Introduction to STATA

by Jonathan B. Hill
University of North Carolina – Chapel Hill

Example List **Page**

Data Explanation

This document makes repeated use of two datasets.

1. The first dataset is U.S. macroeconomics aggregates over 528 months from 1960-2006. The variables are

GDP	Gross Domestic Product
M1, **M2**	Money supply measures
IPI	Industrial Production Index
t_bill	90 day Treasury Bill rate

The Excel data file is available at ***www.unc.edu/~jbhill/gdp_m1_t_bill.xls***.

2. The second dataset is state-wide U.S. mortality rates. The variables are

mort	U.S. state-wide per mortality rate (per 100,000)
inc_pc	income per capita
pov	% residents at or below poverty line
ed_hs	% residents with high school education
ed2_coll	% residents with at least 2 years of college education
alc_pc	alcohol consumption per capita
tob_pc	tobacco consumption per capita
health_pc	health care expenditure per capita
phys	number of physicians per 100,000
urban	% residents in urban area
aged	% residents over age of 65

The Excel data file is available at *www**.unc.edu/~jbhill/mort.xls***.

1. GETTING STARTED

Boot-up STATA

The are 4 “windows” which specialize in task/output

Stata Command

In this window everything typed, followed by “enter”, is performed by STATA.

A record of what was typed in listed in the **Review** window

The output generated from you command is sent to the **Stata Results** window

Review

STATA stores a list of all commands typed in the Command window.

If you click on one of the commands recorded in Review, the list reappears in Command. Hit “enter” and the command is performed anew.

Print information from the **Review** window by **file, print results.**

Stata Results

This window contains all output and STATA commentary of you commands. Errors are logged here: STATA generates a running commentary on everything you ask it to do, including the command typed and whether it could perform the command.

Variables

This window lists variables loaded into STATA. The data itself is stored in the **Data Editor**.

EXAMPLE #1 (help)

TASK:
We want help from STATA.

OPERATION:
Type **help** and "enter" in *Command*: STATA offers basic information in *Results*.

Type **help sort** and "enter"– STATA explains the "sort" command. Etc.

Type **tutorial intro** – STATA provides a detailed introduction to STATA in *Results*. The introduction includes data provided by STATA for practice.

---

2. LOADING DATA (Command Window)

In this section details on using the *Command* window to *load data* are presented.

2.1 Preliminaries: Copy/Paste into Data Editor, Delete, Change Name

Data is stored in the **Data Editor**.

Data Editor

From the toolbar: **Windows**, **Data Editor**. A pop-up window appears.

Boot-up Excel or Notepad (wherever the data is), open any file you wish to analyze.

For convenience, *assume data names are in the first row*.

Copy/paste the data (and names) into the **Data Editor**.

STATA will ask you if the first row contains variable names.

Close the **Data Editor** – click on the cross in the upper right corner.

The variables names appear in the **Variables** window.

No Names

If you do not provide names, STATA uses "**var1**", "**var2**", etc.

Change Names

In the **Data Editor**, double click anywhere in the column of a variable whose name you want to change. In the pop-up window, change the name.

Variable names are **CASE SENSITIVE**. Thus, ed ≠ ED ≠ Ed ≠ eD.

Create New Variables

generate Command for creating new variables from existing ones.

We want *health care expenditure squared*, and a dummy for high school educatedness being greater than 70%.

```
generate health_2 =health_pc^2
generate ed_hs_70 = ed_hs > .70;
```

Create a lag on GDP, we use

```
generate GDP_lag1 = GDP[_n-1]
```

Create a time trend or sequence of integers:

egen t = seq(), from(##) to(##) the empty "()" is required; from(##) < to(##).

gen t = _n does the same thing with the restriction $t = 1,2,3,...,n$ the sample size.

Save as Stata Data *filename.dta*

The variables in the **Variables** window can be saved as a "**.dta**" file.
From the toolbar: **File, Save As,** select **.dta**, and type the *filename*.

Delete/Drop

Delete Variables – from the toolbar, **Window**, **Data Editor**, then highlight the column of data you want deleted, click **delete**.

Delete Variables – from **Command** type

drop *variablename*.

Replace

You can "re-generate" a variable: if you want to use the variable name "*ed_2*" for something else, either **drop** it, or re-define it with **replace**. For example:

replace ed_2 = ed*sex

Scalars

In all of the above cases STATA generates a vector of sample size *n*. If you want a scalar,

Scalar *scalarname* = *somevalue*

Now the new scalar *scalarname* can be used anywhere else. For example,

Scalar *ed_scale* = 2
generate ed_2 = *ed_scale* *ed

2.2 Change the directory and re-load data

In order to open a stata data file conveniently, first tell stata where all of you files are. This is done by *changing the directory*.

Change Directory – If you have stata files in **c:\temp\somefolder**, in **Command** type
cd c:\temp\somefolder

and "enter".

To load stata data (See 2.3, below, for creating a stata data file) from **c:\temp\somefolder**, in **Command**

use *filename*

If the file is a stata data file (i.e. .dat), then you do not need to add the tag ".dta".

If the data is an Excel file, either copy/paste into the **Data Editor**, or type in **Command**

insheet *filename.xls*

For **help** on loading different *file types*,

infile help

2.3 Saving from Command

Type **save *filename***

Stata will save it as a ".dta" file. If the file already exists, then *to over-write*

save *filename*, replace

2.4 Clearing data

In order to load a new set of variables, the Variables window must be cleared. Type **clear**

EXAMPLE #2 (copy/paste data from Excel, create a stata data file, re-load):

TASK:
We want to create a stata data file from an Excel file, data_1.xls, and store the stata data file in c:\temp\stata_files. The Excel file contains variables named *money*, *income*, and *t_bill*.

Assuming we clear the data, we want to re-load the stata file.

We want a dummy variable for *t_bill* > 4.

Once we re-load the stata data file, we want to delete *t_bill*.

OPERATIONS:
Open data.xls in Excel. Copy all relevant rows and columns, including variable names.

In STATA's **Command** window, type **clear** and "enter".

Paste the data into STATA's **Data Editor**.

Close the Data Editor. The Widow Variables now contains *money, income, t_bill*.

In **Command** type **cd c:\temp\stata_files** to change the default directory.

From the *toolbar*, **File, Save As,** and **data_1**.

Suppose we want to clear all variables: in Command, **clear**.

To re-load, in **Command** type **use data_1.**

Finally, we want to create a dummy variable and delete *t_bill*. In **Command**

generate t_bill > 4
drop t_bill.

---

3. DATA ANALYSIS (Command Window)

All commands in **lower-case bold** are understood to be typed into **Command**.

Assume a stata data file has been loaded with the variables *GDP, M1, M2, IPI,* t_*bill*, where *IPI* is the industrial production index.

3.1 Summary Statistics: SUMMARIZE, MEANS, CI

summarize Reports number of observations (less missing values), sample means, standard deviations, minima, maxima for *all variables* in the Variables Window.

Recall the formulae for the sample *mean*, *variance* and *standard deviation*:

$$\text{Sample Mean} \quad : \quad \overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$$

$$\text{Sample Variance} \quad : \quad \hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2$$

$$\text{Sample Stand. Dev.} \quad : \quad \hat{\sigma} = \sqrt{\hat{\sigma}^2}$$

summarize var1 var2… Reports information for specified variables.

Hint: Instead of typing the variables, just *click on their name* in the **Variables Window**.

Hint: If you want to repeat a command, click on it the previous execution from the **Review Window**.

summarize , detail Reports above information *plus* lower quantiles, skewness, kurtosis, etc., as well as the standard statistics. **Include the comma “,”**

Recall the formulae for *skewness* and *kurtosis*:

$$\text{Skew} \quad : \quad S = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{X_i - \overline{X}}{\hat{\sigma}}\right)^3$$

$$\text{Kurtosis} \quad : \quad \kappa = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{X_i - \overline{X}}{\hat{\sigma}}\right)^4$$

means Reports arithmetic, geometric, and harmonic sample means and their 95% confidence bands.

ci Reports arithmetic means and the 95% confidence bands.

ci var1 var2…, level (##) same, but for ##% confidence bands. **Include the comma ","**

Recall the formula for a (1 - α)% *confidence band* of the *sample mean* of an iid X_i:

$$\overline{X} \pm Z_{\alpha/2} \times \hat{\sigma} / \sqrt{n} \quad \text{where } P\left(|X_i| < Z_{\alpha/2}\right) = 1 - \alpha/2 \text{ for normally distributed } X_i$$

Stata Results for <u>Summarize</u>:

.summarize

Variable	Obs	Mean	Std. Dev.	Min	Max
GDP	528	3907.111	1537.238	1590.6	7121.5
M1	528	576.6426	83.75479	458.7388	775.5485
M2	528	2000.259	527.2667	988.1655	3189.297
IPI	528	65.68998	23.53514	27.977	116.202
t_bill	528	6.034508	2.653475	1.45	15.92

Stata Results for <u>SUMMARIZE, DETAIL</u>:

. summarize GDP, detail

GDP

Percentiles	Smallest			
1%	1624.5	1590.6		
5%	1687.3	1598.2		
10%	1855.7	1609.7	Obs	528
25%	2573.45	1620.2	Sum of Wgt.	528
50%	3700.45		Mean	3907.111
	Largest		Std. Dev.	1537.238
75%	5100.85	7073.9	ariance	2363102
90%	6210.5	7079.8	Skewness	.2819017
95%	6686.5	7100.9	Kurtosis	2.001144
99%	7057.4	7121.5		

Stata Results for CI:

```
. ci GDP M1
```

Variable	Obs	Mean	Std. Err.	[95% Conf.	Interval]
GDP	528	3907.111	66.89972	3775.688	4038.533
M1	528	576.6426	3.644959	569.4822	583.8031

Stata Results for MEANS:

```
. means M2 t_bill
```

Variable	Type	Obs	Mean	[95% Conf.	Interval]
M2	Arithmetic	528	2000.259	1955.181	2045.337
	Geometric	528	1924.908	1878.413	1972.555
	Harmonic	528	1843.873	1795.952	1894.421
t_bill	Arithmetic	528	6.034508	5.807654	6.261361
	Geometric	528	5.511916	5.313327	5.717928
	Harmonic	528	5.017617	4.823927	5.227511

3.2 Storing Sample Mean, Variance, Sample Size, etc. : EGEN

The commands **summarize**, etc., only display sample information. There are numerous situations when we want to use the sample mean, variance, sample size, etc., within a program. Thus, we want to **generate** a variable containing means, variances, the sample size, the number of missing observations, etc. For this, we use **egen**.

egen Generates variable with descriptive statistic information.

Sample Size **egen n = count(GNP)**

stores the number of GNP observations (the sample size) in "*n*". This new variable appears in the **Variables** list. Only *one variable* can be analyzed at a time.

Sample Means **egen GNP_mu = mean(GNP)**

stores the sample mean of GNP in "*GNP_mu*".

Sample Stand. Dev. **egen GNP_s = sd(GNP)**

store the sample variance, and so on.

Mode, median, minimum, maximum, skewness, Kurtosis, standard deviation, sum

egen then **mode, med, min, max, skew, kurt, sd, sum**

3.3 Displaying a Variable : LIST

If you want to display a variable in the **STATA Review** window, use **list**.

We can display GDP by

list GDP

3.4 Correlation Analysis: CORRELATE, PWCORR, SPEARMAN

Recall the sample correlation coefficient for two random variables $\{X_i, Y_i\}$:

$$r_{x,y} = \frac{1}{n}\sum_{i=1}^{n}\left(X_i - \overline{X}\right)\left(Y_i - \overline{Y}\right)$$

There are three ways to produce sample correlations: **correlate, pwcorr, spearman**

correlate produces correlations across all pairs.

correlate var1 var2… produces correlations across all pairs of var1, var2,…

correlate var1 var2… , means does the above, plus means, stand. dev.'s, min's, max's.

pwcorr simply reports all *pair-wise correlations* in the Variables Window.

If you want significance levels of the correlations, use **pwcorr** (this doubles as a test)

pwcorr var1 var2…, sig be sure to type the comma ","

A test of zero correlation between exactly two variables is performed by **sprearman**:

spearman var1 var2 only two variables at a time!

Stata Results <u>PWCORR</u>:

```
. pwcorr
             |      GDP       M1       M2      IPI    t_bill
-------------+-------------------------------------------------
         GDP |   1.0000
          M1 |   0.7893   1.0000
          M2 |   0.9691   0.7555   1.0000
         IPI |   0.9866   0.7627   0.9508   1.0000
      t_bill |   0.0400  -0.2857   0.0842   0.0442   1.0000
```

Stata Results PWCORR, SIG:

```
. pwcorr GDP M1 t_bill, sig
```

	GDP	M1	t_bill
GDP	1.0000		
M1	0.7893	1.0000	
	0.0000		
t_bill	0.0400	-0.2857	1.0000
	0.3588	*0.0000*	

The *p-values* of tests of corr = 0 are below each sample correlation, in *italics*. Notice *t_bill* is not directly correlated at any significance level with *GDP* and *M1*.

Stata Results SPEARMAN:

```
. spearman GDP  M1
```

```
Number of obs =     528
Spearman's rho =     0.7579

Test of Ho: GDP and M1 are independent
Prob > |t| =       0.0000
```

Thus, GDP and M1 are significantly correlated (at any level of significance).

Stata Results CORRELATE, MEANS:

```
. correlate GDP M1, means
```

Variable	Mean	Std. Dev.	Min	Max
GDP	3907.111	1537.238	1590.6	7121.5
M1	576.6426	83.75479	458.7388	775.5485

	GDP	M1
GDP	1.0000	
M1	0.7893	1.0000

3.5 Normality Tests (tests of any distribution) : SKTEST, KSMIRNOV

We want to test whether

$$X_i \sim N\left(\mu, \sigma^2\right)$$

Two popular tests are the **Jarque-Bera** test exploits sample skewness and kurtosis as a test of whether a variable is drawn from a normal distribution. In particular, the **Jarque-Bera** exploits the fact that a standard normal random variable has zero skewness and kurtosis of 3. Therefore, if X_i is normally distributed then

$$S = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{X_i - \overline{X}}{\hat{\sigma}}\right)^3 \approx 0 \quad \text{and} \quad \kappa = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{X_i - \overline{X}}{\hat{\sigma}}\right)^4 \approx 3.$$

The **Kolmogorov-Smirnov** test compares the empirical distribution function of the standardized data to the standard normal distribution.

sktest var1 var2…

Tests the standardized GDP skewness (a standard normal has zero skewness), kurtosis (a standardized normal has 3 for kurtosis), and it tests both (i.e. the Jarque-Bera test of normality).

ksmirnov GDP = normprob(GDP)

Performs the KS-test of normality. The data should be standardized since the test is, by default, for standard normality. Simply generate the mean and standard deviation and use as follows.

EXAMPLE #3: Kolmogorov-Smirnov on GDP

TASK:
We want tom test whether U.S. GDP is drawn from a normal distribution. We will perform the KS test of standard normality. Therefore we need the sample mean and standard deviation.

OPERATION:
Create the mean and standard deviation of GDP as variables GDP_mu and GDP_s

egen GDP_mu = mean(GDP)
egen GDP_s = sd(GDP)

Do the KS test on standardized GDP:

ksmirnov GDP = normprob((GDP-GDP_mu)/GDP_s)

STATA RESULTS:

```
. ksmirnov GDP = normprob((GDP-GDP_mu)/GDP_s)
```

One-sample Kolmogorov-Smirnov test against theoretical distribution:
normprob((GDP-GDP_mu)/GDP_s)

Smaller group	D	P-value	Corrected
GDP:	0.0729	0.004	
Cumulative:	-0.0659	0.010	
Combined K-S:	0.0729	0.007	0.006

Inspect *CombinedK-S*: the p-value < .01, so we reject the null of normality.

3.6 Plotting data (plot, scatter, twoway, tsline)

3.6.1 Scatter Plots : GRAPH, TWOWAY, PLOT

graph twoway *scatter* Y X produces a nice scatter plot (Note the first variable listed is on the Y axis!!!!)

The graph is displayed in a separate graph window.

graph matrix GDP M1 M2 IPI produces a matrix of two-by-two scatter plots across all variables.

graph matrix GDP M1 M2 IPI, half produces only the lower half (only one of each scatter plot)

"**gr**" is the same is "**graph**"

graph twoway *histogram* GDP produces a *histogram*.

plot GDP M1 is same as graph, except STATA produces to plot in the **Results** window.

SAVE and PRINT GRAPHS from the STATA GRAPH WINDOW

Once a graph is produced, click-on it, then from the main toolbar **FILE**, **SAVE GRAPH**, or **PRINT GRAPH**.

In the graph window, click on **File**, **Start Graph Editor** to edit (labels, titles).

Copy graphs for pasting into WORD, etc., by clicking on the graph, **EDIT**, **COPY GRAPH**.

3.6.2 Histograms and Distribution Plots : GRAPH, KDENSITY

histogram GDP	produces a histogram
histogram GDP, bin(10)	produces a histrogram with 10 X-axis ranges (must be 2-50).
histogram GDP, normal	produces a historgram and overlays a normal distribution with GDP's sample mean and variance.
kdensity GDP	produces a sample approximation of the distribution of GDP.
pnorm GDP	plots empirical probabilities of standardized GDP against probabilities of a normal.

histogram GDP, title("U.S. Monthly GDP") adds a title

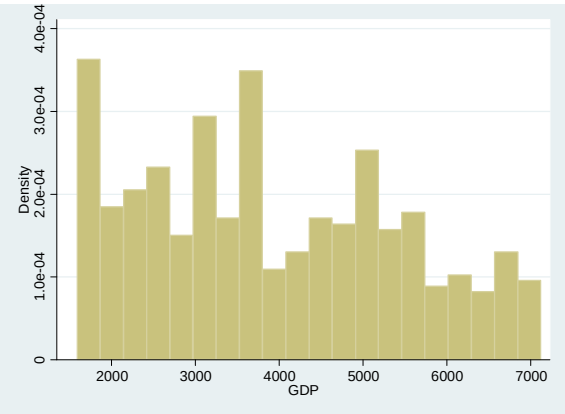
3.6.3 Line Plots : TSLINE

If the data occur over time then STATA allows a line plot. This requires telling STATA the data are a time series.

gen t = _n	creates a time variable $t = 1,2,...,n$ that requires the understand your data occurs over time
tsset t	tells STATA that the entire data set is a time series.
tsline GDP	a line plot of GDP

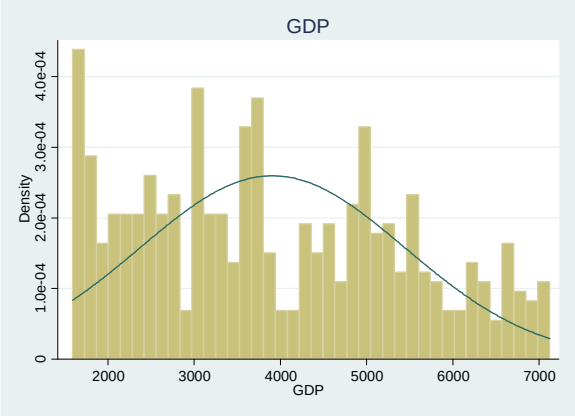
Stata Results from GRAPH:

.histogram GDP, bin(20)

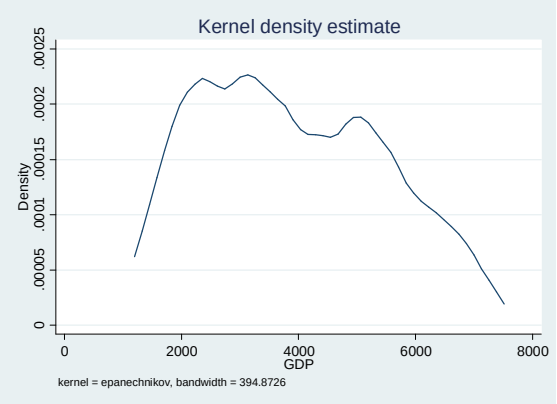


Stata Results GRAPH, NORMAL, TITLE:

. histogram GDP, bin(40) normal title("GDP")



Stata Results from KDENSITY:



Stata Results from TSLINE:

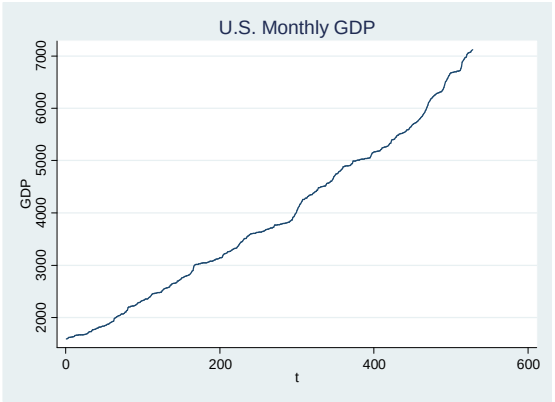
. gen t = _n **(creates time variable *t* = 1,2,...,*n*)**

. tsset t **(tells STATA the entire dataset is a time series)**

time variable: t, 1 to 528

delta: 1 unit

. tsline **(line plot)**



3.6.4 Titles, Axes, Lines, etc.

The following is for graphs that appear in the STATA Graph pop-up window.

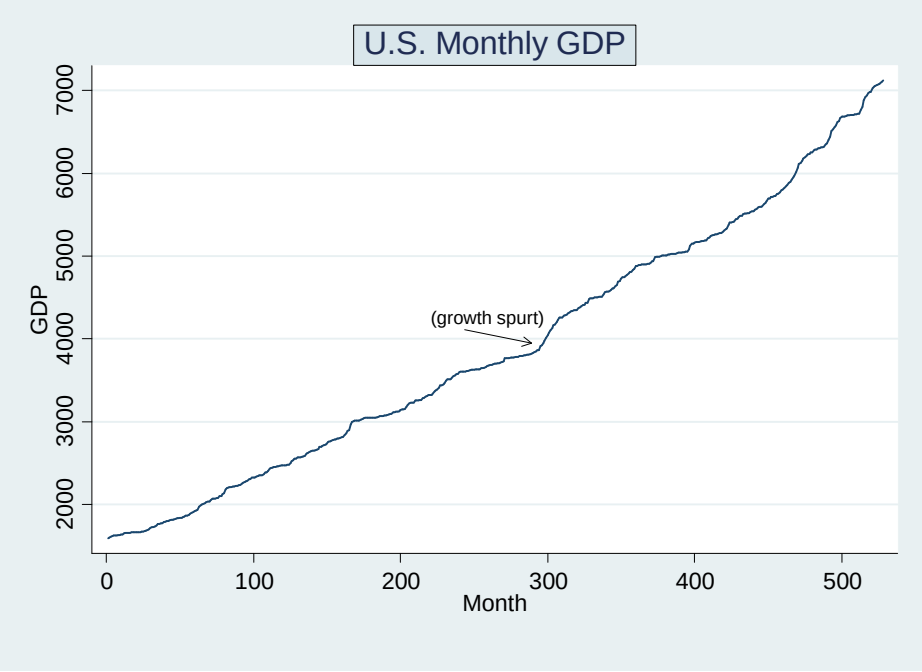
1. **Save**: **File**, **Save** (give it a logical name).
2. **Copy/Paste**: **Edit**, **Copy**, then go to a Word document, etc., to paste.
3. **Title**: **File**, **Start Graph Editor**, **Graph**, **Titles**, etc. You can add a title, put a box around it, and so on.

   If you want to edit an existing title, *double click it*.
4. **Axes**: Click an axis label/title in order to change it.
5. **Notes**, etc.: You can add text boxes (i.e. "notes"), lines, arrows and many other things...

When you are done, **File, Stop Graph Editor.**

Stata Results from <u>TSLINE:</u>

I added a box round the title, changed the X-axis label, added a note, and arrow.



3.7 Hypothesis Testing : TTEST, SPEARMAN, PRTEST, SDTEST

ttest Performs t-test of mean (one sample, or difference in means in two samples). Returns p-values for one-sided and two-sided tests.

Test % adults with high school education is at least 40%:

ttest ed_hs = .4

spearman Performs Spearman's correlation test of independence.

Test for independence between tobacco consumption and high school educatedness:

spearman tob_pc ed_hs

prtest Tests means in one binary, or between two binary series. Thus, it tests proportions (how frequently something occurs).

Test for whether only a few states have at least 80% of adults with a high school education:

generate ed_hs_80 = ed_hs > .8
prtest ed_hs_80 = .2

sdtest Test of variance, or equality of two variances.

