Epigraph

"When our maps do not fit the territory, when we act as if our inferences are factual knowledge, we prepare ourselves for a world that isn't there. If this happens often enough, the inevitable result is frustration and an ever-increasing tendency to warp the territory to fit our maps. We see what we want to see, and the more we see it, the more likely we are to reinforce this distorted perception, in the familiar circular and spiral feedback pattern."

Professor Harry L. Weinberg

Preface: In this tutorial, you will learn about basic concept of Exploratory Spatial Data Analysis, the main object of this handbook is to guide you to dive in spatial data visualization and analysis. Whether you're getting started with your first spatial data analysis or are just looking to learn more about how they work, you'll find useful information for your needs in our written lessons. When you're done, you'll have a fundamental understanding of how to use a spatial data, visualize them or even making decision. You'll also be ready to learn even more about Spatial science with some of our other libraries that will be introduced in here. In this journey you will be introduced to some amazing tools and analysis which will help you to make expert-like decisions. ArcGIS is one of most astonishing software in spatial and geographical field, this software has many tools and feature that one can utilize. Alongside with this software, Python programming language will be exploited too. Since Python is open source, numerous libraries for data related application has been developed and ready to use.

Contents

First chapter

Second chapter

I

Geovisualization

1.1. Concepts of geovisualization

Geovisualization or geovisualisation (short for geographic visualization), also known as cartographic visualization, refers to a set of tools and techniques supporting the analysis of geospatial data through the use of interactive visualization. Like the related fields of scientific visualization and information visualization, geovisualization emphasizes knowledge construction over knowledge storage or information transmission. To do this, geovisualization communicates geospatial information in ways that, when combined with human understanding, allow for data exploration and decision-making processes.

Traditional, static maps have a limited exploratory capability; the graphical representations are inextricably linked to the geographical information beneath. GIS and geovisualization allow for more interactive maps; including the ability to explore different layers of the map, to zoom in or out, and to change the visual appearance of the map, usually on a computer display. Geovisualization represents a set of cartographic technologies and practices that take advantage of the ability of modern microprocessors to render changes to a map in real time, allowing users to adjust the mapped data on the fly.

In other words, "Geovisualization" is the term that is used inconsistently, referring to a map, a display type, a process, a technique, a way of using maps, and an academic discipline. Despite this inconsistency, the context in which the term geovisualization appears almost always has a relationship to interactive digital cartography. One can conclude that "Geovisualization" is an elusive word to define. As computers dominated almost all domains of scholarly work and human life, a need to distinguish "computer" cartography from the thousands of years old art and science of "traditional" cartography has emerged. This need was fuelled by the fact that the digital/dynamic displays offer remarkably more flexibility and new opportunities for the design and use of maps in comparison to static media, and that the questions we can ask and answer with maps and map-like displays have changed with computers. These developments caused a paradigm shift from communication in cartography with

a focus on explanatory approaches, to exploration and knowledge construction in geovisualization.

The fact that today a user can make on-demand changes to the display and access a variety of linked visualizations in real time (and thus explore the data from different perspectives) situates geovisualization at the core of visual information processing to facilitate thinking in complex decision-making tasks and in scientific investigations. A geovisualization environment still enables visual communication, but importantly, one can visualize the data also at early- and mid-stages of the knowledge construction process in spatial analysis, and generate hypotheses based on the insights prompted by the visual stimuli. This line of thinking was affected by developments in statistics, specifically, moving from explanatory analyses to the exploratory data analysis (EDA). According to Google's online ngrams tool the term geovisualization starts frequently appearing in textbooks around 1990s, and slowly replaces digital cartography and computer cartography.

1.2. Geovisualization application

Geovisualization has made inroads in a diverse set of real-world situations calling for the decision-making and knowledge creation processes it can provide. The following list provides a summary of some of these applications.

Wildland fire fighting

Firefighters have been using sandbox environments to rapidly and physically model topography and fire for wildfire incident command strategic planning. The SimTable is a 3D interactive fire simulator, bringing sandtable exercises to life. The SimTable uses advanced computer simulations to model fires in any area, including local neighborhoods, utilizing actual slope, terrain, wind speed/direction, vegetation, and other factors.

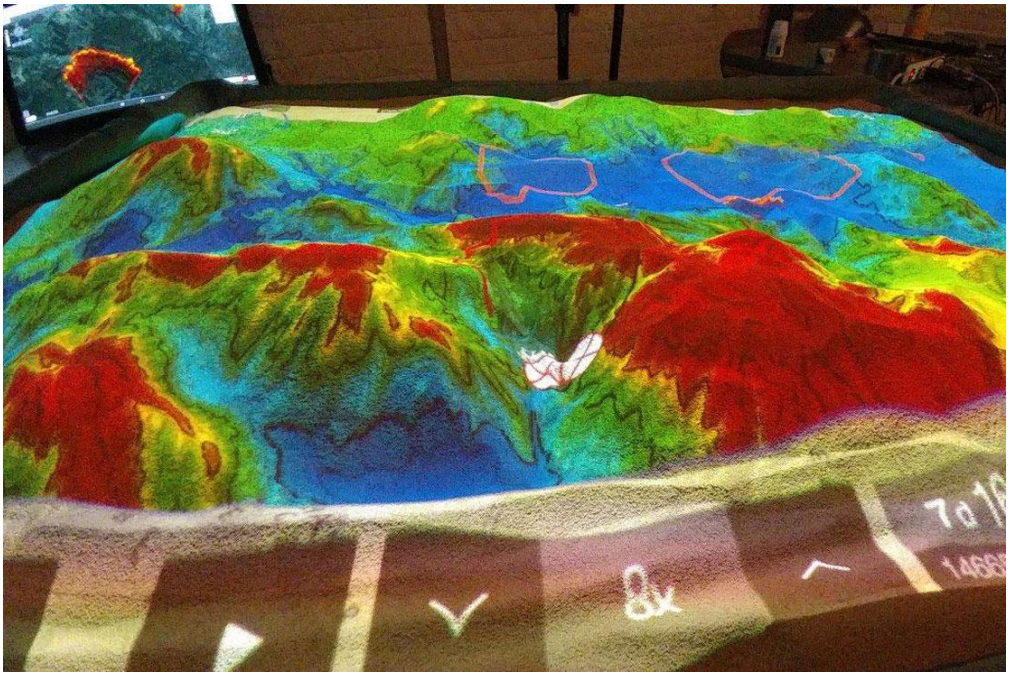


Figure 1- SimTable 3D fire simulator

Forestry

Geovisualizers, working with European foresters, used CommonGIS and Visualization Toolkit (VTK) to visualize a large set of spatio-temporal data related to European forests, allowing the data to be explored by non-experts over the Internet. The research team cited the two major problems as the inability of the geovisualizers to convince the foresters of the efficacy of geovisualization in their work and the foresters' misgivings over the dataset's accessibility to non-experts engaging in "uncontrolled exploration". While the geovisualizers focused on the ability of geovisualization to aid in knowledge construction, the foresters preferred the information-communication role of more traditional forms of cartographic representation.

Archaeology

Geovisualization provides archaeologists with a potential technique for mapping unearthed archaeological environments as well as for accessing and exploring archaeological data in three dimensions. The implications of geovisualization for archaeology are not limited to advances in archaeological

theory and exploration but also include the development of new, collaborative relationships between archaeologists and computer scientists.

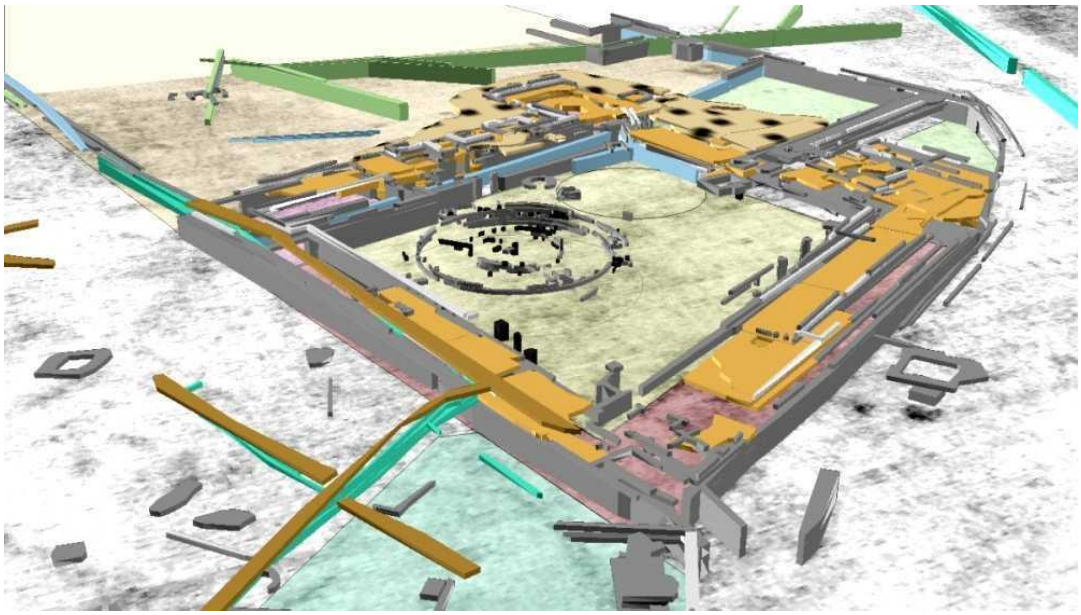


Figure 2- Geovisualization in archaeology

Environmental studies

Geovisualization tools provide multiple stakeholders with the ability to make balanced environmental decisions by taking into account "the complex interacting factors that should be taken into account when studying environmental changes". Geovisualization users can use a georeferenced model to explore a complex set of environmental data, interrogating a number of scenarios or policy options to determine a best fit.

Urban planning

Both planners and the general public can use geovisualization to explore real-world environments and model “what if” scenarios based on spatio-temporal data. While geovisualization in the preceding fields may be divided into two separate domains—the private domain, in which professionals use geovisualization to explore data and generate hypotheses, and the public domain, in which these professionals present their "visual thinking" to the

general public, planning relies more heavily than many other fields on collaboration between the general public and professionals.

Planners use geovisualization as a tool for modeling the environmental interests and policy concerns of the general public. Jiang et al. mention two examples, in which "3D photorealistic representations are used to show urban redevelopment, dynamic computer simulations are used to show possible pollution diffusion over the next few years." The widespread use of the Internet by the general public has implications for these collaborative planning efforts, leading to increased participation by the public while decreasing the amount of time it takes to debate more controversial planning decisions.

1.3. MacEachren's idea of geovisualization

MacEachren coined the word geovisualization by contracting the expression "geographic visualization"; its essence is a new approach to the use of maps. One of its features is that a map is not created for the public but for individual use and its primary purpose is to provide new insights from the data. It supposes an intensive interaction between people and maps in the sense that we can directly manipulate the spatial data to be mapped. If we are talking about visualization, we do not use maps alone, but in combination with other visual aids (charts, tables, photographs, 3D models, etc.). In this sense visualization and communication are complementary events during map use

Along with a plethora of technology-driven developments, important conceptual frameworks also were proposed around 1990s. A defining theoretical framework on geovisualization is MacEachren's Cartography3 (Figure-3 and Figure-4). MacEachren's framework extends the earlier Swoopy framework proposed by DiBiase. DiBiase's Swoopy framework offers a continuum in which we see both visual thinking and visual communication, and as such, it provides foundations as to how we think about geovisualization today (Figure-3).

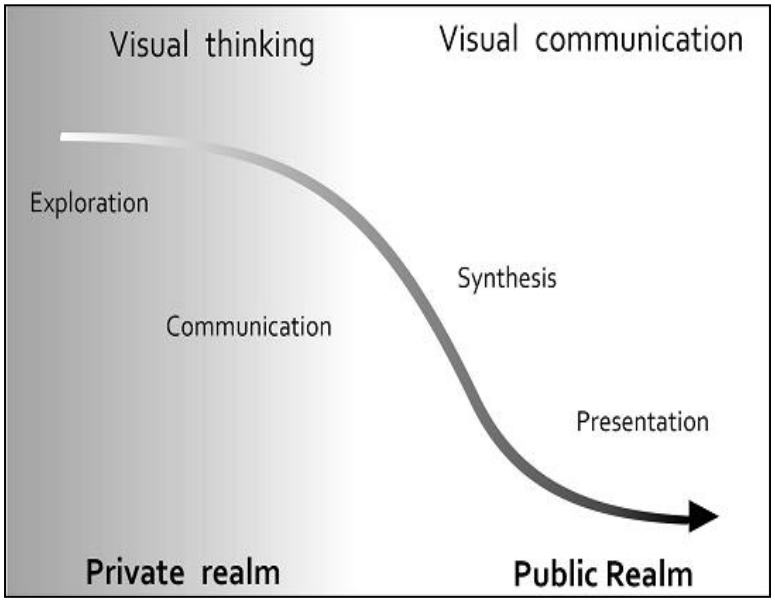


Figure 3- Swoopy framework of visualization

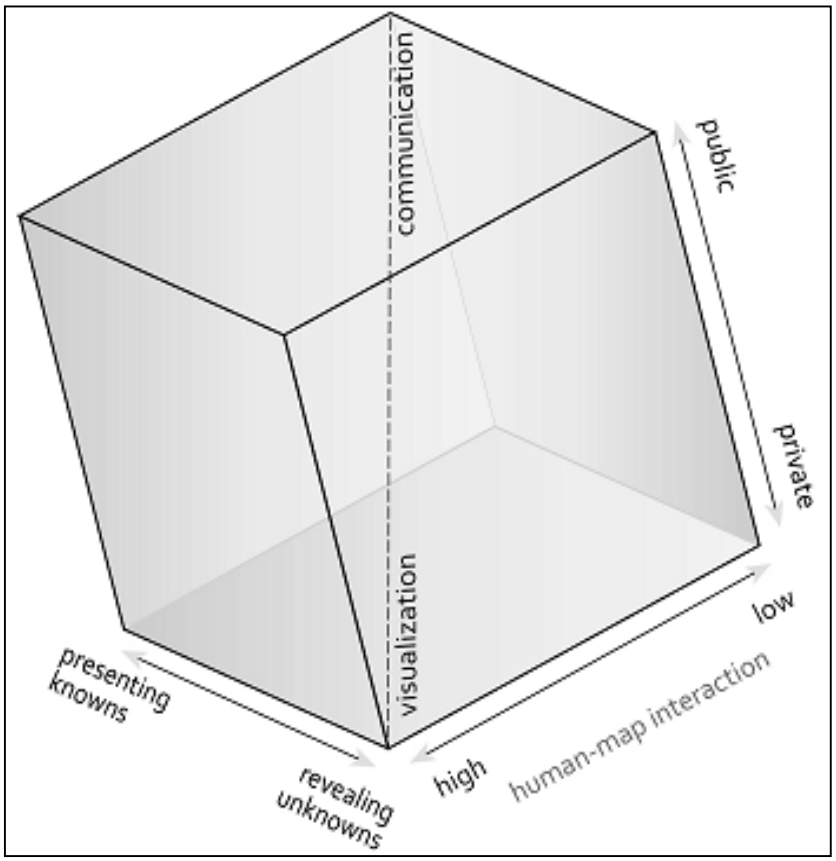


Figure 4- Allen MacEachren's framework of visualization

The 3D Cartography framework extends the Swoopy framework, essentially adding the **interaction** (low vs. high) as a dimension and mapping its relationship to **users** (public vs. expert), and **tasks** (communication vs.

exploration), in a continuum (Figure-4). MacEachren further developed the 3D Cartography model later (Figure-5), slightly adjusting Swoopy's "idealized" research steps exploration, confirmation, synthesis, and presentation (Figure-3) by replacing confirmation step with analysis.

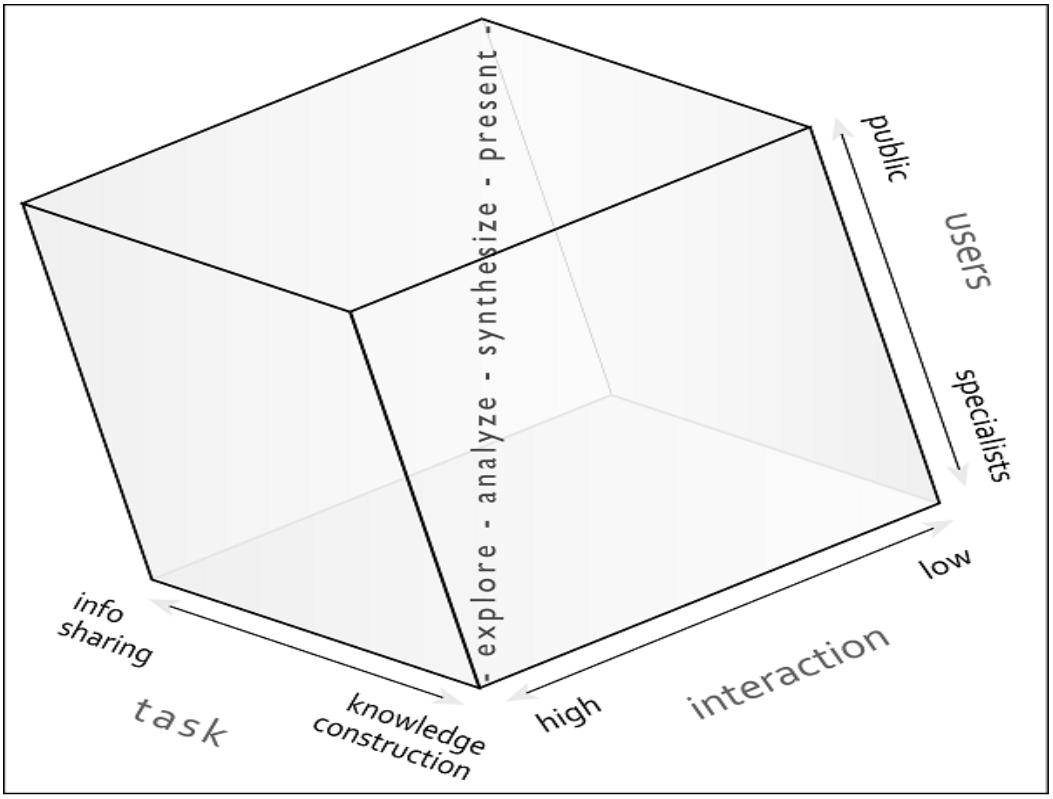


Figure 5- MacEachren's 3D cartographic-visualization conceptual model

The updated framework (Figure-5) sums up the core functions of geovisualization: With the support from geovisualization software environments, the public (e.g., non-expert users) or specialists (e.g., researchers, decision makers) can discover patterns and form informed questions (exploration), conduct analyses to confirm or reject individual hypotheses (analysis/confirmation), generalize the findings (synthesis), and present/communicate these findings. The framework suggests that the specialists use (more) interactive geovisualization environments in exploratory processes for knowledge construction, whereas interaction requirements are lower as we move towards the goal to communicate (info sharing/presentation) with/by public. The concepts covered by Cartography3 —and its 2004 update—

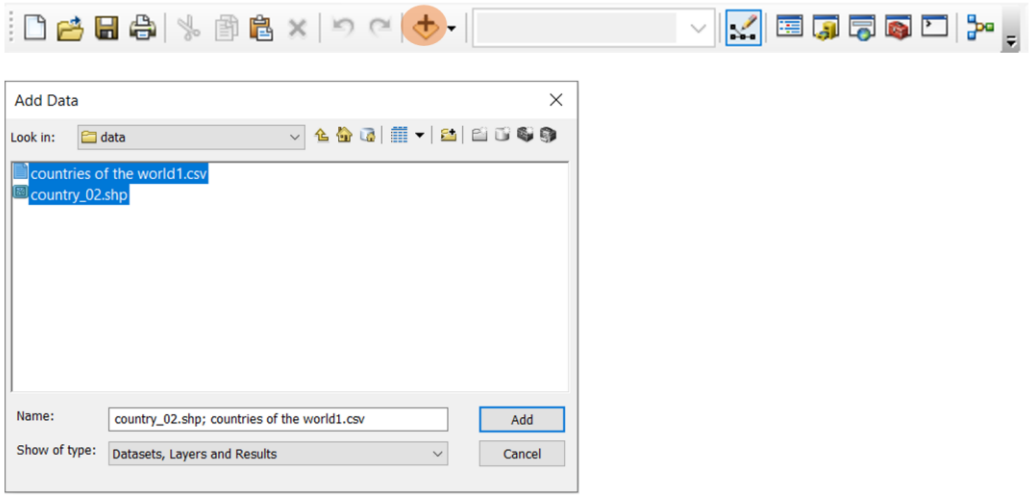
remain relevant today and are core to our understanding of how geovisualization was born as a perspective on cartography.

1.4. Geovisualization examples in ArcGIS

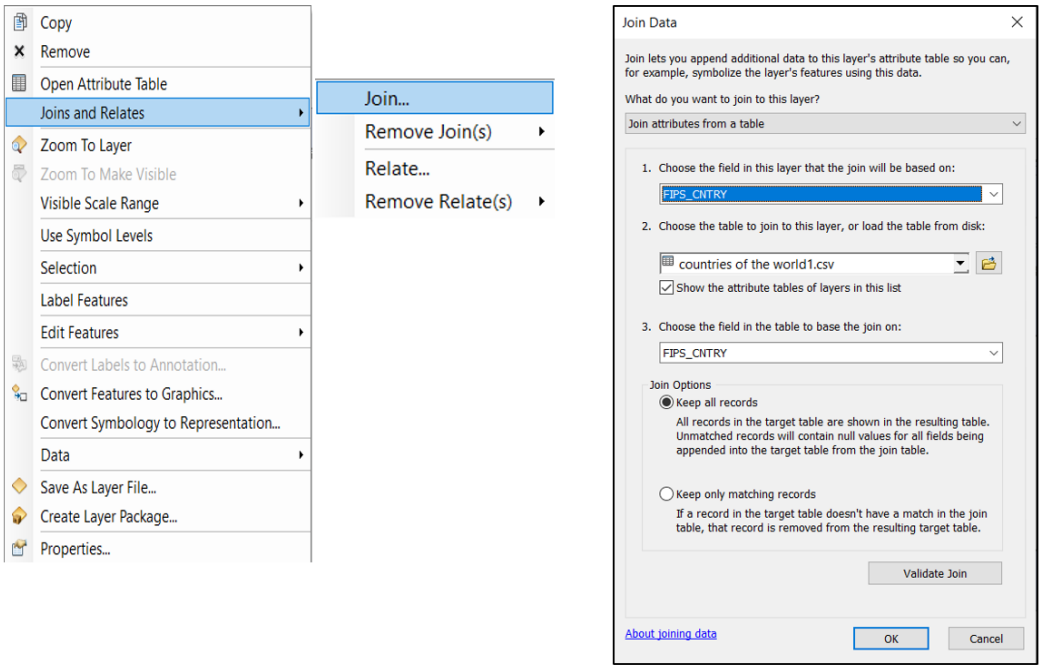
Example 1

Create a map that visualizes infant mortality in Latin America. You can download data for this example from here.

Step 1. adding data: Click on “Add data”, then browse CSV file and Shapefile, and press Add button

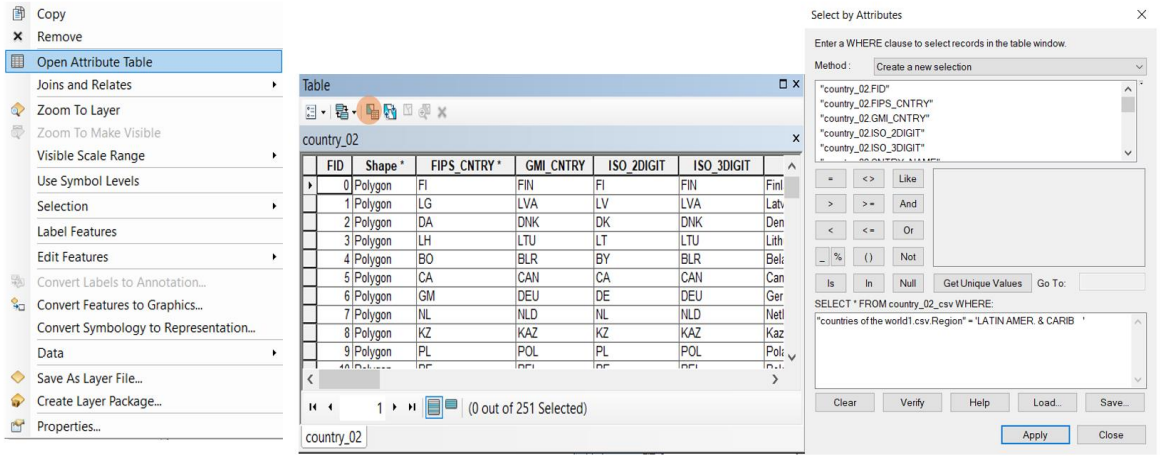


Step 2: Join data’s



We have two different data (world shape file, excel data). These data's must be joint together. At first, Right click on "country_02.shp" and seek "Joins and Relates" and click "join". Then, the right-side menu will be generated. Choose the key fields in files (FIPS_CNTRY) and press OK button.

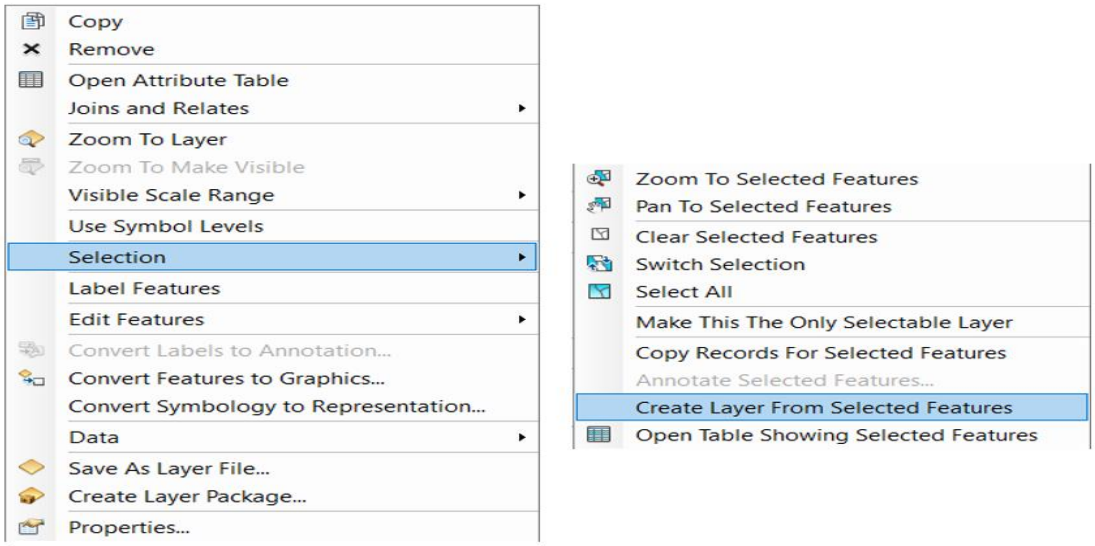
Step 3: select Latin America countries.



At first, Right click on "country_02.shp" and click on "Open Attribute Table". Then, click on mentioned icon called "Select by Attribute". After that, a query for choosing Latin American countries is needed. The query is:

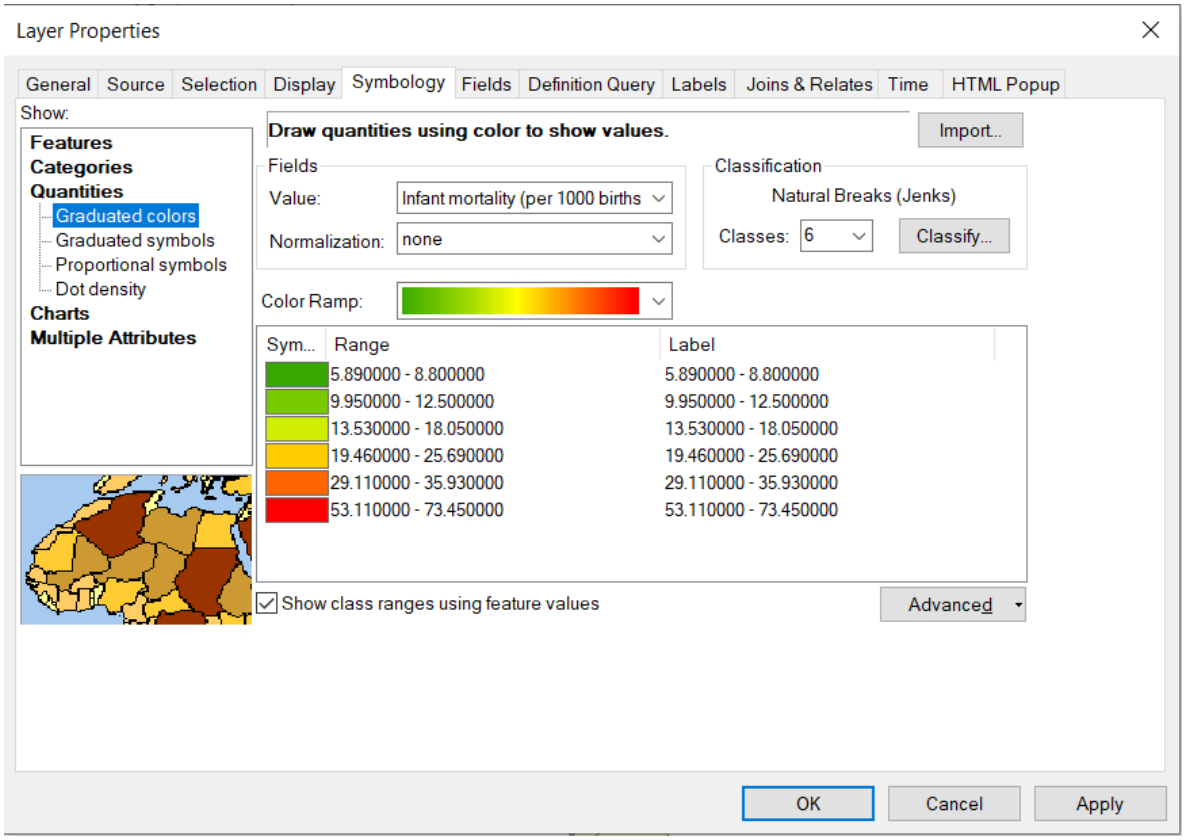
< SELECT * FROM country_02_csv Where: "countries of the world1.csv.Region" = 'LATIN AMER. & CARIB' >

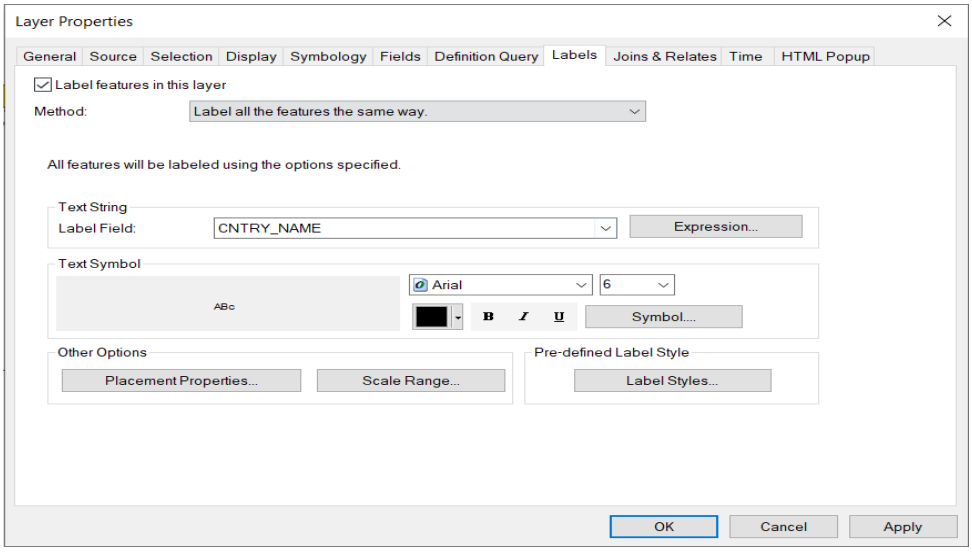
Step 4: Extract selected region



Right click on “country_02.shp” and seek “Selection” then click on “Create Layer From Selected Features”.

Step 5: Create Graduated colors map.



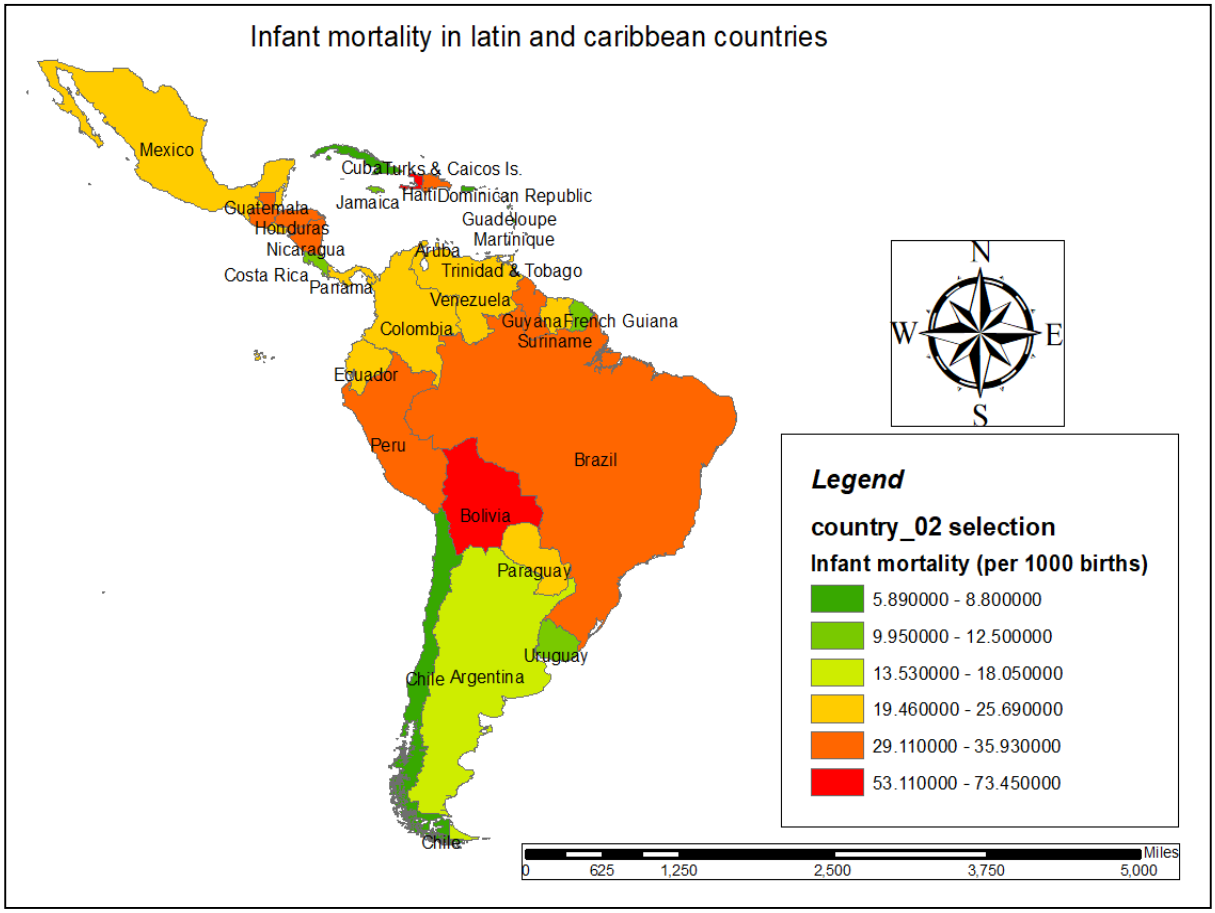


Right click on “country_02_selection.shp”, click on “Properties” and Seek “Symbology” tab. This sub menu is one of the most important menu for creating

map in ArcMap environment. On the left part (show:) map type must be chosen. As our field (infant mortality) is a quantitative value, we seek Quantities and select “Graduated colours”. Then, on the middle part the field must be chosen (“infant mortality”). On the right-side number of classes is needed .If the number of defined classes is large, the map would be meaningless and if the number of defined classes is low, the accuracy would be reduced. Be sides, the colour ramp should be chosen carefully, since the colour influence map viewers felling.

