Question-2

2024-04-22

Aim: Our main goal is to analyze the time series data of monthly average house prices in the East Midlands region from January 2010 to December 2019, and forecast the prices for the first six months of 2020.

Data Analysis: We'll load the dataset and convert it into a time series object. We will then visualize the data, check for stationarity, seasonal patterns, and trends, and select the best fitting time series model based on these characteristics.

Model Fit Check: We'll use diagnostic tools to check the adequacy of our model. This includes looking at residual plots, ACF and PACF plots, and statistical tests to ensure our residuals are behaving as expected (random noise).

Forecast: Finally, we'll use the selected model to forecast the average house prices for January-June 2020 and provide a measure of uncertainty for these predictions.

The dataset "em_house_prices.csv" consists of 120 monthly observations of mean house sale prices in the East Midlands from January 2010 to December 2019. Each entry includes the month, year, and average house price in GBP.

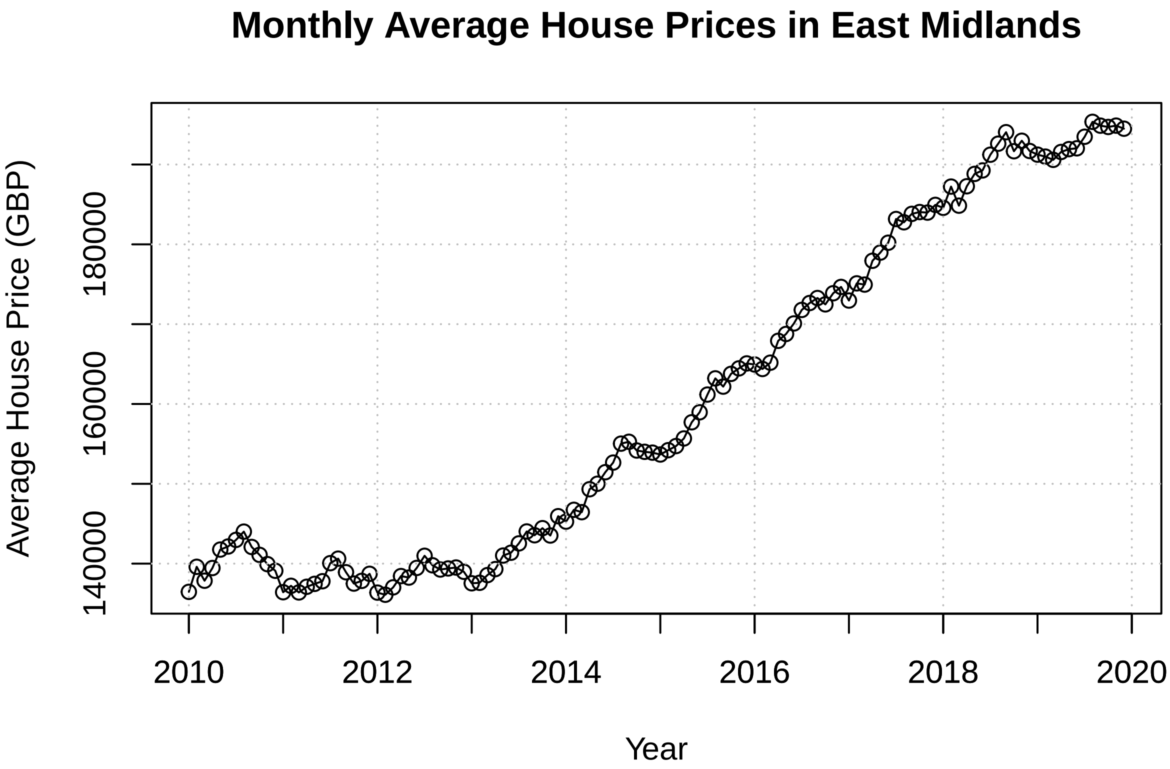
Summary: The dataset spans 10 years without any missing entries, indicating a complete set of monthly data for the period. The average_price_gbp column represents the mean house sale prices, ranging from £136,102 to £195,345 with a mean value of approximately £160,248.

```r
# Reading the dataset.
house_price_ts <- read.csv("em_house_prices.csv")

# Creating a Time Series Object
house_price_ts <- ts(house_price_ts$average_price_gbp, start=2010, frequency=12)
```

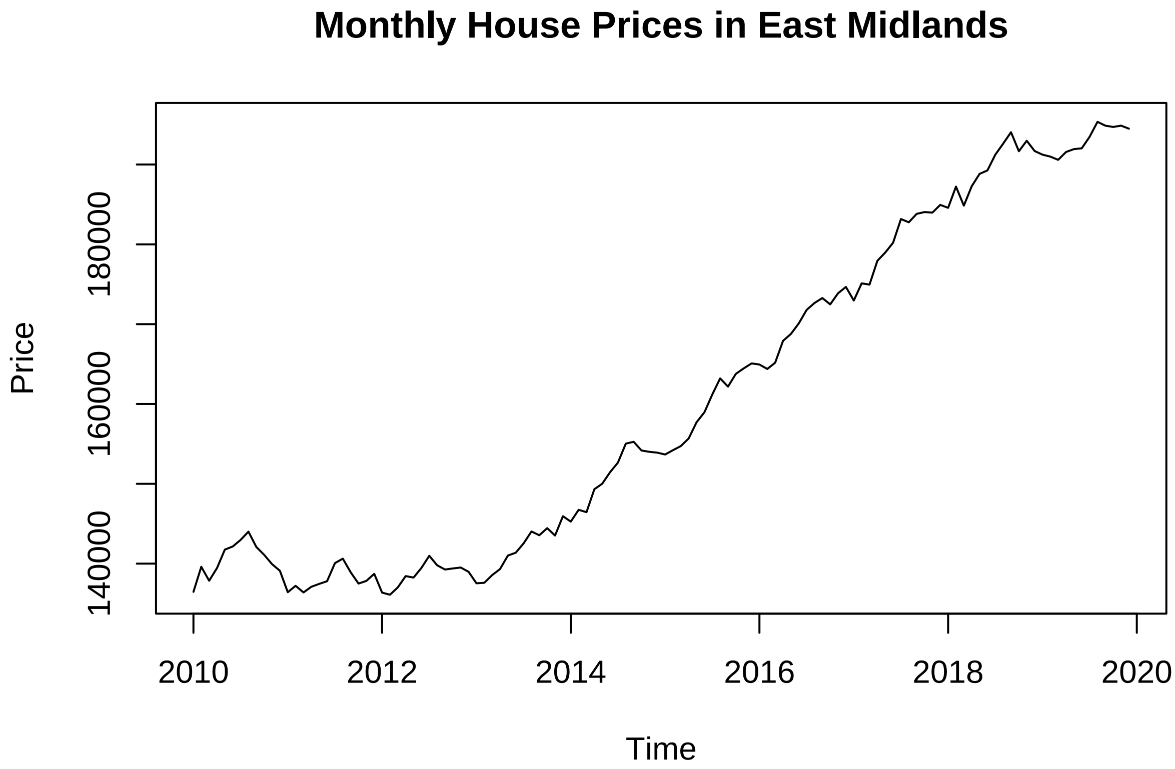
```r
head(house_price_ts)
```

```
## [1] 136462 139602 137857 139450 141760 142143
```

```r
# Plot the time series
plot(house_price_ts, xlab="Year", ylab="Average House Price (GBP)", main="Monthly Average House Prices 

# Customizing the plot further to improve readability
# Adding grid for better visualization
grid(nx = NULL, ny = NULL, col = "gray", lty = "dotted")

# axis to show years
axis(1, at=seq(2010, 2020, by=1), labels=seq(2010, 2020, by=1))
```

