Exploring Patterns of Crime and Urbanization in the United States: A Principal Component and Clustering Analysis of the US Arrests Dataset

A Punchline report – Durga Prasad N

The Issues:

This report presents the results of a data analysis project that aims to identify patterns and clusters in the US Arrests dataset, which contains information on the rates of violent crime, murder, rape, and assault in different US states, as well as the percentage of the population living in urban areas. The report includes a principal component analysis to identify the main dimensions of variation in the data, a k-means clustering analysis to identify groups of states with similar characteristics, and a hierarchical clustering analysis to visualize the relationships between these groups.

Findings:

The principal component analysis revealed that the US Arrests dataset can be summarized using just two principal components, which represent overall violent crime rates and urbanization. The k-means clustering identified three clusters in the data, representing areas with low, moderate, and high levels of overall violent crime rates and urbanization. The hierarchical clustering also identified three distinct clusters, representing similar patterns in the data. In summary, PCA helps to reduce the dimensionality of the data and allows for easier visualization, while k-means and hierarchical clustering provide different approaches to partition the data into groups based on similarity. By comparing the results of these methods, you can gain insights into the relationships between states and their attributes, as well as the performance of different clustering algorithms on the given dataset.

Discussion:

These findings suggest that there are strong relationships between violent crime rates and urbanization in the United States, and that different states can be grouped into clusters based on these characteristics. The implications of these findings for policymakers and law enforcement agencies are discussed, including the potential use of targeted interventions and resource deployment in areas with different levels of crime and urbanization.

Appendix A: Methods

Data collection: The US Arrests dataset was obtained from the MASS library in R and consists of data on 50 US states and the District of Columbia.

Variable creation: The data set contains 5 variables. The predictor variables are State, Murder, Assault, urbanpop, rape.

Analytic methods:

The dataset is loaded into the pandas dataframe. The data was analyzed using principal component analysis, k-means clustering, and hierarchical clustering in Python, using the scikit-learn library. We performed a principal component analysis (PCA) on the USArrests dataset. The dataset contains four variables, namely, Murder, Assault, Rape, and UrbanPop, for 50 US states. We standardized the data using StandardScaler from the scikit-learn library and applied PCA using PCA from the same library. The eigenvalues and eigenvectors were obtained from the PCA.

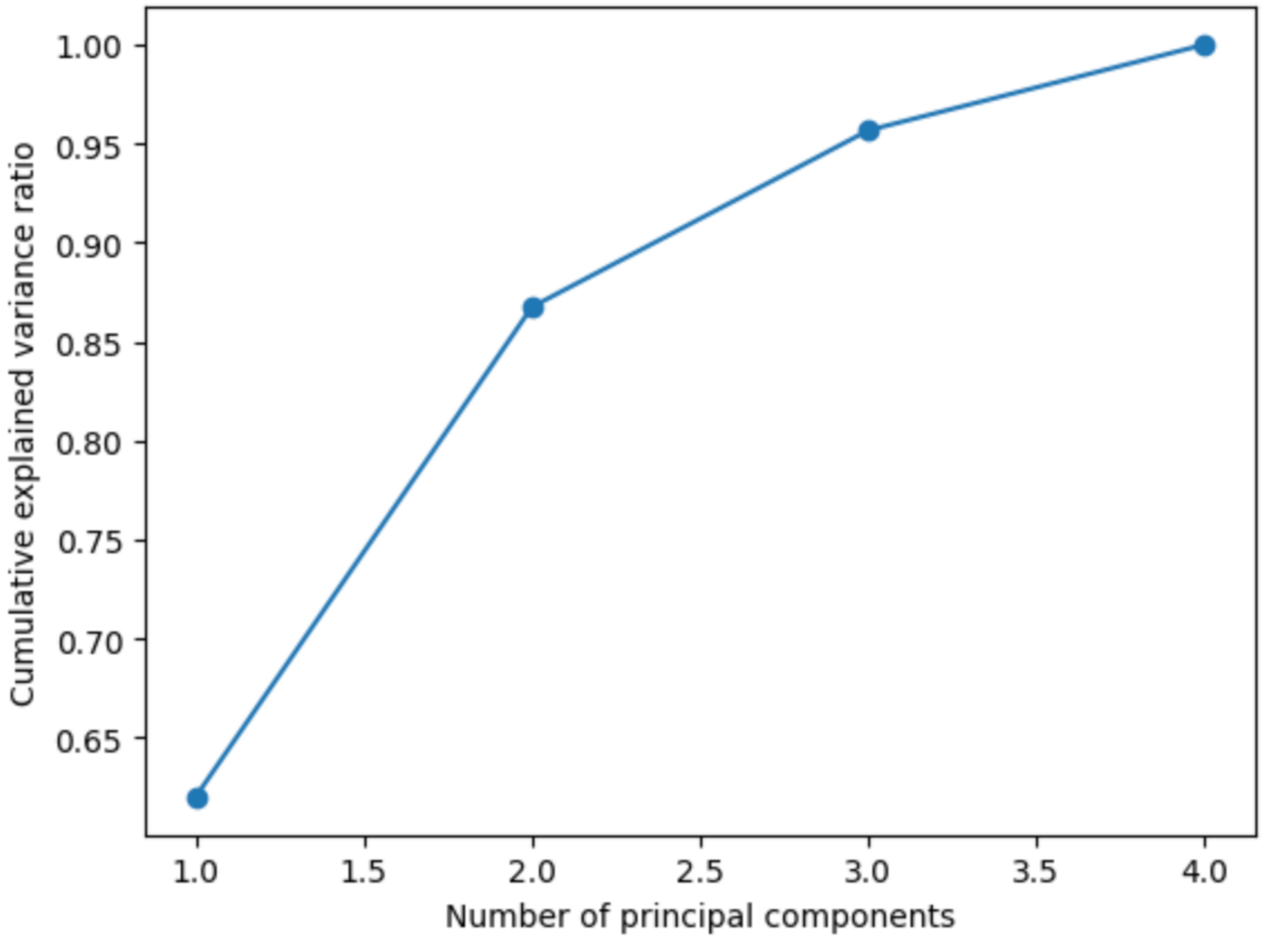
The eigenvalues represent the variance explained by each principal component (PC). The cumulative explained variance ratio plot shows that the first two principal components explain most of the variance in the dataset. PC1 explains 2.53 units of variance, while PC2 explains 1.01 units of variance. The remaining principal components (PC3 and PC4) explain very little of the variance in the data, with 0.36 and 0.18 units of variance, respectively. The eigenvectors represent the loadings of each variable onto each principal component. PC1 has high loadings on all four variables, with Murder and Assault having the highest loadings (0.54 and 0.58, respectively). PC2 has high loadings on UrbanPop and Rape, with loadings of -0.87 and -0.17, respectively. PC3 has the highest loading on UrbanPop (-0.38), while PC4 has the highest loading on Murder (0.65).