“Labels” sub-menu is used for labeling the map. By clicking check-box button the labels will be shown on the map. Other settings are useful for better interface.

Final Result:

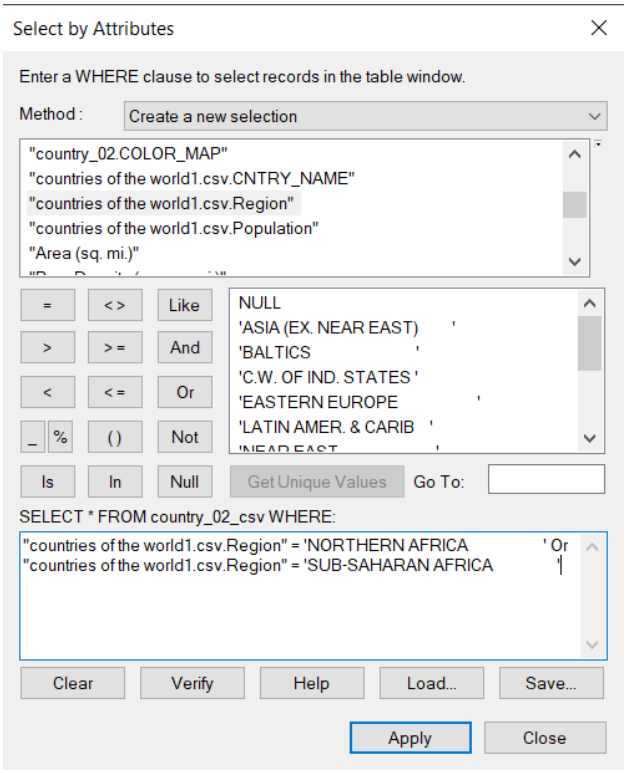


Example 2

Create a map that visualizes GDP in whole African regions.

Step 1 and step 2 are the same as previous example.

Step 3 : Select African countries.



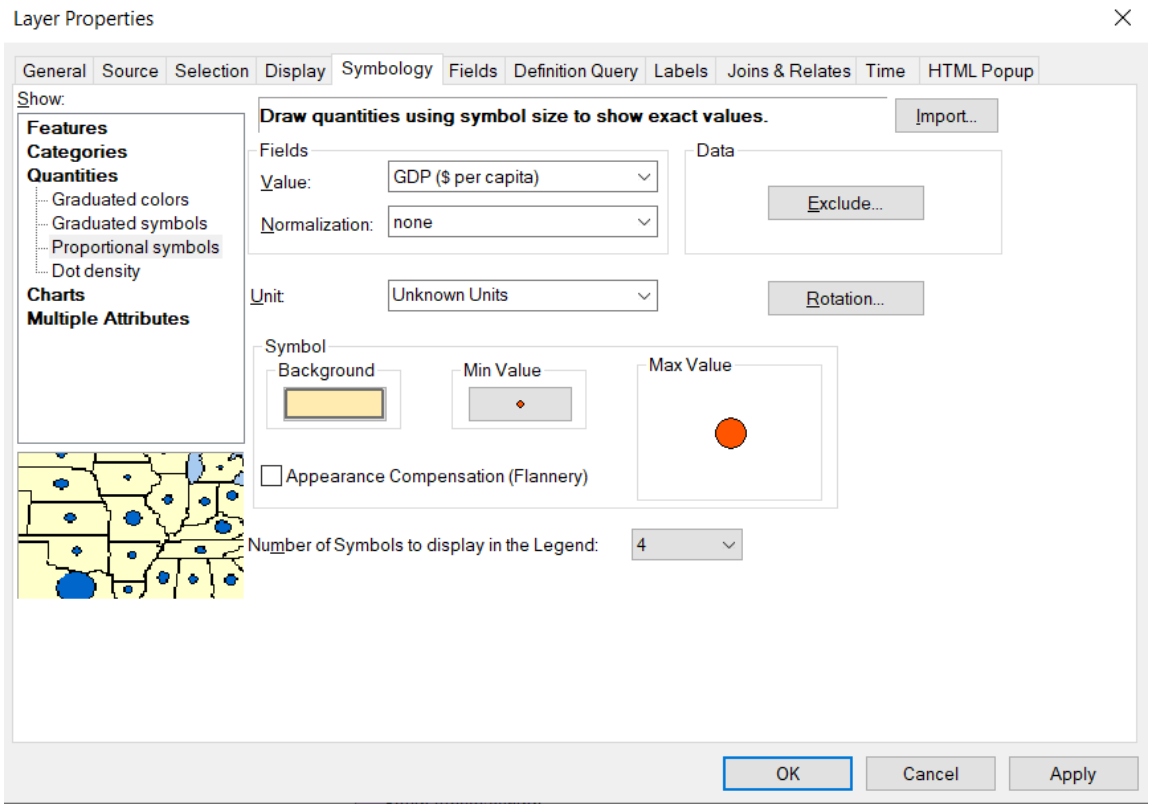
Same as before, Right click on “country_02.shp” and click on “Open Attribute Table”. Then, click on mentioned icon called “Select by Attribute”. After that, a query for choosing African countries is needed. The query is:

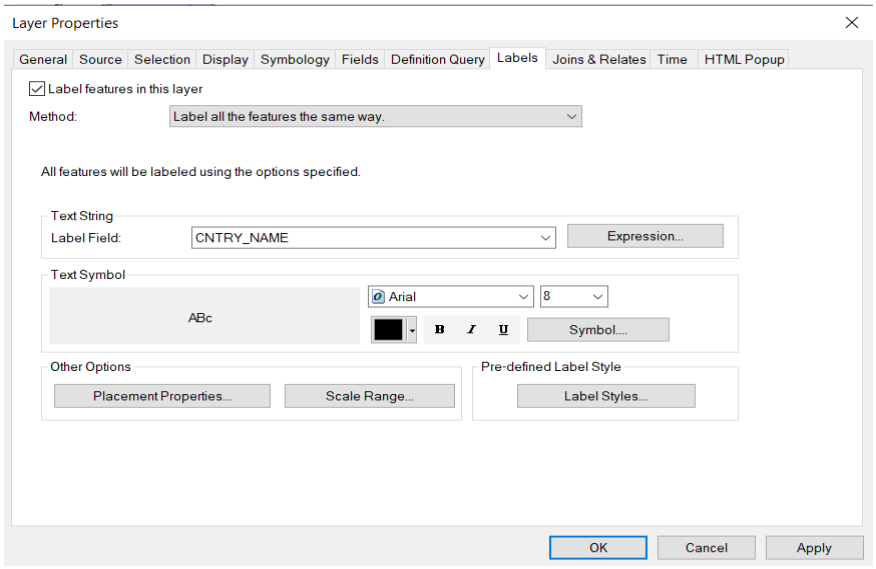
<SELECT * FROM country_02_csv Where: "countries of the world1.csv.Region" = 'NORTHERN AFRICA' OR "countries of the world1.csv.Region" = 'SUB-SAHARAN AFRICA' >

OR operator is used since African countries were divided to two parts and both parts is needed.

Step 4 is same as previous example.

Step 5: Create Proportional symbols map.

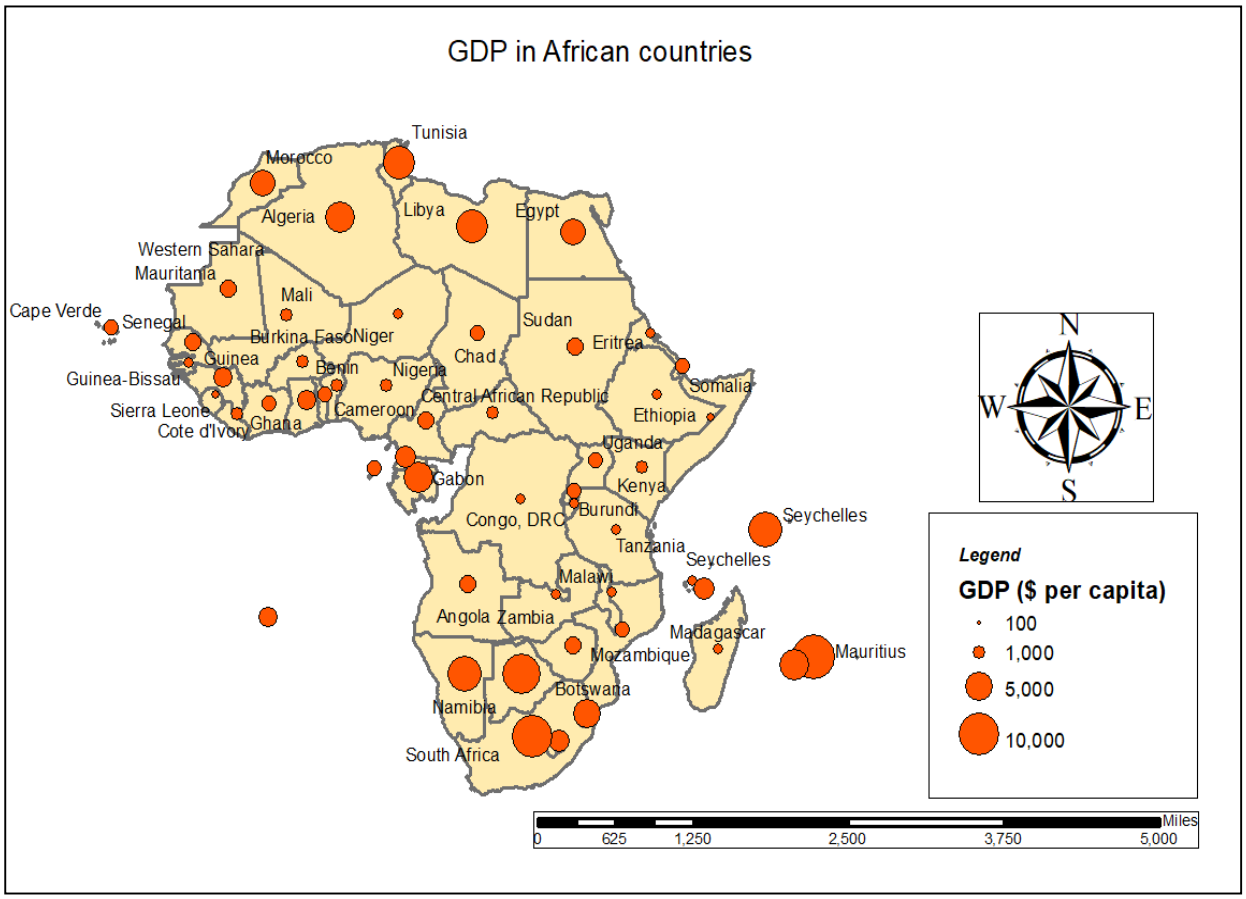




Right click on “country_02.shp”, click on “Properties” and Seek “Symbology” sub-menu. On the left part (show:) map type must be chosen. As our field (GDP) is a quantitative value, we seek Quantities and select “Proportional symbols”. Then, on the middle part the field must be chosen (“GDP”). Be sides, the background and Min value symbols must be chosen.

"Labels" sub-menu is used for labeling the map. By clicking check-box button the labels will be shown on the map. Other settings are useful for better interface.

Final Result:

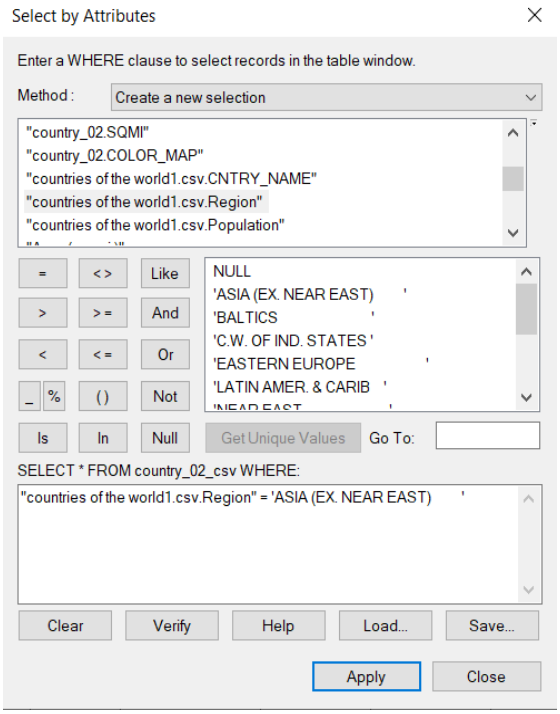


Example 3

Create a map that visualizes birth and death rate in Near East Asia.

Step 1 and step 2 are the same as previous example.

Step 3: Select Near East Asian countries.

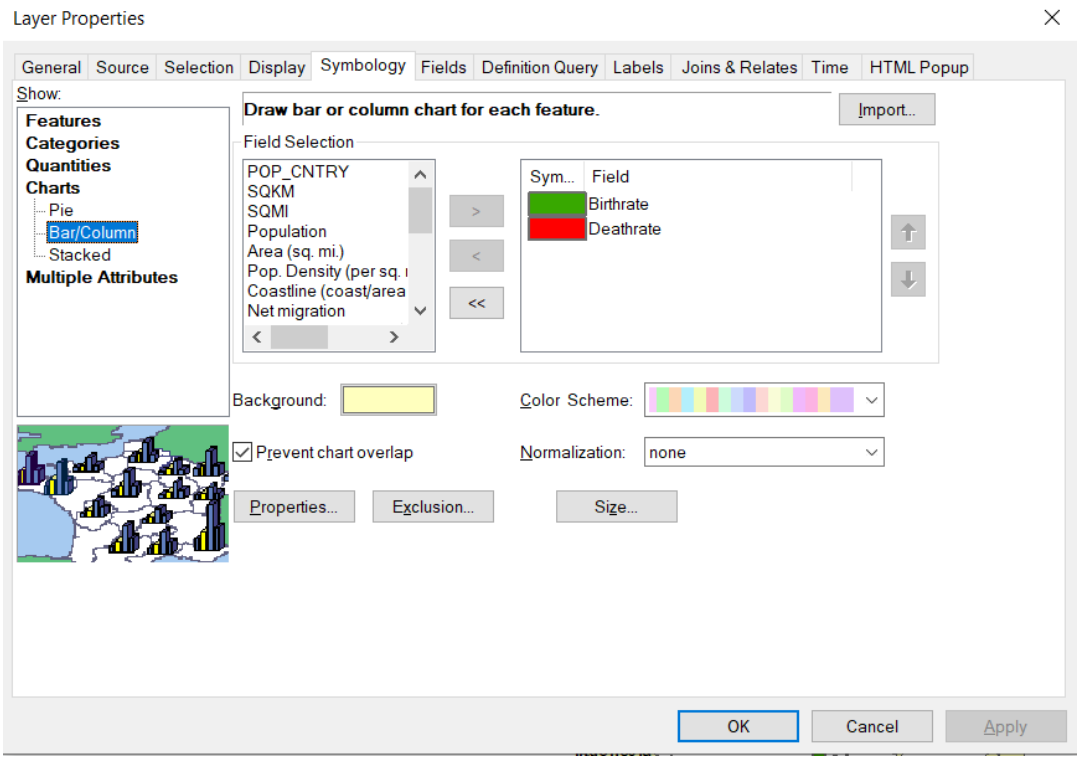


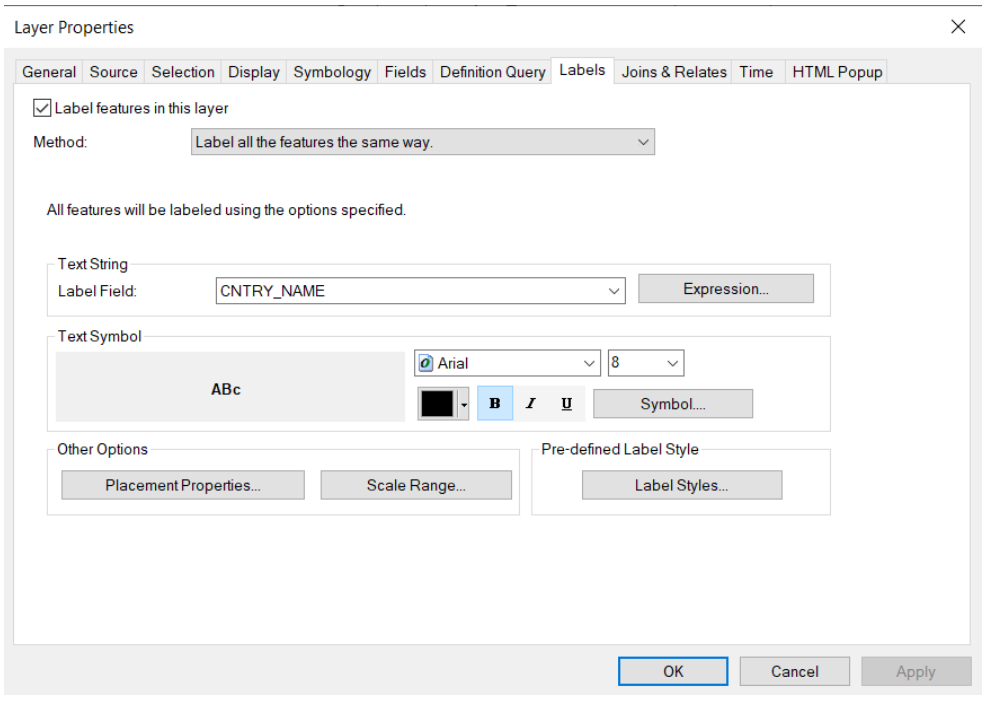
Same as before, Right click on “country_02.shp” and click on “Open Attribute Table”. Then, click on mentioned icon called “Select by Attribute”. After that, a query for choosing Near East Asian countries is needed. The query is:

<SELECT * FROM country_02_csv Where: "countries of the world1.csv.Region" = 'ASIA (EX. NEAR EAST)' >

Step 4 is same as previous example.

Step 5 : Creat Bar chart map.



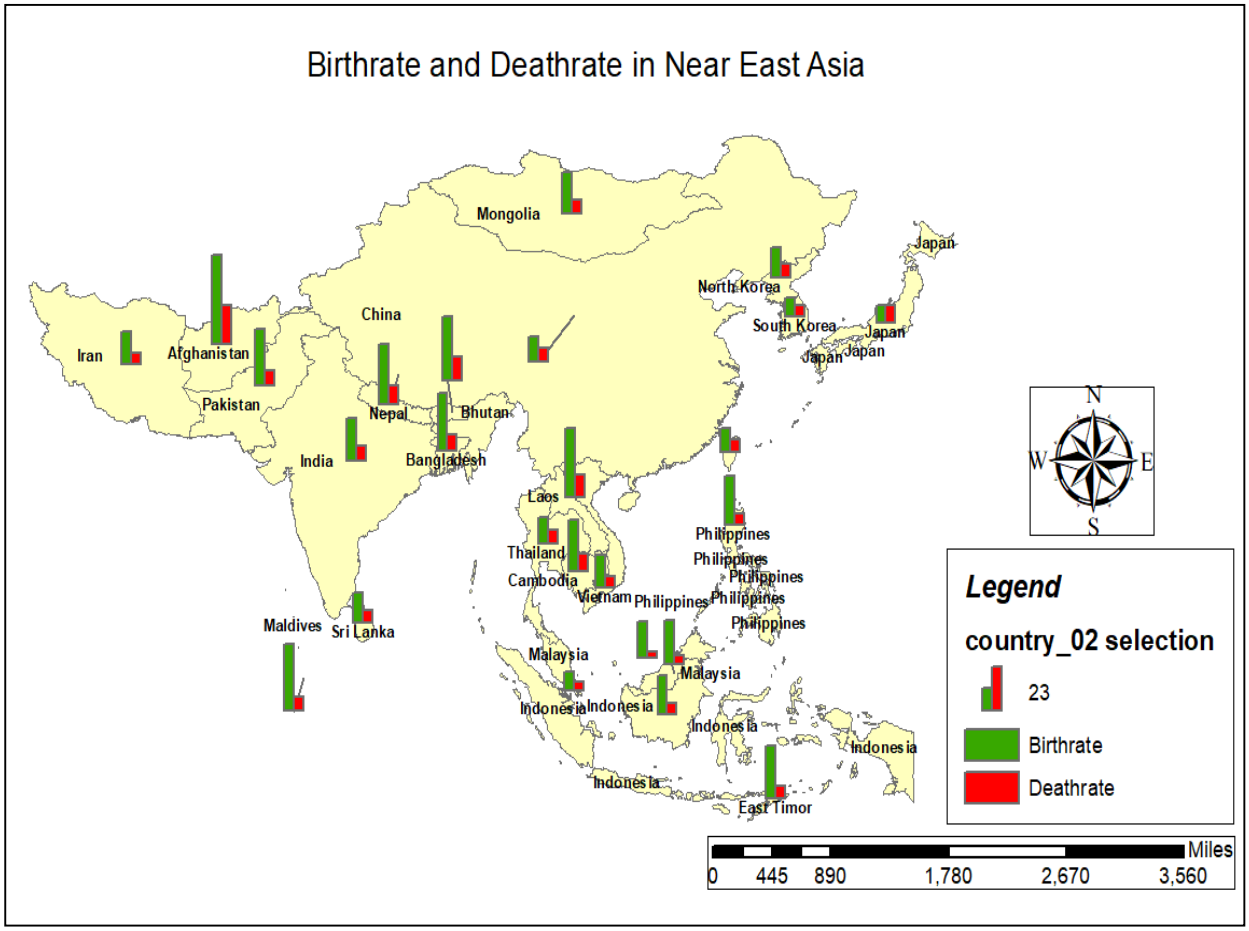


Right click on “country_02.shp”, click on “Properties” and Seek “Symbology” sub-menu. On the left part (show:) map type must be chosen. Since a comparing between birth and death rate is required, we seek Charts and select “Bar/Column”. Then, on Field Selection Birthrate and Deathrate must be

chosen. To clarify the results, prevent chart overlap check-box button is turned ON.

“Labels” sub-menu is used for labeling the map. By clicking check-box button the labels will be shown on the map. Other settings are useful for better interface.

Final Result:

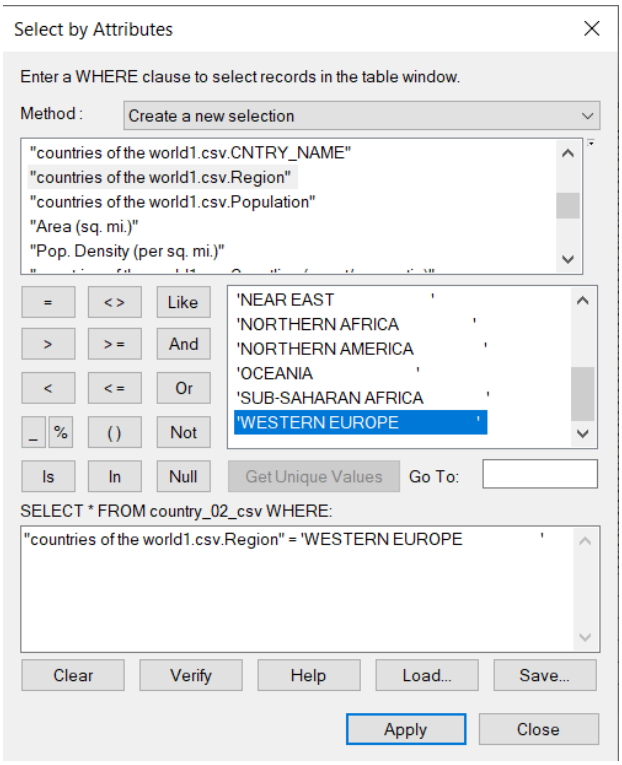


Example 4

Create a map that visualizes Agriculture, Industry and Services in Commonwealth of Independent States.

Step 1 and step 2 are the same as previous example.

Step 3: Select Commonwealth of Independent States.

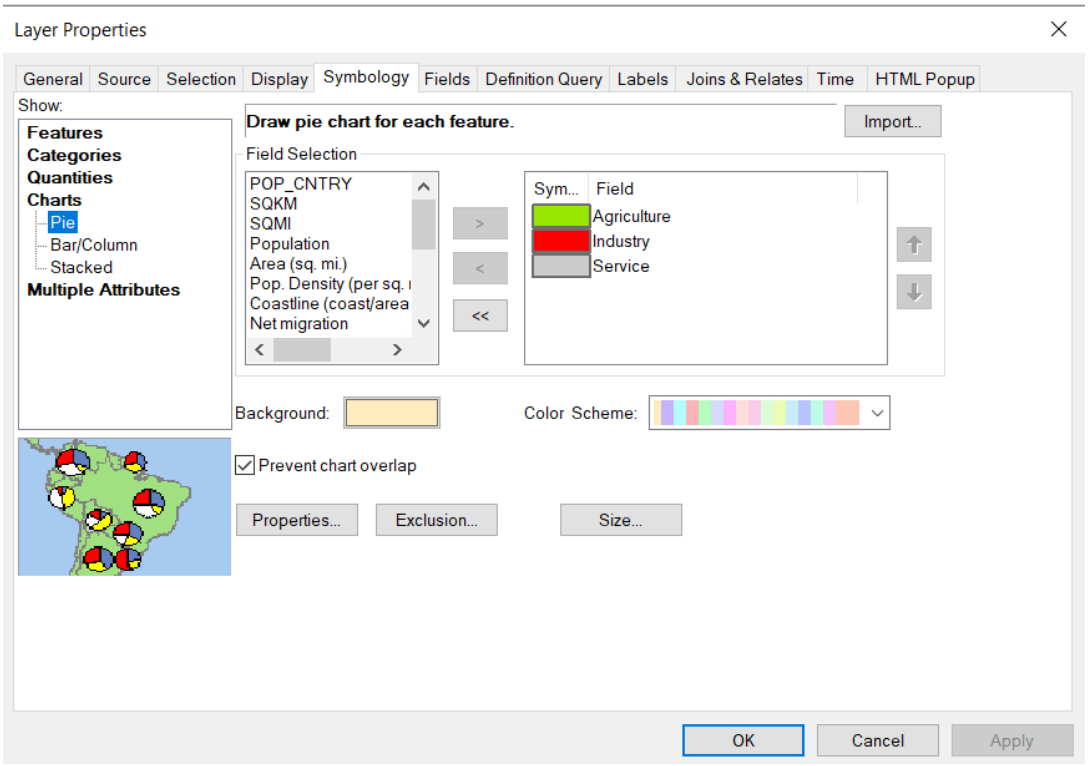


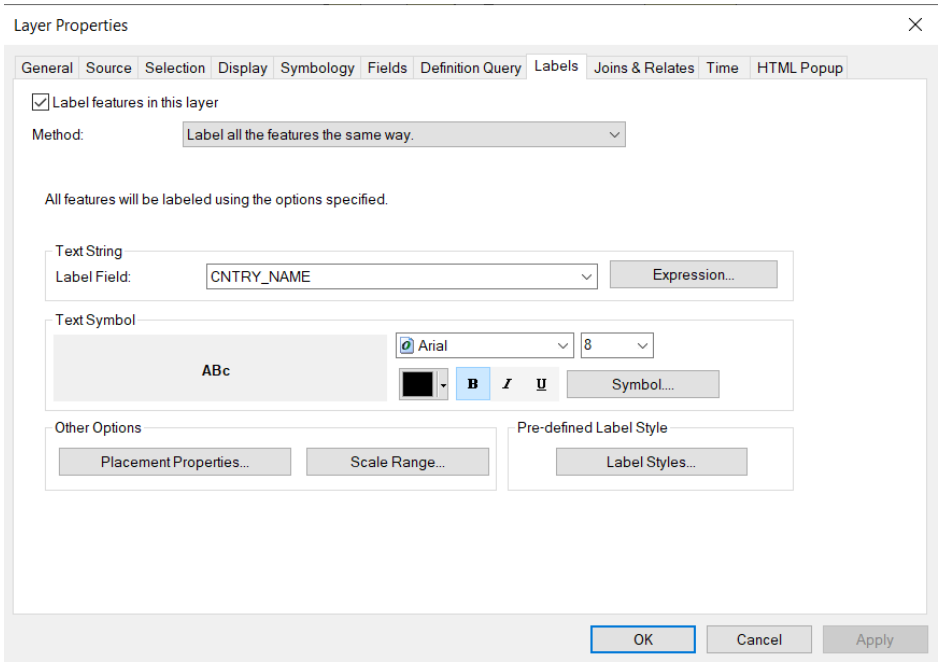
Same as before, Right click on “country_02.shp” and click on “Open Attribute Table”. Then, click on mentioned icon called “Select by Attribute”. After that, a query for choosing Commonwealth of Independent States is needed. The query is:

<SELECT * FROM country_02_csv Where: "countries of the world1.csv.Region" = 'C.W. OF IND. STATES '>

Step 4 is same as previous example.

Step 5: Create Pie chart map.



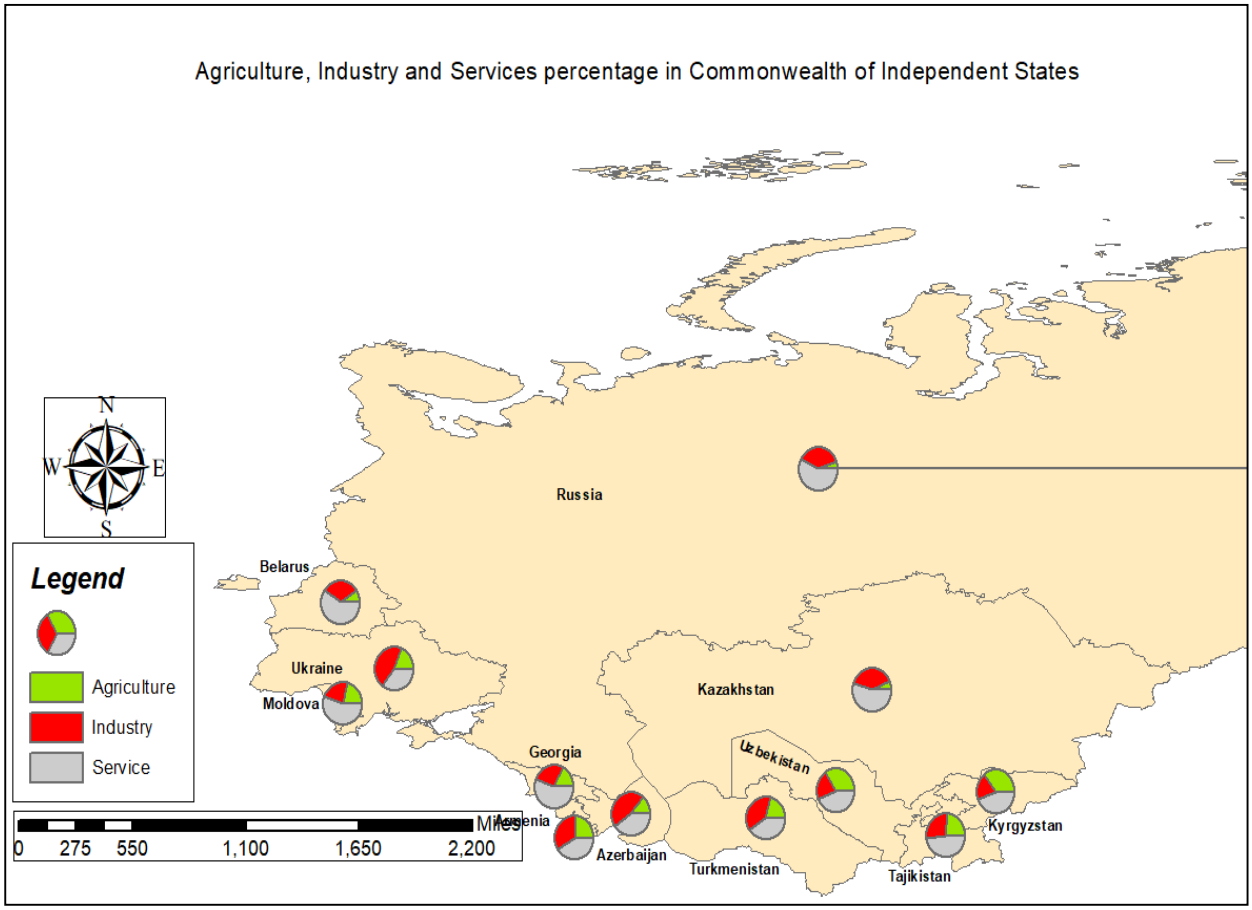


Right click on “country_02.shp”, click on “Properties” and Seek “Symbology” sub-menu. On the left part (show:) map type must be chosen. Since a comparing between Agriculture, Industry and Services is required, we seek Charts and select “Pie”. Then, on Field Selection Agriculture, Industry and Services must

be chosen. To clarify the results, prevent chart overlap check-box button is turned ON.

