

# MATH1318 Time Series Analysis / MATH2204 Time Series and Forecasting Final Project Report

Declaration of contributions:

No	Name of Team Member	Contribution to the project
1	Sasha Nazareth (3912063)	33.33%
2	Pranav Barve (3852394)	33.33%
3	Tran Ngoc Thu Tran (3879312)	33.33%
	Sum:	1

RMIT University,  
School of Science  
2023

# Table of Contents

<b>1. EXECUTIVE SUMMARY .....</b>	<b>3</b>
<b>2. INTRODUCTION .....</b>	<b>3</b>
<b>3. DATA PRE-PROCESSING.....</b>	<b>3</b>
<b>4. DATA EXPLORATION AND VISUALIZATION .....</b>	<b>4</b>
4.1. Descriptive Statistics.....	4
4.2. Time series plot .....	5
4.3. Scatter plot.....	7
<b>5. CHECK ASSUMPTIONS.....</b>	<b>8</b>
5.1. Normality Check.....	8
5.1.1 Q-Q Plot.....	8
5.1.2 Shapiro-Wilk test .....	8
5.2. Stationary check.....	9
5.2.1 ACF and PACF .....	9
5.2.2 The Dickey-Fuller Unit-Root Test (ADF test).....	10
<b>6. METHODOLOGY.....</b>	<b>11</b>
6.1 Variance Stabilising Transformation (Box-Cox Transformation).....	11
6.2 Normality check after Box-Cox Transformation.....	13
6.3. Changing Variance (ARCH) check after Box-Cox Transformation.....	14
<b>7. MODEL SELECTION.....</b>	<b>14</b>
7.1. ARIMA Model Selection .....	15
7.1.1 ACF and PACF Plot.....	15
7.1.2 EACF Table .....	16
7.1.3 BIC Table.....	16
7.1.4 Possible ARIMA models .....	17
7.1.5 Significance Tests of Coefficients .....	18
7.1.6 Parameter Estimation .....	18
7.1.7 Model Fitting – ARIMA (1,0,1) model.....	19
7.1.8 Model Fitting – ARIMA (1,0,4) model.....	20
7.2 GARCH model selection .....	22

7.2.1 Absolute Return Series.....	23
7.2.2 Squared Return Series.....	25
7.2.3 Model Fitting and Significance Tests of Coefficients - GARCH .....	27
<b>7.3 Model Fitting: Combining ARIMA and GARCH .....</b>	<b>30</b>
7.3.1 ARIMA(1,1)+GARCH(0,1).....	30
7.3.2 ARIMA(1,1)+GARCH(1,1).....	31
<b>8. FURTHER DIAGNOSTICS CHECKING ARIMA AND GARCH .....</b>	<b>33</b>
8.1. AIC and BIC Scores .....	33
8.2 ME, RMSE, and MAE Scores.....	33
<b>9. FORECASTING .....</b>	<b>34</b>
<b>10. RESULTS .....</b>	<b>36</b>
<b>11. DISCUSSION.....</b>	<b>36</b>
<b>12. CONCLUSION .....</b>	<b>36</b>
<b>13. RECOMMENDATIONS .....</b>	<b>37</b>
<b>14. APPENDICES: R CODE .....</b>	<b>38</b>
14.1 Clean the data.....	38
14.2 Analyse data .....	41
<b>15. REFERENCES .....</b>	<b>53</b>

## 1. Executive Summary

The objective of this report was to forecast potential patterns for the monthly total rainfall in Melbourne for the next ten months, starting from July 2017. To manage the changing variance and volatility clustering observed in the rainfall data, we utilized Box-Cox transformation and a GARCH model, respectively. Our comprehensive analysis led us to select the ARMA(1,1) + GARCH(0,1) model based on criteria like the Akaike Information Criterion (AIC), the Bayesian Information Criterion (BIC), and diagnostic checking. This model predicted an overall increase in rainfall over the forecast period. We recommend regular updates and reassessment of the model to ensure accuracy as new data becomes available.

## 2. Introduction

Climate forecasting is integral to long-term planning for various sectors, including agriculture, water management, and infrastructure development. Accurate and precise forecasts can provide vital information to guide decisions and strategic plans. In Australia, weather patterns are becoming more unpredictable due to climate change, making reliable forecasting models even more critical. This study aims to answer the following research question: "What potential patterns and variations can we anticipate for the monthly total rainfall in Melbourne for the next ten months, starting from July 2017?" This research focuses on predicting the 'Rainfall' variable, which quantifies the amount of rainfall recorded in a day in millimetres (mm).

## 3. Data Pre-Processing

(Script in [Appendix 14.1](#))

The dataset used in this study was sourced from [Kaggle \[7\]](#), a collection of daily weather observations from various weather stations across Australia, primarily compiled by the Australian Bureau of Meteorology. This dataset's key variable of interest is 'Rainfall', which records the daily rainfall in millimetres.

The pre-processing steps are initiated by loading the dataset into the R environment and importing the necessary libraries. From the entire dataset, the 'Location', 'Rainfall', and 'Date' columns were selected. The 'Date' column was transformed into the R date format for easier manipulation and analysis.

A common challenge with raw data is the handling of missing values. In this dataset, missing values in the 'Rainfall' column were filled with the median rainfall value of their respective location. After this, the data was filtered only to include observations from locations containing the term "Melbourne". The 'Location' column was dropped, and the remaining columns were rearranged to place 'Date' as the first column.

The cleaned data were grouped by year and month, and the sum of rainfall for each month was calculated to provide an overview of the monthly rainfall patterns. Months with zero rainfall were also identified and noted.

To make the data more conducive for time series analysis, a new column, 'YearMonth' was created by extracting the year and month from the 'Date' column. The data was then aggregated monthly, providing a simplified view of the monthly rainfall amounts.

The final pre-processing step addressed the missing data in the 'TotalRainfall' column. Here, linear interpolation was used to fill the gaps, providing a reasonable estimate for missing values. Once all pre-processing steps were completed, the cleaned dataset was saved and ready for subsequent analysis.

The pre-processing of the rainfall dataset from Australia involved a series of steps, including missing value imputation, data aggregation monthly, and linear interpolation for remaining missing values. This robust pre-processing approach has prepared the dataset for subsequent forecasting analyses. The resulting data will provide a basis for building a predictive model to answer the research question regarding future monthly rainfall amounts in Melbourne. These insights will be crucial for various stakeholders in planning and preparing for potential weather patterns.

## 4. Data Exploration and Visualization

### 4.1. Descriptive Statistics

The statistics provide a useful overview of the central tendency and dispersion of the monthly rainfall data.

*Table 1. Summary statistics of monthly total rainfall dataframe*

Statistic	YearMonth	TotalRainfall
Length	108	
Class	character	
Minimum		2.40
1st Quartile (25%)		43.85
Median (50%)		70.10
Mean		84.23
3rd Quartile (75%)		112.40
Maximum		276.00

The dataset comprises 108 monthly total rainfall observations from Melbourne, Australia. The “*TotalRainfall*” variable, which represents the amount of rainfall in millimetres, is summarized as follows:

- **Minimum (2.4mm):** The smallest amount of rainfall recorded in a month.
- **1st quartile (43.85mm):** This is also known as the 25th percentile. It means that in approximately 25% of the observed months, the rainfall was 43.85mm or less.
- **Median (70.1mm):** The 50th percentile or the second quartile is the middle value when the data is sorted in ascending order. It indicates that in 50% of the observed months, the rainfall was 70.1mm or less. The median provides a central value for the data set.
- **Mean (84.23mm):** The monthly average rainfall overall observed months. It is calculated by summing all observed values and dividing by the count (108). Every data point influences the mean in the dataset so that it can be significantly affected by extreme values or outliers.

- **3rd quartile (112.4mm):** The 75th percentile implies that in 75% of the observed months, the rainfall was 112.4mm or less.
- **Maximum (276mm):** This is the highest amount of rainfall recorded in a single month within the observation period.

The difference between the 3rd and 1st quartiles ( $112.4\text{mm} - 43.85\text{mm} = 68.55\text{mm}$ ), known as the interquartile range, gives a sense of the statistical dispersion or variability of the rainfall data.

The “*YearMonth*” variable, a character string representing the year and month of each observation, has 108 unique entries aligning with the number of total observations. This suggests that there is one rainfall observation per month for the duration covered by the dataset. However, this does not provide insights into trends or patterns over time, which would require further time series analysis or visualization.

## 4.2. Time series plot

[1] "ts"	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
2008							44.4	44.8	12.0	14.2	54.2	76.8
2009	2.4	9.0	83.4	72.6	21.8	45.2	64.8	76.8	126.2	42.2	220.6	97.6
2010	52.4	99.4	172.8	39.4	56.4	110.4	58.2	129.4	95.0	276.0	245.2	169.4
2011	186.6	213.4	57.2	84.2	111.2	52.8	58.6	34.8	148.0	136.0	220.8	140.8
2012	40.0	141.4	123.0	86.0	136.6	126.0	93.2	103.2	69.0	54.8	59.6	38.5
2013	17.4	51.5	85.6	52.2	64.0	225.8	97.4	96.8	153.2	94.0	116.0	76.8
2014	22.0	39.6	51.6	142.4	53.0	97.8	39.6	68.4	81.4	88.6	109.8	81.2
2015	35.6	38.8	22.2	35.2	25.4	23.4	63.6	19.4	27.8	16.6	54.2	44.8
2016	44.8	4.4	27.2	36.4	104.0	120.8	127.6	91.8	187.0	140.8	71.2	117.0
2017	57.2	72.6	59.8	221.6	45.0	25.8						

Figure 1. Time series object: Monthly Total Rainfall with frequency of 12

**Time Series Plot of Monthly Total Rainfall (2008-2017)**

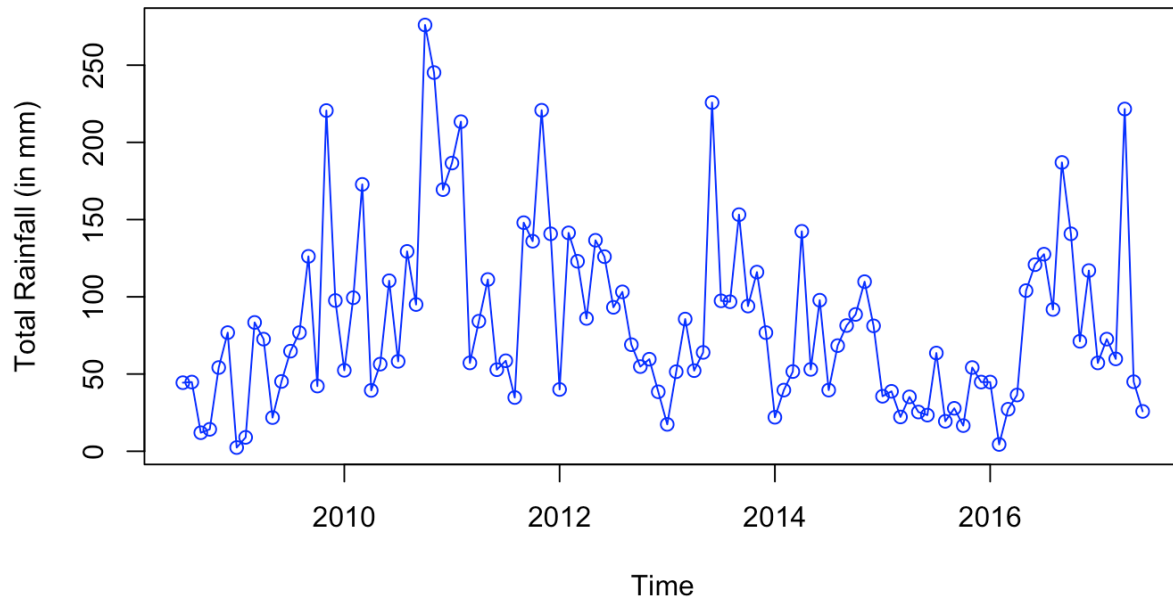


Figure 2. Time series plot of total monthly rainfall from 2008 to 2017 without month labels

**Time series plot  
of Monthly Total Rainfall (2008-2017) with month labels.**

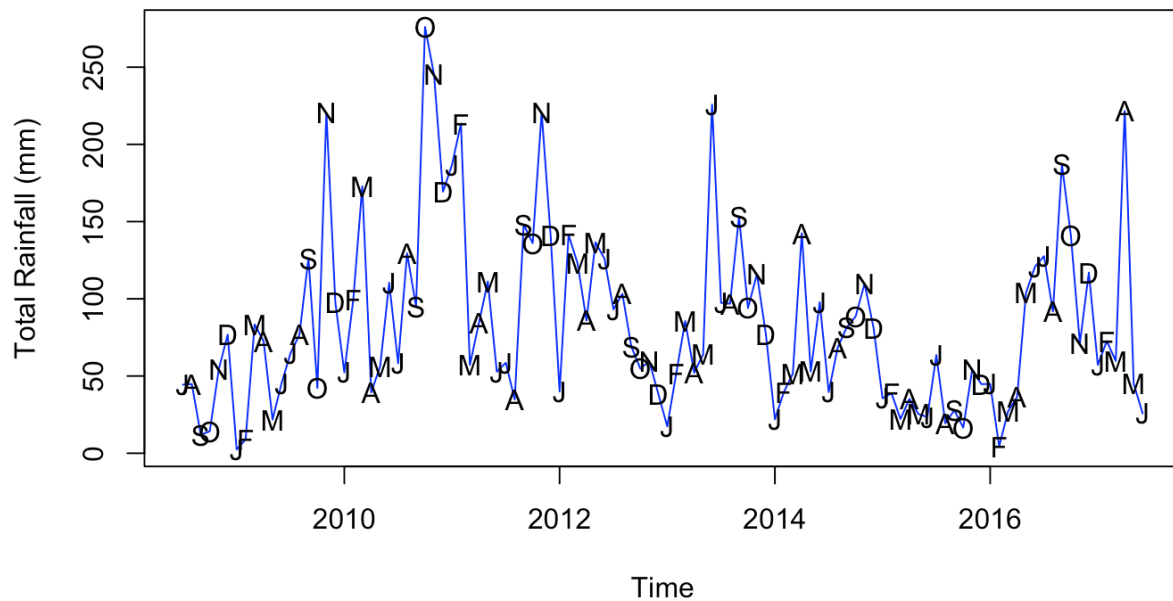


Figure 3. Time series plot of total monthly rainfall from 2008 to 2017 with the month labels to observe the seasonality pattern

We will be analysing the plot using the five main points observing figure 2 and 3:

- + **Trend** - Over the years, there has not been any discernible long-term growth or reduction in the monthly rainfall totals.
- + **Seasonality** - Even with the month names, seasonal fluctuations are imperceptible.
- + **Changing variance** - Over time, there is a fluctuation in the range of rainfall quantities, with some years having a greater range than others.
- + **Behaviour** - The amount of rainfall displays unpredictable behaviour, with inexplicable jumps and drops across successive months.
- + **Change point** - Potential change points are indicated by abrupt changes in rainfall, such those that occurred between April and May 2017.

### 4.3. Scatter plot

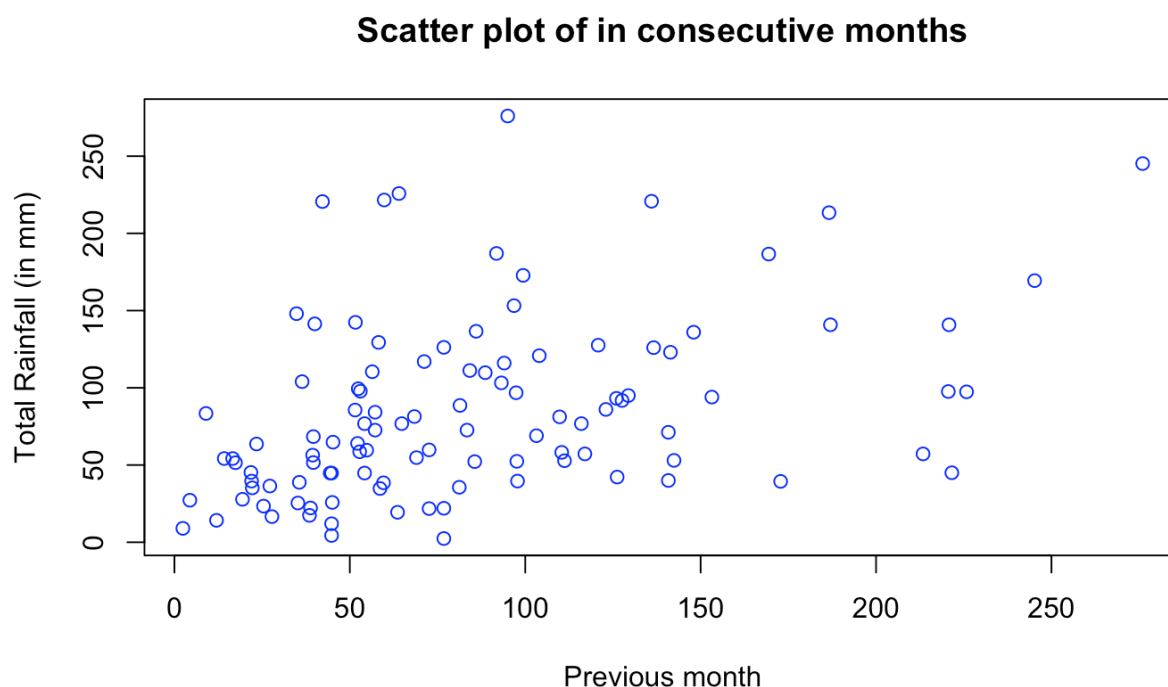


Figure 4. The scatter plot shows total monthly rainfall and preceding months

There is a little association between the overall amount of rainfall and the preceding month, as seen in the scatter figure above (Figure 4). The scatter plot's dots are more numerous towards the bottom left corner of the graph. Therefore, by examining the tenuous connection, we wished to further investigate this connection.

