Historical Background and Overview

Kendall's τ measures the ordinal association between two variables. It assesses the similarity of the orderings (ranks) of data points. A τ test is a non-parametric hypothesis test for statistical dependence based on τ. Developed by Maurice Kendall in 1938, based on a similar concept proposed by Gustav Fechner in 1897. It is a type of rank correlation.

$$\tau_b = \frac{C - D}{\sqrt{(C + D + T_x)(C + D + T_y)}}$$

Where:

C is the number of concordant pairs. D is the number of discordant pairs. T_x and T_y are the number of tied ranks in each variable.

Kendall correlation is high when the rankings of observations are similar (or identical for a perfect correlation of 1).

Kendall correlation is low when the rankings are dissimilar (or perfectly opposite for a correlation of -1).

Kendall's τ and Spearman's ρ are special cases of a more general correlation coefficient.

Concepts of concordance and discordance (matching or differing pairs of ranks) appear in other statistical methods.

One such method is the Rand index used in cluster analysis.

Concordant Pair and Discordant Pair in Kendall's Tau

Concordant Pair

A **concordant pair** is a pair of observations where the order (rank) of both variables is consistent, meaning the two variables agree in terms of relative ordering.

Condition for Concordance:

If you have two pairs of ranks ((x_1, y_1)) and ((x_2, y_2)), they are concordant if:

$$x_1 < x_2 \text{ and } y_1 < y_2 \quad \text{or} \quad x_1 > x_2 \text{ and } y_1 > y_2$$

In other words, the relative order (greater or lesser) of both variables matches for both pairs.

Example of a Concordant Pair:

Pair 1: ((x_1 = 1, y_1 = 3))
Pair 2: ((x_2 = 2, y_2 = 4))

Here, (x_1 < x_2) and (y_1 < y_2), so these two pairs are **concordant**.

Discordant Pair

A **discordant pair** is a pair of observations where the order (rank) of the two variables is inconsistent, meaning the two variables disagree in terms of relative ordering.

Condition for Discordance:

If you have two pairs of ranks ((x_1, y_1)) and ((x_2, y_2)), they are discordant if:

$$x_1 < x_2 \text{ and } y_1 > y_2 \quad \text{or} \quad x_1 > x_2 \text{ and } y_1 < y_2$$

In other words, the relative order of the two variables is opposite for the two pairs.

Example of a Discordant Pair:

Pair 1: ((x_1 = 1, y_1 = 3))
Pair 2: ((x_2 = 2, y_2 = 1))

Here, (x_1 < x_2), but (y_1 > y_2), so these two pairs are **discordant**.

Summary:

- **Concordant pairs**: The relative ranking of both variables agrees (both increase or both decrease together).
- **Discordant pairs**: The relative ranking of both variables is opposite (one increases while the other decreases).

Binomial Coefficient Calculations

A binomial coefficient, denoted as C(n, k) or often written as "n choose k," represents the number of ways to choose k items from a set of n distinct items. It's a fundamental concept in combinatorics and probability.

C(n, k) = n! / (k! * (n-k)!)

where:

n! is the factorial of n (n * (n-1) * (n-2) * ... * 1)

k! is the factorial of k

(n-k)! is the factorial of n-k

The notation

$$\binom{n}{2}$$

refers to the binomial coefficient, which calculates the number of ways to choose 2 items from a set of

$$n$$

items. It is also known as "n choose 2."

The formula for the binomial coefficient is:

$$\binom{n}{2} = \frac{n(n-1)}{2}$$

In your case,

$$n = 4$$

, so we calculate:

$$\binom{4}{2} = \frac{4(4-1)}{2} = \frac{4 \times 3}{2} = \frac{12}{2} = 6$$

Numerical Examples with simple Data

Consider the following data points for two variables (X) and (Y):

(X)	(Y)
1	3
2	1
3	2
4	4

Step 1: List All Pairs

We have 4 data points, so the total number of pairs is

$$(\binom{4}{2} = 6)$$

. The pairs we are considering are:

1. ((X_1 = 1, Y_1 = 3)) and ((X_2 = 2, Y_2 = 1))
2. ((X_1 = 1, Y_1 = 3)) and ((X_3 = 3, Y_3 = 2))
3. ((X_1 = 1, Y_1 = 3)) and ((X_4 = 4, Y_4 = 4))
4. ((X_2 = 2, Y_2 = 1)) and ((X_3 = 3, Y_3 = 2))
5. ((X_2 = 2, Y_2 = 1)) and ((X_4 = 4, Y_4 = 4))
6. ((X_3 = 3, Y_3 = 2)) and ((X_4 = 4, Y_4 = 4))

Step 2: Classify the Pairs

We now classify the pairs as concordant or discordant.

