# 

**DSA301 Time Series Data Analysis  
Term Group Project**

|  |
| --- |
| **Group Members** |
| \*Censored\* |
| \*Censored\* |
| Chong Gao Zhe |
| \*Censored\* |
| \*Censored\* |
| \*Censored\* |

[**Abstract 3**](#_66ctswst94ma)

[**Introduction 3**](#_6d7jts7u53gw)

[MGM Resorts 3](#_4d4no4tojjv1)

[Objective 4](#_8obwwp14be7h)

[Scope and Limitations 5](#_d2jl7iixait2)

[**Data Description and Transformation 6**](#_hr0x2jnepokt)

[Log Transformation 6](#_yi1nxlorjrcl)

[Box-cox transformation - Omitted 7](#_t8ad32r8pxqt)

[Differencing 7](#_ehext2218093)

[Split time series object into training and testing partitions 8](#_tcaog48h9fpp)

[**Benchmark Models 10**](#_6c1z18r5ciwt)

[Determination of decomposition method 10](#_d4wngmgzrit1)

[Extracting seasonal and trend cycle component 11](#_uookoekg9h3)

[Potential decomposition of models 11](#_jc32hmrnxgbx)

[Ljung Box test 14](#_ofsntl51ihvi)

[Out of sample performance 17](#_7l88i0n7a75p)

[**Arima Model 19**](#_jok5lce8l4mg)

[In sample performance and residual analysis 22](#_p52jwbaevqnm)

[Out of sample performance and residual analysis 23](#_p8tpejnrvy8j)

[**ARIMA-X Model 24**](#_pxdfwhdx9h3)

[Identify Exogenous Variables 24](#_ljhck9b1nw6l)

[Granger Causality Test 26](#_522twnlpo0t2)

[Data Transformation of Exogenous Variables 28](#_scw7ab3asbp8)

[Building of ARIMA-X Model 30](#_l6iy80t7ksrp)

[ARIMA-X model with all exogenous variable (Inconsistent with our intuition) 30](#_a4up0cosc4pg)

[ARIMA-X model with only US Federal Reserve Rates (Ambiguous Variable) 32](#_qwhhopydlbty)

[**Prophet Model 34**](#_d0hscasfxrpt)

[**Cubic Spline 37**](#_iijlopmyzrlf)

[**Conclusion 38**](#_w7dz78siym91)

# **Abstract**

This report forecasts MGM revenue using various models. The data was pre-processed to ensure covariance stationarity through logarithmic transformation. The best performing benchmark model was STL Decomposition with Naive & Snaive, with an MAPE of 116.34. An ARIMA(2,1,1) model was selected using AICc, with an out of sample MAPE of 3.77. Attempts to use exogenous variables in ARIMA-X models were unsuccessful due to the absence of Granger-Causality and violation of regression assumption. A prophet model was built to consider seasonality and holiday effects, and a cubic spline model was built to capture nonlinear relationships.

# **Introduction**

The aim of this project is to apply the various time series techniques, from pre-modeling data transformations to modeling and evaluation, on the corporate revenue of MGM (Metro-Goldwyn-Mayer) Resorts. The idea behind choosing the corporate revenue of MGM as the main subject of this project is the hypothesis that MGM’s revenue exhibits seasonalities as the company operates facilities such as hotels and casinos that participate in the tourism industry.

## MGM Resorts

MGM Resorts is a leading global hospitality and entertainment company that operates luxury hotels and casinos in some of the world's most popular tourist destinations. The company has a significant presence in both the United States and Asia, and its performance in these regions has been instrumental in its overall success.  
  
In the United States, MGM Resorts has a strong track record of growth and revenue generation. The company's flagship property, the MGM Grand in Las Vegas, is one of the largest and most profitable hotels and casinos in the world. Additionally, MGM Resorts owns and operates several other iconic properties in the US, including the Bellagio, Mandalay Bay, and The Mirage.  
  
In Asia, MGM Resorts has made significant investments in recent years to expand its presence and capture a share of the region's rapidly growing tourism market. The company's flagship property in Asia is the MGM Cotai in Macau, which opened in 2018 and is one of the largest integrated resorts in the region. In addition to Macau, MGM Resorts also operates a property in Vietnam and has announced plans to develop a resort in Osaka, Japan.  
  
Overall, MGM Resorts has demonstrated strong revenue growth and expansion in both the United States and Asia, driven by a combination of successful operations and strategic investments.

## Objective

Our main research question is which time series models are the best for forecasting MGM’s Revenue. We also aim to achieve a good comparison of different time series forecasting methods to find out which of the time series methods has the best prediction accuracy for the investigated data set.

|  |  |
| --- | --- |
| **O1** | Investigate the dataset |
| **O1.1** | Data processing |
| **O2** | Develop benchmark models and evaluate on its prediction accuracy |
| **O3** | Build an Arima model and evaluate its prediction accuracy |
| **O4** | Explore by building a Vector Autoregression model to check for two way granger causality/ambiguity |
| **O5** | Build a Arima-X model and evaluate its prediction accuracy |
| **O6** | Build a Prophet model and evaluate its prediction accuracy |
| **O7** | Build a Cubic Splines model and evaluate its prediction accuracy |

Note: Evaluation of prediction accuracy will be based on the out-of-sample Mean Average Percentage Error (MAPE). The lower the MAPE, the closer the predicted values are to the actual values, and thus the better the prediction accuracy.  
 **Objective 1, 1.1,** we have to investigate the data set and extrapolate the properties and the characteristics of the data set. This is crucial as we need to know the properties of the data set in order to know which forecasting algorithms can be used for the data set. We need to know if the data set is stationary, non-stationary, follow a trend or seasonality etc.

**Objective 2,** requires us to build a benchmark model, as it is an important step in evaluating the predictive abilities of more advanced time series methods. The purpose of a benchmark model in time series analysis is to establish a baseline level of accuracy that a more complex model must exceed in order to be considered an improvement over the baseline. By comparing the performance of more advanced models against the benchmark model, analysts can determine whether the additional complexity and computational cost of a more advanced model is justified by its improved performance. This can help to avoid overfitting or underfitting of the model.  
  
**Objective 3,** requires us to investigate the predictive potential of utilizing the MGM’s historical revenue data by capturing complex trends and seasonality of the data.   
  
**Objective 4,** building a VAR model can be described as a process of exploratory data analysis as it helps us identify if the variables granger causes to one another. Moreover, we seek to use the results from VAR models to draw conclusions from our economic intuition on the exogenous variable that we have identified and reinforce the argument of developing an ARIMA-X model.

**Objective 5,**  Build an ARIMA-X model with identified exogenous variables that can help improve predictive accuracy of the model

**Objective 6,** Build a prophet model that was innovated by Meta (formerly named Facebook) which works best with time series that have strong seasonality, and with a few years of historical data.

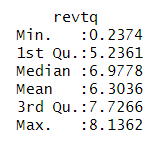
**Objective 7,** Cubic spline models have several advantages over the earlier types of time series models. They are flexible and can capture non-linear relationships between the variables, and they can handle missing data and irregularly spaced observations

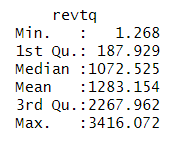
## Scope and Limitations

This report will cover models and techniques that are covered in class; therefore we will not investigate other more advanced models that we have yet to learn. Furthermore, the analysis is limited to supervised machine learning techniques and does not include LSTMs. Additionally, all predictions in this project is the log-revenue of MGM. One limitation of working with log-revenues is the inaccuracy of the prediction intervals. As such all prediction intervals in this project should be ignored.

# 

# **Data Description and Transformation**

Data for the corporate revenue of MGM Resorts is obtained from Capital IQ in quarterly intervals. The range of the data is from 1987 Q3 to 2022 Q3, with a total of 141 data points. For our project, we decided to normalize the variation of revenue via log transformation. These screenshots show us how the variation of the data decreased after taking the logarithm of the data.  
  
Raw values After log transformation

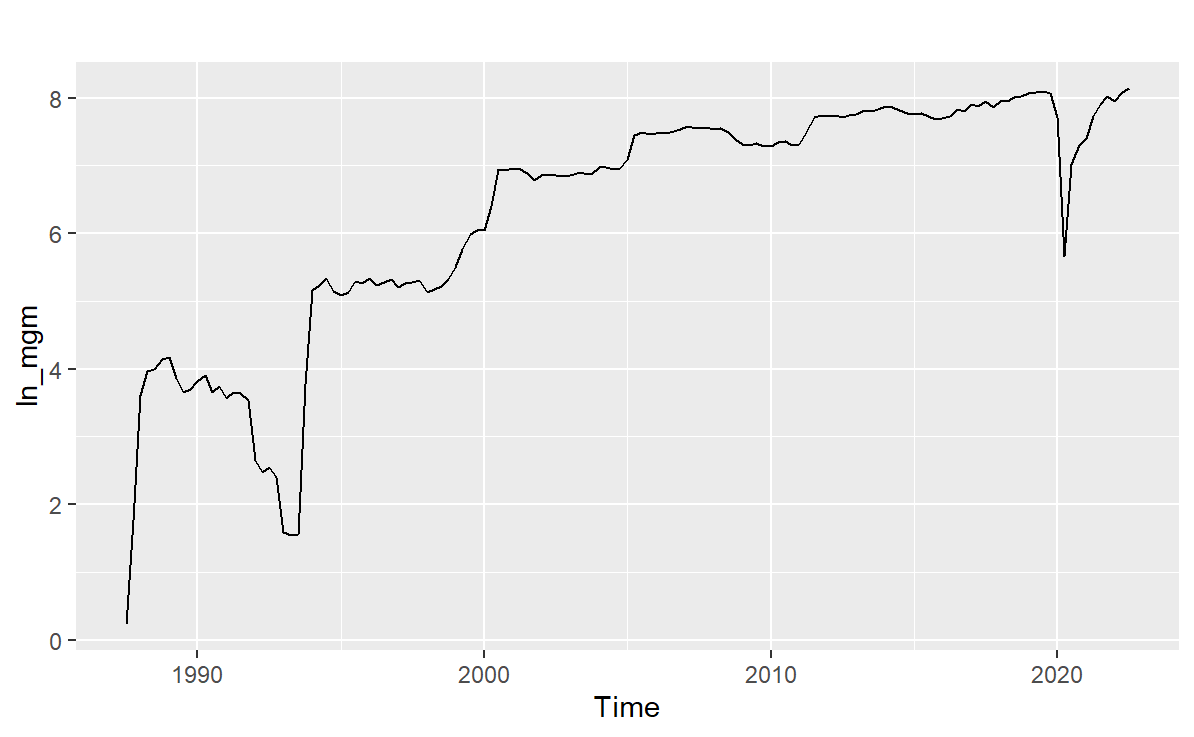
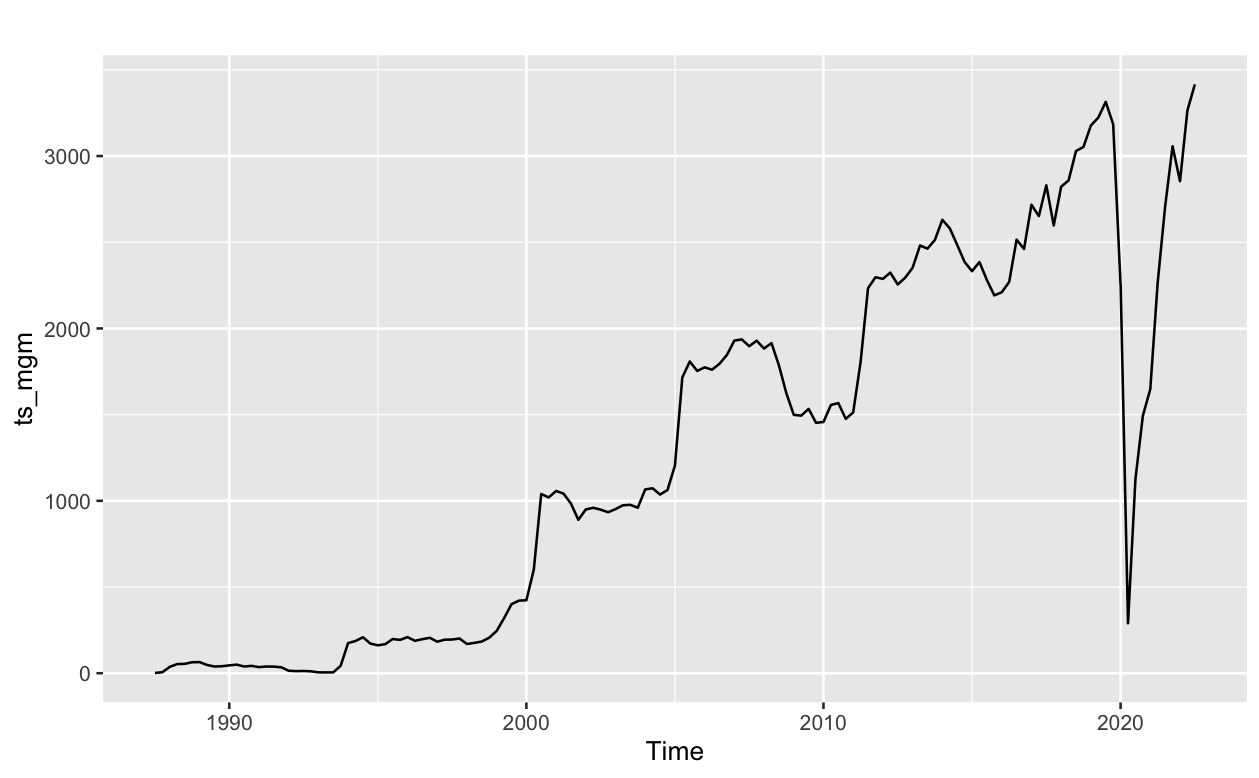


## Log Transformation

As the revenue of MGM Resorts seems to exhibit large variations in magnitude over time, by logging the variable, we can reduce the impact of extreme values and make the data more manageable for analysis. Moreover, log revenue has interesting statistical properties such as to transform the distribution to be more symmetrical and normally distributed, which makes it easier to identify patterns and trends in the data. However, performing log-transformations will render our prediction intervals.  
  
