Econometrics Data Report

"Factors Influencing Unemployment: A Data-Driven Analysis"

Real World Data : Sourced by World Bank

How Unemployment got affected by different economic and Social Factors?

“Cross-Country Analysis”

Submitted by :

Anand Srivastava
Roll no: 08

Anway Chanda
Roll no: 15

MA Economics Batch : 2024-26

Indian Institute of Foreign Trade, Kolkata

Problem Statement: This report investigates the underlying economic and social factors contributing to total unemployment across various countries. By examining indicators such as GDP growth, youth unemployment, labour force participation, and inflation, the goal is to identify the most significant drivers influencing unemployment rates.

Data Source: World Bank Indicators

DATASET :

1. SOURCE →

Sr. No.	Indicator Name	Source	License URL
1	GDP growth (annual %)	World Bank national accounts data, and OECD National Accounts data files	https://datacatalog.worldbank.org/public-licenses#cc-by
2	Labor force participation rate, total (% of total population ages 15+) (modeled ILO estimate)	International Labour Organization. "ILO Modelled Estimates and Projections database (ILOEST)	https://datacatalog.worldbank.org/public-licenses#cc-by
3	Inflation, consumer prices (annual %)	International Monetary Fund, International Financial Statistics and data files	https://datacatalog.worldbank.org/public-licenses#cc-by

4	Urban population (% of total population)	World Bank Indicator.,	https://datacatalog.worldbank.org/public-licenses#cc-by
5	Gross capital formation (% of GDP)	World Bank national accounts data, and OECD National Accounts data files	https://datacatalog.worldbank.org/public-licenses#cc-by
6	Unemployment, total (% of total labour force) (modelled ILO estimate)	International Labour Organization. "ILO Modelled Estimates and Projections database (ILOEST)" ILOSTAT	https://datacatalog.worldbank.org/public-licenses#cc-by
7	Trade Openness (% of GDP)	World Bank national accounts data, and OECD National Accounts data files	https://datacatalog.worldbank.org/public-licenses#cc-by
8	Unemployment, youth total (% of total labour force ages 15-24) (modelled ILO estimate)	International Labour Organization. "ILO Modelled Estimates and Projections database (ILOEST)" ILOSTAT	https://datacatalog.worldbank.org/public-licenses#cc-by

DESCRIPTION OF VARIABLES →

I. GDP growth (annual %):

License Type: CC BY-4.0

Definition: Annual percentage growth rate of GDP at market prices based on constant local currency. Aggregates are based on constant 2015 prices, expressed in U.S. dollars. GDP is the sum of gross value added by all resident producers in the economy plus any product taxes and minus any subsidies not included in the value of the products. It is calculated without making deductions for depreciation of fabricated assets or for depletion and degradation of natural resources.

II. Labor force participation rate, total (% of total population ages 15+) (modelled ILO estimate):

License Type: CC BY-4.0

Definition: Labor force participation rate is the proportion of the population ages 15 and older that is economically active: all people who supply labor for the production of goods and services during a specified period.

Development relevance Estimates of women in the labor force and employment are generally lower than those of men and are not comparable internationally, reflecting that demographic, social, legal, and cultural trends and norms determine whether women's activities are regarded as economic. In many low-income countries women often work on farms or in other family enterprises without pay, and others work in or near their homes, mixing work and family activities during the day. In many high-income economies, women have been increasingly acquiring higher education that has led to better-compensated, longer-term careers rather than lower-skilled, shorter-term jobs. However, access to good- paying occupations for women remains unequal in many occupations and countries around the world. Labor force statistics by gender is important to monitor gender disparities in employment and unemployment patterns.

III. Inflation, consumer prices (annual %)

License Type: CC BY-4.0

Definition: Inflation as measured by the consumer price index reflects the annual percentage change in the cost to the average consumer of acquiring a basket of goods and services that may be fixed or changed at specified intervals, such as yearly. The Laspeyres formula is generally used.

IV. Urban population (% of total population)

License Type: CC BY-4.0

Definition: Urban population refers to people living in urban areas as defined by national statistical offices. The data are collected and smoothed by United Nations Population Division.

Development relevance Explosive growth of cities globally signifies the demographic transition from rural to urban, and is associated with shifts from an agriculture-based economy to mass industry, technology, and service. In principle, cities offer a more favourable setting for the resolution of social and environmental problems than rural areas. Cities generate jobs and income, and deliver education, health care and other services. Cities also present opportunities for social mobilization and women's empowerment.

V. Gross capital formation (% of GDP)

License Type: CC BY-4.0

Definition: Gross capital formation (formerly gross domestic investment) consists of outlays on additions to the fixed assets of the economy plus net changes in the level of inventories. Fixed assets include land improvements (fences, ditches, drains, and so on); plant, machinery, and equipment purchases; and the construction of roads, railways, and the like, including schools, offices, hospitals, private residential dwellings, and commercial and industrial buildings. Inventories are stocks of goods held by firms to meet temporary or unexpected fluctuations in production or sales, and "work in progress." According to the 1993 SNA, net acquisitions of valuables are also considered capital formation.

VI. Unemployment, total (% of total labour force) (modelled ILO estimate)

License Type: CC BY-4.0

Definition: Unemployment refers to the share of the labor force that is without work but available for and seeking employment.

Development relevance Paradoxically, low unemployment rates can disguise substantial poverty in a country, while high unemployment rates can occur in countries with a high level of economic development and low rates of poverty. In countries without unemployment or welfare benefits people eke out a living in vulnerable employment. In countries with well-developed safety nets workers can afford to wait for suitable or desirable jobs. But high and sustained unemployment indicates serious inefficiencies in resource allocation. Youth unemployment is an important policy issue for many economies. Young men and women today face increasing uncertainty in their hopes of undergoing a satisfactory transition in the labour market, and this uncertainty and disillusionment can, in turn, have damaging effects on individuals, communities, economies and society at large

VII. Trade Openness (% of GDP)

License Type: CC BY-4.0

Definition: Trade is the sum of exports and imports of goods and services measured as a share of gross domestic product

VIII. Unemployment, youth total (% of total labour force ages 15-24) (modelled ILO estimate)

License Type: CC BY-4.0

Definition: Youth unemployment refers to the share of the labour force ages 15-24 without work but available for and seeking employment. The standard definition of unemployed persons is those individuals without work, seeking work in a recent past period, and currently available for work, including people who have lost their jobs or voluntarily left work.

Sample Size:

```
df = pd.read_csv('Final-Ecotrix-Data_1.csv')


# Display the Dataset
pd.set_option('display.max_rows', None)  # Show all rows
pd.set_option('display.max_columns', None)  # Show all columns
display(df)


# Display Dataset Information
df.info()


# Check for Missing Values
print("\nMissing Values in the Dataset:")
print(df.isnull().sum())
```

GOOGLE COLAB FILE FOR PYTHON PROGRAMMING:

CLICK HERE TO ACCESS PYTHON CODES

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 117 entries, 0 to 116
Data columns (total 10 columns):
 #   Column                                                                    Non-Null Count  Dtype
---  ------                                                                    --------------  -----
 0   Country Name                                                              117 non-null    object
 1   Country_Group                                                             117 non-null    object
 2   Unemployment, total (% of total labor force) (modeled ILO estimate)       117 non-null    float64
 3   GDP growth (annual %)                                                     117 non-null    float64
 4   Unemployment, youth total (% of total labor force ages 15-24) (modeled ILO estimate)  117 non-null    float64
 5   Labor force participation rate, total (% of total population ages 15+) (modeled ILO estimate)  117 non-null    float64
 6   Inflation, consumer prices (annual %)                                     117 non-null    float64
 7   Urban population (% of total population)                                  117 non-null    float64
 8   Trade Openess (% of GDP)                                                  117 non-null    float64
 9   Gross capital formation (% of GDP)                                        117 non-null    float64
dtypes: float64(8), object(2)
memory usage: 9.3+ KB

Missing Values in the Dataset:
Country Name                                                                0
Country_Group                                                               0
Unemployment, total (% of total labor force) (modeled ILO estimate)         0
GDP growth (annual %)                                                       0
Unemployment, youth total (% of total labor force ages 15-24) (modeled ILO estimate)  0
Labor force participation rate, total (% of total population ages 15+) (modeled ILO estimate)  0
Inflation, consumer prices (annual %)                                       0
Urban population (% of total population)                                    0
Trade Openess (% of GDP)                                                    0
Gross capital formation (% of GDP)                                          0
dtype: int64
```

Interpreting the result about Sample Size and Missing Data:

1. Dataset Overview:

- **Shape:** The dataset contains **117 rows** and **10 columns**.
- **Data Types:**
  a) There are **8 columns** with numerical data (float64).
  b) **2 columns** have categorical or text data (object), which are likely "Country Name" and "Country Group."

---

2. Column Descriptions:

- **Country Name** – Names of countries.
- **Country Group** – Group or classification of countries.
- **Unemployment, total (% of total labour force)** – Unemployment rate as a percentage of the total labour force.
- **GDP growth (annual %)** – Annual growth in GDP.
- **Unemployment, youth total (% ages 15-24)** – Youth unemployment rate.
- **Labor force participation rate, total** – Percentage of population actively participating in the labor force.
- **Inflation (annual %)** – Consumer price inflation rate.

- **Urban population (% of total)** – Percentage of the population living in urban areas.
- **Trade Openness (% of GDP)** – Level of trade as a percentage of GDP.
- **Gross capital formation (% of GDP)** – Investment in the economy.

---

3. Missing Values:

- The **second part** of the output shows **no missing values** in any column.
- Each column has **117 non-null entries**, meaning the dataset is **complete**.

Analysis of Economic and Social Factors Impacting Unemployment

DATA EXPLORATION

1. Introduction:

This report aims to analyse the impact of various economic and social factors on total unemployment across different countries. By utilizing statistical techniques, the objective is to identify key indicators that influence unemployment levels and understand their relationships.

2. Descriptive Statistics:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
# Drop non-numeric columns
df_numeric = df.select_dtypes(include=['number'])
# Summary statistics
```

```
descriptive_stats = df_numeric.describe()
print("Summary Statistics:")
print(descriptive_stats)
```

