In [21]:
```
import arcpy
import requests
import pandas as pd
import numpy as np
import csv
arcpy.env.workspace = r'G:\My Drive\GIS 5571\Lab3\lab3\lab3.gdb'
working_dir = r'G:\My Drive\GIS 5571\Lab3\lab3\lab3.gdb'
```

ETL

In [2]:
```
#url = 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&stati
on=118&station=56&station=11&station=12&station=58&station=13&station=84&stati
on=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&sta
tion=18&station=136&station=65&station=104&station=99&station=19&station=129&s
tation=20&station=101&station=81&station=21&station=97&station=22&station=75&s
tation=2&station=172&station=139&station=23&station=62&station=86&station=24&s
tation=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=13
2&station=28&station=29&station=30&station=31&station=102&station=32&station=1
19&station=4&station=80&station=33&station=59&station=105&station=82&station=3
4&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&statio
n=128&station=42&station=43&station=103&station=116&station=88&station=114&sta
tion=3&station=163&station=64&station=115&station=67&station=44&station=133&st
ation=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=1
49&station=148&station=110&variable=ddavt&year=2022&ttype=daily&quick_pick=30_
d&begin_date=2022-11-15&end_date=2022-11-15"
#file1 = r'G:\My Drive\GIS 5571\Lab3\lab3\NDAWN.csv'
```

In [3]:
```
#data = requests.get(url)
#data1c = data.text.splitlines()
```

In [5]:
```
#with open(file1, mode = 'w', newline = '') as data_file:
#         data1Read = csv.reader(data1c)
#         data1Write = csv.writer(data_file)
#         line_count = 0
#         header = ['Station Name', 'Latitude', 'Longitude', 'Elevation', 'Yea
r', 'Month' , 'Day' , 'Avg Temp']
#         data1Write.writerow(header)
#         for row in data1Read:
#                  if (line_count > 4):
#                           data1Write.writerow(row)
#                  line_count += 1
```

In [ ]:
```
#dataframe = pd.read_csv(file1, index_col=False)
```

In [10]:
```
#new_df = pd.DataFrame(columns = ['Station_name', 'Latitude', 'Longitude', 'Av
g_Temp_Month'])
#uniq = dataframe['Station Name'].unique()
#for i in range(len(uniq)):
#    all_val = dataframe.loc[dataframe['Station Name'] == uniq[i]]
#    Station_name = all_val.iloc[0]['Station Name']
#    Latitude = all_val.iloc[0]['Latitude']
#    Longitude = all_val.iloc[0]['Longitude']
#    Avg_temp = all_val['Avg Temp'].mean()

#    new_df.loc[len(new_df.index)] = [Station_name, Latitude, Longitude, Avg_t
emp]

#new_df.to_csv('NDAWN_30_Day_Avgs.csv')
```

Create Points

In [ ]:
```
#arcpy.management.XYTableToPoint("NDAWN_30_Day_Avgs.csv", r"G:\My Drive\GIS 55
71\Lab3\Lab3\lab3.gdb\Points", "Longitude", "Latitude", "Avg_Temp_Month", 'GEO
GCS["GCS_WGS_1984",DATUM["D_WGS_1984",SPHEROID["WGS_1984",6378137.0,298.257223
563]],PRIMEM["Greenwich",0.0],UNIT["Degree",0.0174532925199433]],VERTCS["WGS_1
984",DATUM["D_WGS_1984",SPHEROID["WGS_1984",6378137.0,298.257223563]],PARAMETE
R["Vertical_Shift",0.0],PARAMETER["Direction",1.0],UNIT["Meter",1.0]];-400 -40
0 1000000000;-100000 10000;-100000 10000;8.98315284119521E-09;0.001;0.001;IsHi
ghPrecision')
```

IDW

In [ ]:
```
#arcpy.ddd.Idw("Points", "Avg_Temp_Month", r"G:\My Drive\GIS 5571\Lab3\Lab3\la
b3.gdb\Idw", 0.0144790399999999, 2, "VARIABLE 12", None)
```

TIN

In [ ]:
```
#arcpy.ddd.CreateTin(r"G:\My Drive\GIS 5571\Lab3\Lab3\TIN", 'GEOGCS["GCS_WGS_1
984",DATUM["D_WGS_1984",SPHEROID["WGS_1984",6378137.0,298.257223563]],PRIMEM
["Greenwich",0.0],UNIT["Degree",0.0174532925199433]],VERTCS["WGS_1984",DATUM
["D_WGS_1984",SPHEROID["WGS_1984",6378137.0,298.257223563]],PARAMETER["Vertica
l_Shift",0.0],PARAMETER["Direction",1.0],UNIT["Meter",1.0]]', "Points Avg_Temp
_Month Mass_Points <None>", "DELAUNAY")
```

