syntax for the `reshape wide` command is

```
reshape wide changing_vars, i(ID_var(s)) j(visit_vars)
```

Both of the options `i()` and `j()` must be given. So in our case, we would type

```
reshape wide score, i(id) j(anniversary)
```

and the resulting dataset would look like Table 11.6

ID	Gender	Score1	Score2	Score5
900108	1	7	15	19
900113	2	0	18	.
900114	1	0	0	.

Table 11.6.: X-Ray data converted to wide form

Notice that there are now 3 variables for `score`, since it changes between visits, but only one variable for `sex`, because it does not. If there is a variable that changes between visits, but it is not included in the list of variables to reshape, that is an error and stata will refuse to continue. Notice also that there are now missing values for the score at 5 years for the second and third subjects, since this data was not in the original (long) dataset.

11.5.2. Converting from wide to long

Now suppose that we have the data in Table 11.6, and need to get to long form (Table 11.5). We need to tell stata

- Which variable is a unique identifier for our subjects
- The name of a variable for it to create to contain the visit number (`anniversary` in Table 11.5
- Which of the variables are in wide form, and need to be converted. Strictly, we give the name that the variable will take *when it is in long form*

So, the command we need will start

```
reshape long score
```

Stata will then look for all variables that begin with `score`, and therefore know that we want `score1`, `score2` and `score5` converted to `score`. Again, our identifier is `id` and the option `j` will can be given the name `anniversary`, to match that in Table 11.5. So the entire command is

```
reshape long score, i(id) j(anniversary)
```

and the result is shown in Table 11.7. This is almost identical to that in Table 11.5, but contains two extra records for 5^{th} anniversary visits for subjects 900113 and 900114. These visits did not take place, so the data is missing.

ID	Gender	Anniversary	Score
900108	1	1	7
900108	1	2	15
900108	1	5	19
900113	2	1	0
900113	2	2	18
900113	2	5	.
900114	1	1	0
900114	1	2	0
900114	1	5	.

Table 11.7.: X-Ray data converted to long form

It is important to notice that in the above example, the variables being reshaped all consisted of the prefix (`score`) followed by a numerical suffix (1, 2 or 5). The variable `anniversary` is therefore numeric. If any of the suffixes are non-numeric, you *must* use the option `string` so that stata uses a string variable for `j`. We will see an example in the practical.

11.6. Other Useful Commands

11.6.1. `display`

The command `display` simply causes its argument to appear in the results window. This can have a number of uses. For example, it can function as a simple calculator:

```
display 2+2
```

It can also be useful to check the contents of a macro, if you are not sure what it contains:

```
display "$mydir"
```

will show what the macro `$mydir` contains, or simply a blank line if `$mydir` is not defined.

11.6.2. **expand**

The command `expand` creates additional observations in your dataset. The command itself is followed by an expression which says how many identical copies of each record should be created. For example,

```
expand 2
```

will create 1 new identical copy of each existing record in the dataset.

This can be useful if you wish to recreate a dataset from a summary table in a paper. Suppose the paper contained the following table:

Exposed	Cases	Controls
No	20	40
Yes	30	10

Table 11.8.: `expand` example data

Then you can enter the following data into stata:

exposed	case	frequency
0	0	40
0	1	20
1	0	10
1	1	30

If you then type

```
expand frequency
```

you will end up with a dataset containing 100 observations, and the command

```
tab exposed case
```

will recreate Table 11.8.

11.6.3. **cmdlog**

As well as being able to start a log file for your output, it is possible to start a command-log file, which contains only the commands that you have entered. This can be done at the same time as a log file, and I recommend doing so everytime that you start stata. I have the following commands in my `profile.do` to achieve this:

```
local logname "$S_DATE $S_TIME"
local newname :  subinstr local logname " " "_", all
local newname :  subinstr local newname ":" "_", all
local logname "C:/cmdlogs/‘newname’.txt"
cmdlog using "‘logname’"
```

The first line gets the current date and time to use as the name for the log-file. The second line changes spaces to underscores, and the third line changes colons to underscores. The fourth line defines a macro to use as the name of the logfile, and the last line starts the log-file.

11.7. Practical on Refinements of Stata

Start stata.

11.7.1. Graphs

Type

```
sysuse uslifeexp
```

to load up a dataset concerning life expectancy in various subgroups in the U.S. from 1900 - 2000. Produce a simple scatterplot of life-expectancy against year with the command

```
twoway scatter le year
```

You should see life expectancy increasing steadily, with a blip of very low life expectancy in 1918. Now we will add a title to this graph. Press `PageUp` to recall the previous command, and add the `title()` option:

```
twoway scatter le year, title("U.S. Life Expectancy")
```

Now we are going to extend the y-axis back to 0, rather than starting at 40. The option to do this is `ylabel(0(20)80)`: this will produce labels every 20 years from 0 to 80.

