Correlation

Correlation is a statistical measure that describes the extent to which two variables are related to each other. It indicates whether and how strongly pairs of variables are associated, ranging from -1 (perfect negative correlation) to +1 (perfect positive correlation), with 0 indicating no correlation.

Types of Correlation tests:

1:- Pearson's correlation coefficient

2:- Spearman's rank correlation coefficient

3:- Kendall's rank correlation coefficient

1:- Pearson's correlation coefficient

Pearson's correlation coefficient is a measure of the linear correlation between two variables X and Y. It has a value between +1 and −1, where 1 is total positive linear correlation, 0 is no linear correlation, and −1 is total negative linear correlation.

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

2:- Spearman's rank correlation coefficient

Spearman's rank correlation coefficient is a nonparametric measure of the monotonicity of the relationship between two datasets. Unlike the Pearson correlation, the Spearman correlation does not assume that both datasets are normally distributed. Like other correlation coefficients, this one varies between +1 and −1 with 0 implying no correlation. Correlations of −1 or +1 imply an exact monotonic relationship. Positive correlations imply that as x increases, so does y. Negative correlations imply that as x increases, y decreases.

$$r_s = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

example of Pearson's correlation coefficient

In [1]:

```
# pearson's correlation coefficient
import pandas as pd
import numpy as np
def pearson(x, y):
    x_mean = np.mean(x)
    y_mean = np.mean(y)
    x_std = np.std(x)
    y_std = np.std(y)
    n = len(x)
    return sum((x-x_mean)*(y-y_mean))/(n*x_std*y_std)
# example dataset
x = np.array([1,2,3,4,5])
y = np.array([2,4,5,4,5])
print(f"Pearson Correlation Coefficient: {pearson(x,y)}")

# print with if else statement
if pearson(x,y) < 0.6 > 0:
    print("Positive Correlation.")
elif pearson(x,y) > 0.6:
    print("Highly Positive Correlation.")
elif pearson(x,y) > -0.6 < 0:
    print("Negative Correlation.")
elif pearson(x,y) < -0.6:
    print("Highly Negative Correlation.")
else:
    print("No Correlation.")
```

```
Pearson Correlation Coefficient: 0.7745966692414834
Highly Positive Correlation.
```

In [2]:

```python
# SPearman's correlation coefficient
def spearman(x, y):
    x_rank = pd.Series(x).rank()
    y_rank = pd.Series(y).rank()
    return pearson(x_rank, y_rank)

print(f"Spearman Correlation Coefficient: {spearman(x,y)}")

# print with if else statement
if spearman(x,y) < 0.6 > 0:
    print("Positive Correlation")
elif spearman(x,y) > 0.6:
    print("Highly Positive Correlation")
elif spearman(x,y) > -0.6 < 0:
    print("Negative Correlation")
elif spearman(x,y) < -0.6:
    print("Highly Negative Correlation")
else:
    print("No Correlation")
```

```
Spearman Correlation Coefficient: 0.7378647873726218
Highly Positive Correlation
```

Other methods to computes correlation

In [3]:

```python
import pandas as pd
import numpy as np

# example dataset
x = np.array([1,2,3,4,5])
y = np.array([2,4,5,4,5])

# pearson's correlation coefficient

pearson = np.corrcoef(x,y)
print(f"Pearson Correlation Coefficient: {pearson[0,1]}")
```

```
Pearson Correlation Coefficient: 0.7745966692414834
```

In [4]:

```python
# creata an example dataset
```

```
x = pd.Series([1,2,3,4,5])
y = pd.Series([2,4,5,4,5])

# pearson's correlation coefficient
pearson_corr = x.corr(y)
print(f"Pearson Correlation Coefficient: {pearson_corr}")
```

```
Pearson Correlation Coefficient: 0.7745966692414834
```