The scatter plot indicates a moderate positive relationship between the overall rainfall amount of a given month and the preceding month, with more data points concentrated towards the lower left of the graph. The calculated correlation coefficient,  $r = 0.4068457$ , confirms this moderate positive linear relationship. Although this suggests that higher rainfall months tend to follow similarly high rainfall months, the relationship could be stronger and merits further investigation for a deeper understanding.



$$r = \text{cov}(X, Y) / (\text{sd}(X) * \text{sd}(Y)) = 0.4068457$$

## 5. Check assumptions

### 5.1. Normality Check

#### 5.1.1 Q-Q Plot

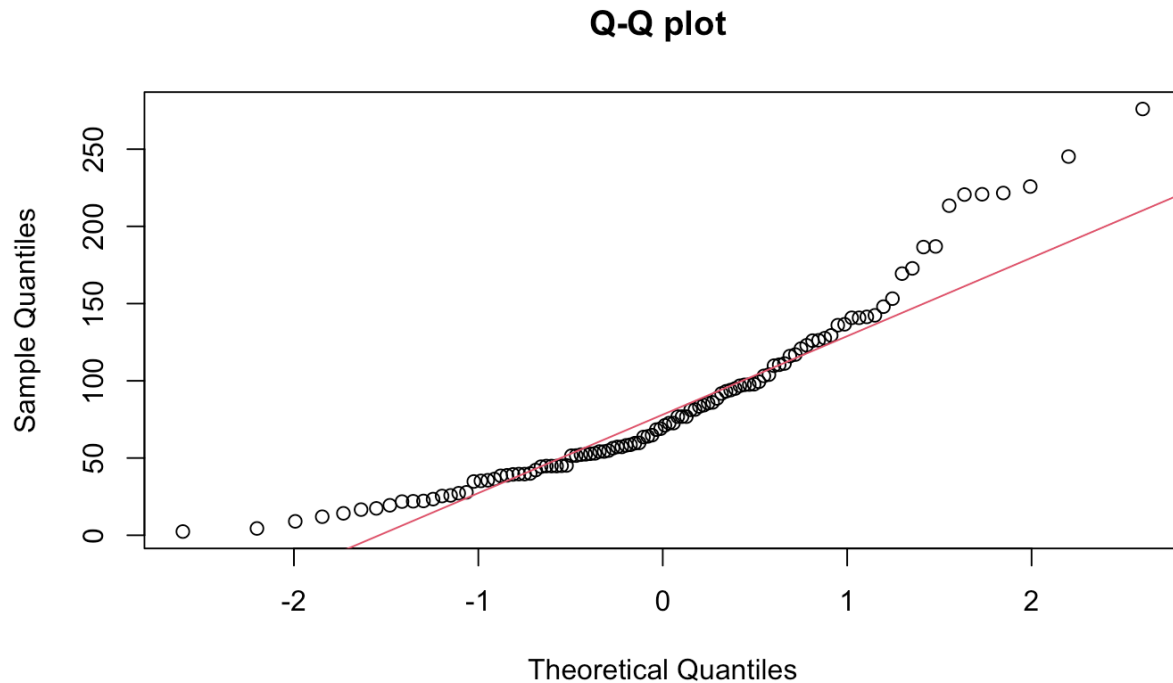


Figure 5. The quantile-quantile (Q-Q) plot shows normality of the time series object.

The points on the Q-Q plot are expected to lie approximately along a straight red line. However, the plot curve points are upward at two tails, suggesting that the dataset has heavier tails than the theoretical distribution. Deviations from a straight line indicate departures from the expected distribution.

#### 5.1.2 Shapiro-Wilk test

We expect the data to be normally distributed with the p-value greater than our alpha level  $\alpha$  of 0.05.

- Null Hypothesis ( $H_0$ ): The data follows a normal distribution.
- Alternative Hypothesis ( $H_a$ ): The data does not follow a normal distribution.

```

Shapiro-Wilk normality test

data:  weatherAUS_ts
W = 0.90995, p-value = 1.987e-06

```

Figure 6. Shapiro-Wilk test to test the normality of the time series object numerically.

We use the Shapiro-Wilk test to validate the normality of a dataset further. The null hypothesis of this test is that the population is normally distributed. In this case, the test statistic,  $W$ , is 0.90995, and the p-value is  $1.987e-06$ , which is very close to 0. We would reject the null hypothesis since the p-value is less than the typical significance level (0.05). Based on this test, we have strong evidence to conclude that the weatherAUS\_ts data is not normally distributed.

After visual and statistical inspection, we have sufficient evidence that the series is not normally distributed and must be further normalized.

## 5.2. Stationary check

### 5.2.1 ACF and PACF

$$\text{ACF}(k) = \text{Corr}(Y_t, Y_{t+k})$$

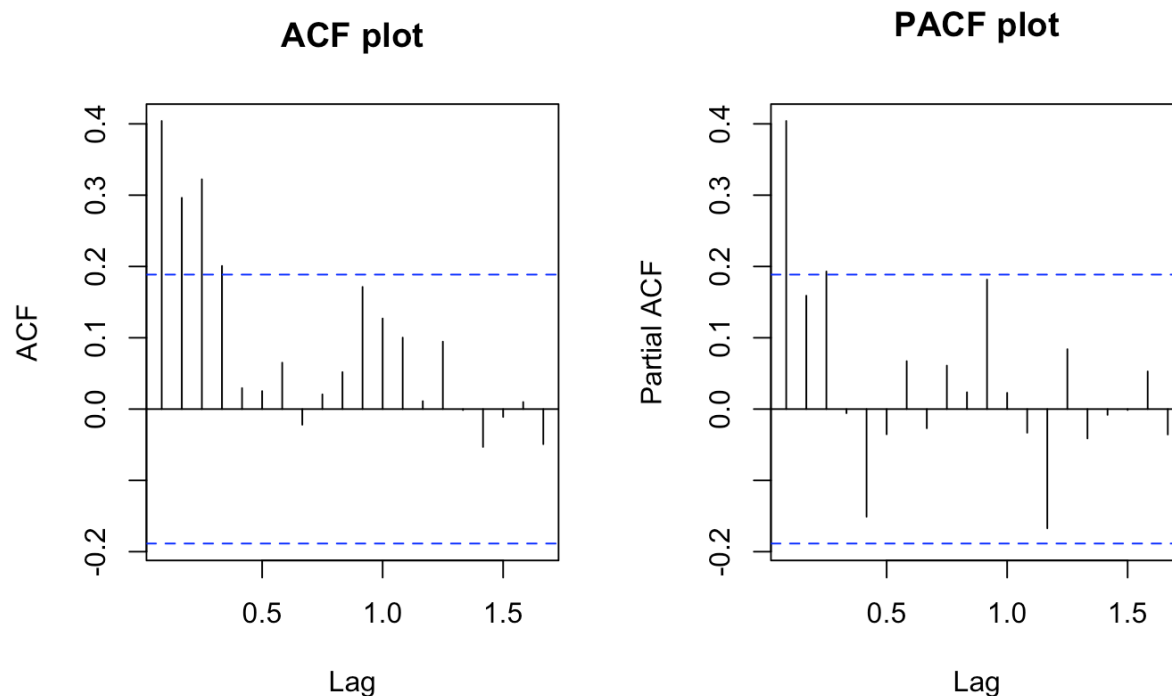


Figure 7. ACF and PACF of the time series object

There is a direct correlation between the value of the coefficient ( $\theta$ ) and the bar results in the autocorrelation function (ACF) plot and the partial autocorrelation function (PACF) plot, and these two significant plot patterns are an indicator of stationary series. The ACF plot does not have a slowly decaying pattern, and the PACF plot does not have an incredibly significant first lag. These two important plot patterns are an indicator of stationary series. For our series, being stationary is a particularly good indicator since it is crucial for the data to remain stationary if we want to forecast ARIMA models.

### 5.2.2 The Dickey-Fuller Unit-Root Test (ADF test)

We expect all three stationarity checks to converge on the same result, confirming that the time series data is stationary.

*Table 2. The table summarizes the results from the three stationary tests for the time series object.*

Stationary Test	Null Hypothesis	Alternative Hypothesis	Result
Augmented Dickey-Fuller	Unit Root Exists	Stationary	Dickey-Fuller = -3.8675, p-value = 0.01825, Stationary
Phillips-Perron Unit Root	Unit Root Exists	Stationary	Dickey-Fuller $Z(\alpha) = -70.445$ , p-value = 0.01, Stationary
KPSS for Level Stationarity	Stationary	Unit Root Exists	KPSS Level = 0.21735, p-value = 0.1, Stationary

The table summarizes the results from the three stationary tests performed: The Augmented Dickey-Fuller Test, the Phillips-Perron Unit Root Test, and the KPSS Test for Level Stationarity. The Null and Alternative Hypotheses are provided for each test, along with the resulting statistics.

The Augmented Dickey-Fuller test and the Phillips-Perron Unit Root test, which operate under the null hypothesis of non-stationarity, returned results allowing us to reject the null hypothesis, thus affirming stationarity. The Dickey-Fuller statistics and the respective p-values (-3.8675 and 0.01825 for the Augmented Dickey-Fuller, -70.445 and 0.01 for the Phillips-Perron) strengthen our confidence in the stationarity of the series.

On the other hand, the KPSS Test for Level Stationarity, which assumes stationarity under the null hypothesis, returned a p-value of 0.1. Since this p-value is higher than the conventional 0.05 cut-off, we fail to reject the null hypothesis, corroborating the data's stationarity.

The agreement among these three tests substantiates the stationarity of the time series data, satisfying a crucial precondition for most time series models. Therefore, we are now equipped to move forward with our subsequent modeling procedures with the assurance of working with a stationary time series.

## 6. Methodology

To make the series normal, we decided to apply Box-Cox Transformation on the data.

### 6.1 Variance Stabilising Transformation (Box-Cox Transformation)

The Box-Cox transformation is a family of power transformations indexed by a parameter lambda and is used to stabilize variance and make the data more closely follow a normal distribution. This transformation often leads to better model fit and more reliable statistical inference.

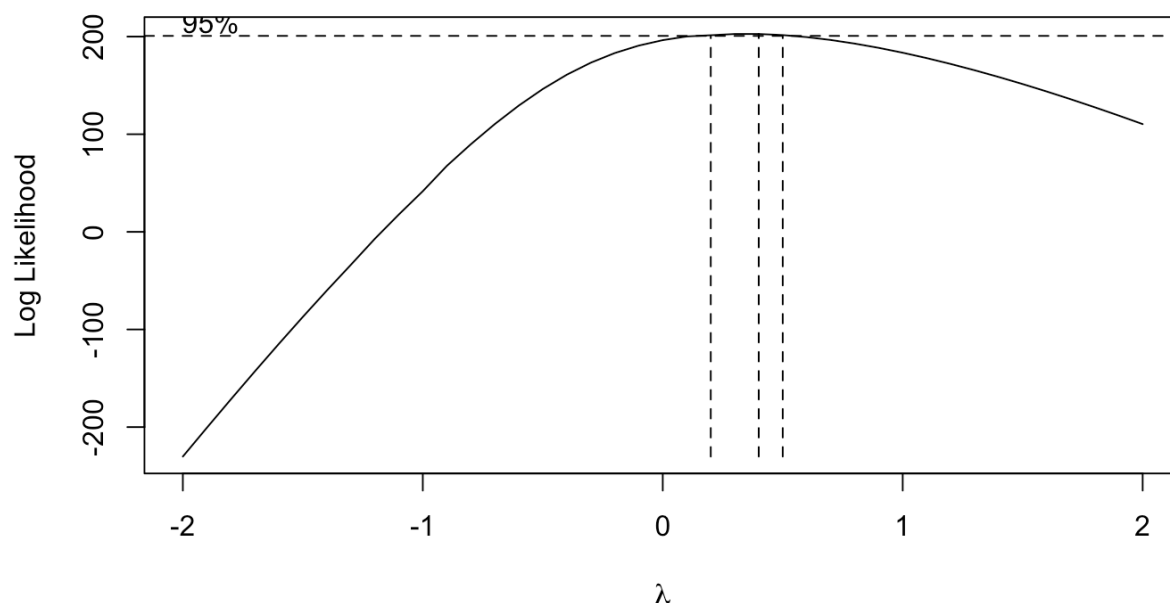


Figure 8. The plot shows an optimal lambda of 0.4. The first and third vertical lines are 0.2 and 0.5, respectively.

Based on the BoxCox.ar output, the optimal lambda value lies between 0.2 and 0.5, with the maximum likelihood estimation yielding a lambda value of 0.4. This indicates that the data can be best transformed to approximate a normal distribution using a power of 0.4.

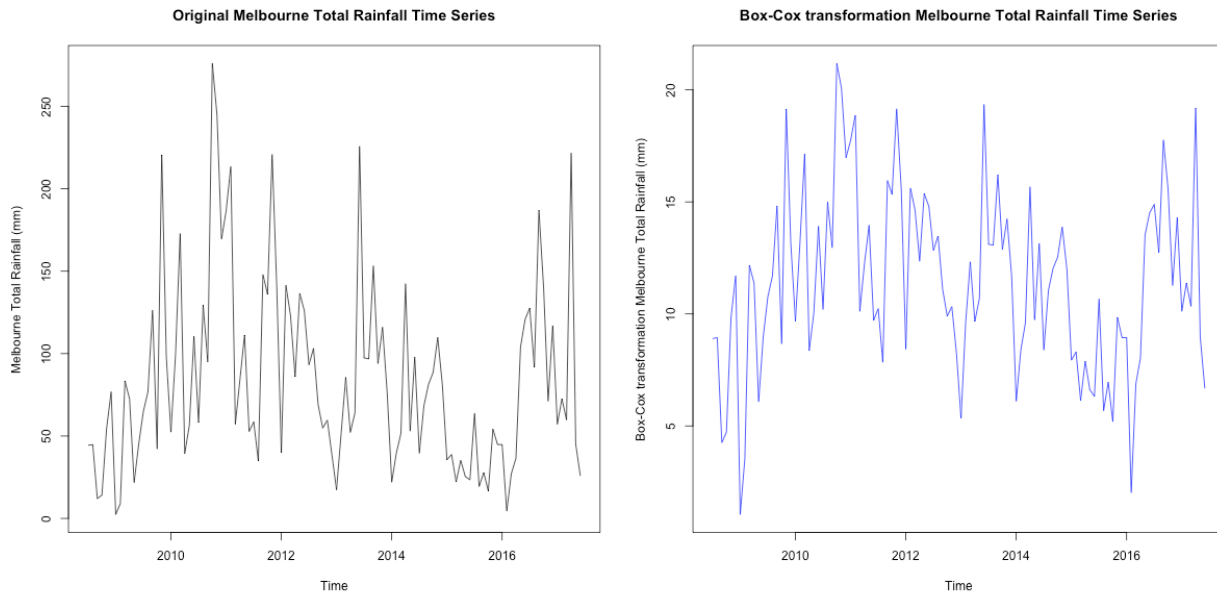


Figure 9. The visual comparison of the original and Box-Cox transformed time series of total monthly rainfall in Melbourne.

After applying the Box-Cox transformation with the optimal lambda value, the transformed time series data (BC.weatherAU) is compared with the original time series plot:

- **Variance:** In the original data, the variance changes over time, indicating non-stationarity. However, the variance is more consistent over time after the Box-Cox transformation. This suggests that the transformation has stabilized the variance, rendering the data more amenable for modeling techniques that assume a constant variance.
- **Scale:** The change in the y-axis scale from 0-250 in the original plot to 0-20 in the Box-Cox transformed plot demonstrates the effect of the transformation. The transformation has significantly reduced the scale of the data, which could be beneficial in managing outliers and extreme values.
- **Normality:** The distribution of values in the original time series appears skewed and non-normal. However, after the Box-Cox transformation, the distribution appears more symmetrical and hence, more akin to a normal distribution.

