Lab Report

Title: Lab 01 Deliverable
Notice: Dr. Bryan Runck
Author: Taylor Andersen-Beaver
Date: 10/8/22

Project Repository: https://github.com/and03449/GIS5571.git
Google Drive Link: N/A
Time Spent: 32.5 hours

Abstract

For the first part of this lab, I will be analyzing the difference between 3 APIs: MN Geospatial Commons, NDAWN Weather Data, and Google Places. Specifically, I will be focusing on the difference between searching within these APIs and the process of pulling data from each.

For the second part of the lab, I will be using my new knowledge about these APIs to pull data from them and analyze them within ArcPro using a Jupyter Notebook. I will be analyzing the actual steps to pull the data from a website, ensuring they are all in the same coordinate reference system, spatially joining the datasets, ensuring the join was run correctly with a field header, and lastly, saving the dataset to a GeoDataBase.

Problem Statement

Analyzing the difference between APIs by capturing digitally and saving locally a dataset from each API: MN GeoCommons, NDAWN, and Google Places. Once saved, analyze the datasets by matching coordinate reference systems, spatially joining the datasets, and saving the new joined dataset to a GeoDataBase.

Table 1. Steps Required to Capture and Analyze API Data

#	Requirement	Defined As	(Spatial) Data	Attribute Data	Dataset	Preparation
1	MN Geospatial Commons API	API used to search for spatial data in Minnesota	Boundaries, XY Coordinates, etc	Locations, Counts, etc	Many possible	Understanding how to search the API for data
2	NDAWN API	API used to search for weather data in North Dakota and some surrounding states	Weather stations	Temperatures, Wind, Precipitation	CSV with multiple possible variables	Understanding how to search the API for data
3	Google Places API	API used to search for endless points of interest	XY Coordinates	Locations, Hours, etc	JSON with data of the POI	Understanding how to search the API for data

4	ArcGIS Pro + Jupyter Notebook	Software used to analyze and visualize spatial data	XY Coordinates, addresses, etc	Locations, Names, etc	Many accepted	Understanding how to operate ArcGIS Pro and arcpy
5	arcpy.SpatialReference	Setting a spatial reference for data	Coordinate Reference System	Any	Any	Understanding coordinate reference systems
6	arcpy.management.Define.Projection	Projecting a set spatial reference to a dataset	Spatial Reference & coordinates	Any	Any	Setting a spatial reference
7	arcpy.analysis.SpatialJoin	Joining two datasets using shared spatial data	XY coordinates, addresses, etc	Any	Any as long as it has spatial data	At least two datasets with matching spatial data
8	arcpy.ListFields	Listing the fields in an attribute table	Any	Any	Any	Having an attribute table
9	arcpy.conversion.FeatureClassToGeodatabase	Converting a feature class to a GeoDataBase	Any	Any	Any	Having a feature class

Input Data

The data I will be using from MN Geospatial Commons is the MN County Boundaries and the status of public data within each county, specifically if it is public, public with policy, or not public/unknown. The data I am using for NDAWN is weather data from Ada, MN of the maximum temperatures and minimum temperatures from the last 7 days. Lastly, I will be using data from Google Places on Blegen Hall at the University of Minnesota.

