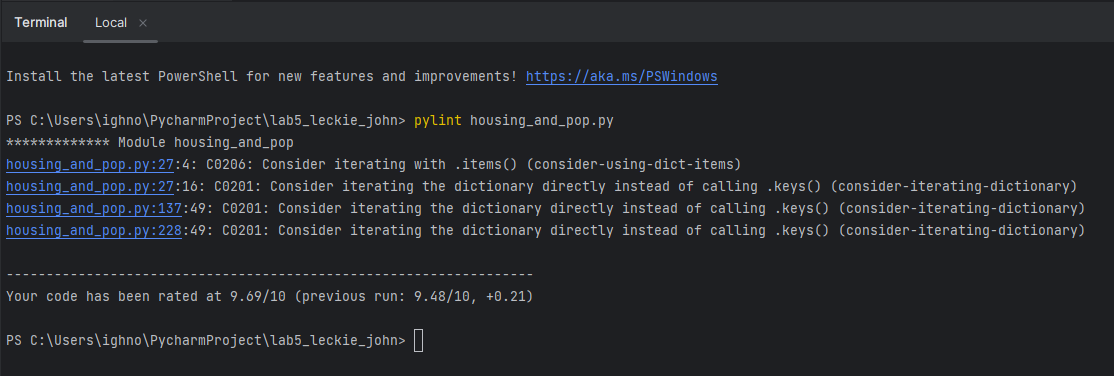
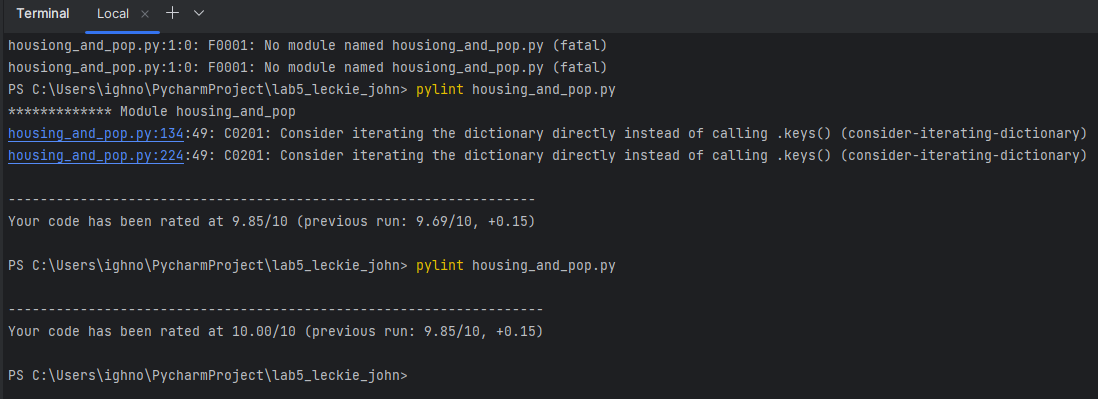
Test matrix for Lab 5, SDEV 300 6381, John Leckie

The following table outlines testing scenarios for Lab 5. Pylint results are first, followed by a table of the testing matrix, and then screens of the multiple runs used to obtain the results that went into the matrix. In the screenshots that appear below the matrix, there will be titles indicating to which test case(s) they are applicable.

PYLINT SCREENS



I did a cursory search on this. I fixed it by changing key to item. It now iterates over dictionaries using items instead of keys. I took out 3 or 4 calls to keys, which I think means it’s more efficient. You get the key and the value. Maybe this means that in really large dictionaries there is a time savings.



Okay, now if fixing this didn’t break my delicate program in some way, time for testing. The documentation and testing for programs in both classes I’m taking have taken a great deal of effort, almost as much as the coding itself!

TEST MATRIX

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TEST CASE | HOW TESTED | EXPECTED OUTPUT | ACTUAL OUTPUT | PASS (Y/N)? |
| 1. Main menu tomfoolery and immediate exit check | On being asked which dataset they wish to analyze, user tries ‘Hello’ ‘123’ ‘$$#’ ‘4’ and finally hits option 3, exit. | Repeatedly remind user of expected inputs on a loop, until ‘3’ is entered for the Exit option. Clean exit of program, goodbye message. | As expected,  See Figure 1 below. | Y |
| 2. Population Data numerical menu selection enforcement and function check | User attempted in Test Case 1 to enter ‘Hello’ ‘123’ ‘$$%’ ‘4’ so here we enter the number 1 for Population.  Here, we will test the menu presented for Pop Data, options 1 through 4, by entering our test characters again, substituting a ‘5’ for the ‘4’ since it’s an option in this case.  Histogram answer will be tested in Test Case 3 | Function check should show proper analysis and display of histogram.  NOTE—for Test case 2, we will also ensure main menu \*return\* options are also enforced, by using ‘Hello’ ‘123’ ‘$$#’ and ‘4’ for entries. | Failure—Using the ‘123’ nonsense option resulted in a return to the root menu. The number ‘5’ also returned to root menu.  All other input types resulted in admonishment and re-print of menu.  See Figure 2 below. | N |
| 3. Housing Data function check | Menu entry enforcement was exercised in Test Case 2. In this test, we’ll mess with the histogram display question, entering ‘1’ ‘2’ ‘Yes’ instead of ‘y’ or ‘n’ | User should be reminded of expected ‘y’ or ‘n’ entry on the nonsense attempts, until ‘y’ (or ‘Y’) is entered.  Following is Function check for Housing data, should show proper analysis and display histogram on demand. | Failure—Function checks were good, but error checking on the Y/N question on histogram resulted in being brought back a level in the menus.  See Figure 3 below. | N |
| 4. Happy Path, all menu options exercised, all histograms viewed | Ran through program with expected inputs sequentially. | User should see analysis table of desired column’s data, followed by an optional Histogram. | EAs Expected, see Figure 4 below. | Y |

