Compare the Interpolation Methods

In [1]:

```
# Import the necessary modules
import requests
import pandas as pd
from bs4 import BeautifulSoup
from datetime import date
import arcpy
```

In [2]:

```
# Set the variable for the URL of data page
data = f"https://ndawn.ndsu.nodak.edu/get-table.html?station=78&station=111&station=98&
station=142&station=138&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=17&station=85&station=140&stat
ion=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=139&station=23&station=62&station=86&station=24&station=89&station=126&statio
n=93&station=90&station=25&station=83&station=107&station=77&station=26&station=70&stat
ion=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=8
2&station=34&station=72&station=135&station=35&station=76&station=120&station=141&stati
on=109&station=36&station=79&station=71&station=37&station=38&station=39&station=130&st
ation=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=64&station
=115&station=67&station=44&station=133&station=106&station=100&station=121&station=45&s
tation=46&station=61&station=66&station=74&station=60&station=125&station=8&station=47&
station=122&station=108&station=5&station=48&station=68&station=49&station=50&station=9
1&station=117&station=63&station=51&station=6&station=52&station=92&station=112&station
=131&station=123&station=95&station=53&station=57&station=110&variable=ddavt&year=2021&
ttype=daily&quick_pick=30_d&begin_date={date.today()}&end_date={date.today()}"
```

In [3]:

```
# Get the website data
page = requests.get(data)
```

In [4]:

```
# Build the BeautifulSoup item from the website data
parser = BeautifulSoup(page.text,"html.parser")
```

In [5]:

```
# Find the string includes "Export CSV File"
target = parser.find_all("a", string="Export CSV File")
```

In [6]:

```
# Add the original url in front of the link
for link in target:
    url="https://ndawn.ndsu.nodak.edu"+link.get('href')
```

In [7]:

```
# Get the data and save it as a CSV file
r = requests.get(url, allow_redirects=True)
open('weather.csv', 'wb').write(r.content)
```

Out[7]:

```
202761
```

In [8]:

```
# Start read the CSV from row 4th to the data frame
df=pd.read_csv("weather.csv",delimiter="\t",header=4)
```

In [9]:

```
# Change the column name into "feature"
df.columns=["features"]
```

In [10]:

```
# Separate the contents based on commas and set new colums to them
df[['Station Name','Latitude','Longitude','Elevation','Year','Month','Day','AvgTemp','AvgTempFlag']] = df.features.str.split(',', expand=True)
```

In [11]:

```
# Eliminate unnecessary columns
df.drop(['features','Year','Month','Day','AvgTempFlag'], axis=1)
```

Out[11]:

	Station Name	Latitude	Longitude	Elevation	AvgTemp
0	Ada	47.3211	-96.5139	910	33.376
1	Ada	47.3211	-96.5139	910	33.611
2	Ada	47.3211	-96.5139	910	40.395
3	Ada	47.3211	-96.5139	910	50.546
4	Ada	47.3211	-96.5139	910	42.461
...	...	...	...	...	...
3925	Zeeland	46.013378	-99.687587	2070	36.447
3926	Zeeland	46.013378	-99.687587	2070	29.455
3927	Zeeland	46.013378	-99.687587	2070	24.417
3928	Zeeland	46.013378	-99.687587	2070	31.173
3929	Zeeland	46.013378	-99.687587	2070	37.556

3930 rows × 5 columns

In [12]:

```
# Change the datatype to make them calculable
df['Latitude'] = df['Latitude'].astype('float64')
df['Longitude'] = df['Longitude'].astype('float64')
df['Elevation'] = df['Elevation'].astype('float64')
df['AvgTemp'] = df['AvgTemp'].astype('float64')
```

In [13]:

```
# Calculate the average temperature based on the same station
AvgTemp=df.groupby("Station Name").mean()
```

In [14]:

```
# Save as a new CSV file for importing into ArcGIS Pro
AvgTemp.to_csv(r'D:\2021-spring\ArcGIS\Lab4\lab04\lab04.gdb\AvgTemp.csv')
```

