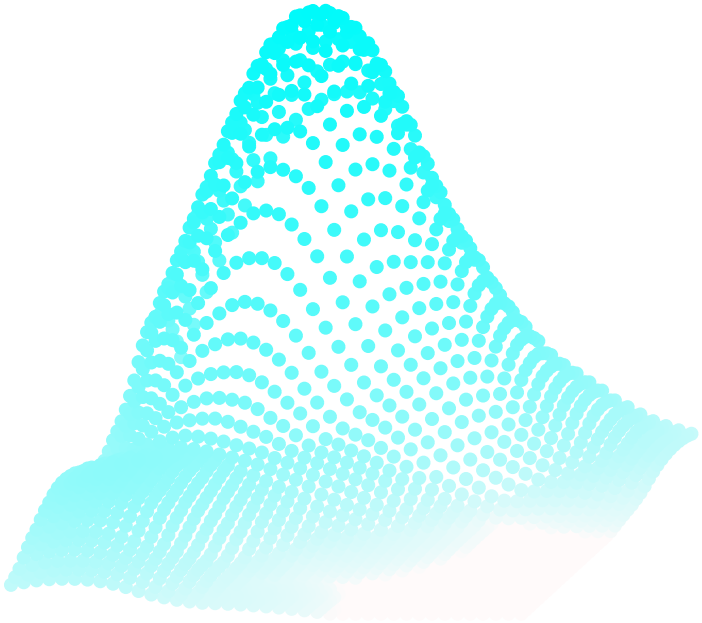

Causal Inference:
Measuring the Effect of X on y

©Austin Nichols
austinnichols@gmail.com

September 28, 2009

List of Figures

Contents

Chapter 1

Basic Ideas and Methods

In designing policy, in the sciences, or in everyday life, we are forever trying to discern the effect of one thing on another. We want to know the effect on college completion and later productivity if we introduce a new tax credit for education. We want to know the effect on survival rates of a new drug, especially if we have the disease the drug is supposed to treat. We want to know what we can say in an interview or write on a resume to get a job offer. And all we have to go on is a collection of observations on what happened in specific circumstances—if we are lucky, a random selection of observations from a broad swath of circumstances. To guess at the true impact, we need a stronger condition: we need the causative factor to be randomly assigned.

This is why we do experiments for new drugs—too often, if a new drug is given to patients in a nonrandom way, it will show positive effects, even if its true impact is zero or negative. If we simply evaluate a treatment by comparing outcomes of those who get the treatment and those who don't, we run the risk of getting entirely the wrong answer. Those who don't get the treatment may be systematically different, for example so sick that they cannot tolerate the treatment or travel to get the treatment, and therefore have worse outcomes for reasons unrelated to treatment.

Even finding individuals who seem similar in many respects to those who were treated, to serve as a control group, is no guarantee of good results. For example, Cameron and Pauling (1976) found that terminal cancer patients who got vitamin C survived four times as long as similar patients who didn't get the vitamin C treatment. But those similar patients were not similar enough—subsequent research (Moertel et al. 1985) using random assignment found no effect.

The random assignment study is the gold standard for inferring a causal rela-

tionship, but what can we do when it is not possible to randomly assign treatment? If the treatment is divorce of parents and the outcome is educational attainment of children, it is highly unlikely you will be allowed to randomly assign treatment and observe outcomes. Or imagine the treatment is race and the outcome is hiring at a firm. In this case, we cannot imagine actually running the random assignment study, but we can imagine a thought experiment where we take existing applicants and reassign race through magic. For measuring certain kinds of discrimination, it is that imaginary magic random reassignment that can give us an estimate of a true causal impact of race, but there are many causal impacts one might measure given that magic power (Rubin 1986). Back in the real world, we need to use data without random assignment, called observational data. This book is about the tricks used to recover an estimate of the true causal impact using observational data.

In mathspeak, we want to explain an outcome (a vector of observations y, one observation y_i for each individual i) as determined by factors we can observe (collected in a matrix X, the columns of which are explanatory variables). So we assume there is some function $y = f(X, e)$ where e is a vector of other factors that determine y (perhaps random noise). It is most convenient to assume a particular family of functions with a parameter to be estimated, e.g. linear functions with parameter b:

$$y = Xb + e$$

We refer to whatever family of functions we specify as our model, and then figure out an estimator to get good estimates of the parameter b. We need not really believe that our model is the truth, just that it is a useful depiction of a more complicated reality, but we would like to believe that we are getting the best estimates we can in whatever family of functions we've limited our attention to. Let's assume for now that we've picked the right family of functions.

The problem is, if X is not randomly assigned, we still cannot say that our estimates of the parameter b are close on average (i.e. the estimates are not *unbiased*) to what we would get in an experiment where X really was randomly assigned. The estimates may not even get closer as we collect more and more data (i.e. the estimates are not *consistent*). We may be narrowing in on the wrong estimate altogether. In an extreme case, we will get the wrong sign, and find a positive effect where the true effect is negative or a negative effect where the true effect is positive. In the rest of this chapter, we will outline techniques used to analyze data when the treatment is randomly assigned, then illustrate how these techniques can fail with observational data, and discuss some general properties we would want from any of the techniques discussed in the later chapters.

The balance of the book focuses on estimating the impact of a treatment on an outcome, such as the effect of education on earnings, where individual units under study have control over their treatment and random assignment is essentially impossible. We try to identify the impact of a specific X on y, or the range of impacts, but we do not look at outcomes and try many potential explanations. In other words, we look for "effects of causes" and not "causes of effects." The exploratory work of identifying possible causes is also important, as is model building or theoretical investigations of likely functional forms, and that kind of work may need to immediately precede the attempts described in this book of estimating the size of an effect of X on y. We will merely assume that a good theory or previous empirical work has already singled out the X variables whose effect on y we wish to estimate.

We also will remain in the frequentist world that most applied researchers operate in, rather than exploring Bayesian or entropic methods that require new derivations for any new kind of estimation problem, though the connections between the methods outlined here and these other approaches is a fruitful and interesting area for further work.

1.1 Basic statistics

There is a lot of interesting methodological subtlety in the basic statistics of the sort you might see on the front page of a newspaper, such as tables of means or proportions, or graphs of these numbers. Most statistics textbooks skip the interesting methodological subtlety, and the book you are holding is no exception, but we will at least touch on a few important points. If you know very little about probability or data (or Stata, the program used throughout), you might want to read the relevant appendices before proceeding.

1.1.1 Tabs

Perhaps the most basic statistical method is the one-way tabulation (tab) or, in graphical form, the histogram. The simplest form of this is the tabulation of an indicator variable X that takes on only values of zero and one (often called a dummy variable or a binary variable or a dichotomous variable, though these terms also have other meanings). The tabulation tells you how many observations in the data have the value one and how many have value zero (first column of numbers in Exhibit 1.1.2).

The tab also indicates the proportions of observations in the data that have the value one and how many have value zero (second column of numbers in Exhibit 1). The first, the proportion of observations in the data that have the value one, is also the sample mean of X. In the tab shown, 60 percent of those surveyed said they "Support the proposal" (whatever that means).

Exhibit 1.1.2 *A simple tab.*

```
. tab support

     Support the |
        Proposal |      Freq.     Percent        Cum.
-----------------+-----------------------------------
0  (No or false) |         40       40.00       40.00
1  (Yes or true) |         60       60.00      100.00
-----------------+-----------------------------------
           Total |        100      100.00
```

With a categorical variable that takes on more than two values, it is often more convenient to create a `histogram` (with a fixed width of bars equal to the minimum distance between two adjacent numerical values) to visualize how the relative weight falls across categories. One might also create categories for a continuous variable and then use `histogram` to visualize the distribution of those categories—though a histogram for a continuous variable can look quite different depending on the bar width. An alternative is to divide the continuous variable into groups at quantiles, for example ten deciles (`help xtile`), in which case each group contains ten percent of the data, but the width of the resulting bars may differ quite a lot (`findit eqprhistogram` for a Stata implementation).

We can do various kinds of inference even with a simple one-way tab, for example testing whether different categories occur with the same frequency, testing[1] whether the observed distribution matches some hypothetical or known distribution, or whether the "true" proportion is equal to some number. For example, if we have a survey of a voting population, of whom 60 percent support a proposal, we might want to know if the population proportion is really 50 percent (i.e. the high proportion of voter approval is just sampling error). This requires a hypothesis test.

[1] Using a "goodness of fit" test (Pearson 1900, Plackett 1983) or alternative methods; `findit mgof` for a Stata implementation.

1.1.3 Tests

For a hypothesis test, we pick α (often 5 percent), which is the proportion of the time we will incorrectly draw the conclusion that the null hypothesis is false (Type I error rate), also called the size of the test. The complement $1 - \alpha$ (often 95 percent) is called the significance level of the test. Then we assume our null hypothesis is true, and we pick a statistic that has a known distribution under the null (or for which an approximate distribution is known). Then we can calculate how likely it would be that we saw the data we actually have (or more extreme data) given that the null is true, and call this probability the p-value. If the p-value is less than α, we reject the null (Lehmann 1993, Berger 2003, Bayarri and Berger 2004).

For a single proportion (or the mean of an indicator variable), we could use the Central Limit Theorem[2] to get an asymptotically valid statistic[3] based on the normal distribution. Suppose we pick $\alpha = 0.05$, and we assume that the true population proportion (or mean) is the hypothesized value $\mu = 0.5$ (50 percent). Now we form the statistic

$$Z = \frac{\overline{X} - \mu}{\sqrt{\mu(1-\mu)/n}} = \frac{.6 - .5}{\sqrt{.5(1-.5)/n}} = .2\sqrt{n}$$

and we can compute the probability that we would see a proportion so far from 0.5 in a sample of size n as $2(1 - \Phi(.2\sqrt{n}))$ where $\Phi(\cdot)$ is the standard normal cumulative distribution function. For a sample size of 100, this would give a p-value of .045 so we reject the null.

However, in this case the asymptotically valid statistic gives us the wrong answer. For a binary variable that takes on the value one (meaning "support the proposal") with probability p, we can figure out that the chance of observing any

[2]The Central Limit Theorem says that a sample mean less its expectation (the population mean) divided by its standard deviation approaches the standard normal (mean zero, standard deviation one) distribution as the sample size gets large; see A.3.4. The standard deviation of a sample mean of n observations on x is the standard deviation of x divided by the square root of n. A variable x that is one with probability p and zero with probability $1 - p$ has standard deviation $\sqrt{p(1-p)}$ as shown in A.3.3.

[3]The term *asymptotically valid* means that the statistic's distribution is essentially known for very large sample sizes (i.e. the deviation from the assumed distribution gets arbitrarily small as the sample size gets larger without bound). So as sample sizes get larger we can be more confident we are not being led astray by assuming we know the distribution—often people assume that 100 is a large sample size, but sometimes a million is not large enough if the approximation is poor.

mix of k ones and $n-k$ zeros is

$$P(k \text{ successes in } n \text{ trials}) = \left(\begin{array}{c} n \\ k \end{array} \right) p^k (1-p)^{n-k}$$

and we can calculate the probability of observing 60 or more ones or 40 or fewer ones. The hypothesis test is easy in Stata: `bitesti 100 60 .5` gives a p-value of 0.057 so we fail to reject the null.

A better approximate test for these situations, that easily generalizes to complex survey data, is the `svy:tab` construction, followed by `test` commands as desired (exhibit 1.1.5). If you don't have complex survey data, you can still use the `svy` commands by specifying `svyset, srs`.

Failure to reject the null is not the same as accepting that the null hypothesis is true; in fact, rejecting the null is not the same as asserting that the null hypothesis is definitely false (since we have picked α to be wrong a fixed proportion of the time). When we fail to reject the null, it may be that we simply do not have enough data, or a design with sufficient statistical power, to distinguish among alternatives. If, in the tab above, the proportion continued to be roughly the same as we collected more data, we would soon reject that the true proportion was one half. We simply cannot say much (given the data we have) about what the true proportion is, but what we can say is often usefully summarized by a confidence interval.

1.1.4 Confidence Intervals

If we conceive a hypothesis test, there is a confidence interval that corresponds to the range of hypothesized values for which we do not reject the null. For the asymptotically valid statistic for a single proportion

$$Z = \frac{\overline{X} - \mu}{\sqrt{\mu(1-\mu)/n}}$$

the values of μ for which we would fail to reject the null produce Z in the range (-1.96,1.96), since values in that range give p-values greater than 0.05 (confidence less than 95 percent). The endpoints are the values that give p-values of exactly 0.05 or confidence of 95 percent, so we call it a 95 percent confidence interval (for any other choice of α we would construct a $1-\alpha$ confidence interval instead).

We can solve for the endpoints of that range:

$$\left| \frac{\overline{X} - \mu}{\sqrt{\mu(1-\mu)/n}} \right| = 1.96 \quad \Rightarrow \quad \mu = \overline{X} \pm 1.96\sqrt{\mu(1-\mu)/n}$$

so for 60 successes in 100 trials we can assert that the true proportion lies in the range (0.502,0.698) with 95 percent confidence. In this case, however, we have already decided that the asymptotically valid statistic is not good enough for our sample of 100.

Using the binomial probability distribution to create a confidence interval is a little trickier, since there is no guarantee that any hypothesized μ will produce a p-value of exactly 5 percent (Brown et al. 2001, Agresti and Coull 1998), but `svy:tab` with the `ci` option is a good approximation. In the example in Exhibit 1.1.5, the confidence interval includes one half, but the `test` output indicates that we would reject the hypothesis that the true proportion is one half.

Exhibit 1.1.5 *Using svy:tab.*

```
clear
set obs 100
g support=_n>40
la def s 0 "0  (No or false)" 1 "1  (Yes or true)"
la val support s
la var support "Support the Proposal"
svyset, srs
svy:tab support, ci
test _b[p2]=.5
```

1.1.6 Crosstabs

The crosstab is the simplest and most intuitive way to compare two distributions. For example, we might compare outcomes for two medical procedures in a simple two by two table. Suppose you have kidney stones and you can pick from two equal-cost treatment options, P or O[4] Published data on outcomes by treatment (Charig et al. 1986) indicates a success rate for P of 83 percent and a success rate for O of 78 percent, shown as row proportions in the tab of treatment versus success in Exhibit 1.1.7.

[4]In fact, O (open surgery) is a substantially higher-cost option than P (percutaneous nephrolithotomy), both in terms of pecuniary cost and morbidity associated with treatment. For the purposes of the illustration, we will pretend they cost the same. There are additional treatment options, including extracorporeal shockwave lithotripsy (ESWL) that have better success rates and lower costs in many cases, which we will ignore for this example.

Exhibit 1.1.7 *A simple crosstab.*

```
. use kidneyst, clear
. ta T S, row nokey

           |        Success
 Treatment |         0          1 |     Total
-----------+----------------------+----------
         P |        61        289 |       350
           |     17.43      82.57 |    100.00
-----------+----------------------+----------
         O |        77        273 |       350
           |     22.00      78.00 |    100.00
-----------+----------------------+----------
     Total |       138        562 |       700
           |     19.71      80.29 |    100.00
```

We could calculate column proportions as well, but those proportions make little sense in this context. They answer the question, what is the chance that a successful procedure was of type P or of type O? This is not what we care about if we are suffering from kidney stones. We want to know which procedure has a higher success rate, which is given by comparing row proportions, and P seems the clear winner at 83 percent. The comparison of success rates naturally invites a causal conclusion—who would choose treatment O given this information?

If we wanted to determine whether such a difference in success rates could have arisen merely from chance, we could use `svy:` `tab` with the `row` and `se` options, then `test` the hypothesis that success rates are identical, and find that the apparent advantage of treatment P is not "statistically significant" (i.e. fail to reject the null that success rates are identical). For the purposes of this illustration, we will pretend we have 10 times the data shown above and the F statistic is 23 instead of 2.3 (by artificially manufacturing 7000 observations from 700), which changes the p-value from ten percent to one and a half millionths. This is merely to abstract from worries about statistical significance and focus on the ranking of treatments.

What happens to our ranking of treatments when we compare the effectiveness of the two treatments P and O for kidney stones in two size categories large (at least 2cm) and not large? For both larger and smaller stones, treatment O dominates, with success rates of 73 and 93 percent compared to 69 and 87 percent (Exhibit 1.1.8). The ranking of treatments according to success rate reverses when we make the comparison in each of two categories.

Exhibit 1.1.8 *Success rate reversal.*

```
. use kidneyst, clear
. qui expand 10
. qui svyset, srs
. qui svy: tab T S, row se
. qui test _b[p12]=_b[p22]
. di "P=" _b[p12] ", O=" _b[p22] ", p-value is " r(p)
P=.82571429, O=.78, p-value is 1.507e-06
. qui svy, subpop(Large): tab T S, row se
. di "P=" _b[p12] ", O=" _b[p22] ", p-value is " r(p)
P=.6875, O=.73003802, p-value is .02176078
. qui svy, subpop(if Large==0): tab T S, row se
. qui test _b[p12]=_b[p22]
. di "P=" _b[p12] ", O=" _b[p22] ", p-value is " r(p)
P=.86666667, O=.93103448, p-value is 2.636e-09
```

The problem, as in all observational studies, is that the treatment is not randomly assigned, so we cannot conclude that the measured impact (or measured difference) is really related to the causal impact. Here, treatment O is much more likely to be applied to the difficult cases of large stones, with an inherently lower success rate. So in the aggregate, treatment O looks worse, but conditioning on the stone size makes it clear that treatment O is more effective.

1.1.9 Comparisons of Means

The mean of a random variable x is denoted $E(\mathrm{x})$, and the mean of a sample x of n observations on x is given by

$$\overline{x} = n^{-1} \sum_{i=1}^{n} x_i$$

though the line over a quantity is not a very congenial notation when we want to sample mean of something more complicated than a single variable, so we can also write

$$E_n(x) = n^{-1} \sum_{i=1}^{n} x_i$$

Often, we want to test whether the sample mean is "really" some number, say whether mean income in some population is really 50 thousand when we observe a sample mean of 60 thousand, and whether the difference could just be sampling error. If we knew the standard deviation of x, call it σ, we could form the statistic

$$Z = \frac{\overline{X} - \mu}{\sigma / \sqrt{n}}$$

as we did with an indicator variable in section 1.1.3 (where we knew the standard deviation of x because if you know the mean of an x that takes on two values, you know its standard deviation as well).

Unfortunately we don't know the standard deviation of x. We can get a good estimate using the sample variance s^2:

$$s^2 = \frac{\sum \left(X - \overline{X}\right)^2}{n-1} \quad \Rightarrow \quad s = \sqrt{\frac{\sum \left(X - \overline{X}\right)^2}{n-1}}$$

and plug it in for σ, but then we have a different ratio, with random variables in the numerator and denominator. Gosset, publishing under the name Student (Gosset 1908), figured out the limiting distribution of that ratio, so we have a good approximation of the distribution of our statistic in the t distribution. The Stata implementation is `ttest`.

For the comparison of two means, for example the mean wage of those who have a college degree (c for college) and those who don't (h for high school), we can hypothesize no difference ($\mu_c - \mu_h = 0$) or some other value b for the difference ($\mu_c - \mu_h = b$). If we knew the variance of wage for each group, the relevant statistic would be

$$Z = \frac{\overline{x}_c - \overline{x}_h - (\mu_c - \mu_h)}{\sqrt{\frac{\sigma_c^2}{n_c} + \frac{\sigma_h^2}{n_h}}}$$

(asymptotically a standard normal) but we don't know those variances, so we use the sample variances to compute

$$t = \frac{\overline{x}_c - \overline{x}_h - (\mu_c - \mu_h)}{\sqrt{\frac{s_c^2}{n_c} + \frac{s_h^2}{n_h}}}$$

which is distributed t as well.

If we assume that both groups have identical variance, we can use the sample variance pooling the two groups:

$$t = \frac{\overline{x}_c - \overline{x}_h - (\mu_c - \mu_h)}{\sqrt{s_{pooled}^2 \left(\frac{1}{n_c} + \frac{1}{n_h}\right)}}$$

which is distributed $t(n-2) = t(n_c + n_h - 2)$.

Exhibit 1.1.10 *Mean earnings by education.*

```
. use http://pped.org/card, clear
. mean wage if educ>9, over(educ) nol noh
------------------------------------------------------------
        Over |       Mean   Std. Err.     [95% Conf. Interval]
-------------+----------------------------------------------
wage         |
          10 |    420.952   15.57002      390.4221    451.4819
          11 |   476.6352   15.40384      446.4312    506.8393
          12 |   563.5343   7.537055      548.7555     578.313
          13 |   562.8007   12.22387       538.832    586.7694
          14 |   596.1711   15.99349      564.8109    627.5313
          15 |    571.025   21.34021      529.1808    612.8692
          16 |   642.8932    12.5294      618.3254    667.4611
          17 |    654.106   24.36092      606.3388    701.8732
          18 |   776.2899   24.67559      727.9056    824.6741
------------------------------------------------------------
. test [wage]10==[wage]11
 ( 1)  [wage]10 - [wage]11 = 0
       F(  1,  2796) =    6.46
            Prob > F =    0.0111
```

When we draw any conclusions from this type of comparison, as seems inevitable, we are firmly back in the realm of incorrect casual inference, possibly without noticing. Since college is not randomly assigned, the true causal impact could be larger or smaller than the difference observed in our sample, or even the difference observed in the population. We will return to this example in section 1.5.2.

1.2 Linear Regression

We return to our model from the beginning by assuming once more that an outcome y is determined by a matrix of variables X, and the true model is one of a particular family of functions with a parameter to be estimated: linear functions with parameter b:

$$y = Xb + e = x_1 b_1 + x_2 b_2 + \ldots + x_k b_k + b_0 + e$$

Note that X has $k + 1$ columns, the last of which is a vector of ones, called the constant term. Spelling this out without the matrix notation takes a lot of room,

which is why matrix notation is so very popular:

$$\begin{pmatrix} y_1 \\ \vdots \\ y_i \\ \vdots \\ y_n \end{pmatrix} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1k} & 1 \\ \vdots & \vdots & & \vdots & 1 \\ x_{i1} & x_{i2} & \ddots & x_{ik} & 1 \\ \vdots & \vdots & & \vdots & 1 \\ x_{n1} & x_{n2} & \dots & x_{nk} & 1 \end{bmatrix} \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_k \\ b_0 \end{pmatrix} + \begin{pmatrix} e_1 \\ \vdots \\ e_i \\ \vdots \\ e_n \end{pmatrix}$$

This model is usually estimated via Ordinary Least Squares (OLS) and is the workhorse of all empirical research. Estimation techniques such as maximum likelihood and generalized method of moments give the same answers as OLS for this model, but OLS has a simple interpretation as minimizing the distance of outcomes y (in the y dimension) from the linear combination of the variables in X we are using to predict y. Those distances are the squared residuals $(\widehat{e})^2$ given by:

$$\widehat{e}^2 = (y - X\widehat{b})^2$$

and we minimize the sum of squared residuals when:

$$X'X\widehat{b} = X'y$$

If $X'X$ is full rank, then we can solve for the unique solution

$$\widehat{b} = (X'X)^{-1}X'y$$

, or rather, we have Stata do it for us by typing `regress y x`. Assuming the linear model we started with is a good one, the error in our estimate of the effect of X on y is linearly related to the correlation between X and the error term e

$$(\widehat{b} - b) = (X'X)^{-1}X'e$$

so if $X'e = 0$ the OLS estimate is exactly right, and if $E[X'e] = 0$ then the OLS estimator is right on average, i.e. it is unbiased. In fact, in that case, the OLS solution is the best linear unbiased estimator (BLUE) of b by the Gauss-Markov Theorem assuming elements of e have mean zero, constant variance, and zero covariance.[5] "Best" in the BLUE acronym means "has the lowest mean squared error" (more on "unbiased" and "mean squared error" in section 1.4).

[5] see (Aitken 1935) for an extension to the case with a more complicated covariance structure, and see (Plackett 1950) for some proofs.

The predicted value, or fitted value, $\widehat{y_i} = X_i\widehat{b}$ is the best estimate of expected y for observation i given the observed value of X, since the mean of y for observation i is X_ib and our best estimate of that is $X_i\widehat{b}$. Thus the predicted value is the conditional mean of the outcome, meaning conditional on X observed. Most forms of estimation in the rest of the book are conditional means, and the treatment of conditional means is not too different from the treatment of means we already have seen, and subject to all the same pitfalls of causal inference. Questions of interpretation in these models that are essentially estimating conditional means often come down to "the mean conditional on what?"

1.2.1 Example with Two Explanatory Variables

The linear regression model fits a plane (or hyperplane, really, which is the higher-dimensional version of a plane) through the data in such a way that the distances of data points from the surface in the y direction are minimized. With two variables x_1 and x_2 included on the right hand side, the interpretation of an estimated coefficient on x_1 is the expected change in y with a one-unit change in x_1, holding x_2 constant. The estimated coefficient on x_2 is the predicted change in y with a one-unit change in x_2, holding x_1 constant.

As a simple example, suppose we look at data on education and experience as x_1 and x_2 and the variable lwage as y. Figure 1.1 shows the distribution of education and experience in the data, and the figure 1.2 rotates that graph into a two-dimensional representation of the three-dimensional space that also includes lwage. It also adds the values of lwage, shown as bars rising from the points in the education and experience plane, and shows a plane fit through the data points. The slopes of that plane where it cuts the $x_1 = 0$ and $x_2 = 0$ planes on the left and right show the coefficients on x_1 and x_2 (education and experience), also called the marginal effects or partial effects of x_1 and x_2.

Note that when an economist says "marginal effect" or "marginal benefit" or the like, the concept is related to the slope of a curve or a derivative, and the idea is what is the increase in the expected outcome given a small change in an explanatory variable. Do not interpret "marginal" to mean "minimal" or somesuch, used by many other authors e.g. in this quote from the New York Times[6] in July 2009:

> Many workers who have lost their jobs are older and had spent their lives working in one industry. In need of a job right away, many

[6] http://www.nytimes.com/2009/07/06/us/06retrain.html

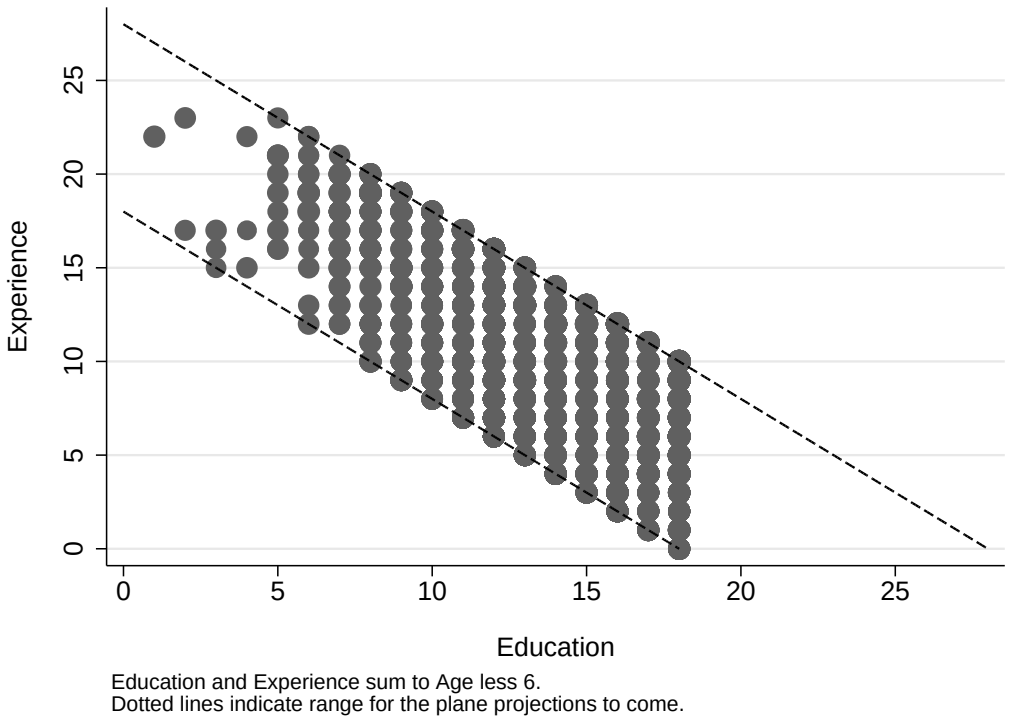

Figure 1.1: The two predictors education and experience.

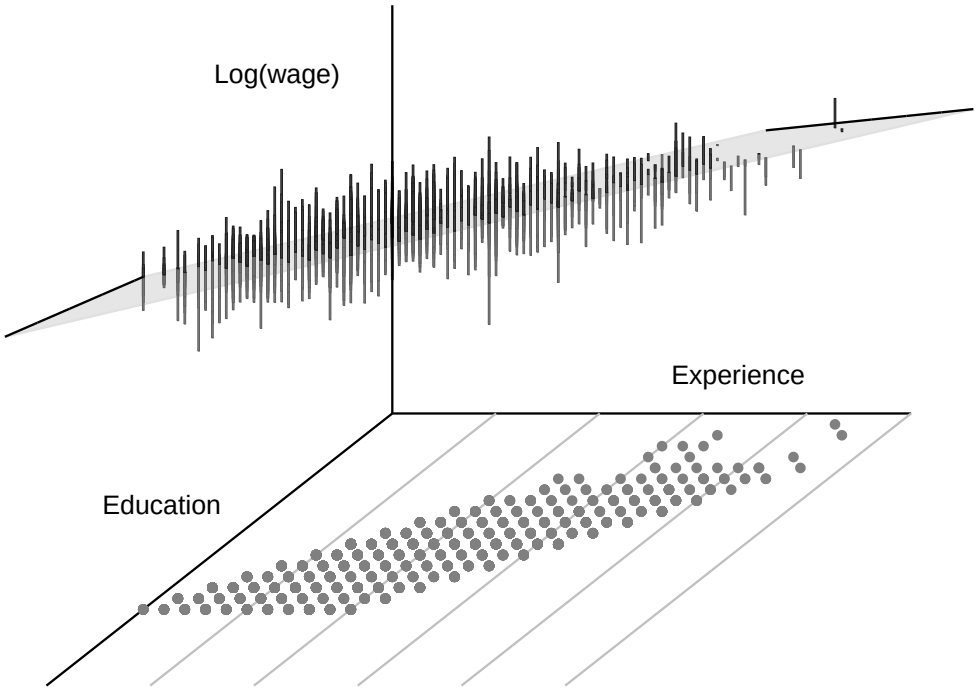

Figure 1.2: Partial effects in linear regression.

pick relatively short training programs, which often have marginal benefits.

Marginal is also use to mean "liminal" or right on the edge, for example the "marginal participant" in a job training program is one who is right on the edge of participating; that is the participant who would not be participating if the cost of participating were increased by a very small amount.

1.2.2 Examples with Dummies and Interactions

Suppose we want to know the effect of education on future wage rate and we are ignorant of the voluminous economic literature on the subject. We might regress wage on an indicator variable for college education, if that was the treatment whose effect we wanted to measure. That regression is equivalent to the comparison of means across college and not-college categories. Comparisons of means across groups can also be done by regressing the outcome on an exhaustive set of mutually exclusive indicators, called the "fully saturated" regression, and testing the equality of coefficients on indicator variables. Exhibit 1.2.3 shows an example using regression with a test that equivalent to the example in section 1.1.9.

Exhibit 1.2.3 *Mean earnings by education.*

```
. use http://pped.org/card, clear
. mean wage if educ>9, over(educ) nol
. test [wage]10==[wage]11
. qui reg wage educ if inlist(educ,10,11), r
. test educ
. qui tab educ, gen(d)
. qui reg wage d11 if inlist(educ,10,11), r
. test d11
```

This illustrates one estimate of the effect of an additional year of education on wage, just comparing those who finish 11th grade to those who finish 10th grade. If we further hypothesize that the same mean difference obtains between those who finish 11th grade and who finish 12th grade, or those who finish 12th grade and those who complete a year of postsecondary education (13th year), or any other pair of years, we can get a more reliable estimate by including education as a linear predictor (Exhibit 1.2.4).

Exhibit 1.2.4 *Mean wage by education.*

```
. use http://pped.org/card, clear
. reg wage educ, r
```

We can see from the table of mean wages that the wage of those with some college (13, 14, or 15 years of education) is more comparable to those with a high school (12 years) and there is a discrete step up at 16 years, so it might make more sense to compare those with 16 or 17 years to those with 12 or 13 years of education, or to use an indicator (dummy) variable for every level of education.

Suppose we think that education does have a linear effect (i.e. every year of education, from 10th grade to 11th and 11th to 12th grade, and so on, has the same effect on mean future wage) but that it is different in the south. We can regress wage on education, a dummy for south, and the interaction term (the product of the two variables). With all three terms on the right-hand-side, we are essentially running two separate regressions, one for the south and one for the not-south. Each has an intercept (mean wage at education equal to zero) and a slope (effect of an additional year of education) term—for the south, the intercept is the sum of the coefficient on the Constant term and the coefficient on south, and the slope is the sum of the coefficient on educ and the coefficient on the interaction sXe. It's helpful to see the estimated coefficients presented as predictions on a graph (Exhibit 1.2.5).

Exhibit 1.2.5 *Mean wage by education.*

```
. g sXe=south*educ
. reg wage educ south sXe, r
. loc a "tw lfit wage educ if south==0, ra(0 18) ||"
. `a'lfit wage educ if south, ra(0 18) leg(lab(1 "N") lab(2 "S"))
. di _b[sXe]*18+_b[south]
```

It's easier to interpret the negative estimated coefficient on the south dummy combined with the positive estimated coefficient on the interaction term on the graph, as opposed to the regression, since we can see that education produces larger apparent returns in the south, but even at the top education level those in the south earn less. To see that in the regression, one has to add the interaction's coefficient times 18 (the maximum education in the sample) to the coefficient on south, to see the difference is still negative at the maximum education level.

1.2.6 Standard Errors

Given our OLS estimates, to do hypothesis testing, we need to form a statistic of the same type we did for means:

$$t = \frac{\widehat{b} - c}{SE(\widehat{b})}$$

where c is some hypothesized value and the standard error $SE(\widehat{b})$ is our estimate of the standard deviation of the estimator. The estimate of the variance-covariance matrix of the estimator is called the VCE, and the square root of the jth diagonal element is the standard error $SE(\widehat{b_j})$, which is our estimate of the standard deviation of a particular coefficient.

Assuming the errors are independently and identically distributed ("identically distributed" means errors are homoskedastic), we can use as our estimate of the VCE

$$\widehat{V} = \widehat{Var}(\widehat{b}) = s^2(X'X)^{-1}$$

where s^2 is an estimate of the variance of residuals, called "the mean squared error of the regression," which is the mean squared residual scaled by a degrees-of-freedom adjustment. In other words, s^2 is the estimate of $e'e$ given by $(n-k)^{-1}\sum_i \widehat{e_i}^2$ where k is the number of columns in X (i.e. the number of explanatory variables plus one, for the constant). This is the standard error reported by Stata after `regress` with no options requesting an alternative calculation.

In fact, we should always use an estimate of the VCE robust to heteroskedasticity, since if errors are homoskedastic the cost is minimal and the standard errors will be nearly identical with or without a correction for heteroskedasticity. In Stata, just put `robust` or `vce(robust)` after the comma in the `regress` command to get a VCE robust to heteroskedasticity.

Often, we cannot reasonably assume that errors are uncorrelated across observations. In a survey, sample units are chosen that may exhibit correlated errors relative to the regression model (e.g. pupils in the same school, or people in a household). In population data, a variable measured at a level higher than the observation level, e.g. mean county transportation times in a regression at the individual person level, may induce correlation of individual errors at the higher level (county in the example).[7] In the cases described, however, we can often still reasonably assume that errors are uncorrelated across clusters of observations

[7] If a variable X has an effect on outcome y at the individual level, and we measure not X but mean of X within group, we can induce clustering where none existed in the individual-level

(sampling units in the survey, or counties in the population data). In that case, we can use the cluster-robust VCE by specifying `cluster(v)` or `vce(cluster v)` after the comma following `regress`, where `v` is the variable that defines cluster membership. For the estimated VCE to work well, the number of clusters less the number of coefficients involved in a test must be large (at least 20, but preferably more than 50) and the clusters must be of roughly equal size (Nichols and Schaffer 2007, Rogers 1993). For survey data, there is often stratification to be taken account of as well, so we must turn to `svy` commands in Stata (see also section 1.5.4).

1.2.7 Nonlinear Effects in the Linear Model

Incorporating nonlinear effects is quite easy in the linear regression framework; simply include the square of a variable, or any other function of a variable in X. Of course, this will produce unbiased and consistent estimates only if the assumptions of OLS still hold ($E[X'e] = 0$ for unbiasedness or $\text{plim}[X'e] = 0$ for consistency).

Marginal Effects and Polynomials

If we have a single explanatory variable and its square in our preferred model, we can write

$$y = xb_1 + x^2b_2 + b_0 + e$$

and calculate OLS estimates in the usual way.

However, the effect of x on $E[y|x]$ in such a model now depends on the level of x. So we speak of the marginal effect or partial effect[8] of x

$$\left.\frac{\partial E[y|x]}{\partial x}\right|_{x=x_0} = b_1 + 2x_0b_2$$

which is an estimate of the effect of a one-unit change in x near x_0 (and will depend on the units in which x is measured). Our estimate of that quantity is

model. The importance of correcting standard errors for clustering is increasing in both the correlation of errors and the correlation of explanatory variables, so the case of using variables measured at a higher level than the level of analysis is an important special case where the explanatory variable does not vary within cluster (fixed effects is another important special case).

[8] Anyone who knows calculus recognizes this as the derivative; those who are unfamiliar with calculus can simply think of this as the slope at the point $x = x_0$, for example the slope at $x = 2$ shown on as the tangent line in figure A.2.

just $\widehat{b}_1 + 2x_0\widehat{b}_2$. If we have regressed y on x and its square, we can compute this quantity at different values x_0 and do hypothesis testing on it—the command in Stata to do hypothesis testing on this kind of linear combination of coefficients is `lincom`. We can also use the factor variables and the `margins` command, both new in Stata 11, to get estimates of this type of model and to characterize marginal effects.

Exhibit 1.2.8 *Quadratic terms and marginal effects.*

```
. use http://pped.org/card, clear
. reg wage educ c.exp##c.exp, r
. margins, post
```

Additional polynomial terms are added likewise. If we have a single explanatory variable and its square and cube in our preferred model, we can write

$$y = xb_1 + x^2b_2 + x^3b_3 + b_0 + e$$

and calculate OLS estimates in the usual way, but the effect of x will again depend on the level of x. The marginal effect of x is

$$\left.\frac{\partial E[y|x]}{\partial x}\right|_{x=x_0} = b_1 + 2x_0b_2 + 3x_0^2b_3$$

and our estimate of that quantity is $\widehat{b}_1 + 2x_0\widehat{b}_2 + 3x_0^2\widehat{b}_3$, and our hypothesis testing still involves a linear combination of coefficients (if we wanted a test involving the ratio of two coefficients, or the product, or some other nonlinear function we would use the Stata command `nlcom`).

When interpreting the output of a regression with polynomial terms in a variable x, there are a few considerations to always keep foremost in your mind. First, just because the sign of a higher-order term has a certain sign, you do not know that sign will dominate at any observed value. For example, if a linear term has a positive estimated sign and a squared term a negative sign, it is not the case that the outcome will "turn down" as a second-order polynomial in x does, and it is not always the case that the outcome is initially increasing in x. Those kinds of conclusions depend on the distribution of x in the data. If the linear term is estimated to be 1 and the squared term negative one half, for example, the slope of the response function will be $1 - x$ from the formula shown above, so the outcome y will be increasing in x for x less than one, and decreasing for larger x. If the whole distribution of x lies between negative 30 and negative 29, the nonlinearity

will be minimal and y will be increasing in x everywhere; if the whole distribution lies between zero and one, y will still be increasing in x everywhere but the nonlinearity will be very noticeable. If x is always greater than one, the effect of x on y is negative, and increasingly so. A graph of marginal effects is the best way to summarize nonlinear effects, as we reiterate in 1.3.2.

The default summary of marginal effects is usually the "average partial effect" (APE) or "average treatment effect" (ATE) which is computed as the mean over all individuals of the marginal effect computed for each individual. It is easy to see how this kind of summary could be quite useless in the context of a quadratic in x if we imagine that the marginal effect of x is $1 - x$ and the distribution of x is symmetric around 1, for example if x is uniformly distributed on the interval [0,2], in which case the average marginal effect is zero but the marginal effect is negative for half the sample and positive for half the sample in a very predictable way. This is a very different story than one where the marginal effect is zero for everyone, or may be positive or negative for everyone in an unpredictable way, so that the average marginal effect is zero for everyone. The information about marginal effects that is implicit in an estimated model is almost always clearer in a graph, but showing that information may require a deep understanding of the marginal effects and some art with Stata graphics.

Marginal Effects and Logarithms

A common specification to estimate a nonlinear relationship in a linear model specifies that the log of the outcome is linear in X (so $\ln(y) = Xb + \varepsilon$), also called a log-linear model, or the outcome is linear in the log of some x (e.g. $y = Xb + \ln(x)c + \varepsilon$) or a log-log specification ($\ln(y) = Xb + \ln(x)c + \varepsilon$). In the log-log specification, the coefficient is interpreted as an elasticity (a one percentage point change in x produces approximately a c percentage point change in y).

In the log-linear specification, the effect of a variable x on y with the reported coefficient $\widehat{b}$ is a proportional change of $\exp(\widehat{b}) - 1$ in y for a one-unit change in x, or a marginal effect of $100\widehat{b}$ percent change in y with a change in x. When $\widehat{b}$ is small, a change of $\exp(\widehat{b}) - 1$ is equivalent to a percent change of $\widehat{b}$, and the approximation gets worse as $\widehat{b}$ gets larger.

The term "marginal effect" can be used to mean several different things, which adds to the confusion. Sometimes the discrete change in y with a small discrete change in x is meant, in which case one might call it a "discrete marginal effect" for clarity. The usual meaning is the slope at a single point, which is the slope of a tangent to the curve at that point. If one is discussing marginal effects in a log-log

model, $100\widehat{b}$ is not an approximation; it is the marginal effect. But the marginal effect is an approximation to any discrete change.

The marginal effect of a one-unit change in x on y in a log-linear model is $100\widehat{b}$ percent using calculus, but the calculus answer will be inadequate for approximating the effect of discrete changes in x when $\widehat{b}$ is large in absolute value. This in turn is often caused by adopting units for x that mean that a one-unit change in X cannot be considered small[9] and $100\widehat{b}$ is less similar to $100(\exp(\widehat{b})-1)$. It is easy to be led astray in these kinds of nonlinear calculations—ask yourself what the marginal effect of an additional week worked is in the models shown in Exhibit 1.2.9.

Exhibit 1.2.9 *Marginal effect with logged dependent variable.*

```
. webuse psidextract, clear
. reg lwage wks, r
. loc w=_b[wks]
. g pwks=wks/52
. reg lwage pwks, r
. loc p=_b[pwks]
. di `w', `p'/52, exp(`w')-1, (exp(`p')-1)/52
```

The last number shown is the "discrete marginal" effect of a 52-week increase, using the $100(\exp(\widehat{b})-1)$ formula, divided by 52, to give the incorrect answer—the correct answer of course divides by 52 first. The $100(\exp(\widehat{b})-1)$ formula is for a one-unit discrete change in X, which is at least as valuable for interpreting coefficients, and often more valuable, but is not what is usually meant by a marginal effect (see the manual entry for `margins` for more discussion), though at least some authors conflate the two.

Whenever the coefficient is large, the interpretation of the marginal effect is complicated by the fact that the calculus answer is probably a poor approximation to plausible discrete changes in X, and even a coefficient of 0.065 can be large for this purpose.

Exhibit 1.2.10 *Marginal effect with logged dependent variable.*

```
. reg lwage ed, nohe
. loc r = _b[ed]
. di `r'*4, (1+`r')^4-1, (exp(`r')-1)*4, (exp(`r'*4)-1)
```

[9]Sometimes we economists say "marginal" to mean "small enough" for calculus to give a reasonable approximation to a discrete change. For a marginal change, we say that only "first order" effects matter, or all other effects are "second order" which refers to the notion of a Taylor series expansion; see A.2.2.

In these cases, of course, one is well-advised to calculate the effect of a discrete change rather than rely on the marginal effect. I.e. the last answer above (.2979877), given by thoughtful application of the $100(\exp(\widehat{b})-1)$ formula, is the best answer, and the naive approach gives an answer seven eighths as big.

One issue is that we want to have the standard error as well, and $\widehat{b}$ has it given in the regression table (100*se), together with the confidence interval and p-value on a test of $\widehat{b} = 0$, whereas $100(\exp(\widehat{b}) - 1)$ needs `nlcom` to calculate those, which makes tabulating estimates look like more work, but note how easy it is to get the discrete marginal effects tabulated using the user-written `estout` on SSC (`esttab` is part of the `estout` package).

Exhibit 1.2.11 *Marginal effect in estout.*

```
. reg lwage ed, nohe
. nlcom exp(_b[ed])-1
. esttab, b(%10.7f) se(%10.7f) nostar transform(exp(@)-1 exp(@))
```

Marginal Effects and Logical Relationships

Even if we are using the new `margins` command in Stata 11 with factor variables to get good calculations of average marginal effects, or we are using another automated solution to get marginal effects averaged over the appropriate population or subpopulation, we cannot suspend our critical faculties and take at face value the results as given. It is always advisable to understand how you would construct the average marginal effect by hand, even if the calculations would be too tedious to actually do by hand, so that you have a good sense of whether they make sense. One common pitfall in the area of calculation of marginal effects is measuring the partial effect of one variable that is logically related to another. This is easy to see in the case of a linear regression of y on x and its square (you cannot change one without changing the other) but more difficult cases frequently trip up even experienced researchers.

One tricky case involves indicator variables that are logically related. Suppose we have an indicator variable `c` for "college" (zero for those without a college degree and one for those with a degree) and `sc` for "selective college" (one for those with a degree from a set of schools identified as selective according to some standard, and zero otherwise). Now we can interpret the coefficient on `c` as the return to college (i.e. the difference in predicted y for `c` equal to one relative to `c` equal to zero) and the sum of coefficients on `c` and `sc` as the return to a selective college, or the coefficient on `sc` as the additional return to a selective college relative to a non-selective college.

If the indicators are also interacted with continuous variables, or y is subjected to a nonlinear transformation (e.g. log), or we are using a nonlinear model, we cannot simply look at the coefficients, and we need to actually compare predictions at different levels. The standard approach is to use the empirical distribution of every variable but the variable of interest, and set the variable of interest to two values in order to make two predictions, then average the difference in predictions across the sample. But note for `c` and `sc` above, if we leave `c` as it appears in the data and set `sc` to zero and one in turn, we are imposing logical contradictions on our counterfactuals–we are making predictions for observations that went to a selective college (where `sc`=1) but did not go to college (i.e. `c` equals zero). Likewise if we leave `sc` as it appears in the data and set `c` to zero and one in turn.

We must somehow take account of the dependency in the variables when we predict. The natural way to get a marginal effect for `sc` is to leave `c` as it appears in the data and set `sc` to zero and one in turn for only the observations where `c` equals one. The natural way to get a marginal effect for `c` is to turn off `sc` when we turn off `c`, but another possibility is to make predictions only for observations where `sc` equals zero, which will give a different answer. The answer depends on the question—do we want to imagine denying college to all those who did not go to selective colleges, and then giving it to all of them, or do we want to imagine denying college to everyone, and then giving everyone college (and a select few a selective college)? We will return to similar questions in 1.3.2.

1.2.12 Model Diagnostics

There are many popular techniques for assessing the quality of a regression. One important class of techniques uses graphs, for example a scatterplot of residuals $\widehat{e}$ versus fitted values $\widehat{y}$. If any structure is discernable in that graph (for example, curvature or cyclicality in the mean residual) then the model is probably misspecified. This is implemented in Stata by typing `rvfplot` after `regress`. Another important class of techniques uses hypothesis tests; for example, a RESET (Ramsey 1969) test testing whether there are any neglected nonlinearities, e.g. whether the square (or another nonlinear transformation) of any of the variables x in X should be included as an additional regressor. This is implemented in Stata by typing `estat ovtest` after `regress`. The Stata manual entry **[R] regress postestimation** covers some other useful techniques, including added-variable plots (`avplots`) and the augmented component-plus-residual plot (`acprplot`).

One popular technique for assessing the quality of a regression is to look at the variance of the fitted values $\widehat{y}$ divided by the variance of the dependent variable

y, called R^2 (R squared). This is sometimes said to be the proportion of variance "explained" but is an extremely unreliable measure of the quality of a regression.

1.2.13 Local Polynomial Regression

There is a very large class of semiparametric models, where one makes fewer assumptions about the model that generated the data, and aims to get estimates more robust to various potential specification errors. Often, these are "local" models, where we assume a model holds only for a subset of the data in some neighborhood, and thus get estimates in neighborhoods of a variety of points. One useful member of this class is local polynomial regression, which is used later in the chapter on Regression Discontinuity designs. In fact, we can limit our attention to polynomials of degree one (linear regressions) since it almost always better to use an odd degree for these models and curvature in the conditional mean function can be observed without using a higher degree polynomial. The manual entry on `lpoly` gives a good description of these models, and further references, but you can just think of `lpoly` as estimating 50 (or 100, or however many you like) linear regressions on 50 subsets of the data and connecting the dots (predictions of the linear regressions at prespecified X values).

Assuming we have already chosen `degree(1)` option for linear regression, there are two other choices that define how smooth our estimates are, and offer a tradeoff between bias and variance. The kernel and bandwidth options of `lpoly` determine how many observations are used in each of the 50 (or however many) linear regressions run, and how the observations farther from the current prespecified X value are weighted in the regression. The shape of the kernel, which defines how quickly weights fall to zero as you move away from the given X value, is less important than the bandwidth, but the kernel is important in deciding what a given bandwidth means in practice. Figure 1.3 shows how the Gaussian and Epanechnikov kernels include twice as many observations as most other kernels with the same bandwidth.

