

Lab Report

Title: Lab 1 – Comparing, Contrasting and Extracting Data from APIs

Notice: Dr. Bryan Runck

Author: Chris Carter

Date: 10/05/2022

Project Repository: <https://github.com/cart0588/GIS5571/tree/main/Lab1>

Google Drive Link:

https://drive.google.com/drive/folders/1mmRohE30_zz_NGNQNOuv6jn3tIzVUHrR?usp=sharing

Time Spent: 30 hours

Abstract

APIs allow users to interact with data stored on various sources. Three such APIs (Minnesota Geospatial Commons, Google Places and North Dakota Agricultural Weather Network) require different processes to interact with and download data. The data access through these sources comes in different formats and requires different processing pipelines to load, process and display the data in ArcGIS Pro or ArcGIS Online. This Lab will download and process two datasets from each of these three APIs and project, spatially join and display the data and their attributes. Using Python in both software packages, data can be accessed and loaded relatively easily, with some changes required depending on the format data is accessed in.

Problem Statement

APIs provide interfaces through which users can access, download, and process data, either individually or in bulk. Different kinds of APIs exist and require different processes to access the data stored by the services that use them. Three APIs will be accessed in this Lab: Minnesota Geospatial Commons, Google Places and North Dakota Agricultural Weather Network.

This lab will use Python in ArcPro and ArcOnline to access these APIs and display, project and spatially join the data accessed through them to explore different APIs and data formats.

Input Data

The input data is taken from three sources: Minnesota Geospatial Commons, Google Places and North Dakota Agricultural Weather Network.

Minnesota Geospatial Commons data is accessed by locating the data on the MGC website, selecting the desired file format, and copying the download link into the Notebook. MGC data will be accessed in Shapefile format.

Google Places data is accessed by constructing a custom hyperlink to access specific search results. First, an API Key must be generated, which allows users to access data using unique customized credentials. Google Places data will be accessed in JSON format.

NDAWN data is accessed by selecting the desired stations and attributes and generating a table, which produces a download link to a CSV table.

Table 1. Data sources

#	Title	Purpose in Analysis	Link to Source
1	Airports Information in Minnesota	Raw input data, downloaded from API, loaded into Arc, processed, and spatially joined	MN Geospatial Commons
2	Impaired Wetlands 2012	Raw input data, downloaded from API, loaded into Arc, processed, and spatially joined	MN Geospatial Commons

3	Google Places Data	Raw input data, downloaded from API, loaded into Arc, processed, and spatially joined	See code for URLs
4	All Stations Daily Data (August 18, 2022)	Raw input data, downloaded from API, loaded into Arc, processed, and spatially joined	<u>NDAWN</u>
5	All Stations Daily Data (February 18, 2022)	Raw input data, downloaded from API, loaded into Arc, processed, and spatially joined	<u>NDAWN</u>

Methods

Minnesota Geospatial Commons uses a CKAN based REST (representational state transfer) API which allow users to access data hosted on the MGC website directly using their own code. MGC hosts data in a range of formats, such as HTML, shapefiles, CSV and others.

Google Places uses the Places API, which returns data using HTTP requests using URLs that users can create using their desired parameters. Google Places returns data in either XML or JSON format.

NDAWN hosts raw data in CSV format, which can be accessed on an individual station level or by attribute across stations.

