Using Stata 9 & Higher for OLS Regression

Richard Williams, University of Notre Dame, https://www3.nd.edu/~rwilliam/
Last revised January 3, 2022

Introduction. This handout shows you how Stata can be used for OLS regression. It assumes knowledge of the statistical concepts that are presented. Several other Stata commands (e.g. `logit, ologit`) often have the same general format and many of the same options.

Rather than specify all options at once, like you do in SPSS, in Stata you often give a series of commands. In some ways this is more tedious, but it also gives you flexibility in that you don't have to rerun the entire analysis if you think of something else you want. As the Stata 9 User's Guide says (p. 43) "The user-interface model is type a little, get a little, etc. so that the user is always in control."

For the most part, I find that either Stata or SPSS can give me the results I want. Overall though, I feel that Stata's statistical commands are generally far more logical and consistent (and sometimes more powerful) than their SPSS counterparts. Luckily, with the separate Stat Transfer program, it is very easy to convert SPSS files to Stata and vice-versa.

Get the data. First, open the previously saved data set. (Stata, of course, also has means for entering, editing and otherwise managing data.) You can give the directory and file name, or even access a file that is on the web. For example,

```
. use https://www3.nd.edu/~rwilliam/statafiles/reg01.dta, clear
```

Descriptive statistics. There are various ways to get descriptive statistics in Stata. Since you are using different commands, you want to be careful that you are analyzing the same data throughout, e.g. missing data could change the cases that get analyzed. The `correlate` command below uses listwise deletion of missing data, which is the same as what the `regress` command does, i.e. a case is deleted if it is missing data on any of the variables in the analysis.

```
. correlate income educ jobexp race, means

(obs=20)

    Variable |        Mean    Std. Dev.          Min          Max
-------------+----------------------------------------------------
      income |      24.415     9.788354            5         48.3
        educ |       12.05     4.477723            2           21
      jobexp |       12.65     5.460625            1           21
        race |          .5     .5129892            0            1

             |   income      educ    jobexp      race
-------------+------------------------------------------
      income |   1.0000
        educ |   0.8457    1.0000
      jobexp |   0.2677   -0.1069    1.0000
        race |  -0.5676   -0.7447    0.2161    1.0000
```

Regression. Use the `regress` command for OLS regression (you can abbreviate it as `reg`). Specify the DV first followed by the IVs. By default, Stata will report the unstandardized (metric) coefficients.

```
. regress income educ jobexp race

      Source |       SS       df       MS              Number of obs =      20
-------------+------------------------------           F(  3,    16) =   29.16
       Model | 1538.92019      3  512.973396           Prob > F      =  0.0000
    Residual | 281.505287     16  17.5940804           R-squared     =  0.8454
-------------+------------------------------           Adj R-squared =  0.8164
       Total | 1820.42548     19  95.8118671           Root MSE      =  4.1945


------------------------------------------------------------------------------
      income |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        educ |   1.981124   .3231024     6.13   0.000     1.296178    2.66607
      jobexp |   .6419622   .1811106     3.54   0.003     .2580248     1.0259
        race |   .5707931   2.871949     0.20   0.845    -5.517466   6.659052
       _cons |  -7.863763   5.369166    -1.46   0.162    -19.24589   3.518362
------------------------------------------------------------------------------
```

Confidence Interval. If you want to change the confidence interval, use the `level` parameter:

```
. regress income educ jobexp race, level(99)

      Source |       SS       df       MS              Number of obs =      20
-------------+------------------------------           F(  3,    16) =   29.16
       Model | 1538.92019      3  512.973396           Prob > F      =  0.0000
    Residual | 281.505287     16  17.5940804           R-squared     =  0.8454
-------------+------------------------------           Adj R-squared =  0.8164
       Total | 1820.42548     19  95.8118671           Root MSE      =  4.1945


------------------------------------------------------------------------------
      income |      Coef.   Std. Err.      t    P>|t|     [99% Conf. Interval]
-------------+----------------------------------------------------------------
        educ |   1.981124   .3231024     6.13   0.000     1.037413    2.924835
      jobexp |   .6419622   .1811106     3.54   0.003     .1129776   1.170947
        race |   .5707931   2.871949     0.20   0.845    -7.817542   8.959128
       _cons |  -7.863763   5.369166    -1.46   0.162    -23.54593     7.8184
------------------------------------------------------------------------------
```

