Time Series Coursework

Shannon Sabastian Barretto

April 2025

Manchester id: 10927437

1 Description of data

```
# Read the data
x <- scan("jnj.txt")

# Convert the data to a time series with frequency=4,
# to reflect the quarterly series earnings
x <- ts(x, start = c(1960, 1), frequency = 4)
```

The data represents Johnson & Johnson's quarterly earnings per share from the first quarter of 1960 through the last quarter of 1980, covering a 21-year period (84 quarters).

2 Analysis

```
# Plot the time series
plot(x, ylab = "Earnings per Share", xlab = "Year")
```

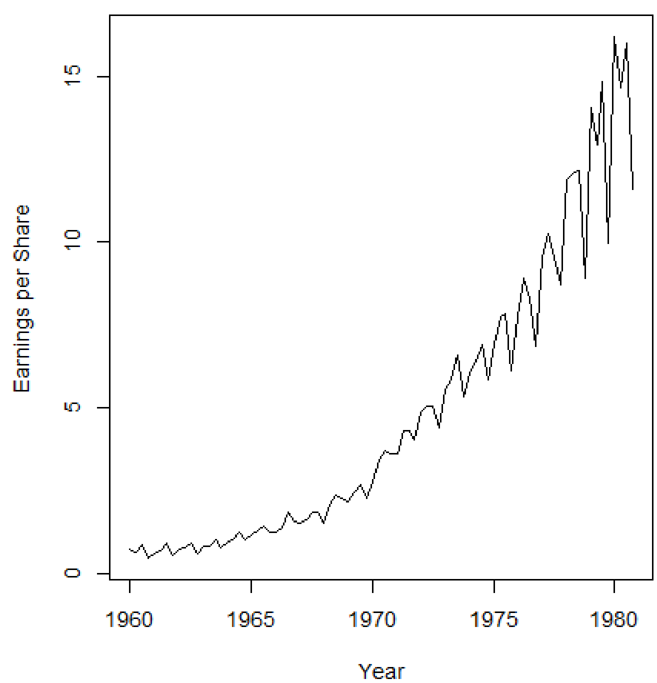
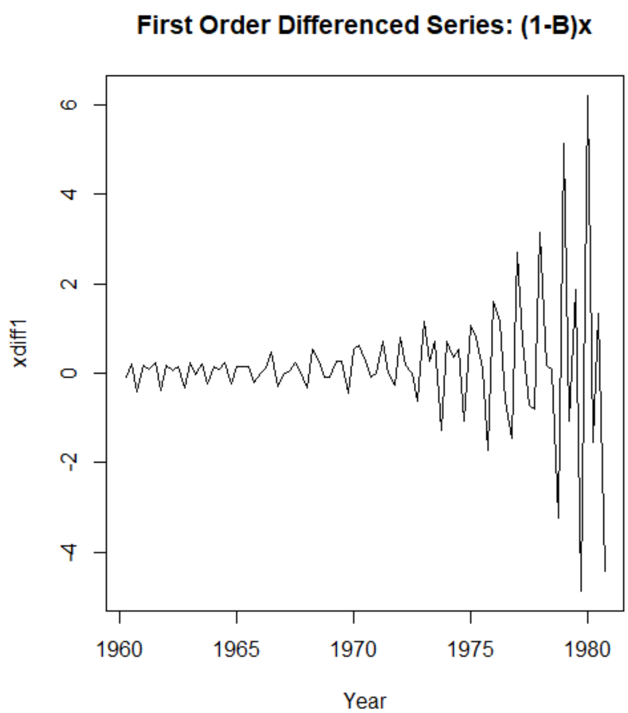


Figure 1: Johnson & Johnson Quarterly Earnings (1960-1980)

After plotting the original time series (Figure 1), we observe a **strong upward trend with increasing variance over time**. This non-stationarity needs to be addressed before fitting an ARIMA model.

First, we applied regular differencing (d=1) to remove the trend. The first-differenced series (Figure 2) shows reduced trend but still exhibits patterns consistent with seasonality, and the variance appears to increase over time. The ACF of the first-differenced series (Figure 2) shows significant spikes at seasonal lags (every 4 quarters), indicating the presence of a quarterly seasonal pattern.

```
xdiff1 <- diff(x)  # First-order differencing (1-B)x
plot(xdiff1, main = "First Order Differenced Series: (1-B)x",
     ylab = "xdiff1", xlab = "Year")
acf(xdiff1, main = "ACF of First Differenced Series: (1-B)x") # ACF
```



