# Time Series Analysis of Retail Sales Volume Index in Clothing Stores

***Ismail Sadouki.*** *Email: ismail.sadouki@protonmail.com*

*Under the supervision of* ***Professor Smicha Aitamokhtar***

Time Series Analysis Course.

**École Nationale Supérieure de Statistique et d'Économie Appliquée. (ENSSEA).**

**Abstract**: *This study explores the turnover volume index for retail sales in clothing, focusing on specialized stores over a period of 17 years, from February 1999 to April 2016. The data, sourced from INSEE, is seasonally adjusted and indexed to 2010 (base year = 100). A key aspect of the analysis is to assess the trends and forecast future sales volume using the ARIMA (AutoRegressive Integrated Moving Average) model. The Box-Jenkins methodology was employed to identify the optimal parameters for the ARIMA model, which includes identifying stationarity, model identification, and diagnostic checking. Various tests such as the Augmented Dickey-Fuller (ADF) and Phillips-Perron (PP) tests were utilized to determine the degree of differencing needed. The results show that, despite fluctuations, the clothing retail sales volume index generally hovers around the base year value, with seasonal variations. The study concludes by generating forecasts based on the identified ARIMA model, providing insights into future trends in the retail clothing sector.*

# Introduction

Retail sales data plays a critical role in understanding market dynamics and consumer behavior. The turnover volume index for retail sales of clothing in specialized stores is a key metric used to evaluate the performance of the clothing retail sector. The data analyzed in this study covers a span of **17 years**, from **February 1999 to April 2016**, and is sourced from INSEE. The dataset, seasonally adjusted and indexed to 2010 (base year = 100), offers a granular view of the fluctuations in retail sales volumes, which can be impacted by various economic and seasonal factors.

Understanding and forecasting retail sales volumes is essential for businesses, policymakers, and economists to make informed decisions. To achieve this, time series analysis provides powerful tools to model, analyze, and predict trends in data. One of the most widely used methods for time series forecasting is the **ARIMA model**, which is capable of modeling both the autoregressive (AR) and moving average (MA) components of time series data, along with the necessary differencing (I) to achieve stationarity.

This study utilizes the ARIMA model to identify trends and forecast future turnover volumes in the clothing retail sector. The Box-Jenkins methodology was applied to determine the appropriate model parameters. The analysis includes statistical tests to ensure the stationarity of the data and employs techniques for model validation to ensure the robustness of the findings. By forecasting future trends based on historical data, this study aims to provide valuable insights into the expected behavior of retail clothing sales in the coming periods.

# Literature Review

# Methodology

## ARIMA Modeling

The ARIMA methodology adopted in this study relies on historical data and decomposes it into three components:

- **Autoregressive (AR)**: Incorporates the influence of past values.
- **Integrated (I)**: Refers to differencing the data to achieve stationarity.
- **Moving Average (MA)**: Accounts for past forecast errors.

Together, these elements form the ARIMA model, which is denoted as ARIMA(p, d, q), where:

- *p* is the order of the autoregressive part,
- *d* is the degree of differencing required to make the series stationary,
- *q* is the order of the moving average part.

The longer the data series, the more accurately the model can forecast, as it learns over time.

## The Box-Jenkins methodology

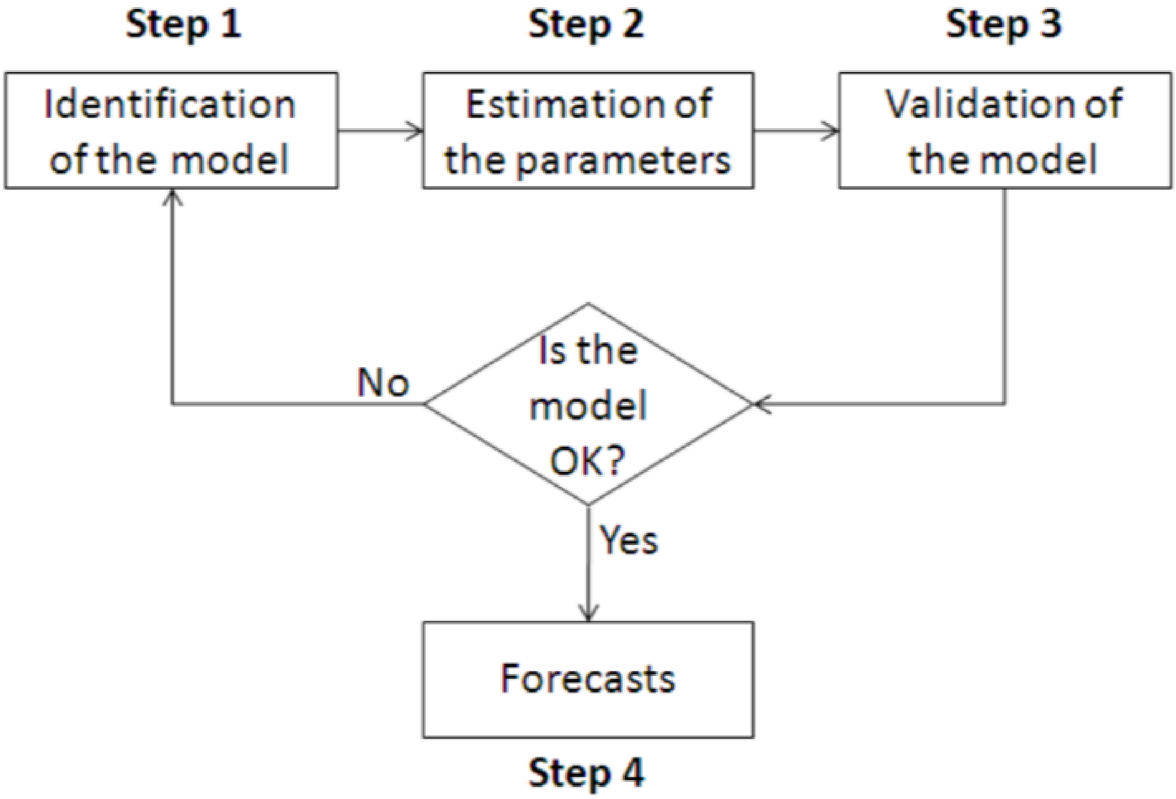


**Figure 3.** The Box-Jenkins methodology.

To identify the appropriate values of *p*, *d*, and *q*, the Box-Jenkins methodology was employed. It includes the following steps:

1. **Model Identification**

2. **Parameter Estimation**
3. **Diagnostic Checking**
4. **Forecast Accuracy Evaluation**

## Model identification

Model identification begins with testing for stationarity to determine the correct differencing order (*d*). This study applied the **Augmented Dickey-Fuller (ADF)** and **Phillips-Perron (PP)** unit root tests.

- The ADF test checks whether the mean and autocorrelation of the series are time-invariant.
- The PP test, unlike the ADF, does not assume a specific structure for serial correlation and heteroskedasticity in the error terms.

Once the data was made stationary, autocorrelation (ACF) and partial autocorrelation (PACF) plots, along with their correlograms, were used to help identify the *p* and *q* parameters. Due to the subjective nature of ACF/PACF interpretation, **Akaike Information Criterion (AIC)** and **Bayesian Information Criterion (BIC)** were also employed to select the best model.

## Model Estimation and Diagnostics

After model selection, parameters were estimated, and several diagnostic tests were applied to ensure the model’s adequacy:

- **AIC, BIC, HQC** for model selection.
- **RMSE, MAE, SER** for forecast accuracy.
- **Durbin-Watson statistic** for autocorrelation.

The **Ljung-Box Q test** was used to check residual autocorrelation. Additionally, **Ljung-Box tests on squared residuals** tested for ARCH effects, while **Shapiro-Wilk** and **Jarque-Bera tests** were used to assess the normality of residuals. If the residuals resemble white noise, it suggests a good model fit.

## Forecasting

Forecasting is the final step in the ARIMA modeling process. It involves generating future values based on the identified model and the historical behavior of the data. For evaluating the forecast accuracy of the ARIMA models, we used a **hold-out sample** approach.

Several metrics were used to assess forecast accuracy:

- **Mean Squared Error (MSE):** The average of the squared differences between the forecasted and actual values.
- **Root Mean Squared Error (RMSE):** The square root of the MSE. It's in the same units as the data, making it easier to interpret.
- **Mean Absolute Error (MAE):** The average of the absolute differences between the forecasted and actual values. It's less sensitive to outliers than MSE/RMSE.
- **Mean Absolute Percentage Error (MAPE):** The average of the absolute percentage differences between the forecasted and actual values. It's scale-independent and easy to understand as a percentage error.
- **Mean Error (ME):** The average of the differences between the forecasted and actual values. It indicates the bias of the forecast (whether the model tends to over- or under-predict).

# Empirical Study

# Data Sources

The dataset used in this study is sourced from **INSEE** (Institut National de la Statistique et des Études Économiques), specifically focusing on the **Turnover Volume Index in Retail Sale of Clothing in Specialized Stores** (NAF Rev. 2, Level: Class, Item 47.71). This data is **seasonally adjusted (SA-WDA)** and uses **2010** as the base year with an index value of **100**.

- **Data Identifier**: 001777097
- **Base Year**: 2010 (Index = 100)
- **Frequency**: Monthly
- **Data Range**: From **February 1999** to **April 2016**

The data represents the volume index of retail sales in the clothing sector, providing insight into the turnover trends over time. The index values are relative to the base year 2010, where an index of **100** represents the turnover volume in that year.

Building an ARlMA model requires an adequate sample size. Box and Jenkins, suggest that about 50 observations is the minimum required number. A large sample size is especially desirable when working with seasonal data.

```
nrow(data)
[1] 208
```

our data contains **208 observations** (monthly data over 17 years) is **definitely suitable** for applying the **Box and Jenkins methodology**

---

# Model Identification

The first step is to identify an ARIMA model determine the order of differencing to make the series stationary. Figure 1 & 2 shows the general view of the data in original pattern. From the below figure it is seen that the series is non stationary in Figure-1 and have a strong upward trend. The rising trend suggests that the mean is nonstationary.
When preparing the data and testing for nonstationarity If the autocorrelation starts high and decline slowly, then the series is non-stationary, and the Box-Jenkins methodology recommend differencing one or more times to get stationarity. Figure-2 shows the estimated ACF falls slowly to zero,this confirms that the mean of the data is probably not
**stationary**.

