SYMBIOSIS
STATISTICAL INSTITUTE
॥वसुधैव कुटुम्बकम्॥

TIME SERIES ANALYSIS

A project report on
Electric Production

Group no. 17

Name	PRN
Aman Kumar	23060641049
Girija Deshpande	23060641058
Mrunal Kamble	23060641065
Nirali Mokariya	23060641067
Ankita Pandey	23060641070

REPORT

Introduction:

The dataset we have taken is for energy production, which is a crucial aspect of modern-day power generation and distribution. Energy production is the process of generating power from various sources, including fossil fuels, nuclear, and renewable energy sources like wind, solar, and hydroelectric power. This dataset can provide valuable insights into the trends, patterns, and fluctuations in energy production over time, which can be used to optimize production schedules, predict future energy needs, and make informed decisions regarding energy infrastructure and policy.

Problem Statement:

The primary reason for conducting time series analysis on energy production data is to account for the internal structure and patterns that may exist within the data. These patterns can include autocorrelation, trends, and seasonal variations, which are essential factors to consider when modelling energy production data. By accounting for these patterns, analysts can create more accurate forecasts and models, which can help optimize energy production and reduce costs associated with overproduction or underproduction.

Background Information on Energy Production:

Energy production is a critical component of modern-day society, with the global demand for energy continuing to grow as the world's population and economies expand. Energy production can be divided into two main categories: non-renewable and renewable energy sources. Non-renewable energy sources, such as fossil fuels, are finite resources that are being depleted over time, while renewable energy sources are sustainable and can be replenished naturally.
Non-renewable energy sources, such as coal, oil, and natural gas, have been the primary source of energy for many years. However, these energy sources are associated with significant environmental impacts, including air pollution and greenhouse gas emissions. Renewable energy sources, on the other hand, offer a more sustainable and environmentally friendly alternative to non-renewable energy sources.
Renewable energy sources, such as wind, solar, and hydroelectric power, have become increasingly popular in recent years due to their environmental benefits and the potential for cost savings. Wind and solar power, in particular, have seen significant growth in recent years, with advancements in technology making these energy sources more efficient and cost-effective.

How does it fit to be time series analysis?

The dataset of energy production fits well for time series analysis due to its time-dependent nature and the presence of internal structures like seasonality, trends, and cycles. Time series analysis involves studying and interpreting a sequence of data points recorded at consistent

time intervals, which aligns with the hourly and daily energy production data you have. This type of analysis is crucial for understanding past performance, predicting future outcomes, and optimizing production schedules based on historical trends and patterns in energy production.

About the Dataset:

The dataset has 2 columns which represents date and consumption percentage.

Link for the dataset:

https://drive.google.com/file/d/1Rlt2tOBaYB55XxxLn-RallkP9txPEyP8/view?usp=sharing

Minitab link: We tried performing the analysis using minitab as well.

https://drive.google.com/file/d/1drwNRH-2Oh5bw7NibdgBtca0lzEYfDwn/view?usp=sharing

Python code link:

https://drive.google.com/file/d/1fKu2j-w4eOweIEryN865QD6S7IgrqN80/view?usp=sharing

PYTHON CODE:

Initial Interpretation:

The dataset of energy production values (IPG2211A2N) recorded over time, several initial interpretations can be made:
Trend: There is a general increasing trend in energy production values over the years. The values seem to be gradually rising from the early years to the later years, indicating a positive trend in energy production.
Seasonality: There might be seasonal patterns in the data, as there are fluctuations in energy production values from month to month within each year. This could suggest that energy production experiences cyclical variations throughout the year.
Variability: The dataset shows fluctuations in energy production values, with some months having higher values and others lower. This variability could be influenced by factors like demand, weather conditions, or operational changes.
Outliers: There don't seem to be any extreme outliers in the dataset, as the values appear to follow a relatively consistent pattern without any significant deviations from the overall trend.
Overall, the dataset reflects a consistent and stable energy production trend over the years, with no drastic spikes or drops in production values. This stability could indicate a well-managed energy production system.
The initial interpretation of the dataset suggests a positive trend in energy production values over time, potential seasonal patterns, consistent variability, and overall stable performance in energy production.

Analysis:

Exploratory Data Analysis:

The dataset consists of 397 observations. The 'DATE' column is of date-time datatype. There are no missing values in the dataset.