Stata Results from TTEST:

```
. ttest ed_hs = .4

One-sample t test
------------------------------------------------------------------------------
Variable |     Obs        Mean    Std. Err.   Std. Dev.   [95% Conf. Interval]
---------+--------------------------------------------------------------------
   ed_hs |      51    .6746078    .0104936    .0749392    .6535309    .6956848
------------------------------------------------------------------------------
Ho: mean(ed_hs) = .4

Ha: mean < .4               Ha: mean ~= .4                  Ha: mean > .4
t =  26.1691                t =  26.1691                    t =  26.1691
P < t =   1.0000            P > |t| =   0.0000              P > t =   0.0000
```

The results imply a one-sided test *mean(ed_hs)* ≤ .4 against *mean(ed_hs)* > .4 is rejected: the *p-value* is roughly 1.00.

Stata Results from SPEARMAN:

```
. spearman tob_pc ed_hs

Number of obs =      51
Spearman's rho =     -0.3912

Test of Ho:        tob_pc and ed_hs are independent
Prob > |t| =       0.0045
```

Tobacco consumption and adult high school educateness are significantly negatively correlated (*corr* = -.39, *p-value* = < .01).

Stata Results from PRTEST:

```
. generate ed_hs_80 = ed_hs> .8
. prtest ed_hs_80 = .2

One-sample test of proportion              ed_hs_80: Number of obs =       51
------------------------------------------------------------------------------
Variable |      Mean        Std. Err.       z     P>|z|      [95% Conf. Interval]
---------+--------------------------------------------------------------------
ed_hs_80 |  .0392157    .0271805   1.44279   0.1491     -.0140572    .0924885
------------------------------------------------------------------------------
Ho: proportion(ed_hs_80) = .2

Ha: ed_hs_80 < .2          Ha: ed_hs_80 ~= .2                 Ha: ed_hs_80 > .2
     z = -2.871                 z = -2.871                      z = -2.871
     P < z = 0.0020             P > |z| = 0.0041                P > z = 0.9980
```

We want to test if *at least* 20% of states have at least 80% adult high school educatedness.
We test H0: *mean(ed_hs_80)* ≥ .2 against H1: *mean(ed_hs_80)* < .2.

The *p-value* < .01, hence we reject the null at the 1% level.

Stata Results from SDTEST:

```
. sdtest ed_hs= ed2_coll

Variance ratio test
------------------------------------------------------------------------------
Variable |     Obs        Mean    Std. Err.   Std. Dev.   [95% Conf. Interval]
---------+--------------------------------------------------------------------
   ed_hs |      51    .6746078    .0104936    .0749392    .6535309    .6956848
ed2_coll |      51    .1631569     .004679    .0334146    .1537589    .1725549
---------+--------------------------------------------------------------------
combined |     102    .4188824    .0260798     .263393    .3671471    .4706177
------------------------------------------------------------------------------
Ho: sd(ed_hs) = sd(ed2_coll)

F(50,50) observed   = F_obs          =    5.030
F(50,50) lower tail = F_L   = 1/F_obs =    0.199
F(50,50) upper tail = F_U   = F_obs   =    5.030

Ha: sd(1) < sd(2)        Ha: sd(1) ~= sd(2)                 Ha: sd(1) > sd(2)
P < F_obs = 1.0000     P < F_L + P > F_U = 0.0000       P > F_obs = 0.0000
```

We want to test if college educatedness has at least the same variance as high school educatedness: H0: *var(ed2_coll)* ≥ *var(ed_hs)* against H1: *var(ed2_coll)* < *var(ed_hs)*. The p-value < .01, so we reject at the 1% level.

4. LINEAR REGRESSION (from the Command Window)

4.1 Ordinary Least Squares : REGRESS

We want to estimate the relationships between a dependent variable of interest *y* and available explanatory variables *x*. The regression model is

$$y_i = \beta_1 + \beta_2 x_{i,2} + ... + \beta_k x_{i,k} + \varepsilon_i, \quad \varepsilon_i \overset{iid}{\sim} (0, \sigma^2)$$

Consider explaining U.S. state-wide mortality rates using income, tobacco and alcohol sales, high and college education.

In the **Command Window**, type

regress mort inc_pc ed_hs ed2_coll alc tob

STATA performs OLS by regressing *mort* on *inc_pc*, etc.

A constant term is automatically included.

Stata Results from <u>REGRESS:</u>

```
. regress  mort inc_pc ed_hs ed2_coll alc tob
```

```
Source |       SS       df       MS              Number of obs =      51
-------------+------------------------------   F(  5,    45) =    4.74
Model  | 328170.058     5   65634               Prob > F      =  0.0015
Residual | 623560.176    45  13856               R-squared     =  0.3448
-------------+------------------------------   Adj R-squared =  0.2720
Total  | 951730.234     50  19034               Root MSE      =  117.72
```

mort	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
inc_pc	.0241829	.0120987	2.00	0.052	-.0001851	.048551
ed_hs	-876.7612	306.6727	-2.86	0.006	-1494.432	-259.0906
ed2_coll	-1449.301	911.4982	-1.59	0.119	-3285.153	386.5503
alc	6.777576	32.47724	0.21	0.836	-58.63495	72.1901
tob	.0186199	.992006	0.02	0.985	-1.979383	2.016622
_cons	1341.904	233.8229	5.74	0.000	870.9602	1812.847

Notation

SS = sum of squares.
"model SS" is the sum of squares of Y, in this case "mort".
"Residual SS" is the sum of squared residuals.
P>|t| is the p-value for the t-statistic: t = Coef/Std.Err.
F(…) is the classical F-test statitic.
Prob > F is the p-value of the F-statistic.

4.2 No-Constant, Repeat, Suppress: BETA, NOCONSTANT, QUIETLY

1. If you want STATA to ***repeat the last regression***, use

   regress, beta the last regression output appears again.

2. If you want to omit the *constant* term, $y_i = \beta_2 x_{i,2} + ... + \beta_k x_{i,k} + \varepsilon_i$ use **noconstant**:

   regress mort inc_pc ed_hs ed2_coll alc tob, noconstant

   Notice the comma ","! That tells STATA "nonconstant" is not a variable, like "tob"!

3. If you want to perform OLS without immediate output, use **quietly regress**:

   quietly regress mort inc_pc ed_hs ed2_coll alc tob

   STATA performs OLS and stores the results for testing, forecasting, etc.

   Hint: The command **quietly** can be used to suppress output in general, and not simply for OLS[1].

4.3 Listing, Storing Regression Output: ERETURN LIST, HOLD, UNHOLD

1. If you want to produce a compact list of all **regress** output, use **ereturn list**:

   regress mort inc_pc ed_hs ed2_coll alc tob
   ereturn list

   STATA describes the regression output, and lists the names it gives various statistics.

2. In order to store regression output for later use, use **_estimates hold.** For example:

   regress mort inc_pc ed_hs ed2_coll alc tob
   _estimates *hold* mort_model_1

   regress mort inc_pc ed_hs ed2_coll alc tob, noconstant
   _estimates *hold* mort_model_no_const

   _estimates *unhold* mort_model_1 (you can only **unhold** once)
   test tob=alc

   Two models are estimated (one without a constant term), and the output is stored under two names, *mort_model_1* and *mort_model_noconst*. You can recall a model by **unhold**. In this case the first model's results are recalled and a test of slope equality is performed.

[1] If you want to be sure appears, use **noisily**. But note simply NOT using **quietly** suffices to ensure output.

If you forgot the model names you used for **_estimates hold**, use **_estimates dir** which lists all **_estimates hold** model names.

Once you **unhold** a regression model it not longer exists under **hold**.

Stata Results <u>ERETURN LIST:</u>

```
. ereturn list
```

scalars:

```
e(N) =  51
e (df_m) =  5
e(df_r) =  45
e(F) =  4.736560541236416      /* the F-statistic  */
e(r2) =  .3448141568639046     /* the R^2  */
e(rmse) =  117.7153040090269
e(mss) =  328170.0582542721
e(rss) =  623560.1759071931
e(r2_a) =  .272015729848783     /* the adjusted R^2  */
e(ll) =  -312.3559256484332      /* the log-likelihood  */
e(ll_0) =  -323.1382526655355  /* the log-likelihood when slopes = zero */
```

macros:

```
e(depvar) : "mort"
e(cmd) : "regress"
e(predict) : "regres_p"
e(model) : "ols"
```

matrices:

```
e(b) :  1 x 6                  /* the stored regression slopes b */
e(V) :  6 x 6                  /* the stored sample covariance matrix if b */
```

functions:

```
e(sample)
```

4.4 Generating Residuals and Predicted Values: PREDICT

In order to produce regression predicted values, after **regress…** type

predict *varname*

As always, ***varname*** can be any name you want. Example: y_**hat**.

STATA automatically creates the predicted values $\hat{\beta}' X_i$ and gives it your chosen name. The new variable is now visible in the **Variables** window.

For standard errors of the predicted values $\hat{V}[\hat{\beta}' X_i \mid X_i]$, type

predict *varname_for_st_err*, stdp

For residuals and standard error of residuals, type

predict *varname*, resid stores the residuals in *varname*

predict *varname_for_st_err_e*, stdr stores the standard error of the predicted value

4.5 Plotting Regression Results : AVPLOT, AVPLOTS, RVPPLOT

There are two useful sets of information after estimating a regression model: how important is each regressor $x_{i,j}$? and how noisy are the residuals? Clearly superfluous regressors should be excluded, and we require the true errors ε_i to be independent (of each other, and of the regressors $x_{i,j}$).

1. Added Variable Plots – **AVPLOT, AVPLOTS**

   STATA can produce scatter plots demonstrating the marginal relationship of the j^{th} regressor $x_{i,j}$ on y_i, after controlling for the associations between the regressors.

   Produce one plot for one regressor with **avplot**:

   regress y x1…xk
   avplot *varx* (list one regressor)

   Produce plots for each regressor with **avplots**:

   regress y x1…xk
   avplots (default all regressors)

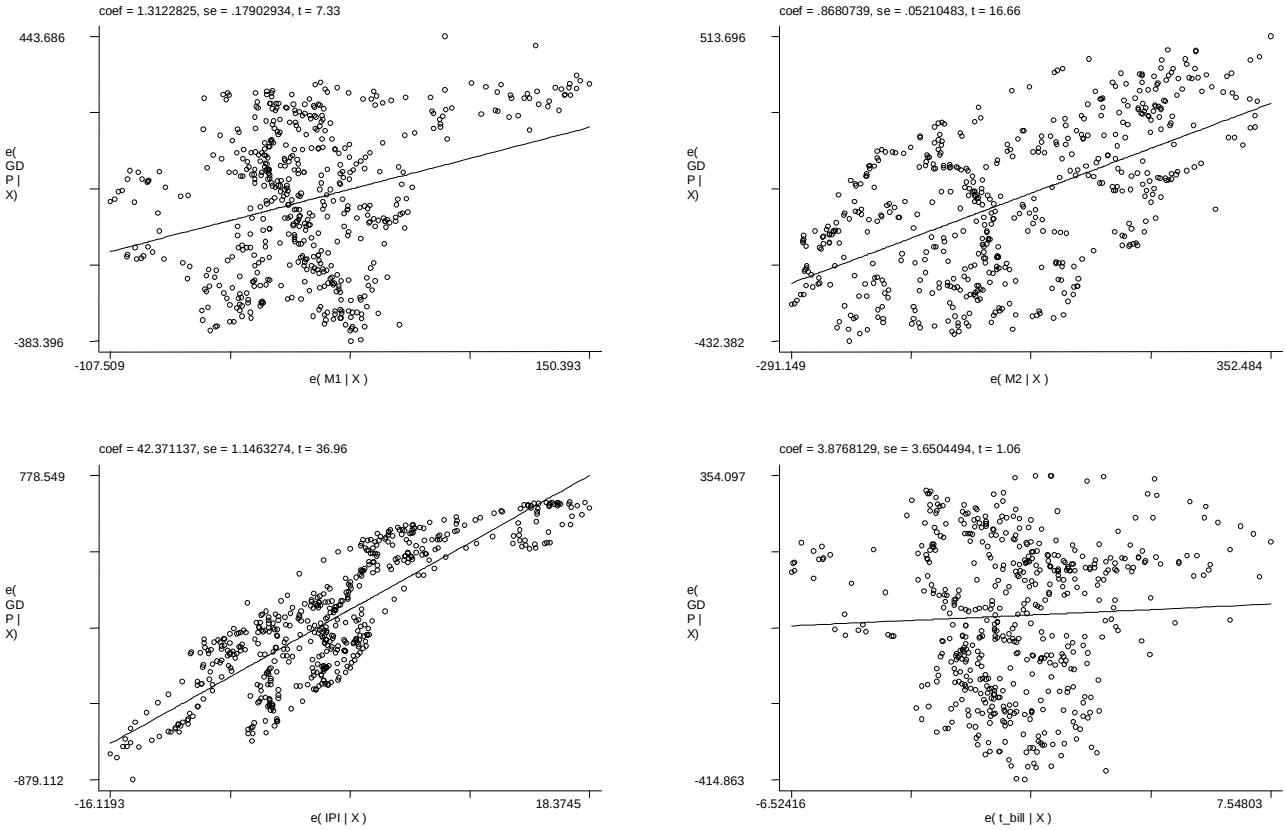
2. Residuals Plots - **RVPPLOT**

   We do the same thing as **avplot** with residuals via **rvpplot**. Since the true errors ε_i should be independent of each other, and of the regressors $x_{i,j}$, any pattern suggests model mis-specification or endogeniety.

