Restaurant GIS Tool Report

Programming for GIS - CP 6581

Bill Huang

Dr. Gervais W Tabopda

12/05/2021

Contents

Spatial Analysis Project Focus

The goal of this project is two-fold. One focus is to exemplify and explore my ability in using python with ArcGIS. The second focus is to create a tool to help people understand where food establishments of various cultures are in Atlanta and to what degree they are present in various areas of Metro Atlanta.

Food is often central to many cultures. Atlanta is place where people from many different ethnicities end up. Food is also a shared experience in many cultures and provides a great means of exploring the culture of other peoples. This project would help to analyze where different cultures are tangible to experience in Atlanta through restaurants of each culture. Mapping out a restaurant category of interest could display where people from that culture tend to reside in Atlanta as well as where people can go to experience different cultures in our own city of Atlanta. I hope to learn where different types of restaurants tend to locate in Atlanta, which areas of Atlanta might be in a food culture deficit and which areas show strong concentrations of food culture. In the project an ArcGIS Pro Toolbox was made, so that everyone using this tool could select a specific culture of food to analyze the city through.

Data Source

The data sources used for this problem include the US Census and the Yelp Fusion API. R was used for both the US Census API and the Yelp API to collect the data necessary for this project. The US Census was used to collect the spatial boundaries of all Census Tracts within Fulton, Dekalb, and Clayton counties. These counties were chosen because the city of Atlanta boundary includes property from all these three counties. Yelp is a business review service that collects and displays a physical business's details that would be of interest to the public. They started mainly specializing in restaurant reviews. Census Tract boundaries were needed to poll the Yelp Fusion API. If larger area geometries were used to request restaurant data from the Yelp API, the Yelp API would have errored because the returned restaurant number would be to high per geometry request. Therefore, smaller geometries like the Census Tract were needed to not hit a maximum threshold of returned restaurant hits. Each Census Tract was geometry was then fed into a separate business search request to the Yelp Fusion API to get data on all restaurants within every census tract within Fulton, Dekalb, and Clayton counties. The data was then cleaned by removing duplicates, unnesting data structures in the returned Yelp API data, and removing restaurants with no geographic data attached. 5491 restaurants were returned from the Yelp API after cleaning. Figure 1 displays the entire dataset that was worked with for this project. It is important to note that the Yelp Fusion API is mainly crowd and self-reported data from restaurant frequenters and restaurant owners. This causes the data to not be fully unbiased and representative of all possible restaurants in the study area. Some restaurants may be missing, and some restaurants may have better reviews than most people might agree with. The API requesting and data cleaning were done in the R programming language and then the results were ported exported as a shapefile for this project's use in ArcGIS.

