MATH 4022 – Time Series and Forecasting

Coursework Assignment

Spring Semester 2023 -2024

School of Mathematical Sciences

University of Nottingham

Submitted by:

Mohid Ali Gill
20591293

Under the guidance of:

Dr. Aidan O'Keeffe

Submission Date:

29th April 2024

Question 1: Analysis of Annual Mean Temperatures

Introduction

The report investigates and identifies a suitable time series model that effectively describes the dataset 'cet_temp.csv' containing annual mean temperature data from the year 1900 to 2021 for the Midlands region of England which is sourced from the UK Meteorological Office Hadley Climate Center. R studio is used to model the time series. R packages used to perform the analysis are forecast, tseries and Metrics.

Metrics offers functions to compute accuracy metrics such as MAPE (Mean Absolute Percentage Error) and RMSE (Root Mean Squared Error), forecast is used forecasting the model and working with time series, tseries provides tools for time series analysis and unit root tests.

Exploratory Data Analysis

The dataset is loaded and converted into a time series using the ts function. The avg_annual_temp_C column is converted to a time series object, starting from year 1900 with an annual frequency.

The time series plot in Fig 1 provides more insights.

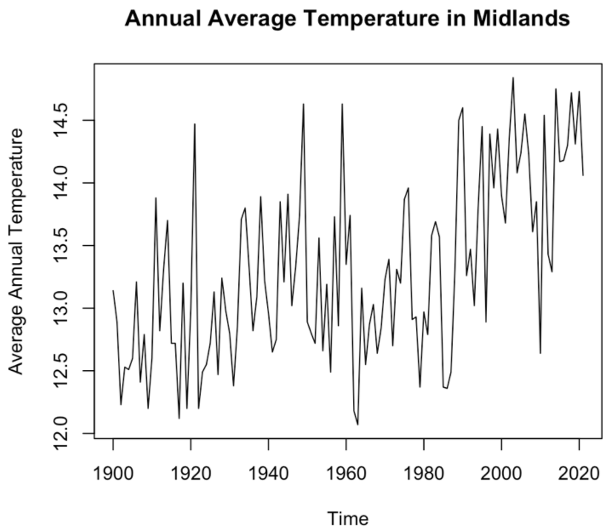


Fig 1: Time series plot

A slight upward trend is seen in the time series plot, but it seems to be very little. The series looks stationary as no periodic patterns or seasonality is seen. To confirm the stationarity of series a few statistical tests are performed.

The sample ACF is plotted in Fig 2 which helps us determine if the autoregressive moving average (ARMA) model is sutible for the time series. It shows us the number of past period data points which influence the current data point.

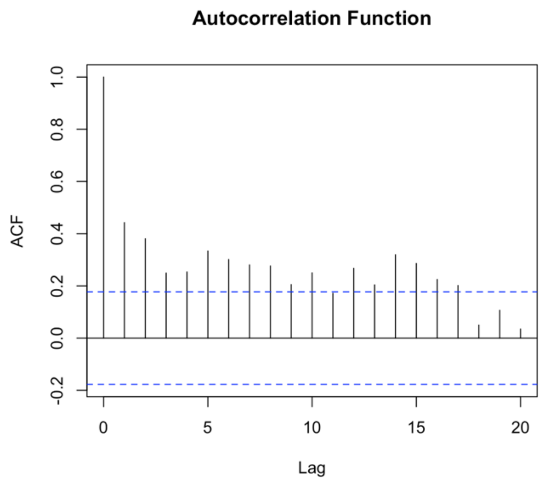


Fig 2: ACF plot

In this plot the sample ACF is gradually decaying which tells us that AR model can be a good fit and it has significant values that are outside the confidence interval till lag 17. The stationarity of the model is yet to be confirmed, statistical tests will be performed later for that.

The sample PACF is plotted in Fig 3 which helps us determine the order of the AR part of model. An instant cut-off in the PACF after a few number of lags shows the number of lag after which the values are not directly correlated to the new values.

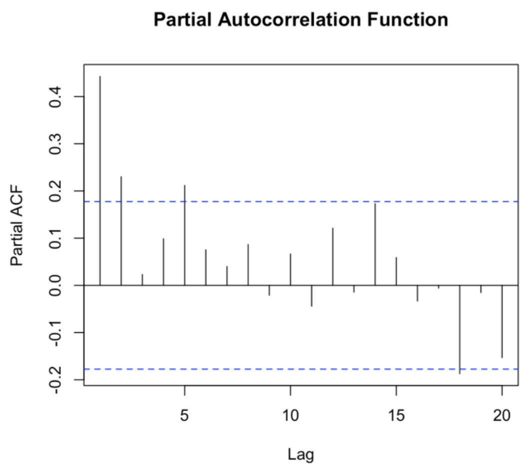


Fig 3. PACF plot

As per the plot the sample PACF cuts-off after the 2^{nd} lag suggesting that an AR(2) model can be a good fit for the time series. However, in the plot some significant values can be seen at different lags, we can assume that they are due to some sampling errors.

Together the ACF and PACF plots guide us in choosing the right paratemers for the time series model which is AR(2) in our case, ensuring that model is not overfitting neither underfitting.

Stationarity Test

To test the stationarity of the time series Phillips-Perron (PP) test and Augmented Dickey-Fuller (ADF) tests are performed which aims to test for a unit root in the time series.

To perform the PP test we will first decide the null hypothesis and alternate hypothesis.

