ECON 0150 | Part 0 | Day 2 | Framework

Opener

In 1992 New Jersey passed a law increasing the state minimum wage. Economists have long questioned whether these kinds of policies are good for the labor market. If you've studied any economic theory you have likely seen a part of this debate.

Competitive markets with many buyers and many sellers will reach an equilibrium that maximizes the total surplus in the market. Introducing a price control changes the market's equilibrium. If we introduce a binding price ceiling, a maximum price that doesn't allow the market equilibrium price, we restrict the market price to be lower than without the price control. With this price control the market moves the price as close to the uncontrolled equilibrium as possible but gets stuck *at* the price control. A binding price ceiling leaves the market choosing a price *at* that binding price ceiling.

The same is true of a binding price floor in a competitive market. A binding price floor is a minimum price set *above* the uncontrolled market equilibrium price. As you've likely seen, the same incentives are at play here as in a binding price ceiling. The market pushes the price as close to the uncontrolled market equilibrium but are prevented from reaching it by the price control, leaving the price *at* the price control. With a binding price control in a competitive market this leads to a higher price and a quantity demanded lower than quantity supplied, what we call a shortage.

In a labor market buyers are firms doing the hiring of labor and sellers are workers who bring labor to the firm. In a competitive labor market with many buyers of labor (many companies competing for their hires) and many sellers of labor (many workers competing for jobs) the equilibrium wage the market reaches can in theory be so low that it's difficult to afford to pay ones bills. This was the motivation for New Jersey to raise it's minimum wage in 1992 with the aim of impacting labor markets like the fast-food industry.

Right across the border at the same time Pennsylvania did not raise *their* state minimum wage. This meant the nearby local labor markets in New Jersey and Pennsylvania would experience different minimum wage policies while sharing similar economic conditions, similar weather, and similar workers crossing borders for jobs. Two economists at Princeton, David Card and Alan Krueger, recognized this as an opportunity. Rather than relying on economic theory alone to predict what would happen, they could collect data and measure what *actually* happened.

Data

In early 1992, Card and Krueger designed a telephone survey of fast-food restaurants in New Jersey and eastern Pennsylvania. They chose the fast-food industry for several practical reasons. Fast-food restaurants are major employers of low-wage workers, they comply with minimum wage laws, and the jobs and products are similar across stores, making comparison straightforward. They built a sample of 410 restaurants from the Burger King, KFC, Wendy's, and Roy Rogers chains.

The first wave of surveys went out in late February and early March 1992, just before New Jersey's minimum wage rose from $4.25 to $5.05 on April 1st. Card and Krueger asked managers about employment levels, starting wages, prices, and store characteristics. The second wave followed in November and December 1992, about eight months after the increase. Remarkably, they successfully re-interviewed nearly 100 percent of the original stores, including tracking down the six restaurants that had permanently closed.

Summarization

Before the minimum wage increase, restaurants in both states looked similar. The average starting wage was $4.61 in New Jersey and $4.63 in Pennsylvania. About 30 percent of New Jersey stores were paying exactly the $4.25 federal minimum. Average employment was 20.4 full-time-equivalent workers per store in New Jersey and 23.3 in Pennsylvania.

After the increase, the picture changed. In New Jersey, virtually all restaurants that had been paying below $5.05 raised their wages to meet the new minimum. By November 1992, 85 percent of New Jersey stores reported a starting wage of exactly $5.05. Pennsylvania stores, facing no new requirement, stayed put. About a quarter were still paying $4.25.

Analysis

Card and Krueger used what economists call a **difference-in-differences** approach. The logic is simple: compare how employment changed at New Jersey restaurants to how it changed at Pennsylvania restaurants over the same period. Any difference in the *changes* can be attributed to the minimum wage increase, since both groups of restaurants faced similar economic conditions otherwise.

They also ran a second comparison entirely within New Jersey. Some New Jersey stores had already been paying $5.00 or more before the law changed. These "high-wage" stores were largely unaffected by the new minimum. Comparing employment changes at low-wage stores (those forced to raise wages) to high-wage stores (those already above the threshold) provided another test of the minimum wage's effect.

Results

The results contradicted the standard prediction. Employment at Pennsylvania stores *fell* by an average of 2.16 full-time-equivalent workers per store while employment at New Jersey stores *rose* by 0.59 workers per store. This difference-in-differences of 2.76 more employees per store in New Jersey than Pennsylvania meant that the state with the minimum wage increase actually saw *relative employment gains* of about 13 percent.

The within-New Jersey comparison told the same story. Stores that had been paying $4.25 before the increase saw employment rise. Stores already paying $5.00 or more saw employment fall by almost exactly the same amount as Pennsylvania stores. If unobserved factors were driving New Jersey's gains, they should have affected high-wage and low-wage stores alike. They didn't.

Card and Krueger also found that fast-food prices rose about 4 percent faster in New Jersey than Pennsylvania, suggesting restaurants passed some of the higher labor costs on to customers rather than cutting workers.

