Stata: Statistical, Survey and Graphical Analysis

How to Use This Course Book

This handbook accompanies the taught session for the course. Each section contains a brief overview of a topic for your reference and some sections are followed by exercises.

The Exercises

Exercises are arranged as follows:

- A title and brief overview of the tasks to be carried out;
- A numbered set of tasks, together with a brief description of each;
- A key at the back of the course book.

Some exercises, particularly those within the same section, assume that you have completed earlier exercises. Your lecturer will direct you to the location of files that are needed for the exercises. If you have any problems with the text or the exercises, please ask the lecturer or one of the demonstrators for help.

This book includes plenty of exercise activities – more than can usually be completed during the hands-on sessions of the course as well as some tasks that can be performed as a homework. These are clearly outlined throughout the course book.

Writing Conventions

Certain conventions are used to help you to be clear about what you need to do in each step of a task.

- Stata commands are presented with a small font on a new line similarly to the official Stata syntax conventions.
- A button to be clicked will look like this.

Objectives

From this course book you should:

- Be familiar with univariate and multivariate analysis commands
- To able to use regression analysis in Stata
- Be familiar with survey commands in Stata
- Be able to produce a variety of Stata graphs

Software Used

STATA 13

Files Used

bhps_for_class.dta

Stataanalysisandgraphics.do

Revision Information

Version	Date	Author	Changes made
1.0	July 2007	Adam Whitworth & Kate Wilkinson	Created
1.9	Sept 2008	Neli Demireva	Revised and Corrected
2.4 2.5	March 2013 July 2013	Neli Demireva Ladislav Kozak	Major Revisions Revised and Corrected
3.3	January 2016	Ines Rombach	Revised and Updated

Copyright

The copyright of this document lies with Oxford University IT Services.

Useful Information

ITLP Portfolio: http://portfolio.it.ox.ac.uk

Contents

1 Univariate and Multivariate Analysis

Stata's flexibility in statistical analysis is one reason why it has become a popular software package. The range of statistical analyses which Stata can perform is vast and this section covers just some examples of the most commonly used statistical commands in Stata and we will look both at syntax and at interpreting the output. This section covers the following:

Correlation

T-tests

Chi-square tests

Linear regression

Logistic regression

Post-estimation commands & overview of more advanced regression models

1.1. Correlation: Listwise correlation

Correlations analyse the extent to which changes in the values of one **interval** level (i.e. continuous) variable correspond to changes in the values of one or more other interval level variables. More precisely, correlation coefficients measure the strength and direction of the linear relationship between data points of two variables (how closely the data follows a straight line trend). It can be imagined as scattering the data points of the two variables, then fitting the line that represents the best fit to the data. The correlation coefficient stands for the direction and slope of that line. It can also be imagined as an estimate of the value of one variable based on the value of the other variable. Correlation coefficients range between -1 to +1 where:

-1 indicates a perfect negative relationship (e.g. if we were correlating age and income then a perfect negative relationship would be where age goes up by one unit then income goes down by one unit)

+1 indicates a perfect positive relationship (e.g. if we were correlating age and income then a perfect positive relationship would be where age goes up by one unit then income also goes up by one unit)

and 0 indicates that there is no correlation at all between the two variables.

Hence, correlation coefficients tell us firstly about the direction of the linear relationship between the variables (i.e. positive or negative) and secondly about the strength of that relationship (i.e. ranging from 0 to +/- 1, where values closer to 0 denote a weaker correlation). If we imagined these data to be plotted on a scatter then a stronger correlation coefficient would suggest that the points lie closer to the line of best fit (i.e. they have a tighter distribution) whereas a correlation coefficient close to 0 would suggest that the data points are fairly

randomly distributed at quite some distance from the line of best fit (which, clearly, would not fit the data very well).

The syntax to calculate the correlation coefficient between two variables is simple:

```
correlate varname1 varname2
```

For example, assume we wanted to calculate the correlation between total household income and labour income in our dataset then the syntax would be:

```
correlate tot_hh_inc inc_lab
```

The output looks like this:

```
. correlate tot_hh_inc inc_lab
(obs=8181)

             | tot_hh~c  inc_lab
-------------+------------------
  tot_hh_inc |   1.0000
     inc_lab |   0.7842   1.0000
```

It can be seen that the correlation between total household income and labour income is 0.7842 and that the correlations between each variable and themselves is (of course) 1. Focussing on the correlation between tot_hh_inc and inc_lab, the value of 0.7842 tells us that the two variables are positively correlated (the value is greater than zero and so when the value of one variable increases then the value of the other variable also tends to increase) and that this is a relatively strong relationship. This is an unsurprising result in this case as labour income (inc_lab) is a component element of total household income (tot_hh_inc), and is indeed the largest income source within the total household income variable which would explain the direction and strength of the correlation.

It is possible to run correlations with more than two variables and also with a by group if desired. For example, if we had individual level data then to correlate income and age for men and women separately we could run the following:

```
bys gender: correlate income age
```

1.2. Pairwise correlation (pwcorr)

Correlation can also be done '*pairwise*' using the pwcorr command rather than the correlate (or corr) command as above. If we used pwcorr then the output would be interpreted in just the same way and the only difference would be in the way that missing values are handled. For example, if we wanted to find correlation coefficients between four variables

```
correlate var1 var2 var3 var4
```

and half of the cases of var3 were missing but no cases for any of the other variables were missing then using correlate would result in the correlation only including half of the cases. This is because the correlate command uses listwise deletion, meaning that any case which has a missing value for any of the variables specified in the correlation command is excluded from all of the requested correlations, even those not involving the variable with the missing data.

By contrast, *pwcorr* **gives pairwise correlation coefficients and carries out pairwise deletion of missing values**. This means that a pair of data points are deleted from the calculation only if one (or both) of the data points in that pair are missing. In the example above, this would mean that any correlations involving var3 would be done with half of the cases but that any correlations not involving var3 (e.g. the correlation between var1 and var2) would involve the whole dataset as neither of these variables have any missing values. Clearly using pairwise correlation (pwcorr) can mean that different data are used for the different correlation coefficients calculated i.e. **the data are 'maximised' in the sense that they are used in the calculations where possible, whilst using listwise deletion (correlate) guarantees that the same data are used in all of the specified correlations.** There are no rules to say which is the right approach to use and it will depend upon the data and the research question in each situation.

Additionally, pwcorr enables significance tests to be carried out for each entry in the correlation table, by specifying options, and also allows the number of cases to be reported. For example, to run a pairwise correlation between total household income, labour income and investment income which reports both statistical significance and the number of cases for each two-way correlation in the table we would type:

```
pwcorr tot_hh_inc inc_lab inc_inv, sig obs
```

This would produce the following output which shows that all of the correlations are statistically significant (all have p-values of 0.000) and that there are 421 cases with missing data for labour income and/or investment income:

```
. pwcorr  hhsize inc_lab inc_inv, sig obs

             |   hhsize  inc_lab  inc_inv
-------------+---------------------------
      hhsize |   1.0000
             |
             |     8602
             |
     inc_lab |   0.3358   1.0000
             |   0.0000
             |     8181     8181
             |
     inc_inv |   0.0280   0.1103   1.0000
             |   0.0114   0.0000
             |     8181     8181     8181
             |
```

The star option can also be specified to place an asterisk by significant correlations according to the desired level of significance, help pwcorr.

1.3. Verifying appropriateness

Correlation coefficients are based on an assumption of a linear relationship between the variables and where the variables appear to be associated but in a non-linear way then correlation coefficients are arguably not the most appropriate calculation to use. The data should therefore be investigated to verify the assumption of linearity. This can be done using scatter plots which are discussed later on in this course book. Additionally, correlate assumes that the data has a normal distribution. Non-parametric correlation methods such as Chi-square, Speaman's rho and Kendall's tau may be more appropriate when distributions are not normal. For instance, if the data are ranked or if there are outliers then we could use the Spearman's rank correlation, which will automatically rank the data if it is not already ranked. The syntax is simple and the output is interpreted in a similar way to the above:

```
spearman varname1 varname2
```

1.4. T-test

T-tests are used with **interval** level (i.e. continuous) variables to estimate whether the difference between two values can be considered statistically significant. It is commonly used to test whether the difference between two means is statistically significant and can be used in a few slightly different ways.

1.5. One-sample t-tests

A one-way t-test tests if the mean of a variable equals a particular value. For example, the median value of total household income (tot_hh_inc) is £30,930. We can use ttest to test whether the mean is significantly different to this value,

```
. ttest tot_hh_inc==30930

One-sample t test
```

Variable	Obs	Mean	Std. Err.	Std. Dev.	[95% Conf.	Interval]
tot_hh~c	8602	35174.27	295.3474	27392.58	34595.32	35753.22

```
    mean = mean(tot_hh_inc)                                       t =  14.3704
Ho: mean = 30930                                 degrees of freedom =     8601

  Ha: mean < 30930               Ha: mean != 30930                 Ha: mean > 30930
 Pr(T < t) = 1.0000         Pr(|T| > |t|) = 0.0000          Pr(T > t) = 0.0000
```

In the output above, the first line tells us the number of observations used, the mean and its standard error, standard deviation, and the 95% confidence intervals. On the left, Ho: mean lists the value against which we are testing the actual mean of total household income, and Ho relates to the null hypothesis which is being tested (i.e. in this case the null hypothesis is that the mean household income equals the median value of 30930). At the bottom of the output are reports relating to whether the actual mean (35174) can be said to be different to the median (30930) with statistical significance at the 5% level. At the bottom left Stata reports that there is no statistical evidence that the mean is less than the median (as one would expect). The other two hypotheses are both statistically significant at the 5% level, however, which suggests that there is statistically significant evidence at the 5% level both that the mean is not equal to the median (bottom centre) and that the mean is greater than the median (bottom right). **This analysis therefore suggests that the mean is larger than the median and that this difference is statistically significant at the 5% level.** As with many of Stata's statistical commands, the 'level' option can be used with ttest to change the confidence level specified.

1.6. Two-group T-tests (with by-groups)

The t-test can also be used to compare the means of two by-groups. For example, if we had individuals' income and we had a binary gender variable then we could test if the mean income of men and women was significantly different from one another. Our data is at household rather than individual level and so we will instead test whether the total income of households is significantly different between households with and without a garden. The syntax to do so is simple, although note that besides specifying the by group we add no further options:

```
ttest tot_hh_inc, by(garden)
```

The output looks as follows:

```
. ttest tot_hh_inc, by(garden)

Two-sample t test with equal variances
```

Group	Obs	Mean	Std. Err.	Std. Dev.	[95% Conf.	Interval]
yes	7630	37615.95	311.026	27168.08	37006.26	38225.65
no	550	28288.77	871.1529	20430.35	26577.57	29999.97
combined	8180	36988.82	297.0812	26869.02	36406.46	37571.17
diff		9327.187	1181.854		7010.452	11643.92

```
    diff = mean(yes) - mean(no)                                   t =   7.8920
Ho: diff = 0                                     degrees of freedom =     8178

    Ha: diff < 0                 Ha: diff != 0                 Ha: diff > 0
 Pr(T < t) = 1.0000         Pr(|T| > |t|) = 0.0000          Pr(T > t) = 0.0000
```

The output shows the number of cases, mean, standard error, standard deviation and confidence intervals for the two groups - 'yes' (have a garden) and 'no' (do not have a garden) - as well as for the two groups combined, and sets out differences between the two groups on the 'diff' row. On the left (underneath the table), the calculation of difference is set out: this shows that for the purposes of the test the 'difference' will be taken as the mean income of the 'yes' (do have a garden) group minus the mean income of the 'no' (do not have a garden) group.

Below, the null hypothesis H_o (there is no difference between the two groups) is stated, and this is what we are testing.

Along the bottom of the output are the main results. At the left, it can be seen that there is no evidence in a statistical sense that the difference is negative – i.e. that the mean income of the households without a garden is larger than the mean income of households with a garden (as this is how the equation is specified). Next, the output shows that there is statistically significant evidence at the 5% level that the difference between the mean incomes of the two groups is not equal to zero. To the right, the output shows that there is evidence (statistically significant at the 5% level) that the difference between the mean incomes of the two groups of households is positive i.e. that the mean income of households with gardens is higher than the mean income of households without a garden and that this difference is statistically significant at the 5% level.

NB: Equal or unequal variances?

By default ttest assumes that the variables have equal variances. If you have reason to believe that the variances are not equal then the unequal option should be specified.

1.7. Two-sample t-tests

Finally, a two-way t-test performs a test of the difference of the means of the two variables. For example, to test if the mean of labour income of households is significantly different from their total income we would type:

```
ttest inc_lab==tot_hh_inc
```

The output can be analysed as discussed above. In this example, the equation being tested is the difference between the two means which is defined as the mean of labour income (inc_lab) minus the mean of total household income (tot_hh_inc), with the null hypothesis (H_0) being that the mean difference is zero. The results along the bottom suggest that there is evidence to suggest that the difference is negative – i.e that total household income is larger than labour income – and that this evidence is statistically significant at the 5% level. This result is unsurprising given that labour income is one of five income sources which makes up the total household income variable.

```
. ttest inc_lab==tot_hh_inc

Paired t test
```

Variable	Obs	Mean	Std. Err.	Std. Dev.	[95% Conf.	Interval]
inc_lab	8181	21652.16	290.3101	26258.23	21083.08	22221.24
tot_hh~c	8181	36984.36	297.0783	26870.4	36402.01	37566.71
diff	8181	-15332.2	193.0518	17461.32	-15710.63	-14953.77

```
     mean(diff) = mean(inc_lab - tot_hh_inc)                      t = -79.4202
 Ho: mean(diff) = 0                                     degrees of freedom =     8180

 Ha: mean(diff) < 0           Ha: mean(diff) != 0           Ha: mean(diff) > 0
 Pr(T < t) = 0.0000         Pr(|T| > |t|) = 0.0000          Pr(T > t) = 1.0000
```

1.8. Paired (dependent samples) or unpaired (independent samples) t-tests?

By default ttest assumes that data are paired. This is sometimes also referred to as a dependent samples t-test. Paired data are typically where two values are taken for the same units of analysis (e.g. individuals, households) at different points in time (e.g. past and present exam scores). As the two values relate to the same entity (e.g. same individual) then the data are assumed to have some degree of correlation between the two variables, hence the reference to them being dependent samples (i.e. one score is dependent (at least in part) to the other score). The 'unpaired' option indicates that the two variables being tested should are from unrelated units of observation (e.g. different people or households).

1.9. Testing the difference of means across multiple comparison groups (oneway)

If we wished to test the difference between the means of some interval level variable (e.g. total household income) across more than two categories of another variable (e.g. house tenure) then the oneway command can be used to do this. If we had a large number of levels of the grouping variable then loneway might be more suitable, and the two produce slightly difference statistics in their output. It would be possible to use ttest to test each pair of categories in turn but oneway will do this more conveniently within a single command. For example, in the data we have five categories of household tenure and we wish to compare mean total household income between the five categories and to test if there are statistically significant differences between the five tenure types. Let us firstly have a quick look at the means of the groups:

```
table tenure, c(mean tot_hh_inc)
```

which lists the means for the different tenure types:

```
. table tenure,c(mean tot_hh_inc)

------------------------------------------
house owned or |
rented         | mean(tot_hh~c)
---------------+--------------------------
owned or on mortgage |        38447.63
    shared ownership |        31281.73
              rented |        26093.47
           rent free |        27516.23
               other |        31606.87
------------------------------------------
```

This suggests that there may be differences in total household income between the tenure types and we can use oneway to test the statistical significance of these apparent differences:

```
oneway tot_hh_inc tenure
```

```
. oneway tot_hh_inc tenure

                        Analysis of Variance
    Source              SS         df      MS            F     Prob > F
------------------------------------------------------------------------
Between groups      2.5137e+11      4   6.2843e+10     87.10     0.0000
 Within groups      6.2024e+12   8597    721463247
------------------------------------------------------------------------
    Total           6.4538e+12   8601    750353427

Bartlett's test for equal variances:  chi2(4) =  42.2480  Prob>chi2 = 0.000
```

The above output is a standard analysis-of-variance (ANOVA) output and shows the group sum of squares for the model (SS), its degrees of freedom (df) and the mean square figure (MS) which equals SS/df. The corresponding F statistic (F) is reported, as is its significance level (Prob>F). The output above shows that the model is statistically significant at the 1% level. At the bottom of the table, Bartlett's test for equal variances is reported and this is significant at the 1% level in this output, meaning that we can reject the null hypothesis that the variances are equal. This means that the assumptions underlying the ANOVA test (e.g. equal variances across the populations) are not satisfied and the test may not be entirely appropriate.

The output confirms our suspicion from the table we first created that there are differences in total household income between tenure types: the output from the oneway command provides evidence of differences in means which are statistically significant at the 1% level. In order to look more closely at the significance of differences between each of the tenure types additional options can be specified to provide more detailed output:

```
oneway tot_hh_inc tenure, bonferroni
```

Here we request that the Bonferroni multiple-comparison test is also reported (alternative options include the Scheffe or Sidak tests):

The same output as above is reported, along with more detailed output:

```
                Comparison of tot_hh_inc by house owned or rented
                                    (Bonferroni)
Row Mean-
Col Mean    |  owned or   shared o     rented   rent fre
------------+--------------------------------------------
shared o    |  -7165.89
            |     0.968
            |
  rented    |  -12354.2   -5188.27
            |     0.000      1.000
            |
rent fre    |  -10931.4    -3765.5    1422.77
            |     0.000      1.000      1.000
            |
   other    |  -6840.76    325.132     5513.4    4090.63
            |     1.000      1.000      1.000      1.000
```

In the output above, the top left hand corner relates to the difference between tenure types 'rent free' and 'owned' (labelled Row Mean-Col Mean in the upper left of the output i.e. the values on the top row of the cells represents the Row Mean subtract the Column Mean). Thus, the figure of -10931.4 in the top left corner of the table represents the mean total household income of houses which are lived in rent free minus the mean total household income of houses which are owned. Looking back to the table we produced earlier, this figure of -10931.4 can be seen to equal the 27516.23 - 38447.63 reported in the table for these two

groups, where 27516.23 equals the mean of the category listed in the relevant row (rent free mean total household income) and 38447.63 equals the mean of the subgroup listed in the relevant column (total household income of owned houses). The output also shows underneath the figure of -10931.4 a Bonferroni adjusted p-value of 0.000 i.e. this particular difference in means is statistically significant at the 1% level based upon the Bonferroni-adjusted significance of the difference. Looking through the results in the table, the output suggests that households which own their houses have the highest total household income, although the differences between owned, shared and unspecified houses are not statistically significant.

Rather than using the oneway command these analyses could have been done using anova. Oneway is simpler to use than anova for one-way ANOVA models and allows multiple comparison tests such as this. Anova enables more powerful postestimation commands (e.g. generating predictions, running underlying regressions and postestimation commands).

1.10. One-sided/one-tailed t-tests

Stata, by default, always reports two-tailed t-tests. If you want the p-value for a one-tailed test then it is necessary to simply divide the p-value reported by two or (this is much more accurate if you access the full p-value from the saved results). This latter method gives precision to several decimal places. The Stata help provides information on how to access saved results.

1.11. Verifying appropriateness

The use of the t-test is valid only if certain underlying assumptions hold. Most importantly, the t-test assumes that the sample mean is a valid measure of centre, which is not a problem if the variable of interest is interval level but is a problem with ordinal level variables as the distance between values is arbitrary. Even when the sample mean can be assumed to be a valid measure of centre, a t-test assumes that the underlying distribution of the variables is approximately normal and if this is not the case then non-parametric tests (e.g. Mann-Whitney, Wilcoxon or Kolmogorov-Smirnov) rather than parametric tests (e.g. t-test) may be preferable. Common ways to test for normality include plotting histograms, boxplots and the normal Q-Q plot (qnorm), the Shapiro-Wilk W test (swilk), and tests of skewness and kurtosis (sktest).

1.12. Chi-square

Whilst t-test is used when the variable of interest is an interval (continuous) variable, the Chi-square test is used when the variable of interest is categorical. In

order to run the Chi-square test we simply add it as an option within the tabulate command:

```
tabulate tenure garden, chi2 exact
```

```
. tabulate tenure garden, chi2 exact

Enumerating sample-space combinations:
stage 5:  enumerations = 1
stage 4:  enumerations = 26
stage 3:  enumerations = 933
stage 2:  enumerations = 50691
stage 1:  enumerations = 0
```

house owned or rented	accom: has terrace/garden yes	no	Total
owned or on mortgage	5,747	229	5,976
shared ownership	32	4	36
rented	1,724	302	2,026
rent free	103	14	117
other	24	1	25
Total	7,630	550	8,180

```
          Pearson chi2(4) = 302.4933   Pr = 0.000
           Fisher's exact =                 0.000
```

This output reports observed frequencies in each cell and at the bottom of the output reports that the differences seen are statistically significantly different to the frequencies that would be expected were there no association between the variables. Additional options can be included in the tabulate command to also report relative column frequencies (col), relative row frequencies (row), expected frequencies (expected), to suppress frequencies (nofreq), and to show the contribution of each cell to the overall Chi-square (cchi2). As well as Pearson's Chi-squared (chi2), other tests of association can also be reported with extra options: Cramer's V (V), Fisher's exact test (exact), Goodman and Kruskal's gamma (gamma), Kendall's tau (taub), and the likelihood-ratio Chi-squared (lrchi2). All of these different tests can be reported by default with the all option (all).

The exact option is used when the expected counts are below 5.

Whilst the above output provides evidence that there is a statistically significant association between the type of tenure and the property having a garden/ terrace (statistically significant at the 0.01 level, i.e. the p-value of the Fisher's exact test is significant at the 1% level), it does not provide any evidence as to which individual cells shows evidence of having observed frequencies which are significantly different to their expected frequencies. This can be done by adding additional options to the command, including the specification of the expected values. More options can be found using the help tabulate command.

Exercise 1 Univariate and multivariate commands (15 mins)

- *Use the bp_data.dta saved in "H:\StataLevel3\Raw data" (fictional data)*
- *Practice the use of basic univariate analysis commands*

Task 1

- For the entire sample, what is the correlation between the baseline and follow-up blood pressure (i.e. bp_before and bp_after)?
  Use both the corr and pwcorr commands to perform this task.
- What does the coefficient mean?
  Hint: Details on the relevant commands are described in sections 1.1 and 1.2.
- Adjust the code to find out if the correlation is statistically significant, and how many observations are included into the summary.
  Hint: these options can only be used for one of the commands.

Task 2

- Using the chi-squared test, find out if there is an association between the allocated treatment (trt) and low blood pressure (low_bp).
  Hint: the chi-squared commands are introduced in section 1.12.
- Extend the command to include the expected frequency in each cell.
  Is the number of observed patients with low blood pressure randomised to drug A higher or lower than expected, assuming no association between the variables?

Task 3

- Use a paired t-test to compare the baseline blood pressure values (bp_before) to the follow-up blood pressure values (bp_after).
  Hint: details on the two-sample t-test command are provided in section 1.7. Here, you need to use the command for the two-sample t-test because the data is saved in separate variables. However, remember that the data is paired.
- Is there any evidence to suggest the mean blood pressure values are different before and after the intervention? If so, at which time point is the blood pressure higher?

