GEOG 5222

Final Project--Question 2 User's Manual

12/14/2022

SHANNON CURLEY

Table of Contents

Instructions

1. Open the ‘program_code_FINAL.py’ file using appropriate software to run a Python program.
2. Make sure that the package/module imports map to the correct folders on your computer.
    a. The program makes use of modules from https://github.com/gisalgs
    b. It also uses the ‘xcolorbrewer.py’ file from https://github.com/compgeog/cgl/blob/master/cgl/util/xcolorbrewer.py
3. Press ‘F5’ or the corresponding key for your system to run the program code.
4. Follow the program prompts.
    a. Shapefile input
        i. Copy the file pathway from your computer to a polygon shapefile.
        ii. The program was designed chiefly with the US counties shapefile we used in class, and there is a New York State counties shapefile in a .zip file in the project submission that can also be downloaded and used with the program.
    b. Attribute choice
        i. The program will print a list of numerical attributes that can be mapped
        ii. Choose ONE of these attributes from the list and type the corresponding number into the ‘Choose an attribute:’ prompt
    c. Class number choice
        i. Type in the number that corresponds to the desired number of classes.
    d. Classification method choice
        i. Choose either equal interval (1) or quantile (2) classification by typing in the corresponding number from the program prompt.
    e. Color scheme choice
        i. Choose ONE of three color options by typing in the corresponding number to the scheme as shown in the prompt.
5. Once all the choices have been made, the program will print the name of the mapped attribute and class information and draw the map.
    a. The class information is different depending on the classification method. For equal interval, it prints a list of values that represent class breaks. For quantile, it prints the approximate number of features that fall into each class.
6. You can save the map image if desired.
7. To end the program, close the map image. If you wish to create another map, you can restart the program from Step 3.
    a. If at any point in the above steps you are unhappy with a choice you have made, you can exit the program with ‘Ctrl +SHIFT+ C’ and then restart the program with ‘F5’

