Project 2: Equity in Athletics Data Analysis

McCourt School of Public Policy, Georgetown University

Overview

This project will focus on categorical variables. In quant class, you've likely discussed categorical variables such as race and highest level of education completed.

We will be investigating data collected from college athletic programs. Look for Download Data files on the page linked below. http://ope.ed.gov/athletics (http://ope.ed.gov/athletics) Download the "Data for academic year 2013-2014"

Alternatively, the direct link is: http://ope.ed.gov/athletics/dataFiles/EADA%202013-2014.zip (http://ope.ed.gov/athletics/dataFiles/EADA%202013-2014.zip)

Background information on this data can be found here: http://www2.ed.gov/finaid/prof/resources/athletics/eada.html (http://www2.ed.gov/finaid/prof/resources/athletics/eada.html)

Week 1:

Key Ideas:

- Categorical variables: analysis, graphing, and data management
- Import excel data into Stata

Key Commands / Concepts:

- import excel
- bar graph
- factor variable notation (help fvvarlist)
- encode/decode
- recode

Questions

2.1 Import Excel Data

- Download the data set and documentation from the website given above.
- Extract all files into a new folder for this project.

- The files we are interested in are Schools.xlsx and SchoolsDoc2014.doc.
- Open Stata and set up your new do-file, including a command to change the working directory to the location where you extracted the data files.
- Open the data file Schools.xlsx in Stata using either the dialogue window or the `import excel` command.
- If you used the dialogue window, be sure to include the resulting `import excel` command in your do-file.
- Verify that the variable names were imported correctly.
- Verify that your data set has 17,134 observations and 128 variables.
- What does one observation in this data set represent? (Hint: `browse institution_name Sports`)
- The id variables in this data set are `institution_name` and `Sports`.
- Use the `order` command to move these two variables to the first two columns of the data set.
- For more information on the `order` command, type `help order`.

2.2 Regression on Dummy Variables

- Suppose we want to consider the relationship between total expenditure on sports programs on the size of the academic institution.
- We can judge size by the number of enrolled students, `EFTotalCount`.
- We don't think the relationship with size is completely linear, so we want to control for size categories instead.
- Size categories: Small = 0 to 999 students, Medium = 1000 to 4999 students, and Large = 5000 or more students.
- One way to accomplish this would be to create dummy variables
- Create two new dummy variables for medium and large schools.
- Verify each new variable with a two-way tabulation, reporting any missing values that exist.
- Run a regression of total expenditure, `TOTAL_EXPENSE_ALL` on the school-size dummies, leaving small schools as the baseline category.

2.3 Bar Graph

- Suppose we want to look at this data graphically.
- You can use a bar graph to visualize the relationship shown in the regression above.
- A bar graph will show the average value of variable(s) for each category of school size.
- To create a bar graph, we need a single categorical variable with three values depending on school size.
- Create a new categorical variable called `schoolsize` based on school size using the categories from the previous question.
- The new variable should have values of 1 for small schools, 2 for medium schools, and 3 for large schools.

- Confirm the creation of your new variable with a two-way tabulation.
- Create a bar graph showing the average total expenditure and average total revenue for each school category.

2.4 Factor Variables

- The previous two questions have shown two ways of representing categorical data.
- The same information can be represented as a series of dummy variables or as a single categorical variable.
- When working in Stata, it is generally better to use a single categorical variable, rather than constructing dummies.
- You need dummies for regression, but Stata has tools for creating these dummy variables automatically.
- Run the following regression and compare the results to the previous regression:

```
regress TOTAL_EXPENSE_ALL i.schoolsize
```

- This is called factor variable notation. Details can be found on the help page: `help fvvarlist`.

