

Aldo Daniel Villaseñor Fierro

A01637907

```
In [ ]: import numpy as np
import pandas as pd
import statsmodels.api as sm
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA
import plotly.express as px
```

```
In [ ]: data=pd.read_csv('países_mundo.csv')
```

Matriz de correlaciones y de covarianzas

```
In [ ]: # Matriz de varianza-covarianza
cov_matrix = data.cov()

# Matriz de correlaciones
correlation_matrix = data.corr()
```

```
In [ ]: # Paso 2: Cálculo de valores y vectores propios
# Valores y vectores propios de la matriz de varianza-covarianza
eigenvalues_cov, eigenvectors_cov = np.linalg.eig(cov_matrix)

# Valores y vectores propios de la matriz de correlaciones
eigenvalues_cor, eigenvectors_cor = np.linalg.eig(correlation_matrix)
```

```
In [ ]: # Paso 3: Proporción de varianza explicada
variance_explained_cov = eigenvalues_cov / np.trace(cov_matrix) # Total varianza = su
variance_explained_cor = eigenvalues_cor / np.trace(correlation_matrix)
```

```
In [ ]: # Paso 4: Acumulación de resultados
cumulative_variance_cov = np.cumsum(variance_explained_cov)
cumulative_variance_cor = np.cumsum(variance_explained_cor)
```

```
In [ ]: # Paso 5: Análisis de componentes más importantes
# Proporciones de varianza explicada y las matrices de vectores propios.
print("Proporción de varianza explicada por componentes (varianza-covarianza):")
print(variance_explained_cov)

print("Proporción de varianza explicada por componentes (matriz de correlaciones):")
print(variance_explained_cor)

# Comparación de resultados entre matriz de varianza-covarianza y matriz de correlación
print("Varianza acumulada (varianza-covarianza):")
print(cumulative_variance_cov)

print("Varianza acumulada (matriz de correlaciones):")
print(cumulative_variance_cor)
```

```

Proporción de varianza explicada por componentes (varianza-covarianza):
[9.03454311e-01 9.64729842e-02 6.79580362e-05 4.55456679e-06
 1.78242937e-07 7.53091641e-09 5.31773802e-09 6.65776295e-10
 8.50288738e-11 6.98903508e-12 2.10784328e-11]
Proporción de varianza explicada por componentes (matriz de correlaciones):
[0.36635264 0.17545381 0.12458283 0.00630533 0.01330256 0.01527895
 0.02970918 0.05193683 0.07859236 0.0721946 0.06629091]
Varianza acumulada (varianza-covarianza):
[0.90345431 0.9999273 0.99999525 0.99999981 0.99999999 0.99999999
 1. 1. 1. 1. 1. ]
Varianza acumulada (matriz de correlaciones):
[0.36635264 0.54180645 0.66638928 0.67269462 0.68599718 0.70127613
 0.73098531 0.78292214 0.8615145 0.93370909 1. ]

```

La principal diferencia entre los resultados de la matriz de varianza-covarianza y la matriz de correlaciones radica en la escala de las variables. La matriz de varianza-covarianza considera la variabilidad original de las variables, mientras que la matriz de correlaciones elimina esta variabilidad y solo se centra en las relaciones lineales entre las variables.

En este caso, parece que la matriz de varianza-covarianza muestra que las primeras dos componentes explican una cantidad significativamente mayor de la varianza en comparación con la matriz de correlaciones. Esto sugiere que las unidades y la varianza original de las variables son relevantes en el análisis. Por lo cual las primeras dos componentes en la matriz de varianza-covarianza son las más importantes. Las variables con las mayores cargas en estas componentes son las que más contribuyen a ellas.

```

In [ ]: # Realizamos el análisis de componentes principales (PCA)
pca = PCA()
cpS = pca.fit(data)

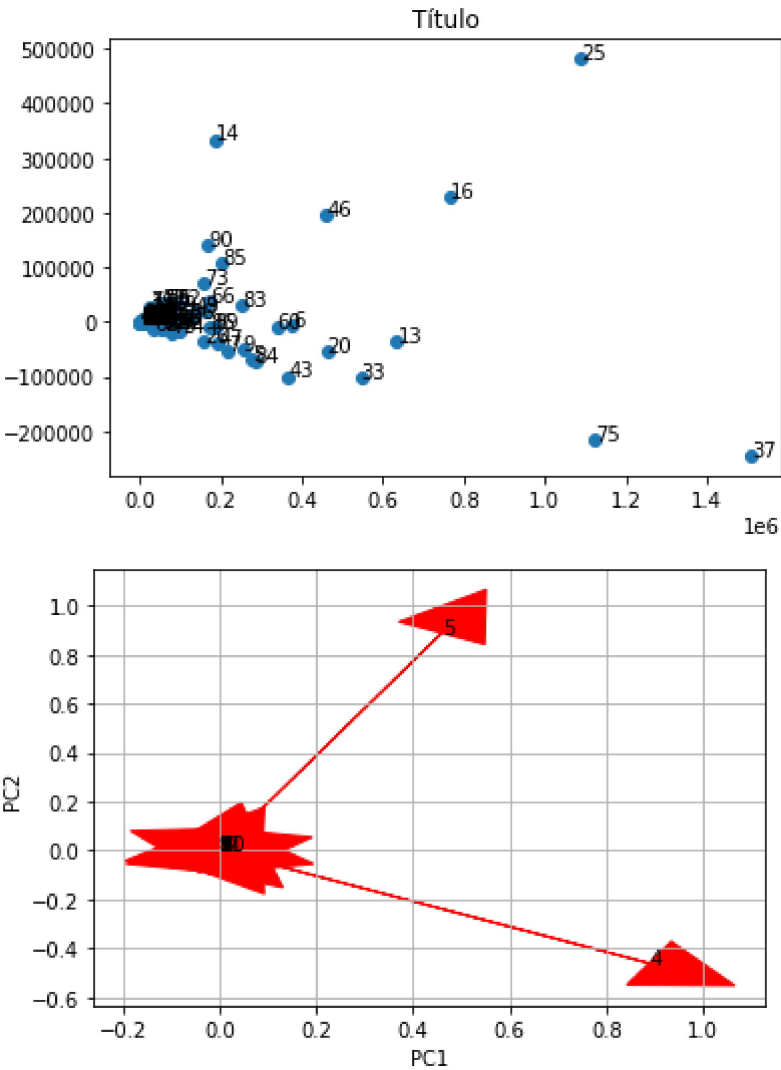
# Proyectamos los datos en las nuevas dimensiones
cpaS = np.dot(data, cpS.components_.T)

# Graficamos las dos primeras componentes principales
plt.scatter(cpaS[:, 0], cpaS[:, 1])
plt.title("Título")
for i in range(len(cpaS)):
    plt.text(cpaS[i, 0], cpaS[i, 1], str(i + 1))

# Gráfico biplot
plt.figure()
for i in range(len(cpS.components_)):
    plt.arrow(0, 0, cpS.components_[0, i], cpS.components_[1, i], head_width=0.2, head_height=0.2)
    plt.text(cpS.components_[0, i], cpS.components_[1, i], str(i + 1))

plt.xlabel("PC1")
plt.ylabel("PC2")
plt.grid()
plt.show()

```



```
In [ ]:
```