Here's an example using the simplistic auto data (note that mileage is approximately inversely related to weight, which is why US fleet standards use a harmonic mean, and the rest of the world uses fuel consumption per distance rather than distance per fuel expenditure to measure efficiency) which illustrates how a small bandwidth that minimizes bias for a local estimate can produce high-variance estimates especially where data is sparse:

Exhibit 1.2.14 *Local linear regressions with various bandwidths.*

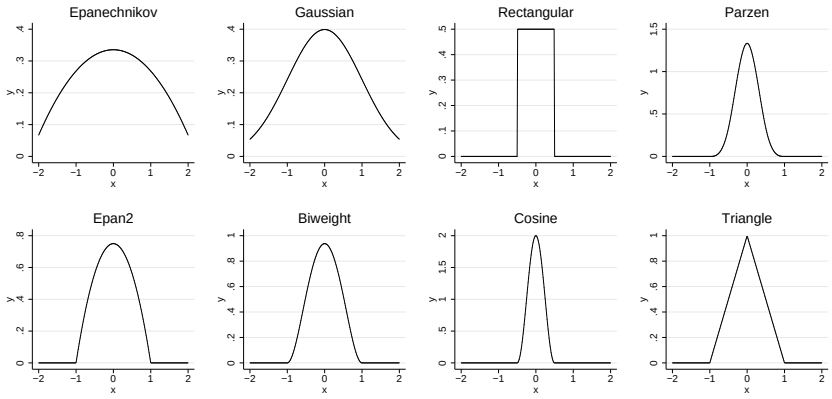

Figure 1.3: Kernel choices.

```
clear all
sysuse auto
lpoly mpg weight, k(tri) deg(1) bw(100) gen(x1 y1) nogr
lpoly mpg weight, k(tri) ddeg(1) bw(200) gen(x2 y2) nogr
lpoly mpg weight, k(tri) ddeg(1) bw(400) gen(x3 y3) nogr
sc mpg wei||line y3 x3||line y2 x2||line y1 x1, name(n) ti(Nonparametric)
qui glm mpg weight, link(power -1)
predict mhat
sc mpg weight||line mhat weight, sort name(p) ti(Parametric)
gr combine n p, nocopies ycommon graphr(fc(white)) xsize(6) ysize(3)
```

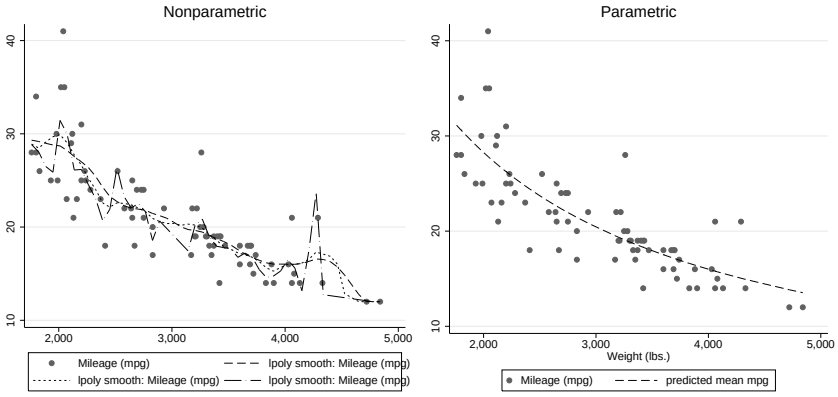

Figure 1.4: Local linear regression.

While local polynomial regressions may be useful for exploring functional

form, we assume here that theory provides guidance on functional form. Recovering functional form where it is unknown, and where the variables that are arguments of the function that determines the outcome are also endogenous, is too hard a problem for this book. If one has plausibly exogenous explanatory variables, one can engage in "model building" by using local regression, restricted cubic splines (Stata command `mkspline`), multivariable fractional polynomials (Stata command `mfp`), or penalized splines—see `pspline` on SSC and Ruppert et al. (2003).

1.3 Regression for Limited Dependent Variables

Often, the outcome variable is not continuous and unbounded, as is typically assumed for a linear regression (if e in the equation $y = Xb + e$ is normally distributed, the outcome variable y must be continuous and unbounded). Typical cases are where the outcome is a binary (zero or one, no or yes) variable, a categorical variable, a count, or a nonnegative variable (such as labor earnings, which may be zero but not negative), including a duration (such as time to exiting some state).

1.3.1 Linear Probability Model

Even if the dependent variable is a binary (zero or one) variable, one can still run a linear regression, called the linear probability model or LPM. However, the error is either $1 - Xb$ or Xb for each observation, which is obviously not normally distributed. If we can specify how the error is distributed, we will get a much more efficient estimate, in general. Even if we don't, and just run OLS, we want to make sure our estimates are heteroskedasticity-robust, since heteroskedasticity is guaranteed. One way to improve efficiency is to use weighted least squares, but `probit` and `logit` and similar models are more common estimation strategies with a binary zero/one variable. One advantage of the linear probability model is that the estimates are directly interpretable as the marginal effect of X on the predicted probability y is one; one disadvantage is that this marginal effect is assumed to be the same regardless of the level of X, which is never true. Imagine a case where the coefficient on X is one half and the probability at mean X is one quarter. An increase in X of one tenth may produce an increase in the probability of about 0.05 at mean X, but now imagine X is two units higher and the predicted

probability is above 0.95; now it is simply not possible for an increase in X of one tenth to produce an increase in the probability of about 0.05.

1.3.2 Probit and Logit and alternatives

A useful starting point is to construct a linear index model

$$y = I[Xb + e > 0]$$

where the indicator function $I[.]$ is one when the condition is true, and zero otherwise. We can also write

$$\Pr(y = 1) = F[Xb]$$

and get an equivalent model if F is the cumulative distribution function of e. If we pick a family of distributions for e, or a family of distribution functions for F, we can easily estimate via maximum likelihood. Assuming e is normal (or F is the cumulative normal distribution function) produces a `probit` model and assuming e is logistic (or F is the cumulative logistic distribution function) produces a `logit` model.

The easy way to conceptualize these two models in comparison with the linear model is to imagine regressing on a single X variable and graph predicted probabilities. It's easy to see in figure 1.5 that all three models are similar near the center, but have different behavior in the tails. The `probit` and `logit` are so similar as to be nearly identical unless the effect of a variable is very large or its variance is very large, so that predicted probabilities in the tails are more common.

In the `probit` and `logit` models above, we have to pick a variance s for the error term as well. Even in the case where the family of distributions is chosen correctly, `probit` and `logit` and similar models identify b/s, not b. That is, the coefficients are known only up to some scale, but ratios of coefficients can be identified, since the scaling parameter s cancels for a ratio of two coefficients. There is also no problem for predicted probabilities conditional on X. However, if s is not independently and identically distributed, the `probit` and `logit` models fall apart. See the manual entry for the heteroskedastic probit `hetprob` for one approach to that problem. If errors e exhibit "unobserved heterogeneity" (for example, the "error components" model of a fixed-effect probit), point estimates are typically biased in nonlinear models (Yatchew and Griliches 1985), but the marginal effects can be estimated under fairly weak assumptions (Wooldridge 2005a); see also 3.7.

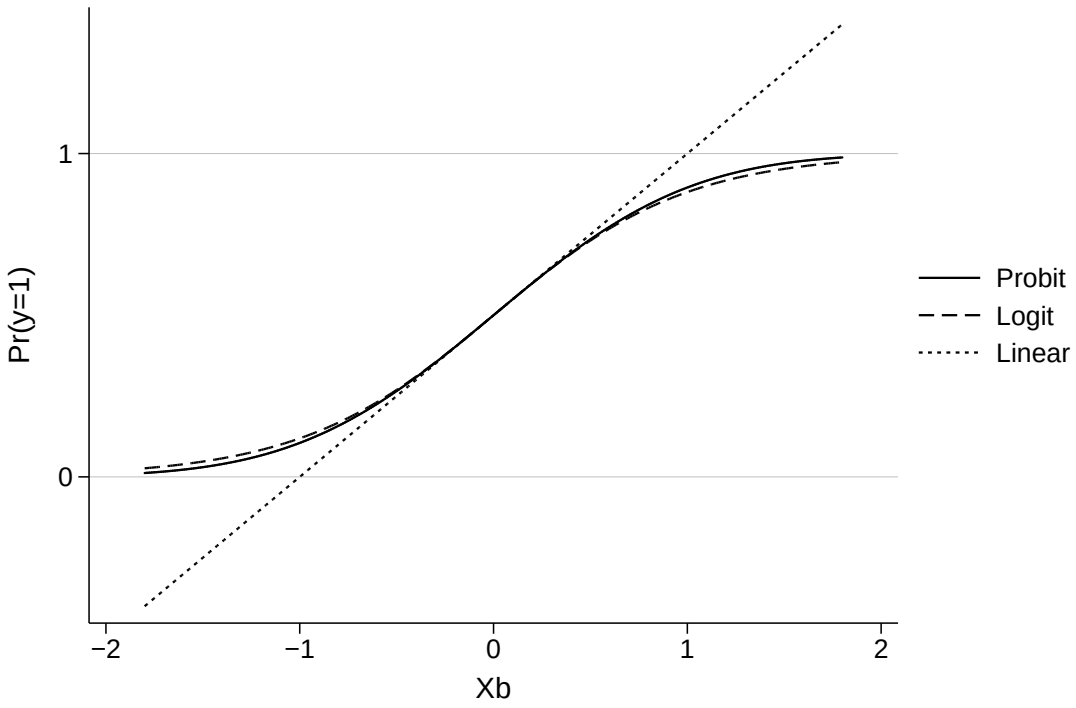

Figure 1.5: Logit, probit, and LPM.

The `probit` and `logit` models both produce estimated coefficients that are not directly usable, except for comparisons of ratios of coefficients; see also Ai and Norton (2003) and Norton et al. (2004). The marginal effects are often computed as a derivative at the mean (no longer the default behavior as of Stata 11, I am happy to report) or as the average across some group of observations (e.g. the whole estimation sample). The estimated effect on expected y (predicted probability) for a one-unit change in one variable x for an individual i can be approximated as:

$$E(y_i|Xb+g(x+1))-E(y_i|Xb+gx) = F(Xb+g(x+1))-F(Xb+gx) \cong \frac{dp_i}{dx}$$

$$\frac{dp_i}{dx} = D_x p_i = D_x F(X_i b + gx) = f(X_i b + gx)g$$

but the approximation is only good for small gx and of course

$$E\left[f(X_i b + gx)g\right] \neq f(E[X_i b + gx])g$$

in general, when f is nonlinear.

Variants of the logit and probit models include choice models (discussed in help files and manual entries for `asclogit`, `asmprobit` and `asroprobit`). A close relative of many choice models is the Conditional Logit (see help files and manual entries for `xtlogit` and `clogit`). Other models in this family include Ordered Logit and Probit (see help files and manual entries for `ologit` or `oprobit`) for ranked categorical outcomes, Multinomial Logit and Probit (see help files and manual entries for `mlogit` or `mprobit`) for unranked categorical outcomes, and Stereotype Logit (`slogit`, a compromise between `ologit` and `mlogit`). There are a number of user-written variants as well, findable with the `findit` command, including `gologit2` and `soreg`.

1.3.3 Survival Regression

In the bad old days, researchers regressed duration (time in some state, e.g. poverty or unemployment, as an outcome variable) on covariates, but there are a variety of problems with that approach related to censoring and truncation, and appropriate functional forms. Today, researchers use a "survival" regression, treating the hazard of an event occurring at each point in time, instead of the time to first event, as the outcome variable which is a function of X. In these models, an event is typically called a "failure" because of many implementations designed to model

equipment failure. An event might be death in some models, hence "survival" regression.

Time may be thought of as continuous or coming in discrete chunks, and may be measured either way in the data. Continuous-time models are discussed at length in [ST], for example in the manual entries for `streg` and `stcox` for estimating the classic Cox (1972) proportional hazard model, and Cleves et al. (2008), so the focus here is on discrete-time models. In fact, when periods of time are very short, the two classes of model are virtually identical, whether we think of events as actually only occurring (or not) in each period, or whether they may happen in continuous time and they are "interval censored" i.e. we only know which longer time period they occur in. Time periods in a state, up until an event occurs, are known as "spells" (e.g. a spell of unemployment, where getting a job is the event).

Discrete-time hazard models are discussed in detail in Singer and Willett (2003) and Jenkins (2005), among other sources. The hazard of an event that can occur with some probability within a period is the probability of occurrence within the current period conditional on not having observed an occurrence in the periods preceding the current one. For a probability distribution f(t) over time until event, with the cumulative probability F(t), the hazard is h(t)=f(t)/[1-F(t)], which in a discrete-time model is the conditional probability of failure in the current period (conditional on not having already failed). F(t) is also called the failure function, since it specifies the probability of observing a failure by time t.

One appealing way to model the hazard is to specify that the odds of observing an event in the current period conditional on not having observed an occurrence in the periods preceding the current one are proportional to some baseline odds (odds when X=0) that varies with elapsed time:

$$\frac{h(t|X)}{1-h(t|X)} = \frac{h(t|X=0)}{1-h(t|X=0)} \exp[Xb]$$

so

$$\text{logit}[h(t|X)] = \ln h(t|X=0) - \ln[1-h(t|X=0)] + Xb$$

Maximizing the likelihood of this "proportional odds" model is equivalent to estimating a logit model (estimating a cloglog model on identical data gives a similar "proportional hazard" model) with suitable controls for time periods. The logit regression includes observations on individuals for each time period up to and including the date of an event, where the outcome variable is zero for each period at risk where an event did not occur, one in the first period where an event occurred, and missing elsewhere. Thus, when someone in a eight-month-long survey is poor

for the first time in month two, beginning a spell of poverty, then exits the spell in month six and is not poor again, their sequence of outcome values is .,0,0,0,0,1,.,. in a logit regression of exiting poverty on covariates (note that covariates need not be constant over time). The underlying data on poverty is 0,1,1,1,1,0,0,0 in this example and some recoding is necessary to get the data into the right form for estimation; see Cox (2007) for relevant code.

Several common difficulties of duration modeling, including the pervasive problem of right censoring, are easily solved in this framework. The baseline odds can be captured in indicator variables for each time period (also known as time dummies), or several indicator variables for groups of time periods (such as first year, second year, third and later years), or the duration dependence can be parametrized by including a single variable that is some function of time. The advantage of parametrically modeling duration dependence (as opposed to including time dummies) is that mean and median duration can be calculated. A common specification includes the natural log of time measured in elapsed periods. When this takes a negative coefficient, the hazard decreases over time, but at a slower and slower rate.

Censoring and truncation

Censoring of a spell in some state, or the duration of such a spell, means that we do not observe the start or end date of the spell, but only know that it is above or below some threshold. Right censoring occurs when no further data is available on an individual, but no event has occurred to date. Thus, all that is known for the individual is that the event must occur after the latest observed date. This is a common feature of duration data, and typically occurs because the data comes to an end before an event occurs. Left censoring occurs when the beginning date of time at risk is not observed. If we wanted to know the duration of spells of poverty or unemployment in survey data, for example, we would need to know when each surveyed individual who is poor or unemployed at the time of the survey became poor or unemployed. Interval censoring means we only know that an event happened in some time period (we know the day, but not the time of day, or we know the month, but not which day), which is a common motivation for using discrete time survival models.

Truncation refers not to whether a variable's value is measured precisely or its value is known to fall in some range, but whether it is measured at all. Left truncation occurs when only those who have not experienced an event to date are included in the sample; for example cancer patients are sampled and only those who are still alive can be sampled. Right truncation occurs when only those who

have experienced an event are included in the sample, such as a sample composed entirely of those who got jobs after a spell of unemployment (those who are still unemployed don't appear in the sample). Unfortunately, there is little that can be done about truncation in a typical model, though Klein and Moeschberger (2003) describe various approaches.

Predicted duration

Define the survival function $S(t) = 1 - F(t)$, which gives the probability of no event to date as of each period t. Then the median duration until event is the first t such that $S(t) < .5$, i.e. half of the individuals with survival function $S(t)$ will have experienced an event by that date. The mean duration is the mean time until event, or the sum

$$m = \sum_{k=1}^{\infty} kf(x) = \sum_{k=1}^{\infty} kh(x)S(x)$$

and since S(k) becomes very small as k gets large and h(k) becomes very small as k gets large, we can compute a finite sum with very little error by brute force, adding terms until the change is negligible. Since S(k) and h(k) vary with observable characteristics X, we can calculate projected median and mean duration for each individual at each point in time.

It makes most sense to calculate projected median and mean duration for each individual at the first point in time they are at risk, and then take the mean of projected median and mean duration across all individuals. This answers a question of the form, what would I predict duration until the first event to be, conditional on being observed newly at risk? Survey weights are straightforward to use in computing a weighted mean across individuals. We cannot account for possible future changes in characteristics in future time periods, but the evolution of these characteristics is to some degree captured in the measured risk differentials. Interactions between explanatory variables in this setting would capture more projected future changes, but would result in estimates much more difficult to interpret. In general, we seek a balance between a suitably saturated model so that important differences in risk may be observed, and a parsimonious one so that we may make sense of our estimates.

Heterogeneity

A common concern is that some individuals may simply have inherently lower risk than others (lower "frailty" in the parlance of survival regression). It is possible to model individual-level heterogeneity in frailty by assuming a distribution

for individual-level frailty terms. For example, if we assume gamma heterogeneity we can use the method described by Meyer (1990), and if we assume normal heterogeneity, we can use `xtcloglog` in place of `logit` (Jenkins 1995), (Jenkins 2005); see also `pgmhaz8` on SSC. In practice, one is likely to see significant differences between models that account for individual heterogeneity and those that don't, but not substantial differences between various models that account for individual heterogeneity. Thus the (essentially arbitrary) choice of distribution does not seem too restrictive, and less parametric alternatives also often produce similar results. Of course, it is the responsibility of the researcher to estimate several plausible alternative models to assess the robustness of results in the given data.

Multiple failures

When the same individual may experience more than one event in the data (think of a model predicting the end of a spell of smoking, where quitting smoking is the failure, and recurrence is common), we cannot usually treat these separate spells or failures as independent draws (Cleves 1999). One method for accounting for dependence is to explicitly model heterogeneity across individuals but assume the separate times to failure are independent draws from that individual-specific distribution; another is to adjust the VCE for clustering, which is very simple in practice in the discrete-time model with `vce(cluster id)`. If order of failures is important, and we have information on order, we can use explicit models for each failure, treating each in turn as a single-failure model. For example, second marriages may have an intrinsically higher or lower failure rate than first marriages, conditional on individual frailty and covariates, and we often have information on marital history.

While multiple-failure data can be more troublesome, it also offers the potential for improved estimates with higher efficiency and more plausible assumptions justifying estimation. For example, multiple-spell data can allow for identification under much less stringent conditions than single-spell data (Honoré 1993).

Competing risks

In many cases where more than one event can take place, one event may preclude another. For example, if we are modeling the duration of marriage, death competes to occur before divorce. If we observe someone married until death, it may be the case that they would have gotten divorced the next year, so we may want to treat the death as a censoring of the observed data (pretending that the divorce would have happened eventually had they lived forever), but a more natural

model is one that assumes people are at no risk of divorce once they are dead.

While censoring means you cannot observe the event of interest, but know that it happened after the last observed date, a competing event prevents the event of interest from occurring, which requires a different model for many purposes. For example, the Kaplan and Meier estimator is no longer a good measure of prevalence for an event of interest. Gooley et al. point out that one minus the KaplanMeier estimator is a biased estimator of the failure function F(t) because the KaplanMeier estimator treats competing events as if they were censored.

Competing risks models specify the time to failure time as the minimum of a set of competing "latent" failure times. Unfortunately, the joint distribution of latent failure times is not identified from the distribution of minimum times to failure together with types of failure (Cox 1959; 1962, Tsiatis 1975, Gail 1975). The joint distribution of the latent failure times can only be identified if some additional structure is imposed, for example independence of latent failure times. Stata 11 introduced the Fine and Gray (1999) semiparametric proportional hazards model for continuous time, as described in the manual entry for `stcrreg` (see especially the section **stcrreg as an alternative to stcox**).

A popular class of competing risks models assumes that the hazards have a Mixed Proportional Hazard specification, where hazards depend multiplicatively on elapsed duration, observed regressors and unobserved heterogeneity (frailty) components; for more detail, see e.g. Lancaster (1979; 1990), Vaupel et al. (1979), Kiefer (1988), Van den Berg (2001), Hahn (1994), and Ridder and Woutersen (2003). Heckman and Honoré (1989) extend the Mixed Proportional Hazard model using a nonparametric model, and Abbring and Van den Berg (2003a) extend their result to a larger class of empirical applications, and multiple failure data.

Ordered choice models are also used as duration models. (Ridder 1990) showed the equivalence of the conventional ordered choice model and Generalized Accelerated Failure Time models for discrete time duration data, which include the Mixed Proportional Hazard model as a special case. (Cunha et al. 2007) added stochastic thresholds and interval-specific outcomes, and discuss nonparametric identification of the richer model.

1.3.4 Tobit

The Tobit model (Tobin 1958, McDonald and Moffitt 1980, Amemiya 1984, Breen 1996) in its pure form assumes that

$$\begin{array}{rcl} y = y* & \iff & y^* = Xb + u \geq 0 \\ y = 0 & \iff & y^* = Xb + u \leq 0 \end{array}$$

i.e. a positive outcome y is observed only when some latent variable is positive ($y^* = Xb + u > 0$) and otherwise is $y = 0$. The error is assumed to be normally distributed and results are very sensitive to violations of that assumption. In addition, the "lower limit" may be some value other than zero, and there may also be an upper limit (see the , but these values must be known. If a lower limit is not specified, Stata will assume the lowest observed value of y is the lower limit, but if this is not the correct assumption, biased estimates will result.

The output is also hard to interpret because estimates of b do not easily translate into effects of x on y. There are a variety of marginal effects possible after `tobit` including the effect on the expected value of the latent variable $y*$, the effect on the expected value of the outcome variable conditional on it being larger than the lower bound, and the effect on the probability of the outcome variable being larger than the lower bound. By default Stata's `margins` (or `mfx` in older Stata) command for marginal effects `margins, dydx(*)` shows the effects on the latent variable, which are just the coefficients reported by `tobit`. You have to specify the `predict()` option to `margins` (or `mfx` in older versions of Stata) to get the other marginal effects; see the manual entry on `tobit postestimation` for more.

Suppose the outcome is hours of work, and we observe wages for everyone (including those at zero hours). We could think of the $y = 0$ observations as wanting to work exactly zero hours, or to "buy labor" at that wage, i.e. to work negative hours, if such a thing could be observed in our data. Suppose we think that the log of hours is a function of the log wage, not hours, i.e. a given percentage increase in wage causes a constant percentage increase in hours worked, in expectation. The trouble is, the log of zero is undefined, so the lower limit cannot be defined. Sometimes researchers will define the log of hours in this case to be some number lower than the lowest observed value, which produces biased and inconsistent estimates (using `intreg` gives the same wrong answer). It makes no sense to think of negative hours if a given percentage increase in wage causes a constant percentage increase in hours worked, in expectation. It also makes no sense to think of zero hours in that case!

Often, a researcher wants to model the log of y as a linear function of a set of variables X, but there are some cases where $y = 0$, so the outcome variable `lny` is missing since the log of zero is undefined. One approach used in practice are to replace zero outcomes with a small positive number a (with a smaller than any observed positive outcome) and then take logs and run `tobit` with a lower limit at $\ln(a)$. The only situation where this produces good estimates is where there is imperfectly sensitive equipment measuring outcomes, where a is the detection limit, so any outcome below a is recorded as a zero, and a is known *a priori*. One example in economics is where outcomes are rounded, say to the nearest hundred, so the lower limit on detection is 100, and a zero might be 50 or anything less, whereas any outcome in the interval from 50 to 150 is recorded as 100. Even if a detection limit describes the general type of problem, but the lower limit a is unknown, picking a small positive number a (with a smaller than any observed positive outcome), taking logs, and running `tobit` with a lower limit at $\ln(a)$ is a bad idea (Carson and Sun 2007).

Another approach is to use a two-part model or a hurdle model (McDowell 2003), or to use the positive values only in a selection model (see section 4.6). But if we believe the conditional mean of y is a exponential function of a linear index Xb (similar to the assumption that the log of y is a linear function of a set of variables X), there is a different approach using GLM or `poisson` in section 1.3.5 that gives consistent estimates. On the other hand, we will see in section 1.5.4 that it makes no sense to regress hours on wage (or quantity on price in any demand system) using any functional form unless we have very special data indeed.

In short, the Tobit model is a finicky model, and I have never seen an empirical application where it made the most sense. In particular, some kind of GLM or two-part model always seems preferable. With endogeneity concerns, one would almost always prefer a GMM model (section B.7.6) to `ivtobit` or the like.

1.3.5 GLM and Poisson

The Generalized Linear Model includes linear regression, logit, probit, and Poisson regression as special cases, in addition to a range of other models. The idea is to use a linear index Xb but introduce a link function g such that

$$E(y) = g^{-1}(Xb)$$

and specify a distribution type for the outcome y. Hardin and Hilbe. (2007) provide a wealth of information on various link functions and families of distribu-

tions.

One common model is to assume a log link, so

$$E(y) = \exp(Xb)$$

which is the same as assuming that the log of $E(y)$ is a linear function of X. Note that regressing $\ln(y)$ on X assumes that $E(\ln y)$ is a linear function of X, but it need not be the case that $\ln(E(y))$ is a linear function of X. However, if we postulate an error term such that

$$y = \exp(Xb)u = \exp(Xb)\exp(v) = \exp(Xb + v) \Rightarrow \ln(y) = Xb + v$$

assuming $y > 0$ there is a natural analog to OLS with a logged dependent variable. The only difference in assumptions about error terms, and of course that `glm` and the equivalent `poisson` model allow observations where $y = 0$. When there are a lot of observations where $y = 0$, the results of OLS using $\ln(y)$ and GLM using y are likely to differ more, but GLM may also fit the data worse in this case. If the cases where $y = 0$ should be small positive values (i.e. zeros are never true zeros) but are observed as zeros due to rounding to a known precision, it may make sense to use a Tobit regression with $\ln(y)$ instead; if they represent the outcome of a different process, a two-part model may be appropriate; if they represent the result of overdispersion, a negative binomial regression (`nbreg`) or zero-inflated model (e.g. `zip`) may be more appropriate.

Another common model is to assume a logit link, so

$$E(y) = \text{invlogit}(Xb) = \frac{1}{1 + \exp(-Xb)}$$

for a model where y is not binary, but is a fraction or proportion, as proposed by Papke and Wooldridge (1996). Stata's `glm` command was modified to fit that model with options family(binomial), link(logit), and robust. Note that if y is binary, this model produces essentially the same output as `logit` with the `robust` option.

The `glm` command also provides the capability to define new link functions and families of distributions; see the manual entry for `glm` for details.

1.4 Properties of Estimators

The way to get good estimates for any particular case is to ensure that our estimator has good general properties. These may be finite-sample properties, meaning for a sample of size n with characteristics like our data, we can say that the

distribution of the estimator has certain properties. Or these may be asymptotic properties, meaning that as we collect samples of size n with n increasing without bound, we can make statements about the limit of that sequence which are reasonably good approximations for large samples. Often asymptotic approximations are good with n fairly small, at 50 or 100, but sometimes millions of observations do not guarantee that the asymptotic approximation works well, so it is always a good idea to get a sense of small-sample performance using a simulation (see B.6).

1.4.1 Bias and Consistency

If $E(\widehat{\theta}) = \theta$ then the estimator $\widehat{\theta}$ is unbiased. This means the estimate is right on average, but says nothing about how close we are on average, or whether we get closer on average as we get more data. An estimator is consistent if the whole distribution of the estimator shrinks toward the true value as the sample size gets large, as shown in figure 1.6. A consistent estimator may be biased in any finite sample, but it gets closer on average with large enough samples. The term "asymptotically unbiased" is used to mean either that the estimator is "consistent" or that the bias shrinks toward zero with increasing sample size, which of course is a different concept, so we will avoid the term.

Formally, if $\text{plim}(\widehat{\theta}) = \theta$ then the estimator $\widehat{\theta}$ is consistent (see section A.3.4 for more). The rate of convergence is also important; if we had two estimators that converged at rates $\sqrt{N}$ and $N^{\frac{1}{3}}$, we would prefer the first estimator because it converges at a faster rate—noting that 100 observations for the first estimator corresponds to 1,000 for the second. Of course in small finite samples, the asymptotic rate of convergence is not strictly relevant; but we know we need much larger samples to get good results with an estimator that converges more slowly. See A.3.4 for more.

1.4.2 Efficiency and MSE

The variance of an estimator $E(\widehat{\theta} - E(\widehat{\theta}))^2$ is at least as important in most circumstances as bias or consistency. If one had to choose between the biased but low-variance estimator $\widehat{\theta}$ and the unbiased but high-variance estimator $\tilde{\theta}$ shown in figure 1.7, there are many circumstances where we would prefer $\widehat{\theta}$, if large errors are more costly than small errors.

The mean squared error of an estimator $E(\widehat{\theta} - \theta)^2$ is a common criterion of how good an estimator is at getting an answer close to the truth, incorporating bias

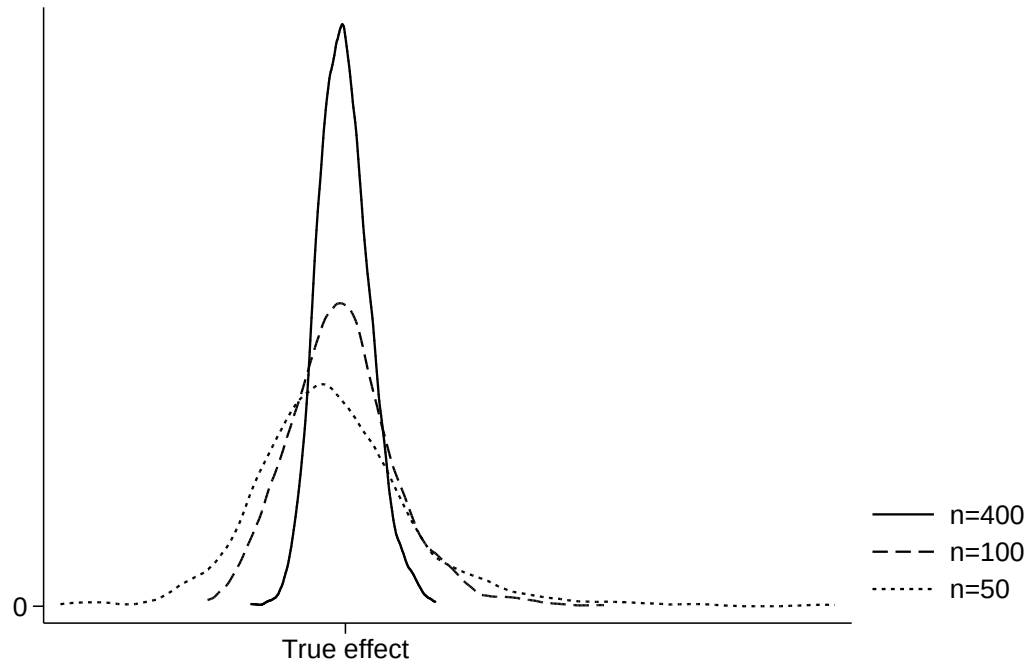

Figure 1.6: A biased but consistent estimator honing in on the true effect.

and variance:

$$MSE(\widehat{\theta}) \equiv E(\widehat{\theta} - \theta)^2 = Var(\widehat{\theta}) + \left[Bias(\widehat{\theta})\right]^2$$

The efficiency of an estimator $\widehat{\theta}$ is the minimum possible variance for an unbiased estimator divided by the actual variance of $\widehat{\theta}$. The Cramér-Rao bound (Cramér 1946, Rao 1945) can be used to prove that $e(\widehat{\theta}) \leq 1$ always, and an estimator is efficient if $e(\widehat{\theta}) = 1$.

Often, there is some kind of tradeoff between bias and variance, and trying to minimize mean squared error is only one way to balance competing objectives. Suppose we have a family of estimators where we can pick a parameter to get low squared bias, or low variance, but not both, and the two are roughly inversely related, e.g. $b^2 v = 1$ or $v = 1/b^2$. The choice that minimizes mean squared error is shown as the tangency between the possibility frontier $v = 1/b^2$ and the constant-MSE line $v = 1 - b^2$ in figure 1.8. But if we want lower bias, or lower variance, we would pick different points along the frontier. The choice depends on whether we care more about being right on average, or large errors, i.e. how costly it is to make large errors in any particular case.

Simulations (see also B.6) are often helpful in determining whether the small-sample behavior of the estimator is acceptable in data that looks like yours (has the given sample size and covariance structure). You can run a simulation for some independent normally distributed X variables picked with `drawnorm` and get one result, then run for some data that looks like yours and get a totally different result, so it makes sense to use data that looks like yours. It's easiest to just start with your data (`use` the data in the first line of a `program` to be called by `simulate`, and modify it as needed. The modifications would be: you specify the errors and the coefficients in each run of the simulation, so that you know the true relationship between X and y, then you try to estimate it. The simulation `program` specifies distributions for error terms by drawing all the error terms needed, and when you repeat this process several thousand times, you can assess the distribution of estimated coefficients around true coefficients, and rejection rates (to assess power or rejection rates, you should use many thousands of simulations). Of course, this is proof by example, so no proof at all, really, but it is convincing evidence nonetheless.

Simulations are especially helpful for assessing statistical power. We specify α (the size of a test, or its significance level), the chance we reject a null hypothesis that is true, but do not know β (the chance we fail to reject a null that is false, which clearly depends on both the null and the true parameter value). Power,

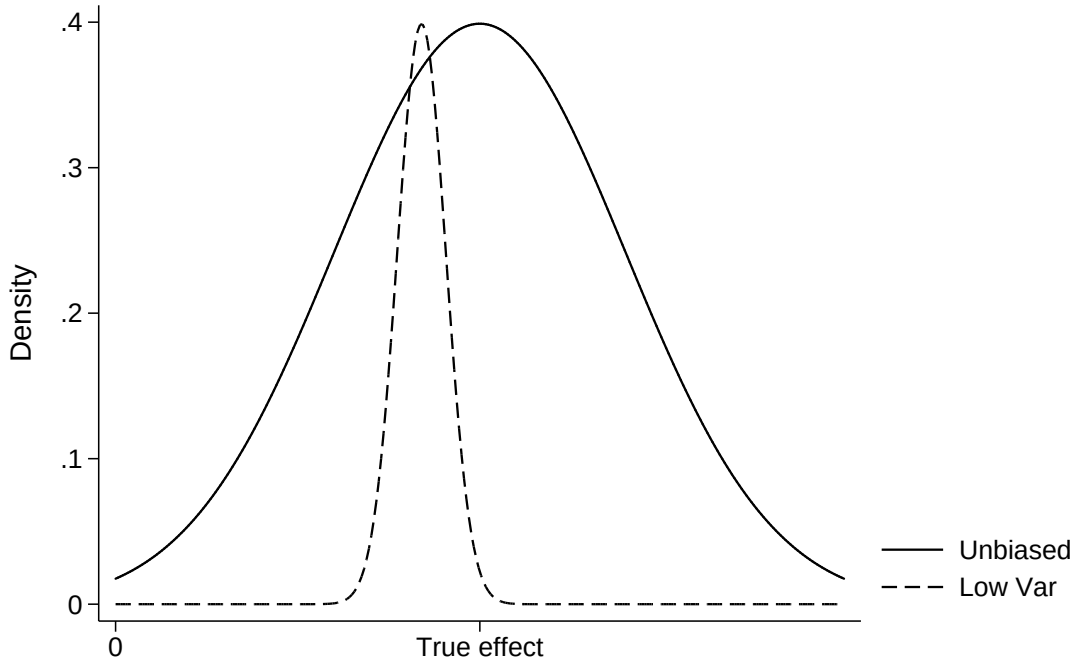

Figure 1.7: A consistent but high-variance estimator versus a biased but low-variance estimator

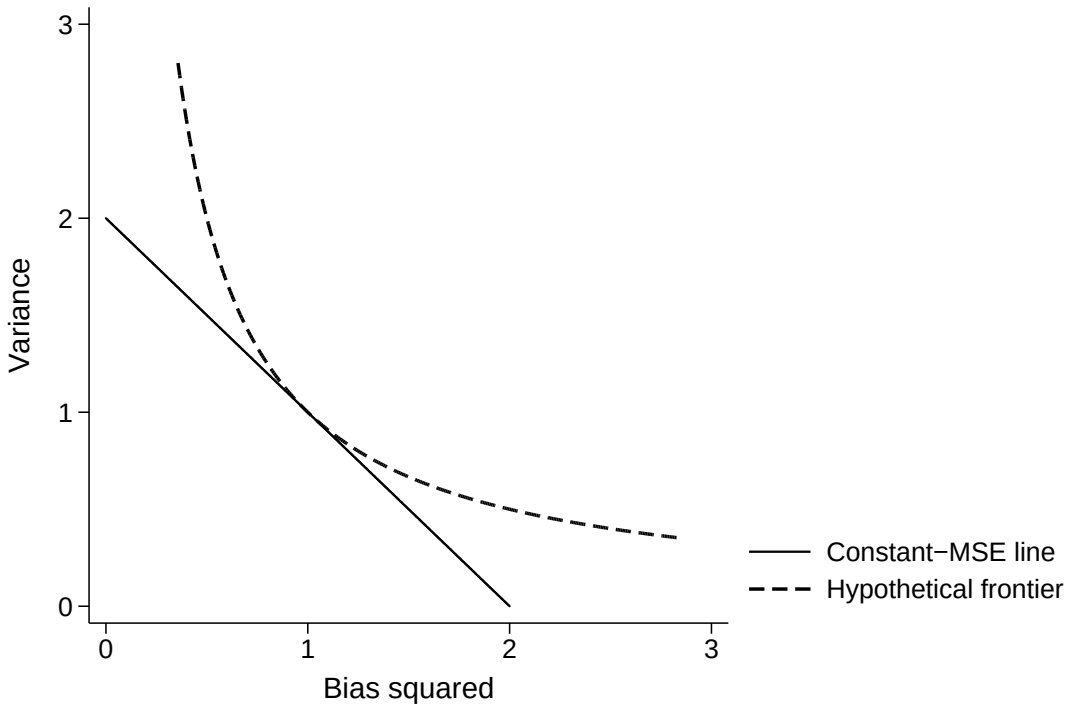

Figure 1.8: A bias-variance tradeoff and MSE minimizing choice

or one minus β, should be as close to one as possible for true parameter values that are plausible, though eighty percent is often taken as an acceptable minimum for power at the true parameter value (implying that one time in five we fail to reject an incorrect null). The power curve graphs power as a function of the true parameter value; it passes through α at the null and rises toward one on either side. If we have several alternative estimators and associated tests, we can compare their power curves—if one estimator's power curve is higher than all others' we say it has Uniformly Maximum Power, which is clearly a good recommendation for that estimator and test.

A useful summary measure of efficiency and power is the minimum detectable effect of a design (the design includes the data collection and estimation strategy, with a test at the end). Assume the null is that the effect of x on y is zero, and the effect may be negative or positive in reality. The minimum detectable effect for a given estimator and test usually answers the question, if the true effect of x on y is b, then how small (in absolute value) does b have to be before the power falls to eighty percent? We could also specify ninety percent, in which case the minimum detectable effect is larger (we fail to detect small effects with higher probability as the effect gets smaller, so if we want to detect small effects with higher probability the effect must be larger). See the Stata commands `sampsi` and `stpower` for more discussion.

1.5 Experiment and Quasi-Experiment

The models discussed so far assume that X is fixed by the researcher, as would be the case with an experiment involving pea plants or assigning drug and placebo for patients. The ideal situation is in fact where the X is randomly assigned to each unit, so it cannot possibly be correlated with any characteristics not measured in the data which would appear in the error term. In the case without random assignment, it is useful to consider how the experiment would have been designed had it been conducted.

1.5.1 Design of Experiments

There is an enormous literature on the design of experiments (Fisher 1926, Neyman 1923, Fisher 1935, Neyman et al. 1935, Cochran and Cox 1957, Cox 1958, Rubin 1990, Rosenbaum 2002), but we just need to scratch the surface. In the simplest experimental design, we randomly assign to each unit a single treatment, or

control. Then the estimate of effect is simply the difference in mean outcomes between groups, obtained with a regression of outcome on a dummy for treatment. We can generally get more efficient estimates, however, by regressing not only on a dummy for treatment, but a variety of observable factors measured before treatment was applied ("baseline" characteristics).

Lady Tasting Tea

A classic example of a simple experiment is a test of a lady's claim that she can tell by drinking it whether tea was made by pouring first milk and then tea or pouring tea first and then milk. Fisher (1935) points out that if you randomly present 4 pairs of cups of tea of each type, and she can identify the two types with probability no better than chance, her chance of correctly identifying each is one in seventy (the number of ways to present a sequence of eight cups of tea of two types). So if she correctly identifies each, we reject the null that she can identify the two types with probability no better than chance (since the p-value 1/70 is less than 0.05), and if she misidentifies any, we fail to reject.

Factorial Designs

Fisher (1926) argued that a good experiment conditions on numerous observable characteristics, to answer questions about the effect of treatments in many different environments: "No aphorism is more frequently repeated in connection with field trials, than that we must ask Nature few questions, or, ideally, one question, at a time. The writer is convinced that this view is wholly mistaken. Nature, he suggests, will best respond to a logical and carefully thought-out questionnaire."

The idea is to divide each potential unit of analysis, or "whole plot" in an agricultural setting, so that each possible combination of factors is treated; this is called a Split Plot Design, and is often implemented using a Latin Square. A Latin Square is a convenient way to randomly assign treatments to each combination of factors. A design where some of the many possible interactions of factors is omitted is called a "fractional factorial design."

In a factorial design, the estimate of effect is simply the difference in mean outcomes between treatment and control groups averaged across all combinations of factors, obtained with a regression of outcome on a dummy for treatment, and dummies for each level of each factor and interactions.

Randomizing treatment within strata of combinations of factors and controlling for the factors not only reduces the potential of an unlucky unbalanced distribution of factors giving a poor estimate using the simple difference in means. It

also reduces the variance of estimates, and increases our statistical power.

Compliance and ITT

The units assigned to treatment groups in a physical science experiment (e.g. fertilizer in an argicultural experiment) typically cannot choose to leave their assigned group, but in a social or economic experiment, the units may "fail to comply" with the experimental protocol. For example, if you assign someone to take a training course, they may not show up, or someone who was assigned to the control group may sneak in.

If compliance is less than perfect, the treatments can be redefined to measure "assignment to treatment group T_j" rather than "received treatment T_j." This is called the "intention to treat" analysis, and measures the causal effect of assigning someone to a treatment group, which is still randomly assigned, as opposed to the treatment itself, which is no longer randomly assigned once individuals nonrandomly fail to comply. To recover causal estimates of average treatment effects, we can use instrumental variables techniques described in 4.2.

1.5.2 The Fundamental Problem of Causal Inference

We already saw the sources of the problem in the context of kidney stone treatment (section 1.1.6) and education as a treatment (section 1.1.9), which is that we don't observe the counterfactual outcome; we don't get to see what would have happened had treatment cases not gotten treatment or control cases gotten treatment. When treatment is not randomly assigned, we can't estimate the average effect of treatment as we would in an experiment.

The dominant model of causal inference draws on the work[10] discussed in 1.5.1 and postulates that every unit under analysis has outcomes given any variety of treatment we consider applying (just the two varieties "treatment" and "control" in the simplest case), and that these outcomes depend only on the unit's characteristics, not what kind of treatment other units receive.

For any unit i we can write

$$\delta_i(T \text{ versus } C) = y_i(T) - y_i(C)$$

[10]The dominant model of causal inference is often called the Rubin Causal Model for Rubin (1974) or the Neyman-Rubin Model (Neyman 1923, **?**), or the Neyman-Rubin-Holland Model (Holland 1986a), or sometimes the Roy Model (Roy 1951, Heckman and Honoré 1990, Heckman 2008), but I prefer the less loaded Counterfactual Causal Model; the crucial steps are the imagining of counterfactual outcomes under any type of treatment, and seeing what effects can be identified. See Pearl (2000) for relevant discussion and improvements.

to measure the hypothetical effect of moving unit i from some treatment or level of treatment C to some treatment or level of treatment T.

The "Fundamental Problem of Causal Inference" is that we only observe one outcome—each unit either gets treatment or not. So we can never measure the individual treatment effect δ_i in any real sense, but we can measure the average treatment effect across individuals in this framework if we randomly assign treatment to many individuals.

The average treatment effect (ATE) for all individuals i is

$$E\left[\delta_i(T \text{ versus } C)\right] = E\left[y_i(T)\right] - E\left[y_i(C)\right]$$

but for one subgroup we observe $y_i(T)$ and for another $y_i(C)$. Let $y_T(T)$ be the mean outcome of the treated group given treatment T, $y_T(C)$ the mean outcome of the treated group given treatment C, $y_C(T)$ the mean outcome of the control group given treatment T, and $y_C(C)$ the mean outcome of the control group given treatment C. Then

$$E\left[\delta_i(T \text{ versus } C)\right] = p_T\left[y_T(T) - y_T(C)\right] + (1 - p_T)\left[y_C(T) - y_C(C)\right]$$

where p_T is the proportion receiving treatment T. We observe sample means of y for those who get treatments T and C so we can estimate $[y_T(T)]$ and $[y_C(C)]$, but the other two are unobservable counterfactuals. However, if treatment is randomly assigned, $[y_C(T)] = [y_T(T)]$ and $[y_T(C)] = [y_C(C)]$ so we have

$$R \quad \Rightarrow \quad E\left[\delta_i(T \text{ versus } C)\right] = [y_T(T)] - [y_C(C)]$$

(where R denotes random assignment of T and C) and we can estimate the causal impact with the difference in sample means.

To make matters concrete, imagine δ_i is the same for all i but there is heterogeneity in levels, and units come in two types labeled 1 and 2:

$Type$	$E[y\|T]$	$E[y\|C]$	TE
1	90	40	50
2	70	20	50

The problem is that the treatment T is not applied with equal probability to each type. For simplicity, suppose only type 1 gets treatment T and put a missing dot in where we cannot compute a sample mean:

$Type$	$E[y\|T]$	$E[y\|C]$	TE
1	90	.	?
2	.	20	?

The difference in sample means overestimates the ATE (70 instead of 50); if only type 2 gets treatment the difference in sample means underestimates the ATE (30 instead of 50).

Random assignment puts equal weight on each of the possible observed outcomes:

$Type$	$E[y\|T]$	$E[y\|C]$	TE
1	90	.	?
2	.	20	?
1	.	40	?
2	70	.	?

and the difference in sample means is an unbiased estimate of the ATE. We can even get estimates of the average treatment effect for each type:

$Type$	$E[y\|T]$	$E[y\|C]$	TE
1	90	.	?
1	.	40	?

gives an estimate of 50 on average, as does

$Type$	$E[y\|T]$	$E[y\|C]$	TE
2	.	20	?
2	70	.	?

assuming the random assignment is done properly. This was the essential point of Fisher (1926), along with the assertion that stratified randomization by type produced lower variance estimates: randomly assigning treatment in cleverly chosen strata gives more and better answers than any other method.

1.5.3 Internal and External Validity

The notions of unbiasedness or consistency (even efficiency or mean squared error, or power, or the unbiasedness or consistency of standard errors) address how well we can identify an effect. These are concerned with the internal validity of our inference, i.e. whether our hypothesis testing is likely to accurately measure the quantity we intend. A separate question is how generalizable the results are—this is the question of external validity (Campbell and Stanley 1963, Bracht and Glass 1968). For example, an agricultural experiment at Rothamsted may produce a very good estimate of the efficacy of a fertilizer in Hertfordshire but offer little guidance on what to do in Iowa.

An experiment where people (or other units that can be thought of as making decisions) are randomly assigned to treatments faces a special problem. The experiment, if done properly, can give an unbiased estimate of the causal effect for those types of people who are willing to participate in an experiment. This may quite a select group, and they may have a different average treatment effect than the whole population of interest. Thus we cannot generalize to the whole population without making a strong assumption about our experimental sample. We face a similar problem in using many of the methods in this book, in that we can identify some average treatment effect, but it may or may not be the average treatment effect we sought to estimate.

It makes sense to try to get an estimate with good internal validity first. This is like looking under the lamppost for your keys first; you may be pretty sure you didn't drop your keys there, and not finding your keys there does not prove they are nowhere on the street, but at least you can be fairly confident of the outcome of your search there before you wander off into the dark.

1.5.4 Common Sources of Biased Inference

The major problem is usually referred to as endogeneity. An explanatory variable is called endogenous if it is correlated with the error, and exogenous if not. The term endogenous applied to a variable originally meant "determined inside a system of equations" so the variable appeared on the left hand side of at least one equation, but now the term is used for the more general situation where an explanatory variable is correlated with the error, which can arise due to selection, omitted variables, measurement error, or other reasons. In fact, endogeneity is only one source of biased inference; incorrect standard errors, model misspecification, and mistakes using data are probably much more important sources of erroneous inferences. There is no general approach to avoid mistakes using data (see Hamermesh (2007) for the best paradigm), but there is a literature on cracking the endogeneity nut which is summarized in this book.

Selection

Selection bias is a term used to describe many settings, including picking a sample of cases in a nonrandom way, for example including in a regression only those cases with outcomes above some level (selection on the dependent variable) or only those cases with (unobservable) errors above some level. The selection problem we are discussing in this book is where the level of the treatment variable has been selected by the individual, who can see components of the error term that

we never can, and this induces a correlation between the treatment variable and the error term, causing bias and inconsistency.

Omitted Variables

We can also treat the selection problem as a case of omitted variables, which provides some useful intuition in many cases. For example, we might surmise that the error ε is the sum of a random noise component u and a component observable to individuals (and not to those of us who have the data) and is correlated with the treatment variable due to choice or some other joint determination. Reformulating as a case of omitted variables, perhaps with a unobservable (as opposed to merely unobserved) omitted variable, we can nonetheless make some predictions from theory to sign the likely bias.

The formula for omitted variable bias in linear regression is the useful here. With a true model

$$y = \beta_0 + X^T\beta_T + X^U\beta_U + u$$

where we regress y on the treatment variables X^T but leave out X^U (for example, because we cannot observe it), the estimate of β_T (the coefficients of interest) has bias

$$E(\hat{\beta}_T) - \beta_T = \delta\beta_U$$

where δ is the coefficient of an auxiliary regression of X^U on X^T so the bias is proportional to the correlation of X^U and X^T and to the causal effect of X^U (the omitted variables) on the outcome y. [11]

Simultaneity

A simultaneous equations model assumes that the outcome y is the regressor in another model, so it is apparent that it is not exogenous, and other variables may also appear as outcome variables in some equations, so they are also seen as endogenous. A classic example is supply and demand for a good, where price affects the quantity demanded and the quantity demanded affects price (roughly). We observe data on the intersection of different supply and demand functions (across space or time), but both curves may be shifting around, so drawing a line through these intersections tells us nothing about either function; figure 1.9 shows a bunch of unobserved supply and demand schedules with slope one and minus one).

[11] In nonlinear models, the estimate will be biased and inconsistent even when X^T and X^U are uncorrelated, though Wooldridge (2002; p. 471) demonstrates that some quantities of interest may still be identified under additional assumptions.

