MODELLING The FUTURE OF AUSTRALIA'S BIOTECH SECTOR

MATH1318 Time Series Analysis

Michael Ditiatkovski s3770701

Benjamin Huang s3383866

Tanmay Nagi s3753836

Yinan Chris Zhang S3789428

# Contents

[Contents 1](#_Toc42535737)

[Table of Figures: 2](#_Toc42535738)

[Executive Summary 3](#_Toc42535739)

[1. Introduction 4](#_Toc42535740)

[1.1. Background 4](#_Toc42535741)

[1.1. Aim 4](#_Toc42535742)

[1.2. Data 4](#_Toc42535743)

[2. Methodology 5](#_Toc42535744)

[2.1. Time Series Analysis 5](#_Toc42535745)

[2.2. Deterministic Modelling 5](#_Toc42535746)

[2.3. Stochastic Model Identification 5](#_Toc42535747)

[2.4. Stochastic Model Fitting and Selection 5](#_Toc42535748)

[2.5. Residual and Coefficient Analysis 6](#_Toc42535749)

[2.6. Forecasting 6](#_Toc42535750)

[3. Analysis of Data 7](#_Toc42535751)

[4. Deterministic Modelling 10](#_Toc42535752)

[4.1. Summary for Deterministic Model 13](#_Toc42535753)

[5. Stochastic Modelling 14](#_Toc42535754)

[5.1. Determine Seasonality 14](#_Toc42535755)

[5.1. Model Selection 16](#_Toc42535756)

[5.1.1. SARIMA Seasonal s(1,0,0) 16](#_Toc42535757)

[5.1.2. SARIMA with Seasonal Order (0,1,1) 18](#_Toc42535758)

[5.1.3. SARIMA with Seasonal Order (3,1,1) 19](#_Toc42535759)

[5.2. Model Evaluation 21](#_Toc42535760)

[5.3. Residual Analysis and Coefficient Tests of Top Models 22](#_Toc42535761)

[5.3.1. SARIMA(5,1,0)X(3,1,1)12 22](#_Toc42535762)

[5.3.1. SARIMA(2,1,3)X(1,0,0)12 24](#_Toc42535763)

[5.3.2. SARIMA(4,1,5)X(1,0,0)12 26](#_Toc42535764)

[5.4. Best Model 27](#_Toc42535765)

[6. Forecasting 27](#_Toc42535766)

[7. Discussion 28](#_Toc42535767)

[8. Conclusion 29](#_Toc42535768)

[9. References 29](#_Toc42535769)

[10. Appendix 30](#_Toc42535770)

# Table of Figures:

[Figure 1: Monthly GMO application - full data from July 2001 6](#_Toc42525745)

[Figure 2: STL Decomposition of the GMO application time series (from top to bottom the entire time series, seasonality, trend and error) 7](#_Toc42525746)

[Figure 3: GMO data from July 2003 -(Data points before intervention removed) 8](#_Toc42525747)

[Figure 4: Confidence interval for the Box Cox transformation - optimal lambda estimation 9](#_Toc42525748)

[Figure 5: Box Cox Transformed GMO data series with Lambda = 0.1 9](#_Toc42525749)

[Figure 6: Residuals of best deterministic model 11](#_Toc42525750)

[Figure 7: ACF and PACF test of the transformed GMO data. Seasonal is set to 12. 13](#_Toc42525751)

[Figure 8: Residuals of plot after fitting a seasonal (1,0,0) 14](#_Toc42525752)

[Figure 9: Residuals after fitting a seasonal (0,1,0) 14](#_Toc42525753)

[Figure 11: Residuals after fitting SARIMA(0,1,0)x(1,0,0)12 15](#_Toc42525754)

[Figure 12: EACF on SARIMA(0,1,0)x(1,0,0)12 16](#_Toc42525755)

[Figure 13: BIC table for SARIMA(0,1,0)x(1,0,0)12 with nma = 8 and nar = 8 16](#_Toc42525756)

[Figure 14: Residuals after fitting SARIMA(0,1,0)x(0,1,1)12 17](#_Toc42525757)

[Figure 15: EACF on SARIMA(0,1,0)x(0,1,1)12 17](#_Toc42525758)

[Figure 16: BIC table for SARIMA(0,1,0)x(0,1,1)12 with nma = 7 and nar = 7 18](#_Toc42525759)

[Figure 17: Residuals after fitting SARIMA(0,1,0)x(3,1,1)12 18](#_Toc42525760)

[Figure 18: EACF on SARIMA(0,1,0)x(3,1,1)12 19](#_Toc42525761)

[Figure 19: : BIC table for SARIMA(0,1,0)x(3,1,1)12 with nma = 5 and nar = 5 19](#_Toc42525762)

[Figure 20: Residual plots for SARIMA (5,1,0) \*(3,1,1)12 – A) Fitted model line (blue dashed) vs original values (black); B) Standard residual plot; C) QQ plot of the residuals; D) ACF of the residuals; E) PACF of residuals; F) Ljung-Box plot test 21](#_Toc42525763)

[Figure 21: Residual plots for SARIMA (2,1,3) \*(1,0,0)12 – A) Fitted model line (blue dashed) vs original values (black); B) Standard residual plot; C) QQ plot of the residuals; D) ACF of the residuals; E) PACF of residuals; F) Ljung-Box plot test 23](#_Toc42525764)

[Figure 22: Forecast 10 period using SARIMA(2,1,3)X(1,0,0)\_12 26](#_Toc42525765)

# Executive Summary

As the global biotechnology sector rapidly expands, Australia’s contribution to the industry has so far been considered amongst the top in the world. Therefore, the purpose of this project is to investigate the past trends of Australia’s biotechnology sector in order to find a suitable time series model that could be used to forecast its future trend. This investigation is undertaken by examining the number of monthly Notifiable Low Risk Dealing (NLRD) applications that have been made to the Office of the Gene Technology Regulator between the years 2003 to 2019.

Both deterministic and stochastic time series modelling methods were utilized to analyze the number of NLRD applications to determine the most suitable model for forecasting. The results indicated that the stochastic seasonal autoregressive integrated moving average (SARIMA) models were more suitable for this particular dataset. In particular, SARIMA(2,1,3)X(1,0,0)12 model was chosen as the most suitable, and was consequently used to forecast ten months ahead of the dataset. The forecasted result indicated that there is a flat trend stabilization in the number of NLRD applications within the period.

# Introduction

## Background

Around the world, the Biotech sector is a rapidly growing field. The current turn over in the industry is in billions of dollars worldwide (IBIS World, 2020), while Australia has been consistently placed amongst the top biotech economies (Scientific America, 2016). Part of Australia's success is that there is still great focus on basic research and most of it is in the health sciences. Since the 1940, Australia has had 8 Noble Laureates for medicine and physiology (“*List of Australian Nobel Laureates”,* Wikipedia, 2020) – by far the most successful field for Australia.

Understanding growth in research performance can give insights into how the industry is trending. There are multiple ways to measure this such as the number of patents, grant applications, or publications resulting from research. However, the problem with these measures is that they generally measure research that has already been completed. Additionally, the problem with grant applications is that they are often only submitted by academic institutions, while clinical trials are more reflective of the performance of pharmaceutical giants. One method to avoid these issues is to look at the number of Notifiable Low Risk Dealing (NLRD) applications.

Within Australia genetically modified research is controlled by the office of the gene technology regulator. This is governed by multiple federal and state acts which prescribe that any genetically modified organism (GMO) research must be reported to the regulator, and in most cases, permission must be obtained prior to commencement of the research. There are multiple levels of permissions which dictate the biocontainment levels that have to be adhered to when working with a GMO. All of these application records are available to the public from the Office of the Gene Technology Regulator (OGTR) – the most common of which is the NLRD. The NLRD must be obtained prior to commencement on a project, where failure to comply would result in heavy personal and workplace fines, as well as the potential shutdown of the site undertaking the work (Gene Technology Bill, Explanatory Memorandum, 2000). As such, any researchers interested must apply for permissions prior to starting GMO research.

Due to this reason, it is determined that NLRD applications may be a good predictor of current and future research, and consequently, the health of the Australian bio-industry. Additionally, the benefits of an accurately forecasted model can assist the OGRT to optimise its resources and to manage the workload in processing applications in a timely manner.

## Aim

The aim of this project was to find an optimal model that could predict the future NLRD application numbers for the next ten months.

## Data

Public data on individual Notifiable Low Risk Dealings applications from July 2001 to July 2019 was obtained from the Office of the Gene Technology Regulator (Office of the Gene Technology Regulator, 2019). The data was then processed to remove most of the variables, while monthly application totals were calculated to create time series data. The final dataset used in the study covered 219 months where each data point represents the number of GMO application that month. The time series data set will be here on referred to as GMO data.

It should be noted that the data used for in this report may not be well suited for ARIMA modelling as it is count data. This stems from the authors initial misunderstanding that ARIMA should only be used on continuous data, or those with high counts. The counts in the later years of the data were reasonably high and as such it was concluded that some insight can still be gained by using ARIMA for modelling.

# Methodology

## Time Series Analysis

Time series data will be initially analysed visually for any likely points of interventions, presence of trend and seasonality, changing variance and presence of moving average and autoregressive processes. To aid in the analysis, data will be split into seasonal trend and error components using seasonal and trend decomposition using Loess (STL).

If the analysis finds a change in variance, the Box-Cox family of power transformations will be explored for a value to minimise the variance. Additionally, if any interventions are observed, they will be considered for exclusion.

## Deterministic Modelling

The use of deterministic models to represent the data was considered. These include linear, quadratic, cubic and seasonal components. Cyclical models were rejected due to observations made during the initial time series analysis. All potential models were fitted and assigned a rank based on the lowest mean absolute scaled error and Bayesian information criterion. the R squared value, which determines how well the model explains the variance in the GMO data. This was followed by residual and coefficient analysis.

## Stochastic Model Identification

ARIMA and SARIMA models were used for stochastic modelling of the data. The data was initially evaluated for seasonal patterns via two different approaches. The first was to examine the ACF and PACF for any seasonal orders, while the second was applying the residual approach in order to find seasonal orders, which involved performing seasonal differencing and then finding potential seasonal orders.

Then the ordinary orders for each of these seasonal order models was identified. The ACF/PACF, EACF and BIC table was used for this purpose along with the Augmented Dickey fuller test to check for stationarity.

## Stochastic Model Fitting and Selection

The SARIMA models identified in the previous steps were then fitted using the `Arima()` function in R’s Forecast package. Each model was fitted with the following methods:

* Maximum likelihood method (ML)
* Conditional sum of squared method (CSS)
* CSS-ML.

The last method uses conditional-sum-of-squares to find starting values. Then, ML is applied passing the CSS parameter estimates as starting parameter values for the optimization algorithm. The best model for a particular order was selected based on the lowest mean absolute error.

This gave a list of models for evaluation and comparison. As with deterministic models, all stochastic models were ranked on MASE. MASE was preferred for model comparison as it is free of scale and is a good way to compare models on a single series and their forecast accuracy (Hyndman, 2006). Since different methods were used to fit the models, BIC was not used for ranking the models. It should be noted that BIC could still be used for within model comparison. Other error information criterions and diagnostics for the models were also calculated including:

* Akaike Information Criteria (AIC)
* Root Mean Squared Error (RMSE)
* Mean Absolute Error (MAE)
* Residuals Normality
* Proportion of Significant Coefficients

Best models based on lowest MASE, with consideration of other errors were chosen for further residual and coefficient analysis with preference given to those models where residuals followed a normal distribution and that had a high proportion of significant coefficients. Where models performed well on the residual analysis overfitting approach was used to confirm suitability of the model for further use.

## Residual and Coefficient Analysis

Residual analysis was conducted on selected models. Residuals for deterministic and stochastic models were analysed for:

* Distribution about the mean and change in variance (residual time plot)
* Goodness of fit of the model (original data with fitted data overlay).
* Residual normality with quantile-quantile plot (QQ plot)
* Autocorrelation between residuals (ACF plot).

Deterministic models’ residuals were also analysed for normality using a histogram and residuals were plotted against fitted model values to examine model’s performance. Stochastic models were also analysed for independence (Ljung Box plot) and for partial autocorrelation between residuals (PACF plot).

## Forecasting

Once the best model was identified, it was used to forecast the expected monthly applications for the next 10 months. The forecasts will be plotted along with their 95% confidence interval.

# Analysis of Data

The application data that was downloaded from OGTR website was converted into a timeseries covering 216 months from July 2001 to December 2019. Each data point represents the number of GMO application that month. The full data is shown below in Figure1.

A close up of a logo

Description automatically generated

Figure 1: Monthly GMO application - full data from July 2001

It is clear that all of the data are positive values with minimum number of applications being around 20 in a month and maximum number of processed applications of around 150 applications in a month. The time series shows a sharp increase in the number of applications up to 2003 followed by a significant drop back to 2001 level. After July 2003 a clear upward trend in the data can be observed. Presence of seasonality and increase in variance may also be observed however these are hard to interpret. To aid in visualisations of these STL plot has been used (Figure 2).

A close up of a logo

Description automatically generated

Figure 2: STL Decomposition of the GMO application time series (from top to bottom the entire time series, seasonality, trend and error)

Figure 2 confirms the observations in Figure 1, as well as revealing additional observations:

* Seasonal – A clear seasonality is seen in this series. Further there is no clear additional seasonality observed in the remainder
* Trend – As in figure one a steep increase is seen from 2001 to 2003. After June 2003 we see a sharp decrease in number of applications submitted (possible intervention). A gradual increase in application numbers is observed starting 2006. The decomposition makes it clear that after about 2012 the application submissions numbers have remained fairly constant until the end of the observable time series.
* Changing Variance – The increase in magnitude of spikes at later years in the remainder confirms the observations of changing variance
* Intervention – There may be evidence for intervention in July 2003 as mentioned in point 2 above.
* Behaviour – Both AR/MA behaviour are observed with multiple hanging observations and changing mean over time.

