# Learning Proper and Ethical Data Classification for Thematic Maps

Rex Cammack

University of Nebraska Omaha

Department of Geography and Geology

# Agenda

Why this Topic

The Process

First Implementation

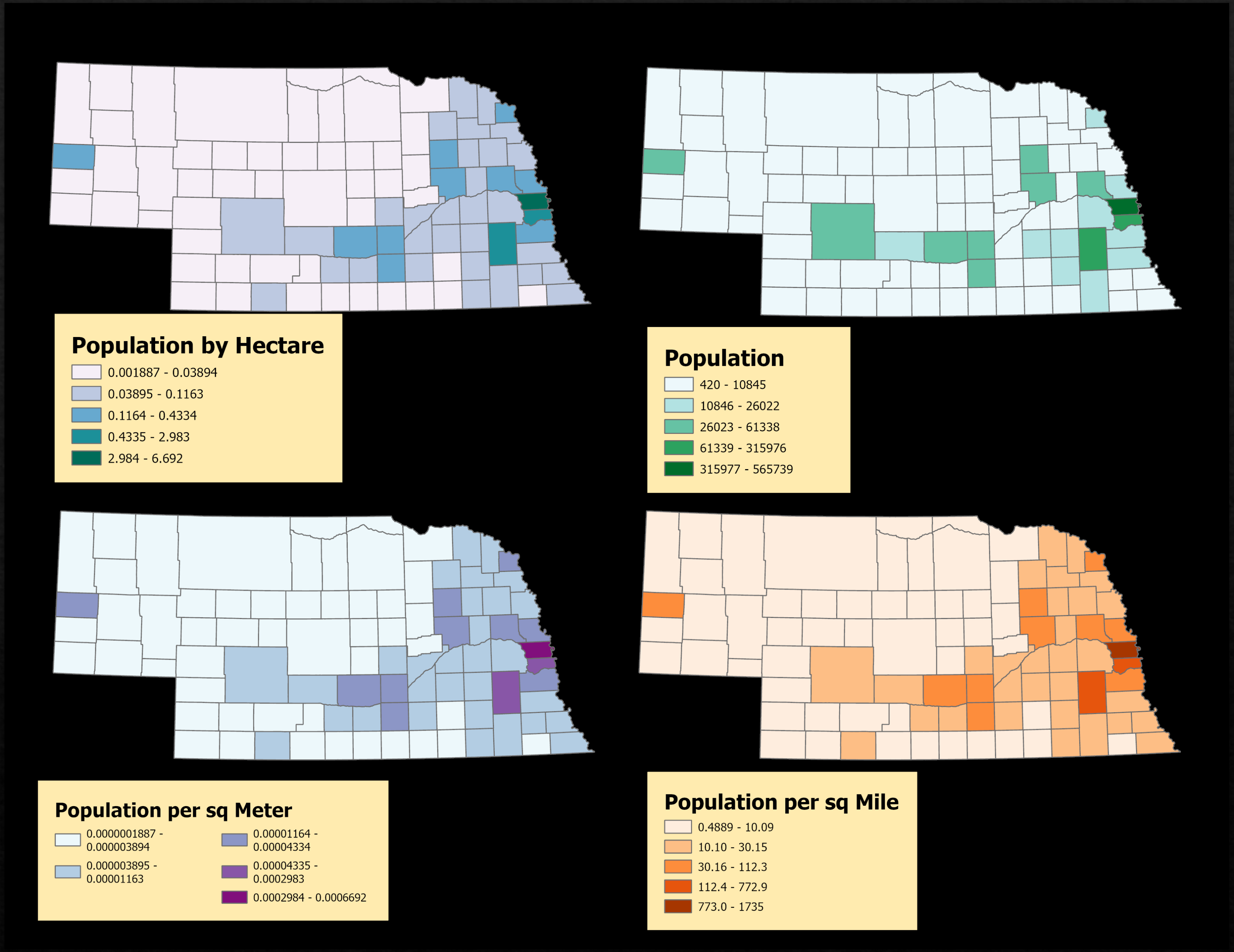
Results

Reflect

Future

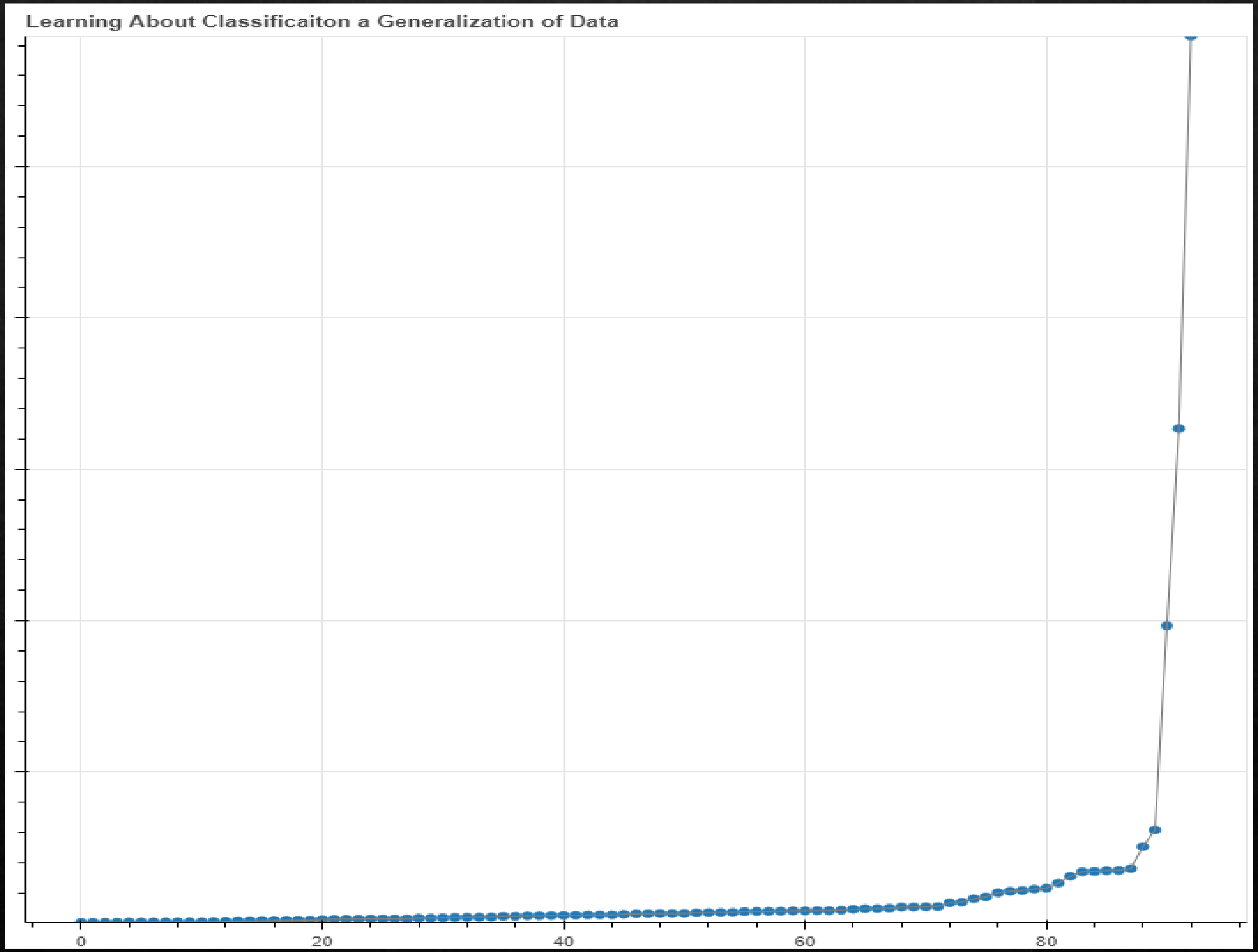
Annie (2021) Buried Treasure Map. https://playerassist.com/use-buried-treasure-map-minecraft/

# Why?

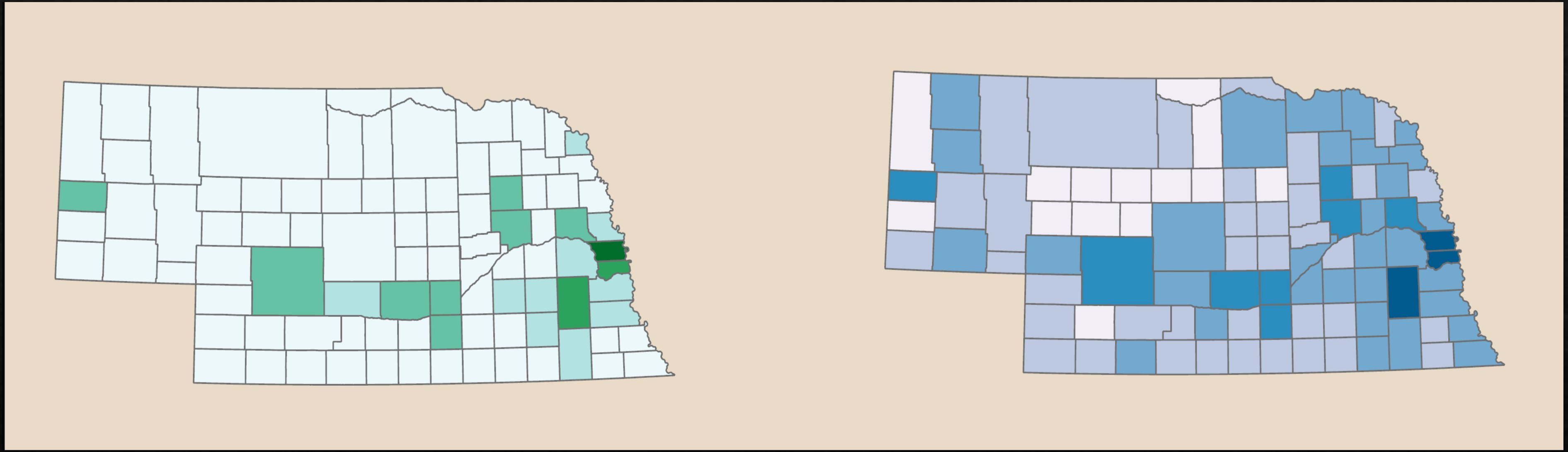

Population by Hectare
0.001887 - 0.03894
0.03895 - 0.1163
0.1164 - 0.4334
0.4335 - 2.983
2.984 - 6.692
Population
420 - 10845
10846 - 26022
26023 - 61338
61339 - 315976
315977 - 565739
Population per sq Meter
0.0000001887 - 0.000003894
0.000003895 - 0.00001163
0.00001164 - 0.00004334
0.00004335 - 0.0002983
0.0002984 - 0.0006692
Population per sq Mile
0.4889 - 10.09
10.10 - 30.15
30.16 - 112.3
112.4 - 772.9
773.0 - 1735

# Nebraska Population Data

- Clearly Natural Breaks ?
  - Pure Maths solution
- Or maybe Geometric Progression?
  - Pure shape of the Data



# Generalization via Classification



# The Process

- Goals
  - Understand Classification
  - Classification without a Map
  - Challenge the Rules
  - Coding is Helpful

- Method
  - Engaged Learning Modality
  - Get Away from Mapping Software
  - Data Interaction Action
  - Student Generated Product