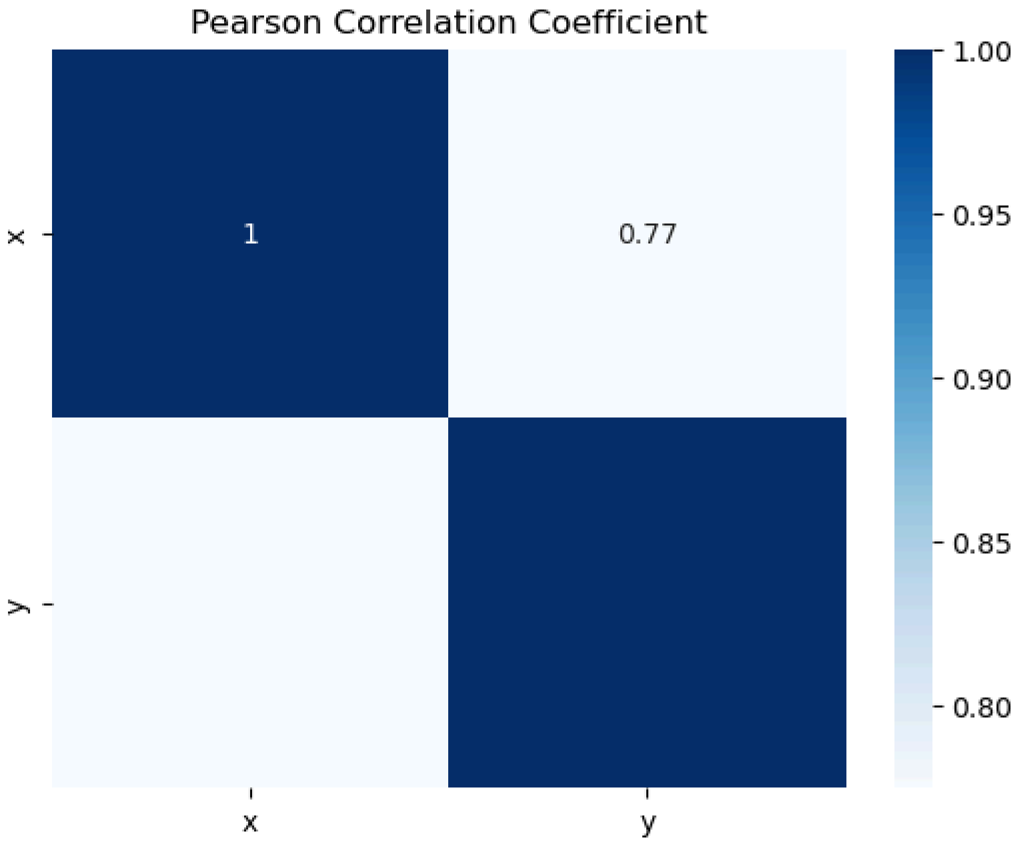
In [5]:

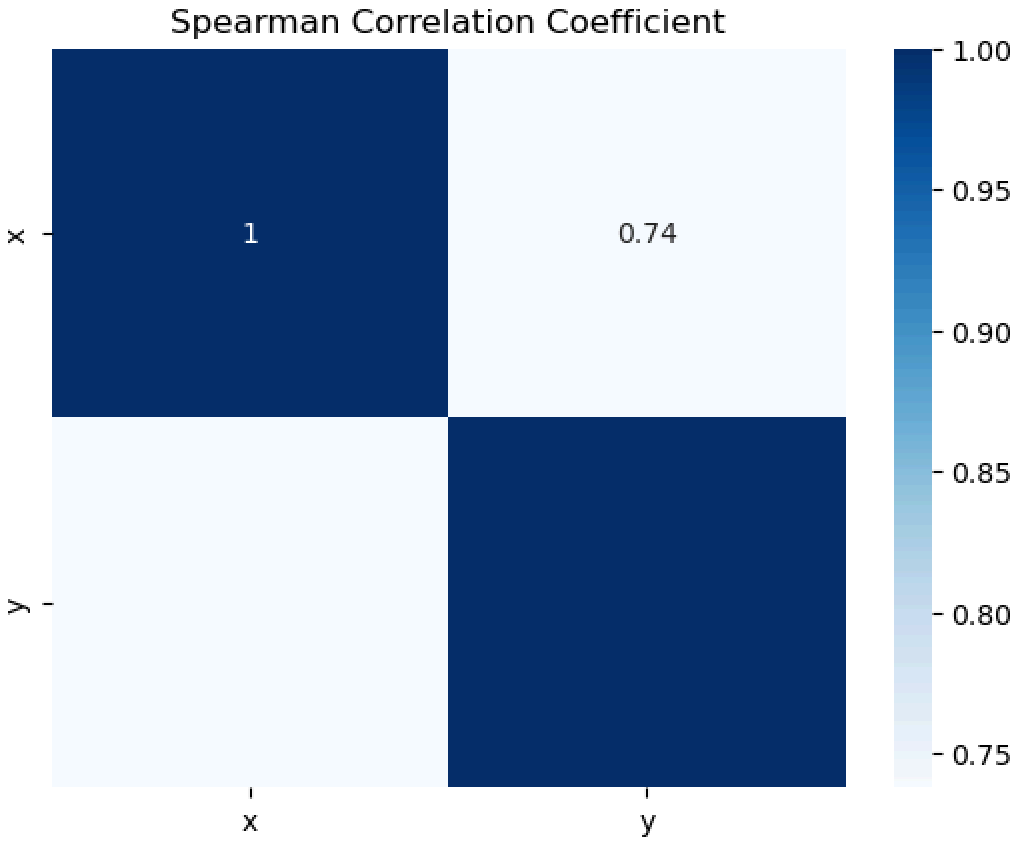
```
# using correlation matrix in pandas

df = pd.DataFrame({'x':x, 'y':y})
print(df.head())

# pearson's correlation coefficient
pearson_corr = df.corr(method = 'pearson')
spearman_corr = df.corr(method = 'spearman')
kendall_corr = df.corr(method = 'kendall')

print(f"Pearson Correlation Coefficient:\n {pearson_corr}")
print("====================================")
print(f"Spearman Correlation Coefficient:\n {spearman_corr}")
print("====================================")
print(f"Kendall Correlation Coefficient:\n {kendall_corr}")
```