2.5 Details of factor variables

- When you use factor variable notation, Stata creates a hidden dummy variable for each category of the original variable.
- You can see the names of the hidden dummy variables by replaying the previous regression results with the option `coeflegend`.
- Look up the `coeflegend` on the help page for regress to see a description of the option.
- Try it out by typing: `regress , coeflegend`.
- The name of each hidden dummy variable appears between the brackets, e.g. _b[2.schoolsize]
- You can use these dummy variable names directly in some commands.
- Try it out: `list EFTotalCount schoolsize 1.schoolsize 2.schoolsize 3.schoolsize in 1/50`

2.6 Postestimation testing with factor variables

- Suppose you want to test the hypothesis that the coefficients on medium and large schools are jointly equal to zero.
- Unfortunately, the test command does not recognize factor variable notation.
- Rerun (or replay) the previous regression and try: `test i.schoolsize`
- Instead, you have to use the names of the hidden dummy variables that we saw in Question 2.4.
- Test the hypothesis that the coefficients on 2.schoolsize and 3.schoolsize are jointly equal to zero.
- You can also test this hypothesis using `testparm`, an alternate version of the `test` command.

- The `testparm` version of the command accepts factor variables, so you can just use `i.schoolsize` directly.
- Retry the hypothesis test of `i.schoolsize` using `testparm` instead of `test`.
- Notice the names of the hidden dummy variables appear in the output of the testparm command, where the two hypotheses being tested are listed.
- The command name `testparm` is not easy to remember, but it is listed on the help page for `test`.
- So if you ever need to remember the `testparm` command, just look at: `help test`.

2.7 Encode String Variables

- Examine the existing categorical variable, `sector_name`.
- Suppose we want to run a regression of total revenue on the dummy variables for each sector.
- Try regressing `TOTAL_REVENUE_ALL` on the categories of `sector_name` using factor variable notation (i.sector_name).
- You should receive an error, because `sector_name` is a string variable.
- The i. notation can only be used with numeric variables.
- This is a very common problem when transfering data from Excel to Stata.
- You could create a new numeric variable manually, like this:

```
gen sectorid = .
replace sectorid = 1 if sector_name=="
replace sectorid = 1 if sector_name=="
etc...
```

- But, there is a much easier way, using the command `encode`.
- Review the help page for this command: `help encode`.
- Use the encode command to create a new, labeled numeric variable called `sectorid`.
- Regress `TOTAL_REVENUE_ALL` on `sectorid` using factor variable notation.

2.7 Test and testparm

- Use the `regress , coeflegend` command to replay the regression results showing the factor variable names.
- Test the hypothesis that the coefficient on private nonprofit 2-year schools is equal to the coefficient on public two-year schools.
- Test the hypothesis that the coefficients on all 5 dummy variables are jointly equal to zero.

2.8 Bar graph

- Create a bar graph of average total revenue and average total expenditure over the categories of sectorid.
- When you have many categories, a horizontal bar graph is often more clear.
- Change your bar graph to a horizontal layout.

2.9 Labels/Recode

- Notice the value labels of `sectorid` are automatically included in regression output and bar graphs.
- You can add value labels manually, using the commands `label define` and `label values`.
- Alternatively, the `recode` command provides a way to recode variables and label them in one step.
- Here is an example using the recode command to generate the school size categorical variable.

```
recode EFTotalCount (0/999 = 1 "Small") (1000/4999 = 2 "Medium") (5000/max =
  3 "Large") , gen(schoolsize)
```

- Create a new categorical variable based on the number of participants in each type of sport including male and female teams.
- First, add the number of participants in the two variables, `PARTIC_MEN` and `PARTIC_WOMEN`.
- Hint: Consider an egen function to deal with missing values.
- Choose the cutoff values for the new variable to divide the data into quartiles.
- Make sure your new variable has value labels.
- Verify the creation of your variable with a two-way tabulation.
- Create a bar graph and run a regression of your choosing with your new categorical variable.

2.10 Reporting Results

- Go back through your do-file and add outreg2 commands to create a table (or multiple tables) reporting your regression results.
- Use the `label` option to use value labels for your factor variables.