Name: Oladayo Siyanbola

Introduction

This report focuses on fitting a suitable time series model to the annual mean temperature (in degree Celsius) data for the Midlands region of England, recorded from 1900 to 2021. We will refer to this data as the 'temperature data' throughout the report.

Data Exploratory

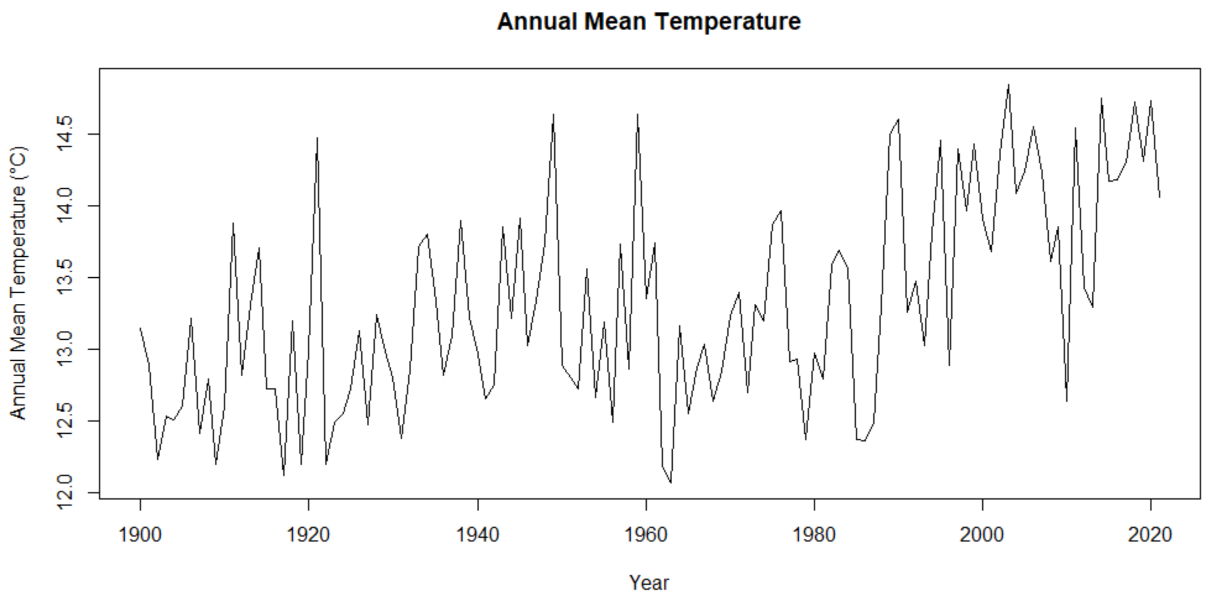

Figure 1: Temperature Time Series

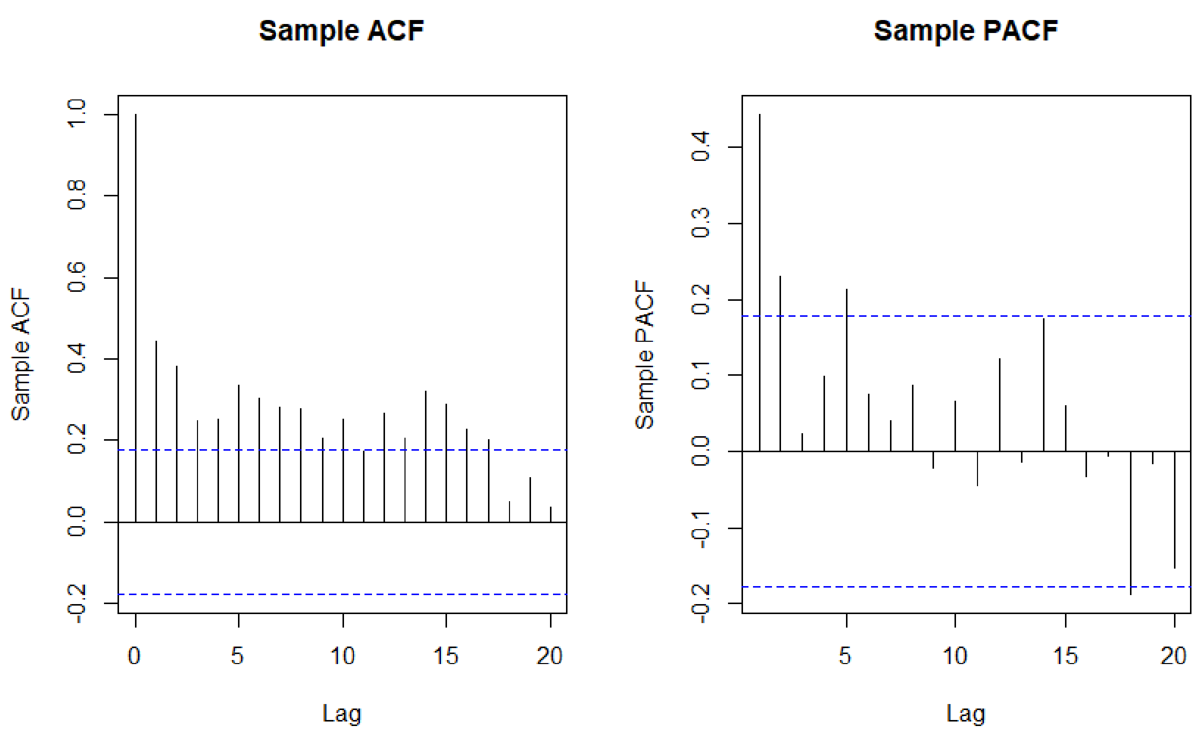

Figure 2: Sample ACF and PACF plots of the Temperature Data

The time plot (Figure 1) shows an upward trend in the average temperature suggesting that the temperature data series is not stationary. This is supported by the sample ACF plot (Figure 2) where the autocorrelation decay slowly with increasing lag, a typical sign of non-stationarity. The sample PACF plot (Figure 2), however, does not offer clear insights into the series stationarity.

To address non-stationarity, we will apply differencing to the series.

Differencing

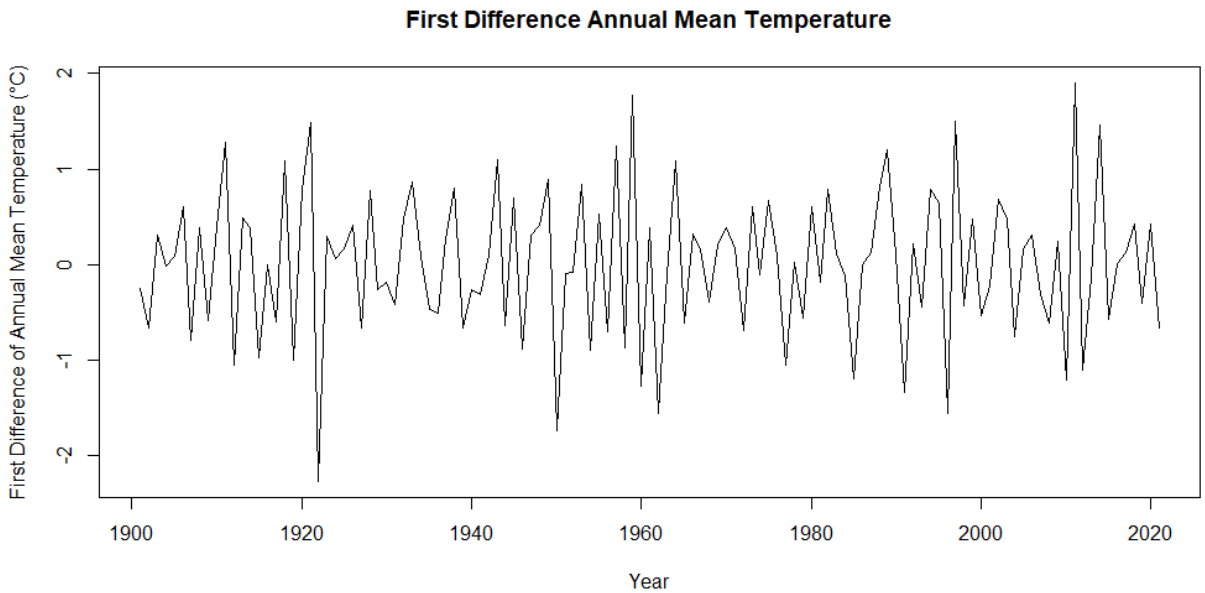

Figure 3: Differenced Temperature Time Series

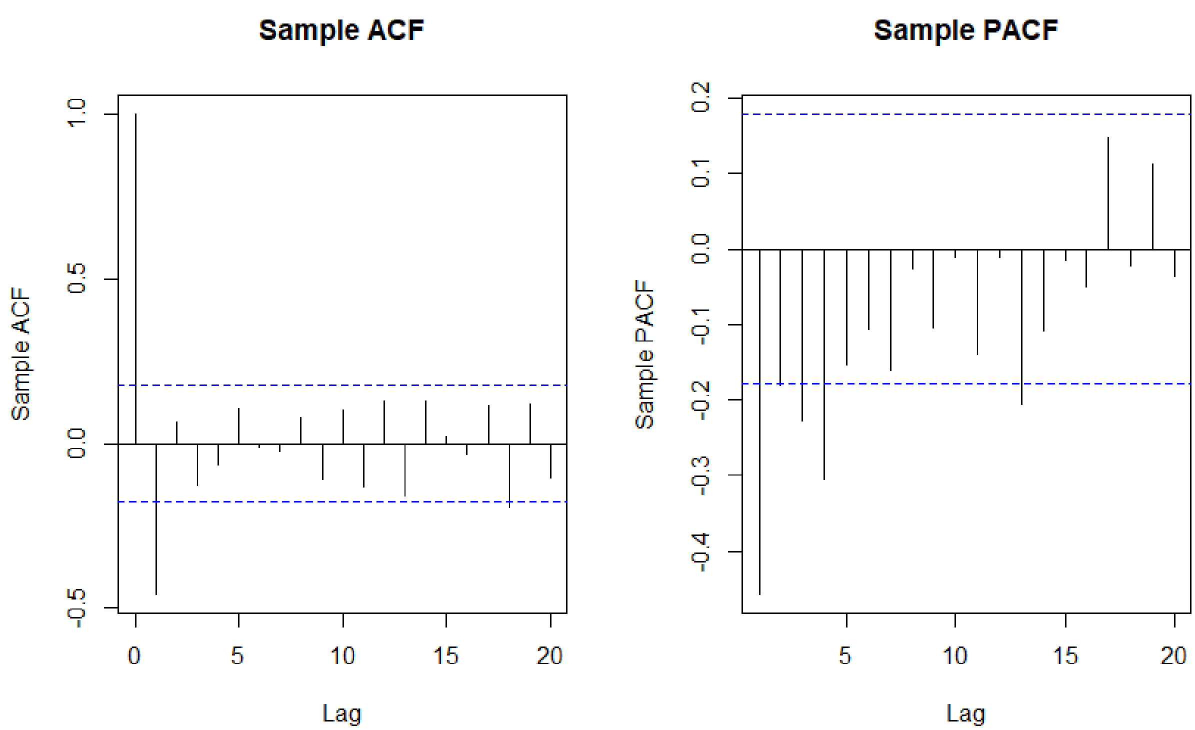

Figure 4: Sample ACF and PACF of the Differenced Temperature Data

After applying first differencing, the series appears stationary. The time plot (Figure 3) shows a constant mean, and the sample ACF plot (Figure 4) exhibits quick decay, cutting off to zero after lag 1. Similarly, the sample PACF plot (Figure 4) also approaches zero fairly rapidly, indicating the removal of the non-stationarity.

