**The Task:**

For my Economics undergraduate thesis, I built upon a past project from my course in Econometrics that required learning Stata, a statistics software, and refreshing our memory of basic statistics. Stata is a data analysis and visualization tool widely used in academia and an accessible introduction to software programming for undergraduate students. My thesis examined the effect the Coronavirus Aid, Relief, and Economic Security (CARES) Act, part of a broader Unemployment Insurance program in response to economic slowdown caused by the pandemic, had on labor market characteristics, including Unemployment, Labor Force participation rates, and Employment rates. This extended benefits (EB) program, enacted March of 2020, extended the period during which the federal government supplied US citizens with Unemployment Insurance (UI) in the form of direct stimulus payments.

I applied a difference-in-difference (DiD) regression model to compare states opting out versus states retaining UI until expiration in September 2021 in terms of unemployment, employment, and labor force participation rates. I control for pandemic impacts using Covid-19 specific deaths and number of cases. The bifurcation of states receiving UI facilitates a more accurate demonstration of the effect of EB on these labor market characteristics.

**Data and Data Cleaning:**

To appropriately construct a DiD model, I use the date of late June 2021 in which 26 states opted-out of the federal UI program 2 months prior to the EB’s expiration in early September. The remaining 34 states retained their EB until the expiration. My data consists of monthly state employment status of the noninstitutional population provided by the BLS Local Area Unemployment Statistics (LAUS) and CDC vaccination rate data. I employ this DiD design to take advantage of this divergence and compare states by grouping them into opt-out states and retain states then compare them in terms of unemployment, employment, and labor force participation rates. I control for pandemic impacts using Covid-19 specific deaths and cases.

After the data has been identified, the next step is to prepare, or clean, the data to ensure a clearer analysis, as demonstrated by the two figures below. The first being the raw data and the second is the cleaned form required to analyze the data in any software program. On the following page, I show what the data looks like directly from the source and how it looks after much work in Excel.

Cleaning data can take a great deal of time especially when pulling data from multiple sources. Cleaning data will reveal discrepancies and requires attention to detail. For example, dates can be in different formats, data in a column may be set as “General” and non-numeric when it should be a date or a number. In Excel, “General” will read as a string variable, rather than as a date.

Dirty Labor Force Data

Table

Description automatically generated

Dirty CDC Data

Graphical user interface

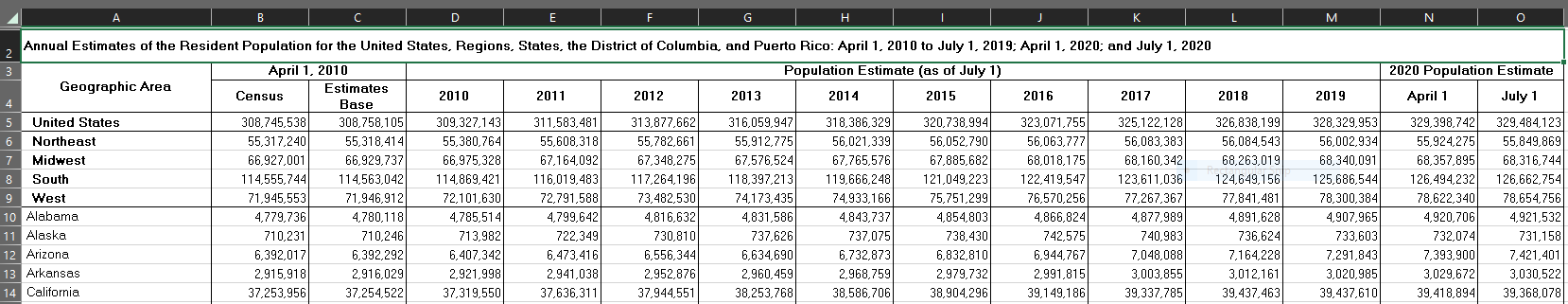
Description automatically generated

Dirty CDC Cont.

A picture containing graphical user interface

Description automatically generated

Dirty State Pop. Data



Data Cleaned

Graphical user interface, application

Description automatically generated

**Stata Software Coding:**

With the data cleaned and organized, I am able to import the data in Stata and run my analysis.

|  |  |
| --- | --- |
| Text  Description automatically generated |  |

The red arrows indicate how I create a path to where the cleaned data is on my system for Stata to refer to and access. This brings Stata to the folder in which the file is located, an import command is necessary to select the specific file to run on. I can also do this manually (but only saved in memory) by using the main interface to import the file in. With the global path set and import command following, and having it saved in a do.file, allows me to just simply run lines 7 through 26 all at once. The main interface, or “global environment” cannot run multiple codes and does not save anything for future reference.

