MODULE ONE, PART ONE: DATA-GENERATING PROCESSES

William E. Becker
Professor of Economics, Indiana University, Bloomington, Indiana, USA
Adjunct Professor of Commerce, University of South Australia, Adelaide, Australia
Research Fellow, Institute for the Study of Labor (IZA), Bonn, Germany
Editor, *Journal of Economic Education*
Editor, *Social Scinece Reseach Network: Economic Research Network Educator*

This is Part One of Module One. It highlights the nature of data and the data-generating process, which is one of the key ideas of modern day econometrics. The difference between cross-section and time-series data is presented and followed by a discussion of continuous and discrete dependent variable data-generating processes. Least-squares and maximum-likelihood estimation is introduced along with analysis of variance testing. This module assumes that the user has some familiarity with estimation and testing previous statistics and introductory econometrics courses. Its purpose is to bring that knowledge up-to-date. These contemporary estimation and testing procedures are demonstrated in Parts Two, Three and Four, where data are respectively entered into LIMDEP, STATA and SAS for estimation of continuous and discrete dependent variable models.

CROSS-SECTION AND TIME-SERIES DATA

In the natural sciences, researchers speak of collecting data but within the social sciences it is advantageous to think of the manner in which data are generated either across individuals or over time. Typically, economic education studies have employed cross-section data. The term cross-section data refer to statistics for each in a broad set of entities in a given time period, for example 100 Test of Economic Literacy (TEL) test scores matched to time usage for final semester 12^{th} graders in a given year. Time-series data, in contrast, are values for a given category in a series of sequential time periods, i.e., the total number of U.S. students who completed a unit in high school economics in each year from 1980 through 2008. Cross-section data sets typically consist of observations of different individuals all collected at a point in time. Time-series data sets have been primarily restricted to institutional data collected over particular intervals of time.

More recently empirical work within education has emphasized panel data, which are a combination of cross-section and time-series data. In panel analysis, the same group of individuals (a cohort) is followed over time. In a cross-section analysis, things that vary among individuals, such as sex, race and ability, must either be averaged out by randomization or taken into account via controls. But sex, race, ability and other personal attributes tend to be constant from one time period to another and thus do not distort a panel study even though the assignment of individuals among treatment/control groups is not random. Only one of these four modules will be explicitly devoted to panel data.

CONTINUOUS DEPENDENT (TEST SCORE) VARIABLES

Test scores, such as those obtained from the TEL or Test of Understanding of College Economics (TUCE), are typically assumed to be the outcome of a continuous variable Y that may be generated by a process involving a deterministic component (e.g., the mean of Y, μ_y, which might itself be a function of some explanatory variables $X_1, X_2 \ldots X_k$) and the purely random perturbation or error term components v and ε:

$$Y_{it} = \mu_y + v_{it} \quad \text{or} \quad Y_{it} = \beta_1 + \beta_2 X_{it2} + \beta_3 X_{it3} + \beta_4 X_{it4} + \varepsilon_{it},$$

where Y_{it} is the test score of the i^{th} person at time t and the it subscripts similarly indicate observations for the i^{th} person on the X explanatory variables at time t. Additionally, normality of the continuous dependent variable is ensured by assuming the error term components are normally distributed with means of zero and constant variances: $v_{it} \sim N(0, \sigma_v^2)$ and $\varepsilon_{it} \sim N(0, \sigma_\varepsilon^2)$.

As a continuous random variable, which gets its normal distribution from epsilon, at least theoretically any value is possible. But as a test score, Y is only supported for values greater than zero and less than the maximum test score, which for the TUCE is 30. In addition, multiple-choice test scores like the TUCE can only assume whole number values between 0 and 30, which poses problems that are addressed in these four modules.

The change score model (also known as the value-added model, gain score model or achievement model) is just a variation on the above basic model:

$$Y_{it} - Y_{it-1} = \lambda_1 + \lambda_2 X_{it2} + \lambda_3 X_{it3} + \lambda_4 X_{it4} + u_{it},$$

where $Y_{it\text{-}1}$ is the test score of the i^{th} person at time $t-1$. If one of the X variables is a bivariate dummy variable included to capture the effect of a treatment over a control, then this model is called a difference in difference model:

[(mean treatment effect at time t) – (mean control effect at time t)] –
[(mean treatment effect at time t−1) − (mean control effect at time t−1)]
$= [E(Y_{it} | treatment = 1) - E(Y_{it} | treatment = 0)] - [E(Y_{it\text{-}1} | treatment = 1) - E(Y_{it\text{-}1} | treatment = 0)]$
$= [E(Y_{it} | treatment = 1) - E(Y_{it\text{-}1} | treatment = 1)] - [E(Y_{it} | treatment = 0) - E(Y_{it\text{-}1} | treatment = 0)]$
= the lambda on the bivariate treatment variable.

Y_{it} is now referred to as the post-treatment score or posttest and $Y_{it\text{-}1}$ is the pre-treatment score or pretest. Again, the dependent variable $Y_{it} - Y_{it\text{-}1}$ can be viewed as a continuous random variable, but for multiple-choice tests, this difference is restricted to whole number values and is bounded by the absolute value of the test score's minimum and maximum.

This difference in difference model is often used with cross-section data that ignores time-series implications associated with the dependent variable (and thus the error term) involving two periods. For such models, ordinary least-squares estimation as performed in

EXCEL and all other computer programs is sufficient. However, time sequencing of testing can cause problems. For example, as will be demonstrated in Module Three on sample selection, it is not a trivial problem to work with observations for which there is a pretest (given at the start of the term) but no posttest scores because the students dropped out of the class before the final exam was given. Single equation least-squares estimators will be biased and inconsistent if the explanatory variables and error term are related because of time-series problems.

Following the lead of Hanushek (1986, 1156-57), the change-score model has been thought of as a special case of an allegedly superior regression involving a lagged dependent variable, where the coefficient of adjustment (λ_0^*) is set equal to one for the change-score model:

$$Y_{it} = \lambda_0^* Y_{it-1} + \lambda_1^* + \lambda_2^* X_{it2} + \lambda_3^* X_{it3} + \lambda_4^* X_{it4} + \omega_{it} .$$

Allison (1990) rightfully called this interpretation into question, arguing that these are two separate models (change score approach and regressor variable approach) involving different assumptions about the data generating process. If it is believed that there is a direct causal relationship $Y_{it-1} \Rightarrow Y_{it}$ or if the other explanatory X variables are related to the $Y_{it\text{-}1}$ to Y_{it} transition, then the regressor variable approach is justified. But, as demonstrated to economic educators as far back as Becker (1983), the regressor variable model has a built-in bias associated with the regression to the mean phenomenon. Allison concluded, "The important point is that there should be no automatic preference for either model and that the only proper basis for a choice is a careful consideration of each empirical application . . . . In ambiguous cases, there may be no recourse but to do the analysis both ways and to trust only those conclusions that are consistent across methods." (p. 110)

As pointed out by Allison (1990) and Becker, Greene and Rosen (1990), at roughly the same time, and earlier by Becker and Salemi (1977) and later by Becker (2004), models to avoid are those that place a change score on the left-hand side and a pretest on the right. Yet, educational researchers continue to employ this inherently faulty design. For example, Hake (1998) constructed a "gap closing variable (g)" as the dependent variable and regressed it on the pretest:

$$g = gap\ closing = \frac{posttest\ score - pretest\ score}{maximum\ \ score - pretest\ score} = f(pretest\ score ...)$$

where the pretest and posttest scores where classroom averages on a standardized physics test, and maximum score was the highest score possible. Apparently, Hake was unaware of the literature on the gap-closing model. The outcome measure g is algebraically related to the starting position of the student as reflected in the pretest: g falls as the *pretest score* rises, for $maximum\ score \geq posttest\ score \geq pretest\ score$.[i] Any attempt to regress a posttest-minus-pretest change score, or its standardized gap-closing measure g on a pretest score yields a biased estimate of the pretest effect.[ii]

As an alternative to the change-score models [of the type posttest − pretest= f(treatement, . . .) or posttest = f(pretest, treatment, …)], labor economics have turned to a

difference-in-difference model employing a panel data specification to assess treatment effects. But not all of these are consistent with the change score models discussed here. For example, Bandiera, Larcinese and Rasul (2010) wanted to assess the effect in the second period of providing students with information on grades in the first period. In the first period, numerical grade scores were assigned to each student for course work, but only those in the treatment were told their scores, and in the second period numerical grade score were given on essays. That is, the treatment dummy variable reflected whether or not the student obtained grade information (feedback) on at least 75 percent of his or her course work in the first period, and zero if not. This treatment dummy then entered in the second period as an explanatory variable for the essay grade.

More specifically, Bandiera, Larcinese and Rasul estimated the following panel data model for the i^{th} student, enrolled on a degree program offered by department d, in time period t,

$$g_{idct} = \alpha_i + \beta [F_c \times T_t] + \gamma T_t + \delta X_c + \sum_{d'} \mu_{d'} TD_{id'} + \varepsilon_{idct}$$

where g_{idct} is the i^{th} student's grade in department d for course (or essay) c at time t and α_i is a fixed effect that captures time-invariant characteristics of the student that affect his or her grade across time periods, such as his or her underlying motivation, ability, and labor market options upon graduation. Because each student can only be enrolled in one department or degree program, α_i also captures all department and program characteristics that affect grades in both periods, such as the quality of teaching and the grading standards. F_c is a equal to one if the student obtains feedback on his or her grade on course c and T_t identifies the first or second time period, X_c includes a series of course characteristics that are relevant for both examined courses and essays, and all other controls are as previously defined. $TD_{id'}$ is equal to one if student i took any examined courses offered by department d' and is zero otherwise; it accounts for differences in grades due to students taking courses in departments other than their own department d. Finally, ε_{idct} is a disturbance term.

As specified, this model does not control for past grades (or expected grades), which is the essence of a change-score model. It should have been specified as either

$$g_{idct} = \alpha_i + \omega g_{idct-1} + \beta [F_c \times T_t] + \gamma T_t + \delta X_c + \sum_{d'} \mu_{d'} TD_{id'} + \varepsilon_{idct}$$

or

$$g_{idct} - g_{idct-1} = \alpha_i + \omega g_{idct-1} + \beta [F_c \times T_t] + \gamma T_t + \delta X_c + \sum_{d'} \mu_{d'} TD_{id'} + \varepsilon_{idct}$$

Obviously, there is no past grade for the first period and that is in part why a panel data set up has historically not been used when only "pre" and "post" measures of performance are available. Notice that the treatment dummy variable coefficient β is inconsistently estimated with bias if the relevant past course grades in the second period essay-grade equation are omitted. As discuss in Module Three on panel data studies, bringing in a lagged dependent variable into panel data analysis poses more estimation problems. The thing emphasized here is that a change-score model must be employed in assessing a treatment effect. In Module Four,

propensity score matching models are introduced for a means of doing this as an alternative to the least squares method employed in this module.

DISCRETE DEPENDENT VARIABLES

In many problems, the dependent variable cannot be treated as continuous. For example, whether one takes another economics course is a bivariate variable that can be represented by $Y =$ 1, if yes or 0, if not, which is a discrete choice involving one of two options. As another example, consider count data of the type generated by the question how many more courses in economics will a student take? 0, 1, 2 … where increasing positive values are increasingly unlikely. Grades provide another example of a discrete dependent variable where order matters but there are no unique number line values that can be assigned. The grade of A is better than B but not necessarily by the same magnitude that B is better than C. Typically A is assigned a 4, B a 3 and C a 2 but these are totally arbitrary and do not reflect true number line values. The dependent variable might also have no apparent order, as the choice of a class to take in a semester – for example, in the decision to enroll in economics 101, sociology 101, psychology 101 or whatever, one course of study cannot be given a number greater or less than another with the magnitude having meaning on a number line.

In this module we will address the simplest of the discrete dependent variable models; namely, those involving the bivariate dependent variable in the linear probability, probit and logit models.

Linear Probability Model

Consider the binary choice model where $Y_i = 1$, with probability P_i, or $Y_i = 0$, with probability $(1–P_i.)$. In the linear probability regression model $Y_i = \beta_1 + \beta_2 x_i + \varepsilon_i$, $E(\varepsilon_i) = 0$ implies $E(Y_i \mid x_i) = \beta_1 + \beta_2 x_i$, where also $E(Y_i \mid x_i) = (0)[1-(P_i \mid x_i)] + (1)(P_i \mid x_i) = P_i \mid x_i$. Thus, $E(Y_i \mid x_i) = \beta_1 + \beta_2 x_i = P_i \mid x_i$, which we will write simply as P_i. That is, the expected value of the 0 or 1 bivariate dependent variable, conditional on the explanatory variable(s), is the probability of a success ($Y = 1$). We can interpret a computer-generated, least-squares prediction of $E(Y|x)$ as the probability that $Y = 1$ at that x value.

In addition, the mean of the population error in the linear probability model is zero:

$$\begin{aligned} E(\varepsilon) &= (1-\beta_1-\beta_2 x)P + (0-\beta_1-\beta_2 x)(1-P) \\ &= P-\beta_1-\beta_2 x = P - E(Y \mid x) = 0 \text{ for } P = E(Y \mid x) \end{aligned}$$

However, the least squares $\hat{Y}$ can be negative or greater than one, which makes it a peculiar predictor of probability. Furthermore, the variance of epsilon is

$$\text{var}(\varepsilon) = P_i[1-(\beta_1+\beta_2 x_i)]^2 + (1–P_i)(\beta_1+\beta_2 x_i)^2 = P_i(1-P_i)^2 + (1–P_i)P_i^2 = P_i(1–P_i),$$

which (because P_i depends on x_i) means that the linear probability model has a problem of heteroscedasticity.

An adjustment for heteroscedasticity in the linear probability model can be made via a generalized least-squares procedure but the problem of constraining $\beta_1 + \beta_2 x_i$ to the zero – one interval cannot be easily overcome. Furthermore, although predictions are continuous, epsilon cannot be assumed to be normally distributed as long as the dependent variable is bivariate, which makes suspect the use of the computer-generated *t* statistic. It is for these reasons that linear probability models are no longer widely used in educational research.

Probit Model

Ideally, the estimates of the probability of success ($Y = 1$) will be consistent with probability theory with values in the 0 to 1 interval. One way to do this is to specify a probit model, which is then estimated by computer programs such as LIMDEP, SAS and STATA that use maximum likelihood routines. Unlike least squares, which selects the sample regression coefficient to minimize the squared residuals, maximum likelihood selects the coefficients in the assumed data-generating model to maximize the probability of getting the observed sample data.

The probit model starts by building a bridge or mapping between the 0s and 1s to be observed for the bivariate dependent variable and an unobservable or hidden (latent) variable that is assumed to be the driving force for the 0s and 1s:

$$I_i^* = \beta_1 + \beta_2 X_{i2} + \beta_3 X_{i3} + \beta_4 X_{i4} + \varepsilon_i = X_i \beta \text{ , where } \varepsilon_{it} \sim N(0,1) .$$

and $I^* > 0$ implies $Y = 1$ and $I^* \leq 0$ implies $Y = 0$ and
$P_i = P(Y = 1 \mid X_i) = G(I_i^* > 0) = G(Z_i \leq X_i \beta)$.

$G(\ )$ and $g(\ )$ are the standard normal distribution and density functions, and
$P(Y = 1) = \int_{-\infty}^{X\beta} g(t)dt$.

Within economics the latent variable I^* is interpreted as net utility or propensity to take action. For instance, I^* might be interpreted as the net utility of taking another economics course. If the net utility of taking another economics course is positive, then I^* is positive, implying another course is taken and $Y = 1$. If the net utility of taking another economics course is negative, then the other course is not taken, I^* is negative and $Y = 0$.

The idea behind maximum likelihood estimation of a probit model is to maximize the density L with respect to β and σ where the likelihood function is

$$L = f(\varepsilon) = (2\pi\sigma^2)^{-n/2} \exp(-\boldsymbol{\varepsilon}'\boldsymbol{\varepsilon} / 2\sigma^2)$$
$$= (2\pi\sigma^2)^{-n/2} \exp[-(\mathbf{y} - \mathbf{X}\beta)'(\mathbf{y} - \mathbf{X}\beta) / 2\sigma^2]$$

The calculation of $\partial L / \partial \beta$ is not convenient but the logarithm (ln) of the likelihood function is easily differentiated

$$\partial \ln L / \partial \beta = L^{-1} \partial L / \partial \beta .$$

Intuitively, the strategy of maximum likelihood (ML) estimation is to maximize (the log of) this joint density for the observed data with respect to the unknown parameters in the beta vector, where σ is set equal to one. The probit maximum likelihood computation is a little more difficult than for the standard classical regression model because it is necessary to compute the integrals of the standard normal distribution. But computer programs can do the ML routines with ease in most cases if the sample sizes are sufficiently large. See William Greene, *Econometric Analysis* (5th Edition, 2003, pp. 670-671) for joint density and likelihood function that leads to the likelihood equations for $\partial \ln L / \partial \beta$.

The unit of measurement and thus the magnitude of the probit coefficients are set by the assumption that the variance of the error term ε is unity. That is, the estimated probit coefficients along a number line have no meaning. If the explanatory variables are continuous, however, the probit coefficients can be employed to calculate a marginal probability of success at specific values of the explanatory variables:

$\partial p(x) / \partial x = g(X\beta)\beta_x$, where $g(\ )$ is density $g(z) = \partial G(z) / \partial z$.

Interpreting coefficients for discrete explanatory variables is more cumbersome as demonstrated graphically in Becker and Waldman (1989) and Becker and Kennedy (1992).

Logit Model

An alternative to the probit model is the logit model, which has nearly identical properties to the probit, but has a different interpretation of the latent variable *I**. To see this, again let

$$P_i = E(Y = 1 \mid X_i) .$$

The logit model is then obtained as an exponential function

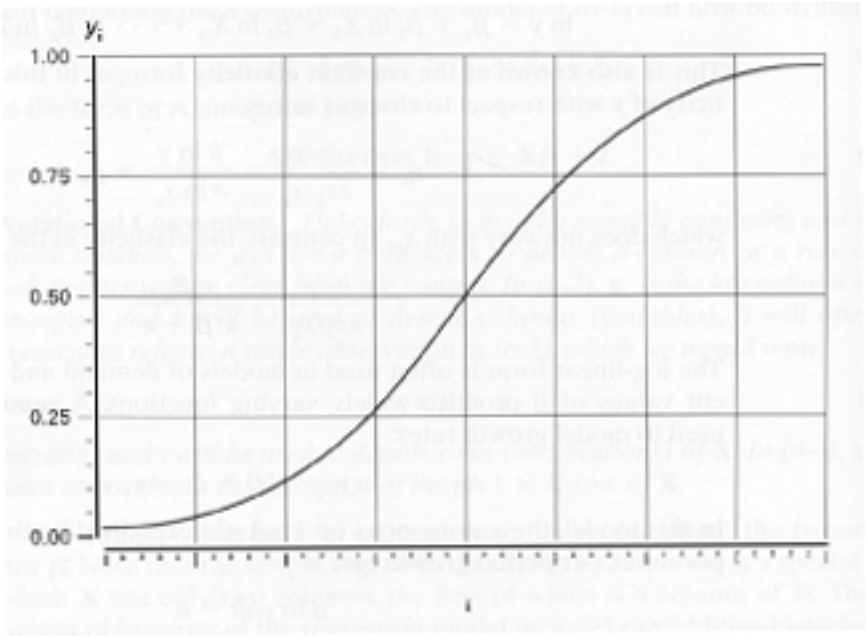
$$P_i = 1/(1 + e^{-X_i\beta}) = 1/(1 + e^{-z_i}) = e^{z_i} /(1 + e^{z_i}) \text{ ; thus,}$$
$$1 - P_i = 1 - e^{z_i} /(1 + e^{z_i}) = 1/(1 + e^{z_i}) \text{, and}$$
$$P_i /(1 - P_i) = e^{z_i} \text{, which is the odd ratio for success } (Y = 1)$$

The log odds ratio is the latent variable logit equation

$$I_i^* = \ln\left(\frac{P_i}{1-P_i}\right) = z_i = X_i\beta\,.$$

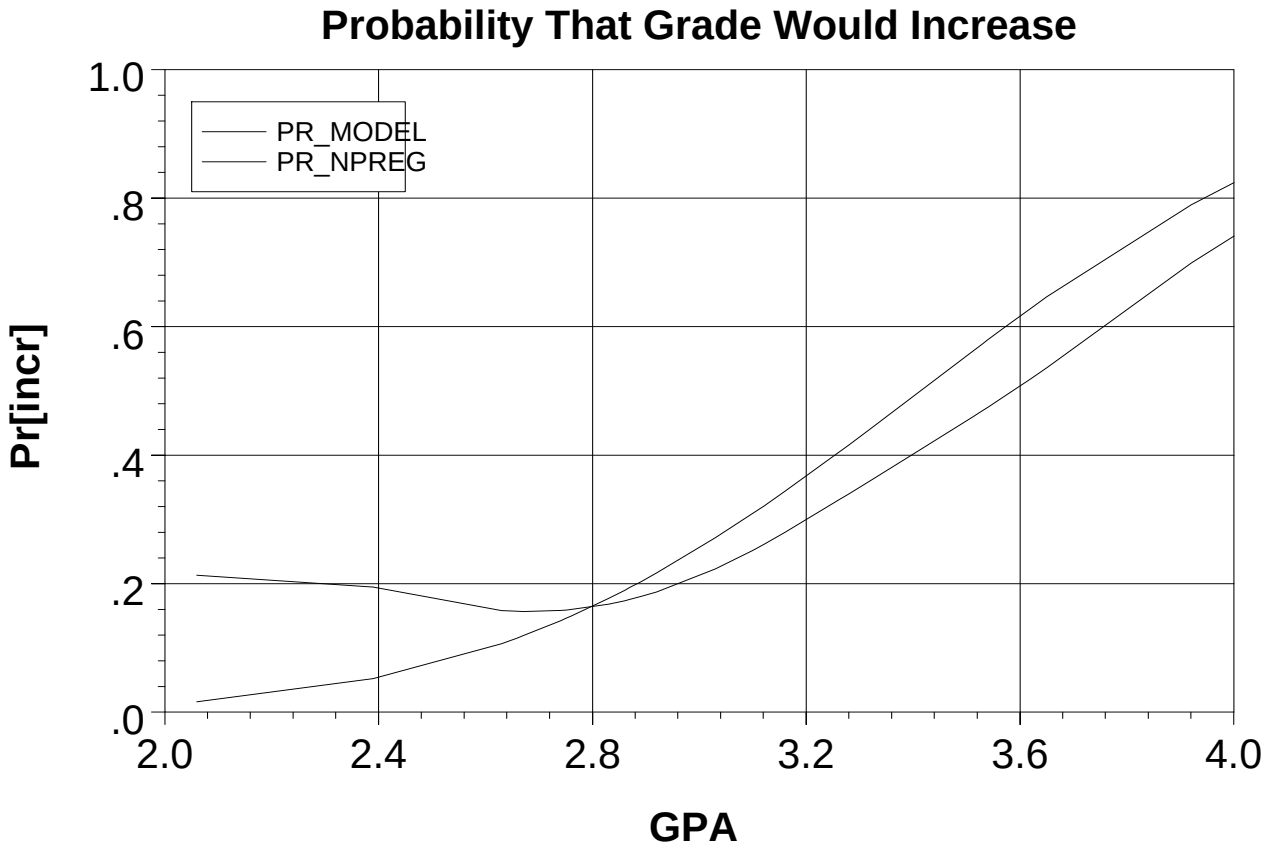
A graph of the logistic function *G(z) = exp(z)/*[1+*exp(z)*] looks like the standard normal, as seen in the following figure, but does not rise or fall to 1.00 and 0.00 as fast:

Graph of Logistic Function



Nonparametrics

As outlined in Becker and Greene (2001), recent developments in theory and computational procedures enable researchers to work with nonlinear modeling of all sorts as well as nonparametric regression techniques. As an example of what can be done consider the widely cited economic education application in Spector and Mazzeo (1980). They estimated a probit model to shed light on how a student's performance in a principles of macroeconomics class relates to his/her grade in an intermediate macroeconomics class, after controlling for such things as grade point average (GPA) going into the class. The effect of GPA on future performance is less obvious than it might appear at first. Certainly it is possible that students with the highest GPA would get the most from the second course. On the other hand, perhaps the best students were already well equipped, and if the second course catered to the mediocre (who had more to gain and more room to improve) then a negative relationship between GPA and increase in grades (GRADE) might arise. A negative relationship might also arise if artificially high grades were given in the first course. The below figure provides an analysis similar to that done by Spector and Mazzeo (using a subset of their data).



In this figure, the horizontal axis shows the initial grade point average of students in the study. The vertical axis shows the relative frequency of the incremental grades that increase from the first to the second course. The solid curve shows the estimated relative frequency of grades that improve in the second course using a probit model (the one used by the authors). These estimates suggest a positive relationship between GPA and the probability of grade improvement in the second macroeconomics throughout the GPA range. The dashed curve in the figure provides the results using a much less-structured nonparametric regression model.[iii] The conclusion reached with this technique is qualitatively similar to that obtained with the probit model for GPAs above 2.6, where the positive relationship between GPA and the probability of grade improvement can be seen, but it is materially different for those with GPAs lower than 2.6, where a negative relationship between GPA and the probability of grade improvement is found. Possibly these poorer students received gift grades in the introductory macroeconomics course.

There are other alternatives to least squares that economic education researchers can employ in programs such as LIMDEP, STATA and SAS. For example, the least-absolute-deviations approach is a useful device for assessing the sensitivity of estimates to outliers. It is likely that examples can be found to show that even if least-squares estimation of the conditional mean is a better estimator in large samples, least-absolute-deviations estimation of the conditional median performs better in small samples. The critical point is that economic education researchers must recognize that there are and will be new alternatives to modeling and

estimation routines as currently found in *Journal of Economic Education* articles and articles in the other journals that publish this work, as listed in Lo, Wong and Mixon (2008). In this module and in the remaining three, only passing mention will be given to these emerging methods of analysis. The emphasis will be on least-squares and maximum-likelihood estimations of continuous and discrete data-generating processes that can be represented parametrically.

INDIVIDUAL OBSERVATIONS OR GROUP AVERAGES: WHAT IS THE UNIT OF ANALYSIS?

In Becker (2004), I called attention to the implications of working with observations on individuals versus working with averages of individuals in different groupings. For example, what is the appropriate unit of measurement for assessing the validity of student evaluations of teaching (as reflected, for example, in the relationship between student evaluations of teaching and student outcomes)? In the case of end-of-term student evaluations of instructors, an administrator's interest may not be how students as individuals rate the instructor but how the class as a whole rates the instructor. Thus, the unit of measure is an aggregate for the class. There is no unique aggregate, although the class mean or median response is typically used.[iv] For the assessment of instructional methods, however, the unit of measurement may arguably be the individual student in a class and not the class as a unit. Is the question: how is the i^{th} student's learning affected by being in a classroom where one versus another teaching method is employed? Or is the question: how is the class's learning affected by one method versus another? The answers to these questions have implications for the statistics employed and interpretation of the results obtained.[v]

Hake (1998) reported that he has test scores for 6,542 individual students in 62 introductory physics courses. He works only with mean scores for the classes; thus, his effective sample size is 62, and not 6,542. The 6,542 students are not irrelevant, but they enter in a way that I did not find mentioned by Hake. The amount of variability around a mean test score for a class of 20 students versus a mean for 200 students cannot be expected to be the same. Estimation of a standard error for a sample of 62, where each of the 62 means receives an equal weight, ignores this heterogeneity.[vi] Francisco, Trautman, and Nicoll (1998) recognized that the number of subjects in each group implies heterogeneity in their analysis of average gain scores in an introductory chemistry course. Similarly, Kennedy and Siegfried (1997) made an adjustment for heterogeneity in their study of class size on student learning in economics.

Fleisher, Hashimoto, and Weinberg (2002) considered the effectiveness (in terms of student course grades and persistences) of 47 foreign graduate student instructors versus 21 native English speaking graduate student instructors in an environment in which English is the language of the majority of their undergraduate students. Fleisher, Hashimoto, and Weinberg recognized the loss of information in using the 92 mean class grades for these 68 graduate student instructors, although they did report aggregate mean class grade effects with the corrected heterogeneity adjustment for standard errors based on class size. They preferred to look at 2,680 individual undergraduate results conditional on which one of the 68 graduate student instructors each of the undergraduates had in any one of 92 sections of the course. To

ensure that their standard errors did not overstate the precision of their estimates when using the individual student data, Fleisher, Hashimoto, and Weinberg explicitly adjusted their standard errors for the clustering of the individual student observations into classes using a procedure akin to that developed by Moulton (1986).[vii]

Whatever the unit of measure for the dependent variable (aggregate or individual) the important point here is recognition of the need for one of two adjustments that must be made to get the correct standard errors. If an aggregate unit is employed (e.g., class means) then an adjustment for the number of observations making up the aggregate is required. If individual observations share a common component (e.g., students grouped into classes) then the standard errors reflect this clustering. Computer programs such as LIMDEP (NLOGIT), SAS and STATA can automatically perform both of these adjustments.

ANALYSIS OF VARIANCE (ANOVA) AND HYPOTHESES TESTING

Student of statistics are familiar with the *F* statistic as computed and printed in most computer regression routines under a banner "Analysis of Variance" or just ANOVA. This *F* is often presented in introductory statistics textbooks as a test of the overall all fit or explanatory power of the regression. I have learned from years of teaching econometrics that it is better to think of this test as one of all population model slope coefficients are zero (the explanatory power is not sufficient to conclude that there is any relations between the *x*s and *y* in the population) versus the alternative that at least one slope coefficient is not zero (there is some explanatory power). Thinking of this *F* statistic as just a joint test of slope coefficients, makes it easier to recognize that an *F* statistics can be calculated for any subset of coefficients to test for joint significance within the subset. Here I present the theoretical underpinnings for extensions of the basic ANOVA to tests of subsets of coefficients. Parts two three and four provide the corresponding commands to do these tests in LIMDEP, STATA and SAS.

As a starting point to ANOVA consider the *F* statistics that is generated by most computer programs. This *F* calculation can be viewed as a decomposition or partitioning of the dependent variable into two components (intercept and slopes) and a residual:

$$\mathbf{y} = \mathbf{i}b_1 + \mathbf{X}_2\mathbf{b}_2 + \mathbf{e}$$

where $\mathbf{i}$ is the column of 1's in the X matrix associated with the intercept b_1 and X_2 is the remaining *(k–1)* explanatory *x* variables associated with the *(k–1)* slope coefficients in the $\mathbf{b}_2$ vector. The total sum of squared deviations

$$\text{TotSS} = \sum_{i=1}^{n}(y_i - \bar{y})^2 = \sum_{i=1}^{n}(y_i)^2 - n\bar{y}^2 = (\mathbf{y}'\mathbf{y} - n\bar{y}^2)$$

measures the amount of variability in *y* around $\bar{y}$, which ignoring any effect of the *x*s (in essence the b_2 vector is assumed to be a vector of zeros). The residual sum of squares

$$\text{ResSS} = \sum_{i=1}^{n}(e_i)^2 = \mathbf{e}'\mathbf{e}$$

measures the amount of variability in y around $\hat{y}$, which lets b_1 and b_2 assume their least squares values.

Partitioning of y in this manner enables us to test the contributions of the xs to explaining variability in the dependent variable. That is,

$$H_0 : \beta_2 = \beta_3 = ...\beta_k = 0 \text{ versus } H_A : \text{at least one slope coefficient is not zero.}$$

For calculating the F statistic, computer programs use the equivalent of the following:

$$F = \frac{[(\mathbf{y'y} - n\bar{y}^2) - \mathbf{e'e}]/[(n-1)-(n-K)]}{\mathbf{e'e}/(n-K)} = \frac{[(\mathbf{y'y} - n\bar{y}^2) - \mathbf{e'e}]/(K-1)}{\mathbf{e'e}/(n-K)} = \frac{(\text{TotSS} - \text{ResSS})/(K-1)}{\text{ResSS}/(n-K)}$$

This F is the ratio of two independently distributed Chi-square random variables adjusted for their respective degrees of freedom. The relevant decision rule for rejecting the null hypothesis is that the probability of this calculated F value or something greater, with $K - 1$ and $n - K$ degrees of freedom, is less than the typical (0.10, 0.05 or 0.01) probabilities of a Type I error.

Calculation of the F statistic in this manner, however, is just a special case of running two regressions: a restricted and an unrestricted. One regression was computed with all the slope coefficients set equal (or restricted) to zero so Y is regressed only on the column of ones. This **restricted regression** is the same as using $\bar{Y}$ to predict Y regardless of the values of the xs. This **restricted residual sum of squares**, $\mathbf{e}_r'\mathbf{e}_r$, is what is usually called the **total sum of squares**, TotSS = $\mathbf{y'y} - n\bar{y}^2$. The unrestricted regression allows all of the slope coefficients to find their values to minimize the residual sum of squares, which is thus called the **unrestricted residual sum of squares,** $\mathbf{e}_u'\mathbf{e}_u$, and is usually just list in a computer printout as the residual sum of squares ResSS= $\mathbf{e'e}$.

The idea of a restricted and unrestricted regression can be extended to test any subset of coefficients. For example, say the full model for a posttest Y is

$$Y_i = \beta_1 + \beta_2 x_{i2} + \beta_3 x_{i3} + \beta_4 x_{i4} + \varepsilon_i .$$

Let's say the claim is made that x_3 and x_4 do not affect Y. One way to interpret this is to specify that $\beta_3 = \beta_4 = 0$, but $\beta_2 \neq 0$. The dependent variable is again decomposed into two components but now x_1 is included with the intercept in the partitioning of the $\mathbf{X}$ matrix:

$$\mathbf{y} = \mathbf{X_1 b_1} + \mathbf{X_2 b_2} + \mathbf{e} .$$

where $\mathbf{X_1}$ is the $n \times 2$ matrix, with the first column containing ones and the second observations on x_1 ($\mathbf{b_1}$ contains the y intercept and x_1 slope coefficient) and $\mathbf{X_2}$ is the $n \times 2$ matrix, with two columns for x_3 and x_4 ($\mathbf{b_2}$ contains x_3 and x_4 slope coefficients). If the claim about x_3 and x_4 not

belonging in the explanation of Y is true, then the two slope coefficients in $\mathbf{b_2}$ should be set to zero because the true model is the restricted specification

$$Y_i = \beta_1 + \beta_2 x_{i2} + \varepsilon_i .$$

The null hypotheses is $H_0 : \beta_3 = \beta_4 = 0$; i.e., x_2 might affect Y but x_3 and x_4 do not affect Y.

The alternative hypothesis is $H_A : \beta_3 \neq 0$ or $\beta_4 \neq 0$; i.e.,. x_3 and x_4 both affect Y.

The F statistic to test the hypotheses is then

$$F = \frac{[\mathbf{e}_\mathbf{r}'\mathbf{e}_\mathbf{r} - \mathbf{e}_u'\mathbf{e}_u]/[(n-K_r)-(n-K_u)]}{\mathbf{e}_u'\mathbf{e}_u/(n-K_u)} ,$$

where the restricted residual sum of squares $\mathbf{e}_\mathbf{r}'\mathbf{e}_\mathbf{r}$ is obtained from a simple regression of Y on x_2, including a constant, and the unrestricted sum of squared residuals $\mathbf{e}_u'\mathbf{e}_u$ is obtained from a regression of Y on x_2, x_3 and x_4, including a constant.

In general, it is best to test the overall fit of the regression model before testing any subset or individual coefficients. The appropriate hypotheses and F statistic are

$$H_0 : \beta_2 = \beta_3 = ... = \beta_K = 0 \quad (\text{or } H_0 : R^2 = 0)$$
$$H_A : \text{at least one slope coefficient is not zero} \quad (\text{or } H_0 : R^2 \neq 0)$$

$$F = \frac{[(\mathbf{y}'\mathbf{y} - n\bar{y}^2) - \mathbf{e}'\mathbf{e}]/(K-1)}{\mathbf{e}'\mathbf{e}/(n-K)} .$$

If the calculated value of this F is significant, then subsets of the coefficients can be tested as

$$H_0 : \beta_s = \beta_t = ... = 0$$
$$H_A : \text{at least one of these slope coefficient is not zero}$$

$$F = \frac{[\mathbf{e}_\mathbf{r}'\mathbf{e}_\mathbf{r} - \mathbf{e}_u'\mathbf{e}_u]/[(K_u - q)]}{\mathbf{e}_u'\mathbf{e}_u/(n-K_u)} , \text{ for } q = k \text{ – number of restrictions.}$$

The restricted residual sum of squares $\mathbf{e}_\mathbf{r}'\mathbf{e}_\mathbf{r}$ is obtained by a regression on only the q xs that did not have their coefficients restricted to zero. Any number of subsets of coefficients can be tested in this framework of restricted and unrestricted regressions as summarized in the following table.

SUMMARY FOR ANOVA TESTING

PANEL A. TRADITIONAL ANOVA FOR TESTING
$R^2 = 0$ versus $R^2 \neq 0$

Sum of Squares	Source	Degrees of Freedom	Mean Square
Total (to be explained)	$\mathbf{y'y} - n\bar{y}^2$	$n - 1$	s_y^2
Residual or Error (unexplained)	$\mathbf{e'e}$	$n - k$	s_e^2
Regression or Model (explained)	$\mathbf{b'X'y} - n\bar{y}^2$	$k - 1$	

$$F = \frac{R^2/(K-1)}{(1-R^2)/(n-K)} = \frac{[1-(\text{ResSS/TotSS})]/(K-1)}{(\text{ResSS/TotSS})/(n-K)} = \frac{(\text{TotSS-ResSS})/(K-1)}{\text{ResSS}/(n-K)}$$

PANEL B. RESTRICTED REGRESSION FOR TESTING ALL THE SLOPES $\beta_2 = \beta_3 = ... = \beta_K = 0$

Sum of Squares	Source	Degrees of Freedom	Mean Square
Restricted (all slopes = 0)	$\mathbf{e}_\mathbf{r}'\mathbf{e}_\mathbf{r} = \mathbf{y'y} - n\bar{y}^2$	$n - 1$	s_y^2
Unrestricted	$\mathbf{e}_u'\mathbf{e}_u = \mathbf{e'e}$	$n - k$	s_e^2
Improvement	$\mathbf{b'X'y} - n\bar{y}^2$	$k - 1$	

$$F = \frac{[\text{Restricted ResSS}(slopes = 0) - \text{Unrestricted ResSS}](K-1)}{\text{Unrestricted ResSS}/(n-k)}$$

PANEL C. RESTRICTED REGRESSION FOR TESTING A SUBSET OF COEFFICIENTS $\beta_s = \beta_t = ... = 0$

Sum of Squares	Source	Degrees of Freedom
Restricted ($\beta_s = \beta_t = ... = 0$)	$\mathbf{e}_\mathbf{r}'\mathbf{e}_\mathbf{r}$	$n - q$, for $q =$ k – number of restrictions
Unrestricted	$\mathbf{e}_u'\mathbf{e}_u$	$n - k$
Improvement	$\mathbf{e}_\mathbf{r}'\mathbf{e}_\mathbf{r} - \mathbf{e}_u'\mathbf{e}_u$	$K - q$

$$F = \frac{[\text{Restricted ResSS}(subset = 0) - \text{Unrestricted ResSS}](K-q)}{\text{Unrestricted ResSS}/(n-k)}$$

The *F* test of subsets of coefficients is ideal for testing interactions. For instance, to test for the treatment effect in the following model both β_4 and β_5 must be jointly tested against zero:

$$ChangeScore = \beta_1 + \beta_2 female + \beta_3 femaletreatment + \beta_4 treatment + \beta_5 GPA + \varepsilon$$

$$H_o : \beta_4 = \beta_5 = 0 \qquad H_A : \beta_4 \text{ or } \beta_5 \neq 0$$

where "*ChangeScore*" is the difference between a student's test scores at the end and beginning of a course in economics, *female* = 1, if female and 0 if male, "*treatment*" = 1, if in the treatment group and 0 if not, and "*GPA*" is the student's grade point average before enrolling in the course.

The *F* test of subsets of coefficients is also ideal for testing for fixed effects as reflected in sets of dummy variables. For example, in Parts Two, Three and Four an *F* test is performed to check whether there is any fixed difference in test performance among four classes taking economics using the following assumed data generating process:

$$post = \beta_1 + \beta_2 pre + \beta_3 class1 + \beta_4 class2 + \beta_5 class3 + \varepsilon$$

$$H_o : \beta_3 = \beta_4 = \beta_5 = 0 \qquad H_A : \beta_3, \beta_4 \text{ or } \beta_5 \neq 0$$

where "post" is a student's post-course test score, "pre" is the student's pre-course test score, and "class" identifies to which one of the four classes the students was assigned, e.g., class3 = 1 if student was in the third class and class3 = 0 if not. The fixed effect for students in the fourth class (*class*1, *class*2 and *class*3 are zero)) is captured in the intercept β_1.

It is important to notices in this test of fixed class effects that the relationship between the *post* and *pre* test (as reflected in the slope coefficient β_2) is assumed to be the same regardless of the class to which the student was assigned. The next section described a test for any structural difference among the groups.

TESTING FOR A SPECIFICATION DIFFERENCE ACROSS GROUPS

Earlier in our discussion of the difference in difference or change score model, a 0-1 bivariate dummy variable was introduced to test for a difference in intercepts between a treatment and control group, which could be done with a single coefficient *t* test. However, the expected difference in the dependent variable for the two groups might not be constant. It might vary with the level of the independent variables. Indeed, the appropriate model might be completely different for the two groups. Or, it might be the same.

Allowing for any type of difference between the control and experimental variables implies that the null and alternative hypotheses are

H_0: $\boldsymbol{\beta}_1 = \boldsymbol{\beta}_2 = \beta$
H_A: $\boldsymbol{\beta}_1 \neq \boldsymbol{\beta}_2$,

where the $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ are KX1 column vectors containing the K coefficients $\beta_1, \beta_2, \beta_3, \ldots \beta_K$ for the control $\boldsymbol{\beta}_1$ and the experimental $\boldsymbol{\beta}_2$ groups . Let $\mathbf{X}_1$ and $\mathbf{X}_2$ contain the observations on the explanatory variables corresponding to the $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$, including the column of ones for the constant β_1. The unrestricted regression is captured by two separate regressions:

$$\begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{X}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{X}_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta}_1 \\ \boldsymbol{\beta}_2 \end{bmatrix} + \begin{bmatrix} \boldsymbol{\varepsilon}_1 \\ \boldsymbol{\varepsilon}_2 \end{bmatrix}.$$

That is, the unrestricted model is estimated by fitting the two regressions separately. The unrestricted residual sum of squares is obtained by adding the residuals from these two regressions. The unrestricted degrees of freedom are similarly obtained by adding the degrees of freedom of each regression.

The restricted regression is just a regression of y on the xs with no group distinction in beta coefficients:

$$\begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{X}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{X}_2 \end{bmatrix} [\boldsymbol{\beta}] + \begin{bmatrix} \boldsymbol{\varepsilon}_1 \\ \boldsymbol{\varepsilon}_2 \end{bmatrix}.$$

That is , the restricted residual sum of squares is obtained from a regression in which the data from the two groups are pooled and a single set of coefficients is estimated for the pooled data set.

The appropriate F statistic is

$$F = \frac{[\text{Restricted ResSS}(\boldsymbol{\beta}_1 = \boldsymbol{\beta}_2) - \text{Unrestricted ResSS}]/K}{\text{Unrestricted ResSS}/[n - 2K]},$$

where unrestricted ResSS = residuals sum of squares from a regression on only those in the control plus residuals from a regression on only those in the treatment groups.

Thus, to test for structure change over J regimes, run separate regressions on each and add up the residuals to obtain the unrestricted residual sum of squares, ResSS_u,with df = $n - JK$. The restricted residual sum of squares is ResSS_r, with df = $n - K$.

$$H_0 : \boldsymbol{\beta_1} = \boldsymbol{\beta_2} = \ldots = \boldsymbol{\beta_J} \text{ and } H_a : \boldsymbol{\beta}'s \text{ are not equal}$$

$$F = \frac{(\text{ResSS}_\text{r} - \text{ResSS}_\text{u}) / K(J-1)}{\text{ResSS}_\text{u} / (n - JK)}$$

This form of testing for a difference among groups is known in economics as a Chow Test. As demonstrated in Part Two using LIMDEP and Parts Three and Four using STATA and SAS, any number of subgroups could be tested by adding up their individual residual sums of squares and degrees of freedom to form the unrestricted residual sums of squares and matching degrees of freedom.

OTHER TEST STATISTICS

Depending on the nature of the model being estimated and the estimation method, computer programs will produce alternatives to the *F* statistics for testing (linear and nonlinear) restrictions and structural changes. What follows is only an introduction to these statistics that should be sufficient to give meaning to the numbers produced based on our discussion of ANOVA above.

The **Wald (*W*) statistic** follows the Chi-squared distribution with *J* degrees of freedom, reflecting the number of restrictions imposed:

$$W = \frac{(\mathbf{e}_r'\mathbf{e}_r - \mathbf{e}_u'\mathbf{e}_u)}{\mathbf{e}_u'\mathbf{e}_u / n} \sim \chi^2(J) \ .$$

If the model and the restriction are linear, then

$$W = \frac{nJ}{n-k} F = \frac{J}{1-(k/n)} F,$$

which for large *n* yields the asymptotic results

$$W = JF \ .$$

The **likelihood ratio (LR) test** is formed by twice the difference between the log-likelihood function for an unrestricted regression (L_{ur}) and its value for the restricted regression (L_r).

$$LR = 2(L_{ur} - L_r) \geq 0 \ .$$

Under the null hypothesis that the J restrictions are true, LR is distributed Chi-square with *J* degrees of freedom.

The relationship between the likelihood ratio test and Wald test can be shown to be

$$LR = \frac{n(\mathbf{e}_r'\mathbf{e}_r - \mathbf{e}_u'\mathbf{e}_u)}{\mathbf{e}_u'\mathbf{e}_u} - \frac{n(\mathbf{e}_r'\mathbf{e}_r - \mathbf{e}_u'\mathbf{e}_u)^2}{2\mathbf{e}_u'\mathbf{e}_u} \leq W .$$

The **Lagrange multiplier test (LM)** is based on the gradient (or score) vector

$$\begin{bmatrix} \partial L/\partial \beta \\ \partial L/\partial \sigma^2 \end{bmatrix} = \begin{bmatrix} \mathbf{X}'\boldsymbol{\varepsilon}/\sigma^2 \\ -(n/2\sigma^2) + (\boldsymbol{\varepsilon}'\boldsymbol{\varepsilon}/2\sigma^4) \end{bmatrix}.$$

where, as before, to evaluate this score vector with the restrictions we replace $\mathbf{e} = \mathbf{y} - \mathbf{Xb}$ with $\mathbf{e_r} = \mathbf{y} - \mathbf{Xb_r}$. After sufficient algebra, the Lagrange statistic is defined by

$$LM = n\mathbf{e}_r'\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{e}_r/\mathbf{e}_r'\mathbf{e}_r = nR^2 \sim \chi^2(J) ,$$

where R^2 is the conventional coefficient of determination from a regression of $\mathbf{e}_r$ on $\mathbf{X}$, where e_r has a zero mean (i.e., only slopes are being tested). It can also be shown that

$$LM = \frac{nJ}{(n-k)[1 + JF/(n-k)]} F = \frac{W}{1 + (W/n)} .$$

Thus, $LM \leq LR \leq W$.