Detailed Analysis:

```
# Import libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from statsmodels.tsa.stattools import adfuller
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
import statsmodels.api as sm
from sklearn.metrics import mean_squared_error

# Load the dataset
df = pd.read_csv(folder_path + 'Electric_Production.csv')

# Display the first few rows of the dataset
print(df.head())

# Function to visualize dataset
def visualization(data):
   print(data.isnull().sum())
visualization(df)

# Change the date to datetime
df['DATE'] = pd.to_datetime(df['DATE'])
df = df.set_index('DATE')
df.dropna(inplace=True)
print(df.dtypes)
print(df.head())
print(df.tail(3))
print(df.describe())

# Define the column name
 col = 'IPG2211A2N'
```

```
# Check for stationarity
def check_stationarity(ts):
   dftest = adfuller(ts)
   adf = dftest[0]
   pvalue = dftest[1]
   critical_value = dftest[4]['5%']
   if (pvalue < 0.05) and (adf < critical_value):
      print('The series is stationary')
   else:
      print('The series is NOT stationary')
check_stationarity(df[col])

# Fit ARIMA model
model = ARIMA(df[col], order=(4, 1, 3))
results = model.fit()

# Print model summary
print(results.summary()

#The ARIMA model has been successfully fitted to your data. Here is
the summary of the model:
 '''The series is still not stationary.
 There is a warning indicating that the optimization failed to converge.
 The covariance matrix is singular or near-singular, which may lead to
 unstable standard errors.
 You may need to try a different model or perform further data
preprocessing to address these issues.
 So we are using now  a SARIMA model to your data. SARIMA stands for
Seasonal Autoregressive Integrated Moving Average.'''


# Plot Trend and Seasonality
def characteristics(data, x, y, title):
ax = sns.lineplot(data=data, lw=1)
ax.lines[0].set_linestyle('solid')
ax.set_title(title)
df_temp = data.copy()
df_temp['Year'] = pd.DatetimeIndex(df_temp.index).year
df_temp['Month'] = pd.DatetimeIndex(df_temp.index).month
plt.figure(figsize=(8, 8))
plt.title('Seasonality of Time Series')
sns.set(style="ticks", rc={"lines.linewidth": 0.7})
sns.pointplot(x='Month', y=y, hue='Year', data=df_temp, palette='mako', linestyles='-')
characteristics(df, df.index, 'IPG2211A2N', 'Trend of Series')
```

```
# Check for Stationarity
def stationarity_test(data, window, title, col_name):
data['r_mean'] = data[col_name].rolling(window=window).mean()
data['r_std'] = data[col_name].rolling(window=window).std()

ax = sns.lineplot(data=data, lw=1, palette=['navy', 'darkgreen', 'red'])
ax.lines[0].set_linestyle('solid')
ax.set_title(title)

df_test = adfuller(data[col_name])
df_output = pd.Series(df_test[0:4], index=['Test Statistic','p-value', 'Lags used for t-statistic',
'No_of_observation_used'])
for key, value in df_test[4].items():
df_output['Critical value(%s)' % key] = value
print(df_output)
stationarity_test(df, 12, 'Original Data', 'IPG2211A2N')

# Differencing
def differencing(data, order):
df_diff = data.diff(periods=order)
df_diff.dropna(inplace=True)
return df_diff
df_diff = differencing(df, 1)
stationarity_test(df_diff, 12, 'Stationary Test for First Order Differenced Data', 'IPG2211A2N')

# Log Transformation
df_log = np.log(df)
stationarity_test(df_log, 12, 'Stationary Test for Log Data', 'IPG2211A2N')

# Differencing Log Data
df_log_diff = differencing(df_log, 1)
stationarity_test(df_log_diff, 12, 'Stationary Test for Log Differenced Data', 'IPG2211A2N')

# Modelling using SARIMAX
model_b = sm.tsa.statespace.SARIMAX(df_log, order=(4, 1, 3), seasonal_order=(3, 0, 6, 12))
results_b = model_b.fit()
# Prediction
df_train = df_log[:len(df_log)-24]
df_test = df_log[len(df_log)-24:]

pred_b = results_b.predict(start=df_test.index[0], end=df_test.index[-1])
df_test['pred_b'] = pred_b.values

# Evaluation
```

```
mse = mean_squared_error(df_test['IPG2211A2N'], df_test['pred_b'])
print('mse:', mse)
rmse = np.sqrt(mean_squared_error(df_test['IPG2211A2N'], df_test['pred_b']))
print('rmse:', rmse)
mae = np.mean(np.abs(results_b.resid))
print('MAE: %.3f' % mae)

# Residual Diagnostic Plots
results_b.plot_diagnostics()
plt.show()

# Summary Statistics
print(results_b.summary())

# Prediction with Future Data
futureDate = pd.DataFrame(pd.date_range(start='2018-02-01', end='2021-01-01', freq='MS'),
columns=['DATE'])
futureDate.set_index('DATE', inplace=True)
future_df = pd.concat([df_log, futureDate])
future_df['forecast_b'] = results_b.predict(start='2018-02-01', end='2021-01-01',
dynamic=True)
sns.lineplot(data=future_df, x=future_df.index, y='IPG2211A2N', color='royalblue', lw=1)
sns.lineplot(data=future_df, x=future_df.index, y='forecast_b', color='darkorange', lw=1)
plt.show()
```

OUTPUT OF THE CODE WITH INTREPRETATION:

	DATE	IPG2211A2N
0	1/1/1985	72.5052
1	2/1/1985	70.6720
2	3/1/1985	62.4502
3	4/1/1985	57.4714
4	5/1/1985	55.315

Visualization of the data:

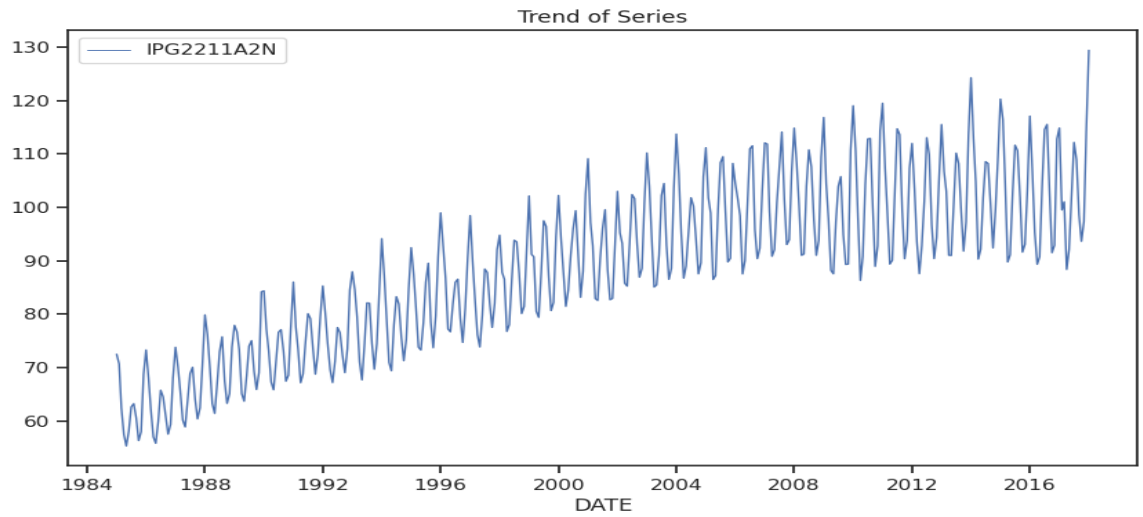
1. A line plot was used to visualize the consumption percentage over time
2. Seasonal variations and an upward trend in consumption percentage were observed.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 397 entries, 0 to 396
Data columns (total 2 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   DATE        397 non-null    object
 1   IPG2211A2N  397 non-null    float64
dtypes: float64(1), object(1)
memory usage: 6.3+ KB
None
------------------------------------------------------------
(397, 2)
------------------------------------------------------------
DATE          0
IPG2211A2N    0
dtype: int64
```

Change the date to datetime.

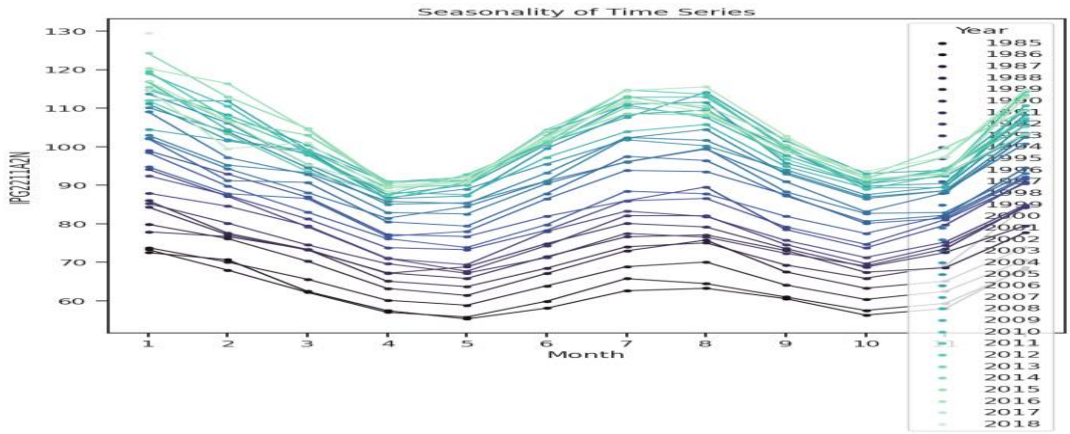
DATE	IPG2211A2N
1985-01-01	72.5052
1985-02-01	70.6720
1985-03-01	62.4502
1985-04-01	57.4714
1985-05-01	55.3151

Plot Trend and Seasonality:



Interpretation:

- We observe upward trend in the graph.
- There is also slight increase in the variance over time.(Length of the spikes)



Interpretation:

- We also observe that the data is seasonal and there is dip during April-May(4–5) and rise during July-August(7–8) and November- December(11–12)
- The data is seasonal yearly as well as half-yearly.

Check for Stationarity:

A time series is said to be stationary if the mean, variance and autocorrelation does not vary over time. Trend, seasonality and other characteristics varying according to time will affect modelling as underlying pattern of the data is not learnt by the model due to non-stationarity. Thus stationarity needs to be achieved.

