

Undergraduate Econometrics using GRETL

Lee Adkins

January 4, 2006

Preface

Contents

1	Introduction	1
1.1	What is Gretl ?	1
1.2	Installing Gretl	2
2	Gretl Basics	3
3	Introduction to Econometrics	5
4	Some Basic Probability Concepts	9
5	Simple Linear Regression	16
5.1	Retrieve the Data	16
5.2	Graph the Data	18
5.3	Estimate the Food Expenditure relationship	18
6	Sampling Properties of Least Squares Estimator	22
7	Inference in the Simple Linear Regression Model	27
7.1	Confidence Intervals	27
7.2	Hypothesis Tests	29

<i>CONTENTS</i>	iii
8 Using R with Gretl	31
9 Reporting Results and Functional Form	36
9.1 Coefficient of Determination	36
9.2 Reporting Results	36
9.3 Functional Forms	39
9.4 Testing for Normality	40

Chapter 1

Introduction

1.1 What is Gretl?

Gretl, which is an acronym for Gnu Regression, Econometrics and Time-series Library, is an easy to use, reasonably powerful software package for doing econometrics. It is available for download at no charge from <http://gretl.sourceforge.net>. Unlike software sold by commercial vendors (SAS, Eviews, Shazam to name a few) you may redistribute and/or modify **gretl** under the terms of the GNU General Public License (GPL) as published by the Free Software Foundation.

Gretl comes with many sample data files and a database of US macroeconomic time series. From the **gretl** web site, you have access to more sample data sets from many of the leading textbooks in econometrics, including ours *Undergraduate Econometrics* by [Hill et al. \(2001\)](#). It can be used to compute the least-squares, weighted least squares, nonlinear least squares, instrumental variables least squares, logit, probit, tobit and a number of time series estimators. It calls another GNU program called *gnuplot* to generate graphs and is capable of generating output in LaTeX format. As of this writing **gretl** is under development so you can probably expect some bugs.

The driving force behind **gretl** is Allin Cottrell of Wake Forest University. He is currently very active in fixing any bugs one may find in **gretl**. Hence, if you encounter what you think is a bug you can either modify the C source code to fix it yourself or you can contact Professor Cottrell. I know which option I like!

1.2 Installing Gretl

To install **gretl** on your system, you will need to download the appropriate executable file for the computer platform you are using. For Microsoft Windows users the appropriate site is <http://gretl.sourceforge.net/win32/>. One of the nice things about **gretl** is that Macintosh and Linux versions are also available out of the box. If you are using some other exotic computer system, you can obtain the source code and compile it whatever form you'd like. No guarantees that this will work, but this is not something available with any commercial software I can think of.

Gretl depends on some other (free) programs to perform some of its magic. If you install **gretl** on your Mac or Windows based machine using the appropriate executable file provided on **gretl**'s download page then everything you need to make **gretl** work should be installed as part of the package. If, on the other hand, you are going to build your own **gretl** using the source files, you may need to install some of the supporting packages yourself. I assume that if you are savvy enough to compile your own version of **gretl** then you probably know what to do. For most, just install the self-extracting executable, **gretl_install.exe**, available at the download site. **Gretl** comes with an Adobe pdf manual that will guide you through installation and introduce you to the interface. I suggest that you start with it, paying particular attention to chapters 1 and 2 which discuss installation in more detail and some basics on how to use the interface.

Since this manual is based on the examples from *Undergraduate Econometrics* by Hill et al. (2001) then you should also download and install the accompanying data files that go with this book. The file is available at

<http://spears.okstate.edu/~ladkins/class/4213/gretl/UEsetup.exe>.

This is a self-extracting windows file that will install the UE data sets onto the `c:\userdata\gretl\data` directory of your harddrive. If you have installed **gretl** in any other place besides `c:\userdata\gretl` then you are given the opportunity to specify a new location in which to install the program during setup.

Chapter 2

Gretl Basics

There are several different ways to work in **gretl**. The one most use takes advantage of its built in graphical user interface (GUI). Those of you who grew up using MS Windows or the Macintosh will find this way of working quite easy. Basically, you are able to point the mouse at what you want to accomplish, fill in the desired options from the menus, and click OK. **Gretl** is using your user input, delivered by mouse clicks and a few keystrokes to generate computer code that is executed in the background.

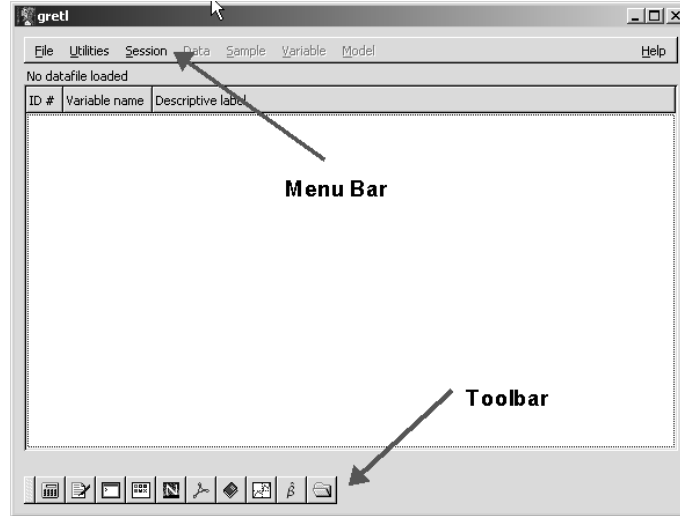
Gretl offers a command line interface as well and those of you who use Linux or are old DOS warriors may want to use it this way. The command line version is launched by executing `gretlcli` in a console window. If you don't know what a console window is, then you can file this piece of information away and stick with the GUI.

One of the great things about **gretl** is that it accumulates this code into a script file that can be run in its entirety at another time. So, if you have completed an analysis that involves many sequential steps, the script can be open and run in one step to get you to the desired result. You can also use the script environment to conduct Monte Carlo studies in econometrics. Monte Carlo studies use computer simulation (sometimes referred to as experiments) to study the properties of a particular technique. This is especially useful when the mathematical properties of your technique are particularly difficult to ascertain. In the exercises below, you will learn a little about doing these kinds of experiments in econometrics.

In figure 2.1 below is the main window in **gretl**.

Across the top of the window you find the Menu Bar. From here you import

Figure 2.1: The main window for gretl's GUI



and manipulate data, analyze data, and manage output. At the bottom of the window is the gretl toolbar. This contains a number of useful utilities that can be launched from within gretl. Among other things, you can get to the gretl web site from here, open the pdf version of the manual, or open the MS Windows calculator (very handy!). More will be said about these functions later.

Chapter 3

Introduction to Econometrics

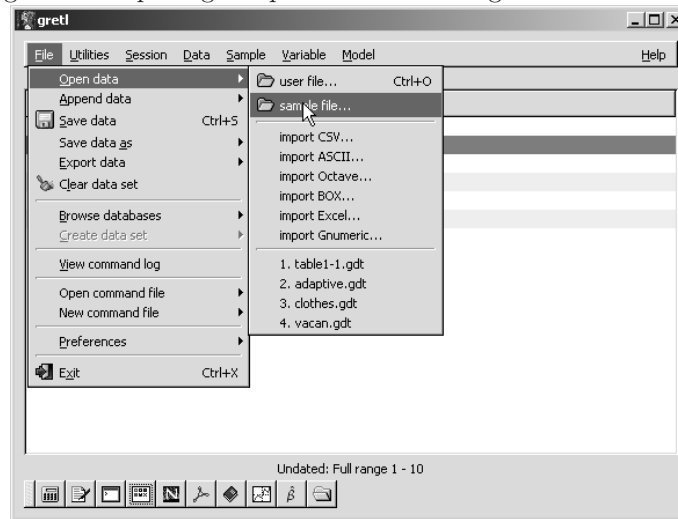
Obtaining data in econometrics and getting it into a format that can be used by your software can be challenging. There are dozens of different pieces of software and many use proprietary data formats that make transferring data between applications difficult. You'll notice that the authors of your book have provided data in several formats for your convenience. In this chapter, we will explore some of the data handling features of **gretl** and show you how to 1) access the data sets that accompany your textbook 2) how to bring one of those data sets into **gretl** 3) how to list the variables in the data set 4) how to modify and save your data. **Gretl** offers great functionality in this regard. Through **gretl** you have access to a very large number of high quality data sets from other textbooks as well as from sources in industry and government. Furthermore, once opened in **gretl** these data sets can be exported to a number of other software formats.

In the beginning, I will illustrate the examples using a number of figures (an excessive number to be sure). As you become familiar with **gretl** the frequency of these figures will diminish and I will direct you to the proper commands using words only. More complex series of commands may require you to use the **gretl** script facilities which basically allow you to write simple programs in their entirety, store them in a script file, and then execute all of the commands in a single batch. The convention used will be to refer to menu items as **A>B>C** which indicates that you are to click on option A on the menu bar, then select B from the pulldown menu and further select option C from B's pulldown menu. All of this is fairly standard practice, but if you don't know what this means, ask your instructor now.

