Time Series Analysis Final Report

A report submitted in fulfilment of the requirements for Assignment 3 for MATH1318: Time Series Analysis

Chris Santosh John (S3954428)	1/3
Sushil Sundar Ramesh (S3835273)	1/3
Krishnakanth Srikanth (S3959200)	1/3
Total	**1**

School of Computing Technologies

Royal Melbourne Institute of Technology

Melbourne, Australia

Monday, 16 June 2024

Contents

1 Problem Definition

The goal of this project is to analyze the yearly mean sunspot numbers dataset from 1700 to 2023, obtained from the Solar Influences Data Analysis Center of the Royal Observatory of Belgium (available here). This dataset represents the yearly mean total sunspot number, calculated as the arithmetic mean of the daily total sunspot number for each year.

The project aims to provide insights into trends and develop a reliable model for the data. Sunspots, which are temporary phenomena on the Sun's photosphere appearing as darker spots, are crucial for understanding solar cycles with implications for space weather and climate. Understanding their patterns and periodicity is essential for these analyses.

2 Aim

The primary objectives are as follows:

1. Conduct an initial exploratory data analysis to understand the characteristics, trends, and patterns present in the dataset. This analysis will involve examining summary statistics, visualizations such as time series plots, and identifying any notable features or anomalies.
2. Propose a set of possible models using various model specification tools such as Autocorrelation Function (ACF), Partial Autocorrelation Function (PACF), Extended Autocorrelation Function (EACF), and Bayesian Information Criterion (BIC) table.
3. Fit all the proposed models to the dataset to obtain parameter estimates. This step involves interpreting the estimated coefficients and assessing their significance.
4. Using the goodness-of-fit metrics such as Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), Mean Squared Error (MSE), etc., to compare, select the best-fitted model among the set of proposed models and utilizing that best model to forecast for next 10 years.

3 Setup

This section ensures that the necessary setup steps are performed before proceeding with the analysis, including setting the working directory and importing required packages for data manipulation, visualization, and time series analysis.This preparation step is crucial for conducting a smooth and effective analysis process.

3.1 Setting Working Directory

```
# Add user working directory path.
setwd("~/Documents/RMIT/TimeSeriesAnalysis/TSA-A3")

# Verify present working directory.
getwd()
```

```
## [1] "/Users/chris/Documents/RMIT/TimeSeriesAnalysis/TSA-A3"
```

3.2 Dependencies

```
# Code to install dependencies.
# install.packages(c('tidyverse', 'TSA', 'forecast', 'lmtest', 'tseries', 'urca'))

# Import Dependencies.
```

```r
library(tidyverse)
library(TSA)
library(forecast)
library(lmtest)
library(urca)
library(tseries)
library(patchwork)

# Source Utility Script
source("./utilities.R")
```

Details on the functions used from the Utility Script (utilities.R) can be found in the Appendix section of this report.

4 Data

The data set under analysis in this assignment represents yearly mean Sunspot Numbers, calculated as the arithmetic mean of the daily total sunspot number for each year from 1700-2023. Sourced from Solar Influences Data Analysis Center of the Royal Observatory of Belgium (available here), the data set offers insights into long-term climate trends and variations.

```r
# Reading Data into R environment.
sunspot_data <- read.csv('./Resources/YearlySunspotData.csv', sep = ';', header =
↪  FALSE)

# Extract the first two columns
sunspot_data <- sunspot_data[, 1:2]

# Assign column names
colnames(sunspot_data) <- c("Year","MeanSunspotNumber")

# Display Sunspot Data
sunspot_data %>% head(15)
```

```
##      Year MeanSunspotNumber
## 1  1700.5               8.3
## 2  1701.5              18.3
## 3  1702.5              26.7
## 4  1703.5              38.3
## 5  1704.5              60.0
## 6  1705.5              96.7
## 7  1706.5              48.3
## 8  1707.5              33.3
## 9  1708.5              16.7
## 10 1709.5              13.3
## 11 1710.5               5.0
## 12 1711.5               0.0
## 13 1712.5               0.0
## 14 1713.5               3.3
## 15 1714.5              18.3
```

The data set consists of observations starting from 1700 to 2023 (constituting 324 years inclusive of the start and end terms). It is essential to check if all years are accounted for in the data set without inconsistent or null values as this might be an indication to an incomplete sequence.

```
# Calculate number of years between 1700 - 2023 (324 years)
true_years = (2023 - 1700) + 1

# Verify year count with true year count
cat("Number of years between 1700 - 2023 (inclusive): ", true_years,
    "\nNumber of years accounted for out of 324 years: ", sunspot_data %>% nrow(),
    "\nCount of null values in the data set: ",
     ↪  sum(is.na(sunspot_data$MeanSunspotNumber)))
```

```
## Number of years between 1700 - 2023 (inclusive):  324
## Number of years accounted for out of 324 years:  324
## Count of null values in the data set:  0
```

4.1 Convert to TS Object

Converting CSV (Comma-Separated Values) data to a ts() object in R for time series analysis serves the purpose of enabling efficient manipulation, exploration, and modeling of time-dependent data using specialized functions and tools designed for such analyses.

Time series data inherently includes temporal information, such as time stamps or time intervals between observations. The conversion to a `ts()` object ensures that R can effectively recognize and leverage this temporal aspect of the data for conducting time-based analyses, thus avoiding the risk of overlooking or disregarding important temporal characteristics.

Note: The frequency for the observed series is set to 11 for two primary reasons. Firstly, the ACF plot (Figure 4.1) of the raw series reveals periodic cycles occurring roughly every 11 years. Secondly, detailed domain analysis of solar cycles corroborates this pattern, as each solar cycle is approximately 11 years in duration.

```
# Convert data to time series object
sunspot_TS <- sunspot_data$MeanSunspotNumber %>% ts(frequency = 11)

# View first 20 years of the series
sunspot_TS %>% head(20)
```

```
## Time Series:
## Start = c(1, 1)
## End = c(2, 9)
## Frequency = 11
##  [1]   8.3  18.3  26.7  38.3  60.0  96.7  48.3  33.3  16.7  13.3   5.0   0.0
## [13]   0.0   3.3  18.3  45.0  78.3 105.0 100.0  65.0
```

5 EDA

The Exploratory Data Analysis (EDA) section initiates the exploration and summarization of the time series data being analyzed. This involves verifying the data type of the time series object (sunspot_TS) to ensure its suitability for further analysis. Furthermore, summary statistics are provided to offer a comprehensive overview of the distribution and features of the mean sunspot numbers data.

5.1 Exploration

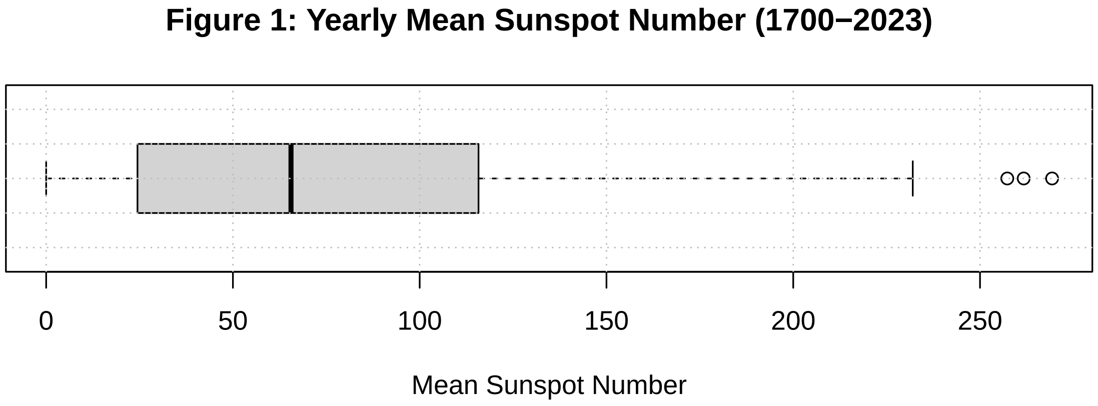
```
# Verify Data type
sunspot_TS %>% class()
```

```
## [1] "ts"
```

```
# Summarize the time series object
sunspot_TS %>% summary()
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    0.00   24.57   65.55   78.53  115.47  269.30
```

```
# Create box plot to visualize sunspot number
sunspot_TS %>% boxplot(horizontal = TRUE,
                       main = "Figure 1: Yearly Mean Sunspot Number (1700-2023)",
                       xlab = "Mean Sunspot Number",
                       border = "black")
# Add grid
grid(nx = NULL, ny = NULL, col = "gray", lty = "dotted")
```



The summary statistics offer insight into the distribution of sunspots spanning from 1700 to 2023. The minimum value of 0.00 and a median value of 65.55 indicating an almost near even split where almost half of the recorded sunspots are below this value and the other half are above. These statistics provide a glimpse into the variability and range of mean sunspots over the analyzed time frame.

Moreover, the box plot (Figure 1) demonstrates a relatively uniform distribution of number of sunspot observations without many outliers. There are only 3 outliers observed which exhibits the largest sunspot areas within the series on the years 1778, 1957 and 1958.

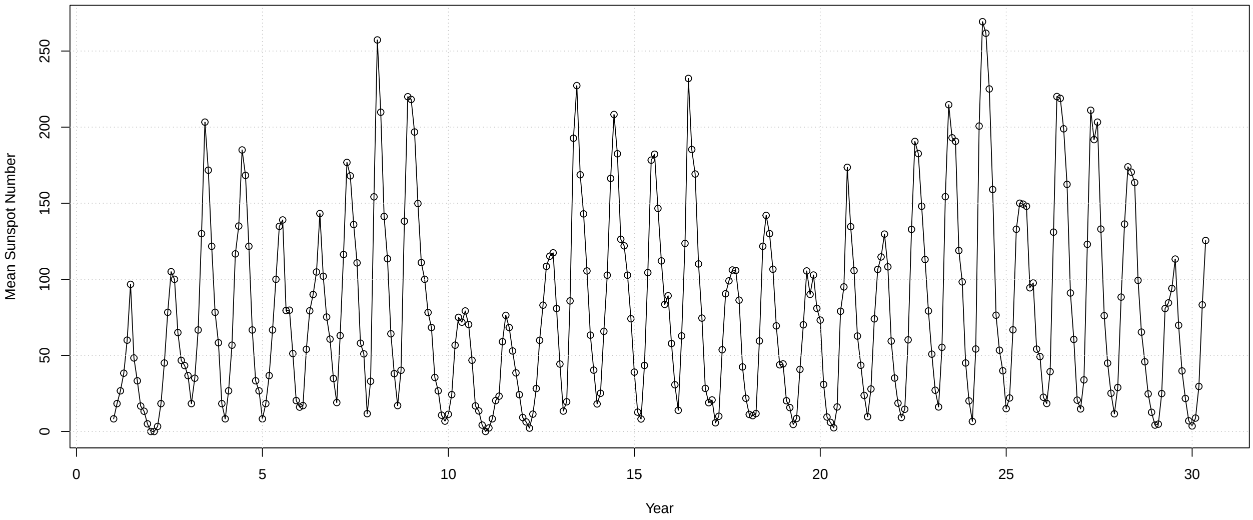
5.2 Visualizing data

In this section, a time series plot is created to depict the trajectory of sunspots over time. The goal is to extract meaningful insights regarding the data's characteristics, including trends, seasonality, variance changes, potential change points, and overall data behavior. This step is crucial for informing the model selection process and guiding decision-making effectively.

```
# Plot sunspot series across time
sunspot_TS %>% plot(type = 'o',
                    main = "Figure 2: Yearly Mean Sunspot Number (1700-2023)",
                    xlab = "Year",
                    ylab = "Mean Sunspot Number")
```

```
# Add grid
grid(nx = NULL, ny = NULL, col = "gray", lty = "dotted")
```

Figure 2: Yearly Mean Sunspot Number (1700−2023)



5.3 Plot Analysis

1. **Trend**:

   The plot indicates that there is minimal trend present within the sunspot number data. This initial visual assessment suggests that the series exhibits stationarity, a crucial characteristic of time series data. The lack of a clear trend and the apparent stability of the statistical properties in the plot support the hypothesis that the series is stationary or trendless (based on visual assessment). However, we do acknowledge that there is a possibility that the underlying seasonality of the data might be masking the trend.

2. **Seasonality**:

   The time series plot clearly exhibits seasonality in the data. Seasonality refers to patterns that repeat over specific periods, such as a year, quarter, or month. In this case, there is a distinct pattern of ups and downs recurring across the years, as evident in the graph. These regular fluctuations indicate that the data has a seasonal component, which is an important aspect to consider in time series analysis and forecasting. Recognizing seasonality helps in building more accurate models by accounting for these predictable changes.

3. **Changing Variance**:

   Visual inspection of the time series reveals no severe fluctuations in the variability of sunspot observations. Some fluctuations manifest as periodic bursts of high variability, followed by intervals with significantly lower variance. During these lower variance periods, the data points exhibit minimal dispersion. This pattern indicates that the sunspot observations experience cycles of high and low activity, which is crucial for understanding the underlying dynamics of the series.

4. **Behavior**:

   Due to the presence of seasonality, it is highly likely that the underlying pattern cannot be inferred solely from a visual assessment. The time series plot indicates strong and predominant autoregressive (AR) behavior, as most points exhibit a pattern where past observations influence future ones, suggesting a relationship between successive observations. However, there are minimal regions that display jumps and gaps, particularly near the peaks and tails of each cycle, indicating a possible moving average (MA) influence. This combination of AR behavior and occasional MA characteristics highlights the complex dynamics of the sunspot observations, where both past values and random shocks play a role in shaping the time series.

5. **Change/Intervention Point**:

   A change point can be defined as a juncture in the series where there is an abrupt and significant alteration in either the series behavior or trend. Based on this definition, it could be argued that no distinct change point is discernible in the series. The overall behavior of the series remains relatively consistent, without any abrupt shifts or notable deviations that would indicate a **significant** change in the underlying pattern or trend. This suggests that the series maintains its general characteristics throughout the observed period.

5.4 Assessing Auto-Correlation

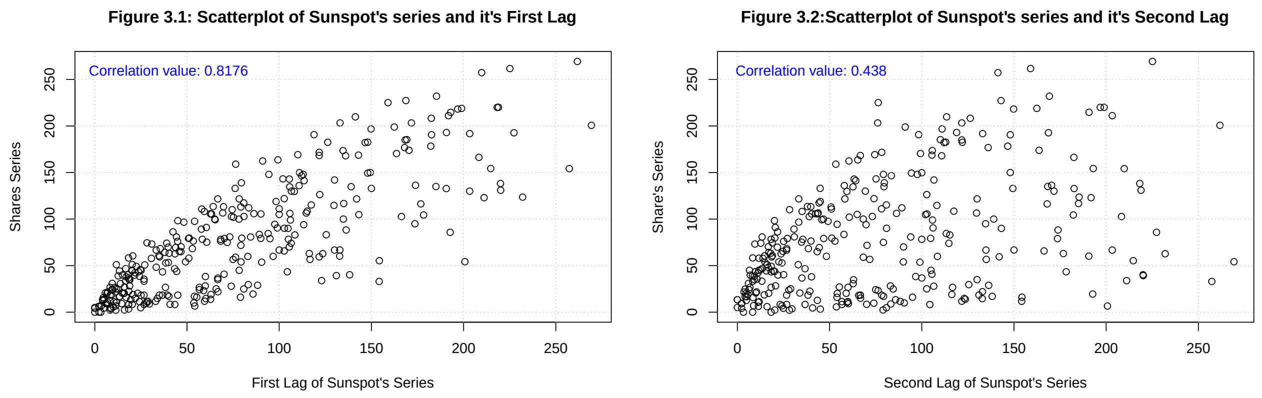
As observed in the previous section Visualizing Data, the series exhibits indications of autocorrelation. To further explore and gain deeper insights into the data, we will conduct an autocorrelation analysis by examining the correlation of the series with its first and second lag.

```
# Creating first and second lags for data
series <- sunspot_TS
first_lag <- zlag(sunspot_TS)
second_lag <- zlag(zlag(sunspot_TS))

# Setting index for correlation test
index1 <- 2:length(first_lag)
index2 <- 3:length(second_lag)

# Pearson's Correlation Test for first and second lag's
cor_first_lag <- cor(series[index1], first_lag[index1])
cor_second_lag <- cor(series[index2], second_lag[index2])

# Results of Pearson's correlation coefficient test
cat("Pearson's Correlation between series and first lag: ", cor_first_lag %>%
↪  round(4),
    "\nPearson's Correlation between series and second lag: ", cor_second_lag %>%
     ↪  round(4))
```

```
## Pearson's Correlation between series and first lag:  0.8176
## Pearson's Correlation between series and second lag:  0.438
```

```
# Set up a 1x2 layout for plots
par(mfrow=c(1,2))

# Visualization of correlation between series and first lag
series[index1] %>% plot(first_lag[index1],
                        xlab = "First Lag of Sunspot's Series",
                        ylab = "Shares Series",
                        main = "Figure 3.1: Scatterplot of Sunspot's series and it's
                        ↪  First Lag") %>%
                   text(x=40, y=260, col='blue',
                        labels=paste0("Correlation value: ", cor_first_lag %>%
                        ↪  round(4)))
```

```r
# Add grid
grid(nx = NULL, ny = NULL, col = "gray", lty = "dotted")

# Visualization of correlation between series and second lag
series[index2] %>% plot(second_lag[index2],
                          xlab = "Second Lag of Sunspot's Series",
                          ylab = "Share's Series",
                          main = "Figure 3.2:Scatterplot of Sunspot's series and it's
                          ↪  Second Lag") %>%
                    text(x=40, y=260, col='blue',
                          labels=paste0("Correlation value: ", cor_second_lag %>%
                          ↪  round(4)))

# Add grid
grid(nx = NULL, ny = NULL, col = "gray", lty = "dotted")
```



The values obtained from the Pearson's Correlation Test indicate strong positive correlations between the series and its first lag, and a moderately positive correlation with its second lag. Additionally, the scatter plots (Figure 3.1 & 3.2) visualize this correlation between the series and its first and second lags. Both plots depict a clear positive linear relationship, with the first lag showing a more dominant linear relationship. This suggests that the current values of the series are heavily influenced by their immediate past values, reinforcing the presence of auto-regressive behavior in the data.

5.5 Assessing Stationarity

From the visual assessment of the sunspot series in the time series plot (Figure 2), there was no indication of a clear trend being present in the series. However, to confirm this observation, this section will include further analysis using various tests and tools. Specifically, we will analyze the Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots, and conduct statistical tests for stationarity. These methods will provide a more rigorous evaluation of the series' properties and help determine if the initial visual assessment holds true.

5.5.1 ACF and PACF plot

```r
# Use utility script to plot ACF and PACF plot
sunspot_TS %>% plot_acf_pacf(acf_main = "Figure 4.1: Autocorrelation Function (ACF)
↪  of Yearly Sunspot Numbers",
                              pacf_main = "Figure 4.2: Partial Autocorrelation
                              ↪  Function (PACF) of Yearly Sunspot Numbers",
                              max_lag = 70)
```

Figure 4.1: Autocorrelation Function (ACF) of Yearly Sunspot Numbers

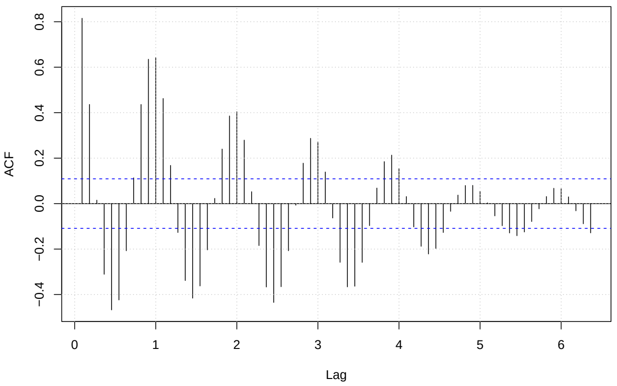
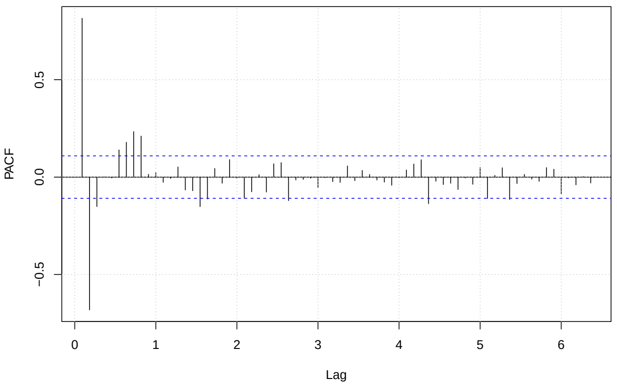


Figure 4.2: Partial Autocorrelation Function (PACF) of Yearly Sunspot Numbers



From the ACF plot (Figure 4.1), we do not observe a strict slowly decaying pattern, which is usually expected in a non-stationary series with a distinct trend. However, the ACF plot does show a decaying wave pattern, which strongly indicates seasonality or seasonal-trend in the series, aligning with our conclusion from the time series plot (Figure 2) that the series exhibits seasonality. The PACF plot (Figure 4.2) shows a very high first lag value, which could possibly indicate non-stationarity, but this high value could also be due to the underlying seasonality. Due to the lack of conclusive evidence, we need to employ statistical tests to confirm the presence or absence of stationarity in the series, providing a more definitive assessment.

Note: The above plots are generated using a utility script written specifically for the analysis to avoid any repetetive code chunks. Please refer to the Appendix section for further details.

5.5.2 Statistical Tests for Sationarity

```
# Augmented Dickey-Fuller Test for stationarity
sunspot_TS %>% adf.test()
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  .
## Dickey-Fuller = -5.3319, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

```
# Phillips-Perron Unit Root Test for stationarity
sunspot_TS %>% pp.test()
```

```
##
##  Phillips-Perron Unit Root Test
##
## data:  .
## Dickey-Fuller Z(alpha) = -83.347, Truncation lag parameter = 5, p-value
## = 0.01
## alternative hypothesis: stationary
```

Both the Augmented Dickey-Fuller Test and Phillips-Perron Test yield p-values of lower than 0.01, well below the conventional threshold of 0.05, confirming that the yearly sunspot series is stationary. Given that the time series plot (Figure 2), the ACF plot (Figure 4.1), and both the Augmented

Dickey-Fuller Test and Phillips-Perron Test show evidence of stationarity in the series, this collective evidence strongly indicates stationarity in our series.

5.6 Assessing Normality

This section will investigate the presence of normality in the series, as normality is an underlying assumption of modeling methods like Maximum Likelihood Estimation (MLE), which will be employed in the modeling sections of this analysis. Assessing normality is crucial to ensure the validity of the modeling approach and to make accurate inferences from the data. To visually and statistically assess normality, we can use tools such as histograms, Q-Q plots, and normality tests like the Shapiro-Wilk test.

```
# Set up a 1x2 layout for plots
par(mfrow=c(1,2))

# Plotting a Histogram for sunspot_TS
sunspot_TS %>% hist(main = "Figure 5.1: Histogram of Sunspot Series",
                    xlab = "Sunspot Values")

# Add grid
grid(nx = NULL, ny = NULL, col = "gray", lty = "dotted")

# Plotting a QQ plot for sunspot_TS
sunspot_TS %>% qqnorm(main = "Figure 5.2: Normal QQ Plot of Sunspot Series")
sunspot_TS %>% qqline(col='blue', lty=2)

# Add grid
grid(nx = NULL, ny = NULL, col = "gray", lty = "dotted")
```

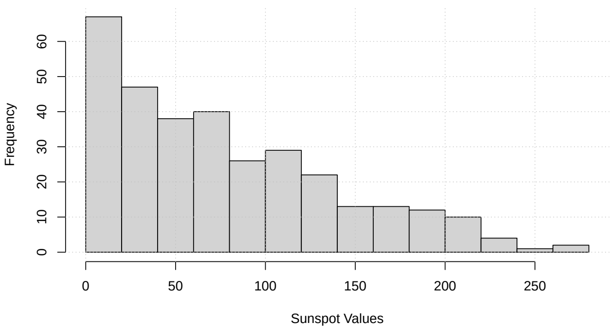


Figure 5.1: Histogram of Sunspot Series

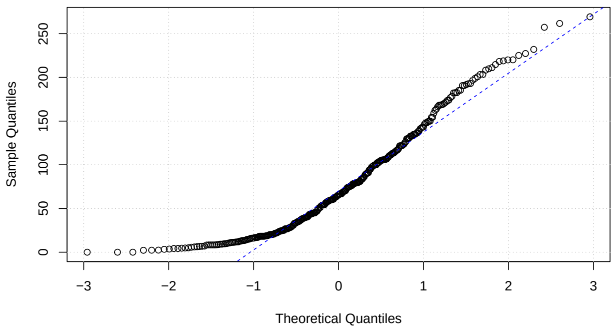


Figure 5.2: Normal QQ Plot of Sunspot Series

```
# Perform Shapiro-Wilks test for normality.
rawTS_shapiro <- sunspot_TS %>% shapiro.test()
rawTS_shapiro
```

```
##
##  Shapiro-Wilk normality test
##
## data:  .
## W = 0.92106, p-value = 4.853e-12
```

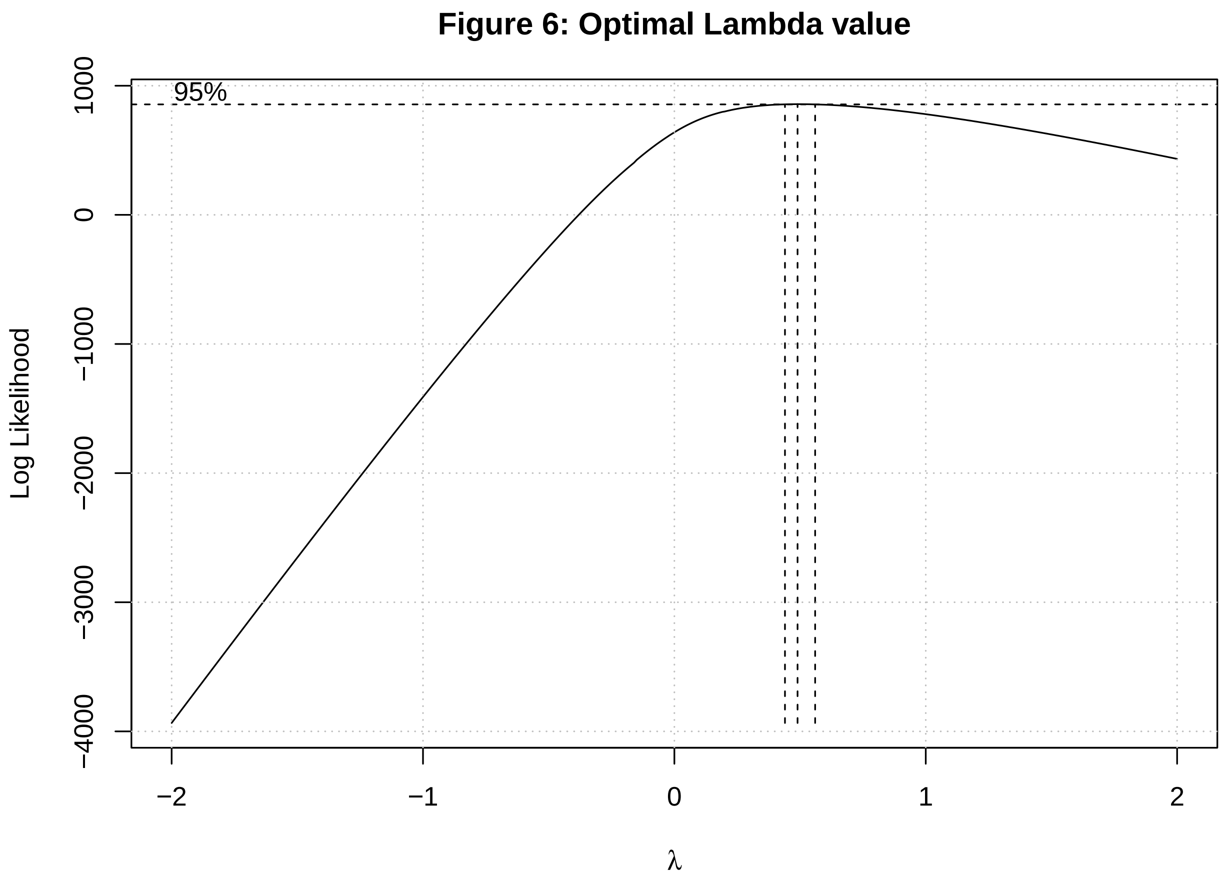
The above histogram (Figure 5.1) does not show a clear normal distribution of the sunspot numbers. Similarly, the QQ-Plot (Figure 5.2) does not indicate normality due to the significant deviation of the observations from the reference line. The Shapiro-Wilk test further supports this, with a p-value of 4.853×10-12, which is significantly lower than the conventional threshold of 0.05. This strongly suggests that the series does not exhibit normality. Given this lack of normality, appropriate transformations of the data may need to be considered to improve the normality of the series.

6 Data Transformation

In order to improve the normality of the series, we will be using the Box-Cox transformation to find an optimal transformation for the series. This process will include determining an optimal lambda value and transforming the data based on this value.

6.1 Box-Cox Transformation

```
# Adding a small positive value to ensure non-zero data
positive_sunspot_TS <- sunspot_TS + (abs(min(sunspot_TS)) + 0.01)

# Applying Box-Cox transformation
BC <- positive_sunspot_TS %>% BoxCox.ar(lambda = seq(-2, 2, 0.01), method =
↪  'yule-walker')

# Add grid
grid(nx = NULL, ny = NULL, col = "gray", lty = "dotted")

# Add main title separately
title(main = 'Figure 6: Optimal Lambda value', adj = 0.5, line = 1.4)
```



```
# Use Log-likelihood estimation to find optimal lambda values
lambda <- BC$lambda[which(max(BC$loglike) == BC$loglike)]

# Print results
cat("Lower bound Confidence Interval: ", BC$ci[1],
    "\nUpper bound Confidence Interval: ", BC$ci[2],
    "\n\nOptimal Lambda value: ", lambda)
```

```
## Lower bound Confidence Interval:  0.44
## Upper bound Confidence Interval:  0.56
##
## Optimal Lambda value:  0.49
```

The optimal lambda value for the Box-Cox transformation is determined using log-likelihood estimation. The Box-Cox transformation function generates a range of lambda values from -2 to 2 with an increment of 0.01. For each lambda value, the log-likelihood of the transformed data is computed. The lambda value corresponding to the highest log-likelihood is considered optimal as it indicates the transformation that best fits the data.

Regarding the confidence intervals, they provide insights into the reliability of the transformation. The confidence intervals, typically derived from statistical tests, indicate a range of lambda values within which the transformation is deemed statistically valid or effective.

These values suggest that lambda values between approximately 0.44 and 0.56 are likely to produce meaningful transformations of the data. Therefore, for the subsequent analysis phases, the optimal lambda value of approximately 0.49 should be considered for applying the Box-Cox transformation.