H_0 = Time series is not stationary
H_1 = Time series is stationary

The test was performed in R using PP.test() and the result is in Fig 4

```
	Phillips-Perron Unit Root Test

data:  temperature_series
Dickey-Fuller = -8.9751, Truncation lag parameter = 4, p-value = 0.01
```

Fig 4. PP test result

Test statistic value = -8.9751
p-value = 0.01

As the p-value is less than 0.05 we will reject the null hypothesis at 5% level of significance and accepting that the time series is stationary

To furthur confirm the stationarity we will perform the ADF test with the same null and alternate hypothesis as the PP test. To perform the test adf.test() in R is used, the result is in Fig 5

```
	Augmented Dickey-Fuller Test

data:  temperature_series
Dickey-Fuller = -3.7718, Lag order = 4, p-value = 0.0227
alternative hypothesis: stationary
```

Fig 5. ADF test result

Test statistic value = -3.7718
p-value = 0.0227

The p-value for the ADF test is also less than 0.05 which rejects the null hypothesis and both the results prove that time series is stationary.

Model Fitting

For modelling and choosing the right AR model, models with different parameters are made and their AIC values are compared to choose the one with least AIC value. The trend in AIC values is found it reduces till AR(5) and then again starts to increase as seen in Fig 6. As AIC is a very good basis for comparison we choose the AR(5) model.

```
[1] "AR(3) AIC: 241.77"
[1] "AR(4) AIC: 242.11"
[1] "AR(5) AIC: 237.19"
[1] "AR(6) AIC: 237.75"
[1] "AR(7) AIC: 239.14"
```

Fig 6. AIC values for AR models

AR(5) model is fit in R and a summary for it produced which is in Fig 7

```
Call:
arima(x = temperature_series, order = c(5, 0, 0))

Coefficients:
         ar1     ar2      ar3     ar4     ar5  intercept
      0.2957  0.2075  -0.0411  0.0520  0.2388    13.3374
s.e.  0.0873  0.0914   0.0936  0.0921  0.0891     0.2081

sigma^2 estimated as 0.3625:  log likelihood = -111.59,  aic = 237.19

Training set error measures:
                     ME      RMSE       MAE        MPE     MAPE      MASE        ACF1
Training set 0.01135916 0.6020975 0.4896441 -0.1178318 3.668807 0.8102699 -0.02240008
```

Fig 7. Summary of AR(5) model

The equation for the fitted model is as follows:

$$\mathbf{X_t = 13.3374 + 0.2957X_{t-1} + 0.2075X_{t-2} - 0.0411X_{t-3} + 0.0520X_{t-4} + 0.2388X_{t-5} + et}$$

Where et = N(0, 0.3625)

Error Analysis

Error analysis is performed to make sure fitted model abides by the ARIMA model assumptions which are average error should be 0, errors should follow normal distribution and comparison of theoratical and sample quantiles
The mean of residuals as seen in Fig 8 is 0.01 which is very close to 0

```
> print(sprintf('Mean of residuals: %.4f', mean_res))
[1] "Mean of residuals: 0.0114"
```

Fig 8. Mean of residuals

To check if they follow a normal distribution sample density is plotted represented by histograms and line graphs, normal distribution is then plotted on it using the sample mean and standard deviation we get a good fit for normal distribution as seen in Fig 9, it has minor deviations from the perfect curve but for small sample size it can be ignored

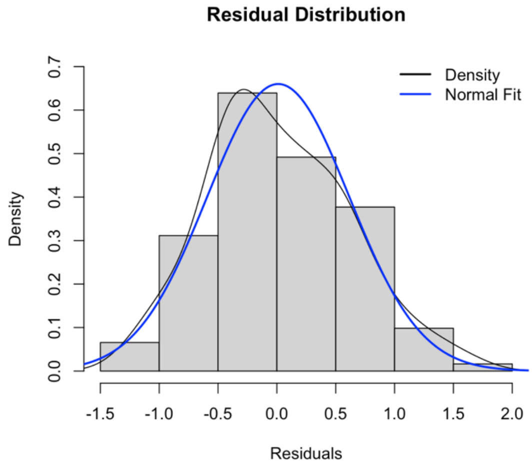


Fig 9. Residual distribution

To compare the theoratical and sample quantiles. A QQ-scatter plot is plotted in Fig 10, In the best case scenerio the scatter points should be as close to the diagonal line as possible, which is true in this case hence the errors follow normal distribution

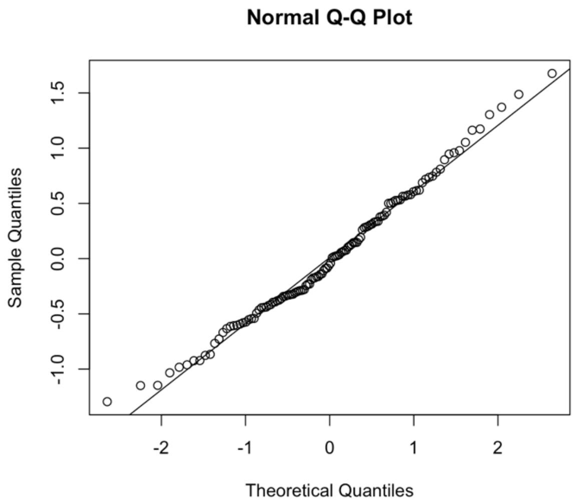


Fig 10. QQ plot

Model Evaluation

To determine the model accuracy the ARIMA model is plotted on the original time series in Fig 11 and check how well it captures it

