Report Project Time Series Econometrics

Chenjie LI IF3

December 23, 2023

1 Context

In this report, we are going to study the Dynamic model : Unit root tests and ARIMA(p,d,q), and especially to see the impact of seasonality, rend and cycle

2 Part 1 : Warm up - Data settings and unit root tests

2.1 Exercise 1 : Data settings - 2.5 pts

1. Data importation can be automated using R API. There exists several API facilitating economic and financial data import and update. Load the unemployment rate from the Federal Reserve of Saint Louis 1 ? (0.5 pt)

```
# Install and load the required packages
library(quantmod)
```

```
## Loading required package: xts
```

```
## Loading required package: zoo
```

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

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

```
## Loading required package: TTR
```

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

```
getSymbols("UNRATENSA", src = "FRED")  # Unemployment rate from FRED
```

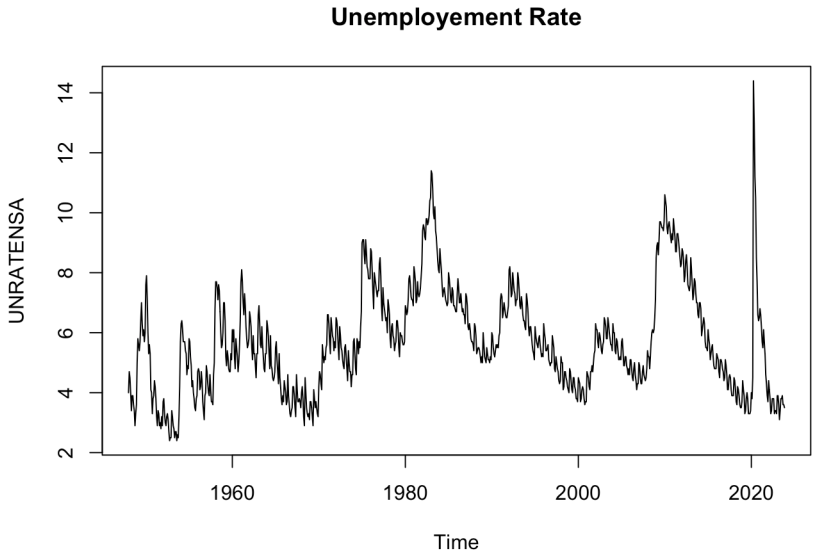
```
## [1] "UNRATENSA"
```

2. Check the status of the imported data and transform it into a right object if necessary. Plot the data. What do you observe? What type of seasonal pattern is? What type of the filters do you propose to clean the retail sales (0.5 pt).

```
str(data)
```

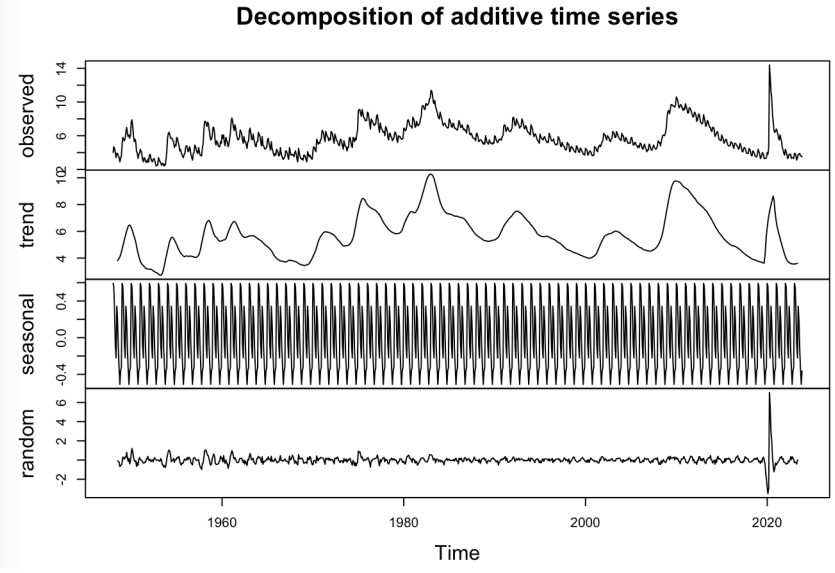
```
## An xts object on 1948-01-01 / 2023-11-01 containing:
##   Data:    double [911, 1]
##   Columns: UNRATENSA
##   Index:   Date [911] (TZ: "UTC")
##   xts Attributes:
##     $ src    : chr "FRED"
##     $ updated: POSIXct[1:1], format: "2023-12-13 12:04:36"
```

```
# Example using decompose function
unemployement_ts <- ts(data, start=c(1948,01), frequency = 12)
plot(unemployement_ts, main = "Unemployement Rate")
```



we can see clearly there's a seasonality in the data and also the cycle of pic where the unemployment rate arises due to a crisis, we then decide to use the decompose function to proof these hypothesis.

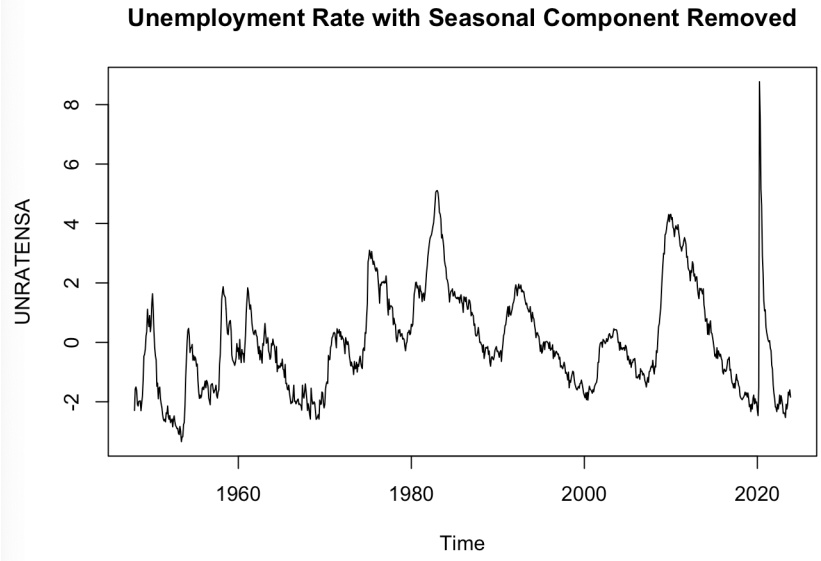
```
decomposed_data <- decompose(unemployement_ts)
plot(decomposed_data)
```



3. Run the required filter to cut the seasonal pattern? You can choose between the seasonal re- gression, the moving average filter and the decompose function (this function is based on moving average seasonal filters as well) (1 pt).

```
# Fit a seasonal regression model
model <- lm(unemployement_ts ~ as.factor(cycle(unemployement_ts)))

# Get the estimated seasonal component
seasonal_component <- predict(model)

# Remove the seasonal component
seasonal_removed <- unemployement_ts - seasonal_component

# Plot the original series and the series with seasonal component removed

plot(unemployement_ts, main = "Original Unemployment Rate")
```

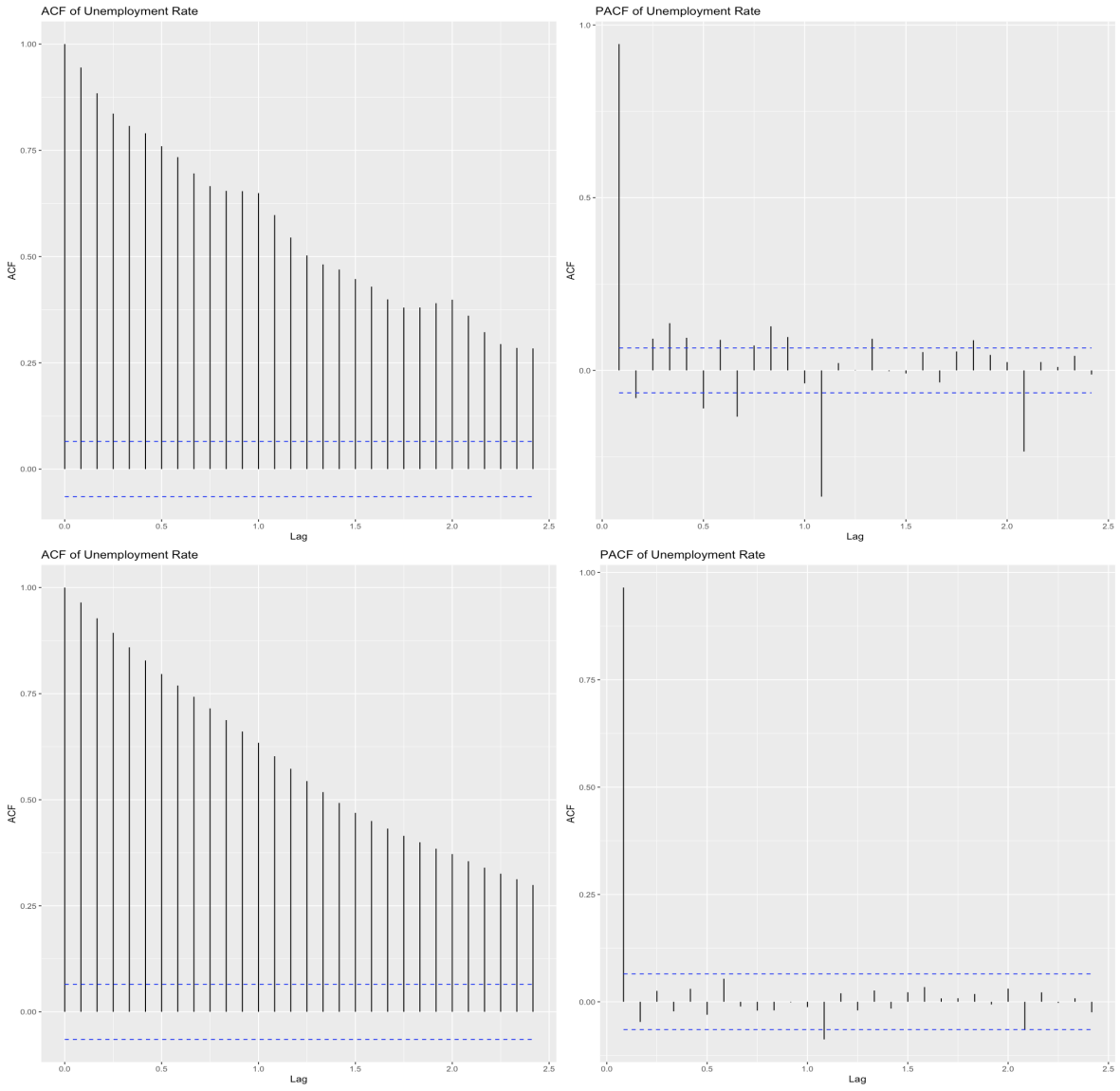
```
plot(seasonal_removed, main = "Unemployment Rate with Seasonal Component Removed")
```



