Varun Sagar
Theegala

HOW TO PERFORM CORRESPONDENCE ANALYSIS IN PYTHON WITH PRINCE

Varun Sagar
Theegala

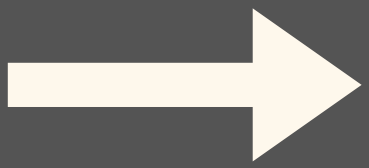
Correspondence Analysis (CA) is a powerful statistical technique used to **analyze relationships between two categorical variables.**

It allows you to visualize how categories in one variable (e.g., emotions) relate to categories in another (e.g., drinks), **showing associations in a low-dimensional space.**

It’s frequently used in marketing, surveys, and behavioral research.

Varun Sagar Theegala

LET'S TAKE AN EXAMPLE OF COLD-DRINKS & THE EMOTION BRANDS WANT TO CONNECT THEM TO



STEP 1

IMPORT LIBRARIES

To perform Correspondence Analysis, you'll need the following Python libraries:

- **prince** for the analysis,
- **pandas** for handling data
- **matplotlib** for visualization.

```
import pandas as pd
import matplotlib.pyplot as plt

#%pip install prince
import prince
```

STEP 2

CREATE YOUR DATASET

Start by creating a dataset in the form of a contingency table. Each row represents one category (e.g., emotions like happiness or excitement), and each column represents another (e.g., drinks like Pepsi or Coca Cola).

```
In [8]: data = {
            'Pepsi': [5, 30, 10, 45],
            'Sprite': [25, 10, 5, 60],
            'Coca Cola': [40, 5, 35, 20]
        }

        # Emotions people associate with each drink
        emotions = ['Happiness', 'Excitement', 'Nostalgia', 'Refreshment']

        # Create DataFrame
        df = pd.DataFrame(data, index=emotions)
        print(df)
```

```
              Pepsi  Sprite  Coca Cola
Happiness         5      25         40
Excitement       30      10          5
Nostalgia        10       5         35
Refreshment      45      60         20
```

Varun Sagar Theegala

NEXT

STEP 3

INITIALIZE & FIT MODEL FOR CORRESPONDENCE ANALYSIS

Use the prince library to initialize and fit the Correspondence Analysis model.

In this case, we set n_components=2 to reduce the data into two dimensions for better visualization.

Fitting the model allows it to compute the relationships between categories (rows) and factors (columns).

```
# Perform Correspondence Analysis using Prince
ca = prince.CA(n_components=2)  # n_components=2 for 2D plot
ca = ca.fit(df)
```

Varun Sagar Theegala

NEXT ➡

STEP 4

EXTRACT ROW & COLUMN COORDINATES

Once the model is fit, extract the coordinates for both rows (emotions) and columns (drinks) to prepare for visualization.

These coordinates represent the position of each category and factor in the 2-D space, allowing us to see which category & factor are close.

```
# Get row and column coordinates for visualization
row_coordinates = ca.row_coordinates(df)
col_coordinates = ca.column_coordinates(df)
```

STEP 5

EXTRACT ROW & COLUMN COORDINATES

Use matplotlib to create a scatter plot that visually represents the associations between rows and columns.

```
# Plot the Correspondence Analysis result
fig, ax = plt.subplots(figsize=(10, 7))

# Plot the row coordinates (emotions)
ax.scatter(row_coordinates[0], row_coordinates[1], color='red', label='Emotions')
for i, txt in enumerate(df.index):
    ax.annotate(txt, (row_coordinates[0][i], row_coordinates[1][i]), color='red')

# Plot the column coordinates (cold drinks)
ax.scatter(col_coordinates[0], col_coordinates[1], color='blue', label='Cold Drinks')
for i, txt in enumerate(df.columns):
    ax.annotate(txt, (col_coordinates[0][i], col_coordinates[1][i]), color='blue')

# Add labels and title
plt.title('Correspondence Analysis: Cold Drinks and Associated Emotions', fontsize=16)
plt.xlabel('Component 1')
plt.ylabel('Component 2')
plt.axhline(0, color='black', linewidth=0.5)
plt.axvline(0, color='black', linewidth=0.5)
plt.legend()
plt.grid(True)
plt.show()
```