## 6.2 Normality check after Box-Cox Transformation

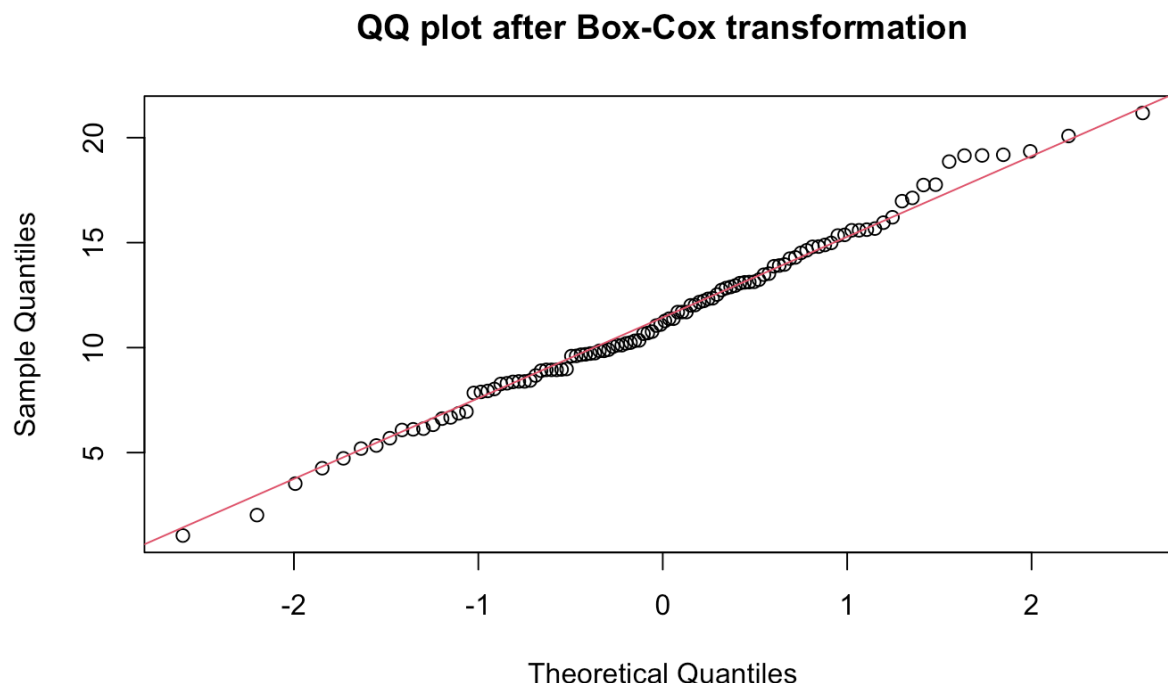


Figure 10. The quantile-quantile (Q-Q) plot shows normality of the time series object after transformation.

The QQ plot following the Box-Cox transformation displays the quantiles of the transformed data against the quantiles of a standard normal distribution. A good alignment of the data points with the theoretical red line in the Q-Q plot suggests that the transformed data closely follows a normal distribution after the transformation.

```
Shapiro-Wilk normality test

data:  BC.weatherAU
W = 0.99327, p-value = 0.8783
```

Figure 11. The Shapiro-Wilk test to check the normality of the transformed time series object.

A non-significant p-value of 0.8783 is greater than the alpha level 0.05; we fail to reject the null hypothesis. Thus, there is insufficient evidence to conclude that the data deviates from a normal distribution. Based on these results, the Box-Cox transformed BC.weatherAU data appears normally distributed.

This finding supports our earlier observation from the QQ plot and indicates that the Box-Cox transformation successfully made the data more normally distributed.

### 6.3. Changing Variance (ARCH) check after Box-Cox Transformation

We applied the McLeod-Li test to the Box-Cox transformed total rainfall time series data in our analysis. The McLeod-Li test is crucial for detecting the presence of heteroscedasticity or volatility clustering in the data.

- $H_0$ : There is no volatility clustering in the series. There are no ARCH effects present in the series.
- $H_a$ : There is volatility clustering in the series. There are ARCH effects that exist in the series.

**P-values of McLeod-Li test statistics  
for Transformed Total Rainfall Time Series**

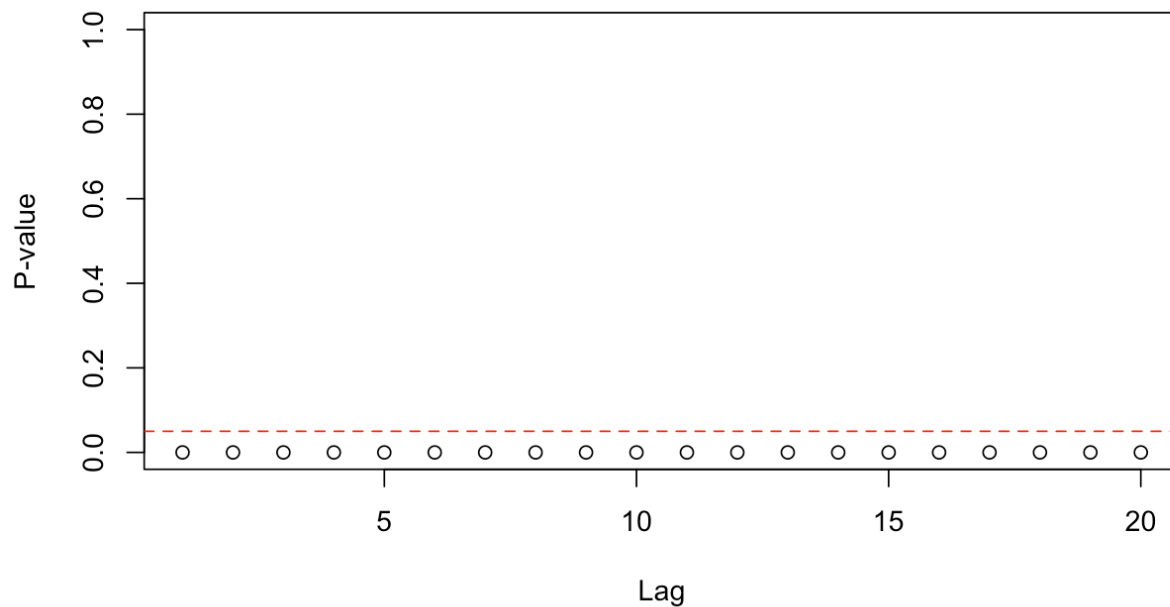


Figure 12. McLeod-Li test to check the changing variance in the transformed time series object.

Our results from the McLeod-Li test indicate significant results at the 5% significance level across all lags. Specifically, all the p-values obtained from the test are less than the 0.05 significance level (all the dots lie below the reference red line), affirming our suspicion of changing variance within the data set. This is a crucial observation as it suggests the presence of heteroscedasticity (ARCH), which needs to be addressed in our subsequent modelling to ensure robust and reliable forecasts. This outcome provides compelling evidence of volatility clustering in the time series data, as a statistically significant result implies that the variance in the time series is not constant but changes over time. As such, our analysis would benefit from using models designed to handle this kind of volatility by modelling the conditional variance structure of time series data, such as Generalized Autoregressive Conditional Heteroscedasticity (GARCH) models.

## 7. Model selection

The model selection process begins with the transformed series we obtained using the Box-Cox transformation. This transformation was utilized to stabilize the variance and ensure that the

series more closely aligns with a normal distribution, a key assumption of **ARIMA and GARCH models**.

### 7.1. ARIMA Model Selection

We first aim to identify an optimal ARIMA model. To do this, we consider various possible models suggested by the ACF, PACF, EACF, and BIC methodologies. **Then we use diagnostic checking to confirm the chosen models. Each potential ARIMA model is fitted to the transformed series, and we evaluate the models based on their goodness-of-fit and complexity, utilizing criteria such as the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC).**

#### 7.1.1 ACF and PACF Plot

The Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots are commonly used tools in time series analysis to determine the order of Autoregressive Integrated Moving Average (ARIMA) models. The ARIMA model is denoted  $ARIMA(p,d,q)$  where:

- 'p' is the order of the Autoregressive part.
- 'd' is the order of differencing required to make the time series stationary (in our case, it is 0 because our data is stationary)
- 'q' is the order of the Moving average part.

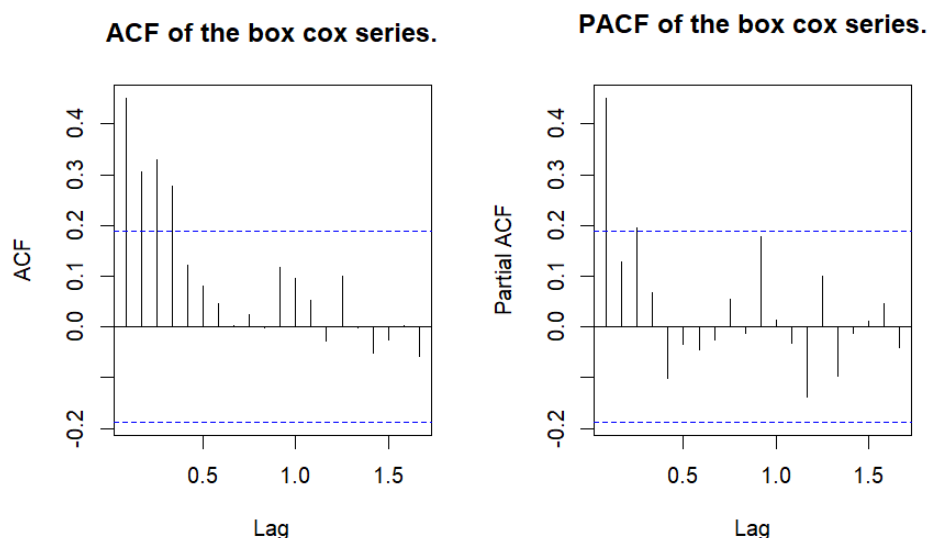


Figure 13. ACF and PACF plots for ARIMA model selection.

Our analysis closely examined the ACF and PACF plots and identified two potential models:  $ARIMA(3,0,4)$  and  $ARIMA(1,0,4)$ .

The order of this model was suggested based on the following observations from the PACF and ACF plots:

- The ACF plot shows a slow decay, particularly from lag 4 onward, suggesting the possible use of an  $MA(4)$  model.
- The PACF plot has two significant spikes at lags 1 and 3, suggesting the possible use of an  $AR(3)$  model.



### 7.1.2 EACF Table

The Extended Autocorrelation Function (EACF) is a diagnostic tool used to identify potential ARIMA models by looking at correlations for various combinations of autoregressive (AR) and moving average (MA) orders. In the EACF table, each cell provides an autocorrelation value for a corresponding AR and MA order. The presence of an 'o' in a cell indicates a strong autocorrelation, and thus the intersection of the row (AR order) and column (MA order) at this 'o' suggests a potential ARIMA model.

AR/MA	0	1	2	3	4	5	6	7	8	9	10
0	x	x	x	x	o	o	o	o	o	o	o
1	x	o	o	o	o	o	o	o	o	o	o
2	x	x	o	o	o	o	o	o	o	o	o
3	x	x	o	o	o	o	o	o	o	o	o
4	x	x	x	o	o	o	o	o	o	o	o
5	x	o	o	x	o	o	o	o	o	o	o
6	x	x	o	o	o	o	o	o	o	o	o
7	x	o	o	o	o	o	o	o	o	o	o
8	o	x	o	o	o	o	o	o	o	o	o
9	x	o	o	o	o	x	o	o	o	o	o
10	o	o	o	o	o	o	o	o	x	o	o

Figure 14. EACF table for ARIMA model selection.

Looking at the EACF table, specifically, the red line highlighted in the consecutive good models, the top left vertex (the smallest possible model) occurs at AR=1 and MA=1, which suggests an ARIMA(1,0,1) model. This model is characterized by one previous value (lag) and the last error term being significant for predicting the current value of the time series.

However, the EACF table also suggests additional models beyond the vertex model. The adjacent vertex (AR=1 and MA=2) points towards an ARIMA(1,0,2) model. This model implies that one previous value (lag) and the last two error terms are significant for predicting the current value of the time series. Additionally, the table shows a strong autocorrelation at AR=2 and MA=2, suggesting an ARIMA(2,0,2) model. This model suggests that the last two values (lags) and the last two error terms are significant for predicting the current value of the time series. From this EACF plot, we get a set of another possible ARIMA (p, d, q) models: ARIMA (1,0,1), ARIMA (1,0,2), ARIMA (2,0,2).

### 7.1.3 BIC Table

The Bayesian Information Criterion (BIC) can be visualized in a table format. In this table, different shades of grey represent the BIC value's magnitude for each ARIMA model, with

darker grey indicating lower BIC values, thus, better models. A low BIC value for an ARIMA(p, d, q) model indicates a potentially good balance of model fit and simplicity. This is because the BIC considers both the likelihood of the data given the model (rewarding good fit) and the number of parameters in the model (penalizing complexity). We aim to find the smallest possible model that sufficiently explains the data. The consecutive grey areas in the table help identify the appropriate orders of autoregressive (p) and moving average (q) terms in our ARIMA model.

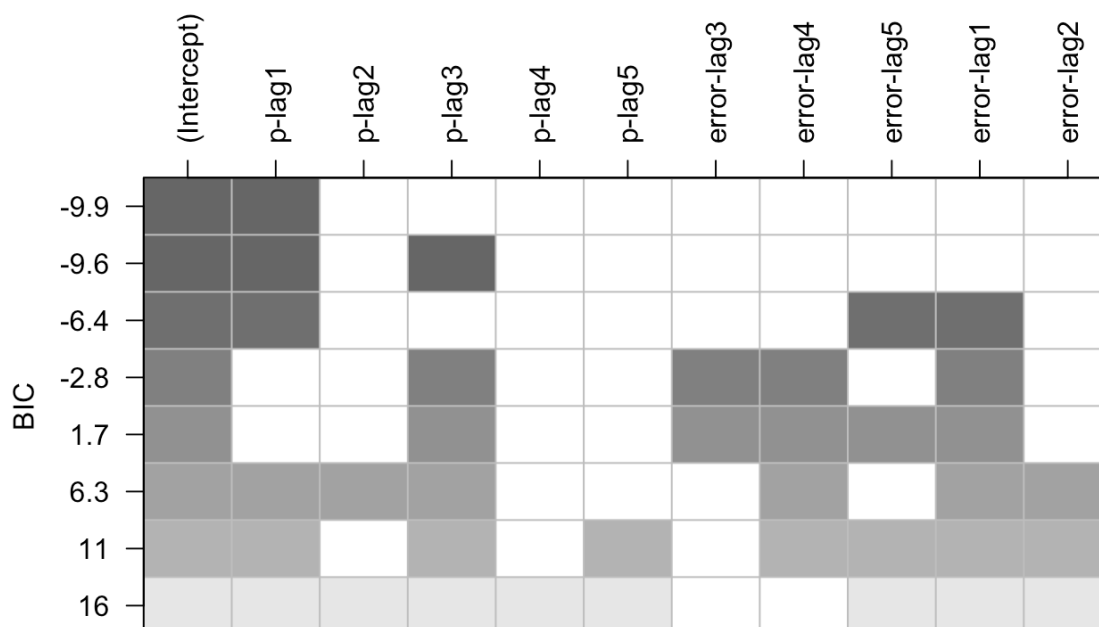


Figure 15. BIC table for ARIMA model selection

From the BIC plot, we get following ARIMA models:

ARIMA (p, d, q): ARIMA (1,0,1), ARIMA (1,0,4), ARIMA (1,0,5), ARIMA (3,0,1), ARIMA (3,0,4), ARIMA (3,0,5)

#### 7.1.4 Possible ARIMA models

- The Autocorrelation Function (ACF) and the Partial Autocorrelation Function (PACF) plots have suggested two possible models: ARIMA (3,0,4) and ARIMA (1,0,4).
- A further model identification step, the Extended Autocorrelation Function (EACF) plot, has proposed an additional set of potential models: ARIMA (1,0,1), ARIMA (1,0,2), and ARIMA (2,0,2).
- Lastly, we leveraged the Bayesian Information Criterion (BIC) plot for model selection. The BIC approach has recommended the following models: ARIMA (1,0,1), ARIMA (1,0,4), ARIMA (1,0,5), ARIMA (3,0,1), ARIMA (3,0,4), and ARIMA (3,0,5).

From the above potential ARIMA models suggested by different diagnostic tests, the models validated by at least two tests are:

- ARIMA (1,0,1): This model is suggested by both the EACF and BIC plots.

- ARIMA (1,0,4): This model is recommended by the ACF and PACF plots and the BIC plot.

Multiple methods have consistently pointed out these two models, ARIMA (1,0,1) and ARIMA (1,0,4), strengthening their validity and making them strong contenders for further analysis and, potentially, model fitting for the time series data.

While these two models are all potential candidates, it is important to consider other factors, such as the principle of parsimony (choosing the simplest model that adequately explains the data) and further model diagnostics (like examining coefficient significant, residuals, and comparing information criteria such as AIC or BIC) to choose the best fitting model for the data.

### 7.1.5 Significance Tests of Coefficients

*Table 3. The coefficient significance interpretation*

Significance Level	Symbol	p-value range
Highly significant	***	p-value < 0.001
Very significant	**	0.001 ≤ p-value < 0.01
Significant	*	0.01 ≤ p-value < 0.05
Suggestive	.	0.05 ≤ p-value < 0.1
Not significant	(blank)	p-value ≥ 0.1

The z-test of coefficients is used to determine the significance of the estimated parameters in each model. We expect all coefficients in each model to be significant. Smaller p-values (below 0.05) indicate a significant relationship between the predictor and the response variable. The significance level is marked with asterisks or a dot. A blank in the symbol column indicates that the p-value is not significant.

