**Forecasting Automotive Demand In Quebec**

**Mahalakshmi Jayapal and Mohammad Mahmoudighaznavi**

**Shiley-Marcos School of Engineering, University of San Diego**

**ADS 506: Applied Time Series Analysis**

**Professor Mikael Graindorge**

**December 02, 2022**

# Abstract

Auto sales is considered an important indicator of economic growth. Automotive industry is one of the most important sectors for the developed and developing economies. An unknown demand for vehicles is a significant issue that decreases productivity of the production environment and sales of the economy. Demand forecasting reduces risks in making better financial decisions which increases profit and cash flow, improves resource allocation, and creates opportunities for growth. This makes forecasting very crucial for the production and inventory management of the automobile industry. Time series forecasting uses the previous data to determine the trend and seasonality of the sales for the future. For this project different models were used to forecast the automobile demand for the city of Quebec in Canada. The city of Quebec was chosen because of the strong seasonality in sales which can be used to study the demand based on the seasonality in the data. The sales forecast helps to analyze the economic trend and also helps the local automotive industry with planning and execution of the production units and the showrooms, with increasing or decreasing the production of the cars and increasing or decreasing the hours and staff of the showrooms based on the forecast. The dataset for car sales in Quebec was acquired from Kaggle. Naive, seasonal naive, regression with seasonality and trend as predictors, exponential smoothing, and ARIMA models were used to make the forecast and the AIC score, MAPE, and RMSE was used as performance indicators. Seasonal ARIMA performed the best among all the models used.

**Table of Contents**

[Abstract 2](#_Toc120863805)

[Introduction 4](#_Toc120863806)

[Problem Statement 4](#_Toc120863807)

[Data Acquisition 4](#_Toc120863808)

[Objective 5](#_Toc120863809)

[Literature Review 5](#_Toc120863810)

[Exploratory Data Analysis 6](#_Toc120863811)

[Data Wrangling and Pre-Processing 9](#_Toc120863812)

[Data Splitting 9](#_Toc120863813)

[Model Strategies 10](#_Toc120863814)

[Evaluation Metrics 11](#_Toc120863815)

[Model Evaluation 11](#_Toc120863816)

[Conclusion 14](#_Toc120863817)

[References 17](#_Toc120863818)

[Appendix A 18](#_Toc120863819)

**Introduction**

Forecasting demand is very important in any business to make any financial decisions. Automobiles have a heavier production cost, so this makes forecasting the sales in that industry to be very crucial. Our project involves forecasting automobile demand for a city in Canada. Canada is usually associated with extremely cold weather. Winter in Canada is characterized by very low temperatures and frequent snowfalls. This seasonal change also reflects the demand for automobiles. We obtained our dataset for the ‘Monthly Car Sales in Quebec 1960’ which describes the car sales in Quebec, Canada for the years between 1960 to 1968. The variables are the number of cars sold for every month from January 1960 to December 1968. There are a total of 108 observations accounting for every month for 8 years. As we know Canada has peak winters from November to February and summers start from April/May, we see that most of the buyers tend to purchase the cars in summers as cold winter snow doesn't allow the public to wander around.

# Problem Statement

Automobile demand fluctuates throughout the year, it would be very beneficial for the car manufactures to be able to forecast their sales in advance. This will help car companies to manufacture cars based on the forecasted sales and plan the sales showroom logistics. This analysis and forecast help the business with planning and execution. This study aims to forecast car sales in Quebec, Canada for the next 10 years. Quebec has cold and long winters, and the weather is a possible parameter that could have a direct impact on car sales. We will consider all possible parameters and patterns to find the best model in this study.

# Data Acquisition

Kaggle offers a huge repository of datasets for the online community of data scientists and machine learning practitioners. We acquired our dataset “Monthly Car Sales in Quebec 1960” from Kaggle.

# Objective

Our main objective is to forecast car sales for the next 5 to 10 years based on seasonal conditions. The success of our project would be forecasting the car sales as accurately as possible for the next 5 to 10 years which will help the business have an approximate manufacturing target and planning the number of sales personnel and the hours of operation of the showroom depending on the sales forecast.

# Literature Review

This project is essential to the economy because of how influential car sales is in many people’s lives. This is especially important for the automobile industry as they can use these predictions to prevent the excess usage of raw materials. For example, in an article about the recent trends in new vehicle sales, IBIS World discusses the key factors that influence car sales are limited car production which drives the car prices up, and one of the recurring issues that arose in car sales was “limited car production" (*IBISWorld - Industry Market Research, Reports, and Statistics*, n.d.). Because of how detrimental it would be for the supply to not match the demand, especially since car companies would be losing millions if they are unable to sell as many cars, predicting trends in car sales in the future would increase profits while giving buyers more opportunities to meet their needs.

Supply and demand influence the price of the commodity of cars, and the price of the commodities in the market is pivotal to overall trade in the economy. As such, it is impossible to simply guess the trends without analyzing previous trends in car sales, especially when increasing the supply of cars at times that demand is little would only lead to a monetary loss. Thus, by using past factors that have influenced the economy, this research will accurately predict car sales in the following months, allowing companies to maximize their earnings.  For example, an article about the Canadian and US vehicle sales clearly states that “pandemic factors are likely to continue to affect auto sales in the months ahead” (*Canadian and US Vehicle Sales (December 2021): Auto News Flash*, n.d.). Forecasting car sales becomes very crucial for businesses because of the fluctuations in sales.

After finalizing our problem statement and understating the business objective, we did a preliminary exploratory analysis of the data. The EDA revealed seasons and trends in the series. Before we start building different models, we wanted to have a baseline model which will be helpful to compare the performances of the models used to make sure the seasons and trends are being captured. As Nair suggests to “pick a simple algorithm when building a baseline” (Nair, 2022) we decided to go with Naive which uses the “previous time step (t-1) to predict the expected outcome at the next time step (t+1)” (Brownlee, 2016) which makes it simple and easy to interpret. We build naive, seasonal naive, regression model with seasonality and trend as predictors, exponential smoothing, moving average, and ARIMA models with different components and compared the results with the baseline model and the other models to decide on a final model with the best performance to make the forecast for the next ten years.

# Exploratory Data Analysis

The objective of exploratory data analysis (EDA) is to obtain a preliminary understanding of the time series and to determine the existence of any seasonality or trend in the data set.

Initial exploratory data analysis revealed trend and seasonal components in our time series. Figure 1 shows the time series plot for the Car Sales data set. As shown in Figure 1 the dataset includes car sales from January1960 to December 1968 and there are no external predictors available for this study. The time series plot also shows that the number of sales made ranges from 6,000 to 26,000. The plot also reveals a larger spike followed by a smaller spike in sales for every year.

**Figure1**

*Car Sales Time Series Plot*

Graphical user interface

Description automatically generated with medium confidence

Figure 2 is the decomposition of the additive time series and illustrates the existence of seasonality and trend.  There is a strong upward trend and also seasonal pattern of rise and fall in data values that repeats after regular intervals.

