

```
In [21]: import arcpy
import geopandas as gpd
import pandas as pd
from shapely.geometry import Point
from urllib.request import urlopen as retrieve
import datetime
import dateutil.relativedelta
```

```
In [22]: # Set workspace
arcpy.env.workspace = r'E:\ArcGIS_1\Lab3\NDAWN'
wksp = arcpy.env.workspace
```

```
In [23]: # Retrieve today's date
now = datetime.datetime.now()
end_date = str(now)[0:10]

# Calculate what date was a month ago based on today's date
month_ago = now + dateutil.relativedelta.relativedelta(months=-1)
begin_date = str(month_ago)[0:10]
```

```

In [24]: # Retrieve data from NDAWN
url_max = r'https://ndawn.ndsu.nodak.edu/table.csv?station=78&station=111&station=98&station=174&station=142&station=138&station=161&station=9&station=10&station=118&station=56&station=11&station=12&station=58&station=13&station=84&station=55&station=7&station=87&station=14&station=15&station=96&station=16&station=137&station=124&station=143&station=17&station=85&station=140&station=134&station=18&station=136&station=65&station=104&station=99&station=19&station=129&station=20&station=101&station=81&station=21&station=97&station=22&station=75&station=2&station=172&station=139&station=23&station=62&station=86&station=24&station=89&station=126&station=93&station=90&station=25&station=83&station=107&station=156&station=77&station=26&station=70&station=127&station=27&station=132&station=28&station=29&station=30&station=31&station=102&station=32&station=119&station=4&station=80&station=33&station=59&station=105&station=82&station=34&station=72&station=135&station=35&station=76&station=120&station=141&station=109&station=36&station=79&station=71&station=37&station=38&station=39&station=130&station=73&station=40&station=41&station=54&station=69&station=113&station=128&station=42&station=43&station=103&station=116&station=88&station=114&station=3&station=163&station=64&station=115&station=67&station=44&station=133&station=106&station=100&station=121&station=45&station=46&station=61&station=66&station=74&station=60&station=125&station=8&station=47&station=122&station=108&station=5&station=152&station=48&station=68&station=49&station=50&station=91&station=117&station=63&station=150&station=51&station=6&station=52&station=92&station=112&station=131&station=123&station=95&station=53&station=57&station=149&station=148&station=110&variable=ddmxt&year=2022&ttype=daily&quick_pick=30_d&begin_date=' + begin_date + '&end_date=' + end_date
retrieve(url_max, 'temperature_max.csv')

url_min = r'https://ndawn.ndsu.nodak.edu/table.csv?station=78&station=111&station=98&station=174&station=142&station=138&station=161&station=9&station=10&station=118&station=56&station=11&station=12&station=58&station=13&station=84&station=55&station=7&station=87&station=14&station=15&station=96&station=16&station=137&station=124&station=143&station=17&station=85&station=140&station=134&station=18&station=136&station=65&station=104&station=99&station=19&station=129&station=20&station=101&station=81&station=21&station=97&station=22&station=75&station=2&station=172&station=139&station=23&station=62&station=86&station=24&station=89&station=126&station=93&station=90&station=25&station=83&station=107&station=156&station=77&station=26&station=70&station=127&station=27&station=132&station=28&station=29&station=30&station=31&station=102&station=32&station=119&station=4&station=80&station=33&station=59&station=105&station=82&station=34&station=72&station=135&station=35&station=76&station=120&station=141&station=109&station=36&station=79&station=71&station=37&station=38&station=39&station=130&station=73&station=40&station=41&station=54&station=69&station=113&station=128&station=42&station=43&station=103&station=116&station=88&station=114&station=3&station=163&station=64&station=115&station=67&station=44&station=133&station=106&station=100&station=121&station=45&station=46&station=61&station=66&station=74&station=60&station=125&station=8&station=47&station=122&station=108&station=5&station=152&station=48&station=68&station=49&station=50&station=91&station=117&station=63&station=150&station=51&station=6&station=52&station=92&station=112&station=131&station=123&station=95&station=53&station=57&station=149&station=148&station=110&variable=ddmnt&year=2022&ttype=daily&quick_pick=30_d&begin_date=' + begin_date + '&end_date=' + end_date
retrieve(url_min, 'temperature_min.csv')