DATA ENTRY AND ESTIMATION

I like to say to students in my classes on econometrics that theory is easy, data are hard – hard to find and hard to get into a computer program for statistical analysis. In this first of four parts in Module One, I provided an introduction to the theoretical data generating processes associated with continuous versus discrete dependent variables. Parts Two, Three and Four concentrate on getting the data into one of three computer programs: LIMDEP (NLOGIT), STATA and SAS. Attention is also given to estimation and testing within regressions employing individual cross-sectional observations within these programs. Later modules will address complications introduced by panel data and sources of endogeneity.

REFERENCES

Allison, Paul D. (1990). "Change Scores as Dependent Variables in Regression Analysis," *Sociological Methodology*, Vol. 20: 93-114.

Bandiera, Oriana, Valentino Larcinese and Imron Rasul (2010). "Blissful Ignorance? Evidence from a Natural Experiment on the Effect of Individual Feedback on Performance , " IZA Seminar, Bonn Germany, December 5, 2009. January 2010 version downloadable at http://www.iza.org/index_html?lang=en&mainframe=http%3A//www.iza.org/en/webcontent/events/izaseminar_description_html%3Fsem_id%3D1703&topSelect=events&subSelect=seminar

Becker, William E. (2004). "Quantitative Research on Teaching Methods in Tertiary Education," in W. E. Becker and M. L. Andrews (eds), *The Scholarship of Teaching and Learning in Higher Education: Contributions of the Research Universities*, Indiana University Press: 265-309.

Becker, William E. (Summer 1983). "Economic Education Research: Partr III, Statistical Estimation Methods," *Journal of Economic Education*, Vol. 14 (Sumer): 4-15

Becker, William E. and William H. Greene (2001). "Teaching Statistics and Econometrics to Undergraduates," *Journal of Economic Perspectives*, Vol. 15 (Fall): 169-182.

Becker, William E., William Greene and Sherwin Rosen (1990). "Research on High School Economic Education," *American Economic Review*, Vol. 80, (May): 14-23, and an expanded version in *Journal of Economic Education*, Summer 1990: 231-253.

Becker, William E. and Peter Kennedy (1992). "A Graphical Exposition of the Ordered Probit," with P. Kennedy, *Econometric Theory*, Vol. 8: 127-131.

Becker, William E. and Michael Salemi (1977). "The Learning and Cost Effectiveness of AVT Supplemented Instruction: Specification of Learning Models," *Journal of Economic Education* Vol. 8 (Spring) : 77-92.

Becker, William E. and Donald Waldman (1989). "Graphical Interpretation of Probit Coefficients," *Journal of Economic Education*, Vol. 20 (Fall): 371-378.

Campbell, D., and D. Kenny (1999). *A Primer on Regression Artifacts*. New York: The Guilford Press.

Fleisher, B., M. Hashimoto, and B. Weinberg. 2002. "Foreign GTAs can be Effective Teachers of Economics." *Journal of Economic Education*, Vol. 33 (Fall): 299-326.

Francisco, J. S., M. Trautmann, and G. Nicoll. 1998. "Integrating a Study Skills Workshop and Pre-Examination to Improve Student's Chemistry Performance." *Journal of College Science Teaching*, Vol. 28 (February): 273-278.

Friedman, M. 1992. "Communication: Do Old Fallacies Ever Eie?" *Journal of Economic Literature,* Vol. 30 (December): 2129-2132.

Greene, William (2003). *Econometric Analysis*. 5th Edition, New Jersey: Prentice Hall.

Hake, R. R. (1998). "Interactive-Engagement versus Traditional Methods: A Six-Thousand-Student Survey of Mechanics Test Data for Introductory Physics Courses." *American Journal of Physics*, Vol. 66 (January): 64-74.

Hanushek, Eric A. (1986). "The Economics of Schooling: Production and Efficiency in Public Schools," *Journal of Economic Literature*, 24(September)): 1141-1177.

Lo, Melody, Sunny Wong and Franklin Mixon (2008). "Ranking Economics Journals Economics Departments, and Economists Using Teaching-Focused Research Productivity." *Southern Economics Journal* 2008, 74(January): 894-906.

Moulton, B. R. (1986). "Random Group Effects and the Precision of Regression Estimators." *Journal of Econometrics*, Vol. 32 (August): 385-97.

Kennedy, P., and J. Siegfried. (1997). "Class Size and Achievement in Introductory Economics: Evidence from the TUCE III Data." *Economics of Education Review*, Vol. 16 (August): 385-394.

Kvam, Paul. (2000). "The Effect of Active Learning Methods on Student Retention in Engineering Statistics." *American Statistician*, 54 (2): 136-40.

Ramsden, P. (1998). "Managing the Effective University." *Higher Education Research & Development*, 17 (3): 347-70.

Salemi, Michael and George Tauchen. 1987. "Simultaneous Nonlinear Learning Models." In W. E. Becker and W. Walstad, eds., *Econometric modeling in economic education research*, pp. 207-23. Boston: Kluwer-Nijhoff.

Spector, Lee C. and Michael Mazzeo (1980). "Probit Analysis and Economic Education" *Journal of Economic Education,* Vol. 11(Spring), 11(2): 37-44.

Wainer, H. 2000. "Kelley's Paradox." *Chance*, 13 (Winter): 47-48.

ENDNOTES

[i] Let the change or gain score be $\Delta y = [y_1 - y_0]$, which is the posttest score minus the pretest score, and let the maximum change score be $\Delta y_{\max} = [y_{\max} - y_0]$, then

$$\frac{\partial(\Delta y / \Delta y_{\max})}{\partial y_0} = \frac{-(y_{\max} - y_1)}{(y_{\max} - y_0)^2} \leq 0, \text{for } y_{\max} \geq y_1 \geq y_0$$

[ii] Let the posttest score (y_1) and pretest score (y_0) be defined on the same scale, then the model of the i^{th} student's pretest is

$$y_{0i} = \beta_0 (ability)_i + v_{0i},$$

where β_0 is the slope coefficient to be estimated, v_{0i} is the population error in predicting the i^{th} student's pretest score with ability, and all variables are measured as deviations from their means. The i^{th} student's posttest is similarly defined by

$$y_{1i} = \beta_1 (ability)_i + v_{1i}$$

The change or gain score model is then

$$y_{1i} - y_{0i} = (\beta_1 - \beta_0) ability + v_{1i} - v_{0i}$$

And after substituting the pretest for unobserved true ability we have

$$\Delta y_i = (\Delta\beta / \beta_0) y_{0i} + v_{1i} - v_{0i}[1 + (\Delta\beta / \beta_0)]$$

The least squares slope estimator $(\Delta b / b_0)$ has an expected value of

$$E(\Delta b / b_0) = E(\sum_i \Delta y_1 y_{0i} / \sum_i y_{0i}^2)$$
$$E(\Delta b / b_0) = (\Delta\beta / \beta_0) + E\{\sum_i [v_{1i} - v_{0i} - v_{01}(\Delta\beta / \beta_0)]_{1i} y_{0i} / \sum_i y_{0i}^2\}$$
$$E(\Delta b / b_0) \leq (\Delta\beta / \beta_0)$$

Although v_{1i} and y_{0i} are unrelated, $E(v_{1i}\ y_{0i}) = 0$, v_{0i} and y_{0i} are positively related, $E(v_{0i}\ y_{0i}) > 0$; thus, $E(\Delta b / b_0) \leq \Delta\beta / \beta_0$. Becker and Salemi (1977) suggested an instrumental variable technique to address this source of bias and Salemi and Tauchen (1987) suggested a modeling of the error term structure.

Hake (1998) makes no reference to this bias when he discusses his regressions and correlation of average normalized gain, average gain score and posttest score on the average pretest score. In

http://www.consecol.org/vol5/iss2/art28/, he continued to be unaware of, unable or unwilling to specify the mathematics of the population model from which student data are believed to be generated and the method of parameter estimation employed. As the algebra of this endnote suggests, if a negative relationship is expected between the gap closing measure

$$g = (posttest - pretest)/(maxscore - pretest)$$

and the pretest, but a least-squares estimator does not yield a significant negative relationship for sample data, then there is evidence that something is peculiar. It is the lack of independence between the pretest and the population error term (caused, for example, by measurement error in the pretest, simultaneity between *g* and the pretest, or possible missing but relevant variables) that is the problem. Hotelling received credit for recognizing this endogenous regressor problem (in the 1930s) and the resulting regression to the mean phenomenon. Milton Friedman received a Nobel prize in economics for coming up with an instrumental variable technique (for estimation of consumption functions in the 1950s) to remove the resulting bias inherent in least-squares estimators when measurement error in a regressor is suspected. Later Friedman (1992, p. 2131) concluded: "I suspect that the regression fallacy is the most common fallacy in the statistical analysis of economic data ..." Similarly, psychologists Campbell and Kenny (1999, p. xiii) stated: "Regression toward the mean is a artifact that as easily fools statistical experts as lay people." But unlike Friedman, Campbell and Kenny did not recognize the instrumental variable method for addressing the problem.

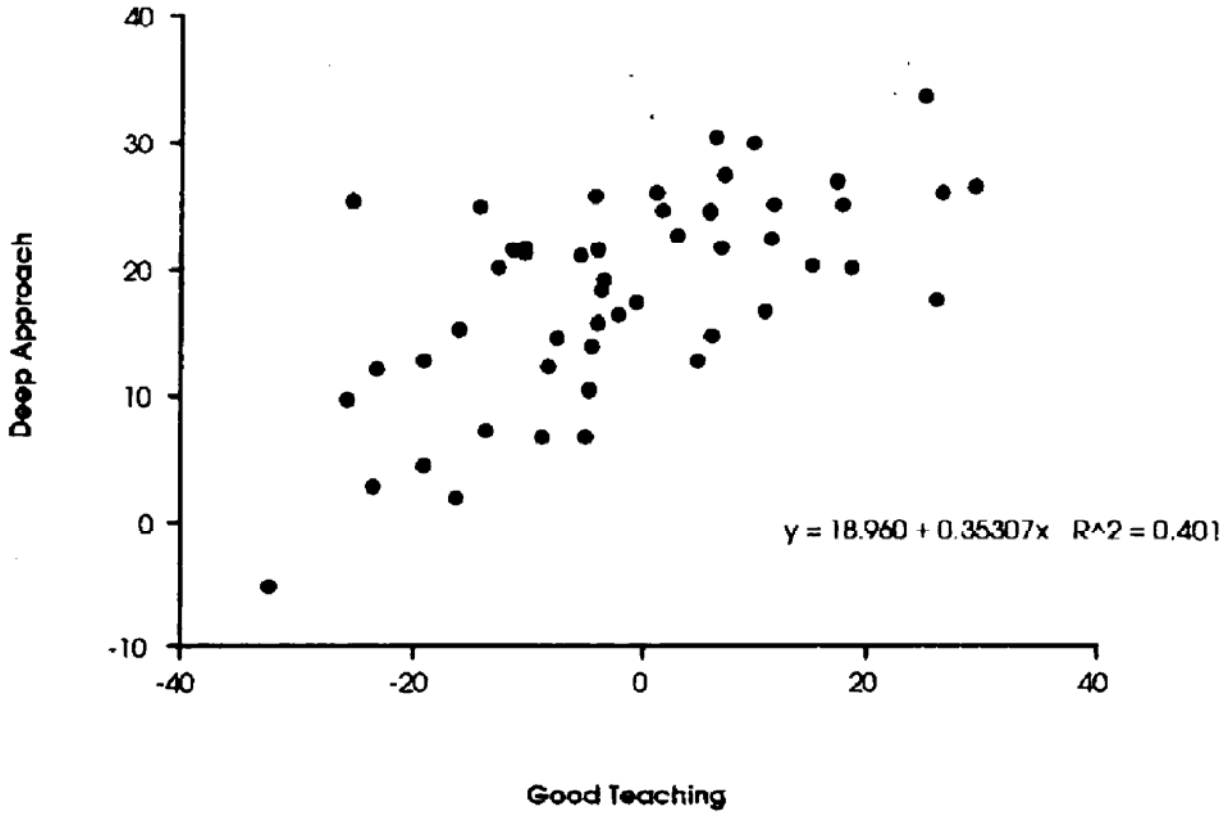
In an otherwise innovative study, Paul Kvam (2000) correctly concluded that there was insufficient statistical evidence to conclude that active-learning methods (primarily through integrating students' projects into lectures) resulted in better retention of quantitative skills than traditional methods, but then went out on a limb by concluding from a scatter plot of individual student pretest and posttest scores that students who fared worse on the first exam retain concepts better if they were taught using active-learning methods. Kvan never addressed the measurement error problem inherent in using the pretest as an explanatory variable. Wainer (2000) called attention to others who fail to take measurement error into account in labeling students as "strivers" because their observed test scores exceed values predicted by a regression equation.

[iii] The plot for the probability model was produced by first fitting a probit model of the binary variable GRADE, as a function of GPA. This produces a functional relationship of the form $\text{Prob}(\text{GRADE} = 1) = \Phi(\alpha + \beta\text{GRADE})$, where estimates of α and β are produced by maximum likelihood techniques. The graph is produced by plotting the standard normal distribution function, $\Phi(\alpha + \beta\text{GRADE})$ for the values of GRADE in the sample, which range between 2.0 and 4.0, then connecting the dots. The nonparametric regression, although intuitively appealing because it can be viewed as making use of weighted relative frequencies, is computationally more complicated. [Today the binomial probit model can be fitted with just about any statistical package but software for nonparametric estimation is less common. LIMDEP (NLOGIT) version 8.0 (Econometric Software, Inc., 2001) was used for both the probit and nonparametric estimations.] The nonparametric approach is based on the assumption that there is some as yet

unknown functional relationship between the Prob(GRADE = 1) and the independent variable, GPA, say Prob(Grade = 1 | GPA) = F(GPA). The probit model based on the normal distribution is one functional candidate, but the normality assumption is more specific than we need at this point. We proceed to use the data to find an approximation to this function. The form of the 'estimator' of this function is $F(\text{GPA}^*) = \Sigma_{\text{i = all observations}}\, w(\text{GPA}^* - \text{GPA}_\text{i})\text{GRADE}_\text{i}$. The weights, '$w$(.),' are positive weight functions that sum to 1.0, so for any specific value GPA*, the approximation is a weighted average of the values of GRADE. The weights in the function are based on the desired value of GPA, that is GPA*, as well as all the data. The nature of the computation is such that if there is a positive relationship between GPA and GRADE =1, then as GPA* gets larger, the larger weights in the average shown above will tend to be associated with the larger values of GRADE. (Because GRADE is zeros and ones, this means that for larger values of GPA*, the weights associated with the observations on GRADE that equal one will generally be larger than those associated with the zeros.) The specific form of these weights is as follows: $w(\text{GPA}^* - \text{GPA}_\text{i}) = (1/A)\times(1/h)K[(\text{GPA}^* - \text{GPA}_\text{i})/h]$. The '$h$' is called the smoothing parameter, or bandwidth, K[.] is the 'kernel density function' and A is the sum of the functions, ensuring that the entire expression sums to one. Discussion of nonparametric regression using a kernel density estimator is given in Greene (2003, pp. 706-708). The nonparametric regression of GRADE on GPA plotted in the figure was produced using a logistic distribution as the kernel function and the following computation of the bandwidth: let r equal one third of the sample range of GPA and let s equal the sample standard deviation of GPA. The bandwidth is then $h = .9\times\text{Min}(r,s)/n^{1/5}$. (In spite of their apparent technical cache, bandwidths are found largely by experimentation. There is no general rule that dictates what one should use in a particular case, which is unfortunate because the shapes of kernel density plots are heavily dependent upon them.)

[iv] Unlike the mean, the median reflects relative but not absolute magnitude; thus, the median may be a poor measure of change. For example, the series 1, 2, 3 and the series 1, 2, 300 have the same median (2) but different means (2 versus 101).

[v] To appreciate the importance of the unit of analysis, consider a study done by Ramsden (1998, pp. 352-354) in which he provided a scatter plot showing a positive relationship between a y-axis index for his "deep approach" (aimed at student understanding versus "surface learning") and an x-axis index of "good teaching" (including feedback of assessed work, clear goals, etc.):



Ramsden's regression ($y = 18.960 + 0.35307x$) seems to imply that a decrease (increase) in the good teaching index by one unit leads to a 0.35307 decrease (increase) in the predicted deep approach index; that is, good teaching positively affects deep learning. But does it?

Ramsden (1998) ignored the fact that each of his 50 data points represent a type of institutional average that is based on multiple inputs; thus, questions of heteroscedasticity and the calculation of appropriate standard errors for testing statistical inference are relevant. In addition, because Ramsden reports working only with the aggregate data from each university, it is possible that within each university the relationship between good teaching (x) and the deep approach (y) could be negative but yet appear positive in the aggregate.

When I contacted Ramsden to get a copy of his data and his coauthored "Paper presented at the Annual Conference of the Australian Association for Research in Education, Brisbane (December 1997)," which was listed as the source for his regression of the deep approach index on the good teaching index in his 1998 published article, he confessed that this conference paper never got written and that he no longer had ready access to the data (email correspondence August 22, 2000).

Aside from the murky issue of Ramsden citing his 1997 paper, which he subsequently admitted does not exist, and his not providing the data on which the published 1998 paper is allegedly based, a potential problem of working with data aggregated at the university level can be seen

with three hypothetical data sets. The three regressions for each of the following hypothetical universities show a negative relationship for *y* (deep approach) and *x* (good teaching), with slope coefficients of –0.4516, –0.0297, and –0.4664, but a regression on the university means shows a positive relationship, with slope coefficient of +0.1848. This is a demonstration of "Simpson's paradox," where aggregate results are different from dissaggregated results.

University One

$\hat{y}(1) = 21.3881 - 0.4516x(1)$ Std. Error = 2.8622 $R^2 = 0.81$ n = 4
y(1): 21.8 15.86 26.25 14.72
x(1): -4.11 6.82 -5.12 17.74

University Two

$\hat{y}(2) = 17.4847 - 0.0297x(2)$ Std. Error = 2.8341 $R^2 = 0.01$ n = 8
y(2): 12.60 17.90 19.00 16.45 21.96 17.1 18.61 17.85
x(2): -10.54 -10.53 -5.57 -11.54 -15.96 -2.1 -9.64 12.25

University Three

$\hat{y}(3) = 17.1663 - 0.4664x(3)$ Std. Error = 2.4286 $R^2 = 0.91$ n = 12
y(3): 27.10 2.02 16.81 15.42 8.84 22.90 12.77 17.52 23.20 22.60 25.90
x(3): -23.16 26.63 5.86 9.75 11.19 –14.29 11.51 –0.63 –19.21 –4.89 –16.16

University Means

$\hat{y}(means) = 18.6105 + 0.1848x(means)$ Std. Error = 0.7973 $R^2 = 0.75$ n = 3
y(means): 19.658 17.684 17.735
x(means): 3.833 -6.704 -1.218

[vi] Let y_{it} be the observed test score index of the i^{th} student in the t^{th} class, who has an expected test score index value of μ_{it}. That is, $y_{it} = \mu_{it} + \varepsilon_{it}$, where ε_{it} is the random error in testing such that its expected value is zero, $E(\varepsilon_{it}) = 0$, and variance is σ^2, $E(\varepsilon_{it}^2) = \sigma^2$, for all i and t. Let $\bar{y}_t$ be the sample mean of a test score index for the t^{th} class of n_t students. That is, $\bar{y}_t = \bar{\mu}_t + \bar{\varepsilon}_t$ and $E(\bar{\varepsilon}_t^2) = \sigma^2 / n_t$. Thus, the variance of the class mean test score index is inversely related to class size.

[vii] As in Fleisher, Hashimoto, and Weinberg (2002), let y_{gi} be the performance measure of the i^{th} student in a class taught by instructor g, let F_g be a dummy variable reflecting a characteristics of the instructor (e.g., nonnative English speaker), let x_{gi} be a $(1 \times n)$ vector of the student's

observable attributes, and let the random error associated with the i^{th} student taught by the g^{th} instructor be ε_{gi} . The performance of the i^{th} student is then generated by

$$y_{gi} = F_g\gamma + x_{gi}\beta + \varepsilon_{gi}$$

where γ and β are parameters to be estimated. The error term, however, has two components: one unique to the i^{th} student in the g^{th} instructor's class (u_{gi}) and one that is shared by all students in this class (ξ_g): $\varepsilon_{gi} = \xi_g + u_{gi}$. It is the presence of the shared error ξ_g for which an adjustment in standard errors is required. The ordinary least squares routines employed by the standard computer programs are based on a model in which the variance-covariance matrix of error terms is diagonal, with element σ_u^2 . The presence of the ξ_g terms makes this matrix block diagonal, where each student in the g^{th} instructor's class has an off-diagonal element σ_ξ^2 .

In (May 11, 2008) email correspondence, Bill Greene called my attention to the fact that Moulton (1986) gave a specific functional form for the shared error term component computation. Fleisher, Hashimoto, and Weinberg actually used an approximation that is aligned with the White estimator (as presented in Parts Two, Three and Four of this module), which is the "CLUSTER" estimator in STATA. In LIMDEP (NLOGIT), Moulton's shared error term adjustment is done by first arranging the data as in a panel with the groups contained in contiguous blocks of observations. Then, the command is "REGRESS ; ... ; CLUSTER = spec. $" where "spec" is either a fixed number of observations in a group, or the name of an identification variable that contains a class number. The important point is to recognize that heterogeneity could be the result of each group having its own variance and each individual within a group having its own variance. As discussed in detail in Parts Two, Three and Four, heteroscedasticity in general is handled in STATA with the "ROBUST" command and in LIMDEP with the "HETRO" command.

MODULE ONE, PART TWO: READING DATA INTO LIMDEP, CREATING AND RECODING VARIABLES, AND ESTIMATING AND TESTING MODELS IN LIMDEP

This Part Two of Module One provides a cookbook-type demonstration of the steps required to read or import data into LIMDEP. The reading of both small and large text and Excel files are shown though real data examples. The procedures to recode and create variables within LIMDEP are demonstrated. Commands for least-squares regression estimation and maximum likelihood estimation of probit and logit models are provided. Consideration is given to analysis of variance and the testing of linear restrictions and structural differences, as outlined in Part One. (Parts Three and Four provide the STATA and SAS commands for the same operations undertaken here in Part Two with LIMDEP. For a thorough review of LIMDEP, see Hilbe, 2006.)

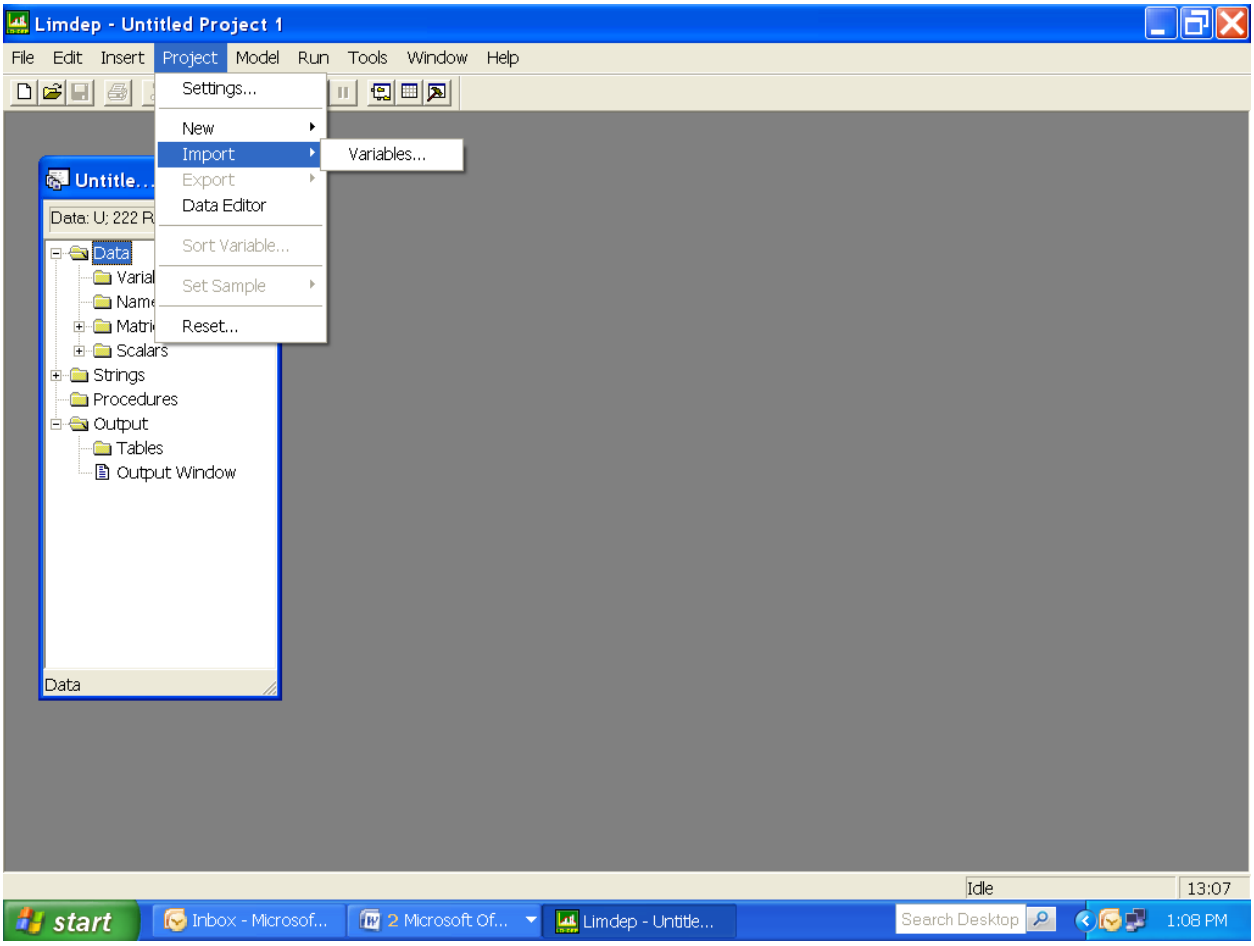
IMPORTING EXCEL FILES INTO LIMDEP

LIMDEP can read or import data in several ways. The most easily imported files are those created in Microsoft Excel with the ".xls" file name extension. To see how this is done, consider the data set in the Excel file "post-pre.xls," which consists of test scores for 24 students in four classes. The column titled "Student" identifies the 24 students by number, "post" provides each student's post-course test score, "pre" is each student's pre-course test score, and "class" identifies to which one of the four classes the students was assigned, e.g., class4 = 1 if student was in the fourth class and class4 = 0 if not. The "." in the post column for student 24 indicates that the student is missing a post-course test score.

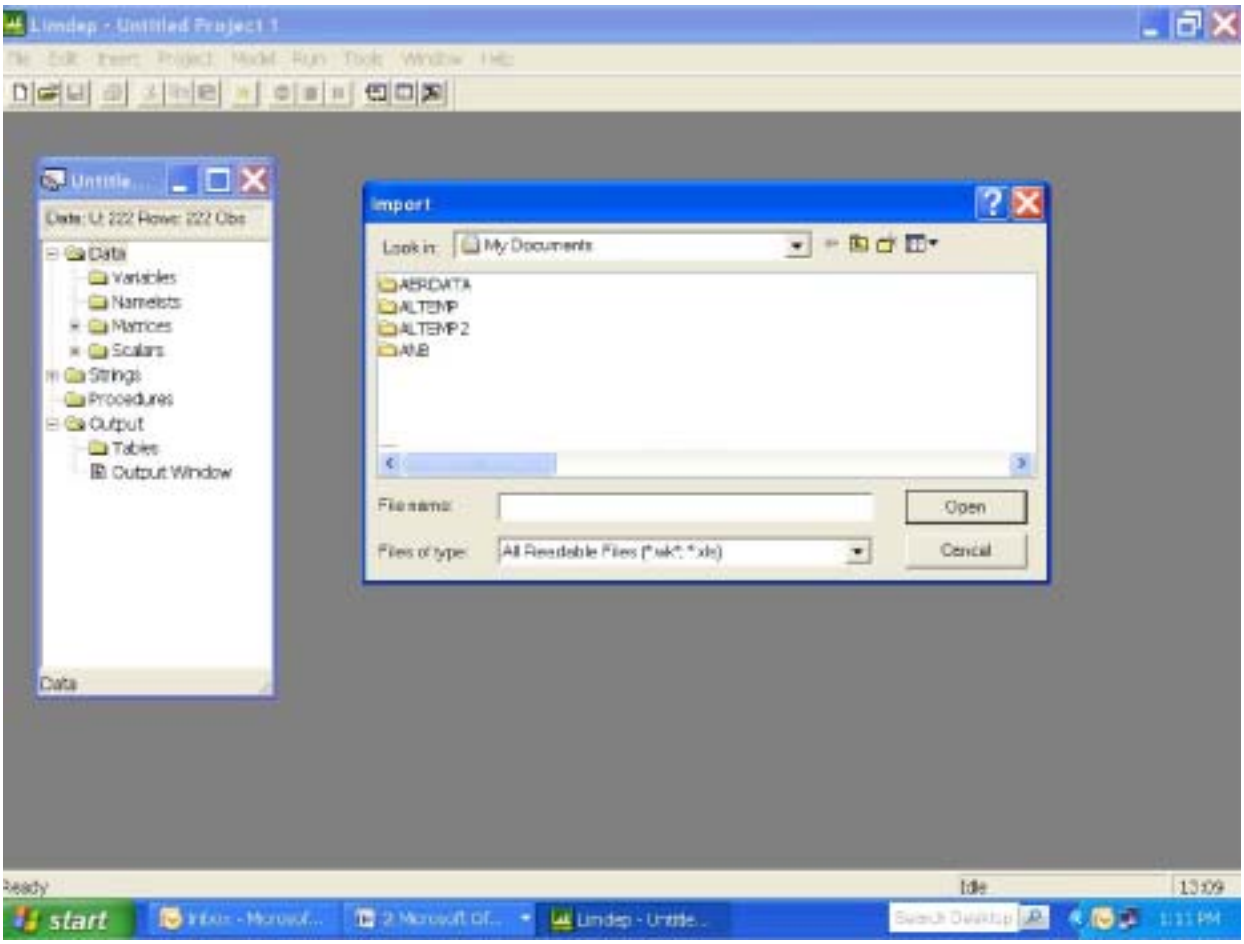
student	post	pre	class1	class2	class3	class4
1	31	22	1	0	0	0
2	30	21	1	0	0	0
3	33	23	1	0	0	0
4	31	22	1	0	0	0
5	25	18	1	0	0	0
6	32	24	0	1	0	0
7	32	23	0	1	0	0
8	30	20	0	1	0	0
9	31	22	0	1	0	0
10	23	17	0	1	0	0

11	22	16	0	1	0	0
12	21	15	0	1	0	0
13	30	19	0	0	1	0
14	21	14	0	0	1	0
15	19	13	0	0	1	0
16	23	17	0	0	1	0
17	30	20	0	0	1	0
18	31	21	0	0	1	0
19	20	15	0	0	0	1
20	26	18	0	0	0	1
21	20	16	0	0	0	1
22	14	13	0	0	0	1
23	28	21	0	0	0	1
24	.	12	0	0	0	1

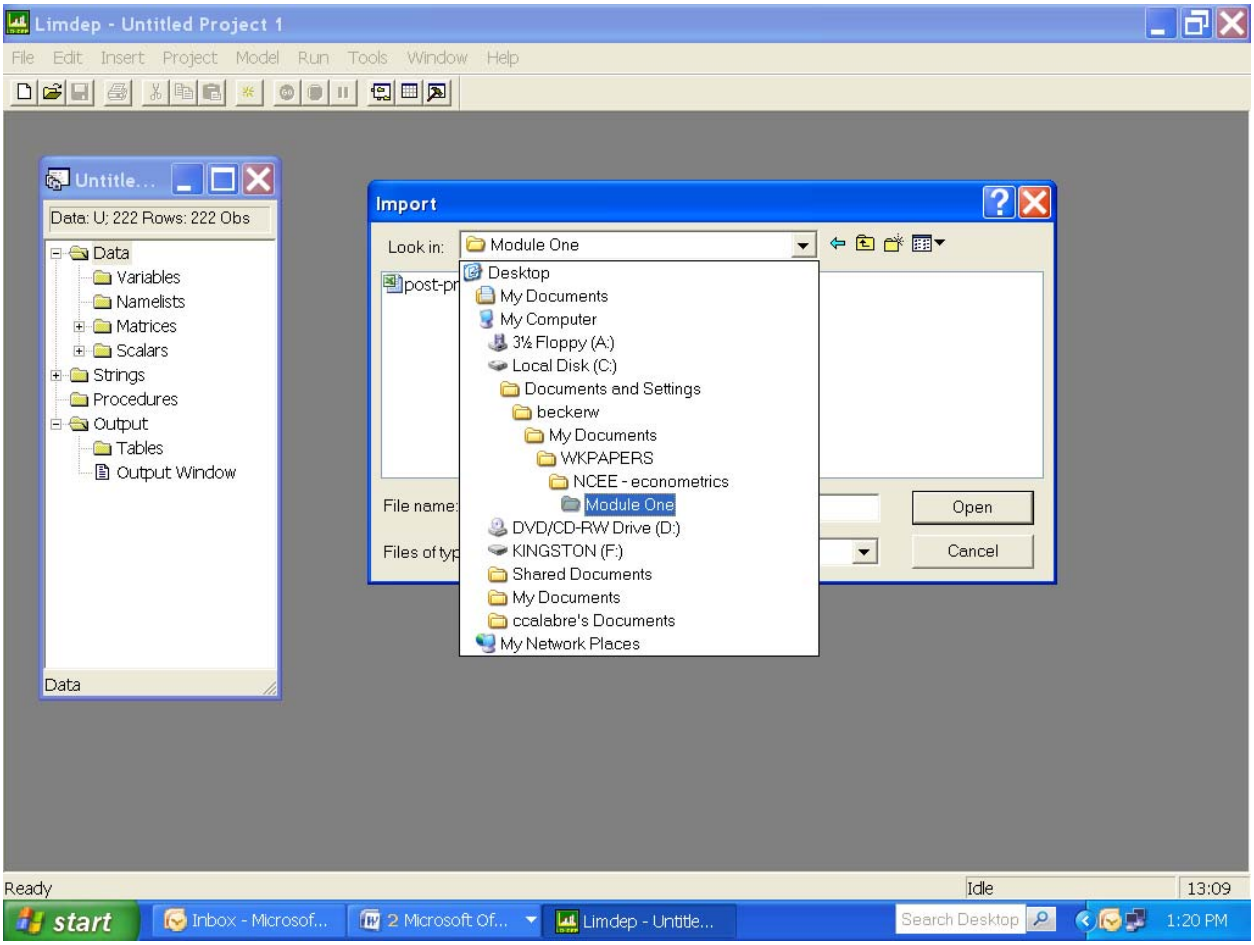
To start, the file "post-pre.xls" must be downloaded and copied to your computer's hard drive. Once this is done open LIMDEP. Clicking on "Project," "Import," and "Variables…" yields the following screen display:



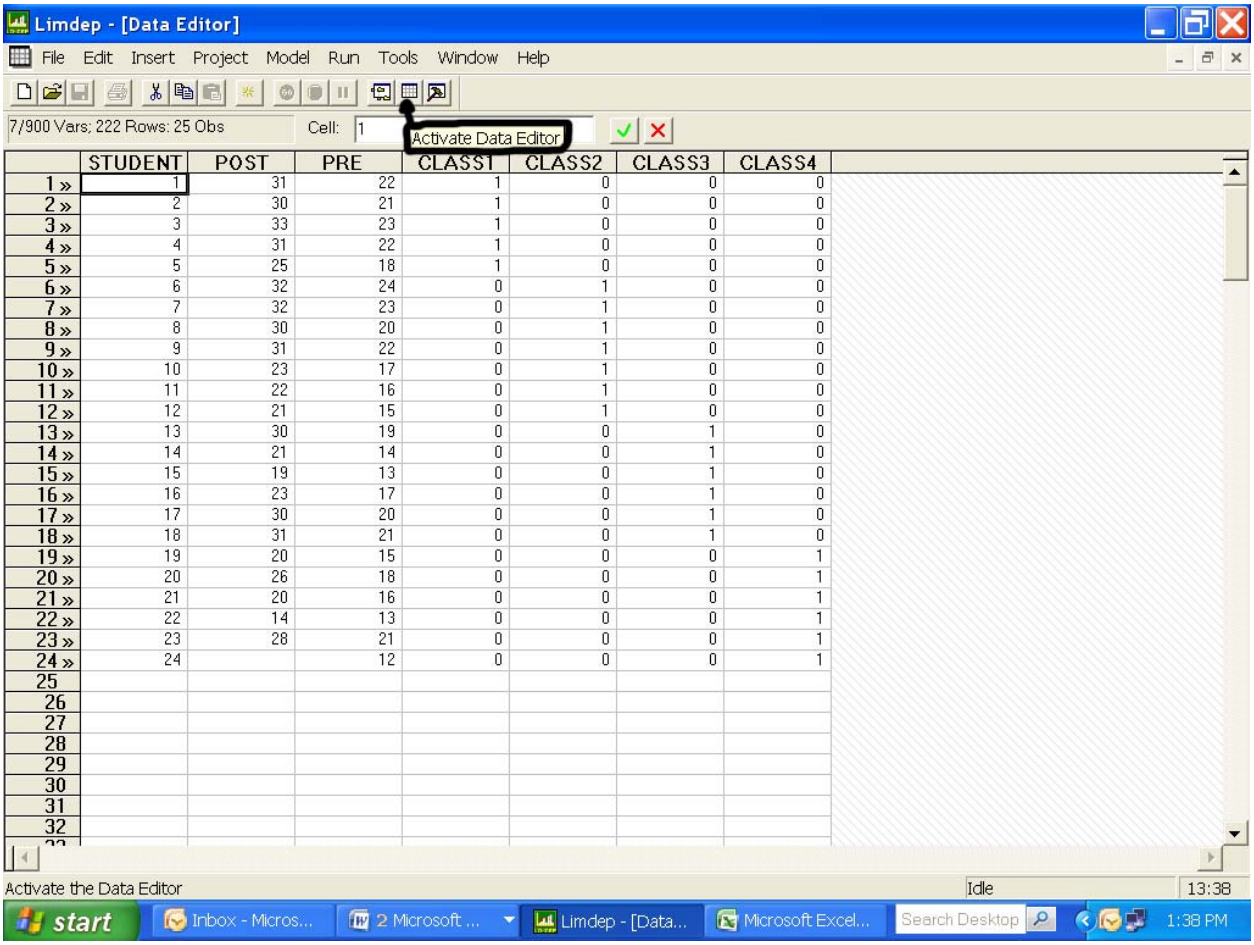
Clicking “Variable” gives a screen display of your folders in “My Documents,” in which you can locate files containing Excel (.wk and .xls) files.



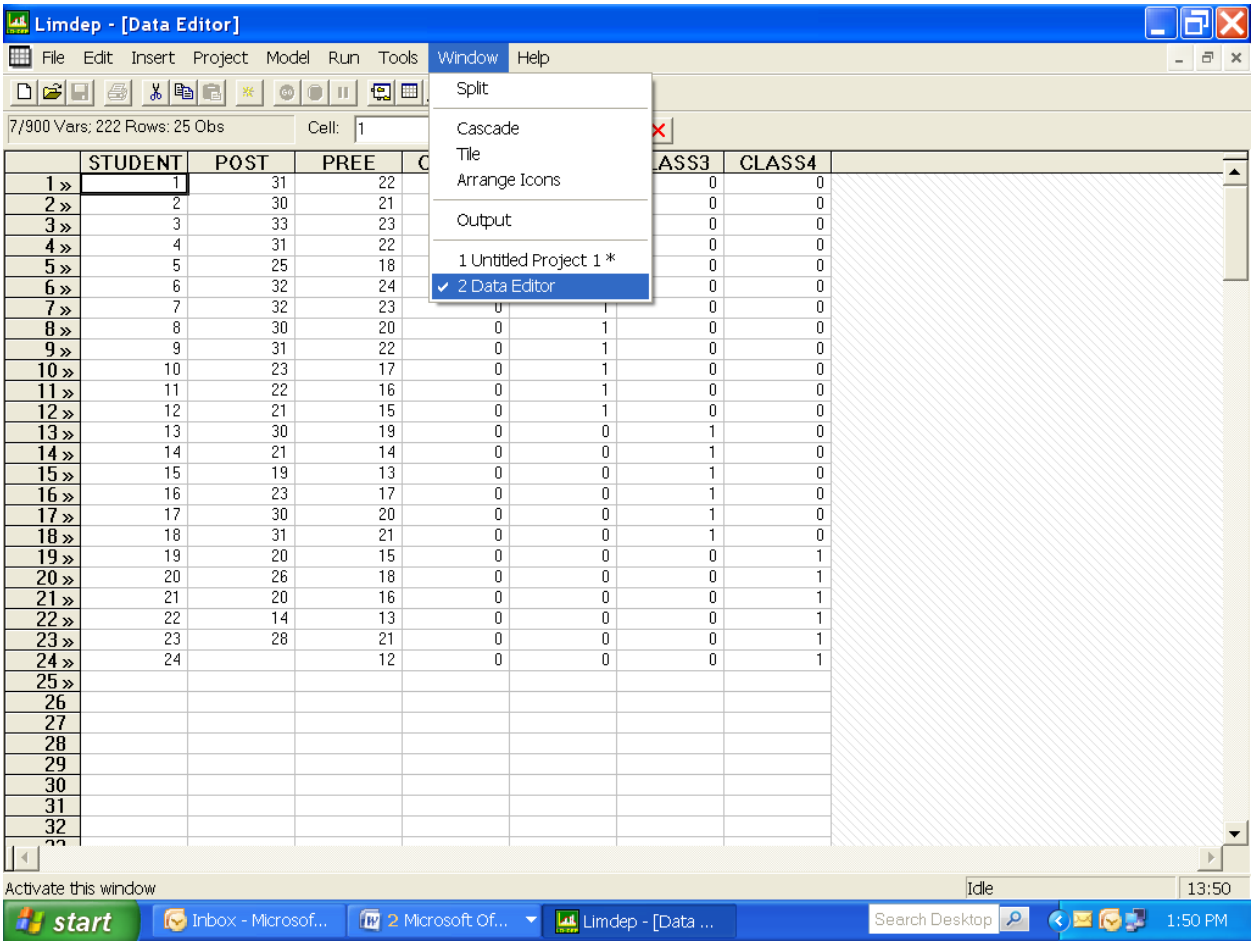
The next slide shows a path to the file “post-pre.xls.” (The path to your copy of “post-pre.xls” will obviously depend on where you placed it on your computer’s hard drive.) Clicking “Open” imports the file into LIMDEP.