*ln\_mgm <- log(ts\_mgm)*

*​​*

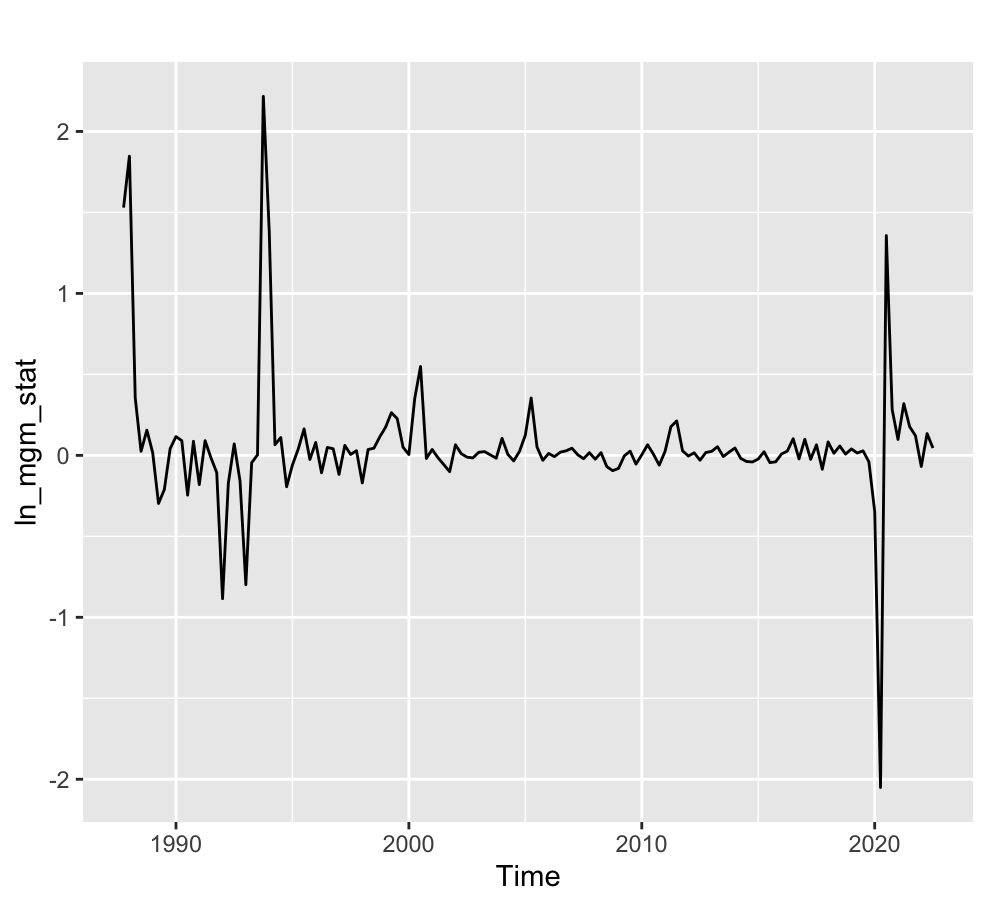
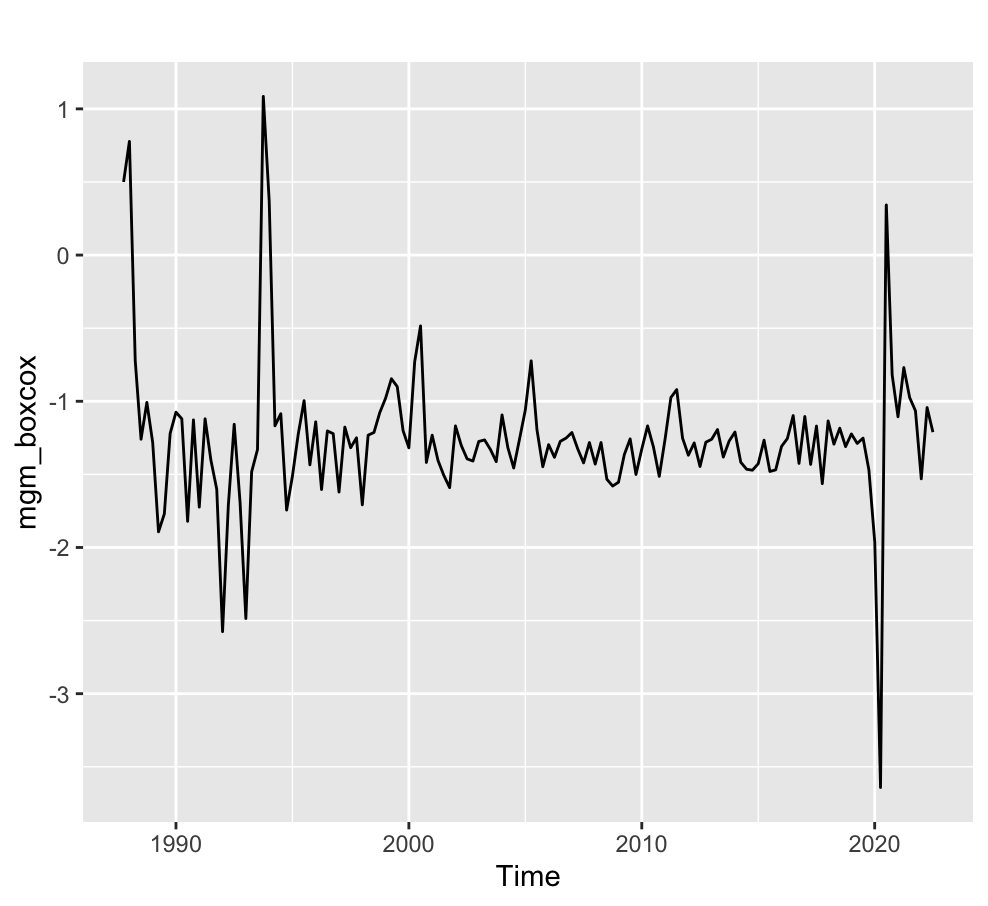
*Revenue log(revenue)*



It was hypothesized that given MGM’s industry, it should exhibit seasonality due to tourism patterns, holiday seasons, weather patterns, sporting events, etc. Contrary to our hypothesis, the ACF shows no seasonality in MGM’s revenue. A plausible explanation would be the interval of the data, quarterly data can be more difficult to capture seasonalities than monthly or weekly data.

## Box-cox transformation - Omitted

Box Cox transformation was not used as it does not seem to have a significant impact on normalizing the data set. A large number of transformations will only result in difficulty in interpreting the solutions. Hence, no box cox transformation is used in this project.

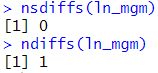
 

## Differencing

The log revenue of MGM undergoes differencing with the first lag value of itself to achieve stationarity.

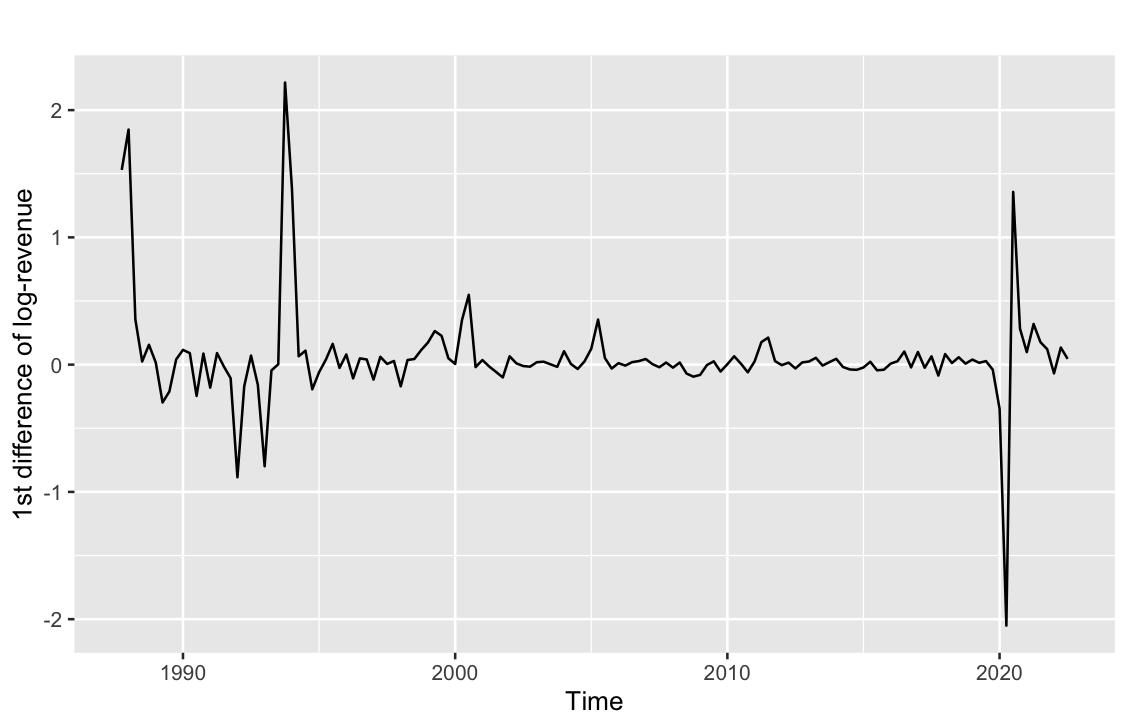
*nsdiffs(ln\_mgm)*

*ndiffs(ln\_mgm)*



*ln\_mgm\_stat <- diff(ln\_mgm)*

*autoplot(ln\_mgm\_stat)*

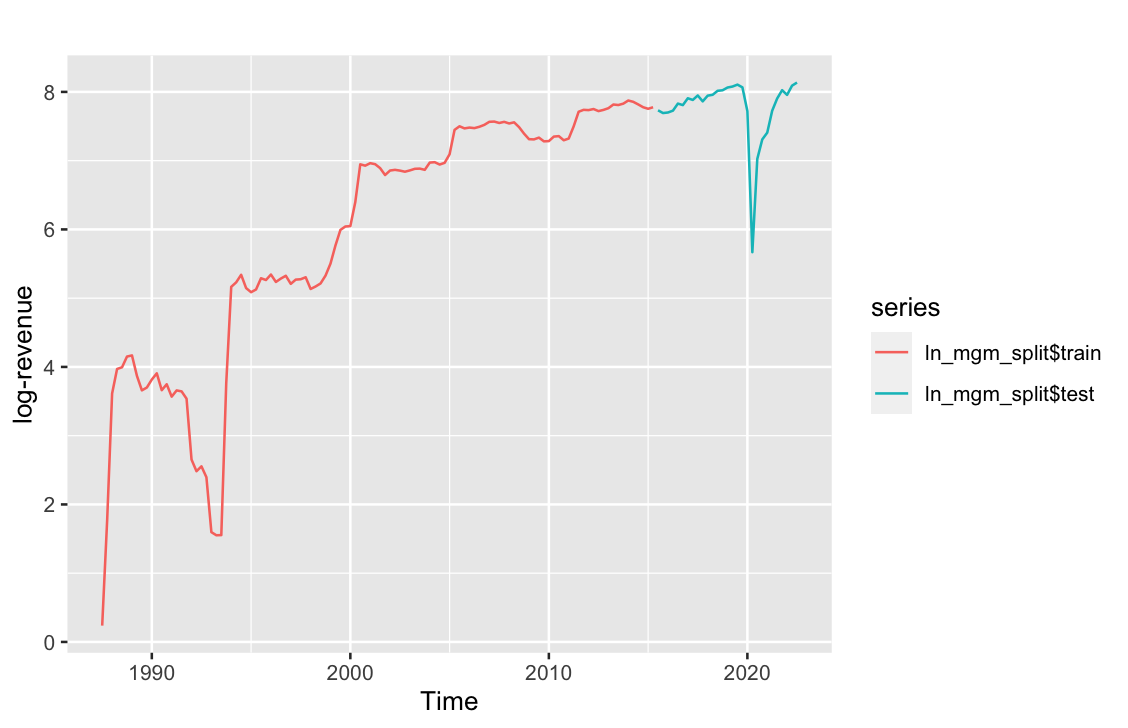


## Split time series object into training and testing partitions

To prepare for in-sample testing and out-sample testing later on in the report, the differenced log revenue of MGM is split into testing and training sets consisting of 112 and 29 data points respectively. This yields a testing size of 20% of the original data.

*outofsamplequarters = 29*

*ln\_mgm\_stat\_split = ts\_split(ln\_mgm\_stat, sample.out = outofsamplequarters) # Differenced Log-revenue of MGM split into training and test set  
ln\_mgm\_split <- ts\_split(ln\_mgm, sample.out = outofsamplequarters) # Log-revenue of MGM split into training and test set*



# 

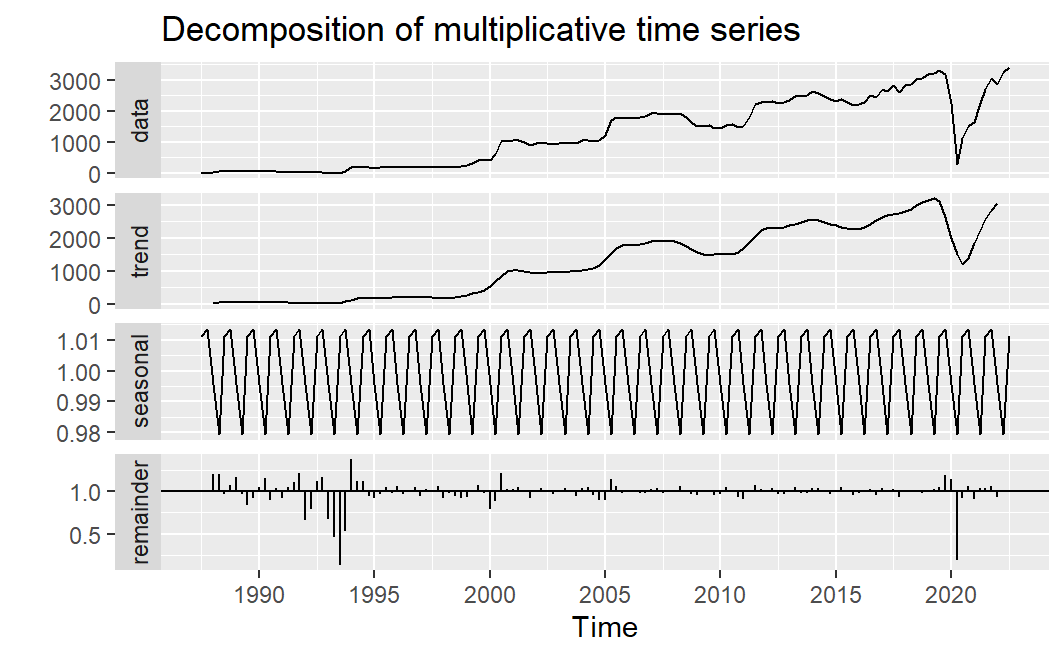
# **Benchmark Models**

The first step for model building is to come up with benchmark models. The purpose of a benchmark model in time series analysis is to establish a baseline level of accuracy that a more complex model must exceed in order to be considered an improvement over the baseline. Benchmark models are simple and easy to construct, while providing reasonably accurate results.

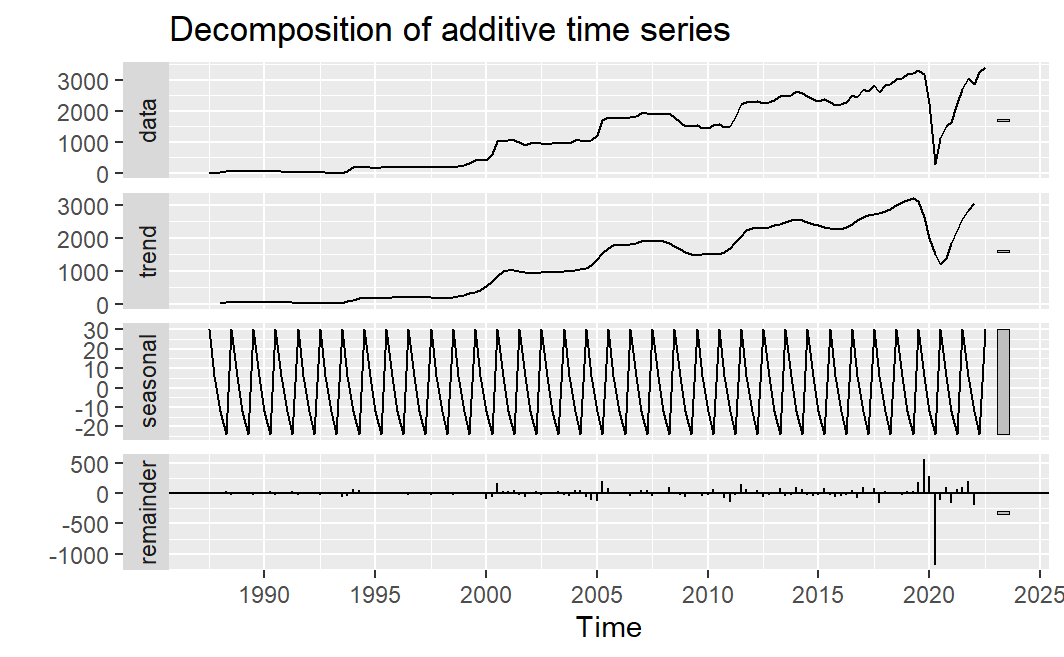
## Determination of decomposition method

Firstly, to determine the type of decomposition we are going to use, we plot the graphs of each decomposition method.

*autoplot(decompose(ts\_mgm, type=multiplicative))*



*autoplot(decompose(ts\_mgm, type=additive)*



From the graph, the amplitude appears to be stationary over time, hence additive decomposition is more suitable. Furthermore, we use stl decomposition as it brings about several advantages such as its functionality in handling any seasonal period, robust to outliers, can control rate of change in seasonal component and smoothness of trend-cycle component.

## Extracting seasonal and trend cycle component

There are two methods used to extract the seasonal and trend cycle components from a time series data: Classical decomposition and STL decomposition. This proposal considers both decomposition methods to decompose MGM’s revenue into the seasonal component and trend cycle component.