First, take a look at Table 1.1 in your textbook. It contains monthly sales data for Honda Accords. In this exercise, you will learn to import data from **gretl** and be able to reproduce this table.

Open the main **gretl** window and click on **File>Open data>sample file**. The result appears in figure 3.1.

Figure 3.1: Opening sample data files from **gretl**'s main window



This will open another window that contains tabs for each of the data compilations that you have installed in the **gretl/data** directory of your program. If you installed the data sets that accompany this book using the self extracting windows program then a tab will appear like the one shown in figure 3.2.

Scroll down to find the data set called ‘table1-1’ and open it using the ‘open’ button at the bottom of the window. This will bring the variables that make up Table 1.1 into **gretl**. At this point use the **Data** tab and select **Display values** as shown in figure 3.3.

From the this pulldown menu a lot can be accomplished. You can edit, sort, graph, and add to your data. You can also perform simple tests, obtain summary statistics like the sample mean and standard deviation, and obtain correlations.

Notice in figure 3.1 that **gretl** gives you the opportunity to import data from several other formats, including ASCII, CSV, EXCEL and others. Also, from the **Data** pulldown menu you can append observations onto the end of a data set and export a data set to another format. If you click on **Browse databases>on database server** you will be taken to a web site (provided your computer is

Figure 3.2: This is Gretl's data files window. Notice that in addition to UE2, data sets from [Ramanathan \(2002\)](#), [Davidson and MacKinnon \(2004\)](#), [Greene \(2003\)](#), [Stock and Watson \(2003\)](#), and [Wooldridge \(2003\)](#) are also installed on my system.

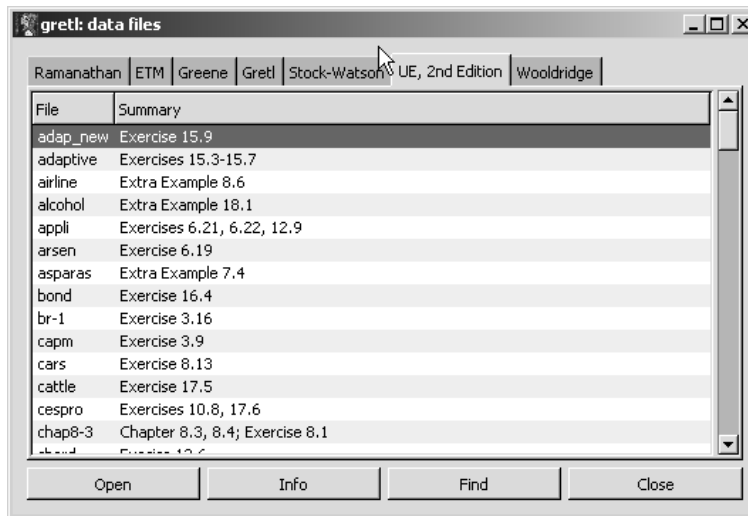
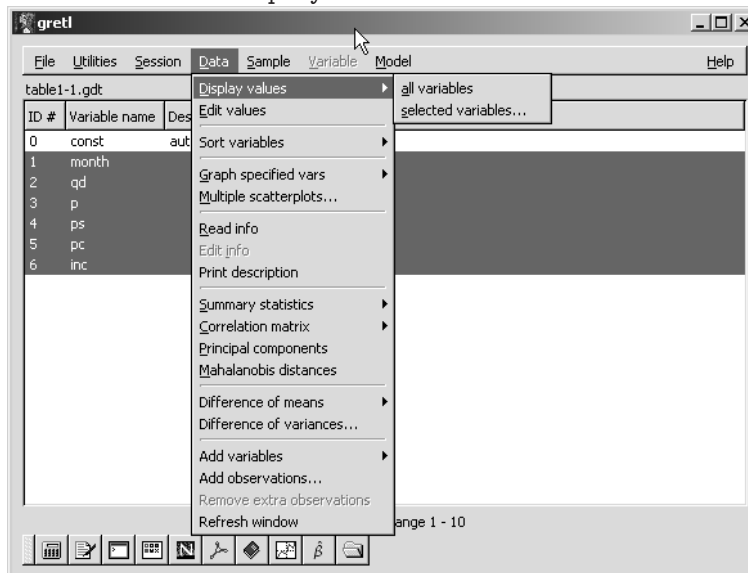


Figure 3.3: Use the Data>Display values>all variables to list the data set.



connected to the internet) that contains a very large number of high quality data sets. You can pull any of these data sets into **gretl** in the same manner as that described above for the UE, 2nd edition data sets. If you are required to write a term paper in one of your classes, these data sets may provide you with all the data that you need.

Chapter 4

Some Basic Probability Concepts

In this chapter, you learned some basic concepts about probability. Since the actual values that economic variables take on are not actually known before they are observed, we say that they are *random*. Probability is the theory that helps us to express uncertainty about the possible values of these variables. Each time we observe the outcome of a random variable we obtain an observation. Once observed, its value is known and hence it is no longer random. So, there is a distinction to be made between variables whose values are not yet observed (random variables) and those whose values have been observed (observations). Keep in mind, though, an observation is merely one of many possible values that the variables can take. Another draw will usually result in a different value being observed.

A probability distribution is just a mathematical statement about the possible values that our random variable can take on. The probability distribution tells us the relative frequency (or probability) with which each possible value is observed. In their mathematical form probability distributions can be rather complicated; either because there are too many possible values to describe succinctly, or because the formula that describes them is complex. In any event, it is common to summarize this complexity by concentrating on some simple numerical characteristics that they possess. The numerical characteristics of these mathematical functions are often referred to as *parameters*. Examples are the mean and variance of a probability distribution. The mean of a probability distribution describes the average value of the random variable over *all* of its possible realizations. Conceptually, there are an infinite number of realizations therefore parameters are not known to us. As econometricians, our goal is to

try to estimate these parameters using a finite amount of information available to us. We collect a number of realizations (called a sample) and then estimate the unknown parameters using a *statistic*. Just as a parameter is an unknown numerical characteristic of a probability distribution, a statistic is an observable numerical characteristic of a sample. Since the value of the statistic will be different for each sample drawn, it too is a random variable. The statistic is used to gain information about the parameter.

In chapter 2 of UE, you used the concept of expected values to obtain certain information about probability distributions. For instance, if X is a random variable that can take on the values 0,1,2,3 and these values occur with probability $1/6$, $1/3$, $1/3$, and $1/6$, respectively. The mean of the probability distribution, designated μ , is obtained *analytically* using its expected value.

$$\mu = E[X] = \sum x f(x) = 0 \cdot \frac{1}{6} + 1 \cdot \frac{1}{3} + 2 \cdot \frac{1}{3} + 3 \cdot \frac{1}{6} = \frac{3}{2} \quad (4.1)$$

So, μ is a parameter. Its value can be obtained mathematically if we know the probability density function of the random variable, X . If this probability distribution is known, then there is no reason to take samples or to study statistics! We can ascertain the mean, or average value, of a random variable without every firing up our calculator. Of course, in the real world we only know that the value of X is not known before drawing it and we don't know what the actual probabilities are that make up the density function, $f(x)$. In order to figure out what the value of μ is, we have to resort to different methods. In this case, we try to infer what it is by drawing a sample and estimating it using a statistic.

One of the ways we bridge the mathematical world of probability theory with the observable world of statistics is through the concept of a *population*. A statistical population is the collection of individuals that you are interested in studying. Since it is normally too expensive to collect information on everyone of interest, the econometrician collects information on a *subset* of this population—in other words, he takes a *sample*.

The population in statistics has an analogue in probability theory. In probability theory one must specify the set of all possible values that the random variable can be. In the example above, a random variable is said to take on 0,1,2, or 3. This set must be complete in the sense that the variable cannot take on any other value. In statistics, the population plays a similar role. It consists of the set that is relevant to the purpose of your inquiry and that is possible to observe. Thus it is common to refer to parameters as describing characteristics of populations. Statistics are the analogues to these and describe characteristics of the sample.

This roundabout discussion leads me to an important point. We often use the

words mean, variance, covariance, correlation rather casually in econometrics, but their meanings are quite different depending on whether we are referring to a probability distribution or a sample. When referring to the analytic concepts of mean, variance, covariance, and correlation we are specifically talking about characteristics of a probability distribution; these can only be ascertained through complete knowledge of the probability distribution functions. It is common to refer to them in this sense as population mean, population variance, and so on. These concepts do not have anything to do with samples or observations!

In statistics we attempt to estimate these (population) parameters using samples and explicit formulae. For instance, we might use the average value of a sample to estimate the average value of the population (or probability distribution).

	Probability Distribution	Sample
mean	$E[X] = \mu$	$\frac{1}{n} \sum x_i = \bar{x}$
variance	$E[X - \mu]^2 = \sigma^2$	$\frac{1}{n-1} \sum (x_i - \bar{x})^2 = s_x^2$

When you are asked to obtain the mean or variance of random variables, make sure you know whether the person asking wants the characteristics of the probability distribution or of the sample. The former requires knowledge of the probability distribution and the latter requires a sample.

In **gretl** you are given the facility to obtain sample means, variances, covariances and correlations. You are also given the ability to compute tail probabilities using the normal, t-, F and chi-square distributions. First we'll examine how to get summary statistics.

