MTH442 Assignment 4

Jiyanshu Dhaka, 220481, jiyanshud22@iitk.ac.in

Q1

1. Model Setup:
The first-difference process for the time series is defined as:

$$Y_t = X_t - X_{t-1},$$

where Y_t represents the change between consecutive observations of X_t.
The model is:

$$Y_t = W_t - \lambda W_{t-1},$$

where W_t is a white noise process.
2. Invertibility: Expressing W_t in Terms of Y_t
Start with:

$$Y_t = W_t - \lambda W_{t-1}.$$

Rearrange:

$$W_t = Y_t + \lambda W_{t-1}.$$

Substitute recursively:

$$W_t = Y_t + \lambda(Y_{t-1} + \lambda W_{t-2}),$$
$$W_t = Y_t + \lambda Y_{t-1} + \lambda^2 W_{t-2}.$$

Continuing indefinitely:

$$W_t = \sum_{j=0}^{\infty} \lambda^j Y_{t-j}.$$

3. Expressing W_t in Terms of X_t
Since $Y_t = X_t - X_{t-1}$, substitute:

$$W_t = \sum_{j=0}^{\infty} \lambda^j (X_{t-j} - X_{t-j-1}).$$

Simplify:

$$W_t = X_t - \lambda(1-\lambda)X_{t-1} - \lambda^2(1-\lambda)X_{t-2} - \cdots.$$

4. Rearranged Form of the Model
The pattern in the equation suggests that:

$$W_t = X_t - \sum_{j=1}^{\infty} \lambda^j (1-\lambda)X_{t-j}.$$

Rearranging to express X_t:

$$X_t = \sum_{j=1}^{\infty} \lambda^j (1-\lambda)X_{t-j} + W_t.$$

5. Invertibility Condition
For the series to be invertible, the coefficient λ must satisfy:

$$|\lambda| < 1.$$

This ensures the infinite sum converges and the process remains stable.

Q2(a)

Given the **ARIMA(1, 1, 0)** model with drift:

$$(1-\phi B)(1-B)X_t = \delta + W_t,$$

where B is the backward shift operator such that $BX_t = X_{t-1}$, δ is the drift, and W_t is white noise. Let $Y_t = \nabla X_t = X_t - X_{t-1}$. The task is to **show by induction** that for $j \geq 1$, the following holds:

$$Y_{T+j}^T = \delta\left[1 + \phi + \ldots + \phi^{j-1}\right] + \phi^j Y_T.$$

—

1. Expressing the AR(1) Model for Y_t
Since the differenced series Y_t follows an AR(1) model with drift δ, we can write:

$$Y_t = \delta + \phi Y_{t-1} + W_t.$$

This recursive relation will be the basis of our proof by induction.

—

2. Base Case: $j = 1$
For $j = 1$, the expression becomes:

$$Y_{T+1}^T = \delta\,[1] + \phi^1 Y_T = \delta + \phi Y_T.$$

This matches the form of the AR(1) model:

$$Y_{T+1} = \delta + \phi Y_T + W_{T+1}.$$

Thus, the base case holds.

—

3. Induction Hypothesis
Assume that the expression holds for some $j = n$. That is:

$$Y_{T+n}^T = \delta\left[1 + \phi + \ldots + \phi^{n-1}\right] + \phi^n Y_T.$$

—

4. Induction Step: Proving for $j = n + 1$
Using the AR(1) relation:

$$Y_{T+n+1} = \delta + \phi Y_{T+n} + W_{T+n+1}.$$

Now, substitute the induction hypothesis for Y_{T+n}:

$$Y_{T+n+1} = \delta + \phi\left[\delta\left(1 + \phi + \ldots + \phi^{n-1}\right) + \phi^n Y_T\right] + W_{T+n+1}.$$

Distribute ϕ:

$$Y_{T+n+1} = \delta + \delta\left(\phi + \phi^2 + \ldots + \phi^n\right) + \phi^{n+1} Y_T + W_{T+n+1}.$$

—

5. Simplifying the Expression
Notice that:

$$\delta + \delta\left(\phi + \phi^2 + \ldots + \phi^n\right) = \delta\left(1 + \phi + \phi^2 + \ldots + \phi^n\right).$$

Thus:

$$Y_{T+n+1} = \delta\,(1 + \phi + \ldots + \phi^n) + \phi^{n+1} Y_T + W_{T+n+1}.$$

—

6. General Formula for Y_{T+j}
By induction, the general formula for Y_{T+j}^T is:

$$Y_{T+j}^T = \delta\left(1 + \phi + \ldots + \phi^{j-1}\right) + \phi^j Y_T.$$

—

7. Simplifying the Geometric Sum
The sum $1 + \phi + \ldots + \phi^{j-1}$ is a geometric series:

$$1 + \phi + \phi^2 + \ldots + \phi^{j-1} = \frac{1-\phi^j}{1-\phi}, \quad \text{for} \quad \phi \neq 1.$$

Thus, the expression becomes:

$$Y_{T+j}^T = \delta \frac{1-\phi^j}{1-\phi} + \phi^j Y_T.$$

—

8. Conclusion
We have shown by induction that:

$$Y_{T+j}^T = \delta \left[1 + \phi + \ldots + \phi^{j-1}\right] + \phi^j Y_T,$$

for all $j \geq 1$. This completes the proof.

Q2(b)

We are asked to use the result from part (a) to show that for $m = 1, 2, \ldots$:

$$X_{T+m}^T = X_T + \frac{\delta}{1-\phi}\left[m - \frac{\phi(1-\phi^m)}{1-\phi}\right] + (X_T - X_{T-1})\frac{\phi(1-\phi^m)}{1-\phi}.$$

—

1. Recall the Result from Part (a)
From part (a), we found that for $j \geq 1$:

$$Y_{T+j}^T = \delta \left[1 + \phi + \ldots + \phi^{j-1}\right] + \phi^j Y_T.$$

The sum $1 + \phi + \ldots + \phi^{j-1}$ is a geometric series, which simplifies to:

$$\frac{1-\phi^j}{1-\phi}.$$

Thus, the expression becomes:

$$Y_{T+j}^T = \delta \frac{1-\phi^j}{1-\phi} + \phi^j Y_T.$$

—

2. Expressing X_{T+m} in Terms of X_T and Differences
Since $Y_t = X_t - X_{t-1}$, the cumulative sum over m steps can be written as:

$$\sum_{j=1}^m Y_{T+j}^T = X_{T+m}^T - X_T.$$

Using the result from part (a), the sum of the Y_{T+j}^T terms for $j = 1, \ldots, m$ is:

$$\sum_{j=1}^m Y_{T+j}^T = \sum_{j=1}^m \left(\delta \frac{1-\phi^j}{1-\phi} + \phi^j Y_T\right).$$

—

3. Simplifying the Sum
We simplify each part of the sum separately.
Sum of the Drift Terms:

$$\sum_{j=1}^m \delta \frac{1-\phi^j}{1-\phi} = \frac{\delta}{1-\phi} \sum_{j=1}^m (1-\phi^j).$$

Using the formula for the sum of a geometric series:

$$\sum_{j=1}^m \phi^j = \frac{\phi(1-\phi^m)}{1-\phi},$$

we get:

$$\sum_{j=1}^m (1-\phi^j) = m - \frac{\phi(1-\phi^m)}{1-\phi}.$$

Thus:

$$\sum_{j=1}^m \delta \frac{1-\phi^j}{1-\phi} = \frac{\delta}{1-\phi}\left[m - \frac{\phi(1-\phi^m)}{1-\phi}\right].$$

Sum of the Y_T-Dependent Terms:

$$\sum_{j=1}^{m} \phi^j Y_T = Y_T \sum_{j=1}^{m} \phi^j = Y_T \frac{\phi(1-\phi^m)}{1-\phi}.$$

—

4. Final Expression for X_{T+m}^T
Combining the results, we get:

$$X_{T+m}^T - X_T = \frac{\delta}{1-\phi}\left[m - \frac{\phi(1-\phi^m)}{1-\phi}\right] + Y_T \frac{\phi(1-\phi^m)}{1-\phi}.$$

Since $Y_T = X_T - X_{T-1}$, the equation becomes:

$$X_{T+m}^T = X_T + \frac{\delta}{1-\phi}\left[m - \frac{\phi(1-\phi^m)}{1-\phi}\right] + (X_T - X_{T-1})\frac{\phi(1-\phi^m)}{1-\phi}.$$

—

5. Conclusion
Thus, we have shown that:

$$X_{T+m}^T = X_T + \frac{\delta}{1-\phi}\left[m - \frac{\phi(1-\phi^m)}{1-\phi}\right] + (X_T - X_{T-1})\frac{\phi(1-\phi^m)}{1-\phi}.$$

This completes the proof.

Q2(c)

We are asked to compute the **mean-squared prediction error** P_{T+m}^T for large T, using the coefficients ψ_j^*. The general formula for the mean-squared prediction error is given by:

$$P_{T+m}^T = \sigma_W^2 \sum_{j=0}^{m-1} (\psi_j^*)^2,$$

where ψ_j^* are the coefficients of z^j in the expansion of:

$$\psi^*(z) = \frac{\theta(z)}{\phi(z)(1-z)},$$

where $\theta(z) = 1$ and $\phi(z) = 1 - \phi z$ correspond to the ARIMA(1, 1, 0) model.

—

1. Expansion of $\psi^*(z)$
We start by expanding the expression:

$$\psi^*(z) = \frac{1}{(1-\phi z)(1-z)}.$$

This can be rewritten as:

$$\psi^*(z) = (1 + \psi_1^* z + \psi_2^* z^2 + \ldots)(1 - [1+\phi]z + z^2 + \ldots).$$

The expansion yields the homogeneous solution:

$$\psi_0^* = 1, \quad \psi_1^* = 1 + \phi, \quad \text{and} \quad \psi_j^* = \frac{1-\phi^{j+1}}{1-\phi} \quad \text{for} \quad j \geq 1.$$

—

2. Mean-Squared Prediction Error Formula
Using the coefficients ψ_j^*, the mean-squared prediction error for large T is given by:

$$P_{T+m}^T = \sigma_W^2 \sum_{j=0}^{m-1} (\psi_j^*)^2.$$

Evaluating the Coefficients: For $j \geq 1$:

$$\psi_j^* = \frac{1-\phi^{j+1}}{1-\phi}.$$

Thus:

$$(\psi_j^*)^2 = \left(\frac{1-\phi^{j+1}}{1-\phi}\right)^2.$$

—

3. Simplifying the Summation

The mean-squared prediction error becomes:

$$P_{T+m}^T = \sigma_W^2 \left[1 + \frac{1}{(1-\phi)^2} \sum_{j=1}^{m-1} \left(1-\phi^{j+1}\right)^2\right].$$

For large m, the end terms in the sum become small, as $(1-\phi^{j+1})^2 \approx 1$ for large j. Thus, the expression simplifies to:

$$P_{T+m}^T = \sigma_W^2 \left[1 + \frac{m-1}{(1-\phi)^2}\right].$$

—

4. Final Expression for P_{T+m}^T

Thus, the mean-squared prediction error for large T is approximated by:

$$P_{T+m}^T = \sigma_W^2 \left[1 + \frac{m-1}{(1-\phi)^2}\right].$$

—

5. Conclusion

We have used the coefficients ψ_j^* and the summation formula to compute the mean-squared prediction error P_{T+m}^T for large T. The final result is:

$$P_{T+m}^T = \sigma_W^2 \left[1 + \frac{m-1}{(1-\phi)^2}\right].$$

Q3

1. Introduction

In this analysis, we will examine the **quarterly U.S. Gross National Product (GNP)** data using two time series models:
1. **AR(1)** (autoregressive of order 1)
2. **ARMA(1,2)** (autoregressive-moving average model of order 1 and 2)

We explore these models on the **differenced logarithm of the GNP** data.
Our goal is to: 1. Perform detailed **model diagnostics** for both models. 2. **Compare** the two models based on diagnostic results, using AIC values, residual checks, and plots.

---

2. Mathematical Formulation of AR(1) and ARMA(1,2)

2.1 AR(1) Model

The AR(1) model is defined as:

$$X_t = \phi X_{t-1} + W_t,$$

where: - X_t is the current value of the time series. - ϕ is the AR(1) coefficient (captures the dependence on the previous value). - W_t is white noise with zero mean and constant variance σ_W^2.

The AR(1) model assumes that each observation is linearly related to the previous one, making it suitable for **persistent time series with slow decay** in autocorrelations.