Please note, the failures were chased down and fixed after testing. I have run through it several times in this current version and it now successfully enforces the inputs. I had failed to put try/excepts in the correct positions in the original code. Figure 4 contains screenshots of all successful actions.

FIGURE 1

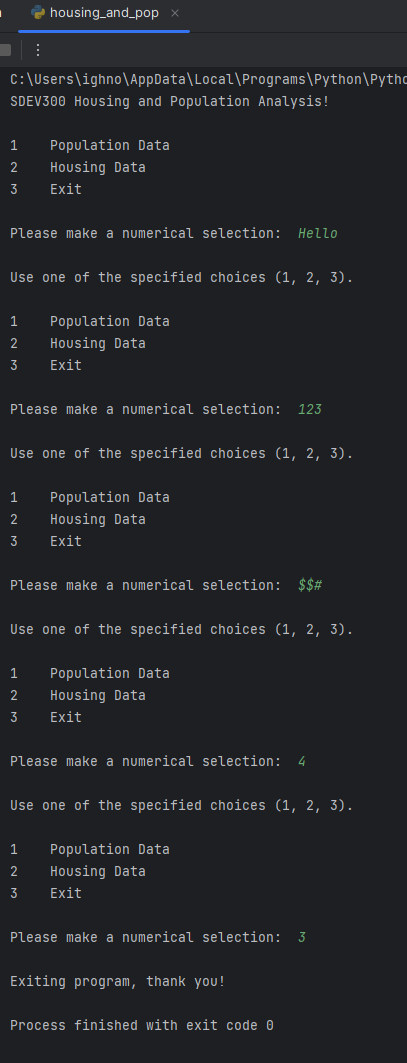


FIGURE 2

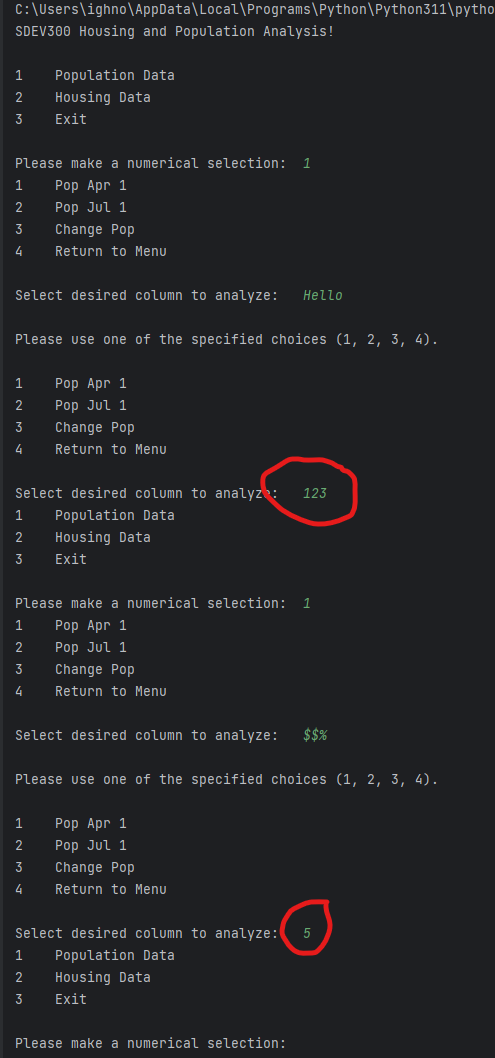
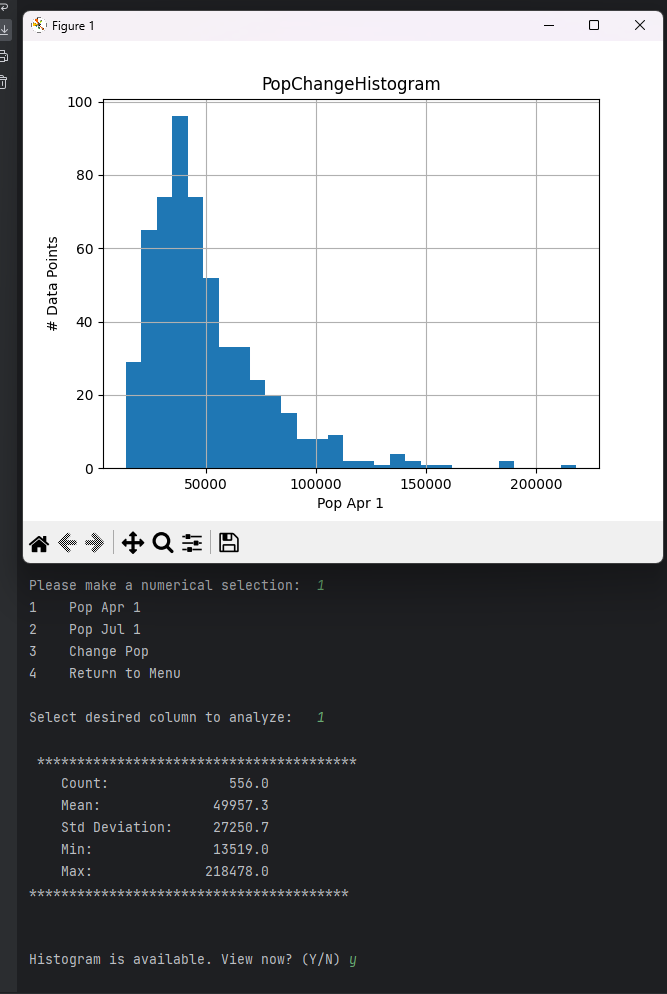
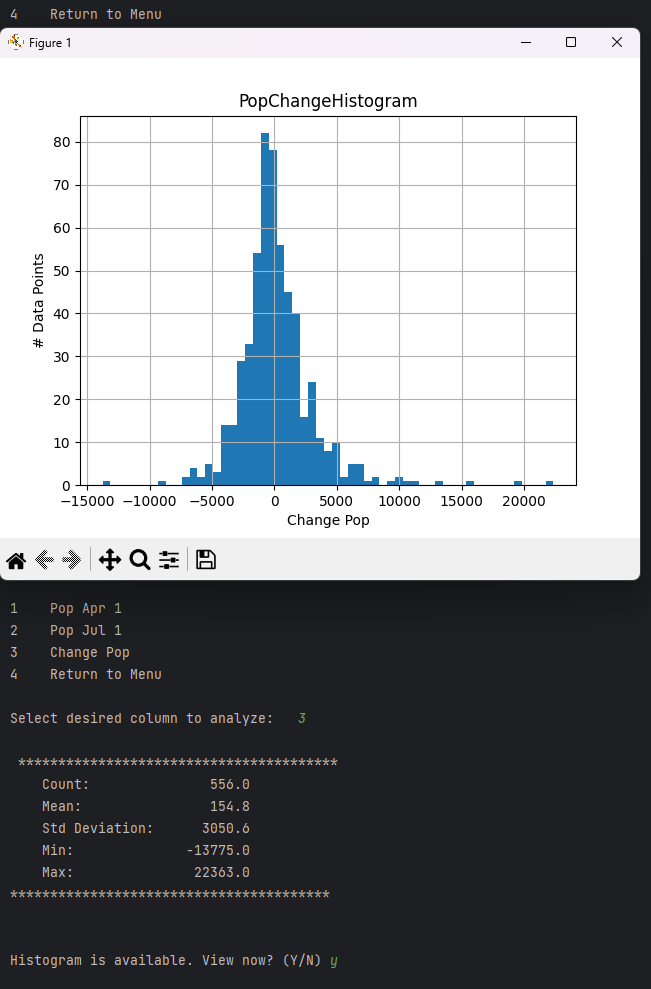
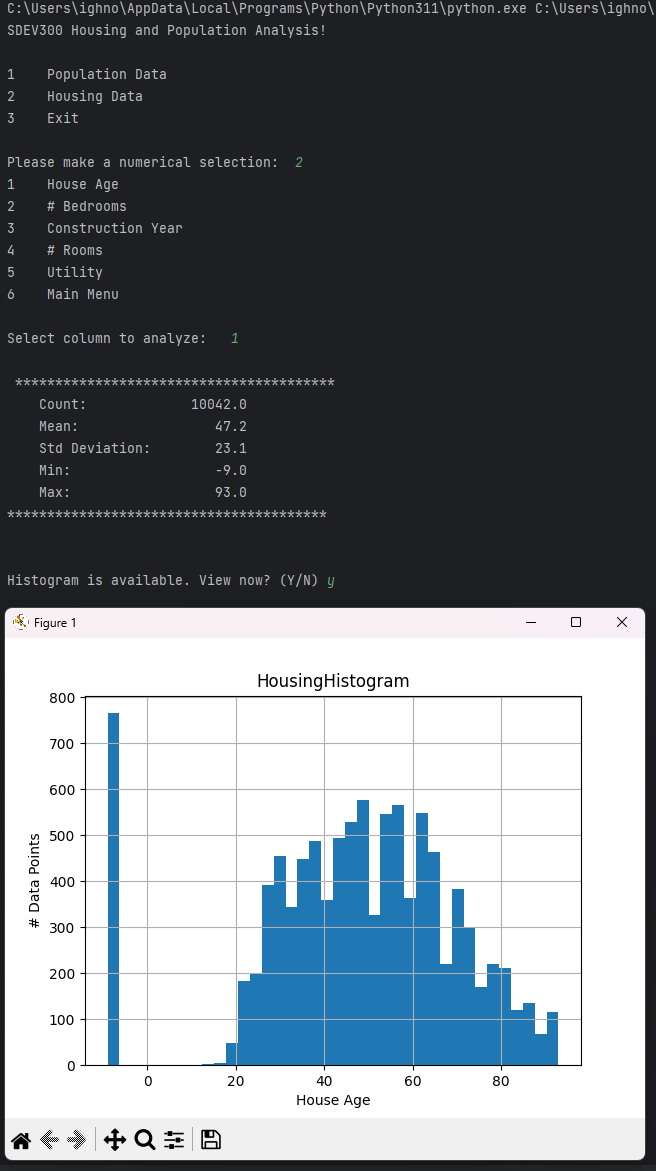