Summary statistics usually refers to some basic measures of the numerical characteristics of your sample. In **gretl**, summary statistics can be obtained in at least two different ways. Once your data are loaded into the program, you can select **Data>Summary statistics** from the pull down menu. Which leads to the output in figure 4.2. **Gretl** computes the sample mean, median, minimum, maximum, standard deviation (S.D.), coefficient of variation (C.V.), skewness and excess kurtosis for each variable in the data set. You may recall from your introductory statistics courses that there are an equal number of observations in your sample that are larger and smaller in value than the median. The standard deviation is the square root of your sample variance. The coefficient of variation is simply the standard deviation divided by the sample mean. Large values of the C.V. indicate that your mean is not very precisely measured. Skewness is a measure of the degree of symmetry of a distribution. If the left tail (tail at small end of the distribution) extends over a relatively larger range of the variable than the right tail, the distribution is negatively skewed. If the

Figure 4.1: Choosing summary statistics from the pull down menu

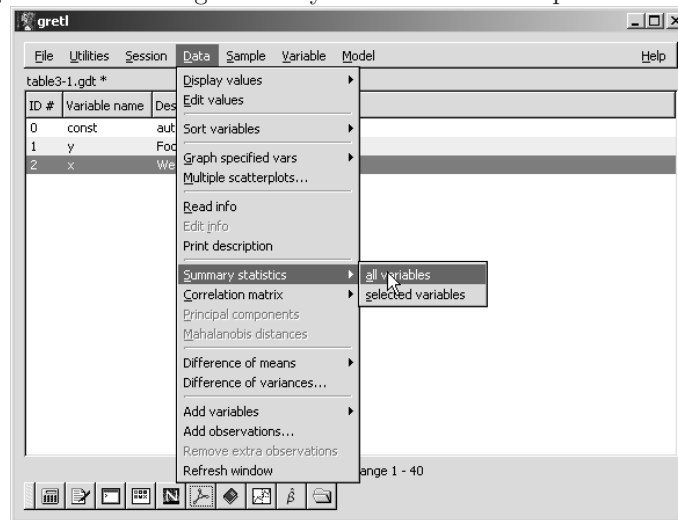


Figure 4.2: Choosing summary statistics from the pull down menu yields these results.

Summary Statistics, using the observations 1 - 40				
Variable	MEAN	MEDIAN	MIN	MAX
y	130.31	120.71	52.250	269.03
x	698.00	712.30	258.30	1154.6
Variable	S.D.	C.V.	SKEW	EXCKURT
y	45.159	0.34654	1.0348	1.1297
x	198.23	0.28399	0.21866	-0.014140

right tail covers a larger range of values then it is positively skewed. Normal and t-distributions are symmetric and have zero skewness. The χ_n^2 is positively skewed. Excess kurtosis refers to the fourth sample moment about the mean of the distribution. ‘Excess’ refers to the kurtosis of the normal distribution, which is equal to three. Therefor if this number reported by **gretl** is positive, then the kurtosis is greater than that of the normal; this means that it is more peaked around the mean than the normal. If excess kurtosis is negative, then the distribution is flatter than the normal.

Sample Statistic	Formula
Mean	$\sum x_i/n = \bar{x}$
Variance	$\frac{1}{n-1} \sum (x_i - \bar{x})^2 = s_x^2$
Standard Deviation	$s = \sqrt{s^2}$
Coefficient of Variation	s/\bar{x}
Skewness	$\frac{1}{n-1} \sum (x_i - \bar{x})^3 / s^3$
Excess Kurtosis	$\frac{1}{n-1} \sum (x_i - \bar{x})^4 / s^4 - 3$

You can also use **gretl** to obtain tail probabilities for various distributions. For example if $X \sim N(3, 9)$ then $P(X \geq 4)$ is

$$P[X \geq 4] = P[Z \geq (4 - 3)/\sqrt{9}] = P[Z \geq 0.334] \doteq 0.3694 \quad (4.2)$$

To obtain this probability, you can use the **Utilities>p value finder** from the pull down menu. Then, give **gretl** the value of X, the mean of the distribution and its standard deviation using the dialog box shown in figure 4.3. The result appears in figure 4.4.

In your book you are given another example $X \sim N(3, 9)$ then find $P(4 \leq X \leq 6)$ is

$$P[4 \leq X \leq 6] = P[0.334 \leq Z \leq 1] = P[Z \leq 1] - P[Z \leq .33] \quad (4.3)$$

Take advantage of the fact that $P[Z \leq z] = 1 - P[Z > z]$ to obtain use the pvalue finder to obtain:

$$(1 - 0.1587) - (1 - 0.3694) = (0.3694 - 0.1587) = 0.2107 \quad (4.4)$$

Note, this value differs slightly from the one given in your book due to rounding error that occurs from using the normal probability table. When using the table, the $P[Z \leq .334]$ was truncated to $P[Z \leq .33]$; this is because your tables are only

Figure 4.3: Dialog box for finding right hand side tail areas of various probability distributions.

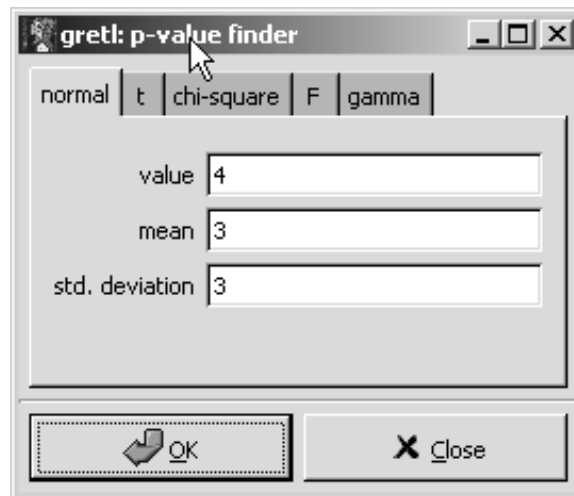
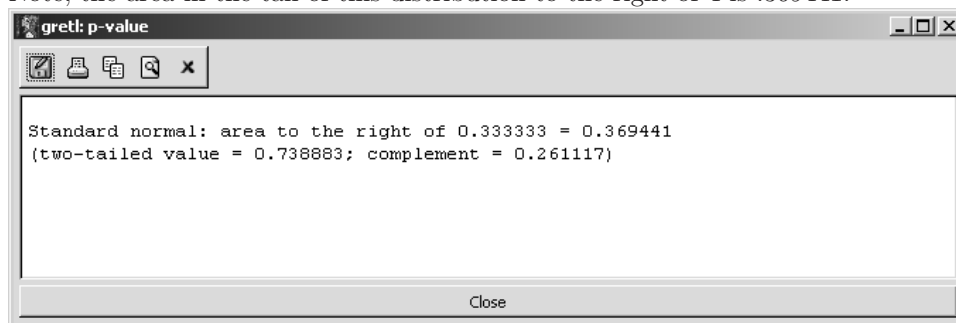


Figure 4.4: Results from the p value finder of $P[X \geq 4]$ where $X \sim N(3, 9)$. Note, the area in the tail of this distribution to the right of 4 is .369441.



taken out to two decimal places and a practical decision was made by the authors of your book to forgo interpolation (contrary to what your Intro to Statistics professor may have told you, it is hardly ever worth the effort to interpolate when you have to do it manually). **Gretl**, on the other hand computes this probability out to machine precision as $P[Z \leq \frac{1}{3}]$. Hence, a discrepancy occurs. Rest assured though that these results are, aside from rounding error, the same.

Chapter 5

Simple Linear Regression

In this chapter you are introduced to the simple linear regression model which is then estimated using the principle of least squares.

5.1 Retrieve the Data

The first step is to load the food expenditure and income data into **gretl**. The data file is included in your **gretl** sample files provided that you have installed the UE2 data supplement that is available from our website. See section 1.2 for details.

Load the data from Table 3.1 of your textbook. Recall, this is accomplished by the commands **File>Open data>sample files** from the menu bar.¹ Choose Table3-1 from the list. When you bring the file containing the data into **gretl** your window will look like the one in figure 5.1. Notice that in the Descriptive label column is blank for the two variables. Before you graph your output or to generate output for a report or paper you may want to label your variables to make the output easier to organize. This can be accomplished by editing the attributes of the variables.

To do this, first highlight the variable whose attributes you want to edit, then go up to the menu bar and click **Variables>Edit attributes** from the pull down menus (see figure 5.2. This yields a dialog box where you can assign variable descriptions and display names. Describe and label the variable y as

¹Alternately, you could click on the open data button on the toolbar. It's the one that looks like a folder on the far right-hand side.

Figure 5.1: Food Expenditure data is imported from Table3-1.