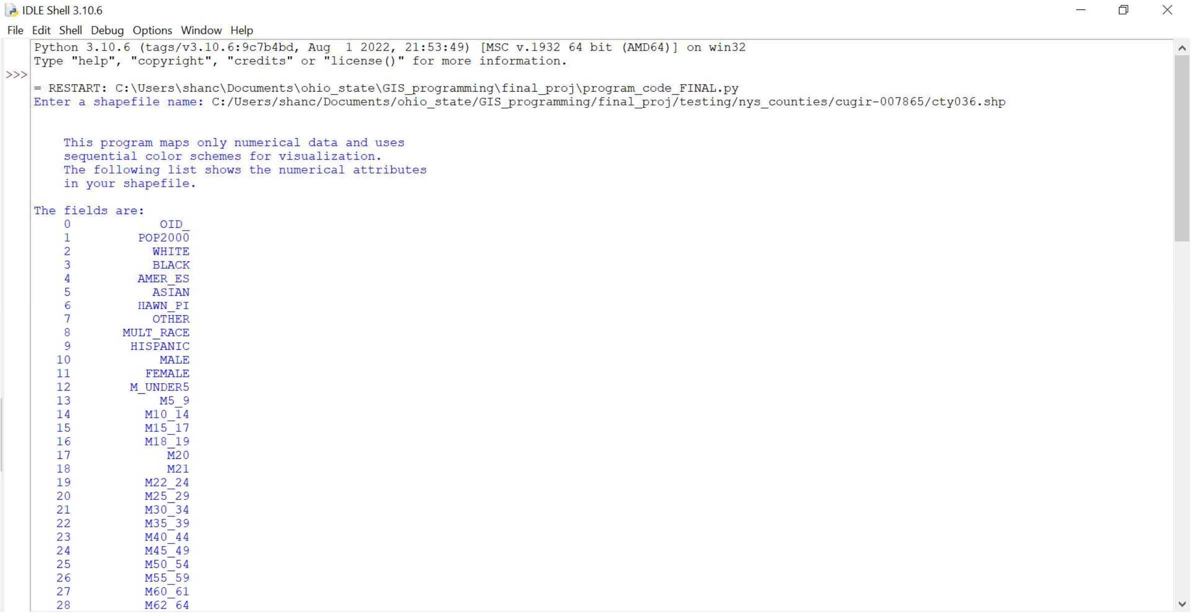
Code Explanation

- Imports (Line 15)
  - The program relies on modules from the 'gisalgs' github and matplotlib to function. It uses shapex to allow shapefiles to be read / searched by the program. It uses an additional module from github 'xcolorbrewer' to get lists of colorbrewer2 color scheme hex codes. It uses matplotlib to prepare polygons for drawing and show the map.
- Shapefile input (Line 295)
  - Tests user's input using shapex() to ensure the file is a shapefile. It uses the try/except method to re-prompt the user for another input if the conditions are not met.
    - NOTE: Any aspect of the project that asks for user input makes use of the try/except logic, so I will not repeat explanation on this aspect.
- Numerical attribute filtering (Line 316) and 'att_choice' function (Line 27)
  - The variable 'f' is assigned a sample feature from the shapefile (Ex. all data associated with Franklin County OH). The sample feature is the first in the shapefile (shp[0]) for simplicity. This sample feature's properties are then searched with a for loop to find attributes that contain float or integer values.
  - These numerical attributes are then appended to list 'num_atts' which are then passed to function att_choice().
    - This function explains key features of the program to the user and uses a for loop to print the items in 'num_atts' as options with their list index number.
    - The user then must choose one attribute from the list.
    - The output of att_choice() is the variable 'map_att'.
- Creation of the 'values' list (Line 329)
  - This next section of code uses a for loop to append values of the chosen attribute to a list called 'values'. Important for classification and later on mapping, this list is based on the original shapefile's index.
- Classification choice and methods (Line 337)
  - This section calls two functions to obtain the number of classes a user wants and the list that represents which class each value is in.
  - 'cl_def' function (Line 78)
    - The user must choose a minimum of 3 and maximum of 9 classes. The limits were set based on color recommendations from colorbrewer2.org. Because the three utilized color schemes are sequential, they necessitate this range.
  - 'cl_method' function (Line 141)
    - The two chosen classifications are both appropriate for choropleth maps with sequential color schemes, which is the output of this program.
    - The cl_method() function takes two arguments: values (list created based on map_att) and cl_num (user chosen number of classes from cl_def() function).
    - Equal interval (Line 174)
      - This classification system is based on the range of the values and number of classes. The function uses sorted() and creates a new list ('sort_vals') to store the values in ascending order because the original list order needs to be retained.

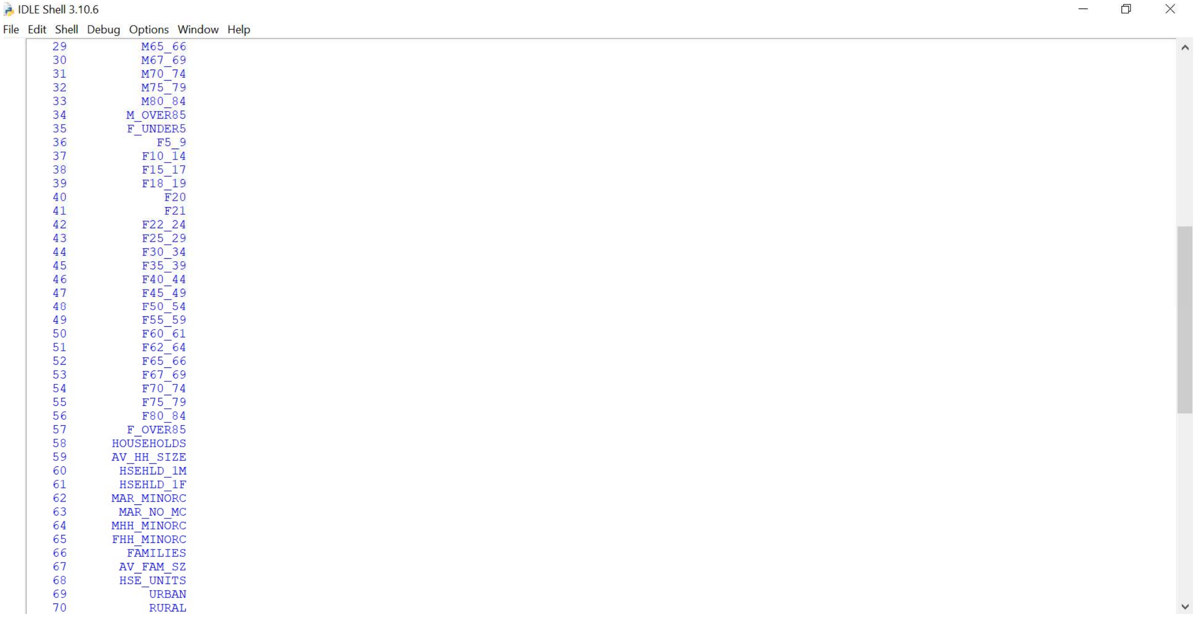
      - A list comprehension creates the class break values (cl_bounds) which are then used in variable ‘info’ which is an output of the function. (To be shown to the user later on.)
      - List ‘val_classes’ is developed based on the order of the original values list. It uses for and while loops to assess each value and a counter (‘b’) that is based on the number of classes and used to create the class break values within the while loop.
    - Quantile (Line 203)
      - This system is based on the same number of features being in each class. For this reason, before sorting, it first creates a list of sub-lists (val_ind) that contain the value and its original index number (Ex. [[2, 0], [140, 1],...]). The new list is then sorted according to value number to create ’val_nums’.
      - The ‘quant_num’ variable is how many features are in each class. This is used in this classification system’s ‘info’ variable and also used with the counter in the loop to assign classes.
      - The loop uses similar logic to the equal interval loop, but instead appends the class number to the sub-lists of ‘sort_vals’.
      - Each item of ‘sort_vals’ now has three items in its list [0-value, 1-original index number, 2-class assignment number]. Using a lambda function, this list is re-sorted based on the original index.
      - Finally, ‘val_classes’ is created by taking only the class assignment numbers from the re-sorted list.
- Color (Line 343)
  - ‘colorchoice’ function (Line 105)
    - The class options range 3-9 because the program uses colorbrewer2’s sequential color schemes which are appropriate for this range.
  - Colorbrewerx.py file
    - This section also uses of get_colorbrewer_specs() from the colorbrewerx.py file to obtain the list of color hex codes and assign them to variable ‘colors’.
- Program outputs (Line 348)
  - The first output is contextual information about the map: the name from the shapefile of the mapped attribute and the ‘info’ as prepared in cl_method().
  - Drawing of the map image (Line 353 to the end)
    - Uses matplotlib functions and an edited version of draw_polygon() to start the drawing and prepare polygons for mapping
    - The draw_polygon() function was edited to move definition of the plot’s x/y limits out to the overall program code. This is because the limits need to be based on all polygons in the shapefile, not just one feature. The first feature in shapefile is used to create the first x/y min/max variables which are then redefined using if statements.
    - The final x/y limits are used to draw the plot’s boundaries and the map is shown.