To make sure the data has been correctly imported into LIMDEP, click the "Activate Data Editor" button, which is second from the right on the tool bar or go to Data Editor in the Window's menu. Notice that the missing observation for Student 24 appears as a blank in this data editor. The sequencing of these steps and the associated screens follow:

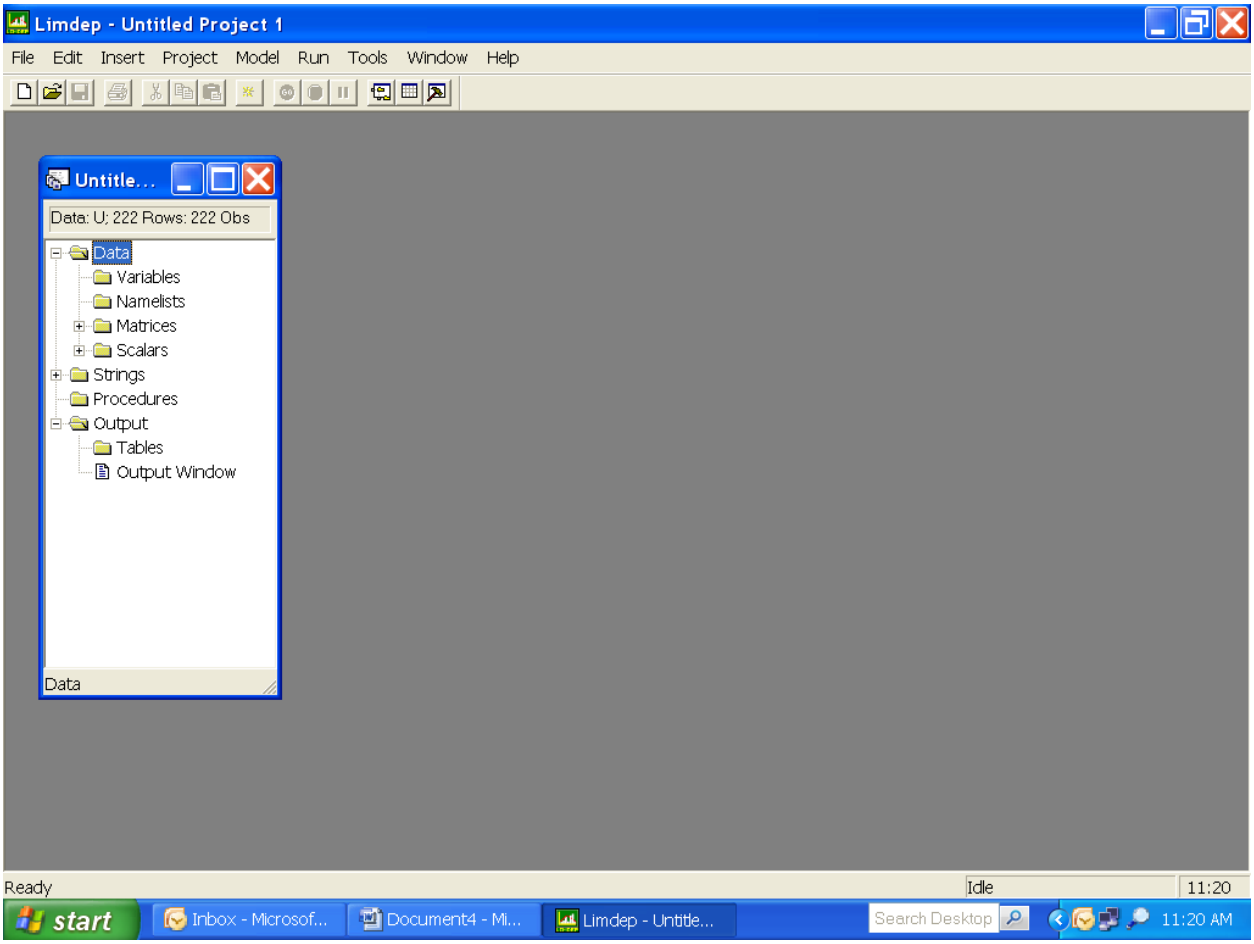


	STUDENT	POST	PRE	CLASS1	CLASS2	CLASS3	CLASS4
1 »	1	31	22	1	0	0	0
2 »	2	30	21	1	0	0	0
3 »	3	33	23	1	0	0	0
4 »	4	31	22	1	0	0	0
5 »	5	25	18	1	0	0	0
6 »	6	32	24	0	1	0	0
7 »	7	32	23	0	1	0	0
8 »	8	30	20	0	1	0	0
9 »	9	31	22	0	1	0	0
10 »	10	23	17	0	1	0	0
11 »	11	22	16	0	1	0	0
12 »	12	21	15	0	1	0	0
13 »	13	30	19	0	0	1	0
14 »	14	21	14	0	0	1	0
15 »	15	19	13	0	0	1	0
16 »	16	23	17	0	0	1	0
17 »	17	30	20	0	0	1	0
18 »	18	31	21	0	0	1	0
19 »	19	20	15	0	0	0	1
20 »	20	26	18	0	0	0	1
21 »	21	20	16	0	0	0	1
22 »	22	14	13	0	0	0	1
23 »	23	28	21	0	0	0	1
24 »	24		12	0	0	0	1
25							
26							
27							
28							
29							
30							
31							
32							

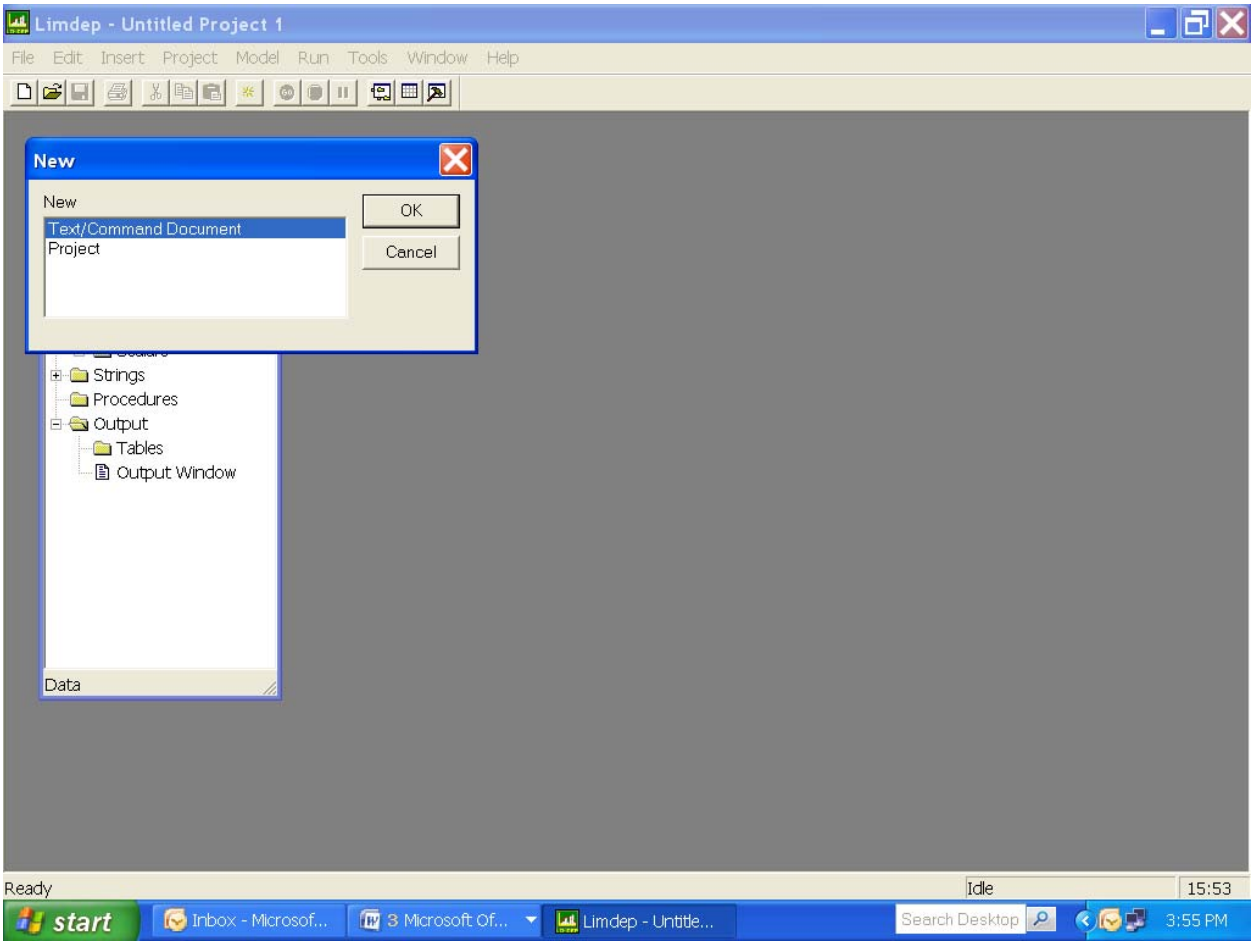

Limdep - [Data Editor]
File Edit Insert Project Model Run Tools Window Help
Split
Cascade
Tile
Arrange Icons
Output
1 Untitled Project 1 *
2 Data Editor
7/900 Vars; 222 Rows: 25 Obs
Cell: 1
STUDENT POST PREE CLASS3 CLASS4
1 » 1 31 22 0 0
2 » 2 30 21 0 0
3 » 3 33 23 0 0
4 » 4 31 22 0 0
5 » 5 25 18 0 0
6 » 6 32 24 0 0
7 » 7 32 23 0 0
8 » 8 30 20 0 1 0 0
9 » 9 31 22 0 1 0 0
10 » 10 23 17 0 1 0 0
11 » 11 22 16 0 1 0 0
12 » 12 21 15 0 1 0 0
13 » 13 30 19 0 0 1 0
14 » 14 21 14 0 0 1 0
15 » 15 19 13 0 0 1 0
16 » 16 23 17 0 0 1 0
17 » 17 30 20 0 0 1 0
18 » 18 31 21 0 0 1 0
19 » 19 20 15 0 0 0 1
20 » 20 26 18 0 0 0 1
21 » 21 20 16 0 0 0 1
22 » 22 14 13 0 0 0 1
23 » 23 28 21 0 0 0 1
24 » 24 12 0 0 0 1
25 »
26
27
28
29
30
31
32
Activate this window
Idle
13:50
start
Inbox - Microsof...
2 Microsoft Of...
Limdep - [Data ...
Search Desktop
1:50 PM

READING SPACE DELINEATED TEXT FILES INTO LIMDEP

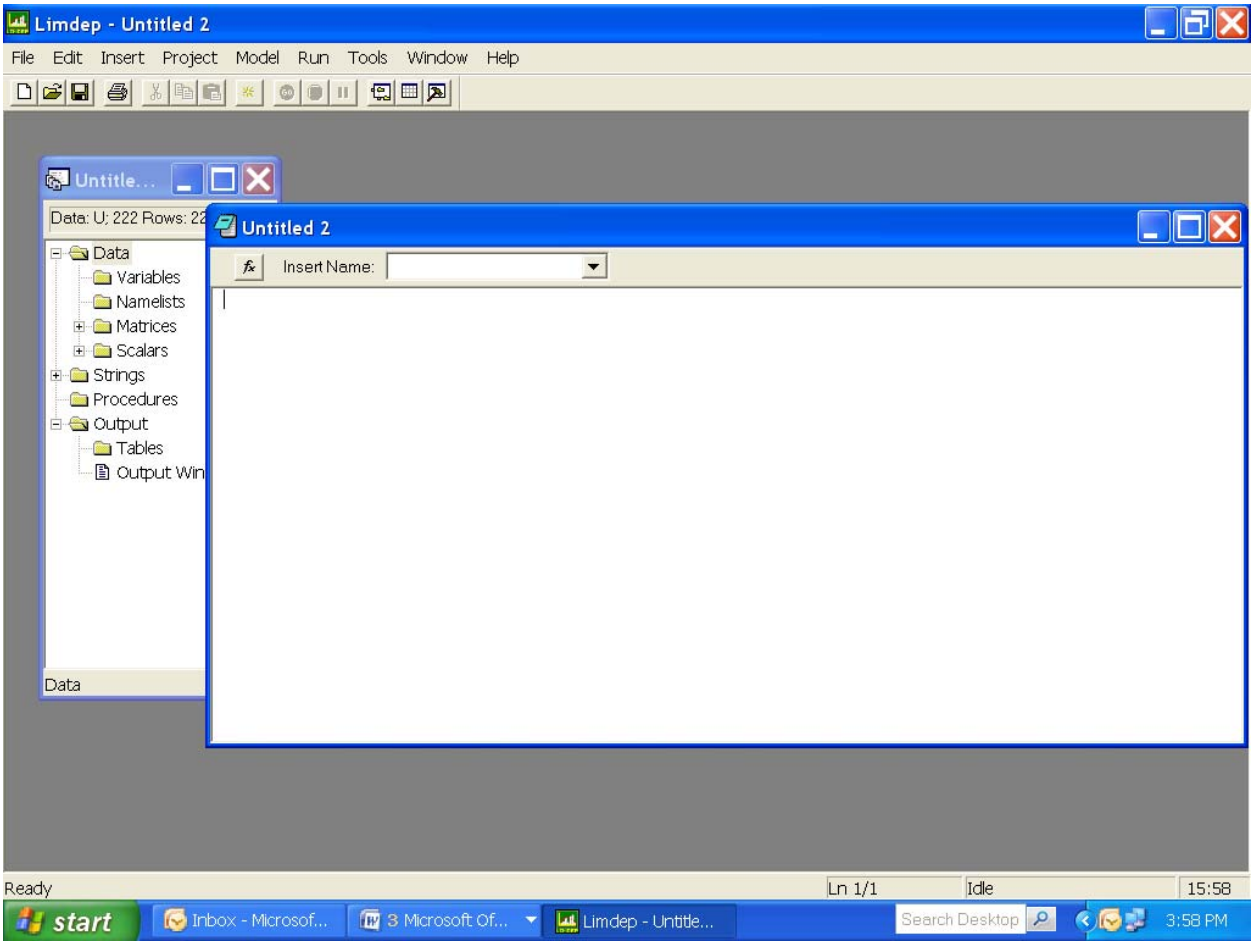
Next we consider externally created text files that are typically accompanied by the ".txt" or ".prn" extensions. For demonstration purposes, the data set we just employed with 24 observations on the 7 variables ("student," "post," "pre," "class1," "class2," "class3," and "class4") was saved as the space delineated text file "post-pre.txt." After downloading this file to your hard drive open LIMDEP to its first screen:



To read the file "post-pre.txt," begin by clicking "File" in the upper left-hand corner of the ribbon, which will yield the following screen display:



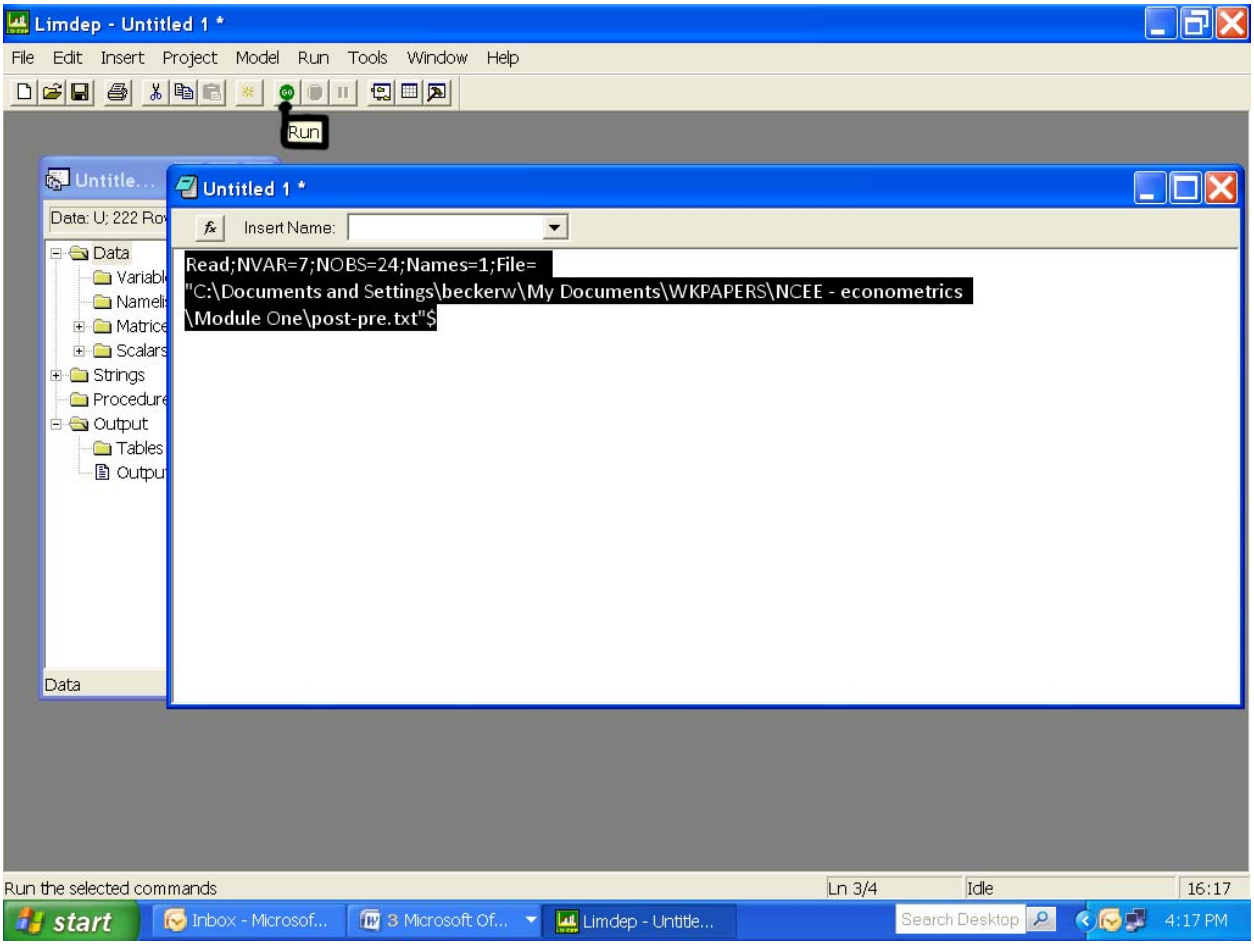
Click on “OK” to “Text/Command Document” to create a file into which all your commands will go.



The "read" command is typed or copied into the "Untitled" command file, with all subparts of a command separated with semicolons (;). The program is not case sensitive; thus, upper and lower case letters can be used interchangeably. The read command includes the number of variables or columns to be read (nvar=), the number of records or observations for each variable (nobs=), and the place to find the file (File=). Because the names of the variables are on the first row of the file to be read, we tell this to LIMDEP with the Names=1 command. If the file path is long and involves spaces (as it is here, but your path will depend on where you placed your file), then quote marks are required around the path. The $ indicates the end of the command.

```
Read;NVAR=7;NOBS=24;Names=1;File=
"C:\Documents and Settings\beckerw\My Documents\WKPAPERS\NCEE - econometrics
\Module One\post-pre.txt"$
```

Upon copying or typing this read command into the command file and highlighting the entire three lines, the screen display appears as below and the “Go” button is pressed to run the command.

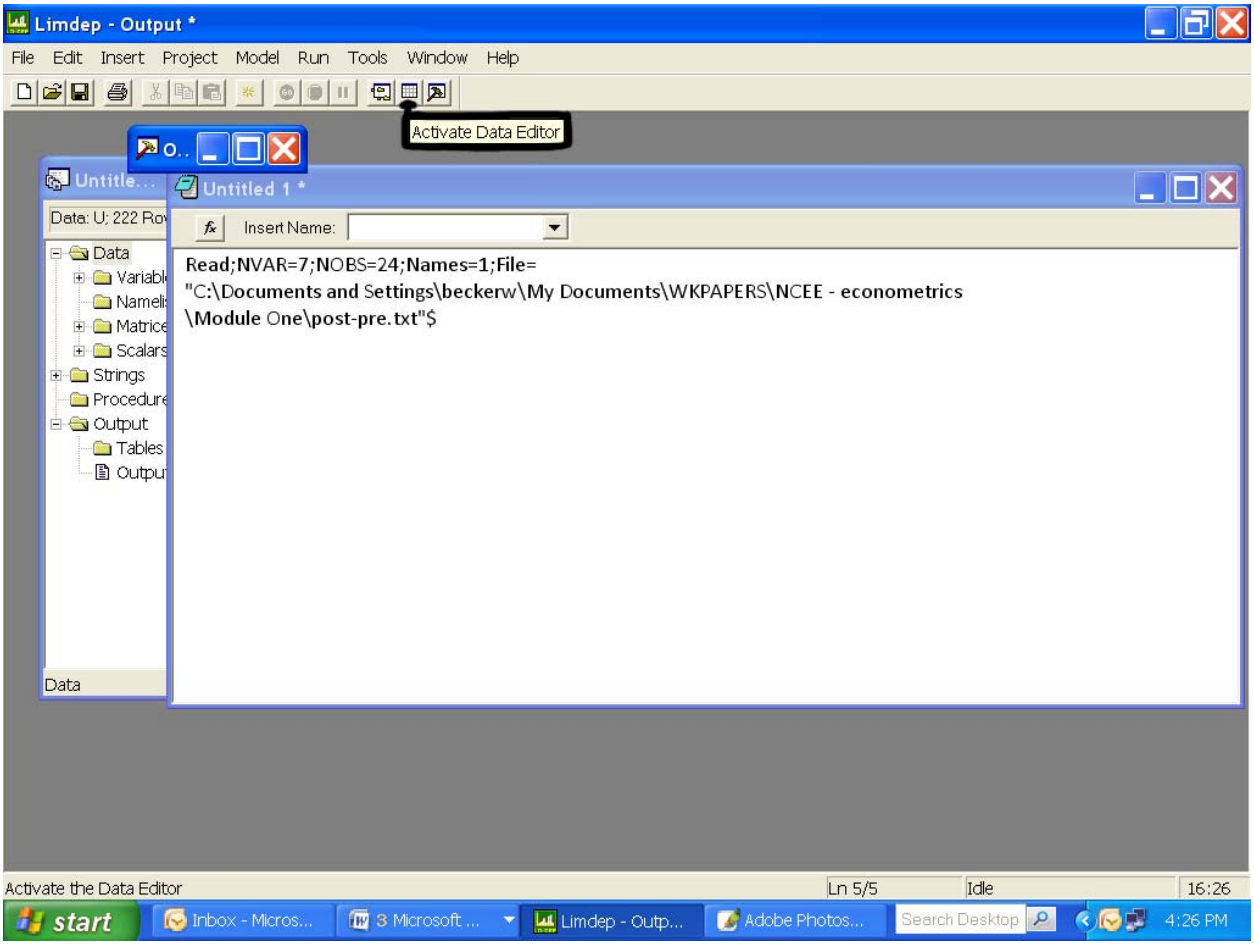


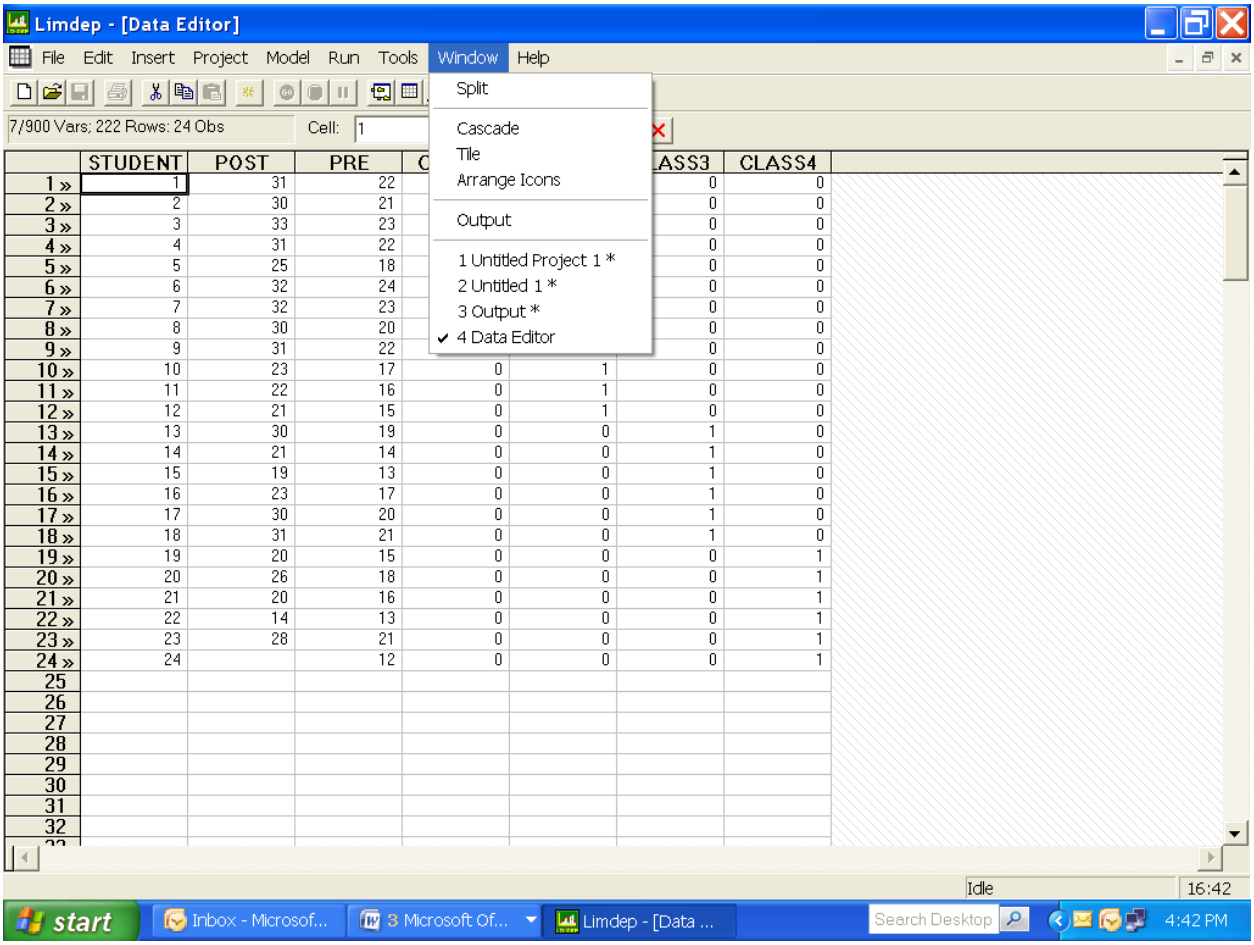
LIMDEP tells the user that it has attempted the command with the appearance of

To check on the correct reading of the data, click the “Activate Data Editor” button, which is second from the right on the tool bar or go to Data Editor in the Window’s menu. Notice that if

you use the Window's menus, there are now four files open within Limdep: Untitled Project 1*, Untitled 1*, Output 1*, and Data Editor. As you already know, Untitled 1 contains your read command and Untitled Project is just information in the opening LIMDEP screen. Output contains the commands that LIMDEP has attempted, which so far only includes the read command. This output file could have also been accessed by clicking on the view square next to the X box in the following rectangle

When it appeared to check on whether the read command was properly executed.

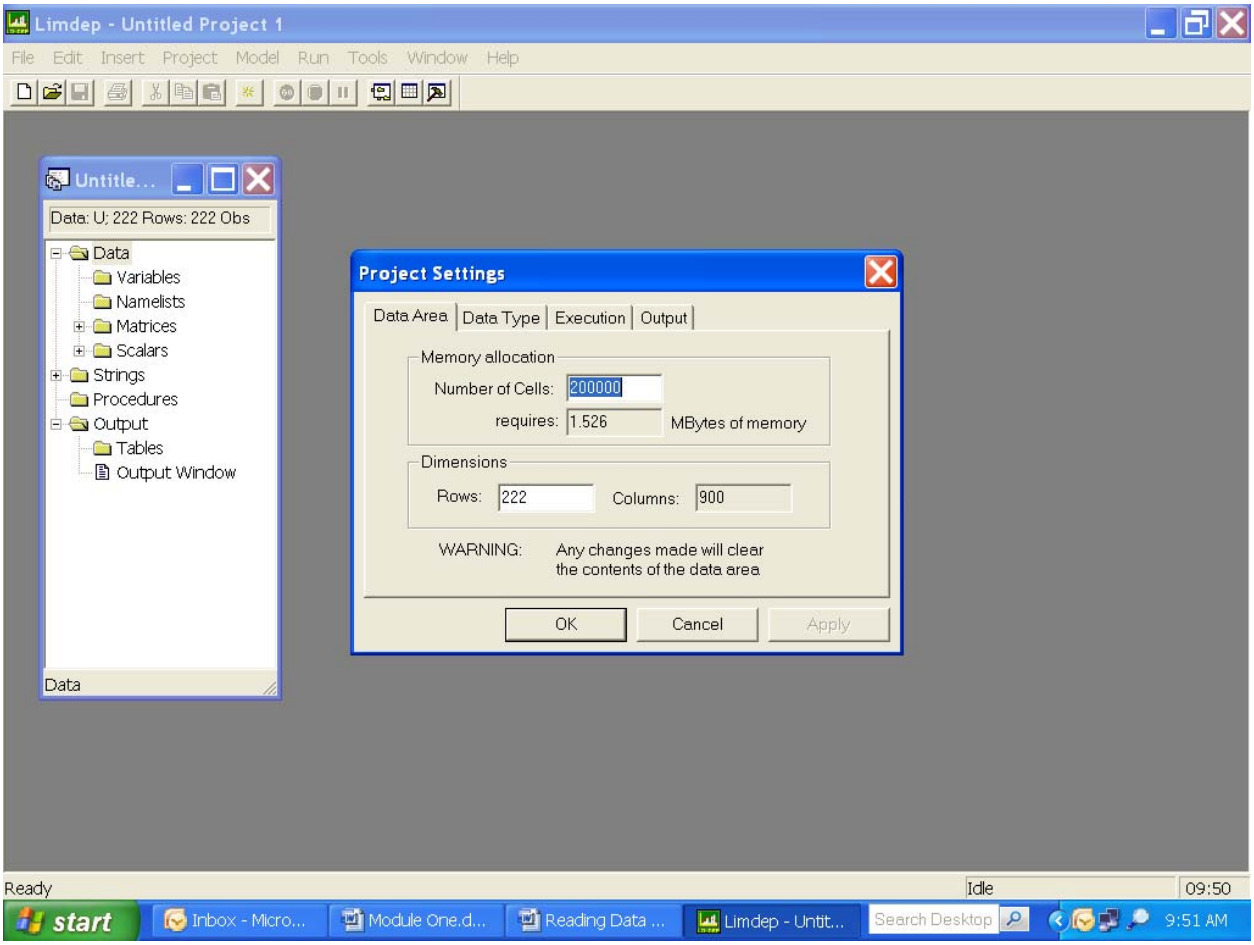




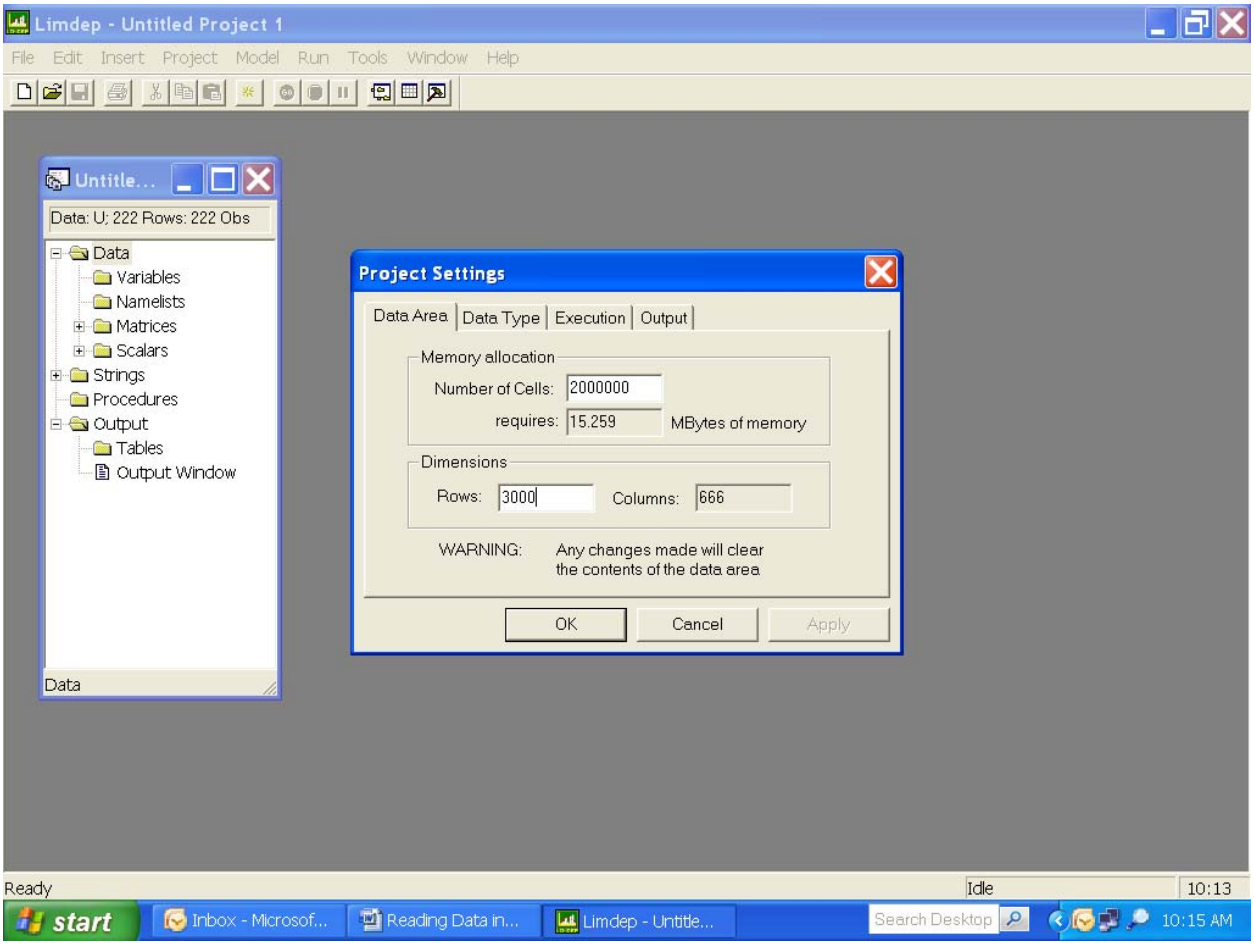
READING LARGE FILES INTO LIMDEP

LIMDEP has a data matrix default restriction of no more than 222 rows (records per variable), 900 columns (number of variables) and 200,000 cells. To read, import or create nonconforming data sets this default setting must be changed. For example, to accommodate larger data sets the number of rows must be increased. If the creation of more than 900 variables is anticipated, even if less than 900 variables were initially imported, then the number of columns must be increased before any data is read. This is accomplished by clicking the project button on the top ribbon, going to settings, and changing the number of cells and number of rows.

As an example, consider the data set employed by Becker and Powers (2001), which initially had 2,837 records. Open LIMDEP and go to "Project" and then "Settings…," which yields the following screen display:



Increasing the "Number of Cells" from 200,000 to 2,000,000 and increasing "Rows" from 222 to 3,000, automatically resets the "Columns" to 666, which is more than sufficient to read the 64 variables in the initial data set and to accommodate any variables to be created within LIMDEP. Pressing "OK" resets the memory allocation that LIMDEP will employ for this data set.



This Becker and Powers data set does not have variable names imbedded in it. Thus, they will be added to the read command. To now read the data follow the path "Files" to "New" to "Text/Command Document" and click "OK." Entering the following read statement into the Text/Command file, highlighting it, and pushing the green "Go" button will enter the 2,837 records on 64 variables in file beck8WO into LIMDEP and each of the variables will be named as indicated by each two character label.

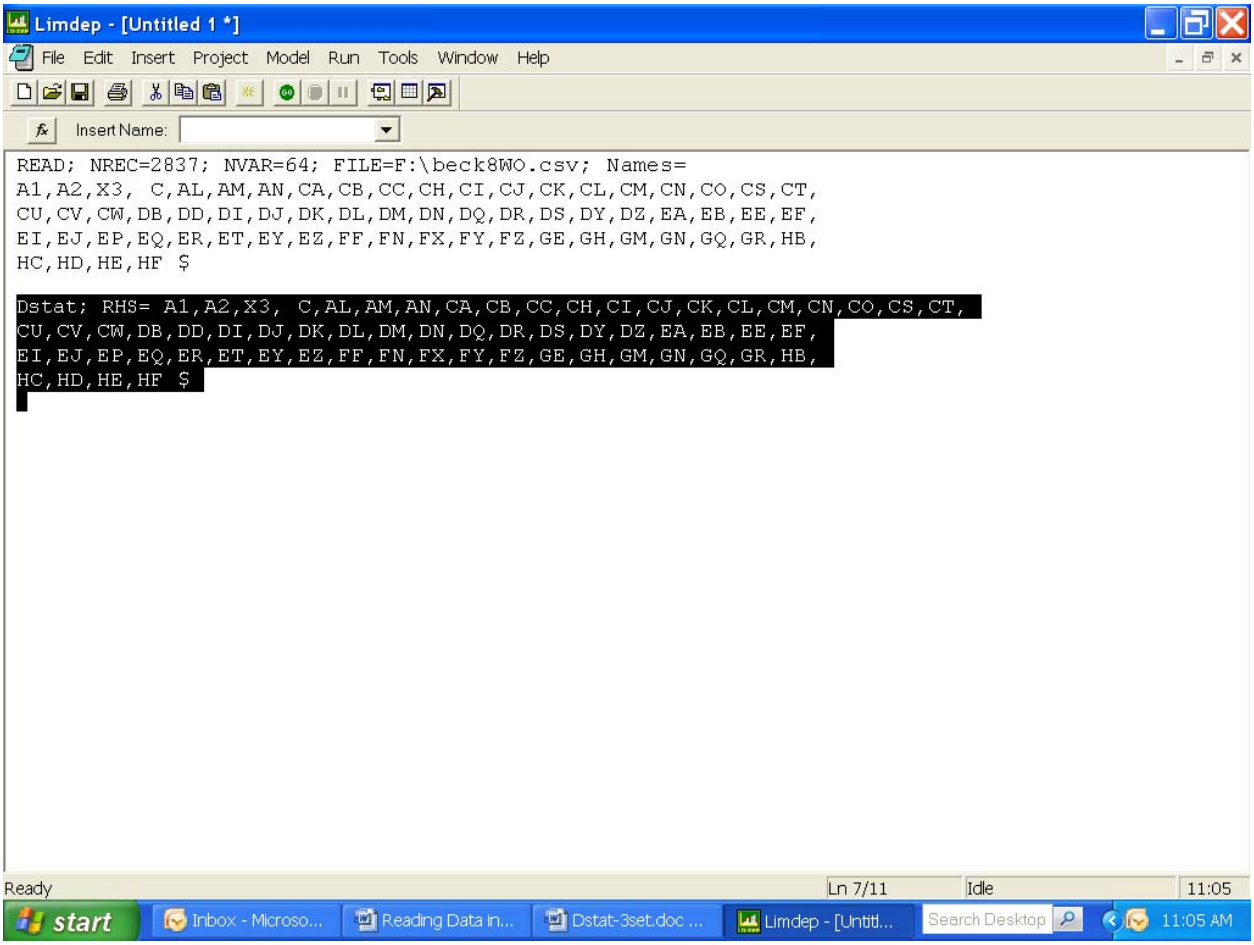
```
READ; NREC=2837; NVAR=64; FILE=F:\beck8WO.csv; Names=
A1,A2,X3, C,AL,AM,AN,CA,CB,CC,CH,CI,CJ,CK,CL,CM,CN,CO,CS,CT,
CU,CV,CW,DB,DD,DI,DJ,DK,DL,DM,DN,DQ,DR,DS,DY,DZ,EA,EB,EE,EF,
EI,EJ,EP,EQ,ER,ET,EY,EZ,FF,FN,FX,FY,FZ,GE,GH,GM,GN,GQ,GR,HB,
HC,HD,HE,HF $
```

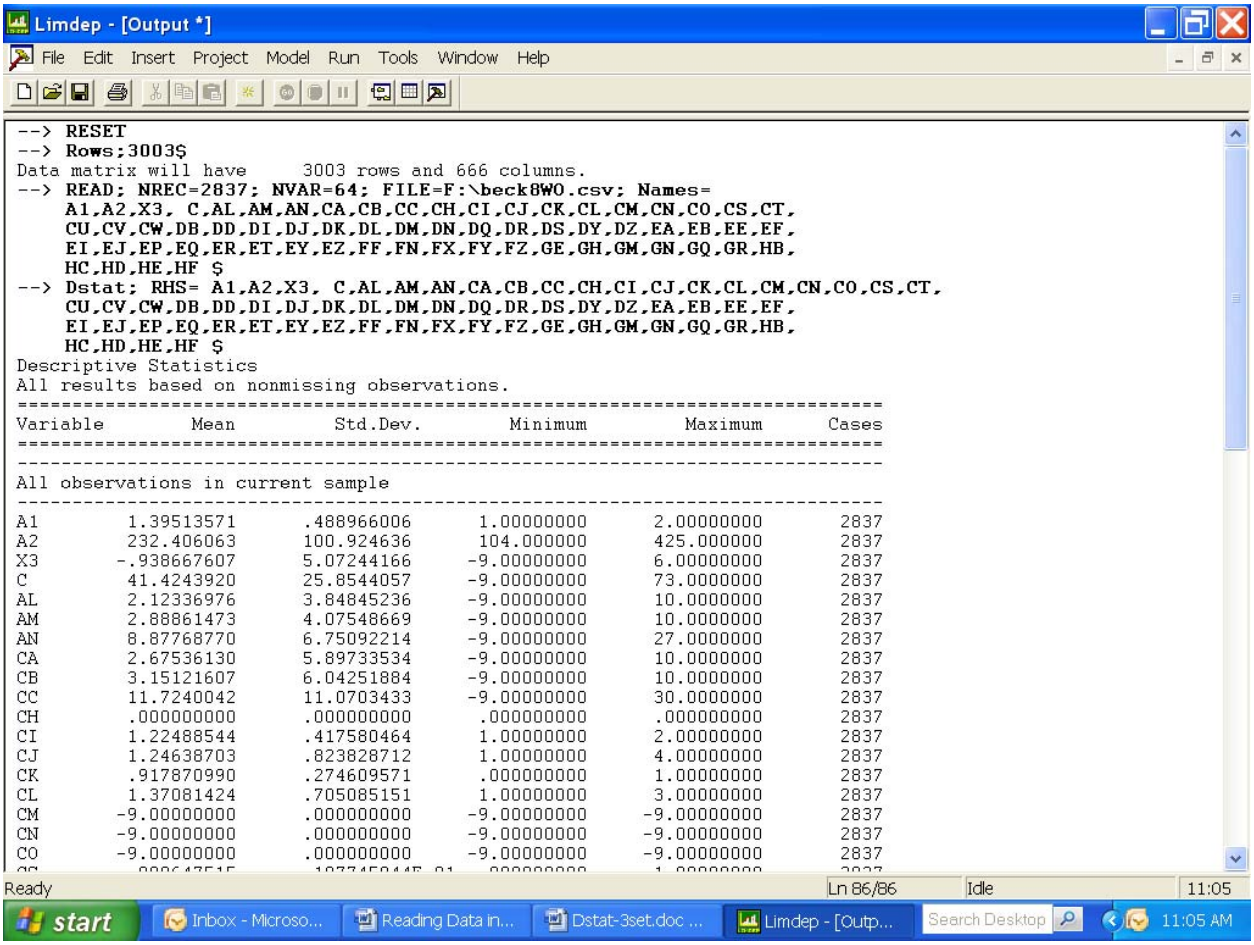
Defining all the variables is not critical for our purposes here, but the variables used in the Becker and Power's study required the following definitions (where names are not upper- and lower-case sensitive):

A1: Term, where 1= fall, 2 = spring
A2: School code, where 100/199 = doctorate, 200/299 = comprehensive, 300/399 = lib arts, 400/499 = 2 year
hb: initial class size (number taking preTUCE)
hc: final class size (number taking postTUCE)
dm: experiences measured by number of years teaching
dj: teacher's highest degree, where Bachelors=1, Masters=2, PhD=3
cc: postTUCE score (0 to 30)
an: preTUCE score (0 to 30)
ge: Student evaluation measured interest
gh: Student evaluation measured textbook quality
gm: Student evaluation measured regular instructor's English ability
gq: Student evaluation measured overall teaching effectiveness
ci: Instructor sex (Male=1, Female=2)
ck: English is native language of instructor (Yes=1, No=0)
cs: PostTUCE score counts toward course grade (Yes=1, No=0)
ff: GPA*100
fn: Student had high school economics (Yes=1, No=0)
ey: Student's sex (Male=1, Female=2)
fx: Student working in a job (Yes=1, No=0)

Notice that this data set is too large to fit in LIMDEP's "Active Data Editor" but all of the data are there as verified with the following DSTAT command, which is entered in the Text/Command file and highlighted. Upon clicking on the Go button, the descriptive statistics for each variable appear in the output file. Again, the output file is accessed via the Window tab in the upper ribbon. (Notice that in this data set, all missing values were coded −9.)

```
Dstat; RHS= A1,A2,X3, C,AL,AM,AN,CA,CB,CC,CH,CI,CJ,CK,CL,CM,CN,CO,CS,CT,
CU,CV,CW,DB,DD,DI,DJ,DK,DL,DM,DN,DQ,DR,DS,DY,DZ,EA,EB,EE,EF,
EI,EJ,EP,EQ,ER,ET,EY,EZ,FF,FN,FX,FY,FZ,GE,GH,GM,GN,GQ,GR,HB,
HC,HD,HE,HF $
```


Limdep - [Untitled 1 *]
File Edit Insert Project Model Run Tools Window Help
Insert Name:
READ; NREC=2837; NVAR=64; FILE=F:\beck8WO.csv; Names=
A1,A2,X3, C,AL,AM,AN,CA,CB,CC,CH,CI,CJ,CK,CL,CM,CN,CO,CS,CT,
CU,CV,CW,DB,DD,DI,DJ,DK,DL,DM,DN,DQ,DR,DS,DY,DZ,EA,EB,EE,EF,
EI,EJ,EP,EQ,ER,ET,EY,EZ,FF,FN,FX,FY,FZ,GE,GH,GM,GN,GQ,GR,HB,
HC,HD,HE,HF $
Dstat; RHS= A1,A2,X3, C,AL,AM,AN,CA,CB,CC,CH,CI,CJ,CK,CL,CM,CN,CO,CS,CT,
CU,CV,CW,DB,DD,DI,DJ,DK,DL,DM,DN,DQ,DR,DS,DY,DZ,EA,EB,EE,EF,
EI,EJ,EP,EQ,ER,ET,EY,EZ,FF,FN,FX,FY,FZ,GE,GH,GM,GN,GQ,GR,HB,
HC,HD,HE,HF $
Ready
Ln 7/11
Idle
11:05
start
Inbox - Microso...
Reading Data in...
Dstat-3set.doc ...
Limdep - [Untitl...
Search Desktop
11:05 AM



In summary, the LIMDEP Help menu states that the READ command is of the general form

READ ; Nobs = number of observations
; Nvar = number of variables
; Names = list of Nvar names
; File = name of the data file $

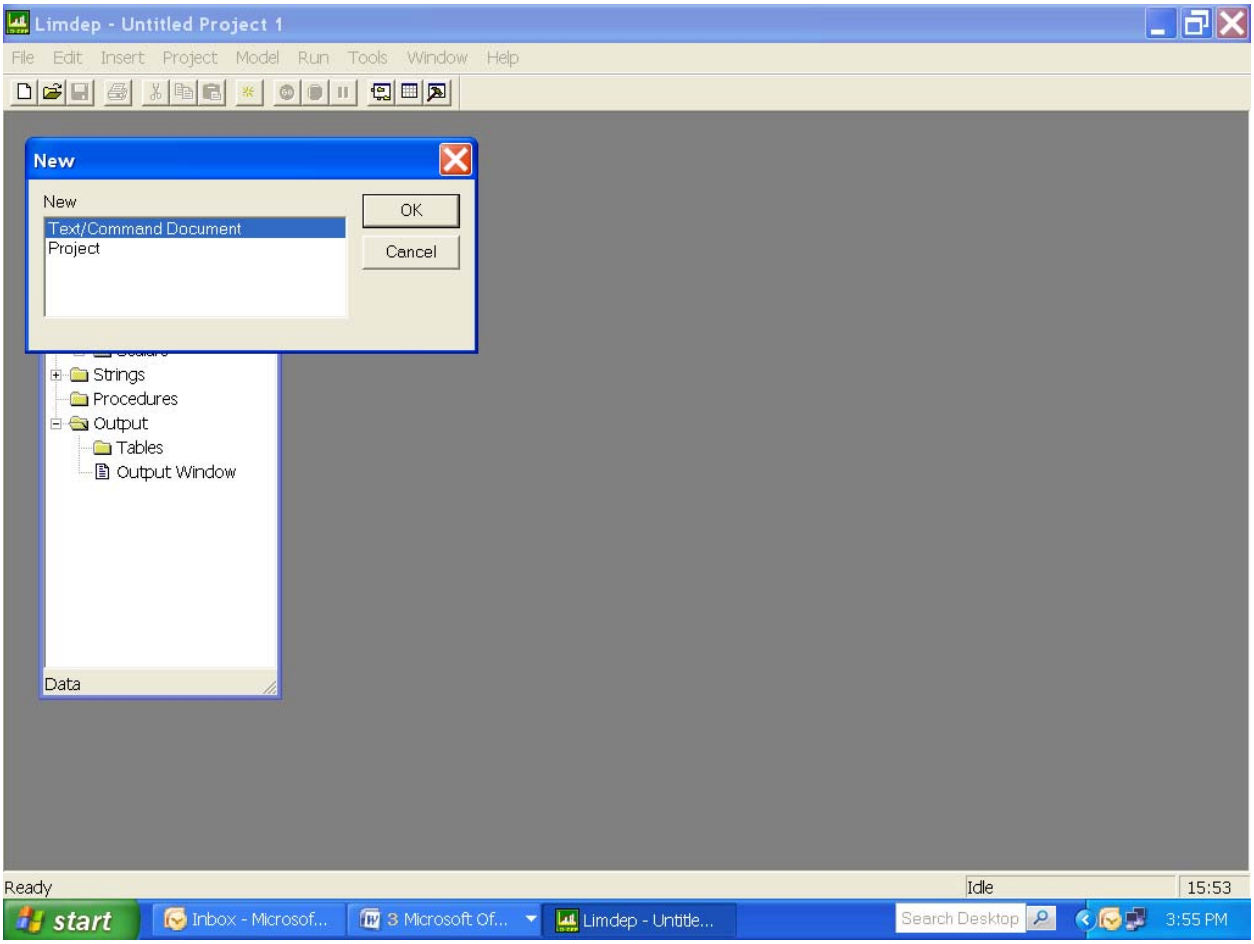
The default is an ASCII (or text) data file in which numbers are separated by blanks, tabs, and/or commas. Although not demonstrated here, LIMDEP will also read formatted files by adding the option "; Format = (Fortran format)" to the read command. In addition, although not demonstrated, small data sets can be cut and pasted directly into the Test/Command Document, preceded by a simple read command "READ ; Nobs = number of observations; Nvar = number of variables$", where ";Names = 1" would also be added if names appear on the line before the data.