we could see that the little seasonal pattern between each crisis is moved. We are going to compare the acf and pacf of the original data and the data where the seasonal pattern is removed.

```
# Plot the original series and the series with seasonal component removed
library(ggplot2)
library(ggfortify)
library(gridExtra)

# Calculate ACF and PACF for unemployement_ts
acf_unemployment_ts <- acf(unemployement_ts, plot = FALSE)
pacf_unemployment_ts <- pacf(unemployement_ts, plot = FALSE)

# Create separate plots for ACF and PACF
acf_plot <- autoplot(acf_unemployment_ts) + labs(title = "ACF of Unemployment Rate")
pacf_plot <- autoplot(pacf_unemployment_ts) + labs(title = "PACF of Unemployment Rate")

# Combine ACF and PACF plots on the same line
grid.arrange(acf_plot, pacf_plot, ncol = 2)
```

4. Grab the filtered data and check using the right tool the seasonal pattern has been deleted. Is this filtered data be modeled using an ARMA(p,q) approach (0.5 pt)



We could see on the acf and pacf graphs that non-filtered data has seasonal pattern and a lot of pacf values fall out of the confidence interval, it suggests a significant relationship between the time series and that particular lag

2.2 Exercise 2 : Unit Root tests - (2.75 pts)

We can notice the seasonally adjusted data (or the filtered one) is still showing a cyclical pattern. We need to characterize this pattern, i.e. whether the time series is stationary or not and what is the nature of the non-stationary,if so

1. Load the "urca" package. Why focusing on non-stationary is crucial in time series analysis. Summarize quickly the step-wise approach of the Dickey Fuller test.

Focusing on non-stationary is crucial in time series analysis because many time series models and statistical techniques assume stationarity. Stationarity refers to a time series where the statistical properties, such as mean and variance, remain constant over time. If a time series is non-stationary, it can lead to unreliable forecasts and inaccurate statistical inferences.

```r
# Load the "urca" package
library(urca)

seasonal_removed <- seasonal_removed[complete.cases(seasonal_removed)]

# Perform the Dickey-Fuller test
adf_test <- ur.df(seasonal_removed)

# Print the test results
summary(adf_test)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression none
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.0207 -0.1807 -0.0192  0.1386 10.1624
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## z.lag.1    -0.035364   0.008589  -4.117 4.18e-05 ***
## z.diff.lag  0.059065   0.033098   1.785   0.0747 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4395 on 907 degrees of freedom
## Multiple R-squared:  0.02006,    Adjusted R-squared:  0.01789
## F-statistic: 9.281 on 2 and 907 DF,  p-value: 0.0001023
##
##
## Value of test-statistic is: -4.1174
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau1 -2.58 -1.95 -1.62
```

2. Compute the Augmented Dickey Fuller test using the right function on the filtered data (note that the selection of the number of lags can be performed on a discretionary way or automatically).(1 pt)

```
# Perform the ADF test with selection of lags using information criteria
adf_test <- ur.df(seasonal_removed, type = "drift", selectlags = "BIC")

# Print the test results
summary(adf_test)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression drift
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.0205 -0.1806 -0.0190  0.1388 10.1626
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.0001882  0.0145871  -0.013   0.9897
## z.lag.1     -0.0353640  0.0085937  -4.115 4.22e-05 ***
## z.diff.lag   0.0590657  0.0331165   1.784   0.0748 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4398 on 906 degrees of freedom
## Multiple R-squared:  0.02006,    Adjusted R-squared:  0.01789
## F-statistic: 9.271 on 2 and 906 DF,  p-value: 0.0001034
##
##
## Value of test-statistic is: -4.1151 8.4672
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau2 -3.43 -2.86 -2.57
## phi1  6.43  4.59  3.78
```

3. Determine then the integration degree of the data ? (0.25 pts)

The integration degree of the data is 0, as indicated by the results of the Augmented Dickey-Fuller (ADF) test: The test regression includes a drift term (a constant term) and lagged differences. The coefficient of the lagged first difference ($z_{\text{diff.lag}}$) is 0.0590657, which is positive but not statistically significant at conventional levels (p-value = 0.0748 << 0.05).

The test statistic for the ADF test is -4.1151, and the critical values for the test statistic are as follows:

The 1% critical value	-3.43
The 5% critical value	-2.86
The 10% critical value	-2.57

Since the test statistic (-4.1151) is more negative than any of the critical values, we can reject the null hypothesis of a unit root (non-stationarity). This suggests that the time series is stationary or integrated of order 0.

4. Compute the Phillips and Perron test on the filtered data. Does it confirm your previous result ? (0.5 pt)

```
library(tseries)
```

```
## Warning: package 'tseries' was built under R version 4.2.3
```

```
pp.test(seasonal_removed)
```

```
## Warning in pp.test(seasonal_removed): p-value smaller than printed p-value
```

```
##
##  Phillips-Perron Unit Root Test
##
## data:  seasonal_removed
## Dickey-Fuller Z(alpha) = -32.439, Truncation lag parameter = 6, p-value
## = 0.01
## alternative hypothesis: stationary
```

The Phillips-Perron unit root test is another statistical test used to determine the stationarity of a time series. We find that the test statistic (Dickey-Fuller Z) is -32.439, and the p-value is 0.01.

Based on the p-value, we can reject the null hypothesis of a unit root and conclude that the time series is stationary. This aligns with the resualts of the test of ADF, which means the integration degree is 0.

5. Compute the KPSS test using one of the artificial data generated previously. Do we find the same conclusion ? 0.5 pt)

We are going to generate three types of artificial data: random walk; stationary data; and data with trends. Then we will use the KPSS test to determine whether they are stationary or not.

```
set.seed(123)  # For reproducibility
n <- 100  # Number of observations
random_walk <- cumsum(rnorm(n))  # Cumulative sum of normal random variables

drift <- 0.5  # Define the drift magnitude
random_walk_drift <- cumsum(rnorm(n) + drift)

stationary_series <- rnorm(n)  # Random normal variables

# KPSS Test on a Pure Random Walk
kpss_test_rw <- ur.kpss(random_walk)
summary(kpss_test_rw)
```

```
##
## ########################
## # KPSS Unit Root Test #
## ########################
##
## Test is of type: mu with 4 lags.
##
## Value of test-statistic is: 0.953
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

```
# KPSS Test on a Random Walk with Drift
kpss_test_rw_drift <- ur.kpss(random_walk_drift)
summary(kpss_test_rw_drift)
```

```
##
## ######################
## # KPSS Unit Root Test #
## ######################
##
## Test is of type: mu with 4 lags.
##
## Value of test-statistic is: 2.0529
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

```
# KPSS Test on a Stationary Series
kpss_test_stationary <- ur.kpss(stationary_series)
summary(kpss_test_stationary)
```

```
##
## ######################
## # KPSS Unit Root Test #
## ######################
##
## Test is of type: mu with 4 lags.
##
## Value of test-statistic is: 0.2667
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

Random walk and stationary should have a p-value smaller than printed p-value. The null hypothesis of the KPSS test is that the series is stationary, if the p-value is greater than the printed-value, we could consider that the ts is stationary but if the p-value is smaller than the printed value, we could conclude that the null hypothesis is rejected.

6. Find the degree of integration of the unemployement rate using the KPSS test. Does it validates the previous result 0.5 pt) ?

