

```
In [ ]: import requests
import os
import pandas
import geopandas
from shapely.geometry import Point
```

Loop Through for Avg, Max, and Min Temps

```
In [ ]: #Define URL from NDAWN API
variablelist = ['ddavt','ddmxt','ddmnt']
#Loop through variables
for temp in variablelist:
    ndurl = fr'https://ndawn.ndsu.nodak.edu/table.csv?station=78&station=111&station=98&station=162&station=174&station=142&station=164&station=138&station=161&station=9&station=160&station=159&station=108'
    #Define Name and Location of New CSV
    ndoutput = fr'\\Mac\Home\Documents\ArcGIS\Projects\Lab3-2\Lab3Data\NDAWN_30Day_{temp}_Raw.csv'
    #Pulls CSV from URL
    ndresponse = requests.get(ndurl, stream=True)
    #Writing CSV to file on local drive if status code is correct.
    if ndresponse.status_code == 200:
        with open(ndoutput, 'wb') as file:
            file.write(ndresponse.content)

    #Reads CSV file
    nddataframe = pandas.read_csv(fr'\\Mac\Home\Documents\ArcGIS\Projects\Lab3-2\Lab3Data\NDAWN_30Day_{temp}_Raw.csv', skiprows=4, index_col=False)
    #Defines new names for CSV Headers
    new_names = ['Location', 'Lat', 'Long', 'Elevation ft', 'Year', 'Month', 'Day', f'{temp}', 'Flag']
    nddataframe.columns = new_names
    #Saves New Names to CSV
    nddataframe.to_csv(fr'\\Mac\Home\Documents\ArcGIS\Projects\Lab3-2\Lab3Data\NDAWN_30Day_{temp}.csv', index=False)

    # Read the CSV file into a DataFrame, Information on how to do the groupby provided by OpenAI,2023
    nddataframe = pandas.read_csv(fr'\\Mac\Home\Documents\ArcGIS\Projects\Lab3-2\Lab3Data\NDAWN_30Day_{temp}.csv')

    # Group the columns with the same station names and average the temp
    ndresult = nddataframe.groupby(['Lat', 'Long', 'Location'])[f'{temp}'].mean().reset_index()

    #Save to CSV
    ndresult.to_csv(fr'\\Mac\Home\Documents\ArcGIS\Projects\Lab3-2\Lab3Data\NDAWN_30Day_{temp}_Avg.csv', index=False)
    #Read CSV
    ndresult = pandas.read_csv(fr'\\Mac\Home\Documents\ArcGIS\Projects\Lab3-2\Lab3Data\NDAWN_30Day_{temp}_Avg.csv')
    #Create points with Lat and long coordinates
    ndgeometry = [Point(xy) for xy in zip(ndresult['Long'], ndresult['Lat'])]
    ndgeoframe = geopandas.GeoDataFrame(ndresult, geometry=ndgeometry, crs='4326')
    #Write geometry to shapefile
    ndgeoframe.to_file(fr'\\Mac\Home\Documents\ArcGIS\Projects\Lab3-2\Lab3Data\ND30Day_{temp}.shp')
```

Loop Through Points for Different Interpolation Methods

```
In [ ]: #List of Temp Variables
variablelist = ['ddavt','ddmxt','ddmnt']
#Loop through Variables
for temp in variablelist:
    #Inverse Distance Weighting
    arcpy.ddd.Idw(
        in_point_features=fr'\\Mac\Home\Documents\ArcGIS\Projects\Lab3-2\Lab3Data\ND30Day_{temp}.shp',
        z_field=temp,
        out_raster=fr'\\Mac\Home\Documents\ArcGIS\Projects\Lab3-2\Lab3-2.gdb\IDW_{temp}',
        power=2,
        in_barrier_polyline_features=None
    )
    #Empirical Bayesian Kriging
    arcpy.ga.EmpiricalBayesianKriging(
        in_features=fr'\\Mac\Home\Documents\ArcGIS\Projects\Lab3-2\Lab3Data\ND30Day_{temp}.shp',
        z_field=temp,
        out_ga_layer=None,
        out_raster=fr'\\Mac\Home\Documents\ArcGIS\Projects\Lab3-2\Lab3-2.gdb\Kriging_{temp}',
    )
    #Diffusion Interpolation
    arcpy.ga.DiffusionInterpolationWithBarriers(
        in_features=fr'\\Mac\Home\Documents\ArcGIS\Projects\Lab3-2\Lab3Data\ND30Day_{temp}.shp',
        z_field=temp,
        out_raster=fr'\\Mac\Home\Documents\ArcGIS\Projects\Lab3-2\Lab3-2.gdb\Diffusion_{temp}',
        number_iterations=100,
    )
```