Classical additive decomposition:

*seasonalcomponent = seasonal(decompose(ln\_mgm\_stat\_split$train))*

*trendcyclecomponent = (trendcycle(decompose(ln\_mgm\_stat\_split$train)))*

Automated STL decomposition:

*seasonalcomponent2 = seasonal(mstl(ln\_mgm\_stat\_split$train))*

*trendcyclecomponent2 = (trendcycle(mstl(ln\_mgm\_stat\_split$train)))*

## Potential decomposition of models

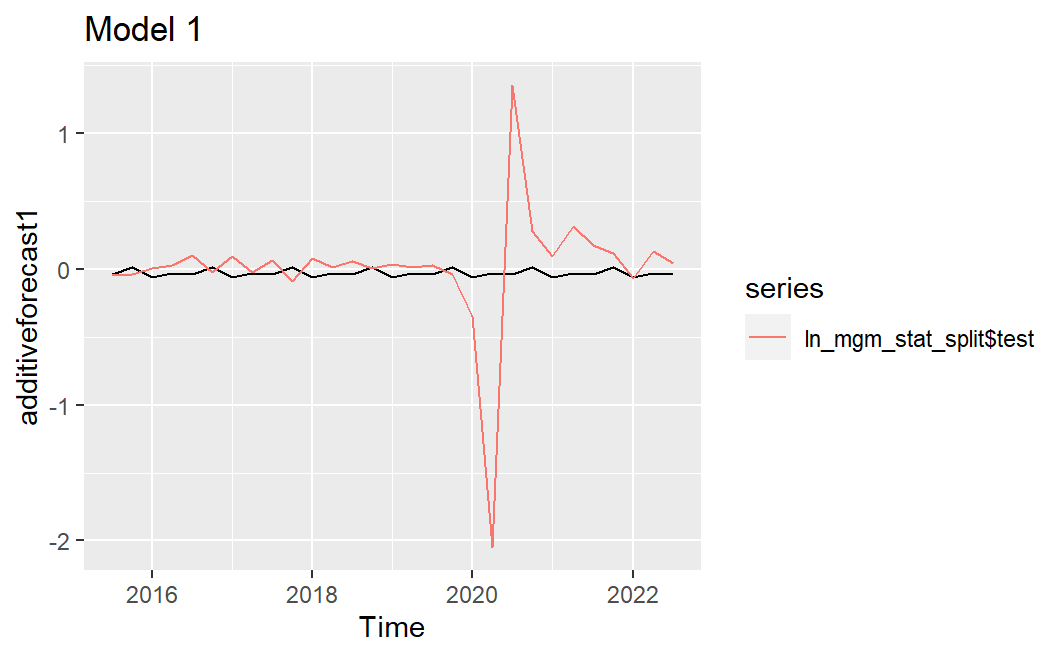
This proposal considers five possible benchmark models:

|  |  |  |  |
| --- | --- | --- | --- |
| Model no. | Trend Cycle | Seasonal | Decomposition Method |
| 1 | Naive | Seasonal Naive | Classical Decomposition |
| 2 | Mean | Seasonal Naive | Classical Decomposition |
| 3 | Drift | Seasonal Naive | Classical Decomposition |
| 4 | Naive | Seasonal Naive | STL Decomposition |
| 5 | Drift | Seasonal Naive | STL Decomposition |

1. Model 1: Classical Decomposition, naive + snaive mode

*additiveforecast1 = (snaive(seasonalcomponent, h = outofsamplequarters))$mean + (naive(trendcyclecomponent, h = outofsamplequarters))$mean*

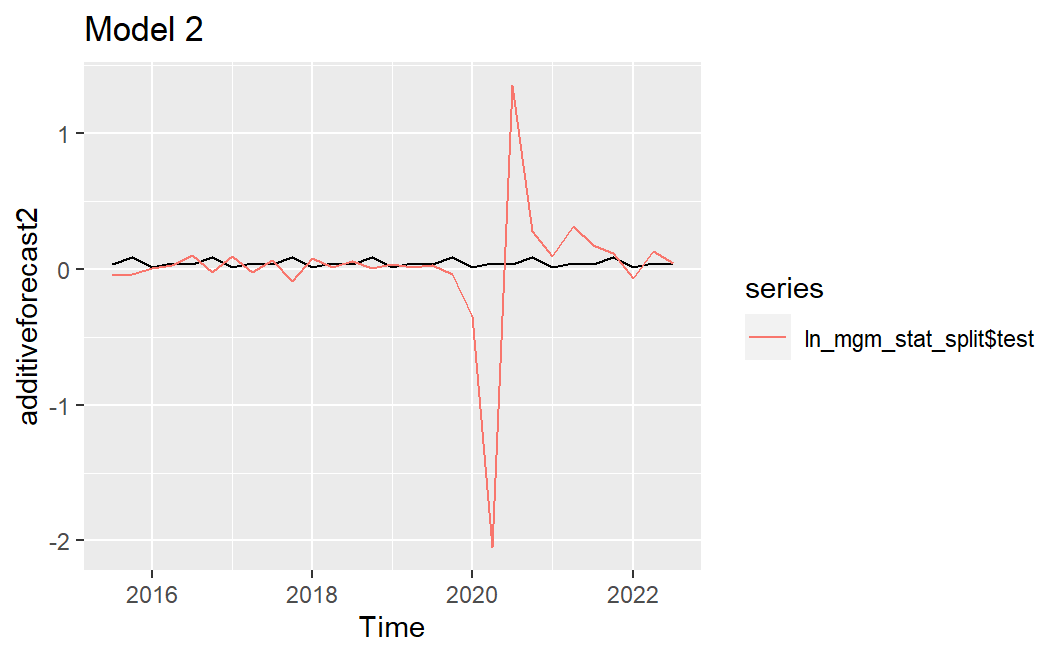
*autoplot(additiveforecast1) + autolayer(ln\_mgm\_stat\_split$test)*



1. Model 2: Classical Decomposition, mean + snaive model

*additiveforecast2 = (snaive(seasonalcomponent, h = outofsamplequarters))$mean + (meanf(trendcyclecomponent, h = outofsamplequarters))$mean*

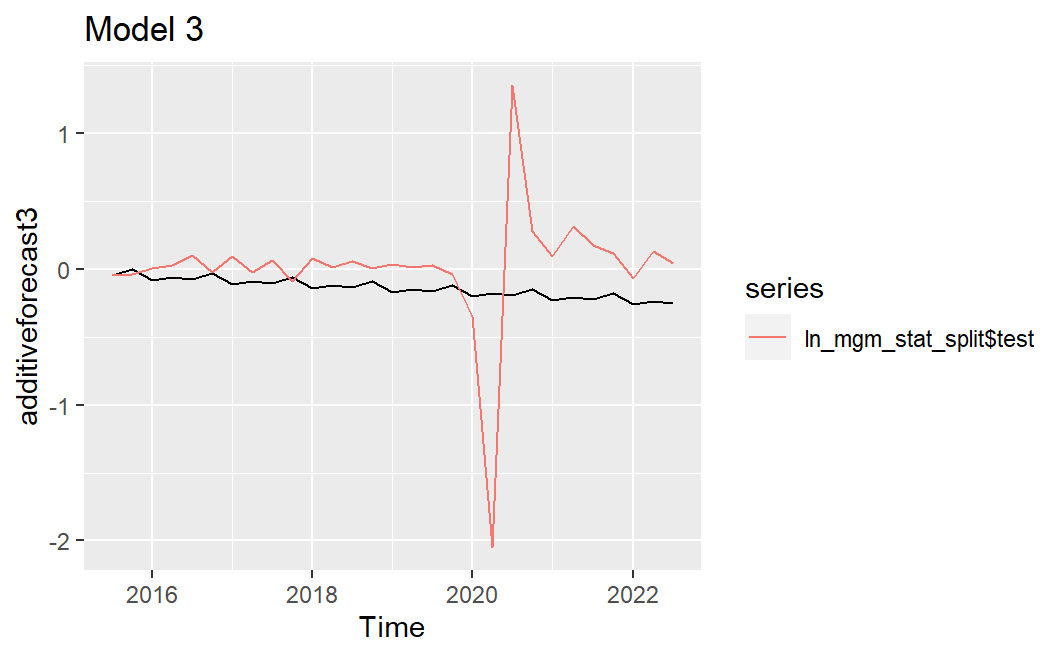
*autoplot(additiveforecast2) + autolayer(ln\_mgm\_stat\_split$test)*



1. Model 3: Classical Decomposition, drift + snaive model

*additiveforecast3 = (snaive(seasonalcomponent, h = outofsamplequarters))$mean + (rwf(trendcyclecomponent, h = outofsamplequarters, drift=TRUE))$mean*

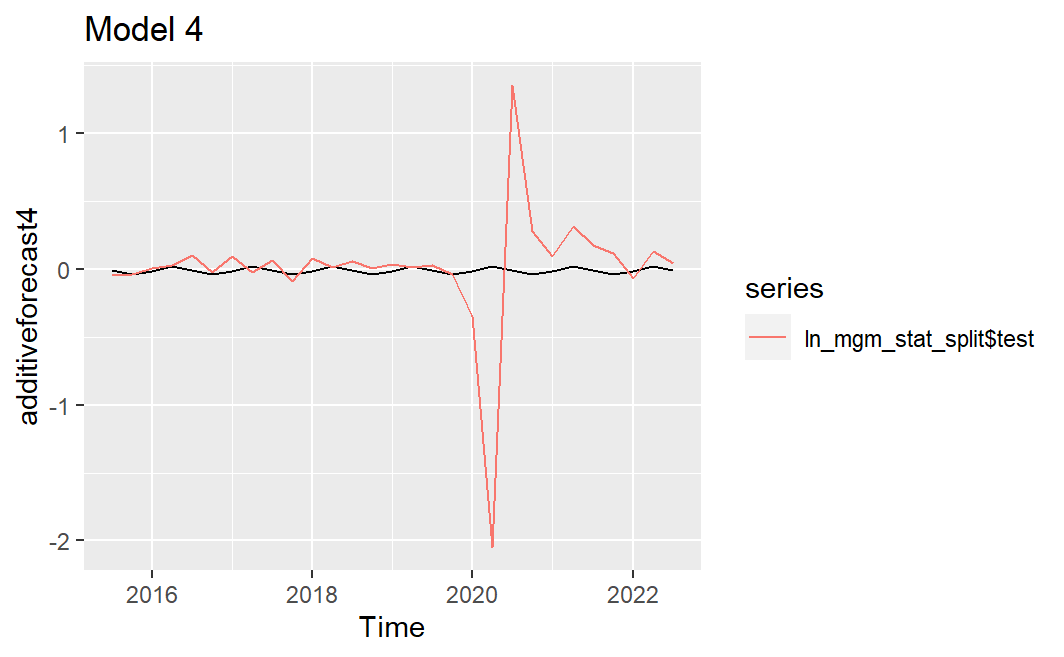
*autoplot(additiveforecast3) + autolayer(ln\_mgm\_stat\_split$test)*



1. Model 4: STL Decomposition, naive + snaive

*additiveforecast4 = (snaive(seasonalcomponent2, h = outofsamplequarters))$mean + (naive(trendcyclecomponent2, h = outofsamplequarters))$mean*

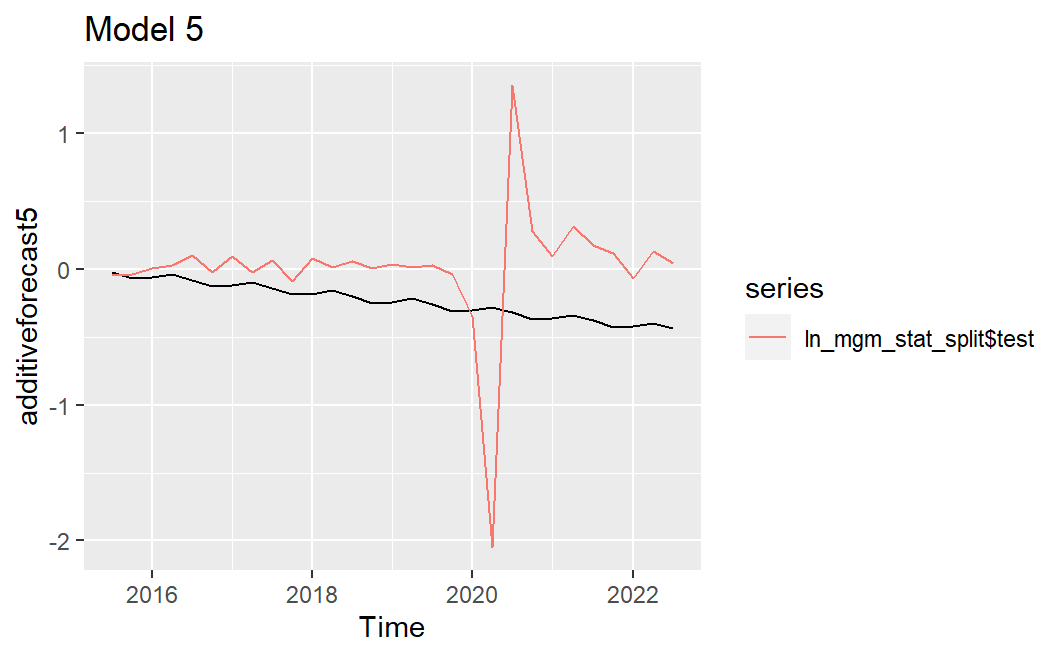
*autoplot(additiveforecast4) + autolayer(ln\_mgm\_stat\_split$test)*



1. Model 5: STL Decomposition, drift + snaive model

*additiveforecast5 = (snaive(seasonalcomponent2, h = outofsamplequarters))$mean + (rwf(trendcyclecomponent2, h = outofsamplequarters, drift=TRUE))$mean*

*autoplot(additiveforecast5) + autolayer(ln\_mgm\_stat\_split$test)*



## Ljung Box test

Extraction of seasonal and trend cycle for Ljung Box test

Classical Additive Decomposition:

*seasonalcomponent\_ln\_ts\_mgm = seasonal(decompose(ln\_mgm\_stat, type = decomptype))*

*seasadjcomponent\_ln\_ts\_mgm = (trendcycle(decompose(ln\_mgm\_stat, type = decomptype)))*

Automated STL Decomposition:

*seasonalcomponent\_ln\_ts\_mgm\_2 = seasonal(mstl(ln\_mgm\_stat))*

*seasadjcomponent\_ln\_ts\_mgm\_2 = (trendcycle(mstl(ln\_mgm\_stat)))*

Calculating residuals to test for the presence of time series information

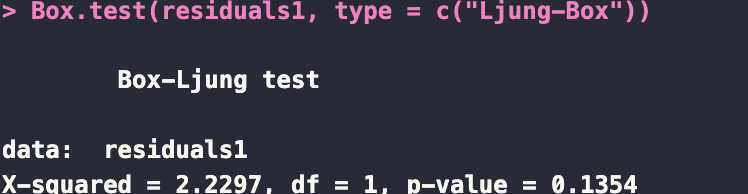
1. Model 1:

*fitted1 = (snaive(seasonalcomponent\_ln\_ts\_mgm))$fitted + (naive(seasadjcomponent\_ln\_ts\_mgm))$fitted*

*autoplot(fitted1) + autolayer(ln\_mgm\_stat)*

*residuals1 = ln\_mgm\_stat- fitted1*

*Box.test(residuals1, type = c("Ljung-Box"))*



Since p-value=0.1354 > 0.05, this report fail to reject the null hypothesis at a 5% significance level and conclude that there is no time series information present in residuals, hence this report can consider this benchmark model

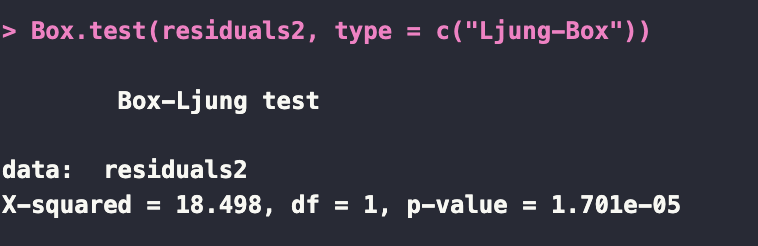
1. Model 2:

*fitted2 = (snaive(seasonalcomponent\_ln\_ts\_mgm))$fitted + (meanf(seasadjcomponent\_ln\_ts\_mgm))$fitted*

*autoplot(fitted2) + autolayer(ln\_mgm\_stat)*

*residuals2 = ln\_mgm\_stat - fitted2*

*Box.test(residuals2, type = c("Ljung-Box"))*



Since p-value= 1.701e^-5 < 0.05, this report reject the null hypothesis at a 5% significance level and conclude that there is time series information present in residuals, hence this report cannot use this benchmark model

