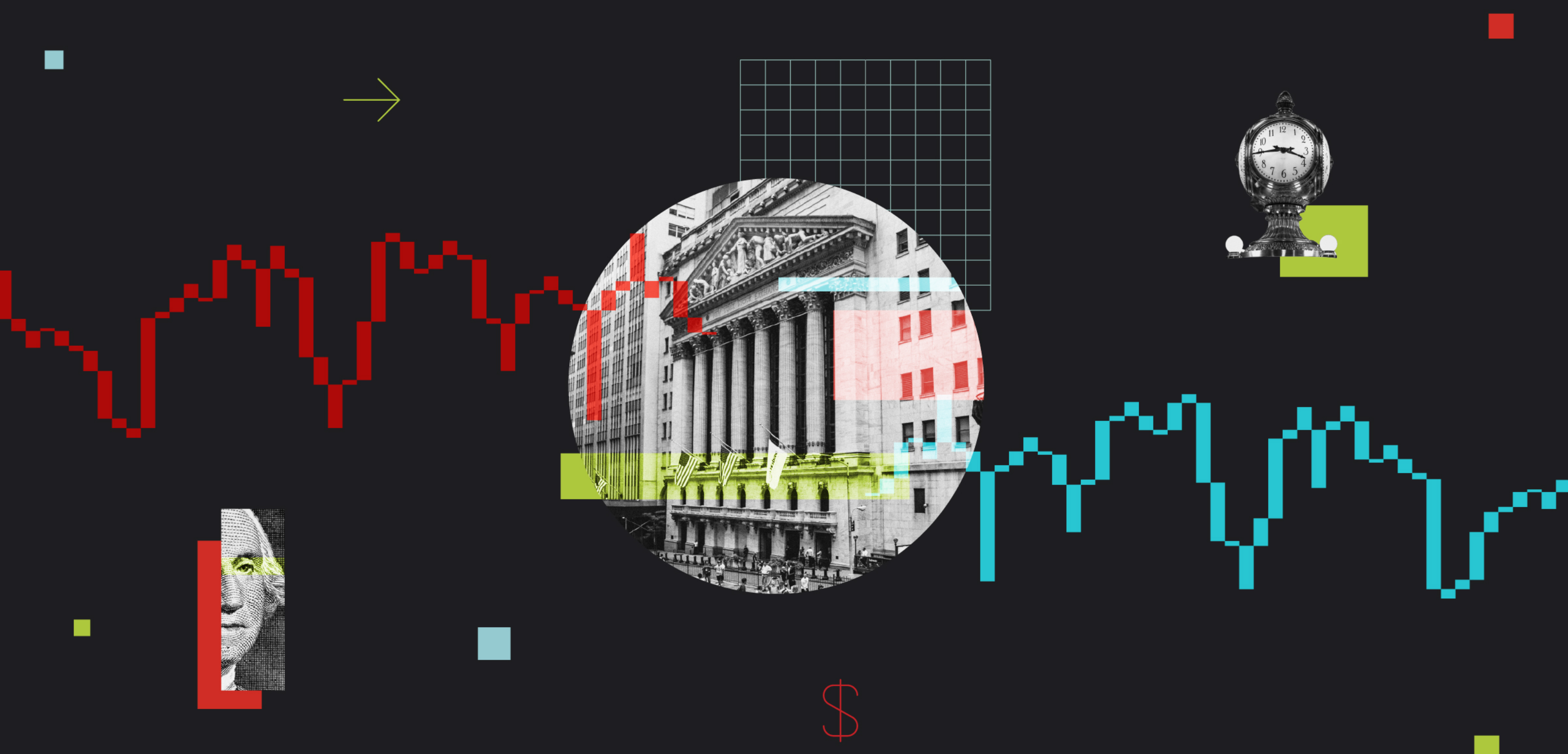
Time Series Forecasting & Trend Analysis

Predicción de Series Temporales y Análisis de Tendencias



Por Antonio Richaud

Primera Versión

Time Series Forecasting & Trend Analysis

Predicción y Análisis de Tendencias

Autor: Antonio Richaud

Introducción

Este notebook tiene como objetivo realizar un análisis exhaustivo de los datos financieros, centrándose en la relación entre los pasivos totales (TL), los pasivos no corrientes (NCL) y los pasivos corrientes (CL). A lo largo de este análisis, se aplicarán diversas técnicas estadísticas y econométricas para explorar, modelar y predecir las tendencias en una empresa.

Objetivos Específicos:

1. **Análisis Descriptivo**:
   - Calcular y presentar estadísticas descriptivas para obtener una comprensión inicial de la distribución y características de los datos.
   - Evaluar la normalidad, asimetría y curtosis de los datos.
2. **Pruebas de Hipótesis**:
   - Realizar pruebas de normalidad de Jarque-Bera para evaluar si los datos siguen una distribución normal.
   - Aplicar la prueba ADF (Dickey-Fuller aumentada) para verificar la estacionariedad de las series temporales.
   - Ejecutar la prueba BG-LM (Breusch-Godfrey) para detectar autocorrelación en los residuos del modelo de regresión.
   - Calcular el estadístico de Durbin-Watson para evaluar la autocorrelación de los errores.
3. **Modelado de Regresión**:
   - Construir un modelo de regresión lineal para investigar la relación entre TL, NCL y CL.
   - Evaluar la bondad de ajuste del modelo mediante el R-cuadrado y el R-cuadrado ajustado.
   - Analizar la significancia de las variables independientes mediante los valores p y los estadísticos t.
4. **Análisis de Errores**:
   - Realizar pruebas adicionales como el correlograma y la prueba BG-LM para examinar los residuos del modelo.
   - Aplicar la prueba ADF en los residuos para verificar la presencia de raíces unitarias.
   - Evaluar la normalidad de los errores mediante la prueba de Jarque-Bera y visualizar los residuos.
5. **Predicción y Análisis de Tendencias**:

- Utilizar modelos de series temporales para predecir las tendencias futuras en el mercado de acciones.
- Analizar las tendencias identificadas y discutir su implicación en el contexto financiero.

Importancia del Análisis:

La predicción y análisis de las tendencias en los datos de una empresa es fundamental para la toma de decisiones financieras informadas. Este estudio proporciona una visión detallada de los factores que afectan los pasivos de las empresas, permitiendo a los analistas y tomadores de decisiones comprender mejor las dinámicas de los datos y anticipar posibles cambios.

1 - Descripción del dataframe:

Y = f (x1, x2)

Pasivos Totales = f (Pasivos No Corrientes, Pasivos Corrientes)

Datos y Fuente

Variable	Código de Variable	Descripción y Medición
Pasivos Totales (Total Liabilities)	TL	MONTO
Pasivos No Corrientes (Non-Current Liabilities)	NCL	MONTO
Pasivos Corrientes (Current Liabilities)	CL	MONTO

Interpretación:

Los datos muestran la estructura financiera de una empresa, incluyendo pasivos totales (Total Liabilities - TL), pasivos no corrientes (Non-Current Liabilities - NCL) y pasivos corrientes (Current Liabilities - CL) junto con sus respectivos montos. A continuación se hace una descripción de los datos:

- **Pasivos Totales (TL)**: Todas las obligaciones financieras de una empresa u organización, incluyendo tanto los pasivos corrientes como los no corrientes, se denominan pasivos totales.

- **Pasivos No Corrientes (NCL)**: Estas son deudas que no se anticipa que se liquiden durante el ciclo operativo actual o el término fiscal de la empresa u organización. Las deudas a largo plazo, como los bonos o los préstamos a largo plazo, generalmente se incluyen en los pasivos no corrientes.

- **Pasivos Corrientes (CL)**: Estas son las deudas que la empresa u organización espera pagar durante su año fiscal o ciclo operativo actual. Las cuentas por pagar, los salarios y los impuestos son ejemplos de deudas a corto plazo que típicamente se incluyen en los pasivos corrientes.

```python
import pandas as pd

# Ruta al archivo CSV
file_path = '/content/sample_data/financial_data.csv'

# Cargar el dataframe desde el archivo CSV
df = pd.read_csv(file_path)

# Mostramos todo el contenido
print(df)
```

```
      Year          TL         CL        NCL
0   Mar-23 28040.6000 6987.9400 3138.4900
1   Mar-22 25864.7500 4180.9700 4566.4400
2   Mar-21 24093.3200 6756.5800 1412.1300
3   Mar-20 19563.6100 6372.2500  138.8600
4   Mar-19 18140.7200 6736.4600   53.6400
5   Mar-18 15793.8400 5754.1800   57.0800
6   Mar-17 13022.8400 4446.9700  139.6900
7   Mar-16 12558.7000 5343.6900  349.1600
8   Mar-15 10225.8800 3946.1900  920.1500
9   Mar-14  8575.5300 3317.2800 1245.8600
10  Mar-13  6824.5700 2767.9600 1117.5000
11  Mar-12  5991.2100 2656.3800  841.6600
12  Mar-11  5841.1700 2671.7000  599.8600
13  Mar-10  4594.0800  639.8700 2039.8500
14  Mar-09  4097.0400  583.1300 2193.0800
15  Mar-08  3494.4400  445.7300 1828.1100
16  Mar-07  3319.3600  346.4800 2043.5600
17  Mar-06  2480.7400  298.2100 1277.0800
18  Mar-05  1914.8300  206.1400  885.1000
19  Mar-04  1637.2100  175.3400  670.4900
20  Mar-03  1322.9700  226.0100  561.4200
21  Mar-02   886.3100  154.5100  369.3800
22  Mar-01   639.2300  136.7100  225.9000
23  Mar-00   448.5400   87.9200  140.9000
24  Mar-99   345.7000  110.0100  110.9600
25  Mar-98   196.9200   56.0000   70.8600
26  Mar-97   143.6100   42.6200   52.7000
27  Mar-96    89.8400   20.7700   32.0800
28  Mar-95    48.0700   11.4300    5.6000
29  Mar-94    11.8200    3.3200    2.7800
```

2 - Estadísticas Descriptivas

```python
import numpy as np
import scipy.stats as stats

# Excluimos la columna 'Year' del análisis estadístico
df_numeric = df.drop(columns=['Year'])
```

```python
# Calculamos estadísticas descriptivas
descriptive_stats = df_numeric.describe()

# Calculamos skewness y kurtosis con scipy.stats
skewness = df_numeric.apply(lambda x: stats.skew(x, bias=False))
kurtosis = df_numeric.apply(lambda x: stats.kurtosis(x, bias=False,
fisher=False))

# Creamos DataFrame para estadísticas adicionales
additional_stats = pd.DataFrame({
    'skewness': skewness,
    'kurtosis': kurtosis,
    'jarque_bera': df_numeric.apply(lambda x: stats.jarque_bera(x)
[0]),
    'jb_pvalue': df_numeric.apply(lambda x: stats.jarque_bera(x)[1]),
    'sum': df_numeric.sum(),
    'sum_sq_dev': ((df_numeric - df_numeric.mean())**2).sum()
})

# Conteo
count = df_numeric.count().to_frame().T

# Unimos las estadísticas descriptivas con las adicionales
descriptive_stats = pd.concat([descriptive_stats, additional_stats.T,
count])

# Reorganizar el orden de las filas
ordered_stats = descriptive_stats.loc[['mean', '50%', 'max', 'min',
'std', 'skewness', 'kurtosis', 'jarque_bera', 'jb_pvalue', 'sum',
'sum_sq_dev', 'count']]

# Redondear los valores a 4 decimales para una mejor presentación
ordered_stats = ordered_stats.round(4)

# Quitamos la notación cientifica
pd.set_option('display.float_format', lambda x: '%.4f' % x)

# Mostrar estadísticas descriptivas completas con mejor formato
print("Estadísticas descriptivas completas:")
print(ordered_stats.to_string())
```

```
Estadísticas descriptivas completas:
                      TL          CL         NCL
mean           7340.2483   2182.7583    903.0123
50%            3795.7400    514.4300    580.6400
max           28040.6000   6987.9400   4566.4400
min              11.8200      3.3200      2.7800
std            8468.2086   2534.1723   1066.9435
skewness          1.2151      0.7818      1.8068
```

```
kurtosis               3.3718          2.0251         6.7919
jarque_bera            6.6699          4.0374        25.9234
jb_pvalue              0.0356          0.1328         0.0000
sum               220207.4500      65482.7500     27090.3700
sum_sq_dev  2079606146.4444 186238852.7546 33012683.2485
count                 30.0000         30.0000        30.0000
```

Prueba de Hipótesis:

- **H0 (Hipótesis nula)**: Los datos están distribuidos normalmente (valor p > 0.05).
- **H1 (Hipótesis alternativa)**: Los datos no están distribuidos normalmente (valor p < 0.05).