Closer examination of legislations found that in 200 and 2001 there was an introduction of Gene Technology Act (2000) and Gene Technology Regulations (2001). These changed how the federal government treats GMO applications and moved the control from multiple other agencies and acts to OGTR. As the changes were implemented, an increase in number of applications was observed to comply with the new rules reducing to baseline levels in July 2003. These changes to legislations are clearly creating point of intervention within the data. The inclusion of this early intervention may introduce errors in modelling and impact current forecast. As such it was decided to remove this intervention data from further analysis. All further analysis was performed on data starting from July 2003. The shortened GMO time series can be observed in Figure 3.

A picture containing bird

Description automatically generated

Figure 3: GMO data from July 2003 -(Data points before intervention removed)

Following removal of intervention, the increase in variance is more obvious (Figure 3) which may also impact normality of the data. Applying Shapiro Wilks test to the data resulted in p value of 2.118 X10-6 indicating lack of normality in the dataset. It may be beneficial for the subsequent modelling to reduce this. One way of performing this is to apply Box-Cox transformation.

A screenshot of a cell phone

Description automatically generated

**Box Cox optimal lambda estimate by MLE**

Figure 4: Confidence interval for the Box Cox transformation - optimal lambda estimation

Maximum likelihood method suggested optimal lambda value of 0.1, as visualised in Figure 4, which is close to log transformation. Applying the Box-Cox transformation with lambda of 0.1 largely reduced the change in variance. Further the p value of 0.08 from Shapiro Wilks test on transformed data suggests that this dataset can be considered to follow a normal distribution.

A screenshot of a cell phone

Description automatically generated

Figure 5: Box Cox Transformed GMO data series with Lambda = 0.1

# Deterministic Modelling

From figure 2 we can observe the series has an obvious positive trend and seasonality. It may be possible that a deterministic model may be able to represent the data. Since the seasonality does not appear to follow an obvious sinusoidal pattern, a seasonal model would be more appropriate then a harmonic one. While the trend appears to be linear, all possible permutations of the regression models up to order of 3 shall be considered in model selection. The models will then be ranked on lowest mean absolute scaled error (MASE) and Bayesian information criterion (BIC). These ranks will be combined for model selection. The model with the lowest combined rank score is selected as the best model to fit the data. BIC was chosen over Akaike information criterion (AIC) as the former penalises models with larger number of parameters more heavily, thereby making it more in line with the concept of parsimony.

The results for top five models are listed in Table 1 below:

*Table 1: Top 5 list of deterministic models fitted to the transformed GMO data ordered by combination of MASE and BIC rank*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Adjusted R** | **Residual Error** | **p value** | **Shapiro-Wilks p** | **BIC** | **MASE** | **BICRank** | **MASERank** | **RankSum** |
| data ~ season + Power\_2 + Power\_3 + 0 | 0.99 | 0.45 | 0.00 | 0.85 | 301.72 | 0.59 | 1 | 1 | 2 |
| data ~ season + Power\_2 + Power\_3 + 1 | 0.61 | 0.45 | 0.00 | 0.85 | 301.72 | 0.59 | 2 | 2 | 4 |
| data ~ season + Power\_1 + Power\_3 + 1 | 0.61 | 0.45 | 0.00 | 0.85 | 301.73 | 0.59 | 3 | 3 | 6 |
| data ~ season + Power\_1 + Power\_3 + 0 | 0.99 | 0.45 | 0.00 | 0.85 | 301.73 | 0.59 | 4 | 4 | 8 |
| data ~ season + Power\_1 + Power\_2 + Power\_3 + 1 | 0.61 | 0.45 | 0.00 | 0.85 | 301.75 | 0.59 | 5 | 5 | 10 |

Two models with the lowest MASE have a power 2 and power 3 component and only differ on presence or absence of an intercept (Table 1). Overall absence of the intercept resulted in lower adjusted R, suggested an overall poorer fit of the model. Overall, most models in the top 5 had similar BIC and MASE values around the 301.7 and 0.59 for BIC and MASE respectively. Residuals for the top five models followed a normal distribution.

Due to the small overall difference in MASE and BIC a larger preference was given to the model with higher adjusted r squared and low MASE. The best model based on the highest adjusted R squared and lowest MASE had power 2 power 3 and seasonal components only. This was also the model with the lowest MASE overall. The aforementioned model was picked for further residual analysis (Figure 6).

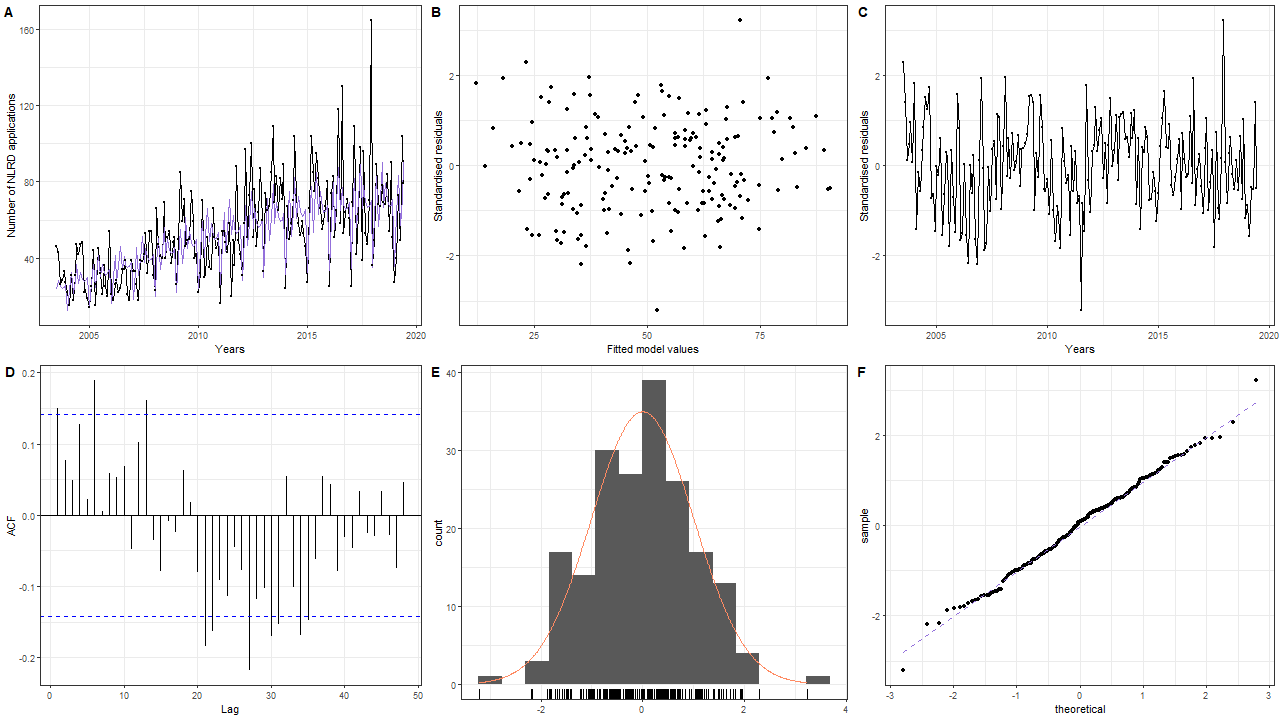


Figure 6: Residuals of best deterministic model

Figure 6A shows how the deterministic model fits GMO data. The model clearly captured the trend and seasonality in the data and overall has a good fit, though some outliers could still be observed. The model consistently underestimates the range of movements in the peak times and overestimate the application rates in the down times. This divergence of the model and the data is most obvious at the later dates possibly due to the increased variance.

Residual verses fitted value plot (Figure 6B) shows a fairly good fit of the model. The model has errors staying around the mean of 0 for all the fitted values, there is no obvious trend, nor change in error variances. These observations are corroborated by timeseries plot of the standardised residuals (Figure 6C). The line does not seem to be deviating from the zero mean and variance seems to be constant.

ACF plot (Figure 6D) of the residuals shows the presence of significant lags that follow a wave pattern. Positive correlations between lags 0 to 20 and then negative lags between periods 20 to 36 can be observed gradually diminishing indicating a possible presence of an autoregressive processes in the residuals. From this it can be inferred that the deterministic modelling did not adequately capture the behaviour of the data.

Visual analysis of the residual’s normality with histogram and Q-Q plots suggest that they follow normal variance (Figure 6E & 6F) fairly closely. Though both show presence of outliers. This result is validated in Table 1 where the Shapiro-Wilks test shows a p values of 0.85 suggesting we do not have evidence to reject the null hypothesis that the residuals follow a normal distribution.

Table 2: Coefficient summary for deterministic seasonal model.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Term** | **estimate** | **std.error** | **statistic** | **p.value** |
| **seasonJanuary** | -8331.04 | 2290.506 | -3.63721 | 0.000361 |
| **seasonFebruary** | -8330.2 | 2290.506 | -3.63684 | 0.000361 |
| **seasonMarch** | -8329.77 | 2290.506 | -3.63665 | 0.000361 |
| **seasonApril** | -8330.34 | 2290.506 | -3.6369 | 0.000361 |
| **seasonMay** | -8330.01 | 2290.506 | -3.63676 | 0.000361 |
| **seasonJune** | -8329.64 | 2290.506 | -3.63659 | 0.000362 |
| **seasonJuly** | -8330.1 | 2290.506 | -3.6368 | 0.000361 |
| **seasonAugust** | -8329.85 | 2290.506 | -3.63668 | 0.000361 |
| **seasonSeptember** | -8330.03 | 2290.506 | -3.63676 | 0.000361 |
| **seasonOctober** | -8330.1 | 2290.506 | -3.63679 | 0.000361 |
| **seasonNovember** | -8330.09 | 2290.506 | -3.63679 | 0.000361 |
| **seasonDecember** | -8330.01 | 2290.506 | -3.63676 | 0.000361 |
| **Power\_2** | 0.006132 | 0.001698 | 3.610753 | 0.000397 |
| **Power\_3** | -2.02E-06 | 5.63E-07 | -3.5967 | 0.000417 |

All coefficients, seasonal or power, appear to be significant for this model (Table 2). It should be noted that the seasonal components have a very large standard error, implying that these may be of limited use for accurately predicting values in this model.

## Summary for Deterministic Model

The best deterministic model had power 2 power 3 and seasonal components only. This model showed a very high adjusted R squared value of 0.99 suggesting good fit. This was confirmed by visual analysis of the fit. The model captured the general trend and the seasonality in the data. Residual analysis showed that they follow a normal distribution. However, residual analysis also showed presence of significant autoregressive behaviour and the coefficients, while statistically significant showed high error rates. Therefore, based on these analysis deterministic models should not be used on this data.

# Stochastic Modelling

## Determine Seasonality

In section 3, the seasonality component of the data was identified visually and confirmed by applying STL decomposition to the data set. Prior to identifying potential ARIMA models the seasonality component in the data must be addressed. This shall be done using two approaches: Classic approach via examination of ACF and PACF plots and residual analysis approach following application of first differenced seasonal model of (0,1,0). For the classic approach the ACF and PACF plot of the original series is shown below.

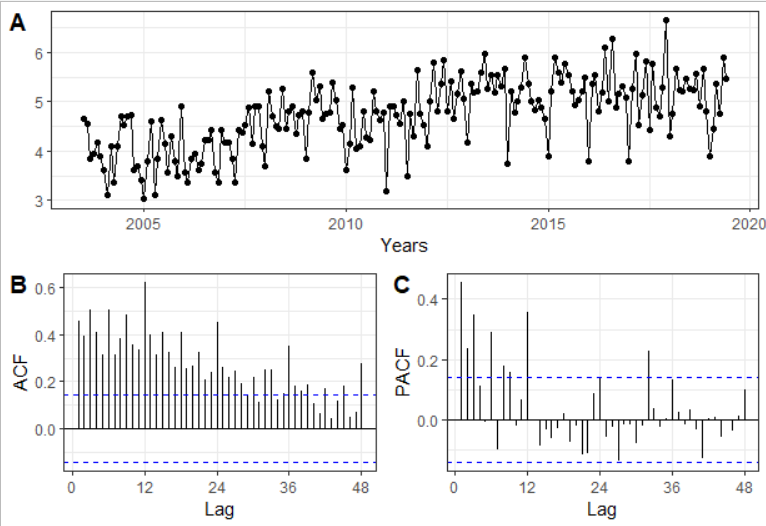


Figure 7: ACF and PACF test of the transformed GMO data. Seasonal is set to 12.

It can be observed that the seasonal peaks in ACF (Figure 7A) ( lags 12, 24, 36, 48) are slowly fading. There is a sharp cut off after first seasonal lag in the PACF plot (Figure 7B). This pattern implies two possibilities, one is that seasonality follows AR(1) process. The second possibility is that it has a trend and needs to be seasonally differentiated, which will be examined during the residual approach. From this we can derive the seasonal component of **(1,0,0)** as a potential candidate.