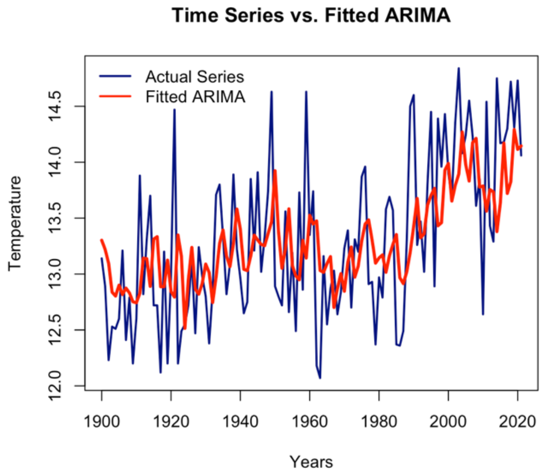


Fig 11. ARIMA plotted on time series

The ARIMA model (red line) captures the trend well but does not capture the magnitudes the peaks are not close to the orignal time series

Furthur we calculate Mean Average Percentage Error (MAPE) which measures the accuracy of the forecasting in percentage making it easier to understand as the value is comparable to the original magnitude, we got the MAPE to be 3.67%. The formula to calculate MAPE is as follows

$$MAPE = \frac{1}{n} \times \sum \left| \frac{actual\ value\ - forecast\ value}{actual\ value} \right|$$

Root Mean Squared Error (RMSE) measures the standard deviation of the residuals from the original line, it tells us how close the data is to the line of best fit, the RMSE value we got for the model is 0.6. The formula to calculate RMSE is as follows

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}$$

$$\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n \text{ are predicted values}$$
$$y_1, y_2, \ldots, y_n \text{ are observed values}$$
$$n \text{ is the number of observations}$$

In conclusion the analysis of annual mean temperatures from 1900 to 2021 using the AR(5) model captured the trends in the dataset effectively, the model was validated through different stationarity checks and was provided a good model to understand the dataset

Appendix A – Question 1 Code

```
# Required package loading
library(forecast)
library(tseries)
library(Metrics)

# Data loading and conversion
temperature_data <- read.csv('cet_temp.csv')

# Converting dataframe column to a time series object
temperature_series <- ts(temperature_data$avg_annual_temp_C, start=1900,
frequency=1)

# Time Series Visualization
# (1) Plotting the time series
plot(temperature_series, ylab='Average Annual Temperature', main="Annual Average
Temperature in Midlands")

# (2) Autocorrelation function
acf(temperature_series, main='Autocorrelation Function')

# (3) Partial autocorrelation function
pacf(temperature_series, main='Partial Autocorrelation Function')

# Stationarity Tests
# (1) Philips-Perron Test
PP.test(temperature_series)

# (2) Augmented Dickey-Fuller test
adf.test(temperature_series)

# Model Estimation using ARIMA
for(order in 3:7){
  model_ar <- arima(temperature_series, order=c(order,0,0))
  print(sprintf('AR(%d) AIC: %.2f', order, model_ar$aic))
}

# Choosing AR(5) based on AIC
best_model <- arima(temperature_series, order=c(5,0,0))
summary(best_model)

# Analysis of model residuals
# (1) Mean of residuals
mean_res <- mean(best_model$residuals, na.rm=TRUE)
print(sprintf('Mean of residuals: %.4f', mean_res))

# (2) Residuals distribution
hist(best_model$residuals, freq=FALSE, ylim=c(0, 0.7), xlab='Residuals',
main='Residual Distribution')
```

```
res_density <- density(best_model$residuals, na.rm=TRUE)
lines(res_density)
normal_curve <- dnorm(res_density$x, mean=mean_res, sd=sd(best_model$residuals,
na.rm=TRUE))
lines(res_density$x, normal_curve, col='blue', lwd=2)
legend('topright', legend=c('Density', 'Normal Fit'), col=c('black', 'blue'), lwd=2,
bty='n')

# (3) Quantile-Quantile plot
qqnorm(best_model$residuals)
qqline(best_model$residuals)

# Model accuracy assessment
# (1) Error metrics
mape_value <- mape(as.numeric(temperature_series), fitted(best_model))
rmse_value <- rmse(as.numeric(temperature_series), fitted(best_model))
print(sprintf('MAPE: %.2f%%', mape_value * 100))
print(sprintf('RMSE: %.2f', rmse_value))

# (2) Overlay of Actual and Fitted Time Series
ts.plot(temperature_series, col='navy', lwd=2, main='Time Series vs. Fitted ARIMA',
ylab='Temperature', xlab='Years')
lines(fitted(best_model), col='red', lwd=3)
legend('topleft', legend=c('Actual Series', 'Fitted ARIMA'), col=c('navy', 'red'), lwd=2,
bty='n')
```

Executive Summary: Forecasting Monthly House Prices in East Midlands

Overview of Analysis:

The analysis focuses on the dataset which contains the average monthly house prices from January 2010 to December 2019 in the East Midlands. The aim of the study is to predict the house prices in first half of 2020. Statistical techniques were used to ensure that the forecast is reliable and accurate providing essential insights for planning and development.

Key Findings:

Trend Identification: The time series showed an upward trend in the house prices over the decade. No seasonal variations were found suggesting that the increase in price was trend driven and not seasonal.

Stationarity and Model Selection: The tests showed that the time series is non-stationary as it showed trends over time. To make the series stationary differencing technique was applied which helped stabilize the mean of series which prepared the data for accurate modelling using the seasonal ARIMA.

