DATA-51100: Statistical Programming
Programming Assignment 6 – Visualizing ACS PUMS Data

Introduction

For this assignment, you will work with a survey dataset and use the matplotlib package to visualize data. The data set you will be working with comes from the 2013 American Community Survey (ACS) data. According to census.gov, ACS "is a mandatory, ongoing statistical survey that samples a small percentage of the population every year -- giving communities the information they need to plan investments and services." [see http://www.census.gov/acs/www/] More specifically, you will be using the ACS Public Use Microdata Sample (PUMS), which census.gov describes as "files [that] are a set of untabulated records about individual people or housing units."

You can download the 2013 ACS 1-year PUMS data for Illinois Housing Unit Records here:
http://www.census.gov/acs/www/data_documentation/pums_data/

You can also access documentation for the PUMS dataset, including the Data Dictionary, here:
http://www.census.gov/acs/www/data_documentation/pums_documentation/

Requirements

You are required to create a program in Python that performs the following using the matplotlib and pandas packages:

1. Loads the ss13hil.csv file that contains the PUMS dataset (assume it's in the current directory) and create a DataFrame object from it.
2. Create a figure with 2x2 subplots. The required subplots are as follows:
   Upper Left Subplot - **Pie chart** containing the number of household records for different values of the HHL (household language) attribute. The plot should have no wedge labels, but should have a legend in the upper left corner. The pie needs to be rotated appropriately (see example figure on last page).
   Upper Right Subplot - **Histogram** of HINCP (household income) **with KDE plot** superimposed. You should use a log scale on the x-axis with log-spaced bins (HINT: use np.logspace).
   Lower Left Subplot - Bar chart of Thousands of Households for each VEH (vehicles available) value (exclude NaN). Make sure to use the WGTP value to count how many households are represented by each household record and divide the sum by 1000.
   Lower Right Subplot - **Scatter plot** of TAXP (property taxes) vs. VALP (property value). Make sure to convert TAXP into the appropriate numeric value, using the lower bound of the interval (e.g. 2 -> $1, 16 -> $700, …). Use WGTP as the size of each marker, 'o' as the marker type, and MRGP (first mortgage payment) as the color value. Add a colorbar.

3. Display the figure and save it in a file called 'pums.png'

Additional Requirements

1. The name of your source code file should be `vispums.py`. All your code should be within a single file.
2. You need to use the pandas DataFrame object for storing data and matplotlib for visualization.
3. Your code should follow good coding practices, including good use of whitespace and use of both inline and block comments.
4. You need to use meaningful identifier names that conform to standard naming conventions.
5. At the top of each file, you need to put in a block comment with the following information: your name, date, course name, semester, and assignment name.
6. The output image file should **exactly** match the sample output shown on the last page.

What to Turn In

You will need to turn in the **single vispums.py file** and the file **pums.png (that has the 2x2 subplots)** using BlackBoard.

Sample Output