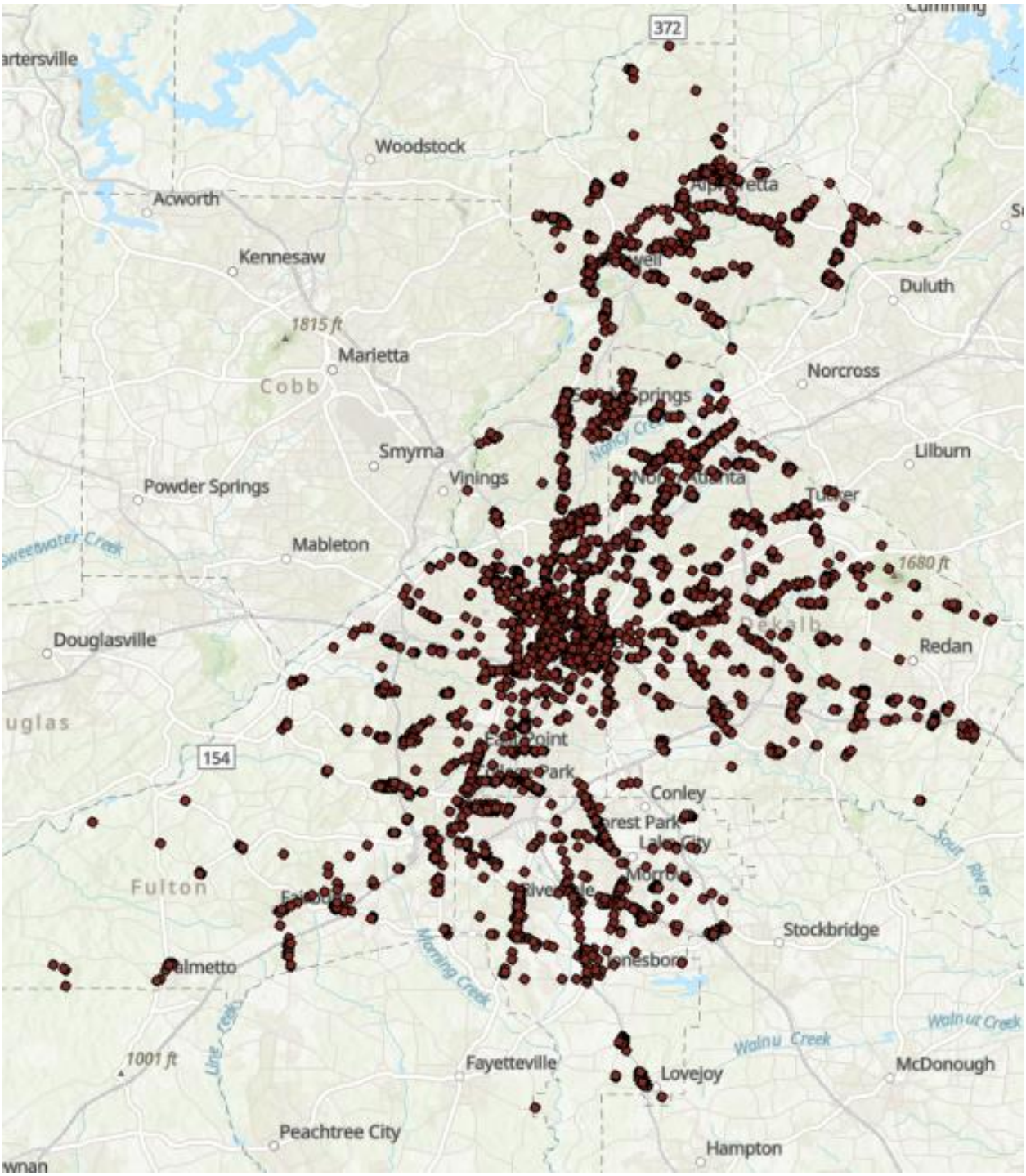

Figure 1: Cleaned restaurant data within Fulton, Dekalb, Clayton Counties

Analysis Techniques

The techniques used for this project include creating an ArcGIS Pro Toolbox along with traditional techniques like SQL querying of a shapefile attribute table and implementing graduated color symbology.

The process that is needed to pick out a specific type of food category from all of the original data points is a query of some sort. ArcGIS provides such query functionality through integration with SQL processes. The specific column of interest is the “categrs” column in the shapefile. The categrs column displays all the business tags that the restaurant might fit into based upon crowd-sourced input form the Yelp app. Figure 2 displays this column in the shapefile’s attribute table. For example, let us say that we are interested in the East Asian category of food. One would look for all East Asian text lines that appear in the categrs column and keep all those with the East Asian text string within that categrs field. However, East Asian will never appear in the categrs column because Yelp does not have an East Asian food tag. It does have Chinese, Japanese, etc. Because of this, a list had to be made for every general category of food. In the code, this took the form of a dictionary. So the actual process that was required was for every Yelp category food within the East Asian category I wanted to analyze, an SQL query had to be performed to keep all the records.

categrs
Barre Classes, Women's Clothing, Cardio Classes
Women's Clothing, Accessories, Pop-up Shops
Barre Classes, Women's Clothing, Cardio Classes
Barre Classes, Women's Clothing, Cardio Classes
Barbers, Cosmetics & Beauty Supply
Barre Classes, Women's Clothing, Cardio Classes
Barre Classes, Women's Clothing, Cardio Classes
Festivals, Pop-up Shops, Flowers & Gifts
Sports Wear, Outdoor Gear, Hunting & Fishing Supplies
Barre Classes, Women's Clothing, Cardio Classes
Festivals, Pop-up Shops
Sports Wear, Outdoor Gear, Hunting & Fishing Supplies
Arts & Entertainment, Pop-up Shops
Barre Classes, Women's Clothing, Cardio Classes
Barbers, Cosmetics & Beauty Supply
Barre Classes, Women's Clothing, Cardio Classes
International Grocery, Seafood Markets, Wholesale Stores
Sports Wear, Outdoor Gear, Hunting & Fishing Supplies
Candle Stores, Pop-up Shops
Art Museums, Pop-up Shops
Barre Classes, Women's Clothing, Cardio Classes
Nurseries & Gardening, Livestock Feed & Supply, Pet Stores
Florists, Chocolatiers & Shops, Juice Bars & Smoothies

