Aidan Erickson
Diga 620
4/26/25

NOAA Weather API Demo Report

The end goal of this report was to investigate the feasibility and scalability of providing NOAA historical weather data to support farmers with interactive dashboards so they can use this information to make meaningful decisions. This report looks into the data completeness, monthly precipitation, and temperatures. By providing this historical data and its geographical location, we predict that farmers will be able to gather insights on how the weather will affect their yields and crop cycles during a given timeframe for a specific area.

This report contains three months of historical weather data in North Carolina from 6/1/2019-8/31/2019. This historical weather data was pulled via an API from:

- https://www.ncdc.noaa.gov/cdo-web/webservices/v2.

This report utilized manual queries to identify and record the latitude, longitude, elevation, station name, county, state, zip code that were saved into a separate GeoJSON file. The data was gathered manually from:

- https://www.ncdc.noaa.gov/cdo-web/datasets/GHCND/stations/

The first step in this process was pulling the correct data from the NOAA API using a key granted by the Government via email request. I put together four API queries to return the StationID, PRCP (precipitation), temperature minimum & maximum, start and end dates. The API queries pulled this data by matching the following zip codes from North Carolina (27601, 28801, 27260, 28516). A python “for” loop was created to pull these four APIs and transform their appended data into a JSON file. The count of the number of rows in this table was 1,823.

Figure 1: JSON file output


```
{
  "metadata": {
    "resultset": {
      "offset": 1,
      "count": 276,
      "limit": 1000
    }
  },
  "results": [
    {
      "date": "2019-06-01T00:00:00",
      "datatype": "PRCP",
      "station": "GHCND:USC00317079",
      "attributes": ",,7,0800",
      "value": 1.15
    },
    {
      "date": "2019-06-01T00:00:00",
      "datatype": "TMAX",
      "station": "GHCND:USC00317079",
      "attributes": ",,7,0800",
      "value": 89.0
```

The second GeoJSON file and third Facts excel file were manually created by searching the Station_IDs on the NOAA website. The GeoJSON file was manually converted from a list of values by entering the created CSV into:

- https://www.convertcsv.com/csv-to-geojson.htm

Figure 2: GeoJSON output

```
{
  "type": "FeatureCollection",
  "features": [
  {
    "type": "Feature",
    "geometry": {
       "type": "Point",
       "coordinates":  [ -78.78,35.89 ]
    },
    "properties": {
    "StationId":"GHCND:USC00317079"
    }
  },
  {
    "type": "Feature",
    "geometry": {
       "type": "Point",
       "coordinates":  [ -82.54,35.43 ]
    },
    "properties": {
    "StationId":"GHCND:USW00013872"
```

The fourth step in this process was loading each of these three files in Excel Power Query and merging them together based on Station_IDs. Before the NOAA JSON files could be merged with the other two, it needed to have it's "datatypes" and "values" columns split into three columns for each "datatype" (PRCP, TMAX, TMIN). These three separate tables could then be joined using a LEFT JOIN based on the Station_ID, DATE, & "datatype" columns. This was done three times for each table. Then this new NOAA table could be merged with the GEOJSON data and Address Facts based on the Station_Id, Using a LEFT JOIN. Now all the data was correctly merged into a single table. The final count of rows in this merged table was 1823, matching the initial count of the NOAA API file. This means that no records were lost during the join and merge process. There were no truncated values in this merge process. The following two figures are visualizations of this process.

Figure 3: Tables

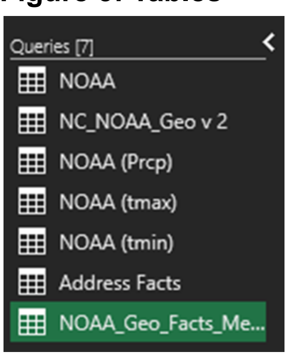


Figure 4: Table Merge Process

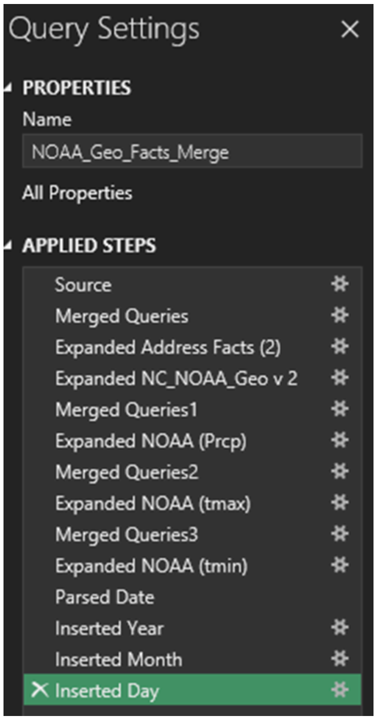
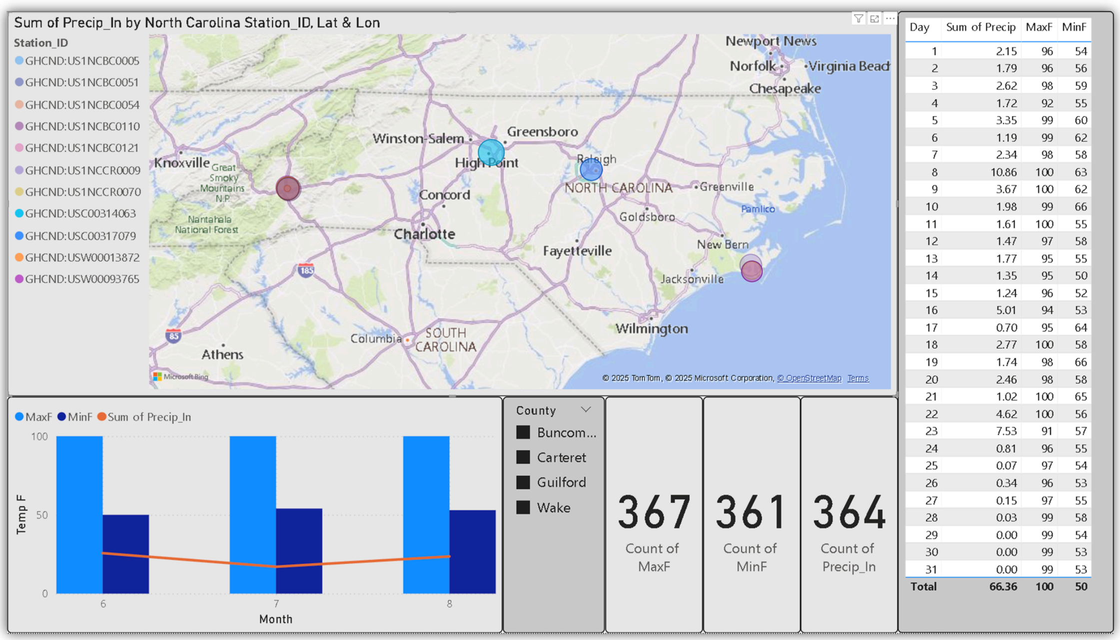
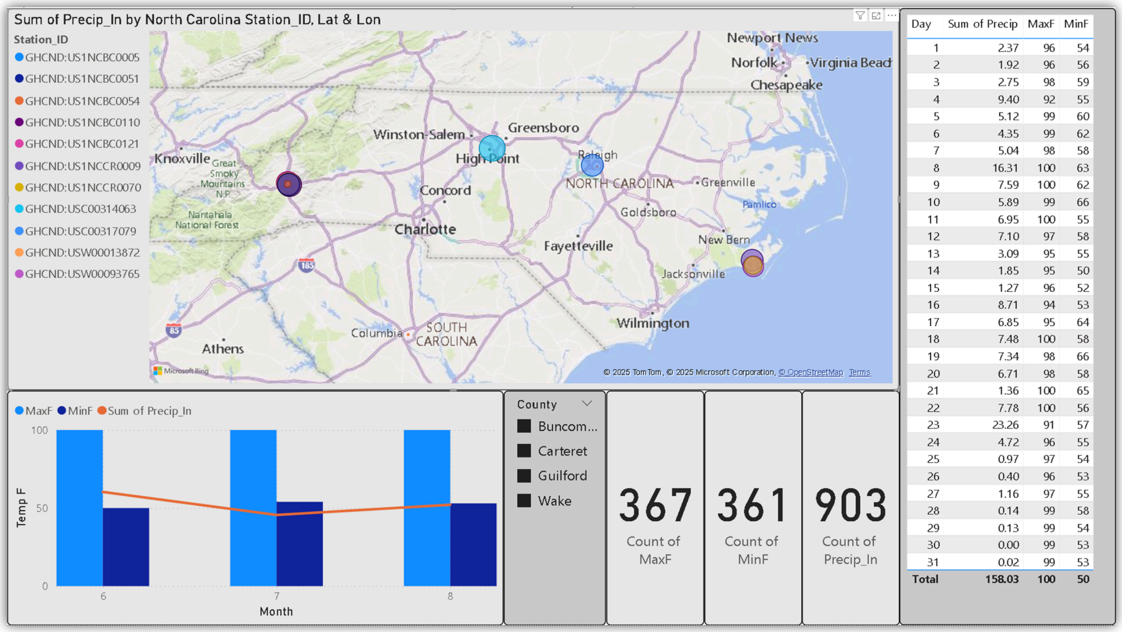