And we applied the elbow method to determine the optimal number of clusters. The plot shows that the elbow point occurs at k=2, indicating that two clusters are suitable for the dataset. Finally, We used the ward linkage method to construct the dendrogram, which shows the similarity between the states. The dendrogram shows that the states can be grouped into two main clusters, with some sub-clusters within each of the two clusters.

Appendix B: Results

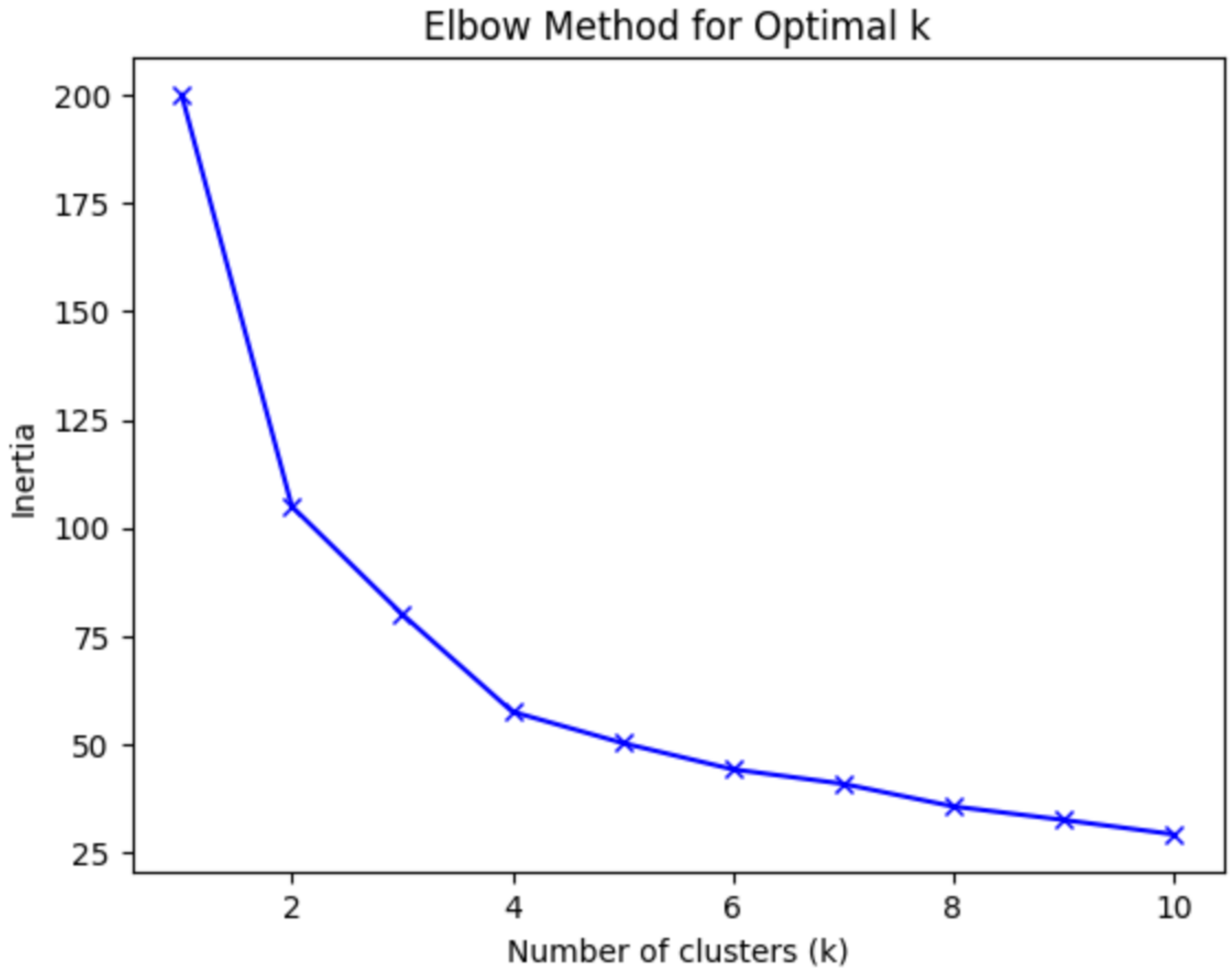
The dataset contains 50 records and 5 columns. The PCA results are as follows.

```
Eigenvalues:
                                   PC1: 2.53
                                   PC2: 1.01
                                   PC3: 0.36
                                   PC4: 0.18
```