The yellow arrow indicates where I have renamed some variables that would be tedious and inefficient to write out multiple times. Mistakes happen frequently and require a keen eye to pay attention to the finer details. I rename these variables that will later be used in other forms of analysis. In Economics, variables can have long and ambiguous titles, especially when they include acronyms and special characters. Though this will take a few more seconds to write out, it adds up at the end of the project and prevents future typos one can spend several minutes finding and consulting a colleague for assistance.

The blue arrow presents a very important part to the difference-in-difference model. I will be creating two variables that will be the concrete footing to my regression. This will be a time and state variables that are necessary to have in Fixed Effects regressions. FE are variables that are constant across individuals; these variables, like age, sex, and in this case a state’s laws and regulations, do not change or change at a constant rate over time. The downside of FE is that is it cannot control for variables that vary overtime.

The “egen” command consists of functions that extend the capability of the normal ‘gen’ command. The various functions within ‘egen’ create variables that hold information about patterns and calculations within subgroups or across columns. ‘egen’ creates a new variable from the data I have already imported. Because I have panel data, I have multiple entities with multiple time periods. I must group each state so that California in Month 1 of 2019 is the same entity as California in Month 2 of 2019. I do want to analyze how California will change overtime however I want to keep CA as a single entity. There cannot be more than one CA, and Stata will want to follow this train of thought. To correct Stata, I create a new variable within the data and group it by the existing states in the data. That will recognize all my states for each month as their own unique entities.

I am generating a variable that is based on a function that is written in latter part of the egen statement. In this case, I generate a state\_id variable (to have 50 states) in which the function is going to be grouped by state. Egen state\_id = group(state) is basically saying I want to create a new variable based on the function of grouping together our states. I want this because I have 50 states and monthly data. With this, there will be 12 California’s, 12 Arizona’s, etc. for each year (monthly data) that will all be treated to show that Arizona Month 1 is different from Arizona Month 2 but still knowing that Arizona is still the same state. Like measuring a child’s height overtime, Child in 2019 will be the same entity in 2020 but will have a different height we want to capture and analyze their growth overtime. I want the states to be the same BUT I will be looking at the progress of change overtime for each state for 3 years (36 months).

Before I create the next variable, I use ‘destring’ which is exactly what is sounds like. Because Stata did not recognize the months in the excel file as numbers, and rather as a string of characters, I lacked a time variable and needed to create a new one. I destring ‘Month’ and create a new variable that simply makes month a numeric time variable.

I use time and state FE because I cannot measure all the variables I need to control for in my model. When I have fixed effects for each state being unique in their approach to the Covid-19 pandemic, I want to explain things that I cannot measure that are being picked up in the state and time FE that will explain why our variable of interest may be small or statistically insignificant. I want to control for the differences in states over a range of time, but I cannot simply include the many possible variables available to narrow down what the main driver is to disparities in my chosen labor market characteristics. Because my focus is on just the effect of withdrawing from the federal EB 2 months prior to expiration, I am examining the discrepancy appear between withdraw and retain states while including variables like covid-19 effects to see what happens. In addition, I need to control for state and time FE because their will be omitted variable bias that can be picked up through this control that will be visible in the coefficients through erratic and overly large/small numbers. States and time change, but I make the assumption here that I can hold them constant in the short run (less than 3 years) to analyze the disparity between retain and withdraw states. Fixed effects focuses on the individual, meaning I discard any variation between individuals. The individuals in my analysis are states. What I have now is just the variation not between each state, but variation within the states. That variation I want to control for is changes in states over time.

The green arrow is my final line of code for the main regressison. The ‘i.’ in front of state and time Stata handles factor (categorical) variables elegantly. I can prefix my state and time variables with “i.” to specify indicators for each level (category) of the variable. This creates a category for each state and each time (in this case months over the course of a few years) so I can analyze the effect of our independent variables on our dependent variable.

**Running the Regression:**

A screenshot of a computer

Description automatically generated with medium confidence

Running the regression will present in the global view of Stata a Regression Output. The information on the top right is the Model fit information that tells how well our my linear model fits the data from a graphical standpoint. This is the same thing as the Line of Best Fit that draws a straight line through a cloud of data on an XY chart. R-squared is the coefficient that represents how well this line fits the data. It is simply the fraction of the variation in the dependent variable that can be explained by the independent variable. For my output, 83.99% of the variation in the dependent variable, in this case the unemployment rate, is explained by the model.