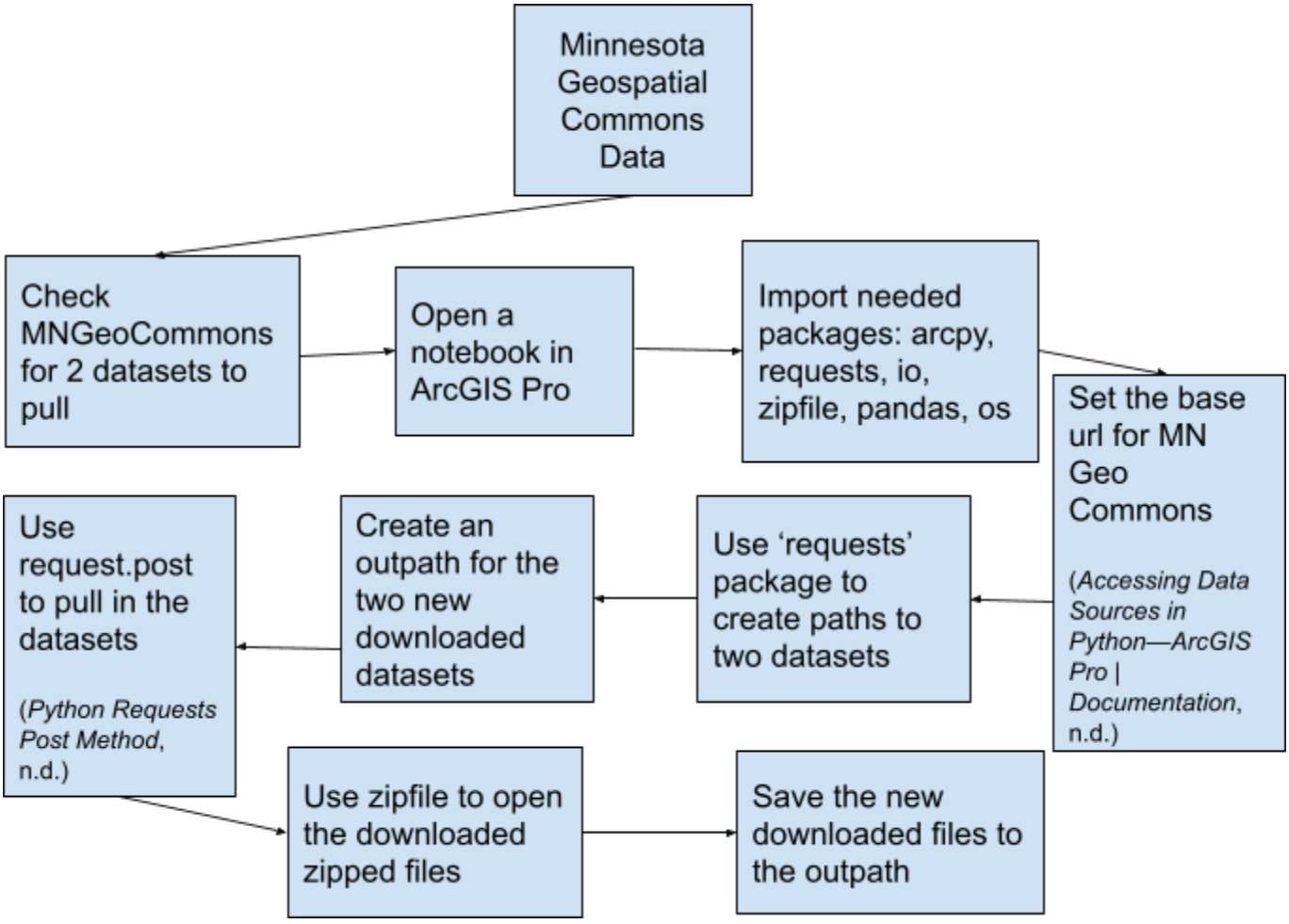
Table 2. All Datasets from Each API: MN GeoCommons, NDAWN Weather Data, Google Places