As an alternative, you could use the `set level` command before `regress`:

```
. set level 99
. regress income educ jobexp race
```

Standardized coefficients. To get the standardized coefficients, add the beta parameter:

```
. regress income educ jobexp race, beta

      Source |       SS       df       MS              Number of obs =      20
-------------+------------------------------           F(  3,    16) =   29.16
       Model | 1538.92019      3  512.973396           Prob > F      =  0.0000
    Residual | 281.505287     16  17.5940804           R-squared     =  0.8454
-------------+------------------------------           Adj R-squared =  0.8164
       Total | 1820.42548     19  95.8118671           Root MSE      =  4.1945


------------------------------------------------------------------------------
      income |      Coef.   Std. Err.      t    P>|t|                     Beta
-------------+----------------------------------------------------------------
        educ |   1.981124   .3231024     6.13   0.000                 .9062733
      jobexp |   .6419622   .1811106     3.54   0.003                 .3581312
        race |   .5707931   2.871949     0.20   0.845                 .0299142
       _cons |  -7.863763   5.369166    -1.46   0.162                        .
------------------------------------------------------------------------------
```

NOTE: The `listcoef` command from Long and Freese's `spost13` package of routines (type `findit spost13_ado` from within Stata) provides alternative ways of standardizing coefficients.

Incidentally, you do not have to repeat the entire command when you change a parameter (indeed, if the data set is large, you don't want to repeat the entire command, because then Stata will redo all the calculations.) The last three regressions could have been executed via the commands

```
. regress income educ jobexp race
. regress, level(99)
. regress, beta
```

Also, if you just type `regress` Stata will "replay" (print out again) your earlier results.

VIF & Tolerances. Use the `vif` command to get the variance inflation factors (VIFs) and the tolerances (1/VIF). `vif` is one of many post-estimation commands. You run it AFTER running a regression. It uses information Stata has stored internally.

```
. vif

    Variable |       VIF       1/VIF
-------------+----------------------
        race |      2.34    0.426622
        educ |      2.26    0.442403
      jobexp |      1.06    0.946761
-------------+----------------------
    Mean VIF |      1.89
```

NOTE: `vif` only works after regress, which is unfortunate because the information it offers can be useful with many other commands, e.g. `logit`. Phil Ender's `collin` command (which needs to be downloaded from SSC) gives more information and can be used with estimation commands besides `regress`, e.g.

```
. collin educ jobexp race if !missing(income)
```

Hypothesis testing. Stata has some very nice hypothesis testing procedures; indeed I think it has some big advantages over SPSS here. Again, these are post-estimation commands; you run the regression first and then do the hypothesis tests. To test whether the effects of educ and/or jobexp differ from zero (i.e. to test $\beta_1 = \beta_2 = 0$), use the `test` command:

```
. test educ jobexp

 ( 1)   educ = 0
 ( 2)   jobexp = 0

       F(  2,    16) =   27.07
            Prob > F =    0.0000
```

The `test` command does what is known as a Wald test. In this case, it gives the same result as an incremental F test.