**Figure 2**

*Decomposition of additive time series*

Diagram

Description automatically generated

The ACF plot showed that there is strong positive autocorrelation at lags 12, 24, and 36 in monthly data which will reflect in an annual seasonality. It means values during a given month each year are positively correlated. The PACF shows that there is a seasonal lag at 12. The AR 1 coefficient is 0.7320 which is closer to one but the autocorrelation with the differenced series showed that the bars are not inside the threshold margin, which is closer to zero, which infers that the series is not random walk. Figure 3 shows the ACF and PACF plot for the first 5 seasons.

**Figure 3**

*ACF and PACF plot before any difference*

Chart, histogram

Description automatically generated

# Data Wrangling and Pre-Processing

The month column which represents the month of the sales from 1960 to 1968 were as character in the dataset. We changed it to the time series month format starting from 1960, 1 to 1968, 12. There are no missing values and data is ready to be fit in a model.

# Data Splitting

To find the best model, the data set was split into training and validation sets. The validation set was used to compare the model performances. The dataset has the number of car sales for 9 years. The first eight years from 1960 to 1967 were used to train our model and the number of car sales of the year of 1968 was used as the validation set to validate the models based on the performance metric. Figure 4 shows the partition between train and validation sets. After concluding on the final optimal model, the train and validation sets will be combined together to make a forecast for the next ten years. This helps in having a longer time period of the data to be trained, and more importantly have the latest data which is in the validation set to be made available to make the forecast.

**Figure 4**

*Train and Validation Set*

*Chart, line chart

Description automatically generated*

# Model Strategies

We will test different data mining models to find the best fitting model. Naive model was chosen to be the baseline model. Since the time series plots showed a strong seasonality, seasonal naive was also used along with the Regression model with season and trend as predictors, moving average, exponential smoothing, neural network, and Arima models. The moving average and the regression model performed much better among all the preliminary models, so an ARIMA with different values for different components were used. Seasonal ARIMA performed the best as anticipated because of the strong seasonality to forecast the car sales in Quebec for the next ten years. These components are based on seasonal behavior through ACF and PACF plots.

# Evaluation Metrics

We will validate our models with the performance metrics AIC scores, RMSE, and MAPE. We used AIC scores to choose the optimal ARIMA model with different values for different components. The AIC score is the measure of the goodness-of-fit of the model, which also penalizes the model for overfitting the data. AIC scores are used to compare the performance of the models used. This score helped us determine that the model fits the data optimally by not over or under fitting. A lower AIC gives a better goodness-of-fit and a decreased tendency to over-fitting. The mean squared error is the squared distance between the actual and the predicted values. The error is squared to cancel the negative terms which makes the MSE a squared unit of the output. This is fixed by taking the root of the mean square error which brings the output to the same unit as the output. The main disadvantage of the RMSE is that it is not robust to outliers since it considers all the error values. The mean absolute percentage error (MAPE) measures the accuracy of a forecast as a percentage of the average absolute error. The most commonly used measure of the forecast error is MAPE because it is scaled as a percentage of the variable units which can be easily interpreted. Regression analysis uses MAPE as a loss function for model evaluation. Since percentage errors are calculated in terms of absolute values of errors, this works well with positive and negative values in the errors.  Our dataset does not have any outliers. After confirming that the dataset is clean without any outliers the MAPE, AIC score and the RMSE was chosen as the performance metrics.

# Model Evaluation

It is necessary to examine differences in data when we have seasonality. Seasonality usually causes the series to be non-stationary because the average value at some particular time within the seasonal span may be different from the average values at other times. The seasonal ARIMA model incorporates both non-seasonal and seasonal factors in the multiplicative model. The shorthand notation is ARIMA(p,d,q) (P,D,Q) which p (P) is the number of autoregressive (AR) terms in the model, q (Q) is the number of moving average (MA) terms, and d (D) is the number of integrated terms (or the number of times the series is differenced before an ARIMA model is applied). We used ACF and PACF plots to get the best seasonal and non-seasonal components. We took 1 seasonal and 1 non-seasonal difference to bring the lag spikes under control. ACF and PACF plots after differencing are shown in Figure 5. PACF plot shows p and P values, and ACF plot shows q and Q component for the model.

**Figure 5**

*ACF and PACF plot after 1 difference*

Chart, histogram

Description automatically generated

As discussed in model strategy, we ran different models through the train set and evaluated the results with the validation set. Each model was assessed based on the performance metrics such as AIC, RMSE, and MAPE values. Naive model was used as a baseline model to compare the performance of the other models. Table 1.1 shows the RMSE, and MAPE of all the models, and the AIC scores of the ARIMA model.

**Table 1.1**:

*Models Performance Metrics*

|  |  |  |  |
| --- | --- | --- | --- |
| Models | RMSE(Train) | RMSE(Validation) | MAPE |
| Naive | 3249 | 5865 | 22.27 |
| Seasonal Naive | 1983 | 2290 | 10.83 |
| Regression Season Trend | 1342 | 1723 | 8.66 |
| Moving Average | 448 | 3424 | 15.99 |
| Exponential Smoothing | 3394 | 5942 | 22.09 |
| Auto Arima(2,0,0)x(0,1,1)   (AIC=1480) | 1395 | 1452 | 6.85 |
| Arima (2,1,2)x(1,1,1)   (AIC Score = 1468) | 1381 | 1797 | 7.28 |
| Arima (2,1,3)x(1,1,1)  (AIC Score = 1470) | 1381 | 1796 | 7.29 |
| Arima (2,1,4)x(1,1,1)   (AIC Score = 1453) | 1210 | 2317 | 10.38 |
| Arima (2,1,5)x(1,1,1)   (AIC Score = 1455) | 1199 | 1796 | 7.29 |
| Arima (2,1,6)x(1,1,1)   (AIC Score = 1455) | 1187 | 2237 | 9.91 |
| Neural Network | 1534 | 1974 | 7.99 |

# Conclusion

AIC, RMSE, and MAPE values indicate that an Arima model with (2,1,5) x(1,1,1) components should be employed. In addition to low RMSE, AIC, and MAPE, this model has the closest RMSE in train and validation set. The residuals analysis shown in Figure 6 emphasize few below points:

* Histogram shows normal distribution of the residuals.
* ACF plot shows presence of white noise.
* QQ plot shows normality with 1 outlier at the head.
* TS plots do not show any trend except an intervention point.

**Figure 6**

*Residuals Analysis for Final Arima Model*

Diagram, schematic

Description automatically generated

The business objective is to forecast car sales in Quebec, Canada for next 10 years. The prediction sales are shown in Figure 7 below.

**Figure 7**

*Forecast car sales in Quebec for next 10 years*

A picture containing line chart

Description automatically generated

It is important to consider external factors to optimize the forecast. For a higher prediction performance, more advanced methods of feature selection are necessary.