Figure 2: Categories column of the shapefile

The second technique required to visualize the results was implementing a graduated color scheme symbology. This technique helps to visually shows that of a category of food such as East Asian, which East Asian restaurants are highly rated or popular in various parts of the city. Through this visualization, one could see if the most popular or highly rated East Asian restaurants are in the center of Atlanta, evenly distributed across the Atlanta region, or if they are in a different part of Atlanta. The numeric range of the rating on Yelp is 1-5 while the review count value range is between zero and a very large number. Because the rating column is tiered, it is beneficial to set an equal interval for the graduated colors as opposed to a distribution related coloring interval, so that each different number rating will have a different color.

Results

There are 32 food categories built into this tool which cover hundred of Yelp Restaurant tags. Each food type can be polled to display either popularity or rating which means that there are 64 possible maps that could be generated using the current version of this tool. For brevity's sake, only two of these maps will be shown. The first chosen result example is African food by Popularity, and the second result example is African food by Rating.

For the first example, the input parameters to the tool are shown in Figure 3. An input feature class must be selected for Input Features. In this case, Input Features is the data of Atlanta that was prepared as detailed in the previous data section of this report. This tool could also be used for other cities provided that you have the restaurant data for other cities, so this tool can display Food Type by Popularity for Philadelphia instead of Atlanta if one had the input data for Philadelphia(this can be pulled from Yelp API instead of Atlanta).

Figure 3: Input Parameters for African Food by Popularity

Figure 4 below displays the layer that is outputted when the tool is run with the parameters in Figure 3. It shows where the African Restaurants are in Atlanta, and it shows how popular they are. The takeaways from the map will vary for every map generated, but, for this example, it shows that African Restaurants are somewhat evenly distributed across all the Counties overlapping the City of Atlanta boundaries. There might be less African restaurants in the Westside and Northside of Fulton County

though. It also seems that there's a concentration of African restaurants in Decatur for some reason. These concentrations of lack of thereof could indicate other things about the prescence of African culture in Decatur and/or the West and Northside of Fulton. Looking at the graduated coloration of the restaurant points, it seems like the most popular restaurants are located closer to the center of Atlanta despite there being a large concentration of African restaurants in Decatur. The popularity is determined by review count on Yelp. There is an assumption that the more reviews a restaurant has on Yelp, the more popular it is. This could be a bad assumption sometimes though, as a restaurant may be popular, but frequenters may not be placing reviews on Yelp. However, it is not unreasonable that this assumption will generally hold true