```
# Set up a 1x2 layout for plots
par(mfrow=c(3,2))

# Plot original sunspot series across time
sunspot_TS %>% plot(type = 'o',
                    main = "Figure 7.1: Original Sunspot series (1850-2023)",
                    xlab = "Year",
                    ylab = "Temperature Anomalies (°C)")

# Add grid
grid(nx = NULL, ny = NULL, col = "gray", lty = "dotted")

# Plot Box-Cox Transformed sunspot series across time
BC_sunspot_TS <- ((positive_sunspot_TS^lambda) - 1) / lambda

BC_sunspot_TS %>% plot(type = 'o',
                       main = 'Figure 7.2: Box-Cox Transformed Sunspot series
                       ↪  (1850-2023)',
                       xlab = 'Year',
                       ylab = 'Transformed Temperature anomalies (°C)')

# Add grid
grid(nx = NULL, ny = NULL, col = "gray", lty = "dotted")

# Test for normality with Box-Cox transformed series.
BC_TS_Stest <- BC_sunspot_TS %>% shapiro.test()
```

```
# Plotting a Histogram for sunspot_TS
sunspot_TS %>% hist(main = "Figure 7.3: Histogram of Original Sunspot Series",
                    xlab = "Sunspot Values")

# Plotting a Histogram for sunspot_TS
BC_sunspot_TS %>% hist(main = "Figure 7.4: Histogram of Transformed Sunspot Series",
                       xlab = "Sunspot Values")

# Plotting a QQ plot for sunspot_TS
sunspot_TS %>% qqnorm(main = "Figure 7.5: Normal QQ Plot of Original Sunspot
↪  series")
sunspot_TS %>% qqline(col='blue', lty=2) %>%
text(x = 1.4, y = 3,
     labels = paste0("Shapiro-Wilks (p-value): ", rawTS_shapiro[2]),
     col='blue', cex=1.5)
grid(nx = NULL, ny = NULL, col = "gray", lty = "dotted")

# QQ-Plot Box-Cox transformed series
BC_sunspot_TS %>% qqnorm(main = "Figure 7.6: Normal QQ Plot of Transformed Sunspot
↪  series")
BC_sunspot_TS %>% qqline(lty=2, col='blue') %>%
text(x = 1.2, y = -1.7,
     labels = paste0("Shapiro-Wilks (p-value): ", BC_TS_Stest[2]),
     col='blue', cex=1.5)
grid(nx = NULL, ny = NULL, col = "gray", lty = "dotted")
```

Figure 7.1: Original Sunspot series (1850−2023)

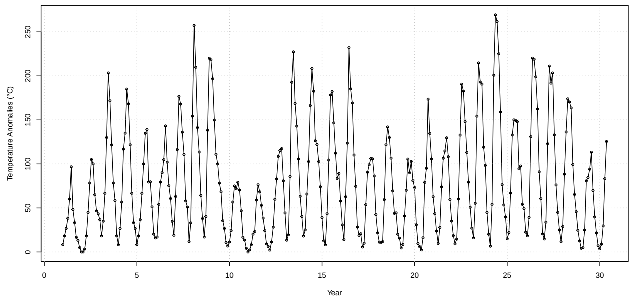


Figure 7.2: Box−Cox Transformed Sunspot series (1850−2023)

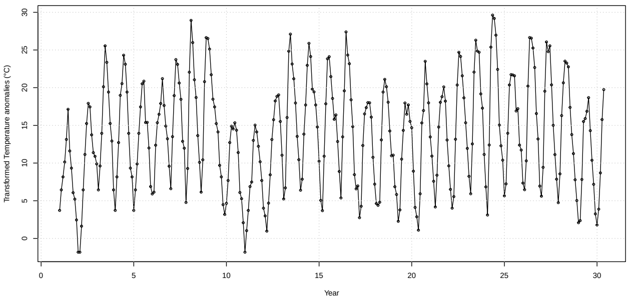


Figure 7.3: Histogram of Original Sunspot Series

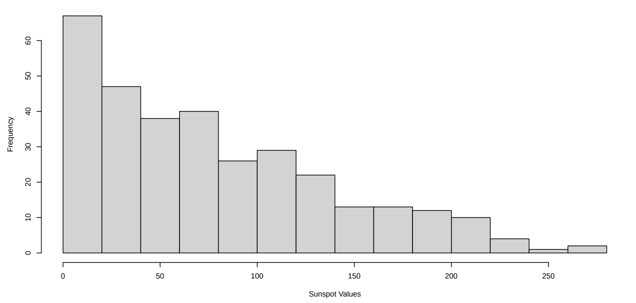


Figure 7.4: Histogram of Transformed Sunspot Series

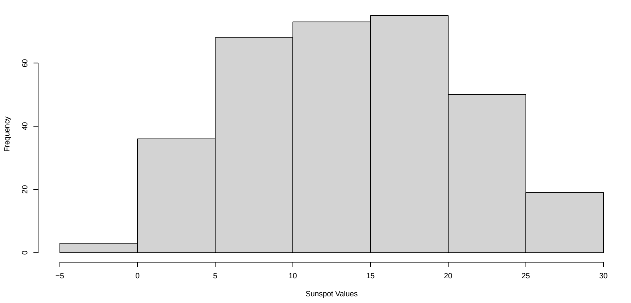


Figure 7.5: Normal QQ Plot of Original Sunspot series

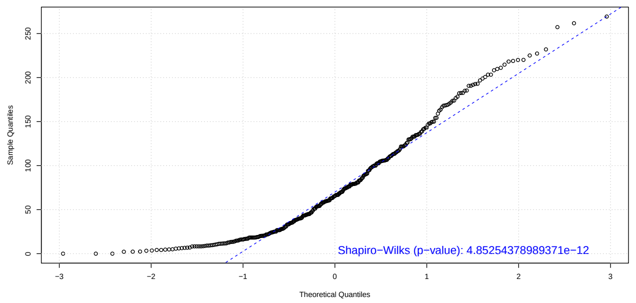
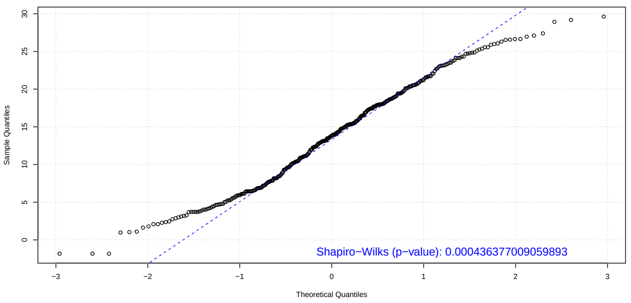


Figure 7.6: Normal QQ Plot of Transformed Sunspot series



Although the Box-Cox transformation does not fully achieve normality for the data, it does result in a significant improvement in the normality of the series. The improved normality of the series can be observed through the QQ-Plot of the original and the transformed series (Figure 7.3 & 7.4) and also from the histogram of the respective series. Initially, the Shapiro-Wilk test produced a p-value of 4.853e-12, which is extremely low and indicates a strong deviation from normality.

After applying the Box-Cox transformation, the Shapiro-Wilk p-value increased to 0.0004. This substantial increase in the p-value reflects a notable enhancement in the normality of the series, making it more suitable for modeling methods that assume normality, such as Maximum Likelihood Estimation (MLE). Given this information we will proceed using the `BC_sunspot_TS` for the successive part of the analysis.

Note: We do not conduct a stationarity test on the transformed series because we expect the stationarity to remain unaffected by the transformation. The transformation is performed primarily for two reasons: first, to address any variability in our series, and second, to improve the normality of the series. Since the series is already stationary and no ordinary differencing is performed, the stationarity will remain unchanged.

7 Model Specification

Through the initial exploratory data analysis, we identified a strong seasonal component in the series. The ACF and PACF plots of the non-transformed data further supported this observation. Given the presence of seasonality, we have several options for modeling the sunspot number series. For this analysis, we will proceed with a Seasonal AutoRegressive Integrated Moving Average (SARIMA) modeling approach. This approach allows us to account for both the seasonal and non-seasonal patterns in the data, providing a comprehensive framework for accurate modeling and forecasting of the sunspot numbers.

The SARIMA model can be expressed as follows: $\mathbf{SARIMA}(p, d, q)(P, D, Q)_s$

where:

- p is the order of the non-seasonal autoregressive (AR) component,
- d is the order of non-seasonal differencing,
- q is the order of the non-seasonal moving average (MA) component,
- P is the order of the seasonal autoregressive (SAR) component,
- D is the order of seasonal differencing,
- Q is the order of the seasonal moving average (SMA) component, and
- s is the length of the seasonal cycle i.e. periodicity.

7.1 Exclusion of Trend Models: Rationale

In the analysis of the sunspot number series, we have opted not to use trend models for several reasons. Instead, we chose the Seasonal AutoRegressive Integrated Moving Average (SARIMA) model, which is better suited for capturing the underlying patterns in the data.

Through initial exploratory data analysis and visual inspection of the time series plot, it became evident that the sunspot numbers exhibit a strong seasonal pattern. The Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots of the non-transformed data further confirmed the presence of seasonality, displaying a decaying wave pattern typical of seasonal data. In contrast, trend models primarily focus on capturing long-term trends and are not designed to account for seasonality effectively.

The sunspot series, known for its periodic fluctuations corresponding to solar cycles, exhibits cycles **approximately** every 11 years, though this period can vary between 9 and 14 years. This variability in the natural phenomenon necessitates a model that can explicitly account for these changing cycles. Traditional trend models lack the capability to capture such cyclical patterns, resulting in inadequate representations and potentially misleading forecasts.

The SARIMA model is specifically designed to handle time series data with both seasonal and non-seasonal components. It incorporates terms to address autoregressive behavior, moving averages, and differencing for both seasonal and non-seasonal parts of the series. This makes it a comprehensive and flexible approach for modeling the complex dynamics observed in the sunspot data. By using SARIMA, we can accurately capture the periodicity, autoregressive influences, and any random shocks present in the data.

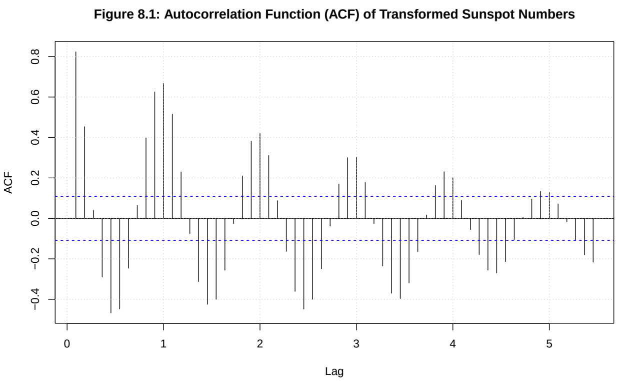
7.2 Addressing Seasonality

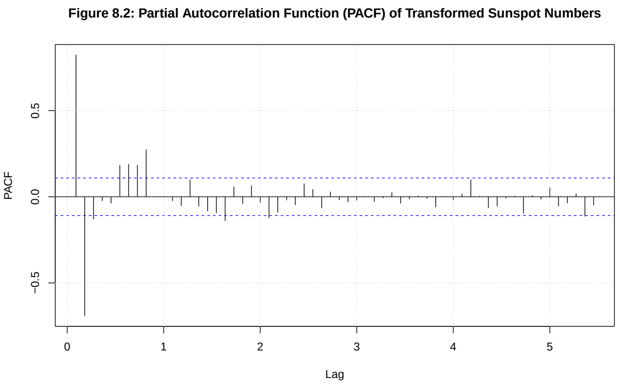
In this section, we aim to determine the optimal seasonal orders for the SARIMA model, which include the seasonal autoregressive (AR) order (P), seasonal differencing (D), seasonal moving average (MA) order (Q), and the periodicity (s). Accurately identifying these parameters is crucial for capturing the seasonal patterns in the time series data.

To achieve this, we will employ a **residuals-based approach**. This method involves fitting a series of SARIMA models with different combinations of seasonal orders to the time series data. Each model will be evaluated based on how well it captures the underlying seasonal patterns. After fitting each model, we will analyze the residuals for any signs of remaining seasonality. Ideally, the residuals will not show any seasonal pattern, indicating that the model has successfully captured the **seasonal** structure of the data.

We will use tools such as Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots to examine the residuals. Through this iterative process, we aim to select the most appropriate seasonal orders that provide the best fit for the data. This ensures that the final SARIMA model is robust and capable of accurately capturing the seasonal dynamics present in the time series.

```
# Plot ACF and PACF plots BC_sunspot_TS
BC_sunspot_TS %>%
  plot_acf_pacf(acf_main = "Figure 8.1: Autocorrelation Function (ACF) of
  ↪  Transformed Sunspot Numbers",
                pacf_main = "Figure 8.2: Partial Autocorrelation Function (PACF) of
                ↪  Transformed Sunspot Numbers",
                max_lag = 60)
```





The above ACF (Figure 8.1) of the transformed sunspot series further confirms the data's periodicity, with the most significant lags predominantly occurring around the 11-year mark.

7.2.1 Seasonal Differencing (D)

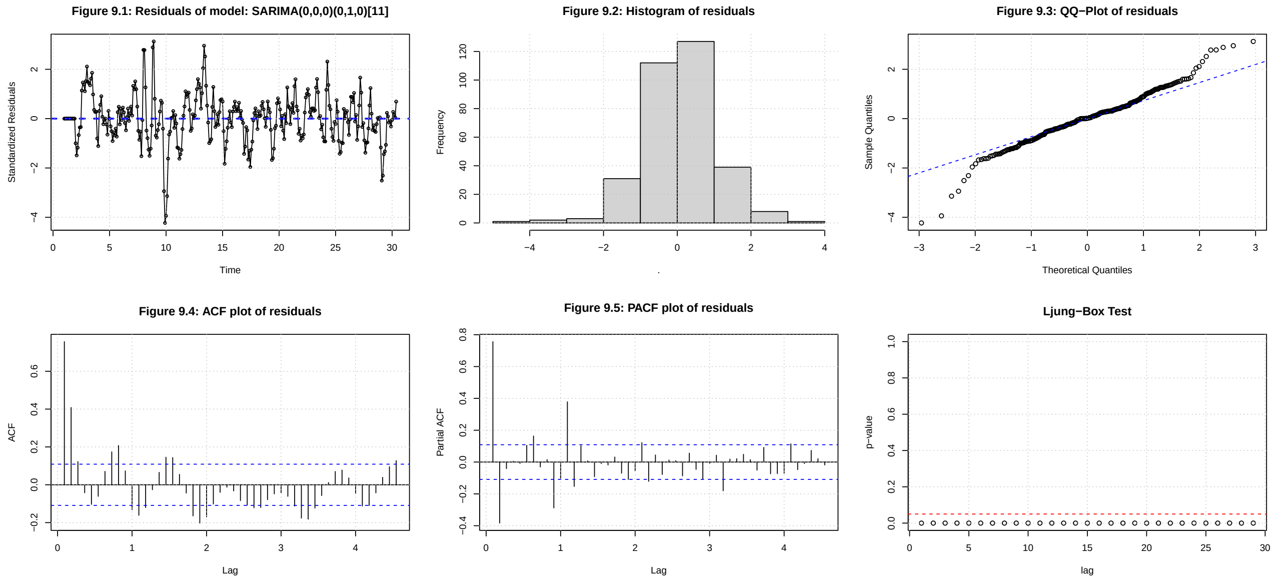
In this section, we will perform seasonal differencing on the time series data using a SARIMA(0,0,0)(0,1,0)[11] model. This model has all parameters set to zero except for the seasonal differencing order (D), which is set to 1. The purpose of this configuration is to remove any seasonal pattern with a periodicity of 11 units from the data.

By setting the seasonal differencing component (D) to 1, we expect the residuals of the model to show no remaining seasonal patterns, indicating that the seasonal effects have been effectively removed. This process helps in isolating the underlying trends and irregular components of the time series.

It is crucial to note that setting higher orders for the seasonal differencing parameter (D) can lead to data loss and may result in over-differencing. Over-differencing can distort the data and obscure meaningful patterns, making the time series analysis less reliable. Therefore, careful consideration is required when selecting the order of differencing to ensure that the seasonal effects are adequately removed without compromising the integrity of the data.

```
# Checking residuals of seasonally differenced model - SARIMA(0,0,0)(0,1,0)[11]
BC_sunspot_TS %>%
  check_residuals(order = c(0,0,0),
                  seasonal_order = c(0,1,0),
                  period = 11,
                  acf_lag = 50,
                  pacf_lag = 50,
                  res_main = 'Figure 9.1: Residuals of model:
                  ↪  SARIMA(0,0,0)(0,1,0)[11]',
                  hist_main = 'Figure 9.2: Histogram of residuals',
                  qq_main = 'Figure 9.3: QQ-Plot of residuals',
                  acf_main = 'Figure 9.4: ACF plot of residuals',
                  pacf_main = 'Figure 9.5: PACF plot of residuals')
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals
## W = 0.96977, p-value = 2.693e-06
```



We observe that seasonality in the series is significantly reduced, as evidenced by the residuals, which exhibit minimal signs of seasonality. However, it's important to note that not all stochastic components have been fully addressed as evidenced by the time series plot (Figure 9.1) and Ljung-Box Plot. This is also indicated by the Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots, which show significant autocorrelation at the seasonal lags. To address this remaining seasonality, we need to determine the appropriate seasonal Autoregressive (AR) and Moving Average (MA) orders for the series.

From the ACF plot, we observe three significant seasonal lags at the first and second seasonal periods. Given this information, the potential orders for the seasonal MA component (Q) is likely to be 2. The PACF plot suggests that the seasonal AR order (P) is likely to be 0 or 1.

7.2.2 Seasonal AR and MA orders

```
# Checking residuals of seasonally differenced model with P and Q orders -
↪  SARIMA(0,0,0)(1,1,2)[11]
BC_sunspot_TS %>% check_residuals(order = c(0,0,0),
                                  seasonal_order = c(1,1,2),
                                  period = 11,
                                  acf_lag = 50,
                                  pacf_lag = 50,
                                  res_main = 'Figure 10.1: Residuals of model:
                                  ↪  SARIMA(0,0,0)(1,1,2)[11]',
                                  hist_main = 'Figure 10.2: Histogram of residuals',
                                  qq_main = 'Figure 10.3: QQ-Plot of residuals',
                                  acf_main = 'Figure 10.4: ACF plot of residuals',
                                  pacf_main = 'Figure 10.5: PACF plot of residuals')
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals
## W = 0.98416, p-value = 0.001234
```

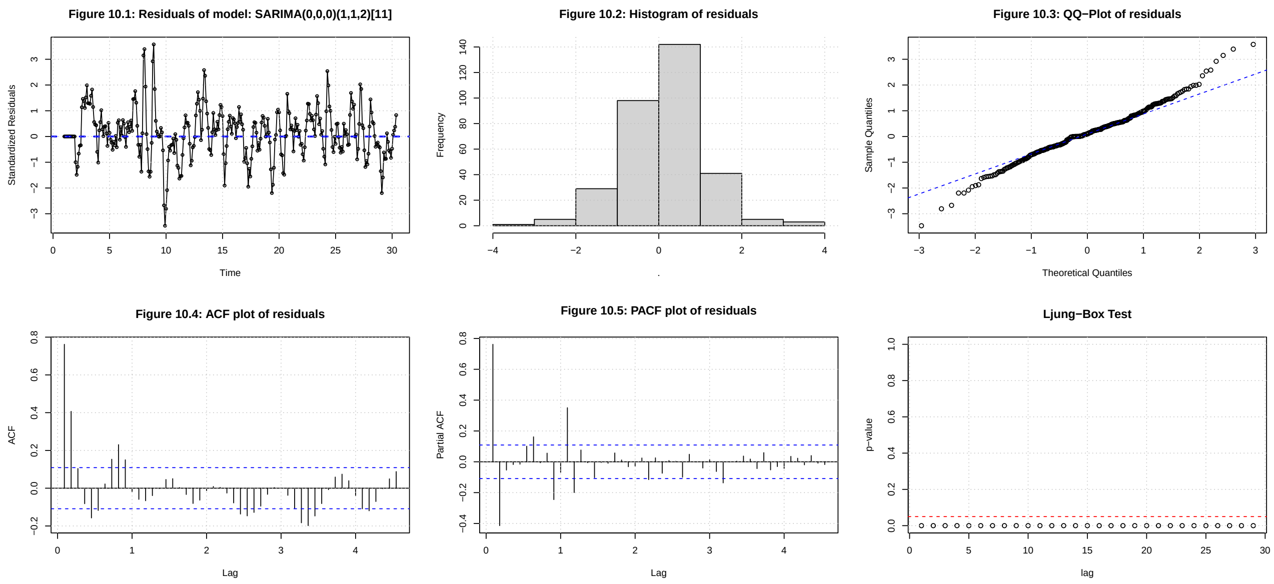


Figure 10.1: Residuals of model: SARIMA(0,0,0)(1,1,2)[11]

Figure 10.2: Histogram of residuals

Figure 10.3: QQ−Plot of residuals

Figure 10.4: ACF plot of residuals

Figure 10.5: PACF plot of residuals

Ljung−Box Test

We can observe from the ACF (Figure 10.4) and PACF plot (Figure 10.5) there are no more significant spikes at the seasonal lags. The residuals plot also shows significant reduction in the seasonality when compared to the original series. After experimenting with possible combinations of the P and Q derived from the previous section of the analysis. The SARIMA(0,0,0)(1,1,2)[11] model shows the best results in addressing the seasonal component underlying in the data.

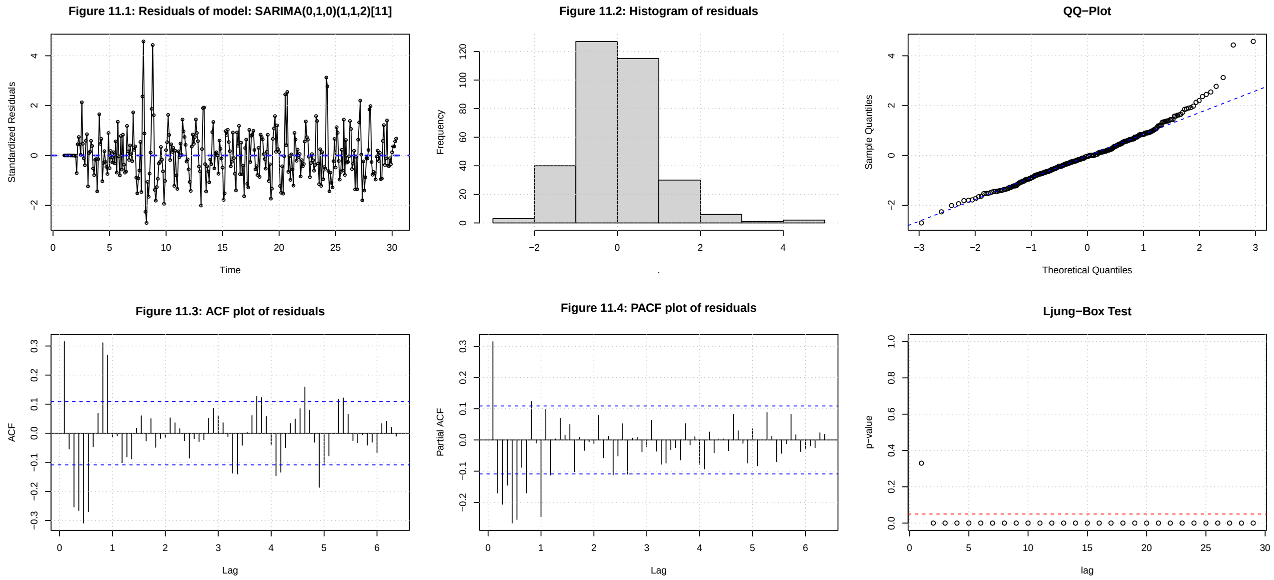
Given this information we, set the seasonal orders (P,D,Q) as (1,1,2)

7.3 Ordinary Differencing (d)

From the above section, once we remove the predominant seasonal behaviour inherent in the data, we see that the series does show signs of a trend which is assessed using the ACF (Figure 10.4) and PACF (Figure 10.5) plots. The ACF plot shows a set of gradually decaying lags while the PACF show a significantly high first lag, both indications of a possible pattern carrying over in the residuals which is addressed through ordinary differencing below.