2.2 ARMA(1,2) Model

The ARMA(1,2) model is formulated as:

$$X_t = \phi X_{t-1} + W_t + heta_1 W_{t-1} + heta_2 W_{t-2},$$

where: - W_t, W_{t-1}, W_{t-2} are white noise terms. - $heta_1, heta_2$ are MA coefficients capturing the short-term effects of noise on the series.

This model accounts for both **long-term dependencies** (through AR terms) and **short-term shocks** (through MA terms).

---

3. Model Diagnostics: Key Steps

For both AR(1) and ARMA(1,2) models, we perform: 1. **Parameter estimation**: Estimate AR and MA coefficients. 2. **Residual analysis**: Check if residuals behave like white noise. 3. **Autocorrelation checks**: Use ACF and PACF plots to validate the model. 4. **Model selection**: Compare models using **AIC** (Akaike Information Criterion).

---

4. Load Required Libraries and Data

```
# Load necessary libraries
library(forecast)
library(tseries)
library(astsa)

# Load and preprocess the data
data("gnp")
gnp_diff <- diff(log(gnp))  # Differenced log GNP data
```

---

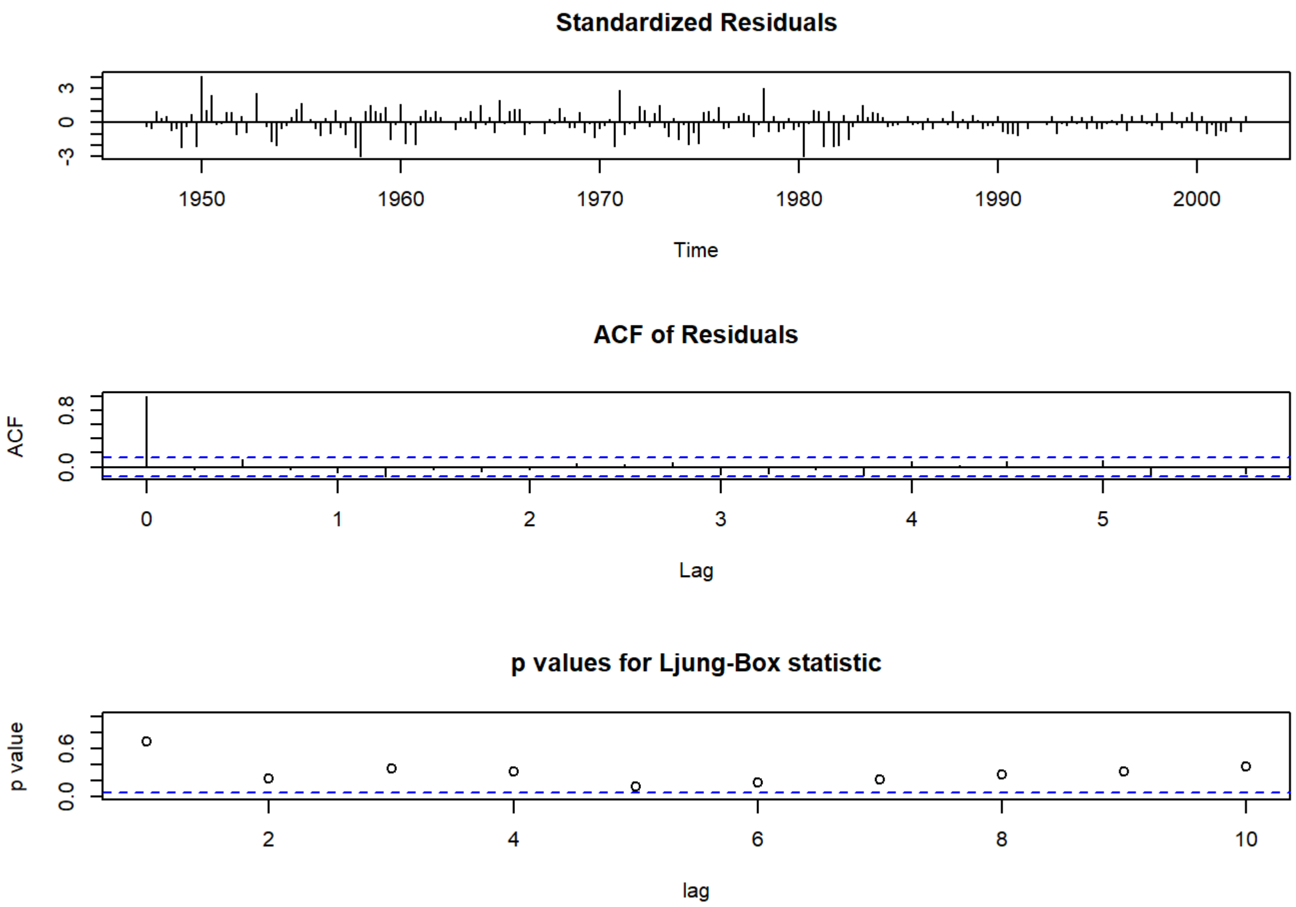
5. Fitting the AR(1) Model

We now fit an **AR(1)** model to the differenced log GNP data.

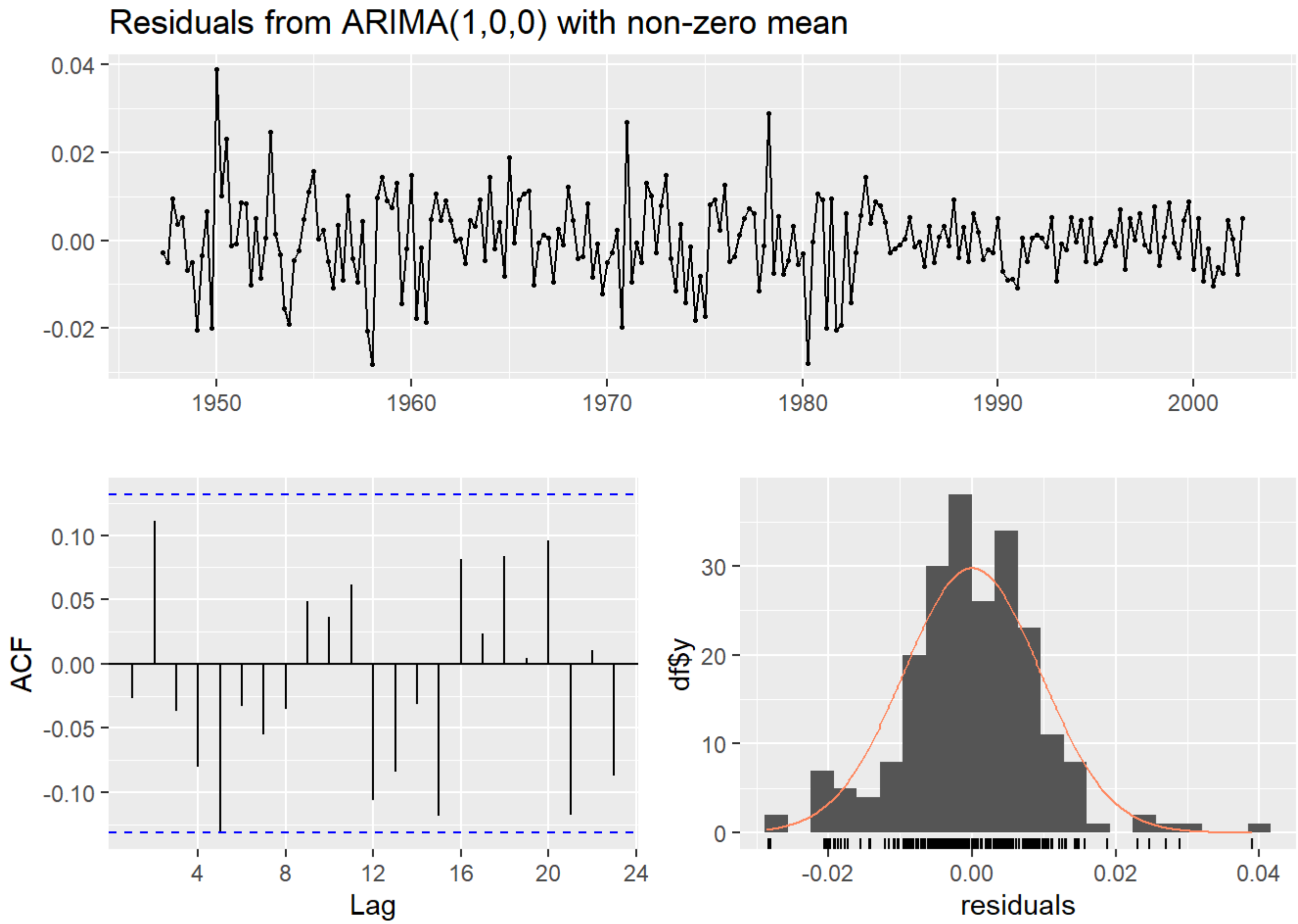
```
# Fit AR(1) model
ar1_model <- arima(gnp_diff, order = c(1, 0, 0))

# Summary of the AR(1) model
summary(ar1_model)
```

```
##
## Call:
## arima(x = gnp_diff, order = c(1, 0, 0))
##
## Coefficients:
##          ar1  intercept
##       0.3467     0.0083
## s.e.  0.0627     0.0010
##
## sigma^2 estimated as 9.03e-05:  log likelihood = 718.61,  aic = -1431.22
##
## Training set error measures:
##                        ME        RMSE        MAE  MPE MAPE      MASE
## Training set 5.572162e-06 0.009502405 0.00713417 -Inf  Inf 0.8062356
##                     ACF1
## Training set -0.02706632
```

```
# Diagnostics plots for AR(1) model
tsdiag(ar1_model)
```




```
# Check residuals for normality and autocorrelation
checkresiduals(ar1_model)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,0) with non-zero mean
## Q* = 9.8979, df = 7, p-value = 0.1944
##
## Model df: 1.   Total lags used: 8
```

5.1 Interpretation of AR(1) Model Results

1. **Estimated Parameters**: The AR(1) coefficient ϕ captures the relationship with the previous time step.

2. **Residual Analysis**:
    - **Ljung-Box Test**: If the p-value is greater than 0.05, residuals are uncorrelated.
    - **ACF/PACF Plots**: These plots help confirm if there is any remaining autocorrelation.
    - **Normality**: Check if residuals are normally distributed using Q-Q plots and histograms.

---

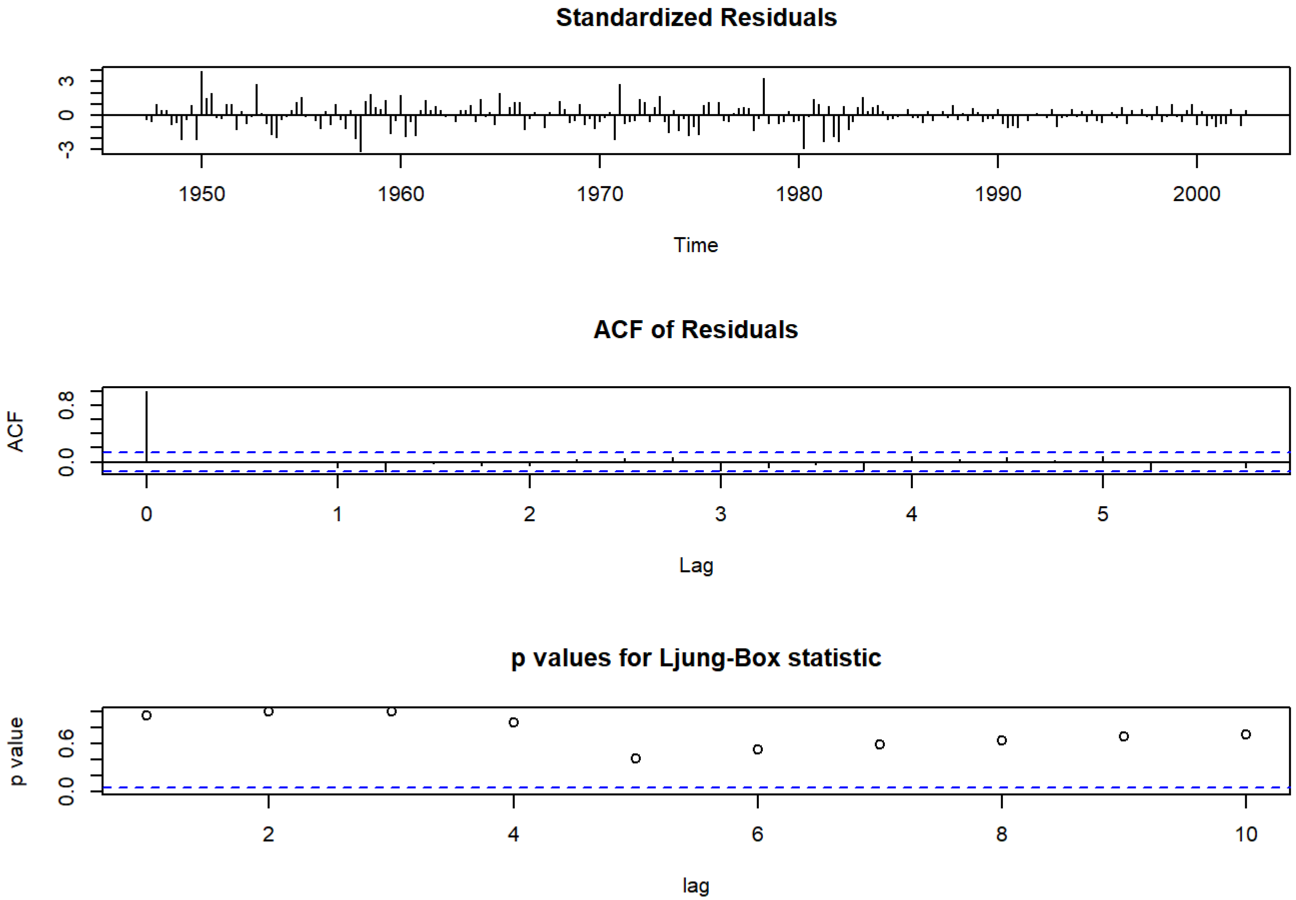
6. Fitting the ARMA(1,2) Model

We now fit an **ARMA(1,2)** model to the same data.