## Differencing

Differencing is used when the mean of a series is changing over time. Figure-3 after Applying the first difference to our data we get  a stationary series with constant mean and variance over a long period of time. the differenced series appears to have a constant mean.

```
# mean of the data before differencing
mean(data$value, na.rm= TRUE)
[1] 90.67149
# after differencing
mean(data_diff$diff_value, na.rm= TRUE)
[1] 0.1705314
```

```
# Plot the data
plot(data$value, type = "l", main = "Time Series Plot")
```

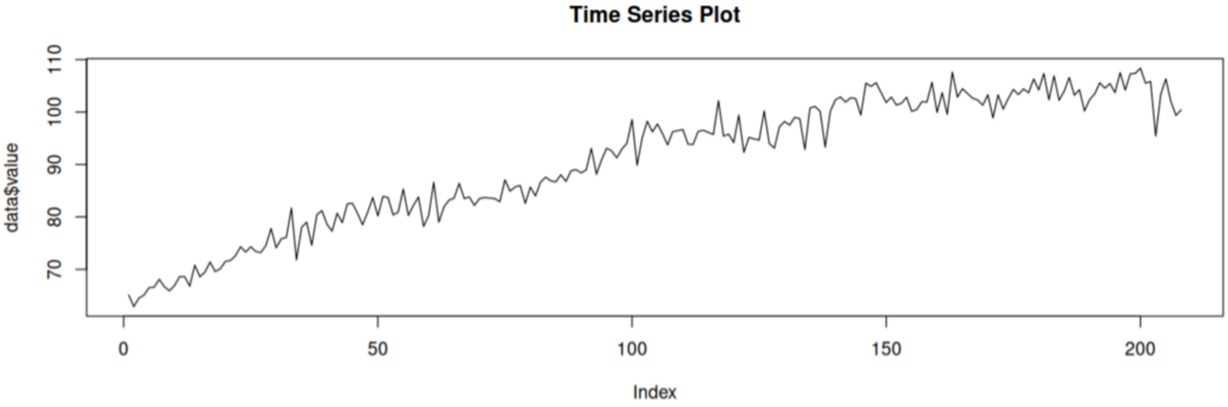


Fig-1: Graphical representation of the data

```
#Plot the estimated ACF
acf(data$value, lag.max = 40)
```

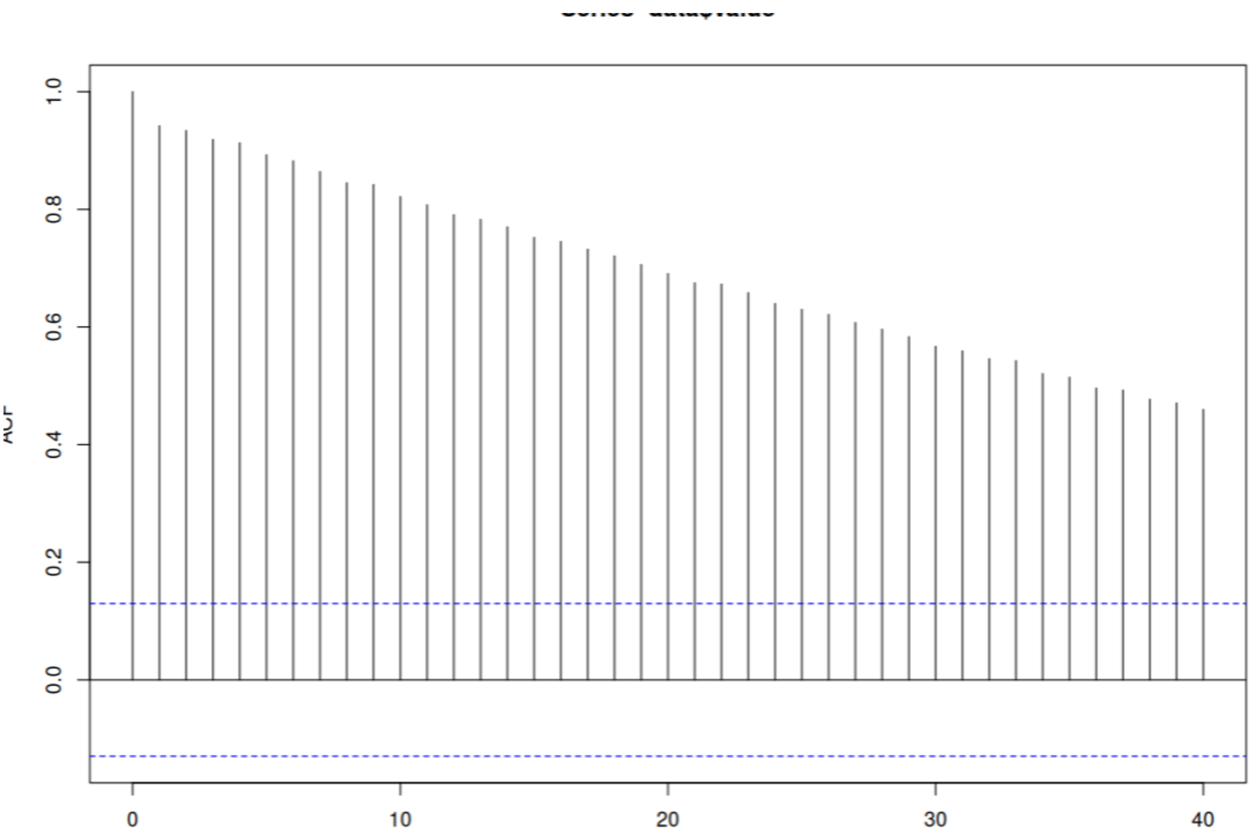


Fig-2: estimated ACF

```
#Appying first difference
data_diff ← data %>%
  mutate(diff_value = difference(value))
data_diff %>%
  autoplot(diff_value)
```

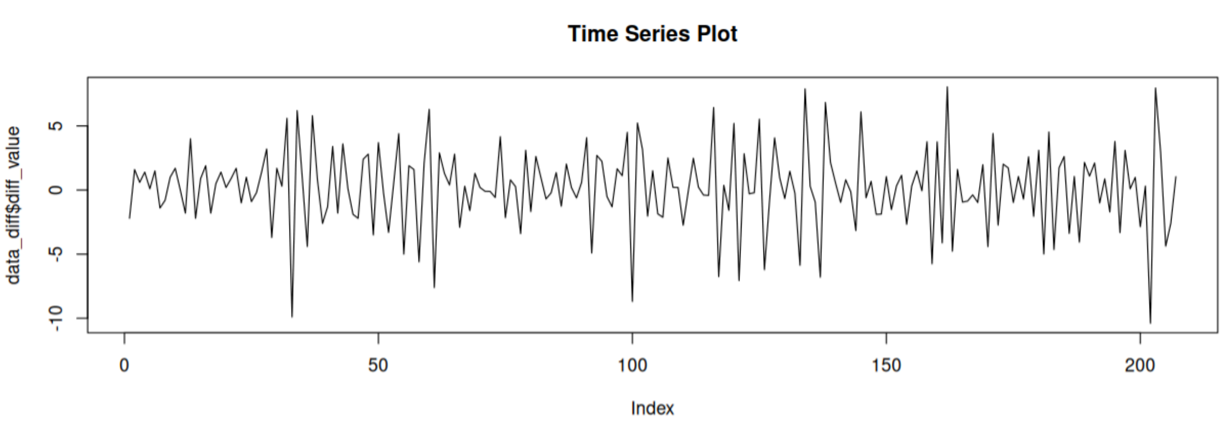


Fig-3: Applying the first difference to our data

A stationary data set will allow predicting our model that the mean and variance will be same in future. The stationarity could be identified according to the t-statistic values in ADF and PP unit root test. In other words, if the t-statistic value exceeds the calculated value, then the series is considered as stationary. In Code-block 1 the model checking was done with ADF and PP Unit Root Test.

```
###########################
# Augmented Dickey-Fuller Test
##########################
library(tseries)
adf.test(data_diff$diff_value)



data:  data_diff$diff_value
Dickey-Fuller = -8.5811, Lag order = 5, p-value = 0.01
alternative hypothesis: stationary


library(urca)

pp_test ← ur.pp(ts_data, type = "Z-tau", model = "constant", lags = "short")
summary(pp_test)
##################################
```

```
# Phillips-Perron Unit Root Test #
####################################
...
...
Value of test-statistic, type: Z-tau  is: -36.5896

        aux. Z statistics
Z-tau-mu          2.0296

Critical values for Z statistics:
                   1pct      5pct     10pct
critical values -3.463311 -2.875588 -2.574186
```

For the **ADF** since **p-value = 0.01** < 0.05 we **reject the null hypothesis** of a unit root.

And for the **PP** test since the test statistic **(-36.5896)** is **much lower** than all critical values, we **reject the null hypothesis** at all significance levels (1%, 5%, and 10%).

from a **mean stationarity** perspective, the series is **stationary now.**

## Nonstationary variance: Levene's Test :

Some realizations have a variance that changes through time. This occurs most frequently with business and economic data covering a long time span, especially when there is a seasonal element in the data.

the null hypothesis for the Levene’s test is that the variances are equal between the groups.

```
n <- nrow(data_diff)
first_half <- data_diff$diff_value[1:(n/2)]
second_half <- data_diff$diff_value[((n/2)+1):n]



group <- factor(c(rep(1, length(first_half)), rep(2, length(second_half))))
values <- c(first_half, second_half)



leveneTest(values ~ group)
```

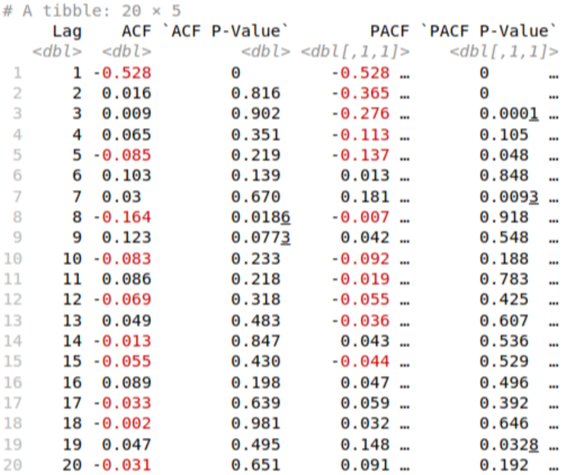
```
Resutls: Levene's Test for Homogeneity of Variance (center = median)
      Df F value Pr(>F)
group   1  1.8254 0.1782
     204
```

Since **p = 0.1782 > 0.05**, we **fail to reject the null hypothesis**.($H_0$ the variances are equal between the groups.)

## ACF/PACF

After the stationarity processing of the original sequence, we need to use ACF and PACF diagrams to identify the model form, and use the information criterion to determine the lag order p and q. Figure 4 & 5 represents the estimated ACF and PACF of the data after the first differencing.

```
# Plot the ACF and PACF
par(mfrow = c(2, 1))  # 1 row, 2 columns
acf(na.omit(data_diff$diff_value), lag.max = 30)
pacf(na.omit(data_diff$diff_value), lag.max = 30)
```



```
# A tibble: 20 × 5
     Lag    ACF `ACF P-Value`         PACF `PACF P-Value`
   <dbl>  <dbl>         <dbl> <dbl[,1,1]>    <dbl[,1,1]>
 1     1 -0.528        0           -0.528 …     0        …
 2     2  0.016        0.816       -0.365 …     0        …
 3     3  0.009        0.902       -0.276 …     0.0001   …
 4     4  0.065        0.351       -0.113 …     0.105    …
 5     5 -0.085        0.219       -0.137 …     0.048    …
 6     6  0.103        0.139        0.013 …     0.848    …
 7     7  0.03         0.670        0.181 …     0.0093   …
 8     8 -0.164        0.0186      -0.007 …     0.918    …
 9     9  0.123        0.0773       0.042 …     0.548    …
10    10 -0.083        0.233       -0.092 …     0.188    …
11    11  0.086        0.218       -0.019 …     0.783    …
12    12 -0.069        0.318       -0.055 …     0.425    …
13    13  0.049        0.483       -0.036 …     0.607    …
14    14 -0.013        0.847        0.043 …     0.536    …
15    15 -0.055        0.430       -0.044 …     0.529    …
16    16  0.089        0.198        0.047 …     0.496    …
17    17 -0.033        0.639        0.059 …     0.392    …
18    18 -0.002        0.981        0.032 …     0.646    …
19    19  0.047        0.495        0.148 …     0.0328   …
20    20 -0.031        0.651        0.091 …     0.192    …
```

Fig-4: estimated ACF/PACF of the data

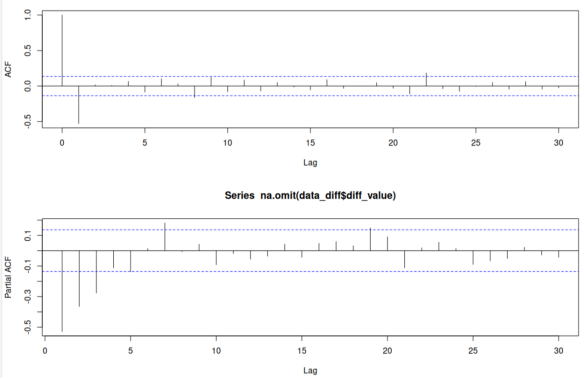


Fig-5: estimated ACF/PACF of the data

*Important Note:in the plot of ACF in R programming language:*

> ***Lag 0*** *corresponds to the autocorrelation of the series with itself — it is always 1.* ***Lag 1*** *is the autocorrelation between $x_t$ and $x_{t-1}$, which is usually what we refer to as the* ***first lag****.*

From the above figures it is noticed that the ACF dies off quickly which has a **strong negative autocorrelation** (below -0.5) at **lag 1**, which is **statistically significant**. Beyond Lag 1, the autocorrelations are within the confidence bounds, meaning they're **not statistically significant**.