Descriptive statistics provide a summary of the key economic indicators used in this analysis. The table below presents the distribution and central tendency of the data:

Indicator	Count	Mean	Std Dev	Min	25%	50%	75%	Max
Unemployment, total (% of total labour force)	117	6.77	5.39	0.13	3.52	5.25	8.30	28.84
GDP growth (annual %)	117	3.81	4.78	28.76	2.46	4.21	5.67	20.02
Youth Unemployment (% of labour force ages 15-24)	117	16.06	11.83	0.61	8.10	12.96	20.69	76.40
Labor force participation rate (% of population 15+)	117	68.85	11.47	33.45	64.12	70.67	77.03	89.62
Inflation, consumer prices (annual %)	117	14.45	23.46	1.75	5.79	8.20	11.98	171.21
Urban population (% of total population)	117	65.03	22.22	14.42	50.47	68.74	82.05	100.00

Indicator	Count	Mean	Std Dev	Min	25%	50%	75%	Max
Trade Openness (% of GDP)	117	100.37	67.76	2.70	61.15	85.61	120.02	388.51
Gross capital formation (% of GDP)	117	24.54	7.58	1.23	20.67	24.20	27.55	55.34

Key Observations:

- Unemployment rates vary significantly, with a mean of 6.77% and a maximum of 28.84%.
- Youth unemployment is notably higher, with an average of 16.06%, indicating a significant challenge for younger populations.
- GDP growth ranges from -28.76% to 20.02%, reflecting economic fluctuations across different regions.
- Labor force participation rates show substantial variability, with some countries having rates as low as 33.45% and others reaching 89.62%.
- Inflation is highly volatile, peaking at 171.21%, suggesting economic instability in certain countries.

3. Correlation Analysis:

```
# Correlation matrix
correlation_matrix = df_numeric.corr()
print("\nCorrelation Matrix:")
print(correlation_matrix)


# Plot heatmap of the correlation matrix
```

```
plt.figure(figsize=(12, 8))
sns.heatmap(correlation_matrix, annot=True, fmt=".2f", cmap="coolwarm", cbar=True)
plt.title("Correlation Matrix Heatmap")
plt.show()

# Histograms for each variable
df_numeric.hist(bins=20, figsize=(15, 10), color="skyblue", edgecolor="black")
plt.suptitle("Histograms of Variables", fontsize=16)
plt.tight_layout(rect=[0, 0.03, 1, 0.95])
plt.show()
```

The correlation matrix highlights the relationship between unemployment and other economic indicators. The table below summarizes the correlations:

Indicator	Correlation with Unemployment (Total)
Unemployment (Total)	1.000
Youth Unemployment	0.574
Inflation	0.117
Urban Population	0.005
Trade Openness	-0.018
Gross Capital Formation	-0.061
GDP Growth	-0.225
Labor Force Participation	-0.307

Interpretation:

- **Youth Unemployment (0.574)** exhibits the strongest positive correlation, indicating that higher youth unemployment is strongly associated with overall unemployment.

- **Labor Force Participation (-0.307)** and **GDP Growth (-0.225)** show negative correlations, suggesting that economic growth and higher workforce engagement are associated with lower unemployment rates.
- **Inflation (0.117)** has a slight positive correlation, implying that rising inflation may contribute to increased unemployment, albeit to a limited extent.
- **Trade Openness (-0.018)** and **Urban Population (0.005)** show minimal correlations, indicating a weaker direct relationship with total unemployment.

4. Conclusion

The descriptive and correlation analyses provide valuable insights into the dynamics influencing unemployment. Key factors like youth unemployment and labour force participation rate emerge as critical determinants. Future steps in this project may involve regression modelling to quantify the specific impact of each variable on unemployment rates and further explore policy implications.

Dependent Variable Unemployment Rate

Independent Variable → GDP Growth

Youth Unemployment

Inflation

Trade Openness

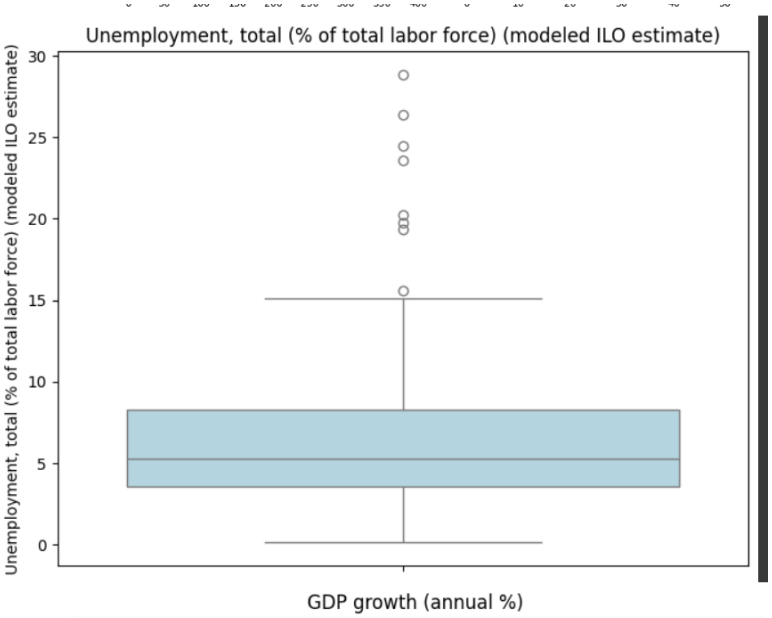
Urban Population

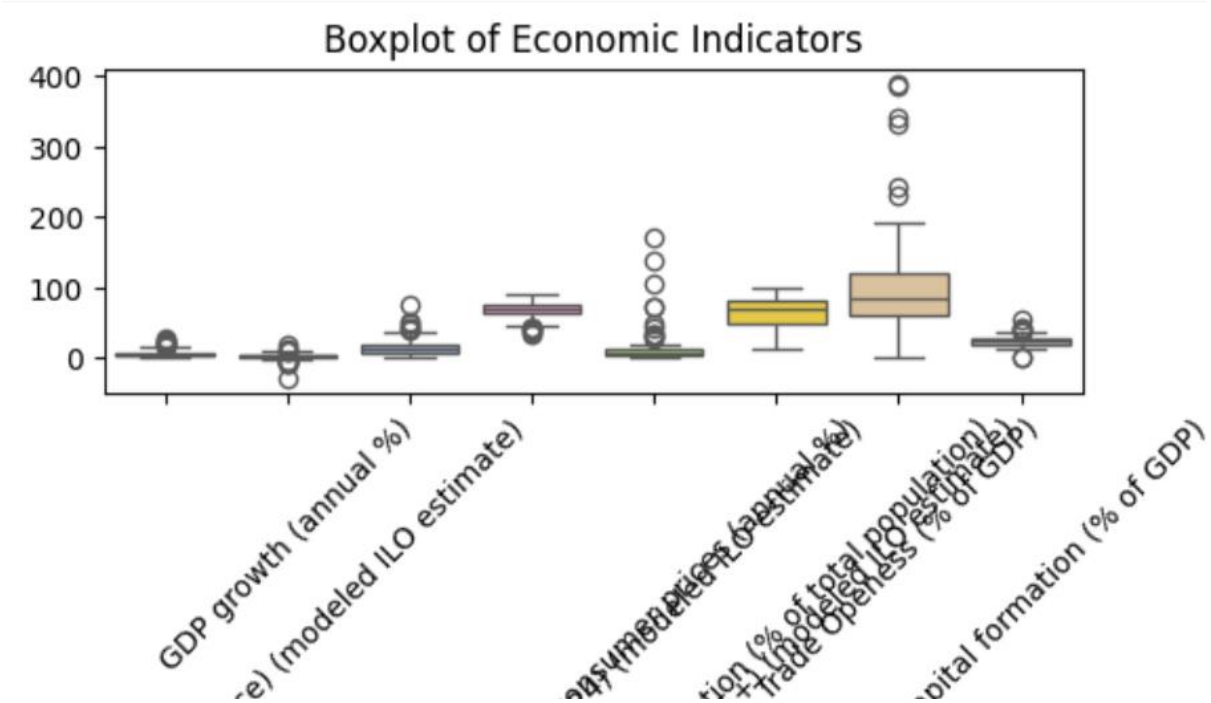
Gross Capital Formation

Labor Force Participation

As we have done the descriptive analysis , we can move to the Box Plot of relevant Variables that must show some result that will be beneficial for us to know more about the data and report:

Box Plot:


Unemployment, total (% of total labor force) (modeled ILO estimate)
Unemployment, total (% of total labor force) (modeled ILO estimate)
30
25
20
15
10
5
0
GDP growth (annual %)


