**Lab Report**

Title: Getting Degrees in GIS: Interpolating Regional Temperature Data

Notice: Dr. Bryan Runck

Author: Cecelia Isaac

Date: 4/22/2021 test

**Project Repository:** https://github.com/CeceliaAi/GIS5572/tree/master/Lab4

**Abstract**

The problem this lab tackles is downloading weather data real-time as recorded by various weather stations, and then using that data to map and interpolate the temperature of the areas surrounding these stations. The data used will come from NDAWN. We are looking for the past 30 days of data from every available station. We will build an ETL in ArcPy in order to download this data. After that, the data will be cleaned and processed, and then run through various interpolation tools. The results will be a series of layers interpolated by three different tools. The results can be verified by successfully running the tools and creating the layers on the map frame. However, as the purpose of the lab is to compare three tools, there is no perfectly correct answer.

**Problem Statement**

The first problem is to create an ETL to obtain our data. We will build the ETL using Python, with the end goal of a CSV of daily temperature data for each station. Then, we will use the CSV to find the 30-day average temperature so that each station only has one number associated with it. Once the points are mapped, we will then have to use different interpolation tools to compare and contrast interpolation methods and results.

*Table 1. Data requirements*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **#** | **Requirement** | **Defined As** | **Spatial Data** | **Attribute Data** | **Dataset** | **Preparation** |
| 1 | Average daily temperature data from every station | CSV file | Lat/Lon columns | Latitude, Longitude, Elevation, Average Temperature | [NDAWN](https://ndawn.ndsu.nodak.edu/get-table.html?station=78&variable=ddmxt&year=2021&ttype=daily&quick_pick=30_d&begin_date=2021-04-17&end_date=2021-04-17) | Averaged over the month and grouped by station, then converted to points layer |

**Input Data**

The data will be a CSV file listing the station name, latitude, longitude, elevation, and average temperature. The table should show the previous thirty days from the current date (the date the data is being downloaded).

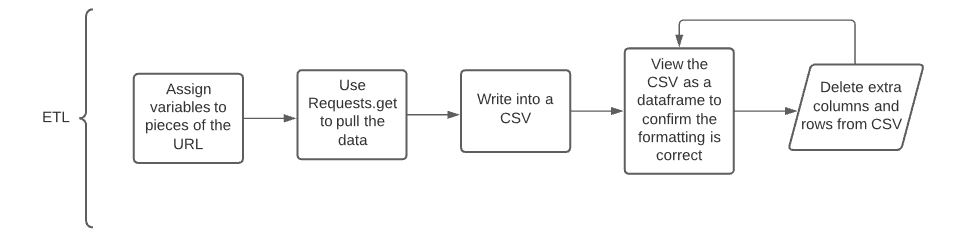
*Table 2. Input data*

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Title** | **Purpose in Analysis** | **Link to Source** |
| 1 | NDAWN Daily Average | Used to find the monthly average by station, which can then be interpolated through the region. | [NDAWN](https://ndawn.ndsu.nodak.edu/get-table.html?station=78&variable=ddmxt&year=2021&ttype=daily&quick_pick=30_d&begin_date=2021-04-17&end_date=2021-04-17) |

**Methods**

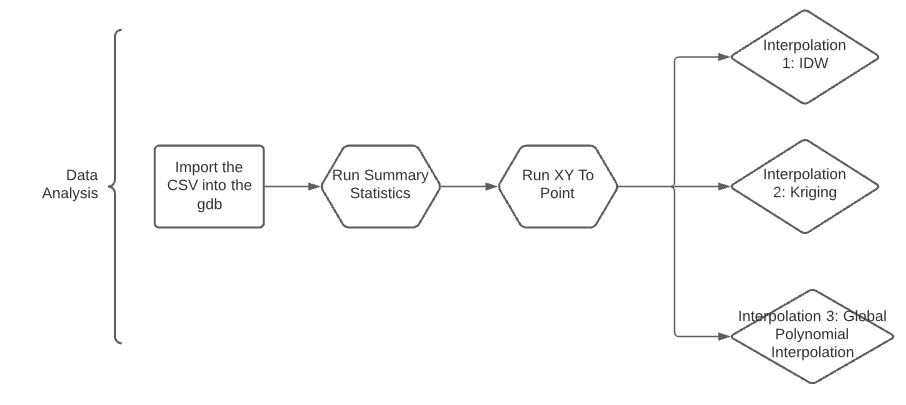
Firstly, we must make the ETL. I found the page I wanted on NDAWN and then parsed the pieces of the URL into variables that could be potentially changed or updated in the future. For the date, I used built-in Python functions and some of Jeff’s syntax to establish a current date, so that the data downloaded always reflects the previous 30 days. After this, I used requests.get to pull the webpage information, and wrote it into a CSV. When trying to view the CSV, I ran into a few errors because of bad rows. Ultimately, the simplest solution was to manually delete the offending rows, rather than use Python to edit the CSV’s content. I went through several iterations of this, which is why the flow diagram (Figure 1) has a loop at the end.