This is a classic sign of a **MA(1)** (Moving Average order 1) process.Figure-6 shows Theoretical acf and pacf for MA(1)

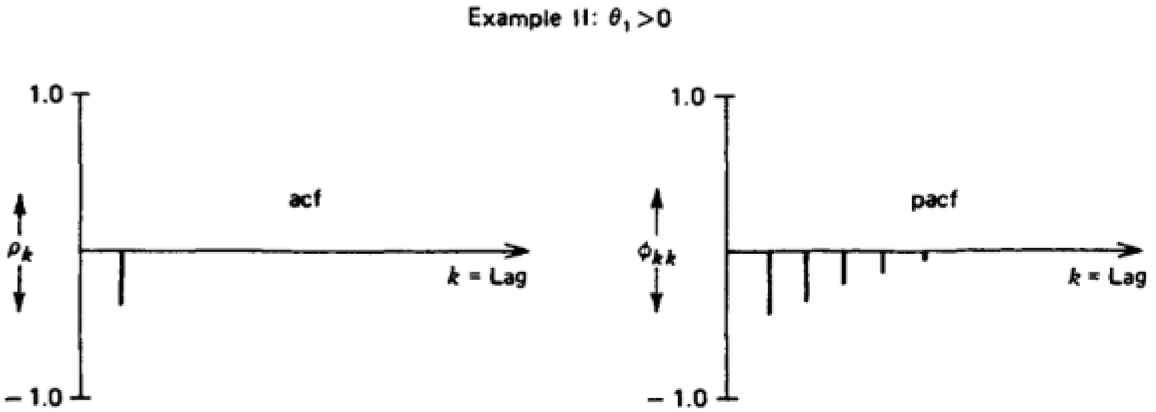


**Figure 6.3** Examples of theoretical acf's and pacf's for two MA(1) processes.

Fig-6: Theoretical acf and pacf for MA(1)

the **PACF** at **Lag 1 to 3** show **significant negative spikes**, especially lag 1. After lag 4, partial autocorrelations become insignificant (within the bounds). This suggests an **AR(3)** or possibly **AR(1)** to **AR(3)** process.

Figure-7 shows the theoretical ACF/PACF for ARMA(1,1) processes

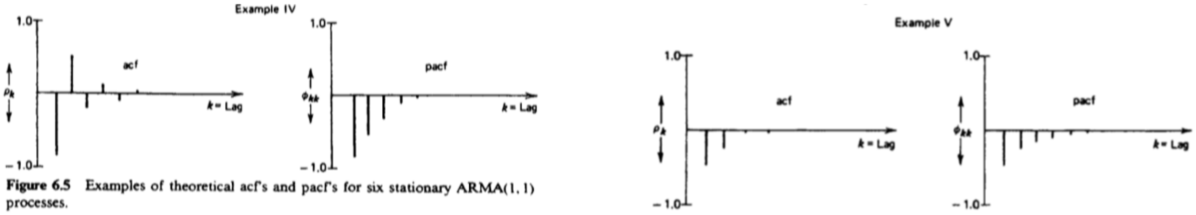


**Figure 6.5** Examples of theoretical acf's and pacf's for six stationary ARMA(1, 1) processes.

Fig-7: theoretical ACF/PACF Arima(1,1)

by comparing the estimated ACF/PACF to the theoretical ACF/PACF of ARMA(1,1) and ARMA(0,1) a candidate ARIMA model might be:

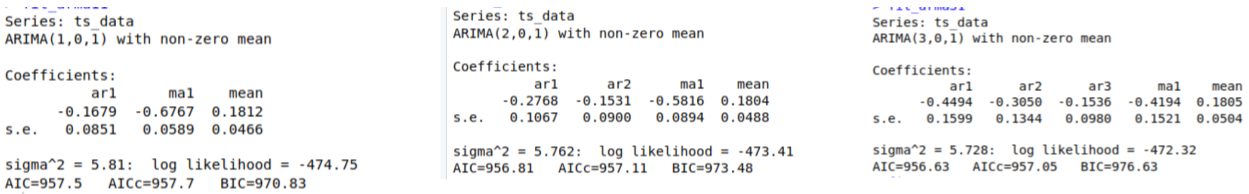
**ARIMA(3,1,1)**, **ARIMA(2,1,1)**, **ARIMA(1,1,1)** (— depending on fit criteria (like AIC/BIC).

The p and q values determined by observing the Autocorrelograms and Partial Autocorrelograms after the first difference of the original sequence are only a rough estimate, and the exact values need to be compared with the nearby values.

# Parameters Estimation

First fitting **ARIMA(3,1,1)**, **ARIMA(2,1,1)**, **ARIMA(1,1,1) models**

```
# Convert to ts object
ts_data ← ts(data_diff$diff_value, start = c(1999, 2), frequency = 12)

# ---- Fit Model: ARMA ----
fit_arma31 ← Arima(ts_data, order = c(3, 0, 1), include.mean = TRUE)
fit_arma21 ← Arima(ts_data, order = c(2, 0, 1), include.mean = TRUE)
fit_arma11 ← Arima(ts_data, order = c(1, 0, 1), include.mean = TRUE)
```



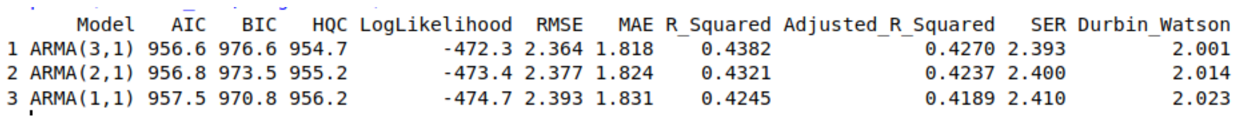
```
Series: ts_data
ARIMA(1,0,1) with non-zero mean

Coefficients:
          ar1      ma1    mean
      -0.1679  -0.6767  0.1812
s.e.   0.0851   0.0589  0.0466

sigma^2 = 5.81:  log likelihood = -474.75
AIC=957.5   AICc=957.7   BIC=970.83
```

```
Series: ts_data
ARIMA(2,0,1) with non-zero mean

Coefficients:
          ar1      ar2      ma1    mean
      -0.2768  -0.1531  -0.5816  0.1804
s.e.   0.1067   0.0900   0.0894  0.0488

sigma^2 = 5.762:  log likelihood = -473.41
AIC=956.81   AICc=957.11   BIC=973.48
```

```
Series: ts_data
ARIMA(3,0,1) with non-zero mean

Coefficients:
          ar1      ar2      ar3      ma1    mean
      -0.4494  -0.3050  -0.1536  -0.4194  0.1805
s.e.   0.1599   0.1344   0.0980   0.1521  0.0504

sigma^2 = 5.728:  log likelihood = -472.32
AIC=956.63   AICc=957.05   BIC=976.63
```

Figure: Fitted ARIMA Models

The penalty function statistic which includes R-square, adjusted R-square, S.E of regression and Durbin-Watson statistic is summarizes in Figure-8



```
      Model   AIC   BIC   HQC LogLikelihood  RMSE   MAE R_Squared Adjusted_R_Squared   SER Durbin_Watson
1 ARMA(3,1) 956.6 976.6 954.7        -472.3 2.364 1.818    0.4382             0.4270 2.393         2.001
2 ARMA(2,1) 956.8 973.5 955.2        -473.4 2.377 1.824    0.4321             0.4237 2.400         2.014
3 ARMA(1,1) 957.5 970.8 956.2        -474.7 2.393 1.831    0.4245             0.4189 2.410         2.023
```

Fig-8 Models Accuarcy

All three models explain the data well, with ARMA(3,1) marginally leading in fit. However, the differences in AIC/BIC/HQC are not large enough to conclusively say one dominates the others.

For RMSE/MAE All models have **very similar predictive accuracy**, with ARMA(3,1) being slightly more accurate. But practically, the gain from increasing model complexity is **minimal**. Prediction errors are small and stable across models.

Around **42–44% of the variance in the stationary time series** is explained by each model. This is **reasonable**, considering we are modeling a differenced series with reduced trend information.

All models have **similar unexplained variability**, with only minor gains in SER when using more lags.

For the Durbin_Watson All models satisfy the assumption of **no residual autocorrelation**, which confirms that the models have adequately captured the time dependence in the data.

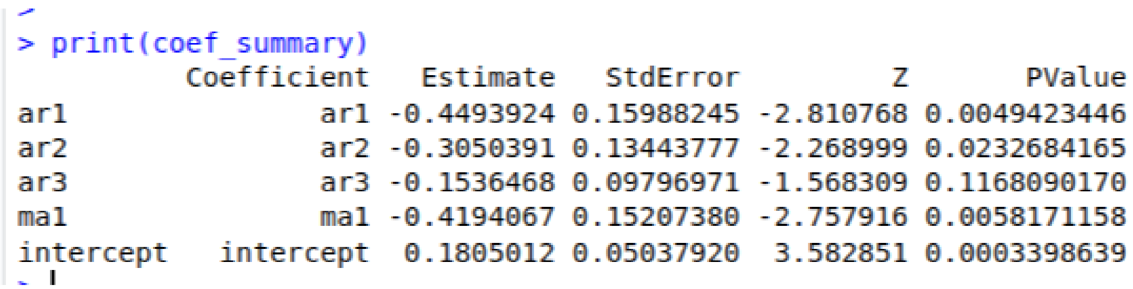
The values of Durbin-Watson statistic are approximately 2.000 strongly suggested that, there is neither positive nor negative first order serial correlation in the series. From regression analysis aspect it also detects the absence of autocorrelation at the first lag in the preceding error terms.