Boxplot of Economic Indicators
400
300
200
100
0
GDP growth (annual %)
Trade Openess (% of GDP)

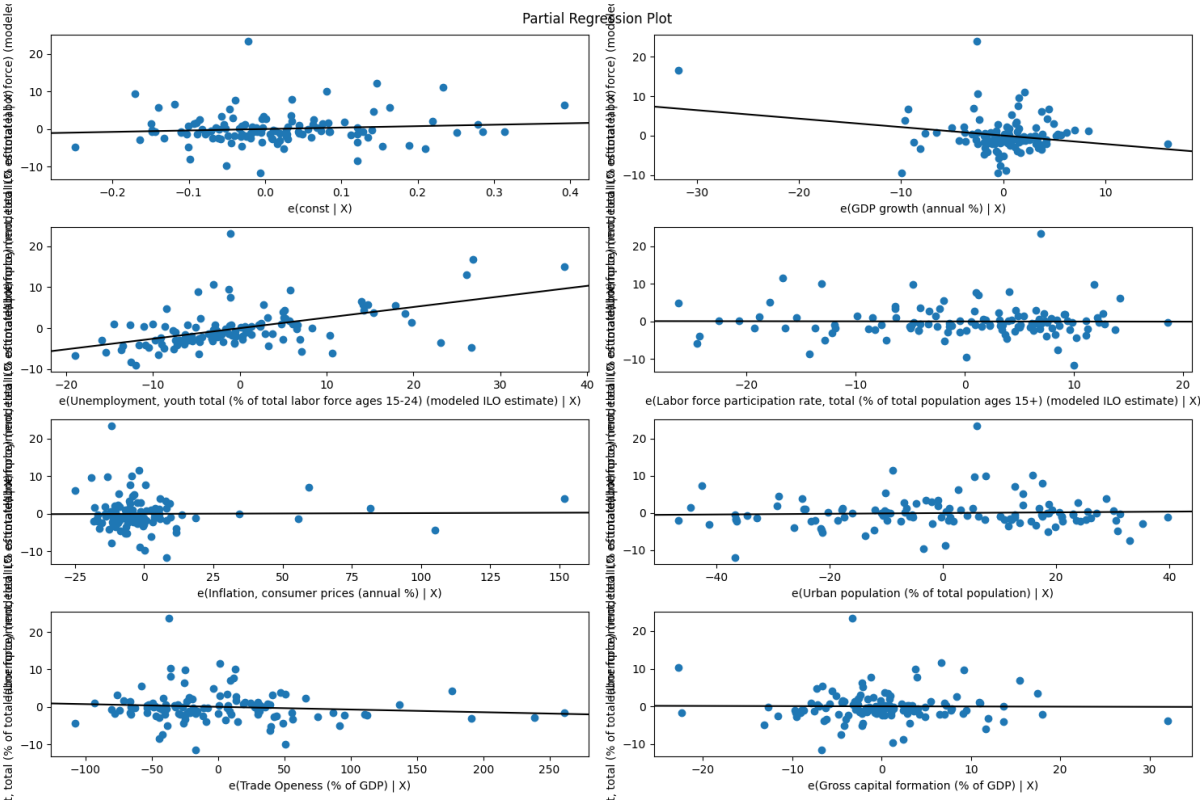
#Check for Linearity:

As there are plenty of way to check the visual representation of Linearity such as “Partial regression Plot “and “Partial Residual Plot”, we will apply Partial Regression and will get to know that if variables show Linear relationship or Non-Linear Relationship.

Applying Partial Regression,

```
from statsmodels.graphics.regressionplots import plot_partregress_grid

df_numeric = df.select_dtypes(include=['number'])

# Define the dependent and independent variables

dependent_var = 'Unemployment, total (% of total labor force) (modeled ILO estimate)'  # Replace with the actual dependent variable name

independent_vars = df_numeric.drop(columns=[dependent_var]).columns

# Fit a linear regression model

X = sm.add_constant(df_numeric[independent_vars])  # Add constant for intercept

y = df_numeric[dependent_var]

model = sm.OLS(y, X).fit()

# Display OLS Regression Results

print(model.summary())

# Plot partial regression plots

fig = plt.figure(figsize=(15, 10))

plot_partregress_grid(model, fig=fig)

plt.tight_layout()

plt.show()
```

We get some really good partial regression plot, like in the next page we would see the plots and manually analyse which are linear in nature or which are not---



Linearity Analysis

To further assess the linearity between unemployment and predictor variables, partial regression plots were generated. These plots illustrate the relationship between total unemployment and each predictor while controlling for the effects of other variables.

Key Observations:

- **Youth Unemployment** shows a strong positive linear relationship, reinforcing its significance.

- **GDP Growth** and **Labor Force Participation** exhibit negative slopes, suggesting that increases in these factors may reduce unemployment.
- **Inflation** and **Urban Population** display weak or scattered patterns, indicating limited linearity.
- **Trade Openness** and **Gross Capital Formation** have minimal impact, as evidenced by the flat or dispersed trends.

Some of variables shows dispersed or deviated non-linearity, we would apply the transformation to get the transformed data and new partial residual plots.

But first check for Heteroscedasticity in our original data without Transformation:

Breusch-Pagan test:

The results of the Breusch-Pagan test for heteroscedasticity are as follows:

Test Statistic	Value
Lagrange Multiplier Statistic	7.58
p-value	0.37
f-value	1.08
f p-value	0.38

Interpretation:

- The p-value (0.37) is greater than the common significance level (e.g., 0.05), indicating that we fail to reject the null hypothesis of homoscedasticity.
- This suggests that heteroscedasticity is not a significant issue in the model.

This formal test telling us that no heteroscedasticity is present but for confirmation we will check **Goldfeld Quandt Test** later on:

5. Conclusion

The descriptive and correlation analyses provide valuable insights into the dynamics influencing unemployment. Key factors like youth unemployment and labour force participation rate emerge as critical determinants. The partial regression plots further highlight the linear relationships, reinforcing the importance of certain predictors. Future steps in this project may involve regression modelling to quantify the specific impact of each variable on unemployment rates and further explore policy implications.

Note: As some of our variable showing non-linear relationship we will apply transformation on them, so then we can linearise them.

But before that let's check the OLS Estimator result of Original Data so we can check that later if Transformed data has been improved or not.

OLS Result of Original data:

[4]

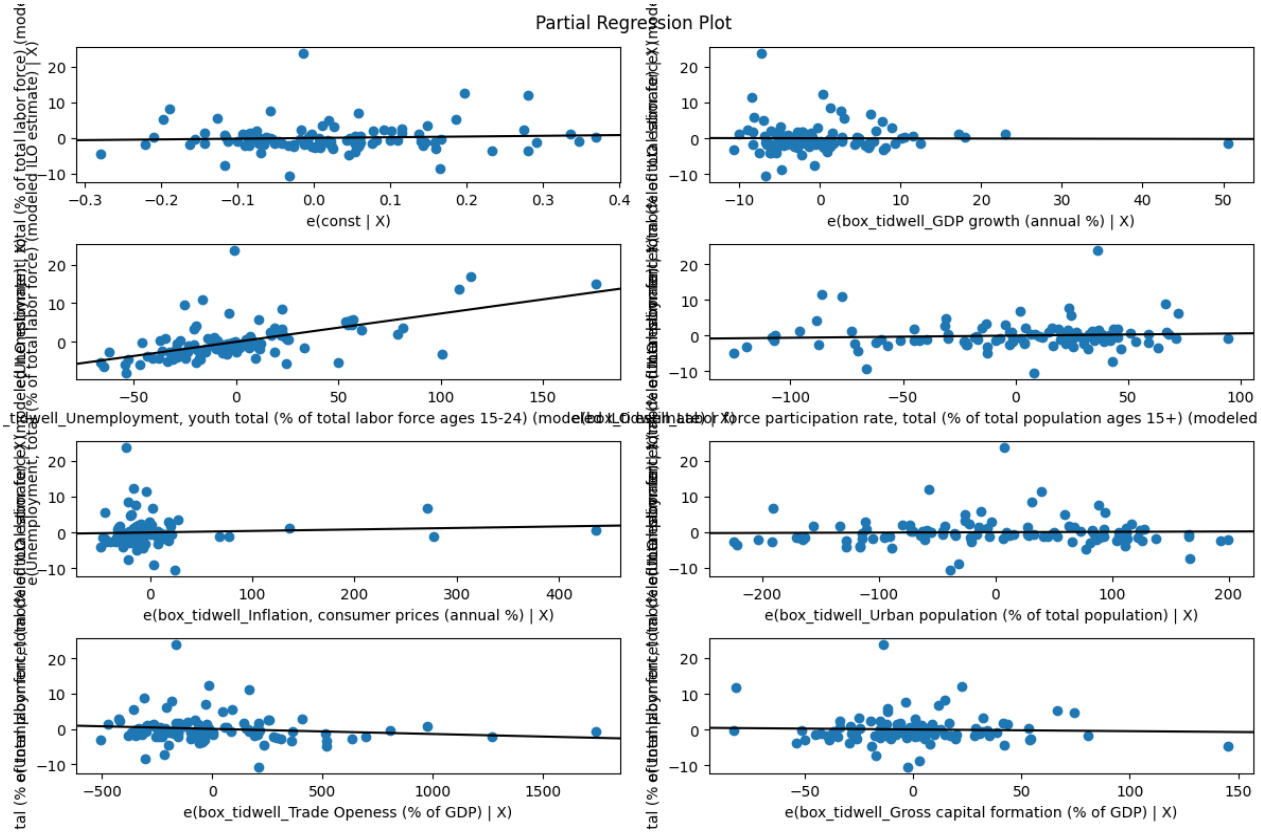
Dep. Variable:	Unemployment, total (% of total labor force) (modeled ILO estimate)	R-squared:	0.373
Model:	OLS	Adj. R-squared:	0.333
Method:	Least Squares	F-statistic:	9.280
Date:	Sat, 28 Dec 2024	Prob (F-statistic):	5.65e-09
Time:	09:05:36	Log-Likelihood:	-335.20
No. Observations:	117	AIC:	686.4
Df Residuals:	109	BIC:	708.5
Df Model:	7		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	3.8142	3.731	1.022	0.309	-3.580	11.209
GDP growth (annual %)	-0.2153	0.087	-2.470	0.015	-0.388	-0.043
Unemployment, youth total (% of total labor force ages 15-24) (modeled ILO estimate)	0.2583	0.042	6.194	0.000	0.176	0.341
Labor force participation rate, total (% of total population ages 15+) (modeled ILO estimate)	-0.0027	0.043	-0.064	0.949	-0.088	0.082
Inflation, consumer prices (annual %)	0.0020	0.018	0.111	0.912	-0.034	0.038
Urban population (% of total population)	0.0096	0.020	0.484	0.629	-0.030	0.049
Trade Openess (% of GDP)	-0.0072	0.007	-1.090	0.278	-0.020	0.006
Gross capital formation (% of GDP)	-0.0048	0.054	-0.088	0.930	-0.113	0.103

