Introduction to Stata – Handout 2: Regression

Hayley Fisher

30 November 2010


1 Introduction

This lecture introduces basic econometric techniques using Stata. It uses the dataset from lecture 1, an additional dataset of fertility rates. The examples given here are illustrative only rather than being part of a valid research project. We start by making some small changes to the dataset we have. First create two dummy variables for ‘female’ and ‘married’:

```stata
generate female=sex==2
generate married=marst==1
```

Next we drop all observations with missing values for `lnearn`, as we will use this as our dependent variable. Of course, this generates a selected sample which introduces biases. It would be better to perform analysis controlling for this selection – see chapter 16 of Cameron and Trivedi (2009) for more information.

```stata
.drop if lnearn=.  
(18894 observations deleted)
```

Finally, we take a random 10% sample of the dataset to give a more manageable size for analysis (estimating models for very large datasets can take some time). It is important to ensure this is reproducible so we set the seed of the pseudo-random number generator first.

```stata
.set seed 10101
.sample 10
(57086 observations deleted)
```

2 Ordinary Least Squares

Standard OLS is implemented using the `regress` command which has the following syntax:

```stata
regress depvar [indepvars] [if] [in] [weight] [, options]
```

Here anything in square brackets is optional. This is a typical syntax for Stata commands (very similar to the `probit` and `logit` commands we will see later).

We start by regressing log earnings on various characteristics and hours worked:

```stata
.regress lnearn age age2 female married white hours eddummy1 eddummy2 unemp
```

```
<table>
<thead>
<tr>
<th>Source</th>
<th>SS</th>
<th>df</th>
<th>MS</th>
<th>Number of obs = 6343</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3752.58034</td>
<td>9</td>
<td>416.953371</td>
<td>F( 9, 6333) = 595.68</td>
</tr>
<tr>
<td>Model</td>
<td>F( 9,  6333) = 595.68</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prob &gt; F = 0.0000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

1
The output reports coefficients, standard errors, t-statistics, p-values and confidence intervals for all regressors. Note that a constant is automatically included. We see here that the coefficient on the white dummy is insignificantly different from zero, as is that on the state unemployment rate. Since our independent variable is in logs, the coefficients should be interpreted as semielasticities – getting a year older is associated with a 9.1% increase in earnings. The regressors are jointly statistically significant, with an overall $F$ statistic of 595.68.

### 2.1 Hypothesis tests

We can test simple hypotheses using the `test` command, for example:

```
. test white
  ( 1) white = 0
     F(  1, 6333) =  0.02
     Prob > F =  0.8823

. test female married white
  ( 1) female = 0
  ( 2) married = 0
  ( 3) white = 0
     F(  3, 6333) = 113.79
     Prob > F =  0.0000

. test married=white
  ( 1) married - white = 0
     F(  1, 6333) =  5.67
     Prob > F =  0.0173
```

In all cases the p-values are reported making it easy to assess whether the hypothesis is rejected (as it is in all cases above).

### 2.2 Standard errors

The standard OLS estimates above assume homoscedasticity of the error term. We can include the option `vce()` to report standard errors that are robust to heteroscedasticity – for example `vce(robust)` where observations are independent, or `vce(robust cluster var)` where there is correlation between observations in...
particular clusters (for example if you have multiple observations from the same individual). Below are the results for robust standard errors, and robust standard errors clustered by state:

```stata
.regress lnearn age age2 female married white hours eddummy1 eddummy2 unemp,
vce(robust)
```

```
Linear regression
Number of obs = 6343
F( 9, 6333) = 432.16
Prob > F = 0.0000
R-squared = 0.4584
Root MSE = .83664

------------------------------------------------------------------------------
| Robust
| lnearn | Coef. Std. Err. t P>|t| [95% Conf. Interval]
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>age</td>
<td>.0911226 .0066377 13.73 0.000 .0781105 .1041347</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>age2</td>
<td>-.0009574 .0000822 -11.64 0.000 -.0011186 -.0007961</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>female</td>
<td>-.3825226 .0214864 -17.80 0.000 -.4246431 -.3404021</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>married</td>
<td>.099764 .0214864 4.36 0.000 .0548682 .1446598</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>white</td>
<td>.0044039 .0214864 4.36 0.000 .0548682 .1446598</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hours</td>
<td>.0004843 .0000822 -11.64 0.000 -.0011186 -.0007961</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>eddummy1</td>
<td>-.8259626 .0379387 -21.77 0.000 -.9003353 -.7515899</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>eddummy2</td>
<td>-.3996877 .0235131 -17.00 0.000 -.4457813 -.3535942</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>unemp</td>
<td>.0164931 .0132059 1.25 0.212 -.0093949 .0423811</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>_cons</td>
<td>7.463438 .1400551 53.29 0.000 7.188882 7.737993</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
------------------------------------------------------------------------------