Task 4

- Now use a t-test to compare the follow-up blood pressure values by treatment group. In which group are the blood pressure values higher?
  Hint: all relevant outcome data is now stored in one variable. Section 1.6 provides more information on the two-group t-test with by-groups.

Task 5

- Use the appropriate command to find out if there is statistical evidence that the follow-up blood pressure differs by age group (agegrp), using a 5% significance level.
  Hint: details of the oneway command are provided in section 1.9. Remember that the continuous variable needs to be listed first in this command.
- Is there evidence of unequal variances between the groups?
- Finally, use the Bonferroni adjustment to find out where potential differences are.

1.13. Ordinary Least Squares (OLS) linear regression

Regression analysis is a useful tool in that it isolates the independent predictive power of individual independent (explanatory) variables holding other variables in the model constant. There are many types of regression analysis, each suited to particular data characteristics and analyses.

OLS is a common regression type and is used when the dependent variable of interest is an interval variable such as height or income which has a continuous scale, and where it is appropriate to use OLS in terms of the assumptions which underpin it, namely: (i) linearity of the relationship between dependent and independent variables; (ii) independence of the error terms; (iii) homoscedasticity (i.e. constant variance of the error terms); (iv) normality of the distribution of the error terms. The data should be checked prior to running an OLS regression to see that it satisfies these assumptions[1].

In order to fit a linear OLS regression in Stata,19 the generic syntax structure is that the regress command starts the line and this is followed by the dependent variable and then the explanatory variables which are to be included in the model:

```
regress depvar indvar(s)
```

For example, expenditure on food (exp_food) is an interval variable in our dataset. To run a regression with expenditure on food as our dependent variable (interval variable) and total household income and number of rooms as two independent variables (both interval variables) the syntax would be:

```
regress exp_food tot_hh_inc rooms
```

```
. regress exp_food tot_hh_inc rooms

      Source |       SS       df       MS              Number of obs =    8171
-------------+------------------------------           F(  2,  8168) = 1479.28
       Model |  110755790     2  55377895.2             Prob > F      =  0.0000
    Residual |  305774326  8168  37435.6422             R-squared     =  0.2659
-------------+------------------------------           Adj R-squared =  0.2657
       Total |  416530116  8170  50982.8784             Root MSE      =  193.48

------------------------------------------------------------------------------
    exp_food |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
  tot_hh_inc |   .0022568   .0000852    26.49   0.000     .0020898    .0024238
       rooms |   47.37758   1.352297    35.03   0.000     44.72674    50.02843
       _cons |   331.2274   6.267147    52.85   0.000     318.9422    343.5126
------------------------------------------------------------------------------
```

[1] Useful resources on how to run regression diagnostics can be found at statistics.ats.ucla.edu/stat/stata/webbooks/reg/default.htm (Chapter two in particular) and http://www.duke.edu/~rnau/testing.htm

Starting at the top left of the output, total variance is made up of variance that is explained by the independent variables in the model plus that which is not explained by the model (residual or error). Conceptually, the total sum of squares (SS Total) is the total variability around the mean $S(Y - Ybar)^2$, the residual sum of squares (SS Residual) is the sum of squared errors in prediction $S(Y - Ypredicted)^2$, and the model sum of squares (SS Model) is the improvement in prediction by using the predicted value of Y compared with just using the mean of Y. Hence, SSTotal = SSModel + SSResidual. Additionally, SSModel / SSTotal = R^2 (0.2659) and this is because R-Square is the proportion of the variance in the outcome variable which is explained by the independent variables in the model. Df relates to the degrees of freedom in the model and the MS column shows the Mean Squares, which are the Sum of Squares divided by the respective degrees of freedom. These enable the F ratio to be calculated (Mean Square of the Model / Mean Square of the Residuals) and this tests the overall statistical significance of the model as whole.

At the top right of the output, the p-value associated with the F-value has a value of 0.000 and this leads us to conclude that the model is a better predictor of the dependent variable than taking its mean value. Typically, statistical significance at the 5% level (p-value less than 0.05) is taken as sufficient to accept a statistically significant association. The F-value and corresponding p-value relate to the model as a whole and so do not allow us to say anything about the statistical significance of any of the individual explanatory variables in the model. The R-Squared value is the proportion of the variance in the dependent variable (exp_food) which can be predicted from the independent variables, which in this example indicates that 26.59% of the variance in food expenditure can be predicted by the regression on total household income and the number of rooms in the house. In other words, the model explains only around a quarter of the variance in the dependent variable. Note that this is an overall measure of association and does not tell us how much contribution each independent variable makes to the predictive power of the model. As more explanatory variables are added to the model these will explain some of the variance in the dependent variable simply due to chance, meaning that the R-Squared value increases as we add explanatory variables (irrespective of their real predictive contribution). The adjusted R-Square is designed to take this into account and seeks to give a more accurate estimate of the R-Squared, particularly when many explanatory variables are used. The difference between the R-Squared and adjusted R-Squared will tend to be smaller when fewer explanatory variables are used.

Turning now to the coefficients for the explanatory variables, the column headed exp_food lists the dependent variable at the top (exp_food) and each of the explanatory variables in the model beneath, including the constant term which shows at what point the regression line cuts the Y intercept (in other words, the predicted value of the dependent variable when all other variables equal zero). Looking first of all at the column headed P>t, this shows a p-value for each explanatory variable, i.e. indicating whether the coefficient of that explanatory variable can be considered significantly different to zero. In this example, all of the explanatory variables are statistically significant at the 5% level and so for each one we can be confident in rejecting the null hypothesis that the coefficient is zero. Hence, we have statistical evidence from the model that each explanatory

variable has predictive power in relation to food expenditure. If we wanted to change the level of statistical significance used to something other than the default level of 95% then the level( ) option can be included. For example, to use the 99% level of significance we would simply add level(99) as an option in the regression command. This level option can be added to many statistical commands in Stata including, for example, regress, logistic and ttest.

Having established that the coefficients are statistically significant, the Coef column reports coefficients for each explanatory variable. These coefficients mathematically describe the relationship between each explanatory variable and the dependent variable, whilst holding constant all other variables in the model. Hence, the coefficients represent the values for the regression equation for predicting the dependent variable from the independent variables. They tell us the prediction of the amount that the dependent variable (food expenditure) would increase by if we had a one unit increase in that independent variable. Naturally, we should be extremely cautious about interpreting coefficients whose p-values are not statistically significant as we cannot say with any statistical confidence that these coefficients are not in fact equal to zero, and even where p-values for particular coefficients are significant at the 5% level it is still necessary to take into account not just the value of the coefficient but also of the range of the confidence intervals around this estimate. Therefore, the regression equation in this case would be:

Y = cons + b1*X1 + b2*X2...+ error

which in this model is:

Predicted food expenditure = 331.2274 + (.00226 * total household income) + (47.37758 * number of rooms) + error

In this example, the coefficients suggest that for every one unit (i.e. a one pound increase in this data) increase in total household income we see a small increase in food expenditure and that for a one unit increase in the number of rooms in the house we see a 47.3 unit increase in food expenditure. Of course, these estimates are based on controlling only for those variables which are included in the model: in this model only two main effects were specified and it is therefore very likely that these coefficients would change if we fitted a more complete model.

The standard errors associated with the coefficients are also reported and these are used both to form confidence intervals and to test the significance of each parameter. Finally, the confidence intervals are useful in that they set out the range within which the estimate of the coefficient might fall and yet still be within the 5% level of statistical significance. Hence, the tightness of the confidence interval is an important factor in analysing regression coefficients alongside analyses of the statistical significance and point estimate of the coefficient.

If we wished to be more conservative then we could add the robust option:

```
regress exp_food tot_hh_inc rooms, robust
```

This can be used with many different estimation commands and produces Huber-White robust estimates of the standard errors which are less sensitive to violation of assumptions relating to normality and homogeneity of variance of the residuals. If we did this then we would see in the output that the coefficients are unchanged, the overall F-value is lower and the standard errors are larger.

1.14. Automatically creating dummies for a categorical explanatory variable

In the above example, both explanatory variables were – like the dependent variable –interval level (continuous) variables and hence it was possible to simply include them in the model as explanatory variables without any additional code.

If we want to include a categorical (including binary) variable as an independent variable, this needs to be made clear in the code as otherwise Stata will treat the categorical data as continuous by default. This is done by pre-fixing categorical data with an i. in the regression coding. Note that this is not necessary for binary data that is coded as 0 and 1, but it is good practice to use the code consistently.

In terms of the syntax, if we continued the previous model example for food expenditure but also wanted to include the categorical variable relating to household type as an independent variable in the model above then we could type:

```
regress exp_food tot_hh_inc rooms i.house_type
```

Source	SS	df	MS		
Model	115393278	15	7692885.17	Number of obs	8171
Residual	301136839	8155	36926.651	F(15, 8155)	208.33
				Prob > F	0.0000
				R-squared	0.2770
				Adj R-squared	0.2757
Total	416530116	8170	50982.8784	Root MSE	192.16

exp_food	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
tot_hh_inc	.0022311	.000085	26.23	0.000	.0020644	.0023978
rooms	39.40337	1.591661	24.76	0.000	36.28331	42.52343
house_type						
semi-det'd house/bun	-6.612475	5.983773	-1.11	0.269	-18.3422	5.117245
end terraced house	-36.4402	8.816357	-4.13	0.000	-53.72251	-19.15789
terraced house	-23.09887	6.974416	-3.31	0.001	-36.77051	-9.427242
purpose blt flat<10	-68.42107	9.530692	-7.18	0.000	-87.10365	-49.73848
purpose blt flat >10	-80.20332	13.68671	-5.86	0.000	-107.0328	-53.37387
converted flat <10	-97.46997	16.18923	-6.02	0.000	-129.205	-65.73495
converted flat >10	-116.708	35.09705	-3.33	0.001	-185.5071	-47.9088
incl business prem	-47.74758	53.55774	-0.89	0.373	-152.7344	57.23924
bedsitter under10	-149.7722	61.06267	-2.45	0.014	-269.4706	-30.0738
bedsitter 10+ dwellg	-75.86939	53.63067	-1.41	0.157	-180.9992	29.2604
bedsitter single occ	-44.81009	72.95028	-0.61	0.539	-187.8112	98.19105
sheltered accommdn	-139.3155	32.89321	-4.24	0.000	-203.7945	-74.83639
other	-63.18846	23.04392	-2.74	0.006	-108.3604	-18.01649
_cons	390.4827	10.09922	38.66	0.000	370.6856	410.2798

Much of this output can be interpreted as in the above example, and it can be seen that including the categorical house type variable added only a very small amount to the R-Squared value of the model: this suggests that the house type variable itself does not explain very much of the variable in food expenditure and is not a particularly powerful predictor (although some of its categories have coefficients which are statistically significant). Focussing on the new coefficients reported which relate to house type, it can be seen that of the fifteen categories of house type, 14 are displayed in the output. Category one (detached house/bungalow) is the reference category and this has been excluded from the model. In terms of interpretation, the coefficients relating to each of the house type categories set out the extent to which food expenditure for a household of that house type would be predicted to change in relation to food expenditure of households living in detached houses/ bungalows.

In the output above, all of these coefficients are negative and where these are statistically significant this suggests that – controlling for the other variables included in the model - these household types spend less on food than the reference group of detached houses/bungalows.

Therefore, the statistically significant coefficient of -23.09 for house type four (terraced houses) means that a terraced household, on average, spends 23 units (in this case pounds sterling) less on food each month than the reference category (detached houses/bungalows) controlling for the other factors included in the model.

1.15. Changing the base/reference category of a categorical explanatory variable

The reference category is by default chosen to be the one with the lowest numeric code attached to it.

Luckily since Stata11, changing a reference category can happen on the go. If you put a prefix i. then the lowest category is the reference category. With a prefix b2.variable the reference category changes to the category which is coded 2.

So the above regression model output can be changed to use the category with value label 2 as the reference category by using the following command:

```
regress exp_food tot_hh_inc rooms ib2.house_type
```

The interpretation of the statistical output changes to reflect the new reference category.

1.16. Including interaction terms

In the previous example, the categorical house type variable was included in the model predicting expenditure on food. In this previous example, category one of the house type variable (detached houses/bungalow) was omitted from the model as it was the reference category by adding i. to the code. Each of the remaining categories of the house type variable were compared to the reference category.

The previous section discussed how the interpretation of the coefficients for these fourteen categories must be made in relation to the reference group for that variable. Therefore, repeating the example above, the statistically significant coefficient of -23.09 for house type four (terraced houses) means that a terraced household is predicted to spend 23 units (in this case pounds sterling) less than the reference category (detached houses/bungalows) controlling for other factors (i.e. keeping all other variables constant).

If we were to plot the regression line of feed expenditure based on total household income separately for these two household types then these would be plotted at two parallel lines, with lower food expenditure for households in terraced houses. The key point hereby is that the above regression model allows the position of the regression lines to change but does not allow the slopes of these lines to change: it assumes that the difference in food expenditure between households in detached houses vs. terraced house is always the same, regardless of the other variables.

Interaction effects are needed if we wish to allow the slope of the regression lines to vary.

For example, it is commonly found that the relationship between education and income is – for a number of reasons – not constant between men and women: on average, for the same level of education men tend to earn higher incomes than females, however, the difference in pay is not constant.

Hence, if we were fitting a model to seek to predict income, then we may not only wish to include main effects for income and sex but we may also wish to include an interaction term between income and sex in order to allow the regression slopes to vary by these terms, reflecting the fact that evidence suggests the slope of the lines will differ between men and women. By this it is meant that being a male or female may affect income (sex main effect), the level of education may affect income (education main effect), but also the extent to which education affects income will depend upon whether the person is male or female (education*sex interaction term). In this example, this interaction could easily be

included in the regression as an additional explanatory variable to test this hypothesis.

Let us assume that not only did we think that total household income and whether or not the house has a garden affected food expenditure but also that the relationship between total household income and food expenditure was not the same depending on whether or not a property has a garden. The following model can be fitted to test this hypothesis, adding an interaction term between the two explanatory variables to the model:

```
regress exp_food i.garden tot_hh_inc i.garden#c.tot_hh_inc
```

The following output is generated:

```
. regress exp_food i.garden tot_hh_inc i.garden#c.tot_hh_inc, level(99)
```

Source	SS	df	MS		
Model	70382486.5	3	23460828.8	Number of obs	= 8171
Residual	346147630	8167	42383.6941	F(3, 8167)	= 553.53
				Prob > F	= 0.0000
Total	416530116	8170	50982.8784	R-squared	= 0.1690
				Adj R-squared	= 0.1687
				Root MSE	= 205.87

exp_food	Coef.	Std. Err.	t	P>|t|	[99% Conf.	Interval]
garden						
no	-145.0567	15.53381	-9.34	0.000	-185.0785	-105.0349
tot_hh_inc	.0031714	.0000868	36.54	0.000	.0029478	.0033951
garden#c.tot_hh_inc						
no	.001599	.0004387	3.64	0.000	.0004686	.0027293
_cons	521.926	4.027888	129.58	0.000	511.5484	532.3035

Note that in this model we include the two main effects (total household income and the garden variable) and the interaction term separately, hence the explanatory variables of this model are tenure, total household income, and an interaction term between tenure and total household income.

Also note that in the interaction term, we have to clarify (as before) that garden is a categorical variable, but also that tot_hh_inc is a continuous variable (prefix c.).

Using `regress exp_food i.garden##c.tot_hh_inc` gives the same output. The `##` specifies that the full factorial of the variables, i.e. the main effects and all possible interactions are to be used in the statistical model.

In the older Stata syntax, it is only necessary to state the interaction term as this implies the inclusion of the main effects:

```
xi: regress exp_food i.garden*tot_hh_inc
```

However, the output used to be less intuitive, and does not provide the value labels for the different categories.

This output shows that the model is statistically significant (Prob>F =0.0000) and that it explains around 17% of the variance in food expenditure. Turning to the coefficients, the first explanatory variable in the model can be seen to relate to the dummy variables for the "no" category of the garden variable. This means that on average, households in properties without a garden spend around £145 less on food a month than those with a garden, keeping all other variables constant. These are the coefficients relating to the tenure status main effect. Next comes the coefficient for total household income which shows only a slight positive association between total household income and food expenditure when controlling for the other factors included in the model, albeit a statistically significant association. This is the coefficient relating to the total household income main effect, and it is a single coefficient because this is an interval level variable (whereas the garden variable, in contrast, is a categorical variable). Following this comes the coefficient relating to the interaction between garden and total household income: garden#c.tot_hh_inc relates to the interaction between garden (no) and total household income. Garden group one (yes) is again omitted as the reference category for the interaction terms, though this could be changed, as discussed above.

All of the coefficients are statistically significant at the 5% level.

It is important to note how the interpretation of the coefficients changes as a result of including the interaction terms in the model. In the previous model in which only main effects were specified the coefficient relating to total household income related to the association between total household income and food expenditure for all garden types: in this sense it could be considered a *global* parameter as it applies to every garden group. By including the interaction terms between total household income and garden, however, we are asking whether this global parameter for total household income across the different garden types actually applies or whether, each garden type should have its own coefficient (i.e. what can be understood as *local* parameters between each individual garden types and total household income). Therefore, in the output the coefficient relating to the total household income main effect relates to the association between total household income and food expenditure for households with a garden (as this is the reference group for the tenure type main effect) and not for all garden types. For all other garden types (i.e. here those without a garden), the interaction terms state what the coefficient between total household income and food expenditure is significantly larger. This means that for households without garden, each unit increase in total household income leads to a larger unit increase in food expenditure than is the case for the reference group (households with a garden). If we were to plot the regression lines this would lead to a regression slope for households without garden which at the intercept is about £145 below the regression line for households with garden, (given by the negative coefficient against the main effect for garden - no) but which also has a different slope to the line for households with gardens (given by the coefficient on the interaction term). In this case, the positive coefficient of the interaction term shows that the regression lines for house with and without garden will move closer together as total household income increases (i.e. the difference in food expenditure between households with and without garden is predicted to decrease as total household income increases).

- **Another way**

Stata handles factor (categorical) variables elegantly. You can put ## instead to specify a full factorial of the variables—main effects for each variable and an

interaction. If you want to interact a continuous variable with a factor variable, just prefix the continuous variable with **c.**. You can specify up to eight-way interactions.

1.17. Logistic regression

The logistic command is used when the dependent variable is a binary variable and estimates odds ratios; the logit command fits an identical model but reports beta coefficients rather than odds ratios. An alternative way to gain odds ratios is to use the logit command and to specify the 'or' option (requesting odds ratios).

The syntax is equivalent to that used with OLS. For example, assume we wanted to run a model with 'garden' as the dependent variable. Currently, garden equals 1 for houses with a garden and 2 for houses without a garden. Firstly, we would need to set up this variable as a binary (i.e. 0/1) variable as Stata expects a 0/1 binary for the dependent variable. Therefore we generate a new binary variable called garden2 where 'no – do not have a garden' equals 0 and 'yes – have a garden' equals 1 (the category of interest in the dependent variable should be set to equal 1 – normally this is the 'positive' category). We can now run the model, in which we predict this binary outcome variable with three interval level independent variables:

```
logistic garden2 inc_lab hhsize total_mortgage
```

The following output is produced:

```
. logistic garden2 inc_lab hhsize total_mortgage

Logistic regression                               Number of obs   =       3078
                                                  LR chi2(3)      =      48.77
                                                  Prob > chi2     =     0.0000
Log likelihood = -526.63821                       Pseudo R2       =     0.0443

------------------------------------------------------------------------------
     garden2 | Odds Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     inc_lab |   1.000006   4.53e-06     1.23   0.219     .9999967    1.000014
      hhsize |   1.644186   .1379937     5.92   0.000     1.394797    1.938166
total_mortg~e |          1   4.00e-07    -0.09   0.926     .9999992    1.000001
       _cons |   5.073256   1.140796     7.22   0.000     3.264988    7.883008
------------------------------------------------------------------------------
```

The first thing to notice is that in the top right corner Stata reports that 3078 observations were used in the model. Given that our dataset has close to 9000 observations this means that we have a large number of cases with missing data for one or more of the variables used in the model: any case with a missing value for any of the variables included in a regression model will be omitted from the model. This amount of missing data and dropped cases would warrant further investigation as the estimates in this model may be biased given that they rely on only a portion of the cases. The top right of the output shows that the model we ran is statistically significant (Prob > chi2=0.00). Logistic (and logit) regression do not have an R-Square in the way that OLS regression does but psudo-R^2 estimates such as this have been created in a range of ways. In this example, the

pseudo-R2 value suggests that around 4.4% of the variance in garden possession is explained by this model. It is advisable to interpret the psudo-R^2 estimate in a logistic/logit regression with extreme caution.

Turning to the coefficients, it can be seen that the only coefficient which is statistically significant is that for household size. The amount of labour income and the total mortgage are not statistically significant predictors, perhaps surprisingly. Focussing on the coefficient of 1.64 for household size, the interpretation of this coefficient is that for each unit increase in household size (i.e. as the household becomes one person larger) the odds of a household having a garden increase 1.64 fold. Note that with logistic regression odds ratios centre around one (rather than zero as with OLS for instance): here, explanatory variables with odds ratios greater than one relate to households being more likely to have a garden whilst odds ratios of less than one mean households are less likely to have a garden.

Again, independent variables which are categorical can be included using i. prefix:

```
logistic garden2 inc_lab hhsize total_mortgage i.tenure
```

The logit command will fit exactly the same model but will report beta coefficients rather than odds ratios (unless specified otherwise). Hence,

```
logit garden2 inc_lab hhsize total_mortgage
```

produces the following output:

```
. logit garden2 inc_lab hhsize total_mortgage

Iteration 0:   log likelihood =  -551.0221
Iteration 1:   log likelihood = -528.53089
Iteration 2:   log likelihood = -526.64075
Iteration 3:   log likelihood = -526.63821
Iteration 4:   log likelihood = -526.63821

Logistic regression                             Number of obs   =       3078
                                                LR chi2(3)      =      48.77
                                                Prob > chi2     =     0.0000
Log likelihood = -526.63821                     Pseudo R2       =     0.0443

------------------------------------------------------------------------------
     garden2 |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      inc_lab |   5.57e-06   4.53e-06     1.23   0.219    -3.31e-06    .0000145
       hhsize |   .4972456   .0839283     5.92   0.000     .3327492    .6617419
total_mortgage |  -3.74e-08   4.00e-07    -0.09   0.926    -8.22e-07    7.47e-07
        _cons |   1.623983   .2248647     7.22   0.000     1.183256     2.06471
------------------------------------------------------------------------------
```

At the top of the output Stata reports the log likelihood at each iteration, with iteration zero relating to the null (or empty) model. Logistic regression uses maximum likelihood, which is an iterative process, and seeks to maximise the log likelihood at each iteration, stopping when the difference between successive iterations is small and the model is said to have converged. At the top right of the output, it can be verified that the logit model uses the same number of cases as the logistic model above, the likelihood ratio Chi-square test is reported and the model is statistically significant (this is the probability of obtaining the Chi-square statistic given that the null hypothesis is true) and the psudo-R^2 is reported (again, 4.4% of the variance explained as this is the same model as above).

