Correspondence Analysis (CA) - Mini Porjet

Team members : **Atsamnia Abderraouf │ Toubal Seghir Kheireddine**

Group : **02 │ ISI**

Step 1: Constructing the Contingency Table

We first construct the contingency table that contains the frequency of occurrence for each sector in each country. This is represented as a matrix with countries as rows and sectors as columns.

```
python
CopyEdit
data = {
    'Tech': [120, 90, 110, 30, 70, 60],
    'Finance': [80, 40, 60, 20, 60, 40],
    'Energy': [60, 80, 100, 40, 50, 70],
    'Agriculture': [40, 30, 50, 70, 25, 60],
    'Health': [90, 70, 60, 20, 50, 30],
    'Retail': [70, 50, 80, 30, 40, 60]
}

countries = ['USA', 'Germany', 'China', 'Brazil', 'France', 'India']
df = pd.DataFrame(data, index=countries)
```

Step 2: Row and Column Profiles

To better understand the relationship between rows (countries) and columns (sectors), we compute the **row profile** and **column profile**.

Row Profile:

For each country, the row profile is the relative distribution of values. It is computed by dividing each entry in a row by the sum of the row.

```
row_profiles = df.div(df.sum(axis=1), axis=0)
```

Column Profile:

For each sector, the column profile is the relative distribution of values. It is computed by dividing each entry in a column by the sum of the column.

```
col_profiles = df.div(df.sum(axis=0), axis=1)
```

Step 3: Performing Correspondence Analysis (CA)

Now we apply the **Correspondence Analysis** technique using the `prince` Python library. The goal is to reduce the dimensions of the contingency table into a smaller number of components (F1, F2), which represent the most significant relationships.

```
ca = prince.CA(n_components=2, n_iter=10, copy=True, check_input=True, engine='sklearn')
ca = ca.fit(df)
```

Step 4: Eigenvalues and Explained Inertia

Eigenvalues represent the variance captured by each component. **Inertia** is a measure of how much variance is explained by each factor.

```
eigenvalues = ca.eigenvalues_
total_inertia = sum(eigenvalues)
explained_inertia = eigenvalues / total_inertia
```

Step 5: Factor Coordinates (Projections)

The **factor coordinates** are the projections of the rows (countries) and columns (sectors) onto the principal components (F1 and F2).

Row Coordinates (Countries):

The row coordinates are obtained by projecting the rows onto the principal components.

Column Coordinates (Sectors):

The column coordinates are obtained by projecting the columns onto the principal components.

```
row_coords = ca.row_coordinates(df)
col_coords = ca.column_coordinates(df)
```

Step 6: Biplot Visualization

We use a **biplot** to visualize the relationship between countries (rows) and sectors (columns) on the first two principal components (F1, F2). This provides a graphical representation of how countries are associated with sectors.

```
plt.figure(figsize=(10, 8))
plt.axhline(0, color='gray', lw=1)
plt.axvline(0, color='gray', lw=1)
```

```
# Row points (countries)
for i, country in enumerate(df.index):
    x, y = row_coords.iloc[i, 0], row_coords.iloc[i, 1]
    plt.scatter(x, y, marker='o', color='blue')
    plt.text(x + 0.01, y, country, fontsize=10)

# Column points (sectors)
for j, sector in enumerate(df.columns):
    x, y = col_coords.iloc[j, 0], col_coords.iloc[j, 1]
    plt.scatter(x, y, marker='^', color='red')
    plt.text(x + 0.01, y, sector, fontsize=10)

plt.title("Step 6: Correspondence Analysis Biplot (F1 vs F2)")
plt.xlabel(f"F1 ({explained_inertia[0]*100:.2f}%)")
plt.ylabel(f"F2 ({explained_inertia[1]*100:.2f}%)")
plt.grid(True)
plt.show()
```

Step 7: Quality of Representation and Contributions

The **quality of representation** (cos²) and **contributions** of rows and columns are calculated to assess how well the rows and columns are represented by the two principal components.

Cosine of the Angle (cos²):

This value indicates the quality of representation of the rows and columns by the principal components.

Contributions:

The contributions represent how much each row or column contributes to the variance explained by each factor.

```
# Cosine Similarities (Quality of Representation)
```

```
print("\nStep 7: Quality of Representation (cos²) for Rows")
print(ca.row_cosine_similarities(df).round(3))

# Contributions
print("\nStep 7: Contributions of Rows to Axes")
print(ca.row_contributions_.round(3))

print("\nStep 7: Contributions of Columns to Axes")
print(ca.column_contributions_.round(3))
```

Step 8: Chi-Square and Cramer's V

To validate the results statistically, we calculate the **Chi-Square** statistic and **Cramer's V** for measuring the association strength between countries and sectors.

```
from scipy.stats import chi2_contingency

chi2, p, dof, expected = chi2_contingency(df)
cramers_v = np.sqrt(chi2 / (df.values.sum() * (min(df.shape) - 1)))

print("\nStep 8: Validation")
print(f"Chi² Statistic: {chi2:.3f}")
print(f"Degrees of Freedom: {dof}")
print(f"Cramer's V: {cramers_v:.3f}")
```

Conclusion

This report demonstrates the power of **Correspondence Analysis** in uncovering hidden patterns between categorical variables in contingency tables. By visualizing the results and examining the statistical metrics (such as Chi-Square and Cramer's V), we gain insights into the associations between countries and sectors.

References

- Greenacre, M. J. (1984). *Correspondence Analysis in Practice*. Academic Press.
- László, G., & Móricz, R. (2009). *Correspondence Analysis for Categorical Data: Applications to Marketing Research*