LEAST-SQUARES ESTIMATION AND LINEAR RESTRICTIONS IN LIMDEP

To demonstrate some of the least-squares regression commands in LIMDEP, read either the Excel or space delineated text version of the 24 observations and 7 variable "post-pre" data set into LIMDEP. The model to be estimated is

$$post = \beta_1 + \beta_2 pre + f(classes) + \varepsilon$$

All statistical and mathematical instructions must be placed in the "Text/Command Document" of LIMDEP, which is accessed via the "File" to "New" route described earlier:



Once in the "Text/Command Document," the command for a regression can be entered. Before doing this, however, recall that the posttest score is missing for the 24th person, as can be seen in the "Active Data Editor." LIMDEP automatically codes all missing data that appear in a text or Excel file as "." with the value −999. If a regression is estimated with the entire data set,

then this fictitious −999 place holding value would be incorrectly employed. To avoid this, all commands can be prefaced with "skip," which tells LIMDEP to not use records involving −999. (In the highly unlikely event that −999 is a legitimate value, then a recoding is required as discussed below.) The syntax for regressions in LIMDEP is

Regress; lhs= ???; rhs=one, ??? $

where "lhs=" is the left-hand-side dependent variable and "rhs=" is the right-hand-side explanatory variable. The "one" is included on the right-hand-side to estimate a *y*-intercept. If this "one" is not specified then the regression is forced to go through the origin – that is, no constant term is estimated. Finally, LIMDEP will automatically predict the value of the dependent variable, including 95 percent confidence intervals, and show the results in the output file by adding "fill; list" to the regression command:

Regress; lhs= ???; rhs=one, ???; fill; list $

To demonstrate some of the least-squares regression commands in LIMDEP read either the Excel or space delineated text version of the 24 observations and 7 variables "post-pre" data set into LIMDEP. The model to be estimated is

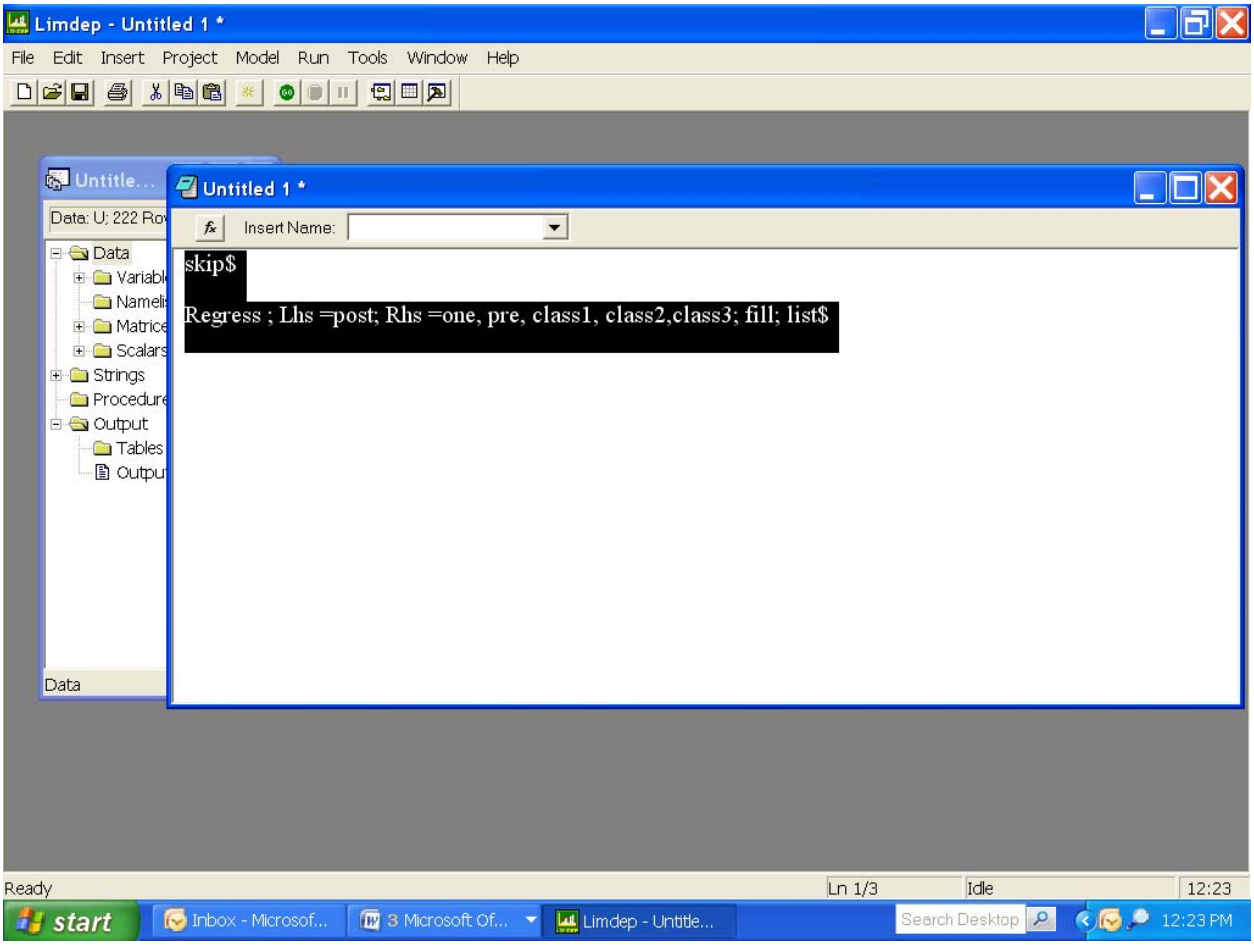
$$post = \beta_1 + \beta_2 pre + \beta_3 class1 + \beta_4 class2 + \beta_5 class3 + \varepsilon$$

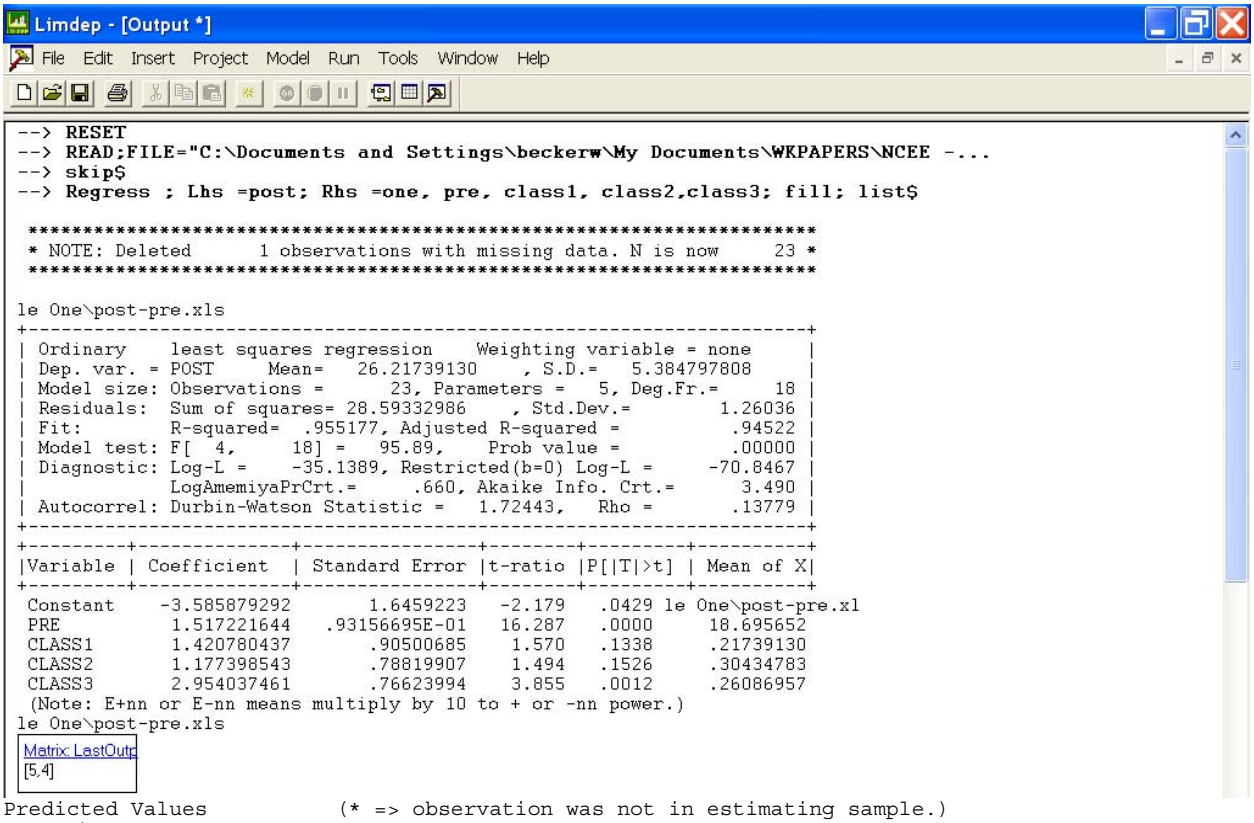
To avoid the sum of the four dummy variables equaling the column of ones in the data set, the fourth class is not included. The commands to be typed into the Untitled Text/Command Document are now

skip$

Regress ; Lhs =post; Rhs =one, pre, class1, class2,class3; fill; list$

Highlighting these commands and pressing "Go" gives the results in the LIMDEP output file:


Limdep - Untitled 1 *
File Edit Insert Project Model Run Tools Window Help
Untitle...
Data: U; 222 Rov
Data
Variabl
Namelis
Matrice
Scalars
Strings
Procedure
Output
Tables
Output
Data
Untitled 1 *
Insert Name:
skip$
Regress ; Lhs =post; Rhs =one, pre, class1, class2,class3; fill; list$
Ready
Ln 1/3
Idle
12:23
start
Inbox - Microsof...
3 Microsoft Of...
Limdep - Untitle...
Search Desktop
12:23 PM




```
Predicted Values          (* => observation was not in estimating sample.)
le One\post-pre.xls
Observation        Observed Y   Predicted Y   Residual   95% Forecast Interval
        1          31.000        31.214        -.2138      28.3089    34.1187
        2          30.000        29.697         .3034      26.7956    32.5975
        3          33.000        32.731         .2690      29.8090    35.6530
        4          31.000        31.214        -.2138      28.3089    34.1187
        5          25.000        25.145        -.1449      22.1774    28.1124
        6          32.000        34.005       -2.0048      31.0444    36.9653
        7          32.000        32.488        -.4876      29.5784    35.3968
        8          30.000        27.936        2.0640      25.1040    30.7679
        9          31.000        30.970         .0296      28.1000    33.8408
       10          23.000        23.384        -.3843      20.5091    26.2594
       11          22.000        21.867         .1329      18.9513    24.7828
       12          21.000        20.350         .6502      17.3811    23.3186
       13          30.000        28.195        1.8046      25.3167    31.0740
       14          21.000        20.609         .3907      17.6757    23.5428
       15          19.000        19.092        -.0920      16.1089    22.0752
       16          23.000        25.161       -2.1609      22.3001    28.0218
       17          30.000        29.713         .2874      26.8053    32.6199
       18          31.000        31.230        -.2298      28.2811    34.1786
       19          20.000        19.172         .8276      16.2549    22.0900
       20          26.000        23.724        2.2759      20.8105    26.6377
       21          20.000        20.690        -.6897      17.7866    23.5927
       22          14.000        16.138       -2.1380      13.1530    19.1230
       23          28.000        28.276        -.2758      25.2500    31.3016
*      24         No data        14.621      No data       11.5836    17.6579
```

From this output the Predicted posttest score is 14.621, with 95 percent confidence interval equal to 11.5836< E(y|$\mathbf{X_{24}}$)< 17.6579.

A researcher might be interested to test whether the class in which a student is enrolled affects his/her post-course test score, assuming fixed effects only. This linear restriction is done

automatically in LIMDEP by adding the following "cls:" command to the regression statement in the Text/Command Document.

Regress ; Lhs =post; Rhs =one, pre, class1, class2,class3; CLS: b(3)=0,b(4)=0,b(5)=0$

Upon highlighting and pressing the Go button, the following results will appear in the output file:

```
--> Regress ; Lhs =post; Rhs =one, pre, class1, class2,class3; CLS: b(3)=0,b(...

 ******************************************************************************
 * NOTE: Deleted      1 observations with missing data. N is now     23 *
 ******************************************************************************

le One\post-pre.xls
+------------------------------------------------------------------------------+
| Ordinary    least squares regression    Weighting variable = none           |
| Dep. var. = POST     Mean=   26.21739130    , S.D.=   5.384797808           |
| Model size: Observations =      23, Parameters =   5, Deg.Fr.=     18        |
| Residuals:  Sum of squares= 28.59332986    , Std.Dev.=        1.26036        |
| Fit:        R-squared=  .955177, Adjusted R-squared =         .94522        |
| Model test: F[  4,     18] =   95.89,    Prob value =          .00000        |
| Diagnostic: Log-L =    -35.1389, Restricted(b=0) Log-L =     -70.8467        |
|             LogAmemiyaPrCrt.=     .660, Akaike Info. Crt.=      3.490        |
| Autocorrel: Durbin-Watson Statistic =   1.72443,   Rho =       .13779        |
+------------------------------------------------------------------------------+
+---------+--------------+----------------+--------+---------+----------+
|Variable | Coefficient  | Standard Error |t-ratio |P[|T|>t] | Mean of X|
+---------+--------------+----------------+--------+---------+----------+
 Constant    -3.585879292       1.6459223   -2.179   .0429 le One\post-pre.xl
 PRE          1.517221644   .93156695E-01   16.287   .0000     18.695652
 CLASS1       1.420780437       .90500685    1.570   .1338     .21739130
 CLASS2       1.177398543       .78819907    1.494   .1526     .30434783
 CLASS3       2.954037461       .76623994    3.855   .0012     .26086957
  (Note: E+nn or E-nn means multiply by 10 to + or -nn power.)
le One\post-pre.xls

le One\post-pre.xls
+------------------------------------------------------------------------------+
| Linearly restricted regression                                               |
| Ordinary    least squares regression    Weighting variable = none           |
| Dep. var. = POST     Mean=   26.21739130    , S.D.=   5.384797808           |
| Model size: Observations =      23, Parameters =   2, Deg.Fr.=     21        |
| Residuals:  Sum of squares= 53.19669876    , Std.Dev.=        1.59160        |
| Fit:        R-squared=  .916608, Adjusted R-squared =         .91264        |
|             (Note:  Not using OLS.  R-squared is not bounded in [0,1]        |
| Model test: F[  1,     21] =  230.82,    Prob value =          .00000        |
| Diagnostic: Log-L =    -42.2784, Restricted(b=0) Log-L =     -70.8467        |
|             LogAmemiyaPrCrt.=    1.013, Akaike Info. Crt.=      3.850        |
| Note, when restrictions are imposed, R-squared can be less than zero.        |
| F[ 3,    18] for the restrictions =      5.1627, Prob =   .0095              |
| Autocorrel: Durbin-Watson Statistic =   1.12383,   Rho =       .43808        |
+------------------------------------------------------------------------------+
+---------+--------------+----------------+--------+---------+----------+
|Variable | Coefficient  | Standard Error |t-ratio |P[|T|>t] | Mean of X|
+---------+--------------+----------------+--------+---------+----------+
 Constant    -2.211829436       1.9004224   -1.164   .2597 le One\post-pre.xl
 PRE          1.520632737       .10008855   15.193   .0000     18.695652
 CLASS1       .0000000000 ........(Fixed Parameter)........    .21739130
 CLASS2   -.4440892099E-15........(Fixed Parameter)........    .30434783
 CLASS3   -.4440892099E-15........(Fixed Parameter)........    .26086957
```

```
 (Note: E+nn or E-nn means multiply by 10 to + or -nn power.)
le One\post-pre.xls
```

From the second part of this printout, the appropriate F to test

H_o: $\beta_3 = \beta_4 = \beta_5 = 0$ versus H_a: at least one Beta is nonzero

is F[df1=3,df2=18] = 5.1627, with p-value = 0.0095. Thus, null hypothesis that none of the dummies is significant at 0.05 Type I error level can be rejected in favor of the hypothesis that at least one class is significant, assuming that the effect of pre-course test score on post-course test score is the same in all classes and only the constant is affected by class assignment.

STRUCTURAL (CHOW) TEST

The above test of the linear restriction $\beta_3 = \beta_4 = \beta_5 = 0$ (no difference among classes), assumed that the pretest slope coefficient was constant, fixed and unaffected by the class to which a student belonged. A full structural test requires the fitting of four separate regressions to obtain the four residual sum of squares that are added to obtain the unrestricted sum of squares. The restricted sum of squares is obtained from a regression of posttest on pretest with no dummies for the classes; that is, the class to which a student belongs is irrelevant in the manner in which pretests determine the posttest score.

The commands to be entered into the Document/text file of LIMDEP are as follows:

Restricted Regression

```
Sample; 1-23$
Regress ; Lhs =post; Rhs =one, pre$
Calc    ; SSall = Sumsqdev$
```

Unrestricted Regressions

```
Sample; 1-5$
Regress ; Lhs =post; Rhs =one, pre$
Calc    ; SS1 = Sumsqdev$

Sample; 6-12$
Regress ; Lhs =post; Rhs =one, pre$
Calc    ; SS2 = Sumsqdev$

Sample; 13-18$
Regress ; Lhs =post; Rhs =one, pre$
Calc    ; SS3 = Sumsqdev$
```

Sample; 19-23$
Regress ; Lhs =post; Rhs =one, pre$
Calc ; SS4 = Sumsqdev$

Calc;List ;F=((SSall-(SS1+SS2+SS3+SS4))/3*2) / ((SS1+SS2+SS3+SS4)/(23-4*2))$

The LIMDEP output is

```
--> RESET
--> READ;FILE="C:\Documents and Settings\beckerw\My Documents\WKPAPERS\NCEE -...
--> Reject; post=-999$
--> Regress ; Lhs =post; Rhs =one, pre, class1, class2,class3; CLS: b(3)=0,b(...

+----------------------------------------------------------------------+
| Ordinary    least squares regression    Weighting variable = none     |
| Dep. var. = POST     Mean=   26.21739130    , S.D.=   5.384797808     |
| Model size: Observations =      23, Parameters =   5, Deg.Fr.=     18 |
| Residuals:  Sum of squares= 28.59332986    , Std.Dev.=        1.26036 |
| Fit:        R-squared=  .955177, Adjusted R-squared =          .94522 |
| Model test: F[  4,     18] =   95.89,    Prob value =          .00000 |
| Diagnostic: Log-L =    -35.1389, Restricted(b=0) Log-L =     -70.8467 |
|             LogAmemiyaPrCrt.=     .660, Akaike Info. Crt.=      3.490 |
| Autocorrel: Durbin-Watson Statistic =   1.72443,   Rho =       .13779 |
+----------------------------------------------------------------------+
+---------+-------------+----------------+--------+---------+----------+
|Variable | Coefficient  | Standard Error |t-ratio |P[|T|>t] | Mean of X|
+---------+-------------+----------------+--------+---------+----------+
 Constant    -3.585879292       1.6459223   -2.179   .0429
 PRE          1.517221644   .93156695E-01   16.287   .0000     18.695652
 CLASS1       1.420780437       .90500685    1.570   .1338     .21739130
 CLASS2       1.177398543       .78819907    1.494   .1526     .30434783
 CLASS3       2.954037461       .76623994    3.855   .0012     .26086957
 (Note: E+nn or E-nn means multiply by 10 to + or -nn power.)
+----------------------------------------------------------------------+
| Linearly restricted regression                                       |
| Ordinary    least squares regression    Weighting variable = none     |
| Dep. var. = POST     Mean=   26.21739130    , S.D.=   5.384797808     |
| Model size: Observations =      23, Parameters =   2, Deg.Fr.=     21 |
| Residuals:  Sum of squares= 53.19669876    , Std.Dev.=        1.59160 |
| Fit:        R-squared=  .916608, Adjusted R-squared =          .91264 |
|             (Note:  Not using OLS.  R-squared is not bounded in [0,1] |
| Model test: F[  1,     21] =  230.82,    Prob value =          .00000 |
| Diagnostic: Log-L =    -42.2784, Restricted(b=0) Log-L =     -70.8467 |
|             LogAmemiyaPrCrt.=    1.013, Akaike Info. Crt.=      3.850 |
| Note, when restrictions are imposed, R-squared can be less than zero. |
| F[ 3,    18] for the restrictions =      5.1627, Prob =   .0095       |
| Autocorrel: Durbin-Watson Statistic =   1.12383,   Rho =       .43808 |
+----------------------------------------------------------------------+
+---------+-------------+----------------+--------+---------+----------+
|Variable | Coefficient  | Standard Error |t-ratio |P[|T|>t] | Mean of X|
+---------+-------------+----------------+--------+---------+----------+
 Constant    -2.211829436       1.9004224   -1.164   .2597
 PRE          1.520632737       .10008855   15.193   .0000     18.695652
 CLASS1        .0000000000 ........(Fixed Parameter)........   .21739130
 CLASS2   -.4440892099E-15........(Fixed Parameter)........   .30434783
 CLASS3   -.4440892099E-15........(Fixed Parameter)........   .26086957
```

```
 (Note: E+nn or E-nn means multiply by 10 to + or -nn power.)

--> Sample; 1-23$
--> Regress ; Lhs =post; Rhs =one, pre$

+-----------------------------------------------------------------------+
| Ordinary    least squares regression    Weighting variable = none     |
| Dep. var. = POST     Mean=   26.21739130    , S.D.=   5.384797808     |
| Model size: Observations =      23, Parameters =   2, Deg.Fr.=     21 |
| Residuals:  Sum of squares= 53.19669876    , Std.Dev.=        1.59160 |
| Fit:        R-squared=  .916608, Adjusted R-squared =          .91264 |
| Model test: F[  1,     21] =  230.82,    Prob value =          .00000 |
| Diagnostic: Log-L =    -42.2784, Restricted(b=0) Log-L =     -70.8467 |
|             LogAmemiyaPrCrt.=    1.013, Akaike Info. Crt.=      3.850 |
| Autocorrel: Durbin-Watson Statistic =   1.12383,   Rho =       .43808 |
+-----------------------------------------------------------------------+
+---------+-------------+----------------+--------+---------+----------+
|Variable | Coefficient  | Standard Error |t-ratio |P[|T|>t] | Mean of X|
+---------+-------------+----------------+--------+---------+----------+
 Constant    -2.211829436       1.9004224   -1.164   .2575
 PRE          1.520632737       .10008855   15.193   .0000     18.695652

--> Calc    ; SSall = Sumsqdev $
--> Sample; 1-5$
--> Regress ; Lhs =post; Rhs =one, pre$

+-----------------------------------------------------------------------+
| Ordinary    least squares regression    Weighting variable = none     |
| Dep. var. = POST     Mean=   30.00000000    , S.D.=   3.000000000     |
| Model size: Observations =       5, Parameters =   2, Deg.Fr.=      3 |
| Residuals:  Sum of squares= .2567567568    , Std.Dev.=         .29255 |
| Fit:        R-squared=  .992868, Adjusted R-squared =          .99049 |
| Model test: F[  1,      3] =  417.63,    Prob value =          .00026 |
| Diagnostic: Log-L =       .3280, Restricted(b=0) Log-L =     -12.0299 |
|             LogAmemiyaPrCrt.=   -2.122, Akaike Info. Crt.=       .669 |
| Autocorrel: Durbin-Watson Statistic =   2.19772,   Rho =      -.09886 |
+-----------------------------------------------------------------------+
+---------+-------------+----------------+--------+---------+----------+
|Variable | Coefficient  | Standard Error |t-ratio |P[|T|>t] | Mean of X|
+---------+-------------+----------------+--------+---------+----------+
 Constant    -2.945945946       1.6174496   -1.821   .1661
 PRE          1.554054054   .76044788E-01   20.436   .0003     21.200000
 (Note: E+nn or E-nn means multiply by 10 to + or -nn power.)

--> Calc    ; SS1 = Sumsqdev $
--> Sample; 6-12$
--> Regress ; Lhs =post; Rhs =one, pre$

+-----------------------------------------------------------------------+
| Ordinary    least squares regression    Weighting variable = none     |
| Dep. var. = POST     Mean=   27.28571429    , S.D.=   5.023753103     |
| Model size: Observations =       7, Parameters =   2, Deg.Fr.=      5 |
| Residuals:  Sum of squares= 7.237132353    , Std.Dev.=        1.20309 |
| Fit:        R-squared=  .952208, Adjusted R-squared =          .94265 |
| Model test: F[  1,      5] =   99.62,    Prob value =          .00017 |
| Diagnostic: Log-L =    -10.0492, Restricted(b=0) Log-L =     -20.6923 |
|             LogAmemiyaPrCrt.=     .621, Akaike Info. Crt.=      3.443 |
| Autocorrel: Durbin-Watson Statistic =   1.50037,   Rho =       .24982 |
+-----------------------------------------------------------------------+
+---------+-------------+----------------+--------+---------+----------+
```

```
|Variable | Coefficient  | Standard Error |t-ratio |P[|T|>t] | Mean of X|
+---------+--------------+----------------+--------+---------+----------+
 Constant     .6268382353       2.7094095     .231   .8262
 PRE          1.362132353       .13647334    9.981   .0002     19.571429

--> Calc    ; SS2 = Sumsqdev $
--> Sample; 13-18$
--> Regress ; Lhs =post; Rhs =one, pre$

+-----------------------------------------------------------------------+
| Ordinary    least squares regression    Weighting variable = none     |
| Dep. var. = POST     Mean=   25.66666667    , S.D.=   5.278888772     |
| Model size: Observations =       6, Parameters =   2, Deg.Fr.=      4 |
| Residuals:  Sum of squares= 8.081250000    , Std.Dev.=        1.42138 |
| Fit:        R-squared=  .942001, Adjusted R-squared =          .92750 |
| Model test: F[  1,      4] =   64.97,    Prob value =          .00129 |
| Diagnostic: Log-L =     -9.4070, Restricted(b=0) Log-L =     -17.9490 |
|             LogAmemiyaPrCrt.=     .991, Akaike Info. Crt.=      3.802 |
| Autocorrel: Durbin-Watson Statistic =   1.51997,   Rho =       .24001 |
+-----------------------------------------------------------------------+
+---------+--------------+----------------+--------+---------+----------+
|Variable | Coefficient  | Standard Error |t-ratio |P[|T|>t] | Mean of X|
+---------+--------------+----------------+--------+---------+----------+
 Constant    -1.525000000       3.4231291    -.445   .6790
 PRE          1.568750000       .19463006    8.060   .0013     17.333333

--> Calc    ; SS3 = Sumsqdev $
--> Sample; 19-23$
--> Regress ; Lhs =post; Rhs =one, pre$

+-----------------------------------------------------------------------+
| Ordinary    least squares regression    Weighting variable = none     |
| Dep. var. = POST     Mean=   21.60000000    , S.D.=   5.549774770     |
| Model size: Observations =       5, Parameters =   2, Deg.Fr.=      3 |
| Residuals:  Sum of squares= 8.924731183    , Std.Dev.=        1.72479 |
| Fit:        R-squared=  .927559, Adjusted R-squared =          .90341 |
| Model test: F[  1,      3] =   38.41,    Prob value =          .00846 |
| Diagnostic: Log-L =     -8.5432, Restricted(b=0) Log-L =     -15.1056 |
|             LogAmemiyaPrCrt.=    1.427, Akaike Info. Crt.=      4.217 |
| Autocorrel: Durbin-Watson Statistic =    .82070,   Rho =       .58965 |
+-----------------------------------------------------------------------+
+---------+--------------+----------------+--------+---------+----------+
|Variable | Coefficient  | Standard Error |t-ratio |P[|T|>t] | Mean of X|
+---------+--------------+----------------+--------+---------+----------+
 Constant    -7.494623656       4.7572798   -1.575   .2132
 PRE          1.752688172       .28279093    6.198   .0085     16.600000

--> Calc    ; SS4 = Sumsqdev $
--> Calc;List ;F=((SSall-(SS1+SS2+SS3+SS4))/(3*2)) / ((SS1+SS2+SS3+SS4)/(23-4*2))$
    F       =  .29282633057790450D+01
```

The structural test across all classes is

$$H_0 : \boldsymbol{\beta}_1 = \boldsymbol{\beta}_2 = \ldots = \boldsymbol{\beta}_4 \text{ and } H_a : \boldsymbol{\beta}'s \text{ are not equal}$$

$$F = \frac{(\text{Re}\,sSS_r - \text{Re}\,sSS_u)/2(J-1K)}{\text{Re}\,sSS_u/(n-JK)}$$

Because the calculated $F = 2.92$, and the critical F (Prob of Type I error =0.05, df1=6, df2=15) = 2.79, we reject the null hypothesis and conclude that at least one class is significantly different from another, allowing the slope on pre-course test score to vary from one class to another. That is, the class in which a student is enrolled is important because of a change in slope and/or the intercept.

HETEROSCEDASTICITY

To adjust for either heteroscedasticity across individual observations or a common error term within groups but not across groups the "hetro" and "cluster" command can be added to the standard "regress" command in LIMDEP in the following manner:

Skip

Regress; Lhs= post; Rhs= one, pre, class1, class2, class3$

Regress; Lhs= post; Rhs= one, pre, class1, class2, class3
;hetro $

Create; Class = class1+2*class2+3*class3+4*class4$
Regress ; Lhs= post; Rhs= one, pre, class1, class2, class3
;cluster=class $

where the "class" variable is created to name the classes 1, 2, 3 and 4 to enable their identification in the "cluster" command.

The resulting LIMDEP output shows a marked increase in the significance of the individual group effects, as reflected in their respective *p*-values.

```
--> RESET
--> READ;FILE="C:\Documents and Settings\beckerw\My Documents\WKPAPERS\NCEE -...
--> skip
--> Regress; Lhs= post; Rhs= one, pre, class1, class2, class3$

 ********************************************************************
 * NOTE: Deleted      1 observations with missing data. N is now     23 *
 ********************************************************************

le One\post-pre.xls
+--------------------------------------------------------------------+
| Ordinary    least squares regression    Weighting variable = none  |
| Dep. var. = POST     Mean=   26.21739130    , S.D.=   5.384797808  |
```

```
| Model size: Observations =      23, Parameters =   5, Deg.Fr.=     18 |
| Residuals:  Sum of squares= 28.59332986    , Std.Dev.=        1.26036 |
| Fit:        R-squared=  .955177, Adjusted R-squared =          .94522 |
| Model test: F[  4,     18] =   95.89,    Prob value =          .00000 |
| Diagnostic: Log-L =    -35.1389, Restricted(b=0) Log-L =     -70.8467 |
|             LogAmemiyaPrCrt.=     .660, Akaike Info. Crt.=      3.490 |
| Autocorrel: Durbin-Watson Statistic =   1.72443,   Rho =       .13779 |
+----------------------------------------------------------------------+
+---------+--------------+----------------+--------+---------+----------+
|Variable | Coefficient  | Standard Error |t-ratio |P[|T|>t] | Mean of X|
+---------+--------------+----------------+--------+---------+----------+
 Constant    -3.585879292       1.6459223   -2.179   .0429 le One\post-pre.xl
 PRE          1.517221644   .93156695E-01   16.287   .0000     18.695652
 CLASS1       1.420780437       .90500685    1.570   .1338     .21739130
 CLASS2       1.177398543       .78819907    1.494   .1526     .30434783
 CLASS3       2.954037461       .76623994    3.855   .0012     .26086957
 (Note: E+nn or E-nn means multiply by 10 to + or -nn power.)
le One\post-pre.xls


--> Regress; Lhs= post; Rhs= one, pre, class1, class2, class3
    ;hetro $

 ******************************************************************************
 * NOTE: Deleted      1 observations with missing data. N is now     23 *
 ******************************************************************************

le One\post-pre.xls
+----------------------------------------------------------------------+
| Ordinary    least squares regression    Weighting variable = none     |
| Dep. var. = POST     Mean=   26.21739130    , S.D.=   5.384797808     |
| Model size: Observations =      23, Parameters =   5, Deg.Fr.=     18 |
| Residuals:  Sum of squares= 28.59332986    , Std.Dev.=        1.26036 |
| Fit:        R-squared=  .955177, Adjusted R-squared =          .94522 |
| Model test: F[  4,     18] =   95.89,    Prob value =          .00000 |
| Diagnostic: Log-L =    -35.1389, Restricted(b=0) Log-L =     -70.8467 |
|             LogAmemiyaPrCrt.=     .660, Akaike Info. Crt.=      3.490 |
| Autocorrel: Durbin-Watson Statistic =   1.72443,   Rho =       .13779 |
| Results Corrected for heteroskedasticity                             |
| Breusch - Pagan chi-squared =     4.0352, with   4 degrees of freedom |
+----------------------------------------------------------------------+
+---------+--------------+----------------+--------+---------+----------+
|Variable | Coefficient  | Standard Error |t-ratio |P[|T|>t] | Mean of X|
+---------+--------------+----------------+--------+---------+----------+
 Constant    -3.585879292       1.5096560   -2.375   .0289 le One\post-pre.xl
 PRE          1.517221644   .72981808E-01   20.789   .0000     18.695652
 CLASS1       1.420780437       .67752835    2.097   .0504     .21739130
 CLASS2       1.177398543       .72249740    1.630   .1206     .30434783
 CLASS3       2.954037461       .80582075    3.666   .0018     .26086957
 (Note: E+nn or E-nn means multiply by 10 to + or -nn power.)
le One\post-pre.xls

--> Create; Class = class1+2*class2+3*class3+4*class4$
--> Regress ; Lhs= post; Rhs= one, pre, class1, class2, class3
    ;cluster=class $

 ******************************************************************************
 * NOTE: Deleted      1 observations with missing data. N is now     23 *
 ******************************************************************************

le One\post-pre.xls
+----------------------------------------------------------------------+
| Ordinary    least squares regression    Weighting variable = none     |
| Dep. var. = POST     Mean=   26.21739130    , S.D.=   5.384797808     |
| Model size: Observations =      23, Parameters =   5, Deg.Fr.=     18 |
```

```
| Residuals:  Sum of squares= 28.59332986    , Std.Dev.=        1.26036 |
| Fit:        R-squared=  .955177, Adjusted R-squared =         .94522 |
| Model test: F[  4,     18] =   95.89,    Prob value =         .00000 |
| Diagnostic: Log-L =    -35.1389, Restricted(b=0) Log-L =     -70.8467 |
|             LogAmemiyaPrCrt.=     .660, Akaike Info. Crt.=      3.490 |
| Autocorrel: Durbin-Watson Statistic =   1.72443,   Rho =       .13779 |
+----------------------------------------------------------------------+
+----------------------------------------------------------------------+
| Covariance matrix for the model is adjusted for data clustering.     |
| Sample of     23 observations contained      4 clusters defined by   |
| variable CLASS    which identifies by a value a cluster ID.          |
+---------+--------------+----------------+--------+---------+----------+
|Variable | Coefficient  | Standard Error |t-ratio |P[|T|>t] | Mean of X|
+---------+--------------+----------------+--------+---------+----------+
 Constant    -3.585879292       1.5875538   -2.259   .0365 le One\post-pre.xl
 PRE          1.517221644   .95635769E-01   15.865   .0000     18.695652
 CLASS1       1.420780437       .43992454    3.230   .0046     .21739130
 CLASS2       1.177398543       .28417486    4.143   .0006     .30434783
 CLASS3       2.954037461   .70132897E-01   42.121   .0000     .26086957
 (Note: E+nn or E-nn means multiply by 10 to + or -nn power.)
le One\post-pre.xls
```

ESTIMATING PROBIT MODELS IN LIMDEP

Often variables need to be transformed or created within a computer program to perform the desired analysis. To demonstrate the process and commands in LIMDEP, start with the Becker and Power's data that have been or can be read into LIMDEP as shown earlier. After reading the data into LIMDEP the first task is to recode the qualitative data into appropriate dummies.

A2 contains a range of values representing various classes of institutions. These are recoded via the "recode" command, where A2 is set equal to 1 for doctoral institutions (100/199), 2 for comprehensive or master's degree granting institutions (200/299), 3 for liberal arts colleges (300/399) and 4 for two-year colleges (400/499) . The "create" command is then used to create 1 and 0 bivariate variables for each of these institutions of post-secondary education:

```
recode; a2; 100/199 = 1; 200/299 = 2; 300/399 = 3; 400/499 =4$
create; doc=a2=1; comp=a2=2; lib=a2=3; twoyr=a2=4$
```

As should be apparent, this syntax says if a2 has a value between 100 and 199 recode it to be 1. If a2 has a value between 200 and 299 recode it to be 2 and so on. Next, create a variable called "doc" and if a2=1, then set doc=1 and for any other value of a2 let doc=0. Create a variable called "comp" and if a2=2, then set comp=1 and for any other value of a2 let comp=0, and so on.

Next 1 - 0 bivariates are created to show whether the instructor had a PhD degree and where the student got a positive score on the postTUCE:

```
create; phd=dj=3; final=cc>0$
```

To allow for quadratic forms in teacher experiences and class size the following variables are created:

```
create; dmsq=dm^2; hbsq=hb^2$
```

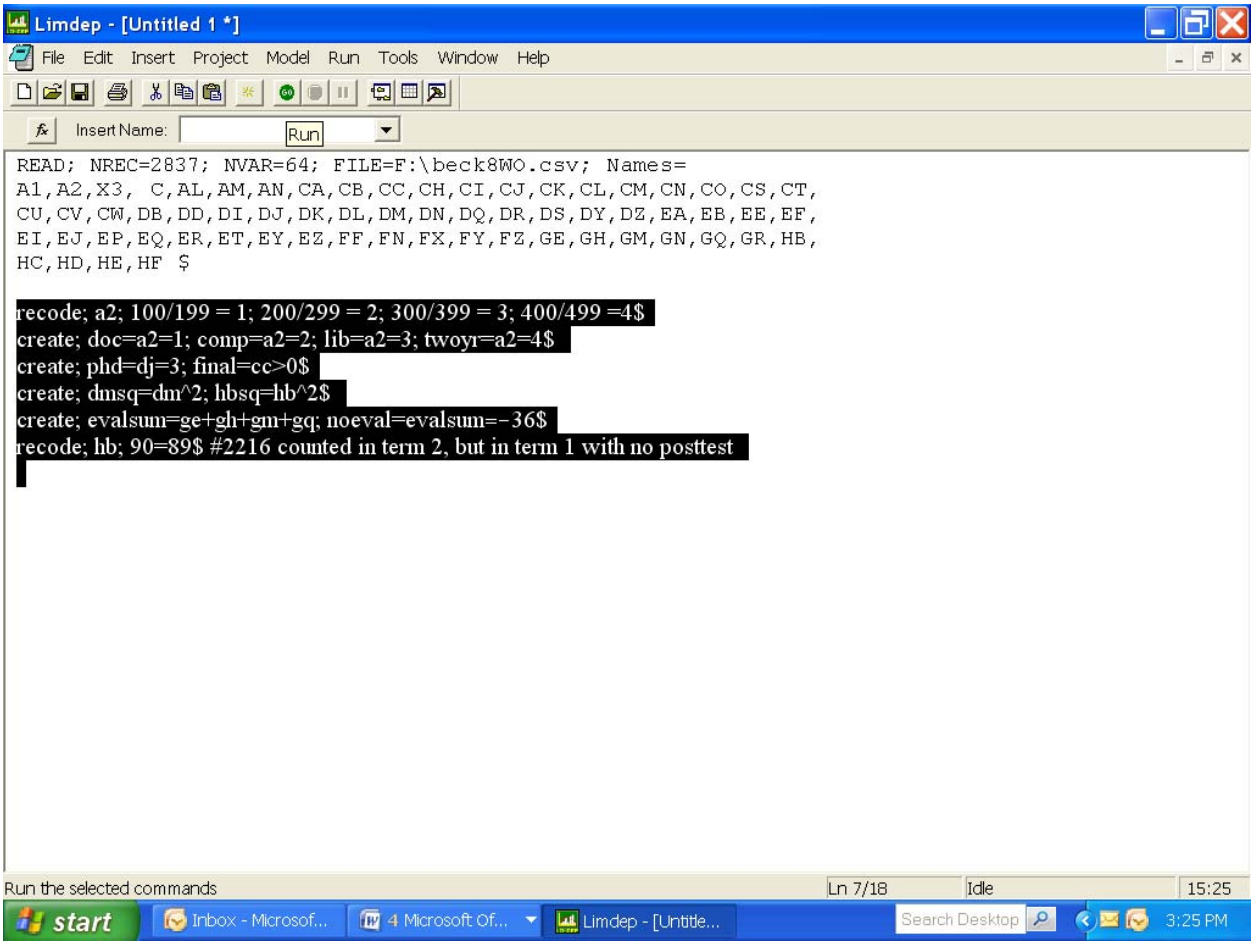
In this data set, as can be seen in the descriptive statistics (DSTAT), all missing values were coded −9. Thus, adding together some of the responses to the student evaluations gives information on whether a student actually completed an evaluation. For example, if the sum of ge, gh, gm, and gq equals −36, we know that the student did not complete a student evaluation in a meaningful way. A dummy variable to reflect this fact is then created by:

```
create; evalsum=ge+gh+gm+gq; noeval=evalsum=−36$
```

Finally, from the TUCE developer it is known that student number 2216 was counted in term 2 but was in term 1 but no postTUCE was taken. This error is corrected with the following command:

```
recode; hb; 90=89$  #2216 counted in term 2, but in term 1 with no posttest
```

These "create" and "recode" commands can be entered into LIMDEP as a block, highlighted and run with the "Go" button. Notice, also, that descriptive statements can be written after the "$" as a reminder or for later justification or reference as to why the command was included.



One of the things of interest to Becker and Powers was whether class size at the beginning or end of the term influenced whether a student completed the postTUCE. This can be assessed by fitting a probit model to the 1 – 0 discrete dependent variable "final." To do this, however, we must make sure that there are no missing data on the variables to be included as regressors. In this data set, all missing values were coded −9. LIMDEP's "reject" command can be employed to remove all records with a −9 value. The "probit" command is used to invoke a maximum likelihood estimation with the following syntax:

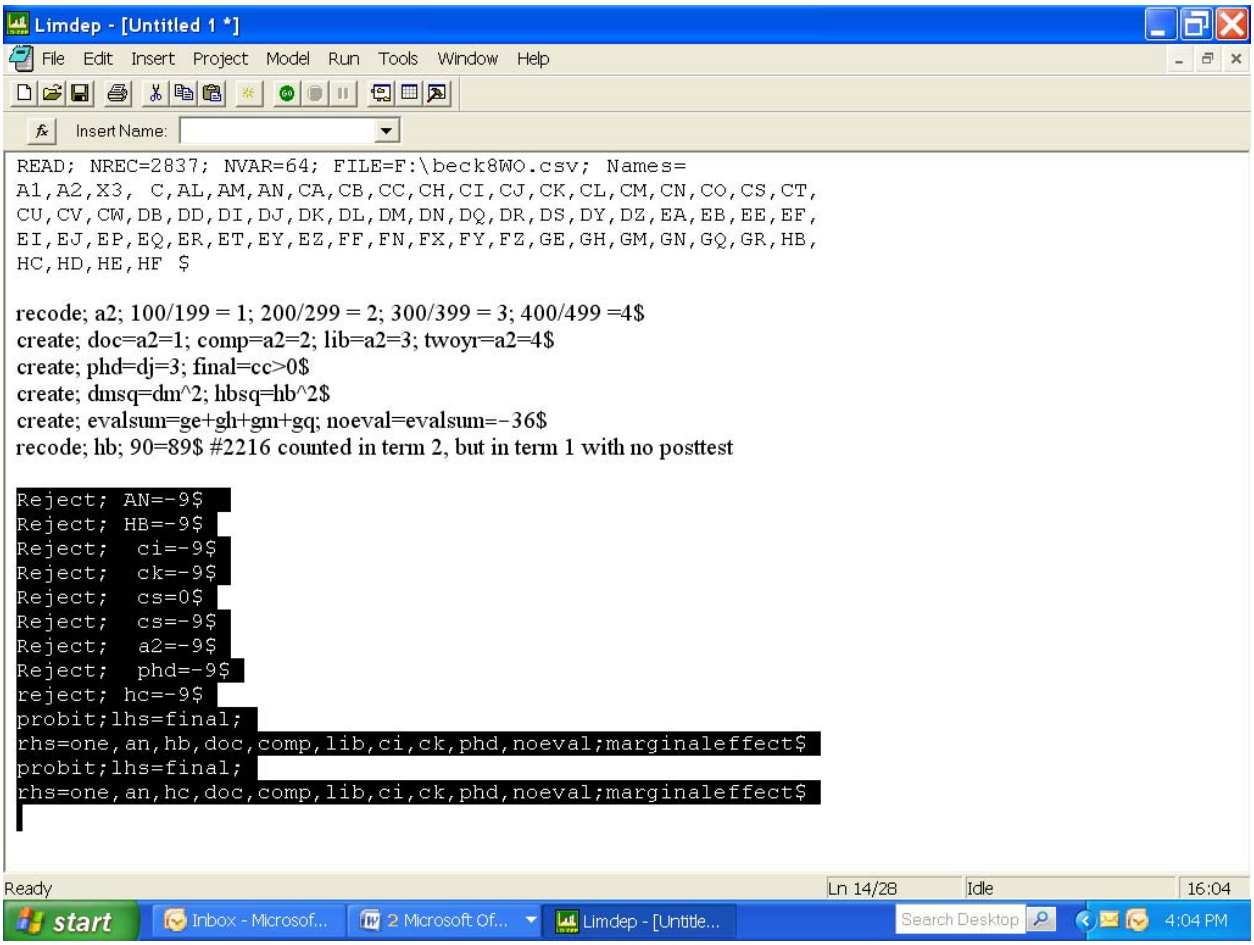
Probit; Lhs=???; rhs=one, ???; marginaleffect$

where the addition of the "marginaleffect" tells LIMDEP to calculate marginal effects regardless of whether the explanatory variable is or is not continuous. The commands to be entered into the Text/Command Document are then