Considering external factors like global oil prices, local oil prices, interest rates, and the economic trend, collecting data from more cities to analyze the trend and season across cities, and collecting more data about the cars being sold will help us get a better picture of the market. These are all some of the recommendations which will improve the performance and the scope of the project for the automobile industry. With more data about the customers, we can study the buying pattern of them to send marketing campaigns to increase the sales in the automotive industry.  Car sales typically grow with the economy as a whole, durable goods like vehicles are purchased when consumers feel comfortable with their financial status. By studying the trend in the forecasted car sales, the economic growth trend can also be forecasted and analyzed.

**Github link:**

**https://github.com/MahaJayapal/ADS506FinalProject**

# References

1. *4.2 Identifying Seasonal Models and R Code | STAT 510*. (n.d.). PennState: Statistics Online Courses. <https://online.stat.psu.edu/stat510/lesson/4/4.2>
2. *Brownlee, J. (2016) Naive Bayes for Machine Learning.* <https://machinelearningmastery.com/naive-bayes-for-machine-learning/>
3. *Canadian and US Vehicle Sales (December 2021): Auto News Flash*. (n.d.). Post. <https://www.scotiabank.com/ca/en/about/economics/economics-publications/post.other-publications.autos.auto-news-flash.january-5--2022.html>
4. *IBISWorld - Industry Market Research, Reports, and Statistics*. (n.d.). <https://www.ibisworld.com/ca/bed/new-vehicle-sales/15016/>
5. *Monthly Car Sales in Quebec 1960*. (2020, October 27). Kaggle. <https://www.kaggle.com/datasets/dinirimameev/monthly-car-sales-in-quebec-1960>
6. Nair, A. (2022, April 9). *Baseline Models: Your Guide For Model Building - Towards Data Science*. Medium. <https://towardsdatascience.com/baseline-models-your-guide-for-model-building-1ec3aa244b8d>
7. ‘PennState’ Eberly College of Science [<https://online.stat.psu.edu/stat510/lesson/4/4.2>]
8. Shmueli, G., & Jr, K. L. C. (2016). *Practical Time Series Forecasting with R: A Hands-On Guide [2nd Edition] (Practical Analytics)* (2nd ed.). Axelrod Schnall Publishers.
9. Singh, B. P. (2021, December 13). *Predicting Sales: Time Series Analysis & Forecasting with Python*. Medium. <https://medium.com/analytics-vidhya/predicting-sales-time-series-analysis-forecasting-with-python-b81d3e8ff03f>
10. Soares, L. (2022, November 29). *Sales Forecasting: from Traditional Time Series to Modern Deep Learning*. Medium. <https://towardsdatascience.com/sales-forecasting-from-time-series-to-deep-learning-5d115514bfac>

**Appendix A**. R Mark Down

**Forecasting Automotive Demand In Quebec**

Mahalakshmi Jayapal and Mohammad Mahmoudighaznavi

Shiley-Marcos School of Engineering, University of San Diego

ADS 506: Applied Time Series Analysis

Professor Mikael Graindorge

December 02, 2022

**Data Set Link**

Data obtained from: https://www.kaggle.com/code/kerneler/starter-monthly-car-sales-in-quebec-6083d2b7-1/data

#load necessary packages  
set.seed(506)  
library(fpp2)

## ✔ ggplot2 3.4.0 ✔ fma 2.4   
## ✔ forecast 8.18 ✔ expsmooth 2.3

library(readr)  
library(forecast)  
library(zoo)

#Import data set  
  
car\_sales <- read\_csv("monthly-car-sales.csv")

## Rows: 108 Columns: 2

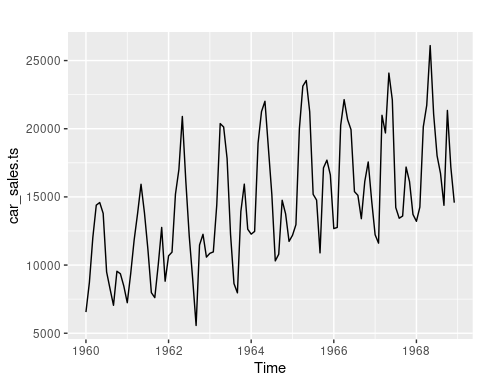
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (1): Month  
## dbl (1): Monthly car sales in Quebec 1960-1968

head(car\_sales)

## # A tibble: 6 × 2  
## Month `Monthly car sales in Quebec 1960-1968`  
## <chr> <dbl>  
## 1 1960-01 6550  
## 2 1960-02 8728  
## 3 1960-03 12026  
## 4 1960-04 14395  
## 5 1960-05 14587  
## 6 1960-06 13791

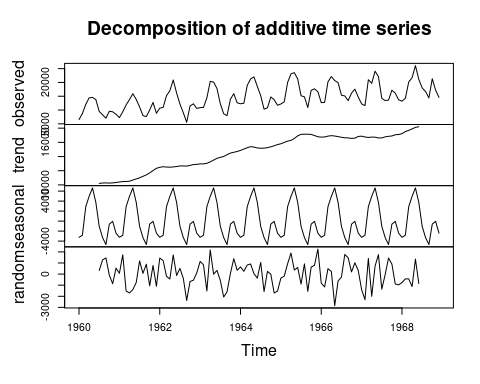
**EDA**

#change the column name and remove spaces  
colnames(car\_sales) <- c("Month", "Car\_Sales")  
  
#convert to time series  
car\_sales.ts <- ts(car\_sales$Car\_Sales, start=c(1960,1),frequency=12)  
autoplot(car\_sales.ts)



**Decomposition**

#check if there are trend and seasonality in our time series  
#decompose  
components = decompose(car\_sales.ts)  
plot(components)



There is a strong upward trend and also seasonal pattern is rise and fall in data values that repeats after regular intervals.

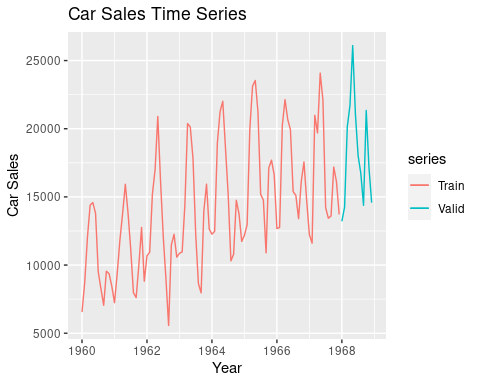
**Data Partitioning**

The full year of 1968 is chosen as validation period.

