Geospatial Data II

This section provides an overview of Geospatial data visualization; Geospatial Python Libraries; and most importantly Basic Spatial Analysis Techniques like Point Pattern Analysis, Spatial Interpolation, and Spatial Correlation models.

Spatial statistics

Spatial statistics is a branch of statistics dedicated to analyzing data tied to spatial locations. Unlike conventional statistical methods, spatial statistics incorporate spatial dependence, recognizing that spatially proximate data points are often more alike than distant ones. This approach is crucial for accurately modeling and interpreting spatially distributed data.

Spatial statistics has applications in many domains. In the context of environmental data analytics, we are particularly concerned with a subfield of spatial statistics called **geostatistics**.

Geostatistics focuses primarily on the modeling and prediction of **spatially continuous phenomena** (e.g., soil properties, temperature fields) often based on sampled data.

Geostatistics is founded on several key concepts, many of which are also relevant in the broader field of spatial statistics. Here we will focus on four of these key concepts, namely spatial autocorrelation, point pattern analysis, spatial interpolation, and Spatial Regression.

A) Spatial Autocorrelation

Spatial autocorrelation refers to the correlation of a variable with itself through space. It quantifies the degree to which objects or values located near each other in geographic space are similar or dissimilar.

- **Positive Spatial Autocorrelation**: Occurs when geographically proximate locations have similar values. For example, areas with high rainfall are often surrounded by other areas with high rainfall. This clustering of similar values indicates positive autocorrelation.
- **Negative Spatial Autocorrelation**: Occurs when neighboring locations have dissimilar values. For instance, high property values may be adjacent to areas with low property values, indicating a pattern of dispersion.

- **No Spatial Autocorrelation**: When there is no discernible pattern between neighboring locations, meaning that the spatial distribution is random.

Spatial autocorrelation measures can be categorized into two groups:

1. Global Measures

Global measures of spatial autocorrelation assess the overall pattern of spatial dependence across an entire area. These statistics provide a single summary value that indicates whether spatial data exhibit clustering (positive spatial autocorrelation), dispersion (negative spatial autocorrelation), or randomness (no spatial autocorrelation). Global measures are useful for understanding general spatial trends but may fail to capture local variations or specific clusters (hotspots) and outliers within the dataset.

One widely used global measure is **Moran's I**, which evaluates the overall similarity between values at neighboring locations.

$$I = \frac{N}{W} \times \frac{\sum_i \sum_j w_{ij}(x_i - \bar{x})(x_j - \bar{x})}{\sum_i (x_i - \bar{x})^2}$$

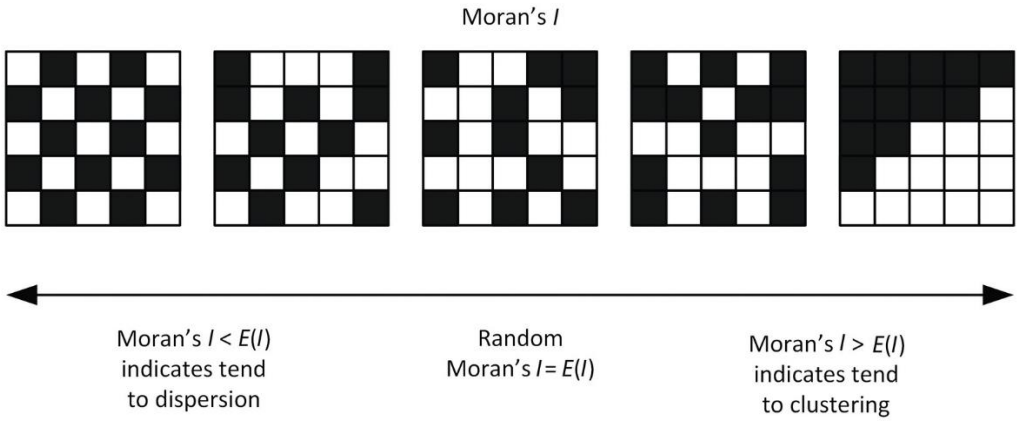
where:

- N is the number of observations.
- x_i and x_j are the values at locations i and j, respectively.
- $\bar{x}$ is the mean of the variable.
- w_{ij} is the spatial weight, indicating the degree of spatial proximity between locations i and j.
- W is the sum of the weights.

- Moran's I values range from **-1** (perfect dispersion) to **+1** (perfect clustering), with **0** indicating random spatial distribution.

In Moran's I formula, the spatial weight ω_{ij} represents the strength of the spatial relationship between locations i and j. This weight ω_{ij} can be defined in various ways, often based on distance or other criteria such as adjacency, but it is not necessarily the same as distance itself. Instead, ω_{ij} may represent a function of distance or connectivity between locations. A common way to define ω_{ij} is to use the **inverse distance,** giving

more weight to closer neighbors, assuming that spatial autocorrelation decreases as the distance between points increases.