Reject; AN=-9$
Reject; HB=-9$
Reject; ci=-9$
Reject; ck=-9$

```
Reject;  cs=0$
Reject;  cs=-9$
Reject;  a2=-9$
Reject;  phd=-9$
reject; hc=-9$
probit;lhs=final;
rhs=one,an,hb,doc,comp,lib,ci,ck,phd,noeval;marginaleffect$
probit;lhs=final;
rhs=one,an,hc,doc,comp,lib,ci,ck,phd,noeval;marginaleffect$
```

which upon highlighting and pressing the Go button yields the output for these two probit models.




```
--> probit;lhs=final;
    rhs=one,an,hb,doc,comp,lib,ci,ck,phd,noeval;marginaleffect$
Normal exit from iterations. Exit status=0.

+---------------------------------------------+
| Binomial Probit Model                       |
```

```
| Maximum Likelihood Estimates                  |
| Model estimated: May 05, 2008 at 04:07:02PM.|
| Dependent variable                FINAL      |
| Weighting variable                 None      |
| Number of observations             2587      |
| Iterations completed                  6      |
| Log likelihood function       -822.7411      |
| Restricted log likelihood     -1284.216      |
| Chi squared                    922.9501      |
| Degrees of freedom                    9      |
| Prob[ChiSqd > value] =         .0000000      |
| Hosmer-Lemeshow chi-squared =  26.06658      |
| P-value=  .00102 with deg.fr. =       8      |
+----------------------------------------------+
+---------+--------------+----------------+--------+---------+----------+
|Variable | Coefficient  | Standard Error |b/St.Er.|P[|Z|>z] | Mean of X|
+---------+--------------+----------------+--------+---------+----------+
          Index function for probability
 Constant     .9953497702       .24326247    4.092   .0000
 AN        .2203899720E-01  .94751772E-02    2.326   .0200     10.596830
 HB       -.4882560519E-02  .19241005E-02   -2.538   .0112     55.558949
 DOC          .9757147902       .14636173    6.666   .0000     .31774256
 COMP         .4064945318       .13926507    2.919   .0035     .41785852
 LIB          .5214436028       .17664590    2.952   .0032     .13567839
 CI           .1987315042   .91686501E-01    2.168   .0302     1.2311558
 CK        .8778999306E-01       .13428742     .654   .5133     .91998454
 PHD         -.1335050091       .10303166   -1.296   .1951     .68612292
 NOEVAL      -1.930522400   .72391102E-01  -26.668   .0000     .29068419
 (Note: E+nn or E-nn means multiply by 10 to + or -nn power.)


+-----------------------------------------+
| Partial derivatives of E[y] = F[*]   with |
| respect to the vector of characteristics. |
| They are computed at the means of the Xs. |
| Observations used for means are All Obs.  |
+-----------------------------------------+
+---------+--------------+----------------+--------+---------+----------+
|Variable | Coefficient  | Standard Error |b/St.Er.|P[|Z|>z] | Mean of X|
+---------+--------------+----------------+--------+---------+----------+
          Index function for probability
 Constant     .1977242134   .48193408E-01    4.103   .0000
 AN         .4378002101E-02  .18769275E-02    2.333   .0197     10.596830
 HB        -.9699107460E-03  .38243741E-03   -2.536   .0112     55.558949
          Marginal effect for dummy variable is P|1 - P|0.
 DOC          .1595047130   .20392136E-01    7.822   .0000     .31774256
          Marginal effect for dummy variable is P|1 - P|0.
 COMP       .7783344522E-01  .25881201E-01    3.007   .0026     .41785852
          Marginal effect for dummy variable is P|1 - P|0.
 LIB        .8208261358E-01  .21451464E-01    3.826   .0001     .13567839
 CI         .3947761030E-01  .18186048E-01    2.171   .0299     1.2311558
          Marginal effect for dummy variable is P|1 - P|0.
 CK         .1820482750E-01  .29016989E-01     .627   .5304     .91998454
          Marginal effect for dummy variable is P|1 - P|0.
 PHD       -.2575430653E-01  .19325466E-01   -1.333   .1826     .68612292
          Marginal effect for dummy variable is P|1 - P|0.
 NOEVAL       -.5339850032   .19586185E-01  -27.263   .0000     .29068419
 (Note: E+nn or E-nn means multiply by 10 to + or -nn power.)

+---------------------------------------+
| Fit Measures for Binomial Choice Model |
| Probit   model for variable FINAL      |
+---------------------------------------+
| Proportions P0= .197140   P1= .802860  |
| N =    2587 N0=     510   N1=    2077  |
```

```
| LogL =  -822.74107 LogL0 = -1284.2161  |
| Estrella = 1-(L/L0)^(-2L0/n) = .35729  |
+----------------------------------------+
|     Efron |  McFadden  |  Ben./Lerman  |
|    .39635 |    .35934  |       .80562  |
|    Cramer | Veall/Zim. |     Rsqrd_ML  |
|    .38789 |    .52781  |       .30006  |
+----------------------------------------+
| Information  Akaike I.C. Schwarz I.C.  |
| Criteria         .64379    1724.06468  |
+----------------------------------------+
Frequencies of actual & predicted outcomes
Predicted outcome has maximum probability.
Threshold value for predicting Y=1 = .5000
            Predicted
------  ----------  +  -----
Actual      0    1  |  Total
------  ----------  +  -----
  0       342  168  |    510
  1       197 1880  |   2077
------  ----------  +  -----
Total     539 2048  |   2587
```

```
--> probit;lhs=final;
    rhs=one,an,hc,doc,comp,lib,ci,ck,phd,noeval;marginaleffect$
Normal exit from iterations. Exit status=0.

+---------------------------------------------+
| Binomial Probit Model                       |
| Maximum Likelihood Estimates                |
| Model estimated: May 05, 2008 at 04:07:03PM.|
| Dependent variable                FINAL     |
| Weighting variable                 None     |
| Number of observations             2587     |
| Iterations completed                  6     |
| Log likelihood function       -825.9472     |
| Restricted log likelihood     -1284.216     |
| Chi squared                    916.5379     |
| Degrees of freedom                    9     |
| Prob[ChiSqd > value] =          .0000000    |
| Hosmer-Lemeshow chi-squared =  22.57308     |
| P-value=  .00396 with deg.fr. =       8     |
+---------------------------------------------+
+---------+--------------+----------------+--------+---------+----------+
|Variable | Coefficient  | Standard Error |b/St.Er.|P[|Z|>z] | Mean of X|
+---------+--------------+----------------+--------+---------+----------+
          Index function for probability
 Constant     .8712666323       .24117408    3.613   .0003
 AN        .2259549490E-01  .94553383E-02    2.390   .0169     10.596830
 HC        .1585898886E-03  .21039762E-02     .075   .9399     49.974874
 DOC          .8804040395       .14866411    5.922   .0000     .31774256
 COMP         .4596088640       .13798168    3.331   .0009     .41785852
 LIB          .5585267697       .17568141    3.179   .0015     .13567839
 CI           .1797199200   .90808055E-01    1.979   .0478     1.2311558
 CK        .1415663447E-01       .13332671     .106   .9154     .91998454
 PHD          -.2351326125       .10107423   -2.326   .0200     .68612292
 NOEVAL      -1.928215642   .72363621E-01  -26.646   .0000     .29068419
 (Note: E+nn or E-nn means multiply by 10 to + or -nn power.)


+-----------------------------------------+
| Partial derivatives of E[y] = F[*]   with |
| respect to the vector of characteristics. |
| They are computed at the means of the Xs. |
| Observations used for means are All Obs.  |
```

```
+----------------------------------------+
+---------+-------------+----------------+--------+---------+----------+
|Variable | Coefficient  | Standard Error |b/St.Er.|P[|Z|>z] | Mean of X|
+---------+-------------+----------------+--------+---------+----------+
          Index function for probability
 Constant      .1735365132   .47945637E-01    3.619   .0003
 AN          .4500509092E-02  .18776909E-02    2.397   .0165     10.596830
 HC          .3158750180E-04  .41902052E-03     .075   .9399     49.974874
          Marginal effect for dummy variable is P|1 - P|0.
 DOC           .1467543687   .21319420E-01    6.884   .0000     .31774256
          Marginal effect for dummy variable is P|1 - P|0.
 COMP        .8785901674E-01  .25536388E-01    3.441   .0006     .41785852
          Marginal effect for dummy variable is P|1 - P|0.
 LIB         .8672357482E-01  .20661637E-01    4.197   .0000     .13567839
 CI          .3579612385E-01  .18068050E-01    1.981   .0476     1.2311558
          Marginal effect for dummy variable is P|1 - P|0.
 CK          .2839467767E-02  .26927626E-01     .105   .9160     .91998454
          Marginal effect for dummy variable is P|1 - P|0.
 PHD        -.4448632109E-01  .18193388E-01   -2.445   .0145     .68612292
          Marginal effect for dummy variable is P|1 - P|0.
 NOEVAL       -.5339710749   .19569243E-01  -27.286   .0000     .29068419
 (Note: E+nn or E-nn means multiply by 10 to + or -nn power.)

+----------------------------------------+
| Fit Measures for Binomial Choice Model |
| Probit   model for variable FINAL      |
+----------------------------------------+
| Proportions P0= .197140   P1= .802860  |
| N =    2587 N0=     510   N1=    2077  |
| LogL =  -825.94717 LogL0 = -1284.2161  |
| Estrella = 1-(L/L0)^(-2L0/n) = .35481  |
+----------------------------------------+
|     Efron |  McFadden  |  Ben./Lerman  |
|    .39186 |     .35685 |       .80450  |
|    Cramer | Veall/Zim. |     Rsqrd_ML  |
|    .38436 |    .52510  |       .29833  |
+----------------------------------------+
| Information  Akaike I.C. Schwarz I.C.  |
| Criteria         .64627    1730.47688  |
+----------------------------------------+
Frequencies of actual & predicted outcomes
Predicted outcome has maximum probability.
Threshold value for predicting Y=1 = .5000
            Predicted
------  ----------  +  -----
Actual       0    1  |  Total
------  ----------  +  -----
  0        337  173  |    510
  1        192 1885  |   2077
------  ----------  +  -----
Total      529 2058  |   2587
```

For each of these two probits, the first block of coefficients are for the latent variable probit equation. The second block provides the marginal effects. The initial class size (hb) probit coefficient −0.004883, however, is highly significant with a two-tail *p*-value of 0.0112. On the other hand, the end-of-term class size (hc) probit coefficient (0.000159) is insignificant with a two-tail *p*-value of 0.9399.

The overall goodness of fit can be assessed in several ways. The easiest is the proportion of correct 0 and 1 predictions: For the first probit, using initial class size (hb) as an explanatory variable, the proportion of correct prediction is 0.859 = (342+1880)/2587. For the second probit, using end-of-term class size (hc) as an explanatory variable, the proportion of correct prediction is also 0.859 = (337+1885)/2587. The Chi-square (922.95, df =9) for the probit employing the initial class size is slightly higher than that for the end-of-term probit (916.5379, df =9) but they are both highly significant.

Finally, worth noting when using the "reject" command is that the record is not removed. It can be reactivated with the "include" command. Active and inactive status can be observed in LIMDEP's editor by the presence or lack of presence of chevrons (>>) next to the row number down the left-hand side of the display.

If you wish to save you work in LIMDEP you must make sure to save each of the files you want separately. Your Text/Command Document, data file, and output files must be saved individually in LIMDEP. There is no global saving of all three files.

CONCLUDING REMARKS

The goal of this hands-on component of this first of four modules was to enable users to get data into LIMDEP, create variables and run regressions on continuous and discrete variables; it was not to explain all of the statistics produced by computer output. For this an intermediate level econometrics textbook (such as Jeffrey Wooldridge, *Introductory Econometrics*) or advanced econometrics textbook such as (William Greene, *Econometric Analysis*) must be consulted.

REFERENCES

Becker, William E. and John Powers (2001). "Student Performance, Attrition, and Class Size Given Missing Student Data," *Economics of Education Review*, Vol. 20, August: 377-388.

Hilbe, Joseph M. (2006). "A Review of LIMDEP 9.0 and NLOGIT 4.0." *The American Statistician,* 60(May): 187-202.

MODULE ONE, PART THREE: READING DATA INTO STATA, CREATING AND RECODING VARIABLES, AND ESTIMATING AND TESTING MODELS IN STATA

This Part Three of Module One provides a cookbook-type demonstration of the steps required to read or import data into STATA. The reading of both small and large text and Excel files are shown though real data examples. The procedures to recode and create variables within STATA are demonstrated. Commands for least-squares regression estimation and maximum likelihood estimation of probit and logit models are provided. Consideration is given to analysis of variance and the testing of linear restrictions and structural differences, as outlined in Part One. (Parts Two and Four provide the LIMDEP and SAS commands for the same operations undertaken here in Part Three with STATA. For a review of STATA, version 7, see Kolenikov (2001).)

IMPORTING EXCEL FILES INTO STATA

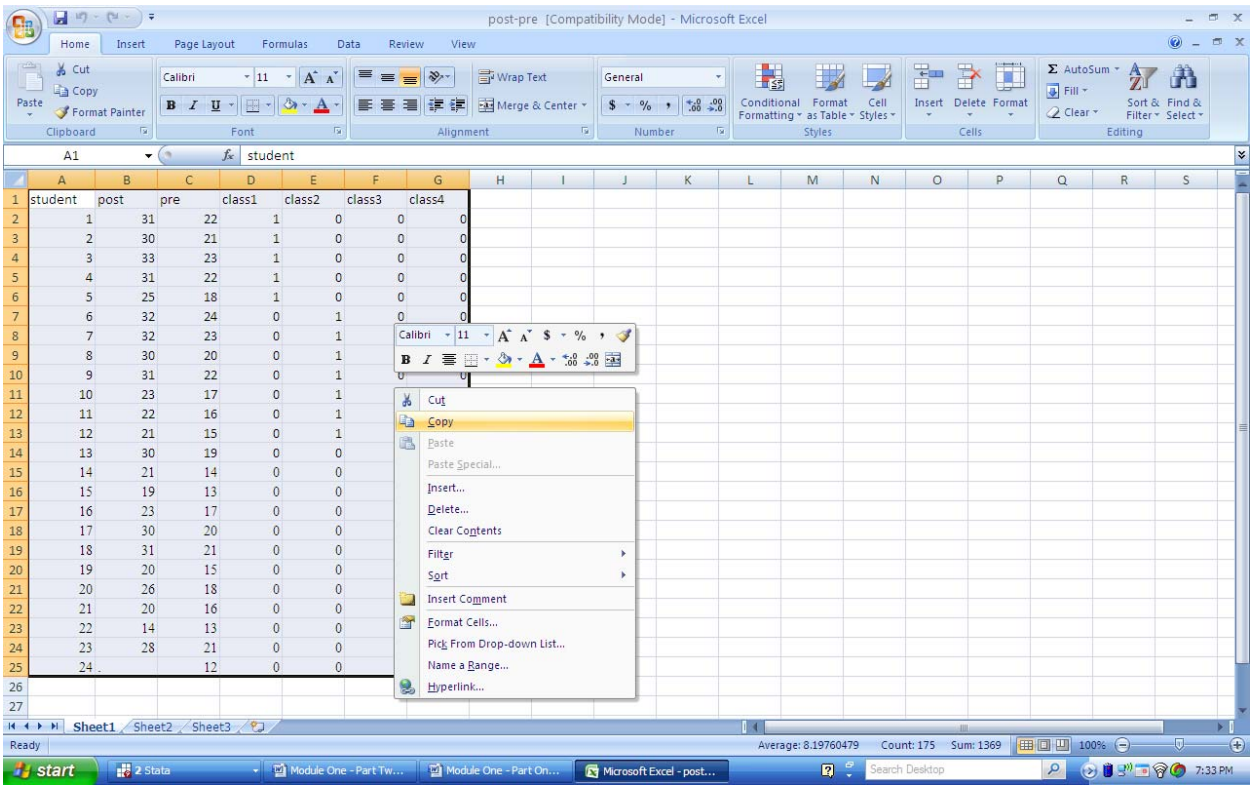
STATA can read data from many different formats. As an example of how to read data created in an Excel spreadsheet, consider the data from the Excel file "post-pre.xls," which consists of test scores for 24 students in four classes. The column titled "Student" identifies the 24 students by number, "post" provides each student's post-course test score, "pre" is each student's pre-course test score, and "class" identifies to which one of the four classes the students was assigned, e.g., class4 = 1 if student was in the fourth class and class4 = 0 if not. The "." in the post column for student 24 indicates that the student is missing a post-course test score.

To start, the file "post-pre.xls" must be downloaded and copied to your computer's hard drive. Unfortunately, STATA does not work with ".xls" data by default (i.e., there is no default "import" function or command to get ".xls" data into STATA's data editor); however, we can still transfer data from an Excel spreadsheet into STATA by copy and paste.[*] First, open the "post-pre.xls" file in Excel. The raw data are given below:

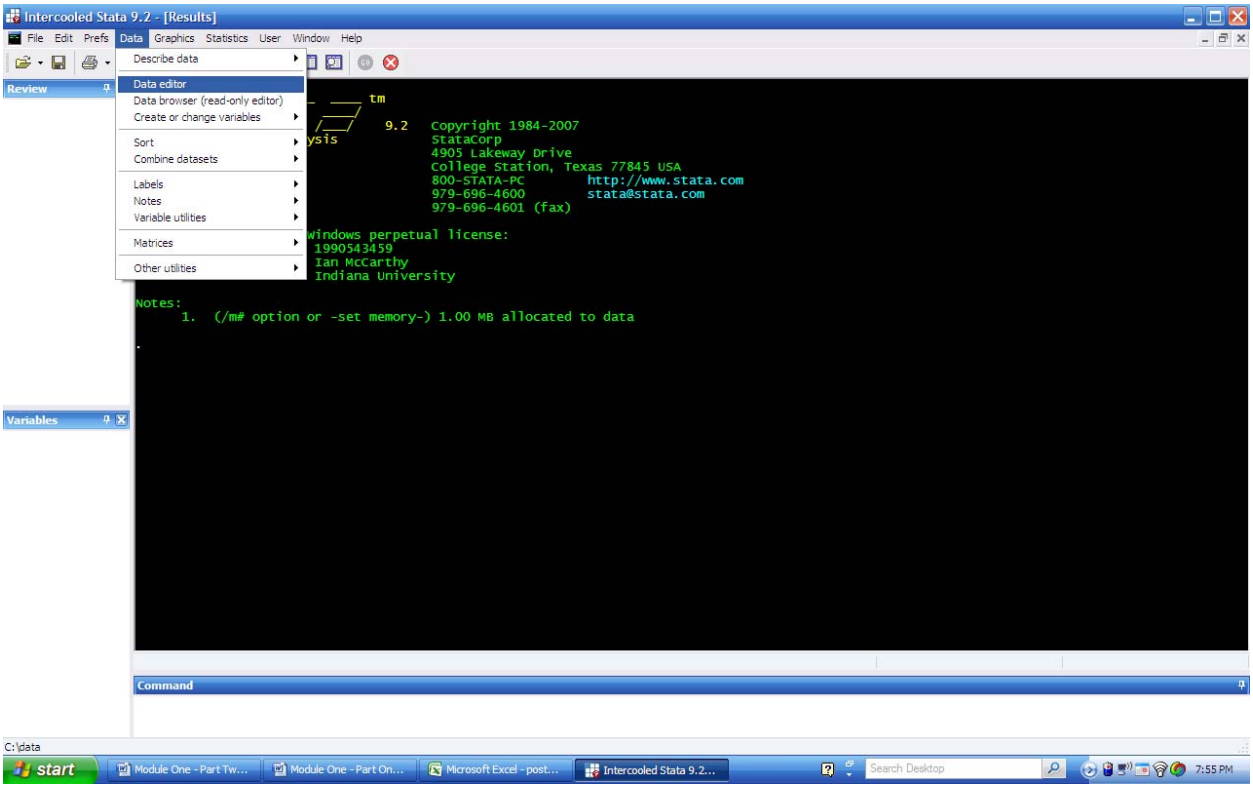
[*] See Appendix A for a description of Stat/Transfer, a program to convert data from one format to another.

student	post	pre	class1	class2	class3	class4
1	31	22	1	0	0	0
2	30	21	1	0	0	0
3	33	23	1	0	0	0
4	31	22	1	0	0	0
5	25	18	1	0	0	0
6	32	24	0	1	0	0
7	32	23	0	1	0	0
8	30	20	0	1	0	0
9	31	22	0	1	0	0
10	23	17	0	1	0	0
11	22	16	0	1	0	0
12	21	15	0	1	0	0
13	30	19	0	0	1	0
14	21	14	0	0	1	0
15	19	13	0	0	1	0
16	23	17	0	0	1	0
17	30	20	0	0	1	0
18	31	21	0	0	1	0
19	20	15	0	0	0	1
20	26	18	0	0	0	1
21	20	16	0	0	0	1
22	14	13	0	0	0	1
23	28	21	0	0	0	1
24	.	12	0	0	0	1

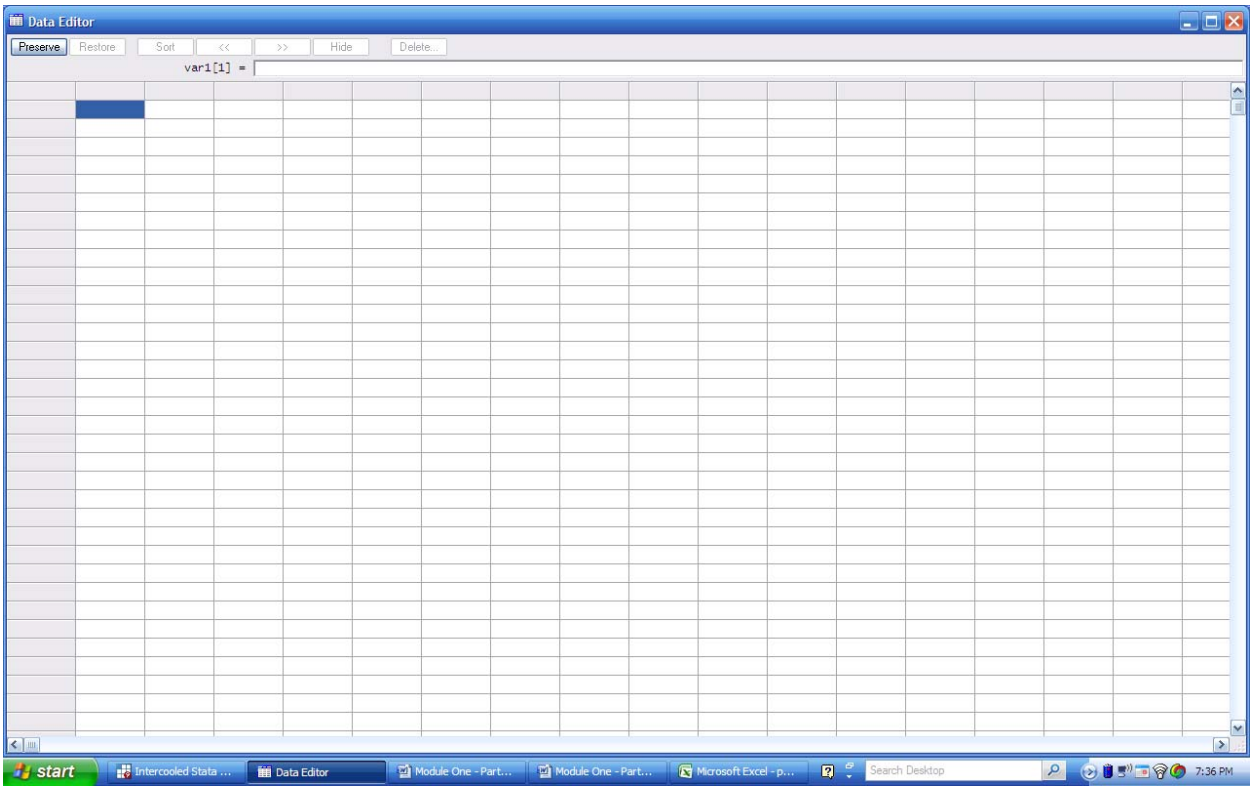
In Excel, highlight the appropriate cells, right-click on the highlighted area and click “copy”. Your screen should look something like:



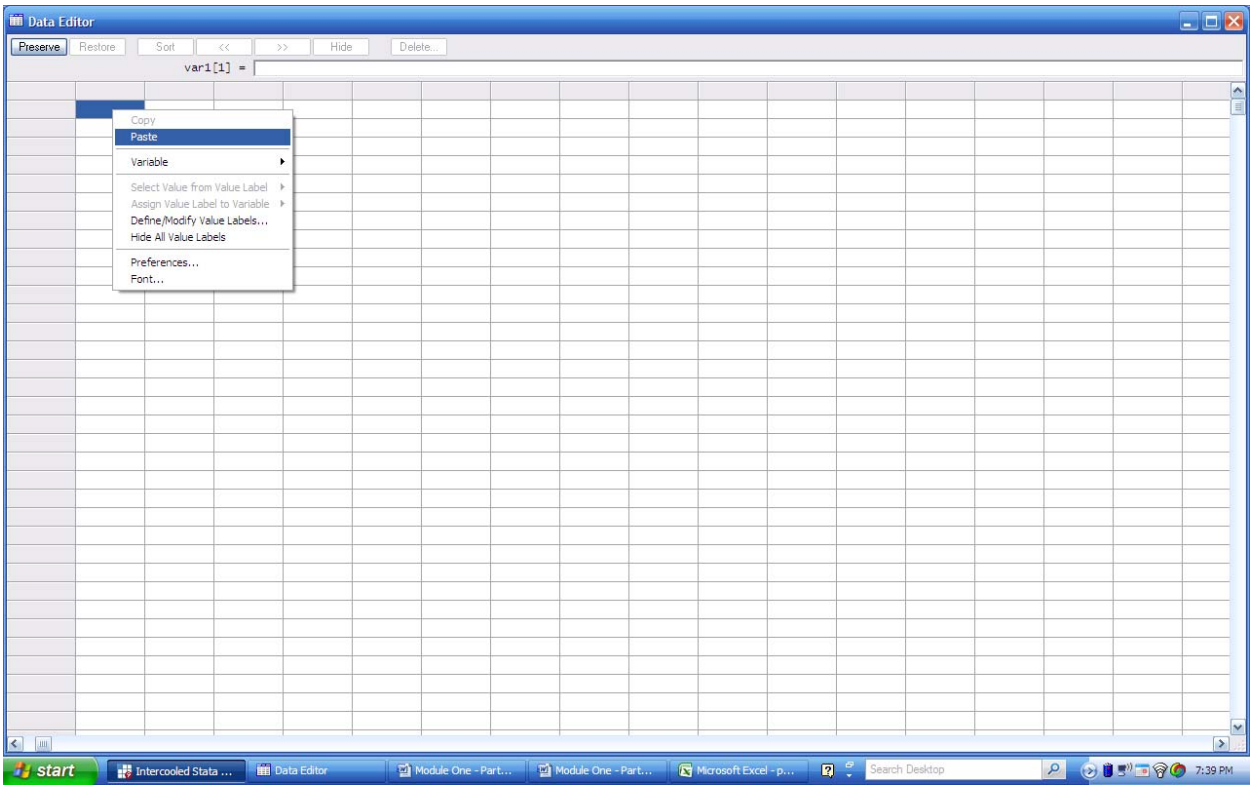
Then open STATA. Go to “Data”, and click on “Data Editor”:



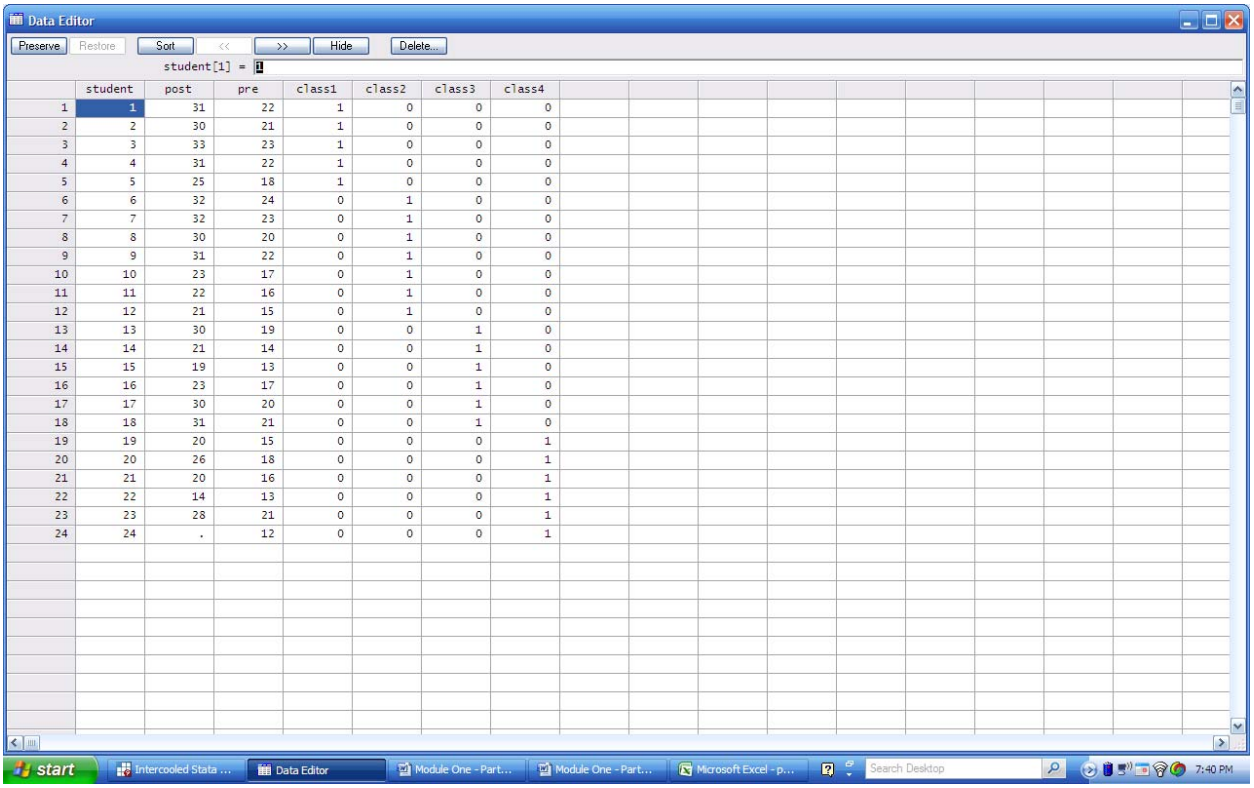
Clicking on “Data Editor” will yield the following screen:



From here, right-click on the highlighted cell and click “paste”:

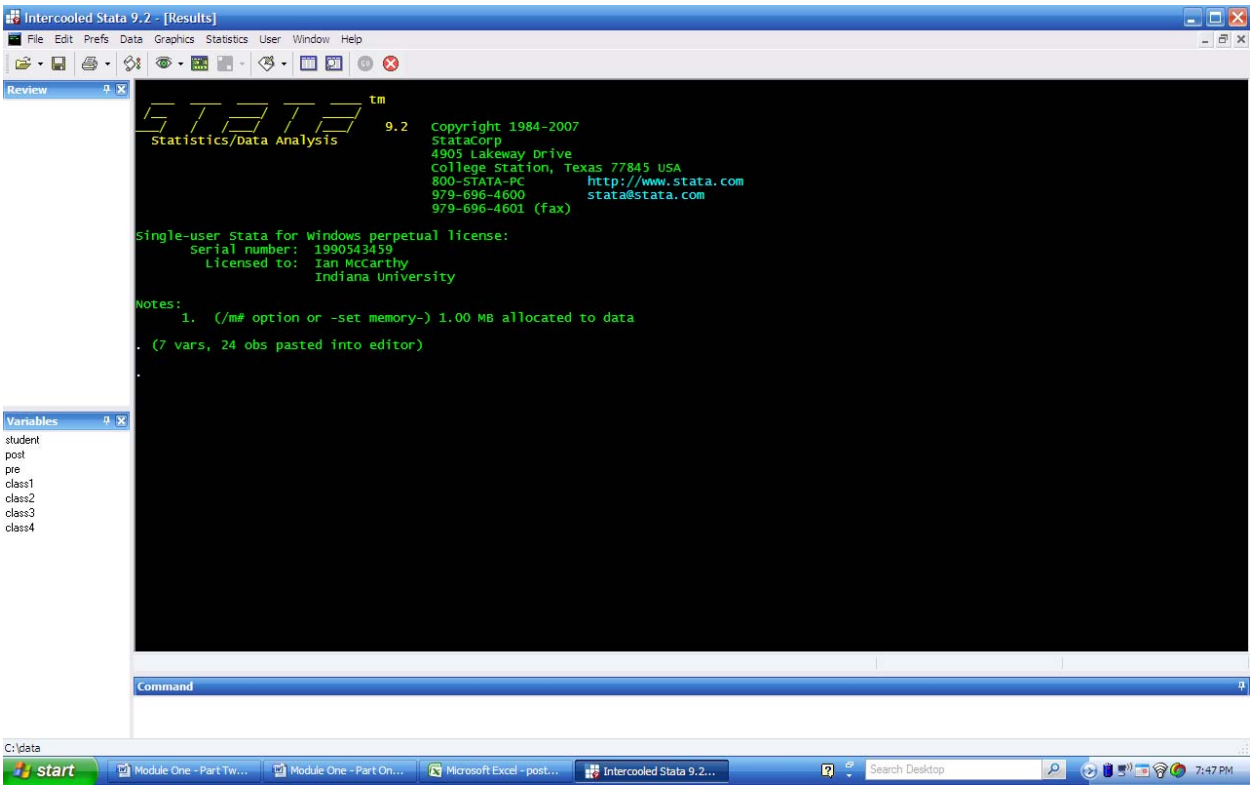


Your data is now in the STATA data editor, which yields the following screen:



	student	post	pre	class1	class2	class3	class4
1	1	31	22	1	0	0	0
2	2	30	21	1	0	0	0
3	3	33	23	1	0	0	0
4	4	31	22	1	0	0	0
5	5	25	18	1	0	0	0
6	6	32	24	0	1	0	0
7	7	32	23	0	1	0	0
8	8	30	20	0	1	0	0
9	9	31	22	0	1	0	0
10	10	23	17	0	1	0	0
11	11	22	16	0	1	0	0
12	12	21	15	0	1	0	0
13	13	30	19	0	0	1	0
14	14	21	14	0	0	1	0
15	15	19	13	0	0	1	0
16	16	23	17	0	0	1	0
17	17	30	20	0	0	1	0
18	18	31	21	0	0	1	0
19	19	20	15	0	0	0	1
20	20	26	18	0	0	0	1
21	21	20	16	0	0	0	1
22	22	14	13	0	0	0	1
23	23	28	21	0	0	0	1
24	24	.	12	0	0	0	1

Note that STATA, unlike LIMDEP, records missing observations with a period, rather than a blank space. Closing the data editor, we see that our variables are now added to the variable list in STATA, and we are correctly told that our data consist of 7 variables and 24 observations.



Any time you wish to see your current data, you can go back to the data editor. We can also view the data by typing in "browse" in the command window. As the terms suggest, "browse" only allows you to see the data, while you can manually alter data in the "data editor".

READING SPACE, TAB, OR COMMA DELINEATED FILES INTO STATA

Next we consider externally created text files that are typically accompanied by the ".txt" or ".prn" extensions. As an example, we use the previous dataset with 24 observations on the 7 variables ("student," "post," "pre," "class1," "class2," "class3," and "class4") and saved it as a space delineated text file "post-pre.txt." To read the data into STATA, we need to utilize the "insheet" command. In the command window, type

insheet using "F:\NCEE (Becker)\post-pre.txt", delimiter(" ")

The "insheet" tells STATA to read in text data and "using" directs STATA to a particular file name. In this case, the file is saved in the location "F:\NCEE (Becker)\post-pre.txt", but this will vary by user. Finally, the "delimiter(" ")" option tells STATA that the data points in this file are separated by a space. If your data were tab delimited, you could type

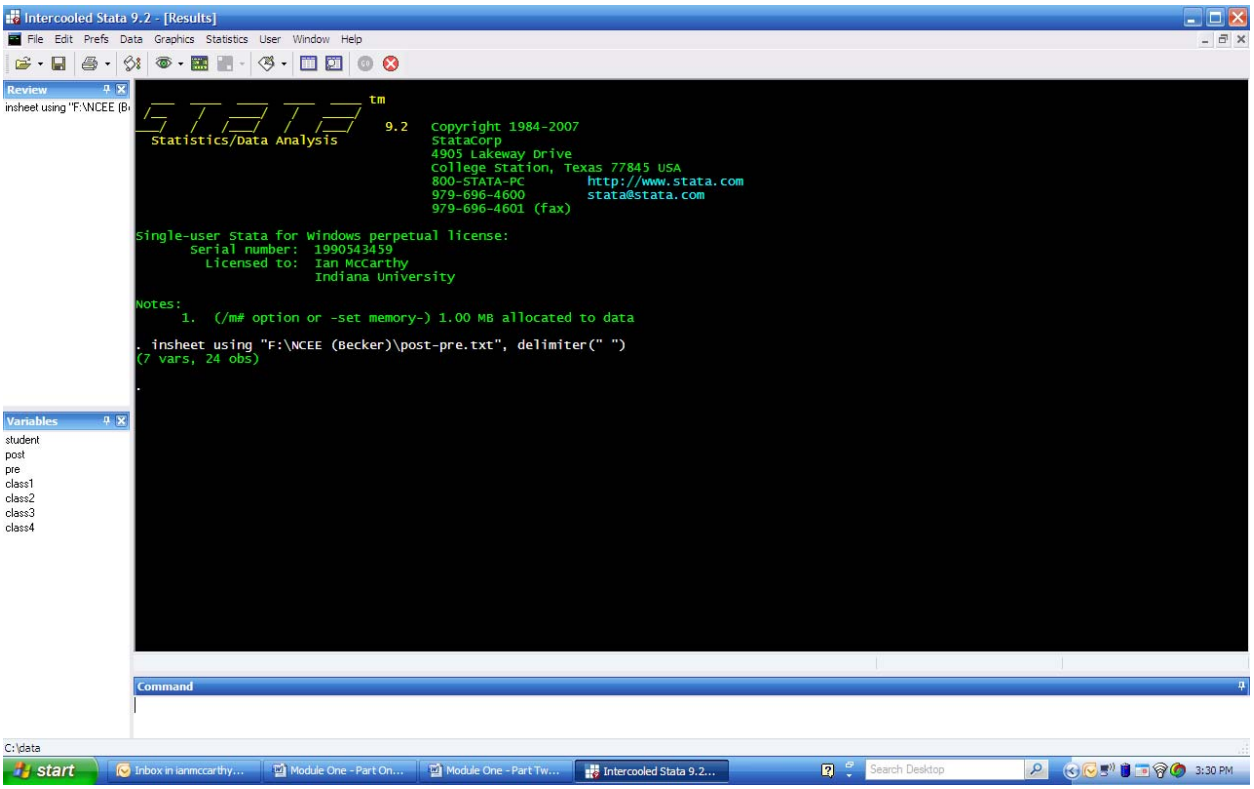
insheet using “F:\NCEE (Becker)\post-pre.txt”, tab

and if you were using a “.csv” file, you could type

insheet using “F:\NCEE (Becker)\post-pre.csv”, comma

In general, the “delimiter( )” option is used when your data have a less standard delimiter (e.g., a colon, semicolon, etc.).

Once you’ve typed the appropriate command into the command window, press enter to run that line of text. This should yield the following screen:

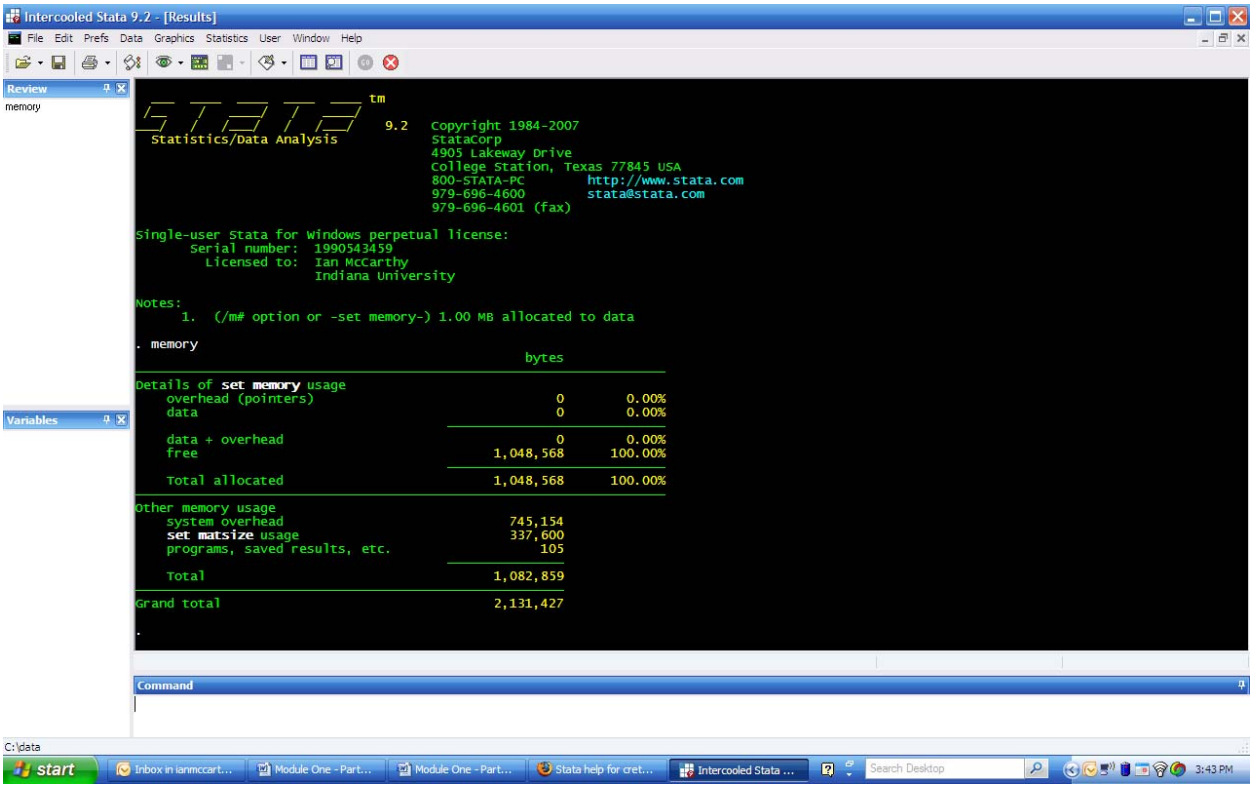


Just as before, STATA tells us that it has read a data set consisting of 7 variables and 24 observations, and we can access our variable list in the lower-left window pane. We can also see previously written lines from the “review” window in the upper-left window pane. Again, we can view our data by typing “browse” in the command window and pressing enter.

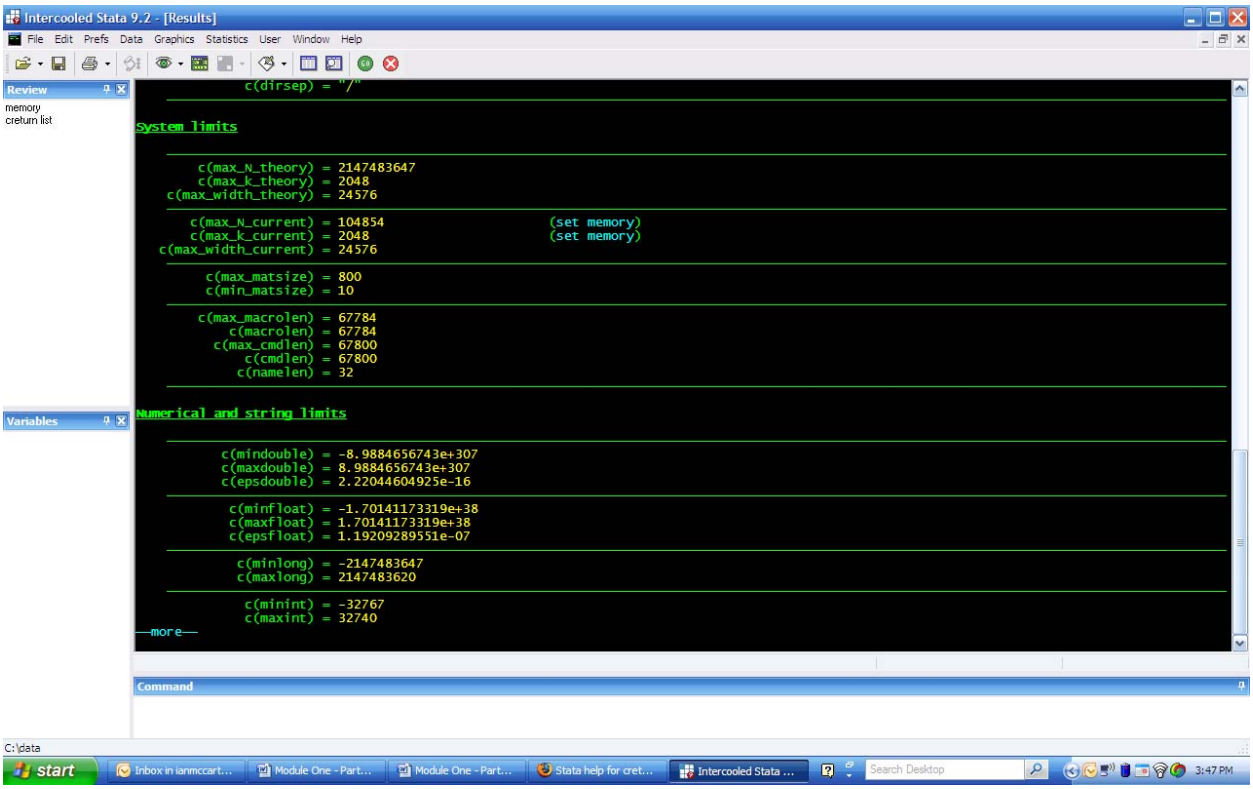
READING LARGE DATA FILES INTO STATA

The default memory allocation is different depending on the version of STATA you are using. When STATA first opens, it will indicate how much memory is allocated by default. From the previous screenshot, for instance, STATA indicates that 1.00mb is set aside for STATA’s use.

This is shown in the note directly above the entered command, which appears every time we start STATA. The 1.00mb memory is the standard for Intercooled STATA, which is the version used for this module. For a slightly more detailed look at the current memory allocation, you can type into the command window, "memory" and press enter. This provides the following:

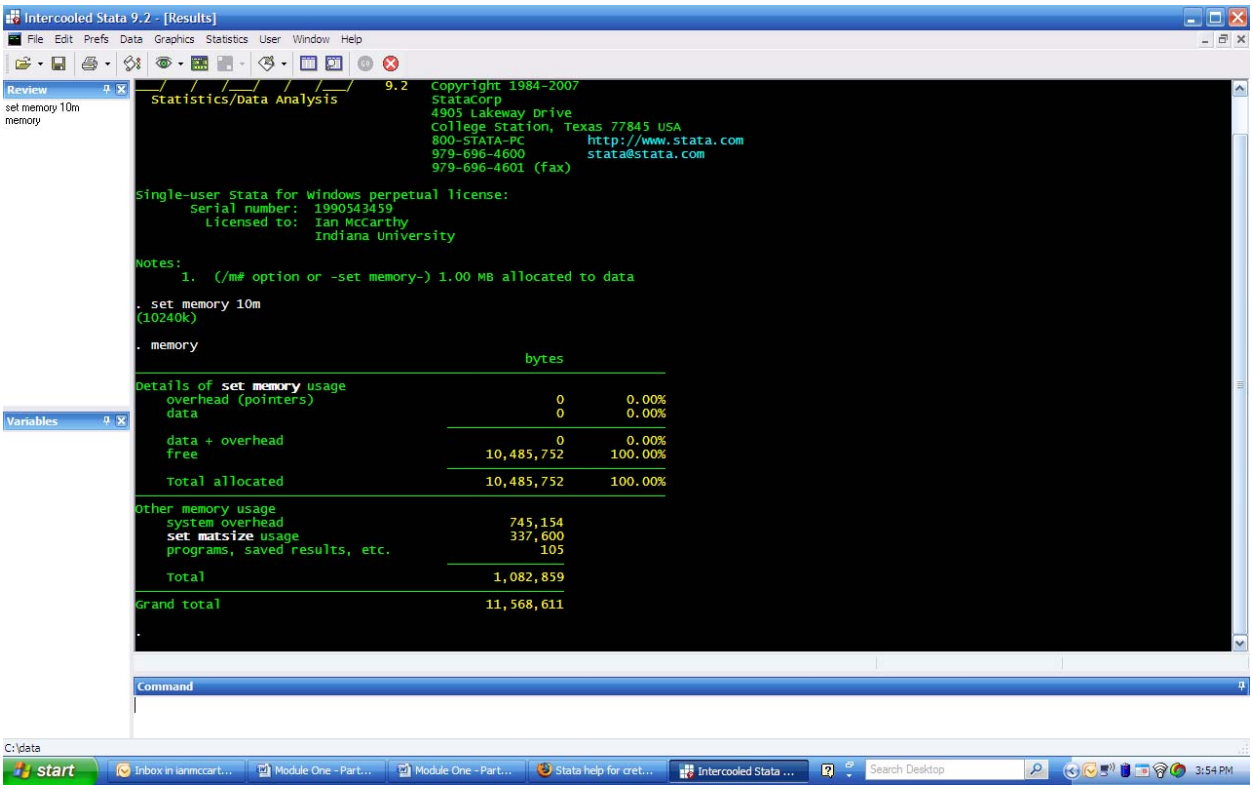


A more useful (and detailed) description of STATA's memory usage (among other things) can be obtained by typing "creturn list" into the command window. This provides:



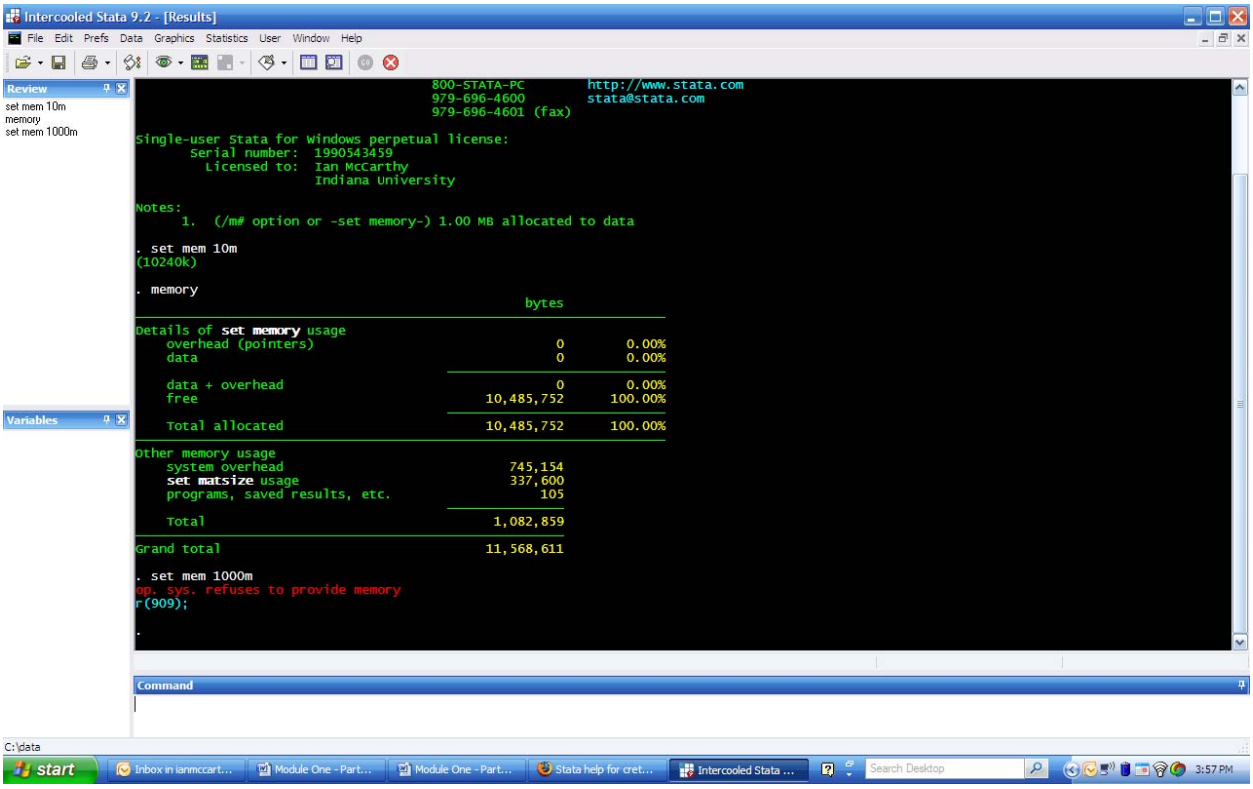
You may have to click on the "-more-" link (bottom left of the STATA output window) to see this output. You can also press spacebar in the command window (or any key) to advance screens whenever you see "-more-" at the bottom. Two things to notice from this screen are: (1) c(max_N_theory) tells us the maximum possible number of records our version of STATA will allow, while c(max_N_current) tells us the maximum possible number of records we have currently allocated to STATA based on our memory allocation, and (2) c(max_k_theory) tells us the maximum possible number of variables, while c(max_k_current) tells us the maximum number of variables based on our current memory allocation.