#partition data to train and validation  
#consider last full year for validation set  
ValidLength <- 12  
TrainLength <- length(car\_sales.ts) - ValidLength  
Train <- window(car\_sales.ts,end = c(1960, TrainLength))  
Valid <- window(car\_sales.ts, start = c(1960, TrainLength + 1), end = c(1980, TrainLength + ValidLength))

## Warning in window.default(x, ...): 'end' value not changed

autoplot(Train, series="Train")+  
 autolayer(Valid, series="Valid")+  
 labs(title="Car Sales Time Series",  
 x="Year", y="Car Sales")



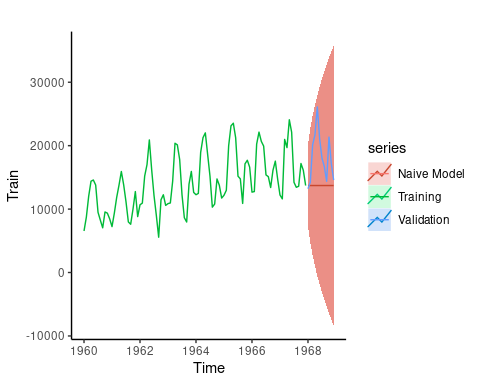
**Model Training**

1. **Naive model:**

# naive model  
n\_model <- naive(Train, h = 12, level = 95)  
  
  
#forecast validation set   
n\_model\_forecast <- forecast(n\_model, ValidLength)  
accuracy(n\_model\_forecast, Valid)

## ME RMSE MAE MPE MAPE MASE ACF1  
## Training set 75.400 3249.349 2553.084 -2.080684 18.87198 1.654675 0.1936749  
## Test set 4515.167 5865.374 4599.000 21.634224 22.26884 2.980649 0.3754679  
## Theil's U  
## Training set NA  
## Test set 1.382695

#plot  
autoplot(Train, series="Training") +  
 autolayer(n\_model\_forecast, series="Naive Model") +  
 autolayer(Valid, series = "Validation")+  
 theme\_classic()

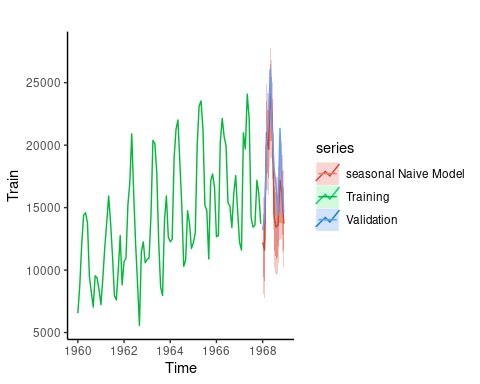


b) **Seasonal naive model:**

#we add seasonal to our naive model  
#seasonal naive  
sn\_model <- snaive(Train, h = 12)  
  
  
#forecast validation set   
sn\_model\_forecast <- forecast(sn\_model, h=12)  
accuracy(sn\_model\_forecast, Valid)

## ME RMSE MAE MPE MAPE MASE ACF1  
## Training set 913.5238 1983.602 1542.952 5.985723 10.68091 1.000000 0.2692255  
## Test set 1646.8333 2290.827 1959.500 9.318081 10.83242 1.269968 -0.3867255  
## Theil's U  
## Training set NA  
## Test set 0.6054476

#plot  
autoplot(Train, series="Training") +  
 autolayer(sn\_model\_forecast, series="seasonal Naive Model") +  
 autolayer(Valid, series = "Validation")+  
 theme\_classic()



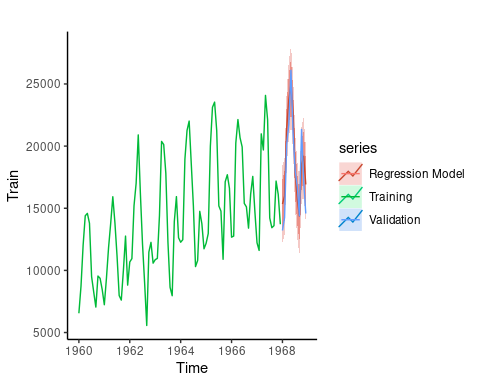
Comparing RMSE and MAPE between naive and seasonal naive shows that seasonal naive performs much better due to existence of seasonality.

1. **Regression season trend model:**

#regression season trend model  
  
st\_model <- tslm(Train ~ trend + season)  
  
  
#forecast validation set  
st\_forecast <- forecast(st\_model, h = 12)  
accuracy(st\_forecast, Valid)

## ME RMSE MAE MPE MAPE MASE  
## Training set 5.684342e-14 1341.994 1089.861 -0.9026902 8.242856 0.7063476  
## Test set -6.702500e+02 1723.445 1521.698 -4.5827769 8.663157 0.9862248  
## ACF1 Theil's U  
## Training set 0.29507763 NA  
## Test set 0.02707458 0.4298715

#plot  
autoplot(Train, series="Training") +  
 autolayer(st\_forecast, series="Regression Model") +  
 autolayer(Valid, series = "Validation")+  
 theme\_classic()



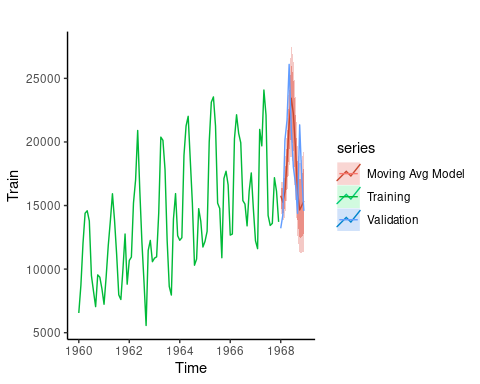
We have a better RMSE and MAPE compared to seasonal naive model.

1. **moving average:**

#moving average  
car\_sales.ma <- rollmean(Train, k = 4, align = "right")  
  
  
#forecast validation set  
ma\_forecast <- forecast(car\_sales.ma, h=12)  
accuracy(ma\_forecast, Valid)

## ME RMSE MAE MPE MAPE MASE  
## Training set 29.44936 448.4691 334.7317 0.1421066 2.406855 0.2767316  
## Test set 456.10392 3424.2491 3022.9496 0.1952128 15.995655 2.4991533  
## ACF1 Theil's U  
## Training set 0.2838495 NA  
## Test set 0.2893178 0.8845582

#plot  
autoplot(Train, series="Training") +  
 autolayer(ma\_forecast, series="Moving Avg Model") +  
 autolayer(Valid, series = "Validation")+  
 theme\_classic()



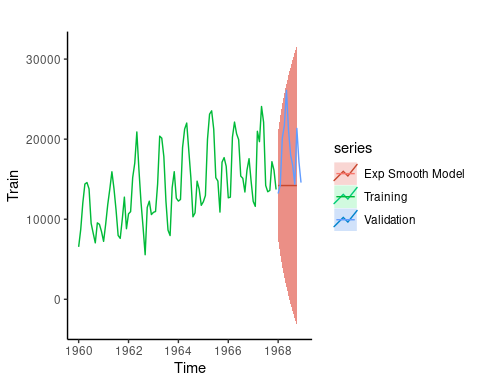
Moving average is not a good fit for our data set.