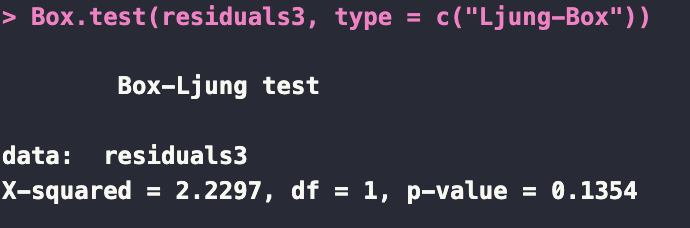
1. Model 3:

*fitted3 = (snaive(seasonalcomponent\_ln\_ts\_mgm))$fitted + (rwf(seasadjcomponent\_ln\_ts\_mgm, drift = TRUE))$fitted*

*autoplot(fitted3) + autolayer(ln\_mgm\_stat)*

*residuals3 = ln\_mgm\_stat - fitted3*

*Box.test(residuals3, type = c("Ljung-Box"))*



Since p-value=0.1354 > 0.05, this report fail to reject the null hypothesis at a 5% significance level and conclude that there is no time series information present in residuals, hence this report can consider this benchmark model

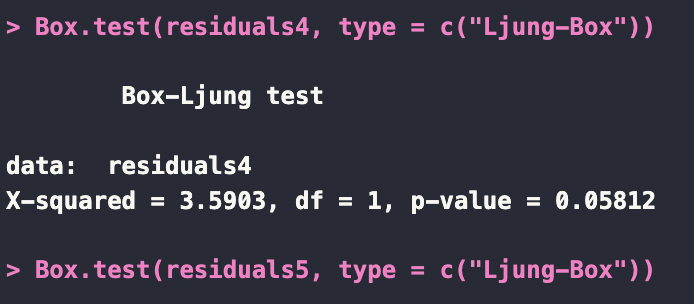
1. Model 4:

*fitted4 = (snaive(seasonalcomponent\_ln\_ts\_mgm\_2))$fitted + (naive(seasadjcomponent\_ln\_ts\_mgm\_2))$fitted*

*autoplot(fitted4) + autolayer(ln\_mgm\_stat)*

*residuals4 = ln\_mgm\_stat - fitted4*

*Box.test(residuals4, type = c("Ljung-Box"))*



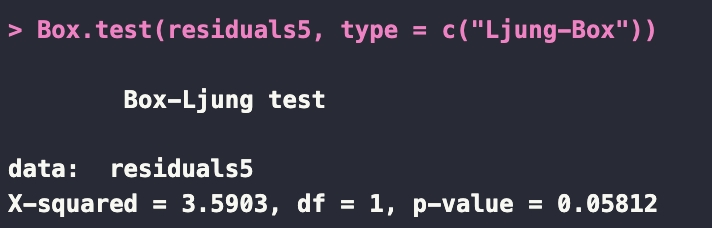
Since p-value=0.05812 > 0.05, this report fail to reject the null hypothesis at a 5% significance level and conclude that there is no time series information present in residuals, hence this report can consider this benchmark model

1. Model 5:

*fitted5 = (snaive(seasonalcomponent\_ln\_ts\_mgm\_2))$fitted + (rwf(seasadjcomponent\_ln\_ts\_mgm\_2 , drift = TRUE))$fitted*

*autoplot(fitted5) + autolayer(ln\_mgm\_stat)*

*residuals5 = ln\_mgm\_stat - fitted5*

**

*Box.test(residuals5, type = c("Ljung-Box"))*

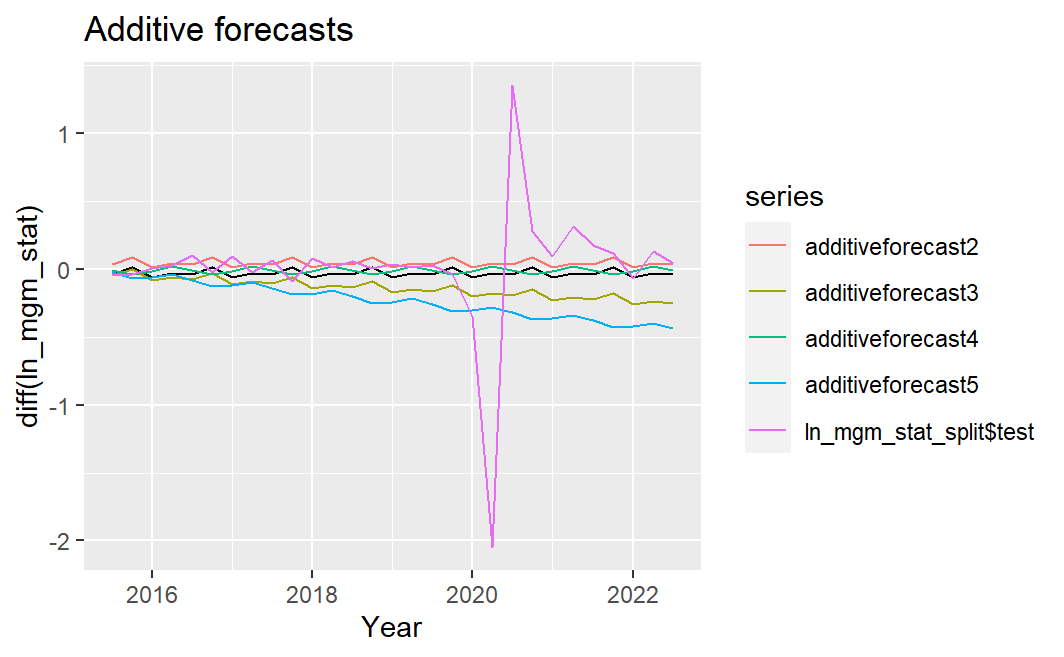
Since p-value=0.05812 > 0.05, this report fail to reject the null hypothesis at a 5% significance level and conclude that there is no time series information present in residuals, hence this report can consider this benchmark model

Since model 1,3,4 and 5 have no time series information present in the residuals, we can then proceed to check the out of sample performance to choose the best model.

## Out of sample performance

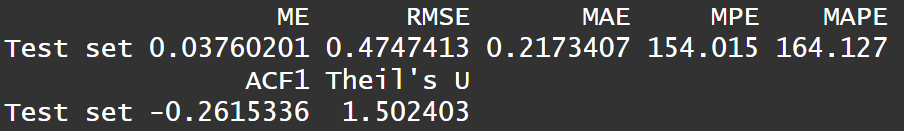
We first plot the graphs of each model against our insample data.

*autoplot(additiveforecast1) + autolayer(additiveforecast2) + autolayer(additiveforecast3) + autolayer(additiveforecast4) + autolayer(additiveforecast5) + autolayer(ln\_mgm\_stat\_split$test) + ggtitle("Additive forecasts") + xlab("Year") + ylab("diff(ln\_mgm\_stat)")*

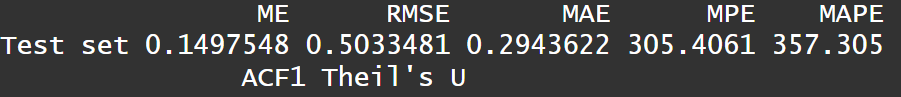


Accuracy tests:

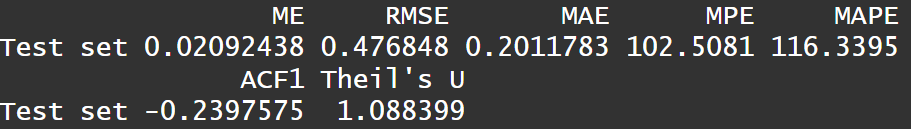
*accuracy(additiveforecast1, x = ln\_mgm\_stat\_split$test)*



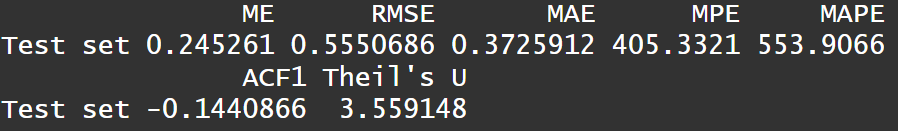
*accuracy(additiveforecast3, x = ln\_mgm\_stat\_split$test)*



*accuracy(additiveforecast4, x = ln\_mgm\_stat\_split$test)*



*accuracy(additiveforecast5, x = ln\_mgm\_stat\_split$test)*

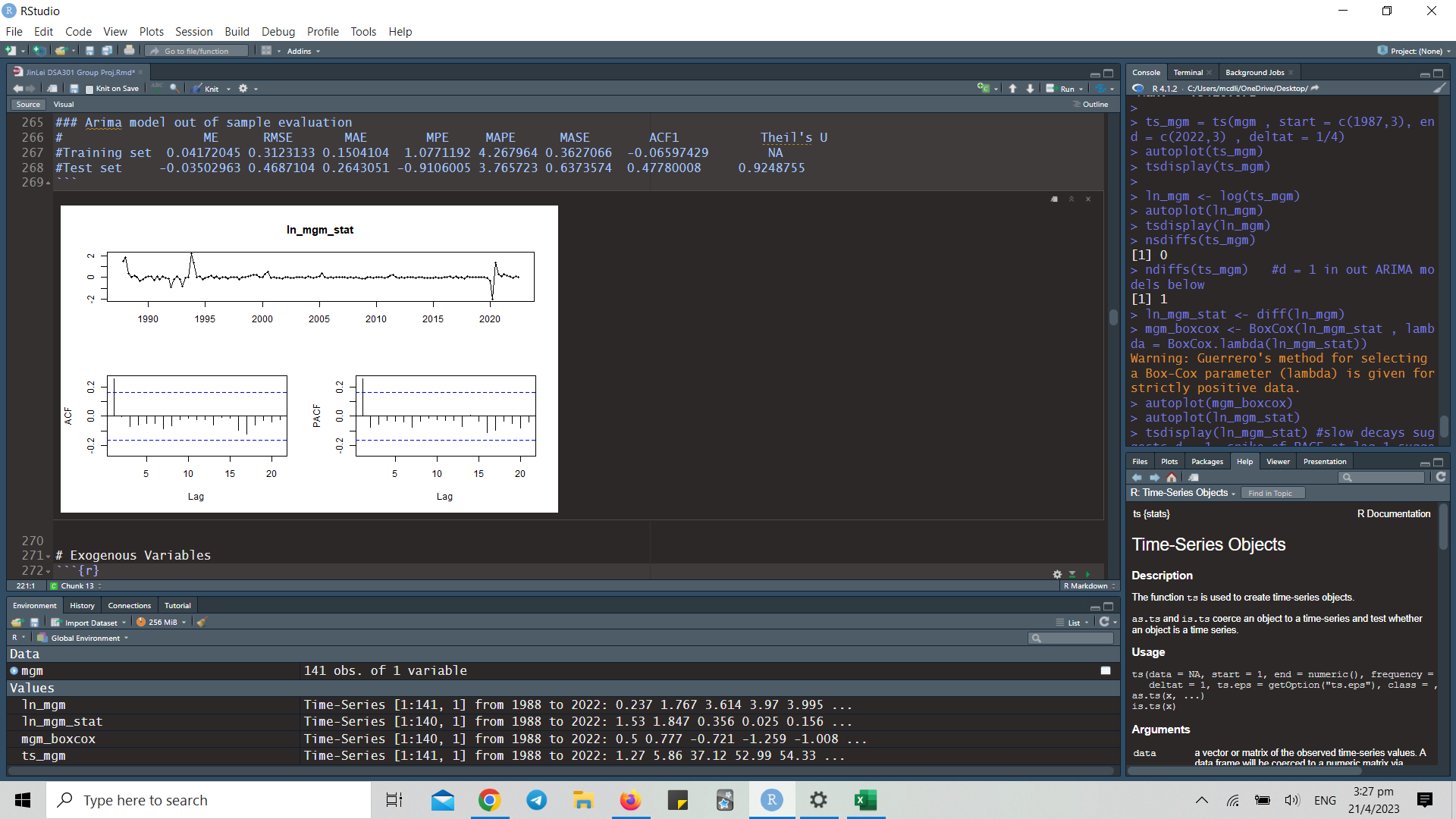
****

From the outputs, we see that Model 4 (STL Decomposition, naive + snaive) has the lowest MAPE and RMSE and we will select this as our benchmark model of choice.

# 

# **Arima Model**

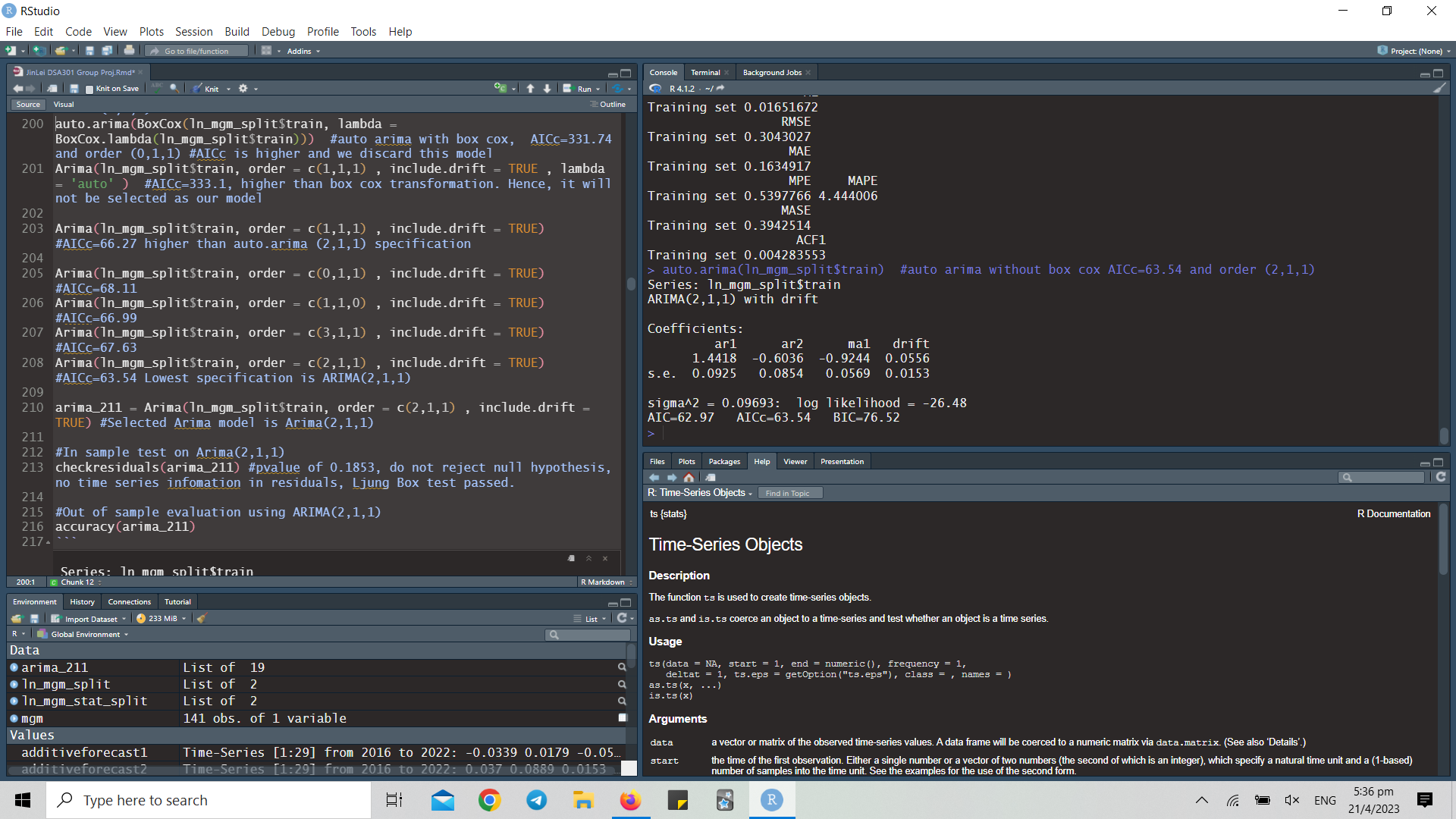
Before building the ARIMA model, we called tsdisplay(ln\_mgm\_stat) to look at the Autocorrelation function and Partial Autocorrelation function of ln\_mgm\_stat, to identify any potential spikes and lags that might allow us to determine the order of the model. The output for tsdisplay(ln\_mgm\_stat) is as follows:



From this output, we can see that there is a lag at time 1 for both the ACF and PACF, with no seasonalities. This indicates that a possible model specification would be an ARIMA(1,1,1) model, as one difference to the data has been taken, and there is a strong correlation between the current observation and the immediately preceding one. Using this as a baseline, we went on to build several ARIMA models on our insample data using both auto.arima() and ARIMA(). The model with the lowest AICc was chosen to be the best model, and we used it to test the model’s out of sample performance and compare its performance against other classes of models.

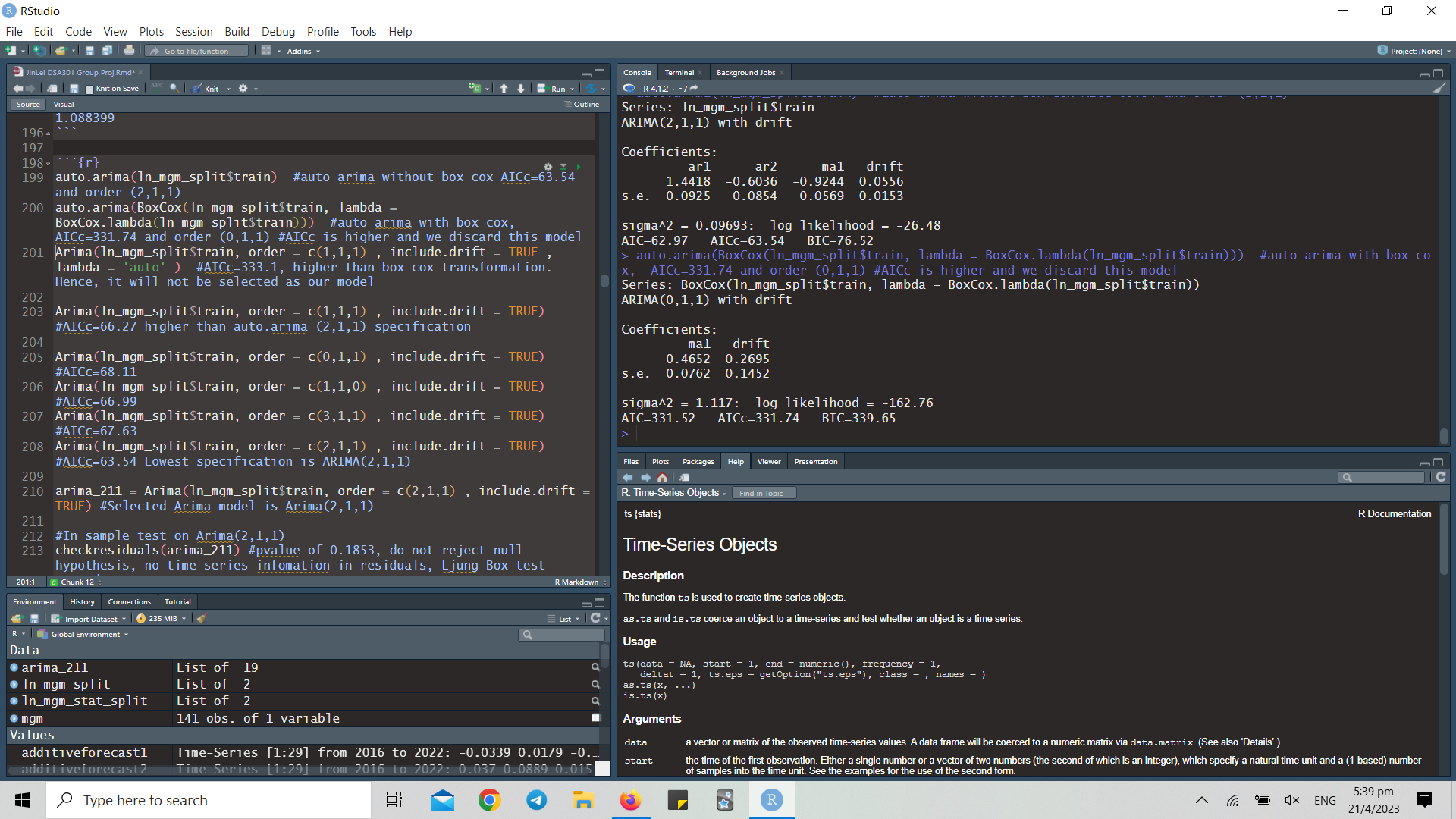
**The model specifications are as follows:**

*auto.arima(ln\_mgm\_split$train):*



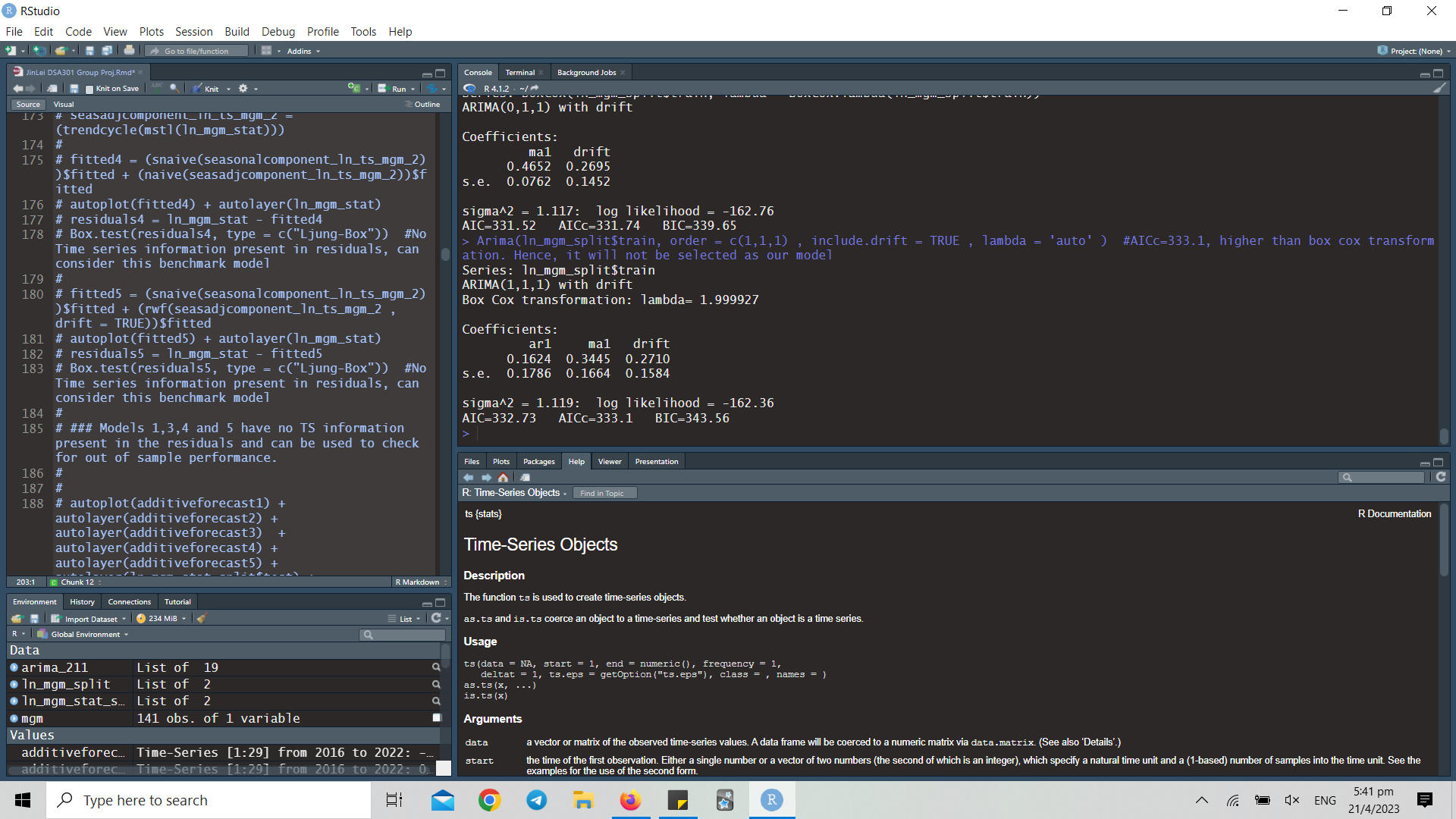
Returns ARIMA(2,1,1) model. AICc=63.54

*auto.arima(BoxCox(ln\_mgm\_split$train, lambda = BoxCox.lambda(ln\_mgm\_split$train))):*



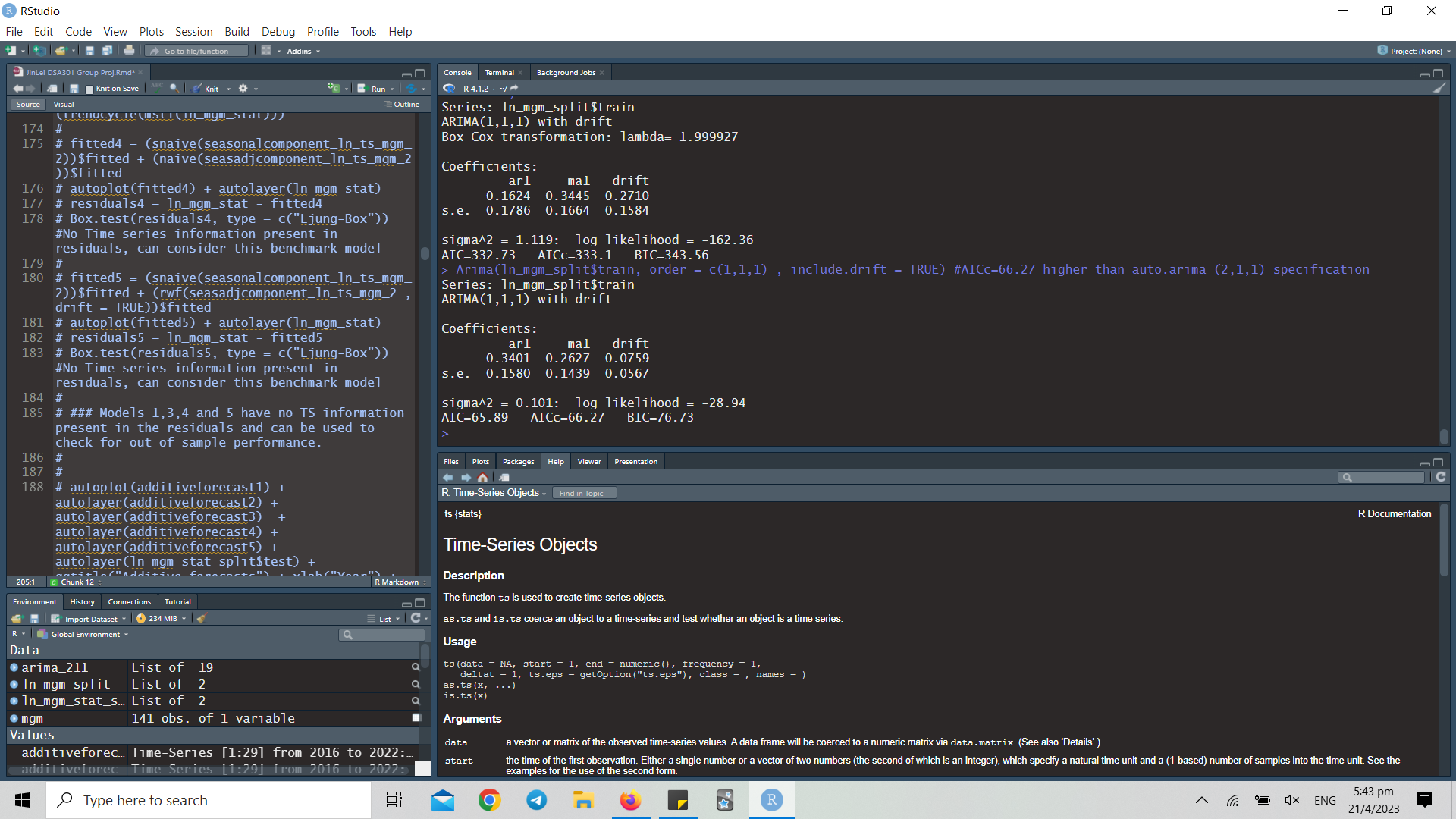
Returns ARIMA(0,1,1) model with drift term. AICc=331.74

*Arima(ln\_mgm\_split$train, order = c(1,1,1) , include.drift = TRUE , lambda = 'auto'):*



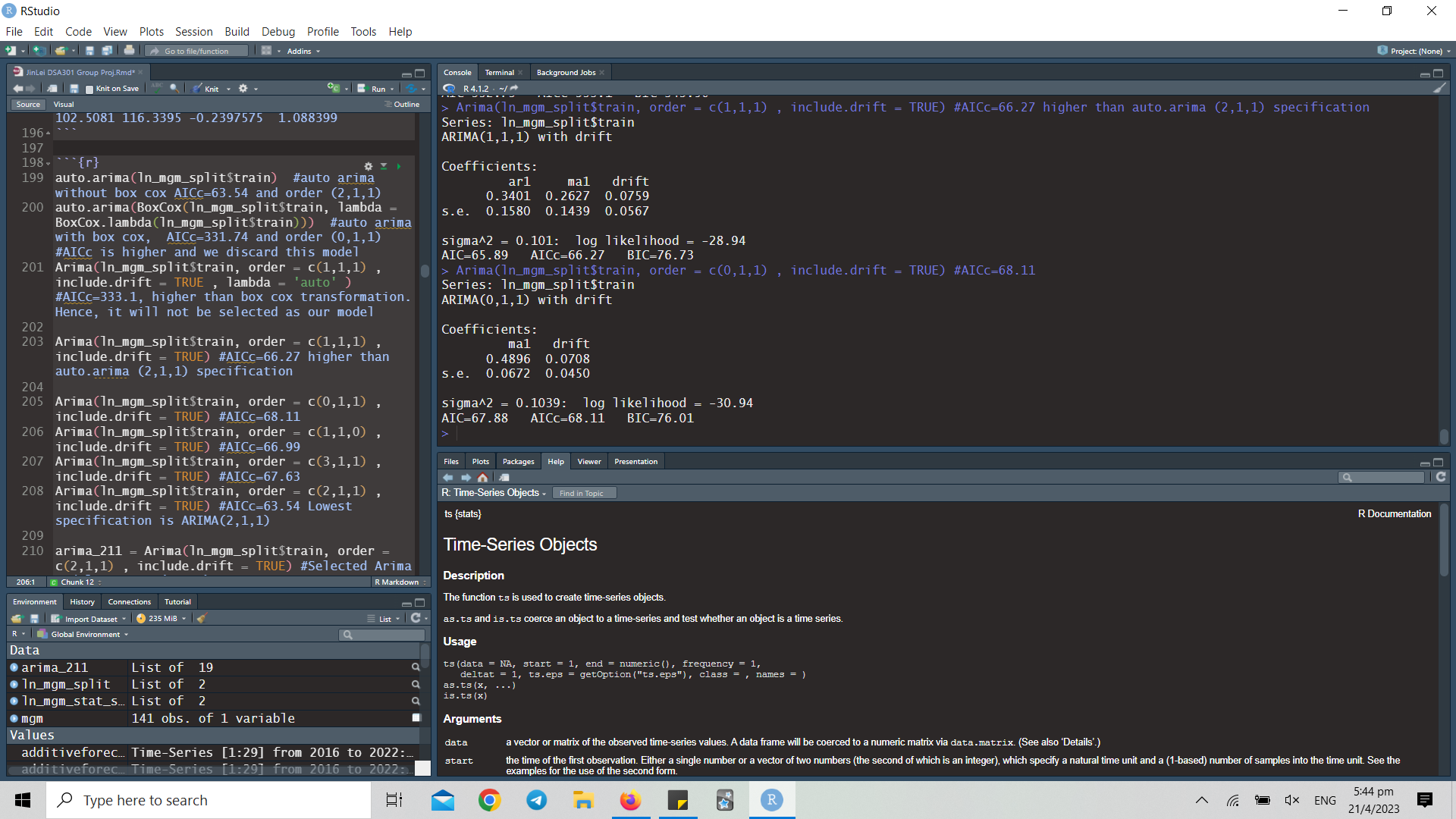
Returns ARIMA(1,1,1) model with automatic lambda for box-cox transform. AICc = 333.1

*Arima(ln\_mgm\_split$train, order = c(1,1,1) , include.drift = TRUE)*:



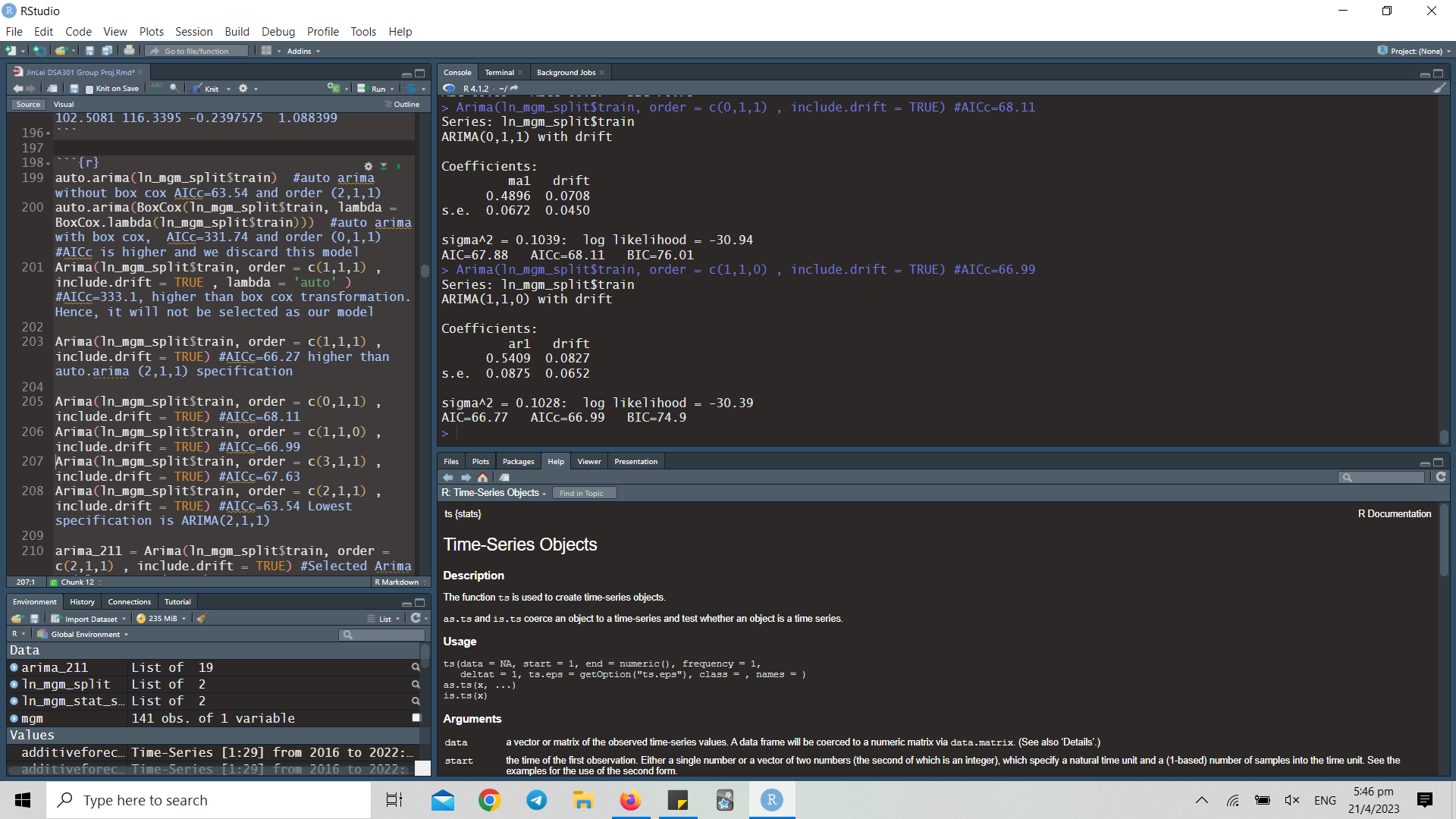
Returns ARIMA(1,1,1) model with drift term. AICc=66.27

*Arima(ln\_mgm\_split$train, order = c(0,1,1) , include.drift = TRUE)*:



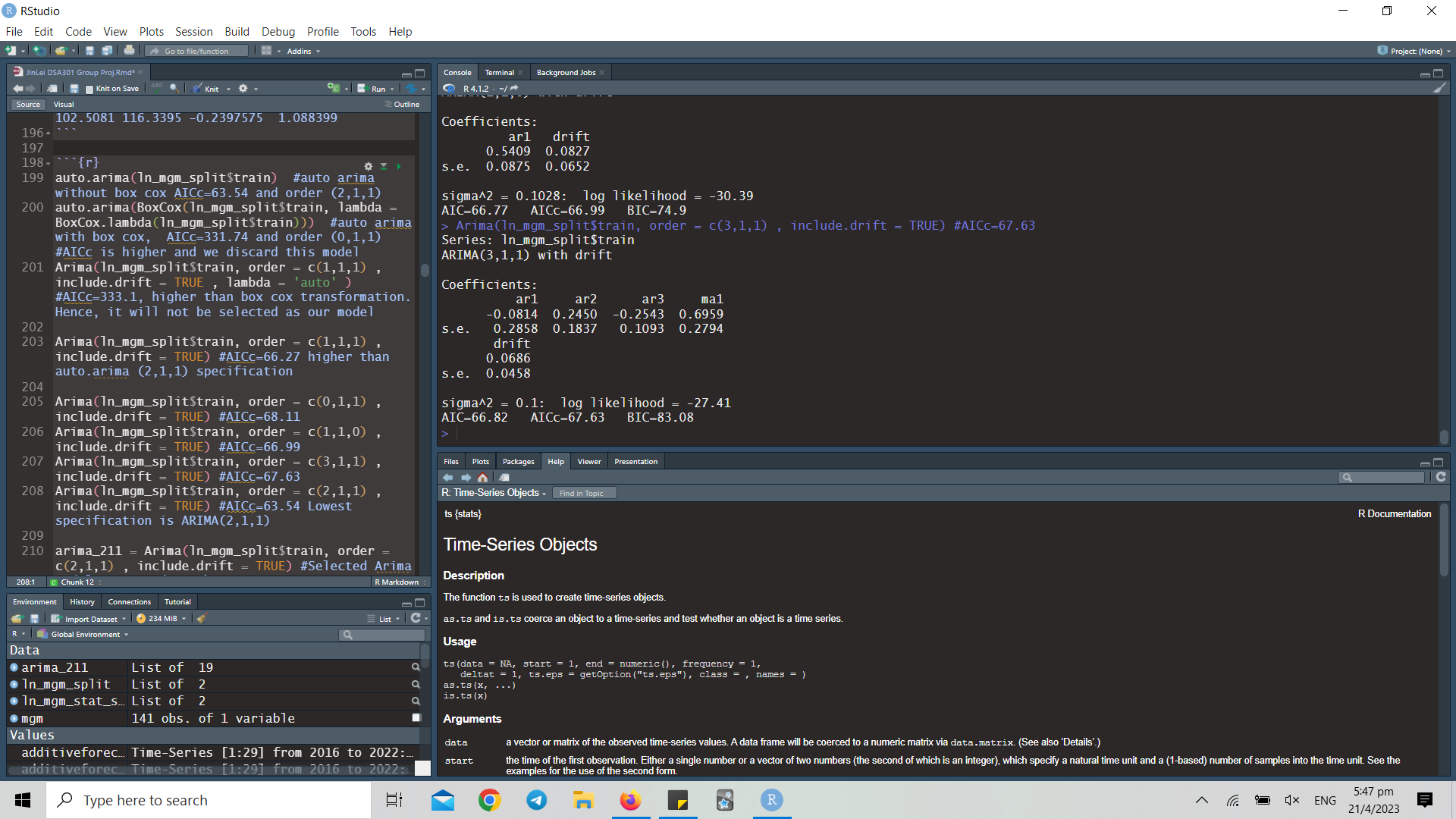
Returns ARIMA(0,1,1) with drift term. AICc=68.11

*Arima(ln\_mgm\_split$train, order = c(1,1,0) , include.drift = TRUE)*:



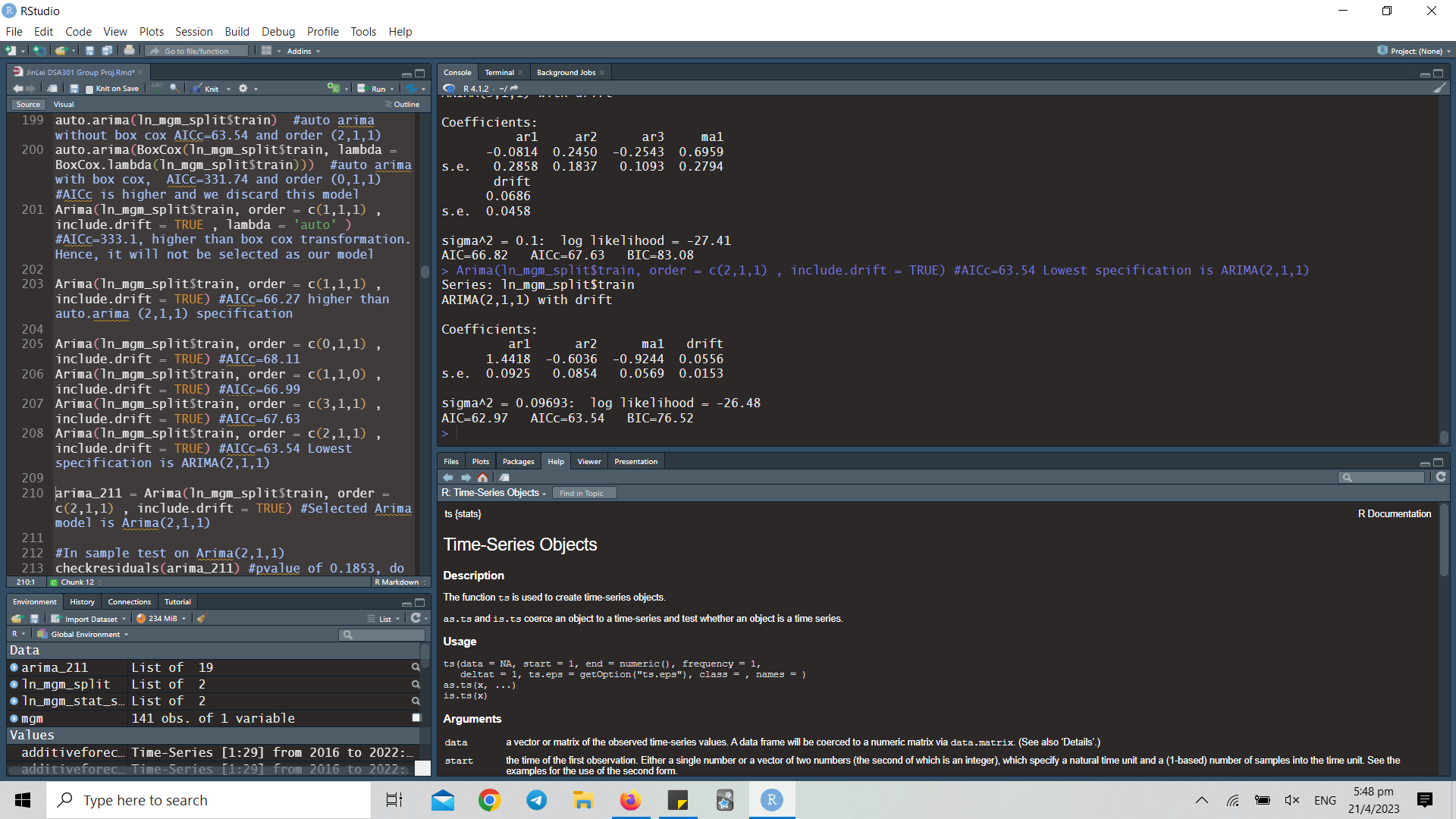
Returns ARIMA(1,1,0) model with drift term. AICc=66.99

*Arima(ln\_mgm\_split$train, order = c(3,1,1) , include.drift = TRUE)*:



Returns ARIMA(3,1,1) model with drift term. AICc=67.63

*Arima(ln\_mgm\_split$train, order = c(2,1,1) , include.drift = TRUE)*:



Returns ARIMA(2,1,1) model with drift term. AICc=63.54

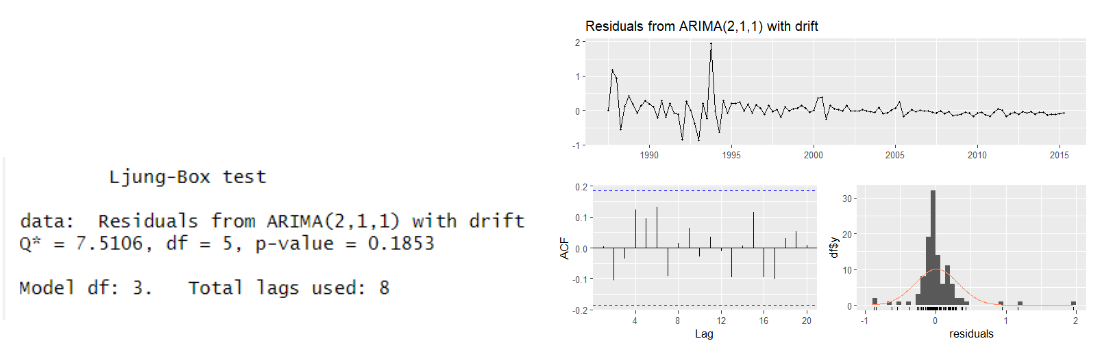
The ARIMA(2,1,1) model has the lowest AICc out of all the possible models, which is commensurate with the output of auto.arima(). Therefore, we choose to proceed with the ARIMA(2,1,1) model for the out of sample analysis. We used this code:

*arima\_211 = Arima(ln\_mgm\_split$train, order = c(2,1,1) , include.drift = TRUE)*

To assign the the model built on the training data to the variable arima\_211

## In sample performance and residual analysis

*checkresiduals(arima\_211)* returned this output:



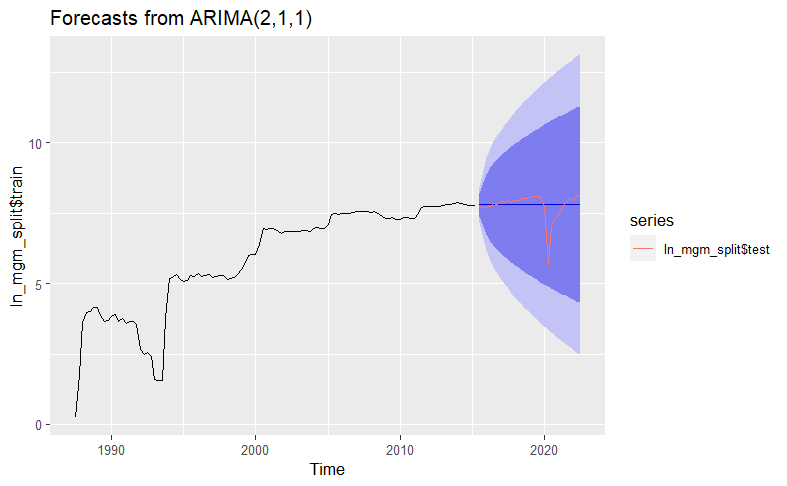
As the pvalue of the Ljung Box test is 0.1853, we conclude that there is insufficient evidence to reject the null hypothesis at the 10% significance level. Therefore, we can also conclude that there is no time series information in the residuals.

## 

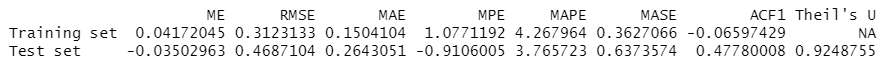
## Out of sample performance and residual analysis

*model1 = Arima(ln\_mgm\_split$train, order=c(2,1,1), seasonal=c(0,0,0))*

*autoplot(forecast(model1, h=outofsamplequarters), include= 112) + autolayer(ln\_mgm\_split$test)*



*accuracy(x = ln\_mgm\_split$test, forecast(model1, h= outofsamplequarters), include= 112)*

**

# **ARIMA-X Model**

## Identify Exogenous Variables

We hypothesize that MGM’s revenue may be affected by a few exogenous variables:

1. US Federal Reserve Rates

MGM's revenue can be indirectly influenced by the federal funds rate through its impact on the broader economy and consumer behavior. If the federal funds rate is decreased, it can stimulate borrowing and investment, thereby increasing economic activity and consumer spending. As a result, companies like MGM that depend on consumer discretionary spending may experience higher revenue.