Results of Running the Project Code

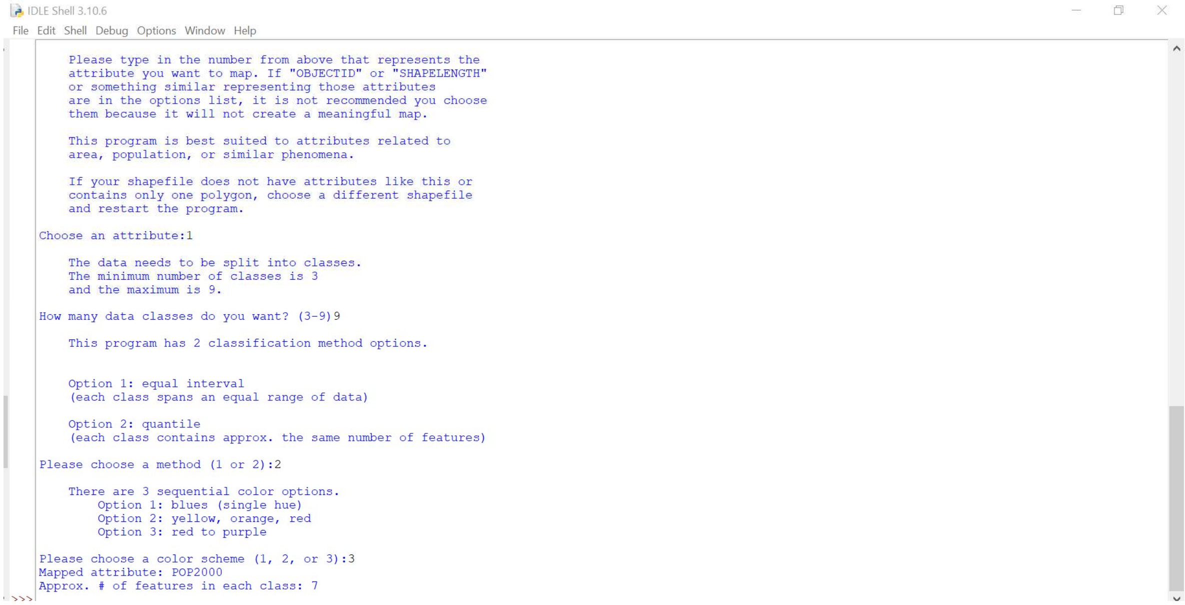
The following images show one code run through of the program and the resulting map. This is an input scenario in which the user inputs an acceptable choice every time.