(Std. Err. adjusted for 51 clusters in statefip)
```

```stata
.regress lnearn age age2 female married white hours eddummy1 eddummy2 unemp,
vce(cluster state)
```

```
Linear regression
Number of obs = 6343
F( 9, 50) = 363.60
Prob > F = 0.0000
R-squared = 0.4584
Root MSE = .83664

------------------------------------------------------------------------------
| Robust
| lnearn | Coef. Std. Err. t P>|t| [95% Conf. Interval]
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>age</td>
<td>.0911226 .0066377 13.73 0.000 .0781105 .1041347</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>age2</td>
<td>-.0009574 .0000822 -11.64 0.000 -.0011186 -.0007961</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>female</td>
<td>-.3825226 .0214864 -17.80 0.000 -.4246431 -.3404021</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>married</td>
<td>.099764 .0214864 5.36 0.000 .0548682 .1446598</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>white</td>
<td>.0044039 .0214864 5.36 0.000 .0548682 .1446598</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hours</td>
<td>.0004843 .0000876 -10.92 0.000 -.0011186 -.0007961</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>eddummy1</td>
<td>-.8259626 .0379387 -21.77 0.000 -.9003353 -.7515899</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>eddummy2</td>
<td>-.3996877 .0259839 -17.00 0.000 -.4457813 -.3535942</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>unemp</td>
<td>.0164931 .0132059 1.25 0.212 -.0093949 .0423811</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>_cons</td>
<td>7.463438 .1400551 53.29 0.000 7.188882 7.737993</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
------------------------------------------------------------------------------
```
The standard error on the unemployment rate coefficient is notably larger when standard errors are clustered by state.

2.3 Comparing results for different groups

We can perform this regression separately for men and women using the by prefix. The data must first be sorted by the variable in question:

```
. sort sex
. by sex: regress lnearn age age2 married white hours eddummy1 eddummy2 unemp, vce(robust)
```

-> sex = Male

```
Linear regression
Number of obs = 3178
F( 8, 3169) = 214.42
Prob > F = 0.0000
R-squared = 0.4257
Root MSE = .79875
```

|         | Coef.  | Std. Err. | t     | P>|t|   | [95% Conf. Interval] |
|---------|--------|-----------|-------|-------|----------------------|
| lnearn  |        |           |       |       |                      |
| age     | .1113265 | .0093605  | 11.89 | 0.000 | .0929733 .1296797   |
| age2    | -.001178 | .0001148  | -10.26| 0.000 | -.001403 -.0009529 |
| married | .2060571 | .0310567  | 6.63  | 0.000 | .1451638 .2669505  |
| white   | .0686291 | .0397102  | 1.73  | 0.084 | -.0092313 .1464894 |
| hours   | .00038   | .0000205  | 18.56 | 0.000 | .0003399 .0004202  |
| eddummy1| -.7206743| .0462098  | -15.60| 0.000 | -.8112784 -.6300703 |
| eddummy2| -.3835108| .0324672  | -11.81| 0.000 | -.4471695 -.319852  |
| unemp   | -.0010111| .0172191  | -0.06 | 0.953 | -.0347728 .0327505 |
| _cons   | 7.172249 | .1937134  | 37.03 | 0.000 | 6.792433 7.552066  |

-> sex = Female