```
kpss.test(seasonal_removed)
```

```
## Warning in kpss.test(seasonal_removed): p-value smaller than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  seasonal_removed
## KPSS Level = 1.0782, Truncation lag parameter = 6, p-value = 0.01
```

In this output, the KPSS Level statistic is 1.0782, and the p-value is 0.01, since the p-value is less than the significance level of 0.05, we reject the null hypothesis of level stationarity. This suggests that the time series is non-stationary.

This results differ from the previous conclusion that the time series is stationary based on the ADF test. It is not uncommon for different tests to yield different results. If the truncation lag parameter is 0, it suggests that the time series is stationary (I(0)) but in our case we have lag = 6, which is greater than 0, it suggests the presence of a trend or drift, indicating a higher degree of integration (I(1) or higher).

2.3 Exercise 3 : Modeling - (2.25 pts)

1. Given the results derived from the previous sections, propose the most relevant ARMA(p,q) framework to model the retail sales dynamics. Is there an alternative to the ARMA(p,q) approach which directly deal with non-stationarity ? (0.5 pt)

Given the resulats of previous section with three tests: ADF, PP, KPSS, two over three indicates that our time series is stationary and the third one indicates the non-stationary due to the trend. we shall do more studies like using the information criteria AIC and BIC, we will use the arima(p,d,q) function in order to calculate the AIC and BIC with different differentiation degree (d) in order to find the right degree by minimizing the AIC and BIC criteria

```
# Assuming 'retail_sales_data' is your time series
arma_model <- auto.arima(seasonal_removed,d=0)
summary(arma_model)
```

```
## Series: seasonal_removed
## ARIMA(2,0,1) with zero mean
##
## Coefficients:
##          ar1     ar2     ma1
##       0.3029  0.6381  0.7271
## s.e.  0.1317  0.1282  0.1198
##
## sigma^2 = 0.1931:  log likelihood = -543.34
## AIC=1094.68   AICc=1094.72   BIC=1113.93
##
## Training set error measures:
##                         ME      RMSE       MAE       MPE     MAPE    MASE
## Training set -0.0001115977 0.4386599 0.2179726 -8.54183 93.42577 1.01772
##                      ACF1
## Training set -0.002968371
```

```
arma_model2 <- auto.arima(seasonal_removed,d=1)
summary(arma_model2)
```

```
## Series: seasonal_removed
## ARIMA(0,1,0)
##
## sigma^2 = 0.1971:  log likelihood = -552.27
## AIC=1106.53   AICc=1106.54   BIC=1111.35
##
## Training set error measures:
##                        ME     RMSE       MAE       MPE     MAPE     MASE
## Training set 0.0004914481 0.443702 0.2139448 -8.379717 92.39778 0.998914
##                    ACF1
## Training set 0.04162379
```

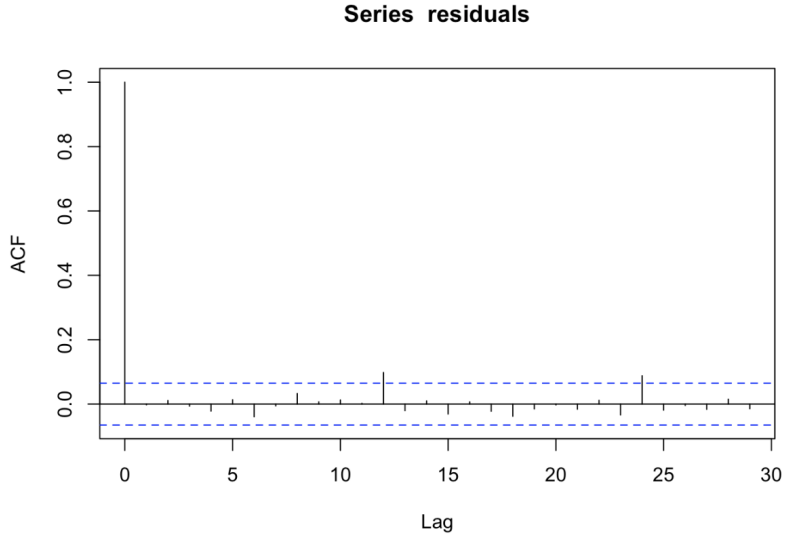
Here's the table of AIC and BIC value for different degree of differentiation :

	AIC	BIC
d = 0	1094.68	1094.72
d = 1	1106.53	1111.35

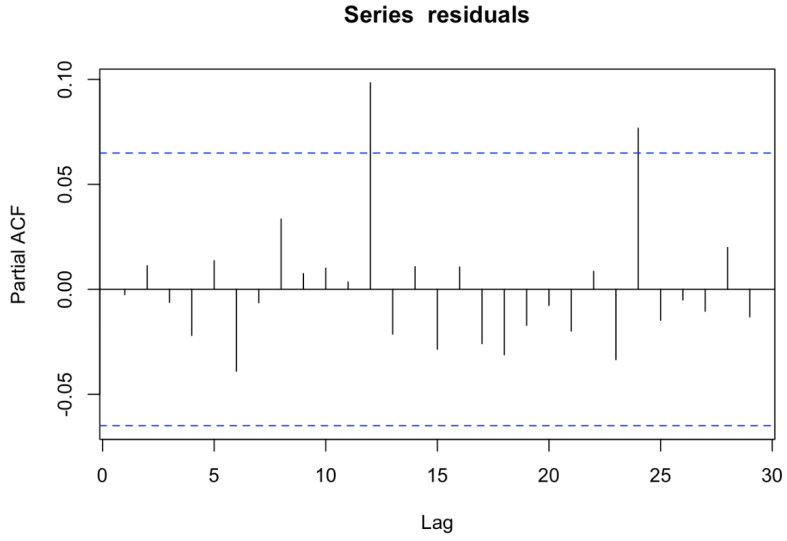
We could conclude that the best one is the d = 0, we shall than use this degree of differentiation to fit our model and conduct more quality tests.

2. Having choose the correct specification, justify the relevance of your choice and run the required quality check tests to validate you choice. (1.75 pts).

```
# Fit the ARIMA model
model <- arima(seasonal_removed, order = c(2, 0, 1))

# Residual Analysis
residuals <- residuals(model)

# ACF and PACF Plots
acf(residuals)
```




```
pacf(residuals)
```



ACF: there is a significant spike at a specific lag (bar outside the confidence interval), it suggests a strong correlation at that lag. This indicates a potential periodic pattern(trend: each crisis) in the data.

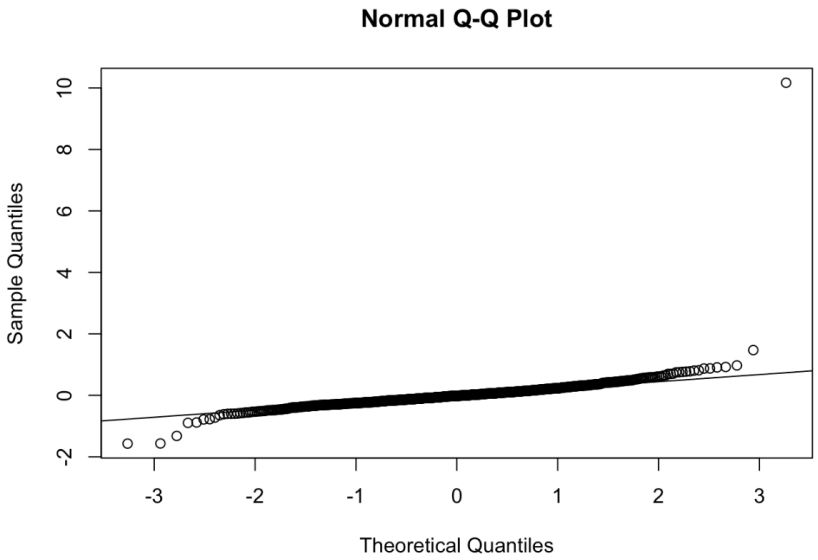
```r
# Perform the Ljung-Box test
lag <- 10  # Number of lags to test (adjust as needed)
ljung_box_test <- Box.test(residuals, lag = lag, type = "Ljung-Box")

# Print the test results
print(ljung_box_test)
```

```
##
##  Box-Ljung test
##
## data:  residuals
## X-squared = 3.4179, df = 10, p-value = 0.9698
```

Based on the provided results of the Ljung-Box test, here is the interpretation: The Ljung-Box test statistic (X-squared) is 3.4179, and the degrees of freedom (df) are 10. The p-value associated with the test is 0.9698.

Since the p-value (0.9698) is greater than the significance level of 0.05, we fail to reject the null hypothesis. This indicates that there is no evidence of significant auto correlation in the residuals, this suggests the residuals of the model are likely independently distributed, and there is no strong indication of auto correlation at the tested lags.

```r
# Create the QQ plot
qqnorm(residuals)
qqline(residuals)
```



the points on the QQ plot closely follow the reference line (i.e. they form a roughly straight line), it suggests that the data follows a normal distribution. This is a desirable outcome which validates the former interpretation.

2.4 Exercise 4 : Estimating an ARIMA(p,d,q) - (6 pts)

As seen during the class, ARIMA model were designed to deal with non stationary time series. We propose to use this kind of specification to model the Johnson Johnson stock price from 1997 until now, on a monthly basis. Prior moving to the stock price modeling, we nee to load the data from yahoo finance website. To do so, run the following code

1. Determine the degree of integration of the Johnson Johnson stock prices. (1 pt)

we firstly import the data with the function provided, and we run the ADF test.

```
jnj = tq_get("JNJ", get="stock.prices", from="1997-01-01") %>% tq_transmute(mutate_fun=to.period ,period="months")
```

```
# Load the required packages
library(tseries)

jnj_close=ts(jnj$close,start = c(1997,01),frequency = 12)
# Perform the ADF test
adf_test <- ur.df(jnj_close, type = "drift", lags = 0)

