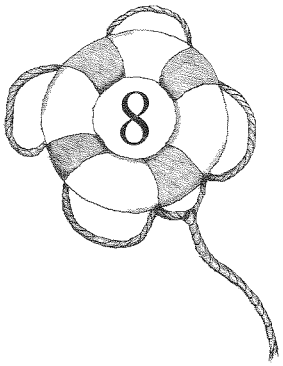


Regression

In this chapter we tackle how to conduct regression analyses in Stata. We concentrate on ordinary least squares (OLS) regression, which requires a reasonably normally distributed, interval level dependent variable, and logistic regression, which requires a dichotomous or binary dependent variable. We briefly introduce commands for multinomial logistic and ordered logistic regression models, the parallel family of commands for binary, multinomial and ordered probit, and Poisson or negative binomial models for a count dependent variable (see Box 8.1).

Among all of this we also look at the characteristics and effects of the independent variables. The majority of these techniques can be used on independent variables in any of the regression models mentioned above with more or less ease of interpretation! We will also discuss ways of dealing with categorical independent variables, non-linear associations, interaction effects, as well as regression diagnostics.

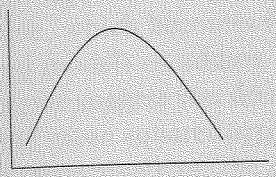
ORDINARY LEAST SQUARES REGRESSION

To carry out a bivariate regression use the **regress** (or **reg**) command, immediately followed by the *Y* (dependent) variable and the *X* (independent) variable.

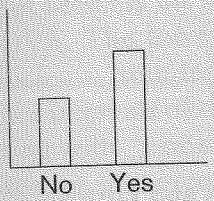
```
regress Y X
```

There are a number of variables in the example data set that are suitable for regression. In this example we will use monthly income (*fimn*) as our dependent variable, but considering only those in paid employment. We could use an **if** statement so that our regressions are only done where **jbstat==2**. Another way would be use a **keep** command so that only those people in employment

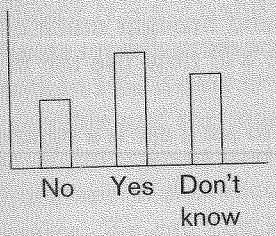
Box 8.1: Characteristics of dependent variables and choosing regression models



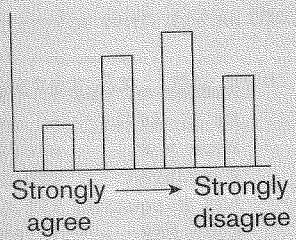
Ordinary least squares (OLS)



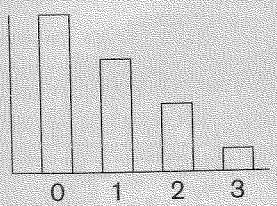
Binary logistic, logit or probit



Multinomial logistic, logit or probit



Ordered logistic, logit or probit



Poisson or negative binomial

are kept in the active data set: **`keep if jbstat==2`**. We will use the latter.

It is usual for income distributions to be positively skewed (or skewed to the right), in which case a natural logarithm transformation usually helps to bring the distribution closer to normality. As

we have done before in Chapter 5, we can check the skewness of the original income variable and the transformed variable:

```
gen ln_inc=ln(fimn)
tabstat fimn ln_inc, s(sk kur)
```

```
. gen ln_inc=ln(fimn)
. tabstat fimn ln_inc,s(sk kur)
   stats |      fimn      ln_inc
---------+------------------------
skewness |  2.473931   -.5381867
kurtosis |  18.37824    3.587047
----------------------------------
```

We can see from the output that the skewness and kurtosis of the variable has been considerably reduced and brought much closer to normality by the transformation. We will now use the *ln_inc* variable as our dependent variable.

First, we will use a bivariate regression to see if age is a significant determinant of income.

```
regress ln_inc age
```

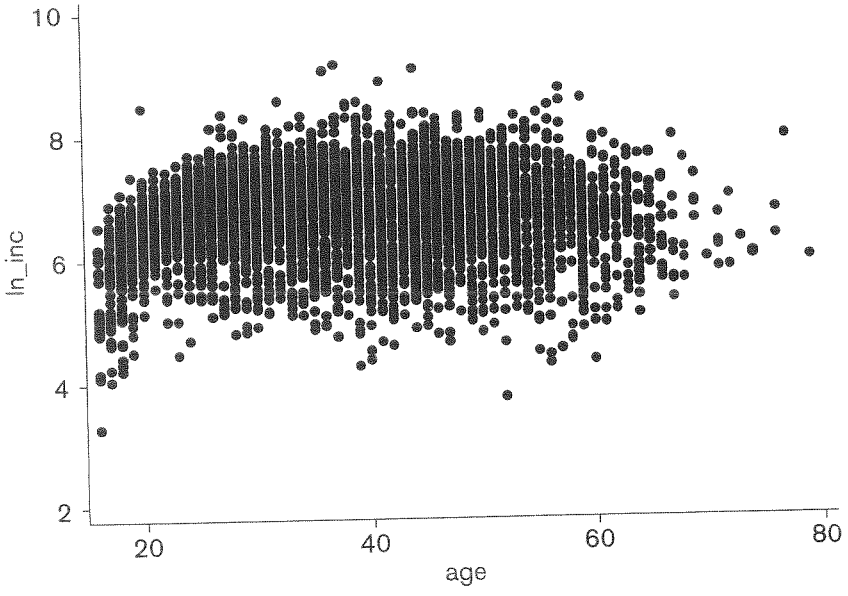
```
. reg ln_inc age
  Source |       SS       df       MS      Number of obs =    4973
---------+------------------------------   F( 1, 4971)   =   44.96
   Model |  21.1283285     1  21.1283285   Prob > F      =  0.0000
Residual |  2336.13556  4971  .469952838   R-squared     =  0.0090
---------+------------------------------   Adj R-squared =  0.0088
   Total |  2357.26389  4972  .474107781   Root MSE      =  .68553

------------------------------------------------------------------------------
  ln_inc |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
---------+--------------------------------------------------------------------
     age |   .0052946   .0007896     6.71   0.000     .0037465    .0068426
   _cons |   6.517582   .0313587   207.84   0.000     6.456105    6.579059
------------------------------------------------------------------------------
```

The regression output is relatively concise; check the equivalent output in SPSS if you don't believe us. In the upper right-hand side is the information concerning the number of observations used in the model as the **regress** command uses listwise deletion (i.e. only cases with non-missing values on all the variables in the model will be included) and model 'fit' statistics. The most commonly used 'fit' statistic in OLS regression is R^2 (`R-squared` on the output).

This indicates the amount of variance in the dependent variable explained by the independent variables; the higher the value, the more explanatory power the model has generally. Also in the upper right-hand corner are an F statistic and its associated p value, which becomes more useful when you are working with nested models or adding blocks of independent variables. At this stage, we suggest you do not to concern yourself with the adjusted R^2 and mean squared error statistics (`Adj R-squared` and `Root MSE` respectively on the output). The upper left-hand side of the output presents the sums of squares details as you would get from ANOVA. The lower panel of the output shows the regression coefficients, standard errors, t values, p values and 95% confidence intervals of those coefficients in each row. The bottom row starting with `_cons` is the intercept or constant for the model.

We see that even though the coefficient for age is 0.005 and is significant (t = 6.71 and p = 0.000) this isn't a very good model (bivariate models often are not), as the R^2 value is very low at 0.009 – less than 1%. It may be that the association between the dependent variable and the independent variable is not linear. Previous research informs us that age often has a curvilinear relationship with income in that income initially increases with age and then decreases. We can check if this is the case in these data with a scatterplot with *age* as the X variable and *ln_inc* as the Y variable.

```
scatter ln_inc age
```



Capturing this type of non-linear association often requires the addition of a squared term of the independent variable.

```
gen agesq=age*age
```

or

```
gen agesq=age^2
```

If we rerun our regression with the transformed income variable and the *age* variable plus its square we see that our model fits much better; R^2 is now 0.067 (6.7%). The significance of the age terms (*age* and *agesq*) tells us that we were correct to assume a curvilinear relationship.

```
regress ln_inc age agesq
```

```
. reg ln_inc age agesq

  Source |       SS       df       MS              Number of obs =    4973
---------+------------------------------           F(  2, 4970)  =  180.02
   Model |  159.233417     2  79.6167087            Prob > F      =  0.0000
Residual |  2198.03047  4970  .442259652            R-squared     =  0.0676
---------+------------------------------           Adj R-squared =  0.0672
   Total |  2357.26389  4972  .474107781            Root MSE      =  .66503

------------------------------------------------------------------------------
  ln_inc |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
---------+--------------------------------------------------------------------
     age |   .0841962   .0045302    18.59   0.000     .0753149    .0930774
   agesq |  -.0009989   .0000565   -17.67   0.000    -.0011097   -.000888
   _cons |   5.113853   .0850617    60.12   0.000     4.947095    5.280612
------------------------------------------------------------------------------
```

We can now add some additional predictors of income to our model. First we add a variable for gender. The current variable *sex* is coded 1 = male and 2 = female. We can either recode this to a dummy variable (0,1) for either males or females or use the **xi** command (see also Box 8.2). If we prefix the **regress** command with **xi:** and putting an **i.** in front of our categorical variables of interest, Stata automatically converts them to dummy variables in our regression equation. **xi** expands terms containing categorical variables into indicator (also called dummy) variable sets by creating new variables and then executes the specified command with the expanded terms.