To work with large datasets (in this case, anything larger than 1mb), we can type "set memory 10m" into the command window and press enter. This increases the memory allocation to 10 mb, and you can increase by more or less to your preference. You can also increase STATA's memory allocation permanently by typing, "set memory 10m, permanently" into the command line. To check that our memory has actually increased, again type "memory" into the command window and press enter. We get the following screen:




```
  ___  ____  ____  ____  ____ ©                9.2   Copyright 1984-2007
 /__    /   ____/   /   ____/                        StataCorp
___/   /   /___/   /   /___/                         4905 Lakeway Drive
  Statistics/Data Analysis                           College Station, Texas 77845 USA
                                                     800-STATA-PC        http://www.stata.com
                                                     979-696-4600        stata@stata.com
                                                     979-696-4601 (fax)

Single-user Stata for Windows perpetual license:
       Serial number:  1990543459
         Licensed to:  Ian McCarthy
                       Indiana University

Notes:
      1.  (/m# option or -set memory-) 1.00 MB allocated to data

. set memory 10m
(10240k)

. memory
                                             bytes
-------------------------------------------------------------------------------
Details of set memory usage
    overhead (pointers)                          0        0.00%
    data                                         0        0.00%
                                        ------------------------
    data + overhead                              0        0.00%
    free                                10,485,752      100.00%
                                        ------------------------
    Total allocated                     10,485,752      100.00%
-------------------------------------------------------------------------------
Other memory usage
    system overhead                        745,154
    set matsize usage                      337,600
    programs, saved results, etc.              105
                                        ------------
    Total                                1,082,859
-------------------------------------------------------------------------------
Grand total                             11,568,611

.
```

The maximum amount of memory you can allocate to STATA varies based on your computer's performance. If we try to allocate more memory that our RAM can allow, we get an error:




```
                                                     800-STATA-PC        http://www.stata.com
                                                     979-696-4600        stata@stata.com
                                                     979-696-4601 (fax)

Single-user Stata for Windows perpetual license:
       Serial number:  1990543459
         Licensed to:  Ian McCarthy
                       Indiana University

Notes:
      1.  (/m# option or -set memory-) 1.00 MB allocated to data

. set mem 10m
(10240k)

. memory
                                             bytes
-------------------------------------------------------------------------------
Details of set memory usage
    overhead (pointers)                          0        0.00%
    data                                         0        0.00%
                                        ------------------------
    data + overhead                              0        0.00%
    free                                10,485,752      100.00%
                                        ------------------------
    Total allocated                     10,485,752      100.00%
-------------------------------------------------------------------------------
Other memory usage
    system overhead                        745,154
    set matsize usage                      337,600
    programs, saved results, etc.              105
                                        ------------
    Total                                1,082,859
-------------------------------------------------------------------------------
Grand total                             11,568,611

. set mem 1000m
op. sys. refuses to provide memory
r(909);

.
```

Note that the total amount of memory allowed depends on the computer's performance; however, the total number of variables allowed may be restricted by your version of STATA. (If you're using Small STATA, then the memory allocation is also limited.) For Intercooled STATA, for instance, we cannot have more than 2048 variables in our data set.

For the Becker and Powers data set, the 1mb allocation is sufficient, so we need only follow the process to import a ".csv" file described above. Note, however, that this data set does not contain variable names in the top row. You can assign names yourself with a slight addition to the insheet command:

insheet var1 var2 var3 … using "filename.csv", comma

Where, var1 var2 var3 …, are the variable names for each of the 64 variables in the data set. Of course, manually adding all 64 variable names can be irritating. For more details on how to import data sets with data dictionaries (i.e., variable names and definitions in external files), try typing "help infile" into the command window. If you do not assign variable names, then STATA will provide default variable names of "v1, v2, v3, etc.".

LEAST-SQUARES ESTIMATION AND LINEAR RESTRICTIONS IN STATA

As in the previous section using LIMDEP, we now demonstrate various regression tools in STATA using the "post-pre" data set. Recall the model being estimated is

$$post = \beta_1 + \beta_2 pre + f(classes) + \varepsilon .$$

STATA automatically drops any missing observations from our analysis, so we need not restrict the data in any of our commands. In general, the syntax for a basic OLS regression in STATA is

regress *y-variable x-variables,*

where *y-variable* is just the independent variable name and *x-variables* are the dependent variable names. Now is a good time to mention STATA's very useful help menu. Typing "help regress" into the command window and pressing enter will open a thorough description of the regress command and all of its options, and similarly with any command in STATA.

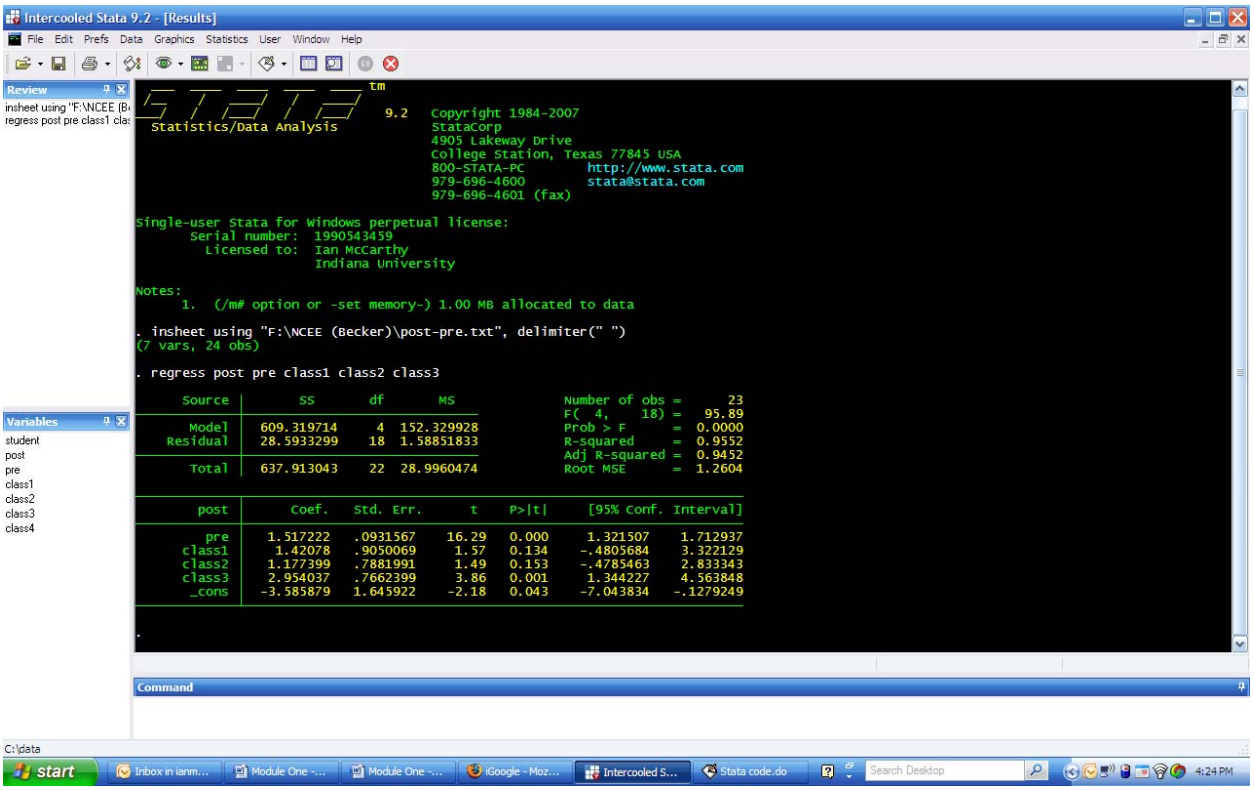
Once you have your data read into STATA, let's estimate the model

$$post = \beta_1 + \beta_2 pre + \beta_3 class1 + \beta_4 class2 + \beta_5 class3 + \varepsilon$$

by typing:

regress post pre class1 class2 class3

into the command window and pressing enter. We get the following output:

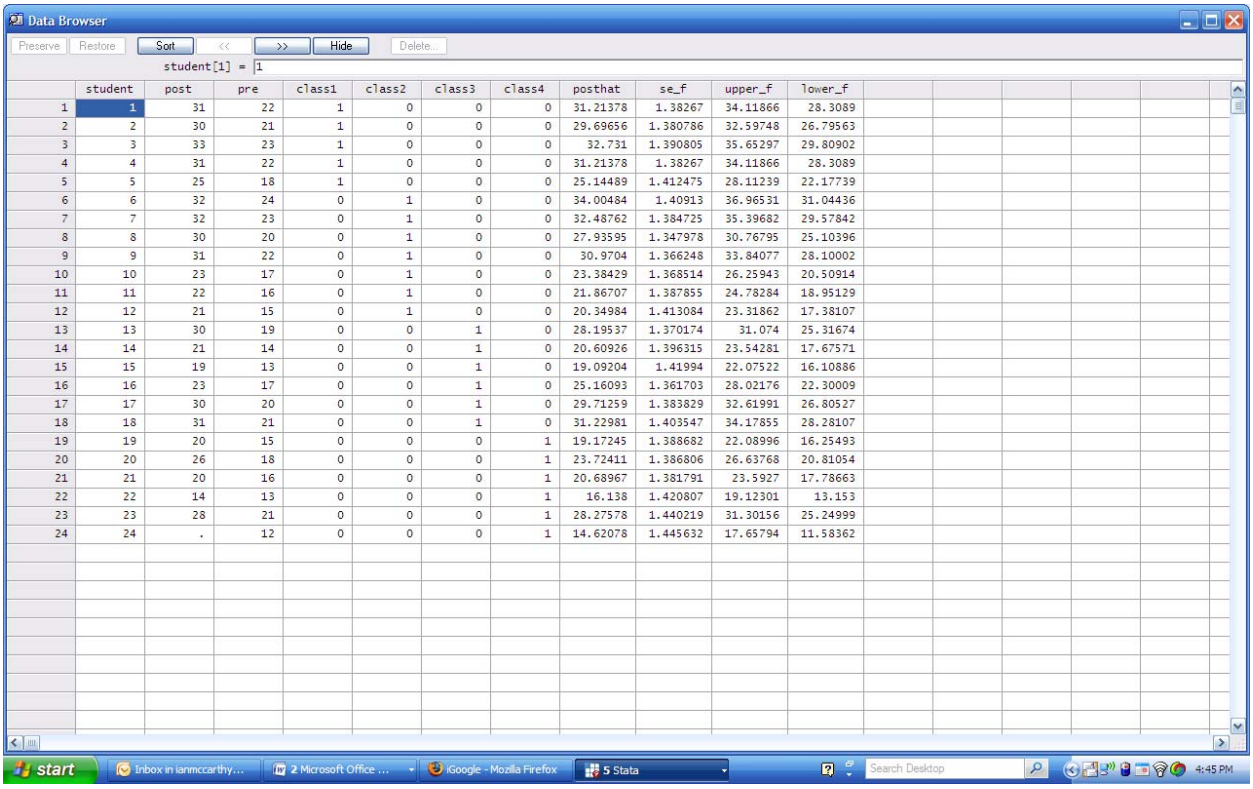


To see the predicted post-test scores (with confidence intervals) from our regression, we type:

```
predict posthat
predict se_f, stdf
generate upper_f = posthat + invttail(e(df_r),0.025)*se_f
generate lower_f = posthat + invttail(e(df_r),0.025)*se_f
```

You can either copy and paste these commands directly into the command window and press enter, or you can enter each one directly into the command window and press enter one at a time. Notice the use of the "predict" and "generate" keywords in the previous set of commands. After running a regression, STATA has lots of data stored away, some of which is shown in the output and some that is not. By typing "predict posthat", STATA applies the estimated regression equation to the 24 observations in the sample to get predicted *y*-values. These predicted *y*-values are the default prediction for the "predict" command, and if we want the standard error of these predictions, we need to use "predict" again but this time specify the option "stdf". This stands for the standard deviation of the forecast. Both posthat and se_f are new variables that STATA has created for us. Now, to get the upper and lower bounds of a 95% confidence interval, we apply the usual formula taking the predicted value plus/minus the margin of error. Typing "generate upper_f=…" and "generate lower_f=…" creates two new variables named "upper_f" and

"lower_f", respectively. To see our predictions, we can type "browse" into the command window and press enter. This yields:



	student	post	pre	class1	class2	class3	class4	posthat	se_f	upper_f	lower_f
1	1	31	22	1	0	0	0	31.21378	1.38267	34.11866	28.3089
2	2	30	21	1	0	0	0	29.69656	1.380786	32.59748	26.79563
3	3	33	23	1	0	0	0	32.731	1.390805	35.65297	29.80902
4	4	31	22	1	0	0	0	31.21378	1.38267	34.11866	28.3089
5	5	25	18	1	0	0	0	25.14489	1.412475	28.11239	22.17739
6	6	32	24	0	1	0	0	34.00484	1.40913	36.96531	31.04436
7	7	32	23	0	1	0	0	32.48762	1.384725	35.39682	29.57842
8	8	30	20	0	1	0	0	27.93595	1.347978	30.76795	25.10396
9	9	31	22	0	1	0	0	30.9704	1.366248	33.84077	28.10002
10	10	23	17	0	1	0	0	23.38429	1.368514	26.25943	20.50914
11	11	22	16	0	1	0	0	21.86707	1.387855	24.78284	18.95129
12	12	21	15	0	1	0	0	20.34984	1.413084	23.31862	17.38107
13	13	30	19	0	0	1	0	28.19537	1.370174	31.074	25.31674
14	14	21	14	0	0	1	0	20.60926	1.396315	23.54281	17.67571
15	15	19	13	0	0	1	0	19.09204	1.41994	22.07522	16.10886
16	16	23	17	0	0	1	0	25.16093	1.361703	28.02176	22.30009
17	17	30	20	0	0	1	0	29.71259	1.383829	32.61991	26.80527
18	18	31	21	0	0	1	0	31.22981	1.403547	34.17855	28.28107
19	19	20	15	0	0	0	1	19.17245	1.388682	22.08996	16.25493
20	20	26	18	0	0	0	1	23.72411	1.386806	26.63768	20.81054
21	21	20	16	0	0	0	1	20.68967	1.381791	23.5927	17.78663
22	22	14	13	0	0	0	1	16.138	1.420807	19.12301	13.153
23	23	28	21	0	0	0	1	28.27578	1.440219	31.30156	25.24999
24	24	.	12	0	0	0	1	14.62078	1.445632	17.65794	11.58362

Just as with LIMDEP, our 95% confidence interval for the 24th student's predicted post-test score is [11.5836, 17.6579]. For more information on the "predict" command, try typing "help predict" into the command window.

To test the linear restriction of all class coefficients being zero, we type:

test class1 class2 class3

into the command window and press enter. STATA automatically forms the correct test statistic, and we see

$$F(3, 18) = 5.16$$
$$Prob > F = 0.0095$$

The second line gives us the p-value, where we see that we can reject the null that all class coefficients are zero at any probability of Type I error greater than 0.0095.

TEST FOR A STRUCTURAL BREAK (CHOW TEST)

The above test of the linear restriction $\beta_3 = \beta_4 = \beta_5 = 0$ (no difference among classes), assumed that the pretest slope coefficient was constant, fixed and unaffected by the class to which a student belonged. A full structural test can be performed in two possible ways. One, we can run each restricted regression and the unrestricted regression, take note of the residual sums of squares from each regression, and explicitly calculate the F-statistic. This requires the fitting of four separate regressions to obtain the four residual sum of squares that are added to obtain the unrestricted sum of squares. The restricted sum of squares is obtained from a regression of posttest on pretest with no dummies for the classes; that is, the class to which a student belongs is irrelevant in the manner in which pretests determine the posttest score.

For this, we can type:

regress post pre if class1==1

into the command window and press enter. The resulting output is as follows:

```
. regress post pre if class1==1

      Source |       SS       df       MS              Number of obs =       5
-------------+------------------------------           F(  1,     3) =  417.63
       Model |  35.7432432     1  35.7432432           Prob > F      =  0.0003
    Residual |  .256756757     3  .085585586           R-squared     =  0.9929
-------------+------------------------------           Adj R-squared =  0.9905
       Total |          36     4           9           Root MSE      =  .29255

------------------------------------------------------------------------------
        post |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         pre |   1.554054   .0760448    20.44   0.000     1.312046    1.796063
       _cons |  -2.945946    1.61745    -1.82   0.166    -8.093392    2.201501
------------------------------------------------------------------------------
```

We see in the upper-left portion of this output that the residual sum of squares from this restricted regression is 0.2568. We can similarly run a restricted regression for only students in class 2 by specifying the option "if class2==1", and so forth for classes 3 and 4.

The second way to test for a structural break is to create several interaction terms and test whether the dummy and interaction terms are jointly significantly different from zero. To perform the Chow test this way, we first generate interaction terms between all dummy variables and independent variables. To do this in STATA, type the following into the command window and press enter:

generate pre_c1=pre*class1
generate pre_c2=pre*class2
generate pre_c3=pre*class3

With our new variables created, we now run a regression with all dummy and interaction terms included, as well as the original independent variable. In STATA, we need to type:

regress post pre class1 class2 class3 pre_c1 pre_c2 pre_c3

into the command window and press enter. The output for this regression is not meaningful, as it is only the test that we're interested in. To run the test, we can then type:

test class1 class2 class3 pre_c1 pre_c2 pre_c3

into the command window and press enter. The resulting output is:

```
. test class1 class2 class3 pre_c1 pre_c2 pre_c3

 ( 1)  class1 = 0
 ( 2)  class2 = 0
 ( 3)  class3 = 0
 ( 4)  pre_c1 = 0
 ( 5)  pre_c2 = 0
 ( 6)  pre_c3 = 0

       F(  6,    15) =    2.93
            Prob > F =    0.0427
```

Just as we saw in LIMDEP, our *F*-statistic is 2.93, with a p-value of 0.0427. We again reject the null (at a probability of Type I error=0.05) and conclude that class is important either through the slope or intercept coefficients. This type of test will always yield results identical to the restricted regression approach.

HETEROSCEDASTICITY

You can control for heteroscedasticity across observations or within specific groups (in this class, within a given class, but not across classes) by specifying the "robust" or "cluster" option, respectively, at the end of your regression command.

To account for a common error term within groups, but not across groups, we first create a class variable that identifies each student into one of the 4 classes. This is used to specify which group (or cluster) a student is in. To generate this variable, type:

generate class=class1 + 2*class2 + 3*class3 + 4*class4

into the command window and press enter. Then to allow for clustered error terms, our regression command is:

regress post pre class1 class2 class3, cluster(class)

This gives us the following output:

```
. regress post pre class1 class2 class3, cluster(class)

Linear regression                                      Number of obs =      23
```

```
                                                      F(  0,     3) =       .
                                                      Prob > F      =       .
                                                      R-squared     =  0.9552
Number of clusters (class) = 4                        Root MSE      =  1.2604

------------------------------------------------------------------------------
             |               Robust
        post |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         pre |   1.517222   .1057293    14.35   0.001     1.180744      1.8537
      class1 |    1.42078   .4863549     2.92   0.061    -.1270178     2.968579
      class2 |   1.177399   .3141671     3.75   0.033     .1775785     2.177219
      class3 |   2.954037   .0775348    38.10   0.000     2.707287     3.200788
       _cons |  -3.585879   1.755107    -2.04   0.134    -9.171412     1.999654
------------------------------------------------------------------------------
```

Similarly, to account for general heteroscedasticity across individual observations, our regression command is:

regress post pre class1 class2 class3, robust

and we get the following output:

```
. regress post pre class1 class2 class3, robust

Linear regression                                     Number of obs =      23
                                                      F(  4,    18) =  165.74
                                                      Prob > F      =  0.0000
                                                      R-squared     =  0.9552
                                                      Root MSE      =  1.2604

------------------------------------------------------------------------------
             |               Robust
        post |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         pre |   1.517222   .0824978    18.39   0.000       1.3439    1.690543
      class1 |    1.42078   .7658701     1.86   0.080     -.188253    3.029814
      class2 |   1.177399   .8167026     1.44   0.167      -.53843    2.893227
      class3 |   2.954037   .9108904     3.24   0.005     1.040328    4.867747
       _cons |  -3.585879   1.706498    -2.10   0.050    -7.171098   -.0006609
------------------------------------------------------------------------------
```

ESTIMATING PROBIT MODELS IN STATA

We now want to estimate a probit model using the Becker and Powers data set. First, read in the ".csv" file:[i]

```
. insheet a1 a2 x3 c al am an ca cb cc ch ci cj ck cl cm cn co cs ct cu  ///
> cv cw db dd di dj dk dl dm dn dq dr ds dy dz ea eb ee ef               ///
> ei ej ep eq er et ey ez ff fn fx fy fz ge gh gm gn gq gr hb            ///
> hc hd he hf using "F:\NCEE (Becker)\BECK8WO2.csv", comma
(64 vars, 2849 obs)
```

Notice the "///" at the end of each line. Because STATA by default reads the end of the line as the end of a command, you have to tell it when the command actually goes on to the next line. The "///" tells STATA to continue reading this command through the next line.

As always, we should look at our data before we start doing any work. Typing "browse" into the command window and pressing enter, it looks as if several variables have been read as character strings rather than numeric values. We can see this by typing "describe" into the command window or simply by noting that string variables appear in red in the browsing window. This is a somewhat common problem when using STATA with Excel, usually because of variable names in the Excel files or because of spaces placed in front or after numeric values. If there are spaces in any cell that contains an otherwise numeric value, STATA will read the entire column as a character string. Since we know all variables should be numeric, we can fix this problem by typing:

```
destring, replace
```

into the command window and pressing enter. This automatically codes all variables as numeric variables.

Also note that the original Excel .csv file has several "extra" observations at the end of the data set. These are essentially extra rows that have been left blank but were somehow utilized in the original Excel file (for instance, just pressing enter at last cell will generate a new record with all missing variables). STATA correctly reads these 12 observations as missing values, but because we know these are not real observations, we can just drop these with the command "drop if a1==.". This works because a1 is not missing for any of the other observations.

Now we recode the variable a2 as a categorical variable, where a2=1 for doctorate institutions (between 100 and 199), a2=2 for comprehensive master's degree granting institutions (between 200 and 299), a2=3 for liberal arts colleges (between 300 and 399), and a2=4 for two-year colleges (between 400 and 499). To do this, type the following command into the command window:

```
recode a2 (100/199=1) (200/299=2) (300/399=3) (400/499=4)
```

Once we've recoded the variable, we can generate the 4 dummy variables as follows:[ii]

```
generate doc=(a2==1) if a2!=.
generate comp=(a2==2) if a2!=.
generate lib=(a2==3) if a2!=.
generate twoyr=(a2==4) if a2!=.
```

The more lengthy way to generate these variables would be to first generate new variables equal to zero, and then replace each one if the relevant condition holds. But the above commands are a more concise way.

Next 1 - 0 bivariates are created to show whether the instructor had a PhD degree and where the student got a positive score on the postTUCE. We also create new variables, dmsq and hbsq, to allow for quadratic forms in teacher experiences and class size:

```
generate phd=(dj==3)  if dj!=.
generate final=(cc>0) if cc!=.
generate dmsq=dm^2
generate hbsq=hb^2
```

In this data set, all missing values are coded −9. Thus, adding together some of the responses to the student evaluations provides information as to whether a student actually completed an evaluation. For example, if the sum of ge, gh, gm, and gq equals −36, we know that the student did not complete a student evaluation in a meaningful way. A dummy variable to reflect this fact is then created by:[iii]

```
generate noeval=(ge + gh + gm + gq == -36)
```

Finally, from the TUCE developer it is known that student number 2216 was counted in term 2 but was in term 1 but no postTUCE was taken. This error is corrected with the following command:

```
recode hb (90=89)
```

We are now ready to estimate the probit model with final as our dependent variable. Because missing values are coded as -9 in this data set, we need to avoid these observations in our analysis. The quickest way to avoid this problem is just to recode all of the variables, setting every variable equal to "." if it equals "-9". Because there are 64 variables, we do not want to do this one at a time, so instead we type:

```
foreach x of varlist * {
replace `x'=. if `x'==-9
}
```

You should type this command exactly as is for it to work correctly, including pressing enter after the first open bracket. Also note that the single quotes surrounding each x in the replace statement are two different characters. The first single quote is the key directly underneath the escape key (for most keyboards) while the closing single quote is the standard single quote keystroke by the enter key. For more help on this, type "help foreach" into the command window.

Finally, we drop all observations where an=. and where cs=0 and run the probit model by typing

```
drop if an==.
```

drop if cs==0
probit final an hb doc comp lib ci ck phd noeval

into the command window and pressing enter. We can then retrieve the marginal effects by typing "mfx" into the command window and pressing enter. This yields the following output:

```
. drop if cs==0
(1 observation deleted)

. drop if an==.
(249 observations deleted)

. probit final an hb doc comp lib ci ck phd noeval

Iteration 0:   log likelihood = -1284.2161
Iteration 1:   log likelihood = -840.66421
Iteration 2:   log likelihood = -823.09278
Iteration 3:   log likelihood = -822.74126
Iteration 4:   log likelihood = -822.74107

Probit regression                                 Number of obs   =       2587
                                                  LR chi2(9)      =     922.95
                                                  Prob > chi2     =     0.0000
Log likelihood = -822.74107                       Pseudo R2       =     0.3593

------------------------------------------------------------------------------
       final |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          an |    .022039   .0094752     2.33   0.020      .003468      .04061
          hb |  -.0048826   .0019241    -2.54   0.011    -.0086537   -.0011114
         doc |   .9757148   .1463617     6.67   0.000     .6888511    1.262578
        comp |   .4064945   .1392651     2.92   0.004       .13354     .679449
         lib |   .5214436   .1766459     2.95   0.003      .175224    .8676632
          ci |   .1987315   .0916865     2.17   0.030     .0190293    .3784337
          ck |     .08779   .1342874     0.65   0.513    -.1754085    .3509885
         phd |   -.133505   .1030316    -1.30   0.195    -.3354433    .0684333
      noeval |  -1.930522   .0723911   -26.67   0.000    -2.072406   -1.788638
       _cons |   .9953498   .2432624     4.09   0.000     .5185642    1.472135
------------------------------------------------------------------------------

. mfx

Marginal effects after probit
      y  = Pr(final) (predict)
         =  .88118215
------------------------------------------------------------------------------
variable |      dy/dx    Std. Err.     z    P>|z|  [    95% C.I.   ]      X
---------+--------------------------------------------------------------------
      an |    .004378      .00188    2.33   0.020   .000699  .008057   10.5968
      hb |  -.0009699      .00038   -2.54   0.011  -.001719  -.00022   55.5589
     doc*|   .1595047      .02039    7.82   0.000   .119537  .199473   .317743
    comp*|   .0778334      .02588    3.01   0.003   .027107   .12856   .417859
     lib*|   .0820826      .02145    3.83   0.000   .040039  .124127   .135678
      ci |   .0394776      .01819    2.17   0.030   .003834  .075122   1.23116
      ck*|   .0182048      .02902    0.63   0.530  -.038667  .075077   .919985
     phd*|  -.0257543      .01933   -1.33   0.183  -.063632  .012123   .686123
  noeval*|   -.533985      .01959  -27.26   0.000  -.572373 -.495597   .290684
------------------------------------------------------------------------------
(*) dy/dx is for discrete change of dummy variable from 0 to 1
```

For the other probit model (using hc rather than hb), we get:

```
. probit final an hc doc comp lib ci ck phd noeval

Iteration 0:   log likelihood = -1284.2161
Iteration 1:   log likelihood = -843.39917
Iteration 2:   log likelihood = -826.28953
Iteration 3:   log likelihood = -825.94736
Iteration 4:   log likelihood = -825.94717

Probit regression                                 Number of obs   =       2587
                                                  LR chi2(9)      =     916.54
                                                  Prob > chi2     =     0.0000
Log likelihood = -825.94717                       Pseudo R2       =     0.3568

------------------------------------------------------------------------------
       final |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          an |   .0225955   .0094553     2.39   0.017     .0040634    .0411276
          hc |   .0001586    .002104     0.08   0.940    -.0039651    .0042823
         doc |    .880404   .1486641     5.92   0.000     .5890278     1.17178
        comp |   .4596089   .1379817     3.33   0.001     .1891698     .730048
         lib |   .5585268   .1756814     3.18   0.001     .2141976     .902856
          ci |   .1797199    .090808     1.98   0.048     .0017394    .3577004
          ck |   .0141566   .1333267     0.11   0.915    -.2471589    .2754722
         phd |  -.2351326   .1010742    -2.33   0.020    -.4332344   -.0370308
      noeval |  -1.928216   .0723636   -26.65   0.000    -2.070046   -1.786386
       _cons |   .8712666   .2411741     3.61   0.000     .3985742    1.343959
------------------------------------------------------------------------------

. mfx

Marginal effects after probit
      y  = Pr(final) (predict)
         =  .88073351
------------------------------------------------------------------------------
variable |      dy/dx    Std. Err.     z    P>|z|  [    95% C.I.   ]      X
---------+--------------------------------------------------------------------
      an |   .0045005      .00188    2.40   0.017    .00082  .008181   10.5968
      hc |   .0000316      .00042    0.08   0.940   -.00079  .000853   49.9749
     doc*|   .1467544      .02132    6.88   0.000   .104969   .18854   .317743
    comp*|    .087859      .02554    3.44   0.001   .037809  .137909   .417859
     lib*|   .0867236      .02066    4.20   0.000   .046228   .12722   .135678
      ci |   .0357961      .01807    1.98   0.048   .000383  .071209   1.23116
      ck*|   .0028395      .02693    0.11   0.916  -.049938  .055617   .919985
     phd*|  -.0444863      .01819   -2.45   0.014  -.080145 -.008828   .686123
  noeval*|  -.5339711      .01957  -27.29   0.000  -.572326 -.495616   .290684
------------------------------------------------------------------------------
(*) dy/dx is for discrete change of dummy variable from 0 to 1
```

Results from each model are equivalent to those of LIMDEP, where we see that the estimated coefficient on hb is -0.005 with a p-value of 0.01, and the estimated coefficient on hc is 0.00007 with a p-value of 0.974. These results imply that initial class size is strongly significant while final class size is insignificant.

To assess model fit, we can form the predicted 0 or 1 values by first taking the predicted probabilities and then transforming these into 0 or 1 depending on whether the predicted probability is greater than .5. Then we can look at a tabulation to see how many correct 0s and 1s our probit model predicts. Because we have already run the models, we are not interested in the output, so to look only at these predictions, type the following into the command window:

```
quietly probit final an hb doc comp lib ci ck phd noeval
predict prob1
generate finalhat1=(prob1>.5)
```

this yields:

```
. quietly probit final an hb doc comp lib ci ck phd noeval

. predict prob1
(option p assumed; Pr(final))

. generate finalhat1=(prob1>.5)

. tab finalhat1 final

           |         final
 finalhat1 |         0          1 |     Total
-----------+----------------------+----------
         0 |       342        197 |       539
         1 |       168      1,880 |     2,048
-----------+----------------------+----------
     Total |       510      2,077 |     2,587
```

These results are exactly the same as with LIMDEP. For the second model, we get

```
. quietly probit final an hc doc comp lib ci ck phd noeval

. predict prob2
(option p assumed; Pr(final))

. generate finalhat2=(prob2>.5)

. tab finalhat2 final

           |         final
 finalhat2 |         0          1 |     Total
-----------+----------------------+----------
         0 |       337        192 |       529
         1 |       173      1,885 |     2,058
-----------+----------------------+----------
     Total |       510      2,077 |     2,587
```

Again, these results are identical to those of LIMDEP.

We can also use the built-in processes to do these calculations. To do so, type "estat class" after the model you've run. Part of the resulting output will be the tabulation of predicted versus actual values. Furthermore, to perform a Pearson goodness of fit test, type "estat gof" into the command window after you have run your model. This will provide a Chi-square value. All of these postestimation tools conclude that both models do a sufficient job of prediction.

REFERENCES

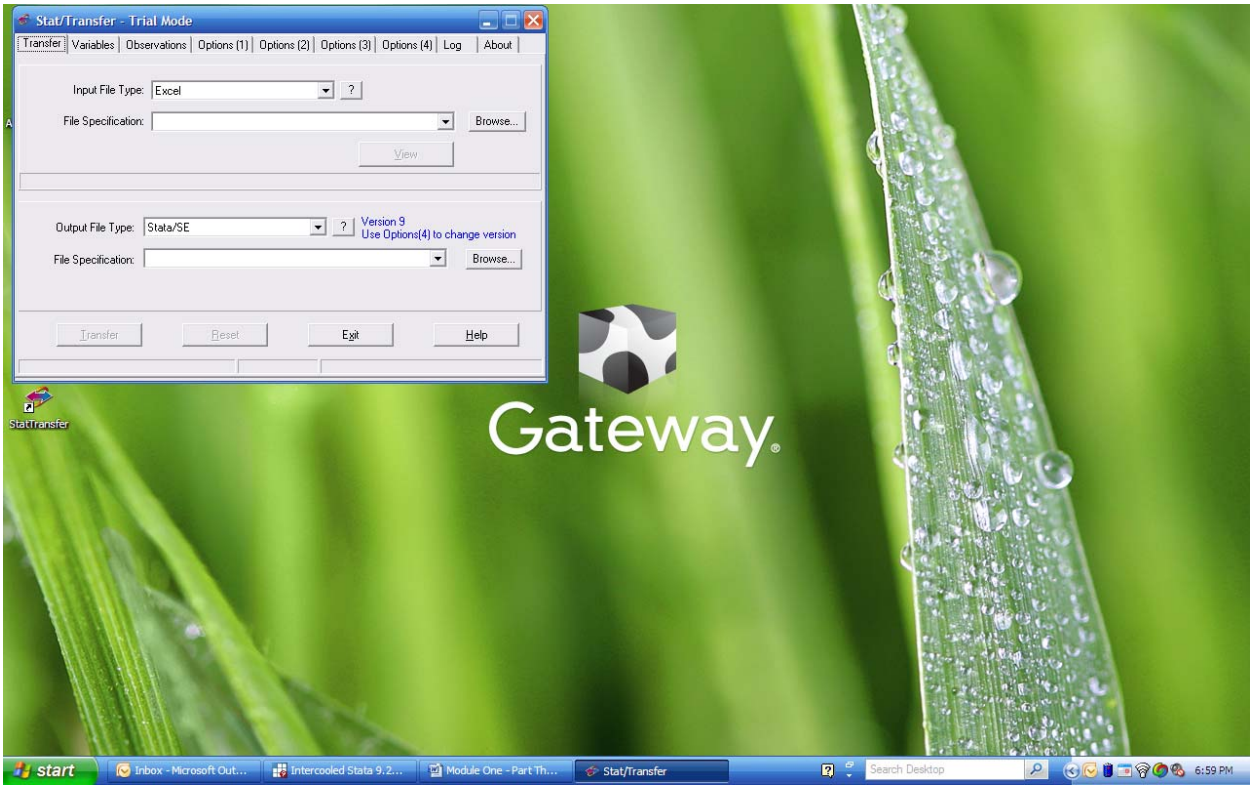
Kolenikov, Stanislav (2001). "Review of STATA 7" *The Journal of Applied Econometrics,* 16(5), October: 637-46.

Becker, William E. and John Powers (2001). "Student Performance, Attrition, and Class Size Given Missing Student Data," *Economics of Education Review*, Vol. 20, August: 377-388.

APPENDIX A : Using Stat/Transfer

Stat/Transfer is a convenient program used to convert data from one format to another. Although this program is not free, there is a free trial version available at www.stattransfer.com. Note that the trial program will not convert the entire data set—it will drop one observation.

Nonetheless, Stat/Transfer is very user friendly. If you install and open the trial program, your screen should look something like:



We want to convert the ".xls" file into a STATA format (".dta"). To do this, we need to first specify the original file type (e.g., Excel), then specify the location of the file. We then specify the format that we want (in this case, a STATA ".dta" file). Then click on Transfer, and Stat/Transfer automatically converts the data into the format you've asked.

To open this new ".dta" file in STATA, simply type

use "filename.dta"

into the command window and press enter.

ENDNOTES

---

[i] When using the "insheet" command, STATA automatically converts A1 to a1, A2 to a2 and so forth. STATA is, however, case sensitive. Therefore, whether users specify "insheet A1 A2 …" or "insheet a1 a2 …," we must still call the variables in lower case. For instance, the following insheet command will work the exact same as that provided in the main text:

```
. insheet A1 A2 X3 C AL AM AN CA CB CC CH CI CJ CK CL CM CN CO CS CT CU  ///
> CV CW DB DD DI DJ DK DL DM DN DQ DR DS DY DZ EA EB EE EF              ///
> EI EJ EP EQ ER ET EY EZ FF FN FX FY FZ GE GH GM GN GQ GR HB           ///
> HC HD HE HF using "F:\NCEE (Becker)\BECK8WO2.csv", comma
```

[ii] The conditions "if a2!=." tell STATA to run the command only if a2 is not missing. Although this particular dataset does not contain any missing values, it is generally good practice to always use this type of condition when creating dummy variables the way we have done here. For example, if there were a missing observation, the command "gen doc=(a2==1)" would set doc=0 even if a2 is missing.

[iii] An alternative procedure is to first set all variables to missing if they equal -9 and then generate the dummy variable using:

```
generate noeval=(ge==.&gh==.&gm==.&gq==.)
```

MODULE ONE, PART FOUR: READING DATA INTO SAS, CREATING AND RECODING VARIABLES, AND ESTIMATING AND TESTING MODELS IN SAS

This Part Four of Module One provides a cookbook-type demonstration of the steps required to read or import data into SAS. The reading of both small and large text and Excel files are shown though real data examples. The procedures to recode and create variables within SAS are demonstrated. Commands for least-squares regression estimation and maximum likelihood estimation of probit and logit models are provided. Consideration is given to analysis of variance and the testing of linear restrictions and structural differences, as outlined in Part One. Parts Two and Three provide the LIMDEP and STATA commands for the same operations undertaken here in Part Four with SAS. For a thorough review of SAS, see Delwiche and Slaughter's *The Little SAS Book: A Primer* (2003).

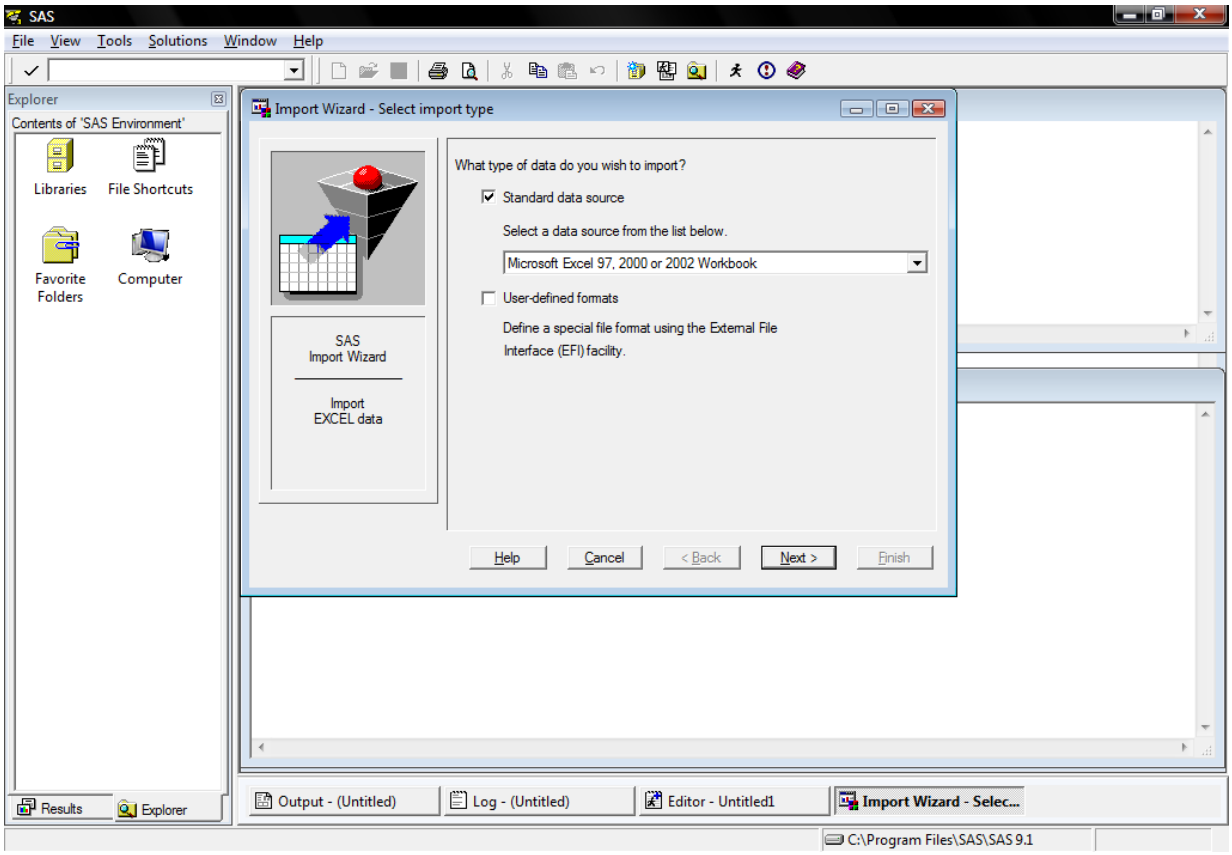
IMPORTING EXCEL FILES INTO SAS

SAS can read or import data in several ways. The most commonly imported files by students are those created in Microsoft Excel with the ".xls" file name extension. Researchers tend to use flat files that compactly store data. In what follows, we focus on importing data using Excel files. To see how this is done, consider the data set in the Excel file "post-pre.xls," which consists of test scores for 24 students in four classes. The column title "Student" identifies the 24 students by number, "post" provides each student's post-course test score, "pre" is each student's pre-course test score, and "class" identifies to which one of the four classes the student was assigned, e.g., class4 = 1 if student was in the fourth class and class4 = 0 if not. The "." in the post column for student 24 indicates that the student is missing a post-course test score.