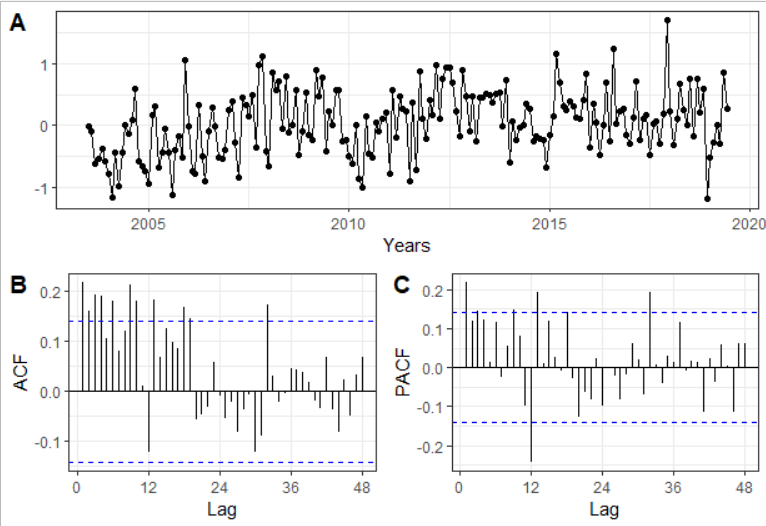


Figure 8: Residuals of plot after fitting a seasonal (1,0,0)

After fitting the seasonal (1,0,0) process, to confirm suitability of this seasonal component, the residuals were visually analysed (Figure 8). The time series plot does not appear to show any seasonality (Figure 8A). Further, the ACF plot (Figure 8B) does not show any significant seasonal lag. PACF plot (Figure 8C) had one significant lag seasonal lag. These observations suggest that there is no obvious presence of seasonality in the plot. Hence this will be used as a potential seasonality component in our analysis.

Seasonality components of the data were then assessed using the residual approach. Following the application of a plain first differenced seasonal model of (0,1,0), the residuals of the model can be analysed to find the other potential components of the seasonal series. The ACF, PACF, and time series of the plain model’s residuals are shown in Figure 9.

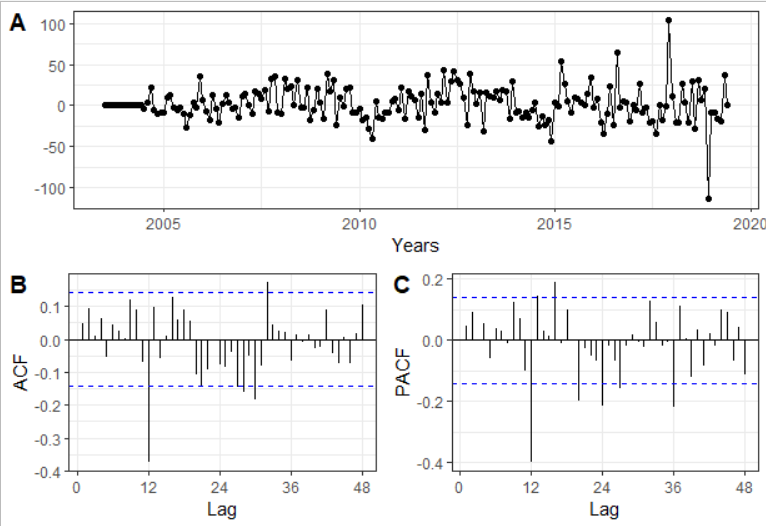


Figure 9: Residuals after fitting a seasonal (0,1,0)

Applying a single seasonal difference appears to have also - at least partially - detrended the data set (Figure 9A). Only one significant seasonal lag (lag 12) in the ACF plot (Figure 9B) could be observed. PACF plot (Figure 9C) showed significant lags at 12, 24, and 36 indicating decaying seasonal lags.

There can be two interpretations of these observations:

1. PACF seasonal lags appear to decay and there is a single significant lag in ACF. This is typical of MA(1) behaviour. Indicating a (0,1,1) seasonal model being suitable.
2. Second interpretations is that there are both AR and MA processes present and an ARMA(3,1,1) seasonal component may be appropriate.

Both of these seasonal candidates will be explored further.

In summary, after applying classical and residual approaches to identifying seasonal components the following were selected:

* (1,0,0)
* (3,1,1)
* (0,1,1)

These were used in the subsequent section to identify potential SARIMA models.

## Model Selection

### SARIMA Seasonal s(1,0,0)

Diagnostic plots for the plain s(1,0,0) model (Figure 10) show the presence of both MA and AR processes and a slight trend; the former suggesting lack of stationarity. Conducting an Augmented Dickey-Fuller (ADF) Test resulted in a p-value of 0.22 failing to reject null hypothesis for the presence of a unit root. This could be handled by taking a first difference on the ordinary order. In other words fitting a SARIMA(0,1,0)x(1,0,0)12 model.

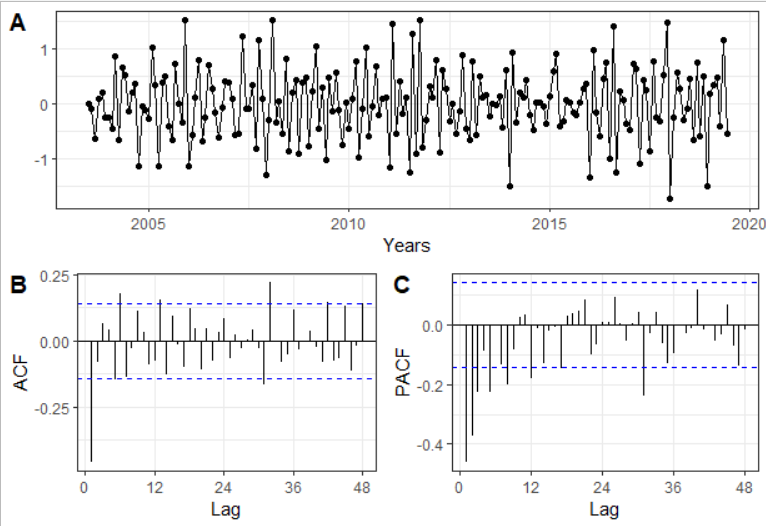


Figure 10: Residuals after fitting SARIMA(0,1,0)x(1,0,0)12

After 1 difference, the application of the ADF test resulted in a p-value of 0.01, suggesting stationarity. This was confirmed visually by examining the time series plot of the residuals (Figure 10A). The ACF plot of the residuals showed a single significant lag (Figure 9B), while the PACF plot of the residuals (Figure 10C) demonstrated either a series of decaying lags or just 3 significant lags. As such the following ARIMA models are possible: (3,1,0), (0,1,1), (3,1,1).

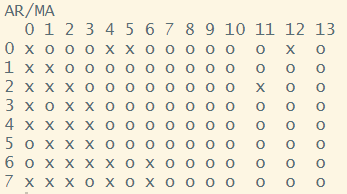


Figure 11: EACF on SARIMA(0,1,0)x(1,0,0)12

EACF plot was used to identify additional candidate models (Figure 11). In this plot there appears to be a vertex at (0,1). While there are two x’s at MA(4) and MA(5), the rest of the vortex is pure and can be used to identify potential models. The likely candidate models identified with EACF are (0,1,1), (0,1,2), and (1,1,2).

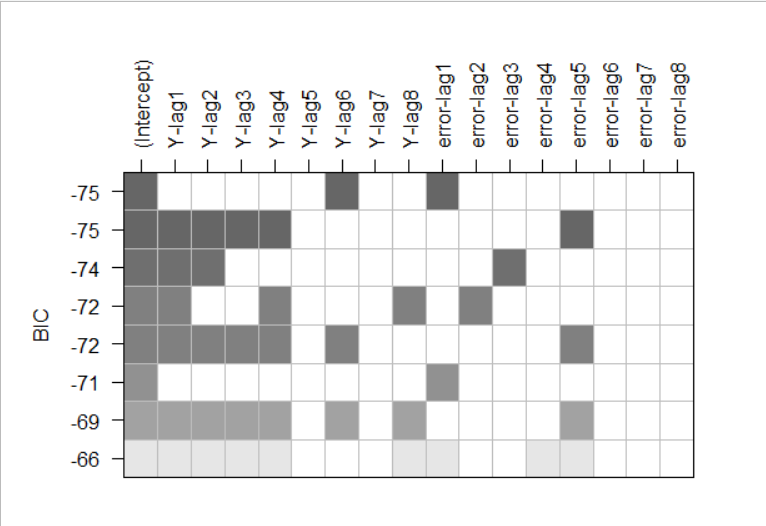


Figure 12: BIC table for SARIMA(0,1,0)x(1,0,0)12 with nma = 8 and nar = 8

Figure 12 shows BIC table plot with nma and nar values of 8. It should be noted that here and below BIC tables with other maximum nma and nar were considered. Those that showed the most common models with highest BIC are shown and used for model selection. From the BIC table we can infer additional possible models of (6,1,1) (4,1,5) and (2,1,3).

In summary, the possible ordinary orders with a seasonality order of s(1,0,0) are (0,1,1), (0,1,2), (1,1,2), (2,1,3), (3,1,0), (3,1,1), (4,1,5), and (6,1,1).

### SARIMA with Seasonal Order (0,1,1)

Applying ADF test to the residuals of plain seasonal (0,1,1) model results in a p-value of 0.32 suggesting non stationarity and presence of unit root. As with the above section this will be addressed by applying an ordinary difference to the series.

ADF test on the residual of SARIMA(0,1,0)x(0,1,1)12 shows a p-value of 0.01, suggesting stationarity and implying that this can be used to find candidate models ARIMA orders. Time series plot of the residuals (Figure 13A) shows lack of trend and relatively constant variance providing further proof for presence of stationarity.

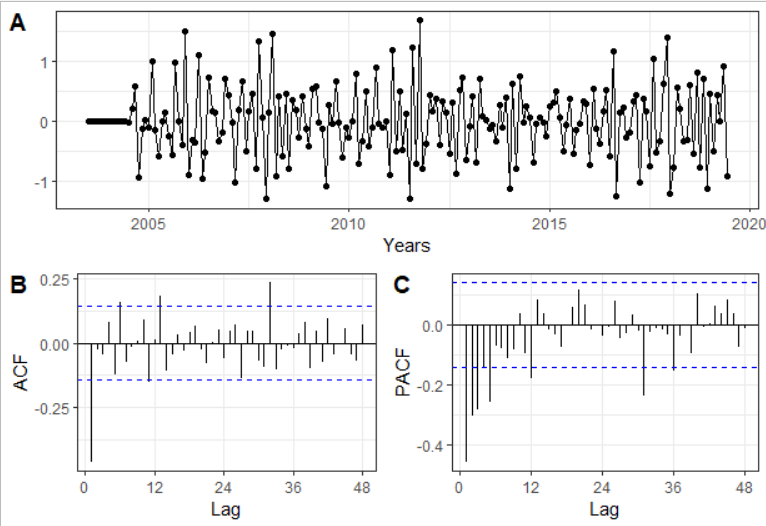


Figure 13: Residuals after fitting SARIMA(0,1,0)x(0,1,1)12

Analysis of seasonal lags shows that the presence of a single slightly significant seasonal lag in PACF plot (Figure 13C). Since this is only slightly above the significance line it can be ignored for the purposes of picking a model as it could be a sign of interplay of other autocorrelations or a result of possible intervention around 2018.

Analysing the ACF and PACF plots (Figure 13B and 13C) for ordinary lags, a single significant lag in ACF, and 5 in PACF were observed. It should be noted that lag 3 in the PACF plot is just significant. Therefore, possible models are (0,1,1), (3,1,0), (5,1,0), (3,1,1) (5,1,1)

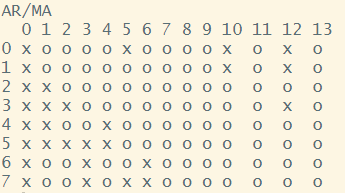


Figure 14: EACF on SARIMA(0,1,0)x(0,1,1)12

From the EACF plot (Figure 14), the possible candidate models identified were (0,1,1), (0,1,2), (1,1,1), (1,1,2).

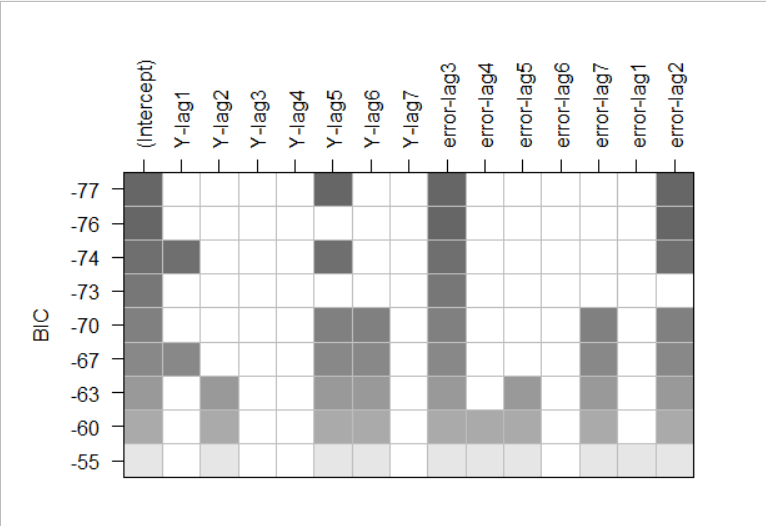


Figure 15: BIC table for SARIMA(0,1,0)x(0,1,1)12 with nma = 7 and nar = 7

Using BIC table plot (Figure 15) with nma and nar of 7 ordinary orders of (5,1,3) (0,1,3) were identified.

In summary, possible ordinary orders with s(0,1,1) are (0,1,1), (0,1,2), (1,1,1), (1,1,2), (3,1,0), (3,1,1), (0,1,3), (5,1,0), (5,1,1), and (5,1,3).

### SARIMA with Seasonal Order (3,1,1)

Lastly, the seasonal order of (3,1,1) was considered. Applying ADF test on the residuals of SARIMA(0,0,0)x(0,1,1)12 resulted in a p-value of 0.36, suggesting lack of stationarity and the presence of a unit root. As with other seasonal models, applying first order ordinary differencing achieved stationarity as suggested by the ADF p-value of 0.01. The time series plot of the residuals (Figure 16A) again confirmed these observations.