Moran's I as a global measure of spatial autocorrelation.

2) Local Measures

Local measures, often referred to as **Local Indicators of Spatial Association (LISA)**, assess spatial autocorrelation at a specific location or small region within the study area. They are used to detect spatial heterogeneity, meaning that spatial autocorrelation may vary across the dataset. Local measures help identify clusters of similar or dissimilar values (e.g., hotspots or cold spots) and spatial outliers. The most common local measure of spatial autocorrelation is **Local Moran's I**. It is a local version of the global Moran's I statistic:

$$I_i = \frac{(x_i - \bar{x}) \sum_j w_{ij}(x_j - \bar{x})}{\sum_i (x_i - \bar{x})^2 / n}$$

where:

- x_i is the value at location i,
- $\bar{x}$ is the mean of the variable,
- w_{ij} is the spatial weight between locations i and j,
- n is the number of observations.

- I_i can theoretically range from **-1** to **+1**, but in practice, the range might not reach these extremes.
  - $I_i > 0$: Positive local spatial autocorrelation (similar values cluster together).
  - $I_i < 0$: Negative local spatial autocorrelation (dissimilar values are near each other).
  - $I_i = 0$: No significant local spatial autocorrelation (random distribution of values).

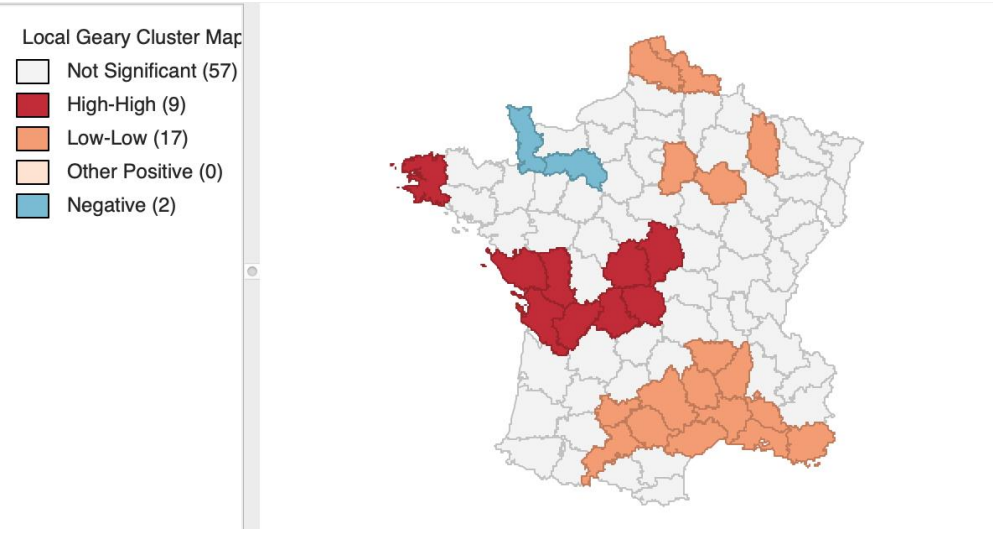
Local Moran's I detects clusters of similar values (high-high or low-low) and identifies spatial outliers (high-low or low-high) within a dataset:

High-High clusters: A high value (positive and significant) indicates that the location has a high value and is surrounded by neighbors with high values (hotspot).

Low-Low clusters: A high positive also means that the location has a low value and is surrounded by other low values (cold spot).

High-Low or Low-High clusters: Indicate spatial outliers, where a location with a high value is surrounded by low-value neighbors (High-Low), or a location with a low value is surrounded by high-value neighbors (Low-High). These patterns highlight significant local deviations from the surrounding spatial context.

In **Local Moran's I maps**, the **statistically insignificant areas** represent locations where the calculated Local Moran's I value does not show a strong or reliable spatial autocorrelation. In other words, these locations do not exhibit a clear pattern of clustering or dispersion that is distinguishable from random chance. This may random distribution, no strong spatial dependence or unclear spatial pattern.



Sample map showing local spatial correlation.