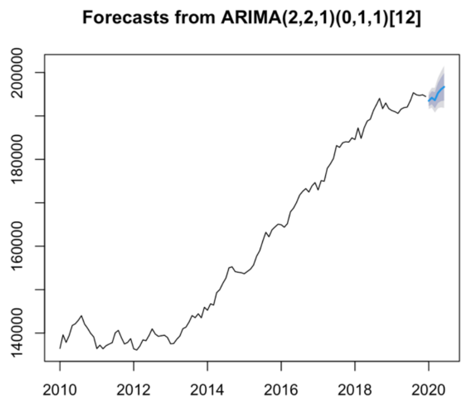
SARIMA Model: The best fitting model was the SARIMA(2,2,1)(0,1,1)[12]. It is a seasonal autoregressive integrated moving average model; the first part of the model is the non-seasonal part followed by the seasonal part and lastly the period of seasonality. It is designed to capture both the seasonal and non-seasonal parts of the data. The model helped in forecasting the future house prices for the first half of the next year helping the developers and government in making the right decisions.

Forecast Accuracy: The model provided good performance in predicting the values with error metrics such as the Root Mean Square Error and Mean Absolute Percentage Error within the acceptable ranges. The metrics helps decide that the predicted values are close to the actual values giving a reliable base for future forecasting.

Practical Implications: The forecast suggests that the upward trend is continued in the house prices. This information is important for the government and agencies responsible in dealing with the housing policies and economic development as it helps in budgeting and resource allocation.

Conclusion

The analysis provides a good tool for predicting the future trends in the housing market of East Midlands. These predictions can help in making better policies and decisions in housing policies, time series helps in a way that it does not only forecast on previous and present values but takes in account all the seasonal fluctuations and long-term trends. The forecasted data with the original data is as follows



Question 2: Forecasting Monthly House Prices

Introduction

The report caters the 'em_house_prices.csv' dataset, which contains data for average monthly house prices from January 2010 to December 2019 in the East Midlands. The dataset contains 120 data points which provides insight into the trends in house prices over the decade. The aim of analysis is to forecast house prices for the first half of the year 2020 by introducing an applicable and sutible time series model, thereby helping the stakeholders in strategic planning and investment decisions.

Data Handling and Exploratory Data Analysis

The time series is plotted in Fig 12 for the dataset to check for any visual trends and seasonalities.

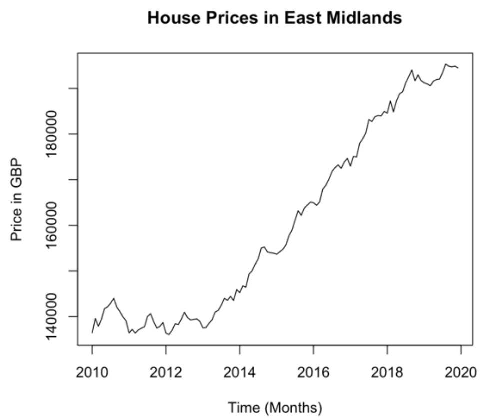


Fig 12. Time series plot

Inspecting the plot carefully shows a clear increasing trend with potiential long term growth in the house prices no evidence of seasonality is found.

To distinguish between the underlying trend, seasonality and the noise in the data a decompositional analysis is performed which helps understand the different factors which have an effect on the time series therefore selecting the appropirate forecasting model. The technique dissects the time series into different components such as trend, seasonal and residual components. The equation can be represented as:

$$\mathbf{Y_t = T_t + S_t + R_t}$$

Where Y_t is the observed series, T_t is the trend component, S_T is the seasonal component and R_t is the residual component

As per Fig 13 the variance for Seasonal and Remainder are very similar which means that there could be no seasonality in the data and it could just be randomness that feels like it is seasonal, both variances range between -2000 and 2000.

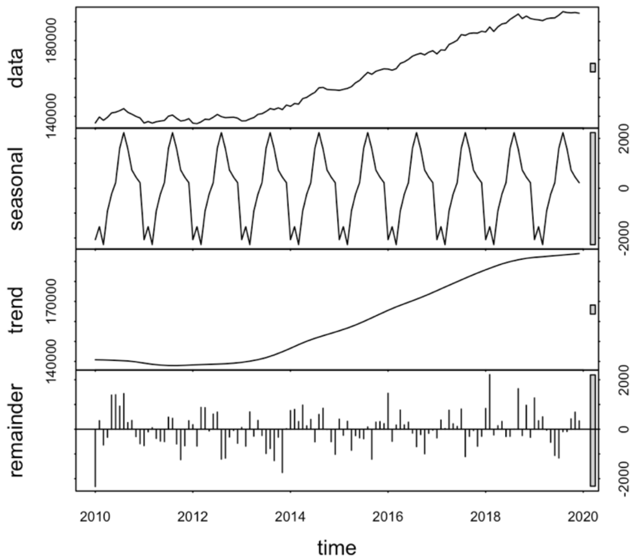


Fig 13. Decomposition plot

The sample ACF and PACF are plotted in Fig 14 to choose the right model with parameters

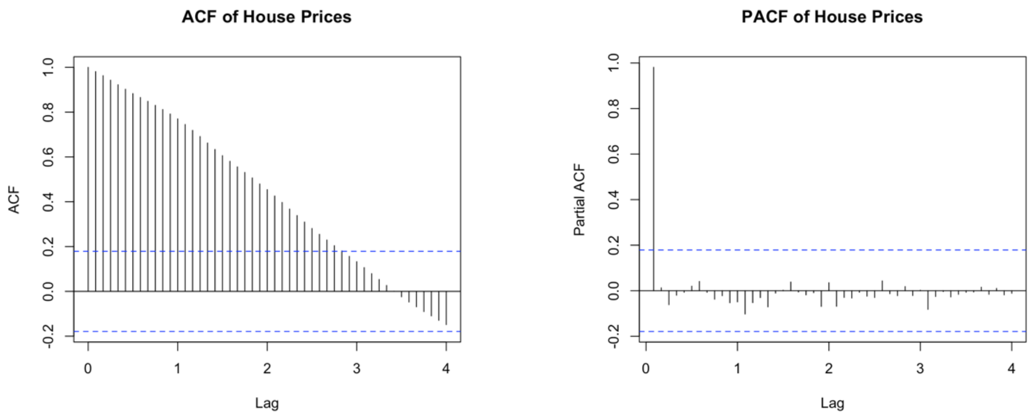


Fig 14. Plots fot ACF and PACF

The sample ACF exponentially decreases which shows that the series is non stationary, It may require differencing to make it stationary. As the PACF plot cut offs after the 1st lag it can be assumed that AR(1) model can be a good fit for the time series but needs more study.

