**Lab Report**

Title: Using APIs to retrieve and process data from the Minnesota Geospatial Commons, ArcGIS Online, and NDAWN.

Notice: Dr. Bryan Runck

Author: Timothy Johnson

Date: 8 October 2024

**Project Repository:**<https://github.com/tjjohnson1415/GIS5571>

**Time Spent:** 38 hours

**Abstract**

APIs are used to retrieve data stored on the internet. The Minnesota Geospatial Commons, ArcGIS Online, and NDAWN all have data that can be accessed via an API, but their APIs function very differently. Data was acquired using the three APIs and converted into spatially enabled data frames. The data from the Minnesota Geospatial Commons and ArcGIS Online were then spatially joined and saved to a file geodatabase. I don’t come from a computer science background, so, at first, I found the APIs challenging to use, especially the API behind the Minnesota Geospatial Commons.

**Problem Statement**

Retrieving data from the internet is done using APIs. Although they are very useful, APIs are not standardized, which can make them challenging to navigate. Some APIs have a lot of documentation, such as the ArcGIS Online REST API, while others, like the API behind NDAWN, have none. Data will be called using the CKAN API behind the Minnesota Geospatial Commons, the ArcGIS Online REST API, and the API behind NDAWN. This data will be read into spatially enabled data frames. A spatial join will be performed on the data retrieved from the Minnesota Geospatial Commons and ArcGIS REST Services.

Table 1: Data retrieved using the three APIs.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **#** | **Requirement** | **Defined As** | **(Spatial) Data** | **Attribute Data** | **Dataset** | **Preparation** |
| 1 | Data from Minnesota Geospatial Commons | Raw input dataset of Minnesota counties from Minnesota Department of Natural Resources | County polygons |  | [Minnesota Geospatial Commons](https://gisdata.mn.gov/dataset/bdry-counties-in-minnesota) |  |
| 2 | Data from ArcGIS Online | Raw input dataset of airports in Minnesota from Minnesota Department of Transportation | Minnesota airport points |  | [ArcGIS REST Services](https://webgis.dot.state.mn.us/65agsf1/rest/services/sdw_trans/AIRPORT/FeatureServer/0) |  |
| 3 | Data from NDAWN | Input data of daily maximum temperature on October 2, 2024 | Weather station points | Daily maximum air temperature | [NDAWN](https://ndawn.ndsu.nodak.edu/get-table.html?station=78&station=111&station=98&station=162&station=174&station=142&station=164&station=138&station=161&station=9&station=160&station=224&station=159&station=10&station=118&station=56&station=165&station=11&station=12&station=58&station=13&station=84&station=218&station=55&station=179&station=7&station=186&station=87&station=14&station=15&station=96&station=191&station=16&station=210&station=201&station=137&station=124&station=143&station=17&station=85&station=226&station=140&station=134&station=18&station=136&station=219&station=65&station=104&station=99&station=192&station=19&station=227&station=129&station=20&station=101&station=166&station=178&station=81&station=21&station=97&station=22&station=75&station=184&station=2&station=211&station=172&station=139&station=158&station=23&station=157&station=220&station=62&station=86&station=24&station=89&station=126&station=223&station=167&station=93&station=183&station=90&station=25&station=205&station=83&station=107&station=156&station=77&station=26&station=155&station=70&station=127&station=144&station=27&station=173&station=132&station=28&station=195&station=185&station=29&station=30&station=154&station=31&station=187&station=102&station=32&station=119&station=4&station=217&station=80&station=33&station=59&station=153&station=105&station=82&station=225&station=34&station=198&station=72&station=135&station=35&station=76&station=120&station=209&station=141&station=109&station=36&station=207&station=79&station=193&station=71&station=212&station=37&station=38&station=189&station=39&station=130&station=73&station=188&station=40&station=41&station=54&station=69&station=194&station=145&station=214&station=113&station=128&station=42&station=43&station=103&station=171&station=116&station=196&station=88&station=114&station=3&station=163&station=200&station=216&station=64&station=115&station=168&station=67&station=175&station=146&station=170&station=197&station=44&station=206&station=133&station=106&station=100&station=121&station=45&station=46&station=61&station=66&) | Limit data to October 2, 2024 |

**Input Data**

The datasets used for this lab are a Minnesota counties dataset from the Minnesota Department of Natural Resources, a Minnesota airports dataset from the Minnesota Department of Transportation, and a daily maximum temperature dataset for weather stations that report to the North Dakota Agricultural Weather Network (NDAWN). The counties dataset contains county names and polygon geometry. The airports dataset contains airport names and point locations. The weather stations dataset contains station names, point locations, daily maximum temperatures, and the date the measurement was taken.