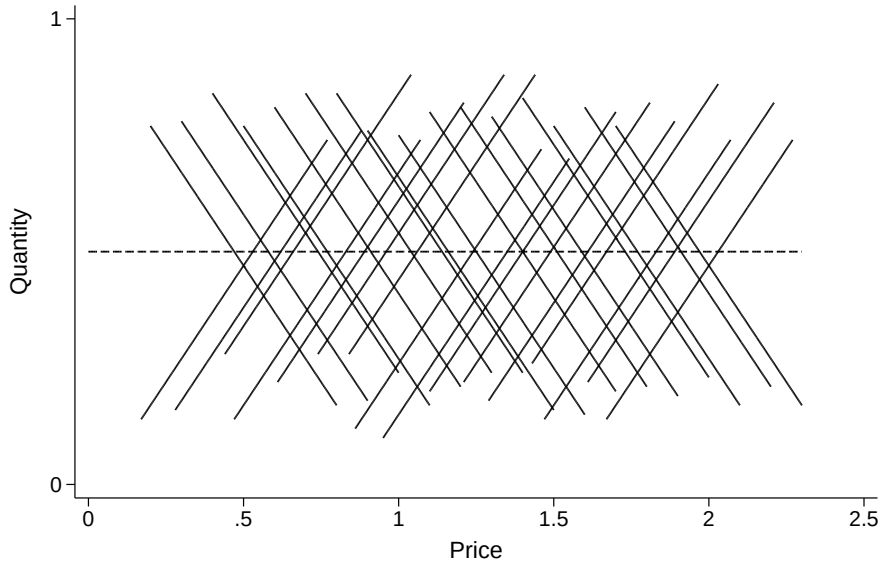

Figure 1.9: Supply and demand curves observed only at uninformative crossings

Some of the techniques discussed below to address selection bias are also used in the simultaneous equations setting. The literature on structural equations models is extensive, and a system of equations may encode a very complicated conceptual causal model, with many "causal arrows" drawn to and from many variables. The present exercise of identifying the causal impact of some limited set of variables X^T on a single outcome y can be seen as restricting our attention in such a complicated system to just one equation, and identifying just some subset of causal effects.

For example, in a simplified supply and demand system:

$$\mathrm{lnQ}_{supply} = e_s \mathrm{lnP} + a\mathrm{TransportCost} + \varepsilon_s$$

$$\mathrm{lnQ}_{demand} = e_d \mathrm{lnP} + b\mathrm{Income} + \varepsilon_d$$

where price (lnP) is endogenously determined by a market-clearing condition $\mathrm{lnQ}_{supply} = \mathrm{lnQ}_{demand}$, our present enterprise limits us to identifying only the demand elasticity e_d using factors that shift supply to identify exogenous shifts in price faced by consumers (exogenous relative to the second equation's error ε_s), or identifying only the supply elasticity e_s using factors that shift demand to identify exogenous shifts in price faced by firms (exogenous relative to the first

equation's error ε_s). We can use instrumental variables techniques (chapter 4) to estimate these impacts.

See [R] `reg3` for alternative approaches that can simultaneously identify parameters in multiple equations, and Heckman and Vytlacil (2004) and Goldberger and Duncan (1973) for more detail.

Measurement Error

Measurement error in X variables can produce substantial bias as well. The "classical" measurement error of most textbooks assumes a normally distributed random noise term is added to the true value of one X variable, and in that case, the estimator of the coefficient on that variable is biased toward zero (biased down if the true coefficient is positive, and biased up if the true coefficient is negative). However, most measurement error is not of that classical type, but involves some kind of heterogeneous rounding or mean-reverting error, or worse, deliberate and systematic misstatements. Instrumental variables techniques (chapter 4) can get consistent estimate in many cases of measurement error, for example if we have more than one measurement of the underlying variable. See also the special issue of the Stata Journal (volume 3, number 4, available in PDF format online) and associated materials discussed at http://stata.com/merror, and the help file of `ivpois` on SSC for more discussion.

Missing Data

If cases are missing data on X or y, or both, they contribute nothing to our estimates. Computing estimates excluding these cases is called "complete case analysis" and introduces no bias unless the data is missing in a systematic way (introducing selection). If cases are dropped from the analysis randomly conditional on some set of variables Z, and we know something about the process, we can always improve on estimates using a selection model (see section 4.6) or multiple imputation (Little and Rubin 2002). The idea of multiple imputation is to use a reasonable model to impute all missing values with some randomness, and do it M times, then use the variation across the estimates in the M now-complete data sets to estimate the variation due to imputation.

Stata 11 introduced multivariate normal multiple imputation (see `help mi` and the MI reference manual) and the `ice` and `mim` commands on SSC offer multiple imputation via chained equations (Carlin et al. 2008).

How to choose M is as much art as science, but Royston (2004) and Royston et al. (2009) discuss the impact of M on the precision of estimates. In general, it makes sense to pick M as large as possible given time and memory constraints,

though in some cases M equal to five or ten can produce good results.

Standard Errors

The focus in most papers and books on causal inference is getting point estimates approximately right, by using a consistent estimator and a lot of data. But inference has two big moving parts: the point estimate and the standard error. If you get a point estimate exactly right and the standard error is way too big, you may conclude there is no impact when there is one. More commonly, one gets a good point estimate but the standard error is way too small, and a small and statistically insignificant effect looks statistically significant instead.

A big part of this is that the model building part of inference is neglected and there is no correction for multiple hypothesis testing. Equally important, dependent errors may be neglected, and standard errors drastically understated. For example, in clustered sampling, correcting for correlation of errors within cluster by using a cluster-robust VCE estimator often increases standard errors by a factor of two to four; anyone who fails to correct standard errors here is pretending they have more information than they do, often with bad results.

There is also a question about survey weights in regression. Many economists adopt the position that a well-specified model is efficiently estimated by OLS, and weights only increase the variance of estimates and should therefore be neglected. However, if there is a chance the model is even slightly misspecified, using survey weights can substantially reduce erroneous inference (Binder and Roberts (2003),Binder et al. (2004)). Using survey weights (called pweights in Stata) and accounting for clustering should be considered mandatory for users of survey data, which sacrifices some efficiency for a whole lot of robustness against various mistakes.

Of course, survey weights are not free from error, either. They are constructed by people, who make mistakes, and also are frequently designed for inferences about means for the finite population, rather than inferences about relationships in a hypothetical data-generating process that created the finite population and the survey drawn from it. Nevertheless, using the weights and cluster information and neglecting any finite population correction gets you very good estimates of the true sampling error in a unstratified sample. In a stratified sample, Graubard and Korn (2002) shows that the estimates of the variance of regression coefficients need to be modified slightly, but it seems reasonable in many cases to neglect this correction.

There are a number of unsolved problems in this realm, mostly because the survey statisticians and econometricians don't speak the same language or go to

the same conferences. Much of survey statistics starts with an assumption that there is a true correlation or conditional mean in a hypothetically observable population, and imagines taking larger and larger samples from the population to get closer to the true answer. In this book, we start with the assumption that unobservable errors and treatment variables are correlated in the population so that the true relationship between outcomes and treatment variables cannot be recovered through simple regressions, even with all the population data in hand. Many of the models considered in the book are not designed for use with complex survey data, even though that is where they are applied most often. Similar problems apply with bootstrap or jackknife approaches to computing standard errors in complex survey data. In practice, weighting and accounting for clustering is probably good enough to get correct inference.

Multiple Comparisons

Another problem with the size of tests is that no paper or project uses a single hypothesis test. If we pick an alpha of five percent, accepting that we will reject a null hypothesis of zero incorrectly when the true effect is in fact zero, then proceed to look at a hundred tests, we will find about five significant effects where there are none. In Stata, `help _mtest` discusses several corrections to p-values that can be done. Bonferroni's method is the most common approach, where the p-value required for rejection in n tests is alpha divided by n, so 100 tests with a five percent nominal alpha would have a new criterion of p less than 0.0005 in order to reject the null. The danger is nowhere more clear than in Bennett et al. (2009) where the authors look for brain cell activity in salmon presented with pictures of human faces, using functional MRI, and find signficant differences in activity (typically taken as evidence of recognition activity). But their test subjects are dead salmon, so any conclusion of a causal effect is guaranteed to be erroneous. See Benjamini and Hochberg (1995) for more detail on multiple tests.

Programming Mistakes

In fact, the bias created by omitted variables, measurement error, and non-randomly missing data, and mistaken inference due to improper standard errors, probably pale in comparison to simple data management or programming mistakes. All too often, a bad merge, a data entry error, or just a typo (e.g. putting less-than sign in place of a greater-than sign in a long list of code) dominate all other errors. There is an appendix in this book on programming in Stata, and there is also a stable of books on the Stata website devoted to better programming, but there is no substitute to checking your code against someone else's.

A good way to start any major empirical project is by replicating some published result using the same data and/or method you plan to use, though of course there is no guarantee that you are not making the same mistake someone else did. Hamermesh (2007) extols the virtues of publishing replications, and the exchange between Rothstein (2007b) and Hoxby (2007) illustrates some costs and benefits of publishing replications, but David Roodman is a better example of what I have in mind. He has written two widely-used Stata programs (`xtabond2` and `cmp`) in an effort to replicate and extend papers he was interested in, and found very interesting results (Easterly et al. 2004, Roodman and Morduch 2009).

1.5.5 Spillover and Heterogeneity of Effects

The assumption underlying most causal inference methods is that the treatment does not have different effects depending on who gets it; for example, the effect of college on earnings is the same regardless of who or how many people go to college. This strong form of the Stable Unit Value Treatment Assumption (SUTVA) is clearly very hard to satisfy. The people you go to college with can affect your future earnings, and if fewer people go to college, those who do will earn more. In general, the effect of treatment often "spills over" onto other units (both treatment and control cases) and the spillover can be a major obstacle to using the techniques discussed in the rest of this book. However, sometimes changing the question to which a causal answer is sought can help circumvent the problem. For example, if you are worried that a treatment applied to students in a class, say hours of a particular type of instruction, may spill over to other class members, you can change your unit of analysis and consider treatments applied to the class (albeit at the student level). See also Rubin (1986) for an excellent discussion of violations of the SUTVA, and Holland (1986b).

Many estimators also assume the effect of a treatment is the same regardless of who receives it. It seems likely that in most cases, the effect of treatment is heterogeneous, either producing different increases in mean outcomes by group, or producing different gains for each individual. The first is relatively easy, since we can nearly always estimate separately by subgroup and test whether the estimates are the same (using `suest`, for example). Estimating with individual heterogeneity of effects is substantially harder, and requires extremely strong distributional assumptions. Note that this is not the individual heterogeneity of section 1.3.2, which is heterogeneity in the constant term, but heterogeneity in the coefficient on a treatment variable of interest, which makes it very hard to estimate an average effect without strong assumptions (of course, the assumption that everyone

gets the same effect from treatment is also strong).

1.5.6 Bounds

Instead of getting a consistent point estimate of the effect of X on y, or an interval that defines a plausible range for the effect given a large enough dataset, under some plausible assumptions, we could adopt another approach. Often, we can say with a very high degree of confidence that the effect cannot be larger than one estimate and cannot be lower than another, implying that the true effect is inside those bounds.

The accessible exposition by Manski (1995) demonstrates how a causal effect can be bounded under very unrestrictive assumptions, and then the bounds can be narrowed under more restrictive parametric assumptions. Lee (2009) advocates another very useful method of bounding treatment effects, used in Leibbrandt and McCrary (2005).

A related method to describe the sensitivity of estimates to violations of assumptions is described by Rosenbaum (2002), who provides a wealth of detail on formal sensitivity testing for matching methods (see 2.2.1).

Given how sensitive the quasi-experimental methods are to assumptions (selection on observables, exclusion restrictions, exchangeability, or various other assumptions described in subsequent chapters), some kind of sensitivity testing is order no matter what method is used.

Chapter 2

Matching and Reweighting Methods

The basic idea behind matching models is to compare outcomes only for individuals who are otherwise identical, except one gets treatment and one doesn't (or the two get different levels of treatment). In the simplest case, we can make comparisons for groups of individuals who have different treatment status but exactly the same values for every other possible covariate, which is equivalent to a fully saturated regression[1] and is known as exact matching.

In the case of a single discrete set of treatments X^T we wish to compare means or proportions much as we would in an experimental setting. The simplest case is where $y = X^T\beta + X^C\gamma + \varepsilon$ and we are interested in the effect of a dummy treatment variable X^T (observations are either in the treatment or control groups). We may be able to include indicators and interactions for factors (in X^C) that affect selection into the treatment group (defined by $X^T = 1$), to estimate the impact of treatment within groups of identical X^C using a fully saturated regression. But if there are even a few variable with many different possible values, the number of required dummies will quickly make estimation infeasible. With continuous variables, exact matching is typically impossible. Matching cases that are similar is still feasible, but with many observable variables, we are confronted with a "curse of dimensionality" where the number of directions in which to search for a good match makes matching too difficult.

There are also matching estimators (Cochran and Rubin 1973, Stuart and Rubin 2007) which compare observations with like X^C by pairing observations that are close by some metric; see also Imai and van Dyk (2004) for generalizations of matching estimators. A set of alternative approaches involve reweighting so the

[1] A fully saturated regression includes indicator variables for every possible value of each covariate and their interactions, so allows maximal flexibility in the specification.

distributions of X^C are nearly identical for different groups.

We can let similar be defined by some measure of distance between covariate patterns, typically Mahalonobis distance, and match on distance alone, but Rosenbaum and Rubin (1983) suggest that all we need is the probability of treatment. Rosenbaum and Rubin (1983) showed that comparing individuals with like probabilities of treatment (also called the propensity score) is just as good as exact matching, which greatly reduces the dimensionality of matching to just one—the propensity score lies strictly between zero and one, so we just need to match in the unit interval. If we don't know the probability of treatment, we can estimate it conditional on the X variables, and do just as well as if we knew the true probability, or even better (Hirano et al. 2003).

Matching or reweighting approaches can give consistent estimates of a variety of average treatment effects, under assumptions that the selection process depends on observables, and that the model used to match or reweight is a good one. The standard assumption for selection on observables is called the conditional independence assumption, or unconfoundedness, which requires that the potential outcomes are independent of treatment conditional on a set of observable controlvariables X^C; but see Frölich (2007). A weaker assumption that only the potential outcome under no treatment ($X^T = 0$) is independent of treatment conditional on a set of variables X^C allows us to identify the average effect of treatment on the treated (ATT), because we can treat outcomes averaged over some cases that got no treatment as the counterfactual outcome for treated cases. The second crucial assumption is that the propensity score lies strictly between zero and one, called the "overlap" assumption, or sometimes the assumption that the treatment and control groups' distributions of propensity scores have identical supports. The combination of unconfoundedness and overlap is called "strong ignorability" in Rosenbaum and Rubin (1983).

In some sense, these estimators push the problems associated with observational data from estimating the effect of X^T on y down onto estimating the effect of X^C on X^T. For this reason, estimates based on reweighting or matching are unlikely to convince someone unconvinced by standard regression results. Selection on observables is not the type of selection most critics have in mind.

Still, comparing standard regression results and results from matching can provide very good information about the extent of selection bias due to selection on observables, and relaxes the functional form assumption required for regression. If we can believe that the extent and direction of selection bias due to selection on unobservables is similar (due to theory about the process involved), then at least we have a notion about plausible bias.

2.1 Nearest Neighbor Matching

Nearest Neighbor Matching pairs observations in the treatment and control groups and computes the difference in outcome y for each pair, then the mean difference across pairs. The Stata implementation `nnmatch` was described by Abadie et al. (2004). Imbens (2004) covered details of Nearest Neighbor Matching methods, and provided useful examples.

The typical way to pick a match in Nearest Neighbor Matching is to compute a distance between vectors of observable exogenous variables; call these Z and suppose some of them also have a direct impact on outcomes y. We can compute the distance of one observation z_0 from another z_1 as a quadratic form[2] in Z given by $(z_0 - z_1)W(z_0 - z_1)'$ and let W be a function of the standard deviations of each variable, so we measure deviations in standard deviation units. This is called the Mahalanobis distance. Other metrics are also offered in `nnmatch`, and the properties of each discussed in Abadie et al. (2004). Then we pick the control group observation with the minimum distance from a given treatment case as the estimate of a counterfactual outcome for that case, or possibly several observations; see also B.5.5.

The downside to Nearest Neighbor Matching is that it can be computationally intensive, unless some restrictions are imposed on the problem. With many observable variables available to match on, taking on many values, the "curse of dimensionality" makes the problem of finding close matches a very hard one. Also, bootstrap standard errors are infeasible owing to the discontinuous nature of matching (Abadie and Imbens 2006). The approach also does not easily generalize to a continuous treatment variable, or more than one treatment.

2.2 Propensity score matching

Propensity score matching essentially estimates each individual's propensity to receive a binary treatment (via a `probit` or `logit`) as a function of observables and matches individuals with similar propensities. As Rosenbaum and Rubin (1983) showed, if the propensity were known for each case, it would incorporate all the relevant information about selection, and propensity score matching could achieve optimal efficiency and consistency. In practice, the propensity must be estimated, but this is not a serious problem. More importantly, selection is usually not only on observables, so the estimator will be biased and inconsistent.

[2] See A.1.4 for definitions.

Morgan and Harding (2006) provide an excellent overview of practical and theoretical issues in matching, and comparisons of nearest neighbor matching and propensity score matching. Their expositions of different types of propensity score matching, and simulations showing when it performs badly, are particularly helpful. Stuart and Rubin (2007) offer a more formal but equally helpful discussion of best practices in matching.

Typically, one treatment case is matched to several control cases, but one-to-one matching is also common, and may be preferred (Glazerman et al. 2003). One Stata implementation `psmatch2` (Leuven and Sianesi, 2003) is available from SSC (`ssc desc psmatch2`) and has a useful help file (its predecessor `psmatch` is described at http://www.stata.com/meeting/7uk/sianesi.pdf). There is another Stata implementation described by Becker and Ichino (2002) (`findit pscore` in Stata). `psmatch2` will perform one-to-one or match k neighbors (sampling the nearest neighbor or randomly within an allowable range, with or without replacement) using radius, kernel, local linear regression, or Mahalanobis matching (a variety of distance concepts).

Propensity score methods typically assume a common support, i.e. the range of propensities to be treated is the same for treated and control cases, even if the density functions have quite different shapes. In practice, it is rarely the case that the ranges of estimated propensity scores are the same, but they do nearly always overlap, and generalizations about treatment effects are often limited to the smallest connected area of common support.

Often a density estimate below some threshold greater than zero defines the end of common support; see citetHIT1997 for more discussion. This is because the common support is the range where both densities are nonzero, but the estimated propensity scores take on a finite number of values, so the empirical densities will be zero almost everywhere. Generally, we need to use a kernel density estimator like `kdensity` to obtain smooth estimated densities of the propensity score for both treatment and control groups, but then areas of zero density will have positive density estimates. Thus some small value f_0 is redefined to be effectively zero, and the smallest connected range of estimated propensity scores $\widehat{\lambda}$ with $\widehat{f}(\widehat{\lambda}) \geq f_0$ for both treatment and control groups is used in the analysis, and observations outside this range are discarded.

Regardless of whether the estimation or extrapolation of estimates is limited to a range of propensities or ranges of X^C variables, the analyst should present evidence on how the treatment and control groups differ, and which subpopulation is being studied. The standard graph here is an overlay of kernel density estimates of propensity scores for treatment and control groups, easy in Stata with `twoway`

`kdensity`.

The real problem with matching is that the variance of matching estimators is often very high, but the estimated standard errors often do not correctly account for the variance. If you randomly sort your data and match many times, your variance of estimates will often be much larger than standard errors would seem to suggest; however, these replications or bootstrap replications cannot easily be used to better estimate a standard error. Also, it is far from straightforward to incorporate sample weights or other survey design features, though these are easily incorporated in a nearly equivalent propensity score reweighting estimator.

2.2.1 Sensitivity Testing

Matching estimators have perhaps the most detailed literature on formal sensitivity testing. Rosenbaum (2002) provides a comprehensive treatment of formal sensitivity testing. Rosenbaum bounds on treatment effects may be constructed using `psmatch2` and `rbounds`, a user-written command due to DiPrete and Gangl (2004), who compare Rosenbaum bounds in a matching model to IV estimates. `sensatt` by Nannicini (2007) and `mhbounds` by Becker and Caliendo (2007) are additional Stata programs for sensitivity testing in matching models.

Rosenbaum's "gamma method" describes the sensitivity of the estimate in terms of how large an effect unobserved covariates would need to have to negate the test result we observe. For example, in a model using the estimated probabilities of treatment conditional on some control variables or "propensity scores" (discussed in the next chapter), if we reject the null hypothesis of zero effect at the five percent level with a p-value of 0.01, a five-fold change in the p-value would negate our result. The gamma measure Γ Rosenbaum proposes is the bound on the odds ratio of propensity scores for units with the same covariates (but different unobserved covariate) that would produce that required change in the p-value:

$$\frac{1}{\Gamma} < \frac{\widehat{p_i}(X_i)/(1-\widehat{p_i}(X_i))}{\widehat{p_j}(X_j)/(1-\widehat{p_j}(X_j))} < \Gamma$$

for units i and j. A Γ of 2 implies that the odds of assignment to the treatment group would have to double or halve based on some unobservable in order to negate our conclusions; a Γ of 4 implies that the odds of assignment to the treatment group would have to quadruple or quarter based on some unobservable variable. A larger Γ thus implies more robustness against failures of assumptions, and the metric is more or less comparable across a variety of settings. Rosenbaum is in the process of extending this method to other estimation techniques.

2.3 Reweighting

The propensity score can also be used to reweight the control group so the distribution of observables X^C looks the same as in the treatment group. The basic idea is to use a `probit` or `logit` or semiparametric regression of treatment on X^C to estimate the conditional probability $\hat{\lambda}$ for each observation of being in the treatment group and to use the odds $\hat{\lambda}/(1-\hat{\lambda})$ as a weight for observations in the control group. This is like inverting the test of randomization used in experimental designs to make the group status look as if it were randomly assigned.

As Morgan and Harding (2006) point out, all the matching estimators can also be thought of various reweighting schemes whereby treatment and control observations are reweighted to allow causal inference on the difference in means. Note that a treatment case i matched to k cases in an interval, or k nearest neighbors, contributes $y_i - k^{-1}\sum_1^k y_j$ to the estimate of a treatment effect, and one could just as easily rewrite the estimate of a treatment effect as a weighted mean difference. With propensity score weighting, instead of a few observations in the control group serving as the counterfactual for a given treatment case, the whole control group (suitably reweighted) serves as the counterfactual. Busso and McCrary (2009b) show that reweighting estimators often perform better than matching.

Researchers typically ignore the fact that weights are estimated, which may not be a bad idea. It turns out that we can actually do better throwing away information on the true probability of treatment and estimating the probability of treatment. Hirano et al. (2003) show that nonparametric estimation of the probability of treatment can achieve maximal efficiency, but simulations not shown here demonstrate that parametric estimates fare substantially better than using the true probability of treatment, even when the parametric model is using the wrong family of distributions. In the end, it may be the case that estimated standard errors are reasonably conservative already with propensity score reweighting.

The reweighting approach is particularly useful in combining matching-type estimators with other methods, e.g. fixed-effects regression. After constructing weights $w = \hat{\lambda}/(1-\hat{\lambda})$ (or the product of weights $w = w_0\hat{\lambda}/(1-\hat{\lambda})$ where w_0 is an existing weight on the data used in the construction of $\hat{\lambda}$) that equalize the distributions of X^C, other commands can be run on the reweighted data, e.g. `areg` for a fixed-effect estimator.

Note that we have discussed calculating weights only for control group, implying that the weight equals one for observations receiving treatment (e.g. those who completed college), i.e. those having `_tr==1`. It is also advisable, however, to scale weights within the treatment and control groups so that the reweighted pro-

portions are similar to those observed in the original sample. In fact, the reweighting of the control group to resemble the treatment group is only one of several plausible reweighting schemes, and a regression of outcomes on a treatment indicator using this weight can be considered an estimate of the average treatment effect on the treated (ATT).

2.3.1 Alternative weighting schemes

A simple way to estimate probability of treatment (suppose this is measured by a dummy `_tr`) conditional on a group of observable variables `z*` (all their names start with z) is to use the proportion receiving treatment in each distinct combination, like so:

```
egen g=group(z*)
egen _ps=mean(_tr), by(g)
```

but a more common method is to use a parametric model, for example:

```
logit _tr z*
predict _ps if e(sample)
```

If we have estimated the probability of treatment in a variable `_ps`, a command to generate weights can be written

```
g w=cond(_tr,1,_ps/(1-_ps))
```

A rescaled weight to approximately preserve proportions in treatment and control would be

```
su _tr
g w1=cond(_tr,r(mean)/(1-r(mean)),_ps/(1-_ps))
```

noting in passing that multiplying treatment weights by p/(1-p) where p is the proportion of the sample receiving treatment, or multiplying control weights by (1-p)/p, or multiplying treatment weights by p and control weights by (1-p), all produce identical results if weights are themselves rescaled to sum to N (note that Stata internally rescales aweights to sum to N). The weight $\widehat{\lambda}/\left(1-\widehat{\lambda}\right)$ for untreated "control" observations reweights the distribution of observable characteristics included in the `logit` or `probit` model to be like that of the treated group. A weighted regression of outcome on treatment is thus a comparison of means across treatment and control groups, but the control group is reweighted to represent the average outcome that treatment group would have exhibited in the

absence of treatment. That is, every control group observation is contributing to an estimate of the mean counterfactual outcome for all treated observations (rather than specific observations being matched).

An alternative weighting scheme of the form

```
su _tr
g w2=cond(_tr,(1-_ps)/_ps*r(mean)/(1-r(mean)),1)
```

reweights the distribution of observables in the treatment group to be like that of the control group. A comparison of means across (reweighted) treatment and control groups, for example using a weighted regression of an outcome variable on the treatment indicator, is then an estimate of the average treatment effect on the controls (ATC). The treatment group is reweighted to represent the average outcome that control group would have exhibited in the presence of treatment.

One method of computing an estimate of the average treatment effect for the population (ATE) is to take the weighted mean of these two estimates, with the weight attached to the ATT equal to the proportion receiving treatment and the weight attached to the ATC equal to one minus the proportion receiving treatment.

An alternative estimate of the ATE is available. First, note that the outcome under treatment for the whole population, i.e. the mean outcome if every unit received treatment, can be estimated by a weighted mean of outcomes in the treatment group with weights $1/\widehat{\lambda}$ (Brunell and DiNardo 2004). Similarly, the outcome under control for the whole population, i.e. the mean outcome if every unit received no treatment, can be estimated by a weighted mean of outcomes in the control group with weights $1/\left(1-\widehat{\lambda}\right)$. The weights for both groups are given by

```
su _tr
g w3=cond(_tr,1/_ps*r(mean)/(1-r(mean)),1/(1-_ps))
```

and an ATE estimate is then simply a weighted comparison of means (e.g. via a regression). One problem that is exacerbated in this scheme is measurement error in the estimated propensity score: as DiNardo (2002) writes: "small errors in estimating $\rho(x)$ can produce potentially large errors in the weights. Since the weight is a nuisance parameter from the viewpoint of estimating a density or a specific moment of the distribution, this is not a straightforward problem."

A fourth reweighting scheme

```
g w4=cond(_tr,(1-_ps),_ps)
```

minimizes the observable distance between treatment and control groups in the sense that a test statistic for the difference in means (the Hotelling test) is zero

(and the weighted groups are of equal size, so the mean of the treatment indicator is one half), but a difference in means using this weight is not so readily interpreted as an average treatment effect. Nevertheless, simulation evidence not presented here indicates it may be very effective (in the sense of having small bias and MSE) in estimating the average treatment effect, especially when estimated probabilities are near zero or one.[3] It also exhibits good robustness to omitted variables in the selection equation (the first stage `logit` or `probit`). The reason for this, shortly, is that the weights are linear in estimated probabilities so they don't "explode" near zero and one, but the ratio of the weights at each estimated probability is still correct, so the relative weights produce ATE estimates that are fairly robust.

See Lunceford and Davidian (2004) and Busso and McCrary (2009a) for additional discussion of construction of weights and rescaling, including an asymptotically variance-minimizing choice for one class of designs.

2.3.2 Result of reweighting

The results of reweighting are clear in a Hotelling test (help `hotelling`) or an equivalent linear discriminant model (the common test of sample balance in an experiment, where sampling is known to be random, but researchers still want to check they did not get an unlucky imbalance in treatment and control groups). The example in Exhibit 2.3.3[4] using `hotelling` and `regress` give identical F statistics, but the `regress` approach allows relaxing of the assumption of equal variance across groups via the `vce(robust)` option.

Exhibit 2.3.3 *The effect of reweighting on a test of balance.*

```
webuse nlswork, clear
keep if year==77
local x "collgrad age tenure not_smsa c_city south nev_mar"
hotelling `x', by(union)
regress union `x'
```

[3]Note that estimated propensities near zero or one represent a possible violation of the condition required for matching or reweighting that the probability of treatment is bounded away from zero and one. In this case, it is advisable to restrict to a subpopulation in which estimated propensities are never near zero or one and reestimate. The densities of propensities near the zero and one boundaries should be estimated using `kdens`, with boundary correction options, available from SSC.

[4]Note that this extract of the National Longitudinal Survey of Young Women 14-26 years of age in 1968 does not include the sample weights, but in general we would prefer to convolve the weights by multiplying our reweighting factor by the sample weights.

```
regress union `x', vce(robust)
logit union `x'
predict _ps if e(sample)
summarize union if e(sample)
local p=r(mean)
generate w3=cond(union,`p'/_ps/(1-`p'),1/(1-_ps))
hotelling `x' [aw=w], by(union)
regress union `x' [aw=w]
regress union `x' [aw=w], vce(robust)
regress ln_wage union `x' [aw=w3], vce(robust)
```

Note that the F statistic drops from 20 to 0.1 after reweighting (18 to 0.1 using heteroskedasticity-robust statistics), and the weighted means of each individual variable look much closer. The last regression of log(wage) on union using the inverse probability weights based on propensity scores gives an estimate of the effect of union membership on wages, over both union and nonunion workers, suggesting that an individual would earn fourteen percent more in 1977 as a union member than as a nonunion worker, on average. Using weights generated by

```
g w1=cond(union,`p'/(1-`p'),_ps/(1-_ps))
regress ln_wage union `x' [aw=w1], vce(robust)
```

instead gives an estimate of the effect of union membership on wages for union members, suggesting that union members earned fourteen and one half percent more in 1977 than they would have as nonunion workers.

What is not clear from `hotelling` or `regress` is that even if the means of X variables are equal in the reweighted sample, that does not imply that their distributions are similar. As long as treatment status can be inferred from higher moments of the X variables, we have not fully controlled for the observable differences across treatment and control groups. In practice, however, reweighting to make means match seems to make the distributions of observables very similar, for the same reason that matching on the propensity score does.

The difference in distributions is most cleanly observable in the distribution of estimated propensity scores, but can be seen in the individual variables (e.g. `tenure` in the example in 2.3.4; note that `kdens` is available from SSC).

Exhibit 2.3.4 *The effect of reweighting on the distribution of variables.*

```
webuse nlswork, clear
keep if year==77
local x "collgrad age tenure not_smsa c_city south nev_mar"
logit union `x'
predict _ps if e(sample)
kdens _ps if union, bw(.03) ll(0) ul(1) gen(f1 x) nogr
kdens _ps if !union, bw(.03) ll(0) ul(1) gen(f0) at(x) nogr
```

```
label var f0 "pdf of propensities for unweighted control obs"
label var f1 "pdf of propensities for unweighted treatment obs"
line f1 f0 x, leg(col(1)) name(unwtd, replace)
summarize union if e(sample)
local p=r(mean)
generate w3=cond(union,`p'/(1-`p'),_ps/(1-_ps))
kdens _ps if union [aw=w3], bw(.03) ll(0) ul(1) gen(g1 x1) nogr
kdens _ps if !union [aw=w3], bw(.03) ll(0) ul(1) gen(g0) at(x1) nogr
label var g0 "pdf of propensities for reweighted control obs"
label var g1 "pdf of propensities for reweighted treatment obs"
line g1 g0 x1, leg(col(1)) name(rewtd, replace)
kdens tenure if union [aw=w3], bw(1.5) ll(0) gen(td) at(tenure) nogr
label var td "Density for union members"
kdens tenure if !union [aw=w3], bw(1.5) ll(0) gen(cd) at(tenure) nogr
label var cd "Density for C reweighted to resemble T"
line td cd tenure, sort leg(col(1))
```

Matching on the propensity score ensures the distributions of estimated propensity scores are virtually identical in (matched) treatment and control groups, especially if matching models are iterated until balance is achieved, but reweighting does not. For example, if the distribution of some variable (including propensity scores) is bimodal in the control group and single-peaked in the treatment group, those properties will typically still be observable in the reweighted data. Nevertheless, reweighting achieves much of the balancing achievable via matching on the propensity score.

The fourth reweighting scheme achieves the smallest difference in means, with an F statistic as close to zero as is feasible, but the distributions of observable characteristics and estimated propensity scores are very similar under all these approaches to reweighting. The fourth reweighting scheme is less theoretically justifiable, so one of the other choices is preferable, depending on what kind of inferences are sought.

See Iacus et al. (2008) for an alternative matching method that controls, up to specified levels, "for all imbalances in central absolute moments, comoments, coskewness, interactions, nonlinearities, and other multidimensional distributional differences between treated and control groups."

We can also match or reweight when the treatment takes on many values (not just treatment and control) using the generalized propensity score approach of Hirano and Imbens (2004), described by Bia and Mattei (2008). The generalized propensity score $r(t|x)$ is the density of treatment conditional on $X = x$, estimated as $\widehat{r}(t, x)$.

2.3.5 Uses of reweighting

With a propensity-based reweighting approach, we can compare not only mean outcomes, but entire distributions. For example, reweighting is at the heart of the method for comparing distributions proposed by DiNardo et al. (1996). The paradigmatic example of that approach uses two years of data, estimates the probability that an observation is in the first year or the second, then reweights the second year's observations by $\widehat{\lambda}/\left(1-\widehat{\lambda}\right)$ so that the distributions are nearly equivalent across the two years. Changes in means or distributions (of some outcome variables) in the reweighted data are then interpreted as estimates of change had the means of the X variables not changed over time.[5] Firpo (2007) gives an approach for efficient estimation of quantile treatment effects.

A similar method could be applied to estimate the proportion of a wage gap observed across men and women or white and nonwhite workers that are attributable to characteristics, along the lines of the Blinder (1973) and Oaxaca (1973) decomposition method. That decomposition method, described by Jann (2008), attributes observed differences in an outcome y both to differences in exogenous variables X^C and differences in the associations between X^C and y. The general approach is most useful for comparing two distributions where the binary variable defining which group an observation belongs to is properly considered exogenous, e.g. sex or calendar year. The reweighting approach leads to a whole class of weighted least squares estimators, and is connected to the techniques described by DiNardo et al. (1996), Autor et al. (2005), and Machado and Mata (2005). These techniques are also related to various decomposition techniques in Yun (2004; 2005a;b), Gomulka and Stern (1990), Juhn et al. (1993), and Bauer and Sinning (2008) and Sinning et al. (2008). The `dfl`, `jmpierce`, and `oaxaca` programs on SSC are a few user-written Stata implementations for decomposition techniques. The connections among these methods are discussed by e.g. DiNardo (2002) and Lemieux (2002).

The reweighting approach also extends easily to a polychotomous categorical variable, by considering the analogy to the DiNardo et al. (1996) approach applied to multiple years. For example, each subsequent year's data can be reweighted to have observable characteristics similar to the first year, or each year can be reweighted to match some other base year's distribution. In the same way, observations receiving various levels of treatment can be reweighted to match some base category. Of course, the choice of base category may affect the interpretation

[5] See also Altonji et al. (2008) for a paper dealing not only with differences in distributions in samples, but sample attrition and missing values.

of results. Extensions of the reweighting approach to the case of a continuous treatment are also possible using the generalized propensity score approach of citetHiranoImbens2004, described by Bia and Mattei (2008).

2.4 Examples

Imagine the outcome is wage and the treatment variable is union membership. One can imagine reweighting union members to have distributions of education, age, race/ethnicity, and other job and demographic characteristics equivalent to nonunion workers (or a subset of nonunion workers). One could compare otherwise identical persons within occupation and industry cells using a regression approach or `nnmatch` with exact matching on some characteristics.

An example comparing several regressions with propensity score matching is given in Exhibit 2.4.1 where the estimated union wage premium is about 13% in a regression, but about 15% in the matching estimate of the average benefit to union workers (the ATT) and about 10% on average for everyone (the ATE). The reweighted regressions give different estimates: for the more than 70% of individuals who are unlikely to be unionized (propensity under 30%), the wage premium is about 9%, and for the full sample, it is about 18%.

Exhibit 2.4.1 *An example of reweighting and matching.*

```
webuse nlswork, clear
xi i.race i.ind i.occ
loc x "union coll age ten not_s c_city south nev_m _I*"
reg ln_w union
reg ln_w `x'
g u=uniform()
sort u
psmatch2 `x', out(ln_w) ate
g w=cond(_tr,1,_ps/(1-_ps))
reg ln_w `x' [pw=w] if _ps<.3
reg ln_w `x' [pw=w]
```

Arguably none of these estimates of wage premia correspond to a readily specified thought experiment, such as "what is the estimated effect on wages of being in a union for a randomly chosen individual?" (the ATE) or "what is the estimated effect on wages of being in a union for an individual just on the margin of being in a union or not?" (the LATE). DiNardo and Lee (2002) offer a much more convincing set of causal estimates of the LATE using an Regression Discontinuity design.

We could also have estimated the wage premium of a college education by simply putting `coll` in the place of `union` (to find a wage premium of 25% in a regression or 27% using `psmatch2`).

We could also use data from Card (1995) on education and wages:

Exhibit 2.4.2 *Another example of matching.*

```
use http://pped.org/card, clear
g byte coll=educ>15
loc x "coll age exper* smsa* south mar black reg662-reg669"
reg lwage `x'
psmatch2 `x', out(lwage) ate
```

to find a college wage premium of about 29% using a regression or about 30% using `psmatch2`. We return to this example in the next chapter using an Instrumental Variables method.

Chapter 3

Panel Methods

If an omitted variable can be measured, or proxied by another variable, then an ordinary regression may yield an unbiased estimate. In the case where ordinary least squares is unbiased, or at least consistent, it would nearly always be preferred, since it produces the most efficient estimates (ignoring issues around weights or errors that are not i.i.d.). The measurement error entailed in a proxy for an unobservable could actually exacerbate bias, rather than reducing it, however. In addition, one is usually concerned that cases with differing X^T may differ in other ways even conditional on all other observables X^C ("control" variables). Nonetheless, a sequence of ordinary regressions that add or drop variables can be instructive as to the nature of various forms of omitted variable bias in the data at hand.

The idea of panel methods is to use individuals as their own controls, for example by observing units before and after they get treatment. A complete discussion of panel methods would not fit in any one book, but the [XT] manual contains a panoply of estimation options and references for further reading.

The basic idea can be usefully illuminated with one short example using linear regression. Suppose we have a model of the form

$$y = \beta_0 + X^T\beta_T + X^U\beta_U + \varepsilon$$

where we do not observe X^U. Suppose the omitted variables X^U vary only across groups, where group membership is indexed by i, so a representative observation can be written:

$$y_{it} = \beta_0 + X^T_{it}\beta_T + u_i + \varepsilon_{it}$$

where $u_i = X^U_i\beta_U$. Then we can eliminate the bias arising from omission of X^U

by differencing:

$$y_{it} - y_{is} = (X_{it}^T - X_{is}^T)\beta_T + (\varepsilon_{it} - \varepsilon_{is})$$

using various definitions of s.

The idea is to use an individual i as its own control group by including information from multiple points in time t. In fact, the second dimension of the data indexed by t need not be time, but it is a convenient viewpoint.

3.1 Fixed Effects (FE), First Difference (FD), and Long Difference (LD)

A fixed-effect (FE) model such as `xtreg, fe i(id)` or `areg, a(id)` effectively subtracts off the within-i mean values of each variable, so for example $X_{is}^T = \bar{X^T}_i = \frac{1}{N_i}\sum_{s=1}^{N_i} X_{is}^T$, and the model

$$y_{it} - \bar{y}_i = (X_{it}^T - \bar{X^T}_i)\beta_T + (\varepsilon_{it} - \bar{\varepsilon}_i)$$

can be estimated via OLS. This is also called the "within estimator" and is equivalent to a regression that includes an indicator variable for each group i, allowing for a different intercept term for each group.

An alternative to the FE model is to use the first difference (FD), i.e. $s = (t-1)$ or

$$y_{it} - y_{i(t-1)} = (X_{it}^T - X_{i(t-1)}^T)\beta_T + (\varepsilon_{it} - \varepsilon_{i(t-1)})$$

which is just `reg d.y d.x` in `tsset` data, or `xtivreg2 y x, fd` (Schaffer and Stillman 2007), which offers additional standard error (SE) corrections beyond `cluster` and `robust`.

A third option is to use the long difference (LD), keeping only two observations per group. For a balanced panel, if $t = b$ is the last observation and $t = a$ is the first, the model is:

$$y_{ib} - y_{ia} = (X_{ib}^T - X_{ia}^T)\beta_T + (\varepsilon_{ib} - \varepsilon_{ia})$$

producing only one observation per group (the difference of the first and last observations).

Figure 3.1 shows the interpretation of these three types of estimates by showing a single panel's contribution to the estimated effect of an indicator variable

that is one for all $t > 3$ (t in $0, ..., 10$) and zero elsewhere, e.g. a policy that comes into effect at some point in time. The FE estimate compares the mean outcomes before and after; the FD estimate compares the outcome just prior to and just after the change in policy; and the LD estimate compares outcomes well before and well after the change in policy.

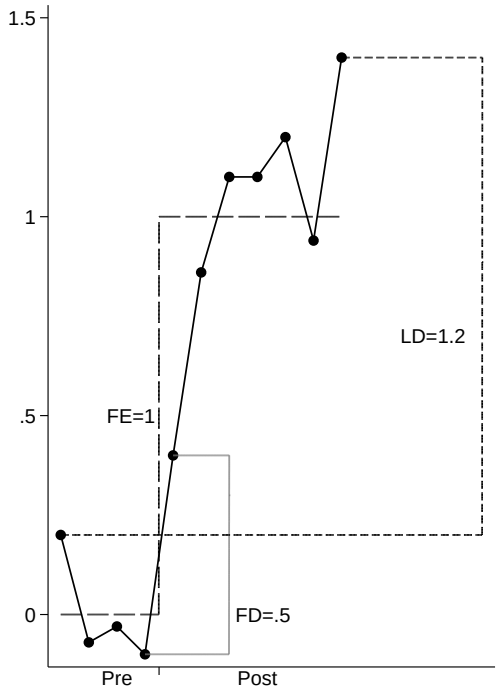

Figure 3.1: A single panel's contributions to various estimates

Clearly, different assumptions about the error process apply in each case, in addition to assumptions about the speed with which X^T affects y, and the FD and LD models require an ordered t index (such as time). The `cluster` option used above should be considered nearly *de rigeur* in panel models, to allow for errors that may be correlated within group, and not identically distributed across groups. The performance of the cluster-robust estimator is quite good with 50 or more clusters, or fewer if the clusters are large and balanced (Nichols and Schaffer 2007). In the LD case, the `cluster` option is equivalent to the `robust` option, since each group is represented by one observation.

A pair of mechanical examples (i.e. these are not really sensible causal models, but merely illustrative of the approaches in small publicly available datasets) of code for all three estimators (FE, FD, and LD) is given in Exhibit 3.1.1.

Exhibit 3.1.1 *Sample code for fixed effects, first differences, and long differences.*

```
webuse grunfeld, clear
xtreg invest kstock, fe cl(com)
est sto FE
reg d.invest d.kstock, cl(com)
est sto FD
su time, meanonly
gen t=time==r(max) if time==r(min) | time==r(max)
tsset com t
g dks=d.kstock if t<.
g dinv=d.invest if t<.
reg dinv dks, cl(com)
est sto LD

use http://fmwww.bc.edu/ec-p/data/macro/abdata.dta, clear
xtreg ys k n, fe cl(id)
est sto FE
reg d.ys d.k d.n, cl(id)
est sto FD1
xtivreg2 ys k n, fd cl(id) small
est sto FD2
su year, meanonly
gen t=year==r(max) if year==r(min) | year==r(max)
tsset id t
g dys=d.ys if t<.
g dk=d.k if t<.
g dn=d.n if t<.
reg dys dk dn, cl(id)
est sto LD
esttab *, nogaps
```

To choose among FE/FD/LD models, one must impose some assumptions on the speed with which X^T affects y, or have some evidence as to the right time frame for estimation. This type of choice comes up frequently when stock prices are supposed to have adjusted to some news, especially given the frequency of data available; economists believe the new information is capitalized in prices, but not instantaneously. Taking a difference in stock prices between 3:01pm and 3pm is inappropriate, but taking a long difference over a year is clearly inappropriate as well, since new information arrives continuously.

One common pitfall in first-difference models is endogeneity of timing in changes in the treatment variable X. For example, if we measure the impact of a job training program on earnings, we may see a large response comparing earnings just after training to just before, even if training has no real impact on

earnings. Those who partake of training are more likely to be those who had a substantial drop in earnings relative to their long-run average, and may experience a bump above their long-run average earnings due to a placebo motivational effect after training, but if we compare earnings a few periods before to a few periods after training, there might well be no difference. The common feature of a dip in outcomes just before treatment is known as Ashefelter's dip after Ashenfelter (1978).

In many panel models, one must think carefully about within-panel trends and the frequency of measurement. On within-panel trends, note that we cannot usually obtain consistent estimates of within-panel trends for the same reason we cannot usually obtain consistent estimates of fixed effects; the number of parameters increases linearly in the number of panels N, and we usually think of asymptotic results justifying our estimates using samples with increasing number of panels N (and fixed panel length). But we can treat the fixed effects and within-panel trends as nuisance parameters, and in the standard linear model, estimating these terms does not bias our coefficient estimates for the variables of interest.

On the frequency of measurement, we can use various filtering techniques to get different frequency "signals" from noisy data. A very simple method used in Baker et al. (1999) is often attractive, as it offers a easy way to decompose any variable X_t into two orthogonal components: a high-frequency component $(X_t - X_{t-1})/2$ and a low-frequency component $(X_t + X_{t-1})/2$ that together sum to X_t. More sophisticated filters are often discussed in the time-series literature, for example the Kalman filter and other methods discussed in [TS] arima.

3.2 Diff-in-Diff

Having eliminated bias due to unobservable heterogeneity across units indexed by i by differencing, it is often tempting to difference again, leading to a multidimensional fixed-effects model.

A very simple version of a multidimensional fixed-effects model is the difference-in-difference model, or diff-in-diff, and the simplest form of that model has only two categories for each dimension. For example, if some states in the US experienced a policy shift in one year and some did not, we can compare outcomes before and after for both groups of states, shown in the table in Exhibit 3.2.1 with mean outcomes for states that experienced a policy shift in the treatment group, labeled T. In general, we think of measuring mean outcome before and after the start of treatment, so $X = 0$ for group T in period Pre and $X = 1$ for group T in

period Post, whereas $X = 0$ for group C in period Pre and $X = 0$ for group C in period Post.

Exhibit 3.2.1 *Mean outcomes in the simplest diff-in-diff design.*

	Pre	*Post*
T	*y1*	*y2*
C	*y3*	*y4*

The cross-sectional OLS estimator is $y2 - y4$ and the "interrupted time series" or "pre-post" estimator is $y2 - y1$, but these have very low internal validity (we suspect they are badly biased in general, if there is a persistent group difference and a fixed time period effect or "common shock"). The difference-in-difference estimator is $(y2 - y4) - (y1 - y3)$ viewing the pre-treatment difference in means as an estimate of baseline differences, or $(y2 - y1) - (y4 - y3)$ viewing the change over time in means for the control group as an estimate of the counterfactual change over time in means for the treatment group (had treatment not occurred). The two estimates are identical, and are equivalent to a fixed effects estimator where an indicator for T is included for the group fixed effect, and an indicator for Post is included for the time fixed effect, and the interaction term $T \times Post$ is included to measure the difference in differences.[1] See Athey and Imbens (2006) for a generalization of the difference-in-difference model that identifies the entire distribution of effects.

3.3 More Fixed Effects Models

A more general strategy is to include different sets of exhaustive indicators to difference out fixed effect in multiple dimensions. For example, it is common to include indicator variables for time t in fixed-effects models, as shown in example code in Exhibit 3.3.1 (simpler in Stata 11 using factor variables), thus creating a two-way fixed-effects model.

Exhibit 3.3.1 *Including indicator variables for an additional dimension of FE.*

[1] Note that the simple interpretation for the coefficient on an interaction term does not apply in nonlinear models; see Ai and Norton (2003) and Norton et al. (2004) on interpretation of interactions in `logit` and `probit`.

```
webuse grunfeld, clear
qui tab year, gen(d)
drop d1
xtreg inv ks d*, fe cl(com)
```

If individuals i are observed in different settings j, for example students who attend various schools or workers who reside in various locales over time, we can also include indicator variables for j in a fixed-effects model. Thus we can consider various n-way fixed-effects models, though models with large numbers of dimensions for fixed effects may rapidly become unstable or computationally challenging to estimate (see e.g. `felsdvreg` and other packages on SSC for estimating multidimensional fixed effects).

3.4 Random Effects (RE)

The LD, FD, and FE estimators use none of the cross-sectional differences across groups (individuals) i, which can lead to lower efficiency (relative to an estimator that exploits cross-sectional variation), and they also drop any variables that do not vary over t within i, so the coefficients on these variables (which may be of some intrinsic interest) cannot be estimated with these methods.

The random-effects estimator (RE) available from `xtreg` exploits cross-sectional variation and reports coefficients on variables that do not vary over t within i, but requires strong assumptions about error terms that are often violated in practice. In particular, for RE to be unbiased in situations where FE is unbiased, we must assume that u_{it} is uncorrelated with ${X^T}_{it}$ (which contradicts our starting point above, where we worried about a X^U correlated with X^T). There is no direct test of this assumption about an unobservable disturbance term, but the Hausman test (`hausman`) is a test of the null that the coefficients estimated in both the RE and FE models are the same. A better alternative is the user-written command `xtoverid` on SSC (by Mark Schaffer and Stephen Stillman) which has the same null but offers standard errors robust to heteroskedasticity or clustering. In Exhibit 3.4.1, a rejection casts doubt on whether RE is consistent when FE is, so we should switch to fixed effects.

Exhibit 3.4.1 *A generalized Hausman test for RE versus FE.*

```
webuse grunfeld, clear
egen ik=max(ks*(year==1935)), by(com)
xtreg inv ks ik, re cl(com)
xtoverid
```

An alternative approach that assumes all of the observable variables are uncorrelated with e but may be correlated with u includes the estimators of Hausman and Taylor (1981) and Amemiya and MaCurdy (1986), both of which use instrumental variables and are estimated by the `xthtaylor` command in Stata. The estimators implemented in `xthtaylor` offer the possibility to identify the impact of a time-invariant variable on y, but use an instrumental variables (IV) estimator, so they are subject to many of the caveats in chapter 4 on those models.

Various other panel methods use further assumptions about the distributions of errors or other effects to get more efficient consistent estimates. Mixed models, also called multilevel models, or hierarchical linear models, estimates by `xtmixed` or `gllamm` (on SSC) assume that there are a variety of fixed and random effects, so they are a bit like random effects on steroids. Typically, there is a tradeoff between improved efficiency bought by making assumptions about the data-generating process versus robustness to various violations of assumptions. For example, `xtmixed` can estimate a model like

$$y_i t = X_{it} b_i + e_{it}$$

and give estimates for the mean effect of X on y only under very restrictive assumptions; more generally an instrumental variables estimator is needed to identify he mean effect of X on y in such a model. One common application of mixed models to panel data is known as "growth curve" modeling, in which individual-specific slopes are assumed to be normally distributed between measurement points so that growth of y is piecewise linear over intervals defined by measurement frequency. In this specification, time variables are the main explanatory variables and many covariates may be interacted with time variables so that they are assumed to affect the rate of growth in y rather than the level of y directly.