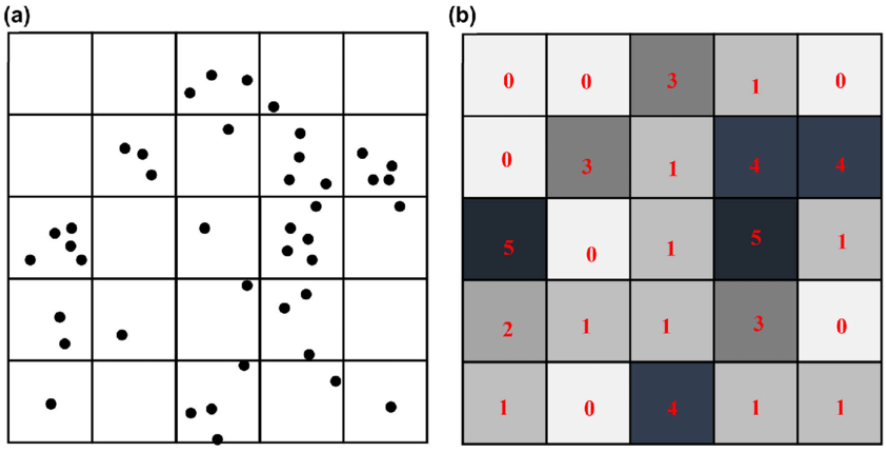
B) Point Pattern Analysis

Point pattern analysis in geostatistics is the study of the spatial arrangement or distribution of individual points. Such points can represent the existence of an object or event at a specific location. Point pattern analysis seeks to determine whether such objects or events are randomly distributed, clustered (grouped together), or regularly spaced (dispersed) within the study area.

In environmental data analytics, point pattern analysis can be applied in various ways. For example, it can be used to analyze the spatial distribution of trees in a forest to understand ecological processes such as competition for resources or species coexistence. It can also be used to study the spatial pattern of wildfire incidents across a large region to identify high-risk areas for fire outbreaks. Additionally, point pattern analysis can investigate the spatial distribution of groundwater extraction wells to assess resource usage and potential environmental impacts, such as over-extraction or groundwater depletion. These are just a few examples among many potential applications. Several statistical techniques are used to analyze and interpret point patterns. Here we will review just one example of such methods.

Quadrat Analysis

Quadrat Analysis is a method in point pattern analysis that helps determine whether a spatial distribution of points is random, clustered, or dispersed. The study area is divided into smaller, equally sized subregions called quadrats, and the number of points within each quadrat is counted. The distribution of these counts is then compared to a theoretical distribution (often a Poisson distribution) to assess the pattern of the points. Here are the steps in Quadrat Analysis:



Quadrat count method: (a) point (event) locations in an area overlay by N 9 N contiguous grid size; (b) number of points observed in each quadrat.

1. Study Area and Quadrat Setup:
   - The study area A is divided into N equal quadrats, each with an area of A_q such that:

$$A_q = \frac{A}{N}$$

   - Let x_i represent the number of points in quadrat i, for $i = 1, 2, ..., N$.
2. Mean Number of Points per Quadrat ($\bar{x}$):
   - The mean number of points per quadrat is calculated as:

$$\bar{x} = \frac{\sum_{i=1}^{N} x_i}{N}$$

     where:
     - x_i is the number of points in quadrat i.
     - N is the total number of quadrats.
3. Variance of the Number of Points per Quadrat (s^2):
   - The variance in the number of points across quadrats is calculated as:

$$s^2 = \frac{\sum_{i=1}^{N} (x_i - \bar{x})^2}{N - 1}$$

     This variance indicates how spread out the counts are across quadrats. If the variance is high, it suggests more variability in the counts, which could indicate clustering or dispersion.
4. Variance-to-Mean Ratio (VMR):
   - The **variance-to-mean ratio** (also called the **Index of Dispersion**) is calculated as:

$$VMR = \frac{s^2}{\bar{x}}$$

     The VMR helps determine whether the point pattern is random, clustered, or dispersed:
     - **VMR ≈ 1**: The points are likely randomly distributed (Complete Spatial Randomness, CSR).
     - **VMR > 1**: The points are clustered (there is more variability in the quadrat counts than expected under randomness).
     - **VMR < 1**: The points are dispersed (less variability in quadrat counts, with points more evenly spaced than expected under randomness).

5. Chi-Square Test for Randomness:
    - To statistically test whether the point distribution differs significantly from a random distribution, a **chi-square test** can be applied. The chi-square statistic is computed as:

$$\chi^2 = \sum_{i=1}^{N} \frac{(x_i - \bar{x})^2}{\bar{x}}$$

      This chi-square value is then compared to the critical value from a chi-square distribution with $N - 1$ degrees of freedom to assess whether the observed distribution is significantly different from a random distribution.

6. Hypothesis Testing:
    - **Null Hypothesis** (H_0): The point pattern follows a random (Poisson) distribution.
    - **Alternative Hypothesis** (H_a): The point pattern does not follow a random distribution (it is either clustered or dispersed).
    - If the calculated chi-square value exceeds the critical value, the null hypothesis is rejected, indicating that the point pa[illegible]n is not random.

C) Spatial Interpolation

Spatial interpolation is a technique used to estimate values at unsampled locations based on the values of nearby sampled points. It leverages the principle of spatial autocorrelation, which assumes that points closer together are more likely to have similar values. Spatial interpolation methods generate a continuous surface from discrete spatial data, allowing analysts to predict unknown values across a study area.