Model Fitting

The sample ACF plot (Figure 4) cut off to zero at lag 1, suggesting an ARIMA(0, 1, 1) model might be suitable model to the temperature data. This '1' in the middle position of the model specification refers to the differencing applied to achieve stationarity. However, the sample PACF plot (Figure 4) doesn't exhibit a clear cut off, indicating an ARIMA(p, 1, 0) model might not be ideal. This suggests the process might rely on both its own past values and past errors, making an ARIMA(p, 1, q) model potentially suitable model.

Fitting an ARIMA(0,1,1) Model

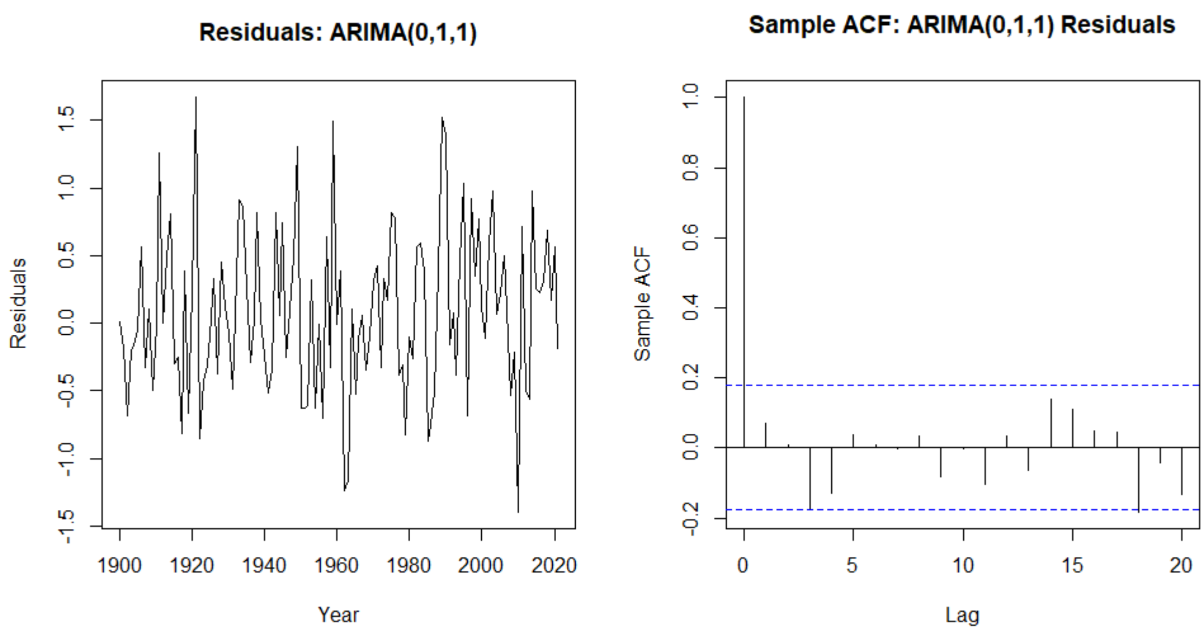

Figure 5: Time Series and Sample ACF plots of ARIMA(0,1,1) Residuals

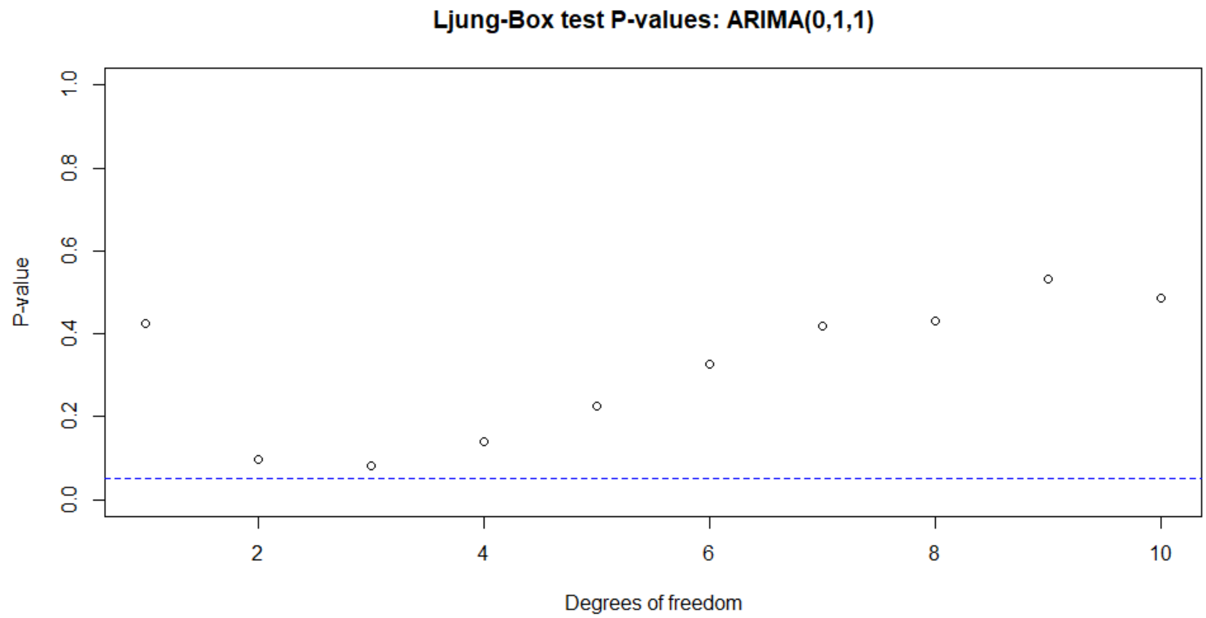

Figure 6: Ljung-Box test for the ARIMA(0,1,1) Residuals

The ARIMA(0,1,1) model provides a good fit for the temperature data. The residuals (Figure 5) from the model exhibit stationarity (constant mean around 0) and uncorrelated behavior in the sample ACF plot (Figure 5), characteristics of white noise.

The Ljung-Box test plot (Figure 6) reinforces this with non-significant p-values (above 0.05) indicating no significant autocorrelation in the residuals.

The ARIMA(0,1,1) model fit to the temperature data yielded the following parameter estimates:

ma1 ($\Theta 1$) = -0.8495 (standard error = 0.048). The model's AIC value is 226.86.

Fitting an ARIMA(1,1,1) Model

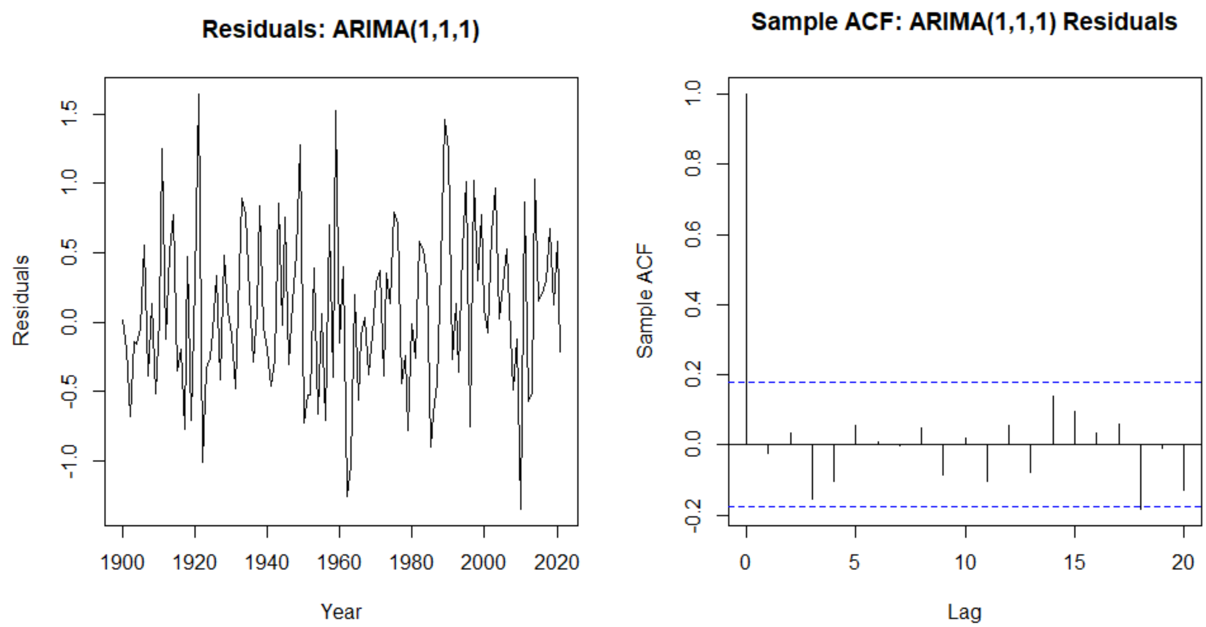

Figure 7: Time Series and Sample ACF plots of ARIMA(1,1,1) Residuals

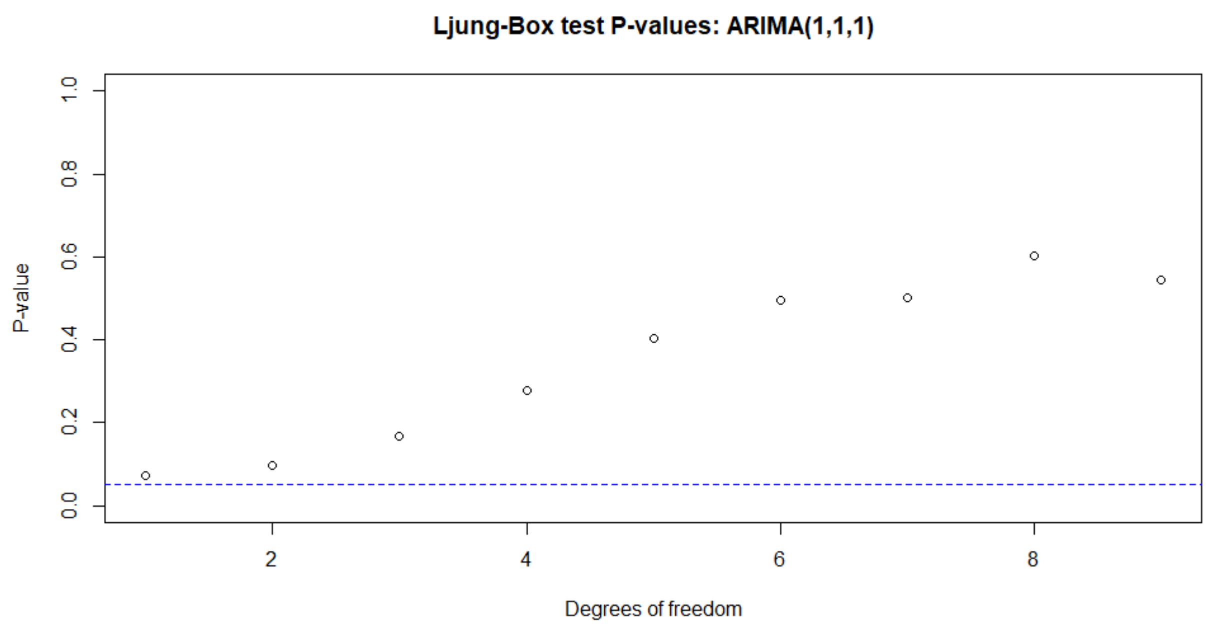

Figure 8: Ljung-Box test for the ARIMA(1,1,1) Residuals

The ARIMA(1,1,1) model also provides a good fit for the temperature data. Residuals from this model exhibit characteristics of white noise, including stationarity (constant mean around 0) and uncorrelated behavior in the sample ACF plot (Figure 7).