   regress y x1…xk
   rvpplot *varx* (list one regressor)

Stata Results <u>AVPLOTS:</u>

```
. quietly regress  GDP M1 M2 IPI t_bill
. avplots
```

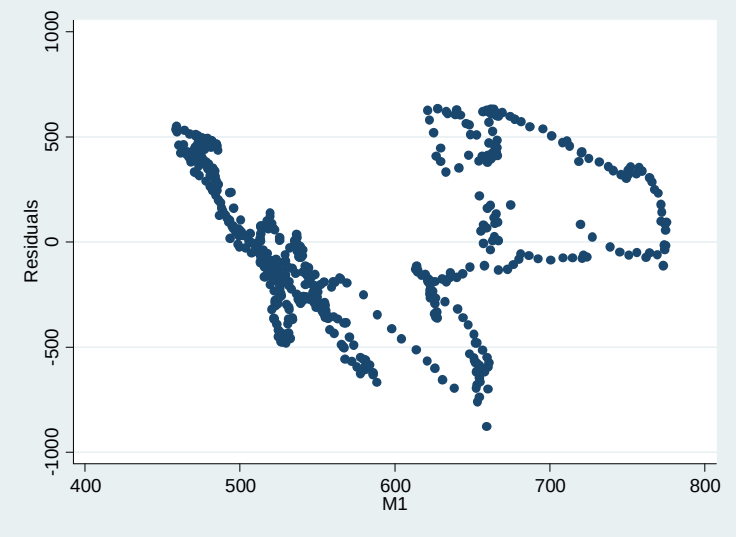


The X-axis contains the regressor x after controlling for its relationship with *other* regressors; the Y-axis contains GDP after controlling for its relationship with the *remaining* regressors.

A clear, sharp positive or negative slope is indicative of a significant, pure relationship between regressor and DGP. Consider IPI (production index) and GDP: obviously positively associated!

Stata Results <u>RVPPLOT:</u>

```
. quietly regress  GDP M1 M2 t_bill
. rvpplot M1
```



No apparent pattern suggests the regression error is not related to M1.

4.6 Test Linear Equality Constraints - TEST

test performs a small-sample "F" test (the help screen incorrectly claims the test is an asymptotic Wald test).

Test *ed_hs* and *ed2_coll* jointly have zero slopes:

test ed_hs ed2_coll

The default test is a test that each associated parameter is zero.

We can test *ed_hs* + *ed2_coll* = 0 (e.g. a high school may add to the mortality rate, but a college education exactly offsets the effect, on average):

test ed_hs + ed2_coll = 0

Test *ed_hs* and *ed2_coll* have the same marginal effect on mortality:

test ed_hs = ed2_coll

Test whether a parameter equals a specific value:

test inc_pc = .025

Tests can be performed symbolically, where **_b[*varname*]** denotes the parameter:

test _b[ed_hs] = _b[ed2_coll]

is identical to

test ed_hs = ed2_coll

The constant term is denoted **_cons**. Thus, we test if the constant = 1000 by

test _cons = 1000

4.7 Test Nonlinear Equality Constraints

A test that ed_hs and ed2_coll have the same marginal effect can be performed as

testnl _b[ed_hs]/_b[ed2_coll] = 1 : this also tests for parameter identity

Notice it must be typed with the **_b[…]** notation (laborious!).

Stata Results <u>TESTNL:</u>

```
. testnl _b[ed_hs]/_b[ed2_coll]=1

(1) _b[ed_hs]/_b[ed2_coll]=1

           F(1, 45) =       0.66
           Prob > F =       0.4209
```

4.8 Test of Omitted Variables/Structure

In order to test for omitted variables, or omitted structure (e.g. quadratic terms, or natural log for growth), the Ramsey RESET (*Regression Specification Error Test*) command is popular (but very weak!). Use

ovtest

This tests for omitted structure by including powers of the predicted values (which are functions of the X's).

The command

ovtest, rhs

performs the RESET test based *only on powers of the regressors*.

Stata Results <u>OVTEST</u> for mortality model:

```
. ovtest
Ramsey RESET test using powers of the fitted values of mort
       Ho:  model has no omitted variables
       F(3, 42) =      1.66
             Prob > F =      0.1912 (this provides evidence the model is well specified)

. ovtest, rhs
Ramsey RESET test using powers of the independent variables
       Ho:  model has no omitted variables
             F(15, 30) =      5.35
       Prob > F =      0.0000   (this provides evidence of mis-specification)
```

Since the first test's p-value is not that large (above 5%, but not huge), the two tests together suggest additional powers of the regressors (e.g. *inc_pc*^{2}) may improve the model fit.

4.9 Test Model Fit: LINKTEST

A simple test of model fit is a regression of the dependent variable on the predictive value (STATA also includes the squared predicted value). If the predicted value actually predicts the true value well, the slope on the predicted value will be *near one*.

Use **linktest**

Stata Results LINKTEST:

```
. linktest
```

```
      Source |       SS       df       MS              Number of obs =      51
-------------+------------------------------           F(  2,    48) =   14.54
       Model |  359011.659     2  179505.829            Prob > F      =  0.0000
    Residual |  592718.575    48  12348.3037            R-squared     =  0.3772
-------------+------------------------------           Adj R-squared =  0.3513
       Total |  951730.234    50  19034.6047            Root MSE      =  111.12

------------------------------------------------------------------------------
        mort |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       _hat |    5.964705   3.147421     1.90   0.064    -.3636097    12.29302
     _hatsq |   -.0029816   .0018866    -1.58   0.121    -.0067749    .0008117
       _cons |  -2046.014   1305.298    -1.57   0.124    -4670.491     578.462
------------------------------------------------------------------------------
```

The slope 5.96 is very far from 1.00! suggesting the mortality model is now well specified. This corroborates the RESET test outcome, above.

EXAMPLE #4 (OLS on simulated data)

In order to gauge how well OLS works, and how informative STATA's resgression package is, consider the following simulated model:

$$Y_i = \beta_0 + \beta_1 X_{i,1} + \beta_2 X_{i,2} + \varepsilon_i = 10 + 2X_{i,1} - 5X_{i,2} + \varepsilon_i$$

where all regressors $X_{i,j}$ and the errors ε_i are *iid* standard normal distributed. We will regress Y on X1 and X2.

We will test whether the residuals are normally distributed (*the true errors are*).

We will perform the following tests: (1) $\beta_1 = \beta_2 = 0$; (2) $\beta_2 = -5$; (3) $\beta_0/\beta_1 = 5$. Hypotheses (2) and (3) are true.

We will perform the RESET test. The model is correctly specified (I simulated Y to be a linear function of X1 and X2, hence the above model is correct by construction). Thus, the RESET should not reveal mis-specification (i.e. if the test is performed at the 5% level, there is at most a 5% probability we will reject the null of correct specification).

Finally, we will perform a "link test". The slope on the predicted value should be close to one.

Stata Results REGRESS, OVTEST, LINKTEST:

```
. regress y x1 x2

      Source |       SS       df       MS              Number of obs =     100
-------------+------------------------------           F(  2,    97) = 1271.89
       Model |  2973.98538     2  1486.99269            Prob > F      =  0.0000
    Residual |  113.404579    97  1.16911937           R-squared     =  0.9633
-------------+------------------------------           Adj R-squared =  0.9625
       Total |  3087.38996    99  31.1857572           Root MSE      =  1.0813

------------------------------------------------------------------------------
           y |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          x1 |    1.838796   .1171392    15.70   0.000     1.606307    2.071285
          x2 |   -4.878635   .1027937   -47.46   0.000    -5.082652   -4.674618
       _cons |    9.941118   .1084732    91.65   0.000     9.725829    10.15641
------------------------------------------------------------------------------
```

Everything is very sharp, and the R^2 > .96.

```
. predict ehat, resid        (generate residuals)
. sktest ehat                (tests if residuals are normally distributed)
```

Skewness/Kurtosis tests for Normality

Variable	Pr(Skewness)	Pr(Kurtosis)	joint adj chi2(2)	joint Prob>chi2	
ehat	0.850	0.284	1.21	0.5458	**(the p-val > .20 suggests normality)**

Ramsey RESET test using powers of the independent variables
Ho: model has no omitted variables
F(6, 91) = 1.35
Prob > F = 0.2416 (no evidence of mis-specification)

. linktest **(model specification test of dependent variable on predicted value)**

y	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
_hat	**.9197584**	.0610805	15.06	0.000	.7985306	1.040986
_hatsq	.0039691	.0028797	1.38	0.171	-.0017463	.0096845
_cons	.2911855	.3023668	0.96	0.338	-.3089289	.8912999

In summary, the OLS estimates are sharp; the true regression errors are normally distributed and the test suggests they are; the RESET test does not indicate model mis-specification by omitted powers of the regressors *x*; and the prediced values *y_hat* are very similar to the dependent variable *y*.

In total, our methods worked very well.

5. PROGRAMMING in STATA – THE DO FILE ".do"

Needless to say, using the Command window is laborious, and all such commands are not permanently stored in a file that can run at any time on any data set.

In this section we use a do-file for code storage.

5.1 Opening an Existing Do-File

From the main toolbar, click on Window, Do-File Editor. In the pop-up box, File, Open, then find the *filename.do* file.

5.2 Running a Do-File

From the **Do-File Editor** pop-up window, click on **Tools, Do**. Or just click the symbol of the *page with text with a down-arrow on the right*.

5.3 Creating and Editing a Do-File

Consider the U.S. state-wide morality data from Section 4. In the following it is assumed that this stata data **mort_tob.dta** file with variables *mort, inc_pc, tob,alc, ed_hs, ed2_coll* is stored in file **c:\temp\stata_files**.

In the main tool-bar click-on **Window**, **File Do-Editor**. A new **Do File Editor** window appears for code editing.

The following code can be directly typed or copy/pasted into the **Do File Editor**.

Note: anything between comment markers **/* .. */** is ignored by STATA.

EXAMPLE #5 (Do-File code for estimating mortality model)

The following is a representative Do-File that loads the mortality dataset from a pre-specified folder. A regression model is estimated and analyzed. Copy-paste it into the **DO-EDITOR**.

```
cap log close                          /* closes existing log files */
set more 1                             /* tells STATA not to wait for key-board input   */
clear
cd c:\temp\stata_files                 /* loaded and saved files to and from here       */
log using mort_log, replace            /* creates a log of current session*/

use mort                               /* without a file tag, it must be a .dta file                  */
                                       /* STATA will look in c:\temp\stata_files for mort             */

summarize                                  /* summary statistics                                      */
regress mort inc_pc ed_hs ed2_coll alc tob /* OLS                                                     */

predict yhat                               /* created predicted values                                */
predict ste_yhat,stdp                      /* standard errors for y_hat                               */
predict ehat, resid                        /* residuals                                               */

ksmirnov ehat = normprob(ehat)         /* tests if the regression errors are normal – strong test     */
sktest ehat                            /* tests if the regression errors are normal – weak test       */

test ed_hs ed2_coll                    /* tests for joint non-influence of ed.      */
test ed_hs + ed2_coll = 0              /* tests for ed. cancellation                */
test ed_hs = ed2_coll                  /* tests for equality of ed. impacts         */
test inc_pc = .025
testnl _b[ed_hs]/_b[ed2_coll] = 1

ovtest                                 /* Ramsey's RESET model mis-spec. test                         */
ovtest, rhs                            /* Modified RESET: only regressors cross-products used         */
linktest                               /* Link test of model mis-specification                        */

log close
```

In order to **run** the code, in the toolbar of the **DO-EDITOR**: **TOOLS**, **DO**.

Save the Do-File and use it in the future with whatever modifications you require.

6. GENERALIZED LINEAR MODEL

The regression model is still

$$y_i = \beta_1 + \beta_2 x_{i,2} + ... + \beta_k x_{i,k} + \varepsilon_i$$

but now we are interested in whether the errors have a non-constant variance, or are correlated.