# Print the test results
summary(adf_test)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression drift
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17.9867  -2.0498  -0.0837   2.1386  18.7071
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.753129   0.564041   1.335    0.183
## z.lag.1     -0.004196   0.005904  -0.711    0.478
##
## Residual standard error: 4.449 on 321 degrees of freedom
## Multiple R-squared:  0.001571,   Adjusted R-squared:  -0.001539
## F-statistic: 0.5051 on 1 and 321 DF,  p-value: 0.4778
##
##
## Value of test-statistic is: -0.7107 1.5123
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau2 -3.44 -2.87 -2.57
## phi1  6.47  4.61  3.79
```

In this case, the p-value is 0.4778, which is greater than the typical significance level of 0.05. Therefore, we fail to reject the null hypothesis of the ADF test, suggesting that the time series has a unit root and is non-stationary. This means that the integration degree is greater than 1.

We shall use the arima function to find the best integration degree by minimizing the AIC and BIC criteria values.

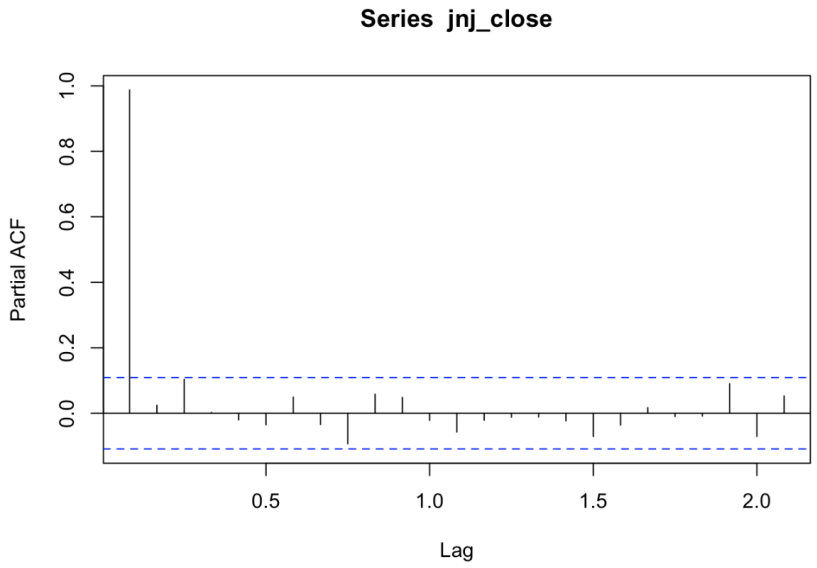
2. Determine the order of the ARIMA model, i.e the values of p, d, q to be used to model the stock prices. Note first, the assessment of p and q cannot be performed on the data expressed in level. Note besides, the determination of the values of p and q can be performed using different methods (graphical approach vs information criteria). (1.5 pts)

```
library(forecast)

auto_model <- auto.arima(jnj_close)
summary(auto_model)
```

```
## Series: jnj_close
## ARIMA(0,1,2) with drift
##
## Coefficients:
##           ma1      ma2   drift
##       -0.1207  -0.2540  0.3939
## s.e.   0.0546   0.0573  0.1499
##
## sigma^2 = 18.63:  log likelihood = -929.21
## AIC=1866.41   AICc=1866.54   BIC=1881.52
##
## Training set error measures:
##                        ME     RMSE      MAE        MPE    MAPE      MASE
## Training set -0.002066488 4.289073 3.145714 -0.2536525 3.90443 0.3465946
##                    ACF1
## Training set 0.0105827
```

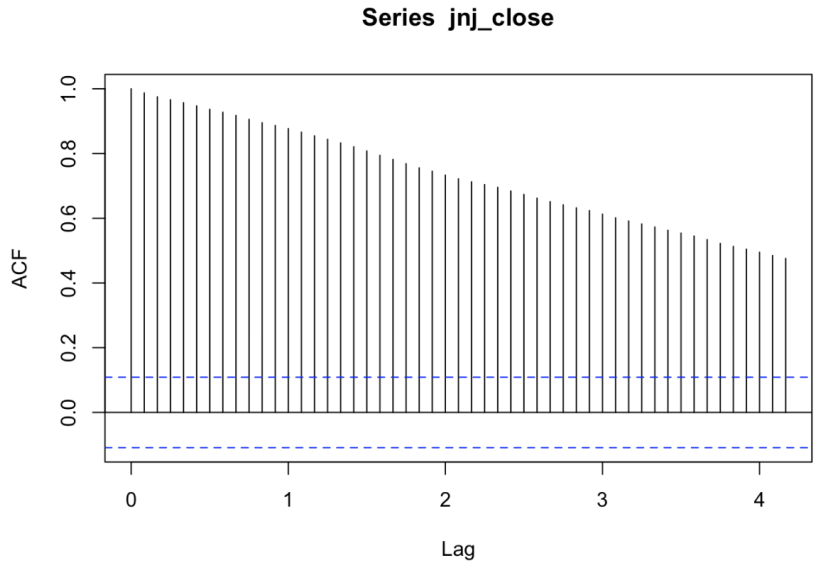
By using the information criteria AIC and BIC, we found that the integration degree is 1 and p = 0, q = 2, whith trend.

```
pacf(jnj_close)
```



By plotting the pacf, we could see that there's non partial auto correlation after the lag=0, which means that the AR(p) part has a p = 0

```
acf(jnj_close,lag.max = 50)
```



By plotting the acf, we could see that there's a decaying auto-correlation after the lag=0, which means that the MA(q) part has q different than 0

After all theses graphic and information criteria tests, we find out that the arima(0,1,2) is right and logical.

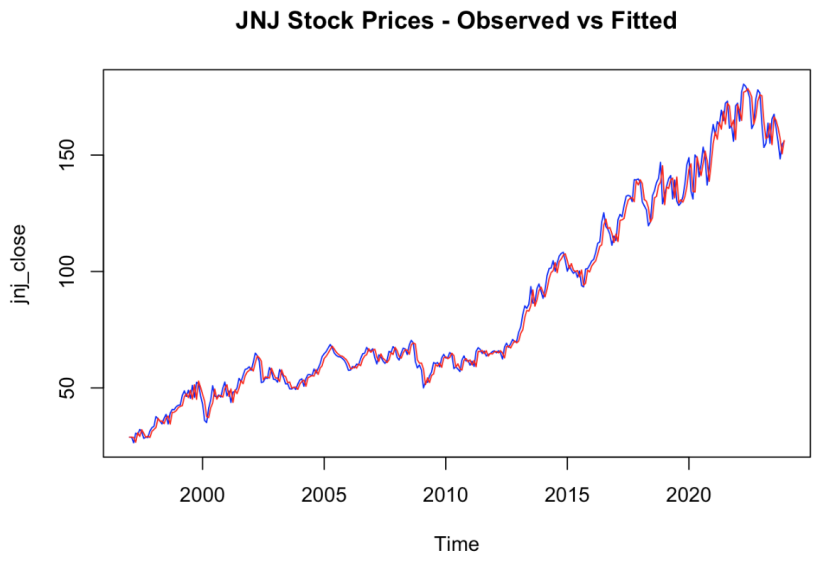
We decide to choose this model for the forecasting part

3. Estimate the corresponding ARIMA(p,d,q) model to the values of p,d,q selected previously. Check the estimated coefficients and compute the fit of the model. Plot (within the same chart) the estimated values of the stock price and the observed one. (1.25 pts)

```
arima_model_jnj <- arima(jnj_close, order = c(0, 1, 2))
summary(arima_model_jnj)
```

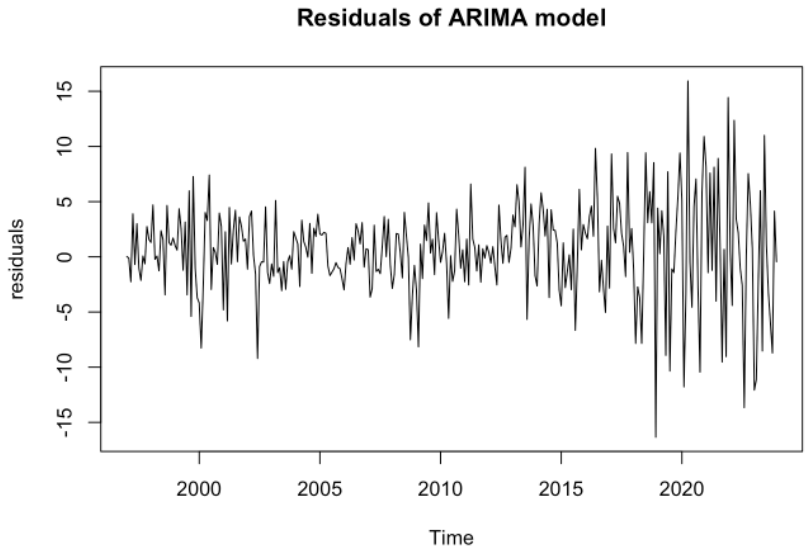
```
##
## Call:
## arima(x = jnj_close, order = c(0, 1, 2))
##
## Coefficients:
##           ma1     ma2
##       -0.0950  -0.226
## s.e.   0.0546   0.056
##
## sigma^2 estimated as 18.82:  log likelihood = -932.34,  aic = 1870.67
##
## Training set error measures:
##                      ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 0.5737446 4.331102 3.174102 0.5932878 3.904111 0.9922891
##                      ACF1
## Training set -0.01272293
```

```
# Plotting the estimated vs observed values
plot(jnj_close,type = "l", main = "JNJ Stock Prices - Observed vs Fitted", col = "blue")
lines(fitted(arima_model_jnj), col = "red")
```

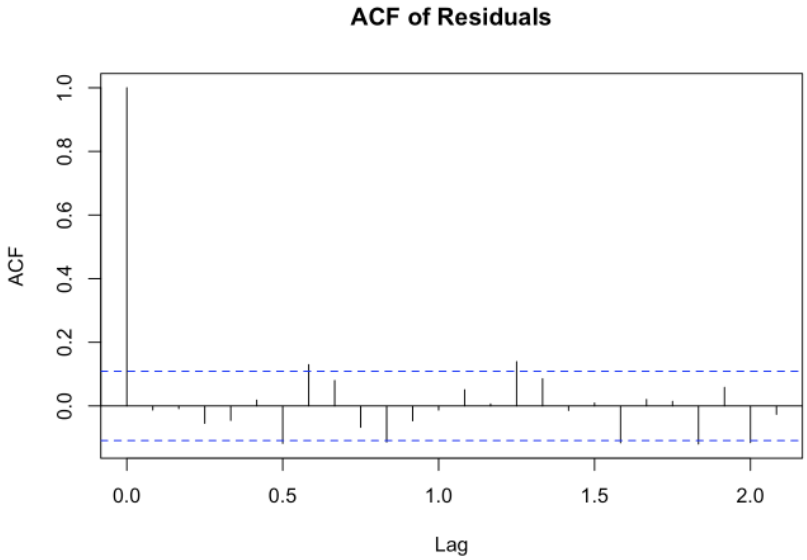


4. Calculate the residual of the model, given as the difference between JJ and JJ. Compute the required quality checks on the residuals (1.75 pts).