```
Linear regression
Number of obs = 3165
F( 8, 3156) = 222.97
Prob > F = 0.0000
R-squared = 0.4364
Root MSE = .85867
```

|         | Coef.  | Std. Err. | t     | P>|t|   | [95% Conf. Interval] |
|---------|--------|-----------|-------|-------|----------------------|
| lnearn  |        |           |       |       |                      |
| age     | .0750468 | .0093228  | 8.05  | 0.000 | .0567675 .0933262   |
| age2    | -.0008003| .0001164  | -6.87 | 0.000 | -.0010286 -.000572  |
| married | .0265275 | .0329567  | 0.80  | 0.421 | -.0380913 .0911463  |
| white   | -.0365954| .0424572  | -0.86 | 0.389 | -.1198418 .046651   |
| hours   | .0005827 | .0000235  | 24.84 | 0.000 | .0005367 .0006287   |
| eddummy1| -.9438047| .0622072  | -15.17| 0.000 | -1.065775 -.8218341 |
Whilst this provides the estimates, it is easier to compare the differences between the results for men and women by putting the results into a table – two ways of doing this are the `estimates table` command and the (user-written) `esttab` command. To use these the estimates must be stored. Here I use the `quietly` prefix to omit the initial output from the estimation.

```
. quietly regress lnearn age age2 married white hours eddummy1 eddummy2 unemp if sex==1, vce(robust)
. estimates store men
. quietly regress lnearn age age2 married white hours eddummy1 eddummy2 unemp if sex==2, vce(robust)
. estimates store women
```

The simplest table using `estimates table` is shown below.

```
. estimates table men women

----------------------------------------
Variable | men    | women   
-------------+--------------------------
age | .11132648 | .07504684
age2 | -.00117797 | -.00080028
married | .20605714 | .02652752
white | .06862907 | -.03659541
hours | .00038002 | .00058268
eddummy1 | -.72067435 | -.94380466
eddummy2 | -.38351076 | -.40949046
unemp | -.00101113 | .0299791
_cons | 7.1722493 | 7.3324514

----------------------------------------
```

A better table can be produced by adding some options – for example reporting coefficients, standard errors and t-statistics with three decimal places (see `help estimates table` for more options):

```
. estimates table men women, b(%5.3f) se(%5.3f) t(%5.3f)

------------------------------------
Variable |    men   |    women  
-------------+----------+----------
age | 0.111 0.075
| 0.009 0.009
| 11.893 8.050
age2 | -0.001 -0.001
| 0.000 0.000
| -10.263 -6.873
married | 0.206 0.027
| 0.031 0.033
| 6.635 0.805
white | 0.069 -0.037
| 0.040 0.042
| 1.728 -0.862

------------------------------------
```
<table>
<thead>
<tr>
<th></th>
<th>(1)</th>
<th>(2)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>lnearn</td>
<td>lnearn</td>
</tr>
<tr>
<td>age</td>
<td>0.111***</td>
<td>0.075***</td>
</tr>
<tr>
<td></td>
<td>(0.009)</td>
<td>(0.009)</td>
</tr>
<tr>
<td>age2</td>
<td>-0.001***</td>
<td>-0.001***</td>
</tr>
<tr>
<td></td>
<td>(0.000)</td>
<td>(0.000)</td>
</tr>
<tr>
<td>married</td>
<td>0.206***</td>
<td>0.027</td>
</tr>
<tr>
<td></td>
<td>(0.031)</td>
<td>(0.033)</td>
</tr>
<tr>
<td>white</td>
<td>0.069</td>
<td>-0.037</td>
</tr>
<tr>
<td></td>
<td>(0.040)</td>
<td>(0.042)</td>
</tr>
<tr>
<td>hours</td>
<td>0.000***</td>
<td>0.001***</td>
</tr>
<tr>
<td></td>
<td>(0.000)</td>
<td>(0.000)</td>
</tr>
<tr>
<td>eddummy1</td>
<td>-0.721***</td>
<td>-0.944***</td>
</tr>
<tr>
<td></td>
<td>(0.046)</td>
<td>(0.062)</td>
</tr>
<tr>
<td>eddummy2</td>
<td>-0.384***</td>
<td>-0.409***</td>
</tr>
<tr>
<td></td>
<td>(0.032)</td>
<td>(0.034)</td>
</tr>
<tr>
<td>unemp</td>
<td>-0.001</td>
<td>0.030</td>
</tr>
<tr>
<td></td>
<td>(0.017)</td>
<td>(0.020)</td>
</tr>
<tr>
<td>_cons</td>
<td>7.172***</td>
<td>7.332***</td>
</tr>
<tr>
<td></td>
<td>(0.194)</td>
<td>(0.199)</td>
</tr>
</tbody>
</table>

Better tables of estimates are created using esttab. These can also be outputted in several formats to be put directly into reports or papers (for example rich text format to put into a word processor, or tex to put directly into latex).

. esttab men women, b(%5.3f) se(%5.3f) r2
When you are running several regressions with the same independent variables, it often makes sense to create a local macro containing the variable list to save you typing it out each time. This is done by typing `local macroname varlist`, and the macro is called by typing `macroname` (note that the first single quote is a backtick).