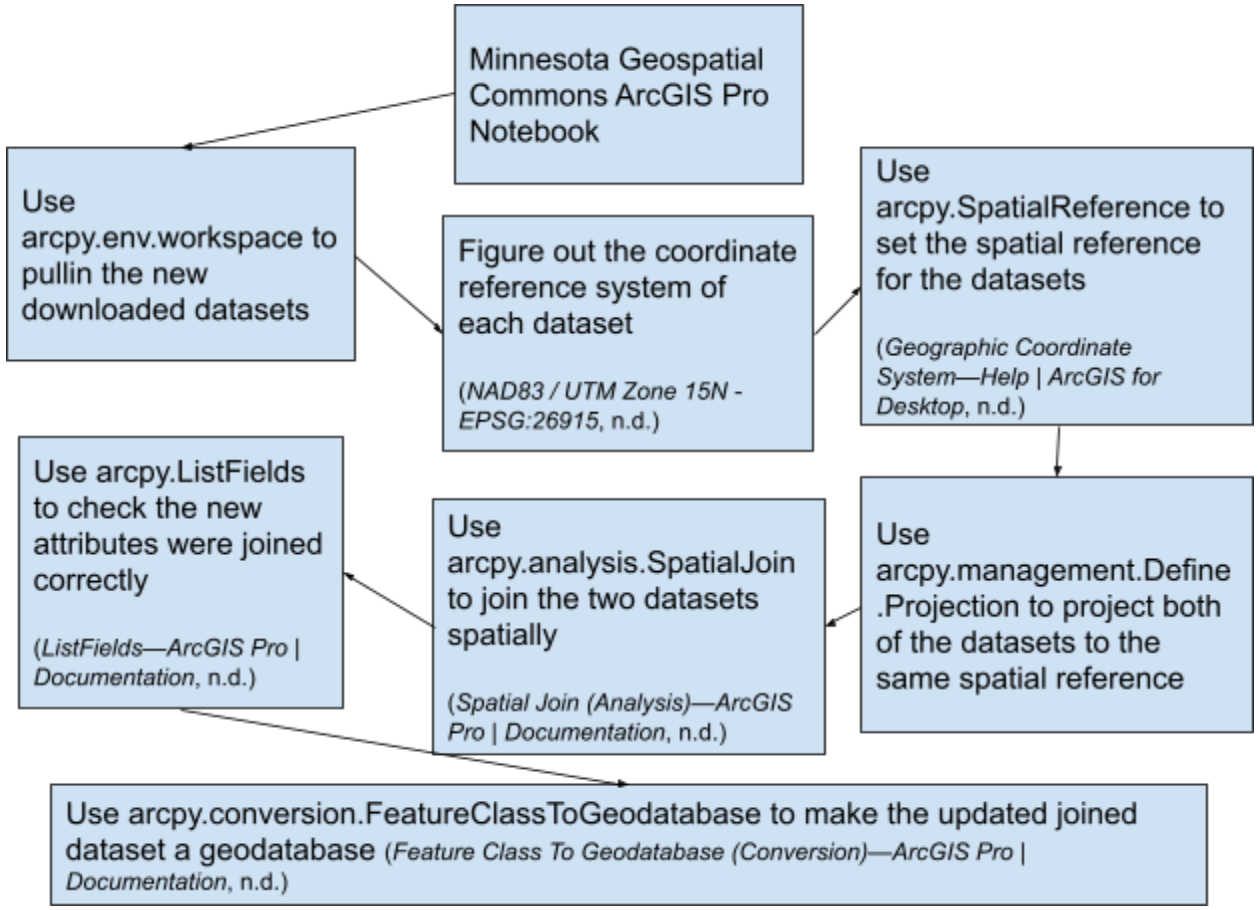
#	**Title**	**Purpose in Analysis**	**Link to Source**
1	MN County Boundaries	To have set boundaries for analysis	https://gisdata.mn.gov/dataset/bdry-counties-in-minnesota
2	MN County Status on Public or Not Public Data	To have a data for each MN county to be able to join to the boundaries	https://gisdata.mn.gov/dataset/bdry-mn-county-open-data-status
3	Ada, MN Max Temperature Data for last 7 days	To have a dataset with temp averages and a location	https://ndawn.ndsu.nodak.edu/get-table.html?station=78&variable=ddmxt&year=2022&ttype=daily&quick_pick=7_d&begin_date=2022-10-06&end_date=2022-10-06

4	Ada, MN Min Temperature Data for last 7 days	To have a dataset with different temp averages and the same location	https://ndawn.ndsu.nodak.edu/get-table.html?station=78&variable=ddmnt&year=2022&ttype=daily&quick_pick=7_d&begin_date=2022-10-06&end_date=2022-10-06
5	Blegen Hall, University of Minnesota	To analyze the data structure from a Google Places search	