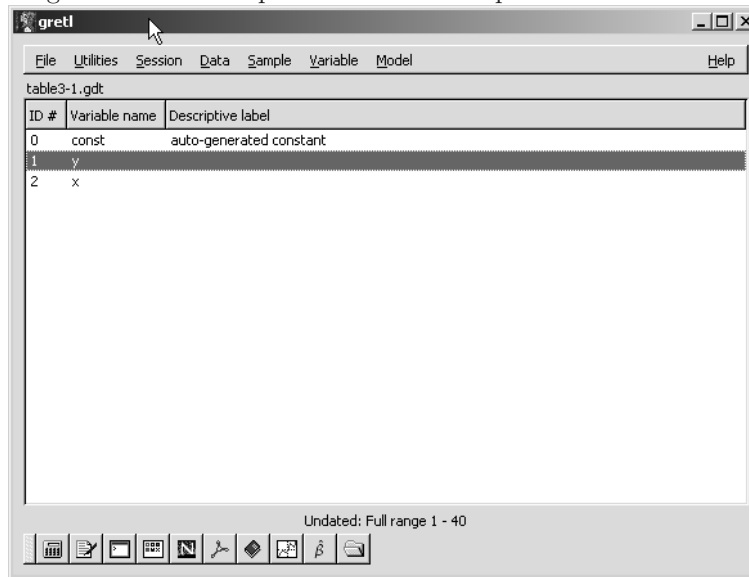
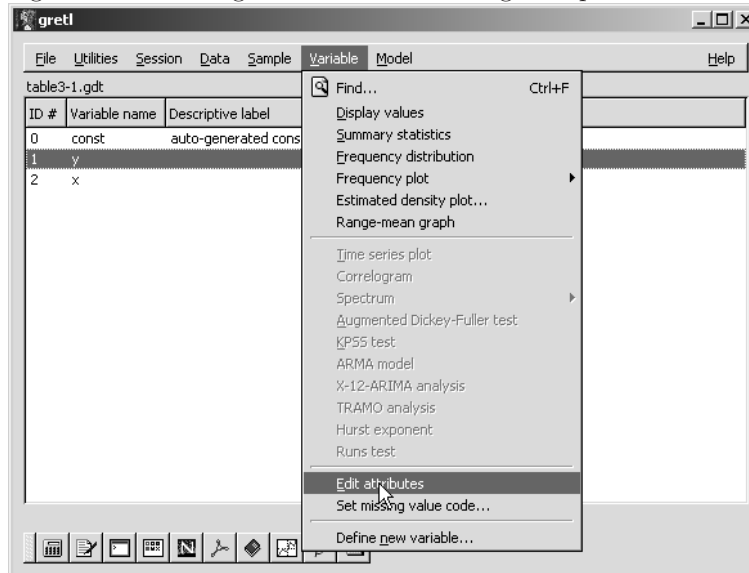
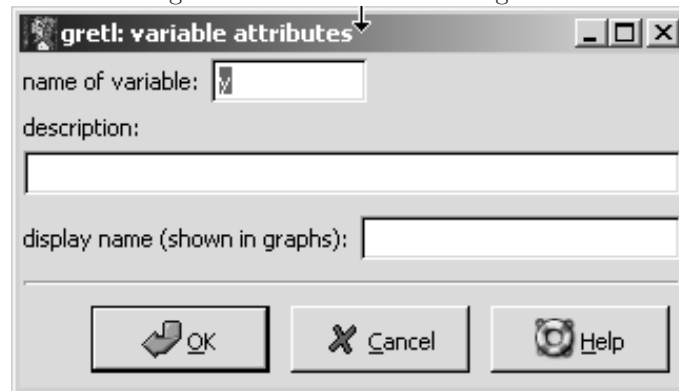


Figure 5.2: Selecting Edit attributes from gretl's pulldown menus



‘Food Expenditure’ and x as ‘Weekly Income.’ An easier way to bring up the variable edit dialog is to highlight the desired variable and to execute a right mouse click. This brings up a pull down menu that allows you to do a number of things to the selected variable, including edit its attributes.

Figure 5.3: Variable edit dialog box



5.2 Graph the Data


To generate a graph of the Food Expenditure data that resembles the one in figure 3.6 of your textbook, you can use the  button on the **gretl** toolbar (third button from the right). Clicking this button brings up a dialog to plot the two variables against one another. Figure 5.4 shows this dialog where x is placed on the x-axis and y on the y-axis. The result appears in figure 5.5. Notice that the labels applied above now appear on the axes of the graph.

Figure 5.5 plots Food Expenditures on the y axis and Weekly Income on the X. **Gretl**, by default, also plots the fitted regression line. More on this later.

5.3 Estimate the Food Expenditure relationship

now you are ready to use **Gretl** to estimate the parameters of the Food Expenditure equation.

$$y = \beta_1 + \beta_2 x + e \quad (5.1)$$

From the menu bar, select **Model>Ordinary Least Squares** from the pull down menu to generate the dialog shown in figure 5.6.

Figure 5.4: Use the dialog to plot of the Food Expenditure (y) against Weekly Income (x)

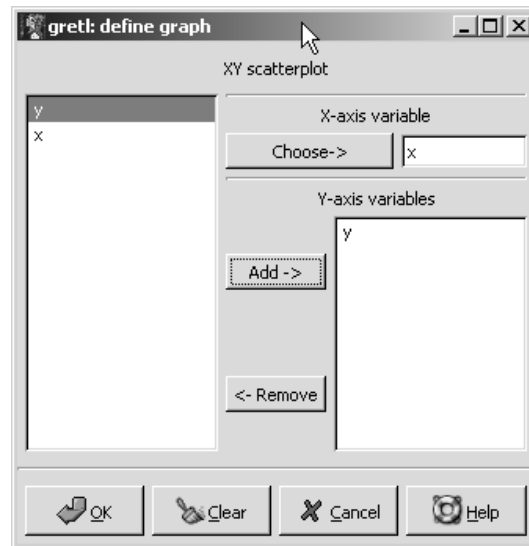


Figure 5.5: XY plot of the Food Expenditure data

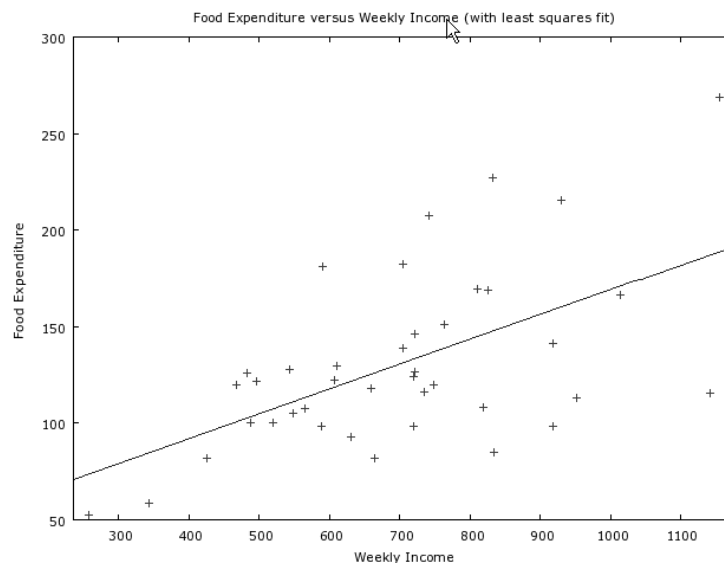



Figure 5.6: From the menu bar, select Model>Ordinary Least Squares to open this dialog box



From this dialog you'll need to tell **gretl** which variable to use as the dependent variable and which is the independent variable. Notice that by default, **gretl** assumes that you want to estimate an intercept (β_1) and includes this in the independent variable list by default. To include *x* as an independent variable, highlight it with the cursor and click the Add button.

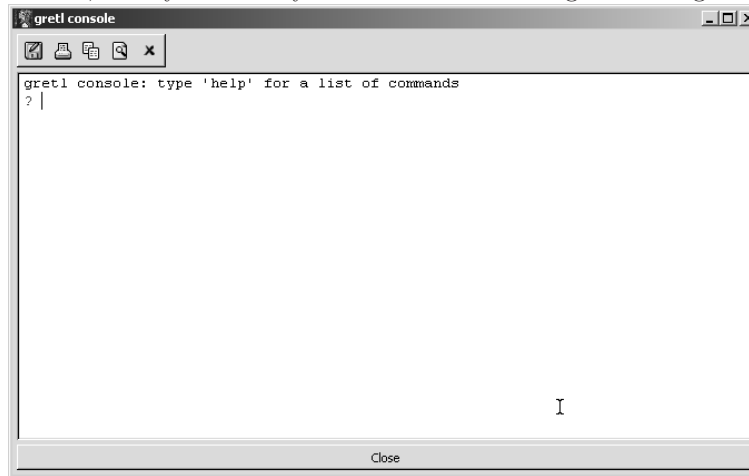
An easy way to run a regression is using the **gretl** console. The **gretl** console is opened by clicking the console button on the toolbar, . This button opens the console shown in figure 5.6.

At the question mark in the console simply type

```
OLS y const x
```

to estimate your regression function. The syntax is very simple, OLS tells **gretl** that you want to estimate a linear function using ordinary least squares. The first variable listed will be your dependent variable and any that follow the independent variables. These names must match the appropriate names of your variables given in your data set. Since ours are named, *y* and *x*, respectively, these are the names used here. Don't forget the constant (*const*).