The coefficients are reported in log odds and relate to the prediction of the dependent variable given the values of the independent variables. In terms of the regression equation, this can be written as:

$$Log(p/1-p) = cons + b1*X1 + b2*X2...+ error$$

where p is the probability of having a garden, b relates to the estimates of the coefficients and X relates to the values of the independent variables.

The coefficients report the predicted increase in the log of the odds having a garden given a one unit increase in the independent variable, holding all other predictors constant. For example, in the above output the coefficient for household size is statistically significant and has a value of 0.497. This tells us that for each unit increase in household size we predict a 0.497 increase in the log odds of having a garden, controlling for all other predictors in the model. As log-odds are difficult to interpret odds ratios are often used instead.

1.18. Post-estimation commands

Following any estimation command there are a range of post-estimation commands which can be used to obtain further information or carry out further tests. Two of the most commonly used are predict and test.

1.19. Predict

After running estimation commands it is possible to use the predict command to fit the previous model to the data and, therefore, to create predicted values of the dependent variable for each case based on the model coefficients applied to the data values of the case in the dataset. Predict must follow the estimation command immediately and consists of the command itself plus the name of the new variable to create. After the comma, options are specified; in this case xb for linear predictions. For example,

```
regress exp_food tot_hh_inc rooms
predict pred_exp_food, xb
```

In this example we initially run the linear regression command in line one and this gives us all of the usual model output which we have seen above. The predict command in the second line uses this output to take the values for each observation and fit these values to the regression model's coefficients in order to predict the value of the outcome variable based on the regression equation (which we decided to call pred_exp_food). Often, this will be used to compare the actual data for the dependent variable against the predicted value using a scatter diagram. This can also be useful to, for example, impute predicted values of exp_food in cases with missing values or for cases outside the range of values covered in our dataset. What predict does exactly depends on the estimation command which it follows.

The predict command can take various different options in order to gain extra information including:

Option after predict	**Option calculates**
resid	Residuals
rstandard	Standardized residuals
rstudent	Studentized or jacknifed residuals
lev or hat	Leverage
stdr	Standard error of the residual
cooksd	Cook's D
stdf	Standard error of the predicted individual
stdp	Standard error of the predicted mean

1.20. Test

Following a regression model it is also possible to test whether one or more explanatory variables collectively are significant predictors. For example, in our dataset we have five individual variables relating to different income sources, and collectively these can conceptually be thought of as representing the contribution to the model of 'income' as an explanatory factor. We could include each of these in the model as explanatory variables – using the inc* to include all variables beginning with 'inc' in the model as independent variables - and then use test to assess the collective contribution of these variables to the predictive power of the model.

```
regress exp_food inc* rooms i.tenure
```

produces the normal regression output. We can then follow this with the test command and specify the five income variables which we wish to collectively test

```
test inc_lab inc_nonlab inc_pens inc_bens inc_inv
```

which produces the following output:

```
. test inc_lab inc_nonlab inc_pens inc_bens inc_inv

 ( 1)  inc_lab = 0
 ( 2)  inc_nonlab = 0
 ( 3)  inc_pens = 0
 ( 4)  inc_bens = 0
 ( 5)  inc_inv = 0

       F(  5,  8159) =   22.09
            Prob > F =    0.0000
```

This tests whether each of the coefficients equal zero. The fact that the p-value at the bottom of the output is statistically significant means that the collective contribution of these five variables is statistically significant. In other words, there is a statistically significant difference between a model with these variables included and a model without them: 'income' as a collective concept seems to be a useful predictor of food expenditure.

If we wish to test whether the coefficients for labour income and pension income can be said to be significantly different to one another in a statistical sense then the test command can also be used to do this:

```
regress exp_food inc* rooms i.tenure
test inc_lab=inc_pens
```

which produces the following output:

```
. test inc_lab=inc_pens

 ( 1)  inc_lab - inc_pens = 0

       F(  1,  8159) =    1.69
            Prob > F =    0.1934
```

In this case test tests the null hypothesis that the difference between the two coefficients is equal to zero. It can be seen that the p-value is not statistically significant at the 5% level and we cannot therefore reject the null hypothesis. This shows that there is insufficient evidence to suggest that the two variables have a statistically significantly different effect on the outcome.

1.21. More advanced post-estimation commands in Stata

Stata's flexibility and statistical power has made it popular with those seeking advanced modelling commands and these are beyond the scope of this course, although a good understanding of the use and interpretation of linear and logistic regression models is advisable before applying other estimation commands. Some of the (many!) other estimation commands available include:

Command	Type of model	Description of model
rreg	Robust regression	OLS regression which is less sensitive to outliers
mlogit	Multinomial logistic regression	Categorical dependent variable which has no natural ordering (e.g. tenure type)
ologit	Ordered logistic regression	Categorical dependent variable which has a natural ordering but where the distances between adjacent levels are unknown (e.g. degree class – Distinction, Merit, Pass, Fail)
probit	Probit regression	Binary dependent variable and estimates predicted probabilities (similar to logit)
oprobit	Ordered probit regression	Categorical dependent variable which has a natural ordering but where the distances between adjacent levels are unknown (e.g. degree class – Distinction, Merit, Pass, Fail)
tobit	Tobit regression	Generates a model that predicts the dependent variable to be within a specified range. Used when the dependent variable is left and/or right censored (e.g. if dependent variable is test scores where the minimum test score is 30 and the maximum is 80)
intreg	Interval regression	Used when the dependent variable is banded and censored (e.g. income bands which begin at £0-£500 as the lowest band and end at £5000+ as the top band)
truncreg	Truncated regression	Used when the data includes only a section of the population. For example, a sports training programme seeks to improve speed, but an evaluation concern is that individuals must be able to run 100m in less than 12 seconds in order to be admitted to the training programme. There data (and model) will therefore exclude all slower runners.
xtreg	Regression models for panel data	Used when there panel datasets across multiple time points, or on non-panel data for fitting random and fixed effects.
svy:regress, etc	Survey regression models	Regression models for survey data (weights, clustering and stratification can be handled)
heckman	Heckman selection models	Used when there is self-selection in the data (e.g. a dataset of females' earnings where some females choose not to work and so have zero income.

		Heckman models could be used if we wanted to predict wage income to take account of the fact that these women could have chosen to work and hence earn positive income, hence it would give us an unbiased estimate)
arima & arch	Time-series models	Used to fit dynamic regression models where the data over a time period (e.g. crime counts for a ten year period)
poisson	Poisson regression	Used to model count data with a poisson distribution, typically for rate data in which the outcome variable is the rate at which an event occurs over a given time period
psmatch2	Propensity score matching (downloadable .ado file)	Used for generating propensity scores and carrying out propensity score matching of various types

Exercise 2 Regression analysis (15 mins)

- *Use the bp_data.dta saved in "H:\StataLevel3\Raw data" (fictional data)*
- *Apply some of the regression analysis commands introduced in this session*

Task 1

- Generate a linear regression model with bp_after as the outcome variable, and bp_before as the explanatory variable.
  Hint: Information in Stata code for simple linear regression is provided in section 1.13.
- Is the explanatory variable (i.e. bp_before) statistically significant (at the 5% level) in this model?
- What is the R^2(adjusted) value, and what does it mean?

Task 2

- To improve the model, add "weight" as another continuous or interval variable to the model. Also add "trt" as a categorical variable to the model.
  Hint: Which additional code do you need to use to implement categorical variables? Information provided in section 1.14.
- How much more of the variation in the outcome variable does this model explain?
- Are the additional variables significant in the model?
- What effect does drug A have on the outcome variable, compared to Drug B?
- Change the code in the above model so that Drug A is now the reference category (i.e. the effect of drug B on the outcome variable should be displayed in the output, instead of drug A).
  Hint: Drug B is coded 0, drug A is coded 1.

Task 3

- Keep the three explanatory variables in your model, but change the outcome to the "low_bp" variable. This is a binary variable.
- Ensure that odds ratios are displayed in the output.
  Hint: information on logistic regression models is provided in section 1.17.
- Is bp_before statistically significant in the model?
- Is the drug received ("trt") statistically significant in the model? How do you interpret this explanatory variable in the model?

2 Survey analysis in Stata

2.1. Why use Stata's survey commands?

If we have a survey dataset and we wish to make more general claims not about the survey sample but about the population as a whole which that sample represents then we need to use specific survey commands to analyse the data. Most basically, this is done by applying weights to any analyses in order to correct, for example, for sampling and non-response and to make inferences about the total population. More complex sampling designs may also require us to take account of survey data which is clustered and/or stratified. Not accounting for these features where they apply is likely to lead to inaccurate, i.e. biased estimates of point estimates and/or variance (hence, confidence intervals).

Stata has a variety of simple commands to analyse survey data – svy commands - and these can handle complex survey data in which the survey analysis needs to take account of the fact that the survey data are weighted, stratified and clustered, or some combination thereof. In order for Stata's survey commands to work correctly it is first necessary to set Stata up for survey analyses by:

1. identifying the relevant survey variables
2. identifying the weight to use
3. identifying the primary sampling unit (PSU) from which the data were sampled (e.g. sampling clustered by postcode sector or area)
4. identifying strata for which a particular sample is desired in order to ensure a 'representative' sample (e.g. region, ethnicity or income group).

The estimation of point estimates (e.g. survey means) will be correct if only the weight is set, but in order to maximise the accuracy of the estimates of variance and standard errors it is necessary to identify as many of these three variables as possible (where they are relevant) when setting up the survey analysis in Stata. In order to know which variables ought to be defined as weights, primary sampling unit (PSU) and strata it is usually necessary to consult the metadata or the documentation that comes with the survey data and this will usually have a section dedicated to these issues.

2.2. Setting up Stata for survey analyses

In order to set up Stata to analyse survey data the syntax is as follows: Survey commands are one area where the syntax structure has changed slightly between versions 8 and 9 to 11 of Stata and so it is worthwhile setting up the survey commands for the particular version you will be working on. Here this is done for current versions of Stata (versions 11 to 13). The general syntax to set up survey data is:

```
svyset PSU [pweight= weight variable], strata(strata variable, if any) ///
fpc (finite population correction, if any)
```

In our case therefore we have a weight variable and an region variable which was the primary sampling unit:

```
use "$raw\bhps_for_class.dta"
```

```
svyset area [pweight=hh_wt2]
```

To check the survey setup you use the svydes command and to clear all survey setting so that you can set them differently the syntax is svyset,clear.

Once set up, Stata can perform most of the analysis commands that you would normally use, but for survey data. In terms of descriptive statistics the most common commands are svy:mean (survey means), svy:prop (survey proportions), svy:ratio (survey ratios), svy:total (survey totals) and svy:tabulate (survey tabulate). A wide range of regression commands can also be used as survey commands, for example svy:regress (linear regression with survey data), svy:logistic (logistic regression with survey data, reporting odds ratios) and a wide range of others.

2.3. Survey means

Let us assume we want to use the survey data to estimate mean total household income in England (i.e the wider 'population' to which the survey data relate). To do this the syntax would be:

```
svy:mean tot_hh_inc
```

The output produced shows that 8602 observations were used in the calculation (all of our dataset) and that the total weighted population size is around 18 million households (1797429), which looks reasonable.

The survey mean of total household income in the dataset is reported as 34,291. The output also reports the standard error and confidence around this point estimate, with the 95% confidence interval in this case ranging from 33,507 to 35,076.

```
. svy:mean tot_hh_inc
(running mean on estimation sample)

Survey: Mean estimation

Number of strata =       1          Number of obs    =     8602
Number of PSUs   =     156          Population size  =  1797429
                                    Design df        =      155
```

	Mean	Linearized Std. Err.	[95% Conf.	Interval]
tot_hh_inc	34291.61	396.9596	33507.46	35075.76

Just to compare, below we show the standard mean of total household income which is produced using the summarize command. This calculation therefore shows only the data within the sample: it does not make use of Stata's survey commands and so cannot tell us anything about the population as a whole beyond this particular survey. It can be seen that the unweighted mean figure is somewhat higher than that calculated using Stata's survey commands.

```
summ tot_hh_inc
```

```
. summ tot_hh_inc
```

Variable	Obs	Mean	Std. Dev.	Min	Max
tot_hh_inc	8602	35174.27	27392.58	0	1013920

2.4. Survey proportions

Having calculated the mean total household income using the survey commands let us apply the svy:prop command (survey proportions) to calculate the proportion of the population as a whole who live in poverty. A simplistic definition of poverty will be used, namely below 60% mean income:

```
/**svy:prop - survey proportions**/

/*first generate a new poorflag variable which is based on survey
mean of the data and is numeric: 1=deprived, 0=not deprived*/

gen deprived_flag=0

replace deprived_flag=1 if tot_hh_inc < (0.6 * 34291)


/*svy:prop - now calculate survey proportions*/

svy:prop deprived_flag
```

The following output is produced and this tells us that in the population as a whole we can estimate on the basis of this survey that 28% of households are poor based on this (admittedly crude) definition of poverty, and again the 95% confidence intervals are reported for each of these point estimates.

```
. svy:prop deprived_flag
(running proportion on estimation sample)

Survey: Proportion estimation

Number of strata =       1          Number of obs    =     8602
Number of PSUs   =     156          Population size  =  1797429
                                    Design df        =      155
```

	Proportion	Linearized Std. Err.	[95% Conf.	Interval]
deprived_flag				
0	.7172998	.0073697	.7027418	.7318579
1	.2827002	.0073697	.2681421	.2972582

2.5. Running survey commands within subgroups

It is also possible to combine survey commands with 'over' groups. Therefore, if we wanted to calculate mean total household income for each region and then to calculate the proportion of households which were poor according to their own regional poverty line then we would be able to do this as follows:

```
/**combining survey commands over bygroups**/

svy: mean tot_hh_inc, over(region)
```

This gives us survey means for each region and it is not surprising to find that the mean income in London is the largest whilst that for the North East is the lowest:

```
Survey: Mean estimation

Number of strata =       1          Number of obs    =     4055
Number of PSUs   =     108          Population size  =   841600
                                    Design df        =      107

       london: region = london
   south_east: region = south_east
   south_west: region = south_west
     midlands: region = midlands
   north_west: region = north_west
   north_east: region = north_east
```

Over	Mean	Linearized Std. Err.	[95% Conf.	Interval]
tot_hh_inc				
london	40403.12	1735.971	36961.76	43844.48
south_east	37097.75	843.2371	35426.13	38769.36
south_west	34991.65	1452.465	32112.31	37870.99
midlands	32958.86	1210.694	30558.8	35358.92
north_west	35103.45	957.8364	33204.65	37002.25
north_east	32724.39	899.4504	30941.34	34507.45

Having calculated regional means, we can now make a poor flag specific to each region and then calculate survey proportions of poor households based on these regionally defined poverty thresholds. The syntax we use here is quite cumbersome and it would be possible in reality to use matrices and macros to access the results saved by the svy:mean automatically and make use of them. This is a more advanced topic and is covered in the Stata *Introduction to Stata programming* course, but it is worth highlighting that this is possible (and useful).

```
/* create regional poverty flag */
gen reg_poor=0
replace reg_poor=1 if tot_hh_inc < (0.6 * 40403) & region==1 /*london*/
replace reg_poor=1 if tot_hh_inc < (0.6 * 37097) & region==2 /*south east*/
replace reg_poor=1 if tot_hh_inc < (0.6 * 34991) & region==3 /*south west*/
replace reg_poor=1 if tot_hh_inc < (0.6 * 32958) & region==4 /*midlands*/
replace reg_poor=1 if tot_hh_inc < (0.6 * 35103) & region==5 /*north west*/
replace reg_poor=1 if tot_hh_inc < (0.6 * 32724) & region==6 /*north east*/

/* and calculate regional proportions in poverty */
svy:prop reg_poor, over(region)
```

This syntax produces the following output. The number of observations used is shown in the top right corner – note that around half of our dataset have a missing value for the region variable and so were not used in the calculation, and in reality we should investigate this further as this could lead to biased estimates. The number of PSUs is listed as 108 (i.e. we have 108 different values in the 'area' variable within which the sample is clustered), and the labels for the table are shown: _prop_1 means the household is not poor and _prop_2 means that the household is poor (the top part of the main body of the output shows these labels). Based on the regionally defined poverty thresholds these output show slight differences in the point estimates of the proportion of households poor in each region, although the differences are small and the confidence intervals are relatively wide.

```
. svy:prop reg_deprived, over(region)
(running proportion on estimation sample)

Survey: Proportion estimation

Number of strata =       1          Number of obs    =    4055
Number of PSUs   =     108          Population size  =  841600
                                    Design df        =     107

      _prop_1: reg_deprived = 0
      _prop_2: reg_deprived = 1

       london: region = london
   south_east: region = south_east
   south_west: region = south_west
     midlands: region = midlands
   north_west: region = north_west
   north_east: region = north_east
```

Over	Proportion	Linearized Std. Err.	[95% Conf.	Interval]
_prop_1				
london	.7099829	.0216196	.6671247	.7528412
south_east	.7009363	.0162479	.6687268	.7331459
south_west	.7179271	.0281903	.6620432	.773811
midlands	.7083921	.0272613	.6543498	.7624344
north_west	.7111152	.0234929	.6645433	.757687
north_east	.7252398	.0190834	.6874091	.7630705
_prop_2				
london	.2900171	.0216196	.2471588	.3328753
south_east	.2990637	.0162479	.2668541	.3312732
south_west	.2820729	.0281903	.226189	.3379568
midlands	.2916079	.0272613	.2375656	.3456502
north_west	.2888848	.0234929	.242313	.3354567
north_east	.2747602	.0190834	.2369295	.3125909

2.6. Survey totals

svy:total is another commonly used survey command and calculates survey totals. The dataset we are using has a variable called 'kids' which relates to the number of children in the household. If we wanted to create a survey total of the total number of children in each region then the syntax would be:

```
svy:total kids, over(region)
```

and if we wanted to calculate a survey total of the number of poor children in each region then we could run the command 'over' both region and poor_flag:

```
svy:total kids, over(deprived_flag region)
```

The output produced from this syntax is shown below. The top section lists the strata (we did not set a strata variable and so we only have one strata), PSUs, the

number of observations used, weighted population size (of children in this case), and the degrees of freedom (PSUs – 1). Below this, the output lists the labels which are used in the main body of the output: Stata creates one subpopulation for each group within the over option. In this example we used two variables in the over command and so we have 12 subpopulations (non-poor in each of the six regions and poor in each of the six regions). Hence, this list shows that non-poor children in London are labelled in the main output below as subpopulation 1. Looking down the main table of results, the output shows that there are estimated to be 275,101 non-poor children in London (subpopulation one), though in this instance there are wide confidence intervals around this point estimate.

```
. svy:total kids, over(deprived_flag region)
(running total on estimation sample)

Survey: Total estimation

Number of strata =       1          Number of obs    =    4030
Number of PSUs   =     108          Population size  =  838837
                                    Design df        =     107

         Over: deprived_flag region
    _subpop_1: 0 london
    _subpop_2: 0 south_east
    _subpop_3: 0 south_west
    _subpop_4: 0 midlands
    _subpop_5: 0 north_west
    _subpop_6: 0 north_east
    _subpop_7: 1 london
    _subpop_8: 1 south_east
    _subpop_9: 1 south_west
   _subpop_10: 1 midlands
   _subpop_11: 1 north_west
   _subpop_12: 1 north_east
```

Over	Total	Linearized Std. Err.	[95% Conf.	Interval]
kids				
_subpop_1	28169.74	4489.94	19268.95	37070.52
_subpop_2	139645.6	12817.49	114236.5	165054.8
_subpop_3	32362.49	5118.979	22214.71	42510.27
_subpop_4	41516.23	8317.24	25028.27	58004.19
_subpop_5	41163.61	7418.698	26456.91	55870.32
_subpop_6	59286.17	6808.688	45788.74	72783.6
_subpop_7	8479.56	2314.243	3891.843	13067.28
_subpop_8	18321.51	3333.764	11712.71	24930.31
_subpop_9	8597.257	2391.162	3857.057	13337.46
_subpop_10	6092.647	1982.88	2161.818	10023.48
_subpop_11	8347.849	2109.829	4165.359	12530.34
_subpop_12	9010.183	2016.152	5013.398	13006.97

2.7. Survey ratios

svy:ratio is used to calculate survey ratios. In addition to the 'kids' variable which lists the number of children in the household our dataset also contains a variable called 'wage' which denotes the number of working age persons in the household. Using these two variables we can use svy:ratio to calculate a survey ratio of the child dependency ratio in each region. The child dependency ratio is a measure of the ratio between children (who are assumed to be economically unproductive in the labour market) and working age adults (who are the potential economically active persons in the household). Larger ratios are taken to imply a greater burden on the economically active to support the economically inactive (in this case just children as we ignore the elderly). Formally, the child dependency ratio is usually calculated as:

(number of children / number of working age adults) * 100

We would calculate this using svy:ratio with the following syntax:

```
. svy:ratio kids wage, over(region)
(running ratio on estimation sample)

Survey: Ratio estimation

Number of strata =       1          Number of obs    =    4030
Number of PSUs   =     108          Population size  =  838837
                                    Design df        =     107

     _ratio_1: kids/wage

       london: region = london
   south_east: region = south_east
   south_west: region = south_west
     midlands: region = midlands
   north_west: region = north_west
   north_east: region = north_east
```

Over	Ratio	Linearized Std. Err.	[95% Conf.	Interval]
_ratio_1				
london	.2969649	.05107	.1957245	.3982053
south_east	.3374855	.0271269	.2837096	.3912615
south_west	.3246024	.0554292	.2147206	.4344843
midlands	.3917171	.0439216	.3046476	.4787865
north_west	.3245649	.0350828	.2550173	.3941124
north_east	.3368389	.0388612	.2598012	.4138766

This shows that the survey ratio of children to working age adults is highest in the Midlands – estimated to be 39.17% and with the 95% confidence interval ranging from 33.25% to 45.09% - and lowest in London at 29.70%. Naturally, if we wanted to calculate the child dependency ratio for the country as a whole then we would simply exclude the over option.

2.8. Survey tabulate

svy:tabulate is used to run survey tables of frequencies and percentages, and both one and two-way tables can be generated. The following syntax creates a one-way table of household size and two common options are specified: the count option requests survey counts for each cell and the cell option requests survey proportions for each cell. The output produced looks as follows:

```
svy:tabulate hhsize, count cell
```

```
. svy:tabulate hhsize, count cell
(running tabulate on estimation sample)

Number of strata   =         1                  Number of obs      =      8602
Number of PSUs     =       156                  Population size    = 1797428.8
                                                Design df          =       155
```

number of persons in household	count	proportions
1	5.3e+05	.2972
2	5.9e+05	.3306
3	2.7e+05	.152
4	2.7e+05	.1514
5	9.2e+04	.051
6	2.1e+04	.0119
7	6690	.0037
8	1896	.0011
9	1168	6.5e-04
10	151.5	8.4e-05
11	379.5	2.1e-04
13	239.3	1.3e-04
Total	1.8e+06	1

Key: count = weighted counts
propor~s = cell proportions

The top right of the output lists the number of observations used in the calculation (all of our dataset) and gives the total population size (i.e. an estimate of all households in the population in this case). In the main output table estimates of survey counts and of survey proportions are given. This output shows that it is estimated that there are just under 1900 households in the population which have eight people and that this constitutes 0.01% of all households in the population as a whole. If we wanted these data by region then the over(region) option could be added.