# First Implementation

Python/Anaconda

Python 3.7.1

Anaconda 2.2.1

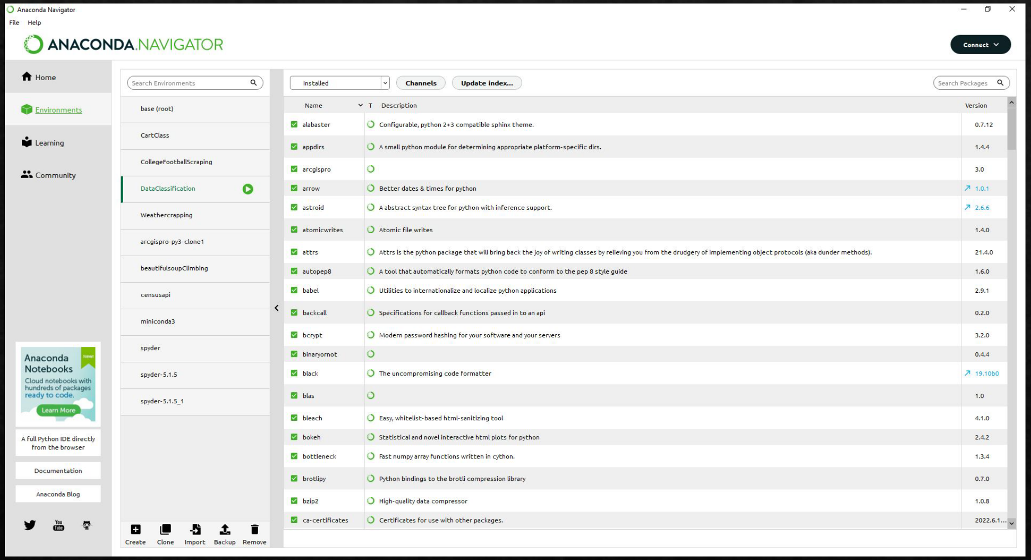
Spyder/Python console

Spyder 5.1.5

Bokeh

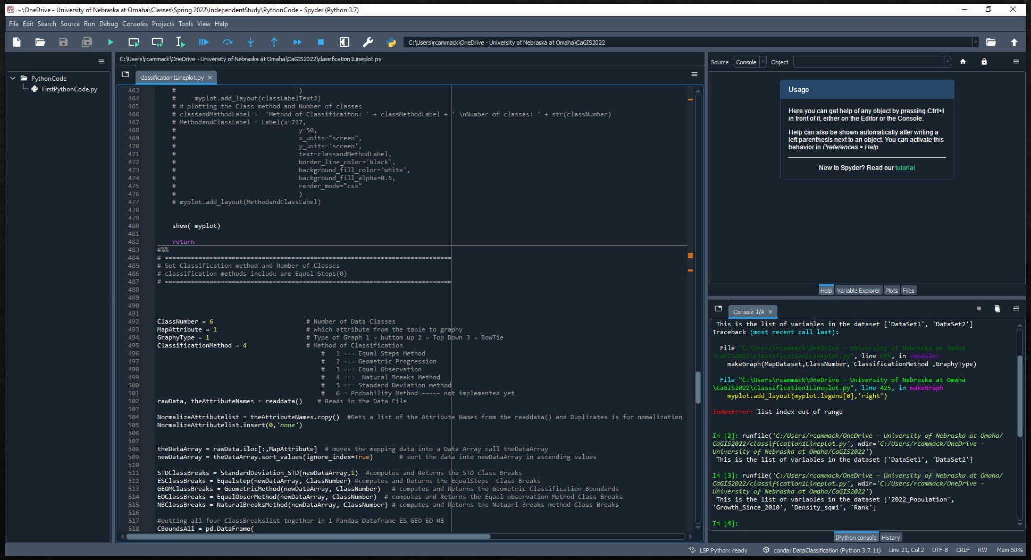
Bokeh 2.4.2

# Anaconda and Spyder



base (root)
CartClass
CollegeFootballScraping
DataClassification
Weathercrapping
arcgispro-py3-clone1
beautifulsoupClimbing
censusapi
miniconda3
spyder
spyder-5.1.5
spyder-5.1.5_1

| Name | T | Description | Version |
|---|---|---|---|
| ☑ alabaster | ○ | Configurable, python 2+3 compatible sphinx theme. | 0.7.12 |
| ☑ appdirs | ○ | A small python module for determining appropriate platform-specific dirs. | 1.4.4 |
| ☑ arcgispro | ○ | | 3.0 |
| ☑ arrow | ○ | Better dates & times for python | ↗ 1.0.1 |
| ☑ astroid | ○ | A abstract syntax tree for python with inference support. | ↗ 2.6.6 |
| ☑ atomicwrites | ○ | Atomic file writes | 1.4.0 |
| ☑ attrs | ○ | Attrs is the python package that will bring back the joy of writing classes by relieving you from the drudgery of implementing object protocols (aka dunder methods). | 21.4.0 |
| ☑ autopep8 | ○ | A tool that automatically formats python code to conform to the pep 8 style guide | 1.6.0 |
| ☑ babel | ○ | Utilities to internationalize and localize python applications | 2.9.1 |
| ☑ backcall | ○ | Specifications for callback functions passed in to an api | 0.2.0 |
| ☑ bcrypt | ○ | Modern password hashing for your software and your servers | 3.2.0 |
| ☑ binaryornot | ○ | | 0.4.4 |
| ☑ black | ○ | The uncompromising code formatter | ↗ 19.10b0 |
| ☑ blas | ○ | | 1.0 |
| ☑ bleach | ○ | Easy, whitelist-based html-sanitizing tool | 4.1.0 |
| ☑ bokeh | ○ | Statistical and novel interactive html plots for python | 2.4.2 |
| ☑ bottleneck | ○ | Fast numpy array functions written in cython. | 1.3.4 |
| ☑ brotlipy | ○ | Python bindings to the brotli compression library | 0.7.0 |
| ☑ bzip2 | ○ | High-quality data compressor | 1.0.8 |
| ☑ ca-certificates | ○ | Certificates for use with other packages. | 2022.6.1... |