3.5 Measurement Error in Panel Models

Griliches and Hausman (1986) point out that measurement error which has a larger variance within unit than across units may be a good reason to prefer a pooled specification to any other panel method. They also show that a long-difference estimator has many desirable robustness properties. Measurement error is a major headache in any regression model, but moreso in a panel regression, particularly if there is reason to think that measurement error is correlated across variables. For example, if log hours worked are regressed on log wage in an attempt to measure

labor supply elasticities, and wages are computed from the ratio of reported earnings and reported hours, any measurement error in hours will contaminate both y and X in a predictable way, causing "division bias" (Borjas 1980). In a panel regression which throws away the cross-sectional variation and uses only within-panel variation, it is possible that most of the variation purportedly identifying the impact of X on y is actually due to correlated measurement error!

3.6 Dynamic Panel Models

Panel models that include lags of the outcome variable y as explanatory variables and contain unobserved fixed (or random) effects u_i are called dynamic panel models. The u_i are correlated with the lagged y, because u_i in one period affects the level of y in that period, which affects the level of y in a later period. The correlation makes standard estimators inconsistent, even for the linear model. Arellano and Bond (1991) gave a consistent GMM estimator and Blundell and Bond (1998) gave a superior system estimator in the same vein, designed for datasets with many panels (large N, or many units i) and few periods (small T, or few observations per panel i). The standard Stata implementations are `xtabond` and `xtdpd` (or `xtdpdsys`)[2] respectively, but see also Roodman (2009a) and Bruno (2005). To use this approach, we have to assume no autocorrelation in the idiosyncratic errors e_{it} and the initial condition that the u_i are uncorrelated with the first observed first difference of the outcome variable $y_{i2} - y_{i1}$.

Note that another class of models that include lags of the outcome variable y as explanatory variables, where the lags are with respect to distance rather than time, are called "spatial models" (or "network" models if distance is discrete) and suffer from similar estimation difficulties. Lee (2004) describes estimation of a model that allows for spatially correlated errors, but does not include a spatially lagged dependent variable. A GMM approach similar to that taken in the dynamic panel model case can give consistent estimates in a wide variety of these models (Kelejian and Prucha 1999; 2004, Kapoor et al. 2007, Kelejian and Prucha 2008, Badinger and Egger 2009, Lee 2007, Lee and Liu 2009, Drukker 2008).

[2]Note that the GMM two-step estimator for standard errors is badly biased, but `xtdpd` implements the bias-corrected estimator of Windmeijer (2005).

3.7 Nonlinear Panel Models

You can't generally get consistent estimates in a nonlinear model with many panels i and few observations T per panel by manually including fixed effects via indicator variables for individuals, though the bias is often overstated (simulations often show the bias with even $T = 30$ or more to be negligible, depending on the model estimated and the data). It is often the case that estimates using a linear panel regression to approximately estimate marginal effects for a nonlinear model are close enough to be qualitatively indistinguishable from a correct model, but see Chernozhukov et al. (2008).

The `xtlogit` and `xtprobit` commands will estimate a model

$$Pr\left(y_{it}|X_{it}, u_i\right) = G\left(X_{it}b + u_i\right)$$

where the panel effects u_i are assumed uncorrelated with the X_{it}, or

$$y_{it} = I\left(X_{it}b + u_i + e_{it} > 0\right)$$

writing e_{it} for the idiosyncratic error and $I()$ for the indicator function. The panel effects u_i can be assumed to be random effects in `xtprobit` and `xtlogit`, or fixed effects in the case of `xtlogit` and `clogit` where the observed number of $y = 1$ outcomes is a sufficient statistic for the level of u_i in the "conditional fixed effects" model. However, we would generally like to allow some unknown form of dependence between the panel effects and treatment variables.

Various methods to estimate nonlinear models assuming fixed effects u_i potentially correlated with X require "strict exogeneity" (Chamberlain 1984), so that the distribution of the outcome at one point y_{it} given all realizations of X and u_i is the same as the distribution of y_{it} given the current X_{it} and u_i. This means X at one point in time has no effect on *future* realizations of the idiosyncratic error e, so for example an increase in treatment at an earlier period cannot influence other unobserved factors that vary over time and affect otucomes.

Wooldridge (2005b) discusses a strategy to estimate the average marginal effects in a fixed-effects probit assuming the distribution of panel effects u_i is normal and integrating them out using the standard command `xtreg` with the `re` option (and some extra variables included as explanatory variables), which compares favorably with related methods (Arulampalam and Stewart 2009). Papke and Wooldridge (2008) discuss a class of GLM strategies to estimate average marginal effects. In general, it is much harder to consistently estimate the coefficients b than the average marginal effects $\frac{\partial Pr(y_{it}=1|X_{it},u_i)}{\partial X}$.

The Correlated Random Effects framework allows dependence between u_i and X_i,if the dependence is restricted in some way. One approach due to Chamberlain (1980; 1984) is to assume a parametric family of distributions for the distribution of u_i given X, for example,

$$u_i = X\gamma + e \quad e \sim N(0, v)$$

which is clearly a restrictive assumption. One also can instead assume that the distribution of panel effects depends only on the panel-specific means of X (not on within-panel variation) without arbitrarily restricting the distribution of u_i given X (Altonji and Matzkin 2005). See also Graham and Powell (2009).

3.8 Falsification Tests

Often, the variation over time (or space) within panel in X is naturally occurring, and is taken to represent a "natural experiment" (for example, when a scientific advance reaches some parts of the world sooner than others, or a policy is rolled out in some place sooner than others, and we take the timing of the diffusion to be essentially random). In this case, some kind of "falsification test" or "placebo test" is usually called for.

A falsification test regresses on X an outcome that should not, or cannot, be caused by the naturally occurring variation in X, and a significant coefficient is taken as evidence of nonrandom variation in X. This is a bit like regressing pre-treatment characteristics on treatment status in an experiment to assess whether randomization failed somehow. Rothstein (2007a) offers a useful applied examination of identifying assumptions in FE models and correlated random effects models. He estimates the "effect" of a later class assignment (think of this as a teacher effect) on earlier test score gains, which cannot have a causal interpretation, and finds similar-sized effects as in a regression of test scores on teacher fixed effects. This calls into question a vast literature on estimating teacher fixed effects, and has spawned a search for better estimation strategies for teacher effectiveness (and undercut the political will for high-stakes tenure decisions based on "value-added" models of teacher effectiveness).

This type of placebo test regresses the actual outcome y on an X that should not have a causal effect, and a significant coefficient is taken as evidence of non-random variation in y that could be due to an unobserved factor that causes variation in X and Z and y so we cannot yet ascribe a causal role to X. This is analogous to the experimental model where we do not ascribe a causal role to the

sugar in a sugar pill, and if we find that a sugar pill and a drug have the same effect, we conclude the drug has no effect over and above the pure effect of convincing someone they are taking a pill.

An alternative placebo test regresses the actual outcome y on a fake X, either via reassigning X values across units i or by generating random X vectors for each i from a distribution chosen to look like the real X process. Here the finding of significant coefficients when regressing actual outcome y on pure noise X is taken to indicate inconsistent standard errors, or small sample failings of asymptotic approximations.

3.9 Segue

Generally, panel methods eliminate the bias due to some unobserved factors and not others. Considering the FE, FD, and LD models, it is often hard to believe that all of the selection on unobservables is due to time-invariant factors. If we don't believe that all of the selection is on observables (suggesting a regression, matching, or reweighting model) or that all of the selection on unobservables is due to time-invariant factors (suggesting a panel model), we can turn to Instrumental Variables (IV) methods.

Chapter 4

Instrumental Variables Methods

An alternative to panel methods and matching estimators is to find another set of variables Z correlated with X^T but not correlated with the error term, e.g. e in

$$y = X^T\beta_T + X^C\beta_C + e$$

so Z must satisfy $E(Z'e) = 0$ and $E(Z'X^T) \neq 0$. The variables Z are called **excluded instruments**, and one of a large class of Instrumental Variables (IV) methods can then be used to consistently estimate an impact of X^T on y. The basic idea is, loosely, to get the component of the outcome y along the portion of X that is exogenous, which is the component of X along Z. Since we are "throwing away" all the "bad" endogenous variation in X we also sacrifice efficiency, and IV estimators have much greater variance than standard regression techniques.

Perhaps the earliest and most famous example of the method at the heart of IV is the investigation of a cholera outbreak in London in 1853 to 1854 undertaken by Snow (1855); see Freedman (1991) and Deaton (1997) for more. Snow mapped cholera deaths (Figure 4.1) and collected data on which water company supplied each household in most London streets and found that the choice of water company by households seemed essentially random, with pipes from each company in each street. The intake pipes for one companies drew water from the Thames below the main sewage outflow and another from upriver; cholera deaths were more than eight times likelier among those supplied by the first company. The treatment is contaminated water, and the water company supplying a household is the excluded instrument. Even if Snow had direct measurements on water quality, there are many factors affecting water quality which are not randomly assigned; the water company can supply the needed random variation in treatment (in fact,

Snow's analysis was the "reduced form" association of the outcome and the excluded instrument).

Angrist and Krueger (2001) cite the Wright (1928) analysis of agricultural supply and demand as the first modern development of instrumental variables. Recent famous examples include using compulsory schooling laws to estimate the effect of education on earnings, where the requirement to stay in school until a certain age induces some individuals to get a little more education than they otherwise would (Angrist and Krueger 1991, Bound et al. 1995), or using number of streams running through a city to discern the effect of more choice over school districts on student outcomes (Rothstein 2007b, Hoxby 2007). The estimation of the effect of education on wages by Card (1995) using proximity to colleges at an earlier age is a running example in the following discussion.

Because IV can lead one badly astray if any of the assumptions is violated, anyone using an IV estimator should conduct and report tests of:

1. instrument validity (overidentification or **overid** tests),
2. endogeneity,
3. identification,
4. presence of weak instruments,
5. and misspecification of functional form (RESET).

Further discussion and suggestions on what to do when a test is failed appear in the relevant sections below.

The instrumental variables estimators are in general only as good as the excluded instruments used, so naturally criticisms of the predictors in a standard regression model become criticisms of the excluded instruments in an IV model.

The IV estimators are still biased, but consistent, and are much less efficient than OLS, so failure to reject the null should not be taken as acceptance of the alternative. That is, one should never compare the IV estimate only to a zero effect, but also to other plausible values, including the OLS estimate. Some other common pitfalls discussed below include improper exclusion restrictions (checked via overidentification tests) and weak identification (checked via various diagnostics and followed up by robust inference); see also Murray (2006).

It is worth repeating that IV estimators are biased in finite samples, and are only justified for very large samples. Nelson and Startz (1990) showed how strange the finite sample behavior of an IV estimator can be, and Bound et al.

Figure 4.1: Snow's map of cholera deaths

(1995) argued that even samples of millions of observations can be insufficient for asymptotic justifications to apply in the presence of weak instruments; see also Stock and Yogo (2005) and Cruz and Moreira (2005).

4.1 Interpretation of estimated coefficients

Various interpretations of the IV estimate have been advanced, typically as the Local Average Treatment Effect, or **LATE**, meaning the effect of X^T on y for those who are induced by their level of Z to have higher X^T. If the treatment effect is the same across the whole population, then the LATE and ATE are the same, and we consistently estimate the effect of X^T on y with IV; if not, we may be required to consider a "random coefficient" model where individuals not only have distinct y_i and X_i^T but distinct coefficients b_i as well. IV can also estimate average treatment effects in the random coefficient model, but only under additional assumptions does IV still identify an average treatment effect of interest (Wooldridge 1997, Heckman and Vytlacil 1998, Wooldridge 2003).

The LATE interpretation is particularly straightforward for a single endogenous indicator variable indicating treatment status (Angrist et al. 1996). For the college graduate example, this might be the average gain $E_i[y_i(t) - y_i(0)]$ over all those i in the treatment group with $Z = 1$ (where Z might be "lived close to a college" or "received a Pell grant"), arising from an increase from $X^T = 0$ to $X^T = t$ in treatment, i.e. the wage premium due to college averaged over those who were induced to go to college by Z.

For example, Card (1995) estimates the wage premium due to an additional year of education for who were induced to go to college by growing up near a college, which effectively lowers the cost of going to college (variation in the effective cost is a good source of exogenous variation, as argued in C). The interpretation of the IV coefficient shown in Exhibit 4.1.1 for an individual is thus the percentage difference in mean earnings for an individual given the 16 years of education they got, less the 15 years they might have gotten had they not grown up near a college. The IV coefficient averages across individuals induced to get more education by this Z, to get the LATE of an extra year of education, implying the return to an additional year of education is about 18 percent and lies in the interval 7 to 29 percent (which includes the OLS estimate).

Exhibit 4.1.1 *The Card (1995) data, renamed to match notation.*

```
use http://pped.org/card, clear
loc xc "exper* smsa* south black reg662-reg669"
```

```
ren lwage y
loc z "nearc2 nearc4"
ren educ xt
regress y xt `xc' [pw=weight]
est sto OLS
ivreg2 y `xc' (xt=`z') [pw=weight], first
est sto IV
esttab OLS IV, drop(s* reg*) mti nonum nogap nostar nonote
--------------------------------------
                      OLS            IV
--------------------------------------
xt                 0.0753         0.182
                  (18.80)        (3.24)
exper              0.0928         0.129
                  (12.02)        (6.23)
expersq          -0.00253      -0.00225
                  (-6.89)        (-4.72)
black              -0.210         -0.128
                  (-9.23)        (-2.57)
_cons                4.566          2.792
                  (57.15)        (2.99)
--------------------------------------
N                    3010           3010
--------------------------------------
```

As Kling (2001) points out, the increase in education arising from Z seems to occur not only in the teens of years of education but in the high-school years as well, which presents some trouble for the theory. Why should people be induced to finish eleventh grade by the proximity of a college, when they will not finish high school? It is possible, of course, that people intend to go on to college because one is nearby, but life intervenes in some way. A more serious critique is that colleges are not sprinkled randomly around the country, and people are not sprinkled randomly at various distances from colleges. For example, the children of college professors are much more likely to grow up near a college.

4.2 IV for Experiments

Perhaps the best-case scenario for IV is where treatment group status has been randomly assigned, but there is some modest failure of the experimental protocol, so not every case gets the level of treatment that was randomly assigned to it. In that case, the assignment can be used as an excluded instrument for the endogenous level of treatment actually received. The interpretation of a coefficient regressing outcomes on random assignment is the effect of "intention to treat" rather than treatment, whereas the IV model gives an interpretation of the (local) average treatment effect for those individuals induced to get treatment by the

random assignment to a treatment group.

4.3 Forms of IV

The standard IV estimator in a model

$$y = X^T\beta_T + X^C\beta_C + e$$

where we have Z satisfying $E(Z'e) = 0$ and $E(Z'X^T) \neq 0$ is

$$\hat{\beta}^{IV} = \begin{pmatrix} \hat{\beta}_T^{IV} \\ \\ \hat{\beta}_C^{IV} \end{pmatrix} = (X'P_ZX)^{-1}X'P_Zy$$

(ignoring weights), where $X = (X^T X^C)$ and P_Z is the projection matrix $Z_a(Z_a'Z_a)^{-1}Z_a'$ with $Z_a = (ZX^C)$. The idea is, we use the component of X^T along Z, which is exogenous, as the only source of variation in X^T that we use to estimate the effect on y.

These estimates are easily obtained in Stata 6 through 9 with the syntax `ivreg y xc* (xt* = z*)`, where `xc*` are all exogenous "included instruments" X^C and `xt*` are endogenous variables X^T. In Stata 10 and higher, the syntax is `ivregress 2sls y xc* (xt* = z*)`. For Stata 9 and above, the `ivreg2` command on SSC would be written `ivreg2 y xc* (xt* = z*)` or as in Exhibit 4.1.1.

The standard IV estimator is equivalent to several forms of two-stage estimators. The first, which gave rise to the moniker **Two-Stage Least Squares** (2SLS), has you regress X^T on X^C and Z and predict $\hat{X}^T$, then regress y on $\hat{X}^T$ and X^C. The coefficient on $\hat{X}^T$ is $\hat{\beta}_T^{IV}$.

Exhibit 4.3.1 *A manual two-step estimate using the Card (1995) data.*

```
foreach x of varlist xt {
 reg `x' `xc' `z' [pw=weight]
 predict double `x'_hat
 }
reg y *_hat `xc' [pw=weight]
est sto TS1
esttab IV TS1, b(%12.8f) drop(s* reg*) mti nonum nogap nostar nonote order(xt xt_hat)
--------------------------------------
                      IV          TS1
--------------------------------------
xt             0.18206326
```

```
                    (3.24)
xt_hat                          0.18206326
                                    (3.44)
exper            0.12940770     0.12940770
                    (6.23)         (6.61)
expersq         -0.00224836    -0.00224836
                   (-4.72)        (-5.48)
black           -0.12752139    -0.12752139
                   (-2.57)        (-2.76)
_cons            2.79195292     2.79195292
                    (2.99)         (3.18)
---------------------------------------------
N                     3010           3010
---------------------------------------------
```

The two-stage version gives the same estimates as the standard IV commands, but the reported standard errors are wrong, as Stata will use $\hat{X}^T$ rather than X^T to compute them. Even though IV is not implemented in these two stages, the conceptual model of these first-stage and second-stage regressions is pervasive, and the properties of the "first-stage regressions" are central to the section on identification and weak instruments below.

The second two-stage estimator that generates identical estimates is often called a **control-function approach**. Regress each variable in X^T on the other variables in X^T and X^C and Z to predict the errors $\hat{v}_T = X^T - \hat{X}^T$, then regress y on X^T, $\hat{v}_T$, and X^C, and you will find that the coefficient on X^T is $\hat{\beta}_T^{IV}$, and tests of significance on each $\hat{v}_T$ are tests of endogeneity of each X^T. With multiple treatment variables `xt*` we can use the code in Exhibit 4.3.2.

Exhibit 4.3.2 *Code for the manual two-stage control-function approach.*

```
cap drop *_hat
unab xt : xt*
foreach v of loc xt {
 loc otht: list xt-v
 reg `v' xc* z* `otht'
 predict v_`xt', resid
 }
reg y xt* xc* v_*
```

In the Card data, the code in Exhibit 4.3.4 will give the IV estimates, though again the standard errors will be wrong. However, the tests of endogeneity (given by the reported p-values on variables `v_*` above) will be correct. A similar approach works for nonlinear models such as `probit` or `poisson` (`help ivprobit` and `findit ivpois` for relevant commands). The tests of endogeneity in nonlinear models given by the control-function approach are also quite robust (see e.g. Wooldridge 2002; p.474,665).

Exhibit 4.3.3 *Code for the manual two-stage control-function approach in the Card data.*

```
foreach x in xt {
 reg `x' `xc' `z' [pw=weight]
 predict double v_`x', resid
 }
reg y xt `xc' v_* [pw=weight]
est sto TS2
esttab IV TS1 TS2, b(%12.8f) drop(s* reg*) mti nonum nogap nostar nonote
------------------------------------------------
                      IV          TS1          TS2
------------------------------------------------
xt             0.18206326                0.18206326
                   (3.24)                    (3.67)
exper          0.12940770   0.12940770   0.12940770
                   (6.23)       (6.61)       (7.06)
expersq       -0.00224836  -0.00224836  -0.00224836
                  (-4.72)      (-5.48)      (-5.74)
black         -0.12752139  -0.12752139  -0.12752139
                  (-2.57)      (-2.76)      (-2.93)
xt_hat                      0.18206326
                                 (3.44)
v_xt                                     -0.10749438
                                             (-2.17)
_cons          2.79195292   2.79195292   2.79195292
                   (2.99)       (3.18)       (3.39)
------------------------------------------------
N                    3010         3010         3010
------------------------------------------------
```

The third two-stage version of the IV strategy, which applies in the case of one endogenous variable and one excluded instrument, is sometimes called the **Wald estimator**. First regress X^T on X^C and Z (let $\hat{\pi}$ be the estimated coefficient on Z), then regress y on Z and X^C, (let $\hat{\gamma}$ be the estimated coefficient on Z), and the ratio of coefficients on Z ($\hat{\gamma}/\hat{\pi}$) is $\hat{\beta}_{IV}$, which will give the same estimate as the standard IV estimator. The regression of y on Z and X^C is sometimes called the **reduced form regression**, though this name is often applied to other regressions, so I will avoid using the term.

Exhibit 4.3.4 *Code showing equivalence of Wald estimator to IV using the Card data.*

```
use http://pped.org/card, clear
loc xc "exper* smsa* south black reg662-reg669"
ren lwage y
loc z "nearc4"
g xt=(educ>=16)
ivreg2 y `xc' (xt=`z') [pw=weight], first
loc iv=_b[xt]
```

```
regress xt `z' `xc' [pw=weight]
loc p=_b[`z']
regress y `z' `xc'  [pw=weight]
loc g=_b[`z']
di "compare " `g'/`p' " to " `iv'
```

The generalized method of moments (GMM), limited-information maximum likelihood (LIML), and continuously-updated GMM estimation (CUE) forms of IV are discussed at length in Baum et al. (2007) and various implementations are available from the `ivregress` and `ivreg2` commands. Some forms of IV may be expressed as k-class estimation (including the deprecated Jackknife IV Estimator, or JIVE), available from `ivreg2`, and there are many other forms of IV models, including official Stata commands such as `ivprobit`, `treatreg`, and `ivtobit`, and user-written additions such as `qvf` (Hardin et al. 2003), `jive` (Poi, 2006), and `ivpois` (on SSC). Abbring and Van den Berg (2003b) discusses identification of a survival regression with endogenous regressors, but there is no simple analog to IV for survival regression. Even a logit regression is difficult to generalize to the IV-GMM setting (Lucchetti 2002).

The GMM flavor of IV offers increased efficiency over the standard 2SLS estimator, and the LIML flavor offers increased robustness. JIVE is out of favor due to results showing its poor performance in finite samples, notably Davidson and MacKinnon (2007); see also Kinal (1980) and Fuller (1977) on the moments of IV estimators.

4.4 Finding Excluded Instruments

The hard part of IV is finding a suitable Z matrix. The excluded instruments in Z have to be strongly correlated with the endogenous X^T and uncorrelated with the unobservable error e. But note that the problem we wish to solve is that the endogenous X^T is correlated with the unobservable error e. A good story is the crucial element in any plausible IV specification. We the readers need to believe that Z is strongly correlated with the endogenous X^T but has no direct impact on y (is uncorrelated with the unobservable error e), since the assumptions are not directly testable. However, the tests discussed in the following sections can help buttress a convincing story, and should be reported in any case.

In general, specification search in the first stage regressions of X^T on some Z does not bias estimates or inference, nor does using generated regressors. However, it is easy to produce counterexamples to this general rule. For example,

taking $Z = X^T + \nu$ where ν is a small random error will produce strong identification diagnostics, and might pass overidentification tests described in the next section, but will not improve estimates (and could lead to substantially less accurate inference).

If some Z are weak instruments, then regressing X^T on Z to get $\hat{X}^T$ and using $\hat{X}^T$ as the excluded instruments in an IV regression of y on X^T and X^C will likewise produce strong identification diagnostics, but will not improve estimates or inference. Hall et al. (1996) reported that choosing instruments based on measures of the strength of identification could actually increase bias and size distortions.

4.5 Testing Assumptions Required for IV

Because the IV estimator is so sensitive to violations of assumptions, it is especially crucial to check that the assumptions are satisfied, insofar as possible. The code in Exhibit 4.5.1 shows performing a variety of checks, each of which is discussed below, using the data from Card (1995) to estimate the impact of education on wages, where nearness to a college is posited as a source of exogenous variation in educational attainment. Note that `ivreg2`, `ranktest`, `ivreset`, and `estout`, all on SSC, are required to run the code. We find the return to an additional year of education is about 7% using ordinary regression or 18% using an IV method.

Exhibit 4.5.1 *Code showing IV specification tests using the Card data.*

```
use http://pped.org/card.dta, clear
g m=married==1
la var m "Married"
loc xc "exper expersq m black south smsa reg662-reg669 smsa66"
reg lwage educ `xc' [pw=weight]
loc ols=_b[educ]
est sto OLS
ivreg2 lwage `xc' (educ=near*) [pw=weight], endog(educ)
est sto IV
test educ=`ols'
estadd scalar vs_OLS=r(p)
ivreset
estadd scalar RESET=r(chi2)
estadd scalar RESET_p=r(p)
esttab OLS IV using c.tex, la mti sca(`s') keep(e*)
```

	(1) OLS	(2) IV
Educ attainment in years	0.0730***	0.179**
	(18.61)	(3.22)
Experience (years working) in 1976	0.0810***	0.120***
	(10.61)	(5.55)
Squared experience (years working) in 1976	-0.00217***	-0.00196***
	(-5.98)	(-4.23)
Observations	3010	3010
F	73.29	37.94
r2	0.288	0.0621
vs_OLS		0.0565
estatp		0.0345
jp		0.240
idp		0.000402
widstat		7.937
RESET_p		0.898

t statistics in parentheses

* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

4.5.2 Test IV versus OLS

We always want to test whether the OLS point estimate is included in the IV confidence interval, which is done in Exhibit 4.5.1 by saving the OLS point estimate, and testing equality to that value of the IV estimate. The p-value is reported in the table as 5.65 percent, so we cannot reject the null that the true value underlying the IV estimate is equal to the OLS point estimate. This indicates that the higher value of the IV point estimate merely indicates that the OLS estimate could be biased downward, but the OLS point estimate is also a plausible estimate of the effect of education on earnings. With a larger sample size, the precision of the IV estimator would improve, and we would find that the OLS point estimate is no longer included in the IV confidence interval, but there is no support for that assertion in Exhibit 4.5.1.

Since the IV confidence region includes the OLS confidence interval, the evidence on OLS bias is very weak, and we have learned little from the IV estimate. Still, if we accept the exclusion restrictions as valid, the evidence does not sup-

port a story where omitting ability (causing both increased wages and increased education) leads to positive bias. If anything, the bias seems likely to be negative, perhaps due to unobserved heterogeneity in discount rates or credit market failures. In the latter case, the omitted factor may be a social or economic disadvantage. Those who have the most to gain from additional education may have bad credit, or no access to good information about how much they stand to gain from additional education. Stories about ability bias are less plausible, since effects much smaller than the OLS point estimate lie outside the IV confidence interval.

4.5.3 Tests of Endogeneity

We would not want to use a high-variance IV estimator if the relevant explanatory variables were not endogenous, so we test endogeneity. Even if we have an excluded instrument that satisfies $E(Z'e) = 0$ there is no guarantee that $E(X^{T'}\varepsilon) \neq 0$ as we have been assuming, and if $E(X^{T'}\varepsilon) = 0$ we prefer ordinary regression to IV. So we should test the null that $E(X^{T'}\varepsilon) = 0$ (a test of endogeneity), though this test requires instrument validity ($E(Z'e) = 0$), so it should follow any feasible overid tests.

Baum et al. (2007) describe several methods to test the endogeneity of a variable in X^T, including the `endog` option of `ivreg2` and the standalone `ivendog` command (both available from SSC, with excellent help files). Section 4.2 above also shows how the control function form of IV can be used to test endogeneity of a variable in X^T.

The test statistic in Exhibit 4.5.1 is labeled "Endogeneity test of endogenous regressors" in the output for `ivreg2`, and under the null hypothesis that the specified endogenous regressors are exogenous, the test statistic is distributed $\chi^2(k)$ with degrees of freedom k equal to the number of regressors tested. The p-value is reported in the table as 3.45 percent, so we reject the null that education is exogenous.

4.5.4 Exclusion Restrictions in IV

The exclusion restrictions $E(Z'e) = 0$ cannot be directly tested, but if there are more excluded instruments than endogenous regressors, an overidentification or **overid** (pronounced over-eye-dee) test is feasible and the result should be reported. If there as exactly as many excluded instruments as endogenous regressors, the equation is **exactly identified**, and no overid test is feasible.

However, if Z is truly exogenous, it is likely also true that $E(W'e) = 0$, where W contains Z and squares and cross products of Z. Thus there is always a feasible overid test using an augmented set of excluded instruments (though $E(W'e) = 0$ is a stronger condition than $E(Z'e) = 0$, only in the case where we reject the overid null for each element of W not in Z do we not gain anything by augmenting the set of excluded instruments). For example, if you have two good excluded instruments, you might multiply them together, and square each, to produce five excluded instruments. Testing the three extra overidentification restrictions is like a RESET test of excluded instruments. In addition, interactions of Z and X^C may be good candidates for excluded instruments. For reasons discussed below, adding excluded instruments willy-nilly is a bad idea, and with many weak instruments, LIML or CUE is preferred to standard IV/2SLS.

Baum et al. (2007) and the help file for `ivreg2` discuss the implementation of overid tests in `ivreg2`; see also `overid` (on SSC). Passing the overid test (i.e. failing to reject the null of zero correlation) is neither necessary nor sufficient for instrument validity ($E(Z'e) = 0$), but rejecting the null in an overid test should lead you to reconsider your IV strategy, and perhaps to look for different excluded instruments.

The test statistic in Exhibit 4.5.1 is labeled the "Hansen J statistic" in the output for `ivreg2`, and it is distributed $\chi^2(1)$ under the null that the excluded instruments satisfy the exclusion restrictions $E(Z'e) = 0$. The p-value is reported in the table as 24 percent, so we do not reject the null that the excluded instruments satisfy the exclusion restrictions.

4.5.5 Identification and Weak Instruments

Even if we have an excluded instrument that satisfies $E(Z'e) = 0$ there is no guarantee that $E(Z'X^T) \neq 0$. This is the second of the two crucial assumptions, and presents problems of various sizes in almost all IV specifications. The extent to which $E(Z'X^T) \neq 0$ determined the strength of identification. Baum et al. (2007) describe tests of identification, which amount to tests of the rank of $E(Z'X^T)$. These rank tests address the concern that a number of excluded instruments may generate exogenous variation in one endogenous variable and be uncorrelated with another endogenous variable, so the equation is not identified even though it satisfies the order condition (the number of excluded instruments is at least as great as the number of endogenous variables). For example, if we have two endogenous variables X_1 and X_2 and three excluded instruments, all three excluded instruments may be correlated with X_1 and not with X_2. The identification

tests look at the least partial correlation, or the minimum eigenvalue of the Cragg-Donald statistic (Stock and Yogo 2005), for example, and measures of whether at least one endogenous variable has no correlation with the excluded instruments.

Even if we reject the null of underidentification and conclude $E(Z'X^T) \neq 0$, we can still face a "weak instruments" problem if some elements of $E(Z'X^T)$ are close to zero. The IV estimate is always biased, but is less biased than OLS to the extent that identification is strong. In the limit of weak instruments, there would be no improvement over OLS in terms of bias and the bias would be 100% of OLS, and in the other limit, the bias would be zero percent of the OLS bias (though this would require that the correlation between X^T and Z be perfect, which is impossible since X^T is endogenous and Z is exogenous). In applications, you'd like to know where you are on that spectrum, even if only approximately.

There is also a major distortion in the size of hypothesis tests. If you believe you are incorrectly rejecting a null hypothesis about five percent of the time (i.e. you have chosen a size $\alpha = 0.05$), you may actually face a size of 20 percent, or more.

Stock and Yogo (2005) reported rule-of-thumb critical values to measure the extent of both of these problems. Their Table 1 shows the value of a statistic measuring the predictive power of the excluded instruments that will imply a limit of the bias to some percentage of OLS. For two endogenous variables and three excluded instruments (n=2, $K_2 = 5$) the minimum value to limit the bias to 20% of OLS is 5.91. `ivreg2` reports these values as **Stock-Yogo weak ID test critical values**: one set for various percentages of "maximal IV relative bias" (largest bias relative to OLS) and one set for "maximal IV size" (the largest size of a nominal 5% test).

The key point is that all IV and IV-type specifications can suffer from bias and size distortions, not to mention inefficiency and sometimes failures of exclusion restrictions. The Stock and Yogo (2005) approach measures how strong identification is in your sample, and `ranktest` (on SSC) offers a way forward for cases where errors are not assumed to be independently and identically distributed, but neither provides solutions in the event that weak instruments appear to be a problem.

A further limitation is that these identification statistics only apply to the linear case, not the nonlinear analogs, including those estimated via generalized linear models (GLM). In practice, researchers typically report the identification statistics for the closest linear analog, i.e. run `ivreg2` and report the output alongside the output from `ivprobit` or `ivpois` (on SSC) or some other model.

If you suspect weak instruments may be producing large bias or size distor-

tions, you have several options. You can find better excluded instruments, possibly simply by transforming your existing instruments, or including additional covariates. You can use LIML or CUE which are more robust to many weak instruments than standard IV. Perhaps best of all, you can conduct inference that is robust to weak instruments; with one endogenous variable, use `condivreg` (Mikusheva and Poi 2006), if you are willing to assume i.i.d. errors, or with more than one endogenous variable or non-i.i.d. errors, use tests from Anderson and Rubin (1949) described by Baum et al. (2007), sections 7.4 and 8.

The test statistic in Exhibit 4.5.1 is labeled "Weak identification test (Kleibergen-Paap rk Wald F statistic)" in the output for `ivreg2`, and the test statistic has a rather strange distribution. Stock and Yogo (2005) give illustrative critical values derived from simulations, given desired bounds on size of tests and bias relative to OLS, but the output in Exhibit 4.5.1 reports only critical values for size of test. The test statistic is 7.937 and the critical values indicate that we can expect the size of tests with a nominal size of 5 percent to be between 20 and 25 percent; if we set the maximal size of test we would accept at 25 percent, we would reject that we had a weak instruments problem, but if we set the maximal size of test we would accept at 20 percent, we would fail to reject that we had a weak instruments problem. We might want a wider 99 percent confidence interval in this case, to approximate a 95 percent confidence interval, but the relative bias is probably also substantial. If we re-estimate using also the interaction of the two excluded instruments, as shown in Exhibit 4.5.6, we can see that the size is probably now greater than 25 percent and bias roughly 30 percent of OLS. We can re-estimate using a LIML estimator, and find the size distortion results in a test with a size in the 10 to 15 percent range, and presumably a smaller bias as well. By choosing a smaller nominal size, e.g. a 98 percent confidence interval, we can come closer to our target size of 95 percent.

Exhibit 4.5.6 *Code showing IV specification tests using the Card data.*

```
g nboth=nearc2*nearc4
ivreg2 lwage 'xc' (educ=n*) [pw=weight]
------------------------------------------------------------------------------
Weak identification test (Kleibergen-Paap rk Wald F statistic):          5.614
Stock-Yogo weak ID test critical values:  5% maximal IV relative bias    13.91
                                         10% maximal IV relative bias     9.08
                                         20% maximal IV relative bias     6.46
                                         30% maximal IV relative bias     5.39
                                         10% maximal IV size             22.30
                                         15% maximal IV size             12.83
                                         20% maximal IV size              9.54
                                         25% maximal IV size              7.80
Source: Stock-Yogo (2005).  Reproduced by permission.
```

```
NB: Critical values are for Cragg-Donald F statistic and i.i.d. errors.
------------------------------------------------------------------------------
ivreg2 lwage 'xc' (educ=near*) [pw=weight], liml level(98)
------------------------------------------------------------------------------
Weak identification test (Kleibergen-Paap rk Wald F statistic):          7.937
Stock-Yogo weak ID test critical values: 10% maximal LIML size            8.68
                                         15% maximal LIML size            5.33
                                         20% maximal LIML size            4.42
                                         25% maximal LIML size            3.92
Source: Stock-Yogo (2005).  Reproduced by permission.
NB: Critical values are for Cragg-Donald F statistic and i.i.d. errors.
------------------------------------------------------------------------------
```

To construct an Anderson-Rubin confidence interval, we can do a simple grid search, as in Exhibit 4.5.7, giving a confidence interval of (.074,.41); see also Nichols (2006; p. 18) and Baum et al. (2007; p. 30).

Exhibit 4.5.7 *Manually constructing an Anderson-Rubin confidence interval.*

```
g y=.
foreach beta of numlist 70/75 370/420 {
 qui replace y=lwage-(`beta'/1000)*educ
 qui reg y `xc' nearc2 nearc4 [pw=weight]
 qui test nearc2 nearc4
 if inrange(r(p),.04,.06) di as res "Test of beta=0." `beta',r(p)
 }
```

4.5.8 Functional Form Tests in IV

As Baum et al. (2007; Sec. 9) and Wooldridge (2002; p. 125) discuss, the RESET test regressing residuals on predicted y and powers thereof is properly a test of a linearity assumption, or a test of functional form restrictions. `ivreset` performs the IV version of the test in Stata. The null is that $E(y|X)$ is linear in X and the test statistic in Exhibit 4.5.1 (labeled "RESET_p" in the table), gives a p-value of 90 percent so we do not reject the null. This does not mean that $E(y|X)$ is truly linear in X but it does mean that our data and the RESET test do not reject that possibility; RESET is a fairly low-powered test.

A more informative specification check is the graphical version of RESET: predict $\hat{X}^T$ after the first stage regressions, then compute forecasts $\hat{y} = X^T\hat{\beta}_T^{IV} + X^C\hat{\beta}_C$ and $\hat{y}_f = \hat{X}^T\hat{\beta}_T^{IV} + X^C\hat{\beta}_C$ and graph a `scatter` of the residuals $\hat{\varepsilon} = y - \hat{y}$ against $\hat{y}_f$. Any unmodeled nonlinearities may be apparent as a pattern in the scatterplot.

```
ivreg2 y xc* (xt = z*), first
quietly regress xt xc* z*
```

```
predict xhat
quietly regress y xhat xc*
predict ehat, resid
predict yhatf
drop xhat
ren xt xhat
predict yhat
scatter ehat yhatf, name(f)
scatter ehat yhat, name(y)
scatter ehat xhat, name(x)
```

4.5.9 Standard Errors in IV

The largest issue in IV estimation is often that the variance of the estimator is much larger than ordinary regression. Just as with ordinary regression, the standard errors are asymptotically valid for inference under the restrictive assumptions that the disturbances are independently and identically distributed. Getting standard errors robust to various violations of these assumptions is easily accomplished using the `ivreg2` command described in Baum et al. (2007). Many other commands estimating IV models offer no equivalent robust SE estimates, but it may be possible to assess the size and direction of SE corrections using the nearest linear analog, in the spirit of using estimated design effects in the survey regression context.

The estimation shown in Exhibit 4.5.1 is weighted using sample weights specified as pweights (see `help weights` in Stata), and is therefore robust to heteroskedasticity (see also 1.5.4 above). We would also like to make it robust to clustering, to account the sample design at the very least, but the clustering variable is not on the data. If we cluster by `gp` as a proxy for sampling clusters, however, the standard errors change only modestly, so we are reassured that the bias in standard errors is not too drastic. Note that the weak instrument diagnostics assume i.i.d. errors, so we are less confident of the interpretation of those statistics.

4.5.10 Inference in IV

Assuming we have computed consistent standard errors, and the best IV estimate we can, using a good set of Z and X^C variables, there remains the question of how we interpret the estimates and tests. Typically, IV identifies a particular LATE, namely the effect of an increase in X^T due to an increase in Z. If X^T were college and Z an exogenous source of financial aid, then the IV estimate of the

effect of X^T on wages would be the college wage premium for those who were induced to attend college by being eligible for the marginally more generous aid package.

Angrist and Krueger (1991) estimated the effect of education on earnings using compulsory schooling laws as a justification for quarter of birth as Z. Even if the critiques of Bound et al. (1995) did not apply, the identified effect would be for an increase in education due to being forced to remain in school a few months more. That is, the measured wage effect of an additional year of education is for the eleventh grade, roughly, and only for those who would have dropped out if not for compulsory schooling laws.

Sometimes, a LATE of this form is exactly the estimate desired, but if we cannot reject that the IV estimate differs from the OLS estimate, or the IV confidence region includes the OLS confidence region, we may not have improved estimates, but merely produced noisier ones. Only in the case where the IV estimate differs can we hope to ascertain the nature of selection bias.

Most importantly, the tests enumerated here are only a rough guide to whether instruments are useful or not. Theory must be the first and foremost guide. If you pick variables that are alternative outcomes of your treatment variable, they may be highly correlated with the treatment variable, and uncorrelated with the unobserved error, and pass all the feasible tests, but may produce meaningless results. If the treatment variable is not being affected by the excluded instruments, but is rather affecting it, how should we interpret the results? For example, Kelly (2000) regresses crime on income inequality and police, treating police as potentially endogenous and using mean income, non-police government spending, and percent Republican votes in 1988 presidential election as instruments. Ignore for a moment the test results in the paper, and pretend these pass the tests for excluded instruments; is it plausible that the size of the police force is affected by non-police government spending, instead of affecting it? Is it possible that income inequality is endogenously determined with these excluded instruments? You must ask yourself these types of questions when you read research using instrumental variables, or use them yourself.

4.6 Binary variables

With a single endogenous variable T that is a binary indicator, the instrumental variables model is especially inefficient, but there are several possible improvements. The "treatment effect regression" model used in `treatreg` is closely re-

lated to the "selection" model (Heckman 1974; 1976; 1979; 1990). If we assume that selection of treatment status T (or whether or not the outcome is observed in the selection model estimated by `heckman`) is a function of exogenous variables Z (some of which also affect outcomes directly and are therefore included in X), and the errors are normal, we can estimate the generalized residual from a first-stage regression of T on Z and include it as an additional explanatory variable in the second stage, or estimate both equations via maximum likelihood.

The two-step estimator (analogous to second two-stage IV estimator discussed in 4.3) for the treatment effects regression regresses the endogenous dummy T on Z with a probit, predicts the generalized residual, then includes that generalized residual[1] in the second stage regression of y on X. The Heckman two-step model has identical steps, but in that model, y is missing for observations where the dummy T is zero, and observed where T is one. In either case, the generalized residual from the first stage is included in the second stage, but `heckman` does not estimate the effect of T since it would be one for each observation in the sample, whereas `treatreg` does estimate that coefficient.

In fact, rather than use the consistent two-step estimator, we would usually use the maximum likelihood estimator (the default option) in either `heckman` or `treatreg`. The model for `heckman` is

$$\begin{aligned} T &= I(Zg + v > 0) \\ y &= Ta + Xb + e \end{aligned}$$

$$T = I(Zg + v > 0) y = Ta + Xb + e$$

and the model for `treatreg` is

$$T = I(Zg + v > 0) y = Xb + e \quad \text{observed} \, \forall T = 1$$

where Z includes variables known to affect selection, and we can estimate via maximum likelihood by assuming v and e are jointly normal. This assumption is an unpleasant feature of both `heckman` and `treatreg` in that a small departure from joint normality can make the estimators no longer consistent, and we could wind up with worse estimates than we would have ignoring selection or endogeneity. There are a variety of semiparametric estimators for treatment effects and selection models, some of which are discussed by Das et al. (2003).

[1] The generalized residual of the probit is called a hazard in Stata's manuals, because it is a density divided by a cumulative probability, and is called an Inverse Mills Ratio by many authors; see also http://www.stata.com/support/faqs/stat/invmills.html

With a binary endogenous treatment variable and a binary outcome, the bivariate probit is a convenient estimation strategy. `biprobit` is a maximum likelihood estimator of two correlated probits, so it estimates the correlation between v and e in a model of the form:

$$\begin{aligned} T &= I(Zg + v > 0) \\ y &= I(Ta + Xb + e > 0) \end{aligned}$$

to get a consistent estimate of a. An alternative semiparametric approach for this "triangular" or "recursive" system is given by assuming heteroskedasticity which is a function of observables and modeling the distribution of errors using kernel estimation (Newey et al. 1999, Klein and Vella 2009; forthcoming). Other models with different structures (e.g. multinomial) can be estimated via a similar strategy to `biprobit` by `cmp` (on SSC); see Roodman (2009b).

With a single endogenous variable T that is a binary indicator, and a continuous outcome, we could use `treatreg` or regular IV. We can also use a modified IV with greater efficiency that predicts the probability that $T = 1$ and uses the predicted probability as an instrument (Wooldridge 2002; p. 626), which makes a similar assumption about the process determining selection into treatment, but weaker assumptions than `treatreg` on the joint distribution of errors. In general, as always, if we make assumptions about errors, we can gain some efficiency in the event that our assumptions are good ones, but lose some robustness against violations of assumptions.

In the example in Exhibit 4.6.1, we use the Card (1995) data as before, but model education as a binary treatment **college**. These regressions also indicate that the OLS estimate may be biased downward, though the OLS confidence interval is contained in the `treatreg` confidence interval and each of the IV intervals, so we cannot conclude much with confidence.

Exhibit 4.6.1 *A single binary endogenous variable and a continuous outcome.*

```
use http://pped.org/card, clear
loc x "exper* smsa* south mar black reg662-reg669"
g byte coll=educ>15
probit coll `x' nearc2 nearc4
predict p
treatreg lw `x', treat(coll=nearc2 nearc4)
ivreg2 lw `x' (coll=nearc2 nearc4), first liml
ivreg2 lw `x' (coll=p), first liml
```

4.7 Panel IV

If we don't feel comfortable with the assumption that our excluded instruments Z are uncorrelated with every unmeasured factor in e that affects y, we may be more comfortable assuming that changes in Z are uncorrelated with changes in e, i.e. Z may be correlated with time-invariant components of e only. Then we write

$$y_{it} = X_{it}\beta + u_i + e_{it}$$

where u_i captures the effect of time-invariant individual-specific unobservable factors (the individual effect) and e_{it} is the idiosyncratic error. We difference and write

$$\Delta y_{it} = \Delta X_{it}\beta + \xi_{it}$$

where the u_i drops out. If we assume $\xi_{it} = e_{it} - e_{is)}$ is uncorrelated with changes in Z, we have a panel IV model, estimated with `xtivreg` with the option `fe` for a fixed-effects model or option `fd` for a first-difference model.

The fixed-effects IV estimator is fairly robust to mistakenly assuming a common effect β in the presence of individual-specific effects β_i, provided a full set of period dummy variables is included and endogenous explanatory variables are continuous (Murtazashvili and Wooldridge 2008). However, it is important to remember that the maintained assumption of a common effect is implausible in many cases.

Note that the estimators in `xthtaylor` and `xtivreg` use instrumental variables (see chapter 4), but each is designed for different problems. The estimators implemented in `xtivreg` assume that some variables in the model (the endogenous variables) are correlated with the idiosyncratic error e but there are excluded instruments that have no direct effect on the outcome and are uncorrelated with e. On the other hand, the `xthtaylor` estimators assume that some of the explanatory variables are correlated with the individual effects u_i, but none are correlated with the idiosyncratic error e_{it}.

4.8 Heterogeneity

The assumption that everyone has the same mean response to treatment allows us to generalize to the entire population from an IV estimate, and allowing some kind of heterogeneity in treatment effects leads us to interpret IV estimates as the local average treatment effect for those induced by the instruments to get more

treatment (i.e. increase X^T). With heterogeneity in treatment effects, we may be interested in the distribution of heterogeneous treatment effects, rather than the average treatment effects, which requires a IV version of quantile regression (Chernozhukov and Hansen 2005; 2006, Horowitz and Lee 2007, Chernozhukov et al. 2007, Chernozhukov and Hansen 2008).

More generally, there is a budding literature on local instrumental variables which seeks to estimate the average treatment effect over some well-defined subpopulation. See for example Blundell and Powell (2003), Heckman and Vytlacil (1999; 2001; 2004), Heckman et al. (2006)

Chapter 5

Regression Discontinuity Methods

The idea of the regression discontinuity (RD) design is to exploit an observable discontinuity in the level of treatment related to an assignment variable Z, so the level of treatment X^T jumps discontinuously at some value of Z, the "cutoff." Let Z_0 denote the cutoff. In the neighborhood of Z_0, under some often plausible assumptions, a discontinuous jump in the outcome y can be attributed to the change in the level of treatment. Near Z_0, the level of treatment can be treated *as if* it is randomly assigned. For this reason, the RD design is generally regarded as having the greatest internal validity of the quasi-experimental estimators.

This is perhaps clearest in a picture, or two. Figure 5.1 shows the conditional mean of outcome y given Z, where every unit with Z greater than some cutoff level c gets treatment ($X = 1$) and every unit with Z less than c is untreated ($X = 0$). The data we see is shown as solid lines, and the data needed to estimate the average treatment effect is shown as dotted lines. In this figure, the treatment effect is negative for $Z < b$ and increasing in Z. But we can't estimate the treatment effect everwhere on the interval $[a, d]$, since we cannot distinguish this picture from one with identical solid lines and different dashed lines in figure 5.2 where the treatment effect is declining in Z and is actually negative for some treated units with high Z. In both graphs, however, we can estimate the treatment effect at exactly one point, where we see both the outcome and the counterfactual; at the point $Z = c$. This is true as long as the curves are continuous at that point and the value of Z cannot be manipulated by units to determine treatment status (i.e. units on either side of c are exchangeable).

Examples of assignment and treatment congenial to this estimation technique include share of votes received in a US Congressional election by the Democratic candidate as Z, which induces a clear discontinuity in X^T, whether a Democrat

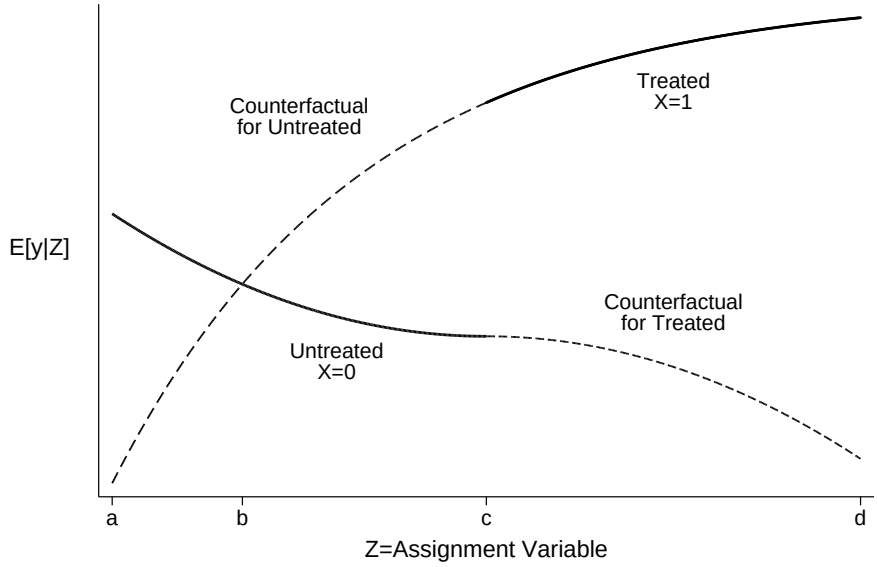

Figure 5.1: A picture for the regression discontinuity design.