## have we found a good model?

### Coefficient quality: statistical significance.

```
# Summary of the coefficients of each model
# Repeat this code for each model

# Extract coefficients and standard errors
coefs ← coef(fit_arma31)
se ← sqrt(diag(fit_arma31$var.coef))

# Compute z-values and p-values
z_values ← coefs / se
p_values ← 2 * (1 - pnorm(abs(z_values)))

# Combine into a data frame
coef_summary ← data.frame(
  Coefficient = names(coefs),
  Estimate = coefs,
  StdError = se,
  Z = z_values,
  PValue = p_values
)

print(coef_summary)
```

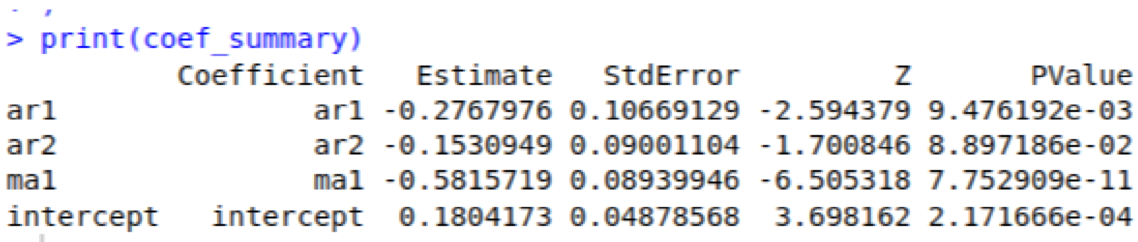
▼ For **ARMA(3,1)**



```
> print(coef_summary)
          Coefficient   Estimate   StdError         Z       PValue
ar1               ar1 -0.4493924 0.15988245 -2.810768 0.0049423446
ar2               ar2 -0.3050391 0.13443777 -2.268999 0.0232684165
ar3               ar3 -0.1536468 0.09796971 -1.568309 0.1168090170
ma1               ma1 -0.4194067 0.15207380 -2.757916 0.0058171158
intercept   intercept  0.1805012 0.05037920  3.582851 0.0003398639
```

`ar3` **is not statistically significant** (p ≈ 0.12), this term might be unnecessary, and we could consider the other simpler models like **ARMA(2,1)** without losing much predictive power. We may Consider
**comparing the ARMA(3,1) vs ARMA(2,1)** using AIC/BIC **and** this statistical significance. If removing `ar3` simplifies the model and doesn't degrade performance.

▼ For ARMA(2,1)



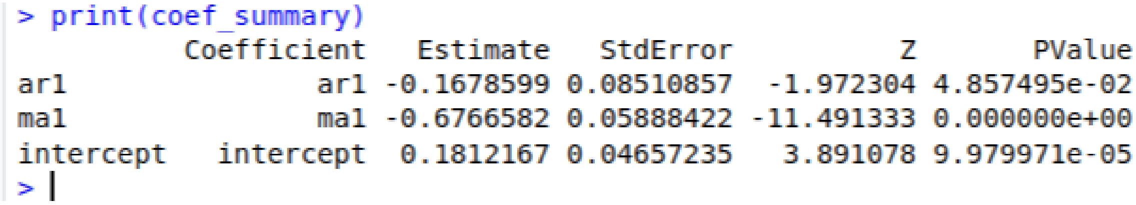
```
> print(coef_summary)
          Coefficient   Estimate   StdError         Z       PValue
ar1               ar1 -0.2767976 0.10669129 -2.594379 9.476192e-03
ar2               ar2 -0.1530949 0.09001104 -1.700846 8.897186e-02
ma1               ma1 -0.5815719 0.08939946 -6.505318 7.752909e-11
intercept   intercept  0.1804173 0.04878568  3.698162 2.171666e-04
```

Nearly all coefficients are statistically significant.

Compared to ARMA(3,1) Both models have similar **log-likelihoods** and **AIC/BIC** values. where **ARMA(3,1)** has an extra parameter ( `ar3` ) that is **not significant (p = 0.1168)**. also

**ARMA(2,1)** has fewer parameters and **all except ar2 are significant**, with **ar2 only marginally insignificant (p ≈ 0.089)**.

▼ For **ARMA(1,1)**



```
> print(coef_summary)
          Coefficient   Estimate   StdError          Z       PValue
ar1               ar1 -0.1678599 0.08510857  -1.972304 4.857495e-02
ma1               ma1 -0.6766582 0.05888422 -11.491333 0.000000e+00
intercept   intercept  0.1812167 0.04657235   3.891078 9.979971e-05
> |
```

**ARMA(1,1)** is the Smallest, simplest model. where All parameters significant. with Slightly worse AIC/BIC.

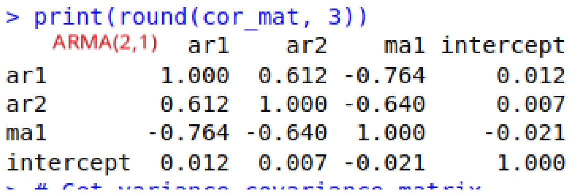
We conclude that **ARMA(2,1)** offers the best trade-off between model fit and parameter significance. If interpretability and parsimony are critical, **ARMA(1,1)** is also acceptable.

From now we are gonna focus only on ARMA(2,1) and ARMA(1,1)

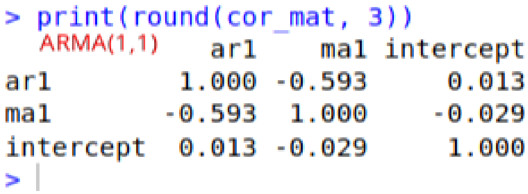
## Coefficient quality: correlation matrix.

We cannot avoid getting estimates that are correlated, but very high correlations between estimated coefficients suggest that the estimates may be of poor quality.

```
vcov_mat <- vcov(fit_arma11)
cor_mat <- cov2cor(vcov_mat)
print(round(cor_mat, 3))
```



```
> print(round(cor_mat, 3))
  ARMA(2,1)    ar1    ar2    ma1 intercept
ar1          1.000  0.612 -0.764     0.012
ar2          0.612  1.000 -0.640     0.007
ma1         -0.764 -0.640  1.000    -0.021
intercept    0.012  0.007 -0.021     1.000
> # Get variance covariance matrix
```



```
> print(round(cor_mat, 3))
  ARMA(1,1)    ar1    ma1 intercept
ar1          1.000 -0.593     0.013
ma1         -0.593  1.000    -0.029
intercept    0.013 -0.029     1.000
> |
```

As a practical rule, one should suspect that the estimates are somewhat unstable when the absolute correlation coefficient between any two estimated ARIMA coefficients is 0.9 or larger. When this happens we should consider whether some alternative models are justified by the estimated acf and pacf. One of these alternatives might provide an adequate fit with more stable parameter estimates [1]. Therefore, in our case the estimated models are satisfactory in this regard.

## Checking coefficients for stationarity and invertibility.

The invertibility requirement applies only to the moving-average part of the models. The requirement for an $ARMA(1,1), ARMA(2,1), ARMA(3,1)$ is the same as that for an MA(1): $|\theta| < 1$, the estimated coefficient for our models $\theta = -0.6767, -0.5816, -0.4194$ clearly satisfies this requirement.

For checking **Stationarity of the coefficients**

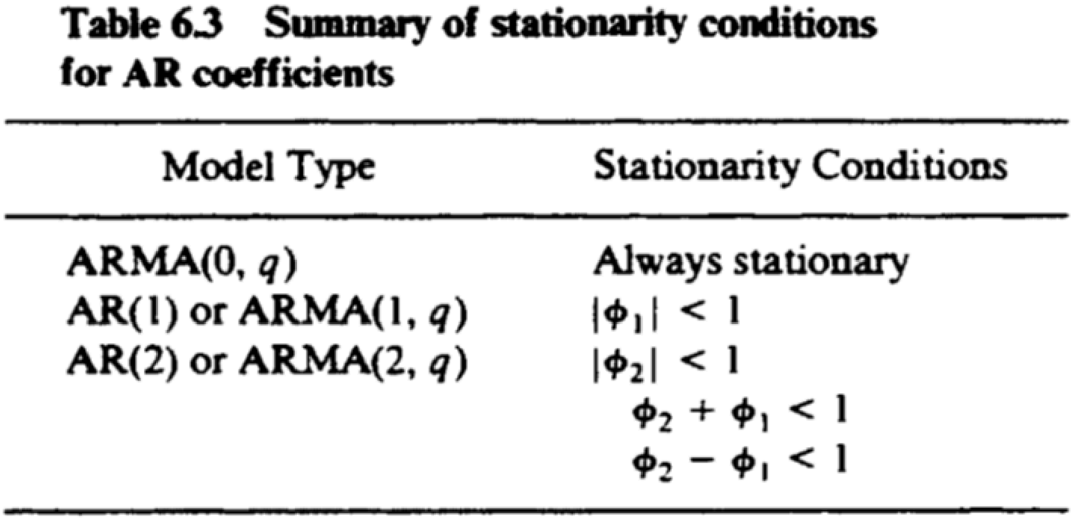


**Table 6.3 Summary of stationarity conditions for AR coefficients**

| Model Type | Stationarity Conditions |
|---|---|
| ARMA(0, $q$) | Always stationary |
| AR(1) or ARMA(1, $q$) | $\lvert\phi_1\rvert < 1$ |
| AR(2) or ARMA(2, $q$) | $\lvert\phi_2\rvert < 1$ |
| | $\phi_2 + \phi_1 < 1$ |
| | $\phi_2 - \phi_1 < 1$ |

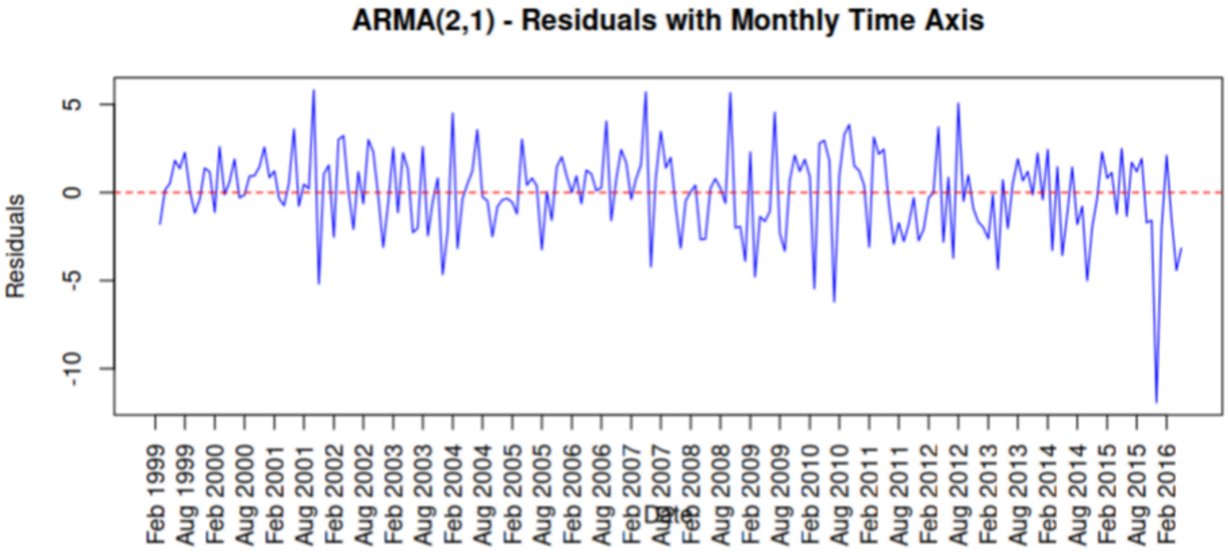
For the ARMA(1,1) we have $|\theta = -0.1679| < 1$ hence the coefficient is stationary. and for the ARMA(2,1) we have $|\theta_2 = -0.1531| < 1, \theta_2 + \theta_1 = -0.4299$ and $\theta_2 - \theta_1 = 0.1237$. hence the stationarity checked.

