EViews 4.1 Tutorial

by Manfred W. Keil

to Accompany

Introduction to Econometrics

by James H. Stock and Mark W. Watson

1. EVIEWS: INTRODUCTION

This tutorial will introduce you to a statistical and econometric software package called EViews, or Econometric Views. EViews runs on both the Windows (9x, Me, NT 4.0 or 2000) and Macintosh platforms. It is produced by Quantitative Micro Software (QMS) in Irvine, California. You can read about various product information at the firm's Web site, www.eviews.com. The program comes with two manuals, a *User's Guide* and a *Command and Programming Reference*. Both manuals can be ordered separately ($40 each, $75 for the pair) by calling (949) 856-3368 or writing to sales@eviews.com. The *User's Guide* is better for first-time users. There is also a cheaper student version (EViews 3.1 Student Version for Windows 9x, 2000, NT 4.0, $39.95) and EViews Basics available. The difference between the student version and the full version is in the limitation on the size of data sets ("capacity limitation" is 1,000 observations for each series and no more than 10,000 observations for all series) and the availability of some estimation methods such as ARCH (discussed in Chapter 14), GMM, SURE, FIML, 3SLS, FIML and TSLS system estimators (don't worry what these estimation methods stand for at this point), as well as some of the estimators discussed in the appendix to Chapter 9. Furthermore, and perhaps most importantly for you right now, the student version does not allow you to run EViews in "batch mode." This tutorial will explain the difference between interactive use and batch mode below. Once you have gone through the first series of commands in interactive, you will almost certainly run programs in batch mode.[1]

Econometrics deals with three types of data: cross-sectional data, time series data, and panel data, or longitudinal data (see Chapter 1 of the textbook). In a *time series* you observe the behavior of a single entity over multiple time periods. This can range from high frequency data such as financial data (hours, days); to data observed at somewhat lower (monthly) frequencies, such as industrial production rates and inflation and unemployment rates; to quarterly data (GDP)

[1] All results in this tutorial were computed using EViews 4.1. If you have purchased the EViews 3.1 Student Version and/or use EViews in labs/networks that have not upgraded from the previous version, then you should use the EViews 3.1 tutorial, also found at the Stock and Watson Web site. Some of the commands differ between the two versions and there are some features missing from Version 3.1, such as the DF-GLS test discussed in Chapter 14 and some statistical features.

or annual (historical) data. In a *cross-section* you analyze data from multiple entities at a single point in time. One big difference between time series and cross-sectional analysis is that the order of the observation numbers does not matter in cross-sections. With time series, you would lose some of the most interesting features if you shuffled the observations. Finally, *panel data* can be viewed as a combination of time series and cross-sectional data, since multiple entities are observed at multiple time periods. EViews allows you to work with all three types of data.

EViews is the most commonly used econometrics package for time series analysis in academics, business, and government. It can also be used for cross-section and panel data. EViews allows you to save results within a program and to "retrieve" these results for further calculations. Remember how you calculated confidence intervals in statistics say for a population mean? Basically you needed the sample mean, the standard deviation, and some value from a statistical table. In EViews you can calculate the mean and standard deviation of a sample and then temporarily "store" these. You then work with these numbers in a standard formula for confidence intervals. In addition, EViews provides the required numbers from the relevant distribution (normal, χ^2, F, etc.).

While EViews is truly interactive, you can also run a program as a "batch" job, i.e., you write a sequence of commands and then execute the program in one go. In the good old days the equivalent was to submit a "batch" of cards, each containing a single command, to a technician, who would use a card reader to enter these into the computer, and the computer would execute the sequence of statements. While you will work at first in interactive mode by clicking on buttons, you will very soon discover the advantage of running your regressions in batch mode. This method allows you to see the history of commands, and you can also analyze where exactly things went wrong if there are problems with any of your commands. This tutorial will initially explain the interactive use of EViews, since it is more intuitive. However, we will switch as soon as it makes sense into the batch mode.

EViews is not particularly good at graphing; actually, the graphs look a bit unprofessional. If you need to graph data, it is best to save it in a spreadsheet or ASCII format, which EViews allows you to do, and then to import the data into Excel (or another spreadsheet program you prefer). Even better, since EViews works in a Windows format, it allows you to cut and paste the data into any other Windows-based program.

Finally, there is a warning about the limitations of this tutorial. The purpose is to help you gain an initial understanding of how to work with EViews. I hope that the tutorial looks less daunting than the manuals. However, it cannot replace the accompanying manuals, which you will have to consult for more detailed questions (alternatively use "Help" in the program). Feel free to provide me with feedback of how we can improve the tutorial for future generations of students (manfred.keil@claremontmckenna.edu).

2. CROSS-SECTIONAL DATA

<u>Interactive Use</u>

Let's get started. Click on the EViews icon to begin your session. What you see next is the EViews window, with the *title bar* at the top, the *command window* immediately below and the *status line* at the very bottom.

(Your command window may appear smaller, but you can enlarge it using the usual windows methods.)

The results of your various operations will be displayed between the command window and the status line in the so-called *work area*. In interactive use, EViews allows you to execute commands either by clicking on command buttons or by typing the equivalent command into the *command window*.

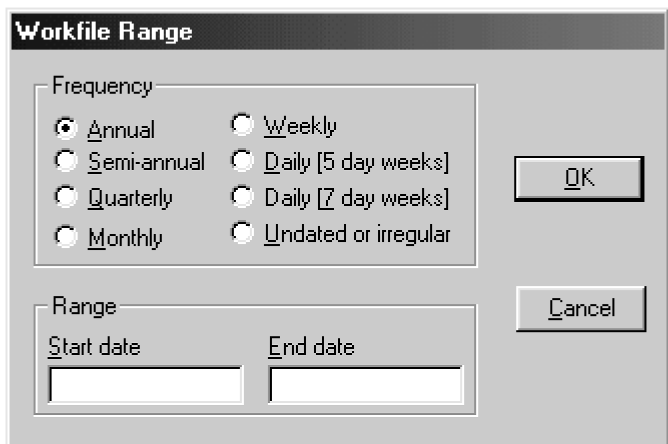
In this tutorial, we will work with two data applications, one cross-sectional (student test scores and student-teacher ratio) and one time series (forecasting inflation).

Data Input and Simple Data Analysis

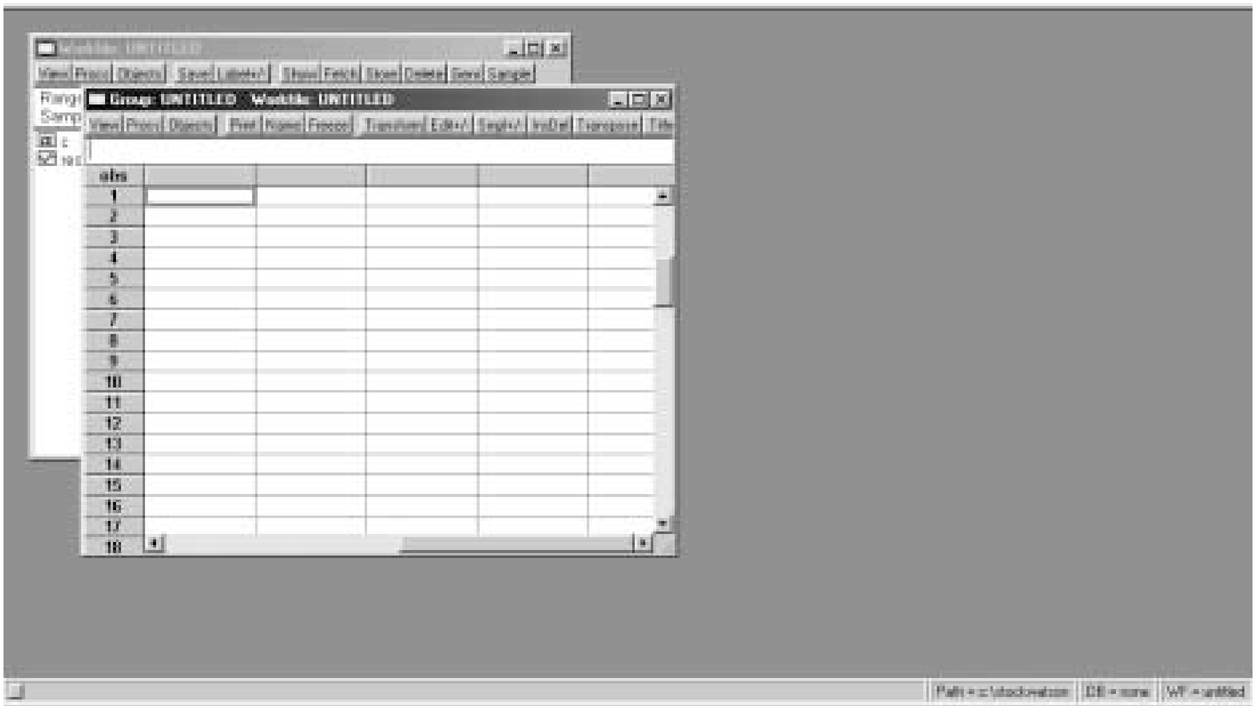
a) The Easy and Tedious Way: Clicking and Entering

In Chapters 4 to 7 you will work with the California test score data set. These are cross-sectional data, referred to in EViews as "undated or irregular" data. There are 420 observations from K-6 and K-8 school districts for the years 1998 and 1999. You will not want to enter a large amount of data manually, but for the purpose of this introduction it will be useful that you become aware of this possibility. As a result, I will use a sub-sample of 20 observations from this data set.

To start, we must establish a *workfile* in EViews. Click on the **File** pull-down menu, and then on **New** and **Workfile**. As is common in Windows programs, you will see a *dialog box*.



This particular dialog box asks you for the start and end dates of your data set, and for the type of data you are entering. We are working with undated or irregular data, so check this field and enter 20 in the "End date" box. You will see a workfile window, which contains two entries. Do not worry about these for the moment. To enter the data into a format similar to the spreadsheets you have become familiar with, click on **Quick** in the title bar, and then on **Empty Group (Edit Series)**.



Next enter the variables, starting with the name, in the grey box to the immediate right of "obs." (Click on the grey box and enter first "testscr" and "str" in the adjacent one to the right.)

Manual Data Entry

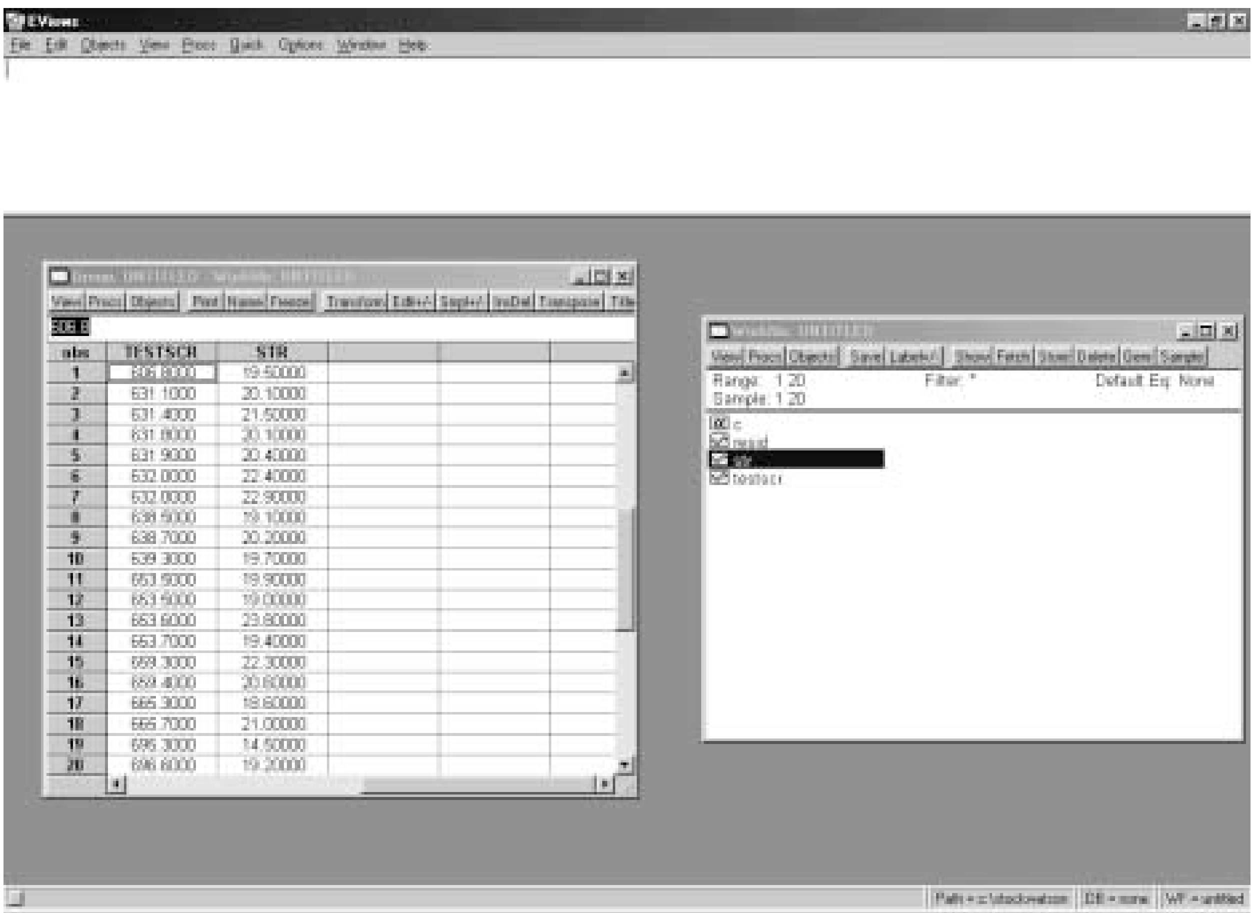
Here are the data to enter. (EViews will add zeros. You will see later how to get rid of these.)

obs	TESTSCR	STR
1	606.8	19.5
2	631.1	20.1
3	631.4	21.5
4	631.8	20.1
5	631.9	20.4
6	632.0	22.4
7	632.0	22.9
8	638.5	19.1
9	638.7	20.2
10	639.3	19.7
11	653.5	19.9
12	653.5	19.0

13	**653.6**	**23.8**
14	**653.7**	**19.4**
15	**659.3**	**22.3**
16	**659.4**	**20.6**
17	**665.3**	**18.6**
18	**665.7**	**21.0**
19	**695.3**	**14.5**
20	**696.6**	**19.2**

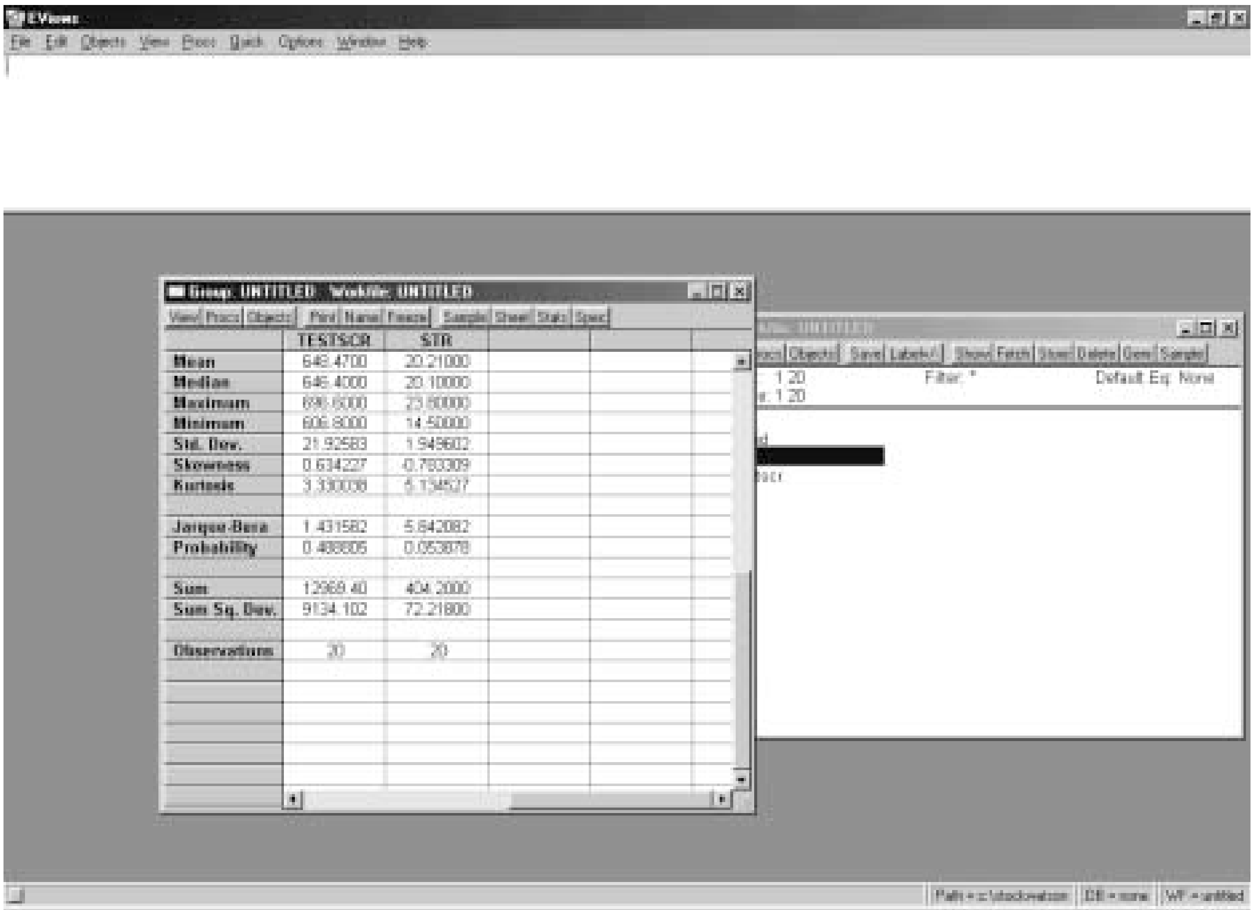
Entering data in this way is very tedious, and you will make data input errors frequently. You will see below how to enter data directly from a spreadsheet or an ASCII file, which are the most common forms of data you will receive in the future. Also, you noticed when you entered the test score (*testscr*) first and then the student-teacher ratio (*str*) that you were automatically moved into the test score column after entering each student-teacher data point. This is an unfortunate feature, but there is no alternative unless you enter all the data by observation.