```
# Checking residuals of seasonally differenced model with P and Q orders -
↪  SARIMA(0,1,0)(1,1,2)[11]
BC_sunspot_TS %>%
  check_residuals(order = c(0,1,0),
                  seasonal_order = c(1,1,2),
                  period = 11,
                  acf_lag = 70,
                  pacf_lag = 70,
                  res_main = 'Figure 11.1: Residuals of model:
                   ↪  SARIMA(0,1,0)(1,1,2)[11]',
                  hist_main = 'Figure 11.2: Histogram of residuals',
                  acf_main = 'Figure 11.3: ACF plot of residuals',
                  pacf_main = 'Figure 11.4: PACF plot of residuals')
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals
## W = 0.96562, p-value = 6.102e-07
```



After introducing ordinary differencing (d=1) we can see a significant improvement of the residuals. The time series plot (Figure 11.1) does not show any significant seasonal patterns and is well centered around the zero line. We can also see that the ACF no longer features a deacying pattern and the PACF no longer has a significant first lag. Additionally, we see can also observe some improvement in the Ljung box plot which was not observed earlier in any of the plots.

Two interesting insights:

1. We can see in the histogram (Figure 11.2) that the standardized residuals do not conform to the expected -3 to 3 range, which likely indicates outliers in the residuals. From the initial EDA Exploration section we can see that these residuals are resultant from the outliers seen in our series for the years 1778, 1957 and 1958.
2. We can see a significant lag that appears at the first season after applying ordinary differencing, given that we have addressed the seasonality effectively with optimal seasonal order we assume that this shows up due to the trend of the series rather an indication of leftover seasonality. We expect the successive steps to give us more information on if this is a result of an underlying stochastic component other than seasonality.

In conclusion we will now proceed to find the ordinary Auto-Regressive (p) and Moving-Average (q) order with the above set orders which are as follows -> SARIMA(p,1,q)(1,1,2)[11]

Note: The residuals from this model (SARIMA(0,1,0)(1,1,2)[11]) will hereafter be referred to as **base residuals** in this report.

7.4 Determining Potential Models

We will now conduct a thorough analysis to determine potential models to accurately forecast future sunspot numbers. To find the sets of possible models we will use tools such as the ACF plot, PACF Plot, EACF Plot and the BIC Table. These test/tools will be applied over the base residuals we determined from the SARIMA(0,1,0)(1,1,2)[11] model.

7.4.1 ACF and PACF Analysis of residuals

```
# Store the residuals of the SARIMA(0,1,0)(1,1,2)[11] model
base_residuals <- BC_sunspot_TS %>%
  Arima(order = c(0,1,0),
        seasonal = list(order=c(1,1,2),
                        period = 11)) %>% rstandard()

# Plot ACF and PACF of the base residuals
base_residuals %>%
  plot_acf_pacf(acf_main = 'Figure 12.1: ACF of SARIMA(0,1,0)(1,1,2)[11] residuals',
                pacf_main = 'Figure 12.2: PACF of SARIMA(0,1,0)(1,1,2)[11]
                ↪  residuals')
```

Figure 12.1: ACF of SARIMA(0,1,0)(1,1,2)[11] residuals

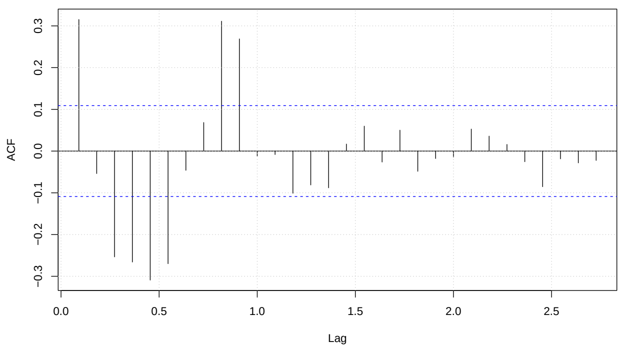
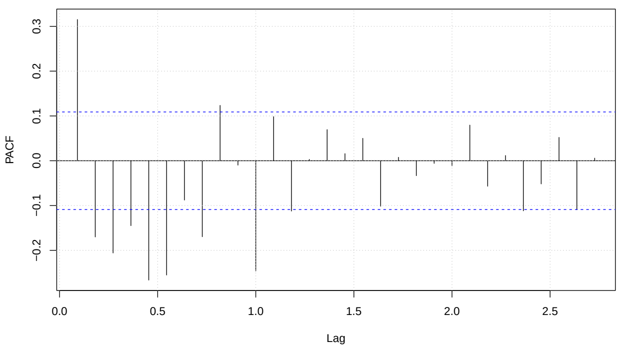


Figure 12.2: PACF of SARIMA(0,1,0)(1,1,2)[11] residuals



From the analysis of the ACF plot (Figure 12.1) we can observe 7 significant lags and 9 significant lags from the PACF plot (Figure 12.2).

Set of possible models:

- SARIMA(9,1,7)(1,1,2)[11]

7.4.2 EACF Analysis

```
# Plot EACF
base_residuals %>% eacf(ar.max = 15, ma.max = 15)
```

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
## 0 x o x x x x o o x x o  o  o  o  o  o
## 1 x o x o o x o o x x x  o  o  o  o  o
## 2 x x x o o x o o x x o  o  o  o  o  o
## 3 x x o o o o o x x x x  o  o  o  o  o
## 4 x x x x x o o o o x x  o  x  o  o  o
## 5 x x x x x o o x o x x  x  x  o  x  o
## 6 x x x x x o o o o x o  o  o  o  x  o
## 7  x x x x x o x o o x o  o  o  o  o  o
## 8  x x x x x x x o o x x  o  o  o  o  o
## 9  o x x o x x x o o x o  o  o  o  o  o
## 10 o x x x x x x o x x x  o  o  o  o  o
## 11 x x x x x o x o o o x  o  o  o  o  o
## 12 x x x x x o x o o o o  o  o  o  o  o
## 13 o x o x o x x x o o o  o  o  o  o  o
## 14 o x x x o o x o o o x  o  o  o  o  o
## 15 x x x x x o x x o o o  o  o  o  o  o
```

In the EACF table, each cell represents a combination of AR and MA orders. The "**x**" symbol indicates that the corresponding combination of orders is not suitable for modeling the series, while the "**o**" symbol suggests that the combination may be viable.

When interpreting the EACF table, attention is typically focused on patterns of "o" symbols. Successive "o" symbols near a particular cell suggest that the corresponding combination of AR and MA orders is supported by the data. Conversely, the presence of "x" symbols nearby indicates potential inadequacies in the corresponding model orders.

In the provided EACF table, the "o" symbols suggest possible combinations of AR and MA orders that are worth considering for modeling the series. In this case, the top model orders identified as promising are 0 and 10, indicating that a model with zero AR terms and ten MA terms may be suitable. Additionally, neighboring cells with "o" symbols provide further options to explore for extending the set of possible models.

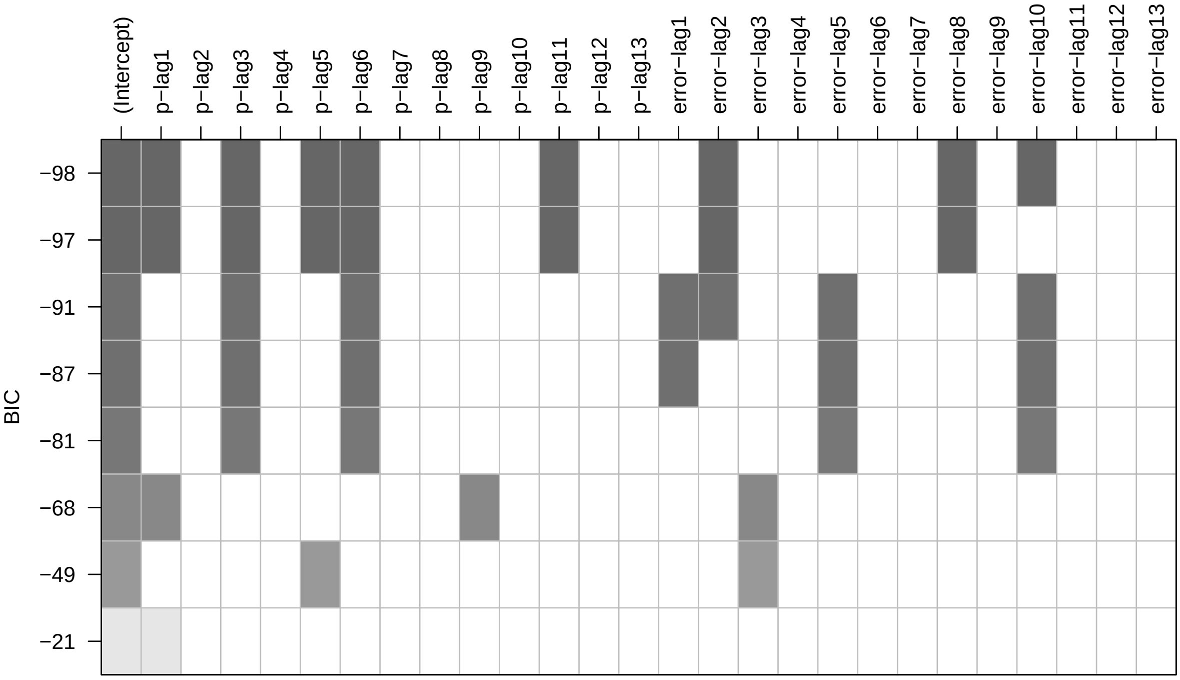
Set of possible models:

- SARIMA(0,1,10)(1,1,2)[11]
- SARIMA(0,1,11)(1,1,2)[11]
- SARIMA(1,1,11)(1,1,2)[11]

7.4.3 BIC Table

```
# Create BIC table
bic_table <- base_residuals %>% armasubsets(nar =13, nma = 13,
                                           y.name = 'p',
                                           ar.method = 'ols')

# Plot BIC Table without main title
bic_table %>% plot()

# Add main title separately
title(main = 'Figure 13: BIC Table of possible model orders', adj = 0.5, line = 5.7)
```

Figure 13: BIC Table of possible model orders



The BIC Table (Figure 13) presents the outcomes of fitting various ARMA (AutoRegressive Moving Average) models to the time series data. It displays different combinations of AutoRegressive (AR) and Moving Average (MA) orders alongside their respective BIC values.

The selection of `nar` and `nma` parameters profoundly influences the ARMA model's goodness-of-fit to the data. A balance must be struck: too few terms may lead to underfitting, missing crucial data patterns, while an excess of terms may cause overfitting, capturing noise instead of genuine patterns.

In our analysis, based on the BIC Table (Figure 13), the "p-lag" column signifies the AR(p) orders, while the "error-lag" column represents the MA(q) orders. To avoid excessively large models, we capped `nar` and `nma` at 10, considering the highest orders estimated from the ACF, PACF, and EACF, which was 9 (EACF).

Based on the BIC table, the optimal model indicates an order of p=9 and q=10. However, it is noteworthy that the AR(3) and AR(6) terms consistently appear in at least four of the top models. Similarly, the MA(2) and MA(5) terms also feature in at least four of the top models. This consistency suggests that these specific AR and MA terms may play a significant role in accurately capturing the underlying structure of the time series data.

Hence, we have four potential AR orders: (3, 5, 6, 11) and four possible MA orders: (2, 5, 8, 10).

Set of possible models (p, d, q): SARIMA(3,1,2)(1,1,2)[11], SARIMA(3,1,5)(1,1,2)[11], SARIMA(3,1,8)(1,1,2)[11], SARIMA(3,1,10)(1,1,2)[11], SARIMA(5,1,2)(1,1,2)[11], SARIMA(5,1,5)(1,1,2)[11], SARIMA(5,1,8)(1,1,2)[11], SARIMA(5,1,10)(1,1,2)[11], SARIMA(6,1,2)(1,1,2)[11], SARIMA(6,1,5)(1,1,2)[11], SARIMA(6,1,8)(1,1,2)[11], SARIMA(6,1,10)(1,1,2)[11], SARIMA(11,1,2)(1,1,2)[11], SARIMA(11,1,5)(1,1,2)[11], SARIMA(11,1,8)(1,1,2)[11], SARIMA(11,1,10)(1,1,2)[11]

7.5 Set of possible model SARIMA(p,d,q)(P,D,Q)[s]

Based on the comprehensive analysis conducted in the model specification section, we have identified a set of 20 possible models. These models, along with their corresponding ARIMA specifications, are as follows:

- SARIMA(9,1,7)(1,1,2)[11]

- SARIMA(0,1,10)(1,1,2)[11]
- SARIMA(0,1,11)(1,1,2)[11]
- SARIMA(1,1,11)(1,1,2)[11]
- SARIMA(3,1,2)(1,1,2)[11]
- SARIMA(3,1,5)(1,1,2)[11]
- SARIMA(3,1,8)(1,1,2)[11]
- SARIMA(3,1,10)(1,1,2)[11]
- SARIMA(5,1,2)(1,1,2)[11]
- SARIMA(5,1,5)(1,1,2)[11]
- SARIMA(5,1,8)(1,1,2)[11]
- SARIMA(5,1,10)(1,1,2)[11]
- SARIMA(6,1,2)(1,1,2)[11]
- SARIMA(6,1,5)(1,1,2)[11]
- SARIMA(6,1,8)(1,1,2)[11]
- SARIMA(6,1,10)(1,1,2)[11]
- SARIMA(11,1,2)(1,1,2)[11]
- SARIMA(11,1,5)(1,1,2)[11]
- SARIMA(11,1,8)(1,1,2)[11]
- SARIMA(11,1,10)(1,1,2)[11]

8 Parameter Estimation

This section evaluates 11 candidate models to assess the significance of their coefficients. We anticipate the most suitable model to exhibit consistently significant coefficients. Three parameter estimation methods are employed: Least Squares (CSS), Maximum Likelihood (ML), and a combined approach (CSS-ML).

The analysis leverages functions within the utilities.R script. Refer to the Appendix for details on these functions and their implementation.

8.1 Model 1: SARIMA(9,1,7)(1,1,2)[11]

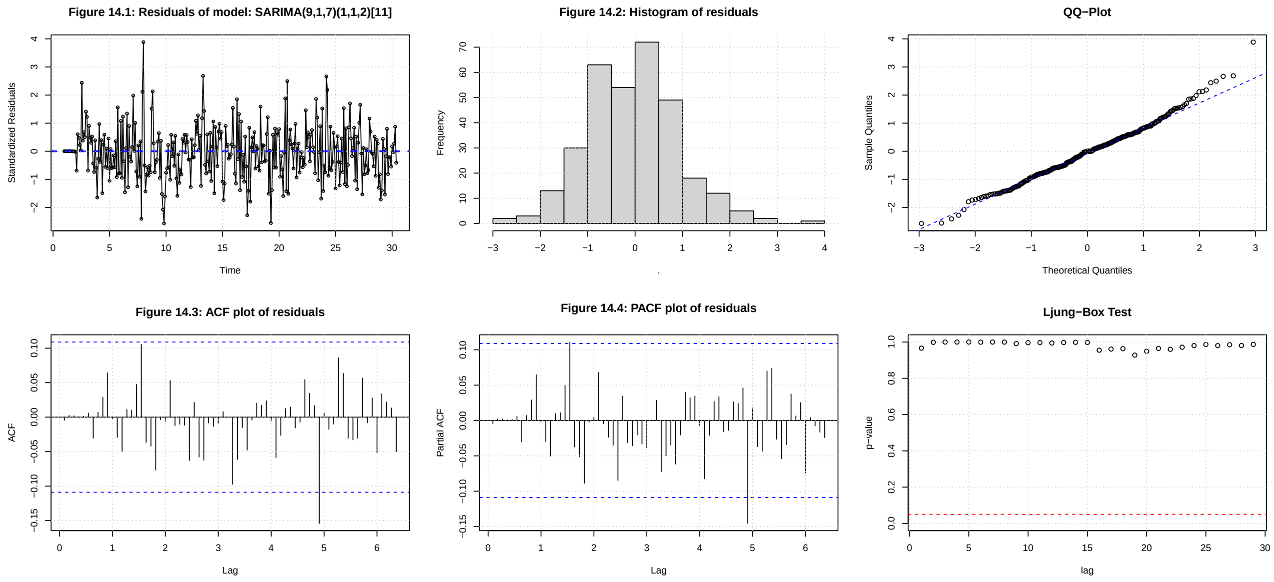
```
model_917 <- BC_sunspot_TS %>% arima_summary(pdq_order = c(9,1,7),
                                             seasonal_order = c(1,1,2),
                                             period = 11)
```

```
##
## Parameter Estimation through Least Squares (CSS) Method for ARIMA( 9 1 7 )(1,1,2)[11]
##
## z test of coefficients:
##
##         Estimate Std. Error  z value  Pr(>|z|)
## ar1    0.1887995  0.0260277   7.2538 4.053e-13 ***
## ar2   0.0579510  0.0391655   1.4796   0.13897
## ar3   0.7966241  0.0326275  24.4157 < 2.2e-16 ***
## ar4  -0.5152647  0.0378080 -13.6285 < 2.2e-16 ***
## ar5  -0.1194261        NaN      NaN       NaN
## ar6  -0.3340801  0.0540948  -6.1758 6.582e-10 ***
## ar7   0.4741542  0.0437099  10.8478 < 2.2e-16 ***
## ar8  -0.1038509  0.0494717  -2.0992   0.03580 *
## ar9   0.2689644  0.0430325   6.2503 4.098e-10 ***
## ma1   0.0099671  0.0552930   0.1803   0.85695
## ma2  -0.3823712  0.0502265  -7.6129 2.679e-14 ***
## ma3  -1.2705370        NaN      NaN       NaN
## ma4   0.2552061  0.1093632   2.3336   0.01962 *
## ma5   0.2504926        NaN      NaN       NaN
## ma6   0.4290733  0.0379407  11.3090 < 2.2e-16 ***
## ma7  -0.3294628  0.0534325  -6.1660 7.006e-10 ***
## sar1 -0.3233574        NaN      NaN       NaN
## sma1 -0.4109943        NaN      NaN       NaN
## sma2 -0.4864935        NaN      NaN       NaN
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ===============================================================
## Parameter Estimation through Maximum Likelihood (ML) Method for ARIMA( 9 1 7 )(1,1,2)[11]
##
## z test of coefficients:
##
##        Estimate Std. Error   z value  Pr(>|z|)
## ar1   0.1996071        NaN       NaN       NaN
## ar2  -0.3582741        NaN       NaN       NaN
## ar3  -0.0952555        NaN       NaN       NaN
## ar4  -0.4639609        NaN       NaN       NaN
## ar5   0.0168828        NaN       NaN       NaN
## ar6  -0.4517563  0.0025850 -174.7635 < 2.2e-16 ***
## ar7  -0.0658765        NaN       NaN       NaN
## ar8  -0.2941838  0.0042168  -69.7643 < 2.2e-16 ***
## ar9   0.0067189        NaN       NaN       NaN
## ma1   0.0481609        NaN       NaN       NaN
## ma2   0.1130323        NaN       NaN       NaN
## ma3  -0.2336162        NaN       NaN       NaN
## ma4   0.2909870        NaN       NaN       NaN
## ma5  -0.3182610        NaN       NaN       NaN
## ma6   0.0830742  0.0136964    6.0654 1.316e-09 ***
## ma7  -0.0609221        NaN       NaN       NaN
## sar1 -0.1811515        NaN       NaN       NaN
## sma1 -0.7501405        NaN       NaN       NaN
## sma2 -0.2446651        NaN       NaN       NaN
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ===============================================================
## Parameter Estimation through Combination (CSS-ML) Method for ARIMA( 9 1 7 )(1,1,2)[11]
##
```

```
## z test of coefficients:
##
##       Estimate Std. Error z value  Pr(>|z|)
## ar1   0.426868   0.167494  2.5486  0.010817 *
## ar2  -0.031574   0.139496 -0.2263  0.820935
## ar3   0.760948   0.130334  5.8384 5.270e-09 ***
## ar4  -0.619941   0.196123 -3.1610  0.001572 **
## ar5  -0.060172   0.112740 -0.5337  0.593530
## ar6  -0.311101   0.113156 -2.7493  0.005972 **
## ar7   0.515102   0.108810  4.7340 2.202e-06 ***
## ar8  -0.178578   0.094451 -1.8907  0.058666 .
## ar9   0.252594   0.081591  3.0958  0.001963 **
## ma1  -0.203868   0.166498 -1.2244  0.220784
## ma2  -0.296883   0.139366 -2.1302  0.033151 *
## ma3  -1.158492   0.142825 -8.1113 5.010e-16 ***
## ma4   0.420874   0.262921  1.6008  0.109430
## ma5   0.191685   0.155136  1.2356  0.216610
## ma6   0.465590   0.109768  4.2416 2.220e-05 ***
## ma7  -0.418901   0.157546 -2.6589  0.007839 **
## sar1 -0.091794   0.418438 -0.2194  0.826360
## sma1 -0.732317   0.395621 -1.8511  0.064161 .
## sma2 -0.267678   0.393011 -0.6811  0.495810
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ==============================================================
```

```
BC_sunspot_TS %>%
  check_residuals(order = c(9,1,7),
                  seasonal_order = c(1,1,2),
                  period = 11,
                  acf_lag = 70,
                  pacf_lag = 70,
                  res_main = 'Figure 14.1: Residuals of model:
                   ↪  SARIMA(9,1,7)(1,1,2)[11]',
                  hist_main = 'Figure 14.2: Histogram of residuals',
                  acf_main = 'Figure 14.3: ACF plot of residuals',
                  pacf_main = 'Figure 14.4: PACF plot of residuals')
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals
## W = 0.98944, p-value = 0.01912
```



The SARIMA(9,1,7)(1,1,2)[11] model was inferred from the ACF and PACF plots. The parameter estimation was conducted using three methods: CSS, ML, and CSS-ML. In the CSS and ML methods, several parameters showed NaN values, indicating issues with parameter estimation. However, the CSS-ML method provided complete parameter estimates, with most parameters being significant. Notably, higher-order AR and MA terms were significant, suggesting their necessity to capture the underlying series dynamics.

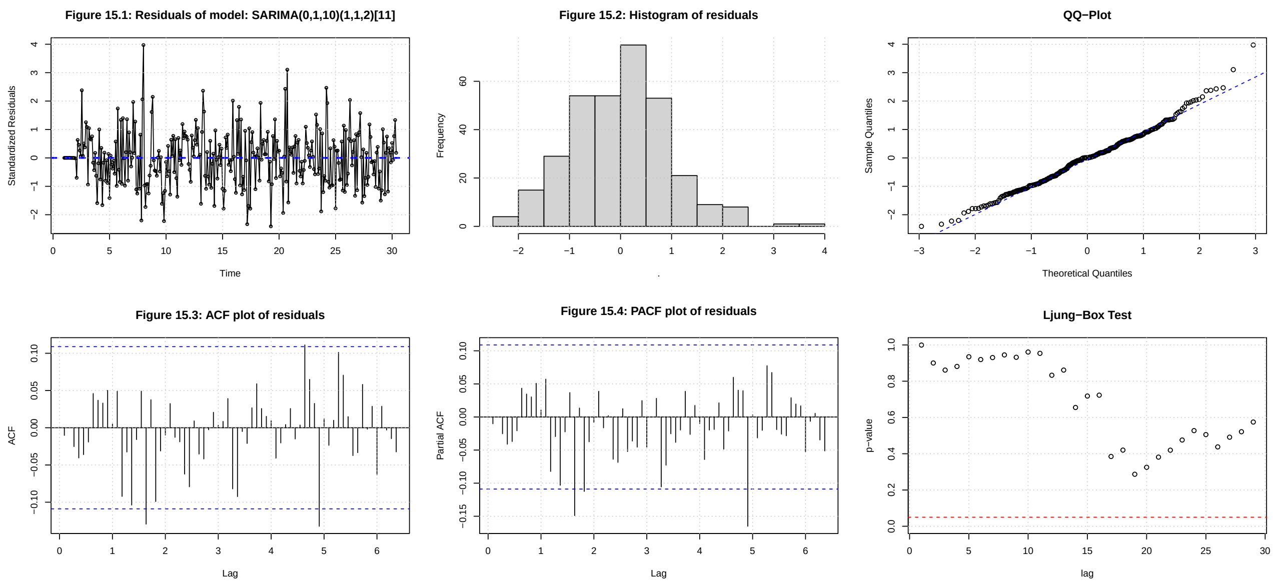
The residuals analysis showed a near-normal distribution, with a histogram indicating one outlier and a QQ-Plot confirming the near-normality of residuals. The ACF and PACF plots revealed one significant lag near the 5th season, but the Ljung-Box test results indicated that the residuals follow a white noise pattern. This suggests the model adequately captures the data's underlying patterns. Despite the presence of some `NaN` values in individual methods, the overall model, particularly through the CSS-ML method, is robust, with significant higher-order terms and residuals behaving as expected.

8.2 Model 2: SARIMA(0,1,10)(1,1,2)[11]

```
model_0110 <- BC_sunspot_TS %>% arima_summary(pdq_order = c(0,1,10),
                                               seasonal_order = c(1,1,2),
                                               period = 11)
```

```
##
## Parameter Estimation through Least Squares (CSS) Method for ARIMA( 0 1 10 )(1,1,2)[11]
##
## z test of coefficients:
##
##        Estimate Std. Error z value  Pr(>|z|)
## ma1    0.234556   0.056814  4.1285 3.651e-05 ***
## ma2   -0.205578   0.054615 -3.7641 0.0001672 ***
## ma3   -0.385557   0.061398 -6.2797 3.393e-10 ***
## ma4   -0.155015   0.068312 -2.2692 0.0232552 *
## ma5   -0.224982   0.056836 -3.9584 7.544e-05 ***
## ma6   -0.181621   0.062911 -2.8870 0.0038898 **
## ma7   -0.035669   0.082526 -0.4322 0.6655800
## ma8   -0.061573   0.055244 -1.1146 0.2650389
## ma9    0.329253   0.058402  5.6377 1.723e-08 ***
## ma10   0.331326   0.075936  4.3632 1.282e-05 ***
## sar1 -0.267905   0.161799 -1.6558 0.0977659 .
## sma1 -0.395924   0.146092 -2.7101 0.0067263 **
## sma2 -0.384342   0.122605 -3.1348 0.0017197 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ==============================================================
## Parameter Estimation through Maximum Likelihood (ML) Method for ARIMA( 0 1 10 )(1,1,2)[11]
##
## z test of coefficients:
##
##       Estimate Std. Error z value  Pr(>|z|)
## ma1   0.251683   0.056961  4.4185 9.937e-06 ***
## ma2  -0.149860   0.055636 -2.6936  0.007069 **
## ma3  -0.395845   0.055471 -7.1360 9.607e-13 ***
## ma4  -0.144278   0.060769 -2.3742  0.017587 *
## ma5  -0.228713   0.055844 -4.0956 4.212e-05 ***
## ma6  -0.235389   0.061813 -3.8081  0.000140 ***
## ma7  -0.046853   0.071367 -0.6565  0.511500
## ma8  -0.095074   0.056610 -1.6795  0.093064 .
## ma9   0.284956   0.057652  4.9427 7.706e-07 ***
## ma10  0.332581   0.077049  4.3165 1.585e-05 ***
## sar1  0.434690   0.170681  2.5468  0.010872 *
## sma1 -1.128005   0.188833 -5.9735 2.321e-09 ***
## sma2  0.128005   0.183332  0.6982  0.485042
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ==============================================================
## Parameter Estimation through Combination (CSS-ML) Method for ARIMA( 0 1 10 )(1,1,2)[11]
##
## z test of coefficients:
##
##       Estimate Std. Error z value  Pr(>|z|)
## ma1   0.251692   0.056960  4.4187 9.927e-06 ***
## ma2  -0.149861   0.055636 -2.6936  0.007068 **
## ma3  -0.395858   0.055470 -7.1364 9.582e-13 ***
## ma4  -0.144277   0.060770 -2.3742  0.017588 *
## ma5  -0.228734   0.055845 -4.0959 4.205e-05 ***
## ma6  -0.235390   0.061813 -3.8081  0.000140 ***
## ma7  -0.046873   0.071367 -0.6568  0.511323
## ma8  -0.095070   0.056611 -1.6794  0.093080 .
## ma9   0.284960   0.057653  4.9427 7.704e-07 ***
## ma10  0.332595   0.077049  4.3167 1.584e-05 ***
## sar1  0.434704   0.170663  2.5472  0.010861 *
## sma1 -1.128027   0.188821 -5.9741 2.314e-09 ***
## sma2  0.128029   0.183317  0.6984  0.484925
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ==============================================================
BC_sunspot_TS %>%
  check_residuals(order = c(0,1,10),
                  seasonal_order = c(1,1,2),
                  period = 11,
                  acf_lag = 70,
                  pacf_lag = 70,
                  res_main = 'Figure 15.1: Residuals of model:
                  ↪  SARIMA(0,1,10)(1,1,2)[11]',
                  hist_main = 'Figure 15.2: Histogram of residuals',
                  acf_main = 'Figure 15.3: ACF plot of residuals',
                  pacf_main = 'Figure 15.4: PACF plot of residuals')
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals
## W = 0.9879, p-value = 0.008379
```



In the SARIMA(0,1,10)(1,1,2)[11] model, the majority of the parameters are highly significant. Similar to the previous model, the insignificant parameters tend to be lower-order MA terms, which is consistent across all three estimation methods (CSS, ML, and CSS-ML).

The residual diagnostics plots show a near-normal distribution of the standardized residuals, with a persistent outlier. While the ACF and PACF plots indicate some slightly significant lags, the Ljung-Box test results show all p-values above the conventional 0.05 threshold, indicating that the residuals follow a white noise pattern. This suggests the model adequately captures the underlying data patterns.