In [15]:

```
# Set up the Arcpy environment
arcpy.env.workspace = r"D:\2021-spring\ArcGIS\Lab4\lab04\lab04.gdb"
```

In [16]:

```
aprx = arcpy.mp.ArcGISProject(r'D:\2021-spring\ArcGIS\Lab4\lab04\lab4.aprx')
```

In [17]:

```
# Import the CSV data into ArcGIS Pro based on the coordinate
arcpy.management.XYTableToPoint(r"D:\2021-spring\ArcGIS\Lab4\lab04\AvgTemp.csv", "AvgTe
mp", "Longitude", "Latitude")
```

Out[17]:

Output

D:\2021-spring\ArcGIS\Lab4\lab04\lab04.gdb\AvgTemp

Messages

Start Time: 2021年4月16日 下午 05:34:17
Succeeded at 2021年4月16日 下午 05:34:18 (Elapsed Time: 0.90 seconds)

In [18]:

```
AvgTemp = aprx.listLayouts("AvgTemp")[0]
```

In [19]:

```
# Save the average temperature map
AvgTemp.exportToPDF(r'D:\2021-spring\ArcGIS\Lab4\AvgTemp')
```

Out[19]:

```
'D:\\2021-spring\\ArcGIS\\Lab4\\AvgTemp.pdf'
```

In [20]:

```
# Execute IDW interpolation
arcpy.ga.IDW(r"D:\2021-spring\ArcGIS\Lab4\lab04\lab04.gdb\AvgTemp", "AvgTemp", "IDW")
```

Out[20]:

Output

id	value
0	a Layer object
1	

Messages

Start Time: 2021年4月16日 下午 05:34:49
Succeeded at 2021年4月16日 下午 05:34:49 (Elapsed Time: 0.15 seconds)

In [21]:

```
# Execute EBK interpolation
arcpy.ga.EmpiricalBayesianKriging(r"D:\2021-spring\ArcGIS\Lab4\lab04\lab04.gdb\AvgTemp"
, "AvgTemp", "EBK")
```

Out[21]:

Output

id	value
0	a Layer object
1	

Messages

Start Time: 2021年4月16日 下午 05:34:51

Warning(s) for dataset: Length of the radius of the search circle = 3.2203e+05 meters.
Succeeded at 2021年4月16日 下午 05:34:51 (Elapsed Time: 0.16 seconds)

In [22]:

```
# Execute RBF interpolation
arcpy.ga.RadialBasisFunctions(r"D:\2021-spring\ArcGIS\Lab4\lab04\lab04.gdb\AvgTemp", "A
vgTemp", "RBF")
```

Out[22]:

Output

id	value
0	a Layer object
1	

Messages

Start Time: 2021年4月16日 下午 05:34:52

Succeeded at 2021年4月16日 下午 05:34:52 (Elapsed Time: 0.16 seconds)

In [23]:

```
IDW = aprx.listLayouts("IDW")[0]
```

In [24]:

```
# Save the average temperature map with IDW interpolation
IDW.exportToPDF(r'D:\2021-spring\ArcGIS\Lab4\IDW')
```

Out[24]:

```
'D:\\2021-spring\\ArcGIS\\Lab4\\IDW.pdf'
```

In [25]:

```
EBK = aprx.listLayouts("EBK")[0]
```

In [26]:

```
# Save the average temperature map with EBK interpolation
EBK.exportToPDF(r'D:\2021-spring\ArcGIS\Lab4\EBK')
```

Out[26]:

```
'D:\\2021-spring\\ArcGIS\\Lab4\\EBK.pdf'
```

In [27]:

```
RBF = aprx.listLayouts("RBF")[0]
```

In [28]:

```
# Save the average temperature map with RBF interpolation
RBF.exportToPDF(r'D:\2021-spring\ArcGIS\Lab4\RBF')
```

Out[28]:

```
'D:\\2021-spring\\ArcGIS\\Lab4\\RBF.pdf'
```