

Introduction to STATA

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

Topic	Page
1. Getting Started	3
2. Loading Data (Command Window) Uploading data from a <i>text file</i> Saving a STATA data file “.dta” Create new variables (generate, drop, replace, scalar; lags)	
3. Data Analysis (Command Window)	9
3.1 Describing data – Sample statistics (summarize)	
3.2 Storing the sample mean, var., sample size (egen, mean, min, max, median, skew, count)	
3.3 Display a variable (list)	
3.4 Correlation analysis (correlate)	
3.5 Normality Tests (sktest, ksmirnov)	
3.6 Plotting data – scatter plots, histogram (graph, scatter, twoway)	
3.7 Hypothesis testing – mean, variance, correlation (ttest, prtest, sdtest, spearman)	
4. Linear Regression from the Command Window	20
Ordinary Least Squares – Linear model estimation (regress)	
Listing output (estimates list, hold, unhold)	
Residual analysis (predict ehat, resid)	
Plotting Regression Results	
Forecasting (predict yhat)	
Hypothesis testing – Test linear and nonlinear restrictions	
Tests of omitted variables/structure: <i>RESET</i> test (ovtest)	
Test model goodness of fit (linktest)	
5. Programming in STATA – The Do-File “.do”	29
Basic syntax - the core program (load a stata data file or text file)	
Regression and Post-regression analysis	
6. Generalized Linear Model	31
Robust Estimation: Robust Standard Errors	
Heteroscedasticity – Test, Estimation	
Correlated Errors – Test, Estimation	
List of Examples	2
Data Explanation	2

Example List	Page
1. View the various windows, <i>help</i> .	4
2. Create a stata data file, create a variable, delete a variable, re-load the data (cd, use, generate, drop)	8
3. Kolmogorov-Smirnov test of standard normality on U.S. GDP (ksmirnov)	14
4. Estimate and analyze model of simulated data (with known parameter values) using OLS (regress, test, testnl, predict, ovtest, sktest, ksmirnov, linktest).	27
5. Do-File code for estimating <i>mortality</i> model1 (clear, cd, log, summarize, regress, predict, test, testnl, ovtest, linktest)	30

Data Explanation

This document makes repeated use of two datasets.

- 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 STATA data file is available at <http://www.unc.edu/~jbhill/gdp.dta>.

- 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 STATA data file is available at www.unc.edu/~jbhill/mort.dta.

1. GETTING STARTED

Boot-up STATA

There 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 is listed in the **Review** window

The output generated from your 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 your 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**.

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, $ed \neq ED \neq Ed \neq eD$.

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(##).
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 your 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*:

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

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

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

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*:

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

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

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 $(1 - \alpha)\%$ *confidence band* of the *sample mean* of an iid X_i :

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

Stata Results for Summarize:

```
.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 SUMMARIZE, DETAIL:

```
. summarize GDP, detail
```

GDP		
Percentiles	Smallest	
1%	1624.5	1590.6
5%	1687.3	1598.2
10%	1855.7	1609.7
25%	2573.45	1620.2
50%	3700.45	
	Largest	
75%	5100.85	7073.9
90%	6210.5	7079.8
95%	6686.5	7100.9
99%	7057.4	7121.5

Obs	528
Sum of Wgt.	528
Mean	3907.111
Std. Dev.	1537.238
ariance	2363102
Skewness	.2819017
Kurtosis	2.001144

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 $\{X_i, Y_i\}$:

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

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 **spearman**:

spearman var1 var2 only two variables at a time!

Stata Results **PWCORR**:

```
. 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	
	<i>0.0000</i>		
t_bill	0.0400	-0.2857	1.0000
	<i>0.3588</i>	<i>0.0000</i>	

The *p-values* of tests of $\text{corr} = 0$ 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 (a 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

