Time Series Forecasting on Monthly Total Production of All Industries Excluding Construction from 1961 to 2022 in Canada

**Table of Contents**

[1. Data 3](#_Toc143196113)

[2. Visualization 4](#_Toc143196114)

[3. Data Transformation 10](#_Toc143196115)

[3.1 Box-Cox Transformation 10](#_Toc143196116)

[3.2 Decompositions 12](#_Toc143196118)

[4. Forecasting and Analysis 15](#_Toc143196119)

[4.1 ETS 15](#_Toc143196120)

[4.2 Seasonal ARIMA 17](#_Toc143196121)

[4.3 Other Models 20](#_Toc143196122)

[4.4 Model Evaluation 21](#_Toc143196123)

[4.5 Forecasting 23](#_Toc143196124)

[4.6 Summary 24](#_Toc143196125)

[5. References 25](#_Toc143196126)

[6. Appendix 26](#_Toc143196127)

## Data

Using data from the Organization for Economic Co-operation and Development (OECD), this project examines the total industry production, excluding construction in Canada. The OECD monthly data, retrieved from FRED - Federal Reserve Bank of St. Louis, are not seasonally adjusted (FRED, 2023). In particular, the time series data include monthly total production of all the industries excluding construction in Canada from January 1961 to May 2023 (749 observations). We removed the last five observations in 2023 to maintain the full seasonal effect.

There are two columns in the dataset: DATE and CANPRINTO01MLM. The DATE column contains dates in “YYYY-MM-DD” format, which R cannot identify. We transformed the format to “YYYY MMM” so that R can recognize them as the date data type. The CANPRINTO01MLM column contains the monthly total production of billions of Canadian dollars. We scaled the column by dividing the figures by a billion. We also changed the column names to Date and Production.

First, converting the column with Date observations from text to a monthly time object with the function yearmonth was necessary to start the analysis. After that, using the tsibble function, the data frame was converted into a table designed for time series analysis (Hyndman & Athanasopoulos, 2021).

The project aims to analyze total industry production in Canada using standard statistical forecasting techniques, such as ETS and ARIMA models, and compare different models to choose the best-performing models to forecast future data. Regarding the analysis, the project estimates how the production in the Canadian industry will continue in the near future, forecasting the monthly production.

Particularly, this project aims to answer the following questions:

* how the total industry production has changed over time;
* how the total industry production will continue into the future;
* which is the best model to produce forecasts for the chosen time series data.

Production is one of the key economic indicators of a country’s growth. Increasing production tends to impact consumption and economic development positively and can also affect the employment rate. With the financial crisis in 2008 and, even more, with the COVID-19 pandemic, the media pay more attention to economic indicators than ever.

## Visualization

To identify the patterns in the data, we created some time plots. As shown in Figure 1, the total production exhibits a cyclical pattern, positive trend, and clear seasonality. In general, the time series data has an increasing trend over the past six decades. There is also a strong seasonal pattern, decreasing in January, July and December, and a cyclical pattern due to economic conditions in the first half of the 1980s, 1990s, and after the 2008 crisis.

A graph showing a line graph

Description automatically generated

Figure 1: Time Plot for Total Industry Monthly Production (from 1961 Jan to 2022 Dec)

However, the data patterns seem to change before 1975 and after 2008. The data variance gradually increased before 1975 and became more constant afterwards. This growing change in data variance will lead to inaccurate forecasts. The changing economy and advancing technology also make the data before 1975 unreliable for making forecasts. As a result, we disposed of the data from the first two decades and started our data from 1983, four decades away from the last observation.

A graph showing a line of growth

Description automatically generated with medium confidence

Figure 2: Time Plot for Total Industry Monthly Production (from 1983 Jan to 2022 Dec)

We then plotted the 20-year data before and after 2003 to examine the pattern differences in the twentieth century. We decided to use 2003 as the split point because it is a year near 2008 and is two decades away from our last observation. Figures 2 and 3 show the time plots for the 20-year data before and after 2003. We can see apparent differences in the patterns between Figures 3 and 4. The data has fewer variations in some months after 2008, possibly due to the economic recession. There was also a big lap in 2020 when the Covid-19 pandemic happened.

A graph showing a line

Description automatically generated

Figure 3: Time Plot for Total Industry Monthly Production (from 1983 Jan to 2003 Dec)

A graph showing a line

Description automatically generated

Figure 4: Time Plot for Total Industry Monthly Production (from 2003 Jan to 2022 Dec)

We also looked into the monthly pattern between years for the data before and after 2008 to identify the years in which the pattern changes. As shown below, the differences in monthly patterns become more transparent. The monthly data seems to be flat, and the annual growth is more considerable in Figure 5, while the monthly data seems to fluctuate more, and the yearly growth became smaller after 2008 in Figure 6. Both figures show a slight dip every July and December, the time for summer vacation and the holiday season, respectively. And invariably, the highest production each year is November, just before the holiday season. Figure 6 shows some additional insights about the data. The 2009 curve is well at the bottom due to the economic crisis, and the 2020 curve has a sharp dip from April due to COVID-19.

A graph of different colored lines

Description automatically generated

Figure 5: Seasonal Plot for Total Industry Monthly Production (from 1983 Jan to 2008 Dec)

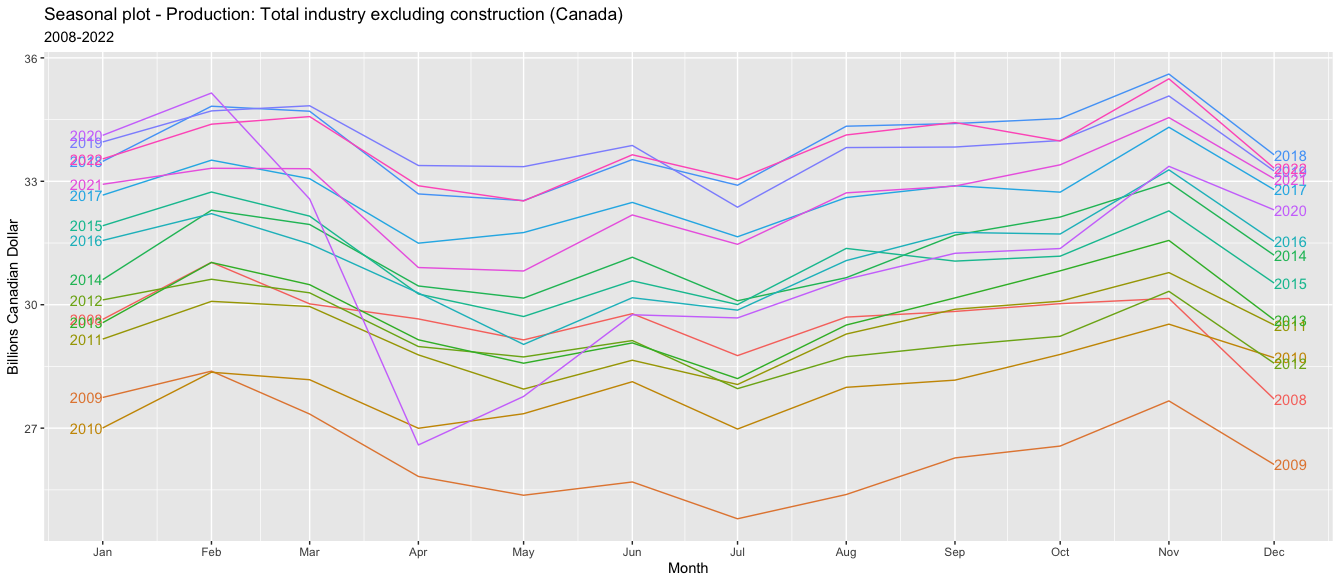


Figure 6: Seasonal Plot for Total Industry Monthly Production (from 2008 Jan to 2022 Dec)

Because there is a difference in data pattern, we decided to predict the future using all the data (1983-2022) and the data after 2003 and see if the predictions are also different.

In terms of overall monthly patterns, we plotted the monthly data over the years to emphasize seasonal patterns. The blue horizontal bar suggests the mean value for each month. We can see from Figure 7 that, in general, November will have the highest production and July the lowest.

A graph of a graph of a graph

Description automatically generated with medium confidence

Figure 7: Seasonal Subseries for Total Industry Monthly Production (from 1983 Jan to 2003 Dec)

Figure 8 shows the impacts of the 2008 financial crisis and COVID-19 in 2020, indicated by the downward spikes.

A graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of

Description automatically generated

Figure 8: Seasonal Subseries for Total Industry Monthly Production (from 2003 Jan to 2022 Dec)

Autocorrelation is always a problem in time series data. We can use the ACF plot to identify the severity of autocorrelation. Both figures show strong and inverse autocorrelations, reflecting that the total production experienced cyclical upwards and downwards over the four decades. The spikes in every 12 lags suggest evident seasonality every 12 months.

A graph of a graph

Description automatically generated

Figure 9: ACF Plot for Total Industry Monthly Production (from 1983 Jan to 2003 Dec)

A graph with lines and numbers

Description automatically generated

Figure 10: ACF Plot for Total Industry Monthly Production (from 2003 Jan to 2022 Dec)

## Data Transformation

### 3.1 Box-Cox Transformation

If the variance increases over time in the time series data, typically, we transform the data using algorithms like Box-cox transformations.

We can see from Figure 11 (scale added from 0 on the y-axis) that the variance does not seem to change over time for the given time period. Assuming we are ignoring the sharp dips in 2009 and 2020 due to the economic crisis and COVID-19, respectively,

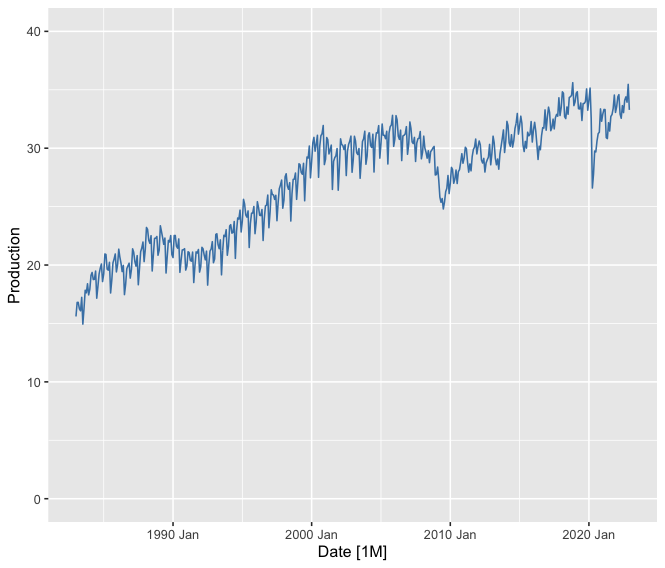
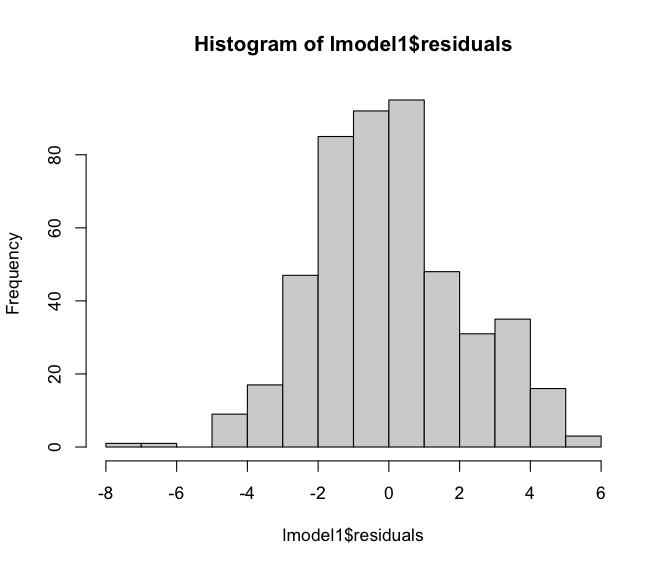
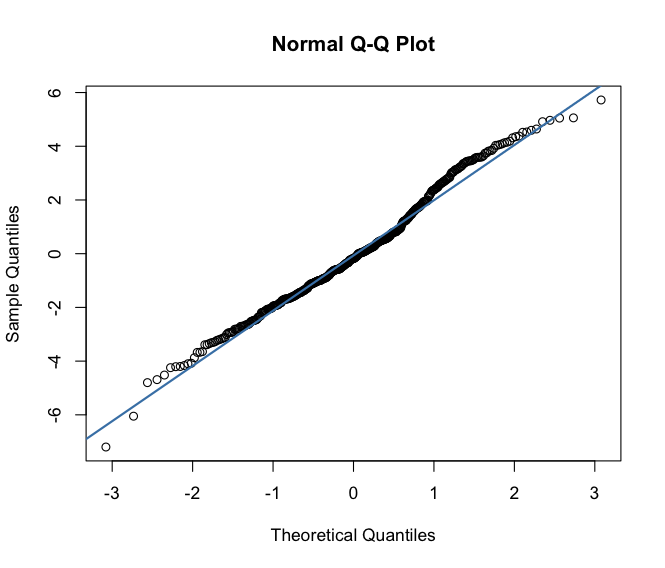


Figure 11: Time Plot for Total Industry Monthly Production (from 1983 Jan to 2022 Dec)

While a time plot can provide the initial insights, we need some statistical methods to prove this. So, we built a linear model to assess the distribution of the residuals. Figure 12(a) shows that the residuals are almost normally distributed (even though it’s right skewed, it’s only a minor skew). Also, from Figure 12(b), the Q-Q plot suggests that most of the data lies on the line, but only a few deviate. We also analyzed the scatterplot of the residuals vs fitted values in Figure 12(c). The residuals seem to be scattered enough.

|  |  |
| --- | --- |
| Figure 12(a): Histogram of the residuals | Figure 12(b): Q-Q plot |

### 

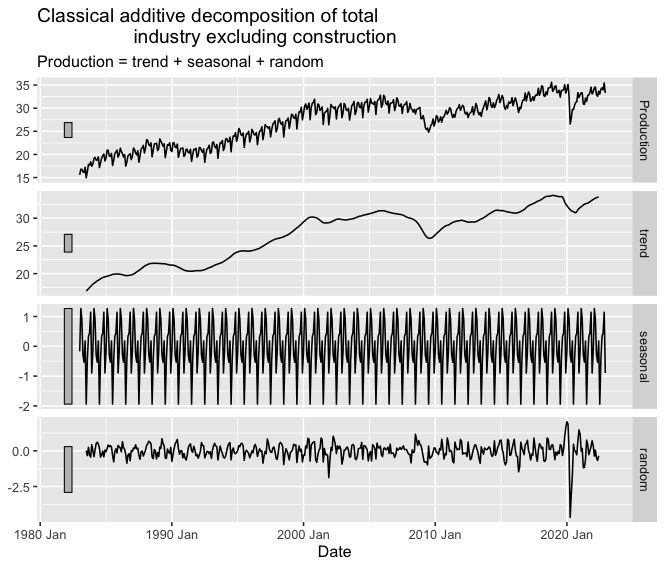
|  |
| --- |
| Figure 12(c): Scatterplot of the residuals vs fitted values |

From the above analysis, it is evident that a data transformation is unnecessary.

### 3.2 Decompositions

In this section, we decomposed the time series data from 1983 to 2022 to analyze trends and seasonality. We applied the Classical, X-11 and STL decompositions.

1. **Classical decomposition** is suitable for relatively consistent seasonality. Our data seemed to exhibit this, so we started with this. From Figure 13(a), we can see **an increasing trend over time,** and it also assumes that seasonality remains consistent. Other details are captured in the “random” (noise) graph, where we can see the sharp dip during 2020. To better explain these, we applied other decomposition techniques.

Figure 13(a): Classical decomposition for Total Industry Monthly Production (from 1983 Jan to 2022 Dec)

1. **STL decomposition,** on the other hand, is more flexible and can handle nonlinearities and changing patterns in trend and seasonality. Figure 13(b) shows that the long-term upward trend has been captured along with the dips in 2009 and 2020. Also, seasonality has been captured, which is relatively consistent and confirms the analysis using visual time series plots by month. Since all the patterns have been captured in “trend” and “season\_year,” we can also see no remainder.