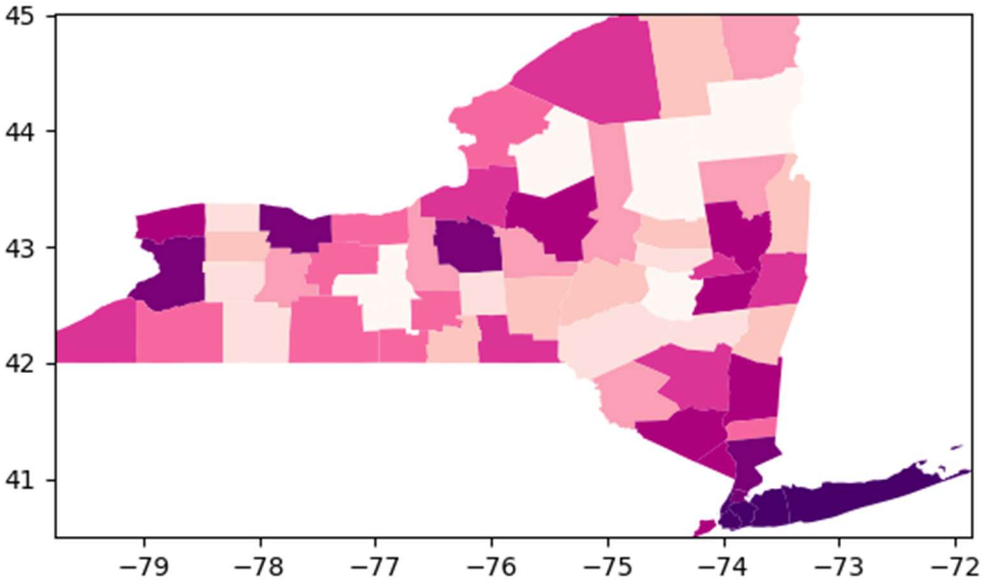
```
Python 3.10.6 (tags/v3.10.6:9c7b4bd, Aug  1 2022, 21:53:49) [MSC v.1932 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\shanc\Documents\ohio_state\GIS_programming\final_proj\program_code_FINAL.py
Enter a shapefile name: C:/Users/shanc/Documents/ohio_state/GIS_programming/final_proj/testing/nys_counties/cugir-007865/cty036.shp


    This program maps only numerical data and uses
    sequential color schemes for visualization.
    The following list shows the numerical attributes
    in your shapefile.

The fields are:
    0            OID_
    1         POP2000
    2           WHITE
    3           BLACK
    4         AMER_ES
    5           ASIAN
    6         HAWN_PI
    7           OTHER
    8       MULT_RACE
    9        HISPANIC
   10            MALE
   11          FEMALE
   12        M_UNDER5
   13            M5_9
   14          M10_14
   15          M15_17
   16          M18_19
   17             M20
   18             M21
   19          M22_24
   20          M25_29
   21          M30_34
   22          M35_39
   23          M40_44
   24          M45_49
   25          M50_54
   26          M55_59
   27          M60_61
   28          M62_64
```



```
   29          M65_66
   30          M67_69
   31          M70_74
   32          M75_79
   33          M80_84
   34        M_OVER85
   35        F_UNDER5
   36            F5_9
   37          F10_14
   38          F15_17
   39          F18_19
   40             F20
   41             F21
   42          F22_24
   43          F25_29
   44          F30_34
   45          F35_39
   46          F40_44
   47          F45_49
   48          F50_54
   49          F55_59
   50          F60_61
   51          F62_64
   52          F65_66
   53          F67_69
   54          F70_74
   55          F75_79
   56          F80_84
   57        F_OVER85
   58      HOUSEHOLDS
   59      AV_HH_SIZE
   60       HSEHLD_1M
   61       HSEHLD_1F
   62      MAR_MINORC
   63       MAR_NO_MC
   64      MHH_MINORC
   65      FHH_MINORC
   66        FAMILIES
   67       AV_FAM_SZ
   68       HSE_UNITS
   69           URBAN
   70           RURAL
```

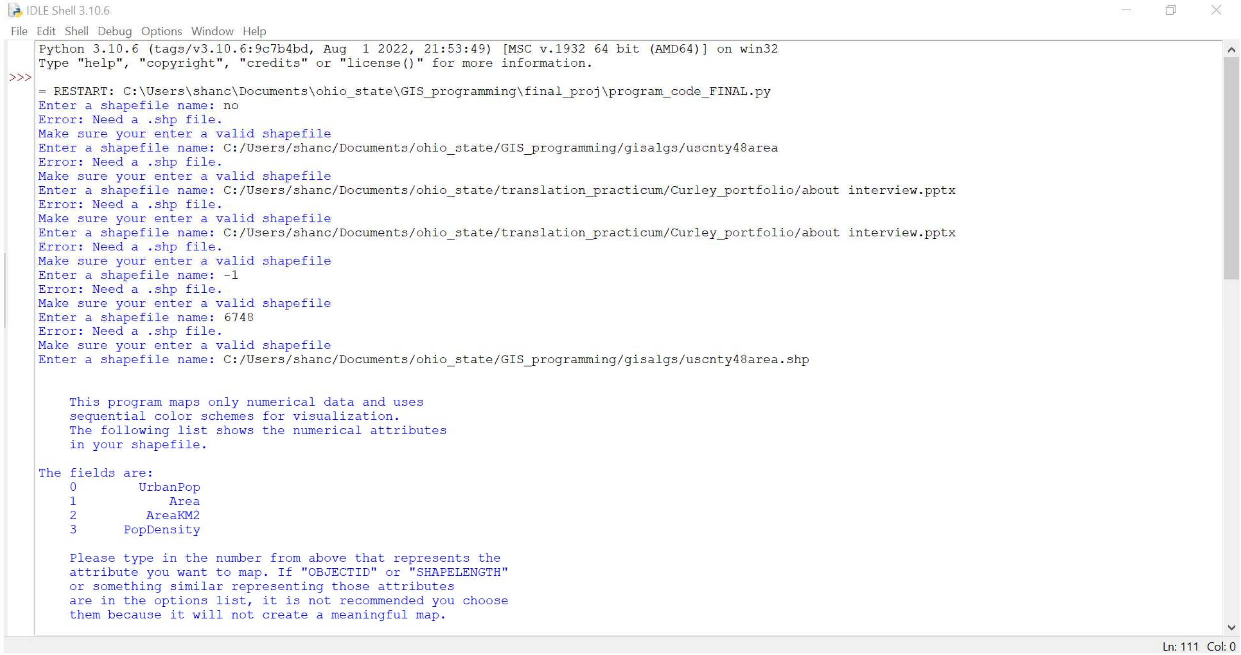


```
    Please type in the number from above that represents the
    attribute you want to map. If "OBJECTID" or "SHAPELENGTH"
    or something similar representing those attributes
    are in the options list, it is not recommended you choose
    them because it will not create a meaningful map.

    This program is best suited to attributes related to
    area, population, or similar phenomena.

    If your shapefile does not have attributes like this or
    contains only one polygon, choose a different shapefile
    and restart the program.

Choose an attribute:1

    The data needs to be split into classes.
    The minimum number of classes is 3
    and the maximum is 9.

How many data classes do you want? (3-9)9

    This program has 2 classification method options.


    Option 1: equal interval
    (each class spans an equal range of data)

    Option 2: quantile
    (each class contains approx. the same number of features)

Please choose a method (1 or 2):2

    There are 3 sequential color options.
        Option 1: blues (single hue)
        Option 2: yellow, orange, red
        Option 3: red to purple

Please choose a color scheme (1, 2, or 3):3
Mapped attribute: POP2000
Approx. # of features in each class: 7
>>>
```