“Labels” sub-menu is used for labeling the map. By clicking check-box button the labels will be shown on the map. Other settings are useful for better interface.

Final Result:



II

Data Visualization

2.1. Histogram

One of the simplest means of generating a graphical summary is by plotting the frequency distribution of a single variable (univariate) that is measured on an interval scale. It reveals the centre of the data (mean, median, and mode), the spread of the data (dispersion or unevenness), the shape and distribution (skewness) of the data, and evidence of potential outliers in the data.

A frequency distribution shows how often each different value in a set of data occurs. A histogram is the most commonly used graph to show frequency distributions. It looks very much like a bar chart, but there are important differences between them. This helpful data collection and analysis tool is considered one of the seven basic quality tools.

Use a histogram when, the data are numerical or want to see the shape of the data's distribution, especially when determining whether the output of a process is distributed approximately normally. Histogram also useful for analysing whether a process can meet the requirements or analysing what the output from a process looks like, seeing whether a process change has occurred from one-time period to another and determining whether the outputs of two or more processes are different.

2.1.1. Histogram in Python

Python programming language will be used to generate Histogram of infant mortality rate around the world. For this purpose, data that used in chapter one will be used in here to, in case you skipped first chapter examples, data accessible in here.

Python is rich programming language, there are many free libraries for almost any use. a package which is useful and most used for dealing with data is "Pandas". To installing libraries on your python environment, use !pip install <package>. Next step is to use import command to call needed libraries.

```
import pandas as pd
```

Next step is to read "countries of the world1.csv" from download folder, this file contains some information such as infant mortality about countries in world. It is good to mention that, in absence of CSV the "Shape file" data also could be read. To read shape file data as Data frame like pandas reads excel files, one can use "GeoPandas" library. After reading file rest of the procedure is same.

```
df = pd.read_csv("countries of the world1.csv")
```

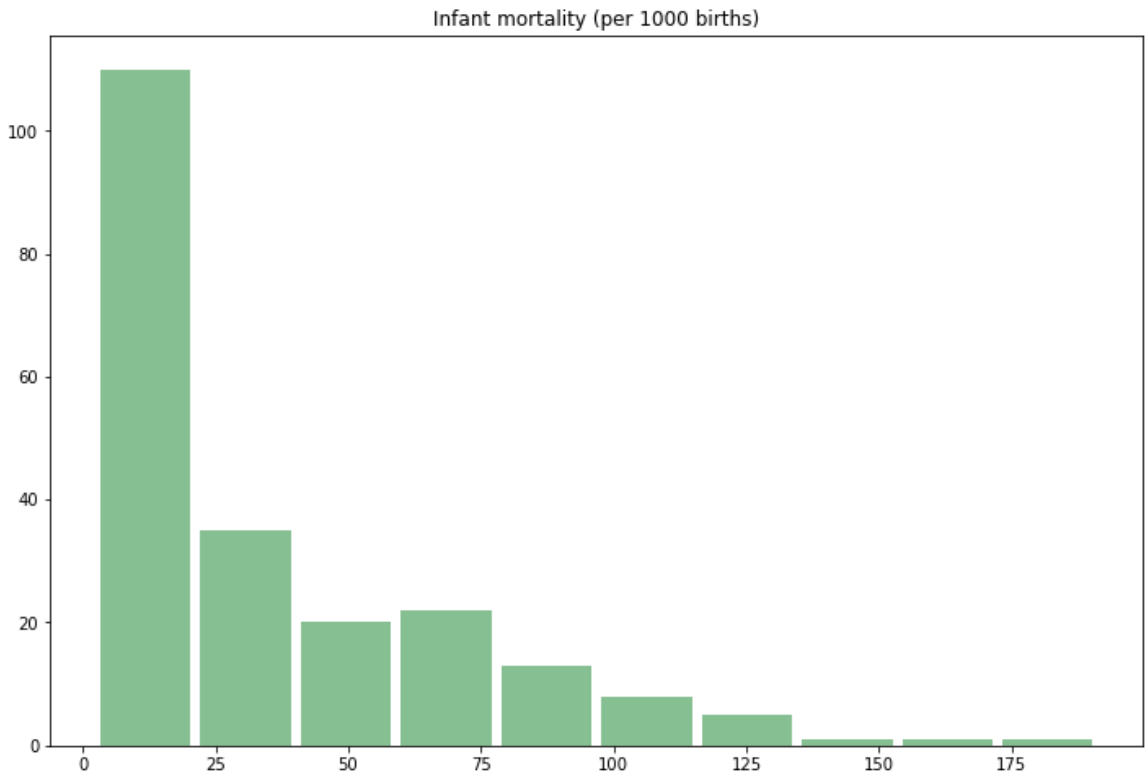
To visualize any data, its essential to clean data from null or irrelevant values, also sometimes it's necessary to change data formats so they can be suitable for visualizing.

```
df["Infant mortality (per 1000 births)"] = df["Infant mortality (per 1000 births)"].str.replace(",",".").astype(float)
```

As mentioned above, to clean data, comma will be replaced by dot, datatype from string will be changed to float numbers and all will be replaced in the infant mortality column. The next step is to draw histogram plot for this data, the "Pandas" library has this options too, also the other popular libraries like "matplotlib", "seaborn" or "Geoplotlib" also could be used.

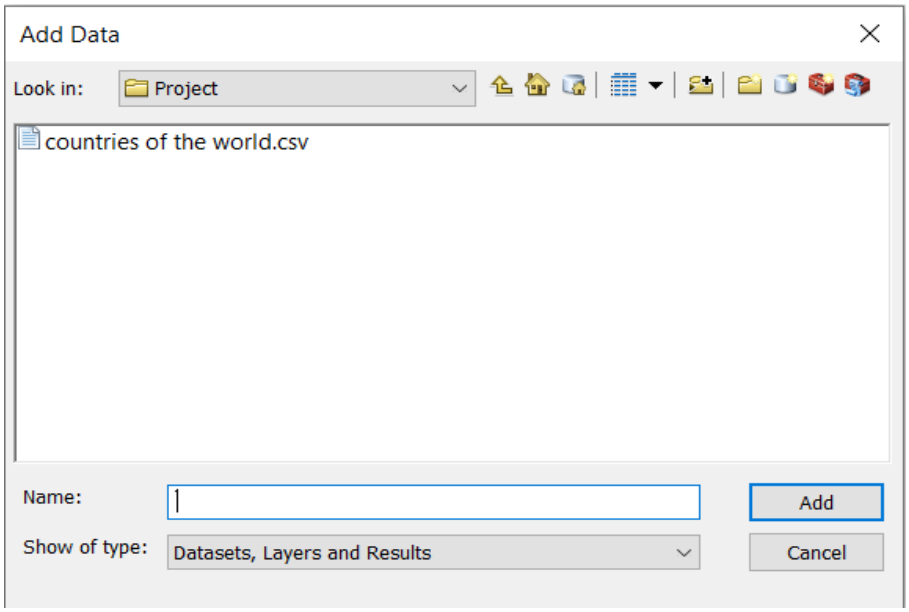
```
df.hist(column='Infant mortality (per 1000 births)', grid=False, figsize=(12,8), color='#86bf91', zorder=2, rwidth=0.9)
```

The column in data frame needed to be introduced to "hist" module of pandas to draw histogram, other arguments like figure size, color of bars or space between bars are optional. This example final result can be seen in below figure.



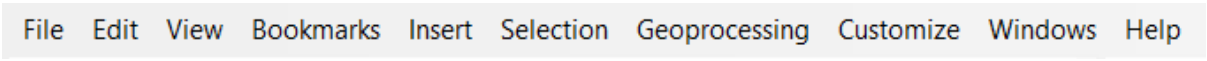
2.1.2. Histogram in ArcGIS

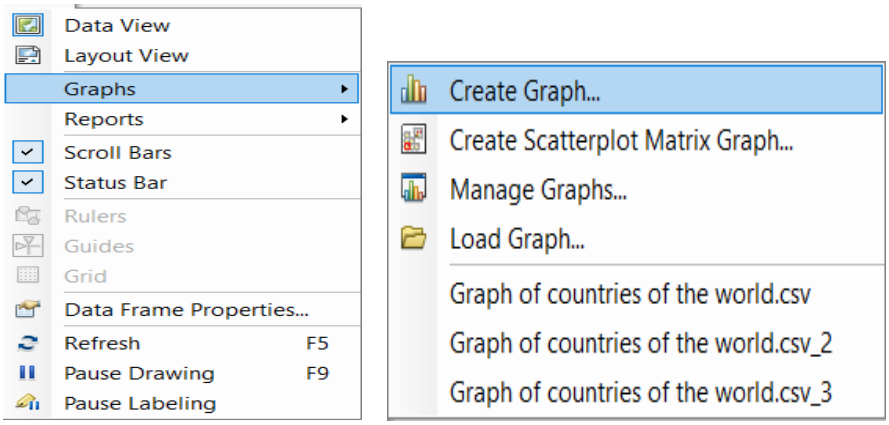
It's also possible to generate histogram plot using desktop applications like ArcGIS. World countries used in example to create a histogram of countries literacy. First step is: *adding data*



Click on "Add data", then browse excel file, and press Add button.

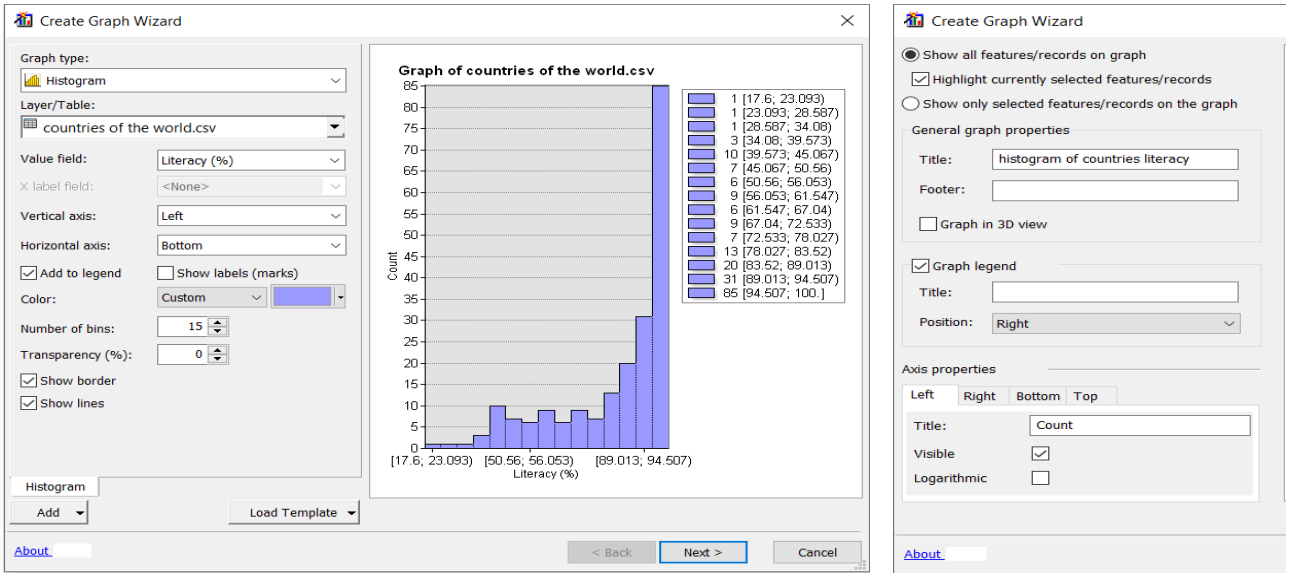
Step 2: Open graph's menu.





Click on "View" tab, then click on "Graphs" and finally press "Create Graph"

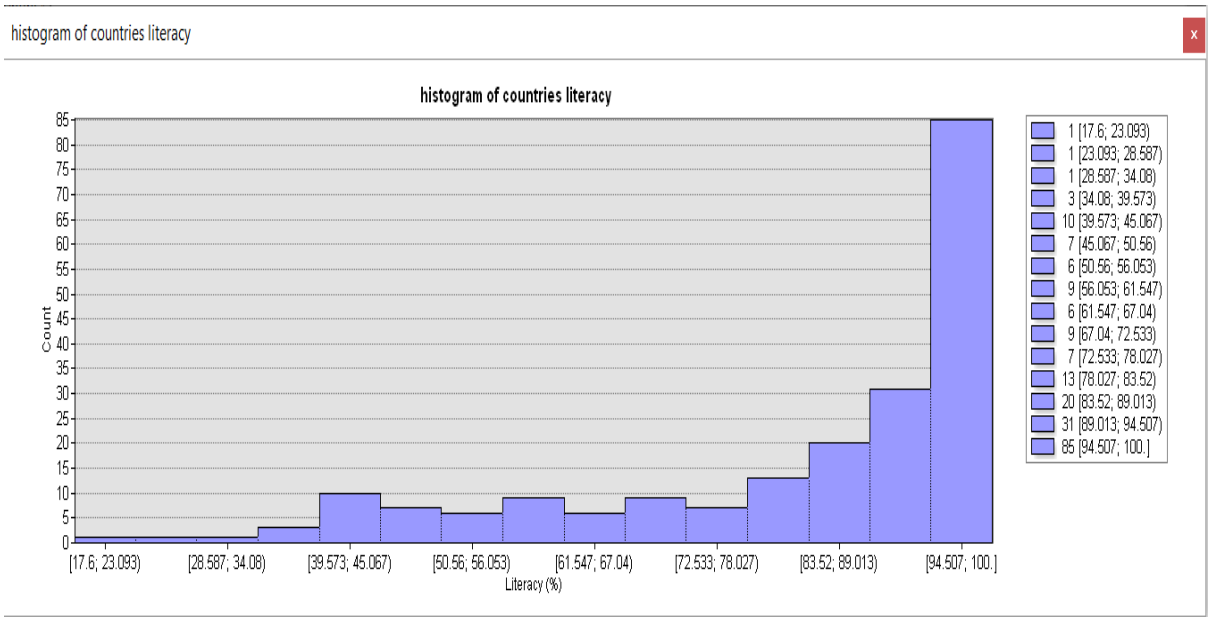
Step3: Create a histogram graph.



At first, we must choose graph type (“Histogram”). Then we add the layer or table that we want to plot (“countries of the world.csv”). After that we need a value field (“Literacy”). At last, we can select number of bins or parts in histogram graph (“15”). Other settings are useful for better interface.

And by pressing “Next” button the right menu will appear. We can create the histogram either by all records or just by selected records. Other settings are about General graph and axis properties, and are useful for better interface. By pressing “Finish” button the output graph will be executed.

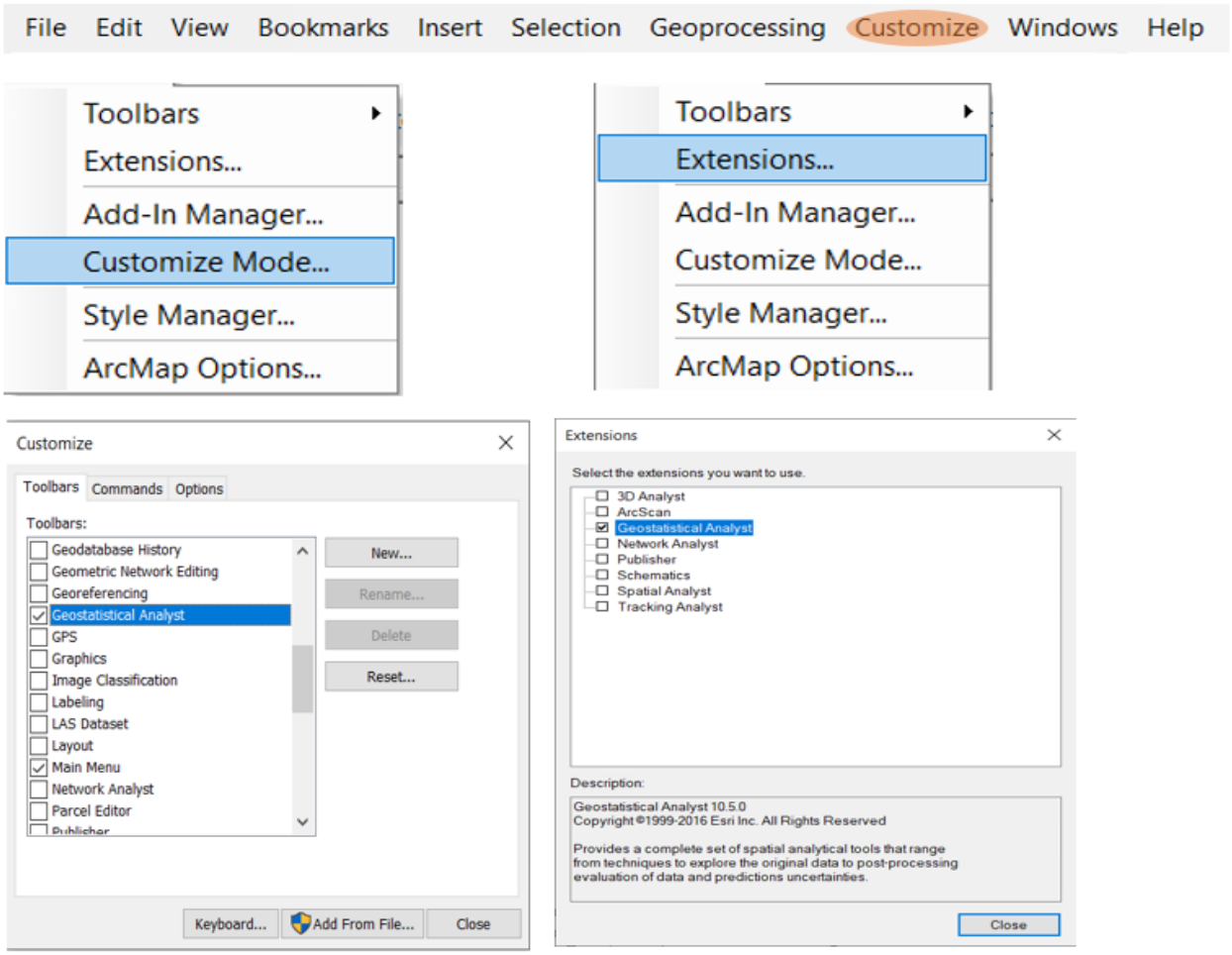
Results:



There is another way of creating histogram using ArcMap. In previous examples "Create Graph" function from "View" module used, in upcoming instance Geo-Statistical Analyst toolbars will be used.

For this part another dataset will be used, this dataset conation weather condition which collected from different stations across Canada country. Link to this data also provided and could be downloaded from here. In next part, step by step procedure will be described.

Step 0: activate Geo-Statistical Analyst toolbars.



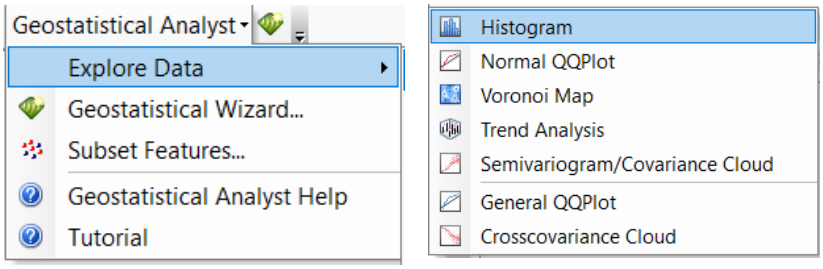
Click on "Customize" tab, then click on "Customize Mode" the left menu will be shown. In toolbars tab seek Geostatistical Analyst and turn on the checkbox button.

After that, Click on "Customize" tab, then click on "Extension" the right menu will be shown. Seek Geostatistical Analyst and turn on the checkbox button.

Step 1: adding data

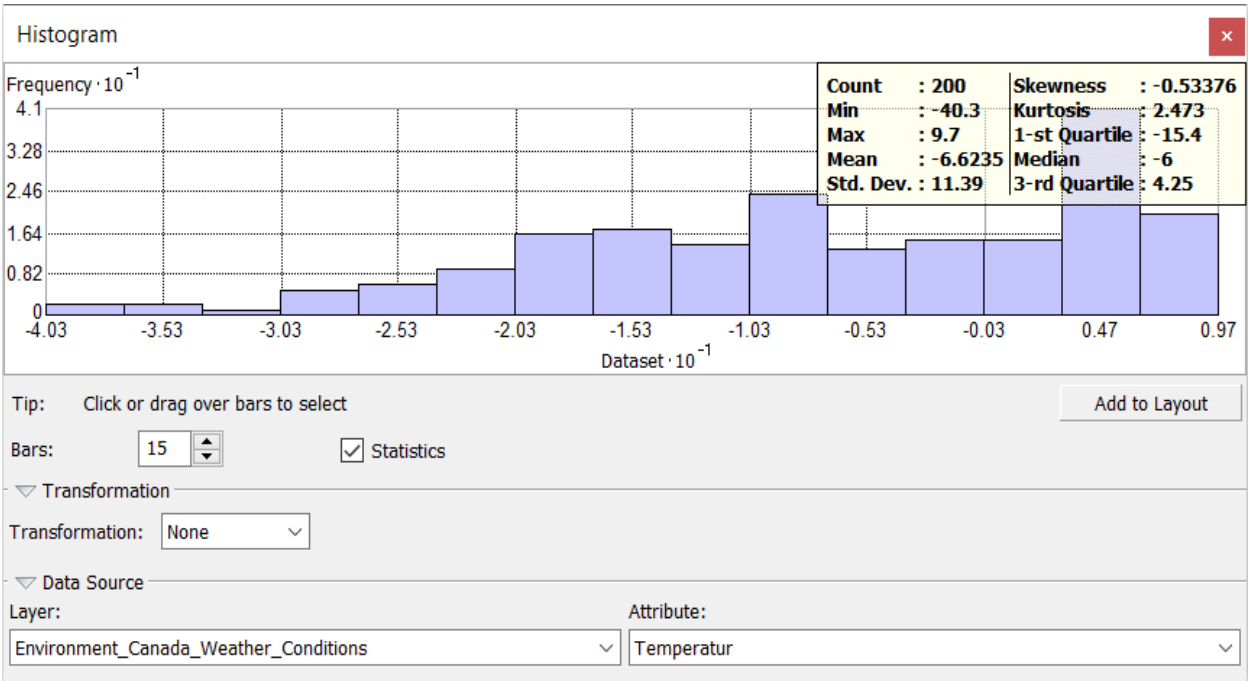
Click on “Add data”, then browse “Environment_Canada_Weather_Conditions” shapefile, and press Add button.

Step 2: open histogram menu.



Click on “Geostatistical Analyst” toolbar, then click on “Explore Data” and finally press “Histogram”.

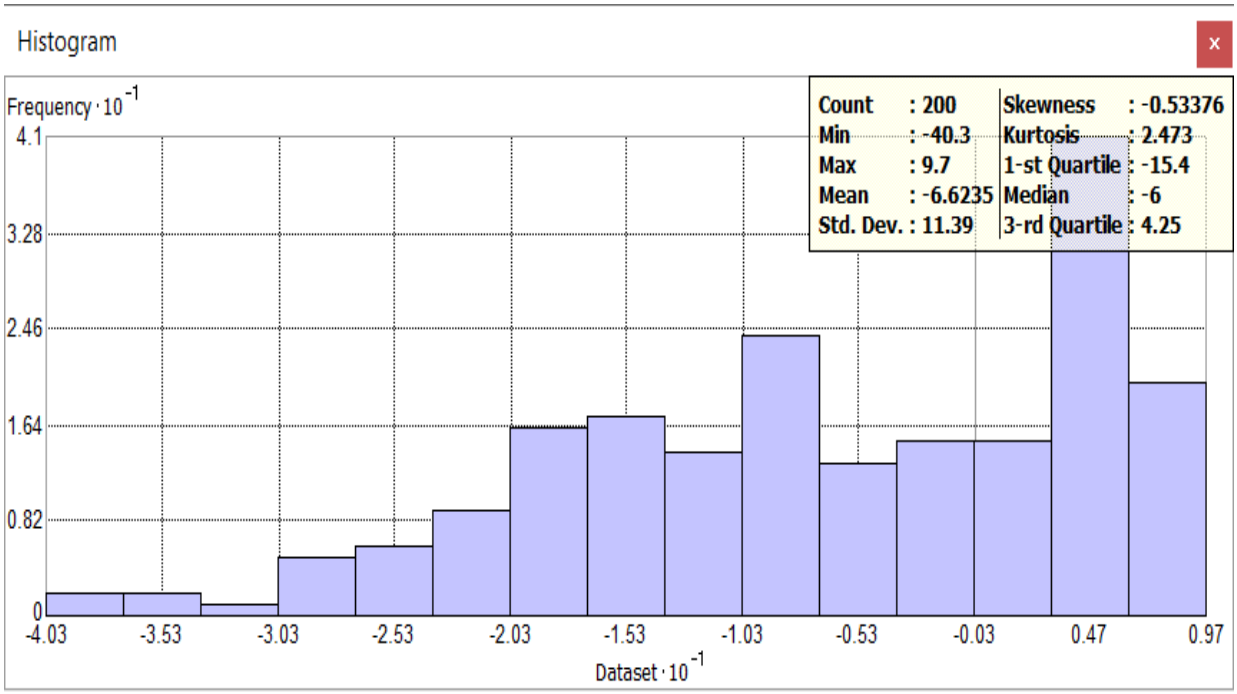
Step 3: Create a histogram.



First, we add the layer or table that we want to plot (“Environment_Canada_Weather_Conditions.shp”). After that target attribute must be chosen (“Temperatur”). then, we can select number of bars or parts in histogram graph (“15”) and a checkbox button is used for statistical information. At last, we can transform values in logarithm or without changes.

Previous histogram plot which used “view” menu was much more interactive and interface was one of the most important elements.

Result:



2.2. Boxplot

Another visual approach that is useful for summarizing a set of observations measured on an interval scale is the boxplot. The boxplot shows the shape of the distribution, the center of the data, and its dispersion. It is sometimes called a five-number graphic summary because the diagram specifically captures five statistical measures: the minimum and maximum values (range), lower and upper interquartile, and the median. This plot can be used to indicate whether the distribution is skewed or not, and the presence of outliers. Plotting the distribution for two or more groups allows for a comparative assessment of the observed patterns.

2.2.1 Boxplot in Python

As mentioned above, Python is full option and flexible environment for data visualization purposes. “Geopandas” library will be used to read “Canada weather condition” point type shape file which is available in here. If you encountered error while importing library try and use !pip install geopandas command.

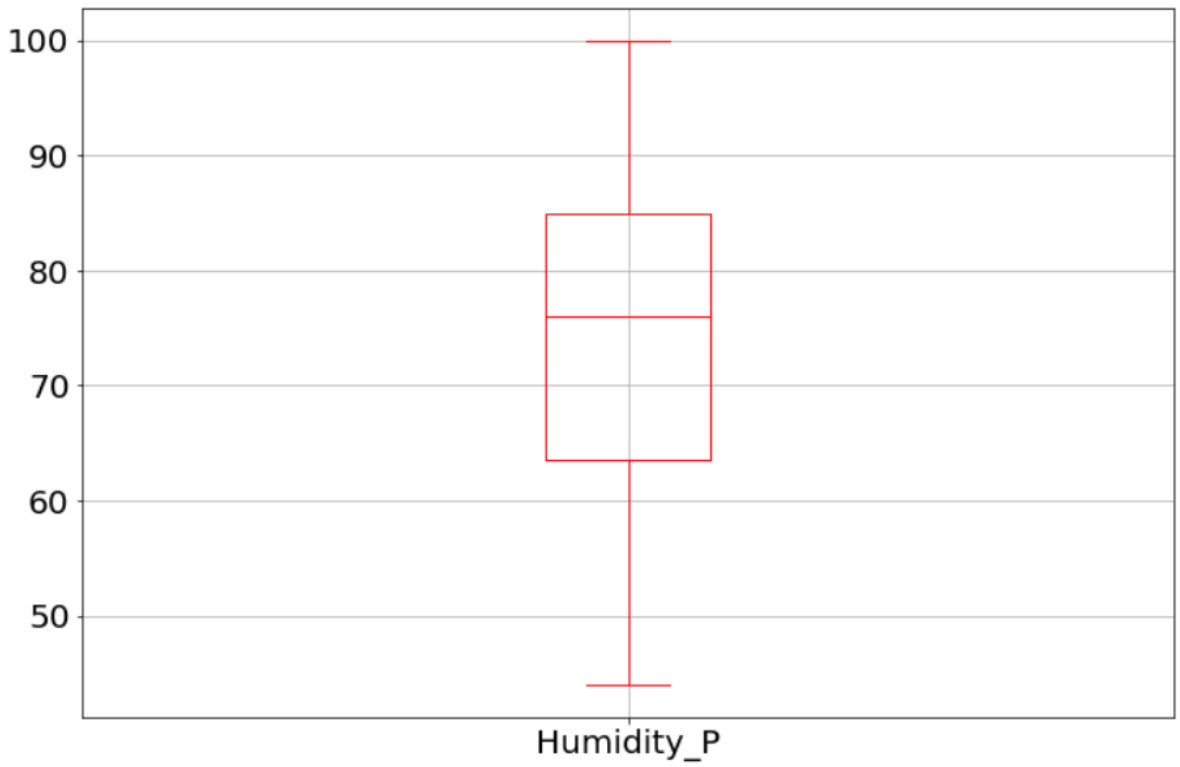
```
import geopandas as gpd
```

```
shpData = gpd.read_file("Environment_Canada_Weather_Conditions.shp")
```

By using shpData.dtypes command, attribute data format will be shown, since all data format are reasonable there is no need to change data format.

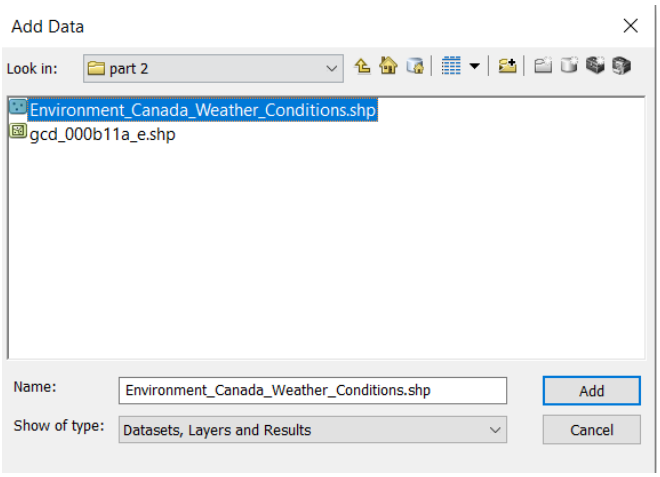
```
shpData.boxplot(column = "Humidity_P",  fontsize = 20.0 , figsize=(12,8), color='red')
```

By using boxplot function and giving the column that need to be plotted, the result will be produced. Other parameters in boxplot function is optional, like the color, or showing grid and others. Also it’s possible to create boxplot for all attribute in dataset which has numerical format type, it’s done just by not giving specific columns to boxplot.



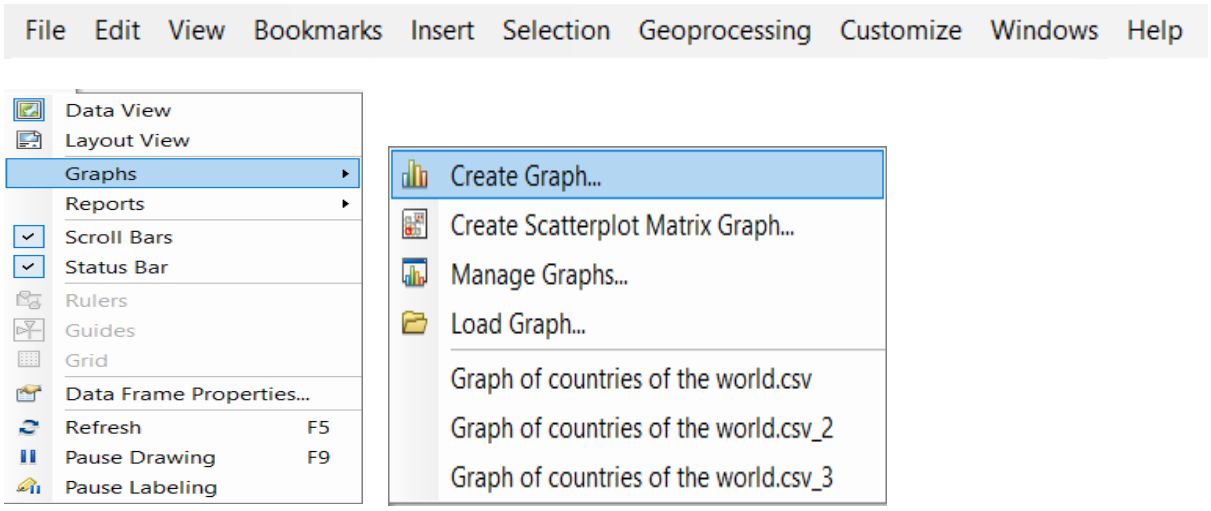
2.2.2. Boxplot in ArcGIS

In this part boxplot will be created by use of ArcMap from ESRI company, the used data are same as pervious part, Canada weather condition and humidity attribute. As always first step is loading data.