Figure 5.7: Gretl console. From this window you can type in **gretl** commands directly and perform analyses very quickly—if you know the proper **gretl** commands. If not, then you can rely on the GUI and dialog boxes to guide you.



This yields the following output:

Model 3: OLS estimates using the 40 observations 1–40
Dependent variable: y

Variable	Coefficient	Std. Error	<i>t</i> -statistic	p-value
const	40.7676	22.1387	1.8415	0.0734
x	0.128289	0.0305393	4.2008	0.0002

An equivalent way to present results, especially in very small models like the simple linear regression, is to use equation form. In this format, the **gretl** results are:

$$\hat{y} = 40.7676 + 0.128289x$$

(1.841) (4.201)

$$T = 40 \quad \bar{R}^2 = 0.2991 \quad F(1, 38) = 17.647 \quad \hat{\sigma} = 37.805$$

(*t*-statistics in parentheses)

Chapter 6

Sampling Properties of Least Squares Estimator

Perhaps the best way to illustrate the sampling properties of least squares is through an experiment. In section 4.2.1 of your book you are presented with results from 10 different regressions (UE2 Table 4.1). In this chapter of the manual, you will generate 100 samples of data from the food expenditure data, estimate the slope and intercept parameters with each data set, and then study how the least squares estimator performed over those 100 different samples. What will become clear is this, the outcome from any single sample is a poor indicator of the true value of the parameters. Keep this in mind whenever you estimate a model with what is invariably only 1 sample or instance of the true (but always unknown) data generation process.

We start with the food expenditure model:

$$y = \beta_1 + \beta_2 x + e \tag{6.1}$$

where y is total food expenditure for the given time period and x is income. Suppose further that we know how much income each of 40 households earns in a week. Additionally, we know that on average a household spends at \$50 on food whether it has income or not and that an average household will spend twelve cents of each new dollar of income on additional food. In terms of the regression this translates into parameter values of $\beta_1 = 50$ and $\beta_2 = 0.12$.

Our knowledge of any particular household is considerably less. We don't know how much it actually spends on food in any given week and other than differences based on income, we don't know how their food expenditures might otherwise differ. Food expenditures surely vary for reasons other than income.

Some families are larger than others, tastes and preferences differ, and some may travel more often or farther making food consumption more costly. For whatever reasons, it is impossible for us to know beforehand exactly how much any household will spend on food, even if we know how much income it earns. All of this uncertainty is captured by the error term in the model. For the sake of experimentation, suppose we also know that $e \sim N(0, 35^2)$.

With this knowledge, we can study the properties of the least squares estimator by generating samples of size 40 using the known data generation mechanism. We generate 100 samples using the known parameter values, estimate the model for each using least squares, and then use summary statistics to determine whether least squares, on average anyway, is either very accurate or precise. So in this instance, we know how much each household earns, and we know how much the **average** household spends on food that is not related to income ($\beta_1 = 50$) and how much that expenditure rises **on average** as income rises. What we do not know is how any **particular** household's expenditures are responds to income or how much is autonomous.

A single sample can be generated in the following way. The systematic component of food expenditure for the i th household is $50 + 0.12 * x_i$. This differs from its actual food expenditure by a random amount that varies according to a normal distribution having zero mean and standard deviation equal to 35. So, we use computer generated random numbers to generate a random error, u_i , from that particular distribution. We repeat this for the remaining 39 individuals. The generates one Monte Carlo sample and it is then used to estimate the parameters of the model. The results are saved and then another Monte Carlo sample is generated and used to estimate the model and so on.

In this way, we can generate as many different samples of size 40 as we desire. Furthermore, since we know what the underlying parameters are for these samples, we can later see how close our estimators get to revealing these true values.

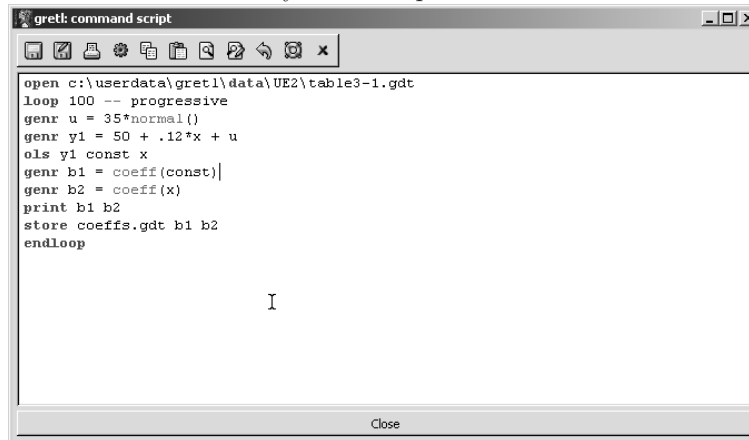
Now, computer generated random numbers are not actually random in the true sense of the word; they can be replicated exactly if you know the mathematical formula used to generate them and the 'key' that initiates the sequence. In most cases, these numbers behave as if they were in fact randomly generated by a physical process.

To conduct an experiment using least square in **gretl** one could use the script found in figure 6.1.

Let's look at what each line accomplishes. The first line

```
open c:\userdata\gretl\data\UE2\table3-1.gdt
```

Figure 6.1: In the **gretl** console window you can use the following commands to execute a Monte Carlo study of least squares.



opens the food expenditure data set that resides in the UE2 folder of the data directory. The loop construct in **gretl** begins with the command `loop NMC --progressive` and ends with `endloop`. NMC in this case is the number of Monte Carlo samples you want to use and the option `--progressive` is a command that suppresses the individual output at each iteration from being printed and to allows you to store the results in a file.

Within this loop construct, you tell **gretl** how to generate each sample and state how you want that sample to be used. The data generation is accomplished here as

```

genr u = 35*normal()
genr y1 = 50 + .12*x + u

```

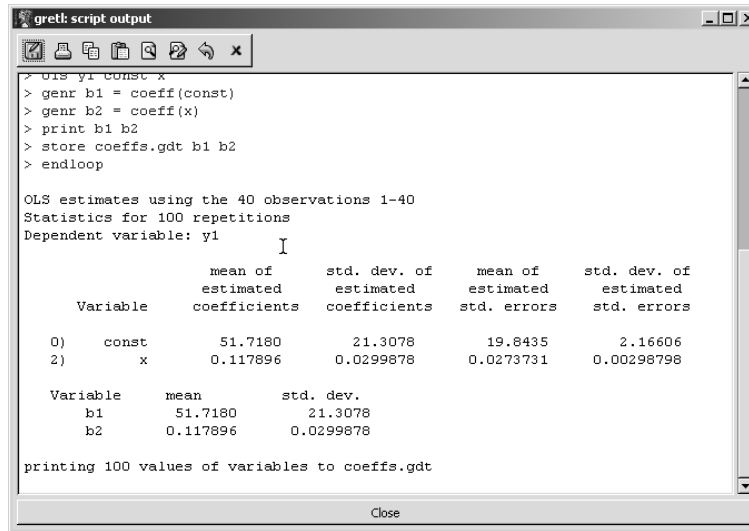
The `genr` command is used to generate new variables. In the first line u is generated by multiplying a normal random variable by the desired standard deviation. Recall, that for any constant, c and random variable, X , $Var(cX) = c^2 Var(X)$. `normal()` produces a computer generated standard normal random variable. The next line adds this random element to the systematic portion of the model to generate a new sample for food expenditures (using the known values of income in x).

Next, the model is estimated using least squares. Then, the coefficients are stored internally in variables you create `a` and `b` (I called them `b1` and `b2`, but you can name them as you like). These are then stored to a data set `coeffs.gdt`.

After executing the script, **gretl** prints out some summary statistics to the

screen. These appear below in figure 6.2. Note that the average value of the

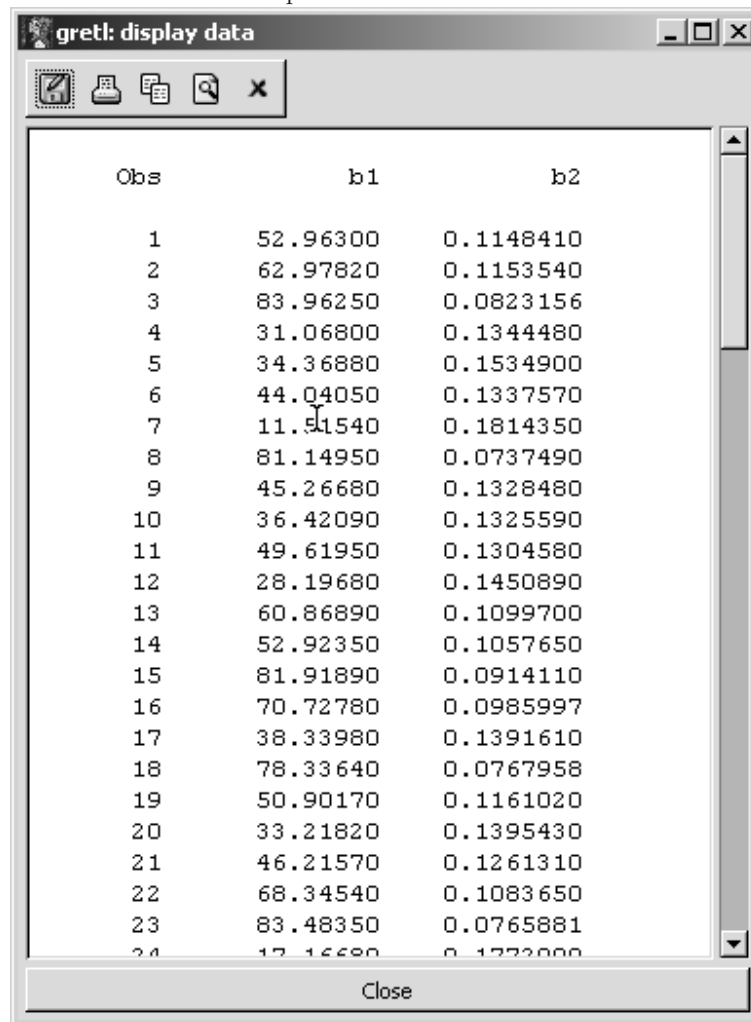
Figure 6.2: The summary results from 100 random samples of the Monte Carlo experiment.



