Lab PCA: Principal Component Analysis (PCA) using prince on the Iris Dataset

Lab Duration: 3 Hours
Level: Intermediate
Prerequisites: Basic understanding of Python, Pandas, Matplotlib, and Principal Component Analysis (PCA)

Lab Objectives

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

1. Perform PCA using the prince library.
2. Visualize the explained variance using a Scree plot.
3. Create a Correlation Circle to interpret the relationships between original features.
4. Generate a Score Plot to visualize the data in the principal component space.

Lab Outline

Part 1: Introduction to PCA and the Iris Dataset (15 minutes)

- Overview of PCA.
- Description of the Iris dataset.

Part 2: Setting up the Environment and Loading Data (15 minutes)

- Installing required libraries.
- Loading the Iris dataset using sklearn.
- Preprocessing data with Pandas.

Part 3: Performing PCA using prince (30 minutes)

- Initializing the prince PCA model.
- Fitting the model to the data.
- Extracting principal components and explained variance.

Part 4: Visualization (60 minutes)

- **Scree Plot**: Visualizing explained variance by each principal component.
- **Correlation Circle**: Understanding relationships between original features and principal components.

- **Score Plot**: Visualizing the dataset in the new principal component space.

Part 5: Interpretation and Analysis (30 minutes)

- Discussing the insights obtained from the plots.
- How to interpret the variance and the position of the features in the correlation circle.
- Analyzing the grouping of species in the score plot.

Part 6: Conclusion and Q&A (30 minutes)

- Summarize key points.
- Open the floor for questions and additional discussion.

Part 1: Introduction to PCA and the Iris Dataset

Time: 15 minutes

PCA Overview

- **Principal Component Analysis (PCA)** is a dimensionality reduction technique that transforms a dataset into a set of orthogonal (uncorrelated) variables called principal components.
- PCA is useful for reducing the complexity of data, visualizing high-dimensional datasets, and identifying patterns.

The Iris Dataset

- The Iris dataset consists of 150 samples of iris flowers, with 50 samples each from three species: Setosa, Versicolor, and Virginica.
- The dataset has four features: sepal length, sepal width, petal length, and petal width.

Part 2: Setting up the Environment and Loading Data

Time: 15 minutes

Step 1: Install Required Libraries

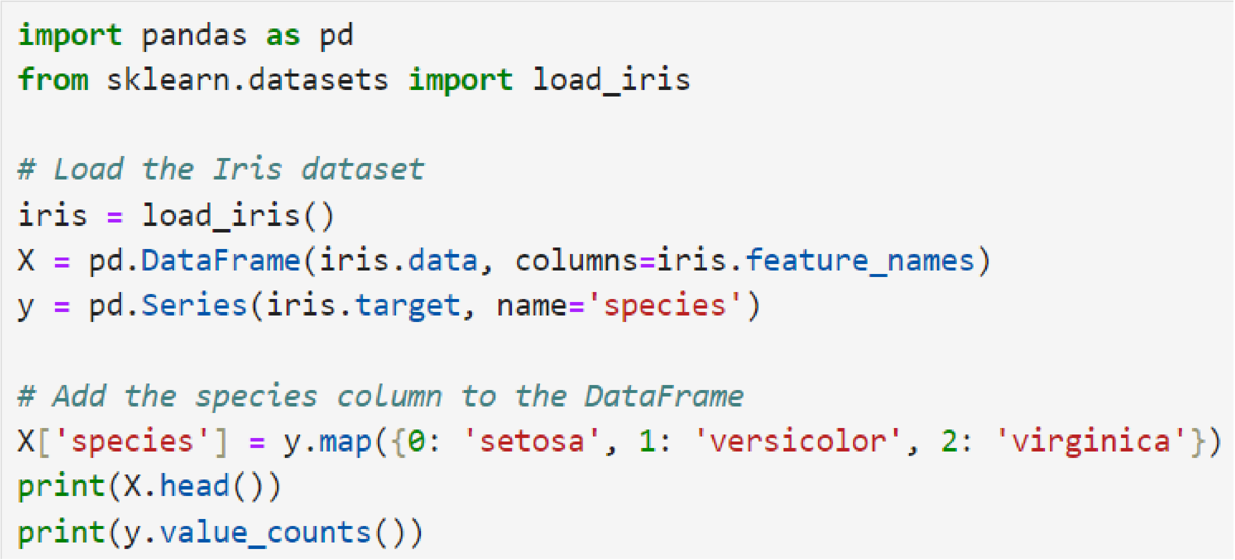
- Install the required Python packages:

pip install prince pandas seaborn matplotlib scikit-learn

Step 2: Load the Iris Dataset

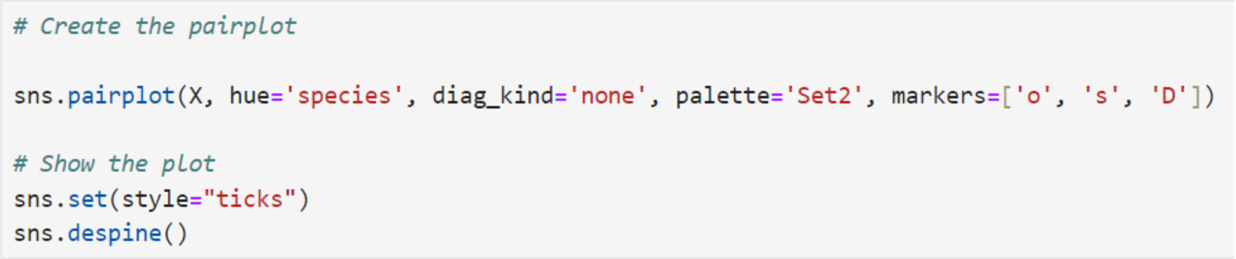
Step 3: Data Preprocessing

- Briefly explore the dataset:



```python
import pandas as pd
from sklearn.datasets import load_iris

# Load the Iris dataset
iris = load_iris()
X = pd.DataFrame(iris.data, columns=iris.feature_names)
y = pd.Series(iris.target, name='species')

# Add the species column to the DataFrame
X['species'] = y.map({0: 'setosa', 1: 'versicolor', 2: 'virginica'})
print(X.head())
print(y.value_counts())
```

- Exploratory data analysis: using a pairplot:



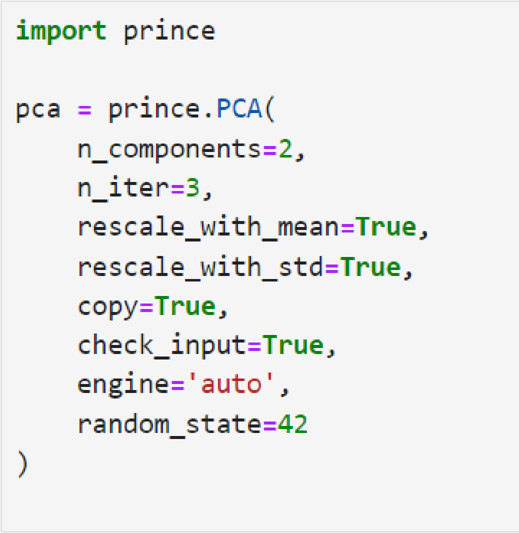
```python
# Create the pairplot

sns.pairplot(X, hue='species', diag_kind='none', palette='Set2', markers=['o', 's', 'D'])

# Show the plot
sns.set(style="ticks")
sns.despine()
```

Part 3: Performing PCA using prince

Time: 30 minutes

Step 1: Initialize the PCA Model

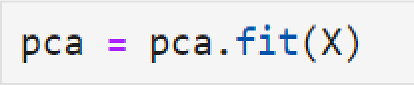
- Initialize PCA with 2 components:



```python
import prince

pca = prince.PCA(
    n_components=2,
    n_iter=3,
    rescale_with_mean=True,
    rescale_with_std=True,
    copy=True,
    check_input=True,
    engine='auto',
    random_state=42
)
```

Step 2: Fit the PCA Model

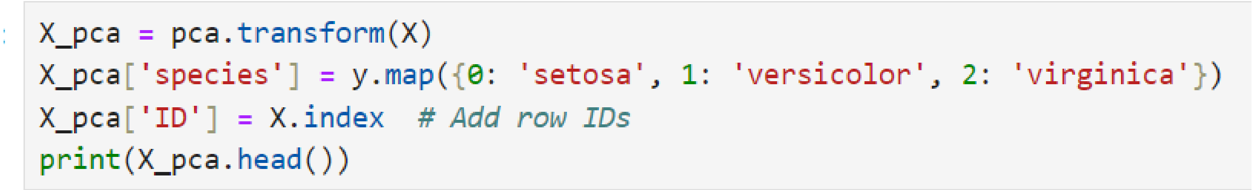
- Fit the PCA model to the Iris dataset:



```python
pca = pca.fit(X)
```

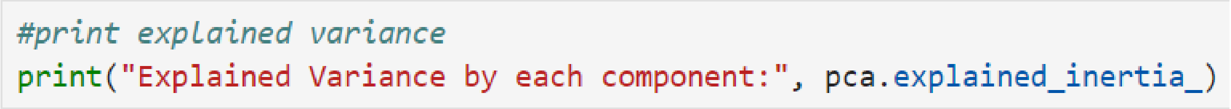
Step 3: Extract Principal Components and Explained Variance

- Transform the data to principal components:



```python
X_pca = pca.transform(X)
X_pca['species'] = y.map({0: 'setosa', 1: 'versicolor', 2: 'virginica'})
X_pca['ID'] = X.index  # Add row IDs
print(X_pca.head())
```

- Print explained variance:



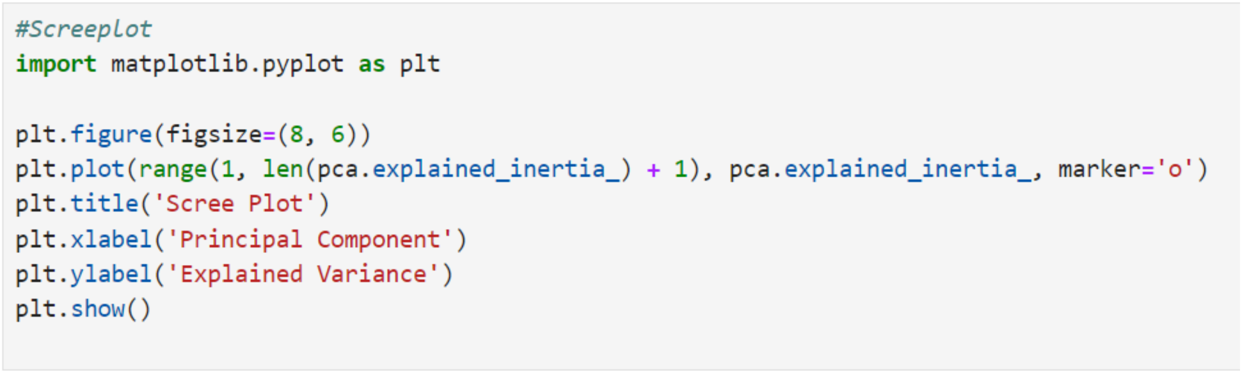
```python
#print explained variance
print("Explained Variance by each component:", pca.explained_inertia_)
```