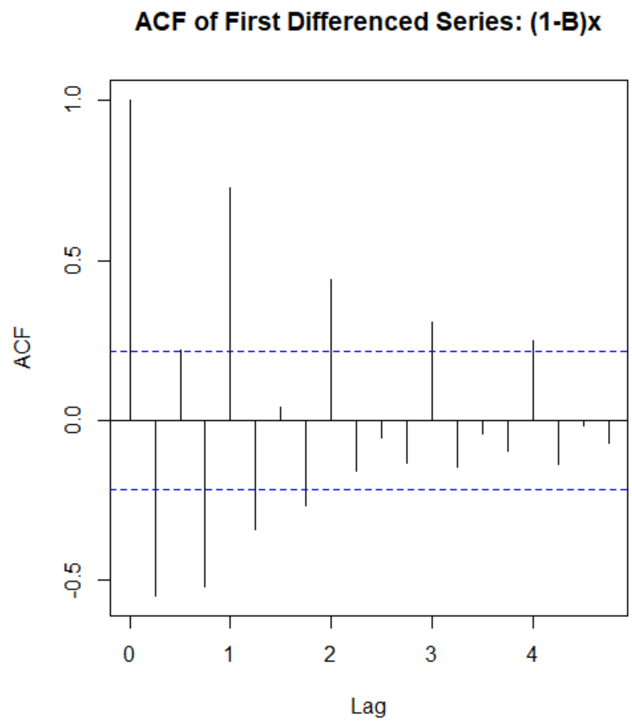


Figure 2: Plot and ACF of series after first differencing: $(1 - B)x$, i.e. removing trend

To address both trend and seasonality, we applied seasonal differencing (D=1) to the already first-differenced series, resulting in a doubly-differenced series (Figure 3). This transformation appears to have stabilized both the trend and seasonal patterns, producing a more stationary series suitable for ARIMA modeling. Even though the variance is increasing over time in Figure 3, it is better than the series in Figure 2

```
xdiff2 <- diff(diff(x, lag = 4))    # Seasonally: (1-B^4)(1-B)x
plot(xdiff2, ylab = "xdiff2", xlab = "Year")
acf(xdiff2, main = "ACF of second differenced series")       # ACF
pacf(xdiff2, main = "PACF of second differenced series")     # PACF
```

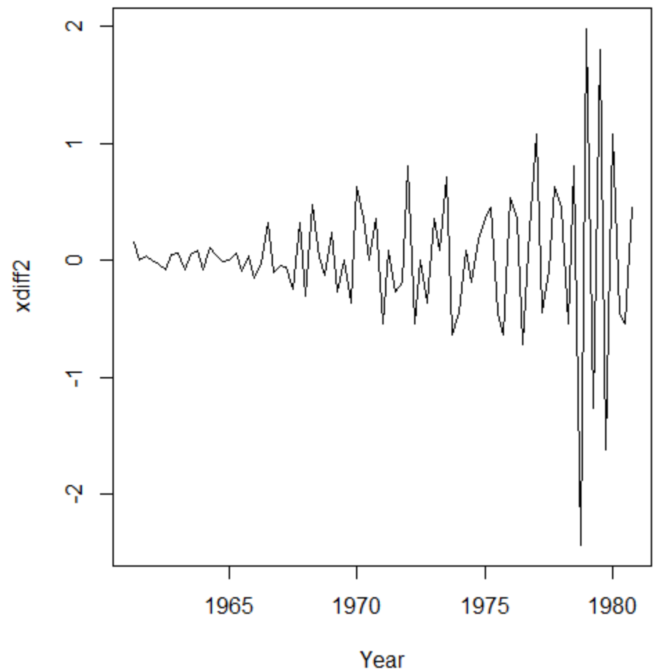
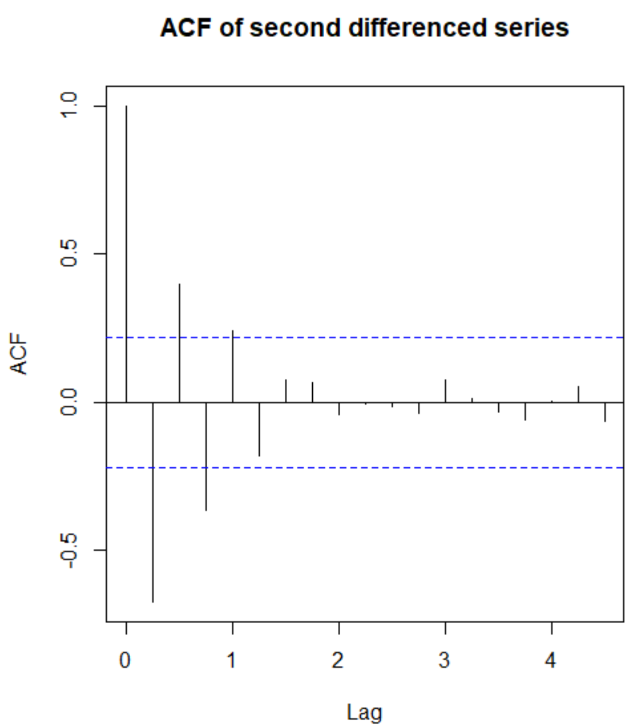