### 7.1.6 Parameter Estimation

Parameter estimation forms the crux of ARMA model analysis, as the quality of the parameters directly influences the effectiveness and accuracy of the model's predictions. In this analysis, we focused on estimating the parameters of our ARIMA models using three distinct methodologies, namely 'CSS': Conditional Sum of Squares, 'ML': Maximum Likelihood, and 'CSS-ML': A Hybrid Method. Each of these methods has its unique strengths and applications.

The 'CSS' method minimizes the sum of squared residuals to find the best fit for the model. The 'ML' method, on the other hand, maximizes the likelihood function to determine the model's parameters. The 'CSS-ML' method is a hybrid approach that combines the strengths of both the 'CSS' and 'ML' methodologies.

The analysis of the ARIMA(1,0,1) model has been carried out using three different estimation methods: Maximum Likelihood (ML), Conditional Sum of Squares (CSS), and a combination of both CSS and ML. All three methods have resulted in significant coefficient estimates for the autoregressive (AR1), moving average (MA1), and intercept terms, suggesting that all these parameters play a crucial role in the model.

```

```{r Parameter Estimation}
model_101_ML <- arima(BC.weatherAU, order = c(1,0,1), method='ML')
coefest(model_101_ML)

model_101_CSS <- arima(BC.weatherAU, order = c(1,0,1), method='CSS')
coefest(model_101_CSS)