The Ljung-Box test plot (Figure 8) further supports this with non-significant p-values (above 0.05) indicating no significant autocorrelation in the residuals.

The ARIMA(1,1,1) model fit to the temperature data yielded the following parameter estimates:

ar1($\Phi 1$) = 0.1137 (standard error = 0.1026), ma1($\Theta 1$) = -0.8749 (standard error = 0.0454). The model's AIC value is 227.63.

Model Selection

Given that both the ARIMA(0,1,1) and ARIMA(1,1,1) models appear suitable, a two-step approach will be used to select the parsimonious model. This involves hypothesis testing and comparing the Akaike Information Criterion (AIC) values of both models.

Hypothesis Testing

To identify the more parsimonious model, we perform a hypothesis on the ar1(Φ1) parameter of the ARIMA(1,1,1) model. The null hypothesis (H_o) states that ar1(Φ1) is equal to zero, implying the ARIMA(1,1,1) simplifies to an ARIMA(0,1,1). The alternative hypothesis (H_1) is that ar1(Φ1) is not equal to zero.

A common test statistic for this purpose is the absolute value of the parameter estimate divided by its standard error (|Φ1 / standard error(Φ1)|). We reject H_o if this value is greater than a critical value (typically 2). In this case, the estimated Φ1 is 0.1137 with a standard error of 0.1026. The test statistic, calculated as 1.1081, is less than the critical value (2).

Therefore, we fail to reject the null hypothesis. This suggests that the ar1(Φ1) component in the ARIMA(1,1,1) model might not be statistically significant, potentially justifying a simpler ARIMA(0,1,1) model.

Akaike Information Criterion

Akaike Information Criterion (AIC) supports the selection of the ARIMA(0,1,1) model. The ARIMA(0,1,1) model has a lower AIC value (226.86) compared to the ARIMA(1,1,1) model (227.63) indicating a better fit for the temperature data.

This aligns with the hypothesis test result, where we failed to reject the null hypothesis suggesting the ARIMA(0,1,1) model might be sufficient.

Therefore, the ARIMA(0,1,1) model appears to be the most suitable choice for the temperature data.

Equation of the Final Fitted Model

The equation of the final fitted model will be:

$$X_t - X_{t-1} = Z_t + \theta 1 Z_{t-1}$$

$$X_t = X_{t-1} + Z_t - 0.8495\, Z_{t-1}$$

Conclusion

This analysis identified the ARIMA(0,1,1) model as the most suitable choice for capturing the temperature data's time series behavior. Both hypothesis testing and AIC supported this selection. The ARIMA(0,1,1) model effectively addresses non-stationarity through differencing (integrated order I(1)) and incorporates the influence of past errors (moving average term MA(1)) on the current changes in annual mean temperature.

Appendix

R Code

```
#Read in the dataset ‘cet_temp.csv’

cet_temp <- read.csv('cet_temp.csv', header = TRUE)

#Set the dataset to be time series data

cet_temp_ts <- ts(cet_temp$avg_annual_temp_C, start = 1900, frequency = 1)

#Produce a time plot of the time series data ‘cet_temp_ts’

ts.plot(cet_temp_ts, ylab = 'Annual Mean Temperature (\u00B0C)', xlab = 'Year', main = 'Annual Mean Temperature')

#Set the R graphics device to contain two plots (1 row, 2 columns)

par(mfrow = c(1,2))

#Plot the sample ACF of the time series data

acf(cet_temp_ts, ylab = 'Sample ACF', main = 'Sample ACF')

#Plot the sample PACF of the time series data

pacf(cet_temp_ts, ylab = 'Sample PACF', main = 'Sample PACF')

#Take the first difference of the time series data to remove the trend

cet_temp_ts_diff <- diff(cet_temp_ts)

#Set the R graphics device to contain 1 plot (1 row, 1 column)

par(mfrow = c(1,1))

#Produce a time plot of the first difference time series data ‘cet_temp_ts_diff’

ts.plot(cet_temp_ts_diff, ylab = 'First Difference of Annual Mean Temperature (\u00B0C)', xlab = 'Year', main = 'First Difference Annual Mean Temperature')

#Set the R graphics device to contain 1 plot (1 row, 2 columns)

par(mfrow = c(1,2))

#Plot the sample ACF of the first difference time series data

acf(cet_temp_ts_diff, ylab = 'Sample ACF', main = 'Sample ACF')

#Plot the sample PACF of the first difference time series data

pacf(cet_temp_ts_diff, ylab = 'Sample PACF', main = 'Sample PACF')
```

```
#Fit an ARIMA(0,1,1) model to the time series data 'cet_temp_ts'

arima011_model <- arima(cet_temp_ts, order = c(0,1,1), method = 'ML')

#Call the arima011_model (ARIMA(0,1,1)) to get it estimated parameters

arima011_model

#Extract the residuals of the arima011_model

arima011_residuals <- residuals(arima011_model)

#Produce a time plot of the arima011_model residuals

ts.plot(arima011_residuals, ylab = 'Residuals', xlab = 'Year', main = 'Residuals: ARIMA(0,1,1)')

#Plot the sample ACF of the arima011_model residuals

acf(arima011_residuals, ylab = 'Sample ACF', main = 'Sample ACF: ARIMA(0,1,1) Residuals')

#Declare the Ljung-Box test variable term

LB_test<-function(resid,max.k,p,q){

 lb_result<-list()

 df<-list()

 p_value<-list()

  for(i in (p+q+1):max.k){

   lb_result[[i]]<-Box.test(resid,lag=i,type=c("Ljung-Box"),fitdf=(p+q))

   df[[i]]<-lb_result[[i]]$parameter

   p_value[[i]]<-lb_result[[i]]$p.value

  }

 df<-as.vector(unlist(df))

 p_value<-as.vector(unlist(p_value))

 test_output<-data.frame(df,p_value)

 names(test_output)<-c("deg_freedom","LB_p_value")

 return(test_output)

 }

#Perform Ljung-Box tests for the arima011 (ARIMA(0,1,1)) model residuals

ARIMA011.LB <- LB_test(arima011_residuals, max.k = 11, p = 0, q = 1)
```

```
#Set the R graphics device to contain 1 plot (1 row, 1 column)

par(mfrow = c(1,1))

#Produce a plot of P-values against the degrees of freedom and add a blue dashed line at 0.05

plot(ARIMA011.LB$deg_freedom, ARIMA011.LB$LB_p_value, xlab = 'Degrees of freedom', ylab = 'P-value', main = 'Ljung-Box test P-values: ARIMA(0,1,1) Residuals', ylim = c(0,1))

abline(h = 0.05, col = 'blue', lty = 2)


#Fit an ARIMA(1,1,1) model to the time series data ‘cet_temp_ts’

arima111_model <- arima(cet_temp_ts, order = c(1,1,1), method = 'ML')

#Call the arima111_model (ARIMA(1,1,1)) to get it estimated parameters

arima111_model

#Extract the residuals of the arima111_model

arima111_residuals <- residuals(arima111_model)

#Set the R graphics device to contain 1 plot (1 row, 2 columns)

par(mfrow = c(1,2))

#Produce a time plot of the arima111_model residuals

ts.plot(arima111_residuals, ylab = 'Residuals', xlab = 'Year', main = 'Residuals: ARIMA(1,1,1)')

#Plot the sample ACF of the arima111_model residuals

acf(arima111_residuals, ylab = 'Sample ACF', main = 'Sample ACF: ARIMA(1,1,1) Residuals')

#Perform Ljung-Box tests for the arima111 (ARIMA(1,1,1)) model residuals

ARIMA111.LB <- LB_test(arima111_residuals, max.k = 11, p = 1, q = 1)

#Set the R graphics device to contain 1 plot (1 row, 1 column)

par(mfrow = c(1,1))

#Produce a plot of P-values against the degrees of freedom and add a blue dashed line at 0.05

plot(ARIMA111.LB$deg_freedom, ARIMA111.LB$LB_p_value, xlab = 'Degrees of freedom', ylab = 'P-value', main = 'Ljung-Box test P-values: ARIMA(1,1,1) Residuals', ylim = c(0,1))

abline(h = 0.05, col = 'blue', lty = 2)
```

Executive Summary

Understanding Midlands Housing Trends: A Tool for Smarter Decisions

The report examines Midlands average monthly house sale prices from January 2010 to December 2019. We observed house prices generally rise each year compared to the previous year. Additionally, a predictable seasonal pattern emerges annually: prices start lower in January, peak in the summer, and decrease towards December.

We have developed a method to accurately capture these trends and patterns, enabling us to forecast future house prices. This forecasting tool can assist local government officials make informed decisions about infrastructure planning and policy development related to housing affordability.

For example, our forecast suggest continued house prices increases in the first half of 2020, reaching their peak in June. By understanding these trends, local government can make proactive, informed decisions that benefit the entire Midlands community.

Introduction

This report analyzes average monthly house sale prices (GBP) data in the Midlands region of England from January 2010 to December 2019. Our goal is to fit a time series model to forecast house sale prices for the first half of 2020. Throughout this report, we will refer to the data as 'housing price data'.