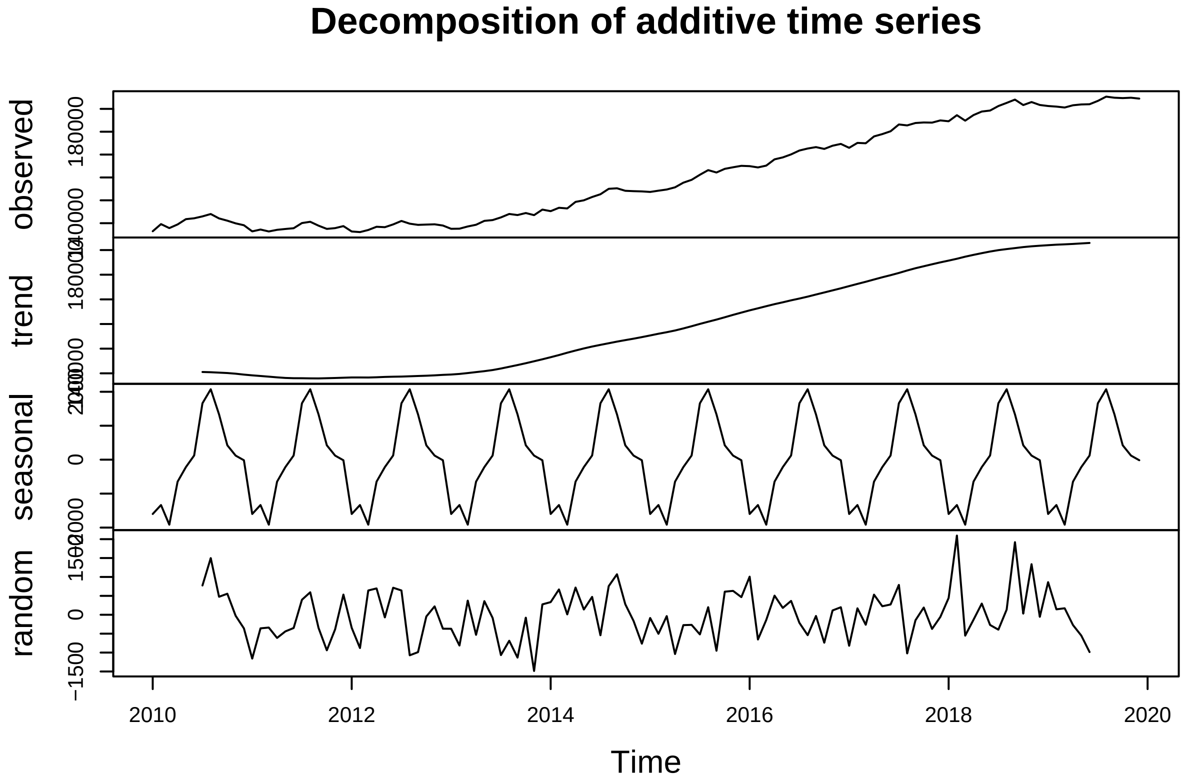


```r
plot(house_price_ts, main="Monthly House Prices in East Midlands", xlab="Time", ylab="Price")
```



The time series of monthly house prices in the East Midlands shows a clear upward trend over the decade from 2010 to 2020. There do not appear to be any extreme outliers or abrupt shifts that would indicate external shocks to the market or data recording errors.

```r
# Decompose the time series
decomposed <- decompose(house_price_ts)
plot(decomposed)
```



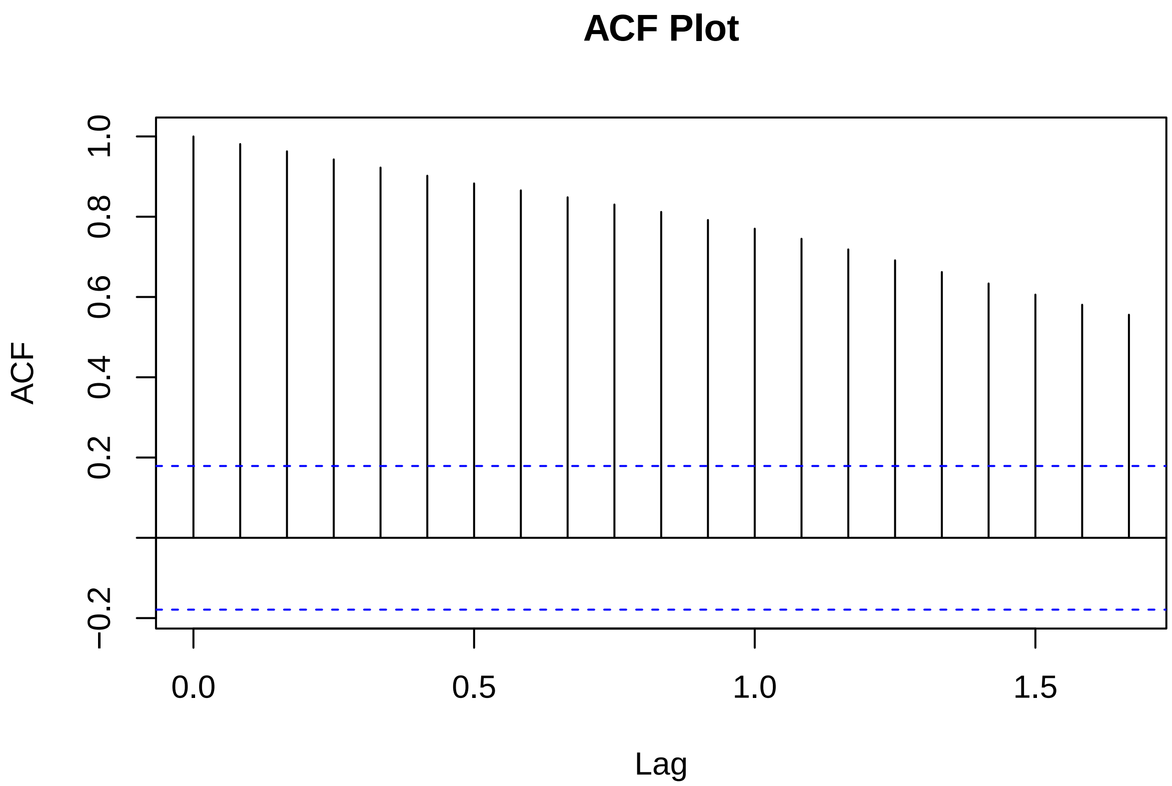
Trend: There's a clear upward trend over time, as we initially saw in the overall time series plot. This suggests that average house prices have been consistently increasing over the period from 2010 to 2019.

