Lab Report

Title: Lab 1
Notice: Dr. Bryan Runck
Author: Kyle Smith
Date: October 8, 2024

Project Repository: /kylejsmith4/GIS5571/Lab1
Time Spent: 75 hours

Abstract

This project analyzes three spatial APIs: Minnesota Geospatial Commons, ArcGIS Online REST API, and North Dakota Agricultural Weather Network (NDAWN). By accessing and manipulating data from these APIs, we demonstrate their functionality and explore their differences. The project involves extracting federally designated strategic highway data, county boundary polygons, and average monthly high and low temperatures over a 48 month period for two NDAWN stations. An Extract, Transform, Load (ETL) pipeline was developed to perform a spatial join between NDAWN weather data and Minnesota county boundaries, identifying which counties the selected stations are in.

The project highlights how code can be used to manipulate spatial data through APIs. Each API requires different approaches, ranging from user friendly interfaces with detailed documentation, to more complex APIs that require custom URL construction. This project provides insights into spatial API interactions, the development of ETL pipelines, and the challenges in spatial analysis.

Problem Statement

This project involves the analysis of three different APIs – from the Minnesota Geospatial Commons, ArcGIS Online REST API and the North Dakota Agricultural Weather Network (NDAWN). An exercise in accessing each API and manipulating data from each will help to better understand how each API functions. Upon accessing each API the following data will be shown: Road geometry of federally designated strategic highways in Minnesota, Polygons of county boundaries in the state of Minnesota, and the average monthly high & low temperatures over a 48 month period at two NDAWN station sites selected in the state of Minnesota. Additionally, the development of an Extract, Transform, Load (ETL) using these APIs will further demonstrate how code can be used to solve problems and manipulate abstractions. This project will prepare a ETL pipeline which performs a spatial join between the NDAWN and County boundary data, and as to demonstrate which Minnesota counties each of the selected NDAWN sites are in.

#	Requirement	Defined As	(Spatial) Data	Attribute Data	Dataset	Preparation
1	Strategic Highway Network in Minnesota	Feature Service from MNDOT	Road geometry of federally designated strategic highways in Minnesota	OBJECTID ROUTE_ID FROM_MEASURE TO_MEASURE STRAHNET_TYPE DESCRIPTION Shape__Length	Mn GeoSpatial Commons	Access via CKAN
2	County Boundaries in Minnesota	Feature Service from MNDOT	Polygons of county boundaries in the state of Minnesota	County_Name County_Code FIPS55_Code GNIS_Feature_ID ATP_Code	MN Geospatial Commons	Access via ARC GIS REST API

3	North Dakota Agricultural Weather Network (NDAWN)	API	Raw data. Specifically, the average monthly high & low temperatures over a 48 month period at two NDAWN station sites selected in the state of Minnesota.	STATION NAME Avg Max Temp Avg Min Temp Latitude Longitude SHAPE	NDAWN	Access via user defined code
4	Spatial Join	Defined by user	NDAWN and County boundary data will be joined as to demonstrate which Minnesota counties each of the selected NDAWN sites are in.	STATION NAME Avg Max Temp Avg Min Temp Latitude Longitude SHAPE COUNTY_NAME	Defined by user	Spatial Join (ARC GIS API)

Input Data

#	Title	Purpose in Analysis	Link to Source
1	Strategic Highway Network in Minnesota	Road geometry of federally designated strategic highways in Minnesota	Link
2	County Boundaries in Minnesota	Polygons of county boundaries in the state of Minnesota	Link
3	North Dakota Agricultural Weather Network (NDAWN)	Raw data. Specifically, the average monthly high & low temperatures over a 48 month period at two NDAWN station sites selected in the state of Minnesota.	Link