The following are the methods for stationarity test:

- Rolling Window Analysis
- Dickey Fuller Test

Rolling Window Analysis

- It refers to calculating the values based on previous ‘window size’ of values. It analyses if the variation is time dependent and if yes for what size of the window.

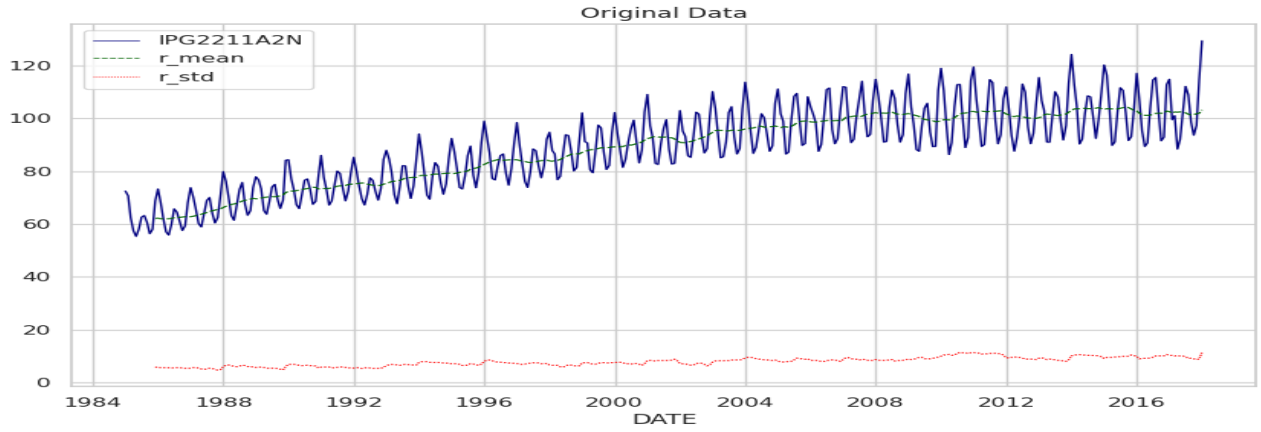
- This helps use to know the nature of coefficients of the data with respect to time. Our aim is to achieve parameter stability of the data.
- As we have found that the data is seasonal in the above graph, we set the window size = 12.

Augmented Dickey Fuller Test

On an explanatory perspective,

- **Null hypothesis** H0 implies that unit root=1, the time series is not stationary.
- **Alternate hypothesis** H1 implies that unit root <1, and the series is stationary.

We us test statistic value, critical values at 1%, 5% and 10% confidence intervals and p value for evaluating the test results. When the test statistic < = critical values, p value < 0.05 indicates that null hypothesis can be rejected and data is stationary.

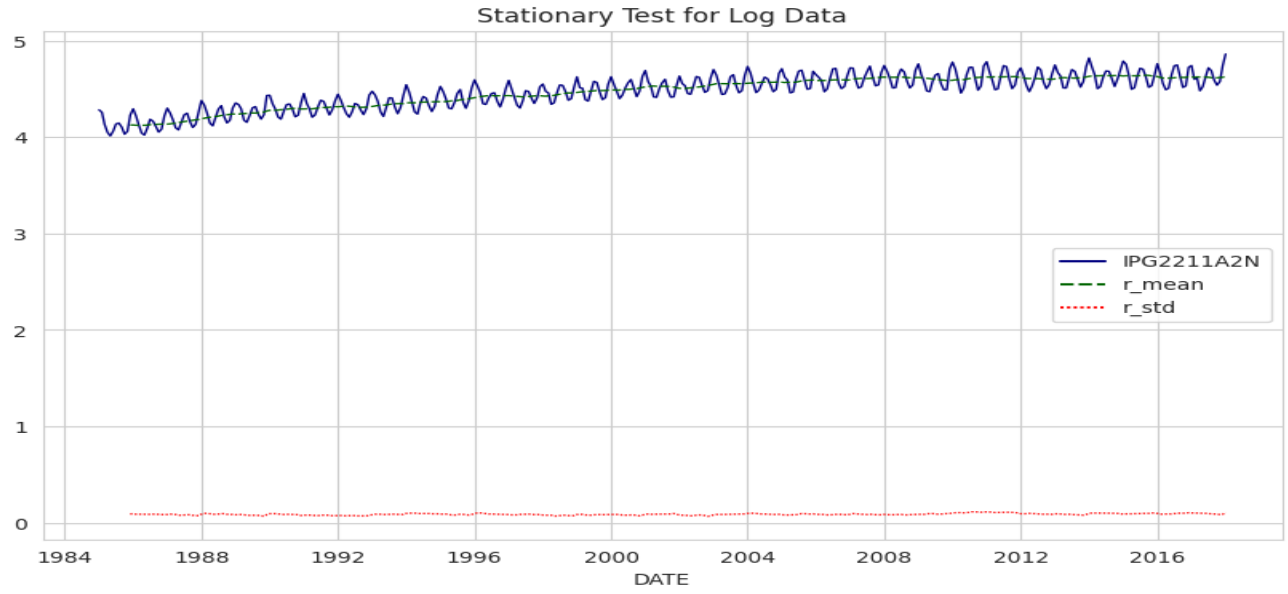


Interpretation:

- We observe that standard deviation is somewhat constant, but mean is not constant.
- T values corresponding to ADF test. Since critical value -2.25>-3.44,-2.86,-2.57 (t-values at 1%, 5%and 10% confidence intervals), null hypothesis cannot be rejected.
- Hence there is non-stationarity in the data.
