Lab CA: Correspondence Analysis (CA) using prince on the French_elections dataset

Lab Duration: 2-3 Hours
Prerequisites: Basic understanding of Python, Pandas, Matplotlib, and Correspondence Analysis (CA)

Lab Objectives

By the end of this lab, students will be able to:

1. Perform CA using the prince library.
2. Show and interpret the eigenvalues, explained variance and cumulative explained variance.
3. Get the new coordinates for rows (Row scores) and columns (Column scores) and show the data in the new Factor map.
4. Get and interpret the column/row contributions to the total explained variance of each dimension.
5. Know the quality of representation of each point in the factor map.

Lab Outline

Part 1: Introduction to CA and the French Elections Voting Data

- Overview of CA.
- Description of the dataset (which comes with the prince library).

Part 2: Setting up the Environment and Loading Data

- Installing required libraries (prince : version 0.13.1)
- Loading the **French Elections Voting Data** dataset using prince.

Part 3: Performing CA using prince

- Initializing the prince CA model.
- Fitting the model to the data.
- Getting the explained variance
- Transforming the data and getting the new columns and rows' coordinates

Part 4: Visualization & Interpretation

- Visualizing the data in the new dimensions (a 2-D space).
- Display rows and columns contributions.

- Display the quality of representation of each data point on the factor maps (cosine similarities)
- Discussing the insights obtained from the plots.

Part 5: Conclusion and Q&A

- Summarize key points.

Part 1: Introduction to CA and the French Elections Voting Data

CA Overview

Correspondence Analysis (CA)

- Correspondence analysis is a useful data visualization technique for finding out and displaying the relationship between categories. It uses a graph that plots data, visually showing the outcome of two or more data points.
- CA uses a contingency table—a table of frequencies—that shows how the categories of the variables are distributed. The data in the table undergoes a series of transformations in relation to the data around it to produce relational data. The resulting data is then plotted to show those relationships visually.

The French Elections Voting Dataset

- This dataset counts the number of voters per region for each candidate in the 2022 French presidential elections. It can be used directly for correspondence analysis.
- The dataset's name in this context is usually **French Elections Voting Data**, and it consists of a contingency table where the rows represent the regions or departments, and the columns represent the candidates. Each cell in the table holds the number of votes for a particular candidate in a specific region.

Part 2: Setting up the Environment and Loading Data

Step 1: Install Required Libraries

- Install the required Python packages:

pip install prince pandas seaborn matplotlib scikit-learn

Step 2: Load the Dataset and display its first 5 rows

```python
import prince

dataset = prince.datasets.load_french_elections()
dataset.head()
```

Question: what do you think about the format of “dataset” ?

This dataset is already available as a contingency matrix. It’s more common to have at one’s disposal a flat dataset. If this is the case, a contigency matrix can be obtained using the pivot_table function in pandas.

Part 3: Performing CA using prince

Step 1: Initialize the CA Model

- Initialize CA with 2 components:

```python
# Initialize CA model
ca = prince.CA(
    n_components=2,
    n_iter=3,
    copy=True,
    check_input=True,
    engine='sklearn',
    random_state=42
)
```

Q. Explain the main parameters ?

Step 2: Fit the CA Model

- Fit the CA model to the dataset:

```python
# Fit the CA model on the data
ca = ca.fit(dataset)
```

- Print eigenvalues, explained variance and total inertia of ca

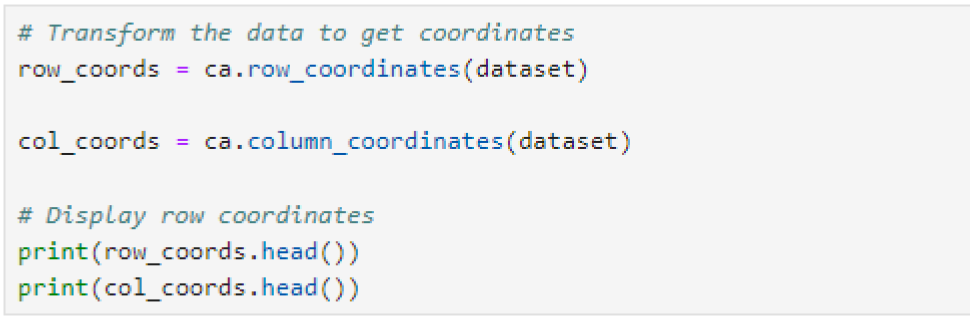
```python
#Eigenvalues
print(ca.eigenvalues_summary)
print(ca.total_inertia_)
```