If you want to test whether the effects of educ and jobexp are equal, i.e. $\beta_1 = \beta_2$,

```
. test educ=jobexp

 ( 1)   educ - jobexp = 0

       F(  1,    16) =   12.21
            Prob > F =    0.0030
```

If you want to see what the coefficients of the constrained model are, add the `coef` parameter:

```
. test educ=jobexp, coef

 ( 1)   educ - jobexp = 0

       F(  1,    16) =   12.21
            Prob > F =    0.0030
```

Constrained coefficients

```
------------------------------------------------------------------------------
             |      Coef.   Std. Err.       z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        educ |   .9852105   .1521516     6.48   0.000     .6869989    1.283422
      jobexp |   .9852105   .1521516     6.48   0.000     .6869989    1.283422
        race |  -6.692116   1.981712    -3.38   0.001    -10.5762    -2.808031
       _cons |   3.426358   4.287976     0.80   0.424    -4.977921    11.83064
------------------------------------------------------------------------------
```

The `testparm` and `cnsreg` commands can also be used to achieve the same results. Stata also has other commands (e.g. `testnl`) that can be used to test even more complicated hypotheses.

Nested models. Alternatively, the `nestreg` command will compute incremental F-tests. The `nestreg` command is particularly handy if you are estimating a series/hierarchy of models and want to see the regression results for each one. To again test whether the effects of educ and/or jobexp differ from zero (i.e. to test $\beta_1 = \beta_2 = 0$), the `nestreg` command would be

```
.  nestreg: regress income (race) (educ jobexp)
```

Block 1: race

```
      Source |       SS       df       MS              Number of obs =      20
-------------+------------------------------           F(  1,    18) =    8.55
       Model |  586.444492       1  586.444492         Prob > F      =  0.0090
    Residual |  1233.98098      18  68.5544991         R-squared     =  0.3221
-------------+------------------------------           Adj R-squared =  0.2845
       Total |  1820.42548      19  95.8118671         Root MSE      =  8.2798


------------------------------------------------------------------------------
      income |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        race |     -10.83   3.702823    -2.92   0.009    -18.60934   -3.050657
       _cons |      29.83   2.618291    11.39   0.000     24.32917    35.33083
------------------------------------------------------------------------------
```

Block 2: educ jobexp

```
      Source |       SS       df       MS              Number of obs =      20
-------------+------------------------------           F(  3,    16) =   29.16
       Model |  1538.92019       3  512.973396         Prob > F      =  0.0000
    Residual |  281.505287      16  17.5940804         R-squared     =  0.8454
-------------+------------------------------           Adj R-squared =  0.8164
       Total |  1820.42548      19  95.8118671         Root MSE      =  4.1945


------------------------------------------------------------------------------
      income |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        race |   .5707931   2.871949     0.20   0.845    -5.517466    6.659052
        educ |   1.981124   .3231024     6.13   0.000     1.296178     2.66607
      jobexp |   .6419622   .1811106     3.54   0.003     .2580248      1.0259
       _cons |  -7.863763   5.369166    -1.46   0.162    -19.24589    3.518362
------------------------------------------------------------------------------
```

```
  +----------------------------------------------------------+
  |       |         Block  Residual                 Change   |
  | Block |      F     df        df   Pr > F      R2  in R2   |
  |-------+--------------------------------------------------|
  |    1  |   8.55      1        18   0.0090  0.3221          |
  |    2  |  27.07      2        16   0.0000  0.8454  0.5232  |
  +----------------------------------------------------------+
```

nestreg works with several other commands (e.g. logit) and has additional options (e.g. lrtable, store) that can be very useful.

Partial and SemiPartial Correlations. There is a separate Stata routine, pcorr, which gives the partial correlations but, prior to Stata 11, did not give the semipartials. I wrote a routine, pcorr2, which gives both the partial and semipartial correlations. The pcorr command in Stata 11 took some of my code from pcorr2 and now gives very similar output. Use pcorr2 if you are stuck with an older version of Stata.

```
. pcorr income educ jobexp race
(obs=20)

Partial and semipartial correlations of income with

             Partial   Semipartial    Partial   Semipartial   Significance
  Variable |   Corr.         Corr.     Corr.^2       Corr.^2          Value
-----------+-------------------------------------------------------------------
      educ |  0.8375        0.6028      0.7015        0.3634         0.0000
    jobexp |  0.6632        0.3485      0.4399        0.1214         0.0027
      race |  0.0496        0.0195      0.0025        0.0004         0.8450
```

Stepwise Regression. The `sw` prefix lets you do stepwise regression and can be used with many commands besides `regress`. Here is how to do backwards stepwise regression. Use the `pr` (probability for removal) parameter to specify how significant the coefficient must be to avoid removal. Note that SPSS is better if you need more detailed step by step results.

```
. sw, pr(.05): regress income educ jobexp race

                     begin with full model
p = 0.8450 >= 0.0500   removing race

      Source |       SS       df       MS              Number of obs =      20
-------------+------------------------------           F(  2,    17) =   46.33
       Model |  1538.22521     2   769.112605          Prob > F      =  0.0000
    Residual |  282.200265    17   16.6000156          R-squared     =  0.8450
-------------+------------------------------           Adj R-squared =  0.8267
       Total |  1820.42548    19   95.8118671          Root MSE      =  4.0743


------------------------------------------------------------------------------
      income |      Coef.   Std. Err.      t    P>|t|     [99% Conf. Interval]
-------------+----------------------------------------------------------------
        educ |   1.933393   .2099494     9.21   0.000     1.324911    2.541875
      jobexp |   .6493654   .1721589     3.77   0.002      .150409    1.148322
       _cons |  -7.096855   3.626412    -1.96   0.067    -17.60703    3.413322
------------------------------------------------------------------------------
```

To do forward stepwise instead, use the `pe` (probability for entry) parameter to specify the level of significance for entering the model.

```
. sw, pe(.05): regress income educ jobexp race
                     begin with empty model
p = 0.0000 <  0.0500   adding   educ
p = 0.0015 <  0.0500   adding   jobexp

      Source |       SS       df       MS              Number of obs =      20
-------------+------------------------------           F(  2,    17) =   46.33
       Model |  1538.22521     2   769.112605          Prob > F      =  0.0000
    Residual |  282.200265    17   16.6000156          R-squared     =  0.8450
-------------+------------------------------           Adj R-squared =  0.8267
       Total |  1820.42548    19   95.8118671          Root MSE      =  4.0743


------------------------------------------------------------------------------
      income |      Coef.   Std. Err.      t    P>|t|     [99% Conf. Interval]
-------------+----------------------------------------------------------------
        educ |   1.933393   .2099494     9.21   0.000     1.324911    2.541875
      jobexp |   .6493654   .1721589     3.77   0.002      .150409    1.148322
       _cons |  -7.096855   3.626412    -1.96   0.067    -17.60703    3.413322
------------------------------------------------------------------------------
```

Sample Selection. In SPSS, you can use `Select If` or `Filter` commands to control which cases get analyzed. Stata also has a variety of means for handling sample selection. One of the most common ways is the use of the `if` parameter on commands. So if, for example, we only wanted to analyze Whites, we could type

```
. reg income educ jobexp if race==0

      Source |       SS       df       MS              Number of obs =      10
-------------+------------------------------           F(  2,      7) =   18.39
       Model | 526.810224      2  263.405112           Prob > F      =  0.0016
    Residual | 100.250763      7  14.3215375           R-squared     =  0.8401
-------------+------------------------------           Adj R-squared =  0.7944
       Total | 627.060987      9   69.673443           Root MSE      =  3.7844


------------------------------------------------------------------------------
      income |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        educ |   2.459518   .5265393     4.67   0.002      1.21445    3.704585
      jobexp |   .5314947   .2216794     2.40   0.048     .0073062    1.055683
       _cons |  -13.91281   7.827619    -1.78   0.119    -32.42219    4.596569
------------------------------------------------------------------------------
```