Interpretación de la Asimetría:

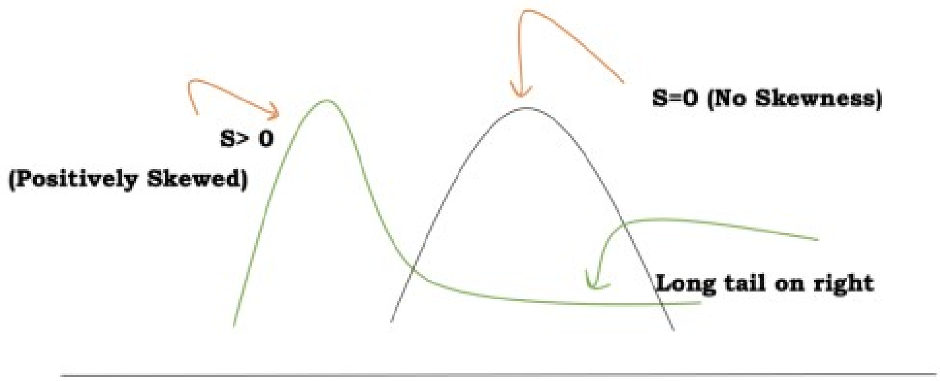
- **Asimetría positiva**: Indica una "Media más alta" que la "Mediana".
- Los valores del conjunto de datos se agrupan hacia la "Izquierda", lo que muestra una "Asimetría positiva" con la cola de mi conjunto de datos hacia la derecha.

Explicación de la Imagen:

La imagen muestra dos curvas de distribución:

1. **Distribución sin Asimetría (S = 0)**:
   - Representada por la curva negra.
   - Esta distribución es simétrica, lo que significa que su media y su mediana son iguales.
   - No tiene cola larga ni hacia la izquierda ni hacia la derecha.
2. **Distribución con Asimetría Positiva (S > 0)**:
   - Representada por la curva verde.
   - Esta distribución es asimétrica con una cola larga hacia la derecha.
   - La media es mayor que la mediana.
   - Los valores se agrupan hacia la izquierda, lo que provoca que la cola se extienda hacia la derecha.

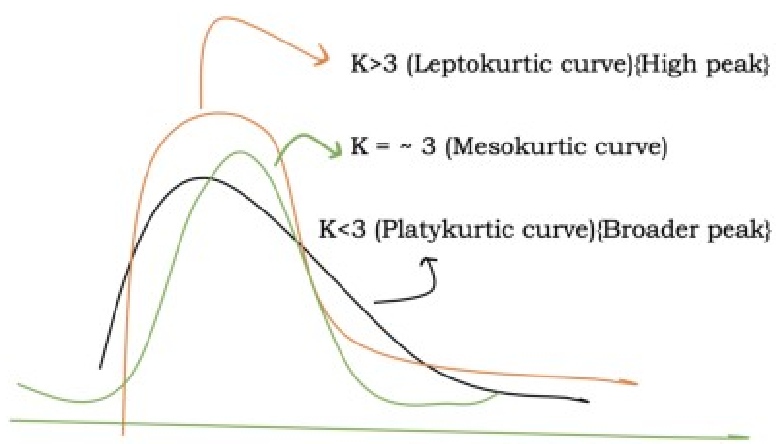
```
import matplotlib.pyplot as plt
from PIL import Image

image_path = '/content/Captura de pantalla 2024-07-10 a la(s)
2.55.05 p.m. (1).png'
image = Image.open(image_path)
plt.imshow(image)
plt.axis('off')
plt.show()
```



Interpretación de la Kurtosis:

- **K > 3 (Curva Leptocúrtica)**: Alto pico.
- **K ≈ 3 (Curva Mesocúrtica)**: Distribución normal.
- **K < 3 (Curva Platicúrtica)**: Pico más ancho.

```
image_path = '/content/Captura de pantalla 2024-07-10 a la(s)
12.19.39 p.m. (1).png'
image = Image.open(image_path)
plt.imshow(image)
plt.axis('off')
plt.show()
```



Interpretación para Pasivos Corrientes (CL):

- **Kurtosis**: 1.9866 (Menor que 3, platicúrtica).
  - Muestra un pico más ancho y una cola más delgada.
  - Los valores están más dispersos respecto a la media.
  - Menor probabilidad de eventos extremos.

Interpretación para Pasivos No Corrientes (NCL):

- **Kurtosis**: 5.9952 (Mayor que 3, leptocúrtica).
  - Muestra un pico alto y una cola pesada.
  - Los valores están agrupados alrededor de la media.
  - Mayor probabilidad de eventos extremos.

Interpretación General:

- **Pasivos Corrientes (CL)**:
  - **Skewness**: 0.74, **Kurtosis**: 1.99
  - Los datos están asimétricamente positivos.
- **Pasivos No Corrientes (NCL)**:
  - **Skewness**: 1.71, **Kurtosis**: 5.99
  - Los datos están asimétricamente positivos.
- **Pasivos Totales (TL)**:
  - **Skewness**: 1.15, **Kurtosis**: 3.11
  - Los datos están asimétricamente positivos.

Prueba de Jarque-Bera:

Interpretación de los Valores de Probabilidad de Jarque-Bera:

- **Para Pasivos Corrientes (CL)**:
  - **Valor p**: 0.132827 (Mayor que 0.05, no significativa).
  - Aceptamos (H_0), los datos son normales.
- **Para Pasivos No Corrientes (NCL)**:
  - **Valor p**: 0.000002 (Menor que 0.05, significativa).
  - Rechazamos (H_0), los datos no son normales.
- **Para Pasivos Totales (TL)**:
  - **Valor p**: 0.035616 (Menor que 0.05, significativa).
  - Rechazamos (H_0), los datos no son normales.

3 - Calcular la Matriz de Correlación

```python
# Se hace el calculo de la matriz de correlación
correlation_matrix = df_numeric.corr()

# Mostrar la matriz de correlación
print("Matriz de Correlación:")
print(correlation_matrix)
```

```
Matriz de Correlación:
            TL      CL     NCL   D(TL)  D(NCL)   D(CL)
TL      1.0000  0.9292  0.4854 -0.7716 -0.1980 -0.1084
CL      0.9292  1.0000  0.1942 -0.7649 -0.2682  0.0362
NCL     0.4854  0.1942  1.0000 -0.1640  0.2920 -0.4321
D(TL)  -0.7716 -0.7649 -0.1640  1.0000  0.1294  0.3513
```

```
D(NCL) -0.1980 -0.2682  0.2920  0.1294  1.0000 -0.8036
D(CL)  -0.1084  0.0362 -0.4321  0.3513 -0.8036  1.0000
```

Interpretación:

- El valor de correlación ideal existe entre -1 y 1.
- De los datos anteriores, la correlación entre CL y TL es 0.9292, lo cual es una correlación positiva. Por lo tanto, un aumento en los pasivos corrientes resultará en un aumento en los pasivos totales.
- La correlación entre NCL y TL es 0.4853, lo cual es una correlación positiva. Por lo tanto, un aumento en los pasivos no corrientes resultará en un aumento en los pasivos totales.

4 - Test de Raíz Unitaria para los Pasivos Totales (TL)

Explicación

El test de raíz unitaria se utiliza para determinar si una serie temporal es estacionaria o no. En este caso, se realiza el test de Dickey-Fuller Aumentado (ADF) para los pasivos totales (TL).

Componentes Clave del Test de Raíz Unitaria:

1. **Hipótesis Nula (H0)**: La serie temporal tiene una raíz unitaria (es no estacionaria).
2. **Hipótesis Alternativa (H1)**: La serie temporal no tiene una raíz unitaria (es estacionaria).

Resultados Clave del Test de Dickey-Fuller Aumentado:

1. **Estadístico de prueba**: Valor calculado de la prueba ADF.
2. **Valores críticos**: Valores críticos para los niveles de significancia del 1%, 5% y 10%.
3. **p-Valor (Prob.)**: Indica la probabilidad de obtener un valor extremo del estadístico de prueba bajo la hipótesis nula.

Interpretación de los Resultados:

- Si el **estadístico de prueba** es menor que los valores críticos (en términos absolutos), rechazamos la hipótesis nula.
- Si el **p-Valor** es menor que el nivel de significancia (0.05, por ejemplo), rechazamos la hipótesis nula, lo que indica que la serie temporal es estacionaria.

Ecuación del Test de Dickey-Fuller Aumentado:

La ecuación del test ADF incluye las diferencias retardadas de la serie temporal para ajustar por autocorrelación.

Variables de la Ecuación:

- **TL(-1)**: Pasivos totales con un retardo de 1 periodo.
- **D(TL(-1))**: Primera diferencia de los pasivos totales con un retardo de 1 periodo.
- **C**: Constante.
- **Coeficiente**: Valor del coeficiente estimado.
- **Std. Error**: Error estándar del coeficiente estimado.
- **t-Statistic**: Valor del estadístico t para el coeficiente.

- **Prob.**: Valor p para el estadístico t.

