CHRIST
(DEEMED TO BE UNIVERSITY)
BANGALORE · INDIA

CIA-3

ASSIGNMENT

Time Series Analysis of GDP Data- Residual Analysis

By

Ann Mathew

2348010

4 MSc Data Science

(Time Series Forecasting)

Introduction

The Gross Domestic Product (GDP) is a crucial economic indicator that reflects the economic health of a country. This report aims to analyze the GDP time series data of a selected country to understand its underlying patterns, model the data using an ARMA (AutoRegressive Moving Average) model, and perform residual analysis to validate the model's effectiveness. This report focuses on the time series analysis of India's GDP from 1960 to 2020 using the ARMA (AutoRegressive Moving Average) modeling approach. The objective is to transform the data into a stationary series, identify the appropriate ARMA model, fit the model, and perform residual analysis to ensure its adequacy.

Objectives

- To transform India's GDP time series data into a stationary series.
- To identify an appropriate ARMA model using ACF and PACF plots.
- To fit the selected ARMA model to the GDP data.
- To perform residual analysis to validate the model.

Methodology and Results

Loading the Data

The dataset chosen for this analysis is the GDP of India from the World Bank, which can be downloaded as a CSV file.

```
# Install and load the required packages
install.packages("WDI")
```

```
## Installing package into 'C:/Users/Ann Mathew/AppData/Local/R/win-library/4
.3'
## (as 'lib' is unspecified)
```

```
## package 'WDI' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\Ann Mathew\AppData\Local\Temp\RtmpeokLjy\downloaded_packages
```

```
library(WDI)
```

```
## Warning: package 'WDI' was built under R version 4.3.3
```

```
library(tseries)
```

```
## Warning: package 'tseries' was built under R version 4.3.3
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 4.3.3
```

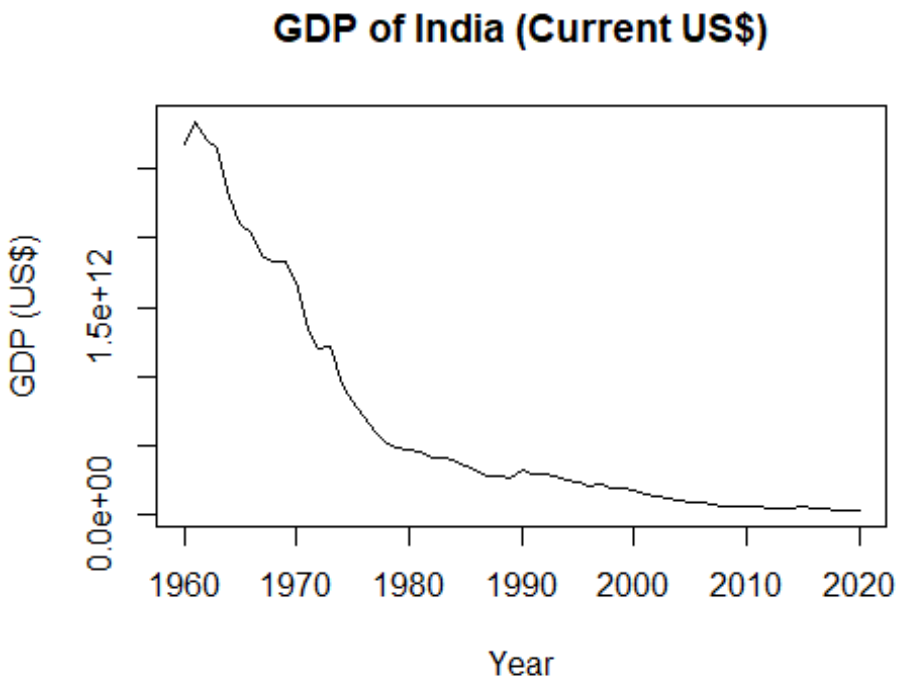
```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.3.3
```

```
# Load the GDP data for India from the World Bank
gdp_data <- WDI(country = "IN", indicator = "NY.GDP.MKTP.CD", start = 1960, e
nd = 2020)

# Inspect the data
head(gdp_data)
```