Separate Models for Groups. Or, suppose we wanted to estimate separate models for Blacks and Whites. In SPSS, we could use the `Split File` command. In Stata, we can use the `by` command (data must be sorted first if they aren't sorted already):

```
. sort race
. by race: reg income educ jobexp


_____
-> race = white

      Source |       SS       df       MS              Number of obs =      10
-------------+------------------------------           F(  2,      7) =   18.39
       Model | 526.810224      2  263.405112           Prob > F      =  0.0016
    Residual | 100.250763      7  14.3215375           R-squared     =  0.8401
-------------+------------------------------           Adj R-squared =  0.7944
       Total | 627.060987      9   69.673443           Root MSE      =  3.7844


------------------------------------------------------------------------------
      income |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        educ |   2.459518   .5265393     4.67   0.002      1.21445    3.704585
      jobexp |   .5314947   .2216794     2.40   0.048     .0073062    1.055683
       _cons |  -13.91281   7.827619    -1.78   0.119    -32.42219    4.596569
------------------------------------------------------------------------------
```

```
-> race = black

     Source |       SS       df       MS              Number of obs =      10
-------------+------------------------------            F(  2,      7) =    9.53
      Model | 443.889459      2   221.94473            Prob > F      =  0.0100
   Residual | 163.030538      7  23.2900768            R-squared     =  0.7314
-------------+------------------------------            Adj R-squared =  0.6546
      Total | 606.919997      9  67.4355552            Root MSE      =   4.826


------------------------------------------------------------------------------
     income |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       educ |   1.788485   .4541661     3.94   0.006     .7145528    2.862417
     jobexp |   .7074115   .3237189     2.19   0.065    -.058062    1.472885
      _cons |  -6.500947   6.406053    -1.01   0.344    -21.64886    8.646961
------------------------------------------------------------------------------
```

As an alternative, rather than using separate `sort` and `by` commands, you could use `bysort`:

```
. bysort race: reg income educ jobexp
```

Analyzing Means, Correlations and Standard Deviations in Stata. Sometimes you might want to replicate or modify a published analysis. You don't have the original data, but the authors have published their means, correlations and standard deviations. SPSS lets you input and analyze these directly. In Stata, you must first create a *pseudo-replication* (my term, explained in a moment) of the original data. You use Stata's `corr2data` command for this.

For example, in their classic 1985 paper, "Ability grouping and contextual determinants of educational expectations in Israel," Shavit and Williams examined the effect of ethnicity and other variables on the achievement of Israeli school children. There are two main ethnic groups in Israel: the Ashkenazim - of European birth or extraction - and the Sephardim, most of whose families immigrated to Israel during the early fifties from North Africa, Iraq, and other Mid-eastern countries. Their variables included:

X1 - Ethnicity (SPHRD) - a dummy variable coded 1 if the respondent or both his parents were born in an Asian or North African country, 0 otherwise

X2 - Parental Education (PARED) - A scale which ranges from a low of 0 to a high of 1.697

X3 - Scholastic Aptitude (APTD) - A composite score based on seven achievement tests.

Y - Grades (GRADES) - Respondent's grade-point average during the first trimester of eighth grade. This scale ranges from a low of 4 to a high of 10.

Shavit and Williams' published analysis included the following information for students who were ability grouped in their classes.

Descriptive Statistics

	Mean	Std. Deviation	N
Sephardim	.44000	.50000	10609
Parental Education Scale	.82000	.46000	10609
Scholastic Aptitude	6.46000	2.11000	10609
Grade Point Average	7.12000	1.42000	10609

Correlations

		Sephardim	Parental Education Scale	Scholastic Aptitude	Grade Point Average
Pearson Correlation	Sephardim	1.000	-.590	-.460	-.260
	Parental Education Scale	-.590	1.000	.530	.330
	Scholastic Aptitude	-.460	.530	1.000	.720
	Grade Point Average	-.260	.330	.720	1.000