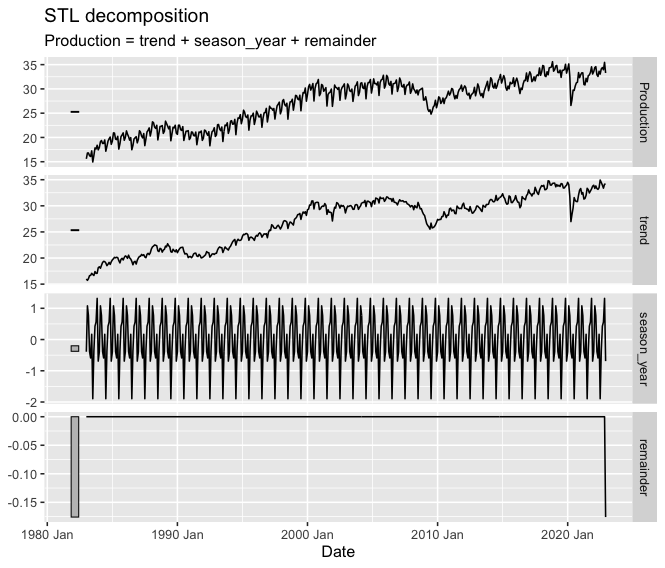


Figure 13(b): STL decomposition for Total Industry Monthly Production (from 1983 Jan to 2022 Dec)

1. **X-11 decomposition** is designed for seasonal adjustment, which is useful when we want to analyze the underlying trends and irregularities by removing the seasonal component.Figure 13(c) shows that the long-term upward trend has been captured well. Also, the seasonal effects seem to change over time. However, this decomposition technique didn’t capture the 2009 dip, whereas it captured the sharp dip in 2020. Overall, though, X-11 captures the seasonal components better than the others, as is evident from the Timeplot (Figure 11).

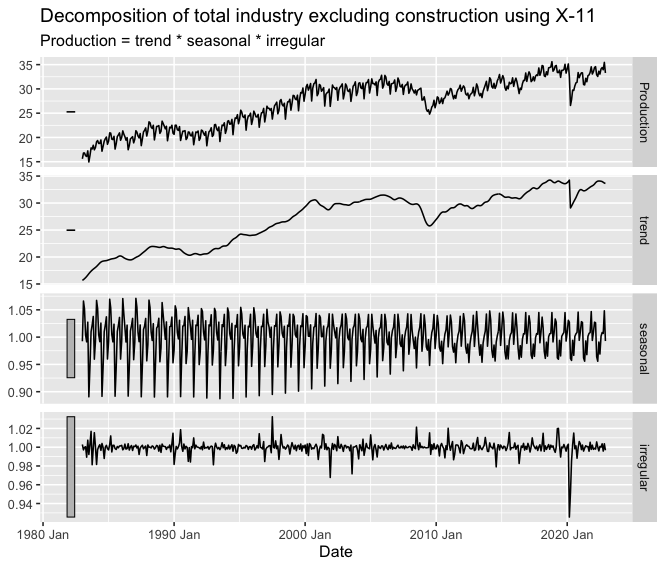


Figure 13(c): X-11 decomposition for Total Industry Monthly Production (from 1983 Jan to 2022 Dec)

## Forecasting and Analysis

In this section, we performed forecasting models for Total Industry Monthly Production time series data from 1983 to 2022 and analyzed the technique that delivered the best forecasts.

Initially, each team member independently analyzed the dataset and chose their best model. Subsequently, we compared the accuracy of various models and analyzed the residuals.

### 4.1 ETS

First, we applied the ETS model. The model is based on the exponential smoothing method, which assigns decaying weights to past data. It considers the individual pattern using the error, trend, and seasonal components for times series data exhibiting both trend and seasonality. The model can deal with both additive and multiplicative variations in the data and the damping trend.

We tried several models with different parameters and chose the best with the lowest AIC/BIC scores. The best model (Figure 14) has a multiplicative error term, a damping trend, and a multiplicative seasonality (M, Ad, M). As shown in Figure 2, the data trend is not a positive straight line but a curved line with occasional decreases. The damping factor reflects this decreasing trend. The multiplicative error term and seasonality suggest that the best model assumes the data variations change with the data level.

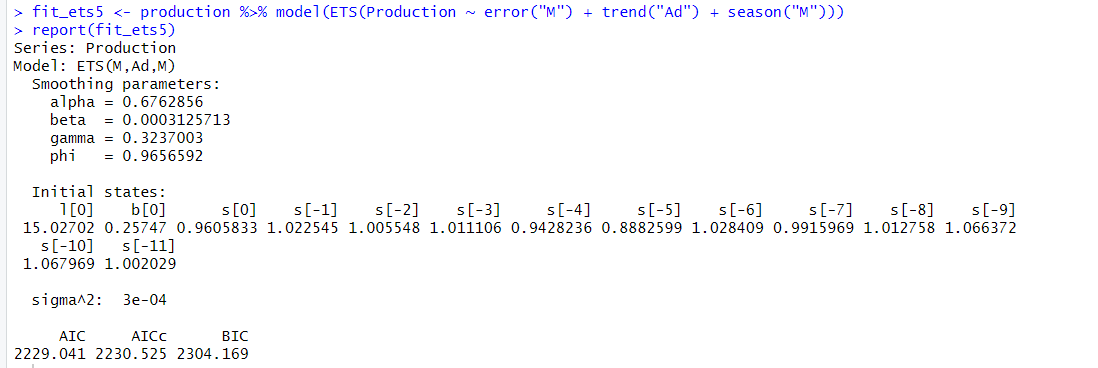


Figure 14: Summary of the best ETS model (M, Ad, M)

Summary of all the ETS models we experimented with:

|  |  |  |  |
| --- | --- | --- | --- |
| ETS parameters | AIC | AICc | BIC |
| **error (M) + trend (A) + season (A)** | 2255.34 | 2256.67 | 2326.3 |
| **error (A) + trend (A) + season (A)** | 2299.67 | 2300.99 | 2370.62 |
| **error (M) + trend (A) + season (M)** | 2270.32 | 2271.65 | 2341.28 |
| **error (A) + trend (A) + season (M)** | 2325.11 | 2326.44 | 2396.07 |
| **error (M) + trend (Ad) + season (A)** | 2244.91 | 2246.39 | 2320.03 |
| **error (M) + trend (Ad) + season (M)** | 2229.04 | 2230.53 | 2304.17 |

Figure 15 shows the forecast plot of the best ETS model for the following year. As we can see, the chosen model has identified the monthly seasonal pattern and seems to capture the trend, too.

A graph of a graph showing the growth of a company

Description automatically generated with medium confidence

Figure 15: Forecast plot of the following year's best ETS (M, Ad, M) model.

(80% and 95% prediction intervals are shown)

### 4.2 Seasonal ARIMA

For ARIMA, the data must be stationary, i.e. the mean and variance should be constant over time. But from Figures 1, 2 and 13, we can clearly see that our data exhibits strong seasonality and is not stationary. We differenced our data to make it stationary.

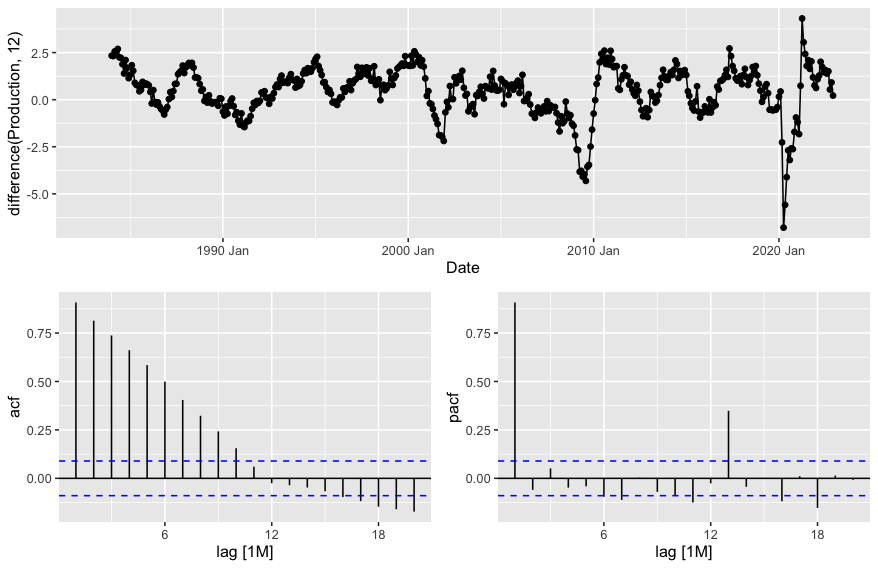


Figure 16: Differencing for Total Industry Monthly Production (from 1983 Jan to 2022 Dec)

After performing seasonal differencing, the ACF plot decays quickly, suggesting that a single differencing is enough to make it stationary. The PACF plot also has a sharp drop-off, which confirms that this data is stationary and that the difference in order 1 is enough.