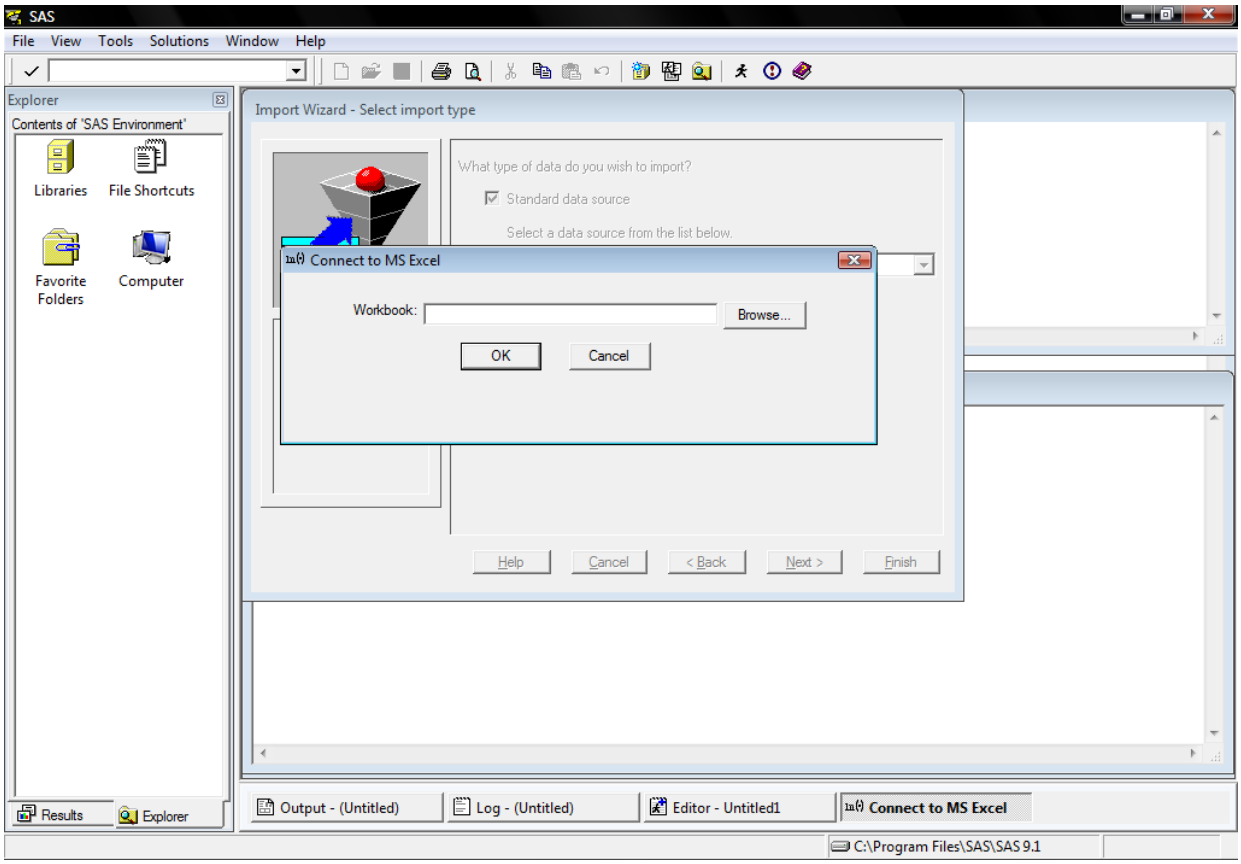
student	post	pre	class1	class2	class3	class4
1	31	22	1	0	0	0
2	30	21	1	0	0	0
3	33	23	1	0	0	0
4	31	22	1	0	0	0
5	25	18	1	0	0	0
6	32	24	0	1	0	0
7	32	23	0	1	0	0
8	30	20	0	1	0	0
9	31	22	0	1	0	0
10	23	17	0	1	0	0
11	22	16	0	1	0	0
12	21	15	0	1	0	0

13	30	19	0	0	1	0
14	21	14	0	0	1	0
15	19	13	0	0	1	0
16	23	17	0	0	1	0
17	30	20	0	0	1	0
18	31	21	0	0	1	0
19	20	15	0	0	0	1
20	26	18	0	0	0	1
21	20	16	0	0	0	1
22	14	13	0	0	0	1
23	28	21	0	0	0	1
24	.	12	0	0	0	1

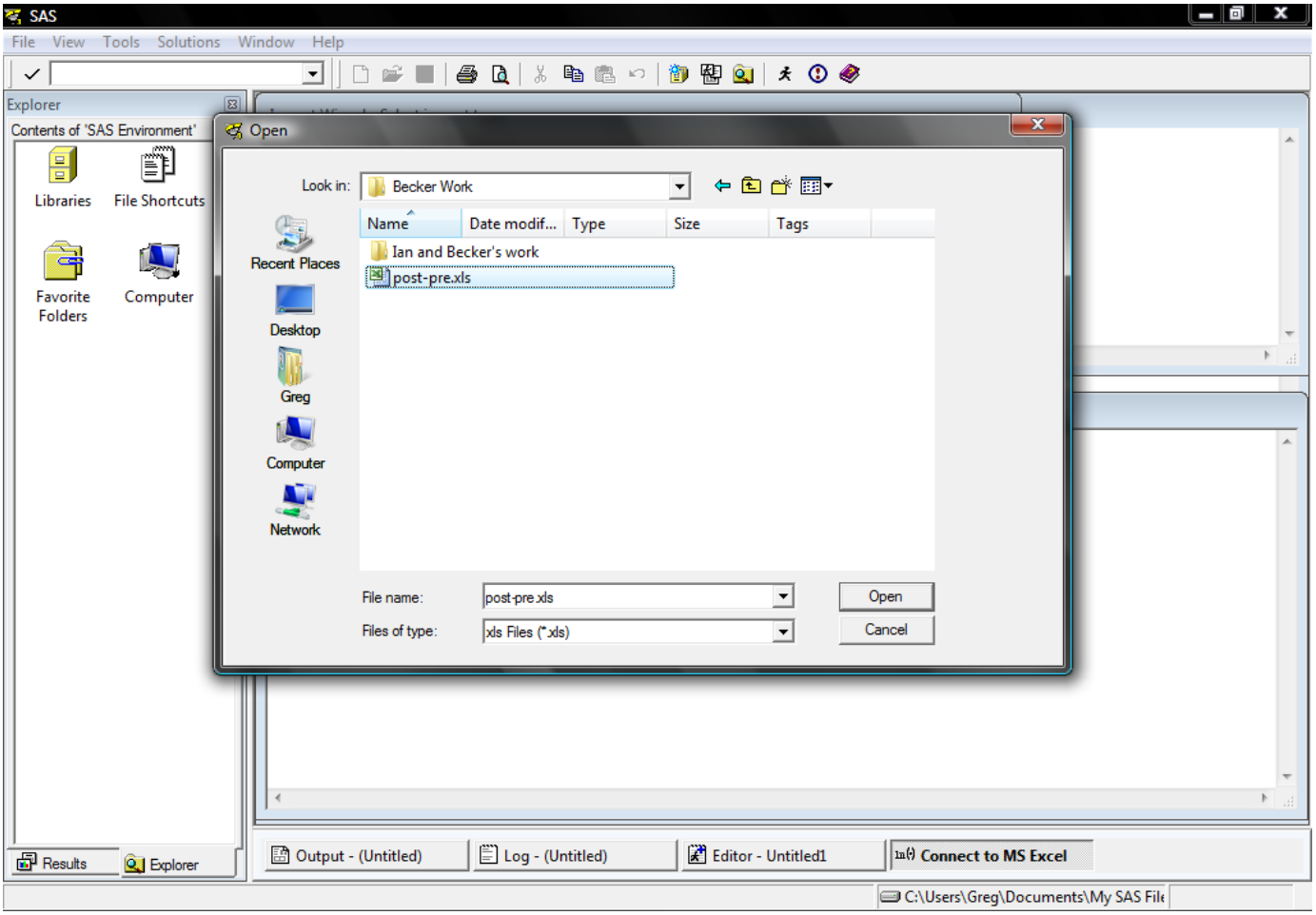
To start, the file "post-pre.xls" must be downloaded and copied to your computer's hard drive. Once this is done open SAS. Click on "File," "Import Data…," and "Standard data source". Selecting "Microsoft Excel 97, 2000 or 2002 Workbook" from the pull down menu yields the following screen display:



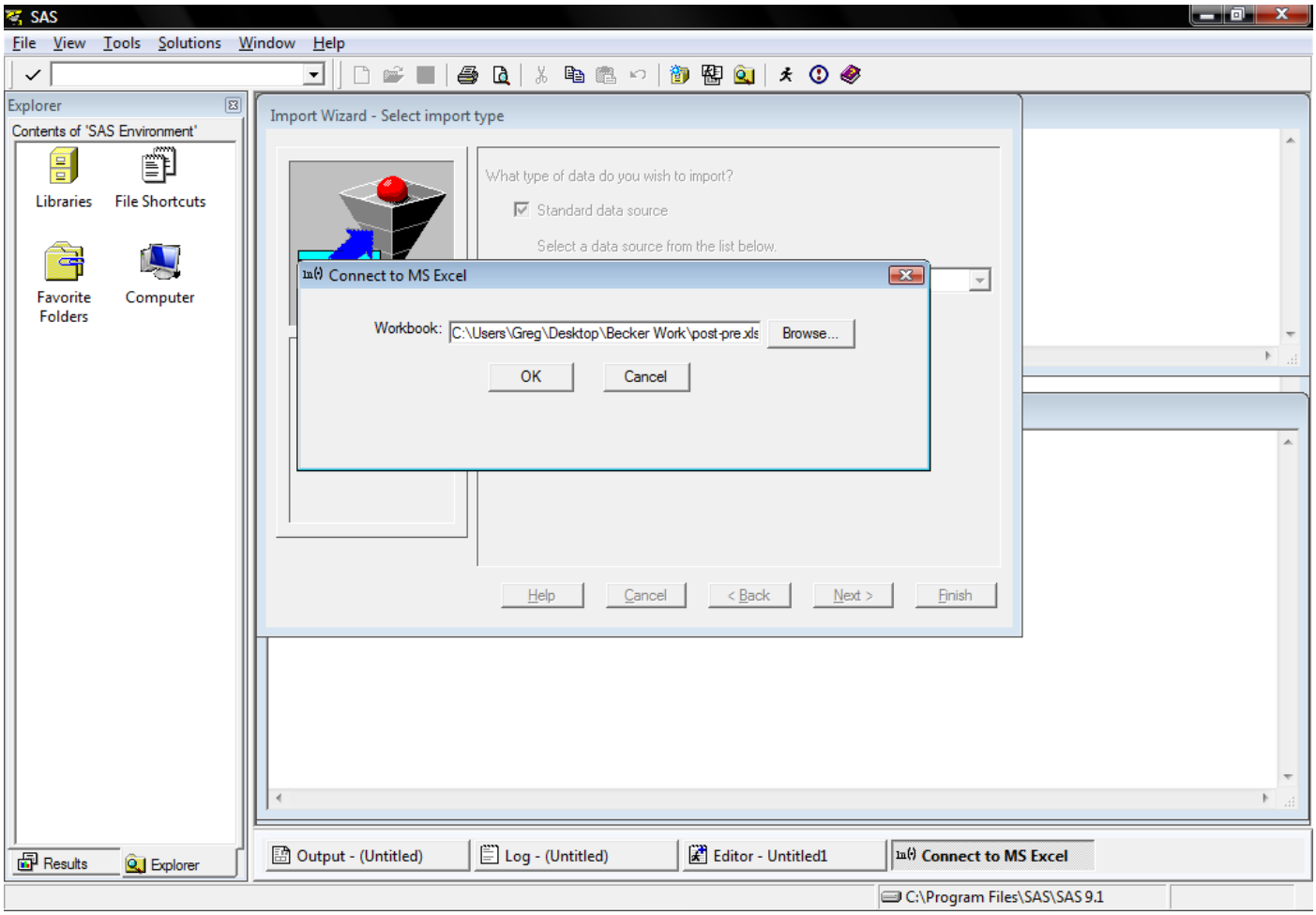
Clicking “Next” gives a pop-up screen window in which you can locate and specify the file containing the Excel file. Using the “Browse…” function is the simplest way to locate your file and is identical to opening any Microsoft file in MS Word or Excel.



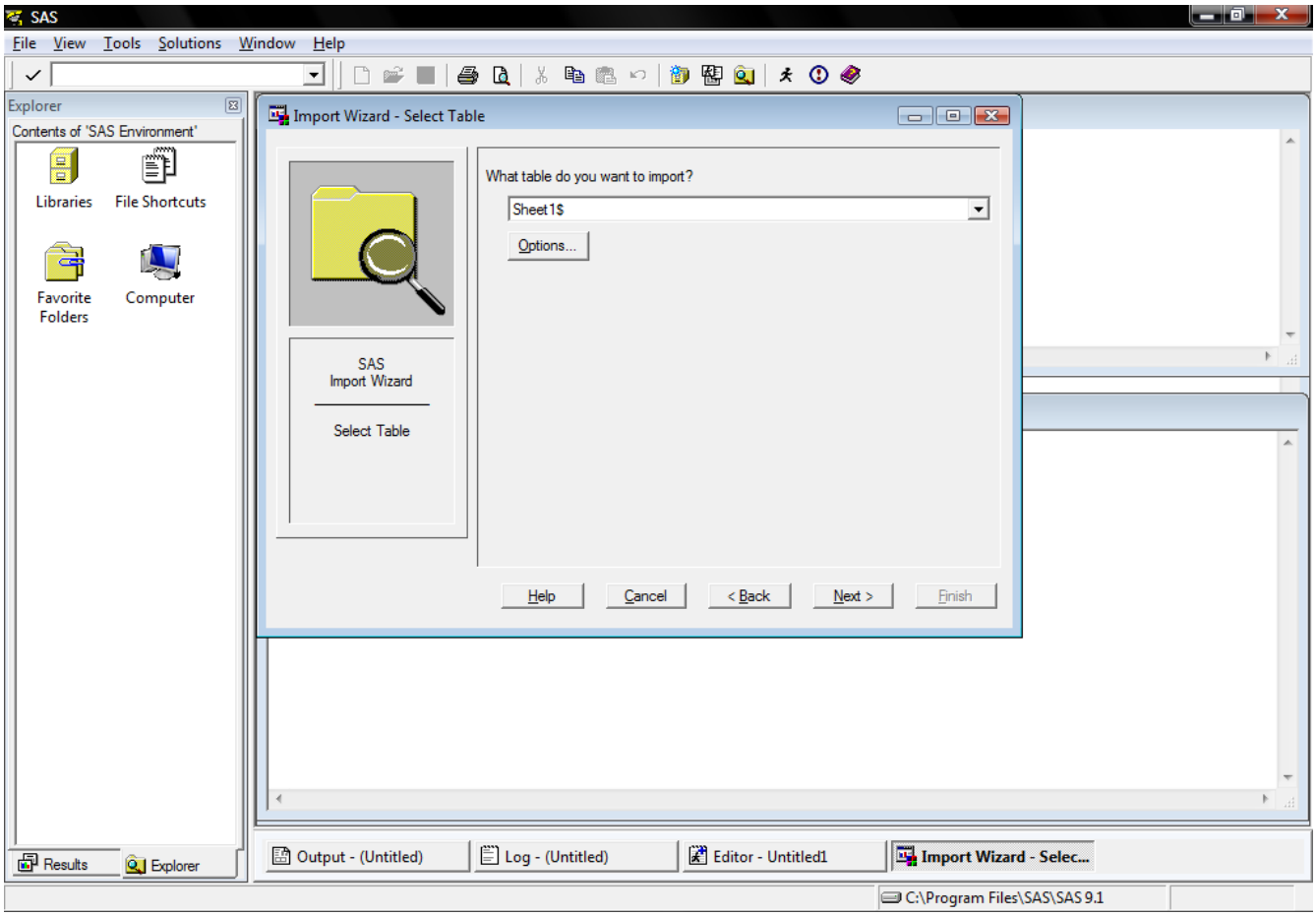
After you have located the file, click “Open”. SAS automatically fills in the path location on the “Connect to MS Excel” pop-up window (The path to “post-pre.xls” will obviously depend on where you placed it on your computer’s hard drive).



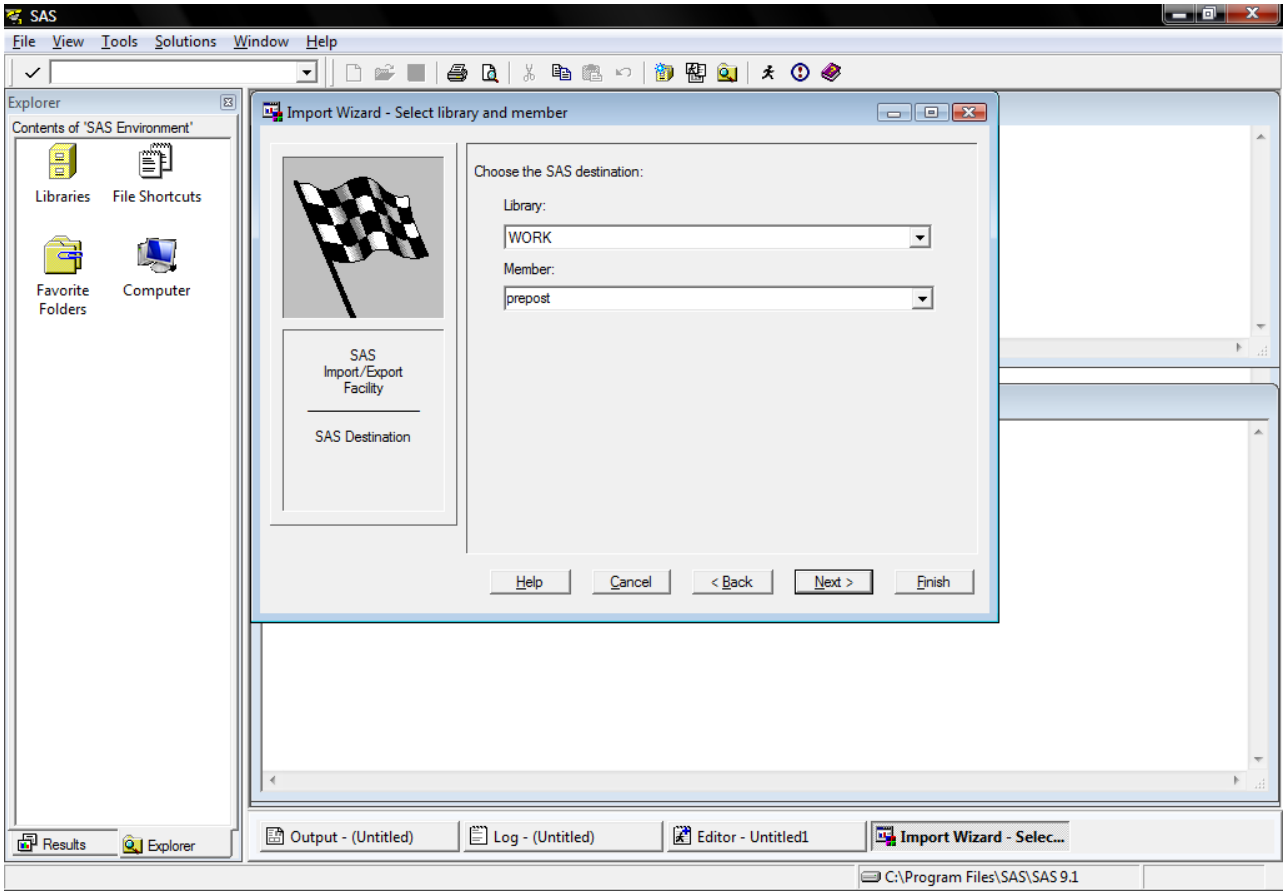
By clicking “OK” in the “Connect to MS Excel”, SAS now has the location of the data you wish to import.



Before the data is successfully imported, the table in your Excel file must be correctly specified. By default, SAS assumes that you want to import the first table (Excel labels this table “Sheet1”). If the data you are importing is on a different sheet, simply use the pull-down menu and click on the correct table name. After you have specified the table from your Excel file, click “Next”.


SAS
File View Tools Solutions Window Help
Explorer
Contents of 'SAS Environment'
Libraries
File Shortcuts
Favorite Folders
Computer
Results
Explorer
Import Wizard - Select Table
SAS Import Wizard
Select Table
What table do you want to import?
Sheet1$
Options...
Help
Cancel
< Back
Next >
Finish
Output - (Untitled)
Log - (Untitled)
Editor - Untitled1
Import Wizard - Selec...
C:\Program Files\SAS\SAS 9.1

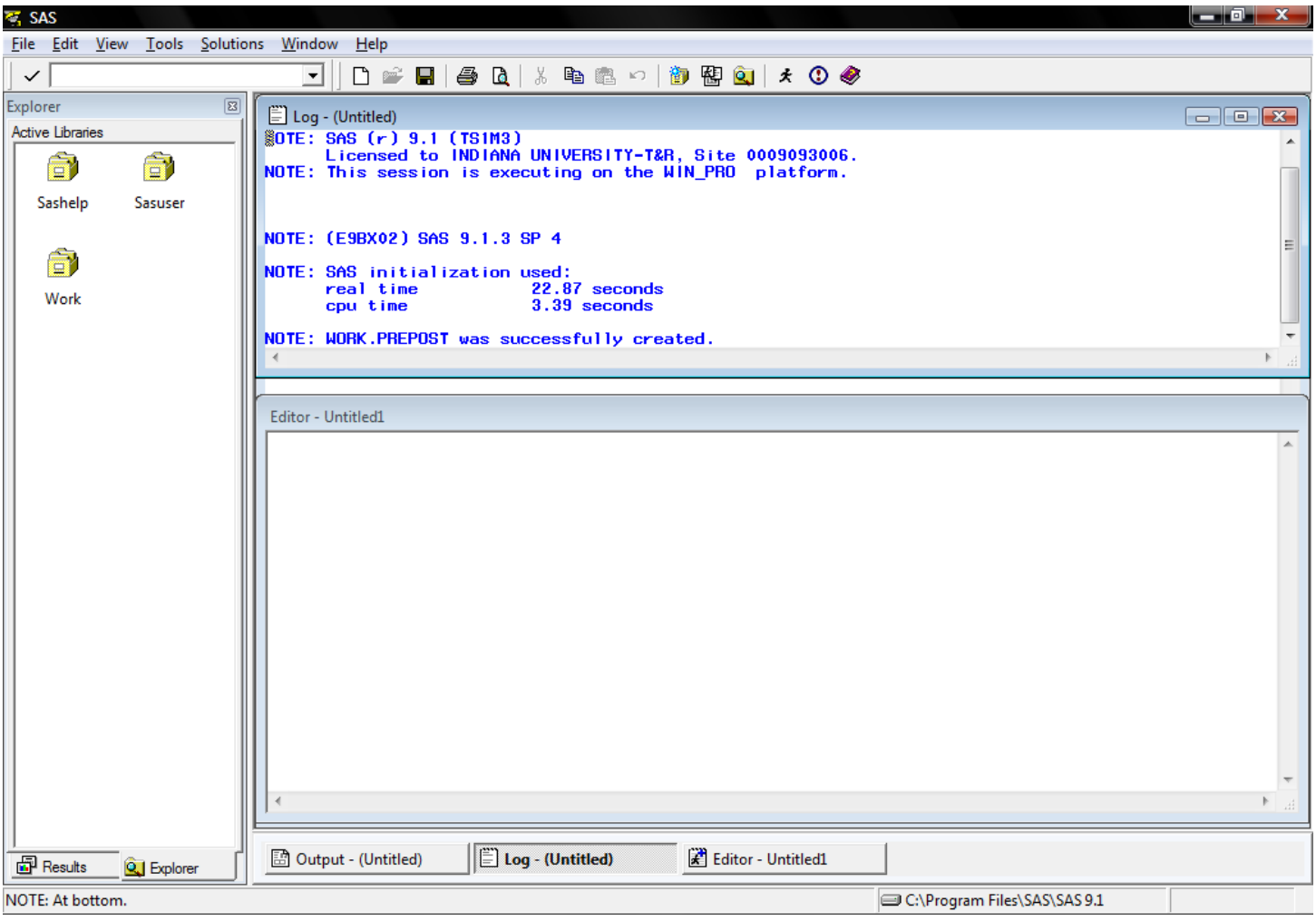
The last step required to import data is to name the file. In the “Member” field, enter the name you want to call your SAS data file. This can be different than the name of the Excel file which contains the data. There are some rules in naming files and SAS will promptly communicate to you if your naming is unacceptable and why. Note that SAS is not case specific. For demonstrative purposes, I have called the SAS dataset “prepost”. It is best to leave the “Library” set to the default “WORK”. Clicking “Finish” imports the file into SAS.

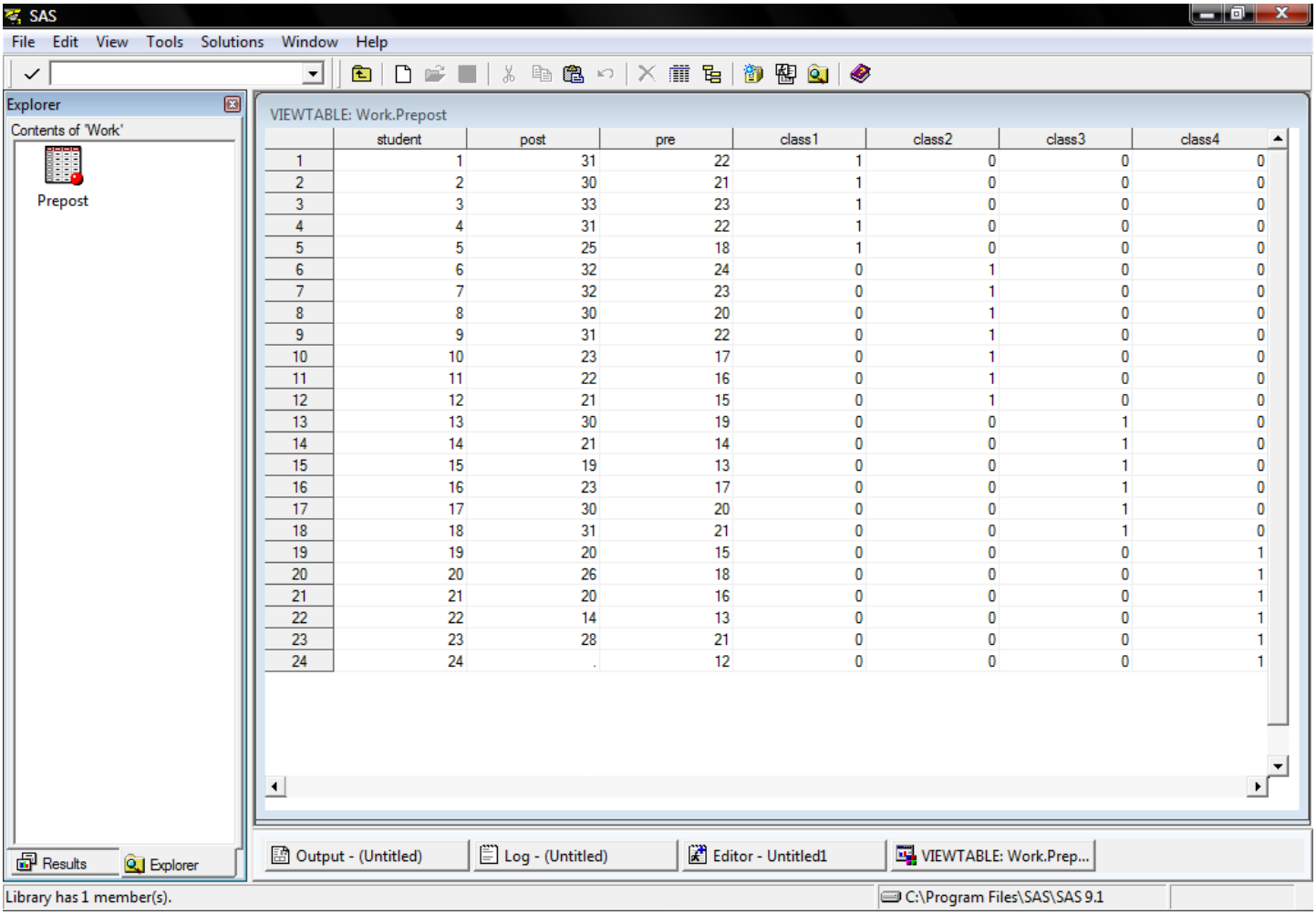


In SAS, the screen is broken up into three main sections: Program Editor, Log, and Explorer/Results. Each time SAS is told to do a command, SAS displays in the log what you requested and how it completed the task. To make sure the data has been correctly imported into SAS, we simply can read the log and verify SAS was able to successfully create the new dataset. The Log in the screen below indicates that WORK.PREPOST was successfully created where WORK indicates the Library and PREPOST is the name chosen. Libraries are the directories to where data sets are located.

The Work Library is a temporary location in the computer's RAM memory, which is permanently deleted once the user exits SAS. To retain your newly created SAS data set, you must place the dataset into a permanent Library.

To view the dataset, click on the "Work" library located in the Explorer section and then on the dataset "Prepost". Note that SAS, unlike LIMDEP, records missing observations with a period, rather than a blank space. The sequencing of these steps and the two associated screens follow:

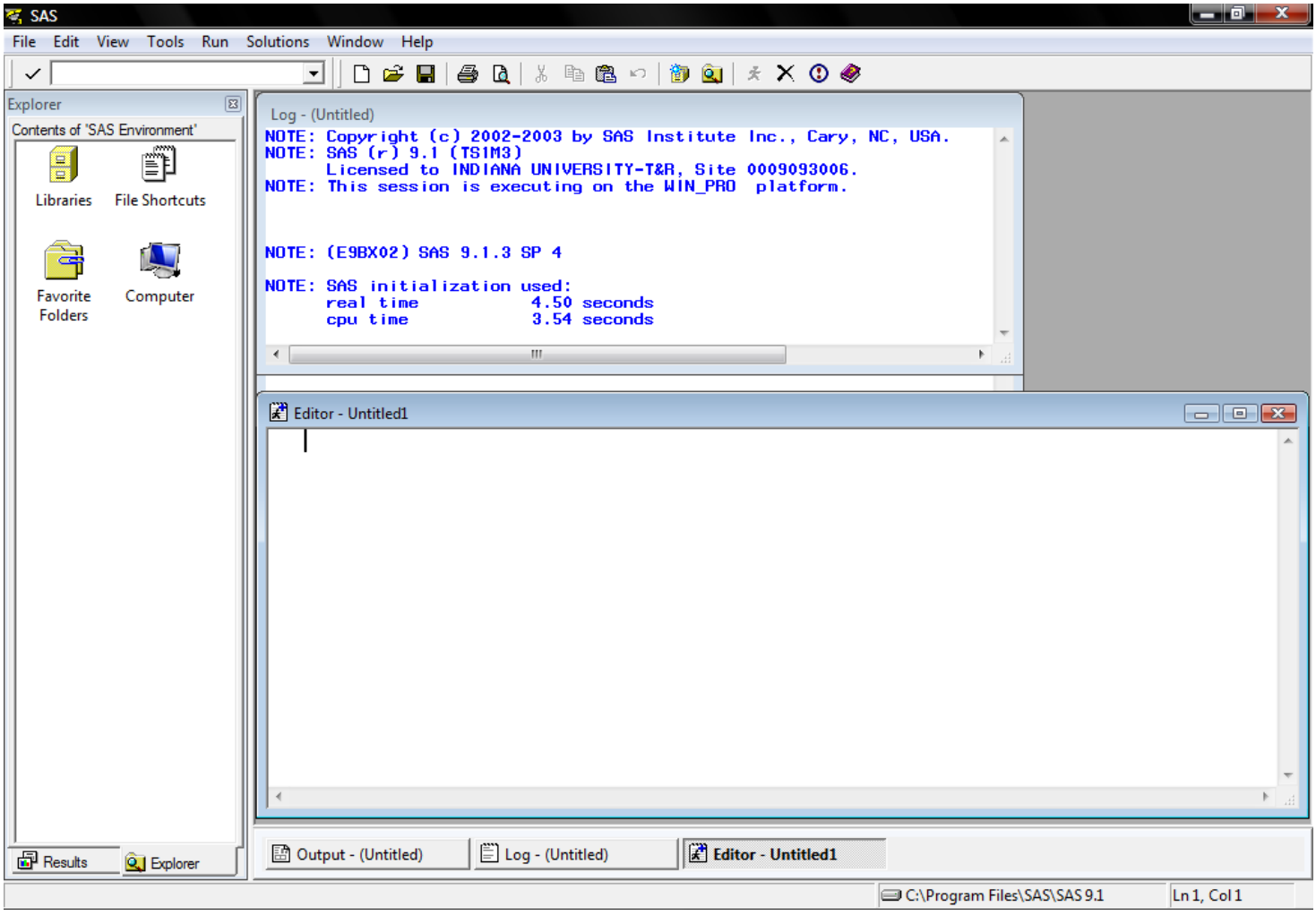




	student	post	pre	class1	class2	class3	class4
1	1	31	22	1	0	0	0
2	2	30	21	1	0	0	0
3	3	33	23	1	0	0	0
4	4	31	22	1	0	0	0
5	5	25	18	1	0	0	0
6	6	32	24	0	1	0	0
7	7	32	23	0	1	0	0
8	8	30	20	0	1	0	0
9	9	31	22	0	1	0	0
10	10	23	17	0	1	0	0
11	11	22	16	0	1	0	0
12	12	21	15	0	1	0	0
13	13	30	19	0	0	1	0
14	14	21	14	0	0	1	0
15	15	19	13	0	0	1	0
16	16	23	17	0	0	1	0
17	17	30	20	0	0	1	0
18	18	31	21	0	0	1	0
19	19	20	15	0	0	0	1
20	20	26	18	0	0	0	1
21	21	20	16	0	0	0	1
22	22	14	13	0	0	0	1
23	23	28	21	0	0	0	1
24	24	.	12	0	0	0	1

READING SPACE, TAB, OR COMMA DELINEATED FILES INTO SAS

Next we consider externally created text files that are typically accompanied by the ".txt" and "csv" extensions. For demonstration purposes, the data set just employed with 24 observations on the 7 variables ("student," "post," "pre," "class1," "class2," "class3," and "class4") was saved as the space delimited text file "post-pre.txt." After downloading this file to your hard drive, open SAS to its first screen:



To read the data into SAS, we need to utilize the “proc import” command. In the Editor window, type

proc import datafile = "F:\NCEE (Becker)\post-pre.txt"

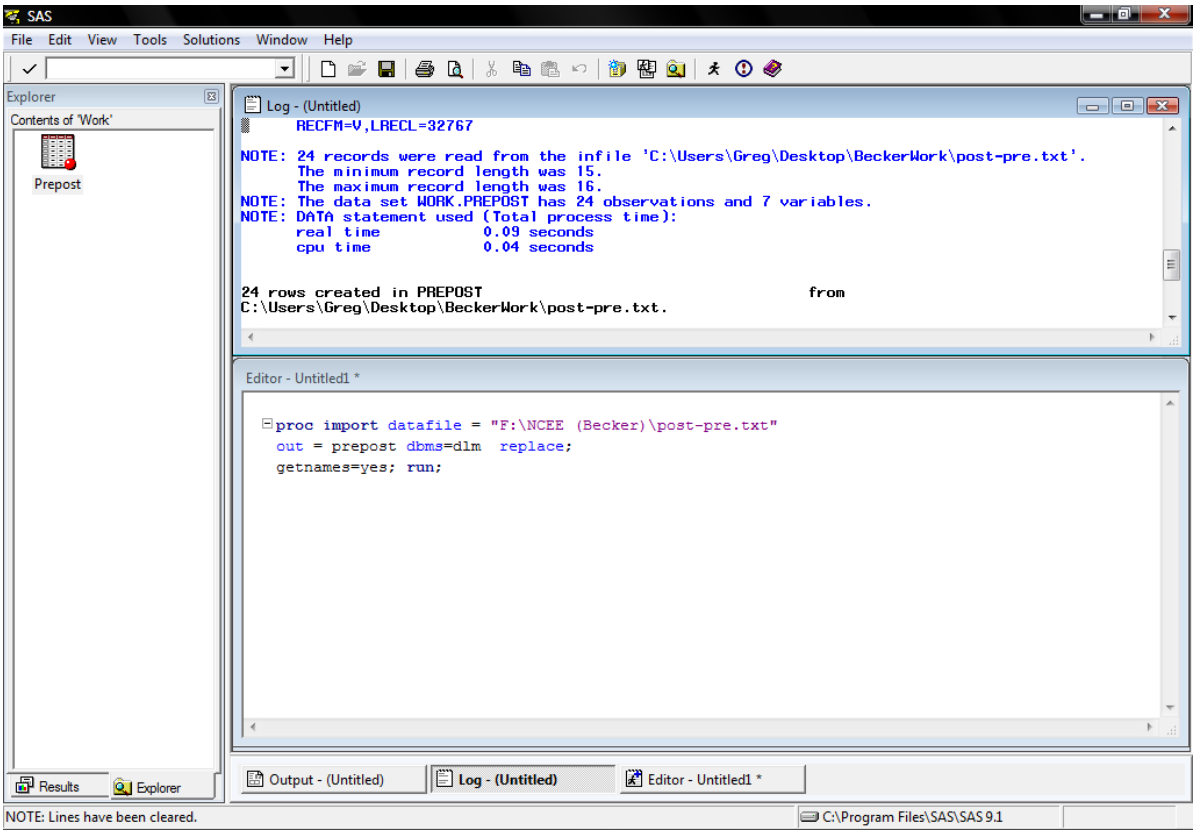
out = prepost dbms = dlm replace;

getnames=yes; run;

and then clicking the “run man” submit button. “proc import” tells SAS to read in text data and “datafile” directs SAS to a particular file name. In this case, the file is saved in the location “F:\NCEE (Becker)\post-pre.txt”, but this will vary by user. Finally, the “dbms=dlm” option tells SAS that the data points in this file are separated by a space.

If your data is tab delimited, change the “dbms” function to dbms = tab and if you were using a “.csv” file, change the “dbms” function to dbms=csv. In general, the “dlm” option is used when your data have a less standard delimiter (*e.g.*, a colon, semicolon, etc.).

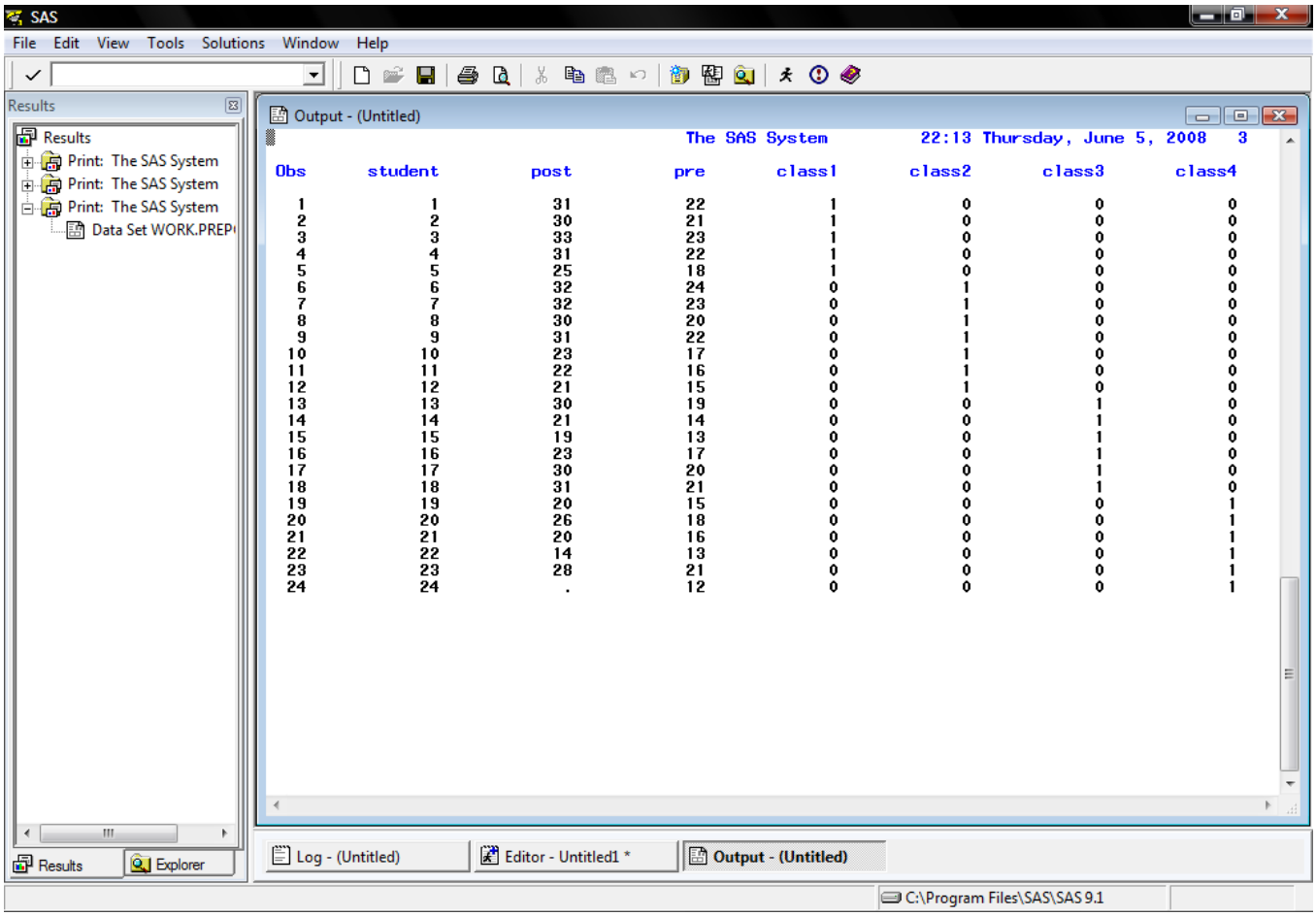
Once you’ve typed the appropriate command into the command window, press the submit button to run that line of text. This should yield the following screen:



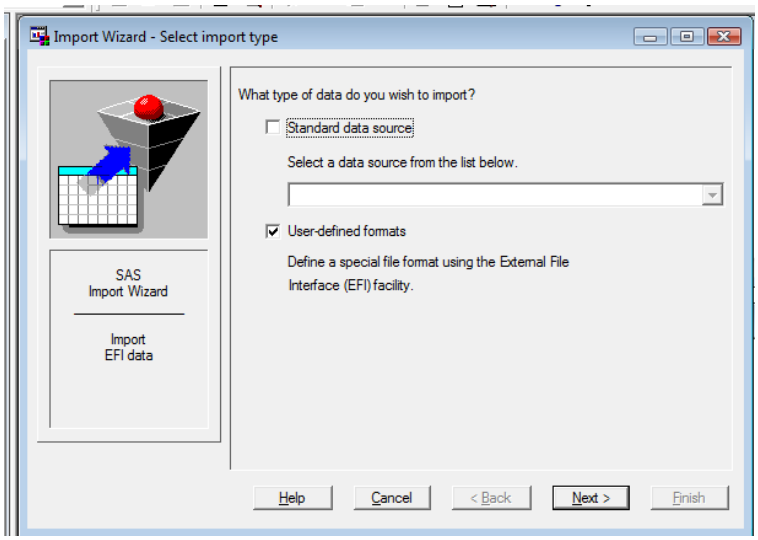
Just as before, the SAS log tells us that it has read a data set consisting of 7 variables and 24 observations, and we can access the dataset from the explorer window. We can view our data by typing

```
proc print data = prepost; run;
```

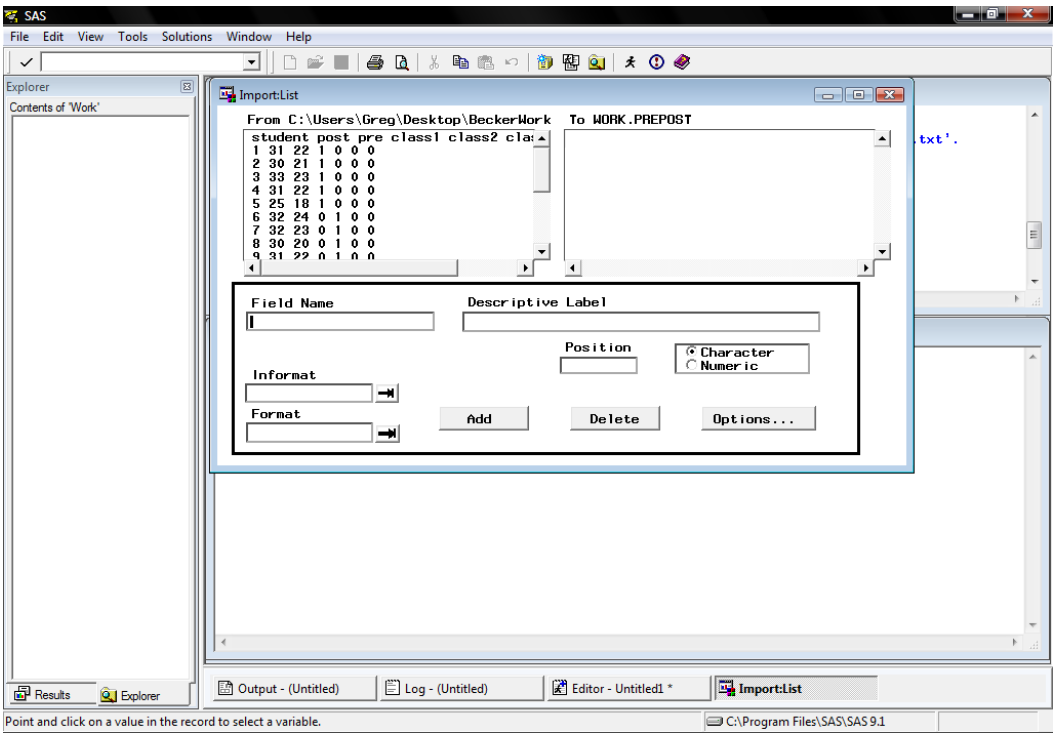
in the editor window, highlight it, and click the submit "Running man" button. The sequencing of these steps and the associated screen is as follows:



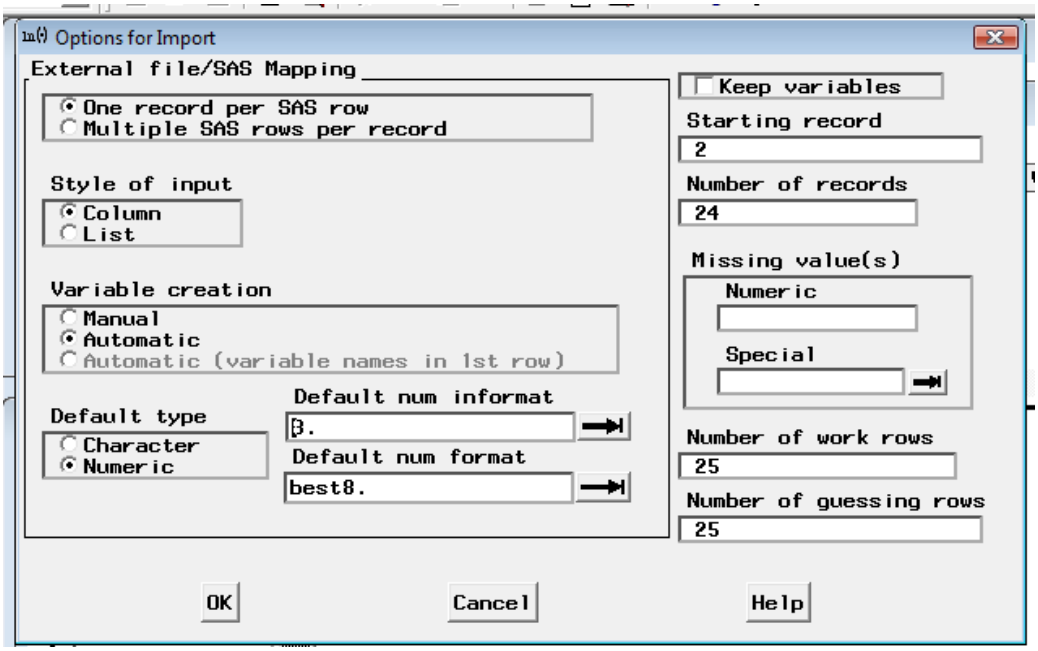
Some files contain multiple delimiters that can cause some difficulty in importing data. One way to address this is to use the "User-defined formats" option while using the import wizard. For demonstration purposes, we will import our "pre-post.txt" file using the user defined formats.

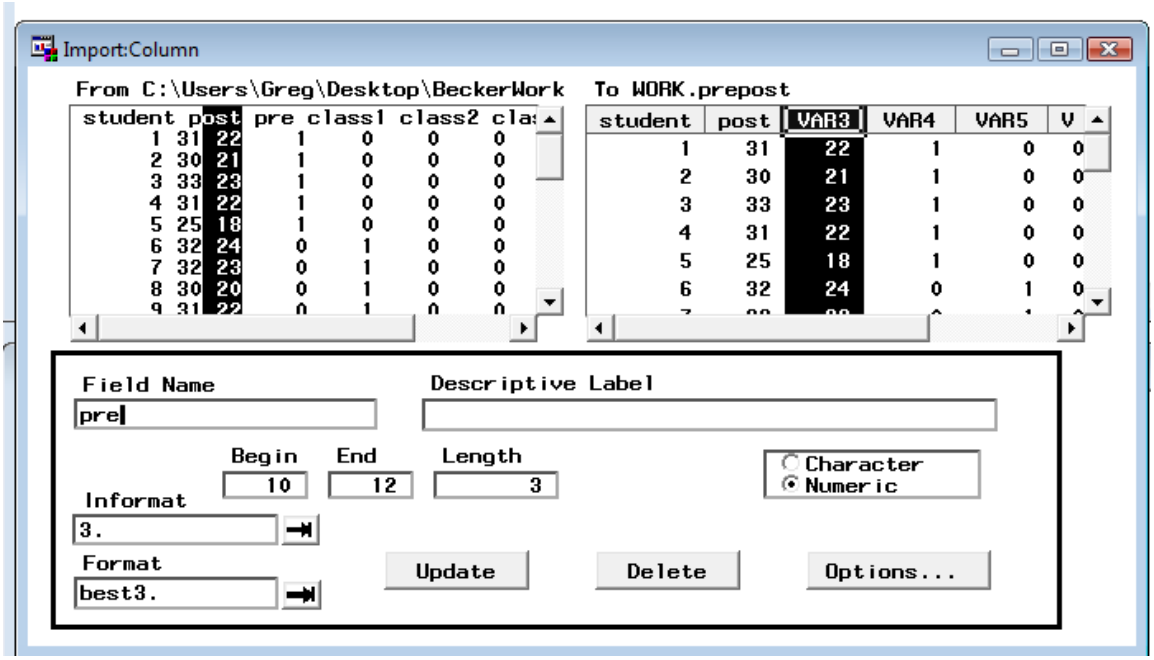


Click "File", "Import Data", "User-defined formats" and then "Next" After specifing the location of the file, click "Next" and name the dataset whatever you want to call it. We will use "prepost" as our SAS dataset name and then click "Next" and then "Finish". This activates the user defined format list where the variables can be properly identified and named. Below is a screen shot of the "Import: List".



