**Lab Report**

Title: lab 1

Notice: Dr. Bryan Runck

Author: Maochuan Wang

Date: Oct 4, 2022

**Project Repository:**https://github.com/MaochuanW/GIS5571/tree/main/Lab1

**Google Drive Link: *N/A***

**Time Spent:** 105 hours

**Abstract**

There are two major sections in this lab, the first one is to compare and contrast the API models for MGC, google, and NDAWN. It is discussed in the methods section. The second part of the lab is to build an ETL that can extract data from those websites and perform some spatial analysis on the dataset. The methodology behind the ETL pipeline is to use various python packages to retrieve the data and then transform the data into something ArcPro can read. For Mngeo and Ndawn, they are CSV files, and for google place API they are JSON files. The process of converting CSV from Mngeo and Ndawn is straightforward with some manipulation of the CSV table. The process of converting JSON to CSV is done by extracting keywords from the JSON and then parsing them into a new CSV file. Once the CSV files are created, the rest of the spatial analysis was done by using the Arcpy package. The results are shown in the shapefile that the ETL created in the geodatabase.

**Problem Statement**

Spatial web APIs are crucial to understanding because with a good understanding of API we can build an ETL pipeline that automates processes and process data easier. Building an ETL requires extensive knowledge of programming and to build a decent ETL, it requires a good understanding of the API conceptual model of each website. There are several important aspects of a good ETL. For example, how easy it is to retrieve information and dataset from the API link, how automate the process is, and if the output file is readable by GIS software so you can do analysis on it.

Table 1. Data requirement

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **#** | **Requirement** | **Defined As** | **(Spatial) Data** | **Attribute Data** | **Dataset** | **Preparation** |
| 1 | Data from MNgeo | Dataset from Mngeo |  | CSV file | [Mn GeoSpatial Commons](https://gisdata.mn.gov/dataset) |  |
| 2 | Data from Google place API | Dataset from google | Json file |  | [Google API](https://maps.googleapis.com/maps/api/place/details) |  |
| 3 | Data from NDAWN | Dataset from NDAWN |  | CSV file | [NDAWN](https://ndawn.ndsu.nodak.edu/) |  |

**Input Data**

Data used in this lab includes 2 csv files from MNGeo that display the MPA’s interests’ sites in the state. The data from google place api is a json file that contains the details of the place from 2 restaurants in dinkytown. Data from NDAWN contains daily precipitation data from 2 different weather stations in North Dakota.

Table 2. Data used in this lab

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Title** | **Purpose in Analysis** | **Link to Source** |
| 1 | MPA interests site | Shows the location of the sites in csv file format so I could do analysis on the file with arcpro | [Mn GeoSpatial Commons](https://resources.gisdata.mn.gov/pub/gdrs/data/pub/us_mn_state_pca/env_mpca_agency_interests) |
| 2 | Google place API | Convert json file into a csv so Arcpro can read | [Google places API](https://maps.googleapis.com/maps/api/place/details/json?place_id=ChIJ9xuqBjsts1IR0AcbOIzg3ZU&fields=name%2Crating%2Cformatted_phone_number%2Cgeometry&key) |
| 3 | NDAWN | CSV file that has the location but need reformatting | [NDAWN](https://ndawn.ndsu.nodak.edu/table.csv?station=26&begin_date=2022-10-02&end_date=2022-10-02&quick_pick=4_w&ttype=hourly&variable=hdt) |
|  |  |  |  |

**Methods**

For the MNGeo CKAN API. In order to make an API Request, we have to post a JSON dictionary in an HTTP POST request to one of CKAN’s API URLs. The JSON file should contain the parameter of the file you want to download, and CKAN will return the response in a JSON dictionary. See flow chart below.

Diagram

Description automatically generated

For the Google place API, the conceptual model is a bit different than then CKAN API. First of all, Google Maps platform products are secured from unauthorized use by restricting API calls to those that have the proper credentials, or the API Keys. After obtaining the API keys, you can start requesting with the use of a https link. There are a couple functions that you can do with Google place api, such as Find place, Nearby Search, Text Search, place details, place photos…etc. And each of those function uses a unique HTTP URL. In general, the response from Google API is either a json output or a xml output. The parameter of the requests are typed in each unique HTTP URL. See flow chart below.

Diagram

Description automatically generated

For the NDAWN API, the API request is similar in a way to Google API, where you modify the parameters in the API link. But in this case, NDAWN uses its HTTPS website as the API call, and there is no unique API Key required. So, to find your desired data, you just have to modify the link to get your data. See flow chart below for the process.