It is possible to obtain confidence intervals around these point estimates, but if requesting confidence intervals then syntax options are restricted. Therefore, to obtain survey counts with confidence intervals the syntax would be:

```
svy:tab hhsize, count ci
```

svy:tab can also be used to produce two-way survey tables. For example, if we wished to produce a survey table of region against garden and to report survey estimates for cell counts, cell proportions, column percentages and row percentages then the syntax would be:

```
svy:tab region garden, count cell row col
```

```
Number of strata   =         1                  Number of obs      =      3848
Number of PSUs     =       106                  Population size    = 799639.02
                                                Design df          =       105
```

RECODE of regioncode (region / metropolitan area)	accom: has terrace/garden		
	yes	no	Total
london	7.3e+04	9507	8.3e+04
	.0917	.0119	.1036
	.8852	.1148	1
	.0977	.1955	.1036
south_ea	2.9e+05	1.6e+04	3.1e+05
	.3615	.0205	.382
	.9463	.0537	1
	.3849	.3375	.382
south_we	8.3e+04	3723	8.6e+04
	.1035	.0047	.1082
	.957	.043	1
	.1102	.0766	.1082
midlands	7.8e+04	3940	8.2e+04
	.097	.0049	.102
	.9517	.0483	1
	.1033	.081	.102
north_we	9.5e+04	7876	1.0e+05
	.1183	.0098	.1282
	.9231	.0769	1
	.126	.162	.1282
north_ea	1.3e+05	7168	1.4e+05
	.1671	.009	.1761
	.9491	.0509	1
	.1779	.1474	.1761
Total	7.5e+05	4.9e+04	8.0e+05
	.9392	.0608	1
	.9392	.0608	1
	1	1	1

```
  Key:  weighted counts
        cell proportions
        row proportions
        column proportions

  Pearson:
    Uncorrected   chi2(5)         =   28.3689
    Design-based  F(4.07, 427.67) =    4.7358     P = 0.0009
```

It can be seen that not all of the cases are used in the calculation due to many missing values on one or other of the variables – if this were a real research analysis we would want to investigate this further to see why there were so many missing values as this may bias results. At the bottom of the output the Chi-square test is reported and in this case the differences between the cells are statistically significant. Also at the bottom of the output Stata lists each element of the cells: at the top of each cell are weighted counts, then cell proportions, next are row proportions and at the bottom of each cell are column proportions. Looking at the top left cell of the main table, therefore, this relates to houses in London which have a garden. This cell shows, i.e. that it is estimated that there are 720,000 houses in London with a garden, approximately 88.5% of houses in London, and that 9.77% of houses with a garden are in London. Household in London that have a garden make up 9.17% or all households. As these are survey analyses, these findings do of course relate to estimates for the general population and not just for the sample data.

2.9. Survey regression

Stata's survey commands also work with a range of estimation commands (regress, logit, probit, poisson, etc) in just the same way as their 'normal' (i.e. non-survey) usage, and predict can also be used as a postestimation command in just the same way (see Statistics section above for a discussion of predict). Therefore, to fit a linear regression model with food expenditure as the dependent variable and total household income and the number of rooms (both interval variables) as the explanatory variables the syntax would be:

```
svy:regress exp_food tot_hh_inc rooms
```

This produces the following output, the interpretation of which is the same as outlined above in the Statistics section of this course:

```
. svy:regress exp_food tot_hh_inc rooms
(running regress on estimation sample)

Survey: Linear regression

Number of strata   =         1                  Number of obs      =      8194
Number of PSUs     =       156                  Population size    = 1712793.2
                                                Design df          =       155
                                                F(   2,    154)    =    593.53
                                                Prob > F           =    0.0000
                                                R-squared          =    0.2702
```

exp_food	Coef.	Linearized Std. Err.	t	P>|t|	[95% Conf.	Interval]
tot_hh_inc	.0024588	.0005064	4.86	0.000	.0014585	.0034591
rooms	47.72532	3.657542	13.05	0.000	40.50026	54.95038
_cons	319.8874	8.920803	35.86	0.000	302.2654	337.5095

As before, the i. code needs to be added for categorical variables.

```
svy: regress exp_food tot_hh_inc rooms i.garden
```

The commands for changing the reference group, and interpreting the output are all exactly the same as outlined in the statistics section on regression models earlier in this course.

2.10. Focussing on subpopulations in survey analyses

Above we have made use of the 'over' option to carry out analyses across multiple sub-groups of interest. If we were interested not in multiple sub-groups of a variable (e.g. gender) but only in one particular sub-group (e.g. women) then the 'subpop' option can be used to calculate survey estimates for this single sub-group. The Stata documentation outlines that in order to correctly compute estimates of variance for subgroups it is necessary to use the 'subpop' option and not to simply restrict the calculation by including an 'if' condition as doing so will lead to incorrect estimates of variance (and therefore of confidence intervals). Instead the subpop( ) option is recommended.

In order for subpop to work, the sub-population of interest (e.g women) should be coded '1', all other cases should be coded '0', and cases with missing values should be coded missing (i.e. '.').

Let us assume that we are only interested in childless households and that we wish to use the survey commands to calculate the mean total household income of this particular subpopulation of interest. First we would need to generate a binary variable to identify the subpopulation of interest (i.e. childless households in this example):

```
/* generate a children_flag for the subpop option to take: our
interest is in childless households so these take the value 1,
missing vales are coded missing and all other values are coded 0
*/

gen children_flag=.

replace children_flag=0 if kids >0 & kids != .

replace children_flag=1 if kids==0
```

Next we run the syntax including the subpopulation option. Here we combine subpop with the over(region) option to focus on childless households in each region separately:

```
svy:  subpop(children_flag): mean tot_hh_inc, over(region)
```

The output produced is shown below. It shows in the top right corner that 4029 cases were included in the calculation, reflecting missing values across the three variables used in the syntax (particularly the region variable). Below Stata lists the labelling of the 'over' subgroups as they are shown in the main table of results (in our case each of the regions). The main results table provides survey means, standard errors and confidence intervals around the point estimates. These data show, for example, that the mean total household income of childless households in London is 38,911 with a 95% confidence interval ranging from 36,204 to 41,617.

```
. svy, subpop(kids_flag): mean tot_hh_inc, over(region)
(running mean on estimation sample)

Survey: Mean estimation

Number of strata =       1         Number of obs    =     5452
Number of PSUs   =     156         Population size  =  1122904
                                   Subpop. no. obs  =     2767
                                   Subpop. size     =   606548
                                   Design df        =      155

      london: region = london
  south_east: region = south_east
  south_west: region = south_west
    midlands: region = midlands
  north_west: region = north_west
  north_east: region = north_east
```

Over	Mean	Linearized Std. Err.	[95% Conf.	Interval]
tot_hh_inc				
london	38911.11	1903.658	35150.65	42671.58
south_east	34851.92	1134.216	32611.41	37092.44
south_west	32845.74	1279.856	30317.53	35373.95
midlands	30058.62	1391.412	27310.04	32807.2
north_west	33299.71	1013.427	31297.8	35301.63
north_east	31471.07	1200.362	29099.9	33842.25

2.11. Postestimation commands for survey analysis

Postestimation commands can also be used with survey commands: predict, test and lincom are typically the most commonly used.

2.12. Predict

Predict can be used in just the same way as described earlier in the Statistics section of this course and creates fitted values based on model coefficients as well as residuals and other options. Hence, it is possible to run a survey regression and then use the predict command to create a predicted (or fitted) estimate of the dependent variable in the model for each case in the dataset. See the section on predict in the postestimation section of the Statistics part of this course for further details.

2.13. Test

Test can also be used in the same way as with non-survey commands in order, for example, to test the statistical significance of the difference between regression

coefficients. See the section on tests in the postestimation section of the statistics part of this course for further details.

2.14. Lincom

Lincom is used to test the difference between survey estimates (means, ratios, etc). For example, lincom could be used following a survey mean command to test the difference between two survey means, and hence lincom can in this context be understood as a t-test for survey data:

```
svy:mean tot_hh_inc, over(region)
```

which gives the following output:

```
. svy:mean tot_hh_inc, over(region)
(running mean on estimation sample)

Survey: Mean estimation

Number of strata =       1          Number of obs    =    4055
Number of PSUs   =     108          Population size  =  841600
                                    Design df        =     107

       london: region = london
   south_east: region = south_east
   south_west: region = south_west
     midlands: region = midlands
   north_west: region = north_west
   north_east: region = north_east
```

Over	Mean	Linearized Std. Err.	[95% Conf.	Interval]
tot_hh_inc				
london	40403.12	1735.971	36961.76	43844.48
south_east	37097.75	843.2371	35426.13	38769.36
south_west	34991.65	1452.465	32112.31	37870.99
midlands	32958.86	1210.694	30558.8	35358.92
north_west	35103.45	957.8364	33204.65	37002.25
north_east	32724.39	899.4504	30941.34	34507.45

It looks like the means for London and the Midlands are different to one another, but assume we wish to test whether this difference can be considered to be statistically significant. There are two main ways to do this.

If we wanted to test the difference of survey means (i.e. we wanted to run a t-test on survey data) then we can use lincom to do this:

```
svy:mean tot_hh_inc, over(region)
lincom [tot_hh_inc]london - [tot_hh_inc]north_east
```

The output of the lincom command, shown below, suggests that the difference between the two survey means is statistically significant at the 5% level.

```
. lincom [tot_hh_inc]london - [tot_hh_inc]north_east

 ( 1)  [tot_hh_inc]london - [tot_hh_inc]north_east = 0
```

Mean	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
(1)	7678.732	1896.793	4.05	0.000	3918.561	11438.9

Alternatively, if we wanted to test the equality of the means for London and the Midlands then we could use test to do this:

```
test [tot_hh_inc]london =  [tot_hh_inc]midlands
```

The output, below, suggests that the two survey means are not equal.

```
. test [tot_hh_inc]london =  [tot_hh_inc]midlands

Adjusted Wald test

 ( 1)  [tot_hh_inc]london - [tot_hh_inc]midlands = 0

       F(  1,   107) =   12.37
            Prob > F =    0.0006
```

Exercise 3 Survey commands (15 mins)

- *Use the bp_data.dta saved in "H:\StataLevel3\Raw data" (fictional data)*
- *Apply some of the survey commands that were introduced in the session*
- *Survey commands would not usually be applied to clinical trial datasets. This exercise is merely to demonstrate how these commands work.*

Task 1

- Use svyset to setup Stata to analyse this survey data using the weight variable 'id_wt' and the 'cluster' variable site as the PSU.
  Hint: information in setting up data for survey analyses is provided in section 2.2.
- Use the svydes command to obtain information on the number of subjects (observations) in each unit (i.e. each site).

Task 2

- Calculate a survey mean of follow-up blood pressure for males and females (the relevant variable is called sex). How wide are the confidence intervals around these point estimates?
  Hint: the coding for survey means is described in section 2.3.
- Compare this results to the mean of the trial dataset you using the usual *summarize* command.

Task 3

- Use lincom to test whether the difference between the mean follow-up blood pressure values (bp_after) for males and females are statistically significant at the 5% level.
  Hint: an example of this has been provided in section 2.14. You can use the value labels after the square brackets. Remember that Stata is case sensitive.

Task 4

- Use svy: command to run a weighted logistic regression testing the potential effect of weight and age group (agegrp) on whether the patient has low blood pressure or not (low_bp). Also add the randomised treatment to the model. Remember that age group and treatment are categorical variables.
  Hint: the survey regression commands follow the same rules as the regression commands in exercise 2. Section 2.9 provides more details.

3 Stata graphics

Stata has a large and flexible range of graphics options, although it can take some time to get the graphs just right and to fully grasp the logic of Stata graphic, **the advantage is that once the syntax is written to make the graph you want; it is then easy to quickly replicate the same, or similar, graphs.**

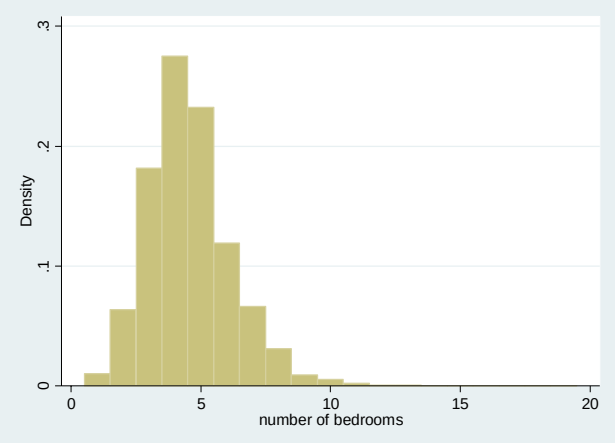
The syntax for Stata graphs can come in up to three parts: (1) main graph command; (2) options; and (3) styles and concepts.

This course only covers basic graphs and options, but additional details can be found in the Stata help or in the more advanced Stata courses taught at the University of Oxford IT services.

3.1. Histogram

A histogram is a graph of a relative frequency distribution for a quantitative variable (Agresti and Finlay 2009). A frequency distribution is a listing of possible values for a variable together with the number of observations for each value. Each interval has a bar over it, with height representing the number of observations in that interval. Histograms are commonly used to also show the bars (often called bins) of densities, frequencies or percentages for each value of a discrete variable along the x-axis. For example, we can create a simple histogram of number of rooms in the house in which the household resides. The default for histogram is to show density and this is what is shown in the graph below. To ensure that one bar is plotted for each different number of rooms, the 'discrete' option is used:

```
/***** histogram *********/
/* density is the default, with no titles or labels */
histogram rooms, discrete
```



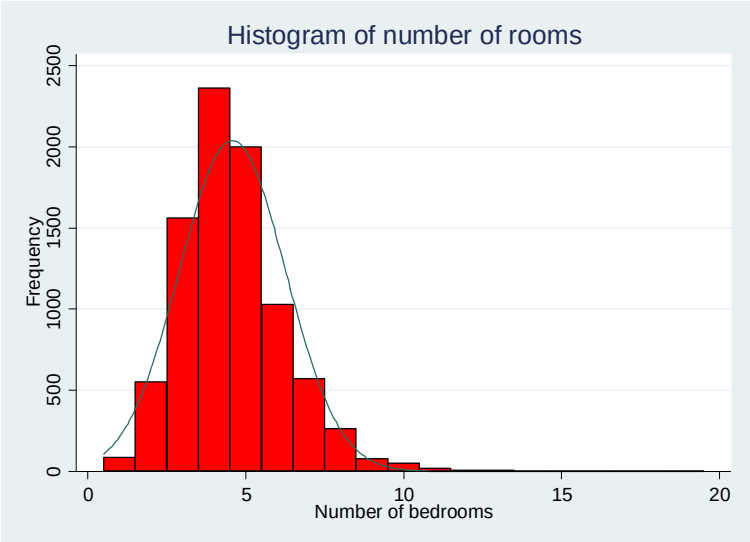
The graph above gives us an indication of the distribution of the variable rooms but visually it may not correspond to textbook and article style graphs. In the next

graph we augment the syntax by adding different options so that the bars represent frequencies rather than densities, we add an overall title to the graph, we title both of the axes, and we change the bar colour to red. Each of these changes is applied by adding options. The frequency option scales the graph in a way that the height of each bar amounts to the number of observations in that category. Alternatively, percent will scale it in a way so that the sum of the heights of all bars will equal 100. A common option is 'normal' and this plots a normal distribution over the top of the histogram.

Writing syntax for graphs often takes several lines (or one very long syntax line). Remember that you can indicate that the command continues to the next line by specifying /// at the end of a line.

Another option among programmers is to use delimit to write the syntax; and change the delimit from a carriage return (the default) to a semi-colon at the start of the syntax and then change it back to a carriage return at the end of the graph syntax: this enables us to press enter during the syntax to make the graph without producing an error message **so long as we remember to include the semi-colon (or whatever you set the new delimit to be) at the end of the block of syntax.** See the *Stata: Data access and management* course for a discussion of the delimit.

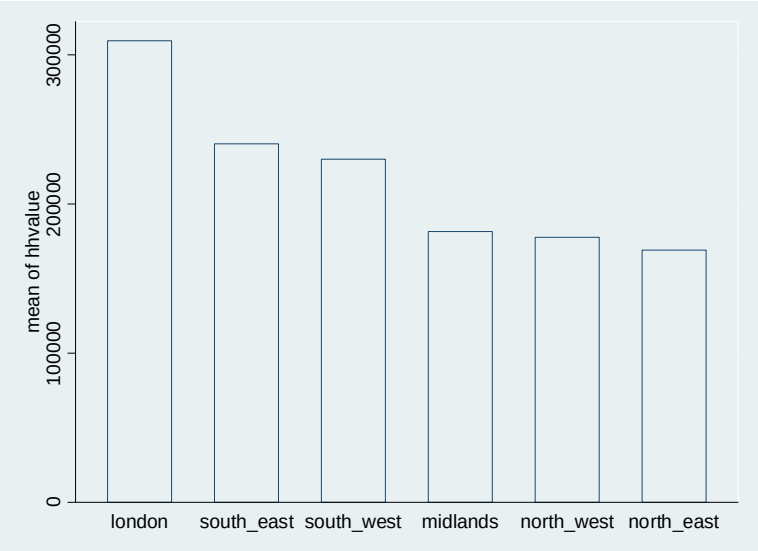
```
/* showing frequency of bins rather than density, with normal
distribution line plotted, with title, xtitle, ytitle and changing
the colour of bins */
histogram rooms, freq discrete normal ///
 title("Histogram of number of rooms") ///
 xtitle("Number of bedrooms") ytitle("Frequency") ///
 fcolor(red) lcolor(black)
```



3.2. Bar graph

Bar graphs are produced with the graph 'bar' command. A bar graph has a rectangular bar drawn over each category. The bars are separated to emphasize that a variable on the x axis is categorical rather than quantitative. For example, if we wanted to produce a bar graph of mean household value for each region then the syntax would be:

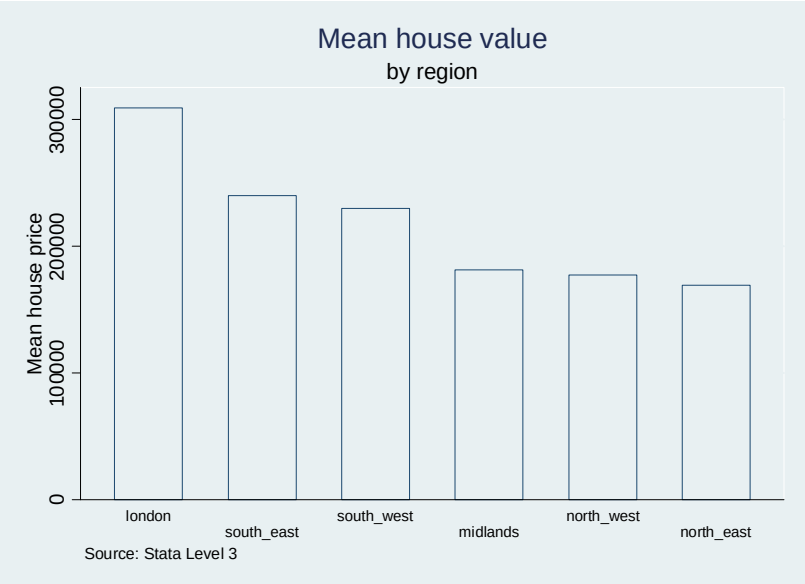
```
graph bar (mean) hhvalue, over(region)
```



Note in the syntax above that the 'over' option is used to specify any categories which you wish the single graph to be presented over. In this example we want a bar for each region and hence the graph is made 'over(region)'. This graph shows us the data we wanted but it would look better with some extra formatting.

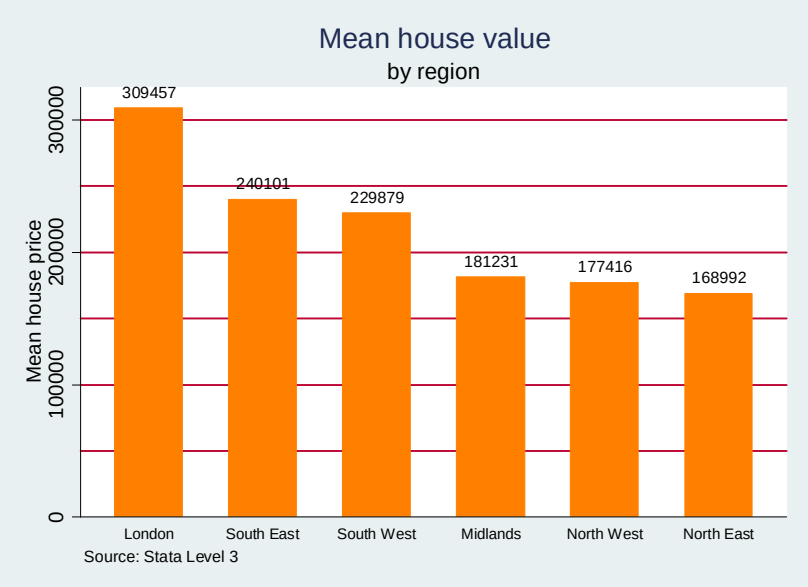
We create the same graph below but with formatting extras: we reduce the label size to 75% of the original size and alternate the labels along the x-axis to give them more room; we also add a title, subtitle, note and title on the y-axis:

```
/* bar graph with reduced label size & alternated labels, title,
subtitle, note and title on y-axis */
graph bar (mean) hhvalue, over(region, label(labsize(*0.75)
alternate))
title("Mean house value")
subtitle("by region")
note("Source: Stata Level 3")
ytitle("Mean house price") ;
```



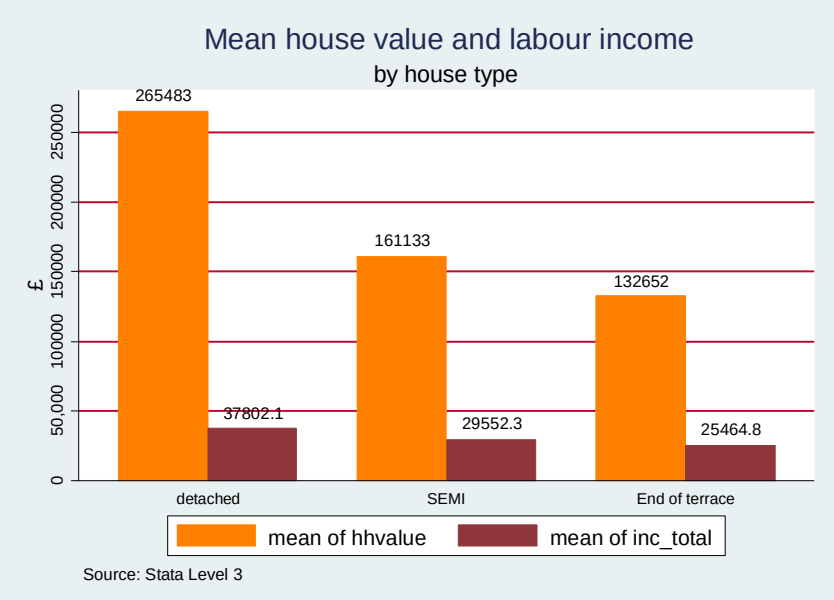
This looks much better but there are still some changes we could make if we wished. In the following graph we relabel the x-axis labels to tidy them up (note that the re-labelling and changing the label size of the over category takes place within the brackets relating to the 'over' category), we label each bar, and we add horizontal lines on the y-axis from 50,000 up to 300,000 at intervals of 50,000. Note that we also specify the bar option to change the colour of the bars to orange: this is a case of specifying which variable you wish to make the changes to (in our case it is number 1 as we only have one variable – hhvalue) then writing a comma and adding in the color option with the desired colour in brackets:

```
/* adding extra options to re-label bars on the x-axis, to add the
mean figure to the top of each bar, and adding y-lines at
intervals of 50,000, and changing bar colour */
graph bar (mean) hhvalue, ///
over(region, relabel(1 "London" 2 "South East" 3 "South West" ///
4 "Midlands" 5 "North West" 6 "North East") label(labsize(*0.75)))
///
title("Mean house value") ///
subtitle("by region") ///
note("Source: Stata Level 3") ///
ytitle("Mean house price")  ///
blabel(bar) bar(1,color(orange)) ///
yline(50000(50000)300000)
```