Stationarity Test and Differencing

A time series is said to be stationary if its mean, variance and other statististical properties don't change over time to test the stationarity Phillips-Perron (PP) test is performed which checks if the unit root is present in the time series

$$\boldsymbol{\Delta yt = \alpha + \beta t + zyt - 1 + \epsilon t}$$

where Δyt is the first difference of the series, α is a constant, βt is the trend and z is the coefficient on the lagged variable

The null hypothesis is that the time series is stationary and otherwise is the alternate hypthesis.

The results from the test give test statistic value of -1.9266 and p-value of 0.607 which is greater than 5% hence we do not find enough evidence to reject the null hypothesis at 5% of confidence level thus the time series is not stationary.

Differencing helps in stabilizing the mean of time series by subtracting the previous data point from the current data point thus helping the time series to become stationary. The formula used for one oeder differencing is as follows.

$$Y'_t = Y_t - Y_{t-1}$$

To obtain the optimal order time series is differenced different orders and the variance is recorded where ever the variance starts to increase is chosen as the optimal order for differencing

```
[1] "Order = 1 Variance = 1616139.25409486"
[1] "Order = 2 Variance = 3350632.57974794"
[1] "Order = 3 Variance = 10584433.3953728"
[1] "Order = 4 Variance = 36014938.599925"
[1] "Order = 5 Variance = 128517445.55042"
```

It is observed that the variance increases after the 1st order hence one order differencing will be used to make the time series stationary. The differenced series is plotted in Fig 15

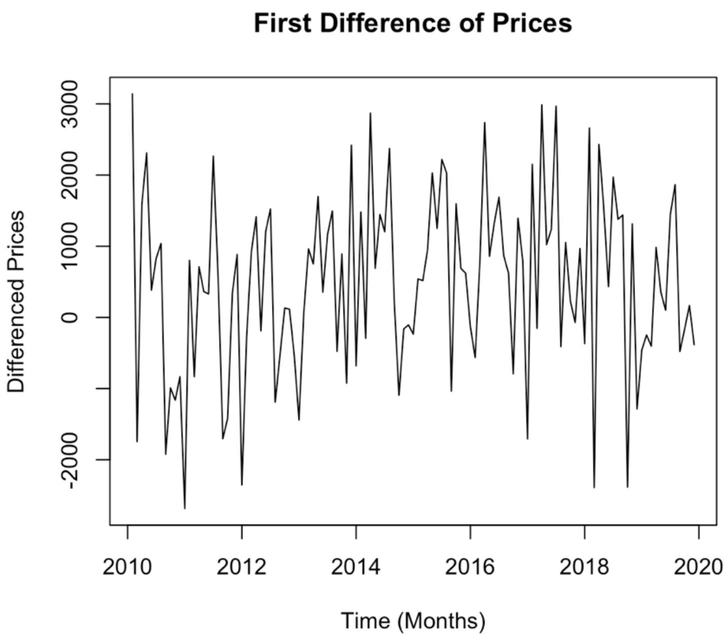


Fig 15. 1st order differenced time series

Visualising the plot the differenced time series looks stationary as no clear seasonality or trends are visible but to confirm the stationaity PP test is performed again on the differenced time series. If the results show the time series to still be non stationary then seasonal differencing would have to be applied.

Performing the PP test with the same null and alternate hypthesis as before on the differenced time series a test statistic value of -11.703 and p-value of 0.01 is obtained. As the p-value is less than 0.05, the null hypothesis can be rejected with 5% significance and the series is said to be stationary hence no need to perform seasonal differencing and seasonal autoregressive (SARIMA) model can be chosen.

Plotting the ACF and PACF of the differenced time series in Fig 16 shows that ACF has some significant values at lag 12, 24, 36 and so on which refers to each year providing evidence the model to fit must have a seasonal component whereas, PACF cuts off indicating that the AR model has to be used

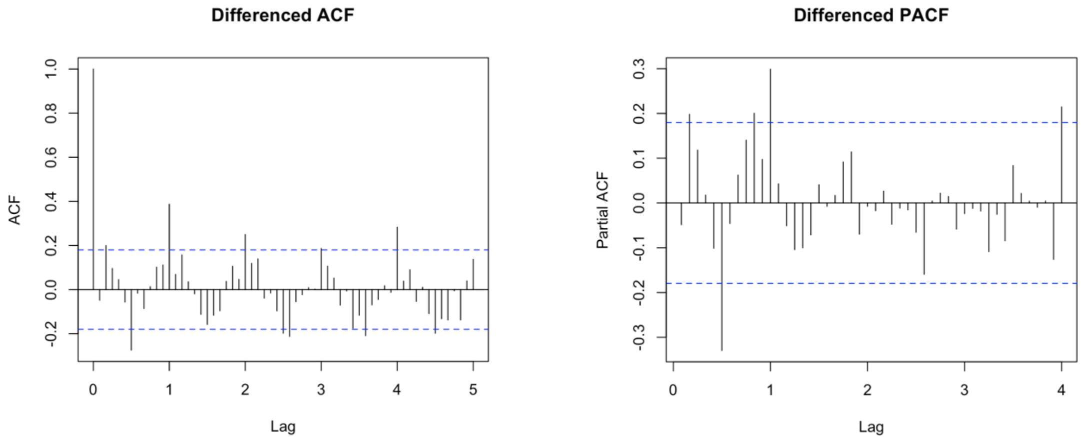


Fig 16. Differenced ACF and PACF

Model Fitting

Based on the autocorrelation and partial correlation analysis it is evident to use the seasonal ARIMA model as it extends the ARIMA model to also take into account the seasonality. The equation of the model is as follows.