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

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

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

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 to 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)
```

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 –Scatter Plots (**graph**, **scatter**, **twoway**) : **GRAPH**, **PLOT**

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

The plot is displayed in a separate **STATA Graph** window.

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

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

“**gr**” is the same as “**graph**”

graph GDP by default produces a *histogram*. **graph GDP M1** by default produces a *scatter* plot.

plot GDP M1 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**.

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

TITLE, LABELS, AXES, etc.

See STATA helps screen: search on **graph**.

3.7 Histograms and Distribution Plots : GRAPH, KDENSITY

graph produces a histogram

graph GDP, bin(10) produces a histogram with 10 X-axis ranges (must be 2-50).

graph GDP, normal produces a histogram and overlays a normal distribution with GDP's sample mean and variance.

kdensity GDP produces a sample approximation of the distribution of GDP.

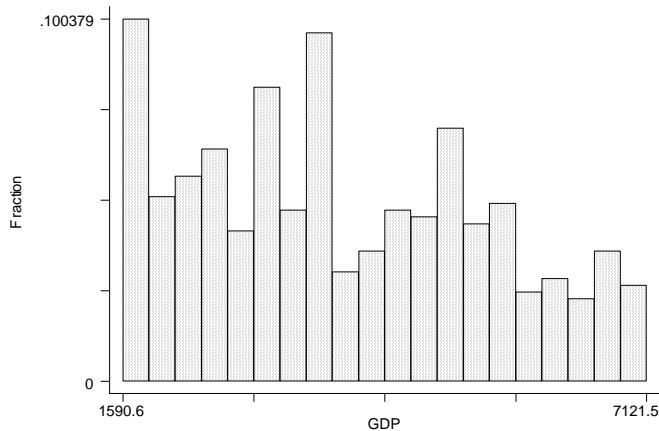
qnorm GDP plots the quantiles of GDP against the quantiles of a normal.

pnorm GDP plots empirical probabilities of standardized GDP against probabilities of a normal.

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

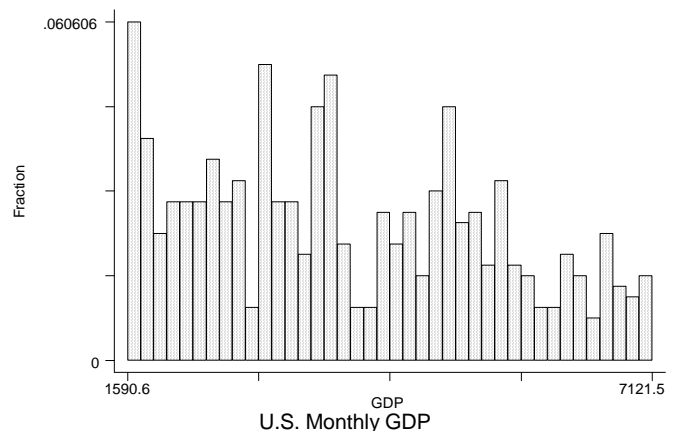
Stata Results from GRAPH:

`.graph GDP, bin(20)`

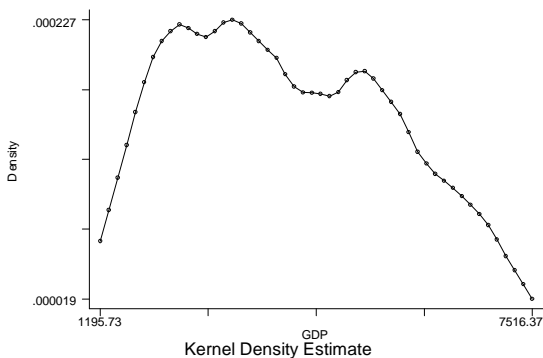


Stata Results GRAPH, NORMAL, TITLE:

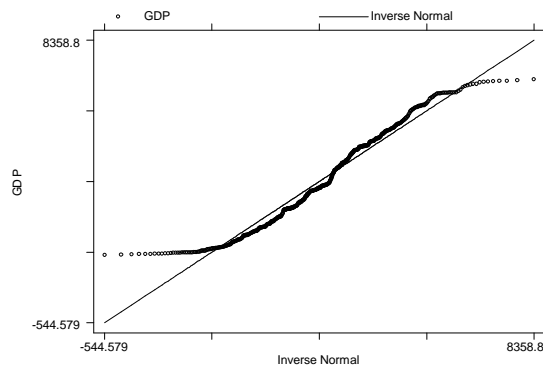
`. graph GDP, bin(40) normal title("GDP")`



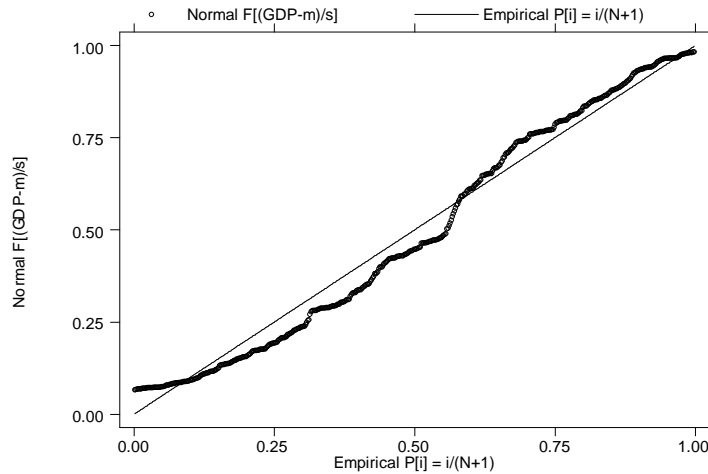
Stata Results from KDENSITY:



Stata Results from QNORM



Stata Results from PNORM:



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 rank 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

t = 26.1691

P < t = 1.0000

Ha: mean ~ = .4

t = 26.1691

P > |t| = 0.0000

Ha: mean > .4

t = 26.1691

P > t = 0.0000

The results imply a one-sided test $mean(ed_hs) \leq .4$ 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

z = -2.871

P < z = 0.0020

Ha: ed_hs_80 ~ = .2

z = -2.871

P > |z| = 0.0041

Ha: ed_hs_80 > .2

z = -2.871

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: $mean(ed_hs_80) \geq .2$ 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: $var(ed2_coll) \geq var(ed_hs)$ 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

$$y_i = \beta_0 + \beta_1 x_{i,1} + \dots + \beta_k x_{i,k} + \varepsilon_i, \quad \varepsilon_i \stackrel{iid}{\sim} (0, \sigma^2)$$

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 income ed_hs ed_coll alc tob
```