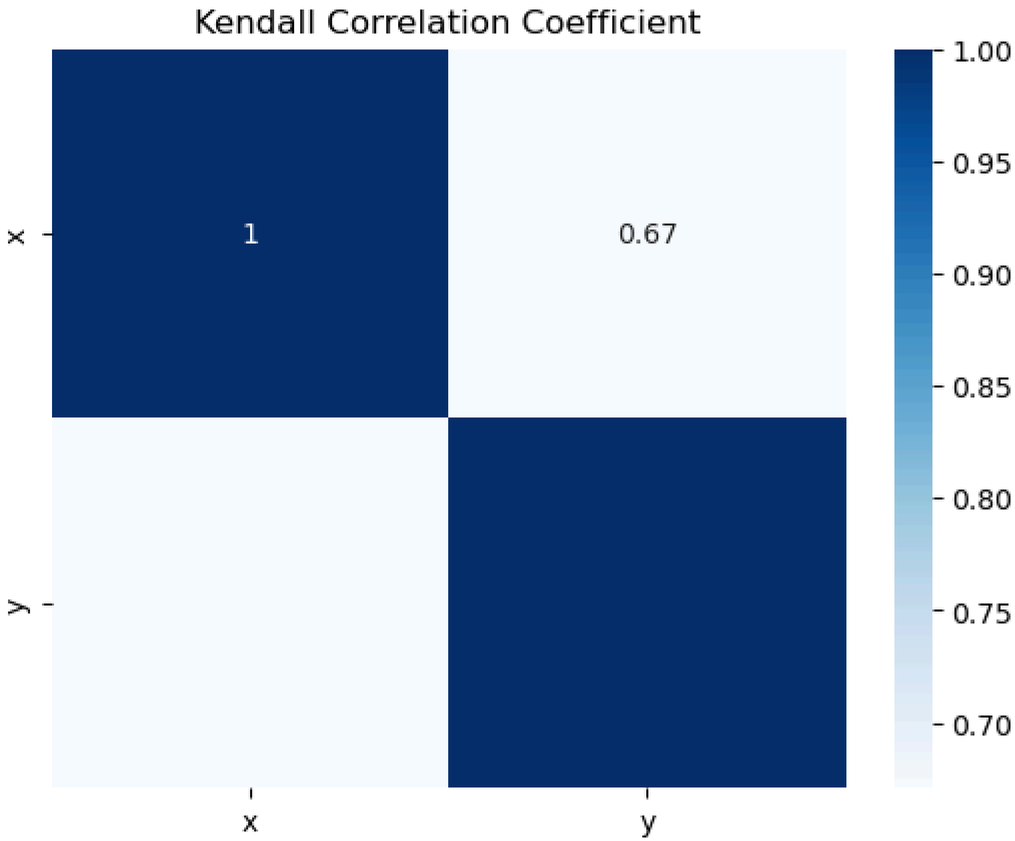
```
   x  y
0  1  2
1  2  4
2  3  5
3  4  4
4  5  5
Pearson Correlation Coefficient:
           x         y
x  1.000000  0.774597
y  0.774597  1.000000
====================================
Spearman Correlation Coefficient:
           x         y
x  1.000000  0.737865
y  0.737865  1.000000
====================================
Kendall Correlation Coefficient:
          x        y
x  1.00000  0.67082
y  0.67082  1.00000
```

In [6]:

```
# Draw a heatmap with the numeric values in each cell
import seaborn as sns
import matplotlib.pyplot as plt
sns.heatmap(pearson_corr, annot=True, cmap='Blues')
plt.title('Pearson Correlation Coefficient')
plt.show()
sns.heatmap(spearman_corr, annot=True, cmap='Blues')
plt.title('Spearman Correlation Coefficient')
plt.show()
sns.heatmap(kendall_corr, annot=True, cmap='Blues')
plt.title('Kendall Correlation Coefficient')
plt.show()
```


Pearson Correlation Coefficient
x
y
1
0.77
x
y
1.00
0.95
0.90
0.85
0.80


Spearman Correlation Coefficient
x
y
1
0.74
x
y
1.00
0.95
0.90
0.85
0.80
0.75


Kendall Correlation Coefficient
x
y
1
0.67
x
y
1.00
0.95
0.90
0.85
0.80
0.75
0.70

```
In [ ]:
```