Diagram

Description automatically generated

For the second part of the lab, which is building an ETL by using API call. The idea behind it is to first get the data from each of the API, then store the data into a location on local hard drive, and then convert the file into something ArcPro can read, and then do the spatial analysis such as spatial join by using Arcpy, and lastly export the file into a geodatabase. The process of using ArcPro and Arc online notebook is almost identical, except that in ArcPro, the work environment is on my local machine, where in ArcOnline, it’s in the cloud. See flowchart below for process.

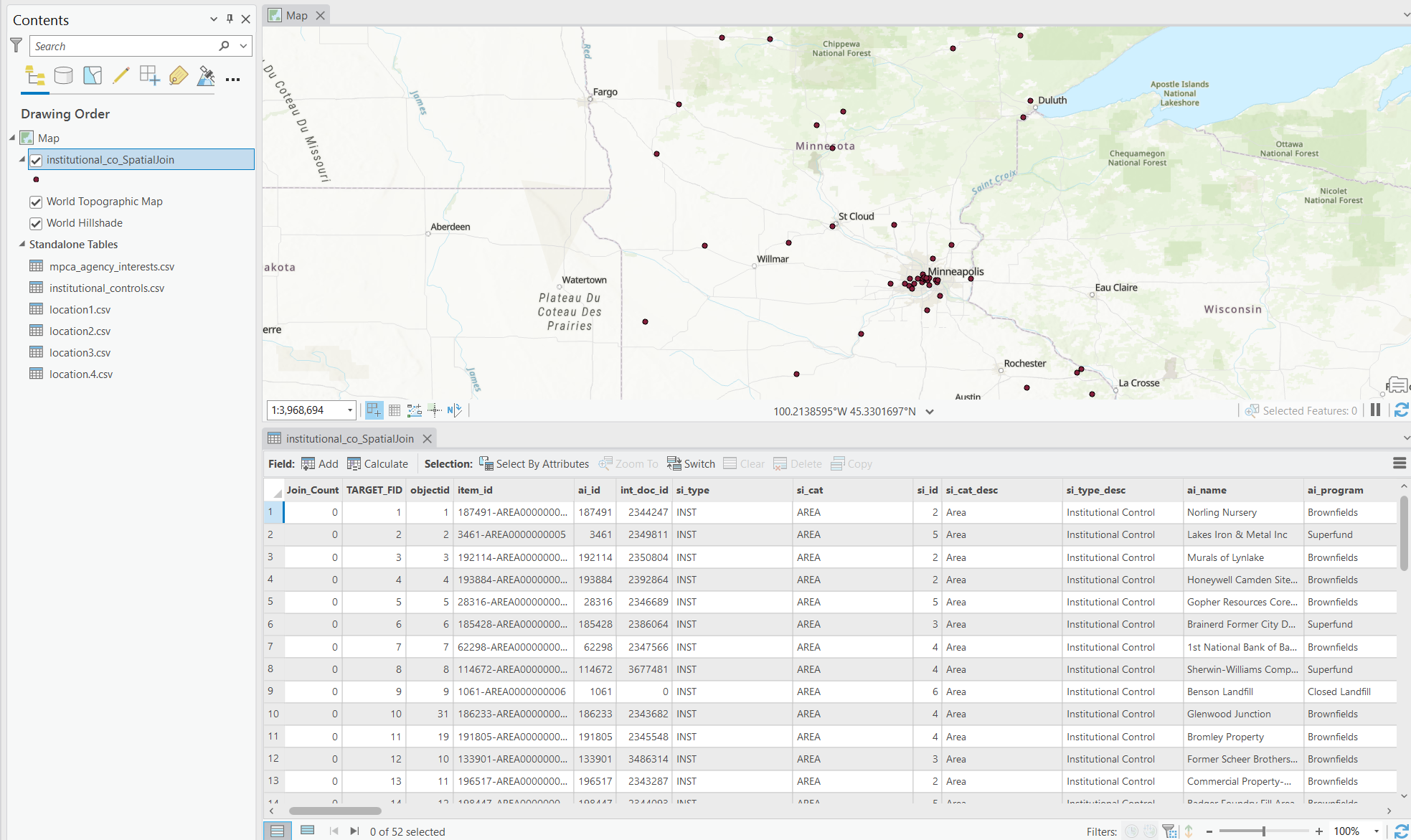
Diagram, schematic

Description automatically generated

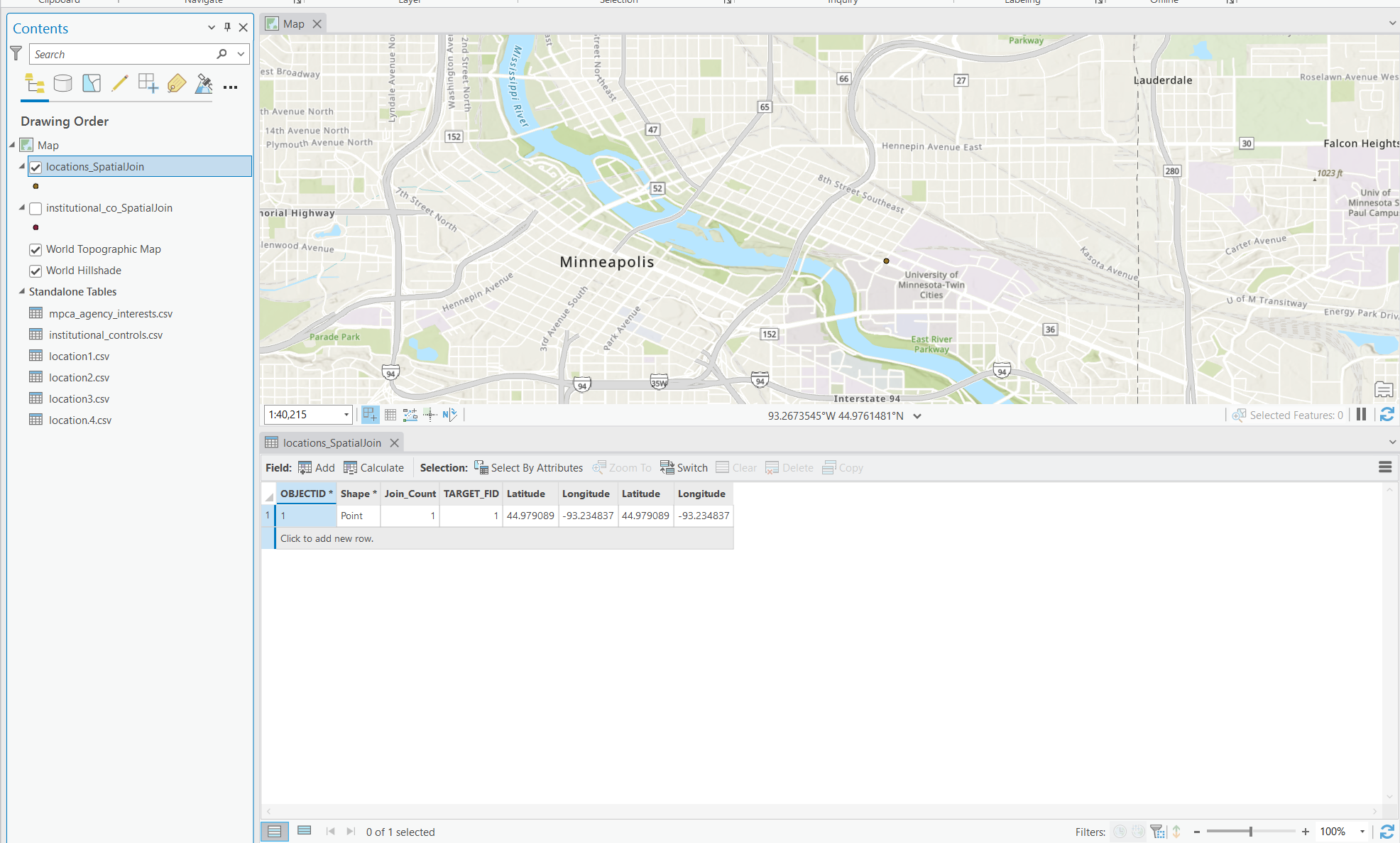
**Results**

The results of the lab can be displayed by the shapefiles that ETL made. Because there are 3 APIs I did and each API has two datasets. Each ETL is supposed to make 2 shapefiles, one is from the XY table to point, the other one is the spatial joined one. In order to show the result, I believe shapefile with its attribute open in arc pro should be sufficient to prove that the ETL process is successful.