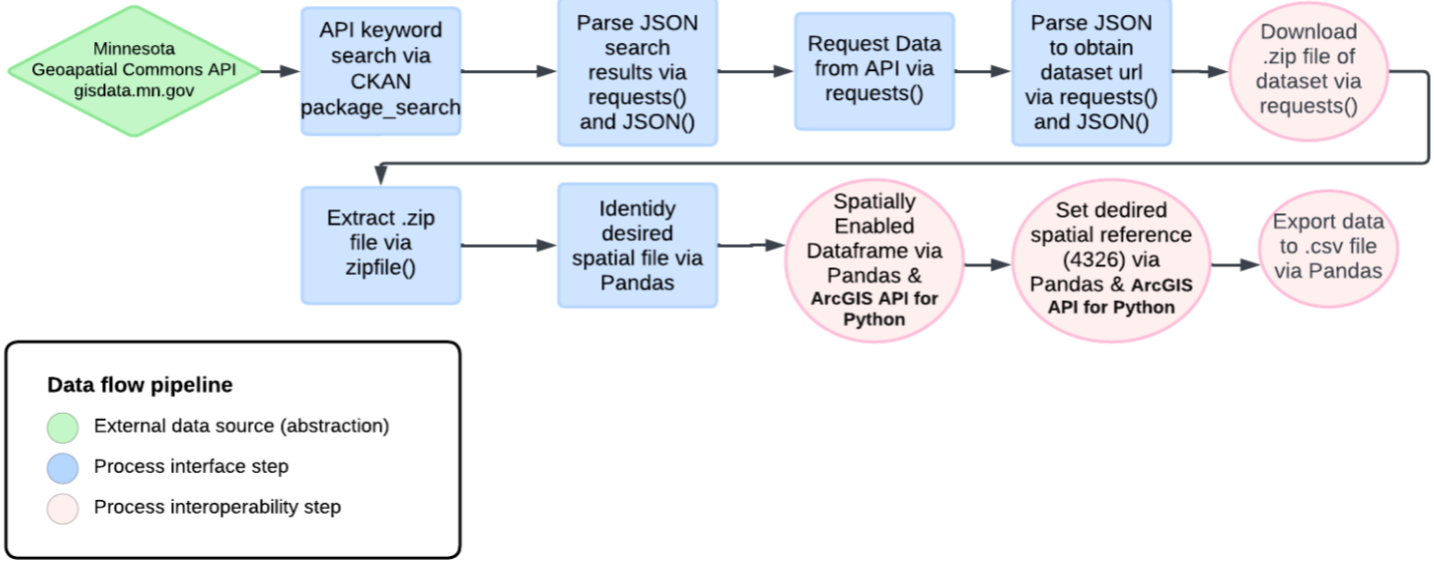
The project utilizes three primary datasets sourced from distinct spatial web APIs. The Minnesota Geospatial Commons hosts the Strategic Highway Network in Minnesota dataset. This consists of road geometry of federally designated strategic highways in Minnesota. A dataset of county boundaries of the state of Minnesota, provided by the state department of transportation, can be accessed via ArcGIS Online REST API. The North Dakota Agricultural Weather Network (NDAWN) is a repository of weather station data from across the northern Midwest.

The Minnesota Geospatial Commons and ArcGIS Online REST API datasets will be accessed via CKAN and ARC GIS API respectively. Data will be analyzed as a JSON, dataframe, spatially enabled dataframe, and finally a csv file. NDAWN data will be analyzed by first understanding the API and its attributes, then producing a custom url to download a csv file. Steps will be taken to ensure all datasets are in the same reference system.