```
residuals <- residuals(arima_model_jnj)

# Perform quality checks
plot(residuals, main = "Residuals of ARIMA model")
```



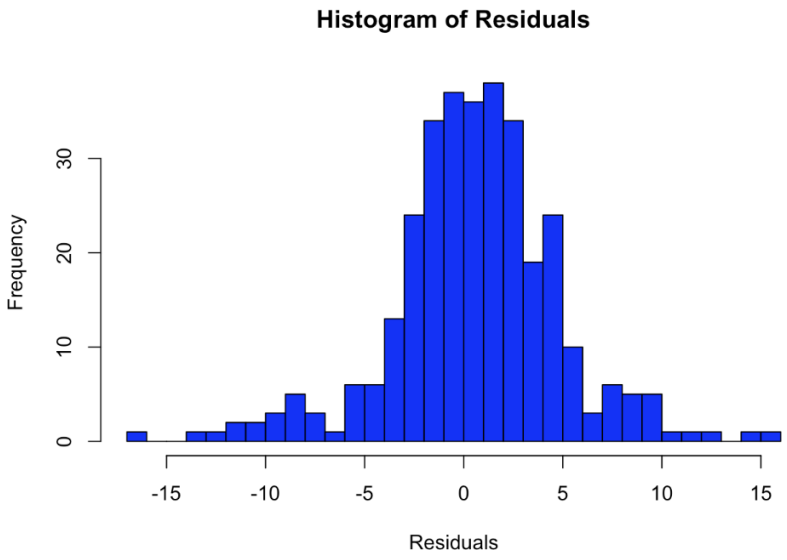

```
acf(residuals, main = "ACF of Residuals")
```



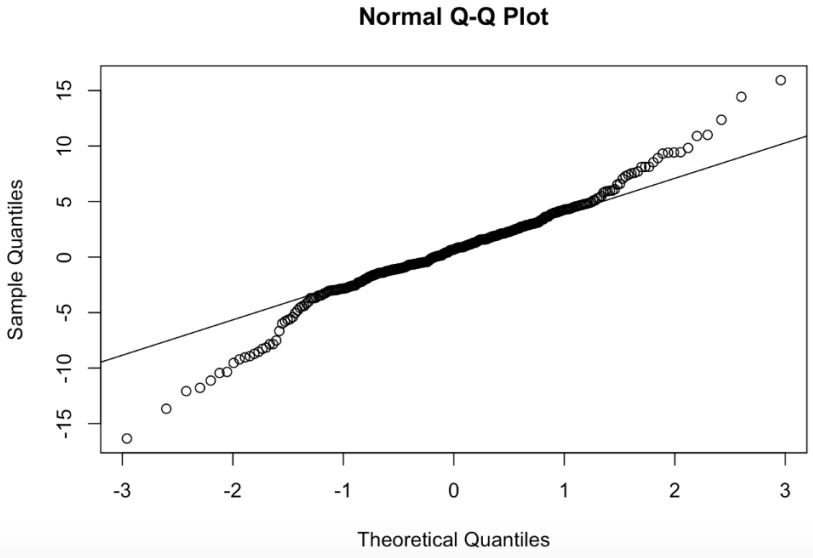
After testing the ACF and PACF graph of the arima model, we find the residuals is nearly 0 between 1990 and 2018 but the error between our model and initial value become more and more great between 2018 and 2023, some explanation I suggest is that the covid crisis which increases the volatility and make the stock more unpredictable and this may matches the spikes of the ACF graph.

We are going to run some tests to know whether the residual values are normally distributed and also whether the residual values have autocorrelation or not

```
hist(residuals, main = "Histogram of Residuals", xlab = "Residuals", breaks = 30, col = "blue")
```




```
qqnorm(residuals); qqline(residuals)
```



We could remark that the residual values are normally distributed on the histogram and the points on the QQ plot closely follow the reference line (i.e. they form a roughly straight line), it suggests that the data follows a normal distribution. This is a desirable outcome.

```
Box.test(residuals, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  residuals
## X-squared = 0.052934, df = 1, p-value = 0.818
```

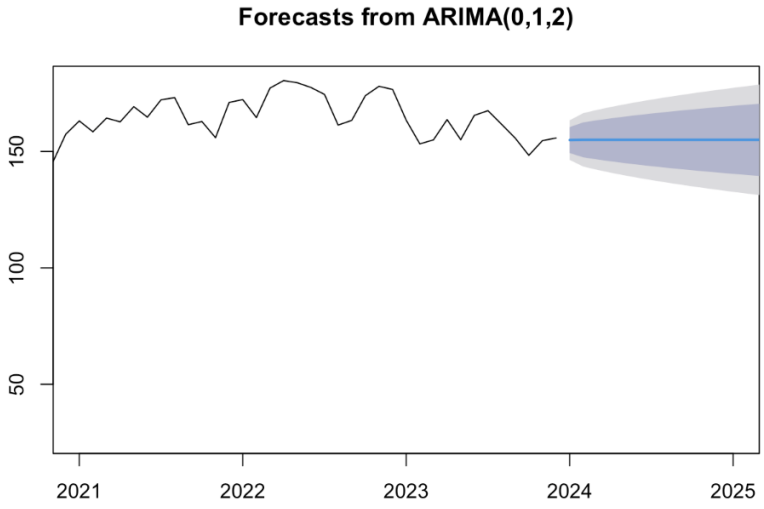
Since the p-value (0.818) of the Ljung box is greater than 0.05, we fail to reject the null hypothesis. This suggests that there is no evidence of significant autocorrelation in the residuals which means that the residuals appear to be independent and do not exhibit any systematic patterns of autocorrelation. This is also a desirable outcome as it indicates that the model adequately captures the autocorrelation structure of the data, and the residuals are not violating the assumption of independence.

5. Using the estimated coefficients, generate a forecast over the next 3 months. Calculate the confidence interval of the forecasted points (0.5 pt) .

```r
library(forecast)
# Assuming 'arima_model' is your fitted ARIMA model
forecasted_values <- forecast(arima_model_jnj, h = 15)  # 'h = 3' for three months ahead

print(forecasted_values)
```

```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jan 2024       154.8968 149.3377 160.4559 146.3948 163.3987
## Feb 2024       154.9976 147.4999 162.4954 143.5309 166.4644
## Mar 2024       154.9976 146.6034 163.3919 142.1597 167.8356
## Apr 2024       154.9976 145.7937 164.2016 140.9214 169.0738
## May 2024       154.9976 145.0498 164.9455 139.7837 170.2116
## Jun 2024       154.9976 144.3577 165.6376 138.7252 171.2701
## Jul 2024       154.9976 143.7080 166.2873 137.7316 172.2637
## Aug 2024       154.9976 143.0936 166.9017 136.7920 173.2033
## Sep 2024       154.9976 142.5095 167.4858 135.8987 174.0966
## Oct 2024       154.9976 141.9515 168.0438 135.0453 174.9500
## Nov 2024       154.9976 141.4164 168.5789 134.2269 175.7684
## Dec 2024       154.9976 140.9016 169.0937 133.4396 176.5557
## Jan 2025       154.9976 140.4050 169.5903 132.6800 177.3152
## Feb 2025       154.9976 139.9247 170.0706 131.9455 178.0498
## Mar 2025       154.9976 139.4592 170.5361 131.2336 178.7616
```

```r
plot(forecasted_values, xlim = c(2021, 2025), type="l")
```



Since the auto regression forecasting values are constant, which is not normal, we are going to calculate predicted values manually.

We are going to predict three future values by implementing manually the algorithm with the coefficients found in the arima model and also taking in count the error terms.

```
# Last observed value
last_value <- tail(jnj_close, n = 1)

# Extract residuals (error terms) and get the last two values
residuals <- residuals(arima_model_jnj)
last_residuals <- tail(residuals, n = 2)

# Coefficients from your ARIMA model
ma1 <- -0.1208
ma2 <- -0.2527
drift <- 0.3924

# Forecast for the next three periods
forecast_1 <- last_value + drift + ma1 * last_residuals[2] + ma2 * last_residuals[1]
forecast_2 <- forecast_1 + drift  # Future errors assumed as 0
forecast_3 <- forecast_2 + drift  # Future errors assumed as 0

# Print forecasted values
forecast_values <- c(forecast_1, forecast_2, forecast_3)
print(forecast_values)
```

```
## [1] 155.1902 155.5826 155.9750
```

```
sigma <- sqrt(arima_model_jnj$sigma2)
# Assuming a 95% confidence interval
critical_value <- qnorm(0.975)  # Approximately 1.96
margin_error_1 <- critical_value * sigma * sqrt(1 + 1)  # For first forecast
margin_error_2 <- critical_value * sigma * sqrt(1 + 2)  # For second forecast
margin_error_3 <- critical_value * sigma * sqrt(1 + 3)  # For third forecast
lower_bounds <- forecast_values - c(margin_error_1, margin_error_2, margin_error_3)
upper_bounds <- forecast_values + c(margin_error_1, margin_error_2, margin_error_3)

confidence_intervals <- data.frame(
  Time = 1:3,
  Forecast = forecast_values,
  Lower_95 = lower_bounds,
  Upper_95 = upper_bounds
)
print(confidence_intervals)
```

