act5-2

October 28, 2024

Jacob Valdenegro Monzón A01640992

[43]:
```
# Paso 1: Importar las librerías necesarias
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
import statsmodels.api as sm
from statsmodels.stats.outliers_influence import variance_inflation_factor
from sklearn.cluster import KMeans
from sklearn.pipeline import make_pipeline
from fpdf import FPDF

# Paso 2: Cargar los datos
data = pd.read_csv("Country-data.csv")
data_dict = pd.read_csv("data-dictionary.csv")

# Visualización inicial de los datos y diccionario
print(data.head())
print(data_dict)
```

```
              country  child_mort  exports  health  imports  income  \
0         Afghanistan        90.2     10.0    7.58     44.9    1610
1             Albania        16.6     28.0    6.55     48.6    9930
2             Algeria        27.3     38.4    4.17     31.4   12900
3              Angola       119.0     62.3    2.85     42.9    5900
4  Antigua and Barbuda        10.3     45.5    6.03     58.9   19100

   inflation  life_expec  total_fer   gdpp
0       9.44        56.2       5.82    553
1       4.49        76.3       1.65   4090
2      16.10        76.5       2.89   4460
3      22.40        60.1       6.16   3530
4       1.44        76.8       2.13  12200
```

```
  Column Name                                          Description
0     country                                  Name of the country
1  child_mort  Death of children under 5 years of age per 100…
2     exports  Exports of goods and services. Given as %age o…
3      health         Total health spending as %age of Total GDP
4     imports  Imports of goods and services. Given as %age o…
5      Income                               Net income per person
6   Inflation  The measurement of the annual growth rate of t…
7  life_expec  The average number of years a new born child w…
8   total_fer  The number of children that would be born to e…
9        gdpp  The GDP per capita. Calculated as the Total GD…
```

Preparación de los datos

[44]:
```
# Eliminar la columna 'country' ya que no es numérica y no se usará como␣
 ↪predictor
X = data.drop(columns=["country", "gdpp"])
y = data["gdpp"]

# Estandarizar las variables predictoras
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

Análisis de multicolinealidad

[45]:
```
# Calcular el VIF para cada predictor
vif_data = pd.DataFrame()
vif_data["Feature"] = X.columns
vif_data["VIF"] = [variance_inflation_factor(X_scaled, i) for i in␣
 ↪range(X_scaled.shape[1])]
print("Valores de VIF:", vif_data)
```

```
Valores de VIF:       Feature       VIF
0  child_mort  7.206594
1     exports  4.926676
2      health  1.367459
3     imports  3.720285
4      income  2.488358
5   inflation  1.260237
6  life_expec  5.676476
7   total_fer  3.720211
```

Interpretación de los VIF

child_mort (7.21) y life_expec (5.68) tienen valores relativamente elevados, lo que indica que podrían estar correlacionadas con otras variables en el modelo. El resto de los VIF son menores a 5, lo que implica baja colinealidad en general.

Modelo de regresión lineal múltiple inicial

[46]:
```
X_with_constant = sm.add_constant(X_scaled)  # Agregar una constante para el␣
↪modelo de regresión
model = sm.OLS(y, X_with_constant).fit()
print(model.summary())