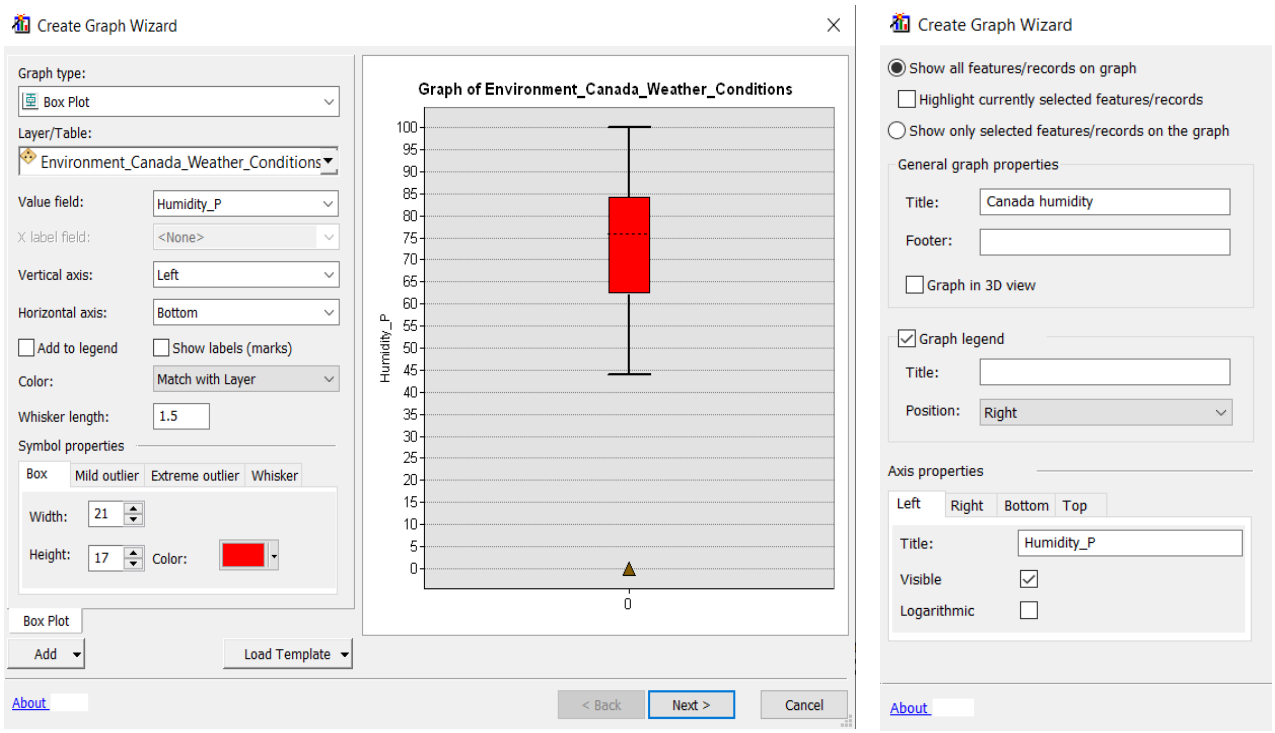
Click on “Add data”, then browse “Environment_Canada_Weather_Conditions” shapefile, and press Add button.

Step 2: Open graph's menu.

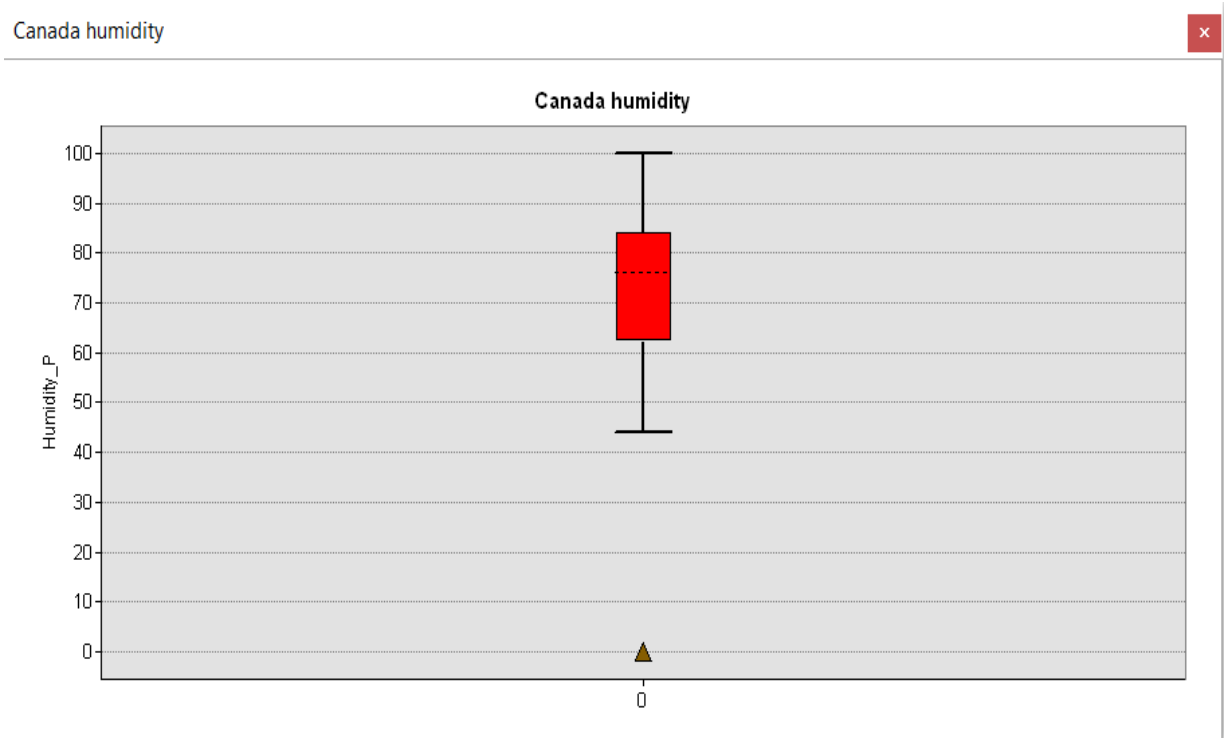


Click on "View" tab, then click on "Graphs" and finally press "Create Graph".

Step3: Create a box plot.



At first, we must choose graph type (“Box Plot”). Then we add the layer or table that we want to plot (“Environment_Canada_Weather_Conditions.shp”). After that we need a value field (“Humidity_P”). Other settings such as symbol properties are useful for better interface. By pressing “Next” button the right menu will appear. We can create the box plot either by all records or just by selected records. Other settings are about General graph and axis properties, and are useful for better interface. By pressing “Finish” button the output graph will be executed.



2.3. Scatterplot

Is a visual representation that shows the direction and strength of a relationship between the two variables, the dependent *Y* against independent *X*. Specifically, the scatterplot explores whether the values of *Y* vary systematically with the corresponding values of *X*. The plot can be based on raw scores obtained for the two variables or it can be based on residuals obtained after fitting a regression model. The patterns reveal statistical relationships or associations between two variables that manifest themselves by any non-random structure in the plot. *Y* is plotted on the *Vertical Axis* of the graph and represents the dependent/response variable whereas *X* is plotted on the *Horizontal Axis* of the graph and represents the independent/predictor variable. The plot can also serve as a useful diagnostic tool for assessing causal associations between variables. If a strong association exists in the data, then it suggests an underlying cause-and-effect mechanism. However, because this plot does not necessarily confirm the presence of a cause-and-effect, it is still mandatory on the statistician to draw knowledge from the underlying science to determine whether there is causality or not.

2.3.1. Scatterplot in Python

Pandas library on “World country” CSV file will be used. Dataset link accessible in first chapter.

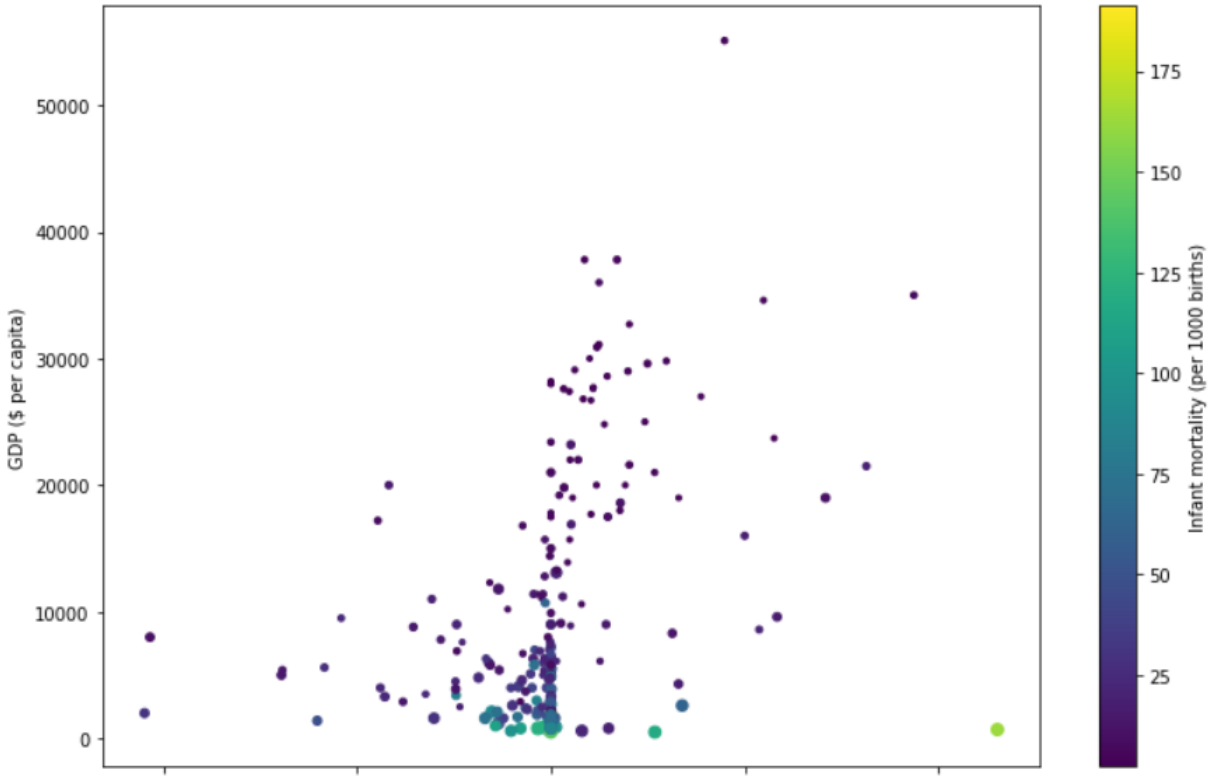
```
import pandas as pd
df= pd.read_csv("countries of the world1.csv")
```

After loading data, a loop will be implemented to change all data format in each possible column, Try Except is a way to avoid errors in programming with python.

```
for i in df.columns:
    try:
        df[i] = df[i].str.replace(",",".").astype(float)
    except:
        continue
```

Next step is to use scatterplot function provided inside “Pandas” library. As an example, four type of world countries attribute will be used. Two will be shown on X and Y axis, size of scatter circle will be indicating “Birthrate” and color bar on right side of plot will be “Infant mortality”. The final outcome shows that, net migration rate and countries GDP around the world has a dependency.

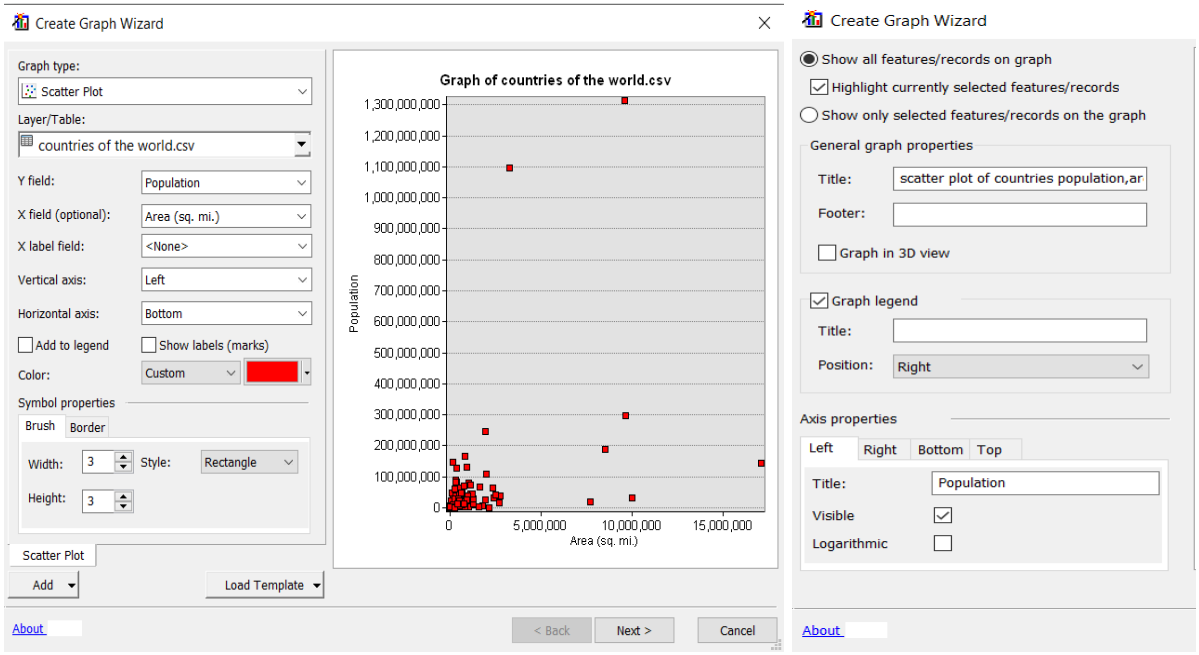
```
df.plot.scatter(x = 'Net migration',  y = 'GDP ($ per capita)',
                c = 'Infant mortality (per 1000 births)',
                s = 'Birthrate',  colormap='viridis')
```



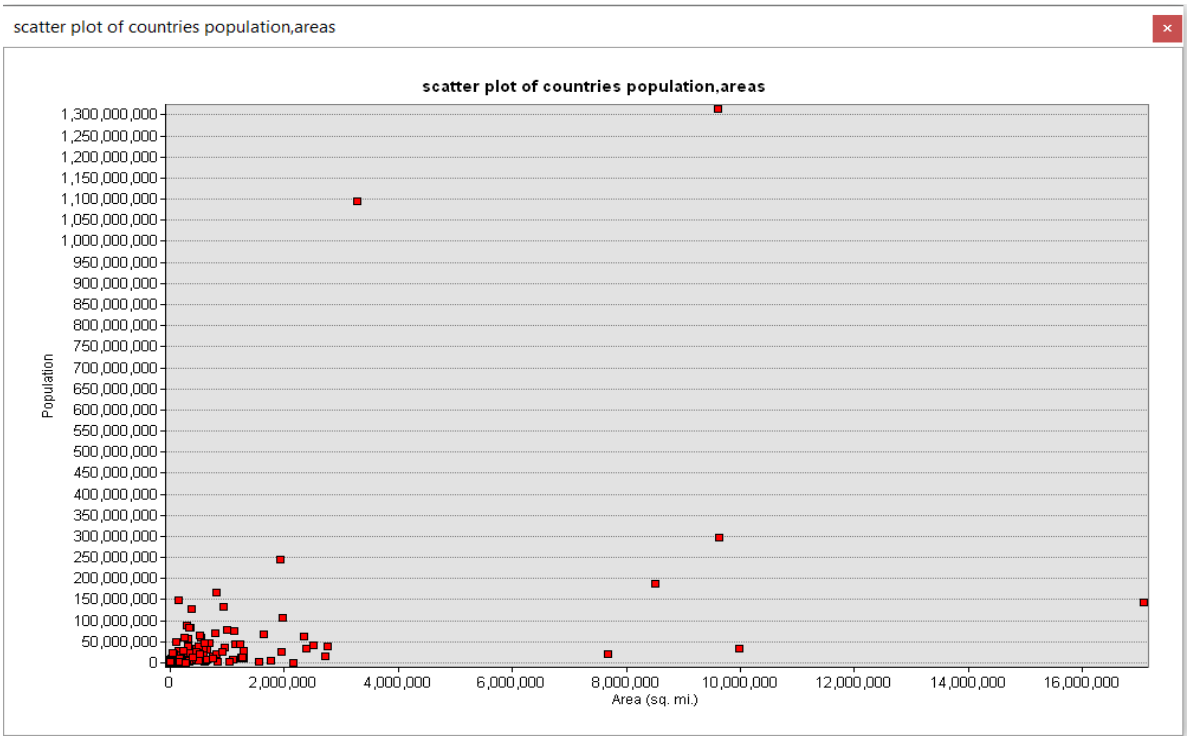
2.3.2. Scatterplot in ArcGIS

To create scatterplot in ArcMap first step procedure is same as Histogram’s first technique that been used, which is adding data and create graph method.

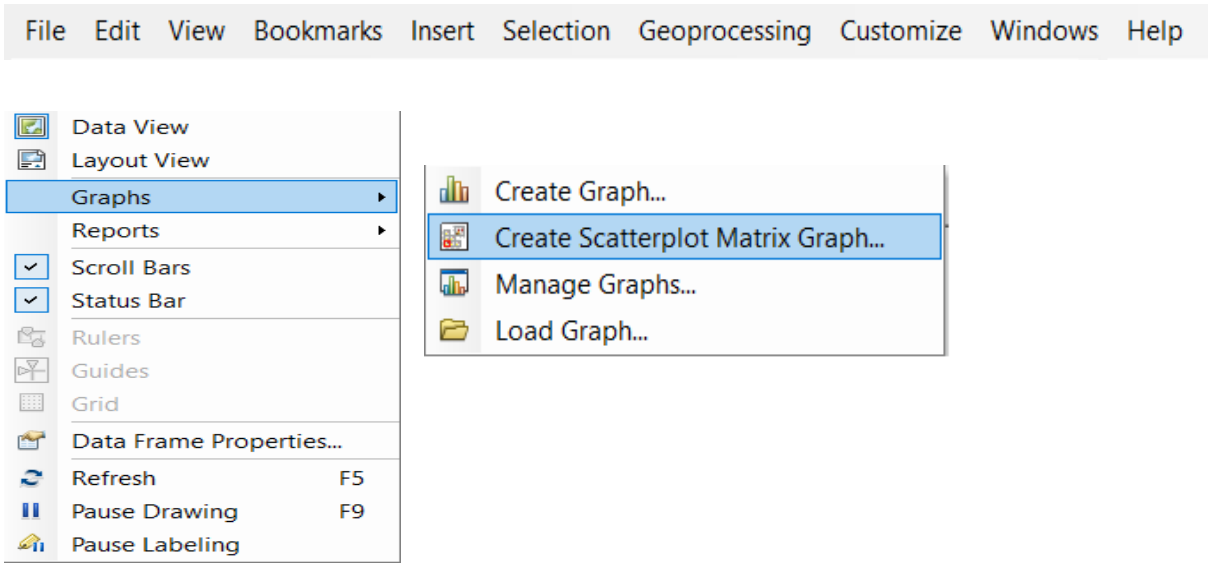
Step 3: create a scatterplot.



At first, we must choose graph type (“Scatterplot”). Then we add the layer or table that we want to plot (“countries of the world.csv”). After that we need a value fields (“Population” for Y field and “Area” for X field). Other settings such as symbol properties are useful for better interface. And by pressing “Next” button the right menu will appear. Remaining items are the same as previous example. *Results:*

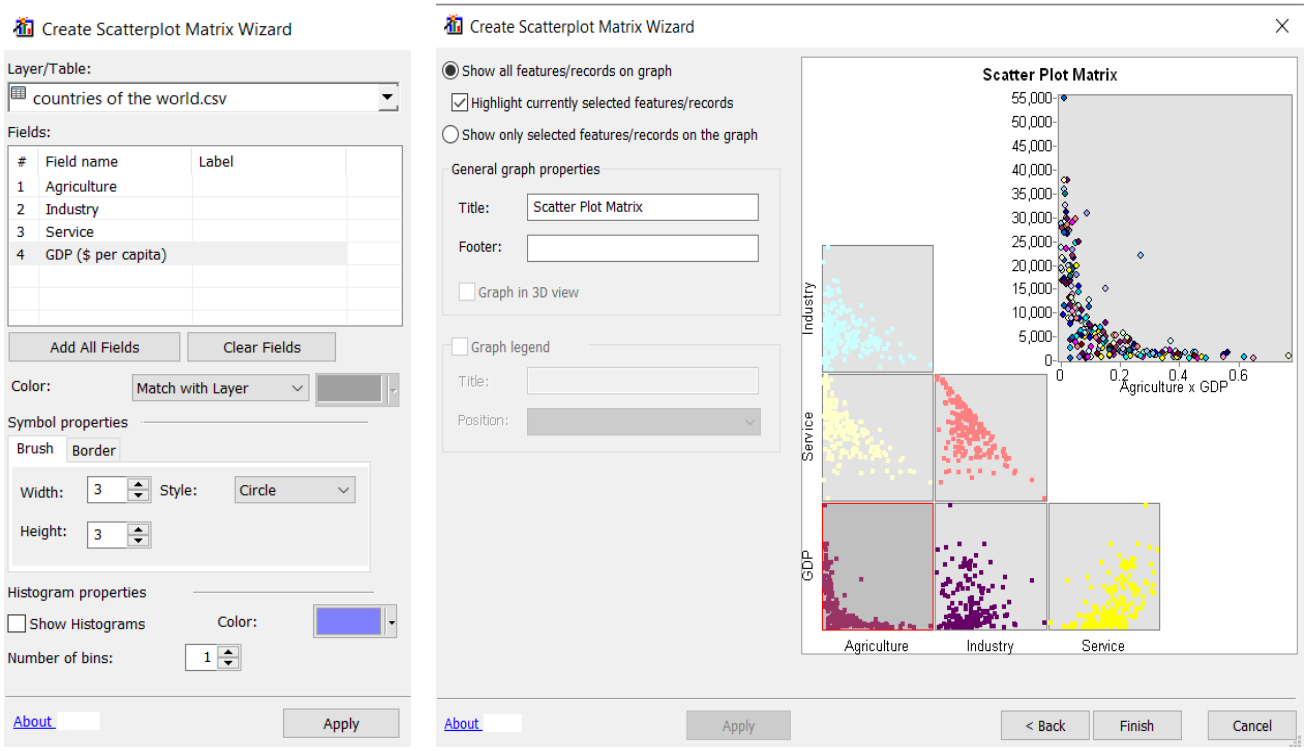


Like Histogram there is another way to generate scatterplot. This procedure first step is to add data and then from View *Open scatterplot matrix graph's menu.*



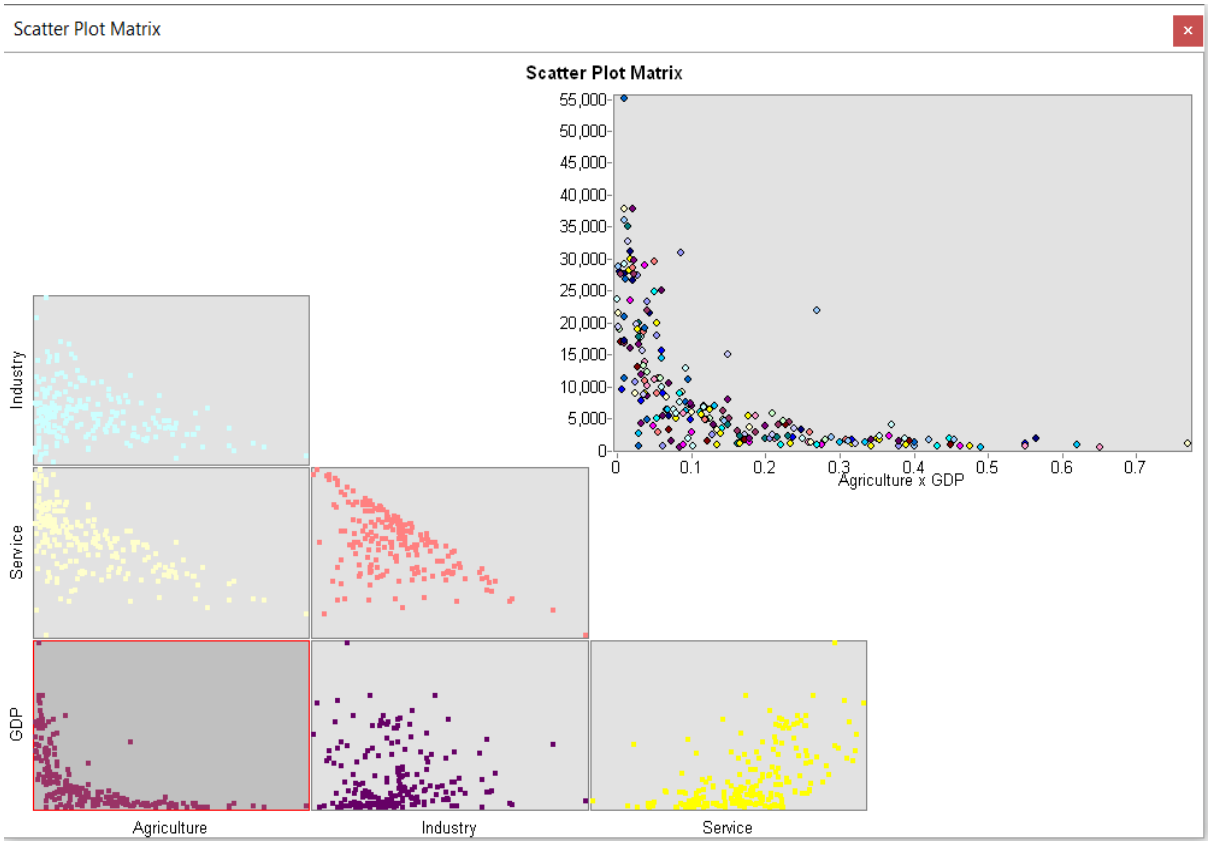
Click on "View" tab, then click on "Graphs" and finally press "Create a scatterplot matrix graph".

Step 3: Create a scatterplot matrix graph.



At first, we add the layer or table that we want to plot (“countries of the world.csv”). Then we must add the fields which is needed (“Agriculture, Industry, Service, GDP”). Other settings such as symbol properties are useful for better interface. And by pressing “Next” button the right menu will appear. Remaining items are the same as previous example. By pressing “Finish” button the output graph will be executed.

Results:



2.4. Trend Analysis

Spatial trend concept is very useful in order to depict the systematic variations of the phenomenon concerned over a region based on geographical locations or on two independent variables that may be any other two event records. Different types of spatial trend alternatives are presented visually and then their mathematical solutions under the title of trend surface analysis is presented with derivation of the necessary spatial regression analysis approach.

The Trend Analysis provides a three-dimensional perspective of the data. The locations of sample points are plotted on the x,y plane. Above each sample point, the value is given by the height of a stick in the z-dimension. A unique feature of the Trend Analysis tool is that the values are then projected onto the x,z plane and the y,z plane as scatterplots. This can be thought of as sideways views through the three-dimensional data. Polynomials are then fit through the scatterplots on the projected planes.

2.4.1. Trend analysis in Python

To generate spatial trend analysis as mentioned above, the procedure should be implemented half way through. As always first step is to load data, in this part “Canada weather condition” will be used.

```
shpData = gpd.read_file("Environment_Canada_Weather_Conditions.shp")
```

Next step is to get the geometry value or geographical coordinate of stations from shape file. Also “Temperature” attribute obtained from each station also will be assigned to another variable and null values of this attribute will be replaced by zero.

```
x, y = [], []
for i in shpData['geometry']:
  x.append(i.x)
  y.append(i.y)
z = np.abs(shpData["Temperatur"].fillna(0))
```

As Trend analysis is three dimensional scrutinize, the figure for drawing the outcomes should be defined. In this part “matplotlib” library and specifically “pyplot” module be utilized.

```
ax = plt.figure(figsize = (12,8)).gca(projection='3d')
```

After defining drawing environment, a rod for each z value which indicate temperature on stations will be plotted. This plot is done by for loop over station position and as tall as absolute temperature value. By use of “pyplot” 3D scatter of temperature values also be plotted.

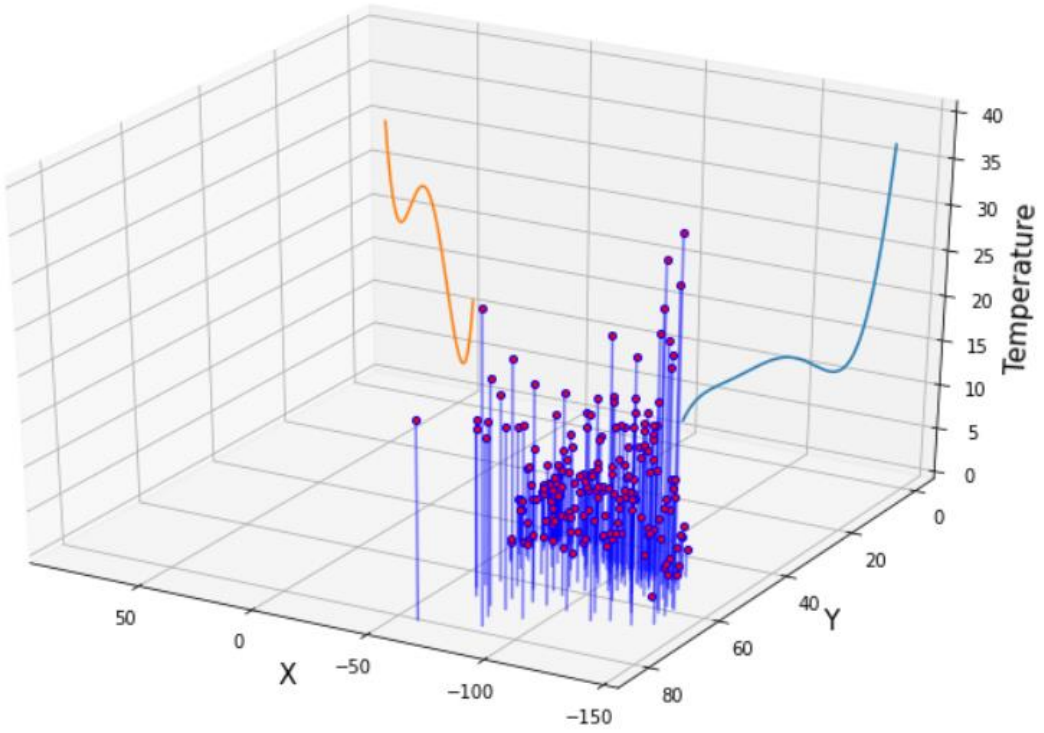
```
for xx, yy, zz in zip(x, y, z):
  ax.plot([xx, xx], [yy, yy], [0, zz], '-', color='b', alpha = 0.5)


ax.plot(x, y, z, 'o', markersize=4, markerfacecolor='r', color='b')
```

Main purpose of generating trend analysis is to draw fitting line for values on each direction and plane. Polynomial fitting modules in “numpy” library will be utilized to calculate a linear regression on value and one of directions. This lines also could be used to predict values in position where there is no data for temperature. The polynomial with adoptive degree for both x.z and y.z plane will be plotted.

```
xzM = np.poly1d(np.polyfit(x, z, 5))
xzL = np.linspace(min(x), max(x), len(z))

ax.plot(xzL, np.zeros_like(x), xzM(xzL))

yzM = np.poly1d(np.polyfit(y, z, 4))
yzL = np.linspace(min(y), max(y), len(z))

ax.plot(yzL, np.zeros_like(y), yzM(yzL))
```

At the end, plot will be shown from view that is comprehensible and final outcome will be plotted.

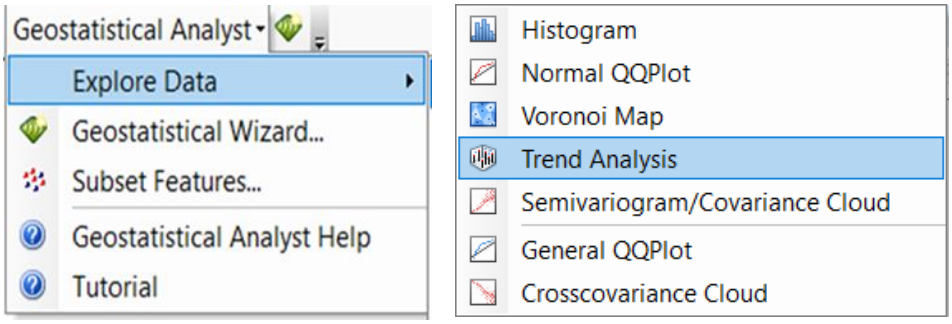
```
ax.view_init(azim = 120)
plt.show()
```



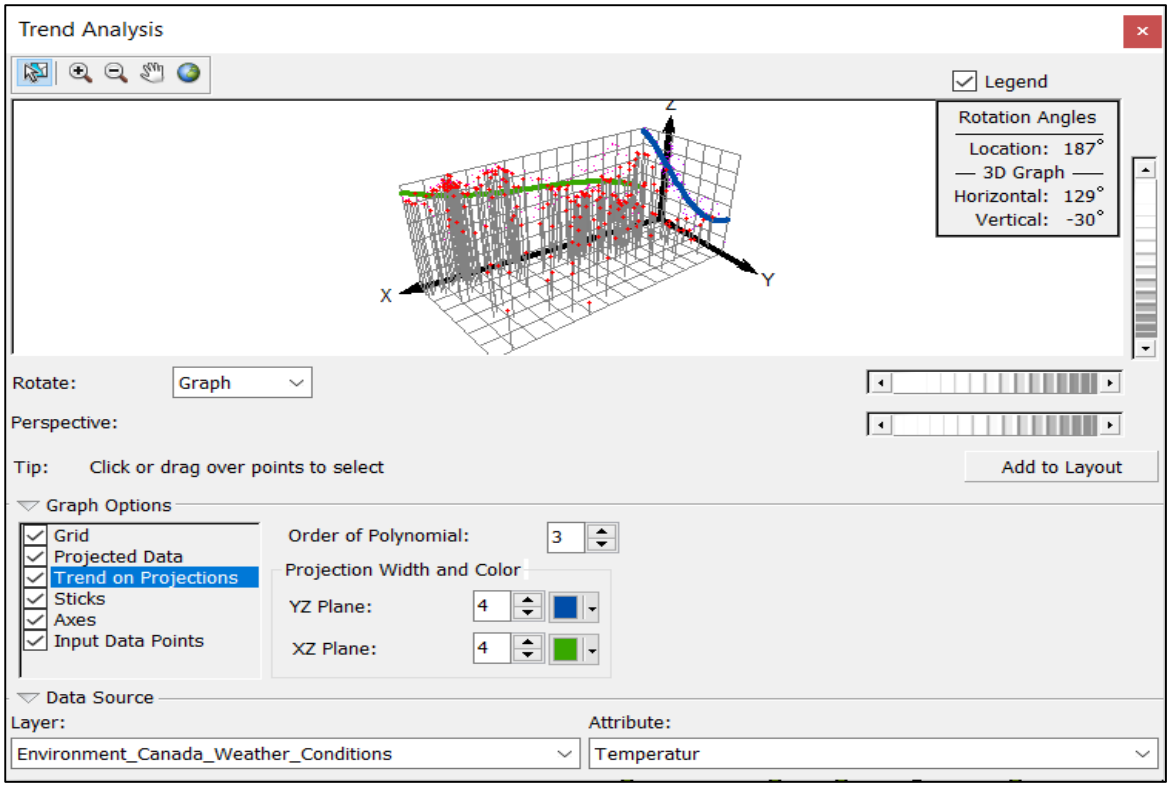
2.4.2. Trend analysis in ArcGIS

Trend analysis for spatial data is available in ArcMap too. As all example in ArcMap, first step is to add data, which in here, same as the Python "Canada weather condition" will be used.