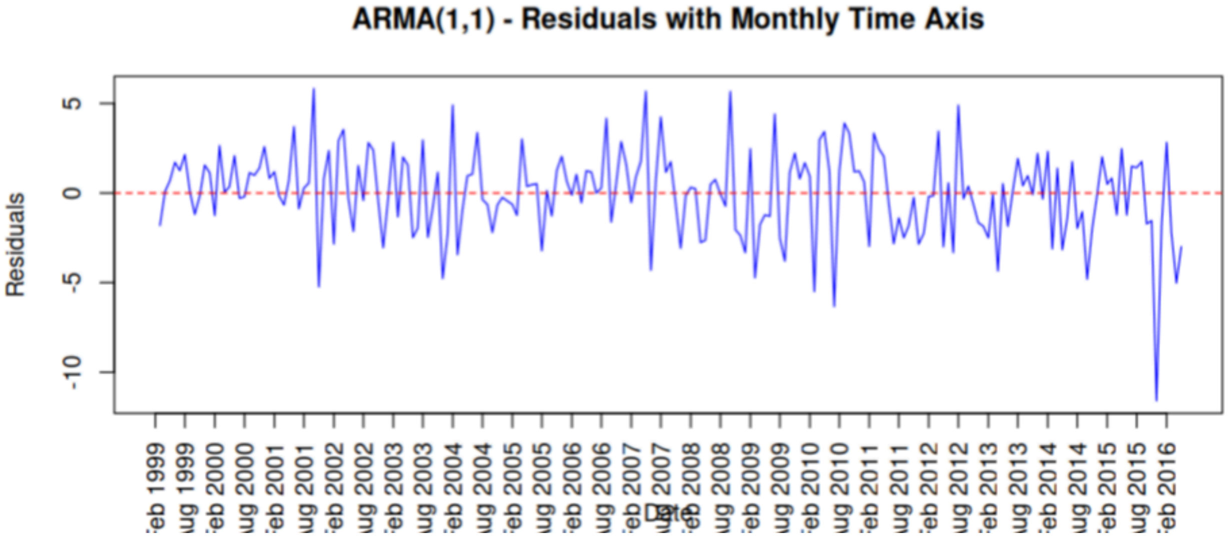
# Diagnostic Checking

In this stage we determine if a model is statistically adequate. In particular, we test if the random shocks are independent. If this assumption is not satisfied, there is an autocorrelation pattern in the original series that has not been explained by the ARIMA model. Our goal, however, is to build a model that fully explains any autocorrelation in the original series [1].

## Are the random shocks independent?

```
plot(residuals(fit_arma21), main = "Residuals - ARMA(2,1)", ylab = "Residuals")
plot(residuals(fit_arma11), main = "Residuals - ARMA(1,1)", ylab = "Residuals")
```

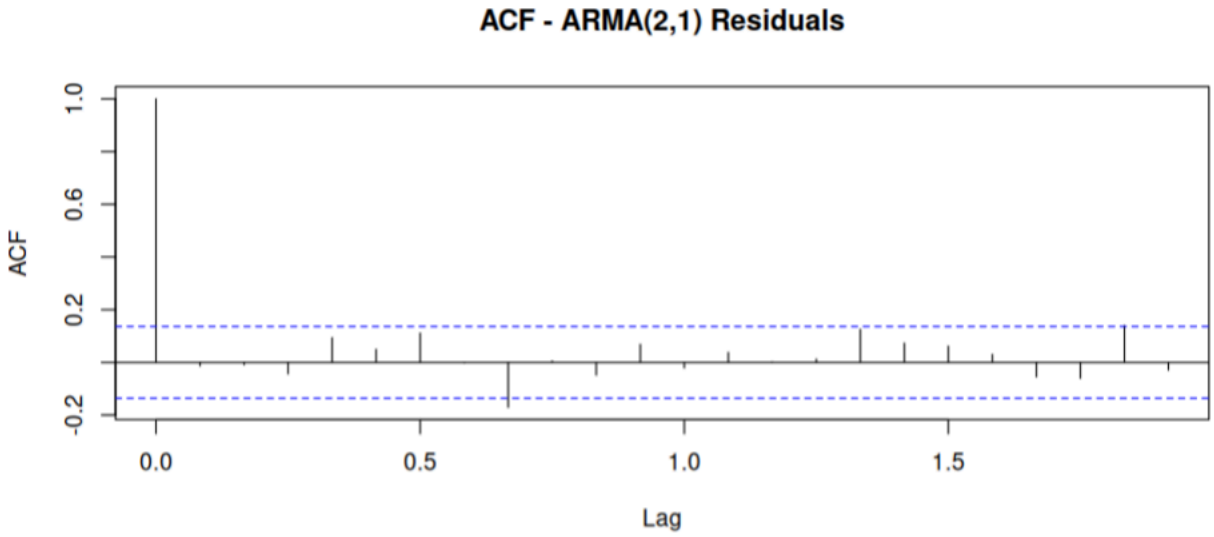


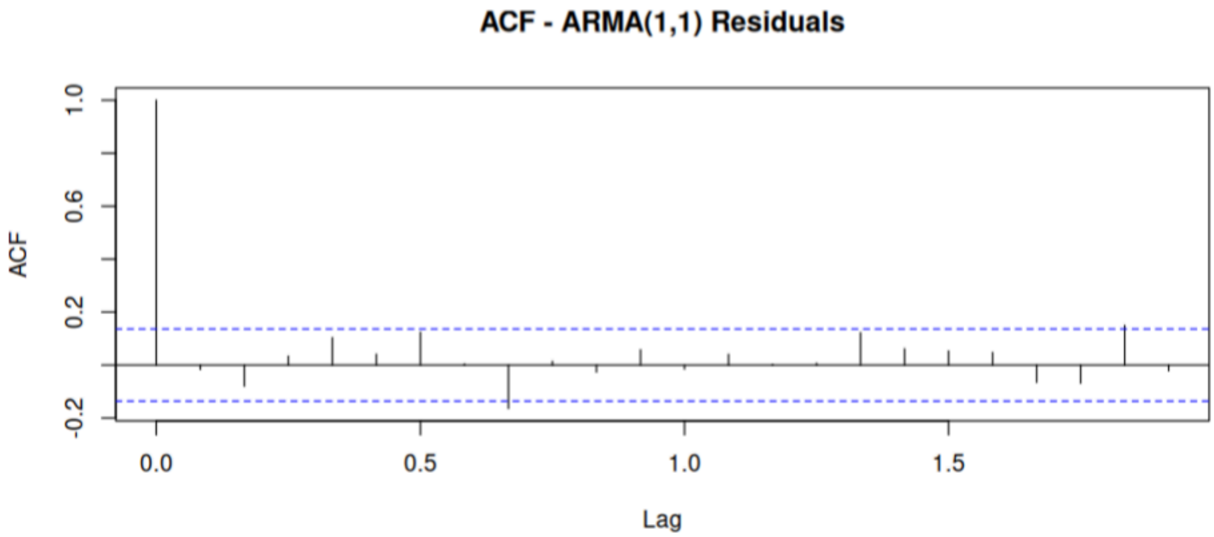


Inspection does not suggest that the variance is changing systematically over time.
also there may be unusual event between the year 2015-2016 that may need further investigations. we must expect some residuals to be large just by chance. But they might also represent data that were incorrectly recorded, or perturbations to the data caused by identifiable exogenous events.

To test the hypothesis that the random shocks are independent we construct a residual acf. This acf is like any estimated acf except we construct it using the estimation residuals $\hat{a}_t$, instead of the realization $z_t$

```
acf(na.omit(residuals(fit_arma21)), main = "ACF - ARMA(2,1) Residuals")
acf(na.omit(residuals(fit_arma11)), main = "ACF - ARMA(1,1) Residuals")
```





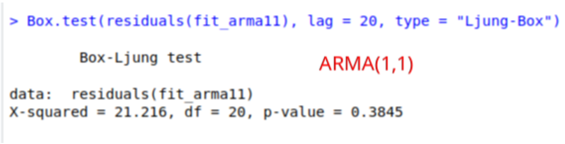
from the acf and pacf of the residuals we conclude that the random shocks are independent. (ALL the p-values are larger than 0.05)

## Ljung-Box test

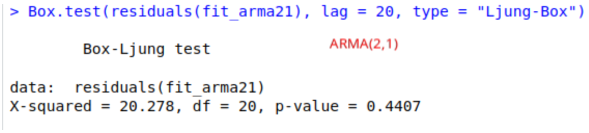
Another way to deal with potentially underestimated residual acf t-values is to test the residual autocorrelations as a set rather than individually. An approximate chi-squared statistic (the

Ljung-Box statistic) is available for this test.

```
Box.test(residuals(fit_arma21), lag = 20, type = "Ljung-Box")
Box.test(residuals(fit_arma11), lag = 20, type = "Ljung-Box")
```



```
> Box.test(residuals(fit_arma11), lag = 20, type = "Ljung-Box")

	Box-Ljung test

data:  residuals(fit_arma11)
X-squared = 21.216, df = 20, p-value = 0.3845
```



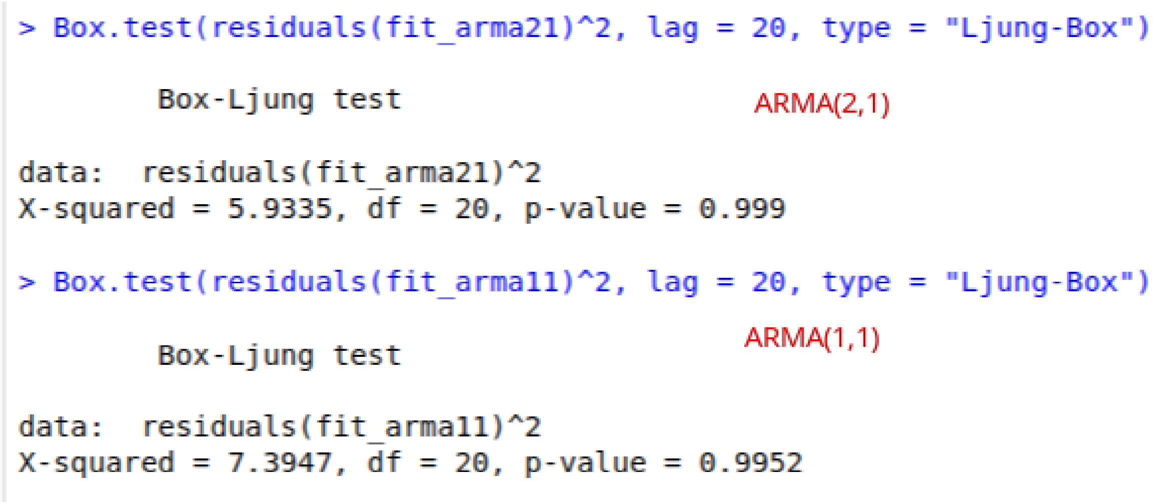
```
> Box.test(residuals(fit_arma21), lag = 20, type = "Ljung-Box")

	Box-Ljung test

data:  residuals(fit_arma21)
X-squared = 20.278, df = 20, p-value = 0.4407
```

**p-value > 0.05** For both the models ARMA(2,1) and ARMA(1,1). we Fail to reject the null hypothesis that residuals are uncorrelated (white noise). This suggests the models capture the time dependence in the data sufficiently well, at least up to lag 20.