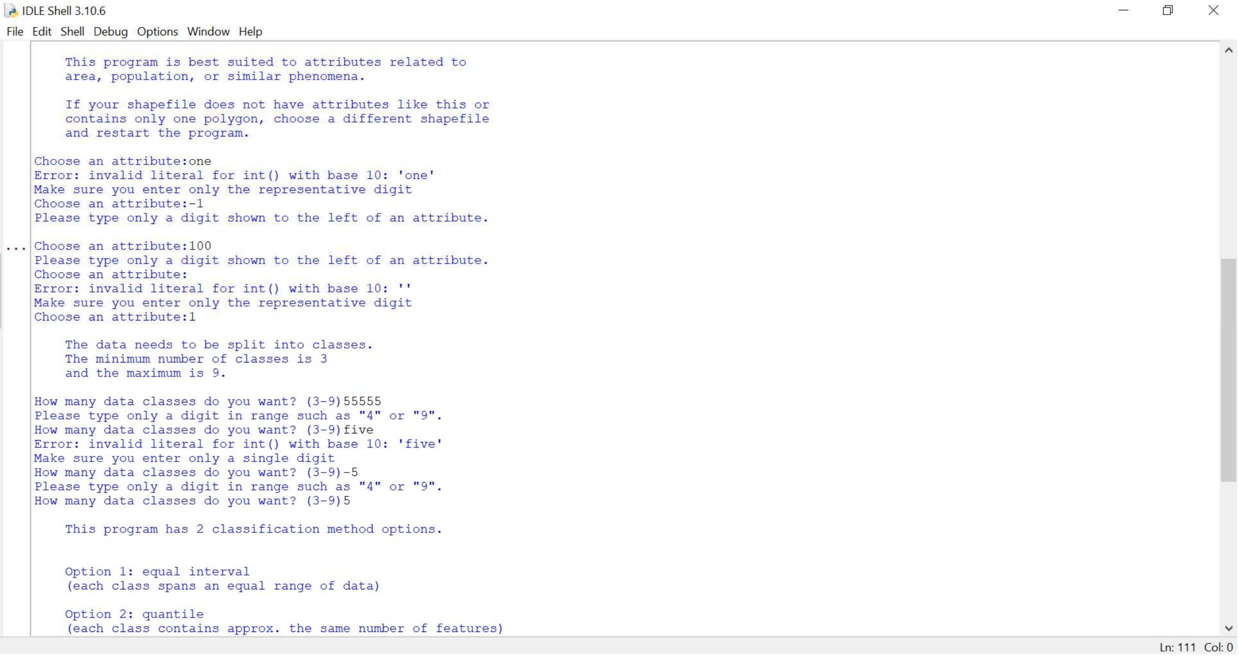


Testing Scenarios

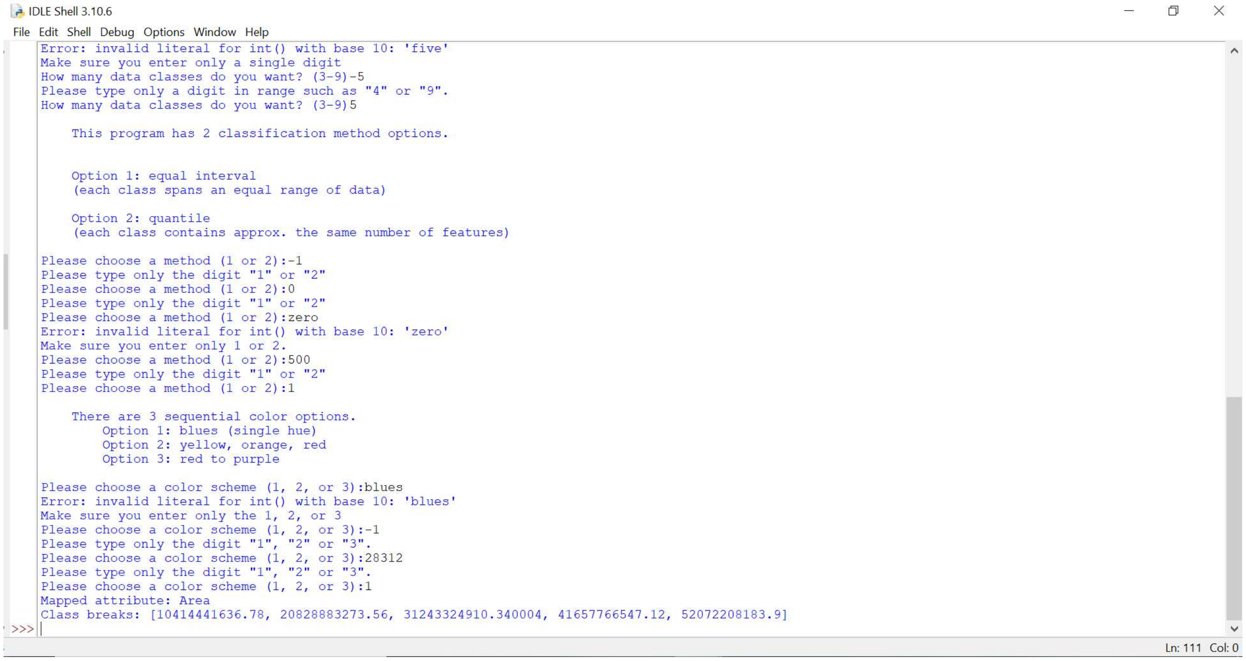
The following images show a code run through with many incorrect user inputs. The program has been tested for negative inputs (Ex. -3), out of range inputs (Ex. 2348325023 or 0), and text inputs (Ex. one) in each of the input areas. This run also utilizes a different shapefile than the above run to show that the program is flexible to any shapefile. The resulting image is also included after the program screenshots.



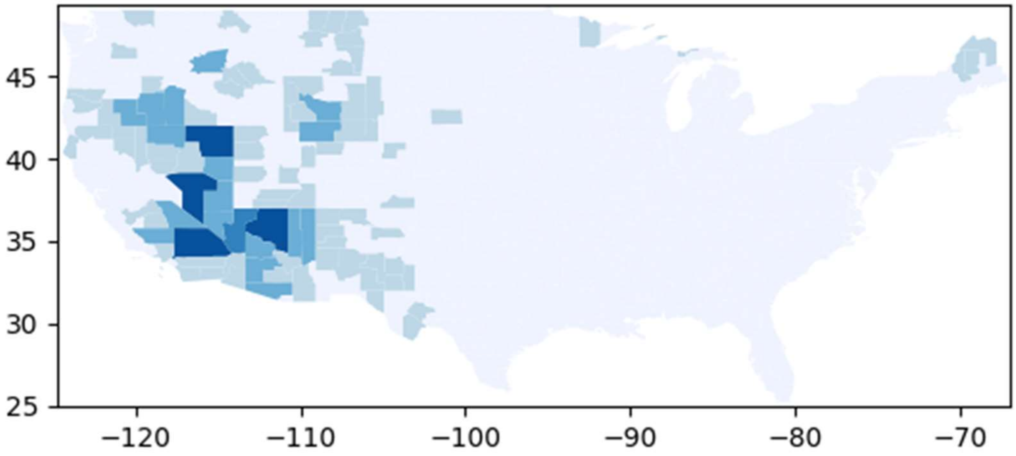
```
Python 3.10.6 (tags/v3.10.6:9c7b4bd, Aug  1 2022, 21:53:49) [MSC v.1932 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\shanc\Documents\ohio_state\GIS_programming\final_proj\program_code_FINAL.py
Enter a shapefile name: no
Error: Need a .shp file.
Make sure your enter a valid shapefile
Enter a shapefile name: C:/Users/shanc/Documents/ohio_state/GIS_programming/gisalgs/uscnty48area
Error: Need a .shp file.
Make sure your enter a valid shapefile
Enter a shapefile name: C:/Users/shanc/Documents/ohio_state/translation_practicum/Curley_portfolio/about interview.pptx
Error: Need a .shp file.
Make sure your enter a valid shapefile
Enter a shapefile name: C:/Users/shanc/Documents/ohio_state/translation_practicum/Curley_portfolio/about interview.pptx
Error: Need a .shp file.
Make sure your enter a valid shapefile
Enter a shapefile name: -1
Error: Need a .shp file.
Make sure your enter a valid shapefile
Enter a shapefile name: 6748
Error: Need a .shp file.
Make sure your enter a valid shapefile
Enter a shapefile name: C:/Users/shanc/Documents/ohio_state/GIS_programming/gisalgs/uscnty48area.shp


    This program maps only numerical data and uses
    sequential color schemes for visualization.
    The following list shows the numerical attributes
    in your shapefile.

The fields are:
    0        UrbanPop
    1            Area
    2         AreaKM2
    3      PopDensity

    Please type in the number from above that represents the
    attribute you want to map. If "OBJECTID" or "SHAPELENGTH"
    or something similar representing those attributes
    are in the options list, it is not recommended you choose
    them because it will not create a meaningful map.
```



```
    This program is best suited to attributes related to
    area, population, or similar phenomena.

    If your shapefile does not have attributes like this or
    contains only one polygon, choose a different shapefile
    and restart the program.

Choose an attribute:one
Error: invalid literal for int() with base 10: 'one'