Step 2: After activating Geostatistical Analyst Toolbars Click on "Geostatistical Analyst" toolbar, then click on "Explore Data" and finally press "Trend Analysis".

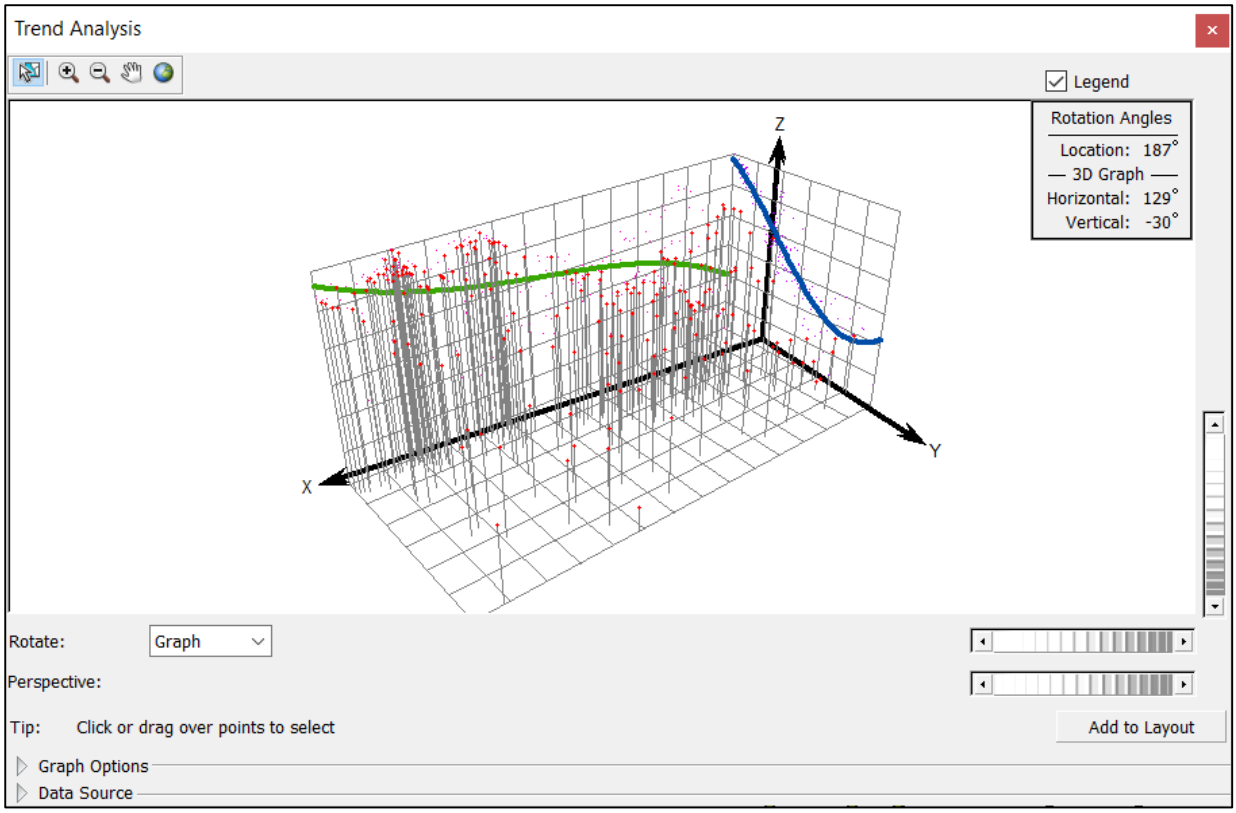


Step 3: Crate a trend analysis.



First, the layer or table that we want to plot will be added from dataset ("Environment_Canada_Weather_Conditions.shp"). After that target attribute must be chosen ("Temperature"). Graph Options menu is an important part in trend analysis, in "Grid" subpart number of grids and the line width is defined, in "Project Data" the projected data symbols and colors is defined, the most important subpart is "Trend on Projections" which the polynomial order is defined and the trend of temperatures could understood easily, in "sticks" subpart the stick properties between input data and XY plane can declare, in "Axes" subpart XYZ axis properties such as width and color is defined and in "input Data Points" subpart input data properties such as symbol size and symbol color is defined. At last, by clicking on scrollbars on right-side the plot will be rotate.

Final Result:



2.5. Clustering

Clustering is the task of dividing the population or data points into a number of groups such that data points in the same groups are more similar to other data points in the same group than those in other groups. In simple words, the aim is to segregate groups with similar traits and assign them into clusters.

Broadly speaking, clustering can be divided into two subgroups: Hard Clustering and Soft Clustering. In hard clustering, each data point either belongs to a cluster completely or not. In soft clustering, instead of putting each data point into a separate cluster, a probability or likelihood of that data point to be in those clusters is assigned.

Since the task of clustering is subjective, the means that can be used for achieving this goal are plenty. Every methodology follows a different set of rules for defining the 'similarity' among data points. In fact, there are more than 100 clustering algorithms known. But few of the algorithms are used popularly.

2.5.1. Clustering in Python

To generate a cluster analysis as mentioned above, the procedure should be implemented half way through. As always first step, needed libraries will be imported. "Geopandas" library will be used to read shape file and "sklearn.cluster" library will be used for clustering data. After importing the libraries, the data must be read.

```
import geopandas as gpd
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
```

```
shapefile = gpd.read_file("/content/drive/My Drive/data/Environment_Canada_Weather_Conditions.shp").dropna()
```

Then the data must be extracted by columns which is needed, these columns are: X, Y, Temperature. After that these three columns must be join and concatenate together. The concatenate data is array type which is useless in clustering analysis, so it must convert to a data frame for further analysis.

```
x = np.array([i for i in shapefile['geometry'].x])
y = np.array([j for j in shapefile['geometry'].y])
z = np.array([i for i in shapefile['Temperatur']])
X = np.array([x,y,z])
df = pd.DataFrame(data=np.transpose(X))
```

Next, by defining number of classes, K-means clustering will be useable. The K-means has an important property called “fit_predict” which cluster data and labelling them according to the number of classes. Another important property is “kmeans.cluster_centers_” which gain us to the clusters centroids.

```
kmeans = KMeans(n_clusters= 4)
label = kmeans.fit_predict(df)
centroids = kmeans.cluster_centers_
```

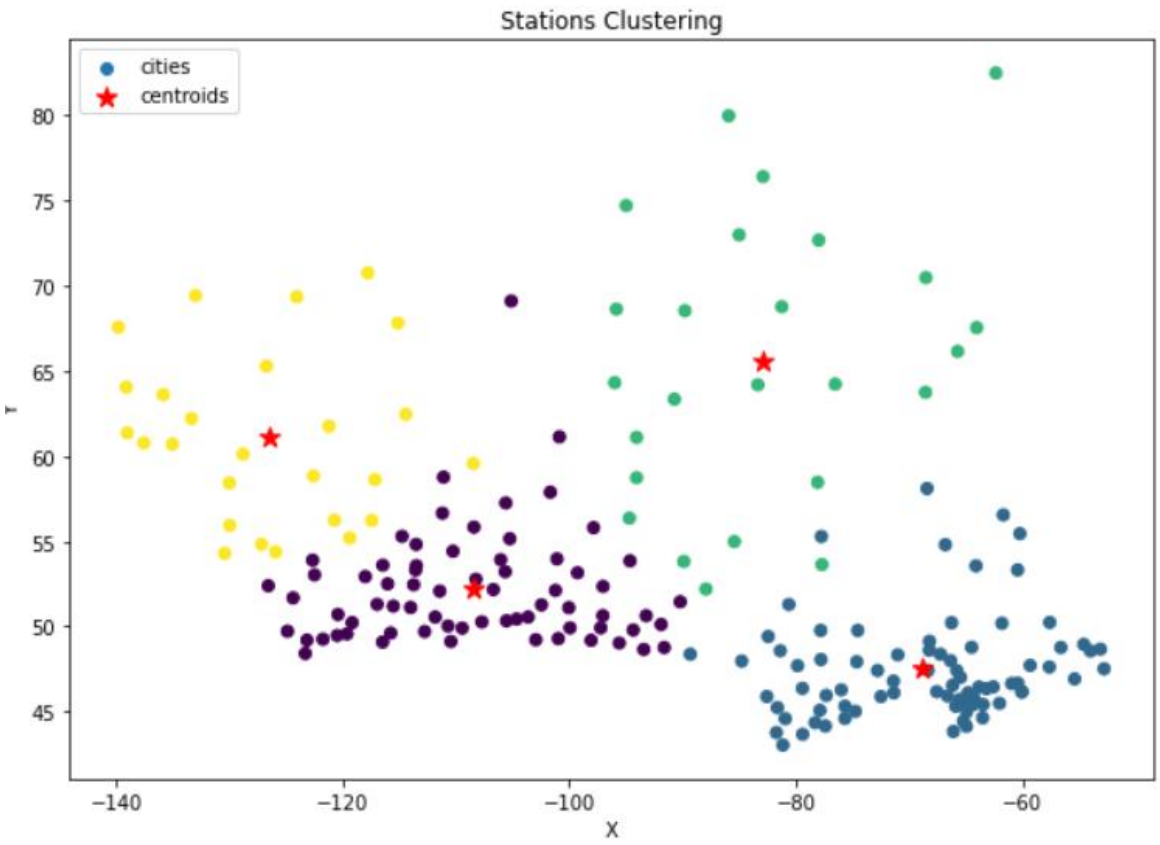
The centre of cluster’s is shown below. Where first column is X coordinates, the second column is Y and third column is temperature.

```
[[ -82.9379527    64.63986622  -17.7       ]
 [ -68.68979641   47.41951189    3.96153846]
 [-108.71316776   52.17945307   -8.32794118]
 [-126.56320708   61.14714692  -22.79230769]]
```

At last, the labeled data is shown by a scatter plot. Each color is a representation of a cluster, so same colors are in a same cluster. Also, the clusters centroid is added to the plot for better understanding.

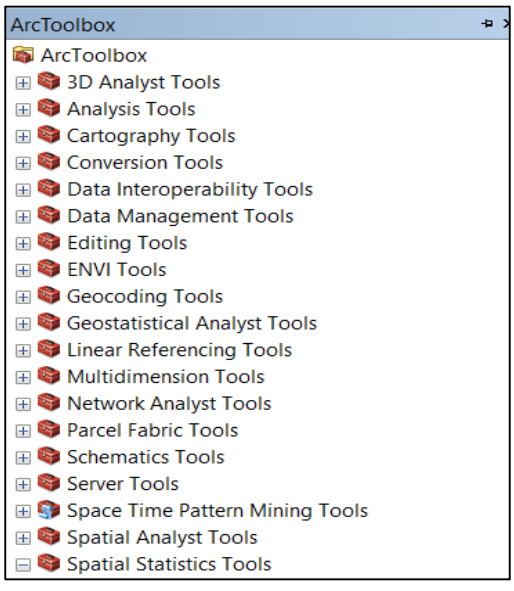
```
plt.figure(figsize=(10,7))
plt.scatter(x, y,c=label)
plt.scatter(centroids[:,0],centroids[:,1], marker='*', s=120, color='red')
plt.legend(['cities','centroids'])
plt.title('Stations Clustering')
plt.xlabel('X')
plt.ylabel('Y')
plt.show()
```

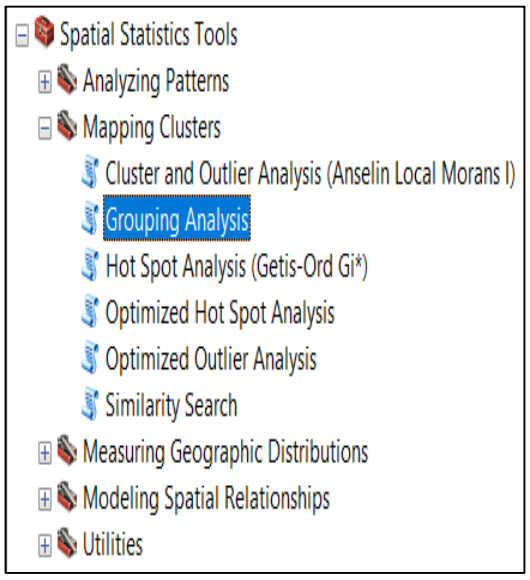
The final result is shown below



2.5.2. Clustering in ArcGIS

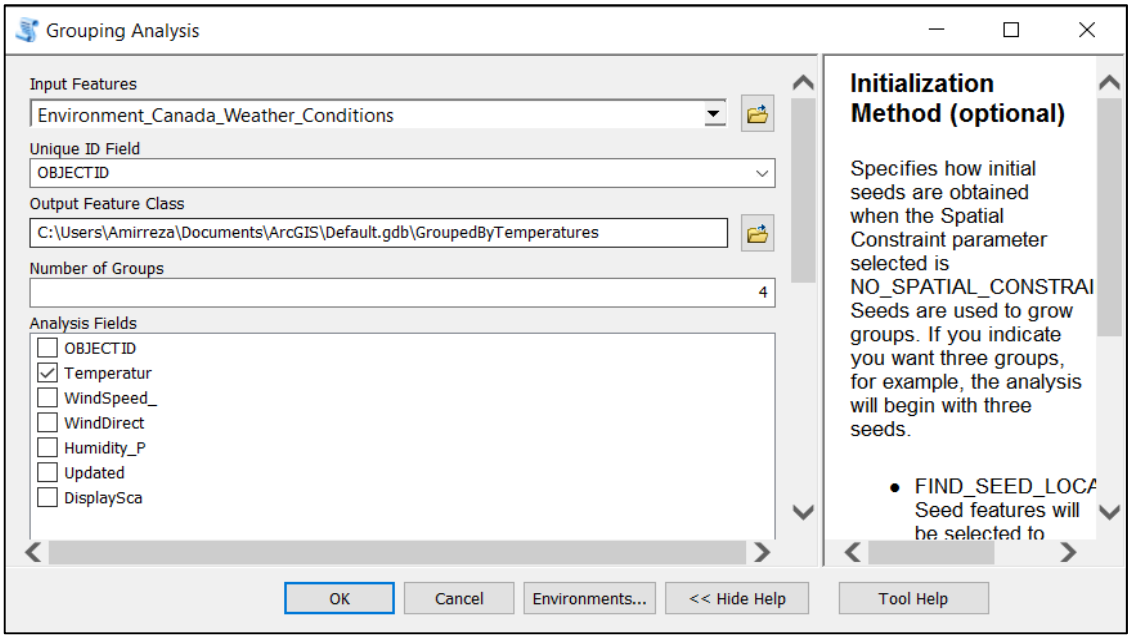
For clustering data in ArcGIS many modules are available. To create a map which is grouped by the temperatures we'll use Group Analysis. As all example in ArcMap, first step is to add data, which in here, "Canada weather condition" will be used.*Step 2: Open Grouping Analysis menu.*





At first, open “ArcToolbox” and find “Spatial Statistics Tools” toolbox. Then, click on “Mapping Clusters” menu and open “Grouping Analysis” analysis.

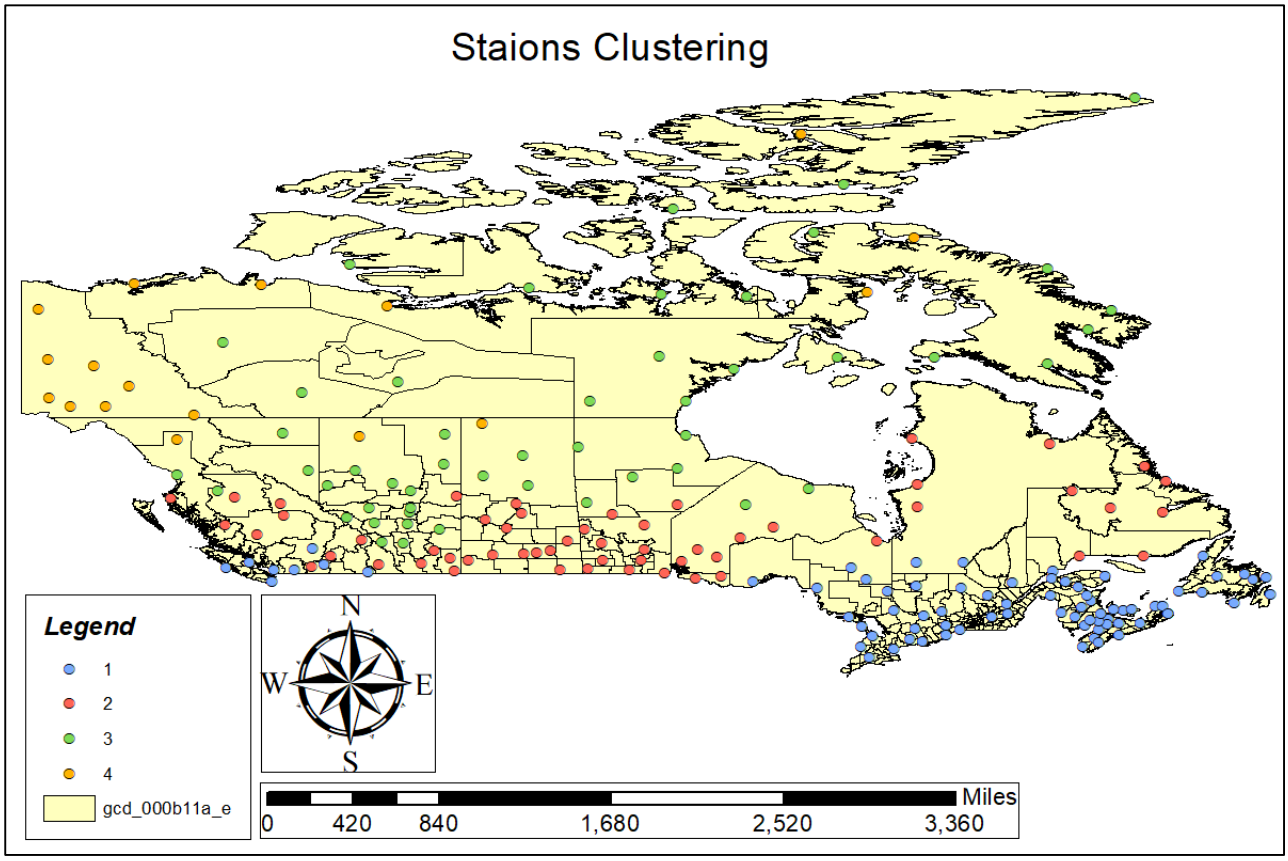
Step 3: Create a grouped feature base on temperatures.



First of all, in Input point features the weather stations shapefile (“Canada weather condition”) must be added. Then Unique ID Field which is an integer field containing a different value for every feature in the input feature class is needed (“OBJECTID”). Next, Number of Groups must be defended; it can be form 2 groups up to 15 groups (“4”). In Analysis Field a list of fields you want to use to distinguish one group from another must be defined (“Temperatur”).

In Spatial_Constraints you can specifies if and how spatial relationships among features should constrain the groups created (“NO_SPATIAL_CONSTRAINT”). By choosing constrains some parts such as Distance Methods, Numbers of Neighbors and Weights Matrix File must defined. Then, Initialization Field specifies how initial seeds are obtained when the constrains parameter selected is NO_SPATIAL_CONSTRAINT. Also, a report can be created by adding path in Output Report File. At last by pressing “OK” button, the map will produce.

Result:


Staions Clustering
Legend
1
2
3
4
gcd_000b11a_e
N
W
E
S
Miles
0
420
840
1,680
2,520
3,360

2.6. Normal QQ plot

The Normal QQ plot or Q-Q plot, or in non-abbreviated words quantile-quantile plot, is a graphical tool to help in assessing if a set of data plausibly came from some theoretical distribution such as a Normal or exponential. For example, after running statistical analysis that assumes dependent variable is Normally distributed, Normal Q-Q plot could be used to check that assumption. It's just a visual check, not an air-tight proof, so it is somewhat subjective. But it allows to see at-a-glance if assumption is plausible, and if not, how the assumption is violated and what data points contribute to the violation.

A Q-Q plot is a scatterplot created by plotting two sets of quantiles against one another. If both sets of quantiles came from the same distribution, seeing that points forming a line that's roughly straight. In a QQ plot if the data is normally distributed, the points in the QQ-normal plot lie on a straight diagonal line. The deviations from the straight line are minimal. This indicates normal distribution or Normal QQ plot.

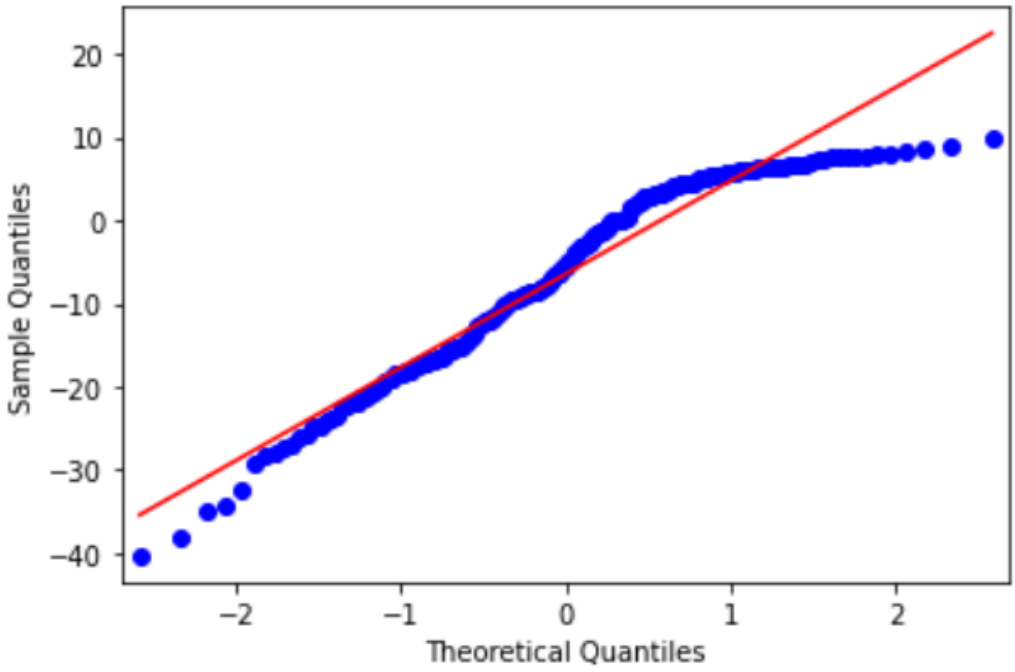
2.6.1. Normal QQ plot in Python

Used data for generating Normal QQ plot in python is same as previous ones, "Environment_Canada_Weather_Conditions.shp" which is accessible in here. For this part "statsmodel.api" module will be used. This module provides classes and functions for the estimation of many different statistical models, as well as for conducting statistical tests, and statistical data exploration. As always after importing libraries first step is to read data.

```
import statsmodels.api as sm
import geopandas as gpd

shpData = gpd.read_file("Environment_Canada_Weather_Conditions.shp")
```

Next step is to provide attribute that should be plotted. In this example "Temperature" of Canada will be used. From "statsmodel", "qqplot" function will be called, by giving prepared data and type of line for fitting on data, the normal QQ plot will be generated.

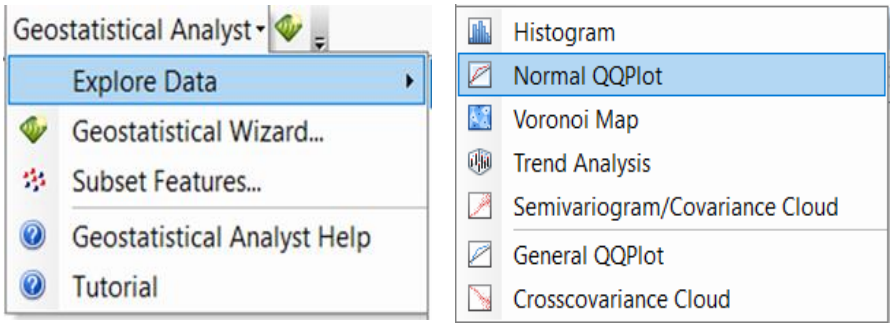
```
data_points = shpData['Temperatur'].fillna(0)

sm.qqplot(data_points, line="r")
```



2.6.2. Normal QQ plot in ArcGIS

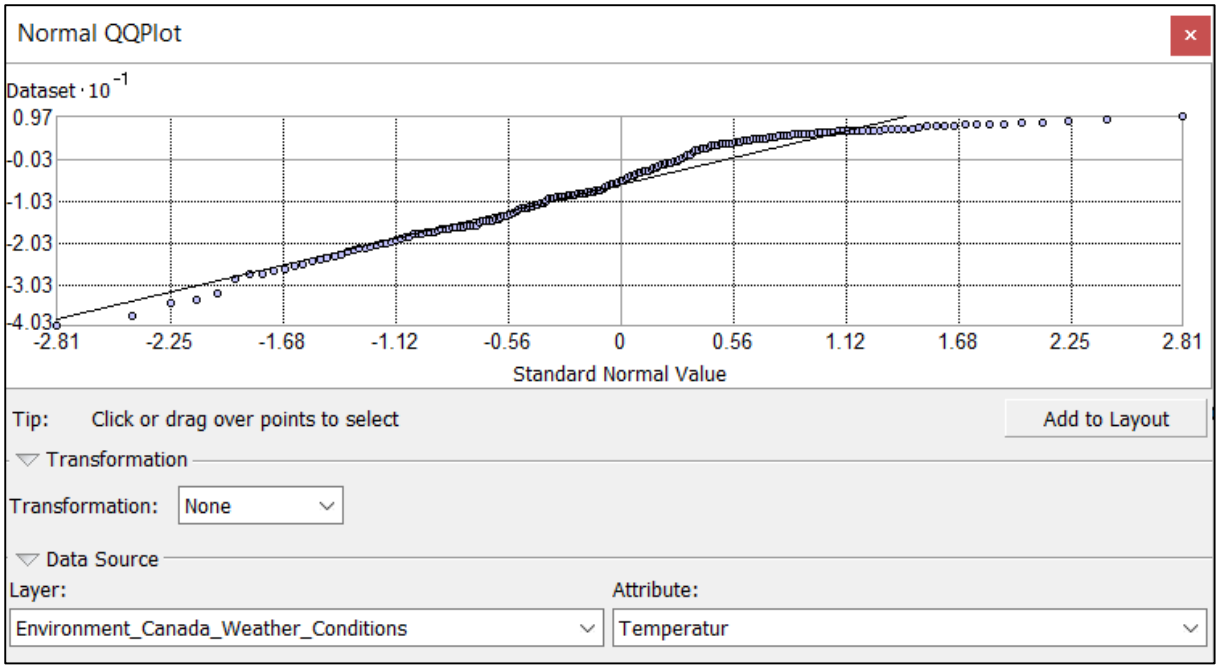
ArcMap of ESRI company also has some option for Normal QQ plotting. After adding data, which in here, same as the Python "Canada weather condition" will be used, next step is activating Geostatistical Analyst Toolbars.

Step 2: open Normal QQ menu.



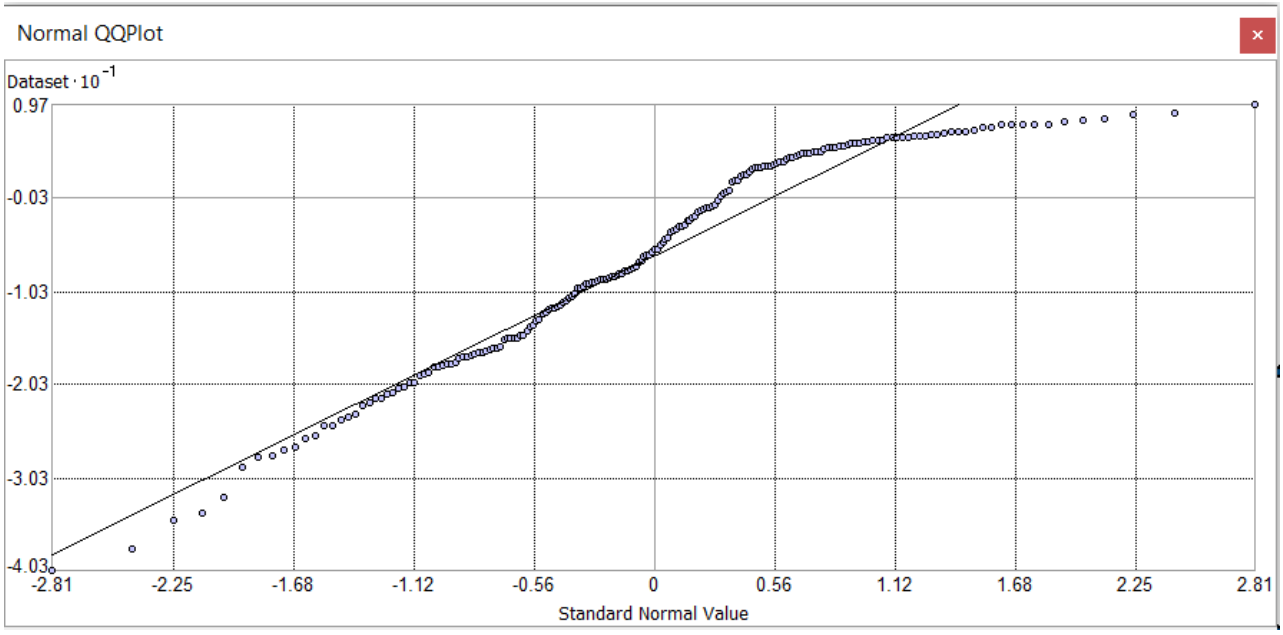
Click on "Geostatistical Analyst" toolbar, then click on "Explore Data" and finally press "Normal QQ".

Step 3: Create a Normal QQ plot.



First, we add the layer or table that we want to plot ("Environment_Canada_Weather_Conditions.shp"). After that target attribute must be chosen ("Temperatur"). At last, we can transform values in logarithm or without changes.

Result:



2.7. Voronoi map

Voronoi map or Voronoi diagram, in mathematics, is a partition of a plane into regions close to each of a given set of objects. In the simplest case, these objects are just finitely many points in the plane. For each point there is a corresponding region, called a Voronoi cell, consisting of all points of the plane closer to that main point than to any other.

Voronoi maps are fundamental tools for uncovering the geometric structures that underlie spatial data. They have been used in applications that draw from the location of point features to delineate space into so-called “spheres of influence” such as trade areas in the retail industry, hospital service areas, and more. The Voronoi method offers an excellent example of how spatial analysis builds upon the synergies between multiple analytical domains including mathematics, computational geometry, and geographic information systems.