Even though visual plots can be enough for the initial insights, this should be confirmed with a statistical test - the KPSS test, as it evaluates the null hypothesis that the time series is stationary in a deterministic trend.

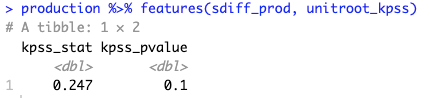


Figure 16(a): KPSS test for the differenced data

We can see from Figure 16(a) that the KPSS stat value is 0.245. The calculated p-value is 0.1, which is greater than 0.05. This means that at a 5% significance level, we don’t have enough evidence to reject the null hypothesis. So, this seasonally differenced data is stationary. This is also evident from Figure 16(b), which shows that a differencing of order one should be good enough to make the data stationary.

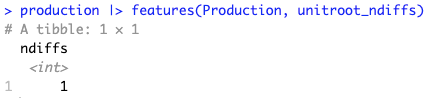


Figure 16(b): KPSS test suggesting the number of differencing required

We applied Seasonal ARIMA on the data post differencing, as this dataset shows strong seasonal trends. We tried multiple parameters for the non-seasonal part (p,d,q) and the seasonal part (P, D, Q) of Seasonal ARIMA.

The best model was with the parameter combination of (0, 1, 2) for non-seasonal and (0, 1, 1) for the seasonal part since it gave us the lowest AIC and BIC scores compared to others. This would mean that:

Non-seasonal part (0, 1, 2):

* AR(0): The model does not consider lagged values of the dependent variable, i.e. current value not directly influenced by its past values
* D(1): Differencing of order one is applied to make the data stationary, as discussed above
* MA(2): The model considers the last two forecast errors, i.e., the current forecast error is influenced by two previous errors

Seasonal part (0, 1, 1):

* Seasonal AR(0): The model doesn’t consider seasonal lagged values
* Seasonal D(1): Seasonal differencing of order one is applied, as discussed above
* Seasonal MA(1): The model considers the last seasonal forecast error

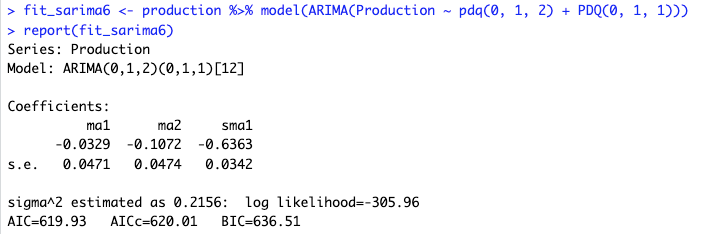


Figure 17: Summary of the best SARIMA model

Summary of all the SARIMA models we experimented with:

|  |  |  |  |
| --- | --- | --- | --- |
| SARIMA parameters | AIC | AICc | BIC |
| **pdq(2, 1, 2) + PDQ(0, 1, 1)** | 622.26 | 622.45 | 647.14 |
| **pdq(1, 1, 1) + PDQ(0, 1, 1)** | 623.36 | 623.44 | 639.94 |
| **pdq(2, 1, 2) + PDQ(1, 1, 1)** | 624.19 | 624.44 | 653.22 |
| **pdq(1, 1, 1) + PDQ(1, 1, 1)** | 625.34 | 625.47 | 646.07 |
| **pdq(1, 1, 2) + PDQ(1, 1, 1)** | 622.2 | 622.38 | 647.07 |
| **pdq(0, 1, 2) + PDQ(0, 1, 1)** | 619.93 | 620.01 | 636.51 |

Figure 18 shows the forecast plot of this chosen SARIMA model for the next three years. From the plot, the chosen model seems to capture the trends and seasonality well.

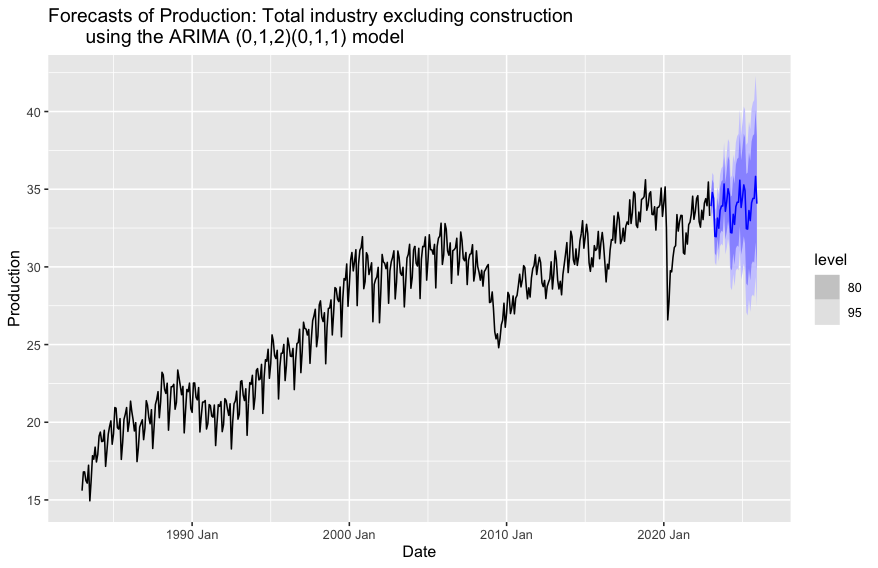


Figure 18: Forecast plot of the best SARIMA model for the next three years.

80% and 95% prediction intervals are shown

### 4.3 Other Models

#### **4.3.1 STL&ETS**

Seasonality can be challenging to handle if it exhibits nonlinearity. The STL decomposition can deal with this by subtracting the seasonal component from the data, after which we can use the seasonally adjusted data for forecasting. In our case, we extracted the seasonally adjusted data using the STL method and then decomposed the seasonally adjusted data again using the ETS method. This combination of decomposition methods aims to capture different underlying patterns in the time series data.

#### **4.3.2 Prophet**

As Figures 7 and 8 indicate, holidays and vacation seasons will affect total production. Therefore, it is reasonable to consider the holiday effects in our models. The prophet model is the most well-developed model incorporating holiday effects using dummy variables. We captured both the yearly seasonal pattern and the monthly seasonal pattern in the parameters.

#### **4.3.2 Combined Model**

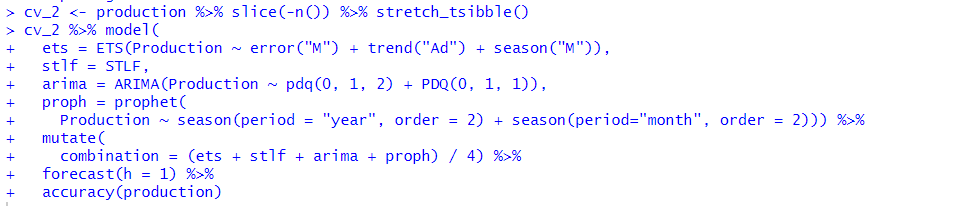
Sometimes, combining the results from multiple models can give more accurate forecasts than a single model. So, we decided to develop a combined model to include all the above models and average the result by the number of models. We had the best models in our combination for ETS and Seasonal ARIMA.

### 4.4 Model Evaluation

This section will compare the models we created (ETS, STL/ETS seasonal model, Seasonal ARIMA and Prophet). We used time series cross-validation to compare the accuracy of the four models. Also, we included a “combination” model, which takes the mean of all the other models, to produce 1-year forecasts. After selecting the best model, we checked the residual diagnostics.

We started with ETS models, and after comparing different methods, the model of choice for our time series data was ETS (M, Ad, M). Damped Holt’s method is our best model for Total Industry Monthly Production. The best ARIMA model was the parameter combination of (0, 1, 2) for the non-seasonal and (0, 1, 1) for the seasonal part.

The following figures (19 and 20) show that the best-performing model is Seasonal ARIMA, which has higher accuracy on the cross-validated performance measures. Even if we included the “combination” approach, the ARIMA model outperforms all the others and seems more accurate based on the RMSE, MAE, MAPE and MASE. The second best is the “combination” model.



