CONESTOGA
Connect Life and Learning

CONESTOGA COLLEGE, DOON CAMPUS
PREDICTIVE ANALYTICS (1498)

PROG8041-SEC1-STATISTICAL FORECASTING II

GROUP PROJECT-II: BORDER TRUCK TRAFFIC FORECASTING

DATE: 08/16/2024

GROUP:3

GROUP MEMBERS:

PRASHANT KUMAR DUA – 886378

RUCHITA KUKADIYA - 8905405

SURAJ KUMAR MISHRA - 8902749

GUIDED BY:

PROF. JONATHAN PLUMBTREE

Data

The Border Crossing Data from Canada-US:

The data used in this project pertains to border crossings between Canada and the United States. It was collected by the Eastern Border Transportation Coalition (EBTC), a non-profit organization consisting of representatives from transportation agencies in states and provinces along the eastern border, including Michigan.

Data Discussion:

The dataset was imported from the Government Website in a .csv file. Initial exploration of the dataset revealed that it contains the following columns:

- **Port Name:** Name of the port of two countries
- **State:** State name
- **Port Code:** Specific code of the port of two countries
- **Border:** Border name; Canada-US and Mexico-US
- **Date:** Date of the data collected
- **Measure:** Mode of transportation; bus, train, truck
- **Value:** Number of vehicles passed
- **Latitude:** The north-south position of a point
- **Longitude:** The east-west position of a point
- **Point:** A specific location on the Earth's surface

Data Quality Checks:

Before proceeding with the analysis, the dataset was checked for null values by using **sum(is.na(US_Canada))** and duplicates by using **sum(duplicated(US_Canada)).** It was confirmed that there are no missing values or duplicate entries in the dataset. This ensures the reliability and accuracy of the subsequent analysis.

Data Cleaning and Preprocessing:

The **mutate()** is a fundamental function in the R toolkit for data manipulation, enabling efficient and straightforward transformations within data frames and **tsibble**. This is particularly useful for handling irregular time series data and ensures consistency in subsequent analysis. We use **mutate(Date = yearmonth(Date))** to convert the date index into year-month format. We get a total of 341 observations representing 341 months of data of a number of trucks coming from Canada and entering the US through the Michigan border.

Data Aggregation:

Data aggregation is a crucial step in data preprocessing, particularly for large datasets or time series data. It simplifies analysis, reduces noise, and highlights key patterns by summarizing detailed or granular data into higher-level summaries. In this dataset, we

use **aggregate()** function for conversion to a total number of trucks for each state irrespective of the port of entry.

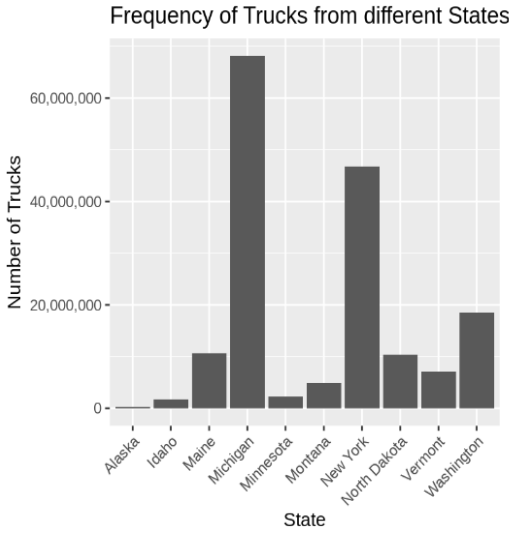
Problem Statement:

The goal of this forecasting project is to develop a robust and practical model for predicting the future number of trucks crossing the Canada-US border. By analyzing historical crossing data, the model aims to provide accurate forecasts that can help anticipate future trends and accommodate the increasing volume of trade between the two countries. This will enable transportation agencies and policymakers to plan effectively for infrastructure and resource needs, ensuring that facilities and personnel are adequately prepared to handle future demands.

Data Visualization

Data Distribution of Number of Trucks

Using '**ggplot**' to visualize the frequency of trucks per state. The x-axis represents the states, and the y-axis shows the number of trucks.

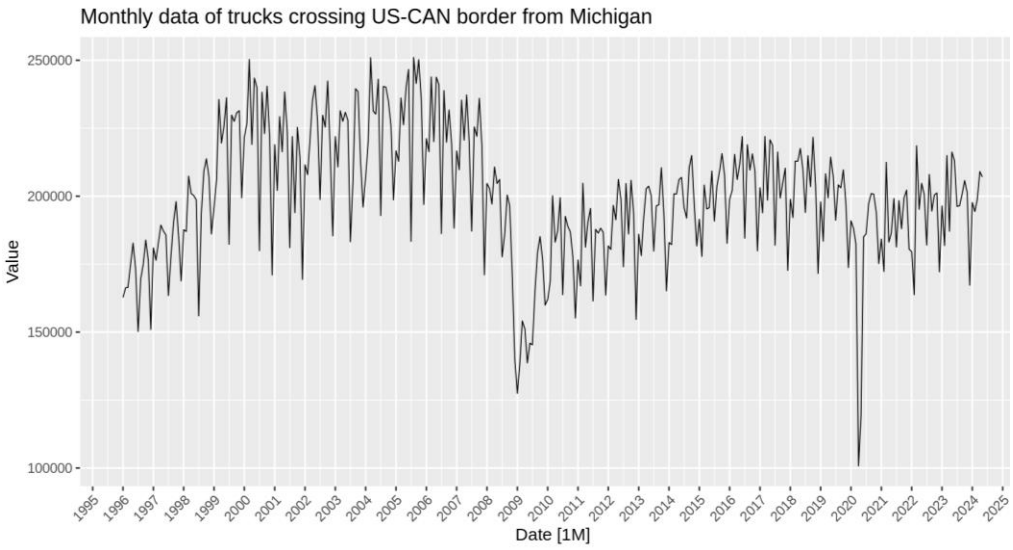


By the plot we can draw the following insights as mentioned below -

- **Michigan** seems to have the highest movement of trucks across the border and hence we wish to study the trend and accordingly forecast the traffic for **the next 6 months**.
- **Detroit** handles the **highest volume** of truck traffic with considerable variability, **Port Huron** has **moderate traffic** with less variability, and **Sault Sainte Marie** has the **lowest**.

