USEFUL CODES FOR STATA

"If you torture the data long enough, nature will confess"
R. Coase

IMPORTING DATA

```
clear all
insheet using "C:\Users\deloach\Documents\Class Stuff\Econometrics\cigarettes.csv", comma
```

To reshape data that is "wide" and convert it to "long" format for panel data:

Wide:

State	*un1990*	*un1991*	*…….pop1990*	*pop1991*
ALA	*4.5*	*5.5*	*2000*	*2056*

.

Long:

State	*year*	*un …….pop*	
ALA	*1990*	*4.5*	*2000*
ALA	*1991*	*5.5*	*2056*

```
insheet using "C:\Users\deloach\Desktop\Copy of Variables.csv", comma
reshape long unemployment population persinc persincpc, i(state) j(year)
```

BASIC OLS REGRESSIONS

To run a simple regression:

```
reg  pcigconspc priceconsumer
```

To run a regression and output the "Predicted Y" (P) and the residuals (R):

```
reg  cigconspc priceconsumer
predict p
predict r, residuals
```

To create a time series plot of the residuals (R):

```
twoway (tsline r);
```

To create a scatter plot of the residuals (R) vs. the predicted (P):

```
twoway(scatter r p)
```
or
```
twoway(scatter r  priceconsumer)
```

To do the univariate analysis:

```
sum cigconspc priceconsumer
```

To get summary stats for those variables in the regression:

```
sum cigconspc priceconsumer if e(sample)
```

To open a log for saving your work and printing:

```
log using "C:\Users\deloach\Documents\Class Stuff\Econometrics\STATA\log.smcl"
reg  cigconspc priceconsumer
log close
print "C:\Users\deloach\Documents\Class Stuff\Econometrics\STATA\log.smcl"
```

CREATING VARAIBLES

To add a new variable to your dataset:

```
gen lngdplab = log(gdpd/et)
gen lnkaplab = log(ktvd/et)
```

To run a regression and divide the data into different groups of years:

```
reg lnwage lgroom teage teage2 dwhite nohigh somecoll collegedeg married if dtesex==1
```

To perform a Chow test to see if there is a structural break or if two samples can be pooled:

```
reg lnwage age age2 white nohigh somecoll collegedeg male
reg lnwage age age2 white nohigh somecoll collegedeg if male==0
reg lnwage age age2 white nohigh somecoll collegedeg if male==1
```

Creating dummy variables (1,0) for a variable (ex: Gender; male=1, female=2, no response= -4). Dummy variables need to be 1 or 0, and "no response" observations should be coded as "." for TATA purposes.

```
gen male = 1 if gender == 1
replace male =0 if gender~=1
```

or

```
gen male = 1 if gender == 1
replace male =0 if male==.
```

Or, if the variable is coded as string, such as region of the country which is "South", "North" etc…

```
gen south = 1 if region == "South"
replace south =0 if region~="South"
```

DATA MANAGEMENT AND CLEANING

To see how many observations you have in each category (e.g., hhid by whether they have a bank in town):

```
table hhid bank
```

Deleting observations from dataset:

```
drop if lnwage ==0
```

Deleting observations from dataset if it is missing and defined as numeric:

```
drop if lnwage==.
```

Deleting observations from dataset if it is missing and defined as string (non-numeric):

```
drop if lnwage==""
```

To change a variable from string to numeric:

```
destring state, force replace
```

To change a set of observations when the variable is defined as string:

```
replace state =  "Alabama" if state == "ALA"
```

To change a set of observations when the variable is defined as numeric:

```
replace wage =  .  if wage= = 0
```

MERGING DATA

Merging two datasets where both have unique observations defined by two variables (e.g., pid and time):

master + using = merged result

pid	time	x1
14	1	0
14	2	0
14	4	0
16	1	1
16	2	1
17	1	0

pid	time	x2
14	1	7
14	2	9
16	1	2
16	2	3
17	1	5
17	2	2

pid	time	x1	x2	_merge
14	1	0	7	3
14	2	0	9	3
14	4	0	.	1
16	1	1	2	3
16	2	1	3	3
17	1	0	5	3
17	2	.	2	2

```
use "E:\master.dta"
merge 1:1 pid time using "E:\using.dta"
drop _merge
```

Merging two datasets where both have unique observations defined by two variables (e.g., pid and time):

master + using = merged result

pid	time	x1
14	1	0
14	2	0
14	4	0
16	1	1
16	2	1
17	1	0

pid	time	x2
14	1	7
14	2	9
16	1	2
16	2	3
17	1	5
17	2	2

pid	time	x1	x2	_merge
14	1	0	7	3
14	2	0	9	3
14	4	0	.	1
16	1	1	2	3
16	2	1	3	3
17	1	0	5	3
17	2	.	2	2

```
use "E:\master.dta"
merge 1:1 pid time using "E:\using.dta"
drop _merge
```