xi:regress ln_inc age agesq i.sex

```
. xi:reg ln_inc age agesq i.sex
i.sex      _Isex_1-2       (naturally coded; _Isex_1 omitted)

  Source |       SS       df       MS              Number of obs =    4973
---------+------------------------------           F(  3,  4969) =  641.84
   Model |  658.344919     3  219.448306           Prob > F      =  0.0000
Residual |  1698.91897  4969  .341903596           R-squared     =  0.2793
---------+------------------------------           Adj R-squared =  0.2788
   Total |  2357.26389  4972  .474107781           Root MSE      =  .58473

------------------------------------------------------------------------------
  ln_inc |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
---------+--------------------------------------------------------------------
     age |   .0905245   .0039866    22.71   0.000     .0827089     .09834
   agesq |  -.0010799   .0000497   -21.71   0.000    -.0011774    -.0009824
  _Isex_2 |  -.6342055    .016599   -38.21   0.000    -.6667469    -.601664
   _cons |   5.322339   .0749894    70.97   0.000     5.175327    5.469352
------------------------------------------------------------------------------
```

Note that at the bottom of the variable list a new variable (*_Isex_2*) has appeared. This is the dummy variable automatically created by Stata for the original variable *sex*. The output line immediately after the command shows how Stata has created dummy or indicator variables out of the *sex* variable:

```
i.sex  _Isex_1-2  (naturally coded; _Isex_1 omitted)
```

This line shows at the left that the variable *sex* was indicated with an **i.** prefix in the command line. The next part (_Isex_1-2) shows that indicator variables have been created (_I) and that the sex variable has categories valued from 1 to 2 (_1-2). It then tells you on the right that the category with the value 1 is the omitted (or reference) category. By default, the dummy-variable set is identified by dropping the dummy corresponding to the smallest value of the variable.

So in this case the indicator variable created by Stata is for females (as females are *sex*=2) compared to males. The negative coefficient for the variable *_Isex_2* shows the mean difference for women compared to men in logged income. In other words, women on average earn less than men after controlling (adjusting) for age. We can also see from the output that the R^2 value has increased to 0.279 (27.9%) from 6.7% in the model with just *age* and *agesq* as independent variables. This indicates that age and sex explain nearly 28% of the variation in logged income for those in employment.

Next we enter marital status (*mastat*) as an independent variable, also using the **i.** prefix in the following command:

```
xi:reg ln_inc age agesq i.sex i.mastat
```

```
. xi:reg ln_inc age agesq i.sex i.mastat
i.sex          _Isex_1-2          (naturally coded; _Isex_1 omitted)
i.mastat       _Imastat_1-6       (naturally coded; _Imastat_1 omitted)

      Source |       SS       df       MS              Number of obs =    4973
-------------+------------------------------           F(  8,  4964) =  255.16
       Model |  686.879702     8  85.8599628            Prob > F      =  0.0000
    Residual |  1670.38419  4964  .336499634            R-squared     =  0.2914
-------------+------------------------------           Adj R-squared =  0.2902
       Total |  2357.26389  4972  .474107781            Root MSE      =  .58009

------------------------------------------------------------------------------
      ln_inc |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         age |   .0939282   .0045723    20.54   0.000     .0849645    .1028919
       agesq |  -.0011177   .0000548   -20.40   0.000    -.0012251   -.0010103
     _Isex_2 |  -.6489609   .0166143   -39.06   0.000    -.6815323   -.6163895
  _Imastat_2 |   .2093674   .0314691     6.65   0.000      .147674    .2710608
  _Imastat_3 |    .310369   .0707913     4.38   0.000     .1715867    .4491513
  _Imastat_4 |   .1887169   .0411138     4.59   0.000     .1081156    .2693181
  _Imastat_5 |   .1691297   .0628323     2.69   0.007     .0459506    .2923088
  _Imastat_6 |   .0238757   .0263734     0.91   0.365    -.0278279    .0755792
       _cons |   5.222049   .0934908    55.86   0.000     5.038766    5.405332
------------------------------------------------------------------------------
```

In this example, the reference category for marital status (*mastat*) is 'married' as that category has the lowest value (1). Remember that the coefficients for the other dummy variables are all compared to the 'married' category. So, for example, category 2 'living as a couple' has a significant coefficient of 0.209 which indicates that those living as a couple, on average, earn more than those who are married, after controlling for age and sex.

If you want your reference categories to be something else, you can change them with the **char** command (short for 'characteristics'). If we wanted to make 'never married' the reference category, then we would use:

```
char mastat[omit] 6
```

as the 'never married' category has a value of 6. Now we can rerun the regression command. To restore to the default reference categories type use:

```
char mastat[omit]
```

Box 8.2: Using the `xi` command for interactions

The **`xi`** command also allows us to do interactions easily. The **`i.var`** syntax is interpreted as follows:

- **`i.var1`** creates dummies for categorical variable *var1*.
- **`i.var1*i.var2`** creates dummies for categorical variables *var1* and *var2*: main effects and interactions.
- **`i.var1*var3`** creates dummies for categorical variable *var1* and includes continuous variable *var3*: all interactions and main effects.
- **`i.var1|var3`** creates dummies for categorical variable *var1* and includes continuous variable *var3*: all interactions and main effect of *var3*, but not main effect of *var1*.

We can also use the **`if`** and **`bysort`** commands with **`regress`**. For example, if you were interested in running different regression models for each sex then you could use the **`if`** command:

```
xi:reg ln_inc age agesq i.mastat if sex==1
xi:reg ln_inc age agesq i.mastat if sex==2
```

If you put one or more instances of **`i.`** in your command you must put the **`xi:`** first then the **`bysort`** command:

```
xi: bysort sex: reg ln_inc age agesq i.mastat
```

There are some commands that cannot be combined with **`by`** and/or **`bysort`**. If you try to combine them, Stata will give you an error message to this effect.

You could include the indicator or dummy variables by making them using the **`tab`** command with the **`gen`** option. For example, using *mastat*=1 as the reference category:

```
tab mastat, gen(mstat)
bysort sex: reg ln_inc age agesq mstat2-mstat6
```

Another slight tweak to the process would be to generate the dummy variables using the **`tab`** command but then to drop the reference category variable and use an * for all the dummy variables.

Putting an asterisk after the common part of the variable name tells Stata to include all variables that start with that common part; so, **mstat*** will include all variables that start with *mstat*. * is the Stata wildcard notation. So the commands would be:

```
tab mastat, gen(mstat)
drop mstat1
bysort sex: reg ln_inc age agesq mstat*
```

As the **tab** command creates dummy variables for every category of the *mastat* variable, if we did not drop the reference category variable using the * wildcard we would put all dummy variables into the regression. Stata will produce results but it will decide which one of the dummy variables to drop and you lose control over the reference category.

Two of the common options for use with **regress** are:

- **beta**, which requests that normalized beta coefficients be reported instead of confidence intervals;
- **level(#)**, which specifies the confidence level, as a percentage, for confidence intervals of the coefficients.

Regression diagnostics

Stata comes with a series of graphs to help assess whether or not your regression models meet some of the assumptions of linear regression. Using the pull-down menu, these are found at

Graphics → Regression diagnostic plots

Before going on to the diagnostics, we will briefly discuss regression assumptions. Fuller discussions are available in most statistical text books, but we suggest reading Berk (2003) for a general critique of the regression method and its common abuses, while Belsley et al. (2004), Fox (1991) and Pedhazur (1997) are good texts for the assumptions and diagnostics (see also Box 8.3).

The main assumptions of OLS regression are as follows:

1. The independent variables are measured without error.
2. The model is properly specified so that it includes all relevant variables and excludes irrelevant variables.
3. The associations between the independent variables and the dependent variable are linear.

Box 8.3: Errors and ERRORS

One of the things that stuck in our minds as students was a short section in Pedhazur (1997: 9) titled 'There are errors and there are ERRORS', in which he encourages researchers to find the balance between failing to meet the assumptions of statistical techniques (or not caring if they are met or not) and the debilitating quest for statistical perfection in real-world research and data.

Some of the assumptions are testable; others are not and have to be justified by logic and argument. Therefore, no matter how many statistical/diagnostic tests you run there will still be a possibility that you have violated one of the many assumptions. So, to avoid the paralysis of perfection we encourage you to adopt Pedhazur's approach and balance your investigations with some pragmatism: is it an error or an ERROR?

> ... understanding when violations of assumptions lead to serious biases, and when they are of little consequence, are essential to meaningful data analysis.
>
> (Pedhazur 1997: 33).

4. The errors are normally distributed. Errors are the difference between predicted and actual values for each case. Predicted values are also called fitted values. Errors are also called residuals or disturbances.
5. The variance of the errors is constant; usually referred to as homoscedasticity. If the errors do not have constant variance they are heteroscedastic.
6. The errors of one observation are not correlated with the errors of any other observation.
7. The errors are not correlated with any of the independent variables.

Then there are a number of what we call 'technical' issues that you need to check:

8. Strange cases or outliers: these may be from coding errors or may be truly different in which case you may need to examine them further in detail.
9. Leverage and influence: to determine if any of the cases have undue leverage or power on the regression line.

10. Multicollinearity: if the independent variables are highly correlated with one another this may affect the regression estimates.

The first assumption is extremely difficult to meet, if not impossible in social research. Measurement error in the independent variables usually results in underestimating the effects, and the extent of the underestimation has been shown to be linked to the reliability of the measure (Pedhazur 1997). We are guilty of violating this assumption ourselves in the examples in this book. Is the GHQ a completely valid and reliable measure of mental well-being? Not at all, but our models do not take that into account. If you are interested in combining measurement and effect models we suggest you delve into structural equation modelling. It's worth noting that measurement error in the dependent variable does not bias the estimates but does inflate their standard errors, which then gives a higher *p* value and so a weakened test of significance.