Figure 5: Power BI Visual focus on four Stations



This first figure is a visualization of the data in PowerBI for Station_IDs with close to complete data. These four stations have 1092 rows, with the three counts combined being 992, there are 100 values missing. Visually the Min and Max temperatures are within the expected range for the Summer months in NC. Going forward as this report is expanded, Power BI should be utilized for visually displaying the coordinates of the stations. The size of the bubbles in this visualization are showing the sum of precipitation.

Figure 6: Power BI Visual all Station_IDs



This 6th figure, visually shows the difference in counts for the 11 Station_IDs. Seven of these stations lacked any temperature measurements to show the trend of min and max temperatures. There are 1571 values when including all the Station_IDs. Some information is filtered out for the sake of visual representation, this includes Station_IDs that did not have a latitude or longitude value. Data completeness needs to be reviewed with each Station_Id to ensure that the reporting is not being skewed. The sum of precipitation across the 11 Stations matches the merged Excel files Sum, meaning the data was loaded correctly into PowerBI.

The scalability of this report is feasible as long as each Station_ID is reviewed for data completeness. There's a limit of 10,000 requests a day or 5 requests per second. The data is secure, and needs a key to be collected. Data integrity should be reviewed after the API pulls data in from the NOAA website. To improve upon this report the data should be transformed and follow the following Data Warehouse Target Schema guidelines:

Figure 7&8: Data Warehouse Target Schema

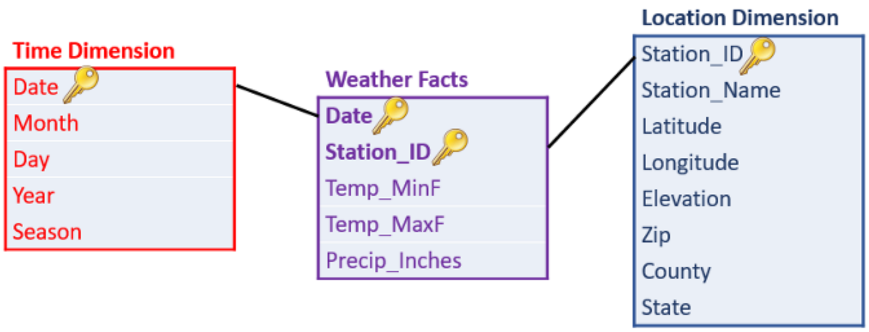
Target_Table	Target_Field	Target_FieldType	Target_FieldDomain	Target_FieldFormat	Source_Table	Source_Field	Transformation	Notes
TimeDimensio	Date	Date		YYYY-MM-DD	NOAA API	date	PowerQuery add Date(YYYY-MM-DD)	Date of measurement, foreign key to Time Dimension
	Month	Int	Range 1-12		NOAA API	date	PowerQuery add Month	Station id of Measurement, Foreign Key to Location Dimension
	Day	Int	Range 1-31		NOAA API	date	PowerQuery add Day	Min daily tempo in degree Farenheight
	Year	Int		YYYY	NOAA API	date	PowerQuery add Year	Max daily tempo in degree Farenheight
	Season	Text	Possible Values = WINTER, SPRING, SUMMER, or FALL	ALL CAPS	Calculated	Calculated	Calculate based on months, 6-8 =summer	Daily precipitation in inches (to nearest hundreth)

