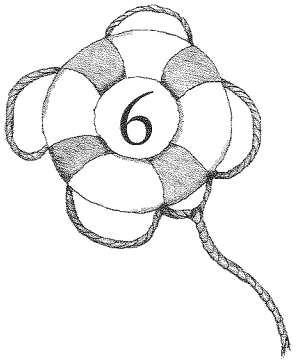


Tables and Correlations

In this chapter we look at some common forms of analysis: tables and correlations. We start off with two-way tables. These are sometimes called crosstabulations or crosstabs. Two-way tables are tables that use the categories of two variables to form a grid of cells. From this information, measures of association are calculated. Our explanations of the statistical tests are very brief and we focus on the basic interpretation of output. We encourage readers to consult sources that focus on the detailed explanation of these types of tests if they desire information in greater depth; for example, Agresti (2007) and Liebetrau (1983). The measures of association that you can use are largely contingent upon the level of measurement of your two variables. In Box 6.1 we present a matrix of potential measures of association by level of measurement of the two variables. The matrix is not exhaustive and other measures of association can be added as you come across them.

After two-way tables and measures of association, we move on to tables that incorporate summary statistics of variables. These can use two, three, four or even five variables in their construction.

In the section on correlations we look at parametric tests, including partial correlations and non-parametric statistics.

TWO-WAY TABLES

The **`tabulate`** command (shortened to **`tab`** or **`ta`**) is used with two variables to create a two-way table or crosstabulation. The level of measurement of both variables can be nominal, ordinal, or interval. While it is possible to do crosstabulations with continuous variables, the output is tedious and difficult to interpret. Furthermore, there are far more appropriate tests to use on continuous variables, as will be discussed later in Chapter 7.

Box 6.1: Matrix for appropriate tests depending on levels of measurement

'Measures of association' is a generic term for numerous statistics. There are too many to cover in this table, so five have been chosen that cover common usage. You can add measures of association to the table when you come across them.

	dichotomous	nominal	ordinal	interval
dichotomous	phi (φ)			
nominal	chi-squared (χ^2)	chi-squared (χ^2)		
ordinal	chi-squared (χ^2)	chi-squared (χ^2)	Spearman's rho (ρ)	
interval	eta (η)*	eta (η)*	Spearman's rho (ρ)	Pearson's *r*

* Provided the interval variable is dependent. See also differences between groups in the next chapter.

If you expect a causal association between the two variables, it is perhaps easiest to put the independent variable (i.e. the 'cause') in the columns and the dependent variable (i.e. the 'effect') in the rows. The first variable after the **tab** command makes the rows of the table while the second makes the columns. In other words, after typing **tab** (or **ta**) you would then type your dependent variable followed by your independent variable. It should be emphasized here that all the techniques described in this chapter do not assume causality. Basically what are you are testing using bivariate tests is whether there is an association – and quite literally what you should be asking when doing a crosstab is, 'Does the distribution of the values of one variable depend on the categories of another variable?' That is all crosstabulations show you – nothing more. In the case of crosstabulations using ordinal variables (described below), the question is slightly modified to, 'Is there an ordered relationship between the ordered distribution of categories?'

For ease of viewing some tables it is best to put the variable with the most categories in the rows so that the table doesn't break across the output screen. The first variable after the **tab** command will form the rows of the table and the second will form the columns. Try:

tab sex jbstat

and you will see that the output is divided into two parts which makes interpretation a little more difficult.

sex	current labour force status: self empl	in paid e	unemploye	retired	family ca	ft studen	Total
male	557	2,467	374	764	15	218	4,599
female	203	2,507	161	999	1,117	180	5,313
Total	760	4,974	535	1,763	1,132	398	9,912

sex	current labour force status: long term	on matern	govt trng	something	Total
male	160	0	33	11	4,599
female	106	13	4	23	5,313
Total	266	13	37	34	9,912

Now compare it with:

tab jbstat sex

shown below.

Following on with a table of sex by labour force status measured in 10 categories (*jbstat*) where sex is considered 'the cause' and job status is considered the outcome variable of interest ('the effect'):

tab jbstat sex

```
. ta jbstat sex
```

labour force status	sex: male	female	Total
self employed	557	203	760
in paid employ	2,467	2,506	4,973
unemployed	374	162	536
retired	764	999	1,763
family care	15	1,117	1,132
ft student	218	180	398
long term sick/disabl	160	106	266
on matern leave	0	13	13
govt trng scheme	33	4	37
something else	11	23	34
Total	4,599	5,313	9,912

The two-way table produced above shows the frequencies in each cell. We can see, for example, that 557 males reported being in self-employment, compared to 203 females. There were a total of 760 people who reported being self-employed. The default settings for this command also give you row and column totals. So we know that there were 4599 males and 5313 females which sums to 9912 who are included in the table calculations. You can add a variety of options to **tab**, but here we show you some of the most popular ones.

As the missing values have already been coded to missing (.), cases are not included in the table if they are missing on either of the variables. If you want to see missing values in the table then you can use the option **missing**, which can be shortened to **m**:

```
tab jbstat sex, m
```

```
. ta jbstat sex, m

                    |        sex
  labour force status |    male   female |     Total
----------------------+----------------------+-------
          self employed |     557      203 |       760
          in paid employ |   2,467    2,506 |     4,973
              unemployed |     374      162 |       536
                 retired |     764      999 |     1,763
             family care |      15    1,117 |     1,132
              ft student |     218      180 |       398
long term sick/disabl |     160      106 |       266
       on matern leave |       0       13 |        13
       govt trng scheme |      33        4 |        37
         something else |      11       23 |        34
                       . |     234      118 |       352
----------------------+----------------------+-------
                   Total |   4,833    5,431 |    10,264
```

You can see from this table that there are no cases with missing values on *sex* but 352 cases with missing values on *jbstat* as shown by the row category with a dot (.); 234 males and 118 females.

Measures of association are available with the option **all**. They will tell you if the results reported in your table are statistically significant. In other words, how likely is it that the results are due to chance alone, or how likely is it that they represent true

associations in the population? Pearson's chi-squared, likelihood-ratio chi-squared, Cramér's *V*, gamma, and Kendall's tau-b are reported.

As with most software, Stata will compute the statistics but you have to make the decision about which statistics are the most appropriate. In the example below Stata will produce tau-b and gamma statistics, but these are not appropriate for this table.

tab jbstat sex, all

```
. ta jbstat sex, all

                    |        sex
  labour force status |    male  female |  Total
----------------------+----------------+-------
        self employed |     557     203 |    760
        in paid employ |   2,467   2,506 |  4,973
            unemployed |     374     162 |    536
               retired |     764     999 |  1,763
           family care |      15   1,117 |  1,132
            ft student |     218     180 |    398
long term sick/disabl |     160     106 |    266
      on matern leave |       0      13 |     13
     govt trng scheme |      33       4 |     37
       something else |      11      23 |     34
----------------------+----------------+-------
                Total |   4,599   5,313 |  9,912

          Pearson chi2(9) = 1.4e+03  Pr   = 0.000
 likelihood-ratio chi2(9) = 1.7e+03  Pr   = 0.000
             Cramér's V = 0.3709
                  gamma = 0.3142     ASE = 0.015
        Kendall's tau-b = 0.1879     ASE = 0.009
```

The first two measures of association – Pearson's and likelihood ratio chi-squared – are shown with their *p* value (i.e. their likelihood that the results are due to chance alone). A general guideline is that you would be looking for these values to be less than 0.05 in order to consider them statistically significant, although other common values are 0.01 and 0.001. These values should be determined a priori (before conducting the tests!). You will also notice that there is a number in parentheses after `chi2`. This is the degrees of freedom associated with the statistical test. If you look

at a table for the chi-square distribution (which is at the back of most statistics textbooks), you would look for 9 degrees of freedom and a 0.95 level of significance. You would find a value of 16.92. The chi-square statistic reported is in excess of 1000 (which is a lot bigger than 16.92!) and therefore you can reject the null hypothesis that the distribution of one of the variables did not depend on the categories of another. The *p* value is a shortcut to this information (rejecting the null). In other words, both pieces of information lead to the same conclusion.

It is useful to briefly explain the other measures of association. Cramér's *V* statistic is reported on its own (it ranges from 0 to 1, with 1 indicating a perfect relationship) and is relevant here because both variables are nominal. The value reported is 0.3710, which is considered strong (generally values around 0.25 are considered strong; see Liebetrau 1983). Although not relevant to the current example as both variables used in the crosstabulation are nominal, gamma and Kendall's tau-b range between −1 and +1 and are reported with their asymptotic standard error (ASE), which is a type of standard error that can be used for statistical significance tests (i.e. in place of *p*).

If you just want one of the measures of association use **chi2** for Pearson's chi-squared, **lrchi2** for likelihood-ratio chi-squared, **gamma** for gamma, **v** for Cramér's V, and **taub** for Kendall's tau-b.

The statistic for Fisher's exact test (typically used for 2 × 2 tables) can be obtained with the option **exact**. An example would be if we created a table of a dichotomous measure of being married (1 = married, 0 = not married) with sex.

```
tab married sex, exact
```

```
. ta married sex, exact
    married |        sex
  indicator |    male   female |     Total
------------+------------------+---------
not married |   1,886    2,369 |     4,255
    married |   2,947    3,062 |     6,009
------------+------------------+---------
      Total |   4,833    5,431 |    10,264

          Fisher's exact = 0.000
  1-sided Fisher's exact = 0.000
```

We can see that the *p* value is 0.000, which is less than the standard *p* value of 0.05, so we can conclude that the distribution in the table is unlikely to be due to chance alone. The one-sided Fisher's exact test is used if you have a directional hypothesis – for instance, if you had a reason to believe the females were more likely to be married than males, rather than a two-tailed test, which just hypothesizes that there will be some kind of difference in the distribution of the cases in the table by sex. Typically it is about half the size of the Fisher's exact test, although in our example the *p* value is so small that dividing it by 2 still produces 0.000.

Percentages can be included in the table cells by using options. If you are interested in including row percentages in your crosstabulation, use the **row** option after the **tab** command. The following examples use the variables self-reported health status (*hlstat*) and *sex*.

tab hlstat sex, row

```
. tab hlstat sex, row
+----------------+
| Key            |
|----------------|
|    frequency   |
| row percentage |
+----------------+

    health over |        sex
 last 12 months |    male   female |     Total
----------------+-------------------+---------
      excellent |   1,536    1,394 |     2,930
                |   52.42    47.58 |    100.00
----------------+-------------------+---------
           good |   2,149    2,464 |     4,613
                |   46.59    53.41 |    100.00
----------------+-------------------+---------
           fair |     808    1,045 |     1,853
                |   43.60    56.40 |    100.00
----------------+-------------------+---------
           poor |     246      395 |       641
                |   38.38    61.62 |    100.00
----------------+-------------------+---------
      very poor |      93      126 |       219
                |   42.47    57.53 |    100.00
----------------+-------------------+---------
          Total |   4,832    5,424 |    10,256
                |   47.11    52.89 |    100.00
```

You can now see that of all the individuals who reported excellent health, 52.42% were males and 47.58% were females. Similarly, if you are interested in column percentages you would type **column** or **col**

```
tab hlstat sex, col
```

```
. tab hlstat sex, col

+--------------------+
| Key                |
|--------------------|
|     frequency      |
| column percentage  |
+--------------------+

    health over |        sex
 last 12 months |      male     female |     Total
----------------+----------------------+----------
      excellent |     1,536      1,394 |     2,930
                |     31.79      25.70 |     28.57
----------------+----------------------+----------
           good |     2,149      2,464 |     4,613
                |     44.47      45.43 |     44.98
----------------+----------------------+----------
           fair |       808      1,045 |     1,853
                |     16.72      19.27 |     18.07
----------------+----------------------+----------
           poor |       246        395 |       641
                |      5.09       7.28 |      6.25
----------------+----------------------+----------
      very poor |        93        126 |       219
                |      1.92       2.32 |      2.14
----------------+----------------------+----------
          Total |     4,832      5,424 |    10,256
                |    100.00     100.00 |    100.00
```

The column percentages tell you that of out of all males, 31.79% reported having excellent health compared to 25.70% of all females.

The option **cell** tells Stata to produce cell percentages where the total of the cell percentages equals 100%.

tab hlstat sex, cell

```
. tab hlstat sex, cell

+-----------------+
| Key             |
|-----------------|
|     frequency   |
| cell percentage |
+-----------------+

    health over |        sex
 last 12 months |      male     female |     Total
----------------+----------------------+---------
      excellent |     1,536      1,394 |     2,930
                |     14.98      13.59 |     28.57
----------------+----------------------+---------
           good |     2,149      2,464 |     4,613
                |     20.95      24.02 |     44.98
----------------+----------------------+---------
           fair |       808      1,045 |     1,853
                |      7.88      10.19 |     18.07
----------------+----------------------+---------
           poor |       246        395 |       641
                |      2.40       3.85 |      6.25
----------------+----------------------+---------
      very poor |        93        126 |       219
                |      0.91       1.23 |      2.14
----------------+----------------------+---------
          Total |     4,832      5,424 |    10,256
                |     47.11      52.89 |    100.00
```

In this table the marginals – the right-hand column and the bottom row – add up to 100% while all the cells 'inside' the table also add up to 100%, so, for example, males who report their health as 'good' ($N = 2149$) are 20.95% of the whole sample ($N = 10{,}256$).