1. **Exponential smooth:**

#exponential smooth  
exp\_smooth.model <- ses(Train, alpha = .8, level = c(.95))  
  
#forecast  
exp\_smooth.pred <- forecast(exp\_smooth.model, h = 10)  
accuracy(exp\_smooth.pred, Valid)

## ME RMSE MAE MPE MAPE MASE ACF1  
## Training set 92.0655 3393.987 2750.609 -2.843922 20.56584 1.782692 0.3303156  
## Test set 4489.3840 5942.207 4689.127 20.579473 22.09153 3.039062 0.3910382  
## Theil's U  
## Training set NA  
## Test set 1.346528

#plot  
autoplot(Train, series="Training") +  
 autolayer(exp\_smooth.pred, series="Exp Smooth Model") +  
 autolayer(Valid, series = "Validation")+  
 theme\_classic()



Same as moving average model, exponential smooth make the RMSE and MAPE number larger.

1. **Arima model**

* **Auto Arima**

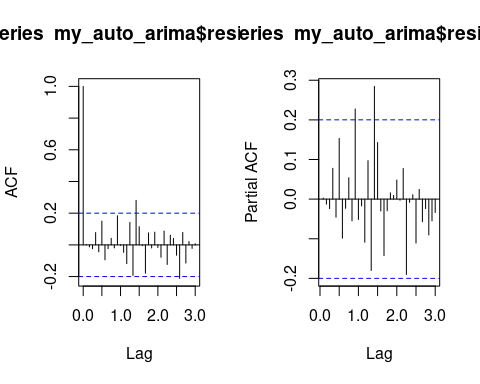
#auto-arima  
my\_auto\_arima <- auto.arima(Train)  
  
my\_auto\_arima

## Series: Train   
## ARIMA(2,0,0)(0,1,1)[12] with drift   
##   
## Coefficients:  
## ar1 ar2 sma1 drift  
## 0.2662 0.2011 -0.4911 82.5901  
## s.e. 0.1079 0.1116 0.1522 15.1559  
##   
## sigma^2 = 2335120: log likelihood = -734.77  
## AIC=1479.53 AICc=1480.3 BIC=1491.69

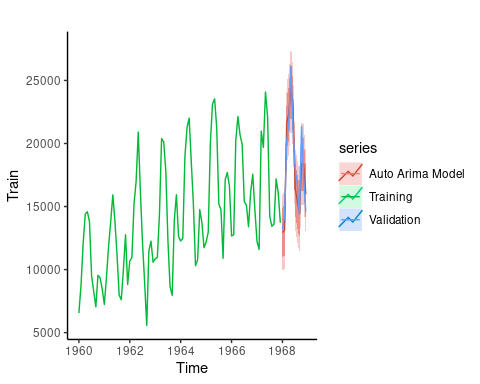
#forecast  
auto\_arima.forecast <- forecast(my\_auto\_arima, h = 12)  
accuracy(auto\_arima.forecast, Valid)

## ME RMSE MAE MPE MAPE MASE  
## Training set 12.43502 1394.967 1033.690 -0.8854028 7.493405 0.6699427  
## Test set 182.51209 1452.313 1277.397 0.8347746 6.849333 0.8278915  
## ACF1 Theil's U  
## Training set 0.002648354 NA  
## Test set -0.318310632 0.3874412

#Residuals ACF and PACF  
par(mfrow=c(1,2))  
acf(my\_auto\_arima$residuals, lag.max = 36)  
pacf(my\_auto\_arima$residuals, lag.max = 36)



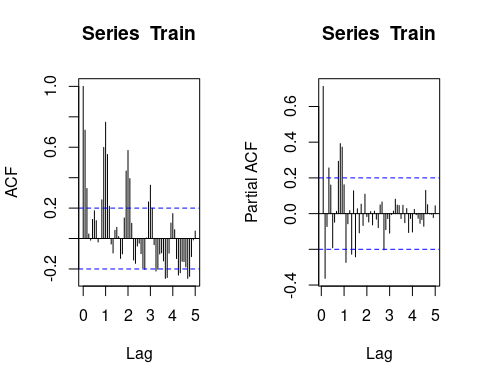
#plot  
autoplot(Train, series="Training") +  
 autolayer(auto\_arima.forecast, series="Auto Arima Model") +  
 autolayer(Valid, series = "Validation")+  
 theme\_classic()



Auto arima model works much better compared to previous models. Auto arima components are (2,0,0)(0,1,1). still there are some significant spikes due to ACF and PACF plots.

EDA for finding Arima seasonal and non-seasonal components:

#Train ACF and PACF plot  
par(mfrow=c(1,2))  
acf(Train,lag.max = 60)  
pacf(Train, lag.max = 60)

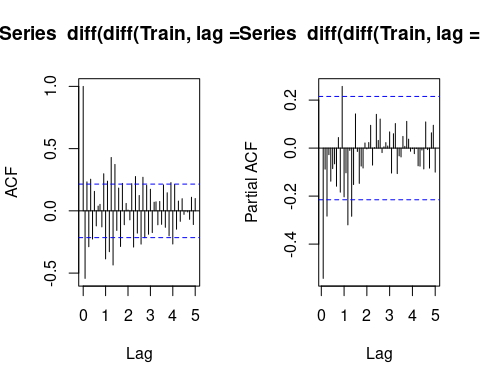


There is strong positive autocorrelation at lags 12, 24, and 36 in monthly data which will reflect an annual seasonality. It means values during a given month each year are positively correlated. The PACF shows that there is a seasonal lag at 12.

The seasonal ARIMA model incorporates both non-seasonal and seasonal factors in multiplicative model.

lets find out best (p, d, q)x(P, D, Q) values. first we take a seasonal and non seasonal difference of our train data set, because we have significant spikes in our lags.

#seasonal difference  
par(mfrow=c(1,2))  
acf(diff(diff(Train, lag=12)), lag.max = 60)  
pacf(diff(diff(Train, lag=12)), lag.max = 60)



We took 1 seasonal and 1 non-seasonal difference. so the d and D are equal to 1. p is AR component, and we take a look at PACF plot to find a seasonal P component. we have just 1 seasonal spike. so, P is 1. for non-seasonal we look at the spikes before first season in PACF plot, so, p is 2. For q or MA part we look at ACF plot. for seasonal we have 1 spike, and for non-seasonal we start with 2 spikes.