Q. How is the percentage of explained variance calculated for each component?

Step 3: Transforming the data and getting the new columns and rows' coordinates

Get the rows (regions) and columns (candidates) coordinates.

Note that there is no "ca.transform()" as ca.row_coordinates() and ca.column_coordinates() already do the transformation.

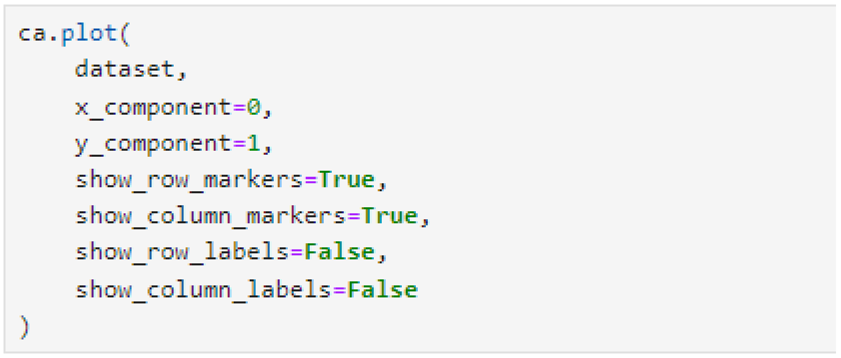


```
# Transform the data to get coordinates
row_coords = ca.row_coordinates(dataset)

col_coords = ca.column_coordinates(dataset)

# Display row coordinates
print(row_coords.head())
print(col_coords.head())
```

Part 4: Visualization & Interpretation

We can use the plot() function of prince library or use matplotlib (as we did in the PCA lab):

- **1st visualization: Biplot showing the row and column factor maps:**

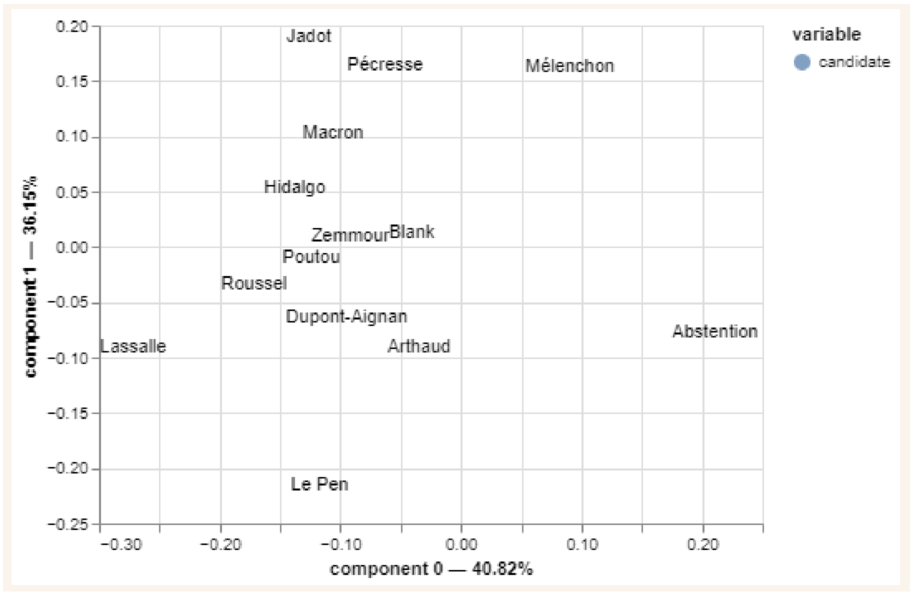


```
ca.plot(
    dataset,
    x_component=0,
    y_component=1,
    show_row_markers=True,
    show_column_markers=True,
    show_row_labels=False,
    show_column_labels=False
)
```

Q1. Explain the different parameters?