```
local xlist "age age2 female married white hours eddummy1 eddummy2 unemp"
```

### 2.4 Predicted values

After estimating a model, new variables can be created which contain predicted values and residuals. These can then be used to create graphs. Use the command `predict`. After using `regress` the default is to calculate the linear prediction (other regression commands have different defaults) – other results can be calculated by including an option, eg. `residuals`.

```
quietly regress lnearn 'xlist', vce(robust)
predict yhat
(option xb assumed; fitted values)
predict resid, residuals
twoway (scatter resid yhat)
```

### 3 Binary outcome models

We now consider estimation for models where the dependent variable has a binary outcome. Here we consider the outcome of marital status – whether an individual is married or not. We might estimate the association...
between being married and various other outcomes using a linear probability model (implemented by OLS as above), or by probit or logit models. Let’s start by creating a local macro for our RHS variables:

```
. local xlist1 "age age2 nchild race eddummy1 eddummy2 earnings hours unemp"
```

Whilst the linear probability model is implemented using `regress`, we use `probit` and `logit` for the alternative methods (which have the same syntax as `regress`). If we store the results we can create a table to easily compare the coefficient estimates.

```
. regress married 'xlist1'
output omitted
. estimates store LPM

. probit married 'xlist1'
Iteration 0: log likelihood = -4346.6806
Iteration 1: log likelihood = -3320.7888
Iteration 2: log likelihood = -3272.4848
Iteration 3: log likelihood = -3272.077
Iteration 4: log likelihood = -3272.077
Probit regression
Number of obs = 6343
LR chi2(9) = 2149.21
Prob > chi2 = 0.0000
Log likelihood = -3272.077 Pseudo R2 = 0.2472

------------------------------------------------------------------------------
marrried | Coef.  Std. Err.   z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
age |  .0823594  .0109988  7.49    0.000     .0608021    .1039167
age2 |  -.0005835  .0001347  -4.33   0.000    -.0008475   -.0003195
nchild |  .5518246  .0189196  29.17   0.000     .514743    .5889063
race |  -.0009319  .0001765  -5.28   0.000    -.0012779   -.0005859
eddummy1 |  -.3668029  .0636444  -5.76   0.000    -.4915436   -.2420621
eddummy2 |  -.1662635  .0410681  -4.05   0.000    -.2467554   -.0857716
earnings |  3.26e-06   7.55e-07   4.32   0.000     1.78e-06    4.74e-06
hours |  .0000449  .0000204   2.20   0.028     4.83e-06    .0000849
unemp |  -.0054954  .0226362  -0.24   0.808     -.0498616   .0388708
_cons |  -2.417738  .2279786  -10.61   0.000    -2.864568   -1.970908
------------------------------------------------------------------------------
```