So, lets start with (2 , 1, 2)x(1 , 1, 1)

* **Arima (2 , 1, 2)x(1 , 1, 1)**

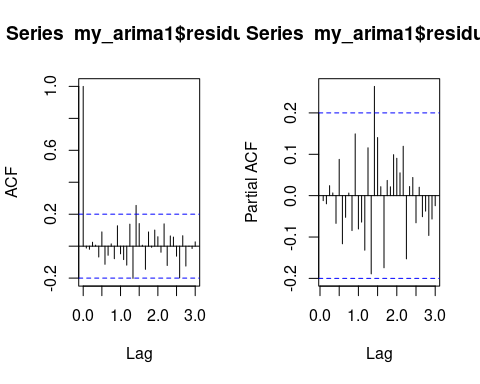
my\_arima1 <- arima(Train, order = c(2 , 1, 2), seasonal = c(1 , 1, 1))  
my\_arima1

##   
## Call:  
## arima(x = Train, order = c(2, 1, 2), seasonal = c(1, 1, 1))  
##   
## Coefficients:  
## ar1 ar2 ma1 ma2 sar1 sma1  
## -0.4296 0.2760 -0.2776 -0.5653 0.2265 -0.6965  
## s.e. 0.2598 0.1293 0.2530 0.2102 0.4712 0.4918  
##   
## sigma^2 estimated as 2204311: log likelihood = -726.9, aic = 1467.8

sqrt(mean(my\_arima1$residuals^2))

## [1] 1380.516

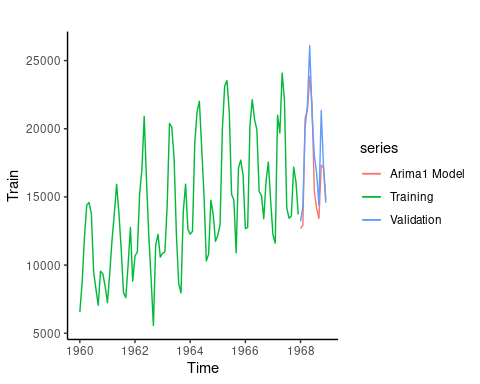
#ACF & PACF  
  
par(mfrow=c(1,2))  
acf(my\_arima1$residuals, lag.max = 36)  
pacf(my\_arima1$residuals, lag.max = 36)



#forecast  
arima1.forecast <- predict(my\_arima1 , n.ahead=12)$pred  
accuracy(arima1.forecast, Valid)

## ME RMSE MAE MPE MAPE ACF1 Theil's U  
## Test set 1087.445 1796.633 1355.318 5.900013 7.284794 -0.2065803 0.4847213

#plot  
autoplot(Train, series="Training") +  
 autolayer(arima1.forecast, series="Arima1 Model") +  
 autolayer(Valid, series = "Validation")+  
 theme\_classic()



we will increase the MA component by 1 point.

* **Arima (2 , 1, 3)x(1 , 1, 1)**

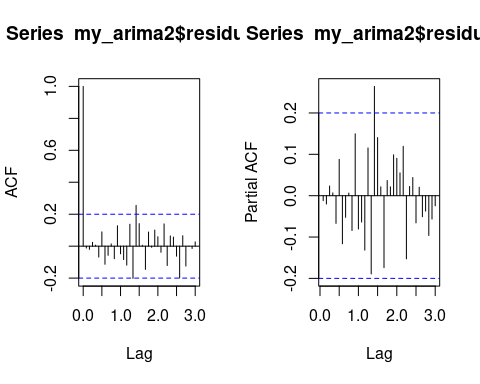
my\_arima2 <- arima(Train, order = c(2 , 1, 3), seasonal = c(1 , 1, 1))  
my\_arima2

##   
## Call:  
## arima(x = Train, order = c(2, 1, 3), seasonal = c(1, 1, 1))  
##   
## Coefficients:  
## ar1 ar2 ma1 ma2 ma3 sar1 sma1  
## -0.4261 0.2789 -0.2813 -0.5651 0.0026 0.2258 -0.6957  
## s.e. 0.5654 0.4430 0.5960 0.2108 0.3899 0.4718 0.4923  
##   
## sigma^2 estimated as 2204555: log likelihood = -726.9, aic = 1469.8

sqrt(mean(my\_arima2$residuals^2))

## [1] 1380.592

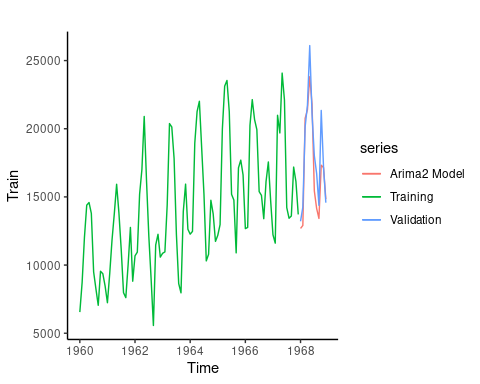
#ACF & PACF  
  
par(mfrow=c(1,2))  
acf(my\_arima2$residuals, lag.max = 36)  
pacf(my\_arima2$residuals, lag.max = 36)



#forecast  
arima2.forecast <- predict(my\_arima2 , n.ahead=12)$pred  
accuracy(arima2.forecast, Valid)

## ME RMSE MAE MPE MAPE ACF1 Theil's U  
## Test set 1087.152 1796.444 1355.344 5.89875 7.285325 -0.20708 0.4846878

#plot  
autoplot(Train, series="Training") +  
 autolayer(arima2.forecast, series="Arima2 Model") +  
 autolayer(Valid, series = "Validation")+  
 theme\_classic()



We increase MA component to 6 based on the ACF plot.

* **Arima (2 , 1, 4)x(1 , 1, 1)**

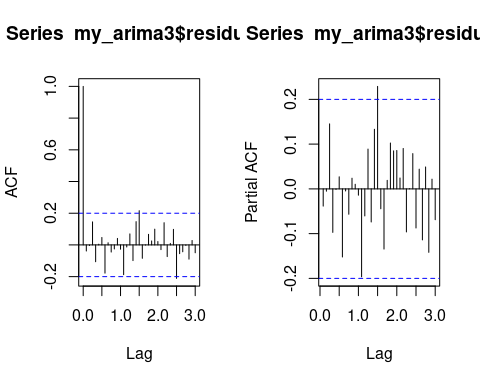
my\_arima3 <- arima(Train, order = c(2 , 1, 4), seasonal = c(1 , 1, 1))  
my\_arima3

##   
## Call:  
## arima(x = Train, order = c(2, 1, 4), seasonal = c(1, 1, 1))  
##   
## Coefficients:  
## ar1 ar2 ma1 ma2 ma3 ma4 sar1 sma1  
## -1.9589 -0.9994 1.6223 -0.1546 -1.2974 -0.4923 0.0082 -0.3131  
## s.e. 0.0021 0.0012 0.0723 0.1438 0.1463 0.0702 0.9037 0.9308  
##   
## sigma^2 estimated as 1692825: log likelihood = -717.53, aic = 1453.05

sqrt(mean(my\_arima3$residuals^2))

## [1] 1209.794

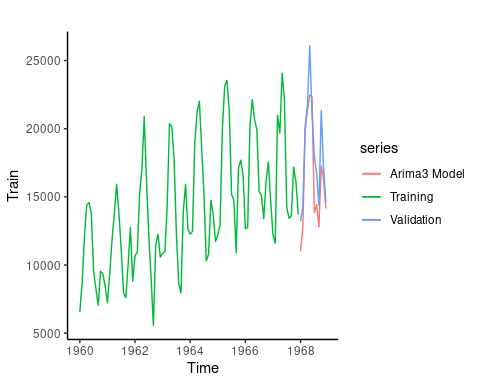
#ACF & PACF  
  
par(mfrow=c(1,2))  
acf(my\_arima3$residuals, lag.max = 36)  
pacf(my\_arima3$residuals, lag.max = 36)