Make sure you enter only the representative digit
Choose an attribute:-1
Please type only a digit shown to the left of an attribute.

... Choose an attribute:100
Please type only a digit shown to the left of an attribute.
Choose an attribute:
Error: invalid literal for int() with base 10: ''
Make sure you enter only the representative digit
Choose an attribute:1

    The data needs to be split into classes.
    The minimum number of classes is 3
    and the maximum is 9.

How many data classes do you want? (3-9)55555
Please type only a digit in range such as "4" or "9".
How many data classes do you want? (3-9)five
Error: invalid literal for int() with base 10: 'five'
Make sure you enter only a single digit
How many data classes do you want? (3-9)-5
Please type only a digit in range such as "4" or "9".
How many data classes do you want? (3-9)5

    This program has 2 classification method options.


    Option 1: equal interval
    (each class spans an equal range of data)

    Option 2: quantile
    (each class contains approx. the same number of features)
```



```
Error: invalid literal for int() with base 10: 'five'
Make sure you enter only a single digit
How many data classes do you want? (3-9)-5
Please type only a digit in range such as "4" or "9".
How many data classes do you want? (3-9)5

    This program has 2 classification method options.


    Option 1: equal interval
    (each class spans an equal range of data)

    Option 2: quantile
    (each class contains approx. the same number of features)

Please choose a method (1 or 2):-1
Please type only the digit "1" or "2"
Please choose a method (1 or 2):0
Please type only the digit "1" or "2"
Please choose a method (1 or 2):zero
Error: invalid literal for int() with base 10: 'zero'
Make sure you enter only 1 or 2.
Please choose a method (1 or 2):500
Please type only the digit "1" or "2"
Please choose a method (1 or 2):1

    There are 3 sequential color options.
        Option 1: blues (single hue)
        Option 2: yellow, orange, red
        Option 3: red to purple

Please choose a color scheme (1, 2, or 3):blues
Error: invalid literal for int() with base 10: 'blues'
Make sure you enter only the 1, 2, or 3
Please choose a color scheme (1, 2, or 3):-1
Please type only the digit "1", "2" or "3".
Please choose a color scheme (1, 2, or 3):28312
Please type only the digit "1", "2" or "3".
Please choose a color scheme (1, 2, or 3):1
Mapped attribute: Area
Class breaks: [10414441636.78, 20828883273.56, 31243324910.340004, 41657766547.12, 52072208183.9]
>>>
```