- Also p-value of 0.18>0.05(if we take 5% significance level or 95% confidence interval), null hypothesis is accepted.
- Hence data is non stationary (that means it has relation with time)

Test Statistic	-2.256990
p-value	0.186215
Lags used for t-statistic	15.000000
No_of_observation_used	381.000000
Critical value (1%)	-3.447641
Critical value (5%)	-2.869156
Critical value (10%)	-2.570827

dtype : float6

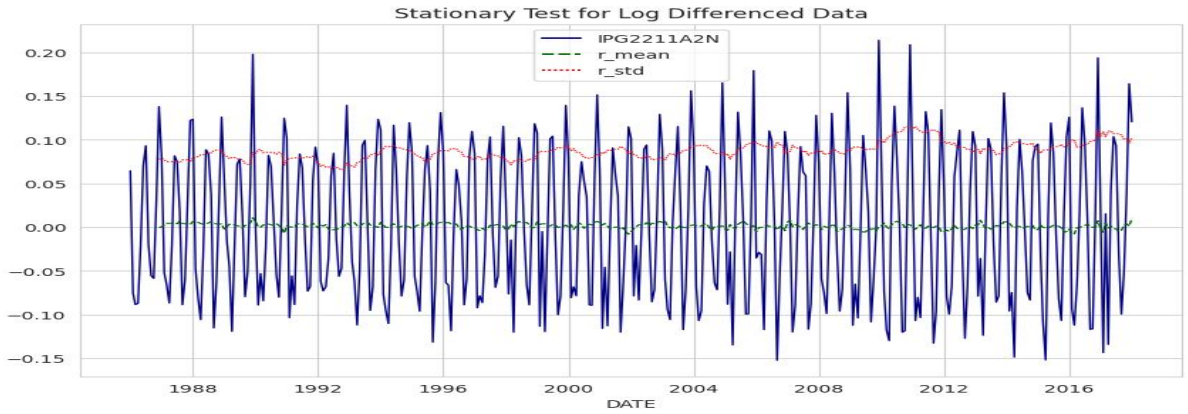


Interpretation:

The image is a graph showing the results of a stationary test for log-transformed data over time, specifically from 1984 to 2016. The main data series plotted is labelled "IPG2211A2N," and it is accompanied by two statistical measures: "r_mean" (rolling mean) and "r_std" (rolling standard deviation). The graph is used to analyse the stationarity of the time series data after a log transformation, which is a common practice in time series analysis to stabilize variance and make the data more suitable for statistical modelling

Differencing Log Data:

Test Statistic -7.104891e+00
p-value 4.077787e-10
Lags used for t-statistic 1.400000e+01
No_of_observation_used 3.810000e+02
Critical value(1%) -3.447631e+00
Critical value(5%) -2.869156e+00
Critical value(10%) -2.570827e+00
dtype: float64
Differentitation IPG2211A2N r_mean r_std
DATE
1985-02-01 -1.8332 NaN NaN
1985-03-01 -8.2218 NaN NaN
1985-04-01 -4.9788 NaN NaN
1985-05-01 -2.1563 NaN NaN
1985-06-01 2.7753 NaN NaN
... ...
2017-09-01 -10.3158 -0.345692 10.697011