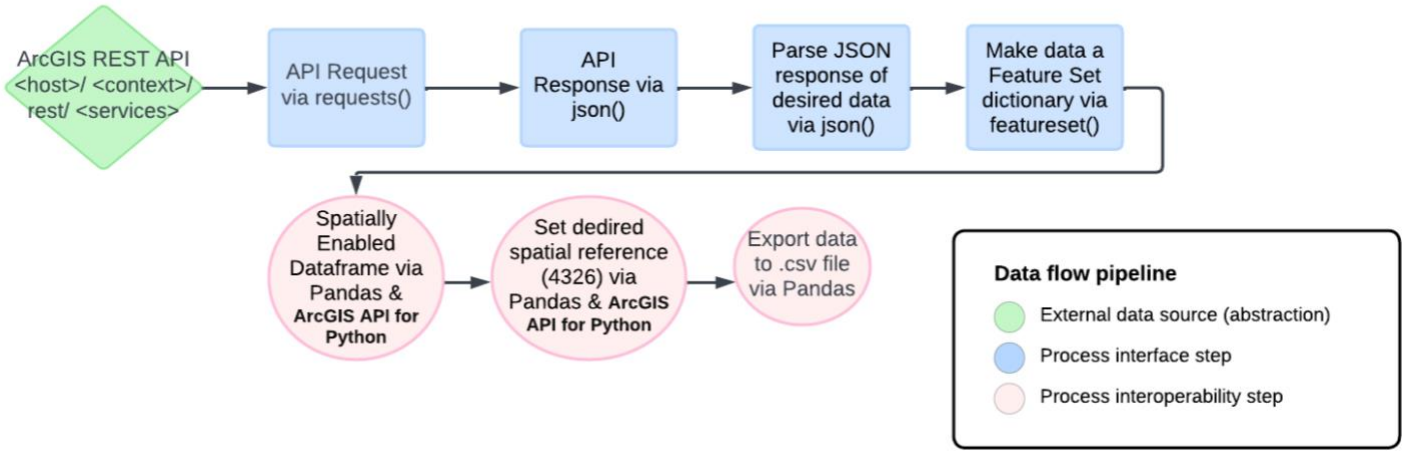
Methods

APIs are tools to enable different software programs to communicate. Upon direction from the user, APIs can be useful in accessing datasets, which can be manipulated by code. As we see in this project, there are different kinds of APIs with different degrees of useability and customization. Some APIs can be much easier to access as platforms exist to guide the user, others leave the user to take a deep dive into code and urls in order to understand and access what is desired. The flow of data and information to and from each API discussed in this project, as well as the spatial join of data can be demonstrated below:

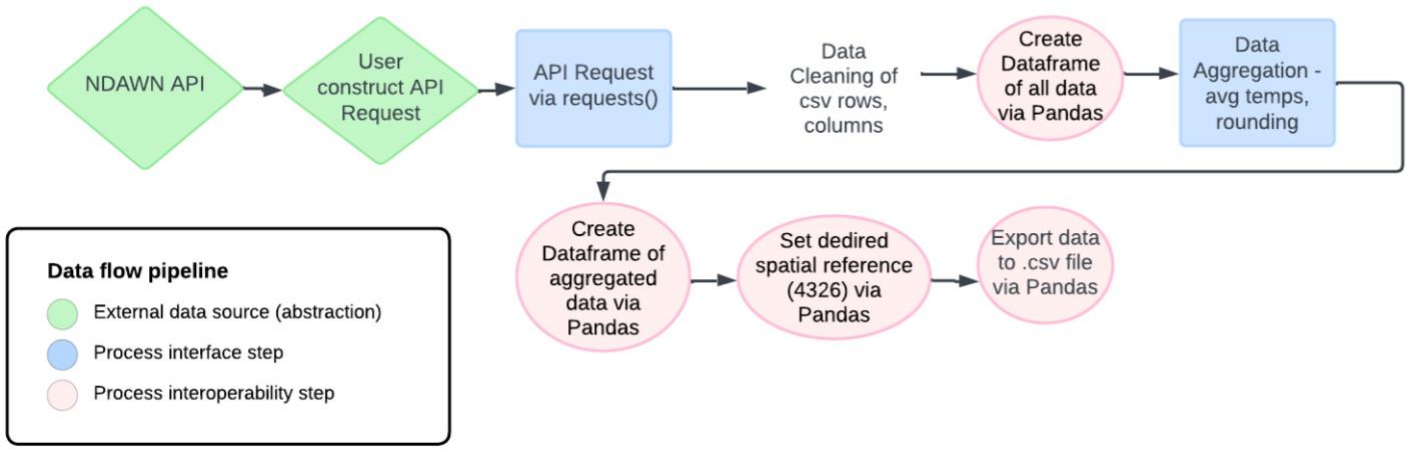
1. Minnesota Geospatial Commons API (Strategic_Highways)