Figure 3: Plot of series after differencing again $(1 - B^4)(1 - B)x$



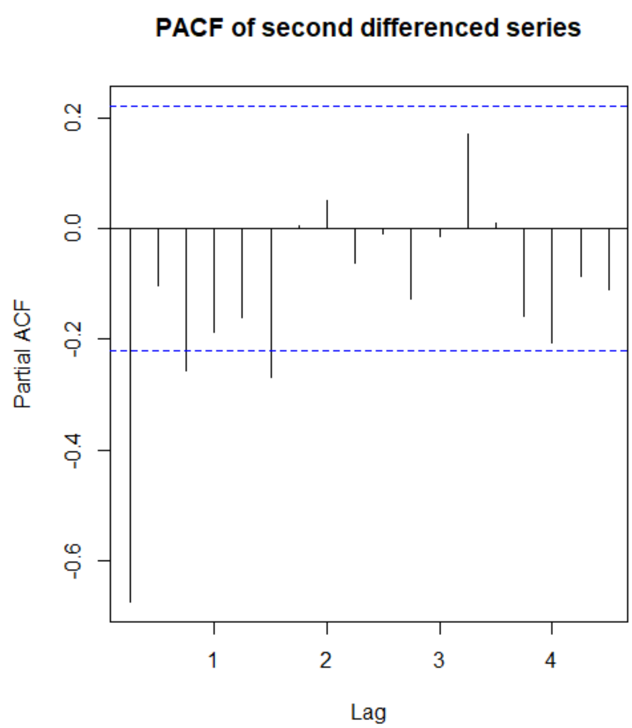


Figure 4: ACF and PACF after differencing again $(1 - B^4)(1 - B)x$ to remove seasonality

From the ACF in Figure 4:

- There's a significant negative spike at lag 1

- Most other lags fall within the significant bounds
- This pattern suggest an MA(1) component

From the PACF in Figure 4:

- There's a significant negative spike at lag 1
- There is also spike at lag 3, but since its within the 5% bound, we ignore it
- This pattern suggests an AR(1) component

Since we applied regular differencing (d=1) first to address trend, and then applied seasonal differencing (D=1) to address quarterly trend, we conclude a **tentative ARIMA model** is an $ARIMA(1,1,1) \times (0,1,0)_4$. Where:

- p=1 (AR component)
- q=1 (MA component)
- d=1 (regular differencing)
- P=0 (no seasonal AR component)
- Q=0 (no seasonal MA component)
- D=1 (seasonal differencing)
- s=4 (quarterly period)

To estimate the parameters of the above model:

```
# Fit the ARIMA(1, 1, 1)(0, 1, 0)[4] model
model <- arima(xdiff2, order = c(1, 1, 1),
               seasonal = list(order = c(0, 1, 0), period = 4))
model   # Call the model to get the parameter estimates
```

```
Call:
arima(x = jnjdiffboth, order = c(1, 1, 1), seasonal = list(order = c(0, 1, 0),
    period = 4))

Coefficients:
          ar1      ma1
      -0.5291  -1.0000
s.e.   0.1043   0.0346

sigma^2 estimated as 0.4092:  log likelihood = -74.68,  aic = 155.36
```

From the above output:

- AR component (ϕ) is -0.5291
- MA component (θ) is -1.0000

This means our $ARIMA(1,1,1) \times (0,1,0)_4$ can be written as: $(1-0.5291B)(1-B^4)(1-B)x_t = (1-1.0000B)\epsilon_t$

Note: An MA coefficient of exactly -1.0000 is at the boundary of invertibility, which indicates a potential issue with the model specification. This suggests that we may need to reconsider our differencing approach or model structure.

```
source("tsdiags.r")
# To examine the residuals of the fitted model,
# and to check correlation and Gaussianity of the residuals
tsdiags(model)
```