Now we will practice overlaying graphs, using the same dataset. We can compare male and female life-expectancy with the command

```
twoway scatter le_male year || scatter le_female year
```

You should see that life-expectancy is increasing over time in both sexes, but consistently higher in females. You can add regression lines to the plot with:

```
twoway scatter le_male year || scatter le_female year || lfit le_male year
|| lfit le_female year
```

(The above command must all be entered on a single line)

You should notice that the label on the y-axis is silly, consisting simply of the names of the four variables being plotted. A more sensible label would be given by the option `ytitle("Life Expectancy")`.

11.7.2. Summarizing Data

Read the `cancer` dataset into stata with the command

```
sysuse cancer, clear
```

Find out what the dataset is about with the command `describe`

7.1 How many observations are there in the dataset ......

Now use the command `codebook` to get some idea of the values taken by the different variables.

7.2 What was the longest follow-up time ? ......

7.3 How many different treatments were in the study ? ......

7.4 How old were the oldest and youngest subjects in the study ? ......

Use the command `summarize age studytime, det` to obtain details about the ages and lengths of follow-up in the study.

7.5 What was the mean age at the start of the study ? ......

7.6 What was the standard deviation of the follow-up time ? ......

Use the command `tab drug died, row` to produce a cross-tabulation of the number, and percentage, of subjects who died on each treatment.

7.7 How many subjects on placebo died ? ......

7.8 What percentage of subjects on treatment 2 died ? ......

11.7.3. Further Syntax

`if`

7.9 What is the mean age of subjects in the cancer study who died ? ......
The command you need is `summarize age if died == 1`

7.10 Again using an `if` clause, find the mean followup time among subjects on placebo ......

7.11 What was the mean age among subjects who died after being treated on placebo ? ......

by

Use by clauses to verify the answers to the previous three questions. *Hint: for the last one, you need to use* `by drug died:`

Now load the `cancer` dataset with

```
sysuse cancer
```

Create a variable called `agegrp`, dividing subjects into 2 more-or-less equal sized age groups:

```
egen agegrp = cut(age), group(2)
```

To find out at what age the groups were split, enter

```
bysort agegrp:  summarize age
```

7.12 How could you have found out the age at which the split would have occured before making `agegrp` ......

Create a label for `agegrp` so that you know the actual age-range in each age-group appears in any printout. Assign this label to `agegrp`. Check that you were successful with

```
tabulate agegrp died
```

Create a new variable containing the number of subjects in each age-group using the command

```
bysort agegrp:  gen group_size = _N
```

You can check that this has worked correctly with

```
tab group_size
```

Save this dataset as `mycancer`, as we will need to use it later.

11.7.4. Looping

For a simple illustration of looping, type

```
foreach x in one two three {
display " `x' "
}
```

(The opening single inverted comma is at the top left of the keyboard, the closing one at the right hand end of the “asdf” row).

You should see the words “one”, “two” and “three” appear, one on each line.

You should still have the dataset called `mycancer` in stata. If not, load it using

```
use mycancer
```

You can now enter the following code:

```
foreach x in drug agegrp {
tab `x' died, row
}
```

This should produce two cross-tabulations, one for `drug` against `died`, and the other for `agegrp` against died.

For a more complex example of using `foreach`, load the life-expectancy data with

```
sysuse uslifeexp
```

We will create graphs for all of the variables with the code

```
foreach x of varlist le* {
twoway scatter `x' year, name("`x'")
}
```

In the name option, you must put inverted commas around the local macro `` `x' `` so that stata knows that it is a string. This will produce a series of scatterplots called `le`, `le_male` etc. You can recall one of these graphs with, for example

```
graph display le_female
```

11.7.5. Reshaping Data

Long to Wide

Read the `bplong` dataset into stata with the command

```
sysuse bplong
```

Use a `by` clause to get the mean and standard deviation of the blood pressure when `when == 1` and `when == 2` separately. Record these you that you can check later that your reshaping was successful.

To reshape this data into wide form, the unique identifier is `patient` and `j` is `when`. Therefore, the command you need is

```
reshape wide bp, i(patient) j(when)
```

Now use the command

```
summarize bp1 bp2
```

to ensure that the data has been transformed correctly.

Wide to Long

Load the life-expectancy data into stata, using the command

```
sysuse uslifeexp, clear
```

There are series of variables giving the life-expectancy in different subgroups of the U.S. population over the years, and a variable `le` containing the overall life-expectancy. We will change this to have several observations for each year, a single variable `le` containing the life-expectancy and a variable `group` saying which subgroup the life-expectancy applies to.