The second assumption, model specification, has to be addressed theoretically, practically, as well as statistically. In developing models to test, the theory needs to be complete, and testable, for the model to be correctly specified. Practical issues such as data availability may also hinder you in specifying a correct model. There are commands in Stata that test whether you have omitted relevant variables. They don't tell you what they are! Nor do they tell you if you have included irrelevant variables. We cover the **linktest** and **ovtest** commands as we go through our example.

The third assumption of linearity is a variation on the second assumption, and we have already discussed ways of dealing with non-linear associations. There are tests for non-linearity but we suggest that these are largely unnecessary if you conduct in-depth univariate and bivariate data analysis before moving on to multivariate analysis.

The distribution of the errors/residuals can be easily attended to after a regression command and the distribution can be visually inspected in graphs and then formally tested using the normality tests covered in Chapter 5. We look at the **rdplot** and **qnorm** graphs as well as summary statistics commands such as **su** and **tabstat** combined with appropriate normality tests in our example.

To see if the variance of the errors is homoscedastic we can plot the errors (residuals) against the predicted (fitted) values in a scatterplot. In such a plot we are looking for no discernable

pattern and that the residuals are in an even band across all of the predicted values. This is created by the **rvfplot** command. We can formally test this using the **hettest** command.

The sixth assumption of non-correlated errors is difficult to assess, and with most non-experimental data it is probably safer to assume that these exist, rather than that they don't! Cluster sampling strategies will almost certainly mean this assumption is violated. Again, we have fallen foul of this assumption in our examples as we are using household data and the people who share the same household are probably more alike than those who don't. The effect is to underestimate the standard errors of the coefficients of the independent variables, possibly giving coefficients statistical significance when they shouldn't. A common solution when using cross-sectional data is to use robust standard errors. This can be done in our regression by either using **vce(robust)** as an option or, better, as we know that individuals are clustered in households in our data, the **cluster(hid)** option:

```
xi:reg ln_inc age agesq i.sex i.mastat, ///
   cluster(hid)
```

See what happens to the standard errors, *t* values and *p* values compared to the original, partial, output shown below. The coefficients have remained the same but the standard errors have increased resulting in lower *t* values:

```
------------------------------------------------------------------------------
       age |   .0939282   .0050763    18.50   0.000     .0839764    .10388
     agesq |  -.0011177   .0000608   -18.39   0.000    -.0012369   -.0009985
  _Isex_2 |  -.6489609   .0166627   -38.95   0.000    -.6816271   -.6162947
------------------------------------------------------------------------------
       age |   .0939282   .0045723    20.54   0.000     .0849645    .1028919
     agesq |  -.0011177   .0000548   -20.40   0.000    -.0012251   -.0010103
  _Isex_2 |  -.6489609   .0166143   -39.06   0.000    -.6815323   -.6163895
------------------------------------------------------------------------------
```

The last assumption is linked to model specification, especially in non-experimental data. It follows that if there is an omitted variable that is also correlated with one of the independent variables then, as the effect of that omitted variable is in the error term, then the errors will be correlated with the independent variable. For example, suppose we were investigating children's educational attainment with a model that had parents' education, social class, residence area and number of siblings as independent variables. Parents' income is not available and so is not in the

model. However, we know that parents' education and income are likely to be correlated. Therefore, the error term, which includes the effect of parents' income, will be correlated with the included independent variable parents' education.

The three technical issues are discussed more as we work through our example.

We suggest that you adopt a systematic approach to regression diagnostics, and as the diagnostic commands to be used after every regression are generic you could easily copy and paste a set of diagnostic commands into a do file after each regression. This way you know that you haven't missed anything. Such an annotated do file is shown in Box 8.4.

Box 8.4: Diagnostic commands

This summary of Stata commands and the assumption or technical issue they help check for is adapted from Chen et al. (2003).

Model specification

linktest	performs a link test for model specification.
ovtest	performs regression specification error test for omitted variables.

Normality of errors

rdplot	graphs a histogram of the residuals. Use **findit rdplot** to install.
pnorm	graphs a standardized normal probability plot.
swilk	performs the Shapiro–Wilk *W*-test for normality.

Homoscedasticity

rvfplot	graphs residual-versus-fitted plot.
hettest	performs Cook and Weisberg test for heteroscedasticity.

Leverage and influence

predict	create predicted values, residuals, and measures of influence.
rvfplot	graphs residual-versus-fitted plot.
lvr2plot	graphs a leverage-versus-squared-residual plot.
dfbeta	calculates DFBETAs for all the independent variables.

Multicollinearity

vif calculates the variance inflation factor for the independent variables.

An example do file for regression diagnostics is shown below. There are other tests and graphs that you may wish to add later.

```
** Model Specification **
linktest  /* performs a link test
          for model specification
          Look for _hat being sig p<.05
          and _hatsq being not sig p>.05
          _hatsq not sig means no
          omitted vars if _hatsq sig
          then omitted vars */

ovtest    /* performs regression
          specification error test
          for omitted variables. Look
          for p>.05 so not to reject
          hypothesis: model has no
          omitted vars*/

** Normality of errors **
predict res,res       /* use predict to
                      create new var res
                      (residuals) */
predict stres, rsta   /* use predict to
                      create standardized
                      res */

rdplot  /* graphs a histogram of the
        residuals
        Look for a normal distribution
        with no outliers */
** save graph?

** if you haven't installed rdplot then use:
histogram res

pnorm res  /* graphs a standardized normal
           probability (P-P) plot of res
```

```
            Look for plot to be close to
            diagonal */
** save graph?

su res stres  /* summary statistics for
              res
              Look for mean=0 and no min
              and max values >abs 2.5 */
swilk res     /* performs the Shapiro-Wilk
              W test for normality on
              res testing hypothesis of
              normality so p<.05
              rejects */

sktest res    /* for larger samples
              testing hypothesis of
              normality so p<.05
              rejects */

tabstat res, s(sk kur)  /* to actually see
                        the skew and kurt
                        stats remember no
                        skew = 0, no
                        kurt = 3 */

** Homoscedasticity **
rvfplot  /* graphs residual-versus-fitted
         plot
         Look for even distribution "no
         pattern" and possible cases of
         high influence */
** save graph?

hettest  /* performs Cook and Weisberg
         test for heteroscedasticity
         testing hypothesis of constant
         variance so p<.05 rejects */

** Leverage and influence **
predict lev, leverage  /* create leverage
                       values critical
                       value 2(k+1)/N */
```

```
predict cooks, cooksd  /* create Cook's
                       D stats critical
                       value 4/N */

dfbeta                 /* calculates
                       DFBETAs for all
                       the independent
                       variables critical
                       value 2/sqroot N */

su lev cooks DF*       /* summary stats
                       for inspection and
                       checking against
                       critical values */

lvr2plot  /* graphs a leverage-versus-
          squared-residual plot
          Look for cases with large
          leverage values */

** Multicollinearity **
vif  /* calculates the variance inflation
     factor for ind vars
     Look for VIF > 10 or 1/VIF
     (tolerance) < 0.1 */

drop res stres lev cooks DF*  /*otherwise
                              error
                              after next
                              regession! */
```

A feature of the estimation procedures in Stata is the post-estimation commands; type **help postest** for an introduction. However, there is usually more specific information about post-estimation commands in the sections on the estimation commands, such as **regress**, themselves. Many of the post-estimation commands we cover here are straightforward to apply after a **regress** command, but the results can be obtained in other ways (no surprise there, then!) and most is done through the **predict** command and its options.

We now follow on from our regression example where we had income with a logarithmic transformation as the dependent variable and then age, age squared, sex, and marital status as

independent variables in a sample of employed people. While these independent variables explain almost 30% of the variance in logged income, we are not expecting this to be a satisfactory model as we all could think of a number of other factors that would have an effect on income. However, let's proceed with the diagnostics for that model using the do file commands in Box 8.4.

First, we use the two tests of model specification: **linktest** and **ovtest**. These tests use a similar process whereby new variables are created and then tested in the model. The **linktest** results are more transparent as they are displayed as a usual regression output, whereas the **ovtest** produces just a single test statistic and its *p* value. We have annotated the do file to indicate what to look for in these test results so you can see that both indicate that we have omitted variables, which is not a surprise.

```
. ** Model Specification **
. linktest  /* performs a link test for model specification
>           Look for _hat being sig p<.05 and _hatsq being not
>           sig p>.05
>           _hatsq not sig means no omitted vars
>           if _hatsq sig then omitted vars */

  Source |       SS       df       MS      Number of obs =    4973
---------+------------------------------   F( 2, 4970)    = 1032.16
   Model | 691.771172     2 345.885586   Prob > F        =  0.0000
Residual | 1665.49272  4970 .335109198   R-squared       =  0.2935
---------+------------------------------   Adj R-squared =  0.2932
   Total | 2357.26389  4972 .474107781   Root MSE        =  .57889

------------------------------------------------------------------------------
  ln_inc |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
---------+--------------------------------------------------------------------
    _hat |  -2.007193   .7874233    -2.55   0.011    -3.550891   -.4634963
  _hatsq |   .2246087   .0587899     3.82   0.000     .1093546    .3398627
   _cons |   10.03438   2.630634     3.81   0.000     4.877174    15.19158
------------------------------------------------------------------------------

.
. ovtest  /* performs regression specification error test for
>                       omitted variables
>          Look for p>.05 so not to reject hypothesis: model
>                      has no omitted vars*/
Ramsey RESET test using powers of the fitted values of ln_inc
       Ho: model has no omitted variables
                F(3, 4961) = 44.95
                Prob > F   = 0.0000
```

In this next step we use the **predict** command to create two new variables: one for the errors or residuals and one for the standardized residuals. We examine the distribution of the errors

Box 8.5: Alternative diagnostic commands

The **linktest** command can be replicated by:

```
predict yhat, xb
gen yhatsq=yhat^2
reg ln_inc yhat yhatsq
```

The **xb** option to **predict** creates a new variable of the predicted (fitted) values of Y for each case.