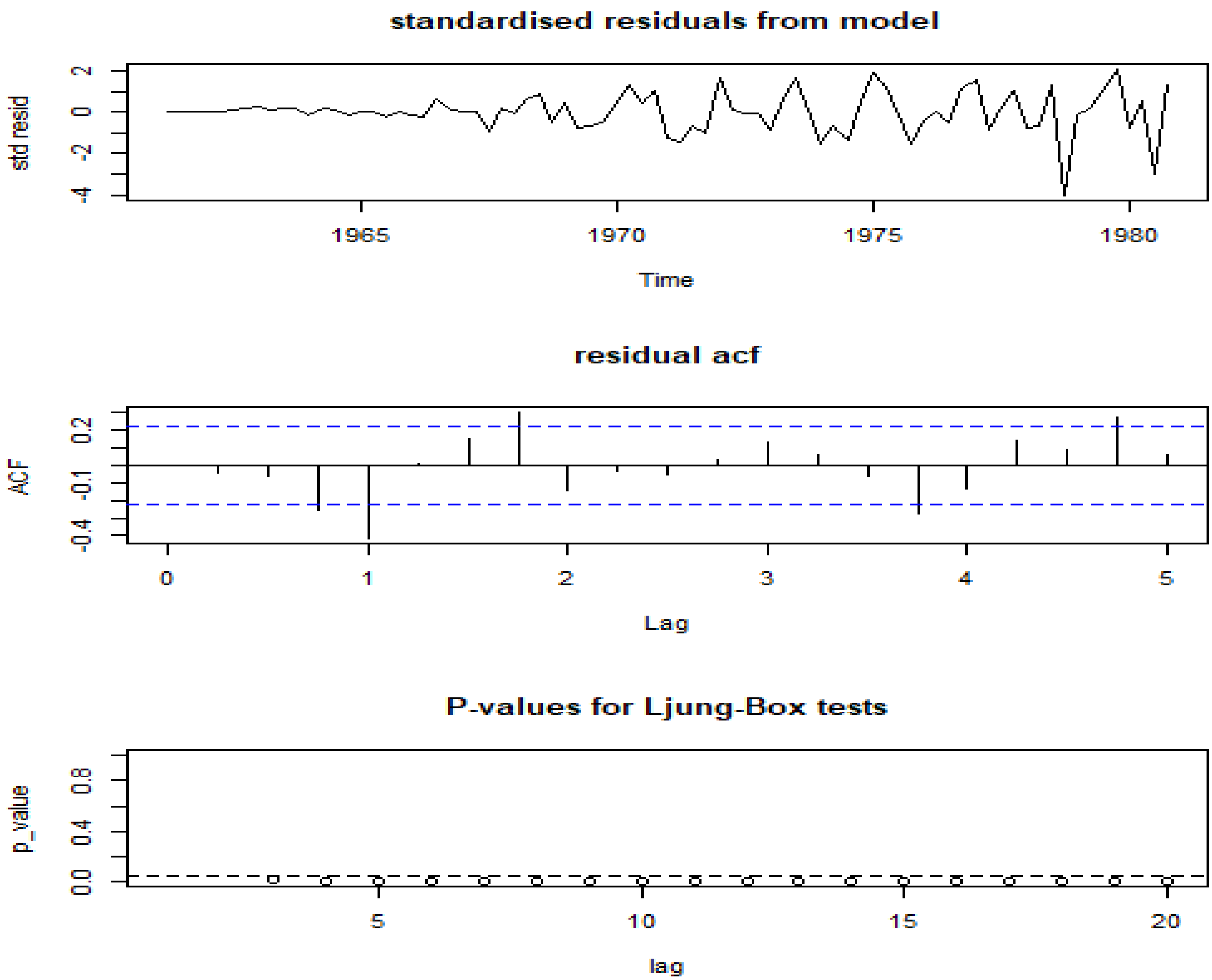


Figure 5: Residual diagnostics for model: *model*

Based on the diagnostic plots in Figure 5, we can see several issues with our $ARIMA(1,1,1) \times (0,1,0)_4$ model:

- The standardized residuals plot show increasing volatility over time. This suggest the model is not fully capturing the variance structure
- In the residual acf plot, there are multiple significant spikes outside the %5 confidence bounds

- All the p-values appear below the 0.05 significance line, which confirms that significant autocorrelation remains in the residuals

```
residuals <- resid(model)   # Extract the residuals from the fitted model
hist(residuals, main = "Histogram of Residuals")             # Histogram
qqnorm(residuals, main = "Normal Q-Q Plot of Residuals")     # Q-Q plot
qqline(residuals, col = "red")                             # Q-Q line
shapiro.test(residuals)                      # Shapiro-Wilk normality test
```

```
> shapiro.test(residuals)

        Shapiro-Wilk normality test

data:  residuals
W = 0.92431, p-value = 0.0001683
```

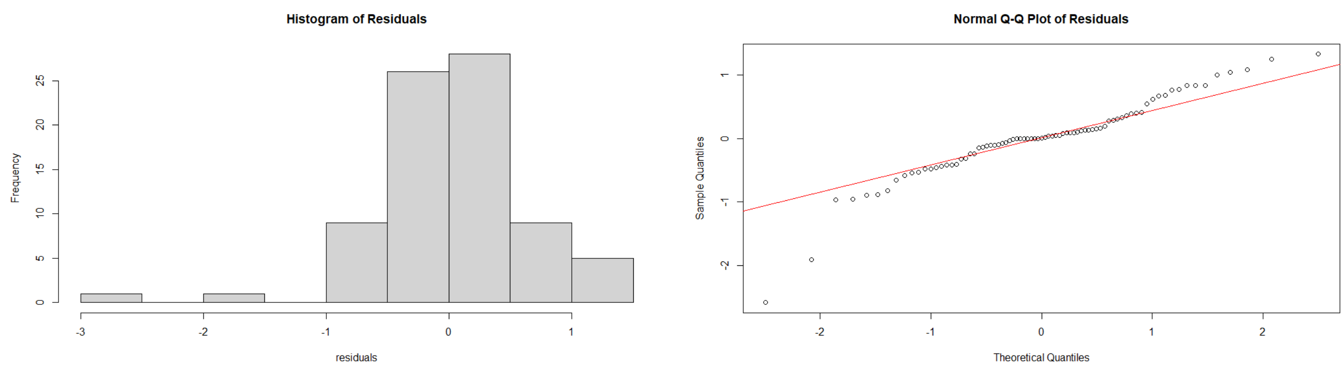


Figure 6: Histogram and Q-Q plot for residuals from model

For Gaussianity check, we look at the histogram and Q-Q plot in Figure 6. From the histogram, we can see the plot is **skewed left**. And from the Q-Q plot, the Q-Q points **do not follow the straight line**. The Shapiro test is **significant** (P-value = $0.0001683 < 0.05$), which supports the non normality of the residuals.