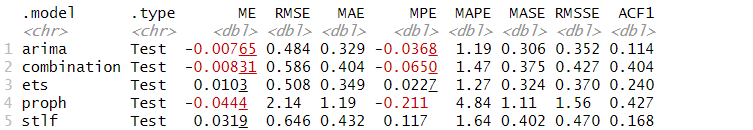


Figure 19: Summary of all the models

Summary of all the models we experimented with:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Models | RMSE | MAE | MAPE | MASE |
| **SARIMA: pdq(0, 1, 2) + PDQ(0, 1, 1)** | 0.484 | 0.329 | 1.19 | 0.306 |
| **Combination** | 0.586 | 0.404 | 1.47 | 0.375 |
| **ETS: error (M) + trend (Ad) + season (A)** | 0.508 | 0.349 | 1.27 | 0.324 |
| **Prophet** | 2.14 | 1.19 | 4.84 | 1.11 |
| **STL/ ETS** | 0.646 | 0.432 | 1.64 | 0.402 |

The residuals for the best model are shown in Figure 20. The ACF shows us one significant spike at lag 27, but this model generally has no major issues.

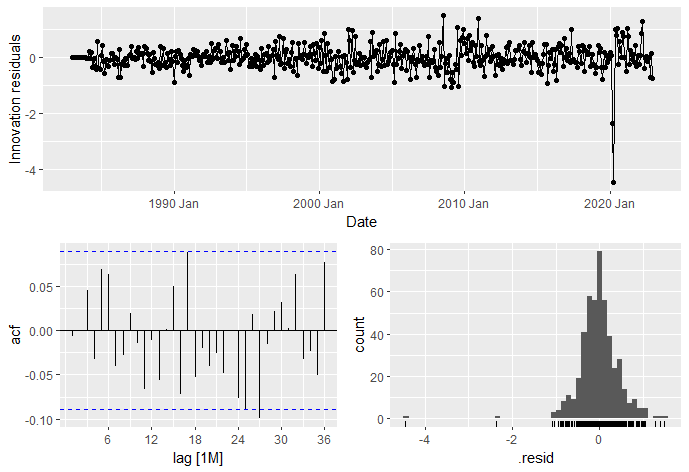


Figure 20: Residuals from the SARIMA model

### 4.5 Forecasting

In this last section, we generated and plotted forecasts from our best Seasonal ARIMA model for the next five years. We also calculated an 80% and 95% prediction interval for the best model. Forecasts from the chosen model are shown in the following figures.

We plotted forecasts for the next five years using all the data (Figure 21) and the data after 2003 (Figure 22). In general, both the forecasts have captured the seasonal pattern and the slightly increasing trend quite well. The data clearly identified an obvious seasonal pattern, with peaks in November each year in the total industry production.

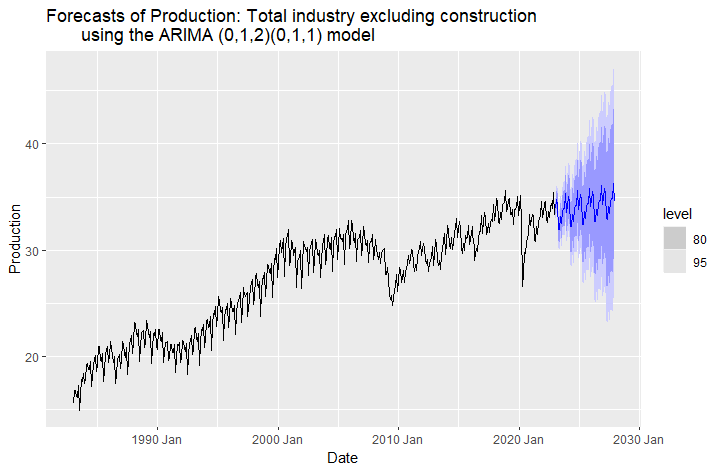


Figure 21:Forecasts from the SARIMA model fitted to all the data production since 1983 for the next five years.

(80% and 95% prediction intervals are shown)

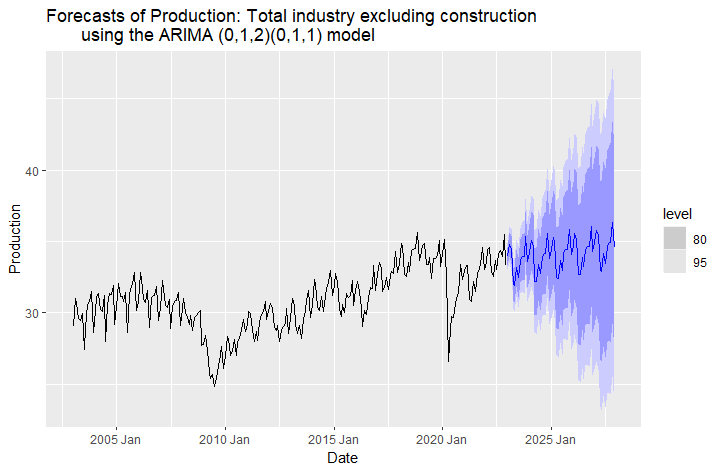


Figure 22:Forecasts from the SARIMA model fitted to all the data production since 2023 for the next five years.

80% and 95% prediction intervals are shown

### 4.6 Summary

In this project, we analyzed the Canadian total Industry Monthly Production from 1983 to 2022. The data has a strong seasonality within each year and a general increasing trend over this period. There is also evidence of some strong cyclic behaviour.

We applied different forecasting models (ETS, ARIMA, STL/ETS seasonal model, Prophet) and analyzed the accuracy to select the better-performing method. Based on the RMSE, MAE, MAPE and MASE, the technique that delivers the best forecasts for our data is the Seasonal Arima.

## References

FRED. (2023, August 10*). Production: Industry: Total industry: Total industry excluding*

*construction for Canada. (CANPRINTO01MLM)*. [Data set].

<https://fred.stlouisfed.org/series/CANPRINTO01MLM>

Hyndman, R.J., & Athanasopoulos, G. (2021). Forecasting: principles and practice (3rd

edition). OTexts: Melbourne, Australia.<https://otexts.com/fpp3/>

## Appendix

# load libraries  
library(fpp3)

## ── Attaching packages ────────────────────────────────────────────── fpp3 0.5 ──

## ✔ tibble 3.2.1 ✔ tsibble 1.1.3  
## ✔ dplyr 1.1.2 ✔ tsibbledata 0.4.1  
## ✔ tidyr 1.3.0 ✔ feasts 0.3.1  
## ✔ lubridate 1.9.2 ✔ fable 0.3.3  
## ✔ ggplot2 3.4.2 ✔ fabletools 0.3.3

## ── Conflicts ───────────────────────────────────────────────── fpp3\_conflicts ──  
## ✖ lubridate::date() masks base::date()  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ tsibble::intersect() masks base::intersect()  
## ✖ tsibble::interval() masks lubridate::interval()  
## ✖ dplyr::lag() masks stats::lag()  
## ✖ tsibble::setdiff() masks base::setdiff()  
## ✖ tsibble::union() masks base::union()

library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ forcats 1.0.0 ✔ readr 2.1.4  
## ✔ purrr 1.0.1 ✔ stringr 1.5.0

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ tsibble::interval() masks lubridate::interval()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(tsibble)  
library(knitr)  
library(caret)

## Loading required package: lattice  
##   
## Attaching package: 'caret'  
##   
## The following object is masked from 'package:purrr':  
##   
## lift  
##   
## The following objects are masked from 'package:fabletools':  
##   
## MAE, RMSE

library(MASS)

##   
## Attaching package: 'MASS'  
##   
## The following object is masked from 'package:dplyr':  
##   
## select

library(dplyr)  
library(fable.prophet)

## Loading required package: Rcpp

library(readxl)  
  
# load dataset  
all\_production <- read\_excel("CANPRINTO01MLM\_production.xls")  
  
# 1. data cleaning  
all\_production <- all\_production %>% mutate(Production = Production/1e9)  
all\_production <- all\_production %>% mutate(Date = yearmonth(Date)) %>% as\_tsibble(index = Date) %>% dplyr::select(Date, Production)  
  
  
# 2. data visualization  
# Time plot of all data  
all\_production %>% autoplot(Production) +   
 labs(y = "Billions Canadian Dollar", x="Month",  
 title = "Production: Total industry excluding construction (Canada)",  
 subtitle = "1961-2022")

A graph showing the construction of a company

Description automatically generated

# Time plot from 1983 to 2003  
all\_production %>% filter(year(Date) <= 2003 & year(Date) >= 1983) %>%   
 autoplot(Production, col = 'purple') +  
 labs(y = "Billions Canadian Dollar", x="Month",  
 title = "Production: Total industry excluding construction (Canada)",  
 subtitle = "1983-2003")