```
. estimates store probit

. logit married 'xlist1'
Iteration 0: log likelihood = -4346.6806
Iteration 1: log likelihood = -3326.4994
Iteration 2: log likelihood = -3258.5613
Iteration 3: log likelihood = -3256.2592
Iteration 4: log likelihood = -3256.2551
Logistic regression
Number of obs = 6343
LR chi2(9) = 2180.85
```

Log likelihood = -3256.2551  
Prob > chi2 = 0.0000  
Pseudo R2 = 0.2509  

------------------------------------------------------------------------------
married | Coef. Std. Err. z P>|z|   [95% Conf. Interval]
-------------+----------------------------------------------------------------
age | .1275906 .0186985 6.82 0.000 .0909422 .1642389
age2 | -.0008544 .0002285 -3.74 0.000 -.0013022 -.0004067
nchild | .9995099 .0362551 27.57 0.000 .9284513 1.070569
race | -.0015718 .0003055 -5.14 0.000 -.0021706 -.000973
eddummy1 | -.6138059 .1095583 -5.60 0.000 -.8285363 -.4088788
eddummy2 | -.2716175 .0700338 -3.88 0.000 -.4088818 -.1343537
earnings | 5.68e-06 1.37e-06 4.14 0.000 2.99e-06 8.36e-06
hours | .0000699 .000035 1.99 0.046 1.20e-06 .0001386
unemp | -.0161583 .0384904 -0.42 0.675 -.0915982 .0592815
_cons | -3.849691 .3891784 -9.89 0.000 -4.612466 -3.086915
------------------------------------------------------------------------------

. estimates store logit  
Note that the probit and logit results here report coefficient estimates which are not equal to marginal effects 
as they are in the linear probability model. We do not expect coefficients from the different models to be 
equal (see p.451 in Cameron and Trivedi (2009) for rough conversions).

. esttab LPM probit logit, b(%5.3f) se(%5.3f) mtitles

<table>
<thead>
<tr>
<th></th>
<th>(1)</th>
<th>(2)</th>
<th>(3)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LPM</td>
<td>probit</td>
<td>logit</td>
</tr>
<tr>
<td>age</td>
<td>0.026***</td>
<td>0.082***</td>
<td>0.128***</td>
</tr>
<tr>
<td></td>
<td>(0.003)</td>
<td>(0.011)</td>
<td>(0.019)</td>
</tr>
<tr>
<td>age2</td>
<td>-0.000***</td>
<td>-0.001***</td>
<td>-0.001***</td>
</tr>
<tr>
<td></td>
<td>(0.000)</td>
<td>(0.000)</td>
<td>(0.000)</td>
</tr>
<tr>
<td>nchild</td>
<td>0.163***</td>
<td>0.552***</td>
<td>1.000***</td>
</tr>
<tr>
<td></td>
<td>(0.005)</td>
<td>(0.019)</td>
<td>(0.036)</td>
</tr>
<tr>
<td>race</td>
<td>-0.000***</td>
<td>-0.001***</td>
<td>-0.002***</td>
</tr>
<tr>
<td></td>
<td>(0.000)</td>
<td>(0.000)</td>
<td>(0.000)</td>
</tr>
<tr>
<td>eddummy1</td>
<td>-0.100***</td>
<td>-0.367***</td>
<td>-0.614***</td>
</tr>
<tr>
<td></td>
<td>(0.019)</td>
<td>(0.064)</td>
<td>(0.110)</td>
</tr>
<tr>
<td>eddummy2</td>
<td>-0.045***</td>
<td>-0.166***</td>
<td>-0.272***</td>
</tr>
<tr>
<td></td>
<td>(0.012)</td>
<td>(0.041)</td>
<td>(0.070)</td>
</tr>
<tr>
<td>earnings</td>
<td>0.000***</td>
<td>0.000***</td>
<td>0.000***</td>
</tr>
<tr>
<td></td>
<td>(0.000)</td>
<td>(0.000)</td>
<td>(0.000)</td>
</tr>
<tr>
<td>hours</td>
<td>0.000</td>
<td>0.000*</td>
<td>0.000*</td>
</tr>
<tr>
<td></td>
<td>(0.000)</td>
<td>(0.000)</td>
<td>(0.000)</td>
</tr>
<tr>
<td></td>
<td>Model 1</td>
<td>Model 2</td>
<td>Model 3</td>
</tr>
<tr>
<td>--------</td>
<td>---------</td>
<td>---------</td>
<td>---------</td>
</tr>
<tr>
<td>unemp</td>
<td>-0.003</td>
<td>-0.005</td>
<td>-0.016</td>
</tr>
<tr>
<td></td>
<td>(0.007)</td>
<td>(0.023)</td>
<td>(0.038)</td>
</tr>
<tr>
<td>_cons</td>
<td>-0.255***</td>
<td>-2.418***</td>
<td>-3.850***</td>
</tr>
<tr>
<td></td>
<td>(0.064)</td>
<td>(0.228)</td>
<td>(0.389)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Model 1</th>
<th>Model 2</th>
<th>Model 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>6343</td>
<td>6343</td>
<td>6343</td>
</tr>
</tbody>
</table>

Standard errors in parentheses
* p<0.05, ** p<0.01, *** p<0.001

The `mtitles` option for `esttab` includes the estimates titles in the header of the table. Here we see that the signs and significance of the coefficients are similar between the three models. However, it is better to compare marginal effects between models, shown later.

Another useful trick to making simple do-files and saving stages of work is to use the `xi` prefix if you wish to include a set of dummy variables from a categorical variable. For example, if we want to allow the marriage rate to vary across states, we can take this approach and save having to create a large set of dummy variables. This reduces the size of the dataset, but adds to the time taken to calculate the results.

We include `i.statefip` as a regressor here, and Stata creates a set of variables `_Istatefip_1-56`, of which one is immediately dropped to avoid collinearity.