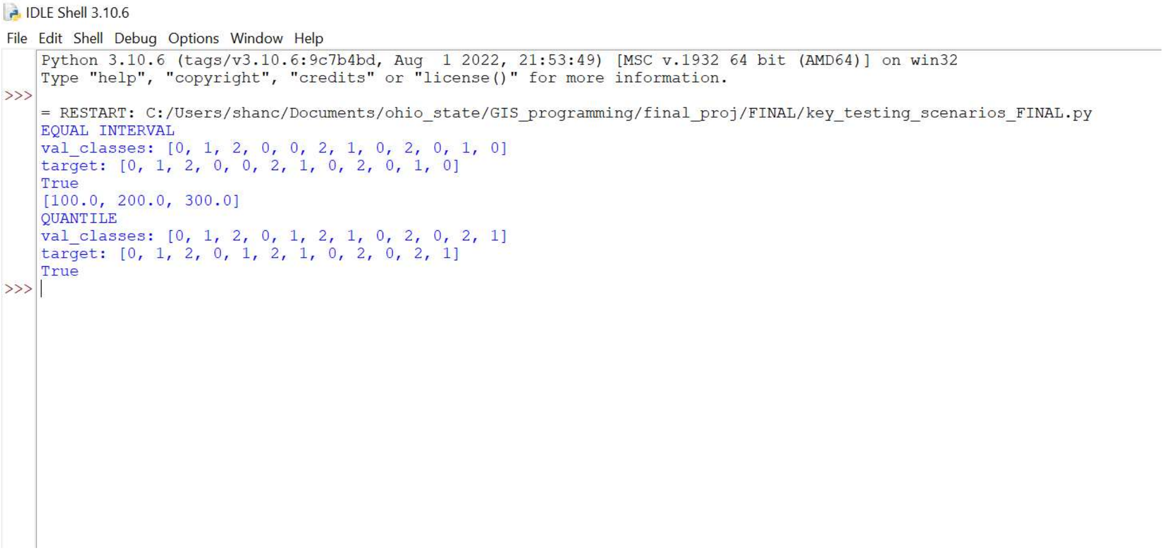
The project submission also includes a testing file (‘key_testing_scenarios_FINAL.py’) that tests key parts of the program code. Each of these scenarios was developed as its own .py file, corrected for errors, and compiled into this testing file with sample data that shows how the parts of the code run. Below is a list explaining each tested scenario and its data. Following the list is a screenshot of the testing code when it is run and the output map.

The explanations below are also in the .py file as comments.