In the user defined format wizard, click “Options...”. This will allow us to specify the type of data which we are dealing. As shown below, we define the style of input as “column” and allow automatic variable creation. Automatic variables inputted are numeric and starting record is row 2 (row one is the variable names). Specify the number of records as 24 and the number of working rows as 25. Click “Okay”. SAS prompts you to inspect the data in the “To: Work.Prepost” window for accuracy of importing the data. Also, under our current specification, variable names have been omitted. We can manually update the variable names now by clicking on the variable names in “To: Work.Prepost” and entering in the new name and then “Update” or later during a data step once we have finished importing the data. In this case because the number of variables is small, we manually enter the variable names. To finish importing the data, simply click the red box in the upper right-hand corner and then “save”. Your screens should look something like this:





READING LARGE FILES INTO SAS

SAS is extremely powerful in handling large datasets. There are few default restrictions on importing raw data files. The most common restriction is line length. When importing data one can specify the physical line length of the file using the LRECL command (see syntax below). As an example, consider the data set employed by Becker and Powers (2001), which initially had 2,837 records. The default restrictions are sufficient, so we need only follow the process of import a ".csv" file described above. Note, however, that this data set does not contain variable names in the top row. You can assign names yourself with a simple, but lengthy, addition to the importing command. Also note that we have specified what type of input variables the data contains and what type of variables we want them to be. "best32." corresponds to a numeric variable list of length 32.

```
data BECPOW;

 infile 'C:\Users\Greg\Desktop\BeckerWork\BECK8WO.CSV' delimiter = ','
MISSOVER DSD lrecl=32767 ;

informat A1 best32.; informat A2 best32.; informat X3 best32.;

informat C best32. ; informat AL best32.; informat AM best32.;

informat AN best32.; informat CA best32.; informat CB best32.;

informat CC best32.; informat CH best32.; informat CI best32.;

informat CJ best32.; informat CK best32.; informat CL best32.;
```

```
informat CM best32.; informat CN best32.; informat CO best32.;
informat CS best32.; informat CT best32.; informat CU best32.;
informat CV best32.; informat CW best32.; informat DB best32.;
informat DD best32.; informat DI best32.; informat DJ best32.;
informat DK best32.; informat DL best32.; informat DM best32.;
informat DN best32.; informat DQ best32.; informat DR best32.;
informat DS best32.; informat DY best32.; informat DZ best32.;
informat EA best32.; informat EB best32.; informat EE best32.;
informat EF best32.; informat EI best32.; informat EJ best32.;
informat EP best32.; informat EQ best32.; informat ER best32.;
informat ET best32.; informat EY best32.; informat EZ best32.;
informat FF best32.; informat FN best32.; informat FX best32.;
informat FY best32.; informat FZ best32.; informat GE best32.;
informat GH best32.; informat GM best32.; informat GN best32.;
informat GQ best32.; informat GR best32.; informat HB best32.;
informat HC best32.; informat HD best32.; informat HE best32.;
informat HF best32.;

format A1 best12.; format A2 best12.; format X3 best12.;
format C best12. ; format AL best12.; format AM best12.;
format AN best12.; format CA best12.; format CB best12.;
format CC best12.; format CH best12.; format CI best12.;
format CJ best12.; format CK best12.; format CL best12.;
format CM best12.; format CN best12.; format CO best12.;
format CS best12.; format CT best12.; format CU best12.;
```

```
format CV best12.; format CW best12.; format DB best12.;
format DD best12.; format DI best12.; format DJ best12.;
format DK best12.; format DL best12.; format DM best12.;
format DN best12.; format DQ best12.; format DR best12.;
format DS best12.; format DY best12.; format DZ best12.;
format EA best12.; format EB best12.; format EE best12.;
format EF best12.; format EI best12.; format EJ best12.;
format EP best12.; format EQ best12.; format ER best12.;
format ET best12.; format EY best12.; format EZ best12.;
format FF best12.; format FN best12.; format FX best12.;
format FY best12.; format FZ best12.; format GE best12.;
format GH best12.; format GM best12.; format GN best12.;
format GQ best12.; format GR best12.; format HB best12.;
format HC best12.; format HD best12.; format HE best12.;
format HF best12.;
input
A1 A2 X3 C AL AM AN CA CB CC CH CI CJ CK CL CM CN CO CS CT CU
CV CW DB DD DI DJ DK DL DM DN DQ DR DS DY DZ EA EB EE EF
EI EJ EP EQ ER ET EY EZ FF FN FX FY FZ GE GH GM GN GQ GR HB
HC HD HE HF; run;
```

For more details on how to import data sets with data dictionaries (*i.e.,* variable names and definitions in external files), try typing "infile" into the "SAS Help and Documentation" under the Help tab. If you do not assign variable names, then SAS will provide default variable names of "var1, var2, var3, etc.".

LEAST-SQUARES ESTIMATION AND LINEAR RESTRICTIONS IN SAS

As in the previous Module One, Part Two using LIMDEP, we now demonstrate some various regression tools in SAS using the "post-pre" data set. Recall the model being estimated is

$$post = \beta_1 + \beta_2 pre + f(classes) + \varepsilon .$$

SAS automatically drops any missing observations from our analysis, so we need not restrict the data in any of our commands. In general, the syntax for a basic OLS regression in SAS is

proc reg data = FILENAME; model *y-variable* = *x-variables*; run;

where *y-variable* is just the independent variable name and *x-variables* are the dependent variable names.

Once you have your data read into SAS, we estimate the model

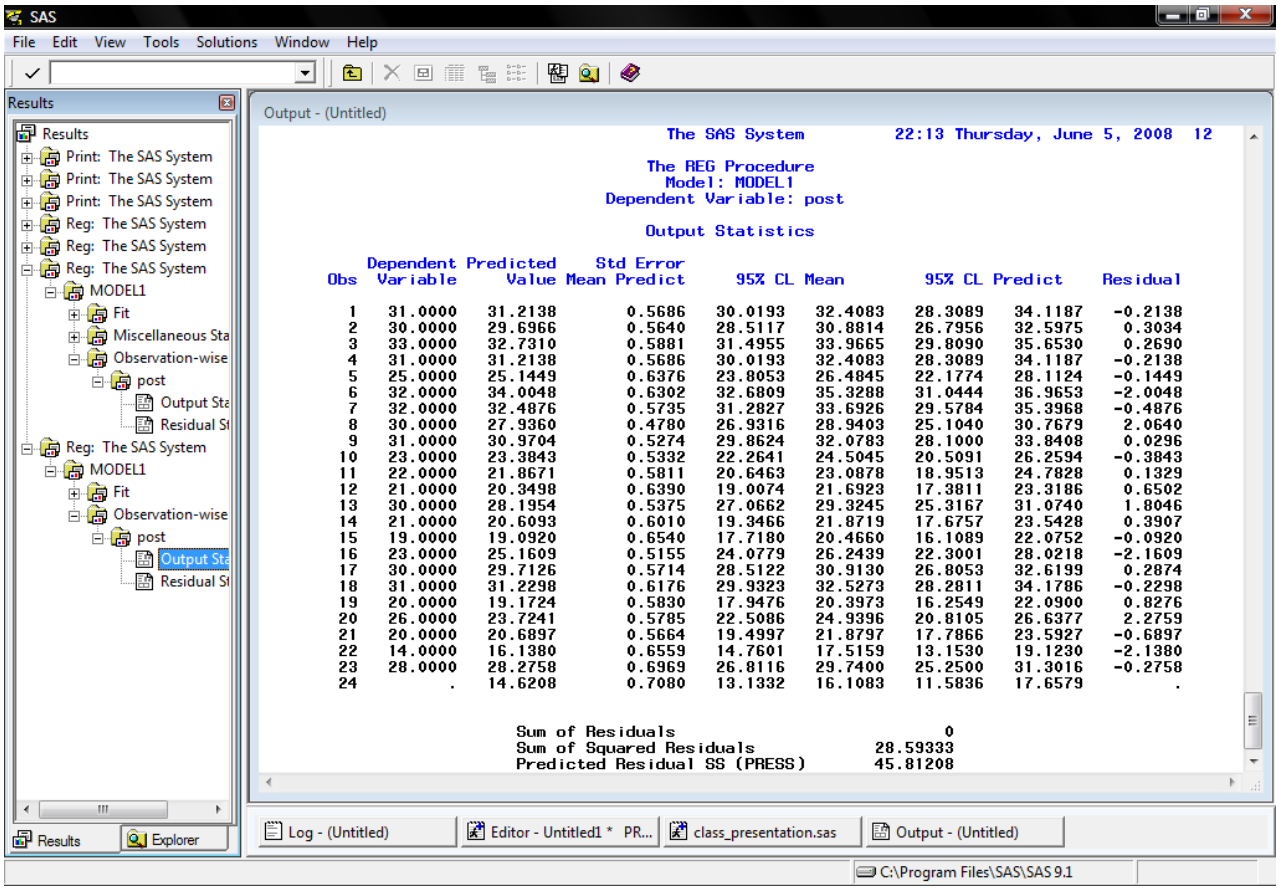
$$post = \beta_1 + \beta_2 pre + \beta_3 class1 + \beta_4 class2 + \beta_5 class3 + \varepsilon$$

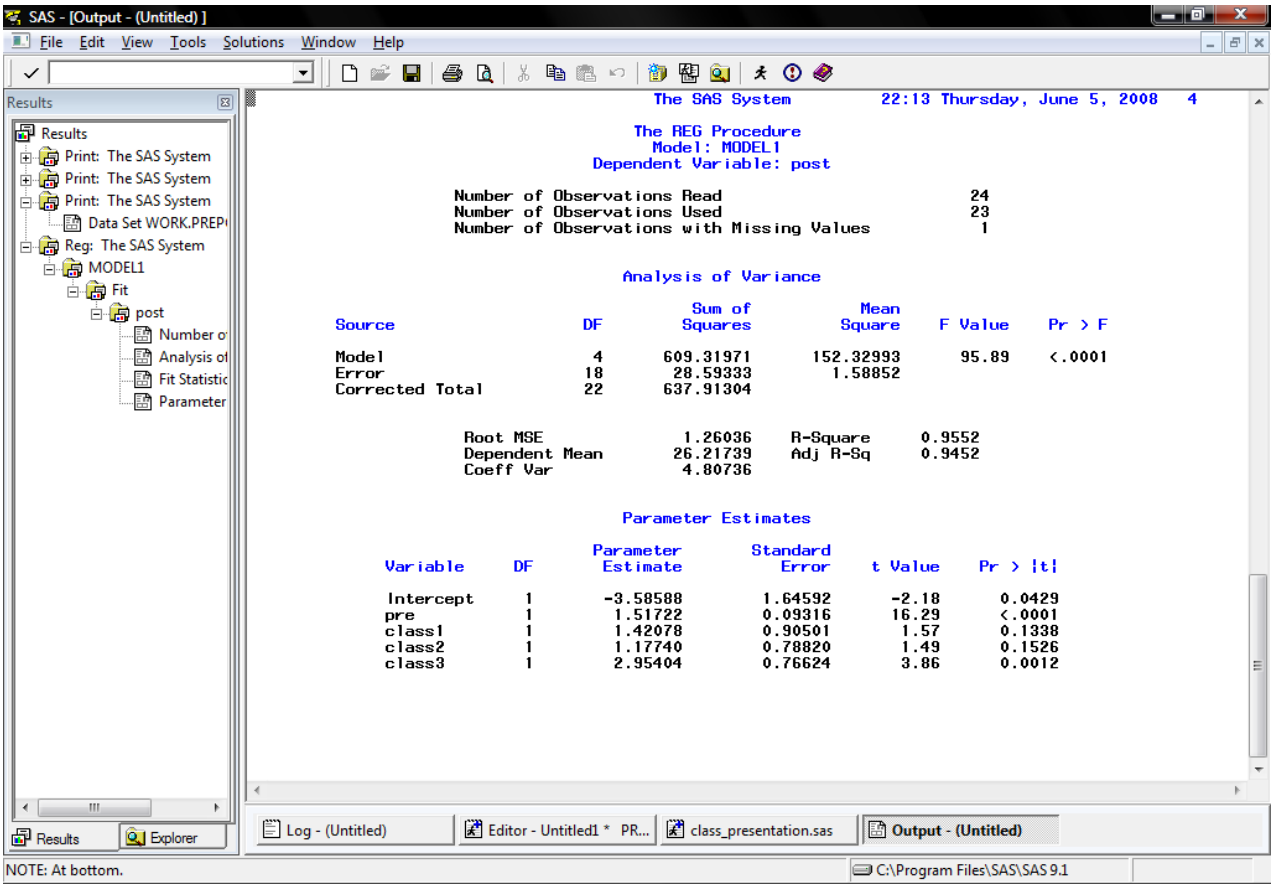
by typing:

proc reg data = prepost; model post = pre class1 class2 class3 / p cli clm; run;

into the editor window and pressing submit. Typing "/ p cli clm" after specifying the model outputs predicted value and a 95% confidence interval. From the output the predicted posttest score is 14.621, with 95 percent confidence interval equal to 11.5836< E(y|$\mathbf{X_{24}}$)< 17.6579.

We get the following output:


SAS
File Edit View Tools Solutions Window Help
Results
Output - (Untitled)
The SAS System 22:13 Thursday, June 5, 2008 12
The REG Procedure
Model: MODEL1
Dependent Variable: post
Output Statistics
Obs Dependent Variable Predicted Value Std Error Mean Predict 95% CL Mean 95% CL Predict Residual
1 31.0000 31.2138 0.5686 30.0193 32.4083 28.3089 34.1187 -0.2138
2 30.0000 29.6966 0.5640 28.5117 30.8814 26.7956 32.5975 0.3034
3 33.0000 32.7310 0.5881 31.4955 33.9665 29.8090 35.6530 0.2690
4 31.0000 31.2138 0.5686 30.0193 32.4083 28.3089 34.1187 -0.2138
5 25.0000 25.1449 0.6376 23.8053 26.4845 22.1774 28.1124 -0.1449
6 32.0000 34.0048 0.6302 32.6809 35.3288 31.0444 36.9653 -2.0048
7 32.0000 32.4876 0.5735 31.2827 33.6926 29.5784 35.3968 -0.4876
8 30.0000 27.9360 0.4780 26.9316 28.9403 25.1040 30.7679 2.0640
9 31.0000 30.9704 0.5274 29.8624 32.0783 28.1000 33.8408 0.0296
10 23.0000 23.3843 0.5332 22.2641 24.5045 20.5091 26.2594 -0.3843
11 22.0000 21.8671 0.5811 20.6463 23.0878 18.9513 24.7828 0.1329
12 21.0000 20.3498 0.6390 19.0074 21.6923 17.3811 23.3186 0.6502
13 30.0000 28.1954 0.5375 27.0662 29.3245 25.3167 31.0740 1.8046
14 21.0000 20.6093 0.6010 19.3466 21.8719 17.6757 23.5428 0.3907
15 19.0000 19.0920 0.6540 17.7180 20.4660 16.1089 22.0752 -0.0920
16 23.0000 25.1609 0.5155 24.0779 26.2439 22.3001 28.0218 -2.1609
17 30.0000 29.7126 0.5714 28.5122 30.9130 26.8053 32.6199 0.2874
18 31.0000 31.2298 0.6176 29.9323 32.5273 28.2811 34.1786 -0.2298
19 20.0000 19.1724 0.5830 17.9476 20.3973 16.2549 22.0900 0.8276
20 26.0000 23.7241 0.5785 22.5086 24.9396 20.8105 26.6377 2.2759
21 20.0000 20.6897 0.5664 19.4997 21.8797 17.7866 23.5927 -0.6897
22 14.0000 16.1380 0.6559 14.7601 17.5159 13.1530 19.1230 -2.1380
23 28.0000 28.2758 0.6969 26.8116 29.7400 25.2500 31.3016 -0.2758
24 . 14.6208 0.7080 13.1332 16.1083 11.5836 17.6579 .
Sum of Residuals 0
Sum of Squared Residuals 28.59333
Predicted Residual SS (PRESS) 45.81208
Log - (Untitled)
Editor - Untitled1 * PR...
class_presentation.sas
Output - (Untitled)
C:\Program Files\SAS\SAS 9.1


SAS - [Output - (Untitled)]
File Edit View Tools Solutions Window Help
Results
The SAS System 22:13 Thursday, June 5, 2008 4
The REG Procedure
Model: MODEL1
Dependent Variable: post
Number of Observations Read 24
Number of Observations Used 23
Number of Observations with Missing Values 1
Analysis of Variance
Source DF Sum of Squares Mean Square F Value Pr > F
Model 4 609.31971 152.32993 95.89 <.0001
Error 18 28.59333 1.58852
Corrected Total 22 637.91304
Root MSE 1.26036 R-Square 0.9552
Dependent Mean 26.21739 Adj R-Sq 0.9452
Coeff Var 4.80736
Parameter Estimates
Variable DF Parameter Estimate Standard Error t Value Pr > |t|
Intercept 1 -3.58588 1.64592 -2.18 0.0429
pre 1 1.51722 0.09316 16.29 <.0001
class1 1 1.42078 0.90501 1.57 0.1338
class2 1 1.17740 0.78820 1.49 0.1526
class3 1 2.95404 0.76624 3.86 0.0012
Log - (Untitled)
Editor - Untitled1 * PR...
class_presentation.sas
Output - (Untitled)
NOTE: At bottom.
C:\Program Files\SAS\SAS 9.1

A researcher might be interested to test whether the class in which a student is enrolled affects his/her post-course test score, assuming fixed effects only. This linear restriction is done automatically in SAS by adding the following "test" command to the regression statement in the Editor Text.

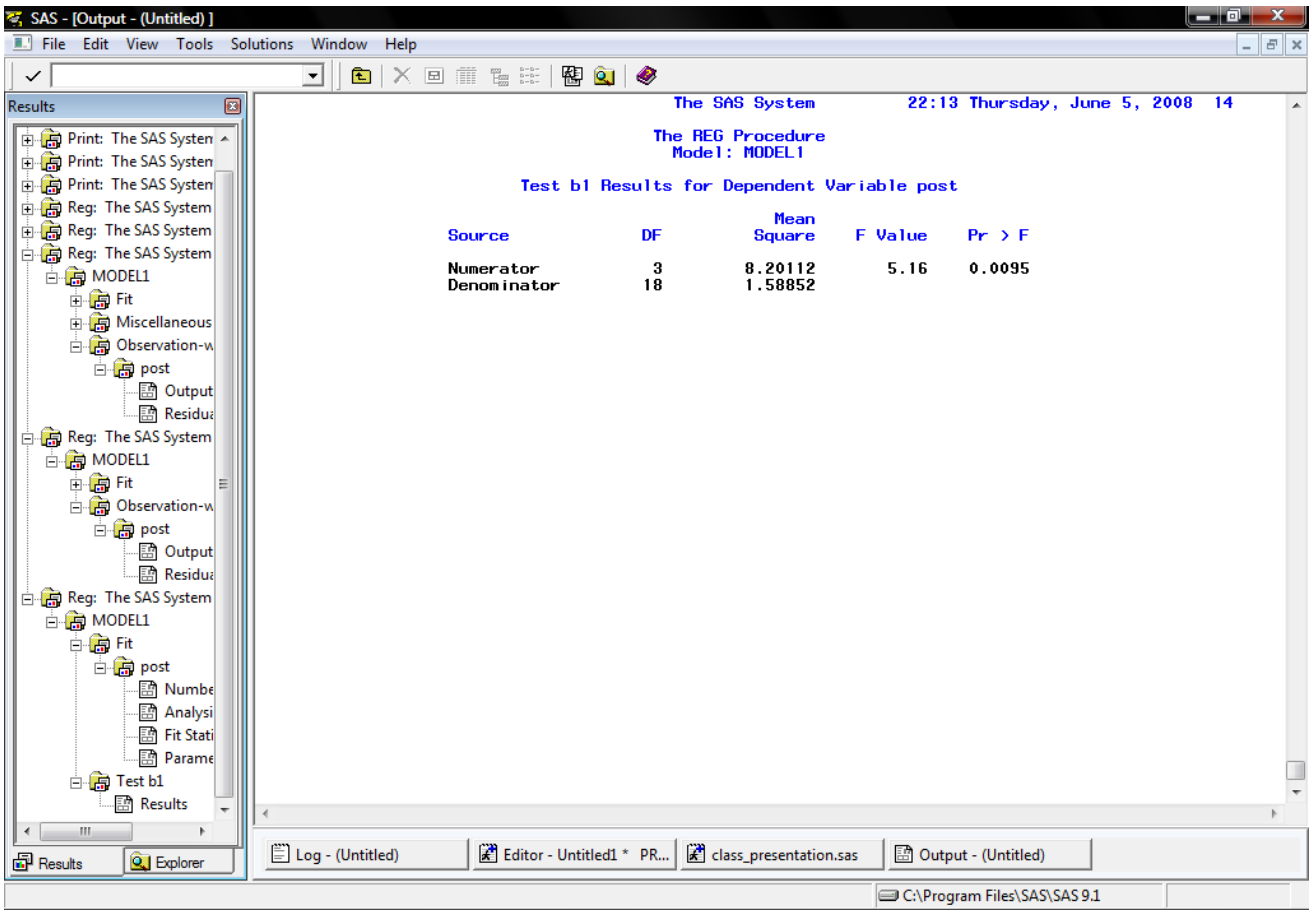
```
proc reg data = prepost; model post = pre class1 class2 class3;
        b1:   test class1=0, class2=0, class3=0;
run;
```

In this case we have named the test "b1". Upon highlighting and pressing the submit button, SAS automatically forms the correct test statistic, and we see

$$F(3, 18) = 5.16$$
$$\text{Prob} > F = 0.0095$$

The first line gives us the value of the *F* statistic and the associated P-value, where we see that we can reject the null that all class coefficients are zero at any probability of Type I error greater than 0.0095.

The following results will appear in the output:



TEST FOR A STRUCTURAL BREAK (CHOW TEST)

The above test of the linear restriction $\beta 3 = \beta 4 = \beta 5 = 0$ (no difference among classes), assumed that the pretest slope coefficient was constant, fixed and unaffected by the class to which a student belonged. A full structural test requires the fitting of four separate regressions to obtain the four residual sum of squares that are added to obtain the unrestricted sum of squares. The restricted sum of squares is obtained from a regression of posttest on pretest with no dummies for the classes; that is, the class to which a student belongs is irrelevant in the manner in which pretests determined the posttest score.

We can perform this test one of two ways. One, we can run each restricted regression and the unrestricted regression, take note of the residual sums of squares from each regression, and explicitly calculate the *F* statistic. We already know how to run basic regressions in SAS, so the new part is how to run a restricted regression. For this, we create a new variable that identifies the restricted observations; we want to run the regression separately and then we can run all the restrictions simultaneously. We first create the restriction variable "class" by typing into the editor window:

```
data prepost; set prepost; if class1 = 1 then class = 1; if class2 = 1 then class = 2;
if class3 = 1 then class = 3; if class4 = 1 then class = 4; run;
```

Highlight the command and click on the submit button. We now run all four restricted regressions simultaneously by typing:

```
proc reg data = prepost; model post = pre; by class; run;
```

into the editor window, highlighting the text and press submit. The resulting output is as follows:

---------- class=1 ----------

The REG Procedure
Model: MODEL1
Dependent Variable: post post

Number of Observations Read	5
Number of Observations Used	5

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	35.74324	35.74324	417.63	0.0003
Error	3	0.25676	0.08559		
Corrected Total	4	36.00000			

Root MSE	0.29255	R-Square	0.9929
Dependent Mean	30.00000	Adj R-Sq	0.9905
Coeff Var	0.97517		

Parameter Estimates

Variable	Label	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	Intercept	1	-2.94595	1.61745	-1.82	0.1661
pre	pre	1	1.55405	0.07604	20.44	0.0003

ns

The structural test across all classes is

$$H_0 : \boldsymbol{\beta}_1 = \boldsymbol{\beta}_2 = \ldots = \boldsymbol{\beta}_4 \text{ and } H_a : \boldsymbol{\beta}\text{'s are not equal}$$

$$F = \frac{(ErrorSS_r - ErrorSS_u)/2(J-1K)}{ErrorSS_u/(n-JK)}$$

Because the calculated $F = 2.92$, and the critical F (Prob of Type I error =0.05, df1=6, df2=15) = 2.79, we reject the null hypothesis and conclude that at least one class is significantly difference from another, allowing for the slope on pre-course test score to vary is from one class to another. That is, the class in which a student is enrolled is important because of a change in slope and or intercept.

The second way to test for a structural break is to create several interaction terms and test whether the dummy and interaction terms are jointly significantly different from zero. To perform the Chow test this way, we first generate interaction terms between all dummy variables and independent variables. To do this in SAS, type the following into the editor window, highlight it and press submit:

```
data prepost; set prepost; pre_c1=pre*class1; pre_c2=pre*class2; pre_c3=pre*class3; run;
```

With our new variables created, we now run a regression with all dummy and interaction terms included, as well as the original independent variable and run the *F* test. In SAS, we need to type

```
proc reg data = prepost; model post = pre class1 class2 class3 pre_c1 pre_c2 pre_c3;

b3: test class1, class2, class3, pre_c1, pre_c2, pre_c3; run;
```

into the editor window, highlight it, and press enter. The output for this regression is not meaningful, as it is only the test that we're interested in. The resulting output is:

```
 F(  6,    15) =    2.93
      Prob > F =    0.0427
```

Just as we could see in LIMDEP, our F statistic is 2.93, with a P-value of 0.0427. We again reject the null (at a probability of Type I error=0.05) and conclude that class is important either through the slope or intercept coefficients. This type of test will always yield results identical to the restricted regression approach.

HETEROSCEDASTICITY

You can control for heteroscedasticity across observations or within specific groups (in this case, within a given class, but not across classes) by specifying the "robust" or "cluster" option, respectively, at the end of your regression command.

To account for a common error term within groups, but not across groups, we create a class variable that identifies each student into one of the 4 classes. This is used to specify which group (or cluster) a student is in. To generate this variable, type:

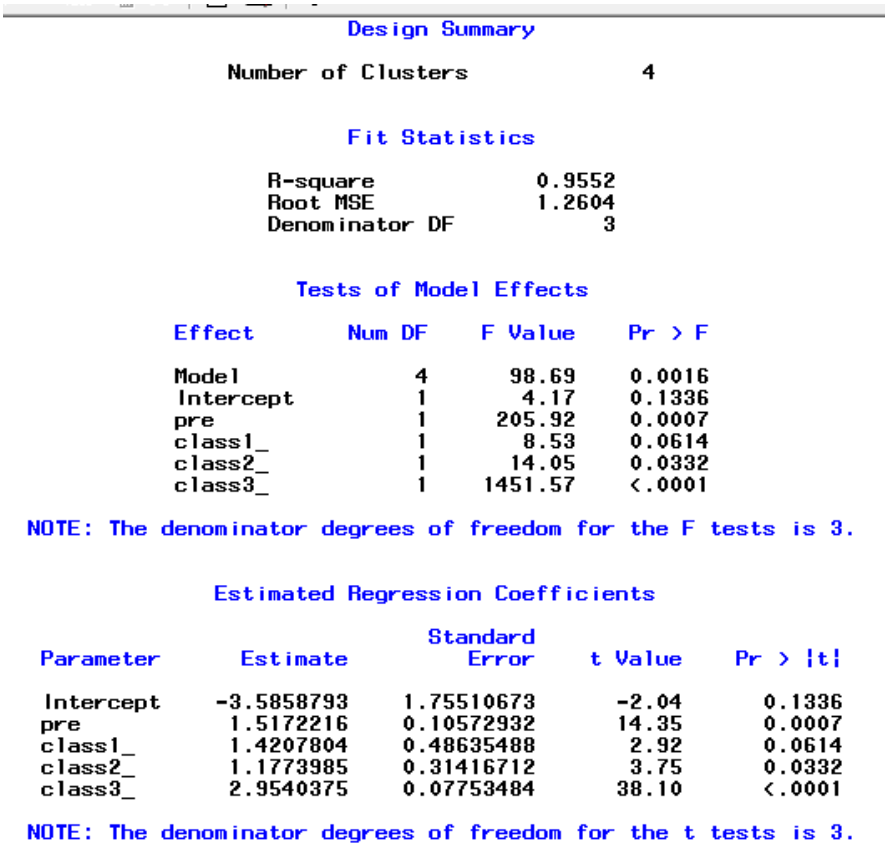
```
data prepost; set prepost; if class1 = 1 then class = 1; if class2 = 1 then class = 2;

if class3 = 1 then class = 3; if class4 = 1 then class = 4; run;
```

Highlight the command and click on the submit button.

Then to allow for clustered error terms, our regression command is:

```
proc surveyreg data=prepost; cluster class;

    model post = pre class1 class2 class3; run;
```

This gives us the following output:



Design Summary

Number of Clusters	4

Fit Statistics

R-square	0.9552
Root MSE	1.2604
Denominator DF	3

Tests of Model Effects

Effect	Num DF	F Value	Pr > F
Model	4	98.69	0.0016
Intercept	1	4.17	0.1336
pre	1	205.92	0.0007
class1_	1	8.53	0.0614
class2_	1	14.05	0.0332
class3_	1	1451.57	<.0001

NOTE: The denominator degrees of freedom for the F tests is 3.

Estimated Regression Coefficients

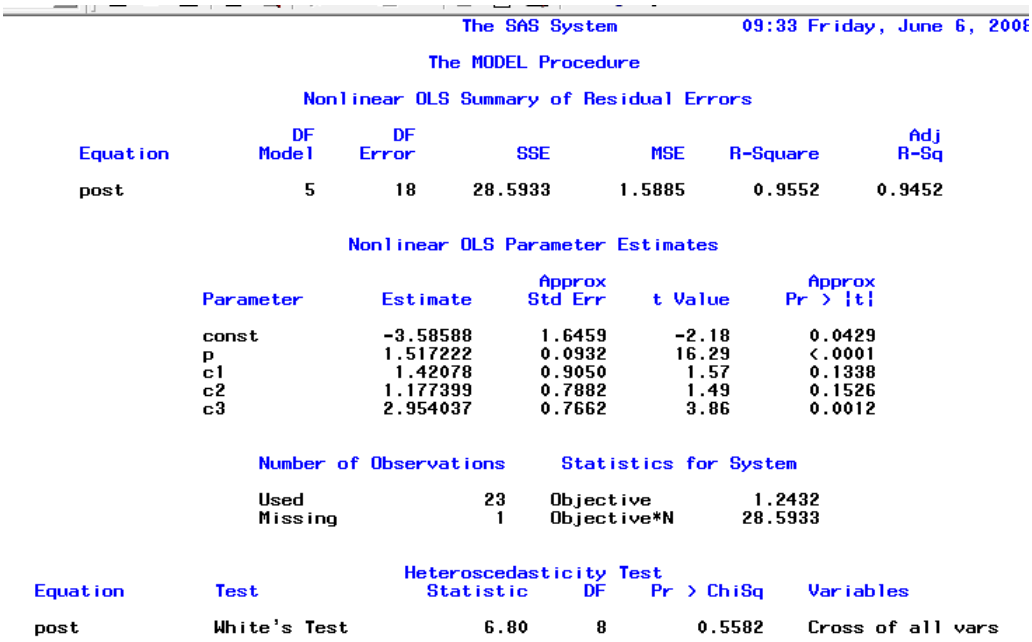
Parameter	Estimate	Standard Error	t Value	Pr > |t|
Intercept	-3.5858793	1.75510673	-2.04	0.1336
pre	1.5172216	0.10572932	14.35	0.0007
class1_	1.4207804	0.48635488	2.92	0.0614
class2_	1.1773985	0.31416712	3.75	0.0332
class3_	2.9540375	0.07753484	38.10	<.0001

NOTE: The denominator degrees of freedom for the t tests is 3.

Similarly, to account for general heteroscedasticity across individual observations, our regression command is:

```
proc model data=prepost; parms const p c1 c2 c3;
    post = const + p*pre + c1*class1 + c2*class2 + c3*class3;
    fit post / white; run; quit;
```

and we get the following output:



The SAS System 09:33 Friday, June 6, 2008

The MODEL Procedure

Nonlinear OLS Summary of Residual Errors

Equation	DF Model	DF Error	SSE	MSE	R-Square	Adj R-Sq
post	5	18	28.5933	1.5885	0.9552	0.9452

Nonlinear OLS Parameter Estimates

Parameter	Estimate	Approx Std Err	t Value	Approx Pr > |t|
const	-3.58588	1.6459	-2.18	0.0429
p	1.517222	0.0932	16.29	<.0001
c1	1.42078	0.9050	1.57	0.1338
c2	1.177399	0.7882	1.49	0.1526
c3	2.954037	0.7662	3.86	0.0012

Number of Observations		Statistics for System	
Used	23	Objective	1.2432
Missing	1	Objective*N	28.5933

Heteroscedasticity Test

Equation	Test	Statistic	DF	Pr > ChiSq	Variables
post	White's Test	6.80	8	0.5582	Cross of all vars

ESTIMATING PROBIT MODELS IN SAS

As seen in the Becker and Powers' (2001) study, often variables need to be transformed or created within a computer programs to perform the desired analysis. To demonstrate the process and commands in SAS, start with the Becker and Powers data that have been or can be read into SAS as shown earlier.

As always, we should look at our log file and data before we start doing any work. Viewing the log file, the data set BECPOW has 2849 observations and 64 variables. Upon viewing the dataset, we should notice there are several "extra" observations at the end of the data set. These are essentially extra rows that have been left blank but were somehow utilized in the original Excel file (for instance, just pressing enter at last cell will generate a new record with all missing variables). SAS correctly reads these 12 observations as missing values, but because we know these are not real observations, we can just drop these with the command

```
data becpow; set becpow; if a1 = . then delete; run;
```

This works because a1 is not missing for any of the other observations.

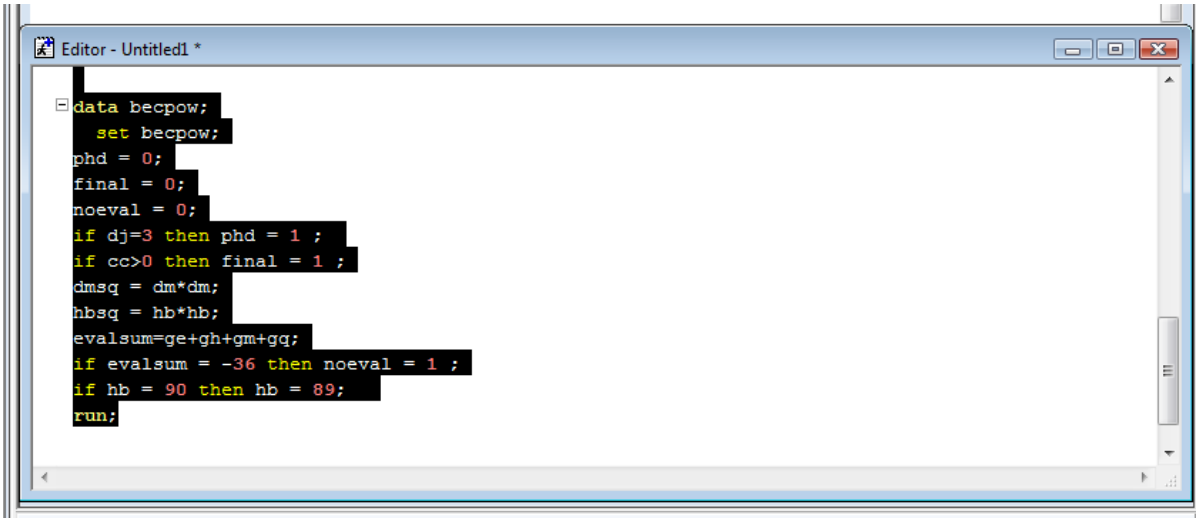
After reading the data into SAS the first task is to recode the qualitative data into appropriate dummies. A2 contains a range of values representing various classes of institutions. SAS does not have a recode command, so we will use a series of if-then/else commands in a data step to do the job. We need to create variables for doctorate institutions (100/199), for comprehensive or master's degree granting institutions (200/299), for liberal arts colleges (300/399) and for two-year colleges (400/499). The following code creates these variables:

```
data becpow;  set becpow; doc = 0; comp = 0; lib = 0; twoyr = 0;

        if  99 < A2 < 200 then doc = 1;

        if 199 < A2 < 300 then comp = 1;

        if 299 < A2 < 400 then lib = 1;

        if 399 < A2 < 500 then twoyr = 1; run;
```

Next 1 - 0 bivariates, "phd" and "final" are created to show whether the instructor had a PhD degree and where the student got a positive score on the postTUCE . To allow for quadratic forms in teacher experiences and class size the variables "dmsq" and "hbsq" are created. In this data set, all missing values were coded −9. Thus, adding together some of the responses to the student evaluations gives information on whether a student actually completed an evaluation. For example, if the sum of "ge", "gh", "gm", and "gq" equals −36, we know that the student did not complete a student evaluation in a meaningful way. A dummy variable "noeval" is created to reflect this fact. Finally, from the TUCE developer it is known that student number 2216 was counted in term 2 but was in term 1 but no postTUCE was taken (see "hb" line in syntax). The following are the commands:

```
data becpow; set becpow;
        phd = 0;
        final = 0;
        noeval = 0;
        if dj=3 then phd = 1;
        if cc>0 then final = 1;
        dmsq = dm*dm;
        hbsq = hb*hb;
        evalsum=ge+gh+gm+gq;
        if evalsum = -36 then noeval = 1;
        if hb = 90 then hb = 89;
run;
```

These commands can be entered into SAS as a block, highlighted and run with the "Submit" button.



One of the things of interest to Becker and Powers was whether class size at the beginning or end of the term influenced whether a student completed the postTUCE. This can be assessed by fitting a probit model to the 1 – 0 discrete dependent variable "final." Because missing values are coded as −9 in this data set, we need to avoid these observations in our analysis. The quickest way to avoid this problem is just to create a new dataset and delete those observations that have −9 included in them. This is done by typing:

```
data becpowp; set becpow;
    if an = -9 then delete;
    if hb = -9 then delete;
    if doc  = -9 then delete;
    if comp = -9 then delete;
    if lib = -9 then delete;
    if ci = -9 then delete;
    if phd  = -9 then delete;
    if noeval = -9 then delete;
    if an = . then delete;
    if cs = 0 then delete;
```

```
run;
```

Finally, we run the probit model by typing:

```
proc logistic data= becpowp descending;
        model final = an hb doc comp lib ci ck phd noeval  / link=probit tech= newton;
        ods output parameterestimates=prbparms;
        output out = outprb xbeta = xb prob = probpr;
run;
```

into the editor window, highlighting it, and pressing enter. The SAS "probit" procedure by default uses a smaller value in the dependent variable as success. Thus, the magnitudes of the coefficients remain the same, but the signs are opposite to those of the STATA, and LIMDEP. The "descending" option forces SAS to use a larger value as success. Alternatively, you may explicitly specify the category of successful "final" using the "event" option. The option "link=probit" tells SAS that instead of running a logistic regression, we would like to do a probit regression. We can then retrieve the marginal effects by typing:

```
proc transpose data=prbparms out=tprb  (rename=(an = tan hb = thb doc = tdoc
        comp=tcomp lib=tlib ci = tci ck = tck phd =tphd noeval = tnoeval));
         var estimate; id variable; run;
data outprb; if _n_=1 then set tprb; set outprb;
        MEffan = pdf('NORMAL',xb)*tan; MEffhb = pdf('NORMAL',xb)*thb;
        MEffdoc = pdf('NORMAL',xb)*tdoc; MEffcomp= pdf('NORMAL',xb)*tcomp;
        MEfflib = pdf('NORMAL',xb)*tlib; MEffci = pdf('NORMAL',xb)*tci;
        MEffck = pdf('NORMAL',xb)*tck; MEffphd = pdf('NORMAL',xb)*tphd;
        MEffeval = pdf('NORMAL',xb)*tnoeval; run;
proc means data=outprb; run;
```

into the editor window, highlighting it and pressing enter. This yields the following coefficient estimates:

The LOGISTIC Procedure

Analysis of Maximum Likelihood Estimates

Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	0.9953	0.2445	16.5697	<.0001
AN	1	0.0220	0.00948	5.4028	0.0201
HB	1	-0.00488	0.00190	6.5742	0.0103
doc	1	0.9757	0.1454	45.0081	<.0001
comp	1	0.4065	0.1389	8.5708	0.0034
lib	1	0.5214	0.1747	8.9088	0.0028
CI	1	0.1987	0.0915	4.7189	0.0298
CK	1	0.0878	0.1385	0.4019	0.5261
phd	1	-0.1335	0.1045	1.6321	0.2014
noeval	1	-1.9305	0.0726	706.2485	<.0001

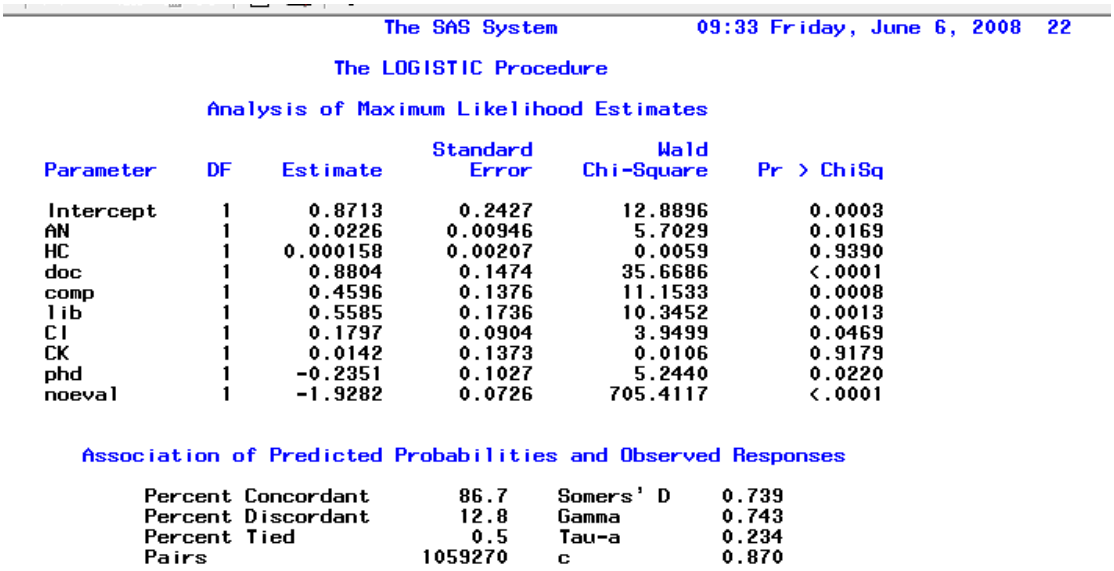
Association of Predicted Probabilities and Observed Responses

Percent Concordant	87.0	Somers' D	0.743
Percent Discordant	12.6	Gamma	0.746
Percent Tied	0.4	Tau-a	0.235
Pairs	1059270	c	0.872

and marginal effects:

Variable	N	Mean	Std Dev	Minimum	Maximum
MEffan	2587	0.0038989	0.0029059	0.000224032	0.0087922
MEffhb	2587	-0.000863776	0.000643782	-0.0019478	-0.000049632
MEffdoc	2587	0.1726140	0.1286512	0.0099184	0.3892486
MEffcomp	2587	0.0719131	0.0535977	0.0041321	0.1621657
MEfflib	2587	0.0922488	0.0687541	0.0053006	0.2080231
MEffci	2587	0.0351577	0.0262034	0.0020202	0.0792813
MEffck	2587	0.0155310	0.0115754	0.000892407	0.0350227
MEffphd	2587	-0.0236184	0.0176031	-0.0532601	-0.0013571
MEffeval	2587	-0.3415295	0.2545458	-0.7701568	-0.0196242

For the other probit model (using hc rather than hb), we get coefficient estimates of:



The SAS System 09:33 Friday, June 6, 2008 22

The LOGISTIC Procedure

Analysis of Maximum Likelihood Estimates

Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	0.8713	0.2427	12.8896	0.0003
AN	1	0.0226	0.00946	5.7029	0.0169
HC	1	0.000158	0.00207	0.0059	0.9390
doc	1	0.8804	0.1474	35.6686	<.0001
comp	1	0.4596	0.1376	11.1533	0.0008
lib	1	0.5585	0.1736	10.3452	0.0013
CI	1	0.1797	0.0904	3.9499	0.0469
CK	1	0.0142	0.1373	0.0106	0.9179
phd	1	-0.2351	0.1027	5.2440	0.0220
noeval	1	-1.9282	0.0726	705.4117	<.0001

Association of Predicted Probabilities and Observed Responses

Percent Concordant	86.7	Somers' D	0.739
Percent Discordant	12.8	Gamma	0.743
Percent Tied	0.5	Tau-a	0.234
Pairs	1059270	c	0.870

and marginal effects of:

The MEANS Procedure

Variable	N	Mean	Std Dev	Minimum	Maximum
MEffan	2587	0.0040104	0.0029923	0.000301942	0.0090143
MEffhc	2587	0.000028147	0.000021002	2.1191781E-6	0.000063267
MEffdoc	2587	0.1562603	0.1165923	0.0117648	0.3512295
MEffcomp	2587	0.0815747	0.0608662	0.0061417	0.1833570
MEfflib	2587	0.0991313	0.0739660	0.0074636	0.2228194
MEffci	2587	0.0318980	0.0238004	0.0024016	0.0716977
MEffck	2587	0.0025126	0.0018748	0.000189174	0.0056477
MEffphd	2587	-0.0417330	0.0311387	-0.0938041	-0.0031421
MEffeval	2587	-0.3422335	0.2553545	-0.7692449	-0.0257666

Results from each model are equivalent to those of LIMDEP and STATA, where we see the initial class size (hb) probit coefficient is −0.004883 with a P-value of 0.0112, and the estimated coefficient of "hc" is 0.0000159 with a P-value of 0.9399. These results imply that initial class size is strongly significant however final class size is insignificant.

The overall goodness of fit can be assessed in several ways. A straight forward way is using the Chi-square statistic found in the "Fit Statistics" output. The Chi-squared is (922.95, df =9) for

the probit employing the initial class size is slightly higher than that for the end-of-term probit (916.5379, df =9) but they are both highly significant.

CONCLUDING REMARKS

The goal of this hands-on component of Module One, Part Four was to enable users to get data into SAS, create variables and run regressions on continuous and discrete variables; it was not to explain all of the statistics produced by computer output. For this an intermediate level econometrics textbook (such as Jeffrey Wooldridge, *introductory Econometrics*) or advanced econometrics textbook such as William Greene, *Econometric Analysis* must be consulted.

REFERENCES

Becker, William E. and John Powers (2001). "Student Performance, Attrition, and Class Size Given Missing Student Data," *Economics of Education Review*, Vol. 20, August: 377-388.

Delwiche, Lora and Susan Slaughter (2003).*The Little SAS Book: A Primer*, Third Edition, SAS Publishing.