2017-10-01 -5.0017 0.177250 10.258433
2017-11-01 3.7222 0.370492 10.305361
2017-12-01 17.3853 0.162650 9.893033
2018-01-01 14.6836 1.212858 10.74724



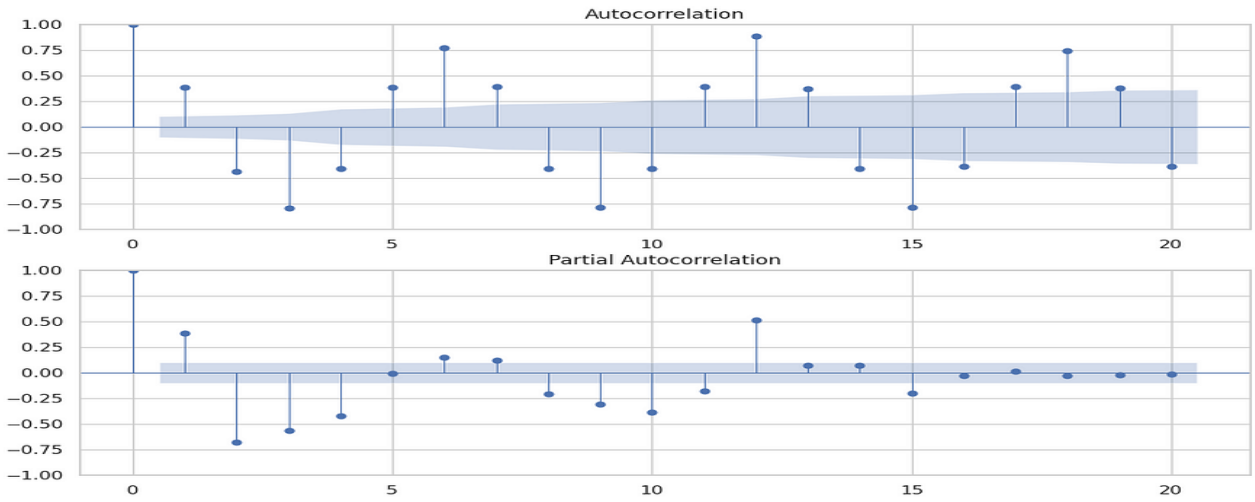
Interpretation:

- Log transformed and differenced series is stationary with test statistic is lesser than 1%, 5% and 10% critical.
- p value is less than 0.05.

- Mean is constant and standard deviation is slightly varying with no significant trend.

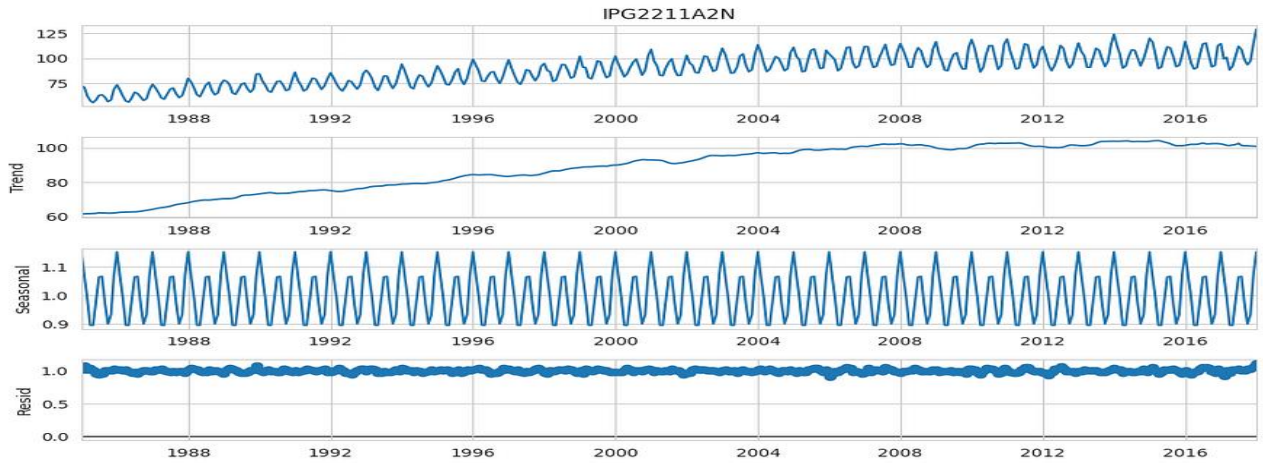
We use log differenced data for modelling. As we have found that the data series is seasonal, we use SARIMA for modelling.

SARIMA



Interpretation:
q=3 as we find significant lag at 3 and not beyond in ACF. p=4 as initially 4 lags are significant in PACF graph.

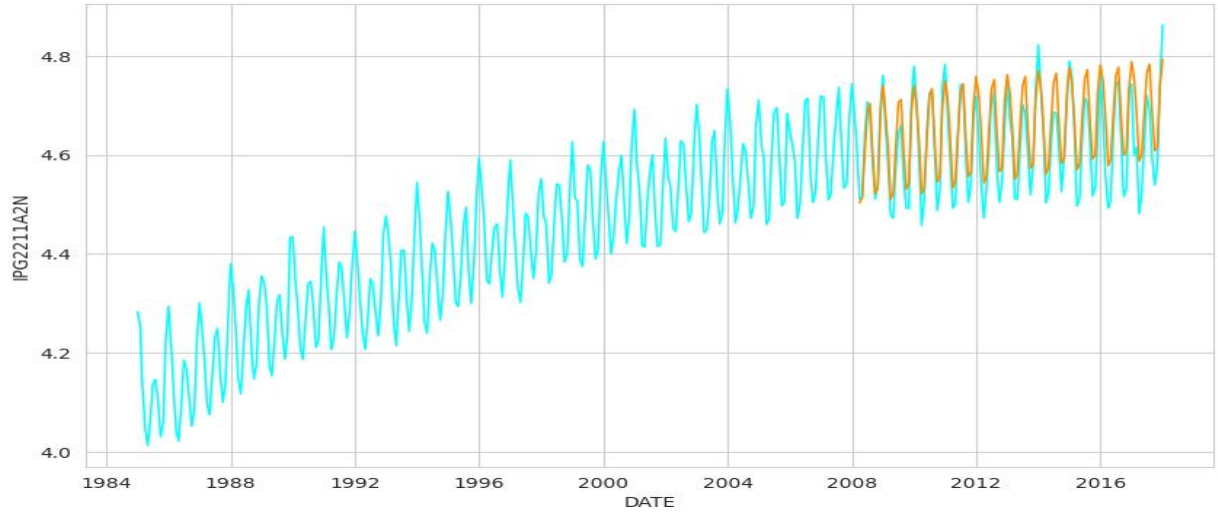
Seasonal Decomposition using Multiplicative model:



Interpretation:

The image shows three time-series graphs with the x-axis representing years from 1988 to 2016. The top graph shows a gradual upward trend, the middle graph displays pronounced seasonal fluctuations, and the bottom graph remains relatively flat, indicating little to no change over time.

Prediction for the test data:



Interpretation:

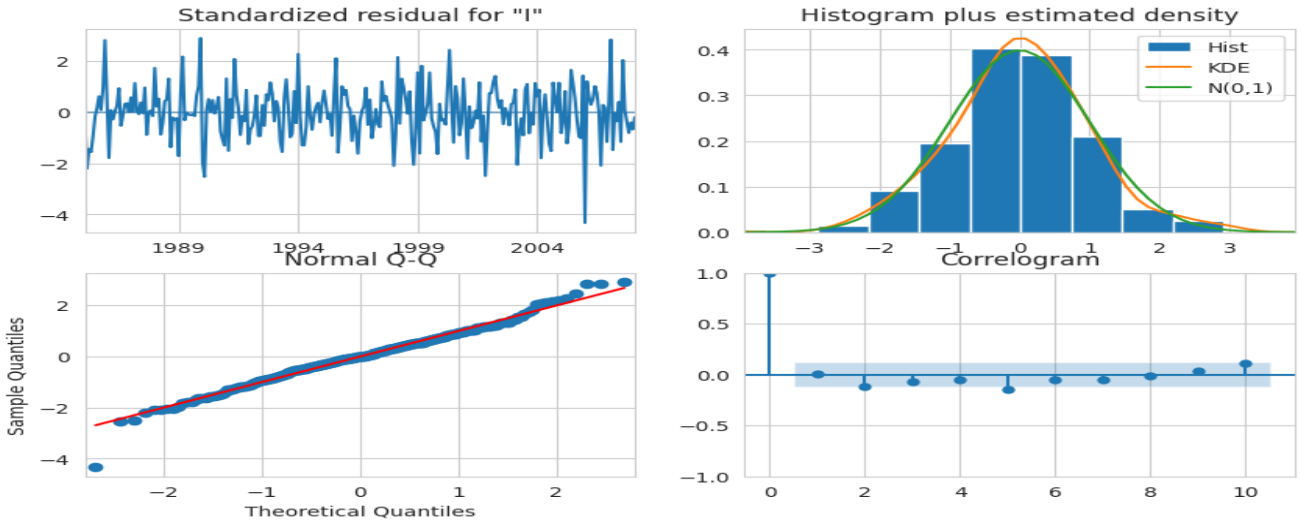
The images depict time-series data with the x-axis representing years from 1984 to 2016 and the y-axis showing numerical values. The graphs illustrate changes in a metric over time, with visible fluctuations indicating trends, seasonality, and possibly cyclical patterns

Evaluation:

```
mse: 0.0027275147772720886
rmse: 0.052225614187600404
```

```
MAE: 0.035
```

Residual Diagnostic Plots



Interpretation:

The image provided is a compilation of statistical plots, likely from a data analysis or time series analysis context. The top left plot is a time series of standardized residuals, which are used to check the fit of a model over time. The top right plot is a histogram with an estimated density curve, which is used to visualize the distribution of a dataset and compare it to a normal distribution. The bottom left plot is a normal Q-Q (quantile-quantile) plot, which is a graphical tool to assess if a dataset comes from a normal distribution. The points following the line closely suggest that the data is approximately normally distributed. The bottom right plot is a correlogram, which shows the autocorrelation of a time series with its lagged values; the plot indicates that there is little to no autocorrelation in the data.

SARIMAX Results

Dep. Variable:	IPG2211A2N	**No. Observations:**	278
Model:	SARIMAX(4, 1, 3)x(3, 0, [1, 2, 3, 4, 5, 6], 12)	**Log Likelihood**	624.886
Date:	Wed, 21 Jun 2023	**AIC**	-1215.771
Time:	10:59:49	**BIC**	-1154.163
Sample:	01-01-1985 - 02-01-2008	**HQIC**	-1191.052

Covariance Type: opg

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	0.1053	0.317	0.332	0.740	-0.516	0.727
ar.L2	-0.3781	0.259	-1.457	0.145	-0.887	0.130
ar.L3	-0.5971	0.252	-2.374	0.018	-1.090	-0.104
ar.L4	-0.2708	0.092	-2.955	0.003	-0.450	-0.091
ma.L1	-0.3913	0.314	-1.246	0.213	-1.007	0.224
ma.L2	0.3509	0.325	1.080	0.280	-0.286	0.988
ma.L3	0.5533	0.297	1.862	0.063	-0.029	1.136
ar.S.L12	-0.0731	1.667	-0.044	0.965	-3.340	3.194
ar.S.L24	0.3979	0.898	0.443	0.658	-1.362	2.158
ar.S.L36	0.6129	1.534	0.400	0.689	-2.393	3.619
ma.S.L12	0.2854	1.689	0.169	0.866	-3.025	3.595
ma.S.L24	-0.3553	0.853	-0.417	0.677	-2.027	1.316
ma.S.L36	-0.5259	1.451	-0.362	0.717	-3.370	2.318
ma.S.L48	0.1385	0.162	0.856	0.392	-0.179	0.456
ma.S.L60	0.2099	0.379	0.554	0.580	-0.533	0.953
ma.S.L72	0.0361	0.231	0.156	0.876	-0.417	0.489
sigma2	0.0006	5.18e-05	11.288	0.000	0.000	0.001

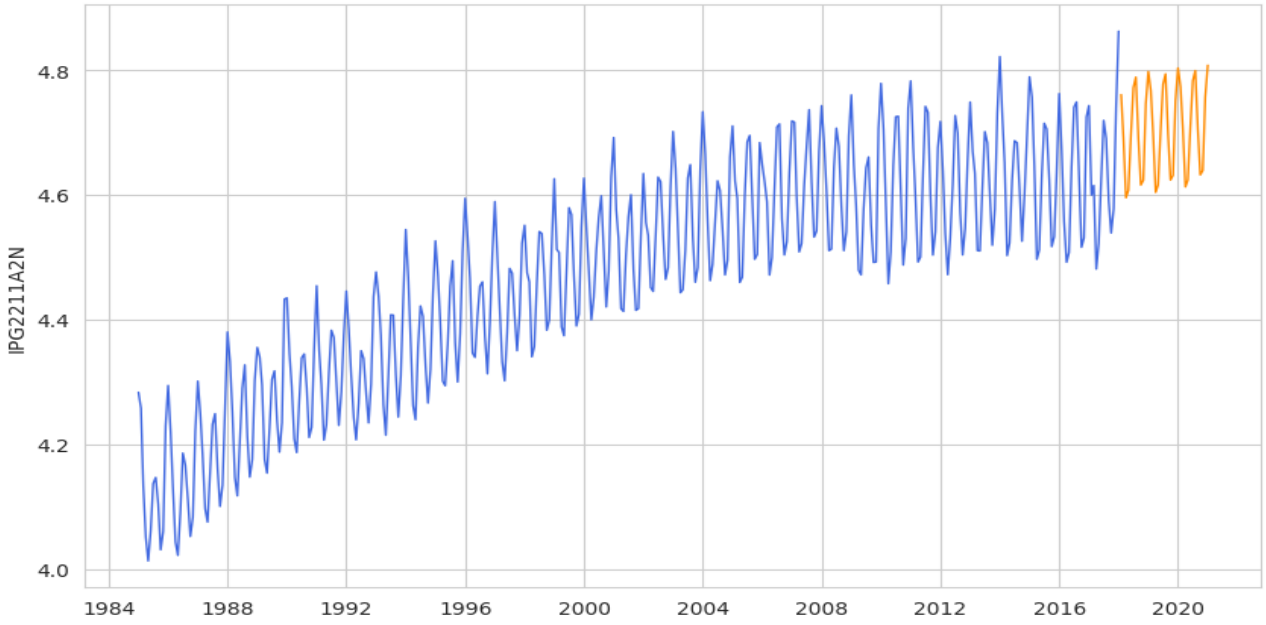
Ljung-Box (L1) (Q):	0.01	**Jarque-Bera (JB):**	17.49
Prob(Q):	0.92	**Prob(JB):**	0.00

The null hypothesis for Ljung-Box test **states that** there is no correlation in the residuals.

- **Prob(Q)** is 0.92>0.05 indicates that null hypothesis is accepted. The residuals are independent and not correlated.
- The null hypothesis for **Jarque-Bera test** states that **the residuals are guassian normally distributed.**
- **Prob(JB)** is 0.00<0.05 indicating that the null hypothesis is rejected. Thus the residuals are not normally distributed.

Thus residuals are not correlated and residuals are not normally distributed