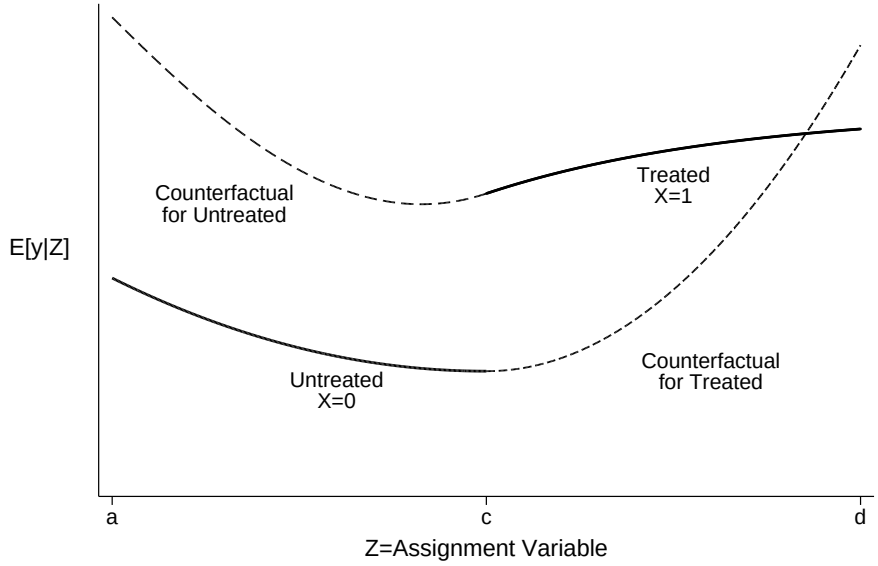

Figure 5.2: An alternative picture for the regression discontinuity design, changing only unobserved counterfactuals.

occupies the office the following term. Then X^T may affect various outcomes y, if Democratic and Republican candidates actually differ in close races (Lee 1993; 2008). DiNardo and Lee (2002) use the share of votes received for a union as Z, and find little effect of unionization in these evenly split firms. Note that unions may may affect the survival of a firm (but do not seem to), and they point out that the union wage premium y can be consistently estimated only if survival is not affected (no differential attrition around Z_0).

The standard treatment of RD is Hahn et al. (2001), who clarified the link to IV methods. Recent papers by Imbens and Lemieux (2008), Lee and Card (2008), and McCrary (2008) discuss some important practical issues related to RD designs.

Many authors stress a distinction between "sharp" and "fuzzy" RD. In sharp RD designs, the level of treatment rises from zero to one at Z_0, as in the case where treatment is having a Democratic representative in the US Congress, or establishing a union, and a winning vote share defines Z_0. In fuzzy RD designs, the level of treatment increases discontinuously, or the probability of treatment increases discontinuously, but not from zero to one, so we may want to deflate by the expected increase in X^T at Z_0 in constructing our estimate of the causal impact of a one-unit change in X^T.

In sharp RD designs, the jump in y at Z_0 is the estimate of the causal impact of X^T at Z_0. In a fuzzy RD design, the jump in y divided by the jump in X^T at Z_0 is the local Wald estimate (equivalent to a local IV estimate) of the causal impact. Note that the local Wald estimate reduces to the jump in y at Z_0 in a sharp RD design as the jump in X^T is one, so the distinction between fuzzy and sharp RD is not that sharp.

Some authors, e.g. Shadish et al. (2002; page 229), seem to characterize as fuzzy RD a wider class of problems where the cutoff itself may not be sharply defined, but without a true discontinuity, there can be no RD. The fuzziness in fuzzy RD arises only from probabilistic assignment of X^T in the neighborhood of Z_0. There is still a sharp discontinuity in the expected level of treatment at a cutoff, which is how the treatment effect is identified.

5.1 Key assumptions and tests

The assumptions that allow us to infer a causal effect on y due to an abrupt change in X^T at Z_0 are that the change in X^T at Z_0 is truly discontinuous, Z is observed without error (See also Lee and Card 2008), y is a continuous function of Z at

Z_0 conditional on treatment status for each individual, and that individuals are not sorted nonrandomly with respect to Z in the neighborhood of Z_0 (for example, based on their responsiveness to treatment). None of these assumptions can be directly tested, but there are diagnostic tests that should always be employed.

The first is to test the null that no discontinuity in treatment occurs at Z_0, since without identifying a jump in X^T we will be unable to claim to identify the causal impact of said jump. The second is to test that there are no other extraneous discontinuities in X^T or y away from Z_0, since this would call into question whether the functions would be smooth through Z_0 in the absence of the dinscontinuous assignment of treatment. The third and fourth test that predetermined characteristics and the density of Z exhibit no jump at Z_0, since these call into question the exchangeability of observations on either side of Z_0. Then the estimate itself usually supplies a test that the treatment effect is nonzero (y jumps at Z_0 because X^T jumps at Z_0).

Abusing notation somewhat so that Δ is an estimate of the discontinuous jump in a variable, we can enumerate these tests as:

1. $\Delta X^T(Z_0) \neq 0$
2. $\Delta X^T(Z \neq Z_0) = 0$ and $\Delta y(Z \neq Z_0) = 0$
3. $\Delta X^C(Z_0) = 0$
4. $\Delta f(Z_0) = 0$
5. $\Delta y(Z_0) \neq 0$ or $\left(\frac{\Delta y(Z_0)}{\Delta X^T(Z_0)}\right) \neq 0$

5.2 Methodological choices

Estimating the size of a discontinuous jump can be accomplished by comparing means in small bins of Z to the left and right of Z_0, or via a regression with various powers of Z, an indicator D for $Z > Z_0$, and interactions of all Z terms with D (estimating a polynomial in Z on both sides of Z_0, and comparing the intercepts at Z_0). However, since the goal is to compute an effect at precisely one point (Z_0) using only the closest observations, the standard approach is to use local linear regression, which minimizes bias (Fan and Gijbels 1996). In Stata 10 and later, this is done with the `lpoly` command; users of previous versions of Stata can use `locpoly` (`findit locpoly`) described by Gutierrez et al. (2003).

Having chosen to use local linear regression, other key issues are the choice of bandwidth and kernel. Various techniques are available for choosing bandwidths (Fan and Gijbels 1996, Stone 1974; 1977), and the triangle kernel has good properties in the RD context, due to being boundary optimal (Cheng et al. 1997).

There are several rule-of-thumb bandwidth choosers and cross-validation techniques for automating bandwidth choice, but none is foolproof. McCrary (2008) contains a useful discussion of bandwidth choice, and asserts that there is no substitute for visual inspection comparing the `lpoly` smooth with the pattern in a `scatter` graph. Imbens and Kalyanaraman (2009) provides an optimal bandwidth (to minimize mean squared error) for the most common type of design.

Whether we use a rule-of-thumb bandwidth, a cross-validated bandwidth, or a bandwidth deemed optimal for some set of models, we want to test whether estimates are robust to such a bandwidth choice. Because different bandwidth choices can produce different estimates, the researcher should report at least three estimates as an informal sensitivity test: one using the preferred bandwidth, and estimates using twice and half the preferred bandwidth.

5.3 Testing assumptions

The same methodological choices apply in many of tests of required assumptions. While the assumptions cannot be directly verified, as in IV, we can provide evidence supporting or undermining each of the assumptions.

5.3.1 Test X^T jumps at Z_0

The identifying assumption is that X^T jumps at Z_0 due to some known legal or program design rules, but we can test that assumption easily enough. A standard approach to computing standard errors is to `bootstrap` the local linear regression, which requires wrapping the estimation in a program, as shown below.

```
program discont, rclass
 version 10
 syntax [varlist(min=2 max=2)] [, *]
 tokenize `varlist'
 tempvar z f0 f1
 qui g `z'=0 in 1
 local opt "at(`z') nogr k(tri) deg(1) `options'"
 lpoly `1' `2' if `2'<0, gen(`f0') `opt'
 lpoly `1' `2' if `2'>=0, gen(`f1') `opt'
 return scalar d=`=`f1'[1]-`f0'[1]'
 di as txt "Estimate: " as res `f1'[1]-`f0'[1]
```

```
  eret clear
end
```

In the program, the assignment variable Z is assumed to be defined so that the cutoff $Z_0 = 0$ (easily done with a single `replace` or `generate` command subtracting Z_0 from Z). The triangle kernel is used, and the default bandwidth is chosen by `lpoly`, which is probably suboptimal for this application. The local linear regressions are computed twice, once using observations on one side of the cutoff, for $Z < 0$, and once for $Z \geq 0$. The estimate of the jump uses only the predictions at the cutoff $Z_0 = 0$, so these are the only values computed by `lpoly`.

We can easily generate data to try this example program out:

```
ssc inst rd, replace
net get rd
use votex if i==1
ren lne y
ren win xt
ren d z
foreach v of varlist pop-vet {
 ren `v' xc_`v'
 }
bs r(d): discont y z
```

In a more elaborate version of this program called `rd` (which also supports earlier versions of Stata), available by typing `ssc inst rd, replace` in Stata, the default bandwidth is selected to include at least 30 observations in estimates at both sides of the boundary; other options are also available. Try `findit bandwidth` to find more sophisticated bandwidth choosers for Stata. The key point is to use the `at()` option so that the difference in local regression predictions can be computed at Z_0.

A slightly more elaborate version of this program would save local linear regression estimates at a number of points, and offer a graph:

```
program discont2, rclass
 version 10
 syntax [varlist(min=2 max=2)] [, s(str) Graph *]
 tokenize `varlist'
 tempvar z f0 f1 se0 se1 ub0 ub1 lb0 lb1
 su `2', meanonly
 local N=round(100*(r(max)-r(min)))
 cap set obs `N'
 qui g `z'=(_n-1)/100 in 1/50
 qui replace `z'=-(_n-50)/100 in 51/`N'
 local opt "at(`z') nogr k(tri) deg(1) `options'"
 lpoly `1' `2' if `2'<0, gen(`f0') se(`se0') `opt'
 qui replace `f0'=. if `z'>0
 qui g `ub0'=`f0'+1.96*`se0'
```

```
  qui g `lb0'=`f0'-1.96*`se0'
  lpoly `1' `2' if `2'>=0, gen(`f1') se(`se1') `opt'
  qui replace `f1'=. if `z'<0
  qui g `ub1'=`f1'+1.96*`se1'
  qui g `lb1'=`f1'-1.96*`se1'
  return scalar d=`=`f1'[1]-`f0'[1]'
  return scalar f1=`=`f1'[1]'
  return scalar f0=`=`f0'[1]'
  forv i=1/50 {
   return scalar p`i'=`=`f1'[`i']'
   }
  forv i=51/`N' {
   return scalar n`=`i'-50'=`=`f0'[`i']'
   }
  di as txt "Estimate: " as res `f1'[1]-`f0'[1]
  if "`graph'"!="" {
   la var `z' "Assignment Variable"
   loc lines "|| line `f0' `f1' `z'"
   loc a "tw rarea `lb0' `ub0' `z' || rarea `lb1' `ub1' `z'"
   `a' || sc `1' `2', mc(gs14) leg(off) sort `lines'
   }
  if "`s'"!="" {
   ren `z' `s'`2'
   ren `f0' `s'`1'0
   ren `lb0' `s'`1'lb0
   ren `ub0' `s'`1'ub0
   ren `f1' `s'`1'1
   ren `lb1' `s'`1'lb1
   ren `ub1' `s'`1'ub1
   }
  eret clear
 end
```

In this version, the local linear regressions are computed at a number of points (in the example, the maximum of Z is assumed to be 0.5, so the program uses hundredths as a convenient unit for Z) on either side of the cutoff $Z_0 = 0$, but the estimate of a jump still uses only the two estimates at $Z_0 = 0$. Note that the `s()` option in the above program saves the local linear regression predictions (and `lpoly` confidence intervals) to new variables that can then be graphed. Graphs of all output are advisable, to assess the quality of the fit for each of several bandwidths. This program may also be bootstrapped, though recovering the standard errors around each point estimate from `bootstrap` for graphing the fit is much more work.

5.3.2 Test y and X^C continuous away from Z_0

While we need only assume continuity at Z_0, and need no assumption that the outcome and treatment variables are continuous at values of Z away from the cutoff Z_0 (i.e. $\Delta X^T(Z \neq Z_0) = 0$ and $\Delta y(Z \neq Z_0) = 0$), it is reassuring if

we fail to reject the null of a zero jump at various values of Z away from the cutoff Z_0 (or reject the null only in 5% of cases or so). Having defined a program `discont`, we can easily randomly choose 100 placebo cutoff points $Z_p \neq Z_0$, without replacement in the example below, and test the continuity of X^T and y at each.

```
bys z: g f=_n>1 if z!=0
g u=uniform()
sort f u
replace u=(_n<=100)
levelsof z if u, loc(p)
foreach val of local p {
 cap drop newz
 g newz=z-`val'
 bootstrap r(d), reps(100): discont y znew
 bootstrap r(d), reps(100): discont xt znew
}
```

5.3.3 Test X^C continuous around Z_0

If we can regard an increase in treatment X^T as randomly assigned in the neighborhood of the cutoff Z_0, then predetermined characteristics X^C such as race or sex of treated individuals should not exhibit a discontinuity at the cutoff Z_0. This is equivalent to the standard test of randomization in an experimental design, using a test of the equality of the mean of every variable in X^C across treatment and control groups (see `help hotelling` in Stata), or the logically equivalent test that all the coefficients on X^C in a regression of X^T on X^C are zero. As in the experimental setting, in practice the tests are usually done one at a time with no adjustment for multiple hypothesis testing (see `help _mtest` in Stata).

In the RD setting, this is simply a test that the measured jump in each predetermined X^C is zero at the cutoff Z_0, or $\Delta X^C(Z_0) = 0$ for all X^C. If we fail to reject that the measured jump in X^C is zero, for all X^C, we have some additional evidence that observations on both sides of the cutoff are exchangeable, at least in some neighborhood of the cutoff, and we can treat them as if they were randomly assigned treatment in that neighborhood.

Having defined the programs `discont` and `discont2`, we can simply type:

```
foreach v of varlist xc* {
 bootstrap r(d), reps(100): discont `v' z
 discont2 `v' z, s(h)
 sc `v' z, mc(gs14) sort || line h`v'0 h`v'1 hz, name(`v')
 drop hz
 }
```

5.3.4 Test density of Z continuous at cutoff

McCrary (2008) gives an excellent account of a violation of exchangability of observations around the cutoff. If individuals have preferences over treatment and can manipulate assignment, for instance by altering their Z or misreporting it, then individuals close to Z_0 may shift across the boundary. For example, some non-randomly selected subpopulation of those who are nearly eligible for food stamps may misreport income, while those who are eligible do not. This creates a discontinuity in the density of Z at Z_0. McCrary (2008) points out that the absence of a discontinuity in the density of Z at Z_0 is neither necessary nor sufficient for exchangability, but a failure to reject the null hypothesis that the jump in the density of Z at Z_0 is zero is reassuring nonetheless.

McCrary (2008) discusses a test in detail, and advocates a bandwidth chooser. We can also adapt our existing program to the purpose by using `kdens` (on SSC) to estimate the density to the left and right of Z_0:

```
kdens z if z<0, ul(0) gen(f0) at(z) tri nogr
count f0 if z>=0
replace f0=f0/r(N)*`=_N'/4
kdens z if z>=0, ll(0) gen(f1) at(z) tri nogr
count f1 if z<0
replace f1=f1/r(N)*`=_N'/4
generate f=cond(z>=0,f1,f0)
bootstrap r(d), reps(100): discont f z
discont2 f z, s(h) g
```

Or better, we could wrap the density estimation inside the program that estimates the jump, so that both are bootstrapped together; this approach is taken by `rd` available from `ssc inst rd, replace`. McCrary's own Stata code is available on the web, however, so that should be preferred to the ad hoc method described here.

5.3.5 Treatment Effect Estimator

Having defined the program `discont`, we can simply:

```
bootstrap r(d), reps(100): discont y z
```

to get an estimate of the treatment effect in a "sharp" RD setting where X^T jumps from zero to one at Z_0. For a "fuzzy" RD design, we wish to compute the jump in y scaled by the jump in X^T at Z_0, or the **local Wald estimate**, for which we need to modify our program to estimate both discontinuities. The program `rd`

available from `ssc inst rd, replace` does this, but the idea is illustrated in the program below, using the previously defined `discont` program twice.

```
prog lwald, rclass
 version 10
 syntax varlist [, w(real .06) ]
 tokenize `varlist'
 di as txt "Numerator"
 discont `1' `3', bw(`w')
 loc n=r(d)
 return scalar numerator=`n'
 di as txt "Denominator"
 discont `2' `3', bw(`w')
 loc d=r(d)
 return scalar denominator=`d'
 return scalar lwald=`n'/`d'
 di as txt "Local Wald Estimate:" as res `n'/`d'
 eret clear
end
```

This program takes three arguments, the variables y, X^T, and Z, assumes $Z_0 = 0$, and uses a hardwired default bandwidth of 0.06. Note that the default bandwidth selected by `lpoly` is inappropriate for these models, since we do not use a Gaussian kernel, and are interested in boundary estimates. The `rd` program on SSC is similar in spirit to the above, though it offers more options and does not require you to type in the whole program.

5.4 Examples

Voting examples abound. A novel estimate in Nichols and Rader (2007) measures the effect of electing as a Representative a Democratic incumbent versus a Republican incumbent on a district's receipt of federal grants:

```
 ssc inst rd, replace
 net get rd
 use votex if i==1
 g lnpc=lne-ln(votingpop)
 rd lnpc d, gr bw(.04)
 bs: rd lnpc d, x(pop-vet)
```

The above estimates that the marginally victorious Democratic incumbent brings roughly 20% less in federal grant dollars per voter to his home district than a marginally victorious Republican incumbent. However, we cannot reject the null of zero difference, and that is true for a variety of bandwidth choices (figure 5.3 shows the small insignificant effect). Note that the above is a sharp RD design,

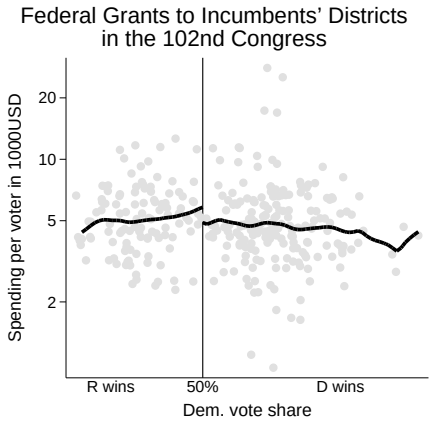

Figure 5.3: RD voting example

but the Wald estimator can be also used to estimate the effect, since the jump in `win` at 50% of vote share is one, and dividing by one has no impact on estimates.

Many good examples of fuzzy RD designs concern educational policy or interventions, e.g. van der Klaauw (2002) or Ludwig and Miller (2007). Many educational grants are awarded using deterministic functions of predetermined characteristics, lending themselves to evaluation using RD. For example, some US Department of Education grants are awarded to districts with poverty (or near-poverty) rates above a threshold, as determined by data from a prior Census, which satisfies all of the requirements for RD, though the size of the discontinuity in funding may often be insufficient to identify an effect. In many cases, a power analysis is warranted to determine the minimum detectable effect.

Returning to the Card (1995) example of the effect of education on earnings, we can imagine exploiting a discontinuity in the availability of college to residents of certain US states at the state boundary. College applicants who live 4.8 miles and 5 miles from a college may look very similar in various observable characteristics, but if a state boundary separates them at 4.9 miles from the college, and the college is a state institution, they may face very different probabilities of admission or tuition costs. The data in Card (1995) do not support this strategy, of course, since we would need to know the exact locations of all individuals relative to state boundaries, but it helps to clarify the assumptions that justify the IV approach. We need to assume that location relative to colleges is randomly sprinkled over potential applicants, which seems questionable (Black 1999), es-

pecially when one considers including parental education in the model, but that is the assumption we maintained in the IV example.

5.5 Extensions

In many cases, there is a known discontinuity in treatment probability but there may be other discontinuities at the same point, for example if the discontinuity happens at age 65 or age 18 (focal age of retirement or age of majority in the US); but see Card et al. (2008; 2009) on age 65. In such a case, we may be worried that cases on either side of the discontinuity are not exchangeable because other factors change (i.e. there is more than one treatment that discontinuously jumps). However, we may identify a causal impact by using a panel including observations before and after a new treatment is introduced. In that case, we need to maintain the usual assumptions for a panel regression, so we have lower internal validity, but we may sidestep confounding other treatment effects with the one of interest. The simplest hybrid of RD with panel methods just subtracts off the estimated jump in outcomes before the new treatment is introduced from the estimated jump in outcomes after the new treatment is introduced, and then divides by the jump in the new treatment (Nichols and Sorensen 2009, Beaule et al. 2009).

As with matching/reweighting, panel, and IV methods, we may be interested in the distribution of treatment effects, rather than a mean treatment effect. Frölich and Melly (2008) shows how to estimate quantiles of the response using a discontinuity design.

Chapter 6

Concluding Remarks

Tabulation of results from most data where the "treatment" variables (the causal factors) are not randomly assigned to units can produce very misleading inference. For example, comparing mean earnings of those who participate in a program (or go to college) to those who don't, does not tell you anything about the true effect of participation.

When experiments are infeasible, exploring data using quasi-experimental methods is the only reasonable option for estimating a causal effect. Quasi-experimental methods discussed in this book may even be preferred even when an experiment is feasible, particularly if a marginal treatment effect (the expected effect of a small increase in a treatment variable, for the subpopulation likely to receive the additional treatment) is of interest. However, the methods can suffer a number of severe problems when assumptions are violated, even weakly violated. For this reason, the details of implementation are frequently crucial, and a kind of cookbook or checklist for verifying essential assumptions are satisfied has been provided in this book for the interested (or budding) researcher. As the topics discussed continue to be active research areas, this cookbook should be taken merely as a starting point for further explorations of the applied econometric literature on the relevant subjects.

The focus here has been on getting a good estimate of a particular effect of some set of "treatments" which have already been identified, with a functional form that comes from theory. In many empirical papers, the focus is on identifying which of a set of possible explanatory variables seems to have most "explanatory power" for some outcome, and this approach is a search for causes of some known outcome, rather than a measurement of some causal effect. This "model building" approach is fundamentally different (from the approach here, where one starts

with a model and then goes to the data), but even if you take this approach, you would be wise to reconceptualize the model you wind up with in experimental terms and then look for a quasi-experimental estimator. I.e. ask yourself what the counterfactual is—what would a one-unit change in some variable mean in an experimental setting, holding other factors constant, and how might that effect be measured without an experiment?

On the other hand, ideas from the "model building" approach are equally crucial to add to the standard econometric approach adopted here of assuming a model a priori. An assumed model is never the truth, and a wise researcher tests assumptions not only with a few statistics, but by estimating different models and the same model in different samples (possibly subsamples of the estimation sample) to assess the robustness of results to small variations in design.

As a complement to statistical tests of required assumptions, looking for robustness in an estimator is crucial. We know assumptions are routinely violated, so we would like to know that the results are not too different if our assumptions are not quite right. For example, we may assume no spillover effects or homogeneous treatment effects, but we know these are not the case. One solution can be modifying the research design, but choosing an estimator that is reasonably robust to small violations is always a good idea; re-estimation in subsamples to assess robustness is a good practical rule of thumb.

The question of relative cost has not figured in this book, but if we seek to estimate the effects of various treatments relative to some baseline (or control condition, or placebo, or what have you), we should also care about relative cost. Reframing effects on outcome variables in terms of constant-cost interventions is a useful mechanism; for example if you evaluate three types of educational interventions that cost a hundred, two hundred, and three hundred dollars a day, then examining the relative impact of six days, three days, and two days (respectively) of each program seems a useful comparison, since each of those interventions costs six hundred dollars and the outcome gains per dollar are directly comparable.

There is also the possibility that the treatment has effects on quite different outcomes than you anticipate, so you should always be aware of other potential outcomes, and estimate those effects where possible. This feature is well known in the medical literature, but frequently forgotten in practical applications of statistics or policy. For example, in the 1950s, weak evidence suggested that giving diethylstilboestrol (DES) to pregnant women who had previously had miscarriages would increase the likelihood of a successful birth. This led to DES being prescribed to millions of pregnant women, and many of their children later

developed major health problems, including cancer, physical deformities, and infertility (Apfel and Fisher 1986).

I wish introductory econometrics were required of every journalist and bureaucrat. For all its warts, the framework of causal inference outlined in this book is clearly superior to the status quo, observable in most research read by bureaucrats and reported by journalists. I hope this book will help the reader become a more discerning consumer, and careful producer, of causal inferences in practice. Unfortunately, to really understand these methods, you have to slog through trying to estimate some effect yourself—luckily, you can count on help in that endeavor, should you undertake it. Anyone who has gotten this far in the book can count on a return email when posting a question on Statalist (see B.1.2), at least.

Appendix A

Some Math Topics

There are a lot of good introductions to these topics, some of which are freely available on the internet. But for those who like to have an introduction all in one book, I have included the following summary of math you should know to facilitate reading this book and the Stata manuals, or other similar books and articles.

A.1 Matrix Algebra

Matrix algebra is usually introduced as a way to solve a system of n linear equations in k unknowns. Here the unknowns are x_1 through x_k with coefficients in each of n equations.

$$\begin{aligned} b_{11}x_1 + b_{12}x_2 + \ldots + b_{1k}x_k &= a_1 \\ &\vdots \\ b_{i1}x_1 + b_{i2}x_2 + \ldots + b_{ik}x_k &= a_i \\ &\vdots \\ b_{n1}x_1 + b_{n2}x_2 + \ldots + b_{nk}x_k &= a_n \end{aligned}$$

This can be rewritten using matrix notation as

$$\begin{pmatrix} b_{11} & b_{12} & \ldots & b_{1k} \\ \vdots & & & \\ b_{i1} & b_{i2} & \ldots & b_{ik} \\ \vdots & & & \\ b_{n1} & b_{n2} & \ldots & b_{nk} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{pmatrix} = \begin{pmatrix} a_1 \\ \vdots \\ a_i \\ \vdots \\ a_n \end{pmatrix}$$

or

$$Bx = a$$

using the definition of matrix multiplication.

A.1.1 Matrix Multiplication

A vector is an ordered list of n numbers $(a_1, ..., a_n)$, also seen as a point in an n-dimensional space, or an arrow from the origin $(0, ..., 0)$ to that point. The inner product of two vectors is the sum of the product of each element in turn (the first element of the first vector times first element of the the second vector, plus the second element of the first vector times the second element of the second vector, and so on), which is a scalar (a single number). Obviously, each vector has to have the same number of elements (same dimension) in order to compute an inner product. The inner product of a vector with itself is its length squared (viewing the vector as an arrow).

The product of two matrices A and B is a matrix of inner products of row vectors of A and column vectors of B. Thus the first matrix A must have as many columns as there are rows in the second matrix B, in which case the matrices are said to be conformable. The the product of an $n \times k$ matrix and a $k \times m$ matrix is an $n \times m$ matrix. Note that not only is AB not generally the same as BA (unlike in scalar multiplication), but BA may not even be defined, if the dimensions of the matrices are such that B and A are not conformable.

Writing two n-dimensional vectors v and w as column matrices, we can write the inner product as $v'w$ (a 1 by 1 matrix, or scalar) where v' indicates the transpose of v (swap rows for columns). The outer product is vw' (an n by n matrix). Two vectors v and w are said to be orthogonal if their inner product is zero, i.e. $v'w = w'v = 0$. The inner product is also written $v \cdot w$ and called the dot product. The inner product is a measure of the extent of a projection of one vector on another (think of this as the shadow when the light source is right overhead with respect to the second vector). The inner product $v'w$ divided by the length of v (this ratio is also a scalar) times v is called the component of w along v, shown in figure A.1 (similarly, the component of w along v is the ratio of the inner product $v'w$ to the length of w, times w). Thus, the projection of v on an orthogonal vector u always has length zero, as it should, so the component is the zero vector.

Note that the sum of two vectors or matrices is the elementwise sum, or the sum of the corresponding elements, so matrices and vectors must have the same dimensions to be added. The geometric interpretation of vector addition can some-

times be useful (shown in figure A.1 as the diagonal of a parallelogram, or the result of moving the tail of one vector to the head of another).

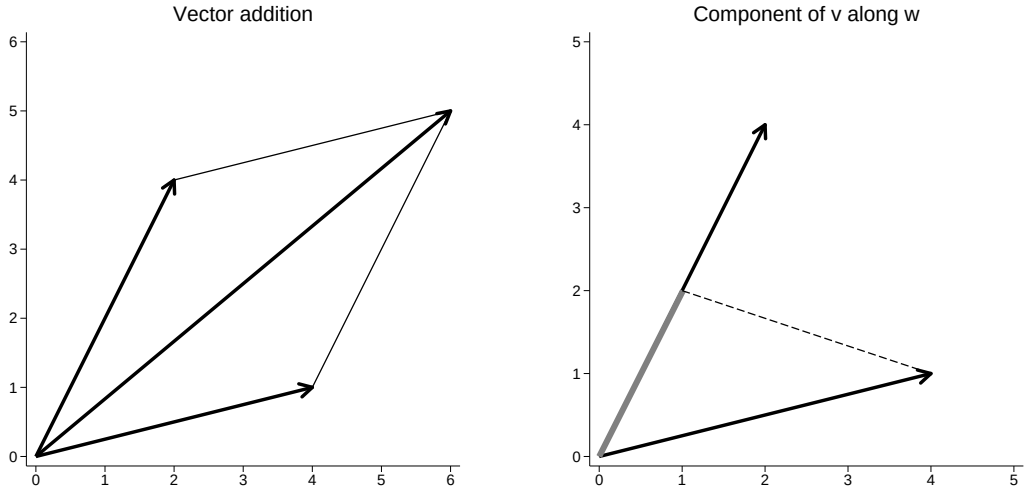

Figure A.1: Geometric properties of vectors.

Allowable operations in a system of equations (changing the order of equations, multiplying an equation by a constant, adding a multiple of one equation to another) are recast as operations on the matrix of coefficients B in Gauss-Jordan elimination, which results in a product of matrices premultiplying both sides of the equation. If $n = k$ and the matrix is invertible, one can write the solution as

$$x = B^{-1}a$$

which corresponds to performing all the operations required to make the matrix multiplying x the identity matrix. [1] If the matrix is not invertible (the inverse does not exist) then the system of linear equations may have no solutions, or infinitely many.

[1]The identity matrix I is a $n \times n$ matrix with ones along the diagonal and zeros elsewhere, which when multiplied by any conformable matrix A, gives A as the product, i.e. $IA = A$ and $AI = A$ for any matrix A.

A.1.2 Geometric interpretation and Rank

Forget about the system of linear equations for a moment. More generally, one can view a matrix B multiplying a vector of unknowns (or variables) x as a transformation of the space, rotating and stretching any vector plugged in for x is a particular way.

Alternatively, one can view the vector x multiplying B in $Bx = a$ as taking a weighted sum of the columns of B (a weighted sum where weights could be zero or negative, and need not sum to one, also called a linear combination). This is the most useful geometric view for regression purposes, since the columns of X in a regression model like $y = Xb$ are variables with some interpretation attached to each, and the elements of b are the "effect" of each variable on the outcome y.

Each column in a $n \times k$ matrix B is a n-dimensional vector; each row is a k-dimensional row vector. Linear combinations of the columns of a matrix B are said to span a space if any vector in the space[2] can be expressed as a linear combination of those column vectors. The set of vectors needed to span a space are called a basis of the space. The column rank of the matrix is the largest dimension of spaces spanned in this way; the column rank is the number of linearly independent column vectors in the matrix. The row rank is defined analogously, and the rank of a matrix is the minimum of those two ranks. The rank of a matrix is thus the dimension of a space that can be generated via linear combinations of the vectors in the matrix.

A.1.3 Inverses

A generalized inverse A^- for a matrix A satisfies $AA^-A = A$. A square matrix A (with as many columns as rows, corresponding to a system of k equations in k unknowns) has an inverse A^{-1} that satisfies $A^{-1}A = AA^{-1} = I$ if and only if it is of full rank (rank k). A generalized inverse need not be a unique solution to $AA^-A = A$, but the inverse is a unique solution to $A^{-1}A = AA^{-1} = I$. The inverse of A is one over the determinant $|A|$, a scalar, times the adjoint matrix $\mathrm{adj}(A)$ (neither term is defined here, as it would take several pages, and definitions are ubiquitous on the internet).

$$A^{-1} = \frac{1}{|A|}\mathrm{adj}(A)$$

[2] A vector space is a set of vectors with the operations of addition and scalar multiplication that is closed under those operations; that is, if v and w are in the set, then $u + v$ is in the set and ru is in the set for any real number r.

Evidently, the determinant $|A|$ is nonzero whenever the inverse is defined, i.e. whenever A has full rank.

A square matrix which has less than full rank, and therefore has no inverse, is said to be singular. A scalar r is said to be an eigenvalue of a matrix A if $A - rI$ is singular. An $n \times n$ square matrix A is singular if and only if zero is an eigenvalue of A. With r an eigenvalue of A, a nonzero vector v of dimension n that solves $(A - rI)v = 0$ is called an eigenvector of A corresponding to the eigenvalue r. The determinant of $(A - rI)$ is a polynomial of order n, which has n roots, or zeros, such that $|A - rI| = 0$. That polynomial is called the characteristic polynomial of A and can be solved to find eigenvalues and eigenvectors (note that roots need not be distinct, and some roots may be complex numbers). The sum of eigenvalues is equal to the trace of A (sum of diagonal entries) and the product of eigenvalues equals $|A|$ (so the role of a zero eigenvalue is clear here).

A symmetric matrix is a square matrix A such that $A' = A$. A symmetric matrix has all real eigenvalues, and eigenvectors corresponding to distinct eigenvalues are orthogonal. If there are repeated roots for the characteristic polynomial of A so there are non-distinct eigenvalues, there is still a nonsingular matrix M whose columns are eigenvectors of A such that all its columns are mutually orthogonal. Further, $M^{-1} = M'$ so M is said to be an orthogonal matrix. Also, the matrix $M^{-1}AM$ is a diagonal matrix with the corresponding eigenvalues down the diagonal.

A.1.4 Quadratic Forms

A quadratic form is a polynomial in many variables where every term has degree two, for example

$$f(x_1, x_2) = ax_1^2 + bx_1x_2 + cx_2^2$$

with two variables x_1 and x_2 and constant coefficients a, b, and c, or

$$f(x_1, x_2, x_3) = a_1x_1^2 + a_2x_1x_2 + a_3x_2^2 + a_4x_1x_3 + a_5x_2x_3 + a_6x_3^2$$

with three variables. Every quadratic form can be written as a matrix product

$$f(x) = x'Ax$$

and without loss of generality we can let A be symmetric. That is, a quadratic form

$$f(x_1, x_2) = ax_1^2 + bx_1x_2 + cx_2^2$$

can be represented as the matrix product

$$f(x_1, x_2) = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \begin{pmatrix} a & b/3 \\ 2b/3 & c \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

or

$$f(x_1, x_2) = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \begin{pmatrix} a & b/2 \\ b/2 & c \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

but it is much more convenient to write it so that the matrix of coefficients A is symmetric.

A quadratic form is called positive definite if $f(x) > 0$ for all x, and any matrix A that generates that quadratic form is also called positive definite. A quadratic form is called negative definite if $f(x) < 0$ for all x, and likewise for associated matrices. A quadratic form is called positive semidefinite if $f(x) \geq 0$ for all x and negative semidefinite if $f(x) \leq 0$ for all x. A quadratic form is called indefinite if $f(x) > 0$ for some x and $f(x) < 0$ for some other x. These properties correspond to some common shapes of three-dimensional surfaces that have special significance, in that these shapes capture most of the important shapes an arbitrary surface can take on in a neighborhood of some point (i.e. these are five important classes of curvature near a point).

The eigenvalues of symmetric matrices A correspond to these properties. That is, a symmetric matrix A is positive definite if and only if all the eigenvalues of A are strictly positive. A symmetric matrix A is negative definite if and only if all the eigenvalues of A are strictly negative, a symmetric matrix A is positive semidefinite if and only if all the eigenvalues of A are weakly positive ($r \geq 0 \forall r$), and negative semidefinite if and only if all the eigenvalues of A are weakly negative ($r \leq 0 \forall r$). A symmetric matrix A is indefinite if and only if some eigenvalues of A are positive and some negative.

A.2 Calculus

The calculus is a handy tool, nothing more, and you need not understand all of its intricacies to be a good researcher, but a cursory understanding of the major concepts is absolutely essential.

A.2.1 Derivatives and Gradients

The derivative of a function captures the slope of the function at each point, or equivalently the slope of a tangent linear function. In figure A.2, the derivative at $x = 2$ is the slope of the curve at that point or the slope of the tangent line at that point, which is four; at $x = 3$ the slope is two and at $x = 3$ the slope is zero (the tangent is horizontal); at higher values of x the slope would be negative.

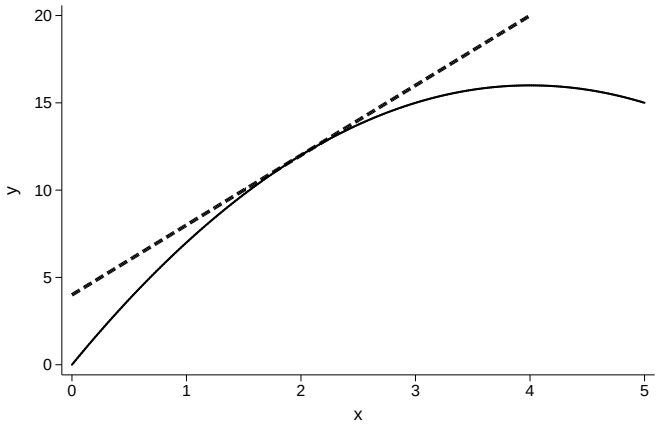

Figure A.2: The derivative as the slope of a linear approximation.

For a real-valued function of a scalar $F(x)$ the derivative may be written $f(x)$ or $F'(x)$ or dF/dx or $\partial F/\partial x$ or various other ways. One useful way is to write $D_x F(x)$ meaning the derivative with respect to x of $F(x)$ because the derivative can be thought of as a linear operator:

$$D_x \left[aF(x) + bG(x)\right] = a \left[D_x F(x)\right] + b \left[D_x G(x)\right]$$

where F and G are functions and a and b are constants, and one can simply distribute the D_x operator across the sum. The derivative of a function is itself a function, so one can take the derivative of a derivative, and the result is the second derivative. One can compute third and higher-order derivatives likewise.

For a real-valued function $F(x)$ of a vector x, the derivative is a set of functions, called the gradient, measuring the slope in any of the various directions one might shift the vector x. For example, if the vector x is two dimensional and we write the vector x as an ordered pair (x_1, x_2), we can imagine moving in the direction of x_1, e.g. from (x_1, x_2) to $(x_1 + c, x_2)$, or in the direction of x_2, e.g. from

(x_1, x_2) to $(x_1, x_2 + c)$. Each of those directions defines a partial derivative, and there are as many of them as there are dimensions of the vector x. In other words, the gradient of the function $F(x)$ of a vector x has as many dimensions as the vector x, and we can say the gradient function is vector-valued. We write

$$\nabla F(x) = D_x F(x) = \frac{\partial F(x)}{\partial x} = \begin{bmatrix} \frac{\partial F(x)}{\partial x_1} \\ \vdots \\ \frac{\partial F(x)}{\partial x_n} \end{bmatrix}$$

and note that the derivative of a vector-valued function is a matrix. Thus the second derivative of a real-valued function $F(x)$ of an n-dimensional vector x is an n by n matrix of "second partials" or derivatives in each of the n possible directions associated with changes to x of each of the (partial) derivatives in each of the n possible directions associated with changes to x.

Some simple rules for derivatives are helpful to bear in mind at all times:

$$D_x a x^b = a b x^{b-1}$$

$$D_x \ln x = \frac{1}{x}$$

$$D_x \exp x = \exp x$$

The chain rule gives the derivative of $f(g(x))$ for a function f of the value g of a function $g(x)$ as $\frac{\partial f}{\partial g}\frac{\partial g(x)}{\partial x}$, and the product rule gives the derivative of $f(x)g(x)$ for a product of two functions f and g of x as $\frac{\partial f(x)}{\partial x}g(x) + \frac{\partial g(x)}{\partial x}f(x)$.

Not all functions have well-defined gradients, but most models make extensive use of those that do, called "differentiable" functions. In fact, most models make use of functions that have continuous second and higher derivatives, called "smooth" functions. A slightly smaller class of "analytic" functions has a convergent[3] Taylor series, meaning that the function $F(x)$ can be represented to arbitrarily good precision by a well-defined series with terms that are polynomial in x and have coefficients given by the derivatives of $F(x)$.

[3] An analytic function has a convergent Taylor series in the complex field. As Apostol (1974) points out, a function may have an apparently convergent Taylor series in the reals, and the series may converge to a different function.

A.2.2 Taylor Series

The first-order Taylor expansion of a function $f(x)$ around $x = x_0$ is written

$$f(x_0 + h) = f(x_0) + \frac{\partial f}{\partial x}(x_0)h + R_2$$

where there is a number a between zero and h such that $\frac{1}{2}\left[\frac{\partial^2 f}{\partial x^2}(x_0 + a)\right] h^2 = R_2$. Or it can be written

$$f(x) = f(x_0) + \frac{\partial f}{\partial x}(x_0)(x - x_0) + R_2$$

letting $x = x_0 + h$ and we can see how it is linear in x and approximates $f(x)$ in a neighborhood of x_0.

The ratio

$$\frac{R_2}{h^2}$$

approaches zero as h approaches zero, so we say that R_2 has order less than h^2, or write $R_2 = o(h^2)$ (called "little oh" notation), meaning that R_2 converges to zero faster than h^2 as h approaches zero. This means that the first-order Taylor approximation is very good for small h (i.e. close to $x = x_0$), e.g. if $h = 0.001$ then the error is about one part in a million.

The second-order Taylor expansion of a function is an even better approximation:

$$f(x_0 + h) = f(x_0) + \left[\frac{\partial f}{\partial x}(x_0)\right] h + \frac{1}{2}h' \left[\frac{\partial^2 f}{\partial x^2}(x_0)\right] h + R_3$$

because the error of approximation R_3 is of order $o(h^3)$, meaning that if $h = 0.001$ then the error is about one part in a billion.

We could also write the second-order Taylor approximation as:

$$f(x) = f(x_0) + \left[\frac{\partial f}{\partial x}(x_0)\right] (x - x_0) + \frac{1}{2}(x - x_0)' \left[\frac{\partial^2 f}{\partial x^2}(x_0)\right] (x - x_0) + R_3$$

or with one variable in x write:

$$f(x) = f(x_0) + \left[\frac{\partial f}{\partial x}(x_0)\right] (x - x_0) + \frac{1}{2}\left[\frac{\partial^2 f}{\partial x^2}(x_0)\right] (x - x_0)^2 + R_3$$

You will rarely see Taylor approximations beyond the second order because the second order approximation captures the local curvature around x_0. In fact,

the first order approximation is enough for most purposes because the error in that case of order $h^2 = (x - x_0)^2$ not only gets arbitrarily close to zero but it is effectively flat, meaning further improvements are negligible, as x gets close to x_0.

A.2.3 Optimization

If you want to find the maximum or minimum (both sometimes referred to as an extremum, or optimum) of a function, one useful method is to note that its derivative is zero at that point, as at the point $x = 4$ in figure A.2. If it were not, we could improve on the value of the function by moving a tiny bit in a direction in which the derivative is positive. In fact, the derivative may be zero at other points as well, but extremums for smooth functions can often be found fairly easy by setting the derivative equal to zero and solving for x. For a strictly convex function, such as a quadratic (or a positive definite quadratic form), the derivative is zero only at the minimum, and similarly for a strictly concave function the derivative is zero only at the maximum. In these cases, finding the optimum is fairly easy.

More generally, we may find many points that satisfy the first order condition that the derivative is zero at that point. We can then turn to a second order condition that the matrix of second partials, or Hessian, must be negative semidefinite at a local maximum and must be positive semidefinite at a local minimum. If it is indefinite at a point where the first order condition is satisfied, the point is called an inflection point, and is neither a local maximum nor minimum. If the Hessian is negative definite the point is guaranteed to be a local maximum, and if it is positive definite the point is guaranteed to be a local minimum, but if it is semidefinite, further investigation is required.

Constrained optimization where we optimize the function subject to an inequality $g(x) \leq 0$ is similar, except that one of two things must be true: either the constraint $g(x) = 0$ must be satisfied with equality and the function optimized over the set where $g(x) = 0$, or there is an optimum in the interior where $g(x) < 0$ and the constraint does not bind. This increases the number of potential optimums we have to compare. For more, see Simon and Blume (1994), e.g. Theorem 19.12 (though the proof using Farkas' Lemma is unfortunately omitted).

A.2.4 Integrals

The integral is the area under a curve, for a nonnegative function of a scalar variable (area under a nonnegative surface is a multiple integral), and the antiderivative in the sense that if $f(x)$ if the derivative of $F(x)$ then $F(x)$ is the integral of $f(x)$. The integral is usually defined as the limit of Riemann sums, rectangular areas between a curve and the axis (for a nonnegative function—for a function that dips below the axis, we subtract the areas where the curve dips below the axis) with height given by some value of the function in a sequence of intervals, and we let the number of intervals increase without bound by subdividing each interval, and let the width of the largest interval decrease toward zero, to get an arbitrarily good approximation to the area under the curve. If the limit of such a sequence exists, it is called the Riemann integral.

A more general and related concept is the Lebesgue integral, constructed using sets of measure zero and step functions. A subset T of an interval S on the real line has measure zero, if, for any ϵ, T can be covered by a countable collection of intervals such that the sum of the lengths of the intervals is less than ϵ. A property which holds everywhere on S except on a set of measure zero is said to hold almost everywhere on S, and a set U which is S less a set of measure zero is dense on S (e.g. the set of irrational numbers is dense on the unit interval). A step function is a function which is constant on each of k intervals:

$$s(x) = c_k \, \forall x \in (x_{k-1}, x_k) \forall k = 1, \ldots, k$$

and its integral is defined as the area under its steps:

$$\int_a^b s(x)dx = \sum_{k=1}^{n} c_k(x_k - x_{k-1}).$$

An upper function is the limit of a sequence of increasing step functions defined on some interval $S = (a, b)$ on the real line, where a sequence of increasing step functions satisfies $s_n \leq s_{n+1}$ everywhere on S, for all n. We write $s_n(x) \nearrow s(x)$ to mean the sequence of increasing step functions has a limit $s(x)$. The integral of an upper function is:

$$\int_a^b s(x) = \lim_{n\to\infty} \int_a^b s_n(x)$$

The Lebesgue integral for a function $f = u - v$ where u and v are both upper functions is given by $\int f = \int u - \int v$. In general, $\int(u + v) = \int u + \int v$ and

$\int cu = c \int u$ for any real number c and Lebesgue-integrable functions u and v. The Lebesgue integral equals the Riemann integral when both exist, but some functions are Lebesgue-integrable when the Riemann integral does not exist. See Apostol (1974) for more detail.

Integration by parts is a trick, inverting the product rule for derivative, often helpful for figuring out difficult integrals. If we write the product rule as

$$D_x f(x)g(x) = \frac{\partial f(x)}{\partial x} g(x) + \frac{\partial g(x)}{\partial x} f(x)$$

and integrate both sides, we get

$$f(x)g(x) = \int \frac{\partial f(x)}{\partial x} g(x)dx + \int \frac{\partial g(x)}{\partial x} f(x)dx$$

which we rewrite

$$\int \frac{\partial f(x)}{\partial x} g(x)dx = f(x)g(x) - \int \frac{\partial g(x)}{\partial x} f(x)dx$$

and this can be used to turn some very difficult integrals into easier problems.

A.2.5 Derivatives involving Integrals

The derivative of a definite integral is given by

$$\frac{\partial}{\partial y} \int_{a(y)}^{b(y)} f(x,y)dx = \int_{a(y)}^{b(y)} \frac{\partial f}{\partial y} dx + f(b(y),y)\frac{\partial b}{\partial y} - f(a(y),y)\frac{\partial a}{\partial y}$$

when the limits of integration are functions of y. If the limits of integration are not functions of y, then the last two terms are zero:

$$\frac{\partial}{\partial y} \int_a^b f(x,y)dx = \int_a^b \frac{\partial f}{\partial y} dx$$

See also Apostol (1974), page 167 and page 283.

The rule is known as Leibniz's rule, or as differentiation under the integral sign. It can be used to evaluate tricky definite integrals, e.g.

$$\int_0^\pi \ln(1 - 2\alpha\cos(x) + \alpha^2)dx = 2\pi \ln|\alpha| \quad \forall|\alpha| > 1.$$

Feynman (1997) described seeing this use of the rule in Woods (1926) and remarked "because I was self-taught using that book, I had peculiar methods for doing integrals," and "used that one damn tool again and again." (The memoir is quoted at the mathworld.wolfram.com website entry on Leibniz's rule, but I remember this Feynman quote from the 1985 edition, which I read in tenth grade; there is clearly something very compelling about Feynman's story.)

A.2.6 Optimal Control

In an optimal control problem, we want to choose c at each point in time t to optimize a function of the form

$$\max_{c(t)} \int_0^T u(c, X, t)dt$$

with X is a "stock" that depends on the control variable c, i.e.

$$\frac{\partial X}{\partial t} = f(c, X, t)$$

and we know $X(0) = X_0$ and $X(T) = X_T$ (the level of the stock at the beginning and end of the problem). Here u represents a "flow" value, and we want to maximize (or minimize) the sum of flows over all t in $[0, T]$. Rather than choosing c at one point in time (or a finite number of values c_i in a vector c), we want to maximize over all functions c of t.

To solve this optimal control problem we set up a Hamiltonian:

$$H = u(c, X, t) + \mu f(c, X, t)$$

where μ is a function of time that captures the marginal value of an additional unit of the stock X. The necessary conditions for an optimal solution, known as Pontryagin's maximum principle, are:

$$\begin{aligned} \frac{\partial H}{\partial c} &= 0 \\ \frac{\partial H}{\partial \mu} &= \frac{\partial X}{\partial t} \\ \frac{\partial H}{\partial X} &= -\frac{\partial \mu}{\partial t} \end{aligned}$$

at each point in time.

For example, if we

$$\max_{c(t)} \int_0^T e^{-\rho t} \ln(c)dt$$

subject to

$$\frac{\partial A}{\partial t} = rA - c$$

and $A(0) = 1$ and $A(T) = 0$, this is the problem of maximizing log consumption (c) over a finite lifetime (until T) with no bequest (assets are A). The Hamiltonian is:

$$H = e^{-\rho t} \ln(c) + \mu(rA - c)$$

and the necessary conditions for a maximum are:

$$\begin{aligned} \frac{e^{-\rho t}}{c} - \mu &= 0 \\ \frac{\partial A}{\partial t} &= rA - c \\ r\mu &= -\frac{\partial \mu}{\partial t} \end{aligned}$$

at each point in time. The third equation is an ordinary differential equation with the solution $\mu(t) = \mu(0)e^{-rt}$ and we can solve the first equations for c in terms of μ then plug into the second equation:

$$\begin{aligned} c &= \frac{e^{-\rho t}}{\mu} = e^{(\rho+r)t}\mu_0^{-1} \\ \frac{\partial A}{\partial t} &= rA - c = rA - e^{(r-\rho)t}\mu_0^{-1} \end{aligned}$$

or $\frac{\partial A}{\partial t} - rA = -e^{(r-\rho)t}\mu_0^{-1}$
. This is an ordinary differential equation with the solution $A(t) = e^{(r-\rho)t}\left(\rho\mu_0\right)^{-1} + Be^{rt}$ where B is a constant which can be determined since we know $A(0) = 1$ and $A(T) = 0$. The optimal consumption path is then

$$c(t) = e^{(r-\rho)t}\left(\frac{\rho}{1 - e^{-\rho T}}\right)$$

where r is the interest rate and ρ is the discount rate and the term in parentheses is a constant. If $r > \rho$ optimal consumption rises exponentially over time, but $r < \rho$ implies optimal consumption falls exponentially over time.