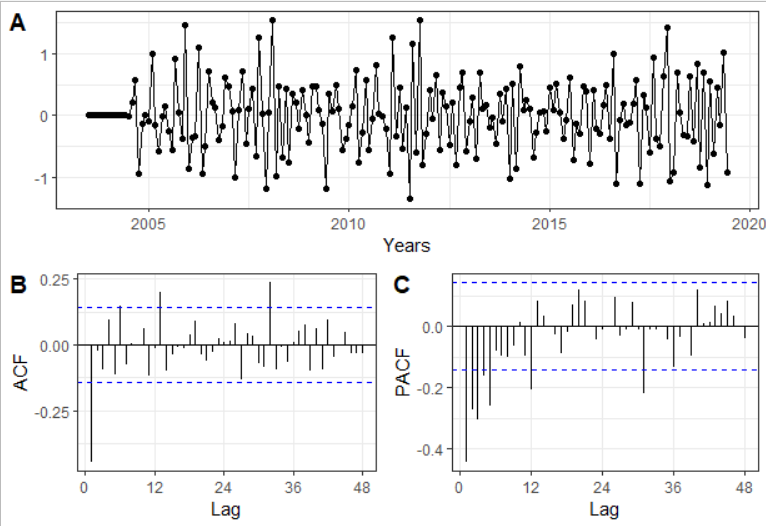


Figure 16: Residuals after fitting SARIMA(0,1,0)x(3,1,1)12

The PACF plot (Figure 16B) showed one significant seasonal lag. No other significant seasonal lag could be observed in both ACF and PACF plots (Figure 16B and 16C). ACF plot showed 3 significant ordinary lags while the PACF plot showed 4 significant ordinary lags followed by a sharp drop off. This could be interpreted as a presence of decaying lag in the PACF, which would suggest MA behaviour. Therefore, the candidate ordinary orders identified in these plots are: (0,1,1), (3,1,0), (5,1,0), (3,1,1), and (5,1,1).

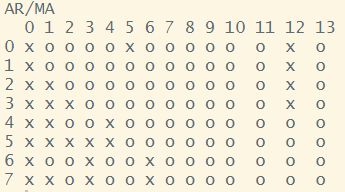


Figure 17: EACF on SARIMA(0,1,0)x(3,1,1)12

Figure 17 shows the EACF plot for the residuals of SARIMA(0,1,0)x(3,1,1). A possible vertex at (0,1) can be observed. This suggests possible candidate models of (0,1,1), (0,1,2), (1,1,1), and (1,1,2).

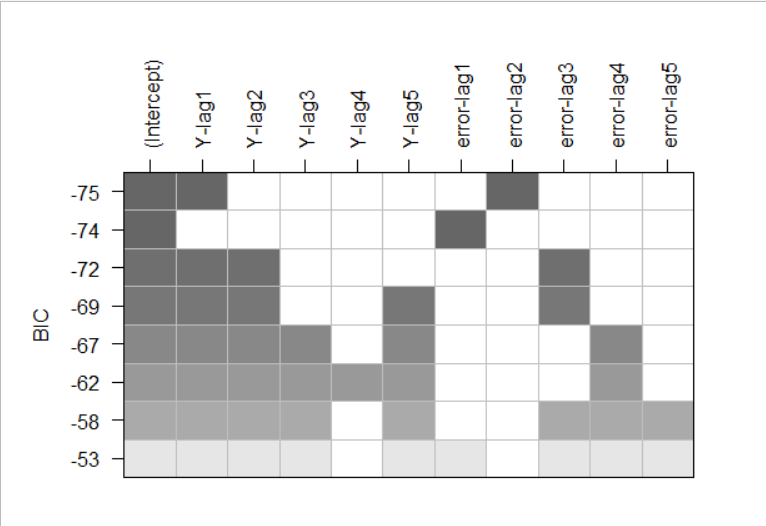


Figure 18: : BIC table for SARIMA(0,1,0)x(3,1,1)12 with nma = 5 and nar = 5

Figure 18 shows the BIC table with nma and nar 5, which suggests ordinary orders of (1,1,2) and (0,1,1).

In summary, possible ordinary orders for SARIMA with s(3,1,1) are (0,1,1), (3,1,0), (5,1,0), (3,1,1) (5,1,1), (0,1,1), (0,1,2), (1,1,1), and (1,1,2).

Below (Table 3) is the final list of candidate models.

Table 3: Table of all candidate models

|  |  |  |
| --- | --- | --- |
| **S(1,0,0)** | **S(0,1,1)** | **S(3,1,1)** |
| (0,1,1) | (0,1,1) | (0,1,1) |
| (0,1,2) | (0,1,2) | (3,1,0) |
| (1,1,2) | (1,1,1) | (5,1,0) |
| (2,1,3) | (1,1,2) | (3,1,1) |
| (3,1,0) | (1,1,2) | (5,1,1) |
| (3,1,1) | (3,1,1) | (0,1,2) |
| (4,1,5) | (0,1,3) | (1,1,1) |
| (6,1,1) | (5,1,0) | (1,1,2) |
|  | (5,1,1) |  |
|  | (5,1,3) |  |

## Model Evaluation

Candidate models identified in section 5.1 were fitted using the l3 methods for the arima() function, namely ML, CSS and CSS-ML. These were compared on MASE, and the one with the lowest error was selected as the best candidate model for a particular order. It should be noted that for some order combinations were not allowed with certain arima() methods by R. Those were not fitted and were not used in the primary model selection. All of the candidate SARMIA models were combined in a diagnostic table. Table 4 shows the top 15 models.

Table 4: Table of top 15 models by MASE

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Rank** | **Ordinary Order** | **Seasonal Order** | **k** | **Method** | **MASE** | **RMSE** | **AIC** | **BIC** | **Normal/ Non normal** | **P-value for Shapiro Wilk test** | **Proportion of significant coefficients** |
| 1 | 5,1,1 | 3,1,1 | 12 | CSS | 0.66 | 16.93 | 252.30 | 291.39 | No | 0.000 | 0.80 |
| **2\*** | 5,1,0 | 3,1,1 | 11 | CSS | 0.66 | 17.11 | 252.22 | 288.04 | No | 0.000 | 0.89 |
| **3** | 3,1,1 | 3,1,1 | 10 | CSS | 0.67 | 16.94 | 254.37 | 286.94 | No | 0.000 | 0.63 |
| **4** | 0,1,2 | 3,1,1 | 8 | CSS | 0.67 | 16.87 | 253.74 | 279.79 | No | 0.000 | 0.67 |
| **5** | 0,1,1 | 3,1,1 | 7 | CSS | 0.67 | 16.92 | 252.14 | 274.93 | No | 0.000 | 0.80 |
| **6** | 1,1,1 | 3,1,1 | 8 | CSS | 0.68 | 16.85 | 251.23 | 277.29 | No | 0.000 | 0.67 |
| **7** | 1,1,2 | 3,1,1 | 9 | CSS | 0.68 | 16.85 | 253.23 | 282.54 | No | 0.000 | 0.43 |
| **8\*** | 2,1,3 | 1,0,0 | 7 | CSS | 0.72 | 16.99 | 258.45 | 281.25 | Yes | 0.250 | 1.00 |
| **9\*** | 4,1,5 | 1,0,0 | 11 | CSS | 0.72 | 17.08 | 262.49 | 298.32 | Yes | 0.136 | 0.80 |
| **10** | 3,1,0 | 3,1,1 | 9 | CSS | 0.72 | 18.10 | 273.03 | 302.35 | No | 0.000 | 0.86 |
| **11** | 5,1,3 | 0,1,1 | 11 | ML | 0.74 | 16.94 | 266.78 | 302.61 | Yes | 0.155 | 0.11 |
| **12** | 5,1,1 | 0,1,1 | 9 | ML | 0.75 | 17.07 | 265.07 | 294.38 | Yes | 0.091 | 0.29 |
| **13** | 1,1,2 | 0,1,1 | 6 | ML | 0.75 | 17.25 | 261.33 | 280.87 | Yes | 0.082 | 0.25 |
| **14** | 0,1,2 | 0,1,1 | 5 | CSS-ML | 0.75 | 17.25 | 259.36 | 275.64 | Yes | 0.082 | 0.67 |
| **15** | 1,1,1 | 0,1,1 | 5 | ML | 0.75 | 17.25 | 259.37 | 275.66 | Yes | 0.082 | 0.67 |

\* Models picked for further evaluation

The top 15 models fitted to the data appear to favour CSS method, though there are some after the top 10 that were fitted with ML. Most of the models with the lowest MASE values belong to the seasonal order of (3,1,1). Curiously all models for that order had residuals that did not follow normal distribution.

The top model based on the lowest MASE value is SARIMA(5,1,1)\*(3,1,1)12 fitted with CSS method. The second best model (SARIMA(5,1,0)\*(3,1,1)12) shows virtually the same MASE value but has a lower BIC value and it also has higher number of significant coefficients. , This model is an “underfit” version of the top model and since it has fewer problem with residual diagnostics will be considered as the top model.

Top 7 models ranked on MASE7 have non-normal residuals. This may indicate that these may he top fail on goodness of fit. The only models in the top 10 that have residuals that follow normal distribution are SARIMA(2,1,3)\*(1,0,0)12 and SARIMA(4,1,5)\*(1,0,0)12. A closer look at the first of these models (model 8) shows that it has a lower BIC than our top picked model and a comparable RMSE. SARIMA(4,1,5)\*(1,0,0)12 model also has a comparable RMSE, AIC values and a large proportion of significant coefficients and shall also be considered. The remaining models in the top 15 all show low proportion of coefficients significance and can be disregarded from further considerations.

## Residual Analysis and Coefficient Tests of Top Models

### SARIMA(5,1,0)X(3,1,1)12

To assess the suitability of SARIMA(5,1,0)X(3,1,1)12 for further use the models residuals and coefficient will be analysed.

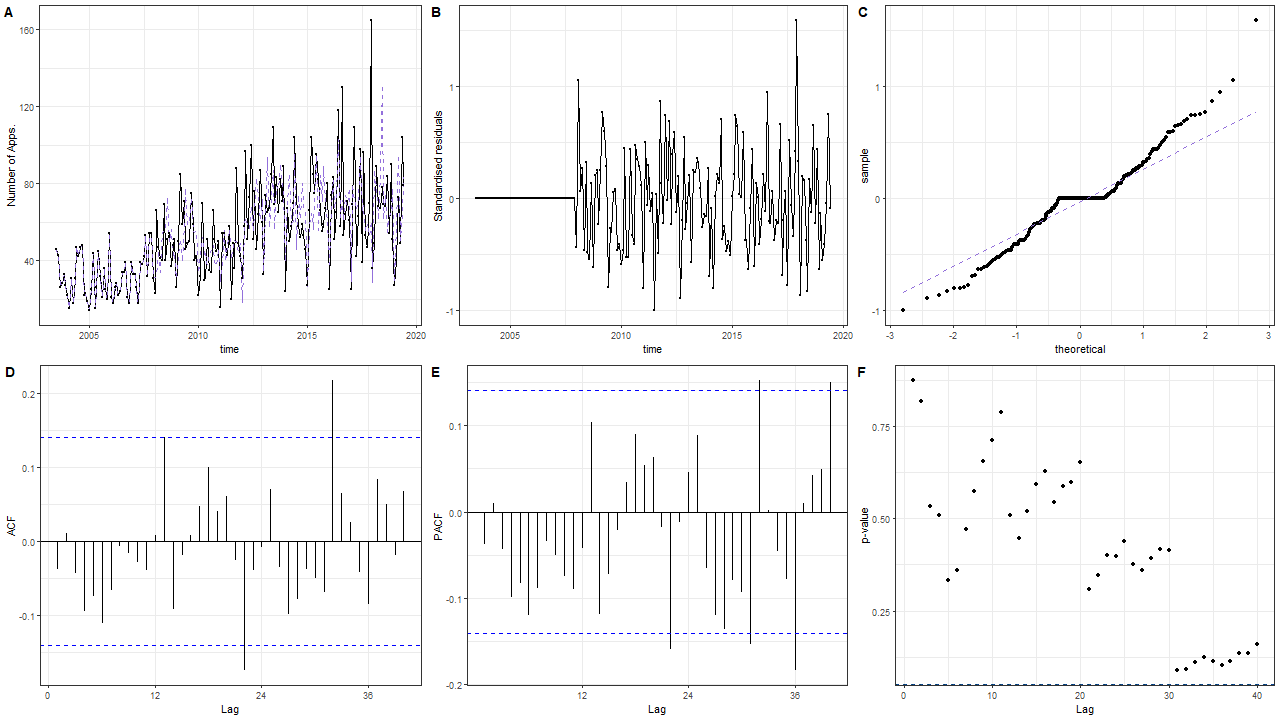


Figure 19: Residual plots for SARIMA (5,1,0) \*(3,1,1)12 – A) Fitted model line (blue dashed) vs original values (black); B) Standard residual plot; C) QQ plot of the residuals; D) ACF of the residuals; E) PACF of residuals; F) Ljung-Box plot test

From Figure 19 for diagnosing SARIMA (5,1,0)\*(3,1,1)12 model:

The fitted model values show some discrepancies away from the original data (Figure 19A). The residual’s variance appear to be constant (Figure 19B) and are distributed around the mean of zero. However, there is a large spike at around 2018 and there are sections where several consecutive observations below zero or above zero suggesting presence of autocorrelation. QQ plot of the residuals (Figure 19C) shows significant deviation away from the central quantile line, confirming the Shapiro Wilks suggestion that the residuals are not normal. In both ACF and PACF plots (Figure 19D and 19F) significant lags can be observed after lag 20. This suggest the model have not captured all of the features of the data. Ljung-Box plot (Figure 19F) suggests presence of correlation of residuals from lag 30 onwards. From these observations, it does not seem that the residuals follow a white noise process.