# Interpretación de los resultados de regresión y VIF
# Ver p-values, coeficientes y diagnósticos de residuales
residuals = model.resid
fig, ax = plt.subplots(1, 2, figsize=(15, 5))
sns.residplot(x=model.fittedvalues, y=residuals, lowess=True, ax=ax[0],␣
↪line_kws={'color': 'red'})
ax[0].set_title("Residuales vs Ajustados")
sm.qqplot(residuals, line='45', ax=ax[1])
ax[1].set_title("Gráfica Q-Q de los Residuales")
plt.show()
```

OLS Regression Results

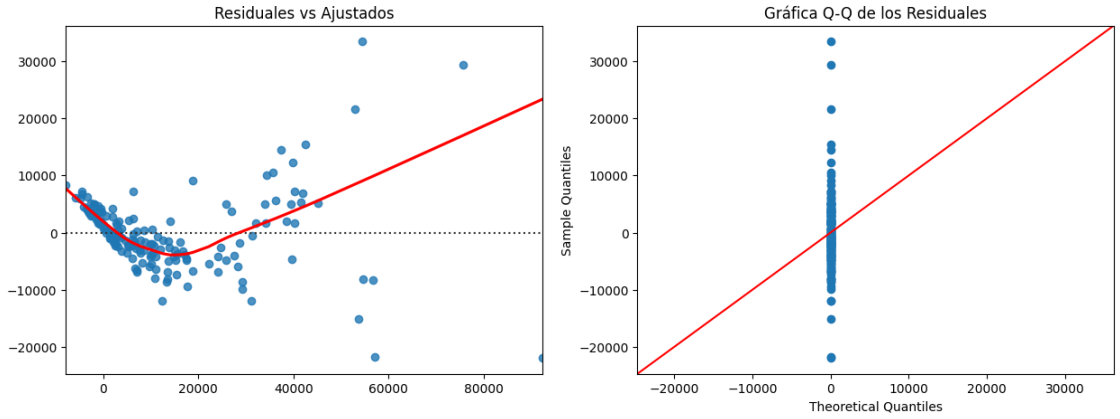
Dep. Variable:	gdpp	R-squared:	0.866
Model:	OLS	Adj. R-squared:	0.859
Method:	Least Squares	F-statistic:	127.7
Date:	Mon, 28 Oct 2024	Prob (F-statistic):	6.13e-65
Time:	05:44:42	Log-Likelihood:	-1707.9
No. Observations:	167	AIC:	3434.
Df Residuals:	158	BIC:	3462.
Df Model:	8		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	1.296e+04	532.047	24.367	0.000	1.19e+04	1.4e+04
x1	2676.5161	1428.286	1.874	0.063	-144.481	5497.513
x2	778.5286	1180.939	0.659	0.511	-1553.934	3110.991
x3	4241.5150	622.168	6.817	0.000	3012.677	5470.353
x4	-678.7091	1026.215	-0.661	0.509	-2705.579	1348.160
x5	1.51e+04	839.280	17.990	0.000	1.34e+04	1.68e+04
x6	-1059.1469	597.278	-1.773	0.078	-2238.825	120.532
x7	3448.6094	1267.622	2.721	0.007	944.940	5952.279
x8	928.3608	1026.205	0.905	0.367	-1098.488	2955.210

Omnibus:	53.684	Durbin-Watson:	1.914
Prob(Omnibus):	0.000	Jarque-Bera (JB):	287.333
Skew:	1.040	Prob(JB):	4.04e-63
Kurtosis:	9.080	Cond. No.	6.48

Notes:

```
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
```



Coeficientes y p-values

Las variables health, income y life_expec son las más relevantes para explicar el PIB per cápita (gdpp), con relaciones significativas y positivas.

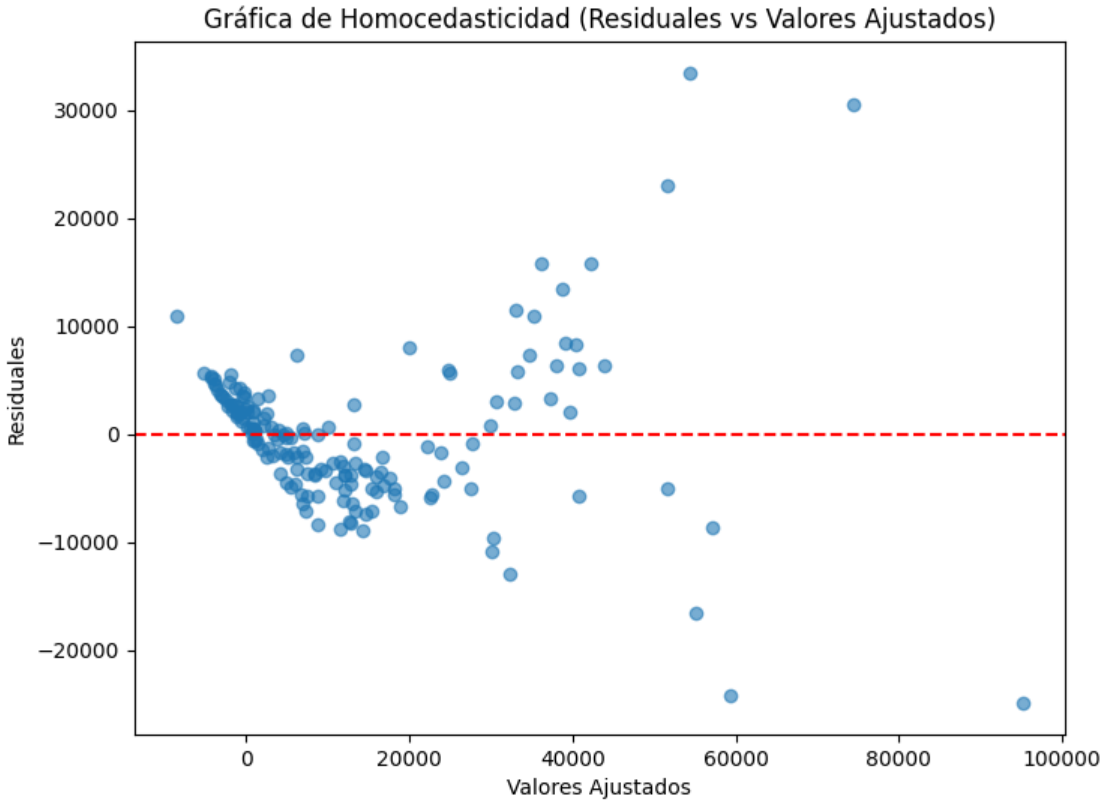
Otras variables, como child_mort, inflation, y total_fer, muestran relaciones más débiles o no significativas.

Supuestos del Modelo de Regresión Lineal

Linealidad: La relación entre las variables predictoras y la respuesta es aproximadamente lineal.

Independencia de los errores: El estadístico de Durbin-Watson es 1.914, cercano a 2, lo que indica que los errores son independientes.

[50]:
```
model = sm.OLS(y, X).fit()
residuals = model.resid
fitted_values = model.fittedvalues