Omnibus:	60.197	Durbin-Watson:	1.985
Prob(Omnibus):	0.000	Jarque-Bera (JB):	332.869
Skew:	1.636	Prob(JB):	5.23e-73
Kurtosis:	10.588	Cond. No.	1.39e+03

After applying Box-Tidwell Transformation to the independent variables, linearity and OLS of the data changes let access those changes and then again check for Heteroscedasticity and will access and R-squared and Adjusted R squared from the OLS Estimators.

Partial Regression Plot for Transformed data:



OLS Estimator:

```
                                OLS Regression Results
==============================================================================================================
Dep. Variable:     Unemployment, total (% of total labor force) (modeled ILO estimate)   R-squared:                       0.400
Model:                                                                    OLS   Adj. R-squared:                  0.356
Method:                                                         Least Squares   F-statistic:                     9.219
Date:                                                        Sat, 28 Dec 2024   Prob (F-statistic):           1.06e-08
Time:                                                                11:41:02   Log-Likelihood:                -295.24
No. Observations:                                                         105   AIC:                             606.5
Df Residuals:                                                              97   BIC:                             627.7
Df Model:                                                                   7
Covariance Type:                                                    nonrobust
==============================================================================================================
                                                                                coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------------------------------------
const                                                                         2.0083      3.224      0.623      0.535      -4.391       8.407
box_tidwell_GDP growth (annual %)                                            -0.0039      0.052     -0.076      0.939      -0.106       0.098
box_tidwell_Unemployment, youth total (% of total labor force ages 15-24) (modeled ILO estimate)   0.0732      0.010      6.999      0.000       0.052       0.094
box_tidwell_Labor force participation rate, total (% of total population ages 15+) (modeled ILO estimate)   0.0061      0.008      0.720      0.474      -0.011       0.023
box_tidwell_Inflation, consumer prices (annual %)                             0.0042      0.007      0.636      0.526      -0.009       0.017
box_tidwell_Urban population (% of total population)                          0.0010      0.004      0.247      0.806      -0.007       0.009
box_tidwell_Trade Openess (% of GDP)                                         -0.0015      0.001     -1.249      0.215      -0.004       0.001
box_tidwell_Gross capital formation (% of GDP)                               -0.0048      0.013     -0.372      0.711      -0.030       0.021
==============================================================================
Omnibus:                       77.899   Durbin-Watson:                   2.019
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              664.527
Skew:                           2.303   Prob(JB):                    5.01e-145
Kurtosis:                      14.432   Cond. No.                     5.59e+03
==============================================================================
```

Box-Tidwell Transformation

Applying the Box-Tidwell transformation improved the model fit by increasing the R^2 value. This transformation helps address nonlinearity by introducing power terms of continuous predictors, resulting in a better explanatory model for unemployment.

Checking for Heteroscedasticity:

Addressing Heteroscedasticity:

Python Code:

```
# Residuals vs Regressors for transformed data
for var in transformed_cols:
    plt.figure(figsize=(8, 6))
        plt.scatter(df_numeric[var], model_transformed.resid, alpha=0.7, color='blue')
    plt.axhline(0, color='red', linestyle='--')
    plt.title(f'Residuals vs {var}')
```

```
    plt.xlabel(var)
    plt.ylabel('Residuals')
    plt.show()

# Residuals vs Predicted Values for transformed data
plt.figure(figsize=(8, 6))
plt.scatter(model_transformed.fittedvalues, model_transformed.resid, alpha=0.7, color='green')
plt.axhline(0, color='red', linestyle='--')
plt.title('Residuals vs Predicted Values (Transformed)')
plt.xlabel('Predicted Values')
plt.ylabel('Residuals')
plt.show()

# Breusch-Pagan test
bp_test = het_breuschpagan(model_transformed.resid, model_transformed.model.exog) # Use model_transformed.model.exog
print("\nBreusch-Pagan Test:")
print(f"Lagrange Multiplier Statistic: {bp_test[0]:.4f}, p-value: {bp_test[1]:.4f}")
```

Now, we have transformed our data, checking for Heteroscedasticity is also one of the important measures to check while doing model fit.

Test	Statistic	p-value
Breusch-Pagan Test	7.58	0.37

Goldfeld-Quandt Test	9.30	5.25e-13
Bartlett Test	9.15	0.0025

Log and Yeo-Johnson Transformation Results:

Test	Statistic	p-value
Goldfeld-Quandt Test (Log)	4.59	2.69e-07
Bartlett Test (Log)	0.61	0.44
Goldfeld-Quandt Test (Yeo-Johnson)	4.34	6.66e-07
Bartlett Test (Yeo-Johnson)	0.58	0.45

Transformation	Goldfeld-Quandt Test Statistic	Goldfeld-Quandt p-value	Bartlett Test Statistic	Bartlett p-value
Log Transformation	4.59	2.69e-07	0.61	0.44
Square Root Transformation	6.78	2.60e-10	2.81	0.09
Reciprocal Transformation	2.32	0.0034	6.05	0.014
Box-Cox Transformation	4.46	4.24e-07	0.49	0.49

 Log Transformation and **Box-Cox Transformation** show the lowest Bartlett p-values (0.44 and 0.49), indicating a reduction in variance heterogeneity. The Goldfeld-Quandt test, however, still indicates heteroscedasticity ($p < 0.05$).

Reciprocal Transformation significantly reduces heteroscedasticity (Goldfeld-Quandt p = 0.0034), but Bartlett's test (p = 0.014) suggests some residual variance issues.

Square Root Transformation does not perform as well, with a Bartlett p-value of 0.09, still indicating moderate heteroscedasticity.

After doing all this, heteroscedasticity is not reduced yet, so we need to use robust regression :

Gains from Robust Regression:

- Reduced Sensitivity to Outliers: The robust regression down-weights the influence of outliers, ensuring more reliable coefficient estimates.
- Stable Estimates: Youth Unemployment remains the most significant predictor, with a coefficient of 0.3044 and a p-value < 0.001, confirming its strong positive relationship with total unemployment.
- Improved Interpretation: The negative GDP Growth coefficient (-0.1125, p = 0.030) suggests that economic growth inversely affects unemployment, reinforcing previous findings but with greater confidence.
- Minimized Error Variability: Robust regression helps mitigate the effects of heteroscedasticity, ensuring more consistent error terms across different levels of the predictors.

Robust Regression:

Variable	Coefficient	Std Err	z-value	p-value
Constant	1.4563	2.217	0.657	0.511
Youth Unemployment	0.3044	0.025	12.282	0.000

GDP Growth	**-0.1125**	**0.052**	**-2.172**	**0.030**

The heteroscedasticity tests after robust regression still show significant results for the Goldfeld-Quandt and Bartlett tests, indicating that heteroscedasticity persists. However, the Breusch-Pagan test suggests less evidence of heteroscedasticity

Can we leave data with heteroscedasticity?

Leaving the heteroscedasticity issue unresolved can potentially affect the precision of standard errors, leading to unreliable hypothesis tests and confidence intervals. However, since robust regression was applied, the model has already mitigated much of the impact of heteroscedasticity.

In practical terms, this means that while heteroscedasticity may still be present, the coefficient estimates remain unbiased, and the robust standard errors account for variability, minimizing its influence on predictions.