You can see in the three examples above that Stata gives you a key to the cell contents which becomes more useful if you combine options. You can combine options to give you the exact table you need. For example, if we wish to have column and cell percentages along with Pearson's chi-squared and likelihood ratio chi-squared statistics:

tab hlstat sex, col cell chi2 lr

```
. tab hlstat sex, col cell chi lr
```

Key
frequency
column percentage
cell percentage

health over last 12 months	sex male	sex female	Total
excellent	1,536	1,394	2,930
	31.79	25.70	28.57
	14.98	13.59	28.57
good	2,149	2,464	4,613
	44.47	45.43	44.98
	20.95	24.02	44.98
fair	808	1,045	1,853
	16.72	19.27	18.07
	7.88	10.19	18.07
poor	246	395	641
	5.09	7.28	6.25
	2.40	3.85	6.25
very poor	93	126	219
	1.92	2.32	2.14
	0.91	1.23	2.14
Total	4,832	5,424	10,256
	100.00	100.00	100.00
	47.11	52.89	100.00

```
          Pearson chi2(4) = 64.3546   Pr = 0.000
likelihood-ratio chi2(4) = 64.5617   Pr = 0.000
```

If you prefer to copy and paste your tables into Excel for graphing, a useful option is **nofreq**. This tells Stata not to report

the cell or marginal frequencies. It must be combined with another option, otherwise no output is shown. In our example we wish to see column percentages so we add the **col** option.

```
tab hlstat sex, nofreq col
```

```
. tab hlstat sex, nofreq col

    health over |        sex
 last 12 months |      male    female |     Total
----------------+----------------------+---------
      excellent |     31.79     25.70 |     28.57
           good |     44.47     45.43 |     44.98
           fair |     16.72     19.27 |     18.07
           poor |      5.09      7.28 |      6.25
      very poor |      1.92      2.32 |      2.14
----------------+----------------------+---------
          Total |    100.00    100.00 |    100.00
```

Note that as only one piece of information per cell has been requested that a table key is not shown. The table key can be suppressed by using the **nokey** option, but we suggest leaving it shown until you become familiar with the way Stata presents its output.

The **nofreq** option is also useful if you are only interested in the measures of association being reported rather than whole tables. You just type **nofreq** followed by the measure(s) of association you are interested to see. In the example below, we select only Pearson's chi-squared statistic.

```
tab hlstat sex, nofreq chi2
```

```
. tab hlstat sex, nofreq chi2
          Pearson chi2(4) =  64.3546  Pr = 0.000
```

CONDITIONAL CROSSTABULATIONS

In combination with the **bysort** command from Chapter 4, you can use the **tabulate** command to do conditional crosstabulations. For example, if you wanted to see crosstabulations of self-reported health (*hlstat*) by registered disabled status (*hldsbl*) for both sexes:

```
bysort sex: ta hlstat hldsbl
```

All the options of the **ta** command are still available when you separate your results using a **bysort** command, so in this example we also want each table to have column percentages along with Pearson's chi-squared statistic:

```
bysort sex: ta hlstat hldsbl, col chi2
```

```
. bysort sex: ta hlstat hldsbl, col chi2
-------------------------------------------------------------------------------
-> sex = male

+--------------------+
| Key                |
|--------------------|
|     frequency      |
| column percentage  |
+--------------------+

   health over |  registered disabled
last 12 months |       yes        no |     Total
---------------+--------------------+----------
     excellent |        10     1,519 |     1,529
               |      4.12     33.20 |     31.74
---------------+--------------------+----------
          good |        56     2,088 |     2,144
               |     23.05     45.64 |     44.50
---------------+--------------------+----------
          fair |        65       742 |       807
               |     26.75     16.22 |     16.75
---------------+--------------------+----------
          poor |        70       176 |       246
               |     28.81      3.85 |      5.11
---------------+--------------------+----------
     very poor |        42        50 |        92
               |     17.28      1.09 |      1.91
---------------+--------------------+----------
         Total |       243     4,575 |     4,818
               |    100.00    100.00 |    100.00

    Pearson chi2(4) =  701.5820  Pr = 0.000
```

```
-------------------------------------------------
-> sex = female

+--------------------+
| Key                |
|--------------------|
|     frequency      |
| column percentage  |
+--------------------+
```

health over last 12 months	registered disabled: yes	no	Total
excellent	3	1,387	1,390
	1.59	26.62	25.74
good	39	2,413	2,452
	20.63	46.31	45.41
fair	60	978	1,038
	31.75	18.77	19.22
poor	55	339	394
	29.10	6.51	7.30
very poor	32	94	126
	16.93	1.80	2.33
Total	189	5,211	5,400
	100.00	100.00	100.00

```
     Pearson chi2(4) = 393.3277  Pr = 0.000
```

You can see that the tables for self-reported health status and registered disabled are reported first for males and then for females. You can see that the chi-squared statistic was statistically significant for both groups with p values of 0.000.

You can also refine your analysis by using an **if** statement. The **if** statement tells Stata the condition(s) that you want to set. So, if you wanted to do a table of health status by sex but only for people who are in some age range, say between 20 and 30, you would use an **if** statement after the **ta** command. We set the age

range to be between 20 and 30 by specifying that cases where the age is greater than 19 and less than 31 should be included in the analysis.

```
ta hlstat sex if (age>19 & age<31),all
```

```
. ta hlstat sex if (age>19 & age<31),all

    health over |        sex
 last 12 months |    male   female |     Total
----------------+--------------------+----------
      excellent |     412      338 |       750
           good |     447      548 |       995
           fair |     135      201 |       336
           poor |      33       65 |        98
      very poor |       5       11 |        16
----------------+--------------------+----------
          Total |   1,032    1,163 |     2,195

          Pearson chi2(4) =  35.5252  Pr = 0.000
 likelihood-ratio chi2(4) =  35.7583  Pr = 0.000
                Cramér's V =   0.1272
                     gamma =   0.2060  ASE = 0.034
           Kendall's tau-b =   0.1180  ASE = 0.020
```

Notice that the **all** option goes after the **if** statement and that a space after the comma is not necessary.

Of course, you could take this a step even further. You could use the **bysort** command in combination with an **if** statement to further refine the tables. For example:

```
bysort sex: ta hlstat hldsbl if (age>59 & ///
   age<81),col chi2
```

This would generate two crosstabulations, one for men and one for women, of self-reported health by registered disabled status for persons aged 60–80. The tables would contain column percentages and report Pearson's chi-squared statistic.

```
. bysort sex: ta hlstat hldsbl if (age>59 & ///
   age<81),col chi2
```

```
-------------------------------------------------
-> sex = male

+--------------------+
| Key                |
|--------------------|
|     frequency      |
| column percentage  |
+--------------------+

    health over | registered disabled
 last 12 months |       yes        no |     Total
----------------+---------------------+----------
      excellent |         5       209 |       214
                |      4.24     25.49 |     22.81
----------------+---------------------+----------
           good |        25       354 |       379
                |     21.19     43.17 |     40.41
----------------+---------------------+----------
           fair |        32       195 |       227
                |     27.12     23.78 |     24.20
----------------+---------------------+----------
           poor |        36        51 |        87
                |     30.51      6.22 |      9.28
----------------+---------------------+----------
      very poor |        20        11 |        31
                |     16.95      1.34 |      3.30
----------------+---------------------+----------
          Total |       118       820 |       938
                |    100.00    100.00 |    100.00

     Pearson chi2(4) = 174.8807  Pr = 0.000

-------------------------------------------------
-> sex = female

+--------------------+
| Key                |
|--------------------|
|     frequency      |
| column percentage  |
+--------------------+
```

health over last 12 months	registered disabled yes	no	Total
excellent	0	210	210
	0.00	18.88	17.50
good	16	494	510
	18.18	44.42	42.50
fair	31	275	306
	35.23	24.73	25.50
poor	28	92	120
	31.82	8.27	10.00
very poor	13	41	54
	14.77	3.69	4.50
Total	88	1,112	1,200
	100.00	100.00	100.00

Pearson chi2(4) = 100.8323 Pr = 0.000

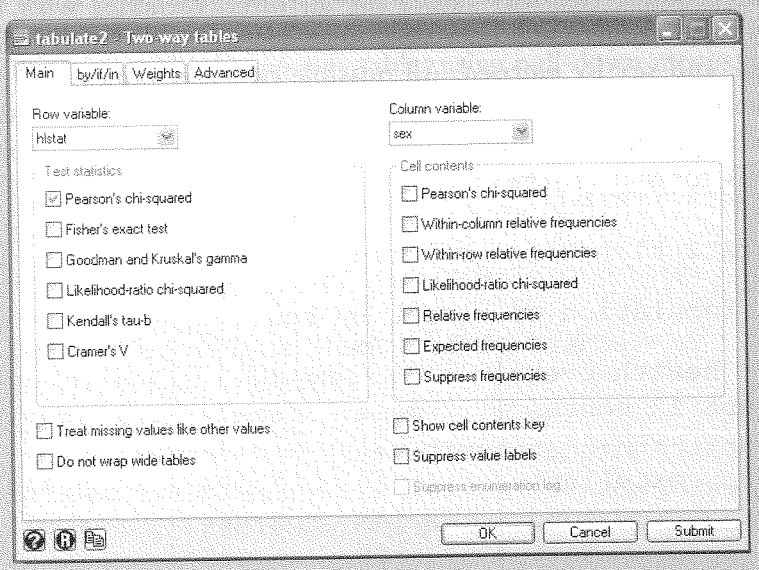
To produce two-way tables using pull-down menus, see Box 6.2.

Box 6.2: Crosstabulations using pull-down menus

To obtain crosstabulations or two-way tables by using the pull-down menus use the following path:

Statistics → Summaries, tables, and tests → Tables → Two-way tables with measures of association

This takes you to the **tabulate2 – Two-way tables** dialogue box and its **Main** tab shown below. The row and column variables are entered by typing in or scrolling down. All the other options for crosstabulations (test statistics and cell contents) are then ticked if they are required.



Which test statistics relate to the commands we have shown is fairly straightforward. However, some of the other options are not so clear so we show their equivalents in this table:

Code option	Pull-down menu tick box
missing	Treat missing values like other values
nofreq	Suppress frequencies
column	Within-column relative frequencies
row	Within-row relative frequencies
cell	Relative frequencies

In the **by/if/in** tab you can specify conditions for the tables equivalent to using the **bysort** and **if** commands.

MULTIPLE TWO-WAY TABLES

The command **tab2** will produce two-way tables between all variables in a variable list. For example, the command

```
tab2 hlstat sex married
```

will produce two-way tables for all three variables with one another. So there will be a table for *hlstat* by *sex*, *hlstat* by *married*, and *sex* by *married*. All possible two-way tables (i.e. pair combinations) will be produced, depending on the number of variables specified. For three variables, there are 3 possible combinations, for four variables there would be 6, and so on.

```
. tab2 hlstat sex married

-> tabulation of hlstat by sex

    health over |        sex
 last 12 months |    male   female |     Total
----------------+--------------------+---------
      excellent |   1,536     1,394 |     2,930
           good |   2,149     2,464 |     4,613
           fair |     808     1,045 |     1,853
           poor |     246       395 |       641
      very poor |      93       126 |       219
----------------+--------------------+---------
          Total |   4,832     5,424 |    10,256

-> tabulation of hlstat by married

    health over |  married indicator
 last 12 months |  not marri   married |     Total
----------------+----------------------+---------
      excellent |       1,187     1,743 |     2,930
           good |       1,870     2,743 |     4,613
           fair |         807     1,046 |     1,853
           poor |         283       358 |       641
      very poor |         102       117 |       219
----------------+----------------------+---------
          Total |       4,249     6,007 |    10,256

-> tabulation of sex by married

           |  married indicator
       sex |  not marri   married |     Total
-----------+----------------------+---------
      male |       1,886     2,947 |     4,833
    female |       2,369     3,062 |     5,431
-----------+----------------------+---------
     Total |       4,255     6,009 |    10,264
```

The **tab2** command can be combined with **bysort** and **if** in the same way as the **tab** command.

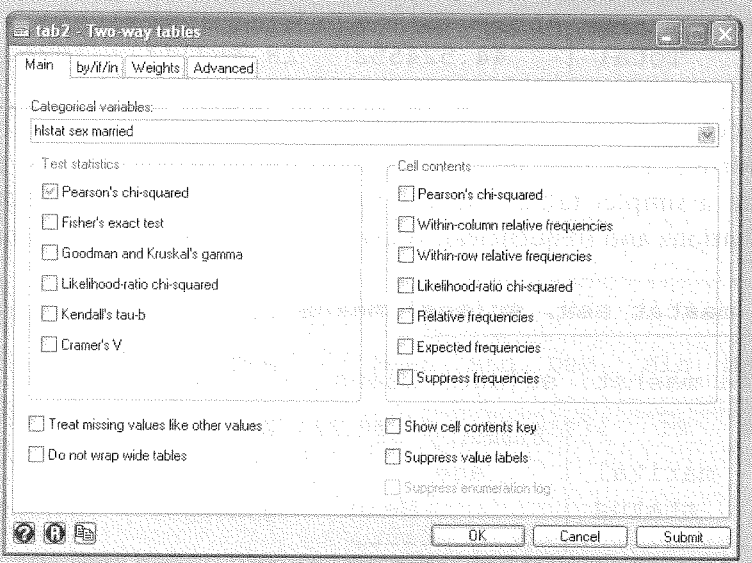
To create multiple two-way tables using pull-down menus, see Box 6.3.

Box 6.3: Multiple crosstabulations using pull-down menus

The pull-down menu equivalent of the **tab2** command is found at:

Statistics → Summaries, tables, and tests → Tables → All possible two-way tabulations

This takes you to the **tab2 – Two-way tables** dialogue box where you enter the list of variables by either typing or scrolling down and selecting by right-clicking on all the variables you want including in the list. All of the options are the same as the single two-way table window shown in Box 6.2.



COMBINING TABLES AND SUMMARY STATISTICS

You can use **su** as an option within a **tabulate** command to produce tables that report means and standard deviations for an interval variable across categories of a nominal or ordinal variable.

For example, if you wanted to see the mean ages of people according to their marital status you could use any of the following three commands.