$$(1-\phi_1 B-\phi_2 B_2)(1-B)(1-\Phi_1 B_{12})(1-B_{12})X_t=(1+\theta_1 B)(1+\Theta_1 B_{12})\epsilon_t$$

Where B is the backshift operator, ϕ_1 and ϕ_2 are the parameters for the non seasonal AR terms, θ_1 is the parameter for the non seasonal MA term and Θ is the seasonal MA term at lag 12.

To choose the right parameters for the SARIMA model, different parameters are passed to the model and the AIC value is returned. The AIC values are compared and the model with the least AIC value is then chosen .

After changing multiple parameters the model with parameters SARIMA(2,2,1)(0,1,1)[12] gave the lowest AIC which was 1804.37.

The fitted model summary is in Fig 17

```
Call:
arima(x = first_diff, order = c(2, 2, 1), seasonal = list(order = c(0, 1, 1),
    period = 12))

Coefficients:
          ar1      ar2      ma1     sma1
      -0.9921  -0.4776  -0.9993  -0.8773
s.e.   0.0881   0.0872   0.0387   0.1679

sigma^2 estimated as 1141715:  log likelihood = -897.18,  aic = 1804.37

Training set error measures:
                    ME     RMSE      MAE     MPE     MAPE      MASE       ACF1
Training set -87.11505 1003.691 760.3948 59.6435 156.107 0.5394492 -0.1745607
```

Fig 17. Summary of SARIMA model

Using the equation for the SARIMA model and the coefficients generated by the model we can rearrange the equation to be

$$(1 + 0.9921B + 0.4776B_2)(1 - B)(1 - B_{12})X_t = (1 - 0.9993B)(1 - 0.8773B_{12})\, \epsilon_t$$

Error Analysis and Model Performance

To confirm if the fitted model follows the ARIMA model assumptions error analysis is performed. Analysing the residuals helps in verifying the assumption that errors are distributed normally and if they do not follow normal distribution and shows some patterns it can be said that there is some information missing in the model. The aim is to have residuals resemble white noise.
Firstly a time series diagnosis plot (Fig 18) is plotted

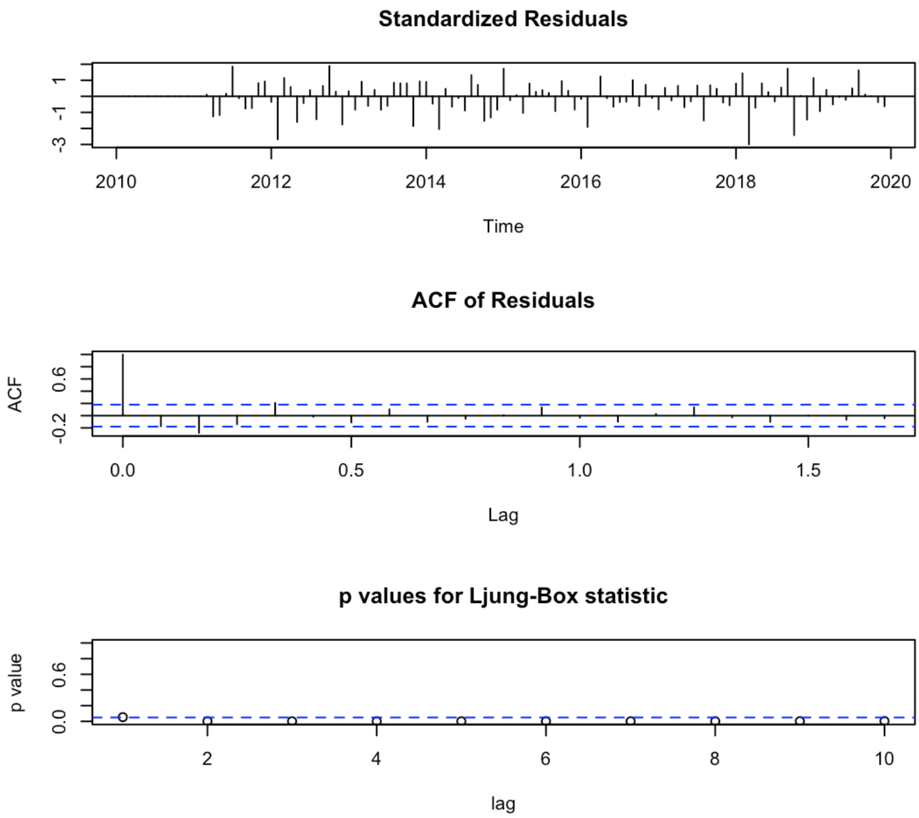


Fig 18. Time series diagnosis plot

From the plot it can be seen that no clear patterns are visible in the standardized residuals which is expected as it could be due to the white noise present in the time series. Looking at the plot for ACF of residuals there are no significant values above the confidence level after the 1st lag which provides evidence that the errors are not correlated which can again be due to the white noise present. Finally, using the Ljung box statistic we can test for white noise presense having the H_0 to be errors have white noise. The statistic has almost all the value above 5% hence we can reject the null hypothesis proving that the errors contain white noise.