- **Pair 1**: ((X_1 = 1, Y_1 = 3)) and ((X_2 = 2, Y_2 = 1))

  (X_1 < X_2) but (Y_1 > Y_2), so this is **discordant**.

- **Pair 2**: ((X_1 = 1, Y_1 = 3)) and ((X_3 = 3, Y_3 = 2))

(X_1 < X_3) and (Y_1 > Y_3), so this is **discordant**.

- **Pair 3**: ((X_1 = 1, Y_1 = 3)) and ((X_4 = 4, Y_4 = 4))

  (X_1 < X_4) and (Y_1 < Y_4), so this is **concordant**.

- **Pair 4**: ((X_2 = 2, Y_2 = 1)) and ((X_3 = 3, Y_3 = 2))

  (X_2 < X_3) and (Y_2 < Y_3), so this is **concordant**.

- **Pair 5**: ((X_2 = 2, Y_2 = 1)) and ((X_4 = 4, Y_4 = 4))

  (X_2 < X_4) and (Y_2 < Y_4), so this is **concordant**.

- **Pair 6**: ((X_3 = 3, Y_3 = 2)) and ((X_4 = 4, Y_4 = 4))

  (X_3 < X_4) and (Y_3 < Y_4), so this is **concordant**.

Step 3: Count Concordant and Discordant Pairs

- Concordant pairs: 4 (pairs 3, 4, 5, 6)
- Discordant pairs: 2 (pairs 1, 2)

Step 4: Calculate Kendall's Tau

Now, we can calculate Kendall's Tau using the formula:

$$\tau = \frac{(\text{number of concordant pairs}) - (\text{number of discordant pairs})}{\binom{n}{2}} = 1 - \frac{2 \times (\text{number of}}{(}$$