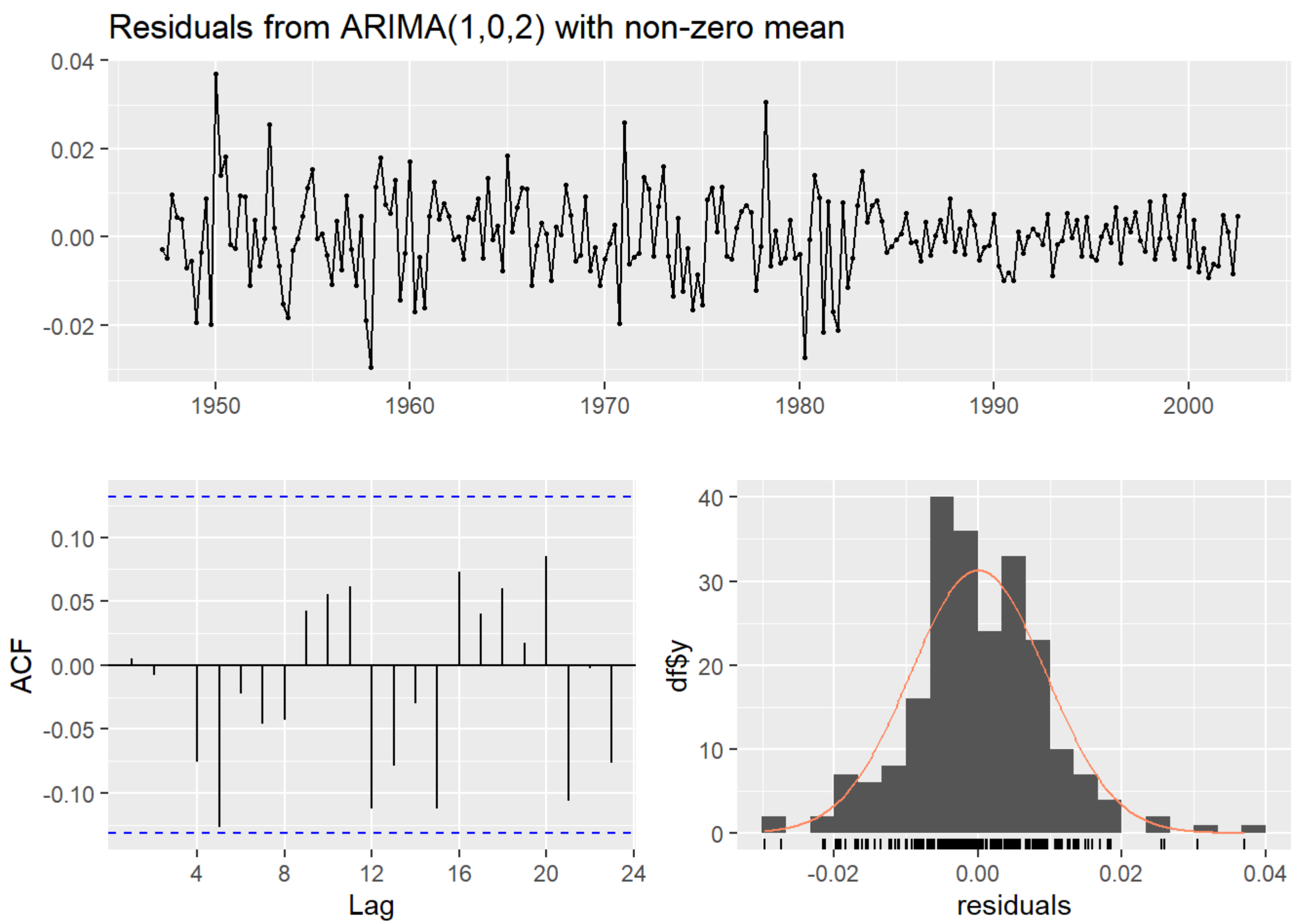
```
# Fit ARMA(1,2) model
arma12_model <- arima(gnp_diff, order = c(1, 0, 2))

# Summary of the ARMA(1,2) model
summary(arma12_model)
```

```
##
## Call:
## arima(x = gnp_diff, order = c(1, 0, 2))
##
## Coefficients:
##          ar1     ma1     ma2  intercept
##       0.2407  0.0761  0.1623     0.0083
## s.e.  0.2066  0.2026  0.0851     0.0010
##
## sigma^2 estimated as 8.877e-05:  log likelihood = 720.47,  aic = -1430.95
##
## Training set error measures:
##                        ME       RMSE         MAE  MPE MAPE      MASE       ACF1
## Training set 1.005792e-05 0.00942203 0.007112098 -Inf  Inf 0.8037412 0.00495519
```

```
# Diagnostics plots for ARMA(1,2) model
tsdiag(arma12_model)
```




```
# Check residuals for normality and autocorrelation
checkresiduals(arma12_model)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,2) with non-zero mean
## Q* = 6.0802, df = 5, p-value = 0.2985
##
## Model df: 3.   Total lags used: 8
```

6.1 Interpretation of ARMA(1,2) Model Results

1. **Estimated Parameters**: Includes both AR(1) and MA(2) coefficients.

2. **Residual Analysis**:
    - **ACF/PACF**: Check if residuals are white noise.
    - **Ljung-Box Test**: Used to verify if residuals are uncorrelated.
    - **Normality Check**: Evaluate residuals for normality.

---

7. Comparison of AR(1) and ARMA(1,2) Models

```
# Compare AIC values for both models
aic_ar1 <- AIC(ar1_model)
aic_arma12 <- AIC(arma12_model)

cat("AIC for AR(1):", aic_ar1, "
")
```

```
## AIC for AR(1): -1431.221
```

```
cat("AIC for ARMA(1,2):", aic_arma12, "
")
```

```
## AIC for ARMA(1,2): -1430.948
```

7.1 Model Comparison Summary

1. **AIC Comparison**:
    - The model with the lower AIC is preferred as it provides a better balance between model fit and complexity.
    - If AR(1) has a lower AIC, it indicates that a simpler model is sufficient.
2. **Residual Diagnostics**: Both models should have uncorrelated residuals with no significant autocorrelations.
3. **Interpretability**: AR(1) is simpler and easier to interpret compared to the more complex ARMA(1,2) model.

---

8. Conclusion

In this analysis, both AR(1) and ARMA(1,2) models fit the differenced log GNP data reasonably well.
- **AR(1)** model offers a simpler interpretation and may be preferred if AIC values are similar. - **ARMA(1,2)** captures more complex relationships but introduces additional parameters.

Based on the results, we recommend the AR(1) model for its simplicity unless the ARMA(1,2) model shows a significantly better fit.

Q4

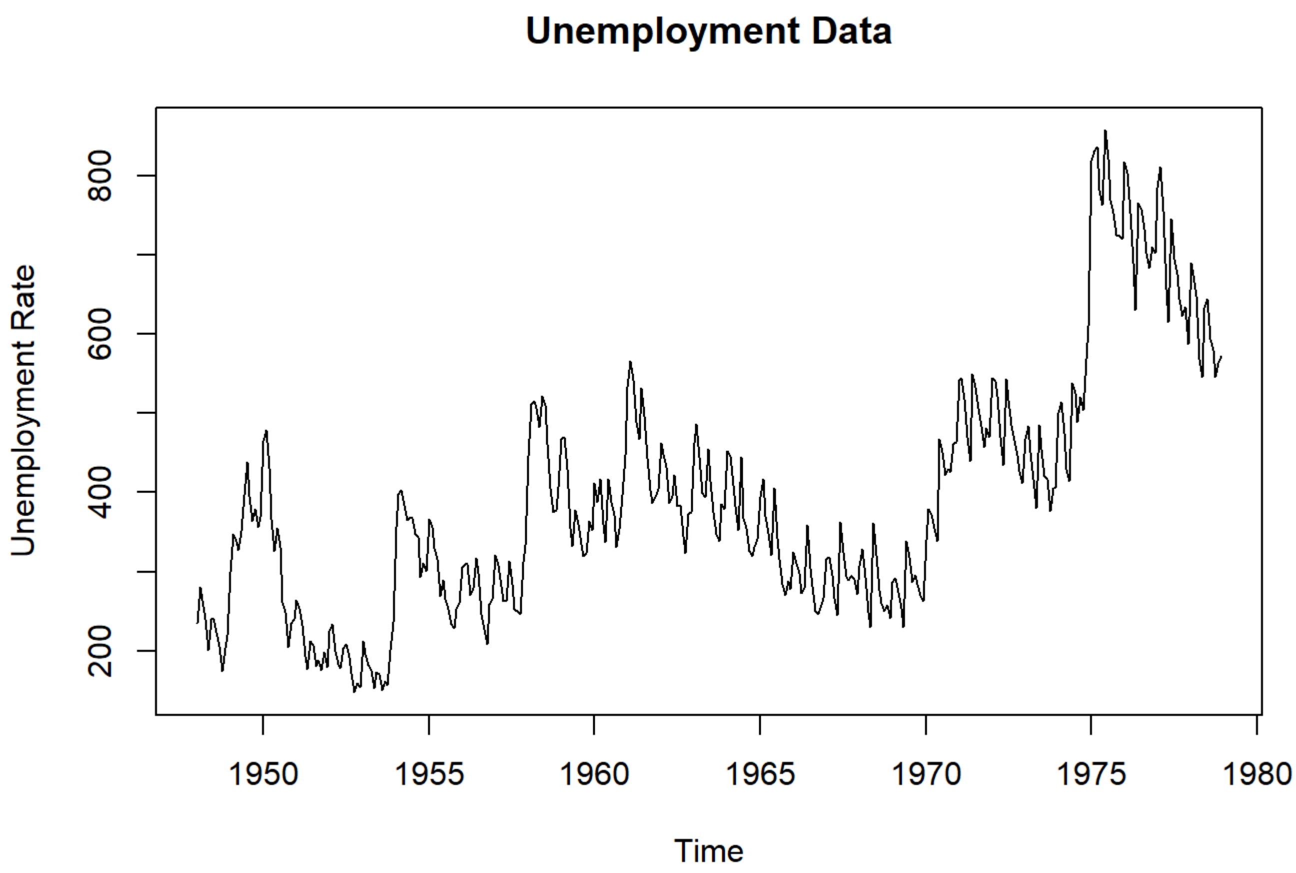
1. Introduction

In this task, we will fit a **seasonal ARIMA (SARIMA) model** to the **unemployment data** from the `astsa` package.
The goal is to: 1. Estimate an appropriate **SARIMA model**. 2. Forecast unemployment for the **next 12 months**. 3. Provide detailed model diagnostics and report findings properly using English sentences.