2.7.1. Voronoi map in Python

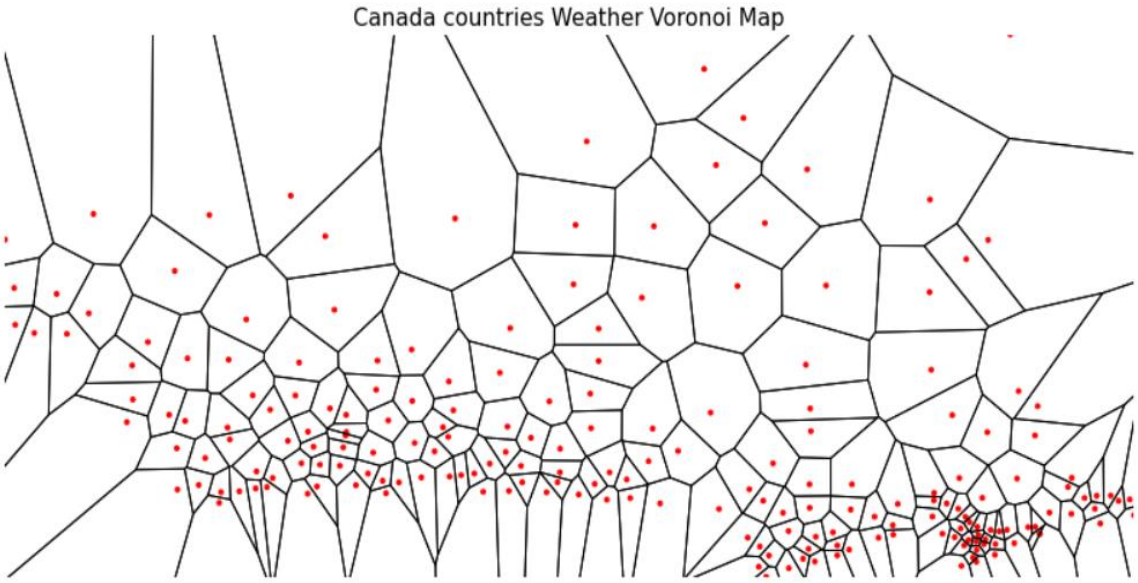
There are some plausible options to use for generating Voronoi map with Python. “Geoplot” is the next python module in that used in this tutorial. This library has Voronoi diagram calculator and plotter in itself. To plot map first need to install this library and load the data.

```
import geopandas as gpd
import geoplot as gplt
import matplotlib.pyplot as plt

shpData = gpd.read_file("Environment_Canada_Weather_Conditions.shp")
```

Next phase is to plot Voronoi map, in this example, only point data used but it’s good to also plot polygon of data. After applying Geoplot’s Voronoi function to “Canada Weather” shape file by using pointplot function, Voronoi cell for each station will be plotted.

```
ax = gplt.voronoi(shpData, figsize = (14,12))

gplt.pointplot(shpData, color='red', ax=ax, s=3, extent=shpData.total_bounds)

plt.title('Canada countries Weather Voronoi Map', fontsize = 15);
```

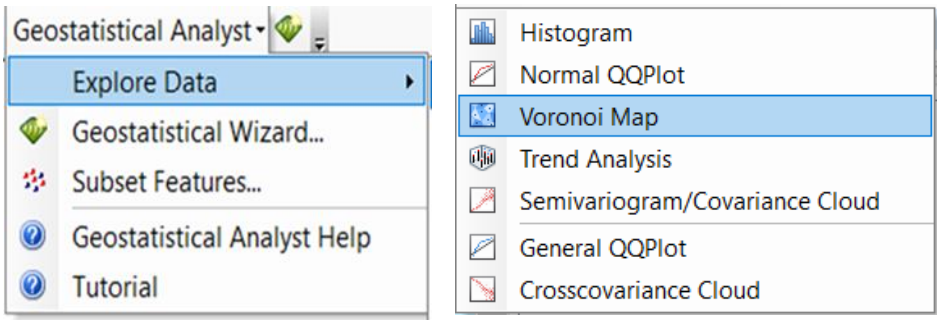
The final result of using only point data is can be seen in next figure, red points are Meteorological stations and polygons are cell of Voronoi map for them:



2.7.2. Voronoi map in ArcGIS

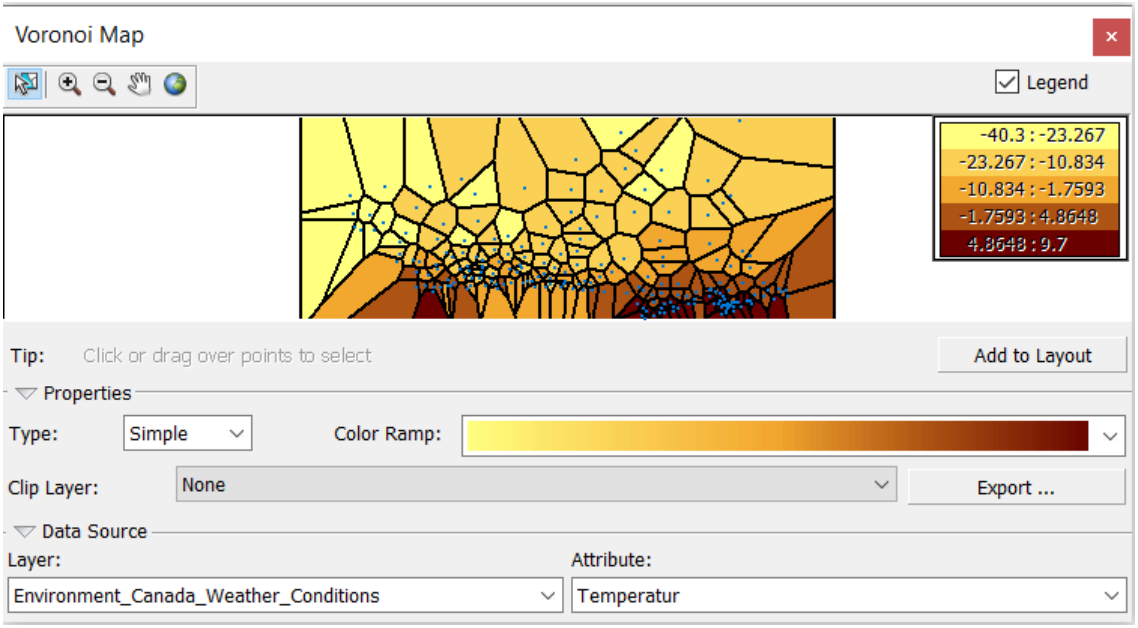
To create a Graduated colours Voronoi map for Canada temperatures in ArcMap. used data, which in here are “Canada weather condition” and Canada country boundaries named “gcd_000b11a_e.shp” will be used, the “gcd_000b11a_e” could be downloaded from here as zip file. Next step is activating Geostatistical Analyst Toolbars.

Step 2: open Voronoi Map menu.



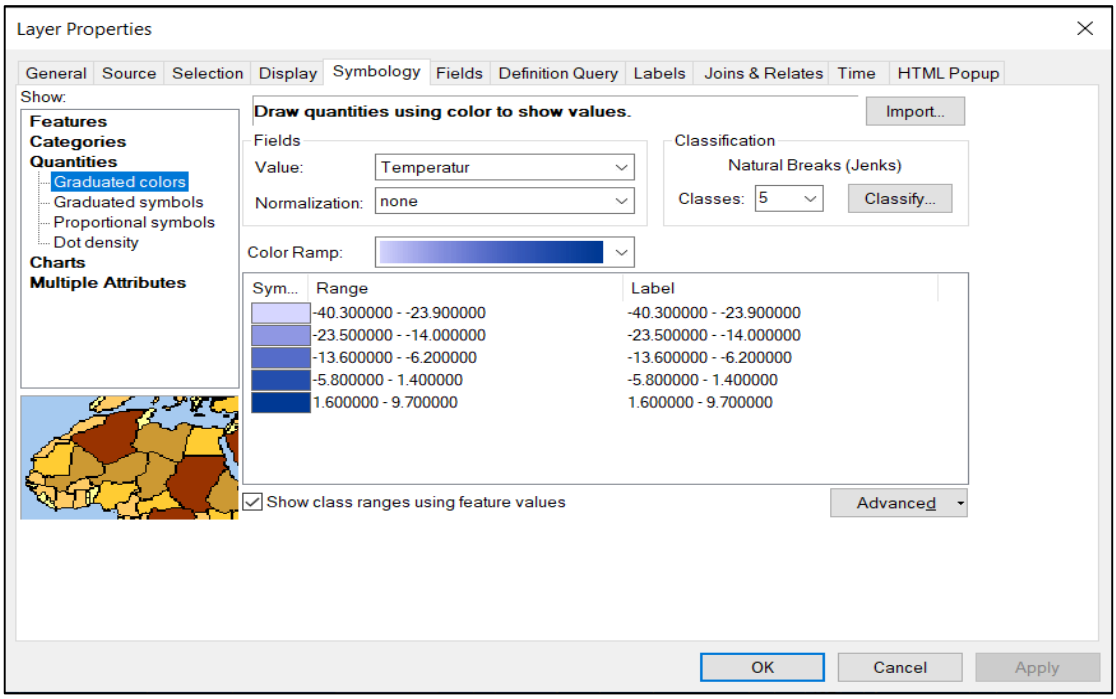
Click on “Geostatistical Analyst” toolbar, then click on “Explore Data” and finally press “Voronoi Map”.

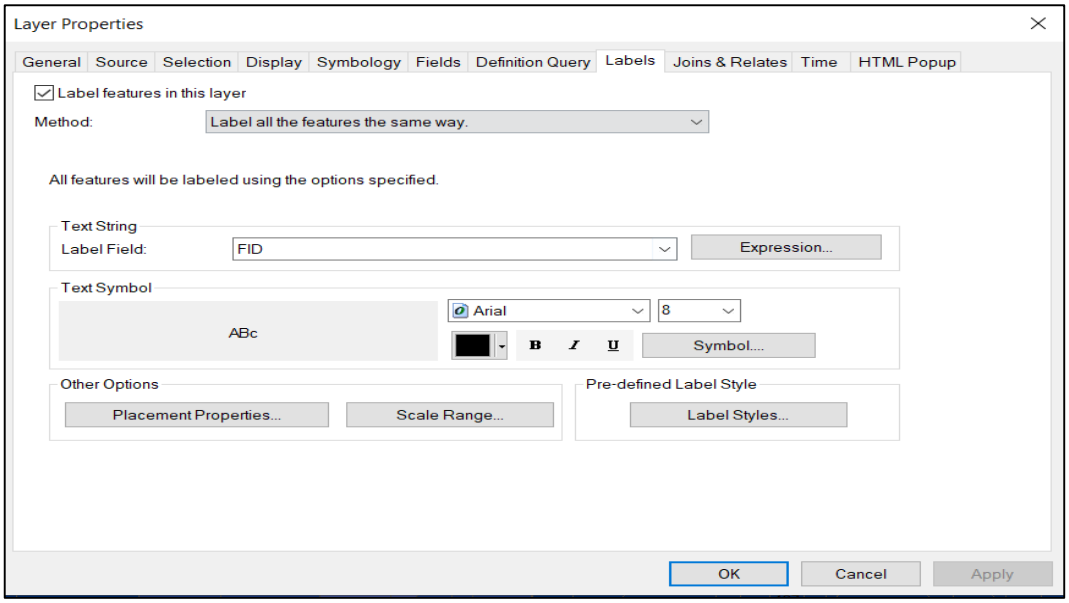
Step 3: Create a Voronoi map.



First, we add the layer or table that we want to plot ("Environment_Canada_Weather_Conditions.shp"). After that target attribute must be chosen ("Temperatur"). In properties subpart type of classification must be defined. At last, by pressing "Export" button we can save this map and export it as shapefile, geodatabase and etc. in this example we Exported the result in .shp format.

Step 4: Create a Graduated colors Voronoi map.

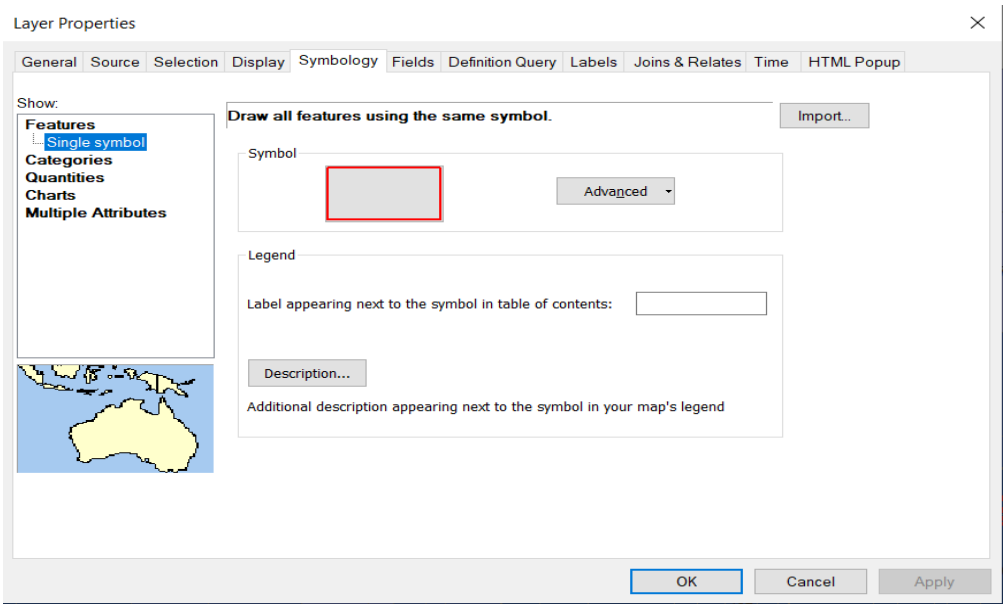


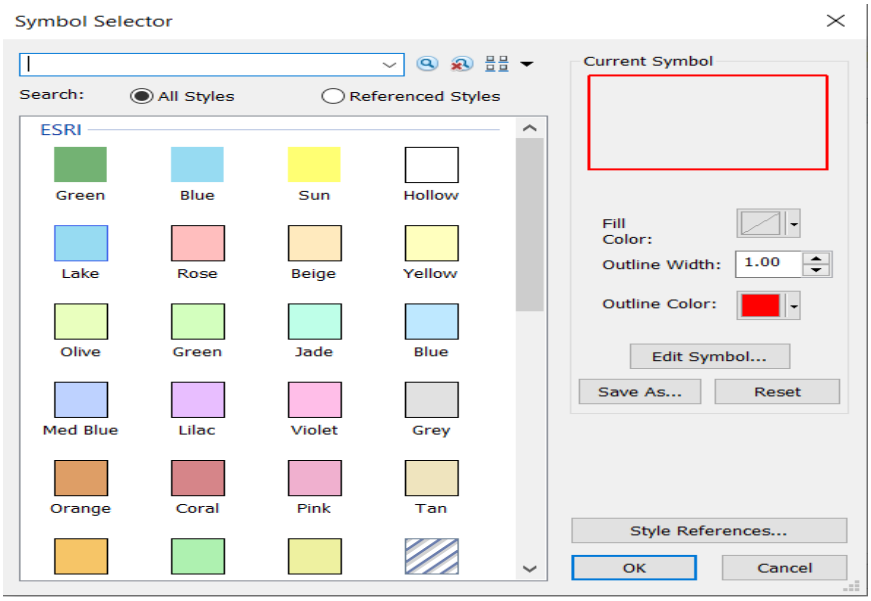


Right click on "voronoi.shp", click on "Properties" and Seek "Symbology" tab. This sub menu is one of the most important menus for creating map in ArcMap environment. On the left part (show:) map type must be chosen. As our field (temperature) is a quantitative value, we seek Quantities and select "Graduated colours". Then, on the middle part the field must be chosen ("Temperatur"). At last, on the right-side number of classes must be declared ("5").

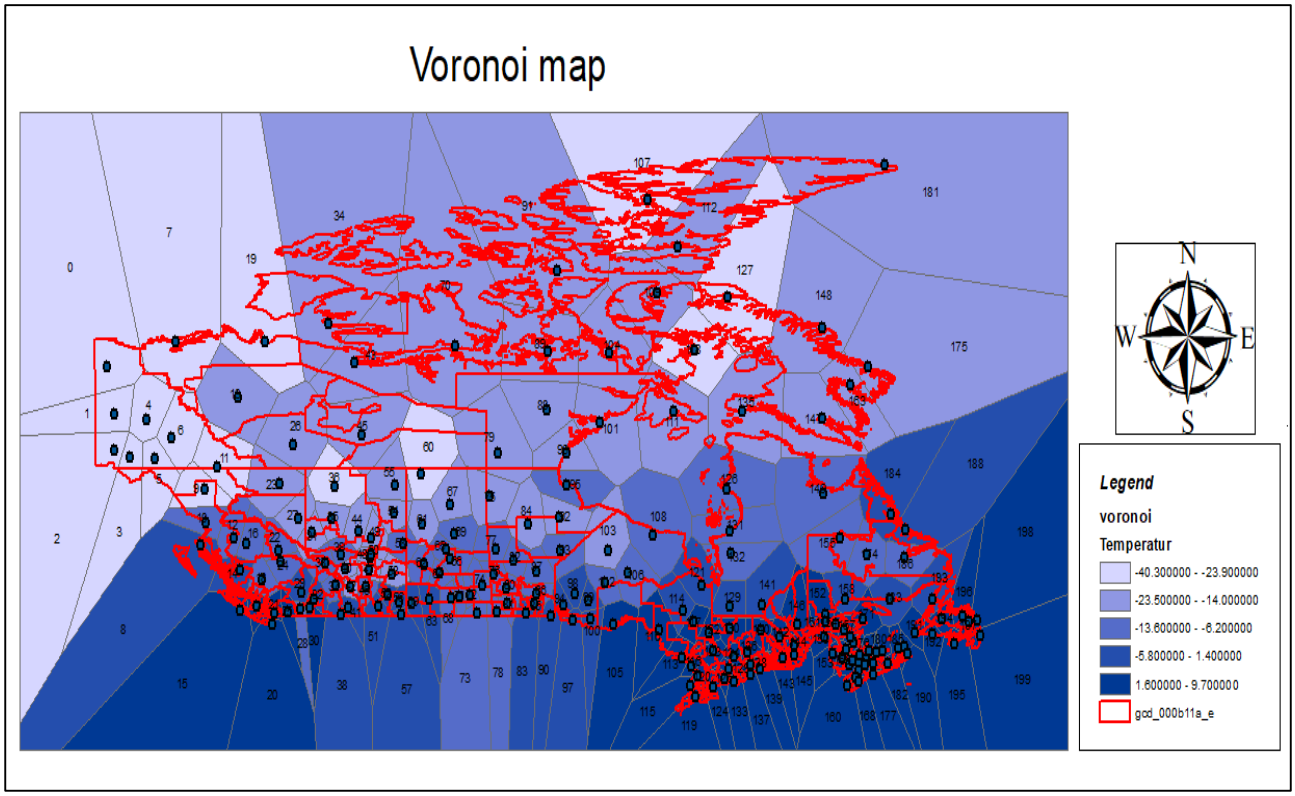
"Labels" sub-menu is used for labeling the map. By clicking check-box button the labels will be shown on the map. Other settings are useful for better interface.

Step 5: Overlay Graduated colors Voronoi map with Canada provinces.





Click on “Add data”, then browse “gcd_000b11a_e” shapefile, and press Add button. Then, Right click on “gcd_000b11a_e.shp”, click on “Properties” and Seek “Symbology” tab. On the left part (show:) map type must be chosen (“Single symbol:). Then, by clicking the symbol second menu will appear. In the left side (Fill Color:) the polygon color can be declare (“No Color”), and the outline color must be a color that can be separated from graduated color map (“Red”). Final Result:



III

Multivariate -Bivariate

3.1. Stationary

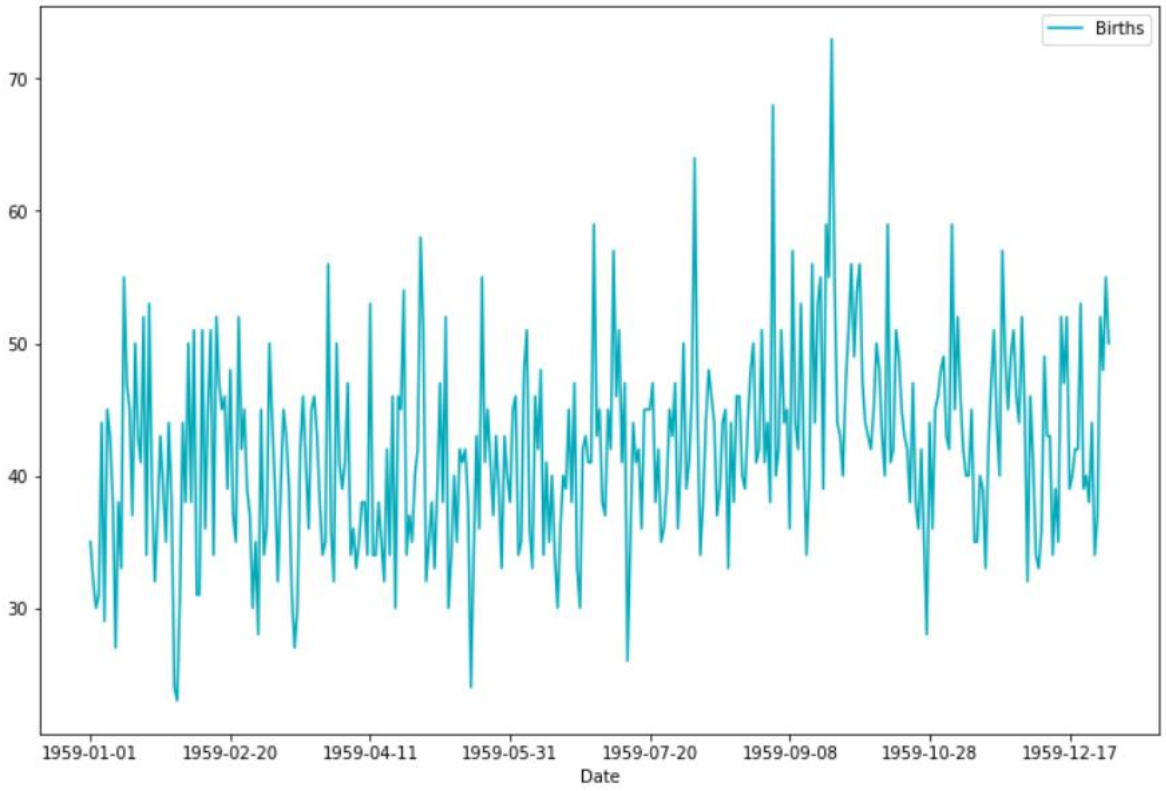
Stationary features contain data about a single place over time, such as a sensor, and change/growth features represent a feature that changes in size, such as a forest fire. Time series are stationary if they do not have trend or seasonal effects. Summary statistics calculated on the time series are consistent over time, like the mean or the variance of the observations. When a time series is stationary, it can be easier to model. Statistical modelling methods assume or require the time series to be stationary to be effective. The notion of stationarity comes from the theoretical study of time series and it is a useful abstraction when forecasting.

There are some finer-grained notions of stationarity that may come encounter if dive deeper into this topic, they are:

- **Stationary Process**: A process that generates a stationary series of observations.
- **Stationary Model**: A model that describes a stationary series of observations.
- **Trend Stationary**: A time series that does not exhibit a trend.
- **Seasonal Stationary**: A time series that does not exhibit seasonality.
- **Strictly Stationary**: A mathematical definition of a stationary process, specifically that the joint distribution of observations is invariant to time shift.

In here, a time series "Daily Female Births dataset" data will be explored. This dataset encompasses monthly number of female children delivery spreads across 1959's. This data set can be downloaded from here. Used libraries in this part include "Pandas" and python's "Matplotlib", first impression of data set can be seen in next figure.

```
import pandas as pd
from matplotlib import pyplot
series = pd.read_csv('daily-total-female-births.csv', header=0, index_col=0)
series.plot(figsize = (12,8), color = '#00aabb')
```



There are many methods to check whether a time series (direct observations, residuals, otherwise) is stationary or non-stationary.

1. **Look at Plots**: Reviewing a time series plot of data and visually checking if there are any obvious trends or seasonality.
2. **Summary Statistics**: Evaluation of the summary statistics for data for seasons or random partitions and check for obvious or significant differences.
3. **Statistical Tests**: Using statistical tests to check if the expectations of stationarity are met or have been violated.

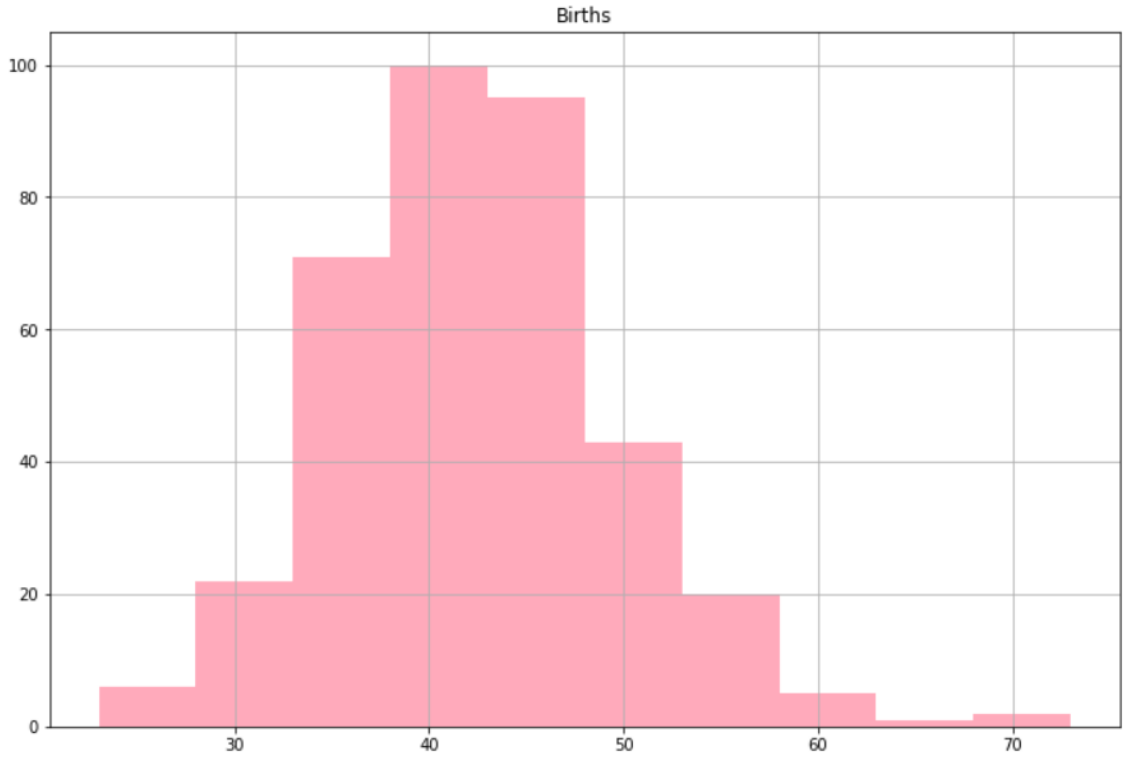
Above, Daily Female Births has been introduced respectively with plot showing an obvious lack and presence of trend and seasonality components. Next step is to calculate and review summary statistics on time series dataset for checking to see if it is stationary.

3.1.1. Summary Statistics

By splitting time series into two or more partitions and compare the mean and variance of each group it is conceivable that if they differ and the difference is statistically significant, the time series is likely non-stationary. Because by looking at the mean and variance, assuming that the data conforms to a

Gaussian distribution is sign of stationary. Running the example plots a histogram of values from the time series, it's clear that the bell curve-like shape of the Gaussian distribution, perhaps with a longer right tail.

```
series.hist(figsize = (12,8), color = "#ffaabb");
```



Next, split the time series into two contiguous sequences, then calculate the mean and variance of each group of numbers and compare the values. Running this example shows that the mean and variance values are different, but in the same ball-park.

```
X = series.values
split = round(len(X) / 2)
X1, X2 = X[0:split], X[split:]
mean1, mean2 = X1.mean(), X2.mean()
var1, var2 = X1.var(), X2.var()
print('mean1=%f, mean2=%f' % (mean1, mean2))
print('\nvariance1=%f, variance2=%f' % (var1, var2))
```

```
mean1=39.763736, mean2=44.185792

variance1=49.213410, variance2=48.708651
```

3.1.2. Augmented Dickey-Fuller test

Statistical tests make strong assumptions about datasets. They can only be used to inform the degree to which a null hypothesis can be rejected or fail to be reject. The result must be interpreted for a given problem to be meaningful. Nevertheless, they can provide a quick check and confirmatory evidence that time series is stationary or non-stationary.

The Augmented Dickey-Fuller test is a type of statistical test called a unit root test. The intuition behind a unit root test is that it determines how strongly a time series is defined by a trend. There are a number of unit root tests and the Augmented Dickey-Fuller may be one of the more widely used. It uses an autoregressive model and optimizes an information criterion across multiple different lag values.

The null hypothesis of the test is that the time series can be represented by a unit root, that it is not stationary or has some time-dependent structure. The alternate hypothesis or rejecting the null hypothesis is that the time series is stationary.

- **Null Hypothesis (H0)**: If failed to be rejected, it suggests the time series has a unit root, meaning it is non-stationary. It has some time dependent structure.
- **Alternate Hypothesis (H1)**: The null hypothesis is rejected; it suggests the time series does not have a unit root, meaning it is stationary. It does not have time-dependent structure.

Interpreting that this result using the p-value from the test. A p-value below a threshold (such as 5% or 1%) suggests reject the null hypothesis (stationary), otherwise a p-value above the threshold suggests we fail to reject the null hypothesis (non-stationary).

- **p-value > 0.05**: Fail to reject the null hypothesis (H0), the data has a unit root and is non-stationary.
- **p-value <= 0.05**: Reject the null hypothesis (H0), the data does not have a unit root and is stationary.

Below is an example of calculating the Augmented Dickey-Fuller test on the Daily Female Births dataset. The “statsmodels” library provides the “adfuller” function that implements the test. “statsmodel” library also used in QQ plot part to generate normal QQ plot.

```
print("Results of Dickey-Fuller Test:")
dftest = adfuller(series['Births'], autolag="AIC")
dfoutput = pd.Series(dftest[0:4], index=["Test Statistic",
                                         "p-value", "#Lags Used",
                                         "Number of data Used",],)
for key, value in dftest[4].items():
  dfoutput["Critical Value (%s)" % key] = value
print(dfoutput)
```

```
Results of Dickey-Fuller Test:
Test Statistic                  -4.808291
p-value                          0.000052
#Lags Used                       6.000000
Number of Observations Used    358.000000
Critical Value (1%)             -3.448749
Critical Value (5%)             -2.869647
Critical Value (10%)            -2.571089
dtype: float64
```

As shown in picture, running the example prints the test statistic value of -4. The more negative this statistic, the more likely to reject the null hypothesis or in other words, dataset is stationary dataset. As part of the output, can see that statistic value of -4 is less than the value of -3.449 at 1%. This suggests that the null hypothesis with a significance level of less than 1% could be rejected, i.e. a low probability that the result is a statistical fluke. Rejecting the null hypothesis means that the process has no unit root, and in turn that the time series is stationary or does not have time-dependent structure.

There is also KPSS that is another test for checking the stationarity of a time series. The null and alternate hypothesis for the KPSS test are opposite that of the ADF test.

- **Null Hypothesis (H0)**: The process is trend stationary.
- **Alternate Hypothesis (H1)**: The series has a unit root (series is not stationary).

Another function is created on “statsmodel” library to carry out the KPSS test on a time series data, this function could also be applied to dataset in evaluation of stationary or non-stationary.

3.2. Probabilistic modelling

Probabilistic modelling is a statistical technique used to take into account the impact of random events or actions in predicting the potential occurrence of future outcomes. Complex questions in spatial science are not usually answered by a single model simulation generated by one set of idealized input parameters. Rather, the best solution may be a collection or ensemble of model results that characterize the range of environmental conditions observed in the real world. The probabilistic modelling being used to understand the range and likelihood of outcomes and to place quantitative bounds on uncertainty in results.

Analysis begins by generating probability distributions of model input parameters based on the best available spatial and temporal data for real world observations. Depending on the number of uncertain inputs, a sampling technique, such as Monte Carlo or Latin Hypercube Sampling, being selected to create realizations of model input parameters.

3.2.1. Random Variable

A random variable, usually written X, is a variable whose possible values are numerical outcomes of a random phenomenon. There are two types of random variables, discrete and continuous.

A discrete random variable is one which may take on only a countable number of distinct values such as 0,1,2,3,4, etc. Discrete random variables are usually (but not necessarily) counts. If a random variable can take only a finite number of distinct values, then it must be discrete. Examples of discrete random variables include the number of children in a family, the Friday night attendance at a cinema, the number of patients in a doctor's surgery, the number of defective light bulbs in a box of ten. The probability distribution of a discrete random variable is a list of probabilities associated with each of its possible values. It is also sometimes called the probability function or the probability mass function.

A continuous random variable is one which takes an infinite number of possible values. Continuous random variables are usually measurements.

Examples include height, weight, the amount of sugar in an orange, the time required to run a mile. A continuous random variable is not defined at specific values. Instead, it is defined over an interval of values, and is represented by the area under a curve, in advanced mathematics, this is known as an integral. The probability of observing any single value is equal to 0, since the number of values which may be assumed by the random variable is infinite.

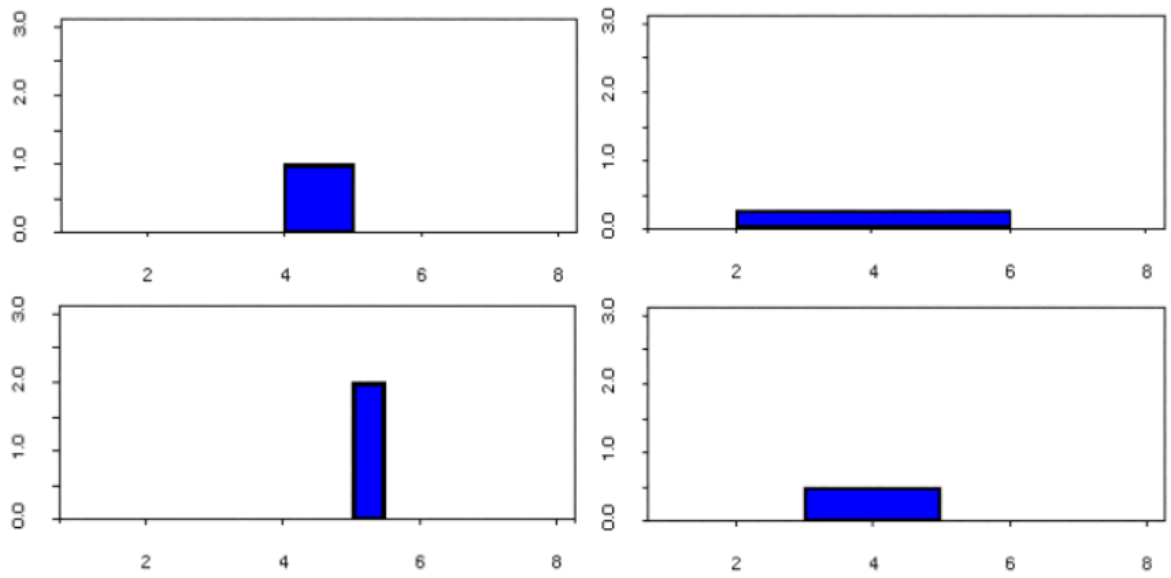
Suppose a random variable X may take all values over an interval of real numbers. Then the probability that X is in the set of outcomes A, P(A), is defined to be the area above A and under a curve. The curve, which represents a function p(x), must satisfy the following:

- The curve has no negative values (p(x) > 0 for all x)
- The total area under the curve is equal to 1.

A curve meeting these requirements is known as a density curve. A random number generator acting over an interval of numbers (a, b) has a continuous distribution. Since any interval of numbers of equal width has an equal probability of being observed, the curve describing the distribution is a rectangle, with constant height across the interval and 0 height elsewhere. Since the area under the curve must be equal to 1, the length of the interval determines the height of the curve.

Another type of continuous density curve is the normal distribution. The area under the curve is not easy to calculate for a normal random variable *X* with mean μ and standard deviation ð. However, tables (and computer functions) are available for the standard random variable *Z*, which is computed from *X* by subtracting μ and dividing by ð. All of the rules of probability apply to the normal distribution.

The following graphs plot the density curves for random number generators over the intervals (4,5) (top left), (2,6) (top right), (5,5.5) (lower left), and (3,5) (lower right). The distributions corresponding to these curves are known as uniform distributions.



Random variable is being used in many ways of spatial interpolation, one of the most used is Kriging. Ordinary kriging one is one of kriging method's that exploit RV. In next part this Stochastic interpolation method will briefly be introduced.

3.2.2. Kriging

Kriging is a form of probabilistic and local interpolation. The value that has to be assigned to one of the points of the field does not depend on all the values available, but only on those observed in the closest points. These methods are commonly used in the geosciences. Probabilistic interpolations by Kriging involve two phases: a variography and a process of Kriging. The variography indicates the distance after which it becomes useless to integrate data in order to determine the value we wish to find in a point, and the analysis of the variograms reveals the presence or the force of an anisotropic effect. This information, together with other constraints, allows us to choose the Kriging neighborhood, i.e. the domain of the field that contains the site to be estimated and the data required to do so. A Kriging method can be deduced.