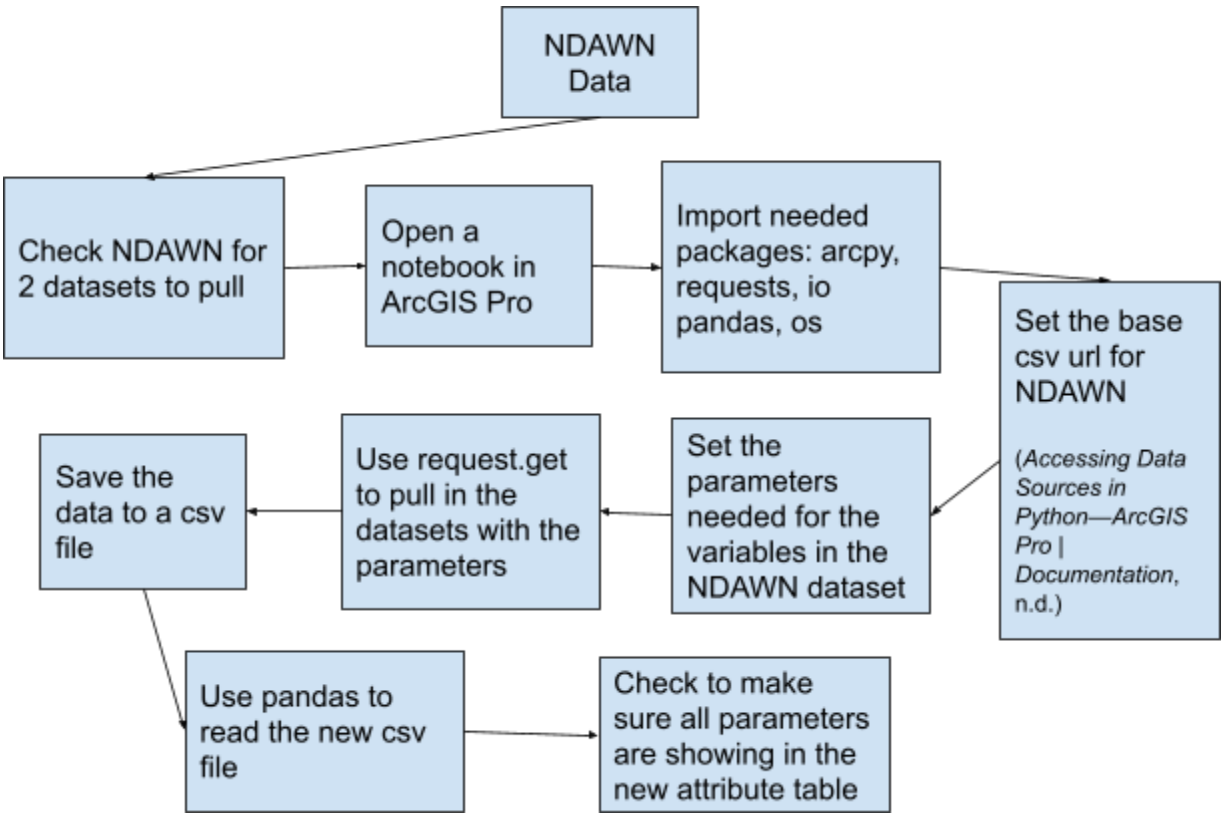
Methods

MN Geospatial Commons

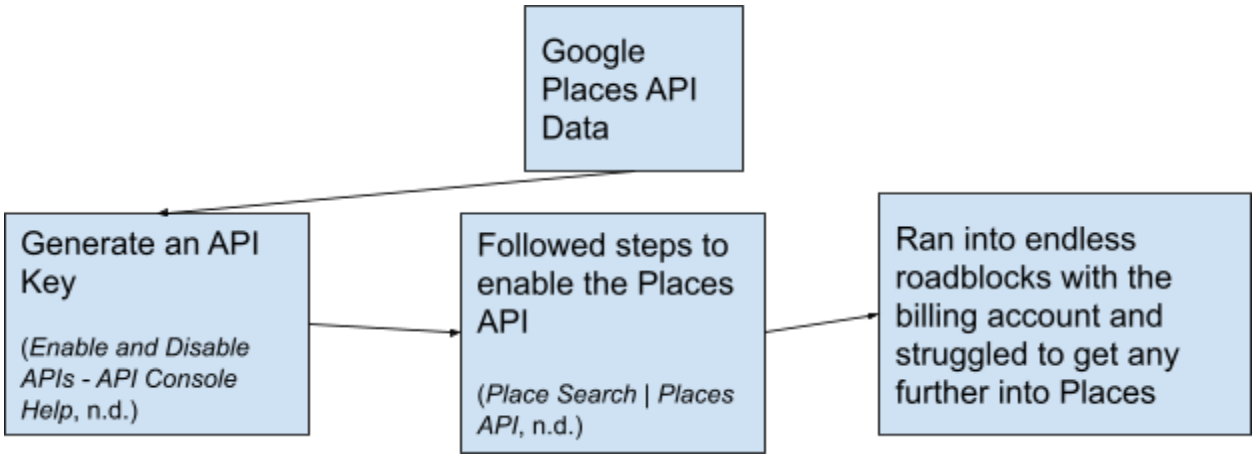



Minnesota Geospatial Commons ArcGIS Pro Notebook
Use arcpy.env.workspace to pullin the new downloaded datasets
Figure out the coordinate reference system of each dataset
(NAD83 / UTM Zone 15N - EPSG:26915, n.d.)
Use arcpy.SpatialReference to set the spatial reference for the datasets
(Geographic Coordinate System—Help | ArcGIS for Desktop, n.d.)
Use arcpy.management.Define .Projection to project both of the datasets to the same spatial reference
Use arcpy.analysis.SpatialJoin to join the two datasets spatially
(Spatial Join (Analysis)—ArcGIS Pro | Documentation, n.d.)
Use arcpy.ListFields to check the new attributes were joined correctly
(ListFields—ArcGIS Pro | Documentation, n.d.)
Use arcpy.conversion.FeatureClassToGeodatabase to make the updated joined dataset a geodatabase (Feature Class To Geodatabase (Conversion)—ArcGIS Pro | Documentation, n.d.)