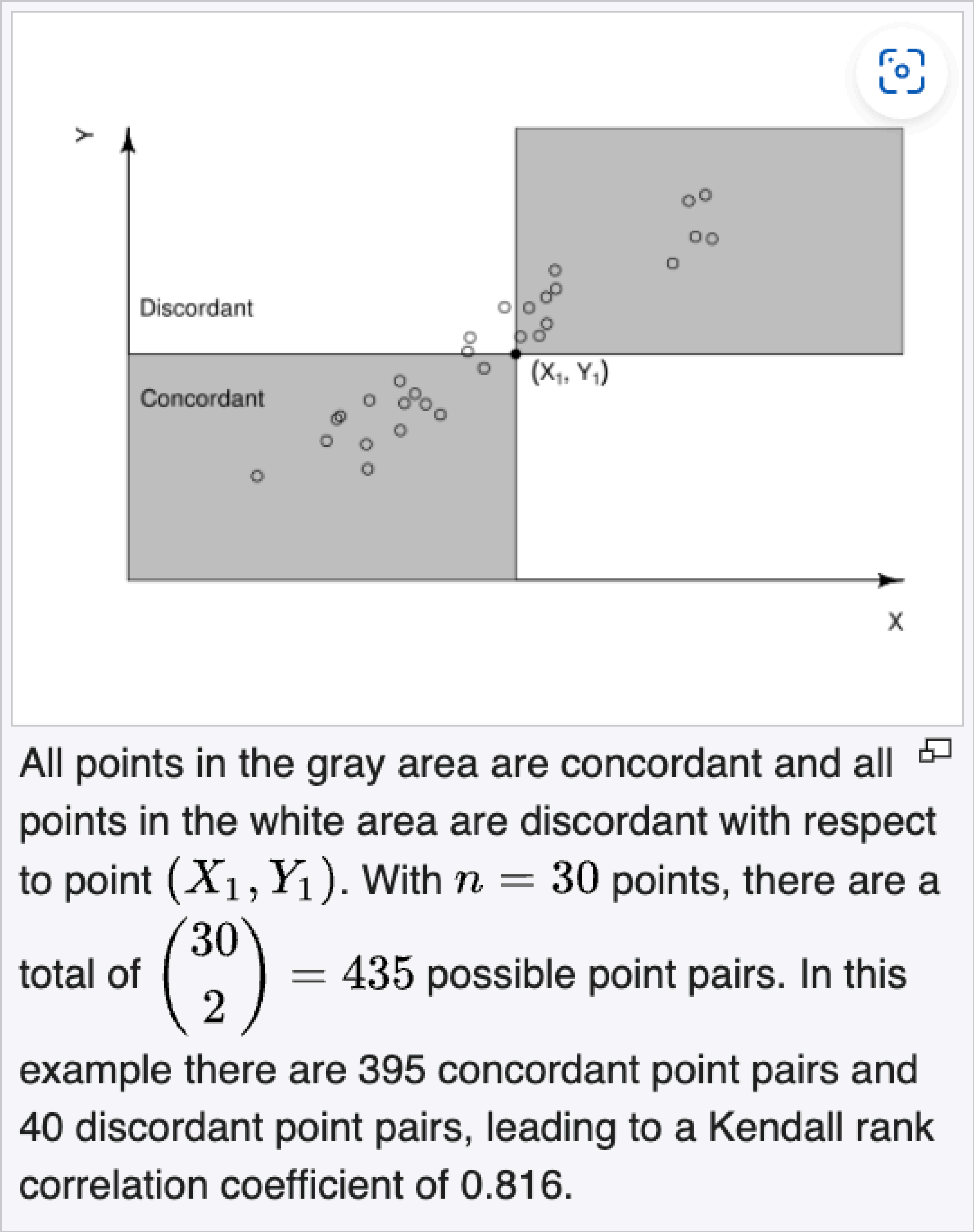
Where (n = 4), so the total number of pairs is (\binom{4}{2} = 6).

Substitute the values:

$$\tau = \frac{4 - 2}{6} = \frac{2}{6} = 0.33$$

Final Answer:

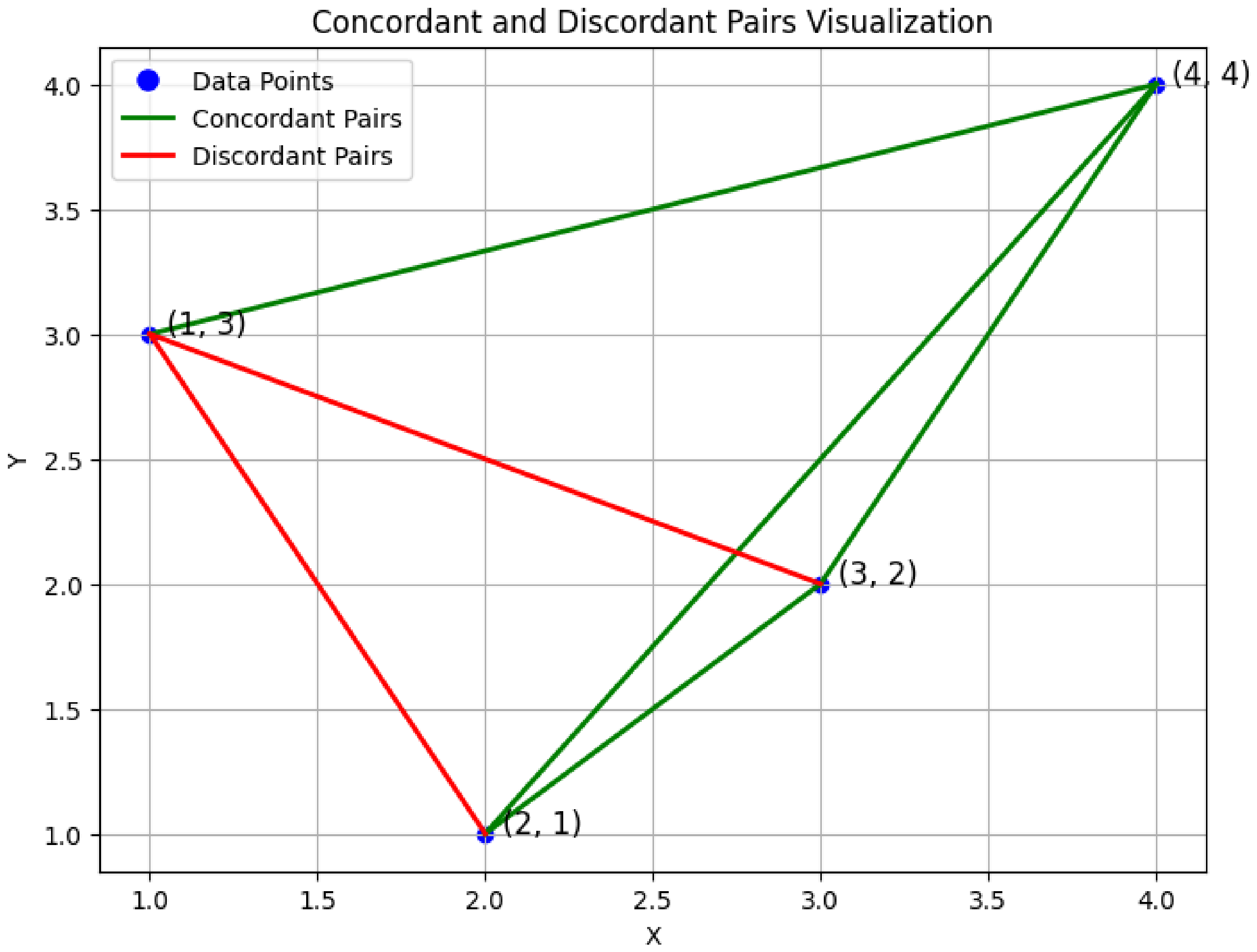
The Kendall's Tau rank correlation coefficient for this dataset is (0.33), indicating a moderate positive correlation between the variables.