Table 5: Coefficient tests of SARIMA (5,1,0) \*(3,1,1)12

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **term** | **estimate** | **std.error** | **statistic** | **p.value** |
| ar1 | -0.7665 | 0.0728 | -10.5336 | 0 |
| ar2 | -0.5792 | 0.0877 | -6.6014 | 0 |
| ar3 | -0.5469 | 0.0917 | -5.9669 | 0 |
| ar4 | -0.3703 | 0.0898 | -4.1258 | 0 |
| ar5 | -0.2745 | 0.0721 | -3.8085 | 1.00E-04 |
| sar1 | -0.5008 | 0.1434 | -3.4915 | 5.00E-04 |
| sar2 | -0.395 | 0.0951 | -4.1549 | 0 |
| sar3 | -0.2163 | 0.0794 | -2.726 | 0.0064 |
| sma1 | -0.2618 | 0.171 | -1.5307 | 0.1259 |

Table 5 shows the results from the coefficient test. All AR and SAR components are significant and only SMA component was found not to be significant. Since the residuals are not normally distributed, and coefficient tests required modelling normality, these results should be taken cautiously.

Additional diagnostics for suitability of the model to represent the data could be gained by overfitting the model. Since there were multiple peaks observed in PACF plot, and no correlations in Ljung box plot addition of an MA component may enhance the model performance.

Table 6: Overfitting - coefficient test of SARIMA (5,1,1) \*(3,1,1)12

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **term** | **estimate** | **std.error** | **statistic** | **p.value** |
| ar1 | -0.5165 | 0.1877 | -2.7517 | 0.0059 |
| ar2 | -0.3987 | 0.1505 | -2.6484 | 0.0081 |
| ar3 | -0.425 | 0.123 | -3.4544 | 6.00E-04 |
| ar4 | -0.2657 | 0.113 | -2.3508 | 0.0187 |
| ar5 | -0.2291 | 0.0839 | -2.7309 | 0.0063 |
| ma1 | -0.281 | 0.1984 | -1.4161 | 0.1567 |
| sar1 | -0.4655 | 0.1424 | -3.2694 | 0.0011 |
| sar2 | -0.3772 | 0.0966 | -3.9052 | 1.00E-04 |
| sar3 | -0.2043 | 0.081 | -2.5225 | 0.0117 |
| sma1 | -0.2955 | 0.1695 | -1.7431 | 0.0813 |

Table 6 shows the coefficient diagnostic of the overfitted model. The added MA(1) component is insignificant and coefficients values are similar. Since overfitting did not improve the coefficient diagnostics SARIMA (5,1,0) \*(3,1,1)\_12 model is acceptable from a coefficient test point. SARIMA(2,1,3)X(1,0,0)\_12. However, due to multiple issues with residual diagnostics this model may not be the best model for further consideration for forecasting.

### SARIMA(2,1,3)X(1,0,0)12

Model residuals and significance of models coefficients for SARIMA(2,1,3)X(1,0,0)12 were considered to evaluate the suitability of the model for use in forecasting.

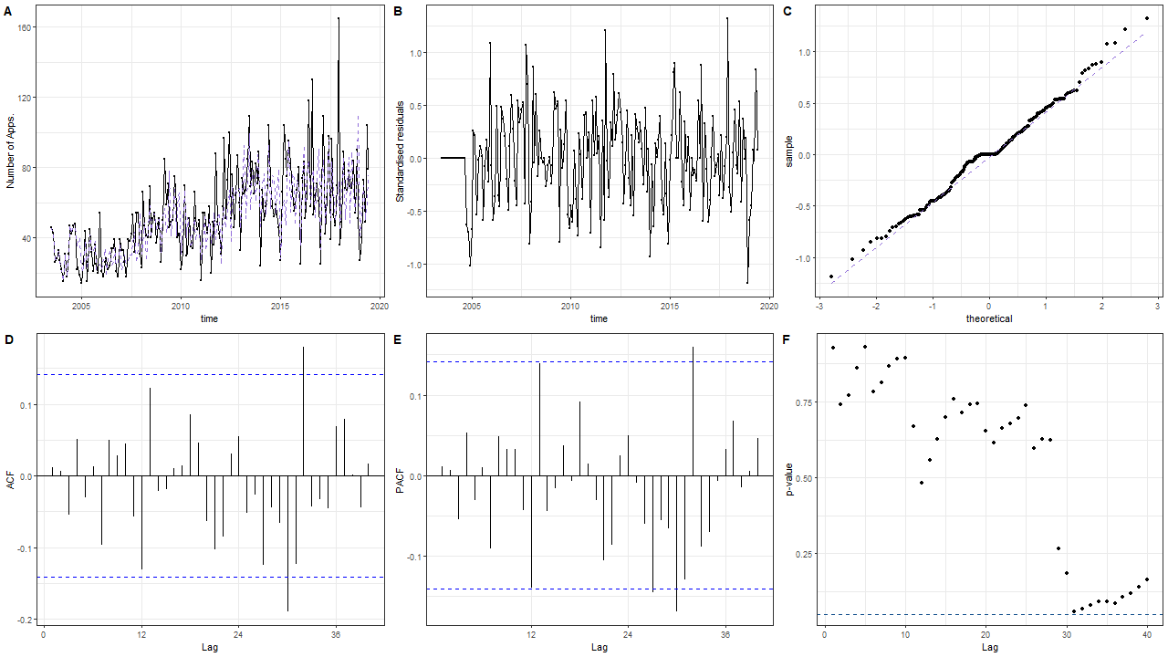


Figure 20: Residual plots for SARIMA (2,1,3) \*(1,0,0)12 – A) Fitted model line (blue dashed) vs original values (black); B) Standard residual plot; C) QQ plot of the residuals; D) ACF of the residuals; E) PACF of residuals; F) Ljung-Box plot test

Figure 20 is the diagnostic plots for the model SARIMA(2,1,3)X(1,0,0)12. There were no significant deviations observed of fitted values for SARIMA(2,1,3)X(1,0,0)12 model away from the original data (Figure 20A). The time series plot of the residuals (Figure 20B) shows the residuals of the model are centred around the zero mean and follows a zigzag pattern with constant variance. However, a couple of spikes are evident, though these are still close to either -1 or 1. The residuals also appear to follow a normal distribution according to QQ plot (Figure 20C). This confirms the observation in Table 5 where the Shapiro Wilk’s test value of this model from Table 5 failed to find significant evidence of non-normality in the residuals. The ACF and PACF plots in (Figure 20D & E) show largely insignificant lags except at around lag 30 and 32. The residual lags show independence using Ljung-Box test plot where none of the p values are significant (Figure 20F).

Table 7: Coefficient test of SARIMA (2,1,3) \*(1,0,0)12

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **term** | **estimate** | **std.error** | **statistic** | **p.value** |
| ar1 | -0.9838 | 2.00E-04 | -4992.05 | 0 |
| ar2 | -0.9715 | 2.00E-04 | -4285.73 | 0 |
| ma1 | 0.2326 | 0.0485 | 4.7987 | 0 |
| ma2 | 0.1708 | 0.0536 | 3.1897 | 0.0014 |
| ma3 | -0.9063 | 0.052 | -17.444 | 0 |
| sar1 | 0.4108 | 0.0032 | 128.4042 | 0 |

Table 7 shows the coefficient analysis for SARIMA (2,1,3) \*(1,0,0)12. As expected from table 5 all the coefficient have a p-value of less than 0.05 indicating significance.

From the residual diagnostic plots, it is clear that this model outperforms the model analysed in section 5.4.1. The residuals are normal, with uncorrelated lags and the autocorrelations are almost all insignificant. Further all the coefficients are significant suggesting SARIMA (2,1,3) \*(1,0,0)12 may be a good candidate model for forecasting. Overfitting this model may provide additional diagnostics for suitability of this model.

The model will be overfitted with (individually) MA and AR components. Overfitting with an additional MA component (Table 8) shows that the overfitted ma4 coefficient is insignificant with a p value of 0.876. Adding an MA component failed to significantly affect the value of the coefficient estimates.

Table 8: Overfitting test with SARIMA (2,1,4) \*(1,0,0)12 model

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **term** | **estimate** | **std.error** | **statistic** | **p.value** |
| ar1 | -0.9931 | 4.00E-04 | -2556.14 | 0 |
| ar2 | -0.9825 | 4.00E-04 | -2375.37 | 0 |
| ma1 | 0.2483 | 0.0701 | 3.5403 | 4.00E-04 |
| ma2 | 0.1336 | 0.1015 | 1.3167 | 0.1879 |
| ma3 | -0.9247 | 0.1007 | -9.1786 | 0 |
| ma4 | -0.0206 | 0.1317 | -0.1563 | 0.8758 |
| sar1 | 0.3631 | 0.0062 | 58.4703 | 0 |

Overfitting with an additional AR component (Table 9) resulted in insignificant overfitted ar3 coefficient with p value of 0.64 as were ar2,ar1, ma2 and ma3 . It should be noted that the coefficient estimates are significantly different from the original model values suggesting that this is a different model.

Table 9: Overfitting test with SARIMA (3,1,3) \*(1,0,0)12 model

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **term** | **estimate** | **std.error** | **statistic** | **p.value** |
| ar1 | -0.089 | 0.2731 | -0.3258 | 0.7446 |
| ar2 | -0.1686 | 0.2087 | -0.8079 | 0.4191 |
| ar3 | 0.0406 | 0.0872 | 0.466 | 0.6412 |
| ma1 | -0.7991 | 0.2846 | -2.8074 | 0.005 |
| ma2 | 0.0224 | 0.349 | 0.0643 | 0.9487 |
| ma3 | -0.0822 | 0.1962 | -0.4192 | 0.6751 |
| sar1 | 0.4484 | 0.0661 | 6.7858 | 0 |

Overfitting SARIMA(2,1,3)X(1,0,0)12 confirms the validity of the coefficients in model as All of the additional coefficients were insignificant. This confirms that SARIMA(2,1,3)X(1,0,0)12 is likely to be a good model for forecasting NLRD applications.

### SARIMA(4,1,5)X(1,0,0)12

Lastly SARIMA(4,1,5)X(1,0,0)12 model residuals and coefficients were evaluated to examine the models suitability for forecasting. The model in section 5.3.2 could be thought of as a potential nested version of this model and it is likely that the residuals and coefficients will be similar.

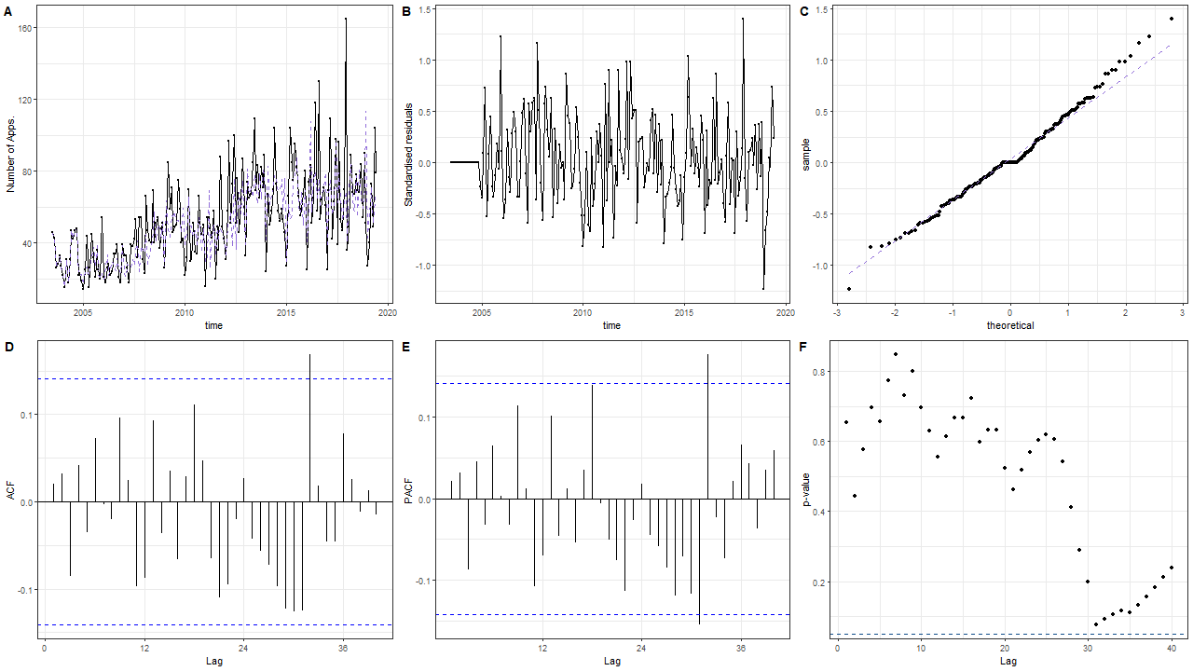


Figure 21: Residual plots for ARIMA (4,1,5) \*(1,0,0)12

Figure 21 shows the diagnostic plots for the model SARIMA(4,1,5)X(1,0,0)12. The diagnostics appear to be virtually the same as those for SARIMA(2,13)X(1,0,0). The fitted values for the model show only a few deviations from the original data (Figure 21A). The residuals follow what can be generally thought of white noise and have a normal distribution (Figure 20B and C). The residuals however do have a higher value in the upper bonds reaching closer to 1.5. The ACF and PACF plots (Figure 21D and E) show largely insignificant lags except one or two around lag 32 for both plots. The residual lags failed to show correlation at any lags according to Ljung-Box test plot (Figure 21F)

Table 10: Coefficient test of SARIMA (4,1,5) \*(1,0,0)12

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **term** | **estimate** | **std.error** | **statistic** | **p.value** |
| ar1 | -0.6673 | 0.0244 | -27.3579 | 0 |
| ar2 | -0.2046 | 0.0122 | -16.7425 | 0 |
| ar3 | -0.1326 | 0.0114 | -11.5928 | 0 |
| ar4 | -0.4819 | 0.0095 | -50.7365 | 0 |
| ma1 | -0.1049 | 0.0591 | -1.7731 | 0.0762 |
| ma2 | -0.5676 | 0.1106 | -5.1298 | 0 |
| ma3 | 0.0287 | 0.0586 | 0.4895 | 0.6245 |
| ma4 | 0.5498 | 0.0913 | 6.0245 | 0 |
| ma5 | -0.6699 | 0.0973 | -6.8846 | 0 |
| sar1 | 0.497 | 0.0743 | 6.6902 | 0 |

Table 10 shows the coefficient test of SARIMA (4,1,5) \*(1,0,0)12. All coefficients were found to be significant with the exception of ma1 and ma3.

