Model estimation

Model estimation with Stata

- Stata features a vast number of empirical models that you can fit to your data.
- Most model estimation commands in Stata use a standard syntax:

  `model_command` `depvar indepvarlist` [*if*] [*weight*], `options`

  - **`model_command`** is the name of a model estimation command.
  - `depvar` is the name of the dependent variable (outcome).
  - `indepvarlist` is a list of independent variables (predictors).
  - *if* restricts your model to subgroups of your data.
  - Use *weight*s to make your sample representative of the underlying population.
  - `options` may be specific to an estimated model.

Linear Regression

- As an illustrative example, we will estimate a linear regression model with **Ordinary Least Squares (OLS)** with the academic performance data from our last session:

```
use "https://stats.idre.ucla.edu/stat/data/hs0", clear
```

```
regress depvar indepvarlist
```

Do writing scores predict math scores?

Let's estimate a model for regressing math scores on writing scores and gender.

- Writing scores are a **continuous** variable which we can make explicit by putting `c.` in front of its name.
- Use `i.` to tell Stata that you are using a **categorical** variable such as gender.
- By using `i.`, Stata will automatically create dummy (0/1) indicator variables and enter all but one (the first, by default) into the regression. This way, Stata protects you from falling into the dummy variable trap.

! In the data, gender is not coded as a dummy (0/1) variable. By using `i.` Stata will convert the variable for you. However, it may make sense to **`recode`** such variables before doing statistical analyses.

Regression output

```
regress math c.write i.gender

      Source |       SS           df       MS      Number of obs   =       200
-------------+----------------------------------   F(2, 197)       =     71.02
       Model |  7317.4569         2  3658.72845   Prob > F        =    0.0000
    Residual |  10148.3381       197  51.5144066   R-squared       =    0.4190
-------------+----------------------------------   Adj R-squared   =    0.4131
       Total |  17465.795       199  87.7678141   Root MSE        =    7.1774

------------------------------------------------------------------------------
        math | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
       write |   .6612119   .0555356    11.91   0.000     .5516913    .7707325
    2.gender |  -3.770626   1.054442    -3.58   0.000    -5.850069   -1.691184
       _cons |   19.80453   2.883388     6.87   0.000     14.11826     25.4908
------------------------------------------------------------------------------
```

Tuning your model

- The `vce` option allows you to estimate your model with (cluster) robust standard errors.
- You can set base levels *k* for dummies by using `ibk.` in front of the categorical variable.
- You can include interactions combining two or more variables with #.
- When using ## for interactions, Stata automatically estimates lower level effects.

```
. regress math c.write##i.gender, vce(robust)

Linear regression
```

math	Coefficient	Robust std. err.	t
write	.5878634	.0765533	7.68
2.gender	-12.84243	5.57816	-2.30
gender#c.write			
2	.171465	.1072757	1.60
_cons	23.48083	3.773693	6.22

What's in store, again?

- We've already learnt that Stata stores results in lists, e.g., the *mean* after **summarize** in an `r-list`.
- Similarly, Stata stores results of estimated models in an `e-list`.
- Estimated coefficients are stored in the matrix `e(b)`.
- The variance-covariance matrix of the estimates is stored in `e(V)`.
- To find out under what name Stata stores an estimate, use the **coeflegend** option in your **regress** command.

```
regress math write i.gender, coeflegend

      Source |       SS           df       MS
-------------+----------------------------------
       Model |  7317.4569         2  3658.
    Residual |  10148.3381       197  51.51
-------------+----------------------------------
       Total |  17465.795       199  87.76

------------------------------------------------
        math | Coefficient  Legend
-------------+----------------------------------
       write |   .6612119  _b[write]
    2.gender |  -3.770626  _b[2.gender]
       _cons |   19.80453  _b[_cons]
------------------------------------------------
```

Estimation and beyond

- Stata includes some basic model statistics in its default regression output. However, you might want to run additional tests and analyses based on your estimated model.

- Use Stata's `postestimation` commands to conduct additional analysis:

  help `regress postestimation`

- Examples: model predictions, joint tests of coefficients or linear combination of statistics, marginal estimates

Postestimation: prediction

- Based on your estimated model, you can `predict` your outcome variable, e.g., math scores for each observation:

  ```
  predict math_hat
  ```

- You may use predicted values to calculate **residuals** by hand.
- Alternatively, use the option `residuals` with `predict`:

  ```
  predict resid_hat, residuals
  ```

! Residuals are important for model diagnostics.