We can conclude our $ARIMA(1,1,1)\times(0,1,0)_4$ model is not adequately capturing the structure of our data.

```
# Fitting the same model to the log quarterly earnings
logmodel <- arima(log(x), order = c(1, 1, 1),
                  seasonal = list(order = c(0, 1, 0), period = 4))
logmodel    # Call the model to get the parameter estimates
```

```
Call:
arima(x = log(x), order = c(1, 1, 1), seasonal = list(order = c(0, 1, 0), period = 4))

Coefficients:
         ar1      ma1
      0.2384  -0.8891
s.e.  0.1518   0.0958

sigma^2 estimated as 0.008444:  log likelihood = 75.91,  aic = -145.83
```

The log-transformed model shows substantial improvement with an MA parameter no longer at the boundary (-0.8891 vs -1.0000), suggesting a more appropriate model specification.

Because of the above reason, we will choose the ***logmodel*** model, and see if we can reduce the number of parameters.

The chosen model has $\theta_1 = -0.8891$ with s.e. $= 0.0958$. Its estimate is significant at 1% with $|z_{0.005}| = |-0.8891/0.0958| = 9.280793 > 2.57$. Therefore we cannot remove this parameter.
But, in the chosen model, $\phi_1 = 0.2384$ with s.e.. $= 0.1518$. Its estimate is not significant neither at 1% nor at 5%. $|z_{0.025}| = |0.2384/0.1518| = 1.570487 < 1.96$. Therefore we can simplify the *logmodel* model by dropping the ϕ_1 parameter (AR component) from it.

We try a $ARIMA(0,1,1) \times (0,1,0)_4$ model.

```
# Fitting a parsimonious model: ARIMA(0, 1, 1)(0, 1, 0)[4]
logmodel1 <- arima(log(x), order = c(0, 1, 1),
                   seasonal = list(order = c(0, 1, 0), period = 4))
logmodel1   # Call the model to get the parameter estimates
```

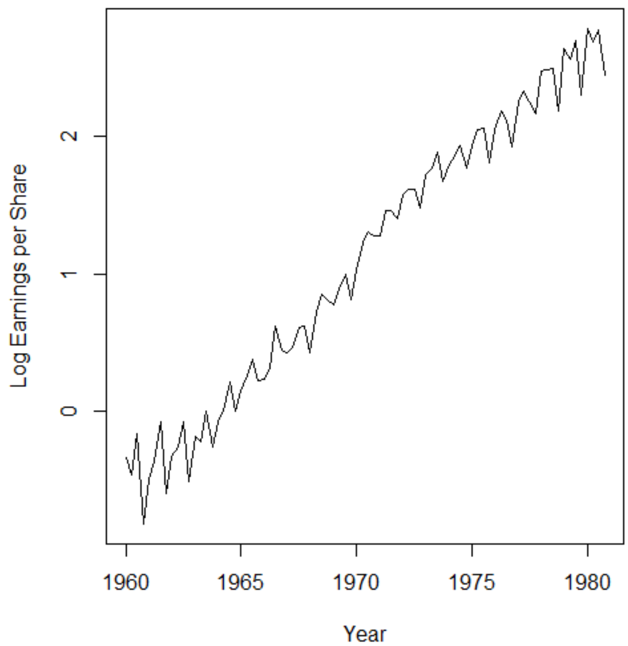
```
Call:
arima(x = log(x), order = c(0, 1, 1), seasonal = list(order = c(0, 1, 0), period = 4))

Coefficients:
          ma1
      -0.7666
s.e.   0.1107

sigma^2 estimated as 0.008724:  log likelihood = 74.76,  aic = -145.51
```

The MA(1) parameter estimate is significant at 1%. $|z| = |-0.7666/0.1107| = 6.925023 > 2.57$

```
# Plot the log quarterly earnings per share
plot(log(x), ylab = "Log Earnings per Share", xlab = "Year")
acf(log(x), main = "ACF of Log Earnings per Share") # ACF
```