This is what you should see after completing the data entry:



Summary Statistics

For the moment, let's just see if we are working with the same data set. Locate the **View** button at the upper-left corner of the workfile, click on it, and then click on **Descriptive Statistics** and **Common sample**. You should see the following output:



If your summary statistics differ, then check the data again. Once you have located the data problem, click the **Edit+/-** button on the workfile toolbar, move to the observation in question, enter the correct value, and press Enter. You may want to explore some of the other toolbar buttons to see their functions. **Number**, for example, allows you to get rid of unnecessary digits after the decimal point, but appears only after you "freeze" the object. Once you freeze an object, you can cut and paste it into your word processing file.

Once you have entered the data, there are various things you can do with it. First, let's get back to the data. Either click on **View** and then choose **Spread Sheet**, or simply click on the **Sheet** button. This allows you to see the data again. You may want to keep a hard copy of what you just entered. If so, click on the **Print** button.

In general, it is a good idea to save the data and your work frequently in some form. Many of us have learned through painful experiences how easy it is to lose hours of work by not backing up data/results in some fashion. There are two ways to save data in EViews. One is to save an entire workfile (**Save**), and the other is to store individual series (**Store**).

Press the **Save** button in the workfile toolbar, or click on **File** and then **SaveAs** in the *main menu*. Follow the usual Windows format for saving files (drives, directories, file type, etc.). If you save workfiles in EViews readable format, then you should use the extension ".WF1." Once you have saved a workfile, you can call it up the next time you intend to use it by clicking on **File** and then **Open**. Try these operations by saving the current workfile under the name "SW20smpl.wf1."

Alternatively, you may want to just save a few series of the current workfile. The reason is that sometimes you use some of these original series, or transformations of these series, in a different workfile. Let's save the test score and student-teacher series. First mark the two series in the workfile by clicking on *testscr*, then hold down the control button and click on *str*. (Make sure that you are doing this in the Workfile window, not in the Group View window.) After that, press the **Store** button in the workfile toolbar. Once again, a dialog box will pop up. Store the two data series in the EViews subdirectory with the extension ".db." Next time you need to retrieve these two series, you can simply click on the **Fetch** button in the workfile toolbar.

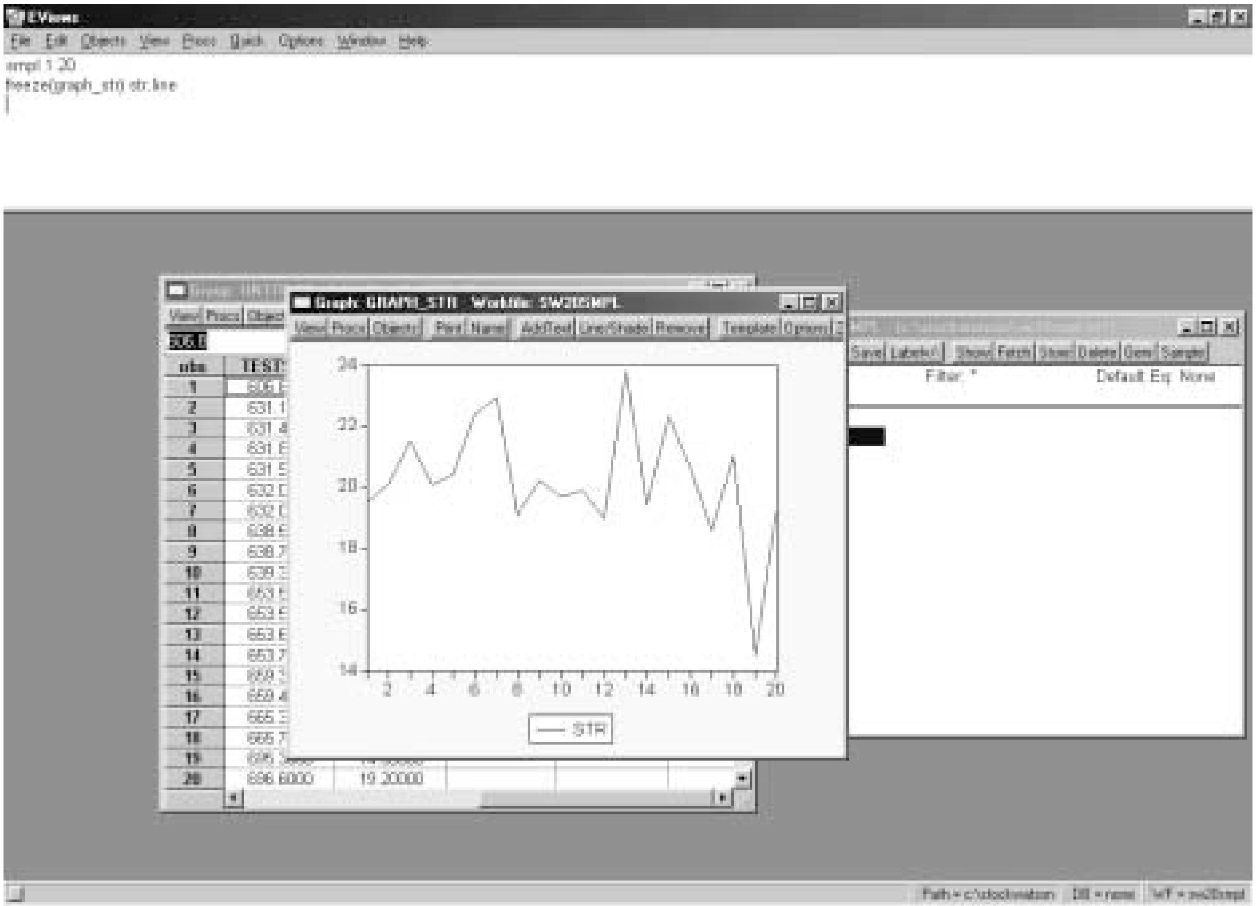
Graphical Presentations

Most often it is a good idea to generate graphs ("pictures") to get some "feel" for the data. Although Eviews offers many graphing options, there are two that you will use most often: line graphs, where one or more variables are plotted across entities, and scatterplots (crossplots), where one variable is graphed against another.

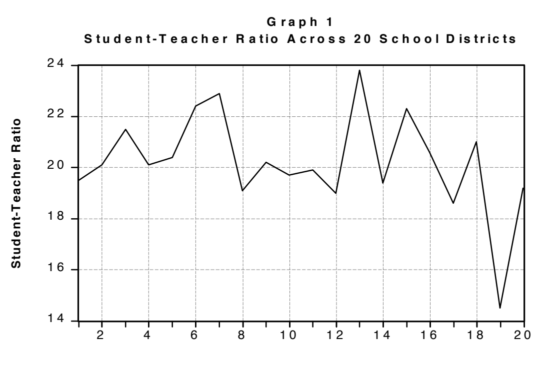
First set the sample to 1-20 either by clicking on the **Sample** button in the workfile toolbar or by entering "smpl 1 20" in the command line. Then type, in the next line, the command "freeze(graph_str) str.line" in the command line.[2] This will create the line graph and give it a name (graph1 here, but other names, such as Graph_1 or mygraph, can be chosen instead). Think of freezing an object as taking a photograph of it and giving it a name. This allows you to locate it easily in your photo album later. You can still edit the photograph later. Most importantly, you can cut and paste it into your word-processing program. "graph_str" now appears in the *workfile* window. Double click on it to see the graph you just created. In the future, in interactive use you will most often work in the *command window* rather than clicking on buttons.

---

[2] Alternatively the same graph can be generated by marking the variable *testscr* first and then double clicking on it. In the resulting *Series* window, click on **View /Graph/Line**. You can then freeze the graph by clicking on the **Freeze** button.