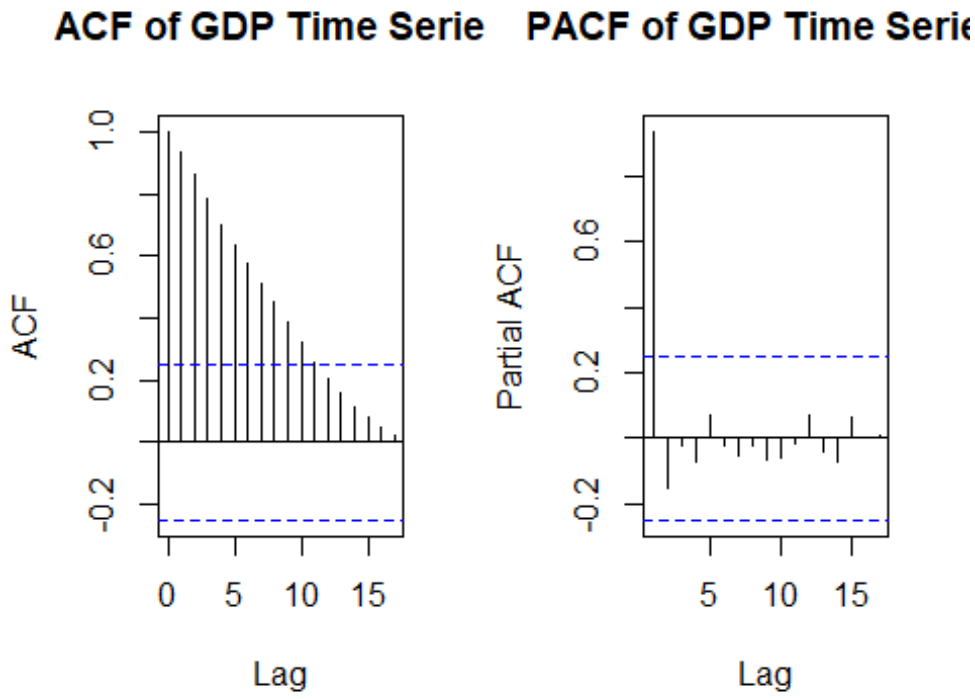
```
##   country iso2c iso3c year NY.GDP.MKTP.CD
## 1   India    IN   IND 2020   2.674852e+12
## 2   India    IN   IND 2019   2.835606e+12
## 3   India    IN   IND 2018   2.702930e+12
## 4   India    IN   IND 2017   2.651474e+12
```

```
## 5   India    IN   IND 2016   2.294797e+12
## 6   India    IN   IND 2015   2.103588e+12

# Convert the data to a time series object
gdp_ts <- ts(gdp_data$NY.GDP.MKTP.CD, start = c(1960), frequency = 1)

# Plot the original GDP time series
plot(gdp_ts, main="GDP of India (Current US$)", ylab="GDP (US$)", xlab="Year"
)
```



Description: Here, we load the GDP data for India from 1960 to 2020 using the WDI package. The data is converted into a time series object to facilitate further analysis. The original GDP time series is then plotted to visualize its trend.

```r
# Plot the ACF and PACF
par(mfrow=c(1,2)) # Set up the plotting area for two plots
acf(gdp_ts, main="ACF of GDP Time Series")
pacf(gdp_ts, main="PACF of GDP Time Series")
```