~\OneDrive - University of Nebraska at Omaha\Classes\Spring 2022\IndependentStudy\PythonCode - Spyder (Python 3.7)
File
Edit
Search
Source
Run
Debug
Consoles
Projects
Tools
View
Help
C:\Users\rcammack\OneDrive - University of Nebraska at Omaha\CaGIS2022
C:\Users\rcammack\OneDrive - University of Nebraska at Omaha\CaGIS2022\classification1Lineplot.py
classification1Lineplot.py
PythonCode
FirstPythonCode.py
Source
Console
Object
Help
Variable Explorer
Plots
Files
Console 1/A
IPython console
History
LSP Python: ready
conda: DataClassification (Python 3.7.11)
Line 21, Col 2
UTF-8
CRLF
RW
Mem 50%

```
    #                               )
    #    myplot.add_layout(classLabelText2)
    # # plotting the Class method and Number of classes
    # classandMethodLabel =  'Method of Classificaiton: ' + classMethodLabel + ' \nNumber of classes: ' + str(classNumber)
    # MethodandClassLabel = Label(x=717,
    #                               y=50,
    #                               x_units="screen",
    #                               y_units='screen',
    #                               text=classandMethodLabel,
    #                               border_line_color='black',
    #                               background_fill_color='white',
    #                               background_fill_alpha=0.5,
    #                               render_mode="css"
    #                               )
    # myplot.add_layout(MethodandClassLabel)


    show( myplot)

    return
#%%
# =============================================================================
# Set Classification method and Number of Classes
# classification methods include are Equal Steps(0)
# =============================================================================




ClassNumber = 6                         # Number of Data Classes
MapAttribute = 1                        # which attribute from the table to graphy
GraphyType = 1                          # Type of Graph 1 = buttom up 2 = Top Down 3 = BowTie
ClassificationMethod = 4                # Method of Classification
                                            #   1 === Equal Steps Method
                                            #   2 === Geometric Progression
                                            #   3 === Equal Observation
                                            #   4 ===  Natural Breaks Method
                                            #   5 === Standard Deviation method
                                            #   6 = Probability Method ----- not implemented yet
rawData, theAttributeNames = readdata()    # Reads in the Data File

NormalizeAttributelist = theAttributeNames.copy()  #Gets a list of the Attribute Names from the readdata() and Duplicates is for nomalization
NormalizeAttributelist.insert(0,'none')


theDataArray = rawData.iloc[:,MapAttribute]  # moves the mapping data into a Data Array call theDataArray
newDataArray = theDataArray.sort_values(ignore_index=True)       # sort the data into newDataArray in ascending values

STDClassBreaks = StandardDeviation_STD(newDataArray,1)  #computes and Returns the STD class Breaks
ESClassBreaks = Equalstep(newDataArray, ClassNumber) #computes and Returns the EqualSteps  Class Breaks
GEOMClassBreaks = GeometricMethod(newDataArray, ClassNumber)   # computes and Returns the Geometric Classification Boundards
EOClassBreaks = EqualObserMethod(newDataArray, ClassNumber)  # computes and Returns the Eqaul observation Method Class Breaks
NBClassBreaks = NaturalBreaksMethod(newDataArray, ClassNumber) # computes and Returns the Natuarl Breaks method Class Breaks

#putting all four ClassBreakslist together in 1 Pandas Dataframe ES GEO EO NB
CBoundsAll = pd.DataFrame(
```

Usage

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

New to Spyder? Read our tutorial

```
 This is the list of variables in the dataset ['DataSet1', 'DataSet2']
Traceback (most recent call last):

  File "C:\Users\rcammack\OneDrive - University of Nebraska at Omaha
\CaGIS2022\classification1Lineplot.py", line 695, in <module>
    makeGraph(MapDataset,ClassNumber, ClassificationMethod ,GraphyType)

  File "C:\Users\rcammack\OneDrive - University of Nebraska at Omaha
\CaGIS2022\classification1Lineplot.py", line 425, in makeGraph
    myplot.add_layout(myplot.legend[0],'right')

IndexError: list index out of range


In [2]: runfile('C:/Users/rcammack/OneDrive - University of Nebraska at Omaha/
CaGIS2022/classification1Lineplot.py', wdir='C:/Users/rcammack/OneDrive -
University of Nebraska at Omaha/CaGIS2022')
 This is the list of variables in the dataset ['DataSet1', 'DataSet2']

In [3]: runfile('C:/Users/rcammack/OneDrive - University of Nebraska at Omaha/
CaGIS2022/classification1Lineplot.py', wdir='C:/Users/rcammack/OneDrive -
University of Nebraska at Omaha/CaGIS2022')
 This is the list of variables in the dataset ['2022_Population',
'Growth_Since_2010', 'Density_sqmi', 'Rank']

In [4]:
```

# Implementation Process

- Build a Python environment
  - Used Anaconda to handle this environment
  - Export the environment to other Computers

