# Importing necessary libraries

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.linear\_model import LinearRegression

from sklearn.cluster import KMeans

from sklearn.preprocessing import StandardScaler

from sklearn.metrics import mean\_squared\_error, r2\_score

# Example Dataset Creation

# Sample data for demonstration

data = {

'Year': [2000, 2005, 2010, 2015, 2020],

'Population Growth Rate (%)': [1.2, 1.1, 1.05, 1.03, 0.98],

'Urbanization Rate (%)': [45.3, 48.7, 52.1, 55.8, 58.4],

'GDP Growth Rate (%)': [4.5, 4.8, 5.2, 3.7, 2.9]

}

# Convert to DataFrame

df = pd.DataFrame(data)

# Display Dataset

print("Dataset:")

print(df)

# Data Visualization

plt.figure(figsize=(10, 6))

sns.lineplot(x='Year', y='Population Growth Rate (%)', data=df, label='Population Growth Rate')

sns.lineplot(x='Year', y='Urbanization Rate (%)', data=df, label='Urbanization Rate')

sns.lineplot(x='Year', y='GDP Growth Rate (%)', data=df, label='GDP Growth Rate')

plt.title("Trends in Population, Urbanization, and GDP Growth Rates")

plt.xlabel("Year")

plt.ylabel("Rate (%)")

plt.legend()

plt.show()

# Linear Regression to Analyze Correlations

X = df[['Population Growth Rate (%)', 'Urbanization Rate (%)']]

y = df['GDP Growth Rate (%)']

# Train a Linear Regression Model

regressor = LinearRegression()

regressor.fit(X, y)

# Predict GDP Growth Rate using the model

y\_pred = regressor.predict(X)

# Evaluate the Model

print("\nLinear Regression Model Evaluation:")

print(f"Coefficients: {regressor.coef\_}")

print(f"Intercept: {regressor.intercept\_}")

print(f"R² Score: {r2\_score(y, y\_pred):.2f}")

print(f"Mean Squared Error: {mean\_squared\_error(y, y\_pred):.2f}")

# Clustering Analysis

# Standardize the data

scaler = StandardScaler()

scaled\_data = scaler.fit\_transform(df[['Population Growth Rate (%)', 'Urbanization Rate (%)', 'GDP Growth Rate (%)']])

# Apply KMeans Clustering

kmeans = KMeans(n\_clusters=3, random\_state=42)

clusters = kmeans.fit\_predict(scaled\_data)

df['Cluster'] = clusters

# Visualize Clustering

plt.figure(figsize=(8, 6))

sns.scatterplot(

x='Urbanization Rate (%)',

y='GDP Growth Rate (%)',

hue='Cluster',

data=df,

palette='Set1',

s=100

)

plt.title("Clustering Analysis of Population, Urbanization, and GDP Growth")

plt.xlabel("Urbanization Rate (%)")

plt.ylabel("GDP Growth Rate (%)")

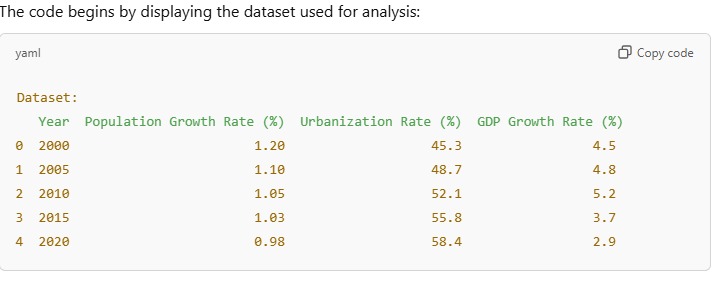
plt.legend(title="Cluster")

plt.show()

# Display Clustering Results

print("\nClustering Results:")

print(df)



A graph with a green line and blue line

Description automatically generated