FIGURE 3







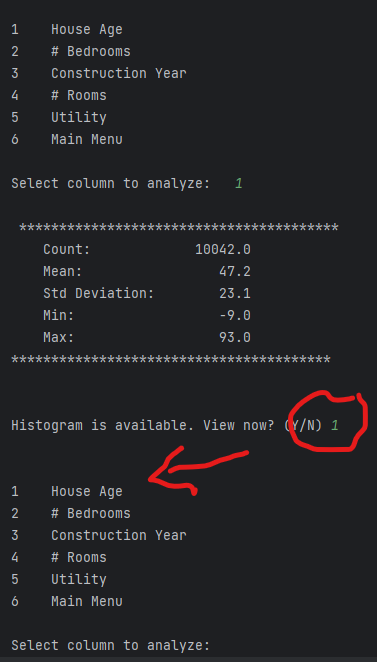
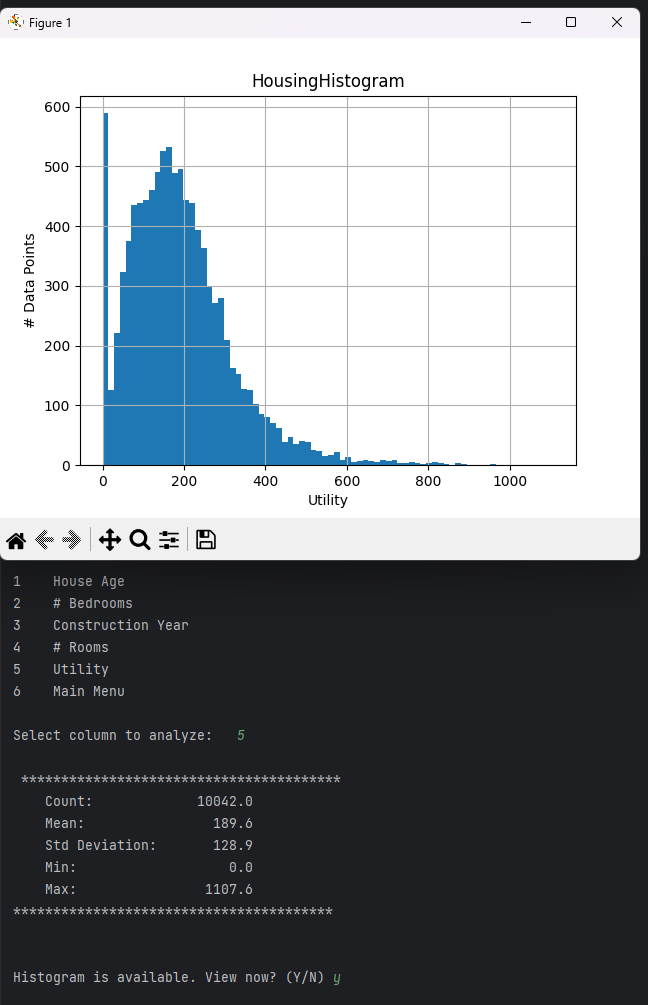