The sample residuals are plotted in histograms and line graph to check if it follows normal distribution. The plot is in Fig 19

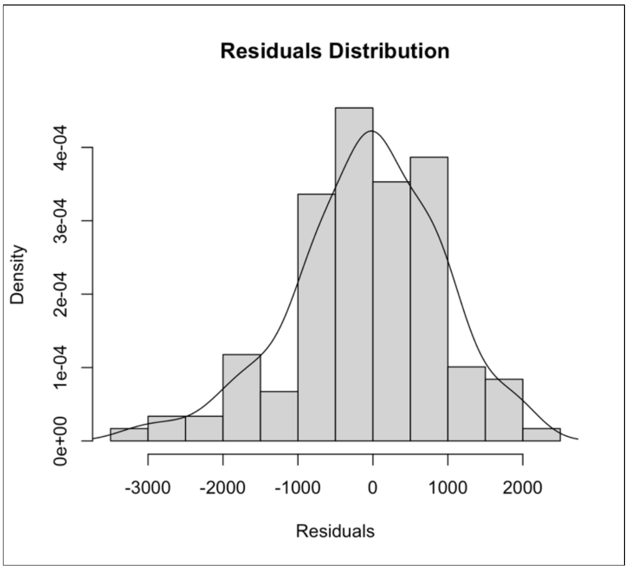


Fig 19. Residual distribution plot

From the plot it can be visualised that the residuals take the shape of a bell curve indicating the errors are normally distributed. The curve is not perfect it has a few deviations but that is acceptable with the small sample size provided.

To check how close is the sample to the theoratical quantiles a QQ plot is plotted in Fig 20 the scatter should be as close to the diagonal line as possible. In this case the residuals are close to the diagonal line with some deviations which can be ignored.

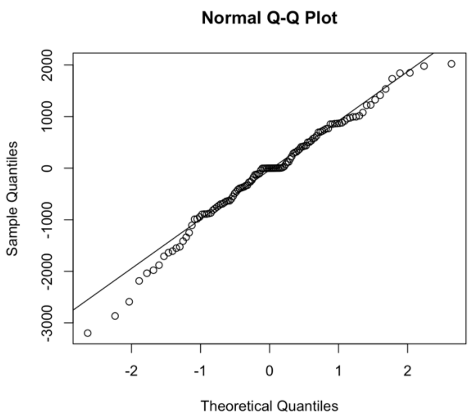


Fig 20. QQ plot

To evalute the model furthur model performance is tested, RMSE and MAPE are calculated using the following formulas.

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{t=1}^{n}(\hat{y}_t - y_t)^2}{n}} \qquad \mathrm{MAPE} = \frac{100\%}{n}\sum_{t=1}^{n}\left|\frac{y_t - \hat{y}_t}{y_t}\right|$$

The error values generated from the model are 1003.691 and 156.107 respectively.

To furthur evaluate the model a plot of fitted vs the actual values is plotted in Fig 21

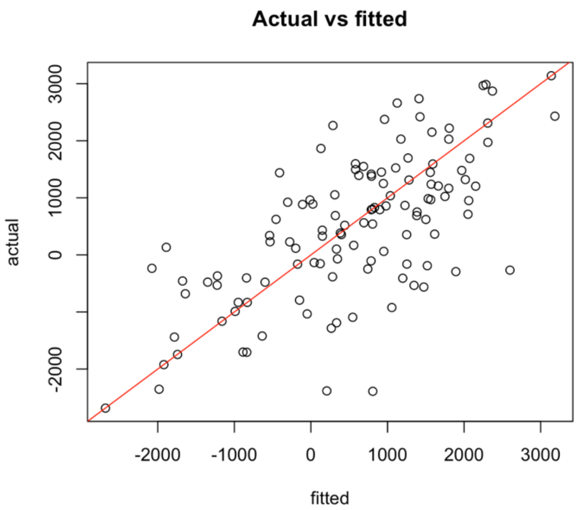


Fig 21. Scatter plot of Actual vs Fitted values

The plot shows that the values are quite scattered but most of the values are near the diagonal line, it can not be called a perfect model but is a good fit, a better judgement can be made from the numerical metrics calculated before.

Forecasting and Interpretation

Forecasting uses the model to predict the values in future based on the previous values. It is a primary objective of time series analysis which is used to predict how the series will behave in the future. Here forecasting will help local government and developers in properly planning and budgeting. The confidence intervals in forecasting provide a range of likely outcomes, adding risk assessment to the predictions.

The forecasting can be done with differenced series but it'll require the results to be converted to the original units, otherwise, the SARIMA(2,2,1)(0,1,1)[12] model can be directly modeled on the original data and forecasted, it will remove the need of extra steps to convert differenced series to original units.

The plot for differenced time series is as follows

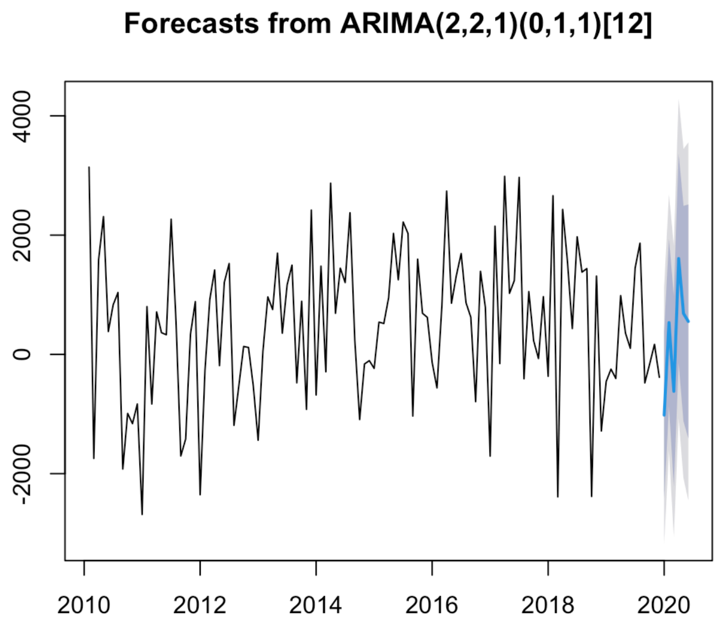


Fig 22. Forecasted differenced series

The same was modelled on the original data and forecasted the plot for that is as follows