```
##   Time Forecast Lower_95 Upper_95
## 1    1 155.1902 143.1667 167.2138
## 2    2 155.5826 140.8568 170.3084
## 3    3 155.9750 138.9711 172.9789
```

In conclusion we find out different values by forecasting manually the future values than the auto foresting model. This seems more logical.

2.5 Exercise 5 : Unit root test another one - 4 pts

1. Present the Zivot and Andrews (1992) test paying attention to the nature of the breaks. Explain the strategy of the test (0.5 pt).

The Zivot and Andrews (1992) unit root test is an important variation of the traditional Augmented Dickey-Fuller (ADF) test, particularly useful for time series that have experienced significant changes due to external shocks or structural changes in the underlying process. By considering a potential structural break, this test provides a more nuanced approach than standard unit root tests, which might misinterpret such breaks as evidence of non-stationarity.Here's the nature and strategy of this test:

The nature of the Zivot and Andrews Test:

It bases on structural breaks consideration: It is able to accommodate one structural break in the time series.

There are different types of break:

- A single break in the intercept (level shift).

- A single break in both the intercept and the trend (level and trend shift).

- A break in the trend only (trend shift).

Strategy of the Test:

- **Testing for Unit Root with Breaks:** The Zivot and Andrews test enhances the ADF test by including a one-time structural break in the series. This break is not pre-specified but is determined endogenously within the test procedure.

- **Sequential Approach:** The test involves running a sequence of ADF regressions, each time with a different break date. For each potential break date, a dummy variable is included in the regression to capture the break.

- **Selecting the Break Date:** The break date is chosen where the t-statistic of the unit root test is most negative across all the regressions. This implies selecting the date where the evidence against the null hypothesis of a unit root (with a break) is strongest.

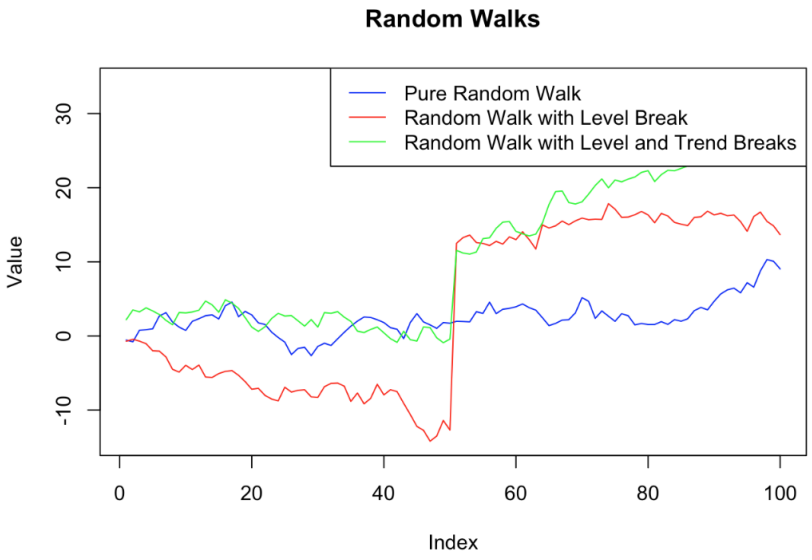
- **Null Hypothesis:** The null hypothesis of the Zivot and Andrews test is that the time series has a unit root with a structural break at an unknown point in time.

- **Alternative Hypothesis:** The alternative is that the series is stationary with a structural break.

- **Rejection of Null:** The null hypothesis is rejected if the t-value statistic associated with the unit root is lower than the critical value, suggesting stationarity with a break.

2. Generate 3 new random walks. The first one is a pure random walk, the second is a random walk with a break in level and the third on will be a random walk with both a break in level and in the trend.(0.75 pt)

```
set.seed(123)  # For reproducibility
n <- 100       # Number of observations

# 1. Pure Random Walk
rw1 <- cumsum(rnorm(n))  # Cumulative sum of normal random variables

# 2. Random Walk with a Break in Level
breakpoint <- 50  # Define the point of level break
level_change <- 10  # Define the magnitude of the level change

# Generating the random walk with a break in level
rw2 <- c(cumsum(rnorm(breakpoint)), cumsum(rnorm(n - breakpoint)) + level_change + rw1[breakpoint])

# 3. Random Walk with Breaks in Both Level and Trend
trend_change <- 0.2  # Define the magnitude of trend change

# Generating the random walk with both level and trend breaks
rw3 <- c(cumsum(rnorm(breakpoint)), cumsum(rnorm(n - breakpoint, mean = trend_change)) + level_change + rw1[breakp
oint])

plot(rw1, type = "l", col = "blue", ylim = range(c(rw1, rw2, rw3)), main = "Random Walks", ylab = "Value")
lines(rw2, col = "red")
lines(rw3, col = "green")
legend("topright", legend = c("Pure Random Walk", "Random Walk with Level Break", "Random Walk with Level and Tren
d Breaks"), col = c("blue", "red", "green"), lty = 1)
```



3. Compute the appropriate Zivot and Andrews (1992) test for the generated random walk. Summarize your output within a table.(1.25 pts)

```r
library(urca)

# Zivot and Andrews test for the first random walk (rw1)
za_rw1 <- ur.za(rw1, model = "both", lag = 2)  # 'both' for break in level and trend

# Zivot and Andrews test for the second random walk (rw2)
za_rw2 <- ur.za(rw2, model = "both", lag = 2)

# Zivot and Andrews test for the third random walk (rw3)
za_rw3 <- ur.za(rw3, model = "both", lag = 2)
```

```r
# Assuming za_rw1, za_rw2, za_rw3 are the Zivot-Andrews test results for the random walks

# Summary for each random walk
summary_rw1 <- summary(za_rw1)
summary_rw2 <- summary(za_rw2)
summary_rw3 <- summary(za_rw3)

# Print the summaries
print("Summary for Pure Random Walk:")
```

Type of ts	Test Statistic	Critical Values	Potential Break Point
Pure Random Walk	-3.0532	-5.57 (1%)	Position 77
Random Walk with Level Break	-18.6758	-5.57 (1%)	Position 50
Random Walk with Level and Trend Breaks	-8.1066	-5.57 (1%)	Position 50

The Zivot-Andrews test successfully identified and accounted for structural breaks in the second and third series (random walks with level break and level+trend breaks).

Because the test statistic are much more lower than the critical values which indicates the rejection of the null hypothesis (unit root with)

In the case of the pure random walk without an actual structural break, the test did not falsely indicate stationarity, underscoring the test's effectiveness in distinguishing between true structural breaks and random fluctuations typical of a random walk.

4. Is it relevant to use such test for the filtered retail sales. Justify. Compute the Zivot and Andrews (1992) unit root test using the US retail sales (1 pt).

Using the Zivot and Andrews unit root test on the unemployment rate can be relevant and beneficial for several reasons:

Structural Breaks in Economic Data: The unemployment rates are often subject to structural breaks due to policy changes, economic crises, or other significant events. The Zivot and Andrews test can detect unit roots in the presence of such structural breaks.

Differentiating Between Non-stationarity and Breaks: Traditional unit root tests, like the Augmented Dickey-Fuller (ADF) test, may not distinguish between non-stationarity due to a unit root and non-stationarity caused by structural breaks. The Zivot and Andrews test addresses this by allowing for a break in the series.

Policy and Economic Analysis: Understanding whether changes in unemployment rates are due to temporary shocks or long-term trends (structural changes) is crucial for economic policy and forecasting.

Seasonal Adjustment: Once a series is seasonally adjusted, analyzing it for unit roots and structural breaks becomes more meaningful, as seasonal patterns are already accounted for, and the focus can shift to the underlying trend and cyclical components.

```
summary(ur.za(unemployement_ts,model = "both"))
```

```
##
## ################################
## # Zivot-Andrews Unit Root Test #
## ################################
##
##
## Call:
## lm(formula = testmat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.5418 -0.2905 -0.1083  0.1737  9.9227
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.653e-01  7.324e-02   4.989 7.29e-07 ***
## y.l1         9.292e-01  1.197e-02  77.619  < 2e-16 ***
## trend        8.705e-05  1.024e-04   0.850  0.39566
## du           2.531e-01  9.810e-02   2.580  0.01003 *
## dt          -2.765e-03  8.492e-04  -3.256  0.00117 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5645 on 905 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.8965, Adjusted R-squared:  0.896
## F-statistic:  1959 on 4 and 905 DF,  p-value: < 2.2e-16
##
##
## Teststatistic: -5.9139
## Critical values: 0.01= -5.57 0.05= -5.08 0.1= -4.82
##
## Potential break point at position: 724
```

As we could see that the zw test results, the test statistic value is much lower than the critical values which imply that the null hypothesis is rejected and there's a break in the time series which could explain more precisely the trend we found in the first section.

2.6 Exercise 6 : Modeling the business cycle - 4,5 pts

Based on all your knowledge, propose the most appropriate specification to model the monthly credit spread dynamics in the US. The data is available under the Fed of Saint Louis website. You have to load the Moody's Seasoned Aaa Corporate Bond and the Moody's Seasoned Baa Corporate Bond Yield years interest rate. The credit spread is the difference between the Baa index and the Aaa one. Produce a complete analysis (seasonality, stationarity, modeling, residual checking and forecasting of the credit spread. The forecasting horizon is set at three months. The study should incorporate illustrating charts, detailed and motivated comments and relevant results.

Importation of corporate bond yield rates AAA and BAA and we plot the monthly credit spread and also the statute of this time series.