```

```
Out[24]: ('temperature_min.csv', <http.client.HTTPMessage object at 0x0000029306E483D0>)
```

```
In [26]: # Read the csv file as a DataFrame and calculate the monthly average maximum temperature for each station
temp_max = pd.read_csv("temperature_max.csv", skiprows = [0, 1, 2, 4])
temp_max.drop(['Elevation', 'Year', 'Month', 'Day', 'Max Temp Flag'], axis=1)
temp_max = temp_max.groupby('Station Name').agg({'Latitude': 'first', 'Longitude': 'first', 'Max Temp': 'mean' })

# Convert the data to a GeoDataFrame
temp_max['geometry'] = temp_max.apply(lambda x: Point((float(x.Longitude), float(x.Latitude))), axis=1)
temp_max_gdf = gpd.GeoDataFrame(temp_max, geometry = 'geometry')

# Create a shapefile and define its spatial reference
temp_max_gdf.to_file('Stations_max.shp', driver='ESRI Shapefile')
arcpy.management.DefineProjection("Stations_max.shp", 'GEOGCS["GCS_WGS_1984", DATUM["D_WGS_1984", SPHEROID["WGS_1984", 6378137.0, 298.257223563]], PRIMEM["Greenwich", 0.0], UNIT["Degree", 0.0174532925199433]]')
```

[26]:11: UserWarning: Column names longer than 10 characters will be truncated when saved to ESRI Shapefile.

Out[26]:

Messages

```
In [27]: # Read the csv file as a DataFrame and calculate the monthly average minimum
          temperature for each station
temp_min = pd.read_csv("temperature_min.csv", skiprows = [0, 1, 2, 4])
temp_min.drop(['Elevation', 'Year', 'Month', 'Day', 'Min Temp Flag'], axis=1)
temp_min = temp_min.groupby('Station Name').agg({'Latitude': 'first', 'Longitude': 'first', 'Min Temp': 'mean' })

# Convert the data to a GeoDataFrame
temp_min['geometry'] = temp_min.apply(lambda x: Point((float(x.Longitude), float(x.Latitude))), axis=1)
temp_min_gdf = gpd.GeoDataFrame(temp_min, geometry = 'geometry')

# Create a shapefile and define its spatial reference
temp_min_gdf.to_file('Stations_min.shp', driver='ESRI Shapefile')
arcpy.management.DefineProjection("Stations_min.shp", 'GEOGCS["GCS_WGS_1984", DATUM["D_WGS_1984", SPHEROID["WGS_1984", 6378137.0, 298.257223563]], PRIMEM["Greenwich", 0.0], UNIT["Degree", 0.0174532925199433]]')
```

Out[27]:

Messages

Interpolation

```
In [28]: # IDW, alpha = 2, # variables = 5
arcpy.ddd.Idw("Stations_max.shp", "Max Temp", "IDW_max.tif", 0.01447904, 2, "VARIABLE 12", None)
arcpy.ddd.Idw("Stations_min.shp", "Min Temp", "IDW_min.tif", 0.01447904, 2, "VARIABLE 12", None)

# GPI, order = 2
arcpy.ga.GlobalPolynomialInterpolation("Stations_max.shp", "Max Temp", None, "GPI_max.tif", 0.01447904, 2, None)
arcpy.ga.GlobalPolynomialInterpolation("Stations_min.shp", "Min Temp", None, "GPI_min.tif", 0.01447904, 2, None)

# Krigin ordinary, semi-variogram model = spherical
arcpy.ddd.Kriging("Stations_max.shp", "Max Temp", "Kriging_Ordinary_max.tif", "Spherical 0.014479 # # #", 0.01447904, "VARIABLE 12", None)
arcpy.ddd.Kriging("Stations_min.shp", "Min Temp", "Kriging_Ordinary_min.tif", "Spherical 0.014479 # # #", 0.01447904, "VARIABLE 12", None)

# Krigin universal, semi-variogram model = linear drift
arcpy.ddd.Kriging("Stations_max.shp", "Max Temp", "Kriging_Universal_max.tif", "LinearDrift 0.014479 # # #", 0.01447904, "VARIABLE 12", None)
arcpy.ddd.Kriging("Stations_min.shp", "Min Temp", "Kriging_Universal_min.tif", "LinearDrift 0.014479 # # #", 0.01447904, "VARIABLE 12", None)
```

Out[28]:

Messages