Kriging can be carried out in a stationary or non-stationary context. Simple Kriging is chosen if the mean is known, which happens only exceptionally. Instead, the opposite is often true: the average is unknown and has to be found. In that case, ordinary Kriging is chosen. Constant progress is being made in

geostatic research. For example, Kriging could be performed on spatial mesh structures by units and not only by points. Disjunctive Kriging is advisable for non-linear approaches. Implementing probabilistic approaches is therefore more time consuming and complex; however, on the contrary, they can adapt more fittingly to the constraints imposed by terrain and the data collected.

3.2.2.1. Kriging in Python

As mentioned above, ordinary kriging is one of the methods could be used for interpolation. In ordinary kriging the unknown value is interpreted as a random variable located in , as well as the values of neighbours samples The estimator Z is also interpreted as a random variable located in , a result of the linear combination of variables.

In order to deduce the kriging system for the assumptions of the model, the following error committed while estimating is declared:

$$\epsilon(x_0) = \hat{Z}(x_0) - Z(x_0) = [W^T \quad -1] \cdot [Z(x_1) \quad \cdots \quad Z(x_N) \quad Z(x_0)]^T = \sum_{i=1}^{N} w_i(x_0) \times Z(x_i) - Z(x_0).$$

The two quality criteria referred to previously can now be expressed in terms of the mean and variance of the new random variable The stationary checking method also could apply here to check the trueness of random variable.

Ordinary kriging will estimate an appropriate mean of the field, based on the given observations/conditions and the covariance model used. The resulting system of equations for W is given by:

$$\begin{pmatrix} W \\ \mu \end{pmatrix} = \begin{pmatrix} c(x_1, x_1) & \cdots & c(x_1, x_n) & 1 \\ \vdots & \ddots & \vdots & \vdots \\ c(x_n, x_1) & \cdots & c(x_n, x_n) & 1 \\ 1 & \cdots & 1 & 0 \end{pmatrix}^{-1} \begin{pmatrix} c(x_1, x_0) \\ \vdots \\ c(x_n, x_0) \\ 1 \end{pmatrix}$$

Thereby c (xi, xj) is the covariance of the given observations and μ is a Lagrange multiplier to minimize the kriging error and estimate the mean. To implement this method in python “GSTools” library will be utilized. There is other libraries like “HPGL” which include many of kriging methods in

themselves. Used library in here contains useful function for either stochastic or deterministic models.

Use data for this part is “Canada Weather condition” which has been presented in pervious chapter. In case of encountering error whilst installing “GSTools” try to upgrade “numpy” library by “!pip install –U numpy” command. After calling for library and loading shape file, the coordinate of each station should be delineate and the data attribute for interpolation should be selected.

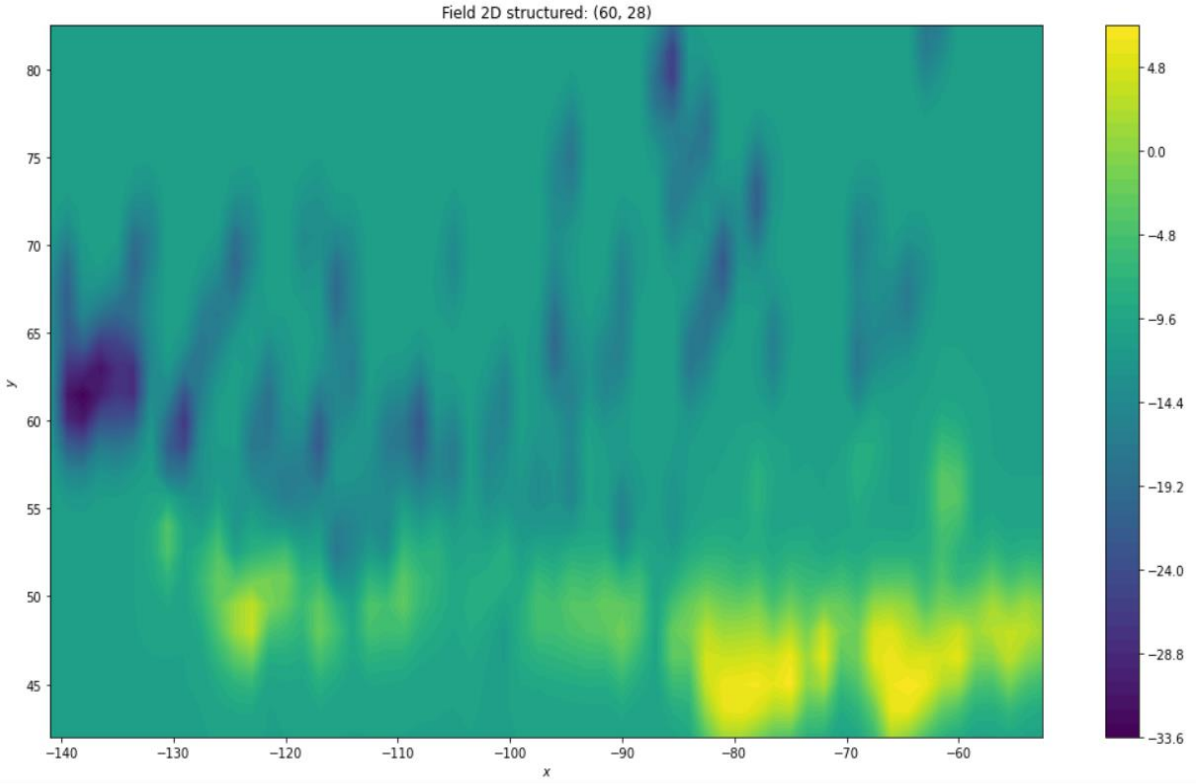
```
import gstools as gs
import numpy as np
import geopandas as gpd
shpData = gpd.read_file("Environment_Canada_Weather_Conditions.shp").dropna()

geomX = [i for i in shpData['geometry'].x]
geomY = [j for j in shpData['geometry'].y]
windSpeed = [i for i in shpData['Temperatur']]
```

Next step is to define Gaussian model to be used as variogram model in kriging interpolation. A grid to project interpolated data is necessary too, this grid should be compatible with data bonding box, step in defining this grid is cell size. Defining these two parameters, variogram model and grid, is project and data based.

```
model = gs.Gaussian(dim=2, len_scale=1, anis=2.92,
                    angles=-0.1, var=0.5, nugget=0.1)

gridx = np.arange(np.floor(min(geomX)) - 1, np.ceil(max(geomX)) + 1, 1.5)
gridy = np.arange(np.floor(min(geomY)) - 1, np.ceil(max(geomY)) + 1, 1.5)
```

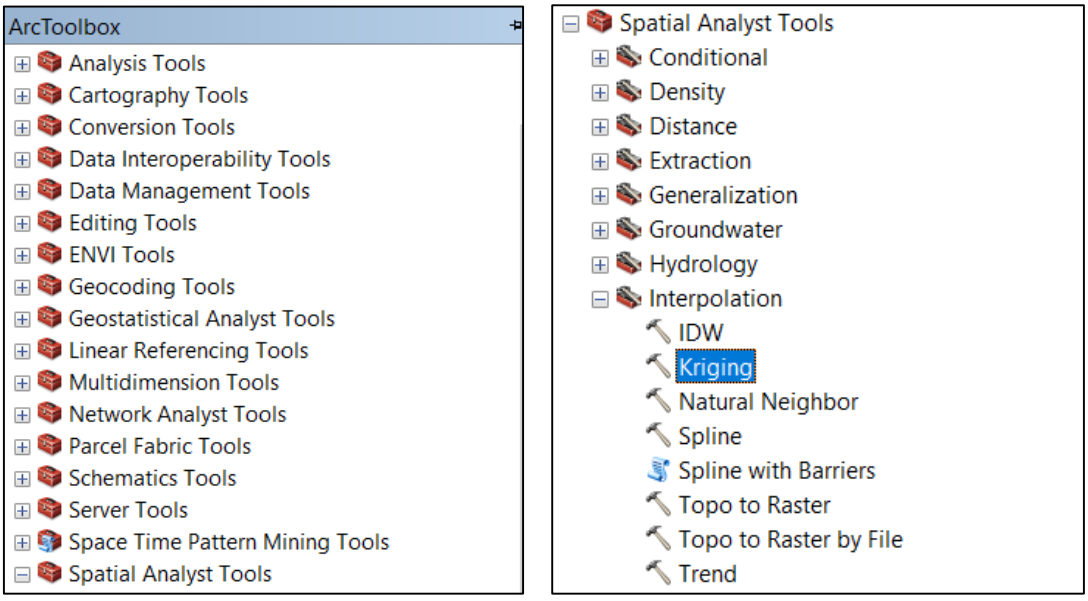
At the end of interpolation procedure in python, from “gstools” library, OrdinaryKriging function which utilize RV will be called. Coordinates, value attribute, model and “exact=True” are input arguments of this function. After applying this method on data, result will be projected on defined grid and plotted. The final result is shown in below.

```
OK = gs.krige.Ordinary(model, [geomX, geomY], windSpeed, exact=True)
OK.structured([gridx, gridy])
fig1, ax1 = plt.subplots(1, sharex=True, figsize=(18,10))
ax = OK.plot(fig = fig1, ax = ax1)
```



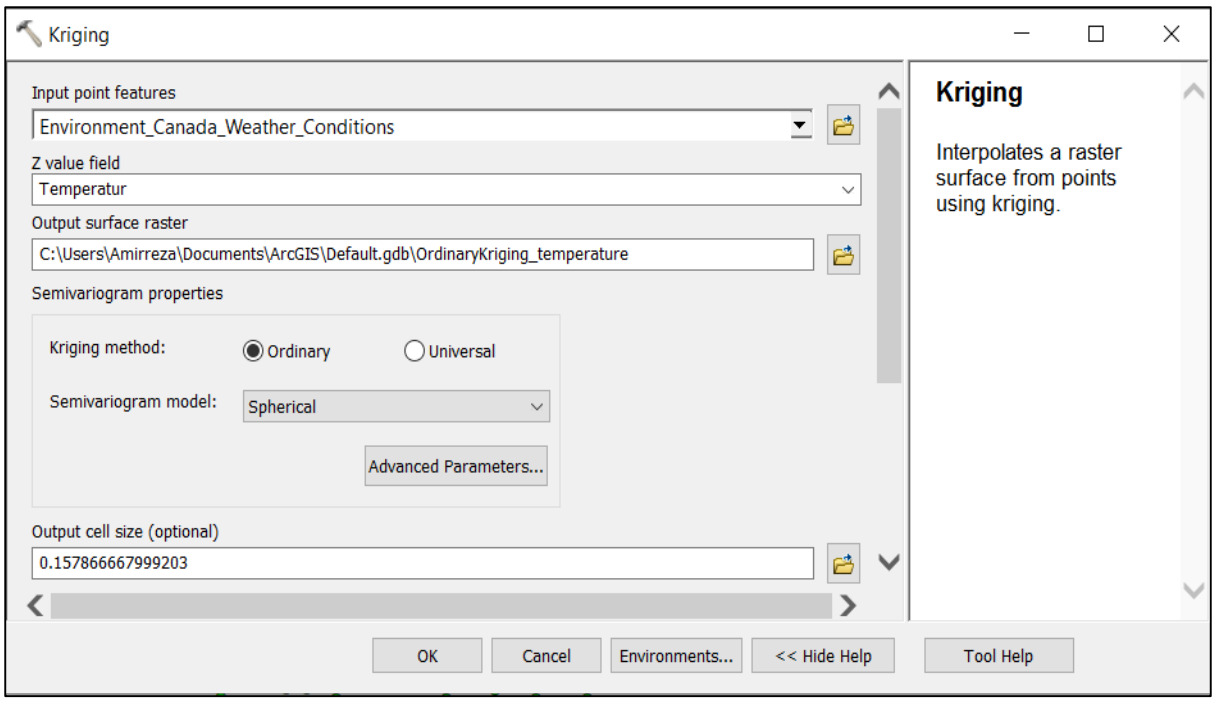
3.2.2.2. Kriging in ArcGIS

To create a raster that estimates whole region temperatures by a probabilistic model, Kriging can be used. first step is to add data, which in here, same as the Python “Canada weather condition” will be used.



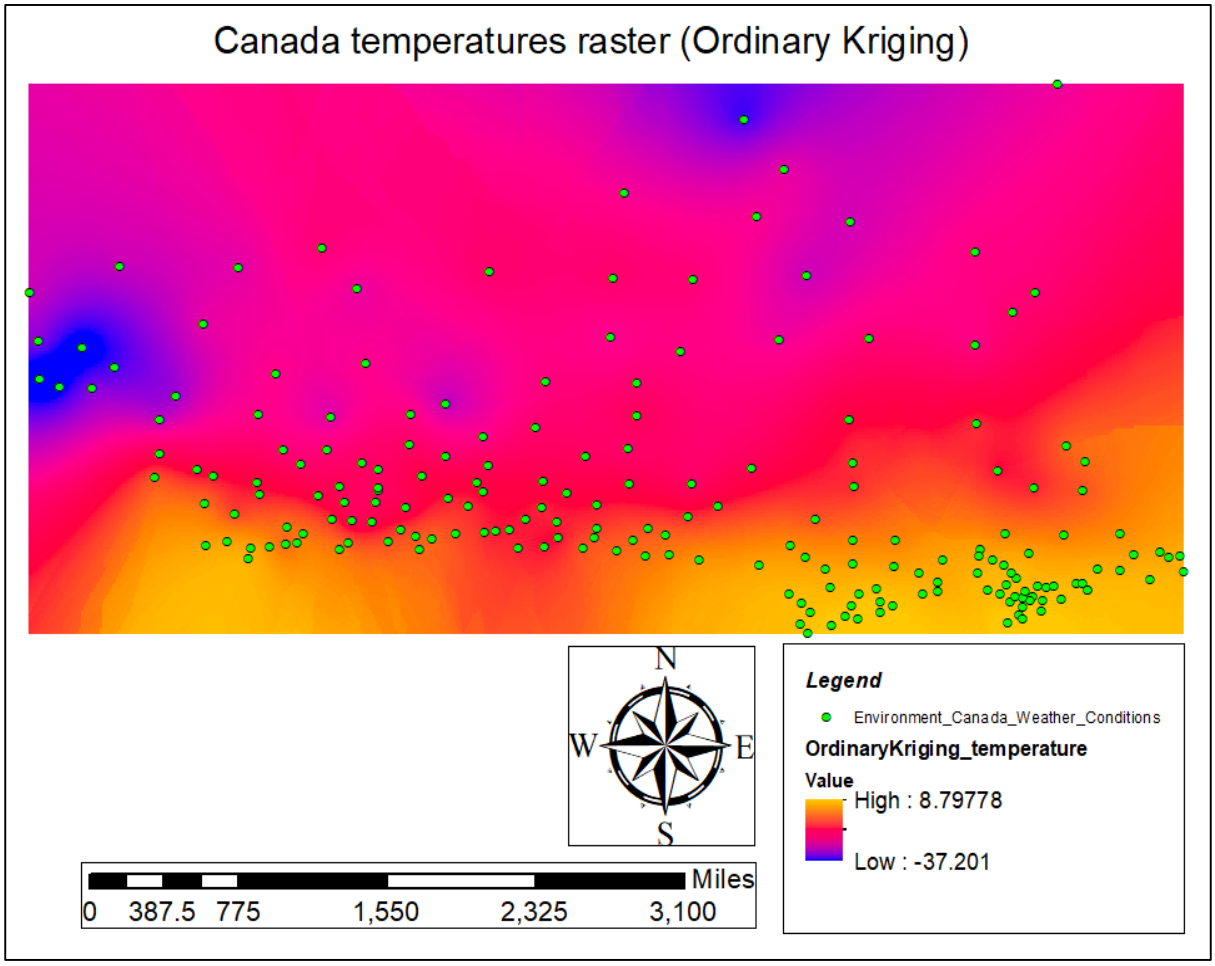
First step is to open "ArcToolbox" and find "Spatial Analyst Tools" toolbox. Then, click on "interpolation" menu and open "Kriging" analysis.

Step 3: Create an Ordinary Kriging raster.

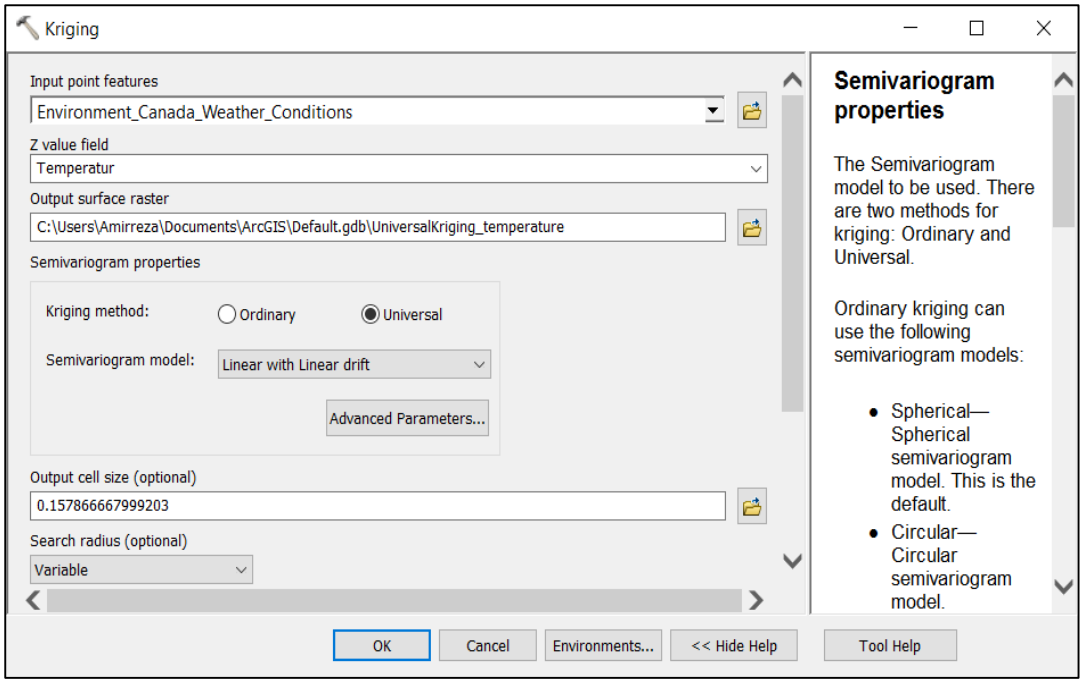


Same as before examples (3.1.2 and 3.1.4), Input point features, Z value field, Output cell size and search radius must be added. The most important part in Kriging analysis is Semivariogram properties. There are two kriging methods: ordinary and universal. Ordinary kriging is the most general and widely used of the kriging methods. It assumes the constant mean is unknown.

This is a reasonable assumption unless there is a scientific reason to reject it. Universal kriging assumes that there is an overriding trend in the data and it can be modeled by a deterministic function, a polynomial. This polynomial is subtracted from the original measured points, and the autocorrelation is modeled from the random errors. *fina result:*

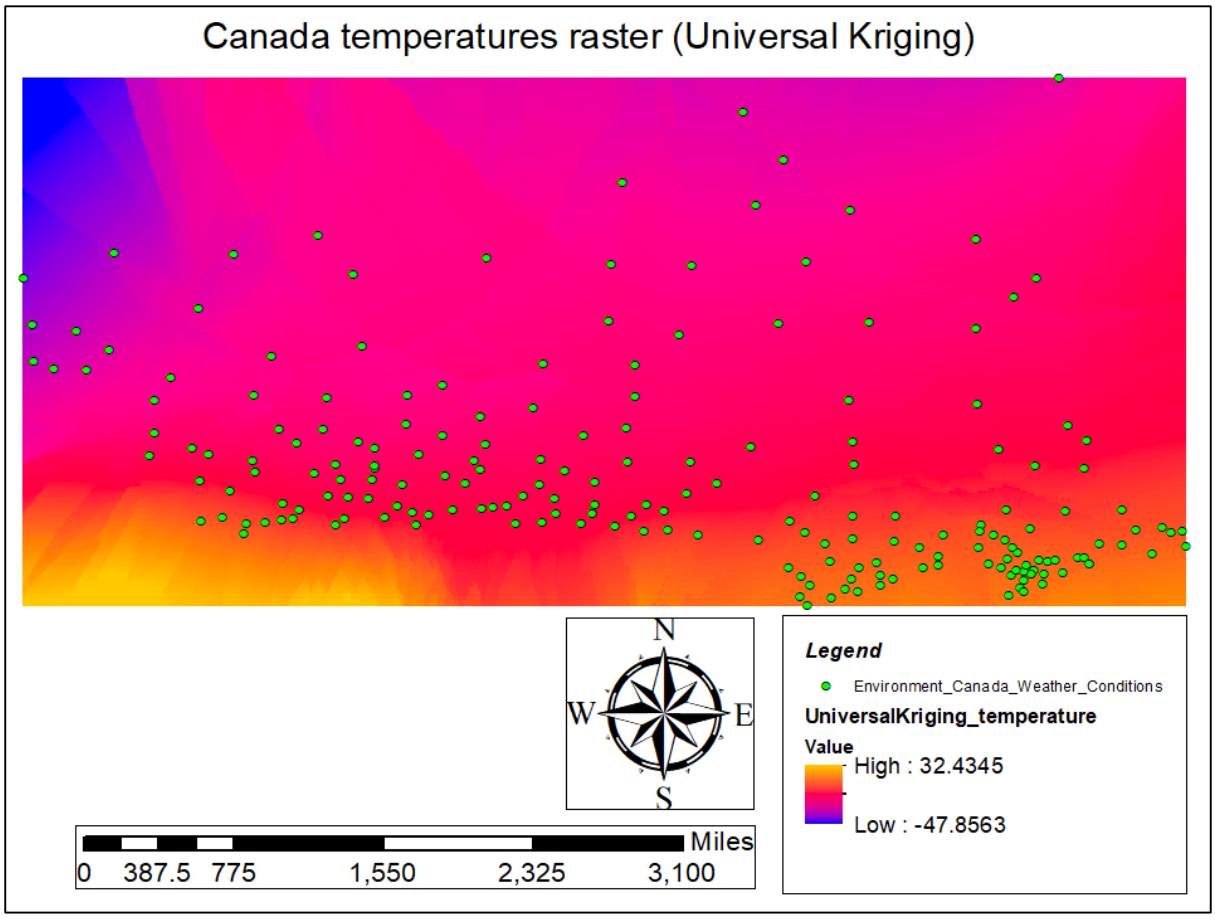


Step 3: Create a Universal Kriging raster.



To evaluate the difference between ordinary and universal Kriging, both methods will be used.

Result:



3.3. Deterministic Model

Deterministic interpolation techniques create surfaces from measured points, based on either the extent of similarity (inverse distance weighted) or the degree of smoothing (radial basis functions). Geo-Stochastic interpolation models which introduced in pervious part (RV - kriging) utilize the statistical properties of the measured points. Geo-statistical techniques quantify the spatial autocorrelation among measured points and account for the spatial configuration of the sample points around the prediction location.

Deterministic interpolation techniques can be divided into two groups, global and local. Global techniques calculate predictions using the entire dataset. Local techniques calculate predictions from the measured points within neighbourhoods, which are smaller spatial areas within the larger study area. Global techniques like Trend surface which are inexact are not mentioned in here as result of similarity. Local techniques include Thiessen polygon, Density estimation, Inverse distance weighted(IDW), spline.

A deterministic interpolation can either force the resulting surface to pass through the data values or not. An interpolation technique that predicts a value that is identical to the measured value at a sampled location is known as an exact interpolator. An inexact interpolator predicts a value that is different from the measured value. The latter can be used to avoid sharp peaks or troughs in the output surface. Inverse distance weighted and radial basis functions are exact interpolators, while global polynomial, local polynomial, kernel interpolation with barriers, and diffusion interpolation with barriers are inexact.

3.3.1. Deterministic Model in Python

There are some well-known libraries to use for this purpose, like “scipy” or “GDAL” that could be used. In here one of implemented models will be developed from scratch.

IDW from scratch

To implement Invers Distance Weighted method in python “numpy” library and “Canada Weather condition” will be exploited also “Matplotlib” will be

used for presenting the outcomes. First step in IDW, is calculation of distance, for this purpose a function will be settled. Known points coordinate list and unknown points coordinate list are this function input arguments. First, by use of “numpy.vstack” arrays in sequence vertically will be stacked. Next step is to compute the m by p matrix of distances, the “numpy.subtract.outer” calls make two such matrices of scalar differences along the two axes, then “numpy.hypot” calls turns those into a same-shape matrix of scalar Euclidean distances and return result.

```
def distance_matrix(x0, y0, x1, y1):
    obs = np.vstack((x0, y0)).T
    interp = np.vstack((x1, y1)).T
    # Make a distance matrix between pairwise observations
    d0 = np.subtract.outer(obs[:,0], interp[:,0])
    d1 = np.subtract.outer(obs[:,1], interp[:,1])
    return np.hypot(d0, d1)
```

Next function to define is IDW, known points coordinates array, known points value, unknown points coordinates as well as known in separate arrays, and power of IDW, which is optional, are function input arguments. First step of IDW function is to call for Distance calculator. Weight for each calculated distance between known and unknown points, attainable by inverse of distance power. Last do of function is to multiply the weights for each interpolated point by all known values that need to be interpolated.

```
def simple_idw(x, y, z, xi, yi, p = 2):
    dist = distance_matrix(x,y, xi,yi)
    # In IDW, weights are 1 / distance
    weights = 1.0 / dist ** p
    # Make weights sum to one
    weights /= weights.sum(axis=0)
    # Multiply the weights for each interpolated point by all observed Z-values
    zi = np.dot(weights.T, z)
    return zi
```

Next function to define is plotting function. This part is only for convince of demonstration purposes. Input arguments for this function include knows points and value attribute in separate 1D also grid of IDW result points that being

interpolated as 2D array is necessary. This function scatter points on top of interpolated models of data.

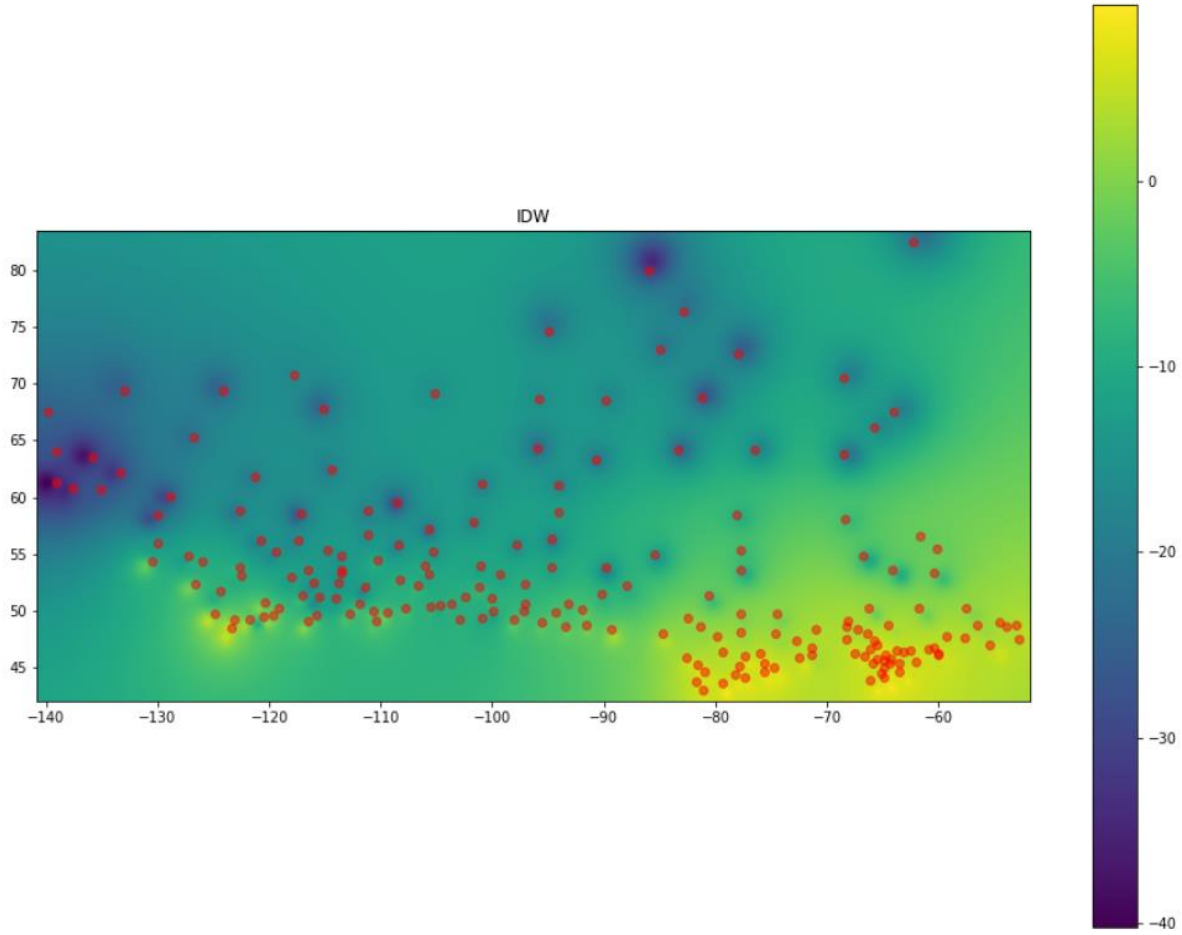
```
def plot(x,y,z,grid):
    plt.figure( figsize = (16,12))
    plt.scatter(x,y,color = 'r', alpha= 0.5)
    plt.imshow(grid, extent=(x.min() - 1, x.max() + 1, y.min() - 1, y.max() + 1))
    plt.colorbar()
```

After defining all functions, dataset with “Geopnadas” library will be added. From dataset, geometry related to each point and “Temperature” value will be exploited as an array.

```
import geopandas as gpd
shpData = gpd.read_file("Environment_Canada_Weather_Conditions.shp").dropna()

x = np.array([i for i in shpData['geometry'].x])
y = np.array([j for j in shpData['geometry'].y])
z = np.array([i for i in shpData['Temperatur']])
```

Afterward, number of points for interpolation and a grid will be defined to project the result value on each point of this grid. Due to functions 1D array inputs, all unknown values of grid coordinates will be flattened. Next step is to call for IDW function. Result of IDW function is 1D array too, for plotting grid it should be reshaped. At the end plot function will be called and result will be displayed as shown in below.

```
nx, ny = 400, 400
xi = np.linspace(x.min(), x.max(), nx)
yi = np.linspace(y.max(), y.min(), ny)
xi, yi = np.meshgrid(xi, yi)
xi, yi = xi.flatten(), yi.flatten()

grid1 = simple_idw(x,y,z,xi,yi)
grid1 = grid1.reshape((ny, nx))

plot(x,y,z,grid1)
plt.title('IDW')
plt.show()
```



Spline in Python

To do this, a module of “SciPy” library utilized. As well as pervious part, used dataset is “Canada Weather Condition”.