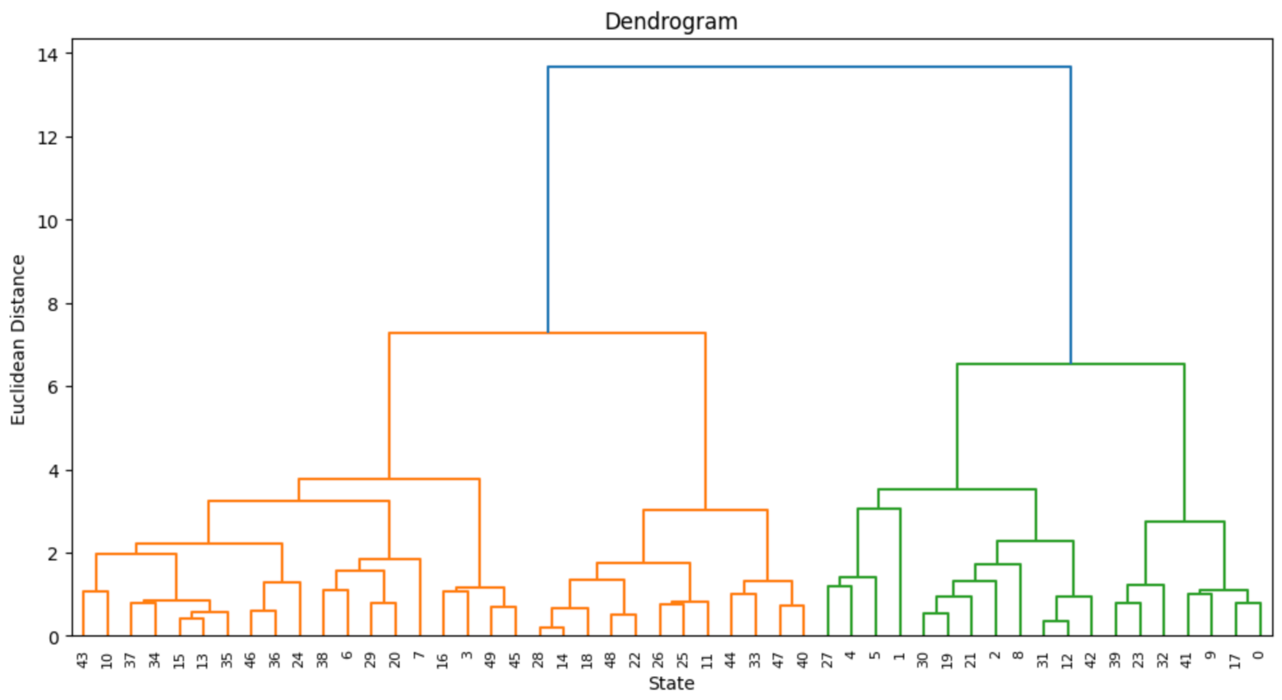


```
Eigenvectors:
                          PC1: 0.54, 0.58, 0.28, 0.54
                          PC2: 0.42, 0.19, -0.87, -0.17
                          PC3: -0.34, -0.27, -0.38, 0.82
                          PC4: 0.65, -0.74, 0.13, 0.09
```