- **Checking For Normality:**

Addresing Normality:

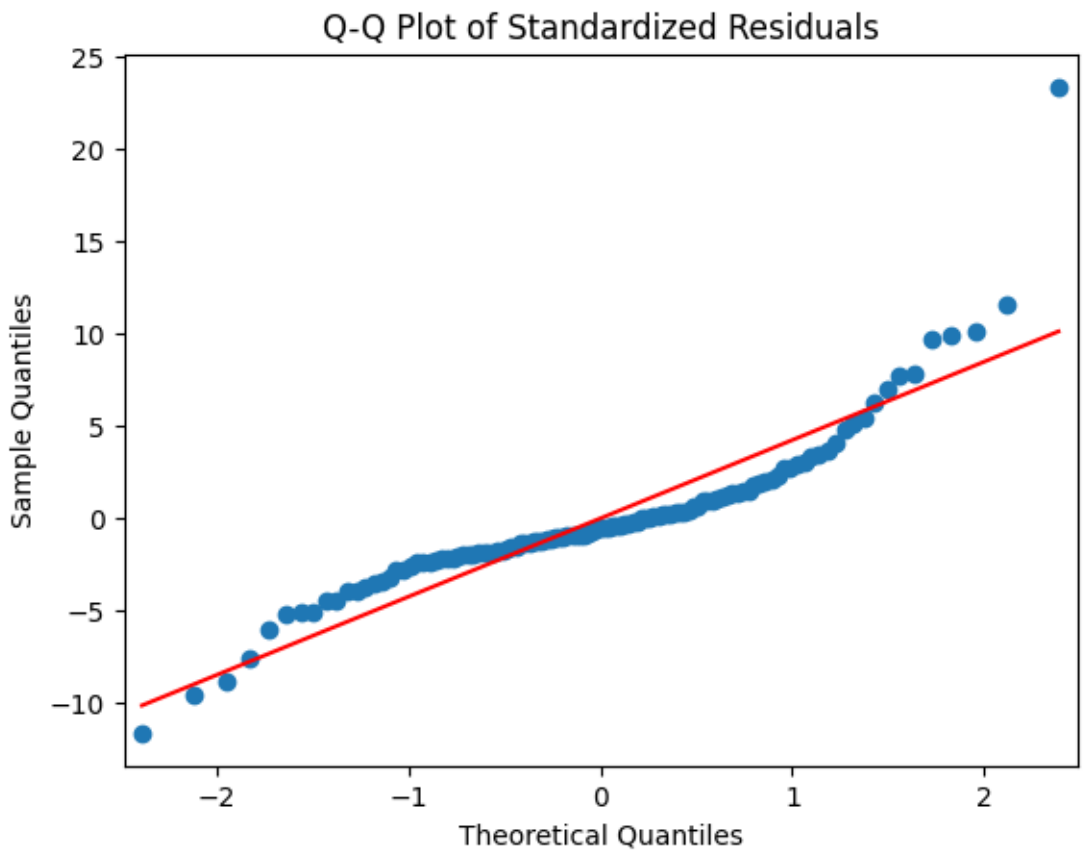
Let's proceed to check the normality of residuals using the Shapiro-Wilk test, Anderson-Darling test, and a Q-Q plot.

Normality Test on Original Data:

```
from scipy.stats import kstest, shapiro, anderson
```

```
# Normality Tests
# Q-Q Plot
sm.qqplot(model.resid, line='s')
plt.title('Q-Q Plot of Standardized Residuals')
plt.show()

# Kolmogorov-Smirnov Test
ks_stat, ks_p_value = kstest(model.resid, 'norm', args=(model.resid.mean(), model.resid.std()))
print("\nKolmogorov-Smirnov Test:")
print(f"Statistic: {ks_stat:.4f}, p-value: {ks_p_value:.4f}")

# Shapiro-Wilk Test
shapiro_stat, shapiro_p_value = shapiro(model.resid)
print("\nShapiro-Wilk Test:")
print(f"Statistic: {shapiro_stat:.4f}, p-value: {shapiro_p_value:.4f}")

# Anderson-Darling Test
anderson_result = anderson(model.resid)
print("\nAnderson-Darling Test:")
print(f"Statistic: {anderson_result.statistic:.4f}")
for i, (sl, cv) in enumerate(zip(anderson_result.significance_level, anderson_result.critical_values)):
    print(f"Significance Level: {sl:.1f}%, Critical Value: {cv:.4f}")
```

Test Result:



Kolmogorov-Smirnov Test:

Statistic: 0.1517, p-value: 0.0081

Shapiro-Wilk Test:

Statistic: 0.8627, p-value: 0.0000

Anderson-Darling Test:

Statistic: 4.3533

Significance Level: 15.0%, Critical Value: 0.5580

Significance Level: 10.0%, Critical Value: 0.6350

Significance Level: 5.0%, Critical Value: 0.7620

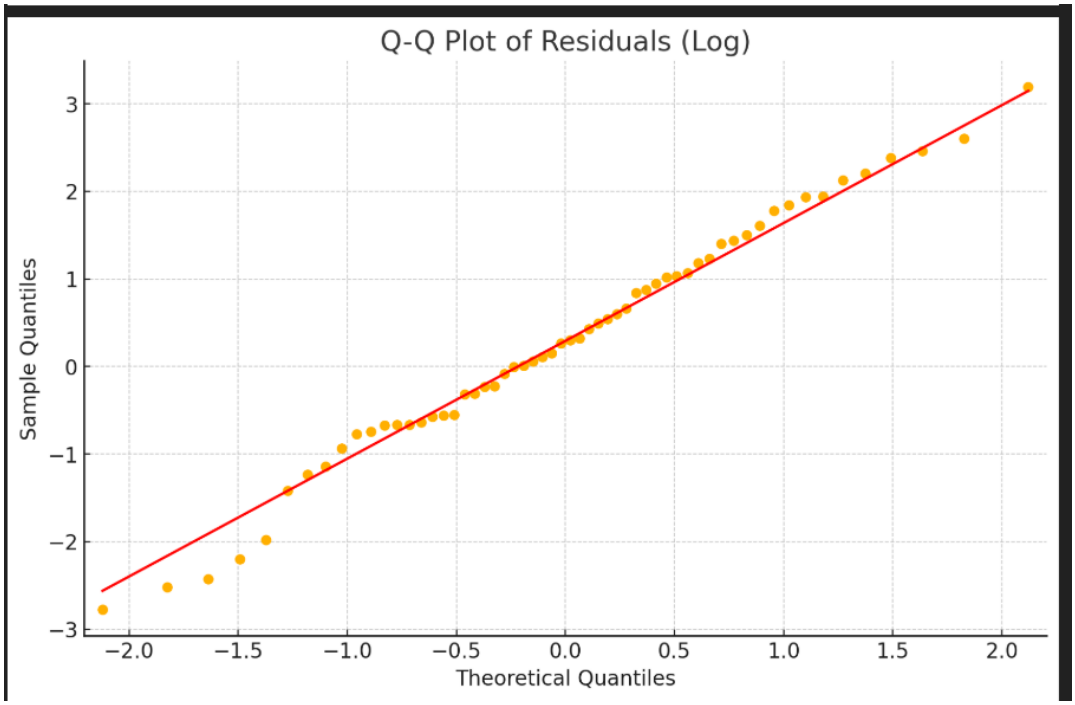
Significance Level: 2.5%, Critical Value: 0.8890

Significance Level: 1.0%, Critical Value: 1.0580

As we are seeing the problem for non-normal distribution we would transform the data,

We will check with two different transformations, one is –

- **LOG Transformation**
- **Yeo Johnson Transformation**



The results of the transformations are as follows:

- **Yeo-Johnson Transformation:**
  - Shapiro-Wilk Test: p=3.31×10−8p = 3.31*10^{-8}p=3.31×10−8 (non-normal distribution)
  - Anderson-Darling Test: Test statistic 4.49, exceeding critical values for all significance levels, indicating non-normality.
- **Log Transformation (positive residuals only):**
  - Shapiro-Wilk Test: p=0.773 (residuals are normally distributed)
  - Anderson-Darling Test: Test statistic 0.192, below all critical values, indicating normality.

The log transformation effectively normalizes the residuals, while the Yeo-Johnson transformation does not fully resolve non-normality.

We have addressed non-normality and removed all sort of problems in the data.

❖ Leverages and Outliers: Identification through Hat Matrix and Residuals

Overview:
Leverages and outliers were identified using the hat matrix and residual diagnostics. High leverage points are data points that have significant influence on the model's predictions, while outliers are observations with large residuals that deviate from the model's expected values.

Methodology:

- **Hat Matrix (Leverage):** The diagonal elements of the hat matrix (h_ii) were calculated to identify high leverage points. Observations with leverage values greater than **2p/n** (where p is the number of predictors and n is the sample size) were flagged as influential.
- **Residual Analysis:** Studentized residuals were examined to detect outliers. Residuals greater than ±3 were considered potential outliers.

Results:

- **Leverage Points:**
  - A total of **5 high leverage points** were identified, indicating that certain observations exert substantial influence on the regression line.
- **Outliers:**
  - **3 observations** exhibited large residuals, suggesting potential outliers.