model_101_CSS_ML <- arima(BC.weatherAU, order = c(1,0,1), method='CSS-ML')
coefest(model_101_CSS_ML)
```

z test of coefficients:

      Estimate Std. Error z value Pr(>|z|)
ar1      0.79342   0.10878   7.2936 3.017e-13 ***
ma1     -0.45532   0.15877  -2.8678 0.004133 **
intercept 11.24886   0.87051  12.9222 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

z test of coefficients:

      Estimate Std. Error z value Pr(>|z|)
ar1      0.80602   0.10053   8.0179 1.076e-15 ***
ma1     -0.48158   0.15360  -3.1353 0.001717 **
intercept 11.49589   0.91333  12.5868 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

z test of coefficients:

      Estimate Std. Error z value Pr(>|z|)
ar1      0.79370   0.10872   7.3003 2.872e-13 ***
ma1     -0.45560   0.15871  -2.8706 0.004097 **
intercept 11.24756   0.87127  12.9094 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Figure 16. Parameter Estimation methods and models' outputs

Even though all three estimation methods provide consistent results in terms of significance, we are selecting the Maximum Likelihood estimation method for the final model. The reason for this choice is its robustness. Maximum Likelihood estimation has solid theoretical properties, and under normal conditions, it provides estimates that are the most efficient among all other methods. It provides unbiased, consistent estimates with the smallest variance compared to all other consistent estimators when the sample size exceeds infinity. Given the normality of our transformed data and the increased precision of the 'ML' method, we decided to proceed with the Maximum Likelihood estimation for our models. This decision is since 'ML' estimators have desirable statistical properties like consistency and efficiency when the model's assumptions are satisfied, which is the case with our data.

### 7.1.7 Model Fitting – ARIMA (1,0,1) model

The ARIMA (1,0,1) model's parameter estimation and diagnostic tests results:

- **Parameter estimation results** - All coefficients (ar1, ma1, intercept) are significant as the p-values are less than the 0.05 significance level. Specifically, the autoregressive term (ar1) coefficient is 0.79342, the moving average term (ma1) is -0.45532, and the intercept term is 11.24886.

Table 4. ARIMA (1,0,1) model's z-test of coefficients

|           | Estimate | Std. Error | z value | Pr(> z )  | Significance |
|-----------|----------|------------|---------|-----------|--------------|
| ar1       | 0.79342  | 0.10878    | 7.2936  | 3.017e-13 | ***          |
| ma1       | -0.45532 | 0.15877    | -2.8678 | 0.004133  | **           |
| intercept | 11.24886 | 0.87051    | 12.9222 | < 2.2e-16 | ***          |

- **Residuals analysis** - The mean of residuals is approximately 0.01157422, close to zero, indicating that the model has successfully captured most of the systematic information in the data. The standard deviation of residuals is 1.004595, which indicates the spread of the residuals. The skewness of residuals is 0.1665204, close to zero, signifying the symmetry in the residuals. The kurtosis of residuals is 0.06426577, close to zero, indicating the lack of outlier or extreme values in the residuals.

$$\text{residual} = \text{actual} - \text{predicted}$$

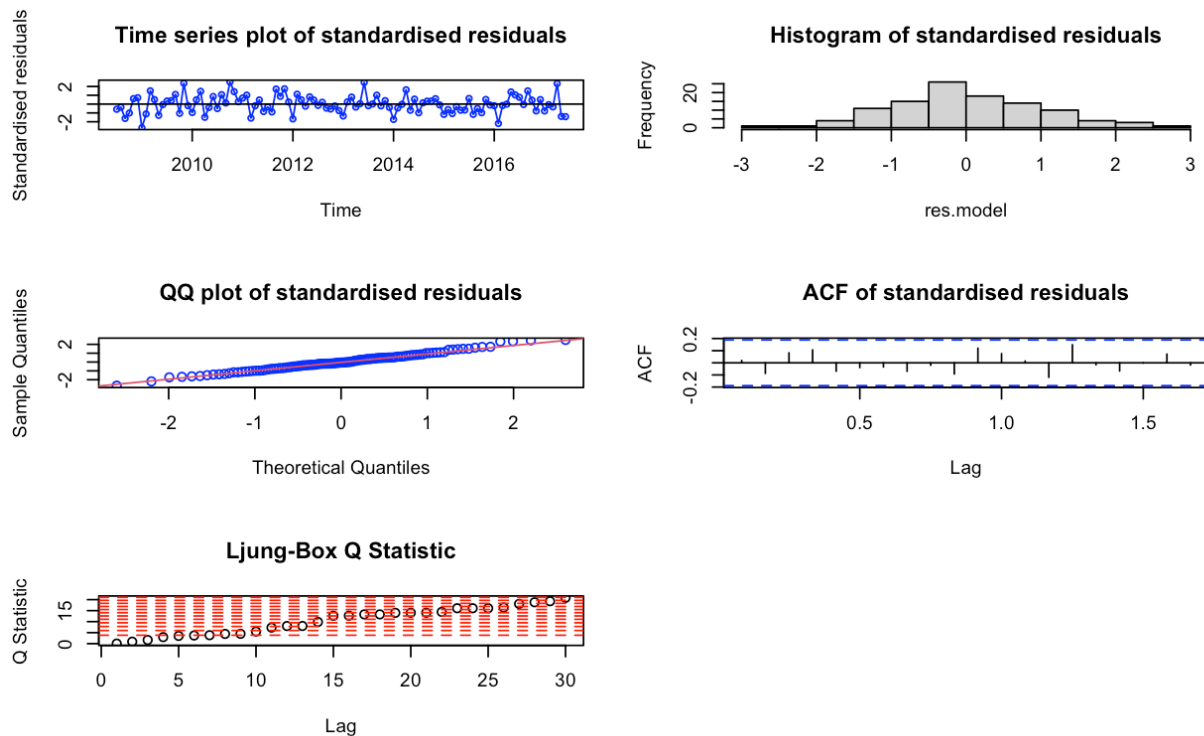


Figure 17. ARIMA (1,0,1) model's diagnostic checking plots

- **Box-Ljung test** - The p-value is 0.8421, which is greater than 0.05, suggesting that we fail to reject the null hypothesis. This indicates that the residuals are independently distributed, implying no autocorrelation exists in the residuals.
- **Shapiro-Wilk normality test** - The p-value is 0.8091, which is more significant than 0.05, leading us to fail to reject the null hypothesis. This indicates that the residuals are normally distributed.

Therefore, the ARIMA (1,0,1) model has significant parameters, and the residuals behave like white noise (normally distributed and exhibit no autocorrelation). Thus, the model provides a good fit for the data.

### 7.1.8 Model Fitting – ARIMA (1,0,4) model

The ARIMA (1,0,4) model presents the following outcomes:

- Coefficient Significance:** The z-test of coefficients determines the significance of the estimated parameters in the model. All coefficients must be significant to have a robust model. The significant coefficients are marked with asterisks. However, not all parameters in this model are significant, as their p-values are not less than the usual significance level of 0.05. Specifically,  $ma1$  ( $p=0.49273$ ) and  $ma2$  ( $p=0.97131$ ) are not statistically significant.  $ma4$  ( $p=0.08756$ ) has a marginal significance at the 10% level.

Table 5. ARIMA (1,0,4) model's z-test of coefficients

| Coefficient | Estimate   | Std. Error | z value | p-value | Significance |
|-------------|------------|------------|---------|---------|--------------|
| ar1         | 0.4893321  | 0.2013056  | 2.4308  | 0.01507 | *            |
| ma1         | -0.1437912 | 0.2096173  | -0.6860 | 0.49273 |              |
| ma2         | 0.0035096  | 0.0975820  | 0.0360  | 0.97131 |              |
| ma3         | 0.1924655  | 0.0935360  | 2.0577  | 0.03962 | *            |
| ma4         | 0.2216599  | 0.1297482  | 1.7084  | 0.08756 | .            |
| intercept   | 11.2963263 | 0.8080167  | 13.9803 | < 2e-16 | ***          |

- Residuals Analysis:** The mean of residuals is 0.00918, which is quite close to zero, suggesting no obvious bias in the model's residuals. The standard deviation of residuals is 1.00462. The skewness of residuals is 0.2589799, indicating a slight right skew. The kurtosis of residuals is 0.130119, which is quite low, suggesting that data has light tails and lacks outliers.

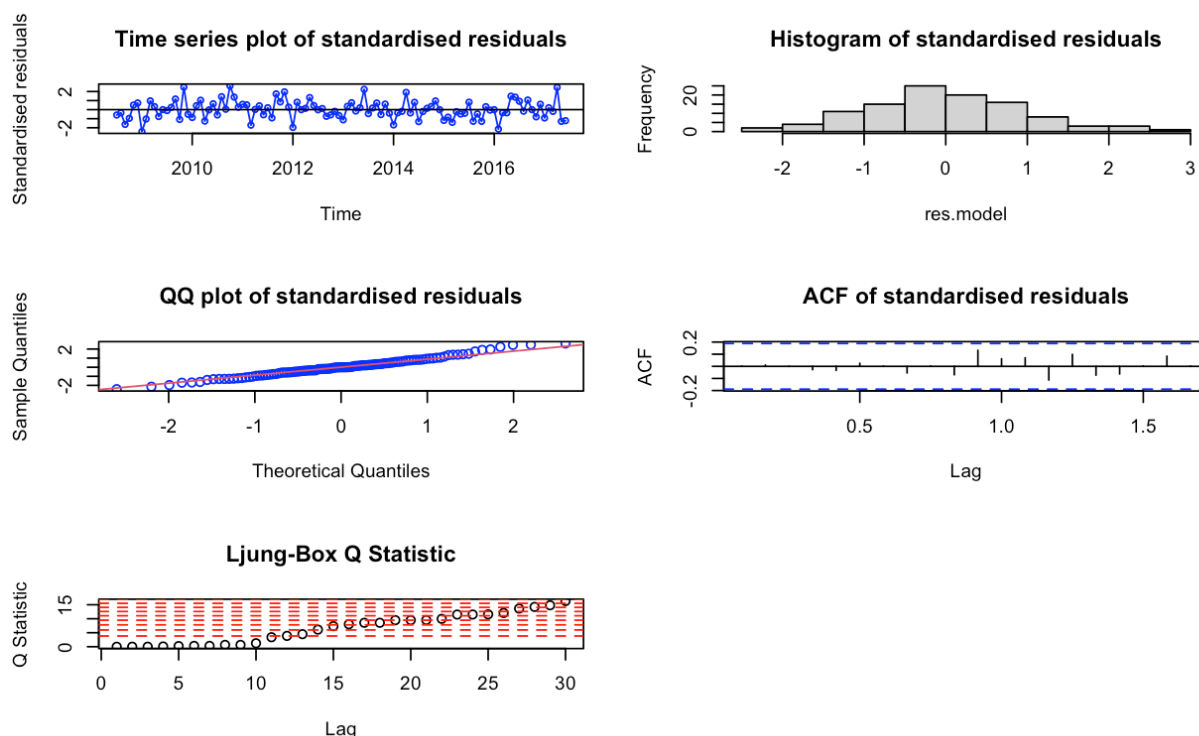


Figure 18. ARIMA (1,0,4) model's diagnostic checking plots

- Autocorrelation Check for White Noise:** The Box-Ljung test shows a p-value of 0.9843, much larger than 0.05, suggesting we do not reject the null hypothesis that the

residuals are independently distributed. There is no evidence of autocorrelation in the residuals.

- **Normality Test of Residuals:** The Shapiro-Wilk test for normality shows a p-value of 0.5956. Since the p-value is much larger than 0.05, we do not reject the null hypothesis that the residuals follow a normal distribution.

In summary, although the model's residuals pass the checks for autocorrelation and normality, the model does not have all significant coefficients. Therefore, there might be better choices than this ARIMA (1,0,4) model which is ARIMA (1,0,1).

## 7.2 GARCH model selection

After selecting ARIMA (1,0,1) as an optimal ARIMA model, we focus on the GARCH model. The Generalized Autoregressive Conditional Heteroskedasticity (GARCH) model is particularly useful when we believe volatility changes over time. We can model the variance in the error terms by using the residuals obtained from the selected ARIMA model. Like the ARIMA model selection process, we fit several potential GARCH models and evaluate their goodness of fit and complexity.

The GARCH(p, q) model incorporates both the lagged squared residuals (q terms as in the ARCH model) and the lagged conditional variances (p terms). The equation for a GARCH(p, q) model is typically given as follows:

$$\sigma^2(t|t-1) = \omega + \alpha_1 r^2(t-1) + \alpha_2 r^2(t-2) + \dots + \alpha_q r^2(t-q) + \beta_1 \sigma^2(t-1) + \beta_2 \sigma^2(t-2) + \dots + \beta_p \sigma^2(t-p)$$

Where  $\omega$  is the constant term,  $r$  is the returns,  $\alpha$  is the coefficients for the ARCH terms (squared returns), and  $\beta$  is the coefficients for the GARCH terms (lagged variances).

Identification of the orders  $p$  and  $q$  in the GARCH model is similar to the identification process in the ARIMA models. For the specification of GARCH model, the residuals of the ARIMA(1,0,1) model are used to compute absolute return series (abs.r.res.BC.weatherAU) and squared return series (sq.r.res.BC.weatherAU).

$$\text{ARMA}(\max(p,q),q) \iff \text{GARCH}(p,q)$$

1. Analyze the ACF and PACF plots of the squared or absolute value of the return series.
2. If the  $q$  value (representing the ARCH term or the number of lagged error terms in the model) is smaller than the  $p$ -value (the GARCH term or the number of lagged variances in the model), it may not appear prominently in the ACF or PACF plots. In such cases, we can first fit a GARCH( $p$ ,  $p$ ) model and then estimate the  $q$  value by examining the significance of the estimated ARCH coefficients.
3. If the time series values are independent, nonlinear transformations such as log, square, or absolute values will preserve this independence. Therefore, applying these transformations will not affect the identification of the GARCH orders.
4. Following the analysis of ACF, PACF, and EACF plots, derive the maximum values for  $p$  and  $q$  ( $\max(p, q)$ ), and establish a set of possible GARCH models.

### 7.2.1 Absolute Return Series

#### *ACF and PACF plots*

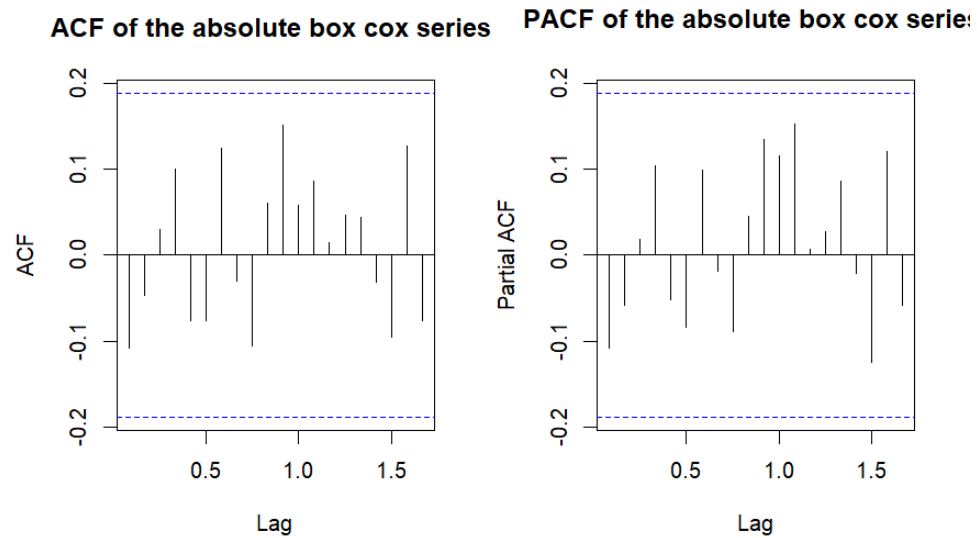


Figure 19. ACF and PACF plots for absolute return series

Since there is no significant lag so there is no GARCH model selected from ACF and PACF plots.

#### *EACF Table*

The Extended Autocorrelation Function (EACF) was then used with maximum AR and MA orders of 10.



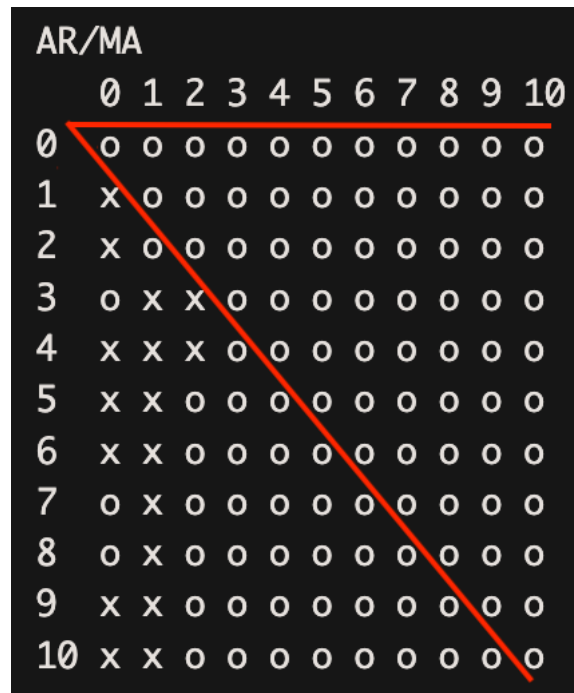


Figure 20. EACF table for absolute return series

$\max(p, q) = 0, q = 1 \implies \max(p, q = 1) = 0$ . Does not lead to any models.  
 $\max(p, q) = 0$  and  $q = 2 \implies \max(p, q = 2) = 0$ . Does not lead to any models.  
 $\max(p, q) = 1$  and  $q = 2 \implies \max(p, q = 2) = 1$ . Does not lead to any models.  
 $\max(p, q) = 1$  and  $q = 1 \implies \max(p, q = 1) = 1$ ; hence  $p = 0, 1$  and GARCH(0,1), GARCH(1,1).  
 The EACF analysis, along with specific  $\max(p, q)$  and  $q$  scenarios, was able to suggest two potential models: GARCH(0,1) and GARCH(1,1).

### BIC Table

Bayesian Information Criterion (BIC) plotting further supported the model selection process.

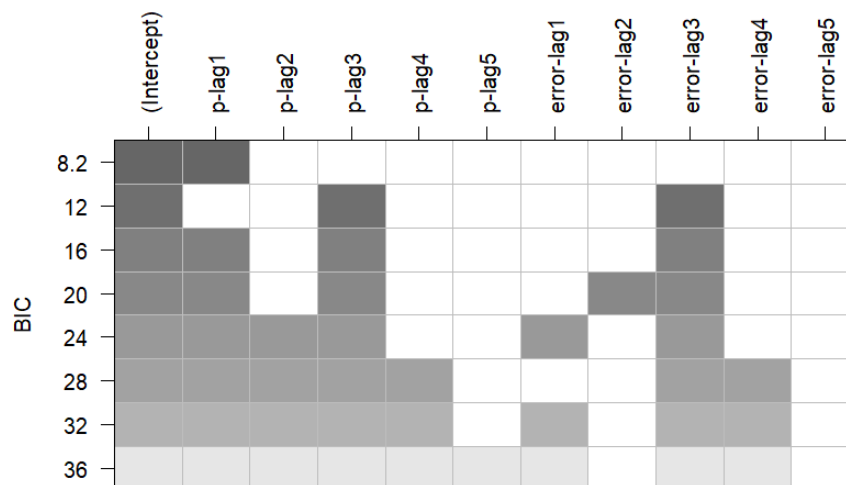


Figure 21. BIC table for absolute return series

In the Bayesian Information Criterion (BIC) plot, it was noticed that all the significant  $q$  values ( $q=3, q=4$ ) were larger than the corresponding  $p$ -lag values ( $\max(p,q) = 1$  and  $2$ ). Since such relationships do not yield a viable model, these were discarded.

Therefore, following the analyses conducted via ACF, PACF, EACF, and BIC, two GARCH models emerged as suitable candidates: GARCH(0,1) and GARCH(1,1). These models, suggested by the EACF table, provide a suitable fit for the absolute return series derived from the ARIMA model residuals. The selected models will be further evaluated for optimal fit and effectiveness.

### 7.2.2 Squared Return Series

*ACF and PACF plots*

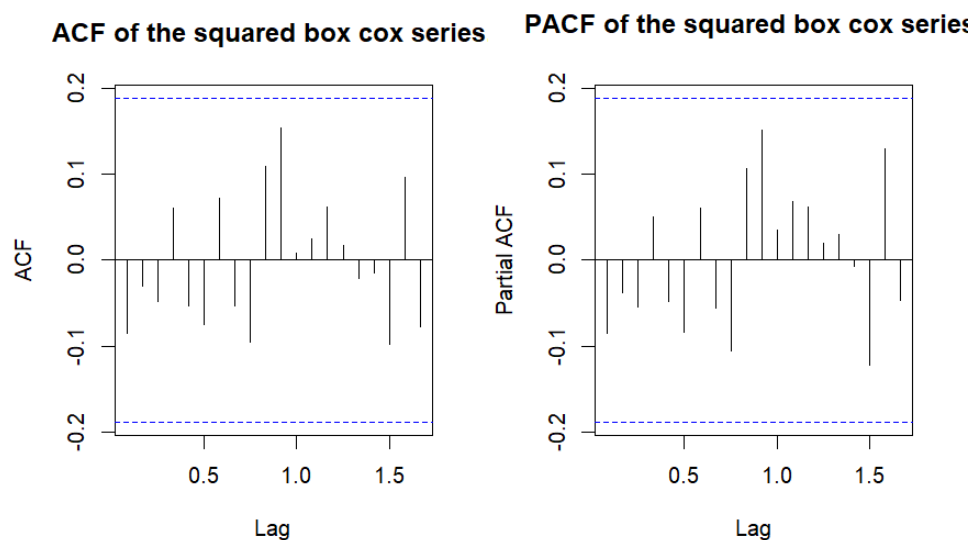


Figure 22. ACF and PACF plots for the squared return series.

Notably, no significant lags were indicated in these plots for the squared return series like absolute return. Hence no direct model recommendations could be made from this analysis.

*EACF Table*

| AR/MA | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|-------|---|---|---|---|---|---|---|---|---|---|----|
| 0     | o | o | o | o | o | o | o | o | o | o | o  |
| 1     | x | o | o | o | o | o | o | o | o | o | o  |
| 2     | x | x | o | o | o | o | o | o | o | o | o  |
| 3     | x | o | o | o | o | o | o | o | o | o | o  |
| 4     | x | o | x | o | o | o | o | o | o | o | o  |
| 5     | x | x | x | x | o | o | o | o | o | o | o  |
| 6     | x | o | x | o | o | o | o | o | o | o | o  |
| 7     | x | o | x | o | o | o | o | o | o | o | o  |
| 8     | x | x | o | o | o | o | x | o | o | o | o  |
| 9     | x | x | o | o | o | o | x | o | o | o | o  |
| 10    | x | x | x | o | o | o | x | o | o | o | o  |

*Figure 23. EACF table for the squared return series.*

- $\max(p, q) = 0, q = 1 \implies \max(p, q = 1) = 0$ . Does not lead to any models.
- $\max(p, q) = 0$  and  $q = 2 \implies \max(p, q = 2) = 0$ . Does not lead to any models.
- $\max(p, q) = 1$  and  $q = 2 \implies \max(p, q = 2) = 1$ . Does not lead to any models.
- $\max(p, q) = 1$  and  $q = 1 \implies \max(p, q = 1) = 1$ ; hence  $p = 0, 1$  and GARCH(0,1), GARCH(1,1).

The EACF analysis similar to the absolute return series pattern, along with specific  $\max(p, q)$  and  $q$  scenarios, was able to suggest two potential models: GARCH(0,1) and GARCH(1,1).

### BIC Table

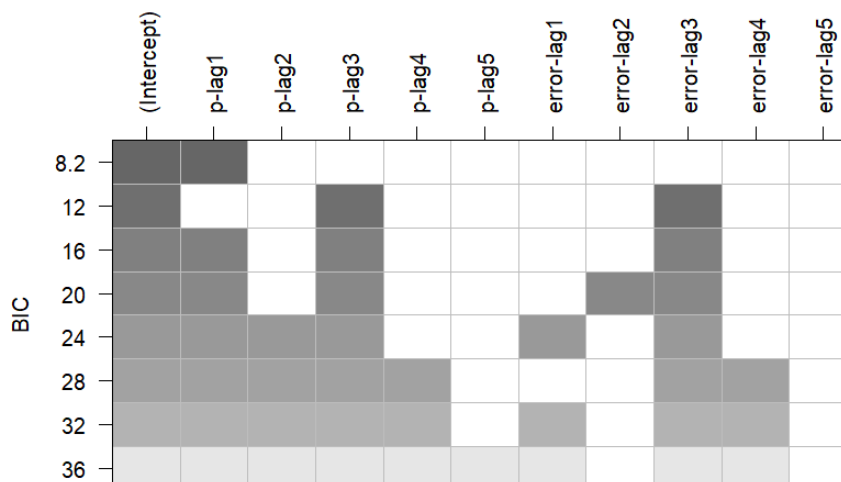


Figure 24. BIC table for the squared return series.

For  $\max(p,q) = 3$  and  $q = 3$ , the results suggested  $p = 0, 1, 2, 3$ . This indicates the additional potential models of GARCH(0,3), GARCH(1,3), GARCH(2,3), and GARCH(3,3).

In summary, our analysis suggests that for the squared return series, the GARCH models GARCH(0,1), GARCH(1,1), GARCH(0,3), GARCH(1,3), GARCH(2,3), and GARCH(3,3) may be suitable choices. We identified several potential GARCH models from our analysis of both the absolute and squared return series. These include GARCH(0,1) and GARCH(1,1) models.

In other words, a model that appears in both the absolute and squared return series has more supporting evidence and, thus, is more likely to capture the underlying structure of the data. However, we should further validate these models by assessing their residuals, conducting goodness-of-fit tests, and comparing their out-of-sample forecast accuracy. Ultimately, the goal is to select a model that provides the best trade-off between goodness-of-fit and complexity and, most importantly, has strong predictive performance.

### 7.2.3 Model Fitting and Significance Tests of Coefficients - GARCH

GARCH model summary

- $\mu$  is the mean of the series.
- $ar1$  and  $ma1$  are the coefficients for the AR (1) and MA (1) parts of the model, respectively.
- $\omega$  is the constant term in the variance equation.
- $\alpha1$  is the coefficient for the GARCH term.

$$GARCH(0,1) = ARCH(1)$$

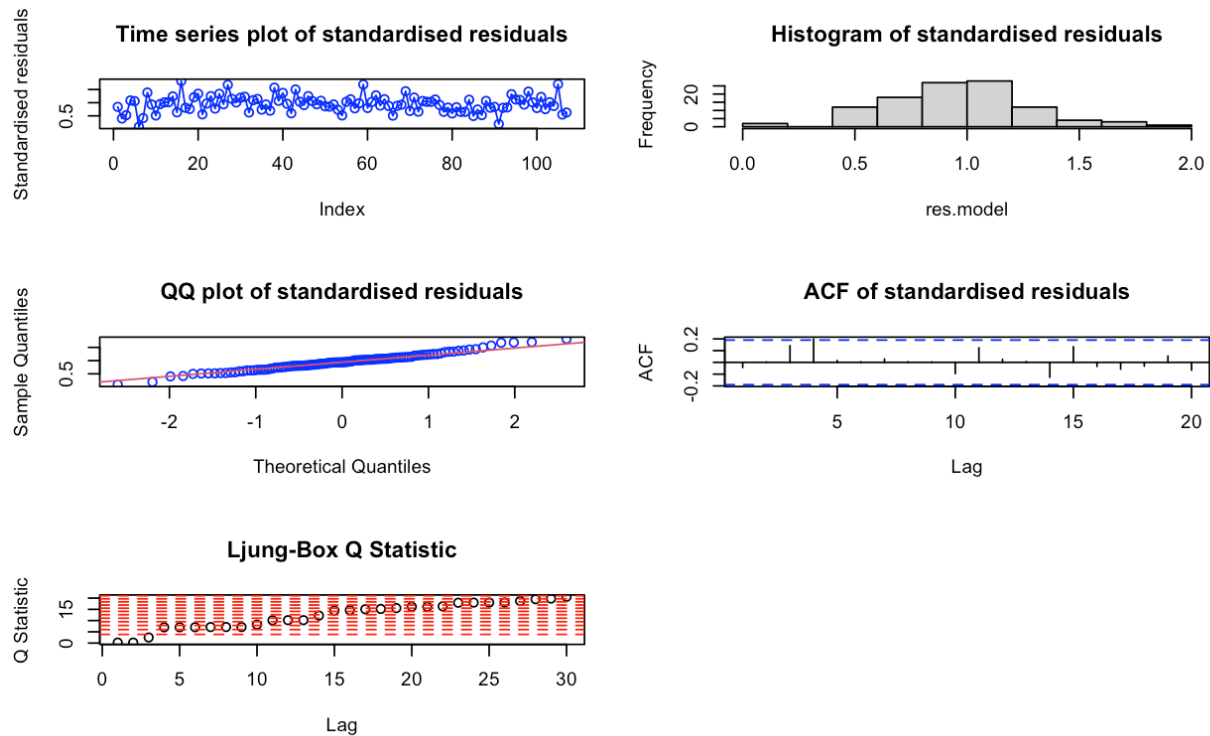


Figure 25.  $GARCH(0,1)$  model's diagnostic checking

The output above provides the results of the  $GARCH(0,1)$  model, also known as the  $ARCH(1)$  model, fit to the "BC.weatherAU" data.

- **Coefficients:** The estimate of "a0" (the constant term) is 70.7621 with a standard error of 70.3996, resulting in a t-value of 1.005 and a p-value of 0.315. The estimate of "a1" (the ARCH term) is 0.5283 with a standard error of 0.6689, resulting in a t-value of 0.790 and a p-value of 0.430. Neither coefficient is statistically significant at the conventional 5% level (as their p-values are both greater than 0.05), suggesting the coefficients are not significantly different from zero.
- **Residuals:** The mean of the residuals is 0.9504178, and the standard deviation is 0.3124393. The skewness is 0.2013518, indicating a slight right skew. The kurtosis is 0.448108, less than 3 (the kurtosis of a normal distribution), suggesting a platykurtic distribution (i.e., fewer extreme events or outliers than a normal distribution).
- **Diagnostic Tests:** The Jarque-Bera test checks if the residuals are normally distributed. The p-value is 0.4452, which is greater than 0.05, indicating that we cannot reject the null hypothesis of normality. The Box-Ljung test is used to check for autocorrelation in the residuals. The p-value is 0.236, greater than 0.05, indicating that we cannot reject the null hypothesis of no autocorrelation.
- **Shapiro-Wilk normality test:** Here, the p-value is 0.5169, which is greater than 0.05, supporting the conclusion from the Jarque-Bera test that we cannot reject the null hypothesis of normality.

The  $ARCH(1)$  model does not provide significant coefficients. Additionally, the residual diagnostics suggest that the residuals are not autocorrelated and are normally distributed,

indicating that the model fits the data adequately. However, given the lack of significant coefficients, it may be beneficial to consider other models.

### *GARCH (1,1)*

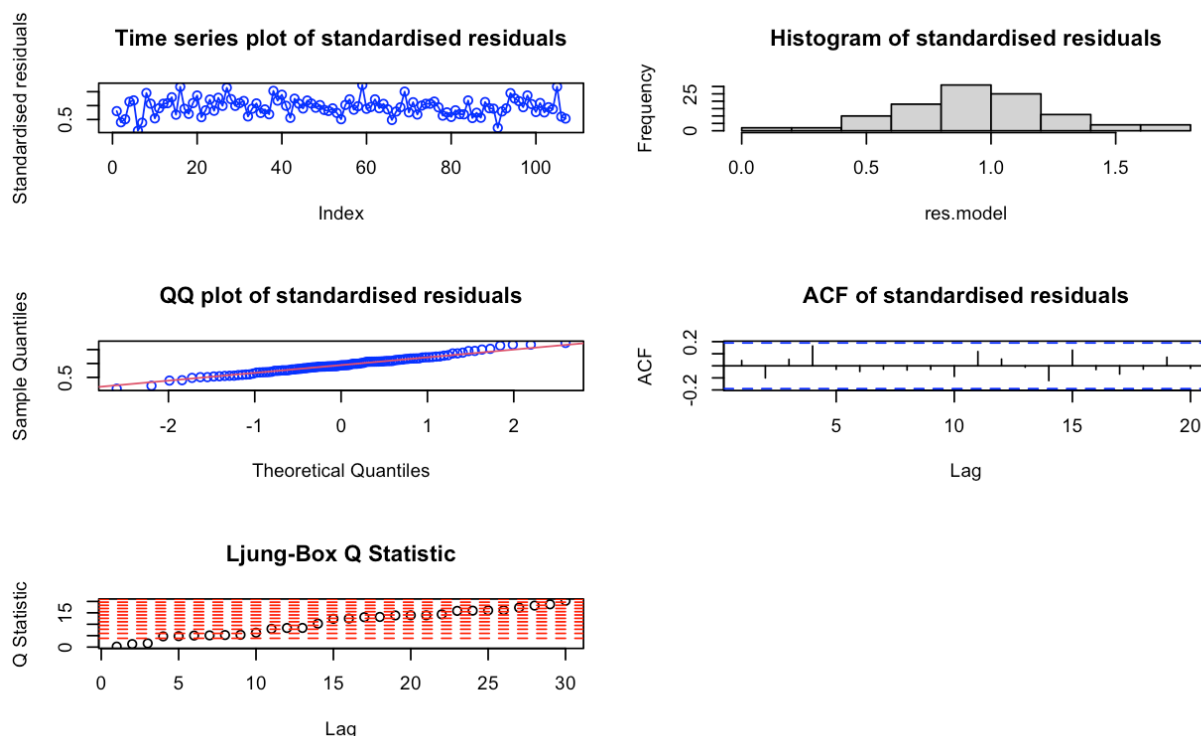


Figure 26. *GARCH(1,1) model's diagnostic checking plots*

The output above provides the results of the GARCH(1,1) model fit to the "BC.weatherAU" data.

- Coefficients:** The estimate of "a0" (the constant term) is 28.0607 with a standard error of 97.0419, resulting in a t-value of 0.289 and a p-value of 0.772. The estimate of "a1" (the GARCH term) is 0.3620 with a standard error of 0.6798, resulting in a t-value of 0.533 and a p-value of 0.594. The estimate of "b1" (the ARCH term) is 0.4514 with a standard error of 1.1093, resulting in a t-value of 0.407 and a p-value of 0.684. None of these coefficients are statistically significant at the conventional 5% level (as their p-values are all greater than 0.05), which suggests that the coefficients are not significantly different from zero.