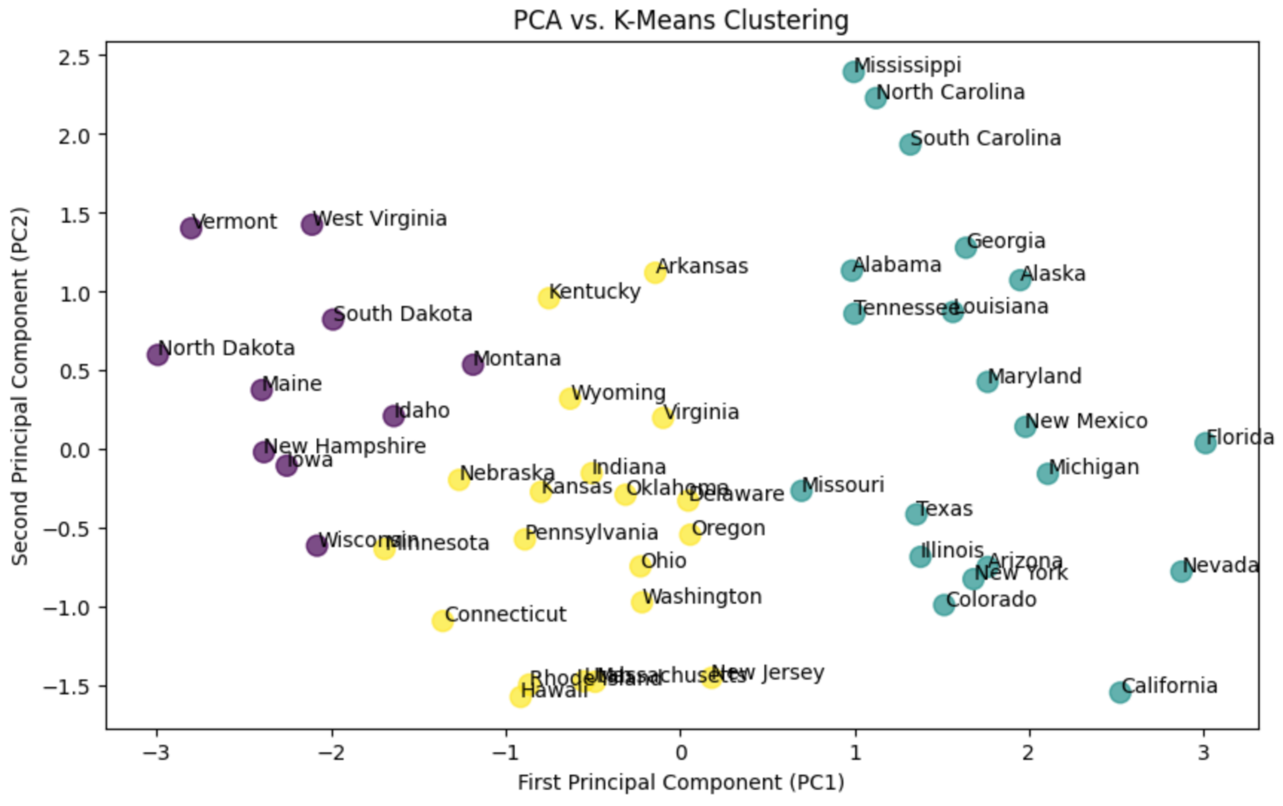
Performed a K-means clustering on the same dataset and the elbow plot is shown below.

