```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.preprocessing import StandardScaler

from sklearn.cluster import KMeans

from sklearn.decomposition import PCA


# Sample data: Replace with your actual dataset

df = pd.read_csv('tourism_data.csv')


# Quick overview

print(df.head())

print(df.info())

# Drop rows with missing values

df.dropna(inplace=True)


# Convert date columns (if applicable)

if 'Date' in df.columns:

    df['Date'] = pd.to_datetime(df['Date'])


# Select relevant features (e.g., number of tourists, spending, duration)

features = ['Tourist_Count', 'Avg_Stay_Days', 'Spending_per_Visitor']

data = df[features]


# Normalize features

scaler = StandardScaler()

scaled_data = scaler.fit_transform(data)
```

```
# Determine optimal number of clusters using Elbow Method
sse = []
for k in range(1, 11):
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(scaled_data)
    sse.append(kmeans.inertia_)

# Plot the Elbow Curve
plt.plot(range(1, 11), sse, marker='o')
plt.xlabel('Number of clusters')
plt.ylabel('SSE')
plt.title('Elbow Method for Optimal k')
plt.show()

# Apply KMeans with chosen number of clusters (e.g., 3)
kmeans = KMeans(n_clusters=3, random_state=42)
df['Cluster'] = kmeans.fit_predict(scaled_data)
# PCA for 2D visualization
pca = PCA(n_components=2)
pca_data = pca.fit_transform(scaled_data)

# Plot
plt.figure(figsize=(10, 6))
sns.scatterplot(x=pca_data[:, 0], y=pca_data[:, 1], hue=df['Cluster'], palette='Set2')
plt.title('Tourism Data Clusters (PCA Projection)')
plt.xlabel('PC1')
plt.ylabel('PC2')
plt.legend()
plt.show()
```