The **rdplot** can also be produced by:

```
predict res, r
histogram res
```

The **r** option to **predict** creates a new variable of the residuals (error) for each case. You may also want to examine standardized residuals, in which case use the **rsta** option and graph these.

```
predict stres, rsta
histogram stres
```

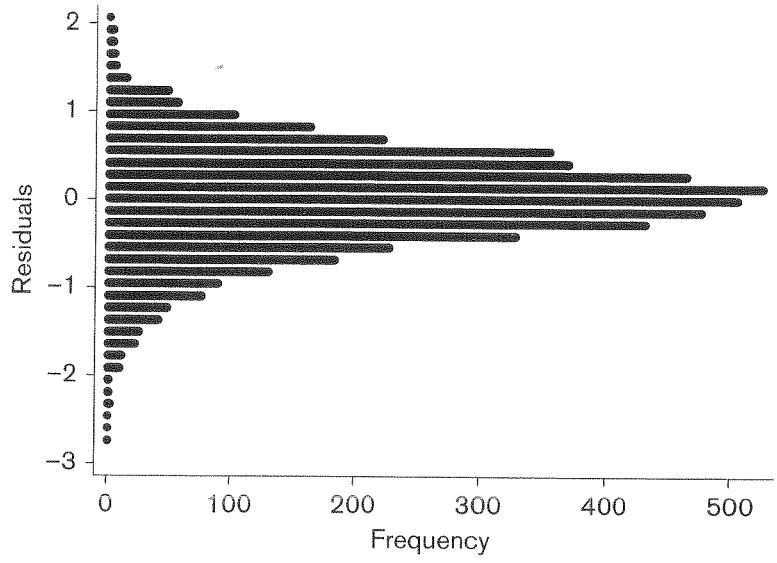
The **rvfplot** can also be produced (assuming you have already created the *yhat* and *res* variables) by:

```
scatter res yhat
```

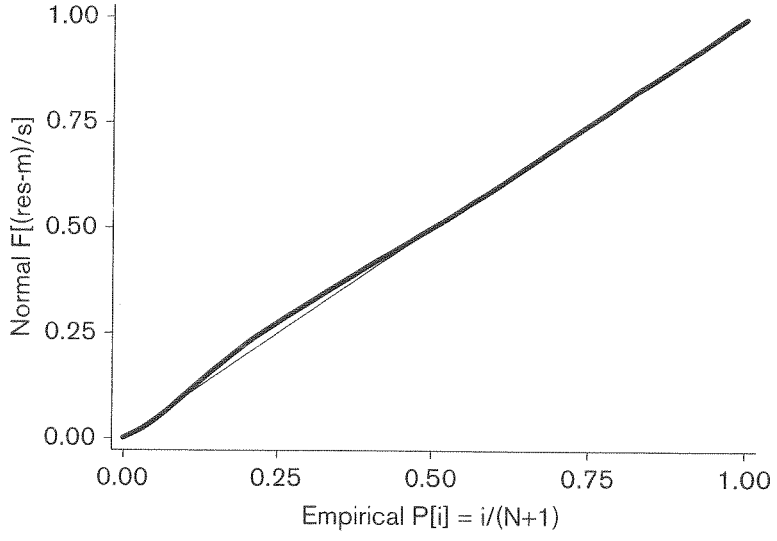
or residuals by first visually inspecting two graphs. The **rdplot** command needs to be installed, so type **findit rdplot** and follow the instructions (see also Box 8.5). If you haven't done this, you can still use **histogram res** to get a similar graph. The histogram shows that there is a longer negative tail on the distribution, indicating that it is probably negatively skewed. In the **pnorm** graph there is also a departure from the diagonal. You may also want to add a **qnorm** plot here.

```
. predict res,res       /* use predict to create
                        new var res (residuals) */
. predict stres, rsta   /* use predict to create
                        standardized res */
```

```
. ** Normality of errors **
. rdplot  /* graphs a histogram of the residuals
>         Look for a normal distribution
>         with no outliers */
```




```
. pnorm res  /* graphs a standardized normal
>            probability (P-P) plot of res
>            Look for plot to be close to
>            diagonal */
```



Next, we inspect the summary statistics of the two new variables of the residuals and the standardized residuals using the **su** command. We can see that there are cases with standardized residuals considerably larger than 3, or even 3.5. This indicates that we probably have outliers and that the residuals may not be normally distributed.

```
. su res stres  /* summary statistics for res
                Look for mean=0 and no min and max
                values >abs 2.5 */

. su res stres
```

Variable	Obs	Mean	Std. Dev.	Min	Max
res	4973	-1.47e-10	.579619	-2.696645	2.102549
stres	4973	-.0000186	1.000119	-4.660335	3.403345

We formally test the distribution of the errors using the normality tests shown in Chapter 5. These also confirm that the distribution departs from normality in both skewness and kurtosis.

```
. swilk res  /* performs the Shapiro-Wilk W test for
>            normality on res testing hypothesis of
>            normality so p<.05 rejects */
```

Shapiro-Wilk W test for normal data

Variable	Obs	W	V	z	Prob>z
res	4973	0.98902	29.619	8.889	0.00000

```
.
. sktest res /* for larger samples
>               testing hypothesis of normality so p<.05
                rejects */
```

Skewness/Kurtosis tests for Normality

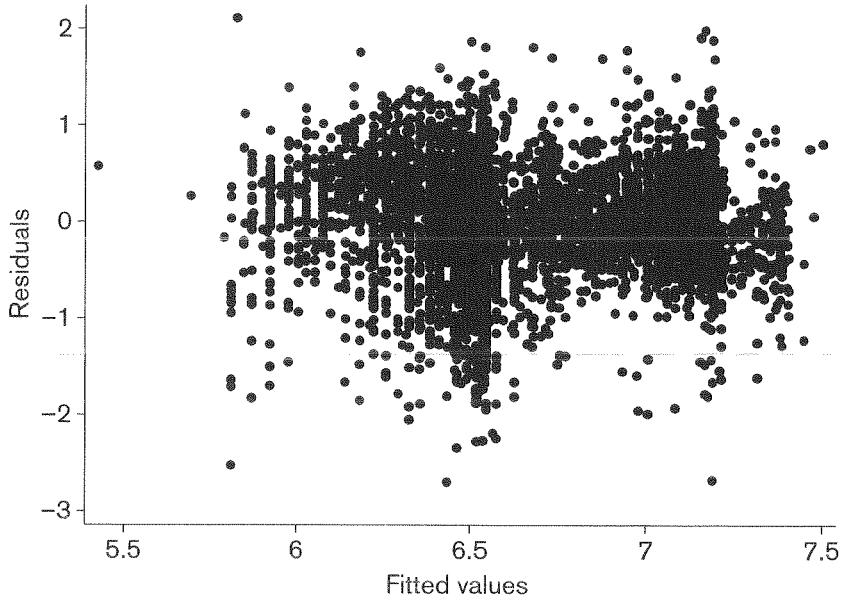
			joint	
Variable	Pr(Skewness)	Pr(Kurtosis)	adj chi2(2)	Prob>chi2
res	0.000	0.000	.	0.0000

```
.
. tabstat res, s(sk kur) /* to actually see the skew and
>       kurt stats remember no skew = 0, no kurt = 3 */
```

variable	skewness	kurtosis
res	-.4042151	3.79337

The visual inspection of the graph of fitted values (predicted values) against residuals (errors) clearly shows that the variance of the errors is not constant across the range of fitted values. Therefore, we have violated the assumption of homoscedasticity. This is confirmed by the statistical test which rejects the hypothesis of constant variance.