Part 4: Visualization

Time: 60 minutes

Step 1: Scree Plot

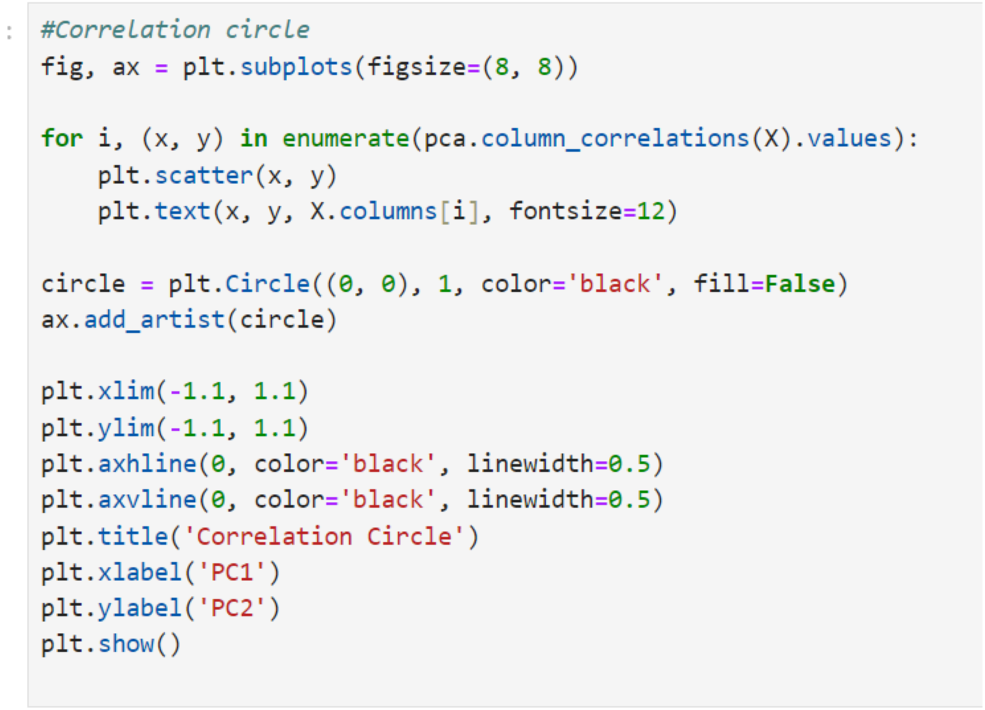
- **Objective**: To visualize the explained variance by each principal component.
- **Code**:



```
#Screeplot
import matplotlib.pyplot as plt

plt.figure(figsize=(8, 6))
plt.plot(range(1, len(pca.explained_inertia_) + 1), pca.explained_inertia_, marker='o')
plt.title('Scree Plot')
plt.xlabel('Principal Component')
plt.ylabel('Explained Variance')
plt.show()
```