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

A constant term is automatically included.

Stata Results from **REGRESS**:

```
. regress mort income ed_hs ed_coll alc tob
```

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

mort	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]	
income	.0241829	.0120987	2.00	0.052	-.0001851	.048551
ed_hs	-876.7612	306.6727	-2.86	0.006	-1494.432	-259.0906
ed_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: $t = \text{Coef}/\text{Std.Err.}$

F(...) is the classical F-test statistic.

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, $y_i = \beta_1 x_{i,1} + \dots + \beta_k x_{i,k} + \varepsilon_i$ use **noconstant**:

regress mort income ed_hs ed_coll alc tob, noconstant

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

quietly regress mort income ed_hs ed_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. If you want to turn output back on, use **noisily**.

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

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

regress mort income ed_hs ed_coll alc tob
estimates 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 income ed_hs ed_coll alc tob
estimates hold mort_model_1

regress mort income ed_hs ed_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, 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.

If you forgot the model names you put on **estimates hold**, use **estimates dir** which lists all **estimates hold** model names.

Stata Results ESTIMATES LIST:

```
. estimates 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*

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 $\hat{V}[\hat{\beta}' X_i | X_i]$, type

predict *anothervarname*, stdp

For residuals and standard error of residuals, type

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

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

4.5 Plotting Regression Results : AVPLOT, AVPLOTS, RVPLOT

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)
```

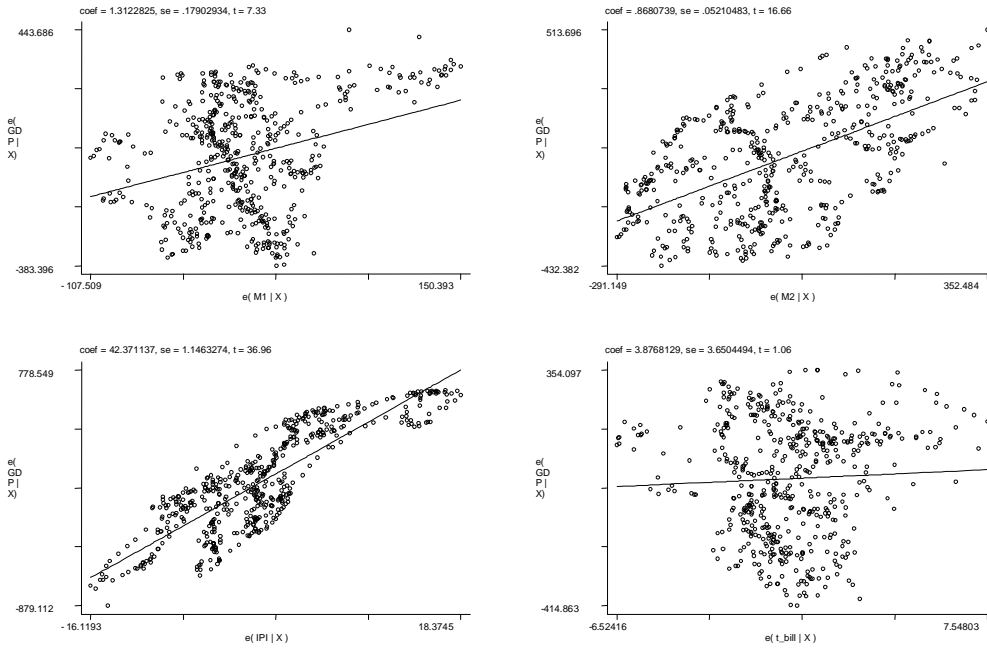
2. Residuals Plots - RVPLOT

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 endogeneity.

```
regress y x1...xk  
rvpplot varx       (list one regressor)
```