All points in the gray area are concordant and all points in the white area are discordant with respect to point (X_1, Y_1). With $n = 30$ points, there are a total of $\binom{30}{2} = 435$ possible point pairs. In this example there are 395 concordant point pairs and 40 discordant point pairs, leading to a Kendall rank correlation coefficient of 0.816.

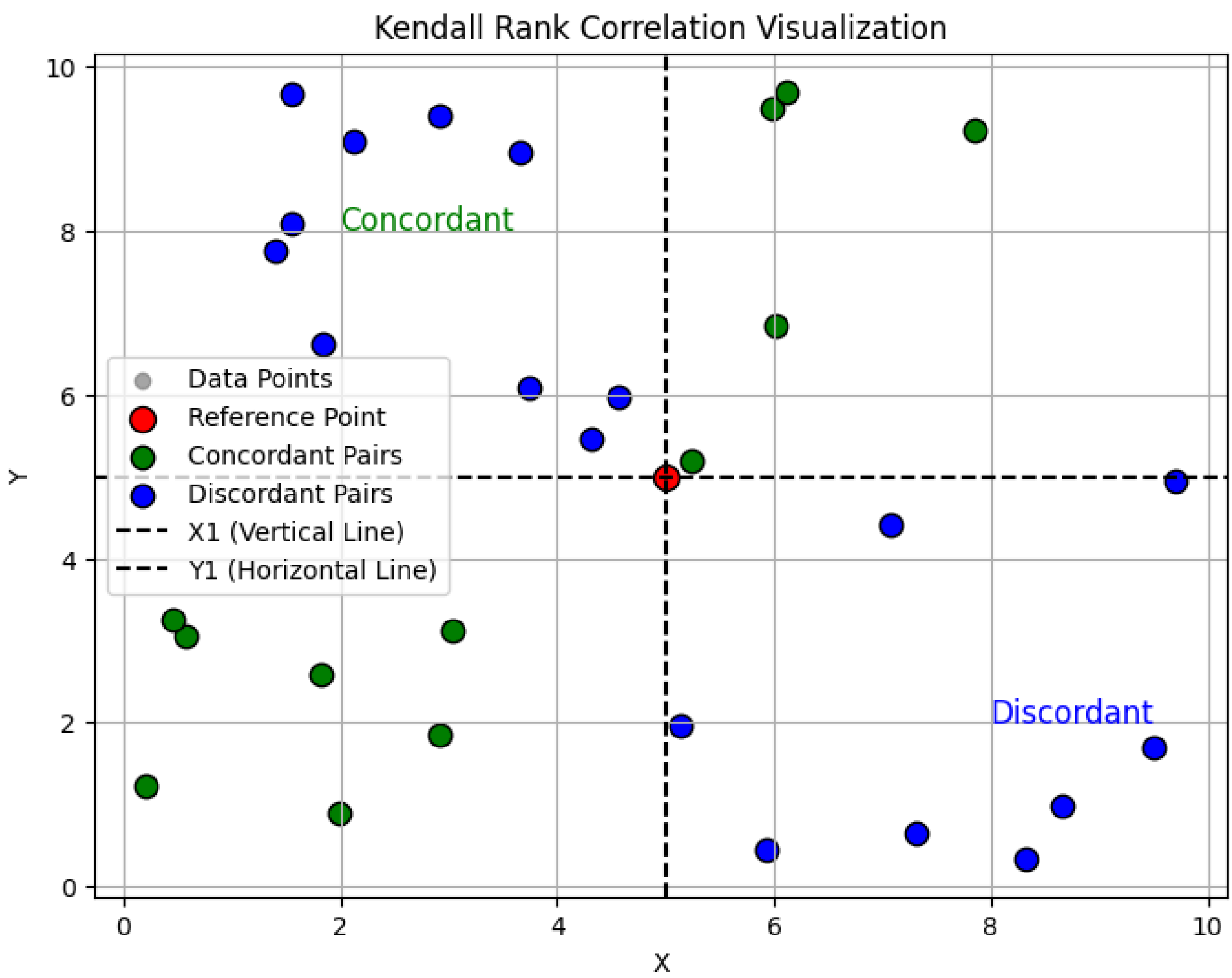
```
import matplotlib.pyplot as plt
import itertools
import numpy as np

# Data points for X and Y
X = [1, 2, 3, 4]
Y = [3, 1, 2, 4]

# List all pairs of points (combinations of 2 elements)
pairs = list(itertools.combinations(range(len(X)), 2))

# Classify pairs (concordant or discordant)
concordant_pairs = []
discordant_pairs = []
```

```
for (i, j) in pairs:
    if (X[i] < X[j] and Y[i] < Y[j]) or (X[i] > X[j] and Y[i] > Y[j]):
        concordant_pairs.append((i, j))
    else:
        discordant_pairs.append((i, j))

# Plotting the data points
plt.figure(figsize=(8, 6))
plt.scatter(X, Y, color='blue', label='Data Points')

# Highlight concordant pairs in green
for (i, j) in concordant_pairs:
    plt.plot([X[i], X[j]], [Y[i], Y[j]], color='green', lw=2)

# Highlight discordant pairs in red
for (i, j) in discordant_pairs:
    plt.plot([X[i], X[j]], [Y[i], Y[j]], color='red', lw=2)

# Add labels for points
for i in range(len(X)):
    plt.text(X[i] + 0.05, Y[i], f'({X[i]}, {Y[i]})', fontsize=12)

# Labels and title
plt.xlabel('X')
plt.ylabel('Y')
plt.title("Concordant and Discordant Pairs Visualization")

# Add legend
plt.legend(
    handles=[
        plt.Line2D([0], [0], marker='o', color='w', markerfacecolor='blue', marke
        plt.Line2D([0], [0], color='green', lw=2, label='Concordant Pairs'),
        plt.Line2D([0], [0], color='red', lw=2, label='Discordant Pairs')
    ],
    loc='best'
)

# Show the plot
plt.grid(True)
plt.show()
```



```
import matplotlib.pyplot as plt
import numpy as np

# Reference point (X1, Y1)
x1, y1 = 5, 5

# Generate random data points
np.random.seed(42)
x = np.random.uniform(0, 10, 30)
y = np.random.uniform(0, 10, 30)

# Create the scatter plot
plt.figure(figsize=(8, 6))
plt.scatter(x, y, color='gray', alpha=0.7, label='Data Points')

# Plot the reference point in a more prominent way
plt.scatter(x1, y1, color='red', s=100, edgecolors='black', label='Reference Poin

# Calculate Concordant and Discordant pairs
concordant_x = []
concordant_y = []
discordant_x = []
discordant_y = []
```

```
for i in range(len(x)):
    if (x[i] > x1 and y[i] > y1) or (x[i] < x1 and y[i] < y1):
        concordant_x.append(x[i])
        concordant_y.append(y[i])
    else:
        discordant_x.append(x[i])
        discordant_y.append(y[i])

# Plot the concordant points
plt.scatter(concordant_x, concordant_y, color='green', label='Concordant Pairs',

# Plot the discordant points
plt.scatter(discordant_x, discordant_y, color='blue', label='Discordant Pairs', s

# Plot the axes for reference
plt.axvline(x1, color='black', linestyle='--', label='X1 (Vertical Line)')
plt.axhline(y1, color='black', linestyle='--', label='Y1 (Horizontal Line)')

# Add annotations for regions
plt.text(2, 8, 'Concordant', fontsize=12, color='green')
plt.text(8, 2, 'Discordant', fontsize=12, color='blue')

# Add a title and labels
plt.xlabel('X')
plt.ylabel('Y')
plt.title('Kendall Rank Correlation Visualization')

# Add legend
plt.legend()

# Display the grid for better readability
plt.grid(True)

# Show the plot
plt.show()
```