Métricas de Ajuste del Modelo:

- **R-squared**: Coeficiente de determinación, indica qué proporción de la variabilidad en la variable dependiente es explicada por el modelo.
- **Akaike info criterion (AIC)**: Criterio de información de Akaike, utilizado para comparar modelos.
- **Durbin-Watson stat**: Estadístico de Durbin-Watson, utilizado para detectar la autocorrelación en los residuos del modelo.

```python
import statsmodels.tsa.stattools as adf

# Realizar el test de Dickey-Fuller Aumentado para los pasivos totales
(TL) con especificación de retardos
result = adf.adfuller(df_numeric['TL'], maxlag=6)

# Mostrar los resultados del test ADF
print('Estadístico de prueba:', result[0])
print('p-Valor:', result[1])
print('Valores críticos:', result[4])
print('Número de retardos utilizados:', result[2])
print('Número de observaciones:', result[3])
print('Información completa del resultado:', result)
```

```
Estadístico de prueba: -3.0845303743283368
p-Valor: 0.02771939390168125
Valores críticos: {'1%': -3.7377092158564813, '5%': -
2.9922162731481485, '10%': -2.635746736111111}
Número de retardos utilizados: 5
Número de observaciones: 24
Información completa del resultado: (-3.0845303743283368,
0.02771939390168125, 5, 24, {'1%': -3.7377092158564813, '5%': -
2.9922162731481485, '10%': -2.635746736111111}, 339.3076178543226)
```

5 - Test de Raíz Unitaria para la Primera Diferencia de los Pasivos Totales (TL)

Explicación

El test de raíz unitaria se utiliza para determinar si una serie temporal es estacionaria o no. En este caso, se realiza el test de Dickey-Fuller Aumentado (ADF) para la primera diferencia de los pasivos totales (D(TL)). Esto se hace para verificar si la serie temporal se vuelve estacionaria después de tomar la primera diferencia.

Componentes Clave del Test de Raíz Unitaria para la Primera Diferencia:

1. **Hipótesis Nula (H0)**: La primera diferencia de la serie temporal tiene una raíz unitaria (es no estacionaria).

2. **Hipótesis Alternativa (H1)**: La primera diferencia de la serie temporal no tiene una raíz unitaria (es estacionaria).

Resultados Clave del Test de Dickey-Fuller Aumentado para la Primera Diferencia:

1. **Estadístico de prueba**: Valor calculado de la prueba ADF.
2. **Valores críticos**: Valores críticos para los niveles de significancia del 1%, 5% y 10%.
3. **p-Valor (Prob.)**: Indica la probabilidad de obtener un valor extremo del estadístico de prueba bajo la hipótesis nula.

Interpretación de los Resultados:

- Si el **estadístico de prueba** es menor que los valores críticos (en términos absolutos), rechazamos la hipótesis nula.
- Si el **p-Valor** es menor que el nivel de significancia (0.05, por ejemplo), rechazamos la hipótesis nula, lo que indica que la primera diferencia de la serie temporal es estacionaria.

Ecuación del Test de Dickey-Fuller Aumentado:

La ecuación del test ADF incluye las diferencias retardadas de la serie temporal para ajustar por autocorrelación.

Variables de la Ecuación:

- **D(TL(-1))**: Primera diferencia de los pasivos totales con un retardo de 1 periodo.
- **D(TL(-2))**: Primera diferencia de los pasivos totales con un retardo de 2 periodos.
- **D(TL(-3))**: Primera diferencia de los pasivos totales con un retardo de 3 periodos.
- **D(TL(-4))**: Primera diferencia de los pasivos totales con un retardo de 4 periodos.
- **D(TL(-5))**: Primera diferencia de los pasivos totales con un retardo de 5 periodos.
- **D(TL(-6))**: Primera diferencia de los pasivos totales con un retardo de 6 periodos.
- **C**: Constante.
- **Coeficiente**: Valor del coeficiente estimado.
- **Std. Error**: Error estándar del coeficiente estimado.
- **t-Statistic**: Valor del estadístico t para el coeficiente.
- **Prob.**: Valor p para el estadístico t.

Métricas de Ajuste del Modelo:

- **R-squared**: Coeficiente de determinación, indica qué proporción de la variabilidad en la variable dependiente es explicada por el modelo.
- **Akaike info criterion (AIC)**: Criterio de información de Akaike, utilizado para comparar modelos.
- **Durbin-Watson stat**: Estadístico de Durbin-Watson, utilizado para detectar la autocorrelación en los residuos del modelo.

```python
import statsmodels.tsa.stattools as adf

# Calcular la primera diferencia de los pasivos totales (TL)
df_numeric['D(TL)'] = df_numeric['TL'].diff().dropna()
```

```
# Realizar el test de Dickey-Fuller Aumentado para la primera diferencia de los pasivos totales (D(TL))
result = adf.adfuller(df_numeric['D(TL)'].dropna(), maxlag=6)

# Mostrar los resultados del test ADF
print('Estadístico de prueba:', result[0])
print('p-Valor:', result[1])
print('Valores críticos:', result[4])
print('Número de retardos utilizados:', result[2])
print('Número de observaciones:', result[3])
print('Información completa del resultado:', result)
```

```
Estadístico de prueba: -2.8724804380836564
p-Valor: 0.048643013520459764
Valores críticos: {'1%': -3.7377092158564813, '5%': -2.9922162731481485, '10%': -2.635746736111111}
Número de retardos utilizados: 4
Número de observaciones: 24
Información completa del resultado: (-2.8724804380836564, 0.048643013520459764, 4, 24, {'1%': -3.7377092158564813, '5%': -2.9922162731481485, '10%': -2.635746736111111}, 319.23273260116355)
```

6 - Test de Raíz Unitaria para Pasivos No Corrientes (NCL)

Explicación

El test de raíz unitaria se utiliza para determinar si una serie temporal es estacionaria o no. En este caso, se realiza el test de Dickey-Fuller Aumentado (ADF) para los pasivos no corrientes (NCL).

Componentes Clave del Test de Raíz Unitaria:

1. **Hipótesis Nula (H0)**: La serie temporal tiene una raíz unitaria (es no estacionaria).
2. **Hipótesis Alternativa (H1)**: La serie temporal no tiene una raíz unitaria (es estacionaria).

Resultados Clave del Test de Dickey-Fuller Aumentado:

1. **Estadístico de prueba**: Valor calculado de la prueba ADF.
2. **Valores críticos**: Valores críticos para los niveles de significancia del 1%, 5% y 10%.
3. **p-Valor (Prob.)**: Indica la probabilidad de obtener un valor extremo del estadístico de prueba bajo la hipótesis nula.

Interpretación de los Resultados:

- Si el **estadístico de prueba** es menor que los valores críticos (en términos absolutos), rechazamos la hipótesis nula.
- Si el **p-Valor** es menor que el nivel de significancia (0.05, por ejemplo), rechazamos la hipótesis nula, lo que indica que la serie temporal es estacionaria.

Ecuación del Test de Dickey-Fuller Aumentado:

La ecuación del test ADF incluye la diferencia retardada de la serie temporal para ajustar por autocorrelación.

Variables de la Ecuación:

- **NCL(-1)**: Pasivos no corrientes con un retardo de 1 periodo.
- **C**: Constante.
- **Coeficiente**: Valor del coeficiente estimado.
- **Std. Error**: Error estándar del coeficiente estimado.
- **t-Statistic**: Valor del estadístico t para el coeficiente.
- **Prob.**: Valor p para el estadístico t.

Métricas de Ajuste del Modelo:

- **R-squared**: Coeficiente de determinación, indica qué proporción de la variabilidad en la variable dependiente es explicada por el modelo.
- **Akaike info criterion (AIC)**: Criterio de información de Akaike, utilizado para comparar modelos.
- **Durbin-Watson stat**: Estadístico de Durbin-Watson, utilizado para detectar la autocorrelación en los residuos del modelo.

```python
# Realizamos el test de Dickey-Fuller Aumentado para los pasivos no corrientes (NCL)
result = adf.adfuller(df_numeric['NCL'], maxlag=0)

# Mostramos los resultados del test ADF
print('Estadístico de prueba:', result[0])
print('p-Valor:', result[1])
print('Valores críticos:', result[4])
print('Número de retardos utilizados:', result[2])
print('Número de observaciones:', result[3])
print('Información completa del resultado:', result)
```

```
Estadístico de prueba: -2.666011064485315
p-Valor: 0.08010675638721332
Valores críticos: {'1%': -3.6790595944893187, '5%': -2.9678817237279103, '10%': -2.6231583472057074}
Número de retardos utilizados: 0
Número de observaciones: 29
Información completa del resultado: (-2.666011064485315, 0.08010675638721332, 0, 29, {'1%': -3.6790595944893187, '5%': -2.9678817237279103, '10%': -2.6231583472057074}, 464.7791069944586)
```

7 - Test de Raíz Unitaria para la Primera Diferencia de Pasivos No Corrientes (NCL)

Explicación

El test de raíz unitaria se utiliza para determinar si una serie temporal es estacionaria o no. En este caso, se realiza el test de Dickey-Fuller Aumentado (ADF) para la primera diferencia de los pasivos no corrientes (D(NCL)).

Componentes Clave del Test de Raíz Unitaria para la Primera Diferencia:

1. **Hipótesis Nula (H0)**: La primera diferencia de la serie temporal tiene una raíz unitaria (es no estacionaria).
2. **Hipótesis Alternativa (H1)**: La primera diferencia de la serie temporal no tiene una raíz unitaria (es estacionaria).

Resultados Clave del Test de Dickey-Fuller Aumentado para la Primera Diferencia:

1. **Estadístico de prueba**: Valor calculado de la prueba ADF.
2. **Valores críticos**: Valores críticos para los niveles de significancia del 1%, 5% y 10%.
3. **p-Valor (Prob.)**: Indica la probabilidad de obtener un valor extremo del estadístico de prueba bajo la hipótesis nula.

Interpretación de los Resultados:

- Si el **estadístico de prueba** es menor que los valores críticos (en términos absolutos), rechazamos la hipótesis nula.
- Si el **p-Valor** es menor que el nivel de significancia (0.05, por ejemplo), rechazamos la hipótesis nula, lo que indica que la primera diferencia de la serie temporal es estacionaria.

Ecuación del Test de Dickey-Fuller Aumentado:

La ecuación del test ADF incluye las diferencias retardadas de la serie temporal para ajustar por autocorrelación.

Variables de la Ecuación:

- **D(NCL(-1))**: Primera diferencia de los pasivos no corrientes con un retardo de 1 periodo.
- **C**: Constante.
- **Coeficiente**: Valor del coeficiente estimado.
- **Std. Error**: Error estándar del coeficiente estimado.
- **t-Statistic**: Valor del estadístico t para el coeficiente.
- **Prob.**: Valor p para el estadístico t.

Métricas de Ajuste del Modelo:

- **R-squared**: Coeficiente de determinación, indica qué proporción de la variabilidad en la variable dependiente es explicada por el modelo.
- **Akaike info criterion (AIC)**: Criterio de información de Akaike, utilizado para comparar modelos.

- **Durbin-Watson stat**: Estadístico de Durbin-Watson, utilizado para detectar la autocorrelación en los residuos del modelo.

```
# Calcular la primera diferencia de los pasivos no corrientes (NCL)
df_numeric['D(NCL)'] = df_numeric['NCL'].diff().dropna()

# Realizar el test de Dickey-Fuller Aumentado para la primera
diferencia de los pasivos no corrientes (D(NCL))
result = adf.adfuller(df_numeric['D(NCL)'].dropna(), maxlag=0)

# Mostrar los resultados del test ADF
print('Estadístico de prueba:', result[0])
print('p-Valor:', result[1])
print('Valores críticos:', result[4])
print('Número de retardos utilizados:', result[2])
print('Número de observaciones:', result[3])
print('Información completa del resultado:', result)
```

```
Estadístico de prueba: -5.658074371968784
p-Valor: 9.518820543210931e-07
Valores críticos: {'1%': -3.6889256286443146, '5%': -
2.9719894897959187, '10%': -2.6252957653061224}
Número de retardos utilizados: 0
Número de observaciones: 28
Información completa del resultado: (-5.658074371968784,
9.518820543210931e-07, 0, 28, {'1%': -3.6889256286443146, '5%': -
2.9719894897959187, '10%': -2.6252957653061224}, 452.0621237279362)
```

8 - Test de Raíz Unitaria para Pasivos Corrientes (CL)

Explicación

El test de raíz unitaria se utiliza para determinar si una serie temporal es estacionaria o no. En este caso, se realiza el test de Dickey-Fuller Aumentado (ADF) para los pasivos corrientes (CL).