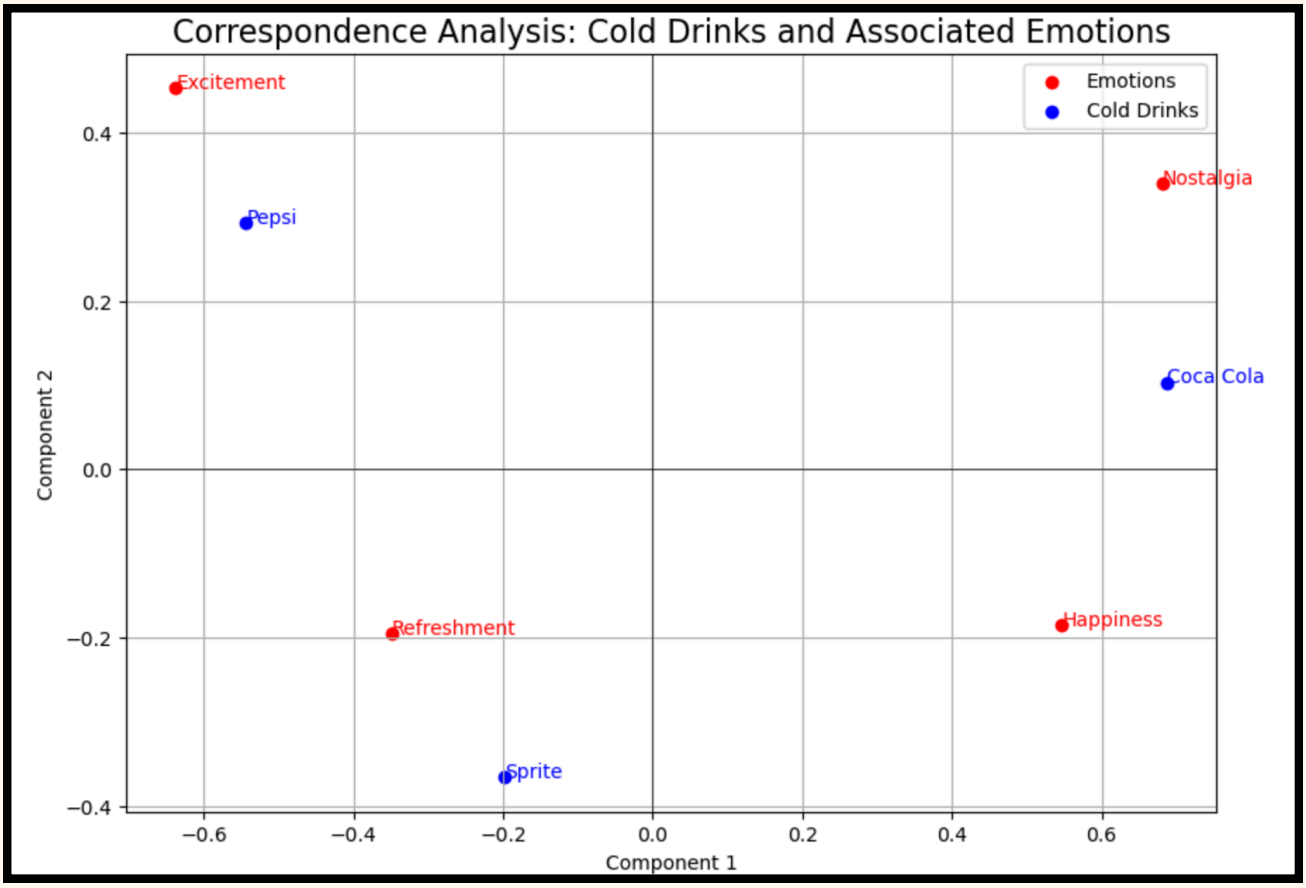
Varun Sagar Theegala

NEXT ➡

STEP 6

INTERPRET RESULTS

Categories closer together are more closely related. For example, if "Pepsi" is close to "Excitement," it means people associate Pepsi more with that emotion.



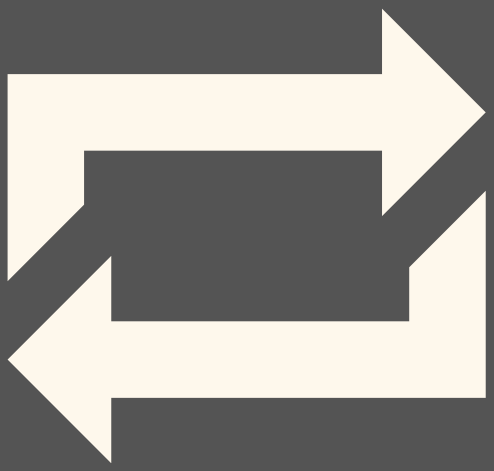
Varun Sagar Theegala

NEXT

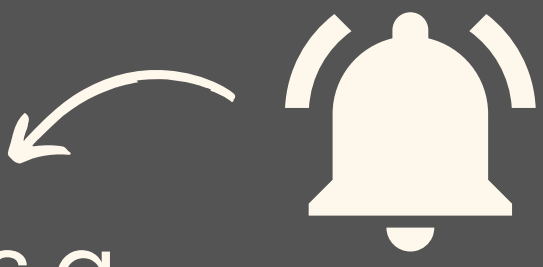
Varun Sagar Theegala

SHARE THIS

If you think **your network** would find this **valuable**



FOLLOW ME



I help you **GROW & SUSTAIN** as a **Data Analyst**