```
library(quantmod)

# Load Moody's Seasoned Aaa and Baa Corporate Bond Yield rates
getSymbols("AAA", src = "FRED")  # Aaa Corporate Bond Yield
```

```
## [1] "AAA"
```

```
getSymbols("BAA", src = "FRED")  # Baa Corporate Bond Yield
```

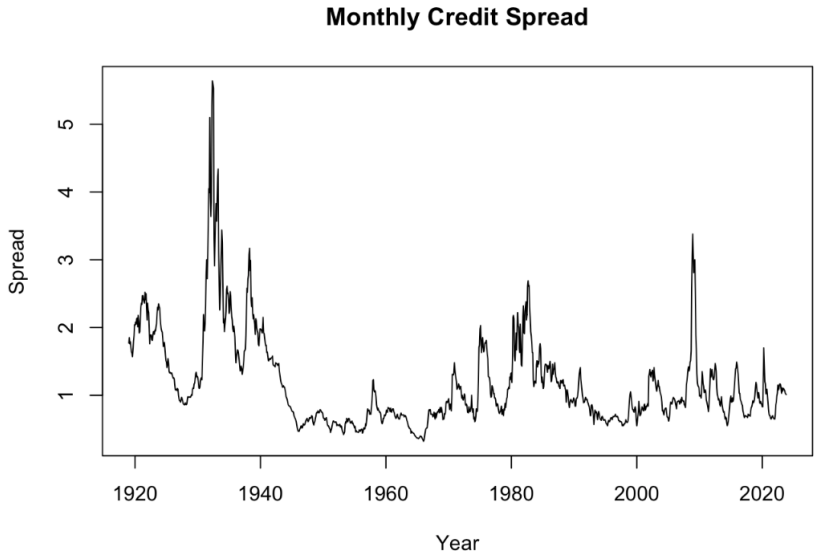
```
## [1] "BAA"
```

```
# Calculate the credit spread
credit_spread <- BAA - AAA

str(credit_spread)
```

```
## An xts object on 1919-01-01 / 2023-11-01 containing:
##   Data:    double [1259, 1]
##   Columns: BAA
##   Index:   Date [1259] (TZ: "UTC")
##   xts Attributes:
##     $ src    : chr "FRED"
##     $ updated: POSIXct[1:1], format: "2023-12-22 19:52:15"
```

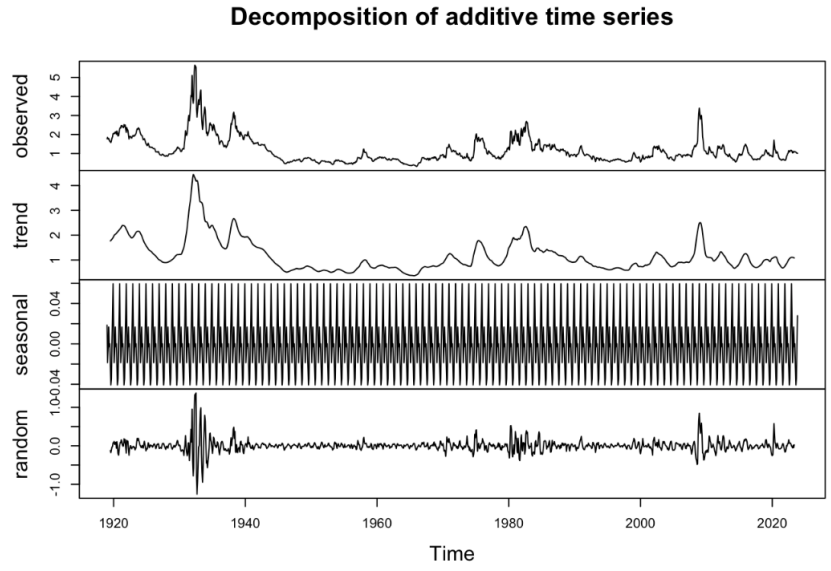
```
# Assuming 'credit_spread' is calculated as the difference between BAA and AAA
credit_spread_ts <- ts(credit_spread, start=c(1919,1),frequency = 12)
```

```
# Plot the credit spread
plot(credit_spread_ts, main = "Monthly Credit Spread", xlab = "Year", ylab = "Spread")
```

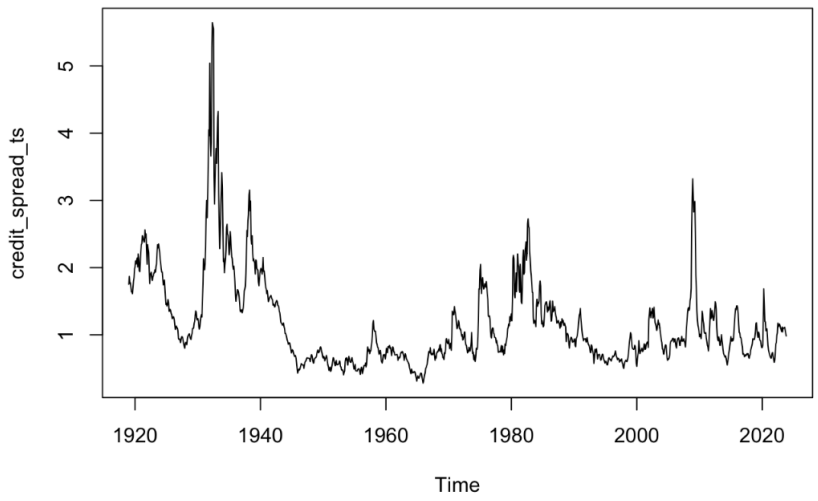


We use the decomposition function to decompose the time series in order to see the seasonal, trend and cycle components. and we would like to eliminates the seasonal part in order to make this times series stationary.

```
# Seasonality check using decomposition
decomposed_spread <- decompose(credit_spread_ts)
plot(decomposed_spread)
```




```
credit_spread_seasonal_removed <- credit_spread_ts - decomposed_spread$seasonal
plot(credit_spread_seasonal_removed)
```



Then we proceed by doing some tests in order to see if our time series is stationary or not.

Firstly, Let's begin with the ADF and PP test:

```
adf_test <- ur.df(credit_spread_seasonal_removed)
summary(adf_test)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression none
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.15145 -0.03531  0.00520  0.04601  1.40519
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## z.lag.1    -0.007500   0.002965  -2.529   0.0116 *
## z.diff.lag  0.216157   0.027540   7.849 8.93e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1424 on 1255 degrees of freedom
## Multiple R-squared:  0.0501, Adjusted R-squared:  0.04859
## F-statistic:  33.1 on 2 and 1255 DF,  p-value: 9.843e-15
##
##
## Value of test-statistic is: -2.529
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau1 -2.58 -1.95 -1.62
```

```
summary((ur.pp(credit_spread_ts)))
```

```
##
## ###################################
## # Phillips-Perron Unit Root Test #
## ###################################
##
## Test regression with intercept
##
##
## Call:
## lm(formula = y ~ y.l1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.11633 -0.04341 -0.01080  0.03180  1.50395
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.027075   0.008224   3.292  0.00102 **
## y.l1        0.976356   0.006063 161.048  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1474 on 1256 degrees of freedom
## Multiple R-squared:  0.9538, Adjusted R-squared:  0.9538
## F-statistic: 2.594e+04 on 1 and 1256 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic, type: Z-alpha  is: -31.4301
##
##          aux. Z statistics
## Z-tau-mu            3.3862
```

```
summary((ur.kpss(credit_spread_ts)))
```

```
##
## ######################
## # KPSS Unit Root Test #
## ######################
##
## Test is of type: mu with 7 lags.
##
## Value of test-statistic is: 2.5978
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

```
summary(ur.za(credit_spread_ts))
```

```
##
## ################################
## # Zivot-Andrews Unit Root Test #
## ################################
##
##
## Call:
## lm(formula = testmat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.09042 -0.04485 -0.01063  0.03314  1.50514
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.997e-02  1.735e-02   4.609 4.46e-06 ***
## y.l1         9.616e-01  7.277e-03 132.146  < 2e-16 ***
## trend        1.707e-05  1.537e-05   1.111 0.266770
## du          -5.682e-02  1.628e-02  -3.490 0.000499 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1467 on 1254 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.9543, Adjusted R-squared:  0.9542
## F-statistic:  8732 on 3 and 1254 DF,  p-value: < 2.2e-16
##
##
## Teststatistic: -5.2772
## Critical values: 0.01= -5.34 0.05= -4.8 0.1= -4.58
##
## Potential break point at position: 232
```

Augmented Dickey-Fuller Test:
The p-value is very small (9.84e-15), indicating strong evidence against the null hypothesis of a unit root. This suggests that the time series is stationary.

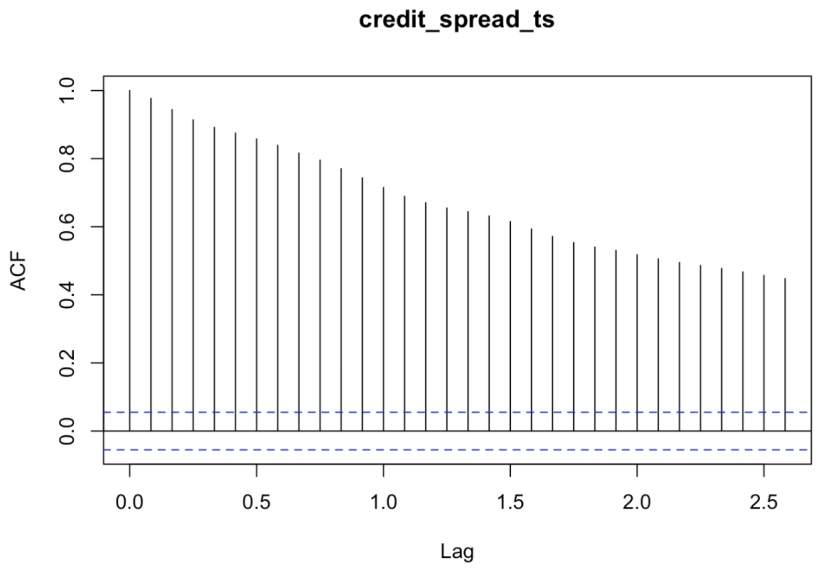
Phillips-Perron Unit Root Test:
The p-value is less than 2.2e-16, which is again strong evidence against the null hypothesis of a unit root, suggesting stationarity.