Componentes Clave del Test de Raíz Unitaria:

1. **Hipótesis Nula (H0)**: La serie temporal tiene una raíz unitaria (es no estacionaria).
2. **Hipótesis Alternativa (H1)**: La serie temporal no tiene una raíz unitaria (es estacionaria).

Resultados Clave del Test de Dickey-Fuller Aumentado:

1. **Estadístico de prueba**: Valor calculado de la prueba ADF.
2. **Valores críticos**: Valores críticos para los niveles de significancia del 1%, 5% y 10%.
3. **p-Valor (Prob.)**: Indica la probabilidad de obtener un valor extremo del estadístico de prueba bajo la hipótesis nula.

Interpretación de los Resultados:

- Si el **estadístico de prueba** es menor que los valores críticos (en términos absolutos), rechazamos la hipótesis nula.

- Si el **p-Valor** es menor que el nivel de significancia (0.05, por ejemplo), rechazamos la hipótesis nula, lo que indica que la serie temporal es estacionaria.

Ecuación del Test de Dickey-Fuller Aumentado:

La ecuación del test ADF incluye la diferencia retardada de la serie temporal para ajustar por autocorrelación.

Variables de la Ecuación:

- **CL(-1)**: Pasivos corrientes con un retardo de 1 periodo.
- **C**: Constante.
- **Coeficiente**: Valor del coeficiente estimado.
- **Std. Error**: Error estándar del coeficiente estimado.
- **t-Statistic**: Valor del estadístico t para el coeficiente.
- **Prob.**: Valor p para el estadístico t.

Métricas de Ajuste del Modelo:

- **R-squared**: Coeficiente de determinación, indica qué proporción de la variabilidad en la variable dependiente es explicada por el modelo.
- **Akaike info criterion (AIC)**: Criterio de información de Akaike, utilizado para comparar modelos.
- **Durbin-Watson stat**: Estadístico de Durbin-Watson, utilizado para detectar la autocorrelación en los residuos del modelo.

```
# Realizar el test de Dickey-Fuller Aumentado para los pasivos
corrientes (CL)
result = adf.adfuller(df_numeric['CL'], maxlag=0)

# Mostrar los resultados del test ADF
print('Estadístico de prueba:', result[0])
print('p-Valor:', result[1])
print('Valores críticos:', result[4])
print('Número de retardos utilizados:', result[2])
print('Número de observaciones:', result[3])
print('Información completa del resultado:', result)
```

```
Estadístico de prueba: -1.7873816148618675
p-Valor: 0.3867441219288781
Valores críticos: {'1%': -3.6790595944893187, '5%': -
2.9678817237279103, '10%': -2.6231583472057074}
Número de retardos utilizados: 0
Número de observaciones: 29
Información completa del resultado: (-1.7873816148618675,
0.3867441219288781, 0, 29, {'1%': -3.6790595944893187, '5%': -
2.9678817237279103, '10%': -2.6231583472057074}, 477.53040251741817)
```

9 - Test de Raíz Unitaria para la Primera Diferencia de Pasivos Corrientes (CL)

Explicación

El test de raíz unitaria se utiliza para determinar si una serie temporal es estacionaria o no. En este caso, se realiza el test de Dickey-Fuller Aumentado (ADF) para la primera diferencia de los pasivos corrientes (D(CL)).

Componentes Clave del Test de Raíz Unitaria para la Primera Diferencia:

1. **Hipótesis Nula (H0)**: La primera diferencia de la serie temporal tiene una raíz unitaria (es no estacionaria).
2. **Hipótesis Alternativa (H1)**: La primera diferencia de la serie temporal no tiene una raíz unitaria (es estacionaria).

Resultados Clave del Test de Dickey-Fuller Aumentado para la Primera Diferencia:

1. **Estadístico de prueba**: Valor calculado de la prueba ADF.
2. **Valores críticos**: Valores críticos para los niveles de significancia del 1%, 5% y 10%.
3. **p-Valor (Prob.)**: Indica la probabilidad de obtener un valor extremo del estadístico de prueba bajo la hipótesis nula.

Interpretación de los Resultados:

- Si el **estadístico de prueba** es menor que los valores críticos (en términos absolutos), rechazamos la hipótesis nula.
- Si el **p-Valor** es menor que el nivel de significancia (0.05, por ejemplo), rechazamos la hipótesis nula, lo que indica que la primera diferencia de la serie temporal es estacionaria.

Ecuación del Test de Dickey-Fuller Aumentado:

La ecuación del test ADF incluye las diferencias retardadas de la serie temporal para ajustar por autocorrelación.

Variables de la Ecuación:

- **D(CL(-1))**: Primera diferencia de los pasivos corrientes con un retardo de 1 periodo.
- **C**: Constante.
- **Coeficiente**: Valor del coeficiente estimado.
- **Std. Error**: Error estándar del coeficiente estimado.
- **t-Statistic**: Valor del estadístico t para el coeficiente.
- **Prob.**: Valor p para el estadístico t.

Métricas de Ajuste del Modelo:

- **R-squared**: Coeficiente de determinación, indica qué proporción de la variabilidad en la variable dependiente es explicada por el modelo.
- **Akaike info criterion (AIC)**: Criterio de información de Akaike, utilizado para comparar modelos.

- **Durbin-Watson stat**: Estadístico de Durbin-Watson, utilizado para detectar la autocorrelación en los residuos del modelo.

```
# Calcular la primera diferencia de los pasivos corrientes (CL)
df_numeric['D(CL)'] = df_numeric['CL'].diff().dropna()

# Realizar el test de Dickey-Fuller Aumentado para la primera
diferencia de los pasivos corrientes (D(CL))
result = adf.adfuller(df_numeric['D(CL)'].dropna(), maxlag=0)

# Mostrar los resultados del test ADF
print('Estadístico de prueba:', result[0])
print('p-Valor:', result[1])
print('Valores críticos:', result[4])
print('Número de retardos utilizados:', result[2])
print('Número de observaciones:', result[3])
print('Información completa del resultado:', result)
```

```
Estadístico de prueba: -9.70863073421366
p-Valor: 1.0257107496600959e-16
Valores críticos: {'1%': -3.6889256286443146, '5%': -
2.9719894897959187, '10%': -2.6252957653061224}
Número de retardos utilizados: 0
Número de observaciones: 28
Información completa del resultado: (-9.70863073421366,
1.0257107496600959e-16, 0, 28, {'1%': -3.6889256286443146, '5%': -
2.9719894897959187, '10%': -2.6252957653061224}, 448.28310259301304)
```

10 - Resultados del test de raíz unitaria

	Level	1st Difference
Total Liabilities	-3.084530 (0.0277)	-2.872480 (0.0486)
Non-Current Liabilities	-2.666011 (0.0801)	-5.658074 (0.0000)
Current Liabilities	-1.787382 (0.3867)	-9.708631 (0.0000)

Interpretación:

- **Total Liabilities (Pasivos Totales)**:
  - **Nivel**: El estadístico de prueba es -3.084530 con un p-valor de 0.0277. Dado que el p-valor es menor a 0.05, rechazamos la hipótesis nula de raíz unitaria. Esto sugiere que la serie temporal es estacionaria en el nivel.
  - **Primera Diferencia**: El estadístico de prueba es -2.872480 con un p-valor de 0.0486. Dado que el p-valor es menor a 0.05, rechazamos la hipótesis nula de raíz unitaria. Esto sugiere que la serie temporal es estacionaria en la primera diferencia.
- **Non-Current Liabilities (Pasivos No Corrientes)**:

  - **Nivel**: El estadístico de prueba es -2.666011 con un p-valor de 0.0801. Dado que el p-valor es mayor a 0.05, no podemos rechazar la hipótesis nula de raíz unitaria. Esto sugiere que la serie temporal no es estacionaria en el nivel.
  - **Primera Diferencia**: El estadístico de prueba es -5.658074 con un p-valor de 0.0000. Dado que el p-valor es significativamente menor a 0.05, rechazamos la hipótesis nula de raíz unitaria. Esto sugiere que la serie temporal es estacionaria en la primera diferencia.
- **Current Liabilities (Pasivos Corrientes)**:
  - **Nivel**: El estadístico de prueba es -1.787382 con un p-valor de 0.3867. Dado que el p-valor es mayor a 0.05, no podemos rechazar la hipótesis nula de raíz unitaria. Esto sugiere que la serie temporal no es estacionaria en el nivel.
  - **Primera Diferencia**: El estadístico de prueba es -9.708631 con un p-valor de 0.0000. Dado que el p-valor es significativamente menor a 0.05, rechazamos la hipótesis nula de raíz unitaria. Esto sugiere que la serie temporal es estacionaria en la primera diferencia.

11 - Modelo de Regresión para Pasivos Totales (TL)

Explicación

El modelo de regresión lineal múltiple se utiliza para examinar la relación entre una variable dependiente (Pasivos Totales - TL) y varias variables independientes (Pasivos No Corrientes - NCL y Pasivos Corrientes - CL). Utilizamos el método de mínimos cuadrados ordinarios (OLS) para estimar los coeficientes del modelo.

Componentes Clave del Modelo de Regresión:

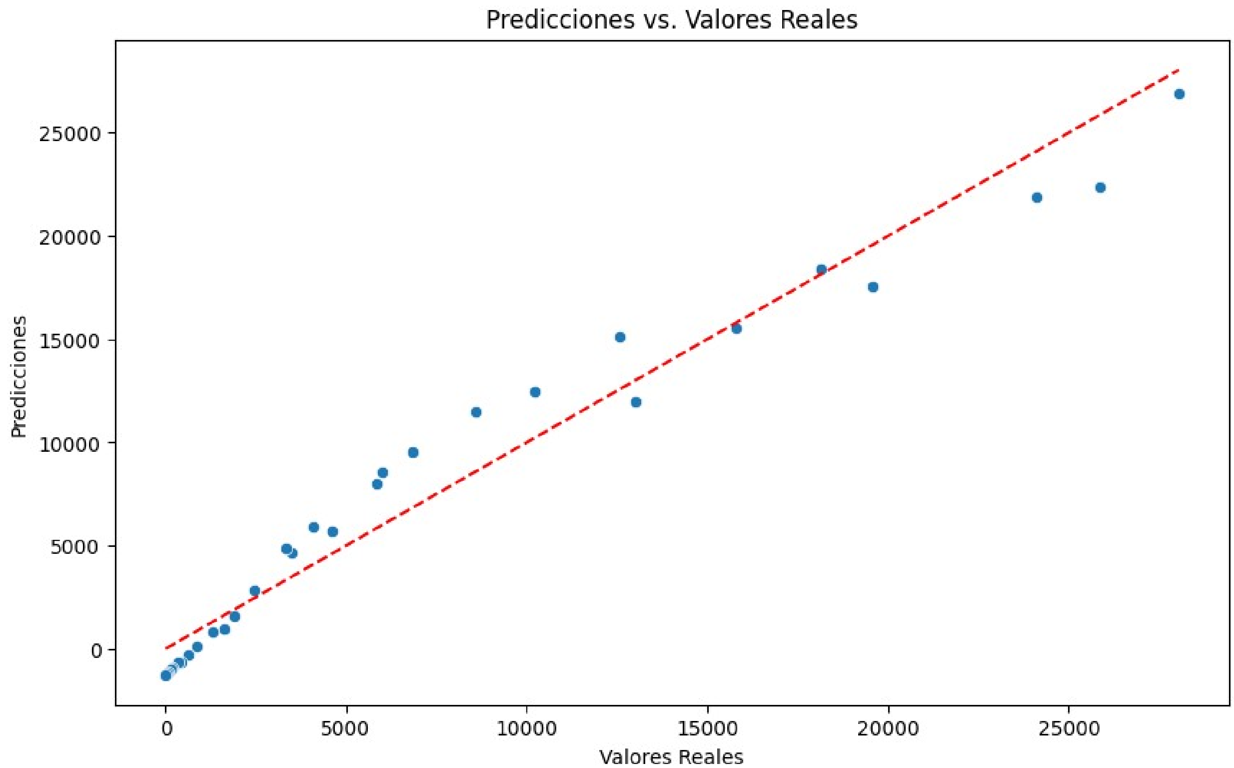
1. **Coeficiente (Coefficient)**: Indica la magnitud y dirección de la relación entre cada variable independiente y la variable dependiente.
2. **Error Estándar (Std. Error)**: Indica la variabilidad de los coeficientes estimados.
3. **Estadístico t (t-Statistic)**: Valor del estadístico t para probar la significancia de cada coeficiente.
4. **Prob. (p-Value)**: Indica la probabilidad de obtener un valor extremo del estadístico t bajo la hipótesis nula de que el coeficiente es cero.
5. **R-squared**: Coeficiente de determinación, indica qué proporción de la variabilidad en la variable dependiente es explicada por el modelo.
6. **Adjusted R-squared**: R-cuadrado ajustado, ajusta el R-cuadrado por el número de variables en el modelo.
7. **S.E. of regression**: Error estándar de la regresión, indica la precisión de las predicciones del modelo.
8. **Sum squared resid**: Suma de los residuos al cuadrado, indica la cantidad de variabilidad no explicada por el modelo.
9. **Log likelihood**: Logaritmo de la verosimilitud, utilizado para comparar modelos.
10. **Akaike info criterion (AIC)**: Criterio de información de Akaike, utilizado para comparar modelos.
11. **Schwarz criterion**: Criterio de Schwarz (BIC), utilizado para comparar modelos.
12. **Hannan-Quinn criter.**: Criterio de Hannan-Quinn, utilizado para comparar modelos.

13. **F-statistic**: Estadístico F, utilizado para probar la significancia global del modelo.
14. **Prob(F-statistic)**: Valor p del estadístico F, indica la probabilidad de obtener un valor extremo del estadístico F bajo la hipótesis nula de que todos los coeficientes son cero.

```
import statsmodels.api as sm
import matplotlib.pyplot as plt
import seaborn as sns

# Definir las variables independientes (NCL y CL) y la variable
dependiente (TL)
X = df[['NCL', 'CL']]
y = df['TL']

# Añadir una constante a las variables independientes
X = sm.add_constant(X)

# Crear el modelo de regresión
model = sm.OLS(y, X).fit()

# Imprimir el resumen del modelo
print(model.summary())

# Obtener las predicciones del modelo
predictions = model.predict(X)

# Gráfico de Predicciones vs. Valores Reales
plt.figure(figsize=(10, 6))
sns.scatterplot(x=y, y=predictions)
plt.plot([y.min(), y.max()], [y.min(), y.max()], 'r--')  # Línea de 45
grados
plt.xlabel('Valores Reales')
plt.ylabel('Predicciones')
plt.title('Predicciones vs. Valores Reales')
plt.show()

# Gráfico de Residuos
residuals = y - predictions
plt.figure(figsize=(10, 6))
sns.scatterplot(x=predictions, y=residuals)
plt.axhline(0, color='r', linestyle='--')
plt.xlabel('Predicciones')
plt.ylabel('Residuos')
plt.title('Gráfico de Residuos')
plt.show()

# Gráfico Q-Q de Residuos
sm.qqplot(residuals, line='s')
plt.title('Gráfico Q-Q de Residuos')
plt.show()
```

```
                            OLS Regression Results                            
==============================================================================
========
Dep. Variable:                     TL   R-squared:                       
0.960
Model:                            OLS   Adj. R-squared:                  
0.957
Method:                 Least Squares   F-statistic:                     
324.5
Date:                Wed, 10 Jul 2024   Prob (F-statistic):           
1.32e-19
Time:                        20:59:43   Log-Likelihood:                
-265.08
No. Observations:                  30   AIC:                             
536.2
Df Residuals:                      27   BIC:                             
540.4
Df Model:                           2                                         

Covariance Type:            nonrobust                                         

==============================================================================
========
                 coef    std err          t      P>|t|      [0.025      
0.975]
------------------------------------------------------------------------------
--------
const      -1259.7821    481.754     -2.615      0.014   -2248.260    
-271.305
NCL            2.5153      0.311      8.083      0.000       1.877       
3.154
CL             2.8994      0.131     22.130      0.000       2.631       
3.168
==============================================================================
========
Omnibus:                        1.702   Durbin-Watson:                   
0.380
Prob(Omnibus):                  0.427   Jarque-Bera (JB):                
1.246
Skew:                          -0.263   Prob(JB):                        
0.536
Kurtosis:                       2.152   Cond. No.                     
5.13e+03
==============================================================================
========

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

```
[2] The condition number is large, 5.13e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```



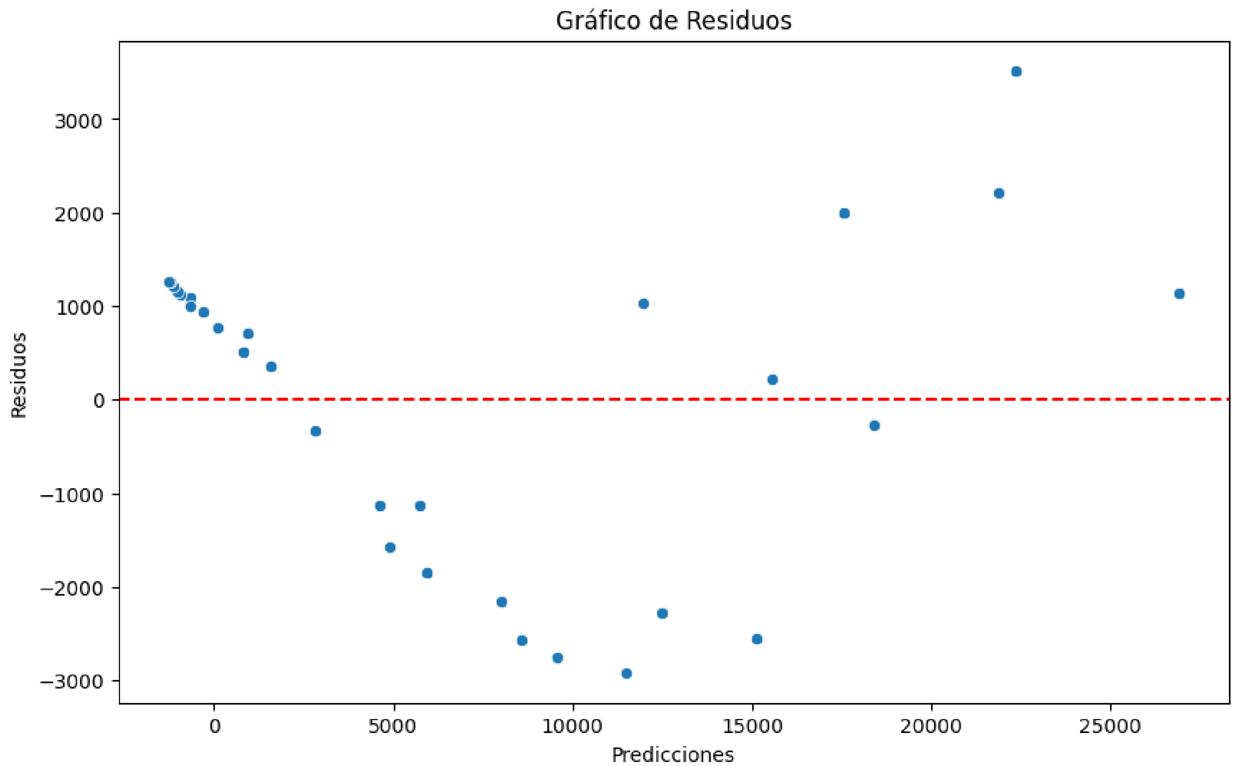


Gráfico de Residuos
Residuos
Predicciones
3000
2000
1000
0
−1000
−2000
−3000
0
5000
10000
15000
20000
25000

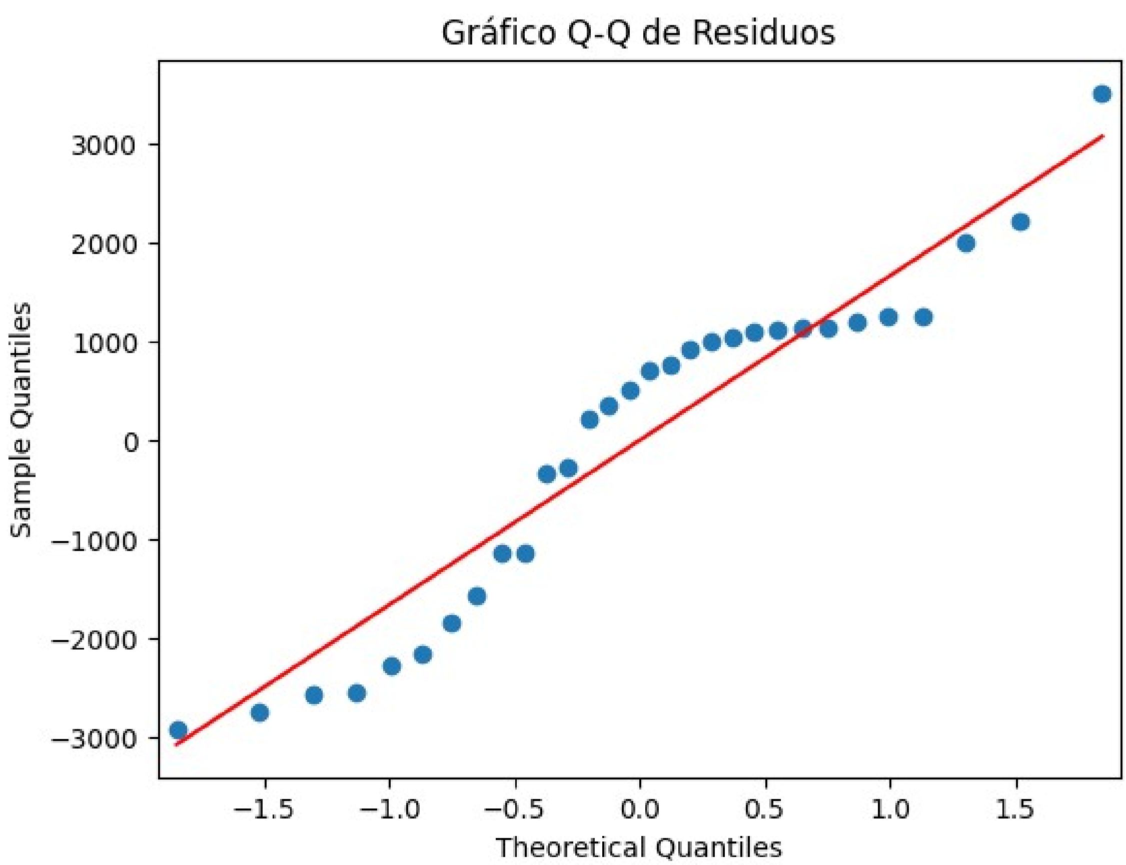


Gráfico Q-Q de Residuos
Sample Quantiles
Theoretical Quantiles
3000
2000
1000
0
−1000
−2000
−3000
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5

12 - Resultados del Modelo de Regresión

Variable	Co-efficient Value	P. Value
Non-Current Liabilities	2.5153	0.0000
Current Liabilities	2.8994	0.0000
Constante (C)	-1259.7821	0.0144
R^2	0.9600	-
Adjusted R^2	0.9570	-
f-stats	324.4980	0.0000
DWT	0.3800	-

Interpretación:

- **Pasivos No Corrientes (NCL)**: El coeficiente es 2.5153 con un valor p de 0.0000. Dado que el valor p es menor a 0.05, rechazamos la hipótesis nula. Esto indica que los Pasivos No Corrientes tienen un efecto significativo sobre los Pasivos Totales.