intercept is about 51.718. This is getting close to the truth. The average value of the slope is 0.1179, also close to the true value. If you were to repeat the experiments with larger numbers of Monte Carlo iterations, you will find that these averages get closer to the values of the parameters used to generate the data. This is what it means to be unbiased. Unbiasedness only has meaning within the context of repeated sampling. In your experiments, you generated many samples and averaged results over those samples to get closer to the truth. In actual practice, you do not have this luxury. In practice you have one sample and the proximity of your estimates to the true values of the parameters is always unknown.

After executing the script, open the `coeffs.gdt` data file and view the data. From the example this yields the output in figure 6.3. Notice that even though the actual value of $\beta_1 = 50$ there is considerable variation in the estimates. In sample 12 it was estimated to be 28.19. and in sample 8 it was nearly 81.15. Likewise, β_2 also varies around its true value of .12. Notice that the estimates are never equal to the true parameter value!

Figure 6.3: The results from the first 23 sets of estimates from the 100 random samples of the Monte Carlo experiment.



Obs	b1	b2
1	52.96300	0.1148410
2	62.97820	0.1153540
3	83.96250	0.0823156
4	31.06800	0.1344480
5	34.36880	0.1534900
6	44.04050	0.1337570
7	11.51540	0.1814350
8	81.14950	0.0737490
9	45.26680	0.1328480
10	36.42090	0.1325590
11	49.61950	0.1304580
12	28.19680	0.1450890
13	60.86890	0.1099700
14	52.92350	0.1057650
15	81.91890	0.0914110
16	70.72780	0.0985997
17	38.33980	0.1391610
18	78.33640	0.0767958
19	50.90170	0.1161020
20	33.21820	0.1395430
21	46.21570	0.1261310
22	68.34540	0.1083650
23	83.48350	0.0765881

Chapter 7

Inference in the Simple Linear Regression Model

7.1 Confidence Intervals

The purpose of confidence intervals is to give the user some notion of how variable the parameter estimates are. One way of doing this is to present the least squares parameter estimate along with its estimated standard error. The estimated standard error is an estimate of how precisely least squares is able to measure the parameter of interest.

The confidence interval serves a similar purpose, though it is much more straightforward to interpret because it gives you upper and lower bounds between which the unknown parameter will lie with a given probability.¹

In **gretl** you have to do a little work to compute confidence intervals. They can be constructed manually using the **genr** command, though you can let **gretl** do the arithmetic. To construct an interval in **gretl** you will first need to look up the appropriate critical value from a table in order to get the correct computation.

¹This is probability in the frequency sense. Much ado is made of this (incorrectly I think) in statistics as you are often given stern warnings not to interpret a confidence interval as containing the unknown parameter with the given probability. However, probability in its frequency definition refers to the long run relative frequency with which some event occurs. If this is what probability is, then saying that a parameter falls within an interval with given probability means that intervals so constructed will contain the parameter that proportion of the time.

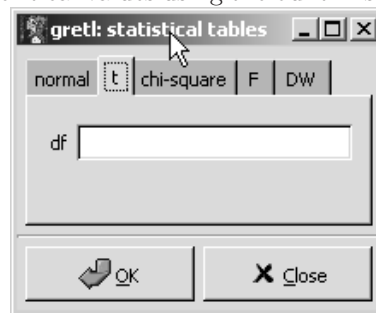
Here is how it works. Taking equation (5.1.13) from your text

$$P[b_2 - t_c se(b_2) \leq \beta_2 \leq b_2 + t_c se(b_2)] = 1 - \alpha \quad (7.1)$$

Recall that b_2 is the least squares estimator of β_2 , and that $se(b_2)$ is its estimated standard error. The constant t_c is the $\alpha/2$ critical value from the t-distribution and α is the total desired probability associated with the “rejection” area (the area outside of the confidence interval).

In **gretl** you’ll need to look up t_c either in a statistical table or using the **Utilities>Statistical tables** dialog contained in the program. The **gretl** dialog box is shown in figure ???. Pick the tab for the t distribution and tell **gretl** how many degrees of freedom your t-statistic has. Once you do, click on OK and choose the the 0.025 critical value for the t_{38} distribution, which is 2.024.

Figure 7.1: Obtaining critical values using the built in statistical tables in **gretl**.



Then generate the lower and upper bounds (using the **gretl** console) with the commands:

```
open c:\userdata\gretl\data\UE2\table3-1.gdt
ols y const x
genr lb = coeff(x) - 2.024*stderr(x)
genr ub = coeff(x) + 2.024*stderr(x)
print lb ub
```

The first line opens the data set. The second line (ols) minimizes the sum of squared errors in a linear model that has y as the dependent variable with a constant and x as independent variables. The next two lines generate the lower and upper bounds for the 95% confidence interval for the slope parameter (β_2). The last line prints the results of the computation.

The consequences of repeated sampling can be explored using a simple Monte Carlo study. In this case, we will add the two statements that compute the lower and upper bounds to our previous program listed in figure 6.1.

The new script looks like this:

```
open c:\userdata\gretl\data\UE2\table3-1.gdt
loop 100 -- progressive
  genr u = 35*normal()
  genr y1 = 50 + .12*x + u
  ols y1 const x
  genr b1 = coeff(const)
  genr b2 = coeff(x)
  genr s1 = stderr(const)
  genr s2 = stderr(x)
  # 2.024 is the .025 critical value from the t(38) distribution
  genr c1L = b1 - 2.024*s1
  genr c1R = b1 + 2.024*s1
  genr c2L = b2 - 2.024*s2
  genr c2R = b2 + 2.024*s2
  print b1
  print b2
  store coeffs.gdt b1 b2 c1L c1R c2L c2R
endloop
```

The results are stored in the gretl data set `coeffs.gdt`. Opening this data set (open `C:\userdata\gretl\user\coeffs.gdt`) and examining the data will reveal interval estimates that vary much like those in Table 5.2 or your textbook.

7.2 Hypothesis Tests

Hypothesis testing allows us to confront any prior notions we may have about the model with what we actually observe. Thus, if before drawing a sample, I believe that autonomous weekly food expenditure is no less than \$40, then once the sample is drawn I can determine via a hypothesis test whether experience is actually consistent with this belief.

In section 5.2.5 of your book the authors test the null hypothesis that $\beta_2 = 0.10$ against the alternative that it is not ($\beta_2 \neq 0.10$). The test statistic is:

$$t = (b_2 - 0.10)/se(b_2) \sim t_{38} \quad (7.2)$$

provided that $\beta_2 = 0.10$ (the null hypothesis is true). Select $\alpha = 0.05$ which makes the critical value for the two sided alternative ($\beta_2 \neq 0.10$) equal to 2.024. The decision rule is to reject H_0 in favor of the alternative if the computed value of your t statistic falls within the rejection region of your test; that is if it is less than -2.024 or greater than 2.024.

The information you need to compute t is on the printout of your least squares estimation. Thus,

Model 2: OLS estimates using the 40 observations 1–40
Dependent variable: y

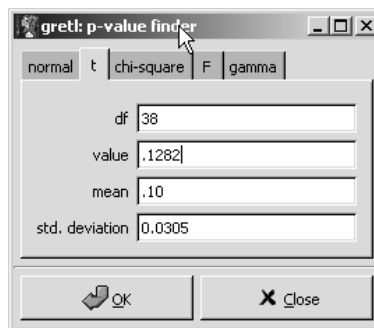
Variable	Coefficient	Std. Error	t -statistic	p-value
const	40.7676	22.1387	1.8415	0.0734
x	0.128289	0.0305393	4.2008	0.0002

The computations

$$t = (b_2 - 0.10)/se(b_2) = (.1282 - .10)/0.0305 = 0.9263 \quad (7.3)$$

Since this value is not within the rejection region, then we do not have enough evidence to dissuade us from our null hypothesis that the coefficient is 0.10; the null hypothesis is not rejected at this level of significance.

Figure 7.2: The dialog box for obtaining p-values using the built in statistical tables in **gretl**.