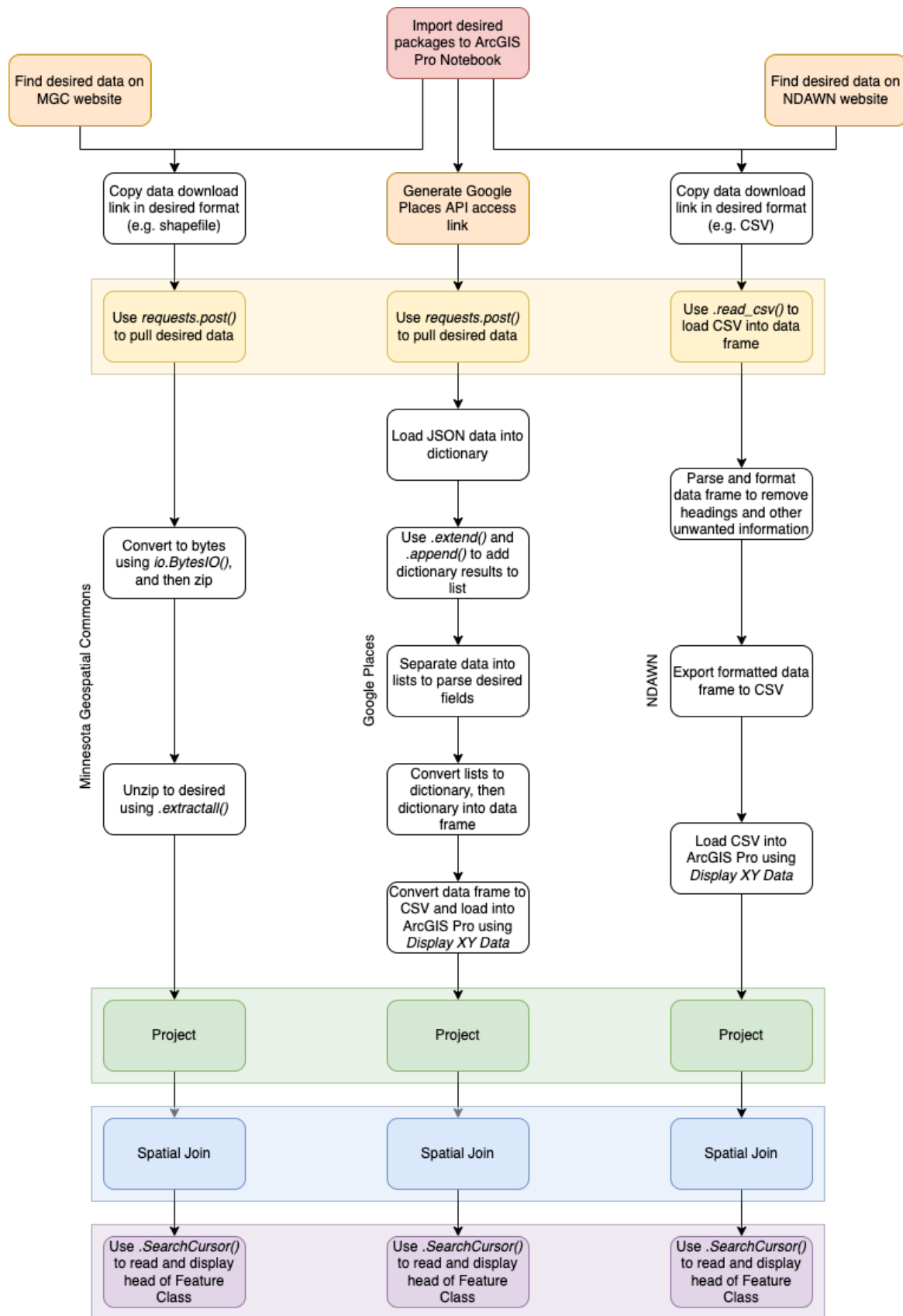
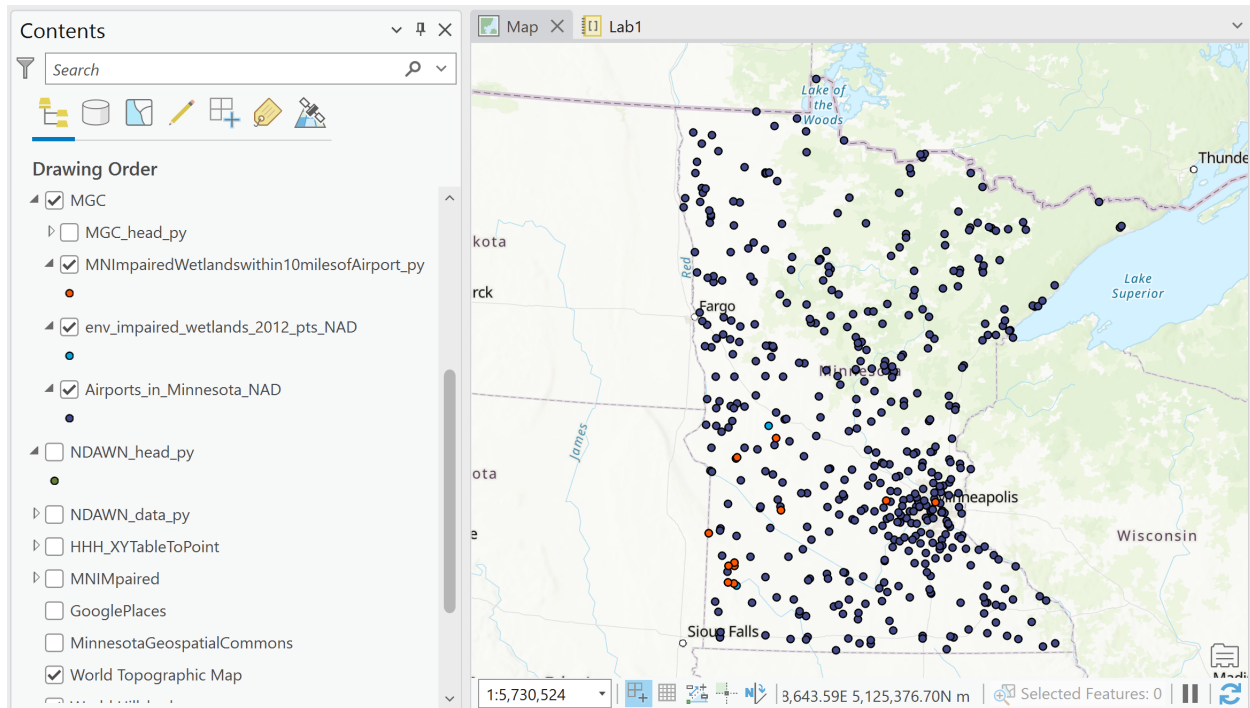