The first command is a simple **ta** which uses **su** as an option in order to report the means of a second variable.

```
ta mastat, su(age)
```

```
. ta mastat, su(age)

    marital |           Summary of age
     status |        Mean   Std. Dev.   Freq.
------------+------------------------------------
    married |   47.198536   14.944372     6009
  living as |   32.378338   12.135964      674
    widowed |   72.182448   11.085019      866
   divorced |   46.919355    12.65734      434
  separated |   42.444444   14.628133      189
  never mar |   28.999044   15.932846     2092
------------+------------------------------------
      Total |   44.524552   18.467111    10264
```

By default the means, standard deviations and frequencies are reported. This may be more information than you need. If you want a simpler table that just reports means (and not standard deviations and frequencies), you would use **means** as a option:

```
ta mastat sex, su(age) means
```

```
. ta mastat, su(age) means

            | Summary of
    marital |     age
     status |        Mean
------------+------------
    married |   47.198536
  living as |   32.378338
    widowed |   72.182448
   divorced |   46.919355
  separated |   42.444444
  never mar |   28.999044
------------+------------
      Total |   44.524552
```

The next command uses **su** in combination with **bysort**. The output here differs from the previous example as summary statistics are presented separately for all marital status groups. Unlike the example above, the minimum and maximum values are also reported.

bysort mastat: su age

```
. bysort mastat: su age
-------------------------------------------------------
-> mastat = married
    Variable |    Obs       Mean   Std. Dev.   Min   Max
-------------+--------------------------------------------
         age |   6009   47.19854   14.94437    18    94
-------------------------------------------------------
-> mastat = living a
    Variable |    Obs       Mean   Std. Dev.   Min   Max
-------------+--------------------------------------------
         age |    674   32.37834   12.13596    17    91
-------------------------------------------------------
-> mastat = widowed
    Variable |    Obs       Mean   Std. Dev.   Min   Max
-------------+--------------------------------------------
         age |    866   72.18245   11.08502    27    96
-------------------------------------------------------
-> mastat = divorced
    Variable |    Obs       Mean   Std. Dev.   Min   Max
-------------+--------------------------------------------
         age |    434   46.91935   12.65734    22    85
-------------------------------------------------------
-> mastat = separate
    Variable |    Obs       Mean   Std. Dev.   Min   Max
-------------+--------------------------------------------
         age |    189   42.44444   14.62813    22    86
-------------------------------------------------------
-> mastat = never ma
    Variable |    Obs       Mean   Std. Dev.   Min   Max
-------------+--------------------------------------------
         age |   2092   28.99904   15.93285    16    97
```

As we've said before, there is often more than one way to get to essentially the same results. Another command that can produce similar types of tables for you is **tabstat**. You can get very similar results to the ones produced above with the following command:

```
tabstat age, by(mastat)
```

```
. tabstat age, by(mastat)

Summary for variables: age
     by categories of: mastat (marital status)

          mastat |      mean
-----------------+----------
         married |  47.19854
living as couple |  32.37834
         widowed |  72.18245
        divorced |  46.91935
       separated |  42.44444
   never married |  28.99904
-----------------+----------
           Total |  44.52455
-------------------------------
```

The command **tabstat** is for creating tables of summary statistics, typically for variables at the interval level (i.e. because the means, standard deviations, etc. for nominal and ordinal variables don't mean much). You can condition the results produced in the table by another variable, like we did above – we asked for the means of age conditioned on marital status. In this sense, **tabstat** is like the **ta** command combined with the option **su**. It is different in that you can request more types of statistics to be presented. For example, you can request the inter-quartile range (**iqr**), kurtosis (**k**), skewness (**ske**), variance (**v**) and coefficient of variation (**cv**), as well as others. See **help tabstat** for a full list of options.

For example, you could have Stata report means, minimum values and the 25th percentile using the following command:

```
tabstat age, by(mastat) stats(mean min p25)
```

```
. tabstat age, by(mastat) stats(mean min p25)

Summary for variables: age
     by categories of: mastat (marital status)

          mastat |      mean   min   p25
-----------------+--------------------
         married |   47.19854    18    35
living as couple |   32.37834    17    24
         widowed |   72.18245    27    67
        divorced |   46.91935    22    37
       separated |   42.44444    22    31
   never married |   28.99904    16    19
-----------------+--------------------
           Total |   44.52455    16    29
-------------------------------------------
```

The command **tabstat** also has the advantage of allowing you to customize the presentation of the table. For example, by using the command

```
tabstat age, by(mastat) s(mean range iqr) ///
   nototal
```

we can request that the mean, range, and inter-quartile range are reported, and the **nototal** option tells Stata not to report a final line with column totals.

```
. tabstat age, by(mastat) s(mean range iqr) ///
   nototal

Summary for variables: age
     by categories of: mastat (marital status)

          mastat |      mean   range   iqr
-----------------+--------------------
         married |   47.19854      76    23
living as couple |   32.37834      74    13
         widowed |   72.18245      69    13
        divorced |   46.91935      63    19
       separated |   42.44444      64    20
   never married |   28.99904      81    13
-------------------------------------------
```

We can also use the **col(var)** option to display the variables in the columns of the tables instead of the statistics, which is the default.

```
. tabstat age, by(mastat) stats(mean range ///
    iqr) col(var) nototal

Summary statistics: mean, range, iqr
  by categories of: mastat (marital status)

          mastat |        age
-----------------+-----------
         married |   47.19854
                 |         76
                 |         23
-----------------+-----------
living as couple |   32.37834
                 |         74
                 |         13
-----------------+-----------
         widowed |   72.18245
                 |         69
                 |         13
-----------------+-----------
        divorced |   46.91935
                 |         63
                 |         19
-----------------+-----------
       separated |   42.44444
                 |         64
                 |         20
-----------------+-----------
   never married |   28.99904
                 |         81
                 |         13
-----------------+-----------
```

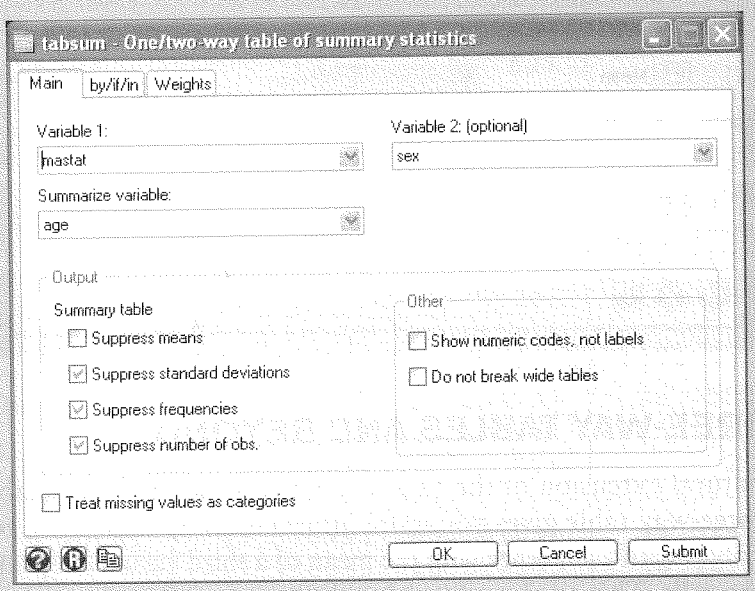
To create tables of summary statistics using pull-down menus, see Box 6.4.

Box 6.4: Tables of summary statistics

The pull-down menu equivalent of the **tab** command with the **su** option is found at:

Statistics → Summaries, tables, and tests → Tables → One/two-way table of summary statistics

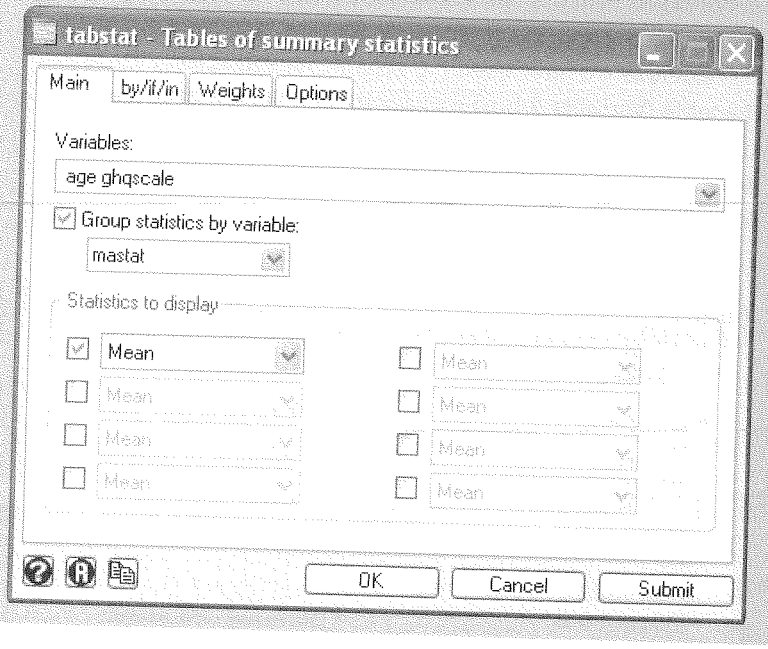
This brings you to the **Main** tab in the **tabsum – One/two-way table of summary statistics** dialogue box. This will also make one-way tables with summary statistics of a second variable. Either type in or scroll down and select the two variables for the categories of the table. The variable being summarized in the cells is entered in the **Summarize variable** box. There are options to customize the output. The default is for means, standard deviations and frequencies to be reported but each of these can be suppressed with the tick boxes. In the **by/if/in** tab you can specify conditions for the tables equivalent to using **bysort** and **if**. In this example we have used the variable *mastat* to form the rows of the table (**Variable 1** box) and *sex* to form the columns (**Variable 2** box) with the mean of the *age* variable reported in each cell. The standard deviations, frequencies and number of observations are all suppressed using the tick boxes.



▶ The pull-down menu equivalent of the **tabstat** command is found at:

Statistics → Summaries, tables, and tests → Tables → Table of summary statistics (tabstat)

This brings you to the **Main** tab in the **tabstat – Tables of summary statistics** dialogue box where you specify which variables you want summarized in the **Variables** box. The categorical variable to form the table is specified in the **Group statistics by variable** box. If this isn't specified then the summary statistics refer to the whole sample in the open data set. In this example we want to create a table of categories of the *mastat* variable that shows the mean of the variables *age* and *ghqscale* for each category. As with all the other tabulation commands, you can specify conditions for the tables equivalent to using **bysort** and **if** in the **by/if/in** tab.



THREE-WAY TABLES AND BEYOND

A natural extension of the two-way table is the three-way table. A three-way table gives you more information by tabulating two variables with each other with the mean of a third variable reported in the cells. This can be achieved with the generic command

```
ta variable1 variable2, su(variable3)
```

so you would end up with a table of *variable1* by *variable2*, but instead of the frequencies of each cell, you would have the mean value of *variable3* reported.

It makes more sense to use an example. Suppose we are interested in knowing the mean age of men and women by marital status. We can get this with the command:

```
ta mastat sex, su(age)
```

```
. ta mastat sex, su(age)

        Means, Standard Deviations and
              Frequencies of age

   marital |          sex
    status |       male     female |      Total
-----------+----------------------+-----------
   married |  48.603325  45.846506 |  47.198536
           |  15.081306  14.687464 |  14.944372
           |        2947        3062 |        6009
-----------+----------------------+-----------
 living as |  33.458084  31.317647 |  32.378338
           |   12.36032  11.833866 |  12.135964
           |         334         340 |         674
-----------+----------------------+-----------
   widowed |  73.704142  71.813486 |  72.182448
           |   10.53543  11.190343 |  11.085019
           |         169         697 |         866
-----------+----------------------+-----------
  divorced |  47.886667  46.408451 |  46.919355
           |  12.970559  12.481593 |   12.65734
           |         150         284 |         434
-----------+----------------------+-----------
 separated |  44.915254  41.323077 |  42.444444
           |  14.152236  14.756008 |  14.628133
           |          59         130 |         189
-----------+----------------------+-----------
 never mar |  28.035775  30.230937 |  28.999044
           |  14.262205  17.775071 |  15.932846
           |        1174         918 |        2092
-----------+----------------------+-----------
     Total |  43.370991  45.551096 |  44.524552
           |  17.986082  18.827175 |  18.467111
           |        4833        5431 |       10264
```

We can see here that the average age of a married man in this sample is 48.60 years, with a standard deviation of 15.08, while the average age of a married female is 45.85 with a standard deviation of 14.69. The overall average age of married people is 47.20 years, as indicated in the row total, while the overall average age of men in the sample is 43.37, and 45.55 years for females, as indicated in the column totals. If you did not want the standard deviations and frequencies which come with the default settings you can specify a **means** option:

```
ta mastat sex, su(age) means
```

```
. ta mastat sex, su(age) means

                Means of age

   marital |          sex
    status |       male     female |      Total
-----------+----------------------+-----------
   married |  48.603325  45.846506 |  47.198536
 living as |  33.458084  31.317647 |  32.378338
   widowed |  73.704142  71.813486 |  72.182448
  divorced |  47.886667  46.408451 |  46.919355
 separated |  44.915254  41.323077 |  42.444444
 never mar |  28.035775  30.230937 |  28.999044
-----------+----------------------+-----------
     Total |  43.370991  45.551096    44.524552
```

These ways of creating three-way tables using the **ta** command are also available for **tab2**, so, for example, the command below would create three two-way tables each with the mean age in the cells.

```
tab2 hlstat sex married, su(age) means
```

The **table** command in Stata is useful for producing customized tables, particularly three-way tables. Look at the results for:

```
table mastat sex, c(m age)
```

```
. table mastat sex, c(m age)
```

```
------------------------------------
                |        sex
 marital status |      male     female
----------------+-------------------
        married |   48.6033    45.8465
living as couple|   33.4581    31.3176
        widowed |   73.7041    71.8135
       divorced |   47.8867    46.4085
      separated |   44.9153    41.3231
  never married |   28.0358    30.2309
------------------------------------
```

Note that the **c** in the command means content of the cells and the **m** indicates that the mean value of the *age* variable should be displayed. As with other tabulation commands, there is a variety of options for reported statistics and presentation. For example, we use the **table** command to also include the median (**med**) of the *ghqscale* variable in the cells and format (**f**) the cells to reduce the number of decimal places shown. The **table** command is the most flexible way of presenting results such as these.

```
table mastat sex, c(m age med ghqscale) ///
   f(%4.2f)
```

```
. table mastat sex, c(m age med ghqscale) ///
   f(%4.2f)

------------------------------------
                |        sex
 marital status |      male     female
----------------+-------------------
        married |     48.60      45.85
                |      9.00      10.00
living as couple|     33.46      31.32
                |      9.00      10.00
        widowed |     73.70      71.81
                |     10.00      11.00
       divorced |     47.89      46.41
                |     10.00      12.00
      separated |     44.92      41.32
                |     11.00      12.00
  never married |     28.04      30.23
                |      9.00      10.00
------------------------------------
```

The **table** command has ability to create 'four-way' and 'five-way' tables as well (see Box 6.5), with various customizing options for statistics and presentation. For example, the command

```
table jbstat married sex, c(m age)
```

will produce a table for labour force status (*jbstat*) by *married* and *sex*, with the mean values of *age* reported in the cells.

```
. table jbstat married sex, c(m age)
```

	sex and married indicator			
	------ male ------		----- female ------	
labour force status	not married	married	not married	married
self employed	38.1635	44.9221	40.0877	43.4795
in paid employ	29.9297	42.2769	32.2243	41.1005
unemployed	30.848	42.9353	29.6754	37.2917
retired	74.3148	70.5821	73.1495	67.0911
family care	36.4	44.9	51.1145	44.5823
ft student	18.8545	35.6	19.4226	34.3333
long term sick/disabled	50.25	53.8462	52.3276	50.5208
on matern leave			20	29.8333
govt trng scheme	21.1538	37	29	
something else	27.1429	43.25	54.3	41.3077

Some of the cells are blank, such as males who were on maternity leave, as there were no observations. You should note that the variable with the most categories should be specified first – in this case *jbstat* – so that the table is easier to read. You can use nominal and ordinal variables for these tables (just the one variable you want the mean value reported for must be interval), but the more categories you have, the more complicated your table is going to be. This is why we have displayed the simpler *married* variable rather than the original variable *mastat*.