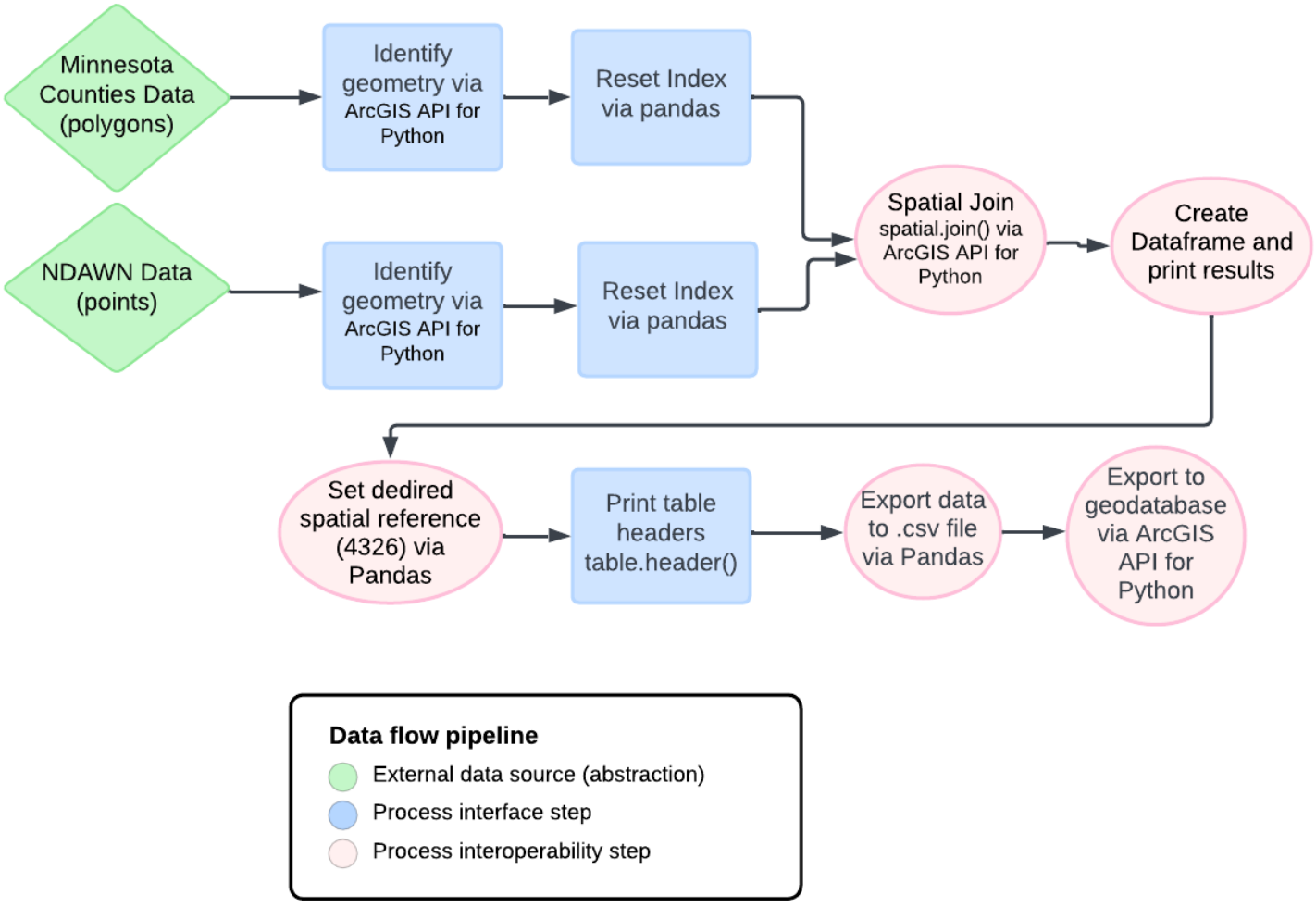
2. REST API (mn_counties)



3. NDAWN (Humboldt & Becker, MN)



Spatial Join



Charts by LucidChart

Each API can now be further constructed piece by piece as to better understand its construction:

Minnesota Geospatial Commons API (Strategic_Highways)

https://gisdata.mn.gov/	main domain and portal for the website
api/	indicates that an api request of the domain is to follow
3/	version of the api?
action/	requesting the api to perform an action
package_search?	type of query - here the api is being asked to use the terms which follow to perform a search
q=trans-federal-routes	the query (q) search terms

This api request produces a lengthy json which includes search results. Next steps are to parse the json to isolate the specific file to be downloaded. For this project, we identified a database file of federal roads, which can be downloaded, opened, and the appropriate spatial file identified

JSON parse	['result']['results'][0]['resources'][0]['url']
targeted zip file	trans_federal_routes.zip
local file to receive zipped data	trans_federal_routes_data
spatial file to be extracted	Strategic_Highway_Network_in_Minnesota.shp