To create a pseudo-replication of this data in Stata, we do the following. (I find that entering the data is most easily done via the `input matrix by hand` submenu of `Data`).

```
. * First, input the means, sds, and correlations
. matrix input Mean = (.44,.82,6.46,7.12)

. matrix input SD = (.5,.46,2.11,1.42)

. matrix input Corr = (1.00,-.59,-.46,-.26\-.59,1.00,.53,.33\-.46,.53,1.00,.72\-
.26,.33,.72,1.00)

. * Now use corr2data to create a pseudo-simulation of the data
. corr2data sphrd pared aptd grades, n(10609) means(Mean) corr(Corr) sds(SD)
(obs 10609)

. * Label the variables
. label variable sphrd "Sephardim"

. label variable pared "Parental Education Scale"

. label variable aptd "Scholastic Aptitude"

. label variable grades "Grade Point Average"

. * Confirm that all is well
. corr sphrd pared aptd grades, means
(obs=10609)

    Variable |        Mean   Std. Dev.        Min         Max
-------------+----------------------------------------------------
       sphrd |         .44          .5   -1.339484    2.122911
       pared |         .82         .46   -1.092856     2.57268
        aptd |        6.46        2.11   -1.667998    14.04385
      grades |        7.12        1.42     2.49931    11.74549


             |    sphrd    pared     aptd   grades
-------------+------------------------------------
       sphrd |   1.0000
       pared |  -0.5900   1.0000
        aptd |  -0.4600   0.5300   1.0000
      grades |  -0.2600   0.3300   0.7200   1.0000
```

As you can see, the new data set has the same means, correlations and standard deviations as the original data. You can now use the `regress` command to analyze these data. HOWEVER, several cautions should be kept in mind.

- **THESE ARE NOT THE REAL DATA!!!** The most obvious indication of this is that sphrd is not limited to values of 0 and 1. These data will work fine for correlational/regression analysis where you analyze different sets of variables. But, you should not try to analyze subsets of the data, recode variables, or compute new variables. (With SPSS, you simply input the means, correlations and standard deviations and it can handle things from there; an advantage of this is that it is more idiot-proof than an analysis of data created by `corr2data` is.) This is why I call the data a pseudo-replication of the original.

- Even if you have entered the data correctly, you may not be able to perfectly replicate published results. Simple rounding in the published results (e.g. only reporting correlations to 2 or 3 decimal places) can cause slight differences. More critically, because of missing data, subsample analyses, and other reasons, cases examined are not always the same throughout an analysis, e.g. 10609 cases might be analyzed in one regression, 10274 might be analyzed in another, etc. For example, if you regress grades on pared, sphrd, and aptd you get results that are very close, but not identical, to those reported by Shavit and Williams on p.71, Table 4 of their paper. If you get results that are very different, then the cases used to compute the correlations may be very different from the cases analyzed in that portion of the paper. (Either that, or you've entered the data wrong.)

Factor Variables (Stata 11 and higher).

Factor variables (not to be confused with factor analysis) were introduced in Stata 11. Factor variables provide a convenient means of computing and including dummy variables, interaction terms, and squared terms in models. They can be used with `regress` and several other (albeit not all) commands. For example,

```
. use https://www3.nd.edu/~rwilliam/statafiles/reg01.dta, clear
. regress income educ jobexp i.race
```

Source	SS	df	MS		Number of obs =	20
					F(3, 16) =	29.16
Model	1538.92019	3	512.973396		Prob > F =	0.0000
Residual	281.505287	16	17.5940804		R-squared =	0.8454
					Adj R-squared =	0.8164
Total	1820.42548	19	95.8118671		Root MSE =	4.1945

income	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
educ	1.981124	.3231024	6.13	0.000	1.296178	2.66607
jobexp	.6419622	.1811106	3.54	0.003	.2580248	1.0259
1.race	.5707931	2.871949	0.20	0.845	-5.517466	6.659052
_cons	-7.863763	5.369166	-1.46	0.162	-19.24589	3.518362

The i.black notation tells Stata that black is a categorical variable rather than continuous. As the Stata 11 User Manual explains (section 11.4.3.1), "i.group is called a factor variable, although more correctly, we should say that group is a categorical variable to which factor-variable operators have been applied...When you type i.group, it forms the indicators for the unique values of group."