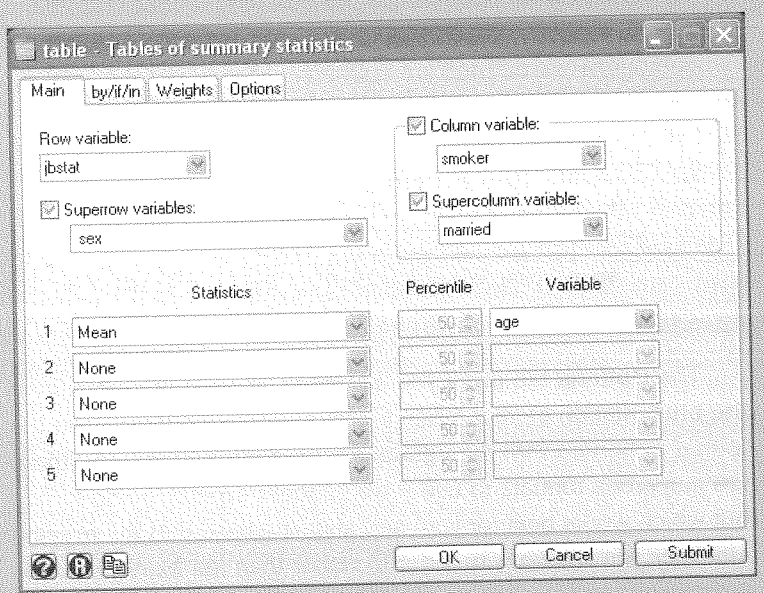
The final message before moving on to correlations is that there are a lot of different ways to make two- and three-way tables in Stata. You will find the command that you like the best, after practising with the ones we have shown you. There are special options available in the different commands, but the basics are pretty much the same. You just have to figure out the commands that are most suited for the types of results you are interested in presenting.

Box 6.5: Three-way and higher-way tables using pull-down menus

The pull-down menu equivalent of the **`table`** command is found at:

Statistics → Summaries, tables, and tests → Tables → Table of summary statistics (table)

This brings you to the **Main** tab in the **table - Tables of summary statistics** dialogue box. Here you specify your row and column variables; in this example we have used the variables *jbstat* and *smoker*, respectively. If you wish you can also specify **Superrow** and **Supercolumn** variables; for illustration of how complex the tables can be we have chosen *sex* and *married*. The cell contents are specified by the numbered rows of boxes; we have chosen the mean of *age*, but up to five statistics could be specified from different variables if needed.

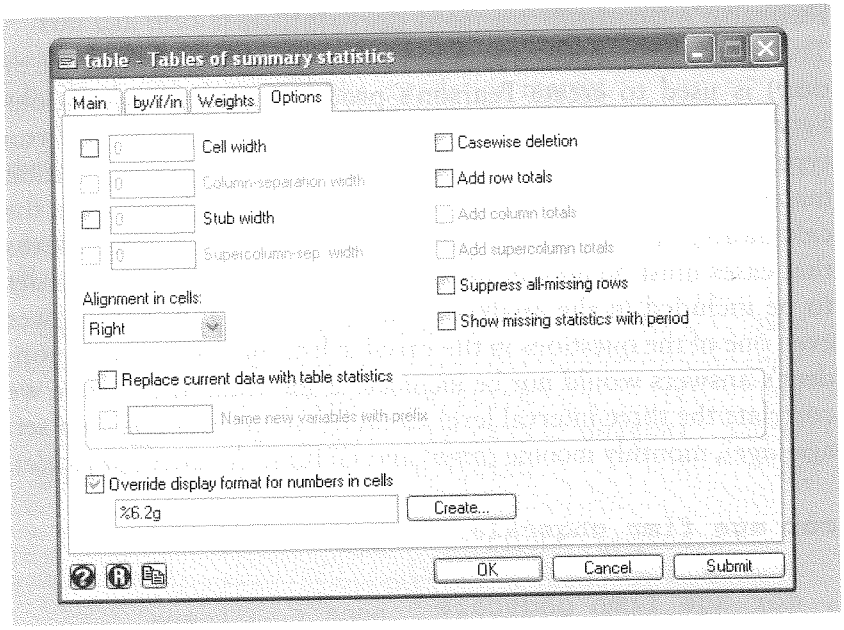


This will produce the following table – quite a bit of information! And you could further combine this with **`bysort`** or **`if`** in the by/if/in tab. ▶

▶

sex and labour force status	married indicator and smoker			
	- not marr -		-- married --	
	yes	no	yes	no
male				
self employed	34.85	40.06	41.87	46.16
in paid employ	29.99	29.93	41.80	42.43
unemployed	29.95	31.99	39.12	46.49
retired	71.81	75.26	68.57	71.07
family care	37.25	33.00	49.00	40.80
ft student	19.05	18.81	.	35.60
long term sick/disabled	50.32	50.16	52.64	54.84
on matern leave	.	.	.	.
govt trng scheme	24.75	19.56	32.00	39.00
something else	26.25	28.33	36.00	45.67
female				
self employed	38.50	40.82	42.24	43.90
in paid employ	32.65	32.01	41.16	41.08
unemployed	29.04	29.89	38.50	36.43
retired	69.86	74.00	66.19	67.26
family care	36.04	60.57	42.11	45.52
ft student	19.77	19.34	36.00	34.18
long term sick/disabled	50.21	54.45	49.11	51.37
on matern leave	.	20.00	30.00	29.80
govt trng scheme	.	29.00	.	.
something else	39.33	60.71	41.67	41.20

The **Options** tab gives you a number of ways of customizing the presentation of your table. In this example we have used the format option so that the figures are presented concisely (there are numerous formats to use – we suggest experimenting to find out which works best for you) and also specified that the missing statistics are shown with a dot (.) rather than left blank, which is the default.



CORRELATIONS

There are two types of correlations that are widely used in the social and behavioural sciences – Pearson's product-moment correlation and the non-parametric varieties which include Spearman's rank correlation coefficient and Kendall's rank-order correlation coefficient. The most important thing to remember is that all statistics have their own set of assumptions – that is, the conditions under which they work properly. Pearson's product-moment correlation measures the linear association between two variables and both variables must be measured at the interval level. Like most non-parametric statistics, the big difference between this type of estimate and its parametric cousin is that non-parametric statistics are calculated based on ranks while parametric statistics focus on mean values. As we know, means are 'meaningless' unless we have variables measured at the interval level. Thus, we typically use Spearman's or Kendall's calculation when we have ordinal variables.

Pearson's product-moment correlation

The command **correlate** (which can be shortened to **corr** or **cor**) is used to create Pearson's product-moment correlations between variables (usually just referred to as Pearson's *r*). If you type **cor** followed by a list of variables you will get a correlation matrix for all those variables. Observations are excluded from the calculation due to missing values on a listwise basis. This means that cases must be present on all items in the variable list in order to be included in the analysis. If a respondent failed to answer even one of the questions in the variable list, the rest of the respondent's answers would not be included in the correlation. Here we correlate the three interval level variables in our example data set: age (*age*), monthly income (*fimn*) and GHQ scale score (*ghqscale*).

```
cor age fimn ghqscale
```

```
. cor age fimn ghqscale
(obs=9613)

             |      age      fimn   ghqscale
-------------+---------------------------
         age |   1.0000
        fimn |  -0.0890    1.0000
         ghq |   0.0511   -0.0892     1.0000
```

Across the diagonal are the variables correlated with themselves, which is always a perfect correlation (i.e. 1.00). Pearson's correlation values range from –1.0 to 1.0, with values closer to +1 or –1 indicating a stronger association. A correlation of 0 means there is no linear association. A positive association means there is a tendency for the values of one variable to increase as the values of the other variables increase. This also holds true if the values decrease – as one decreases, the other decreases as well. A negative relationship means that as one variable increases, the other decreases, and vice versa (i.e. as one variable decreases, the other increases). Remember, correlations do not imply causation; that they simply tell you the extent to which variables tend to increase or decrease together. This relationship does not tell you that one variable causes the other to change – only that there is some association between their values. In terms of interpretation, values below 0.30 suggest there is little association between the variables (see Hinkle et al. 1988).

You can also specify that summary statistics of the variables (means, standard deviations, minimum and maximum values) are to be displayed along with the correlation matrix.

cor age fimn ghqscale, means

```
. cor age fimn ghqscale, means
(obs=9613)
   Variable |       Mean   Std. Dev.       Min     Max
------------+--------------------------------------------
        age |   44.20243   18.20314         16      97
       fimn |   744.1302   743.5433  .0045041   11297
   ghqscale |   10.77125   4.914182          0      36

             |      age      fimn   ghqscale
-------------+------------------------------
         age |   1.0000
        fimn |  -0.0890    1.0000
   ghqscale |   0.0511   -0.0892     1.0000
```

If you want pairwise correlations – that is correlations between all possible cases within the data, even if they are missing on some of the variables in the variable list – then you can use the command **pwcorr**. This command displays all the pairwise correlation coefficients between the variables in the variable list or, if a variable list is not specified, between all the variables in the data set.

pwcorr age fimn ghqscale

```
. pwcorr age fimn ghqscale
             |      age      fimn   ghqscale
-------------+------------------------------
         age |   1.0000
        fimn |  -0.0969    1.0000
   ghqscale |   0.0511   -0.0892     1.0000
```

A pairwise correlation restricts the correlation to those cases which have non-missing values for the two variables under consideration. This should become clearer when you add the option **obs** below. You can see that the results here are slightly different

than the ones achieved with the **corr** command. Using the **obs** option, we can better understand these slight discrepancies.

```
pwcorr age fimn ghqscale, obs
```

```
. pwcorr age fimn ghqscale, obs
             |      age      fimn   ghqscale
-------------+---------------------------
         age |   1.0000
             |    10264
             |
        fimn |  -0.0969    1.0000
             |    9912      9912
             |
   ghqscale |   0.0511   -0.0892    1.0000
             |    9613      9613      9613
```

You can see here how pairwise correlations allow the sample sizes to differ. For example, we have 9613 cases in the correlation between *age* and *ghqscale*, compared to 9912 between *age* and *fimn*.

There is a variety of options that can be used with the command **pwcorr** depending upon what you want Stata to display.

- **sig** adds a line to each row of the matrix reporting the significance level of each correlation coefficient.
- **print(#)** specifies the significance level of correlation coefficients to be printed. Coefficients with larger significance levels are left blank. **print(.05)** would list only coefficients significant at the 5% level or better.
- **star(#)** specifies the significance level of coefficients to be starred. **star(.01)** would star all coefficients significant at the 1% level or better.
- **listwise** is new for version 10 and tells Stata to treat the missing values listwise as in the **corr** command. Therefore, all the **obs** values will be the same for each correlation reported.

Of course, these options can be combined, for example:

```
pwcorr age fimn ghqscale, obs print(.05) ///
   star(.01)
```

```
. pwcorr age fimn ghqscale, obs print(.05) ///
   star(.01)

             |      age      fimn   ghqscale
-------------+---------------------------------
         age |   1.0000
             |    10264
             |
        fimn |  -0.0969*   1.0000
             |     9912      9912
             |
    ghqscale |   0.0511*  -0.0892*    1.0000
             |     9613      9613       9613
```

Here you can see that the number of observations for each pairwise correlation is printed, as well as stars. We told Stata to put stars only next to coefficients that are significant at the 0.01 level and to print correlations only if significant at the 0.05 level. Since all are starred, they are all significant at the 0.01 level. You may wonder why, especially since we told you that values below 0.30 are not considered strong enough to suggest that there is any relationship, yet values of 0.05 are considered statistically significant at the 0.01 level. An important thing to remember about significance levels with Pearson's *r* is that they are highly connected to sample size. Because our sample size is almost 10,000, it is more likely that associations are statistically significant. But just because a correlation is statistically significant does not mean it is substantially significant.