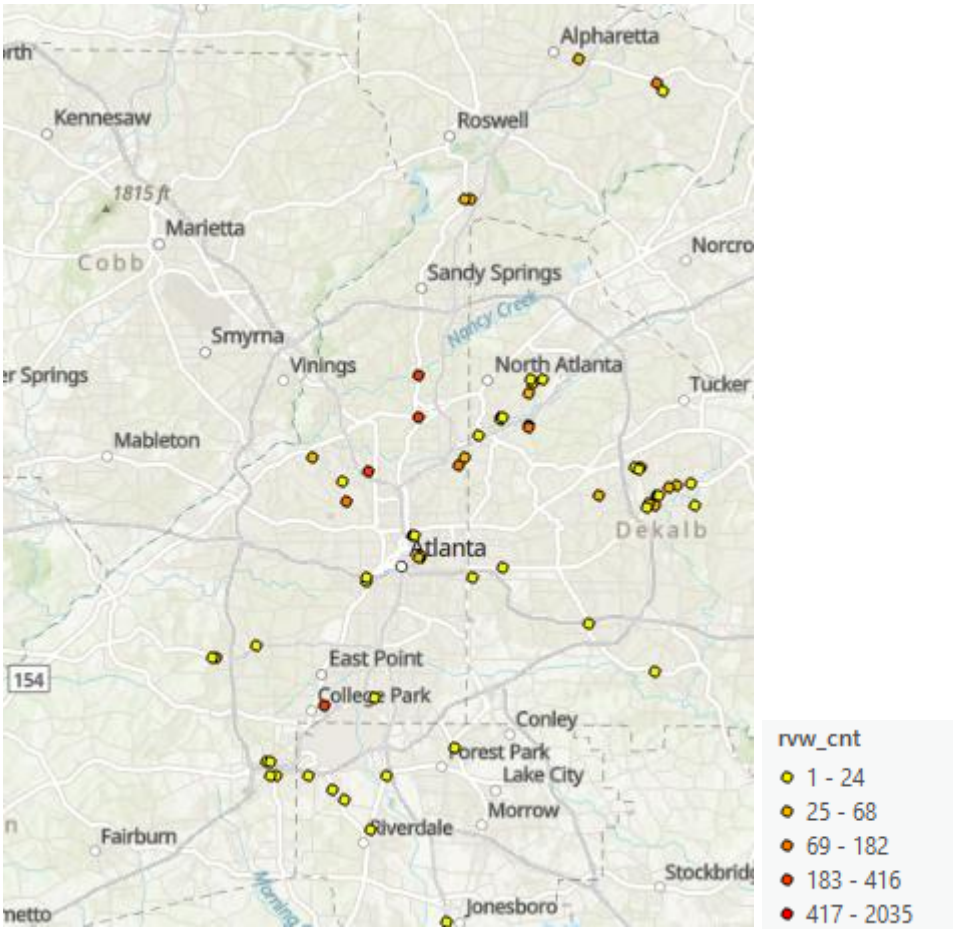

Figure 4: African Food in Atlanta by Populairty Map Output from Custom Coded Tool

Figure 5 displays the user inputted paramters for African Food by Rating. Figure 6 shows the outputted results by the tool. One can observe from this map that all the African restaurants are somewhat highly rated, but one closer to the center of Atlanta have worse review ratings for some reason. Additionally, if one is interested in any given restaurant on the map, the user could click on the data point to show the restaurant name, address, and yelp page amongst other field information.

Figure 5: Input Parameters for African Food by Rating

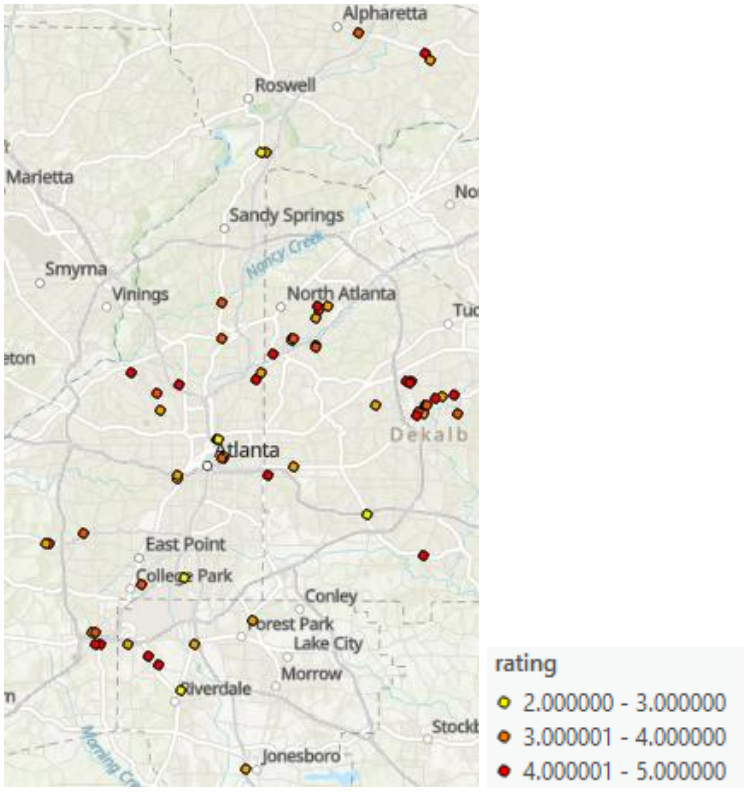

Figure 6: African Food in Atlanta by Rating Map Output from Custom Coded Tool

This tool actually generated a shapefile with a custom name based on the parameters you set for the tool when using it. Figure 7 show that these outputted layers are stored automatically in the project geodatabase with custom parameter names for easy identification.