While this model has promising residuals diagnostics and only a couple of problems in coefficients it does not appear to be as good as the model analysed in section 5.3.2. Firstly this model is a much larger model compared to the model analysed in section 5.3.2. Secondly the errors from table 5, (RMSE, AIC and BIC) are also larger. As such and in accordance with the laws of parsimony we will not choose this model

## Best Model

After taking in to account the MSE, RMSE AIC, BIC, normality of residuals, and significance proportion of coefficients, the best model chosen is SARIMA(2,1,3)X(1,0,0)12. This model will be used to forecast NLRD applications in the next section.

# Forecasting

SARIMA(2,1,3)X(1,0,0)12. Model was used to forecast the next 10 10 periods of NLRD applications

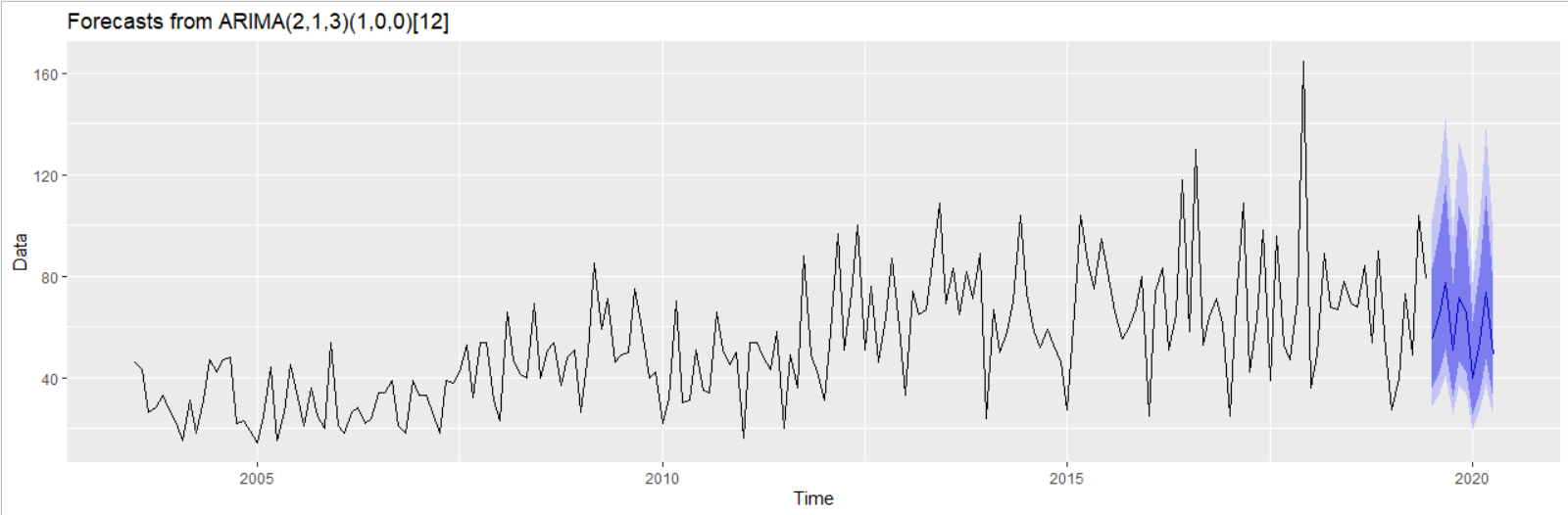


Figure 22: Forecast 10 period using SARIMA(2,1,3)X(1,0,0)\_12

Figure 22 shows the forecast NLRD applications. The original data is in the black line and the forecasted numbers are shown as the blue line. The darker blue section of the graph represents the 80% confidence interval while the lighter blue represents the 95% confidence interval. Forecasted figures are shown in the table below (Table 11). The numbers are rounded off to the nearest integer since the data is count of applications.

Table 11: Ten period forecasted figures using SARIMA(2,1,3)X(1,0,0)12

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Point | | Forecast | Lo 80 | Hi 80 | Lo 95 | Hi 95 |
| Jul | 2019 | 55 | 36 | 83 | 29 | 102 |
| Aug | 2019 | 65 | 42 | 97 | 34 | 120 |
| Sep | 2019 | 78 | 51 | 116 | 41 | 142 |
| Oct | 2019 | 51 | 33 | 77 | 26 | 96 |
| Nov | 2019 | 72 | 47 | 108 | 37 | 133 |
| Dec | 2019 | 65 | 42 | 99 | 33 | 122 |
| Jan | 2020 | 40 | 25 | 62 | 20 | 78 |
| Feb | 2020 | 54 | 35 | 83 | 27 | 104 |
| Mar | 2020 | 74 | 48 | 112 | 37 | 138 |
| Apr | 2020 | 49 | 31 | 76 | 24 | 95 |

Monthly numbers and forecasts values oscillate within ≈ 40 to 74 range and show a reducing trend during the 10-month forecast period. Overall, forecasted numbers seems to be stable. This is consistent with our initial observations of the data.

# Discussion

In this study we explored the time series data of applications to work with GMO in particular NLRD. Early in the full data set a periods of intervention was identified and removed. A changing variance over time was found and a Box-Cox power transformation was applied to the data to rectify this.

Both Deterministic and stochastic models were applied to the data to identify a best fit model to use for prediction. The best deterministic model selected on basis of MASE and BIC was:

The residuals from this model were not normally distributed and were found to still have significant autocorrelation From these results, it was inferred that the fitted deterministic models are not suitable to model NLRD applications.

A number of stochastic models were fitted and the best sthe best model identified, through a combination of MASE and BIC and some residual analysis was:

Residuals of this model confirmed to normality and were largely uncorrelated. The autocorrelations of the residuals are also almost all insignificant. Therefore, the residuals were inferred to be a white noise process. All of the model’s coefficients were significant. This was confirmed by overfitting. On these basis, this model was selected as the best model for forecasting.

Although we removed most of the intervention by dropping the data from before July 2003, there appears to be additional possible interventions in the data around 2018 that produced large spike in applications. This spike added error to residuals and possibly some autocorrelations. This created difficulties in modelling of the data. The issues can be observed within the residuals of the SARIMA models where there is always at least one significant correlation present around lags 30 to 32. Potential methods to improve upon the accuracy can be to treat the value as an outlier and smooth it out spike with statistical methods, or modelling, such as simple exponential smoothing model. However, these methods are not part of the scope for this course.

Overall, the model produced good results in the forecasting with confidence interval ranges not too large. The results can be used to give confidence in number of GMO applications to remain roughly the same level as in the previous year with neither significant growth or decline anticipate.

A major limitation in this study is due to the model being count data. In particular, SARIMA models are designed for continuous data and the forecast are as continuous data. We rounded off the final forecast figures to the nearest integer. But that means the confidence intervals is not exact. We recognised for count data, there are more appropriate models to evaluate than SARIMA models. However, SARIMA models were used to evaluate the data source since that is the scope of this course. While it would be interesting to see the differences between the method applied here and the method for count data in terms of the forecasts, but this is not the aim of this project and can be considered in other analysis.

# Conclusion

Multiple models were identified that could predict the GMO application monthly rate. The best model identified was SARIMA(2,1,3)X(1,0,0)12. Forecasting using this figure produced results that indicated stabilisation in the use of GMO and stabilisation of the industry.

# References

* Ditiatkovski M., (2020), MATH1318 Time Series Analysis Assignment 1, multiple functions, RMIT 2020.
* Gene Technology Act (2000), Accessed on 26th May, <https://www.legislation.gov.au/Details/C2016C00792>.
* Gene Technology Bill, Explanatory Memorandum (2000), Accessed on 26th May, <https://www.legislation.gov.au/Details/C2004B00717/Explanatory%20Memorandum/Text>.
* Gene Technology Regulations (2001), Accessed on 26th May, <https://www.legislation.gov.au/Details/F2019C00781>.
* IBIS World, (2020), Global Biotechnology Industry, industry report C1214, IBIS World, viewed 27 May 2020, IBISWorld database.
* Office of the Gene Technology Regulator, (2019), Excel version of list of NLRDs as notified to the Gene Technology Regulator , Data source, saved on 14 May 2019, <http://www.ogtr.gov.au/internet/ogtr/publishing.nsf/Content/1C59C053353D94A0CA257CD00009FD88/$File/NLRD%20list%20December%202019.xlsx>.
* Office of the Gene Technology Regulator, (2019), List of Notifiable Low Risk Dealings (NLRDs), accessed on 14 May 2019, <http://www.ogtr.gov.au/internet/ogtr/publishing.nsf/Content/nlrdrec-1>.
* Rob J Hyndman, Anne B Koehler, (2006), International Journal of Forecasting 22(4), 679-688.
* Scientific American, (2016), WorldView – A global biotechnology perspective, Scientific American, viewed 27 May 2020, <static.scientificamerican.com/wv/assets/2016\_SciAmWorldView.pdf>.
* Wikipedia, (2020), List of Australian Nobel Laureates, Wikipedia, viewed 27 May 2020, <en.wikipedia.org/wiki/List\_of\_Australian\_Nobel\_laureates>.
* Zhang Y.C.(2020) MATH1318 Time Series Analysis Assignment 2, Timeseries Functions Chris, RMIT 2020.

# Appendix

R-codes:

library(dplyr)

library(TSA)

library(stringr)

library(ggplot2)

library(forecast)

library(ggpubr)

library(readxl)

library(lubridate)

library(tseries)

library(urca)

library(x12)

library(tidyverse)

library(lmtest)

library(broom)

library(gridExtra)

#suppress warnings

options(warn=-1)

################################

# Self defined functions #

################################

# Addapted from Ditiatkovski M, 2020

# Custom function to extract summary statistics for a lm

# type model and put them into a single line

summaryTable <- function(model){

residuals <- rstudent(model)

tempSummary <- summary(model)

outputDump <- capture.output(tempSummary)

adjR <- tempSummary$adj.r.squared

resErr <- tempSummary$sigma

if(is.null(resErr)){

resErr <- str\_extract(outputDump, "standard error: \\d+.\\d")

resErr <- str\_extract(resErr[!is.na(resErr)],"\\d+.\\d") %>% as.numeric()

}

# extract p-value from summary dump

pVal <- str\_extract(outputDump, "p-value: \\d+.\\d")

if(sum(!is.na(pVal)) == 0){

pVal <- str\_extract(outputDump, "p-value: < \\d+.\\d+e-\\d+")

pVal <- str\_extract(pVal[!is.na(pVal)], "\\d+.\\d+e-\\d+") %>% as.numeric()

}else{

pVal <- str\_extract(pVal[!is.na(pVal)], "\\d+.\\d") %>% as.numeric()

}

# calculate p value for normality

shw <- shapiro.test(residuals)

#this does not make sence for for power 1 no intercept this fails if combined with previou line

shw <- shw$p.value

resErr <- tempSummary$sigma

# make a nice summary table

BIC <- BIC(model)

MASE <- dLagM::GoF(model)$MASE

summaryTable <- tibble("Adjusted R" = adjR, "Residual Error" = resErr,

"p value" = pVal, "Shapiro-Wilks p" = shw, "BIC" = BIC, "MASE"= MASE)

return(summaryTable)

}

# Addapted from Ditiatkovski M, 2020

# Supporting function that ranks model on MASE and BIC. Expects table in format of sumaryTable() output

# Returns the rank table and the model in a list

modelRank <- function(sumTable){

#Add rank for R^2 and residual error

sumTable <-dplyr::arrange(sumTable, `BIC`) %>%

mutate(BICRank = cumsum(index/index))

sumTable <-dplyr::arrange(sumTable, `MASE`) %>%

mutate(MASERank = cumsum(index/index),

RankSum = MASERank + BICRank) %>%

dplyr::arrange(`RankSum`, `MASE`)

# Check if residuals are normal and p value for model is significant

norm <- which(sumTable$`Shapiro-Wilks p`>0.05)

sig <- which(sumTable$`p value`<0.05)

# find the highest ranking model that has normal residuals p<0.05

# In case of a tie the one with higher adjusted R is selected

if(length(norm)>0 | length(sig)>0){

suppressWarnings(modelNo <- min(intersect(norm, sig)))

if(is.infinite(modelNo)){

modelNo <- 1

}

}

else{

modelNo <- 1

}

# Return updated table and the index of the model in a list

modelAndTable <- list("table" = sumTable, "selectedModel" = sumTable$index[modelNo])

modelAndTable$table$index <- NULL

return (modelAndTable )