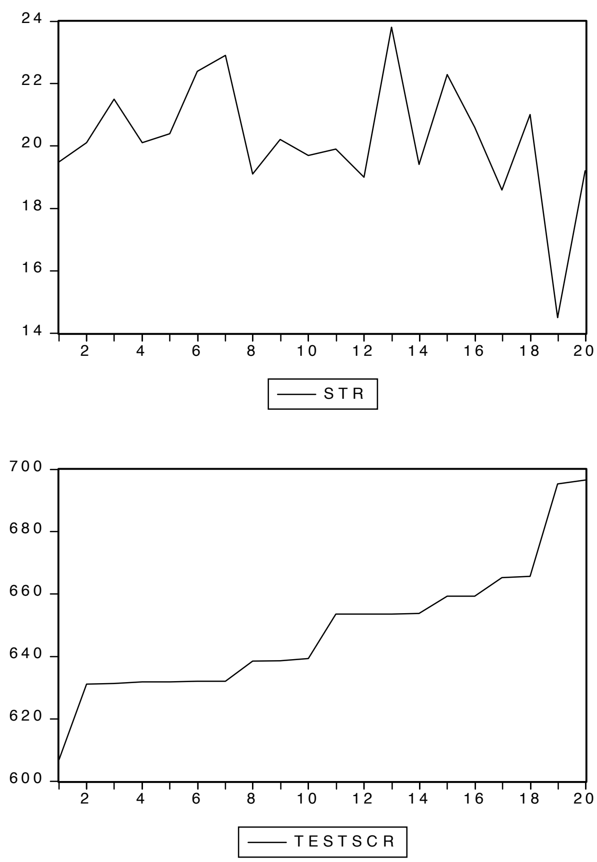
After the graph appears, either double click on the graph or click on the **Options** button, and alter it until it looks like the one below. Some of the alterations can be made in the resulting dialog box; others, such as text inserted, title of the graph, etc., have to be edited in.



Because in general we are interested either in causal relationships between variables or in the ability of one variable to forecast another, it is a good idea to plot two variables together. Commands, such as line, can often be modified by an option in parentheses. In this case, "m" means "display multiple graphs." Use the line command to generate the graph below.[3] This will require you first to define or create a "group" by giving it a name (here *size_perform* but others, such as *mygroup* are possible). Next you tell the program which series form the group, here *str* and *testscr*. Then "freeze" the graph as before.

The line commands are

group size_perform str testscr
freeze(two_series_plot) size_perform.line(m)



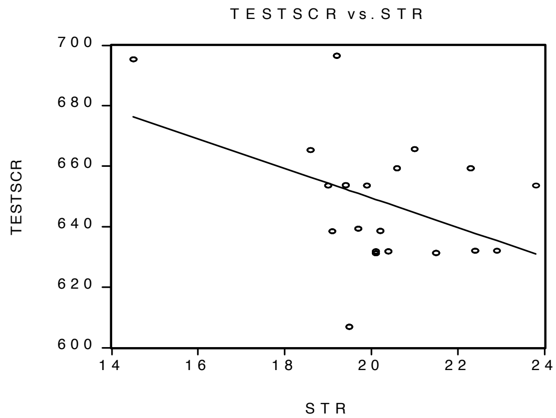
To get an even better idea about the relationship, you can display a two-dimensional relationship in a scatterplot. The command is size_perform.*linefit*, where *size_perform* refers to the name of a previously created group.[4] In the resulting *Global Fit Options* Box, choose the default

---

[3] Pushing buttons is relegated to footnotes from here on. You should work with commands now. If you have to, mark *testscr* and *str*, opening the two variables as a group, then select **View/Multiple Graphs/Line**).

[4] In the previous graph, select **View/Graph/Scatter/Scatter with Regression**. Choose *None* in the resulting *Global Fit Options* Box.

(*None*).



Simple Regression

There is a commonly held belief among many parents that lower student-teacher ratios will result in better student performance. Consequently, in California, for example, all K-3 classes now have a maximum student-teacher ratio of 20. For the 20 school districts in our sample, we seem to have confirmed the existence of a negative relationship between large classes and poor student performance. We even included a regression line in the scatterplot, something that you should have encountered towards the end of your statistics course. However, the graph of the regression line does not allow you to make exact quantitative statements about the relationship. You may want to predict what the effect of reduction by one in the student-teacher ratio would be.

To answer the questions relating to the more precise nature of the relationship between large classes and poor student performance, you need to estimate the regression intercept and slope. A regression line is little else than fitting a line through the observations in the scatterplot according to some principle. You could, for example, draw a line from the test score for the lowest student-teacher ratio to the test score for the highest student-teacher ratio, ignoring all the observations in between. Or you could sort the data by student-teacher ratio and split the sample in half so that the observations with the lowest ten student-teacher ratios are in one set, and the observations with the highest ten student-teacher ratios are in the other set. For each of the two sets you could calculate the average student-teacher ratio and the corresponding average test score, and then connect the two resulting points. Or you could just eyeball the relationship. Some of these principles have better properties than others to infer the true underlying (population) relationship from the given sample. The principle of ordinary least squares (OLS), for example, will give you desirable properties under certain restrictive assumptions that are discussed in Chapter 4 of the Stock/Watson textbook.

Back to computing. If the dependent variable, Y, is only determined by a single explanatory variable *X* in a linear fashion of the type

$$Y_i = \beta_0 + \beta_1 X_i + u_i \qquad i=1,2, ..., N$$

with "*u*" representing the error, or random disturbance, not accounted for by the linear equation, then the task is to find some value for β_0 and β_1 . If you had values for these coefficients, β_1 describes the effect of a unit increase in *X* on *Y*. Often a regression line is a linear approximation to an underlying relationship and the intercept β_0 only has a useful meaning if observations around *X*=0 occur in the data. As we have seen in the scatterplot above, there are no observations around the student-teacher ratio of zero, and it is therefore better not to interpret the numerical value of the intercept at all. Your professor most likely will give you a serious penalty in the exam for interpreting the intercept here because with no students present, there is no score to record. (What would be the function of the teacher in that case?)

There are various ways to estimate the regression line. The command for regressing a variable *Y* on a constant (intercept) and another variable *X* is: ls Y c X, where "ls" stands for least squares. Here, working with the *command* window,[5] type

ls(h) testscr c str

where the "h" in parentheses indicates that you are using heteroskedasticity-robust standard errors ("c" stands for the intercept). The output appears as follows:

Dependent Variable: TESTSCR
Method: Least Squares
Date: xx/xx/xx Time: xx:xx
Sample: 1 20
Included observations: 20
White Heteroskedasticity-Consistent Standard Errors & Covariance

Variable	**Coefficient**	**Std. Error**	**t-Statistic**	**Prob.**
C	**747.0976**	**47.19888**	**15.82872**	**0.0000**
STR	**-4.880141**	**2.284288**	**-2.136395**	**0.0466**

R-squared	**0.188297**	**Mean dependent var**	**648.4700**
Adjusted R-squared	**0.143203**	**S.D. dependent var**	**21.92583**
S.E. of regression	**20.29528**	**Akaike info criterion**	**8.953294**
Sum squared resid	**7414.174**	**Schwarz criterion**	**9.052867**
Log likelihood	**-87.53294**	**F-statistic**	**4.175609**

[5] If you are working in a Group Window, possibly by having invoked the **Show** option, then click on **Procs**. Next press **Make Equation,** and a dialog box will open. If EViews has not suggested a regression of the test score on the student-teacher ratio plus a constant ("*C*"; this letter is reserved in EViews for the constant – actually a vector of ones – and you are not allowed to give another variable this name), then type in the variable names in that order (EViews takes the first variable as the dependent variable; it does not matter if you place the constant before the explanatory variable or after). Alternatively, start in the Main menu and click on **Objects** and the **New Object** and finally **Equation**. The same dialog box will open.

Durbin-Watson stat	**0.484121**	**Prob(F-statistic)**	**0.055929**

According to these results, lowering the student-teacher ratio by one student per class results in an increase of almost five points, on average, in the districtwide test score. Using the notation of your textbook, you should display the results as follows:

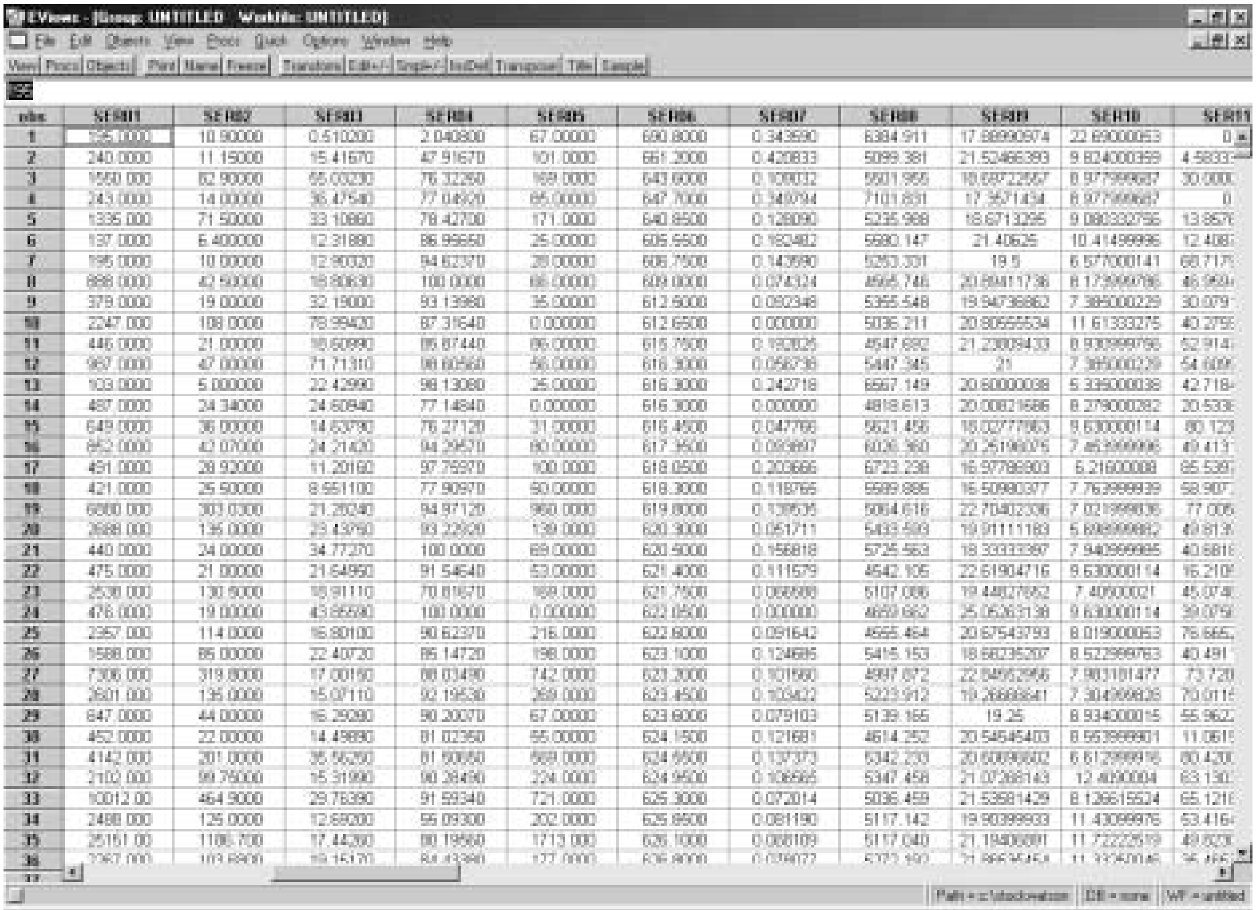
$$\widehat{TestScore} = 747.1 - 4.88 \times \text{STR}$$
$$(47.2) \quad (2.28)$$

Note that the result for the 20 chosen school districts is more than twice as strong as for the sample of all 420 school districts. However, this is a rather small sample and the standard error of the estimator is relatively large.

a) Entering Data from a Spreadsheet

So far you entered data manually. Most often you will work with larger data sets that are *external* to the EViews program, i.e., they will not be included in, or be part of, the program itself. This makes sense as data sets either become very large or are generated by another program, such as a spreadsheet.

Stock and Watson present the California test score data set in Chapter 4 of the textbook. Locate the corresponding Excel file caschool.xls and open it. Next, following the procedures discussed previously, open a new EViews workfile with 420 observations, and use the **Quick/Edit Group (Empty Series)** procedure. Return to the Excel file and mark F2:R421. Next, using the "copy" and "paste" commands common to Windows programs, move the data block to EViews. You presumably are familiar with this procedure. This is what you should see in EViews:

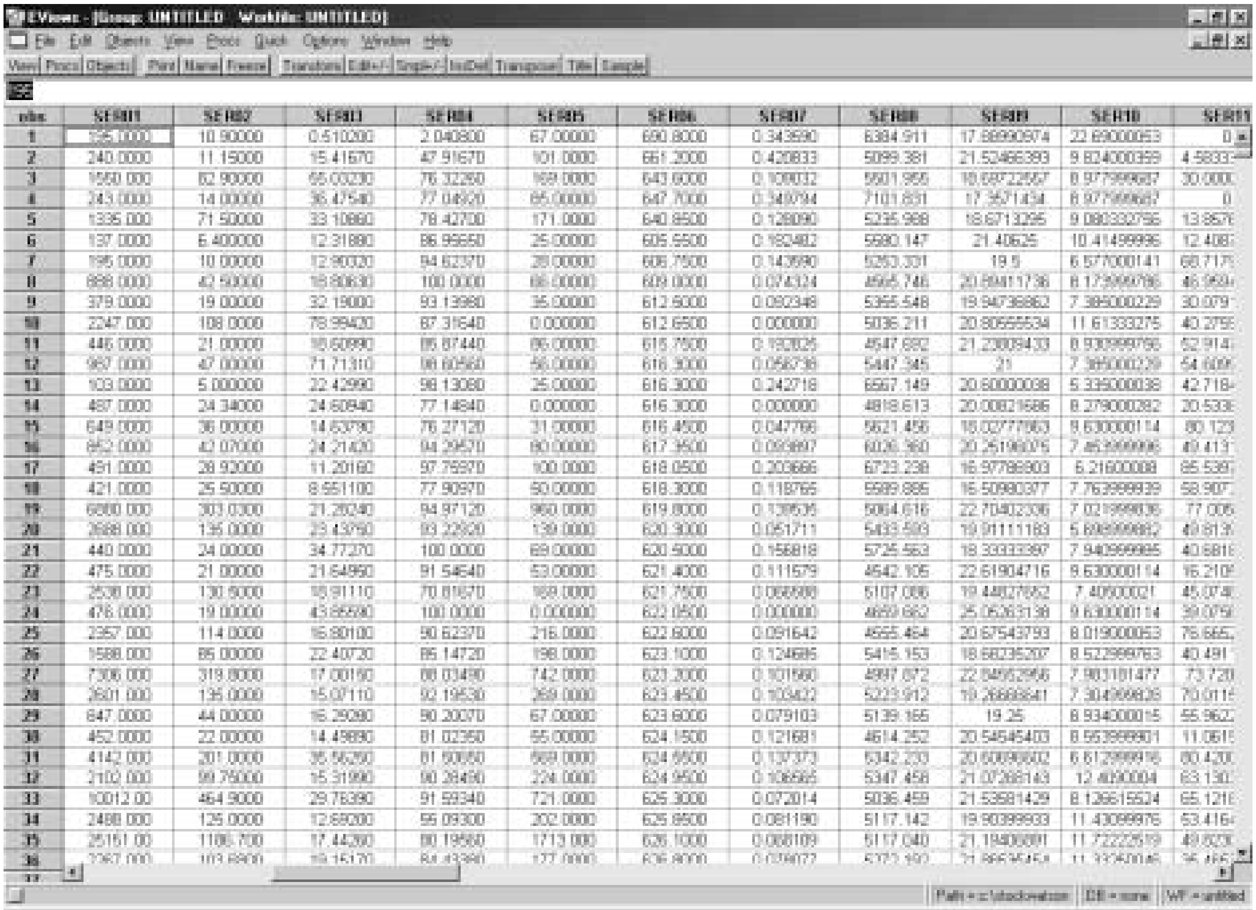


obs	SER01	SER02	SER03	SER04	SER05	SER06	SER07	SER08	SER09	SER10	SER11
1	195.0000	10.90000	0.510200	2.040800	67.00000	690.8000	0.343590	6384.911	17.88990974	22.69000053	0
2	240.0000	11.15000	15.41670	47.91670	101.0000	661.2000	0.420833	5099.381	21.52466393	9.824000359	4.5833
3	1550.000	82.90000	55.03230	76.32260	169.0000	643.6000	0.109032	5501.955	18.69722557	8.977999687	30.000
4	243.0000	14.00000	36.47540	77.04920	85.00000	647.7000	0.349794	7101.831	17.3571434	8.977999687	0
5	1335.000	71.50000	33.10860	78.42700	171.0000	640.8500	0.128090	5235.988	18.6713295	9.080332756	13.857
6	137.0000	6.400000	12.31880	86.95650	25.00000	605.5500	0.182482	5580.147	21.40625	10.41499996	12.408
7	195.0000	10.00000	12.90320	94.62370	28.00000	606.7500	0.143590	5253.331	19.5	6.577000141	68.717
8	888.0000	42.50000	18.80630	100.0000	66.00000	609.0000	0.074324	4565.746	20.89411736	8.173999786	46.959
9	379.0000	19.00000	32.19000	93.13980	35.00000	612.5000	0.092348	5355.548	19.94736862	7.385000229	30.079
10	2247.000	108.0000	78.99420	87.31640	0.000000	612.6500	0.000000	5036.211	20.80555534	11.61333275	40.275
11	446.0000	21.00000	18.60990	85.87440	86.00000	615.7500	0.192825	4547.692	21.23809433	8.930999756	52.914
12	987.0000	47.00000	71.71310	98.60560	56.00000	616.3000	0.056738	5447.345	21	7.385000229	54.609
13	103.0000	5.000000	22.42990	98.13080	25.00000	616.3000	0.242718	6567.149	20.60000038	5.335000038	42.718
14	487.0000	24.34000	24.60940	77.14840	0.000000	616.3000	0.000000	4818.613	20.00821686	8.279000282	20.533
15	649.0000	36.00000	14.63790	76.27120	31.00000	616.4500	0.047766	5621.456	18.02777863	9.630000114	80.123
16	852.0000	42.07000	24.21420	94.29570	80.00000	617.3500	0.093897	6026.360	20.25196075	7.453999996	49.413
17	491.0000	28.92000	11.20160	97.75970	100.0000	618.0500	0.203666	6723.238	16.97786903	6.21600008	85.539
18	421.0000	25.50000	8.651100	77.90970	50.00000	618.3000	0.118765	5589.885	16.50980377	7.763999939	58.907
19	6880.000	303.0300	21.28240	94.97120	960.0000	619.8000	0.139535	5064.616	22.70402306	7.021999836	77.006
20	2688.000	135.0000	23.43750	93.22920	139.0000	620.3000	0.051711	5433.593	19.91111183	5.698999882	49.813
21	440.0000	24.00000	34.77270	100.0000	69.00000	620.5000	0.156818	5725.563	18.33333397	7.940999985	40.681
22	475.0000	21.00000	21.64950	91.54640	53.00000	621.4000	0.111579	4542.105	22.61904716	9.630000114	16.210
23	2538.000	130.5000	18.91110	70.81670	169.0000	621.7500	0.066588	5107.086	19.44827652	7.40500021	45.074
24	476.0000	19.00000	43.65590	100.0000	0.000000	622.0500	0.000000	4669.662	25.05263138	9.630000114	39.075
25	2357.000	114.0000	16.80100	90.62370	216.0000	622.6000	0.091642	4555.464	20.67543793	8.019000053	76.665
26	1588.000	85.00000	22.40720	86.14720	198.0000	623.1000	0.124686	5415.153	18.68235207	8.522999763	40.491
27	7306.000	319.8000	17.00190	88.03490	742.0000	623.2000	0.101560	4997.872	22.84552956	7.983181477	73.720
28	2601.000	135.0000	15.07110	92.19530	269.0000	623.4500	0.103422	5223.912	19.26666641	7.304999828	70.011
29	847.0000	44.00000	16.29280	90.20070	67.00000	623.6000	0.079103	5139.165	19.25	8.934000015	55.962
30	452.0000	22.00000	14.49890	81.02350	55.00000	624.1500	0.121681	4614.252	20.54545403	8.553999901	11.061
31	4142.000	201.0000	35.56250	81.50050	569.0000	624.5500	0.137373	5342.233	20.60696602	6.612999916	80.420
32	2102.000	99.75000	15.31990	90.28490	224.0000	624.9500	0.106565	5347.458	21.07268143	12.4090004	63.130
33	10012.00	464.9000	29.76390	91.59340	721.0000	625.3000	0.072014	5036.459	21.53581429	8.126615524	65.121
34	2488.000	125.0000	12.69200	56.09300	202.0000	625.8500	0.081190	5117.142	19.90399933	11.43099976	53.416
35	25151.00	1186.700	17.44260	80.19560	1713.000	626.1000	0.068109	5117.040	21.19406891	11.72222519	49.828
36	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

Next you need to rename Ser01 to Ser13 with the names as they appeared in the original Excel worksheet. The names were as follows:

enrl_tot teachers calw_pct meal_pct computer testscr comp_stu expn_stu str avginc el_pct read_scr math_scr

In the EViews program, click on *Ser01*, type *enrl_tot*, and press enter. You will be asked if you want to replace SER01 with ENRL_TOT. Say yes and continue. EViews is ready to receive the second variable name (*teachers*), and so on. Write in all the variable names. When you are done, you are ready to save the workfile. Name it caschool.wf1.



You can now reproduce Equation (4.7) from the textbook. Use the regression command you previously learned to generate the following output ("freeze" the output and use the numbers button to adjust the number of digits after the decimal point).

Dependent Variable: TESTSCR
Method: Least Squares
Sample: 1 420
Included observations: 420
White Heteroskedasticity-Consistent Standard Errors & Covariance

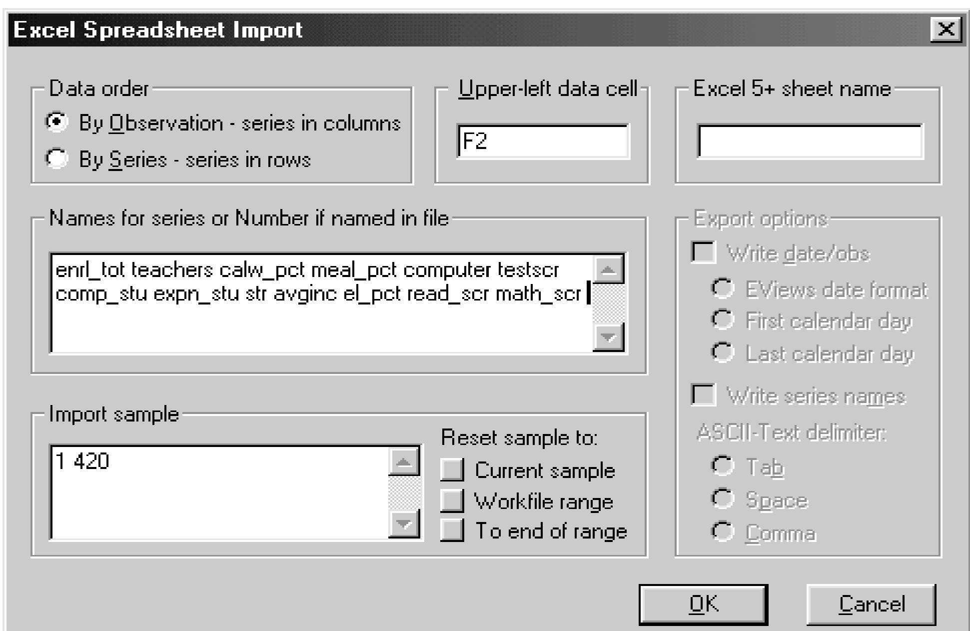
Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	698.9	10.36	67.44	0.00
STR	-2.28	0.52	-4.39	0.00

R-squared	0.05	Mean dependent var	654.16
Adjusted R-squared	0.05	S.D. dependent var	19.05
S.E. of regression	18.58	Akaike info criterion	8.69
Sum squared resid	144315.48	Schwarz criterion	8.71
Log likelihood	-1822.25	F-statistic	22.58
Durbin-Watson stat	0.13	Prob(F-statistic)	0.00

(You can find the standard errors and the *t*-statistic on p. 114 of the Stock/Watson textbook. The regression R^2 , sum of squared residuals (*SSR*), and standard error of the regression (*SER*) are presented in Section 4.8.)

*Importing Data Files directly into EView*s

Even though the cut and paste method seemed straightforward enough, there is a second, more direct way to import data into EViews from Excel, which does not involve copying and pasting data points. Start again with a new workfile in EViews. Next press **Procs /Import /Read Text-Lotus-Excel**. A dialog box will open, and you will first have to specify the location where your data file (caschool.xls) resides. After you double click on the file, another dialog box opens. The data in your Excel file are ordered by observation, so you do not have to change the automatic setting at the top. You do have to provide names for the series though, so copy the names of the variables from cell F1 (*enrl_tot*) to R1 (*math_scr*) in your Excel file. These have to be pasted into the empty field. Finally, EViews has suggested that the first data is in cell B2. Change this to F2, the first data point corresponding to *enrl_tot*. Before you click OK, you need to close the Excel file. The following window is what the dialog box should look like before hitting the return button. Note that EViews also allows you to import other types of data files, e.g. STATA files, although this may be a bit more complicated.



EViews will show that the data exist in the Workfile Window. You may want to check that the data were properly retrieved by typing **Show testscr str** or running the same regression as above.

Finally, you can also save data in ASCII or spreadsheet format by clicking on **Procs** and then **Export Data**.

Multiple Regression Model

Economic theory most often suggests that the behavior of a certain variable is influenced not only by another single variable, but by a multitude of factors. The demand for a product depends not only on the price of the product but also on the price of other goods, income, taste, etc. Similarly, the Phillips curve suggests that inflation depends not only on the unemployment rate, but also on inflationary expectations, productivity growth, the change in price mark-ups, etc.

An extension of the simple regression model is the multiple regression model, which incorporates more than one regressor (see Equation (5.7) in the textbook).

$$Y_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + ... + \beta_k X_{ki} + u_i\,,\ i = 1,\ldots,n.$$

To estimate the coefficients of the multiple regression model, you proceed in a similar way as in the simple regression model. The difference is that you now need to list the additional explanatory variables. In general, the command is: ls Y c X1 X2 … Xk. See if you can reproduce the following regression output, which corresponds to Column 5 in Table 5.2 of the textbook.