Data Exploratory

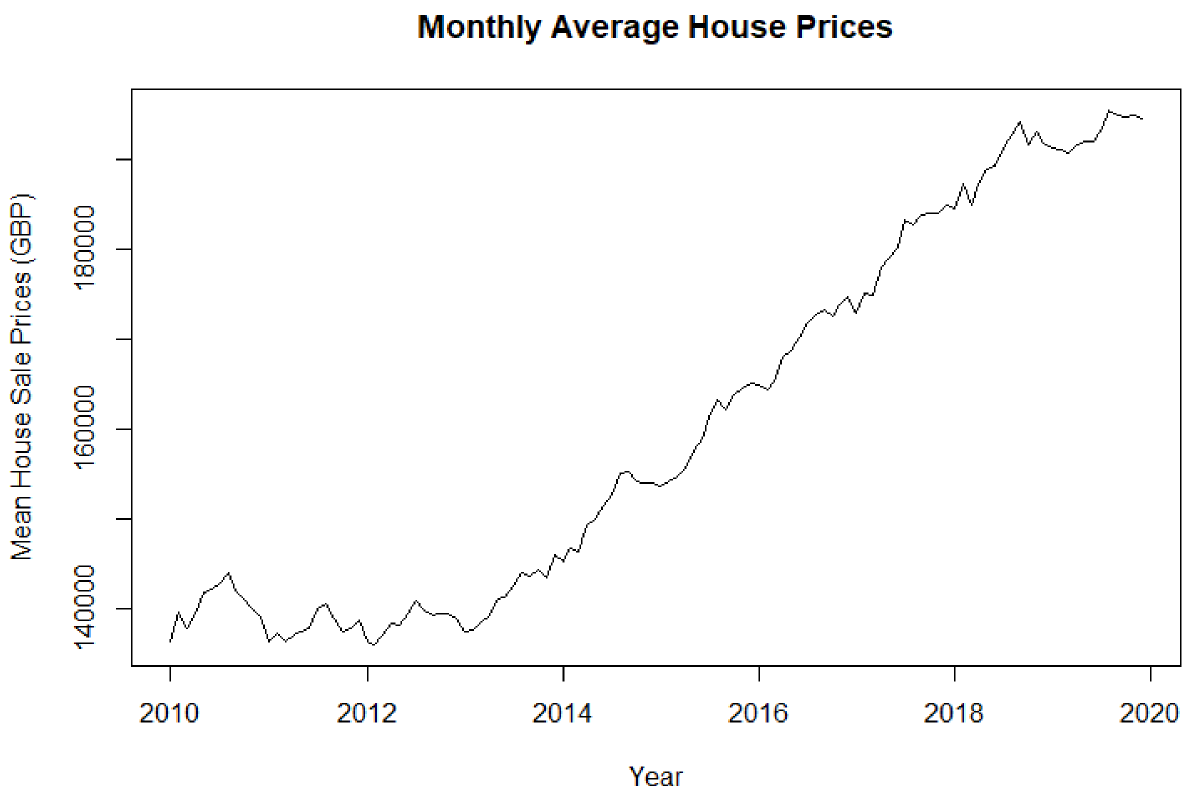

Figure 9: Housing Price Time Series

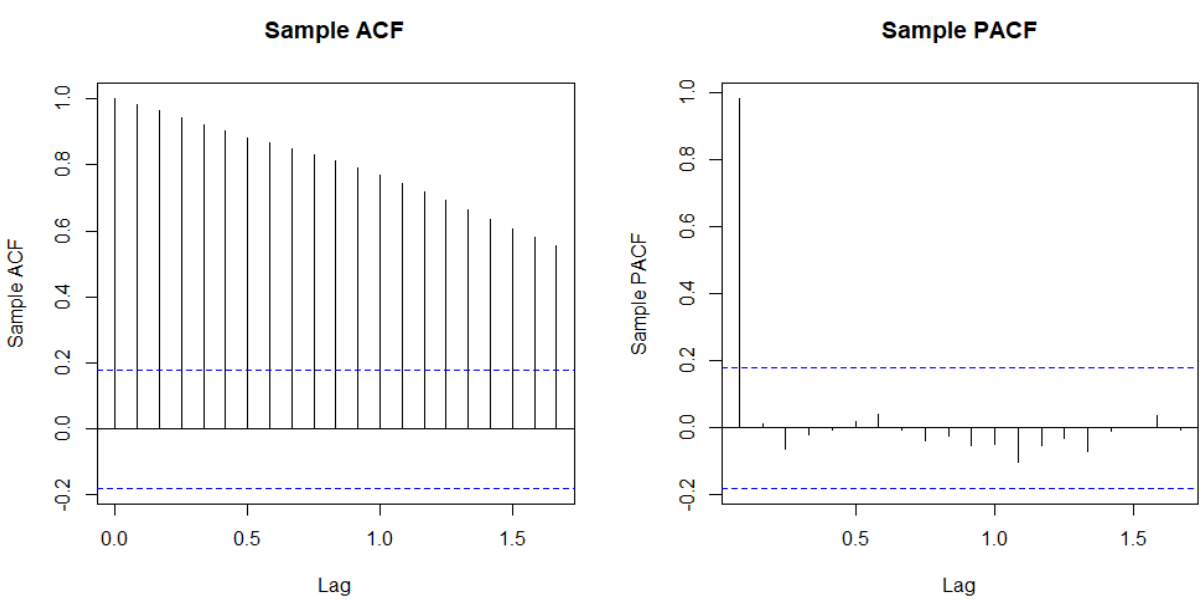

Figure 10: Sample ACF and PACF plots of the Housing Price Data

The time plot (Figure 9) indicates a non-stationary series. House sale prices exhibit an upward trend over time and seasonality. Prices typically start low in January, peak around mid-year, and then decline towards December, repeating this seasonal pattern annually.

The sample ACF plot (Figure 10) supports the non-stationarity claim. The ACF plot shows slow decay in autocorrelation with increasing lag, indicating non-stationarity.

To address non-stationarity, we will apply differencing to the series.

Differencing

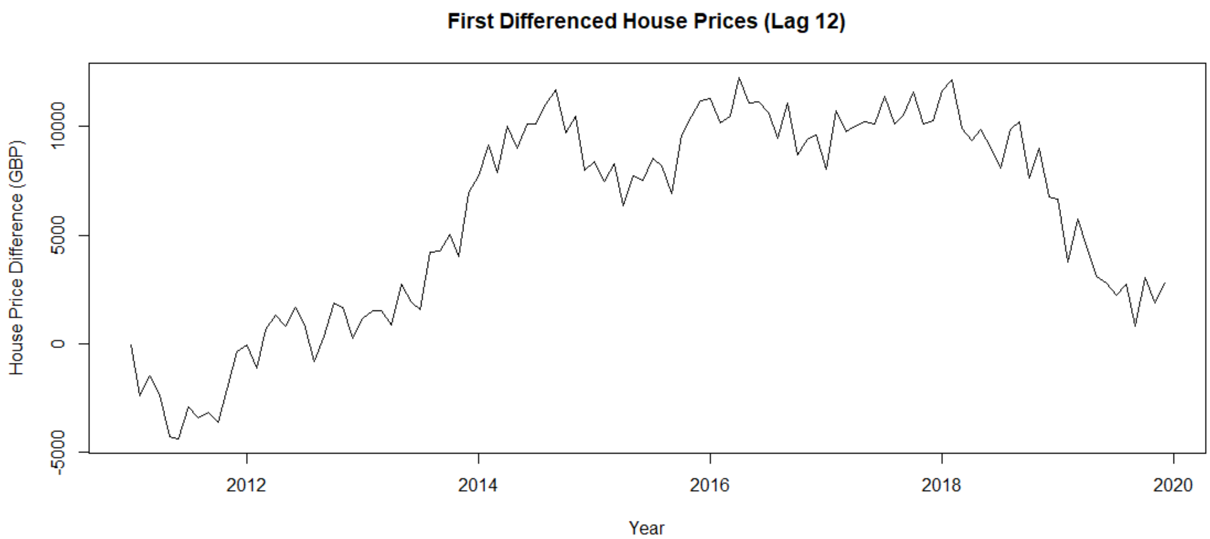

Figure 11: First-Differenced (Lag 12) Housing Price Time Series

Figure 12: ACF and PACF plots of the First Differenced Housing Price Data (Lag 12)

Differencing the housing price data series at lag 12 successfully removes seasonality, as evident in the time plot (Figure 11). However, a trend persists in the differenced data suggesting non-stationarity. The slow decay of the sample ACF plot (Figure 12) at increasing lags further confirms this.

To remove the trend, we will apply a second differencing step to the series.

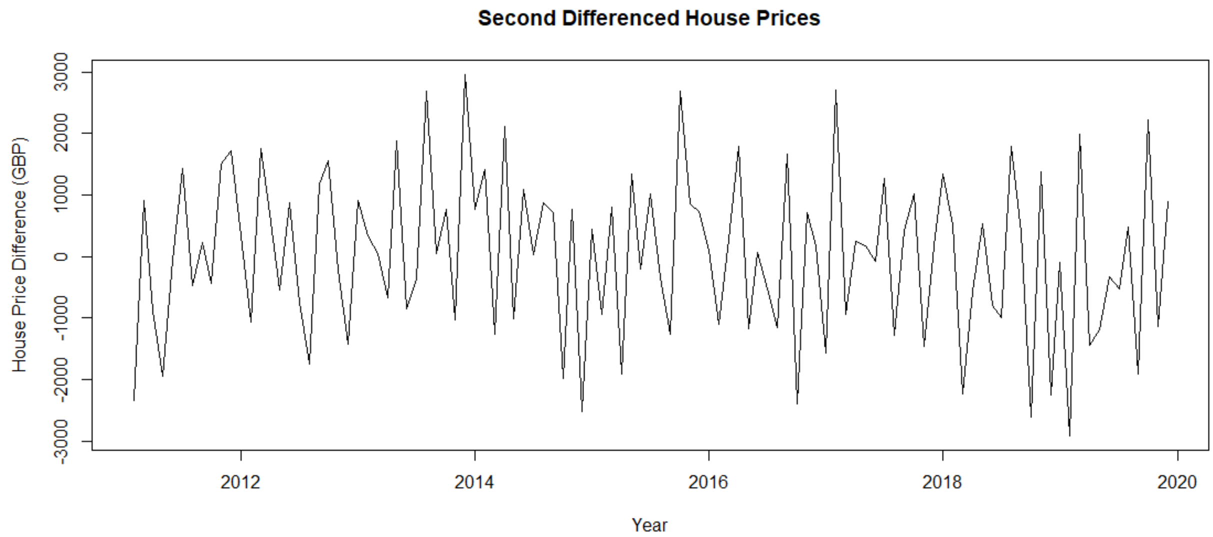

Figure 13: Second-Differenced Housing Price Time Series

Figure 14: ACF and PACF plots of the Second Differenced Housing Price Data

Second differencing removes the trend as seen in Figure 13. The series now exhibits stationary, characterized by a constant mean and a rapid decline in the sample ACF (Figure 14) to zero after lag 1.

Model Fitting

The sample ACF (Figure 14) cuts off to zero after lag 1, suggesting a possible moving average (MA) process. On the other hand, the sample PACF (Figure 14) also appears to cut off to zero after lag 1, suggesting a potential autoregressive (AR) process. To explore potential model fits, we will initially consider ARIMA(0,1,1) x $(0,1,1)_{12}$, ARIMA(1,1,0) x $(1,1,0)_{12}$ and ARIMA(1,1,1) x $(1,1,1)_{12}$ to the housing prices data. These models include both non-seasonal and seasonal components. . This ‘1’ in the middle position of the non-seasonal part indicates differencing to remove the trend, while the ‘1’ in the middle position of the seasonal component’s represent differencing at lag 12 to remove the seasonality.

Fitting an ARIMA(0,1,1) x $(0,1,1)_{12}$ Model

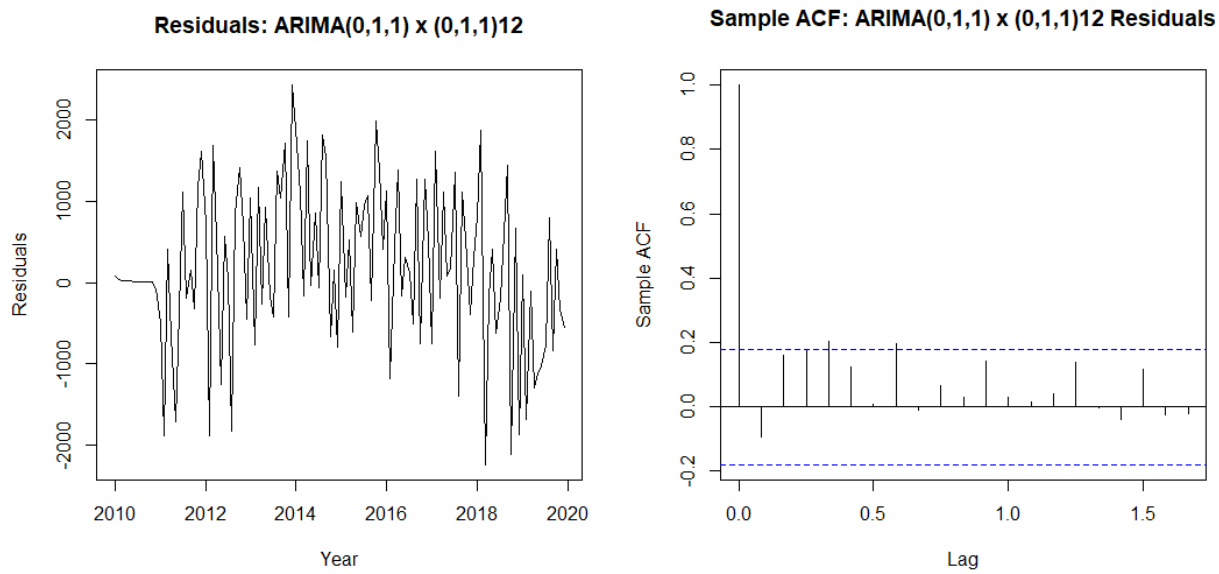

Figure 15: Time and Sample ACF plots of the ARIMA (0,1,1) x $(0,1,1)_{12}$ Residuals