Kendall's Tau Calculation

Formula for Kendall's Tau (τ)

Kendall's Tau (τ) is given by the formula:

$$\tau = \frac{2}{n(n-1)} \sum_{i<j} \operatorname{sgn}(x_i - x_j) \cdot \operatorname{sgn}(y_i - y_j)$$

Where:

- (n) is the number of data points.
- (x_i, x_j) and (y_i, y_j) are the data points in the two variables.
- sgn(z) is the sign function, defined as:

$$\operatorname{sgn}(z) = \begin{cases} 1 & \text{if } z > 0 \\ 0 & \text{if } z = 0 \\ -1 & \text{if } z < 0 \end{cases}$$

Properties of Kendall's Tau

- **Range**: The value of Kendall's Tau (\tau) is always between -1 and 1:

$$-1 \le \tau \le 1$$

- **Perfect Agreement**: If the rankings are identical, then ($\tau = 1$).
- **Perfect Disagreement**: If the rankings are completely reversed, then ($\tau = -1$).
- **No Correlation**: If (X) and (Y) are independent, ($E[\tau] = 0$).

```
import numpy as np
from scipy.stats import kendalltau

# Sample data
# X = np.array([3, 1, 4, 5])
# Y = np.array([2, 4, 1, 3])

X = np.array([1, 2, 3, 4])
Y = np.array([3, 1, 2, 4])

# Compute Tau-a and Tau-b using scipy's kendalltau function
tau_a, _ = kendalltau(X, Y, method='asymptotic')  # Tau-a
tau_b, _ = kendalltau(X, Y, method='exact')  # Tau-b

# Compute Tau-c using scipy.stats
def kendall_tau_c(X, Y):
    """
    Function to compute Tau-c using scipy built-in functions for concordant and d
    """
    n = len(X)
    n0 = n * (n - 1) / 2

    # Compute concordant and discordant pairs using kendalltau
    tau_c, _ = kendalltau(X, Y)  # Use kendalltau directly for Tau-c

    return tau_c

# Compute Tau-c using scipy's kendalltau function
tau_c = kendall_tau_c(X, Y)

# Output the results
print(f"Kendall's Tau-a: {tau_a}")
print(f"Kendall's Tau-b: {tau_b}")
print(f"Kendall's Tau-c: {tau_c}")
```

```
Kendall's Tau-a: 0.3333333333333334
Kendall's Tau-b: 0.3333333333333334
Kendall's Tau-c: 0.3333333333333334
```

Tau-a, Tau-b, and Tau-c: Kendall's Rank Correlation Coefficients

Kendall's tau coefficients are a family of non-parametric statistics used to measure the association between two ranked variables. They are particularly useful when the data is not normally distributed or when the relationship between the variables is not linear.

1. Tau-a:

Simple Calculation: It's calculated by subtracting the number of discordant pairs from the number of concordant pairs, and then dividing by the total number of pairs. Limitation: It doesn't account for ties in the data.