*Figure 1. ETL flow diagram*

**

Once the data is downloaded and cleaned, it can be analyzed (Figure 2). I first ran Table to Table to move the CSV into my geodatabase. I also set the column data types. Then I could run Summary Statistics to average out the past 30 days of data for each station, and group the averages by station. I used this condensed table when running XY Table to Point to get a feature layer of the stations as points, with the averages as attributes. Finally, the point layer could be used to run through the interpolation tools. I used both the Geostatistical Wizard and ArcPy code to produce the interpolation. The Wizard allowed more control over the additional factors, like elevation or weighting, than the code.

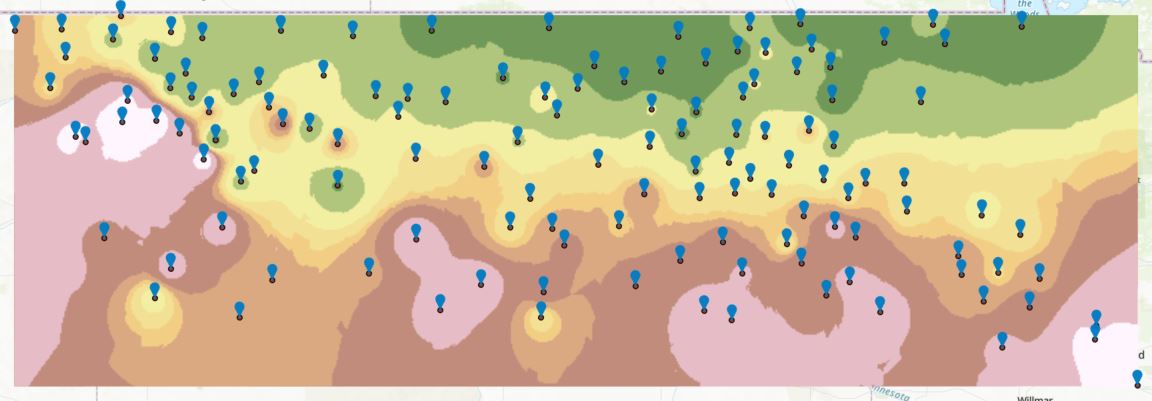
*Figure 2. Data analysis process flow diagram*

**

**Results**

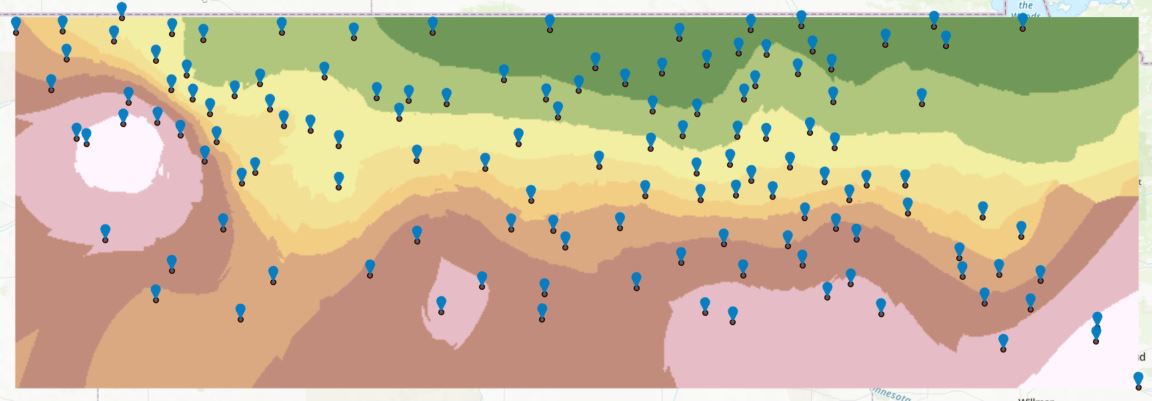
The results for Inverse Distance Weighted interpolation (Figure 3) show bands of temperatures interrupted by pockets of disruption where new data, in the form of a station, changes the map. This is because with IDW, individual known values have a lot of weight on the area immediately around them.