## Ljung-Box test on squared residuals (ARCH effects (non-constant variance))

**Null Hypothesis**: No autocorrelation in the **squared residuals**, i.e., no ARCH effects.

```
Box.test(residuals(fit_arma21)^2, lag = 20, type = "Ljung-Box")
Box.test(residuals(fit_arma11)^2, lag = 20, type = "Ljung-Box")
```



```
> Box.test(residuals(fit_arma21)^2, lag = 20, type = "Ljung-Box")

	Box-Ljung test

data:  residuals(fit_arma21)^2
X-squared = 5.9335, df = 20, p-value = 0.999

> Box.test(residuals(fit_arma11)^2, lag = 20, type = "Ljung-Box")

	Box-Ljung test

data:  residuals(fit_arma11)^2
X-squared = 7.3947, df = 20, p-value = 0.9952
```

Since the **p-value is extremely high**, we **fail to reject** the null hypothesis.

To assess the adequacy of the fitted ARMA model, a histogram of the residuals was examined (see Figure 9&10). This diagnostic tool provides insight into the distributional characteristics of the model errors.

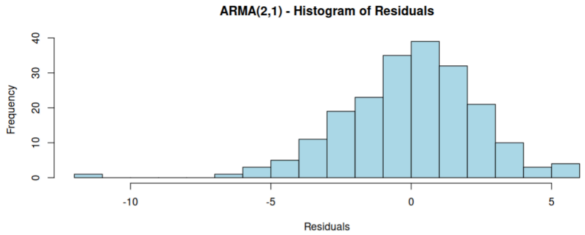


Fig-9: Histogram of Residuals for ARMA(2,1)

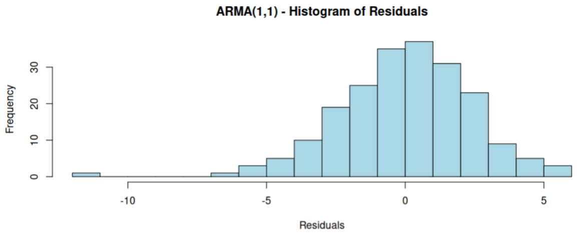
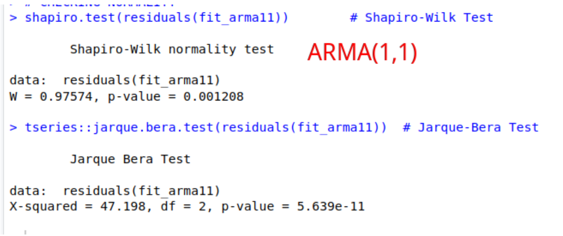


Fig-10: Histogram of Residuals for ARMA(1,1)

The histogram of residuals is approximately symmetric and centered around zero, indicating that the model errors are unbiased. However, a slight left skew and a few large negative residuals suggest occasional underestimation. Overall, we need to check for the normality of the residuals.

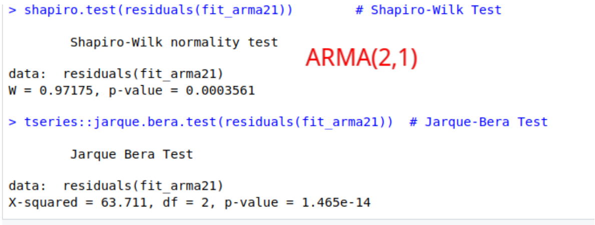
## Shapiro-Wilk and Jarque-Bera Normality Test

```
shapiro.test(residuals(fit_arma11))       # Shapiro-Wilk Test
tseries::jarque.bera.test(residuals(fit_arma11))  # Jarque-Bera Test
```



```
> shapiro.test(residuals(fit_arma11))        # Shapiro-Wilk Test

        Shapiro-Wilk normality test

data:  residuals(fit_arma11)
W = 0.97574, p-value = 0.001208

> tseries::jarque.bera.test(residuals(fit_arma11))  # Jarque-Bera Test

        Jarque Bera Test

data:  residuals(fit_arma11)
X-squared = 47.198, df = 2, p-value = 5.639e-11
```

ARMA(1,1)



```
> shapiro.test(residuals(fit_arma21))        # Shapiro-Wilk Test

        Shapiro-Wilk normality test

data:  residuals(fit_arma21)
W = 0.97175, p-value = 0.0003561

> tseries::jarque.bera.test(residuals(fit_arma21))  # Jarque-Bera Test

        Jarque Bera Test

data:  residuals(fit_arma21)
X-squared = 63.711, df = 2, p-value = 1.465e-14
```

ARMA(2,1)

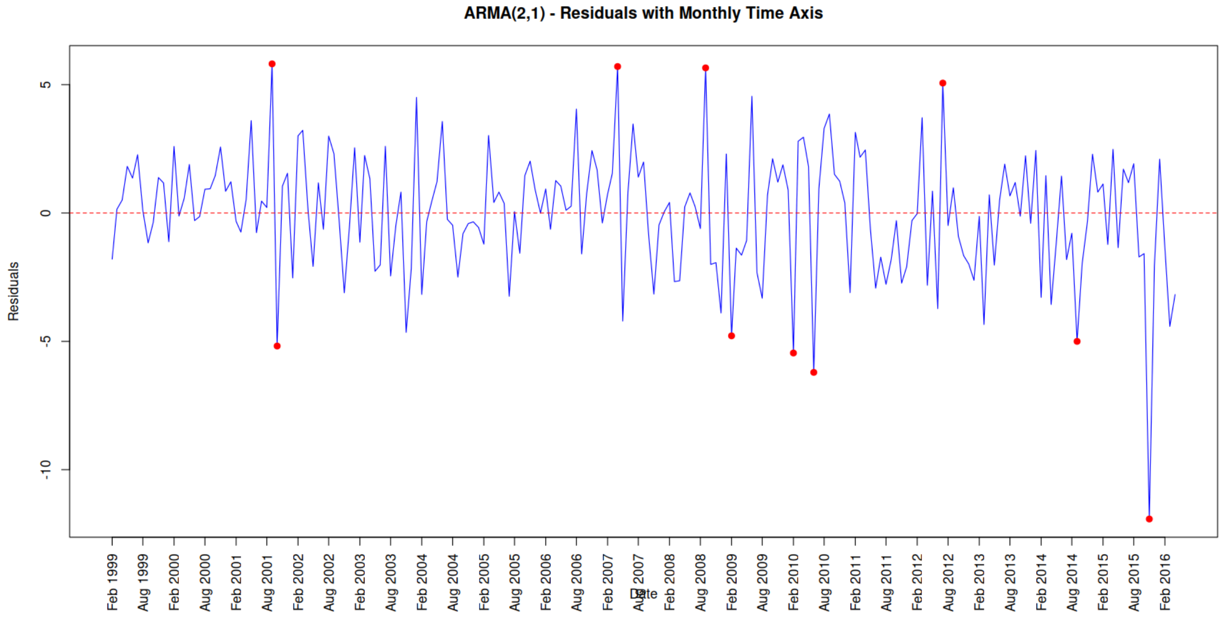
The p-value for both tests is less than 0.05, which suggests that the residuals **do not follow a normal distribution**.

Both the **Shapiro-Wilk** and **Jarque-Bera** tests strongly suggest that the residuals of the ARIMA(1,1,1)&ARIMA(2,1,1) modelS deviate from normality. This is a critical observation, as normal residuals are often assumed in time series modeling. The non-normality of the residuals could indicate that the model has not fully captured the underlying data structure or that there are significant outliers influencing the model's fit.

The non-normality of residuals could also suggest that further diagnostics or model improvements are needed. In particular, **outliers** may be driving the non-normality. One potential approach to address this issue is to **impute the outliers** and refit the model to see if the residuals improve in terms of normality.

# Model Refinement

## Outlier Detection and Imputation

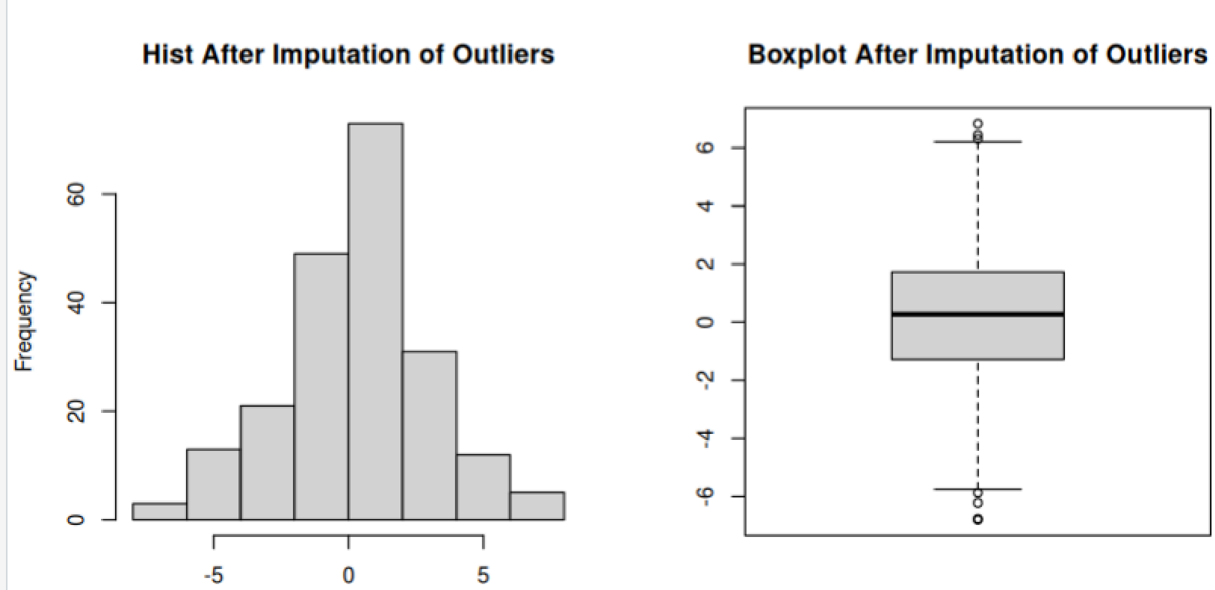


Outlier Detection

In this step, outliers in the `data` series were identified (above figure) using two methods: **Z-scores** and the **Interquartile Range (IQR)**. Outliers were then imputed to mitigate their influence on the model. Below is the detailed code:

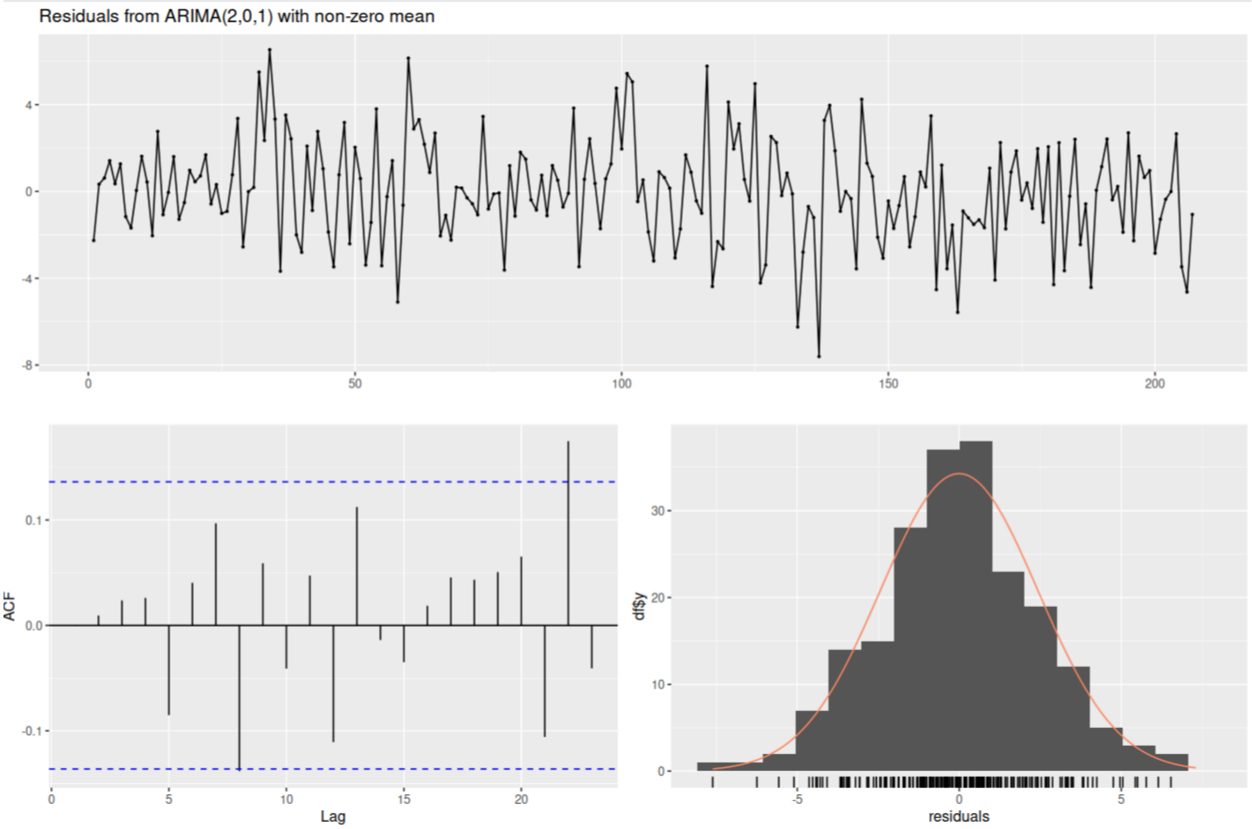
```
# Calculate the Z-scores of data_diff$diff_value
z_scores ← scale(data_diff$diff_value)

# Identify outliers (Z-scores greater than 3 or less than -3)
```

```
outliers ← which(abs(z_scores) > 3)

# Print the indices of outliers
print(outliers)



# Calculate the IQR for data_diff$diff_value
Q1 ← quantile(data_diff$diff_value, 0.25)
Q3 ← quantile(data_diff$diff_value, 0.75)
IQR_value ← IQR(data_diff$diff_value)

# Identify outliers
outliers_iqr ← which(data_diff$diff_value < (Q1 - 1.5 * IQR_value) | data_diff$diff_v



# Impute outliers (from both Z-score and IQR methods) with the mean
data_diff$diff_value[outliers] ← mean(data_diff$diff_value, na.rm = TRUE)
data_diff$diff_value[outliers_iqr] ← mean(data_diff$diff_value, na.rm = TRUE)
```

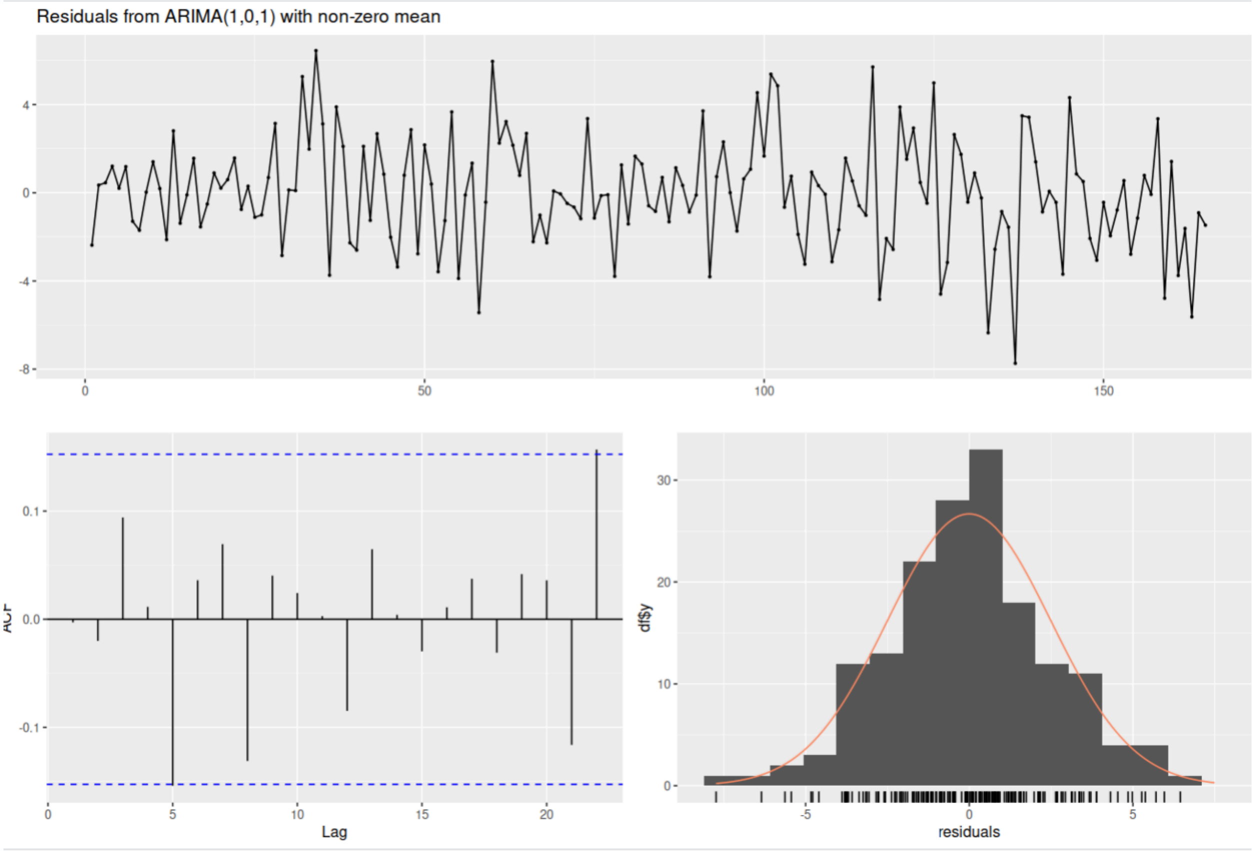


Histogram and boxplot after imputing the outliers

Fitting the models again and checking for residuals:



Residuals From ARIMA(2,1,1) after outlier imputation



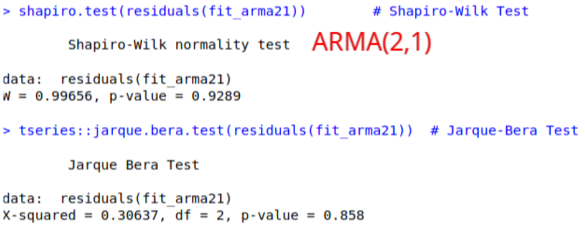
Residuals From ARIMA(1,1,1) after outlier imputation

Applying the same statistical tests again the results from the Box-Ljung and ARCH effects tests suggest that both the new ARIMA models adequately capture the time series dynamics. There is no significant autocorrelation in the residuals, indicating that the models are well-specified and that the residuals behave like white noise. Additionally, the absence of significant ARCH effects points to constant variance in the residuals, implying that there is no need for further volatility modeling.

The results from the normality tests (Shapiro-Wilk and Jarque-Bera tests) for the residuals of both ARIMA models indicate that the residuals are approximately normally distributed.

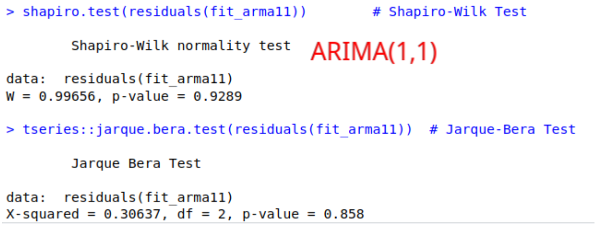
For both the ARIMA(1,1,1) and ARIMA(2,1,1) models, the Shapiro-Wilk test yields p-values of 0.9289, indicating that we fail to reject the null hypothesis of normality. Similarly, the Jarque-Bera test produces p-values of 0.858, further suggesting that the residuals are consistent with normal distribution.

These results imply that the residuals of both models follow a normal distribution, which is an important assumption for the validity of the models.



```
> shapiro.test(residuals(fit_arma21))          # Shapiro-Wilk Test

        Shapiro-Wilk normality test

data:  residuals(fit_arma21)
W = 0.99656, p-value = 0.9289

> tseries::jarque.bera.test(residuals(fit_arma21))  # Jarque-Bera Test

        Jarque Bera Test

data:  residuals(fit_arma21)
X-squared = 0.30637, df = 2, p-value = 0.858
```

Shapiro-Wilk and Jarque-Bera tests ARMA(2,1)



```
> shapiro.test(residuals(fit_arma11))          # Shapiro-Wilk Test

        Shapiro-Wilk normality test

data:  residuals(fit_arma11)
W = 0.99656, p-value = 0.9289

> tseries::jarque.bera.test(residuals(fit_arma11))  # Jarque-Bera Test

        Jarque Bera Test

data:  residuals(fit_arma11)
X-squared = 0.30637, df = 2, p-value = 0.858
```

Shapiro-Wilk and Jarque-Bera tests ARMA(1,1)

These tests confirm that the residuals behave like **white noise** , meaning the model is appropriately specified and no further adjustments are necessary.

# FORECASTING STAGE

The 24-period forecasts for the differenced Turnover Volume Index generated by both ARMA(1,1) and ARMA(2,1) were showed in figure 11&12.

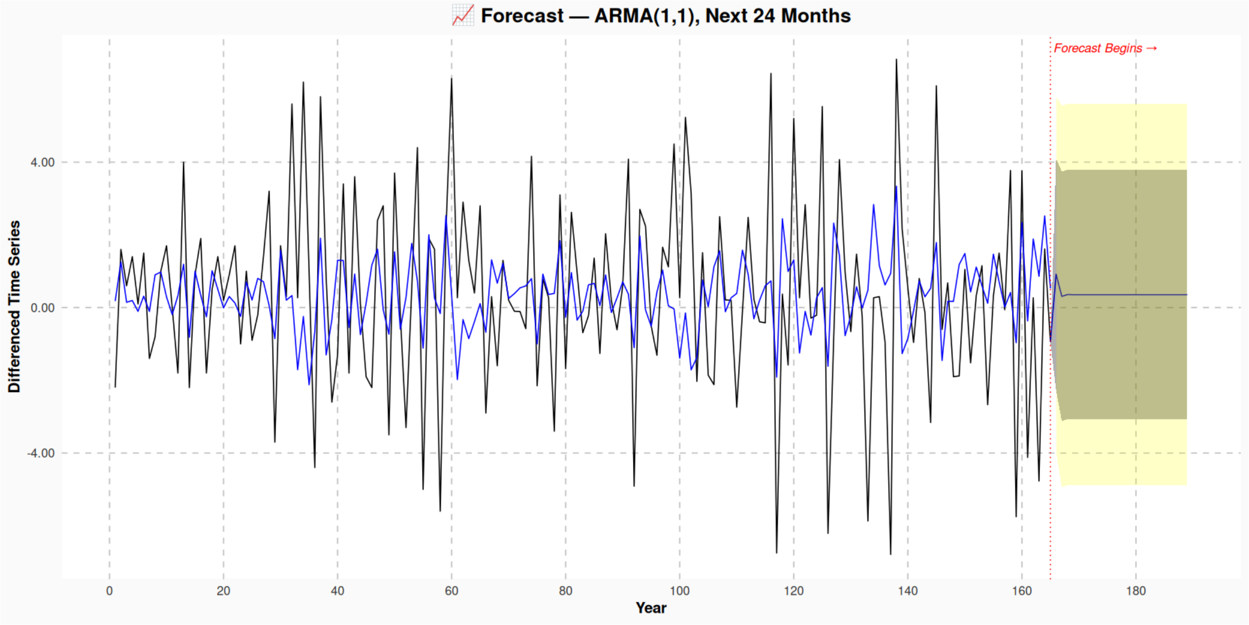


Fig-11: Forcast for 24-period ARMA(1,1)