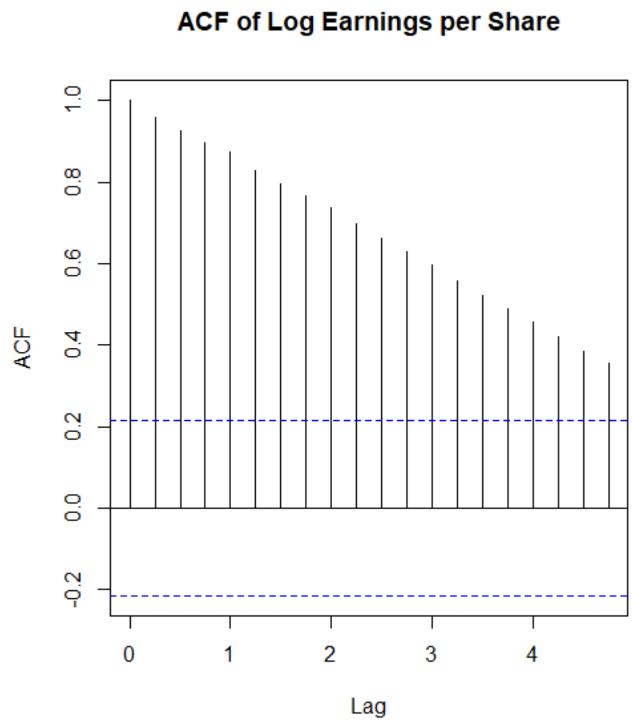


Figure 7: Plot and ACF of log quarterly earnings, $log(x)$

Comparing Figure 1 and the plot of the log quarterly earnings in Figure 7, we can see that the **log transform has stabilized the seasonal variation** in the data. From the ACF of the log quarterly earning in Figure 7, we can see a **slow decay in ACF, indicating non-stationarity**.

```
logxdiff1 <- diff(log(x)) # First-order differencing (1-B)log(x)
plot(logxdiff1, main = "First Order Differenced Series: (1-B)log(x)",
     ylab = "logxdiff1", xlab = "Year")
acf(logxdiff1, main = "ACF of First Differenced Series: (1-B)log(x)") # ACF
```

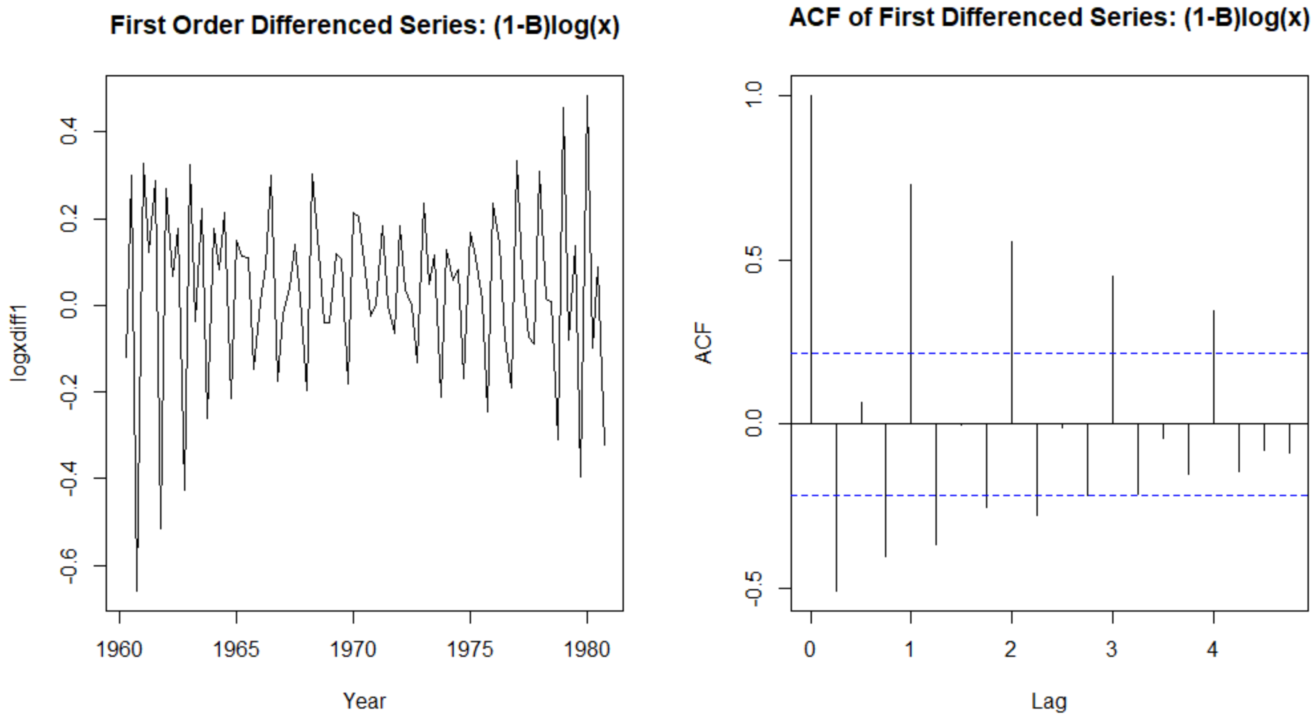


Figure 8: Plot and ACF of series after first differencing: $(1 - B)logx$, i.e. removing trend

After differencing: $(1 - B)logx$, we can still see seasonal lags in the ACF plot (Figure 8), indicating the need for a seasonal differencing.

```
logxdiff2 <- diff(diff(log(x), lag = 4))    # Seasonally: (1-B^4)(1-B)log(x)
plot(logxdiff2, ylab = "logxdiff2", xlab = "Year")
acf(logxdiff2, main = "ACF of second differenced series")    # ACF
pacf(logxdiff2, main = "PACF of second differenced series") # PACF
```