The former case of *heteroscedasticity* (non-constant variance) here will be treated for cross-sectional data (e.g. U.S. mortality rates). The model is

$$y_i = \beta_1 + \beta_2 x_{i,2} + ... + \beta_k x_{i,k} + \varepsilon_i \text{ where } \varepsilon_i \text{ is independent, } E[\varepsilon_i] = 0 \text{ but } V[\varepsilon_i] = \sigma_i^2$$

In theory a non-constant variance σ_i^2 is not a issue: OLS still works. But if the variance is related to an included regressor we need to be careful about computing standard errors and therefore *t*-statistics.

In the latter case of *autocorrelated* errors (the errors are correlated with themselves), the appropriate setting is time series (e.g. U.S. GDP). The model is

$$y_t = \beta_1 + \beta_2 x_{i,2} + ... + \beta_k x_{i,k} + \varepsilon_t \text{ where } \varepsilon_t \sim (0, \sigma^2) \text{ but maybe } corr(\varepsilon_t, \varepsilon_{t-h}) \neq 0 .$$

If the errors are correlated and a regressor(s) $x_{t,i}$ is a lag of y_t (e.g. $x_{t,2} = y_{t-1}$), then OLS is biased and inconsistent.

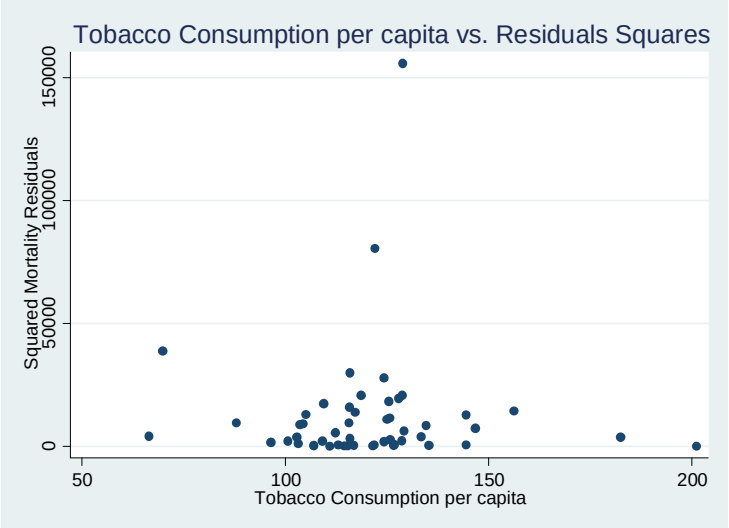
6.1 Graphical Evidence of Heteroscedasticity

Although hardly rigorous, a simple first step for considering if the errors have variances that depend on the observable regressors involves basic scatter plots.

Consider the *mortality* dataset. We might think those states with little tobacco consumption have a lower morality rate. Conversely, we might think higher consumption may or may not lead to higher death rates: the cross-state variable might be quite large.

Stata Results:

. regress mort inc_pc pov ed_hs ed2_coll alc_pc tob_pc	**(regression of mortality rates)**
. predict e, resid	**(create residuals)**
. gen e2 = e^2	**(squared residuals)**
. graph twoway scatter e2 tob_pc	**(scatter plot)**



States with higher tobacco consumption represent a greater dispersion of death rates. Hence, we suspect mortality rates are heterosceastic.

6.2 Heteroscedasticy: Robust Estimation

Suppose the regression model is

$$y_i = \beta_1 + \beta_2 x_{i,2} + ... + \beta_k x_{i,k} + \varepsilon_i, \varepsilon_i \text{ is independent}, E[\varepsilon_i] = 0 \text{ but } V[\varepsilon_i] = \sigma_i^2$$

If regression errors heteroscedastic ($\sigma_i^2 \neq \sigma_j^2$) standard errors generated by STATA will be wrong.

Unless a specific model of heteroscedasticity is entertained, the easiest method is to use robust standard errors that do not require any knowledge of why or how the errors are heteroscedastic.

OLS with White/Huber robust standard errors

regress y x1 x2…, robust

Stata Results <u>ROBUST</u>:

```
. regress  mort inc_pc  pov  ed_hs  ed2_coll  alc_pc  tob_pc, robust
```

mort	Coef.	**Robust Std. Err.**	t	P>|t|	[95% Conf.	Interval]
inc_pc	.0258083	.0253884	1.02	0.315	-.0253586	.0769752
pov	166.6582	1575.319	0.11	0.916	-3008.189	3341.506
ed_hs	-807.5147	925.4417	-0.87	0.388	-2672.62	1057.59
ed2_coll	-1523.302	1396.199	-1.09	0.281	-4337.155	1290.552
alc_pc	4.685375	40.11099	0.12	0.908	-76.15301	85.52376
tob_pc	.0686132	1.143506	0.06	0.952	-2.235971	2.373198
_cons	1264.35	973.2808	1.30	0.201	-697.1687	3225.868

6.3 Heteroscedasticity: Tests

After estimating a model, we can test the residuals for heteroscedasticity by the Breusch-Pagan test.

6.31 General Tests: Multiplicative and Breusch-Pagan

There are two popularly encountered and therefore modeled forms of general heteroscedasticity. In the first, the individual specific variance is a simple *multiplicative* function of some available regressor. For example, mortality rate variance may be lower in higher income states (more income allows residents to uniformly similar health care). This can be represented as

1. $V[\varepsilon_i] = \sigma^2 \times inc\_pc_i$
2. $V[\varepsilon_i] = \sigma^2 \times inc\_pc_i^2$.
3. $V[\varepsilon_i] = \sigma^2 \times \exp\{\gamma \times inc\_pc_i\}$

All three are reasonable since income is positive. Use the latter two if the regressor can be negative. The second one is used by STATA since a test of homoscedasticity is a test of $\gamma = 0$.

Use **hettest** after **regress**:

regress mort ed_hs ed2_coll tob_pc alc_pc inc_pc
hettest inc_pc (performs test #3 above)

The command **hettest** assumes the regression errors are normally distributed. If you suspect, or have evidence, that the errors are non-normal use:

hettest inc_pc, fstat

In the second method, variance is an *general* linear function of available regressors:

1. $V[\varepsilon_i] = \gamma_1 + \gamma_2 x_{i,2} + ... + \gamma_k x_{i,k}$
2. $V[\varepsilon_i] = \left(\gamma_1 + \gamma_2 x_{i,2} + ... + \gamma_k x_{i,k}\right)^2$
3. $V[\varepsilon_i] = \exp\left\{\gamma_1 + \gamma_2 x_{i,2} + ... + \gamma_k x_{i,k}\right\}$

The first model represents variance; the second is standard deviation; and the third is log-variance. The *latter two guarantee that estimates produce positive forecast of the variance*. In all three a test of homoscedasticity is a test of all slopes $\gamma_i = 0$.

Use **hettest** after **regress**:

regress mort ed_hs ed2_coll tob_pc alc_pc
hettest ed_hs ed2_coll tob_pc alc_pc (tests #3 above)

hettest, rhs (uses all "right hand side" variables in the regression model)

hettest, rhs fstat (uses all "right hand side" variables, and does not assume normality)

Stata Results <u>HETTEST</u>:

```
. regress  mort ed_hs ed2_coll alc_pc tob_pc
```

mort	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
ed_hs	-846.6245	316.1168	-2.68	0.010	-1482.935	-210.3143
ed2_coll	-461.4192	790.4127	-0.58	0.562	-2052.437	1129.599
alc_pc	18.44112	32.97247	0.56	0.579	-47.92902	84.81126
tob_pc	.4818908	.9954574	0.48	0.631	-1.521861	2.485643
_cons	1393.322	239.8506	5.81	0.000	910.5278	1876.116

. hettest , rhs **(tests for heteroscedasticcity by using all regressors)**

Breusch-Pagan / Cook-Weisberg test for heteroskedasticity
Ho: Constant variance
Variables: ed_hs ed2_coll alc_pc tob_pc

chi2(4) = 10.89, Prob > chi2 = 0.0278 **(suggests we should use robust s.e.'s)**

. hettest , rhs fstat **(uses all regressors, and does not assume normality)**

Breusch-Pagan / Cook-Weisberg test for heteroskedasticity
Ho: Constant variance
Variables: ed_hs ed2_coll alc_pc tob_pc
F(4 , 46) = 2.95, Prob > F = 0.0299 **(same conclusion)**

6.3.2 Group-Wise Heteroscedasticity Tests

Sometimes we are interested in whether heteroscedasticity arises due to *group characteristics*. Classic examples include consumption or expenditure: high income groups have more opportunities for various levels of expenditure and therefore exhibit more dispersion. Similarly high versus low education; regions (conservative south versus non-conservative coastal areas).

Use **robvar *varx1*, by(*varx2*)** to test heteroscedasticity in *varx1* across two groups depicted in *varx2*.

The two groups can be numerically represented by a dummy variable.

egen m_inc = median(inc_pc)	(create median income)
generate hi_lo_inc = (inc_pc < m_inc)	(= 1 if below median income)
quietly regress mort ed_hs ed2_coll tob_pc alc_pc	(regress without display)
predict e, resid	(generate residuals)
robvar e, by(hi_lo_inc)	(heteroscedasticity test across below-median and above-median income groups)

Stata Results <u>ROBVAR, BY</u>:

```
. egen m_inc = median(inc_pc)
. generate hi_lo_inc = (inc_pc < m_inc)
. quietly regress mort ed_hs ed2_coll tob_pc alc_pc
. predict e, resid
. robvar e, by( hi_lo_inc)
```

Summary of Residuals			
hi_lo_inc	Mean	Std. Dev.	Freq.
0	22.90639	132.42284	26
1	-23.822646	94.139977	25
Total	-1.028e-07	116.52652	51

W0 = 1.1134026 df(1, 49) Pr > F = 0.29651858
W50 = 1.0141617 df(1, 49) Pr > F = 0.31885522
W10 = 1.0154126 df(1, 49) Pr > F = 0.31855998

Robvar generates three versions of same F-test. Here, the p-values are larger that any conventional level, so we fail to reject H0: homoscedasticity.

6.4 Heteroscedasticity: FGLS Estimation

The robust least squares estimator discuss in Section 6.1 is always correct under mild assumptions. If, however, we have a model of the error variance we can use it to generate a *Feasible Generalized Least Squares* [FGLS] estimator.

If the model is

$$y_i = \beta_1 + \beta_2 x_{i,2} + ... + \beta_k x_{i,k} + \varepsilon_i, \varepsilon_i \text{ is independent, } E[\varepsilon_i] = 0 \text{ but } V[\varepsilon_i] = \sigma_i^2$$

and we have a consistent estimator $\hat{\sigma}_i^2$, then the FGLS estimator minimizes

$$\frac{y_i}{\hat{\sigma}_i} = \beta_1 \frac{1}{\hat{\sigma}_i} + \beta_2 \frac{x_{i,2}}{\hat{\sigma}_i} + ... + \beta_k \frac{x_{i,k}}{\hat{\sigma}_i} + \frac{\varepsilon_i}{\hat{\sigma}_i}$$

Since in a large sample $\varepsilon_i / \hat{\sigma}_i$ has a variance of 1, the above denotes a homoscedastic regression model, so all prior theory and methods apply.

Since this is nothing more than Weighted Least Squares we can use **regress** with a few tweaks.