Merging two datasets where both have one variable in common but the master dataset has multiple observations for each of that unique variable (e.g., id across regions):

master + using = merged result

id	region	a
1	2	26
2	1	29
3	2	22
4	3	21
5	1	24
6	5	20

region	x
1	15
2	13
3	12
4	11
	5
	6

id	a	region	x	_merge
2	26	13		3
1	29	15		3
2	22	13		3
3	21	12		3
1	24	15		3
5	20	.		1
.	11	4		2

```
use "E:\master.dta"
merge m:1 region using "E:\using.dta"
drop _merge
```

To find duplicates in the dataset when STATA says "id and region" do not uniquely define an observation in a dataset:

```
duplicates list id region
```

To append one dataset to another (e.g., if you are adding years to a dataset where both datasets have the same variables):

```
use "E:\data2000.dta"
append using "E:\data2010.dta"
```

TESTING FOR BASIC ECONOMETRIC PROBLEMS

To run a regression calculating the VIF (variance inflation factor):

```
reg fert GDPgrow childlab femalelab school3 ratio
estat  vif
```

To perform a partial or marginal F test to test whether a right hand side variable can be excluded from the regression:

```
reg  price score cases
test cases
```

To run a regression calculating the Bruesch-Pagan LM test for heteroscedasticity:

```
reg  price score cases french imported red
predict r, residuals
gen r2 = r^2
reg  r2 score cases french imported red
```

or

```
reg  price score cases french imported red
estat hettest, iid rhs
```

To run a regression calculating White's heteroscedastically-correct standard errors:

```
reg fert GDPgrow childlab femalelab school3 ratio, robust
```

To run a Ramsey RESET test for mis-specification:

*this is done in two steps. First, run the model saving the residuals and predicted Ys to different file. In the second program, run an alternative model with Y squared and Y cubed as additional explanatory variables. Do a partial F test to see if the two variables you added are jointly significant.

```
reg cons gdp
predict p, predicted

gen p2 = p^2
gen p3 = p^3
```

*You have to save your dataset before running the next regression

```
reg cons gdp p2 p3
```

TIME SERIES

To test for first-order autocorrelation:

```
reg cons gdp
predict r, residuals

sort date
reg r l.r
```

To run a FEGLS regression that corrects for first-order autocorrelation using the Prais-Winsten method:

```
prais cons gdp
```

LOGISTIC REGRESSION

To run a logit regression in descending order (meaning SAS evaluates the larger dummy value (for a 1, 0 dummy and the 'yes' response is 1, SAS evaluates the probability of a 1, or 'yes' response))

```
logit abortion college sexed catho baptist working unemp noreas norcen south agepreg2
```

2SLS REGRESSION

NOTE: prior to running this code you need to run:

```
ssc install ranktest
ssc install ivreg2
```

To run 2SLS regression we first define the roles of each variable. There are 2 endogenous variables and the instruments are the exogenous ones. Then we define the structural equations (though for the first one we do not have to include the other endogenous variables only the IVs)

```
ivreg2 lnwage (lgroom = sat sun trtfamily eatdrink) teage teage2 dwhite nohigh somecoll
collegedeg married if dtesex==1, robust first
```

To perform a Hausman test for endogeneity (null is that the suspected variable is in fact exogenous) just add endogtest(varname) as an option

```
ivreg2 zhauk22 (smallmfi =  urban93 electricity  ) lr2hhfoodpc belowave , first
endogtest(smallmfi)
```

PANEL REGRESSIONS

To deal with panel data, we have to define the variables that keep track of the (1) cross section dimension and (2) the time dimension. This must be done before regressions can be run. The tsset statement defines that the individuals are denoted by the variable 'id" while the time is denoted by the variable "year2" and the option after the , tells it that the data is collected yearly.

To generate an id variable:

```
egen id = group(state)
```

To define the id and year variables:

```
tsset id year2, yearly
```

In this example we are estimating a difference model with robust std errors:

```
tsset id year2, yearly
xtreg d.zhauk22  d.smallmfi d.lr2hhfoodpc belowave, robust
```

In the example below, we are estimating a FE model with a condition:

```
tsset id year2, yearly
xtreg zweight mfi  lcomassets lassets  if mfloc93==0 , fe
```

In the example below, we are estimating a RE model with a condition:

```
tsset id year2, yearly
xtreg zweight mfi  lcomassets lassets  if mfloc93==0 , re
```

In the example below, we are estimating a FE model with an endogenous regressor:

```
tsset id year2, yearly
xtivreg zweight (mfi =  urban93 factory93) lcomassets lassets , first fe
```

To test for autocorrelation (AR(1)):

```
tsset id year2, yearly
xtreg zweight mfi  lcomassets lassets  if mfloc93==0 , fe
predict r, e
reg r l.r
```

MAKING TABLES

Using esttab or estout:

First run:

```
ssc install estout
```

After each regression we store the results in a file called ols, etc...after we are all done we use esttab or estout to create a custom table, in this case with only *twork* reported....the table will have beta, se, and ts

```
reg tstudy twork  tesex teage teage2 , robust
eststo ols

ivreg2 tstudy (twork = nonightwork avegasprices) tesex teage teage2, robust first
eststo tsls

esttab ols tsls using "u:\test.csv" , keep(twork) cells(b(star fmt(3)) se(par fmt(3))) replace
star(* 0.10 ** 0.05 *** 0.01) r2
```