Stata Results AVPLOTS:

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

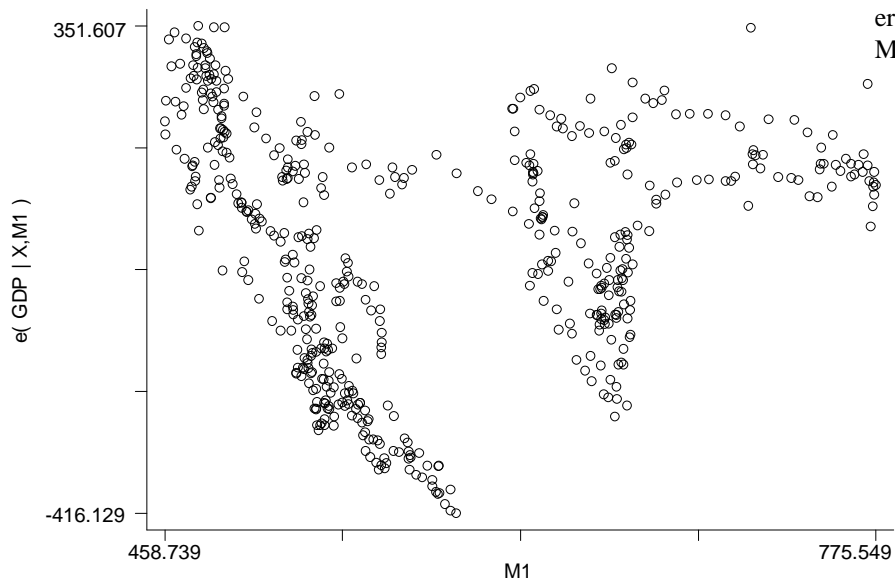


The X-axis contains the regressor x after controlling for its relationship with *other* regressors; the Y-axis contains GDP after controlling for its relationship with the *remaining* regressors.

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 RVPVLOT:

```
. quietly regress GDP M1 M2 IPI 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 *ed_coll* jointly have zero slopes:

```
test ed_hs ed_coll
```

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

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

```
test ed_hs + ed_coll = 0
```

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

```
test ed_hs=ed_coll
```

Test whether a parameter equal a specific value:

```
test income = .025
```

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

```
test _b[ed_hs] = _b[ed_coll]
```

is identical to

```
test ed_hs = ed_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 `ed_coll` have the same marginal effect can be performed as

`testnl _b[ed_hs]/_b[ed_coll] = 1` also tests for parameter identity

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

Stata Results TESTNL:

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

```
(1) _b[ed_hs]/_b[ed_coll]=1
```

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

4.8 Test of Omitted Variables/Structure

In order to test for omitted structure, the Ramsey RESET (Regression Specification Error Test) command is popular (but hugely 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 on powers of the regressors.

Stata Results OVTEST:

```
. 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
```

```
. 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)
```

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		
Model	359011.659	2	179505.829	F(2, 48) = 14.54		
Residual	592718.575	48	12348.3037	Prob > F = 0.0000		
-----				R-squared = 0.3772		
Total	951730.234	50	19034.6047	Adj R-squared = 0.3513		
-----				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

EXAMPLE #4 (OLS on simulated data)

In order to gauge how well OLS works, and how informative STATA's regression 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		
Model	2973.98538	2	1486.99269	F(2, 97)	=	1271.89
Residual	113.404579	97	1.16911937	Prob > F	=	0.0000
				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 R² > .96.