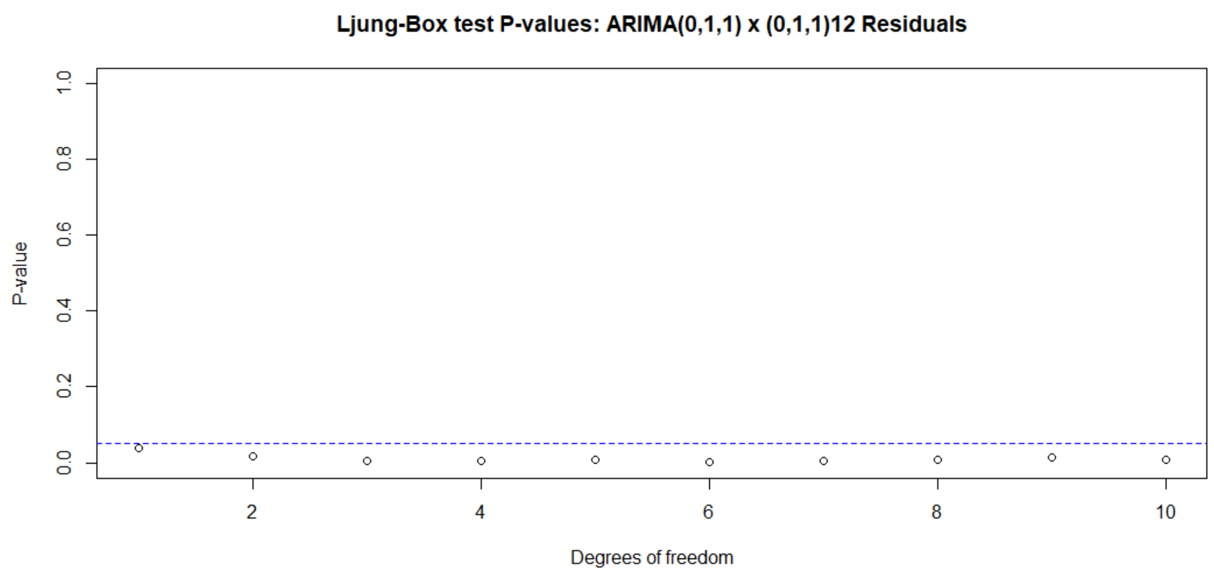

Figure 16: Ljung-Box test for the ARIMA(0,1,1) x $(0,1,1)_{12}$ Residuals

The ARIMA(0,1,1) x $(0,1,1)_{12}$ model fails to capture the housing price data adequately. While the residuals exhibit stationary in the time plot (Figure 15) and the ACF appears uncorrelated (Figure 15), the Ljung-Box test (Figure 16) shows significant autocorrelation in the residuals (p-values < 0.05 for all lags) indicating inadequacy of the model.

Fitting an ARIMA(1,1,0) x $(1,1,0)_{12}$ Model

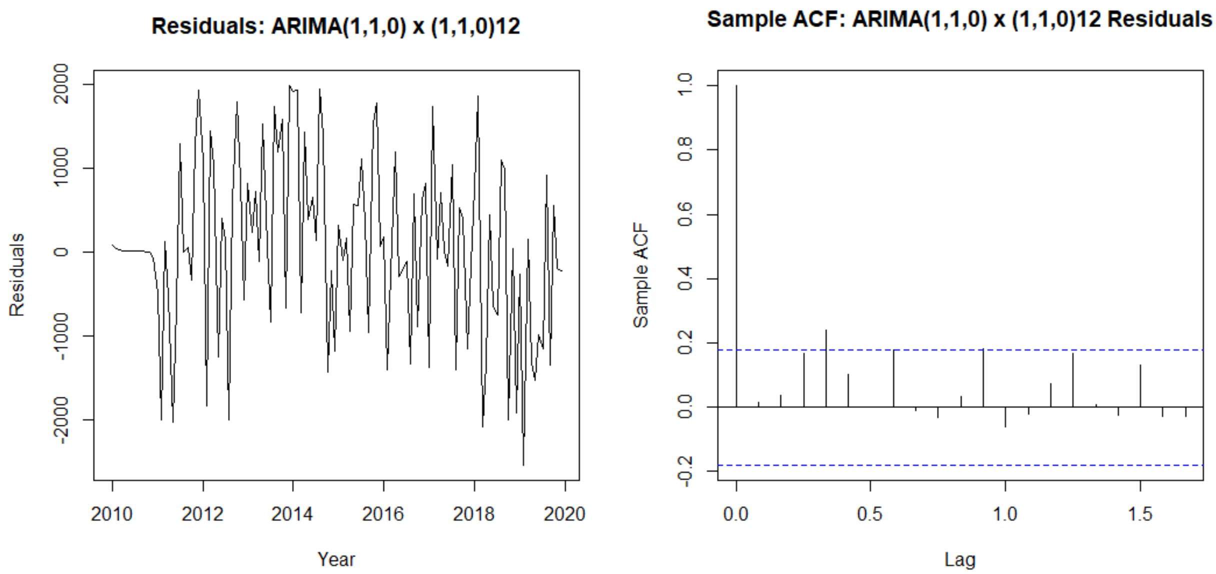

Figure 17: Time and Sample ACF plots of the ARIMA (1,1,0) x $(1,1,0)_{12}$ Residuals

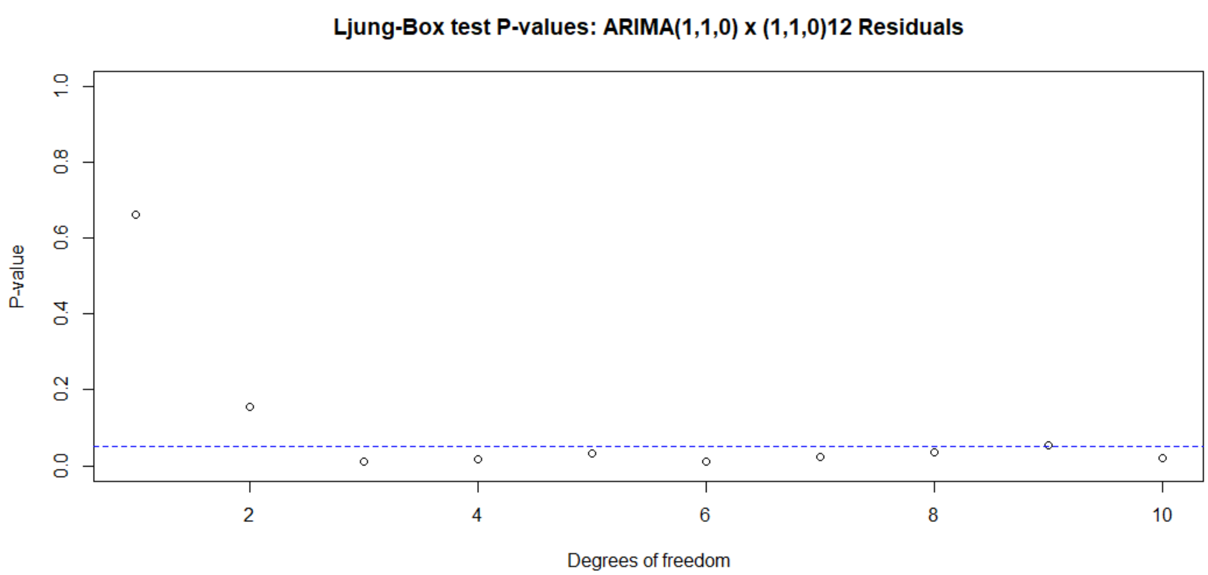

Figure 18: Ljung-Box test for the ARIMA (1,1,0) x $(1,1,0)_{12}$ Residuals

Similar to the ARIMA(0,1,1) x $(0,1,1)_{12}$, the ARIMA(1,1,0) x $(1,1,0)_{12}$ also fails to adequately fit the housing price data. The residuals exhibit stationarity in the time plot (Figure 17) and the ACF appears uncorrelated (Figure 17). However, the Ljung-Box test (Figure 18) indicates significant autocorrelation in the residuals (p-values < 0.05 for most lags) indicating the model inadequacy.

Fitting an ARIMA(1,1,1) x $(1,1,1)_{12}$ Model

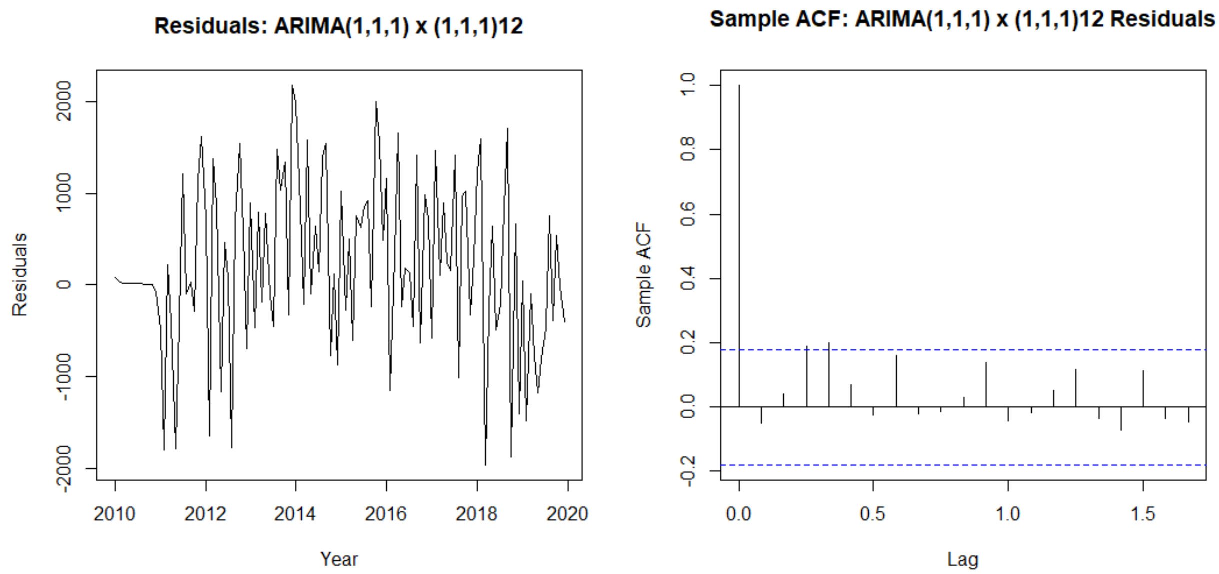

Figure 19: Time and Sample ACF plots of the ARIMA (1,1,1) x $(1,1,1)_{12}$ Residuals