6.4.1 Multiplicative Heteroscedasticity

If variance is a multiplicative scale of some positive variable *xj*

$$\sigma_i^2 = \sigma_i \times x_{i,j}$$

use

regress y x1…xk [aw = xj]

6.4.2 Linear Heteroscedasticity

If variance, or standard deviation, or log-variance is a linear function of available explanatory variances

1. $V[\varepsilon_i] = \gamma_1 + \gamma_2 x_{i,2} + ... + \gamma_k x_{i,k}$
2. $V[\varepsilon_i] = \left(\gamma_1 + \gamma_2 x_{i,2} + ... + \gamma_k x_{i,k}\right)^2$
3. $V[\varepsilon_i] = \exp\left\{\gamma_1 + \gamma_2 x_{i,2} + ... + \gamma_k x_{i,k}\right\}$

then we can generate auxiliary regressions to obtain $\hat{\sigma}_i$ (*sig_hat*) and use this in

regress y x1…xk [aw = sig_hat]

Stata Results REGRESS [AW = …]:

```
. regress mort ed_hs ed2_coll tob_pc alc_pc [aw= inc_pc]          (use income as scale)

Source |       SS       df       MS              Number of obs =      51
-------------+------------------------------   F(  4,    46) =    4.73
Model |  300585.113     4  75146.2783           Prob > F      =  0.0028
Residual |  730656.759    46  15883.8426           R-squared     =  0.2915
-------------+------------------------------   Adj R-squared =  0.2299
Total |  1031241.87    50  20624.8374           Root MSE      =  126.03

------------------------------------------------------------------------------
mort |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
ed_hs |  -1076.487   322.4476    -3.34   0.002    -1725.54   -427.4333
ed2_coll |  -169.0088    785.5912    -0.22   0.831    -1750.322    1412.304
tob_pc |   .2098488   1.046533     0.20   0.842    -1.896712    2.316409
alc_pc |   22.42783   33.51721     0.67   0.507    -45.03881    89.89446
_cons |   1525.946   253.0172     6.03   0.000     1016.649    2035.243
------------------------------------------------------------------------------
```

Stata Results REGRESS [AW = …]:

```
. quietly regress mort ed_hs ed2_coll tob_pc alc_pc                (estimate original model)
. predict e, resid                                                 (generate residuals e)
. generate abs_e = abs(e)                                          (generate absolute value |e|)
. quietly regress abs_e inc_pc pov ed_hs ed2_coll alc_pc tob_pc    (regression |e| on x's)
. predict sig_hat                                                  (predicted value is sig_hat)
. replace sig_hat = abs(sig_hat)
. regress mort ed_hs ed2_coll tob_pc alc_pc [aw= sig_hat]          (weight least squares)

Source |       SS       df       MS              Number of obs =      51
-------------+------------------------------   F(  4,    46) =    5.98
Model |  424785.524     4  106196.381           Prob > F      =  0.0006
Residual |  816578.28    46  17751.7017           R-squared     =  0.3422
-------------+------------------------------   Adj R-squared =  0.2850
Total |  1241363.80    50  24827.2761           Root MSE      =  133.24

------------------------------------------------------------------------------
mort |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
ed_hs |  -1344.335   320.0413    -4.20   0.000    -1988.544   -700.125
ed2_coll |   254.8928    837.3543     0.30   0.762    -1430.614     1940.4
 tob_pc |   .3746327   1.166548     0.32   0.750    -1.973506    2.722772
 alc_pc |   30.29459   36.01464     0.84   0.405    -42.19913    102.7883
_cons |   1591.664   263.4308     6.04   0.000     1061.405    2121.922
------------------------------------------------------------------------------
```

6.5 Correlated Errors

Consider a linear regression model

$$y_t = \beta_1 + \sum_{i=2}^{k} \beta_i x_{i,t} + \varepsilon_i \quad \sigma_\varepsilon^2 = V(\varepsilon_i)$$

with OLS residuals $\hat{\varepsilon}_t$.

The Autocorrelation Function (ACF) of the errors ε_t is simply the correlations between ε_t and ε_{t-h} for arbitrary lag $h > 0$:

$$\text{ACF} : \rho_h = \frac{\text{cov}(\varepsilon_t, \varepsilon_{t-h})}{\sigma_\varepsilon^2}$$

If regression errors ε_t are autocorrelated[2] ($\rho_h \neq 0$ for some lag h) then OLS estimates will be biased if the regressors x_t contain lags of the dependent variable y_t.

The Sample Autocorrelation Function (SACF) is simply

$$\text{SACF} : \hat{\rho}_h = \frac{\hat{\text{cov}}(\varepsilon_t, \varepsilon_{t-h})}{\hat{\sigma}_\varepsilon^2} = \frac{\frac{1}{n}\sum \hat{\varepsilon}_t \hat{\varepsilon}_{t-h}}{\frac{1}{n}\sum \hat{\varepsilon}_t^2}.$$

We can inspect the SACF of the errors, *test for autocorrelation*, and estimate in a way that *corrects for serial correlation*.

If the errors are *iid* then all population autocorrelations $\rho_h = 0$,so for large samples the SAFC 95% confidence bands are

$$0 \pm 1.96 / \sqrt{n}$$

Thus, if the SACF $\hat{\rho}_h$ drifts outside the band $\pm 1.96 / \sqrt{n}$ we have evidence at the 5% level for serial correlation in the errors.

[2] "Autocorrelation" and "serial correlation" are synonymous terms for intertemporal correlation in one time series variable.

A convenient (and efficient) way to test for autocorrelation is to test multiple sample aucorrelations $\hat{\rho}_h$ at the same time. The following so-called "Q-statistic" was suggested by Box and Pierce:

$$Q_h = n\sum_{i=1}^{h}\hat{\rho}_i^2$$

If the errors are *iid* then all population autocorrelations $\rho_h = 0$, so for large samples Q_h is roughly chi-squared distributed with h degrees of freedom:

$$\text{If } \varepsilon_t \text{ is iid then } Q_h \approx \chi^2(h)$$

Thus, a large Q_h provides evidence for serial correlation.

The traditional way to inspect the Q-statistic Q_h is to compute it for multiple lags $I = 1,\ldots,h$ for some logical horizon (e.g. 12 months for monthly data; 4 quarters for quarterly data)

6.5.1 Declaring and Plotting Time Series

In STATA we must first declare the data to be from a time series. We must first create a variable that represents time, and then declare the data to be a time series.

gen t = _n	generates a time variable called "t": t = 1,2,…,*n*.
tsset t	declares the data are a time series, and *var_time* represents time.
tsline *var*	a line plot of *var*: this can only be done AFTER the data set is declared a time series (i.e. only after **gen t** and **tsset t** are done).

NOTE: You have to this each time you open a time series file, but only one time once the file is open.

Sometimes it is beneficial, or imperative, that we work with logged data, or growth which is a difference in logs.

gen ln_gdp - log(GDP)	generates *ln*(*GDP*(*t*))
gen growth_y = ln_gdp - ln_gdp[_n-1]	*ln*(*GDP*(*t*))- *ln*(*GDP*(*t*-1)) which is growth

The slope of *ln*(*GDP*(*t*)) over time is simply growth

Stata Results TSLINE:

```
. gen t = _n                                  (create trend)
. tsset t                                     (declare time series)
      time variable:  t, 1 to 528
            delta:  1 unit
. gen ln_gdp = log(GDP)                       (create natural log of GDP)
. gen growth_y =  ln_gdp- ln_gdp[_n]          (create growth)
```

6.5.1 Analyzing Autocorrelated Errors

We can compute and plot the autocorrelations of the residuals.

corrgram var1, lags(##) produces autocorrelations and Q-statistics for *var1* over multiple lags *i* = 1,…,##.

ac var1, lags(##) produces autocorrelations for *var1* with confidence bands under the null of independence; the output is a *graphic plot* over multiple lags *i* = 1,…,##.

The default band is 95%. For any other, include **level(##)**. For example

corrgram var1, lags(12) level(99)

Stata Results CORRGRAM:

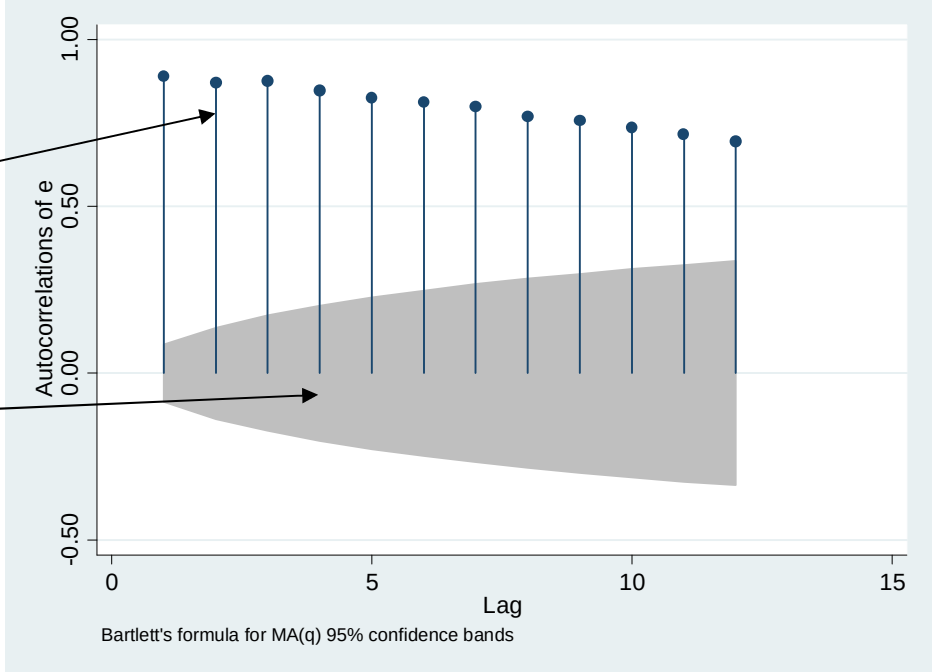
```
. quietly regress  GDP M1 M2  t_bill        (OLS without ouput)
. predict e, resid                          (generates residuals)
. gen t = _n                                (generates a time variable: t = 1,2,…,n)
. tsset t                                   (declares data is time series)
. corrgram e, lags(12)
```

LAG	AC	PAC	Q	Prob>Q	[Autocorrelation]	[Partial Autocor]
1	0.8900	0.8921	420.63	0.0000	|-------	|-------
2	0.8711	0.3909	824.32	0.0000	|------	|---
3	0.8760	0.3394	1233.3	0.0000	|-------	|--
4	0.8485	0.0769	1617.8	0.0000	|------	|
5	0.8257	0.0073	1982.6	0.0000	|------	|
6	0.8128	0.0132	2336.8	0.0000	|------	|
7	0.7998	0.0401	2680.4	0.0000	|------	|
8	0.7691	-0.0733	2998.8	0.0000	|------	|
9	0.7576	0.0133	3308.2	0.0000	|------	|
10	0.7373	-0.0373	3601.9	0.0000	|-----	|
11	0.7173	0.0039	3880.4	0.0000	|-----	|
12	0.6944	-0.0441	4141.9	0.0000	|-----	|

The Q-statistics are all highly significant: their *p*-values are all < .001 (Prob > Q).

Stata Results AC:

```
. ac e, lags(12)
```



This is the AC plot.

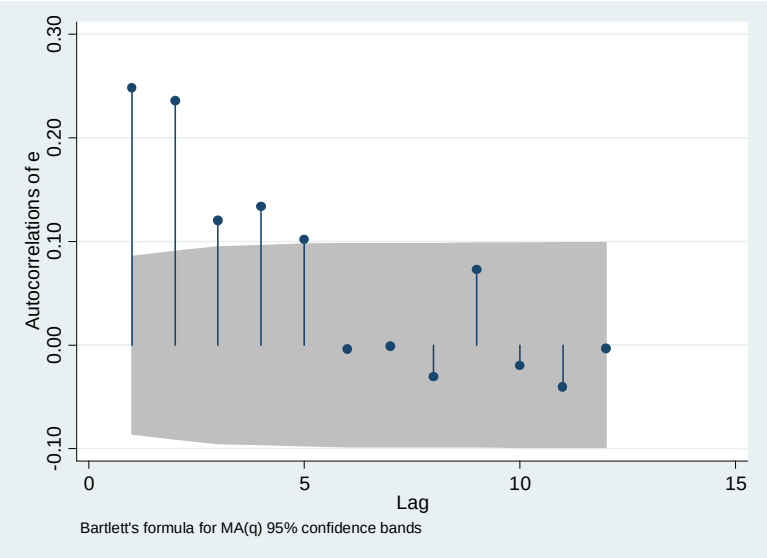
This is the confidence band under the null of no correlation.

The problem may be that we are working with the levels of GDP, M1, M2 and IPI. These are notoriously difficult to model, and they are clearly growing over time: this suggests our model is missing *t* as a regressor.

One solution is work with growth: $y = \%\Delta GDP$, $m1 = \%\Delta M1$, $m2 = \%\Delta M2$.

Stata Results AC:

```
. gen y = log(GDP)-log(GDP[_n-1])
. gen m1 = log(M1)-log(M1[_n-1])
. gen m2 = log(M2)-log(M2[_n-1])
. quietly regress  y m1 m2 t_bill
. predict e, resid
. ac e, lags(12)
```

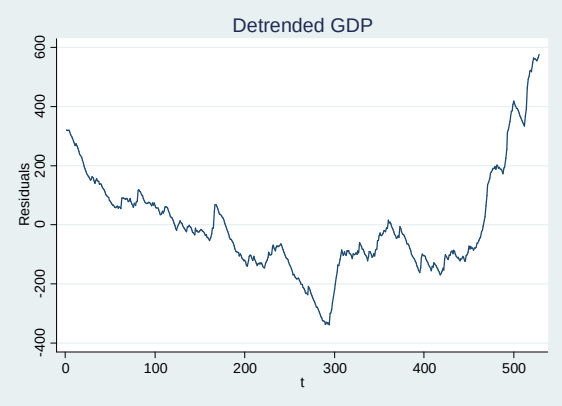


In this case the errors exhibit far less serial correlation. This simply reveals a delicate issue: are income or income growth shocks actually correlated over time (e.g. a recession lasts multiple periods)? or is our model simply mis-specified?