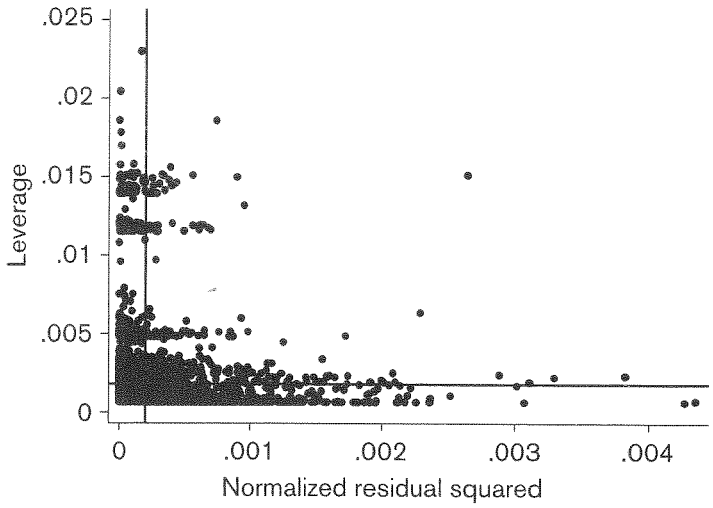
```
. ** Homoscedasticity **
. rvfplot  /* graphs residual-versus-fitted plot
>          Look for even distribution "no
>          pattern" and possible cases of high
>          influence */
```




```
. hettest  /* performs Cook and Weisberg test
>          for heteroscedasticity testing
>          hypothesis of constant variance so
>          p<.05 rejects */

Breusch-Pagan / Cook-Weisberg test for
heteroskedasticity
         Ho: Constant variance
         Variables: fitted values of ln_inc

         chi2(1)      = 190.58
         Prob > chi2  = 0.0000
```

For this next part of the diagnostics we create the leverage and influence values for each case in new variables. The leverage and Cook's *D* values are created using the **predict** command, and the **dfbeta** command automatically produces a DFBETA value for all the independent variables. We then use the **su** command to make a table so that we can see if any of the values are greater than the critical values. This isn't the place to engage in a debate on the use of critical values or cut-offs, but just to say that these are *rules of thumb* rather than commandments set in stone. One point to think about is the effect of having *N* in the denominator in these calculations when using samples in the many thousands. In our current model we have eight independent variables so $k = 8$, and an estimation sample of 4973 so $N = 4973$. Accordingly, the critical values are: leverage, 0.00362; Cook's *D*, 0.0008; and DFBETA, 0.02836. All of the values indicate that there are cases that have undue leverage and/or influence in this model. The leverage–residual plot clearly shows that there are quite a few cases with high leverage values.

```
. ** Leverage and influence **
. predict lev, leverage  /* create leverage values
>                                critical value 2(k+1)/N */
. predict cooks, cooksd  /* create Cook's D stats
>                                critical value 4/N */
. dfbeta                 /* calculates DFBETAs for all the
>                                independent variables
>                                critical value 2/sqroot N */
                    DFage:  DFbeta(age)
                  DFagesq:  DFbeta(agesq)
                DF_Isex_2:  DFbeta(_Isex_2)
              DF_Imastat_2:  DFbeta(_Imastat_2)
              DF_Imastat_3:  DFbeta(_Imastat_3)
              DF_Imastat_4:  DFbeta(_Imastat_4)
              DF_Imastat_5:  DFbeta(_Imastat_5)
              DF_Imastat_6:  DFbeta(_Imastat_6)

. su lev cooks DF*  /* summary stats for inspection and
>                    checking against critical values */

    Variable |  Obs       Mean Std. Dev.       Min       Max
-------------+--------------------------------------------------
         lev |  4973   .0018098  .0024166   .0006241 .0229986
       cooks |  4973   .0001882  .0005523   1.84e-14 .0227647
DF_Imastat_2 |  4973 -2.97e-07   .0128734  -.1459199 .1329711
DF_Imastat_3 |  4973 -3.84e-07   .0125399  -.2566079 .1986158
DF_Imastat_4 |  4973 -3.33e-07   .0129498  -.2335146 .1920423
-------------+--------------------------------------------------
DF_Imastat_5 |  4973  2.39e-08   .0123927   -.197925 .2329337
DF_Imastat_6 |  4973 -1.03e-06   .0135859  -.1608311 .1074181
       DFage |  4973 -2.23e-06   .0158038  -.3712751 .1422175
     DFagesq |  4973  2.53e-06   .0158053  -.1336512 .4085968
   DF_Isex_2 |  4973  2.31e-06   .0142373  -.0704291 .0673829

. lvr2plot  /* graphs a leverage-versus-squared-residual plot
>           Look for cases with large leverage values */
```



Finally, we examine whether any of the independent variables are collinear as a check for multicollinearity. As a precursor to regression you should be looking at the bivariate associations between potential independent variables which would give an early warning about issues of multicollinearity. The **vif** command produces the variance inflation factor and the tolerance, which is simply the reciprocal of the variance inflation factor and preferred by some users. Our results show that there is collinearity between age and age squared, but is to be expected as they have an almost perfect linear correlation! All of the other independent variables have variance inflation factors less than 10 or tolerances greater than 0.1, which shows that multicollinearity does not exist.

```
. ** Multicollinearity **
. vif  /* calculates the variance inflation factor for ind vars
>      Look for VIF > 10 or 1/VIF (tolerance) < 0.1 */

    Variable |      VIF       1/VIF
-------------+----------------------
         age |    46.83    0.021356
       agesq |    43.17    0.023162
  _Imastat_6 |     1.70    0.586526
  _Imastat_2 |     1.16    0.861387
  _Imastat_3 |     1.06    0.946293
  _Imastat_4 |     1.03    0.972071
     _Isex_2 |     1.02    0.980591
  _Imastat_5 |     1.01    0.986029
-------------+----------------------
    Mean VIF |    12.12

.
. drop res stres lev cooks DF*  /*otherwise error after next
                                regession! */
```

So, what does all this mean for our regression model? In terms of model specification, it is not surprising that these results indicate that we have omitted variables; no one would think that age, sex and marital status alone would satisfactorily explain variations in income. Some of the omitted variables could be education, work experience, and sector of industry, for example. The errors or residuals are well dispersed beyond the normal distribution, with some standardized residuals beyond 3.5. The error terms are also heteroscedastic, which is more than likely linked with the poor model specification. A good number of cases have large leverage and/or influence which could be linked to the outliers seen in the residuals, but not necessarily so. However, we are confident that we do not have multicollinearity, which is at least one thing going for this model at this stage. Clearly, quite a lot more work needs to be done before we obtain a more satisfactory model.

LOGISTIC REGRESSION

Logistic regression (also called logit or, to distinguish it from other types of categorical dependent variables, binary logit or binary logistic regression) is used for regression with a dichotomous dependent variable. Stata's **logit** command has the same general format as **regress**. The dependent variable should be a 0/1 dichotomy; for analytic purposes a 0 is referred to as a failure and 1 as a success, regardless of the substantive meaning of the variables. For more discussion on the details and application of logistic regression, see Long (1997), Long and Freese (2006), or Menard (2002).

Many users prefer the **logistic** command to **logit**. Results are the same regardless of which you use, but the **logistic** command reports odds ratios (Box 8.6) rather than logit coefficients by default.

In this example, we will look at the outcome of whether or not a person has a first degree or higher, derived from the variable *educ* (for the variable *educ*, higher degree = 1, first degree = 2). Therefore, to construct the dichotomous or binary variable:

```
recode educ (1/2=1) (3/max=0),gen(degree)
```

We will also use the whole sample of the example data. So, if we were following on from the above example looking at income as

the dependent variable in a sample of those working, we would need to open the data again. And not forgetting to recode the missing values as well!

As the sample contains people aged 16 and older, it is unlikely that the younger people in the sample would have had the opportunity to gain a degree so we'll restrict this analysis to those aged 25 and older by using the command

```
drop if age<25
```

First, we will examine if sex and age are determinants of having a degree:

```
xi:logit degree i.sex age
```

```
. xi:logit degree i.sex age
i.sex        _Isex_1-2        (naturally coded; _Isex_1 omitted)

Iteration 0:  log likelihood = -2301.563
Iteration 1:  log likelihood = -2189.489
Iteration 2:  log likelihood = -2180.6155
Iteration 3:  log likelihood = -2180.5053
Iteration 4:  log likelihood = -2180.5053

Logistic regression                               Number of obs =    8390
                                                  LR chi2(2)    =  242.12
                                                  Prob > chi2   =  0.0000
Log likelihood = -2180.5053                       Pseudo R2     =  0.0526

------------------------------------------------------------------------------
      degree |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     _Isex_2 |  -.4637623   .0830406    -5.58   0.000    -.6265189   -.3010057
         age |  -.0406401   .0030834   -13.18   0.000    -.0466835   -.0345967
       _cons |   -.423095   .1381117    -3.06   0.002     -.693789    -.152401
------------------------------------------------------------------------------
```

Compare the output from the **logit** command above with the output from the **logistic** command below. You can see that odds ratios are presented instead of coefficients but the z and p values are identical, as are the model fit statistics reported in the top right-hand panel.

```
xi:logistic degree i.sex age
```

```
. xi:logistic degree i.sex age
i.sex             _Isex_1-2           (naturally coded; _Isex_1 omitted)

Logistic regression                               Number of obs   =      8390
                                                  LR chi2(2)      =    242.12
                                                  Prob > chi2     =    0.0000
Log likelihood = -2180.5053                       Pseudo R2       =    0.0526

------------------------------------------------------------------------------
      degree | Odds Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      _Isex_2 |    .628913   .0522253    -5.58   0.000     .5344491    .7400735
         age |   .9601746   .0029606   -13.18   0.000     .9543894    .9659949
------------------------------------------------------------------------------
```

Using the **or** option with the **logit** command will give you the same results as using the **logistic** command, such as:

```
logit degree i.sex age, or
```

Box 8.6: Odds ratios

Odds ratios are sometimes preferred over logit coefficients for their ease of interpretation. The logit coefficients report the effect of the independent variable on the logarithm of the odds of being in the 1 category of the dependent variable compared to being in the 0 category. Or, in our example, of having a degree compared to not having one. So, the logit coefficient for *sex* in our example is –0.464 which we would interpret as saying that women have, on average, 0.464 less of the logarithm of the odds of having a degree.

Odds ratios are the exponential of the logit coefficient. Here we simply take the exponential of the logit coefficient using the calculator in Stata:

```
display exp(-.464)
```

```
. display exp(-.464)
.62876355
```

So, the odds ratio for the variable *sex* is 0.63. Odds ratios range from 0 to +∞, with the value for no effect being equal to 1. This means that odds ratios lower than 1 are 'negative' effects and odds ratios greater than 1 are 'positive' effects. The odds ratio of 0.63 in our example is interpreted as 37% less likely to have a degree. An odds ratio of 1.43 would be interpreted as 43% more likely, and so on. As you may gather, the range above 1 is greater than that below 1, which is bounded by zero, so comparing the magnitude of effects either side of 1 is not straightforward and needs care.