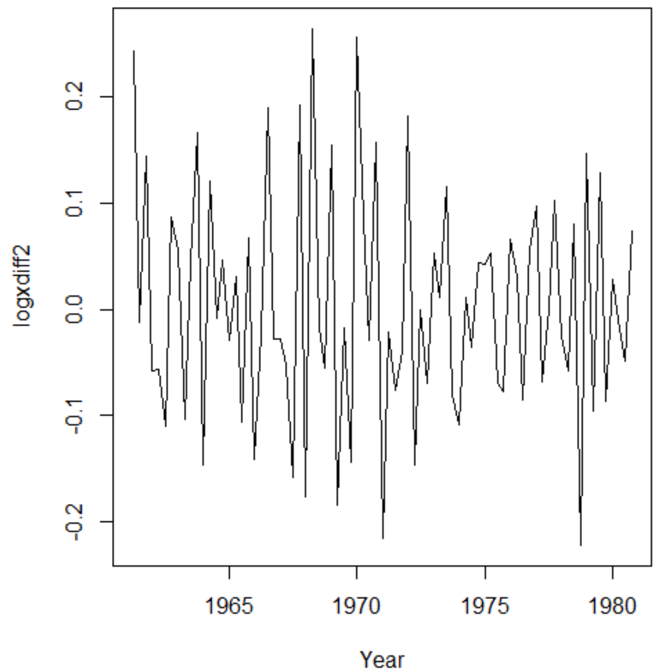


Figure 9: Plot of series after differencing again $(1-B^4)(1-B)logx$

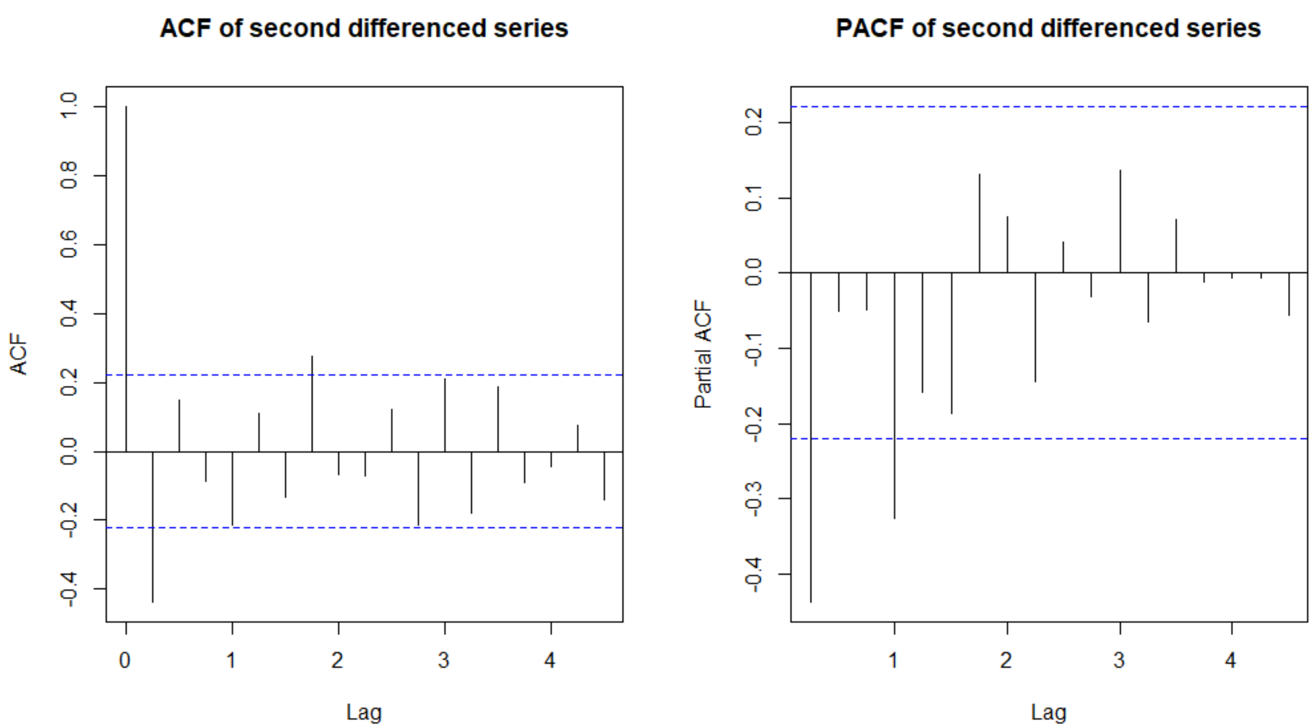


Figure 10: ACF and PACF after differencing again $(1-B^4)(1-B)logx$ to remove seasonality

From Figure 9, we can see the series is now stationary. Examining the ACF plot in Figure 10, we observe a significant negative spikes at lag 1 and we see a decaying pattern in the PACF plot. This pattern suggest an **ARIMA model:** $ARIMA(0,1,1) \times (0,1,0)_4$ model for the log transformed data, which includes the MA(1) components along with regular and seasoning differencing. No AR component and seasonal AR nor MA component as seasonal patterns are not prominent in the differenced series.