. predict ehat,resid **(generate residuals)**
 . sktest ehat **(tests is residuals are normally distributed)**

Skewness/Kurtosis tests for Normality

Variable	Pr(Skewness)	Pr(Kurtosis)	adj chi2(2)	Prob>chi2
ehat	0.850	0.284	1.21	0.5458 (the p-values > .20)

. ksmirnov ehat = normprob(ehat)

Smaller group	D	P-value	Corrected
ehat:	0.0636	0.446	
Cumulative:	-0.0552	0.543	
Combined K-S:	0.0636	0.814	0.779 (p-values > .75)

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

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

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, income, tob,alc, ed_hs, ed_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 from 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 */

summarize              /* summary statistics */
regress mort income ed_hs ed_coll alc tob

predict yhat
predict ste_yhat, stdp
predict ehat, resid

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

$$y_i = \beta_0 + \beta_1 x_{i,1} + \dots + \beta_k x_{i,k} + \varepsilon_i$$

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

$$y_i = \beta_0 + \beta_1 x_{i,1} + \dots + \beta_k x_{i,k} + \varepsilon_i \text{ where } \varepsilon_i \text{ is independent, } E[\varepsilon_i] = 0 \text{ but } V[\varepsilon_i] = \sigma_i^2$$

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

$$y_t = \beta_0 + \beta_1 x_{t,1} + \dots + \beta_k x_{t,k} + \varepsilon_t \text{ where } \varepsilon_t \sim (0, \sigma^2) \text{ but maybe } \text{corr}(\varepsilon_t, \varepsilon_{t-h}) \neq 0 .$$

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 Heteroscedasticity: Robust Estimation

Suppose the regression model is

$$y_i = \beta_0 + \beta_1 x_{i,1} + \dots + \beta_k x_{i,k} + \varepsilon_i, \varepsilon_i \text{ is independent, } E[\varepsilon_i] = 0 \text{ but } V[\varepsilon_i] = \sigma_i^2$$

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.

6.2 Heteroscedasticity: Tests

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

6.2.1 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)
```

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

1. $V[\varepsilon_i] = \gamma_0 + \gamma_1 x_{i,1} + \dots + \gamma_k x_{i,k}$
2. $V[\varepsilon_i] = (\gamma_0 + \gamma_1 x_{i,1} + \dots + \gamma_k x_{i,k})^2$
3. $V[\varepsilon_i] = \exp\{\gamma_0 + \gamma_1 x_{i,1} + \dots + \gamma_k x_{i,k}\}$

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)
```

Stata Results HETTEST:

```
. 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 ed_hs ed2_coll alc_pc tob_pc
```

Cook-Weisberg test for heteroskedasticity using variables specified

Ho: Constant variance chi2(4) = 10.89

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

6.2.2 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.

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

```
egen m_inc = median(inc_pc)           (create mean of income)
generate hi_lo_inc = (inc_pc < mu_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 **ROBVAR, BY:**

```
. egen m_inc = median(inc_pc)
. generate hi_lo_inc = (inc_pc < mu_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 |                18.161149   136.75345    24
1 |                -16.143243   94.834805    27
-----+-----
Total |      -1.028e-07   116.52652    51

W0 = 1.4313306   df(1, 49)   Pr > F = .23730436
W50 = 1.3622181   df(1, 49)   Pr > F = .24880196
W10 = 1.3530487   df(1, 49)   Pr > F = .25037927
```

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

6.3 Heteroscedasticity: FGLS Estimation

The robust least squares estimator discussed 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 Estimator.

If the model is

$$y_i = \beta_0 + \beta_1 x_{i,1} + \dots + \beta_k x_{i,k} + \varepsilon_i, \varepsilon_i \text{ is independent, } E[\varepsilon_i] = 0 \text{ but } V[\varepsilon_i] = \sigma_i^2$$

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

$$\frac{y_i}{\hat{\sigma}_i} = \beta_0 \frac{1}{\hat{\sigma}_i} + \beta_1 \frac{x_{i,1}}{\hat{\sigma}_i} + \dots + \beta_k \frac{x_{i,k}}{\hat{\sigma}_i} + \frac{\varepsilon_i}{\hat{\sigma}_i}$$

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.

6.3.1 Multiplicative Heteroscedasticity

If variance is a multiplicative scale of some positive variable x_j

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

use

regress y x1...xk [aw = xj]

6.3.2 Linear Heteroscedasticity

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

1. $V[\varepsilon_i] = \gamma_0 + \gamma_1 x_{i,1} + \dots + \gamma_k x_{i,k}$
2. $V[\varepsilon_i] = (\gamma_0 + \gamma_1 x_{i,1} + \dots + \gamma_k x_{i,k})^2$
3. $V[\varepsilon_i] = \exp\{\gamma_0 + \gamma_1 x_{i,1} + \dots + \gamma_k x_{i,k}\}$

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]
(sum of wgt is 6.7571e+05)
```