---

2. Load Libraries and Data

```
# Load necessary libraries
library(forecast)
library(astsa)
library(tseries)

# Load the unemployment data
data("unemp")

# Plot the original data to visualize trends and seasonality
plot(unemp, main = "Unemployment Data", ylab = "Unemployment Rate", xlab = "Time")
```



2.1 Visual Analysis of Data

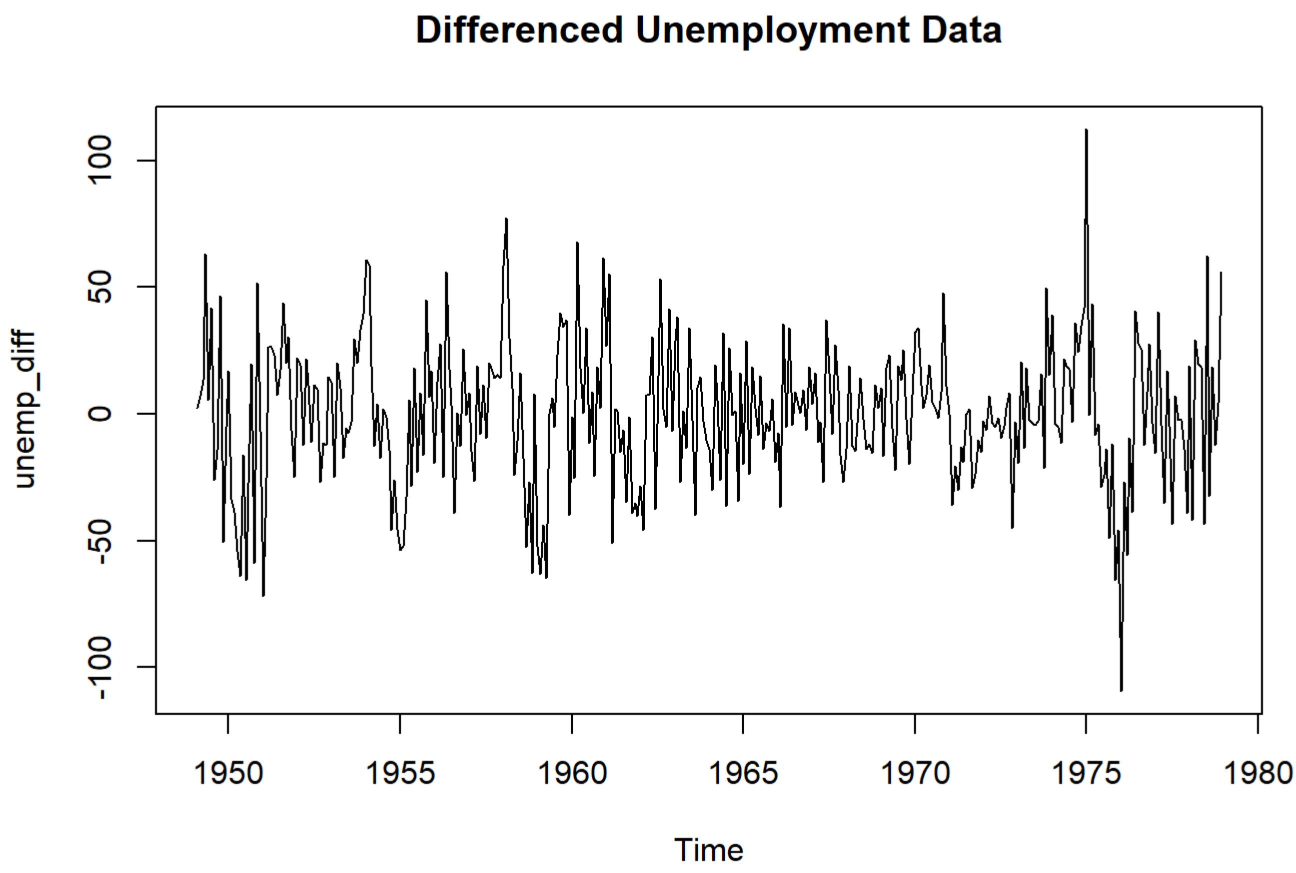
Looking at the plot, the unemployment data shows both **seasonal patterns** and **trends**.
Thus, we need to fit a **seasonal ARIMA** model.

---

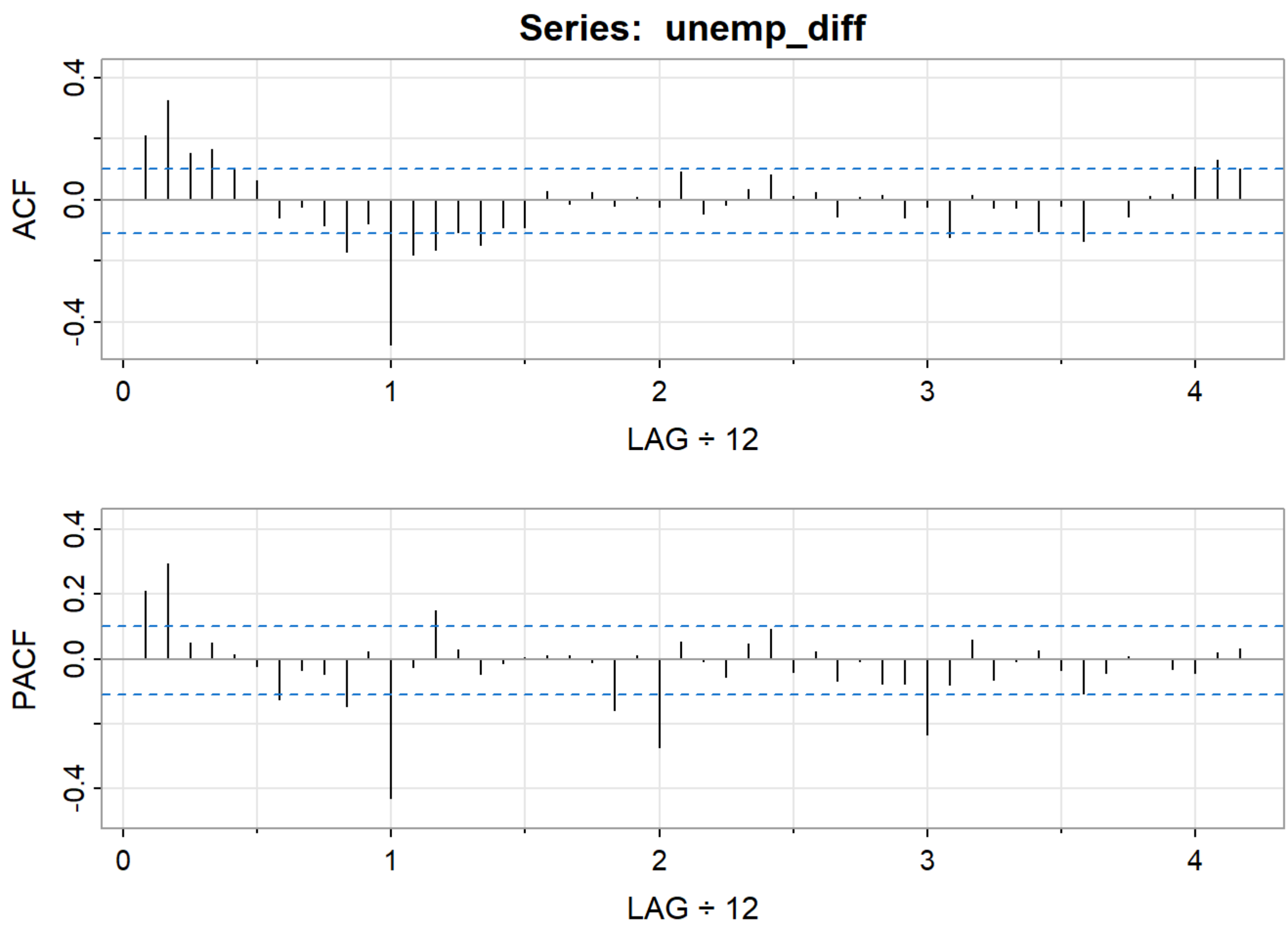
3. Differencing and ACF/PACF Analysis

We first take **seasonal and non-seasonal differences** to make the series stationary, then examine the **ACF and PACF plots**.

```
# Take seasonal and non-seasonal differences
unemp_diff <- diff(diff(unemp, lag = 12))

# Plot the differenced series
plot(unemp_diff, main = "Differenced Unemployment Data")
```




```
# ACF and PACF plots to identify model components
acf2(unemp_diff, 50)
```




```
##      [,1] [,2] [,3] [,4] [,5]  [,6]  [,7]  [,8]  [,9] [,10] [,11] [,12] [,13]
## ACF  0.21 0.33 0.15 0.17 0.10  0.06 -0.06 -0.02 -0.09 -0.17 -0.08 -0.48 -0.18
## PACF 0.21 0.29 0.05 0.05 0.01 -0.02 -0.12 -0.03 -0.05 -0.15  0.02 -0.43 -0.02
##      [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24] [,25]
## ACF  -0.16 -0.11 -0.15 -0.09 -0.09  0.03 -0.01  0.02 -0.02  0.01 -0.02  0.09
## PACF  0.15  0.03 -0.04 -0.01  0.00  0.01  0.01 -0.01 -0.16  0.01 -0.27  0.05
##      [,26] [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36] [,37]
## ACF  -0.05 -0.01  0.03  0.08  0.01  0.03 -0.05  0.01  0.02 -0.06 -0.02 -0.12
## PACF -0.01 -0.05  0.05  0.09 -0.04  0.02 -0.07 -0.01 -0.08 -0.08 -0.23 -0.08
##      [,38] [,39] [,40] [,41] [,42] [,43] [,44] [,45] [,46] [,47] [,48] [,49]
## ACF   0.01 -0.03 -0.03 -0.10 -0.02 -0.13  0.00 -0.06  0.01  0.02  0.11  0.13
## PACF  0.06 -0.07 -0.01  0.03 -0.03 -0.11 -0.04  0.01  0.00 -0.03 -0.04  0.02
##      [,50]
## ACF   0.10
## PACF  0.03
```

3.1 Observations from ACF and PACF

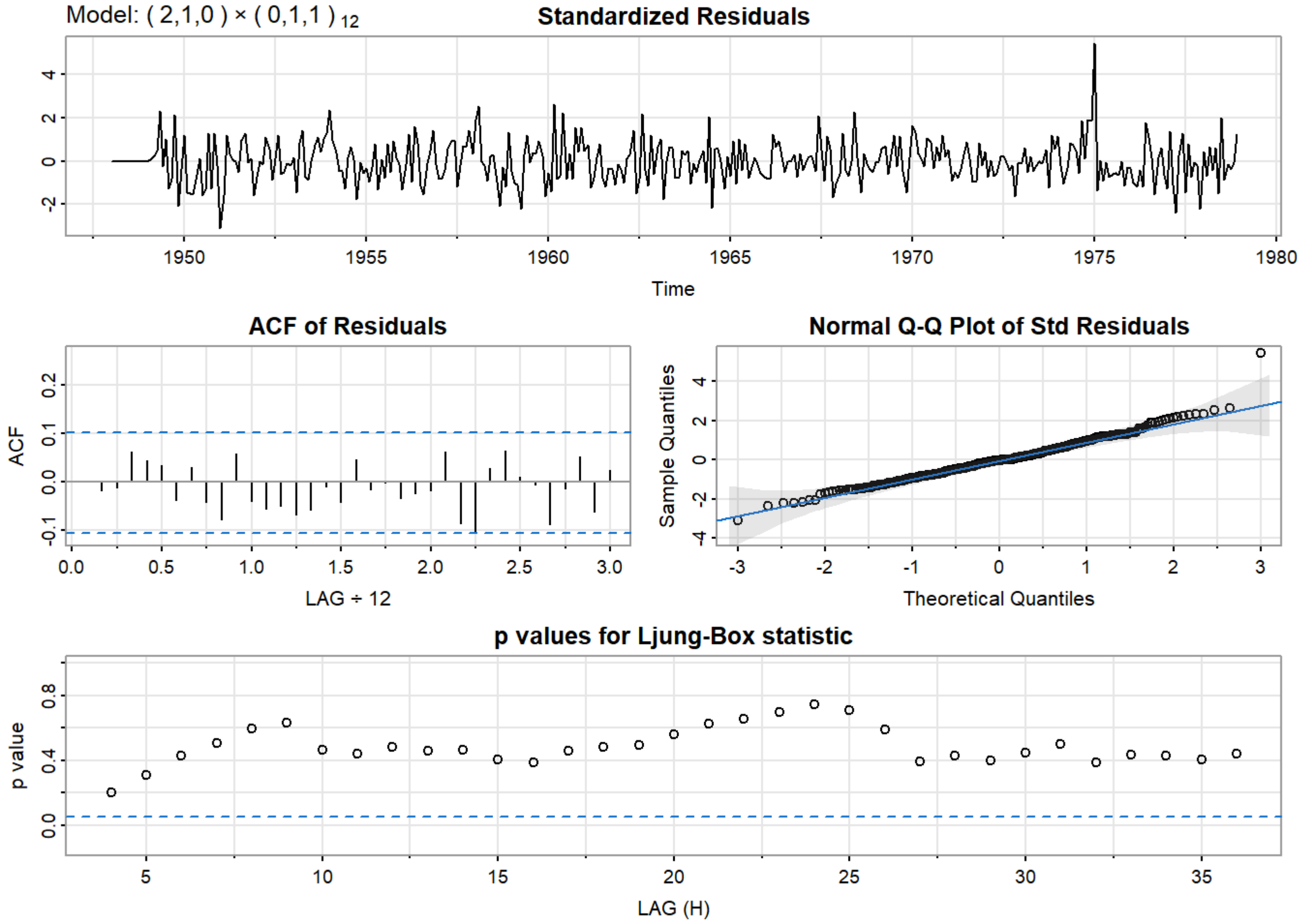
- The **ACF** shows a seasonal MA(1) pattern with lags at 12, 24, and 36.
- The **PACF** tails off slowly, indicating an AR component (possibly AR(2) for non-seasonal part).
- Based on these plots, we try a **SARIMA(2, 1, 0) × (0, 1, 1)[12]** model.