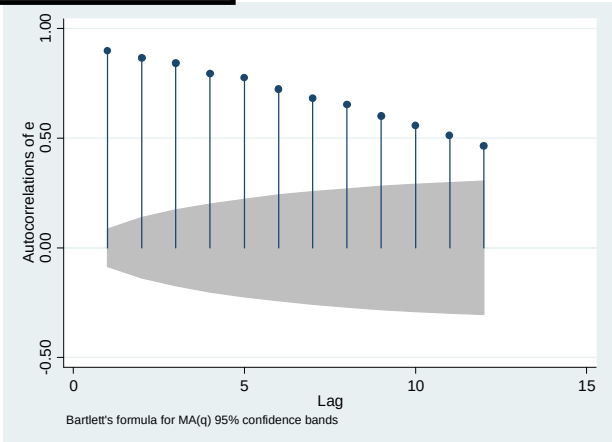
Another solution is to de-trend the series. Each growth with *t*, so we can remove that growth by subtracting out its linear increase over time.

Stata Results AC:

. quietly regress GDP t	**(estimates GDP(*t*) = a + b*t* + *u*(*t*) where a + b*t* represents trend)**
. predict gdp_dt, resid	**(defines *gdp_dt* as the residual GDP(*t*) - (a + b*t*) which is de-trended income)**
. quietly regress M1 t	
. predict m1_dt, resid	
. quietly regress M2 t	
. predict m2_dt, resid	
. tsline gdp_dt	**(notice the plot: shows clear business cycles)**




```
. quietly regress  gdp_dt m1_dt m2_dt t_bill
. predict e, resid
. ac e, lags(12)
```



Working with detrend income and money helps a little. Ultimately we will see in class that lagged income and money, in levels and growth, are strongly associated with current income or income

growth. Thus, what we see above may be due to omitted lagged income and money showing up the residuals, causing the residuals to appear correlated.

6.5.2 Testing for Autocorrelation

There are two ways to perform the Q-test. We have seen one already via correlgram.

corrgram var1, lags(##) produces autocorrelations and Q-statistics for *var1* (and tests for autocorrelation)

wntestq var1, lags(##) performs the Q-Test of no serial correlation for only the specified lag, or the default lag $h = 40$.

An alternative test is the Durbin-Watson test. This tests the residuals for first order autocorrelation (i.e. it tests whether $e_t = a + b\, e_{t-1} + v_t$ for iid v_t: the null hypothesis is no error serial correlation hence $b = 0$).

Perform this task after any regression: it automatically tests the most recent residuals. For example:

```
regress GDP M1 M2  IPI t_bill
dwstat
```

The *Durbin-Watson* test is far *inferior* to the Q-test because: it only tests of first order autocorrelation, and it has a non-standard distribution. In fact, we only have a critical range (a,b) for 1%, 5% and 10% levels. If DW < a there is evidence for serial correlation; if DW > b we have evidence for uncorrelatedness. Otherwise…do a Q-test!

Stata Results <u>WNTESTQ</u>:

```
. wntestq e, lags(1)
```
(this performs the Q-test at $h = 1$ for the same GDP residuals)

```
Portmanteau test for white noise
---------------------------------------
Portmanteau (Q) statistic       =   420.6282
Prob > chi2(1)                  =     0.0000
```

Compare the output to **correlgram**: the Q-statistic is the same for lag $h = 1$.

Stata Results <u>DWSTAT</u>:

```
. dwstat
```

Durbin-Watson d-statistic(5, 528) = .2127027 **(the 1% critical range is 1.44…1.68)**

Since DW < 1.44, we strongly reject the no serial correlation null hypothesis at the 1% level.

6.5.3 Estimation with Autocorrelated Errors

There are two ways to estimate models with correlated errors: **newey** and **prais**

1. The command **newey** performs OLS with the Newey-West robust standard errors in a time series framework.

   newey y x1 x2 …xk, lag(##)

   Note: **lag(##)** *must be declared*: it states how many lags the routine uses to approximate the correlation structure of the errors.

   Note: it is *singular* **lag(##)** and *not plural* **lags(##)**.

2. Use **prais** to use a built-in correction for first order serial correlation in the errors.

   praise y x1 x2…xk

   Include **corc** specifically for the Corchrane-Orchutt method.

   Include **robust** for Huber/White robust standard errors.

   The estimation method use *iterations* where each step involves an update of the sample first order correlation. There is no advantage since each step produces an efficient estimator.

   Include **twostep** to force STATA to skip all but the minimial number of iterations.

STATA Results: NEWEY

Recall *y*, *m*1, *m*2 are growth variables for GDP, M1 and M2.

```
. newey  y m1 m2 t_bill, lag(12)
```

Regression with Newey-West standard errors
maximum lag: 12

Number of obs = 527
F(3, 523) = 4.13
Prob > F = 0.0065

y	Coef.	**Newey-West Std. Err.**	t	P>|t|	[95% Conf.	Interval]
m1	-.0073169	**.0054061**	-1.35	0.176	-.0179373	.0033035
m2	.0190242	**.0083785**	2.27	0.024	.0025645	.0354839
t_bill	-.0001183	**.0000765**	-1.55	0.122	-.0002685	.0000319
_cons	.0035215	**.0004861**	7.25	0.000	.0025666	.0044764

The *only* difference between NEWEY and REGRESS is the method used to compute the standard errors and therefore t-statistics. Compare (the coefficient estimates are identical):

```
. regress  y m1 m2 t_bill
```

```
      Source |       SS           df       MS              Number of obs =     527
-------------+----------------------------------           F(  3,   523) =    3.32
       Model |  .000091651     3   .00003055               Prob > F      =  0.0197
    Residual |  .004817667   523  9.2116e-06               R-squared     =  0.0187
-------------+----------------------------------           dj R-squared =  0.0130
       Total |  .004909318   526  9.3333e-06               Root MSE      =  .00304

------------------------------------------------------------------------------
           y |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          m1 |   -.0073169   .0078602    -0.93   0.352    -.0227584    .0081246
          m2 |    .0190242   .0110547     1.72   0.086    -.0026929    .0407413
      t_bill |   -.0001183   .0000503    -2.35   0.019    -.0002172   -.0000195
       _cons |    .0035215   .0003343    10.53   0.000     .0028648    .0041783
------------------------------------------------------------------------------
```

STATA Results: PRAIS

```
. prais y m1 m2  t_bill, corc twostep
```

```
Iteration 0:  rho = 0.0000
Iteration 1:  rho = 0.2484

Cochrane-Orcutt AR(1) regression -- twostep estimates

      Source |       SS           df       MS              Number of obs =     526
-------------+----------------------------------           F(  3,   522) =    2.31
       Model |  .000059849     3     .00001995              Prob > F      =  0.0759
    Residual |  .004515993     522  8.6513e-06              R-squared     =  0.0131
-------------+----------------------------------           Adj R-squared =  0.0074
       Total |  .004575843   525  8.7159e-06               Root MSE      =  .00294

------------------------------------------------------------------------------
           y |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          m1 |   -.0061595   .0068059    -0.91   0.366    -.0195299    .0072109
          m2 |    .0153733   .0096702     1.59   0.112    -.0036239    .0343706
      t_bill |   -.0001149   .0000631    -1.82   0.070    -.0002389    9.20e-06
       _cons |    .0035034   .0004206     8.33   0.000     .0026772    .0043296
-------------+----------------------------------------------------------------
         rho |   .2484474
------------------------------------------------------------------------------
Durbin-Watson statistic (original)    1.502521      (DW test for original model)
Durbin-Watson statistic (transformed) 2.090229      (DW test for transformed model)
```

7. LIMITED DEPENDENT VARIABLES

The dependent variable now takes on only finitely many possible values. The classic examples are:

y_i = 0 or 1 depending on some behavior or choice:

(e.g. smoke or not, sell an asset or not, accept a job offer or not)

y_i = 1, 2, ..., or R where R is finite, depending on a choice or outcome:
(e.g. a non-U.S. firm builds a plant in a U.S. state: R = 50)

We now want to model the likelihood of a particular limited dependent outcome $y_i = r$, given some known traits x_i:

$$P\left(y_i = r \mid x_{1,i},...,x_{k,i}\right)$$

The standard way to write a binary response model, for example, is

(1)

$$y_i = 1 \text{ if } \varepsilon_i \geq -\left(\beta_1 + \beta_2 x_{2,i} + ... + \beta_k x_{k,i}\right)$$

$$y_i = 0 \text{ if } \varepsilon_i > -\left(\beta_1 + \beta_2 x_{2,i} + ... + \beta_k x_{k,i}\right)$$

It is tempting to say as we have before $y_i = \beta_1 + \beta_2 x_{2,i} + \cdots + \beta_k x_{k,i} + \varepsilon_i$ but this has awkward implications for the errors. Instead, (1) says if some "*latent*" variable, call it y^* is big then the response is y_i = 1. We say "latent" because we do not observe it: if we did, we would use it and not the binary responses!

Consider smoking: if a person gains enough utility from smoking then they smoke. Put into (1):

if utility $y_i^* = \beta_1 + \beta_2 x_{2,i} + \cdots + \beta_k x_{k,i} + \varepsilon_i > 0$ then smoke: y_i = 1.

All we observe are the x's (e.g. age, gender, education) and the response (e.g. smoke or not).

In general, we would like to know how this likelihood responds to changes in the traits:

(2)

$$\frac{\partial}{\partial x_{j,i}} P\left(y_i = 1 \mid x_{1,i},...,x_{k,i}\right) : \text{ the marginal impact } x_{j,i} \text{ has on the choice } y_i = 1.$$

Due to the model'ls complexity, this is not the same thing as β! Thus

$$\beta_j \neq \frac{\partial}{\partial x_{j,i}} P\left(y_i = r \mid x_{1,i},...,x_{k,i}\right)$$

However, if $\beta_j > 0$ or < 0 then the impact $x_{j,i}$ has on the choice $y_i = 1$ probability has the same sign:

$$\beta_j < 0 \text{ implies } \frac{\partial}{\partial x_{j,i}} P\left(y_i = r \mid x_{1,i},...,x_{k,i}\right) < 0, \text{ and so on.}$$

Consider that going to college increases the probability that a person does not smoke:

$$y_i = 0 \text{ if not smoke}, = 1 \text{ if smoke}$$

$$\frac{\partial}{\partial ed} P\left(y_i = 0 \mid x_{1,i},...,x_{k,i}\right) > 0$$

This is a testable hypothesis: we simply test the associated slope $\beta_j \leq 0$ against $\beta_j > 0$.

7.1 Likelihood Function and Maximum Likelihood

In order to estimate the marginal effects above for binary dependent variables (i.e. $y_i = 0$ or 1) we need to select a probability distribution function for y. *Probit* estimation refers to a normal distribution for ε_i. See Section 7.2.

The likelihood function is the joint probability of the observed choices. Suppose our sample has $y_i = \{0, 1, 1, 0 ,...\}$. We assume individual people are independent of each other. Then the joint probability of y_i is

$$P\left(y_1 = 0, y_2 = 1, y_3 = 1, y_4 = 0,...\right) = P\left(y_1 = 0\right) \times P\left(y_2 = 1\right) \times P\left(y_3 = 1\right) \times P\left(y_4 = 0\right) \times ...$$

Now use the latent regression model to get

$$P\left(y_i = 1\right) = P\left(\varepsilon_i \geq -\left(\beta_1 + \beta_2 x_{2,i} + ... + \beta_k x_{k,i}\right)\right)$$

$$P\left(y_i = 0\right) = P\left(\varepsilon_i < -\left(\beta_1 + \beta_2 x_{2,i} + ... + \beta_k x_{k,i}\right)\right)$$

We now have the **likelihood function**:

$$L(\beta) = \prod_{y_i=1} P\left(\varepsilon_i \geq -\left(\beta_1 + \beta_2 x_{2,i} + ... + \beta_k x_{k,i}\right)\right) \times \prod_{y_i=0} P\left(\varepsilon_i < -\left(\beta_1 + \beta_2 x_{2,i} + ... + \beta_k x_{k,i}\right)\right)$$

The notation Π signifies multiplication (like Σ signifies summation).