We can use **gretl** to get the p-value for this test using the Utilities pull down menu. In this dialog, you have to fill in the degrees of freedom for your t -distribution (38), the value of b_2 (.1282), its value under the null hypothesis—something **gretl** refers to as ‘mean’ (.10), and the estimated standard error from your printout (.0305). This will yield the information

```
t(38): area to the right of 0.92459 = 0.180507
(two-tailed value = 0.361014; complement = 0.638986)
```

This indicates that the area in one tail is 0.1805 and that the area in both tails totals 0.36104.

Chapter 8

Using R with Gretl

Another feature of **gretl** that makes it extremely powerful is its ability to work with another free program called R. R is actually a programming language for which many statistical procedures have been written. Although **gretl** is reasonably powerful, there are still many things that it won't do. The ability to export **gretl** data into R makes it possible to do some sophisticated analysis with relative ease.

Quoting from the R web site

R is a language and environment for statistical computing and graphics. It is a GNU project which is similar to the S language and environment which was developed at Bell Laboratories (formerly AT&T, now Lucent Technologies) by John Chambers and colleagues. R can be considered as a different implementation of S. There are some important differences, but much code written for S runs unaltered under R.

R provides a wide variety of statistical (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering, ...) and graphical techniques, and is highly extensible. The S language is often the vehicle of choice for research in statistical methodology, and R provides an Open Source route to participation in that activity.

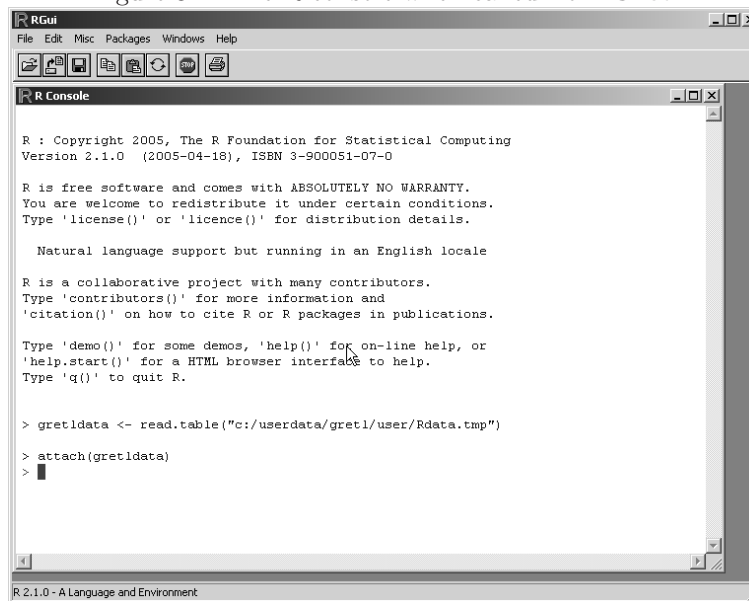
One of R's strengths is the ease with which well-designed publication-quality plots can be produced, including mathematical symbols and formulae where needed. Great care has been taken over the defaults for the minor design choices in graphics, but the user retains full control.

R is available as Free Software under the terms of the Free Software Foundation's GNU General Public License in source code form. It compiles and runs on a wide variety of UNIX platforms and similar systems (including FreeBSD and Linux), Windows and MacOS.

R can be downloaded from <http://www.r-project.org/> which is referred to as CRAN or the comprehensive R archive network. To install R, you'll need to download it and follow the instructions given at the CRAN web site. Also, there is an appendix in the **gretl** manual about using R that you may find useful. The remainder of this brief appendix assumes that you have R installed and linked to **gretl** through the programs tab in the **File>Preferences>General** pull down menu. Make sure that the 'Command to launch GNR R' box points to the **RGui.exe** file associated with your installation of R.

Once you have opened a data set in **gretl**, you may 'start GNU R' using the Utilities pull down menu; when you start R in this fashion, the current **gretl** data set will be transported into R's required format. You'll see the R console which is shown in figure 8.1. To run the regression in R

Figure 8.1: The R console when called from **Gretl**



```
fitols <- lm(y~x,data=gretldata)
```

Before going further, let me comment on this terse piece of computer code. First,

Figure 8.2: The `lm(y x,data=gretldata)` command estimates a linear regression model with `y` as the dependent variable and `x` as an independent variable. R automatically includes an intercept. To print the results to the screen, you have to use the `summary.lm()` command.

```
> gretldata <- read.table("c:/userdata/gretl/user/Rdata.tmp")
> attach(gretldata)
> fitols <- lm(y~x,data=gretldata)
> summary.lm(fitols)

Call:
lm(formula = y ~ x, data = gretldata)

Residuals:
    Min       1Q   Median       3Q      Max
-71.75 -19.67  -5.97   17.75   80.14

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  40.76756    22.13865   1.841 0.073369 .
x             0.12829     0.03054   4.201 0.000155 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 37.81 on 38 degrees of freedom
Multiple R-Squared:  0.3171,    Adjusted R-squared:  0.2991
F-statistic: 17.65 on 1 and 38 DF,  p-value: 0.0001550

> █
```

in R the symbol `<-` is used as the assignment operator; it assigns whatever is on the right hand side (`lm(y~x,data=gretldata)`) to the name you specify on the left (`fitols`). it can be reversed `->` if you want to call the object to its right what is computed on its left. Also, R does not bother to print results unless you ask for them. This is handier than you might think, since most programs produce a lot more output than you actually want and must be coerced into printing less. The `lm` command stands for ‘linear model’ and in this example it contains 2 arguments within the parentheses. The first is your simple regression model. The dependent variable is `y` and the independent variable `x`. They are separated by the symbol `~` which substitutes in this case for an equals sign. The other argument points to the data set that contains these two variables. This data set, pulled into R from **gretl**, is by default called `gretldata`. There are other options for the `lm` command, and you can consult the substantial pdf manual to learn about them. In any event, you’ll notice that when you enter this line and press the return key (which executes this line) R responds by issuing a command prompt, and no results! To print the results from your regression, you issue the command:

```
summary.lm(fitols)
```

which yields the output shown in figure 8.3. Then, to obtain the ANOVA table for this regression

```
anova(fitols)
```

This gives the result in figure 8.3. It's that simple! One thing to note about

Figure 8.3: The `anova(olsfit)` command asks R to print the anova table for the regression results stored in `olsfit`.

```
> anova(fitols)
Analysis of Variance Table

Response: y
      Df Sum Sq Mean Sq F value    Pr(>F)    
x         1  25221    25221  17.647 0.0001550 ***
Residuals 38  54311     1429                      
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

how R reports analysis of variance. It reports the explained variation (25221) in the top line and the unexplained variation in y (54311) below. It does not report total variation. To obtain the total, you just have to add the explained to the unexplained variation together (25221+54311=79532).

To do multiple regression in R, you have to put each of your independent variables (other than the intercept) into a matrix. A matrix is a rectangular array (which means it contains numbers arranged in rows and columns). You can think of a matrix as the rows and columns of numbers that appear in a spreadsheet program like MS Excel. Each row contains an observation on each of your independent variables; each column contains all of the observations on a particular variable. For instance suppose you have two variables, x_1 and x_2 , each having 5 observations. These can be combined horizontally into the matrix, X . Computer programmers sometimes refer to this operation as *horizontal concatenation*. Concatenation essentially means that you connect or link objects in a series or chain; to concatenate horizontally means that you are binding one or more columns of numbers together.

The function in R that binds columns of numbers together is `cbind`. So, to horizontally concatenate x_1 and x_2 use the command

```
X <- cbind(x1,x2)
```

which takes

$$x_1 = \begin{pmatrix} 2 \\ 1 \\ 5 \\ 2 \\ 7 \end{pmatrix}, \quad x_2 = \begin{pmatrix} 4 \\ 2 \\ 1 \\ 3 \\ 1 \end{pmatrix}, \quad \text{and yields } X = \begin{pmatrix} 2 & 4 \\ 1 & 2 \\ 5 & 1 \\ 2 & 3 \\ 7 & 1 \end{pmatrix}.$$

Then the regression is estimated using