KPSS Test:
The test statistic is 2.5978. The null hypothesis for the KPSS test is that the series is stationary. This would typically indicate rejection of the null hypothesis, suggesting that the series is not stationary.

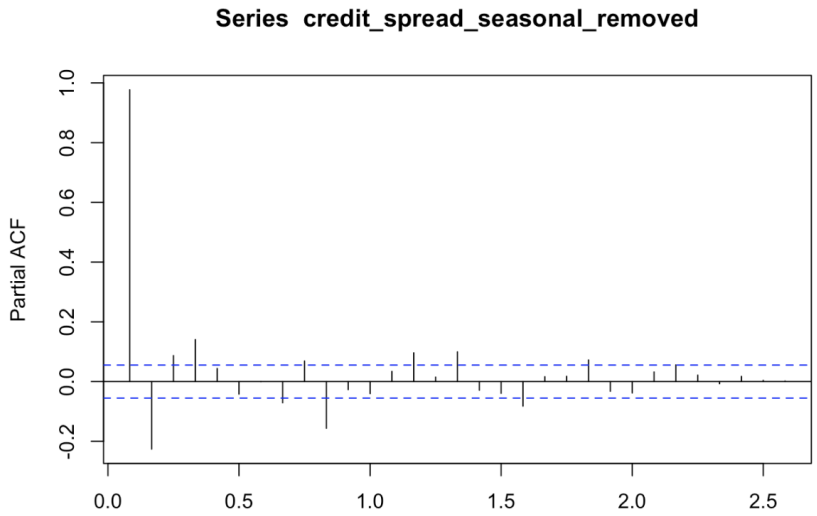
Zivot-Andrews Unit Root Test:
The test statistic(-5.2772), which is less than the critical value (-4.58) at the 1% significance level, which means there is a structural break in the series (point 232). But time series is stationary when this structural break is taken into account.

We could then draw a conclusion of all the tests by saying that there is a structural break which could explain the KPSS non-stationary result and would mean that traditional unit root tests without accounting for such breaks (like the standard ADF test) might be misleading. We then will check the ACF and PACF plot in order to see if our time series is ready to fit and forecast.

```
acf(credit_spread_seasonal_removed)
```




```
pacf(credit_spread_seasonal_removed)
```

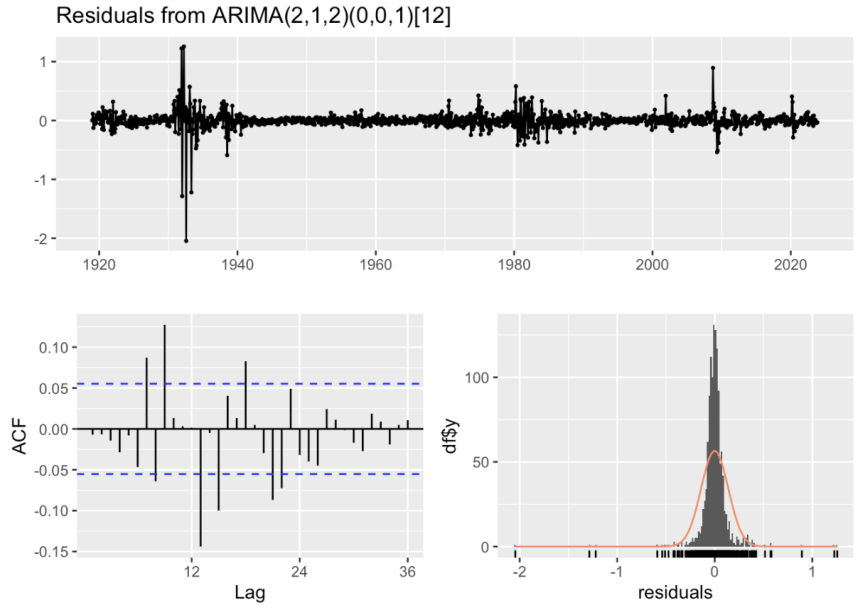


The ACF show a slow decaying => AR(2) and the PACF graph shows that the PACF decay since lag=2 => MA(2)
After identifying the structural break and the seasonality, we could proceed to forecast and fit by the arima model and also check the residual values.

```r
library(forecast)

# Fit a model with the dummy variable
arima_model <- auto.arima(credit_spread_seasonal_removed)
summary(arima_model)
```

```
## Series: credit_spread_seasonal_removed
## ARIMA(2,1,2)(0,0,1)[12]
##
## Coefficients:
##          ar1      ar2      ma1     ma2     sma1
##       0.7215  -0.6396  -0.5092  0.4765  -0.0618
## s.e.  0.0889   0.0828   0.0949  0.1079   0.0315
##
## sigma^2 = 0.01962:  log likelihood = 690.05
## AIC=-1368.1   AICc=-1368.03   BIC=-1337.27
##
## Training set error measures:
##                         ME      RMSE        MAE        MPE     MAPE      MASE
## Training set -0.0005997147 0.1397441 0.07228093 -0.3675266 5.798577 0.2284542
##                      ACF1
## Training set -0.007004666
```

```r
# Check residuals
checkresiduals(arima_model)
```



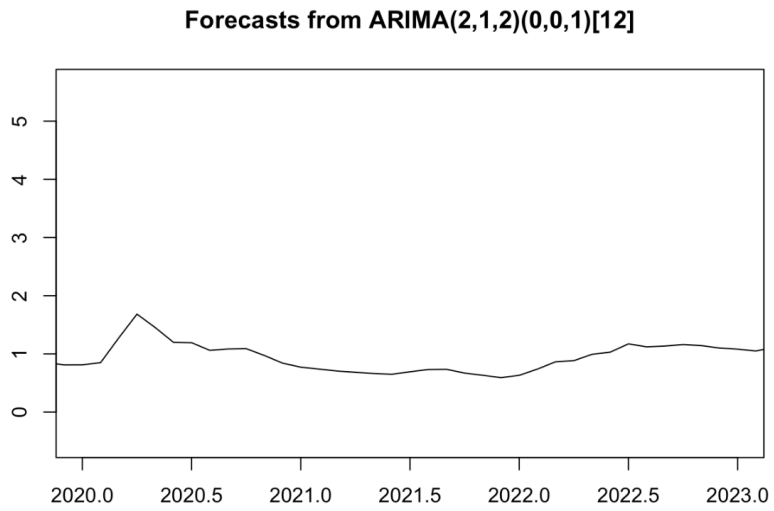

```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,1,2)(0,0,1)[12]
## Q* = 112.21, df = 19, p-value = 3.109e-15
##
## Model df: 5.   Total lags used: 24
```

```r
# Ljung-Box Test
Box.test(residuals(arima_model), type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  residuals(arima_model)
## X-squared = 0.061921, df = 1, p-value = 0.8035
```

We find out that the residuals are normally distributed and the ACF is around 0 except some values maybe linked to the structural breaks. Ljung-box could also valid the fact there's no auto correlation in the residuals as the p-value is $0.8035 > 0.5$.
Here's our predicted values:

```
# Forecast for the next three months
forecast_spread <- forecast(arima_model, h = 30)
plot(forecast_spread, xlim = c(2020,2023))
```




```
print(forecast_spread)
```

```
##          Point Forecast        Lo 80    Hi 80       Lo 95    Hi 95
## Dec 2023      0.9853751  0.805857556 1.164893  0.71082672 1.259923
## Jan 2024      0.9963539  0.714238549 1.278469  0.56489570 1.427812
## Feb 2024      1.0048276  0.649608840 1.360046  0.46156739 1.548088
## Mar 2024      0.9988403  0.595910532 1.401770  0.38261242 1.615068
## Apr 2024      0.9977705  0.559344711 1.436196  0.32725613 1.668285
## May 2024      0.9907021  0.518010520 1.463394  0.26778272 1.713621
## Jun 2024      0.9920506  0.482377148 1.501724  0.21257231 1.771529
## Jul 2024      0.9942158  0.447375569 1.541056  0.15789585 1.830536
## Aug 2024      0.9950884  0.414616852 1.575560  0.10733377 1.882843
## Sep 2024      0.9969588  0.386988770 1.606929  0.06409016 1.929827
## Oct 2024      0.9981644  0.360954791 1.635374  0.02363638 1.972692
## Nov 2024      1.0001220  0.336101865 1.664142 -0.01540915 2.015653
## Dec 2024      1.0006976  0.312861060 1.688534 -0.05125763 2.052653
## Jan 2025      1.0008113  0.290264750 1.711358 -0.08587585 2.087498
## Feb 2025      1.0005250  0.268329092 1.732721 -0.11927202 2.120322
## Mar 2025      1.0002458  0.247087368 1.753404 -0.15161064 2.152102
## Apr 2025      1.0002275  0.226504707 1.773950 -0.18307940 2.183534
## May 2025      1.0003929  0.206478964 1.794307 -0.21379366 2.214579
## Jun 2025      1.0005239  0.186911226 1.814137 -0.24378929 2.244837
## Jul 2025      1.0005126  0.167758237 1.833267 -0.27307531 2.274101
## Aug 2025      1.0004207  0.149024252 1.851817 -0.30167781 2.302519
## Sep 2025      1.0003616  0.130715906 1.870007 -0.32964672 2.330370
## Oct 2025      1.0003777  0.112810633 1.887945 -0.35703901 2.357794
## Nov 2025      1.0004272  0.095264035 1.905590 -0.38390040 2.384755
## Dec 2025      1.0004525  0.078037288 1.922868 -0.41025986 2.411165
## Jan 2026      1.0004392  0.061112680 1.939766 -0.43613675 2.437015
```