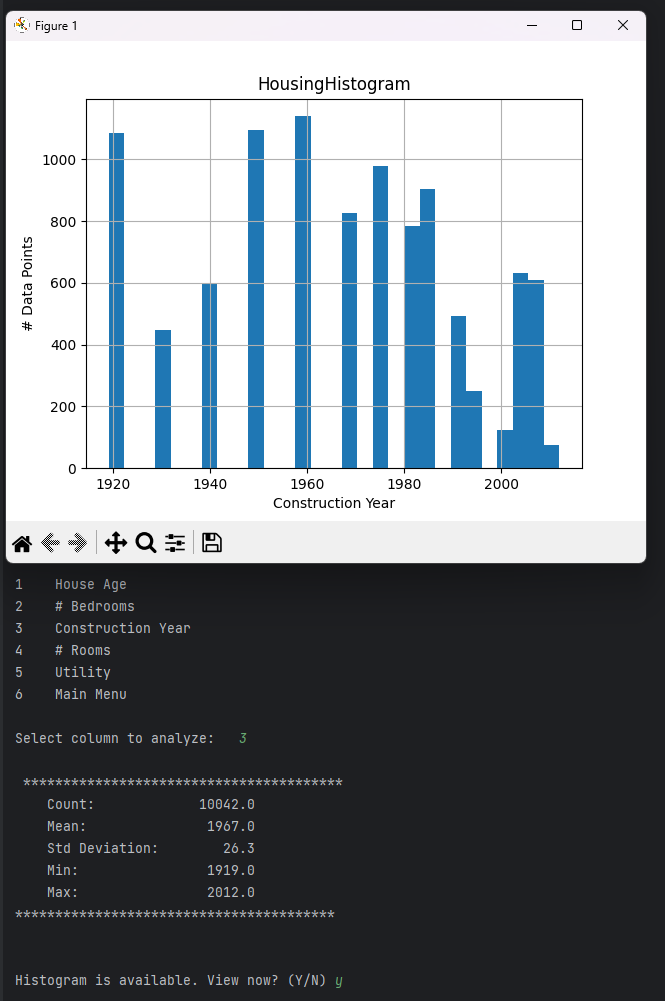
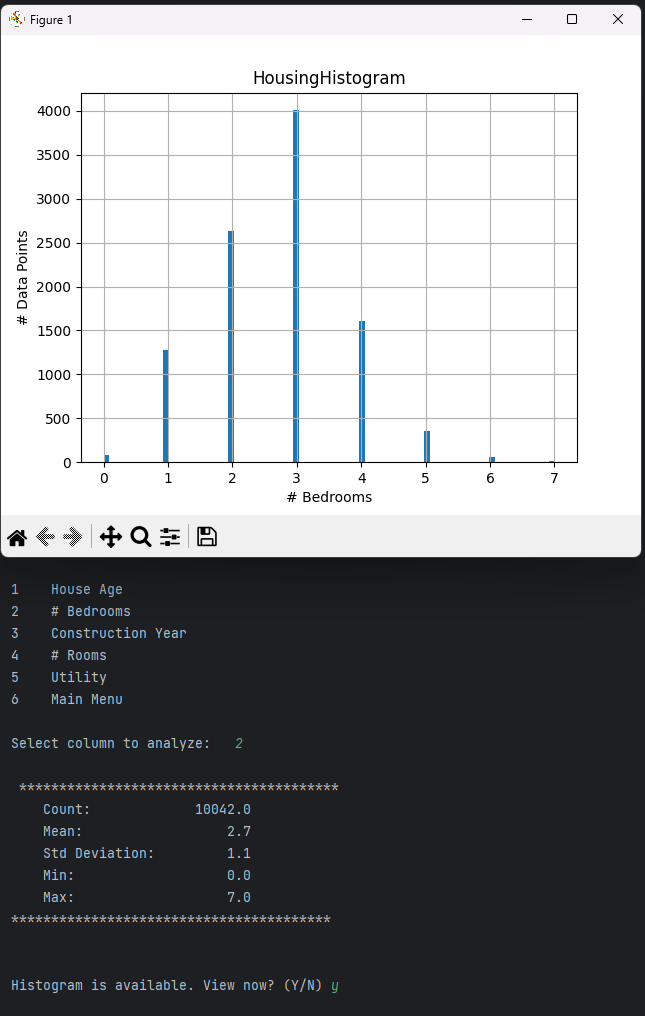
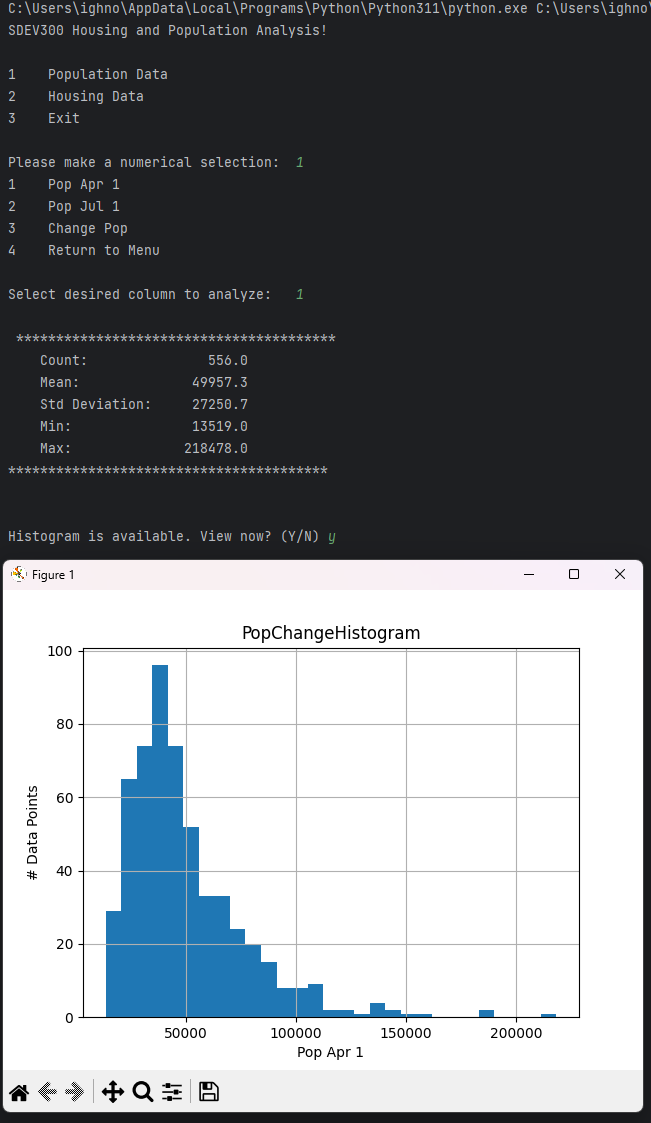