Target_Table	Target_Field	Target_FieldType	Target_FieldDomain	Target_FieldFormat	Source_Table	Source_Field	Transformation	Notes
	Station ID	Text	Length=20	ALL CAPS	NOAA API	station	=upper	Station id, Primary Key
	Station Name	Text	Length=50	ALL CAPS	AddressFacts	Station_Name	No edit needed	Station name
	Latitude	Decimal		#.####	GeoJSON	Lat	Round to 1000ths	Latitude of Station in decimal degrees
	Longitude	Decimal		#.####	GeoJSON	Lon	Round to 1000ths	Longitude of Station in decimal degrees
	Elevation	Decimal		#.#	AddressFacts	Elevation	Convert FT to Meters, then round to 10ths place	Elevation above mean sea level of station in meters (to nearest tenth of a meter)
	Zip	Text	Length=5	ALL CAPS	AddressFacts	Zipcode	No edit needed	5-digit zip code
	County	Text	Length=20	ALL CAPS	AddressFacts	County	=upper	County where station is located
	State	Text	Length=2	ALL CAPS	AddressFacts	State	No edit needed	2 letter abberviation of State

Target_Table	Target_Field	Target_FieldType	Target_FieldDomain	Target_FieldFormat	Source_Table	Source_Field	Transformation	Notes
WeatherFacts	Date	Date		YYYY-MM-DD	NOAA API	date	Create custom date format	Full date, Primary Key
WeatherFacts	Station_ID	Text			NOAA API	station	no change needed	Month component of date
WeatherFacts	Temp_MinF	Int		#	NOAA API	value	Filter to just MinF, create new table in Power Query, Left Outer merge based on date, ID, and MinF. This creates a new column	Day of month component
WeatherFacts	Temp_MaxF	Int		#	NOAA API	value	Filter to just MinF, create new table in Power Query, Left Outer merge based on date, ID, and MaxF. This creates a new column	Year component of date
WeatherFacts	Precip Inches	Int		#.##	NOAA API	value	Filter to just MinF, create new table in Power Query, Left Outer merge based on date, ID, and PRCP. This creates a new column	Season of year (Summer 6/1-8/31)

It is important to adhere to this documented transformation process so the team knows which steps to take, and to ensure that the end product is legible by stakeholders. For this initial report demonstration not all values adhere to the target schema. The initial API that pulls the NOAA data by zip code is efficient, and takes little to no time. The 1823 rows are pulled and converted into a JSON file in under 10 seconds. Primary keys will need to be established and the files should reflect the following Star Schema:

Figure 9: Star Schema



Ethical and legal considerations should be accounted for before presenting this data to stakeholders. Legally this data is publicly provided through the use of private keys. This project should adhere to legal data standards that the Federal Government put in place. Ethically, we should not abuse the amount of keys we request and the amount of data we pull at one time. Data integrity and quality should be reviewed, to mitigate bias and false reporting. It is important to validate the data so it is not skewed and missing large amounts of certain values.

This project is feasible and scalable to include more Station IDs from the rest of the USA. Automation of the GeoJSON and AddressFacts files will be needed before more Station_IDs are pulled via API. If the project workflow is followed as it is, the API Python pull will need to be refreshed or set on a certain timeline. Power Query will need to be refreshed, and a live connection should be made to PowerBI. Improvements will need to be made on the GeoJSON end, to ensure that the data is merged correctly. The biggest concern with this process is data completeness from the NOAA resource. As shown in Figure 6, certain stations are on top of one another and lack crucial values to make meaningful conclusions.

References:

National Oceanic and Atmospheric Administration (NOAA). (n.d.). *Data sharing directive* (Version 3.0). Retrieved April 24, 2025, from https://nosc.noaa.gov/EDMC/documents/Data_Sharing_Directive_v3.0.pdf

National Centers for Environmental Information (NCEI). (n.d.). *Find a station*. National Oceanic and Atmospheric Administration. Retrieved April 24, 2025, from https://www.ncdc.noaa.gov/cdo-web/datatools/findstation

ConvertCSV. (n.d.). *CSV to GeoJSON converter*. Retrieved April 24, 2025, from https://www.convertcsv.com/csv-to-geojson.htm