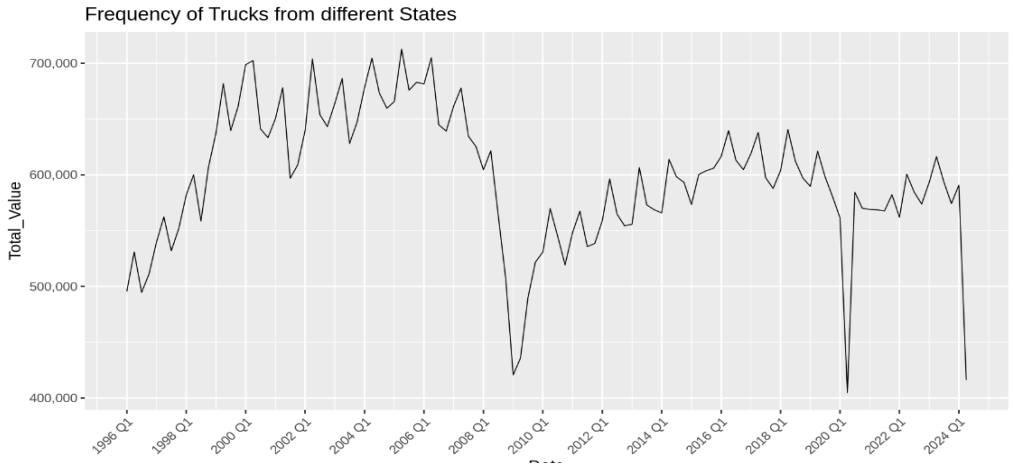
Time-Series Plot (Before Transformation):

A time plot of trucks crossing the US-CAN border from Michigan over the given time period generated. The plot illustrates the trend and variability in border crossing.

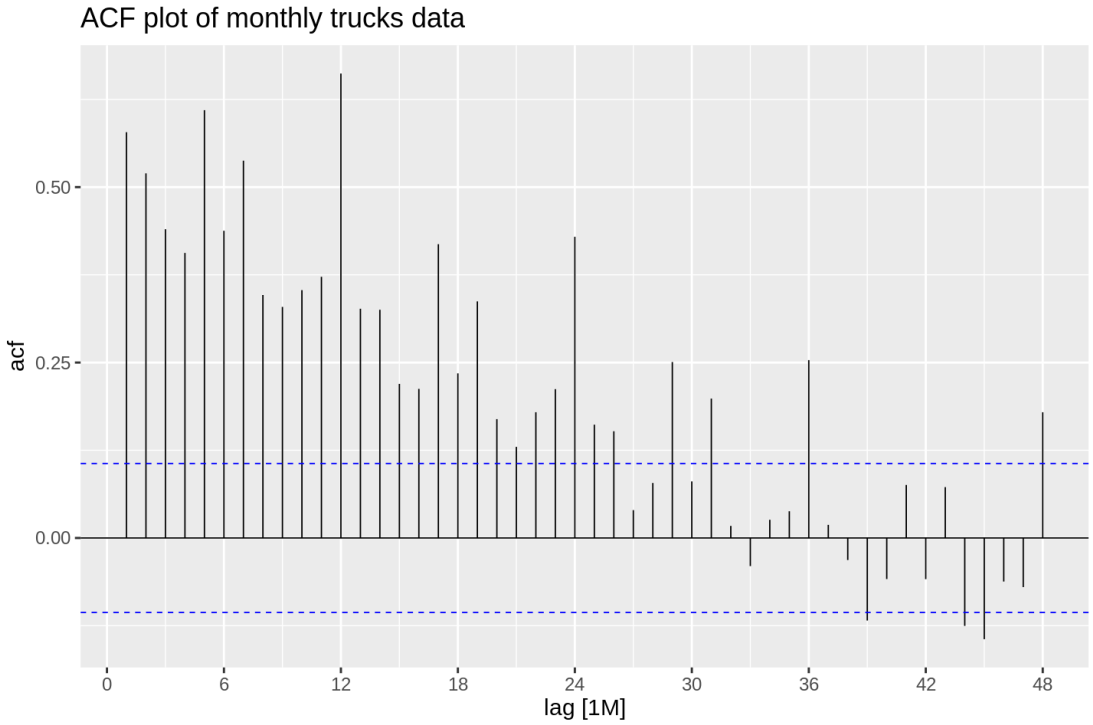


- Data has rising trends within periods **1996 Jan till Dec 2008** and from Dec 2009 till Dec 2019.
- There is an evident **seasonality** after every 6 months as we can see a dip after **every 6 months** in truck data across all the years
- There are two significant and **sharp dips** years **2009** and **2020** representing the Economic Depression and COVID-19 events that disrupted the movement of transportation across the US and Canada due to economic and legal restrictions.
- There is **high variability** in truck data from **1996 -2009** as compared to the period 2010-2024. This is evident from the height of spikes that is significantly higher in 1996-2009. Hence this shows unstable variance in data that we would need to correct in order to get a better visualization of trends and seasonality.
- There is **no stationarity** in data as there is a trend with seasonality

Quarter-wise frequency of trucks from different States



Autocorrelation Plots:



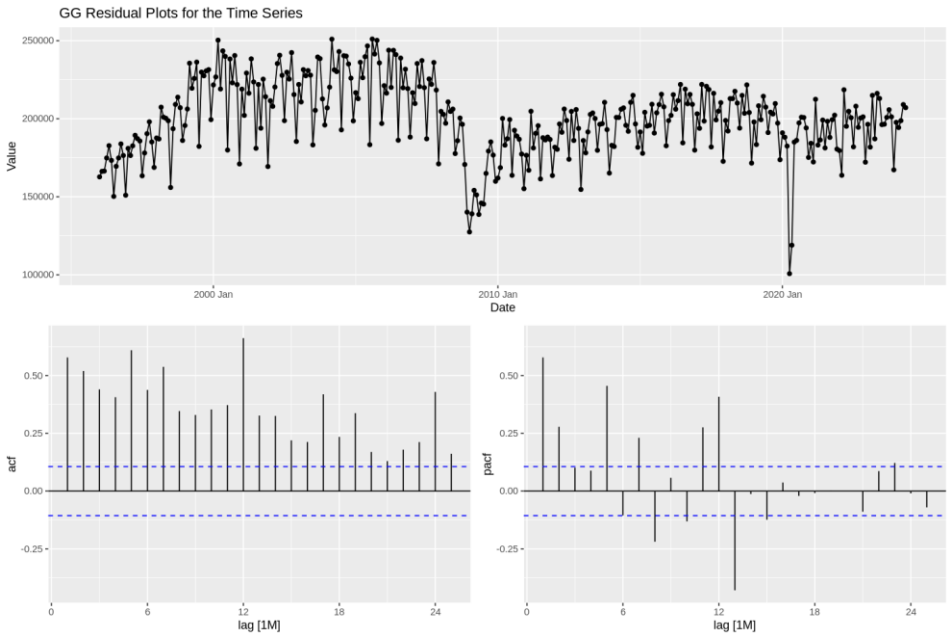
Interpretation:

The Autocorrelation Function (ACF) plot for monthly truck border crossing data reveals information about the time dependencies within the data. On the Y-axis, the autocorrelation coefficient ranges from -1 to 1, **with values closer to 1 indicating a strong positive correlation.** The X-axis represents the number of time steps (lags), where a lag of 1 indicates the correlation between two months. The blue dashed lines mark statistical significance, and the bars above these lines signify significant autocorrelations.

Interpreting the ACF plot, we observe clear positive autocorrelation which shows that the future value of a number of trucks crossing the US-CAN border is positively correlated with past values.

GS Display Plot:

gg_tsdisplay(). Plots a time series along with its ACF along with a graphic of either a PACF, histogram, lagged scatterplot, or spectral density.



- Data in the ACF plot without difference shows some signs of trends with less stationarity.
- Another insight we can see is that trucks number have dropped fairly after 2010 with highest figure of 250,000 around year 2005 while highest figure of 220,000 around year 2008.

Data Transformation

Box- Cox Transformations –

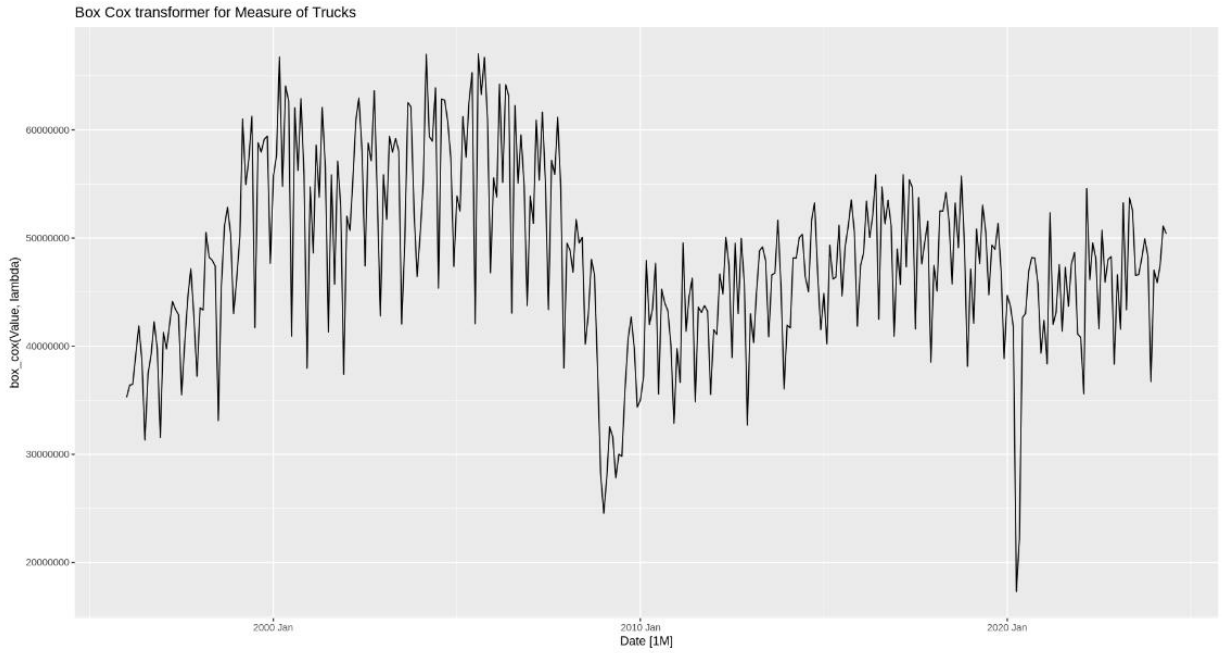
The Box-Cox transformation is a statistical technique used to stabilize variance and make the data more normally distributed. This transformation is particularly useful when dealing with time series data that exhibit heteroscedasticity (i.e., when the variance of the residuals is not constant) or non-normality.