Spatial interpolation is crucial in environmental data analytics because environmental data (e.g., air quality, soil properties, rainfall, temperature) are often collected at a limited number of locations. Interpolation allows for the estimation of values across the entire study region, filling in gaps between sampled points. This is essential for creating continuous maps of environmental phenomena.

There are numerous methods for spatial interpolation. Here we focus on a method that is very commonly used in environmental data analytics.

Kriging

Kriging is a geostatistical interpolation method that not only estimates unknown values at unsampled locations but also provides a measure of the uncertainty (variance)

associated with those estimates. Kriging is based on the concept of spatial autocorrelation, which assumes that points closer to each other are more likely to have similar values. It uses the spatial structure of the data, often modeled through a **semivariogram**, to make predictions and assess uncertainty.

In its basic form, the Kriging method is formulated as follows:

Kriging Estimate: The value $Z^*(s_0)$ at an unsampled location s_0 is estimated as a weighted linear combination of the known values at nearby locations:

$$Z^*(s_0) = \sum_{i=1}^{n} \lambda_i Z(s_i)$$

where:

- $Z(s_i)$ is the known value at location s_i,
- λ_i is the weight assigned to the known value at location s_i,
- n is the number of sampled points used in the estimation.

Weights (λ_i): The weights λ_i are chosen based on the spatial structure of the data, represented by the **semivariogram**. The semivariogram measures how the similarity between points changes as the distance between them increases.

Semivariogram ($\gamma(h)$): The semivariogram $\gamma(h)$ is a function of the distance h between two points:

$$\gamma(h) = \frac{1}{2}\mathbb{E}\left[(Z(s_i) - Z(s_i + h))^2\right]$$

It reflects how the variance between point values changes with increasing separation distance.

In practice, because the expected value is often unknown, the **empirical semivariogram** is computed as an approximation, which is indeed based on the **average** of the squared differences between all pairs of data points separated by a given distance h. The formula for the empirical semivariogram is:

$$\hat{\gamma}(h) = \frac{1}{2N(h)} \sum_{i=1}^{N(h)} (Z(s_i) - Z(s_i + h))^2$$

Where:

- $\hat{\gamma}(h)$ is the empirical semivariogram,
- $N(h)$ is the number of data point pairs separated by distance h,
- $Z(s_i)$ and $Z(s_i + h)$ are the values at locations s_i and $s_i + h$,
- The sum is taken over all pairs of points separated by distance h.

The kriging weights λ_i are calculated by solving a system of **ordinary kriging equations**. These equations are derived by minimizing the **kriging variance** (the variance of the prediction error) while ensuring that the sum of the weights equals 1 (to ensure an unbiased estimate).

The **ordinary kriging system** is:

$$\sum_{j=1}^{n} \lambda_j \gamma(s_i - s_j) + \mu = \gamma(s_i - s_0) \quad \text{for } i = 1, 2, \ldots, n$$

$$\sum_{j=1}^{n} \lambda_j = 1$$

Where:

- λ_j are the kriging weights for each known data point s_j,
- $\gamma(s_i - s_j)$ is the semivariogram value between data points s_i and s_j,
- $\gamma(s_i - s_0)$ is the semivariogram value between known point s_i and the unknown location s_0,
- μ is a Lagrange multiplier used to enforce the unbiased constraint.

To solve for the weights λ, you need the semivariogram values between all pairs of known points and between each known point and the unknown location. This results in a **matrix equation** that can be solved for the weights.

Steps:

1. **Compute Semivariogram Values**: Use the **empirical semivariogram** $\hat{\gamma}(h)$ to compute the semivariogram values for all pairs of known points $s_1, s_2, \ldots, s_n$, as well as the semivariogram values between each known point and the prediction location s_0.
2. **Set Up the Kriging Matrix**:
   - Construct a matrix of semivariogram values between all known points. The matrix will be symmetric, with elements $\gamma(s_i - s_j)$, where s_i and s_j are known points.
   - Create a vector of semivariogram values between each known point and the prediction location s_0, denoted as $\gamma(s_i - s_0)$.
3. **Solve the Kriging Equations**:
   - Solve the system of equations formed by the kriging matrix and the semivariogram vector to obtain the weights $\lambda_1, \lambda_2, \ldots, \lambda_n$.
   - The matrix system is solved using linear algebra techniques (e.g., matrix inversion).