```stata
. xi: probit married 'xlist1' i.statefip
   (naturally coded; _Istatefip_1 omitted)
```

(further output omitted)

```stata
. estimates store probit1
```

We can compare these results to the probit model without the state dummies in a table – dropping the state dummies here.

```stata
. esttab probit probit1, b(%5.3f) se(%5.3f) mtitles drop(_Istate*)
```

<table>
<thead>
<tr>
<th></th>
<th>(1)</th>
<th>(2)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>probit</td>
<td>probit1</td>
</tr>
<tr>
<td>age</td>
<td>0.082***</td>
<td>0.083***</td>
</tr>
<tr>
<td></td>
<td>(0.011)</td>
<td>(0.011)</td>
</tr>
<tr>
<td>age2</td>
<td>-0.001***</td>
<td>-0.001***</td>
</tr>
<tr>
<td></td>
<td>(0.000)</td>
<td>(0.000)</td>
</tr>
<tr>
<td>nchild</td>
<td>0.552***</td>
<td>0.557***</td>
</tr>
<tr>
<td></td>
<td>(0.019)</td>
<td>(0.019)</td>
</tr>
<tr>
<td>race</td>
<td>-0.001***</td>
<td>-0.001***</td>
</tr>
<tr>
<td></td>
<td>(0.000)</td>
<td>(0.000)</td>
</tr>
<tr>
<td>eddummy1</td>
<td>-0.367***</td>
<td>-0.365***</td>
</tr>
<tr>
<td></td>
<td>(0.064)</td>
<td>(0.065)</td>
</tr>
<tr>
<td>eddummy2</td>
<td>-0.166***</td>
<td>-0.181***</td>
</tr>
<tr>
<td></td>
<td>(0.041)</td>
<td>(0.042)</td>
</tr>
</tbody>
</table>
### 3.1 Marginal effects

The marginal effects of variables can be found using the `mfx` command after estimation. The default output of `mfx` after `probit` and `logit` is the marginal effects desired.

```
. quietly probit married 'xlist1'
. mfx
```

```
Marginal effects after probit
   y = Pr(married) (predict)
        = .57985984

------------------------------------------------------------------------------
variable | dy/dx Std. Err. z P>|z| [ 95% C.I. ] X
---------|--------------------------------------------------------------------
   age |  .0321961    0.00431  7.46 0.000  .02374  .040652  38.624
  age2 | -.0002281    0.00005 -4.32 0.000 -.000331 -.000125 1637.34
   nch |  .2157205    0.0073  29.56 0.000  .201418  .230023 .900836
    race| -.0003643    0.00007 -5.28 0.000 -.0005 -.000229 131.696
eddummy1*| -.1452232    0.02514 -5.78 0.000 -.194488 -.095958 .124862
eddummy2*| -.0648577    0.01597 -4.06 0.000 -.096149 -.033566 .530348
   earn | 1.27e-06     0.00000  4.32 0.000  7.0e-07  1.9e-06 30367.8
   hours | .0000175     0.00001  2.20 0.028  .000033  .000036 1704.92
   unemp| -.0021483    0.00885 -0.24 0.808 -.019492 .015195  4.07124
------------------------------------------------------------------------------
(*) dy/dx is for discrete change of dummy variable from 0 to 1
```

So here being a year older is associated with a 3% increase in the probability of being married; having an extra child is associated with a 22% increase (compared to 3% and 16% from the linear probability model).

### References


---

1This command has been superceded by “margins” in Stata 11 – see help margins within Stata 11 for more information. “mfx” still works.