Even though these results indicate that women are less likely to have a degree than men (odds ratio 0.63) and that as age increases the likelihood is reduced (odds ratio 0.96), it is unlikely that the gender difference is constant with all values of age, as we know that in the past much fewer women went to university. It is therefore possible that there is an interaction between age and sex in that the effect of age varies across sexes. For more details on interaction effects in logistic regression, see Jaccard (2001). We can test this by including an interaction term as described in Box 8.2:

```
xi: logistic degree i.sex*age
```

```
. xi:logistic degree i.sex*age
i.sex             _Isex_1-2         (naturally coded; _Isex_1 omitted)
i.sex*age         _IsexXage_#       (coded as above)

Logistic regression                               Number of obs   =       8390
                                                  LR chi2(3)      =     250.61
                                                  Prob > chi2     =     0.0000
Log likelihood = -2176.2561                       Pseudo R2       =     0.0544

------------------------------------------------------------------------------
      degree | Odds Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     _Isex_2 |   1.324136   .356577     1.04   0.297     .7811104    2.244672
         age |   .9677965   .0038777   -8.17   0.000     .9602261    .9754266
_IsexXage_2 |   .9819062   .0062061   -2.89   0.004     .9698176    .9941455
------------------------------------------------------------------------------
```

The results indicate that the interaction term (`_IsexXage_2`) has a significant coefficient which tells us that the effect of age varies across sexes or, conversely, the effect of sex varies with age.

To get a clearer picture of what this means it's a good idea to graph interaction effects. We can easily do this by using the **predict** post-estimation command to calculate predicted, or fitted, values.

```
predict yhat,xb
```

Then use the pull-down menu:

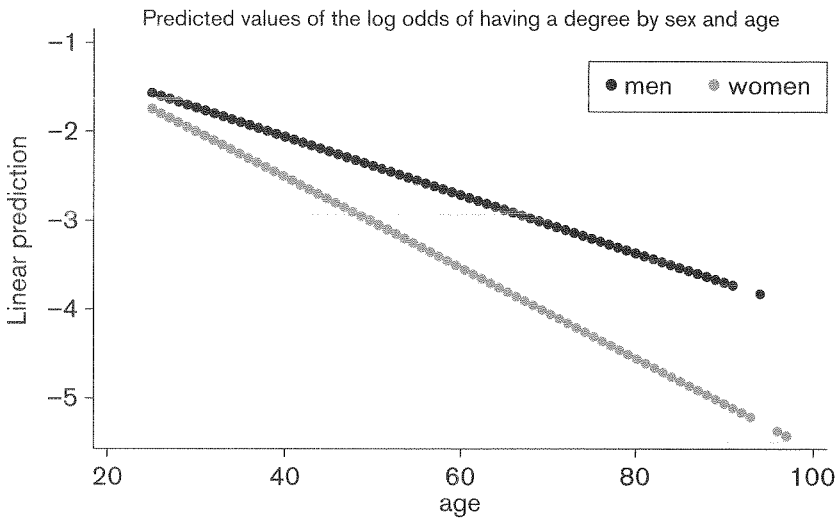
Graphics → Twoway graph

Then create two plots in a similar way to that described in Box 6.7 but with the following entries:

Plot1: X axis = *age*, **Y axis** = *yhat*, **if/in** tab – sex==1
Plot 2: X axis = *age*, **Y axis** = *yhat*, **if/in** tab – sex==2
Legend tab – select **Override default keys** and type 1 'men'
2 'women' in the box.

Add titles as you wish.

The graph shows that with increasing age both men and women are less likely to have a degree than younger people. But when looking at the effect of age on each sex, you can see that at age 25 there is little difference in the likelihood of having a degree but that the gap increases as age increases. Therefore, the gender gap increases as age increases, which makes substantive sense from what we know about recent history of university admissions and accessibility. It is worth noting that the Y-axis units are in logit (log of the odds). Compare this with the results shown in Box 8.7.



OTHER REGRESSION COMMANDS

Basic regression commands in Stata generally have the same structure in that the command is followed by the dependent variable and then a list of independent variables. There are many other regression models; if you wish to extend your knowledge of regression models with categorical or count dependent variables, then we recommend you use Long (1997) or Long and Freese (2006).

Box 8.7: Post-estimation commands

There are a series of very useful post-estimation commands available to download if your copy of Stata is web enabled. These are based on the **spostado** commands developed by Long and Freese (2006). You must first install the **spostado** files which can be found by typing **findit spostado**. The new box will show a link to:

spost9_ado from http://www.indiana.edu/~jslsoc/stata

Click on this link which will take you to another new box and then simply click where it says **click here to install**.

Follow the same three steps to install the **postgr3** and **xi3** commands developed by Michael Mitchell and Phil Ender at UCLA: Academic Technology Services, Statistical Consulting Group. See www.ats.ucla.edu/stat/stata/ado/analysis/.

Now rerun the logistic regression using the **xi3** prefix:

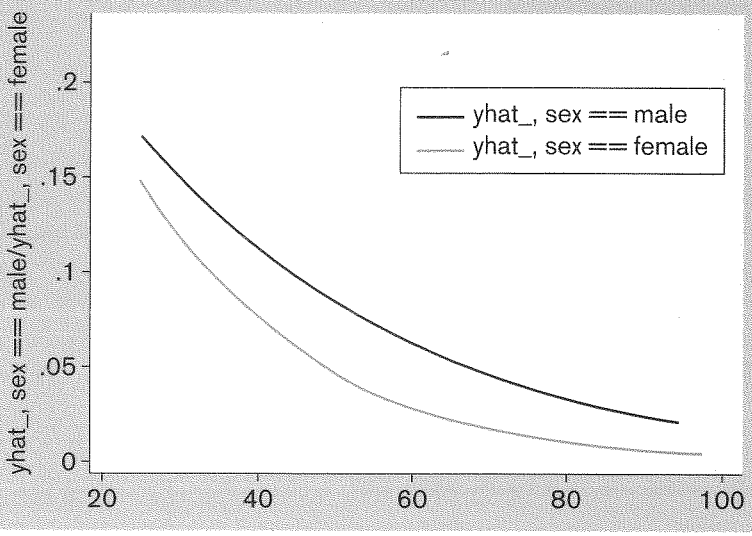
xi3: logistic degree i.sex*age

```
. xi3:logistic degree i.sex*age
i.sex     _Isex_1-2     (naturally coded; _Isex_1 omitted)

Logistic regression                               Number of obs =    8390
                                                  LR chi2(3)    =  250.61
                                                  Prob > chi2   =  0.0000
Log likelihood = -2176.2561                       Pseudo R2     =  0.0544

------------------------------------------------------------------------------
      degree | Odds Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      _Isex_2 |   1.324136   .356577     1.04   0.297     .7811104    2.244672
         age |   .9677965   .0038777    -8.17   0.000     .9602261    .9754266
     _Ise2Xag |   .9819062   .0062061    -2.89   0.004     .9698176    .9941455
------------------------------------------------------------------------------
```

Then use the **postgr3** command to produce a graph of the probability of having a degree by age for both men and women. The lines for men and women are produced by using the **by(sex)** option. Try omitting this and see what graph is produced. The graph varies from the one produced by using linear fitted values as this one has probability of having a degree on the Y axis. In some ways this is easier to understand than fitted logit values. There are many other ways to use this, and other post-estimation graphing

commands. For more information, see Long and Freese (2006) and the UCLA website.

```
. postgr3 age,by(sex)
Variables left asis: age _Isex_2 _Ise2Xag
```



Here in brief are some of the more common regression commands:

- **mlogit** – multinomial logit regression for nominal dependent variables with three or more categories. Note that there is *not* a mlogistic command. Relative risk ratios are reported if the **rrr** option is used.
- **ologit** – ordered logit regression for an ordinal dependent variable. Again, there is *not* an ologistic command, but if you wish to show odds ratios then use the **or** option.
- **probit** – binary probit regression. Probit is the other main method for analysing binary dependent variables. Whereas logit (or logistic) regression is based on log odds, probit uses the cumulative normal probability distribution.
- **mprobit** – multinomial probit regression. Probit for nominal dependent variables with three or more categories.
- **oprobit** – ordered probit regression. Probit for an ordinal dependent variable.
- **poisson** – Poisson regression for a count (non-negative integers) dependent variable.

- **nbreg** – negative binomial regression for a count variable that is overdispersed. A Poisson distribution is a special case of the negative binomial family, and a dependent variable with a true Poisson distribution can also be estimated using the **nbreg** command.

Box 8.8: Weighting

If your data is simply weighted then Stata can use the weights in a number of ways, depending on how and why the weights were constructed – see **help weight**. Weighting is a complicated (and controversial) issue and it is beyond the scope of this book to go into the whys and wherefores of it. Briefly, you can weight tables and most estimation procedures by adding a weight option to the command line in square brackets. Here we show two examples of using weights in a crosstabulation and in a regression model. The weighting variable is *weight*.

```
ta mastat sex [aw=weight]
xi:reg ghqscale i.sex age i.mastat [pw=weight]
```

Box 8.9: Applications of regression modelling in a research project

In a series of analyses using data from the first wave of the Canadian National Longitudinal Survey of Children and Youth (NLSCY) we were interested in the factors associated with birth outcomes (low birthweight, preterm birth and small for gestational age) and then how birth outcomes affected motor and social development in very young children.