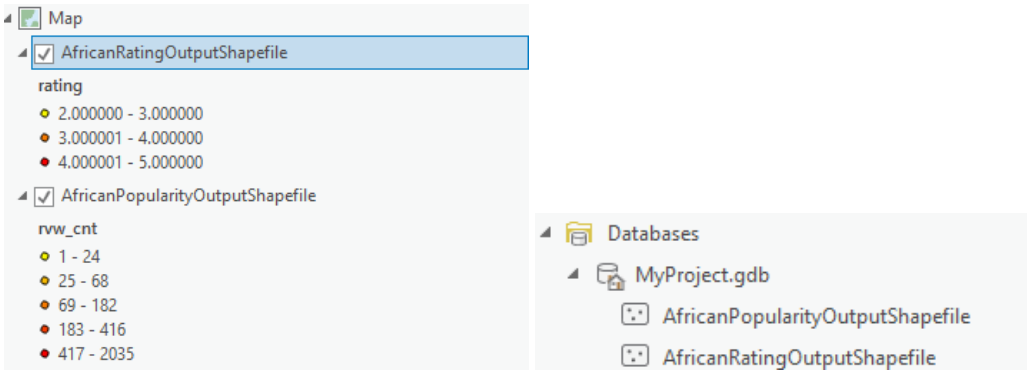

Figure 7: Resulting Tool Output Maps

User Interface

This tool requires that you add a custom toolbox. From that point, you can select the .pyt file which is a python script I made that contains the functionality of this tool. The tool within the toolbox made is called “Generate Restaurant Map by Food Category” as shown in Figure 8. The custom .pyt python code is detailed in the Code section farther down in this report.

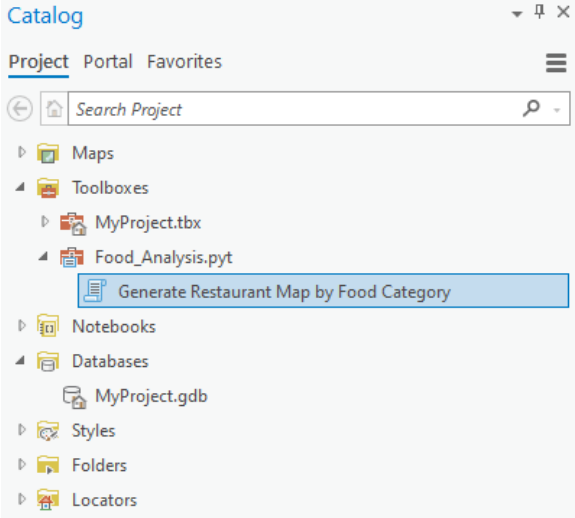

Figure 8: Custom Toolbox Interface

When you run the custom coded toolbox, you will see the graphic in Geoprocessing pane pop up in Figure 9. Figure 10 displays some of the custom coded restaurant food categories you can analyze, and Figure 11 displays the two topics you can choose to analyze within the food category selected.

Figure 9: Custom Tool Interface

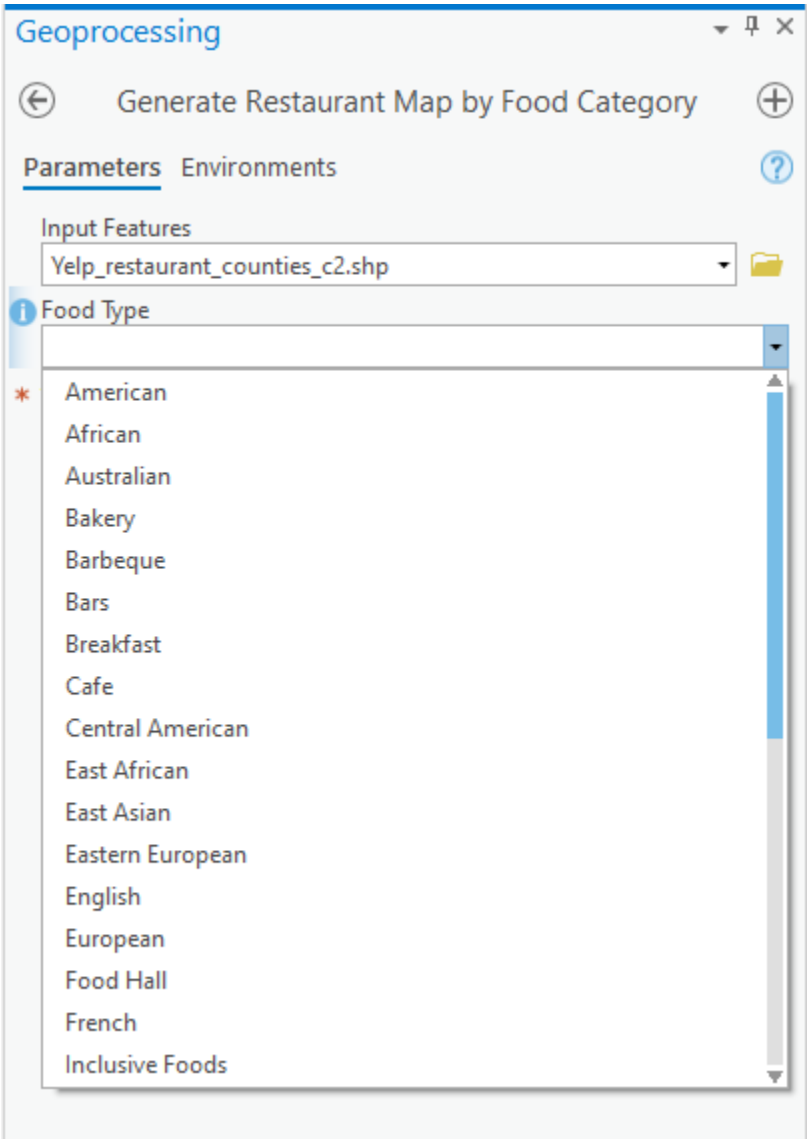

Figure 10: Custom Tool Food Type Selection Options

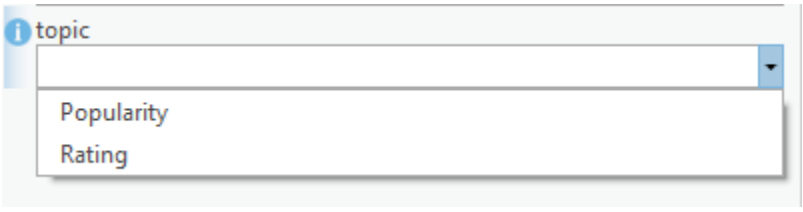

Figure 10: Custom Tool Topic Selection Options

Code Documentation

The following code has a couple sections. The class Toolbox code defines the toolbox. The class ViewFoodType code defines one custom tool within the Toolbox. I only created one tool, but you could create multiple tools within this tool box. The parameters(the original restaurant data, the food category, and the topic) are defined in the getParameterInfo section of the code. They have been customized to only allow certain inputs into this tool. The final section of importance is the execute section. This execute section shows the code that provides the analysis functionality of the tool while the other non-execute sections are to get the toolbox to work in ArcGIS. The code provided below is commented to provide explanations.

```
#import arcpy package
import arcpy

#define toolbox
class Toolbox(object):
    def __init__(self):
        self.label =  "Restaurant toolbox"
        self.alias  = "restaurant"

        # List of tool classes associated with this toolbox
        self.tools = [ViewFoodType]

#define restaurant tool
class ViewFoodType(object):
    def __init__(self):
        self.label       = "Generate Restaurant Map by Food Category" #tool name
        self.description = "View where restaurants are located in the Atlanta Counties (Fulton, Dekalb,
Clayton) by popularity(# of reviews) or rating(scale 1-5) according to Yelp Data" #tool description

    def getParameterInfo(self):
        #Define parameter definitions

        # Input Features(the restaurant original cleaned data) parameter
        in_features = arcpy.Parameter(
            displayName="Input Features", #name of parameter on ArcGIS UI
            name="in_features",
            datatype="GPFeatureLayer", #inputted data type must be a feature layer
            parameterType="Required", #not optional that this parameter is provided
```

```
            direction="Input")

        # Food Type parameter
        food_type = arcpy.Parameter(
            displayName="Food Type", #name of parameter on ArcGIS UI
            name="food_type",
            datatype="GPString", #inputted data type must be text
            parameterType="Required", #not optional that this parameter is provided
            direction="Input")

        #sets the food type parameter property to be a dropdown list in the ArcGIS UI of the custom
defined food groups
        food_type.filter.type = "ValueList"
        food_type.filter.list =
["American","African","Australian","Bakery","Barbeque","Bars","Breakfast","Cafe","Central American",
                                "East African","East Asian","Eastern European","English","European","Food
Hall","French","Inclusive Foods",
                                "Indian","Japanese","Korean","Mediterranean","Mexican","Middle
Eastern","North African","Northern European",
                                "Other","Seafood","Soup","South African","South American""Southeast
Asian","Southern Food","Spanish","West African"]

        # Popularity(Review Count) or Quality parameter
        topic = arcpy.Parameter(
            displayName="topic", #name of parameter on ArcGIS UI
            name="topic",
            datatype="GPString", #inputted data type must be text
            parameterType="Required", #not optional that this parameter is provided
            direction="Input")

        #sets the food type parameter property to be a dropdown list in the ArcGIS UI of the two analysis
options within a food group
        topic.filter.type = "ValueList"
        topic.filter.list = ["Popularity","Rating"]

        parameters = [in_features, food_type, topic] #setting parameters above to be returned globally