# Gráfica de homocedasticidad (Residuales vs Valores Ajustados)
plt.figure(figsize=(8, 6))
plt.scatter(fitted_values, residuals, alpha=0.6)
plt.axhline(0, color='red', linestyle='--')
plt.xlabel('Valores Ajustados')
plt.ylabel('Residuales')
plt.title('Gráfica de Homocedasticidad (Residuales vs Valores Ajustados)')
plt.show()
```

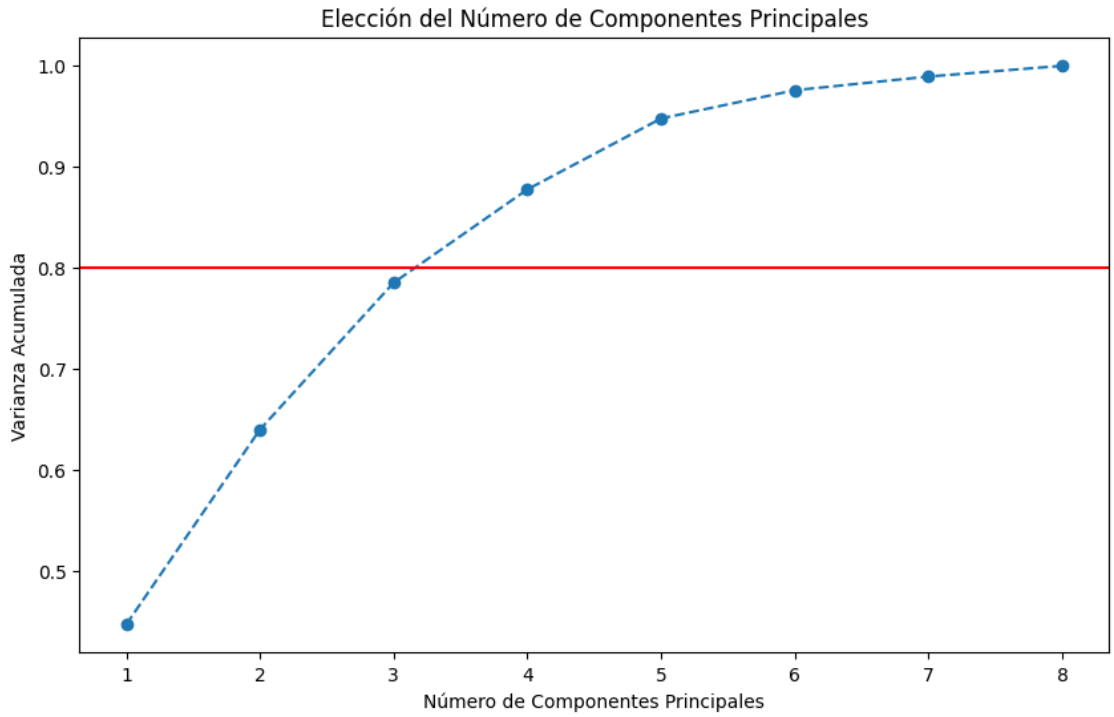


Aplicar Componentes Principales (PCA)

[47]:
```
pca = PCA(n_components=X.shape[1])  # Inicialmente usar todos los componentes
pca.fit(X_scaled)
explained_variance = pca.explained_variance_ratio_
cumulative_variance = np.cumsum(explained_variance)
print("Varianza explicada por cada componente:", explained_variance)
print("Varianza acumulada:", cumulative_variance)