Conclusions

The Card and Krueger study became one of the most cited and debated papers in labor economics. It didn't settle the minimum wage debate. Critics questioned the data, the methodology, and the generalizability of the results. But it demonstrated something important: careful data collection and analysis can challenge long-held theoretical predictions.

Economic theory told us one thing. The data told us something different. Understanding **why** requires more than theory alone. It requires learning how to gather data, how to measure what we care about, how to compare groups fairly, and how to interpret what we find. That is what this course is about.

This make it look easy. But there's a lot of detail. That's where we're going in this course.

Data Framework

So Card and Krueger showed us something important. Theory told us one thing. Data told us something different. That's exciting. But how do we actually do this kind of work ourselves? That's where we're going next.

Lets start with a basic question: what is data?

Data is a sample drawn from some process we want to understand. We write this as x drawn from F. The x is what we observe: our data. The F is what generated it: the underlying process, the random variable, the thing we actually care about. We have x. We care about F. That distinction matters. In the minimum wage example, what we care about is not the impact of a minimum wage increase for New Jersey specifically. But that's the data we observed. That's our x. We care about is the impact of minimum wage increases generally. That's F. And we'll essentially never see F. What we see is x, the specific case.

This gives us two core skills to develop. First, description. How do we summarize the data we have? That's Parts 1 and 2 of this course. Second, inference. How do we use the data we have to learn about the process that generated it? That's Parts 3 through 5.

Now lets talk about how we organize data. Notation helps keep things straight. We write x with subscripts i and t. The i indexes entities: people, firms, countries, whatever we're measuring. It's unordered. Person 1 isn't before Person 2 in any meaningful way. The t indexes time: days, months, years. It is ordered. January comes before February.

This distinction between i and t gives us what we call substantive variables versus index variables. The substantive variable is the thing we're measuring: income, employment, whatever. The index variables tell us how the data is organized: which entity, which time period.

There are three dimensions we use to describe data: structure, variable type, and number of variables.

Data structure depends on which indices are active. If only i varies, we call it cross-sectional data. Think of household incomes measured in 2024. There are many households, one point in time. If only t varies, we call it timeseries data. Think of average US income measured from 1950 to 2024. It's one unit over many time periods. If both i and t vary, we call it panel data. Think of income across many households tracked from 1950 to 2024. There are many units that we track over many time periods.

Variable type refers to what kind of values x can take. There are two main types: categorical and numerical. Categorical variables have distinct categories. Binary means two categories, like employed or unemployed. Nominal means unordered categories, like blood type. Ordinal means ordered categories, like education level. Numerical variables have numbers that mean something. Discrete means countable values, like number of children. Continuous means values that can take any real number, like income.

Number of variables is straightforward. Univariate means one variable. Bivariate means two. Multivariate means more than two.

When we work with data, there are three steps we go through every time. First, select. What does our data contain? This is about where the data comes from and what part of the data we choose. Second, transform. How do we change the data to make it more useful? Sometimes this means changing a variable mathematically in some way, like taking a log of income. Third, encode. How do we turn values into visual elements?

Select, transform, encode. That's the workflow for data summarization.

The course builds complexity along two axes. In Part 0, we're setting up the framework. What tools do we need? In Part 1, we focus on single variables. What does this variable look like? In Part 2, we look at relationships. How do these variables relate? In Part 3, we start inference with the univariate general linear model. What can we infer about the population? In Part 4, we add a predictor with the bivariate model. How does y change with x? In Part 5, we add controls with the multivariate model. How does y change with x, controlling for z?

Each class follows a consistent rhythm. Before class, you will often watch a concept video to learn the core ideas at your own pace. At the start of class, there's a quiz to confirm your understanding. During class, we work through an exercise together with support. After class, you do the homework for independent practice. The exercises are homework prep. If you can do the exercise, you can do the homework. The homework is miniexam prep.

Lets begin.

Exercise 0

Exercise 0 is about recognizing data. For each dataset, we want to identify four things: the index variables, the data structure, the variable type, and the number of variables. Then we visualize it.

Python Intro

If you've never written code before, this might feel intimidating. It's not. Code is just a way of giving instructions to a computer. Instead of clicking buttons, you type out what you want to happen. That's it.

We use something called a notebook. A notebook is a document with cells. Each cell is a little box. Some cells contain code. Some cells contain text. You run cells one at a time by clicking the play button or pressing Shift+Enter. When you run a code cell, the computer reads your instructions and does what you asked. The output appears right below the cell.

Here's the first thing that confuses people: cells run in the order you run them, not the order they appear on the page. If you skip a cell, the computer doesn't know about it. If you run cell 5 before cell 2, and cell 5 depends on something defined in cell 2, you'll get an error. The fix is simple: run cells from top to bottom, in order.

Now lets talk about variables. A variable is just a name that stores a value. When you write `x = 5`, you're telling the computer: remember the value 5 and call it x. The equals sign doesn't mean "equals" like in math. It means "assign." You're assigning the value 5 to the name x. Later, when you write `x`, the computer knows you mean 5.