when the GetParametersInfo function is called

        return parameters

    #checks for a License
    def isLicensed(self):
        return True

    #updates any messages which is none right now
```

```
    def updateMessages(self, parameters):
        return

    #cutom tool code executed to perform food restaurant analysis
    def execute(self, parameters, messages):

        #gets parameters defined above from user input
        inFeatures  = parameters[0].valueasText
        food_type   = parameters[1].valueAsText
        topic = parameters[2].valueAsText

        #sets custom title for output map/layer generated from running the tool
        inFeatures2 = food_type+topic+"OutputShapefile"

        #allows tool to overwrite outputted maps with the same parameters
        arcpy.env.overwriteOutput = True

        #sets UI names for topic options to attribute field data headers
        topic_dict = {"Popularity":"rvw_cnt","Rating":"rating"}
        field_name = topic_dict[topic]

        #defines what yelp tags are searched for given a user's selection in the ArcGIS UI
        food_dict = {"American":["American", "Asian Fusion", "Barbeque",
"Breakfast","Brunch","Pancakes","Buffets","Burgers", "Cajun", "Creole", "Cheesesteaks", "Chicken Wings",
"Chicken Shop","Comfort Food","Delis","Diners","Fast Food", "Gastropubs","Hawaiian","Hot Dogs","Pizza",
"Salad","Sandwiches","Soul Food", "Southern", "Steakhouses","Soup","Tex","Waffles","Wraps",
"Poutineries"],
                     "African":["African", "Senegalese", "Eritrean","Ethiopian","Mauritius", "Somali",
"Moroccan", "South African"],
                     "Australian":["Australian"],
                     "Bakery":["Bakeries"],
                     "Barbeque":["Barbeque"],
                     "Bars":["Bar"],
                     "Breakfast":["Breakfast","Brunch","Pancakes","Waffles"],
                     "Cafe":["Cafes","Themed Cafes", "Coffee"],
                     "Central American":["Honduran","Latin American","Salvadoran","Nicaraguan"],
                     "East African":["Eritrean","Ethiopian","Mauritius", "Somali"],
                     "East Asian":["Asian Fusion", "Chinese","Cantonese","Dim
Sum","Hainan","Shanghainese","Szechuan", "Hong Kong Style Cafe","Hot
Pot","Japanese","Conveyor","Belt","Sushi","Izakaya","Japanese Curry","Ramen","Teppanyaki","Korean","Sushi
Bars","Taiwanese", "Mongolian","Noodles","Pan Asia"],
                     "Eastern European":["Austrian","Bulgarian","Czech",
"German","Hungarian","Polish","Slovakian","Ukrainian"],
                     "English":["British","Chips","Irish","Scottish"],
```

```
                    "European":["Austrian", "Basque",
"Belgian","Brasseries","British","Bulgarian","Catalan","Creperies","Czech", "Chips", "French",
"Greek","Hungarian","Iberian","Irish","Italian","Calabrian","Sardinian","Sicilian","Tuscan", "Modern
European","Polish", "Portuguese","Russian","Scandinavian","Scottish","Slovakian","Spanish", "Tapas
Bars","Tapas","Small Plates", "Ukrainian","Uzbek", "German","Fondue"],
                    "Food Hall":["Food Court","Food Stands", "Up Restaurants"],
                    "French":["Brasseries", "Creperies", "French"],
                    "Inclusive Foods":["Gluten-Free","Kosher","Halal","Vegan","Vegetarian"],
                    "Indian":["Bangladeshi","Indian","Pakistani","Himalayan","Nepalese","Sri Lankan"],
                    "Japanese":["Japanese","Conveyor","Belt","Sushi","Izakaya","Japanese
Curry","Ramen","Teppanyaki","Sushi Bars"],
                    "Korean":["Korean"],
                    "Mediterranean":["Falafel", "Mediterranean","Egyptian","Kebab", "Italian"],
                    "Mexican":["Mexican","Tacos","New Mexican Cuisine","Mex"],
                    "Middle Eastern":["Middle Eastern","Afghan", "Arabian", "Armenian",