```
fitols <- lm(y~X)
```

There is one more thing to mention about R that is very important and this example illustrates it vividly. R is case sensitive. That means that two objects x and X can mean two totally different things to R. Consequently, you have to be careful when defining and calling objects in R to get to distinguish lower from upper case letters.

Chapter 9

Reporting Results and Functional Form

9.1 Coefficient of Determination

One use of regression analysis is to “explain” variation in dependent variable as a function of the independent variable. A summary statistic that is used for this purpose is the coefficient of determination, also known as R^2 .

The R^2 can be computed manually from the analysis of variance table constructed in chapter 8. Figure 8.3 contains the analysis of variance table from a simple linear regression. First, find the total variation in y by adding the explained and unexplained variation together:

$$SSR + SSE = 25221 + 54311 = 79532 \quad (9.1)$$

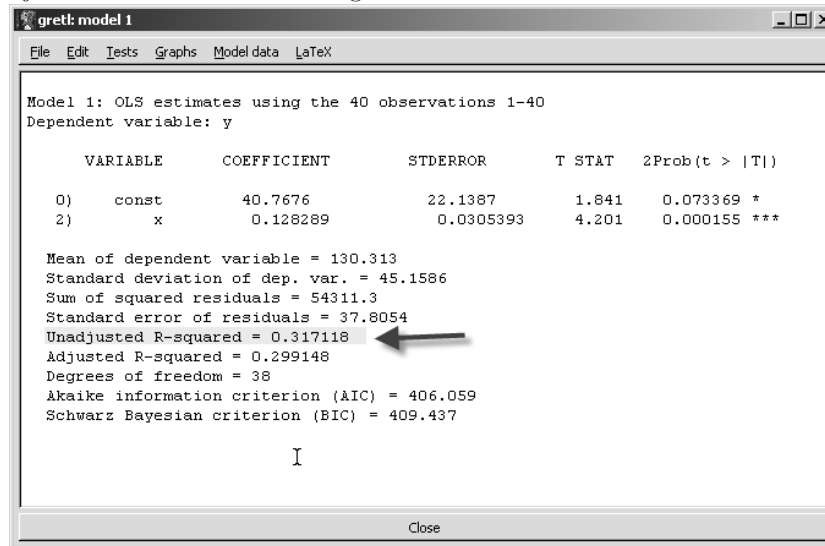
Then, SSR/SST or $1-SSE/SST = 25221/79532 = .317$

The other way is to use **gretl**’s regression output directly. This is shown in figure 9.1.

9.2 Reporting Results

In case you think **gretl** is merely a toy, it includes a very capable utility that enables it to produce professional looking output. LaTeX, usually pronounced

Figure 9.1: In addition to some other summary statistics, **Gretl** computes the unadjusted R^2 from the linear regression.



“Lay-tech”, is typesetting program used by mathematicians and scientists to produce professional looking technical documents. It is widely used by econometricians to prepare manuscripts for wider distribution. In fact, this book is produced in LaTeX.

Although LaTeX is free and can be used to produce very professional looking documents with relative ease, it is not widely used by undergraduate students because it is considered to be relatively hard to learn, especially for those unfamiliar with markup languages (like html, which is used to produce web pages).

In any event, **gretl** includes a facility for producing output that can be pasted directly into LaTeX documents. For users of LaTeX, this makes generating regression output in proper format a breeze. If you don’t already use LaTeX, then this will not concern you. On the other hand, if you already use it, **gretl** can be very handy in this respect.

In figure 9.1 you will notice that on the far right hand side of the menu bar **File Edit Tests Graphs Model data LaTeX** is a pull down menu for LaTeX. From here, you can view, copy, or save the regression output in either tabular form or in equation form. Examples of each are found below in tables 9.2 and 9.2.

Table 9.1: Example of LaTeX output in tabular form

Model 1: OLS estimates using the 40 observations 1–40
Dependent variable: y

Variable	Coefficient	Std. Error	<i>t</i> -statistic	p-value
const	40.7676	22.1387	1.8415	0.0734
x	0.128289	0.0305393	4.2008	0.0002
Mean of dependent variable			130.313	
S.D. of dependent variable			45.1586	
Sum of squared residuals			54311.3	
Standard error of residuals ($\hat{\sigma}$)			37.8054	
Unadjusted R^2			0.317118	
Adjusted \bar{R}^2			0.299148	
Degrees of freedom			38	
Akaike information criterion			406.059	
Schwarz Bayesian criterion			409.437	

Table 9.2: Example of LaTeX output in equation form

$$\hat{y} = 40.7676 + 0.128289 x$$

(1.841) (4.201)

$$T = 40 \quad \bar{R}^2 = 0.2991 \quad F(1, 38) = 17.647 \quad \hat{\sigma} = 37.805$$

(*t*-statistics in parentheses)

9.3 Functional Forms

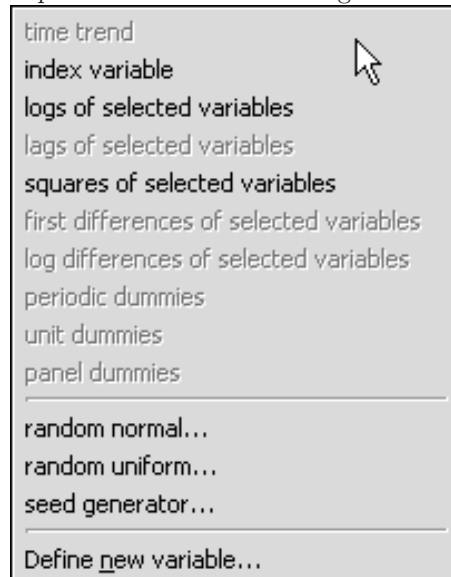
Linear regression is considerably more flexible than its name implies. There is no reason to believe that the relationship between any two variables of interest is necessarily linear. In fact there are many relationships in economics that we know are not linear. The relationship between an input to the production process and output is governed by the law of diminishing returns in the short-run which suggests a convex curve is more appropriate. Fortunately, a simple transformation of the variables (x , y , or both) can still yield a model that is linear in the parameters (but not necessarily in the variables).

Simple transformation of variables can yield regression functions that are quite flexible. The important point to remember, the functional form that you choose should be consistent with how the data are actually being generated. If you choose an inappropriate form, then your estimated model may at best not be very useful and at worst be downright misleading.

In **gretl** you are given a few very useful commands for transforming variables. From the **Data>Add variables** pull down menu you will find a number of transformations that will automatically add the transformed variable and its description to your data set.

Figure 9.2 shows the available selections from this pull down menu. Two of

Figure 9.2: The pull down menu for adding new variables to **gretl**



the options appear in black, the others are greyed out because they are only available if you have time series observations. The available options can be used to add the natural logarithm or the squared values of any highlighted variable to your data set. If neither of these options suits you, then the last option **Define new variable** can be selected. This dialog uses the **genr** command and the large number of built in functions to transform variables in various ways. Just a few of the possibilities include square roots (**sqrt**), sine (**sin**), cosine (**cos**), absolute value (**abs**), exponential (**exp**), minimum (**min**), maximum (**max**), and so on.

9.4 Testing for Normality

Your book discusses the Jarque-Bera test for normality which is computed using the skewness and kurtosis of the least squares residuals. To compute the Jarque-Bera statistic, you'll first need to estimate your model using least squares and then save the residuals to the data set.

From the **gretl** console

```
ols y const x
genr uhat1 = $uhat
summary uhat1
```

The first line is the regression. The next saves the least squares residuals, identified as `$uhat`, into a variable I have called **uhat1**.¹ You could also use the point and click method to add the residuals to the data set. This is accomplished from the output window of your regression. Simply choose **Model data>Add to data set>residuals** from the pull down menu. The last line gives you the summary statistics for the residuals. This yields the output in figure 9.3. One thing to note, **gretl** reports excess kurtosis rather than kurtosis. The excess kurtosis is measured relative to that of the normal distribution which has kurtosis of three. Hence, your computation is

$$JB = \frac{T}{6} \left(\text{Skewness}^2 + \frac{(\text{Excess Kurtosis})^2}{4} \right) \quad (9.2)$$

Which is

$$JB = \frac{40}{6} \left(0.3969^2 + \frac{-0.12585^2}{4} \right) = 1.077 \quad (9.3)$$

Gretl also includes a built in test for normality that has been proposed by Doornik and Hansen (1994). Computationally, it is much more complex than

¹You can't use `uhat` because that name is reserved by **gretl**.

Figure 9.3: The summary statistics for the least squares residuals.

```

? summary uhat1

Summary Statistics, using the observations 1 - 40
for the variable 'uhat1' (40 valid observations)

Mean          I          0.00000
Median        I          -5.9694
Minimum       I          -71.753
Maximum       I           80.140
Standard deviation          37.318
C.V.          I          7.0026E+016
Skewness      I           0.39692
Ex. kurtosis  I          -0.12585

```

the Jarque-Bera test. The Doornik-Hansen test also has a χ^2 distribution if the null hypothesis of normality is true. It can be produced from the **gretl** console after running a regression using the command `testuhat`.

Bibliography

- Davidson, Russell and James G. MacKinnon (2004), *Econometric Theory and Methods*, Oxford University Press, New York.
- Doornik, J. A. and H. Hansen (1994), ‘An omnibus test for univariate and multivariate normality’, working paper, Nuffield College, Oxford.
- Greene, William H. (2003), *Econometric Analysis*, 5th edn, Prentice Hall, Upper Saddle River, N.J.
- Hill, R. Carter, William E. Griffiths and George G. Judge (2001), *Undergraduate Econometrics*, second edn, John Wiley and Sons.
- Ramanathan, Ramu (2002), *Introductory Econometrics with Applications*, The Harcourt series in economics, 5th edn, Harcourt College Publishers, Fort Worth.
- Stock, James H. and Mark W. Watson (2003), *Introduction to Econometrics*, Addison Wesley, Boston, MA.
- Wooldridge, Jeffrey M. (2003), *Introductory Econometrics : a Modern Approach*, 2nd edn, South-Western College Publishers, Cincinnati, Ohio.