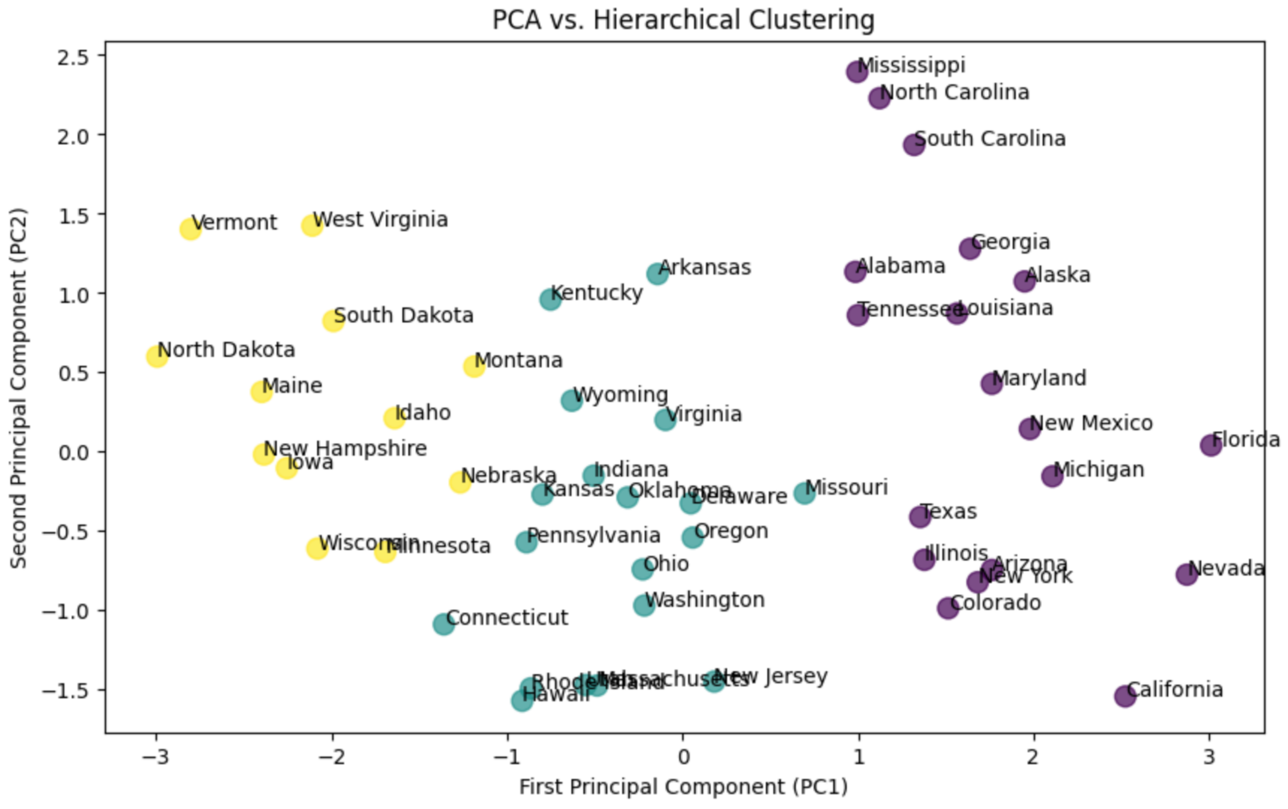


Finally, performed Hierarchical clustering on the data set and the dendogram is shown below.



On further, we performed a comparison between K-means, hierarchical clustering with first two principal components. These scatter plots show the relationships between the states, clusters, and principal components. The color of the points represents the cluster, while the position on the x and y axes corresponds to the first and second principal components, respectively.
By comparing the scatter plots, you can observe the similarities and differences in the clustering patterns obtained from k-means and hierarchical clustering. In both cases, the clustering methods aim to group states with similar characteristics.

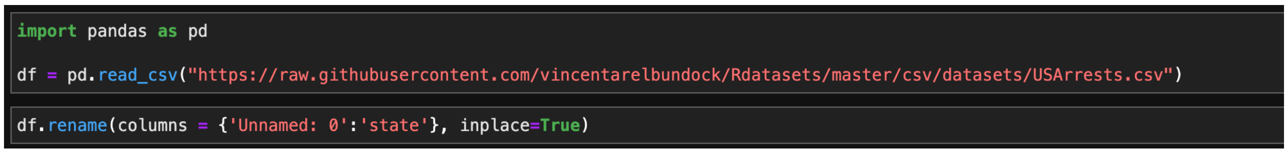




Appendix C: Code

The clustering analysis is performed using following code using sklearn package of python.

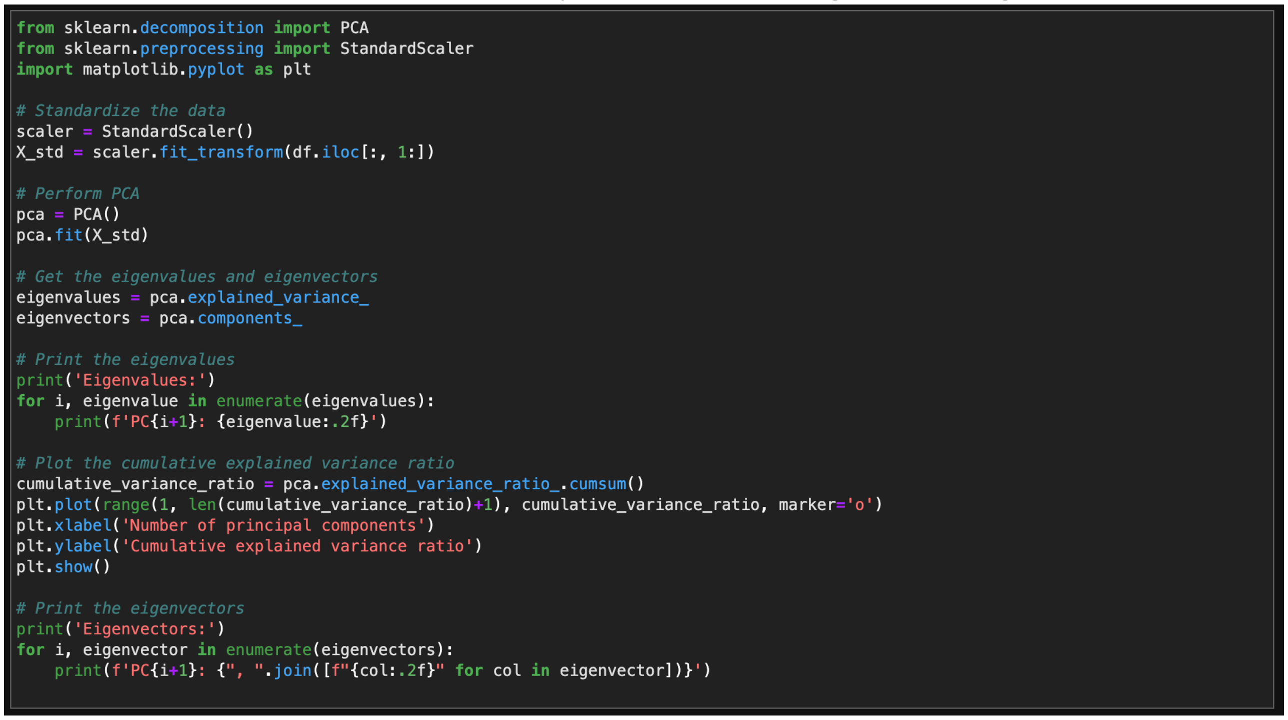
a. Importing required libraries and loading the dataset into pandas dataframe



```python
import pandas as pd

df = pd.read_csv("https://raw.githubusercontent.com/vincentarelbundock/Rdatasets/master/csv/datasets/USArrests.csv")
```