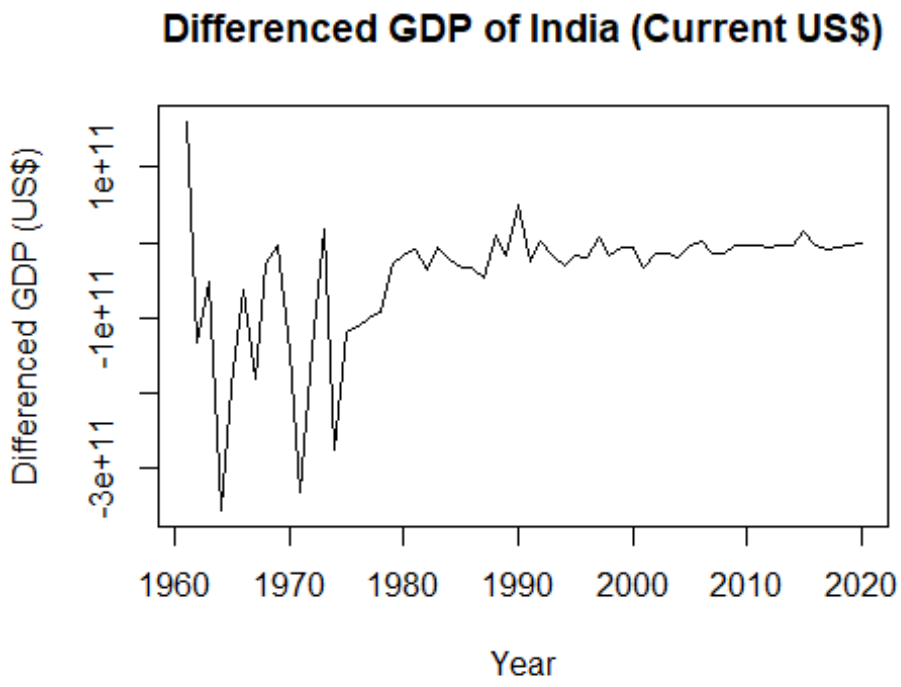


The ACF plot shows a gradually decreasing pattern, indicating that the GDP time series is highly correlated with its past values over several periods. This pattern suggests that the time series may be non-stationary, likely with a trend component. The slow decay often suggests the presence of an autoregressive process (AR) or that differencing may be required to make the series stationary.

The PACF plot shows a significant spike at lag 1 and then drops off quickly. This suggests that the series may follow an AR(1) process, meaning the series can be adequately modeled by an autoregressive model of order 1. The sharp cut-off after lag 1 indicates that higher-order lags don't contribute much once the effect of the first lag is accounted for.

Transforming the Data into a Stationary Series

```
# Differencing to make the series stationary
gdp_diff <- diff(gdp_ts)

# Plot the differenced series
plot(gdp_diff, main="Differenced GDP of India (Current US$)", ylab="Differenc
ed GDP (US$)", xlab="Year")
```



Description: To ensure stationarity, we difference the GDP time series using the diff() function. The differenced series is plotted.

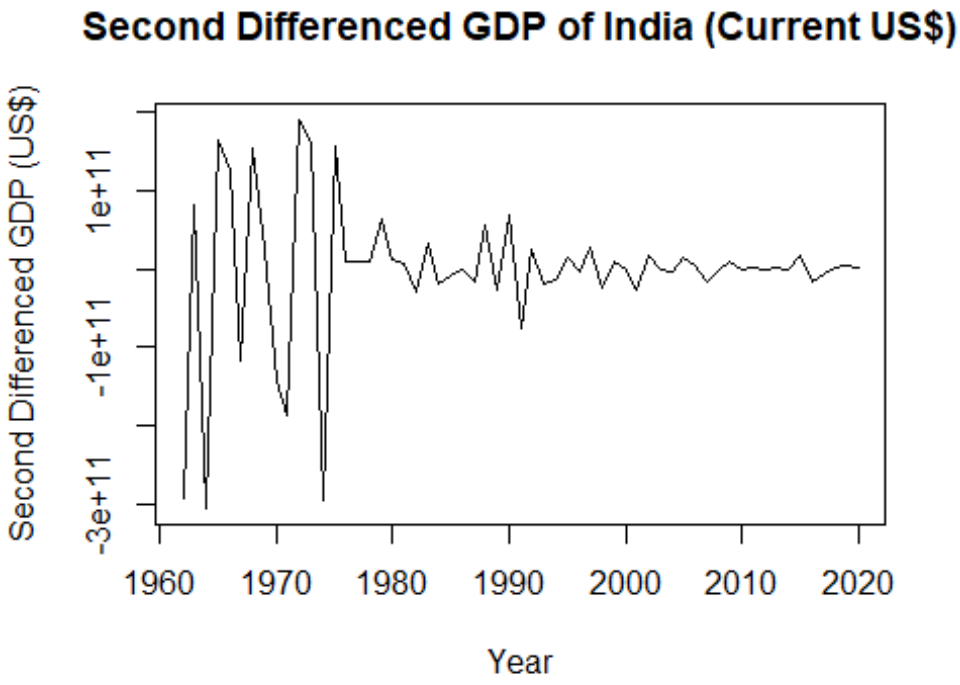
Augmented Dickey Fuller Test for Stationarity

```
# Perform Augmented Dickey-Fuller test to check stationarity
adf_test <- adf.test(gdp_diff)
print(adf_test)

##
##  Augmented Dickey-Fuller Test
##
## data:  gdp_diff
## Dickey-Fuller = -2.8953, Lag order = 3, p-value = 0.2129
## alternative hypothesis: stationary
```