In the first analysis[1] we had three dichotomous birthweight outcomes. Low birthweight (LBW) was defined as those children born weighing less than 2500 g, preterm birth was birth at 258 days' gestation or less, and small for gestational age (SGA) was defined as those under the 10th percentile of the gestational growth curves. These dichotomous outcomes were the dependent variables in a series of logistic regression models with social, environmental and mother's behavioural variables as independent variables. We presented our results as odds ratios as we categorized all of our independent variables and used dummy variables so

they could show either an increased or decreased risk of either LBW or SGA compared to the reference category of the independent variable.

In the second analysis[2] we examined how birthweight, this time classified as LBW (less than 2500 g) and VLBW (less than 1500 g) compared to normal, was associated with motor and social development at ages up to 48 months net of the effects of family and social variables. The motor and social development (MSD) scale used in the analysis was an interval level scale created from a number of items (see our comments in Box 5.1) which was reasonably normally distributed as the scale creation was designed so that the 'average' child for their age scored 100. This enabled us to use a series of OLS regression models to estimate the effects of the independent variables on the MSD scale. We used nested models to test for mediating effects and also tested for interactions (or moderating effects; see this chapter and Chapter 9). We found that there was a significant interaction between mother's education and birthweight which indicated that the low birthweight children with higher educated mothers had 'normal' MSD scores of about 100, while low birthweight children with mothers with lower education had MSD scores less than 90. We presented this interaction as a graph to better convey the moderating effects of birthweight and education on the MSD scale.

[1] Pevalin, D.J., Wade, T.J., Brannigan, A. and Sauve R. (2001) Beyond biology: The social context of prenatal behaviour and birth outcomes. *Social and Preventive Medicine*, 46: 233–239.

[2] Pevalin, D.J., Wade, T.J. and Brannigan, A. (2003) Parental assessment of early childhood development: Biological and social covariates. *Infant and Child Development*, 12: 167–175.

DEMONSTRATION EXERCISE

In Chapter 3 we manipulated the individual level variables and saved a new data set called demodata1.dta. In Chapter 4 we merged a household level variable indicating the region of the country onto the individual level data and saved the data with a new name demodata2.dta. In Chapter 5 we examined the variables we are using for their distribution, measures of central tendency and, for continuous variables, their normality. In Chapter 6 we examined differences in mean GHQ scale scores across groups in the

factors but did not formally test for differences. The dichotomous indicator was tested using the **tab** command and measures of association. Correlations between the GHQ scale and interval level factors were produced. In Chapter 7 we formally tested for differences of mean GHQ scores and proportions above the threshold of the dichotomous GHQ indicator between groups.

At this stage of this demonstration we use multivariate OLS regression with the GHQ scale as the dependent variable and then use multivariate binary logistic regression with the dichotomous GHQ indicator as the dependent variable. In these models we use all of the factors we are interested in to assess their net effects on mental well-being.

In this first regression model we use the **xi:** prefix as we have a number of categorical independent variables which need to be converted into indicator or dummy variables. We also use the age categories to see if the association with age is linear or non-linear.

```
xi:reg ghqscale female i.agecat i.marst2 ///
   i.empstat i.numchd i.region2
```

In the output below we have put the significant coefficients in bold for easier identification. These results indicate that women have on average higher GHQ scores by 1.05 points. The second age category (33–50 years) has significantly higher GHQ scores than the reference (youngest) category (18–32 years), whereas the third category (51–65 years) is not significantly different from the reference category. This suggests that the association is non-linear and possibly could be better defined with a quadratic term for age. The dummy variables for marital status categories show that those who are married are not significantly different from the reference category (single) but those who are separated or divorced (category 3) and widowed (category 4) have significantly higher GHQ scores than those who are single. Most of the people in this sample are married, so it may be more appropriate to use the married category as the reference, and we will change this in the next regression model. For employment status, three of the categories (unemployed, long term sick and family care) have significantly higher GHQ scores than the reference category (employed). Those with one or two children in the household have significantly higher GHQ scores than those with no children, but those with three or more children are not significantly different from those with no children.

```
. xi:reg ghqscale female i.agecat i.marst2 i.empstat ///
   i.numchd i.region2
i.agecat     _Iagecat_1-3     (naturally coded; _Iagecat_1 omitted)
i.marst2     _Imarst2_1-4     (naturally coded; _Imarst2_1 omitted)
i.empstat    _Iempstat_1-6    (naturally coded; _Iempstat_1 omitted)
i.numchd     _Inumchd_1-3     (naturally coded; _Inumchd_1 omitted)
i.region2    _Iregion2_1-7    (naturally coded; _Iregion2_1 omitted)

      Source |       SS       df       MS              Number of obs =    7688
-------------+------------------------------           F( 19, 7668)  =   35.04
       Model |  15242.5628     19  802.240149           Prob > F      =  0.0000
    Residual |  175575.003   7668  22.8971053           R-squared     =  0.0799
-------------+------------------------------           Adj R-squared =  0.0776
       Total |  190817.566   7687  24.8234117           Root MSE      =  4.7851

------------------------------------------------------------------------------
    ghqscale |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      female |    1.051762   .1179439     8.92   0.000     .8205601    1.282965
   _Iagecat_2 |    .3340761   .1366896     2.44   0.015     .0661272     .602025
   _Iagecat_3 |   -.075252   .1838687    -0.41   0.682    -.435685     .285181
   _Imarst2_2 |    .1601081   .1709917     0.94   0.349    -.1750823    .4952986
   _Imarst2_3 |    1.597959   .2581717     6.19   0.000     1.091871    2.104046
   _Imarst2_4 |    1.489015   .4102427     3.63   0.000     .6848276    2.293203
  _Iempstat_2 |    2.912573   .2282301    12.76   0.000     2.465179    3.359966
  _Iempstat_3 |    5.155082   .3274935    15.74   0.000     4.513105    5.797058
  _Iempstat_4 |    .5919008   .3422444     1.73   0.084    -.0789917    1.262793
  _Iempstat_5 |    1.123951   .1904472     5.90   0.000     .7506226     1.49728
  _Iempstat_6 |   -.2368232   .2823437    -0.84   0.402    -.790294     .3166477
   _Inumchd_2 |    .4985591   .1438261     3.47   0.001     .2166205    .7804976
   _Inumchd_3 |    .4114755   .2415769     1.70   0.089    -.0620813    .8850324
  _Iregion2_2 |   -.2086025   .1956044    -1.07   0.286    -.5920406    .1748356
  _Iregion2_3 |    -.172131     .21373    -0.81   0.421    -.5911002    .2468381
  _Iregion2_4 |   -.2925766   .2381277    -1.23   0.219    -.7593719    .1742188
  _Iregion2_5 |    -.141722   .2164165    -0.65   0.513    -.5659575    .2825135
  _Iregion2_6 |    .4176472   .2944849     1.42   0.156    -.1596238    .9949182
  _Iregion2_7 |   -.2650218   .2430078    -1.09   0.275    -.7413834    .2113399
        _cons |    9.322474   .2124677    43.88   0.000      8.90598    9.738969
------------------------------------------------------------------------------
```

There are no significant differences for the dummy variables for region of the country compared to the reference category of London. However, if you examine the coefficients more closely you can see that category 6 (Wales) is 0.417 higher than the reference category and category 4 (Northwest) is 0.292 lower than the reference category. This difference might be significant, but is not tested in this model. We can, however, test this with a post-estimation command:

```
test _Iregion2_6= _Iregion2_4
```

```
. test _Iregion2_6= _Iregion2_4

( 1) - _Iregion2_4 + _Iregion2_6 = 0

       F( 1, 7668) = 5.84
           Prob > F = 0.0157
```

The output above tests for a difference between the two coefficients and the *p* value of the test is less than 0.05 which suggests that they are different. However, for a variable such as *region2* we might want to see if the regions are significantly different from the overall sample mean rather than choose a reference category.

From our observations above, we need to adjust some of the variables and commands to re-estimate this regression model. First, we wish to capture the non-linear nature of the association between age and GHQ score by adding a squared age term to the model. Therefore, we need to create a new variable for the squared value of age.

```
gen age2=age^2
```

Next, we want to change the reference category for marital status to married (category 2).

```
char marst2 [omit] 2
```