One way to illustrate this point is with an example. Let us run the previous command with an additional variable, the one that is the personal identification number of the respondent (*pid*).

```
pwcorr pid age fimn ghqscale, obs print(.05) ///
   star(.01)
```

```
. pwcorr pid age fimn ghqscale, obs print(.05)
   /// star(.01)
```

	pid	age	fimn	ghqscale
pid	1.0000			
	10264			
age		1.0000		
		10264		
fimn	-0.0857*	-0.0969*	1.0000	
	9912	9912	9912	
ghqscale	0.0244	0.0511*	-0.0892*	1.0000
	9613	9613	9613	9613

A researcher would never be interested in the correlation of *pid* with other variables because the values of this variable are actually nominal – they don't imply any sort of quantity. A unique number was created for everyone and its digit amount isn't important – it is just unique and allows us to follow the individual year after year. However, the output shows that it is significantly correlated with *fimn* at the 0.01 level! It is not correlated with *age* at the 0.05 level, which is why it was not printed. It was significantly correlated with *ghqscale* at the 0.05 level, which why it was printed, but not at the 0.01 level, as it was not given a star.

If you are interested in covariances instead of correlation coefficients, you can add the option **covariance** (or **cov**) after the correlation command. Covariance is a measure of how much the mean deviations of the values of two variables match. The major difference between correlation coefficients and covariance coefficients is that correlation coefficients are a scaled version of the covariance, adjusted to be between –1 and 1.

```
cor age fimn ghqscale, cov
```

```
. cor age fimn ghqscale, cov
(obs=9613)
```

	age	fimn	ghqscale
age	331.354		
fimn	-1204.01	552857	
ghqscale	4.56681	-325.952	24.1492

Partial correlations

Before moving on, it is useful to discuss partial correlations. You may want to find out the correlation among two or more variables while controlling for the effects of a third (or more). So, for example, you might want to find out what the correlation is between age and income independently of the effects of the GHQ score.

```
pcorr age fimn ghqscale
```

```
. pcorr age fimn ghqscale
(obs=9613)

Partial correlation of age with

    Variable |      Corr.     Sig.
-------------+------------------
        fimn |   -0.0849    0.000
    ghqscale |    0.0435    0.000
```

In the results above, the partial correlations of *age* with *fimn* and *ghqscale* are given. Thus, the correlation between *age* and *fimn*, controlling for, or independent of the effects of, *ghqscale* is −0.085. Likewise, the correlation between *age* and *ghqscale*, controlling for *fimn*, is 0.044. Both are very small correlations but are statistically significant at the 0.01 level (and less). You could control for numerous interval level variables by just adding them to the variable list after the **pcorr** command.

Spearman's correlation

A non-parametric alternative to Pearson's correlation (*r*) is Spearman's correlation (*rho*). Because the four-point scales on our mental health indicators may not be 'truly' interval, it is appropriate to use a non-parametric alternative. We mean that the measure isn't 'truly' interval in the sense that the categories of response are 1 = better than usual, 2 = same as usual, 3 = less than usual, and 4 = much less than usual. We can't be certain that the distance between 1 and 2 and the distance between 3 and 4 is exactly the same. It may seem like a pedantic point, and in much real-life research ordinal variables like this (and Likert scales) are often just treated as interval level despite this violation of the assumption.

The command **spearman** followed by the variables of interest will produce the Spearman's correlation matrix.

```
spearman ghqa ghqb ghqc
```

```
. spearman ghqa ghqb ghqc
(obs=9709)

             |    ghqa    ghqb    ghqc
-------------+---------------------------
        ghqa |  1.0000
        ghqb |  0.2920  1.0000
        ghqc |  0.2500  0.1334  1.0000
```

You can also add options in a way similar to how you can in the **corr** command. For example, we can request specific statistics, such as rho (the correlation coefficient), the number of observations and the *p* value associated with the estimate. You can also get Stata to only print those that are significant at 0.05 and below and to star those that are significant at the 0.01 level or less.

```
spearman ghqa ghqb ghqc, stats(rho obs p) ///
   print(.05) star (0.01)
```

```
. spearman ghqa ghqb ghqc, stats(rho obs p) ///
   print(.05) star (0.01)

+----------------+
| Key            |
|----------------|
| rho            |
| Number of obs  |
| Sig. level     |
+----------------+

             |    ghqa      ghqb      ghqc
-------------+---------------------------
        ghqa |  1.0000
             |    9709
             |
             |
        ghqb |  0.2920*   1.0000
             |    9709      9709
             |  0.0000
             |
        ghqc |  0.2500*   0.1334*   1.0000
             |    9709      9709      9709
             |  0.0000    0.0000
```

Pairwise correlations are also possible with the option **pw**.

```
spearman ghqa ghqb ghqc, stats(rho obs p) ///
   print(.05) star (0.01) pw
```

```
. spearman ghqa ghqb ghqc, stats(rho obs p) ///
   print(.05) star (0.01) pw

+----------------+
| Key            |
|----------------|
| rho            |
| Number of obs  |
| Sig. level     |
+----------------+

             |    ghqa      ghqb      ghqc
-------------+------------------------------
        ghqa |  1.0000
             |    9728
             |
             |
        ghqb |  0.2922*  1.0000
             |    9722     9728
             |  0.0000
             |
        ghqc |  0.2497*  0.1337*  1.0000
             |    9714     9714     9719
             |  0.0000   0.0000
```

You can see from the matrix that the request for pairwise correlations indicates that there are somewhat different numbers of respondents for each pair of correlations.

To obtain correlations by using pull-down menus, see Box 6.6.

Kendall's tau correlations

Kendall's tau and Spearman's rho are similar in some of their assumptions, but their interpretations are rather different. Spearman's rho is generally interpretable in the same way as Pearson's *r* (we say 'generally'; as they are computed fundamentally differently, they are obviously different statistics and not identical in their interpretation). But Kendall's tau is different. It is probably helpful to lead with an example.

Box 6.6: Correlations using pull-down menus

The pull-down menu equivalent of the **corr** command is:

Statistics → Summaries, tables, and tests → Summary and descriptive statistics → Correlations & covariances

The pull-down menu equivalent of the **pwcorr** command is:

Statistics → Summaries, tables, and tests → Summary and descriptive statistics → Pairwise correlations

The pull-down menu equivalent of the **pcorr** command is:

Statistics → Summaries, tables, and tests → Summary and descriptive statistics → Partial correlations

The pull-down menu equivalent of the **spearman** command is:

Statistics → Summaries, tables, and tests → Nonparametric tests of hypotheses → Spearman's rank correlation

The pull-down menu equivalent of the **ktau** command is:

Statistics → Summaries, tables, and tests → Nonparametric tests of hypotheses → Kendall's rank correlation

We use the command **ktau**:

```
ktau ghqa ghqb ghqc
```

```
. ktau ghqa ghqb ghqc
(obs=9709)

             |    ghqa     ghqb     ghqc
-------------+---------------------------
        ghqa |  0.3850
        ghqb |  0.1341   0.6300
        ghqc |  0.0953   0.0627   0.4179
```

Here we have a matrix of the Kendall tau correlations between all three variables. The difference with the Kendall's correlation is that while Pearson's and Spearman's correlations present results

in terms of proportion of variability accounted for, Kendall's tau measures a probability: that the observed data are in the same order versus the probability that the observed data are not in same order. Its value ranges from −1 to +1. Interpretation is not straightforward, but you can request a *p* value to determine whether or not the value is statistically significant.

Other variants of Kendall's tau are available as well. The default 'tau' is technically known as Kendall's tau-a. Kendall's tau-b is often used for 2 × 2 tables but isn't limited to them.

It should be noted that **ktau** is better suited for use in small data sets as its computation time can be considerable in larger data sets.

By way of example we can show you some of the options available in the **ktau** command.

```
ktau ghqa ghqb ghqc, stats(taua taub p) ///
   print(.05) star (0.01)
```

```
. ktau ghqa ghqb ghqc, stats(taua taub p) ///
   print(.05)star (0.01)
(obs=9709)

+-------------+
| Key         |
|-------------|
| tau_a       |
| tau_b       |
| Sig. level  |
+-------------+

             |    ghqa      ghqb      ghqc
-------------+---------------------------
        ghqa |  0.3850
             |  1.0000
             |
        ghqb |  0.1341*  0.6300
             |  0.2724*  1.0000
             |  0.0000
             |
        ghqc |  0.0953*  0.0627*  0.4179
             |  0.2376*  0.1222*  1.0000
             |  0.0000   0.0000
             |
```

We have asked Stata to report both tau-a and tau-b statistics and their significance levels, indicating that the coefficient is not printed if it is above the 0.05 level of significance and that a star is given for a significance level of were correlated at the 0.01 or less. If we were interested in the pairwise associations, we could have also added the option **pw** to the command above.

All of the correlation commands can be combined with **bysort** and **if** commands to condition the statistics reported. For example, if we want the Spearman's correlation between the three mental health items separately for men and women under 30 years old:

```
bysort sex: spearman ghqa ghqb ghqc if age<30
```

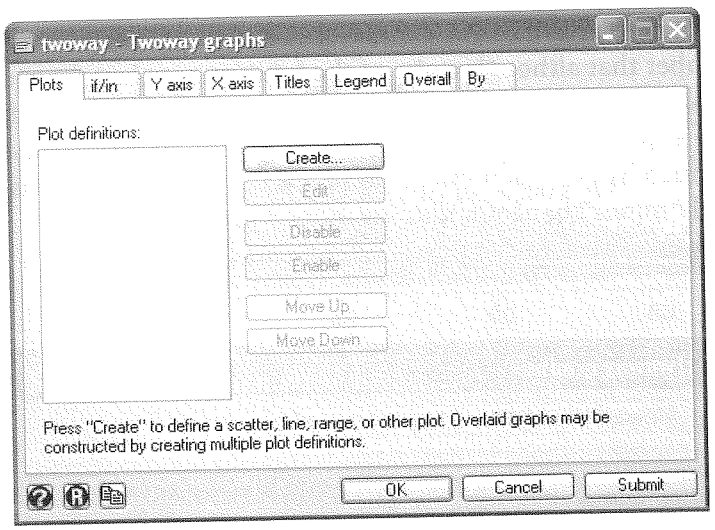
```
. bysort sex: spearman ghqa ghqb ghqc if age<30
-------------------------------------------------------------------------------
-> sex = male
(obs=1211)

             |    ghqa     ghqb     ghqc
-------------+---------------------------
        ghqa |  1.0000
        ghqb |  0.2470   1.0000
        ghqc |  0.2070   0.0786   1.0000

-------------------------------------------------------------------------------
-> sex = female
(obs=1281)

             |    ghqa     ghqb     ghqc
-------------+---------------------------
        ghqa |  1.0000
        ghqb |  0.2740   1.0000
        ghqc |  0.1954   0.0989   1.0000
```

TWO-VARIABLE GRAPHS

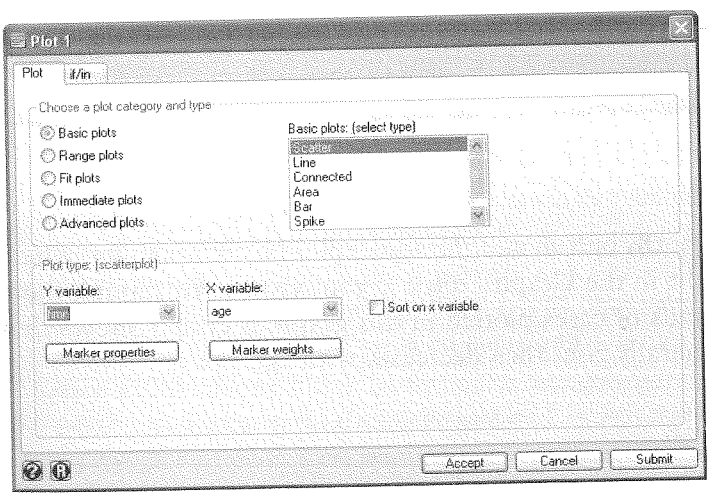
As we have done previously, we start looking at graphs by using the pull-down menus. First, we produce a scatterplot of monthly income against age by selecting:

Graphics → Twoway graph

This brings up the **twoway - Twoway graphs** dialogue box. This is for simple scatterplots and for more complicated overlaid two-way graphs (see Box 6.7). This dialogue box is completely new for version 10, so if you're using version 9 you either use the **Easy graphs** pull-down menu choices or the more detailed **Scatterplot** option.

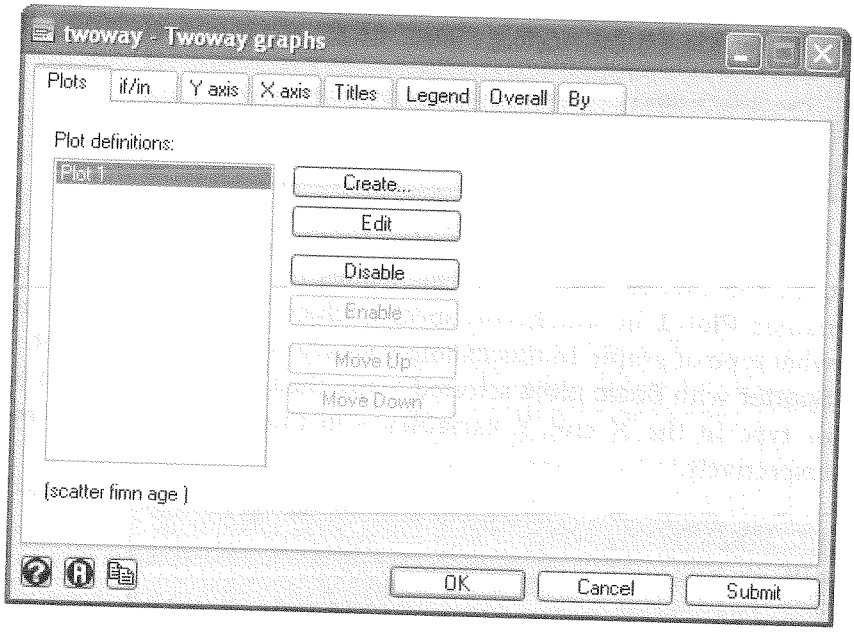


To make a scatterplot, or any other two variable graph, you first click the **Create** button and this brings up a new dialogue box named **Plot 1** in which you specify which variables to use and what type of graph. In this example we choose the type of graph – **Scatter** with **Basic plots** selected – and then either scroll down to or type in the X and Y variables – in this case *age* and *fimn*, respectively.