Seasonality: The seasonal plot shows a repeating pattern within each year, indicating seasonality in house prices. This is common in housing markets, where certain times of the year may be more popular for buying or selling houses.

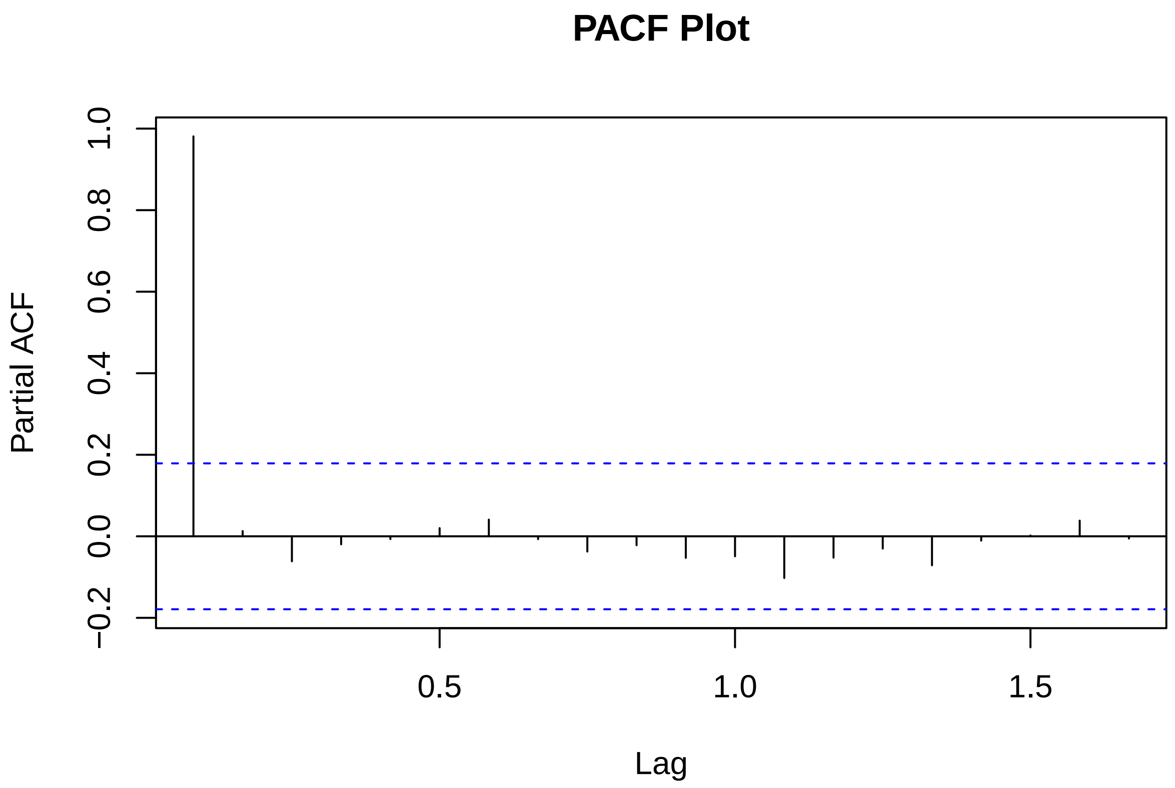
Random: The random (or "residual") component, which is what remains after the trend and seasonality have been accounted for, does not show any alarming patterns at first glance. This suggests that the trend and seasonality components have captured most of the systematic structure in the data.

The presence of both trend and seasonality suggests that an ARIMA model with seasonal differencing might be appropriate, or perhaps a seasonal decomposition of time series (STL) followed by an ARIMA model on the seasonally adjusted data.

```r
# Plotting ACF
acf(house_price_ts, main="ACF Plot")
```



```r
# Plotting PACF
pacf(house_price_ts, main="PACF Plot")
```



ACF Plot Analysis: The ACF plot shows a gradual decline in autocorrelation as the lags increase. The ACF tails off slowly, which typically suggests a non-stationary series. This is consistent with the ADF test result, suggesting that differencing may be required.

PACF Plot Analysis: The PACF plot shows a sharp cut-off after the first lag, then more lags are within the significance level, which suggests a possible AR(1) component. However, there are several significant spikes at higher lags, which may suggest seasonal effects. Implications for Model Selection: Based on the ACF and PACF plots and the previous ADF test:

The non-stationary nature of the time series suggests the need for differencing. This is indicated by the slowly declining ACF plot. The PACF plot suggests that an AR(1) term might be appropriate, as indicated by the significant spike at lag 1. Because the data also contains seasonality, we would typically use a Seasonal ARIMA (SARIMA) model. The SARIMA model has additional seasonal parameters to capture the seasonality observed in the data.

```r
library(tseries)
```

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

```r
# Performing the ADF test for our time-series object, just to check the p-value for stationarity (for t
adf_result <- adf.test(house_price_ts)

# Display the results
print(adf_result)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  house_price_ts
## Dickey-Fuller = -2.9872, Lag order = 4, p-value = 0.1666
## alternative hypothesis: stationary
```