We might also want to optimize over all future time t like so:

$$\max_{c(t)} \int_{t_0}^{\infty} u(c, X, t)dt$$

and all the same necessary conditions apply. An important class is the "autonomous" problem

$$\max_{c(t)} \int_{t_0}^{\infty} e^{-\rho t}u(c, X)dt$$

with exponential discounting and no t in the flow value u, which makes the problem simpler. See Léonard and Long (1992) for extensions and many useful examples.

The stochastic optimal control problem, for example where the state variables X include random components or are observed with error, is a much harder problem with no general solution approach. The sufficient condition for an optimum, the Hamilton-Jacobi-Bellman equation, is a partial differential equation that is difficult to solve even with numerical methods in some cases, and is not guaranteed to exist. Using symmetry (Boyd III 1990) or local approximations (Crespo and Sun 2002) may be an easier path to the global optima in many cases.

A.3 Probability

Probabilities are real-world concepts but also mathematical artifacts, so it can be a bit tricky moving back and forth between the two worlds. A probability of some event being observed is nonnegative, and the probability of some event and its complement sum to one, i.e. $P(A) + P(\overline{A}) = 1$, so either A happens or it doesn't. For a partition A_i of the space of possible events U such that $\bigcup_i A_i = U$ and $\cap_i A_i = \emptyset$, the probabilities of each event sum to one, i.e. $\sum_i P(A_i) = 1$.

The probability that A happens and B happens is written $P(A \wedge B)$ (or $P(A \cap B)$) and read "the probability of A and B." The probability that A happens or B happens is written $P(A \vee B)$ and read "the probability of A or B." These are usefully related via the equation $P(A \vee B) = P(A) + P(B) - P(A \wedge B)$ or the Venn diagram, for example in figure A.3. Note that the probability of a red jack is 2/52 and the probability of a face card with hearts is 3/52, but the probability of either is 4/52=2/52+3/52-1/52, subtracting off the "double counted" jack of hearts.

The probability of an event A conditional on another event B is written $P(A|B)$ and equals $P(A \wedge B)/P(B)$ assuming $P(B)$ is nonzero. The probability of a red jack is 2/52 but if we condition on having drawn a face card with hearts, the probability is 1/3 (1/52 divided by 3/52). If two events A and B are independent, $P(A \wedge B) = P(A)P(B)$, and $P(A|B) = P(A)$. The probability of rolling a six and then another six with a fair die is 1/6 times 1/6, because the two events are independent. Bayes Rule comes straight from the definition of conditional probability:

$$\begin{aligned} P(A|B) &= P(A \wedge B)/P(B) \\ P(B|A) &= \frac{P(B \wedge A)}{P(A)} = \frac{P(A \wedge B)}{P(A)} = \frac{P(A|B)P(B)}{P(A)} \end{aligned}$$

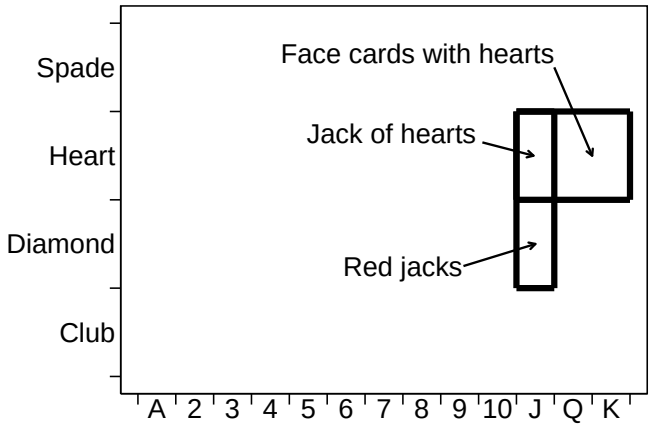

Figure A.3: Probabilities of events as areas on a Venn diagram.

These kinds of discrete problems often involve counting numbers of arrangements using combinations and permutations. Many cases involve a random variable, which is a function mapping events to real numbers (for example, mapping the draw of an Ace to 1, a numbered card to its number, and a jack, queen, or king, to 11, 12, and 13, respectively). Then we can talk about the probability associated with numbers, with the numbers ordered on a line. The calculated probability distributions, with $P(A_i)$ given for any event A_i, are often represented via a histogram, or bar graph of probabilities versus possible values of the random variable. This is sometimes called a probability distribution function (pdf) though for a discrete variable, probability mass function would be less ambiguous.

For example, figure A.4 shows the histogram labeled "pdf" for the number of heads that come up when flipping a fair coin 10 times (called repeated Bernoulli trials with probability one half). The cumulative probability distribution function (cdf) is the probability of observing a value less than or equal to x, and is the integral of the pdf (which is why capital letters are often used to denote a cumulative distribution function and lower case letters a probability distribution function).

For most applications of probability theory to data, we are concerned with the chance that some number falls in a range, where any real number (or vector of real numbers) may be drawn. I.e. the random variable maps to real numbers (or vectors), not a finite set of possible outcomes.

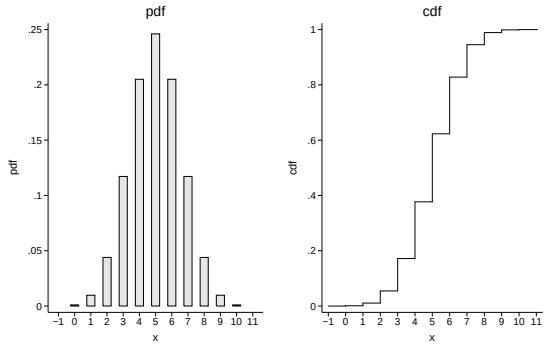

Figure A.4: Histogram (pdf) and distribution function (cdf) of a binomial distribution.

A.3.1 Density and Distribution

If a random variable X maps events to real numbers, then we define the cumulative distribution function (cdf) as

$$F(x) = Pr(X \leq x)$$

which is a non-decreasing, right-continuous function with limit 0 as x goes to negative infinity and 1 as x goes to positive infinity. If $F(x)$ is absolutely continuous, i.e. its derivative exists and integrating the derivative gives us the cdf back again, then the random variable X is said to have a probability density function (pdf) or simply density defined by

$$f(x) = \frac{\partial F}{\partial x}(x)$$

Any event is mapped to a subset of real numbers A and we can calculate the probability of the event as

$$Pr(A) = \int_{x \in A} dF(x) = \int_A dF(x)$$

given by the Lebesgue integral. If the density exists, we can write this as

$$Pr(A) = \int_{x \in A} f(x)dx = \int_A f(x)dx.$$

In most cases we can implicitly define quantiles x_q such that $F(x_q) = q$, for example the median $x_{0.5}$ with $q = 0.5$ such that the probability of observing $X > x_{0.5}$ or observing $X < x_{0.5}$ is one half.

A.3.2 Mean and Conditional Mean

The mean, or expectation, of a discrete random variable is just the weighted sum of its values, with weights given by probabilities.

$$E(X) = \sum xPr(X = x)$$

and for a continuous random variable, the analogous weighted sum

$$E(X) = \int xdF(x)$$

and therefore the expectation operator is linear. That is,

$$E(aX + b) = aE(X) + b$$

for any constants a and b. However, the expectation cannot in general pass through nonlinear functions, so $E(e^X) \neq e^(E(x)$ and $E(XY) \neq E(X)E(Y)$ (though this last is true if X and Y are independent).

For discrete random variables X and Y, the mean of Y conditional on $X = x$ is given by using conditional probabilities as weights:

$$E(Y|X) = \sum yPr(Y = y|X = x) = \sum y\frac{Pr(Y = y \wedge X = x)}{Pr(X = x)}$$

but for a continuous random variable, the probability of observing $X = x$ is often zero, so we cannot define the conditional probability in quite the same way. The general definition given by (Kolmogorov 1933) is far from intuitive; in fact, Rényi (1955) derives probability theory starting from conditional probabilities as primitives.

Ignoring these problems, we can say that for continuous random variables X and Y, the mean of Y conditional on $X = x$ is given by the analogous weighted sum

$$E(Y|X) = \int ydF(y|X = x)$$

where $F(y|X = x)$ is the conditional distribution function (the conditional distribution defines conditional quantiles as above). If the relevant densities exist, we write

$$E(Y|X) = \int ydF(y|X = x) = \int y\frac{f(x, y)}{f_X(x)}dy$$

where $f_X(x)$ is the marginal density integrating out y i.e. $F_X(x) = Pr(X \leq x)$ and $f_X(x) = D_xF_X(x)$.

If we want to calculate a conditional mean for one random variable, the formula is even simpler. We simply integrate over the region given by the event and divide by the probability of the event we are conditioning on. For example, if we want to know the mean of a standard normal $Z \sim N(0,1)$ when we know that $Z > a$, we can calculate

$$\frac{1}{1-\Phi(a)} \int_a^\infty Z\phi(Z)dZ$$

where $\phi(Z)$ is the standard normal density ($f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$ is the normal density with parameters μ and σ, and the standard normal has $\mu = 0$ and $\sigma = 1$). Observations on Z given that $Z > a$ have a "truncated normal distribution" with expectation $\frac{\phi(a)}{1-\Phi(a)}$ (so for example, the mean of Z conditional on Z being greater than 1.96 is 2.3378).

More generally, the conditional distribution of a random variable X given that $X > a$ and $X \leq b$ is given by

$$F(x|a < X \leq b) = \begin{cases} 0 & \forall x \leq a \\ \frac{F(x)-F(a)}{F(b)-F(a)} & \forall a < X \leq b \\ 1 & \forall x > b \end{cases}$$

and the mean of a function $u(X)$ given that $X > a$ and $X \leq b$ is given by

$$E(u(X)|a < X \leq b) = \frac{\int_a^b u(x)dF(x)}{F(b) - F(a)}$$

(the conditional distribution again defines conditional quantiles). If X is normally distributed with parameters μ and σ, and we know $X > a$ and $X \leq b$ then the mean X is

$$E(X|a < X \leq b) = \mu - \sigma \frac{\phi\left(\frac{b-\mu}{\sigma}\right) - \phi\left(\frac{a-\mu}{\sigma}\right)}{\Phi\left(\frac{b-\mu}{\sigma}\right) - \Phi\left(\frac{a-\mu}{\sigma}\right)}$$

A.3.3 Variance and Higher Moments

The variance of a random variable x is defined as

$$\text{Var}(x) = E\left[(x - E[x])^2\right] = E\left[x^2\right] - (E[x])^2$$

and higher central moments are defined similarly: the pth central moment is $\mu_k(x) = E\left[(x - E[x])^k\right]$ and satisfies $\mu_k(ax + b) = a^k\mu_k(x)$ for constants a

and b. The variance measures the "spread" around the mean, which is about as unhelpful a definition as saying it measures the variability around the mean, i.e. the variance. It helps to picture a few "bell curves" (normal distributions) that are progressively flatter and wider, but this is very deceptive, as many interesting distributions are not nice bell curves.

The variance of a binary "dummy" variable has a particularly easy form. If $x = 1$ with probability p and $x = 0$ with probability $(1 - p)$, then its mean is p and its variance is $(1 - p)$ times p, plus $(0 - p)$ times $(1 - p)$, so $\mathrm{Var}(x) = p(1 - p)$. A single draw (sample size 1) of a binary "dummy" variable is called a Bernoulli trial, and the mean of a sample of n independent draws has the binomial distribution with mean np and variance $np(1 - p)$. This makes hypothesis testing of a single proportion easy, since a null hypothesis about the mean (the proportion with $x = 1$) also pins down the variance.

A.3.4 Asymptotics

Only in the very simplest cases can we make statements about the probability that some statistic fall in some range for a dataset of the type we are working with; for example, if we have a sample of n observations on a normally distributed random variable with an unknown mean and variance, we can make precise statements about the probability the sample mean would fall outside a range given a null hypothesis about the true mean. More generally, we use an asymptotic result for "large samples" by taking the limiting behavior as the sample size n increases without bound (approaches infinity) as a reasonable approximation of the distribution of our statistic for samples of size n when n is large. There are a variety of types of asymptotic results, notably convergence in probability and almost sure convergence.

The probability limit (plim) of a sequence of estimators $\widehat{\theta}_n$ with sample size n increasing without bound is θ if

$$\lim_{n\to\infty} \Pr\left(\left|\widehat{\theta}_n - \theta\right| \geq \varepsilon\right) = 0$$

for any $\varepsilon > 0$. Or, in other words, for any $e > 0$ and any $d > 0$, there exists a value ν such that $\Pr\left(\left|\widehat{\theta}_n - \theta\right| > e\right) < d$ for all $n \geq \nu$. In that case, we write

$$\mathrm{plim}\, \widehat{\theta}_n = \theta$$

and say that $\widehat{\theta}_n$ is a consistent estimator of θ. We can also say that $\widehat{\theta}_n$ converges in probability to θ, or write

$$\widehat{\theta}_n \xrightarrow[p]{} \theta$$

to mean the same thing.

Given a sequence of constants a_n if we can show that

$$\frac{\widehat{\theta}_n}{a_n} \xrightarrow[p]{} 0$$

then we say that $\widehat{\theta}_n$ is of order less than a_n or that $\widehat{\theta}_n = o_p(a_n)$ (borrowing the "little oh" notation from calculus, but subscripting by p to mean "order in probability"). When we can prove the convergence of an estimator divided by the square root of the sample size n, we say we have a root-n consistent estimator, or $\sqrt{n}$-consistent estimator. For example, if we have an estimator $\widehat{\beta}_n$ which converges to β at the rate $\sqrt{n}$, we can write $\widehat{\theta}_n = \sqrt{n}(\widehat{\beta}_n - \beta)$ and

$$\widehat{\theta}_n = o_p(n^{-\frac{1}{2}})$$

i.e. even multiplied by the square root of the sample size, which is going to infinity, the difference between the estimator and its limit goes to zero, which implies that the series converges very quickly. Evidently, a $n^{\frac{1}{3}}$) consistent estimator would converge more slowly and a $n^{\frac{2}{3}}$) consistent estimator would converge more quickly.

The Weak Law Of Large Numbers is a famous application of convergence in probability for the most common estimator, the sample mean $\bar{x}_n = n^{-1}\sum_{i=1}^{n} x_i$ of a sample of n observations on a random variable X that are identically and independently distributed. If we let μ be the population mean of X, then $\bar{x}_n \xrightarrow[p]{} \mu$.

A related concept is convergence in distribution. Suppose that

$$F_n(v) = Pr(\widehat{\theta}_n \leq v)$$

is the distribution function for our estimator in a sequence of increasing sample sizes, and $F(u)$ is a distribution function for some random variable θ. We say that $\widehat{\theta}_n$ converges in distribution to θ if

$$\lim_{n\to\infty} F_n(a) = F(a)$$

for any real number a where $F(v)$ is continuous, and we write $F_n(x) \xrightarrow[d]{} F(x)$. Convergence in probability implies convergence in distribution.

If we let μ be the mean of x, and σ its variance, then for the sample mean $\bar{x} = n^{-1}\sum_{i=1}^{n} x_i$ the Central Limit Theorem states:

$$\sqrt{n}\left(\widehat{\theta}_n\right) = \sqrt{n}\left[\frac{\bar{x} - \mu}{\sigma}\right] \xrightarrow[d]{} N(0,1)$$

The nice feature of convergence in probability is that it passes through continuous functions, so if $x_n \xrightarrow[p]{} x$ and the function $g(x)$ is continuous, then $g(x_n) \xrightarrow[p]{} g(x)$ or:

$$\operatorname{plim} g\left(\widehat{\theta}_n\right) = g\left(\operatorname{plim} \widehat{\theta}_n\right)$$

whereas the expectation operator can only be passed though linear functions.

The delta method for calculating standard errors of a function of random variables uses a related result. If $\sqrt{n}\left(\widehat{\theta}_n - \theta_0\right) \xrightarrow[d]{} N(0, \Sigma)$ then

$$\sqrt{n}\left(g(\widehat{\theta}_n) - g(\theta_0)\right) \xrightarrow[d]{} N(0, \Delta)$$

where

$$\Delta = \left[\frac{\partial g}{\partial \theta}(\theta_0)\right] \Sigma \left[\frac{\partial g}{\partial \theta}(\theta_0)\right]'$$

Asymptotic equivalence is a way of establishing convergence in distribution. If there is a sequence b_n such that $\left(\widehat{\theta}_n - b_n\right) \xrightarrow[p]{} 0$ and $\left(\widehat{\theta}_n - b_n\right) \xrightarrow[d]{} b$ for some random variable b, then $\widehat{\theta}_n) \xrightarrow[d]{} b$ and we say that $\widehat{\theta}_n$ and b_n are asymptotically equivalent (Rao 1973; page 123).

Mean square convergence is a useful trick to prove other kinds of convergence. A sequence of estimators $\widehat{\theta}_n$ converges in mean square to a random variable θ if

$$\lim_{n \to \infty} E\left[\left(\widehat{\theta}_n - \theta\right)^2\right] = 0$$

and mean square convergence implies convergence in probability, provided the variance of $\widehat{\theta}_n$ exists (called a "finite second moment" condition).

Almost sure convergence is a much stronger condition; as White (1984; page 16) says, "sequences that converge almost surely can be manipulated in almost exactly the same ways as nonrandom sequences." A sequence of estimators $\widehat{\theta}_n$ converges almost surely to a vector θ if

$$Pr\left(\lim_{n \to \infty} \widehat{\theta}_n = \theta\right) = 1$$

so the estimator converges almost everywhere on the space of possible realizations of the state of the world (the set of convergent sequences is dense on the space of possible sequences). In this case, we say that $\widehat{\theta}_n$ is strongly consistent for θ and we write

$$\widehat{\theta}_n \xrightarrow[a.s.]{} \theta$$

(for example, the Strong Law Of Large Numbers says that $\bar{x} \xrightarrow[a.s.]{} \mu$). Almost sure convergence implies convergence in probability, which in turn implies convergence in distribution, but almost sure convergence does not imply convergence in mean square (nor does convergence in mean square imply almost sure convergence).

Appendix B

Data and Stata

In the real world, we always have finite data, and the computer program we use to construct estimates can have real consequences for the quality of our estimates. I assert without proof that Stata is the best choice.

B.1 Stata Basics

If you're used to some alternative software for statistics, my condolences—you have been missing out. There are some important differences to note up front. Stata commands are almost always on a single line, perhaps with options after a comma, and can be issued interactively at a command line or in a file of commands. A "program" or list of commands in some other software is a Stata do file (just a plain text file with the .do extension by default). A SAS "macro" is a Stata program, more or less. Also, the data structure is a bit different, and handling data is very different, primarily because Stata has all the data in memory at one time, whereas some other programs handle one line of data at a time. The main difference between SAS and Stata is that you can see inside much of Stata's code and get good help very easily (both from the company and from other users). Also, Stata is not an acronym, so it is not spelled in all caps. For a couple of useful introductions to Stata, see http://www.ats.ucla.edu/STAT/stata/ or http://www.cpc.unc.edu/services/computer/presentations/statatutorial or various other university help sites.

B.1.1 Getting Started

Stata is an extremely flexible program for working with data, and the tradeoff for that flexibility is that it is not the simplest program in the world to use, but it is easier than most. There are few other programs that have comparable strengths: SAS is slightly less flexible, and R is slightly more flexible than Stata, but both of these are much harder to use for beginners, and there are easier programs for beginners to use, but are not useful for doing statistical or data manipulation work past a certain point.

As an example, consider running a basic OLS regression in Stata: you just type `regress y x` in the Command window[1] to regress y on x and get a variety of related statistics. It's not that easy in any other program. To get my favorite user-written addition, just type `ssc install estout`, and you've just added the capability to create formatted tables of regression output (using `esttab`, with the `rtf` option for a table that opens in Microsoft Word—if you are using LaTeX, I'm guessing you can navigate the help yourself).

Assume you've just started Stata. There are four windows open, the big Results window, and below it the small Command window where you type commands. On the left are the Review window, showing commands you've already run, and the Variables window, showing variables in the data you are using, both empty right now, of course. You can try moving around the windows, and changing their size. You can right-click on the Results window to choose another font or color scheme. If it ever happens that you accidentally move a window to where you can't see it anymore, you can restore the factory defaults like so: in Stata 9 and earlier, click on Preferences, and choose Manage Preferences, and Load Preferences and click on Factory Settings; in Stata 10, click on Edit, choose Preferences, choose Manage Preferences, and Load Preferences and click on Factory Settings. In Stata 11, click on Edit, choose Preferences, choose Load Preference Set, and click on Factory Settings. If you have a Stata version prior to Stata 11, I recommend you go to General Preferences (Windowing tab) and check "Lock splitters," and uncheck "Enable ability to dock, undock, pin or tab windows" so you don't accidentally move a window to where you can't see it anymore.

[1] You could also use menus and dialog boxes, but I don't, and I advise others not to—it is simply easier in the long run to learn to type your commands in the first place.

B.1.2 Updating and Getting Help

The command to get free updates is `update` and you can check if your Stata is up to date with `update query`. You shouldn't ever have to update if you're using Stata on a network and your network administrator is up to snuff, but if you update on your home computer, make sure you do that last step of updating the executable. If Stata starts producing weird behavior, it is almost certainly because you did a partial update. Type `query` and just notice how many options you can set with the `set` command (we'll discuss some below). Now type `about`. This is the kind of info you will want to include if you ever send an email asking for help from Stata's Tech Support, and if you do not have the most up-to-date version you should include the version number in any email asking for help (from Stata or from other users).

Speaking of help, most of your questions can be answered by typing `help` or `findit`. Suppose you want to find a command for instrumental variables estimators—try typing `findit instrumental` and scroll down to see some of the relevant commands, both official and user-supplied, and FAQ files. You could also be more specific: `findit instrumental first-stage` (the user-written program `ivreg2` which shows up in the output is one of my favorite programs).

If help and findit don't give you an answer, Google often will (often from the archives of stata.com or the Statalist; for example if you try to install Stata 10 on a Windows 2000 machine, it may complain that you don't have gdiplus.dll but a quick Google search turns up http://www.stata.com/support/faqs/win/gdiplus.html with a fix). I often use Google with the **site:stata.com** qualifier that restricts the search to stata.com (including their archives of the Statalist). There is also a large collection of FAQ's at Stata's website (http://www.stata.com/support/faqs).

As a last resort, you can subscribe to the Statalist (http://www.stata.com/statalist/) and ask the experts, but be sure to read the Statalist FAQ first (http://www.stata.com/support/faqs/res/statalist.html), and spend some time framing your question concisely and clearly to maximize your chances of a good answer. I would recommend reading some posts and replies before you send your own post to the list. Make sure you specify your problem clearly, avoiding discipline-specific jargon and provide a short but informative summary, preferably with an example using the publicly available datasets described at `help dta_contents`. I say "as a last resort" only because 8 out of 10 questions posted on Statalist can be answered by looking at a help file, and another 1 out of 10 by searching the Statalist archive.

B.1.3 Syntax

For an example of figuring out the syntax for a command, install one of my favorite user-written commands with `ssc install ivreg2, replace`, then open the help file (`help ivreg2`); this is a really well-written help file. Like the basic `regress` command, the `ivreg2` command operates on a list of variables (click on `varlist` in the basic syntax line for help on how to specify a variable list). You can also specify weights; weights are always optional, but not every command allows weights. You can also specify restrictions with if or in, which we will come back to below. Then you can specify a variety of options after the comma (the open square bracket and comma indicates that all the options are optional, which is not so surprising, I guess).

Some of the options, e..g. cueoptions() and robust have parts of the word underlined. The underlined part is a minimal abbreviation of the option, i.e. you can just type those letters and the meaning is the same as if you had typed the whole thing. Abbreviating variable names can be dangerous, but there is no reason to type the command or option name out—anyone who reads the code will know what you mean, and if they forget what command a particular abbreviation stands for, it's easy to find out.

Try typing help su. Note the minimal abbreviation on the whole command. Now try help sum. Stata tries to help you as much as it can here: do you mean the summarize command or the sum() function? If you see sum in the spot of a command, you know it is the summarize command; if you see sum followed by parentheses, it the function.

The command varlist [weights] [if] [in] [, options] layout is one of the two most common basic syntax diagramsthe other has a =exp part which we will see when we come to generating variables. The if qualifier restricts any command to operate only on observations where the statement is true, and the in qualifier restricts any command to the set of observation numbers specified in a list of numbers (see help numlist). Weights are covered below.

B.1.4 Stata Files

Mostly, you need to know about 5 kinds of files to use Stata: do files (with the .do extension), log files (usually with the .log or .smcl extension), .dta files, .dct files, and .ado files. There are help files, too, with the .hlp extension up through Stata 9.2 and .sthlp for Stata 10 and up (the main reason for the new extension is ongoing problems with Windows trying to open .hlp files as if they were Microsoft

Help Format files, after Microsoft improperly appropriated the extension). Only occasionally will you run into .mo or .mlib files (see help m1_first), or .exe or .dll files (see help plugins). In short, a do file is a list of commands, a log file is a compendium of raw output, a .dta file is a Stata dataset, a .dct file is a dictionary (see help infile) that specifies how some raw file can be turned into a Stata dataset, and an ado file is an "automatic do-file" that loads a program (see Programming below).

B.1.5 Recordkeeping: do files and log files

Everything in this chapter you can do interactively, which means type stuff in one line at a time (or even [heaven forfend] use a dialog box), but the right way to do things is in a do-file. This is a text file of commands, all of which are executed one after another, and provide a good way of making sure you can reproduce your results, which is a basic requirement for good research. Once you've typed commands in a do-file called "p.do" you can run the do-file with the command "do p" and see everything run. You can open a do-file using the Window...Do-file editor...New file command or type doedit or just hit Ctrl-8.

Write set mem 100m and save the file as /ado/profile.do (create the /ado folder if necessary). Then go back to the command window and type "cd /ado" (makes /ado the current directory) and do profile (runs the do-file profile.do). The set mem command increases the amount of memory available to Stata, which makes it a bit easier to open a dataset. Also see help profile.

B.1.6 File environment commands

Of course, you need to work with other files, too, so these types of commands may prove handy:

```
cd "c:\Program Files"
                                change directory to c:\Program Files
pwd
                                 shows your current directory (path)
erase /ado/auto*
            erase the files in \ado whose names start with ``auto"
copy x y
                copy file at location x (could be a URL or a path)
                                    to location y (on a local drive)
!del auto*
                              The ! or shell command runs a program
                                  in the operating system (like del)
winexec "\IEXPLORE.EXE" http://google.com
                       Also runs a program in the operating system,
```

```
                        but doesn't wait for it to finish before
                                 Stata goes on to the next thing.
type profile.do
                          Displays the contents of the text file
                              "profile.do" in the Results window.
```

You can use a text editor such as Textpad or Stata's built-in editor (though Stata's built in editor only allows about 32KB of commands per file, that is really not a binding constraint, since one do file can call another, e.g. myfile1.do can end with the line do myfile2.do) but don't ever use Word or another non-text editor to make your do-files, since it becomes too easy to introduce non-text gibberish. See http://fmwww.bc.edu/repec/bocode/t/textEditors.html for more info on text editor choices and configurations. I use WinEdt myself, which is also handy for writing TEXfiles (how this book was typeset).

You should really start every do-file out by first stating what version of Stata you're using with e.g. version 9.2 (this ensures the do file will run the same way in a subsequent version, and won't accidentally run incorrectly in a prior version) and then making a log file, or a record of everything run, with a command like log using myfile1, text replace and put the about command right after it so you can see what version of Stata you ran in. It helps to use the command capture log close (right before opening a log with log using) to close any open log file (the capture command suppresses any error message that results from no log file being open).

Another handy command, especially for when you first start up Stata, is cmdlog, which opens a log file that just saves just the commands you type, not their output. If you like to just type willy-nilly and see what happens, then write the real do-file later, the cmdlog command can help you make your do-file easily. Just type whatever you want, then edit out the dross after the fact. (I open a cmdlog at the beginning of every session using profile.do, just in case I need to look up something I typed later).

Comments are a necessity in do-files and log files, so you will want to know that you can put anything you want between /* and */ and the contents will be ignored. Any line that starts with * will be ignored, and the rest of any line after /// will be ignored, too. So if you want to put a long command on more than one line you can

```
set mem /*
*/ 100m
```

and Stata will read it all as one line or

```
set mem ///
100m
```

with the same result. You can change the end-of-line delimiter to a semi-colon or back to a carriage return with the #delimit command on a line all by itself.

Useful commands for controlling the flow of output to the Results window include set more on, more, and set more off. If a command produces more output than can be shown in the Results window on the screen size you're using, the command will pause until you hit a key. Sometimes you would prefer that Stata keep going through all the commands, rather than waiting for you. The set more off command at the start of a do file will ensure that Stata doesn't wait for you to hit a key. Other times, you'd like Stata to pause until you've had a chance to review output (after changing the data, as with a merge, for example), and the commands set more on, more, and set more off in that order will make Stata pause. The quietly command can preface any other command, and hide its output, which is often handy. You can also enclose a block of commands inside qui { and } to suppress the output of each command in the block. Inside the block, you can preface a command with noi to show output for that command. If you want to see under Stata's hood, and have Stata show everything that a command does (including programs that it calls, and programs that the programs call), you can set trace on. If you set traced 1, you will see only what the command does, and if you set traced 2, you will see only what the command and any programs it call do. These commands are especially useful when debugging programs and loops (q.v.). To turn off this verbose output, just set trace off.

B.2 Data

After you've got Stata up and running, and you've got some kind of record of your work in place, you need to get some data. A lot of the time, you will be given data in Stata format, which you can load with the use command. Sometimes you will be converting data from another format using StatTransfer or DBMSCopy, or importing text files using infile or related commands. The infiling situations are often idiosyncratic, and covered by help infiling, so I will just touch on two things that may be helpful: insheet and dictionary files (see help infiling for more) . The command insheet will import tab-delimited or comma-delimited files in one step, and if the first line is a bunch of variable names, it's pretty automatic, but you might wind up with text or string variables where you wanted numericsee help destring for a quick fix (also helpful are encode and decode, which turn string variables into numeric variables and vice versa). More complicated situations usually require two files. One is a dictionary file with format information, and

the other a do-file that uses the dictionary to import data via an infile using X command, and then does all the data manipulation you need.

As an example of some really complicated infiling problems and how to easily address them, you can check out the examples in the help file for `ddf2dct` on SSC, for reading CPS or SIPP data stored in raw text files, then assigning labels (see help label), and (for the CPS) turning household and family records into household and family variables attached to individuals.

We'll skip over this, and use some data that comes with Stata. The sysuse command loads a dataset that Stata finds on its search path, so if you type sysuse auto, you will load a commonly used dataset. You can see the variables show up in the variables window. These little toy datasets are good to know about (as are the datasets available via webuse; see `help dta_contents`), because if you ever have a problem that you want to get help with, you should recast it as a problem on one of the little toy datasets that ship with Stata. Most of the people who might answer your question will stop paying attention if you spend the first paragraph describing the variables on the NSAF or SASS, or whatever massive esoteric dataset you're using that no one else cares about.

Try out another set command, to see what kinds of things you can do to your environment:

```
set varlabelpos 8
set varlabelpos 30
```

Did you see what happened in your variables window? This is especially handy when someone else has made your data, and named the variables using some really long names like mydatacamewithniceshortnamesbutImadethesevariablesimpossibletouse. Variable names should be short, and labels short be as short as they can be and still be meaningful. Details can go in notes on variables.

The clear command gets rid of the data in memory, and is therefore a very dangerous command—until you save your data to disk, it is all held in memory, and if your computer shuts down unexpectedly or your clear when you didn't mean to, you will lose your changes. No different than most computer programs, but worth emphasizing more than once.

The save command is crucial, and often dangerous. Stata will not let you save over an existing dataset, unless you specify the replace option—but this is very dangerous. It is all too easy to change your data in some irrevocable way and then overwrite your good data with bad data. It's good practice to start every do file by using or infiling data, and do stuff to the data, and save a different dataset under

a new name at the end of the do-file. Note you can also screw up the replicability goal by saving multiple different versions of your data under the same name in different directories. If you type save auto and save auto.dta in the /ado folder, you have two copies, so you could make corrections to one, then open the other, and get incorrect results in such a way that the source of the error would be very hard to track down.

Some of these problems can be avoided with naming conventions (see Figure B.1). One way is to have a do-file which saves the data at the end with the same name as the do-file, where you never overwrite the original dataset (any corrections to the data are made in the do-file). So you might have raw data in cps.txt read in by cps05dropouts1.do which saves an analysis file cps05dropouts1.dta, then cps05dropouts2.do makes some new variables and runs some tabs and saves cps05dropouts2.dta, and then cps05dropouts3.do drops a bunch of data to make an estimation sample and saves cps05dropouts3.dta.

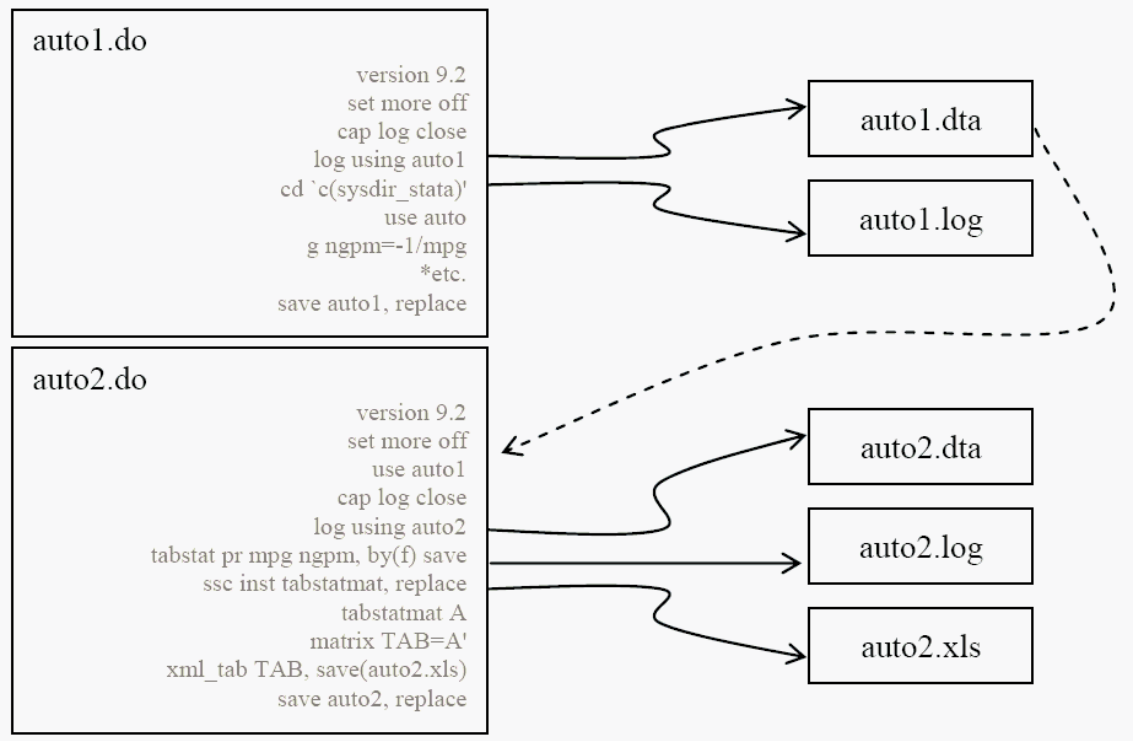

Figure B.1: Simple naming conventions can save you a lot of headache

B.2.1 Annotating the Data

The label and note commands are invaluable. You should always label every variable, describing what it measures, or when you open the data again in 5 years, or someone else does, no one will have any idea.

```
la var rep78 "Repair Record, 1978"
```

You should also label the values of any categorical variable.

```
la def replab 1 "Good"
la def replab 5 "Bad", modify
la val rep78 replab
```

You can also add notes on any variable like so:

```
notes rep78: not sure if 1 is good or bad
```

and add notes to the whole dataset:

```
note: rep78 badly labeled
```

and list the notes by typing notes.

B.2.2 Seeing the Data

You can type list to list your data in the Results window, and I use this command with various options often, to see what I've done to the data. But try typing browse to take a look at the auto data in spreadsheet form. As you can see, the data is a big matrix with variables as columns and observations as rows. Suppose we know that the AMC Pacer really has 10 cu ft of trunk space. Click on the 11 under trunk on the AMC Pacer line and type 10, enter. Nothing happens (this is the beauty of browse) so you cannot accidentally screw up your data. Now type edit and do the same thing, and you've changed the value. Close the window; now Stata asks you if all the changes you made were intentional, and then records the commands that produce those changes on the results screen (and your log file, if you've got one open, but nothing to your cmdlog file, since you didn't actually type the commands). But you have now changed only the data in the computer's memory, not the data on the hard drive. If you type sysuse auto, you will be told NO, but type sysuse auto, clear to force Stata to lose the data it has in its head at the moment, and brow to see that the value on the hard disk copy of the data is unchanged.

You can also see subsets of the data by typing edit varlist, e.g. edit make trunk or edit if some condition, e.g. edit if trunk==11 (note two equal signs), or in some obs numbers e.g. edit in 1/3. The list command will show the same info as browse in the Results window (and therefore in the log file).

Note that the if qualifier restricts any command to operate where the statement is true, and the in qualifier restricts any command to the set of observation numbers specified in a list of numbers (see help numlist). There are two uses of the word if in Stata—the second is described below, but see help if and help ifcmd for details.

In Stata 11, there are a variety of fun new features in the data editor, and you can record changes made to the data in the data editor, but it is still a bad idea to do a lot of work there—only code in a do file is truly reproducible.

B.2.3 Descriptive Statistics

To get an overview of your data, the describe, codebook, inspect, and summarize commands are quite handy. Type su (short for summarize) to get summary stats on all the vars, or su trunk for just one. You can get more detail with options, e.g. su trunk, d. Probably su is the one command I use the most. Another most-commonly-used command is tabulate, abbreviated tab, which produces simple tabs, e.g. tab trunk (from which we see that the median is 14), and crosstabs, e.g. tab trunk for. Cousins table and tabstat can produce more complicated tables of summary statistics.

B.2.4 Making new data

Most of the time, you have to make the variables before you make a tab. The generate command (abbreviated gen or g) makes a new variable, and the replace command replaces values in an existing variable. You type gen (or g) or replace, then a variable name, then a single equal sign and an expression or function (see help functions, help exp, help operators, help subscripting, and help _variables):

```
gen met=tru* (12*.0254)^3
su met tru
Variable | Obs Mean Std. Dev. Min Max
-------------+--------------------------------------------------------
met | 74 .3891653 .121417 .1415842 .6512875
trunk | 74 13.74324 4.2878 5 23
```

Now let's make a dummy variable indicating trunk size at the median:

```
gen met=(trunk==14)
```

Whoops. We already have that variable. Let's say we don't care about the old met var and drop it:

```
drop met
gen met=(trunk==14)
```

But the other thing we could do is just replace it like so:

```
replace met=(trunk==14)
```

Note that the single equal sign means "assign" in a command like gen y=x but the double equal sign means a test of equality as in gen y=x if z==1.

B.2.5 Logical Statements and Missing Values

What is that replace met=(trunk==14) expression doing? It's a logical statement, i.e. it is true or false that trunk==14, so the statement (trunk==14) evaluates to one (true) or zero (false). So it is doing the same thing as:

```
gen met2=1 if trunk==14
replace met2=0 if trunk!=14
```

which first makes met2 equal one whenever trunk equals 14, with a missing value shown as a single period everywhere else, then puts zero whenever trunk doesn't equal 14 see help operator for more operators. Note that != means "not equal to" (and see NOT below). You could also

```
gen met3=1 if trunk==14
replace met3=0 if mi(met)
```

which first makes met2 equal one whenever trunk equals 14, then puts zero whenever the new variable is missing (undefined), and seems like the same thing as

```
gen met4=1 if trunk==14
replace met4=0 if trunk<14 | trunk>14 & trunk<=23
su met*
```

where the | symbol means "or" and the & symbol means "and" but that's not quite true—why? Try the following:

```
tab rep78
gen medr=(rep 78==3)
gen medr2=1 if rep 78==3
gen medr2=0 if rep78==1 | rep78==2 | rep78==4 | rep78==5
su medr*
```

The problem is, “missing” is a value too. There are actually a variety of missing values (see help missing), which are helpful if you want to code the reason something is missing (e.g. .a means “Refused interview” and .b means “Not at home”), and they are all bigger than any real number (i.e. they also represent infinity). So gen hir=(rep 78¿=3) will include all missing rep78 values in the hir “High Rep” category. The better way to code the creation of the hirep dummy variable is

```
gen hirep=(rep78>=3) if !mi(rep78)
```

which will put a missing value in every obs where rep78 is missing, though you could also

```
gen hirep=(rep78>=3) if rep78<.
```

because the missing value . is bigger than any real number, and extended missing values are bigger than the missing value . (and .a is smaller than .z).

You should always account for any potential missing values when you write a generate or replace statement; failure to do so may result in incorrect calculations. This goes for any statistical software, of course. You cannot simply put values in for some missing values (which is what failing to account for missings will do) as that will bias your results; see section 1.5.4 for ways to fill in missing values.

On NOT: the NOT EQUAL operator says A does not equal B, e.g. trunk!=14 means trunk is not 14 for a given observation and evaluates to one or zero. The NOT operator gives the opposite of a true-or-false statement, so !(trunk==14) means the same as (trunk!=14), which can be a handy trick for much more complicated expressions.

B.2.6 System variables

What if you wanted not just the median value but the middle observation (or the first of two middle observations)? Then you would need to reference the observation number directly, or you could

```
gen med=1 in 37
replace med=0 if mi(med)
```

but it’s much more direct to use a built-in variable which is equal to the current observation number and is always available on every dataset: _n

```
gen med2=(_n==37)
```

But what if you didn't know how many observations you had in your dataset, and you didn't want to have to figure it out interactively before you wrote your do-file? Then you just use _N, which is a built-in variable containing the number of the last observation:

```
gen med3=(_n==_N/2)
su med*
```

What if you want instead to change the value of some variable in that middle observation, e.g. change medr2 to 2 in that observation? It's just replace medr2=2 in _N/2 right? No, you can't put calculations after the "in" qualifier. But you can use a trick to force calculations in these kinds of spots that nominally don't allow expressions, but only numbers: put the expression inside a left single quote, equal sign and a right single quote.

```
replace medr2=2 in `=_N/2'
```

and Stata will calculate the thing inside `= and ' before running the command. In this case, you could also

```
replace medr2=2 if _n=_N/2
```

but the trick is handy when you have no alternative.

B.2.7 Subscripting and tsvarlist

Often, you want to refer to other observations when making a variable, e.g. the value before or after the current one. This is when you want to use explicit subscripting (see help subscripting) such as

```
gen med2lag=med2[_n-1]
edit med2 med2lag
```

You can even put a variable name in the brackets indicating which observation you want to reference:

```
gen med2lag=med2[wave]
```

This set of tricks for looking at neighboring observations is one of the minor advantages of keeping all the data in memory (as opposed to reading it in a line at a time as SAS does), but the cost is that you have to have enough memory to keep all the data in memory. There is a whole suite of functions (see help tsvarlist) for making lags and leads and differences without subscripting, which is safer, since a one-period lag is not always the prior observation (if years were 1989, 1990, 1992, 1993, the lag should only be defined in the second and fourth observations).

B.2.8 Functions

When I wrote

```
gen medr2=0 if rep78==1 | rep78==2 | rep78==4 | rep78==5
```

above, I really should have used a handy function:

```
gen medr2=0 if inlist(rep78,1,2,4,5)
```

and of course there is a long list of handy functions at `help functions` organized into categories: Mathematical functions, Probability distributions and density functions, Random-number functions, String functions, Programming functions, Date functions, Time-series functions, and Matrix functions.

Note the distinction between functions and commands; commands occupy the first slot in anything you type in the Command window, whereas functions appear on the "right hand side" of some command, for example `display exp(lnfactorial(10))` which shows the factorial of 10 on the screen. In that command, the command `display` is used in its useful capacity as a hand calculator, and there are two functions `exp()` and `lnfactorial()` used to calculate the scalar shown. Most functions are shown together with parentheses to emphasize they are functions, not commands, which means you can't type `help exp` to get help for that function (try it).

B.2.9 By groups

A lot of tricks in generating variables are used so much, they are coded in the "extensions to generate" command `egen` and you can learn a lot just by reading help egen. Even more are included in user-written `egenmore` (on SSC). But you should know that none of these tricks are that complicated to write using the basic `generate` and `replace`, possibly with a `by` command or loop thrown in (loops are covered below in B.5.1). The by command steps through each unique value of a variable, and treats each set of observations as a little separate dataset. The data must be sorted by values of the by-group variable before using the by command (see help sort).

Suppose we wanted to calculate the minimum value of rep78 for foreign and domestic cars—we could:

```
sort for rep78
by for: gen minrep=rep78[1]
```

since the first line orders all the cars by repair record within for==0 or for==1 (Foreign) and the second line looks only within each group, and assigns the value of rep78 in the first observation to the new variable minrep. Note that the _n variable which records the current observation number resets within each by-group, i.e. each group is treated like its own little dataset. You can combine these two steps with the bysort command (abbreviated bys):

```
bys for (rep78): gen minrep=rep78[1]
```

where the variables after bys but before the parentheses are the variables you want to perform the command by, and the variables inside the parentheses specify the sort order of each little dataset you are performing the command on. For more, see: http://www.stata.com/support/faqs/data/group.html and many other related FAQs on data management, or the Data Management manual.

One common class of problems is to calculate a mean or total over all members of a group, for example all workers in each city given data covering a nation, or to calculate for each observation i a mean over all individuals in the group that are not i (i.e. not including that observation) For example, we might want the unemployment rate for all other workers, which is a mean of a dummy for unemployed over "not i" cases.

Often `egen` provides an easy solution (at least to part of the problem) in one line of code, though it is never faster than using `generate` and `by` and other built-in commands. `egen` is in fact a whole collection of functions, some of which are specifically tailored to these kinds of problems. For example, `egen, total() by()` produces totals, including counts, separately for groups defined by one or more variables specified as arguments to the `by()` options. Note that `total()` in Stata 9 and later is a replacement for `sum()` in Stata 8, which was deemed confusing given the definition of the `sum()` function for `generate` and `replace` as a running sum.

One simple method of calculating a "not i" total using `egen` is to calculate a total for each group, then subtract each member's contribution from that total (noting that the contribution will be zero if the value is zero). If a mean is desired, we can also calculate the "not i" number of nonmissing values, as the total number of nonmissing values within each group less an indicator for whether i is missing, and divide the total by that observation count. A weighted average simply multiplies the variable by weights before calculating a total, then divides by the sum of weights.

As the help file for `egen` shows, `total()` takes an expression as an argument, not just a single variable. If we want to measure the number unemployed

for other adults in the same city, where unemployment is defined by a variable `work` being zero and a variable `lf` being one, we can just `egen n=total(work==0&lf==1&age>=18), by(city)` to get the total, then `replace n=n-(work==0&lf==1&age>=18)` to subtract the contribution of observation i. To get the unemployment rate, we would just divide the number unemployed by the number of nonmissing observations employed to calculate the total.

This approach depends on two properties of sums. First, the sum for "everybody else" is just the sum for "everybody" minus the sum (the single value) for the current observation. Second, the value of a sum is not affected by adding or subtracting 0. If we want other statistics, we cannot count on these properties in general; we need a more general approach using a loop (see B.5.1 below and the FAQ "How do I create variables summarizing for each individual properties of the other members of a group?" at
http://www.stata.com/support/faqs/data/members.html).

A useful trick that can save a lot of time and avoids such loops applies in a specific case where each observation is related to another via a line number or the like. For example, consider a family survey in which we do not have direct information about attributes of other family members attached to each person, but we know family relationships. Suppose we have variables for family id (`family`) and individual id (`person`) and also for father id (`fatherm`) and mother id (`motherm`) (which are missing if a person's mother or father is not a member of the same family), and we want to calculate the age of each person's mother. The example data can be entered like so:

```
input family  person  fatherm  motherm  age
      1       1       .        .        36
      1       2       .        .        37
      1       3       1        2        14
      1       4       1        2        5
      1       5       1        2        2
      2       1       .        .        54
      2       2       .        1        32
      2       3       .        2        10
end
```

So family 1 includes a couple and three children, all of whom are children of the same mother and father, while family 2 includes a grandmother, her daughter and a grandchild, the son or daughter of that daughter.

In this example data, the individual id (`person`) corresponds to the observation number _n within each family (remembering that under `by varlist:`, _n

is interpreted within each group of observations, not for the whole dataset) so, importantly, no person numbers are skipped within family (note that this often will not be true in survey data). In this specific case, we can calculate the attribute of another family member whose individual id is coded in another variable by subscripting (see also `help subscripting` in Stata) using the other variable. To calculate mother's age, we just:

```
bys family (person): g m_age=age[motherm]
```

Before using the previous command, we should always check our assumption that the individual id `person` corresponds to the observation number _n within each family with an `assert` command:

```
bys family (person): assert person == _n
```

In the event that the individual id `person` does not always correspond to the observation number _n within each family, or person numbers are skipped within family, we can use the `fillin` command to ensure every person number exists, possibly with all missing data aside from identifiers. Let's create a problem in our data where one number is skipped in a family:

```
input family  person  fatherm  motherm  age
      1       1       .        .        36
      1       3       .        .        37
      1       4       1        3        14
      1       5       1        3        5
      1       6       1        3        2
      2       1       .        .        54
      2       2       .        1        32
      2       3       .        2        10
end
```

Now if we fail to check the assumption, or ignore the result of the `assert` command, we will get the mother's age in family 1 as 14, since the mother (person number 3) is the second observation and the third observation has `age` equal to 14. In this example, we can just

```
fillin family person
bys family (person): assert person == _n
bys family (person): g m_age=age[motherm]
drop if _fillin==1
drop _fillin
li, noo sepby(fam)
```

But this will not always be true. Let's create a bigger problem in our data, where the same number is absent in every family:

```
input family  person  fatherm  motherm  age
      1       1       .        .        36
      1       3       .        .        37
      1       4       1        3        14
      1       5       1        3        5
      1       6       1        3        2
      2       1       .        .        54
      2       3       .        1        32
      2       4       .        3        10
end
```

In this case `fillin` has no way of knowing that we intend it to add an extra observation with all missing data, but `person` equal to 2, in each family. One solution is to recode every id; another that works well here is to create a fake family with all missing data and every `person` number from 1 to the highest number observed in the real families, preferably using an id number we know so we can drop it when we're done:

```
set obs `=_N+1'
su family, meanonly
local fake=r(max)+1
replace family=`fake' in l
su person, meanonly
expand `r(max)' in l
bys family (person): replace person = _n if family==`fake'
fillin family person
bys family (person): assert person == _n
bys family (person): g m_age=age[motherm]
drop if family==`fake'
drop if _fillin==1
drop _fillin
li, noo sepby(fam)
```

The `fillin` command may add many observations to the data, perhaps making the above code unbearably slow, or even making the data too large for your machine. There are at least two ways around this:

1. `use` only part of the data at any given time, preferably parts with similar maximum person numbers to minimize computation time, and do the `fillin` and calculations on each part before appending parts back together, or

2. `rename` identifiers and variables, `save` temporary files, and use `merge` to put the data back together.