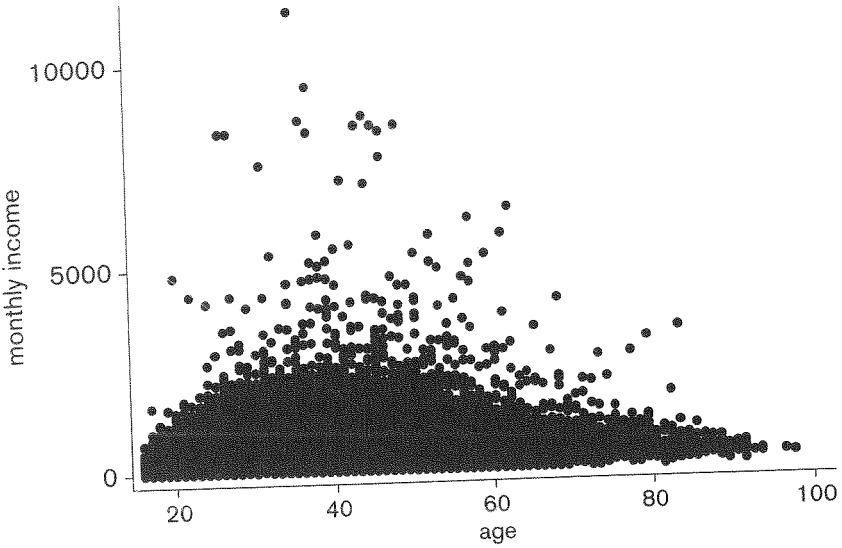


When you have finished creating the graph, click on the **Accept** button and the **Plot 1** dialogue box will close and return you to the **twoway – Twoway graphs** dialogue box. You can see that this now has Plot 1 in the **Plot definitions** box and you are able to change the plot by clicking on the **Edit** button which will bring up the **Plot 1** dialogue box again. The other tabs are fairly self-explanatory if you have read Chapter 5.

Remember that although many of the functions in the tabs can now be done in the Graph Editor once the basic graph is created, the Graph Editor does not produce a command. So we suggest doing as much as possible in the commands or pull-down menus, leaving only minor changes/additions to the Graph Editor.



Click on the **OK** button and the scatterplot below is shown, which you can edit in the Graph Editor if you wish. Even though there are thousands of cases plotted in this graph, it clearly shows that income generally rises then decreases with age.



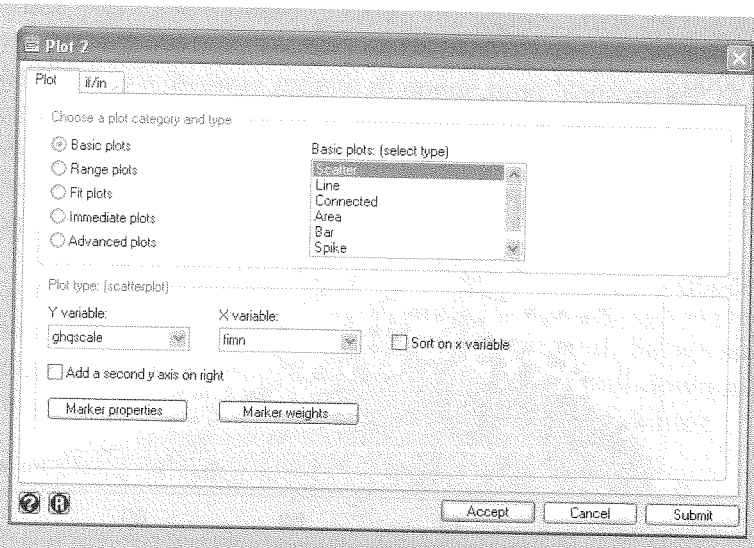
Box 6.7: Overlaid two variable graphs

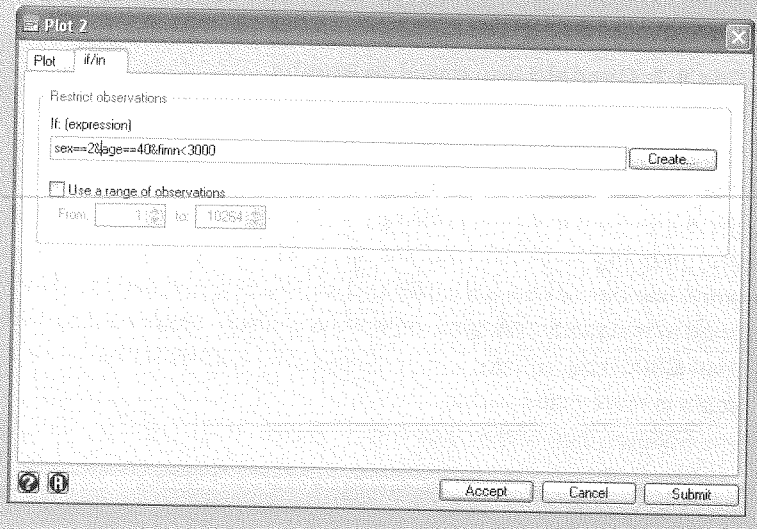
In this demonstration we wish to plot income (*fimn*) against mental health (*ghqscale*) separately for men and women but overlaid on the same axes in order to compare the distributions. We restrict the graphing to those who are aged 40 so that there will be a small number of cases plotted and so that you can see the difference more clearly. We also restrict the cases to those earning less than £3000 per month, which removes the outliers so the income scale is more manageable.

Follow the same pull-down menu path as for simple scatterplots:

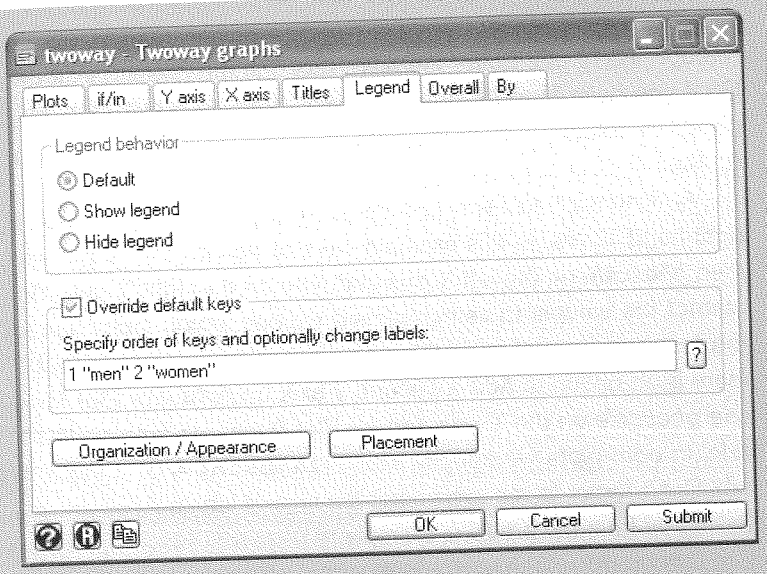
Graphics → Twoway graph

This time we need to create two new plots in the Plot definitions box. First create a plot (or edit Plot 1 if one is still listed there from previous work) that specifies a scatterplot with *fimn* on the X axis and *ghqscale* on the Y axis. For Plot 1 use the **if/in** tab to restrict this plot to men (sex==1), aged 40 (age==40), with income less than £3000 (fimn<3000). Below is the dialogue box for Plot 2 which has the same specifications as Plot 1 except that sex==2, so this plot is restricted to women. ▶

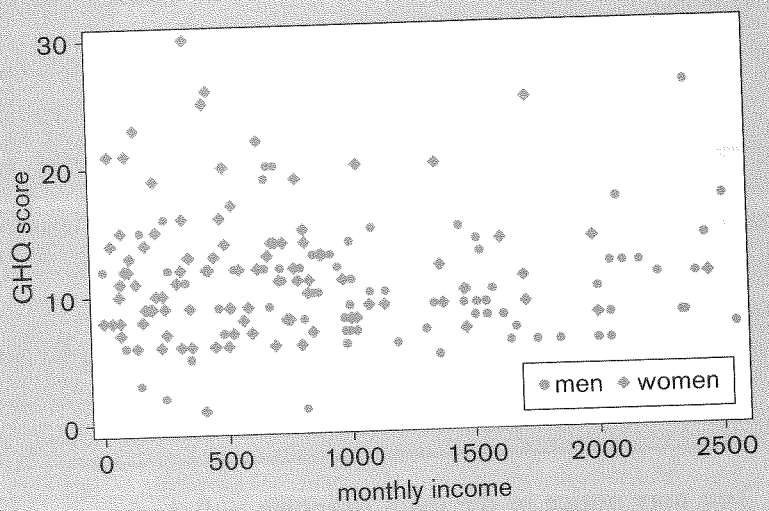




If you now produce the scatterplot you will see that the legend doesn't make any sense when it should tell you which symbols are for men and which are for women. Either you can change this in the **Legend** tab as shown below (click on the **?** button for help on what to type in the **Override default keys** box) or in the **Chart Editor**; you know that men are the first plot and therefore the first in the legend from left to right or top to bottom (depending how your legend is formatted).



This produces an overlaid scatterplot in which the circles are men and diamonds are women. We have changed the scheme to a monochrome one for better printing in black and white. To change the style of the graph use the pull-down menu **Edit** → **Apply new scheme** when the chart first appears, before opening the **Chart Editor**. Try out some of the different schemes available to see which ones are most suitable for your uses.

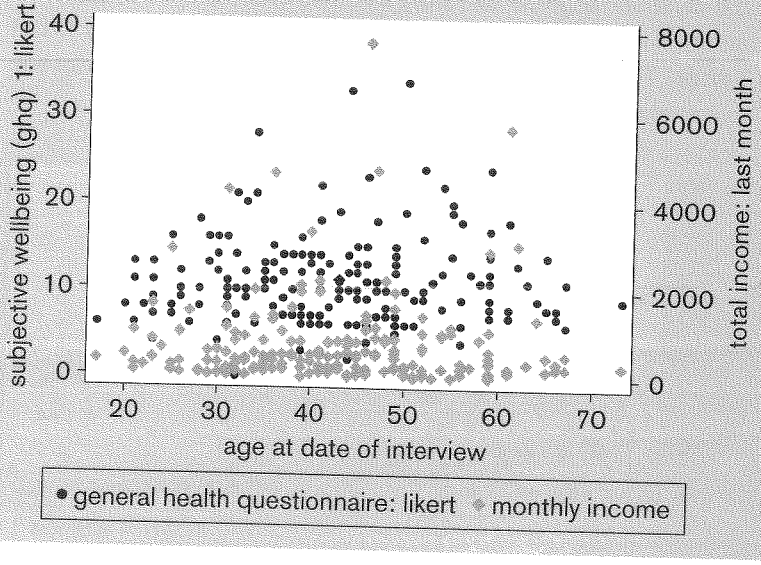


You may have noticed that Stata 'echoes' the commands in the Results window when you use the pull-down menus to ▶

▶ create graphs. But it is worth noting that any changes done through the Graph Editor are not echoed in the Results window. So if you have found a graph that you want to replicate simply copy the command from the Results window and paste it into a do file.

An extension to these graphs is to add a third variable on a right-hand Y axis. In this example we plot *age* on the X axis and then *ghqscale* on the left Y axis and *fimn* on the right Y axis. We restrict the sample to self-employed women. If you look back to the first screen capture in this box, you will notice that in Plot 2 there is an option to **Add a second y axis on right**. For Plot 1 we use *ghqscale* on the Y axis and for Plot 2 we use *fimn* on the Y axis. In both plots we specify `sex==2&jbstat==1` in the **if/in** tab. Don't forget to remove or edit the legend labels if you are following on from the previous example.

This produces the scatterplot below, where we have changed the output to monochrome for better printing in black and white. As you can imagine with the capabilities of Stata, these graphs can be very complex. Enjoy exploring the possibilities!



You may notice in the main **Graphics** pull-down menu and there is something called **Scatterplot matrix**.

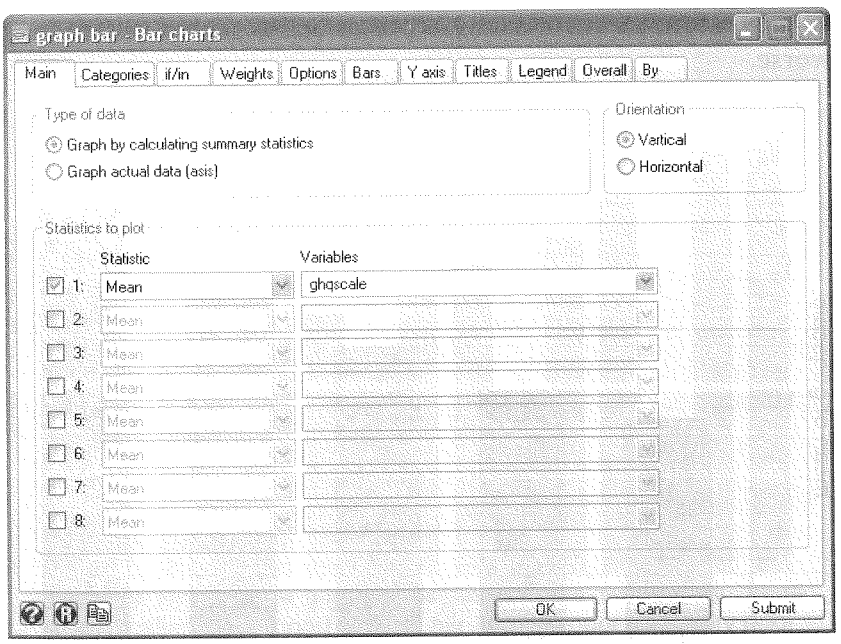
Graphics → Scatterplot matrix

If the three interval level variables (*age*, *fimn*, *ghqscale*) in the example data are entered into the box, Stata produces a matrix of scatterplots of each pair of variables. It's a quick way to inspect the association between a number of variables.