Why Do We Use Box-Cox Transformation?

- **Stabilize Variance:** To reduce the impact of heteroscedasticity.
- **Normalize Data:** To make the data more closely follow a normal distribution.
- **Improve Model Accuracy:** To enhance the performance of statistical models that assume normality and constant variance.

$$y(\lambda) = \begin{cases} \frac{y^{\lambda}-1}{\lambda} & \lambda \neq 0 \\ \ln(y) & \lambda = 0 \end{cases}$$

- y is the original data.
- λ lambda is the transformation parameter.



The transformed plot represents the truck border crossing after applying the Box-Cox transformation. Here's a detailed analysis:

- **Stabilized Variance:** The transformed values start near 20000000 and increase to over 70000000. The transformation effectively stabilizes the variance, making the data smoother and less volatile.
- **Normalized Data:** The transformed data is not as normally distributed as the original data. We can see that Box-Cox transformations hardly had any effect variance in the values across the months.

- **Trend Clarity:** There appears to be a general upward trend from the start, followed by a downturn and fluctuations over time.

From Transformations, we can see that Box-Cox transformations hardly had any effect variance in the values across the months. So, now we calculate the log transformations.

Log Transformation:

Log transformation is a powerful technique used in time series analysis to stabilize the variance, normalize the distribution of the data, and make patterns more apparent. It is particularly useful when dealing with time series data that shows exponential growth or has a multiplicative seasonality component.

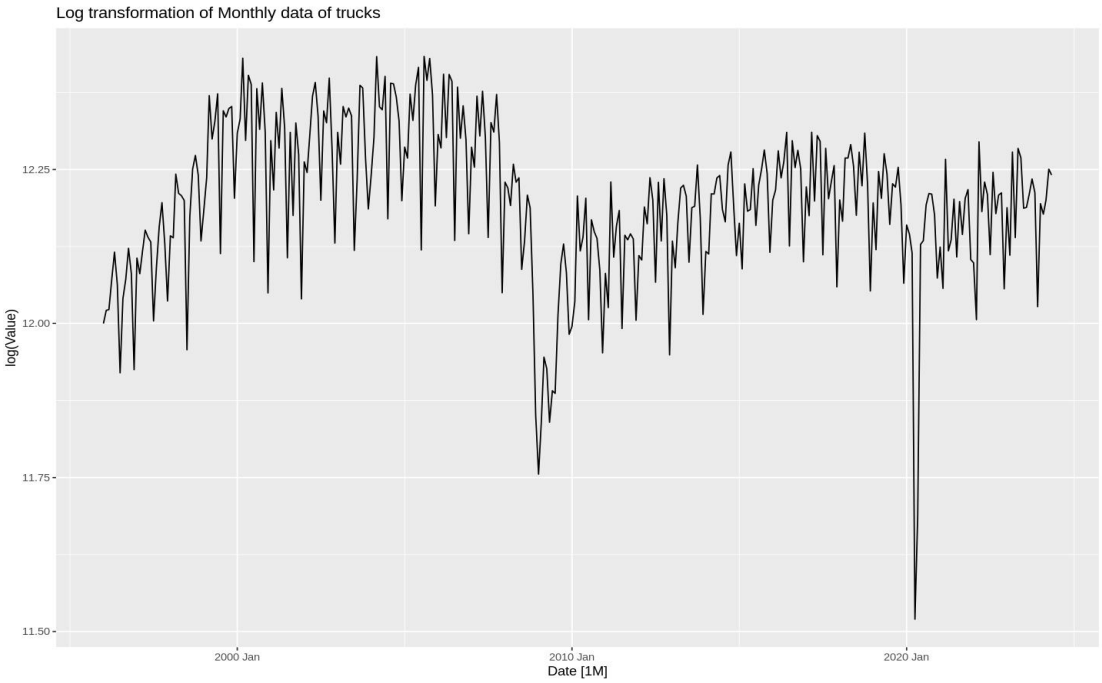
Why do we use log transformation?

- **Stabilize variance:** To reduce heteroscedasticity
- **Normalized Data:** Make positively skewed data more normally distributed.
- **Simplify Multiplicative Relationship:** Turn multiplicative relationships into additive ones.

$Y' = \log(Y)$

Where Y is the original time series data

Y' is the transformed data.



Interpretation:

From the log transformation, we get some significant improvement in capturing the data variance. The transformed plot represents the truck border crossing after applying the log transformation. Here's a detailed analysis:

- The transformed values start near 11.50 and increase to over 12.50. The transformation effectively stabilizes the variance, making the data smoother and less volatile.
- Although log transformations had shown some significant improvement in capturing the data variance over time and stabilized it to some extent. However, the data is still non-stationary.

Decomposition:

Seasonal Decomposition – STL (Seasonal and Trend decomposition using Loess) is a method used to decompose a time series into three components:

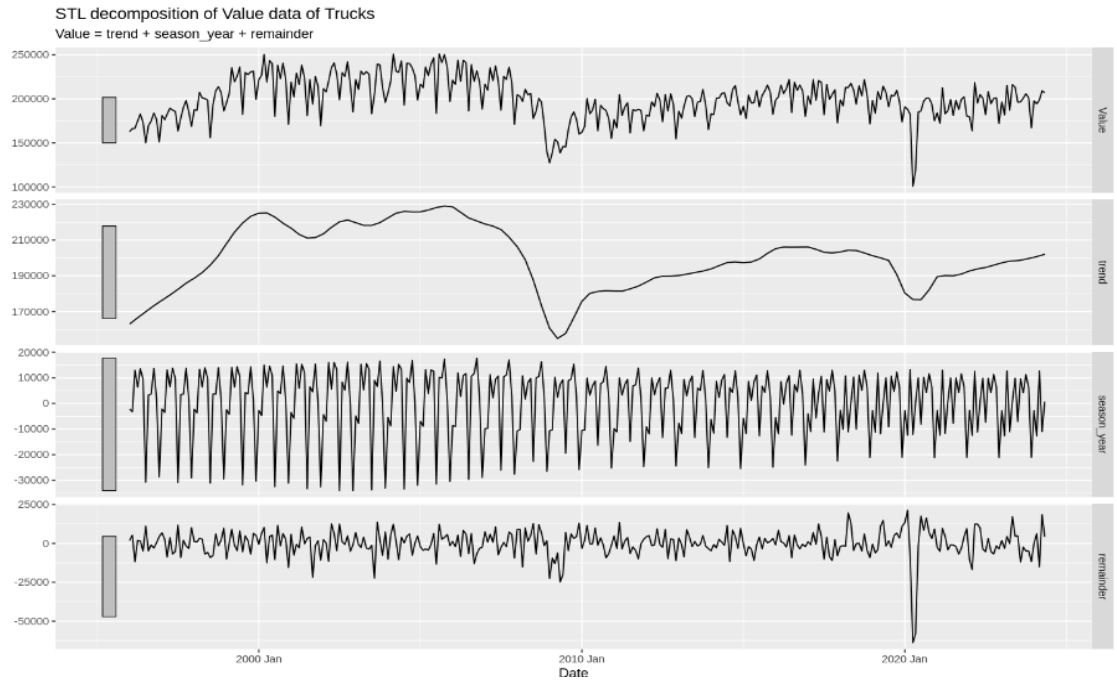
- **Trend**: Long-term movement or direction in the data.
- **Seasonal**: Repeating short-term cycles within the data.
- **Remainder**: Irregular fluctuations not captured by the trend or seasonal components.
- **Trend Analysis**: Understand the long-term direction of the data.
- **Seasonal Patterns**: Identifies recurring patterns.
- **Anomaly Detection**: Isolates irregular variations.
- **Forecasting**: Enhances accuracy by clarifying data components.

Interpretation of the STL Plot – The plot shows the STL decomposition of the value.

Top Panel: Original Time Series (Value) - The initial value is near 15000 and gradually increases. It peaks around Date Jan 2000 at approximately 25000, then declines below 15000 during Jan 2010 due to recession and again Starts to rise around 20000 but due to COVID-19 again it goes down around 10000.

Middle Panel: Trend - The trend starts near 17000 and rises gradually. It peaks at around 23000 date Jan 2000, showing a slight decline but staying above 19000 towards the end.

Bottom Panel: Remainder- The remainder fluctuates around 0, indicating irregular variations. There are noticeable deep around date 2020 Jan, with values below -50000.

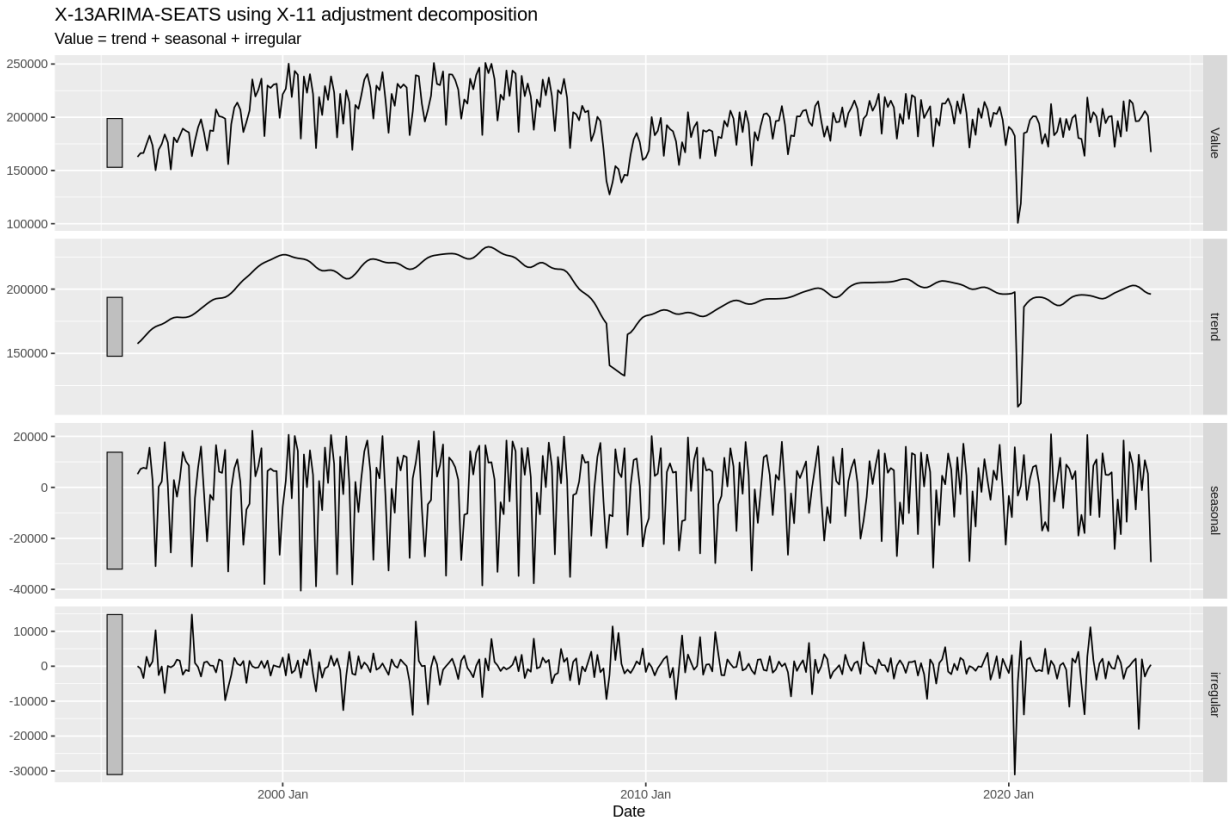


Summary - The trend analysis shows an upward movement with certain deep, peaking at around 250000 around Jan 2010, indicating long-term growth before a decline. The irregular variations exhibit high deep around Jan 2010 and Jan 2020, likely due to external factors affecting the value (World Recession and Covid-19).