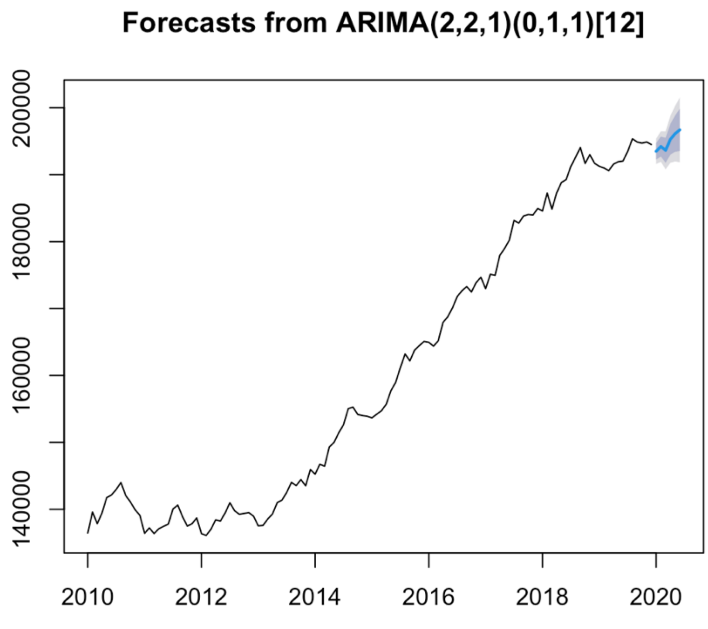


Fig 23. Forecasted on original data

The analysis provides a solid foundation for predicting the house prices in the East Midlands region of UK, using statistical and techniques and models it ensures the accuracy and reliablity in the predictions. It will help the government and developers to get close to accurate predictions for the house prices

Appendix B – Question 2 Code

```
# Load the necessary package for time series analysis
library(forecast)

# Read the dataset
house_prices = read.csv('em_house_prices.csv')

# Create a time series
house_prices_ts = ts(house_prices$average_price_gbp, start = c(2010, 1), frequency = 12)

# Plot the time series
ts.plot(house_prices_ts, main='House Prices in East Midlands', xlab='Time (Months)',
ylab='Price in GBP')

# Decompose the time series to analyze components
decomposed = stl(house_prices_ts, s.window = 'periodic')
plot(decomposed)
range(decomposed$time.series[,'seasonal'])
range(decomposed$time.series[,'remainder'])

# Analyze the autocorrelation of the time series
acf(house_prices_ts, lag.max = 48, main='ACF of House Prices')

pacf(house_prices_ts, lag.max = 48, main='PACF of House Prices')

# Test for stationarity
stationarity_test = PP.test(house_prices_ts)
print(stationarity_test)

# Determine optimal differencing level
for(d in 1:5){
  diff_ts = diff(house_prices_ts, differences = d)
  print(paste('Order =', d, 'Variance =', var(diff_ts)))
}
var(house_prices_ts)

# Plot the first differenced series
first_diff = diff(house_prices_ts)
ts.plot(first_diff, main='First Difference of Prices', xlab='Time (Months)', ylab='Differenced
Prices')

# Analyze the differenced series
acf(first_diff, lag.max = 60, main='Differenced ACF')
abline(v=c(12,24,36,48), col='red', lty=2)

pacf(first_diff, lag.max = 48, main='Differenced PACF')

# Test stationarity again
stationarity_test_diff = PP.test(first_diff)
```

```
cat('P-value for PP test after differencing:', stationarity_test_diff$p.value, '\n')
print (stationarity_test_diff)

# Fit a SARIMA model
sarima_model = arima(first_diff, order = c(2, 2, 1), seasonal = list(order = c(0, 1, 1), period =
12))
print(paste0('SARIMA(2,2,1)(0,1,1)[12] model, AIC = ', round(sarima_model$aic, 2)))

# Select and summarize the model
final_model = arima(first_diff, order = c(2, 2, 1), seasonal = list(order = c(0, 1, 1), period =
12))
summary(final_model)

# Diagnostic plots for the selected model
tsdiag(final_model)

# Distribution of residuals
hist(final_model$residuals, freq = FALSE, main='Residuals Distribution', xlab='Residuals')
lines(density(final_model$resid, na.rm = TRUE))

# Q-Q plot of residuals
qqnorm(final_model$residuals)
qqline(final_model$residuals)

# Evaluate model performance
summary(final_model)  # Extract MAE and RMSE1

# Plot actual vs. fitted values
plot(as.numeric(fitted(final_model)), as.numeric(first_diff), main='Actual vs
fitted',xlab='fitted',ylab='actual')
abline(a=0, b=1, col='red')

# Forecast future values
forecasted_values = forecast(final_model, 6)
plot(forecasted_values)

# Re-fit the model on non-differenced data and forecast
original_model = arima(house_prices_ts, order = c(1, 1, 0), seasonal = list(order = c(2, 0, 0),
period = 12))
forecast_original = forecast(original_model, 6)
plot(forecast_original)
```