NDAWN


NDAWN Data
Check NDAWN for 2 datasets to pull
Open a notebook in ArcGIS Pro
Import needed packages: arcpy, requests, io pandas, os
Set the base csv url for NDAWN
(Accessing Data Sources in Python—ArcGIS Pro | Documentation, n.d.)
Set the parameters needed for the variables in the NDAWN dataset
Use request.get to pull in the datasets with the parameters
Save the data to a csv file
Use pandas to read the new csv file
Check to make sure all parameters are showing in the new attribute table

Google Places



Results

Comparing the APIs:

The MN GeoCommons API is a CKAN with a built-in RESTful API, which is an open-source data management system that makes it easy to store, search and share data (*API Developer Resources | Gisdata.Mn.Gov*, n.d.; *CKAN - The Open Source Data Management System*, n.d.). Because the MN GeoCommons is a CKAN it makes it easy to be able to use code to directly pull data from the website and use it in a script (*CKAN - The Open Source Data Management System*, n.d.). The NDAWN website, however, is also a RESTful API but without the simplicity of sharing, searching, and storing data (*NDAWN - North Dakota Agricultural Weather Network*, n.d.). The NDAWN website is much more simple than the MN GeoCommons in that you can still search data by specific parameters but you can only export in a simplified CSV format. What is useful about the NDAWN API is that you can see the parameters being used in the dataset clearly within the url. Lastly, the Google Places API differs from the other APIs in that the data is not necessarily all organized in a visual format but rather depends on a HTTP search to specify which data you are looking for (*Overview | Places API*, n.d.). Once the data has been scaled down to the locations or points of interest needed, the data can be seen in the url and also within a JSON format to get the more specific data such as XY coordinates.

Results from using the APIs:

- MN Geospatial Commons:

  The Geospatial Commons has always been an easy website to use in terms of locating the data needed for a project however I have always used it in a way of locating it myself, downloading the data, uploading the data to whatever platform I am using and working from there. Using the system from this lab, we are able to locate and download the data to a known location without ever having to leave the Jupyter Notebook, whether in ArcGIS Pro or ArcGIS Online. The tricky part of this lab was finding the base location of where

to pull the data from and learning the 'request' package to be able to pull the chosen data from the website. Once the data was located and pulled correctly however it was easily downloaded, read, and analyzed within ArcGIS Pro. Saving the data as shapefiles and then later onto a GeoDataBase made it easy to be able to pull the data out and perform certain analyses with few issues.

- NDAWN

  The NDAWN data differed from the MN GeoCommons data in that it was only available in a sort of rudimentary CSV format. I say rudimentary because it only had one format pretty simplified however you could pick whichever parameters needed for the project. There are copious amounts of variables: max temp, min temp, times, wind strength, etc; as well as offer a variation of times: last day, last 7 days, last 30 days, etc. This data was pulled into ArcGIS Pro differently from the MN GeoCommons data as well since it was only available in a CSV file. The parameters were easily set as they were right in the url used for the data set and could easily be copied and pasted into the Jupyter Notebook. I did not process this data through any of the ArcGIS Pro analyses since I had with the MN GeoCommons however I think it would be relatively simple since most of the data I have processed in a Jupyter Notebook before is in a CSV format.

- Google Places

  Google Places gave me the most grief as it is a complex API including API keys and administrative access in order to be able to fully investigate the API. Unfortunately I was unable to figure it out in time but my understanding is that using Google Places it is possible to search for any point of interest and get all of the relevant data about it: XY coordinates, open/close hours, nearby points of interest, etc. Once a point of interest is identified, it is possible to extract the data into a JSON that can be pulled into the ArcGIS Pro Notebook and analyzed.

Results Verification

- MN Geospatial Commons

  These results can be verified by following the steps listed above in the 'Method' section and comparing the results to verify accuracy. Another way to verify results is by comparing the data tables from each dataset (before joining them) to the original datasets on the website and making sure they were pulled correctly.

- NDAWN

  Similarly to the MN GeoCommons, these results can be verified by copying the steps above in the 'Method' section and comparing the results as well as comparing the results to the original dataset on the website. This may even be easier than the MN GeoCommons because it is a simple CSV format on the website rather than a shapefile which is the most common method on the MN GeoCommons.