The kriging system can be written in matrix form as:

$$\begin{pmatrix} \gamma(s_1 - s_1) & \gamma(s_1 - s_2) & \ldots & \gamma(s_1 - s_n) & 1 \\ \gamma(s_2 - s_1) & \gamma(s_2 - s_2) & \ldots & \gamma(s_2 - s_n) & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \gamma(s_n - s_1) & \gamma(s_n - s_2) & \ldots & \gamma(s_n - s_n) & 1 \\ 1 & 1 & \ldots & 1 & 0 \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_n \\ \mu \end{pmatrix} = \begin{pmatrix} \gamma(s_1 - s_0) \\ \gamma(s_2 - s_0) \\ \vdots \\ \gamma(s_n - s_0) \\ 1 \end{pmatrix}$$

Where:

- The left matrix is the semivariogram matrix (computed using the empirical semivariogram $\hat{\gamma}(h)$),
- The vector of unknowns $\lambda_1, \lambda_2, \ldots, \lambda_n, \mu$ is what we are solving for,
- The right-hand side vector contains the semivariogram values between each known point and the unknown point, and a 1 for the constraint.

By solving this system of equations, you get the **kriging weights** λ_i. These weights determine how much influence each known point s_i has on the prediction at s_0. Points that are closer to s_0 (with smaller semivariogram values) will generally have higher weights, while points farther away will have smaller weights due to larger semivariogram values (indicating less similarity).

There are many variants of the Kriging method, each differing in how they handle trends, data assumptions, and constraints. Here we introduce 3 of these variants:

1) Ordinary Kriging (OK) (As discussed above):

- Assumes the mean of the data is constant but unknown over the study area.
- Weights are determined based solely on spatial autocorrelation as modeled by the semivariogram.
- Most used form of kriging.

2) Universal Kriging (UK):

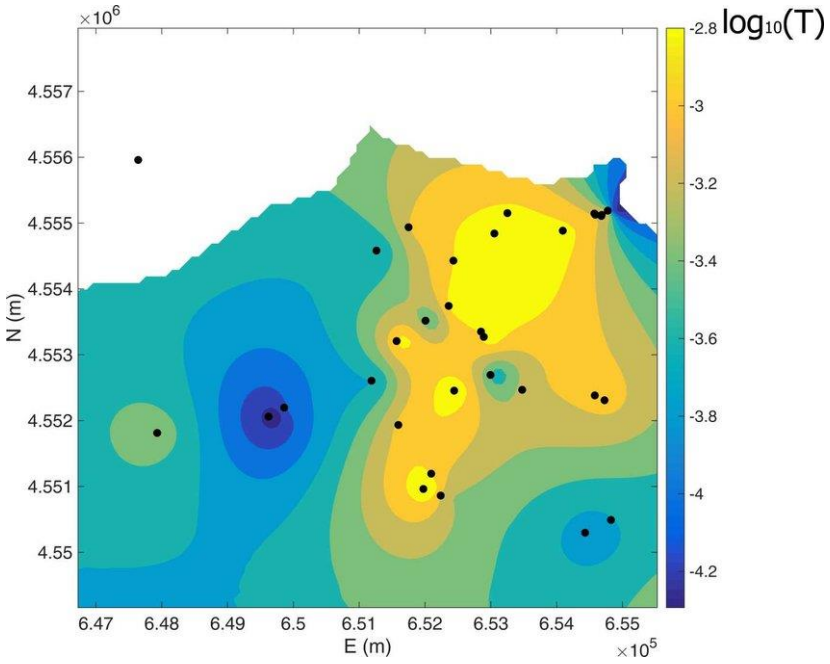
- Accounts for a spatial trend or drift in the data, meaning that the mean is not constant across the study area.
- Incorporates both a deterministic trend and spatial autocorrelation in the kriging model.
- Often used when there is a clear trend in the data (e.g., elevation increasing with latitude).

3) Cokriging:

- A multivariate extension of kriging that interpolates multiple correlated variables simultaneously.
- Uses the correlation between variables to improve the estimation of a target variable (e.g., using soil type to predict crop yield).

D) Spatial Regression

In geostatistics, **spatial regression** refers to a set of statistical techniques used to model relationships between a dependent variable and one or more independent variables while **explicitly accounting for the spatial arrangement** of data points. Traditional regression models assume that observations are independent of each other, but in spatial data, nearby locations often exhibit similar values due to spatial autocorrelation. Spatial regression addresses this by incorporating spatial relationships into the analysis, making it more suitable for geographically distributed data.



Example output of Ordinary Kriging interpolation.

Using spatial regression, compared to traditional regression, tends to improve the accuracy of predictions in environmental data analytics by incorporating the spatial structure of the data.

There are different types of spatial regression models. In the following one popular method is introduced.

Geographically Weighted Regression (GWR)

Geographically Weighted Regression (GWR) is a spatial regression technique that allows for the estimation of local, rather than global, relationships between variables. Unlike traditional regression models, which assume that the relationship between independent variables (predictors) and the dependent variable is the same across all locations, GWR allows the relationships to vary spatially.

Key Concepts of GWR:

1) **Local Parameter Estimation**: GWR estimates a separate set of regression coefficients for each location in the study area. The idea is that the relationship between the dependent and independent variables may change across space.

2) **Spatial Weights**: GWR uses spatial weights to assign more influence to nearby observations when estimating the local regression coefficients. This means that for each location, data points that are geographically closer are given more weight in the local regression, while distant points have less influence.

Unlike traditional regression models, spatial regression explicitly considers the influence of the spatial structure of the data, addressing the fact that observations closer together in space may be more similar than those further apart.

The general form of a **linear regression** model is:

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_k x_{ik} + \epsilon_i$$

Where:

- y_i is the dependent variable at location i,
- $x_{i1}, x_{i2}, \ldots, x_{ik}$ are the independent variables (predictors),
- $\beta_0, \beta_1, \ldots, \beta_k$ are the regression coefficients,
- ϵ_i is the error term at location i.

In **GWR**, the coefficients β_k are allowed to vary across space, so the model becomes:

$$y_i = \beta_0(u_i, v_i) + \beta_1(u_i, v_i) x_{i1} + \beta_2(u_i, v_i) x_{i2} + \cdots + \beta_k(u_i, v_i) x_{ik} + \epsilon_i$$

Where:

- (u_i, v_i) represents the coordinates (e.g., latitude and longitude) of location i,
- $\beta_k(u_i, v_i)$ are the regression coefficients that vary with location.

The coefficients β for location i are not estimated using data from just location i alone, but rather using data from **all other locations** in the dataset, with nearby locations having a larger influence and distant locations having less influence. A **weighting function** defines how much influence each observation j has on the local regression at location i. For example, if location j is close to location i, the weight will be large, meaning that observation j will have a strong influence on the estimation of the local coefficients at location i. If location j is far away from ii, the weight will be small, meaning that observation j will have little influence.

Each location i has its own local regression, where the weights assigned to the observations in the dataset depend on their proximity to location i. The weighting function w_{ij} can be any distance-based function, but a common choice is a **Gaussian** kernel, defined as:

$$w_{ij} = \exp\left(-\frac{d_{ij}^2}{2h^2}\right)$$

Where:

- d_{ij} is the distance between location i and location j,
- h is the bandwidth, which determines the width of the neighborhood (how much weight is given to distant points).

Alternatively, other kernel functions like **bi-square** can be used.

For each location (u_i, v_i), GWR solves a weighted least squares regression problem, where the weights are derived from the spatial proximity of other data points:

$$\hat{\beta}(u_i, v_i) = \left(X^T W_i X\right)^{-1} X^T W_i y$$

Where:

- X is the matrix of independent variables,
- y is the vector of observed values for the dependent variable,
- W_i is the diagonal matrix of spatial weights for location i, with each diagonal element corresponding to the weight w_{ij} for observation j.

Python-based geostatistics

There are several widely used Python libraries that provide support for geostatistics, including tools for spatial autocorrelation, point pattern analysis, spatial interpolation, and spatial regression. Below are some of the most used libraries:

1. PySAL (Python Spatial Analysis Library)

PySAL is the most comprehensive and widely used library for spatial data analysis in Python, and it includes support for many geostatistical techniques.

Key Features:

- **Spatial Autocorrelation**: PySAL includes tools to compute global and local measures of spatial autocorrelation, such as Moran's I and Local Indicators of Spatial Association (LISA).
- **Spatial Regression**: PySAL has modules for spatial econometrics, including geographically weighted regression (GWR).
- **Point Pattern Analysis**: It supports point pattern analysis.
- **Spatial Interpolation**: PySAL includes methods for kriging and inverse distance weighting (IDW) for spatial interpolation.

2. Geostatistical Modeling Library (GSTools)

GSTools is a specialized Python library for geostatistics, focusing on kriging and variogram modeling. GSTools is a good tool when you want to perform advanced kriging models.

3. Geopandas

GeoPandas extends the capabilities of pandas to handle spatial data. While it does not natively support many geostatistical algorithms, it integrates well with PySAL for those advanced functionalities, serving as a tool for reading files, pre-processing, and visualizing spatial data.

Geospatial Data Visualization

Geospatial data visualization involves the graphical representation of data that is tied to specific geographic locations, enabling users to analyze patterns, trends, and relationships based on location. While it shares many concepts, methods, and tools with general data visualization, it introduces unique challenges and techniques due to the inherent spatial component. Geospatial visualization includes specialized methods like heat maps, choropleth maps, and 3D terrain models, and requires handling geographic projections, spatial relationships, and layers of location-based data, making it essential for applications in fields such as environmental monitoring, urban planning, and navigation.

Location-based data can be visualized by plotting the spatial coordinates (e.g., latitude and longitude) on a 2D or 3D grid. Visualizing spatial data in Python is straightforward with libraries like matplotlib, plotly, and pyvista, which support various types of 2D and 3D visualizations, including scatter plots, surface plots, and contour maps. These libraries are standalone, requiring no connection to external software or services,

enabling users to visualize data directly within Python environments. However, when geospatial data requires geographic context—such as overlaying points or features on maps of streets, buildings, or terrain—Python can be integrated with external mapping services like OpenStreetMap, Google Maps, or Google Earth using libraries like folium and OSMnx. These connections allow data to be displayed in a browser or interactive map interface, providing real-world context for the spatial data.

Overlaying

In the context of geospatial data visualization, overlaying means placing spatial data (e.g., points, lines, or polygons representing things like locations, roads, or regions) on top of a base map that shows real-world features such as streets, buildings, terrain, or satellite imagery.

The spatial data and the base map must use the same geographic coordinate system or projection to ensure proper alignment. For example, both must use latitude and longitude or another compatible system. Furthermore, the precision of the overlay depends on the quality of the data. Low-resolution data may not align well with detailed base maps, leading to inaccuracies in visual representation.

Vehicle locations overlaid on an OpenStreetMap base layer.

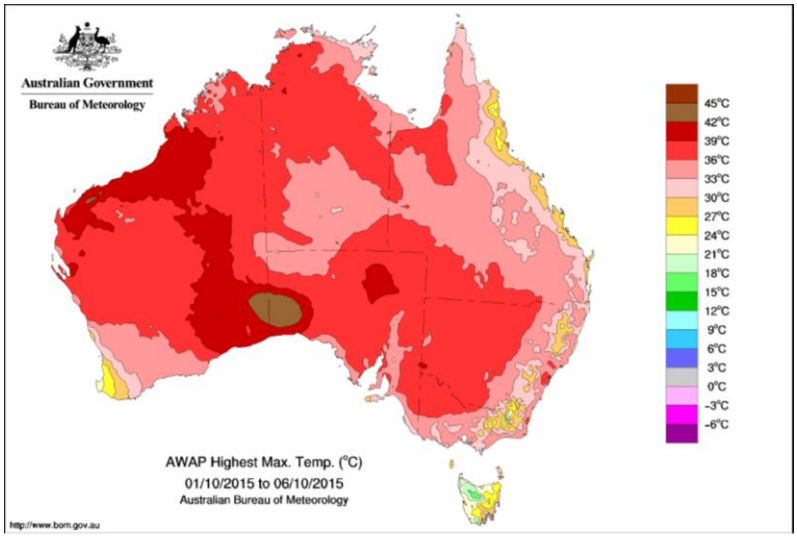
Choropleth Maps

Choropleth mapping is a data visualization technique used in geospatial data to represent the distribution of a variable across predefined regions, such as countries, states, or districts. In a choropleth map, geographic areas are filled with varying shades

or colors based on the value of the data associated with that area. Typically, darker or more intense colors represent higher values, while lighter colors indicate lower values.

Choropleth mapping is generally used for **discrete** variables, especially those that are aggregated over geographic areas (i.e., data is summarized or averaged for specific regions or zones, rather than being shown at every individual point within those regions.). Each region is shaded or colored based on the value of the variable, making it suitable for showing region-specific data.

However, choropleth maps can also be used for **continuous** variables, but this is less common. When used with continuous variables (e.g., temperature or elevation averaged over regions), the values are still aggregated to predefined geographic areas, and color gradients are applied to represent different ranges of the continuous variable.



An example of a Choropleth Map.

In choropleth mapping, the choice of classification method significantly affects how quantitative data is represented across geographic areas. Different classification strategies divide the data into categories or classes, which are then assigned colors on the map. The most common strategies for quantitative data classification in choropleth maps include:

1) Equal Interval Classification

Divides the range of data into equal-sized intervals. It is suitable when the data range is uniform and the focus is on showing how values are distributed evenly across the

range. This classification is easy to interpret and ensures that the same range of values is assigned to each class. But it can lead to misleading visualizations if the data is skewed, as some classes may have few or no observations.

- ***Example***: If the data ranges from 0 to 100 and you want 5 classes, each class would cover an interval of 20 units (0–20, 21–40, and so on).

2) Quantile Classification

Divides the data so that each class contains an equal number (or proportion) of data points. It is useful when you want each category to have the same number of regions or areas. This method ensures that all classes have data, which can help highlight spatial patterns. But for data with wide variability may lead to uneven intervals, where some classes span large ranges and others cover narrow ranges.

- ***Example***: If there are 100 regions, a quantile classification with 5 classes will assign 20 regions to each class.

3) Standard Deviation Classification

Divides the data based on how much values deviate from the mean (average). Class boundaries are typically set at 1 or 0.5 standard deviation intervals. It is useful for emphasizing how values diverge from the average, especially when interested in showing outliers or values that are far from the norm. This method clearly highlights regions that are above or below the average. But it is not ideal for data that doesn't follow a normal distribution, as the method assumes a bell-curve distribution.