- Key Packages used
  - Numpy
  - Pandas*
  - Jenkspy

Pandas was used --- not the best solution

# What is Implemented

## Number of Classes

This program will do any number of classes

Next Phase will restricted this to 4 to 12 classes

## Classification Methods

Equal Steps

Geometric Progression

Equal Observation*

Standard Deviation

Natural Breaks

Next phase will have a probability classification method

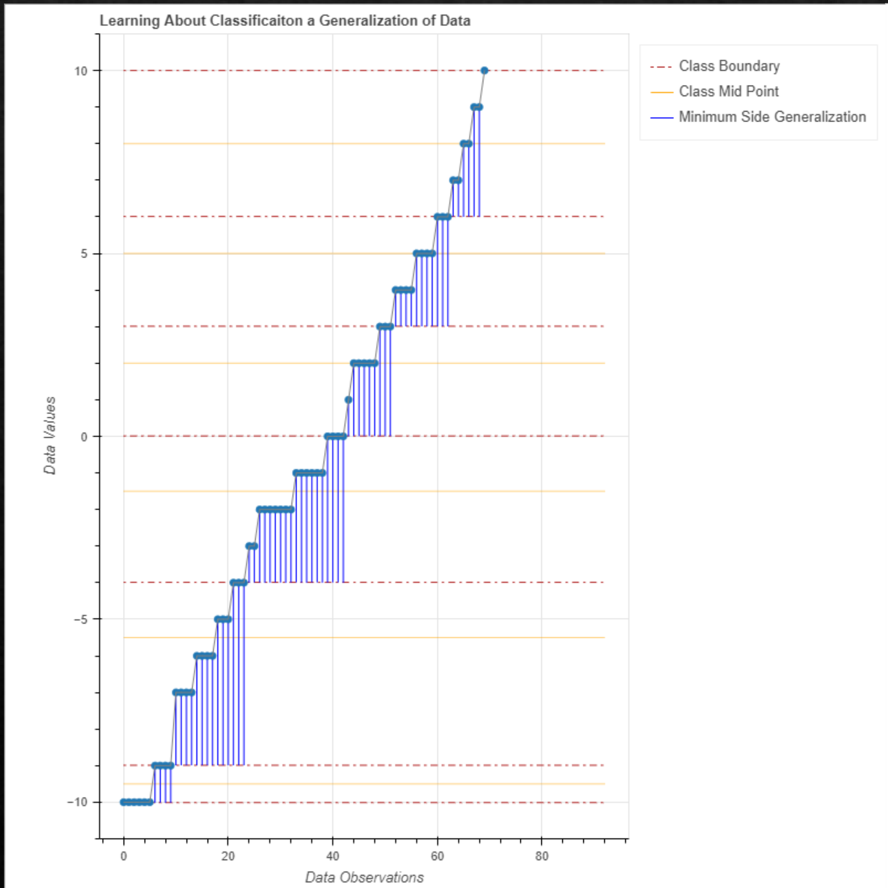
## Graph Forms

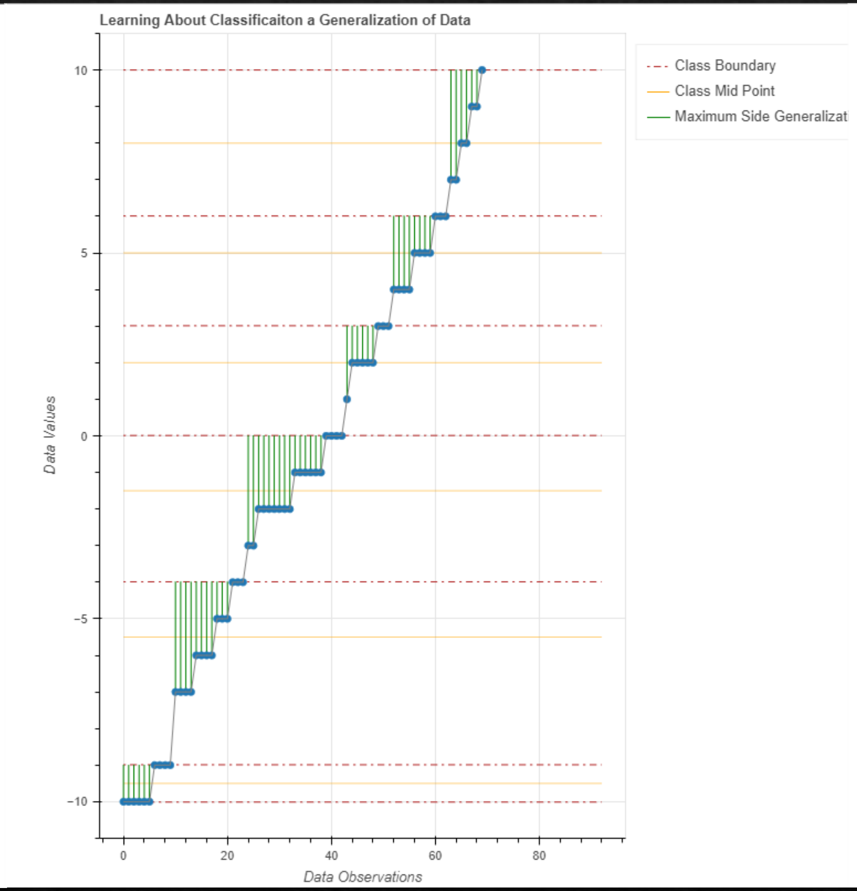
Top Down Graph

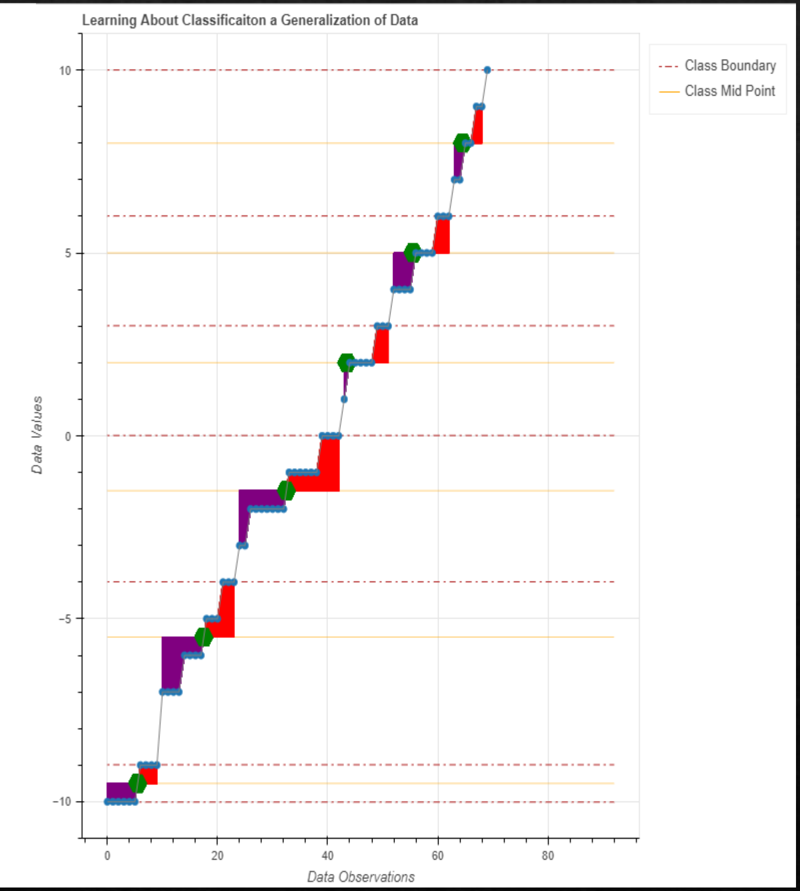