The Dickey-Fuller statistic is not sufficiently negative to fall below the critical values (which are usually around -3.5, -2.9, and -2.6 for the 1%, 5%, and 10% significance levels, respectively). but typically, a value of -2.9872 is not negative enough to declare stationarity at the 5% significance level.The p-value of 0.1666 is above the standard alpha level of 0.05, which means we fail to reject the null hypothesis of a unit root. This suggests that the time series is non-stationary.

```r
# Load the forecast package
library(forecast)

# Using the auto.arima function to automatically select the best ARIMA model for our time series data.
# This function tests various combinations of AR, I, and MA components to find the best fit based on AI
best_model <- auto.arima(house_price_ts)

# Display the model summary
summary(best_model)
```

```
## Series: house_price_ts
## ARIMA(1,1,2)(0,1,1)[12]
##
## Coefficients:
##         ar1      ma1     ma2     sma1
##       0.855  -1.2235  0.5234  -0.8108
## s.e.  0.093   0.0996  0.0901   0.1337
##
## sigma^2 = 911391:  log likelihood = -890.44
## AIC=1790.89   AICc=1791.48   BIC=1804.25
##
## Training set error measures:
##                    ME     RMSE      MAE        MPE     MAPE      MASE
## Training set 82.62172 884.4646 681.5663 0.05423042 0.429677 0.1059921
##                     ACF1
## Training set -0.01188434
```

The fitted model is a SARIMA(1,1,2)(0,1,1)[12].

This indicates one non-seasonal autoregressive (AR) term, one level of non-seasonal differencing, two non-seasonal moving average (MA) terms, one level of seasonal differencing, and one seasonal MA term.

The model coefficients are significant, given the standard errors are relatively small compared to the coefficients.

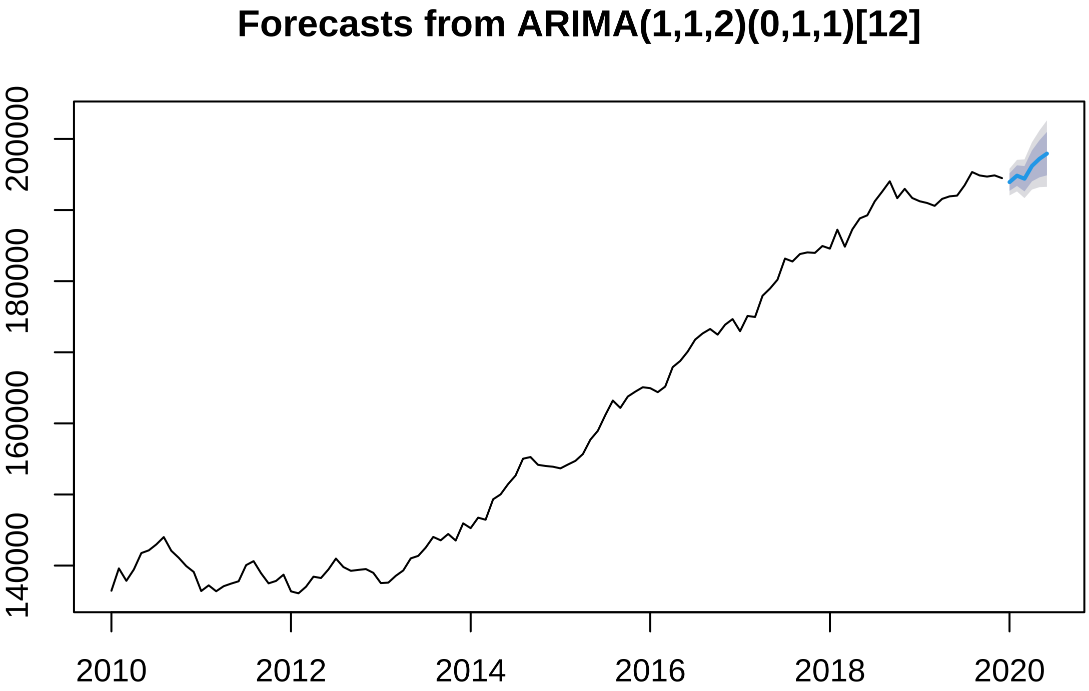
The AIC (Akaike Information Criterion), AICc (corrected AIC), and BIC (Bayesian Information Criterion) values can be used to compare with other models, but without such a comparison, they just tell us that the model is fairly complex due to the penalty for the number of parameters.

The sigma^2 value represents the variance of the residuals, which helps to understand the goodness of fit.

Error metrics on the training set (ME, RMSE, MAE, MPE, MAPE, MASE, and ACF1) provide an indication of the forecast accuracy. The RMSE (Root Mean Square Error) and MAE (Mean Absolute Error) are particularly useful for understanding the average error magnitude.

The Mean Error (ME) is positive, suggesting that the model has a slight bias to underforecast. The MAPE (Mean Absolute Percentage Error) is about 0.4297, which means that the average percentage error is around 42.97%. This may seem high, and depending on the context, it may suggest that the model's accuracy could potentially be improved.

```
# Load the forecast library
library(forecast)


# Forecast the next six months
forecast_prices <- forecast(best_model, h=6)

# Plot the forecast
plot(forecast_prices)
```



```
# Save the plot to a file
jpeg('forecast_prices.jpg')
plot(forecast_prices)
dev.off()
```

```
## pdf
##   2
```

```r
# Saving the forecasted values and the model summary to text files
write.csv(forecast_prices, 'forecasted_prices.csv')
capture.output(summary(best_model), file='model_summary.txt')
```

```r
# Reading the CSV file into a data frame
forecasted_values <- read.csv("forecasted_prices.csv")

# Printing the first few rows of the data frame
head(forecasted_values)
```

```
##          X Point.Forecast    Lo.80    Hi.80    Lo.95    Hi.95
## 1 Jan 2020       193930.5 192702.6 195158.4 192052.6 195808.4
## 2 Feb 2020       194836.4 193384.6 196288.3 192616.0 197056.8
## 3 Mar 2020       194401.0 192620.8 196181.3 191678.3 197123.7
## 4 Apr 2020       196204.1 194029.6 198378.6 192878.5 199529.7
## 5 May 2020       197202.1 194596.7 199807.6 193217.5 201186.8
## 6 Jun 2020       197933.9 194879.2 200988.7 193262.1 202605.7
```