"Halal","Kosher","Falafel","Egyptian","Lebanese","Turkish","Georgian","Kebab","Persian","Iranian","Syrian"
, "Uzbek"],
                    "North African":["Moroccan"],
                    "Northern European":["Russian","Scandinavian"],
                    "Other":["Cafeteria","Dinner Theater", "Reunion","Game Meat","Live","Raw Food","Supper
Clubs"],
                    "Seafood":["Fish","Seafood"],
                    "Soup":["Soup","Noodles"],
                    "South African":["South African"],
                    "South American":["Argentine","Brazilian","Latin
American","Colombian","Venezuelan","Peruvian"],
                    "Southeast Asian":["Asian Fusion", "Burmese","Cambodian", "Filipino", "Indonesian"
,"Laotian","Polynesian", "Singaporean","Thai","Vietnamese", "Malaysian","Noodles","Pan Asia"],
                    "Southern Food":["Cajun", "Creole","Soul Food",
"Southern","Waffles","Pancakes","Brunch"],
                    "Spanish":["Iberian","Basque","Catalan", "Tapas","Tapas Bars","Small Plates",
"Spanish","Portuguese"],
                    "West African":["Senegalese"]}
        food_tags = food_dict[food_type]

        #Sets up the SQL string query syntax that arcpy uses
        delimfield = arcpy.AddFieldDelimiters(inFeatures,"categrs")
        sql_exp = []
        #loops through every yelp food tag in the above list of a selected element fo the dictionary to
search of all yelp tags in a food type category
        for tag in food_tags:
            delimtag = "'%" + tag + "%'"
            sql_exp_part = delimfield + "NOT LIKE" + delimtag
            sql_exp.append(sql_exp_part)
        sql_exp = " And ".join(sql_exp)
```

```
        #Creates a copy of the original data and deletes all restaurants that are not within the food type
category selected by the user
        arcpy.CopyFeatures_management(inFeatures,inFeatures2)
        with arcpy.da.UpdateCursor(inFeatures2,["categrs"],sql_exp) as cursor:
            for row in cursor:
                cursor.deleteRow()

        #Defines the graduated color scheme for the newly generated layer
        p = arcpy.mp.ArcGISProject("CURRENT") #select map
        m = p.listMaps("Map")[0]
        lyr_loc = r'C:\Users\billh\Desktop\CP6581\Week11\MyProject\MyProject.gdb\{0}'.format(inFeatures2)
        messages.addMessage("{0} is location".format(lyr_loc))
        m.addDataFromPath(lyr_loc)
        lyr = m.listLayers(inFeatures2)[0] #select newly generated layer
        if lyr.isFeatureLayer: #checks to see if layer can have symbology edited
            sym = lyr.symbology
        if hasattr(sym, 'renderer'): #checks to see if layer can have graduated color option
            if sym.renderer.type == 'SimpleRenderer':
                messages.addMessage("{0} is location".format(field_name))
                sym.updateRenderer('GraduatedColorsRenderer') #set symbology to graduated colors
                if field_name == "rating": #if the user selected rating as the topic to analyze then
choose equal interval scheme
                    sym.renderer.classificationField = field_name
                    sym.renderer.classificationMethod = "EqualInterval"
                    sym.renderer.breakCount = 4
                else: #if user selected popularity as the topci to analyze then choose jenks breaks method
scheme
                    sym.renderer.classificationField = field_name
                    sym.renderer.breakCount = 5
                lyr.symbology = sym #update symbology changes



    def postExecute(self, parameters):
        return
```

Citations

U.S. Census Bureau. (2020). 2009-2020 American Community Survey 5-year Public Use [API]. Retrieved from https://factfinder.census.gov/faces/nav/jsf/pages/searchresults.xhtml?refresh=t

Yelp. (2022, December 5). *Search businesses*. Yelp. https://docs.developer.yelp.com/reference/v3_business_search