Bottom Up  Graph

Bowtie Graph

# Graph Form Examples







# Student Task

- Get software, data, Python Environment and Instructions
  - Distributed by Canvas to the class
- Setup Python Environment
  - Terminal task:  Conda Create ….
- Start the new environment
- Open program

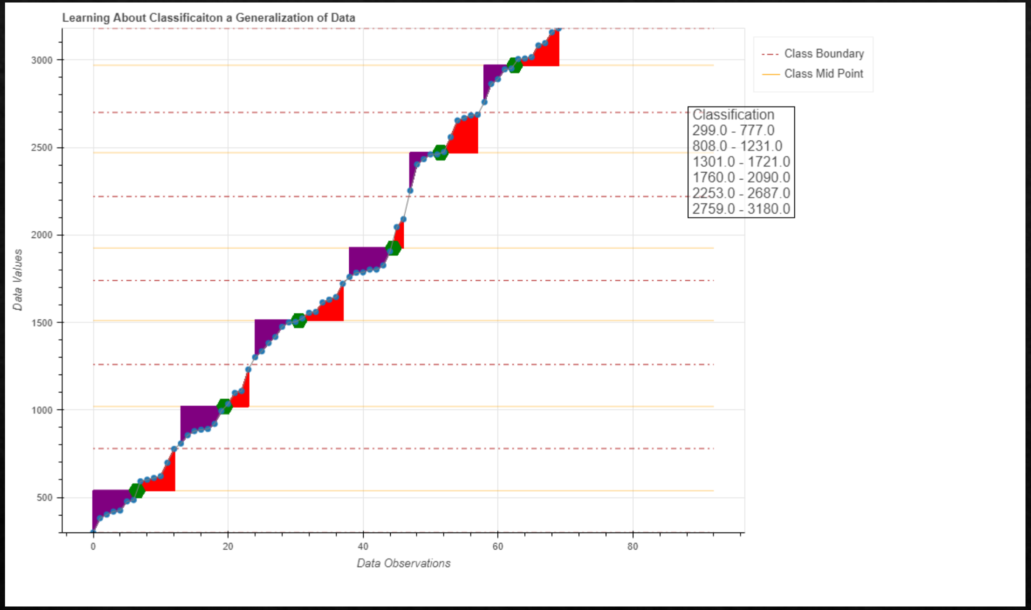
- Program option
  - Unlimited if you want
  - Data used
  - Number of classes
  - Method of classification
  - Graph type
  - Graph design*****

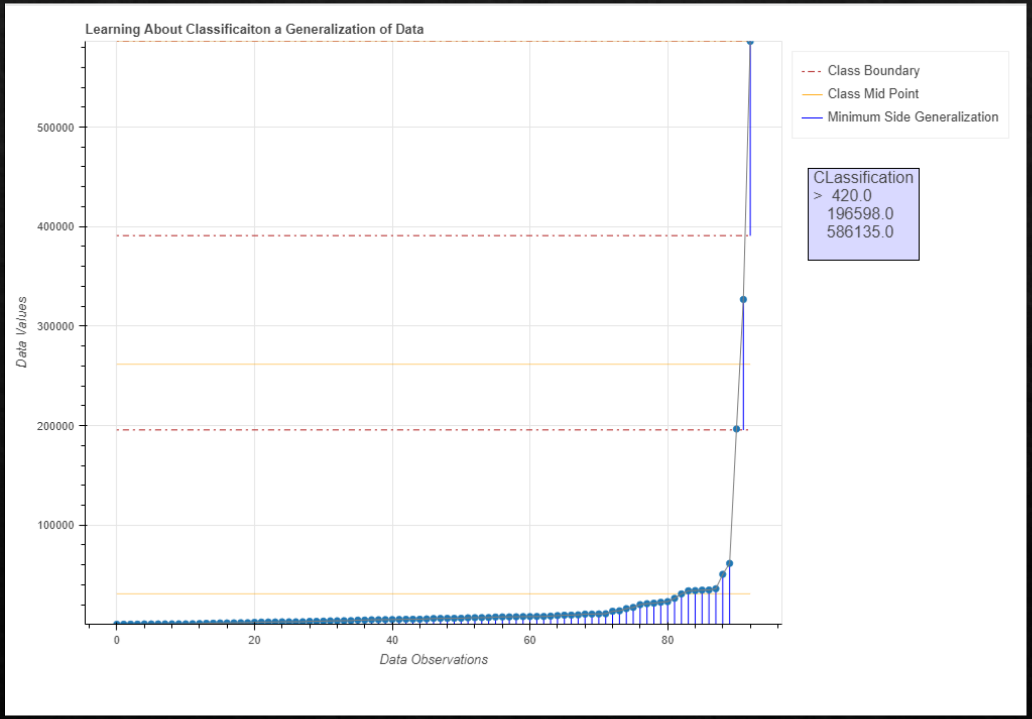
It should be noted: I am not a great programmer so the code is messy