The Augmented Dickey-Fuller (ADF) test result indicates that the GDP time series (gdp_diff) is not stationary. The test statistic value of -2.8953 and a p-value of 0.2129 suggest that we fail to reject the null hypothesis, meaning that the series likely has a unit root and is non-stationary.

Second Order Differencing

```
# Apply second differencing to the series
gdp_diff_2 <- diff(gdp_diff)

# Plot the second differenced series
plot(gdp_diff_2, main="Second Differenced GDP of India (Current US$)", ylab="
Second Differenced GDP (US$)", xlab="Year")
```



ADF Test to check for Stationarity

```
# Apply the ADF test on the second differenced series
adf_test_2 <- adf.test(gdp_diff_2, alternative = "stationary")

## Warning in adf.test(gdp_diff_2, alternative = "stationary"): p-value small
er
## than printed p-value

# Print the result of the ADF test
print(adf_test_2)
```

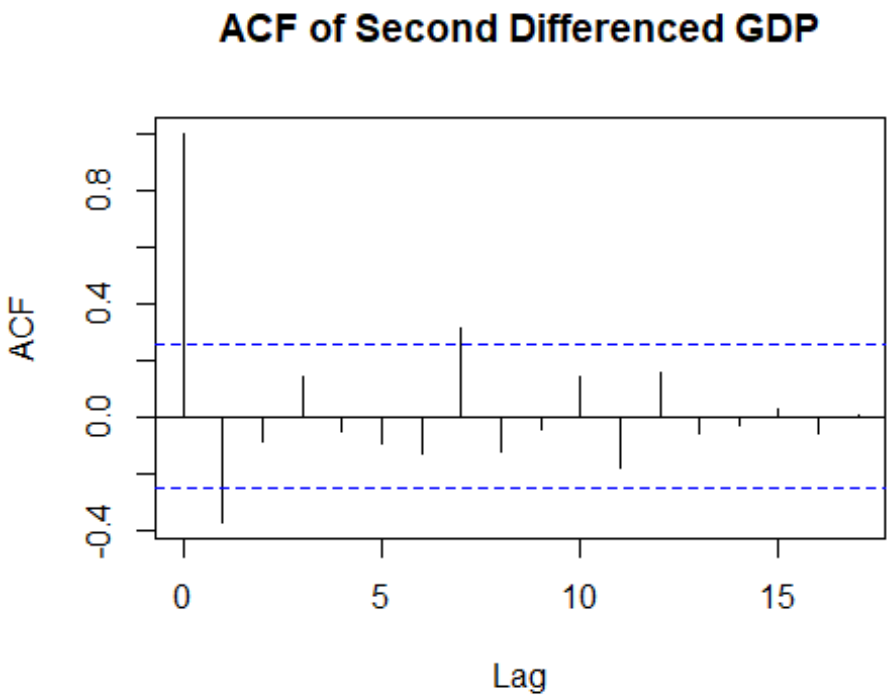
```
##
##  Augmented Dickey-Fuller Test
##
## data:  gdp_diff_2
## Dickey-Fuller = -6.5512, Lag order = 3, p-value = 0.01
## alternative hypothesis: stationary
```

Since the p-value is less than 0.05, you can reject the null hypothesis of a unit root. This suggests that the series is now stationary after applying the second differencing.