Since we observe the sample choices y_i = {0, 1, 1, 0 ,...} we choose that β that gives us the highest probability of actually obtaining those very choices in a sample of *n*. This is the **Maximum Likelihood** estimator:

$$\max_{\beta} L(\beta) \rightarrow \hat{\beta} \quad : \text{ if } \tilde{\beta} \text{ is any other value then } L(\hat{\beta}) \geq L(\tilde{\beta})$$

Of course, STATA will do this for us. We simply need to have a variable that represents the binary choice y_i = 0,1.

7.2 Probit Estimation for Binary Response Models

In order to estimate the marginal effects above for binary dependent variables (i.e. y_i = 0 or 1) we need to select a probability distribution function for *y*. *Probit* estimation refers to a normal distribution for ε_i.

Suppose we want model why mortality rates are high, defined as being above the U.S. average.

Use **probit** to estimate β.

egen mort_mu = mean(mort)	creates the state-wide mean mortality rate
gen mort_hi = mort > mort_mu	*mort_hi* = 1 if mort > mean; =0 otherwise
probit mort_hi inc_pc	

Remember β is the same as the marginal impact on the response probability (2).

Use **mfx compute** after **probit**, or use **dprobit** instead. Either computes

(3)

$$\frac{\partial}{\partial x_{j,i}} P(y_i = r \mid \bar{x}_1, ..., \bar{x}_k)$$

That is, the marginal impact of *x* on the response probability, evaluated at the "*average observation*".

STATA Results: PROBIT

```
. egen mort_mu = mean(mort)
. gen mort_hi = mort > mort_mu
. probit  mort_hi inc_pc pov ed_hs ed2_coll alc_pc tob_pc
```

Probit regression — Number of obs = 51
LR chi2(6) = 15.35
Prob > chi2 = 0.0177
Log likelihood = -27.194708 — Pseudo R2 = 0.2201

mort_hi	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
inc_pc	.0005012	.0001804	2.78	0.005	.0001476	.0008547
pov	14.00707	10.24506	1.37	0.172	-6.072867	34.08701
ed_hs	3.378205	5.251074	0.64	0.520	-6.913711	13.67012
ed2_coll	-24.955	11.68464	-2.14	0.033	-47.85647	-2.053533
alc_pc	-.7386295	.4144753	-1.78	0.075	-1.550986	.0737271
tob_pc	.009009	.01297	0.69	0.487	-.0164118	.0344298
_cons	-5.489341	5.337424	-1.03	0.304	-15.9505	4.971818

```
. mfx compute
```

Marginal effects after probit
y = Pr(mort_hi) (predict)
= .58218276

variable	dy/dx	Std. Err.	z	P>|z|	[95% C.I.]		X
inc_pc	.0001957	.00007	2.76	0.006	.000057	.000334	13249.2
pov	5.469022	4.00319	1.37	0.172	-2.37709	13.3151	.126176
ed_hs	1.31901	2.04916	0.64	0.520	-2.69727	5.33529	.674608
ed2_coll	-9.74361	4.56824	-2.13	0.033	-18.6972	-.790028	.163157
alc_pc	-.2883958	.16151	-1.79	0.074	-.60495	.028158	2.71275
tob_pc	.0035175	.00507	0.69	0.488	-.006415	.01345	120.527

Thus, dy/dx is (3) the marginal impact of *x* on $P(y = 1 \mid x = X)$ where X = sample mean of x.

Compare this with **dpobit**:

```
. dprobit  mort_hi inc_pc pov ed_hs ed2_coll alc_pc tob_pc
```

Probit regression, reporting marginal effects — Number of obs = 51
LR chi2(6) = 15.35
Prob > chi2 = 0.0177
Log likelihood = -27.194708 — Pseudo R2 = 0.2201

mort_hi	dF/dx	Std. Err.	z	P>|z|	x-bar	[95% C.I.]	
inc_pc	.0001957	.0000708	2.78	0.005	13249.2	.000057	.000334
pov	5.469022	4.003193	1.37	0.172	.126176	-2.37709	13.3151
ed_hs	1.31901	2.049161	0.64	0.520	.674608	-2.69727	5.33529
ed2_coll	-9.74361	4.568238	-2.14	0.033	.163157	-18.6972	-.790028
alc_pc	-.2883958	.1615102	-1.78	0.075	2.71275	-.60495	.028158
tob_pc	.0035175	.0050677	0.69	0.487	120.527	-.006415	.01345
obs. P	.5686275						
pred. P	.5821828 (at x-bar)						

z and P>|z| correspond to the test of the underlying coefficient being 0

In this case, STATA properly uses "x_bar" to symbolize the sample mean of x.

Note: dF/dx is exactly (3).

Notice the estimates of dF/dx identically match dy/dx from **probit** with **mfx compute**.

7.3 Logit Estimation for Binary Response Models

The **Logit** model is identical to the **Probit** model, except we assume a logistic distribution for ε_i.

There does note exist a command **dlogit**.

In order to compute (3), use **logit** and then **mfx compute**.

There is usually little difference between the results of **probit** and **logit**,, and many economists use both a as regular practice.

STATA Results: LOGIT

```
. logit mort_hi inc_pc pov ed_hs ed2_coll alc_pc tob_pc
```

```
Logistic regression                             Number of obs   =         51
                                                LR chi2(6)      =      15.51
                                                Prob > chi2     =     0.0166
Log likelihood = -27.111999                     Pseudo R2       =     0.2225

------------------------------------------------------------------------------
     mort_hi |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      inc_pc |   .0008466   .0003399     2.49   0.013     .0001805    .0015127
         pov |   22.74186   18.59145     1.22   0.221    -13.69672    59.18044
       ed_hs |   6.889178   9.476362     0.73   0.467    -11.68415    25.46251
    ed2_coll |  -45.06757   21.81899    -2.07   0.039    -87.83199   -2.303141
      alc_pc |  -1.294466   .7305029    -1.77   0.076    -2.726225    .1372932
      tob_pc |   .0158471   .0233882     0.68   0.498    -.0299928     .061687
       _cons |  -9.477995   9.834898    -0.96   0.335    -28.75404    9.798051
------------------------------------------------------------------------------
```

```
. mfx compute
```

```
Marginal effects after logit
      y  = Pr(mort_hi) (predict)
         =  .57478401
------------------------------------------------------------------------------
variable |      dy/dx   Std. Err.      z    P>|z|    [    95% C.I.   ]      X
---------+--------------------------------------------------------------------
  inc_pc |   .0002069    .00008     2.47   0.014    .000043  .000371   13249.2
     pov |   5.558278   4.53415     1.23   0.220    -3.3285  14.4451    .126176
   ed_hs |   1.683766   2.31494     0.73   0.467   -2.85344  6.22097    .674608
ed2_coll |  -11.01484   5.38944    -2.04   0.041    -21.578 -.451739    .163157
  alc_pc |   -.316377    .17854    -1.77   0.076   -.666317  .033563    2.71275
  tob_pc |   .0038731    .00571     0.68   0.498   -.007322  .015068    120.527
------------------------------------------------------------------------------
```

7.4 Likelihood Ratio Tests

In binary response models we use a latent variable y^* to link regressors x to the choice $y = 0,1$. We could then write out residuals, and hypothetically use them for F-tests, R^2, and so on:

$$\hat{\varepsilon}_i = y_i^* - \hat{\beta}_1 - \hat{\beta}_2 x_{2,i} - \cdots - \hat{\beta}_k x_{k,i}$$

There is, however, one problem: y^* is not observable. If it were, we would not need to use probit or logit! and simply estimate the latent regression model by OLS.

The solution is to use the likelihood function. In the same manner as performing an F-test with OLS estimation (estimate restricted and unrestricted models by OLS), we can perform a **Likelihood Ratio** test by estimating restricted and unrestricted models and comparing $L(\beta)$.

Consider the model and hypothesis:

$$y_i^* = \beta_1 + \beta_2 x_{2,i} + ... + \beta_5 x_{5,i} + \varepsilon_i$$

$$H_0 : \beta_2 = \beta_3 = 0 \text{ against } H_1 : \text{at least one} \neq 0$$

Estimate the model with all x's, then without $x_{2,i}$ and $x_{3,i}$. If the null is true the likelihood functions $L(\beta)$ should be similar. If the null is false then $L(\beta)$ with all x's will be potentially much larger (a better fit implies a larger likelihood of obtaining our sample of choices $y_i = 0,1$ with those x's included).

In STATA we must estimate both models, and store the results.

```
probit y x2 x3 x4
estimates store U

probit y x4
estimates store R

lrtest U R          /* that is an "ell" and note a "one"; in capitals LRTEST */
```

Notice you may name the models anything you want: U and R, A and B, or actual words "Unrestricted" and "Restricted" and so on.

7.5 Estimation of Regression Models with Sample Selection Bias

If people select, even if unknowingly, to be in a sample, then the sample cannot be said to be random. This causes the errors to have a non-zero mean, which causes the OLS estimators to be biased. Hence, "*sample selection bias*".

The classic example is any data set of working people (e.g. work hours, wages, education, gender, etc.). Clearly the people in this data set have, more or less, chosen to work. Hence, in some sense, they are available to be in the data set. All information about those who chose *not to work*, or could not find work, or lost a job recently and are presently unsuccessfully searching for a job, *is not included*. Because of this, we lack important information as to why those who *are* in the sample, *are in it*! This will bias any attempt to estimate a wage model, or work hour model, and so on.

Consider this: my wages (as an employed professor) are due to my previous work experience, my educationlevel, perhaps my gender (I hope not!), my race (I hope not!), and most basic of

all: the simple fact that I choose to work (nor work = no wages!). But a standard wage model does not control for *why I chose to work.*

Consider a simple wage model for working married women (the classic example in this research area):

$$wage_i = \beta_1 + \beta_2 ed_i + \beta_3 children_i + \varepsilon_{1,i}$$

$$\text{work if } \lambda_1 + \lambda_2 wage_i^h + \lambda_3 children_i + \varepsilon_{2,i} > 0$$

Thus, education and the number of children are assumed in impact wages, while the "decision" to work is affected by the husband's wage, and the number of children.

Just like the **probit** model for a binary response, the "*work if*" statement represents a "*latent variable*", something unobservable that underlies the decision to work. We might think of it as the net utility of working: if positive, then they work, and net working is affected by the husband's wage $wage^h$ and the number of children.

Use **heckman** to estimate this type of model.

gen wage_pos = (wage > 0) create 0,1 variable for working

heckman wage ed child, select(wage_pos = wage_h child)
models of wage, and decision to work (giving wage > 0)

STATA Results: HECKMAN

```
. gen wage_pos = (wage > 0)
. heckman  wage ed  child_6, select( wage_pos = wage_h child_6)
```

Heckman selection model — Number of obs = 753
(regression model with sample selection) — Censored obs = 325 **(# wage = 0)**
Uncensored obs = 428 **(# wage > 0)**

Wald chi2(2) = 32.80
Log likelihood = -1505.673 — Prob > chi2 = 0.0000

	Coef.	Std. Err.	z	P>|z|	[95% Conf. Interval]	
wage |						
ed |	.2176315	.0580694	3.75	0.000	.1038176	.3314454
child |	-1.763462	.3946485	-4.47	0.000	-2.536959	-.9899653
_cons |	-1.046753	.7357673	-1.42	0.155	-2.48883	.3953248
wage_pos |						
wage_h |	-.0001429	.0060373	-0.02	0.981	-.0119758	.0116901
child |	-.4217626	.0882115	-4.78	0.000	-.594654	-.2488712
_cons |	.1934339	.0664116	2.91	0.004	.0632697	.3235982
/athrho |	2.561785	.1884358	13.60	0.000	2.192458	2.931113
/lnsigma |	1.472406	.0409865	35.92	0.000	1.392074	1.552738
rho |	.9881611	.0044354			.975379	.9943263
sigma |	4.359712	.1786895			4.023185	4.724388
lambda |	4.308098	.1862695			3.943016	4.673179

LR test of indep. eqns. (rho = 0): chi2(1) = 165.22 Prob > chi2 = 0.0000

Wage Model

Work Decision Model

Note: **rho** is the sample correlation of the wage error $\varepsilon_{1,i}$ and decision to work error $\varepsilon_{2,i}$.

Thus, education is associated with higher wages, and children with lower wages. Both the husband's wage and presence of children are negatively associated with the likelihood a woman works.