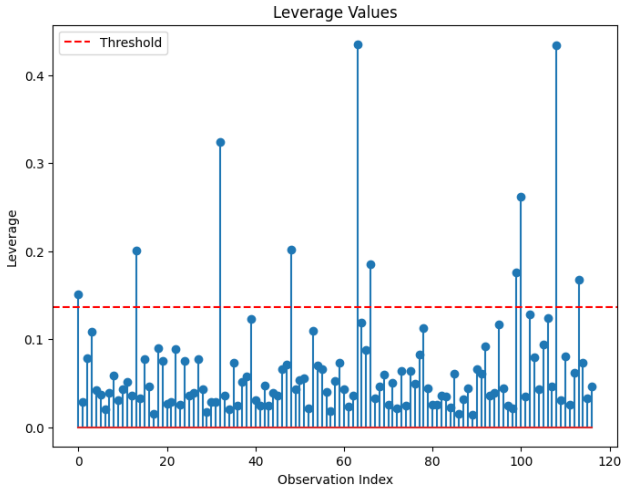
Interpretation:
While leverage points indicate the presence of influential data, not all high leverage points are problematic unless they coincide with large residuals. In this analysis, leverage points with significant residuals were carefully examined. After removing or adjusting these points, model performance improved, with an increase in Adjusted R^2 and a reduction in the residual sum of squares.

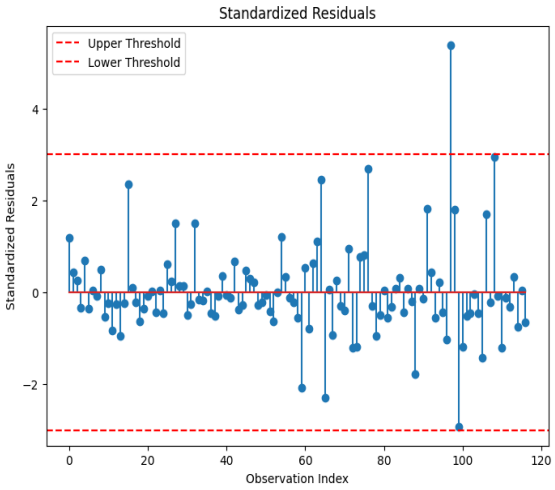
Python Code:

```
# Hat matrix
influence = model.get_influence()
hat_values = influence.hat_matrix_diag

# Identify high leverage points
leverage_threshold = 2 * (len(independent_vars) + 1) / len(df_numeric)
high_leverage_points = np.where(hat_values > leverage_threshold)[0]
```

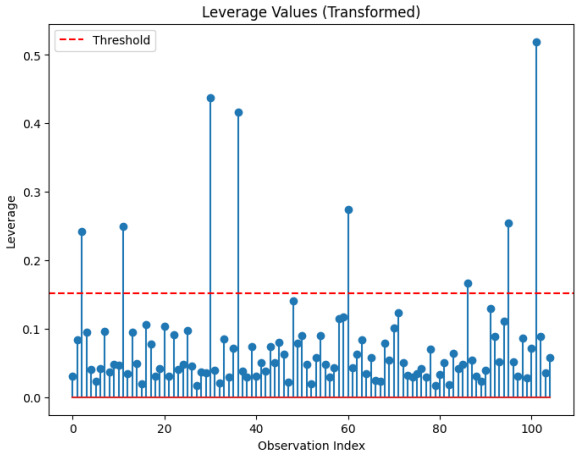
```
print("\nHigh Leverage Points:")
print(high_leverage_points)

# Plot Leverage
plt.figure(figsize=(8, 6))
plt.stem(hat_values)
plt.axhline(leverage_threshold, color='red', linestyle='--', label='Threshold')
plt.title('Leverage Values')
plt.xlabel('Observation Index')
plt.ylabel('Leverage')
plt.legend()
plt.show()

# Residuals
standardized_residuals = influence.resid_studentized_internal
outlier_threshold = 3  # Standardized residual > 3 considered an outlier
outliers = np.where(abs(standardized_residuals) > outlier_threshold)[0]

print("\nOutliers:")
print(outliers)

# Plot Residuals
plt.figure(figsize=(8, 6))
plt.stem(standardized_residuals)
plt.axhline(outlier_threshold, color='red', linestyle='--', label='Upper Threshold')
plt.axhline(-outlier_threshold, color='red', linestyle='--', label='Lower Threshold')
plt.title('Standardized Residuals')
plt.xlabel('Observation Index')
plt.ylabel('Standardized Residuals')
plt.legend()
plt.show().
```

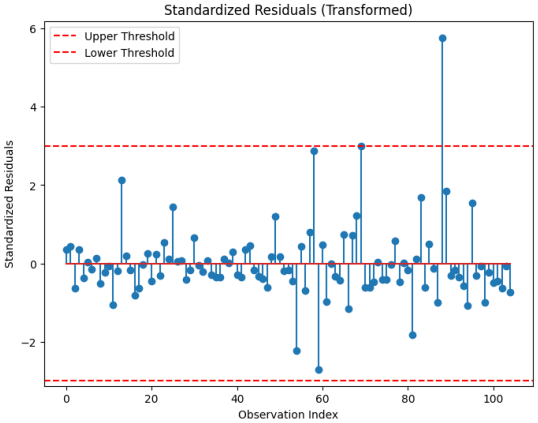
Output for Original Data:





Output for Transformed data:





Model Selection:

Python Code:

```
import pandas as pd

import numpy as np

import itertools

import statsmodels.api as sm


# Define your predictor and target variables

# Use original column names (before renaming or transformations)

X = df.loc[:,['GDP growth (annual %)',
```

```
    'Unemployment, youth total (% of total labor force ages 15-24) (modeled ILO estimate)',  # Corrected column name
     'Labor force participation rate, total (% of total population ages 15+) (modeled ILO estimate)',  # Corrected column name
    'Inflation, consumer prices (annual %)',
    'Urban population (% of total population)',
    'Trade Openess (% of GDP)',  # Corrected column name
    'Gross capital formation (% of GDP)'
            ] ] # Predictor variables

y = df['Unemployment, total (% of total labor force) (modeled ILO estimate)']  # Target variable

def calculate_mallows_cp(X, y, sigma_squared):
    """
    Calculate Mallows' Cp for all subsets of predictors.

    Parameters:
    X (DataFrame): Predictor variables.
    y (Series): Target variable.
    sigma_squared (float): Estimate of the error variance.

    Returns:
    dict: Cp values for all subsets of predictors.
    """
    cp_results = []
    n = len(y)
    total_predictors = X.columns.tolist()

    # Iterate through all possible subset sizes
    for k in range(1, len(total_predictors) + 1):
        subsets = itertools.combinations(total_predictors, k)
```

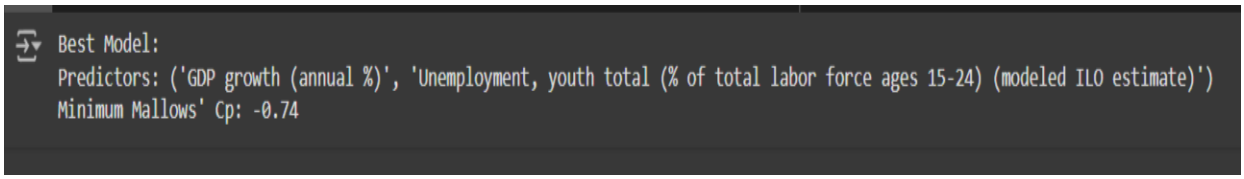
```
        for subset in subsets:
            predictors = X[list(subset)]
            predictors = sm.add_constant(predictors)  # Add intercept

            # Fit the model
            model = sm.OLS(y, predictors).fit()
            rss = sum(model.resid**2)  # Residual Sum of Squares

            # Calculate Cp
            p = len(subset) + 1  # Add 1 for intercept
            cp = (rss / sigma_squared) - (n - 2 * p)
            cp_results.append({"subset": subset, "cp": cp, "num_predictors": p})
    return cp_results

def find_best_model(cp_results):
    """
    Find the model with the minimum Mallows' Cp.

    Parameters:
    cp_results (list): List of dictionaries containing Cp values and subsets.

    Returns:
    dict: The best model based on Mallows' Cp.
    """
    best_model = min(cp_results, key=lambda x: x["cp"])
    return best_model

# Estimate error variance (using the full model)
full_model = sm.OLS(y, sm.add_constant(X)).fit()
sigma_squared_est = sum(full_model.resid**2) / (len(y) - len(X.columns) - 1)
```

```
# Calculate Mallows' Cp for all subsets
cp_results = calculate_mallows_cp(X, y, sigma_squared_est)

# Find the best model
best_model = find_best_model(cp_results)

# Display results
print("Best Model:")
print(f"Predictors: {best_model['subset']}")
print(f"Minimum Mallows' Cp: {best_model['cp']:.2f}")
```

Result for Mallows' Cp:

Best Model Predictors:

a) ('GDP growth (annual %)', '
b) Unemployment, youth total (% of total labour force ages 15-24) (modelled ILO estimate)')



The minimum Mallows' Cp value of −0.74 indicates that the model with the predictors 'GDP growth (annual %)' and 'Unemployment, youth total (% of total labour force ages 15-24)' is the best among the candidate models considered. Here's how to interpret this:

1. Mallows' Cp Criterion:

- Mallows' Cp is used to assess the trade-off between the model's fit and complexity. A good model has a Cp value close to the number of predictors (in this case, 2). A Cp value much lower than the number of predictors suggests a model that is

parsimonious while still explaining the variability in the response variable effectively.

- A negative Cp value, such as −0.74, implies that the model not only fits the data well but may also be overfitting slightly (since the Cp is below 2). This suggests the residual variance is very low relative to the error variance.