2. Tau-b:

Accounts for Ties: It adjusts for ties in the data by considering the number of pairs tied on one or both variables.
Preferred: It's generally preferred over Tau-a as it provides a more accurate measure of association, especially when there are many ties.

3. Tau-c:

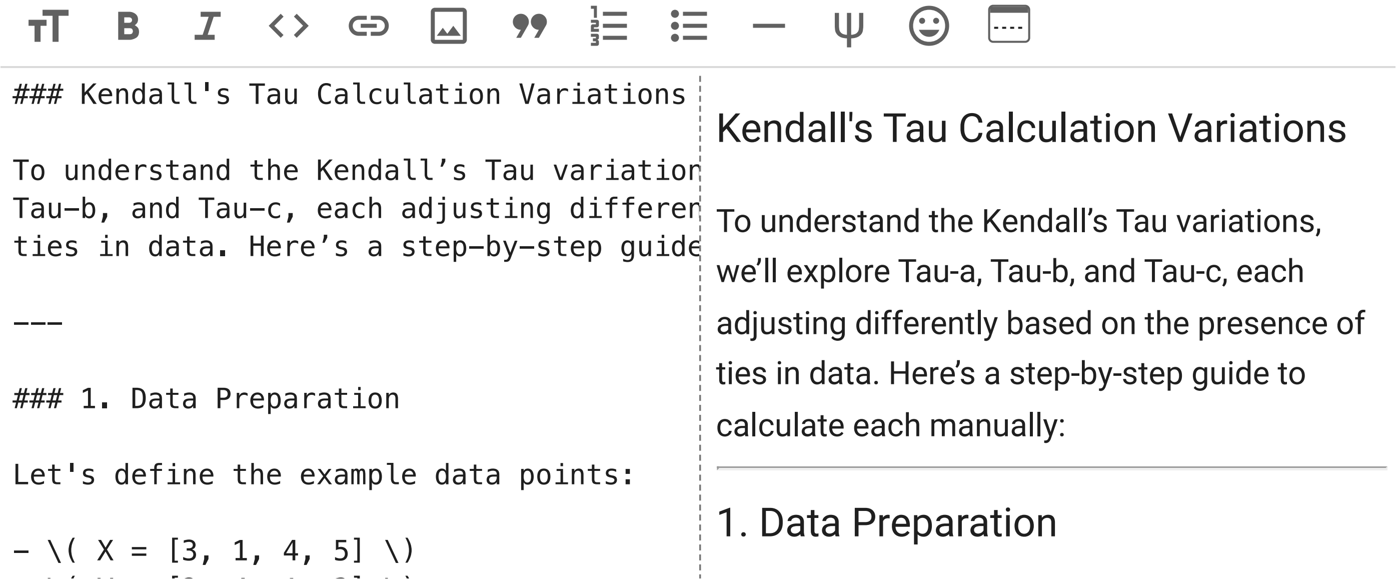
Suitable for Non-Square Tables: It's designed for situations where the two variables have different numbers of categories or levels.
Adjustment for Table Size: It adjusts for the size of the contingency table, making it more appropriate for non-square tables.

When to Use Which?

Tau-a: Use it when the data has no ties or when the number of ties is very small.

Tau-b: Use it when there are ties in the data, especially when the number of ties is significant.

Tau-c: Use it when the two variables have different numbers of categories or levels.



```
### Kendall's Tau Calculation Variations

To understand the Kendall’s Tau variatior
Tau-b, and Tau-c, each adjusting differer
ties in data. Here’s a step-by-step guide

---

### 1. Data Preparation

Let's define the example data points:

- \( X = [3, 1, 4, 5] \)
```

Kendall's Tau Calculation Variations

To understand the Kendall’s Tau variations, we’ll explore Tau-a, Tau-b, and Tau-c, each adjusting differently based on the presence of ties in data. Here’s a step-by-step guide to calculate each manually:

---

1. Data Preparation

```
- \( Y = [2, 4, 1, 3] \)

---

### 2. Pairwise Comparisons

To calculate Kendall's Tau, start by finc
comparisons** to determine **concordant**
For each pair:

- A **concordant pair** means that if one
than another, the corresponding value in
- A **discordant pair** does not satisfy

---

### 3. Kendall's Tau-a (τA)

Tau-a does not account for ties in the da

$$
\tau_A = \frac{n_c - n_d}{n_0}
$$

Where:

- \( n_c \): Number of concordant pairs
- \( n_d \): Number of discordant pairs
- $$ n_0 = \frac{n(n-1)}{2} $$:
Total number of pairwise comparisons, wit
number of data points

---

### 4. Kendall's Tau-b (τB)

Use Tau-b if there are ties in either \(
adjusts for ties by modifying the denomir

$$
\tau_B = \frac{n_c - n_d}{\sqrt{(n_0 - n_
$$

Where:

- \( n_c \): Number of concordant pairs
- \( n_d \): Number of discordant pairs
- \( n_0 = \frac{n(n-1)}{2} \): Total num
- \( n_1 = \sum_i t_i (t_i - 1) / 2 \): S
- \( n_2 = \sum_j u_j (u_j - 1) / 2 \): S
  - \( t_i \): Number of ties in the \( i
  - \( u_j \): Number of ties in the \( j

This formula accounts for the effect of t
more accurate estimate in their presence.
```

Let's define the example data points:

- (X = [3, 1, 4, 5])
- (Y = [2, 4, 1, 3])

---

2. Pairwise Comparisons

To calculate Kendall's Tau, start by finding all **pairwise comparisons** to determine **concordant** and **discordant** pairs. For each pair:

- A **concordant pair** means that if one value in (X) is greater than another, the corresponding value in (Y) is also greater.
- A **discordant pair** does not satisfy this condition.

---

3. Kendall's Tau-a (τA)

Tau-a does not account for ties in the data. The formula for Tau-a is:

$$\tau_A = \frac{n_c - n_d}{n_0}$$

Where:

- (n_c): Number of concordant pairs
- (n_d): Number of discordant pairs
- $$n_0 = \frac{n(n-1)}{2}$$: Total number of pairwise comparisons, with (n) being the total number of data points

---

4. Kendall's Tau-b (τB)

Use Tau-b if there are ties in either (X) or (Y). The formula adjusts for ties by modifying the denominator:

$$\tau_B = \frac{n_c - n_d}{\sqrt{(n_0 - n_1)(n_0 - n_2)}}$$

Where:

- (n_c): Number of concordant pairs

```
---
#### 5. Kendall Tau-c Coefficient

The **Kendall Tau-c** coefficient is defi

$$
\tau_C = \frac{2 (n_c - n_d)}{n^2 \cdot \
\cdot \frac{n - 1}{n} \cdot \frac{m}{m - 
$$

where:

- \( n_c \): Number of concordant pairs
- \( n_d \): Number of discordant pairs
- \( r \): Number of rows
- \( c \): Number of columns
-  m = min(r, c) : The minimum of the num


Tau-c adjusts for different numbers of ro
suitable for non-square tables.

---

### Example Calculation

Given the data points \( X = [3, 1, 4, 5]
\):

1. Calculate each pairwise comparison to 
n_d \).
2. Use the formulas above for each versio
whether there are ties or different numbe

---

By calculating the concordant and discord
formulas, you can determine each version 
```

- (n_d): Number of discordant pairs
- (n_0 = \frac{n(n-1)}{2}): Total number of pairwise comparisons
- (n_1 = \sum_i t_i (t_i - 1) / 2): Sum of tied values in (X)
- (n_2 = \sum_j u_j (u_j - 1) / 2): Sum of tied values in (Y)
    - (t_i): Number of ties in the (i)-th group of (X)
    - (u_j): Number of ties in the (j)-th group of (Y)

This formula accounts for the effect of ties in the data, providing a more accurate estimate in their presence.

---

5. Kendall Tau-c Coefficient

The **Kendall Tau-c** coefficient is defined as:

$$
\tau_C = \frac{2 (n_c - n_d)}{n^2 \cdot \frac{(m-1)}{m}} = \tau_A \cdot \frac{n - 1}{n} \cdot \frac{m}{m - 1}
$$

where:

- (n_c): Number of concordant pairs
- (n_d): Number of discordant pairs
- (r): Number of rows
- (c): Number of columns
- m = min(r, c) : The minimum of the number of rows and columns