# Graficar la varianza acumulada para decidir el número de componentes
plt.figure(figsize=(10, 6))
plt.plot(range(1, len(cumulative_variance)+1), cumulative_variance, marker='o',␣
 ↪linestyle='--')
plt.xlabel("Número de Componentes Principales")
plt.ylabel("Varianza Acumulada")
plt.axhline(y=0.80, color='r', linestyle='-')
plt.title("Elección del Número de Componentes Principales")
plt.show()

# Selección de los componentes que explican al menos el 80% de la varianza
n_components = np.argmax(cumulative_variance >= 0.80) + 1
```

```
pca = PCA(n_components=n_components)
X_pca = pca.fit_transform(X_scaled)
print(f"Usando {n_components} componentes principales.")

# Interpretación de los componentes
components_df = pd.DataFrame(pca.components_, columns=X.columns)
print("Componentes principales:\n", components_df)
```

```
Varianza explicada por cada componente: [0.4468279  0.19299326 0.14542167
0.09234891 0.07027514 0.02793591
 0.01356703 0.01063019]
Varianza acumulada: [0.4468279  0.63982116 0.78524283 0.87759173 0.94786687
0.97580278
 0.98936981 1.        ]
```




```
Usando 4 componentes principales.
Componentes principales:
    child_mort   exports    health   imports    income  inflation  life_expec  \
0    0.472880 -0.308396 -0.144568 -0.194640 -0.386787   0.220475   -0.464191
1    0.214124  0.608374 -0.241608  0.661131  0.031207   0.005771   -0.237343
2    0.099988 -0.146037  0.647403  0.285257 -0.247776  -0.615777   -0.158082
3    0.115187  0.101508  0.680156  0.056361  0.315029   0.621292    0.003857

   total_fer
```

```
0    0.456952
1    0.176702
2    0.051085
3    0.159304
```

El análisis de Componentes Principales (PCA) es una técnica de reducción de dimensionalidad que transforma un conjunto de variables correlacionadas en un nuevo conjunto de variables no correlacionadas, llamadas componentes principales.

Varianza Explicada por Cada Componente

- Primer componente (PC1): 44.68% de la varianza

Las variables que más contribuyen a este componente son child_mort, income, life_expec, y total_fer.

- Segundo componente (PC2): 19.30% de la varianza

Las variables con mayores pesos son exports y imports.

- Tercer componente (PC3): 14.54% de la varianza

Las variables health, inflation y imports tienen los pesos más significativos.

- Cuarto componente (PC4): 9.23% de la varianza

Las variables health e inflation son nuevamente relevantes en este componente, pero en menor medida.

Con estos 4 componentes obtenemos un 78.52% de la varianza acumulada de los datos originales.

Valores Propios y Direcciones de los Componentes

Los valores propios asociados a cada componente indican cuánta varianza de los datos originales es explicada por cada componente principal.

Las direcciones de los componentes se reflejan en los vectores propios (los pesos de cada variable en cada componente), estos vectores propios indican qué variables contribuyen más a cada componente.

Nueva regresión con los componentes principales seleccionados

[48]:
```
X_pca_with_constant = sm.add_constant(X_pca)
model_pca = sm.OLS(y, X_pca_with_constant).fit()
print(model_pca.summary())

# Comparación entre ambos modelos
print("\nModelo con todas las variables predictoras:\n", model.summary())
print("\nModelo con componentes principales:\n", model_pca.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                   gdpp   R-squared:                       0.610
Model:                            OLS   Adj. R-squared:                  0.600
Method:                 Least Squares   F-statistic:                     63.36
Date:                Mon, 28 Oct 2024   Prob (F-statistic):           3.76e-32
```