- Google Places

  Unfortunately these results cannot be verified because I was unable to fully execute this step. However, if I was able to, I would verify them by comparing the results I extracted from the JSON to what shows on the Google Places site. I would assume this could also be verified similarly to the APIs above by copying the steps in the 'Method' section and comparing the results for accuracy.

Discussion and Conclusion

In this lab I learned about the different APIs that are out there and how to pull data from them to be used in a project. This is an important skill to have because it makes it easier to be able to pull data into a notebook and use it in a script. It is also important because when data is pulled right from the site and uploaded straight into whatever software is being used for the analysis, the data will stay the same from when it was pulled. By using the method of pulling the data straight from the website, if there are any changes or updates that will be reflected the next time the data is pulled.

Searching through the different APIs taught me that there are many different ways to store, search, and save data which helps me to approach new APIs. Although this lab was challenging to learn the different APIs, pull data from them, and analyze that data, I am grateful to have done it and learned the process and be able to further this skill in the future.

References

Accessing data sources in Python—ArcGIS Pro | Documentation. (n.d.). Retrieved October 7, 2022, from https://pro.arcgis.com/en/pro-app/latest/arcpy/geoprocessing_and_python/data-sources-in-python.htm

API Developer Resources | gisdata.mn.gov. (n.d.). Retrieved October 8, 2022, from https://gisdata.mn.gov/content/?q=help/api

CKAN - The open source data management system. (n.d.). Ckan.Org. Retrieved October 8, 2022, from http://ckan.org/

Enable and disable APIs—API Console Help. (n.d.). Retrieved October 7, 2022, from https://support.google.com/googleapi/answer/6158841?hl=en

Feature Class To Geodatabase (Conversion)—ArcGIS Pro | Documentation. (n.d.). Retrieved October 8, 2022, from https://pro.arcgis.com/en/pro-app/2.9/tool-reference/conversion/feature-class-to-geodatabase.htm

Geographic Coordinate System—Help | ArcGIS for Desktop. (n.d.). Retrieved October 7, 2022, from https://desktop.arcgis.com/en/arcmap/10.3/guide-books/map-projections/geographic-coordinate-system.htm

ListFields—ArcGIS Pro | Documentation. (n.d.). Retrieved October 8, 2022, from https://pro.arcgis.com/en/pro-app/latest/arcpy/functions/listfields.htm

NAD83 / UTM zone 15N - EPSG:26915. (n.d.). Retrieved October 7, 2022, from https://epsg.io/26915

NDAWN - North Dakota Agricultural Weather Network. (n.d.). Retrieved October 8, 2022, from https://ndawn.ndsu.nodak.edu//

Overview | Places API. (n.d.). Google Developers. Retrieved October 8, 2022, from https://developers.google.com/maps/documentation/places/web-service/overview

Place Search | Places API. (n.d.). Google Developers. Retrieved October 7, 2022, from https://developers.google.com/maps/documentation/places/web-service/search

Python Requests post Method. (n.d.). Retrieved October 7, 2022, from https://www.w3schools.com/python/ref_requests_post.asp

Spatial Join (Analysis)—ArcGIS Pro | Documentation. (n.d.). Retrieved October 7, 2022, from https://pro.arcgis.com/en/pro-app/2.9/tool-reference/analysis/spatial-join.htm

Self-score

Category	Description	Points Possible	Score
Structural Elements	All elements of a lab report are included **(2 points each)**: Title, Notice: Dr. Bryan Runck, Author, Project Repository, Date, Abstract, Problem Statement, Input Data w/ tables, Methods w/ Data, Flow Diagrams, Results, Results Verification, Discussion and Conclusion, References in common format, Self-score	28	**28**
Clarity of Content	Each element above is executed at a professional level so that someone can understand the goal, data, methods, results, and their validity and implications in a 5 minute reading at a cursory-level, and in a 30 minute meeting at a deep level **(12 points)**. There is a clear connection from data to results to discussion and conclusion **(12 points)**.	24	**20**
Reproducibility	Results are completely reproducible by someone with basic GIS training. There is no ambiguity in data flow or rationale for data operations. Every step is documented and justified.	28	**25**

Verification	Results are correct in that they have been verified in comparison to some standard. The standard is clearly stated **(10 points)**, the method of comparison is clearly stated **(5 points)**, and the result of verification is clearly stated **(5 points)**.	20	**20**
		100	**93**