8.3 Model 3: SARIMA(0,1,11)(1,1,2)[11]

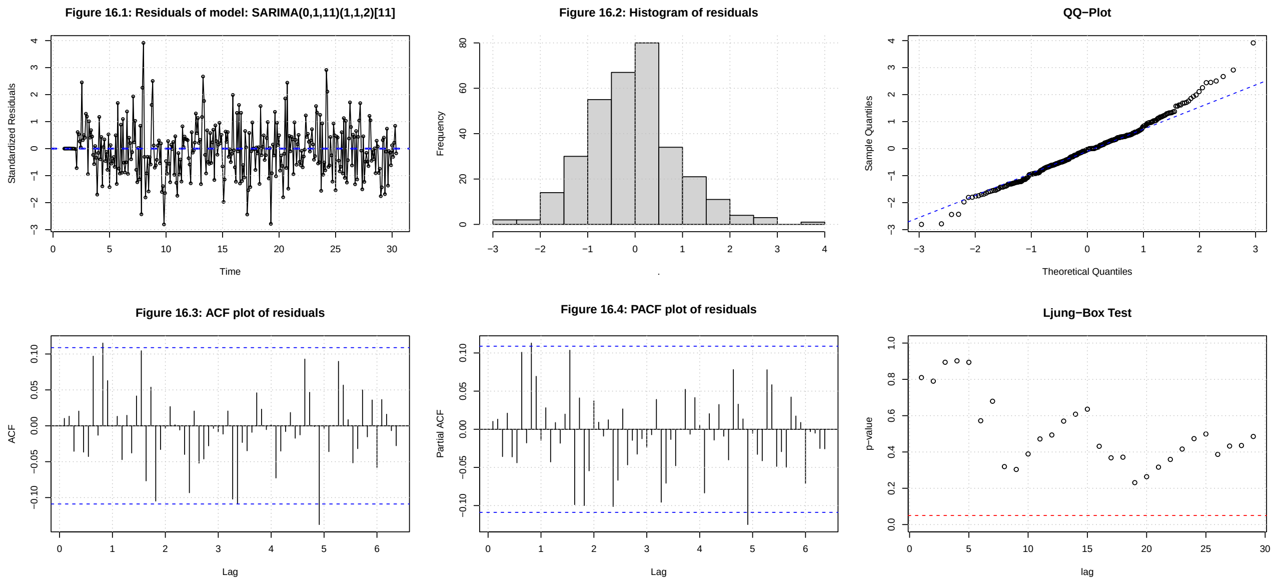
```
model_0111 <- BC_sunspot_TS %>% arima_summary(pdq_order = c(0,1,11),
                                              seasonal_order = c(1,1,2),
                                              period = 11)
```

```
##
```

```
## Parameter Estimation through Least Squares (CSS) Method for ARIMA( 0 1 11 )(1,1,2)[11]
##
## z test of coefficients:
##
##       Estimate Std. Error z value  Pr(>|z|)
## ma1   0.237735   0.056180  4.2317 2.319e-05 ***
## ma2  -0.173185   0.062804 -2.7576 0.0058232 **
## ma3  -0.384121   0.058385 -6.5791 4.734e-11 ***
## ma4  -0.141342   0.062931 -2.2460 0.0247036 *
## ma5  -0.240718   0.058351 -4.1253 3.702e-05 ***
## ma6  -0.209474   0.062803 -3.3354 0.0008518 ***
## ma7  -0.052056   0.073036 -0.7127 0.4760081
## ma8  -0.092885   0.058723 -1.5817 0.1137118
## ma9   0.334364   0.056662  5.9010 3.612e-09 ***
## ma10  0.320569   0.070293  4.5605 5.103e-06 ***
## ma11  0.104537   0.097686  1.0701 0.2845559
## sar1 -0.324618   0.161855 -2.0056 0.0448974 *
## sma1 -0.413838   0.146059 -2.8334 0.0046061 **
## sma2 -0.396896   0.127235 -3.1194 0.0018123 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ===============================================================
## Parameter Estimation through Maximum Likelihood (ML) Method for ARIMA( 0 1 11 )(1,1,2)[11]
##
## z test of coefficients:
##
##       Estimate Std. Error z value  Pr(>|z|)
## ma1   0.198368   0.062920  3.1527 0.0016178 **
## ma2  -0.229945   0.062188 -3.6976 0.0002177 ***
## ma3  -0.446120   0.059953 -7.4412 9.980e-14 ***
## ma4  -0.226200   0.064028 -3.5328 0.0004112 ***
## ma5  -0.244515   0.057680 -4.2392 2.244e-05 ***
## ma6  -0.198697   0.071838 -2.7659 0.0056762 **
## ma7  -0.024921   0.077962 -0.3197 0.7492233
## ma8  -0.028824   0.072212 -0.3992 0.6897752
## ma9   0.204859   0.051573  3.9723 7.120e-05 ***
## ma10  0.327952   0.067012  4.8939 9.885e-07 ***
## ma11 -0.331952   0.082267 -4.0351 5.459e-05 ***
## sar1  0.530847   0.106423  4.9881 6.098e-07 ***
## sma1 -0.972618   0.115123 -8.4485 < 2.2e-16 ***
## sma2 -0.027378   0.105135 -0.2604 0.7945477
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ===============================================================
## Parameter Estimation through Combination (CSS-ML) Method for ARIMA( 0 1 11 )(1,1,2)[11]
##
## z test of coefficients:
##
##       Estimate Std. Error z value  Pr(>|z|)
## ma1   0.198372   0.062922  3.1527 0.0016179 **
## ma2  -0.229922   0.062190 -3.6971 0.0002181 ***
## ma3  -0.446108   0.059952 -7.4411 9.983e-14 ***
## ma4  -0.226187   0.064029 -3.5326 0.0004116 ***
## ma5  -0.244533   0.057680 -4.2395 2.240e-05 ***
## ma6  -0.198718   0.071838 -2.7662 0.0056718 **
## ma7  -0.024929   0.077962 -0.3198 0.7491499
## ma8  -0.028833   0.072214 -0.3993 0.6896896
## ma9   0.204855   0.051572  3.9722 7.121e-05 ***
## ma10  0.327945   0.067013  4.8938 9.893e-07 ***
## ma11 -0.331932   0.082271 -4.0346 5.469e-05 ***
## sar1  0.530840   0.106422  4.9880 6.099e-07 ***
## sma1 -0.972629   0.115126 -8.4484 < 2.2e-16 ***
## sma2 -0.027367   0.105137 -0.2603 0.7946306
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ==============================================================
```

```r
BC_sunspot_TS %>%
  check_residuals(order = c(0,1,11),
                  seasonal_order = c(1,1,2),
                  period = 11,
                  acf_lag = 70,
                  pacf_lag = 70,
                  res_main = 'Figure 16.1: Residuals of model:
                   ↪  SARIMA(0,1,11)(1,1,2)[11]',
                  hist_main = 'Figure 16.2: Histogram of residuals',
                  acf_main = 'Figure 16.3: ACF plot of residuals',
                  pacf_main = 'Figure 16.4: PACF plot of residuals')
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals
## W = 0.98646, p-value = 0.003933
```



In the SARIMA(0,1,11)(1,1,2)[11] model, across all three estimation methods (CSS, ML, and CSS-ML), the ma7 and ma8 terms are consistently insignificant, whereas all other MA terms, including higher-order terms like ma9, ma10, and ma11, are highly significant. The ACF and PACF plots show minimal significant lags, which can be neglected as the Ljung-Box test confirms that the series follows a white noise pattern at all lags, with p-values above the conventional 0.05 threshold.

The residuals' normality is not perfect, likely due to the outlier observed in the data. This is further supported by the Shapiro-Wilk test, which fails to accept the normality of the standardized residuals.

8.4 Model 4: SARIMA(1,1,11)(1,1,2)[11]

```
model_1111 <- BC_sunspot_TS %>% arima_summary(pdq_order = c(1,1,11),
                                              seasonal_order = c(1,1,2),
                                              period = 11)
```

```
##
## Parameter Estimation through Least Squares (CSS) Method for ARIMA( 1 1 11 )(1,1,2)[11]
##
## z test of coefficients:
##
##         Estimate  Std. Error  z value  Pr(>|z|)
## ar1   0.66826453  0.00972981  68.6822 < 2.2e-16 ***
## ma1  -0.50366799  0.04664864 -10.7971 < 2.2e-16 ***
## ma2  -0.43711268  0.06343840  -6.8903 5.566e-12 ***
## ma3  -0.25235497  0.07864510  -3.2088  0.001333 **
## ma4   0.13901723  0.07098198   1.9585  0.050173 .
## ma5  -0.10163292  0.06249330  -1.6263  0.103886
## ma6  -0.05124085  0.08715845  -0.5879  0.556596
## ma7   0.13079360  0.07247064   1.8048  0.071109 .
## ma8   0.00033345  0.06860118   0.0049  0.996122
## ma9   0.31936339  0.07919762   4.0325 5.519e-05 ***
## ma10  0.10734302  0.08337541   1.2875  0.197932
## ma11 -0.37117467  0.07605551  -4.8803 1.059e-06 ***
## sar1 -0.07493061  0.02655735  -2.8215  0.004781 **
## sma1 -0.48298109  0.06981453  -6.9181 4.579e-12 ***
## sma2 -0.33183515  0.06454752  -5.1409 2.734e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ===============================================================
## Parameter Estimation through Maximum Likelihood (ML) Method for ARIMA( 1 1 11 )(1,1,2)[11]
##
## z test of coefficients:
##
##       Estimate Std. Error z value  Pr(>|z|)
## ar1   0.711867   0.145514  4.8921 9.977e-07 ***
## ma1  -0.498743   0.159611 -3.1247  0.001780 **
## ma2  -0.391432   0.070062 -5.5869 2.311e-08 ***
## ma3  -0.334523   0.125690 -2.6615  0.007780 **
## ma4   0.188787   0.141842  1.3310  0.183199
## ma5  -0.104159   0.078588 -1.3254  0.185045
## ma6  -0.073509   0.069493 -1.0578  0.290154
## ma7   0.164323   0.108384  1.5161  0.129489
## ma8  -0.055494   0.088301 -0.6285  0.529702
## ma9   0.359122   0.077196  4.6520 3.287e-06 ***
## ma10  0.194277   0.091009  2.1347  0.032785 *
## ma11 -0.448643   0.076488 -5.8655 4.477e-09 ***
## sar1  0.487415   0.127124  3.8342  0.000126 ***
## sma1 -1.075881   0.152297 -7.0644 1.613e-12 ***
## sma2  0.075882   0.145253  0.5224  0.601383
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ==============================================================
## Parameter Estimation through Combination (CSS-ML) Method for ARIMA( 1 1 11 )(1,1,2)[11]
##
## z test of coefficients:
##
##       Estimate Std. Error z value  Pr(>|z|)
## ar1   0.778612   0.125979  6.1805 6.391e-10 ***
## ma1  -0.590060   0.137582 -4.2888 1.797e-05 ***
## ma2  -0.389548   0.068430 -5.6926 1.251e-08 ***
## ma3  -0.280489   0.078933 -3.5535 0.0003801 ***
## ma4   0.187414   0.083666  2.2400 0.0250888 *
## ma5  -0.087895   0.063951 -1.3744 0.1693164
## ma6  -0.040028   0.075908 -0.5273 0.5979701
## ma7   0.151090   0.076425  1.9770 0.0480453 *
## ma8  -0.038245   0.063619 -0.6012 0.5477331
## ma9   0.364645   0.073312  4.9739 6.562e-07 ***
## ma10  0.114545   0.112783  1.0156 0.3098070
## ma11 -0.391426   0.087700 -4.4632 8.073e-06 ***
## sar1  0.468507   0.140062  3.3450 0.0008229 ***
## sma1 -1.104213   0.163916 -6.7365 1.623e-11 ***
## sma2  0.104214   0.157404  0.6621 0.5079179
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ==============================================================
```

```r
BC_sunspot_TS %>%
  check_residuals(order = c(1,1,11),
                  seasonal_order = c(1,1,2),
                  period = 11,
                  acf_lag = 70,
                  pacf_lag = 70,
                  res_main = 'Figure 17.1: Residuals of model:
                   ↪  SARIMA(1,1,11)(1,1,2)[11]',
                  hist_main = 'Figure 17.2: Histogram of residuals',
                  acf_main = 'Figure 17.3: ACF plot of residuals',
                  pacf_main = 'Figure 17.4: PACF plot of residuals')
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals
## W = 0.98801, p-value = 0.008884
```

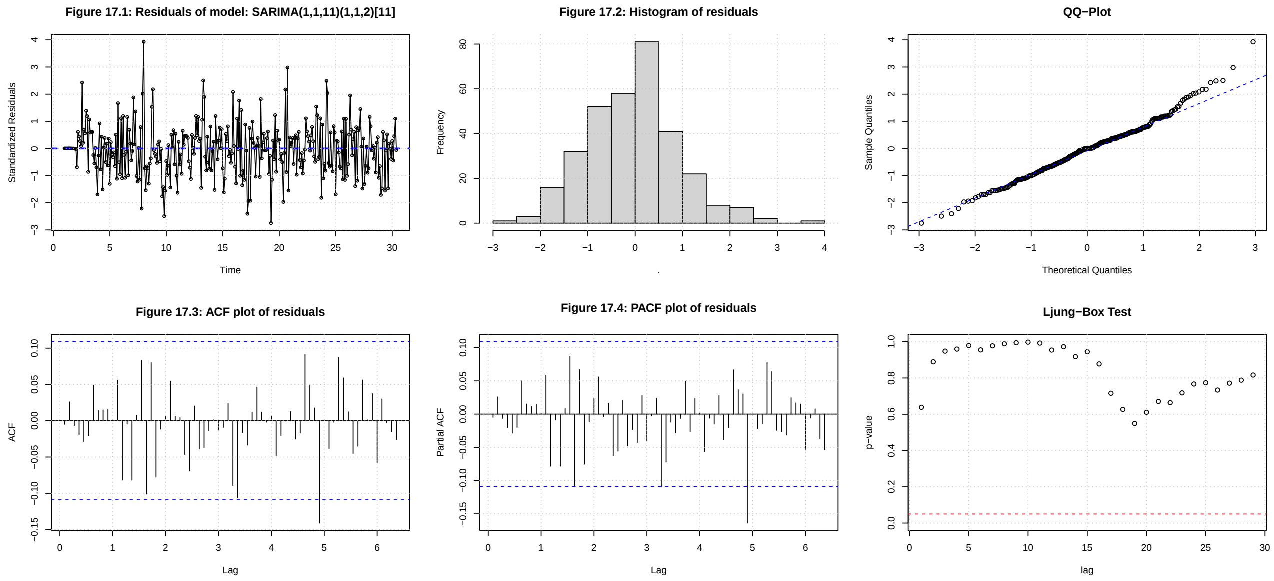


Figure 17.1: Residuals of model: SARIMA(1,1,11)(1,1,2)[11]

Figure 17.2: Histogram of residuals

QQ−Plot

Figure 17.3: ACF plot of residuals

Figure 17.4: PACF plot of residuals

Ljung−Box Test

Similar to the previous models, the SARIMA(1,1,11)(1,1,2)[11] model shows highly significant higher-order parameters. This model also does not exhibit normal behavior in the residuals. Only one significant lag is observed in the ACF and PACF plots, which is neglected as it does not indicate significant correlation, as confirmed by the Ljung-Box test.

8.5 Model 5: SARIMA(3,1,2)(1,1,2)[11]

```
model_312 <- BC_sunspot_TS %>% arima_summary(pdq_order = c(3,1,2),
                                             seasonal_order = c(1,1,2),
                                             period = 11)
```

```
##
## Parameter Estimation through Least Squares (CSS) Method for ARIMA( 3 1 2 )(1,1,2)[11]
##
## z test of coefficients:
##
##       Estimate Std. Error  z value  Pr(>|z|)
## ar1    1.790646   0.068266  26.2304 < 2.2e-16 ***
## ar2   -1.218089   0.097964 -12.4340 < 2.2e-16 ***
## ar3    0.199304   0.058493   3.4073 0.0006560 ***
## ma1   -1.626966   0.065751 -24.7444 < 2.2e-16 ***
## ma2    0.718847   0.060341  11.9130 < 2.2e-16 ***
## sar1 -0.450917   0.135851  -3.3192 0.0009027 ***
## sma1 -0.423546   0.135383  -3.1285 0.0017571 **
## sma2 -0.457851   0.129445  -3.5370 0.0004047 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ============================================================
## Parameter Estimation through Maximum Likelihood (ML) Method for ARIMA( 3 1 2 )(1,1,2)[11]
##
## z test of coefficients:
##
##       Estimate Std. Error  z value Pr(>|z|)
## ar1   1.819015   0.099066  18.3616  < 2e-16 ***
## ar2  -1.254693   0.147558  -8.5031  < 2e-16 ***
## ar3   0.207882   0.086068   2.4153  0.01572 *
## ma1  -1.625103   0.076349 -21.2851  < 2e-16 ***
## ma2   0.724447   0.069180  10.4719  < 2e-16 ***
## sar1  0.882149        NaN      NaN      NaN
## sma1 -1.896019        NaN      NaN      NaN
## sma2  0.896200        NaN      NaN      NaN
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ==============================================================
## Parameter Estimation through Combination (CSS-ML) Method for ARIMA( 3 1 2 )(1,1,2)[11]
##
## z test of coefficients:
##
##       Estimate Std. Error  z value  Pr(>|z|)
## ar1   1.797352   0.110070  16.3292 < 2.2e-16 ***
## ar2  -1.223434   0.159725  -7.6596 1.865e-14 ***
## ar3   0.197962   0.090463   2.1883  0.028646 *
## ma1  -1.609140   0.087424 -18.4061 < 2.2e-16 ***
## ma2   0.707123   0.079912   8.8488 < 2.2e-16 ***
## sar1  0.444716   0.505635   0.8795  0.379119
## sma1 -1.389582   0.513479  -2.7062  0.006806 **
## sma2  0.389593   0.511766   0.7613  0.446495
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ==============================================================
```

```
BC_sunspot_TS %>%
  check_residuals(order = c(3,1,2),
                  seasonal_order = c(1,1,2),
                  period = 11,
                  acf_lag = 70,
                  pacf_lag = 70,
                  res_main = 'Figure 18.1: Residuals of model:
                   ↪   SARIMA(3,1,2)(1,1,2)[11]',
                  hist_main = 'Figure 18.2: Histogram of residuals',
                  acf_main = 'Figure 18.3: ACF plot of residuals',
                  pacf_main = 'Figure 18.4: PACF plot of residuals')
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals
## W = 0.98947, p-value = 0.01953
```

Figure 18.1: Residuals of model: SARIMA(3,1,2)(1,1,2)[11]

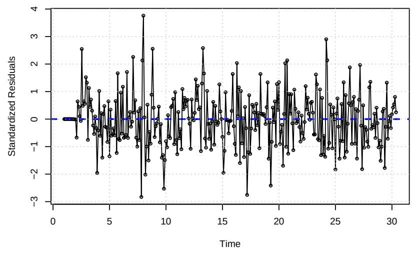
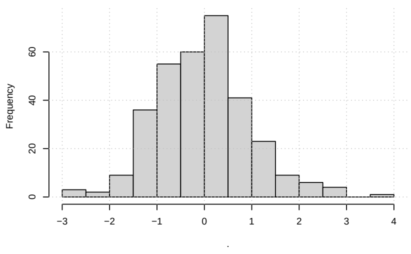


Figure 18.2: Histogram of residuals



QQ−Plot

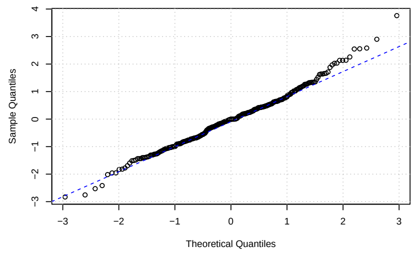


Figure 18.3: ACF plot of residuals

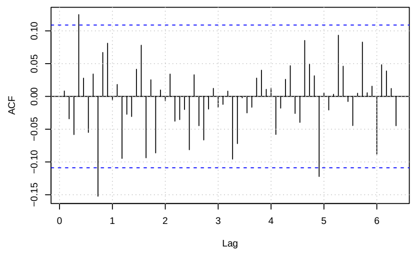
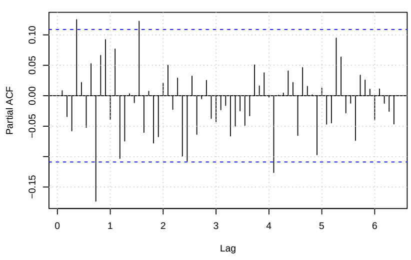
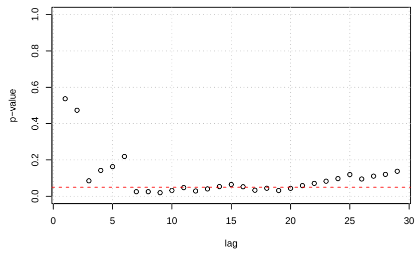


Figure 18.4: PACF plot of residuals



Ljung−Box Test



The SARIMA(3,1,2)(1,1,2)[11] model is comparatively smaller than its predecessors. Visual assessment of the histogram and QQ plot shows some signs of normality, but this is not supported by the Shapiro-Wilk test. The ACF and PACF plots reveal many significant lags, which cannot be neglected, as the Ljung-Box test indicates a significant number of lags fall below the conventional 0.05 value. This suggests that there is autocorrelation left in the residuals that the model fails to capture.

8.6 Model 6: SARIMA(3,1,5)(1,1,2)[11]

```
model_315 <- BC_sunspot_TS %>% arima_summary(pdq_order = c(3,1,5),
                                             seasonal_order = c(1,1,2),
                                             period = 11)
```

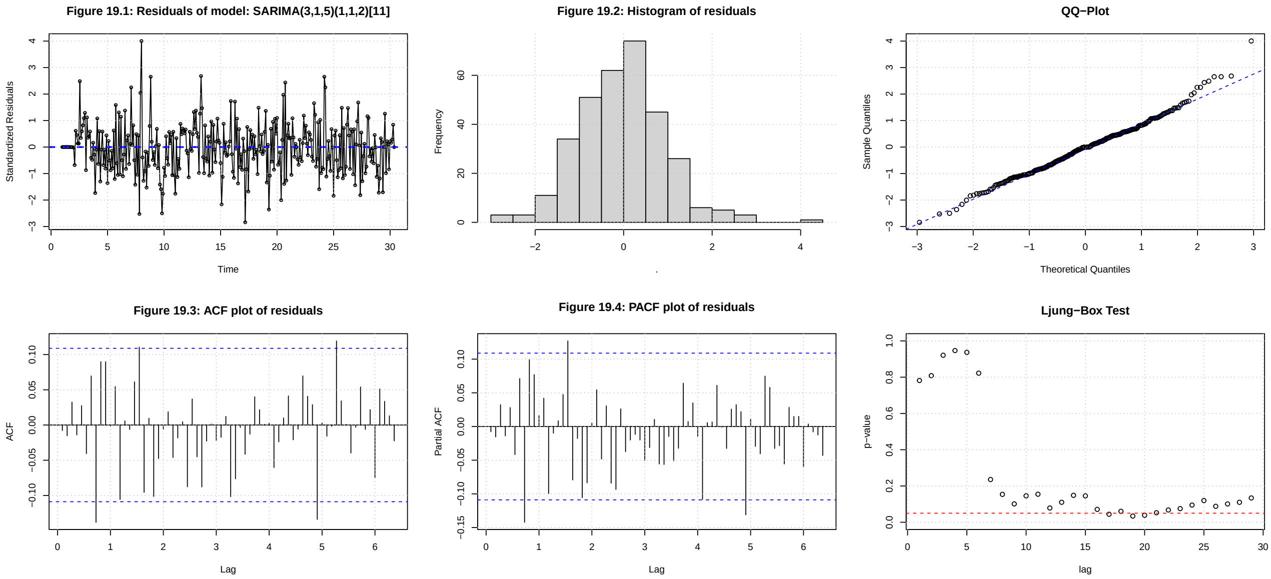
```
##
## Parameter Estimation through Least Squares (CSS) Method for ARIMA( 3 1 5 )(1,1,2)[11]
##
## z test of coefficients:
##
##       Estimate Std. Error z value  Pr(>|z|)
## ar1  -0.329712   0.057408 -5.7433 9.285e-09 ***
## ar2  -0.449662   0.047505 -9.4655 < 2.2e-16 ***
## ar3   0.501466   0.053695  9.3391 < 2.2e-16 ***
## ma1   0.660765   0.055219 11.9663 < 2.2e-16 ***
## ma2   0.461487   0.027020 17.0797 < 2.2e-16 ***
## ma3  -0.868659        NaN     NaN       NaN
## ma4  -0.549594   0.068804 -7.9878 1.373e-15 ***
## ma5  -0.305453   0.051656 -5.9132 3.355e-09 ***
## sar1 -0.361001   0.047585 -7.5865 3.287e-14 ***
## sma1 -0.372762   0.065269 -5.7112 1.122e-08 ***
## sma2 -0.505125   0.058244 -8.6725 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ==============================================================
## Parameter Estimation through Maximum Likelihood (ML) Method for ARIMA( 3 1 5 )(1,1,2)[11]
##
```

```
## z test of coefficients:
##
##        Estimate Std. Error z value  Pr(>|z|)
## ar1   1.362088   0.172454  7.8983 2.828e-15 ***
## ar2  -0.523457   0.271924 -1.9250  0.054228 .
## ar3  -0.232548   0.152812 -1.5218  0.128062
## ma1  -1.143793   0.168057 -6.8060 1.004e-11 ***
## ma2   0.049714   0.218961  0.2270  0.820389
## ma3   0.105341   0.090939  1.1584  0.246715
## ma4   0.380705   0.119429  3.1877  0.001434 **
## ma5  -0.246632   0.076892 -3.2075  0.001339 **
## sar1  0.293455   1.215177  0.2415  0.809174
## sma1 -1.272513   1.217819 -1.0449  0.296064
## sma2  0.272527   1.217055  0.2239  0.822817
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ==============================================================
## Parameter Estimation through Combination (CSS-ML) Method for ARIMA( 3 1 5 )(1,1,2)[11]
##
## z test of coefficients:
##
##        Estimate Std. Error z value  Pr(>|z|)
## ar1   1.362088   0.172454  7.8983 2.828e-15 ***
## ar2  -0.523457   0.271924 -1.9250  0.054228 .
## ar3  -0.232548   0.152812 -1.5218  0.128062
## ma1  -1.143793   0.168057 -6.8060 1.004e-11 ***
## ma2   0.049714   0.218961  0.2270  0.820389
## ma3   0.105341   0.090939  1.1584  0.246715
## ma4   0.380705   0.119429  3.1877  0.001434 **
## ma5  -0.246632   0.076892 -3.2075  0.001339 **
## sar1  0.293455   1.215177  0.2415  0.809174
## sma1 -1.272513   1.217819 -1.0449  0.296064
## sma2  0.272527   1.217055  0.2239  0.822817
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ==============================================================
```

```r
BC_sunspot_TS %>%
  check_residuals(order = c(3,1,5),
                  seasonal_order = c(1,1,2),
                  period = 11,
                  acf_lag = 70,
                  pacf_lag = 70,
                  res_main = 'Figure 19.1: Residuals of model:
                   ↪  SARIMA(3,1,5)(1,1,2)[11]',
                  hist_main = 'Figure 19.2: Histogram of residuals',
                  acf_main = 'Figure 19.3: ACF plot of residuals',
                  pacf_main = 'Figure 19.4: PACF plot of residuals')
```

```
##
##  Shapiro-Wilk normality test
```

```
##
## data:  residuals
## W = 0.98887, p-value = 0.0141
```



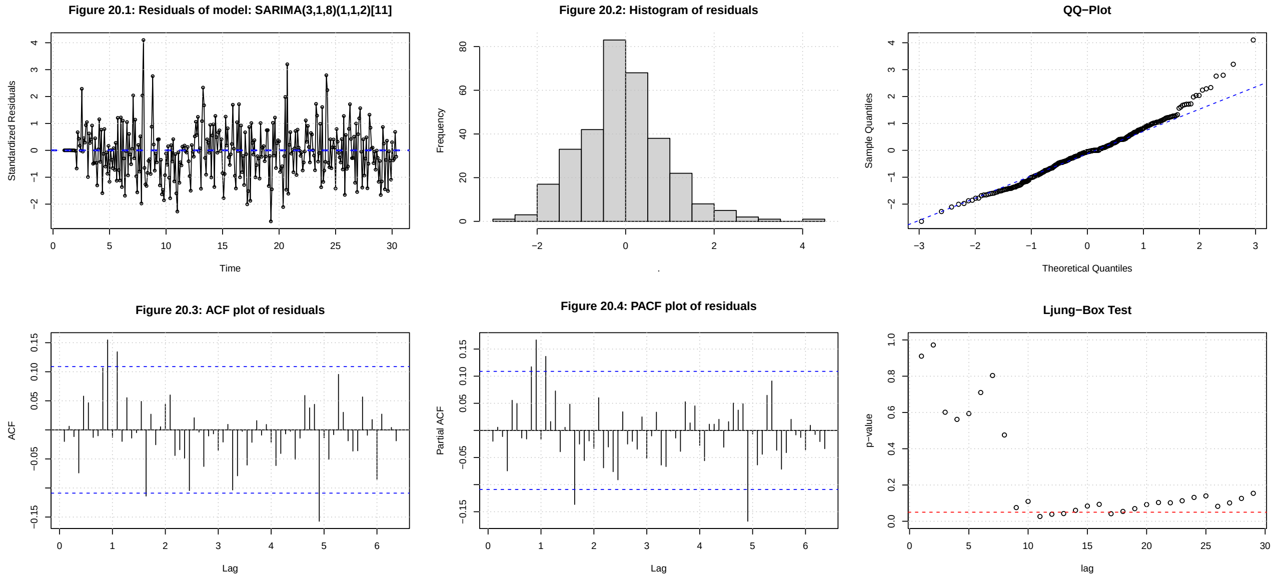
The SARIMA(3,1,5)(1,1,2)[11] model has very few significant parameters. Additionally, the ACF and PACF plots reveal multiple significant lags. The Ljung-Box test confirms that these lags indicate signs of autocorrelation due to their low p-values, suggesting that the model does not adequately capture all patterns in the data. Furthermore, the residuals do not exhibit normality, as supported by both visual assessments and formal tests. This indicates that the model may not be the best fit for the series, as it leaves behind significant autocorrelation and fails to produce normally distributed residuals.