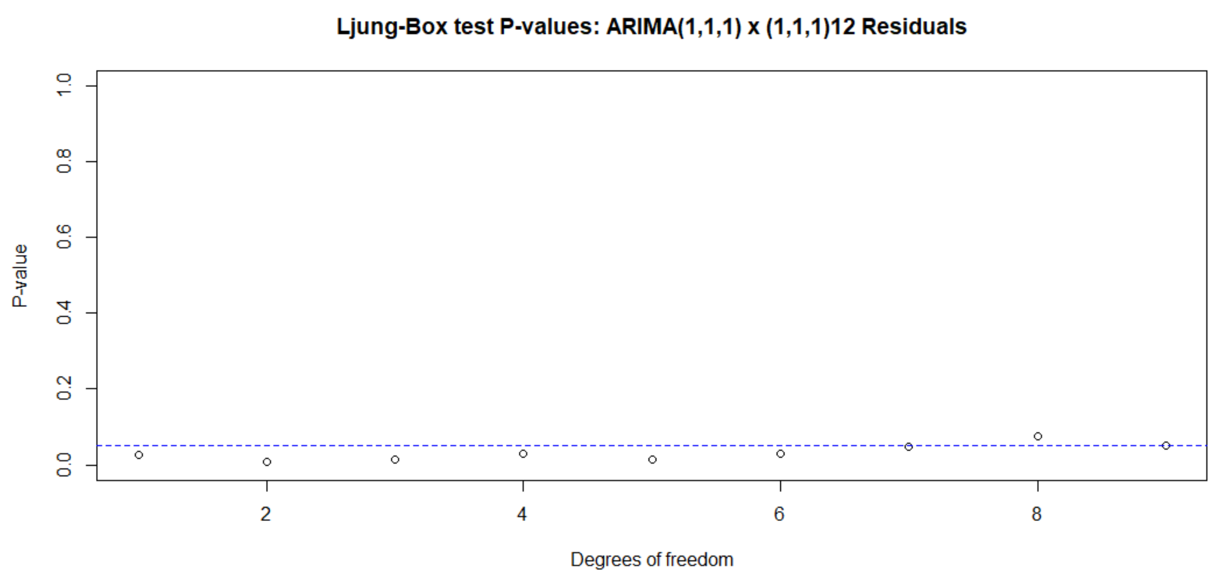

Figure 20: Ljung-Box test for the ARIMA (1,1,1) x $(1,1,1)_{12}$ Residuals

Similar to the previous models, the ARIMA(1,1,1) x $(1,1,1)_{12}$ fails to capture the housing price data adequately. The residuals appears stationary in the time plot (Figure 19) and the ACF suggests no correlations in the residuals (Figure 19). However, the Ljung-Box test (Figure 20) rejects the null hypothesis (p-value < 0.05) for some lags, hinting at possible statistically significant autocorrelation in the residuals thereby making the model inadequate.

Having established that these three (3) models are inadequate, we will explore higher orders of the ARIMA model.

Fitting an ARIMA(1,1,2) x $(1,1,2)_{12}$ Model

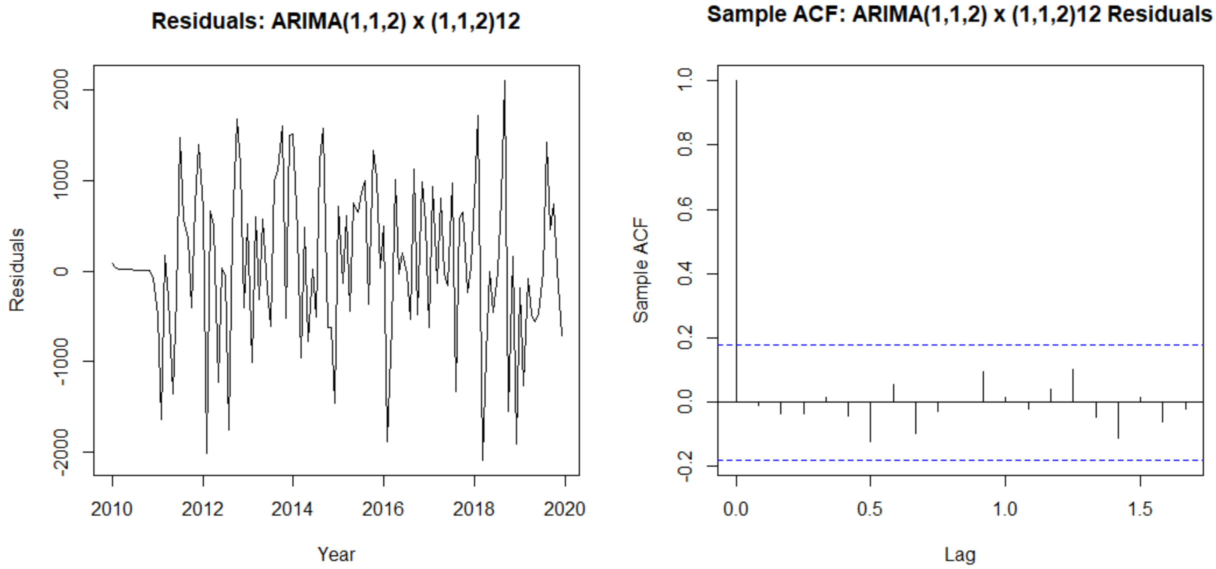

Figure 21: Time and Sample ACF plots of the ARIMA (1,1,2) x $(1,1,2)_{12}$ Residuals

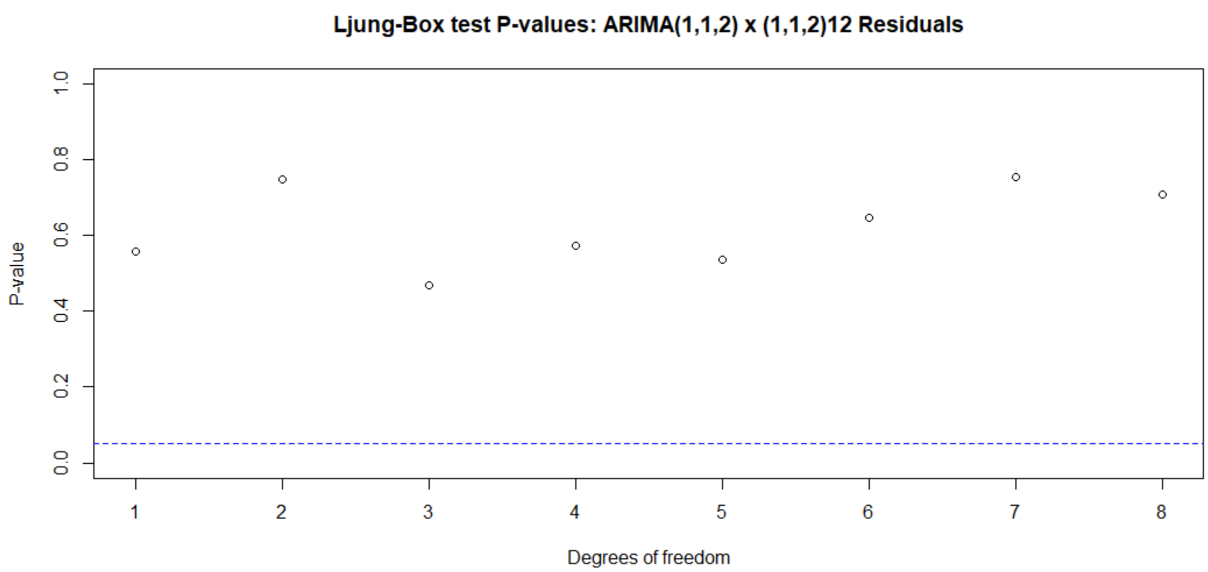

Figure 22: Ljung-Box test for the ARIMA (1,1,2) x $(1,1,2)_{12}$ Residuals

The ARIMA(1,1,2) x $(1,1,2)_{12}$ model appears to be a good fit to the housing price data. The residuals exhibit stationarity with constant mean around 0 (Figure 21) and the ACF appears uncorrelated (Figure 21). Furthermore, Ljung-Box test (Figure 22) confirms this as it fails to reject the null hypothesis at any lag (p-value > 0.05 for all lags). This implies there is no statistically significant autocorrelation in the residuals, consistent with the characteristics of a white noise process.

The ARIMA(1,1,2) x $(1,1,2)_{12}$ model fitted to the housing price data yielded parameter estimates presented in Table 1 below:

Parameter	Estimate	Standard Error of Estimate
ar1 (Φ1)	0.8401	0.0979
ma1 (Θ1)	-1.2135	0.1029
ma2 (Θ2)	0.5285	0.0899
sar1	-0.1855	0.5942
sma1	-0.6054	0.6126
sma2	-0.2188	0.4712

Table 1: Estimates of the ARIMA(1,1,2) x $(1,1,2)_{12}$ Parameters

The model's AIC = 1794.6

To assess the statistical significance of these parameters, we will conduct hypothesis tests. The null hypothesis (H_o) for each parameter is that it equals to zero implying no effect on the model. The alternative hypothesis (H_1) is that the parameter is not equal to zero.

A common test statistic is the absolute value of the parameter estimate divided by its standard error. We reject H_o if this value, shown in Table 2 is greater than a critical value (typically 2)

Parameter	Test Statistic
ar1 (Φ1)	8.5812
ma1 (Θ1)	11.7930
ma2 (Θ2)	5.8787
sar1	0.3121
sma1	0.9882
sma2	0.4643

Table 2: Test Statistic for ARIMA(1,1,2) x $(1,1,2)_{12}$ Parameters

The test statistics in Table 2 suggest that the season AR(1), seasonal MA(1), and seasonal MA(2) parameters are statistically insignificant (we fail to reject the null hypothesis for these parameters). While dropping both the seasonal MA(1) and seasonal MA(2) might be tempting, a more cautious approach is to remove the higher order term first (seasonal MA(2)).

Therefore, we will fit an ARIMA(1,1,2) x $(0,1,1)_{12}$ (excluding the seasonal AR(1) and the seasonal MA(2) terms in the model) to the housing price data.

Fitting an ARIMA(1,1,2) x $(0,1,1)_{12}$ Model

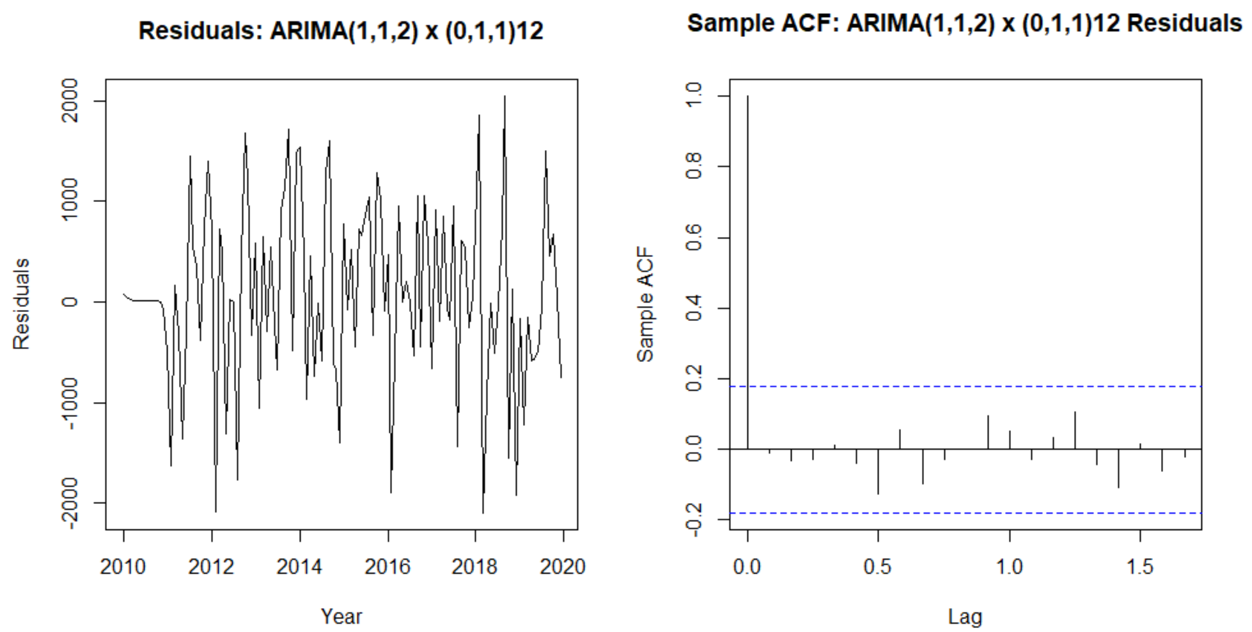

Figure 23: Time and Sample ACF plots of the ARIMA (1,1,2) x $(0,1,1)_{12}$ Residuals