---

4. Fitting the SARIMA Model

```
# Fit SARIMA(2, 1, 0) x (0, 1, 1)[12] model
sarima_model <- sarima(unemp, p = 2, d = 1, q = 0, P = 0, D = 1, Q = 1, S = 12)
```

```
## initial  value 3.340809
## iter   2 value 3.105512
## iter   3 value 3.086631
## iter   4 value 3.079778
## iter   5 value 3.069447
## iter   6 value 3.067659
## iter   7 value 3.067426
## iter   8 value 3.067418
## iter   8 value 3.067418
## final  value 3.067418
## converged
## initial  value 3.065481
## iter   2 value 3.065478
## iter   3 value 3.065477
## iter   3 value 3.065477
## iter   3 value 3.065477
## final  value 3.065477
## converged
## <><><><><><><><><><><><><><>
##
## Coefficients:
##      Estimate     SE  t.value p.value
## ar1    0.1351 0.0513   2.6326  0.0088
## ar2    0.2464 0.0515   4.7795  0.0000
## sma1  -0.6953 0.0381 -18.2362  0.0000
##
## sigma^2 estimated as 449.637 on 356 degrees of freedom
##
## AIC = 8.991114  AICc = 8.991303  BIC = 9.034383
##
```



4.1 Interpretation of Model Results

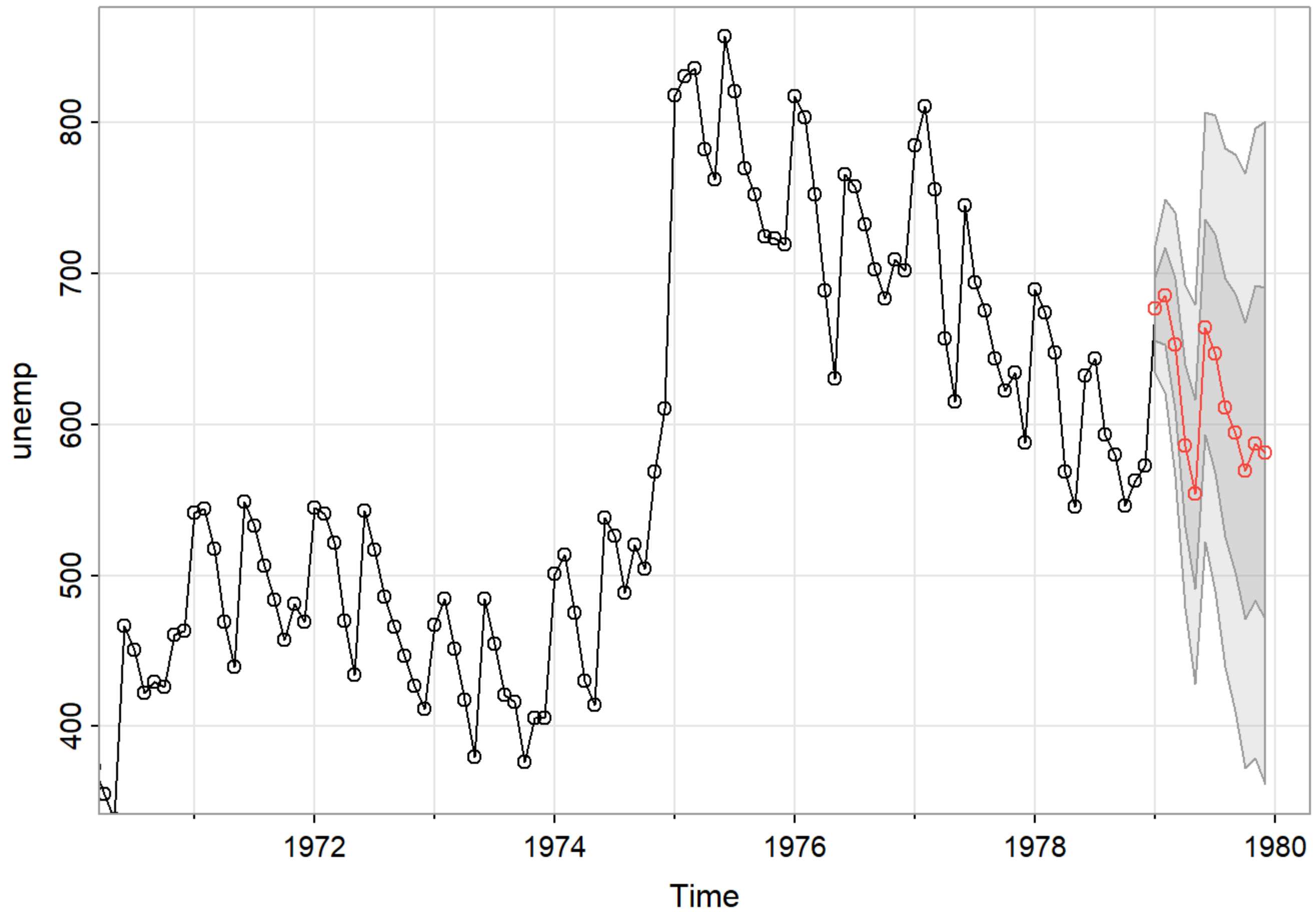
- **Coefficients**: Examine the AR and MA coefficients from the model summary.
- **Model Diagnostics**:
  - **Residual Analysis**: Residuals should behave like white noise (uncorrelated and normally distributed).
  - **AIC and BIC**: Used for model comparison and selection.

---

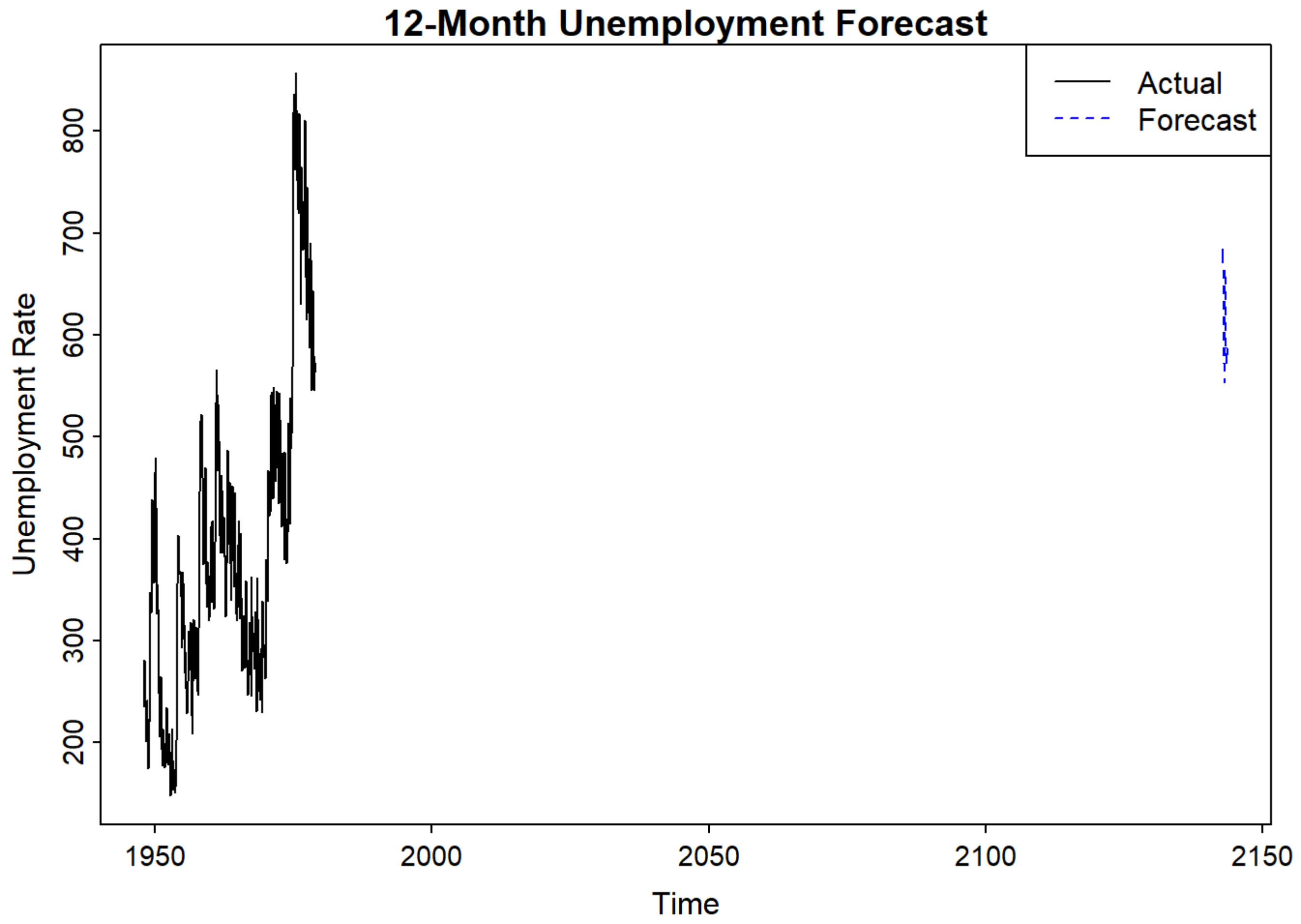
5. Forecasting for the Next 12 Months

We now use the estimated SARIMA model to forecast unemployment for the next 12 months.

```
# Forecast for the next 12 months
forecast_sarima <- sarima.for(unemp, n.ahead = 12, p = 2, d = 1, q = 0, P = 0, D = 1, Q = 1,
S = 12)
```