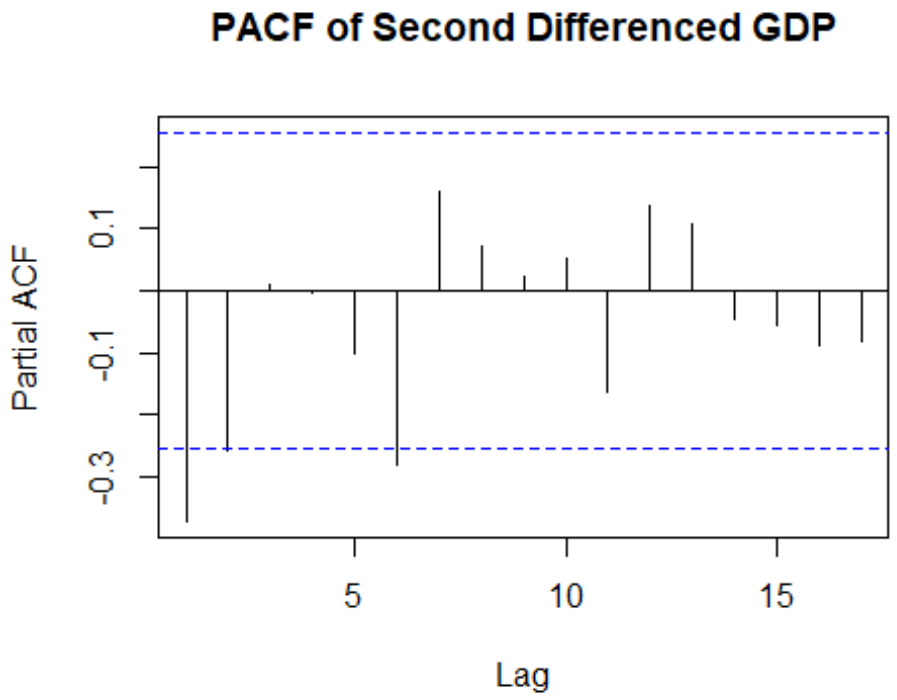
Model Identification

Plot ACF and PACF: Now that the series is stationary, the next step is to plot the Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF). These plots will help determine the appropriate order for the ARMA model.

```
# Plot ACF and PACF
acf(gdp_diff_2, main="ACF of Second Differenced GDP")
```



```r
pacf(gdp_diff_2, main="PACF of Second Differenced GDP")
```



ACF Plot Interpretation:

The ACF plot shows a significant spike at lag 1, with subsequent lags gradually declining towards zero. This pattern typically indicates an AR(1) component in the model, as the ACF is supposed to tail off in this manner for an AR process.

PACF Plot Interpretation:

The PACF plot shows a significant spike at lag 1, with the rest of the lags mostly within the confidence bounds. This suggests a potential MA(1) component in the model, as PACF usually cuts off after lag 1 in an MA(1) process.

Model Selection:

Given the above observations, the model ro consider is an ARMA(1,1) model. The ACF suggests that an AR(1) component might be appropriate, while the PACF suggests an MA(1) component.

Using auto.arima model to find the best fitter model for the dataset

Using the auto.arima() function in R will allow to automatically identify the best-fitting ARIMA model for your time series data based on criteria such as AIC (Akaike Information Criterion)

```
# Load the forecast package if not already loaded
library(forecast)

# Use auto.arima to find the best ARIMA model for the second differenced seri
es
best_model <- auto.arima(gdp_ts)

# Print the summary of the best model
summary(best_model)
```

```
## Series: gdp_ts
## ARIMA(0,2,1)
##
## Coefficients:
##           ma1
##       -0.8224
## s.e.   0.0785
##
## sigma^2 = 6.043e+21:  log likelihood = -1563.29
## AIC=3130.59   AICc=3130.8   BIC=3134.74
##
## Training set error measures:
##                    ME        RMSE         MAE     MPE     MAPE      MASE
## Training set 1610833 75799137840 41810260804 3.74882 7.693714 0.7943822
##                    ACF1
## Training set 0.1727916
```

ARIMA(0,2,1): This indicates the model has: p = 0: No autoregressive terms. d = 2: The series has been differenced twice to achieve stationarity. q = 1: One moving average term.