You can store different types of things in variables. Numbers are just numbers: `x = 5` or `y = 3.14`. Text needs quotes around it: `name = 'Taylor'` or `greeting = "Hello"`. The quotes tell the computer this is text, not a variable name. Without quotes, the computer would think Taylor is the name of some other variable and get confused when it can't find it.

You can do things with variables. With numbers, you can do math: `x + y` gives you 8 if x is 5 and y is 3. With text, you can combine them: `greeting + ' ' + name` gives you 'Hello Taylor'. The + sign does different things depending on what you're working with. With numbers, it adds. With text, it concatenates (sticks them together).

Here's another thing that confuses people: if you change a variable, you need to re-run the cells that use it. Lets say you defined `x = 5` and then later wrote `x + 10` which gave you 15. If you go back and change the first cell to `x = 100` but don't re-run it, the computer still thinks x is 5. You changed the text on the page, but you didn't tell the computer about it. Always re-run cells after making changes.

Now lets talk about functions. A function is something that does work for you. It has a name followed by parentheses. You put inputs inside the parentheses, and it gives you back an output. For example, `len('hello')` tells you how long the text is. The function is called `len`, the input is `'hello'`, and the output is 5. You don't need to know how it works inside. You just need to know what to put in and what comes out.

Some functions belong to objects. You'll see this written with a dot. When you write `df.head()`, you're saying: take the thing called df and run the head function on it. The dot connects the object to its function. Think of it like "df's head" or "the head of df." This is how we work with data in Python.

Speaking of data, we use a package called pandas. Python on its own doesn't know how to work with spreadsheets. Pandas gives it that ability. At the top of the notebook, you'll see `import pandas as pd`. This loads the pandas package and gives it the nickname pd, just like my friends in college called me TayTay. From then on, we can use pd to access pandas functions. And just like when I heard someone call TayTay! from across the quad, python will know what what to do.

The main thing pandas gives us is called a DataFrame. A DataFrame is just like a spreadsheet. It has rows and columns. Each column has a name. To load data from a file, we write `pd.read_csv('filename.csv')`. This reads the file and creates a DataFrame. We usually store it in a variable: `df = pd.read_csv('filename.csv')`. Now df contains our data.

Once you have a DataFrame, you can explore it. `df.head()` shows you the first few rows. `df.shape` tells you how many rows and columns. `df.columns` lists the column names. To look at just one column, you write `df['column_name']` with the column name in quotes inside square brackets. To see what unique values are in a column, you write `df['column_name'].unique()`. To count how many of each value, you write `df['column_name'].value_counts()`.

Errors will happen. That's normal. When you see red text, read it. The last line usually tells you what went wrong. Common errors: NameError means you're using a variable that doesn't exist yet, probably because you forgot to run an earlier cell. KeyError means you're trying to access a column that doesn't exist, probably a typo in the column name. SyntaxError means you wrote something the computer can't understand, probably a missing quote or parenthesis.

The best way to learn is to experiment. Change things and see what happens. Break things on purpose. The notebook won't bite. If something stops working, you can always re-run all cells from the top and start fresh.

Question 1 looks at dataset1.csv. This contains US Real GDP by year. We see a Year column and a Real_GDP column. Year is our index — it's ordered, so this is the t index. That makes this time series data. Real GDP is measured in trillions of dollars. It can take any value, so it's continuous. And we're looking at one substantive variable, so it's univariate. A line chart works well for time series data because it shows how values change over time.

Question 2 looks at dataset2.csv. This contains employment status for different households. Household ID is our index — it's unordered entities, so this is the i index. That makes this cross-sectional data. Employment Status has two possible values: Employed or Unemployed. Two categories makes it binary. And again, we're looking at one substantive variable, so it's univariate. A bar chart or countplot works well for binary data because it shows how many observations fall into each category.

Question 3 looks at dataset3.csv. This is a bit more complex. We have Household ID and Year as columns, plus Income and Savings. Both i and t are active here — we're tracking multiple households across multiple years. That makes this panel data. Income and Savings are both measured in dollars and can take any value, so they're continuous. And we have two substantive variables, so it's bivariate. A boxplot by year can show how the distribution of income changes over time.

Question 4 looks at dataset4.csv. This contains economic optimism ratings from different people. ID is our index — unordered entities, so cross-sectional. Economic Optimism has categories like Very Optimistic, Somewhat Optimistic, Neutral, Somewhat Pessimistic, Very Pessimistic. These are categories with a meaningful order, so it's ordinal. One substantive variable, so univariate. A bar chart with ordered categories works well here because the order matters for interpretation.

Question 5 looks at dataset5.csv. This contains employment sector for different people. ID is our index — again cross-sectional. Sector has categories like Services, Agriculture, Manufacturing, Unemployed. These are categories but there's no inherent order to them. Services isn't more or less than Agriculture in any meaningful way. So it's nominal. One substantive variable, so univariate. A bar chart works well, and we might sort by count to make comparisons easier.

The point of this exercise is to build the habit of asking these questions whenever you encounter a new dataset. What's the structure? What type of variable am I looking at? How many variables? The answers determine which tools are appropriate for summarizing and visualizing the data.

That's where we're going next time!