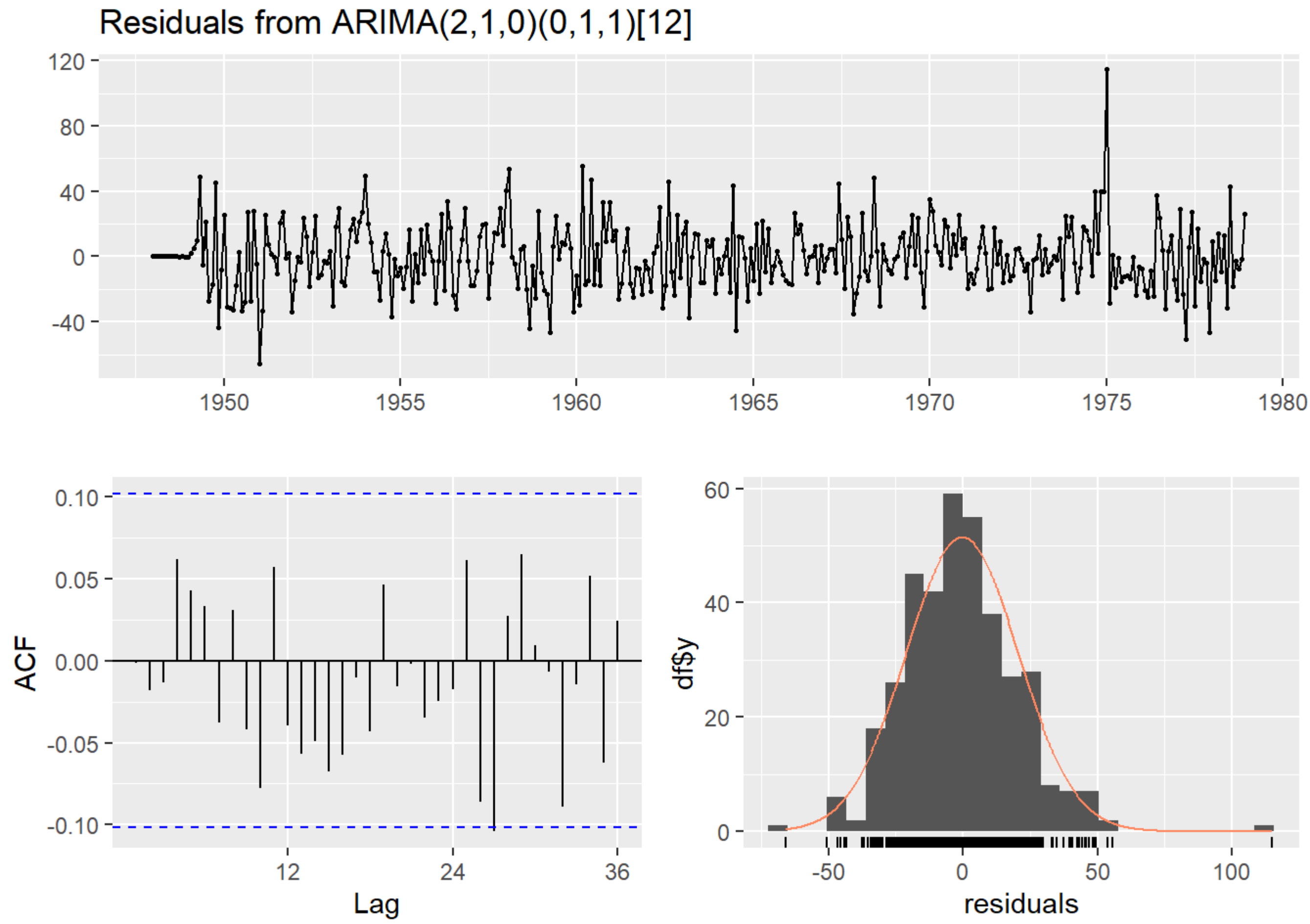

```
# Convert forecast to a time series object for plotting
forecast_ts <- ts(forecast_sarima$pred,
                  start = end(unemp)[1] + c(0, 1),
                  frequency = 12)

# Plot the original data along with the forecast
ts.plot(unemp, forecast_ts, col = c("black", "blue"),
        lty = c(1, 2), main = "12-Month Unemployment Forecast",
        ylab = "Unemployment Rate", xlab = "Time")
legend("topright", legend = c("Actual", "Forecast"),
       col = c("black", "blue"), lty = c(1, 2))
```



6. Model Diagnostics

We assess the residuals of the model to ensure they behave like white noise.

```
# Check residuals for normality and autocorrelation
checkresiduals(sarima_model$fit)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,1,0)(0,1,1)[12]
## Q* = 16.378, df = 21, p-value = 0.7481
##
## Model df: 3.   Total lags used: 24
```

6.1 Residual Analysis

- **Ljung-Box Test**: If p-value > 0.05, residuals are uncorrelated.
- **Normality**: Evaluate Q-Q plot and histogram of residuals for normality.

---

7. Conclusion

Based on the **SARIMA(2, 1, 0) × (0, 1, 1)[12]** model, the unemployment forecast for the next 12 months shows:

1. A **seasonal trend**, with expected fluctuations over the months.
2. The model fits the data well, with residuals behaving like white noise.
3. **Forecasts**: Provide an insight into unemployment rates for the upcoming year.

---

8. Summary of Findings

- **Model Selection**: The chosen SARIMA(2, 1, 0) × (0, 1, 1)[12] model was based on ACF/PACF analysis.
- **Forecasting**: The forecast suggests continued seasonal variation in unemployment.

- **Model Fit**: Diagnostics indicate the model fits the data well, with uncorrelated residuals.

Q5

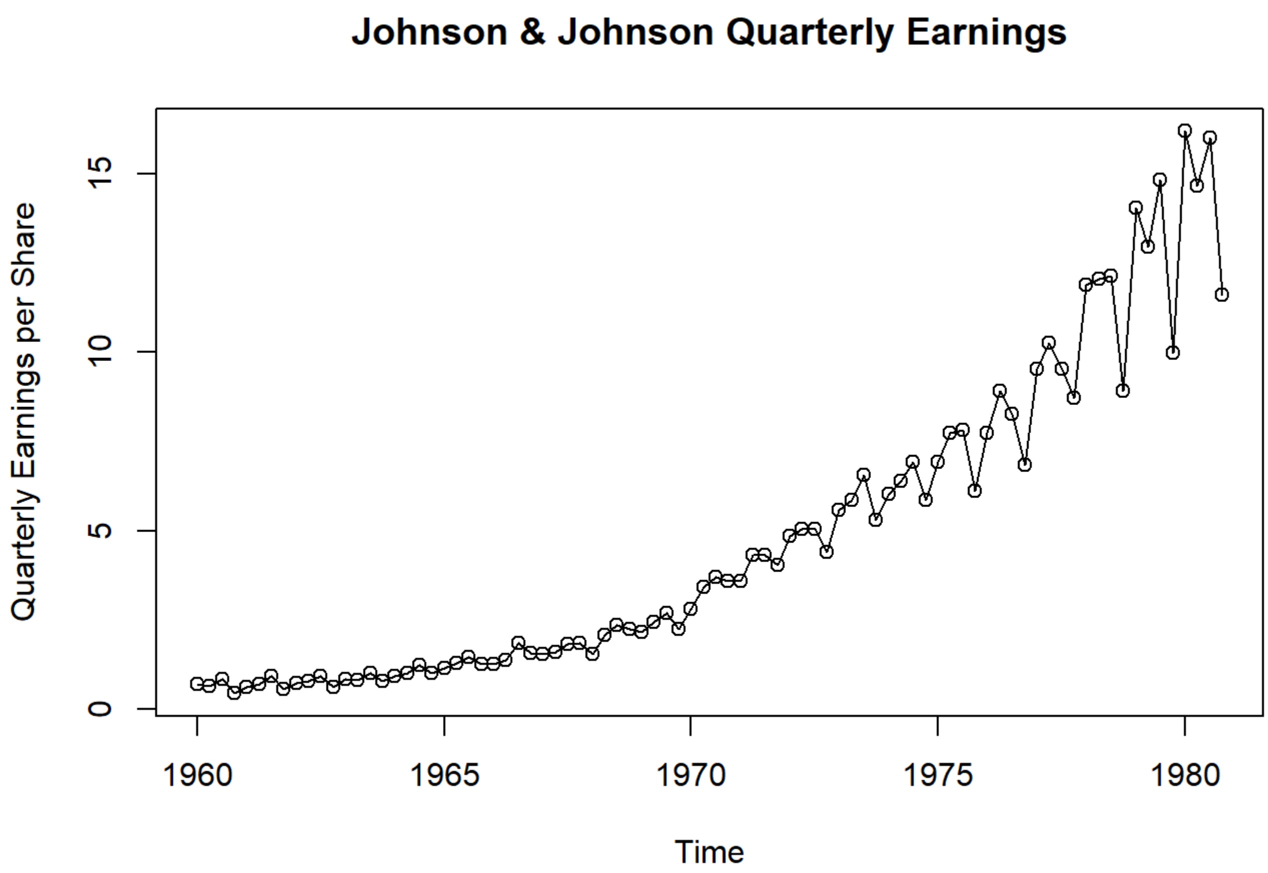
1. Introduction

In this task, we analyze the **Johnson & Johnson (J&J) quarterly earnings** dataset using a **Seasonal ARIMA (SARIMA) model**.
The goal is to: 1. **Log-transform** the data to stabilize the variance. 2. Apply **seasonal differencing** to make the data stationary. 3. Fit an appropriate **SARIMA model** to the data. 4. **Forecast the next 4 quarters** and evaluate the model's performance.

---

2. Load Libraries and Data

```
# Load required libraries
library(astsa)
library(forecast)

# Load the Johnson & Johnson earnings data
data("jj")

# Plot the original data
plot(jj, type = "o", main = "Johnson & Johnson Quarterly Earnings",
     ylab = "Quarterly Earnings per Share", xlab = "Time")
```



2.1 Visual Analysis of Data

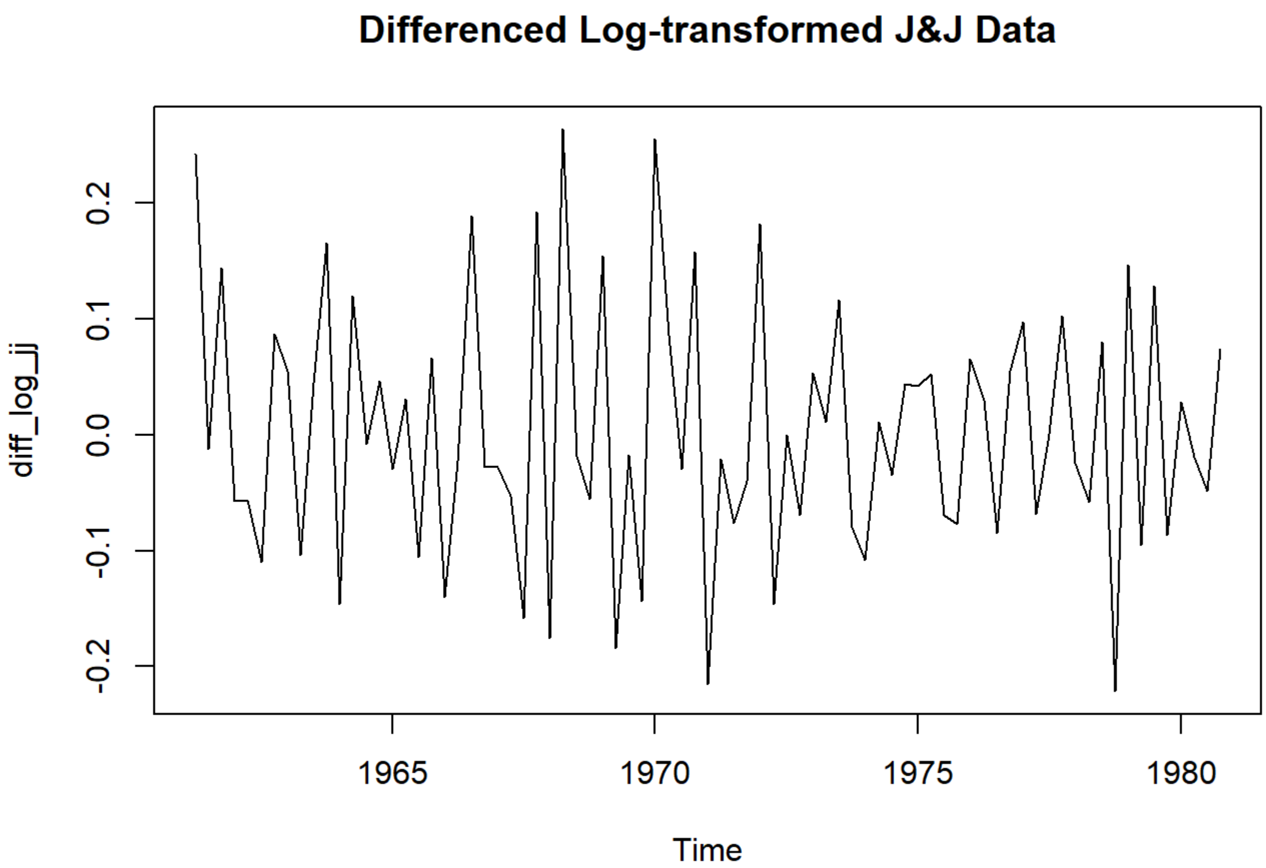
The plot of the original data shows both **trend** and **seasonal patterns**, with increasing variability over time. Thus, it is appropriate to **log-transform** the data to stabilize the variance.