8.7 Model 7: SARIMA(3,1,8)(1,1,2)[11]

```
model_318 <- BC_sunspot_TS %>% arima_summary(pdq_order = c(3,1,8),
                                             seasonal_order = c(1,1,2),
                                             period = 11)
```

```
##
## Parameter Estimation through Least Squares (CSS) Method for ARIMA( 3 1 8 )(1,1,2)[11]
##
## z test of coefficients:
##
##        Estimate Std. Error  z value  Pr(>|z|)
## ar1    1.817218   0.111387  16.3145 < 2.2e-16 ***
## ar2   -1.306665   0.164410  -7.9476 1.902e-15 ***
## ar3    0.252059   0.097753   2.5785 0.0099220 **
## ma1   -1.635106   0.124701 -13.1122 < 2.2e-16 ***
## ma2    0.730341   0.176225   4.1444 3.407e-05 ***
## ma3   -0.010946   0.143755  -0.0761 0.9393025
## ma4    0.270307   0.143261   1.8868 0.0591845 .
## ma5   -0.450055   0.173291  -2.5971 0.0094012 **
## ma6    0.106728   0.159548   0.6689 0.5035318
## ma7    0.124713   0.146896   0.8490 0.3958881
## ma8   -0.034093   0.085153  -0.4004 0.6888787
## sar1  -0.324657   0.180877  -1.7949 0.0726687 .
## sma1 -0.564464   0.169345  -3.3332 0.0008584 ***
## sma2 -0.315306   0.156887  -2.0098 0.0444556 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ==================================================================
## Parameter Estimation through Maximum Likelihood (ML) Method for ARIMA( 3 1 8 )(1,1,2)[11]
##
## z test of coefficients:
##
##       Estimate Std. Error z value  Pr(>|z|)
## ar1   0.698778   0.173128  4.0362 5.432e-05 ***
## ar2   0.011203   0.188384  0.0595 0.9525795
## ar3  -0.526200   0.112221 -4.6890 2.746e-06 ***
## ma1  -0.484913   0.182230 -2.6610 0.0077911 **
## ma2  -0.374090   0.167450 -2.2340 0.0254809 *
## ma3   0.237729   0.075399  3.1530 0.0016162 **
## ma4   0.306265   0.082010  3.7345 0.0001881 ***
## ma5  -0.366158   0.098036 -3.7349 0.0001878 ***
## ma6  -0.359779   0.097582 -3.6869 0.0002270 ***
## ma7   0.201345   0.098578  2.0425 0.0411020 *
## ma8  -0.160398   0.075329 -2.1293 0.0332293 *
## sar1  0.488706   0.158679  3.0798 0.0020711 **
## sma1 -1.242670   0.178347 -6.9677 3.222e-12 ***
## sma2  0.242671   0.173454  1.3991 0.1617976
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ==================================================================
## Parameter Estimation through Combination (CSS-ML) Method for ARIMA( 3 1 8 )(1,1,2)[11]
##
## z test of coefficients:
##
##       Estimate Std. Error z value  Pr(>|z|)
## ar1   0.698778   0.173128  4.0362 5.432e-05 ***
## ar2   0.011203   0.188384  0.0595 0.9525795
## ar3  -0.526200   0.112221 -4.6890 2.746e-06 ***
## ma1  -0.484913   0.182230 -2.6610 0.0077911 **
## ma2  -0.374090   0.167450 -2.2340 0.0254809 *
## ma3   0.237729   0.075399  3.1530 0.0016162 **
## ma4   0.306265   0.082010  3.7345 0.0001881 ***
## ma5  -0.366158   0.098036 -3.7349 0.0001878 ***
## ma6  -0.359779   0.097582 -3.6869 0.0002270 ***
## ma7   0.201345   0.098578  2.0425 0.0411020 *
## ma8  -0.160398   0.075329 -2.1293 0.0332293 *
## sar1  0.488706   0.158679  3.0798 0.0020711 **
## sma1 -1.242670   0.178347 -6.9677 3.222e-12 ***
## sma2  0.242671   0.173454  1.3991 0.1617976
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ==================================================================
BC_sunspot_TS %>%
  check_residuals(order = c(3,1,8),
                  seasonal_order = c(1,1,2),
                  period = 11,
                  acf_lag = 70,
                  pacf_lag = 70,
                  res_main = 'Figure 20.1: Residuals of model:
                   ↪  SARIMA(3,1,8)(1,1,2)[11]',
                  hist_main = 'Figure 20.2: Histogram of residuals',
                  acf_main = 'Figure 20.3: ACF plot of residuals',
                  pacf_main = 'Figure 20.4: PACF plot of residuals')
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals
## W = 0.98177, p-value = 0.0003936
```



The SARIMA(3,1,8)(1,1,2)[11] model demonstrates a regression in normality, as indicated by both visual assessments and formal tests. The histogram and QQ plot suggest deviations from normality, which is confirmed by the Shapiro-Wilk test. Additionally, the Ljung-Box test results show several lags with p-values falling below the 0.05 threshold, indicating the presence of autocorrelation that the model fails to capture. Despite these issues, the model does have almost all significant parameters, suggesting that it is able to capture some of the underlying patterns in the series. However, the remaining autocorrelation and non-normal residuals suggest that the model might not be fully adequate in capturing the complexities of the data.

8.8 Model 8: SARIMA(3,1,10)(1,1,2)[11]

```
model_3110 <- BC_sunspot_TS %>% arima_summary(pdq_order = c(3,1,10),
                                              seasonal_order = c(1,1,2),
                                              period = 11)
```

```
##
```

```
## Parameter Estimation through Least Squares (CSS) Method for ARIMA( 3 1 10 )(1,1,2)[11]
##
## z test of coefficients:
##
##       Estimate Std. Error z value  Pr(>|z|)
## ar1   0.289452   0.172307  1.6799 0.0929838 .
## ar2   0.351445   0.137027  2.5648 0.0103240 *
## ar3  -0.585328   0.103873 -5.6350 1.750e-08 ***
## ma1  -0.059451   0.180508 -0.3294 0.7418892
## ma2  -0.597815   0.121468 -4.9216 8.585e-07 ***
## ma3   0.161813   0.116612  1.3876 0.1652535
## ma4   0.116985   0.082928  1.4107 0.1583398
## ma5  -0.190418   0.073237 -2.6000 0.0093215 **
## ma6  -0.281050   0.068688 -4.0917 4.283e-05 ***
## ma7   0.052941   0.072629  0.7289 0.4660488
## ma8  -0.111868   0.072838 -1.5359 0.1245721
## ma9   0.203798   0.063312  3.2189 0.0012867 **
## ma10  0.279201   0.087333  3.1970 0.0013888 **
## sar1 -0.516777   0.147345 -3.5073 0.0004527 ***
## sma1 -0.320078   0.147532 -2.1696 0.0300408 *
## sma2 -0.460649   0.132753 -3.4700 0.0005205 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ==============================================================
## Parameter Estimation through Maximum Likelihood (ML) Method for ARIMA( 3 1 10 )(1,1,2)[11]
##
## z test of coefficients:
##
##       Estimate Std. Error z value  Pr(>|z|)
## ar1   0.184791   0.116823  1.5818   0.11369
## ar2   0.462679   0.061117  7.5704 3.721e-14 ***
## ar3  -0.636921   0.093157 -6.8371 8.081e-12 ***
## ma1   0.082165   0.121796  0.6746   0.49992
## ma2  -0.685082   0.084687 -8.0896 5.988e-16 ***
## ma3   0.137264   0.122556  1.1200   0.26271
## ma4   0.143143   0.089790  1.5942   0.11089
## ma5  -0.158845   0.069575 -2.2831   0.02243 *
## ma6  -0.378480   0.065064 -5.8170 5.991e-09 ***
## ma7   0.080634   0.083379  0.9671   0.33350
## ma8  -0.101732   0.070262 -1.4479   0.14765
## ma9   0.145977   0.067943  2.1485   0.03167 *
## ma10  0.309350   0.065676  4.7102 2.474e-06 ***
## sar1  0.527478   0.270317  1.9513   0.05102 .
## sma1 -1.421083   0.288360 -4.9282 8.301e-07 ***
## sma2  0.421094   0.285287  1.4760   0.13993
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ==============================================================
## Parameter Estimation through Combination (CSS-ML) Method for ARIMA( 3 1 10 )(1,1,2)[11]
##
```

```
## z test of coefficients:
##
##       Estimate Std. Error z value  Pr(>|z|)
## ar1   0.184569   0.116870  1.5793   0.11427
## ar2   0.462719   0.061117  7.5710 3.704e-14 ***
## ar3  -0.636857   0.093215 -6.8321 8.367e-12 ***
## ma1   0.082391   0.121827  0.6763   0.49885
## ma2  -0.685074   0.084691 -8.0891 6.009e-16 ***
## ma3   0.137151   0.122609  1.1186   0.26331
## ma4   0.143077   0.089810  1.5931   0.11114
## ma5  -0.158785   0.069573 -2.2823   0.02247 *
## ma6  -0.378484   0.065058 -5.8176 5.969e-09 ***
## ma7   0.080503   0.083375  0.9655   0.33427
## ma8  -0.101683   0.070259 -1.4473   0.14782
## ma9   0.146006   0.067941  2.1490   0.03164 *
## ma10  0.309432   0.065661  4.7126 2.446e-06 ***
## sar1  0.528802   0.269543  1.9618   0.04978 *
## sma1 -1.422414   0.287769 -4.9429 7.697e-07 ***
## sma2  0.422420   0.284699  1.4837   0.13788
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ==============================================================
```

```r
BC_sunspot_TS %>%
  check_residuals(order = c(3,1,10),
                  seasonal_order = c(1,1,2),
                  period = 11,
                  acf_lag = 70,
                  pacf_lag = 70,
                  res_main = 'Figure 21.1: Residuals of model:
                   ↪  SARIMA(3,1,10)(1,1,2)[11]',
                  hist_main = 'Figure 21.2: Histogram of residuals',
                  acf_main = 'Figure 21.3: ACF plot of residuals',
                  pacf_main = 'Figure 21.4: PACF plot of residuals')
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals
## W = 0.98879, p-value = 0.01352
```

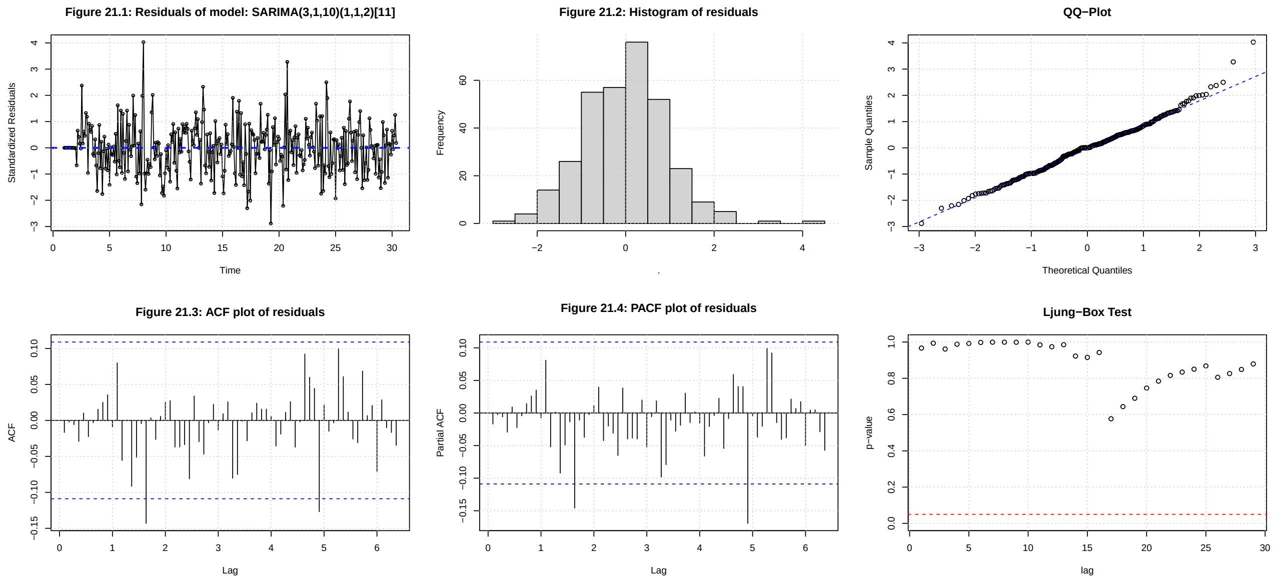


Figure 21.1: Residuals of model: SARIMA(3,1,10)(1,1,2)[11]

Figure 21.2: Histogram of residuals

QQ−Plot

Figure 21.3: ACF plot of residuals

Figure 21.4: PACF plot of residuals

Ljung−Box Test

The SARIMA(3,1,10)(1,1,2)[11] model exhibits a notable improvement in the normality of residuals, although it still falls slightly below the conventional threshold in the Shapiro-Wilk test. Visual assessments, including the histogram and QQ plot, indicate closer adherence to normality. The ACF and PACF plots reveal only two near-significant lags, which can be reasonably neglected. The Ljung-Box test confirms that there is minimal autocorrelation left uncaptured by the model.

This model demonstrates that higher-order terms are highly significant in effectively capturing the series' behavior. The improved normality of residuals and the reduction in significant autocorrelation suggest that the SARIMA(3,1,10)(1,1,2)[11] model is more adept at fitting the data compared to previous models, despite minor deviations in normality.

8.9 Model 9: SARIMA(5,1,2)(1,1,2)[11]

```
model_512 <- BC_sunspot_TS %>% arima_summary(pdq_order = c(5,1,2),
                                             seasonal_order = c(1,1,2),
                                             period = 11)
```

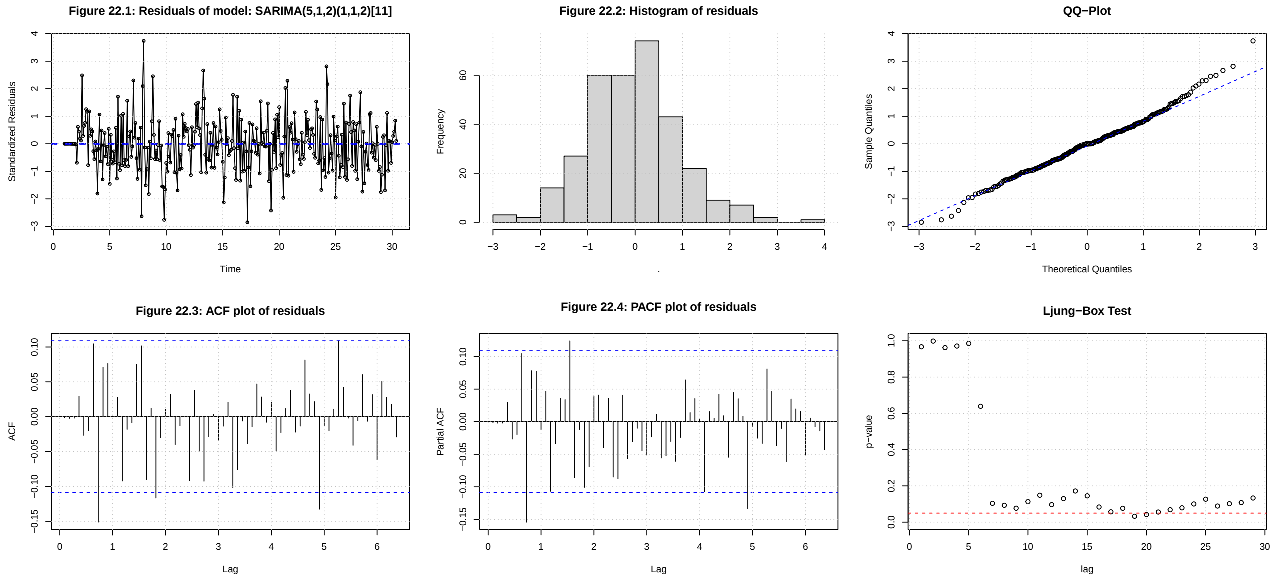
```
##
## Parameter Estimation through Least Squares (CSS) Method for ARIMA( 5 1 2 )(1,1,2)[11]
##
## z test of coefficients:
##
##       Estimate Std. Error z value  Pr(>|z|)
## ar1   1.569604   0.217445  7.2184 5.261e-13 ***
## ar2  -1.056189   0.230641 -4.5794 4.664e-06 ***
## ar3   0.130095   0.151890  0.8565 0.3917138
## ar4   0.061007   0.108976  0.5598 0.5756022
## ar5  -0.132993   0.094721 -1.4041 0.1603038
## ma1  -1.377920   0.219863 -6.2672 3.676e-10 ***
## ma2   0.553503   0.165000  3.3546 0.0007949 ***
## sar1 -0.615931   0.107597 -5.7244 1.038e-08 ***
## sma1 -0.249017   0.107777 -2.3105 0.0208618 *
## sma2 -0.573655   0.103145 -5.5617 2.672e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## ==============================================================
## Parameter Estimation through Maximum Likelihood (ML) Method for ARIMA( 5 1 2 )(1,1,2)[11]
##
## z test of coefficients:
##
##       Estimate Std. Error z value  Pr(>|z|)
## ar1   1.392899   0.184359  7.5554 4.176e-14 ***
## ar2  -0.889126   0.215751 -4.1211 3.771e-05 ***
## ar3   0.039859   0.134700  0.2959  0.767299
## ar4   0.127543   0.099312  1.2843  0.199050
## ar5  -0.231456   0.072302 -3.2012  0.001368 **
## ma1  -1.191472   0.180998 -6.5828 4.617e-11 ***
## ma2   0.406844   0.155129  2.6226  0.008726 **
## sar1  0.256405   1.429280  0.1794  0.857628
## sma1 -1.230167   1.418061 -0.8675  0.385668
## sma2  0.230205   1.417378  0.1624  0.870978
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ==============================================================
## Parameter Estimation through Combination (CSS-ML) Method for ARIMA( 5 1 2 )(1,1,2)[11]
##
## z test of coefficients:
##
##       Estimate Std. Error z value  Pr(>|z|)
## ar1   1.410172   0.168275  8.3802 < 2.2e-16 ***
## ar2  -0.912011   0.198300 -4.5992 4.242e-06 ***
## ar3   0.054603   0.133654  0.4085  0.682879
## ar4   0.122537   0.100365  1.2209  0.222117
## ar5  -0.228573   0.072566 -3.1499  0.001633 **
## ma1  -1.202674   0.167430 -7.1832 6.811e-13 ***
## ma2   0.417026   0.143511  2.9059  0.003662 **
## sar1 -0.932308   0.152673 -6.1066 1.018e-09 ***
## sma1 -0.046576   0.149762 -0.3110  0.755800
## sma2 -0.953419   0.149108 -6.3942 1.614e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ==============================================================
```

```
BC_sunspot_TS %>%
  check_residuals(order = c(5,1,2),
                  seasonal_order = c(1,1,2),
                  period = 11,
                  acf_lag = 70,
                  pacf_lag = 70,
                  res_main = 'Figure 22.1: Residuals of model:
                   ↪  SARIMA(5,1,2)(1,1,2)[11]',
                  hist_main = 'Figure 22.2: Histogram of residuals',
                  acf_main = 'Figure 22.3: ACF plot of residuals',
                  pacf_main = 'Figure 22.4: PACF plot of residuals')
```

```
##
```

```
##  Shapiro-Wilk normality test
##
## data:  residuals
## W = 0.99033, p-value = 0.03119
```



The SARIMA(5,1,2)(1,1,2)[11] model, similar to its predecessor, demonstrates high significance in higher-order AR and MA terms. The model exhibits improved normality of residuals, as indicated by the Shapiro-Wilk test. However, despite this improvement, the ACF and PACF plots reveal a few significant lags. Notably, two of these lags fall below the conventional 0.05 threshold in the Ljung-Box test, indicating the presence of residual autocorrelation.

This suggests that while the SARIMA(5,1,2)(1,1,2)[11] model effectively captures much of the series' behavior, there are still some aspects that it might be overlooking. The significant lags in the ACF and PACF plots, combined with the Ljung-Box test results, imply that the model may not fully account for all underlying patterns in the data. Despite these minor shortcomings, the overall performance of the model, particularly in terms of normality and significance of parameters, marks a improvement over previous models.

8.10 Model 10: SARIMA(5,1,5)(1,1,2)[11]

```
model_515 <- BC_sunspot_TS %>% arima_summary(pdq_order = c(5,1,5),
                                             seasonal_order = c(1,1,2),
                                             period = 11)
```

```
##
## Parameter Estimation through Least Squares (CSS) Method for ARIMA( 5 1 5 )(1,1,2)[11]
##
## z test of coefficients:
##
##        Estimate Std. Error  z value  Pr(>|z|)
## ar1  -0.268027   0.172588  -1.5530  0.120426
## ar2   0.667045   0.075298   8.8587 < 2.2e-16 ***
## ar3   0.217796   0.089944   2.4215  0.015458 *
## ar4  -0.677481   0.060244 -11.2455 < 2.2e-16 ***
## ar5  -0.303725   0.131339  -2.3125  0.020748 *
## ma1   0.523364   0.172545   3.0332  0.002420 **
## ma2  -0.781487   0.092173  -8.4785 < 2.2e-16 ***
## ma3  -0.898594   0.100535  -8.9381 < 2.2e-16 ***
## ma4   0.344387   0.117914   2.9207  0.003493 **
## ma5   0.449857   0.098968   4.5455 5.481e-06 ***
## sar1 -0.699598   0.095105  -7.3560 1.895e-13 ***
## sma1 -0.143730   0.096382  -1.4913  0.135893
## sma2 -0.658493   0.094088  -6.9987 2.583e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ================================================================
## Parameter Estimation through Maximum Likelihood (ML) Method for ARIMA( 5 1 5 )(1,1,2)[11]
##
## z test of coefficients:
##
##       Estimate Std. Error z value  Pr(>|z|)
## ar1  -0.277536   0.207200 -1.3395  0.180422
## ar2   0.672451   0.080828  8.3195 < 2.2e-16 ***
## ar3   0.214189   0.112873  1.8976  0.057748 .
## ar4  -0.656728   0.067985 -9.6599 < 2.2e-16 ***
## ar5  -0.299659   0.155625 -1.9255  0.054165 .
## ma1   0.540818   0.194954  2.7741  0.005536 **
## ma2  -0.772104   0.091558 -8.4330 < 2.2e-16 ***
## ma3  -0.915020   0.116192 -7.8751 3.405e-15 ***
## ma4   0.301020   0.133663  2.2521  0.024317 *
## ma5   0.444973   0.108966  4.0836 4.435e-05 ***
## sar1  0.270561   0.610854  0.4429  0.657822
## sma1 -1.197002   0.617730 -1.9377  0.052655 .
## sma2  0.197012   0.616122  0.3198  0.749150
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ================================================================
## Parameter Estimation through Combination (CSS-ML) Method for ARIMA( 5 1 5 )(1,1,2)[11]
##
## z test of coefficients:
##
##       Estimate Std. Error  z value  Pr(>|z|)
## ar1  -0.299685   0.231203  -1.2962 0.1949060
## ar2   0.689063   0.083038   8.2981 < 2.2e-16 ***
## ar3   0.215656   0.121335   1.7774 0.0755088 .
## ar4  -0.667036   0.065870 -10.1265 < 2.2e-16 ***
## ar5  -0.328017   0.175645  -1.8675 0.0618327 .
## ma1   0.573204   0.217827   2.6315 0.0085017 **
## ma2  -0.780730   0.088001  -8.8718 < 2.2e-16 ***
## ma3  -0.928831   0.130116  -7.1385 9.437e-13 ***
## ma4   0.296202   0.142507   2.0785 0.0376623 *
## ma5   0.466896   0.122024   3.8263 0.0001301 ***
## sar1 -0.895801   0.183026  -4.8944 9.861e-07 ***
## sma1 -0.076661   0.174241  -0.4400 0.6599602
## sma2 -0.923326   0.173099  -5.3341 9.602e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ==============================================================
```

```
BC_sunspot_TS %>%
  check_residuals(order = c(5,1,5),
                  seasonal_order = c(1,1,2),
                  period = 11,
                  acf_lag = 70,
                  pacf_lag = 70,
                  res_main = 'Figure 23.1: Residuals of model:
                   ↪  SARIMA(5,1,5)(1,1,2)[11]',
                  hist_main = 'Figure 23.2: Histogram of residuals',
                  acf_main = 'Figure 23.3: ACF plot of residuals',
                  pacf_main = 'Figure 23.4: PACF plot of residuals')
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals
## W = 0.99029, p-value = 0.03047
```

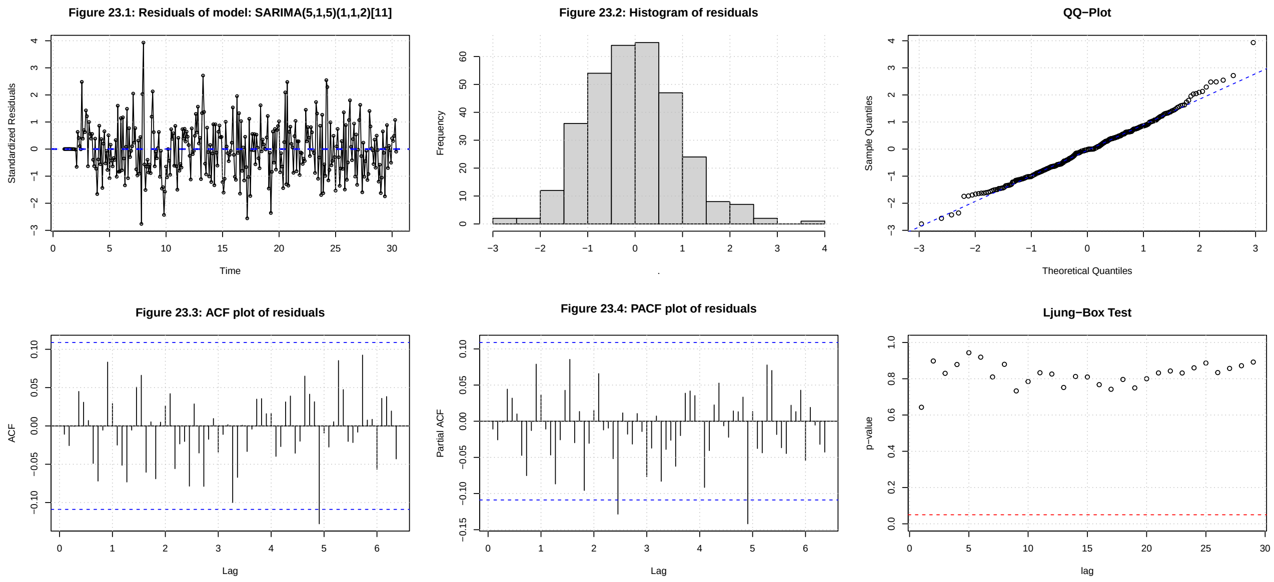


Figure 23.1: Residuals of model: SARIMA(5,1,5)(1,1,2)[11]

Figure 23.2: Histogram of residuals

QQ−Plot

Figure 23.3: ACF plot of residuals

Figure 23.4: PACF plot of residuals

Ljung−Box Test

The SARIMA(5,1,5)(1,1,2)[11] model demonstrates significant improvement across multiple fronts, including parameter significance and residual diagnostics. Almost all terms are significant, with only the lowest AR order and the lowest SMA order showing insignificance. Both visual assessments (histogram and QQ plot) and formal tests indicate a close alignment to normality, although the residuals are not perfectly normal, as the Shapiro-Wilk p-value remains slightly below the conventional 0.05 threshold.

The ACF and PACF plots reveal minimal significant lags, which can be considered negligible. The Ljung-Box test supports this, as all lags have p-values above the 0.05 threshold, indicating that the model captures the autocorrelation structure well.

In summary, the SARIMA(5,1,5)(1,1,2)[11] model effectively captures the series' behavior, with high parameter significance and residuals that closely approximate normality. The minimal significant lags in the ACF and PACF plots, corroborated by the Ljung-Box test, suggest that the model leaves little autocorrelation unaccounted for, marking a substantial improvement over previous models.