In other words, Stata, in effect, creates dummy variables coded 0/1 from the categorical variable. In this case, of course, black is already coded 0/1 – but `margins` and other post-estimation commands still like you to use the i. notation so they know the variable is categorical (rather than,

say, being a continuous variable that just happens to only have the values of 0/1 in this sample). But if, say, we had the variable race coded 1 = White, 2 = Black, the new variable would be coded 0 = White, 1 = Black.

Or, if the variable religion was coded 1 = Catholic, 2 = Protestant, 3 = Jewish, 4 = Other, saying i.religion would cause Stata to create three 0/1 dummies. By default, the first category (in this case Catholic) is the reference category, but we can easily change that, e.g. ib2.religion would make Protestant the reference category, or ib(last).religion would make the last category, Other, the reference.

Factor variables can also be used to include squared terms and interaction terms in models. For example,

```
. reg income educ jobexp i.race c.educ#c.educ

      Source |       SS       df       MS              Number of obs =      20
-------------+------------------------------           F(  4,    15) =   21.76
       Model |  1552.77105    4  388.192763            Prob > F      =  0.0000
    Residual |  267.654424   15  17.8436282            R-squared     =  0.8530
-------------+------------------------------           Adj R-squared =  0.8138
       Total |  1820.42548   19  95.8118671            Root MSE      =  4.2242


------------------------------------------------------------------------------
      income |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        educ |   1.315309    .822787     1.60   0.131    -.4384204    3.069037
      jobexp |   .6223253   .1837473     3.39   0.004     .2306772    1.013973
      1.race |   1.097706    2.95343     0.37   0.715    -5.197381    7.392793
             |
 c.educ#c.educ|   .0318724   .0361758    0.88   0.392    -.0452345    .1089793
             |
       _cons |  -5.090782   6.256426    -0.81   0.429    -18.42604    8.244475
------------------------------------------------------------------------------

. reg income educ jobexp i.race i.race#c.educ

      Source |       SS       df       MS              Number of obs =      20
-------------+------------------------------           F(  4,    15) =   21.80
       Model |   1553.2421    4  388.310524            Prob > F      =  0.0000
    Residual |  267.183378   15  17.8122252            R-squared     =  0.8532
-------------+------------------------------           Adj R-squared =  0.8141
       Total |  1820.42548   19  95.8118671            Root MSE      =  4.2205


------------------------------------------------------------------------------
      income |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        educ |   2.408641   .5770636     4.17   0.001     1.178659    3.638623
      jobexp |   .6076061   .1862143     3.26   0.005     .2106999    1.004512
      1.race |   8.981915   9.815228     0.92   0.375    -11.93875    29.90258
             |
 race#c.educ |
           1 |  -.6310502   .7037567    -0.90   0.384    -2.131072    .8689717
             |
       _cons |  -14.00968   8.727139    -1.61   0.129    -32.61113    4.591777
------------------------------------------------------------------------------
```

The # (pronounced cross) operator is used for interactions and product terms. The use of # implies the i. prefix, i.e. unless you indicate otherwise Stata will assume that the variables on both sides of

the # operator are categorical and will compute interaction terms accordingly. Hence, we use the c. notation to override the default and tell Stata that educ is a continuous variable. So, c.educ#c.educ tells Stata to include educ^2 in the model; we do not want or need to compute the variable separately. Similarly, i.race#c.educ produces the race * educ interaction term. Stata also offers a ## notation, called factorial cross. It can save some typing and/or provide an alternative parameterization of the results.

At first glance, the use of factor variables might seem like a minor convenience at best: They save you the trouble of computing dummy variables and interaction terms beforehand. Further, factor variables have some disadvantages, e.g. as of this writing they cannot be used with `nestreg` or `stepwise`. The advantages of factor variables become much more apparent when used in conjunction with post-estimation commands such as `margins`.