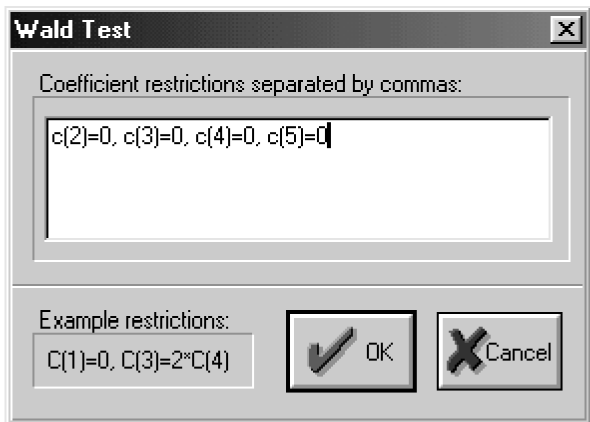
Dependent Variable: TESTSCR
Method: Least Squares
Date: xx/xx/xx Time: xx:xx
Sample: 1 420
Included observations: 420
White Heteroskedasticity-Consistent Standard Errors & Covariance

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	700.4	5.5	126.48	0.000
STR	-1.01	0.27	-3.77	0.000
EL_PCT	-0.130	0.036	-3.58	0.000
MEAL_PCT	-0.529	0.038	-13.87	0.000
CALW_PCT	-0.048	0.059	-0.82	0.415

R-squared	0.775	Mean dependent var	654.157
Adjusted R-squared	0.773	S.D. dependent var	19.053
S.E. of regression	9.084	Akaike info criterion	7.263
Sum squared resid	34247.463	Schwarz criterion	7.311
Log likelihood	-1520.188	F-statistic	357.054
Durbin-Watson stat	1.430	Prob(F-statistic)	0.000

The interpretation of the coefficients is equivalent to that of a controlled science experiment: it indicates the effect of a unit change in the relevant variable on the dependent variable, *holding all other factors constant* (“ceteris paribus”).

Section 5.8 of your textbook discusses the *F*-statistic for testing restrictions involving multiple coefficients. To test whether all of the above coefficients are zero with the exception of the intercept, click on **View/Coefficient Tests/Wald-Coefficient Restrictions**. The regression coefficients are stored in a vector $c(1)$ to $c(k+1)$, where the number in parentheses indicates the order of appearance in the regression output. Thus in the example c(1) is the intercept or constant term, c(1) is the coefficient on STR, and so forth. To execute the above test, enter the following and press enter:



The computer will generate the following output:

Wald Test:
Equation: Untitled

Test Statistic	Value	df
F-statistic	361.6835	(4, 415)
Chi-square	1446.734	4

Null Hypothesis Summary:

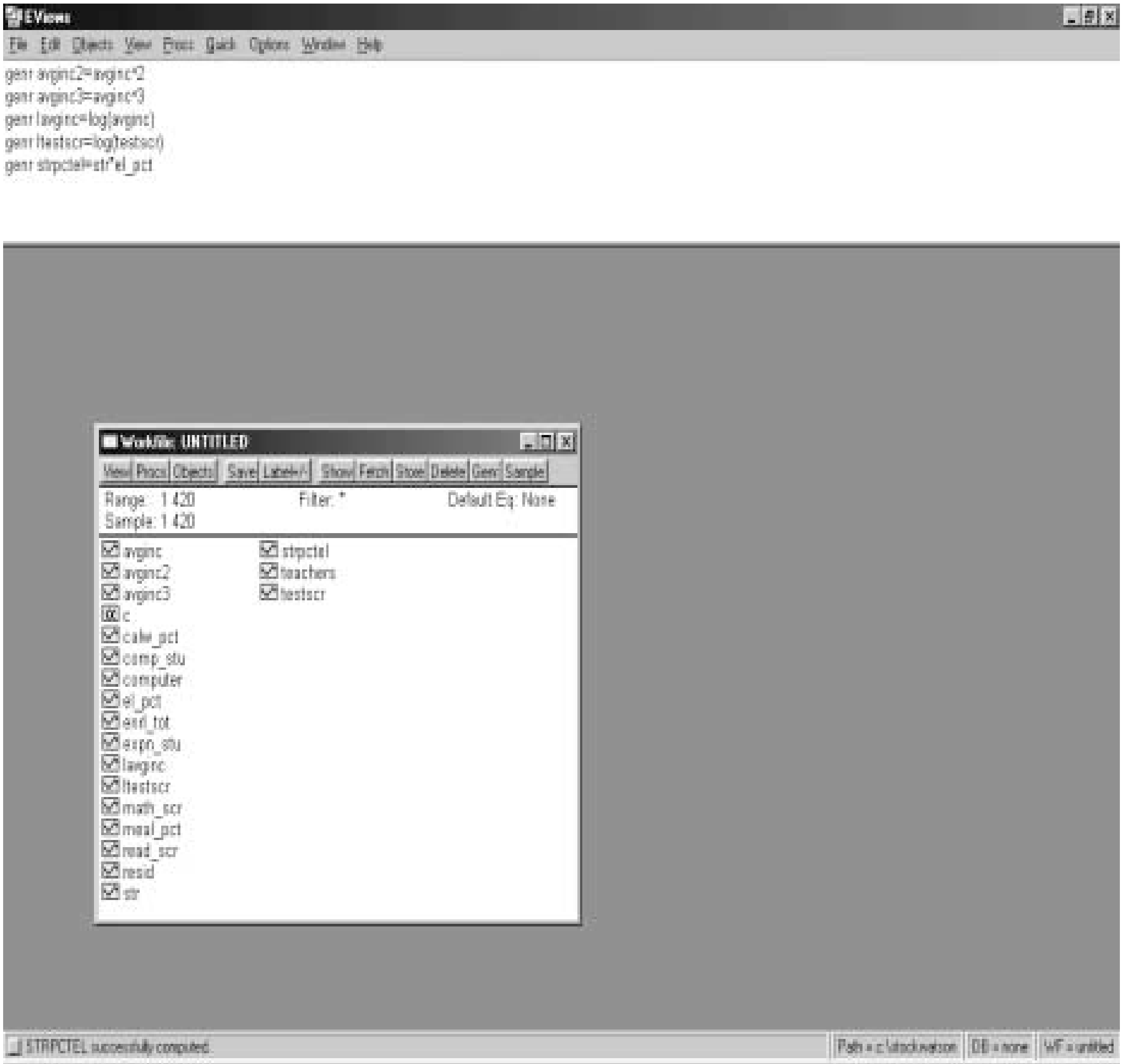
Normalized Restriction (= 0)	Value
C(2)	-1.014353
C(3)	-0.129822
C(4)	-0.528619
C(5)	-0.047854

Restrictions are linear in coefficients.

See if you can generate the *F*-statistic of 5.43 following Equation (5.40) in the text.

Data Transformations

You will be frequently required to transform some of the raw data that you received before you run a regression. In EViews you transform variables by using the “genr” (as in generate) command. For example, Chapter 6 of the Stock/Watson textbook introduces the polynomial regression model, logarithms, and interactions between variables. Let us reproduce Equations (6.11), (6.23), and (6.37) here. The following commands generate the necessary variables:



Next run the three regressions using the same technique as for multiple regression analysis. Finally save your workfile again and exit the workfile.

Batch Files

So far, you have either clicked on buttons in EViews or used the “Command Window” to type executable statements. But what if you wanted to keep a permanent record of all the transformations you made, regressions you tried, graphs you created, etc.? In that case, you would need to create a "program" that consists of line commands similar to those that you used in the "Command Window" previously. After having created such a program, you can then execute ("run") it and view the output afterwards (if you did not make any errors).