A graph showing a growth of construction

Description automatically generated

# Time plot from 2003 to 2022  
all\_production |> filter(year(Date) >= 2003) |>   
 autoplot(Production, col = 'blue') +  
 labs(y = "Billions Canadian Dollar", x="Month",  
 title = "Production: Total industry excluding construction (Canada)",  
 subtitle = "2003-2022")

A graph showing a number of construction workers

Description automatically generated with medium confidence

# Time plot from 1983 to 2022  
all\_production %>% filter(year(Date) >= 1983) %>%  
 autoplot(Production, col = 'darkblue') +  
 labs(y = "Billions Canadian Dollar", x="Month",  
 title = "Production: Total industry excluding construction (Canada)",  
 subtitle = "1983-2022")

A graph showing a number of construction workers

Description automatically generated

# Seasonal plot from 1983 to 2008  
all\_production |>  
 filter(year(Date) <= 2008 & year(Date) >= 1983) |>  
 gg\_season(Production, labels = "both") +  
 labs(y = "Billions Canadian Dollar", x="Month",  
 title = "Seasonal plot - Production: Total industry excluding construction (Canada)",  
 subtitle = "1983-2008")

A graph showing the number of the months and months

Description automatically generated with medium confidence

# Seasonal plot from 2008 to 2022  
all\_production |>  
 filter(year(Date) >= 2008) |>  
 gg\_season(Production, labels = "both") +  
 labs(y = "Billions Canadian Dollar", x="Month",  
 title = "Seasonal plot - Production: Total industry excluding construction (Canada)",  
 subtitle = "2008-2022")

A graph of different colored lines

Description automatically generated

# Seasonal subseries from 1983 to 2003  
all\_production |>   
 filter(year(Date) <= 2003 & year(Date) >= 1983) |>   
 gg\_subseries(Production) +  
 labs(  
 y = "Billions Canadian Dollar",  
 title = "Seasonal subseries - Production: Total industry excluding construction (Canada)",  
 subtitle = "1983-2003")

A graph of different types of sales

Description automatically generated with medium confidence

# Seasonal subseries from 2003 to 2022  
all\_production |>  
 filter(year(Date) >= 2003) |>  
 gg\_subseries(Production) +  
 labs(  
 y = "Billions Canadian Dollar",  
 title = "Seasonal subseries - Production: Total industry excluding construction (Canada)",  
 subtitle = "2003-2022")

A graph showing the number of the month

Description automatically generated with medium confidence

# ACF plot from 1983 to 2003  
all\_production %>%   
 filter(year(Date) <= 2003 & year(Date) >= 1983) %>%   
 ACF(Production, lag\_max = 120) %>% autoplot() +  
 labs(title = "Autocorrelation function of monthly Production: Total industry excluding construction (Canada)",  
 subtitle = "1983-2003")

A graph of a graph showing the growth of a plant

Description automatically generated with medium confidence

# ACF plot from 2003 to 2022  
all\_production %>%  
 filter(year(Date) >= 2003) |>  
 ACF(Production, lag\_max = 120) %>%  
 autoplot() +  
 labs(title = "Autocorrelation function of monthly Production: Total industry excluding construction (Canada)",  
 subtitle = "2003-2022")

A graph of a graph showing the growth of a number of production

Description automatically generated with medium confidence

# 3. data transformation  
# Do we need Data transformation? Let's check how the observations' variance change over time  
production = all\_production %>%  
 filter(year(Date) >= 1983)  
  
production %>% autoplot(Production, col = 'steelblue') + ylim(0, 40)

A graph showing the growth of a stock market

Description automatically generated

# From the timeplot, ignoring the dip in 2009 and 2020 (outlier years), we can see that the variance doesn't change much over time.  
# So there is no need for a transformation here.  
  
# Confirming this with a linear regression model and analyzing residual plots  
lmodel1 <- lm(Production ~ Date, data = production)  
  
# Histogram of the residuals  
hist(lmodel1$residuals)

A graph of a number of individuals

Description automatically generated

# QQplot for the residuals  
qqnorm(lmodel1$residuals)  
qqline(lmodel1$residuals, col = "steelblue", lwd = 2)

A graph of a normal q-q plot

Description automatically generated

# Fitted vs residual plot  
plot(lmodel1)

A graph of a graph with numbers and lines

Description automatically generated with medium confidenceA graph of a normal q-q

Description automatically generatedA graph of a number of dots

Description automatically generated with medium confidenceA graph of a number of circles

Description automatically generated with medium confidence

# From the above plots, the histogram of the residuals seem to be normally distributed. It is slightly left skewed, but not much  
# Coming to the QQ plot, the residuals almost form a straight line, only a few data points deviate from the line  
# With the above data, we can confirm that it is not necessary for a transformation.  
  
bc <- boxcox(lmodel1, plotit = TRUE)

A graph with numbers and lines

Description automatically generated

lambda <- bc$x[which.max(bc$y)]  
# lambda = 1.11 which means no transformation is needed.  
  
  
# Decomposition  
# classical decomposition  
production |>  
 model(  
 classical\_decomposition(Production, type = "additive")  
 ) |>  
 components() |>  
 autoplot() +  
 labs(title = "Classical additive decomposition of total  
 industry excluding construction")

## Warning: Removed 6 rows containing missing values (`geom\_line()`).

A graph of different types of graphs

Description automatically generated with medium confidence

# x-11  
x11\_dcmp <- production |>  
 model(x11 = X\_13ARIMA\_SEATS(Production ~ x11())) |>  
 components()  
autoplot(x11\_dcmp) +  
 labs(title =  
 "Decomposition of total industry excluding construction using X-11")

A graph of a number of different types of industry

Description automatically generated with medium confidence

# STL  
dcmp <- production %>% model(stl = STL(Production ~ trend(window = 3) + season(window = 'periodic'), robust = TRUE))  
dcmp %>% components(dcmp) %>% autoplot()

A graph of a graph of a graph

Description automatically generated with medium confidence

# 4. models  
# ETS  
# model development  
  
fit\_ets1 <- production %>% model(ETS(Production ~ error("M") + trend("A") + season("A")))  
report(fit\_ets1)

## Series: Production   
## Model: ETS(M,A,A)   
## Smoothing parameters:  
## alpha = 0.6696642   
## beta = 0.0001013185   
## gamma = 0.3286568   
##   
## Initial states:  
## l[0] b[0] s[0] s[-1] s[-2] s[-3] s[-4]  
## 15.75607 0.05588438 -0.4465416 0.6975772 0.4381914 0.3052911 -0.6132915  
## s[-5] s[-6] s[-7] s[-8] s[-9] s[-10] s[-11]  
## -2.285853 0.2474023 -0.3317066 0.04033791 0.6653382 1.434321 -0.1510663  
##   
## sigma^2: 3e-04  
##   
## AIC AICc BIC   
## 2255.341 2256.666 2326.295

# AIC AICc BIC   
# 2255.341 2256.666 2326.295  
  
fit\_ets2 <- production %>% model(ETS(Production ~ error("A") + trend("A") + season("A")))  
report(fit\_ets2)

## Series: Production   
## Model: ETS(A,A,A)   
## Smoothing parameters:  
## alpha = 0.6847903   
## beta = 0.0001045559   
## gamma = 0.3152048   
##   
## Initial states:  
## l[0] b[0] s[0] s[-1] s[-2] s[-3] s[-4] s[-5]  
## 15.87585 0.03259521 -0.763358 0.5213164 0.5058286 0.1690047 -1.080283 -1.87863  
## s[-6] s[-7] s[-8] s[-9] s[-10] s[-11]  
## 0.2324006 -0.4089353 0.01491426 1.282422 1.481259 -0.07593915  
##   
## sigma^2: 0.2418  
##   
## AIC AICc BIC   
## 2299.667 2300.991 2370.621

# AIC AICc BIC   
# 2299.667 2300.991 2370.621  
  
fit\_ets3 <- production %>% model(ETS(Production ~ error("M") + trend("A") + season("M")))  
report(fit\_ets3)

## Series: Production   
## Model: ETS(M,A,M)   
## Smoothing parameters:  
## alpha = 0.5995057   
## beta = 0.01449151   
## gamma = 0.3152204   
##   
## Initial states:  
## l[0] b[0] s[0] s[-1] s[-2] s[-3] s[-4] s[-5]  
## 14.96606 0.0244874 0.9657772 1.032187 1.009967 1.003578 0.9543986 0.9010202  
## s[-6] s[-7] s[-8] s[-9] s[-10] s[-11]  
## 1.029674 1.001739 1.001242 1.051415 1.060053 0.9889501  
##   
## sigma^2: 3e-04  
##   
## AIC AICc BIC   
## 2270.324 2271.649 2341.279