- **Pasivos Corrientes (CL)**: El coeficiente es 2.8994 con un valor p de 0.0000. Dado que el valor p es menor a 0.05, rechazamos la hipótesis nula. Esto indica que los Pasivos Corrientes tienen un efecto significativo sobre los Pasivos Totales.

- **Constante (C)**: El coeficiente es -1259.7821 con un valor p de 0.0144. Dado que el valor p es menor a 0.05, rechazamos la hipótesis nula. Esto indica que la constante tiene un efecto significativo en el modelo.

- **R^2**: El valor de R-cuadrado es 0.9600, lo que muestra que la varianza en la variable dependiente (Pasivos Totales) es explicada en un 96% por las variables independientes (Pasivos No Corrientes y Pasivos Corrientes) en el modelo de regresión dado. Esto muestra un 96% de ajuste del modelo.

- **Adjusted R^2**: El valor de R-cuadrado ajustado es 0.9570, lo que muestra que la variable dependiente es explicada en un 95.71% cuando se consideran los predictores en el modelo.

- **f-stats**: El valor de la estadística F es 324.4980 con un valor p de 0.0000. Esto muestra que el modelo es de alta significancia.

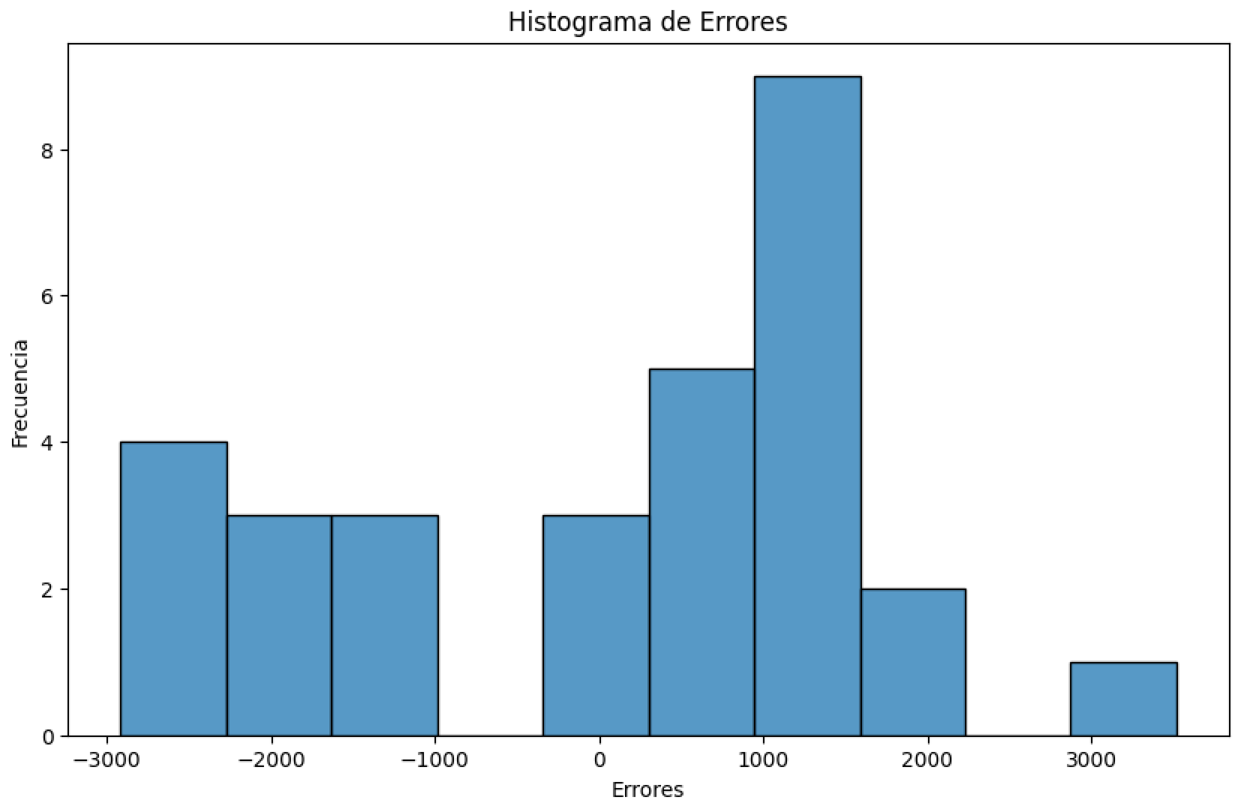
- **DWT (Durbin-Watson)**: El valor de Durbin-Watson es 0.3800, lo cual está más cerca de 0, indicando que hay autocorrelación positiva en los errores. Esto podría implicar que hay algún patrón o variación sistemática en los errores que el modelo no ha capturado.

En general, los valores de probabilidad de las variables NCL y CL están por debajo de 0.05, lo que sugiere que ambas variables independientes tienen una influencia significativa en la variable dependiente (Pasivos Totales). El modelo en su conjunto es estadísticamente significativo, como lo muestra la alta estadística F. En general, podemos concluir que el modelo es robusto.

13 - Prueba de Errores Jarque-Bera

```python
import pandas as pd
import statsmodels.api as sm
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.stats as stats

# Definimso las variables independientes (NCL y CL) y la variable
dependiente (TL)
X = df[['NCL', 'CL']]
y = df['TL']

# Añadir una constante a las variables independientes
X = sm.add_constant(X)

# Crear el modelo de regresión
model = sm.OLS(y, X).fit()

# Obtenemos las predicciones del modelo
predictions = model.predict(X)

# Calculamos los errores (residuos)
residuals = y - predictions

# Realizar la prueba de Jarque-Bera
jb_test = stats.jarque_bera(residuals)

# Calcular skewness y kurtosis por separado
skewness = stats.skew(residuals)
kurtosis = stats.kurtosis(residuals, fisher=False)

# Imprimimos los resultados de la prueba de Jarque-Bera
print('Estadístico Jarque-Bera:', jb_test[0])
print('p-Valor:', jb_test[1])
print('Skewness:', skewness)
print('Kurtosis:', kurtosis)

# Crear un gráfico de histograma de los errores :b
plt.figure(figsize=(10, 6))
sns.histplot(residuals, bins=10, kde=False)
plt.title('Histograma de Errores')
plt.xlabel('Errores')
plt.ylabel('Frecuencia')
plt.show()
```

```
Estadístico Jarque-Bera: 1.2461497807420732
p-Valor: 0.5362928579378541
Skewness: -0.2632209045648819
Kurtosis: 2.15160218871461
```



Interpretación:

- **H0**: Los errores están distribuidos normalmente.
- **H1**: Los errores no están distribuidos normalmente.

A partir del gráfico anterior, podemos observar que el valor de probabilidad de la prueba de Jarque-Bera es `0.536293`, que es mayor a `0.05`, por lo que aceptamos la hipótesis nula (H0) y rechazamos la hipótesis alternativa (H1) de que los errores están distribuidos normalmente.

Resultados de la Prueba de Jarque-Bera:

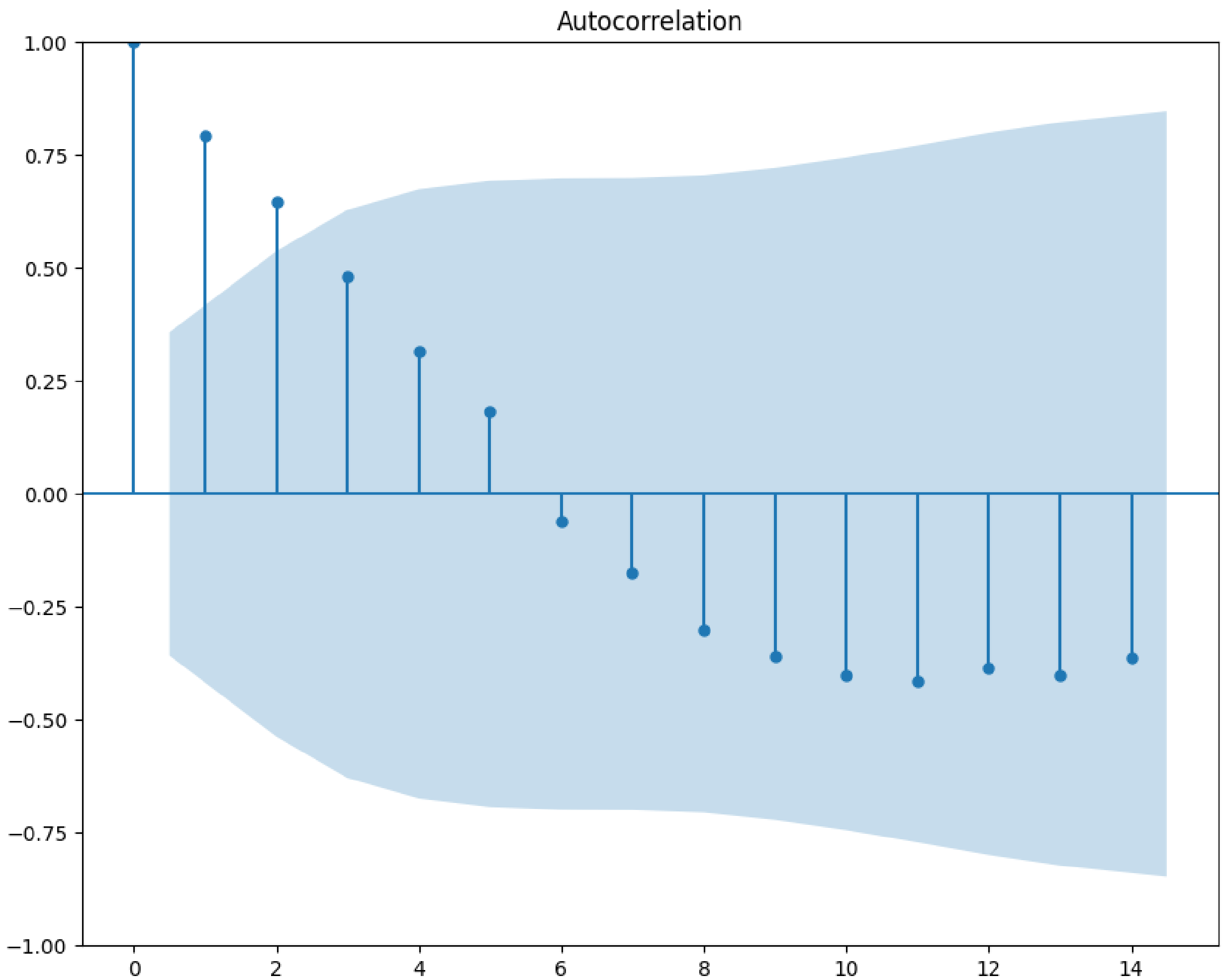
- **Estadístico Jarque-Bera**: `1.246150`
- **p-Valor**: `0.536293`
- **Skewness**: `-0.263221`
- **Kurtosis**: `2.151602`