}

# Addapted from Ditiatkovski M, 2020

# Supporting function for Model picking. Writes permutations of formula for lm with permutations

# up to maximum power of 3 (ie cubic). Expects a vector with list of variable names in formula

# sorted in ascending order, max power (int) and optional seasonal/harmonic vector

formulaList <- function(powers = NULL, maxPower = 1, season = NULL){

formulaStr <- NULL

# make the begining of the formula

if(!is.null(season)){

formulaStr <- paste0("data~", str\_c(season, collapse = " + "))

}else{

formulaStr <- "data ~"

}

formulaList <- NULL

# loop for formula createion (can be combined with later loop but makes it too complex)

# make this a separate function that returns a list of formula

for(p1 in 1:2){

for(p2 in 1:2){

for(p3 in 1:2){

for(inter in 0:1){

# skip if all powers all are 2 (ie season only component)

if(p3+p2+p1 == 6){

next()

}

#skip if not max power

if(maxPower == 1 & (p1 == 2 | (p1 == 1 & p3+p2 <= 3))){

next()

}

if(maxPower == 2 & p3 == 1){

next()

}

# reset the formula to starting version

tempFormula <- formulaStr

# add items as appropriate where 1=true and 2 =false

if(p1 == 1){

tempFormula = paste0(tempFormula,"+", powers[1])

}

if(p2 == 1){

tempFormula = paste0(tempFormula,"+", powers[2])

}

if(p3 == 1){

tempFormula = paste0(tempFormula,"+", powers[3])

}

tempFormula = paste0(tempFormula, "+", inter)

# add created formula to a list of formulas

formulaList <- c(formulaList, as.formula(tempFormula))

}

}

}

}

return(formulaList)

}

# Addapted from Ditiatkovski M, 2020

# Function will cycle through combination of models up to power of 3

# and selects one with the highest R^2 and lowest residual error. Expects orignal data, maximum power

# integer that will be constant in all models and seasonal variable.

modelSelect <- function(data, maxPower, season = NULL){

# setupfor formula prep and data managment

powers <- seq(1:maxPower)

powerNamesForm <- paste0("Power\_", powers)

powerNames <- powerNamesForm

dates <- matrix(time(data), ncol = maxPower, nrow = (length(data)))

dates <-as\_tibble(t(t(dates)^powers), .name\_repair = "minimal")

seasonCol <- NULL

# deal with seasonal variable if needed

if(!is.null(season)){

dimensions <- dim(season)

if(is.null(dimensions)){

seasonCol <- "season"

}else{

seasonCol <- colnames(season)

seasonCol <- str\_replace\_all(seasonCol,"\\\*|\\(|\\)","")

}

dates <- cbind(season, dates)

powerNames <- c(seasonCol, powerNames)

}

# Make the formula list for all models

formulaList <- formulaList(powerNamesForm, maxPower, season = seasonCol)

data <- cbind(as.vector(data), dates)

powerNames <- c("data", powerNames)

# create final frame for model fitting

colnames(data) <- powerNames

data <- as\_tibble(data, .name\_repair = "minimal")

modelList <- list()

sumTable <- NULL

for (i in 1:length(formulaList)) {

modelList[[i]] <- lm(formulaList[[i]], data)

tempRow <- cbind("index" = i, "Model" = deparse(formulaList[[i]]), summaryTable(modelList[[i]]))

sumTable <- rbind(sumTable, tempRow)

}

# get the ranking table to select model, then return both

modelRank <- modelRank(sumTable)

modelAndTable <- list("Table" = as\_tibble(modelRank$table, .name\_repair = "minimal"), "model" = modelList[[modelRank$selectedModel]])

return(modelAndTable)

}

# Addapted from Ditiatkovski M, 2020

# Function makes and arranges Diagnostic plots for Deterministic type variables.

# Expects input of data and the fitted model. If BoxCox transformation has been applied

# supplying lambda will reverse the transformation

plotRes <- function(data, model, lambda = 1, vertical = TRUE) {

# If needed backtransform data and fitted values

if(lambda!=1){

fittedVals <- data.frame("data" = InvBoxCox(fitted(model), lambda = lambda))

data <- InvBoxCox(data, lambda = lambda)

}else{

fittedVals <- data.frame("data" = fitted(model))

}

residuals <- data.frame("residuals" = rstudent(model))

time <- as.vector(time(data))

# Original vs fitted values

dataVsModel <- ggplot(data = data, aes(x = time, y = data)) +

geom\_line() +

geom\_line(data = fittedVals, aes(y = data), colour = "mediumpurple") +

geom\_point(size = 0.4)+

xlab("Years") +

ylab("Number of NLRD applications") +

theme\_bw()

# residuals ploted

resid <- ggplot(data = residuals, aes(x = time, y = residuals))+

geom\_line() +

geom\_point(size = 0.2) +

theme\_bw() +

xlab("Years") +

ylab("Standardised residuals")

# Fitted vs residuals

fitRes <- data.frame("residuals" = unlist(residuals), "fitted" = unlist(fittedVals))

fitVsRes <- ggplot(data = fitRes, aes(x = fitted, y = residuals)) +

geom\_point() +

xlab("Fitted model values") +

ylab("Standardised residuals") +

theme\_bw()

# create ACF plot

ACF <- ggAcf(residuals, lag.max = 48)+

theme\_bw() +

theme(plot.title = element\_blank())

# create histogram with normal overlay (this is more for aesthetic reasons in the figure)

hist <- forecast::gghistogram(residuals[[1]], add.normal = T) +

theme\_bw() +

theme(axis.title.x = element\_blank(),

plot.title = element\_blank(),)

# create qq plot for the residuals

qq <- ggplot(data = residuals, aes(sample = residuals)) +

stat\_qq() +

stat\_qq\_line(linetype = "dashed", color = "mediumpurple") +

theme\_bw()

## arrange everything nicely and add labels

if (vertical){

plotsComb = ggarrange(dataVsModel, fitVsRes, resid, ACF,

hist, qq, labels = c("A","B","C","D","E","F"), ncol = 2, nrow = 3)

}else{

plotsComb = ggarrange(dataVsModel, fitVsRes, resid, ACF,

hist, qq, labels = c("A","B","C","D","E","F"), ncol = 3, nrow = 2)

}

return(plotsComb)

}

################################

###Determinisitc model codes####

################################

# define significant figure numbers

options(pillar.sigfig = 5)

gmo <- read\_xlsx("NLRD list December 2019.xlsx")

colnames(gmo)[5] <- "dates"

gmo1 <- gmo %>% group\_by(year(dates), month(dates))

gmo2 <- gmo1 %>% summarise(n())

gmo2 <- gmo2 %>% arrange(.,`year(dates)`,`month(dates)`)

gmo.ts <- ts(gmo2$`n()`, frequency = 12, start = c(2001,7))

#Plot Figure 1

plot(gmo.ts, type = 'l', main = "GMO monthly research application", ylab = "Number of applications")

#Plot Figure 2

plot(stl(gmo.ts, s.window = "periodic"),cex.axis=20,main="STL decomposition of the GMO Time Series ",cex.main=10)

#reduce seasons

gmo.ts.short <- ts(gmo.ts[25:length(gmo.ts)], start=c(2003,07), frequency = 12)

#Plot Figure 3

plot(gmo.ts.short, type = 'l', main = "GMO monthly research application", ylab = "Number of applications")

paste("p-value for shapirowilk test of normality for the short series is ",

shapiro.test(gmo.ts.short)$p.value)

#Plot figure 4

#Box Cox tranform the data

bc <- BoxCox.ar(gmo.ts.short, method = 'mle')

#set up for lambda

lambda = bc$mle #lambda = 0.1

gmo.bc = (gmo.ts.short^lambda-1)/lambda

paste("p-value for shapirowilk test of normality for the short series is ",

shapiro.test(gmo.bc)$p.value)

#Plot Figure 5

plot(gmo.bc, type = 'l', main = "GMO monthly research application", ylab = "Transformed values")

#Output Table 1

shortGmoSeasons <- season(gmo.ts.short)

shortSeasonalDeterministic <- modelSelect(gmo.bc, 3, shortGmoSeasons)

shortSeasonalDeterministic$Table

# Output Table 2

detSum <- summary(shortSeasonalDeterministic$model)

detSum$coefficients

write.table(detSum$coefficients, "Deterministic Coefs.csv", sep = ",", row.names = FALSE)

#Plot Figure 6

#png("DeterministicHoriz.png", width = 1280, height = 720)

plotRes(gmo.bc, shortSeasonalDeterministic$model, 0.1, F)

#dev.off()

################################################

####Self defined function for Stochastic part###

################################################

acfpacf <- function(Data){

par(mfrow=c(1,2))

acf(Data, ci.type= "ma", main = 'ACF test', lag.max = 60)

pacf(Data, main = 'PACF test', lag.max = 60)

par(mfrow=c(1,1))

}

# addapted from Ditiatkovski M. time series assignment 2

# assumes time series as an input, returns ACF PACF and data plots

plotDiag<- function(data, ylab = ""){

x <- as.vector(time(data))

diffData <- ggplot(data = data, aes(x = x, y = as.vector(data))) +

geom\_line() +

geom\_point() +

xlab("Years") +

ylab(ylab) +

theme\_bw()

# create ACF plot

ACF <- ggAcf(data, lag.max = 48) +

theme\_bw() +

theme(plot.title = element\_blank())

# create PACF plot

PACF <- ggPacf(data, lag.max = 48) +

theme\_bw() +

theme(plot.title = element\_blank())

# return diagnostic plot

row2 <- ggarrange(ACF, PACF, labels = c("B","C"))

return(ggarrange(diffData, row2, labels = "A", nrow = 2))

}

# This function, given the time series data, ARIMA orders and seasonal orders will fit ARIMA models

# The list of models will then be analysed by another function to find the best one

modelFit <- function(ts, order, seasonal=c(0,0,0), lambda = 1, methods = c("ML","CSS","CSS-ML"), freq = 12){

modelList <- list()

# Try to fit the model with every single method by iterating though method list

for (i in methods){

# Error catch in case arima function fails

errorCheck <- tryCatch({

# Attempt to fit Model

model <- Arima(ts, order = order, lambda = lambda, method = i, seasonal=list(order = seasonal, period = 12))

# Get k of a model and error summary for fitting

k = sum(order, seasonal)

modelAccuracy <- as\_tibble(accuracy(model))

# make an entry to ModelList type of an object

modelList = rbind(modelList, list("order" = order,

"seasonal" = seasonal,

"model" = model,

"accuracy" = modelAccuracy,

"k" = k,

"method" = i))

}, error=function(er){

# if arima function fails show a warning, continue on to the next model

warning(paste0(c("ARIMA(", paste0(seasonal, collapse=","),")X(", paste0(order, collapse=","), ") with ", i, " method could not be fitted")))})

}

return(modelSelectError(modelList))

}

# expects a list models and a accuracy (from model\_fit()) and finds the model with the

# smallest error, default being MAE. Returns a list with that model specifications

modelSelectError <- function(modelList, error = "MAE"){

combAccuracy <- bind\_rows(modelList[,"accuracy"])

lowestError <- which(combAccuracy[error] == min(combAccuracy[error]))

return(modelList[lowestError,])

}

# Calculate BIC for Arima type model

BIC <- function(model, k){

return(-2\*model$loglik + k\*log(length(model$residuals)) %>% round(3))

}

# Calculate AIC for Arima type model

AIC <- function(model, k){

return(-2\*model$loglik + 2\*k %>% round(3))

}

# Model to fit models given a list of lists, the latter of which contain a single order named "seasonal" for seasonal

# data and a list named "order" for ARIMA orders to try for that seasonal model.

# Other inputs are needed to send to model\_fit function. Returns a list of models with their diagnostic and a table

modelListFit <- function(ts, orders, lambda = 1, methods = c("ML","CSS","CSS-ML"), freq = 12){

modelList = list()

modelCounter = 1

# outer loop to itterate through seasonal part of models

for(i in seq(1, length(orders))){

seasonal = orders[[i]]$seasonal

orderList = orders[[i]]$order

# Iterate through the normal Arima orders, keeping seasonal orders constant to fit all the models

for(j in orderList){

# for diagnostics

print(paste0(c("seas: ", seasonal, " Order: ", j), collapse = ""))

# some models fail spactacularly

errorCheck <- tryCatch({

# Find the best model for a particular order

temp <- modelFit(ts = ts,

order = j,

seasonal = seasonal,

lambda = lambda,

methods = methods,

freq = freq)

if(is.null(temp)){

next()

}

# add model, and model index to the modelList object

modelList <- rbind(modelList,

c(temp, "ModelIndex" = modelCounter))

modelCounter = modelCounter + 1

}, error=function(er){

# usually shows if all models fail.

warning(paste0(c("ARIMA(", paste0(seasonal, collapse=","),")X(", paste0(i, collapse=","), ") could not be fitted")))})

}

}

# the return fucntion finalises the modelList type output to contain all models and a summary table

return(list("ModelList" = modelList, "Table" = modelSummaryTable(modelList)))

}

# Create a summary table of the supplied models. Expects input generated in model fit,

# or combination of those in a list

modelSummaryTable <- function(modelList){

# setup data frame / columns for the table

BIC <- mapply(BIC, modelList[,"model"], modelList[,"k"])

AIC <- mapply(AIC, modelList[,"model"], modelList[,"k"])

accuracy <- bind\_rows(modelList[,"accuracy"])

# Get all orders

order <- Reduce(rbind, modelList[,"order"]) %>% as\_tibble() %>%

unite(., "ARIMA order", sep = ",")

seasonalOrder <- Reduce(rbind, modelList[,"seasonal"]) %>% as\_tibble() %>%

unite(., "Seasonal order", sep = ",")