3 Results

Data	Parameters	σ^2	Log Likelihood	AIC
Original	$\phi_1 = -0.5291, \theta_1 = -1.0000$	0.4092	-74.68	155.36
Log-transformed (logmodal)	$\phi_1 = 0.2384, \theta_1 = -0.8891$	0.008444	75.91	-145.83
Log-transformed (logmodal1)	$\theta_1 = -0.7666$	0.008724	74.76	-145.51

Table 1: Summary of models fitted to the data

```
# Residual disgnostic for logmodel1 and logmodel
tsdiags(logmodel1)
tsdiags(logmodel)
```

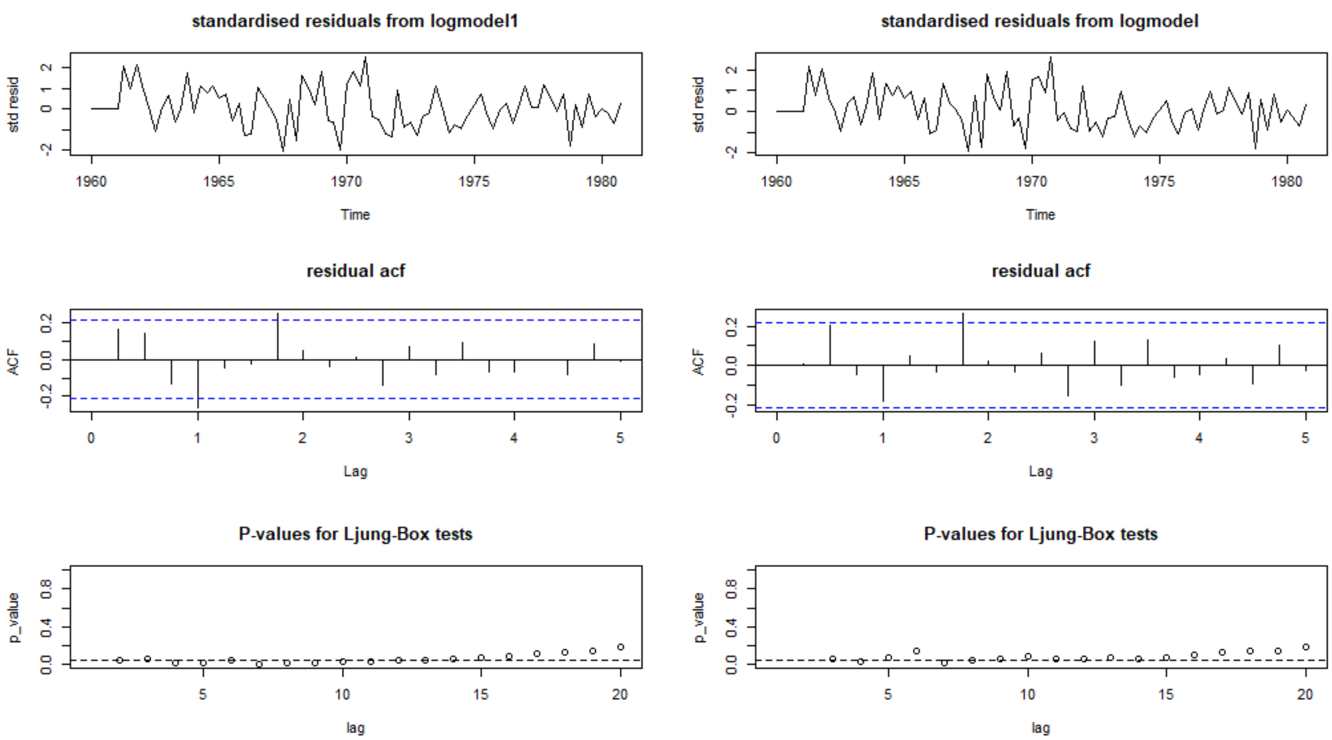


Figure 11: Comparing residual diagnostics for logmodel1 and logmodel

From the above Table 1, we can observe:

- Models fitted to log-transformed data have dramatically better log-likelihood values compared to models fitted to original data, indicating that log-transformation significantly improves model fit as it stabilizes seasonal variation
- The model fitted to original data has an MA coefficient of exactly -1.0000, which lies on the boundary of invertibility. This is a red flag indicating model misspecification. The log-transformed models do not exhibit this issue.

From Table 1, we can compare the log likelihood of *logmodel* and *logmodel1*, as they are nested. Its clear that *logmodel* has higher log likelihood value. Also comparing the residual diagnostics in Figure 11, both the residuals plots seems stationary. We should note there are less significant spikes in residuals ACF of *logmodel* and in the Ljung-Box test, *logmodel* has more p values above the 5% significant lines compared to *logmodel1*.

4 Conclusion

Therefore, if we want a more parsimonious model, where whose residuals are significant, we choose the $ARIMA(0,1,1) \times (0,1,0)$ (logmodel1), which can be written as $(1-B^4)(1-B)logx_t = (1-0.7666B)\epsilon_t$. If we do not mind a more complex model, whose AR component is not statistically significant, and has less significant residuals, then we choose the $ARIMA(1,1,1) \times (0,1,0)_4$ which can be written as $(1+0.2384B)(1-B^4)(1-B)logx_t = (1-0.8891B)\epsilon_t$