XTL Decomposition:

X-13ARIMA-SEATS are advanced statistical methods developed by the U.S. Census Bureau for seasonal adjustment and decomposition of time series data. These methods are widely used in economic and financial analysis to separate different components of a time series, such as trend, seasonal effects, and irregular components.

- **Trend**: Long-term movement or direction in the data.
- **Seasonal**: Repeating short-term cycles within the data.
- **Irregular**: The residual component after removing the trend and seasonal effects.



Interpretation: The plot shows the STL decomposition of the value.

Top Panel: Original Time Series (Value)- The top panel displays raw time series data, including trend, seasonal effects, and random noise. It shows an overall increase from 1995 to 2008, followed by a significant dip in 2010 and another sharp drop in 2020.

Trend Component - The second panel shows a long-term trend in the data, with an upward movement until 2008, followed by a decline in 2010, and another dip in 2020, indicating persistent growth or decline patterns over time.

Seasonal Component- The third panel displays seasonal patterns, capturing predictable changes in data due to seasonal factors. These fluctuations are consistent in amplitude and frequency, indicating that the seasonal effect remains stable over time.

Irregular Component- The bottom panel displays the irregular component, which represents residuals or noise in data after removing trend and seasonal effects. It captures unexpected variations or anomalies, with larger spikes corresponding to sharp drops in the original series between 2010 and 2020.

Summary- The decomposition of a time series into its components, including the trend, seasonal component, and irregular component, helps in understanding the long-term direction, regular fluctuations, and potential outliers.

XTL Decomposition Components of Trucks Border crossing Value

.model	Date	Value	trend	seasonal	irregular	season_adjust
xtl	2023 Dec	167248	197359.1	-29860.25170	-250.8841	197108.3
xtl	2024 Jan	197681	197930.8	60.08857	-309.9355	197620.9
xtl	2024 Feb	194346	198707.0	-6065.86043	1704.8914	200411.9
xtl	2024 Mar	198762	199110.7	280.83453	-629.5343	198481.2
xtl	2024 Apr	209055	198885.9	7320.72338	2848.3690	201734.3
xtl	2024 May	207092	198103.8	11187.40478	-2199.1756	195904.6

Forecasting and Analysis

Train and Test

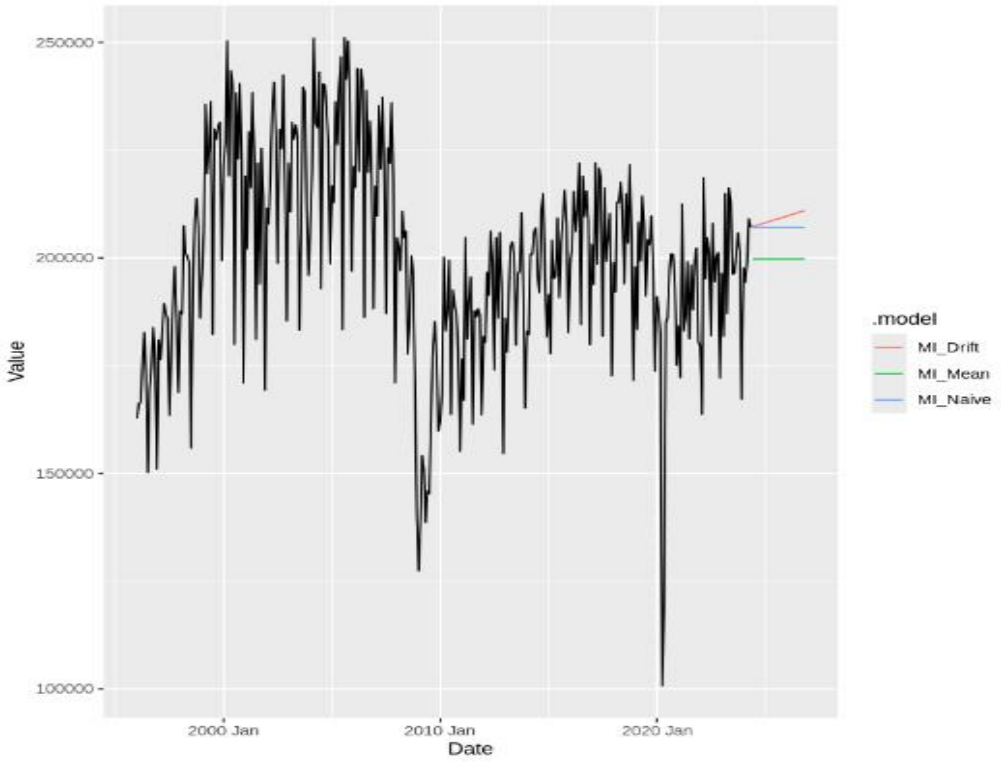
Filter data for training and testing sets - The data is divided by using **filter()** into training and testing sets based on the year. This division helps in training the models on historical data and validating their performance on future data.