To create a program, click on **File** and then **New** and **Program**. This opens the "Program" box. Type in, or cut and paste, the following commands exactly as they appear below. Use ' whenever needed at the beginning of the line to indicate that you have added a comment. These flags are useful if you want to remember later what you were doing or if you want others to understand your program. Then "save [it] as" "Tutorialch4.prg" in your directory and click the "Run" button. Make sure that the workfile "caschool.wf1" resides in the same subdirectory. EViews will execute your commands. (To save time, you may want to omit some of the variable definitions, which appear in commented lines.)

```
' Stock and Watson
'
' chapter 4 (EViews 4.1 Version)
'
' Chapter 4: Linear Regression with One Regressor
'
open ch4_7.wf1
'
' DIST_CODE    DISTRICT CODE
' READ_SCR     AVG READING SCORE
' MATH_SCR         AVG MATH SCORE
' COUNTY       COUTY
' DISTRICT     DISTRICT
' GR_SPAN      GRADE SPAN OF DISTRICT
' ENRL_TOT     TOTAL ENROLLMENT
' TEACHERS     NUMBER OF TEACHERS
' COMPUTER     NUMBER OF COMPUTERS
' TESTSCR      AVG TEST SCORE (=(READ_SCR+MATH_SCR)/2)
' COMP_STU     COMPUTERS PER STUDENT (=COMPUTER/ENRL_TOT)
' EXPN_STU     EXPENDITURES PER STUDENT ($S)
' STR          STUDENT TEACHER RATIO (TEACHERS/ENRL_TOT)
' EL_PCT       PERCENT OF ENGLISH LEARNERS
' MEAL_PCT     PERCENT QUALIFYING FOR REDUCED-PRICE LUNCH
' CLW_PCT      PERCENT QUALIFYING FOR CALWORKS
' AVING        DISTRICT AVERAGE INCOME (IN $1,000S)

'
' Summary statistics for str and testscr in Table 4.1
' Define a group and give the table a name (tab4_1).
'
group tab4_1 str testscr
tab4_1.stats
'
' Correlation between str and testscr
```

```
' Again, define a group first and then give it a name (cor_str_testscr)
'
group cor_str_testscr str testscr
cor_str_testscr.cor
'
' Figure 4.2: Scatterplot of Test Score and Student-Teacher Ratio
'
group Fig4_2 str testscr
Fig4_2.scat
'
' Equation 4.7 and 4.26
'
' Here is an example how to use OLS in the program.
' You first define the equation and then use ls command.
'
equation eq4_7.ls(h) testscr c str
'
' crossplot Figure 4.3 with regression line
'
group Fig4_3 str testscr
Fig4_3.linefit
'
' Equation 4.33
'
' Below a binary variable is defined first by setting it to zero for the entire sample
' then to set it to one for observations where the student-teacher ratio is less than 20.
'
genr dsize=0
smpl if str<20
genr dsize=1
smpl 1 420
equation eq4_33.ls(h) testscr c dsize
'
' End of Chapter 4
' ------------------------------------------------------------------------------
```

In the "caschool.wf1" workfile, you can now click on the equations or tables you have generated. "Eq4_11," for example, has reproduced Equation (4.11) from your textbook. This is identical to the regression generated above.

A summary of frequently used EViews commands is given at the end of the tutorial.

3. TIME SERIES DATA

Let's leave the cross-sectional data and move on to Chapter12 in the textbook. This is where EViews is superior to other regression packages, such as SAS, SPSS, or STATA, which are more suitable for cross-sectional analysis.

In the time series chapters of the textbook, you will use past values or lags of variables to forecast the dependent variable or for data transformations. We refer to "(*t*-1)" as a *lag* (similarly, "(*t*+1)" is a *lead*). Imagine you had entered the data for the CPI, but you wanted to forecast the inflation rate or the change in the inflation rate. The next step is therefore to transform the raw data. Specifically,

$$Inf_t = \frac{CPI_t - CPI_{t-1}}{CPI_{t-1}} * 100 = \left(\frac{CPI_t}{CPI_{t-1}} - 1 \right) * 100$$

To create past values of variables, you generate a lag by adding a "(-1)" after the variable name in the "*genr*" statement. In a spreadsheet, this amounts to copying an entire data series and pasting it into a new column one observation down: the first observation becomes the second observation, etc. The procedure generalizes to higher lags: X_{t-12} is X(-12).[6]

Type or copy and paste the following lines of code into a new program and run it. See if you understand the code in terms of generating the inflation rate and its change. Because the short-run Phillips curve suggests a negative relationship between the unemployment rate and the *future* change in the inflation rate, we also need to calculate the change in the inflation rate, or $\Delta Inf_t = Inf_t - Inf_{t-1}$. In the program below, this variable is called "*dinf*". You need the workfile "ch12_plus_14.wf1," which you can download from the Web site.

After running the program, click on some of the figures and equations to view the output.

```
' Stock and Watson
' chapter 12_plus_14
'
' Chapter 12
'
open ch12_plus_14.wf1
'
' LHUR        Unemployment Rate U.S.
' PUNEW       Consumer Price Index U.S.
' FYFF        Federal Funds Interest Rate U.S.
' FYGM3       3-Month Treasury Bill Interest Rate U.S.
' FYGT1       1-Year Treasury Bond Interest Rate
' EXRUK       Dollar-Pound Exchange Rate
' GDP_JP      Real GDP Japan
'
'
smpl 1959:1 2005:4
```

[6] In mathematics, a lag is defined (loosely) through the use of a "lag-operator" L, where $L^i x_t = x_{t-i}$. Similarly, the "difference operator" $\Delta = (1 - L)$, so that $\Delta x_t = x_t - x_{t-1}$. See Appendix 12.3 of the textbook for more details.

```
'
' generate the annualized inflation rate and its change
'
genr lpunew=log(punew)
genr inf=400*(lpunew-lpunew(-1))
genr dinf=inf-inf(-1)
genr yeardinf=inf-inf(-4)
'
' Figure 12.1
'
smpl 1960:1 1999:4
group Fig12_1 inf lhur
Fig12_1.line(m)
freeze(InflatUR) Fig12_1
'
'
' Equation 12.13
'
smpl 1962:1 1999:4
'
' EViews allows you to use (-i) in an equation on variables
' that you did not have to generate previously
'
equation eq12_13.ls(h) dinf c dinf(-1) dinf(-2) dinf(-3) dinf(-4)
'
' Figure 12.13
'
group Fig12_3 lhur(-4) yeardinf
Fig12_3.scat(r)
freeze(Phillips) Fig12_3
'
' Equation 12.17
'
equation eq12_17.ls(h) dinf c dinf(-1) dinf(-2) dinf(-3) dinf(-4) lhur(-1) lhur(-2) lhur(-3) lhur(-4)
```

4. SUMMARY OF FREQUENTLY USED EVIEWS COMMANDS

The command 'genr' *creates new variables and modifies existing variables.*

Examples:

```
genr expn=expn_stu/1000
```

generates the expenditure variable used in the textbook by dividing the original data by 1,000.

```
genr avginc2=avginc^2
genr lavginc=log(avginc)
```

create the square and log of average income, respectively.

Note that commands of the type

```
genr testscr = testscr/100
```

simply modify an existing variable.

The most frequently used operators are + (addition), - (subtraction), * (multiplication), / (division), ^ (exponentiation). Log(x) calculates the natural logarithm of x (see the above example) and exp(x) computes the exponent of x.

When working with time series data, lags are frequently used. EViews allows you to create these simply by entering (-i) immediately after the variable name:

```
genr dinf=inf-inf(-1)
genr yeardinf=inf-inf(-4)
```

The first command generates the quarterly change in the inflation rate (assuming that you work with quarterly data), while the second generates the annual change in the inflation rate.

The sample range is set through the ‘smpl’ command. The command is of the type: smpl n1 n2, where n1 and n2 are the beginning and end dates (first and last observations) for which EViews will execute the commands that follow.

Examples are

```
smpl 1 420
smpl 1959:1 2001:4
```

In the first case, EViews is instructed to use all 420 observations of the California Test Score Data Set used in Chapters 4-7. The second example restricts the sample to the first quarter of 1959 to the last quarter of 2001.

Note that you can work with a subsample by using relational operators.

```
smpl if str>=20
```

only looks at observations with a student-teacher ratio of less than 20.

The most frequently used *statistical operations* involve running regressions (‘ls’), establishing the correlation between variables (‘cor’), and graphing variables (‘line’). EViews creates results by storing them in so-called objects. Initially, you will use the ‘equation’ object and the ‘group’ object most often, as in the following examples:

```
equation eq4_7.ls(h) testscr c str
```

```
equation eqtab5_2_5.ls(h) testscr c str el_pct meal_pct calw_pct
equation eq12_7.ls(h) dinf c dinf(-1)
```

In each case, an equation object is declared first and a name is assigned to it. 'ls' then instructs EViews to use OLS estimation for the equation. The dependent variable appears first, followed by the regressors, where 'c' is used for the intercept ('c' is a reserved name in EViews, meaning that you cannot use it to generate a variable called 'c').

To create a line graph or to view the correlation between variables, you first must assign the variables to a group and name this group. Next, you execute the correlation and graphing through the 'cor' and 'line' command. Examples:

```
group cor_str_testscr str testscr
cor_str_testscr.cor
```

Here the variables str (student-teacher ratio) and testscr (testscore) are assigned to a group called cor_str_testscr (the name was chosen to indicate what the group was used for, but we could have named it almost anything alternatively), and EViews is then instructed to calculate the correlation between the variables in the group. The group can contain more than two variables.

In the following example, inf (inflation) and lhur (unemployment rate) are assigned to a group called Fig12_1 and are then plotted (where 'm' is an option that allows for the display of multiple graphs).

```
group Fig12_1 inf lhur
Fig12_1.line(m)
```

5. FINAL NOTE

For a complete list of commands, consult the EViews *Command and Programming Reference* or the *User's Guide*. Alternatively, use the "Help" command inside EViews. As mentioned before, this tutorial is not intended to replace them.

EViews replication batch files for all the results in the Stock/Watson textbook are available from the Web site. You are invited to download these and study them.