- Residuals:** The mean of the residuals is 0.9497036, and the standard deviation is 0.3079879. The skewness is 0.1324134, indicating a slight right skew. The kurtosis is 0.2977166, less than 3 (the kurtosis of a normal distribution), suggesting a platykurtic distribution (i.e., fewer extreme events or outliers than a normal distribution).
- Diagnostic Tests:** The Jarque-Bera test is used to check if the residuals are normally distributed. The p-value is 0.7019, which is greater than 0.05, indicating that we cannot reject the null hypothesis of normality.
- Box-Ljung test:** The p-value is 0.8669, greater than 0.05, indicating that we cannot reject the null hypothesis of no autocorrelation.

- **Shapiro-Wilk normality test:** Here, the p-value is 0.656, greater than 0.05, supporting the conclusion from the Jarque-Bera test that we cannot reject the null hypothesis of normality.

The GARCH(1,1) model does not provide significant coefficients. The residual diagnostics suggest that the residuals are not autocorrelated and are normally distributed, indicating that the model fits the data adequately. However, given the lack of significant coefficients, it may be beneficial to consider other models.

### 7.3 Model Fitting: Combining ARIMA and GARCH

The final step in the model selection process is to combine the selected ARIMA and GARCH models. This creates an ARIMA-GARCH model, which incorporates both autoregressive and moving average components and conditional heteroskedasticity. Using this combined model, we can capture temporal dependencies and volatility in the data. Throughout this model selection process, it is crucial to remember that no model is perfect, and each represents a reality simplification. It is important to validate these models on historical data and new, unseen data to assess their predictive performance.

#### 7.3.1 ARIMA(1,1)+GARCH(0,1)

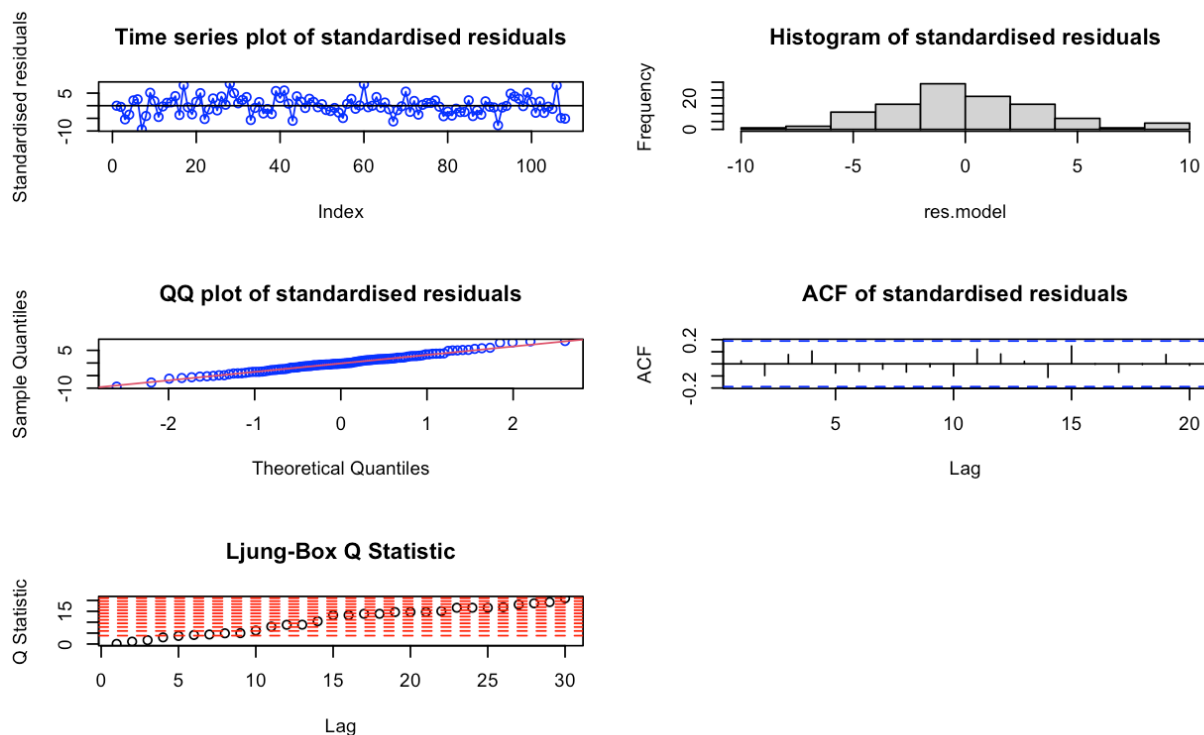


Figure 27. ARMA(1,1) + GARCH(0,1) model's diagnostic checking plots

The model specified is an ARIMA(1,1) for the mean equation and a GARCH(0,1) for the variance equation. The ARIMA(1,1) model indicates a first-order autoregressive process and a first-order moving average process for the time series data after first differencing it. The GARCH(0,1) model, on the other hand, indicates a first-order generalized autoregressive conditional heteroskedasticity model for the variance of the series.

- **Coefficients:** The coefficients of the ARIMA(1,1) model are 0.806 for AR1 and -0.482 for MA1. The coefficient of the GARCH(0,1) model is 12.209 for omega (constant term) and 0.00000001 for alpha1 (GARCH term). This model does not have any ARCH term.
- **Significance of coefficients:** The p-values suggest that AR1, MA1, and omega coefficients are significant at the 0.05 level. However, alpha1 is not significant at this level. This might suggest that there is no significant GARCH effect in the data.
- **Model diagnostics:** Several diagnostic tests are performed on the standardized residuals of the model. The p-values of these tests (Jarque-Bera, Shapiro-Wilk, Ljung-Box, and LM Arch) suggest that we do not reject the null hypotheses of normality, no autocorrelation, and no ARCH effects at the 0.05 level. This indicates that the model fits the data reasonably well.
- **Information Criterion Statistics:** The Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC) are used for model selection, with lower values indicating better models. For this model, the AIC is 5.43, and the BIC is 5.56.

The analysis of the residuals also indicates that the residuals have a mean close to zero, a standard deviation of 3.51, a slight positive skewness, and a kurtosis close to zero, suggesting that the residuals have a nearly normal distribution.

Based on this analysis, the ARIMA(1,1) + GARCH(0,1) model fits the data reasonably well, capturing both the autoregressive and moving average components in the data and the constant volatility. However, the insignificant GARCH term indicates the absence of volatility clustering in the data.

### 7.3.2 ARIMA(1,1)+GARCH(1,1)

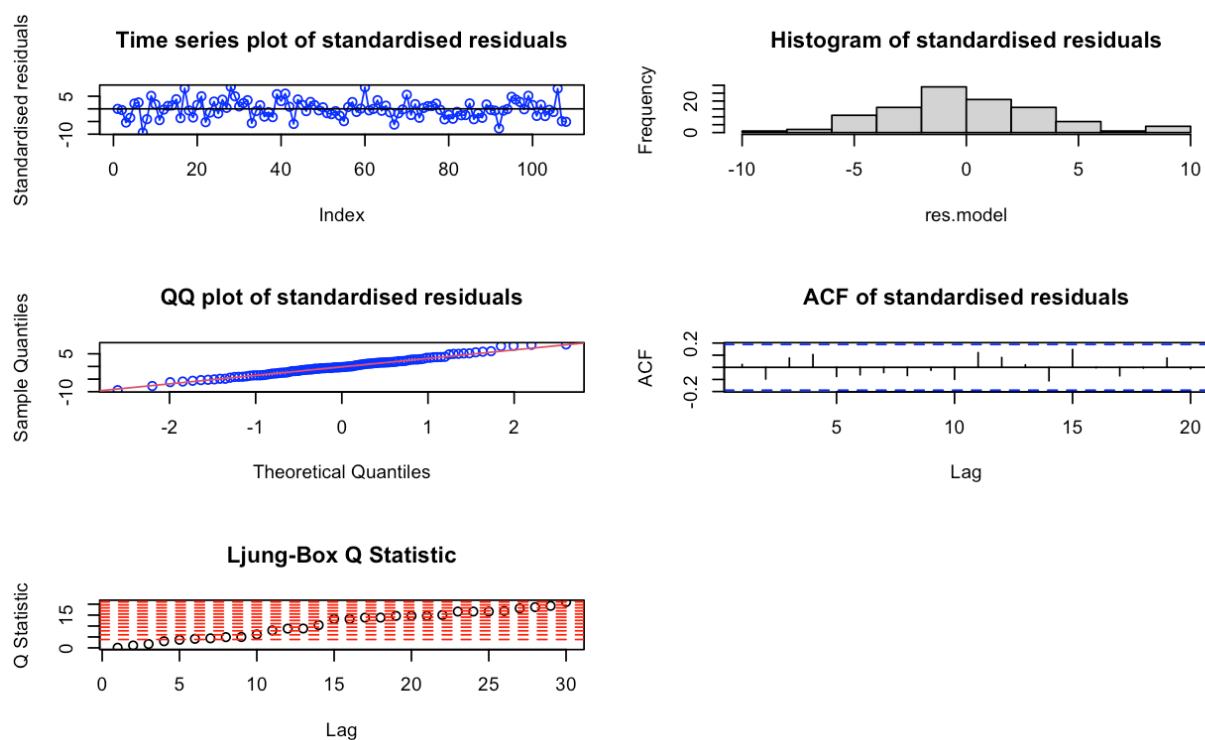


Figure 28. ARIMA(1,1)+GARCH(1,1) model's diagnostic checking plots.



This is an ARIMA(1,1)+GARCH(1,1) model. The ARIMA(1,1) part implies a first-order autoregressive process and a first-order moving average process on the differenced series. The GARCH(1,1) part implies a first-order autoregressive and a first-order moving average process for the variance of the series.

**Coefficients:** The coefficients for the ARIMA(1,1) part are 0.806 for AR1 and -0.482 for MA1. The coefficients for the GARCH(1,1) part are 5.553 for omega (constant term), 0.00000001 for alpha1 (GARCH term), and 0.548 for beta1 (ARCH term).

**Significance of coefficients:** The p-values for the AR1 and MA1 coefficients indicate that these coefficients are significant at the 0.05 level. The omega, alpha1, and beta1 coefficients have no standard errors computed. Hence their significance levels cannot be determined.

**Model diagnostics:** Several tests are conducted on the residuals of the model. The Jarque-Bera and Shapiro-Wilk tests suggest that the residuals are normally distributed as we do not reject the null hypothesis of normality at the 0.05 level. The Ljung-Box tests suggest no autocorrelation in the residuals or squared values up to lag 20. This indicates that the model is adequately capturing the autocorrelation in the data. The LM Arch test suggests no ARCH effect in the residuals.

**Information Criterion Statistics:** The Akaike Information Criterion (AIC) is 5.45, and the Bayesian Information Criterion (BIC) is 5.60. These measures are used for model selection, with lower values indicating a better fit.

The analysis of the residuals indicates that the mean is approximately zero, the standard deviation is 3.51, the skewness is slightly positive, and the kurtosis is close to zero. This suggests that the residuals have a distribution that is close to normal.

In summary, the ARIMA(1,1)+GARCH(1,1) model fits the data well, capturing both the series' autoregressive and moving average components and variance. However, the insignificance of the GARCH and ARCH terms (due to unavailable standard errors) may imply a need for further investigation.

In this analysis, we observed that both the ARMA (1,1) +GARCH (1,0) model and the ARIMA (1, 0, 1) model fit the data satisfactorily, exhibiting similar residual patterns.

Observations from various plots were as follows:

- Time Series Plot: The residuals appeared randomly scattered around the zero-horizontal line, suggesting no discernible patterns or trends.
- Histogram: Residual distribution was symmetrical with no prominent outliers, suggesting normality.
- QQ Plot: Points approximately followed the reference line, indicating that the residuals are normally distributed.
- ACF Plot: The absence of significant autocorrelations at all lags suggests that the models adequately capture the data structure.

In addition, we conducted two statistical tests:

- Shapiro-Wilk Test for Normality: The p-values obtained were above the 0.05 threshold, which allows us to assume that the residuals are normally distributed.
- Ljung-Box Test: The Q statistics for the residuals did not exceed the chi-square quantiles at a 95% confidence level at any lag, suggesting the residuals are randomly distributed, i.e., there is no autocorrelation.

## 8. Further Diagnostics Checking ARIMA and GARCH

### 8.1. AIC and BIC Scores

The table presents the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC) for two models: ARMA(1,1) + GARCH(0,1) and ARMA(1,1)+GARCH(1,1). These metrics are useful for model selection.

Table 6. AIC and BIC tables evaluation metrics for comparing ARMA(1,1) + GARCH(0,1) and ARMA(1,1)+GARCH(1,1)

| Model                | AIC      | BIC      |
|----------------------|----------|----------|
| ARMA(1,1)+GARCH(0,1) | 5.432610 | 5.556782 |
| ARMA(1,1)+GARCH(1,1) | 5.451115 | 5.600122 |

AIC and BIC are both measures of the goodness of fit of an estimated statistical model and can be used for model selection. Both consider the model's complexity and the likelihood of the model given the data. However, they penalize model complexity differently: BIC penalizes model complexity more heavily than AIC.

In this case, the ARMA(1,1)+GARCH(0,1) model has a lower AIC and BIC compared to the ARMA(1,1)+GARCH(1,1) model. The lower the AIC and BIC, the better the model is considered, suggesting a good balance between model fit and complexity. Therefore, based on both AIC and BIC, the ARMA(1,1)+GARCH(0,1) model would be preferred.

### 8.2 ME, RMSE, and MAE Scores

Table 7. ME (Mean Error), RMSE (Root Mean Square Error), and MAE (Mean Absolute Error) are metrics used to evaluate the prediction performance of models.

| Model                | ME         | RMSE     | MAE      |
|----------------------|------------|----------|----------|
| ARMA(1,1)+GARCH(0,1) | 0.07373131 | 3.514517 | 2.769798 |
| ARMA(1,1)+GARCH(1,1) | 0.07389439 | 3.514516 | 2.769788 |

1. **Mean Error (ME):** This is the average prediction error. The closer to zero, the better. Here, both models have a very similar ME, around 0.073. Neither model outperforms the other based on this criterion.
2. **Root Mean Squared Error (RMSE):** This measure indicates the average difference between the predicted and actual values, with more weight to large errors due to squaring. The model with the smaller RMSE value is better. In this case, both models have the same RMSE (~3.5145), indicating similar performance.
3. **Mean Absolute Error (MAE):** The average absolute differences between the predicted and actual values. It measures the average magnitude of errors in a set of predictions without considering their direction. It is less sensitive to outliers compared to RMSE. The model with the smaller MAE is better. As with the previous measures, the MAE for both models is nearly identical (~2.7698), indicating similar predictive performance.

Therefore, both the ARMA (1,1) + GARCH(0,1) and ARMA(1,1)+GARCH(1,1) models have very similar performance based on these metrics. It is hard to recommend one over the other based on these results.

## 9. Forecasting

Table 8. The table below displays the forecasted Series and Sigma (standard deviation of the forecast errors) for each of the ten periods (T+1 through T+10).

| Forecast Period | Series | Sigma |
|-----------------|--------|-------|
| T+1             | 9.793  | 3.275 |
| T+2             | 10.088 | 3.273 |
| T+3             | 10.320 | 3.272 |
| T+4             | 10.502 | 3.270 |
| T+5             | 10.645 | 3.268 |
| T+6             | 10.757 | 3.267 |
| T+7             | 10.845 | 3.265 |
| T+8             | 10.915 | 3.263 |
| T+9             | 10.969 | 3.262 |
| T+10            | 11.012 | 3.260 |

The forecast shows an expected upward trend in the series, representing the monthly total rainfall. The Sigma value, representing the standard deviation of forecast errors, remains relatively stable over the forecasted period, implying consistent forecast precision. The graphical representation of this forecast further visualizes this trend and showcases the estimated rainfall changes over the next ten months.

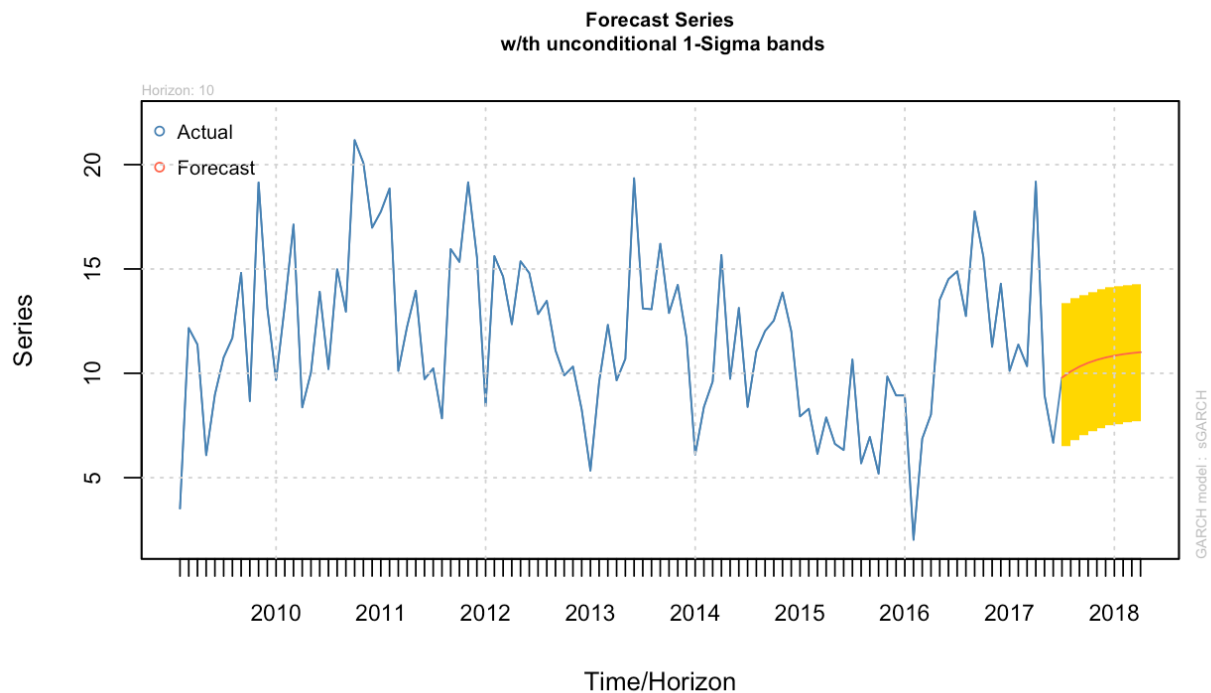


Figure 29. The forecast was set with a horizon of 10 months starting from June 2017.

After applying the GARCH model and the forecast generation, we reversed the Box-Cox transformation to bring our predictions back to the original scale of the data - the monthly total

rainfall in Melbourne. The reverse transformation process is essential as it transforms our forecasted values back to a meaningful context.

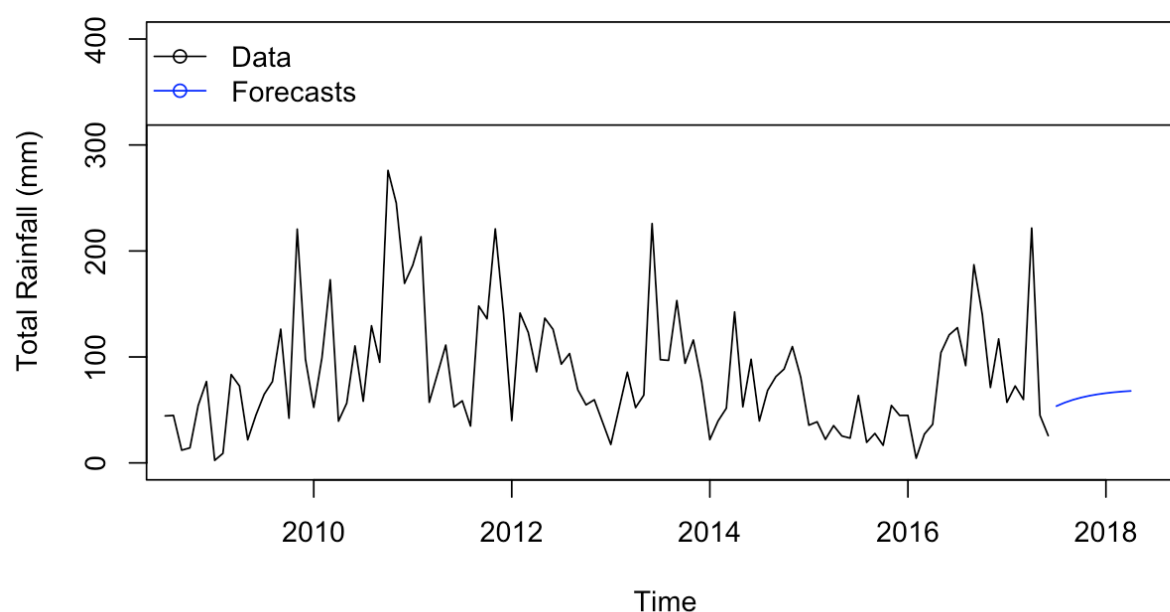
*Table 9. The actual forecasted values for the ten time points following June 2017.*

| Time | Forecast |
|------|----------|
| T+1  | 53.61515 |
| T+2  | 56.89220 |
| T+3  | 59.54790 |
| T+4  | 61.68463 |
| T+5  | 63.39442 |
| T+6  | 64.75686 |
| T+7  | 65.83901 |
| T+8  | 66.69640 |
| T+9  | 67.37439 |
| T+10 | 67.90971 |

The table can be interpreted as the model's prediction of the values for the specified data at each future time point. For instance, at time T+1, the model forecasts a value of 53.61515. By time T+10, the forecasted value has increased to 67.90971.

Over the ten time points, the model predicts a generally increasing trend in the data, with the value at each subsequent time point forecasted to be higher than the one before it. This indicates that the model predicts a positive trend in this data set over time. These are predictions, and actual results may vary depending on factors not accounted for in the model.

### Forecasts from the cosine wave fitted to the total rainfall series.



*Figure 30. Forecasts from the cosine wave fitted to the total rainfall series.*

These forecasted figures represent the estimated total rainfall in Melbourne for each of the next ten months, starting from July 2017. We can observe a gradual increase in the rainfall levels over the forecasted period. This data is a crucial element of our analysis and clearly depicts the expected weather patterns, which could be significant for weather-sensitive sectors like agriculture, construction, and tourism.

## 10. Results

The time series data was analyzed using ARMA and GARCH models. Our model selection criterion was based on the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC). Two models: ARMA(1,1) + GARCH(0,1) and ARMA(1,1) + GARCH(1,1), were found to be the most optimal based on these criteria.

The ARMA(1,1) + GARCH(0,1) model had an AIC of 5.432610 and a BIC of 5.556782, while the ARMA(1,1) + GARCH(1,1) model had an AIC of 5.451115 and a BIC of 5.600122.

Additionally, the forecast accuracy of these models was evaluated. For the ARMA(1,1) + GARCH(0,1) model, the Mean Error (ME) was 0.07373131, the Root Mean Square Error (RMSE) was 3.514517, and the Mean Absolute Error (MAE) was 2.769798. For the ARMA(1,1) + GARCH(1,1) model, the ME was 0.07389439, the RMSE was 3.514516, and the MAE was 2.769788.

A ten-point forecast was also performed, revealing a general upward trend.

## 11. Discussion

An intriguing finding from our time series analysis involves the disparity in the residuals from the applied models. The ARMA and GARCH models displayed a notably wider range of residuals (-10 to +10), while the ARIMA model produced a narrower range (-3 to +3). This suggests higher volatility or larger forecasting errors in the ARMA and GARCH models compared to the ARIMA model.

Several factors could account for this difference. Firstly, model fit is a crucial consideration - the ARMA and GARCH models may not capture all the underlying patterns in the data as effectively as the ARIMA model, leading to larger errors. Secondly, the inherent simplicity of ARMA and GARCH models might restrict their ability to capture the data's complexity, which an ARIMA model could handle better, resulting in smaller residuals.

Furthermore, the nature of the data itself must be considered. The data, characterized by its inherent volatility, might need to be fully encapsulated by the ARMA and GARCH models, despite these models being designed to handle volatile data, leading to larger residuals.

In choosing the most suitable model for forecasting, it is crucial not to rely solely on the size of residuals. Factors like interpretability, prediction accuracy, and compatibility with the underlying assumptions must be considered. Additional model diagnostics are recommended to ensure the best model fit for the data.

## 12. Conclusion

The results of this analysis provide significant insights into the time series data. Based on our selected criteria, the ARMA(1,1) + GARCH(0,1) model is optimal for this data set. The upward

trend predicted by our forecast also provides valuable information for understanding the expected behavior of this time series data in the future.

### 13. Recommendations

Considering the results of this analysis, it is recommended to utilize the ARMA(1,1) + GARCH(0,1) model for forecasting this time series data. Considering the projected upward trend in future decision-making processes related to this data is also suggested.

However, it is essential to note that any model's predictive power is not absolute. Therefore, while these models provide valuable insights, they should be used with other analyses and expert judgment forms. Furthermore, models should be periodically re-evaluated and updated as more data becomes available.

## 14. Appendices: R code

### 14.1 Clean the data.