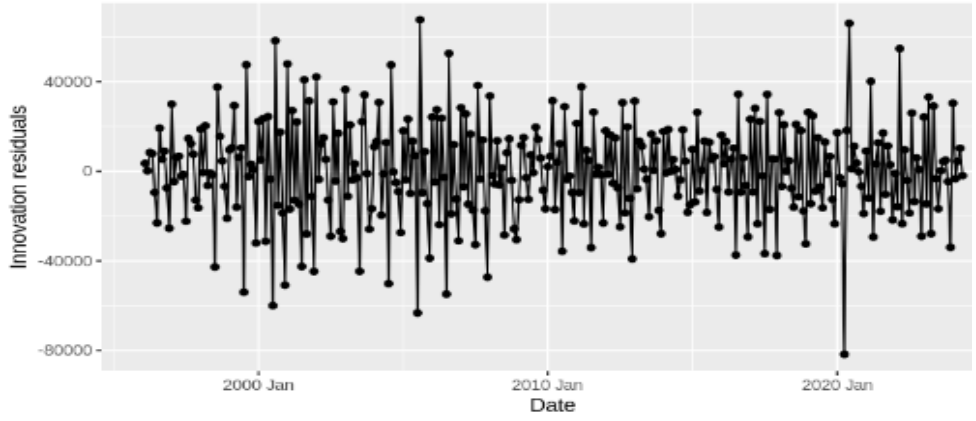
Fit the models -Various forecasting models are fitted to the training data to predict the value of the truck. The models used include:

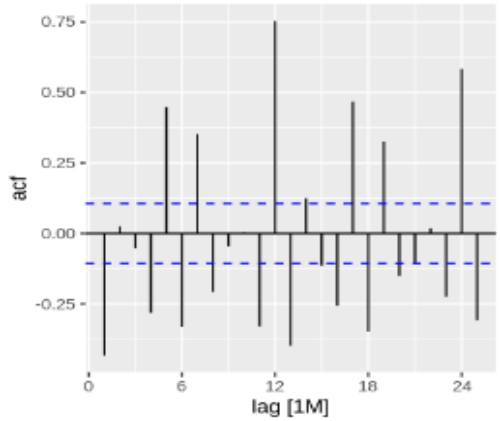
1. Moving Average Model: Predicts future values based on the average of past values, assuming that the best estimate of future prices is the mean of the historical data.

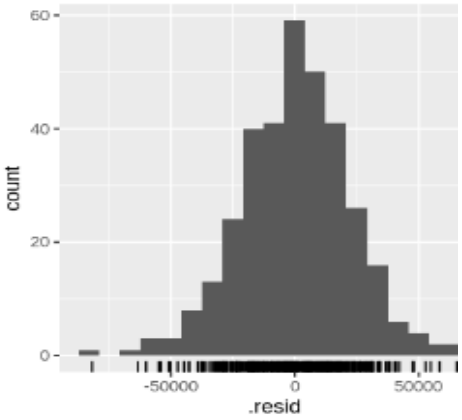
2. Naive Model: Assumes that the most recent observed value is the best predictor of the future value. For example, if the last value is 15700 trucks, the model will predict 15700 trucks for all future points.

3. Drift Model: A variant of the Naive model that incorporates a drift component, accounting for a constant change over time. It projects future values based on the average change in the data. For instance, if the price increases by an average of 1000 trucks per month, this model will add 1000 to the last observed value for each subsequent month.


250000
200000
150000
100000
Value
2000 Jan
2010 Jan
2020 Jan
Date
.model
MI_Drift
MI_Mean
MI_Naive


40000
0
-40000
-80000
Innovation residuals
2000 Jan
2010 Jan
2020 Jan
Date


0.75
0.50
0.25
0.00
-0.25
acf
0
6
12
18
24
lag [1M]


60
40
20
0
count
-50000
0
50000
.resid

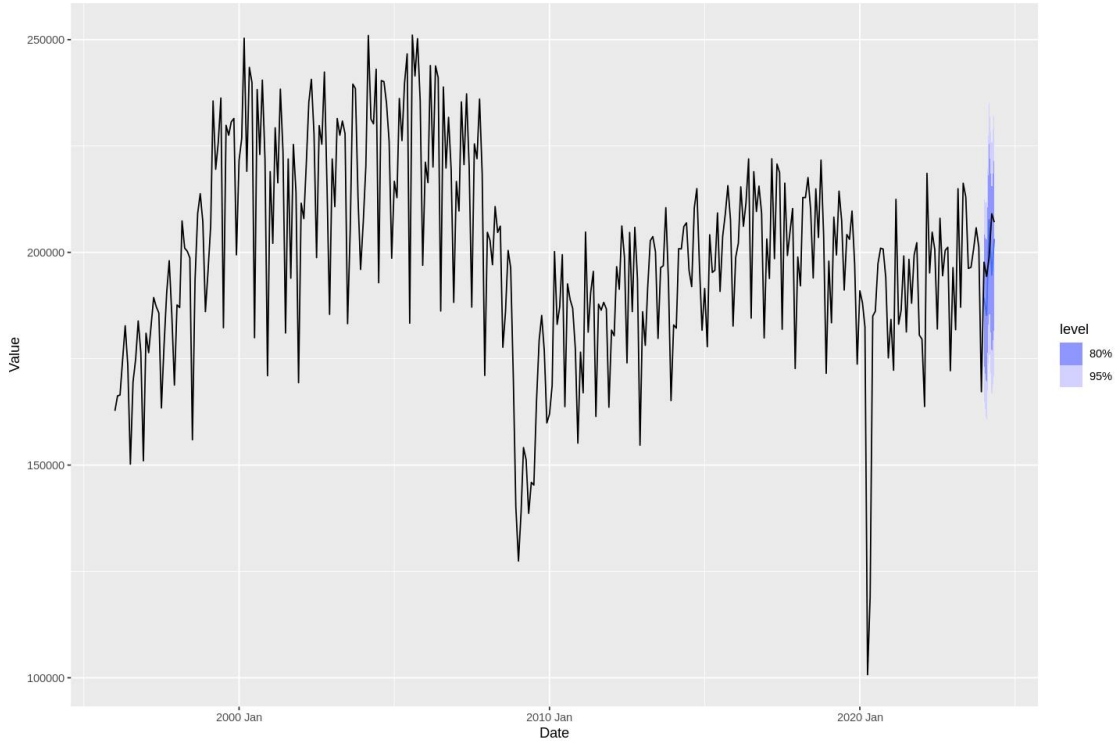
Exponential Smoothing:

Exponential smoothing is a popular technique used in time series analysis to produce smoothed forecasts. It applies weighted averages of past observations, with the weights decaying exponentially as the observations get older.