Data Editor (Browse) - [hs0.dta]

File Edit View Data Tools

var20[1]

	math	math_hat	resid	resid_hat
1	41	54.18755	-13.18755	-13.18755
2	53	55.04541	-2.04541	-2.045409
3	54	41.62453	12.37547	12.37547
4	47	48.89786	-1.897858	-1.897857
5	57	54.18755	2.812447	2.812448
6	51	54.18755	-3.187553	-3.187552
7	42	58.81604	-16.81604	-16.81604
8	45	50.22028	-5.22028	-5.220281
9	54	57.49361	-3.49361	-3.493612
10	52	56.17119	-4.171188	-4.171188
11	51	50.22028	.7797203	.7797195
12	51	62.78331	-11.78331	-11.78331
13	71	59.47725	11.52275	11.52275
14	57	61.46088	-4.460884	-4.460883
15	50	57.49361	-7.49361	-7.493612

Postestimation: testing

- In the regression output, you find test statistics and p-values of some basic hypothesis tests, e.g., for statistical significance at the 5%-level for each individual coefficient.
- Use `test` followed by coefficient identifiers to conduct more complex hypothesis tests.
- You can test if a coefficient equals some value.
- You can test linear combinations of coefficients.
- You can test for joint significance of coefficients (F-tests).
- ...

```
qui regress math c.write i.gender, vce(robust)

test 2.gender=0.5

( 1)  2.gender = .5

       F(  1,   197) =   15.92
            Prob > F =    0.0001

test 2.gender=write

( 1)  - write + 2.gender = 0

       F(  1,   197) =   16.63
            Prob > F =    0.0001

test 2.gender write

( 1)  2.gender = 0
( 2)  write = 0

       F(  2,   197) =   72.35
            Prob > F =    0.0000
```

Postestimation: margins

- Use `margins` to calculate individual-level predictions (default) and predicted means.
- Specify `at()` option to predict for specific values of your variables.
- Specify `over()` option to calculate group-specific means.

! Differences in group means at specific values will be equal to the coefficient on the group dummy in simple linear models.

- Specify `dydx(varlist)` option to calculate marginal effects of your model.

! In non-linear models, use `at()` to determine where Stata should estimate your marginal effects.

```
. margins, at(write=50) over(gender)

Predictive margins
Model VCE: Robust

Expression: Linear prediction, predict()
Over:       gender
At: 1.gender
        write = 50
    2.gender
        write = 50
```

	Margin	Delta-method std. err.	t
gender			
1	52.86513	.7941956	66.56
2	49.0945	.6810005	72.09

```
. margins, dydx(write gender)

Average marginal effects
Model VCE: Robust

Expression: Linear prediction, predict()
dy/dx wrt:  write 2.gender
```

	dy/dx	Delta-method std. err.	t
write	.6612119	.0549737	12.03
2.gender	-3.770626	1.070417	-3.52

Non-linear models

- You may want to estimate non-linear models, e.g., when your outcome is a limited dependent variable.
- Good news: estimation simply requires a different `command` keyword.
- Let's estimate a non-linear model in which we test if writing scores and gender predict high math scores (above 75th percentile).
- The iterations show that Stata uses a numerical algorithm to estimate your model (here: Maximum likelihood).

! Use `margins` to estimate marginal effects which are not directly given by the estimated coefficients.

```
. quietly sum math, d

. gen math_high=(math>r(p75))

.
. logit math_high write i.gender, robust

Iteration 0:   log pseudolikelihood = -111.35502
Iteration 1:   log pseudolikelihood = -81.690505
Iteration 2:   log pseudolikelihood = -76.032423
Iteration 3:   log pseudolikelihood = -75.840565
Iteration 4:   log pseudolikelihood = -75.839916
Iteration 5:   log pseudolikelihood = -75.839916

Logistic regression                                     Nu
                                                        Wa
                                                        Pr
Log pseudolikelihood = -75.839916                       Ps
```

math_high	Coefficient	Robust std. err.	z	P>|z|	
write	.2371745	.0417822	5.68	0.000	
2.gender	-.8472694	.4252116	-1.99	0.046	-
_cons	-14.10938	2.454474	-5.75	0.000	-

Final note on panel data models

- Before estimating panel data models, use `xtset` to tell Stata that it currently stores panel data.
- After `xtset`ting your data, you can access `xtreg`, `xtlogit`, `xtologit`, etc. — usual `postestimation` options are available as well.
- To include individual fixed effects, specify the option `fe` within `xtreg`.