# AIC AICc BIC   
# 2270.324 2271.649 2341.279  
  
fit\_ets4 <- production %>% model(ETS(Production ~ error("A") + trend("A") + season("M")))  
report(fit\_ets4)

## Series: Production   
## Model: ETS(A,A,M)   
## Smoothing parameters:  
## alpha = 0.6957052   
## beta = 0.0603935   
## gamma = 0.3042833   
##   
## Initial states:  
## l[0] b[0] s[0] s[-1] s[-2] s[-3] s[-4] s[-5]  
## 15.82959 0.1848308 0.9707738 1.01778 0.9985835 1.006254 0.9257813 0.8770355  
## s[-6] s[-7] s[-8] s[-9] s[-10] s[-11]  
## 1.017421 0.9908205 1.016566 1.085202 1.084184 1.009598  
##   
## sigma^2: 0.2549  
##   
## AIC AICc BIC   
## 2325.111 2326.435 2396.065

# AIC AICc BIC   
# 2325.111 2326.435 2396.065  
  
#ETS (M, Ad, M)  
fit\_ets5 <- production %>% model(ETS(Production ~ error("M") + trend("Ad") + season("M")))  
report(fit\_ets5)

## Series: Production   
## Model: ETS(M,Ad,M)   
## Smoothing parameters:  
## alpha = 0.6762856   
## beta = 0.0003125713   
## gamma = 0.3237003   
## phi = 0.9656592   
##   
## Initial states:  
## l[0] b[0] s[0] s[-1] s[-2] s[-3] s[-4] s[-5]  
## 15.02702 0.25747 0.9605833 1.022545 1.005548 1.011106 0.9428236 0.8882599  
## s[-6] s[-7] s[-8] s[-9] s[-10] s[-11]  
## 1.028409 0.9915969 1.012758 1.066372 1.067969 1.002029  
##   
## sigma^2: 3e-04  
##   
## AIC AICc BIC   
## 2229.041 2230.525 2304.169

# AIC AICc BIC   
# 2229.041 2230.525 2304.169  
  
fit\_ets6 <- production %>% model(ETS(Production ~ error("M") + trend("Ad") + season("A")))  
report(fit\_ets6)

## Series: Production   
## Model: ETS(M,Ad,A)   
## Smoothing parameters:  
## alpha = 0.6638954   
## beta = 0.0001184304   
## gamma = 0.3360297   
## phi = 0.952822   
##   
## Initial states:  
## l[0] b[0] s[0] s[-1] s[-2] s[-3] s[-4]  
## 15.44176 0.2347784 -0.7930652 0.5774264 0.3568898 0.4195231 -0.9724458  
## s[-5] s[-6] s[-7] s[-8] s[-9] s[-10] s[-11]  
## -1.897869 0.06039722 -0.3867147 -0.03166909 1.102266 1.443206 0.1220546  
##   
## sigma^2: 3e-04  
##   
## AIC AICc BIC   
## 2244.905 2246.389 2320.033

# AIC AICc BIC  
# 2244.905 2246.389 2320.033   
# The best ETS model is fit\_est5.  
  
# Forecast plot of the best ETS MODEL  
forecast(fit\_ets5, h=12) %>%  
 autoplot(production) +  
 labs(title = "Forecasts of Production: Total industry excluding construction  
 using the ETS (M,Ad,M) model")

A graph showing the growth of the ets

Description automatically generated

# ARIMA  
# differencing plot  
production %>% gg\_tsdisplay(difference(Production, 12), plot\_type='partial', lag=20) ###### ANITHA

## Warning: Removed 12 rows containing missing values (`geom\_line()`).

## Warning: Removed 12 rows containing missing values (`geom\_point()`).

A graph of different types of data

Description automatically generated with medium confidence

#Double differencing  
production %>% gg\_tsdisplay(difference(difference(Production, 12)), plot\_type='partial', lag=120)

## Warning: Removed 13 rows containing missing values (`geom\_line()`).

## Warning: Removed 13 rows containing missing values (`geom\_point()`).

A graph of data on a graph

Description automatically generated with medium confidence

#the PACF is suggestive of an AR (2) model, the ACF suggests an MA(2) model.  
  
# non seasonal differencing  
production <- production %>% mutate(diff\_prod = difference(Production))  
production %>% autoplot(diff\_prod)

## Warning: Removed 1 row containing missing values (`geom\_line()`).

A graph showing a graph showing a wave

Description automatically generated with medium confidence

production %>% ACF(diff\_prod, lag\_max = 120) %>% autoplot()

A graph with a line graph

Description automatically generated with medium confidence

production %>% PACF(diff\_prod, lag\_max = 120) %>% autoplot()

A graph with a line graph

Description automatically generated

# seasonal differencing  
production <- production %>% mutate(sdiff\_prod = difference(Production, 12))  
production %>% autoplot(sdiff\_prod)

## Warning: Removed 12 rows containing missing values (`geom\_line()`).

A graph showing the growth of the stock market

Description automatically generated

production %>% ACF(sdiff\_prod, lag\_max = 120) %>% autoplot()

A graph with a line graph

Description automatically generated

production %>% PACF(sdiff\_prod, lag\_max = 120) %>% autoplot()

A graph with a line graph

Description automatically generated

# stationary test  
production %>% features(Production, unitroot\_kpss)

## # A tibble: 1 × 2  
## kpss\_stat kpss\_pvalue  
## <dbl> <dbl>  
## 1 7.03 0.01

#p-value 0.01, the data are not stationary  
  
production %>% features(diff\_prod, unitroot\_kpss)

## # A tibble: 1 × 2  
## kpss\_stat kpss\_pvalue  
## <dbl> <dbl>  
## 1 0.0497 0.1

#p-value 0.1, the differenced data appear stationary  
  
production %>% features(sdiff\_prod, unitroot\_kpss)

## # A tibble: 1 × 2  
## kpss\_stat kpss\_pvalue  
## <dbl> <dbl>  
## 1 0.247 0.1

#p-value 0.1, the differenced data appear stationary  
  
production |> features(Production, unitroot\_ndiffs)

## # A tibble: 1 × 1  
## ndiffs  
## <int>  
## 1 1

#ndiffs  
#<int>  
# 1 1  
#As we saw from the KPSS tests above, one difference is required to make the data stationary.  
  
# Display the ACF/PACF of seasonally differenced data  
production %>% gg\_tsdisplay(sdiff\_prod, plot\_type='partial', lag = 120)

## Warning: Removed 12 rows containing missing values (`geom\_line()`).

## Warning: Removed 12 rows containing missing values (`geom\_point()`).

A graph of a graph of a graph

Description automatically generated with medium confidence

# Display the ACF/PACF of double differenced data  
production %>% gg\_tsdisplay(sdiff\_prod %>% difference(), plot\_type='partial', lag = 120)

## Warning: Removed 13 rows containing missing values (`geom\_line()`).

## Warning: Removed 13 rows containing missing values (`geom\_point()`).

A graph of data on a graph

Description automatically generated with medium confidence

# model development  
fit\_sarima1 <- production %>% model(ARIMA(Production ~ pdq(2, 1, 2) + PDQ(0, 1, 1)))  
report(fit\_sarima1)

## Series: Production   
## Model: ARIMA(2,1,2)(0,1,1)[12]   
##   
## Coefficients:  
## ar1 ar2 ma1 ma2 sma1  
## -0.4659 -0.0139 0.4294 -0.1119 -0.6370  
## s.e. 0.3999 0.3407 0.3984 0.3427 0.0339  
##   
## sigma^2 estimated as 0.2157: log likelihood=-305.13  
## AIC=622.26 AICc=622.45 BIC=647.14

# AIC=622.26 AICc=622.45 BIC=647.14  
  
fit\_sarima2 <- production %>% model(ARIMA(Production ~ pdq(1, 1, 1) + PDQ(0, 1, 1)))  
report(fit\_sarima2)

## Series: Production   
## Model: ARIMA(1,1,1)(0,1,1)[12]   
##   
## Coefficients:  
## ar1 ma1 sma1  
## 0.4537 -0.5189 -0.6378  
## s.e. 0.2973 0.2836 0.0340  
##   
## sigma^2 estimated as 0.2172: log likelihood=-307.68  
## AIC=623.36 AICc=623.44 BIC=639.94