Table 2: Data

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Title** | **Purpose in Analysis** | **Link to Source** |
| 1 | Minnesota Counties | Input polygon dataset to spatial join with airports. | [Minnesota Geospatial Commons](https://gisdata.mn.gov/dataset/bdry-counties-in-minnesota) |
| 2 | Minnesota Airports | Input point dataset to spatial join with counties. | [ArcGIS REST Services](https://webgis.dot.state.mn.us/65agsf1/rest/services/sdw_trans/AIRPORT/FeatureServer/0) |
| 3 | Weather Stations in North Dakota and surrounding areas | Data called from NDAWN. Not used in analysis. | [NDAWN](https://ndawn.ndsu.nodak.edu/get-table.html?station=78&station=111&station=98&station=162&station=174&station=142&station=164&station=138&station=161&station=9&station=160&station=224&station=159&station=10&station=118&station=56&station=165&station=11&station=12&station=58&station=13&station=84&station=218&station=55&station=179&station=7&station=186&station=87&station=14&station=15&station=96&station=191&station=16&station=210&station=201&station=137&station=124&station=143&station=17&station=85&station=226&station=140&station=134&station=18&station=136&station=219&station=65&station=104&station=99&station=192&station=19&station=227&station=129&station=20&station=101&station=166&station=178&station=81&station=21&station=97&station=22&station=75&station=184&station=2&station=211&station=172&station=139&station=158&station=23&station=157&station=220&station=62&station=86&station=24&station=89&station=126&station=223&station=167&station=93&station=183&station=90&station=25&station=205&station=83&station=107&station=156&station=77&station=26&station=155&station=70&station=127&station=144&station=27&station=173&station=132&station=28&station=195&station=185&station=29&station=30&station=154&station=31&station=187&station=102&station=32&station=119&station=4&station=217&station=80&station=33&station=59&station=153&station=105&station=82&station=225&station=34&station=198&station=72&station=135&station=35&station=76&station=120&station=209&station=141&station=109&station=36&station=207&station=79&station=193&station=71&station=212&station=37&station=38&station=189&station=39&station=130&station=73&station=188&station=40&station=41&station=54&station=69&station=194&station=145&station=214&station=113&station=128&station=42&station=43&station=103&station=171&station=116&station=196&station=88&station=114&station=3&station=163&station=200&station=216&station=64&station=115&station=168&station=67&station=175&station=146&station=170&station=197&station=44&station=206&station=133&station=106&station=100&station=121&station=45&station=46&station=61&station=66&) |

**Methods**

Minnesota Geospatial Commons:

A diagram of a flowchart

Description automatically generated

Figure 1: Data pipeline for data retrieved from the Minnesota Geospatial Commons.

A dataset of county polygons was retrieved from the Minnesota Geospatial Commons. The CKAN API was used to get a list of the different formats that the data could take. The URL of the shapefile was extracted from the response of the API call. Another approach that the API could have taken would be for an additional argument to be taken that would directly output the data in a desired format. This would eliminate the two-step nature of the way the API currently functions. The shapefile URL was then used to download the data as a zipped shapefile. The data was then unzipped and loaded into a Jupyter Notebook. It was then converted into a spatially enabled data frame and reprojected to the same coordinate system as the data downloaded from ArcGIS Online and NDAWN.

ArcGIS REST API:

A diagram of a process

Description automatically generated

Figure 2: Data pipeline for data retrieved from ArcGIS Online.

A point dataset of airports in Minnesota was retrieved using ArcGIS REST Services. Using the query “where 1=1” allows all data in the dataset to be accessed. Using different arguments, a subset of the full dataset can be accessed. The argument “f=geojson” defines the requested data format, in this case a GeoJSON. The GeoJSON is read with Pandas and converted into a spatially enabled data frame.

NDAWN:

A diagram of a work flow

Description automatically generated

Figure 3: Data pipeline for data retrieved from NDAWN.

A point dataset of weather stations was retrieved from NDAWN. The requested variable was the daily maximum temperature. The start and end dates define the period that the records should come from. The data is outputted as a CSV, which can be directly read into a data frame using Pandas. It is then converted to a spatially enabled data frame.

A diagram of a data flow

Description automatically generated

Figure 4: Data flow diagram for spatially joining datasets and placing the joined data in a geodatabase.

The airport points retrieved from ArcGIS Online were joined with the county polygons retrieved from Minnesota Geospatial Commons. The joined spatially enabled data frame was converted into a feature class and placed in a newly created file geodatabase.

**Results**

The table below shows the first five entries of the airports dataset spatially joined with counties. The data retains the point geometry of the airports, but has additional fields related to the county the airports are in, such as a county ID number and the county name.

Table 3: Spatially joined data.

*A screenshot of a computer

Description automatically generated*

**Results Verification**

For many of the airports can be verified by comparing the name of the airport to the county name. For example, Norman County Ada/Twin Valley Airport should be in Norman County, which is what it has been assigned. Similarly, Aitkin Muni-Steve Kurtz Field has been assigned to Aitkin County, which makes sense. Because the airports that clearly belong in certain counties have been assigned to those counties correctly, the results are likely correct.

**Discussion and Conclusion**

As someone who does not have any background in computer science, it was quite challenging to wrap my head around APIs and how to use them. This lab helped me become more comfortable reading documentation. I found retrieving data from the Minnesota Geospatial Commons by far the most challenging. While it was easy to use the CKAN API to get a list of datasets, it took me a long time to realize that I would need to call a second URL outside of the CKAN API to get the data in a usable format. I found the ArcGIS REST API much easier to use. The query UI was useful for making sure I was getting the data I wanted. I’m still not quite sure what all the fields in the query builder do, though. Although it doesn’t have any documentation, I found the NDAWN API very easy to understand.

**References**

*ArcGIS server services directory REST API: ARCGIS REST apis: Arcgis developers*. ArcGIS REST APIs | ArcGIS Developers. (n.d.). https://developers.arcgis.com/rest/services-reference/enterprise/get-started-with-the-services-directory/

*The CKAN API*. The CKAN API - CKAN Documentation 2.1.5 documentation. (n.d.). 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 | **22** |
| **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 | **97** |