```

---
title: "clean_data"
author: "Thu Tran, Sasha Nazareth, Pranav Barve"
date: "`r Sys.Date()`"
output: pdf_document
---

```{r Load libraries}
# Load necessary packages
library(dplyr)
library(lubridate) # for month() and year()
library(stringr)
library(zoo)
```

```{r}
# Load dataset
data <- read.csv('./data/raw/weatherAUS.csv')

# Select the Location, Rainfall, and Date columns
data <- data %>% select(Location, Rainfall, Date)

# Convert Date to R date format
data$Date <- as.Date(data$Date)

# Extract year from Date and add a new column for the year
data <- data %>% mutate(Year = year(Date))

# Check missing values and summary before cleaning
missing_values_before <- data %>%
  summarise(across(everything(), ~sum(is.na(.)), .names = "missing_{.col}"))

print("Missing values before cleaning:")
print(missing_values_before)

print("Summary before cleaning:")
summary(data)

# Fill missing values for Rainfall with median value grouped by location
data_cleaned <- data %>%
  group_by(Location) %>%
  mutate(Rainfall = ifelse(is.na(Rainfall), median(Rainfall, na.rm = TRUE), Rainfall)) %>%

```

```

ungroup()

# If there are still NA values in Rainfall, fill them with the median across all locations
data_cleaned$Rainfall[is.na(data_cleaned$Rainfall)] <- median(data_cleaned$Rainfall, na.rm
= TRUE)

#data_cleaned <- data_cleaned %>% filter(Location == "Melbourne")

# Filter for locations that contain the word "Melbourne"
data_cleaned <- data_cleaned %>% filter(str_detect(Location, "Melbourne"))

# Drop the Location column
data_cleaned <- data_cleaned %>% select(-Location)

# Reorder the columns so that Date is the first column
data_cleaned <- data_cleaned %>% select(Date, Rainfall)

# Check missing values and summary after cleaning
missing_values_after <- data_cleaned %>%
  summarise(across(everything(), ~sum(is.na(.)), .names = "missing_{.col}"))

print("Missing values after cleaning:")
print(missing_values_after)

print("Summary after cleaning:")
summary(data_cleaned)
```

```{r}
# Extract year and month from Date and add new columns for them
data_cleaned <- data_cleaned %>% mutate(Year = year(Date), Month = month(Date))

# Group by Year and Month, and calculate the sum of Rainfall for each group
monthly_rainfall <- data_cleaned %>%
  group_by(Year, Month) %>%
  summarise(Total_Rainfall = round(sum(Rainfall),2), .groups = 'drop')

# Find months with 0mm rainfall
zero_rainfall_months <- monthly_rainfall %>%
  filter(Total_Rainfall == 0)

# Print months with 0mm rainfall
print(zero_rainfall_months)
```

```



```

```{r}
# Add a YearMonth column by extracting the year and month from the Date column
data_cleaned <- data_cleaned %>% mutate(YearMonth = format(Date, "%Y-%m"))

# Aggregate the data on a monthly basis
data_cleaned <- data_cleaned %>%
  group_by(YearMonth) %>%
  summarise(TotalRainfall = sum(Rainfall, na.rm = TRUE))

# get the number of rows
row_count <- nrow(data_cleaned)
# Print the number of rows
print(paste("Number of rows:", row_count))

# Export the cleaned data to the 'data/cleaned' folder
write.csv(data_cleaned, "../data/cleaned/cleaned_weatherAUS.csv", row.names = FALSE)

# display first 3 rows of data frame
head(data_cleaned, n=3)

# display summary
print(paste0("Summary rainfall on the monthly basis:", row_count))
summary(data_cleaned)
```

```{r}
# Convert YearMonth to Date format for the purpose of creating sequence
data_cleaned$YearMonth_Date <- as.Date(paste(data_cleaned$YearMonth, "-01", sep = ""),
format = "%Y-%m-%d")

# Create a data frame with all dates in the range
all_dates <- data.frame(YearMonth_Date = seq(min(data_cleaned$YearMonth_Date),
max(data_cleaned$YearMonth_Date), by = "month"))

# Merge your data with all_dates to have NA when there is no data
data_cleaned <- merge(data_cleaned, all_dates, all = TRUE, by = "YearMonth_Date")

# Order data by YearMonth_Date
data_cleaned <- data_cleaned[order(data_cleaned$YearMonth_Date),]

# Apply the zoo package's na.approx function to interpolate missing values
data_cleaned$TotalRainfall <- zoo::na.approx(data_cleaned$TotalRainfall, na.rm = FALSE)

# Update the YearMonth column based on YearMonth_Date
data_cleaned$YearMonth <- format(data_cleaned$YearMonth_Date, "%Y-%m")

```

```

# Now remove the YearMonth_Date column as it's no longer needed
data_cleaned$YearMonth_Date <- NULL

# Verify that there are no more missing values
sum(is.na(data_cleaned$TotalRainfall))
```

```{r}
# Export the cleaned data to the 'data/cleaned' folder
write.csv(data_cleaned, "../data/cleaned/cleaned_weatherAUS.csv", row.names = FALSE)

# Display the first 3 rows of the data frame
head(data_cleaned, n=3)

# Display summary
print("Summary of rainfall on a monthly basis:")
summary(data_cleaned)
```

```

## 14.2 Analyse data

```

---
title: "analysis"
author: "Thu Tran, Sasha Nazareth, Pranav Barve"
date: "`r Sys.Date()`"
output: pdf_document
---

```{r import libraries}
rm(list=ls())

library(TSA)
library(tseries)
library(ggplot2)
library(lmtest)
library(forecast)
library(fUnitRoots)
library(fGarch)
library(rugarch)
```

```{r define functions}

plot_time_series <- function(data, time_series_title, acf_title, pacf_title) {

```

```

plot(data, type='o', ylab='Total Rainfall (mm)',
     main=time_series_title, col="blue")

par(mfrow=c(1,2))
acf(data, main=acf_title)
pacf(data, main=pacf_title)
par(mfrow=c(1,1))
}

plot_qq <- function(data, plot_title) {
  # Conduct Shapiro-Wilk test
  shapiro_test <- shapiro.test(data)

  # Plot QQ plot
  qqnorm(data, main = paste(plot_title, "\n\n Shapiro-Wilk Test: W =",
                             round(shapiro_test$statistic, 4),
                             "p-value =", round(shapiro_test$p.value, 4)), col="blue")
  qqline(data, col = 2)
}

# Function for stationarity tests
stationarity_tests <- function(timeseries) {

  # Augmented Dickey-Fuller Test
  adf_test <- adf.test(timeseries, alternative = "stationary")
  cat("\nAugmented Dickey-Fuller Test\n")
  print(adf_test)
  if(adf_test$p.value < 0.05) {
    print("Output: Stationary")
  } else {
    print("Output: Non-Stationary")
  }
}

# Phillips-Perron Unit Root Test
pp_test <- pp.test(timeseries)
cat("\nPhillips-Perron Unit Root Test\n")
print(pp_test)
if(pp_test$p.value < 0.05) {
  print("Output: Stationary")
} else {
  print("Output: Non-Stationary")
}
}

# KPSS Test for Level Stationarity
kpss_test <- kpss.test(timeseries, null = "Level")