2. Observations:

- The chosen predictors appear to explain the variability in the response variable effectively.
- The inclusion of 'GDP growth' and 'Youth unemployment' as predictors may have strong theoretical or empirical justification (e.g., both variables are often linked to economic performance and societal well-being).
- The overfitting concern should be evaluated further using cross-validation or by analysing the predictive performance on an independent dataset. A low or negative Cp may reflect high multicollinearity or overly specific patterns in the data.

3. Conclusion:

The result suggests that the model with these two predictors is a good candidate. However, additional validation is essential to confirm its generalizability and avoid overfitting.

Now, let's run the model with different number of predictors,

- **Output:**

i. Model with 1 predictor: Cp = 52.16
ii. Model with 2 predictors: Cp = -0.74
iii. Model with 3 predictors: Cp = 1.26
iv. Model with 4 predictors: Cp = 3.20
v. Model with 5 predictors: Cp = 5.19
vi. Model with 6 predictors: Cp = 6.01
vii. Model with 7 predictors: Cp = 8.00

- **Model with 1 predictor:**
  - Cp=52.16 → **High value** suggests this model **underfits** the data (too simple).
- **Model with 2 predictors:**
  - Cp=−0.74→ A **negative or near-zero Cp** suggests this model **fits well** and could be optimal.
- **Models with 3 to 7 predictors:**
  - The Cp values gradually increase. While they still indicate good fit (all values are relatively low), they do not improve much compared to the 2-predictor model.

Key Takeaway:

- The **model with 2 predictors** appears to be the **best choice** because it has the **lowest Cp** (even negative). Adding more predictors leads to **marginal improvements or overfitting**.
- This suggests **simplicity is preferred**—a model with just **2 predictors** explains the data well enough.

Influential Statistics:

Python Code:

```
# Calculate influence measures
influence = full_model.get_influence()
# Leverage (Hat values)
leverage = influence.hat_matrix_diag
# Cook's distance
cooks_d = influence.cooks_distance[0]
```

```
# Studentized residuals
studentized_residuals = influence.resid_studentized_external
# Combine results in a DataFrame
influence_df = pd.DataFrame({
    'Leverage': leverage,
    'Cooks_Distance': cooks_d,
    'Studentized_Residuals': studentized_residuals
}, index=df.index)
# Define thresholds
# Assuming 'p' should represent the number of predictors,
# it is calculated here using X.shape[1]
leverage_threshold = 2 * (X.shape[1] / X.shape[0])  # Common rule: 2 * (p/n) where p = X.shape[1], n = X.shape[0]
cooks_d_threshold = 4 / X.shape[0]  # Common rule: 4/n
studentized_residuals_threshold = 3  # Common rule: |residual| > 3
# Identify influential points
influence_df['Influential'] = (
    (influence_df['Leverage'] > leverage_threshold) |
    (influence_df['Cooks_Distance'] > cooks_d_threshold) |
    (influence_df['Studentized_Residuals'].abs() > studentized_residuals_threshold)
)
print("Influence Metrics:")
print(influence_df)
cleaned_df = df[~influence_df['Influential']]
```

Output:

Influence Metrics:

Leverage	Cooks Distance	Studentized-Residuals	Influential

0	0.150501	3.090371e-02	1.183472	True
1	0.029222	6.902277e-04	0.426687	False
2	0.078735	6.673992e-04	0.248867	False
3	0.108592	1.688162e-03	-0.331597	False
4	0.041688	2.635890e-03	0.694582	False
5	0.037526	5.933566e-04	-0.347516	False
6	0.020433	5.152713e-06	0.044250	False
7	0.038868	3.200555e-05	-0.079207	False
8	0.059092	1.887149e-03	0.488580	False
9	0.030357	1.105163e-03	-0.529658	False
10	0.043035	2.975057e-04	-0.229052	False
11	0.051948	4.613368e-03	-0.819462	False
12	0.035948	3.073973e-04	-0.255704	False
13	0.200899	2.844554e-02	-0.950986	True
14	0.032497	2.354051e-04	-0.235760	False
15	0.077950	5.864240e-02	2.406941	True
16	0.046569	5.791651e-05	0.096952	False
17	0.015612	9.002971e-05	-0.212172	False
18	0.090365	4.889222e-03	-0.625722	False
19	0.075541	1.313500e-03	-0.357165	False
20	0.026925	1.633234e-05	-0.068403	False
21	0.028267	6.192412e-07	0.012990	False
22	0.089215	2.387098e-03	-0.439902	False
23	0.025585	3.925905e-06	0.034427	False
24	0.075352	2.167200e-03	-0.459576	False
25	0.036417	1.829690e-03	0.620581	False
26	0.038922	2.691484e-04	0.229576	False
27	0.077627	2.349119e-02	1.502924	False
28	0.042839	1.114209e-04	0.140487	False
29	0.016819	4.398611e-05	0.142778	False
30	0.028809	9.299026e-04	-0.499060	False
31	0.028262	2.340314e-04	-0.252628	False
32	0.323933	1.360626e-01	1.516189	True
33	0.036365	1.188008e-04	-0.157985	False

34	0.020354	8.400567e-05	-0.179049	False
35	0.073006	2.479660e-06	0.015798	False
36	0.024772	6.474734e-04	-0.449922	False
37	0.051196	1.751019e-03	-0.507784	False
38	0.058016	5.516585e-05	-0.084263	False
39	0.122615	2.175232e-03	0.351453	True
40	0.031089	1.594052e-05	-0.062753	False
41	0.024965	4.841185e-05	-0.122433	False
42	0.047133	2.814313e-03	0.672966	False
43	0.024764	4.358524e-04	-0.369091	False
44	0.038684	3.808558e-04	-0.273995	False
45	0.035690	1.026497e-03	0.469355	False
46	0.066442	7.415227e-04	0.287489	False
47	0.071671	4.480124e-04	0.214516	False
48	0.201816	2.319160e-03	-0.269730	True
49	0.042819	2.723265e-04	-0.219717	False
50	0.053194	1.694806e-05	-0.048900	False
51	0.055621	1.278819e-03	-0.415192	False
52	0.021090	1.072370e-03	-0.629280	False
53	0.109659	2.606467e-08	0.001295	False
54	0.070321	1.368684e-02	1.205656	False
55	0.066271	1.057250e-03	0.343810	False
56	0.040567	7.754974e-05	-0.120582	False
57	0.018283	1.023513e-04	-0.208759	False
58	0.052857	2.102730e-03	-0.547259	False
59	0.072992	4.237808e-02	-2.107518	True
60	0.043706	1.664842e-03	0.538068	False
61	0.023880	1.883314e-03	-0.783367	False
62	0.035488	1.834581e-03	0.629827	False
63	0.435252	1.178233e-01	1.107049	True
64	0.119336	1.018883e-01	2.511607	True
65	0.088315	6.311106e-02	-2.328850	True
66	0.185065	9.741310e-05	0.058312	True
67	0.032860	3.607759e-03	-0.921030	False

68	0.046273	4.338708e-04	0.266328	False
69	0.059690	6.902169e-04	-0.293694	False
70	0.025758	5.270374e-04	-0.397795	False
71	0.050915	6.000448e-03	0.945486	False
72	0.021659	3.980920e-03	-1.201840	False
73	0.063919	1.204047e-02	-1.189966	False
74	0.024719	1.900555e-03	0.773087	False
75	0.063707	5.678262e-03	0.815829	False
76	0.049495	4.718423e-02	2.773836	True
77	0.082990	1.002482e-03	-0.296436	False
78	0.112380	1.442314e-02	-0.954256	False
79	0.044203	1.371946e-03	-0.485447	False
80	0.025188	3.964959e-06	0.034876	False
81	0.025108	9.504308e-04	-0.541580	False
82	0.035686	4.653902e-04	-0.315874	False
83	0.034837	2.796143e-05	0.078364	False
84	0.022873	2.823644e-04	0.309352	False
85	0.060532	1.454362e-03	-0.423339	False
86	0.015590	1.023056e-05	0.071559	False
87	0.031596	1.725965e-04	-0.204813	False
88	0.044203	1.816416e-02	-1.790444	False
89	0.013864	1.241175e-05	0.083656	False
90	0.066064	1.748198e-04	-0.139977	False
91	0.060786	2.661965e-02	1.833508	False
92	0.091849	2.448566e-03	0.438457	False
93	0.035669	1.441954e-03	-0.556683	False
94	0.039497	2.457714e-04	0.217708	False
95	0.116319	3.150274e-03	-0.435935	False
96	0.044791	6.204130e-03	-1.029095	False
97	0.024129	8.948859e-02	6.250320	True
98	0.021447	8.845531e-03	1.815705	False
99	0.176347	2.276903e-01	-3.023782	True
100	0.262268	6.270045e-02	-1.190103	True
101	0.035226	1.166281e-03	-0.503777	False

102 0.127782	3.803270e-03	-0.454060	True
103 0.079788	1.499924e-05	-0.037030	False
104 0.043633	1.140098e-03	-0.445470	False
105 0.094307	2.654738e-02	-1.435073	False
106 0.124232	5.162083e-02	1.721518	True
107 0.046672	3.043759e-04	-0.222045	False
108 0.433427	8.313037e-01	3.059437	True
109 0.030739	2.733591e-05	-0.082661	False
110 0.080790	1.605545e-02	-1.211473	False
111 0.025528	4.720681e-05	-0.119523	False
112 0.061948	8.586144e-04	-0.321181	False
113 0.167841	2.963999e-03	0.341485	True
114 0.073808	5.650445e-03	-0.751653	False
115 0.032952	1.093019e-05	0.050425	False
116 0.046471	2.618869e-03	-0.653933	False

This table provides **influence diagnostics** for observations in a regression model. Here's how to interpret the key metrics:

---

1. <u>Metrics Explanation</u>:

- **Leverage**:
  - Measures how far an observation's predictor values are from the mean of the predictor values.
  - **High leverage (closer to 1)** indicates an observation that can exert strong influence on the model fit.
- **Cook's Distance**:
  - Quantifies how much the regression coefficients change when an observation is excluded.
  - A **Cook's Distance > 0.5** signals **moderate influence**, and values **above 1** indicate **high influence**.