#forecast  
arima3.forecast <- predict(my\_arima3 , n.ahead=12)$pred  
accuracy(arima3.forecast, Valid)

## ME RMSE MAE MPE MAPE ACF1 Theil's U  
## Test set 1670.313 2316.878 1879.247 9.386459 10.37742 -0.4295988 0.5609271

#plot  
autoplot(Train, series="Training") +  
 autolayer(arima3.forecast, series="Arima3 Model") +  
 autolayer(Valid, series = "Validation")+  
 theme\_classic()



* **Arima (2 , 1, 5)x(1 , 1, 1)**

my\_arima4 <- arima(Train, order = c(2 , 1, 5), seasonal = c(1 , 1, 1))

## Warning in log(s2): NaNs produced  
  
## Warning in log(s2): NaNs produced

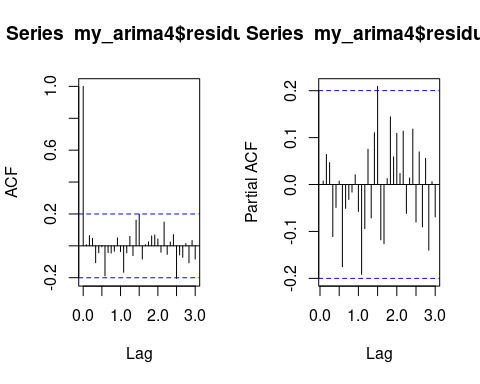
my\_arima4

##   
## Call:  
## arima(x = Train, order = c(2, 1, 5), seasonal = c(1, 1, 1))  
##   
## Coefficients:  
## ar1 ar2 ma1 ma2 ma3 ma4 ma5 sar1  
## -1.9591 -0.9999 1.6255 -0.2237 -1.2959 -0.2537 0.1679 -0.1524  
## s.e. 0.0014 0.0006 0.0481 0.1849 0.2941 0.1967 0.0479 0.6264  
## sma1  
## -0.1916  
## s.e. 0.6347  
##   
## sigma^2 estimated as 1662289: log likelihood = -717.31, aic = 1454.61

sqrt(mean(my\_arima4$residuals^2))

## [1] 1198.833

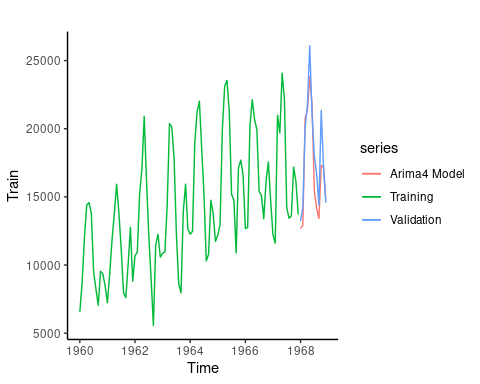
#ACF & PACF  
  
par(mfrow=c(1,2))  
acf(my\_arima4$residuals, lag.max = 36)  
pacf(my\_arima4$residuals, lag.max = 36)



#forecast  
arima4.forecast <- predict(my\_arima2 , n.ahead=12)$pred  
accuracy(arima4.forecast, Valid)

## ME RMSE MAE MPE MAPE ACF1 Theil's U  
## Test set 1087.152 1796.444 1355.344 5.89875 7.285325 -0.20708 0.4846878

#plot  
autoplot(Train, series="Training") +  
 autolayer(arima4.forecast, series="Arima4 Model") +  
 autolayer(Valid, series = "Validation")+  
 theme\_classic()



* **Arima (2 , 1, 6)x(1 , 1, 1)**

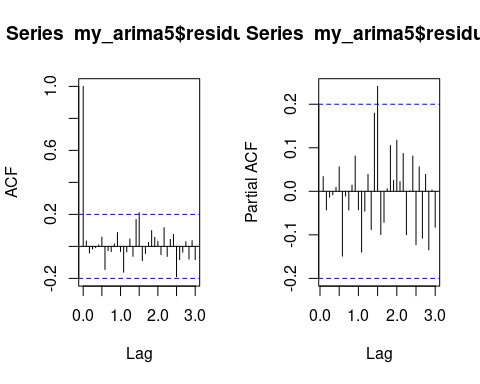
my\_arima5 <- arima(Train, order = c(2 , 1, 6), seasonal = c(1 , 1, 1))  
my\_arima5

##   
## Call:  
## arima(x = Train, order = c(2, 1, 6), seasonal = c(1, 1, 1))  
##   
## Coefficients:  
## ar1 ar2 ma1 ma2 ma3 ma4 ma5 ma6  
## -1.9593 -0.9996 1.5590 -0.2236 -1.0902 -0.2744 -0.1803 -0.1903  
## s.e. 0.0022 0.0018 0.0806 0.2037 0.1972 0.1744 0.2398 0.1044  
## sar1 sma1  
## -0.0510 -0.2810  
## s.e. 1.5494 1.5802  
##   
## sigma^2 estimated as 1630158: log likelihood = -716.31, aic = 1454.61

sqrt(mean(my\_arima5$residuals^2))

## [1] 1187.19

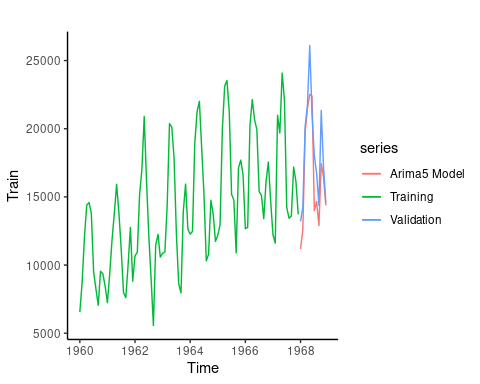
#ACF & PACF  
  
par(mfrow=c(1,2))  
acf(my\_arima5$residuals, lag.max = 36)  
pacf(my\_arima5$residuals, lag.max = 36)



#forecast  
arima5.forecast <- predict(my\_arima5 , n.ahead=12)$pred  
accuracy(arima5.forecast, Valid)

## ME RMSE MAE MPE MAPE ACF1 Theil's U  
## Test set 1581.781 2237.186 1801.125 8.869264 9.909599 -0.4617185 0.5447552

#plot  
autoplot(Train, series="Training") +  
 autolayer(arima5.forecast, series="Arima5 Model") +  
 autolayer(Valid, series = "Validation")+  
 theme\_classic()



**AIC comparison:**

sort.score <- function(x, score = "aic"){  
 if (score == "aic"){  
 x[with(x, order(AIC)),]  
   
 } else {  
 warning('score = "x" only accepts valid arguments "aic" ')  
 }  
}  
  
sc.AIC=AIC(my\_arima1, my\_arima2, my\_arima3, my\_arima4, my\_arima5, my\_auto\_arima)

## Warning in AIC.default(my\_arima1, my\_arima2, my\_arima3, my\_arima4, my\_arima5, :  
## models are not all fitted to the same number of observations

sort.score(sc.AIC, score = "aic")

## df AIC  
## my\_arima3 9 1453.055  
## my\_arima5 11 1454.612  
## my\_arima4 10 1454.615  
## my\_arima1 7 1467.799  
## my\_arima2 8 1469.799  
## my\_auto\_arima 5 1479.532

Best AIC belongs to my\_arima3, my\_arima4, and my\_arima5 models. Since, the RMSE for validation set in my\_arima4 is much better than my\_arima3 and my\_arima 5, we consider it as our best Arima model.