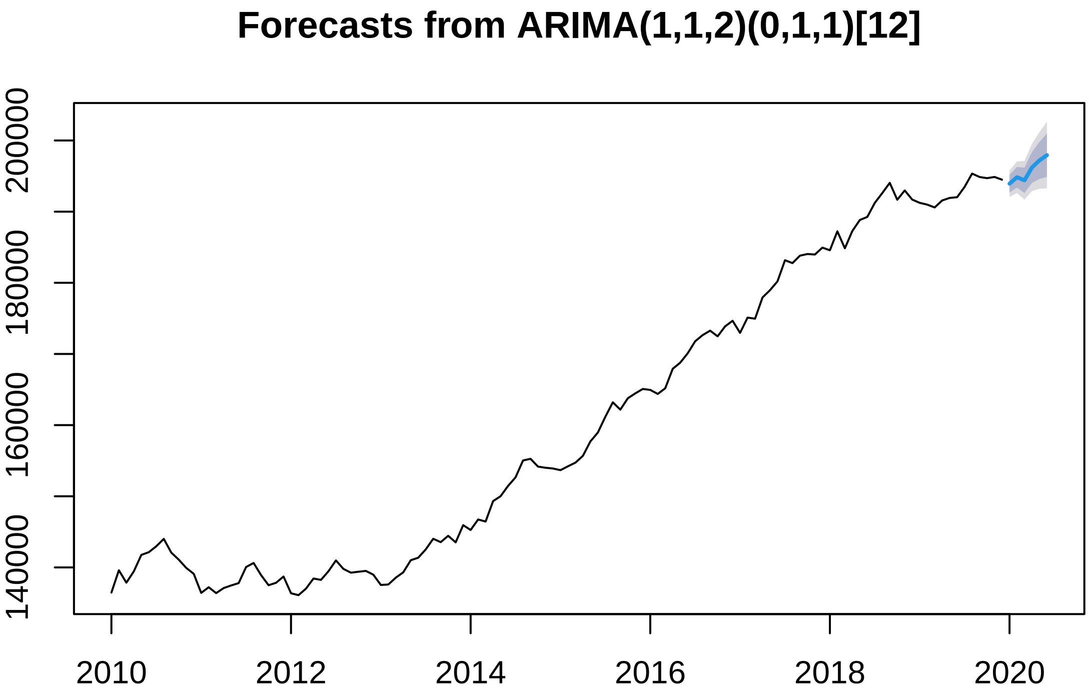
Point Forecasts: The model predicts a general upward trend in house prices from January 2020 to June 2020. The point forecast for January 2020 starts at £193,930 and increases to £197,933 by June 2020.

80% Prediction Intervals: For January 2020, the 80% prediction interval ranges from approximately £192,702 to £195,518. This interval suggests that the model is 80% confident that the actual value will fall within this range. By June 2020, this interval widens from approximately £194,879 to £200,988, reflecting increased uncertainty in the forecast as time goes on.

95% Prediction Intervals: The 95% prediction intervals are wider, as expected, because they represent a higher confidence level. In January 2020, this interval is approximately £192,052 to £195,808. By June 2020, the interval further widens to approximately £193,262 to £202,605, indicating even greater uncertainty further into the future.

Analysis: The forecasted values continue the trend observed in the historical data, which is appropriate if the underlying processes that generated the past data continue into the future. The increasing range of the prediction intervals over time is typical and reflects the accumulating uncertainty associated with forecasting further into the future. The model does not predict any sudden changes or reversals in price trends, which could be expected unless there were known upcoming events or changes in the market that could impact house prices significantly.

```r
library(forecast)


forecast_prices <- forecast(best_model, h=6)

# Round the forecasted prices to whole numbers
forecast_prices$mean <- round(forecast_prices$mean)
forecast_prices$lower[, "80%"] <- round(forecast_prices$lower[, "80%"])
forecast_prices$lower[, "95%"] <- round(forecast_prices$lower[, "95%"])
forecast_prices$upper[, "80%"] <- round(forecast_prices$upper[, "80%"])
forecast_prices$upper[, "95%"] <- round(forecast_prices$upper[, "95%"])


plot(forecast_prices)
```

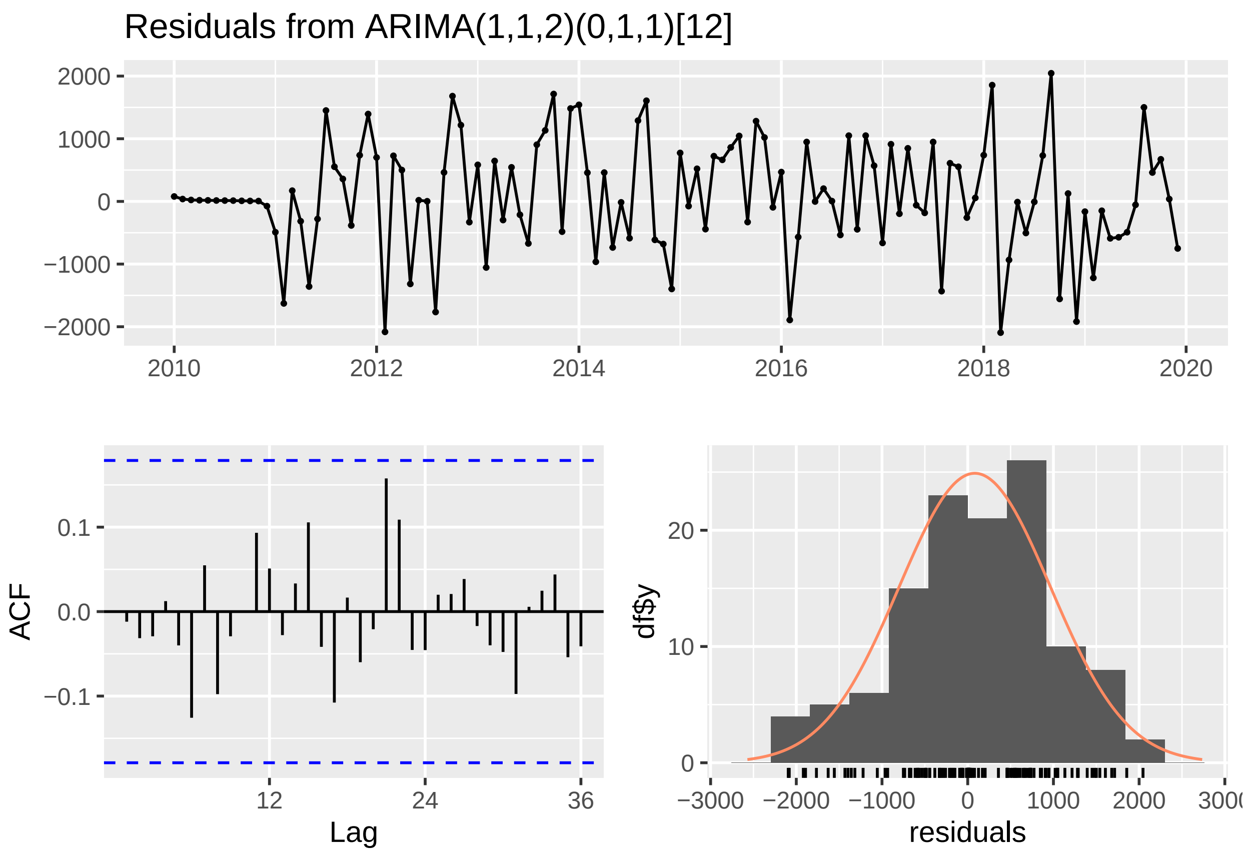


```
jpeg('forecast_prices.jpg')
plot(forecast_prices)
dev.off()
```

```
## pdf
##   2
```