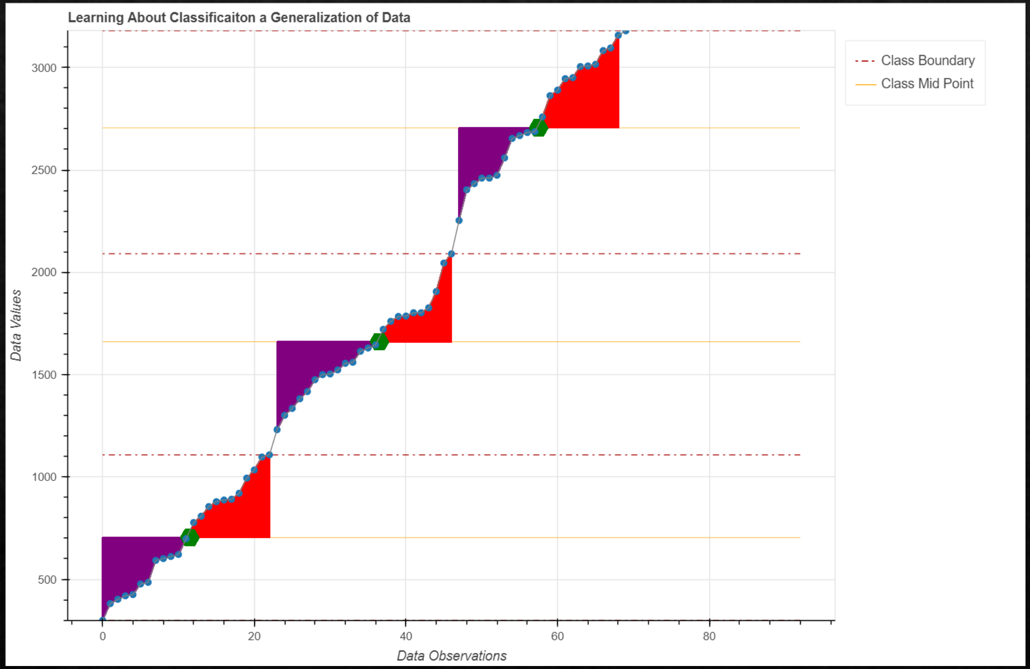
# Results

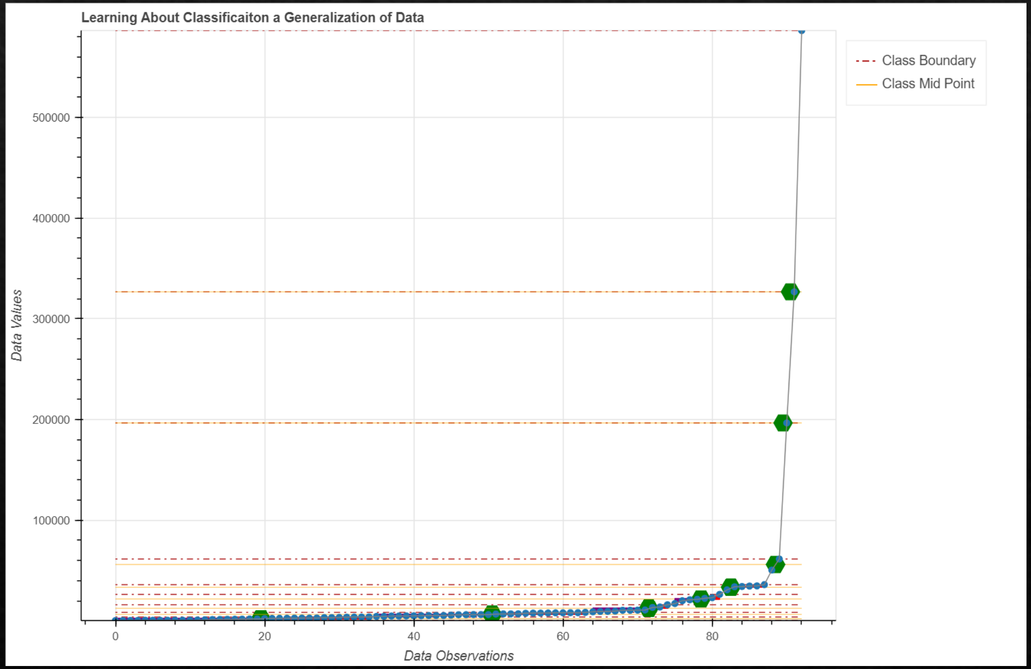
- Student had to create the best classification for two dataset and upload their graphs to a canvas assignment.