First, we need to change the name of the variable `le` to `le_total`, so that there is something to put in the `group` variable for the overall life-expectancy. The command to do this is

```
rename le le_total
```

Now, unique observations are identified by the variable `year`, the variable we want to have in the long data is `le`, and the variable we want to identify which variable in wide form corresponds to an observation in long form is called `group`, so the reshape command we need is

```
reshape long le, i(year) j(group) string
```

The string part is because the parts that follow `le` in the variable names are not numbers, but strings, so `group` must be a string variable.

If you now enter

```
twoway scatter le year if group == "_male" || scatter le year if group
== "_female"
```

you should get the same graph as we have seen earlier.

Bibliography

[1] S. Greenland, S. J. Senn, K. J. Rothman, J. B. Carlin, C. Poole, S. N. Goodman, and D. G. Altman. Statistical tests, P values, confidence intervals, and power: a guide to misinterpretations. *European Journal of Epidemiology*, 31(4):337–350, 2016.

A. Old Stata Syntax for Categorical Variables

Before Stata had factor variables, there was a different process for generating indicator variables. I do not recommend *using* the old method, but it is probably worth being a little familiar with it, since you may well see it in existing do-files.

The main difference was that the old way required a prefix to any command using indicator variables with `xi:` (short for "eXpand Interactions". The `xi` command would generate a series of indicator variables, beginning with `_I`, and include them in the command in place of the categorical variable.

The second main difference was that the symbol for creating an interaction was `*` rather than `#`. So the interaction between two categorical variables `var1` and `var2` would be written as `i.var1*i.var2`, rather than `i.var1#i.var2`. There was no equivalent of the `c` prefix, so if `var2` was continuous, the interaction would be written `i.var1*var2`, rather than `i.var1#c.var2`.

Another major difference is that string variables could be used with `xi`, but only numeric variables can be used as factors. This is not a major drawback, since Stata 14 can show string labels for the levels of the factors in the output, rather than the numbers themselves (I prepared my notes with Stata 12, so only the numbers appear in my output).

A final difference is in how the reference category is selected, which we will meet in a few weeks. The new method is relatively intuitive and simple, the old method I still need to look up the details of the syntax 20 years after I first used it. Not that intuitive.

The differences between the two syntaxes are outlined in Table A.1.

	New syntax	Old Syntax
Prefix	none required	`xi:`
Variable type	Numeric	String or numeric
Interaction	#	*
Creates new variables	No	Yes

Table A.1.: Differences between old and new syntaxes for defining indicators

For full details, see `help fvvarlist` for the new syntax and `help xi` for the old syntax.

B. Stata Crib Sheet

B.1. Data Management

append Add additional records from a new dataset to an an existing dataset, provided that the two datasets contain the same variables (Page 27).

browse Open a spreadsheet view of the data, which can be viewed but not modified (Page 31).

clear Drop all observations, variables, constraints, and a few other things (Page 25).

describe Gives the names, types and labels of all (or some) variables (Page 209).

drop Drop some (or all) observations or variables (Page 24).

edit Open a spreadsheet view of the data, which can be viewed and modified (Page 31).

egen Create a new variable, using one of a wide variety of functions built into `egen` (Page 23).

expand Create duplicates of each record in the dataset (Page 217).

generate Create a new variable, using standard maths expressions or the basic functions built into Stata (Page 22).

keep Drop some observations or variables: the observations or variables specified are *not* dropped (Page 24).

label define Define a set of labels for the values of a categorical variable (Page 23).

label values Apply a previously defined label to a variable (Page 23).

label variable Label a variable to explain its meaning (Page 23).

merge Join two datasets which contain different variables, such that each record in the second dataset is matched to the corresponding record in the first dataset (Page 28).

notes Add a comment to the dataset which will appear when the command `describe` is used (Page **??**).

orthpoly Generate orthogonal polynomials (useful in polynomial regression) (Page 136).

predict Generate a new variable based on the results of a regression command (Page 93).

rename Change the name of a variable (Page 27).

replace Change the values of a variable (Page 22).

reshape Switch a dataset from long form to wide form and vice versa (Page 215).

save Save the current dataset as a stata file (Page 26).

saveold Save the current dataset as a stata 7.0 file (useful with Stat Transfer) (Page 26).

sysuse Use one of stata's built in datasets (Page 25).

use Use a dataset (Page 25).

B.2. Descriptive Statistics

codebook Provides information about the types of each variable and the values it takes (Page 209).

corr Calculates the Pearson correlation coefficient for two variables (Page **??**).

cs Calculates relative risk and risk difference for a cohort study with a dichotomous outcome and a single dichotomous predictor (Page 79).

graph box Produces a box and whisker plot (Page 46).

histogram Draws a histogram (Page 46).

list Lists the values of some or all variables in some or all observations (Page **??**).

sts list Produces a life-table for data previously defined as survival data using `stset` (Page 197).

sts graph Draws a Kaplan-Meier curve for data previously defined as survival data using `stset` (Page 197).

summarize Provides summary statistics for numeric variables (Page 209).

tabstat Produces a table of summary statistics (Page **??**).

tabulate Provides a frequency table for one or two categorical variables (Page 210).