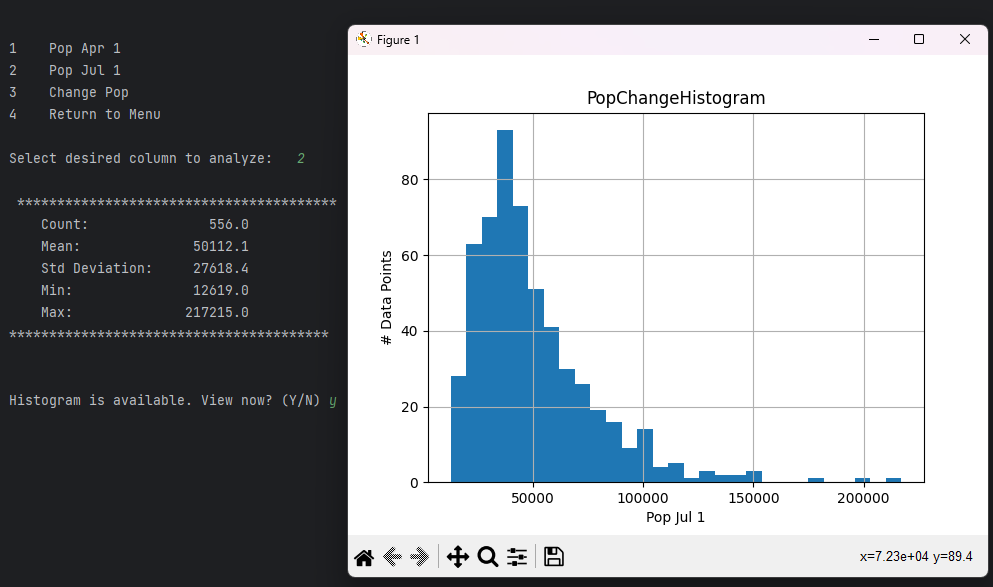
FIGURE 4, HAPPY PATH, ALL MENU OPTIONS

POPULATION DATA

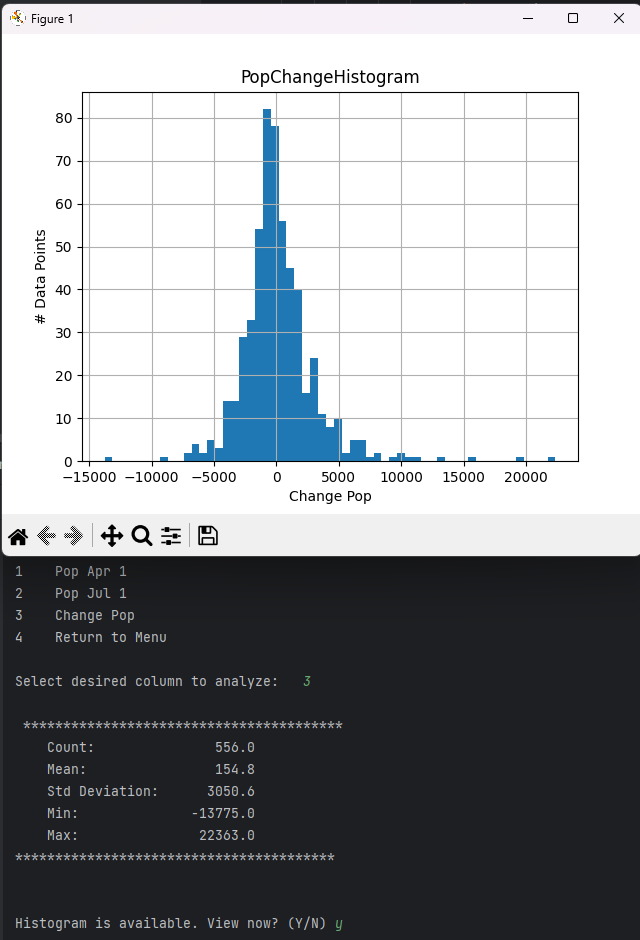