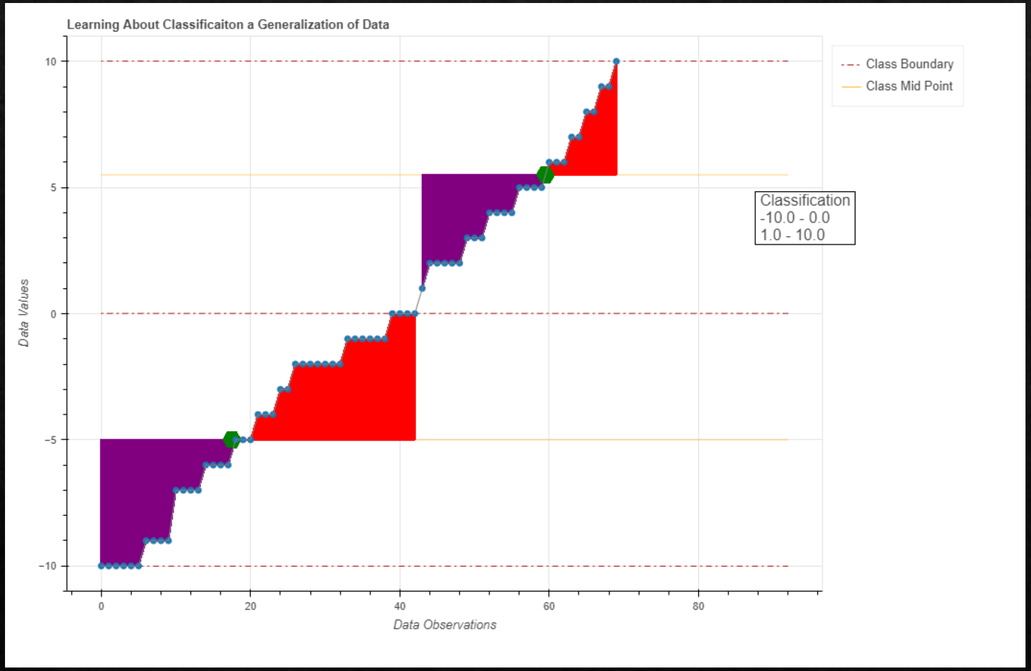
- Observation of Students
  - Looking at the code is scary
  - Needed all the user inputs in one part of the code.
  - Having them change the Bokeh graphic was an option but students stayed with my ugly color choices
  - Students needed more guided interpretation input


Learning About Classificaiton a Generalization of Data
Class Boundary
Class Mid Point
Classification
299.0 - 777.0
808.0 - 1231.0
1301.0 - 1721.0
1760.0 - 2090.0
2253.0 - 2687.0
2759.0 - 3180.0
Data Values
3000
2500
2000
1500
1000
500
0
20
40
60
80
Data Observations

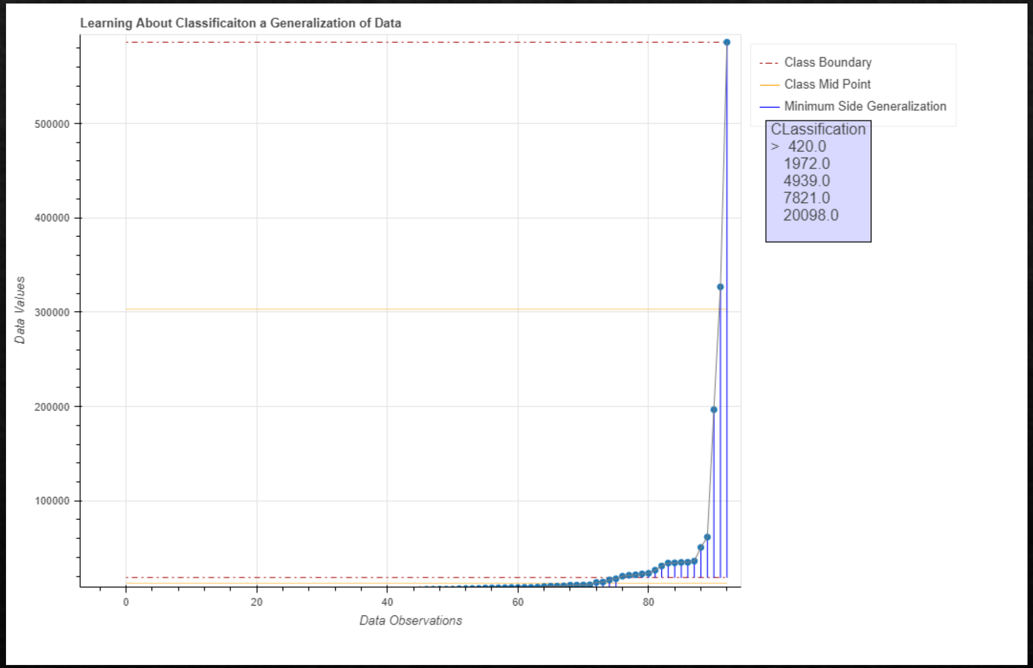

Learning About Classificaiton a Generalization of Data
Class Boundary
Class Mid Point
Minimum Side Generalization
CLassification
> 420.0
196598.0
586135.0
Data Values
500000
400000
300000
200000
100000
0
20
40
60
80
Data Observations


Learning About Classificaiton a Generalization of Data
Class Boundary
Class Mid Point
Data Values
3000
2500
2000
1500
1000
500
0
20
40
60
80
Data Observations


Learning About Classificaiton a Generalization of Data
Class Boundary
Class Mid Point
Data Values
500000
400000
300000
200000
100000
0
20
40
60
80
Data Observations



Learning About Classificaiton a Generalization of Data



Learning About Classificaiton a Generalization of Data

# Reflection

- Streamlining
  - Setup the whole activity?
  - Make it less code more UI
- Educational Efficiency ?
- Student reflection/critique (B. Plewe's idea)

# Future

- Next Semester
  - Work on streamlining it.
    - Better organized code
    - Move to full Bokeh UI
  - Fix code
    - Work on Equal Observation algorithm
    - Implement Probability classification
    - Add error trapping / stop using Pandas
  - New Graph Types?

- Add Guided Learning
  - Get the classic rules on screen
- Critiques by Students
  - Written
  - Verbal

# AUTOCARTO 2022

Thank you all for attending