Conversely, if the federal funds rate is increased, it can lead to reduced economic activity and consumer spending, which can potentially result in lower revenue for companies such as MGM.

1. S&P 500

We have included the S&P 500 index in our analysis since MGM is one of its constituents, and the index's performance can influence MGM Resorts International's revenue in various ways.

Primarily, the overall health of the economy, as indicated by the S&P 500 index, can impact consumer spending and, consequently, affect MGM's revenue. For example, if the S&P 500 index is performing well, consumers may feel more confident about the economy and, therefore, be more inclined to spend on leisure activities like travel, which could boost MGM's revenue.   
  
Conversely, if the S&P 500 index is underperforming and consumers feel less optimistic about the economy, they may spend less on discretionary activities, which could negatively impact MGM's revenue.

Secondly, the performance of the S&P 500 index can also influence the availability and cost of capital for MGM Resorts International. If the S&P 500 index is performing well, and investor sentiment is positive, MGM may be able to access capital at more favorable rates, thereby supporting its growth and revenue. On the other hand, if the S&P 500 index is performing poorly, and investor sentiment is negative, it may be more challenging and expensive for MGM to access capital, thereby potentially limiting its revenue growth.

1. US GDP Per Capita

The US GDP Per Capita can impact MGM's revenue in various ways due to its influence on the economy and consumer behavior, particularly in entertainment, travel, and leisure activities.

When the US GDP Per Capita is growing, consumers tend to have more disposable income to spend on entertainment, such as visiting MGM's casinos, shows, and other attractions. This increase in consumer spending can lead to a rise in MGM's revenue.

However, during an economic downturn or recession, consumers may become more cautious with their spending, leading to a decrease in the amount they spend on leisure activities like visiting casinos. This, in turn, can negatively affect MGM's revenue.

Furthermore, changes in the US GDP Per Capita can also impact MGM's business operations, including the availability of credit and financing, consumer confidence levels, and the overall competitive landscape within the industry.

## 

## 

## Granger Causality Test

1. US Feds

*grangertest(ts\_stat\_USFeds, ln\_mgm\_stat, order = 1) # insignificant*

*grangertest(ts\_stat\_USFeds, ln\_mgm\_stat, order = 2) # insignificant*

*grangertest(ts\_stat\_USFeds, ln\_mgm\_stat, order = 3) # insignificant*

*grangertest(ts\_stat\_USFeds, ln\_mgm\_stat, order = 4) # insignificant*

The granger causality for USFeds Rate on MGM’s revenue is insignificant, this runs counter to our initial intuition on their statistical relationship. Hence, we conclude that USFeds rates do not predict or have significant impact on MGM’s Revenue.

*grangertest(ln\_mgm\_stat, ts\_stat\_USFeds, order = 1) # insignificant*

*grangertest(ln\_mgm\_stat, ts\_stat\_USFeds, order = 2) # insignificant*

*grangertest(ln\_mgm\_stat, ts\_stat\_USFeds, order = 3) # insignificant*

*grangertest(ln\_mgm\_stat, ts\_stat\_USFeds, order = 4) # insignificant*

The granger causality for MGM’s revenue on USFeds Rate is insignificant, this runs consistent with our initial intuition that MGM’s Revenue should have no causal impact to USFeds Rate.

1. SNP

*grangertest(ts\_stat\_SNP, ln\_mgm\_stat, order = 1) # insignificant*

*grangertest(ts\_stat\_SNP, ln\_mgm\_stat, order = 2) # insignificant*

*grangertest(ts\_stat\_SNP, ln\_mgm\_stat, order = 3) # insignificant*

*grangertest(ts\_stat\_SNP, ln\_mgm\_stat, order = 4) # insignificant*

The granger causality for S&P 500 on MGM’s revenue is insignificant, this runs counter to our initial intuition on their statistical relationship. Hence, we conclude that S&P 500 rates do not predict or have significant impact on MGM’s Revenue.

*grangertest(ln\_mgm\_stat, ts\_stat\_SNP, order = 1) # Significant*

*grangertest(ln\_mgm\_stat, ts\_stat\_SNP, order = 2) # Significant*

*grangertest(ln\_mgm\_stat, ts\_stat\_SNP, order = 3) # Significant*

*grangertest(ln\_mgm\_stat, ts\_stat\_SNP, order = 4) # Significant*

Inconsistent with our economic intuition, the granger causality test for MGM's revenue on S&P 500 is significant , indicating that MGM's revenue has a causal impact on S&P 500 performance. This violates the basic assumption of the regression model that S&P 500 is an exogenous variable. Hence, the S&P 500 variable should be dropped.

1. US GDP

*grangertest(ts\_stat\_usgdppc, ln\_mgm\_stat, order = 1) # insignificant*

*grangertest(ts\_stat\_usgdppc, ln\_mgm\_stat, order = 2) # insignificant*

*grangertest(ts\_stat\_usgdppc, ln\_mgm\_stat, order = 3) # insignificant*

*grangertest(ts\_stat\_usgdppc, ln\_mgm\_stat, order = 4) # insignificant*

The granger causality for US GDP Per Capita on MGM’s revenue is insignificant, this runs counter to our initial intuition on their statistical relationship. Hence, we conclude that US GDP Per Capita does not predict or have significant impact on MGM’s Revenue.  
.

*grangertest(ln\_mgm\_stat, ts\_stat\_usgdppc, order = 1) # insignificant*

*grangertest(ln\_mgm\_stat, ts\_stat\_usgdppc, order = 2) # Significant*

*grangertest(ln\_mgm\_stat, ts\_stat\_usgdppc, order = 3) # Significant*

*grangertest(ln\_mgm\_stat, ts\_stat\_usgdppc, order = 4) # Significant*

Inconsistent with our economic intuition, MGM's Revenue granger causes US GDP Per Capita granger. We can conclude that MGM's changes in revenue can be used to predict or measure significant impact on US GDP Per Capita. This violates the basic assumption of the regression model that US GDP Per Capita is an exogenous variable. Hence, the US GDP Per Capita variable should be dropped.

Based on our Granger Causality analysis, MGM’s revenue granger causes the exogenous variables, but not the other way around. This discovery contradicts the assumption of exogeneity, which could potentially result in bias in both the estimation of coefficients and the predictive performance of the model.

In contrast, the US Federal Reserve rates did not pass the Granger causality test on our dependent variable. Despite this, we will proceed with our ARIMA-X analysis to demonstrate our argument and observe any potential drawbacks of incorporating these variables.

# 

## 

## Data Transformation of Exogenous Variables

Differencing

1. US Feds

*nsdiffs(ts\_USFeds)*

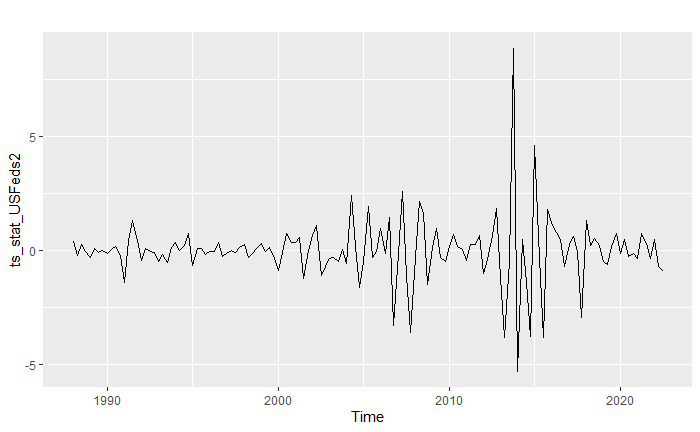
*ndiffs(ts\_USFeds, test = 'kpss', alpha = 0.05)*

*ts\_stat\_USFeds <- diff(ts\_USFeds) #1 diff*

*ts\_stat\_USFeds2 <- diff(ts\_stat\_USFeds) # Apply same level of differencing as S&P500*

*ndiffs(ts\_stat\_USFeds)*

*autoplot(ts\_stat\_USFeds2)*

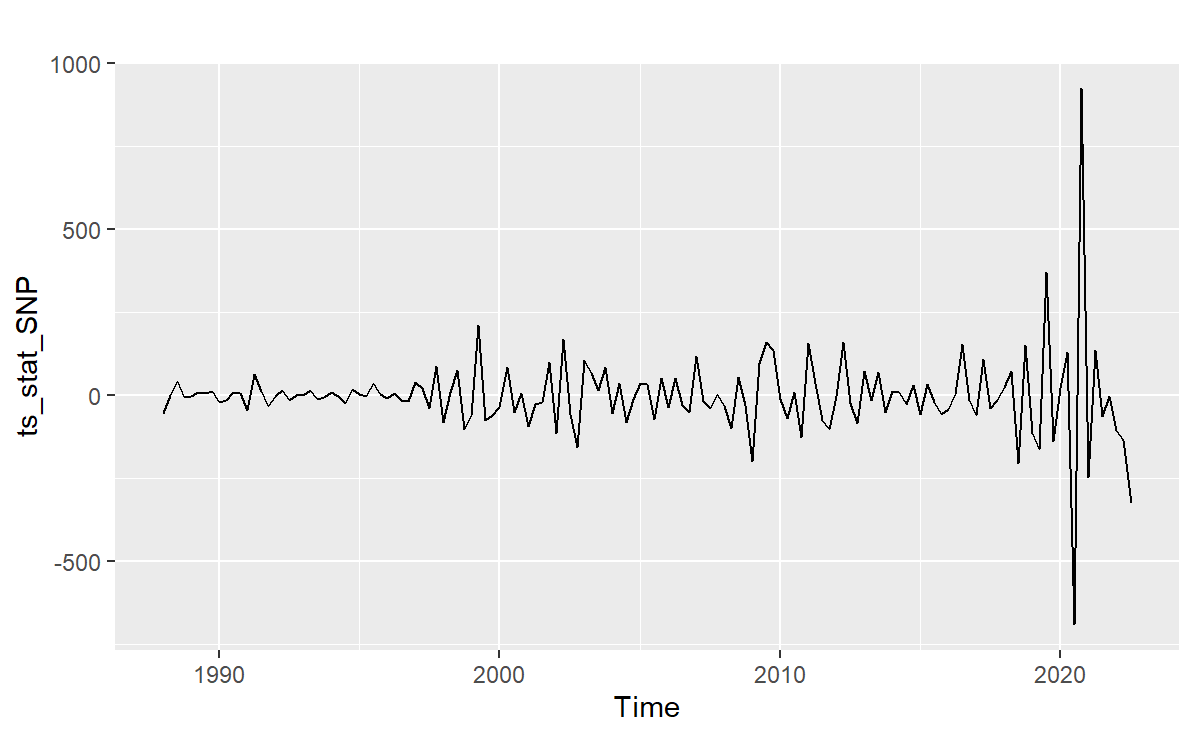


1. SNP

*nsdiffs(ts\_SNP)*

*ndiffs(ts\_SNP)*

*ts\_stat\_SNP <- diff(diff(ts\_SNP)) #2 diff*



1. US GDP

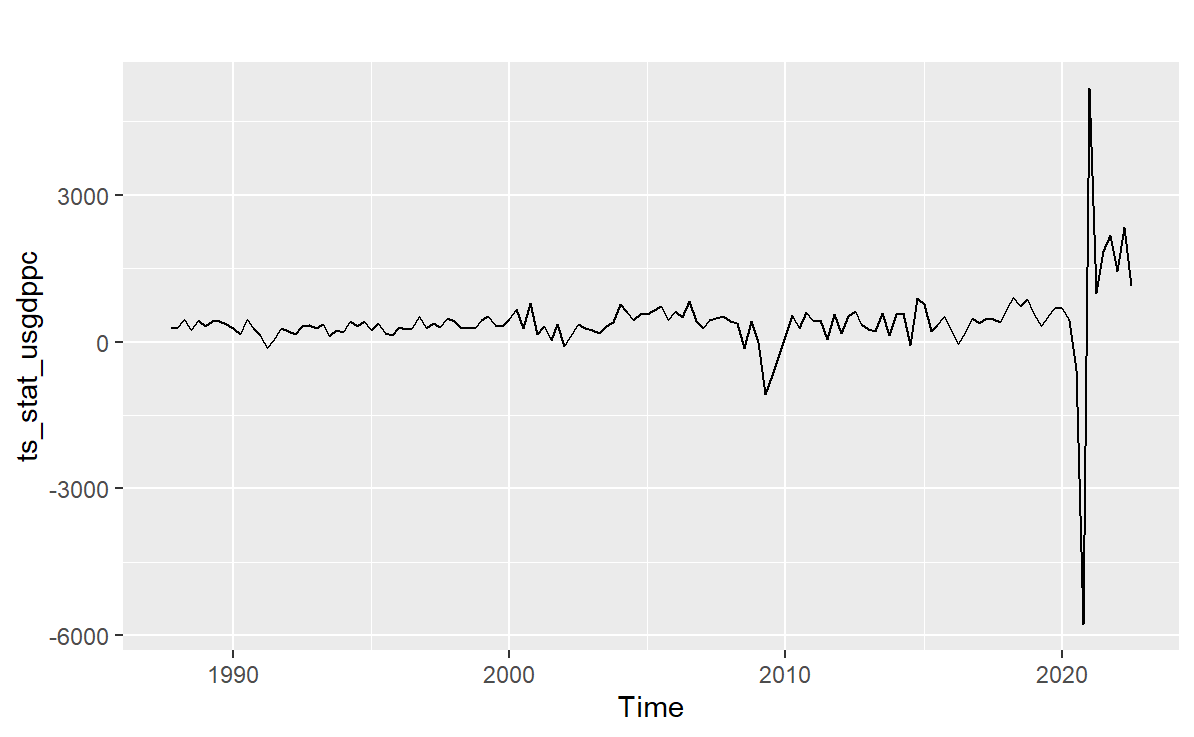
*nsdiffs(ts\_usgdppc)*

*ndiffs(ts\_usgdppc)*

*ts\_stat\_usgdppc <- diff(ts\_usgdppc) #1 diff*

*ts\_stat\_usgdppc2 <- diff(ts\_stat\_usgdppc) # Apply same level of differencing as S&P500*

*autoplot(ts\_stat\_usgdppc2)*



Training and testing sets

*USFeds\_stat\_split = ts\_split(ts\_stat\_USFeds, sample.out = outofsamplequarters)*

*SNP\_stat\_split = ts\_split(ts\_stat\_SNP, sample.out = outofsamplequarters)*

*USGDPPC\_stat\_split = ts\_split(ts\_stat\_usgdppc, sample.out = outofsamplequarters)*

## Building of ARIMA-X Model

Splitting both endogenous and exogenous data into training and test set

*ln\_mgm2 <- ts(ln\_mgm[3:141,], start = c(1988, 1), end = c(2022, 3), deltat = 1/4)*

*ln\_mgm2\_split <- ts\_split(ln\_mgm2, sample.out = outofsamplequarters)*

*arima\_gdp\_split <- ts\_split(ts\_stat\_usgdppc2, sample.out = outofsamplequarters)*

*arima\_Feds\_split <- ts\_split(ts\_stat\_USFeds2, sample.out = outofsamplequarters)*

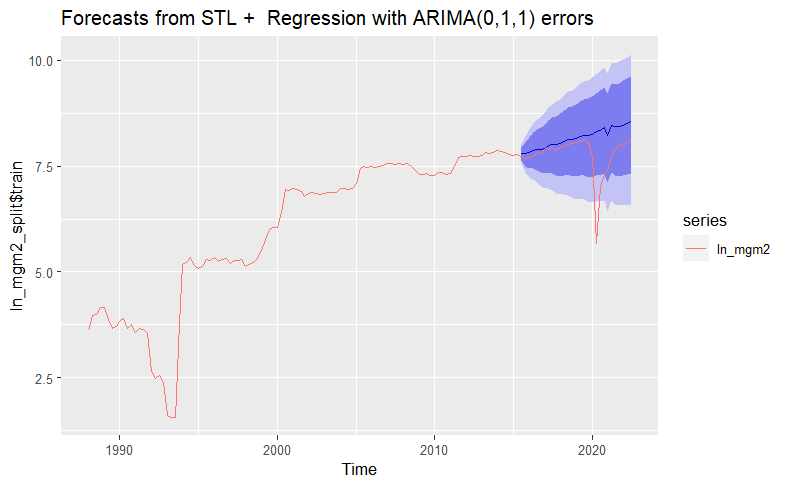
*arima\_SNP\_split <- ts\_split(ts\_stat\_SNP, sample.out = outofsamplequarters)*

### ARIMA-X model with all exogenous variable (Inconsistent with our intuition)

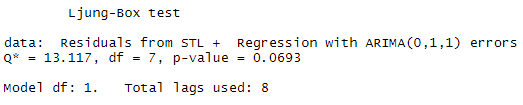
1. ARIMA-X Model

*arimax3 <- stlf(ln\_mgm2\_split$train, h = outofsamplequarters, method = 'arima', lambda='auto' , xreg = cbind(arima\_Feds\_split$train, arima\_gdp\_split$train, arima\_SNP\_split$train), newxreg = cbind(arima\_Feds\_split$test, arima\_gdp\_split$test, arima\_SNP\_split$test))*

*autoplot(arimax3) + autolayer(ln\_mgm2)*



1. Check Residuals



With a p-value of 0.07, we fail to reject the null hypothesis that there is no time series information in the residual at the 10% significance level, which deems it a valid model.