8.11 Model 11: SARIMA(6,1,2)(1,1,2)[11]

```
model_612<- BC_sunspot_TS %>% arima_summary(pdq_order = c(6,1,2),
                                            seasonal_order = c(1,1,2),
                                            period = 11)
```

```
##
## Parameter Estimation through Least Squares (CSS) Method for ARIMA( 6 1 2 )(1,1,2)[11]
##
## z test of coefficients:
##
##       Estimate Std. Error z value  Pr(>|z|)
## ar1   0.276480   0.332084  0.8326   0.40509
## ar2   0.207036   0.322680  0.6416   0.52112
## ar3  -0.400449   0.137840 -2.9052   0.00367 **
## ar4  -0.062141   0.082026 -0.7576   0.44870
## ar5  -0.126510   0.063588 -1.9895   0.04664 *
## ar6  -0.239713   0.090185 -2.6580   0.00786 **
## ma1  -0.067962   0.337450 -0.2014   0.84039
## ma2  -0.433218   0.251595 -1.7219   0.08509 .
## sar1 -0.630204   0.149981 -4.2019 2.647e-05 ***
## sma1 -0.210077   0.139687 -1.5039   0.13260
## sma2 -0.563772   0.131230 -4.2961 1.739e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ==============================================================
## Parameter Estimation through Maximum Likelihood (ML) Method for ARIMA( 6 1 2 )(1,1,2)[11]
##
## z test of coefficients:
##
##       Estimate Std. Error z value  Pr(>|z|)
## ar1   0.188915   0.193449  0.9766  0.328787
## ar2   0.301145   0.198719  1.5154  0.129664
## ar3  -0.445727   0.098699 -4.5160 6.301e-06 ***
## ar4  -0.044249   0.059864 -0.7392  0.459815
## ar5  -0.125213   0.057610 -2.1735  0.029745 *
## ar6  -0.284952   0.066871 -4.2612 2.033e-05 ***
## ma1   0.022143   0.196724  0.1126  0.910382
## ma2  -0.527439   0.161528 -3.2653  0.001093 **
## sar1  0.424834   0.471032  0.9019  0.367099
## sma1 -1.364426   0.478482 -2.8516  0.004350 **
## sma2  0.364427   0.476675  0.7645  0.444559
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ==============================================================
## Parameter Estimation through Combination (CSS-ML) Method for ARIMA( 6 1 2 )(1,1,2)[11]
##
```

```
## z test of coefficients:
##
##       Estimate Std. Error z value  Pr(>|z|)
## ar1   0.153010   0.198595  0.7705 0.4410280
## ar2   0.355712   0.207993  1.7102 0.0872263 .
## ar3  -0.476721   0.099604 -4.7862 1.700e-06 ***
## ar4  -0.042748   0.059550 -0.7179 0.4728487
## ar5  -0.126466   0.057700 -2.1918 0.0283965 *
## ar6  -0.296651   0.063478 -4.6733 2.964e-06 ***
## ma1   0.074605   0.201598  0.3701 0.7113301
## ma2  -0.576444   0.168600 -3.4190 0.0006285 ***
## sar1 -0.927653   0.138034 -6.7205 1.811e-11 ***
## sma1 -0.048376   0.138070 -0.3504 0.7260584
## sma2 -0.951622   0.137155 -6.9383 3.969e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ==============================================================
```

```
BC_sunspot_TS %>%
  check_residuals(order = c(6,1,2),
                  seasonal_order = c(1,1,2),
                  period = 11,
                  acf_lag = 70,
                  pacf_lag = 70,
                  res_main = 'Figure 24.1: Residuals of model:
                   ↪  SARIMA(6,1,2)(1,1,2)[11]',
                  hist_main = 'Figure 24.2: Histogram of residuals',
                  acf_main = 'Figure 24.3: ACF plot of residuals',
                  pacf_main = 'Figure 24.4: PACF plot of residuals')
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals
## W = 0.98853, p-value = 0.01172
```

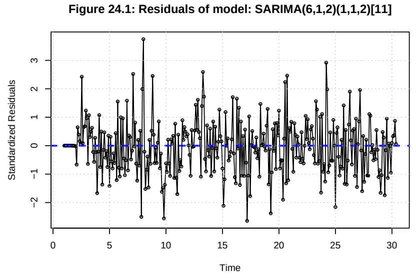


Figure 24.1: Residuals of model: SARIMA(6,1,2)(1,1,2)[11]

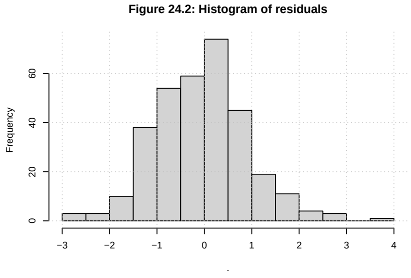
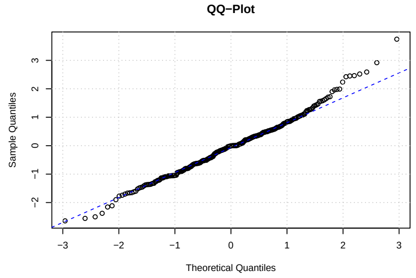


Figure 24.2: Histogram of residuals



QQ−Plot

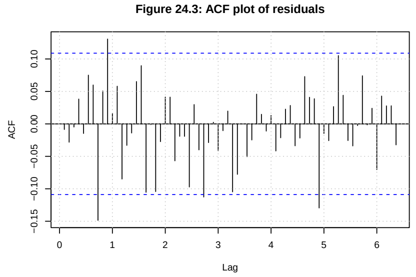


Figure 24.3: ACF plot of residuals

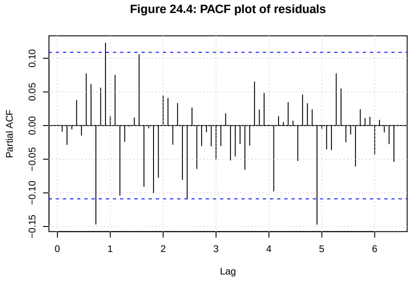
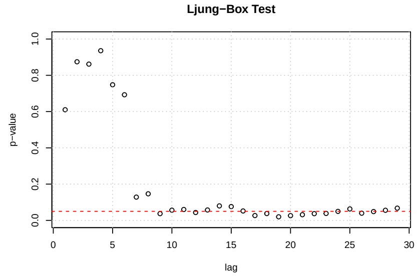


Figure 24.4: PACF plot of residuals



Ljung−Box Test

The SARIMA(6,1,2)(1,1,2)[11] model displays several significant parameters, indicating that it captures some key aspects of the series. However, the residual diagnostics reveal shortcomings. The Ljung-Box test results indicate a significant number of lags with p-values falling below the conventional threshold of 0.05. This suggests that the model fails to fully account for all autocorrelation in the data, leaving some underlying patterns uncaptured. Despite the significance of individual parameters, the presence of these significant lags points to residual autocorrelation, indicating that the model may not be entirely adequate in capturing the series' dynamics.

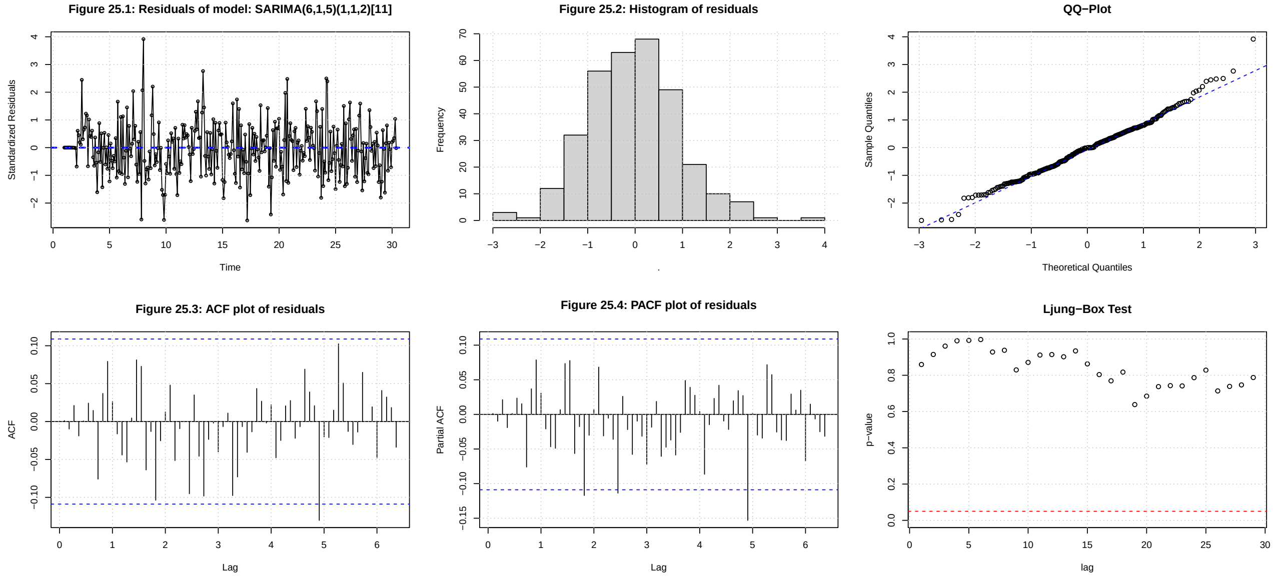
8.12 Model 12: SARIMA(6,1,5)(1,1,2)[11]

```r
model_615 <- BC_sunspot_TS %>% arima_summary(pdq_order = c(6,1,5),
                                             seasonal_order = c(1,1,2),
                                             period = 11)
```

```
##
## Parameter Estimation through Least Squares (CSS) Method for ARIMA( 6 1 5 )(1,1,2)[11]
##
## z test of coefficients:
##
##       Estimate Std. Error z value  Pr(>|z|)
## ar1  -0.319064   0.298361 -1.0694 0.2848935
## ar2   0.685835   0.192878  3.5558 0.0003768 ***
## ar3   0.257294   0.175372  1.4671 0.1423393
## ar4  -0.705732   0.089427 -7.8917 2.980e-15 ***
## ar5  -0.322463   0.239978 -1.3437 0.1790390
## ar6   0.016258   0.133639  0.1217 0.9031708
## ma1   0.557733   0.293920  1.8976 0.0577532 .
## ma2  -0.793040   0.155816 -5.0896 3.589e-07 ***
## ma3  -0.940468   0.243487 -3.8625 0.0001122 ***
## ma4   0.333066   0.157499  2.1147 0.0344545 *
## ma5   0.466388   0.238229  1.9577 0.0502616 .
## sar1 -0.719483   0.081921 -8.7827 < 2.2e-16 ***
## sma1 -0.129032   0.086515 -1.4914 0.1358477
## sma2 -0.670182   0.084288 -7.9511 1.848e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## =================================================================
## Parameter Estimation through Maximum Likelihood (ML) Method for ARIMA( 6 1 5 )(1,1,2)[11]
##
## z test of coefficients:
##
##       Estimate Std. Error z value  Pr(>|z|)
## ar1  -0.080213   0.372851 -0.2151   0.82966
## ar2   0.409771   0.204642  2.0024   0.04524 *
## ar3   0.089438   0.177128  0.5049   0.61360
## ar4  -0.544196   0.089827 -6.0583 1.376e-09 ***
## ar5  -0.151343   0.262068 -0.5775   0.56361
## ar6  -0.184039   0.112455 -1.6366   0.10172
## ma1   0.331672   0.375242  0.8839   0.37676
## ma2  -0.588482   0.148046 -3.9750 7.038e-05 ***
## ma3  -0.672445   0.266008 -2.5279   0.01147 *
## ma4   0.344608   0.213595  1.6134   0.10666
## ma5   0.188914   0.269322  0.7014   0.48303
## sar1  0.168961   0.587448  0.2876   0.77364
## sma1 -1.076123   0.586864 -1.8337   0.06670 .
## sma2  0.076145   0.585206  0.1301   0.89647
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## =================================================================
## Parameter Estimation through Combination (CSS-ML) Method for ARIMA( 6 1 5 )(1,1,2)[11]
##
## z test of coefficients:
##
##      Estimate Std. Error z value  Pr(>|z|)
## ar1   0.45338    0.47863  0.9473  0.343506
## ar2   0.16538    0.28553  0.5792  0.562451
## ar3  -0.14631    0.21911 -0.6677  0.504300
## ar4  -0.52013    0.10276 -5.0615 4.160e-07 ***
## ar5   0.21379    0.32685  0.6541  0.513054
## ar6  -0.27301    0.12007 -2.2737  0.022982 *
## ma1  -0.20198    0.48242 -0.4187  0.675448
## ma2  -0.47488    0.18752 -2.5324  0.011330 *
## ma3  -0.28920    0.35087 -0.8242  0.409810
## ma4   0.60062    0.20259  2.9648  0.003029 **
## ma5  -0.20008    0.38530 -0.5193  0.603566
## sar1 -0.85427    0.19657 -4.3459 1.387e-05 ***
## sma1 -0.11026    0.18861 -0.5846  0.558837
## sma2 -0.88961    0.18754 -4.7435 2.101e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## =================================================================
```

```r
BC_sunspot_TS %>%
  check_residuals(order = c(6,1,5),
                  seasonal_order = c(1,1,2),
                  period = 11,
                  acf_lag = 70,
                  pacf_lag = 70,
                  res_main = 'Figure 25.1: Residuals of model:
                   ↪  SARIMA(6,1,5)(1,1,2)[11]',
                  hist_main = 'Figure 25.2: Histogram of residuals',
                  acf_main = 'Figure 25.3: ACF plot of residuals',
                  pacf_main = 'Figure 25.4: PACF plot of residuals')
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals
## W = 0.98952, p-value = 0.02006
```



Compared to other models, the SARIMA(6,1,5)(1,1,2)[11] model has a reduced number of significant parameters. Despite this, it effectively captures almost all autocorrelation in the series, as evidenced by the Ljung-Box test, where nearly all lags have high p-values, indicating minimal residual autocorrelation. Additionally, the residuals exhibit some normality, supported by both visual assessments (histogram and QQ plot) and the Shapiro-Wilk test. This suggests that the model, while having fewer significant parameters, manages to adequately capture the underlying patterns in the data, ensuring that residuals are close to normal and free from significant autocorrelation.

8.13 Model 13: SARIMA(6,1,8)(1,1,2)[11]

```
model_618 <- BC_sunspot_TS %>% arima_summary(pdq_order = c(6,1,8),
                                             seasonal_order = c(1,1,2),
                                             period = 11)
```

```
##
## Parameter Estimation through Least Squares (CSS) Method for ARIMA( 6 1 8 )(1,1,2)[11]
##
## z test of coefficients:
##
```

```
##        Estimate Std. Error z value  Pr(>|z|)    
## ar1   1.0871661  0.1838503  5.9133 3.353e-09 ***
## ar2  -0.1302195  0.2138516 -0.6089  0.542575    
## ar3  -0.6472835  0.2165250 -2.9894  0.002795 ** 
## ar4  -0.1682060  0.2509596 -0.6703  0.502698    
## ar5   0.6639180  0.1509140  4.3993 1.086e-05 ***
## ar6  -0.4614879  0.1010150 -4.5685 4.912e-06 ***
## ma1  -0.8793468  0.1902652 -4.6217 3.806e-06 ***
## ma2  -0.3084752  0.1960177 -1.5737  0.115554    
## ma3   0.4883894  0.2371014  2.0598  0.039414 *  
## ma4   0.5675075  0.1890738  3.0015  0.002686 ** 
## ma5  -0.9000583  0.1486688 -6.0541 1.412e-09 ***
## ma6   0.0186002  0.1485702  0.1252  0.900369    
## ma7   0.3314817  0.1281891  2.5859  0.009713 ** 
## ma8   0.0027045  0.1117616  0.0242  0.980694    
## sar1 -0.4134783  0.1337037 -3.0925  0.001985 ** 
## sma1 -0.4194273  0.1390580 -3.0162  0.002560 ** 
## sma2 -0.3716523  0.1220402 -3.0453  0.002324 ** 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## ===============================================================
## Parameter Estimation through Maximum Likelihood (ML) Method for ARIMA( 6 1 8 )(1,1,2)[11]
## 
## z test of coefficients:
## 
##        Estimate Std. Error z value  Pr(>|z|)    
## ar1   0.3162665  0.1115409  2.8354  0.004576 ** 
## ar2  -0.4012236  0.1017636 -3.9427 8.057e-05 ***
## ar3  -0.0212418  0.1167344 -0.1820  0.855609    
## ar4  -0.0535545  0.1182081 -0.4531  0.650510    
## ar5  -0.0247772  0.1355476 -0.1828  0.854960    
## ar6  -0.6510250  0.0794470 -8.1945 2.517e-16 ***
## ma1  -0.0712261  0.1179004 -0.6041  0.545763    
## ma2   0.1371151  0.1232146  1.1128  0.265787    
## ma3  -0.2777769  0.0976579 -2.8444  0.004450 ** 
## ma4  -0.1147404  0.1514440 -0.7576  0.448665    
## ma5  -0.3584375  0.1016392 -3.5266  0.000421 ***
## ma6   0.4357221  0.0998352  4.3644 1.275e-05 ***
## ma7   0.1760067  0.0862574  2.0405  0.041302 *  
## ma8  -0.2220500  0.0781097 -2.8428  0.004472 ** 
## sar1  0.0087842  0.6688080  0.0131  0.989521    
## sma1 -0.9096553  0.6697913 -1.3581  0.174426    
## sma2 -0.0903023  0.6681996 -0.1351  0.892499    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## ===============================================================
## Parameter Estimation through Combination (CSS-ML) Method for ARIMA( 6 1 8 )(1,1,2)[11]
## 
## z test of coefficients:
## 
```

```
##       Estimate Std. Error  z value  Pr(>|z|)
## ar1   1.086384   0.093096  11.6695 < 2.2e-16 ***
## ar2  -0.454997   0.159956  -2.8445  0.004448 **
## ar3  -0.521537   0.186322  -2.7991  0.005124 **
## ar4  -0.039534   0.178226  -0.2218  0.824455
## ar5   0.570852   0.135568   4.2108 2.545e-05 ***
## ar6  -0.646970   0.067121  -9.6388 < 2.2e-16 ***
## ma1  -0.863972   0.107546  -8.0335 9.471e-16 ***
## ma2   0.027189   0.158360   0.1717  0.863682
## ma3   0.419818   0.161945   2.5923  0.009532 **
## ma4   0.357519   0.120755   2.9607  0.003069 **
## ma5  -0.949151   0.089136 -10.6484 < 2.2e-16 ***
## ma6   0.234048   0.113768   2.0572  0.039663 *
## ma7   0.308317   0.114738   2.6871  0.007207 **
## ma8  -0.107676   0.084853  -1.2690  0.204454
## sar1 -0.700741        NaN      NaN       NaN
## sma1 -0.272635        NaN      NaN       NaN
## sma2 -0.726695        NaN      NaN       NaN
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## =============================================================
```

```
BC_sunspot_TS %>%
  check_residuals(order = c(6,1,8),
                  seasonal_order = c(1,1,2),
                  period = 11,
                  acf_lag = 70,
                  pacf_lag = 70,
                  res_main = 'Figure 26.1: Residuals of model:
                   ↪  SARIMA(6,1,8)(1,1,2)[11]',
                  hist_main = 'Figure 26.2: Histogram of residuals',
                  acf_main = 'Figure 26.3: ACF plot of residuals',
                  pacf_main = 'Figure 26.4: PACF plot of residuals')
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals
## W = 0.99088, p-value = 0.04211
```

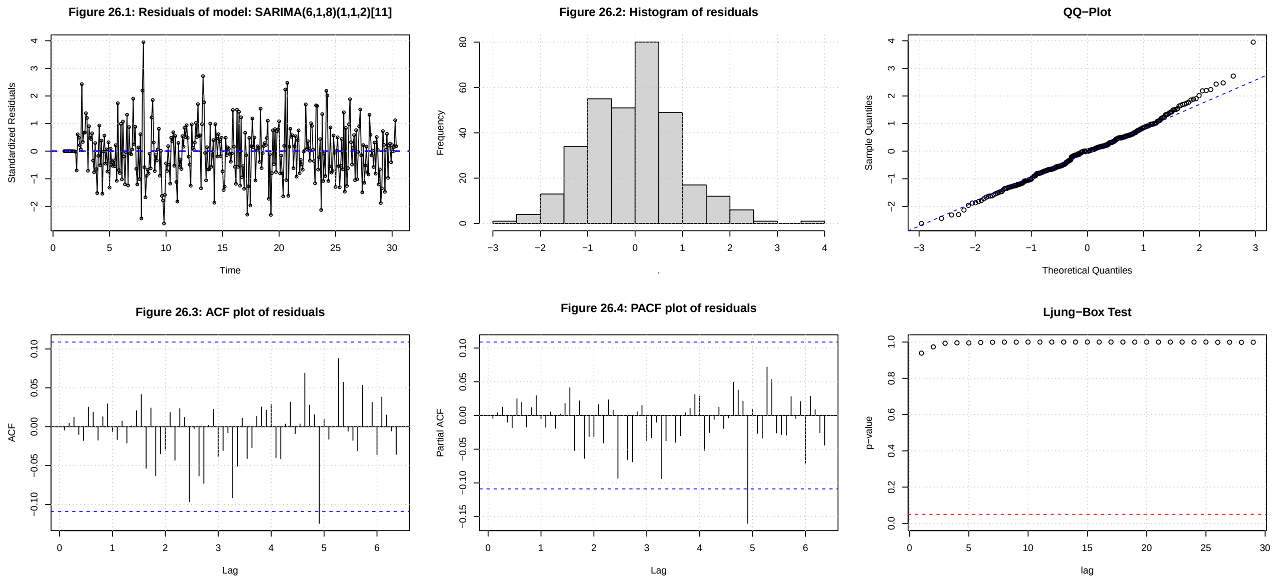


Figure 26.1: Residuals of model: SARIMA(6,1,8)(1,1,2)[11]

Figure 26.2: Histogram of residuals

QQ−Plot

Figure 26.3: ACF plot of residuals

Figure 26.4: PACF plot of residuals

Ljung−Box Test

At this stage of the analysis, the SARIMA(6,1,8)(1,1,2)[11] model demonstrates the best performance in terms of residual diagnostics. This model shows the highest normality of residuals, as indicated by visual assessments and the Shapiro-Wilk test. Only one significant lag appears in the ACF and PACF plots, which can be considered negligible. The Ljung-Box test supports this, with almost all p-values around 1.0, indicating that virtually no autocorrelation remains uncaptured. Thus, this model provides a promising fit, effectively capturing the underlying patterns in the series and ensuring minimal residual autocorrelation.

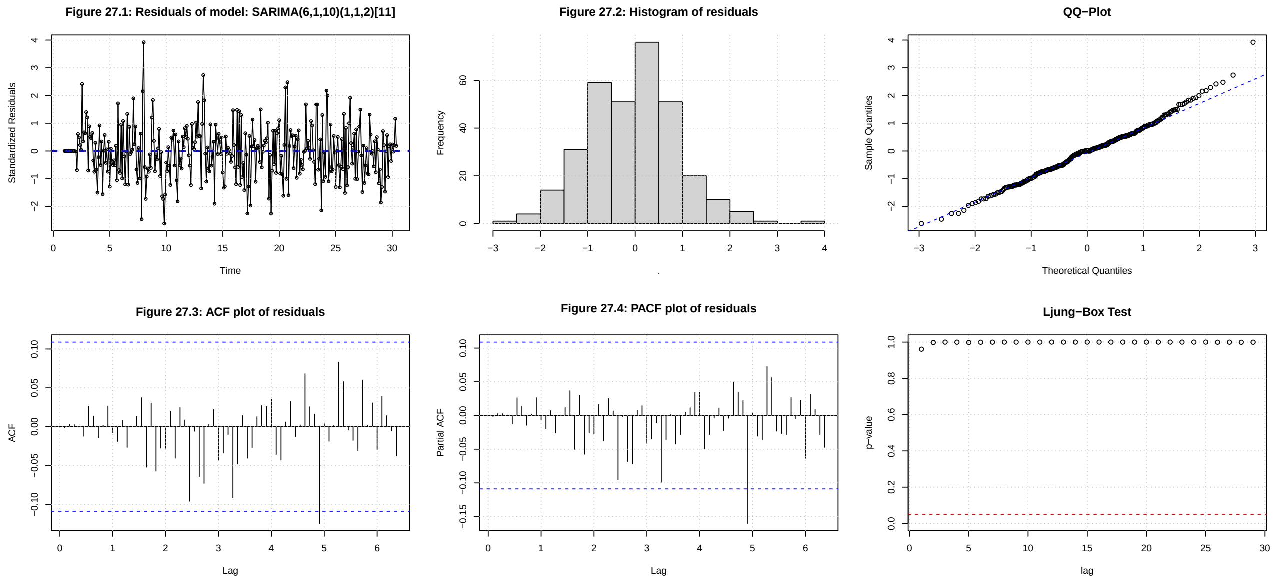
8.14 Model 14: SARIMA(6,1,10)(1,1,2)[11]

```
model_6110 <- BC_sunspot_TS %>% arima_summary(pdq_order = c(6,1,10),
                                               seasonal_order = c(1,1,2),
                                               period = 11)
```

```
##
## Parameter Estimation through Least Squares (CSS) Method for ARIMA( 6 1 10 )(1,1,2)[11]
##
## z test of coefficients:
##
##         Estimate Std. Error z value Pr(>|z|)
## ar1    0.2702839        NaN     NaN      NaN
## ar2    0.2272905        NaN     NaN      NaN
## ar3   -0.3872471        NaN     NaN      NaN
## ar4   -0.3703065        NaN     NaN      NaN
## ar5    0.2725003        NaN     NaN      NaN
## ar6   -0.2260282        NaN     NaN      NaN
## ma1   -0.0555229        NaN     NaN      NaN
## ma2   -0.4663280        NaN     NaN      NaN
## ma3   -0.0212385        NaN     NaN      NaN
## ma4    0.3966504        NaN     NaN      NaN
## ma5   -0.3870241        NaN     NaN      NaN
## ma6   -0.1396809        NaN     NaN      NaN
## ma7    0.0052367        NaN     NaN      NaN
## ma8   -0.0594151        NaN     NaN      NaN
## ma9    0.1394498        NaN     NaN      NaN
## ma10  0.2434638        NaN     NaN      NaN
## sar1 -0.6779063        NaN     NaN      NaN
## sma1 -0.1472663        NaN     NaN      NaN
## sma2 -0.6075604        NaN     NaN      NaN
##
## ==============================================================
## Parameter Estimation through Maximum Likelihood (ML) Method for ARIMA( 6 1 10 )(1,1,2)[11]
##
## z test of coefficients:
##
##       Estimate Std. Error z value  Pr(>|z|)
## ar1  -0.686832   0.264142 -2.6002 0.0093159 **
## ar2  -0.348420   0.260518 -1.3374 0.1810878
## ar3  -0.333901   0.190089 -1.7566 0.0789944 .
## ar4  -0.316820   0.179202 -1.7680 0.0770688 .
## ar5  -0.357118   0.203685 -1.7533 0.0795530 .
## ar6  -0.418137   0.111837 -3.7388 0.0001849 ***
## ma1   0.922453   0.266602  3.4600 0.0005401 ***
## ma2   0.325130   0.311344  1.0443 0.2963564
## ma3  -0.134093   0.199249 -0.6730 0.5009527
## ma4  -0.159886   0.131802 -1.2131 0.2251004
## ma5  -0.185560   0.157376 -1.1791 0.2383638
## ma6  -0.192337   0.233144 -0.8250 0.4093886
## ma7  -0.427897   0.259314 -1.6501 0.0989206 .
## ma8  -0.470561   0.239937 -1.9612 0.0498572 *
## ma9  -0.129054   0.201775 -0.6396 0.5224364
## ma10  0.316176   0.121587  2.6004 0.0093111 **
## sar1  0.508720   0.132403  3.8422 0.0001219 ***
## sma1 -1.062035   0.157455 -6.7450  1.53e-11 ***
## sma2  0.062103   0.149575  0.4152 0.6779966
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ==============================================================
## Parameter Estimation through Combination (CSS-ML) Method for ARIMA( 6 1 10 )(1,1,2)[11]
##
## z test of coefficients:
##
##        Estimate Std. Error z value  Pr(>|z|)
## ar1   1.0969183  0.1745777  6.2833 3.315e-10 ***
## ar2  -0.4752598  0.2699651 -1.7604 0.0783317 .
## ar3  -0.5132541  0.2363085 -2.1720 0.0298582 *
## ar4  -0.0074250  0.2173050 -0.0342 0.9727426
## ar5   0.5305650  0.2299834  2.3070 0.0210565 *
## ar6  -0.6223563  0.1249353 -4.9814 6.312e-07 ***
## ma1  -0.8743246  0.1868290 -4.6798 2.871e-06 ***
## ma2   0.0480161  0.2512914  0.1911 0.8484649
## ma3   0.4263896  0.1813156  2.3516 0.0186907 *
## ma4   0.3039326  0.2052968  1.4805 0.1387518
## ma5  -0.9160326  0.1713213 -5.3469 8.949e-08 ***
## ma6   0.2389781  0.1200526  1.9906 0.0465237 *
## ma7   0.3068026  0.1249755  2.4549 0.0140923 *
## ma8  -0.1429095  0.1345740 -1.0619 0.2882630
## ma9   0.0414880  0.1603839  0.2587 0.7958825
## ma10 -0.0033426  0.1115134 -0.0300 0.9760870
## sar1 -0.8825261  0.2315472 -3.8114 0.0001382 ***
## sma1 -0.0926021  0.2264021 -0.4090 0.6825277
## sma2 -0.9068632  0.2252325 -4.0263 5.665e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ==============================================================
```

```
BC_sunspot_TS %>%
  check_residuals(order = c(6,1,10),
                  seasonal_order = c(1,1,2),
                  period = 11,
                  acf_lag = 70,
                  pacf_lag = 70,
                  res_main = 'Figure 27.1: Residuals of model:
                  ↪  SARIMA(6,1,10)(1,1,2)[11]',
                  hist_main = 'Figure 27.2: Histogram of residuals',
                  acf_main = 'Figure 27.3: ACF plot of residuals',
                  pacf_main = 'Figure 27.4: PACF plot of residuals')
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals
## W = 0.99078, p-value = 0.03984
```



The SARIMA(6,1,10)(1,1,2)[11] model is one of the largest models fitted in the parameter estimation section of this report, featuring a total of 19 parameters. Of these, 12 parameters are significant. Interestingly, some of the significant terms are lower-order MA terms, while the higher-order terms remain insignificant.