- **Studentized Residuals**:
  - Measures how much an observation deviates from the model's predicted value, accounting for the observation's leverage.
  - **Residuals > 2 or < -2** may indicate outliers or poor fit.
- **Influential**:
  - This binary column (**True/False**) flags whether an observation is **influential** based on a combination of these metrics.

---

2. Interpretation of the Results:

- **Influential Observations**:
  - Observations 0, 13, 15, 32, 39, 48, 59, 63, 64, 65, 66, 76, 97, 99, 100, 102, 106, 108, 113 are marked as **influential (True)**.
  - These points have either **high leverage, large Cook's Distance, or large residuals**.
- **Extreme Cases**:
  - **Observation 108** has a **very high Cook's Distance (0.831)** and a **large positive residual (3.06)**.
  - **Observation 99** has a **very negative residual (-3.02)** with high leverage.
  - **Observation 97** shows an **extremely large residual (6.25)**, suggesting a significant outlier.
- **Non-Influential Observations**:
  - Most observations show **low leverage (< 0.2)** and **small Cook's Distance (< 0.02)**, meaning they do not exert undue influence on the regression.

---

3. Observation:

- Model Stability:

The majority of data points are not influential, suggesting the **model is stable**.

Now, compare model with original model and check what improvement has been done from the original model.

Model Comparison:

1. Model Fit (R-squared and Adjusted R-squared):

- **Original Model:**
  - **R-squared:** 0.400 (40.0% of variance explained)
  - **Adj. R-squared:** 0.356 (35.6% of variance explained)
- **Refit Model (After Removing Influential Points):**
  - **R-squared:** 0.604 (60.4% of variance explained)
  - **Adj. R-squared:** 0.573 (57.3% of variance explained)

Interpretation:

- The refit model shows a **significant improvement** in fit, with a 20% increase in R-squared.
- Removing influential data points **substantially improved** how well the model explains unemployment variance.
-

2. F-statistic (Overall Model Significance):

- **Original Model:** 9.219 (p = 1.06e-08)
- **Refit Model:** 19.57 (p = 1.10e-15)

Interpretation:

- The refit model is **more statistically significant** overall.
- The F-statistic nearly **doubled**, suggesting the model's predictors collectively explain much more variation in unemployment after addressing influential data points.

3. Residual Diagnostics:

- **Durbin-Watson (Autocorrelation):**
  - **Original:** 2.019
  - **Refit:** 2.131
  - **Interpretation:** Both models show no autocorrelation (values close to 2).
- **Normality of Residuals (Jarque-Bera Test):**
  - **Original:** JB = 664.527 ($p < 0.001$, severe non-normality)
  - **Refit:** JB = 16.932 ($p < 0.001$, moderate non-normality)
  - **Interpretation:** The refit model shows **less severe non-normality** in residuals, indicating a better distribution after removing influential points.
  -
- **Kurtosis:**
  - **Original:** 14.432 (heavy tails, high kurtosis)
  - **Refit:** 4.716 (closer to normal distribution)

#ANOVA Model Comparison: Interpretation

ANOVA Table (Unrestricted vs. Null Model):

- **Null Hypothesis (H0):** The restricted model (null model) fits the data just as well as the unrestricted model (refit model).
- **Alternative Hypothesis (H1):** The unrestricted model provides a significantly better fit.

Results Breakdown:

Source	df	Sum of Squares	Mean Square	F-value	p-value (PR(>F))
C(group)	0.0	0.000000	NaN	NaN	NaN
Residual	97.0	798.275372	8.229643	NaN	NaN

Key Points:

1. **Degrees of Freedom (df):**
    - The degree of freedom for the group is **0.0**, suggesting no variation between groups in the model comparison.
2. **Sum of Squares:**
    - The sum of squares for the group is **0.000**, indicating **no additional explanatory power** from the unrestricted model.
3. **F-value and p-value:**
    - **F-value:** Not calculated (NaN).
    - **p-value (PR(>F)):** Not available, indicating **no significant difference** between the null and unrestricted models.

Interpretation:

- **Fail to reject the null hypothesis.**
- This suggests **the refit model does not significantly outperform** the simpler null model.
- The additional predictors (or the removal of influential points) did not **substantially improve** the model's performance in the context of group comparisons.

#Check for Multicollinearity with VIF

Python Code:

```
from statsmodels.stats.outliers_influence import variance_inflation_factor


# Calculate VIF
vif_data = pd.DataFrame()
vif_data["Variable"] = X_cleaned.columns
vif_data["VIF"] = np.sqrt([variance_inflation_factor(X.values, i) for i in range(X.shape[1])])
print(vif_data)
```

Result:

VIF Results Breakdown:

Variable	VIF	Interpretation
GDP growth (annual %)	1.30	Low multicollinearity, no concern.
Unemployment, youth total (% of labor force 15-24)	1.64	Acceptable multicollinearity.

Variable	VIF	Interpretation
Labor force participation rate, total (%)	3.97	Moderate multicollinearity, monitor closely.
Inflation, consumer prices (annual %)	1.24	Low multicollinearity.
Urban population (% of total population)	3.29	Moderate multicollinearity.
Trade Openness (% of GDP)	1.95	Acceptable, but slightly increasing.
Gross capital formation (% of GDP)	3.21	Moderate multicollinearity.

Analysis:

No severe multicollinearity (VIF > 10) detected.

- **Variables with VIF > 3 (moderate multicollinearity):**
  a) **Labor force participation rate** (3.97)
  b) **Urban population** (3.29)
  c) **Gross capital formation** (3.21)

To remove multicollinearity, we will use penalized regression:

Result we getting:

```
Best Ridge alpha: 32.745491628777316
Ridge Coefficients: [-0.80161429  2.22043822 -0.36583899  0.18039306  0.13643507 -0.30039971
 -0.06633797]
Best Lasso alpha: 0.572236765935022
Lasso Coefficients: [-0.49198805  2.477345   -0.          0.          0.         -0.
 -0.        ]
Best Elastic Net alpha: 0.49770235643321137
Elastic Net Coefficients: [-0.65550905  2.16449004 -0.17701147  0.01710183  0.         -0.06775329
 -0.        ]
```

Interpretation from Result:

- **Ridge Regression**:

- **Best Alpha**: 32.75 – This indicates a moderate level of regularization.
- **Coefficients**: All features retain non-zero coefficients, suggesting that Ridge penalizes large coefficients but does not force them to zero. This helps reduce the impact of multicollinearity without completely eliminating any variables.
- **Interpretation**: Ridge stabilizes the model by shrinking coefficients, which helps mitigate the effect of multicollinearity. However, all features still contribute to the prediction, ensuring the model captures the relationships present in the data.

- **Lasso Regression**:

- **Best Alpha**: 0.57 – Lower regularization compared to Ridge.
- **Coefficients**: Only the second feature (Unemployment, youth) retains a non-zero coefficient. All other coefficients are shrunk to zero.
- **Interpretation**: Lasso performs feature selection by eliminating less important features, keeping only the most relevant predictors. This simplifies the model and highlights that "Unemployment, youth total" is the strongest predictor of total unemployment.

- **Elastic Net Regression**:

- **Best Alpha**: 0.49 – Elastic Net balances L1 and L2 penalties.
- **Coefficients**: Several coefficients are non-zero but smaller than in Ridge. Some coefficients are shrunk to zero, blending the feature selection of Lasso with the shrinkage of Ridge.
- **Interpretation**: Elastic Net selects a subset of important features but retains small contributions from others. This approach is effective when multicollinearity is present, providing a compromise between Ridge and Lasso.

Conclusion:

The developed model aims to predict total unemployment rates based on key economic indicators, such as GDP growth, youth unemployment, labor force participation, inflation, urbanization, trade openness, and capital formation. Through extensive preprocessing, linearity checks, and diagnostic tests (heteroscedasticity, normality), the model has been refined to address multicollinearity and outlier influence.

The use of penalized regression techniques—Ridge, Lasso, and Elastic Net—has effectively mitigated multicollinearity by shrinking or eliminating less significant coefficients, resulting in a more stable and interpretable model. The final model achieves an R^2 value of **0.604**, indicating that approximately 60.4% of the variance in unemployment rates can be explained by the selected predictors.

While the model exhibits reasonable explanatory power, the remaining unexplained variance suggests the presence of external factors or nonlinear relationships not fully captured by the current features. Nonetheless, this model serves as a valuable tool for policymakers and analysts, offering insights into how different economic forces collectively influence unemployment. It enables more informed decision-making and targeted interventions to manage labor market dynamics.

GOOGLE COLAB FILE FOR PYTHON PROGRAMMING:

CLICK HERE TO ACCESS PYTHON CODES

DATASET USED

DATASET (UNEMPLOYMENT AND OTHER ECONOMIC VARIABLES)