What is Association ?

Association refers to a statistical relationship between two categorical variables. It indicates whether there's a dependency or connection between the categories of these variables.

1. Positive association: An increase in one variable makes it more likely for the other variable to increase.

2. Negative association: An increase in one variable makes it more likely for the other variable to decrease.

3. No association: The variables are independent; changes in one do not predict changes in the other.

> Categorical features are variables that can take on a limited, and usually fixed, number of values or categories. These values represent different groups or labels. Categorical features can be broadly classified into two types:

> Nominal variables: These are categories with no inherent order or ranking between them. Examples include colors, gender, or types of fruits.

> Ordinal variables: These categories have a meaningful order or rank. However, the intervals between the ranks may not be uniform or meaningful. Examples include education levels (e.g., high school, college, graduate), customer satisfaction levels (e.g., low, medium, high), or star ratings.

Example:

Consider a dataset of customers in an online store where we have their membership type (Gold, Silver, Bronze) and purchase frequency. If there's a positive association, it suggests that customers with Gold memberships tend to make purchases more frequently.

```
In [1]: import pandas as pd
        from scipy.stats import chi2_contingency

        data = {'Membership_Type': ['Gold', 'Silver', 'Silver', 'Bronze', 'Gold'],
                'Purchase_Frequency': ['High', 'Medium', 'High', 'Low', 'High']}

        df = pd.DataFrame(data)

        contingency_table = pd.crosstab(df['Membership_Type'], df['Purchase_Frequency'])

        chi2, p, dof, expected = chi2_contingency(contingency_table)

        print(f"Chi2 value: {chi2}")

        print(f"P-value: {p}")

        print(f"Degree of Freedom : {dof}")

        print("Expected frequencies:")
        print(expected)
```

```
C:\Users\Gyanender\anaconda3\lib\site-packages\scipy\__init__.py:155: UserWarning:
A NumPy version >=1.18.5 and <1.25.0 is required for this version of SciPy (detecte
d version 1.26.0
  warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}"

Chi2 value: 6.666666666666668
P-value: 0.15458730450476033
Degree of Freedom : 4
Expected frequencies:
[[0.6 0.2 0.2]
 [1.2 0.4 0.4]
 [1.2 0.4 0.4]]
```

Let's break down each concept involved in the provided code:

1. Pandas DataFrame:

> pd.DataFrame(data): Creates a pandas DataFrame from the given data. In this example, the data is a dictionary with keys as column names ('Membership_Type' and 'Purchase_Frequency') and values as lists representing the data in those columns.

2. Contingency Table:

> pd.crosstab(df['Membership_Type'], df['Purchase_Frequency']): Creates a contingency table, a two-dimensional table that displays the frequency distribution of variables. In this case, it shows how many times each combination of 'Category1' and 'Category2' occurs in the DataFrame.

3. Chi-square Test:

> chi2_contingency(contingency_table): Performs a Chi-square test of independence. This test is used to determine if there is a significant association between the categorical variables represented by the contingency table.

> The function returns several values, including the Chi2 statistic (chi2), p-value (p), degrees of freedom (dof), and expected frequencies (expected). The expected frequencies are the values that would be expected if the variables were independent.

4. Chi2 Value:

> print(f"Chi2 value: {chi2}"): Prints the Chi2 statistic, which measures the discrepancy between the observed and expected frequencies.

> A larger Chi2 value indicates a larger difference, and if it's significantly large, it suggests a relationship between the variables.

5. P-value:

> print(f"P-value: {p}"): Prints the p-value associated with the Chi2 statistic. A small p-value (typically less than 0.05) suggests that you can reject the null hypothesis of independence, indicating a significant association between the categorical variables.

6. Degrees of Freedom:

> print(f"Degrees of freedom: {dof}"): Prints the degrees of freedom associated with the Chi2 test. It's a measure of the amount of information available for estimating the population parameters.

7. Expected Frequencies:

> print("Expected frequencies:"): Prints the expected frequencies obtained from the Chi2 test. These are the values that would be expected under the assumption of independence.

Examples (Numerical with Numerical)

In [2]:
```python
import numpy as np
from scipy.stats import chi2_contingency

# Create a contingency table (replace data with your own)
observed_data = np.array([[30, 10, 20],
                          [15, 25, 35]])

# Perform the Chi-square test
chi2, p, dof, expected = chi2_contingency(observed_data)

# Print the results
print(f"Chi2 value: {chi2}")
print(f"P-value: {p}")
print(f"Degrees of freedom: {dof}")
print("Expected frequencies:")
print(expected)
```

```
Chi2 value: 14.025974025974026
P-value: 0.0009001159109510967
Degrees of freedom: 2
Expected frequencies:
[[20.         15.55555556 24.44444444]
 [25.         19.44444444 30.55555556]]
```

Examples (Categorical with Categorical)