```
Time:                        05:46:57   Log-Likelihood:                -1797.1
No. Observations:                 167   AIC:                             3604.
Df Residuals:                     162   BIC:                             3620.
Df Model:                           4
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const       1.296e+04    896.556     14.460      0.000    1.12e+04    1.47e+04
x1         -6705.6797    474.201    -14.141      0.000   -7642.092   -5769.268
x2          -616.1469    721.542     -0.854      0.394   -2040.987     808.693
x3          -880.3942    831.224     -1.059      0.291   -2521.825     761.036
x4          7493.7030   1043.078      7.184      0.000    5433.921    9553.485
==============================================================================
Omnibus:                       48.034   Durbin-Watson:                   2.179
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              114.651
Skew:                           1.229   Prob(JB):                     1.27e-25
Kurtosis:                       6.230   Cond. No.                         2.20
==============================================================================
```

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Modelo con todas las variables predictoras:

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                   gdpp   R-squared:                       0.866
Model:                            OLS   Adj. R-squared:                  0.859
Method:                 Least Squares   F-statistic:                     127.7
Date:                Mon, 28 Oct 2024   Prob (F-statistic):           6.13e-65
Time:                        05:46:57   Log-Likelihood:                -1707.9
No. Observations:                 167   AIC:                             3434.
Df Residuals:                     158   BIC:                             3462.
Df Model:                           8
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const       1.296e+04    532.047     24.367      0.000    1.19e+04     1.4e+04
x1          2676.5161   1428.286      1.874      0.063    -144.481    5497.513
x2           778.5286   1180.939      0.659      0.511   -1553.934    3110.991
x3          4241.5150    622.168      6.817      0.000    3012.677    5470.353
x4          -678.7091   1026.215     -0.661      0.509   -2705.579    1348.160
x5           1.51e+04    839.280     17.990      0.000    1.34e+04    1.68e+04
x6         -1059.1469    597.278     -1.773      0.078   -2238.825     120.532
x7          3448.6094   1267.622      2.721      0.007     944.940    5952.279
x8           928.3608   1026.205      0.905      0.367   -1098.488    2955.210
```

```
==============================================================================
Omnibus:                       53.684   Durbin-Watson:                   1.914
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              287.333
Skew:                           1.040   Prob(JB):                     4.04e-63
Kurtosis:                       9.080   Cond. No.                         6.48
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
```

Modelo con componentes principales:

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                   gdpp   R-squared:                       0.610
Model:                            OLS   Adj. R-squared:                  0.600
Method:                 Least Squares   F-statistic:                     63.36
Date:                Mon, 28 Oct 2024   Prob (F-statistic):           3.76e-32
Time:                        05:46:57   Log-Likelihood:                -1797.1
No. Observations:                 167   AIC:                             3604.
Df Residuals:                     162   BIC:                             3620.
Df Model:                           4
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const       1.296e+04    896.556     14.460      0.000    1.12e+04    1.47e+04
x1         -6705.6797    474.201    -14.141      0.000   -7642.092   -5769.268
x2          -616.1469    721.542     -0.854      0.394   -2040.987     808.693
x3          -880.3942    831.224     -1.059      0.291   -2521.825     761.036
x4          7493.7030   1043.078      7.184      0.000    5433.921    9553.485
==============================================================================
Omnibus:                       48.034   Durbin-Watson:                   2.179
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              114.651
Skew:                           1.229   Prob(JB):                     1.27e-25
Kurtosis:                       6.230   Cond. No.                         2.20
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
```

El modelo con todas las variables ofrece un mejor ajuste y una interpretación más detallada de cómo cada factor afecta a gdpp, mientras que el modelo basado en PCA es una opción simplificada y estable, aunque con menor capacidad explicativa y menor claridad interpretativa ya que los coeficientes se relacionan con combinaciones de variables.

Análisis de conglomerados (clustering) con KMeans

[49]:
```
kmeans = KMeans(n_clusters=3, random_state=0).fit(X_pca)
data['Cluster'] = kmeans.labels_

# Visualización de los clusters
plt.figure(figsize=(10, 8))
sns.scatterplot(x=X_pca[:, 0], y=X_pca[:, 1], hue=data['Cluster'],␣
 ↪palette='viridis')
plt.title("Visualización de Clusters utilizando Componentes Principales")
plt.xlabel("Componente Principal 1")
plt.ylabel("Componente Principal 2")
plt.legend()
plt.show()
```