In terms of residual normality, this model approximates a normal distribution quite closely, as suggested by the p-value from the Shapiro-Wilk test. The residual diagnostics further reinforce the model's adequacy, with the Ljung-Box test indicating almost no autocorrelation, as nearly all lags have high p-values. This suggests that the model effectively captures the series' underlying patterns, ensuring minimal residual autocorrelation and a distribution of residuals that closely aligns with normality.

8.15 Model 15: SARIMA(11,1,2)(1,1,2)[11]

```
model_1112 <- BC_sunspot_TS %>% arima_summary(pdq_order = c(11,1,2),
                                              seasonal_order = c(1,1,2),
                                              period = 11)
```

```
##
## Parameter Estimation through Least Squares (CSS) Method for ARIMA( 11 1 2 )(1,1,2)[11]
##
## z test of coefficients:
##
##       Estimate Std. Error  z value  Pr(>|z|)
## ar1  -1.032285   0.078828 -13.0954 < 2.2e-16 ***
## ar2  -0.630353   0.107127  -5.8842 4.000e-09 ***
## ar3  -0.447108   0.087237  -5.1252 2.972e-07 ***
## ar4  -0.681654   0.090463  -7.5351 4.878e-14 ***
## ar5  -0.655723   0.092962  -7.0537 1.742e-12 ***
## ar6  -0.712783   0.090493  -7.8767 3.361e-15 ***
## ar7  -0.590904   0.095956  -6.1581 7.364e-10 ***
## ar8  -0.582933   0.092020  -6.3348 2.376e-10 ***
## ar9  -0.373561   0.095882  -3.8961 9.777e-05 ***
## ar10 -0.080611   0.100152  -0.8049 0.4208840
## ar11 -0.018898   0.058909  -0.3208 0.7483604
## ma1   1.314641   0.065017  20.2200 < 2.2e-16 ***
## ma2   0.772394   0.094153   8.2036 2.333e-16 ***
## sar1 -0.445583   0.113907  -3.9118 9.161e-05 ***
## sma1 -0.403272   0.110656  -3.6444 0.0002681 ***
## sma2 -0.509648   0.107658  -4.7340 2.202e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ============================================================
## Parameter Estimation through Maximum Likelihood (ML) Method for ARIMA( 11 1 2 )(1,1,2)[11]
##
## z test of coefficients:
##
##       Estimate Std. Error z value  Pr(>|z|)
## ar1  -1.108806   0.164640 -6.7347 1.642e-11 ***
## ar2  -0.744590   0.146398 -5.0861 3.656e-07 ***
## ar3  -0.473601   0.095042 -4.9831 6.258e-07 ***
## ar4  -0.723782   0.139865 -5.1749 2.281e-07 ***
## ar5  -0.711299   0.120132 -5.9210 3.200e-09 ***
## ar6  -0.774178   0.114844 -6.7411 1.572e-11 ***
## ar7  -0.681345   0.142385 -4.7852 1.708e-06 ***
## ar8  -0.636133   0.126586 -5.0253 5.027e-07 ***
## ar9  -0.426827   0.148663 -2.8711   0.00409 **
## ar10 -0.170056   0.136636 -1.2446   0.21328
## ar11 -0.052356   0.070665 -0.7409   0.45875
## ma1   1.361836   0.154079  8.8386 < 2.2e-16 ***
## ma2   0.834434   0.151088  5.5228 3.336e-08 ***
## sar1  0.033658   0.765355  0.0440   0.96492
## sma1 -0.946436   0.733017 -1.2912   0.19665
## sma2 -0.053563   0.731661 -0.0732   0.94164
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ================================================================
## Parameter Estimation through Combination (CSS-ML) Method for ARIMA( 11 1 2 )(1,1,2)[11]
##
## z test of coefficients:
##
##       Estimate Std. Error z value  Pr(>|z|)
## ar1  -1.109554   0.163277 -6.7955 1.079e-11 ***
## ar2  -0.744888   0.145657 -5.1140 3.155e-07 ***
## ar3  -0.473508   0.095097 -4.9792 6.384e-07 ***
## ar4  -0.724155   0.139341 -5.1970 2.025e-07 ***
## ar5  -0.711371   0.119900 -5.9330 2.974e-09 ***
## ar6  -0.774491   0.114587 -6.7590 1.389e-11 ***
## ar7  -0.681880   0.141885 -4.8059 1.541e-06 ***
## ar8  -0.636502   0.126338 -5.0381 4.702e-07 ***
## ar9  -0.427686   0.148310 -2.8837   0.00393 **
## ar10 -0.170689   0.136372 -1.2516   0.21070
## ar11 -0.052280   0.070466 -0.7419   0.45814
## ma1   1.362774   0.152816  8.9177 < 2.2e-16 ***
## ma2   0.835228   0.149865  5.5732 2.501e-08 ***
## sar1  0.028797   0.772890  0.0373   0.97028
## sma1 -0.942349   0.740058 -1.2733   0.20290
## sma2 -0.057649   0.738549 -0.0781   0.93778
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ================================================================
```

```
BC_sunspot_TS %>%
  check_residuals(order = c(11,1,2),
                  seasonal_order = c(1,1,2),
                  period = 11,
                  acf_lag = 70,
                  pacf_lag = 70,
                  res_main = 'Figure 28.1: Residuals of model:
                   ↪  SARIMA(11,1,2)(1,1,2)[11]',
                  hist_main = 'Figure 28.2: Histogram of residuals',
                  acf_main = 'Figure 28.3: ACF plot of residuals',
                  pacf_main = 'Figure 28.4: PACF plot of residuals')
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals
## W = 0.99064, p-value = 0.03696
```

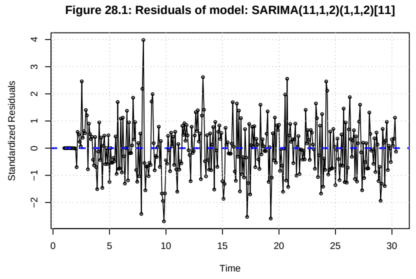


Figure 28.1: Residuals of model: SARIMA(11,1,2)(1,1,2)[11]

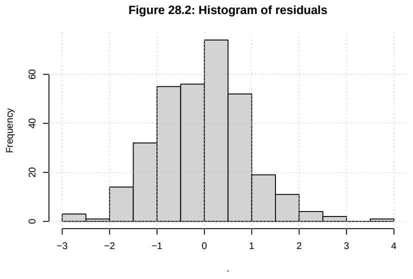
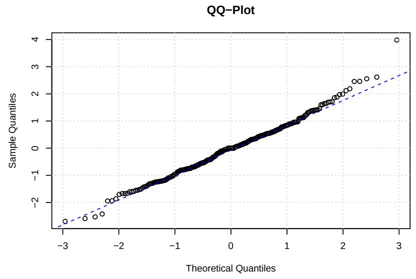


Figure 28.2: Histogram of residuals


QQ−Plot

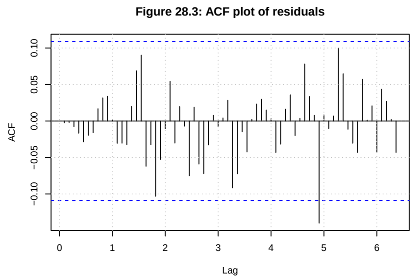


Figure 28.3: ACF plot of residuals

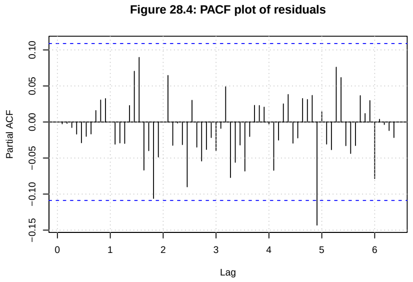
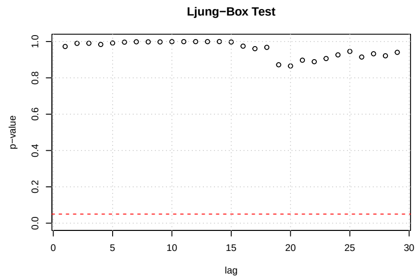


Figure 28.4: PACF plot of residuals


Ljung−Box Test

The SARIMA(11,1,2)(1,1,2)[11] model, like its predecessor, demonstrates a near close alignment to normality and captures almost all autocorrelation. Almost all ordinary AR and MA terms are significant, indicating their importance in modeling the series. However, the seasonal AR and MA terms are found to be insignificant. Despite this, the model effectively captures the underlying patterns in the data, with residual diagnostics showing minimal autocorrelation and a near-normal distribution, as evidenced by the Shapiro-Wilk test and the Ljung-Box test results. This suggests that while the seasonal terms may not be significant, the model still provides a robust fit to the series.

9 Model Selection

This segment will delve into comparing the AIC and BIC scores of the models. In comparing the AIC (Akaike Information Criterion) and BIC (Bayesian Information Criterion) scores of the models, we aim to discern the most suitable model for our dataset. Both criteria provide valuable insights into model selection, but they differ slightly in their approach.

Error measure scores provide a quantitative assessment of how well a model fits the data by evaluating the discrepancies between the observed values and the model's predictions.

Note: Any models not included in the parameter estimation section were excluded due to errors encountered during the estimation process. These errors prevented the models from identifying the optimal coefficients, making them unsuitable for further analysis.

9.1 AIC and BIC Scores

This section will look into analysis the AIC and BIC scores of the set of possible models. While AIC tends to select more complex models when sample sizes are large, BIC tends to favor simpler models. Therefore, comparing both AIC and BIC scores provides a comprehensive perspective on model performance, allowing us to strike a balance between goodness-of-fit and model simplicity.

```r
# AIC Scores
sort.score(AIC(model_917[["model_917_cssml"]], model_0110[["model_0110_cssml"]],
               model_0111[["model_0111_cssml"]], model_1111[["model_1111_cssml"]],
               model_1112[["model_1112_cssml"]], model_312[["model_312_cssml"]],
               model_315[["model_315_cssml"]], model_318[["model_318_cssml"]],
               model_3110[["model_3110_cssml"]], model_512[["model_512_cssml"]],
               model_515[["model_515_cssml"]], model_612[["model_612_cssml"]],
               model_615[["model_615_cssml"]], model_618[["model_618_cssml"]],
               model_6110[["model_6110_cssml"]]), score='aic')
```

```
##                                   df      AIC
## model_618[["model_618_cssml"]]    18 1496.676
## model_6110[["model_6110_cssml"]] 20 1500.068
## model_1111[["model_1111_cssml"]] 16 1505.344
## model_615[["model_615_cssml"]]    15 1508.458
## model_1112[["model_1112_cssml"]] 17 1508.684
## model_0111[["model_0111_cssml"]] 15 1508.754
## model_515[["model_515_cssml"]]    14 1508.871
## model_917[["model_917_cssml"]]    20 1509.665
## model_3110[["model_3110_cssml"]] 17 1509.678
## model_0110[["model_0110_cssml"]] 14 1512.128
## model_512[["model_512_cssml"]]    11 1518.347
## model_315[["model_315_cssml"]]    12 1518.966
## model_612[["model_612_cssml"]]    12 1520.358
## model_312[["model_312_cssml"]]     9 1523.858
## model_318[["model_318_cssml"]]    15 1525.862
```

```
# BIC Scores
sort.score(BIC(model_917[["model_917_cssml"]], model_0110[["model_0110_cssml"]],
               model_0111[["model_0111_cssml"]], model_1111[["model_1111_cssml"]],
               model_1112[["model_1112_cssml"]], model_312[["model_312_cssml"]],
               model_315[["model_315_cssml"]], model_318[["model_318_cssml"]],
               model_3110[["model_3110_cssml"]], model_512[["model_512_cssml"]],
               model_515[["model_515_cssml"]], model_612[["model_612_cssml"]],
               model_615[["model_615_cssml"]], model_618[["model_618_cssml"]],
               model_6110[["model_6110_cssml"]]), score='bic')
```

```
##                                   df      BIC
## model_312[["model_312_cssml"]]     9 1557.545
## model_512[["model_512_cssml"]]    11 1559.520
## model_515[["model_515_cssml"]]    14 1561.273
## model_315[["model_315_cssml"]]    12 1563.882
## model_618[["model_618_cssml"]]    18 1564.050
## model_0110[["model_0110_cssml"]] 14 1564.530
## model_615[["model_615_cssml"]]    15 1564.603
## model_0111[["model_0111_cssml"]] 15 1564.899
## model_1111[["model_1111_cssml"]] 16 1565.232
## model_612[["model_612_cssml"]]    12 1565.274
## model_1112[["model_1112_cssml"]] 17 1572.315
## model_3110[["model_3110_cssml"]] 17 1573.309
## model_6110[["model_6110_cssml"]] 20 1574.928
## model_318[["model_318_cssml"]]    15 1582.007
## model_917[["model_917_cssml"]]    20 1584.525
```

Based on the analysis of AIC and BIC scores, the ARIMA(5,1,5) model emerges as a potentially suitable choice for explaining the time series data. It achieves the third highest BIC score and ranks seventh highest in terms of AIC score among the considered models. Notably, the ARIMA(6,1,8) model also shows promise, securing the highest AIC score within the model set.

However, it ranks fifth in terms of BIC score, likely due to the BIC's stricter penalty for models with larger parameters. Despite this, the ARIMA(6,1,8) model still ranks among the top five models. It is also important to note that the actual BIC score of the ARIMA(6,1,8) model falls short of the other preceding top scores by a small margin.

Therefore, both the ARIMA(5,1,5) and ARIMA(6,1,8) models are considered likely optimal choices based on the AIC scores and the observed AR behavior in the series.

9.2 Error measure table

In this section, we will compare the error measures of the models. The selected measures include:

1. ME (Mean Error)
2. RMSE (Root Mean Squared Error)
3. MAE (Mean Absolute Error)
4. MPE (Mean Percentage Error)
5. MAPE (Mean Absolute Percentage Error)
6. MASE (Mean Absolute Scaled Error)

These error measures provide valuable insights into each model's performance, aiding in informed decisions regarding their suitability for our dataset. However, it's important to note that while error measures offer a glimpse into the accuracy of model predictions, they may not always be the best indicators of overall model performance.

Unlike AIC and BIC scores, which consider both model goodness-of-fit and complexity, error measures solely focus on the discrepancy between observed and predicted values. Consequently, a higher-order model with larger parameter values might yield lower error scores by fitting closely to the data. Yet, such models run the risk of overfitting due to their complexity, potentially compromising their generalization ability on unseen data. Thus, while error measures are valuable, they should be interpreted alongside other metrics like AIC and BIC to ensure a comprehensive evaluation of model performance.

```r
# Compute accuracy measures for each SARIMA model
Smodel_917_css <- accuracy(model_917[["model_917_cssml"]])[1:7]
Smodel_0110_css <- accuracy(model_0110[["model_0110_cssml"]])[1:7]
Smodel_0111_css <- accuracy(model_0111[["model_0111_cssml"]])[1:7]
Smodel_1111_css <- accuracy(model_1111[["model_1111_cssml"]])[1:7]
Smodel_1112_css <- accuracy(model_1112[["model_1112_cssml"]])[1:7]
Smodel_312_css <- accuracy(model_312[["model_312_cssml"]])[1:7]
Smodel_315_css <- accuracy(model_315[["model_315_cssml"]])[1:7]
Smodel_318_css <- accuracy(model_318[["model_318_cssml"]])[1:7]
Smodel_3110_css <- accuracy(model_3110[["model_3110_cssml"]])[1:7]
Smodel_512_css <- accuracy(model_512[["model_512_cssml"]])[1:7]
Smodel_515_css <- accuracy(model_515[["model_515_cssml"]])[1:7]
Smodel_612_css <- accuracy(model_612[["model_612_cssml"]])[1:7]
Smodel_615_css <- accuracy(model_615[["model_615_cssml"]])[1:7]
Smodel_618_css <- accuracy(model_618[["model_618_cssml"]])[1:7]
Smodel_6110_css <- accuracy(model_6110[["model_6110_cssml"]])[1:7]

# Combine accuracy measures into a dataframe
df.Smodels <- rbind(Smodel_917_css, Smodel_0110_css, Smodel_0111_css,
                    Smodel_1111_css, Smodel_1112_css, Smodel_312_css,
                    Smodel_315_css, Smodel_318_css, Smodel_3110_css,
                    Smodel_512_css, Smodel_515_css, Smodel_612_css,
                    Smodel_615_css, Smodel_618_css, Smodel_6110_css) %>%
↪  data.frame()
```

```r
# Assign column names to the dataframe
colnames(df.Smodels) <- c("ME", "RMSE", "MAE", "MPE", "MAPE",
                          "MASE", "ACF1")

# Assign row names indicating the SARIMA order of each model
rownames(df.Smodels) <- c("SARIMA(9,1,7)(1,1,2)[11]", "SARIMA(0,1,10)(1,1,2)[11]",
                          "SARIMA(0,1,11)(1,1,2)[11]", "SARIMA(1,1,11)(1,1,2)[11]",
                          "SARIMA(11,1,2)(1,1,2)[11]", "SARIMA(3,1,2)(1,1,2)[11]",
                          "SARIMA(3,1,5)(1,1,2)[11]", "SARIMA(3,1,8)(1,1,2)[11]",
                          "SARIMA(3,1,10)(1,1,2)[11]", "SARIMA(5,1,2)(1,1,2)[11]",
                          "SARIMA(5,1,5)(1,1,2)[11]", "SARIMA(6,1,2)(1,1,2)[11]",
                          "SARIMA(6,1,5)(1,1,2)[11]", "SARIMA(6,1,8)(1,1,2)[11]",
                          "SARIMA(6,1,10)(1,1,2)[11]")

# Round the values in the dataframe to five decimal places
round(df.Smodels,  digits = 5)
```

```
##                                 ME    RMSE     MAE      MPE     MAPE    MASE
## SARIMA(9,1,7)(1,1,2)[11]  -0.10271 2.34072 1.82086 -7.72255 22.47367 0.44214
## SARIMA(0,1,10)(1,1,2)[11] -0.02880 2.42050 1.86876 -5.51380 22.73224 0.45377
## SARIMA(0,1,11)(1,1,2)[11] -0.13575 2.34847 1.79141 -8.31460 22.77364 0.43498
## SARIMA(1,1,11)(1,1,2)[11] -0.09759 2.35330 1.80567 -7.56188 22.78833 0.43845
## SARIMA(11,1,2)(1,1,2)[11] -0.05754 2.38021 1.84502 -6.82190 22.65616 0.44800
## SARIMA(3,1,2)(1,1,2)[11]  -0.05268 2.50916 1.92968 -7.26319 23.77528 0.46856
## SARIMA(3,1,5)(1,1,2)[11]  -0.08329 2.46255 1.90768 -7.53403 24.14067 0.46322
## SARIMA(3,1,8)(1,1,2)[11]  -0.18056 2.44798 1.85123 -9.00474 24.00828 0.44951
## SARIMA(3,1,10)(1,1,2)[11] -0.03995 2.38263 1.82788 -6.21087 23.13347 0.44384
## SARIMA(5,1,2)(1,1,2)[11]  -0.07757 2.46568 1.88695 -7.63006 23.91869 0.45818
## SARIMA(5,1,5)(1,1,2)[11]  -0.05303 2.40087 1.87396 -6.38338 22.82414 0.45503
## SARIMA(6,1,2)(1,1,2)[11]  -0.10425 2.46273 1.89447 -7.79928 24.02040 0.46001
## SARIMA(6,1,5)(1,1,2)[11]  -0.07528 2.39283 1.86038 -7.09543 23.15822 0.45173
## SARIMA(6,1,8)(1,1,2)[11]  -0.06911 2.31272 1.78097 -6.49166 22.01693 0.43245
## SARIMA(6,1,10)(1,1,2)[11] -0.06661 2.30983 1.77566 -6.35237 21.91554 0.43116
##                               ACF1
## SARIMA(9,1,7)(1,1,2)[11]  -0.00446
## SARIMA(0,1,10)(1,1,2)[11] -0.01023
## SARIMA(0,1,11)(1,1,2)[11]  0.01033
## SARIMA(1,1,11)(1,1,2)[11] -0.00481
## SARIMA(11,1,2)(1,1,2)[11] -0.00262
## SARIMA(3,1,2)(1,1,2)[11]   0.00832
## SARIMA(3,1,5)(1,1,2)[11]  -0.00780
## SARIMA(3,1,8)(1,1,2)[11]  -0.01996
## SARIMA(3,1,10)(1,1,2)[11] -0.01678
## SARIMA(5,1,2)(1,1,2)[11]  -0.00166
## SARIMA(5,1,5)(1,1,2)[11]  -0.01088
## SARIMA(6,1,2)(1,1,2)[11]  -0.00863
## SARIMA(6,1,5)(1,1,2)[11]   0.00122
## SARIMA(6,1,8)(1,1,2)[11]  -0.00431
## SARIMA(6,1,10)(1,1,2)[11] -0.00150
```

The error measure table above reveals that the SARIMA(6,1,10)(1,1,2)[11] model consistently achieves top scores, indicating the lowest errors, across several metrics. Specifically, it outperforms other models in terms of RMSE, MAE, MPE, MASE, however given that the model is considerably bigger we do expect to see this behaviour.

The second best model comes out to be SARIMA(6,1,8)(1,1,2)[11] which follows the SARIMA(6,1,10)(1,1,2)[11] very closely bagging second least scores across the error metrics. The SARIMA(6,1,8)(1,1,2)[11] model also comes up as the top model in the AIC scores and fifth best in the BIC table. It is also important to note that even though BIC favors smaller models the SARIMA(6,1,8)(1,1,2)[11] comes in fifth with very marginal difference in the scores compared to smaller models.

Optimal Model: SARIMA(6,1,8)(1,1,2)[11]

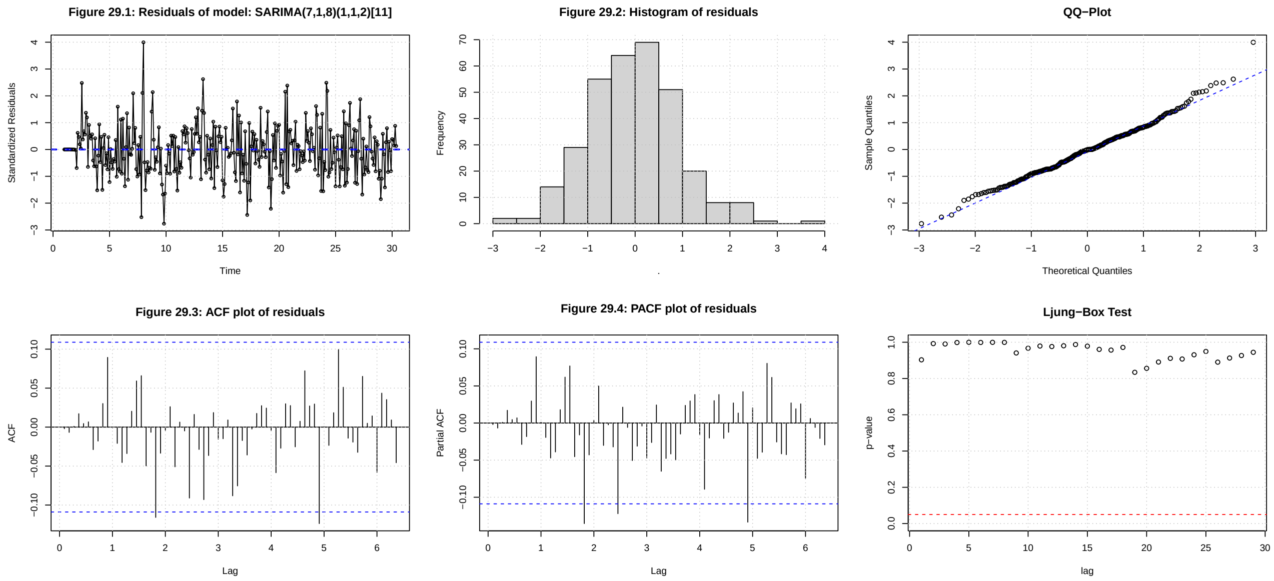
9.3 Neighbouring models

```
model_718 <- BC_sunspot_TS %>% arima_summary(pdq_order = c(7,1,8),
                                             seasonal_order = c(1,1,2),
                                             period = 11)
```

```
##
## Parameter Estimation through Least Squares (CSS) Method for ARIMA( 7 1 8 )(1,1,2)[11]
##
## z test of coefficients:
##
##        Estimate Std. Error  z value  Pr(>|z|)
## ar1  -0.0570920  0.0020425 -27.9527 < 2.2e-16 ***
## ar2  -0.2635238  0.0392418  -6.7154 1.876e-11 ***
## ar3   0.3787528  0.0217164  17.4409 < 2.2e-16 ***
## ar4  -0.2077594  0.0376682  -5.5155 3.478e-08 ***
## ar5  -0.2523145  0.0463243  -5.4467 5.131e-08 ***
## ar6  -0.6421226  0.0135401 -47.4236 < 2.2e-16 ***
## ar7   0.1194448  0.0242063   4.9344 8.038e-07 ***
## ma1   0.3169053  0.0238777  13.2720 < 2.2e-16 ***
## ma2   0.1435161        NaN      NaN       NaN
## ma3  -0.8154041        NaN      NaN       NaN
## ma4  -0.3002203  0.0178969 -16.7750 < 2.2e-16 ***
## ma5  -0.0476116  0.0400687  -1.1882 0.2347354
## ma6   0.6572182        NaN      NaN       NaN
## ma7   0.0334876  0.0336536   0.9951 0.3197030
## ma8  -0.1897575  0.0529823  -3.5815 0.0003416 ***
## sar1 -0.6177875  0.0381181 -16.2072 < 2.2e-16 ***
## sma1 -0.2191173  0.0576027  -3.8039 0.0001424 ***
## sma2 -0.6133780  0.0561446 -10.9250 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## =============================================================
## Parameter Estimation through Maximum Likelihood (ML) Method for ARIMA( 7 1 8 )(1,1,2)[11]
##
## z test of coefficients:
##
##       Estimate Std. Error z value  Pr(>|z|)
## ar1   0.733849   0.493439  1.4872 0.1369584
## ar2  -0.076782   0.524004 -0.1465 0.8835035
## ar3  -0.688131   0.249236 -2.7610 0.0057631 **
## ar4  -0.186655   0.272430 -0.6851 0.4932506
## ar5   0.509878   0.140513  3.6287 0.0002849 ***
## ar6  -0.411897   0.331362 -1.2430 0.2138526
## ar7  -0.239367   0.332426 -0.7201 0.4714876
## ma1  -0.515008   0.493412 -1.0438 0.2965927
## ma2  -0.265358   0.410149 -0.6470 0.5176451
## ma3   0.437244   0.131568  3.3233 0.0008895 ***
## ma4   0.454012   0.184384  2.4623 0.0138044 *
## ma5  -0.784035   0.233207 -3.3620 0.0007739 ***
## ma6  -0.103811   0.472175 -0.2199 0.8259819
## ma7   0.393854   0.143272  2.7490 0.0059777 **
## ma8  -0.026447   0.143253 -0.1846 0.8535282
## sar1  0.066871        NaN     NaN       NaN
## sma1 -1.060890        NaN     NaN       NaN
## sma2  0.060955        NaN     NaN       NaN
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## =================================================================
## Parameter Estimation through Combination (CSS-ML) Method for ARIMA( 7 1 8 )(1,1,2)[11]
##
## z test of coefficients:
##
##       Estimate Std. Error z value  Pr(>|z|)
## ar1  -0.065269   0.255562 -0.2554 0.7984191
## ar2  -0.242724   0.110442 -2.1978 0.0279660 *
## ar3   0.359455   0.135298  2.6568 0.0078893 **
## ar4  -0.179584   0.145453 -1.2347 0.2169592
## ar5  -0.375228   0.101546 -3.6951 0.0002198 ***
## ar6  -0.582007   0.089127 -6.5301 6.573e-11 ***
## ar7   0.115928   0.194334  0.5965 0.5508147
## ma1   0.316667   0.251470  1.2593 0.2079357
## ma2   0.082055   0.159618  0.5141 0.6072032
## ma3  -0.780782   0.125622 -6.2153 5.122e-10 ***
## ma4  -0.239503   0.229055 -1.0456 0.2957394
## ma5   0.079866   0.143163  0.5579 0.5769336
## ma6   0.550941   0.117720  4.6801 2.867e-06 ***
## ma7  -0.019300   0.170313 -0.1133 0.9097758
## ma8  -0.257246   0.083250 -3.0900 0.0020013 **
## sar1  0.128740   0.512190  0.2514 0.8015428
## sma1 -1.016369   0.507856 -2.0013 0.0453607 *
## sma2  0.016420   0.505387  0.0325 0.9740807
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## =================================================================
```

```
BC_sunspot_TS %>%
  check_residuals(order = c(7,1,8),
                  seasonal_order = c(1,1,2),
                  period = 11,
                  acf_lag = 70,
                  pacf_lag = 70,
                  res_main = 'Figure 29.1: Residuals of model:
                  ↪  SARIMA(7,1,8)(1,1,2)[11]',
                  hist_main = 'Figure 29.2: Histogram of residuals',
                  acf_main = 'Figure 29.3: ACF plot of residuals',
                  pacf_main = 'Figure 29.4: PACF plot of residuals')
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals
## W = 0.98992, p-value = 0.02496
```