The second approach is described in the the FAQ "How do I create variables summarizing for each individual properties of the other members of a group?" (at http://www.stata.com/support/faqs/data/members.html) and in various other places.

B.2.10 Data manipulation

To destroy the data in memory, and simultaneously turn it into useful summary statistics, and quickly, the `collapse` command is invaluable. Of course, you may want to save a copy of your data to disk before you destroy the copy in memory–if you want to save a temporary copy, you can use `preserve`, then type `restore` when you want it back. If you wanted to calculate the weighted mean of income at the family level using a sampling weight **finwt**, you could `preserve`, then `collapse income [pw=finwt], by(familyid)` and `save fminc` to save the calculated means and then `restore` to get the full dataset back, and `merge` the mean income back onto the main data.[2]

The `merge` command matches two datasets, one in memory and on the hard drive, usually using an identifier, such as family or person IDs. It is a relatively straightforward command, with a good help file, but for some reason, people always seems to get something wrong in merging. So be careful about checking the diagnostics that the command supplies, and always look at a few observations to make sure it went like you thought it would. Stata 11 updates the syntax to try to cut down the number of mistakes people make with `merge`, but the old syntax still works, with a warning.

The `append` command just adds data to the end of the existing data (like stacking one matrix on top of another). If you had foreign auto data in for.dta and domestic in dom.dta, you could use for, then append using dom, then save auto. It's important to remember that if variables are named differently, they will not be missing in half the data (well, not half, but you get the idea). If variables that are named the same have different value labels (corresponding to different coding), the appended data will lose its coding and use the master data's coding. For example, if gender is 1 for male in one year and 2 for female, and you have defined value labels that reflect that in your data, and then you append the next year's data where 0 is female and 1 is male, you will see:

```
tab gender
gender | Freq. Percent Cum.
```

[2] Try ssc describe _gwtmean for a faster alternative.

```
------------+-----------------------------------
0 | 390,003 25.03 25.03
Male | 612,066 39.28 64.31
Female | 556,212 35.69 100.00
------------+-----------------------------------
Total | 1,558,281 100.00
```

The `reshape` command changes the shape of the data in very useful ways. If you have panel data on earnings, but each year of data are saved as variables, such as e1951, e1952, etc. then you will want to reshape the data to have observations as person-year data points (so you can run panel regressions):

```
li persid e19??, noo
+---------------------------------+
| persid e1951 e1952 e1953 |
|---------------------------------|
| 1 1800 1800 1900 |
| 2 2600 3100 3200 |
| 3 3000 3800 5500 |
+---------------------------------+
reshape long e, i(persid) j(year)
Data wide -\> long
-----------------------------------------------------------------------------
Number of obs. 3 -\> 9
Number of variables 4 -\> 3
j variable (3 values) -\> year
xij variables: e1951 e1952 e1953 -\> e
-----------------------------------------------------------------------------
li persid e*, noo sepby(persid)
+---------------+
| persid e |
|---------------|
| 1 1800 |
| 1 1800 |
| 1 1900 |
|---------------|
| 2 2600 |
| 2 3100 |
| 2 3200 |
|---------------|
| 3 3000 |
| 3 3800 |
| 3 5500 |
+---------------+
```

B.2.11 Returned saved results, precision, scalars

A lot of times, you want to use a value that appears on the screen, either to generate a new variable, or to do some other kind of calculation. Returned saved results are the most useful way to do that. After any command, you can type `return list` to see a list of items you might want to use, e.g.

```
su trunk, d
ret li
gen mt=r(p50)
```

or sometimes you might get a number that way that doesn't show up on the screen explicitly, e.g.

```
tab trunk
ret li
gen nvals=r(r)
```

where r(r) is the number of distinct values in the table.

There's really no reason to generate a variable that takes on the same value for every observation, as I just did, especially on a big dataset (where space in memory is at a premium). If we want to use a single number later in our do-file, we can save it as a scalar like so:

```
scalar nval=r(N)
```

which is just the number of observations summarized in the table, but you have to exercise some caution referring to a scalar, thanks to Stata's eagerness to interpret pieces of expressions as variables:

```
gen test=nval
gen test2= scalar(nval)
su test* nvals
```

Here Stata wanted very badly to interpret the `nval` you typed as a variable, and there was a variable named `nvals` that nval was a good abbreviation for, so Stata used that and bypassed your scalar–a good reason to always use the scalar(name_of_scalar) construction, even when it seems redundant.

B.2.12 Display, Formats, Datatypes, and Precision

If you just want to do a simple calculation, say average number of observations per cell, which is r(N) divided by r(r), you can use the `display` command (abbreviated `di`):

```
tab trunk
di r(N)/r(r)
```

which is very handy as a calculator, too:

```
di _pi, ln(_pi), tan(_pi/4), norm(1.96)
```

and you can change the display format of the numbers on the screen easily:

```
di %20.18f _pi
```

using the same kinds of formats available to assign to variables (see `help format`) which are really helpful for variables mainly when it comes to displaying dates (see `help dates`) in my experience. But the mention of display formats for variables brings up a related point that always seems to trip people up:

```
gen tenth=_n/10
su tenth if tenth==6
su tenth if tenth==5.9
```

Stata finds no observation with `tenth` equal to 5.9 but we clearly think that observation 59 should have that value, and if we look (via `list tenth in 58/60`) it sure looks like it does. The problem here is that Stata thinks in double precision (see `help data_types`) using 8 bytes per number, but variables are usually defined in float precision (4 bytes), and tenths have no exact representation in binary numbers. If you write out the value of 5.9 that was used in `su tenth if tenth==5.9`, and the value of `tenth` in observation number 59, you can see the source of the problem more clearly:

```
di %20.18f 5.9
5.900000000000000400
di %20.18f tenth[59]
5.900000095367431600
```

and these two numbers are not equal. You could create you variable as a double (with higher precision):

```
gen double t2=_n/10
su t2 if t2==5.9
```

or you could round to float precision:

```
su tenth if tenth==float(5.9)
```

to get around this problem. It doesn't come up that often, but it does seem to confuse people when it does. If you always use integer values, you will never run into this problem.

B.3 The Stata Macro

What are called "variables" or "literals" in other programming languages are **macros** in Stata. Before any command is run, all the macros in the command get interpreted.

B.3.1 Globals

Global macros (see help global) are defined as a number or string, with or without =

```
global test="seven"
global test "seven"
global test=7
```

Global macros are referenced using a dollar sign:

```
display \$test
display \${test}
```

The brackets are better style, since display $test7 is interpreted as display $test7 which is evaluated as display which displays nothing, but display $test7 is interpreted as display 77 which displays the number you wanted.

B.3.2 Locals

Local macros (see help local) are defined as a number or string, with or without the equal sign:

```
local test="eight"
local test "eight"
local test=8
```

Local macros are referenced using two kinds of single quotes:

```
display `test'
```

Make sure you get that first quote right (it's on the same key as the tilde ~ on most keyboards).

What does the = do? The equal sign denotes immediate evaluation, instead of assignment, i.e. Stata figures out what goes in the local now, rather than looking back to see what's there when you reference the macro later. Note that immediate evaluation limits the length of a local to about 245 characters, and immediate evaluation means the local won't change if other stuff changes.

What's the difference between locals and globals? Globals stick around, and overwrite existing globals (bad practice when avoidable). Locals disappear, and occupy a safe part of memory (but can't be seen when your do-file or program is done).

Why use them? They don't just save retyping–they save making mistakes. There are also many useful extended functions (see help extended_fcn, which is

linked from help macro). In particular, anything you can display, you can put in a macro with e.g. .

```
local t: display _pi " is the ratio in question"
```

which can come in handy for putting numbers in a given display format (rounding, significant digits, dates, etc.).

The order of evaluation of macros is set up in the only way that makes sense. Macros are evaluated first, before any other parsing is done. As with expressions enclosed in parentheses in math operations, macros are evaluated from the "insidemost" out. If you have a local called test that contains 7 and a global called row7 that contains _pi, you could display $row‘test’. If you type

```
loc a -1
display `a'^2
display (`a')^2
```

the first result is negative one, because Stata first puts the contents of the macro in there, then evaluates `-1^2` as the negative of the square of one. For this reason, it often makes sense to add parentheses around macros in expressions. It never does any harm to wrap them in parentheses, and it could be a big help.

All of the local macro extended functions are available without first defining a local macro:

```
local t: display _pi " is the ratio in question"
gen tvar="`t'"
```

is equivalent to

```
gen tvar="`: display _pi " is the ratio in question" '"
```

which is just a shortcut, but can really come in handy.

B.3.3 Scalars

Scalars are defined only with the = (evaluation, not assignment)

```
scalar test=7
scalar test="seven"
```

Scalars are best referenced using `scalar(name)`, e.g. `display scalar(test)`. Why? Try this:

```
clear
range test 0 1 2
display test
list test in 1
display scalar(test)
```

Scalars, matrices, and variables share the same "namespace" or memory addresses for names, so you it's a good idea to reference a scalar using `scalar()` and to reference a matrix using `matrix()` and to set `varabbrev off`, to make sure you are not accidentally referring to a value in the wrong object altogether.

The "shortcut" method of condensing a local definition into an evaluated bit of code to be included in another command turns out to be very handy, and is especially important to understand when reading other people's code. As an example, consider the following:

```
sysuse auto, clear
ins rep
local n=r(N)
local u=r(N_unique)
local l: var lab rep
hist rep78, ti("`l', `g' obs") bin(`u')
```

versus this:

```
sysuse auto, clear
ins rep
hist rep78, ti("`:var lab rep', `=r(N)' obs") bin(`=r(N_unique)')
```

both of which do the same thing. Sometimes only the first of these approaches will work, but it's often easier to use the latter approach.

B.3.4 Ifcmd

We saw the `if` qualifier already, which restricts the scope of a command like `generate`, but there is another `if` which is a command in its own right (not a qualifier) with a help file at `help ifcmd`. The `if` command executes a block of code (enclosed in curly brackets) once, if the condition that follows the command is true, or skips the block of code if it's not. So it's related to `while`, except it only runs once. The related command `else` follows an `if` command, and executes a block of code (enclosed in curly brackets) once, if the previous `if` command failed to execute its block. This command is particularly handy if you're looping in a do-file, and want to execute some additional code only for certain values, or if

you want to use a returned result (the mean, say) and execute code only if it falls in some range.

One thing that always seems to confuse people is the difference between the two uses of `if` so make sure you know the help files at `help if` and at `help ifcmd`. What output do you think the code below produces?

```
sysuse auto, clear
if foreign==1 {
g dom==0
}
su dom
```

I don't exactly understand how people seem to get this wrong at the rates they do, but I suspect it comes from some *faux ami* analogy to SAS or SPSS. It's common enough to warrant a FAQ: http://www.stata.com/support/faqs/lang/ifqualifier.html

B.4 Graphs

We've seen a plot produced by the `inspect` command, and a similar histogram can be produced by the `histogram` command, abbreviated `hist`. Most graphs are produced by the `graph` command, which has a number of subcommands, including `twoway`, `bar`, `box`, `pie`; and others. The bulk of the good graphs will be produced by `twoway`, a `graph` subcommand for graphing some variable(s) on the y axis (ordinate) versus some variable on the x axis (abscissa), that has its own subcommands, including:

```
scatter     scatterplot
line        line plot
area        line plot with shading
bar         bar graph
rarea       range plot with area shading
tsline      time-series plot
lowess      LOWESS line plot
lfit        linear prediction plot
qfit        quadratic prediction plot
function    line plot of function
histogram   histogram plot
kdensity    kernel density plot
```

all with their own very useful help files. Note that bar graphs can be produced by `graph bar` or by `graph twoway bar`, the first with an easier syntax for some people, and the second with a lot more powerful control over placement and look of bars.

Most of the examples below are linked from help twoway, and there are many examples at http://www.stata.com/support/faqs/graphics/gph/statagraphs.html, but if you read no other graph help file, you should read help scatter at least.[3] I also recommend reading help schemes (I have set scheme s2mono, perm on my computer). If you want 3D graphics, you will have to go to another program—it is one of the few gaps in Stata's repertoire. Or you use the trick in figure 1.2.

One alternative to the graph command is the Stata 7 graph command, now available as the gr7 command. This command produces twoway graphs, including scatter and line graphs, very quickly, though they are not as pretty.

B.4.1 Scatter or Line Graphs

The `scatter` graph, part of the twoway suite, is the workhorse of graph commands, and a workhorse of descriptive statistics (and regression diagnostics as well). Here's a quick example:

```
sysuse uslifeexp2, clear
scatter le year
```

where you can see immediately the impact of the Spanish Flu relative to various wars. You can add options to pretty it up, if you like:

```
sysuse uslifeexp2, clear
scatter le year, subti("Life expectancy at birth, U.S.") note("1") caption("Source: National Vital Statisti
```

and there are a few hundred thousand ways you might modify that graph with options, so I won't go into detail. The options are all in the help files.

One of the common options on a scatter plot is to connect the dots:

```
sysuse uslifeexp2, clear
scatter le year, c(l)
```

and sometimes to suppress the dots after connecting them:

```
scatter le year, c(l) m(i) [fig for Life expectancy at birth, U.S.]
```

The last is so common, in fact, there is a separate command to make it easier to type:

```
sysuse uslifeexp2, clear
line le year
```

[3] There are also some nice FAQs at http://www.stata.com/support/faqs/graphics/ and to really see what Stata can do, check out Mitchell (2008).

but note that the dots are graphed in the sort order of the data, which can result in some unpleasant looking graphs unless you specify the sort option:

```
sysuse auto, clear
line mpg weight
line mpg weight, sort name(s)
```

The name option is handy when you want to have a number of graphs open at once for comparison (or to combine them via graph combine).

B.4.2 Density and Local Polynomial Graphs

The last graph (`line mpg weight, sort`) is getting close to specifying an empirical model: mpg declines as weight increases. As useful as `scatter` or `line` plots and histograms are in small datasets, they rapidly become untenable in large datasets. A good way to get sense of the distribution of a variable (or a residual after a regression) or the functional relationships between pairs of variables in a large dataset is to use kernel estimators like `kdensity` or `lpoly`. Kernel estimators use subsets of the data and reweight to construct local estimates, for example of the proportion of cars with mileage "near" 21 mpg (a kernel density estimator), or the effect of another 100 pounds on mpg "near" 3000 lbs (a kernel regression estimator).

```
sysuse auto, clear
hist mpg, name(h)
kdensity mpg, name(k)
```

You can get the values that `hist` uses with the undocumented command twoway__histogram_gen and then graph them with tw bar, which is handy for combining graphs:

```
twoway__histogram_gen mpg, bin(8) start(12) gen(f x)
tw bar f x || kdensity mpg
```

The user-written `kdens` (on SSC) offers even more flexibility than `kdensity`. In Stata 10, local polynomial regression is performed with the command `lpoly`, but the near-equivalent command `locpoly` is available via findit for prior versions.

```
net from http://www.stata-journal.com/software/sj6-4/
net inst st0053_3
line mpg wei, sort name(bumpy)
locpoly mpg wei, name(smooth)
lpoly mpg wei, name(smooth2)
```

The `lowess` command is also available in Stata versions before Stata 10, but I've never liked it as much, mainly because it does not offer the `at()` option with the `generate()` option, and does not give the same control over kernels. The `at()` option and `generate()` option are especially helpful if you want to overlay two smoothed graphs, and use the default bandwidth of the first and the bandwidth for the second.

```
sysuse auto, clear
g m=(16+_n)*100 in 1/32
la var m "Weight" locpoly mpg wei if for==1, nogr at(m) gen(mfor)
lpoly mpg wei if for==0, nogr at(m) gen(mdom)
line mfor mdom m, leg(lab(1 "Foreign") lab(2 "Domestic"))
```

B.4.3 Bar Graphs

You can get simple bar graphs with graph bar but more flexibility with twoway bar:

```
sysuse sp500, clear
gr bar high in 1/4, over(date)
```

where the legend on categories shows their numeric value (dates measured as days since January 1, 1960) instead of a more legible format. With `twoway bar` the defaults are a bit friendlier:

```
twoway bar high date in 1/4, yla(1300(20)1340) barw(.5)
twoway bar high date in 1/4, yla(1300(20)1340) barw(1.5)
```

and you can alter the data in clever ways to get total control over the look of the graph, e.g. overlapping bars of arbitrary width, different labels, etc.:

```
replace date= date+(2-_n)/3 in 1/4
twoway bar high date in 1/4, yla(1300(20)1340) xla(14977 "Jan 2" 14979 Jan 5")
```

A related command is `twoway dropline`:

```
sysuse sp500, clear
tw dropline change date in 1/57, yline(0, lstyle(foreground))
```

and a handy command when there is a lot of data is twoway spike:

```
sysuse sp500, clear
tw spike change date in 1/57
tw spike change date
```

B.4.4 Area Graphs

Area graphs are handy for showing meaningful areas between a curve and an axis (if the running integral has some real interpretation):

```
sysuse gnp96, clear
twoway area d.gnp96 date
twoway area d.gnp96 date, xla(36(8)164, angle(90))
  yla(-100(50)200, angle(0)) yti("Billions of 1996 Dollars")
  xti("") subti("Change in U.S. GNP", position(11))
  note("Source: U.S. Department of Commerce, Bureau of Economic Analysis")
```

And a range plot with area shading (tw rarea) is useful when the area between two functions is informative:

```
sysuse sp500, clear
twoway rarea high low date in 1/57
```

This is handy for custom standard error graphs (but see `lfitci` for an automated solution):

```
sysuse auto, clear
qui regress
mpg weight
predict hat predict s, stdf
gen low = hat - 1.96*s
gen hi = hat + 1.96*s
tw rarea low hi weight, sort color(gs14) || scatter mpg wei
```

Note that we graphed the shaded area first and then the scatterplot. Typing

```
tw scatter mpg wei || rarea low hi weight, sort color(gs14)
```

would superimpose the shading on the scatter, obscuring the dots.

B.4.5 Mapping

There are handy FAQs on the Stata website describing how to make maps in Stata, using two user-written commands (`tmap` for Stata 8 and `spmap` for more recent Stata versions, both available from SSC).

```
tempfile t
copy http://pped.org/stata/usa.dta `t'.dta
use http://pped.org/stata/spop90.dta, clear
spmap p if LON>=-98 using `t'.dta, id(id) cln(5)
erase `t'.dta
```

Lines can be added with options `xline()` or `yline()`. See `help added_text_options` for one way of adding text boxes. Note you can also construct maps manually if you need more control over the graph.

```
use http://pped.org/stata/spop90.dta, clear
egen c=cut(p), g(5) lab
keep if LON>=-98
g _ID=id
joinby _ID using http://pped.org/stata/usa.dta
g newpoly=sum(mi(_X))
egen g=group(newpoly id)
qui levelsof g, loc(is)
loc g
foreach i of loc is {
 su c if g==`i', mean
 loc p "gs`=16-r(mean)*3'"
 loc g `"`g' area _Y _X if g==`i', fc(`p') lc(gs0) nodropbase||"'
 }
loc o `""text(37 -70 "DC belongs" "over there")"'
loc o `"`o' yla(35 " ", nogrid notick) xla(-85 " ", nogrid notick)"'
loc o `"`o' leg(off) xscale(off) yscale(off fill) xsize(4) ysize(4)"'
tw pcarrowi 35 -71 37.5 -76|| `g' , `o'
```

Similar approaches work for presenting multidimensional graphs as a map where colors of different regions correspond to heights above the plane. This is often called a "heat map" by analogy to the colored weather maps that may appear in your newspaper. For example, `tddens` on SSC presents two-dimensional kernel density estimates as a heat map using this kind of approach.

B.4.6 The Graph Editor

As of Stata 10, you can click on a graph and add graphic elements such as text or lines interactively, using the new Graph Editor. Assuming your Stata is up to date, you can also click a Record button to echo those changes as commands (save them to a text file), which I highly recommend so you can add the commands to a do file so that you can reproduce your work at a later date.

B.5 Looping, Programming, and Automating Output in Stata

Reproducibility, error reduction, ease of use, and portability all depend on good programming style. Here a three simple reasons for using loops:

1: You shouldn't type any parameter (a number or string of characters that you might later change) more than twice in a program or do-file. If you do, you are inviting a situation where you change two instances and forget the third, and never notice the output is flawed.

2: Comments are good, but clean code is better. Comments tell you what the programmer intended, but might not help you fix or adapt the code easily. Clean code does both.

3: A little bit of programming sophistication goes a long way. If you are reformatting tables of output more than once when the output changes, you would have been better off programming the output (table layout and number display format) in the first place. If you are doing the same thing over and over (similar regression, similar graphs, similar paper topics), you are at risk of becoming a dull boy. Doing the same thing over and over is what computers are for.

B.5.1 Looping

Looping is how you make the computer do your work for you, and save yourself some carpal tunnel. Use `foreach` or `forvalues`, rather than `for` or `while` (`while` works, and even `for` still works, but they are harder to use, and the `for` command had some problems and is now undocumented). `foreach` in particular has a beautifully simple syntax:

```
foreach v in some list of stuff {
 does this 4 times, for `v'="some" and `v'="list" etc.
 presumably does something with `v' itself
}
```

Make sure the curly braces are as shown (no code after the first, and the second on a line by itself).

The similar command `forvalues` steps through a numeric list:

```
forvalues v=1/10 {
 does this 10 times, for `v'=1 and `v'=2 etc.
 presumably does something with `v' itself
}
```

Note that

```
forvalues v=1/10 {
```

is equivalent to

```
foreach v in 1 2 3 4 5 6 7 8 9 10 {
```

but the first is easier to type.

Perhaps handiest is the second syntax of foreach, i.e. one of the following:

```
foreach lname of local lmacname {
foreach lname of global gmacname {
foreach lname of varlist varlist {
foreach lname of newlist newvarlist {
foreach lname of numlist numlist {
```

so you step though each item in any list stored in a macro, or have Stata do some operations to eacha variable in a list. Note also that

```
foreach v of numlist 1/10 {
```

is equivalent to

```
forvalues v=1/10 {
```

All of these loops can be nested: just make sure you use different names for the local macros created:

```
foreach v of numlist 1/10 {
foreach v of numlist 1/10 {
di "Row `v' and Column `v' value is "
}
}
```

will obviously not work, but this is fine:

```
foreach r of numlist 1/10 {
foreach c of numlist 1/10 {
di "Row `r' and Column `c' value is "
}
}
```

See also the `continue` and `nobreak` commands for exiting a loop prematurely or preventing exit from a loop or program.

The `levelsof` command works very nicely with `foreach` to do something for every value of a variable, e.g.

```
webuse nhanes, clear
levelsof race, local(r)
foreach v of local r {
svy, subpop(if race==`v'): tab sex higbp, row
}
```

B.5.2 Output: The file, estout, and xmlsave Commands

The file command allows you to read or write to text or binary files. This means that you can write anything you can think of to a file. You could have a whole paper (text, tables, graphics, etc.) written out by one do-file, in theory, or even write out a file that would be an executable program (not that anyone would ever do that) and then run it. In particular, you can write out in any format the output from tabulations or regressions available via return or ereturn or built-in shortcuts like _b[var] (gives the coefficient on var) or _se[var] (gives the SE on var). For example:

```
webuse nhanes2, clear
qui svy: tab race diab, row ci
mat rowpc=e(b)
file open d using /b.txt, write replace
file write d "Race" _tab "Percent Diabetic"
forv r=1/3 {
 file write d _n "`: lab (race) `r''"
 file write d _tab "`: di rowpc[1,`=`r'*2']'"
}
file close _all
type /b.txt
local excel="C:\Program Files\MSOffice2000\Office11\Excel.exe"
winexec `excel' \b.txt
```

And it would be easy to embed the preceding in a larger loop that stepped through a list of diseases, and did tests of significance, etc., and then opened it all up in Excel, if you like that kind of thing.

The estout command from SSC automates the creation of tables of coefficients from estimation commands or of summary statistics, and is far better than crappy alternatives like outreg. Install it with ssc install estout. This is a really useful command, and has myriad options. The command's own website at http://fmwww.bc.edu/repec/bocode/e/estout/ has plentiful examples, showing how to create a table in Microsoft Word or LaTeXwith very little effort and very high reproducibility.

The `xmlsave` command allows you to save a Stata dataset in an XML file format: either Stata's .dta or Microsoft Excel's SpreadsheetML format. So you can collapse your data into a dataset of means by year, say, then save in Excel format. A user-written command `xml_tab` (on SSC) offers a way to write out various formatted tables to Excel-type files. You may find it especially handy when used to create and output various simple tables of conditional means:

```
tabstat price mpg weight length, by(foreign) save
tabstatmat A
```

```
matrix TAB=A'
xml_tab TAB, replace
```

In the above, `tabstat` generates a table of means for the list of variables categorized by foreign. `tabstatmat` (on SSC) saves the results to matrix A with three rows for Domestic, Foreign and Total. In the columns of matrix A are the means for the listed variables. The transposed matrix A is the matrix TAB, which xml_tab outputs into the default XML file. You can see more examples of using `xml_tab` in the file xml_tab_example.do that accompanies `xml_tab` on SSC.

B.5.3 The program Command, and ado Files

Any bit of code you want to repeat should be coded as a program. So a bit of text, or a number, should be coded as a macro, but some lines of code that, say, calculate some statistics, or convert some amounts into real dollars, or write out a table of estimates, can be written as a program, and then to run the code, you just type the name of the program.

```
cap program drop dtab
program dtab
syntax varname
webuse nhanes2, clear
qui svy: tab race `varlist', row ci
mat rowpc=e(b)
file open d using /b.txt, write replace
file write d "Race" _tab "Percent suffering from `varlist'"
forv r=1/3 {
 file write d _n "`: lab (race) `r''"
 file write d _tab "`: di rowpc[1,`=`r'*2']'"
}
file close _all
type /`varlist'.txt
end
```

Having defined a program `dtab`, you can type, e.g.

```
dtab diab
dtab heart
dtab highlead
```

and so on. If you saved that bit of code in a file called dtab.ado, you could type those lines (e.g. dtab diab) to get an instant table at the command prompt, or in an unrelated do-file. The ado file loads a program automatically; that is how a lot of Stata's official commands are written (and you can read their code, with the `viewsource` command, which is instructive, to say the least). There's a lot more about `program` in the `Programming` manual, of course.

B.5.4 Automating Appendices

Blasnik (2005) documented the handy trick of using a mail merge to put a series of Stata output files into a Microsoft Word document. I've used the approach in e.g. the appendices of Horwitz and Nichols (2007) (see also Horwitz and Nichols 2009), so I can attest that it can be quite handy in circumstances where you have to produce a lot of pages of the same type of thing, and you have to work in Microsoft Word.

B.5.5 Finding Nearest Neighbors

Loops can also step through each observation in turn, though this is time-consuming. Most cases can be handled much more quickly and easily with the `by` construct, but occasionally a loop over observations makes sense. One common case is where we have two datasets with locations and we want to find the nearest location in another dataset. This also works when the second dataset is a shapefile of polygon vertices used in mapping, and we want to find the distance of points in the first datasets to shapes in the second (the shapes could be rivers and lakes, for example, and we want to find the nearest water).

In the simplest case, we just want the distance to the nearest object in the second dataset. We `use` the dataset with points we want to compute minimum distance from, `merge` (not on variables, just an unmatched merge) the second dataset with shapes or points we want to compute distance to, then loop over observations in the first dataset: for each, we compute distance to points in the second (if the second is a shape file, we are computing distance to the nearest vertex of a polygon) and store the minimum distance to a variable we have initialized as missing. If we later decide we need other information saved, we simply add commands inside the loop to save that other information (e.g. ID number of the nearest shape, or the area of nearest shape, or the number of shapes within 50 miles, or what have you).

For example, suppose we have a dataset **mnch** with locations of charter schools (stored as signed decimal degrees of latitude and longitude) and a dataset **mnlakes** of lake boundaries, and we want to find the nearest lake to each school. A solution is shown in Exhibit B.5.6. The distance calculation is done by `vincenty` (on SSC), which is a fairly accurate means of calculating distances on the Earth, where Euclidean distance is wholly inadequate, but could also be calculated by hand using whichever formula we prefer.

Exhibit B.5.6 *Code calculating the distance to the nearest object in another dataset.*

```
use http://pped.org/mnch, clear
local n=_N
g double m=.
merge using http://pped.org/mnlakes
qui forv i=1/`n' {
 vincenty lat lon y[`i'] x[`i'], v(d)
 su d, meanonly
 replace m=r(min) in `i'
 drop d
}
drop _m i lid lat lon
li in 1/20
```

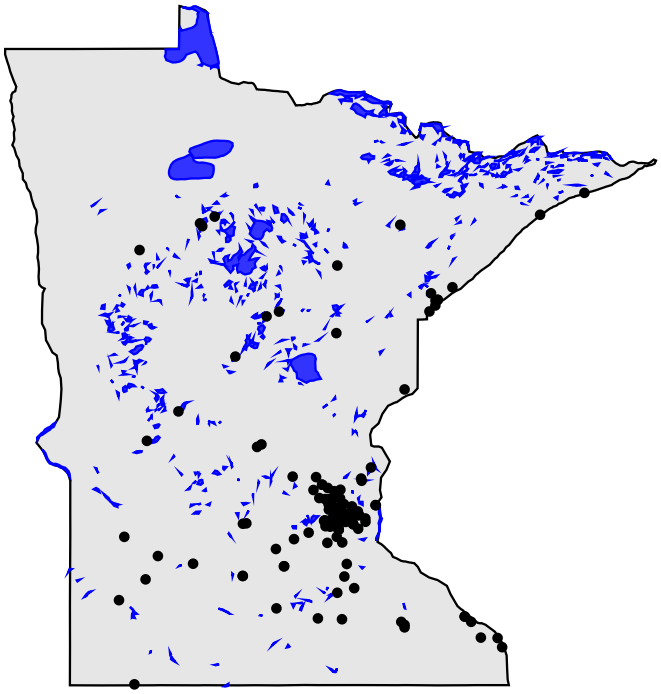

Figure B.2: Schools and larger lakes in Minnesota, pretending Lake Superior does not exist and Lake of the Woods exists only so long as it is in Minnesota.

We could also save characteristics of the nearest match, or average characteristics of the nearest several matches, or a weighted average over all potential matches, with weights related to distance, or any other statistic, by doing more calculations inside the loop. For example, to save the identifiers of every object

within a ten-mile radius we could `generate` an empty string variable `ids` and then add a `foreach` loop inside our loop over observations to add each object identifier to the string variable `ids`.

Exhibit B.5.7 *Code concatenating IDs of objects with distance less than 10.*

```
qui levelsof lid if d<10, loc(vs)
foreach v of loc vs {
 qui replace ids="`v' "+ids in `i'
 }
```

B.6 Simulation and Bootstrap

The basic idea of the simulation model is to draw error terms from some distribution, then generate the outcome variable using known parameter values, and estimate those same parameters. This can give very good estimates of the small sample performance (both bias and coverage) of an estimator for data like yours, if you use your actual data, but impose known parameter estimates and draw errors. To get coverage rates or the actual size of a test, you need to repeat the draws many times, say ten thousand or more, but on modern computers this can be done in a day. As good as the `bootstrap` command is, it may be necessary sometimes to do the resampling manually, and collect results at the end, which can also be done with `simulate`.

Note that `simulate` just calls a `program` over and over, and returns a dataset of collected results. Whatever you want to simulate should happen inside the body of the program. Anything that can be done with `simulate` can also be done with a loop, counting from one to the number of draws desired, and `post` inside the loop to a `postfile`, but it is usually easier to use `simulate`. One point to bear in mind is that `simulate` will automatically report on the last estimation result, even for an rclass program, so if you want `simulate` to collect the returned scalars instead, you can end your program with `ereturn clear`.

B.6.1 Simulation examples

For example, suppose we are interested in the coverage rates of some random-effects estimators, using cluster-robust and heteroskedasticity-robust standard errors. Exhibit B.6.2 shows rejection rates (for one kind of data) for RE of nine percent for a nominal five percent alpha (size of test) with heteroskedasticity-robust SEs but 41 percent with cluster-robust SEs.

Exhibit B.6.2 *A simulation to estimate size of tests in RE with lognormal time-invariant covariates and clustered errors.*

```
clear all
prog reclr, rclass
version 10.1
drawnorm x1-x10 v, n(50) clear
forv i=1/10 {
 replace x`i'=exp(x`i')
 }
g id=_n
expand 20
g e=rnormal()
bys id: replace e=e[_n-1]/2+e/2 if _n>1
g y=(x1+x2+x3+x4+x5)/5+v+e
xtreg y x1-x10, cl(id) i(id) re
test x1 x2 x3 x4 x5
return scalar c1=r(p)<.05
test x6 x7 x8 x9 x10
return scalar c0=r(p)<.05
xtreg y x1-x10, r i(id) re
test x1 x2 x3 x4 x5
return scalar r1=r(p)<.05
test x6 x7 x8 x9 x10
return scalar r0=r(p)<.05
eret clear
end
simul, seed(1) rep(10000): reclr
su
```

Exhibit B.6.3 shows rejection rates (for one kind of data) for RE of nine percent for a nominal five percent alpha (size of test) with heteroskedasticity-robust SEs but 41 percent with cluster-robust SEs.

Exhibit B.6.3 *A simulation to estimate size of tests in FE and RE with lognormal clustered covariates and clustered errors.*

```
clear all
prog reclr, rclass
version 10.1
drawnorm x1-x10 v, n(50) clear
g id=_n
expand 20
forv i=1/10 {
 replace x`i'=exp(x`i'*4/5+rnormal()/5)
 }
g e=rnormal()
bys id: replace e=e[_n-1]*4/5+e/5 if _n>1
g y=(x1+x2+x3+x4+x5)/5+v+e
xtreg y x1-x10, cl(id) i(id) re
test x1 x2 x3 x4 x5
return scalar c1=r(p)<.05
test x6 x7 x8 x9 x10
```

```
return scalar c0=r(p)<.05
xtreg y x1-x10, r i(id) re
test x1 x2 x3 x4 x5
return scalar r1=r(p)<.05
test x6 x7 x8 x9 x10
return scalar r0=r(p)<.05
xtreg y x1-x10, i(id) re
test x1 x2 x3 x4 x5
return scalar o1=r(p)<.05
test x6 x7 x8 x9 x10
return scalar o0=r(p)<.05
xtreg y x1-x10, cl(id) i(id) fe
test x1 x2 x3 x4 x5
return scalar fec1=r(p)<.05
test x6 x7 x8 x9 x10
return scalar fec0=r(p)<.05
xtreg y x1-x10, i(id) fe
test x1 x2 x3 x4 x5
return scalar feo1=r(p)<.05
test x6 x7 x8 x9 x10
return scalar feo0=r(p)<.05
eret clear
end
simul, seed(1) rep(10000): reclr
su
```

B.6.4 Bootstrap examples

One can also do a manual bootstrap using the `simulate` command. One way to do that is as follows (this example with 10 replications takes about 5 minutes to run, so a real application with 1000 replications or more might take many hours).

Exhibit B.6.5 *An example of a manual boostrap for* `mfx` *output.*

```
cap pr drop _all
pr eachone, rclass
drop _all
use /out
bsample, cluster(idcode)
ivprobit work age ttl (msp=south nw)
mfx, predict(p)
mat mfx=e(Xmfx_dydx)
mat es=mfx[1,1..3]
return scalar msp=es[1,1]
return scalar age=es[1,2]
return scalar ttl=es[1,3]
end
webuse nlswork, clear
gen work=wks_w>0 if !mi(wks_w)
g nw=race>1
keep work age ttl msp south nw id
ren ttl ttl
save /out, replace
```

```
qui ivprobit work age ttl (msp=south nw)
mfx, predict(p) diagnostics(vce)
local n=e(N)
mat mfx_est=e(Xmfx_dydx)
mat est=mfx_est[1,1..3]
simulate msp=r(msp) age=r(age) ttl=r(ttl), reps(10) seed(12345): eachone
bstat, stat(est) n(`n')
```

Not all of the options available in `bootstrap` are available this way, and the nomenclature may differ as well. For example, the help file for `bootstrap` says "Type estat bootstrap, bca to display the BCa confidence interval generated by the bootstrap command." whereas the help file for `bstat` shows an option accel(vector) specifying "the acceleration of each statistic, which is used to construct BCa CIs."

B.7 Mata Programming

Introduced with Stata 9, Mata is a separate programming language within Stata that is compiled, and runs with speed comparable to C and other high-level languages. Its syntax is similar to C, and it offers a library of convenient matrix functions. Stata 10 added a large number of new Mata features, including the general `optimize` utility in Mata, which is useful in a variety of ways. Stata 11 added an even larger number of new Mata features, including `moptimize`. The Mata manuals are online; just type `help mata` to get started or visit http://stata.com/help.cgi?mata (all the help files are on the web as well). However, the help files do not have that many helpful examples. Below are two simple examples, demonstrating a small fraction of Mata's tremendous capacities: (1) writing out a simple GMM estimator, and (2) solving a complicated function.

Mata is fast, because functions are compiled, rather than interpreted. I.e. if you look at what Stata sees in the pre-Mata "ado language" version of a program, you see code like the text in Exhibit B.7.1; if you look at what Stata sees in a Mata function, you see bytecode like in Exhibit B.7.2.

Exhibit B.7.1 *A block of interpretable code.*

```
forvalues j = 1/`p' {
tempvar x`j'
qui gen double `x`j'' = .
local xs `xs' `x`j''
}
qui gen double `arg' = .
qui gen double `karg' = .
```

```
forvalues i = 1/`n' {
qui replace `arg'=(`x'-`xgrid'[`i'])/`h' if `touse'
qui replace `karg' = .
if "`kern'" == "biweight" {
local con1 = .9375
qui replace `karg' = `con1'*(1-(`arg')^2)^2 /*
*/ if `touse' & abs(round(`arg',1e-8))<1
}
```

Exhibit B.7.2 *A block of interpreted code.*

```
2220 7b0d 0a09 0971 7569 2072 6570 6c61
6365 2060 6b61 7267 2720 3d20 302e 3520
6966 2061 6273 2872 6f75 6e64 2860 6172
6727 2c31 652d 3829 293c 3120 2620 6074
6f75 7365 270d 0a09 7d0d 0a09 656c 7365
207b 2009 0909 092f 2f20 6570 616e 6563
686e 696b 6f76 0d0a 0909 6c6f 6361 6c20
636f 6e31 203d 2033 2f28 342a 7371 7274
```

Of course there is a readable version of the Mata code, before it is compiled into bytecode that is optimized for the computer to read. You write Mata in the readable version, and then Stata compiles into the bytecode before running it. The first time you run your newly written programs, there may be only a small speed advantage, but on subsequent occasions, it will run perhaps 2 to 10 times faster.

The syntax of Mata is quite different from Stata's "ado language" but still fairly easy, especially if you have used C or C++ before.

B.7.3 Interactive Use

You can enter Mata in several ways; but the easiest is type `mata` and then type Mata commands until you are done in Mata, then type `end`.

For example, to multiply two matrices, you could type in Stata:

```
matrix a=(1,2,3)
matrix aa=a'*a
matrix list aa
```

or you could type:

```
mata
a=(1,2,3)
a'a
end
```

to get the same output from Mata.

Note that to get the output of some calculation, you can just type the calculation. To assign it to some named object, just type name=result. We could have also typed:

```
mata
a=(1,2,3)
aa=a'a
aa
end
```

where we first assign the result to a named entity, then get Mata to display the named entity by simply typing its name (the simplest form of calculation, if you will, is just the identity function).

Or we could type:

```
mata
a=(1,2,3)
aa=a'a
st_matrix("A", aa)
end
matrix list A
```

where we create a matrix A in Stata (from the Mata matrix aa), exit Mata, and then display the resulting calculation in Stata. Or we could:

```
matrix a=(1,2,3)
mata
a=st_matrix("a")
aa=a'a
st_matrix("A", aa)
end
matrix list A
```

This suggests the basic procedure for most interactions—do all the calculations you can in Mata, to improve speed, then return them in Stata.

Mata has all the matrix functions of Stata, plus a lot more. Importantly, there is no limit on the size of matrices, and matrices can contain strings or complex numbers.

B.7.4 Defining New Mata Functions and Type Declarations

You make a new function in Mata like so:

```
type name(arguments)
{
contents
}
```

where type is the type of entity the function returns, such as a matrix, or the special type void, meaning the function returns nothing. The type of a Mata variable has two pieces, the element type and organizational type (help [M-2] declarations) and you can choose any one from each column:

```
eltype       orgtype
------------ ----------
transmorphic matrix
numeric      vector
real         rowvector
complex      colvector
string       scalar
pointer      ------------
----------
```

The default is the most general type: a transmorphic matrix, where transmorphic means that the matrix can be real, complex, string, or pointer; and matrix means that the organization is to be $r \times c$, $r \geq 0$ and $c \geq 0$. Here's an example function that swaps rows in a matrix:

```
real matrix swaprows(real matrix A, real scalar i1, real scalar i2)
{
B = A
v = B[i1, .]
B[i1, .] = B[i2, .]
B[i2, .] = v
return(B)
}
```

but all those type declarations are optional, so you could also write:

```
swaprows(A, i1, i2)
{
B = A
v = B[i1, .]
B[i1, .] = B[i2, .]
B[i2, .] = v
return(B)
}
```

Every variable created within a Mata function is "private" meaning its scope is just within the function, like a local macro. You can also declare variables to be global by using the external declaration, which is a useful but dangerous way of passing info back and forth between programs. Useful because you don't have to pass arguments, dangerous because if you accidentally modify a global variable, it's gone. For example, in a complex program, if you create two globals with the same name, you will be replacing the first with the wrong values (the values of the second), and may get some errors that are hard to track down to their source.

B.7.5 Void Functions

One common type of function is the void function, which returns nothing. For example,

```
void iv_pois(todo,b,crit,g,H)
{
external y,X,Z,W
m=((1/rows(Z)):*Z'((y:*exp(-X*b') :- 1)))'
crit=(m*W*m')
}
```

declares that the function iv_pois has five arguments (two of which are used inside the function, but all must be well-defined for the function to operate without spitting out an error message). Then it says there are four globals that it's going to use, does some calculations to modify some variable crit, and that's it. That particular function calculates the value of a Generalized Method of Moments objective function, also called a criterion function.

B.7.6 GMM Estimation Using Mata

The Generalized Method of Moments (GMM), briefly, supposes that there are some population moments that we can try to match in our sample by a good choice of some parameter vector, e.g. in OLS where the population model is $E(y) = X\beta$ or with a mean-zero error term $y = X\beta + \varepsilon$ the usual assumption is that $E(X'\varepsilon) = 0$ in the population. So we can try to make the sample analog true via a good choice of b in an equation $y = Xb + e$ i.e. we try to make the mean of the vector $m = X'e$ as close to zero as possible. The justification for this approach is given by Hansen (1982), and in Stata 11, there is a new command that automates this type of estimation, but walking through it in Mata is a useful exercise.

It's not always possible to make a vector equal to zero, so instead we try to make the sum of squared deviations from zero as small as possible, i.e. instead of choosing b to set m=0 we choose b to minimize (m'Wm) where W is some weight matrix. Each W defines a different GMM estimator. Choosing a weight matrix to be the inverse variance estimate leads to the most efficient GMM estimator. The instrumental variables (IV) version is a population model $y = X\beta + \varepsilon$ plus the assumption that $E(Z'\varepsilon) = 0$ in the population. We can try to make the sample analog true via a good choice of b in the equation $y = Xb + e$ i.e. we try to make the mean of $m = Z'e = Z'(y - Xb)$ as close to zero as possible by minimizing the real-valued function $m'Wm$ with our choice of b.

As another example, consider the population model

$$E(y) = \exp(X\beta)$$

which is the assumption in Poisson regression (like regressing the log of y on X, but y=0 is not a problem). Usually we specify a non-negative error term ε with mean one like so:

$$y = \exp(X\beta)\varepsilon$$

Note that if $\nu = \ln(\varepsilon)$ then $y = \exp(X\beta)exp(\nu) = \exp(X\beta + \nu)$ so $ln(y) = X\beta+\nu$ if $y > 0$, which is why Poisson regression of y on X is like OLS regression of the log of y on X.

The assumption for an IV version of Poisson regression is that $E(Z'\xi) = 0$ in the population, where ξ_i is the mean-zero error $y_i/\exp(X_i\beta) - 1$ (see also the help file for `ivpois` on SSC). So we can try to make the sample analog true via a good choice of b in the equation

$$y = \exp(Xb)e$$

i.e. we try to make the mean of $m = Z'u = Z'(y/\exp(Xb) - 1)$ as close to zero as possible. Using a weight matrix $W = (Z'Z)^{-1}$, we minimize the real-valued function $m'Wm$ by choosing b.

It turns out to be pretty easy in Mata to minimize or maximize some arbitrary function. Having already defined a function iv_pois that takes a parameter b and calculates the real valued function crit=(m'Wm) given b, we just need to tell Mata that we want to minimize crit with our choice of b.

We need Z and X and y matrices: these are matrices of data, which we can form from the variables on a Stata dataset like so:

```
clear all
use http://pped.org/card
g tousename=!mi(wage,educ,nearc4)
mata
y = st_data(., "wage", "tousename")
X1 = st_data(., "educ", "tousename")
Z1 = st_data(., "nearc4", "tousename")
```

Then we can add a constant term to both the X and Z matrices like so:

```
cons=J(rows(X1),1,1)
X = X1, cons
Z = Z1, cons
```

and calculate the weight matrix W:

```
W=rows(Z)*cholinv(Z'Z)
```

Then pick some arbitrary starting vector, say a vector of zeros:

```
init=J(1,cols(X),0)
```

where the only real limitation on a starting vector in numerical optimization is that the objective function has to be defined in a neighborhood of that starting value, and that is always true in this problem for a zero vector, since $\exp(Xb) = 1$ for $b = 0$ and $\exp(Xb)$ is smooth in b at $b = 0$.

Then we declare a new variable to hold our optimization problem, and call a few functions to set up the type of optimization we want. Then the function optimize() finds the optimum and returns the vector b that optimizes the function. Let's just assign that to a new variable p. We can report the result by just typing p in Mata, or use p in subsequent calculations.

```
S=optimize_init()
optimize_init_evaluator(S, &iv_pois())
optimize_init_which(S,"min")
optimize_init_evaluatortype(S,"d0")
optimize_init_params(S,init)
p=optimize(S)
p
```

The evaluator is designated as iv_pois in the second line, and we specify minimization in the third line. The fourth line is more interesting. "d" type evaluators maximize or minimize a real-valued function, and "v" type evaluators a vector-valued function.

```
evaluatortype   Description
        "d0"    function() returns scalar value
        "d1"    same as "d0" and returns gradient row vector
        "d2"    same as "d1" and returns Hessian matrix
        "v0"    function() returns column vector value
        "v1"    same as "v0" and returns score matrix
        "v2"    same as "v1" and returns Hessian matrix
```

In some statistical applications, "v" type evaluators are more convenient to code than "d" type, particularly since one tends to think of a dataset of values arranged in matrix X, the rows of which are observations. A function h(p, X[i,.]) of a parameter p can be calculated for each row i separately, as with a log likelihood, and it is the sum of those resulting values that forms the function f(p) that is to be maximized or minimized. All of Stata's maximum likelihood routines are being rewritten in this way.

The previous optimization problem can all be wrapped in a function that can be called from within Stata like so:

```
void i_pois(string scalar depvar, string scalar x,
 string scalar z, string scalar tousename, string scalar beta)
{
external y,X,Z,W
y = st_data(., tokens(depvar), tousename)
X1 = st_data(., tokens(x), tousename)
Z1 = st_data(., tokens(z), tousename)
cons=J(rows(X1),1,1)
X = X1, cons
Z = Z1, cons
W=rows(Z)*cholinv(Z'Z)
init=J(1,cols(X),0)
b=init
S=optimize_init()
optimize_init_evaluator(S, &iv_pois())
optimize_init_which(S,"min")
optimize_init_evaluatortype(S,"d0")
optimize_init_params(S,init)
p=optimize(S)
st_replacematrix(beta,p)
}
```

where now we pass to the function i_pois the list of variables that form X, Z, and y, and an indicator for which observations we want to include (tousename), then once the optimization is done, we store the parameter vector in the Stata matrix designated "beta" (the name of which we also passed as an argument to the function i_pois). The above is most of the SSC program ivpois (ssc install ivpois). The remainder is a matter of parsing options, error-checking, and returning results.

So a new GMM estimator that you read about in the econometrics literature is a matter of a few hours to program, once you get the hang of the most common elements of the language (i.e. the first program might take a matter of a day or two to write, but the second an hour or two).

B.7.7 Solving Functions

The foregoing GMM discussion also suggests a way to solve some complicated function, e.g. $f(p) = g(q)$ where f and g are real-valued functions of row vectors p and q, writing $h(r) = f(p) - g(q) = 0$ where r = (p,q), and then we can find solutions (or near-solutions for insoluble problems) by minimizing $h(r)^2$ with our choice of r.

Here's a silly example to calculate the solution to $\ln(r) = 0$

```
void obj(todo,b,crit,g,H)
{
crit=ln(b)^2
}
```

```
init=5
S=optimize_init()
optimize_init_evaluator(S, \&obj())
optimize_init_which(S,"min")
optimize_init_evaluatortype(S,"d0")
optimize_init_params(S,init)
optimize(S)
```

Note in the above that if you started at 0 you would get an error in this case, since ln(0) is undefined. Everything would look fine until the last command, when you would see:

```
: optimize(S)
initial values not feasible
r(1400);
```

In general, of course, there is no guarantee that an optimum is unique, so if it is possible that there are multiple solutions to $h(r) = 0$ you would have to start the optimization at many appropriately chosen starting points to see if the optimizer finds different optimal parameter vectors.

Suppose you wanted to find the zeros of

$$y = x^2 - 5x + 4$$

and you were too lazy to use the quadratic formula (I chose this problem, of course, because its analytic solution is so easy, whereas the numerical methods are trickier). You could just type

```
void q(todo,b,crit,g,H)
{
crit=(b^2-5*b+4)^2
}
sol=J(1,0,0)
void grid(n1,n2)
{
external sol, p
for (i=n1; i<=n2; i++) {
init=i
S=optimize_init()
optimize_init_evaluator(S, &q())
optimize_init_which(S,"min")
optimize_init_evaluatortype(S,"d0")
optimize_init_params(S,init)
p=round(optimize(S),10e-4)
if (!anyof(sol, p)) {
sol=(sol,p)
}
}
sol
}
grid(-10,10)
```

The above code does the optimization for every starting value on a grid over the integers -10 to 10 (and saves any solutions that are new to a Mata variable sol, then reports its values). A function that took a vector as an argument would be no harder to specify (though the appropriate grid search might be harder to specify).