# AIC=623.36 AICc=623.44 BIC=639.940  
  
fit\_sarima3 <- production %>% model(ARIMA(Production ~ pdq(2, 1, 2) + PDQ(1, 1, 1)))  
report(fit\_sarima3)

## Series: Production   
## Model: ARIMA(2,1,2)(1,1,1)[12]   
##   
## Coefficients:  
## ar1 ar2 ma1 ma2 sar1 sma1  
## -0.4886 -0.0320 0.4516 -0.0943 -0.0186 -0.6275  
## s.e. 0.4311 0.3663 0.4305 0.3703 0.0685 0.0495  
##   
## sigma^2 estimated as 0.2161: log likelihood=-305.1  
## AIC=624.19 AICc=624.44 BIC=653.22

# AIC=624.19 AICc=624.44 BIC=653.22  
  
fit\_sarima4 <- production %>% model(ARIMA(Production ~ pdq(1, 1, 1) + PDQ(1, 1, 1)))  
report(fit\_sarima4)

## Series: Production   
## Model: ARIMA(1,1,1)(1,1,1)[12]   
##   
## Coefficients:  
## ar1 ma1 sar1 sma1  
## 0.4495 -0.5150 -0.0075 -0.6338  
## s.e. 0.2965 0.2831 0.0676 0.0489  
##   
## sigma^2 estimated as 0.2176: log likelihood=-307.67  
## AIC=625.34 AICc=625.47 BIC=646.07

# AIC=625.34 AICc=625.47 BIC=646.07  
  
fit\_sarima5 <- production %>% model(ARIMA(Production ~ pdq(1, 1, 2) + PDQ(1, 1, 1)))  
report(fit\_sarima5)

## Series: Production   
## Model: ARIMA(1,1,2)(1,1,1)[12]   
##   
## Coefficients:  
## ar1 ma1 ma2 sar1 sma1  
## -0.4558 0.419 -0.1263 -0.0177 -0.6280  
## s.e. 0.2582 0.257 0.0469 0.0678 0.0491  
##   
## sigma^2 estimated as 0.2157: log likelihood=-305.1  
## AIC=622.2 AICc=622.38 BIC=647.07

# AIC=622.2 AICc=622.38 BIC=647.07  
  
fit\_sarima6 <- production %>% model(ARIMA(Production ~ pdq(0, 1, 2) + PDQ(0, 1, 1)))  
report(fit\_sarima6)

## Series: Production   
## Model: ARIMA(0,1,2)(0,1,1)[12]   
##   
## Coefficients:  
## ma1 ma2 sma1  
## -0.0329 -0.1072 -0.6363  
## s.e. 0.0471 0.0474 0.0342  
##   
## sigma^2 estimated as 0.2156: log likelihood=-305.96  
## AIC=619.93 AICc=620.01 BIC=636.51

# AIC=619.93 AICc=620.01 BIC=636.51  
# The best ARIMA model so far is fit6.  
  
# Forecast plot of the best SARIMA model  
forecast(fit\_sarima6, h=12) %>%  
 autoplot(production)

A graph showing the growth of the stock market

Description automatically generated

# STL\_+ ETS  
STLF <- decomposition\_model(  
 STL(Production ~ season(window = Inf)),  
 ETS(season\_adjust ~ season("N"))  
)  
  
# Prophet  
proph = prophet(  
 Production ~ season(period = "year", order = 2) +   
 season(period="month", order = 2))  
  
  
  
# 5. evaluation  
#comparing models  
cv <- production %>% slice(-n()) %>% stretch\_tsibble()  
cv %>% model(  
 ets = ETS(Production ~ error("M") + trend("Ad") + season("M")),  
 stlf = STLF,  
 arima = ARIMA(Production ~ pdq(0, 1, 2) + PDQ(0, 1, 1)),  
 proph = prophet(  
 Production ~ season(period = "year", order = 2) + season(period="month", order = 2))) %>%   
 mutate(  
 combination = (ets + stlf + arima + proph) / 4) %>%   
 forecast(h = 1) %>%   
 accuracy(production)

## Warning: 1 error encountered for dcmp  
## [1] series is not periodic or has less than two periods

## n.changepoints greater than number of observations. Using 0  
## n.changepoints greater than number of observations. Using 1  
## n.changepoints greater than number of observations. Using 2  
## n.changepoints greater than number of observations. Using 3  
## n.changepoints greater than number of observations. Using 3  
## n.changepoints greater than number of observations. Using 4  
## n.changepoints greater than number of observations. Using 5  
## n.changepoints greater than number of observations. Using 6  
## n.changepoints greater than number of observations. Using 7  
## n.changepoints greater than number of observations. Using 7  
## n.changepoints greater than number of observations. Using 8  
## n.changepoints greater than number of observations. Using 9  
## n.changepoints greater than number of observations. Using 10  
## n.changepoints greater than number of observations. Using 11  
## n.changepoints greater than number of observations. Using 11  
## n.changepoints greater than number of observations. Using 12  
## n.changepoints greater than number of observations. Using 13  
## n.changepoints greater than number of observations. Using 14  
## n.changepoints greater than number of observations. Using 15  
## n.changepoints greater than number of observations. Using 15  
## n.changepoints greater than number of observations. Using 16  
## n.changepoints greater than number of observations. Using 17  
## n.changepoints greater than number of observations. Using 18  
## n.changepoints greater than number of observations. Using 19  
## n.changepoints greater than number of observations. Using 19  
## n.changepoints greater than number of observations. Using 20  
## n.changepoints greater than number of observations. Using 21  
## n.changepoints greater than number of observations. Using 22  
## n.changepoints greater than number of observations. Using 23  
## n.changepoints greater than number of observations. Using 23  
## n.changepoints greater than number of observations. Using 24

## Warning: 18 errors (2 unique) encountered for ets  
## [12] A seasonal ETS model cannot be used for this data.  
## [6] Not enough data to estimate this ETS model.

## Warning: 4 errors (3 unique) encountered for stlf  
## [1] In argument: `cmp = map(.fit, components)`.  
## [2] Not enough data to estimate this ETS model.  
## [1] only 1 case, but 2 variables

## Warning: 35 errors (1 unique) encountered for arima  
## [35] Not enough data to estimate a model with those options of P and Q. Consider allowing smaller values of P and Q to be selected.

## Warning: 1 error encountered for proph  
## [1] Dataframe has less than 2 non-NA rows.

## # A tibble: 5 × 10  
## .model .type ME RMSE MAE MPE MAPE MASE RMSSE ACF1  
## <chr> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 arima Test -0.00765 0.484 0.329 -0.0368 1.19 0.306 0.352 0.114  
## 2 combination Test -0.00790 0.586 0.404 -0.0636 1.47 0.375 0.427 0.403  
## 3 ets Test 0.0103 0.508 0.349 0.0227 1.27 0.324 0.370 0.240  
## 4 proph Test 0.0725 4.00 1.29 0.479 5.40 1.20 2.91 0.221  
## 5 stlf Test 0.0319 0.646 0.432 0.117 1.64 0.402 0.470 0.168

#.model .type ME RMSE MAE MPE MAPE MASE RMSSE ACF1  
#<chr> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
#1 arima Test -0.00765 0.484 0.329 -0.0368 1.19 0.306 0.352 0.114  
#2 combination Test -0.00831 0.586 0.404 -0.0650 1.47 0.375 0.427 0.404  
#3 ets Test 0.0103 0.508 0.349 0.0227 1.27 0.324 0.370 0.240  
#4 proph Test -0.0444 2.14 1.19 -0.211 4.84 1.11 1.56 0.427  
#5 stlf Test 0.0319 0.646 0.432 0.117 1.64 0.402 0.470 0.168  
  
fit\_sarima6 |>   
 gg\_tsresiduals(lag=36)

A graph of data and a graph of data

Description automatically generated with medium confidence

# 6. forecasting  
fit\_sarima6 %>% forecast(h = 60) %>% autoplot(production) +  
 labs(title = "Forecasts of Production: Total industry excluding construction  
 using the ARIMA (0,1,2)(0,1,1) model")

A graph of a graph showing the growth of the company's production

Description automatically generated with medium confidence

fit\_sarima6 %>% forecast(h = 60) %>%  
 autoplot(production |> filter\_index("2003 Jan" ~ .)) +  
 labs(title = "Forecasts of Production: Total industry excluding construction  
 using the ARIMA (0,1,2)(0,1,1) model")

A graph of a graph showing the growth of the company's production

Description automatically generated with medium confidence