REST API (mn_counties)

https://webgis.dot.state.mn.us/	main domain and portal for the website - Minnesota DOT
65agsf1/	unique id for this session?
rest/	the specific type of api, in this case it is a REST api, which interacts with ESRI servers
services/	the api is requesting to access data services hosted by ESRI
sdw_govnt/	specific dataset of the Minnesota DOT (statewide government data?)
COUNTY/	subset of the requested dataset which is specific to county-level data
FeatureServer/	the request is for a ESRI feature service - which means spatial data
0/	the specific layer of the feature service, 0 means the first layer
query?	requesting the api to a search following the "?"
where=	the first query condition - asking how to filter data
1%3D1&	represents the input "1=1" which means the user wants all records
outFields=	the next query condition - asking which record fields to produce
*&	represents the input "*" which means the user wants all fields
returnGeometry=	the next query condition - asking if the geometry should be included
true&	represents the user input that geometry is requested
f=	the final query condition - asking the format of the response
pjson	represents the user requests json format

NDAWN (Humboldt & Becker, MN)

The NDAWN api requires the user to understand the available api attributes and variables before producing a custom url to access requested data. For this project, the api listed below is used. This dataset provides the average monthly high & low temperatures over a 48 month period at two NDAWN station sites selected in the state of Minnesota.

https://ndawn.ndsu.nodak.edu/	main domain and portal for the website - North Dakota Agricultural Weather Network (NDAWN)
table.csv?	the api is requested to provide data in a csv format (comma deiminated) as opposed to an html table. The "?" indicates that quary specifics are to follow
station=118&	The first query is for a "station" location. Each are identified by a number. The user must identify the desired station numbers and add them here
station=4&	stations 118 and 4 are Humboldt and Becker, MN
variable=mdmxt&	the next section of the query are for the "variables" of the type of data, one at a time, to be provided for each of the stations
variable=mdmnt&	mdmxt is monthly daily max temperature, and mdmnt is monthly daily min temperature
year=2024&	the next query is of the year in which the requested data starts (or ends)
ttype=monthly&	this query is for the type of time duration unit requested, in this case it is monthly data
quick_pick=4_y&	quick_pick means that these variables are selected from a list or entered in a field which is prompted, 4_y means the request is for data going back 4 years.
begin_date=2019-01&	this quary is the beginning month for the first requested piece of data, monthly duration units as was selected above
count=12	finally, the query for the number of periods of monthly duration units, 12 is selected, meaning the whole year.

What results is a csv documents of raw data. It can be helpful to view this in a spreadsheet or csv viewer as to identify what need to be cleaned up. For this data, several header rows were merged and need to be skipped so the next steps of calculating averages and displaying results can commence.

Each api is a bit different in its construction, organization, and how it is accessed. The Minnesota Geospatial Commons API and REST API are both similar in that they are both user friendly in determining attribute variables, and that both have a significant amount of documentation available to assist the user. The Minnesota Geospatial Commons is a web based repository for the state's gis data and can be quarried on the web to find what is desired. However, as each database can be updated by the state without notice, it is prudent to learn about the CKAN functions which are the base organization for this data. CKAN allows many different functions, and here we used a search query to find the dataset we want. This means that the dataset can be updated, and even the names of files changed, and yet the user is always directed to the desired dataset.

The REST API is a direct connection to a custom web based repository provided by ESRI for a specific state department, agency, etc. For this project, we used the Department of Transportation data for the county boundaries of Minnesota. This dataset can be found on the rest servers via the web, and can be specifically modified to meet the user's query needs via a api construction table. This is where the user identifies what fields, records, output, etc. is desired and a custom url is created.

Finally, the NDAWN api is the most unique and perhaps can be considered less user friendly despite the general simplicity of the weather data it provides. As has been mentioned, the user must understand the available api attributes and variables before producing a custom url to access requested data via trial and error. Upon successful request of data, NDAWN does not necessarily provide a universally friendly output format as several header rows were merged on the csv output, which requires manual revisions. If I was a heavy user of NDAWN I would suggest it be organized in a more user friendly manner, especially with complex api requests.