Adding to Stata. SPSS is pretty much a closed-ended program. If it doesn't have what you want already built-in, you are out of luck. With Stata, however, it is possible to write your own routines which add to the functionality of the program. Further, many such routines are publicly available and can be easily installed on your machine. I've often found that something that SPSS had and Stata did not could be added to Stata. For example, prior to Stata 11 Stata did NOT have a built-in command for computing semipartial correlations. To see if such a routine exists, from within Stata you can type

```
findit semipartial
```

The `findit` command will give you listings of programs that have the keyword semipartial associated with them. It will also give you FAQs and Stata help associated with the term. Among the things that will pop up is my very own `pcorr2`. Usually a routine includes a brief description and you can view its help file. Sometimes routines are part of a package of related routines and you install the entire package. Once you have found a routine that sounds like what you want, you can easily install it. You can also easily uninstall if you decide you do not want it.

A couple of cautions:

• User written routines are not officially supported by Stata. Indeed, it is entirely possible that such a routine has bugs or gives incorrect results, at least under certain conditions. Most of the routines I have installed seem to work fine, but I have found a few problems. You might want to double-check results against SPSS or published findings.
• If a command works on one machine but not another, it is probably because that command is not installed on both machines. For example, if the `pcorr2` command was not working, type `findit pcorr2` and then install it on your machine. (A possible complication is that you may find newer and older versions of the same command, and you may even find two different commands with the same name. So, check to make sure you are getting what you think you are getting. If you ever write your own routine, I suggest you try something like `findit myprog` to make sure somebody isn't already using the name you had in mind.)

Other Comments.

- Unlike SPSS, Stata is picky about case, e.g. `findit pcorr2` works, `Findit pcorr2` does not. `Income`, `income` and `INCOME` would all be different variable names.
- Stata 8 added a menu-structure that made it more SPSS-like. This can be very handy for commands you are not familiar with. For commands I know, however, I generally find it easier just to type the command in directly.
- There are various other options for the `regress` command and several other post-estimation commands that can be useful. We will discuss these later.

Here is all the Stata code used in this handout:

```
use https://www3.nd.edu/~rwilliam/statafiles/reg01.dta, clear
correlate income educ jobexp race, means
regress income educ jobexp race
regress income educ jobexp race, level(99)
set level 99
regress income educ jobexp race
regress income educ jobexp race, beta
regress income educ jobexp race
regress, level(99)
regress, beta
vif
collin educ jobexp race if !missing(income)
test educ jobexp
test educ=jobexp
test educ=jobexp, coef
nestreg: regress income (race) (educ jobexp)
pcorr income educ jobexp race
sw, pr(.05): regress income educ jobexp race
sw, pe(.05): regress income educ jobexp race
reg income educ jobexp if race==0
sort race
by race: reg income educ jobexp
bysort race: reg income educ jobexp
* First, input the means, sds, and correlations
matrix input Mean = (.44,.82,6.46,7.12)
matrix input SD = (.5,.46,2.11,1.42)
matrix input Corr = (1.00,-.59,-.46,-.26\-.59,1.00,.53,.33\-.46,.53,1.00,.72\-.26,.33,.72,1.00)
* Now use corr2data to create a pseudo-simulation of the data
corr2data sphrd pared aptd grades, n(10609) means(Mean) corr(Corr) sds(SD)
* Label the variables
label variable sphrd "Sephardim"
label variable pared "Parental Education Scale"
label variable aptd "Scholastic Aptitude"
label variable grades "Grade Point Average"
* Confirm that all is well
corr sphrd pared aptd grades, means
use https://www3.nd.edu/~rwilliam/statafiles/reg01.dta, clear
regress income educ jobexp i.race
reg income educ jobexp i.race c.educ#c.educ
reg income educ jobexp i.race i.race#c.educ
```