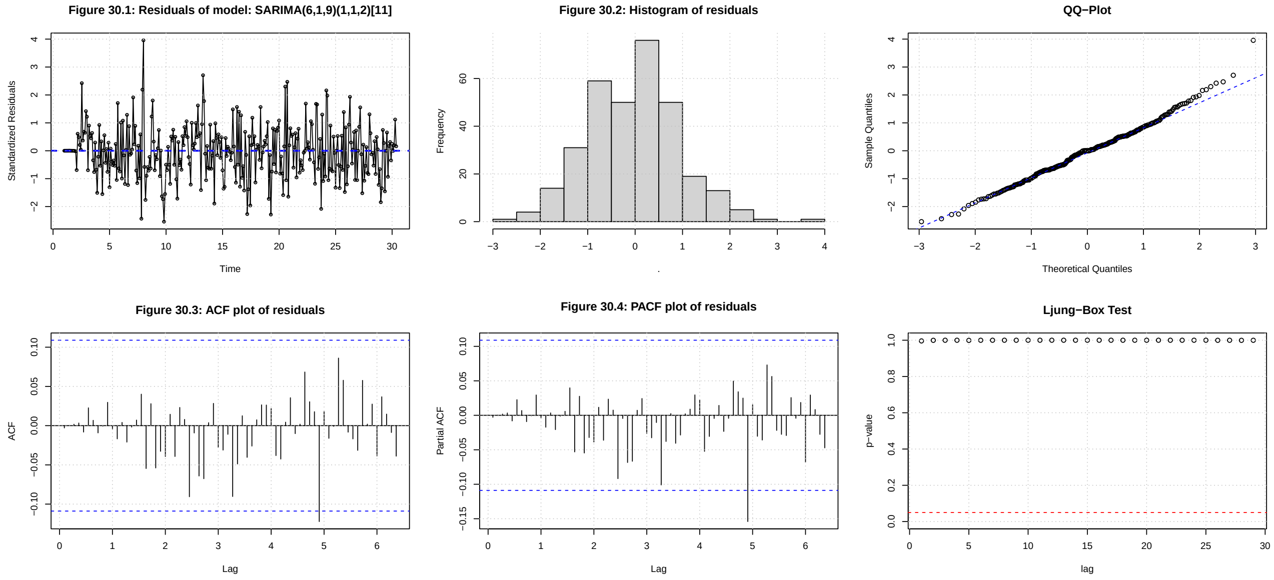


```
model_619 <- BC_sunspot_TS %>% arima_summary(pdq_order = c(6,1,9),
                                             seasonal_order = c(1,1,2),
                                             period = 11)
```

```
##
## Parameter Estimation through Least Squares (CSS) Method for ARIMA( 6 1 9 )(1,1,2)[11]
##
## z test of coefficients:
##
##        Estimate Std. Error z value  Pr(>|z|)
## ar1    1.361564   0.312862  4.3520 1.349e-05 ***
## ar2   -0.758121   0.473662 -1.6006 0.1094759
## ar3   -0.674205   0.303214 -2.2235 0.0261804 *
## ar4    0.947794        NaN     NaN       NaN
## ar5   -0.514036   0.135599 -3.7908 0.0001501 ***
## ar6   -0.095834   0.082425 -1.1627 0.2449605
## ma1   -1.168375   0.288783 -4.0459 5.213e-05 ***
## ma2    0.250386   0.329904  0.7590 0.4478729
## ma3    0.767954        NaN     NaN       NaN
## ma4   -0.768639   0.197820 -3.8856 0.0001021 ***
## ma5    0.076431   0.233443  0.3274 0.7433585
## ma6  -0.044912   0.194053 -0.2314 0.8169712
## ma7   0.356346   0.179758  1.9824 0.0474384 *
## ma8  -0.374549   0.123608 -3.0301 0.0024444 **
## ma9   0.263377        NaN     NaN       NaN
## sar1 -0.520551   0.183599 -2.8353 0.0045787 **
## sma1 -0.478715   0.214454 -2.2322 0.0255985 *
## sma2 -0.347290   0.204245 -1.7004 0.0890629 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ================================================================
## Parameter Estimation through Maximum Likelihood (ML) Method for ARIMA( 6 1 9 )(1,1,2)[11]
##
## z test of coefficients:
##
##       Estimate Std. Error z value  Pr(>|z|)
## ar1   1.101990   0.112379  9.8060 < 2.2e-16 ***
## ar2  -0.479052   0.179196 -2.6733  0.007510 **
## ar3  -0.527294   0.191212 -2.7576  0.005822 **
## ar4   0.030035   0.246325  0.1219  0.902952
## ar5   0.498678   0.218366  2.2837  0.022390 *
## ar6  -0.607704   0.109273 -5.5614 2.677e-08 ***
## ma1  -0.881972   0.125498 -7.0278 2.099e-12 ***
## ma2   0.056125   0.172461  0.3254  0.744850
## ma3   0.434163   0.166947  2.6006  0.009306 **
## ma4   0.271618   0.210960  1.2875  0.197907
## ma5  -0.893043   0.145780 -6.1259 9.015e-10 ***
## ma6   0.236451   0.111134  2.1276  0.033369 *
## ma7   0.311742   0.110972  2.8092  0.004966 **
## ma8  -0.156438   0.121496 -1.2876  0.197885
## ma9   0.055090   0.100356  0.5490  0.583038
## sar1 -0.192370   2.686702 -0.0716  0.942920
## sma1 -0.790622   2.762396 -0.2862  0.774718
## sma2 -0.209147   2.759581 -0.0758  0.939587
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ================================================================
## Parameter Estimation through Combination (CSS-ML) Method for ARIMA( 6 1 9 )(1,1,2)[11]
##
## z test of coefficients:
##
##       Estimate Std. Error z value  Pr(>|z|)
## ar1   1.101990   0.112379  9.8060 < 2.2e-16 ***
## ar2  -0.479052   0.179196 -2.6733  0.007510 **
## ar3  -0.527294   0.191212 -2.7576  0.005822 **
## ar4   0.030035   0.246325  0.1219  0.902952
## ar5   0.498678   0.218366  2.2837  0.022390 *
## ar6  -0.607704   0.109273 -5.5614 2.677e-08 ***
## ma1  -0.881972   0.125498 -7.0278 2.099e-12 ***
## ma2   0.056125   0.172461  0.3254  0.744850
## ma3   0.434163   0.166947  2.6006  0.009306 **
## ma4    0.271618   0.210960  1.2875  0.197907
## ma5   -0.893043   0.145780 -6.1259 9.015e-10 ***
## ma6    0.236451   0.111134  2.1276  0.033369 *
## ma7    0.311742   0.110972  2.8092  0.004966 **
## ma8   -0.156438   0.121496 -1.2876  0.197885
## ma9    0.055090   0.100356  0.5490  0.583038
## sar1 -0.192370   2.686702 -0.0716  0.942920
## sma1 -0.790622   2.762396 -0.2862  0.774718
## sma2 -0.209147   2.759581 -0.0758  0.939587
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ==============================================================
```

```r
BC_sunspot_TS %>%
  check_residuals(order = c(6,1,9),
                  seasonal_order = c(1,1,2),
                  period = 11,
                  acf_lag = 70,
                  pacf_lag = 70,
                  res_main = 'Figure 30.1: Residuals of model:
                  ↪  SARIMA(6,1,9)(1,1,2)[11]',
                  hist_main = 'Figure 30.2: Histogram of residuals',
                  acf_main = 'Figure 30.3: ACF plot of residuals',
                  pacf_main = 'Figure 30.4: PACF plot of residuals')
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals
## W = 0.99108, p-value = 0.04702
```



```r
# Compute AIC Scores for neighboring models
sort.score(AIC(model_718[["model_718_cssml"]], model_619[["model_619_cssml"]]),
↪  score = 'aic')
```

```
##                                  df      AIC
## model_619[["model_619_cssml"]] 19 1498.573
## model_718[["model_718_cssml"]] 19 1512.682
```

```
# Compute BIC Scores for neighboring models
sort.score(BIC(model_718[["model_718_cssml"]], model_619[["model_619_cssml"]]),
↪  score = 'bic')
```

```
##                                  df      BIC
## model_619[["model_619_cssml"]] 19 1569.690
## model_718[["model_718_cssml"]] 19 1583.799
```

```
# Compute accuracy measures for each neighboring models
Smodel_718_css <- accuracy(model_718[["model_718_cssml"]])[1:7]
Smodel_619_css <- accuracy(model_619[["model_619_cssml"]])[1:7]

df.Smodels.neighbors <- rbind(Smodel_718_css, Smodel_619_css) %>% data.frame()

# Assign column names to the dataframe
colnames(df.Smodels.neighbors) <- c("ME", "RMSE", "MAE", "MPE", "MAPE",
                                    "MASE", "ACF1")



# Assign row names indicating the SARIMA order of each model
rownames(df.Smodels.neighbors) <- c("SARIMA(7,1,8)(1,1,2)[11]",
↪  "SARIMA(6,1,9(1,1,2)[11]")

# Round the values in the dataframe to five decimal places
round(df.Smodels.neighbors,  digits = 5)
```

```
##                                 ME    RMSE     MAE      MPE     MAPE    MASE
## SARIMA(7,1,8)(1,1,2)[11] -0.07018 2.37375 1.85039 -6.79022 22.47302 0.44931
## SARIMA(6,1,9(1,1,2)[11]  -0.06476 2.31353 1.78561 -6.40040 21.94323 0.43358
##                               ACF1
## SARIMA(7,1,8)(1,1,2)[11] -0.00204
## SARIMA(6,1,9(1,1,2)[11]  -0.00288
```

The over-parameterized models fail to demonstrate significant enhancements when compared to the SARIMA(6,1,8)(1,1,2)[11] model in terms of AIC, BIC, and various error metrics. Notably, the only exception is the SARIMA(6,1,9)(1,1,2)[11], which shows an improvement in the MAPE score. Despite this, the overall lack of substantial improvement across other evaluation metrics suggests that the increased complexity of the over-parameterized models does not yield meaningful benefits. Consequently, we designate the SARIMA(6,1,8)(1,1,2)[11] model as the optimal choice, balancing model performance with complexity effectively.

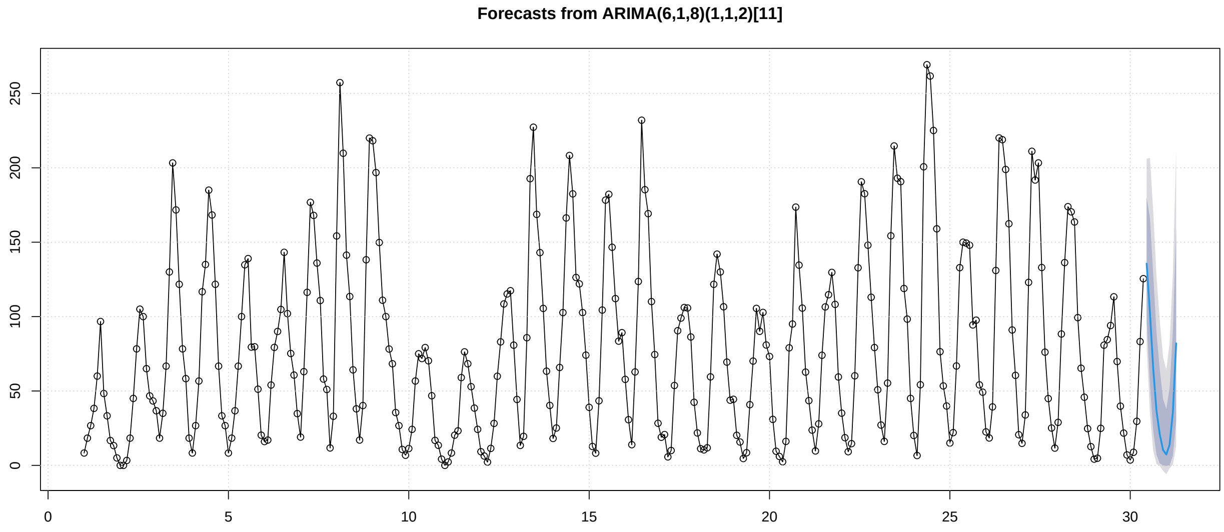
10 Forecast

```
# Store optimal model
optimal_model <- sunspot_TS %>% Arima(order=c(6,1,8),
                                      seasonal=list(order=c(1,1,2), period=11),
                                      method = "CSS", lambda = 0.49)


# Predict for next 10 units of time (2024 - 2054)
```

```r
predictions <- forecast(optimal_model, h = 10)

# Print Predictions
predictions_df <- predictions %>% as.data.frame()
row.names(predictions_df) <- 2024:2033
predictions_df
```

```
##      Point Forecast        Lo 80     Hi 80      Lo 95     Hi 95
## 2024     135.622607 97.64038988 179.97541 80.0986062 206.04574
## 2025     102.759866 54.52924816 166.53731 35.2257223 206.64565
## 2026      65.155511 23.36854168 128.51886  9.8373409 170.90164
## 2027      36.683312  7.25338658  89.49397  0.8916040 127.06990
## 2028      20.820511  1.33565734  64.45230 -0.3299959  97.53543
## 2029      10.395770 -0.01686734  44.67216 -3.6039182  72.78721
## 2030       7.287575 -0.40483713  37.87961 -5.8829644  64.02879
## 2031      13.972900  0.11762360  51.99051 -1.9623612  82.15302
## 2032      35.851367  6.41777111  89.96088  0.5481517 128.76314
## 2033      82.013297 30.25111873 159.79601 13.2132312 211.63029
```

```r
# Plot Forecast
plot(predictions, type='o')
grid(nx = NULL, ny = NULL, col = "gray", lty = "dotted")
```



11 Conclusion

The primary objective of this report was to develop a reliable model for forecasting yearly mean sunspot numbers from 1700 to 2023. Through a comprehensive analysis that included exploratory data analysis, stationarity and normality tests, data transformation, and thorough model specification, we aimed to capture the underlying patterns and trends in the sunspot data effectively.

Our initial exploratory data analysis revealed significant seasonality in the sunspot series, corresponding to the well-known 11-year solar cycle. This necessitated the use of seasonal models, specifically the Seasonal AutoRegressive Integrated Moving Average (SARIMA) models. After identifying the seasonal orders, we focused on determining the optimal non-seasonal AR and MA orders through tools such as ACF and PACF plots, EACF tables, and BIC tables.

We evaluated a set of 15 candidate SARIMA models, estimating their parameters and assessing their goodness-of-fit through various criteria, including AIC, BIC, and error measures (ME, RMSE, MAE, MPE, MAPE, MASE). The analysis revealed that higher-order models like SARIMA(6,1,8)(1,1,2)[11] and SARIMA(6,1,10)(1,1,2)[11] performed exceptionally well across multiple metrics, capturing the series' autocorrelation structure and exhibiting near-normal residuals.

However, given the complexity and over-parameterization of some models, we selected the SARIMA(6,1,8)(1,1,2)[11] model as the optimal choice. This model strikes a balance between performance and simplicity, providing robust forecasts while avoiding unnecessary complexity. It demonstrated the highest AIC score and ranked fifth in BIC score, with a marginal difference from simpler models.

The over-parameterized neighboring models, such as SARIMA(6,1,9)(1,1,2)[11], did not show significant improvements over the SARIMA(6,1,8)(1,1,2)[11] model across most evaluation metrics, further justifying our selection.

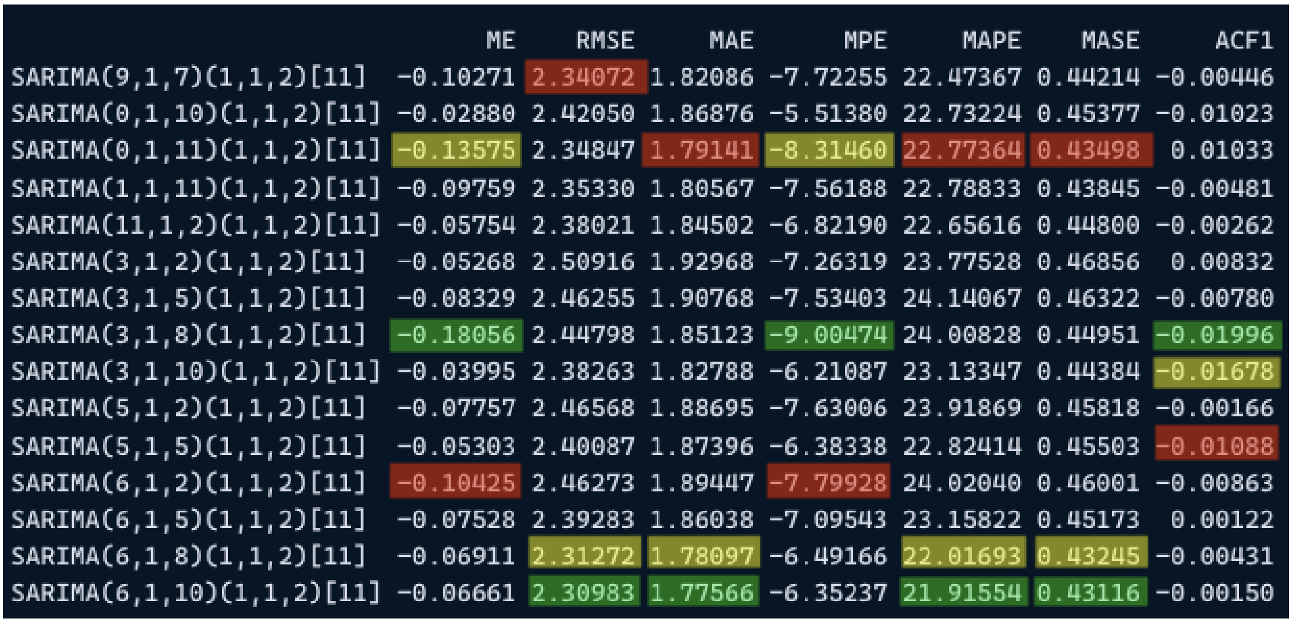
In conclusion, the SARIMA(6,1,8)(1,1,2)[11] model was chosen as the optimal model for forecasting future sunspot numbers. This model effectively captures the seasonal and non-seasonal patterns in the data, providing accurate and reliable forecasts. The resulting model was used to forecast sunspot numbers for the next 10 years, from 2024 to 2033, offering valuable insights for understanding solar cycles and their potential implications for space weather and climate.

12 Appendix

This appendix provides supplementary information to enhance understanding of the analysis and its technical aspects. This section hosts the code from the utilities.R script, which houses functions employed for this analysis. Additionally, it offers further details on other relevant supplementary materials.

12.1 Comparison of Error measure scores (Visual Aid)

In the figure below, areas highlighted in green denote the lowest scores on the error measures, while those highlighted in yellow signify the second lowest scoresand the red color signifies the third lowest.



```
                                    ME    RMSE     MAE      MPE     MAPE    MASE     ACF1
SARIMA(9,1,7)(1,1,2)[11]  -0.10271 2.34072 1.82086 -7.72255 22.47367 0.44214 -0.00446
SARIMA(0,1,10)(1,1,2)[11] -0.02880 2.42050 1.86876 -5.51380 22.73224 0.45377 -0.01023
SARIMA(0,1,11)(1,1,2)[11] -0.13575 2.34847 1.79141 -8.31460 22.77364 0.43498  0.01033
SARIMA(1,1,11)(1,1,2)[11] -0.09759 2.35330 1.80567 -7.56188 22.78833 0.43845 -0.00481
SARIMA(11,1,2)(1,1,2)[11] -0.05754 2.38021 1.84502 -6.82190 22.65616 0.44800 -0.00262
SARIMA(3,1,2)(1,1,2)[11]  -0.05268 2.50916 1.92968 -7.26319 23.77528 0.46856  0.00832
SARIMA(3,1,5)(1,1,2)[11]  -0.08329 2.46255 1.90768 -7.53403 24.14067 0.46322 -0.00780
SARIMA(3,1,8)(1,1,2)[11]  -0.18056 2.44798 1.85123 -9.00474 24.00828 0.44951 -0.01996
SARIMA(3,1,10)(1,1,2)[11] -0.03995 2.38263 1.82788 -6.21087 23.13347 0.44384 -0.01678
SARIMA(5,1,2)(1,1,2)[11]  -0.07757 2.46568 1.88695 -7.63006 23.91869 0.45818 -0.00166
SARIMA(5,1,5)(1,1,2)[11]  -0.05303 2.40087 1.87396 -6.38338 22.82414 0.45503 -0.01088
SARIMA(6,1,2)(1,1,2)[11]  -0.10425 2.46273 1.89447 -7.79928 24.02040 0.46001 -0.00863
SARIMA(6,1,5)(1,1,2)[11]  -0.07528 2.39283 1.86038 -7.09543 23.15822 0.45173  0.00122
SARIMA(6,1,8)(1,1,2)[11]  -0.06911 2.31272 1.78097 -6.49166 22.01693 0.43245 -0.00431
SARIMA(6,1,10)(1,1,2)[11] -0.06661 2.30983 1.77566 -6.35237 21.91554 0.43116 -0.00150
```

12.2 Utility Function - `sort.score()`

```
# Function to sort AIC and BIC Scores
# Example: sort.score(AIC(model1, model2), score = 'aic')
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")')
  }
}
```

The `sort.score()` function facilitates the sorting of models based on either the Akaike Information Criterion (AIC) or the Bayesian Information Criterion (BIC) scores. This function is particularly useful for comparing and selecting models based on their AIC or BIC scores, aiding in the model selection process by identifying the models with the best goodness-of-fit and balance of complexity.

12.3 Utility Function - `arima_summary()`

```
# Fitted ARIMA Models Summary generator
# Example: arima_summary(Data, c(1,1,1))
```

```
arima_summary <- function(TS_data, pdq_order, seasonal_order, period){

  # Check if TS_data is of class 'ts'
  if (!inherits(TS_data, "ts")) {
    stop("TS_data must be of class 'ts'")
  }

  # Estimate ARIMA model using CSS method
  model_css <- TS_data %>% Arima(order = pdq_order,
                                 seasonal = list(order=seasonal_order, period =
                                  ↪  period),
                                 method = 'CSS')

  # Print parameter estimation results for CSS method
  cat("\nParameter Estimation through Least Squares (CSS) Method for ARIMA(",
      pdq_order, ")(1,1,2)[11]\n")
  print(lmtest::coeftest(model_css))
  cat("===============================================================")

  # Estimate ARIMA model using ML method
  model_ml <- TS_data %>% Arima(order = pdq_order,
                                seasonal = list(order=seasonal_order, period =
                                 ↪  period),
                                method = 'ML')

  # Print parameter estimation results for ML method
  cat("\nParameter Estimation through Maximum Likelihood (ML) Method for ARIMA(",
      pdq_order, ")(1,1,2)[11]\n")
  print(lmtest::coeftest(model_ml))
  cat("===============================================================")

  # Estimate ARIMA model using CSS-ML method
  model_cssml <- TS_data %>% Arima(order = pdq_order,
                                   seasonal = list(order=seasonal_order, period =
                                    ↪  period),
                                   method = 'CSS-ML')

  # Print parameter estimation results for CSS-ML method
  cat("\nParameter Estimation through Combination (CSS-ML) Method for ARIMA(",
      pdq_order, ")(1,1,2)[11]\n")
  print(lmtest::coeftest(model_cssml))
  cat("===============================================================")

  # Store the models in a list
  ret_list <- list(model_css, model_ml, model_cssml)

  # Create names for the list elements based on pdq_order
  label <- paste0(pdq_order, collapse = "")
  names(ret_list) <- c(paste0("model_", label, "_css"),
                       paste0("model_", label, "_ml"),
                       paste0("model_", label, "_cssml"))
  # Return the list of models
  return(ret_list)
}
```

The `arima_summary()` function generates parameter estimation results for ARIMA/SARIMA models using three different estimation methods: Least Squares (CSS), Maximum Likelihood (ML), and a combination of both (CSS-ML). It takes time series data (`TS_data`) and the desired ARIMA order (`pdq_order`), seasonal order (`seasonal_order`) and period (`period`) as input.

The function first checks if the input data is of class ‘ts’, then estimates ARIMA models using CSS, ML, and CSS-ML methods. It prints the parameter estimation results for each method and stores the models in a list with appropriate names based on the ARIMA order. Finally, it returns the list of models for further analysis.

This function facilitates the comparison of parameter estimation results obtained through different estimation techniques, providing valuable insights into the stability and reliability of the fitted ARIMA models.

12.4 Utility Function - `plot_acf_pacf()`

```
# Function for Plotting ACF and PACF Plots
plot_acf_pacf <- function(TS_data, acf_main = 'ACF Plot', pacf_main = 'PACF Plot',
↪  max_lag = 30) {
  # Set up the plotting area to have 2 plots side by side
  par(mfrow = c(1, 2))

  # Plot ACF
  acf(TS_data, lag.max = max_lag, main = acf_main, xlab = "Lag", ylab = "ACF")
  grid(nx = NULL, ny = NULL, col = "gray", lty = "dotted")

  # Plot PACF
  pacf(TS_data, lag.max = max_lag, main = pacf_main, xlab = "Lag", ylab = "PACF")
  grid(nx = NULL, ny = NULL, col = "gray", lty = "dotted")

  # Reset plotting area to default
  par(mfrow = c(1, 1))
}
```

This function is utilized in the report to visualize the ACF and PACF of the time series data, providing insights into the autocorrelation and partial autocorrelation structures. These plots are essential for identifying potential ARIMA model parameters by revealing significant lags that indicate autoregressive (AR) and moving average (MA) components in the data.

12.5 Utility Function - `check_residuals()`

```
# Plotting residual diagnostic plots - TS plot, ACF, PACF, Histogram, QQPlot, Ljung
↪  Test Plot
check_residuals <- function(TS_data, order, seasonal_order, period,
                            acf_lag=60,
                            pacf_lag=60,
                            res_main='Residuals',
                            hist_main='Histogram',
                            acf_main='ACF',
                            pacf_main='PACF',
                            qq_main='QQ-Plot'){

  # Set up the plotting area to have 2 plots side by side
  par(mfrow = c(2, 3))

  library(FitAR)
  # Obtain Residuals
  residuals <- TS_data %>%
    Arima(order = order,
          seasonal = list(order=seasonal_order, period = period)) %>%
    rstandard()

  print(shapiro.test(residuals))

  # Plot Residuals
  residuals %>% plot(type = 'o',
                     main = res_main,
                     ylab = 'Standardized Residuals',
                     lwd=1)
  abline(h=0, lty=2, col='blue', lwd=2)

  grid(nx = NULL, ny = NULL, col = "gray", lty = "dotted")

  # Plot Histogram
  residuals %>% hist(main = hist_main)
  grid(nx = NULL, ny = NULL, col = "gray", lty = "dotted")

  # Plot QQ-Plot
  residuals %>% qqnorm(main = qq_main)
  residuals %>% qqline(lty=2, col='blue')
  grid(nx = NULL, ny = NULL, col = "gray", lty = "dotted")

  # Plot ACF
  residuals %>% acf(main=acf_main, lag.max = acf_lag)
  grid(nx = NULL, ny = NULL, col = "gray", lty = "dotted")

  # Plot PACF
  residuals %>% pacf(main=pacf_main, lag.max = pacf_lag)
  grid(nx = NULL, ny = NULL, col = "gray", lty = "dotted")

  # LJung-Box Test
  k=0
  if (length(residuals) < 30){
    lagM <- length(residuals) - 1
  } else {
    lagM <- 29
  }
  LBQPlot(residuals, lag.max = lagM, StartLag = k + 1, k = 0, SquaredQ = FALSE)
  grid(nx = NULL, ny = NULL, col = "gray", lty = "dotted")
```

```r

  # Reset plotting area to default
  par(mfrow = c(1, 1))
}
```

The check_residuals function generates a comprehensive set of diagnostic plots for analyzing the residuals of a fitted ARIMA/SARIMA model. It fits an ARIMA model to the provided time series data, computes standardized residuals, and prints the Shapiro-Wilk test result for normality of residuals.

The function then creates six plots in a 2x3 grid layout: a residual plot over time, a histogram of residuals, a QQ plot to assess normality, an autocorrelation function (ACF) plot, a partial autocorrelation function (PACF) plot, and a Ljung-Box test plot to check for remaining autocorrelation. Finally, it resets the plotting area configuration to the default setting. This function is essential for validating ARIMA/SARIMA model assumptions by examining residuals for any patterns or anomalies.