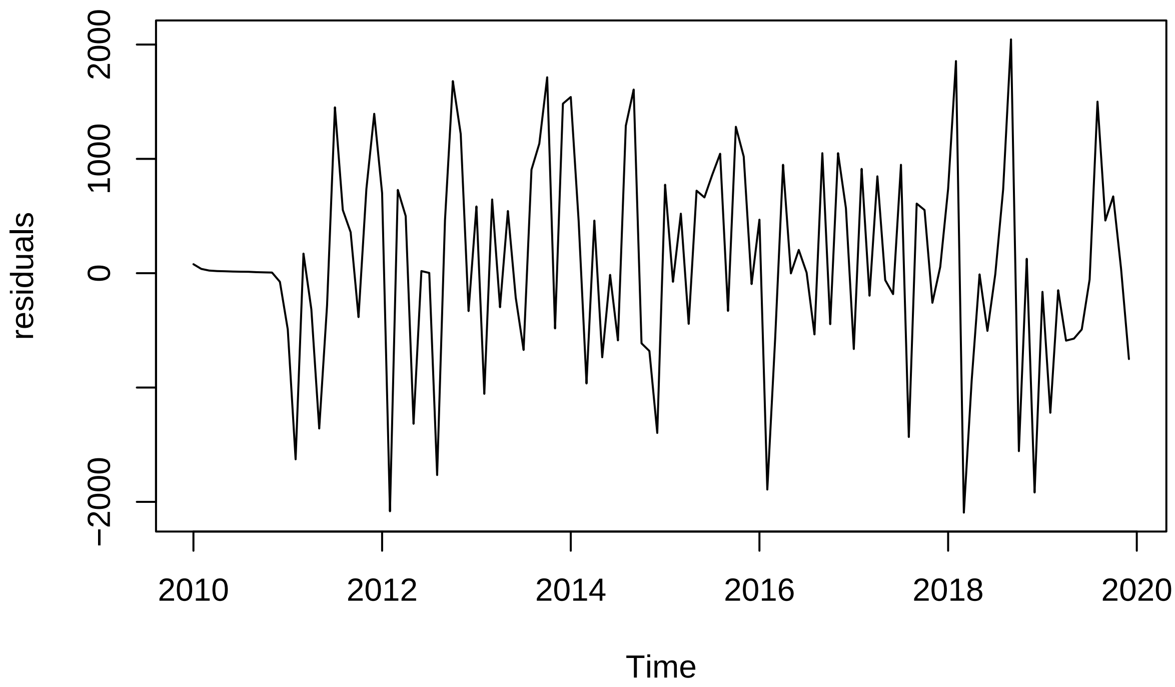
```
# Save the rounded forecasted values to a CSV file
write.csv(forecast_prices, 'rounded_forecasted_prices.csv')


capture.output(summary(best_model), file='model_summary.txt')
```

```
# Plotting model diagnostics
library(forecast)

# Check the residuals of the model
checkresiduals(best_model)
```

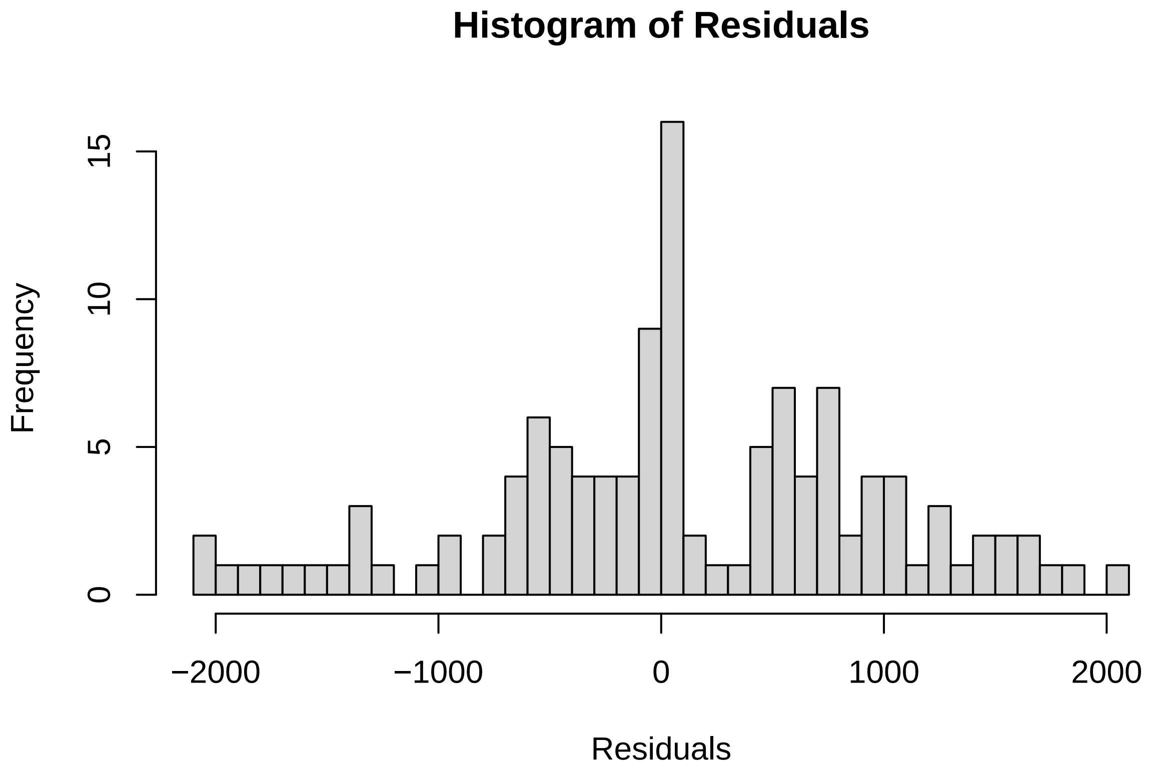


```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,1,2)(0,1,1)[12]
## Q* = 16.172, df = 20, p-value = 0.7059
##
## Model df: 4.   Total lags used: 24
```