In [3]:
```
import pandas as pd
from scipy.stats import chi2_contingency

# Create a DataFrame with categorical data (replace data with your own)
data = {'Category1': ['A', 'B', 'A', 'B', 'A'],
        'Category2': ['X', 'Y', 'X', 'Y', 'Z']}
df = pd.DataFrame(data)

# Create a contingency table
contingency_table = pd.crosstab(df['Category1'], df['Category2'])

# Perform the Chi-square test
chi2, p, dof, expected = chi2_contingency(contingency_table)

# Print the results
print(f"Chi2 value: {chi2}")
print(f"P-value: {p}")
print(f"Degrees of freedom: {dof}")
print("Expected frequencies:")
print(expected)
```

```
Chi2 value: 5.0
P-value: 0.0820849986238988
Degrees of freedom: 2
Expected frequencies:
[[1.2 1.2 0.6]
 [0.8 0.8 0.4]]
```

Let's create an example with both nominal and ordinal categorical values and perform a Chi-square test in Python

In [4]:
```
import pandas as pd
from scipy.stats import chi2_contingency

# Create a DataFrame with nominal and ordinal categorical data
data = {'NominalCategory': ['A', 'B', 'A', 'B', 'A'],
        'OrdinalCategory': ['High', 'Low', 'Medium', 'High', 'Low']}
df = pd.DataFrame(data)

# Create a contingency table
contingency_table = pd.crosstab(df['NominalCategory'], df['OrdinalCategory'])

# Perform the Chi-square test
chi2, p, dof, expected = chi2_contingency(contingency_table)

# Print the results
print(f"Chi2 value: {chi2}")
print(f"P-value: {p}")
print(f"Degrees of freedom: {dof}")
print("Expected frequencies:")
print(expected)
```

```
Chi2 value: 0.8333333333333335
P-value: 0.6592406302004437
Degrees of freedom: 2
Expected frequencies:
[[1.2 1.2 0.6]
 [0.8 0.8 0.4]]
```

What if I want to check correlation and association between more than 2 cateogorical values ?

If you want to check correlation and association between more than two categorical variables, you can use techniques like the Cramér's V statistic for association and consider other statistical methods for exploring relationships. Below is a brief overview:

1. Cramér's V for Association:

Cramér's V:

> Definition: Cramér's V is a measure of association between two categorical variables.

> Purpose: It quantifies the strength of association, similar to how correlation coefficients quantify the strength of linear relationships between continuous variables.

> Range: Values range from 0 to 1, where 0 indicates no association, and 1 indicates a perfect association.

> Cramér's V is a measure of association between two nominal variables. It ranges from 0 to 1, where 0 indicates no association, and 1 indicates a perfect association.

In [5]:
```python
import pandas as pd
import numpy as np
from scipy.stats import chi2_contingency

def cramers_v(x, y):
    confusion_matrix = pd.crosstab(x, y)
    chi2, _, _, _ = chi2_contingency(confusion_matrix)
    n = confusion_matrix.sum().sum()
    phi2 = chi2 / n
    r, k = confusion_matrix.shape
    phi2corr = max(0, phi2 - ((k - 1) * (r - 1)) / (n - 1))
    rcorr = r - ((r - 1) ** 2) / (n - 1)
    kcorr = k - ((k - 1) ** 2) / (n - 1)
    return np.sqrt(phi2corr / min((kcorr - 1), (rcorr - 1)))

data = {'Category1': ['A', 'B', 'A', 'B', 'A'],
        'Category2': ['X', 'Y', 'X', 'Y', 'Z'],
        'Category3': ['High', 'Low', 'Medium', 'High', 'Low']}

df = pd.DataFrame(data)

associations = pd.DataFrame(index=df.columns, columns=df.columns)
for i in range(len(df.columns)):
    for j in range(len(df.columns)):
        associations.iloc[i, j] = cramers_v(df.iloc[:, i], df.iloc[:, j])

# Print the associations

print("Cramér's V for Association:")
print(associations)
```

```
Cramér's V for Association:
          Category1 Category2 Category3
Category1  0.346944  0.816497       0.0
Category2  0.816497       1.0       0.0
Category3       0.0       0.0       1.0
```

Function to Calculate Cramér's V:

Input: Two categorical variables (x and y).

Process:

Computes a confusion matrix, which shows the joint frequency distribution of the two variables.

Calculates the Chi-square statistic from the confusion matrix.

Converts the Chi-square statistic into Cramér's V.