(use income as scale)

Source	SS	df	MS	Number of obs =	51
Model	300585.113	4	75146.2783	F(4, 46) =	4.73
Residual	730656.759	46	15883.8426	Prob > F =	0.0028
				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	
Model	424785.524	4	106196.381	Number of obs = 51
Residual	816578.28	46	17751.7017	F(4, 46) = 5.98
Total	1241363.80	50	24827.2761	Prob > F = 0.0006
				R-squared = 0.3422
				Adj R-squared = 0.2850
				Root MSE = 133.24

	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]	
mort						
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

6.3 Correlated Errors

Consider a linear regression model

$$y_t = \beta_0 + \sum_{i=1}^k \beta_i x_{i,t} + \varepsilon_t \quad \sigma_\varepsilon^2 = V(\varepsilon_t)$$

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 autocorrelated¹ ($\rho_h \neq 0$ for some lag h) then OLS estimates will be biased if the regressors x_t contain lags of the dependent variable y_t .

¹“Autocorrelation” and “serial correlation” are synonymous terms for intertemporal correlation in one time series variable.

The Sample Autocorrelation Function (SACF) is simply

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

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.

A convenient (and efficient) way to test for autocorrelation is to test multiple sample autocorrelations $\hat{\rho}_h$ 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 $I = 1, \dots, h$ for some logical horizon (e.g. 12 months for monthly data; 4 quarters for quarterly data)

6.3.1 Declaring 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”: $t = 1, 2, \dots, n$.

tsset t declares the data are a time series, and *var_time* represents time.

6.3.1 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 $i = 1, \dots, ##$.

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

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

corrgram var1, lags(12) level(99)

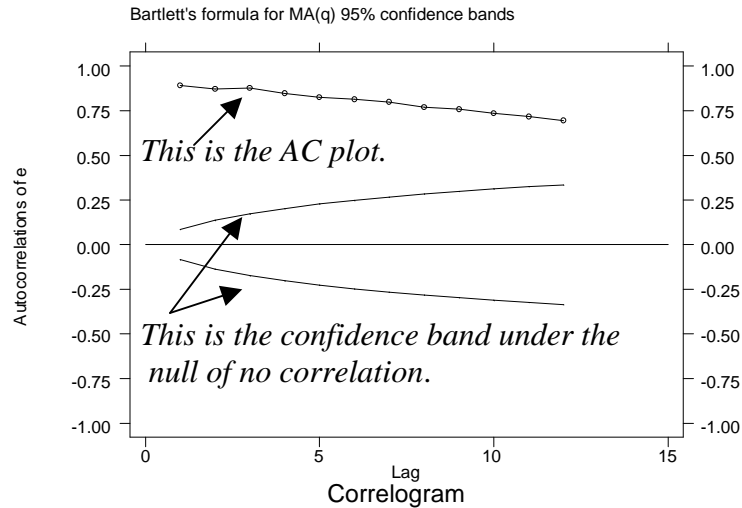
Stata Results **CORRGRAM**:

```
. quietly regress GDP M1 M2 IPI t_bill      (OLS without output)
. 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:



6.3.2 Testing for Autocorrelation

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

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. 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; 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!

As in all time series cases be sure to create a time variable and declare it via **tsset**.

Stata Results WNTESTO:

```
. 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 DWSTAT:

```
. 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.

6.3.3 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(##)*.

Be sure to create a time variable and declare it via **tsset**.

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.

STATA Results: PRAIS

```
. prais GDP M1 M2 IPI t_bill, corc twostep  
Iteration 0: rho = 0.0000  
Iteration 1: rho = 0.8921
```

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

Source	SS	df	MS	
Model	12988417.4	4	3247104.36	Number of obs = 527
Residual	1983939.80	522	3800.65096	F(4, 522) = 854.35
Total	14972357.3	526	28464.5575	Prob > F = 0.0000

R-squared = 0.8675
Adj R-squared = 0.8665
Root MSE = 61.649

GDP	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]
M1	-2.610289	.2213952	-11.79	0.000	-3.045224 -2.175353
M2	1.261453	.0725134	17.40	0.000	1.118999 1.403907
IPI	37.41817	1.59408	23.47	0.000	34.28657 40.54977
t_bill	-11.94446	3.220476	-3.71	0.000	-18.27115 -5.617777
_cons	505.5775	112.5338	4.49	0.000	284.5028 726.6522

rho | .8921201

Durbin-Watson statistic (original) 0.212703
Durbin-Watson statistic (transformed) 1.920640