

Lab 4: Analyze NDAWN Data

This script is designed to be used following the Get_NDAWN_Data script. This script will take the station locations retrieved from the NDAWN HMTL to point features and join the average daily temperature, minimum daily temperature, and maximum daily temperature to the stations. The temperatures will then be used as sample values for different interpolation methods: inverse distance weighted, radial basis functions, empirical bayesian kriging, and original kriging. The outputs of this script are a point feature class of the stations, and jpeg of the output rasters.

```
In [ ]: import os
```

```
In [ ]: os.chdir("../output data")
data_dir = os.getcwd()
```

```
In [ ]: arcpy.env.workspace = r"C:\Users\msong\Desktop\arc21\lab4\lab4_proj\lab4_proj.
gdb"
```

```
In [ ]: # create point feature class from stations coordinates
out_coordinate_system = arcpy.SpatialReference('WGS 1984')
arcpy.management.XYTableToPoint(os.path.join(data_dir, "stations.csv"),
                                "stations",
                                "x",
                                "y",
                                None,
                                out_coordinate_system)
```

```
In [ ]: # create a table of the ndawn data csv
arcpy.management.CopyRows(os.path.join(data_dir, "ndawn_data.csv"),
                           "ndawn_data")
```

```
In [ ]: # get avg monthly temperature for each station
arcpy.analysis.Statistics("ndawn_data",
                           "ndawn_avg_monthly",
                           "Avg_Temp__Degrees_F_ MEAN",
                           "Station_Name")
```

```
In [ ]: # create a copy of stations feature class to join ndawn_data to
arcpy.management.CopyFeatures("stations",
                               os.path.join(arcpy.env.workspace, "stations_ndaw
n"))
```

```
In [ ]: # join calculated average mean to station point features
arcpy.management.AddJoin("stations",
                           "station_name",
                           "ndawn_avg_monthly",
                           "station_name",
                           "KEEP_ALL")
```

```
In [ ]: # get high and low temperature for each station
arcpy.analysis.Statistics("ndawn_data",
                          "ndawn_highlow",
                          "Max_Temp__Degrees_F_ MAX;Min_Temp__Degrees_F_ MIN",
                          "Station_Name")
```

```
In [ ]: # join ndawn data to stations to perform interpolations
arcpy.management.AddJoin("stations_ndawn",
                          "station_name",
                          "ndawn_highlow",
                          "Station_Name",
                          "KEEP_ALL")
```

IDW Interpolation

```
In [ ]: # IDW interpolation of max daily temperatures
idw_high = arcpy.sa.Idw("stations_ndawn",
                         "ndawn_highlow.MAX_Max_Temp__Degrees_F_",
                         0.014477764,
                         2,
                         "VARIABLE 12",
                         None);
idw_high.save(r"C:\Users\msong\Desktop\arc21\lab4\lab4_proj\lab4_proj.gdb\idw_high")
```

```
In [ ]: # IDW interpolation of min daily temperatures
idw_low = arcpy.sa.Idw("stations_ndawn",
                       "ndawn_highlow.MIN_Min_Temp__Degrees_F_",
                       0.014477764,
                       2,
                       "VARIABLE 12",
                       None);
idw_low.save(r"C:\Users\msong\Desktop\arc21\lab4\lab4_proj\lab4_proj.gdb\idw_low")
```

Simple Kriging interpolation

```
In [ ]: # kriging interpolation for max daily temps
krig_high = arcpy.sa.Kriging("stations_ndawn",
                              "ndawn_highlow.MAX_Max_Temp__Degrees_F_"
                              ,
                              "Spherical 0.014478 # # #",
                              0.014477764,
                              "VARIABLE 12",
                              None);
krigh_high.save(r"C:\Users\msong\Desktop\arc21\lab4\lab4_proj\lab4_proj.gdb\krig_high")
```

```
In [ ]: # kriging interpolation for min daily temps
krig_low = arcpy.sa.Kriging("stations_ndawn",
                             "ndawn_highlow.MIN_Min_Temp__Degrees_F_",
                             ,
                             "Spherical 0.014478 # # #",
                             0.014477764,
                             "VARIABLE 12",
                             None);
krig_low.save(r"C:\Users\msong\Desktop\arc21\lab4\lab4_proj\lab4_proj.gdb\krig_low")
```

Radial basis functions interpolation

```
In [ ]: # RBF interpolation of max daily temps
arcpy.ga.RadialBasisFunctions("stations_ndawn",
                              "ndawn_highlow.MAX_Max_Temp__Degrees_F_",
                              None,
                              "rbf_high")
```

```
In [ ]: # RBF interpolation of min daily temps
arcpy.ga.RadialBasisFunctions("stations_ndawn",
                              "ndawn_highlow.MIN_Min_Temp__Degrees_F_",
                              None,
                              "rbf_low")
```

Empirical bayesian kriging

```
In [ ]: # EBK interpolation for max daily temps
arcpy.ga.EmpiricalBayesianKriging("stations_ndawn",
                                   "ndawn_highlow.MAX_Max_Temp__Degrees_F_",
                                   None,
                                   "ebk_high")
```

```
In [ ]: # EBK interpolation for min daily temps
arcpy.ga.EmpiricalBayesianKriging("stations_ndawn",
                                   "ndawn_highlow.MIN_Min_Temp__Degrees_F_",
                                   None,
                                   "ebk_low")
```

```
In [ ]: # Export Layout to jpeg
# I did this for each interpolation result
aprx = arcpy.mp.ArcGISProject("CURRENT")
mxd_list = aprx.listLayouts("Layout")[0]

os.chdir("../interpolation results")
data_dir = os.getcwd()
layout = mxd_list.exportToJPEG(os.path.join(data_dir, "EBK_averages"))
```

In []: