Assignment 1

By

Parsa Besharat

An assignment's handout submitted as part of the requirements

for the lecture, Geomodeling, of MSc Mathematics of Data and Resources Sciences

at the Technische Universität Bergakademie Freiberg

November, 2024

Supervisor: Prof. Jörg Benndorf

Task 1:

A)

1. The random variable is defined via this formula:

$$X = 1.5 \times (Y_1 + Y_2)$$

Where Y_1 and Y_2 are the outcomes of the two dice rolls, which are independent and uniformly distributed over {1,2,3,4,5,6}, their sum $Y = Y_1 + Y_2$ can take values from 2 to 12.

2. Also, since X is scaled by 1.5, the possible values of X are:

$$X \in \{3.0,4.5,6.0,7.5,9.0,10.5,12.0,13.5,15.0,16.5,18.0\}$$

3. For Probability Mass Function of Y, the probability of each value of $Y = Y_1 + Y_2$ can be computed based on the number of ways to achieve that sum:

$$P(Y = k) = \frac{Number\ of\ outcomes\ resulting\ in\ k}{36},$$

$$k \in \{2, \dots, 12\}$$

Therefore, we have:

- $P(Y = 2) = \frac{1}{36}$
- $P(Y = 3) = \frac{2}{36}$
- $P(Y = 4) = \frac{3}{36}$
- $P(Y = 5) = \frac{4}{36}$
- $P(Y = 6) = \frac{5}{36}$
- $P(Y = 7) = \frac{6}{36}$

- $P(Y = 8) = \frac{5}{36}$
- $P(Y = 9) = \frac{4}{36}$
- $P(Y = 10) = \frac{3}{36}$
- $P(Y = 11) = \frac{2}{36}$
- $P(Y = 12) = \frac{1}{36}$

This pattern is symmetric because the probability of rolling a sum of k is the same as the probability of rolling a sum of $14 - k$.

4. Since$X = 1.5 \times Y$, the PMF of X is the same as the PMF of Y but with values scaled by 1.5 which is:

$$P(X = 1.5k) = P(Y = k), k \epsilon \{2, \dots, 12\}$$

Therefore, for the results, The PMF of X is given by:

$$P(X = x) = \begin{cases} \frac{Number\ of\ outcomes\ resulting\ in\ \frac{x}{1.5}}{36} \\ 0, \end{cases},$$

$$if\ x \epsilon \{3.0, 4.5, \dots, 18.0\}, otherwise.$$

B) Here, the condition *D* specifies that the first die roll Y_1 is 4. The goal is to find the conditional probability $P(X = k \mid D)$ where $X = 1.5 \times (Y_1 + Y_2)$.

Steps:

1. Given $Y_1 = 4$, the random variable Y_2 can take values in $\{1,2,3,4,5,6\}$. The sum $Y = Y_1 + Y_2$ now becomes:

$$Y = 4 + Y_2, Y \in \{5,6,7,8,9,10\}.$$

2. Under the condition *D*, the conditional probabilities for *Y* are uniform because Y_2 is still uniformly distributed:

$$P(Y = k \mid D) = \begin{cases} \frac{1}{6}, & if \in \{5,6,7,8,9,10\}, \\ 0, & otherwise \end{cases}$$

3. Since$X = 1.5 \times Y$, the conditional PMF for *X* is:

$$P(Y = k \mid D) = \begin{cases} \frac{1}{6}, & if \in \{7.5,9.0,10.5,12.0,13.5,15.0\}, \\ 0, & otherwise \end{cases}$$

Task 1 results:

A) **PMF of *X*:** Using the distribution derived in Part (A) for all possible values of *X* scaled by 1.5 from *Y*'s PMF.

B) **Conditional PMF $P(X = k \mid D)$:** Under the condition *D, X* has a uniform probability distribution over the scaled values corresponding to Y_2's range given $Y_1 = 4$.

Definitions

1. **Variogram $\gamma(h)$**: The variogram is defined as the expected value of the squared difference between two values of the random variable at two points that are separated by Lag ***h.*** This tool is a valuable tool for understanding the structure and relationships between values in a spatially or temporally distributed dataset. It helps to quantify how data points are related based on their spatial separation and is often used in geostatistical models, such as Kriging, to make predictions and assess uncertainty in spatial data. [1][2]

$$2\,\gamma\,(s_1, s_2) = \mathbb{E}\,[(Z(s_1) - Z(s_2))^2]$$

Where:

- $\gamma\,(s_1, s_2)$ is the **variogram** between locations $s_1\ and\ s_2$.
- $Z(s_1)\ and\ Z(s_2)$ are the random variables or the values of the process at locations $s_1\ and\ s_2$, respectively.
- $\mathbb{E}$ is the **expected value** (or mean), which is the statistical average of the squared difference between the values at locations $s_1\ and\ s_2$.

Note: The factor of 2 in the variogram formula arises because it represents the full squared difference between the values at two locations, without the normalization of the semi-variogram. The semi-variogram, by definition, includes a factor of $\frac{1}{2}$ to ensure that the starting value is zero when the points coincide. Doubling the semi-variogram gives the variogram. [2]

2. **Expectation or mean ($\mathbb{E}$):** The expectation or mean of a random variable X is the average value that X takes when you observe it over many trials. It gives a measure of the "central" value of the random variable. [4]

- For a discrete random variable X:

$$\mathbb{E}[X] = \sum_i P(x_i).x_i$$

where x_i are the possible outcomes of X, and $P(x_i)$ is the probability of each outcome.

- For a continuous random variable X, the expectation is defined as:

$$\mathbb{E}[X] = \int_{-\infty}^{+\infty} xf(x)dx$$

where $f(x)$ is the probability density function (PDF) of X.

The expectation gives the "long-run" average of the random variable. It's useful because it provides a single summary value that characterizes the center of the distribution of X. In the other expression, the average outcome we would expect if you repeated an experiment many times.

3. **Variance *VAR(X):*** In probability theory and statistics, variance is the expected value of the squared deviation from the mean of a random variable. Variance is a measure of dispersion, meaning it is a measure of how far a set of numbers is spread out from their average value. [1][2]

$$S^2 = \frac{\sum(x_i - \bar{x})^2}{n - 1}$$

Where:

- S^2 is sample variance
- x_i = the value of the one observation
- $\bar{x}$ = the mean value of all observations
- n = the number of observations

In terms of writing using the Expectation ($\mathbb{E}$) or mean, we would have: [1]

$$VAR\ (x) = \mathbb{E}[(X - \mathbb{E}[X])^2]$$

Since it is essential for understanding how predictable or uncertain a random variable is [1], we can use variance.

4. **Covariance *COV (X, Y)*:** Covariance in probability theory and statistics is a measure of the joint variability of two random variables. [2][3]

   The sign of the covariance, therefore, shows the tendency in the linear relationship between the variables. If greater values of one variable mainly correspond with greater values of the other variable, and the same holds for lesser values (that is, the variables tend to show similar behavior), the covariance is positive[2][3]

   In the opposite case, when greater values of one variable mainly correspond to lesser values of the other (that is, the variables tend to show opposite behavior), the covariance is negative.

$$Cov_{x,y} = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{N - 1}$$

Where:

- $Cov_{x,y}$ covariance between variable *x and y*
- x_i = the value of the data x
- $\bar{x}$ = mean of *x*
- $\bar{y}$ = mean of *y*
- *N* = the number of data values

$$COV(X, Y) = \mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])]$$

Covariance is useful for understanding how two variables are related. If we are analyzing two datasets (e.g., height and weight), covariance can tell if there is a general trend where larger heights correspond to larger weights (+COV) or smaller weight (-COV). [2][3]

Task 2:

In order to prove $\boldsymbol{\gamma(h) = COV(0) - COV(h)}$, we need to derive the relationship between the variogram *γ(h)* and the covariance. The key is to use the definition of the variogram and the properties of covariance.

Steps:

1. Let us expand the variogram formula in the order below:

$$\big(Z(s_1) - Z(s_2)\big)^2 = Z(s_1)^2 - 2Z(s_1)Z(s_2) + Z(s_2)^2$$

$$\rightarrow 2\gamma(h) = \mathbb{E}[Z(s_1)^2] - 2\mathbb{E}[Z(s_1)Z(s_2)] + \mathbb{E}[Z(s_2)]^2]$$

2. If the random field *Z* is stationary, we can assume:

   - $\mathbb{E}[Z(s_1)] = \mathbb{E}[Z(s_2)] = \mu$ $(the\ mean\ is\ constant)$,
   - The **variance** at any location is constant:
     - $VAR(Z(s)) = E[Z(s)^2] - \mu^2$
   - The covariance between locations only depends the distance of $h = |s_1 - s_2|$ so:

$$COV(Z(s_1), Z(s_2)) = COV(h)$$

3. Now applying the assumptions:
   - $\mathbb{E}[Z(s_1)^2] = VAR(Z(s_1)) + \mu^2$
   - $\mathbb{E}[Z(s_2)^2] = VAR\big(Z(s_2)\big) + \mu^2$
   - $\mathbb{E}[Z(s_1)Z(s_2)] \; = \; COV(Z(s_1), Z(s_2)) = COV(h)$

4. Now, substituting these into the variogram formula:

$$2\gamma(h) = \big(VAR\big(Z(s_1)\big) + \mu^2\big) + \big(VAR\big(Z(s_2)\big) + \mu^2\big) - 2.COV(h)$$

and since $VAR\big(Z(s_1)\big) = \; VAR\big(Z(s_2)\big)$, therefore we have:

$$2\gamma(h) = \; 2.VAR\big(Z(s)\big) + 2\mu^2 - 2\,.COV(h)$$

Now if we use fact 2 to get the equation simpler, we would have:

$$\gamma(h) = \; VAR\big(Z(s)\big) - COV(h)$$

5. Finally, since the covariance at lag *0* is simply the variance we have:

$$\gamma(h) = COV(0) - COV(h)$$

Task 3:

When we have 3 random variables, the variance of their sum is given by the following general formula:

$$VAR\,(X_1 + X_2 + X_3) = VAR\,(X_1) + VAR\,(X_2) + VAR\,(X_3) + 2\,.\,(\,COV(X_1, X_2) + COV\,(X_2, X_3) + COV\,(X_1, X_3))$$

Proof:

According to the Definitions section for Variance, Covariance, Expectations and Variogram, We can now apply the formula for the variance of the sum of random variables to $X_1 + X_2 + X_3$.

We can use combination the variance and variogram formulas like this:

$$VAR\,(X_1 + X_2 + X_3) = \mathbb{E}[(X_1 + X_2 + X_3) - (X_1 + X_2 + X_3)^2]$$

$$With\ expansion \rightarrow \mathbb{E}\,[X_1 + X_2 + X_3] = \mathbb{E}[X_1] + \mathbb{E}[X_2] + \mathbb{E}[X_3]$$

$$square\ expansion \rightarrow \mathbb{E}[(X_1 + X_2 + X_3) - (\mathbb{E}[X_1] + \mathbb{E}[X_2] + \mathbb{E}[X_3])^2 = (X_1 - \mathbb{E}[X_1] + X_2 - \mathbb{E}[X_2] + X_3 - \mathbb{E}[X_3])^2$$

With the expansion of the sum of the squared:

$$(X_1 - \mathbb{E}[X_1])^2 + (X_2 - \mathbb{E}[X_2])^2 + (X_3 - \mathbb{E}[X_3])^2 + 2.((X_1 - \mathbb{E}[X_1])(X_2 - \mathbb{E}[X_2]) + (X_2 - \mathbb{E}[X_2])(X_3 - \mathbb{E}[X_3]) + (X_1 - \mathbb{E}[X_1])(X_3 - \mathbb{E}[X_3]))$$

Putting everything together we reach this expression:

$$VAR\ (X_1 + X_2 + X_3) = VAR\ (X_1) + VAR\ (X_2) + VAR\ (X_3)$$

This means the variance of the sum is just the sum of the individual variances.

Task 4

- Importing modules and reading the file and setting up columns

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.spatial.distance import pdist, squareform
from sklearn.preprocessing import StandardScaler

file_path = 'data.xlsx'
data = pd.read_excel(file_path)
series1 = data['Data Series 1'].values
series2 = data['Data Series 2'].values
```

Explanation:

The code begins by importing essential libraries for data processing and visualization. It uses **pandas** to handle the data in an Excel file, **numpy** for numerical calculations like mean, variance, and standard deviation, and **matplotlib** for plotting graphs. Additionally, it imports functions from **scipy** for calculating distances, which will be helpful in analyzing the variograms, and **StandardScaler** from **sklearn** to normalize data, if needed.

The code then reads data from an Excel file using **pandas** and extracts two columns labeled "Data Series 1" and "Data Series 2" as separate variables. These variables represent the two-time series or datasets to be analyzed. The series are extracted as arrays to enable efficient mathematical and statistical operations. After this, the code proceeds to analyze these series, calculating various statistical metrics and visualizing the results.

- Basic statistics such as Mean, Median, Variance, Standard Deviation, IQR, Min and Max of both series as well as the covariance and the coefficient of correlation.

```
def basic_statistics(series):
    mean = np.mean(series)
    median = np.median(series)
    variance = np.var(series)
    std_dev = np.std(series)
    iqr = np.percentile(series, 75) - np.percentile(series, 25)
    min_value = np.min(series)
    max_value = np.max(series)
    return mean, median, variance, std_dev, iqr, min_value, max_value

mean1, median1, variance1, std_dev1, iqr1, min1, max1 = basic_statistics(series1)
mean2, median2, variance2, std_dev2, iqr2, min2, max2 = basic_statistics(series2)

covariance = np.cov(series1, series2)[0, 1]
correlation = np.corrcoef(seri     es1, series2)[0, 1]

print("Statistics for Series 1:")
print(f"Mean: {mean1}, Median: {median1}, Variance: {variance1}, Std Dev:
{std_dev1}, IQR: {iqr1}, Min: {min1}, Max: {max1}")
print("Statistics for Series 2:")
print(f"Mean: {mean2}, Median: {median2}, Variance: {variance2}, Std Dev:
{std_dev2}, IQR: {iqr2}, Min: {min2}, Max: {max2}")
print(f"Covariance between Series 1 and Series 2: {covariance}")
print(f"Correlation between Series 1 and Series 2: {correlation}")
```

Explanation:

The code defines a function that calculates several key statistical measures for a given dataset. These include the mean (average), median (middle value), variance (spread of data), standard deviation (a measure of the average distance from the mean), interquartile range (the range between the 25th and 75th percentiles), and the minimum and maximum values. This function is applied to two datasets, calculating these statistics for each separately.

Additionally, the code computes two important measures of the relationship between the two datasets: covariance and correlation. Covariance indicates how two datasets vary together, while correlation quantifies the strength and direction of their linear relationship. Finally, the calculated statistics for both datasets are printed, summarizing their characteristics and how they are related to each other.

Results:

Series 1:

- **Mean (1.55):** The average value of Series 1 is approximately 1.55. This indicates that, on average, the values in Series 1 tend to hover around this value.
- **Median (1.26):** The median is the middle value when the data is sorted. Since the mean is higher than the median, it suggests that Series 1 may have a right-skewed distribution (more lower values and a few higher outliers).
- **Variance (1.44) and Standard Deviation (1.20):** These values measure how spread out the data is from the mean. A variance of 1.44

and a standard deviation of 1.20 suggest that the values in Series 1 have a moderate spread, with some variability around the mean.

- **Interquartile Range (IQR = 1.60):** The IQR, which represents the range between the 25th and 75th percentiles, is quite large. This indicates that the middle 50% of the data has a broad spread, with a significant range between the lower and upper quartiles.
- **Min (0.015) and Max (6.02):** The minimum value in Series 1 is very close to zero, and the maximum value is 6.02. This suggests that the data spans a relatively wide range, from low values near 0 to higher values near 6.

Series 2:

- **Mean (1.44):** The average value of Series 2 is slightly lower than Series 1's mean, indicating that, on average, the values in Series 2 are a little smaller.
- **Median (1.27):** Similar to Series 1, the median is very close to the mean, indicating a relatively symmetric distribution. However, the difference between the mean and median is slightly smaller than in Series 1, suggesting less skewness.
- **Variance (1.46) and Standard Deviation (1.21):** The variance and standard deviation are similar to those of Series 1, indicating similar spread and variability around the mean.
- **Interquartile Range (IQR = 2.33):** The IQR is significantly larger than that of Series 1, suggesting that the middle 50% of the data in Series 2 has a much wider spread, with more variability between the 25th and 75th percentiles.

- **Min (0.00) and Max (4.23):** The minimum value in Series 2 is 0, and the maximum value is 4.23. This shows that Series 2 has a smaller range than Series 1, with values ranging from 0 to around 4.23.

Covariance (0.29):

The covariance between Series 1 and Series 2 is 0.29, indicating a small positive relationship between the two datasets. As one series increases, the other tends to increase as well, but the relationship is not very strong. A positive covariance suggests that both series are somewhat aligned in their behavior, though the relationship is not extremely pronounced.

Correlation (0.19):

The correlation coefficient of 0.19 indicates a very weak positive linear relationship between Series 1 and Series 2. A correlation close to 0 suggests that, while there is some tendency for the two series to move in the same direction, the relationship is weak, and the variation in one series does not strongly predict the variation in the other series.

- A variogram for each data series

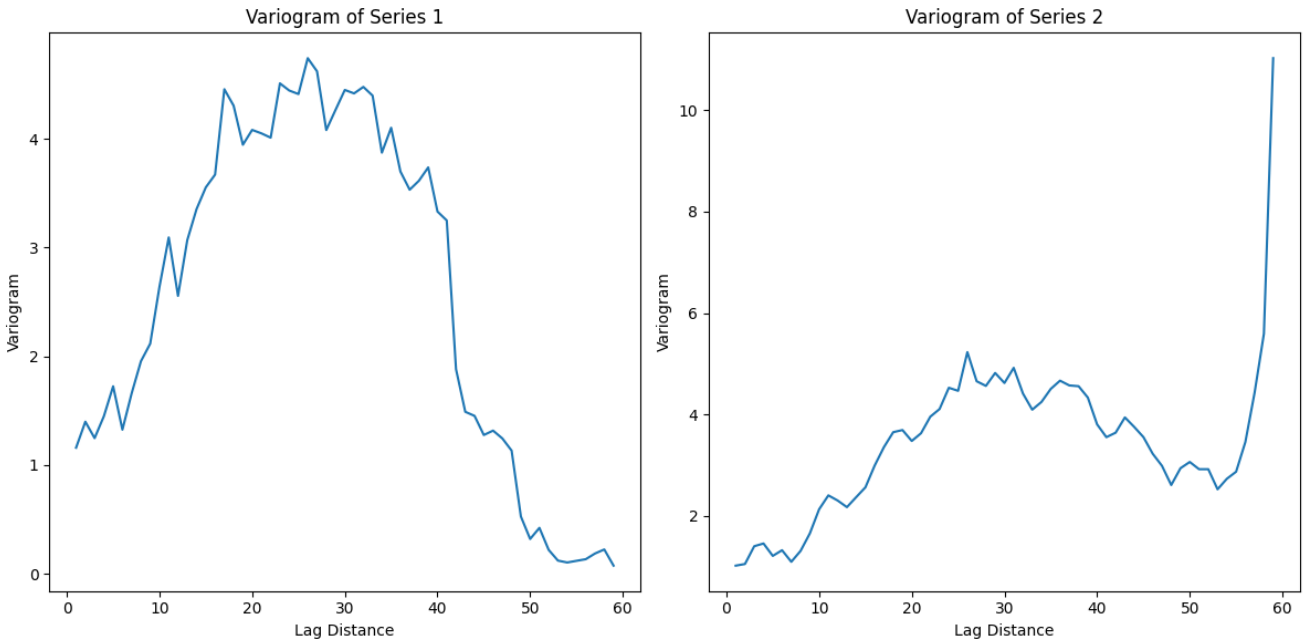
```
def variogram(series):
    n = len(series)
    variogram_values = []
    lags = []

    for lag in range(1, n):
        differences = series[:-lag] - series[lag:]
        squared_diff = differences**2
        variogram_values.append(np.mean(squared_diff))
        lags.append(lag)

    return np.array(lags), np.array(variogram_values)

distances1, variogram1 = variogram(series1)
distances2, variogram2 = variogram(series2)

# Plot the variogram
plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 1)
plt.plot(distances1, variogram1, label="Series 1 Variogram")
plt.xlabel('Lag Distance')
plt.ylabel('Variogram')
plt.title('Variogram of Series 1')

plt.subplot(1, 2, 2)
plt.plot(distances2, variogram2, label="Series 2 Variogram")
plt.xlabel('Lag Distance')
plt.ylabel('Variogram')
plt.title('Variogram of Series 2')
```

Explanation:

The code defines a process for calculating and visualizing the **variogram** of two datasets, Series 1 and Series 2. A variogram is a tool used in spatial analysis to measure the degree of similarity between values at different distances (lags) in a dataset. It helps to understand how the data points change as the distance between them increases.

The function calculates how much the values of a data series differ from one another as the distance between them (referred to as the "lag") increases. For each lag, it computes the squared difference between values that are separated by that lag, averages these squared differences, and then stores them as variogram values. These values indicate how the data behaves over different distances. The code then generates two variograms: one for Series 1 and one for Series 2.

After calculating the variogram values, the code generates two plots side by side. The first plot shows the variogram for Series 1, and the second one shows the variogram for Series 2. These plots visually demonstrate how the data in each series varies with increasing distance between data points. In both plots, the x-axis represents the lag distance (how far apart the data points are), and the y-axis shows the variogram values, representing the average squared differences at each distance.



- A Cross Variogramm between data series 1 and 2

```
def cross_variogram(series1, series2):
    n = len(series1)
    cross_variogram_values = []
    lags = []
    for lag in range(1, n):
        differences = (series1[:-lag] - series2[lag:])
        squared_diff = differences**2
        cross_variogram_values.append(np.mean(squared_diff))
        lags.append(lag)
    return np.array(lags), np.array(cross_variogram_values)

cross_distances, cross_variogram_values = cross_variogram(series1, series2)

plt.figure(figsize=(8, 6))
plt.plot(cross_distances, cross_variogram_values, label="Cross Variogram")
plt.xlabel('Lag Distance')
plt.ylabel('Cross Variogram')
plt.title('Cross Variogram Between Series 1 and Series 2')
plt.legend()
plt.show()
```

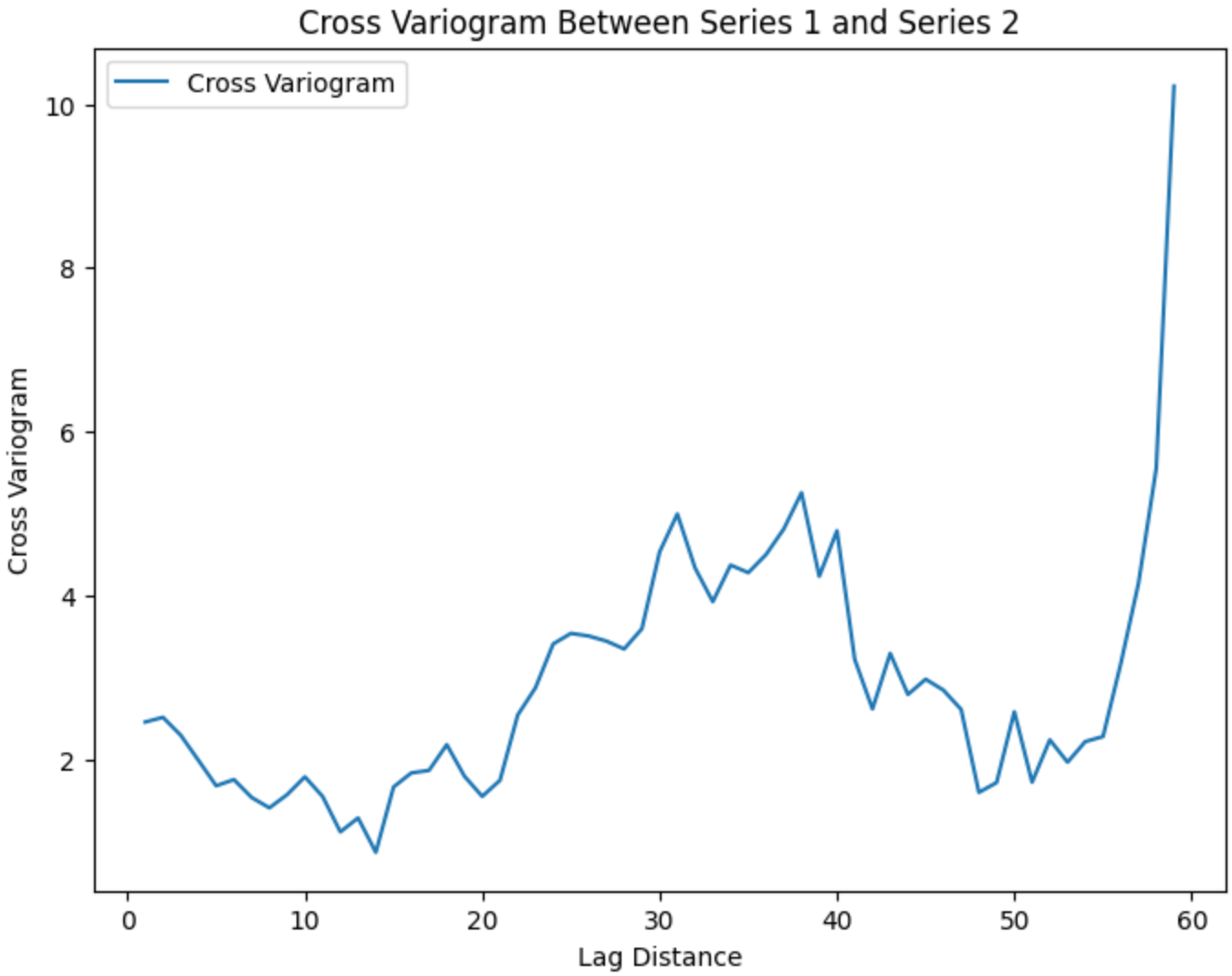
Explanation:

The code defines a process for calculating and visualizing the **cross-variogram** between two datasets, Series 1 and Series 2. A cross-variogram is similar to a variogram, but instead of measuring the differences within a single series, it measures the relationship between two different series at varying distances (lags). It helps to understand how the two datasets relate to each other spatially at different distances.

The function calculates the squared differences between values from **Series 1** and **Series 2**, with the values in **Series 1** being shifted by a lag (distance). For each lag, it calculates how much the values of the two series differ from one another as the lag increases. These squared differences are then averaged, and the result is stored as the cross-variogram value for that specific lag. This is repeated for all lag distances, giving a measure of how the two datasets are related at each distance.

After calculating the cross-variogram values, the code generates a plot that shows how the cross-variogram behaves over different lags. The x-axis represents the lag distance (the difference between the indices of the values being compared), and the y-axis represents the cross-variogram values (the squared differences between the values at each lag). A high value at a specific lag indicates that the two series are very different at that distance, while lower values suggest a stronger similarity between the two series at that lag.

This plot provides insight into how the two series interact at different distances. If the cross-variogram has a steep increase or large values at certain lags, it suggests a strong difference between the series at those distances. Conversely, a flatter or lower cross-variogram suggests that the two series are more similar at those lags.



- Cross Variogram

```
offset = cross_distances[np.argmin(cross_variogram_values)]
print(f"Suggested offset based on the cross-variogram: {offset} meters")
```

Results

```
Suggested offset based on the cross-variogram: 14 meters
```

The suggested offset between the two series based on the cross-variogram is 14 meters, indicating that shifting one series by this distance will best align it with the other. This offset represents the lag where the two series show the closest similarity in their values.

References

[1]. Prof. K. G. van den Boogaart (2024). *Statistical Analysis of Systems***.** MDRS course,

[2]. Wikipedia. Wikipedia,

[3]. Rice, John (2007). *Mathematical Statistics and Data Analysis*. Brooks/Cole Cengage Learning. p. 138. ISBN 9780534399429,

[4]. Wasserman, Larry (December 2010). *All of Statistics: a concise course in statistical inference*. Springer texts in statistics. p. 47. ISBN 9781441923226,