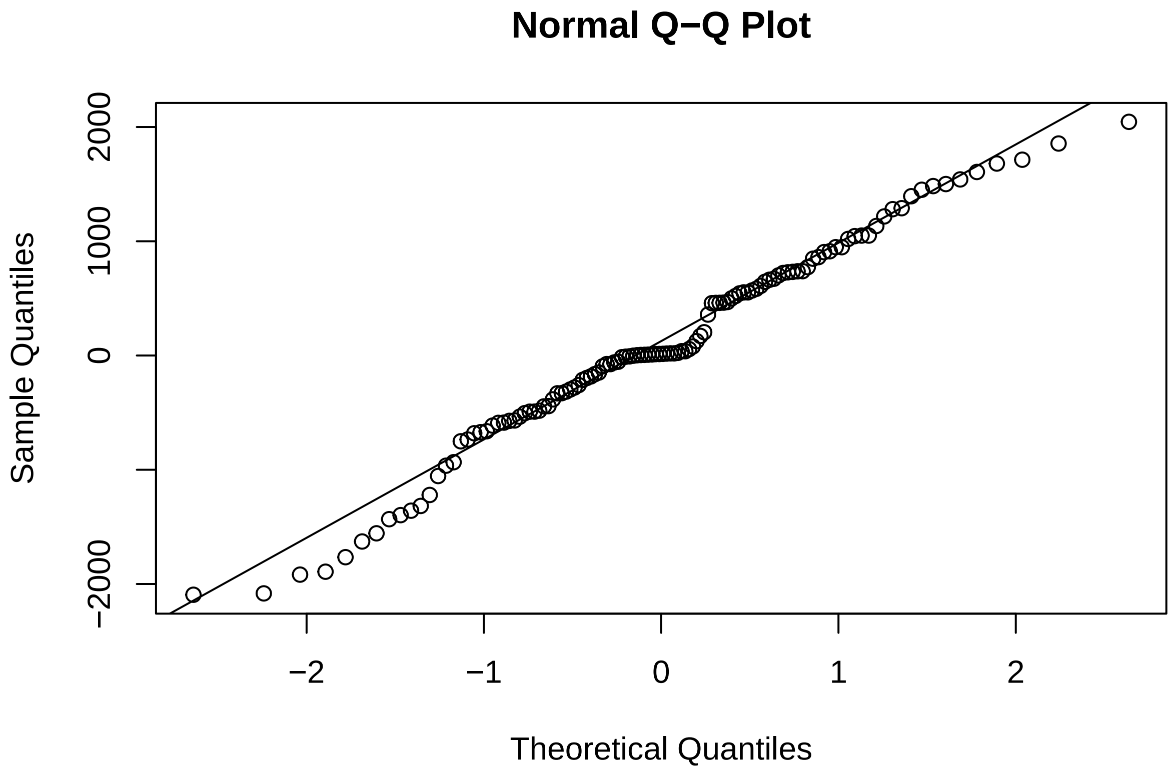
```
# Alternatively, you can manually create the diagnostic plots

# Plot the residuals
residuals <- residuals(best_model)
plot(residuals)
```



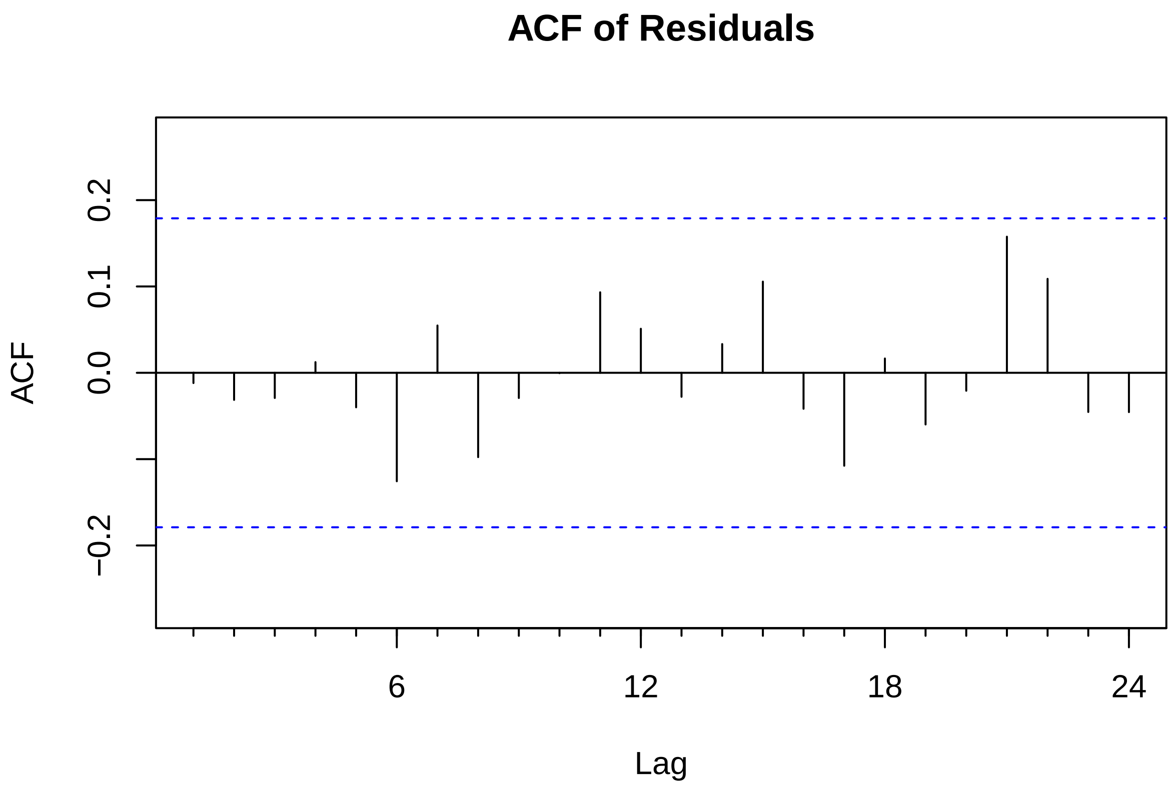
```r
# Histogram of residuals
hist(residuals, breaks=30, main="Histogram of Residuals", xlab="Residuals")
```



```r
# Q-Q plot of residuals
qqnorm(residuals)
qqline(residuals)
```



```r
# ACF plot of residuals
Acf(residuals, main="ACF of Residuals")
```



```
# Perform Ljung-Box test
Box.test(residuals, lag=log(length(residuals)), type="Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  residuals
## X-squared = 0.26588, df = 4.7875, p-value = 0.9975
```