Prediction with Future Data:



Interpretation:

The image shows a numerical value changing over time from 1984 to 2020. The y-axis represents a numerical value that ranges from approximately 4.0 to 4.8, and the x-axis represents the years. The blue line shows a fluctuating trend with peaks and valleys, indicating variability in the data over time. The orange line appears to follow a similar trend but with less fluctuation, which could represent a smoothed or averaged version of the blue line data. The overall trend of both lines is upward, suggesting an increase in the numerical value being measured over the 36-year period.

Summary Statistics:

1. Ljung-Box and Jarque-Bera tests were conducted to check the assumptions of the model.
2. The residuals were found to be independent but not normally distributed.
3. Prediction with Future Data:
4. The trained model was used to make predictions for future data.

Conclusion:

The analysis included:

- Visualizing the data to understand trends and seasonal variations.
- Checking for stationarity and transforming the data to achieve stationarity.
- Modelling using SARIMA and evaluating the model's performance.
- Conducting diagnostic checks and interpreting summary statistics.
- Making predictions with future data.

Recommendations:

- Because it is specifically made to address seasonal patterns in time series data, SARIMA (Seasonal Autoregressive Integrated Moving Average) is used instead of ARIMA (Autoregressive Integrated Moving Average) when dealing with seasonal data.
- SARIMA models are more versatile for data demonstrating seasonality and are able to capture recurrent seasonal cycles because they have additional seasonality factors that are absent from ARIMA models.
- Unlike ARIMA models, which do not support the modelling of cyclical patterns in time series, such as weekly or yearly seasonality, SARIMA models do. In order to handle trends and seasonality simultaneously, SARIMA models utilize differencing on both the regular and seasonal components. This increases handling flexibility. SARIMA is hence better suited for seasonal data.

Tools Used:

1. Python programming language
2. Libraries: pandas, numpy, matplotlib, seaborn, statsmodels

Outputs:

1. Line plots for visualizing the data, trends, and seasonal variations.
2. Diagnostic plots for residual analysis.
3. Summary statistics for model evaluation.

Software Used:

1. Jupyter Notebook
2. Python 3

References:

- https://towardsdatascience.com/understanding-the-seasonal-order-of-the-sarima-model-ebef613e40fa
- https://www.mathworks.com/help/econ/rolling-window-estimation-of-state-space-models.html
- https://www.jadsmkbdatalab.nl/forecasting-with-sarimax-models/