A user-written program `mm_root()` (in the moremata package by Ben Jann on SSC) also finds zeros of a function, also using Mata. Sometimes using mm_root() to find a zero of f will be a lot faster than using official Stata's optimize() to find a min of f^2. For example, finding the modal kth order statistic in a sample of n draws from a standard normal distribution, which is given by the zero of

```
f(x)=(k-1)*(1-normal(x))*normalden(x)
  -(n-k)*normalden(x)*normal(x)-normal(x)*(1-normal(x))*x}
```

per Gupta (1961); see also Guenther (1977). In this case mm_root() is far superior.

For n more than about 13, and k=n, optimize() skips over interior solutions and skitters off toward a very large x where normalden(x) and 1-normal(x) both approach zero. There is an alternative parametrization

```
f=(k-1)/normal(x)-(n-k)/normal(-x)-x/normalden(x)}
```

for which `optimize()` performs as well as `mm_root()`, in terms of finding the right answer, but `optimize()` takes about three times as long as `mm_root()` even in this case.

The example also illustrates a common pitfall in such problems, which frequent users of `ml` know well—quantities like 1-normal(x) can easily reach zero as x gets large even while the equivalent normal(-x) is nonzero. Similar problems come up in many calculations, where numbers very close to zero can be stored and manipulated but numbers equally close to another integer cannot; for example, the logit of a very small number can easily be displayed and manipulated, but the logit of a number close to one cannot.

```
. di invlogit((logit(1e-300)))
1.00e-300
. di invlogit(abs(logit(1e-300)))
1
```

Appendix C

Human Capital

A motivating example throughout the book is the effect of education on earnings, partly because anyone reading this book presumably has made some careful choices about education and likely has thought a bit about the effect of those choices on future earnings potential, and partly because the problem has a long and storied history in economics and econometrics.

This appendix is just intended to give a flavor of the kind of reasoning common in economics when dealing with a thorny estimation problem where outcome and the treatment are both affected by some third factor. The example, measuring the effect of education on subsequent earnings, is the paradigmatic thorny estimation problem in economics, and many authors have explained these types of models, including Willis (1987), Card (1999), Kling (2001), and Card (2001).

We start by making a lot of convenient assumptions, so the model is easy to solve:

- there is no uncertainty, and individuals live forever.
- capital markets are perfect, and an individual i may borrow against future earnings at a rate r_i, so individuals act to maximize lifetime earnings.
- individuals have marginal productivity linearly increasing in completed schooling.
- competitive firms offer wages equal to observed marginal productivity.
- work intensity is constant, so there is no labor-leisure tradeoff, and people either work or go to school.

- schooling occurs at the beginning of life and ends once work begins (a bang-bang solution).
- earnings are constant over the life cycle after schooling is completed.

Since an individual's observed earnings at a point in time are a linear function of completed schooling:

$$y_i = a_i + b_i s_i$$

the natural estimation strategy is to regress individual earnings on education. But regressing earnings on schooling in a heterogeneous population will produce very different answers depending on the source of heterogeneity. Usually, those who have a higher return b_i will also choose a higher s_i, and those who have higher baseline earnings a_i will choose a lower s_i, and it seems likely that b_i and a_i are correlated.

Suppose the only cost of schooling is forgone earnings y_i. Then an individual chooses s_i (the time to end schooling and begin work) to maximize

$$V(s_i) = \int_{s_i}^{\infty} y_i(s_i) e^{-r_i t} dt$$

subject to $s_i \geq 0$. Call the optimal choice $\tilde{s}_i$, and for everyone with $\tilde{s}_i > 0$ we have the first order condition

$$-y_i(\tilde{s}_i) e^{-r_i \tilde{s}_i} + \int_{\tilde{s}_i}^{\infty} y_i'(\tilde{s}_i) e^{-r_i t} dt = 0$$

or

$$\frac{y_i'(\tilde{s}_i)}{r_i} = y_i(\tilde{s}_i)$$

so

$$\frac{b_i}{r_i} = a_i + b_i \tilde{s}_i$$

where the left side of the equation is the marginal return to more schooling (b_i dollars more forever, worth $\frac{b_i}{r_i}$) and the right side is the marginal cost (one time period of earnings lost).

We are abstracting from many important features of real decisions about schooling and labor markets, of course, but the point here is to focus only on the simplest form of worker heterogeneity. Note that individuals have different earnings because of differences in three parameters: a_i or baseline ability (marginal productivity in the absence of education), b_i or returns to education, and r_i or borrowing costs. Let the average return to schooling be $b = E(b_i)$, and likewise let $a = E(a_i)$ and $r = E(r_i)$.

The population version of the regression we are estimating when we regress individual earnings on education is $Y_i = \alpha + \beta X_i + \varepsilon_i$ and let us suppose ε_i is uncorrelated noise (e.g. from classical measurement error in earnings), so

$$Y_i = \left(\frac{b_i}{r_i}\right) + \varepsilon_i = \alpha + \beta\left(\frac{1}{r_i} - \frac{a_i}{b_i}\right) + \varepsilon_i$$

Suppose we are interested in estimating the average return to schooling b.

Even if X_i (schooling, measured without error) is uncorrelated with ε_i, the OLS estimator $\widehat{\beta}$ need not be unbiased or consistent for b since

$$E(\widehat{\beta}) = \frac{\text{Cov}\left(\frac{b}{r_i}, \frac{1}{r_i} - \frac{a_i}{b_i}\right)}{\text{Var}\left(\frac{1}{r_i} - \frac{a_i}{b_i}\right)}$$

and this need not equal b.

First, suppose there is no variation in b_i and a_i (i.e. $b_i = b$ and $a_i = a$ for all i), so all the variation in earnings arises from variation in r_i. Then the OLS estimator $\widehat{\beta}$ is an unbiased estimate of b since

$$E(\widehat{\beta}) = \frac{\text{Cov}\left(\frac{b}{r_i}, \frac{1}{r_i} - \frac{a}{b}\right)}{\text{Var}\left(\frac{1}{r_i} - \frac{a}{b}\right)} = \frac{b\text{Var}\left(\frac{1}{r_i}\right)}{\text{Var}\left(\frac{1}{r_i}\right)} = b.$$

Now suppose there is no variation in r_i and a_i (i.e. $r_i = r$ and $a_i = a$ for all i), so all the variation in earnings arises from variation in returns b_i. Then

$$E(\widehat{\beta}) = \left(\frac{r}{a}\right)\frac{\left[bE\left(\frac{1}{b_i}\right) - 1\right]}{\text{Var}\left(\frac{1}{b_i}\right)}$$

and the OLS estimator $\widehat{\beta}$ is generically not an unbiased estimate of b. We cannot even sign the bias, and even if $r = a$, the bias will vary wildly depending on the distribution of b_i.

If b_i is distributed lognormally, say $b_i = e^{z_i}$ with $z_i \sim N(-3, 0.25)$, then $b = E(b_i) = .064$ and $E(\frac{1}{b_i}) = 26$ and $\text{Var}\left(\frac{1}{b_i}\right) = 431$, so if $r = a$, $E(\widehat{\beta}) = 0.0015$ and the expected estimate is about two percent of the true b. More generally, with $z_i \sim N(m, v)$, then $b = E(b_i) = e^{m+v/2}$ and $E(\frac{1}{b_i}) = e^{-m+v/2}$ and $\text{Var}\left(\frac{1}{b_i}\right) = (e^v - 1)\, e^{-2m+v}$, so the proportional bias is given by

$$\frac{E(\widehat{\beta}) - b}{b} = e^{m-\frac{3}{2}v} - 1$$

which for $m = -3$ is always between 95 and 100 percent (i.e. the estimate will never be more than 5 percent of b).

In reality, there is unobserved heterogeneity in b_i, a_i, and r_i, and these parameters are not independently distributed, so there is no guarantee that unobserved heterogeneity in r_i will not exacerbate bias. It seems likely that b_i and a_i are positively correlated (those who start with higher skill also learn faster), but in the presence of various kinds of measurement error, it is likely that many estimates of b_i and a_i would be negatively correlated. In a richer model, it might also make sense that those with higher b_i have lower r_i (because they may live longer, exhibit lower risk of default on loans, etc.).

References

A. Abadie and G.W. Imbens. On the failure of the bootstrap for matching estimators. *NBER Working Paper T0325*, 2006.

A. Abadie, D. Drukker, J.L. Herr, and G.W. Imbens. Implementing matching estimators for average treatment effects in Stata. *Stata Journal*, 4:290–311, 2004.

J.H. Abbring and G.J. Van den Berg. The identifiability of the mixed proportional hazards competing risks model. *Journal of the Royal Statistical Society. Series B, Statistical Methodology*, pages 701–710, 2003a.

J.H. Abbring and G.J. Van den Berg. The nonparametric identification of treatment effects in duration models. *Econometrica*, pages 1491–1517, 2003b.

A. Agresti and B.A. Coull. Approximate Is Better Than 'Exact' for Interval Estimation of Binomial Proportions. *The American Statistician*, 52(2):119–126, 1998.

C. Ai and E.C. Norton. Interaction terms in logit and probit models. *Economics Letters*, 80(1):123–129, 2003.

A.C. Aitken. On least squares and linear combinations of observations. *Proceedings of the Royal Society of Edinburgh*, 55:42–48, 1935.

J.G. Altonji and R.L. Matzkin. Cross section and panel data estimators for nonseparable models with endogenous regressors. *Econometrica*, 73(4):1053–1102, 2005.

Joseph G. Altonji, Prashant Bharadwaj, and Fabian Lange. Changes in the characteristics of American youth: Implications for adult outcomes. *Yale University Working Paper*, 2008. URL http://www.econ.yale.edu/~fl88/SkillComposition.pdf.

T. Amemiya. Tobit Models: A Survey. *Journal of Econometrics*, 24(1/2):3–61, 1984.

T. Amemiya and T.E. MaCurdy. Instrumental-variable estimation of an error-components model. *Econometrica*, 54:869–880, 1986.

TW Anderson and H. Rubin. Estimation of the parameters of a single equation in a complete system of stochastic equations. *The Annals of Mathematical Statistics*, 21:46–63, 1949.

J.D. Angrist and A.B. Krueger. Does compulsory school attendance affect schooling and earnings? *The Quarterly Journal of Economics*, 106(4):979–1014, 1991.

J.D. Angrist, G.W. Imbens, and D.B. Rubin. Identification of causal effects using instrumental variables. *Journal of the American Statistical Association*, 91 (434):444–472, 1996.

Joshua D. Angrist and Alan B. Krueger. Instrumental Variables and the Search for Identification: From Supply and Demand to Natural Experiments. *Journal of Economic Perspectives*, 15:69–85, 2001.

R.J. Apfel and S.M. Fisher. *To do no harm: DES and the dilemmas of modern medicine*. Yale University Press, 1986.

Tom Apostol. *Mathematical Analysis*. Addison Wesley, second edition, 1974.

M. Arellano and S. Bond. Some tests of specification for panel data: Monte Carlo evidence and an application to employment equations. *The Review of Economic Studies*, 58:277–297, 1991.

Wiji Arulampalam and Mark B. Stewart. Simplified Implementation of the Heckman Estimator of the Dynamic Probit Model and a Comparison with Alternative Estimators. *Oxford Bulletin of Economics and Statistics*, 71(5):659–681, 2009.

O. Ashenfelter. Estimating the effect of training programs on earnings. *The Review of Economics and Statistics*, 60(1):47–57, 1978.

S. Athey and G.W. Imbens. Identification and inference in nonlinear difference-in-differences models. *Econometrica*, 74(2):431–497, 2006.

D. Autor, L.F. Katz, and M.S. Kearney. Rising wage inequality: the role of composition and prices, 2005. URL http://www.nber.org/papers/w11628.

Harald Badinger and Peter Egger. Estimation of Higher-Order Spatial Autoregressive Panel Data Error Component Models . *CESifo Working Paper No. 2556*, 2009. URL http://www.cesifo.de/DocCIDL/cesifo1_wp2556.pdf.

M. Baker, D. Benjamin, and S. Stanger. The highs and lows of the minimum wage effect: A time-series cross-section study of the Canadian law. *Journal of Labor Economics*, 17(2):318–350, 1999.

T.K. Bauer and M. Sinning. An extension of the Blinder–Oaxaca decomposition to nonlinear models. *Advances in Statistical Analysis*, 92(2):197–206, 2008.

C.F. Baum, M.E. Schaffer, and S. Stillman. Enhanced routines for instrumental variables/GMM estimation and testing. *Stata Journal*, 7(4):465–506, 2007.

M. J. Bayarri and James O. Berger. The Interplay of Bayesian and Frequentist Analysis. *Statistical Science*, 19(1):58–80, 2004.

April Beaule, Kate McGonagle, Austin Nichols, and Bob Schoeni. The Impact of Asking for Identification Numbers on Response Propensity in the PSID. *Presented at the UM Survey Methodology seminar in Ann Arbor, MI, March 5, 2009*, 2009.

Sascha O. Becker and Marco Caliendo. mhbounds: Sensitivity Analysis for Average Treatment Effects. *IZA Discussion Paper*, No.2542, 2007. URL http://ftp.iza.org/dp2542.pdf.

S.O. Becker and A. Ichino. Estimation of average treatment effects based on propensity scores. *The Stata Journal*, 2(4):358–377, 2002.

Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 289–300, 1995.

Craig M. Bennett, Abigail A. Baird, Michael B. Miller, and George L. Wolford. Neural correlates of interspecies perspective taking in the post-mortem Atlantic Salmon: An argument for multiple comparisons

correction. *Presented at the Human Brain Mapping conference*, June, 2009. URL http://prefrontal.org/files/posters/Bennett-Salmon-2009.pdf.

James O. Berger. Could Fisher, Jeffreys and Neyman Have Agreed on Testing? *Statistical Science*, 18(1):1–12, 2003.

Michela Bia and Alessandra Mattei. A Stata package for the estimation of the doseresponse function through adjustment for the generalized propensity score. *Stata Journal*, 8(3):354–373, 2008.

D.A. Binder and G.R. Roberts. Design-based and model-based methods for estimating model parameters. In R. L. Chambers and C. J. Skinner, editors, *Analysis of Survey Data*, pages 29–48, 2003.

D.A. Binder, M. Kovacevic, and G. Roberts. Design-based methods for survey data: Alternative uses of estimating functions. In *Proceedings of the Section on Survey Research Methods*, 2004. URL https://www.amstat.org/sections/SRMS/Proceedings/y2004/Files/Jsm2004-000690.pdf.

Sandy Black. Do Better Schools Matter? Parental Valuation of Elementary Education. *Quarterly Journal of Economics*, 114:577–599, 1999.

Michael Blasnik. Mass producing appendices using Stata and word processor mail merge. *Presentation at North American Stata Users' Group Meetings*, 2005 (July 11), 2005. URL http://www.stata.com/meeting/4nasug/mblasniknasug.ppt.

Alan S. Blinder. Wage Discrimination: Reduced Form and Structural Estimates. *The Journal of Human Resources*, 8(4):436–455, 1973.

R. Blundell and S. Bond. Initial conditions and moment restrictions in dynamic panel data models. *Journal of econometrics*, 87(1):115–143, 1998.

Richard W. Blundell and James L. Powell. Endogeneity in nonparametric and semiparametric regression models. In M. Dewatripont, L.P. Hansen, and S. J. Turnovsky, editors, *Advances in Economics and Econometrics: Theory and Applications*, volume 2, pages 312–357. New York: Cambridge University Press, 2003.

George J. Borjas. The Relationship Between Wages and Weekly Hours of Work: The Role of Division Bias. *Journal of Human Resources*, 33(4):409–423, 1980.

John Bound, David A. Jaeger, and Regina Baker. Problems with Instrumental Variables Estimation when the Correlation Between the Instruments and the Endogenous Explanatory Variables is Weak. *Journal of the American Statistical Association*, 90(430):443–450, 1995.

J.H. Boyd III. Symmetries, dynamic equilibria, and the value function. In R. Sato and R. Ramachandran, editors, *Conservation Laws and Symmetry: Applications to Economics and Finance*, pages 225–259. Boston:Kluwer, 1990.

G.H. Bracht and G.V. Glass. The external validity of experiments. *American Educational Research Journal*, 5(4):437–474, 1968.

R. Breen. *Regression Models: Censored, Sample Selected Or Truncated Data*. Sage Publications, 1996.

L.D. Brown, T.T. Cai, and A. DasGupta. Interval Estimation for a Binomial Proportion. *Statistical Science*, 16(2):101–116, 2001.

Thomas L. Brunell and John DiNardo. A Propensity Score Reweighting Approach to Estimating the Partisan Effects of Full Turnout in American Presidential Elections. *Political Analysis*, 12:28–45, 2004.

G.S.F. Bruno. Estimation and inference in dynamic unbalanced panel-data models with a small number of individuals. *The Stata Journal*, 5(4):473–500, 2005.

John DiNardo Busso, Matias and Justin McCrary. Finite sample properties of semiparametric estimators of average treatment effects. *Working paper*, 2009a. URL http://www.econ.berkeley.edu/~jmccrary/BDM_JBES.pdf.

John DiNardo Busso, Matias and Justin McCrary. New Evidence on the Finite Sample Properties of Propensity Score Matching and Reweighting Estimators. *Working paper*, 2009b. URL http://www.econ.berkeley.edu/~jmccrary/BDM2009.pdf.

E. Cameron and L. Pauling. Supplemental Ascorbate in the Supportive Treatment of Cancer: Prolongation of Survival Times in Terminal Human Cancer. *Proceedings of the National Academy of Sciences*, 73(10):3685–3689, 1976.

D.T. Campbell and J.C. Stanley. Experimental and Quasi-Experimental Designs for Research. In N.L. Gage, editor, *Handbook of Research on Teaching*. Chicago, IL: Rand McNally, 1963.

D. Card, C. Dobkin, and N. Maestas. The Impact of Nearly Universal Insurance Coverage on Health Care Utilization: Evidence from Medicare. *American Economic Review*, 98(5):2242–2258, 2008.

D. Card, C. Dobkin, and N. Maestas. Does Medicare Save Lives? *Quarterly Journal of Economics*, 124(2):597–636, 2009.

David E. Card. Estimating the Return to Schooling: Progress on Some Persistent Econometric Problems. *Econometrica*, 69(5):1127–1160, 2001.

David E. Card. Using Geographic Variation in College Proximity to Estimate the Return to Schooling. In E. Kenneth Grant Louis Christofides and Robert Swindinsky, editors, *Aspects of Labour Economics: Essays in Honour of John Vanderkamp*. University of Toronto Press, 1995.

David E. Card. The Causal Effect of Education on Earnings. In Orley Ashenfelter and David Card, editors, *Handbook of Labor Economics*, volume 3A, pages 1801–1863. North-Holland, 1999.

J. B. Carlin, J. C. Galati, and P. Royston. A new framework for managing and analyzing multiply imputed data in Stata. *Stata Journal*, 8:49–67, 2008.

Richard T. Carson and Yixiao Sun. The Tobit model with a non-zero threshold. *The Econometrics Journal*, 10(3), 2007.

G. Chamberlain. Analysis of covariance with qualitative data. *The Review of Economic Studies*, 47(1):225–238, 1980.

Gary. Chamberlain. Panel Data. In Z. Griliches and M.D. Intriligator, editors, *Handbook of econometrics*, volume 2, pages 1247–1318. North Holland, 1984.

C.R. Charig, D.R. Webb, S.R. Payne, and J.E. Wickham. Comparison of treatment of renal calculi by open surgery, percutaneous nephrolithotomy, and extracorporeal shockwave lithotripsy. *British Medical Journal*, 292(6524):879, 1986.

Ming-Yen Cheng, Jianqing Fan, and James S. Marron. On Automatic Boundary Corrections. *Annals of Statistics*, 25(4):1691–1708, 1997.

V. Chernozhukov and C. Hansen. An IV model of quantile treatment effects. *Econometrica*, 73(1):245–261, 2005.

V. Chernozhukov and C. Hansen. Instrumental quantile regression inference for structural and treatment effect models. *Journal of Econometrics*, 132(2):491–525, 2006.

V. Chernozhukov and C. Hansen. Instrumental variable quantile regression: A robust inference approach. *Journal of Econometrics*, 142(1):379–398, 2008.

V. Chernozhukov, C. Hansen, and M. Jansson. Inference approaches for instrumental variable quantile regression. *Economics Letters*, 95(2):272–277, 2007.

Victor Chernozhukov, Iván Fernández-Val, Jinyong Hahn, and Whitney Newey. Identification and estimation of marginal effects in nonlinear panel data models. *CEMMAP Working Paper*, CWP(25/08), 2008.

M. Cleves. Analysis of multiple failure-time data with Stata. *Stata Technical Bulletin*, 49:30–39, 1999.

M. Cleves, W. Gould, R. Gutierrez, and Y. Marchenko. *An introduction to survival analysis using Stata*. Stata Press, 2008.

W.G. Cochran and G.M. Cox. *Experimental designs*. New York: John Wiley and Sons, fourth edition, 1957.

William G. Cochran and Donald B. Rubin. Controlling Bias in Observational Studies: A Review. *Sankhya*, 35:417–46, 1973.

D.R. Cox. *The planning of experiments*. 1958.

DR Cox. The analysis of exponentially distributed life-times with two types of failure. *Journal of the Royal Statistical Society Series B*, 21:411–421, 1959.

DR Cox. Renewal Theory. *Methuen and Co. Ltd., London*, 1962.

D.R. Cox. Regression models and life-tables. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 187–220, 1972.

N. J. Cox. Speaking stata: Identifying spells. *Stata Journal*, 7(2):249–265, 2007. URL http://www.stata-journal.com/article.html?article=dm0029.

Harald Cramér. *Mathematical Methods of Statistics*. Princeton, NJ: Princeton Univ. Press, 1946.

L.G. Crespo and J.Q. Sun. Stochastic optimal control of nonlinear systems via short-time Gaussian approximation and cell mapping. *Nonlinear Dynamics*, 28 (3):323–342, 2002.

L.M. Cruz and M.J. Moreira. On the validity of econometric techniques with weak instruments: Inference on returns to education using compulsory school attendance laws. *Journal of Human Resources*, 40(2):393, 2005.

F. Cunha, J.J. Heckman, and S. Navarro. The identification and economic content of ordered choice models with stochastic thresholds. *International Economic Review*, 48(4):1273–1309, 2007.

M. Das, W.K. Newey, and F. Vella. Nonparametric estimation of sample selection models. *Review of Economic Studies*, 70(1):33–58, 2003.

Russell Davidson and James G. MacKinnon. Moments of IV and JIVE estimators. *Econometrics Journal*, 60(3):541–553, 2007.

A. Deaton. *The Analysis of Household Surveys: A Microeconometric Approach to Development Policy, 1997, The World Bank*. Johns Hopkins University Press, 1997.

John DiNardo. Propensity Score Reweighting and Changes in Wage Distributions. *University of Michigan Working Paper*, 2002. URL http://www-personal.umich.edu/~jdinardo/bztalk5.pdf.

John DiNardo and David Lee. The Impact of Unionization on Establishment Closure: A Regression Discontinuity Analysis of Representation Elections. *NBER Working Paper*, No.8993, 2002. URL http://nber.org/papers/w8993.

John DiNardo, Nicole M. Fortin, and Thomas Lemieux. Labor Market Institutions and the Distribution of Wages, 1973-1992: A Semiparametric Approach. *Econometrica*, 64(5):1001–1044, 1996.

T.A. DiPrete and M. Gangl. Assessing bias in the estimation of causal effects: Rosenbaum bounds on matching estimators and instrumental variables estimation with imperfect instruments. *Sociological Methodology*, pages 271–310, 2004.

D. Drukker. Analyzing spatial autoregressive models using Stata. *Presentation at the Summer North American Stata Users Group meeting*, July 24-25, 2008. URL http://repec.org/snasug08/drukker_spatial.pdf.

W. Easterly, R. Levine, and D. Roodman. Aid, policies, and growth: comment. *American Economic Review*, 94(3):774–780, 2004.

Jianqing Fan and Irene Gijbels. *Local Polynomial Modelling and Its Applications*. 1996.

R. P. Feynman. *Surely You're Joking, Mr. Feynman!* Norton, 1997.

J.P. Fine and R.J. Gray. A Proportional Hazards Model for the Subdistribution of a Competing Risk. *Journal of the American Statistical Association*, 94(446): 496–509, 1999.

S. Firpo. Efficient semiparametric estimation of quantile treatment effects. *Econometrica*, 75(1):259–276, 2007.

R.A. Fisher. *The Design of Experiments*. Edinburgh: Oliver and Boyd, 1935.

Ronald Aylmer Fisher. The arrangement of field experiments. *Journal of the Ministry of Agriculture of Great Britain*, 33:503–513, 1926.

David Freedman. Statistical Models and Shoe Leather. *Sociological Methodology*, 21:291–313, 1991.

Markus Frölich. Propensity Score Matching without Conditional Independence Assumption With an Application to the Gender Wage Gap in the United Kingdom. *Econometrics Journal*, 10(2):359–407, 2007.

Markus Frölich and Blaise Melly. Quantile Treatment Effects in the Regression Discontinuity Design. *IZA Working Paper*, No.3638, 2008. URL http://ftp.iza.org/dp3638.

W. A. Fuller. Some properties of a modification of the limited information estimator. *Econometrica*, 45(4):939–953, 1977.

M. Gail. A review and critique of some models used in competing risk analysis. *Biometrics*, pages 209–222, 1975.

Steven Glazerman, Dan M. Levy, and David Myers. Nonexperimental versus experimental estimates of earnings impacts. *The Annals of the American Academy of Political and Social Science*, 589:63–93, 2003.

Arthur S. Goldberger and Otis D. Duncan. *Structural Equation Models in the Social Sciences*. Seminar Press, New York, 1973.

Joanna Gomulka and Nicholas Stern. The Employment of Married Women in the United Kingdom 1970-83. *Economica*, 57:171–199, 1990.

T.A. Gooley, W. Leisenring, J. Crowley, and B.E. Storer. Estimation of failure probabilities in the presence of competing risks: New representations of old estimators. *Statistics in Medicine*, 18(6):695–706.

W.S. Gosset. The probable error of a mean. *Biometrika*, 6(1):1–25, 1908.

Bryan S. Graham and James Powell. Identification and Estimation of 'Irregular' Correlated Random Coefficient Models. *NBER Working Paper*, (No. 14469), 2009.

B.I. Graubard and E.L. Korn. Inference for superpopulation parameters using sample surveys. *Statistical Science*, pages 73–96, 2002.

Zvi Griliches and Jerry A. Hausman. Errors in Variables in Panel Data. *Journal of Econometrics*, 31:93–118, 1986. See also the 1984 NBER Technical Working Paper No. 37 at [http://www.nber.org/papers/t0037].

William C. Guenther. An easy method for obtaining percentage points of order statistics. *Technometrics*, 10(3):319–321, 1977.

Shanti S. Gupta. Percentage Points and Modes of Order Statistics from the Normal Distribution. *Ann. Math. Statist.*, 32(3):888–893, 1961.

R. Gutierrez, J. M. Linhart, and J. S. Pitblado. From the help desk: Local polynomial regression and Stata plugins. *Stata Journal*, 3(4):412–419, 2003.

J. Hahn. The efficiency bound of the mixed proportional hazard model. *The Review of Economic Studies*, pages 607–629, 1994.

Jinyong Hahn, Petra Todd, and Wilbert van der Klaauw. Identification and Estimation of Treatment Effects with a Regression-Discontinuity Design. *Econometrica*, 69(1):201–209, 2001.

Alastair R. Hall, Glenn D. Rudebusch, and David W. Wilcox. Judging Instrument Relevance in Instrumental Variables Estimation. *International Economic Review*, 37(2):283–298, 1996.

D.S. Hamermesh. Replication in economics. *Canadian Journal of Economics*, 40 (3):715–733, 2007.

L. P. Hansen. Large Sample Properties of Generalized Method of Moments Estimators. *Econometrica*, 50:1029–1054, 1982.

J. W. Hardin and J. M. Hilbe. *Generalized Linear Models and Extensions, Second edition*. College Station, TX: Stata Press., 2007.

James W. Hardin, Henrik Schmiediche, and Raymond J. Carroll. Instrumental variables, bootstrapping, and generalized linear models. *Stata Journal*, 3(4): 351–360, 2003. See also http://www.stata.com/merror.

J.A. Hausman and W.E. Taylor. Panel data and unobservable individual effects. *Econometrica*, 49:1377–1398, 1981.

J. Heckman. Shadow prices, market wages, and labor supply. *Econometrica: Journal of the Econometric Society*, 42:679–694, 1974.

J. Heckman. The common structure of statistical models of truncation, sample selection and limited dependent variables and a simple estimator for such models. *Annals of Economic and social Measurement*, 5(4):475–492, 1976.

J. Heckman. Varieties of selection bias. *The American Economic Review*, 80: 313–318, 1990.

J. Heckman and E. Vytlacil. Instrumental variables methods for the correlated random coefficient model: Estimating the average rate of return to schooling when the return is correlated with schooling. *Journal of Human Resources*, 33 (4):974–987, 1998.

James J. Heckman and Edward Vytlacil. Structural Equations, Treatment Effects and Econometric Policy Evaluation. *Econometrica*, 73(3):669–738, 2004. See also NBER technical working paper 306 at http://www.nber.org/papers/t0306.

James J. Heckman, Sergio Urzua, and Edward Vytlacil. Understanding Instrumental Variables In Models With Essential Heterogeneity. *Review of Economics and Statistics*, 88(3):389–432, 2006.

J.J. Heckman. Sample selection bias as a specification error. *Econometrica: Journal of the econometric society*, 47:153–161, 1979.

J.J. Heckman. Econometric Causality. *International Statistical Review*, 76(1): 1–27, 2008.

J.J. Heckman and B.E. Honoré. The identifiability of the competing risks model. *Biometrika*, 76(2):325–330, 1989.

J.J. Heckman and B.E. Honoré. The empirical content of the Roy model. *Econometrica*, 58(5):1121–1149, 1990.

J.J. Heckman and E.J. Vytlacil. Local instrumental variables and latent variable models for identifying and bounding treatment effects. *Proceedings of the National Academy of Sciences*, 96(8):4730–4734, 1999.

J.J. Heckman and E.J. Vytlacil. Local instrumental variables. In C. Hsiao, K. Morimune, and J. L. Powell, editors, *Nonlinear Statistical Modeling: Proceedings of the Thirteenth International Symposium in Economic Theory and Econometrics: Essays in Honor of Takeshi Amemiya*, pages 1–46. New York: Cambridge University Press, 2001.

K. Hirano and G. W. Imbens. The propensity score with continuous treatments. In A. Gelman and X.-L. Meng, editors, *Applied Bayesian Modeling and Causal Inference from Incomplete-Data Perspectives*, pages 73–84. West Sussex, England: Wiley InterScience, 2004.

Keisuke Hirano, Guido W. Imbens, and Geert Ridder. Efficient Estimation of Average Treatment Effects Using the Estimated Propensity Score. *Econometrica*, 71(4):1161–1189, 2003.

Paul W. Holland. Statistics and causal inference. *Journal of the American Statistical Association*, 81(396):945–960, 1986a.

Paul W. Holland. Statistics and Causal Inference: Rejoinder. *Journal of the American Statistical Association*, 81(396):968–970, 1986b.

B.E. Honoré. Identification results for duration models with multiple spells. *The Review of Economic Studies*, pages 241–246, 1993.

J.L. Horowitz and S. Lee. Nonparametric instrumental variables estimation of a quantile regression model. *Econometrica*, 75(4):1191–1208, 2007.

Jill R. Horwitz and Austin Nichols. What Do Nonprofits Maximize? Nonprofit Hospital Service Provision and Market Ownership Mix. *NBER Working Paper*, No. 13246, 2007. URL http://www.nber.org/papers/w13246.

Jill R. Horwitz and Austin Nichols. Hospital ownership and medical services: Market mix, spillover effects, and nonprofit objectives. *Journal of health economics*, 2009.

C.M. Hoxby. Does competition among public schools benefit students and taxpayers? Reply. *American Economic Review*, 97(5):2038–2055, 2007.

Stefano M. Iacus, Gary King, and Giuseppe Porro. Matching for causal inference without balance checkinge. 2008. URL http://gking.harvard.edu/cem.

Kosuke Imai and David A. van Dyk. Causal inference with general treatment regimes: Generalizing the propensity score. *Journal of the American Statistical Association*, 99(467):854–866, 2004.

Guido Imbens and Karthik Kalyanaraman. Optimal Bandwidth Choice for the Regression Discontinuity Estimator. *NBER Working Paper*, No.14726, 2009. URL http://www.nber.org/papers/w14726.

Guido W. Imbens. Nonparametric Estimation of Average Treatment Effects Under Exogeneity: A Review. *Review of Economics and Statistics*, 86(1):4–29, 2004. See also NBER technical working paper 294 at http://www.nber.org/papers/t0294.

Guido W. Imbens and Thomas Lemieux. Regression Discontinuity Designs: A Guide to Practice. *Journal of Econometrics*, 142(2):615–635, 2008. See also NBER working paper No.13039 at http://www.nber.org/papers/w13039.

Ben Jann. The Blinder-Oaxaca decomposition for linear regression models. *The Stata Journal*, 8(4):453–479, 2008.

S.P. Jenkins. Survival analysis. *Web book, Institute for Social and Economic Research, University of Essex, Colchester, UK*, 2005.

S.P. Jenkins. Easy estimation methods for discrete-time duration models. *Oxford Bulletin of Economics and Statistics*, 57(1):129–138, 1995.

Chinhui Juhn, Kevin M. Murphy, and Brooks Pierce. Wage Inequality and the Rise in Returns to Skill. *Journal of Political Economy*, 101(3):410–442, 1993.

E. L. Kaplan and P. Meier. Nonparametric estimation from incomplete observations. *Journal of the American Statistical Association*, 53:457–481.

M. Kapoor, H.H. Kelejian, and I.R. Prucha. Panel data models with spatially correlated error components. *Journal of Econometrics*, 140(1):97–130, 2007.

H.H. Kelejian and I.R. Prucha. Specification and Estimation of Spatial Autoregressive Models with Autoregressive and Heteroskedastic Disturbances. *CESifo Working Paper No. 2448*, 2008. URL http://www.cesifo.de/DocCIDL/cesifo1_wp2448.pdf.

H.H. Kelejian and I.R. Prucha. A generalized moments estimator for the autoregressive parameter in a spatial model. *International Economic Review*, 40(2): 509–533, 1999.

H.H. Kelejian and I.R. Prucha. Estimation of simultaneous systems of spatially interrelated cross sectional equations. *Journal of Econometrics*, 118(1-2):27–50, 2004.

M. Kelly. Inequality and crime. *Review of Economics and Statistics*, 82(4):530–539, 2000.

N.M. Kiefer. Economic duration data and hazard functions. *Journal of economic literature*, pages 646–679, 1988.

T. W. Kinal. The existence of moments of k-class estimators. *Econometrica*, 48 (1):241–249, 1980.

J.P. Klein and M.L. Moeschberger. *Survival analysis: techniques for censored and truncated data*. Springer Verlag, 2003.

R. Klein and F. Vella. Estimating a Class of Triangular Simultaneous Equations Models Without Exclusion Restrictions. *Journal of Econometrics*, forthcoming.

R. Klein and F. Vella. A semiparametric model for binary response and continuous outcomes under index heteroscedasticity. *Journal of Applied Econometrics*, 24 (5):735–762, 2009.

J.R. Kling. Interpreting instrumental variables estimates of the returns to schooling. *Journal of Business and Economic Statistics*, 19(3):358–364, 2001.

Andrey Kolmogorov. *Grundbegriffe der Wahrscheinlichkeitsrechnung*. Berlin: Julius Springer, 1933. URL http://www.mathematik.com/Kolmogorov.

T. Lancaster. Econometric methods for the duration of unemployment. *Econometrica: Journal of the Econometric Society*, pages 939–956, 1979.

T. Lancaster. *The econometric analysis of transition data*. Cambridge University Press, 1990.

David S. Lee. The Electoral Advantage to Incumbency and Voters' Valuation of Politicians' Experience: A Regression Discontinuity Analysis of Elections to the U.S. House. *NBER Working Paper*, No. 8441, 1993. URL http://www.nber.org/papers/w8441.

D.S. Lee. Randomized experiments from non-random selection in US House elections. *Journal of Econometrics*, 142(2):675–697, 2008. See also the Supplemental Mathematical Appendix at [http://www.princeton.edu/ davidlee/wp/AppendixC.pdf].

D.S. Lee. Training, Wages, and Sample Selection: Estimating Sharp Bounds on Treatment Effects. *Review of Economic Studies*, 76(3):1071–1102, 2009.

D.S. Lee and D. Card. Regression discontinuity inference with specification error. *Journal of Econometrics*, 142(2):655–674, 2008.

L.F. Lee. Asymptotic distributions of quasi-maximum likelihood estimators for spatial autoregressive models. *Econometrica*, 72:1899–1925, 2004.

L.F. Lee. GMM and 2SLS estimation of mixed regressive, spatial autoregressive models. *Journal of Econometrics*, 137(2):489–514, 2007.

L.F. Lee and X. Liu. Efficient GMM estimation of high order spatial autoregressive models with autoregressive disturbances. *Econometric Theory*, 2009. URL http://journals.cambridge.org/action/displayAbstract?aid=6051680.

E.L. Lehmann. The Fisher, Neyman-Pearson theories of testing hypotheses: One theory or two? *Journal of the American Statistical Association*, 88(424):1242–1249, 1993.

James Levinsohn Leibbrandt, Murray and Justin McCrary. Incomes in South Africa Since the Fall of Apartheid. *NBER Working Paper*, No. 11384, 2005. URL http://www.nber.org/papers/w11384.

Thomas Lemieux. Decomposing changes in wage distributions: a unified approach. *Canadian Journal of Economics*, 35:646–688, 2002.

Daniel Léonard and Ngo Van Long. *Optimal control theory and static optimization in economics*. Cambridge Univ Press, 1992.

R. J. A. Little and D. B. Rubin. *Statistical analysis with missing data*. Hoboken, NJ: Wiley, second edition, 2002.

R. Lucchetti. Inconsistency of naive GMM estimation for QR models with endogenous regressors. *Economics Letters*, 75(2):179–185, 2002.

J. Ludwig and D.L. Miller. Does Head Start Improve Children's Life Chances? Evidence from a Regression Discontinuity Design. *The Quarterly Journal of Economics*, 122(1):159–208, 2007.

Jared K. Lunceford and Marie Davidian. Stratification and Weighting via the Propensity Score in Estimation of Causal Treatment Effects: A Comparative Study. *Statistics in Medicine*, 15:2937–2960, 2004.

J.A.F. Machado and J. Mata. Counterfactual decomposition of changes in wage distributions using quantile regression. *Journal of Applied Econometrics*, 20 (4):445–465, 2005.

Charles F. Manski. *Identification Problems in the Social Sciences*. Harvard University Press, Cambridge, MA, 1995.

J. McCrary. Manipulation of the running variable in the regression discontinuity design: A density test. *Journal of Econometrics*, 142(2):698–714, 2008.

J.F. McDonald and R.A. Moffitt. The Uses of Tobit Analysis. *Review of Economics and Statistics*, 62(2):318–321, 1980.

Allen McDowell. From the help desk: hurdle models. *The Stata Journal*, 3(2): 178–184, 2003.

B.D. Meyer. Unemployment insurance and unemployment spells. *Econometrica*, 58(4):757–782, 1990.

A. Mikusheva and B. Poi. Tests and confidence sets with correct size in the simultaneous equations model with potentially weak instruments. *The Stata Journal*, 6(3):335–347, 2006.

M. Mitchell. *A Visual Guide to Stata Graphics, 2nd Edition*. Stata Press, 2nd edition, 2008.

CG Moertel, TR Fleming, ET Creagan, J. Rubin, MJ O'Connell, and MM Ames. High-dose vitamin C versus placebo in the treatment of patients with advanced cancer who have had no prior chemotherapy. A randomized double-blind comparison. *New England Journal of Medicine*, 312(3):137–141, 1985.

S.L. Morgan and D.J. Harding. Matching estimators of causal effects: Prospects and pitfalls in theory and practice. *Sociological Methods & Research*, 35(1): 3–60, 2006.

M.P. Murray. Avoiding invalid instruments and coping with weak instruments. *Journal of Economic Perspectives*, 20(4):111–132, 2006.

I. Murtazashvili and J.M. Wooldridge. Fixed Effects Instrumental Variables Estimation in Correlated Random Coefficient Panel Data Models. *Journal of Econometrics*, 142(1):539–552, 2008.

T. Nannicini. Simulation-based sensitivity analysis for matching estimators. *Stata Journal*, 7(3):334–350, 2007.

C.R. Nelson and R. Startz. Some further results on the exact small sample properties of the instrumental variable estimator. *Econometrica*, 58(4):967–976, 1990.

W.K. Newey, J.L. Powell, and F. Vella. Nonparametric estimation of triangular simultaneous equations models. *Econometrica*, 67(3):565–603, 1999.

J. Neyman. On the application of probability theory to agricultural experiments. Essay on principles. Section 9. *Roczniki Nauk Rolniczych*, X:1–51, 1923.

Translated with an introduction by D. M. Dabrowska and T. P. Speed. *Statistical Science,* 5(4):465–472, 1990.

J. Neyman, K. Iwaszkiewicz, and St. Kołodziejczyk. Statistical problems in agricultural experimentation. *Supplement to the Journal of the Royal Statistical Society*, 2(2):107–180, 1935.

Austin Nichols. Weak Instruments: An Overview and New Techniques. *Presentation at NASUG Meetings*, 2006. URL http://www.stata.com/meeting/5nasug/wiv.pdf.

Austin Nichols and Kelly Rader. Spending in the Districts of Marginal Incumbent Victors in the House of Representatives. *Unpublished Working Paper*, 2007.

Austin Nichols and Mark E. Schaffer. Clustered Errors in Stata. *Presentation at 2007 UK Users Group meeting*, 2007. URL http://www.stata.com/meeting/13uk/nichols_crse.pdf.

Austin Nichols and Elaine Sorensen. Effects of the NY state NCP EITC on child support payments. *Working Paper*, 2009.

E.C. Norton, H. Wang, and C. Ai. Computing interaction effects and standard errors in logit and probit models. *Stata Journal*, 4(2):154–167, 2004. URL http://www.stata-journal.com/sjpdf.html?articlenum=st0063.

Ronald Oaxaca. Male-Female Wage Differentials in Urban Labor Markets. *International Economic Review*, 14(3):693–709, 1973.

L.E. Papke and J.M. Wooldridge. Econometric methods for fractional response variables with an application to 401 (k) plan participation rates. *Journal of Applied Econometrics*, 11:619–632, 1996.

Leslie E. Papke and J.M. Wooldridge. Panel Data Methods for Fractional Response Variables with an Application to Test Pass Rates. *Journal of Econometrics*, 145(1-2):121–133, 2008.

J. Pearl. *Causality: models, reasoning, and inference*. Cambridge University Press, 2000.

Karl Pearson. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *Phil. Mag.*, 5:157–175, 1900.

R. L. Plackett. Some theorems in least squares. *Biometrika*, 37(1-2):149–57, 1950.

R. L. Plackett. Karl Pearson and the chi-squared test. *International Statistical Review*, 51:59–72, 1983.

JB Ramsey. Tests for specification errors in classical linear least-squares regression analysis. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 350–371, 1969.

Calyampudi Radhakrishna Rao. Information and the accuracy attainable in the estimation of statistical parameters. *Bulletin of the Calcutta Mathematical Society*, 37:81–89, 1945.

C.R. Rao. *Linear statistical inference and its applications*. Wiley New York, 1973.

A. Rényi. On a new axiomatic theory of probability. *Acta Mathematica Hungarica*, 6(3):285–335, 1955.

G. Ridder. The non-parametric identification of generalized accelerated failure-time models. *The Review of Economic Studies*, pages 167–181, 1990.

G. Ridder and T.M. Woutersen. The singularity of the information matrix of the mixed proportional hazard model. *Econometrica*, pages 1579–1589, 2003.

W.H. Rogers. Regression standard errors in clustered samples. *Stata Technical Bulletin*, 13:19–23, 1993. URL http://www.stata.com/support/faqs/stat/stb13_rogers.pdf.

David Roodman. How to do xtabond2: An introduction to difference and system GMM in Stata. *The Stata Journal*, 9(1):86–136, 2009a.

David Roodman. Estimating fully observed recursive mixed-process models with **cmp**. *Center for Global Development Working Paper*, 168, 2009b.

David Roodman and Jonathan Morduch. The impact of microcredit on the poor in bangladesh: Revisiting the evidence. 2009. URL http://www.cgdev.org/content/publications/detail/1422302.

Paul R. Rosenbaum. *Observational Studies*. Springer, New York, 2002.

Paul R. Rosenbaum and Donald B. Rubin. The Central Role of the Propensity Score in Observational Studies for Causal Effects. *Biometrika*, 70(1):41–55, 1983.

Jesse Rothstein. Do Value-Added Models Add Value? Tracking, Fixed Effects, and Causal Inference. *Working Paper, Princeton University*, 2007a. URL http://www.princeton.edu/ceps/workingpapers/159rothstein.pdf.

Jesse Rothstein. Does competition among public schools benefit students and taxpayers? Comment. *American Economic Review*, 97(5):2026–2037, 2007b.

A.D. Roy. Some thoughts on the distribution of earnings. *Oxford Economic Papers*, 3(2):135–146, 1951.

P. Royston. Multiple imputation of missing values. *Stata Journal*, 4(3):227–241, 2004.

P. Royston, J.B. Carlin, and I.R. White. Multiple imputation of missing values: New features for mim. *Stata Journal*, 9(2):252–264, 2009.

Donald B. Rubin. Estimating Causal Effects of Treatments in Randomized and Nonrandomized Studies. *Journal of Educational Psychology*, 66(5):688–701, 1974.

Donald B. Rubin. Statistics and Causal Inference: Comment: Which Ifs Have Causal Answers. *Journal of the American Statistical Association*, 81(396):961–962, 1986.

Donald B. Rubin. Comment: Neyman (1923) and Causal Inference in Experiments and Observational Studies. *Statistical Science*, 5(4):472–480, 1990.

D. Ruppert, M.P. Wand, and R.J. Carroll. *Semiparametric regression*. Cambridge Univ Press, 2003.

William R. Shadish, Thomas D. Cook, and Donald T. Campbell. *Experimental and Quasi-Experimental Designs for Generalized Causal Inference*. Boston: Houghton Mifflin, 2002.

C.P. Simon and L. Blume. *Mathematics for economists*. Norton, 1994.

J.D. Singer and J.B. Willett. *Applied longitudinal data analysis: Modeling change and event occurrence*. Oxford University Press, USA, 2003.

M. Sinning, M. Hahn, and T.K. Bauer. The Blinder–Oaxaca decomposition for nonlinear regression models. *The Stata Journal*, 8(4):480–492, 2008.

J. Snow. *On the Mode of Communication of Cholera*. Churchill, London, 1855. Reprinted (1965) by Hafner, New York.

James H. Stock and Motohiro Yogo. Testing for weak instruments in linear IV regression. In Donald W. K. Andrews and James H. Stock, editors, *Identification and inference for econometric models: Essays in honor of Thomas Rothenberg*. Cambridge University Press, 2005. URL http://papers.nber.org/papers/t0284.

Mervyn Stone. Cross-Validation and Multinomial Prediction. *Biometrika*, 61(3): 509–515, 1974.

Mervyn Stone. Asymptotics For and Against Cross-Validation. *Biometrika*, 64 (1):29–35, 1977.

Elizabeth A. Stuart and Donald B. Rubin. Best practices in quasi-experimental designs: Matching methods for causal inference. In Jason Osborne, editor, *Best Practices in Quantitative Social Science*. Thousand Oaks, CA: Sage Publications, 2007.

J. Tobin. Estimation of relationships for limited dependent variables. *Econometrica*, 26(1):24–36, 1958.

A. Tsiatis. A nonidentifiability aspect of the problem of competing risks. *Proceedings of the National Academy of Sciences*, 72(1):20–22, 1975.

G.J. Van den Berg. Duration models: Specification, identification, and multiple durations. In J.J. Heckman and E. Leamer, editors, *Handbook of econometrics*, volume 5, pages 3381–3460. Amsterdam:North-Holland, 2001.

Wilbert van der Klaauw. Estimating the Effect of Financial Aid Offers on College Enrollment: A Regression Discontinuity Approach. *International Economic Review*, 43:1249–1287, 2002.

J.W. Vaupel, K.G. Manton, and E. Stallard. The impact of heterogeneity in individual frailty on the dynamics of mortality. *Demography*, 16:439–454, 1979.

H. White. *Asymptotic Theory for Econometricians*. Academic Press, 1984.

R.J. Willis. Wage determinants: A survey and reinterpretation of human capital earnings functions. In Orley Ashenfelter and Richard Layard, editors, *Handbook of labor economics*, volume 1, pages 525–602. Amsterdam:North-Holland, 1987.

F. Windmeijer. A finite sample correction for the variance of linear efficient two-step GMM estimators. *Journal of econometrics*, 126(1):25–51, 2005.

F. S. Woods. *Advanced Calculus*. Boston, MA: Ginn, 1926.

Jeffrey M. Wooldridge. *Econometric Analysis of Cross Section and Panel Data*. Cambridge, MA: MIT Press, 2002.

Jeffrey M. Wooldridge. *Unobserved Heterogeneity and Estimation of Average Partial Effects*, chapter 3. Cambridge University Press, 2005a.

J.M. Wooldridge. On two stage least squares estimation of the average treatment effect in a random coefficient model. *Economics Letters*, 56(2):129–133, 1997.

J.M. Wooldridge. Further results on instrumental variables estimation of average treatment effects in the correlated random coefficient model. *Economics Letters*, 79(2):185–191, 2003.

J.M. Wooldridge. Simple Solutions to the Initial Conditions Problem for Dynamic, Nonlinear Panel Data Models with Unobserved Heterogeneity. *Journal of Applied Econometrics*, 20(1):39–54, 2005b.

P.G. Wright. *The tariff on animal and vegetable oils*. Macmillan, 1928.

A. Yatchew and Z. Griliches. Specification error in probit models. *The Review of Economics and Statistics*, pages 134–139, 1985.

Myeong-Su Yun. Decomposing Differences in the First Moment. *Economics Letters*, 82(2):275–280, 2004. URL http://ftp.iza.org/dp877.pdf.

Myeong-Su Yun. A Simple Solution to the Identification Problem in Detailed Wage Decompositions. *Economic Inquiry*, 43(4):766–772, 2005a. URL http://ftp.iza.org/dp836.pdf.

Myeong-Su Yun. Normalized Equation and Decomposition Analysis: Computation and Inference. *IZA Discussion Paper*, 1822, 2005b. URL http://ftp.iza.org/dps/dp1822.pdf.