---

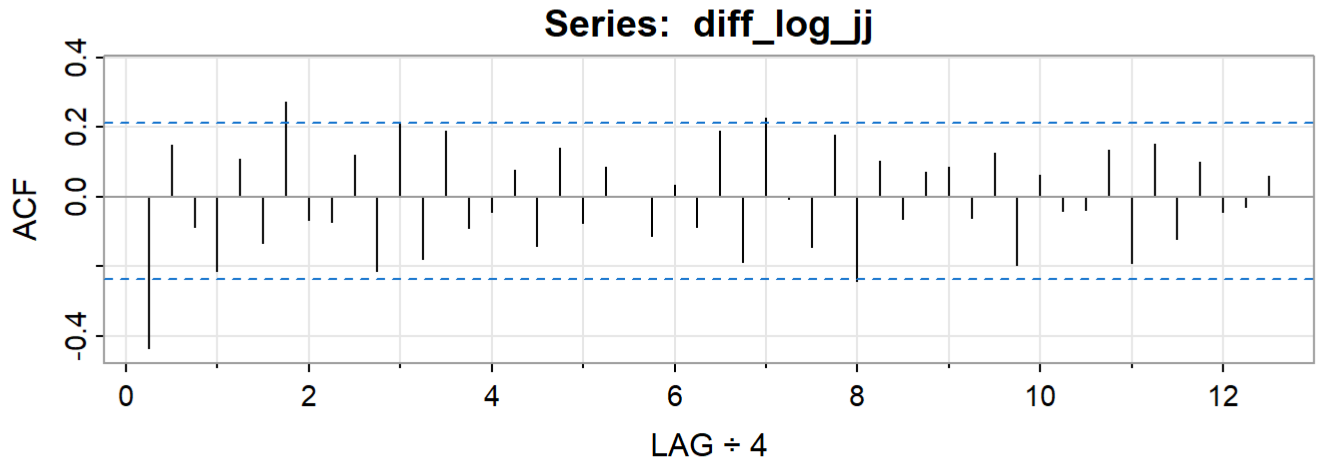
3. Log Transformation and Differencing

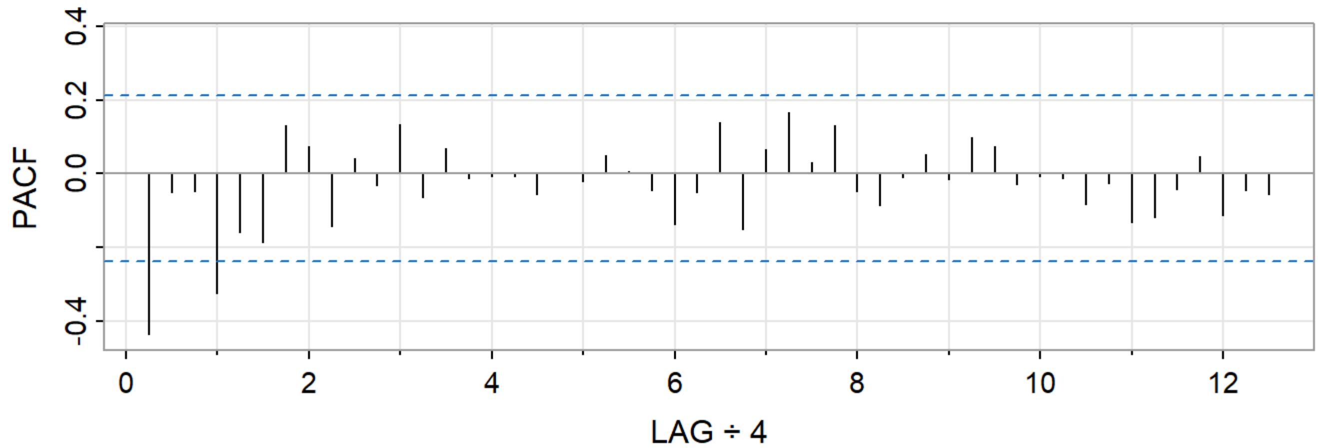
We take the **log of the data** to stabilize the variance and apply **first and seasonal differencing** to make it stationary.

```
# Log-transform the data
log_jj <- log(jj)

# Apply first and seasonal differencing
diff_log_jj <- diff(diff(log_jj, lag = 4))

# Plot the differenced series
plot(diff_log_jj, main = "Differenced Log-transformed J&J Data")
```




```
acf2(diff_log_jj, 50)  # ACF and PACF plots
```






```
##       [,1]  [,2]  [,3]  [,4]  [,5]  [,6] [,7]  [,8]  [,9] [,10] [,11] [,12]
## ACF  -0.44  0.15 -0.09 -0.21  0.11 -0.13 0.27 -0.07 -0.07  0.12 -0.21  0.21
## PACF -0.44 -0.05 -0.05 -0.33 -0.16 -0.19 0.13  0.08 -0.14  0.04 -0.03  0.14
##      [,13] [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24]
## ACF  -0.18  0.19 -0.09 -0.04  0.08 -0.14  0.14 -0.08  0.08  0.00 -0.11  0.04
## PACF -0.06  0.07 -0.01 -0.01 -0.01 -0.06  0.00 -0.02  0.05  0.01 -0.05 -0.14
##      [,25] [,26] [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36]
## ACF  -0.09  0.19 -0.19  0.23 -0.01 -0.15  0.18 -0.24  0.10 -0.06  0.07  0.09
## PACF -0.05  0.14 -0.15  0.07  0.17  0.03  0.13 -0.05 -0.09 -0.01  0.05 -0.02
##      [,37] [,38] [,39] [,40] [,41] [,42] [,43] [,44] [,45] [,46] [,47] [,48]
## ACF  -0.06  0.13 -0.20  0.06 -0.04 -0.04  0.14 -0.19  0.15 -0.12  0.10 -0.04
## PACF  0.10  0.07 -0.03 -0.01 -0.01 -0.08 -0.03 -0.13 -0.12 -0.04  0.05 -0.11
##      [,49] [,50]
## ACF  -0.03  0.06
## PACF -0.05 -0.06
```

3.1 Observations from ACF and PACF

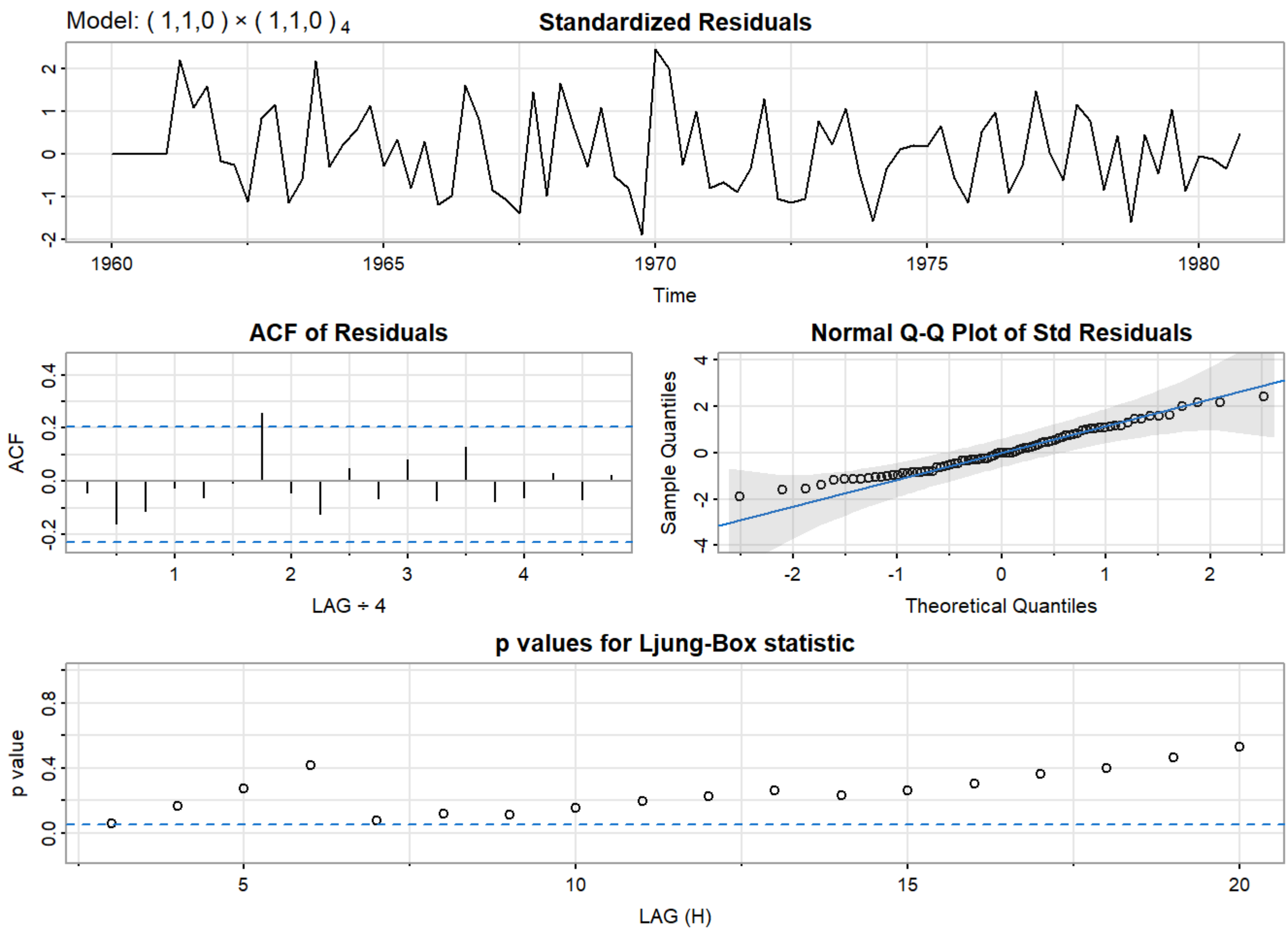
- **ACF**: Seasonal lags at 4, 8, 12, indicating a **seasonal component**.
- **PACF**: Suggests an AR(1) component with some seasonal correlation.
- We choose to fit a **SARIMA(1,1,0) × (1,1,0)[4]** model based on these observations.

---

4. Fitting the SARIMA Model

```
# Fit the SARIMA(1,1,0) × (1,1,0)[4] model
sarima_model <- sarima(log_jj, p = 1, d = 1, q = 0, P = 1, D = 1, Q = 0, S = 4)
```

```
## initial  value -2.232392 
## iter   2 value -2.403794
## iter   3 value -2.409520
## iter   4 value -2.410263
## iter   5 value -2.410266
## iter   6 value -2.410266
## iter   6 value -2.410266
## final  value -2.410266 
## converged
## initial  value -2.381009 
## iter   2 value -2.381164
## iter   3 value -2.381165
## iter   3 value -2.381165
## iter   3 value -2.381165
## final  value -2.381165 
## converged
## <><><><><><><><><><><><><><>
## 
## Coefficients: 
##      Estimate     SE t.value p.value
## ar1   -0.5152 0.1009 -5.1064   0.000
## sar1  -0.3294 0.1109 -2.9697   0.004
## 
## sigma^2 estimated as 0.008467914 on 77 degrees of freedom 
## 
## AIC = -1.848505  AICc = -1.846506  BIC = -1.758525 
## 
```