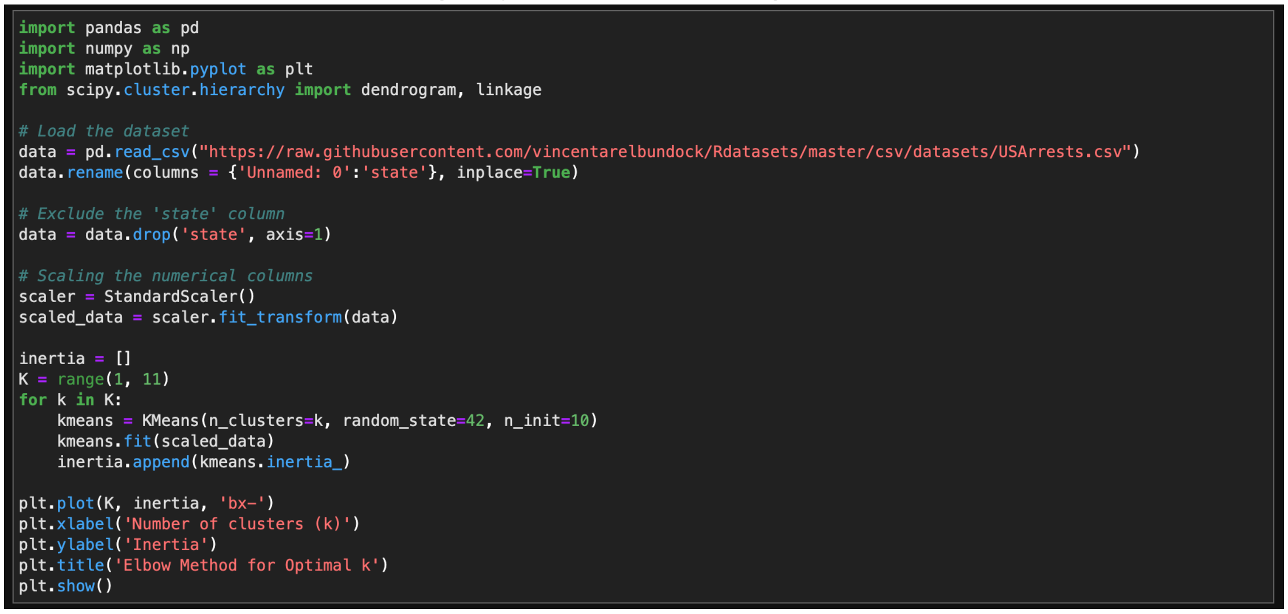
```python
df.rename(columns = {'Unnamed: 0':'state'}, inplace=True)
```

b. The data is scaled and PCA is performed using following code.