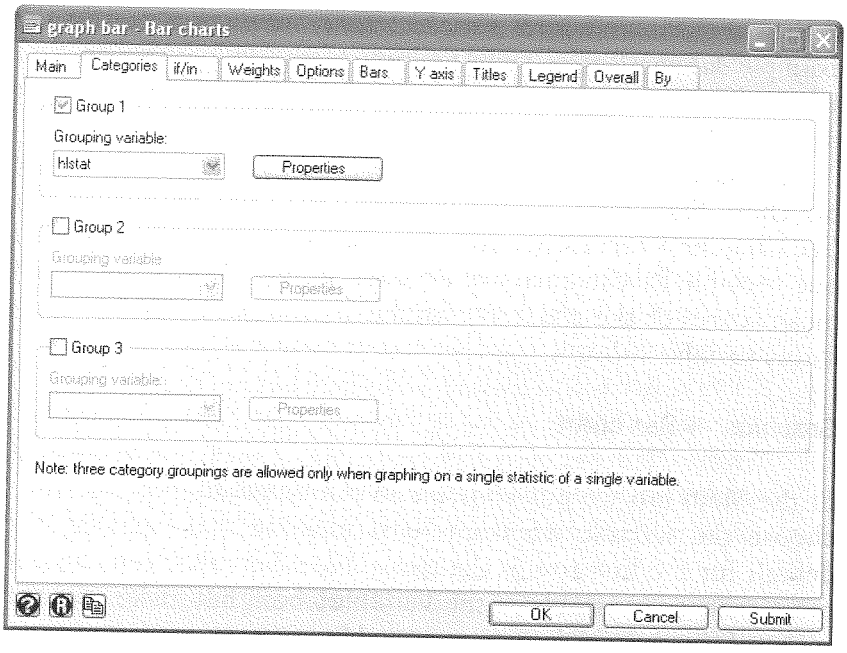
The next type of two-variable graph we introduce here is the bar chart. Stata differentiates between bar charts and histograms (see Chapter 5 for histograms). Bar charts produce a summary of one variable by categories of another. In our example we graph mean scores of the *ghqscale* variable by categories of the health status variable (*hlstat*).

Graphics → Bar chart

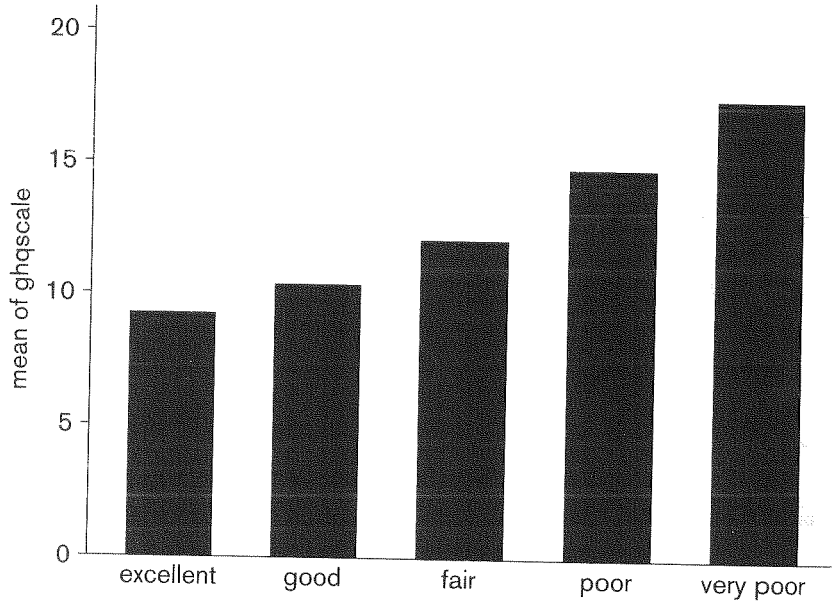
In the **Main** tab we put *ghqscale* in the **Variables** box and then choose the statistic we want to summarize that variable; the default for the first row is the mean. Note here that you can put more than one variable in the **Variables** box, but make sure the scales of those variables are comparable.



The categorical variable that will form the X axis is entered in Group 1 box of the **Categories** tab, in this example *hlstat*.

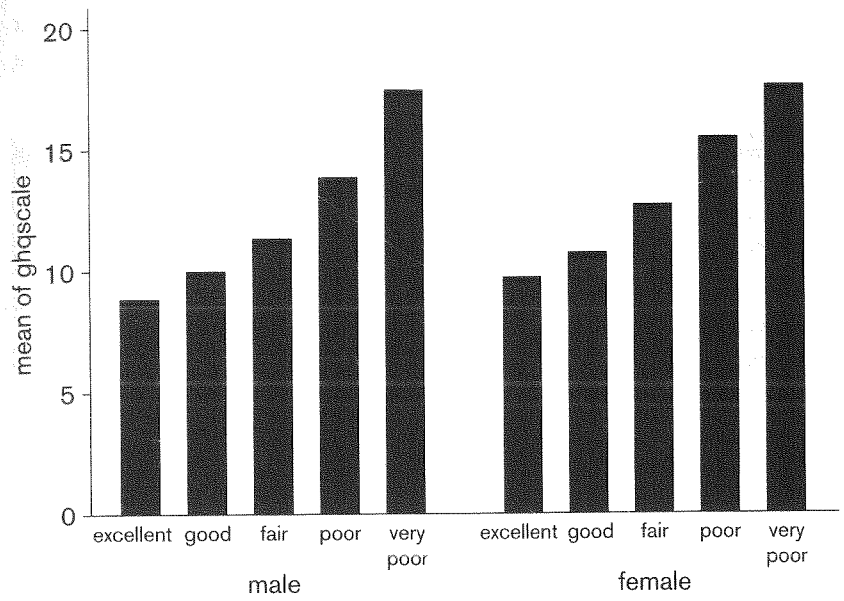


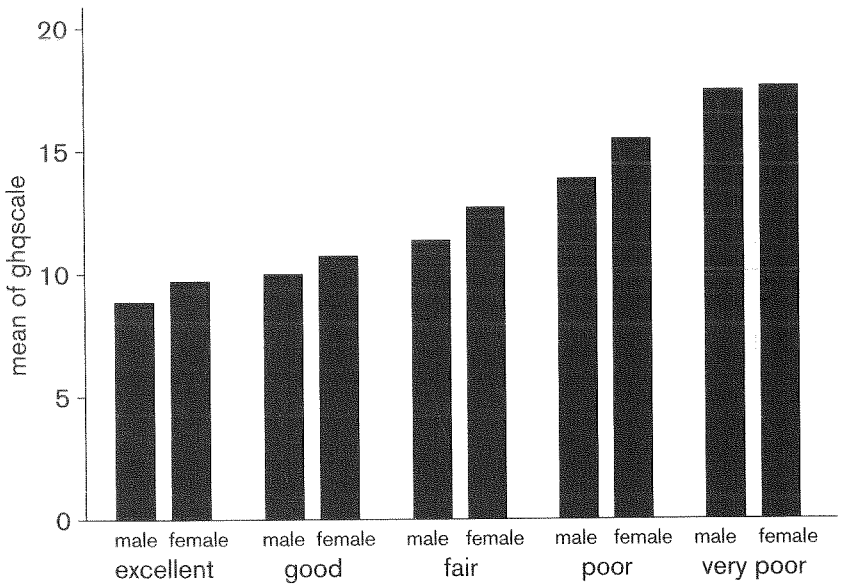
This produces the following bar chart which shows quite clearly an increasing mean of GHQ score as self-reported health gets worse.



In the **Categories** tab you can see that there is space for other 'grouping' variables. Depending on how you enter the variables in the Group 1 and Group 2 boxes, different chart formats will

result, as can be seen the two charts below. In the top chart Group 1 is *hlstat* and Group 2 is *sex*. In the bottom chart Group 1 is *sex* and Group 2 is *hlstat*.

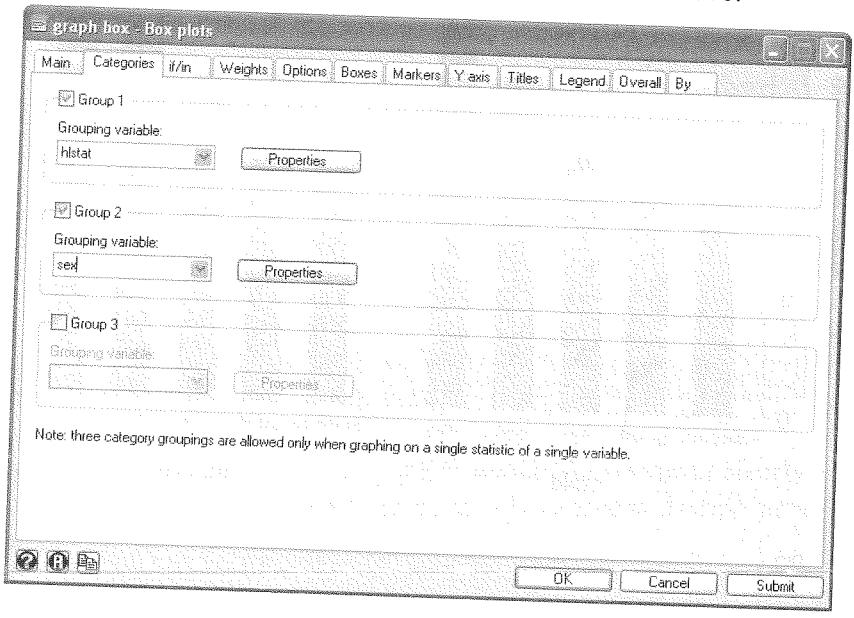




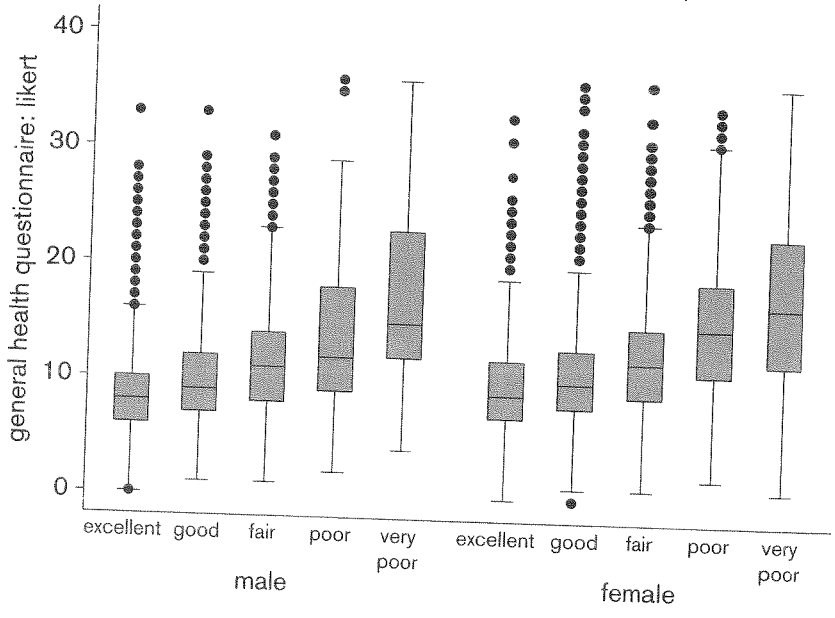
The third type of two-variable graph we cover in this chapter is grouped box plots. If you revise the creation of box plots in Chapter 5 you will remember that there is a **Categories** tab in the **graph box – Box plots** dialogue box that opens when selecting:

Graphics → Box plot

So after putting the variable you wish to summarize in the box plots in the **Main** tab, put the categorical variable in the **Categories** tab. You can have up to three grouping variables, but in this example we have used two: *hlstat* and *sex*. These grouping variables operate in the same way as in bar charts above.



The resulting box plot now plots GHQ score for each category of health status and for both men and women separately.



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 interval variables, their normality.