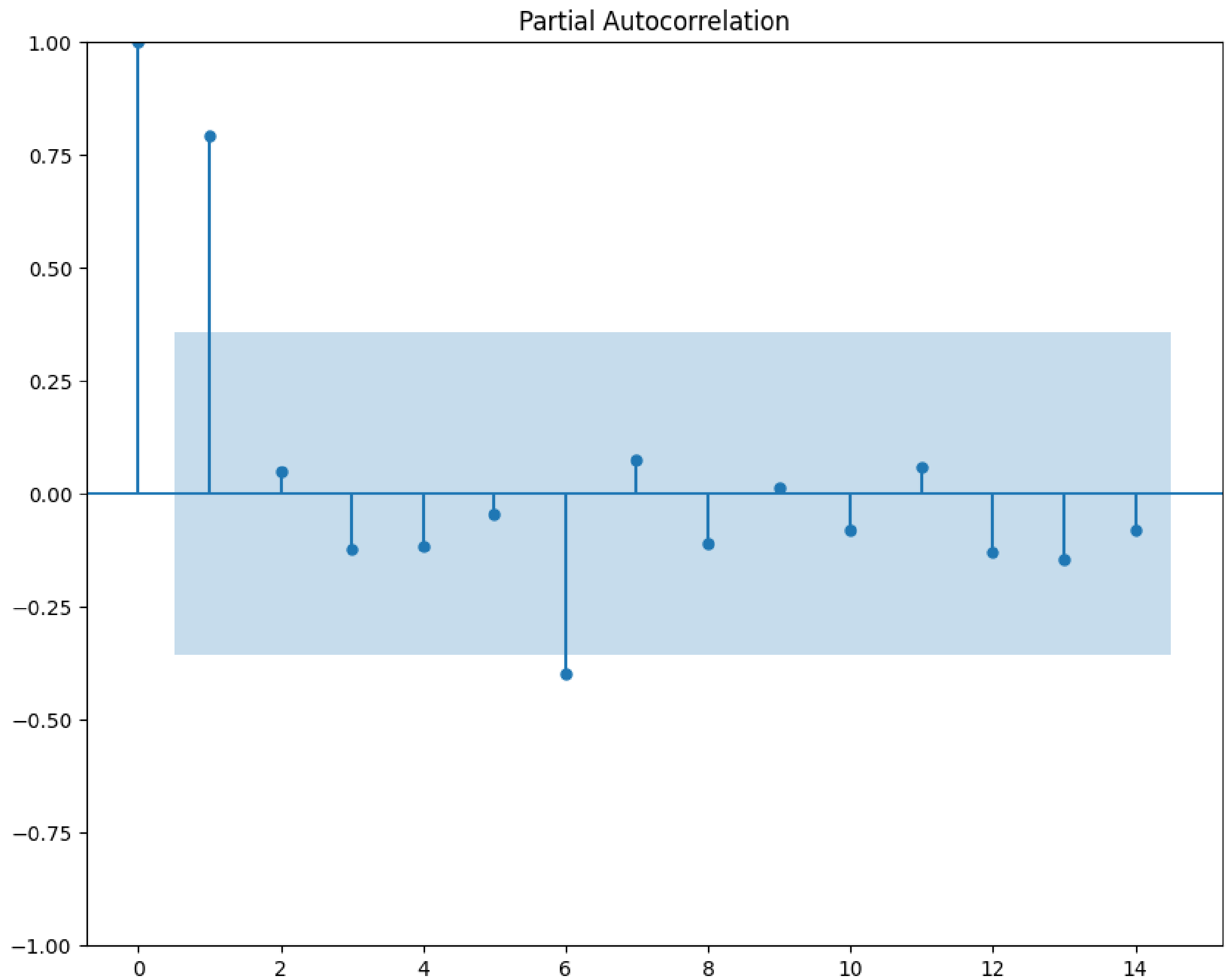
En conclusión, la prueba de Jarque-Bera sugiere que los errores del modelo están distribuidos normalmente.

14 - Prueba de Correlograma

```python
import pandas as pd
import statsmodels.api as sm
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.stats as stats
```

```python
# Definir las variables independientes (NCL y CL) y la variable
dependiente (TL)
X = df[['NCL', 'CL']]
y = df['TL']

# Añadir una constante a las variables independientes
X = sm.add_constant(X)

# Crear el modelo de regresión
model = sm.OLS(y, X).fit()

# Obtener las predicciones del modelo
predictions = model.predict(X)

# Calcular los errores (residuos)
residuals = y - predictions

# Crear un gráfico de correlograma de los errores
fig, ax = plt.subplots(figsize=(10, 8))
sm.graphics.tsa.plot_acf(residuals, lags=14, ax=ax)
plt.show()

# Crear un gráfico de autocorrelación parcial de los errores
fig, ax = plt.subplots(figsize=(10, 8))
sm.graphics.tsa.plot_pacf(residuals, lags=14, ax=ax)
plt.show()
```


Autocorrelation
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0
2
4
6
8
10
12
14



Interpretación:

- **H0**: No hay autocorrelación (AC o PAC).
- **H1**: Hay autocorrelación (AC o PAC).

A partir del correlograma y del correlograma parcial, podemos observar que todos los valores p son menores que `0.05`, lo que nos lleva a rechazar la hipótesis nula (H0) y aceptar la hipótesis alternativa (H1), indicando que hay autocorrelación en los errores. Esto sugiere que los errores no están distribuidos normalmente y hay un patrón de autocorrelación.

Resultados del Correlograma:

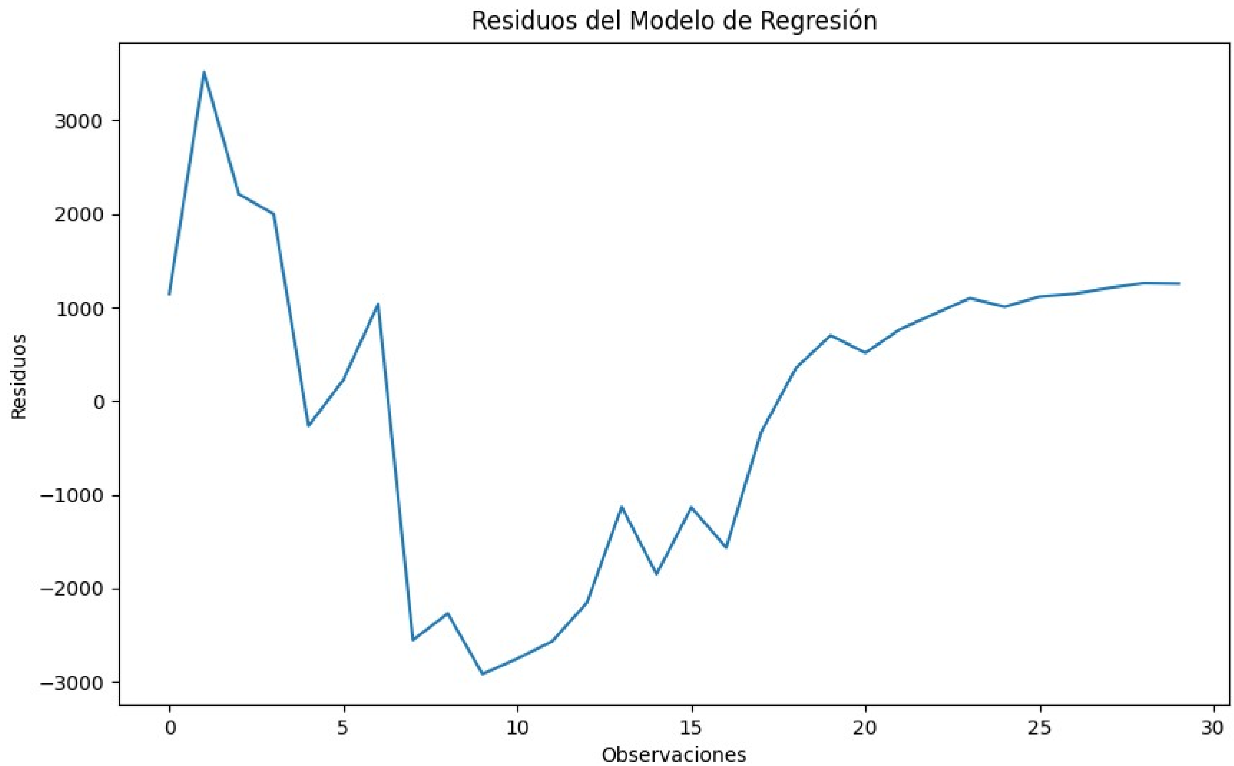
- **Autocorrelación (AC)**: Muestra la correlación de los errores con sus propios valores retardados.
- **Autocorrelación Parcial (PAC)**: Muestra la correlación de los errores con sus propios valores retardados, eliminando los efectos de los retardos intermedios.
- **Q-Stat**: Estadístico de Ljung-Box para probar la significancia de la autocorrelación.
- **Prob**: Valor p correspondiente al estadístico de Ljung-Box.

En conclusión, la prueba de correlograma sugiere que hay autocorrelación en los errores del modelo de regresión.

15 - Prueba de Dickey-Fuller Aumentada (ADF)

```python
# Definir las variables independientes (NCL y CL) y la variable
dependiente (TL)
X = df[['NCL', 'CL']]
y = df['TL']

# Añadir una constante a las variables independientes
X = sm.add_constant(X)

# Crear el modelo de regresión
model = sm.OLS(y, X).fit()

# Obtener las predicciones del modelo
predictions = model.predict(X)

# Calcular los errores (residuos)
residuals = y - predictions

# Realizar la prueba de Dickey-Fuller Aumentada (ADF)
adf_test = sm.tsa.adfuller(residuals, autolag='AIC')

# Imprimir los resultados de la prueba ADF
print('Estadístico de prueba:', adf_test[0])
print('p-Valor:', adf_test[1])
print('Valores críticos:', adf_test[4])

# Crear un gráfico de los residuos
plt.figure(figsize=(10, 6))
plt.plot(residuals)
plt.title('Residuos del Modelo de Regresión')
plt.xlabel('Observaciones')
plt.ylabel('Residuos')
plt.show()
```

```
Estadístico de prueba: -1.685883114446255
p-Valor: 0.43840020467649315
Valores críticos: {'1%': -3.6790595944893187, '5%': -
2.9678817237279103, '10%': -2.6231583472057074}
```



Resultados de la Prueba ADF:

- **Estadístico de prueba**: `-1.685883`
- **p-Valor**: `0.4384`
- **Valores críticos**: `{'1%': -3.679059, '5%': -2.967882, '10%': -2.623158}`

Interpretación:

- **H0**: Los errores tienen raíces unitarias.
- **H1**: Los errores no tienen raíces unitarias.