B.3. Regression Models

constraint Define a constraint on the parameters of a regression model (Page **??**).

fracpoly Use fractional polynomials to fit a regression model (Page 136).

glm Fit a generalised linear model (Page **??**).

lincom Calculate a linear combination of the coefficients of a regressioni model (Page 118).

logistic Fit a logistic regression model (dichotomous outcome) (Page 149).

logit Fit a logistic regression model (dichotomous outcome) (Page 149).

mlogit Fit a multinomial regression model (categorical outcome) (Page 169).

nbreg Fit a negative binomial regression model (for count data, in which the assumption of a Poisson distribution is not safe) (Page **??**).

ologit Fit an ordinal logistic regression model (Page 176).

omodel Fit an ordinal logistic regression model with a test of the assumption of an ordinal association (Page **??**).

poisson Fit a Poisson regression model (count outcome) (Page **??**).

regress Fit a linear regression model (normally distributed outcome) (Page 92).

stcox Fit a Cox (proportional hazards) regression model (survival data) (Page 197).

testparm Test whether a given coefficient (or set of coefficients) significantly improve the fit of a regression model (Page 127).

ttest Performs a t-test (Page **??**).

xi: (eXpand Interactions) Used when fitting categorical predictors in a regression model (Page **??**.

B.4. Regression Diagnostics

cprplot Draws a component-plus-residual plot (partial-residual plot). Useful for testing whether the association between a predictor and the outcome of a linear model is really linear (Page 101).

hettest Formal test that the variance of the residuals from a linear model have constant variance (Page 99.)

estat gof When run after `logit` or `logistic`, performs the Hosmer-Lemeshow test for goodness of fit of a logistic regression model (Page 157).

estat gof When run after `poisson`, provides a goodness of fit test for Poisson regression (Page **??**).

lroc Draws a Receiver Operating Characteristic curve following a logistic regression model, to test the goodness of fit of the model (Page 158).

lvr2plot Draws a leverage-versus-residual plot: can be used to identify influential observations in a linear regression model (Page 103).

ovtest Performs an omitted variable test: tests whether the linear association between the predictors and outcome in a linear model is adequate, or whether there are non-linearities (Page 99).

qnorm Produces a normal plot (graphical test of normality) (Page 104).

rvfplot Residual versus fitted value plot: can be used to detect non-linearity (Page 99).

stcoxkm Draws the observed Kaplan-Meier curve and that expected based on a Cox regression model. If they are similar, the regression model is adequate (Page 197).

stphplot Graphical test of the proportional hazards assumption in Cox regression (Page 197).

stphtest Formal test of the proportional hazards assumption in Cox regression (Page 197).

sw Perform stepwise regression (Page 132).

swilk Swilk's test of normality: can be applied to residuals from a linear model (Page 104).

B.5. Utility Commands

by: Produce output for different groups of subjects separately, provided the dataset is sorted correctly (Page 212).

bysort: Produce output for different groups of subjects separately, sorting the dataset first if necessary (Page 212).

capture Get the error code from the command, but do not stop running if an error occurs (Page **??**).

cmdlog Create a command-log file: like a log file, but only contains commands, not output (Page 218).

display Show in the results window: can be used as a simple calculator (Page 217).

do Run a do file, showing all of the output as it runs (Page 18).

exit Leave stata (Page **??**).

foreach Repeat a command or set of commands, allowing one factor to vary between them (Page 214).

forvalues Repeat a command or set of commands, allowing one factor to vary between them (Page **??**).

global Define a global macro (Page 20).

help Get the help for a given command in the output window (Page 14).

local Define a local macro (Page 20).

log Start or stop a log file (Page 19).

net Retrieve information or programs from the internet (Page **??**).

preserve Store a snapshot of your dataset so that you can `restore` it if it is changed in way that you did not intend (Page 31).

restore Restore a previously `preserve`d dataset (Page 31).

sampsi Calculate sample size and power of a study (Page 80).

search Look through the help files for a given keyword (Page 14).

sort Sort the dataset on a variable or set of variables (Page **??**).

stset Define the data to be survival data (Page 197).

stjoin Rejoin a survival dataset that has been previously `stsplit` (Page 197).

stsplit Split a survival dataset up so that each record may become several records (Page 197).

webseek Search certain stata sites on the internet for a keyword (Page **??**).

whelp Get the help for a given command in a separate window (Page **??**).

B.6. Operating System Commands

cd Change to a given directory (Page 20).

dir List the files in the current directory (Page 20).

mkdir Create a new directory with the given name (Page 20).

pwd Give the name of the current directory (Page 20).

shell Create a command-line shell (Page 20).