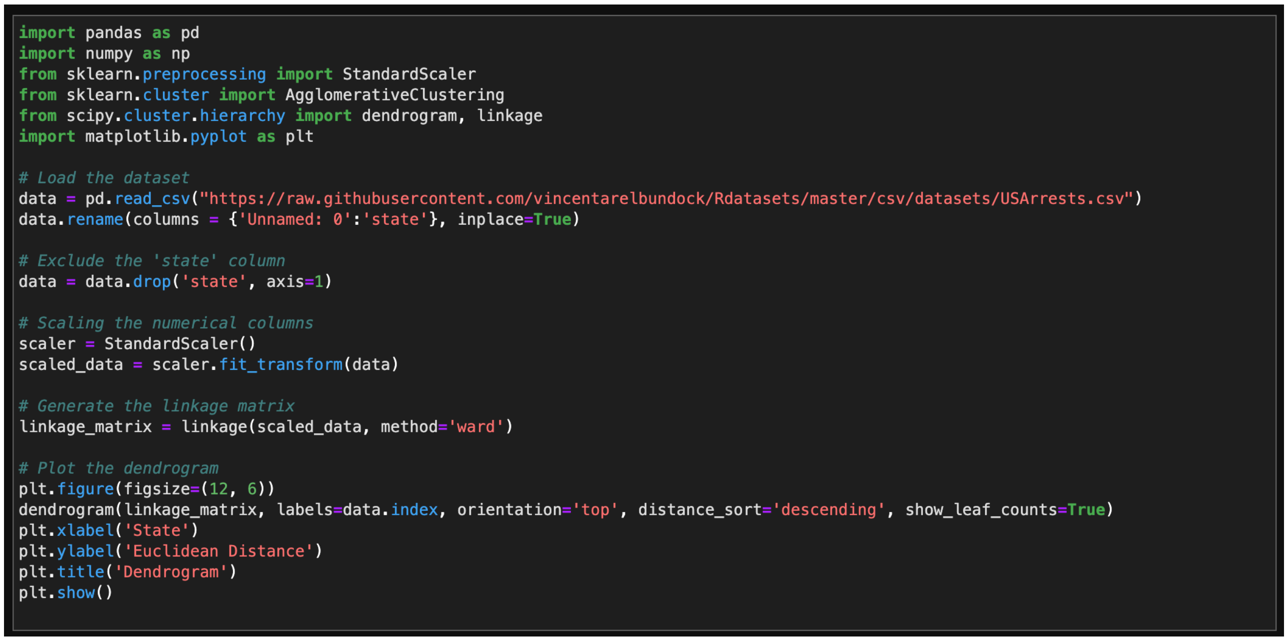
```python
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt

# Standardize the data
scaler = StandardScaler()
X_std = scaler.fit_transform(df.iloc[:, 1:])

# Perform PCA
pca = PCA()
pca.fit(X_std)

# Get the eigenvalues and eigenvectors
eigenvalues = pca.explained_variance_
eigenvectors = pca.components_

# Print the eigenvalues
print('Eigenvalues:')
for i, eigenvalue in enumerate(eigenvalues):
    print(f'PC{i+1}: {eigenvalue:.2f}')

# Plot the cumulative explained variance ratio
cumulative_variance_ratio = pca.explained_variance_ratio_.cumsum()
plt.plot(range(1, len(cumulative_variance_ratio)+1), cumulative_variance_ratio, marker='o')
plt.xlabel('Number of principal components')
plt.ylabel('Cumulative explained variance ratio')
plt.show()

# Print the eigenvectors
print('Eigenvectors:')
for i, eigenvector in enumerate(eigenvectors):
    print(f'PC{i+1}: {", ".join([f"{col:.2f}" for col in eigenvector])}')
```

c. The k-means clustering is performed using below code



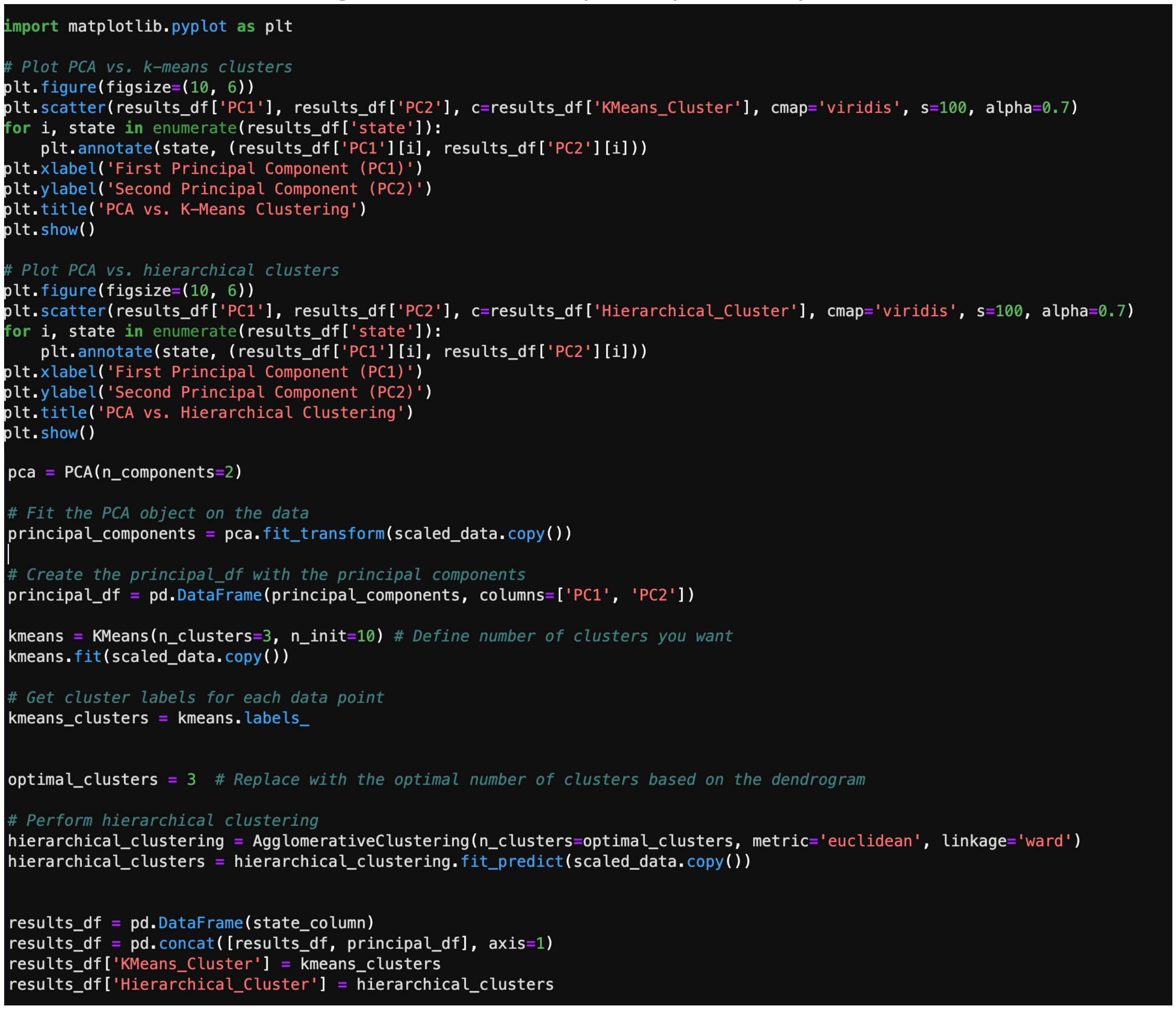
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.cluster.hierarchy import dendrogram, linkage