The main table is referred to as the Parameter Estimation that shows the parameters estimated by the model and their respective statistical significance. Stata also automatically provides a hypothesis test using the t-test to find how each estimated coefficient is significantly different from zero. In statistics, the null assumes that there is no relationship with the dependent variable for each independent variable. I want to prove otherwise and say there is a relationship with an alternate hypothesis that these coefficients are significantly different from zero. The first thing to look at is the 95% CI which is more easily interpreted by the P-value associated with the t-stats. P>|t| is the probability of getting the coefficient or even more (or less if coefficient is negative) that is significantly different from zero. For a threshold of 0.05, which is the critical value for 95%, (0.01 for 99%) I have enough evidence to accept the alternate hypothesis that the estimated coefficients are not equal to 0 because their p-values are all lesser than 0.05. However, I cannot reject the null hypothesis for any coefficient with a p-value is greater than the 0.05 significant threshold. In Laymen’s terms, a 95 percent confidence interval, you have a 5 percent chance of being wrong. With a 90 percent confidence interval, you have a 10 percent chance of being wrong. Being “wrong” indicates the uncertainty surrounding an estimate and I cannot confidently claim that a result from data generated by testing is to be attributable to a specific cause rather than chance. Pulling from a population, I want to ensure that the results I have found are not by luck.

**Visualizing Data:**

An effective data analyst will not only be able to write code and decipher statistical findings, but also be able to present to non-analysts what they have found. Visualizing data is one of the most important and desired skills of any analyst regardless of field. Here I can communicate easily what I have found and present to colleagues, teams, and managers that will save time and money.

Text

Description automatically generated

This code may seem long and complex, however it is quite the opposite. The entirety of this block of code is what will provide some information from the Parameter Estimation and Model fit tables and combine it with academic level presentation of having multiple models that show how coefficients for the dependent variable change with the gradual inclusion of independent variables and other information one at a time.

The red line shows the code that establishes the word document and details I want specified on how the info is presented. It also runs my first model because Stata requires at least one model to be inputted to create the Word output. “Ourteg” establishes the Word document and tells Stata where the results shall go. The yellow lines show the gradual additions of one independent variable at a time. All that was needed was to write the regression and then below it provide the outreg command that tells Stata where the info goes then add “append” to add it to the already established document. The last line is simply creating a new document with the command “replace” that establishes a new document. I wanted to see how lagging my opt out variable by two time periods (1 hashtag = 1 time period, months in this case) would compare to my normal regression. I do this for each dependent variable; unemployment rate, employment rate, and labor force participation rate.

**Outreg Example for Employment Rate as Dependent Variable**

Graphical user interface, text

Description automatically generated

**Stata Graphing: Creating Meaningful Graphics to Illustrate Findings**

Creating the Binary

When presenting findings, it is of upmost importance to clearly communicate methods, findings, and conclusions. I want to show how the three labor market characteristics I study have changed over time using a line graph. To do this, I need to first create two separate groups to which I will assign all 50 states to. The best way to do this is to create a binary variable where 1 is true and 0 is false. Just creating one binary variable will be the most efficient way.

Text

Description automatically generated

I create a new variable “optOutState” and assign it as a binary by setting it equal to 0. The remainder of the code is assigning which of the 50 states that opted out of the EB program early. By setting these 26 states to opt out with the new binary variable, I have now formally distinguished opt-out states from retain states to now examine how they differed overtime.

Graphing with the Binary: Trial Run

Below are 3 separate graphs that graph one of the three labor market characteristics indicated by red arrows. I create a line graph with just two lines where each line represents either opt-out or retain states specified by the binary variable I previously established.

Text

Description automatically generated

The two yellow arrows help me distinguish between the lines because the only distinction the lines have on the graph are colors. Because the lines are very similar, I keep this code to ensure which lines are which when referring to them in my main paper. The two yellow arrows will yield something like this below. They will be different from the main graphs I present because I graph the raw unemployment rate for each group of states here. On single line graphs, I can use raw numbers because only one entity (either retain or opt-out) exists on each graph. Combining two entities with raw numbers into one graph will show a corrupt graph.

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

There is one error I made that I did not correct after making the distinction between the two lines which was not renaming the Y axis that can be seen in the graphs and the code above. This is however corrected for when creating the other 2 main graphs that include both lines.

With this distinction made, I now turn to making the main 3 codes (red arrows). Creating a two-way line graphs requires that for each labor market characteristic I want to graph, I need to use the averages for each on a monthly basis rather than the raw rates. To convert the raw numbers I have for each entity, I use “collapse” then specify what type of data I want it converted to. Because an average is taking a certain number of observations and divided by the number of observations, I need to specify thresholds for when to make new means. I use “by” to specify I want monthly averages.

Graphing with the Binary: Main 3 Graphs

**Average Unemployment Rate for Retain and Opt-out states**

Chart, line chart

Description automatically generated

With some modifications in the graphing window after running the commands, I now have a meaningful and simple interpretation of the situation as a whole. The solid black line indicates the month in which FPUC expires for all states, the dashed line represents the beginning of the pandemic, and the green line at Month 30 is June 2021, which marks the month opt-out states ended their FPUC. Each labor market characteristic will have its own graph.

**Average Employment Rate of Retain and Opt-out States**

Chart, line chart

Description automatically generated

**Average Labor Force Participation Rate**

Chart, line chart

Description automatically generated