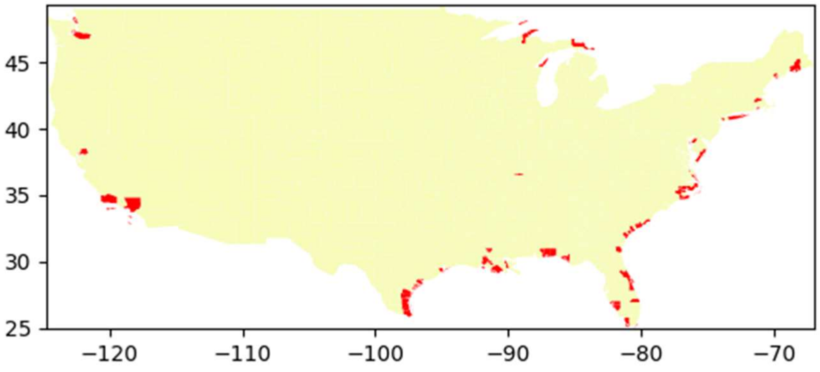
- EQUAL INTERVAL CLASSIFICATION TESTING
    - KEY TESTS:
        - Short 'values' list which was used to write the code.
        - List 'target' shows what the output 'val_classes' should be (This list only works with these values)
        - Second values list can be un-commented to show that the code works when the list does not start at value 0 and has decimals
        - If running the second list, make sure to comment out lines related to the 'target' list
        - Creation of 'cl_bounds' list that shows class breaks
    - LOGIC:
        - Sorts values in ascending order and uses them to calculate class range
        - Uses original values list and a loop to assign class #
    - OUTPUT:
        - Print of the 'val_classes' list
        - Print of 'target' list
        - Statement testing that val_classes and target are identical lists
        - Print of 'cl_bounds'

- QUANTILE CLASSIFICATION TESTING
    - KEY TESTS:
        - Short 'values' list which was used to write the code.
        - List 'target' shows what the output 'val_classes' should be (This list only works with these values)
    - LOGIC:
        - Turns 'values' into sub-lists of [value, index #] (Ex. [[2, 0], [140, 1],...])
        - Sorts the list in ascending order based on value (v[0])
        - Appends a class # to the sub-list
        - Uses a lambda function to re-sort the list based on the original index (v[1])
        - Creates a list of only class #s (v[2]) in original value order
    - OUTPUT:
        - Print of the 'val_classes' list
        - Print of 'target' list
        - Statement testing that val_classes and target are identical lists
        - Print of 'cl_bounds'

- DRAW TESTING
  - KEY TESTS:
    - Changes to the 'draw_polygon' function
      - Moving the x and y min/max portions and plot show to program code
      - x/y limits now encompass entire dataset
    - Loop to draw each part of a multi-polygon
      - Multi-polygons colored in red to show they are being drawn correctly
  - OUTPUT:
    - Map image



```
Python 3.10.6 (tags/v3.10.6:9c7b4bd, Aug  1 2022, 21:53:49) [MSC v.1932 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/shanc/Documents/ohio_state/GIS_programming/final_proj/FINAL/key_testing_scenarios_FINAL.py
EQUAL INTERVAL
val_classes: [0, 1, 2, 0, 0, 2, 1, 0, 2, 0, 1, 0]
target: [0, 1, 2, 0, 0, 2, 1, 0, 2, 0, 1, 0]
True
[100.0, 200.0, 300.0]
QUANTILE
val_classes: [0, 1, 2, 0, 1, 2, 1, 0, 2, 0, 2, 1]
target: [0, 1, 2, 0, 1, 2, 1, 0, 2, 0, 2, 1]
True
>>> |
```



Discussion and Conclusions

Discussion

The additional dataset I used to test this program, a New York State counties shapefile (downloaded from https://gis.ny.gov/gisdata/inventories/details.cfm?DSID=927; available in the .zip file of the project submission), helped illuminate some key aspects of my code and the way that it works with different shapefiles. Unlike the US counties shapefile we used in class, this shapefile has over 70 numerical attributes that are able to be mapped by the program. An important thing to recognize about this NYS dataset is that the first listed numerical attribute is 'OID_'. Unlike the US counties shapefile where the Object ID field was not encoded as numerical, it is numerical in the NYS shapefile. This prompted me to include the warning to users about choosing such an attribute for mapping.

The classification method code was the most difficult part of this project. I wrote what I thought was fool-proof equal interval classification code and then realized through testing the NYS shapefile that it was incompatible with value lists that did not begin with the value 0. In this way, the correct mechanism for writing the correct classification code was not immediately apparent but was perfected through testing. This experience prompted me to test the rest of the code even more thoroughly.

Conclusions

The result of the project is a program that allows for multiple instances of user customization and produces a choropleth map. The other print-out results are (1) name of the attribute mapped and (2) information about the class breaks [equal interval] or number of features in each class [quantile]. These printouts can help the user better understand the map they have just created. The project can be run as many times as a user would like with a variety of polygon shapefiles. The program only uses sequential color schemes and best suits data that should be visualized in this way. Future iterations of the program could include options for diverging color schemes and other classification methods.

Ultimately, the program makes choropleth maps based on quantitative attributes in shapefiles; however, the meaning and usefulness of these maps is based on the quality of the data in the shapefile and the user's input choices (just like a GIS software!).