# Add information about method and number of parameters

method <- Reduce(rbind, modelList[,"method"])[,1]

k <- Reduce(rbind, modelList[,"k"])[,1]

# Find model normality and proportion of coefficents that are significant

normality <- sapply(modelList[,"model"], residualNormPval)

propSigCoefs <- sapply(modelList[,"model"], significantCoefProp)

# Populate model index in the table, so it can easily taken from model list

index <- Reduce(rbind, modelList[,"ModelIndex"])[,1]

#make the table

results <- tibble(order,

seasonalOrder,

"k" = k,

"Method" = method,

"MASE" = accuracy$MASE,

"RMSE" = accuracy$RMSE,

"MAE" = accuracy$MAE,

"AIC" = AIC,

"BIC" = BIC,

"Res. Normality" = normality,

"Prop. Sig. coefs" = propSigCoefs,

"ModelIndex" = index)

return(results %>% arrange(`MASE`, `BIC`, desc(`Prop. Sig. coefs`)))

}

# Calculate p value for shapiro test of an Arima model

residualNormPval <- function(model){

return(shapiro.test(model$residuals)$p.value)

}

# Calculate proportion of significant coeficents in Arima model

significantCoefProp <- function(model){

coefs <- coeftest(model) %>% broom::tidy() %>% .$p.value

return(sum(coefs<0.05, na.rm = T)/length(coefs))

}

# This function is adapted from Ditiatkovski M, 2020, Timeseries Assignment 2

# Custom functions that plots diagnostic plots for model residuals

# expects original ts data and model as an input, vertical controls if the returned

# image is vertical (TRUE) or horizontal (FALSE). Kcalc controls if

# number of parameters are calculated for LBtest.

plotResArima <- function(data, model, title, vertical = T, kcalc = T) {

combTable <- tibble("time" = as.vector(time(data)),

"data" = data,

"residuals" = model$residuals,

"fitted" = fitted(model))

# get number of coefficients assumes ar fits all parameters

dump <- capture.output(model)

# if needed calculate k

if(kcalc){

k<- dump[2] %>% str\_extract\_all(., "[[:digit:]]") %>%

.[[1]] %>%

as.numeric() %>%

sum()

}else{

k = 0

}

lag.max <-if\_else(length(combTable$data)>40, 40, length(combTable$data)-1)

# Original vs fitted values

fittedPlot <- ggplot(data = combTable, aes(x = time, y = data)) +

geom\_line() +

geom\_line(aes(y = fitted), colour = "mediumpurple",

linetype = 2) +

geom\_point(size = 0.4) +

ylab(title) +

theme\_bw()

# residuals plotted

residualsPlot <- ggplot(data = combTable, aes(x = time, y = residuals))+

geom\_line() +

geom\_point(size = 0.2) +

theme\_bw() +

ylab("Standardised residuals")

# create ACF plot

ACF <- ggAcf(combTable$residuals, lag.max = lag.max) +

theme\_bw() +

theme(plot.title = element\_blank())

# create ACF plot

PACF <- ggPacf(combTable$residuals, lag.max = lag.max) +

theme\_bw() +

theme(plot.title = element\_blank())

# create qq plot for the residuals

qq <- ggplot(data = combTable, aes(sample = residuals)) +

stat\_qq() +

stat\_qq\_line(linetype = "dashed", colour = "mediumpurple") +

theme\_bw()

# create Ljung Box plot in ggplot

LBTPval <- FitAR::LjungBoxTest(combTable$residuals, k = 0,

lag.max = lag.max) %>%

as.tibble()

LBplot <- ggplot(LBTPval, aes(x = m, y = pvalue)) +

geom\_point() +

geom\_abline(aes(intercept = 0.05, slope = 0),

colour = "dodgerblue4", linetype = 2) +

xlab("Lag") +

ylab("p-value") +

theme\_bw()

## arrange everything nicely and add labels

if (vertical){

plotsComb = ggarrange(fittedPlot, residualsPlot, ACF, PACF,

qq, LBplot, labels = c("A","B","C","D","E","F"), ncol = 2, nrow = 3)

}else{

plotsComb = ggarrange(fittedPlot, residualsPlot, qq, ACF, PACF,

LBplot, labels = c("A","B","C","D","E","F"), ncol = 3, nrow = 2)

}

return(plotsComb)

}

# Returns coefficent diagnostics in a table for a supplied model

coefTable <- function(model){

return(coeftest(model) %>% round(., 4) %>% tidy())

}

# for use in presentation, modified from previous assignment

plotAcfPacf<- function(data, ylab){

time <- as.vector(time(data))

# Plot differenced data

diffData <- ggplot(data = data, aes(x = time, y = data)) +

geom\_line() +

geom\_point(size = 0.2) +

xlab("Years") +

ylab(ylab) +

theme\_bw()

# create ACF plot

ACF <- ggAcf(data, lag.max = 40) +

theme\_bw() +

theme(plot.title = element\_blank())

# create PACF plot

PACF <- ggPacf(data, lag.max = 40) +

theme\_bw() +

theme(plot.title = element\_blank())

# show the diagnostic plot

row2 <- ggarrange(ACF, PACF)

return(ggarrange(diffData, row2, nrow = 2))

}

#define a function to forecast based on the model and output the plot and data

forecastPlot <- function(Data, order, seasonal, lambda, number\_of\_period){

model = Arima(Data, order=order, seasonal = seasonal, lambda = lambda)

pred <- forecast(model, h = number\_of\_period)

print(pred)

return(autoplot(pred))

}

######################################################

# Code for tables and figures for Stochastic Section #

######################################################

#Plot Figure 7

#show the acf pacf against the transformed data

plotDiag(gmo.bc)

#test for normality of the transformed data

shapiro.test(gmo.bc)

#Plot figure 8

plotDiag(Arima(gmo.ts.short, order=c(0,0,0), seasonal= list(order = c(1,0,0), period = 12), lambda = 0.1)$residuals)

#Plot figure 9

plotDiag(Arima(gmo.ts.short, order=c(0,0,0), seasonal= list(order = c(0,1,0), period  = 12, lambda = 0.1))$residuals)

# applying a single difference appears to introduce an MA component to the seasonality, which may indicate overdifferencing

#Unit root test

ArimaS100Res <- Arima(gmo.ts.short, order=c(0,0,0), seasonal= list(order = c(1,0,0), period  = 12), lambda = 0.1)$residuals

adf.test(ArimaS100Res, k = ar(ArimaS100Res)$order)

#Plot figure 10

Arima010S100 <- Arima(gmo.ts.short, order=c(0,1,0), seasonal= list(order = c(1,0,0), period = 12), lambda = 0.1)$residuals

adf.test(Arima010S100, k = ar(Arima010S100)$order)

plotDiag(Arima010S100)

# Figure 11

eacf(Arima010S100)

# Figure 12

plot(armasubsets(Arima010S100, nar = 8, nma = 8)) # (6,1,1), (4,1,5)

#Seaonal sarima S(0,1,1)

ArimaS011Res <- Arima(gmo.ts.short, order=c(0,0,0), seasonal= list(order = c(0,1,1), period  = 12), lambda = 0.1)$residuals

#ADF test

adf.test(ArimaS011Res, k = ar(ArimaS011Res)$order)

#  Series is non stationary despite lack of trend. A difference can be applied

#ADF test on the differenced series

Arima010S011Res <- Arima(gmo.ts.short, order=c(0,1,0), seasonal= list(order = c(0,1,1), period = 12), lambda = 0.1)$residuals

adf.test(Arima010S011Res, k = ar(Arima010S011Res)$order)

#Figure 13

plotDiag(Arima010S011Res)

#Figure 14

eacf(Arima010S011Res)

# EACF is easier to interpret, possible models : (0,1,1), (0,1,2), (1,1,1), (1,1,2)

#Figure 15

plot(armasubsets(Arima010S011Res, nar = 7, nma = 7))

#Seaonal sarima S(3,1,1)

ArimaS311Res <- Arima(gmo.ts.short, order=c(0,0,0), seasonal= list(order = c(3,1,1), period  = 12), lambda = 0.1)$residuals

#ADF test

adf.test(ArimaS311Res, k = ar(ArimaS311Res)$order)

#ADF test on the differenced series

Arima010S311Res <- Arima(gmo.ts.short, order=c(0,1,0), seasonal= list(order = c(3,1,1), period = 12), lambda = 0.1)$residuals

adf.test(Arima010S311Res, k = ar(Arima010S311Res)$order)

#Figure 16

plotDiag(Arima010S311Res)

#Figure 17

eacf(Arima010S311Res)

#Figure 18

plot(armasubsets(Arima010S011Res, nar = 5, nma = 5))

########### Model Evaluation #####################

#Model setup

S311Models = list("seasonal"= c(3,1,1),

"order" = list(c(0,1,1), c(3,1,0), c(5,1,0),

c(3,1,1), c(5,1,1), c(0,1,2),

c(1,1,1), c(1,1,2)))

S011Models = list("seasonal"= c(0,1,1),

"order" = list (c(0,1,1), c(0,1,2), c(1,1,1), c(1,1,2),

c(3,1,0), c(3,1,1), c(0,1,3), c(5,1,0),

c(5,1,1) , c(5,1,3) ) )

S100MOdels = list("seasonal"= c(1,0,0),

"order" = list(c(0,1,1), c(0,1,2), c(1,1,2),

c(3,1,0), c(3,1,1), c(2,1,3),

c(6,1,1), c(4,1,5)))

# combine the models

allModelsOrders = list(S311Models, S011Models, S100MOdels)

# Generate all models

allModelFitted = modelListFit(gmo.ts.short, allModelsOrders, lambda = 0.1)

#Produce all tables

AllTable <- allModelFitted$Table

# original data

ggsave("originalDataPlotAcfPacf.png", plotAcfPacf(gmo.ts, "Number of Apps."),width = 128, height = 72, units = "mm",

device = "png")

# seasonal diff

ggsave("SeasonalDiffPlotAcfPacf.png", plotAcfPacf(diff(gmo.ts, lag = 12), "Number of Apps."), width = 128, height = 72, units = "mm",

device = "png")

#Write to csv all models and their residual values into a CV

write.table(allModelFitted$Table, "AllModelsDiagnostics.csv", sep = ",", row.names = FALSE)

#Get top 10 models

modelInd <- allModelFitted$Table$ModelIndex[1:10]

top10models <- allModelFitted$ModelList[modelInd,]

############# use this to save residual diagnostic plots, chainge model index as required

# width and hight are just a suggestion.

modelIndex = 1

png("testHoriz.png", width = 1280, height = 720)

plotResArima(gmo.ts.short, top10models[[modelIndex,"model"]], "Number of Apps.", F)

dev.off()

png("testVertical.png", width = 1280, height = 720)

plotResArima(gmo.ts.short, top10models[[modelIndex,"model"]], "Number of Apps.", vertical = T, kcalc = F)

dev.off()

############

for( i in seq(1, 10)){

order <- paste0(top10models[[i, "order"]], collapse = ",")

seasonal <- paste0(top10models[[i, "seasonal"]], collapse = ",")

method <- top10models[[i, "method"]]

fileString <- paste0("ARIMA(",order,")X(",seasonal,") ", method)

png(paste0(fileString, "Horiz.png"),

width = 1280, height = 720)

print(plotResArima(gmo.ts.short, top10models[[i,"model"]], "Number of Apps.", F))

dev.off()

png(paste0(fileString, "Vert.png"),

width = 1280, height = 720)

print(plotResArima(gmo.ts.short, top10models[[i,"model"]], "Number of Apps.", T))

dev.off()

write.table(coefTable(top10models[[i,"model"]]), paste0(fileString, "Coefs.csv"),sep = ",", row.names = FALSE)

}

#overfitting

# we fit 3 final models

# 1. our best one (2,1,3)\*(1,0,0)

# 2. Overfit 1 by increasing AR (3,1,3)\*(1,0,0)

# 3. Overfit 1 by increasing AR (2,1,4)\*(1,0,0)

Finalmodels = list(list("seasonal"= c(1,0,0),

"order" = list( c(2,1,3), c(3,1,3), c(2,1,4))))

# Generate all models

FinalModelFitted = modelListFit(gmo.ts.short, Finalmodels, lambda = 0.1)

#Produce all tables

FinalTable <- FinalModelFitted$Table

FinalTable

#original model is better on MASE, MAE, AIC,BIC and sig coefs

modelInd\_final <- FinalTable$ModelIndex[2:3]

overfitmodels <- FinalModelFitted$ModelList[modelInd\_final,]

# printing coefs if needed

for( i in seq(1, 2)){

order <- paste0(overfitmodels[[i, "order"]], collapse = ",")

seasonal <- paste0(overfitmodels[[i, "seasonal"]], collapse = ",")

method <- overfitmodels[[i, "method"]]

fileString <- paste0("ARIMA(",order,")X(",seasonal,") ", method)

png(paste0(fileString, "Horiz.png"),

width = 1280, height = 720)

print(plotResArima(gmo.ts.short, overfitmodels[[i,"model"]], "Number of Apps.", F))

dev.off()

png(paste0(fileString, "Vert.png"),

width = 1280, height = 720)

print(plotResArima(gmo.ts.short, overfitmodels[[i,"model"]], "Number of Apps.", T))

dev.off()

write.table(coefTable(overfitmodels[[i,"model"]]), paste0(fileString, "Coefs.csv"),sep = ",", row.names = FALSE)

}

# we can choose our best model ARIMA(2,1,3)\*(1,0,0)\_12

#Forecasting with the model

forecastPlot(gmo.ts.short,order = c(2,1,3), seasonal = c(1,0,0), lambda = 0.1, number\_of\_period = 10)