# Load the dataset
data = pd.read_csv("https://raw.githubusercontent.com/vincentarelbundock/Rdatasets/master/csv/datasets/USArrests.csv")
data.rename(columns = {'Unnamed: 0':'state'}, inplace=True)

# Exclude the 'state' column
data = data.drop('state', axis=1)

# Scaling the numerical columns
scaler = StandardScaler()
scaled_data = scaler.fit_transform(data)

inertia = []
K = range(1, 11)
for k in K:
    kmeans = KMeans(n_clusters=k, random_state=42, n_init=10)
    kmeans.fit(scaled_data)
    inertia.append(kmeans.inertia_)

plt.plot(K, inertia, 'bx-')
plt.xlabel('Number of clusters (k)')
plt.ylabel('Inertia')
plt.title('Elbow Method for Optimal k')
plt.show()
```

d. Hierarchical clustering is using below code.



```python
import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import AgglomerativeClustering
from scipy.cluster.hierarchy import dendrogram, linkage
import matplotlib.pyplot as plt

# Load the dataset
data = pd.read_csv("https://raw.githubusercontent.com/vincentarelbundock/Rdatasets/master/csv/datasets/USArrests.csv")
data.rename(columns = {'Unnamed: 0':'state'}, inplace=True)

# Exclude the 'state' column
data = data.drop('state', axis=1)

# Scaling the numerical columns
scaler = StandardScaler()
scaled_data = scaler.fit_transform(data)

# Generate the linkage matrix
linkage_matrix = linkage(scaled_data, method='ward')

# Plot the dendrogram
plt.figure(figsize=(12, 6))
dendrogram(linkage_matrix, labels=data.index, orientation='top', distance_sort='descending', show_leaf_counts=True)
plt.xlabel('State')
plt.ylabel('Euclidean Distance')
plt.title('Dendrogram')
plt.show()
```

e. Additional code for comparison between K-means clustering and hierarchical clustering with first two principle components.



```python
import matplotlib.pyplot as plt

# Plot PCA vs. k-means clusters
plt.figure(figsize=(10, 6))
plt.scatter(results_df['PC1'], results_df['PC2'], c=results_df['KMeans_Cluster'], cmap='viridis', s=100, alpha=0.7)
for i, state in enumerate(results_df['state']):
    plt.annotate(state, (results_df['PC1'][i], results_df['PC2'][i]))
plt.xlabel('First Principal Component (PC1)')
plt.ylabel('Second Principal Component (PC2)')
plt.title('PCA vs. K-Means Clustering')
plt.show()

# Plot PCA vs. hierarchical clusters
plt.figure(figsize=(10, 6))
plt.scatter(results_df['PC1'], results_df['PC2'], c=results_df['Hierarchical_Cluster'], cmap='viridis', s=100, alpha=0.7)
for i, state in enumerate(results_df['state']):
    plt.annotate(state, (results_df['PC1'][i], results_df['PC2'][i]))
plt.xlabel('First Principal Component (PC1)')
plt.ylabel('Second Principal Component (PC2)')
plt.title('PCA vs. Hierarchical Clustering')
plt.show()

pca = PCA(n_components=2)

# Fit the PCA object on the data
principal_components = pca.fit_transform(scaled_data.copy())

# Create the principal_df with the principal components
principal_df = pd.DataFrame(principal_components, columns=['PC1', 'PC2'])

kmeans = KMeans(n_clusters=3, n_init=10) # Define number of clusters you want
kmeans.fit(scaled_data.copy())

# Get cluster labels for each data point
kmeans_clusters = kmeans.labels_


optimal_clusters = 3  # Replace with the optimal number of clusters based on the dendrogram

# Perform hierarchical clustering
hierarchical_clustering = AgglomerativeClustering(n_clusters=optimal_clusters, metric='euclidean', linkage='ward')
hierarchical_clusters = hierarchical_clustering.fit_predict(scaled_data.copy())


results_df = pd.DataFrame(state_column)
results_df = pd.concat([results_df, principal_df], axis=1)
results_df['KMeans_Cluster'] = kmeans_clusters
results_df['Hierarchical_Cluster'] = hierarchical_clusters
```