It can be noted that the spatial join of two apis involves familiarity with the ArcGIS API for Python. Once the desired data is successfully extracted, but before it can be joined, it must be checked for common spatial reference, for identification of the geometry in each dataset (and geometry must be an acceptable spatial format), and for the indexes of the datasets to be reset go make sure they will align. The function to perform the spatial join asks how to join the data: left, right, inner, or outer – this indicates what data the join will be based off and which data isn't relevant to this request. Also, the function asks for how the geometries should be joined: intersect, within, contains, equals, overlaps, touches, or crosses – each of these perform a different action with the geometry types. For this project, we performed a left – within spatial join – meaning that the designated left dataset (NDAWN) will match the attributes of the right dataset (MN Counties), and if a match cannot happen, NaN is displayed. Further, the request is for a within join of the geometries of the two datasets (point data – NDAWN and polygons – MN Counties). The geometries from the left dataset (NDAWN) are completely within the geometries from the right dataset (MN Counties). This can mean some cleaning up and organizing of the data can be necessary before displaying it.

The spatial join looks like this in Python:

```
spatial_join = sdf_NDAWN.spatial.join(sdf_mn_counties, how='left', op='within')
```

Results

The Python notebook submitted with this project includes several markdown comments which explain functions and organization of code as it happens as well as the results of the data requests. However, the question of average monthly max and min temperatures over the 48 month period for the NDAWN sites at Becker and Humboldt can be answered:

Becker, MN:	Average Max Temp: 55.49881
	Average Min Temp: 35.49517
Humboldt, MN:	Average Max Temp: 50.26750
	Average Min Temp: 28.63329

So the average max temperature for the 48 month period at Humboldt, MN (north Minnesota) is about 5.23 degrees cooler then the average max temperature for Becker, MN (south Minnesota). Further, the average min temperature for the 48 month period at Becker, MN (south Minnesota) is about 6.87 degrees warmer then the average min temperature for Humboldt, MN (north Minnesota). All units are in degrees (F).

Finally, the question asked in the spatial join can be answered - Which counties in Minnesota are the two selected NDAWN weather sites are in?

-- Becker, MN is in Sherburne County, Minnesota
-- Humboldt, MN is in Kittson County, Minnesota

Results Verification

The first thought in discussing how to verify the results is that the Python code worked – therefore it must be correct! While it did take some time to work through the code and successfully get the api requests to work – I must acknowledge that there could still be errors in code organization, best practices, and simplicity. I relied quite heavily upon documentation when I was stuck, and while generally helpful, my answers may not necessarily have been properly focused.

Specifically, throughout the code there were several checks of a dataframe and results before moving on to the next step. It is difficult to show this in the final product as it has been condensed for ease of use and review. Each api request was checked to be the same spatial reference, and the calculations of the final results (average temperature data) were confirmed to be correct. This could be done thanks to the small sized data set. Further, the two NDAWN sites were confirmed to be identified in the correct counties – this could also be manually checked due to the small data size.

Discussion and Conclusion

This project taught me much about Python and how to access and manipulate spatial data. While I realize this is just an introductorily exercise, the trial and error of formatting code and building correct functions was valuable. Yes, this lab took me a long time, and we had a long time to complete it – however I hope I gained some efficiencies and knowledge to do better with the next lab.

This lab also successfully provided a deep dive and comparison of 3 spatial apis, and how to develop a functioning ETL pipeline to access, manipulate, and demonstrate the data.

References

ArcGIS API for Python Documentation. (n.d.). Retrieved from https://developers.arcgis.com/python/

LucidChart Tutorials. (n.d.). Retrieved from https://www.lucidchart.com

Python Requests Library Documentation. (n.d.). Retrieved from https://docs.python-requests.org/en/latest/

North Dakota Weather Network API Documentation. (n.d.). Retrieved from https://ndawn.ndsu.nodak.edu/

Minnesota Geospatial Commons API Documentation. (n.d.). Retrieved from hhttps://gisdata.mn.gov

ArcGIS Online REST API Resource Hierarchy. (n.d.). Retrieved from https://developers.arcgis.com/rest/services-reference/

Requests: HTTP for Humans. (n.d.). Retrieved from https://pypi.org/project/requests/

Quickstart: JSON Response Content. (n.d.). Retrieved from https://requests.readthedocs.io/en/latest/user/quickstart/#json-response-content

Pandas Documentation. (n.d.). Retrieved from https://pandas.pydata.org/docs/

Get Started with the ArcGIS Services Directory. (n.d.). Retrieved from https://developers.arcgis.com/rest/services-reference/enterprise/get-started-with-the-services-directory/

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	**21**
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	**12**
		100	**86**