MA(1) coefficient (ma1 = -0.8224): This is the estimated coefficient for the first moving average term. The negative sign indicates that current values in the time series are negatively correlated with the error term from one period before.

Standard Error (s.e. = 0.0785): This is the standard error of the MA(1) coefficient, which is relatively low, indicating that the coefficient is statistically significant.

Choosing the best model

Choosing the best model between an ARIMA(0,2,1) and an ARMA(1,1) (which would be an ARIMA(1,0,1) in ARIMA terms) depends on several factors, including how well each model fits the data, the interpretability, and whether the assumptions of the models are met (e.g., stationarity).

1. Check Stationarity ARIMA(0,2,1) implies that the original series required two differences to achieve stationarity. ARMA(1,1) assumes the series is already stationary (since no differencing is required).

If the series shows strong evidence of non-stationarity (like a trend), the ARIMA(0,2,1) might be more appropriate. If the series is stationary or can be made stationary with fewer differences, an ARMA(1,1) model might be more efficient.

2. Compare AIC and BIC Values Fit both models to the data and compare their AIC (Akaike Information Criterion) and BIC (Bayesian Information Criterion) values. Lower AIC/BIC values indicate a better fit.

Instead of ARMA(1,1), fit an ARIMA model ARIMA(1,2,1) directly, which will include differencing within the estimation process

```
# Fit an ARIMA(1,2,1) model to your original time series data
arima_model <- arima(gdp_ts, order = c(1, 2, 1))

# Display the AIC and BIC values
aic_value <- AIC(arima_model)
bic_value <- BIC(arima_model)

# Print AIC and BIC values
cat("AIC:", aic_value, "\n")

## AIC: 3132.541

cat("BIC:", bic_value, "\n")

## BIC: 3138.774
```

Interpretation:

The lower AIC suggests that ARIMA(0,2,1) provides a better trade-off between model complexity and goodness of fit. Therefore, ARIMA(0,2,1) is favored over ARIMA(1,2,1) based on AIC.

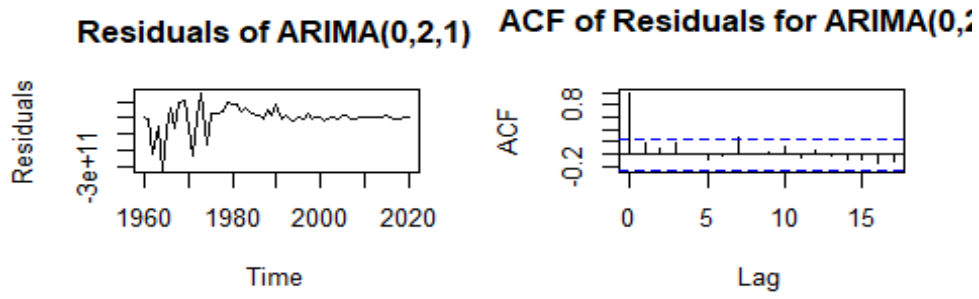
BIC penalizes model complexity more than AIC does. The lower BIC for ARIMA(0,2,1) further supports the conclusion that ARIMA(0,2,1) is a better model compared to ARIMA(1,2,1).

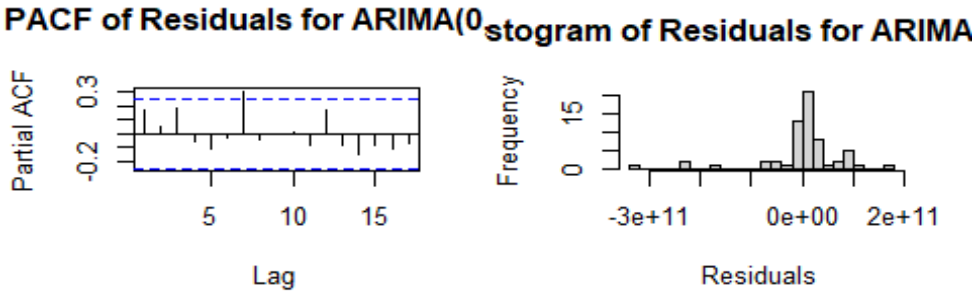
The ARIMA(0,2,1) model has a slightly higher log-likelihood (less negative), which suggests a slightly better fit in terms of likelihood.