It is also possible to plot more than one variable at the same time, as shown below:

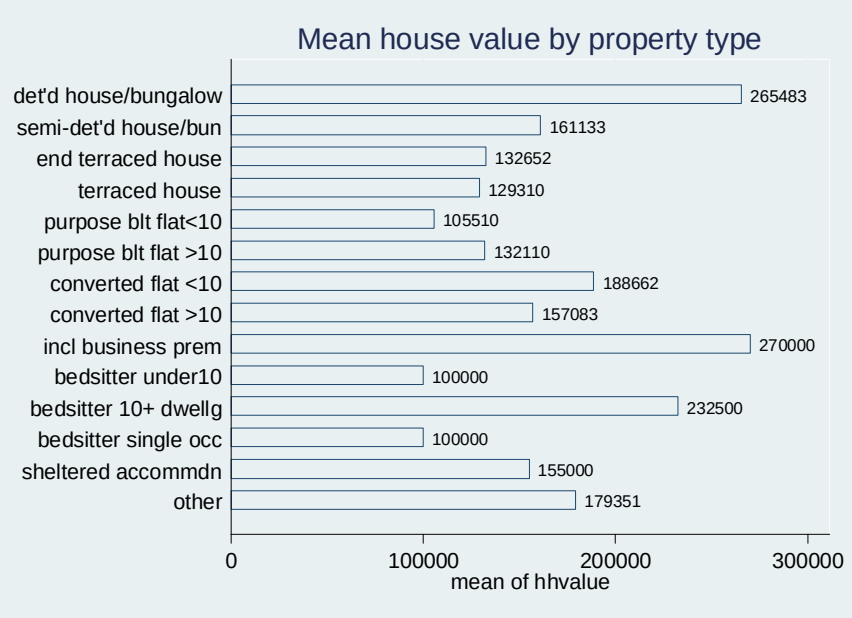
```
graph bar (mean) hhvalue inc_tot if house_type <=3, ///
 over(house_type, relabel(1 "detached" 2 "SEMI" 3 "End of terrace" ) ///
 label(labsize(*0.75))) ylabel(, labsize(small)) ///
 title("Mean house value and labour income")  ///
 subtitle("by house type") ///
 note("Source: Stata Level 3") ///
 ytitle("£")  ///
 blabel(bar) bar(1,color(orange)) ///
 yline(50000(50000)250000)
```



3.3. Horizontal bar graph

It is equally possible to produce horizontal rather than, as in the previous examples, vertical bar graphs. This is simply a case of specifying graph hbar rather than graph bar:

```
graph hbar (mean) hhvalue, over(house_type) ///
title("Mean house value by property type") ///
ytitle(“Mean House Value”) ///
blabel(bar)
```

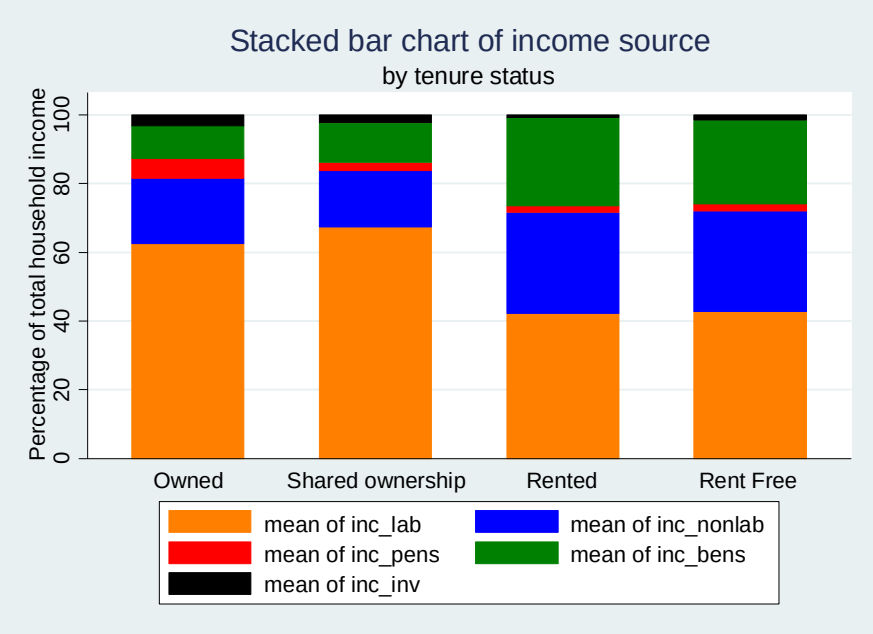


3.4. Stacked bar graph

Stacked bar graphs can be a good way to convey a lot of data in a single graph, particularly when variables could be combined to convey some greater concept. For example, in our dataset we have variables relating to five different income sources and we could use a stacked bar to show how income packages vary across our tenure types (or some other variable of interest such as region) as in the following syntax. Note that we do not wish to create a bar for unspecified tenure (missing) types and those categorised as “other” as these groups are of no interest to us. The mean in brackets is a typical option and indicates that we wish the bars to show that mean of the income variables. The stack option specifies that we wish to make a stacked bar chart and the percentage option specifies that bars should sum to 100% and that stacks within the bar should therefore be the percentage of that tenure type’s total income from each income source. A stacked bar is presented for each tenure type - i.e. the graph is over(tenure) – and there is a title, subtitle and y-axis title specified. Additionally the bar option is specified to adjust the colour of each bar, where the number specified in the bar option relates to the order of the variable in the variable list (e.g. 1 relates to inc_lab):

```
/* income sources over tenure types, with title, subtitle and y-
axis title, and bar colours changed */
```

```
graph bar (mean) inc_lab inc_nonlab inc_pens inc_bens ///
 inc_inv if tenure!=. & tenure!=5,  ///
 over(tenure, relabel(1 "Owned" 2 "Shared ownership" ///
 3 "Rented"  4 "Rent Free")) percentage stack ///
 title("Stacked bar chart of income source") ///
 subtitle("by tenure status") ///
 ytitle("Percentage of total household income") ///
 bar(1,color(orange)) bar(2,color(blue)) bar(3,color(red)) ///
 bar(4,color(green))bar(5,color(black))
```

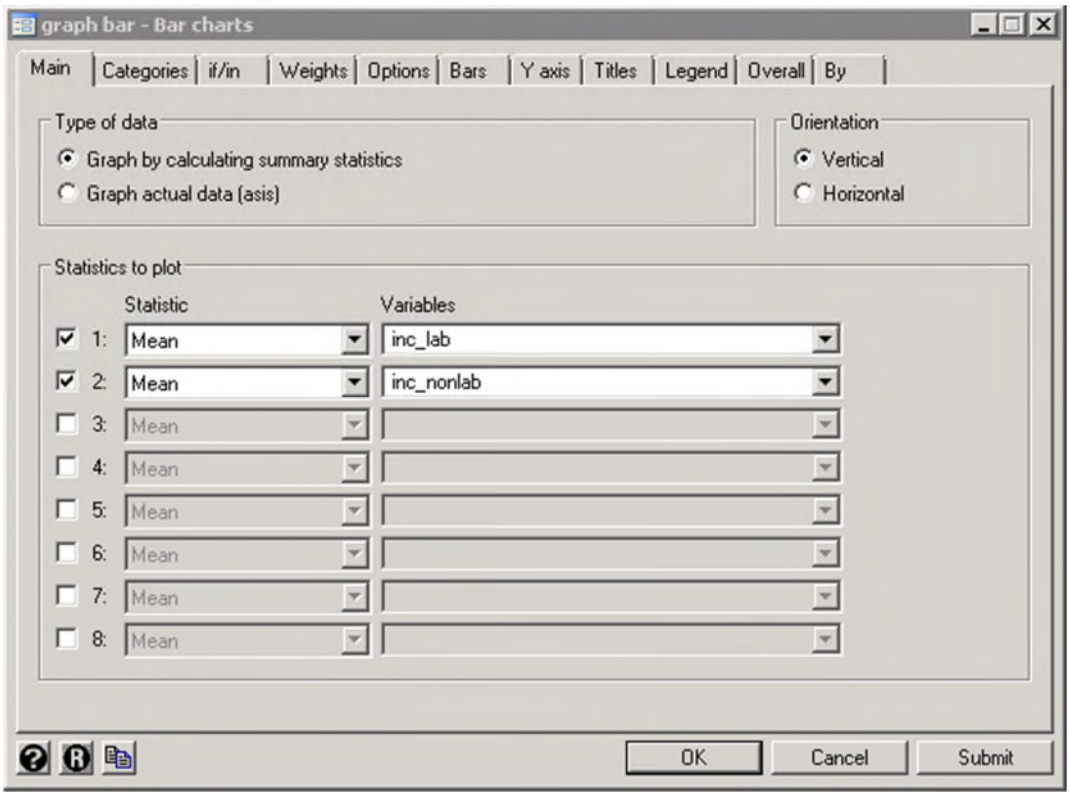


Thus, one possible interpretation in the current case is that over 60 per cent of the income of people in properties that are owned outright or partly owned comes from labour but that share falls over 20% percentage points for people in rented and rent free accommodation.

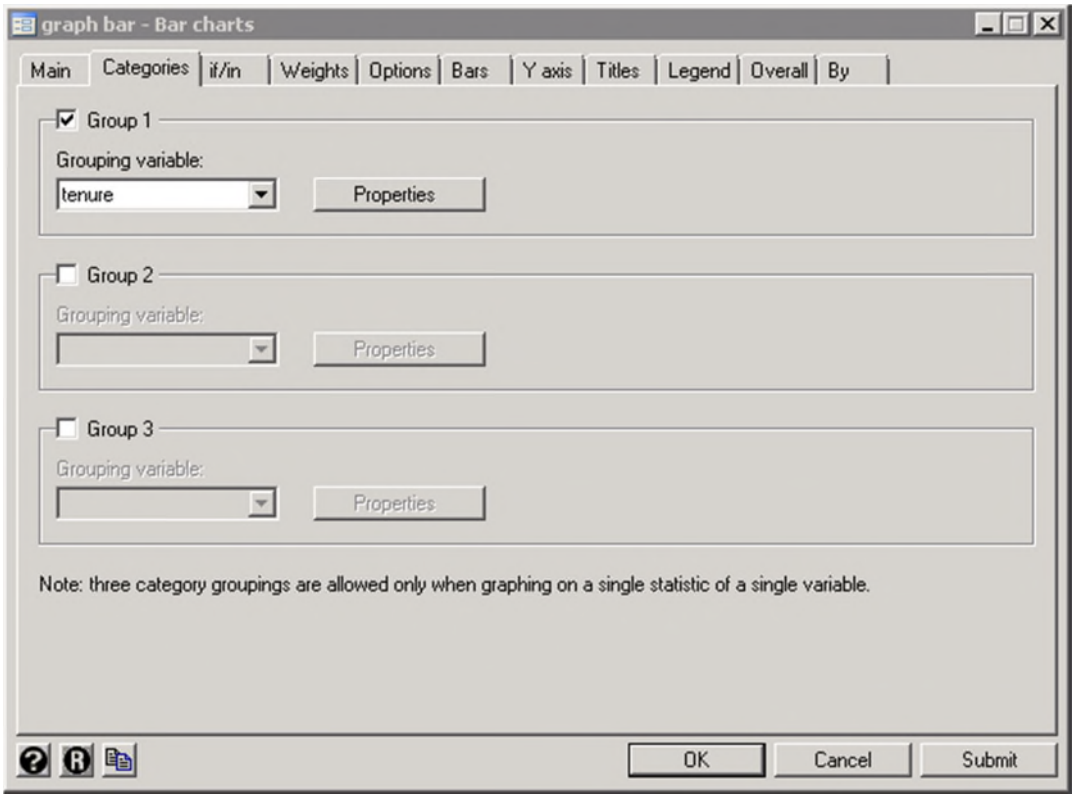
Another way to generate this (or any other) graph is using the drop down menus. Sometimes, it can easier to fine-tune graphs this way, especially for changing appearance and adding a legend. However, the Stata code used to generate the graphs should be copied from the output window and saved in the .do file.

This way, graphs can be easily recreated and amended for different scenarios.

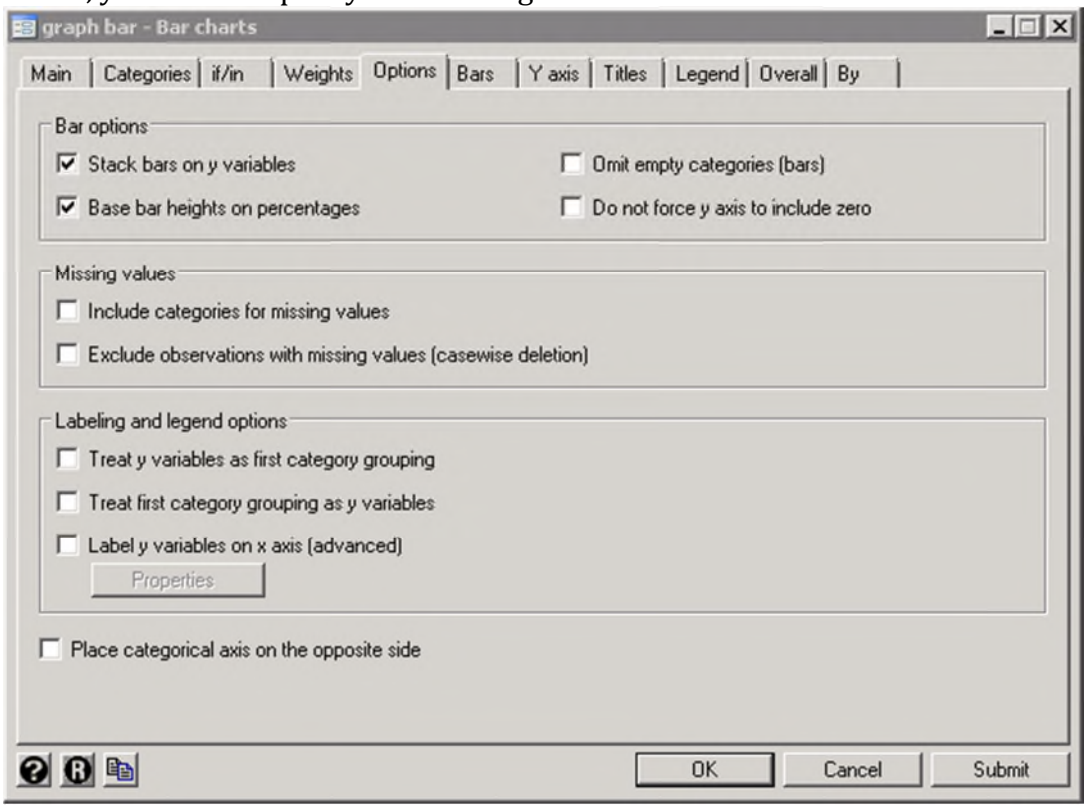
Go to the menu bar and choose Graphics Graphics Choose Bar Chart. You will get the following window:



Then you are interested to specify variable by which categories the mean income of labour will be displayed in our case, this is tenure:



Last but not least, you need to specify the stacking:



In the output window, Stata will specify the code that is required to create manual graphs. Sometimes, it can be easier to create the graph manually, and adjust the code afterwards, and reuse it for other variables. As usual, the code should be saved in the .do file to ensure that output can be reproduced.

3.5. Graphs within graphs: the 'by' option

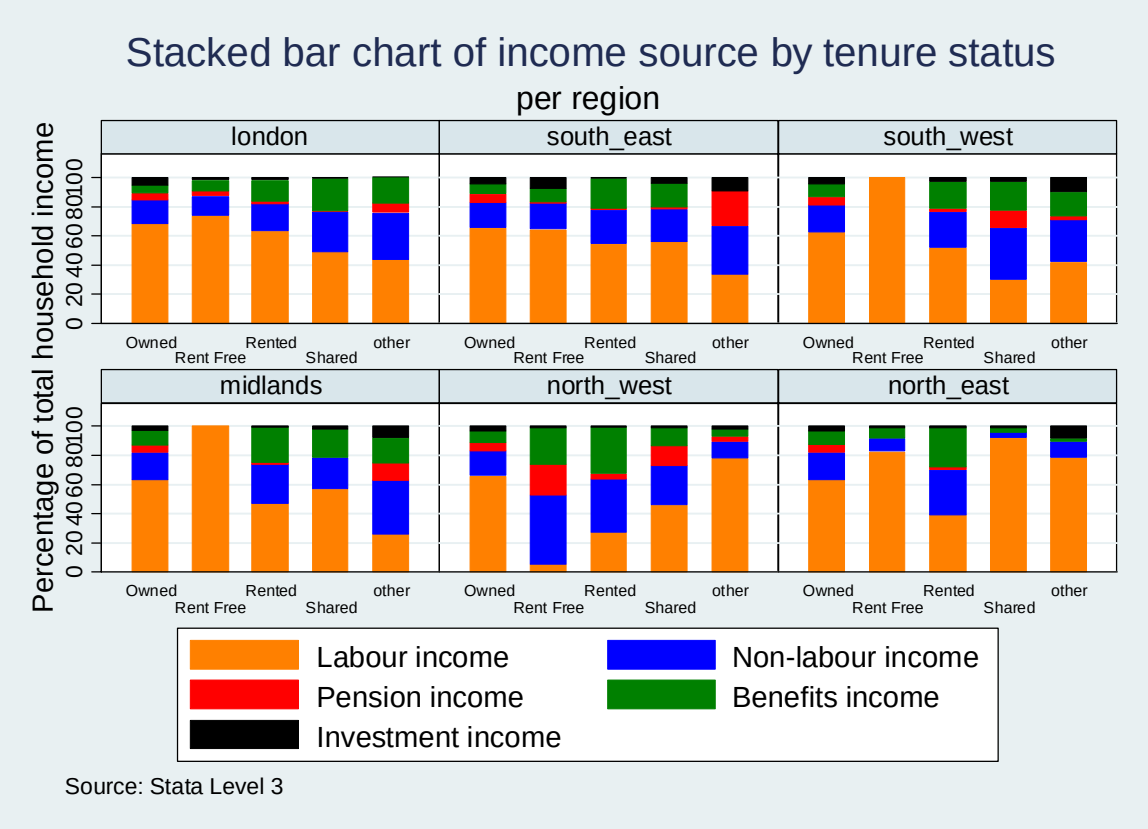
In the graph above a bar is produced for each category of the tenure variable rather than simply producing a single bar for all cases combined: this is done by specifying that the graph should be made 'over' the tenure variable – i.e. over(tenure). However, another thing which we may wish to do is to produce a separate graph for each category of a variable, rather than to just produce a separate bar as here – and this can be done by specifying that the graph should be made 'by' a particular variable.

In the following example we reproduce the above graph of percentage of household income from different sources over tenure types **but in this case we specify that we wish to produce a separate plot for each region in our dataset** – i.e. by(region) (and over(tenure) within each region). As seen in a previous example, changing the labels or label size on the x-axis which relate to the categories of the 'over' variable are specified within the brackets relating to the 'over' variable. **Note too that when graphs are produced 'by' a variable as here, any title, subtitle and notes should be specified within the brackets relating to the 'by' variable** – if these are specified as separate options as in the previous examples then they will be reproduced **for**

each of the separate plots, but if they are specified within the 'by' variable brackets as here then they will appear **only once** in the overall graph.

The graph also has relabelled categories on the x-axis (relabel option); has a compact style to try and leave more space for the actual plots; we change the legend label to tidy up the labels on the x-axis; and we add a title to the y-axis. The if-statement restricts the data displayed to households with up to three persons in employment and four types of tenures. The syntax to do this is as follows:

```
graph bar (mean) inc_lab inc_nonlab inc_pens inc_bens inc_inv ///
if tenure!=. , ///
over(tenure, relabel(1 "Owned" 2 "Rent Free" 3 "Rented" 4 "Shared") ///
label(labsize(*0.7) alternate)) ///
by(region, title("Stacked bar chart of income source by tenure status") ///
subtitle("per region") note("Source: Stata Level 3") ///
style(compact)) percentage stack nofill    ///
 bar(1,color(orange)) bar(2,color(blue)) bar(3,color(red)) ///
bar(4,color(green)) bar(5,color(black))    ///
legend(label(1 "Labour income") label(2 "Non-labour income") ///
label(3 "Pension income") label(4 "Benefits income") ///
label(5 "Investment income")) ///
ytitle("Percentage of total household income")
```

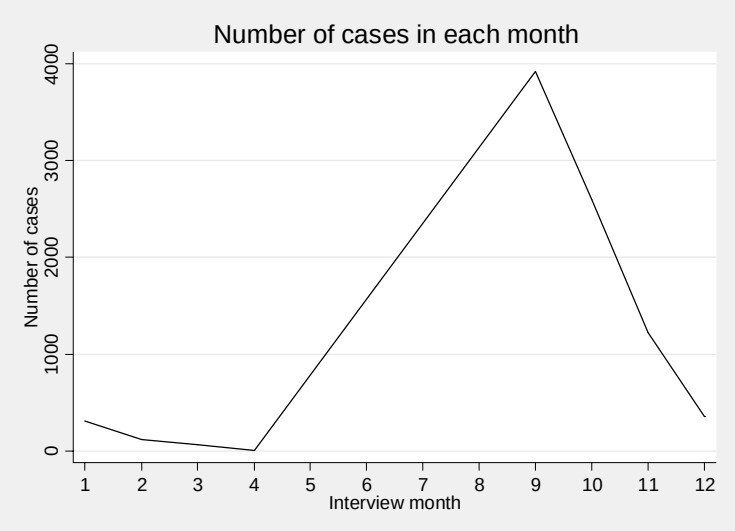


3.6. Line graph

Line graphs are typically used to present changes in data over a period of time, for example a company might plot the number of sales per week where they have data for a number of weeks. In the following graph we plot the number of cases in the dataset interviewed in each month. Whilst this could equally well be presented using a bar graph here we will instead use a line graph.