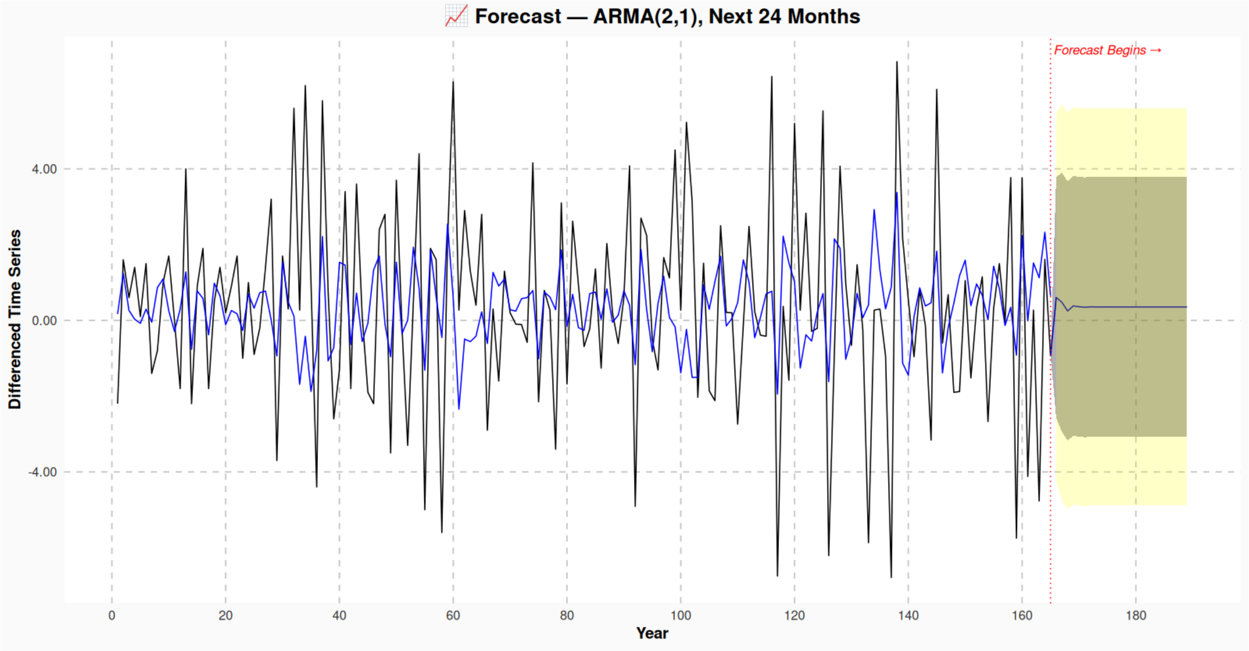


Fig-11: Forcast for 24-period ARMA(2,1)

The models exhibit a rapid stabilization around a level of approximately 0.2. The projected trajectories are nearly identical between the two models, suggesting that the inclusion of an additional autoregressive term in ARMA(2,1) does not yield a significantly different forecast. Furthermore, the confidence intervals for both models widen similarly over the forecast horizon, indicating comparable increases in uncertainty. The models suggest that the rate of change of the original Turnover Volume Index will stabilize, and the similar performance supports the principle of parsimony, potentially favoring the simpler ARMA(1,1) model for forecasting the differenced series.

## Forecast Accuracy

The forecast accuracy of the fitted ARIMA(1,1) and ARIMA(2,1) models was evaluated using a hold-out sample approach. This method assesses the models' ability to predict unseen data, providing insights into their real-world forecasting performance.

The original time series data of the Turnover Volume Index (from February 1999 to April 2016) was divided into two subsets:

- **Training Set:** The data from the beginning of the series up to April 2015 was used to fit the ARIMA(1,1) and ARIMA(2,1) models. This comprised the majority of the dataset.

- **Testing Set (Hold-Out Sample):** The subsequent 12 months of data, from May 2015 to April 2016, were held back and used as the testing set. This represents data the models had not seen during the fitting process.

```
n_total ← length(insee_data$value)
n_test ← 12 # Hold out the last 12 months for testing
n_train ← n_total - n_test

train_data ← insee_data$value[1:n_train]
test_data ← insee_data$value[(n_train + 1):n_total]
```

The ARIMA(1,1) and ARIMA(2,1) models, with the orders determined in the model identification stage, were fitted to the `train_data`:

```
fit_arma11_train ← arima(train_data, order = c(1, 1, 1))
fit_arma21_train ← arima(train_data, order = c(2, 1, 1))
```

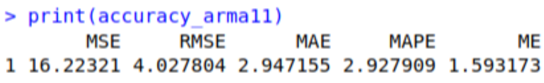
Using the fitted models, 12-step-ahead forecasts were generated to cover the period of the `test_data`:

```
forecast_arma11_test ← forecast(fit_arma11_train, h = n_test)
forecast_arma21_test ← forecast(fit_arma21_train, h = n_test)
```

For calculating Forecast Accuracy a custom R function, `calculate_accuracy()`, was used to compute several common forecast accuracy metrics by comparing the forecasted values to the actual values in the `test_data`. These metrics included Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Mean Error (ME).

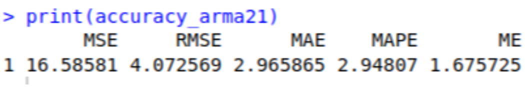
```
calculate_accuracy ← function(forecast_values, actual_values) {
  n ← length(actual_values)
  residuals ← as.numeric(forecast_values) - as.numeric(actual_values)
  mse ← mean(residuals^2)
  rmse ← sqrt(mse)
  mae ← mean(abs(residuals))
  mape ← mean(abs(residuals / actual_values)) * 100
  me ← mean(residuals)

  return(data.frame(
    MSE = mse,
    RMSE = rmse,
    MAE = mae,
    MAPE = mape,
    ME = me
  ))
}

forecasts_arma11 ← as.numeric(forecast_arma11_test$mean)
forecasts_arma21 ← as.numeric(forecast_arma21_test$mean)
test_data_numeric ← as.numeric(coredata(test_data))

accuracy_arma11 ← calculate_accuracy(forecasts_arma11, test_data_numeric)
accuracy_arma21 ← calculate_accuracy(forecasts_arma21, test_data_numeric)

print("Accuracy of ARIMA(1,1) on Testing Set:")
print(accuracy_arma11)

print("Accuracy of ARIMA(2,1) on Testing Set:")
print(accuracy_arma21)
```

The calculated accuracy metrics for both models were then printed in figure-13&14 and compared to assess their out-of-sample forecasting performance.



```
> print(accuracy_arma11)
       MSE     RMSE      MAE     MAPE       ME
1 16.22321 4.027804 2.947155 2.927909 1.593173
```

fig-13: ARMA(1,1)



```
> print(accuracy_arma21)
       MSE     RMSE      MAE    MAPE       ME
1 16.58581 4.072569 2.965865 2.94807 1.675725
```

fig-14: ARMA(2,1)

Based on these out-of-sample forecast accuracy metrics on the testing set, the **ARIMA(1,1) model appears to have performed slightly better than the ARIMA(2,1) model**, exhibiting lower RMSE, MAE, and MAPE.

suggesting marginally better accuracy on the unseen data.

# Discussion

The application of the Box-Jenkins methodology to the Turnover Volume Index in Retail Sale of Clothing in Specialized Stores provided valuable insights into the historical dynamics and potential future trends of this sector. The analysis involved rigorous testing for stationarity, careful model identification using ACF/PACF plots and information criteria, thorough model estimation, and comprehensive diagnostic checks to ensure model adequacy. Two candidate models, ARIMA(1,1) and ARIMA(2,1), demonstrated satisfactory in-sample fit.

The forecasting stage extended 12 months beyond the observed data, revealing a modest upward trend projected by both models. However, a significant widening of the prediction intervals highlighted increasing uncertainty over the forecast horizon.

The out-of-sample forecast accuracy evaluation on a hold-out sample (May 2015 to April 2016) indicated that the ARIMA(1,1) model exhibited slightly better performance, with lower RMSE, MAE, and MAPE values compared to the ARIMA(2,1) model. This, coupled with the principle of parsimony, suggests that the ARIMA(1,1) model is a more suitable choice for forecasting this time series.

The projected slight upward trend from the ARIMA(1,1) model could be cautiously interpreted as a potential for modest growth in the turnover volume. However, the wide prediction intervals underscore the considerable uncertainty surrounding these forecasts, necessitating careful consideration in future planning and decision-making within the retail clothing sector.

# Limitations and Future Work:

This study is subject to certain limitations inherent in the Box-Jenkins methodology. The ARIMA(1,1) model, being a univariate model, relies solely on the historical patterns of the Turnover Volume Index and does not explicitly account for potential external factors that could significantly influence retail sales. Economic indicators (e.g., consumer confidence, disposable income), seasonal events (beyond those addressed by differencing), marketing campaigns, and competitor actions are examples of exogenous variables that could impact the turnover volume but are not included in this analysis.

Future research could explore the application of multivariate time series models, such as ARIMAX, to incorporate relevant exogenous variables and potentially improve forecast accuracy and reduce uncertainty. Furthermore, investigating the stability of the chosen ARIMA(1,1) model by evaluating its performance over different hold-out periods or using rolling forecast origins could provide a more robust assessment of its predictive power. Exploring other time series modeling techniques, such as state-space models or machine learning approaches, might also offer alternative perspectives on forecasting this important economic indicator.

# Conclusion

This research successfully applied the Box-Jenkins methodology to analyze and forecast the Turnover Volume Index in Retail Sale of Clothing in Specialized Stores using monthly data from INSEE spanning February 1999 to April 2016. The ARIMA(1,1) model emerged as a slightly preferred model based on its out-of-sample forecast accuracy and parsimony. While the model projects a modest upward trend in turnover volume, the significant uncertainty highlighted by the widening prediction intervals suggests that future developments should be monitored closely.

The findings underscore the importance of considering the inherent uncertainty in time series forecasting and the potential benefits of incorporating external factors in future modeling efforts to enhance predictive accuracy in this dynamic sector of the economy.

# Refrences

1. Forecasting with Univariate Box - Jenkins Models Concepts and Cases. - ALAN PANKRATZ, DePauw Univenity

2. Forecasting Principles and Practice (3rd ed) Rob J Hyndman and George Athanasopoulos

3. Research on the Augmented Dickey-Fuller Test for Predicting Stock Prices and Returns.DOI: 10.54254/2754-1169/44/20232198

4. Time Series Analysis of Household Electric  Consumption with ARIMA and ARMA Models. IMECS 2013, March 13 - 15, 2013, Hong Kong

5. An Empirical Study on Stock Price Forecasting Based on ARIMA Model. ISSN 2616-7433 Vol. 4, Issue 6: 30-37, DOI: 10.25236/FSST.2022.040605