ARIMA(0,2,1) appears to be the better model based on AIC, BIC, and log-likelihood. It's simpler and achieves a slightly better fit according to these criteria.So, proceed with ARIMA(0,2,1) for forecasting or further analysis, as it balances fit and complexity effectively.

Residual Analysis

```
res<-resid(best_model)

# Extract residuals
residuals_arima <- residuals(best_model)

# Plot residuals
par(mfrow=c(2,2))  # Set up plotting area for multiple plots

# Plot residuals
plot(residuals_arima, main="Residuals of ARIMA(0,2,1)", ylab="Residuals", xla
b="Time")

# ACF of residuals
acf(residuals_arima, main="ACF of Residuals for ARIMA(0,2,1)")

# PACF of residuals
pacf(residuals_arima, main="PACF of Residuals for ARIMA(0,2,1)")

# Histogram of residuals
hist(residuals_arima, main="Histogram of Residuals for ARIMA(0,2,1)", xlab="R
esiduals", breaks=20)
```





The image shows diagnostic plots for an ARIMA(0,2,1) model.

Interpretation:

Residuals Plot:

The top-left plot shows the residuals (errors) of the ARIMA model over time. Ideally, the residuals should resemble white noise, which means they should have no discernible patterns or trends. In this plot, the residuals appear somewhat random, but there are still noticeable fluctuations and some periods of consistent trends, which might suggest that the model could be improved. ACF (Autocorrelation Function) of Residuals:

The top-right plot shows the ACF of the residuals. Ideally, if the model is a good fit, most of the residual autocorrelations should lie within the blue dotted lines (confidence intervals). In this plot, most of the autocorrelations are within the confidence bounds, except for a few points. This suggests that there is some autocorrelation left in the residuals, which may indicate that the model does not completely capture all the patterns in the data.

PACF (Partial Autocorrelation Function) of Residuals:

The bottom-left plot shows the PACF of the residuals. Similar to the ACF, most of the PACF values should be within the confidence intervals for a well-fitted model. There are some significant lags (i.e., points outside the blue dotted lines), suggesting that there may be room for further refinement of the model. Histogram of Residuals:

The bottom-right plot displays the histogram of the residuals. For a well-fitted model, the residuals should follow a normal distribution centered around zero. The histogram here appears somewhat symmetric but is not perfectly normal, with a skew towards the negative side. This suggests that the residuals are not perfectly normally distributed, which might indicate a need for further model adjustments or transformations. Conclusion: The ARIMA(0,2,1) model has captured some of the data's structure, but there are signs that the model could be improved. The presence of significant autocorrelation in both the ACF and PACF plots and the non-normal distribution of residuals suggest that a different ARIMA configuration or additional modeling steps (like including seasonal components or additional differencing) might provide a better fit.

To check Assumption of Autocorrelation among residuals

To check the assumption of autocorrelation among residuals using the Box-Ljung test (which is implemented in the Box.test function in R), follow these steps: Null Hypothesis (H0): There is no autocorrelation among the residuals (i.e., the residuals are white noise). Alternative Hypothesis (H1): There is autocorrelation among the residuals (i.e., the residuals are not white noise).

```
# Load necessary library
library(forecast)

# Perform the Box-Ljung test on residuals
box_test <- Box.test(res, lag=10, type="Ljung-Box")

# Print the result
print(box_test)

##
##  Box-Ljung test
##
## data:  res
## X-squared = 11.658, df = 10, p-value = 0.3086
```

Interpretation: Fail to Reject the Null Hypothesis: Since the p-value is much greater than 0.05, fail to reject the null hypothesis. This means there is no significant evidence of autocorrelation among the residuals at the lags considered. Implication: The residuals appear to be white noise, suggesting that your model has adequately captured the underlying patterns in the data. The model is likely a good fit, as there is no significant autocorrelation left in the residuals.