At this stage of this demonstration we start exploring the associations between our outcome variable (*ghqscale*) and the factors believed to affect mental health. As the *ghqscale* variable is interval level, we can look for differences in mean scores across categories in the other variables: *female*, *agecat*, *marst2*, *empstat*, *numchd* and *region2*. This can be done with a single command using **tab1** and the **su** option:

```
tab1 female agecat marst2 empstat numchd ///
   region2, su(ghq)
```

Note that we have used *ghq* instead of the full name of the variable *ghqscale*. Stata allows you to shorten variable names provided that the shortening only identifies one variable. In other words, if we had two variables that started with the letters *ghq* then Stata would return an error.

```
. tab1 female agecat marst2 empstat numchd ///
   region2, su(ghq)

-> tabulation of female

    female |          Summary of ghq 0-36
 indicator |        Mean    Std. Dev.     Freq.
-----------+------------------------------------
      male |   10.197257    4.7327355      3645
    female |   11.315802    5.0705736      4069
-----------+------------------------------------
     Total |    10.78727    4.9451537      7714

-> tabulation of agecat

       age |          Summary of ghq 0-36
categories |        Mean    Std. Dev.     Freq.
-----------+------------------------------------
 18-32 yea |   10.563872    4.8776024      2779
 33-50 yea |    11.03622    4.9563576      3175
 51-65 yea |   10.690909    5.0129936      1760
-----------+------------------------------------
     Total |    10.78727    4.9451537      7714
```

```
-> tabulation of marst2

   marital |
  status 4 |          Summary of ghq 0-36
categories |      Mean   Std. Dev.      Freq.
-----------+------------------------------------
    single |  10.345222   5.010017       1486
   married |  10.677812  4.7087427       5503
   sep/div |  12.638182  6.2309817        550
   widowed |  12.165714   5.603875        175
-----------+------------------------------------
     Total |   10.78727  4.9451537       7714

-> tabulation of empstat

employment |          Summary of ghq 0-36
    status |      Mean   Std. Dev.      Freq.
-----------+------------------------------------
  employed |  10.254293  4.4085201       5474
unemploye  |  12.934959   6.091281        492
  longterm |  15.365957  6.8533252        235
  studying |  10.375566  5.0631593        221
family ca  |  12.119183  5.6297188        881
   retired |  10.002597  4.5492094        385
-----------+------------------------------------
     Total |  10.786681  4.9406805       7688

-> tabulation of numchd

children 3 |          Summary of ghq 0-36
categories |      Mean   Std. Dev.      Freq.
-----------+------------------------------------
      none |  10.555601   4.926495       4874
one or tw  |  11.174229  4.9786735       2336
 three or  |  11.234127  4.8349237        504
-----------+------------------------------------
     Total |   10.78727  4.9451537       7714

-> tabulation of region2

 regions 7 |          Summary of ghq 0-36
categories |      Mean   Std. Dev.      Freq.
-----------+------------------------------------
    London  |  10.911873  5.1982339        817
     South  |  10.615874   4.649652       2356
  Midlands  |  10.797145  5.0592898       1331
Northwest   |  10.753086  4.8014493        810
North and   |  10.825739  5.0068762       1251
     Wales  |  11.605528  5.6338908        398
  Scotland  |  10.711052  4.9890945        751
------------+------------------------------------
     Total  |   10.78727  4.9451537       7714
```

If we didn't want to see the standard deviation and frequencies then we could add the **means** option so that Stata only produces the mean *ghqscale* value for each category:

```
tab1 female agecat marst2 empstat numchd ///
   region2,su(ghq) means
```

```
. tab1 female agecat marst2 empstat numchd ///
   region2,su(ghq) means

-> tabulation of female

           | Summary of
    female |  ghq 0-36
 indicator |       Mean
-----------+------------
      male |   10.197257
    female |   11.315802
-----------+------------
     Total |    10.78727

-> tabulation of agecat

           | Summary of
       age |  ghq 0-36
categories |       Mean
-----------+------------
 18-32 yea |   10.563872
 33-50 yea |    11.03622
 51-65 yea |   10.690909
-----------+------------
     Total |    10.78727

Etc.
```

However, we would suggest that it is useful to see the number of cases in each category so you can see if any variations in mean values are based on categories with a small number of cases.

As you can see from the first output there are some potential differences in mean GHQ scores between males and females, across categories of marital status, employment status, and number of children in the household. The variations across age categories look small, as do the variations across regions. We will formally test these differences in the next chapter.

In addition to using age categories, we have the interval level measure of age in the data set. In this case we can correlate the two interval level variables: *ghqscale* and *age*.

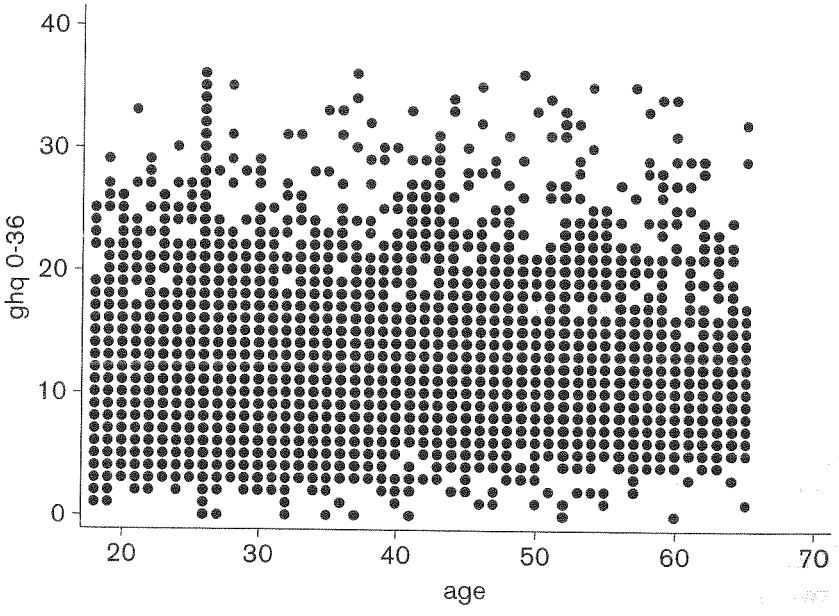
pwcorr ghq age, sig

```
. pwcorr ghqscale age, sig
             |   ghqscale      age
-------------+------------------
   ghqscale |      1.0000
            |
            |
         age |      0.0123   1.0000
            |      0.2800
            |
```

We have used the **pwcorr** command so that we can use the **sig** option which will display the *p* value in the output. Pearson's *r* is 0.012 and the *p* value is 0.28, which indicates that there is a very small and statistically non-significant linear association between these two variables. As Pearson's *r* relates to a linear association there is the possibility that this low correlation disguises a non-linear association. The means tables indicated that the mean GHQ score for the middle age group was higher than the other two. To further investigate if a non-linear association exists we first graph the two variables.

Graphics → Twoway graph

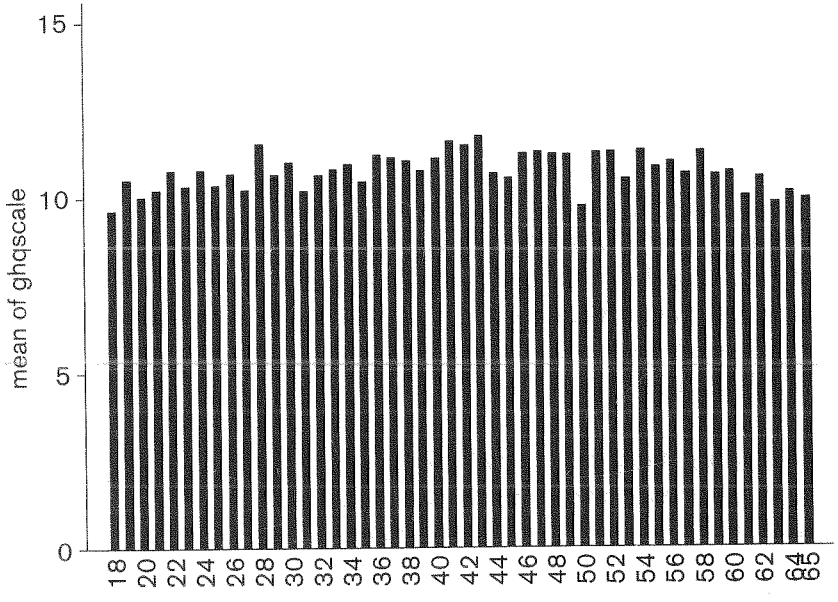
We put *age* as the X variable and *ghqscale* as the Y variable and Stata produces this scatterplot. As there are over 7000 cases it is difficult to distinguish any pattern.



An alternative approach is to graph the mean GHQ values for each age to see the pattern. This can be done in a number of ways but in this exercise we use the bar chart.

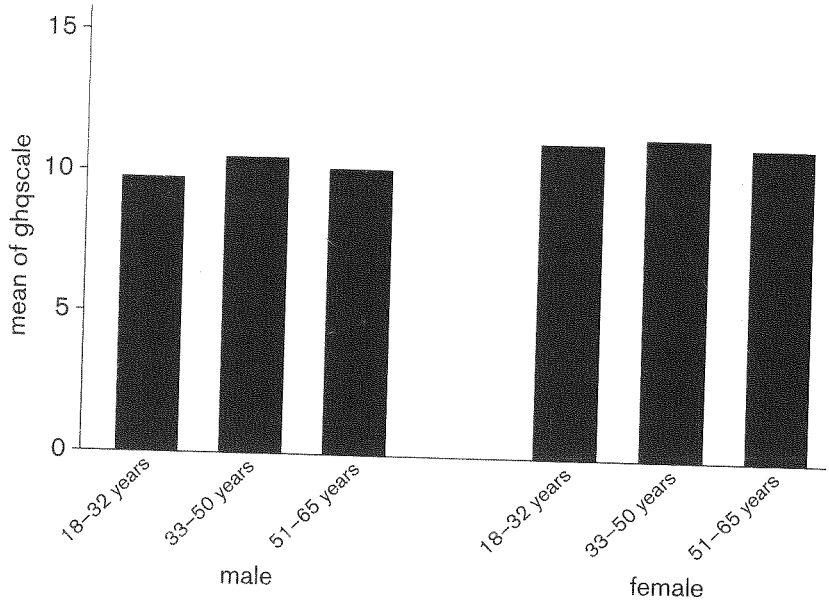
Graphics → Bar charts

This brings up the bar chart dialogue box where the default statistic is the mean which we want to use. In the **Variables** tab we either type or scroll down and select the *ghqscale* variable, and in the **Categories** tab we choose *age* as the Group 1 variable. This produces:



This bar chart is not exactly neat, but it serves the purpose of letting us examine the mean GHQ score at each age. It appears to generally rise as age increases then fall slightly. We test whether a non-linear association captures this information better than a linear one in Chapter 8.

Returning to the point of having two 'Grouping' variables, we can use that option to inspect the age distribution of mean GHQ scores for both men and women. This time we use the age categories rather than age in years as the Group 1 variable and *female* as the Group 2 variable. We also orientate the labels of the age categories so that they don't run into each other and thus make the graph more readable.



Next we use the dichotomous GHQ indicator variable (*d_ghq*) in crosstabs with the nominal or ordinal variables and test whether there is a statistical association. In this example we start off by using the most applicable test statistic – chi-squared – by using a series of **tab** commands. We use the **nofreq** option as we just wish to see the statistics before we go any further in our investigation.

```
ta d_ghq female, chi2 nofreq
ta d_ghq agecat, chi2 nofreq
ta d_ghq marst2, chi2 nofreq
ta d_ghq empstat, chi2 nofreq
ta d_ghq numchd, chi2 nofreq
ta d_ghq region2, chi2 nofreq
```

```
. ta d_ghq female, chi2 nofreq
          Pearson chi2(1) = 25.7939   Pr = 0.000

. ta d_ghq agecat, chi2 nofreq
          Pearson chi2(2) = 6.4265    Pr = 0.040

. ta d_ghq marst2, chi2 nofreq
          Pearson chi2(3) = 53.3636   Pr = 0.000

. ta d_ghq empstat, chi2 nofreq
          Pearson chi2(5) = 284.0623  Pr = 0.000
```

```
. ta d_ghq numchd, chi2 nofreq
          Pearson chi2(2) = 10.8155  Pr = 0.004

. ta d_ghq region2, chi2 nofreq
          Pearson chi2(6) = 17.3147  Pr = 0.008
```

As you can see from this output, there are significant associations ($p < 0.05$) between the independent variables and the dichotomous GHQ variable. The chi-squared test only tells you that there are different distributions across the categories, but it doesn't tell us where these variations are. We test for these in the next chapter.

At this stage we want to examine some of these variations in more detail to help inform our future analyses. In this demonstration we examine the association between age categories and the dichotomous GHQ indicator. First we produce a full crosstabulation with percentages:

`ta agecat d_ghq, row chi2`

```
. ta agecat d_ghq, row chi2

+----------------+
| Key            |
|----------------|
| frequency      |
| row percentage |
+----------------+

        age |       d_ghq
 categories |         0         1 |     Total
------------+----------------------+---------
18-32 years |     2,233       546 |     2,779
            |     80.35     19.65 |    100.00
------------+----------------------+---------
33-50 years |     2,572       603 |     3,175
            |     81.01     18.99 |    100.00
------------+----------------------+---------
51-65 years |     1,466       294 |     1,760
            |     83.30     16.70 |    100.00
------------+----------------------+---------
      Total |     6,271     1,443 |     7,714
            |     81.29     18.71 |    100.00

      Pearson chi2(2) = 6.4265  Pr = 0.040
```

This shows that the percentage of those over the GHQ threshold decreases with age – from 19.6% in the youngest category to 16.7% in the oldest category. If we look back to the bar chart of mean *ghqscale* values by age and gender above, as well as take into account the significant differences between males and females in the dichotomous GHQ indicator, it might be useful to break out this table by gender using the **bysort** command:

```
bysort female:ta agecat d_ghq, row chi2
```

```
. bysort female:ta agecat d_ghq, row chi2
-------------------------------------------------------------------------------
-> female = male

+----------------+
| Key            |
|----------------|
| frequency      |
| row percentage |
+----------------+

        age |       d_ghq
 categories |         0         1 |     Total
------------+---------------------+---------
18-32 years |     1,104       203 |     1,307
            |     84.47     15.53 |    100.00
------------+---------------------+---------
33-50 years |     1,235       263 |     1,498
            |     82.44     17.56 |    100.00
------------+---------------------+---------
51-65 years |       711       129 |       840
            |     84.64     15.36 |    100.00
------------+---------------------+---------
      Total |     3,050       595 |     3,645
            |     83.68     16.32 |    100.00

      Pearson chi2(2) =  2.8421   Pr = 0.241
```

```
-------------------------------------------------
-> female = female

+----------------+
| Key            |
|----------------|
| frequency      |
| row percentage |
+----------------+

        age |       d_ghq
 categories |         0         1 |     Total
------------+----------------------+---------
18-32 years |     1,129       343 |     1,472
            |     76.70     23.30 |    100.00
------------+----------------------+---------
33-50 years |     1,337       340 |     1,677
            |     79.73     20.27 |    100.00
------------+----------------------+---------
51-65 years |       755       165 |       920
            |     82.07     17.93 |    100.00
------------+----------------------+---------
      Total |     3,221       848 |     4,069
            |     79.16     20.84 |    100.00

      Pearson chi2(2) =  10.4390   Pr = 0.005
```

This output shows that the original association was largely driven by the differences in age categories and GHQ indicator for females. The chi-squared test shows that for males there is no significant association ($p = 0.241$), whereas for females there is significant association ($p = 0.005$). An inspection of the percentages reported in the crosstabulations confirms this with only small differences for males, while females show considerably more reduction in the percentage over the GHQ threshold – from 23.3% in the youngest age category to 17.9% in the oldest age category.