Jorge Javier Sosa Briseño
A01749489
September 29, 2023

2.- Componentes Principales

```
# Imprimir proporción de varianza explicada en R
print("Proporción de varianza explicada en R:")
print(f"Suma acumulada: {cum_sum_cor}")
```

✓ 0.0s Python

```
Proporción de varianza explicada en S:
Suma acumulada: [0.90345431 0.9999273  0.99999525 0.99999981 0.99999999 0.99999999
 1.         1.         1.         1.         1.         ]

Proporción de varianza explicada en R:
Suma acumulada: [0.36635264 0.54180645 0.66638928 0.67269462 0.68599718 0.70127613
 0.73098531 0.78292214 0.8615145  0.93370909 1.         ]
```

```
# Identificar las componentes más importantes (por ejemplo, las que explican el 95% de la varianza)
num_componentes = np.sum(prop_var_acumulada <= 0.95)

# Imprimir el número de componentes principales que explican el 95% de la varianza
print("Número de componentes principales que explican el 95% de la varianza:", num_componentes)

if num_componentes > 0:
    # Extraer los vectores propios correspondientes a las componentes más importantes
    vectores_importantes = eigenvectors[:, :num_componentes]

    # Calcular las contribuciones de las variables originales a las componentes principales
    contribuciones_variables = np.dot(cor_matrix, vectores_importantes)

    # Imprimir las contribuciones de las variables a las componentes principales
    for i in range(num_componentes):
        print(f"Contribuciones de las variables a la componente principal {i + 1}:")
        print(contribuciones_variables[:, i])
else:
    print("No hay componentes principales significativas que expliquen el 95% de la varianza.")
```

Export failed. Please check the 'Jupyter' output panel for further deta...

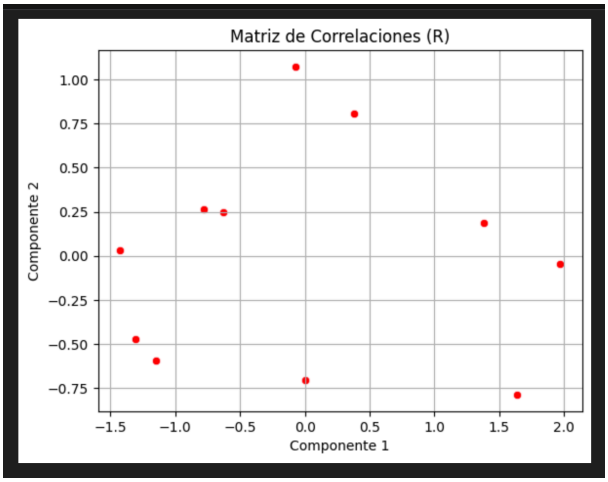
4 0 Ln 4, Col 39 LF Cell 7 of 7

```
        print(contribuciones_variables[:, i])
    else:
        print("No hay componentes principales significativas que expliquen el 95% de la varianza.")
```

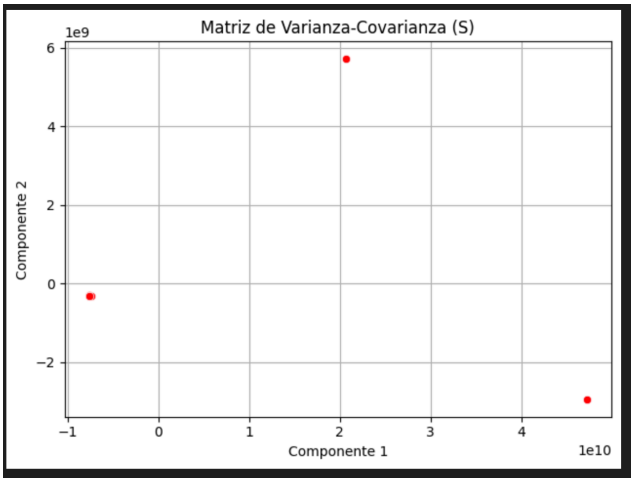
[24] ✓ 0.0s

```
Número de componentes principales que explican el 95% de la varianza: 1
Contribuciones de las variables a la componente principal 1:
[-0.42366985 -0.39290479  0.21513688  1.23137272  1.12187271  0.54747903
  0.15954331  0.06072577 -0.2524727   0.35741886  0.28364183]
```

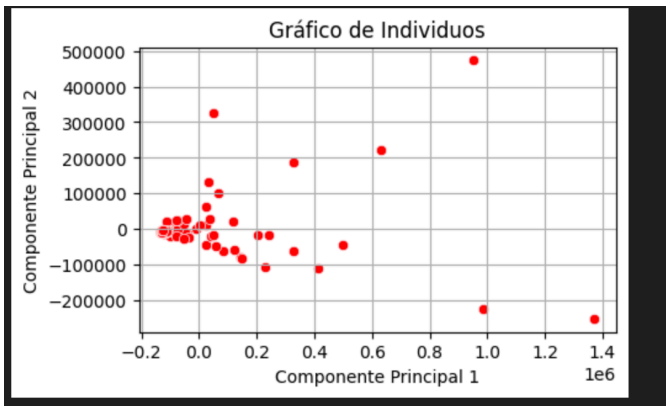
```
# Realizar PCA para matriz de varianza-covarianza (S)
pca_S = PCA(n_components=2)
componentes_S = pca_S.fit_transform(S)
componentes_S = pd.DataFrame(componentes_S, columns=["Componente 1", "Componente 2"])

# Realizar PCA para matriz de correlaciones (R)
pca_R = PCA(n_components=2)
componentes_R = pca_R.fit_transform(R)
componentes_R = pd.DataFrame(componentes_R, columns=["Componente 1", "Componente 2"])


# Gráfico de PCA de Matriz de Correlaciones (R)

sns.scatterplot(x="Componente 1", y="Componente 2", data=componentes_R,color="red")
plt.grid()
plt.title("Matriz de Correlaciones (R)")
plt.show()
# Gráfico de PCA de Matriz de Varianza-Covarianza (S)

sns.scatterplot(x="Componente 1", y="Componente 2", data=componentes_S,color="red")
```



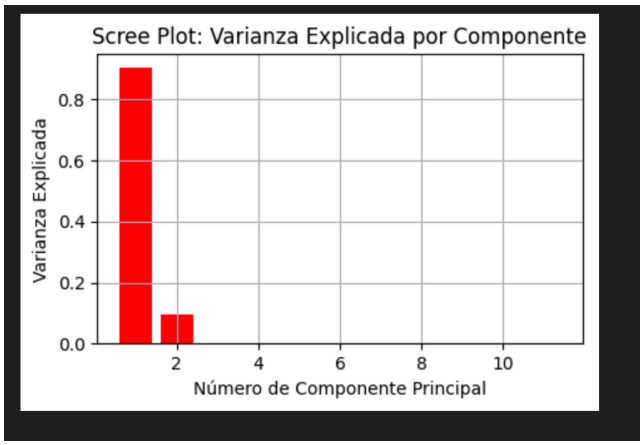
Aquí podemos ver cómo se relacionan la component 1 con la 2 y podemos ver las correlaciones existentes y hay una calara independencia entre los componentes.



Aquí podemos ver cómo se relacionan la component 1 con la 2 con las covarianzas existentes los cual nos indica que existe una falla en la prueba de independencia.



Se puede observar que en la gráfica de individuos no se cumple la homoceasticidad entre la componente principal 1 y la componente 2.



Se puede observar como los valores principales están mas en la primer componente.