Step 2: Correlation Circle

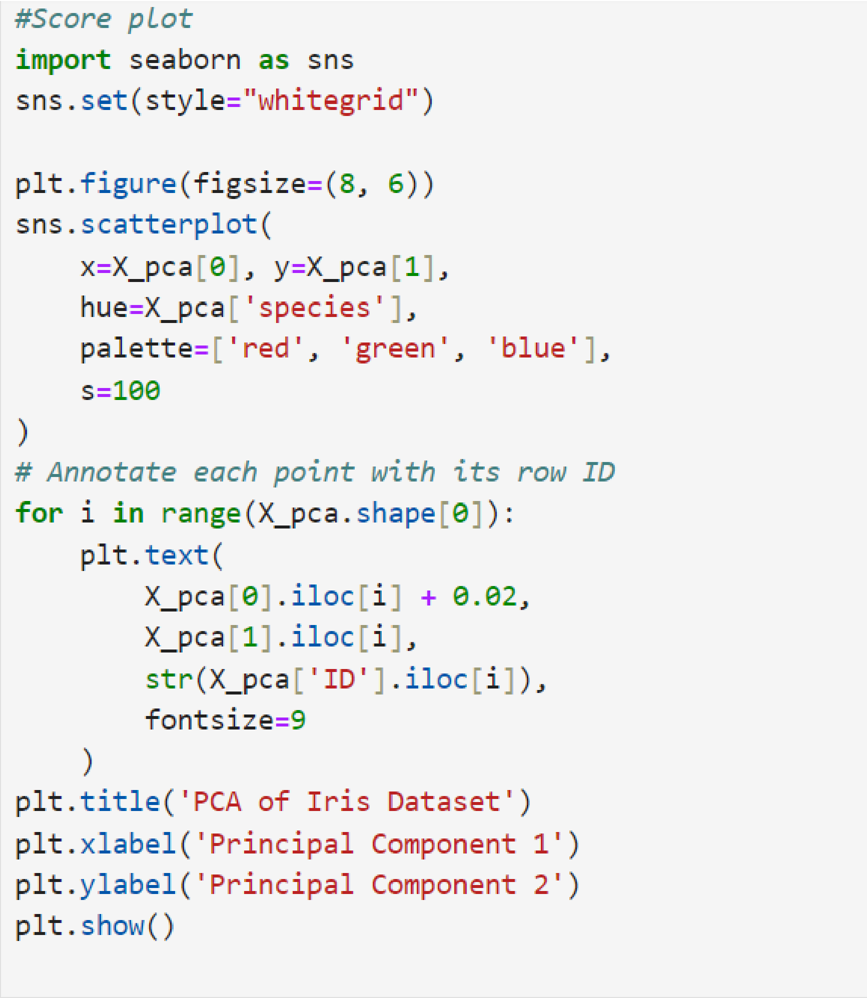
- **Objective**: To visualize how the original features correlate with the principal components.
- **Code**:



```
#Correlation circle
fig, ax = plt.subplots(figsize=(8, 8))

for i, (x, y) in enumerate(pca.column_correlations(X).values):
    plt.scatter(x, y)
    plt.text(x, y, X.columns[i], fontsize=12)

circle = plt.Circle((0, 0), 1, color='black', fill=False)
ax.add_artist(circle)

plt.xlim(-1.1, 1.1)
plt.ylim(-1.1, 1.1)
plt.axhline(0, color='black', linewidth=0.5)
plt.axvline(0, color='black', linewidth=0.5)
plt.title('Correlation Circle')
plt.xlabel('PC1')
plt.ylabel('PC2')
plt.show()
```

Step 3: Score Plot

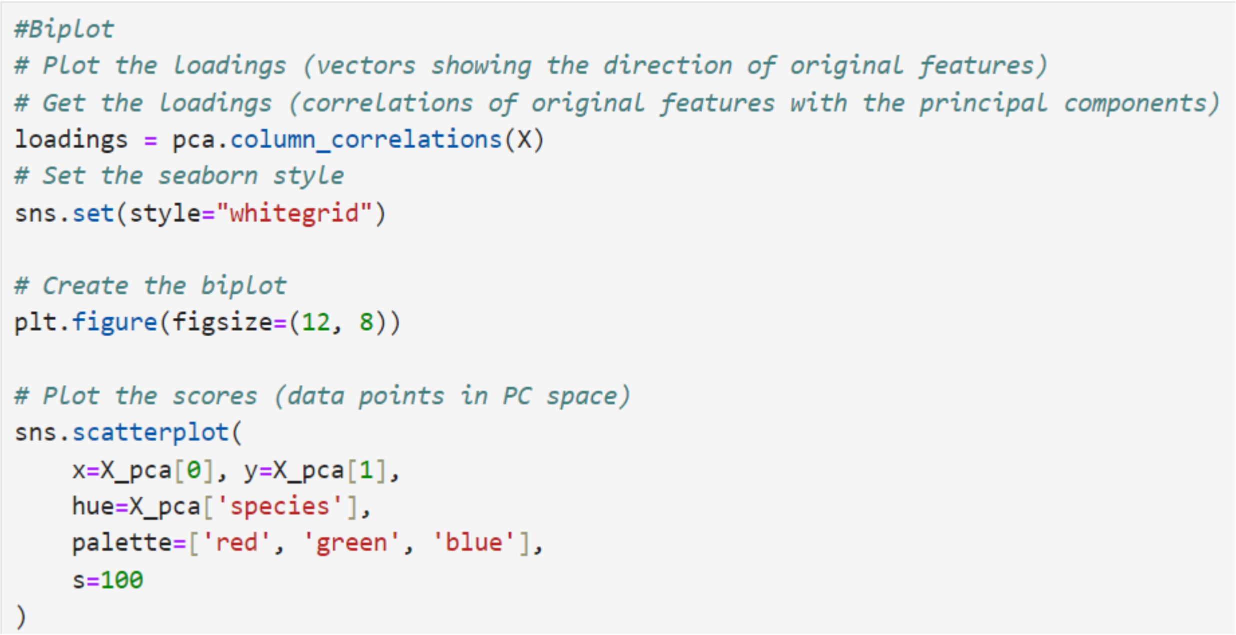
- **Objective**: To visualize the data in the principal component space.
- **Code**:



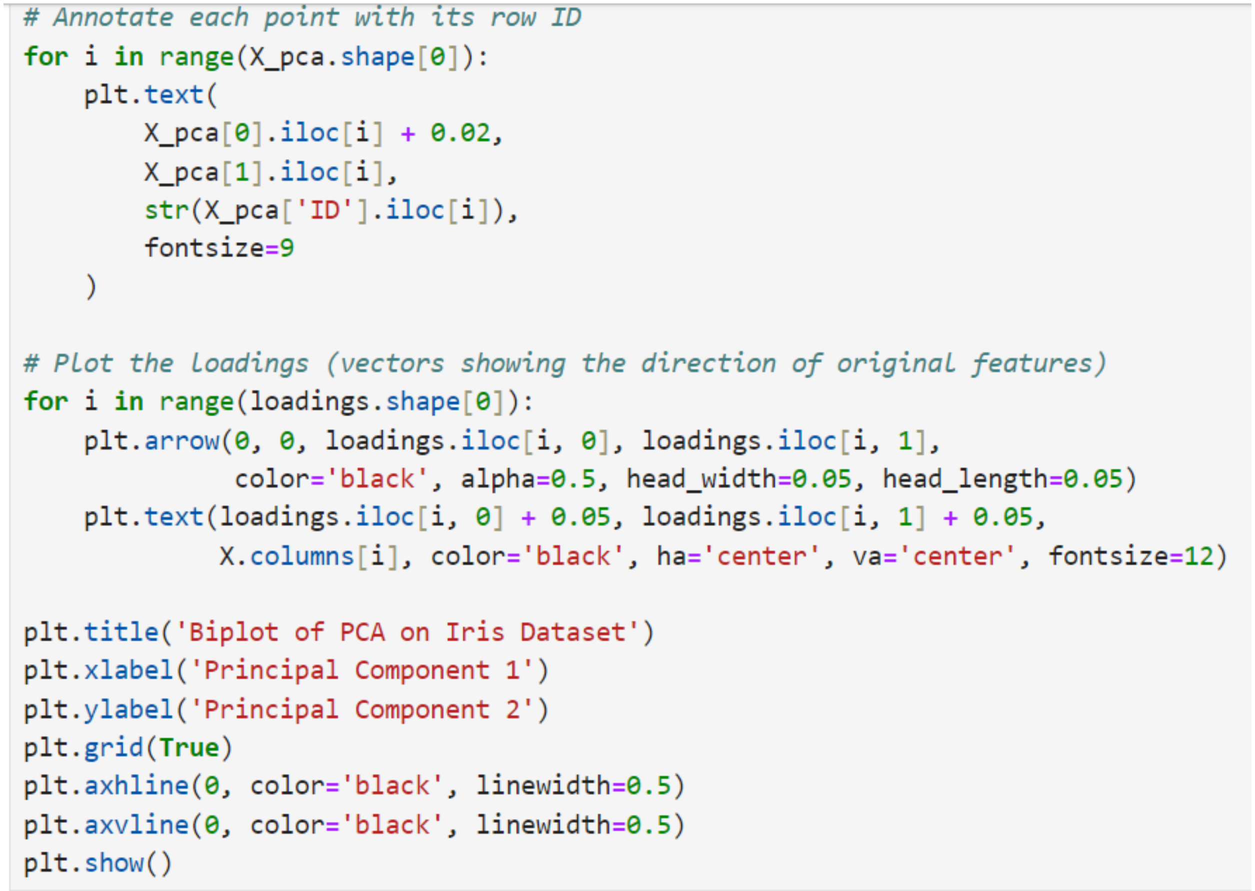
```
#Score plot
import seaborn as sns
sns.set(style="whitegrid")

plt.figure(figsize=(8, 6))
sns.scatterplot(
    x=X_pca[0], y=X_pca[1],
    hue=X_pca['species'],
    palette=['red', 'green', 'blue'],
    s=100
)
# Annotate each point with its row ID
for i in range(X_pca.shape[0]):
    plt.text(
        X_pca[0].iloc[i] + 0.02,
        X_pca[1].iloc[i],
        str(X_pca['ID'].iloc[i]),
        fontsize=9
    )
plt.title('PCA of Iris Dataset')
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.show()
```