First we make a simple count variable telling us how many cases there were interviewed in each month. In the main graph syntax it can be seen that a line graph is one of a large family of graphs known as **twoway** graphs in Stata (i.e. two variables are plotted against each other). In this example, we plot the number of interviews per month on the y-axis and the interview month on the x-axis. We also add a main title, titles to the two axes, and specify that we wish the labels along the x-axis to be the numbered 1-12 for each of the twelve months:

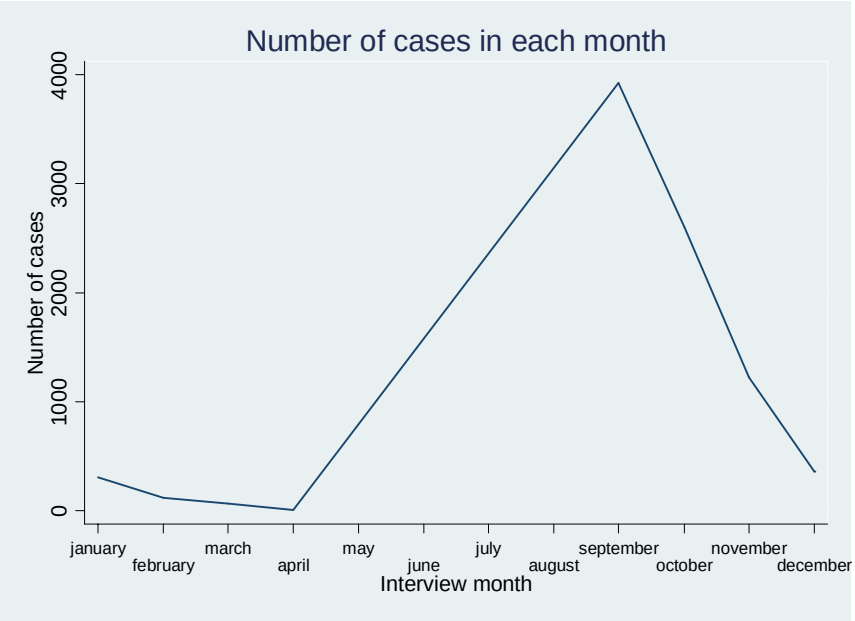
```
bys int_month: egen month_count= count(int_month)

graph twoway line month_count int_month, ///
title("Number of cases in each month") ///
ytitle("Number of cases") ///
xtitle("Interview month") xlabel(1/12)
```



Instead of showing the values along the x-axis we may prefer instead to use the actual value labels. In the line graph below the xlabel option is amended slightly so as to use value labels rather than the data values themselves, and these value labels are alternated along the x-axis and reduced to 80% of their original size to prevent them from overlapping:

```
graph twoway line month_count int_month, ///
title("Number of cases in each month") ///
ytitle("Number of cases") xtitle("Interview month")  ///
```

```
xlabel(1/12, valuelabel alternate labsize(*0.8))
```



Note that you can use, the loptions to make the line dotted, and control its width: explore further: `lcolor(orange) lwidth(thick) lpattern(dash)`.

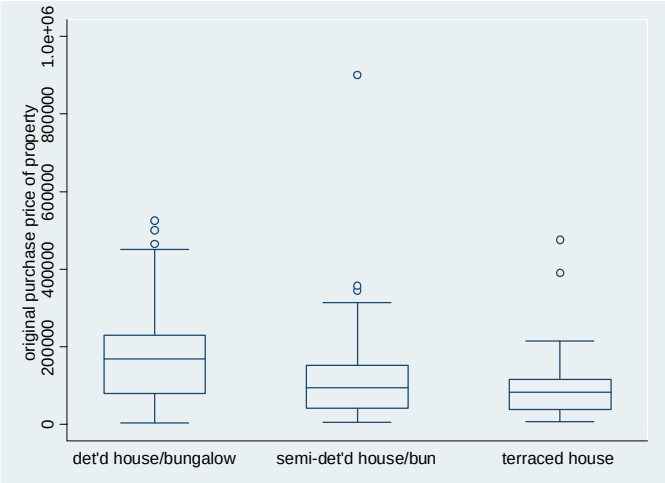
3.7. Box and whisker plot

Box and whisker plots are useful as they allow a visual picture of some key elements of a continuous variable's range of values. The median, the quartiles, and the maximum and minimum are five positions often used to describe center and spread. This summary lies at the heart of box plots. Thus, a box and whisker plot is typically used to show the median value of a variable as well as its interquartile range and its range.

In the following example we produce a box plot of the cost at purchase of three different property types (i.e. 'over' house type) – detached houses, semi-detached houses, and terraced houses. We do not add any other formatting to the graph.

```
/***** box & whisker plot – purchase cost for detached, semi-detached and terraced houses ****/
graph box hhcost if (house_type==1 | house_type==2 | house_type==4),  /// over(house_type)
```

In this box plot, shown below, the blue filled box represents the upper and lower bounds of the interquartile range whilst the horizontal line within this area represents the median value. The vertical lines coming from the box ('whiskers') represent the majority of the remaining range of values and the dots beyond the ends of the whiskers represent extreme outliers. Outliers are calculated as values exceeding p75+1.5*IQR or less than p25-1.5*IQR where p75 and p25 are the 75th and 25th percentiles respectively and IQR is the interquartile range. The whiskers mark the last observations that are not outliers.

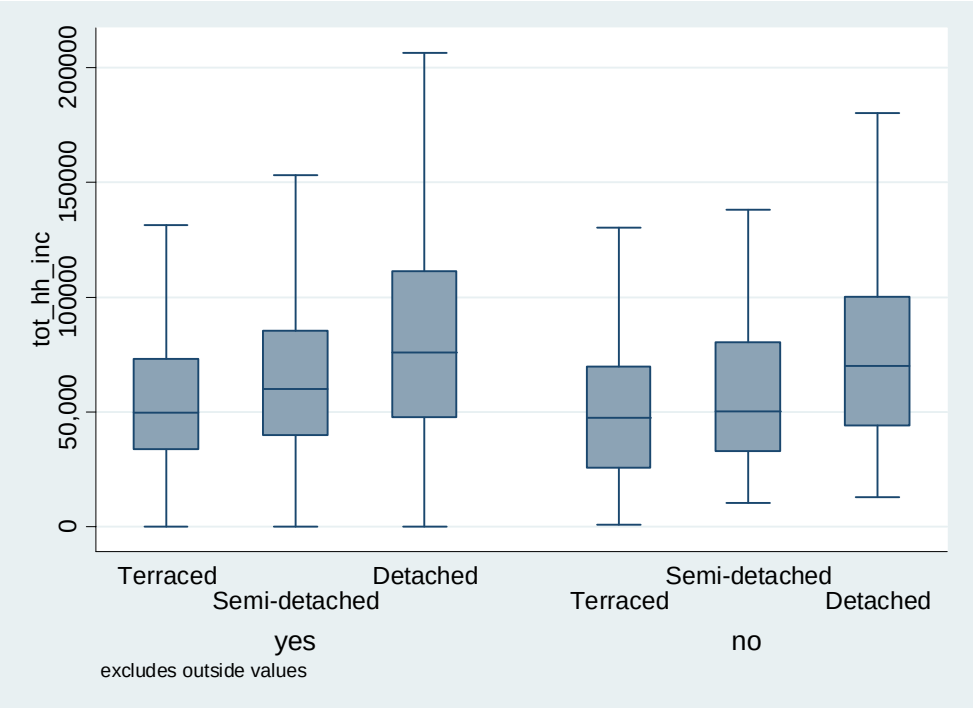


Additionally, it is possible to produce Stata graphs 'over' more than one variable at a time. For example, we may wish to plot the box plot of total household income over both house type and whether the house has a garden or not, as we may think that there is likely to be variation due to both factors. The next box plot produces this graph, relabelling the house type labels along the x-axis using the relabel option within the over brackets relating to house type, and alternating the

positioning of these house type labels using the alternate option (otherwise our graph could become way too heavy). The sort(1) option within the over brackets relating to house type requests that bars be presented in ascending order according to the median value of total household income of each of the three house type categories. The nooutsides option excludes extreme outliers from the box plot and this often helps with the scaling of the graph. Note that both over variables have separate over statements.

Interpretation. The graph shows that (as one would expect) that the median total household income is highest for the respondents living in detached houses, although it also shows that on average the ones without a garden tend to have higher incomes. Additionally, the interquartile range is noticeably wider for the detached house group which points to a wider income distribution within this group than within the other house type categories.

```
 /* two over groups specified and no extreme outliers shown */
graph box tot_hh_inc if (house_type==1 | house_type==2 | house_type==4), ///
over(house_type, relabel(1 "Detached" 2 "Semi-detached" 3 "Terraced") ///
label(alternate) sort(1)) ///
over(garden) nooutsides
```

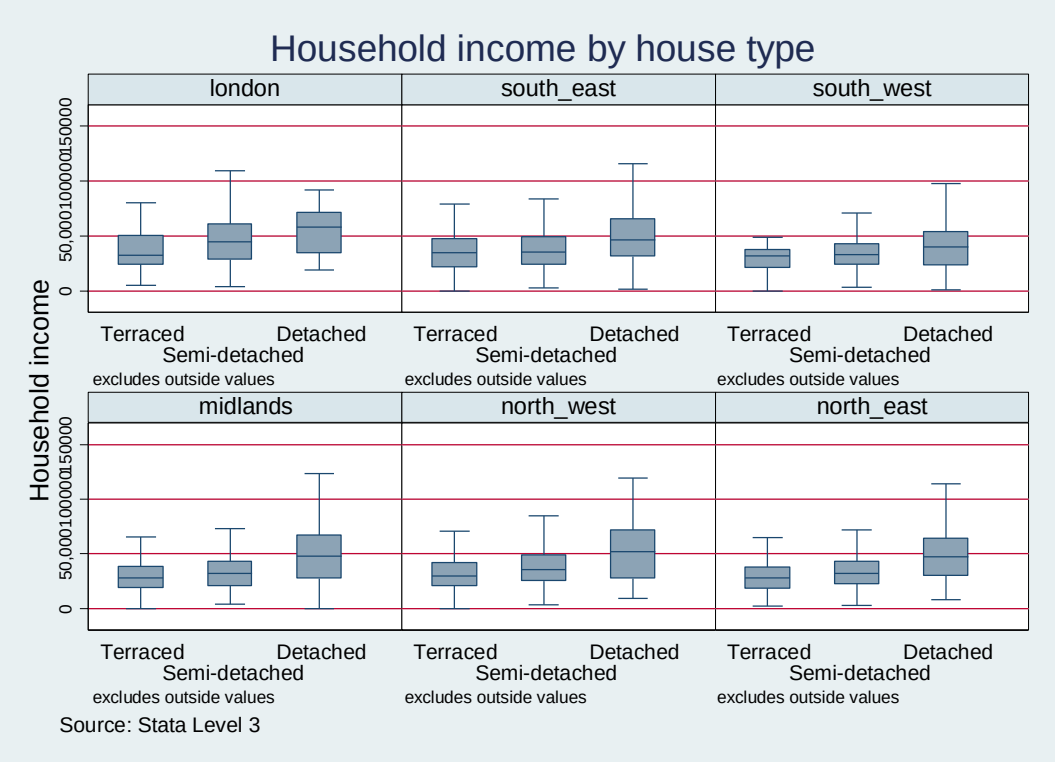


Again, a 'by' option can be introduced if we wished to plot a separate box plot for different categories of a variable. In the following example separate graphs are plotted for each region (i.e. by(region)).

- the style is set to compact in order to maximise the space available for the plots
- extreme outliers are excluded with the nooutsides option

- house type categories which are not included in the command or which have no values do not have empty bars made for them (the nofill option)
- finally, the y-axis is given a title, the labels on the y-axis are specified and horizontal lines across the graphs from the y-axis are requested from 25,000 to 125,000 at intervals of 25,000

```
graph box tot_hh_inc ///
if (house_type==1 | house_type==2 | house_type==4), ///
over(house_type, relabel(1 "Detached" 2 "Semi-detached" 4 "Terraced") ///
sort(1) label(alternate)) ///
by(region, title("Household income by house type") ///
note("Source: Stata Level 3") style(compact)) nooutsides nofill ///
ytitle("Household income") ///
ylabel(0(50000)150000, labsize(small))  ///
yline(0(25000)150000)
```



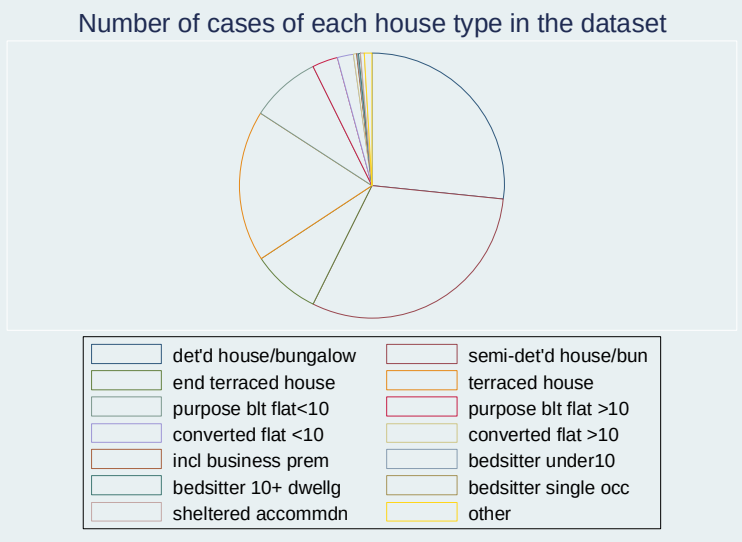
The graph shows a large amount of information for each house type in each of the regions. Naturally, the size of each of the plots is much smaller when producing graphs 'by' variables and so it is necessary to also consider whether the graphs can be understood clearly.

3.8. Pie chart

Visually presenting data in pie charts can be a simple way of conveying information, particularly when that information relates to shares of a total from different sources (in the way that a stacked bar chart is often used).

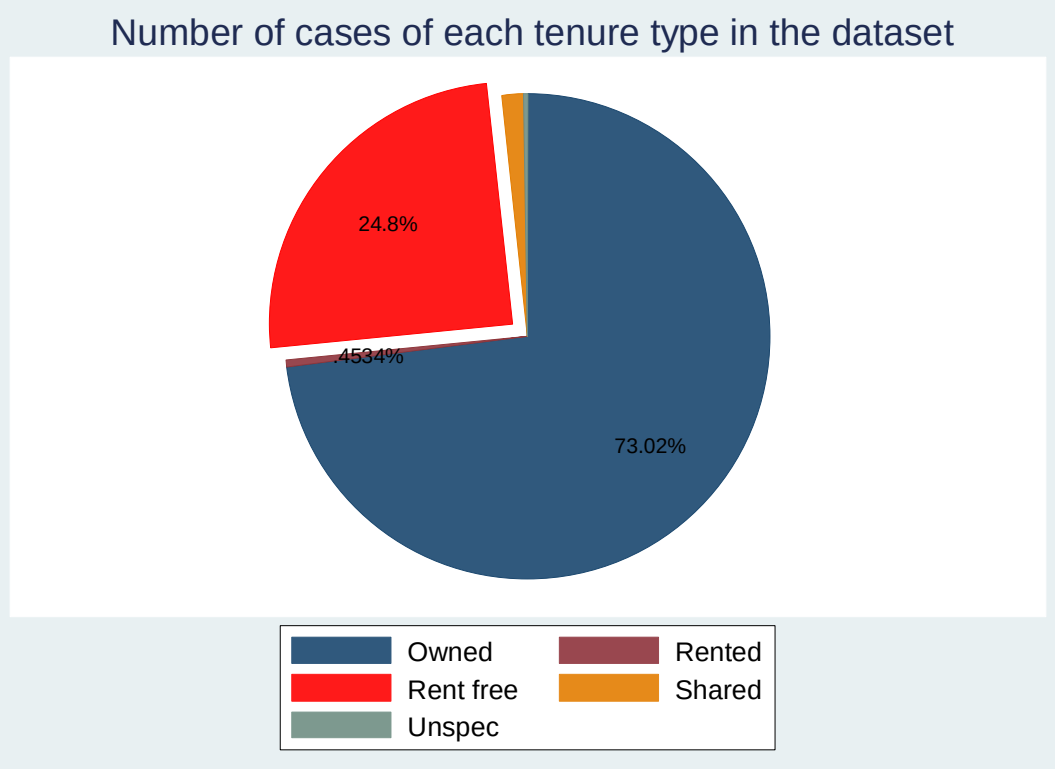
If we wished to make a pie chart of the number of cases in each house type category with a title then the syntax would be very simple. Note that in this example we do not actually specify a main variable – the command understands what we want and produces the graph of frequencies in each pie as we wished.

```
graph pie, over(house_type) ///
title("Number of cases of each house type in the dataset")
```

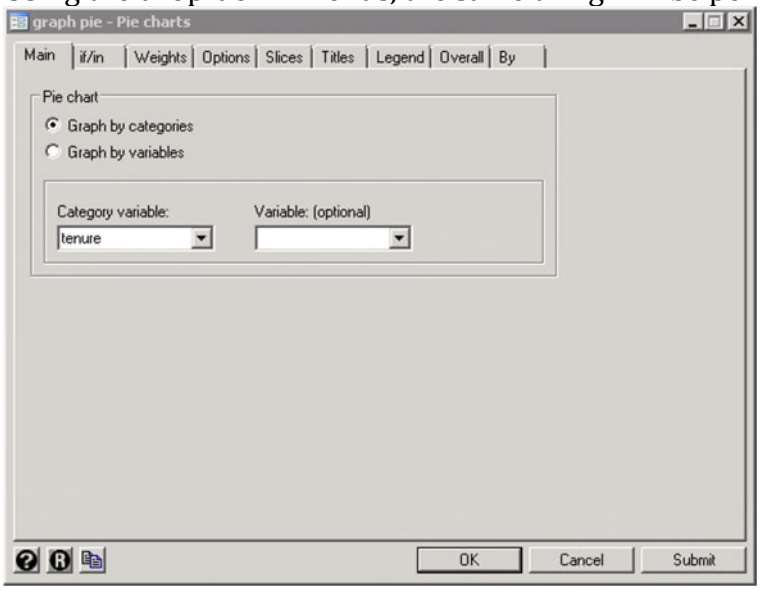


The next example reproduces this graph but adds in some additional formatting options: plabel is used to select the slice of the pie which you wish to format and then to specify what you wish the slice to display, where frequencies (freq) and percentages (percent) are the most commonly used options. In this syntax we ask that slices 1, 2 and 3 reveal their percentages. The pie option is also used to add certain formatting options to particular slices of the pie chart: in this syntax we specify that we would like to adjust the formatting of slice 3 so that it is red and is exploded out of the main pie chart. The legend option, as in all graphs, is typically used to alter the labels in the legend but can also for instance, alter its position on the graph:

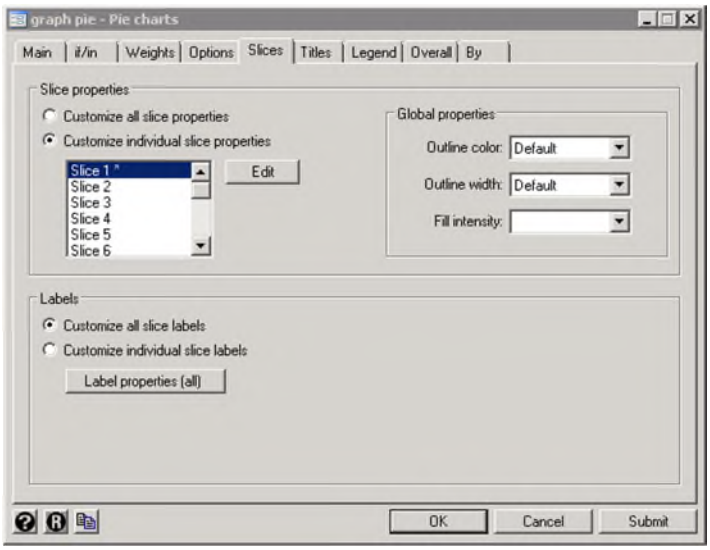
```
/* Number of cases by tenure, with title, slices showing percentages, slice
3 exploded and in red, and tidy legend titles */
graph pie, over(tenure) ///
title("Number of cases of each tenure type in the dataset") ///
plabel(1 percent)  plabel(2 percent)  plabel(3 percent) ///
pie(3, explode color(red)) ///
legend(label(1 "Owned") label(2 "Rented") label(3 "Rent free")  ///
label(4 "Shared") label(5 "Unspec"))
```

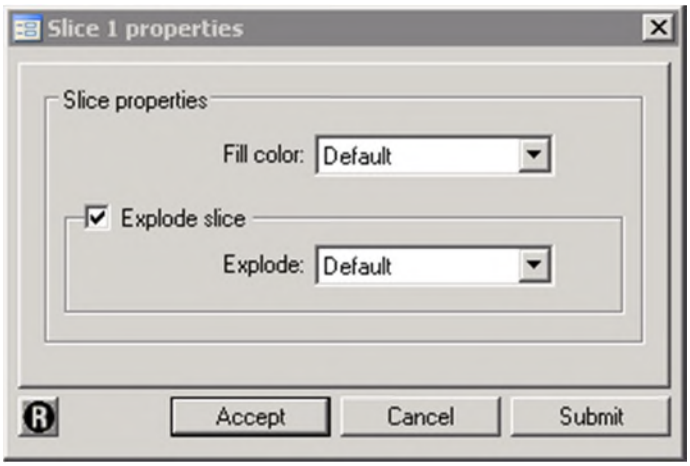


Using the drop-down menus, the same thing will be performed as:

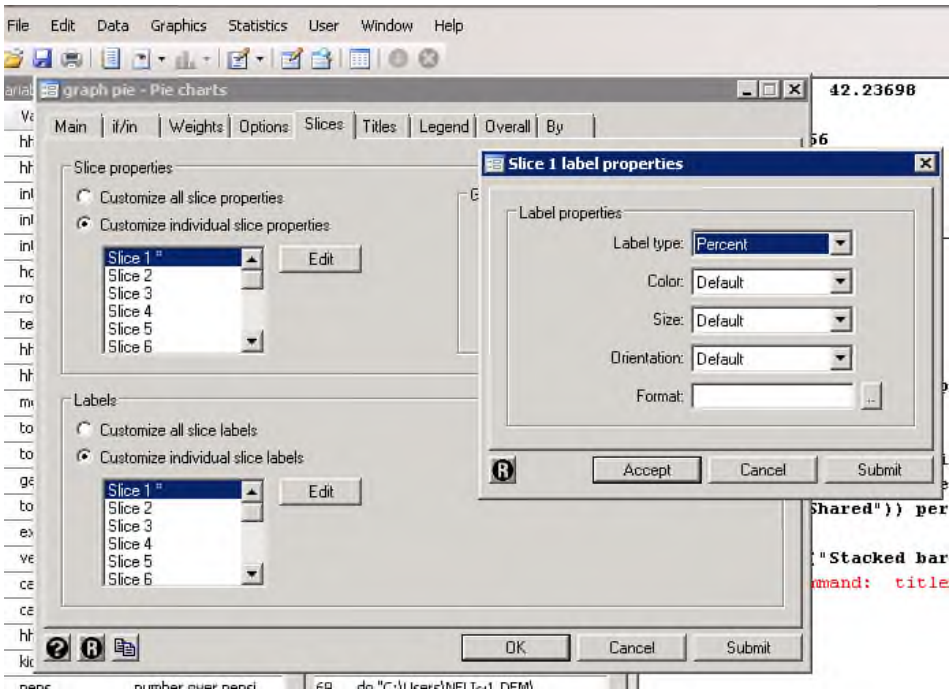


To choose one slide to explode, you have to go again to Options, choose the exploding slide and edit it:





And for percentages:

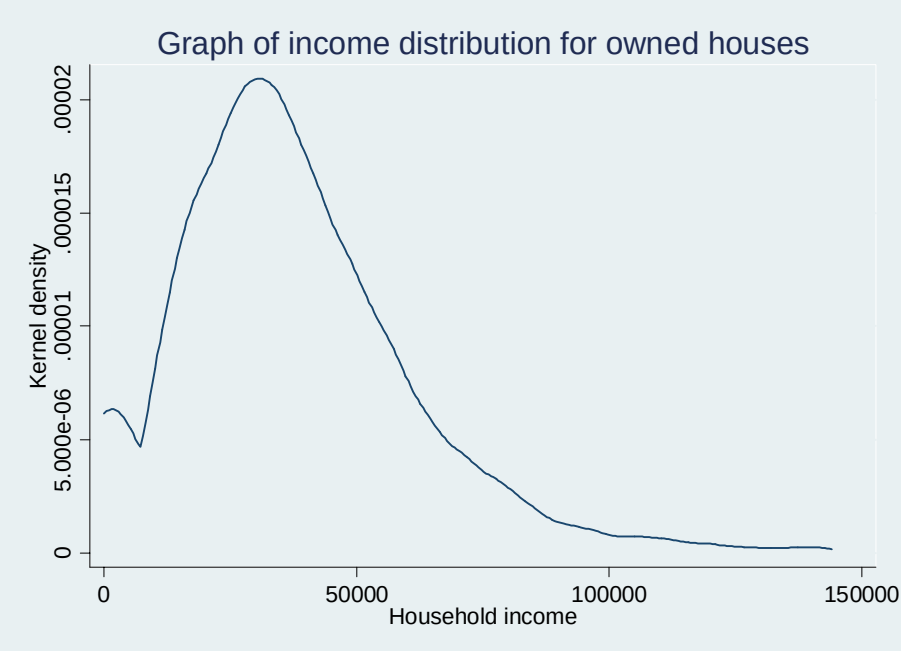


3.9. Kernel density plot

Kernel density plots are another member of the twoway family of Stata graphs and produce density line plots marking a variable's distribution. In the following graph we plot a kernel density graph of total household income only for cases where the house is owned. Extreme outliers act to stretch out the x-axis and effectively compress the 'main' area of the data where the vast majority of the data lie into a smaller space. For this reason the following graph only includes cases where total household income is below 150,000 and this is 99% of cases. Other formatting options relating to titles and label which are by now familiar are also specified:

```
graph twoway kdensity tot_hh_inc ///
if tenure== "owned" & tot_hh_inc<150000, ///
title("Graph of income distribution for owned houses") ///
ytitle("Kernel density") xtitle("Household income")
```

Interpretation. The kernel density plot takes the data and displays the frequency of cases at each value in the style of a line chart: where the curve is far from the x-axis this means that the density of cases is high at this value, in other words that there is a relatively large number of cases in the data at this value in terms of the proportion of the total number of cases in the data (which the area under the curve represents). Where the curve is close to the x-axis this means that there is a low density of cases at this value in the data (e.g. incomes greater than 100,000 in this example). What this kernel density plot shows is that the bulk of the graph's total population (i.e. owned houses with incomes below 150,000) have incomes of around 35,000 and that the majority of the cases have incomes between around 20,000-50,000. There are relatively few cases with incomes larger than around 90,000. There is also an interesting dip in density at very low levels of income, which would warrant closer examination.



3.10. Overlaying multiple graphs on the same plot

The kernel density plot above presented data for just one category – houses which were owned. It is also possible with Stata graphics to lay **multiple** plots on top of each other in the same graph area. This can be done with many graph types and this example shows this feature with the kernel density plots.

In the following graph we place four kernel density plots of total household income for different house tenures onto the same graph area. Most of the options have been seen previously and will not be discussed further.

Several plots are added onto the same graph by enclosing the kdensity command into brackets, and repeating it for the separate graphs, as shown below. Common features, such as the graph title, axis titles and legend only need to be specified once.

The other options within the brackets are specific to each separate graph, as the if statements. It would also be possible to specify colours and line pattern.

```
graph twoway (kdensity tot_hh_inc if tenure== 1 & tot_hh_inc<150000  ///
/* 1st graph - owned*/ ,  ///
legend(label(1 "Owned") label(2 "Rent free") label(3 "Rented") ///
label(4 "Shared") label(5 "Unspec"))  ///
title("Graph of income distribution by tenure type") ///
ytitle("Kernel density") xtitle("Household income") )  ///
( kdensity tot_hh_inc if tenure==2 & tot_hh_inc<150000   ///
 /* 2nd graph laid on top - rent free*/ ) ///
( kdensity tot_hh_inc if tenure==3 & tot_hh_inc<150000 ///
 /* 3rd graph laid on top - rented*/  )  ///
( kdensity tot_hh_inc if tenure==4 & tot_hh_inc<150000 ///
/* 4th graph laid on top - shared*/ )
```

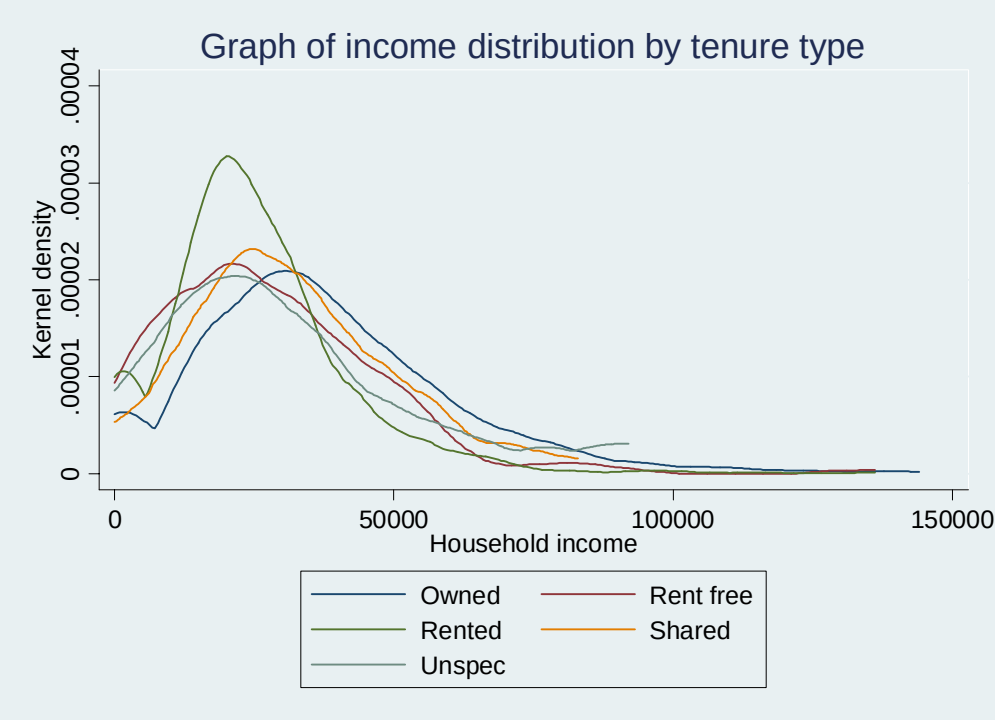
Another option is to use vertical lines (||) between the separate graph commands, instead of the brackets (see below).

```
graph twoway kdensity tot_hh_inc if tenure== 1 & tot_hh_inc<150000  ///
/* 1st graph - owned*/ ,  ///
legend(label(1 "Owned") label(2 "Rent free") label(3 "Rented") ///
label(4 "Shared") label(5 "Unspec"))  ///
title("Graph of income distribution by tenure type") ///
ytitle("Kernel density") xtitle("Household income")   ///
|| kdensity tot_hh_inc if tenure==2 & tot_hh_inc<150000   ///
 /* 2nd graph laid on top - rent free*/  ///
|| kdensity tot_hh_inc if tenure==3 & tot_hh_inc<150000 ///
```

```
 /* 3rd graph laid on top - rented*/    ///
|| kdensity tot_hh_inc if tenure==4 & tot_hh_inc<150000 ///
/* 4th graph laid on top - shared*/
```

In contrast to the previous example, the graph which is produced contains four kernel density plots, each of which is based on one of the four blocks of graph syntax above.

Interpretation. The graph shows that of all the tenure categories there is a higher proportion (density) of rented households with lower household incomes compared to the other tenure groups. Unsurprisingly, the blue curve relating to houses which are owned is the highest curve at higher levels of household income, meaning that a higher proportion of owned houses have relatively larger incomes compared to the other tenure groups.



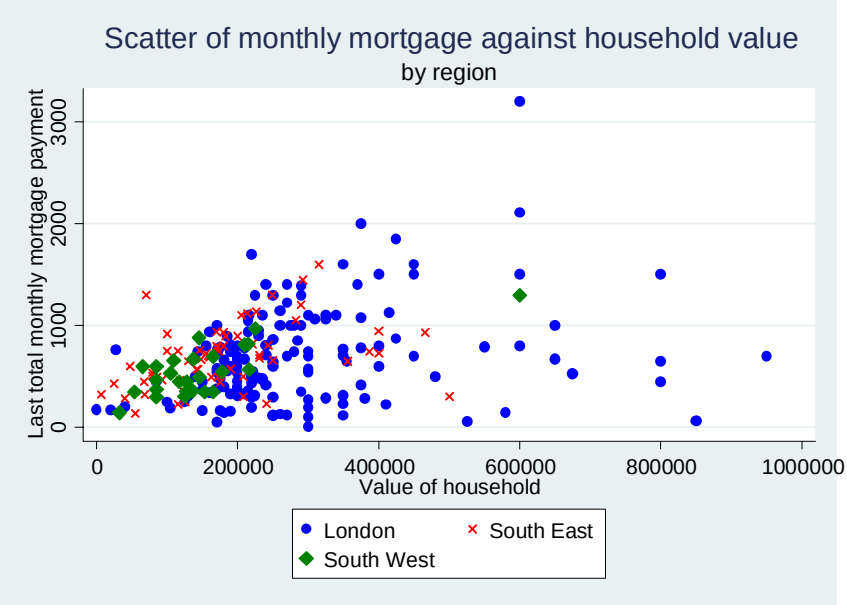
3.11. Scatter graphs

The final graph type covered in this course book is another member of the **twoway** family of Stata graphs: **scatter graphs.** Scatters are a useful way to plot the values of two interval variables against each other and can be understood as a visual way to approach correlations between variables. For example, we could scatter income in one year against income in the next year to see how stable our data are over time (i.e. how close to a 45 degree line does the data appear?).

In this first example we scatter monthly mortgage and household value and, as in the previous example, we effectively make multiple graphs and lay them on top of each other on the same graph area. The first block of syntax produces a scatter for all cases in region 1 (London) whilst the second block of syntax specifies the various title and legend label options for the graph area as a whole. Note that within this block of syntax we specify that we would like the points relating to this first plot (i.e. cases in London) to be blue – this is done using the **mlabel**(blue)option to change the colour of the markers. Common options for markers are mlabel for marker labels, **msize** for marker size, and **msymbol** for marker style (squares, triangles, circles, etc). Note again that each separate scatter command is enclosed by brackets, as explained above. This second graph is a scatter of the same variables for cases in region 2 (South East) and the mcolor option is specified in this plot to show these cases in red. The third graph shows cases in region 3 (South West) in green.

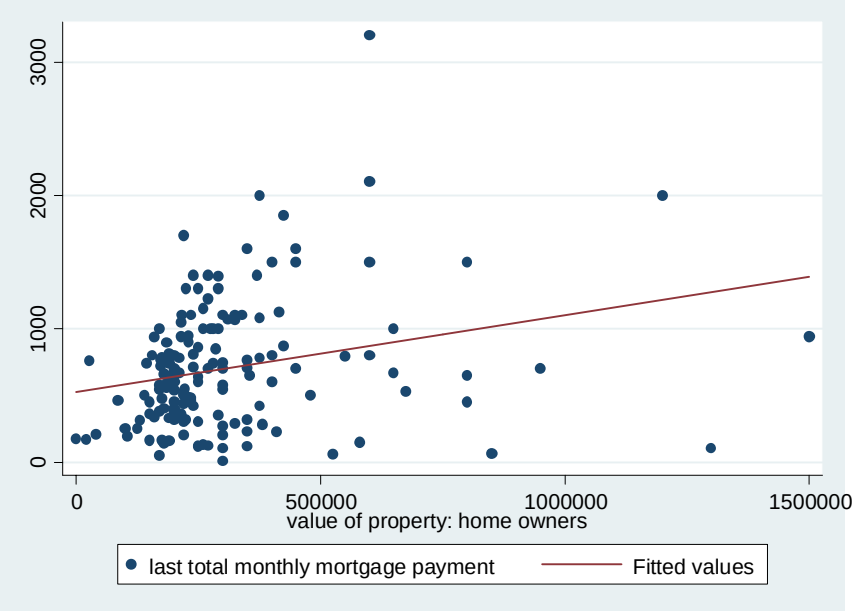
```
/* scatter of monthly mortgage vs household value for three regions
separately */
graph twoway (scatter monthly_mortgage hhvalue ///
 if region==1 & hhvalue < 1000000, mcolor(blue) ///
 title("Scatter of monthly mortgage against household value") ///
 subtitle("by region") ///
 ytitle("Last total monthly mortgage payment") ///
 xtitle("Value of household") ///
 legend(label(1 "London") label(2 "South East") label(3 "South West"))) ///
 (scatter monthly_mortgage hhcost if region==2, mcolor(red) msymbol(X)) ///
 (scatter monthly_mortgage hhcost if region==3,  ///
 mcolor(green) msymbol(diamond) )
```

Interpretation. The graph which is produced shows that most data points are clustered towards the bottom left of the plot with lower values both of properties and monthly repayments. There are relatively few cases to the right and the top of the plot – i.e. those with more expensive houses and with larger repayments – but those cases which are in found here are found in London. If we wished to check a particular case (for example, we may be worried about these outliers) then the mlabel option could be included so as to identify the outlier cases.



Different types of graphs can be combined using the twoway command. The following code will produce a scatter plot as before, but plot the line of best fit (lfit):

```
graph twoway (scatter monthly_mortgage hhvalue) ///
(lfit monthly_mortgage hhvalue) if region==1
```

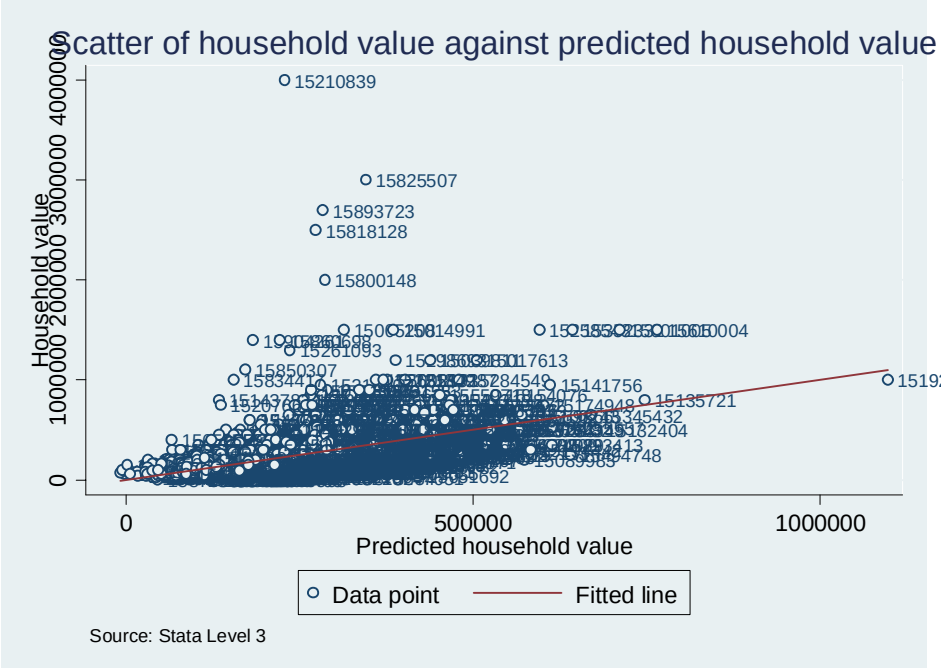


Similarly, this option can also be used following a regression to plot actual and predicted values and to include a line of best fit to the data points. Again this can be helpful in identifying unusual outlier cases which do not fit the model well. In the following example a linear regression is run and the predict postestimation command is specified to produce the predicted_hhvalue variable based on the model.

```
capture drop predict*
xi: regress hhvalue inc* rooms i.tenure
predict predicted_hhvalue
```

A scatter diagram is then produced which plots the actual against predicted values and adds a title, note and titles on the x- and y-axes. Marker labels are also requested. On the same graph area a separate lfit (line fit) plot is produced between these two variables and this plots the line of best fit for these data points. This can be a useful part of regression diagnostics to assess how well the model fits the data, how well actual and predicted values match, and which types of cases tend to be outliers.

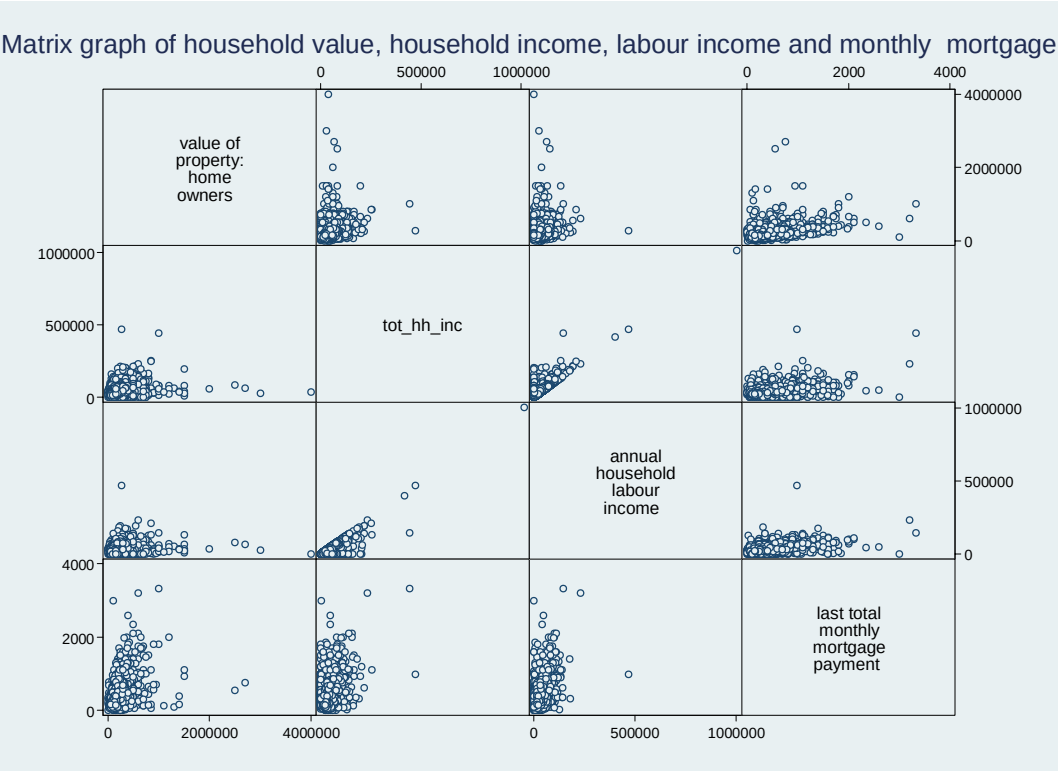
```
graph twoway scatter hhvalue predicted_hhvalue,  ///
title("Scatter of household value vs. predicted household value")  ///
note("Source: Stata Level 3")  ///
ytitle("Household value") xtitle("Predicted household value")  ///
legend(label(1 "Data point") label(2 "Fitted line"))  ///
mlabel(hhid)||  ///
lfit hhvalue predicted_hhvalue
```



3.12. Matrix scatters

Finally, a variant of scatter graphs is the matrix graph and this can be understood as a visual representation through scatter plots of a correlation coefficient table. It is best explained with an example. In the following example we produce a matrix graph of four variables: household value, total household income, labour income and monthly mortgage. We give the graph a title and reduce its scale to 70% of its original size – this is useful when labels or titles are too large for the graph area or are overlapping:

```
graph matrix hhvalue tot_hh_inc inc_lab monthly_mortgage, ///
 title("Matrix graph of household value, household income, labour
income and monthly mortgage") ///
 scale(*0.7) ///
 saving($graph\matrix scatter2)
```



The layout of the graph which is produced can best be understood by thinking about what the table of correlation coefficients between these four variables would look like. **Interpretation**. Across the diagonal are the variables included in the graph, with x- and y-variables of each mini-scatter plot changing according to the location on the graph. For example, if cell two is the left-most graph on the top row then this can be seen to be a scatter of household value (y-axis) against total household income (x-axis). Cell three (the central scatter of the three on the top row) is a scatter of household value (y-axis) against labour income (x-axis). Likewise, the central scatter of the three on the bottom row relates to total household income (x-axis) against last total monthly mortgage payment (y-axis).

Matrix graphs can be a useful way to look at and convey the bivariate relationships between multiple variables in a single graph.

3.13. Saving and opening graphs

Graphs are saved in the same way as data files except that they have the .gph ending rather than .dta. To save the matrix scatter we have just made into our work folder we would type:

```
graph save "$work\matrix scatter.gph",replace
```

It is **not possible** to use the /// break in file paths. It is also possible to add the saving option to the syntax in order to save the graph from within the syntax to make the graph.

Once this graph is saved it can be opened again either by double clicking it from within the folder or by typing:

```
graph use "$work\matrix scatter.gph"
```

The above graph save command saves the graphs in the Stata graph format, and Stata will have to be opened to view the graph.

It is also possible to save the graphs in other graphics formats, such as .png. The following commands are used for this:

```
graph export "$graphs\matrix scatter.png",replace
```

3.14. Editing graphs

This course has focused on writing the underlying syntax to flexibly produce a range of graphs. It can be seen that with this flexibility in graph specification can come a relatively dense graph syntax containing various graph options. It can be frustratingly difficult to manipulate the syntax correctly in order to make what you would consider to be relatively simply changes to the graph until you are comfortable with writing syntax for graphs. Indeed, graphics might even be an area where people may wish to experiment with using menus and then once run to right-click the review window to get the syntax into a .do file. The advantage of getting to grips with graph syntax is that once the syntax is correctly specified it is possible to reproduce this (or similar) graphs quickly. It is possible to click with the right button on the graph after producing it and to make manual changes to the graph (with the Graph Editor), although these manual changes do not make their way into the .do file. We will have a standard toolbar, a contextual toolbar like in any picture modifying tool. This can be helpful although these manual interactions are inevitably slower and less easy to replicate compared with proficient syntax writing.