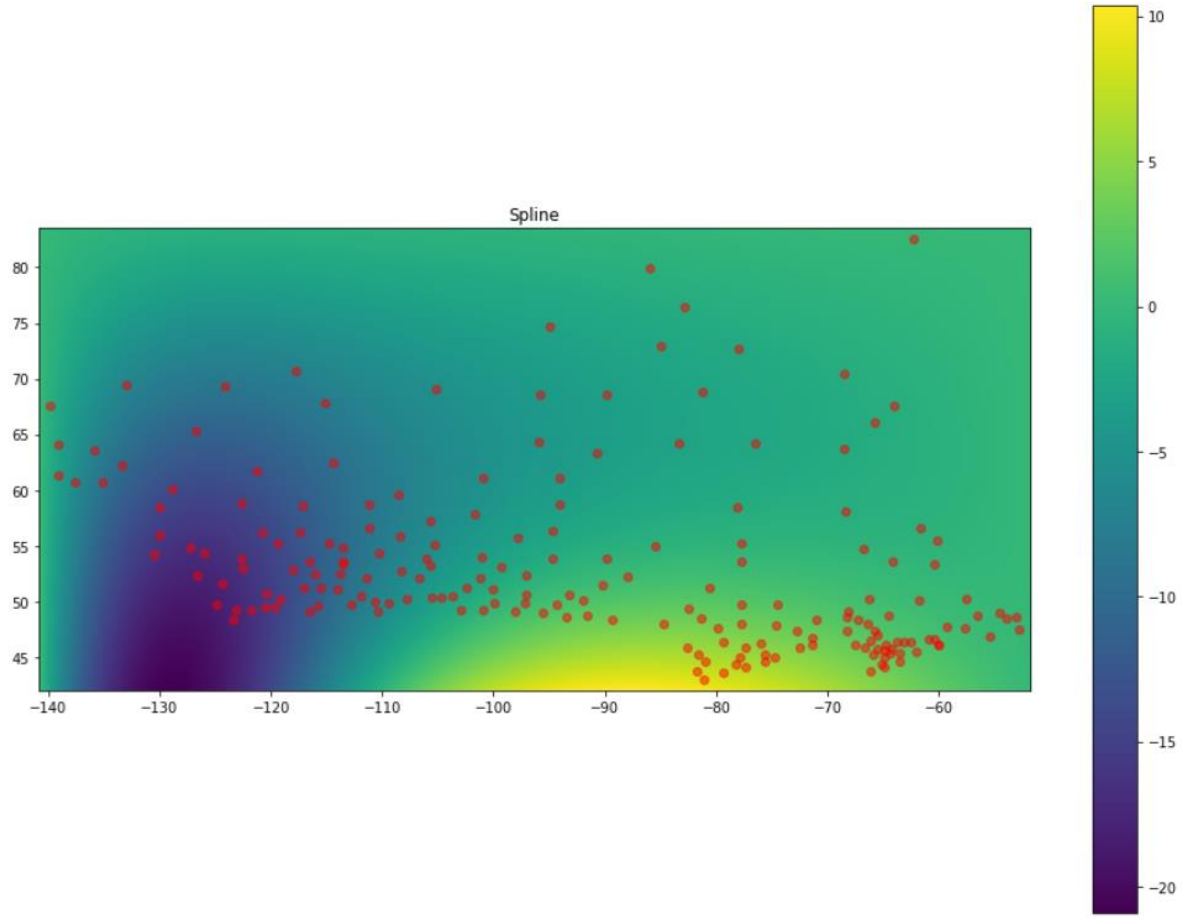
```
import geopandas as gpd
shpData = gpd.read_file("Environment_Canada_Weather_Conditions.shp").dropna()

x = np.array([i for i in shpData['geometry'].x])
y = np.array([j for j in shpData['geometry'].y])
z = np.array([i for i in shpData['Temperatur']])
```

Next step is to add used libraries in pervious part and from “SciPy” library “Interpolate” module will be added. “SmoothBivariateSpline” from this module will be applied to dataset. Parameter of Spline is for degrees of the bivariate spline and a threshold for determining the effective rank of an over-determined linear system of equations. “eps” should have a value within 0 and 1.

```
import numpy as np
from scipy import interpolate
import matplotlib.pyplot as plt

spl = interpolate.SmoothBivariateSpline(x, y, z, kx = 3, ky = 3 , eps=0.99)
```

As well as IDW, a grid should be defined in here too. After defining grid like what discussed in IDW, “spl” will be applied to coordinate of defined grid to interpolate unknown values. Note that, input value for “spl” should be single coordinate. At the end by use of defined plot function in IDW, result will be plotted.

```
nx , ny = 200, 200
xi = np.linspace(x.min(), x.max(), nx)
yi = np.linspace(y.max(), y.min(), ny)
xi, yi = np.meshgrid(xi, yi)
xi, yi = xi.flatten(), yi.flatten()

zi = np.array([float(spl(xi[i], yi[i])) for i in range(len(xi))])

grid1 = zi.reshape((nx, ny))

def plot(x,y,z,grid):
    plt.figure( figsize = (16,12))
    plt.scatter(x,y,color = 'r', alpha= 0.5)
    plt.imshow(grid, extent=(x.min() - 1, x.max() + 1, y.min() - 1, y.max() + 1))
    plt.colorbar()

plot(x,y,z,grid1)
plt.title('Spline')
plt.show()
```

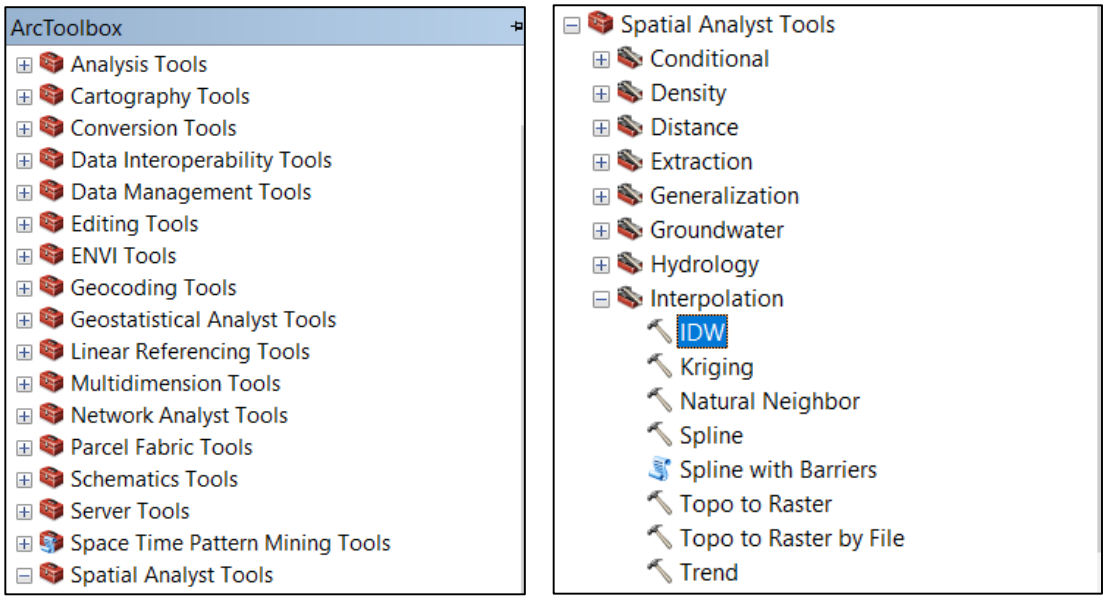


3.3.2. Deterministic Model in ArcGIS

IDW in ArcGIS

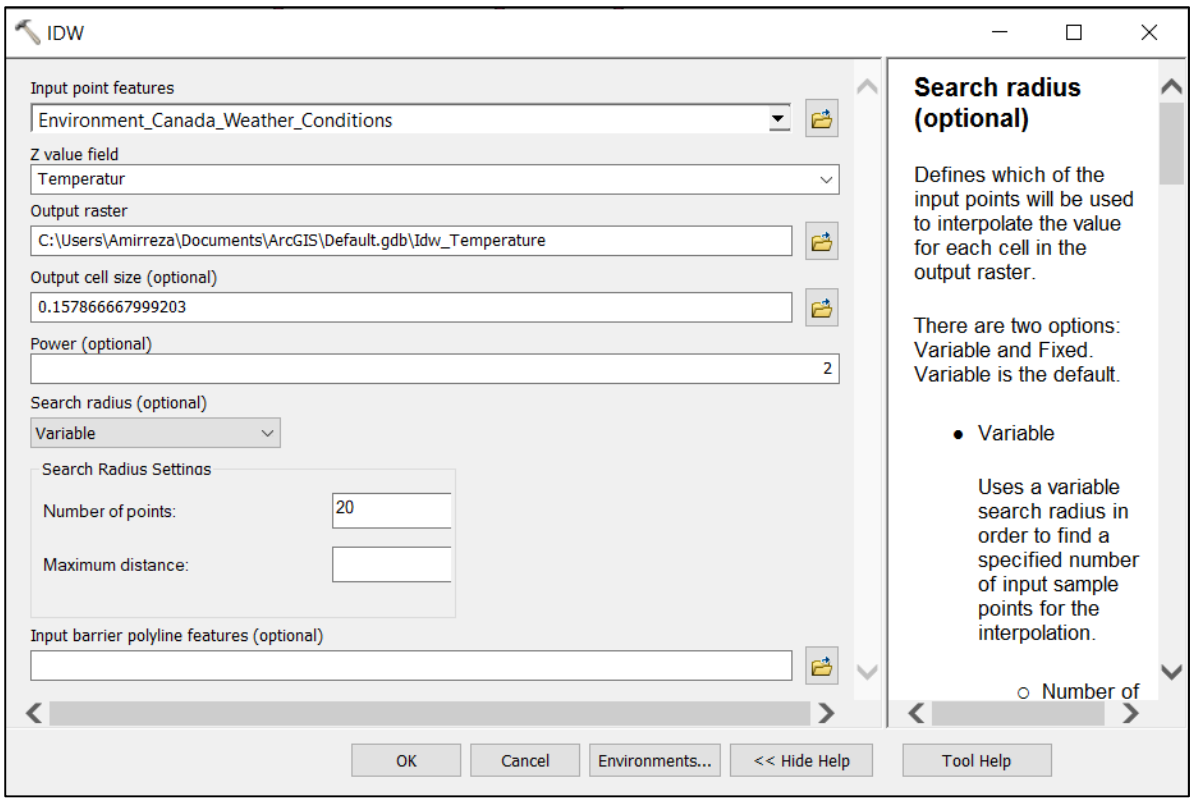
To create a raster for Canada temperatures that estimates the whole region temperature (not only the stations), we can use IDW interpolation method. As all example in ArcMap, first step is to add data, which in here, same as the Python "Canada weather condition" will be used.

Step 2: open IDW menu.



At first, open "ArcToolbox" and find "Spatial Analyst Tools" toolbox. Then, click on "interpolation" menu and open "IDW" analysis.

Step 3: Create an IDW raster.

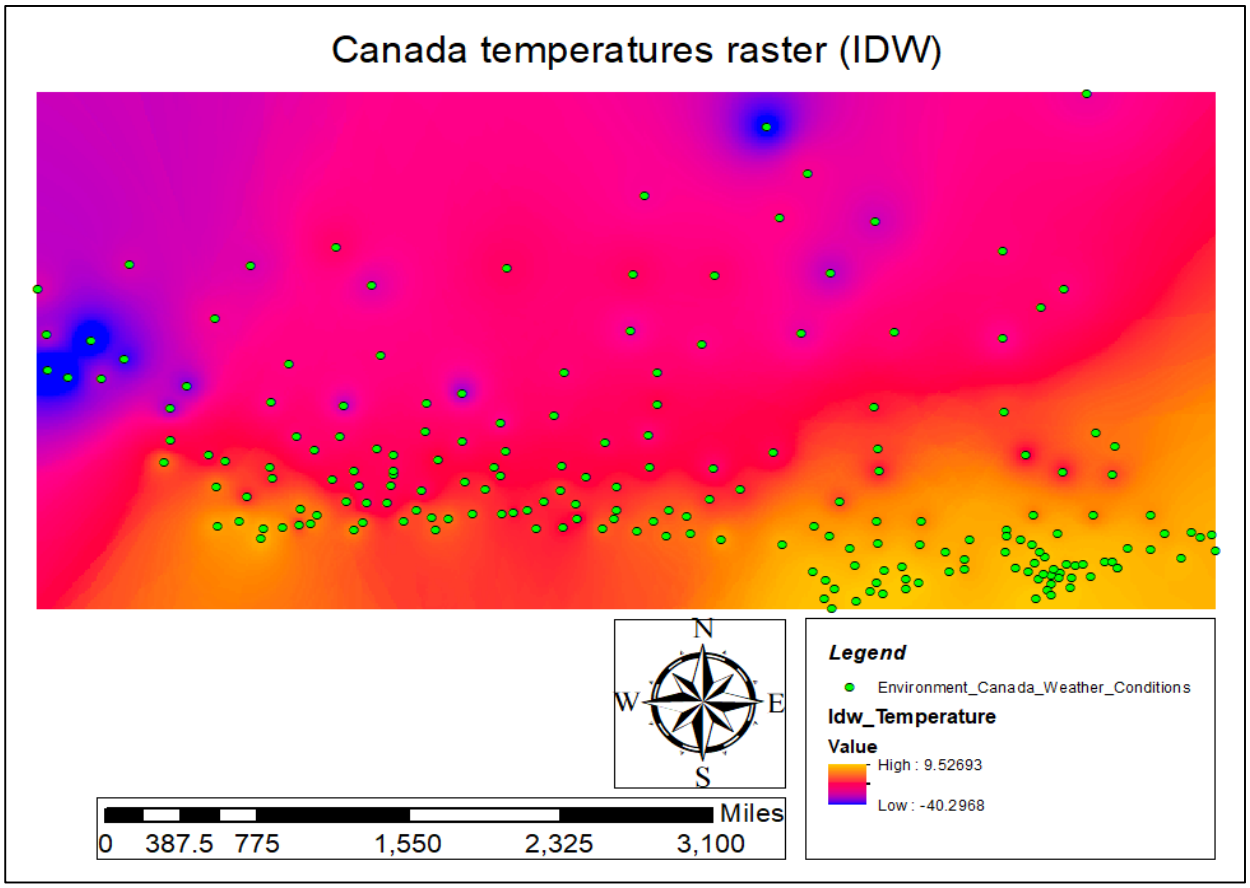


First of all, in Input point features the weather stations shapefile ("Canada weather condition") must be added. Then, the field that holds a magnitude value for each point called Z value field must be chosen ("Temperatur"). In Output

cell size, the pixel size of the output raster will be created can be defined. As you know, in IDW formula the distance is denominator and you can define the denominator power.

There are two types of radius classes: Variable and Fixed. A Variable search radius is used to find a specified number of input sample points for the interpolation. While, the Fixed type uses a specified fixed distance within which all input points will be used for the interpolation. Number of points that is used in interpolation process is needed ("20"), also Maximum distance can be defined (the distance unit is same as data units). At last, Polyline features to be used as a break or limit in searching for the input sample points can be added.

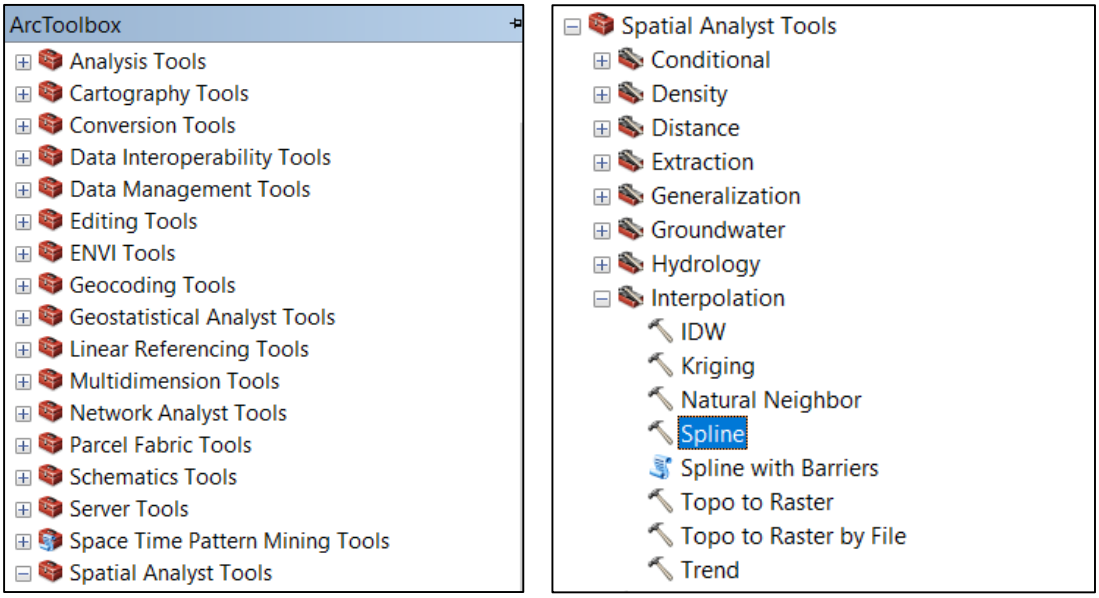
Result:



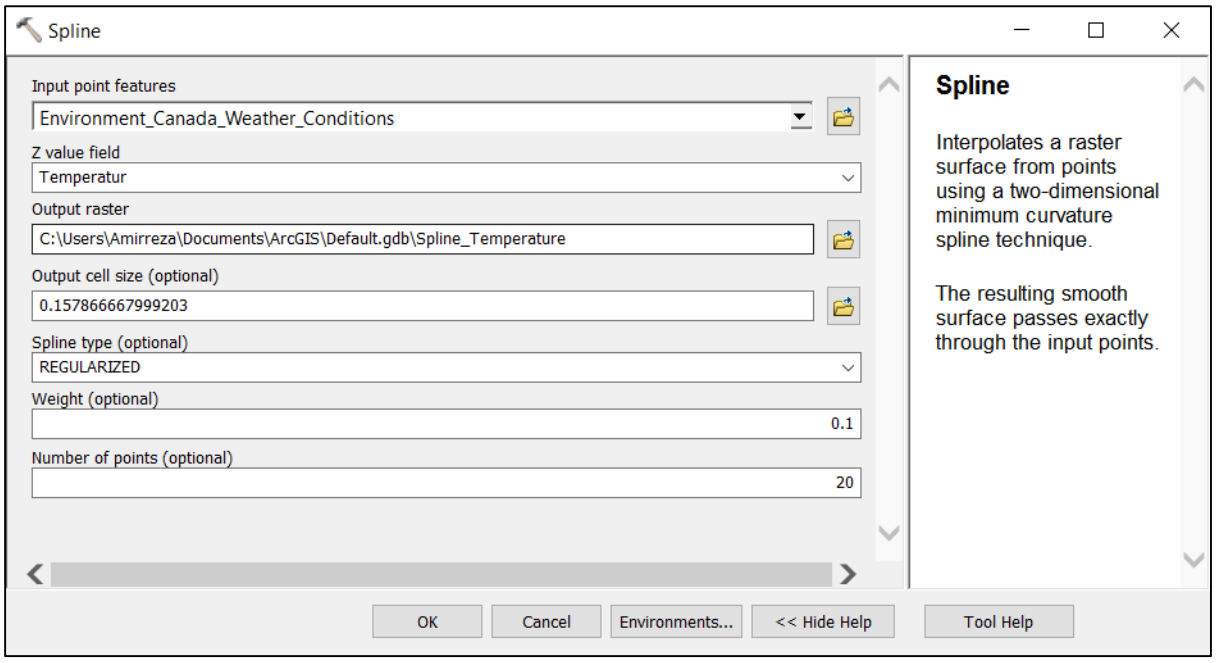
Spline in ArcGIS

To create a raster that estimates whole region temperatures, spline method can be used which interpolates a raster surface from points using a two-

dimensional minimum curvature. first step is to add data, which in here, same as the Python “Canada weather condition” will be used.



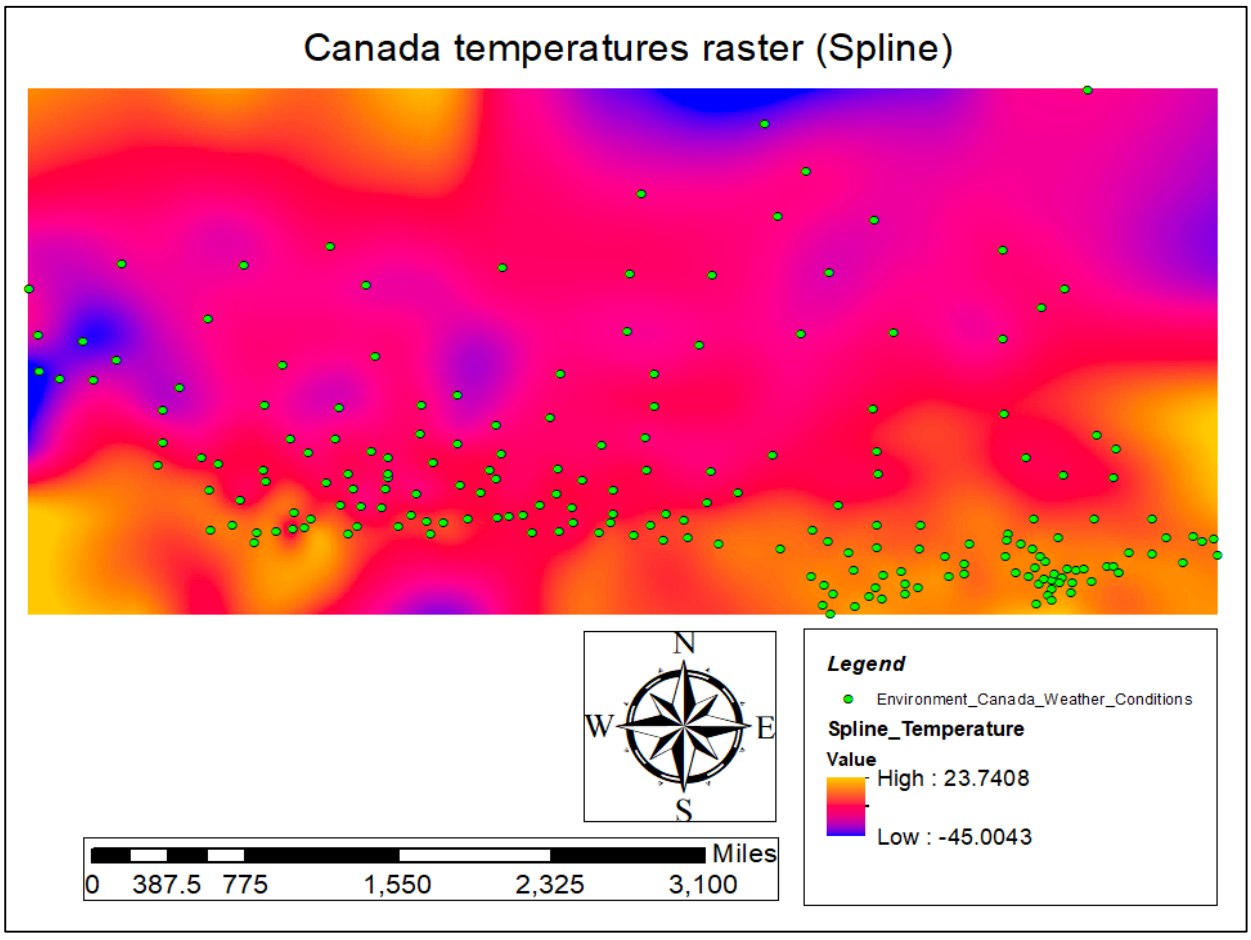
First, open “ArcToolbox” and find “Spatial Analyst Tools” toolbox. Then, click on “interpolation” menu and open “Spline” analysis.



Step 3: Create a Spline raster.

First of all, in Input point features the weather stations shapefile ("Canada weather condition") must be added. Then, the field that holds a magnitude value for each point called Z value field must be chosen ("Temperatur"). In Output cell size, the pixel size of the output raster will be created can be defined.

There are two types of Splines: Regularized and tension. The Regularized Spline type usually produces smoother surfaces than those created with the tension option. Regularized option, higher values used for the weight parameter produce smoother surfaces while, the tension option, higher values entered for the weight parameter result in somewhat coarser surfaces, but surfaces that closely conform to the control points. Another parameter which is needed is weight ("0.1"), It defines the weight of the third derivatives of the surface when regularized option is used. Finally, Number of points that is used in interpolation process is needed ("20"). *Result:*



3.4. Isotropy, Anisotropy

Isotropy

Isotropy comes from the Greek iso, for equal (as in isolateral triangles) and tropos, for direction. Isotropy means that a particular material property is equal in all directions (X, Y and Z). in the other hand, omnidirectional definition is a value that being in or involving all directions; So we can consider isotropy and omnidirectional are the same as each other.

In spatial statistics the assumption of isotropy is very common despite being very restrictive for describing the rich variety of interactions that can characterize spatial processes. This is probably due to a combination of at least two reasons. Isotropic models are obviously mathematically easy to build and, being more parsimonious, the estimation of their parameters is more feasible, in particular when the sample size is small.

Anisotropy

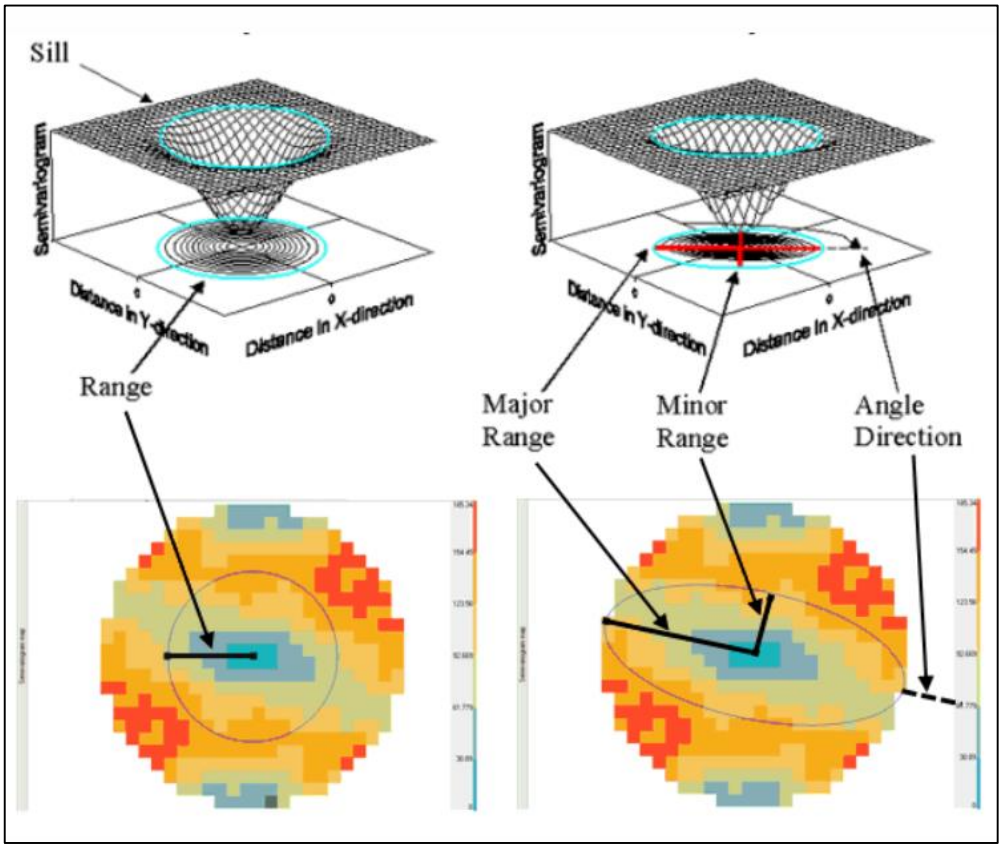
Anisotropy means that the material property might vary depending on direction. In geostatistical analysis, there are two directional influences: one is the trend of a whole region, and the other one is the anisotropic settings in the semivariogram models. The whole-region trend could be estimated through some mathematical estimation models, such as polynomial, and could eliminate influences during the process of geostatistical analysis.

With regard to the isotropy and anisotropy, geostatistical analysis employ the variograms to present the correlations of spatial variability. A variogram model is a distance function. If the variogram of a spatial variance is merely a distance function and the variability does not vary along with spatial directions, this is called “Isotropy”. If the variability varies along with spatial directions, this is called “Anisotropy”. In other words, the anisotropic variogram model is the function of “distance” and “direction”.

For better understanding let’s define an example. Consider two points, si and sj, and the vector that separates them, which is denoted as si - sj. This vector

will have a distance on the x-axis as well as the y-axis. Alternatively, you can think of the vector as having a distance and an angle in polar coordinates. Here anisotropy is described for the semivariogram; the ideas are similar for covariance functions.

The isotropic model is the same in all directions, whereas the anisotropic model reaches the sill more rapidly in some directions than others. The length of the longer axis to reach the sill is called the major range, the length of the shorter axis to reach the sill is called the minor range, and you also have the angle of rotation of the line that forms the major range. In Geostatistical Analyst, an outline of the range is given in blue, over the empirical semivariogram surface.



3.4.1. Anisotropy types

Geometric Anisotropy

The Sill value of the Geometric Anisotropic variogram only vary along with distances, but not along with directions. In other words, when the distances

between samples in various spatial angles are the same, their Sill is the same, but their ranges may vary along with different angles. The direction having biggest range is the direction of variogram. This range is called “Major Range”, and this variogram is called Directional Variogram. The smallest range is called “Minor Range”. If we draw the spatial ranges of all angles on a rose diagram, the shape is an ellipse.

Zonal Anisotropic

The Sill value of the Zonal Anisotropic variogram does not only vary along with all distances, but also varies along with all directions. In other words, when the distances between samples in any spatial angles are the same, their Sill values are not the same, and their ranges also vary along with different angles. Zonal Anisotropic variogram model consists of 2 or more than 2 anisotropic variograms. Natural phenomena usually do not have this kind of anisotropy.

Mixed Anisotropy

The Mixed Anisotropy is a mix of the Geometric Anisotropy and Zonal Anisotropy. Its Sill values and ranges vary along with all directions. Natural phenomena usually have this kind of spatial variability. For example, geological changes usually have a bigger effective range in the horizon direction than in the vertical direction, and changes in the vertical direction have a bigger range than in the horizon direction.

3.4.2. Isotropy in Python

“Gstools” (GeoStatTools) provides geostatistical tools for various purposes such as: random field generation, conditioned field generation, variogram estimation and fitting, directional variogram estimation and modelling and etc. for installation using pip install is fine, but “gstools” could not supported by “numpy” V2.1 since “numpy” V1.9 is needed. To down grade “numpy *pip install -U numpy”* must be used. Then “numpy” and gstools must be imported.

```
import numpy as np
import gstools as gs
```

One of the core-features of “GSTools” is the powerful “CovModel” module which allows you to easily define arbitrary covariance models by yourself. The resulting models provide a bunch of nice features to explore the covariance models. A covariance model is used to characterize the semi-variogram. our self-defined model Gau from the introductory example by setting the exponent as an additional parameter This leads to the so called stable covariance model and we can define it by.

```
class Stab(gs.CovModel):
  def default_opt_arg(self):
    return {"alpha": 1.5}
  def cor(self, h):
    return np.exp(-(h ** self.alpha))
```

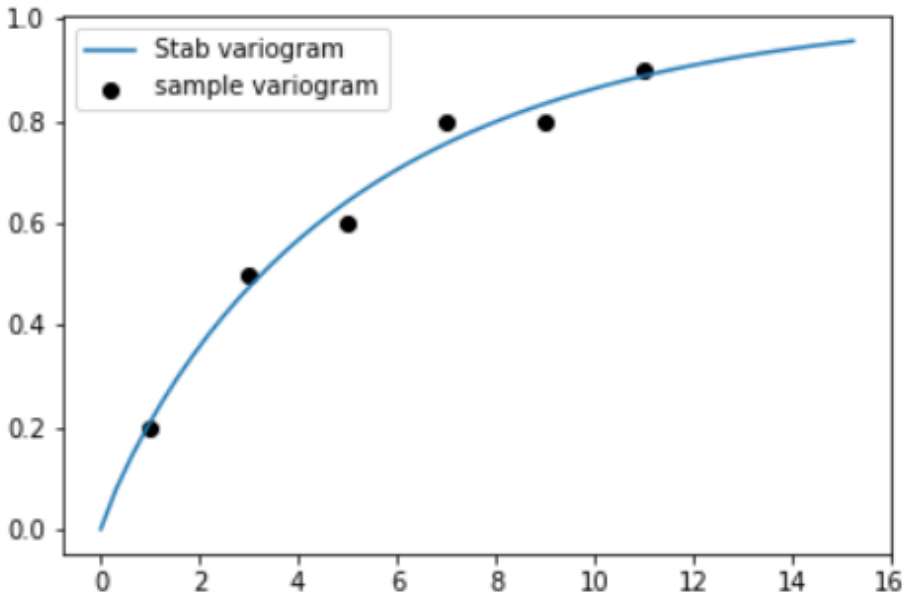
After that, exemplary variogram data (estimated from field observations) is defended. Then, the model which defended in previous section is used and the model boundaries is declared. The result of the model is shown below.

```
bins = [1.0, 3.0, 5.0, 7.0, 9.0, 11.0]
est_vario = [0.2, 0.5, 0.6, 0.8, 0.8, 0.9]
model = Stab(dim=2)
model.set_arg_bounds(alpha=[0, 3])
results, pcov = model.fit_variogram(bins, est_vario, nugget=False)
print("Results:", results)
```

```
Results: {'var': 1.0245740022313239, 'len_scale': 5.081592383444288, 'nugget': 0.0, 'alpha': 0.9067040722375433}
```

Finally, models plot is created and the bins are added to the plot. The plot is shown below:

```
ax = model.plot()
ax.scatter(bins, est_vario, color="k", label="sample variogram")
ax.legend()
```



3.4.3. Anisotropy in python

In this example, we demonstrate how to estimate a directional variogram by setting the estimation directions in 3D. Afterwards we will fit a model to this estimated variogram and show the result. Similar to pervious example, “gstools” and “numpy” is imported. also for plotting and 3D plotting “matplotlib.pyplot” and “mpl_toolkits.mplot3d” will be imported.

```
import matplotlib.pyplot as plt
import numpy as np
from mpl_toolkits.mplot3d import Axes3D
import gstools as gs
```

Then, one should generate synthetic field with anisotropy and rotation by Tait-Bryan angles and create a model. need of the “SRF” module for the actual generation of the spatial random field is essential. But “SRF” also needs a covariance model and we will simply take the Gaussian model. With these simple steps, everything is ready to create first random field

```
dim = 3
angles = [np.deg2rad(90), np.deg2rad(45), np.deg2rad(22.5)]
model = gs.Gaussian(dim=3, len_scale=[16, 8, 4], angles=angles)
x = y = z = range(50)
pos = (x, y, z)
srf = gs.SRF(model, seed=1001)
field = srf.structured(pos)
```

Then, the axes of the rotated coordinate system will be generated to get an impression what the rotation angles do.

```
main_axes = gs.rotated_main_axes(dim, angles)
axis1, axis2, axis3 = main_axes
```

Now the variogram along the main axes will be estimated. When the main axes are unknown, one would need to sample multiple directions and look for the one with the longest correlation length (flattest gradient). Then check the transversal directions and so on.

```
bin_center, dir_vario, counts = gs.vario_estimate(
pos,
field,
direction=main_axes,
bandwidth=10,
sampling_size=2000,
sampling_seed=1001,
mesh_type="structured",
return_counts=True,
)
```

Afterwards by use of the estimated variogram to fit a model to it. Note, that the rotation angles need to be set beforehand.

```
print("Original:")
print(model)
model.fit_variogram(bin_center, dir_vario)
print("Fitted:")
print(model)
```

The result of the model is shown below. As you realize the fitted model is so close to original data which shows the performance of this model. Finally, plotting main axes and the fitted directional variogram. The plots are shown below:

```
Original:
Gaussian(dim=3, var=1.0, len_scale=16.0, nugget=0.0, anis=[0.5, 0.25], angles=[1.57, 0.785, 0.393])
Fitted:
Gaussian(dim=3, var=0.972, len_scale=13.0, nugget=0.0138, anis=[0.542, 0.281], angles=[1.57, 0.785, 0.393])
```

```
fig = plt.figure(figsize=[10, 5])
ax1 = fig.add_subplot(121, projection=Axes3D.name)
ax2 = fig.add_subplot(122)
ax1.plot([0, axis1[0]], [0, axis1[1]], [0, axis1[2]], label="0.")
ax1.plot([0, axis2[0]], [0, axis2[1]], [0, axis2[2]], label="1.")
ax1.plot([0, axis3[0]], [0, axis3[1]], [0, axis3[2]], label="2.")
ax1.set_xlim(-1, 1)
ax1.set_ylim(-1, 1)
ax1.set_zlim(-1, 1)
ax1.set_xlabel("X")
ax1.set_ylabel("Y")
ax1.set_zlabel("Z")
ax1.set_title("Tait-Bryan main axis")
ax1.legend(loc="lower left")
x_max = max(bin_center)
ax2.scatter(bin_center, dir_vario[0], label="0. axis")
ax2.scatter(bin_center, dir_vario[1], label="1. axis")
ax2.scatter(bin_center, dir_vario[2], label="2. axis")
model.plot("vario_axis", axis=0, ax=ax2, x_max=x_max, label="fit on axis 0")
model.plot("vario_axis", axis=1, ax=ax2, x_max=x_max, label="fit on axis 1")
model.plot("vario_axis", axis=2, ax=ax2, x_max=x_max, label="fit on axis 2")
ax2.set_title("Fitting an anisotropic model")
ax2.legend()
plt.show()
```