Kriging

In [ ]:
```
#out_surface_raster = arcpy.sa.Kriging("Points", "Avg_Temp_Month", "Spherical
 # # # #", 0.0144790399999999, "VARIABLE 12", None); out_surface_raster.save
(r"G:\My Drive\GIS 5571\Lab3\Lab3\lab3.gdb\Kriging")
```

Accuracy Assessment

In [ ]:
```
#IDW
```

In [ ]:
```
#count = 1
#errors = []
#predict = []
#actual = []
#with arcpy.da.SearchCursor(data, ['Latitude', 'Longitude', 'Ave_Temp']) as cu
rsor:
#    for row in cursor:
#        count += 1
#        lat = row[0]
#        long = row[1]
#       Avg_temp = row[2]

#       selection_string = "Latitude <>" + str(lat) + "And Longitude <>" +str
(lon)
#        subset = arcpy.analysis.Select(data, r"G:\My Drive\GIS 5571\Lab3\lab3
\lab3.gdb")

#        idw = arcpy.ga.IDW(subset, "Avg_Temp", "Idw", r"G:\My Drive\GIS 5571
\Lab3\lab3\lab3.gdb\Idw_Raster")

#        loc_str = str(lon) + "" + str(lat)
#        val = arcpy.management.GetCellValue(r"G:\My Drive\GIS 5571\Lab3\lab3
\lab3.gdb\Idw_Raster", loc_str)

#       predict.append(float(val.getOutput(0)))
#        actual.append(Avg_Temp)
#        errors.append(Ave_Temp - float(val.getOutput(0)))
```

In [18]:
```
#errors_2 = np.square(errors)
#RMSE= np.mean(errors_2)
#print(RMSE)
```

```
0.4328156
```

In [ ]:
```
#TIN
```

In [ ]:
```
#count = 1
#errors = []
#predict = []
#actual = []
#with arcpy.da.SearchCursor(data, ['Latitude', 'Longitude', 'Ave_Temp']) as cu
rsor:
#    for row in cursor:
#        count += 1
#        lat = row[0]
#        long = row[1]
#       Avg_temp = row[2]

#       selection_string = "Latitude <>" + str(lat) + "And Longitude <>" +str
(lon)
#        subset = arcpy.analysis.Select(data, r"G:\My Drive\GIS 5571\Lab3\lab3
\lab3.gdb")

#        idw = arcpy.ga.IDW(subset, "Avg_Temp", "Idw", r"G:\My Drive\GIS 5571
\Lab3\lab3\lab3.gdb\TIN_Raster")

#        loc_str = str(lon) + "" + str(lat)
#        val = arcpy.management.GetCellValue(r"G:\My Drive\GIS 5571\Lab3\lab3
\lab3.gdb\TIN_Raster", loc_str)

#       predict.append(float(val.getOutput(0)))
#        actual.append(Avg_Temp)
#        errors.append(Ave_Temp - float(val.getOutput(0)))
```

In [19]:
```
#errors_2 = np.square(errors)
#RMSE= np.mean(errors_2)
#print(RMSE)
```

```
0.6721329
```

In [ ]:
```
#Kriging
```

In [ ]:
```
#count = 1
#errors = []
#predict = []
#actual = []
#with arcpy.da.SearchCursor(data, ['Latitude', 'Longitude', 'Ave_Temp']) as cu
rsor:
#    for row in cursor:
#        count += 1
#        lat = row[0]
#        long = row[1]
#       Avg_temp = row[2]

#       selection_string = "Latitude <>" + str(lat) + "And Longitude <>" +str
(lon)
#        subset = arcpy.analysis.Select(data, r"G:\My Drive\GIS 5571\Lab3\lab3
\lab3.gdb")

#        idw = arcpy.ga.IDW(subset, "Avg_Temp", "Idw", r"G:\My Drive\GIS 5571
\Lab3\lab3\lab3.gdb\Krig_Raster")

#        loc_str = str(lon) + "" + str(lat)
#        val = arcpy.management.GetCellValue(r"G:\My Drive\GIS 5571\Lab3\lab3
\lab3.gdb\Krig_Raster", loc_str)

#       predict.append(float(val.getOutput(0)))
#        actual.append(Avg_Temp)
#        errors.append(Ave_Temp - float(val.getOutput(0)))
```

In [20]:
```
#errors_2 = np.square(errors)
#RMSE= np.mean(errors_2)
#print(RMSE)
```

```
0.3944321
```