Residual analysis for my\_arima4 model:

#residual analysis  
residual.analysis <- function(model, std = TRUE){  
 #install.packages("TSA")  
 library(TSA)  
   
 if (std == TRUE){  
 res.model = rstandard(model)  
 }else{  
 res.model = residuals(model)  
 }  
 par(mfrow=c(3,2))  
 plot(res.model,type='o',ylab='Standardized residuals', main="Time series plot of standardized residuals")  
 abline(h=0)  
 hist(res.model,main="Histogram of standardized residuals")  
 qqnorm(res.model,main="QQ plot of standardized residuals")  
 qqline(res.model, col = 2)  
 acf(res.model,main="ACF of standardized residuals")  
 print(shapiro.test(res.model))  
 k=0  
   
 autoplot(res.model, lag.max = length(model$residuals)-1 , StartLag = k + 1, k = 0, SquaredQ = FALSE)  
}  
  
residual.analysis(model=my\_arima4)

## Registered S3 methods overwritten by 'TSA':  
## method from   
## fitted.Arima forecast  
## plot.Arima forecast

##   
## Attaching package: 'TSA'

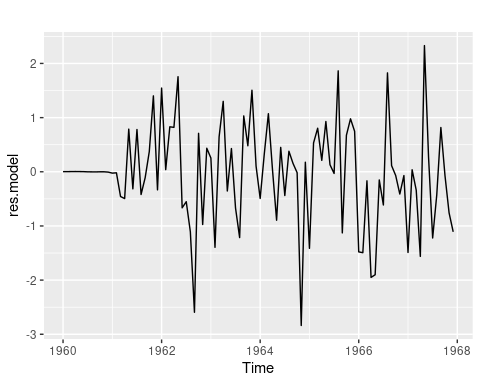
## The following object is masked from 'package:readr':  
##   
## spec

## The following objects are masked from 'package:stats':  
##   
## acf, arima

## The following object is masked from 'package:utils':  
##   
## tar

##   
## Shapiro-Wilk normality test  
##   
## data: res.model  
## W = 0.97636, p-value = 0.08003

## Warning in ggplot2::geom\_line(na.rm = TRUE, ...): Ignoring unknown parameters:  
## `lag.max`, `StartLag`, `k`, and `SquaredQ`



Diagram

Description automatically generated

1. Histogram shows normal distribution of the residuals.
2. ACF plot shows presence of white noise.
3. QQ plot shows normality with 1 outlier at the head.
4. TS plots do not show any trend except an intervention point.
5. **Neural Network**

my\_NN <- nnetar(Train, p=2, P=1, size=2)  
my\_NN

## Series: Train   
## Model: NNAR(2,1,2)[12]   
## Call: nnetar(y = Train, p = 2, P = 1, size = 2)  
##   
## Average of 20 networks, each of which is  
## a 3-2-1 network with 11 weights  
## options were - linear output units   
##   
## sigma^2 estimated as 2386430

#forecast  
nn.pred <- forecast(my\_NN, h=12)

accuracy(nn.pred, Valid)

## ME RMSE MAE MPE MAPE MASE  
## Training set -0.4952791 1544.807 1266.210 -1.337601 9.055072 0.8206407  
## Test set 1269.8089143 1991.989 1560.449 6.757815 8.272307 1.0113395  
## ACF1 Theil's U  
## Training set 0.1033829 NA  
## Test set -0.4492262 0.5124074

#plot  
autoplot(Train, series="Training") +  
 autolayer(nn.pred, series="Neural Network") +  
 autolayer(Valid, series = "Validation")+  
 theme\_classic()

Chart

Description automatically generated

My\_arima4 model perform better than neural network. So, we consider it as our best model.

**Forecast next 10 years car sales**

#combine train and validation  
my\_final\_model <- arima(car\_sales.ts, order = c(2 , 1, 5 ), seasonal =c(1 , 1, 1))  
  
#forecast next 10 years  
forecast\_10 <- predict(my\_final\_model , n.ahead=120)$pred  
forecast\_10

## Jan Feb Mar Apr May Jun Jul Aug  
## 1969 14752.90 15796.12 20818.27 24135.93 25907.09 22131.84 19187.95 16298.79  
## 1970 16402.48 16165.27 22811.77 24946.88 26164.13 24128.54 19333.98 17025.97  
## 1971 16700.76 17788.21 23821.32 25356.99 27794.40 24786.84 19984.64 18603.39  
## 1972 18048.59 18916.40 24264.23 26843.78 28604.45 25435.58 21471.68 19193.33  
## 1973 19242.41 19424.69 25585.62 27800.70 29229.31 26824.04 22220.99 20007.82  
## 1974 19833.16 20609.13 26629.46 28449.04 30505.21 27694.63 22995.17 21291.30  
## 1975 20911.24 21701.09 27318.45 29628.13 31457.89 28459.88 24211.84 22123.44  
## 1976 22022.53 22439.81 28416.47 30634.45 32233.89 29611.82 25114.80 22976.84  
## 1977 22812.47 23474.33 29452.96 31433.71 33327.81 30565.04 25961.10 24091.85  
## 1978 23800.22 24523.50 30281.23 32479.19 34313.75 31416.95 27037.71 25016.18  
## Sep Oct Nov Dec  
## 1969 14852.66 20957.67 18235.18 16684.02  
## 1970 16384.77 20611.23 19771.16 18176.57  
## 1971 16749.90 21524.08 21239.16 18551.19  
## 1972 17641.59 22880.00 21744.14 19705.95  
## 1973 18984.24 23494.93 22766.69 20925.20  
## 1974 19722.59 24424.77 23981.51 21601.91  
## 1975 20601.97 25613.85 24738.37 22602.37  
## 1976 21756.23 26439.47 25686.16 23712.97  
## 1977 22637.92 27354.94 26783.35 24550.58  
## 1978 23536.95 28431.74 27662.19 25506.30

#plot  
autoplot(forecast\_10)+  
 theme\_classic()