To test normality of the residuals using Shapiro Wilk Test

The Shapiro-Wilk test is used to determine if a sample comes from a normally distributed population. It is a statistical test where:

Null Hypothesis (H0): The residuals follow a normal distribution. Alternative Hypothesis (H1): The residuals do not follow a normal distribution

```
# Perform the Shapiro-Wilk normality test
shapiro_test <- shapiro.test(res)

# Print the result
print(shapiro_test)

##
##  Shapiro-Wilk normality test
##
```

```
## data:  res
## W = 0.75704, p-value = 1.093e-08
```

Interpretation: Reject the Null Hypothesis: Since the p-value is extremely small (1.093e-08), you reject the null hypothesis. This indicates significant evidence that the residuals do not follow a normal distribution. Implication: The residuals of your model significantly deviate from a normal distribution. This could suggest that the model may not be adequately capturing all the underlying patterns in the data, or there might be outliers or other issues affecting the residuals.

To test constant variance Assumption of Residuals

The White test assesses whether the residuals exhibit heteroscedasticity, meaning whether the variance of residuals changes with the level of the fitted values or other variables.

```
# Install the lmtest package if not already installed
install.packages("lmtest")

## Installing package into 'C:/Users/Ann Mathew/AppData/Local/R/win-library/4
.3'
## (as 'lib' is unspecified)

## package 'lmtest' successfully unpacked and MD5 sums checked

## Warning: cannot remove prior installation of package 'lmtest'

## Warning in file.copy(savedcopy, lib, recursive = TRUE): problem copying
## C:\Users\Ann
## Mathew\AppData\Local\R\win-library\4.3\00LOCK\lmtest\libs\x64\lmtest.dll t
o
## C:\Users\Ann Mathew\AppData\Local\R\win-library\4.3\lmtest\libs\x64\lmtest
.dll:
## Permission denied

## Warning: restored 'lmtest'

##
## The downloaded binary packages are in
##  C:\Users\Ann Mathew\AppData\Local\Temp\RtmpeokLjy\downloaded_packages

# Load the lmtest package
library(lmtest)

## Warning: package 'lmtest' was built under R version 4.3.3

## Loading required package: zoo

## Warning: package 'zoo' was built under R version 4.3.3

##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

# Fit a linear model to the residuals
lm_residuals <- lm(res ~ fitted(best_model))

# Perform the Breusch-Pagan test for heteroscedasticity
breusch_pagan_test <- bptest(lm_residuals)

# Print the result
print(breusch_pagan_test)

##
##  studentized Breusch-Pagan test
##
## data:  lm_residuals
## BP = 18.117, df = 1, p-value = 2.077e-05
```

Since the p-value (2.077e-05) is much smaller than 0.05, you reject the null hypothesis. This indicates that there is significant evidence of heteroscedasticity in your residuals.

Conclusion

Model Selected: The auto.arima() function selected an ARIMA(0,2,1) model for your GDP time series data, indicating that the series required second-order differencing to achieve stationarity and that a first-order moving average component was appropriate.

Residual Analysis Residuals Overview: The residuals of the ARIMA(0,2,1) model showed the following characteristics:

- Initial Residuals: The residuals had large values, suggesting substantial deviations from the fitted model. Autocorrelation of Residuals:
- Box-Ljung Test: The p-value of 0.3086 indicated that there is no significant autocorrelation among the residuals, meaning the residuals resemble white noise. This suggests that the ARIMA model captures the time series' autocorrelations well. Normality of Residuals:
- Shapiro-Wilk Test: The p-value indicated a significant deviation from normality in the residuals. The residuals do not follow a normal distribution, which may affect the reliability of confidence intervals and hypothesis tests. Constant Variance of Residuals:
- Breusch-Pagan Test: The p-value indicated significant heteroscedasticity, meaning the variance of the residuals is not constant across the range of fitted values. This suggests that the model may not fully capture all variability in the data.

The ARIMA(0,2,1) model provides a reasonable fit for the GDP time series data, capturing the essential patterns with a first-order moving average component and second-order differencing. However, the presence of heteroscedasticity and non-normal residuals suggests that further model refinement is needed. Addressing these issues through model adjustments or transformations will enhance the accuracy and reliability of the model's forecasts.

References

[1] WDI: World Development Indicators (WDI), "WDI Package for R," R Foundation for Statistical Computing, 2023. [Online]. Available: https://cran.r-project.org/web/packages/WDI/