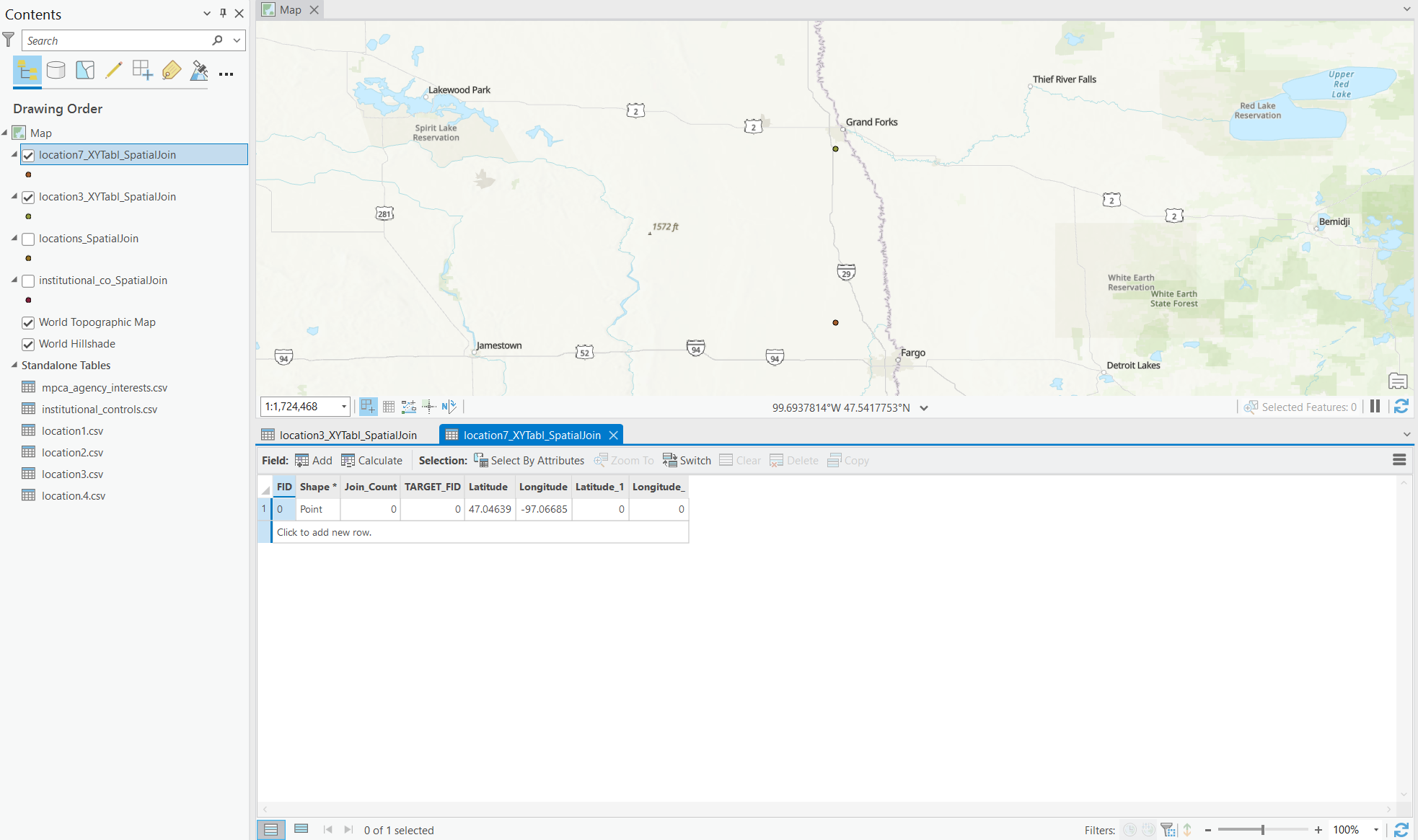
MnGEO data looks properly spatial joined with all the fields append into one single shapefile.



Google API data looks like its properly spatial joined with 2 sets of coordinates.



NDAWN data:



There are some empty values in the NDAWN data, so I’m not really sure why that happened. Probably some parts were wrong in my code, or it could be improper spatial join in the notebook.

**Results Verification**

For results verification, I manually spatial joined in arcpro and compared the data set such as looking at the attribute table, and the map. And it turned out, the results for MNgeo and google place is correct, as it output the identical file as the one that got produced from notebook. The one with NDAWN is different as I mentioned in the result section above, and that could be some error either with the code or with the improper arcpy command.

**Discussion and Conclusion**

After finishing this lab, I had a good understanding of how to build an ETL pipeline to retrieve data from website APIs. The process itself is complicated, especially when it comes to the different file you get from different APIs, so sometimes I had to figure out ways convert those files into something ArcPro can read, which to me I think it’s the hardest part of this lab. Even though I did end up making a ETL that does the job of retrieving, transforming, exporting, there are still many parts of my code that is not an ETL, such as the part where I manually put lat and lon value into a list. I believe there is still a lot of parts of this ETL that can be improved.

**References**

*Use a common format*

*NDAWN - North Dakota Agricultural Weather Network*. (n.d.). Retrieved October 5, 2022, from https://ndawn.ndsu.nodak.edu//

*Overview  |  Places API  |*. (n.d.). Google Developers. Retrieved October 5, 2022, from https://developers.google.com/maps/documentation/places/web-service/overview

*The CKAN API — CKAN Documentation 2.1.5 documentation*. (n.d.). Retrieved October 5, 2022, from https://docs.ckan.org/en/ckan-2.1.5/api.html

**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 | **23** |
| **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 | **27** |
| **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 | **98** |