Q2. What do these factor maps show?

- **2nd visualization:** How can we change the above parameters to get the following column factor map ?



- **Displaying the contributions of columns to the 1st dimension in a barplot:**

```
import matplotlib.pyplot as plt
import seaborn as sns

column_contributions = ca.column_contributions_

# Convert the column contributions to a DataFrame for easy plotting
contrib_df = column_contributions.iloc[:, 0]  # Use first dimension (0-based index)

# Plot the bar chart
plt.figure(figsize=(10, 6))
sns.barplot(x=contrib_df.index, y=contrib_df.values)
plt.title('Column Contributions to the 1st Dimension')
plt.xlabel('Columns (Candidates)')
plt.ylabel('Contribution')
plt.xticks(rotation=45, ha='right')  # Rotate labels for readability
plt.tight_layout()
plt.show()
```

- **Displaying the contributions of rows to the 1st dimension in a barplot.**
  Change the above code.

- **Getting the quality representation of each point in the 1st dimension:**
  - **Cosine similarities of Columns (Candidates):**

```
# Get the quality of the representation of each point in the reduced space. Higher values indicate that a point is well-represented on
#the selected dimensions.
ca.row_cosine_similarities(dataset)
```

```python
import matplotlib.pyplot as plt
import seaborn as sns

col_cos = ca.column_cosine_similarities(dataset)


col_cos_df = col_cos.iloc[:, 0]  # Use first dimension (0-based index)

# Plot the bar chart
plt.figure(figsize=(10, 6))
sns.barplot(x=col_cos_df.index, y=col_cos_df.values)
plt.title('Column Cosine similarities - First dimension')
plt.xlabel('Columns (Candidates)')
plt.ylabel('Cosine similarities')
plt.xticks(rotation=45, ha='right')  # Rotate labels for readability
plt.tight_layout()
plt.show()
```

Part 5: Conclusions & QA

- Discuss how much variance is explained by the first two dimensions and the importance of each dimension.
- Discuss how columns (candidates) and rows (regions) contribute to the factors and what the plot tells us about the relationship about the two variables.

Useful summary:

The final output of Correspondence Analysis (CA) consists of several key components that help interpret the relationships between the rows and columns of a contingency table (i.e., categorical variables). These outputs typically include:

1. Row Coordinates (Row Scores)

These are the coordinates of the rows (e.g., regions, departments) in the reduced factor space. They show how each row is represented on the principal dimensions (factors) derived from the analysis.

These coordinates can be plotted to visualize how different rows are positioned relative to each other, indicating similarities or differences between the categories.

2. Column Coordinates (Column Scores)

These are the coordinates of the columns (e.g., candidates, categories) in the reduced factor space. Similar to row coordinates, they represent how each column relates to the principal dimensions.

These scores can also be visualized on a scatter plot alongside the row coordinates, showing relationships between rows and columns.

3. Eigenvalues and Explained Inertia

Eigenvalues represent the amount of inertia (variance) captured by each dimension. Higher eigenvalues indicate dimensions that explain more of the association in the data.

Explained inertia is the proportion of the total inertia (variance) explained by each dimension. It helps assess how much information is captured by the first few dimensions and determines the optimal number of dimensions to keep.

4. Row Contributions

The contribution of each row to the total inertia of a given dimension. This helps identify which rows (categories) are most responsible for the formation of each dimension.

5. Column Contributions

The contribution of each column to the total inertia of a dimension. This shows which columns (categories) have the greatest influence on a specific dimension.

6. Row and Column Masses

Row masses represent the relative importance (weight) of each row in the contingency table.

Column masses represent the relative importance of each column. These masses are used in normalization during CA to account for uneven distributions.

7. Factor Maps (Biplots)

These are visual representations of the row and column coordinates on the first two or more dimensions. The row factor map shows the distribution of rows in the reduced space, while the column factor map shows the positioning of columns.

A biplot can display both row and column coordinates on the same graph, showing the associations between rows and columns in the reduced dimensional space.