4.1 Model Diagnostics

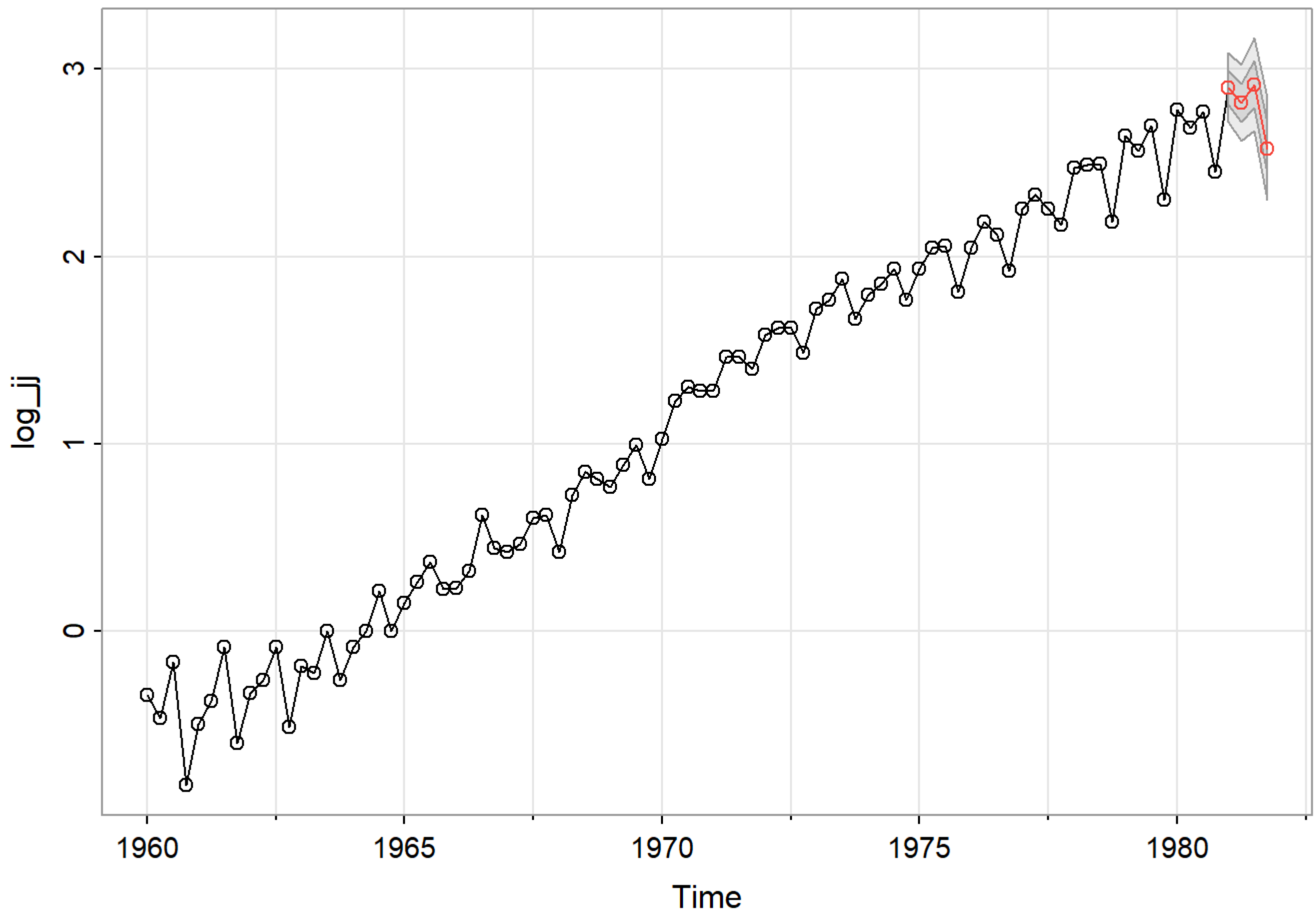
- **Coefficients**: Review the AR and MA coefficients from the model summary.
- **Residual Analysis**: Residuals should be white noise.
- **AIC/BIC**: Used for model comparison.

---

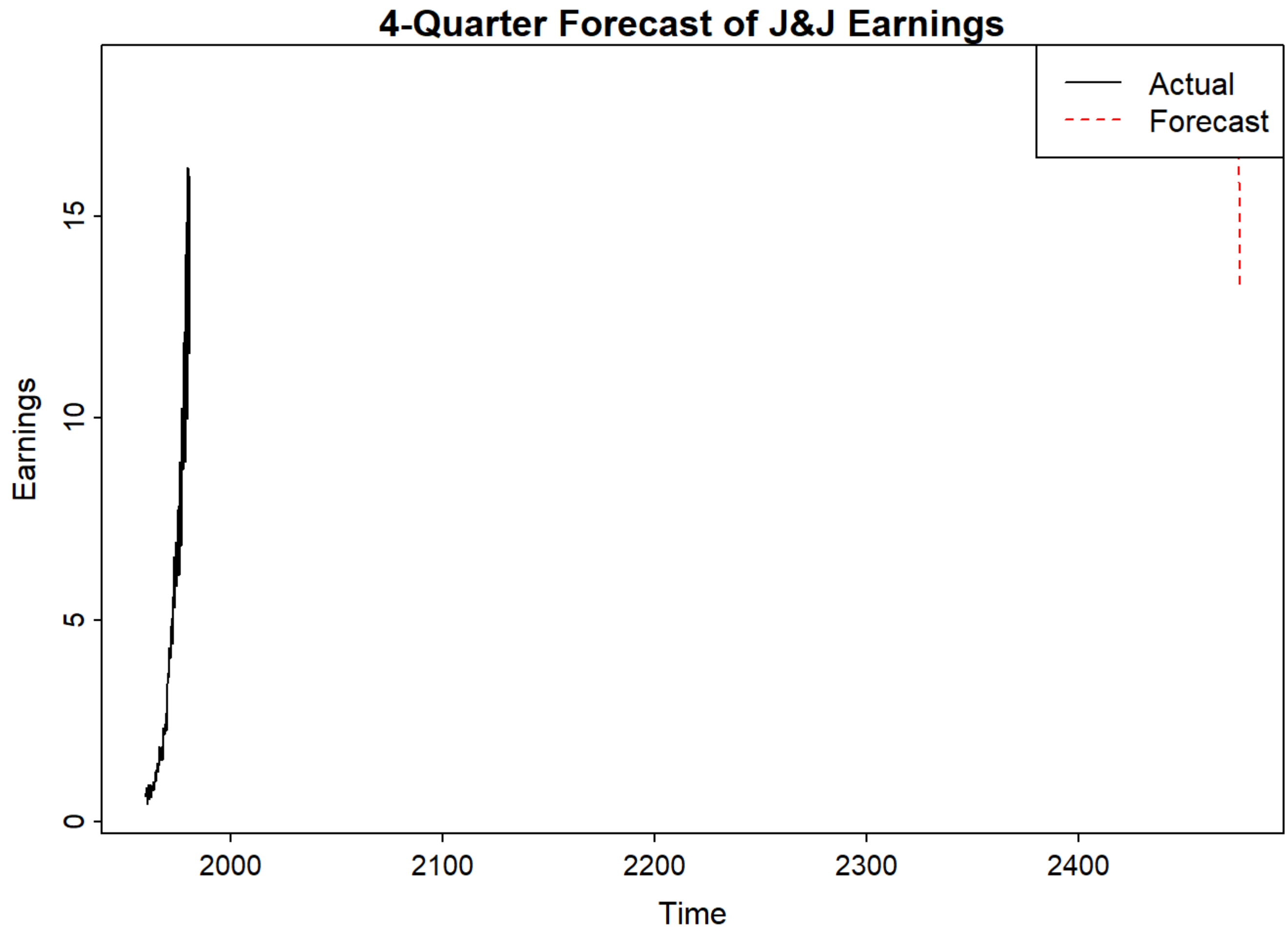
5. Forecasting the Next 4 Quarters

We now forecast the **next 4 quarters** using the fitted SARIMA model.

```
# Forecast the next 4 quarters
forecast_sarima <- sarima.for(log_jj, n.ahead = 4, p = 1, d = 1, q = 0, P = 1, D = 1, Q = 0,
S = 4)
```



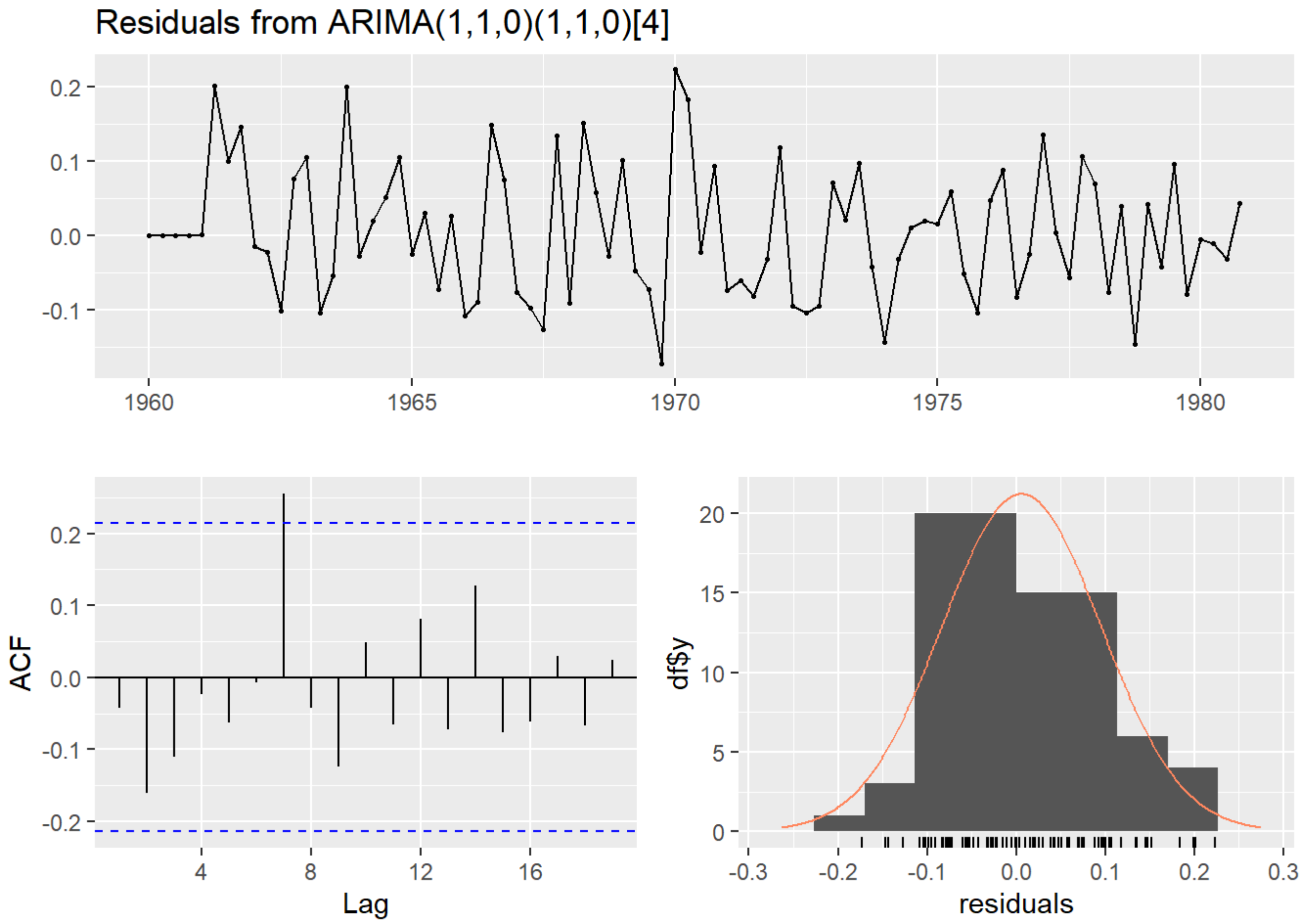

```
# Convert forecast to time series object
forecast_values <- ts(forecast_sarima$pred, start = end(jj)[1] + c(0, 1), frequency = 4)

# Plot original data with forecast
ts.plot(jj, exp(forecast_values), col = c("black", "red"), lty = c(1, 2),
        main = "4-Quarter Forecast of J&J Earnings", ylab = "Earnings", xlab = "Time")
legend("topright", legend = c("Actual", "Forecast"), col = c("black", "red"), lty = c(1, 2))
```



6. Model Diagnostics

We assess the residuals to ensure the model fits well.

```
# Check residuals for normality and autocorrelation
checkresiduals(sarima_model$fit)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,1,0)(1,1,0)[4]
## Q* = 10.176, df = 6, p-value = 0.1174
##
## Model df: 2.   Total lags used: 8
```

6.1 Residual Analysis

- **Ljung-Box Test**: Residuals should be uncorrelated (p-value > 0.05).
- **Q-Q Plot**: Check if residuals are normally distributed.

---

7. Conclusion

Based on the **SARIMA(1,1,0) × (1,1,0)[4]** model, the forecast for the next 4 quarters suggests:

1. A continuation of the seasonal pattern in earnings.
2. The model fits well, with residuals behaving like white noise.
3. **Forecasts** provide insights into future earnings trends.

---

8. Summary of Findings

- **Model Selection**: The SARIMA(1,1,0) × (1,1,0)[4] model was chosen based on ACF/PACF analysis.
- **Forecasting**: The forecast suggests continued seasonal variations in earnings.

- **Model Fit**: Diagnostics indicate the model fits well, with uncorrelated residuals.