Finally, we want the coefficients for the region categories to be compared to the overall or grand mean in the sample. To do this we need to have downloaded the **xi3** command/prefix (see Box 8.7). Dummy variables that indicate differences from the grand mean are usually referred to as effect coding, and this is done by prefixing the regression command with **xi3:** and then prefixing the region variable with **e.** (rather than **i.**). So, the new regression looks like this:

```
xi3:reg ghqscale female age age2 i.marst2 ///
   i.empstat i.numchd e.region2
```

```
. xi3:reg ghqscale female age age2 i.marst2 ///
   i.empstat i.numchd e.region2
i.marst2     _Imarst2_1-4      (naturally coded; _Imarst2_2 omitted)
i.empstat    _Iempstat_1-6     (naturally coded; _Iempstat_1 omitted)
i.numchd     _Inumchd_1-3      (naturally coded; _Inumchd_1 omitted)
e.region2    _Iregion2_1-7     (naturally coded; _Iregion2_1 omitted)
```

```
      Source |       SS       df       MS      Number of obs =    7688
-------------+------------------------------   F( 19,  7668) =   35.66
       Model |  15491.6833    19  815.351754   Prob > F      =  0.0000
    Residual |  175325.883  7668  22.8646169   R-squared     =  0.0812
-------------+------------------------------   Adj R-squared =  0.0789
       Total |  190817.566  7687  24.8234117   Root MSE      =  4.7817
```

ghqscale	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
female	**1.045367**	.1179395	**8.86**	**0.000**	.814173	1.27656
age	**.1502465**	.0342705	**4.38**	**0.000**	.083067	.2174259
age2	**-.0018732**	.000422	**-4.44**	**0.000**	-.0027005	-.001046
_Imarst2_1	-.0464479	.1816749	-0.26	0.798	-.4025804	.3096846
_Imarst2_3	**1.415416**	.2165824	**6.54**	**0.000**	.990855	1.839977
_Imarst2_4	**1.438633**	.3810207	**3.78**	**0.000**	.6917286	2.185538
_Iempstat_2	**2.973164**	.2288571	**12.99**	**0.000**	2.524541	3.421786
_Iempstat_3	**5.182441**	.3275309	**15.82**	**0.000**	4.540391	5.824491
_Iempstat_4	**.7544566**	.3465094	**2.18**	**0.029**	.0752034	1.43371
_Iempstat_5	**1.166145**	.1902912	**6.13**	**0.000**	.7931224	1.539168
_Iempstat_6	.1311709	.3059574	0.43	0.668	-.4685892	.730931
_Inumchd_2	**.4154971**	.1458797	**2.85**	**0.004**	.129533	.7014611
_Inumchd_3	.2894564	.2449152	1.18	0.237	-.1906445	.7695572
_Iregion2_2	-.1058743	.1047368	-1.01	0.312	-.3111871	.0994386
_Iregion2_3	-.0641072	.1273939	-0.50	0.615	-.3138341	.1856197
_Iregion2_4	-.196559	.1551697	-1.27	0.205	-.500734	.1076161
_Iregion2_5	-.0411748	.1305821	-0.32	0.753	-.2971515	.2148019
_Iregion2_6	**.5126046**	.2130926	**2.41**	**0.016**	.0948849	.9303244
_Iregion2_7	-.1743432	.1608996	-1.08	0.279	-.4897504	.141064
_cons	6.794737	.665037	10.22	0.000	5.491082	8.098391

The significant coefficients for both the *age* and *age2* variables indicate that we were correct to model a non-linear association, and the positive coefficient for the *age* variable and the negative coefficient for the *age2* variable show that the association first increases with age and then decreases in an inverted U shape.

The categories of marital status show that those separated or divorced and those widowed have significantly higher GHQ scores than those who are married. If you examine the dummy variables for marital status you can see that now category 2 (*_Imarst2_2*) is missing and is therefore the reference category.

The dummy variables for employment status were not altered but you can see that the coefficient for category 4 (*_Iempstat_4*) is now significant, which it wasn't in the first regression model. The coefficient is larger, which may have resulted from the better specification of other variables in the model.

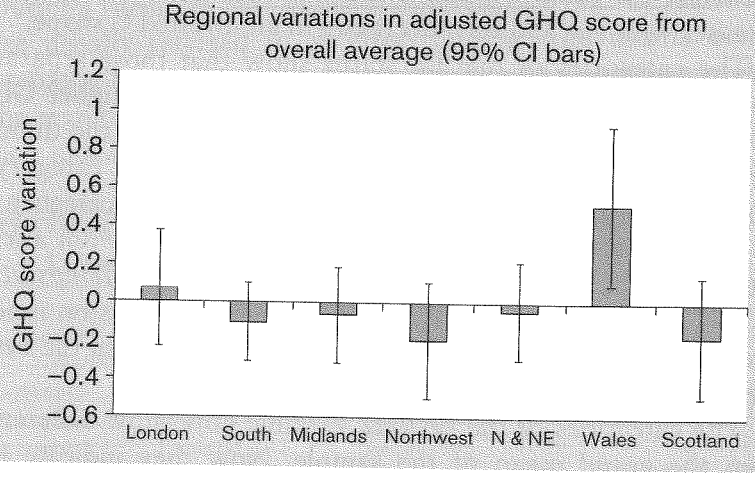
The categories of the *region2* variable now show differences from the grand mean of the sample. Now they indicate that

category 6 (*_Iregion2_6*) has significantly higher GHQ scores than the sample average. You can see that even though the coefficients now show difference from the grand mean, one of the categories is still missing. If you wish to find the difference for this category then you can rerun the regression omitting another category of the *region2* variable. Omitting category 2 produces the following extract of results:

```
_Iregion2_1 |   .0694538   .1555509     0.45   0.655   -.2354686    .3743762
_Iregion2_3 |  -.0641072   .1273939    -0.50   0.615   -.3138341    .1856197
_Iregion2_4 |   -.196559   .1551697    -1.27   0.205    -.500734    .1076161
_Iregion2_5 |  -.0411748   .1305821    -0.32   0.753   -.2971515    .2148019
_Iregion2_6 |   .5126046   .2130926     2.41   0.016    .0948849    .9303244
_Iregion2_7 |  -.1743432   .1608996    -1.08   0.279   -.4897504     .141064
```

Box 8.10: Graphing effect coded categorical variables

Effect coding categorical independent variables gives you the opportunity to graph the information in a way that is intuitively attractive and logical. Copy and paste the regression results into an Excel spreadsheet (see Chapter 2) and then add the coefficient and confidence interval for the omitted category from rerunning the regression with another category omitted (by using the **char** command). Now you can graph the categories' differences from the grand mean along with the 95% confidence interval, and the resulting graph shows very clearly that, on average, Wales has significantly higher GHQ scores than the sample average after controlling for sex, age, marital status, employment status and number of children.



Alternatively, you can add the coefficient from all the other categories, and then the difference from zero is the omitted coefficient. For example, from the extract, 0.06945 – 0.06410 – 0.19655 – 0.04117 + 0.51260 – 0.17434 = 0.10587. The difference from zero is –0.10587 (as with effect coding all the differences add to zero, see Box 8.10) which, if you check the previous output, is the coefficient for category 2 (*_Iregion2_2*).

Now we run a logistic regression using the binary GHQ indicator (*d_ghq*) and the same independent variables as on p. 312.

```
xi3:logistic d_ghq female i.agecat i.marst2 ///
    i.empstat i.numchd e.region2
```

```
. xi3:logistic d_ghq female i.agecat i.marst2 ///
   i.empstat i.numchd e.region2
i.agecat     _Iagecat_1-3     (naturally coded; _Iagecat_1 omitted)
i.marst2     _Imarst2_1-4     (naturally coded; _Imarst2_2 omitted)
i.empstat    _Iempstat_1-6    (naturally coded; _Iempstat_1 omitted)
i.numchd     _Inumchd_1-3     (naturally coded; _Inumchd_1 omitted)
e.region2    _Iregion2_1-7    (naturally coded; _Iregion2_1 omitted)

Logistic regression                               Number of obs =    7688
                                                  LR chi2(19)   =  339.92
                                                  Prob > chi2   =  0.0000
Log likelihood = -3536.4272                       Pseudo R2     =  0.0459

------------------------------------------------------------------------------
      d_ghq | Odds Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
------------+-----------------------------------------------------------------
     female |   1.389437   .0920517     4.96   0.000     1.220242    1.582093
  _Iagecat_2 |   .9136479   .0672408    -1.23   0.220     .7909223    1.055417
  _Iagecat_3 |   .7308843   .0755579    -3.03   0.002     .5968326    .8950448
  _Imarst2_1 |    .906441   .0862042    -1.03   0.302      .752296     1.09217
  _Imarst2_3 |   1.733206   .1806035     5.28   0.000     1.413036    2.125921
  _Imarst2_4 |   1.521458   .2964534     2.15   0.031     1.038496    2.229026
 _Iempstat_2 |   3.248623   .3395573    11.27   0.000     2.646847    3.987217
 _Iempstat_3 |   5.355798   .7744645    11.61   0.000     4.034019     7.11067
 _Iempstat_4 |    1.39944   .2578951     1.82   0.068      .975194    2.008249
 _Iempstat_5 |   1.406923   .1355923     3.54   0.000     1.164758    1.699437
 _Iempstat_6 |   .8380977   .1500266    -0.99   0.324     .5900955    1.190329
  _Inumchd_2 |   1.175711   .0914464     2.08   0.037     1.009472    1.369326
  _Inumchd_3 |   1.041569   .1355878     0.31   0.754     .8070155    1.344295
 _Iregion2_2 |   .9657438    .055782    -0.60   0.546     .8623746    1.081503
 _Iregion2_3 |   .9555518   .0665694    -0.65   0.514     .8335937    1.095353
 _Iregion2_4 |   .9810842   .0826423    -0.23   0.821     .8317729    1.157198
 _Iregion2_5 |   .9029198   .0653833    -1.41   0.158     .7834493    1.040609
 _Iregion2_6 |   1.344802   .1439638     2.77   0.006     1.090274    1.658751
 _Iregion2_7 |   .8339696    .076168    -1.99   0.047     .6972819     .997452
------------------------------------------------------------------------------
```

The results of the logistic regression show similar associations to those in the OLS regression models. One noticeable difference is the association with age. In the OLS models the association was non-linear and best captured with a quadratic term, but the above output, using dummy variables for the age categories (shaded), shows a decreasing likelihood of being over the GHQ threshold with age, thus suggesting that a linear term can capture the association. Using the interval-level *age* variable produced the following coefficient in the logistic regression (other output omitted):

d_ghq	Odds Ratio	Std. Err.	z	P>|z|	[95% Conf.	Interval]
female	1.386245	.0918595	4.93	0.000	1.217405	1.578501
age	.9895113	.0030573	-3.41	0.001	.9835373	.9955217