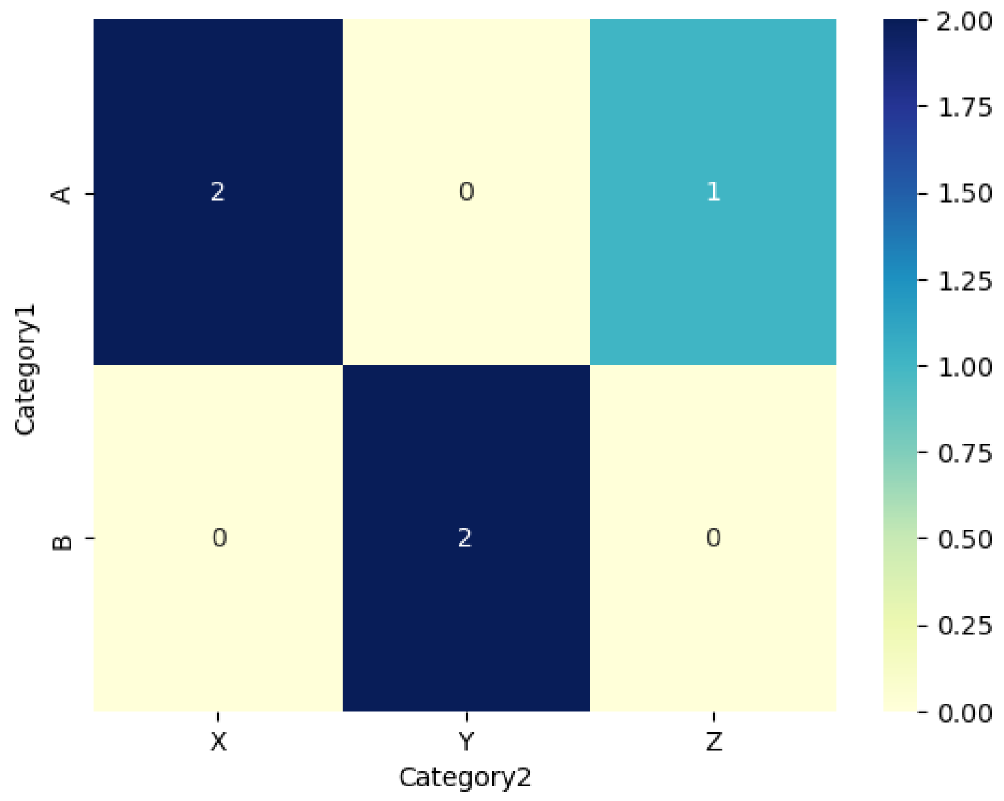
2. Explore Relationships:

Explore relationships between categorical variables visually and using statistical tests. Techniques include:

> Heatmap: Create a heatmap of the contingency tables or correlation coefficients for a visual representation.

In [6]:
```
import seaborn as sns
import matplotlib as plt
sns.heatmap(pd.crosstab(df['Category1'], df['Category2']), annot=True, cmap="YlGnBu"
```

Out[6]:
```
<AxesSubplot:xlabel='Category2', ylabel='Category1'>
```



> Chi-square Test: Perform Chi-square tests between pairs of categorical variables.

In [7]:
```
for col1 in df.columns:
    for col2 in df.columns:
        if col1 != col2:
            chi2, p, _, _ = chi2_contingency(pd.crosstab(df[col1], df[col2]))
            print(f"Chi2 test between {col1} and {col2}: p-value = {p}")
```

```
Chi2 test between Category1 and Category2: p-value = 0.0820849986238988
Chi2 test between Category1 and Category3: p-value = 0.6592406302004437
Chi2 test between Category2 and Category1: p-value = 0.0820849986238988
Chi2 test between Category2 and Category3: p-value = 0.44089552967916945
Chi2 test between Category3 and Category1: p-value = 0.6592406302004438
Chi2 test between Category3 and Category2: p-value = 0.44089552967916945
```

Iterating Through Pairs of Categorical Variables:

For Loops: Nested loops to iterate through each pair of categorical variables in the DataFrame.

Cramér's V Calculation: For each pair, the code calculates Cramér's V using the function mentioned above.

Output:

> Cramér's V Matrix: The resulting matrix (associations) contains Cramér's V values for all pairs of categorical variables.

Interpretation:

> High Cramér's V: Indicates a strong association between the corresponding categorical variables.

> Low Cramér's V: Indicates a weak or no association.

In summary, the code calculates Cramér's V for all pairs of categorical variables in a DataFrame, providing a measure of the strength of association between these variables.

Categorical-Continuous: ANOVA Test

If you have a categorical variable and a continuous variable, you can use ANOVA (Analysis of Variance) to check if there are significant differences in the means across different categories.

In [8]:
```
import pandas as pd
from scipy.stats import f_oneway

# Create a DataFrame with a categorical and a numerical variable
data = {'Category': ['A', 'B', 'A', 'B', 'A'],
        'NumericalVariable': [10, 15, 8, 12, 9]}
df = pd.DataFrame(data)

# Perform one-way ANOVA
f_statistic, p_value = f_oneway(df['NumericalVariable'][df['Category'] == 'A'],
                                 df['NumericalVariable'][df['Category'] == 'B'])

# Print the results
print(f"F-statistic: {f_statistic}")
print(f"P-value: {p_value}")
```

```
F-statistic: 11.215384615384622
P-value: 0.044093614110389194
```