Problem Set 4

Due by 4/11, 11:59PM

A researcher is interested in the relationship between a state's mortality rate and its spending on hospitals (and other health services). The zip file `state_data.zip` contains datasets on state mortality rate, state spending, state education level and state per capita income for the years 1993 through 2015. The data come from different sources such as the US Census, the Bureau of Economic Analysis and the US Mortality Database. More specifically, the data files are as follows:

a. `mortality_data.csv`,

b. `income_data.csv`,

c. education data for the years 1993 through 2006 are in the folder *education*, one file per year: `education_1993.csv`, ..., `education_2006.csv`,

d. education data for years 2007 through 2015: `education_0715.csv`,

e. expenditure data for the years 1993 through 2015 are in the folder *expenditure*, one file per year: `expnd_1993.csv`, ..., `expnd_2015.csv`.

Generate a Python script for the following questions.

(0) Import the *mortality* dataset and name it `mort_data`. Keep only the observations for the years 1993 through 2015.

(1) Change the column names for columns 4 through 11, to, `[mort_rate, prob_death, ave_length_surv, num_of_surv, num_of_deaths, num_years_lived, num_years_left, life_expec]`.

(2) `Age` column is a character type and needs to be changed to a numeric type. As such, first generate a new column, say `Age2`, by locating the `"-"` in the character string, and then slicing the string from the first character to the chracter just before `"-"`. Then, convert this to a numeric type.

(3) Generate a new column, say `age_group` by cutting `Age2` to three intervals: $[0, 18)$, $[18, 64)$, $[65,)$. Use `pd.cut()` function and assign the labels `"<18", "18-64", ">64"`.

(4) Drop `Age` and `Age2` columns, and reorder the columns as `[state, year, age_group, 7 mortality variables]`.

(5) Aggregate the mortality related variables (there are 7 of them) by summing over groups of `[state, year, age_group]`. Make sure that missing observations are not included in the summation.

(6) Import the *income* dataset and name it `inc_data`. Note that the data set is in the *wide* form and needs to be converted to the *long* form. To this end, you can use `pd.wide_to_long()` and set argument `stubnames` to the column names corresponding to multiple years in the wide form, and set argument `sep = "."`.

(7) Drop the last column, and sort the income data by as `[state, year]`.

(8) Append the education datasets from 1993 through 2006 and `education_0715`, name it `educ_data`. Rename columns 3 and 4 as `[phs, pcoll]`.

(9) Append the expenditure datasets from 1993 to 2015, and name it `expnd_data`. Note that the columns may have been named slightly different for some years.

(10) Merge `inc_data` and `educ_data` by `state` and `year`, and name the merged dataset `data`. Notice that this is a *one-to-one* merge.

(11) Merge `data` and `expnd_data` by `state` and `year`, and name the merged dataset again `data`. Notice again that this is a *one-to-one* merge.

(12) Merge `mort_data` and `data` by `state` and `year`, and name the merged dataset again `data`. Notice that this is a *many-to-one* merge.

(13) Remove `mort_data, inc_data, educ_data, expnd_data`.

(14) Change the measurement of `pinc, tot_revenue, taxes, tot_expnd, education, public_welfare, hospital, health` to in $1e4$ dollars, i.e., divide each by $1e4$.

(15) Change the measurement of `phs, pcoll` to ratios, i.e., divide each by 100.

(16) Generate a table of descriptive statistics for your dataset using `pd.describe()`.

(17) Regress `mort_rate` on an `intercept`, `health`, `hospital`, `log(pinc)`, `phs` and `pcoll` for the age group `">64"`. Name the results `spec1`.

(18) Regress `mort_rate` on an `intercept`, `health`, `hospital`, `log(pinc)`, `phs`, `pcoll` and state dummies for the age group `">64"`. Name the results `spec2`.

(19) Regress `mort_rate` on an `intercept`, `health`, `hospital`, `log(pinc)`, `phs`, `pcoll` and state and year dummies for the age group `">64"`. Name the results `spec3`.

(20) Generate a table for the regression results using `summary_col` or `stargazer`.