Pop Apr 1



Pop Jul 1

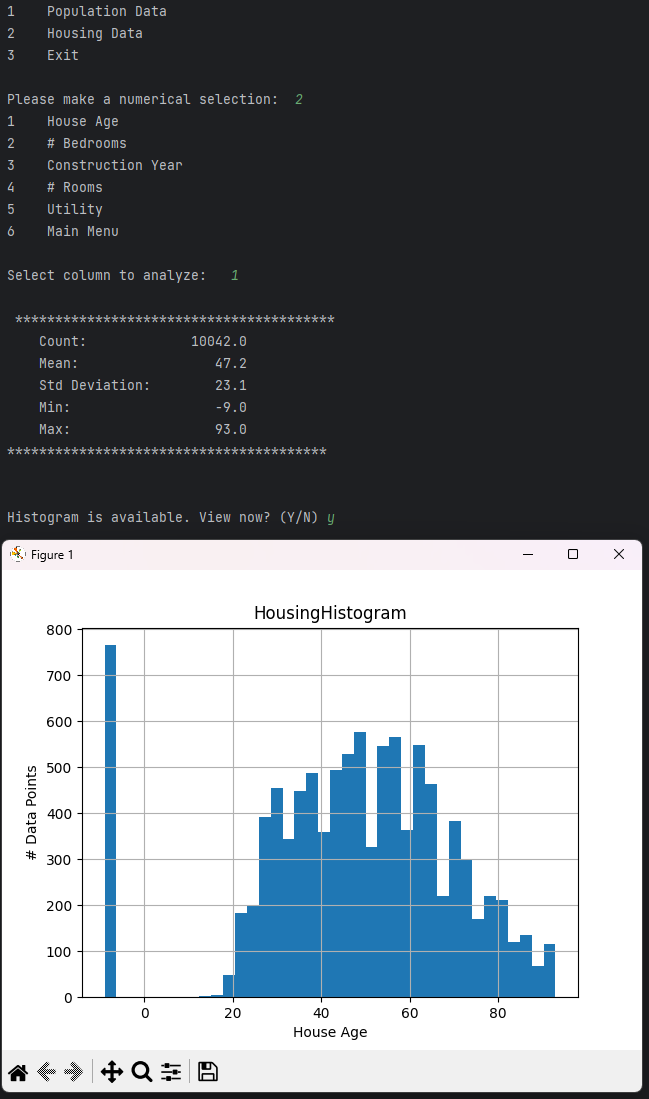


Change Pop

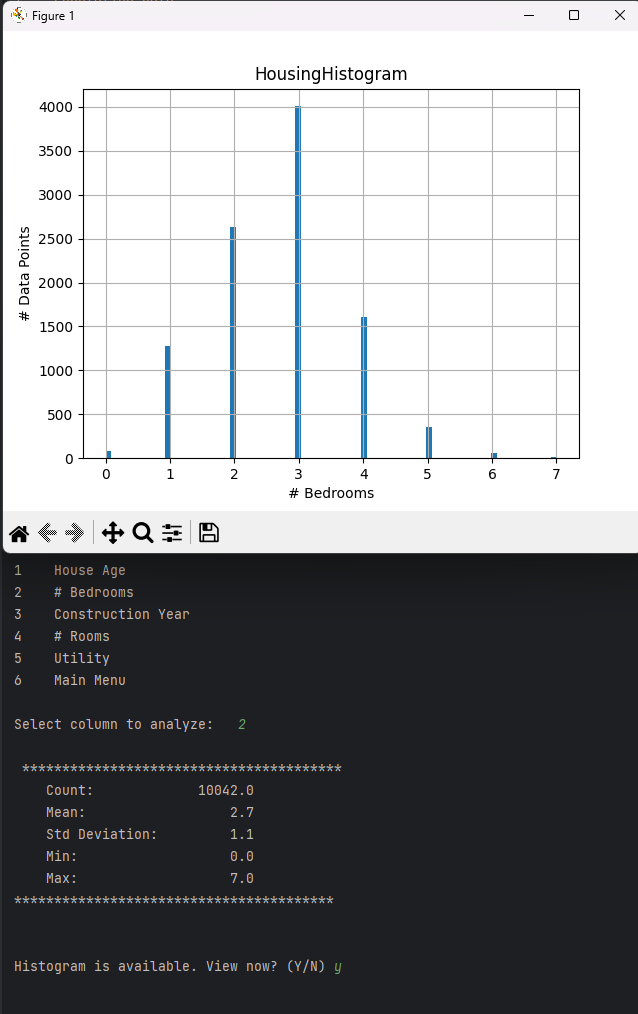


POPULATION DATA