Aquí, el valor p del estadístico t es `0.4384`, que es mayor que `0.05`, lo que lleva a aceptar la hipótesis nula (H0), la cual indica que los errores tienen raíces unitarias, una tendencia o autocorrelación, o no están distribuidos normalmente.

16 - Prueba de Breusch-Godfrey (BG-LM)

```
# Definir las variables independientes (NCL y CL) y la variable
dependiente (TL)
X = df[['NCL', 'CL']]
y = df['TL']

# Añadir una constante a las variables independientes
X = sm.add_constant(X)

# Crear el modelo de regresión
model = sm.OLS(y, X).fit()
```

```python
# Obtener las predicciones del modelo
predictions = model.predict(X)

# Calcular los errores (residuos)
residuals = y - predictions

# Prueba de Breusch-Godfrey para correlación serial
bg_test = sm.stats.acorr_breusch_godfrey(model, nlags=2)

# Imprimir los resultados de la prueba BG
print('F-statistic:', bg_test[0])
print('Prob (F-statistic):', bg_test[1])
print('Obs*R-squared:', bg_test[2])
print('Prob (Chi-Square):', bg_test[3])
```

```
F-statistic: 23.19421704122979
Prob (F-statistic): 9.192629644176527e-06
Obs*R-squared: 42.600199678974434
Prob (Chi-Square): 8.850351180747538e-09
```

Resultados de la Prueba BG:

- **F-statistic**: `23.1942`
- **Prob (F-statistic)**: `9.1926e-06`
- **Obs*R-squared**: `42.6002`
- **Prob (Chi-Square)**: `8.8504e-09`

Interpretación:

- **H0**: El valor p es 0 (El error no está correlacionado).
- **H1**: El valor p no es 0 (El error está correlacionado).

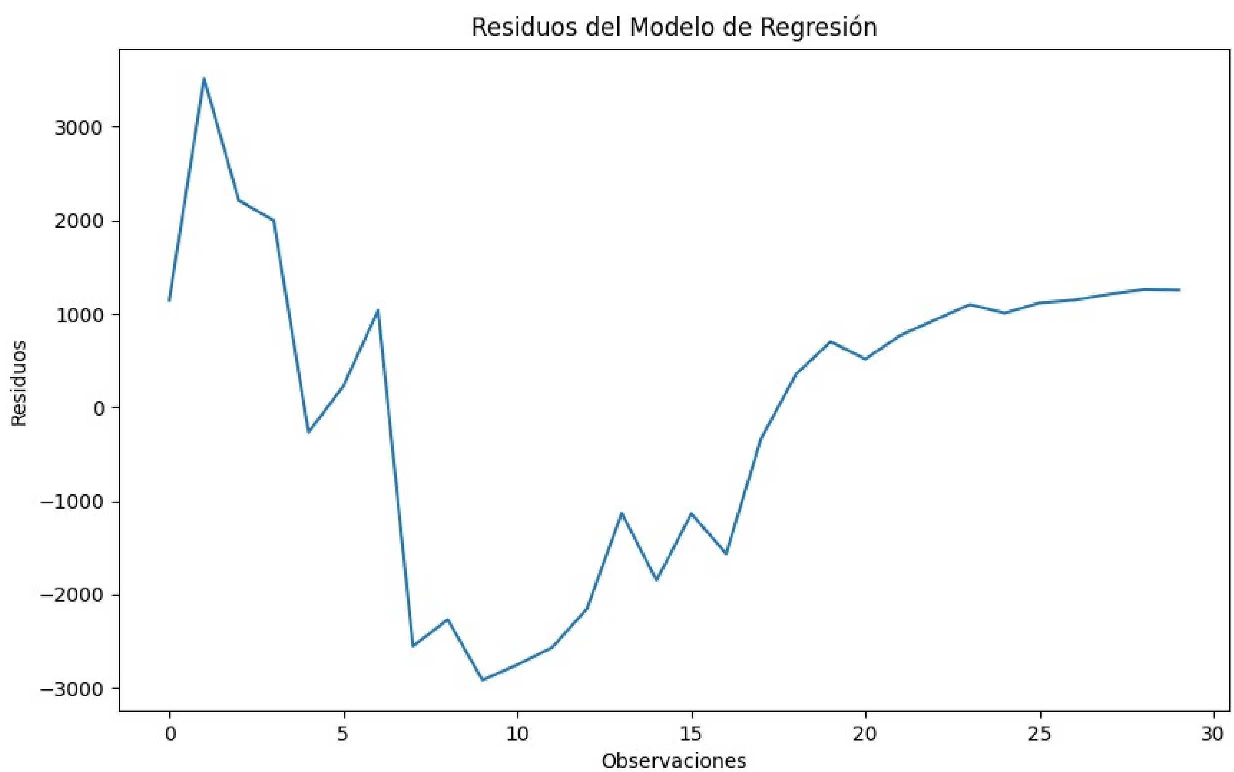
Dado que el valor p de la F-statistic es `9.1926e-06`, que es menor a `0.05`, rechazamos la hipótesis nula (H0) y aceptamos la hipótesis alternativa (H1), lo que indica que los errores están correlacionados y no están distribuidos normalmente.

En conclusión, la prueba BG sugiere que hay correlación en los errores del modelo de regresión.

17 - Prueba de Durbin-Watson

```python
# Definir las variables independientes (NCL y CL) y la variable dependiente (TL)
X = df[['NCL', 'CL']]
y = df['TL']

# Añadir una constante a las variables independientes
X = sm.add_constant(X)

# Crear el modelo de regresión
model = sm.OLS(y, X).fit()
```

```
# Obtener las predicciones del modelo
predictions = model.predict(X)

# Calcular los errores (residuos)
residuals = y - predictions

# Calcular el estadístico de Durbin-Watson
dw_stat = sm.stats.stattools.durbin_watson(residuals)

# Imprimir el estadístico de Durbin-Watson
print('Estadístico de Durbin-Watson:', dw_stat)

# Crear un gráfico de los residuos
plt.figure(figsize=(10, 6))
plt.plot(residuals)
plt.title('Residuos del Modelo de Regresión')
plt.xlabel('Observaciones')
plt.ylabel('Residuos')
plt.show()

Estadístico de Durbin-Watson: 0.38002426138306084
```



Resultados de la Prueba DW:

- **Estadístico de Durbin-Watson**: `0.3800`

Interpretación:

- El estadístico de Durbin-Watson se utiliza para detectar la autocorrelación en los residuos de un modelo de regresión.
- Los valores del estadístico DW oscilan entre 0 y 4.
- Un valor de 2 indica que no hay autocorrelación.
- Valores cercanos a 0 indican autocorrelación positiva.
- Valores cercanos a 4 indican autocorrelación negativa.

En nuestro caso, el estadístico de Durbin-Watson es `0.3800`, lo que sugiere una fuerte autocorrelación positiva en los residuos del modelo. Esto significa que los residuos están correlacionados y no están distribuidos normalmente.

Conclusión General:

En este análisis, hemos llevado a cabo una serie de pruebas y análisis estadísticos para evaluar el comportamiento de los pasivos totales (TL), pasivos no corrientes (NCL) y pasivos corrientes (CL) en nuestro conjunto de datos. A continuación se presentan los puntos clave y aprendizajes obtenidos:

Puntos Clave y Aprendizajes:

1. **Estadísticas Descriptivas**:
   - Los valores de la media, mediana y desviación estándar nos proporcionaron una comprensión inicial de la distribución de los datos.
   - Observamos que la skewness (asimetría) y la kurtosis (curtosis) indicaron distribuciones no normales en algunos casos.
2. **Pruebas de Hipótesis**:
   - **Prueba de Jarque-Bera**: La prueba de Jarque-Bera mostró que los errores no siguen una distribución normal.
   - **Prueba ADF**: La prueba de Dickey-Fuller aumentada (ADF) indicó que algunos de nuestros datos tienen raíces unitarias, sugiriendo no estacionariedad en los niveles, pero estacionariedad en la primera diferencia.
   - **Prueba BG-LM**: La prueba de Breusch-Godfrey reveló autocorrelación en los residuos del modelo, indicando que algunos patrones en los datos no fueron capturados por el modelo.
   - **Estadístico de Durbin-Watson**: El estadístico DW mostró una fuerte autocorrelación positiva en los residuos.
3. **Modelo de Regresión**:
   - El modelo de regresión mostró un alto valor de R-cuadrado (0.960059), lo que indica que el 96% de la variación en los pasivos totales se explica por los pasivos corrientes y no corrientes.
   - Sin embargo, la prueba BG-LM y el estadístico de Durbin-Watson sugieren problemas de autocorrelación que podrían afectar la validez del modelo.

Prueba	Valor Estadístico	Valor P
Jarque-Bera	1.246150	0.536293
Correlograma	20.797	0.000

Prueba	Valor Estadístico	Valor P
ADF	-1.810490	0.3676
BG-LM	42.60020	0.0000
Durbin-Watson	0.380024	-

Limitaciones del Modelo:

La no normalidad en los errores y la presencia de autocorrelación sugieren que las suposiciones del modelo de regresión podrían no sostenerse completamente. Esto no significa que el modelo esté fundamentalmente defectuoso, pero indica que puede tener limitaciones en la explicación de la variación en los datos.

Problemas Potenciales:

Violaciones de suposiciones como la normalidad, independencia y homocedasticidad pueden llevar a estimaciones sesgadas, inferencias no confiables o ineficiencia en la predicción de resultados. La autocorrelación, en particular, indica que el modelo no considera algunos patrones subyacentes en los datos.

Espacio para Mejorar:

Para mejorar el modelo, podríamos: - Revaluar las variables utilizadas. - Usar técnicas diferentes o considerar transformaciones de los datos. - Explorar modelos alternativos, como modelos de series temporales que podrían capturar mejor la dependencia temporal en los datos.

El Contexto Importa:

La calidad de un modelo depende del contexto en el que se utiliza, el propósito del modelo y la medida en que sus predicciones son precisas y confiables para las necesidades específicas.

En resumen, este análisis ha proporcionado una visión profunda de la relación entre los pasivos corrientes, no corrientes y totales, así como de las limitaciones y áreas de mejora para futuros análisis y modelados. Hemos aprendido la importancia de las pruebas de diagnóstico para identificar y abordar posibles problemas en los modelos estadísticos.

Recuerda, ningún modelo es perfecto, e identificar limitaciones a través de pruebas de diagnóstico es un paso crucial para refinar y mejorar tu enfoque de modelado. Es una oportunidad para explorar más a fondo, ajustar el modelo y potencialmente descubrir conocimientos adicionales sobre el proceso de generación de datos.