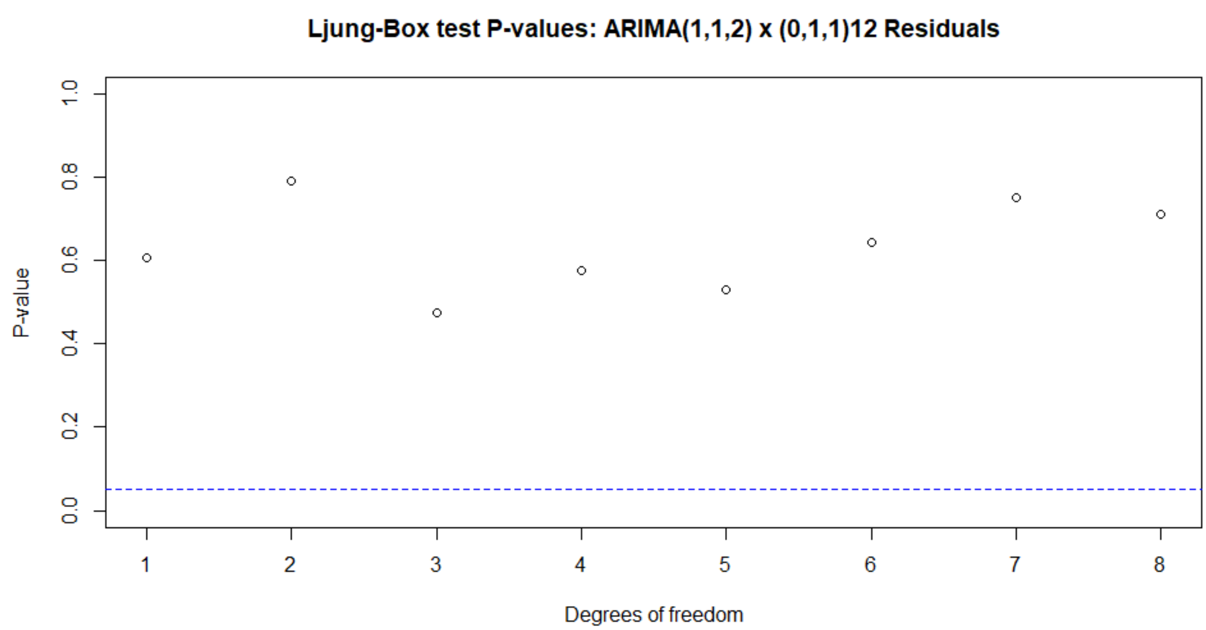

Figure 24: : Ljung-Box test for the ARIMA (1,1,2) x $(0,1,1)_{12}$ Residuals

The ARIMA(1,1,2) x $(0,1,1)_{12}$ adequately captures the housing price data. The residuals exhibit stationarity (Figure 23) and the ACF appears uncorrelated (Figure 23). This Ljung-Box test (Figure 24) reinforces as it fails to reject the null hypothesis at any lag (p-value > 0.05 for all lags). This implies there is no statistically significant autocorrelation in the residuals, consistent with the characteristics of a white noise process.

The ARIMA(1,1,2) x $(0,1,1)_{12}$ model fitted to the housing price data yielded parameter estimates presented in Table 3 below:

Parameter	Estimate	Standard Error of Estimate
ar1 (Φ1)	0.8550	0.0930
ma1 (Θ1)	-1.2235	0.0996
ma2 (Θ2)	0.5234	0.0901
sma1	-0.8109	0.1337

Table 3: Estimates of the ARIMA(1,1,2) x $(0,1,1)_{12}$ Parameters

The model's AIC is 1790.89.

Using our previous approach of using hypothesis tests to assess parameter significance, Table 4 presents the parameters and their corresponding test statistics.

Parameter	Test Statistic
ar1 (Φ1)	9.1935
ma1 (Θ1)	12.2841
ma2 (Θ2)	5.8091
sma1	6.0650

Table 4: Test Statistic for ARIMA(1,1,2) x $(0,1,1)_{12}$ Parameters

The test statistics for the parameters in Table 4 exceed the critical value (often 2), suggesting that all parameters in the ARIMA(1,1,2) x $(0,1,1)_{12}$ model are statistically significant. This implies that they contribute meaningfully to the model's ability to capture the housing price data. Also, the seasonal ma1 component (sma1) that was statistically insignificant in the ARIMA(1,1,2) x (1,1,2)12 is now significant in the ARIMA(1,1,2) x (0,1,1)12 thereby justifying the approach not to drop the component.

Model Selection

Both the ARIMA(1,1,2) x $(1,1,2)_{12}$ and ARIMA(1,1,2) x $(0,1,1)_{12}$ models appear suitable. We will select the most suitable method using the Akaike Information Criterion of both models

The ARIMA(1,1,2) x $(0,1,1)_{12}$ model (1790.89) has the lower AIC value compared to the ARIMA(1,1,2) x $(1,1,2)_{12}$ model (1794.6). Therefore, we will select the ARIMA(1,1,2) x $(0,1,1)_{12}$ as the most suitable model for the housing price data.

Forecasting

Using the selected ARIMA(1,1,2) x $(0,1,1)_{12}$ model , we will forecast average monthly house prices for the first half of 2020 with a 95% confidence level to account for uncertainty in the forecast.

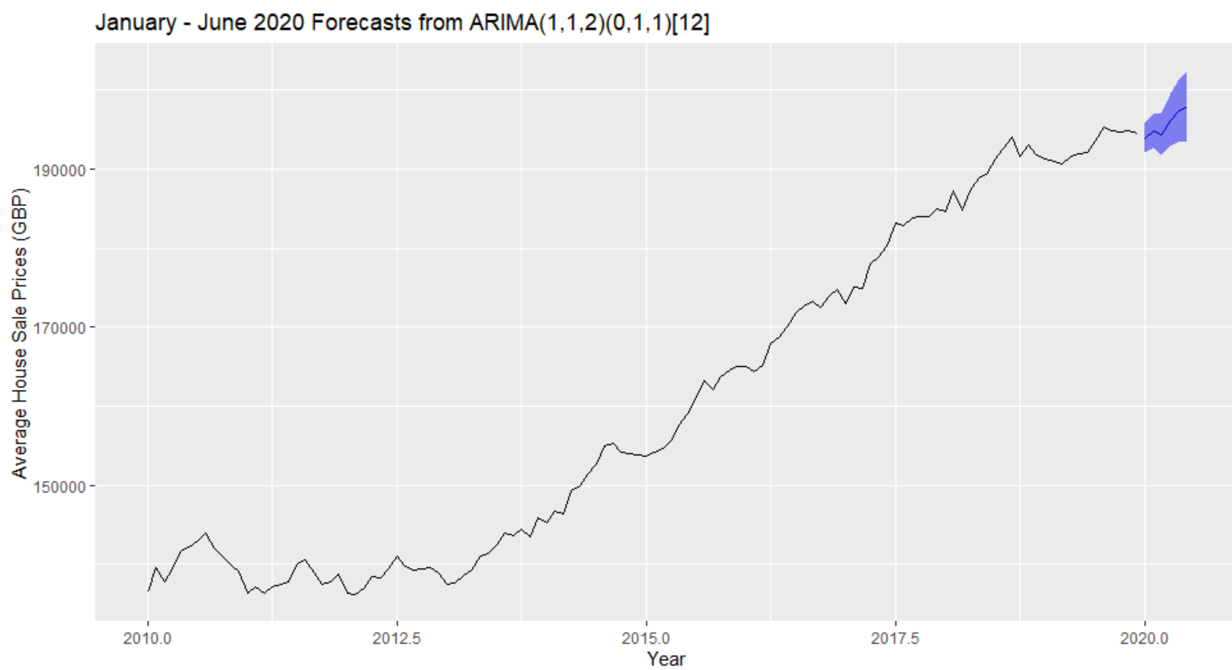

Figure 25: Monthly Housing Prices with Forecast (January – June 2020)

The forecast for January to June 2020 is summarized in the table below.

Month, Year	**Forecast (GBP)**	**Upper Limit of Forecast (GBP)**	**Lower Limit of Forecast (GBP)**
Jan, 2020	193930.5	195770.5	192090.5
Feb, 2020	194836.4	197012.0	192660.9
Mar, 2020	194401.0	197068.7	191733.4
Apr, 2020	196204.1	199462.5	192945.7
May, 2020	197202.1	201106.3	193298.0
Jun, 2020	197933.9	202511.3	193356.6

Table 5: January – June 2020 Housing Price Forecasts and Confidence Intervals

Conclusion

This analysis identified the ARIMA(1,1,2) x $(0,1,1)_{12}$ model as the most effective in capturing the behavior of the average monthly housing price data. Hypothesis tests confirmed statistically significant parameters, and a lower AIC value compared to alternative model support this choice. The model effectively addresses non-stationarity through two differencing steps: the first to remove seasonality and the second to address trend, enabling us to forecast average monthly housing prices for the first half of 2020.