```

```

cat("\nKPSS Test for Level Stationarity\n")
print(kpss_test)
if(kpss_test$p.value > 0.05) { # Please note that for the KPSS test, the null hypothesis is that
the series is stationary
  print("Output: Stationary")
} else {
  print("Output: Non-Stationary")
}
}

sort.score <- function(x, score = c("bic", "aic")){
  if (score == "aic"){
    x[with(x, order(AIC)),]
  } else if (score == "bic") {
    x[with(x, order(BIC)),]
  } else {
    warning('score = "x" only accepts valid arguments ("aic","bic")')
  }
}

residual.analysis <- function(model, std = TRUE, start = 2, class =
c("ARIMA", "GARCH", "ARMA-GARCH", "garch", "fGARCH")[1]){
  library(TSA)
  if (class == "ARIMA"){
    if (std == TRUE){
      res.model = rstandard(model)
    } else {
      res.model = residuals(model)
    }
  } else if (class == "GARCH"){
    res.model = model$residuals[start:model$n.used]
  } else if (class == "garch"){
    res.model = model$residuals[start:model$n.used]
  } else if (class == "ARMA-GARCH"){
    res.model = model@fit$residuals
  } else if (class == "fGARCH"){
    res.model = model@residuals
  } else {
    stop("The argument 'class' must be either 'ARIMA' or 'GARCH' ")
  }

  cat("Mean of residuals: ", mean(res.model), "\n")
  cat("Standard deviation of residuals: ", sd(res.model), "\n")
  cat("Skewness of residuals: ", skewness(res.model), "\n")
  cat("Kurtosis of residuals: ", kurtosis(res.model), "\n")
}

```

```

# Ljung-Box test
lb_test <- Box.test(res.model, type = "Ljung-Box")
print(lb_test)

par(mfrow=c(3,2))
plot(res.model,type='o',ylab='Standardised residuals', main="Time series plot of standardised
residuals", col="blue")
abline(h=0)
hist(res.model,main="Histogram of standardised residuals")
qqnorm(res.model,main="QQ plot of standardised residuals", col="blue")
qqline(res.model, col = 2)
acf(res.model,main="ACF of standardised residuals")
print(shapiro.test(res.model))
# Ljung-Box test
lags <- seq(from = 1, to = 30, by = 1)
lb_values <- sapply(lags, function(lag) Box.test(res.model, lag = lag, type = "Ljung-
Box")$statistic)
plot(lags, lb_values, type = 'b', main = "Ljung-Box Q Statistic", xlab = "Lag", ylab = "Q
Statistic")
# add reference line
abline(h=qchisq(0.95, df=lags), col="red", lty=2)
par(mfrow=c(1,1))
}

# Function to fit ARMA+GARCH models and compute accuracy
fit_armagarch_models <- function(data, arma_orders, garch_orders) {
  models <- list()
  accuracy_measures <- list()

  for (i in 1:length(arma_orders)) {
    spec <- ugarchspec(variance.model = list(model = "sGARCH",
   garchOrder = garch_orders[[i]]),
                      mean.model = list(armaOrder = arma_orders[[i]],
   include.mean = TRUE),
                      distribution.model = "std")
    model <- ugarchfit(spec, data)
    models[[paste0("ARMA(", paste(arma_orders[[i]], collapse = ","),
                    ")+GARCH(", paste(garch_orders[[i]], collapse = ","),")")] <- model

    residuals <- residuals(model, standardize=FALSE)
    ME <- mean(residuals)
    RMSE <- sqrt(mean(residuals^2))
    MAE <- mean(abs(residuals))

    accuracy_measures[[paste0("ARMA(", paste(arma_orders[[i]], collapse = ","),

```

```

    ")+GARCH(", paste(garch_orders[[i]], collapse = ",", ")")"] <- c(ME,
RMSE, MAE)
  }

  df_accuracy <- data.frame(do.call(rbind, accuracy_measures))
  colnames(df_accuracy) <- c("ME", "RMSE", "MAE")

  return(df_accuracy)
}

'''

'''{r data overview}
# Load dataset
weatherAUS <- read.csv('../data/cleaned/cleaned_weatherAUS.csv')
# display first 3 rows of data frame
head(weatherAUS, n=3)
# class of the data
class(weatherAUS)
summary(weatherAUS)
'''

'''{r convert to time series object}
weatherAUS_ts <- weatherAUS$TotalRainfall

# Convert the vector to a time series object
weatherAUS_ts <- ts(weatherAUS_ts, start=c(2008,7),frequency = 12)
class(weatherAUS_ts)
weatherAUS_ts
'''

'''{r time series plot}
plot_time_series(weatherAUS_ts, "Time Series Plot of Monthly Total Rainfall (2008-2017)",
"ACF plot", "PACF plot")
'''

'''{r time series with labels}
plot(weatherAUS_ts,type='l',ylab='Total Rainfall (mm)', main = "Time series plot
of Monthly Total Rainfall (2008-2017) with month labels.", col="blue")
points(y=weatherAUS_ts,x=time(weatherAUS_ts), pch=as.vector(season(weatherAUS_ts)))
'''

```

```

```{r scatter plot to show the relationship between pairs of consecutive Arctic Sea Ice
Minimum Extent}
plot(y=weatherAUS_ts,x=zlag(weatherAUS_ts),ylab='Total Rainfall (in mm)', xlab='Previous
month', main = "Scatter plot of in consecutive months", col="blue")

# the correlation show linear relationship between pairs of consecutive Arctic Sea Ice
Minimum Extent
y = weatherAUS_ts      # Read the annual data into y
x = zlag(weatherAUS_ts) # Generate first lag of the annual series
index = 2:length(x)    # Create an index to get rid of the first NA value in x
cor(y[index],x[index]) # Calculate correlation between numerical values in x and y

```

```{r Normality test of the original time series data}
plot_qq(weatherAUS_ts, "Q-Q plot")
shapiro.test(weatherAUS_ts)
```

```{r Stationary Check: of the original time series data}
stationarity_tests(weatherAUS_ts)
# p-value = 0.01819: stationary
```

```{r Box-Cox transformation}
BC = BoxCox.ar(weatherAUS_ts)
BC$ci # 0.2 and 0.5
lambda <- BC$lambda[which(max(BC$loglike) == BC$loglike)]
lambda # 0.4
BC.weatherAU = (weatherAUS_ts^lambda-1)/lambda

plot_time_series(BC.weatherAU, "Time series plot of BC transformed
melbourne rainfall", "ACF plot of BC transformed
melbourne rainfall", "PACF plot of BC transformed
melbourne rainfall")

plot_qq(BC.weatherAU, "QQ plot after Box-Cox transformation")

shapiro.test(BC.weatherAU) # W = 0.99327, p-value = 0.8783

```

```

'''
'''{r export comparison box-cox transformation before and after}
# Set the output file and dimensions (width, height) in pixels
png("BC_weatherAUS_ts.png", width = 1200, height = 600)

# Set up the layout for side-by-side plots
par(mfrow = c(1, 2))

# Create a time series plot of the original data
plot(weatherAUS_ts,
      type = "l",
      col = "black",
      main = "Original Melbourne Total Rainfall Time Series",
      xlab = "Time",
      ylab = "Melbourne Total Rainfall (mm)")

# Create a time series plot of the Box-Cox transformed data
plot(BC.weatherAU,
      type = "l",
      col = "blue",
      main = "Box-Cox transformation Melbourne Total Rainfall Time Series",
      xlab = "Time",
      ylab = "Box-Cox transformation Melbourne Total Rainfall (mm)")

# Reset the layout
par(mfrow = c(1, 1))
'''

'''{r}
McLeod.Li.test(y=BC.weatherAU,main="P-values of McLeod-Li test statistics
              for Transformed Total Rainfall Time Series")
# McLeod-Li test is significant at 5% level of significance for all lags. This gives a strong idea
about existence of volatility clustering.
# confirm changing variance since all the p-value lower than 0.05
'''

# ARIMA model specification

'''{r Model specification: with the box cox transformation}
# w the box cox series
par(mfrow=c(1,2))
acf(BC.weatherAU, main="ACF of the box cox series.")
pacf(BC.weatherAU, main="PACF of the box cox series.")
par(mfrow=c(1,1))
# from the ACF and PACF i get possible models as:

```



```

# ARIMA(p,d,q): ARIMA (3,0,4) and ARIMA (1,0,4).

eacf(BC.weatherAU, ar.max = 10, ma.max = 10)
# from EACF
# ARIMA (p, d, q): ARIMA (1,0,1), ARIMA (1,0,2), ARIMA (2,0,2)

# from the BIC plot
res = armasubsets(y=BC.weatherAU,nar=5,nma=5,y.name='p',ar.method='ols')
plot(res)

# from the BIC i get possible models as:
# p- 1,3
# q- 1,4,5
# ARIMA(p,d,q):
ARIMA(1,0,1),ARIMA(1,0,4),ARIMA(1,0,5),ARIMA(3,0,1),ARIMA(3,0,4),ARIMA(3,0,5)

# Hence, now the possible set of models are- ARIMA(p,d,q):
# {ARIMA (1,0,1), ARIMA (1,0,2), ARIMA (1,0,4), ARIMA (1,0,5), ARIMA (2,0,2),
ARIMA (3,0,1), ARIMA (3,0,4), ARIMA (3,0,5)}
```



```

```{r Parameter Estimation}
model_101_ML <- arima(BC.weatherAU, order = c(1,0,1), method='ML')
coeftest(model_101_ML)

model_101_CSS <- arima(BC.weatherAU, order = c(1,0,1), method='CSS')
coeftest(model_101_CSS)

model_101_CSS_ML <- arima(BC.weatherAU, order = c(1,0,1), method='CSS-ML')
coeftest(model_101_CSS_ML)
```



```

```{r ARIMA (1,0,1)}
model_101 <- arima(BC.weatherAU, order = c(1,0,1), method='ML')
coeftest(model_101)
residual.analysis(model = model_101, std = TRUE,start = 2, class = "ARIMA")
```



```

```{r ARIMA (1,0,4)}
model_104 <- arima(BC.weatherAU, order = c(1,0,4), method='ML')
coeftest(model_104)
residual.analysis(model = model_104, std = TRUE,start = 2, class = "ARIMA")
```

# Garch

```


```


```


```

```

```{r residuals of ARMA(1,1) model to identify the orders of GARCH model}
abs.r.res.BC.weatherAU = abs(rstandard(model_101))
sq.r.res.BC.weatherAU = rstandard(model_101)^2
```

```{r GARCH model selection for absolute return series}
par(mfrow=c(1,2))
acf(abs.r.res.BC.weatherAU, main="ACF of the absolute box cox series.")
pacf(abs.r.res.BC.weatherAU, main="PACF of the absolute box cox series.")
par(mfrow=c(1,1))
# no significant lags so no model

eacf(abs.r.res.BC.weatherAU, ar.max = 10, ma.max = 10)
# max(p, q) = 0, q = 1 ==> max(p, q = 1) = 0 <> No models
# max(p,q) = 1 and q = 1 ==> max(p, q = 1) = 1; hence p = 0, 1 and
GARCH(0,1),GARCH(1,1).
# max(p,q) = 0 and q = 2 ==> max(p, q = 2) = 0; Does not lead to any models.
# max(p,q) = 1 and q = 2 ==> max(p, q = 2) = 1; Does not lead to any models.

# from the BIC plot
res = armasubsets(y=abs.r.res.BC.weatherAU,nar=5,nma=5,y.name='p',ar.method='ols')
plot(res)

# {GARCH(0,1),GARCH(1,1)}
```

```{r GARCH model selection for squared return series}
par(mfrow=c(1,2))
acf(sq.r.res.BC.weatherAU, main="ACF of the squared box cox series.")
pacf(sq.r.res.BC.weatherAU, main="PACF of the squared box cox series.")
par(mfrow=c(1,1))

eacf(sq.r.res.BC.weatherAU, ar.max = 10, ma.max = 10)
# max(p, q) = 0, q = 1 ==> max(p, q = 1) = 0 ==> max(p, 3) = 1 <> No models
# max(p,q) = 1 and q = 1 ==> max(p, q = 1) = 1; hence p = 0, 1 and
GARCH(0,1),GARCH(1,1).
# max(p,q) = 0 and q = 2 ==> max(p, q = 2) = 0; Does not lead to any models.
# max(p,q) = 1 and q = 2 ==> max(p, q = 2) = 1; Does not lead to any models.

# from the BIC plot
res = armasubsets(y=sq.r.res.BC.weatherAU,nar=5,nma=5,y.name='p',ar.method='ols')
plot(res)

```

```

# max(p,q) = 3 and q = 3 ==> max(p, q = 3) = 3; hence p = 0, 1, 2, 3 and
GARCH(0,3),GARCH(1,3),GARCH(2,3),GARCH(3,3)
'''

'''{r GARCH(0,1) = ARCH(1)}
m.01 = garch(BC.weatherAU, order=c(0,1), trace=FALSE)
summary(m.01)
residual.analysis(model = m.01, std = TRUE,start = 2, class = "GARCH")
'''

'''{r GARCH(1,1)}
m.11 = garch(BC.weatherAU, order=c(1,1), trace=FALSE)
summary(m.11)
residual.analysis(model = m.11, std = TRUE,start = 2, class = "GARCH")
'''

'''{r}
model_101_01 <- fGarch::garchFit(~ arma(1,1)+garch(1,0), data = BC.weatherAU, trace=F)
summary(model_101_01)
residual.analysis(model = model_101_01, std = TRUE,start = 2, class = "fGARCH")
'''

'''{r}
model_101_11 <- fGarch::garchFit(~ arma(1,1)+garch(1,1), data = BC.weatherAU, trace=F)
# Warning: NaNs produced
summary(model_101_11)
residual.analysis(model = model_101_11, std = TRUE,start = 2, class = "fGARCH")
'''

'''{r}
df <- data.frame(AIC = c(model_101_01@fit$ics[1],model_101_11@fit$ics[1]),
BIC = c(model_101_01@fit$ics[2],model_101_11@fit$ics[2]))
rownames(df) <- c("ARMA(1,1)+GARCH(0,1)","ARMA(1,1)+GARCH(1,1)")
df
'''

'''{r}
spec <- ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(0, 1)

```

```

),
mean.model = list(armaOrder = c(1, 1)))
model_101_01_2 <- ugarchfit(spec = spec, data = BC.weatherAU,
solver = "hybrid",
solver.control = list(trace=0))
model_101_01_2
```

```
```{r "ME", "RMSE", "MAE", "MPE", "MAPE", "MASE", "ACF1" scores}

# Define the list of ARMA and GARCH models
arma_orders <- list(c(1, 1), c(1, 1))
garch_orders <- list(c(0, 1), c(1, 1))

# Call the function with your data and the list of ARMA+GARCH models
accuracy_results <- fit_armagarch_models(BC.weatherAU, arma_orders, garch_orders)
print(accuracy_results)
```

```{r}
frc <- ugarchforecast(model_101_01_2,n.ahead=10,data=BC.weatherAU)
frc
plot(frc, which = 1)
```

```{r}
# Reverse Box-Cox Transformation
# frcRaw <- ((lambda * frc@forecast$seriesFor/100) + 1)^(1/lambda)
frcRaw <- ((lambda * frc@forecast$seriesFor) + 1)^(1/lambda)

frcRaw
```

```{r}
plot(weatherAUS_ts, xlim= c(2008.7,2018.4), ylim = c(0,400),
ylab = "Total Rainfall (mm)",
main = "Forecasts from the cosine wave fitted to the total rainfall series.")
#lines(ts(as.vector(frcRaw), start=c(2017, 5), frequency = 12), col="red", type="l")
lines(ts(as.vector(frcRaw), start = c(2017.5), frequency = 12), col="blue", type="l")

```

```
legend("topleft", lty=1, pch=1, col=c("black","blue"), text.width = 10,  
      c("Data", "Forecasts"))  
``
```

## 15. References

- [1] Demirhan, H., "Lecture slides for Time Series Analysis, Semester 1, 2023," RMIT University, Melbourne, Australia.
- [2] A. M. De Livera, R. J. Hyndman, and R. D. Snyder, "Forecasting Time Series With Complex Seasonal Patterns Using Exponential Smoothing," *Journal of the American Statistical Association*, vol. 106, no. 496, pp. 1513–1527, Dec. 2011, doi: <https://doi.org/10.1198/jasa.2011.tm09771>.
- [3] S. Fan and R. J. Hyndman, "Short-Term Load Forecasting Based on a Semi-Parametric Additive Model," *IEEE Transactions on Power Systems*, vol. 27, no. 1, pp. 134–141, Feb. 2012, doi: <https://doi.org/10.1109/tpwrs.2011.2162082>.
- [4] E. Parzen, "ARARMA models for time series analysis and forecasting," *Journal of Forecasting*, vol. 1, no. 1, pp. 67–82, Jan. 1982, doi: <https://doi.org/10.1002/for.3980010108>.
- [5] "11.1 Complex seasonality | Forecasting: Principles and Practice," *Otexts.com*, 2011. <https://otexts.com/fpp2/complexseasonality.html>
- [6] J. D. Cryer and K.-S. Chan, *Time series analysis: with applications in R*. New York: Springer, 2008.
- [7] Young, J. (2020) Rain in Australia, Kaggle. doi: <https://www.kaggle.com/datasets/jsphyg/weather-dataset-rattle-package>
- [8] Hyndman, R.J. and Athanasopoulos, G., *Forecasting: Principles and practice* (2<sup>nd</sup> ed), 3.4 Evaluating forecast accuracy. doi: <https://otexts.com/fpp2/accuracy.html>