Figure 1: Data pipeline for three APIs

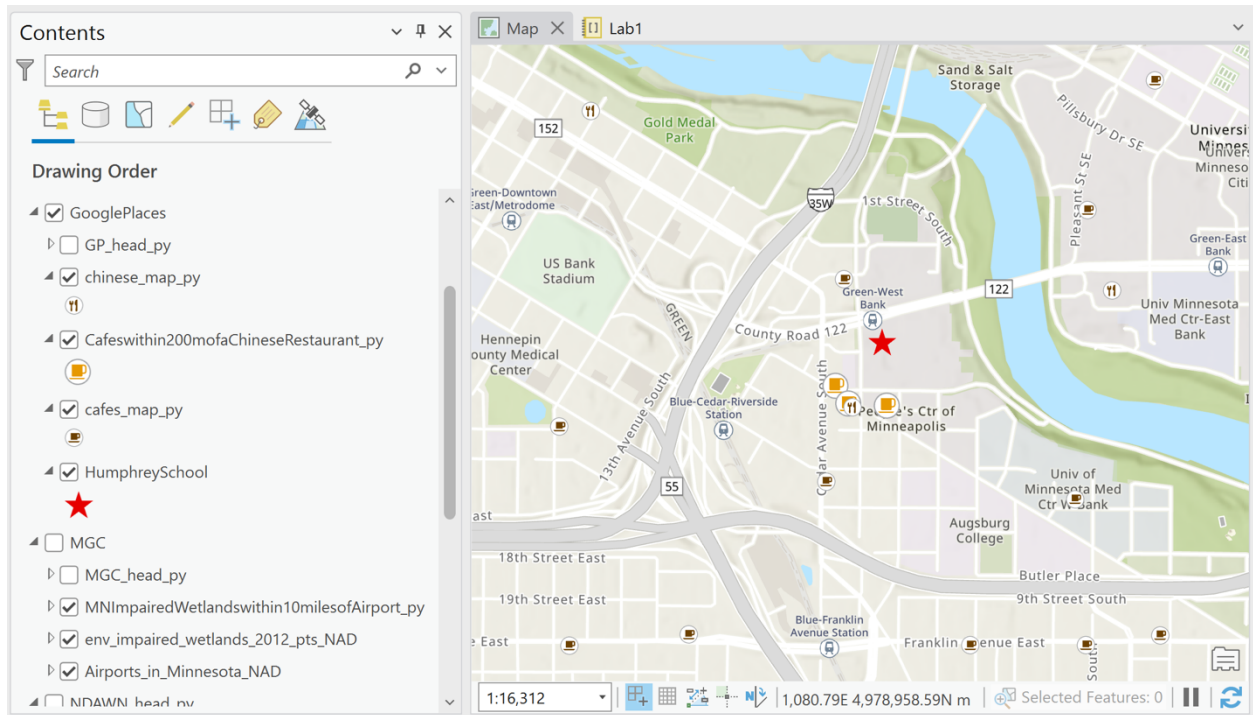
Results

The data downloaded and used was processed as follows:

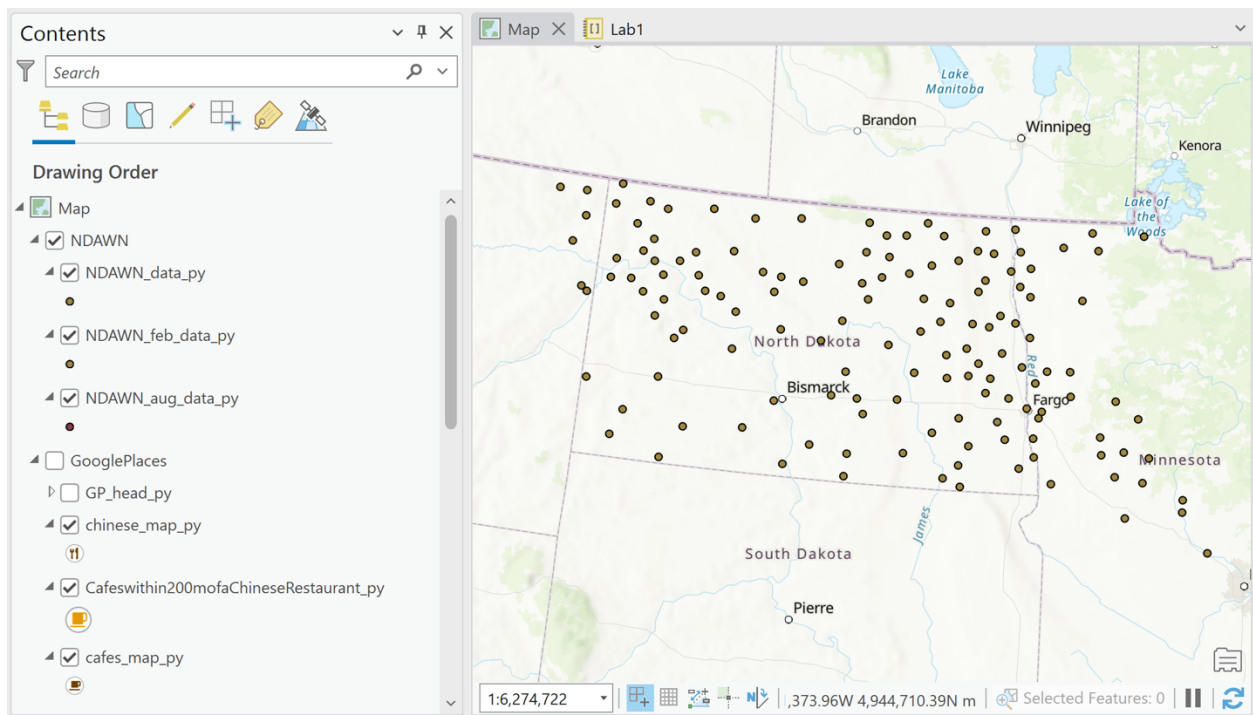
- Minnesota Geospatial Commons
 - o Downloaded airport and impaired wetland data
 - o Project data to a new geographic and projected coordinate system
 - o Spatially join to find all impaired wetlands within 10 miles of an airport



- Google Places
 - o Create custom hyperlink to download locations of all cafes and Chinese restaurants within 1km of the Humphrey School
 - o Parse JSON data and project to a spatial coordinate system
 - o Spatially join data to find all cafes within 200m of a Chinese restaurant



- NDAWN
 - o Download CSV data for all stations in August and February 2022
 - o Project data to spatial coordinate system
 - o Spatially join data



Results Verification

Results were evaluated visually and compared between API sources and ArcGIS Pro and ArcGIS Online to ensure consistency between platforms and data sources. Data was also compared to manually downloaded and processed data to ensure Python-processed data returns the same outputs. Results were consistent and reliable across APIs.

In addition, the head of each output feature class was displayed in order to allow for verification of data.

MGC:

	0	1	2	3	4	5
0	1	Pochardt Slough	Lyon County	LINCOLN	44.291633	-96.150311
1	2	Weltz Slough	Lyon County	LINCOLN	44.291633	-96.150311
2	4	Jones	Ramsey County	RAMSEY	45.076910	-93.204391
3	4	Jones	Ramsey County	RAMSEY	45.033299	-93.125221
4	4	Jones	Ramsey County	RAMSEY	44.942583	-93.107844

Google Places:

	0	1	2	3
0	2	Cafe Jote	44.970162	-93.246703
1	13	Hard Times Cafe	44.969637	-93.246185
2	14	Caribou Coffee	44.969591	-93.244597

NDAWN:

	0	1	2	3	4	5
0	Ada	47.321190	-96.514060	20.863	-2.128	9.368
1	Adams	48.499880	-98.075880	16.160	-8.986	3.587
2	Alamo	48.546520	-103.471860	34.421	-8.500	12.961
3	Alvarado	48.245942	-97.021532	19.330	-3.046	8.142
4	Amidon	46.488440	-103.316290	35.956	6.026	20.991

Discussion and Conclusion

Accessing the APIs was relatively simple to figure out. The primary challenge for this lab was figuring out how each data format could be loaded into Arc and processed in Python. For example, the fact that a JSON file is essentially a Python dictionary in text form was not immediately obvious to me, and thus the processing pipeline was more difficult than it maybe should have been.

Getting back on my feet in Python was an enjoyable experience even if this Lab was frustrating in some ways. ArcOnline once again presented far more problems than ArcPro, with the cloud integration of the system proving challenging to use when trying to download and store data.

I successfully managed to do all necessary processing in both ArcPro and ArcOnline and received consistent and logically accurate results on both platforms.

References

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	26
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	24
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	28
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	20	20

	verification is clearly stated (5 points).		
		100	98