Exercise 4 Stata Graphs (15 mins)

- *Use the bp_data.dta saved in "H:\StataLevel3\Raw data" (fictional data)*
- *Apply some of the commands to produce and modify graphs*

Task 1

- Create a horizontal bar chart of the mean follow-up blood pressure data (bp_after) for each drug type (trt).

  Hint: examples of horizontal bar charts are provided in section 3.3.

- Make your graph more user-friendly by giving your graph a title. *Hint: use the 'title' option.*

Task 2

- Use a box and whisker plot to show how the follow-up blood pressure (bp_after) is associated with the type of drug received (trt). *Hint: examples of box and whisker plots are shown in section 3.7.*
- Change the code to label the drug categories on the x-axis (drug B should be labelled "BBB", drug A "AAA"), add a title for the graph and a title on the y-axis, and do not show extreme outliers.

Task 3

- Create a scatter plot of baseline blood pressure (bp_before) on the x-axis against follow-up blood pressure (bp_after) on the y-axis. *Hint: examples of scatter plots are shown in section 3.11.*
- Then show patients on drug B with green markers, and as "X" and show patients on drug A with red markers and as diamonds.

  Hint: You are effectively trying to plot two graphs laid on top of each other. Information on overlaying graphs is provided in section 3.10.

- Also add a graph title, titles on the y-axis and x-axis, and rename the legend labels to correspond to the two types of drugs received.

Task 4

- Produce a bar graph (vertical bars) showing baseline (bp_before) and follow-up blood pressure (bp_after) for each drug allocation.
- Add a title, amend the legend to using the labels "Baseline BP" and "Follow-up BP", respectively. Also label the y-axis "Blood pressure values".
- Add the relevant code to turn the first bar grey and the second bar black.
- Remove the colour options, and see how the graph changes if you use the options scheme(s1mono) and scheme (s2mono).
- Finally, save the graph in "H:\StataLevel3\Graphs" in .png format. Name the file bp_graph.
  Hint: Information about saving graphs can be found in section 3.13.

4 Appendices

4.1. Solutions to Exercises

Solutions to the exercises will be provided at the end of the course.

4.2. do file for the session

```
/*set up log file library*/
global log "H:\StataLevel3\log"

/*open log file*/
log using "$log\Stata Level 3 stats, survey & graphics log file.log",replace

/*Stata Level 3 - Statistics, survey analysis and graphics*/
/* 23 June 2011 */
/*Syntax file to accompany course booklet */

clear
set more off

/*set libraries for data*/
global raw "H:\StataLevel3\Raw data"
global work "H:\StataLevel3\work"
global final "H:\StataLevel3\Final"
global graphs "H:\StataLevel3\Graphs"

/********* Section 1: Statistical analysis in Stata ********/
use "$raw\bhps_for_class.dta", clear

/*** correlation ***/
/** listwise **/
/*correlate varname1 varname2*/
correlate tot_hh_inc inc_lab

*could also split up the calculations by gender:
bys vehicle_access: correlate inc_inv inc_lab

/** pairwise, including significance tests and observations for each entry **/
pwcorr inc_inv inc_lab
```

```
*pwcorr allows for additional options to be specified
*including the number of observations and the significance of each correlation
pwcorr  tot_hh_inc inc_lab inc_inv, sig obs

*pwcorr uses pairwise deletion, while correlate uses listwise deletion
*show:
preserve
replace inc_lab = . if _n > 8000
corr  hhsize inc_lab inc_inv
pwcorr  hhsize inc_lab inc_inv,  obs
pwcorr  hhsize inc_lab inc_inv, listwise obs
restore

*can also use stars as identifiers of statistical significance:
pwcorr  tot_hh_inc inc_lab inc_inv, sig obs star(0.05)

/*If the data has outliers which may affect the correlation coefficients or if the
data is ranked then Spearman's rank may be more appropriate*/
/*spearman varname1 varname2*/
spearman inc_lab inc_inv



/***** T-tests ******/
/*one sample t-test against a known value*/
summ tot_hh_inc, detail

*compare the mean of the distribution to the median
ttest tot_hh_inc==30930

*the level of significance used in the test can be changed:
ttest tot_hh_inc==30930, level(99.99)



*two-group t-test (with by group)
/*to compare whether the difference in means between two bygroups
is statistically significant*/
ttest tot_hh_inc, by(garden)

*the ttest command assumes equal variances -
* the unequal option needs to be specified if this may not be the case
ttest tot_hh_inc, by(garden) unequal
```

```
/*A two-sample t-test, for example to test if the mean of labour
income is significantly different from the mean of total income*/
ttest inc_lab==tot_hh_inc


/** oneway - tot_hh_inc across tenure groups */
/* look at the means of total household income for the tenure types */
table tenure,c(mean tot_hh_inc)

/* test significance of the different tenure types ANOVA*/
oneway tot_hh_inc tenure


/* oneway with bonferroni option - to report bonferroni adjusted significance */
oneway tot_hh_inc tenure, bonferroni



/********* Chi-square ********/
*use tabulate for categorical variables:
tabulate tenure garden

*add chi-squared option:
tabulate tenure garden, chi2 exact

tabulate tenure garden, chi2 exact expected

***EXERCISE 1

/***** Linear regression ******/
/*regress depvar indvar(s)*/
/*to have exp_food as a dependent and income and bedooms as independents*/
regress exp_food tot_hh_inc rooms

/*to include dummies for binary/categorical independent variables*/
regress exp_food tot_hh_inc rooms i.house_type

label list hh_type
tab house_type

/*to change the reference group of a categorical independent variable*/
regress exp_food tot_hh_inc rooms b3.house_type
```

```
/* interaction effects */
/* now run the regression with an interaction between the categorical
 numeric tenure variable and interval total household income variable */
regress exp_food i.garden tot_hh_inc i.garden#c.tot_hh_inc

*shorten the code - use ## for a full factorial of the variables, i.e. main
* effects and interactions
regress exp_food i.garden##c.tot_hh_inc

*old coding:
xi: regress exp_food i.garden*tot_hh_inc

xi: regress exp_food i.tenure*tot_hh_inc, level(99)

/******* Logistic regression ******/
logistic garden inc_lab hhsize total_mortgage
*error message: need to be coded appropriately!
label list gdn

/*recode garden to binary*/
recode garden (2=0 no) (1=1 yes) (.=.), gen(garden2)

/*to  report odds ratios*/
logistic garden2 inc_lab hhsize total_mortgage

/* categorical independent variables can again be included using xi */
logistic garden2 inc_lab hhsize total_mortgage i.house_type

/*to run the same model but report beta coefficients*/
logit garden2 inc_lab hhsize total_mortgage

/**** Postestimation commands ****/
/** predict **/
regress exp_food tot_hh_inc rooms
predict pred_exp_food, xb

help predict

/** test **/
regress exp_food inc* rooms i.tenure

/* test if the coefficients equal zero and test their joint
```

```
contribution to the model */
test inc_lab inc_nonlab inc_pens inc_bens inc_inv

/* test whether the coefficients equal each other */
test inc_lab=inc_pens

***EXERCISE 2


/******  Section 2: Analysing survey data in Stata  *******/
use "$raw\bhps_for_class.dta", clear

/*set up the data for survey analysis*/
/*To set up the data the syntax is: svyset PSU [pweight=weight variable],
 strata(strata variable if any) */
/* in this case the data has household weights and area as primary
 unit, there is no strata variable */
svyset area [pweight=hh_wt]


/* in this case the data has household weights and area as
primary sampling unit, there is no strata variable */

/*here we do not set strata as we do not have them, but it is
better to set them if possible*/

/*describe whether/how the data is set up for survey analysis*/
svydes

/*to clear all survey settings*/
svyset,clear

/*set again*/
svyset area [pweight=hh_wt]


/*** svymean - survey means ***/
svy:mean tot_hh_inc

summ tot_hh_inc

/**svy:prop - survey proportions**/
/*first generate a new poorflag variable which is based on survey mean
```

```
of the data and is numeric: 1=deprived, 0=not deprived*/
gen deprived_flag=0
replace deprived_flag=1 if tot_hh_inc<(0.6 * 34291)

/*** svy:prop - now calculate survey proportions ***/
svy:prop deprived_flag

/**combining survey commands over bygroups**/
svy:mean tot_hh_inc, over(region)

/* make regional poverty flag */
gen reg_deprived=0
replace reg_deprived=1 if tot_hh_inc < (0.6 * 40403) & region==1
/*london*/
replace reg_deprived=1 if tot_hh_inc < (0.6 * 37097) & region==2
/*south east*/
replace reg_deprived=1 if tot_hh_inc < (0.6 * 34991) & region==3
/*south west*/
replace reg_deprived=1 if tot_hh_inc < (0.6 * 32958) & region==4
/*midlands*/
replace reg_deprived=1 if tot_hh_inc < (0.6 * 35103) & region==5
/*north west*/
replace reg_deprived=1 if tot_hh_inc < (0.6 * 32724) & region==6
/*north east*/

/* and calculate regional proportions in poverty */
svy:prop reg_deprived, over(region)

/*** svy:total - survey totals ***/
/* survey total of kids per region */
svy:total kids, over(region)

/* survey total of poor kids per region according to poor_flag */
svy:total kids, over(deprived_flag region)

/*** svy:ratio - survey ratios ***/
/* child dependency ratio per region */
svy:ratio kids wage, over(region)

/* child development for the country as a whole */
svy:ratio kids wage

/*** svy:tab - survey tables ***/
```

```
/* one way survey table of household size */
svy:tabulate hhsize, count cell

/* with confidence intervals */
svy:tabulate hhsize, count ci

/* two way survey table */
svy:tab region garden, count cell row col

/*** svy:regress ***/
svy:regress exp_food tot_hh_inc rooms

/* using xi with survey regression to automatically make dummies for a categorical
explanatory variable */
svy: regress exp_food tot_hh_inc rooms i.tenure

svy: logistic deprived_flag inc_lab rooms i.tenure

/**** using subpop *****/
/* generate a kids_flag for the subpop option to take:  our interest is in
childless households so these take the value 1, missing vales are coded missing and
all other values are coded 0 */
gen kids_flag=.
replace kids_flag=0 if kids >0 & kids != .
replace kids_flag=1 if kids==0

/* now we can use the subpop option with svy:mean to calculate survey mean of total
household income focusing only on households without children, and with correct
estimates of variance. Here we combine subpop with the over(region) option to focus
on households without children in each region separately */
svy, subpop(kids_flag): mean tot_hh_inc, over(region)

/***** postestimation commands with survey analysis *******/
/*  using lincom to test whether the difference between means are statistically
significant between london and the north east */
svy:mean tot_hh_inc, over(region)

/* testing the difference of survey means - a t-test for survey data essentially */
svy:mean tot_hh_inc, over(region)
lincom [tot_hh_inc]london - [tot_hh_inc]north_east

/* test equality of means using test */
test [tot_hh_inc]london =  [tot_hh_inc]midlands

/* testing the difference between survey regession coefficients */
svy:regress exp_food tot_hh_inc rooms
```

```
svy:regress exp_food tot_hh_inc i.garden

***EXERCISE 3

*********************************************************************
* Section 3: Introduction to Stata graphics                         *
*********************************************************************
use "$raw\bhps_for_class.dta", clear
/***** histogram *********/
histogram rooms, percent

/* density is the default, and here with no titles or labels */
/*if you have a discrete variable that can take only integer values*/
histogram rooms, discrete
histogram rooms, percent

/* showing frequency of bins rather than density, with normal distribution
line plotted, with title, xtitle, ytitle and changing the colour of bins */
histogram rooms, freq discrete normal ///
 title("Histogram of number of rooms") ///
 xtitle("Number of bedrooms") ytitle("Frequency") ///
 fcolor(red) lcolor(black)

/*if you want to plot a normal distribution just add normal  without comma*/
/****  bar graph ****/
graph bar (mean) hhvalue, over(region)
graph bar (max) hhvalue, over(region)

/* bar graph with reduced label size & alternated labels, title, subtitle, note and
title on y-axis */
graph bar (mean) hhvalue, ///
 over(region, label(labsize(*0.75) alternate)) ///
 title("Mean house value")  ///
 subtitle("by region") ///
 note("Source: Stata Level 3") ///
 ytitle("Mean house price")  ///
 bar(1,color(green))

/* adding extra options to relabel bars on the x-axis, to add the mean figure to
the top of each bar, and adding y-lines at intervals of 50,000 */
 graph bar (mean) hhvalue, ///
 over(region, relabel(1 "London" 2 "South East" 3 "South West" ///
 4 "Midlands" 5 "North West" 6 "North East") ///
```

```
 label(labsize(*0.75))) ///
 title("Mean house value")  ///
 subtitle("by region") ///
 note("Source: Stata Level 3") ///
 ytitle("Mean house price")  ///
 blabel(bar) bar(1,color(orange)) ///
 yline(50000(50000)300000)

*plot more than one variable:
graph bar (mean) hhvalue inc_tot if house_type <=3, ///
 over(house_type, relabel(1 "detached" 2 "SEMI" 3 "End of terrace" ) ///
 label(labsize(*0.75))) ylabel(, labsize(small)) ///
 title("Mean house value and labour income")  ///
 subtitle("by house type") ///
 note("Source: Stata Level 3") ///
 ytitle("£")  ///
 blabel(bar) bar(1,color(orange)) ///
 yline(50000(50000)250000)

/* horizontal bar chart */
graph hbar (mean) hhvalue, over(house_type) ///
title("Mean house value by property type") ///
ytitle("Mean House Value") ///
blabel(bar)

/* stacked bar chart - stack option note that in this graph we
have very long x axis labels, so some further relabeling is needed */
graph bar hhvalue inc_lab, ///
 over(house_type, relabel(1 "1" 2 "2")) stack

/*example in textbook of stacked graph*/
graph bar (mean) inc_lab inc_nonlab inc_pens inc_bens ///
 inc_inv if tenure!=. & tenure!=5, ///
 over(tenure, relabel(1 "Owned" 2 "Shared ownership" 3 "Rented"  4 "Rent Free")) ///
 percentage stack ///
 title("Stacked bar chart of income source") ///
 subtitle("by tenure status") ///
 ytitle("Percentage of total household income") ///
 bar(1,color(orange)) bar(2,color(blue)) bar(3,color(red)) ///
 bar(4,color(green)) bar(5,color(black))

/*Stacked bar graphs example in course book*/
*adding a by option to reproduce earlier bar graphs separately by region
```

```
graph bar (mean) inc_lab inc_nonlab inc_pens inc_bens inc_inv ///
 if tenure!=. , over(tenure, relabel(1 "Owned" 2 "Shared ownership" ///
 3 "Rented"  4 "Rent Free") label(labsize(*0.7) alternate)) ///
 by(region, title("Stacked bar chart of income source by tenure status") ///
 subtitle("per region") note("Source: Stata Level 3") ///
 style(compact))percentage stack nofill ///
 bar(1,color(orange)) bar(2,color(blue)) bar(3,color(red)) ///
 bar(4,color(green)) bar(5,color(black))      ///
 legend(label(1 "Labour income") label(2 "Non-labour income") ///
 label(3 "Pension income") label(4 "Benefits income") ///
 label(5 "Investment income")) ///
 ytitle("Percentage of total household income")

/***** line graph *****/
bys int_month: egen month_count= count(int_month)

graph twoway line month_count int_month, ///
 title("Number of cases in each month") ///
 ytitle("Number of cases") ///
 xtitle("Interview month") xlabel(1/12) ///
 scheme(s2mono)

*use the actual value labels instead of numbers 1 to 12 & reduce label size
graph twoway line month_count int_month, ///
title("Number of cases in each month") ///
ytitle("Number of cases") ///
xtitle("Interview month") xlabel(1/12, valuelabel alternate labsize(*0.8))

/*using line options*/
graph twoway line month_count int_month, ///
lcolor(orange) lwidth(thick) lpattern(dash) ///
title("Number of cases in each month") ///
ytitle("Number of cases") ///
xtitle("Interview month") xlabel(1/12, valuelabel alternate labsize(*0.8))

/***** box & whisker plot - number of rooms for detached, semi-detached and
terraced houses ****/
graph box hhcost if (house_type==1 | house_type==2 | house_type==4), ///
over(house_type)

/* two over groups specified  and no extreme outliers shown */
graph box tot_hh_inc if (house_type==1 | house_type==2 | house_type==4), ///
over(house_type, relabel(1 "Detached" 2 "Semi-detached" 3 "Terraced") ///
```

```
 label(alternate) sort(1)) over(garden) nooutsides


/* same graph but now by each region separately, tidy up the labelling of the house
types (relabel), do not show non-specified house types (nofill), set y-axis labels
to show only 0, 50000 & 100000, and add lines`on the graphs at 25000 intervals */
graph box tot_hh_inc if (house_type==1 | house_type==2 | house_type==4), ///
 over(house_type, relabel(1 "Detached" 2 "Semi-detached" 4 "Terraced") ///
 sort(1) label(alternate)) ///
 by(region, title("Household income by house type") ///
 note("Source: Stata Level 3") style(compact)) ///
 nooutsides nofill ///
 ytitle("Household income") ///
 ylabel(0(50000)150000, labsize(small))  ///
 yline(0(25000)150000)

/** pie chart **/
/* Number of cases per house types */
graph pie, over(house_type) ///
 title("Number of cases of each house type in the dataset")

/* Number of cases by tenure, with title, slices showing percentages,
 slice 3 exploded and in red, and tidy legend titles */
graph pie, over(tenure)  ///
title("Number of cases of each tenure type in the dataset") ///
plabel(1 percent)  plabel(2 percent)  plabel(3 percent)  ///
pie(3, explode color(red))  ///
legend(label(1 "Owned") label(2 "Rented") label(3 "Rent free")  ///
label(4 "Shared") label(5 "Unspec"))

/*kernel density plots*/
*remove outliers in graph
graph twoway kdensity tot_hh_inc ///
 if tenure==1 & tot_hh_inc<150000, ///
 legend(label(1 "Owned") label(2 "Rent free") ///
 label(3 "Rented") label(4 "Shared") label(5 "Unspec")) ///
 title("Graph of income distribution for owned houses") ///
 ytitle("Kernel density") xtitle("Household income")

graph twoway kdensity tot_hh_inc if tenure==1 & tot_hh_inc<150000 ///
 /* 1st graph - owned*/ , ///
legend(label(1 "Owned") label(2 "Rent free") label(3 "Rented") ///
 label(4 "Shared") label(5 "Unspec")) ///
 title("Graph of income distribution by tenure type") ///
 ytitle("Kernel density") xtitle("Household income") ///
```

```
 || kdensity tot_hh_inc if tenure==2 & tot_hh_inc<150000   ///
 /* 2nd graph laid on top - rent free*/ ///
 || kdensity tot_hh_inc if tenure==3  & tot_hh_inc<150000 ///
  /* 3rd graph laid on top - rented*/ ///
  || kdensity tot_hh_inc if tenure==4  & tot_hh_inc<150000 ///
  /* 4th graph laid on top - shared*/



/***** scatters ********/
/* scatter of monthly mortgage vs household value for three regions separately -
this is an example of multiple plots laid one on top of the other */
graph twoway scatter monthly_mortgage hhvalue


graph twoway scatter monthly_mortgage hhvalue , ///
 title("Scatter of monthly mortgage against household value") ///
 ytitle("Last total monthly mortgage payment") ///
 xtitle("Value of household")

graph twoway scatter monthly_mortgage hhvalue ///
 if region==1 & hhvalue < 2000000, mcolor(blue) msymbol(O) ///
 title("Scatter of monthly mortgage against household value") //////
 subtitle("by region") ///
 ytitle("Last total monthly mortgage payment") ///
 xtitle("Value of household")

graph twoway (scatter monthly_mortgage hhvalue ///
 if region==1 & hhvalue < 2000000, mcolor(blue) msymbol(O) ///
 title("Scatter of monthly mortgage against household value") ///
 subtitle("by region") ///
 ytitle("Last total monthly mortgage payment") ///
 xtitle("Value of household") ///
 legend(label(1 "London") label(2 "South East") label(3 "South West")) ) ///
 (scatter monthly_mortgage hhcost if region==2, mcolor(red) msymbol(X) ) ///
 (scatter monthly_mortgage hhcost if region==3, mcolor(green) msymbol(diamond) )

*shorten code:
scatter monthly_mortgage hhvalue ///
 if region==1 & hhvalue < 2000000, mcolor(blue) msymbol(O) ///
 title("Scatter of monthly mortgage against household value") ///
 subtitle("by region") ///
 ytitle("Last total monthly mortgage payment") ///
 xtitle("Value of household") ///
 legend(label(1 "London") label(2 "South East") label(3 "South West")) || ///
```

```
 scatter monthly_mortgage hhcost if region==2, mcolor(red) msymbol(X) || ///
 scatter monthly_mortgage hhcost if region==3, mcolor(green) msymbol(diamond)

/*exploring goodness of fit*/
graph twoway (scatter monthly_mortgage hhvalue) ///
(lfit monthly_mortgage hhvalue) if region==1

/*examining predicted values*/
capture drop predict*
regress hhvalue inc* rooms i.tenure
predict predicted_hhvalue

graph twoway (scatter hhvalue predicted_hhvalue,  ///
 title("Scatter of household value against predicted household value")  ///
 note("Source: Stata Level 3")  ///
 ytitle("Household value") xtitle("Predicted household value")  ///
 legend(label(1 "Data point") label(2 "Fitted line"))  ///
 mlabel(hhid) )   ///
 (lfit hhvalue predicted_hhvalue )

/** scatter matrix **/
graph matrix hhvalue tot_hh_inc inc_lab monthly_mortgage, ///
 title("Matrix graph of household value & income, labour income & monthly
mortgage") ///
 scale(*0.7)  ///
 saving($graphs\matrix_scatter2, replace)



/**** opening and saving graphs ***/
graph use "$graphs\matrix_scatter2.gph"

*save a graph in a different format:
graph matrix hhvalue tot_hh_inc inc_lab monthly_mortgage, ///
 title("Matrix graph of household value, household income, labour income and
monthly mortgage") ///
 scale(*0.7)
graph export "$graphs\matrix scatter.png",replace
```

Stata: Statistical, survey and graphical analyses

Ines Rombach
courses@it.ox.ac.uk

UNIVERSITY OF OXFORD

IT Learning Programme

Today's arrangements

Your teacher is	Ines Rombach
Your demonstrator is	
We finish at	
You should have	Class notes Copies of slides

Your safety is important

Where is the fire exit?

Beware of hazards:
- Tripping over bags and coats

Please report any equipment faults to us

Let us know if you have any other concerns

Your comfort is important

- The toilets are along the corridor outside the lecture rooms
- The rest area is where you registered; it has vending machines and a water cooler
- The seats at the computers are adjustable
- You can adjust the monitors for height, tilt and brightness

Objectives for today's course

- Be familiar with univariate and multivariate analysis commands
- To able to use regression analysis in Stata
- Be familiar with survey commands in Stata
- Be able to produce a variety of Stata graphs

Options in the predict command

Option after predict	Option calculates
resid	Residuals
rstandard	Standardized residuals
rstudent	Studentized or jacknifed residuals
lev or hat	Leverage
stdr	Standard error of the residual
cooksd	Cook's D
stdf	Standard error of the predicted individual
stdp	Standard error of the predicted mean


services
UNIVERSITY OF
OXFORD
courses@it.ox.ac.uk