Ljung-Box:

The Ljung-Box test results provided show a p-value of 0.7059 for the test using 20 lags and a p-value of 0.9975 for the test using approximately 5 lags (rounded to 4.7875 due to display). Both p-values are well above the conventional threshold of 0.05, which suggests that there is no significant autocorrelation in the residuals at these lags. This is an indication that the model is adequately capturing the information in the time series, as the residuals appear to be random (white noise), which is what you would expect from a well-fitting time series model.

Interpretation: Lack of Autocorrelation: Since the p-values are high, we fail to reject the null hypothesis of the Ljung-Box test, which means the residuals do not show signs of autocorrelation. This suggests that the model's errors are independent over time, an important assumption in time series forecasting. Model Adequacy: This test result, along with the earlier ADF test and the time series plots, suggests that the SARIMA(1,1,2)(0,1,1)[12] model may be a good fit for this data. Model Selection: Given that the Ljung-Box test indicates a good fit, there might not be a need to look for alternative models, unless there are other considerations, such as performance on a holdout dataset, computational complexity, or specific business requirements. Forecast Confidence: The forecast generated by this model can be considered with a fair amount of confidence for at least the first few lags. For practical decision-making, the confidence intervals of the forecast should be taken into account.

Residuals from ARIMA (1,1,2),(0,1,1)[1 2]

Residual Time Series: The top left plot shows the residuals over time. There are no obvious patterns or trends in the residuals, which is a good sign. This suggests that the model is capturing the underlying structure of the data well.

ACF of Residuals: The bottom left plot is the Autocorrelation Function of the residuals. All autocorrelations are within the blue dashed confidence bands, indicating that there is no significant autocorrelation at any lag. This is another indicator of a good fit since it suggests that the residuals are random, as desired in a well-fitting model.

Histogram of Residuals: The bottom right plot shows the histogram of the residuals with a density overlay. The distribution of residuals appears to be relatively normal, with some minor deviation from the bell curve, particularly a slight skew to the right. However, this is not unusual in practice and may not significantly affect forecasting performance.

Normal Q-Q Plot: The top right plot is the Normal Quantile-Quantile plot. The points follow the line fairly well in the center of the distribution but deviate in the tails, particularly in the right tail. This indicates that the residuals are approximately normal, with some potential outliers or heavy-tailedness.

The residuals do not exhibit any clear autocorrelation, and they appear to be approximately normally distributed, though with a potential slight skew and heavy tails.

Time Series Residuals: Observations: No Obvious Patterns: The residuals fluctuate above and below the zero line without an apparent pattern or trend, which is a positive sign that the model is capturing the main structure of the data without systematic bias. Randomness: The variability of the residuals seems consistent over time, suggesting that there's no obvious change in volatility (homoscedasticity), which is a good characteristic. Outliers: There are a few spikes that stand out, indicating occasional large errors. While a few outliers are common in most time series data, it's worth considering if these could be due to specific events or outliers in the original data that weren't accounted for in the model.

The lack of pattern in the residual plot suggests that the model is appropriate. However, if you frequently observe large spikes (outliers), you might want to consider a model that can handle such anomalies, like a SARIMA model with robust error terms, or perform further analysis to understand what causes these outliers. If these are explainable by known events (e.g., economic crises, policy changes), you could include them as exogenous variables or adjust the series to account for these one-off effects.

HISTOGRAM: Central Tendency: The residuals are centered around zero, which is good because it indicates there is no bias in the forecasts (the model is not systematically overpredicting or underpredicting).

Shape: The shape of the histogram is somewhat bell-shaped, which is an indication that the residuals might be normally distributed. However, there seems to be a slight right skew, given the longer tail on the right side and a

higher frequency of positive residuals compared to negative ones.

Outliers: There are some bins on the far left and right that suggest potential outliers in the residuals, which are residuals that are significantly different from the rest.

The general shape of the histogram is consistent with the assumption of normally distributed residuals, which is an assumption of many time series models. However, the skewness and potential outliers might violate this assumption slightly.

The outliers could be the result of unusual or extreme values in the time series that the model could not account for. It's important to assess whether these are one-off events or indicative of model misspecification.

NORMAL Q-Q PLOT: Central Tendency: Most points in the center of the distribution align well with the line, indicating that the central part of the distribution of residuals is close to normal. Tails: The deviations from the line in the tails (especially the right tail) indicate that the residuals have heavier tails than a normal distribution. This suggests that there are more extreme values in the tails than you would expect if the residuals were perfectly normally distributed. Outliers: The points that deviate significantly from the line in the tails are indicative of outliers. These could be unusual observations that the model did not predict well.

ACF of RESIDUALS:

Observations from the ACF Plot: No Significant Autocorrelation: The autocorrelations at all lags lie within the blue dashed confidence bands, which typically represent a 95% confidence interval. This suggests that the residuals are not significantly autocorrelated. Randomness of Residuals: Since there is no clear pattern in the ACF plot, this implies that the residuals are random, which supports the idea that the model is capturing the underlying process well.

Analysis: Model Sufficiency: The lack of significant autocorrelation in the residuals suggests that the SARIMA(1,1,2)(0,1,1)[12] model is sufficient in capturing the time-dependent structure of the series. Model Fit: This is a good indication that the model fit is appropriate and that the residuals do not have leftover patterns that could be used for further prediction.

Conclusion: The ACF plot supports the conclusion that the SARIMA model is well-specified. Together with the other diagnostic plots and tests (histogram, Q-Q plot, and Ljung-Box test), there is a strong case that the model is performing well. It suggests that the model is a good fit for the historical data and can be used for forecasting.

Model Fit and Residual Analysis: After fitting the SARIMA(1,1,2)(0,1,1)[12] model, diagnostic checks were crucial. The residuals of the model, which represent the differences between observed and predicted values, showed no significant autocorrelation (as evidenced by the Ljung-Box test) and were approximately normally distributed. This lack of pattern or systematic bias in the residuals indicates a good fit, suggesting that the model adequately captures the underlying process in the data .