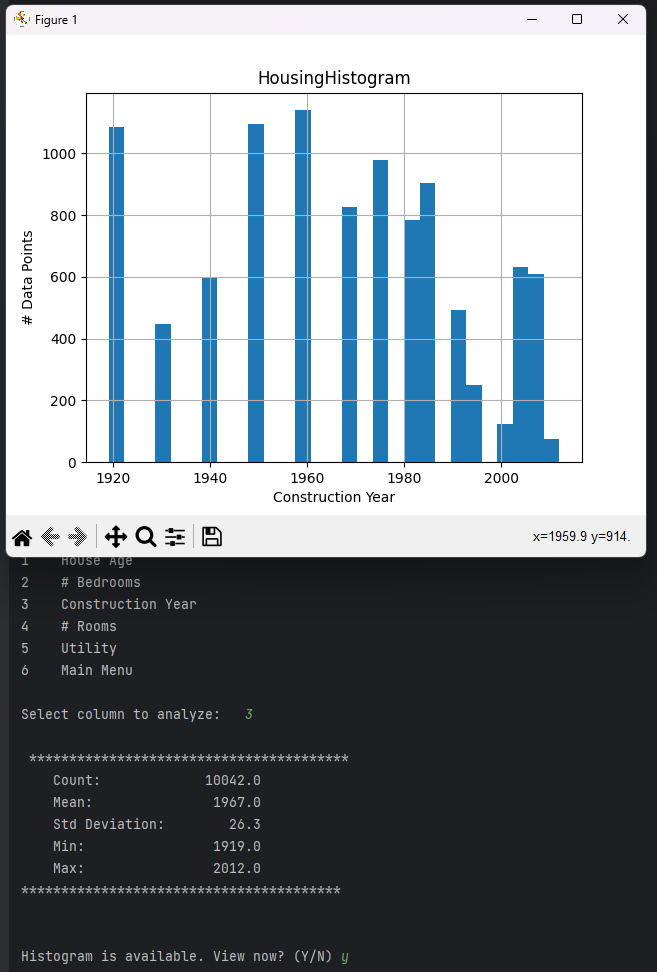
House Age



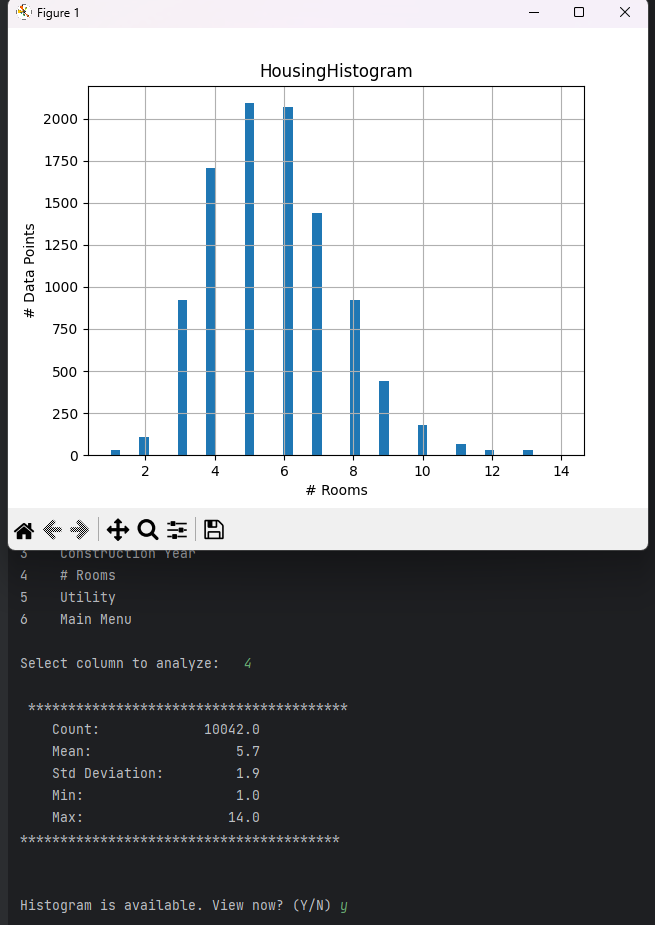
# Bedrooms



Const Year



# Rooms



Utility