Step 4: Biplot

- **Objective**: To visualize the scores of individuals/data (rows) and loadings of variable/features (columns) in the new principal component space.
- **Code**:



```
#Biplot
# Plot the loadings (vectors showing the direction of original features)
# Get the loadings (correlations of original features with the principal components)
loadings = pca.column_correlations(X)
# Set the seaborn style
sns.set(style="whitegrid")

# Create the biplot
plt.figure(figsize=(12, 8))

# Plot the scores (data points in PC space)
sns.scatterplot(
    x=X_pca[0], y=X_pca[1],
    hue=X_pca['species'],
    palette=['red', 'green', 'blue'],
    s=100
)
```



```
# Annotate each point with its row ID
for i in range(X_pca.shape[0]):
    plt.text(
        X_pca[0].iloc[i] + 0.02,
        X_pca[1].iloc[i],
        str(X_pca['ID'].iloc[i]),
        fontsize=9
    )

# Plot the loadings (vectors showing the direction of original features)
for i in range(loadings.shape[0]):
    plt.arrow(0, 0, loadings.iloc[i, 0], loadings.iloc[i, 1],
              color='black', alpha=0.5, head_width=0.05, head_length=0.05)
    plt.text(loadings.iloc[i, 0] + 0.05, loadings.iloc[i, 1] + 0.05,
             X.columns[i], color='black', ha='center', va='center', fontsize=12)

plt.title('Biplot of PCA on Iris Dataset')
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.grid(True)
plt.axhline(0, color='black', linewidth=0.5)
plt.axvline(0, color='black', linewidth=0.5)
plt.show()
```

Part 5: Interpretation and Analysis

Time: 30 minutes

Scree Plot Interpretation

- Discuss how much variance is explained by the first two components and the importance of higher components.

Correlation Circle Interpretation

- Discuss how each feature contributes to the principal components and what the circle tells us about feature correlations.

Score Plot Interpretation

- Discuss how the different species are grouped in the principal component space and what this tells us about the separability of the species.

Part 6: Conclusion and Q&A

Time: 30 minutes

Summary

- Recap the steps of performing PCA using prince.
- Review the insights gained from each of the visualizations.

Q&A

- Open the floor for questions.
- Discuss any challenges faced during the lab.
- Explore possible extensions or applications of PCA in other datasets.

Additional Resources

- **Documentation**: Prince Documentation
- **Further Reading**: Articles on PCA and dimensionality reduction techniques.