Appendix

R Code

```
#Read in the dataset ‘em_house_prices.csv’

em_house_prices <- read.csv('em_house_prices.csv', header = TRUE)

#Set the dataset to be time series data

em_house_prices_ts  <- ts(em_house_prices$average_price_gbp, start = 2010, frequency = 12)

#Produce a time plot of the time series data ‘em_house_prices_ts’

ts.plot(em_house_prices_ts , ylab = 'Mean House Prices (GBP)', xlab = 'Year', main = 'Monthly Average House Prices')

#Set the R graphics device to contain two plots (1 row, 2 columns)

par(mfrow = c(1,2))

#Plot the sample ACF of the time series data

acf(em_house_prices_ts, ylab = 'Sample ACF', main = 'Sample ACF')

#Plot the sample PACF of the time series data

pacf(em_house_prices_ts, ylab = 'Sample PACF', main = 'Sample PACF')

#Take the first difference (lag 12) of the time series data to remove seasonality

em_house_prices_ts_diff <- diff(em_house_prices_ts, lag = 12)

#Set the R graphics device to contain two plots (1 row, 1 column)

par(mfrow = c(1,1))

#Produce a time plot of the first difference (lag 12) time series data ‘em_house_prices_ts_diff’

ts.plot(em_house_prices_ts_diff, ylab = 'House Price Difference (GBP)', xlab = 'Year', main = 'First Differenced House Prices (Lag 12)')

#Set the R graphics device to contain 1 plot (1 row, 2 columns)

par(mfrow = c(1,2))

#Plot the sample ACF of the first difference time series data

acf(em_house_prices_ts_diff, ylab = 'Sample ACF', main = 'Sample ACF of First Differenced House Prices (Lag 12)')

#Plot the sample PACF of the first difference time series data

pacf(em_house_prices_ts_diff, ylab = 'Sample PACF', main = 'Sample PACF of First Differenced House Prices (Lag 12)')
```

```
#Take the second difference of the time series data

em_house_prices_ts_diff2 <- diff(em_house_prices_ts_diff)

#Set the R graphics device to contain two plots (1 row, 1 column)

par(mfrow = c(1,1))

#Produce a time plot of the second difference time series data ‘em_house_prices_ts_diff2’

ts.plot(em_house_prices_ts_diff2, ylab = 'House Price Difference (GBP)', xlab = 'Year', main = 'Second Differenced House Prices')

#Set the R graphics device to contain 1 plot (1 row, 2 columns)

par(mfrow = c(1,2))

#Plot the sample ACF of the second difference time series data

acf(em_house_prices_ts_diff2, ylab = 'Sample ACF', main = 'Sample ACF of Second Differenced House Prices')

#Plot the sample PACF of the second difference time series data

pacf(em_house_prices_ts_diff2, ylab = 'Sample PACF', main = 'Sample PACF of Second Differenced House Prices')


#Fit an ARIMA(0,1,1) x (0,1,1)12 model to the time series data ‘em_house_prices_ts’

arima011011_model <- arima(em_house_prices_ts, order = c(0,1,1), seasonal = list(order = c(0,1,1), period = 12), method = 'ML')

#Call the arima011011_model ARIMA(0,1,1) x (0,1,1)12 to get it estimated parameters

arima011011_model

#Extract the residuals of the arima011011_model

arima011011_residuals <- residuals(arima011011_model)

#Set the R graphics device to contain 1 plot (1 row, 2 columns)

par(mfrow = c(1,2))

#Produce a time plot of the arima011011_model residuals

ts.plot(arima011011_residuals, ylab = 'Residuals', xlab = 'Year', main = 'Residuals: ARIMA(0,1,1) x (0,1,1)12')
```

#Plot the sample ACF of the arima011011_model residuals

acf(arima011011_residuals, ylab = 'Sample ACF', main = 'Sample ACF: ARIMA(0,1,1) x (0,1,1)$_{12}$ Residuals')

#Declare the Ljung-Box test variable term

LB_test<-function(resid,max.k,p,q){

lb_result<-list()

df<-list()

p_value<-list()

for(i in (p+q+1):max.k){

lb_result[[i]]<-Box.test(resid,lag=i,type=c("Ljung-Box"),fitdf=(p+q))

df[[i]]<-lb_result[[i]]$parameter

p_value[[i]]<-lb_result[[i]]$p.value

}

df<-as.vector(unlist(df))

p_value<-as.vector(unlist(p_value))

test_output<-data.frame(df,p_value)

names(test_output)<-c("deg_freedom","LB_p_value")

return(test_output)

}

#Perform Ljung-Box tests for the arima011011 (ARIMA(0,1,1) x (0,1,1)$_{12}$) model residuals

ARIMA011011.LB <- LB_test(arima011011_residuals, max.k = 11, p = 0, q = 1)

#Set the R graphics device to contain 1 plot (1 row, 1 column)

par(mfrow = c(1,1))

#Produce a plot of P-values against the degrees of freedom and add a blue dashed line at 0.05

plot(ARIMA011011.LB$deg_freedom, ARIMA011011.LB$LB_p_value, xlab = 'Degrees of freedom', ylab = 'P-value', main = 'Ljung-Box test P-values: ARIMA(0,1,1) x (0,1,1)$_{12}$ Residuals', ylim = c(0,1))

abline(h = 0.05, col = 'blue', lty = 2)

#Fit an ARIMA(1,1,0) x (1,1,0)$_{12}$ model to the time series data ‘em_house_prices_ts’

arima110110_model <- arima(em_house_prices_ts, order = c(1,1,0), seasonal = list(order = c(1,1,0), period = 12), method = 'ML')

#Call the arima110110_model (ARIMA(0,1,1) x (0,1,1)$_{12}$) to get it estimated parameters

arima110110_model

#Extract the residuals of the arima110110_model

arima110110_residuals <- residuals(arima110110_model)

#Set the R graphics device to contain 1 plot (1 row, 2 columns)

par(mfrow = c(1,2))

#Produce a time plot of the arima110110_model residuals

ts.plot(arima110110_residuals, ylab = 'Residuals', xlab = 'Year', main = 'Residuals: ARIMA(1,1,0) x (1,1,0)$_{12}$')

#Plot the sample ACF of the arima110110_model residuals

acf(arima110110_residuals, ylab = 'Sample ACF', main = 'Sample ACF: ARIMA(1,1,0) x (1,1,0)$_{12}$ Residuals')

#Perform Ljung-Box tests for the arima110110 (ARIMA(1,1,0) x (1,1,0)$_{12}$) model residuals

ARIMA110110.LB <- LB_test(arima110110_residuals, max.k = 11, p = 1, q = 0)

#Set the R graphics device to contain 1 plot (1 row, 1 column)

par(mfrow = c(1,1))

#Produce a plot of P-values against the degrees of freedom and add a blue dashed line at 0.05

plot(ARIMA110110.LB$deg_freedom, ARIMA110110.LB$LB_p_value, xlab = 'Degrees of freedom', ylab = 'P-value', main = 'Ljung-Box test P-values: ARIMA(1,1,0) x (1,1,0)$_{12}$ Residuals', ylim = c(0,1))

abline(h = 0.05, col = 'blue', lty = 2)

#Fit an ARIMA(1,1,1) x (1,1,1)$_{12}$ model to the time series data ‘em_house_prices_ts’

arima111111_model <- arima(em_house_prices_ts, order = c(1,1,1), seasonal = list(order = c(1,1,1), period = 12), method = 'ML')

#Call the arima111111_model (ARIMA(1,1,1) x $(1,1,1)_{12}$) to get it estimated parameters

arima111111_model

#Extract the residuals of the arima111111_model

arima111111_residuals <- residuals(arima111111_model)

#Set the R graphics device to contain 1 plot (1 row, 2 columns)

par(mfrow = c(1,2))

#Produce a time plot of the arima111111_model residuals

ts.plot(arima111111_residuals, ylab = 'Residuals', xlab = 'Year', main = 'Residuals: ARIMA(1,1,1) x $(1,1,1)_{12}$')

#Plot the sample ACF of the arima111111_model residuals

acf(arima111111_residuals, ylab = 'Sample ACF', main = 'Sample ACF: ARIMA(1,1,1) x $(1,1,1)_{12}$ Residuals')

#Perform Ljung-Box tests for the arima111111 (ARIMA(1,1,1) x $(1,1,1)_{12}$) model residuals

ARIMA111111.LB <- LB_test(arima111111_residuals, max.k = 11, p = 1, q = 1)

#Set the R graphics device to contain 1 plot (1 row, 1 column)

par(mfrow = c(1,1))

#Produce a plot of P-values against the degrees of freedom and add a blue dashed line at 0.05

plot(ARIMA111111.LB$deg_freedom, ARIMA111111.LB$LB_p_value, xlab = 'Degrees of freedom', ylab = 'P-value', main = 'Ljung-Box test P-values: ARIMA(1,1,1) x $(1,1,1)_{12}$ Residuals', ylim = c(0,1))

abline(h = 0.05, col = 'blue', lty = 2)

#Fit an ARIMA(1,1,2) x $(1,1,2)_{12}$ model to the time series data ‘em_house_prices_ts’

arima112112_model <- arima(em_house_prices_ts, order = c(1,1,2), seasonal = list(order = c(1,1,2), period = 12), method = 'ML')

#Call the arima112112_model (ARIMA(1,1,2) x $(1,1,2)_{12}$) to get it estimated parameters

arima112112_model

#Extract the residuals of the arima112112_model

arima112112_residuals <- residuals(arima112112_model)

#Set the R graphics device to contain 1 plot (1 row, 2 columns)

par(mfrow = c(1,2))

#Produce a time plot of the arima112112_model residuals

ts.plot(arima112112_residuals, ylab = 'Residuals', xlab = 'Year', main = 'Residuals: ARIMA(1,1,2) x $(1,1,2)_{12}$')

#Plot the sample ACF of the arima112112_model residuals

acf(arima112112_residuals, ylab = 'Sample ACF', main = 'Sample ACF: ARIMA(1,1,2) x $(1,1,2)_{12}$ Residuals')

#Perform Ljung-Box tests for the arima112112 (ARIMA(1,1,2) x $(1,1,2)_{12}$) model residuals

ARIMA112112.LB <- LB_test(arima112112_residuals, max.k = 11, p = 1, q = 2)

#Set the R graphics device to contain 1 plot (1 row, 1 column)

par(mfrow = c(1,1))

#Produce a plot of P-values against the degrees of freedom and add a blue dashed line at 0.05

plot(ARIMA112112.LB$deg_freedom, ARIMA112112.LB$LB_p_value, xlab = 'Degrees of freedom', ylab = 'P-value', main = 'Ljung-Box test P-values: ARIMA(1,1,2) x $(1,1,2)_{12}$ Residuals', ylim = c(0,1))

abline(h = 0.05, col = 'blue', lty = 2)

#Fit an ARIMA(1,1,2) x $(0,1,1)_{12}$ model to the time series data ‘em_house_prices_ts’

arima112011_model <- arima(em_house_prices_ts, order = c(1,1,2), seasonal = list(order = c(0,1,1), period = 12), method = 'ML')

#Call the arima112011_model (ARIMA(1,1,2) x $(0,1,1)_{12}$) to get it estimated parameters

arima112011_model

#Extract the residuals of the arima112011_model

arima112011_residuals <- residuals(arima112011_model)

#Set the R graphics device to contain 1 plot (1 row, 2 columns)

par(mfrow = c(1,2))

#Produce a time plot of the arima112011_model residuals

ts.plot(arima112011_residuals, ylab = 'Residuals', xlab = 'Year', main = 'Residuals: ARIMA(1,1,2) x $(0,1,1)_{12}$')

```
#Plot the sample ACF of the arima112011_model residuals

acf(arima112011_residuals, ylab = 'Sample ACF', main = 'Sample ACF: ARIMA(1,1,2) x (0,1,1)12 Residuals')

#Perform Ljung-Box tests for the arima112011 (ARIMA(1,1,2) x (0,1,1)12) model residuals

ARIMA112011.LB <- LB_test(arima112011_residuals, max.k = 11, p = 1, q = 2)

#Set the R graphics device to contain 1 plot (1 row, 1 column)

par(mfrow = c(1,1))

#Produce a plot of P-values against the degrees of freedom and add a blue dashed line at 0.05

plot(ARIMA112011.LB$deg_freedom, ARIMA112011.LB$LB_p_value, xlab = 'Degrees of freedom', ylab = 'P-value', main = 'Ljung-Box test P-values: ARIMA(1,1,2) x (0,1,1)12 Residuals', ylim = c(0,1))

abline(h = 0.05, col = 'blue', lty = 2)

#Produce a Forecast for the next 6 months (h = 6)

library(forecast)

forecast_6m <-forecast(em_house_prices_ts, h = 6, model = arima112011_model, level = 95)

#Plot the Forecast

autoplot(forecast_6m, ylab = 'Average House Prices (GBP) ', xlab = 'Year', main = 'January-June 2020 Forecasts from ARIMA(1,1,2)(0,1,1)[12] ')

#Produce average(mean) forecast values

forecast_6m$mean

#Produce upper limit of forecast values

forecast_6m$upper

#Produce lower limit of forecast values

forecast_6m$lower
```