- ***Example***: A dataset with a mean of 50 and a standard deviation of 10 would create classes like 0–40, 40–50, 50–60, 60–70, etc.

Contour maps

Contour maps are used to represent continuous data over a 2D plane, where lines (contours) connect points of equal value. These are often used to visualize data such as elevation, temperature, or pressure, where the data changes smoothly over space. The contour lines help to show areas of equal value, and the space between the lines indicates how rapidly the values are changing.

An example of a contour plot showing elevation.

Hotspot maps

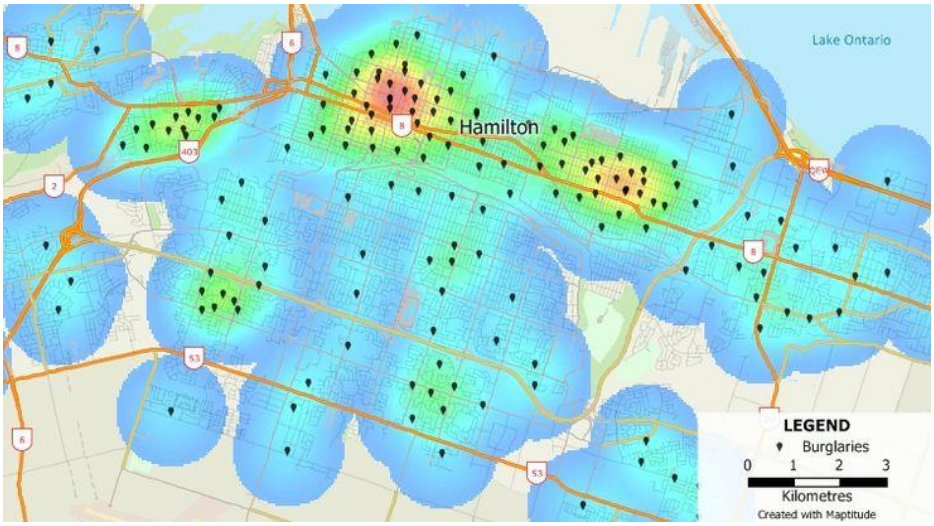
Hotspot maps are focused on showing where events occur frequently. These areas are called "hotspots" because they indicate heightened activity. These maps are commonly used to identify **clusters** or **areas of intensity** within a geographic region. In a hotspot map, regions with more events (or higher values of the variable being mapped) are often marked with warmer or more intense colors (like red), while regions with fewer events are marked with cooler colors (like blue or green).

In environmental data analytics, hotspot maps are widely used to visualize and analyze spatial patterns in phenomena such as wildfires, pollution incidents, and species observations.

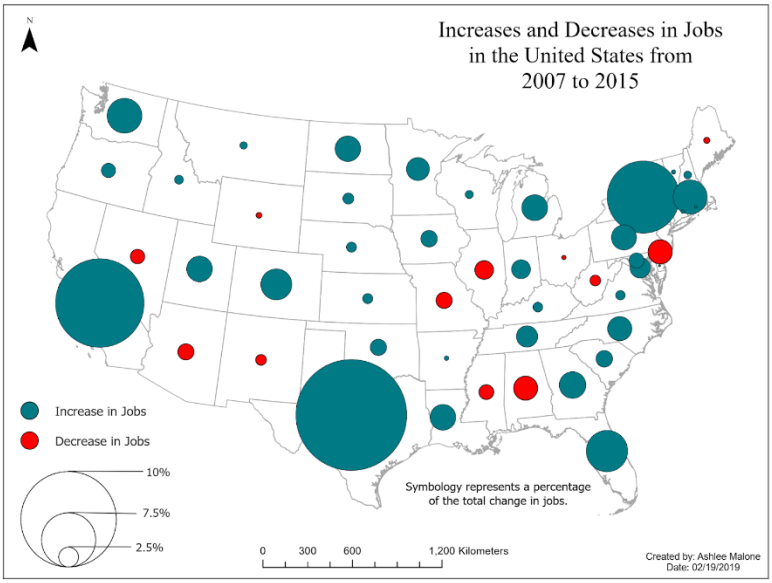
Proportional Symbol Maps

Proportional symbol maps use symbols (e.g., circles, squares) whose size varies in proportion to the value of the data being represented. Larger symbols indicate higher values, while smaller symbols indicate lower values. Common use cases in environmental data analytics include visualizing the amount of waste generated or improperly disposed of across various regions, mapping the capacity or output of renewable energy sources (e.g., wind farms, solar plants, hydroelectric dams) by geographic location, and illustrating the volume of available freshwater resources (e.g., groundwater or reservoir levels) in different areas. These visualizations help highlight

regional patterns and trends, making it easier to identify areas that require targeted interventions or further analysis.



An example of a hotspot map showing incidents of burglary in a city.



An example of Proportional Symbol Maps showing increase or decrease in jobs in the US.