1. Out-of-sample Evaluation

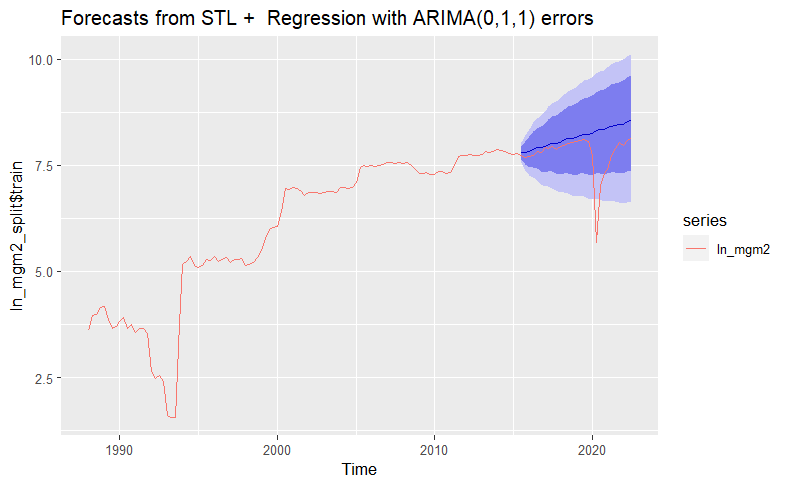
*accuracy(arimax3, ln\_mgm2\_split$test)*  
  


### ARIMA-X model with only US Federal Reserve Rates (Ambiguous Variable)

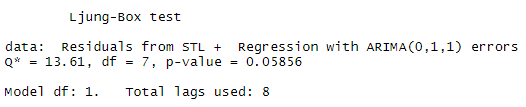
1. ARIMA-X Model

*arimax1 <- stlf(ln\_mgm2\_split$train, h = outofsamplequarters, method = 'arima', lambda='auto' , xreg = cbind(arima\_Feds\_split$train), newxreg = arima\_Feds\_split$test)*

*autoplot(arimax1) + autolayer(ln\_mgm2)*



1. Check Residuals

*checkresiduals(arimax1)*

1. Out-of-sample Evaluation

*accuracy(arimax1,ln\_mgm2\_split$test)*  
  


Conclusion on ARIMA-X

Through our exploratory data analysis of granger causality of exogenous variables on MGM’s revenue, we have concluded that these variables violate the regression assumption. However, to solidify our argument we have also built an ARIMA-X model to compare with other time series techniques as mentioned in the objective.

# 

# **Prophet Model**

The Prophet model is a relatively new innovation by Meta and was introduced in the year 2018 for forecasting daily data with weekly and yearly seasonality. Although our data does not capture daily data, we decided to be extensive and cover as many models as we are taught in class. The Prophet model has four components being, a piecewise linear trend, seasonal model component, model of holiday effects and a white noise error term.

The prophet function expects two columns in the input Dataframe being ds and y. ds contains date values white y contains numeric values. So, before we are able to create the model, we had to extract dates when the MGM revenues were collected and MGM revenues and place them in a Dataframe.

*mgm2 <- read\_excel("Term Project.xls", sheet = 'MGM Quarterly Revenue')*

*mgmdates <- mgm2[3:143,2]*

*mgmrev <- mgm2[,14] %>%*

*filter(revtq>0)*

*PreProphet <- cbind(mgmdates, mgmrev)*

Next, we had to change the dates into the proper year-month-date format, rename the columns into ds and y in order to match what the prophet function is expecting. Additionally we log transformed MGM revenue.

*PreProphet <- mutate(PreProphet, ds = ymd(PreProphet[,1]))*

*PreProphet <- mutate(PreProphet, y = log(revtq))*

*PreProphet <- dplyr::select(PreProphet, ds, y)*

Splitting the Dataframe into training and testing sets using 80% training and 20% test set ratio.

*train <- PreProphet[1:110,]*

*test <- PreProphet[111:141,]*

Created the Prophet model using R function and made predictions with the model

*ProphetModel <- prophet(train, seasonality.mode = "additive")*

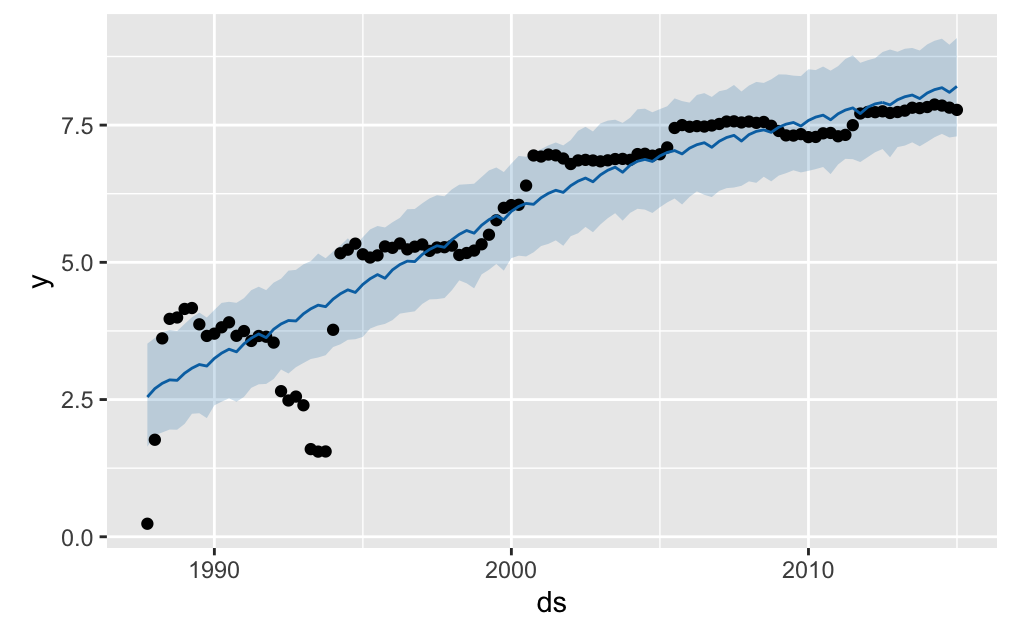
*future = make\_future\_dataframe(ProphetModel,periods = 29,freq = 'quarter')*

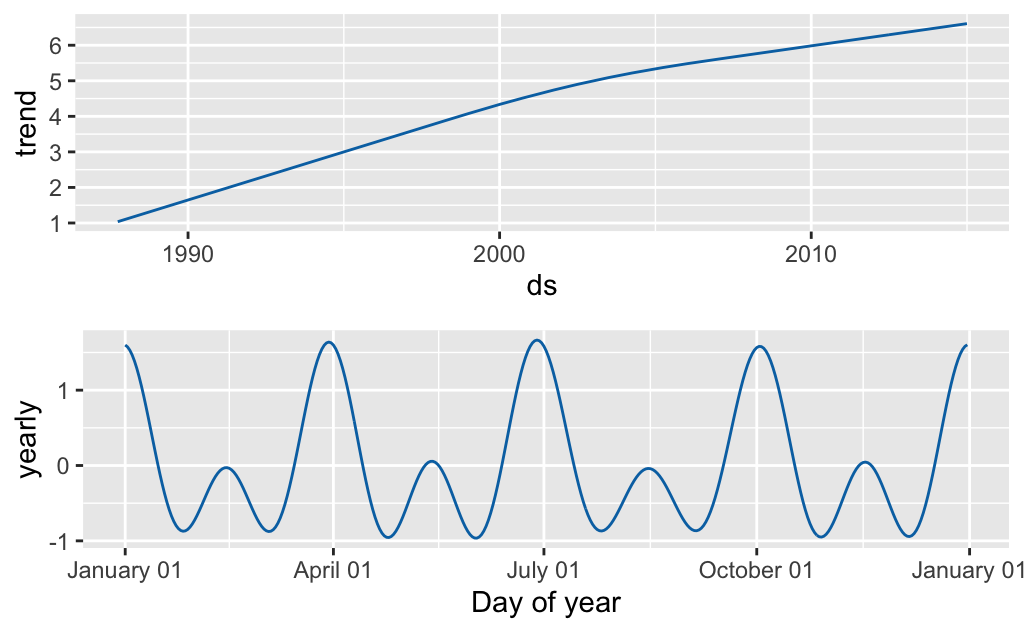
*Prediction <- predict(ProphetModel,period = future)*

Generating plots

*plot(ProphetModel, Prediction)*

*prophet\_plot\_components(ProphetModel, Prediction)*

**

**

Manual calculation of out of sample performance in order to compare the prophet model with other models that we have created.

*forecast\_metric\_data = Prediction %>% as\_tibble()*

*MAPE = mean(abs((test$y - forecast\_metric\_data$yhat)/test$y)\*100)*

*MAPE*

**

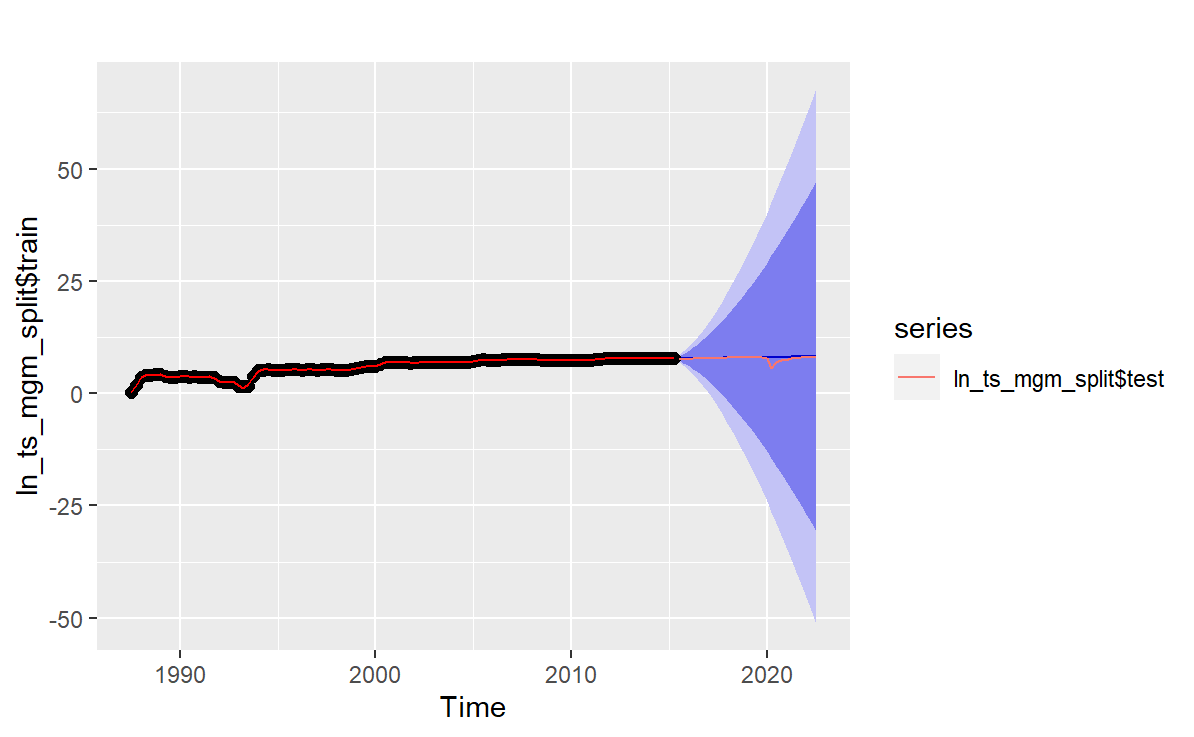
​​

# 

# **Cubic Spline**

Using R’s spline function, we built a cubic spline model on the ln mgm revenue data. Cubic splines are flexible and can capture nonlinear relationships between the variables, and they can handle missing data and irregularly spaced observations. However, they can be computationally intensive and require careful selection of the knot locations to avoid overfitting. We left the selection to knots to the built in function in r.

*autoplot(splinef(ln\_ts\_mgm\_split$train , h = outofsamplequarters)) + autolayer(ln\_ts\_mgm\_split$test)*

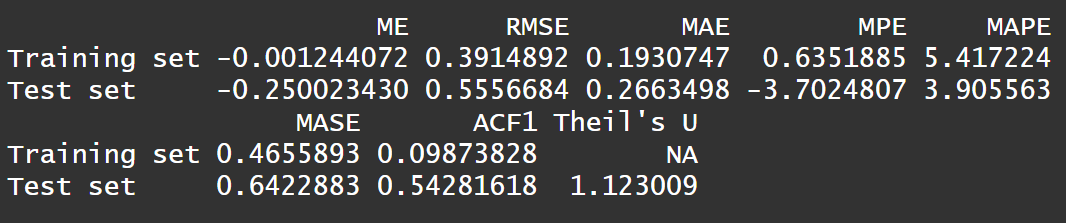


*cubic\_forecast = splinef(ln\_ts\_mgm\_split$train , h = outofsamplequarters)*

*accuracy\_spline = accuracy(cubic\_forecast ,ln\_ts\_mgm\_split$test )*

*accuracy\_spline*

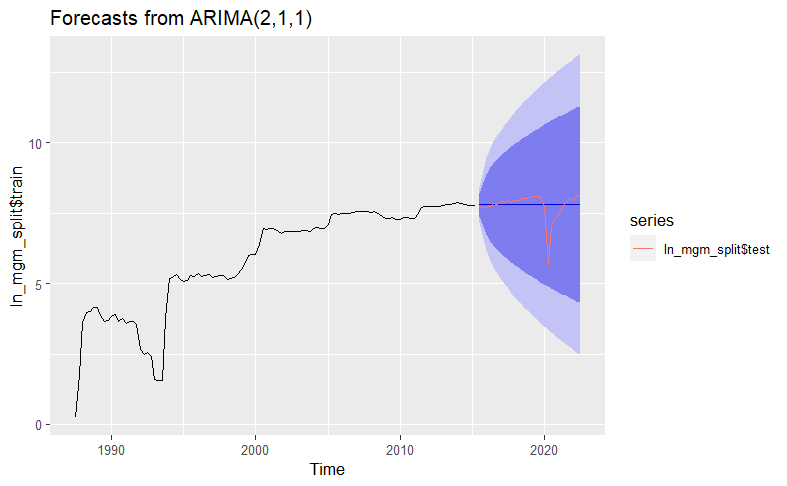
Next we calculate the out-of-sample performance of the cubic spline model and obtain the test statistics as shown below.



# **Conclusion**

To summarize, we have evaluated different models to forecast MGM revenue, including benchmark models, ARIMA models, ARIMA-X models, Prophet models, and Cubic Spline models. Out of all the models, we found that the ARIMA(2,1,1) model provides the best forecasting performance with the lowest AICc and MAPE. The benchmark models and ARIMA-X models do not capture the time series information and the Granger causality test shows no relationship between the exogenous variables and the MGM revenue. The Prophet model is not suitable for our data due to little seasonality in quarters. Therefore, we will be using the ARIMA(2,1,1) model to forecast the log revenue of MGM.

Autoplot of Forecasts



Prediction Accuracy

|  |  |
| --- | --- |
| **Model** | **MAPE** |
| Benchmark | 116.3395 |
| **ARIMA** | **3.765723** |
| ARIMA-X | 5.604528 |
| Prophet | 25.2773 |
| Cubic Spline | 3.905563 |