*Figure 3. Inverse Distance Weighted interpolation output*

**

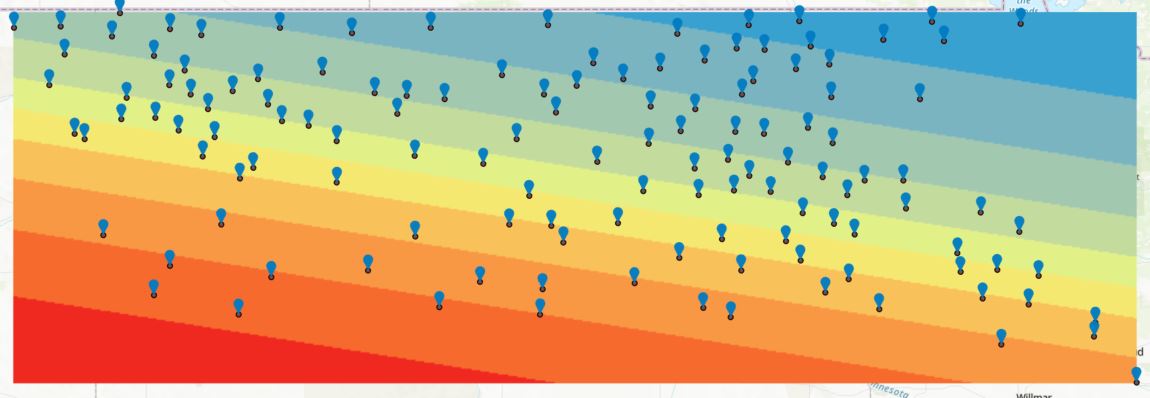
The results for kriging interpolation (Figure 4) show averages that create relatively steady bands of average temperatures which change as they move from south to north, with some pockets of different values. Kriging predicts likelihood of a value based on neighboring points. The influence of neighboring points eventually levels off. In Figure 4, the number of neighbors used to calculate values in the tool was 15.

*Figure 4. Kriging output*

**

The results for Global Polynomial Interpolation (Figure 5) show straight bands of gradually changing temperature. This is the expected result for GPI, as it produces smooth surfaces. This interpolation seems more rigid but it also produces more color bands to show the gradual change.

*Figure 5. Global Polynomial Interpolation output*

**

**Results Verification**

Since we are averaging from a bunch of averages, the data cannot really be correct. Also, the reason we have interpolation options is because none of them are perfect. That said, we have a good number of data points, and the output appears reasonable (temperature cools as you move north). I do not trust anything in the far corners of the maps, since those are the areas with the least comprehensive input.

**Discussion and Conclusion**

I learned how to pull current data with an ETL, and all about the interpolation options and the math or logic behind them. For my third interpolation option, I picked Global Polynomial mostly because it was the most visually distinct output. However, when reading more about kriging and IDW, I could see how my outputs conformed to the information in the documentation.

The general consensus (Esri, 2019) seems to be that kriging is best for temperature data. IDW was presented as a secondary strong option. My preference is for IDW because I liked that the output represented islands of data points. That said, kriging utilizes important factors like spatial arrangement, so I understand why it is useful. Any of the tools that incorporate elevation are also preferable for temperature data.

**References**

*AASJournals/Tutorials*. (n.d.). GitHub. Retrieved April 11, 2021, from <https://github.com/AASJournals/Tutorials>

Allen, D. (2020, September 20). *How to create a Pandas Dataframe from an API Endpoint in a Jupyter Notebook*. Medium. <https://deallen7.medium.com/how-to-create-a-pandas-dataframe-from-an-api-endpoint-in-a-jupyter-notebook-f2561f766ca3>

*Classification trees of the interpolation methods offered in Geostatistical Analyst—ArcMap | Documentation*. (n.d.). Retrieved April 11, 2021, from <https://desktop.arcgis.com/en/arcmap/latest/extensions/geostatistical-analyst/classification-trees-of-the-interpolation-methods-offered-in-geostatistical-analyst.htm>

GeoDelta Labs. (2019, June 4). *Create a Temperature Distribution Map using Spatial Interpolation with ArcGIS*. <https://www.youtube.com/watch?v=OGaMXQtbUQE>

*How inverse distance weighted interpolation works—ArcGIS Pro | Documentation*. (n.d.). Retrieved April 11, 2021, from <https://pro.arcgis.com/en/pro-app/latest/help/analysis/geostatistical-analyst/how-inverse-distance-weighted-interpolation-works.htm>

*How to interpolate weather data*. (2019, June 21). Esri Community. <https://community.esri.com/t5/science-questions/how-to-interpolate-weather-data/m-p/849539#M93>

Jeff Bishop. (2021). *NDAWN\_Weather\_Scraper*. GitHub.

*Kriging in Geostatistical Analyst—ArcGIS Pro | Documentation*. (n.d.). Retrieved April 11, 2021, from <https://pro.arcgis.com/en/pro-app/latest/help/analysis/geostatistical-analyst/kriging-in-geostatistical-analyst.htm>

*qgis—How to interpolate temperature correctly?* (n.d.). Geographic Information Systems Stack Exchange. Retrieved April 11, 2021, from <https://gis.stackexchange.com/questions/30905/how-to-interpolate-temperature-correctly>

*Understanding simple kriging—ArcGIS Pro | Documentation*. (n.d.). Retrieved April 11, 2021, from <https://pro.arcgis.com/en/pro-app/latest/help/analysis/geostatistical-analyst/understanding-simple-kriging.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 | **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 verification is clearly stated **(5 points)**. | 20 | **20** |
|  |  | 100 | **100** |