Key Concept:

- **Smoothing Parameter:**
  The smoothing parameter α ranging from 0 to 1, affects the model's response to recent changes, with high values indicating more responsiveness, and low values indicating slower response.

- **Exponential Weighting:**
  Unlike simple moving averages, exponential smoothing assigns exponentially decreasing weights to older data points. This means recent data has the most significant impact on the forecast, while the influence of older data diminishes exponentially.



Time Series Model Summary: ETS(A, Ad, A)

Smoothing Parameters:

- Alpha (α): 0.4308

- Beta (β): 0.0001000294
- Gamma (γ): 0.0001001801
- Damping Parameter (ϕ): 0.9709474

Model Selection Criteria:

- Akaike Information Criterion (AIC): 8281.540
- Corrected AIC (AICc): 8283.698
- Bayesian Information Criterion (BIC): 8350.248

Model Performance Metrics:

- Mean Squared Error (MSE): 91,303,809.34
- Root Mean Squared Error (RMSE): 9,555.30
- Mean Absolute Percentage Error (MAPE): 4.44%
- Mean Absolute Error (MAE): 8,955.88

Multiple Linear Regression:

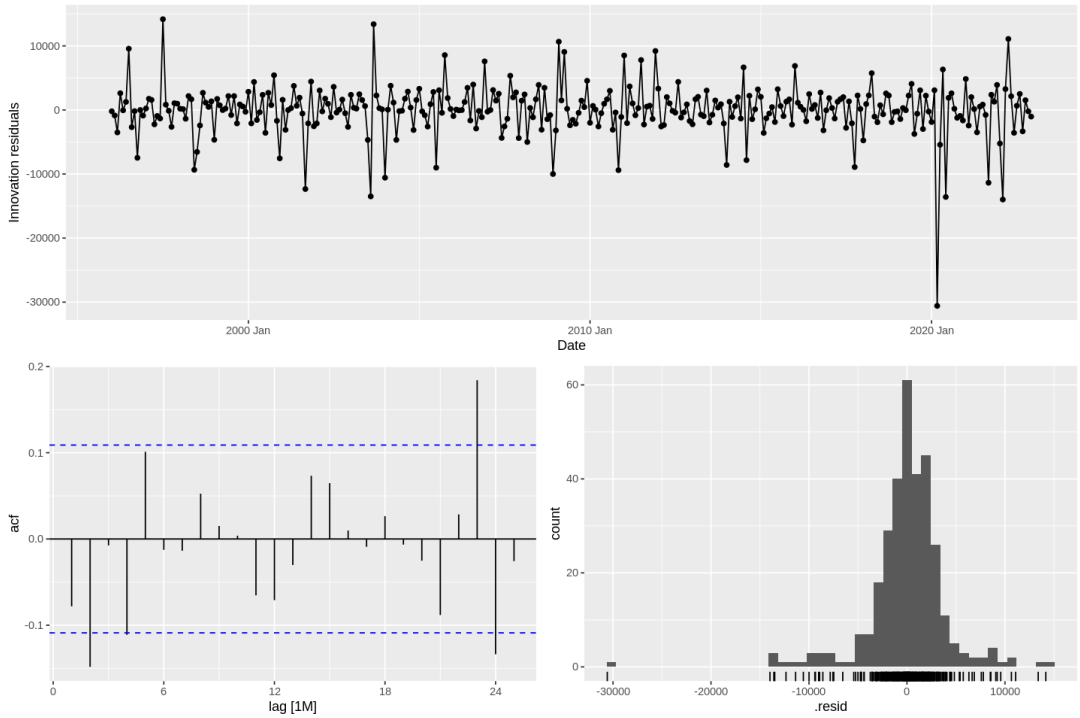
Multiple Regression in Time Series refers to the application of multiple linear regression techniques to time series data. In this context, the goal is to model the relationship between a dependent time series variable and multiple independent variables.

Model Formulation:

$$Yt=\beta 0+\beta 1X1t+\beta 2X2t+\cdots+\beta nXnt+\epsilon t$$

Where:

- Yt is the dependent time series variable at time t.
- X1t, X2t,…, Xnt are the independent variables (predictors) at time t.
- β0 is the intercept.
- β1,β2,…,βn are the coefficients, indicating the effect of each predictor on Yt.
- ϵt is the error term (residual) at time t.



Interpretation:

- In the residuals plot, the residuals should appear as a random scatter around zero with no obvious patterns. In this plot, we see some spikes, particularly around 2020.
- The autocorrelations should lie within the blue confidence bounds (indicating no significant autocorrelation). In this plot, some lags exceed the bounds, especially at lag 24, suggesting that there may be some autocorrelation left in the residuals.
- The residuals should ideally be **normally distributed**. This histogram shows a reasonably symmetric distribution with a peak near zero, which is a good sign.

Model Evaluation:

Residual standard error: 4045 on 321 degrees of freedom

Multiple R-squared: 0.973,

Adjusted R-squared: 0.9728

F-statistic: 5782 on 2 and 321 DF,

p-value: < 0.000000000000000222

ARIMA Model

In our group project, we employed the ARIMA (AutoRegressive Integrated Moving Average) model for forecasting the US-Canada trucks crossing the border data. The ARIMA model is structured with three key components:

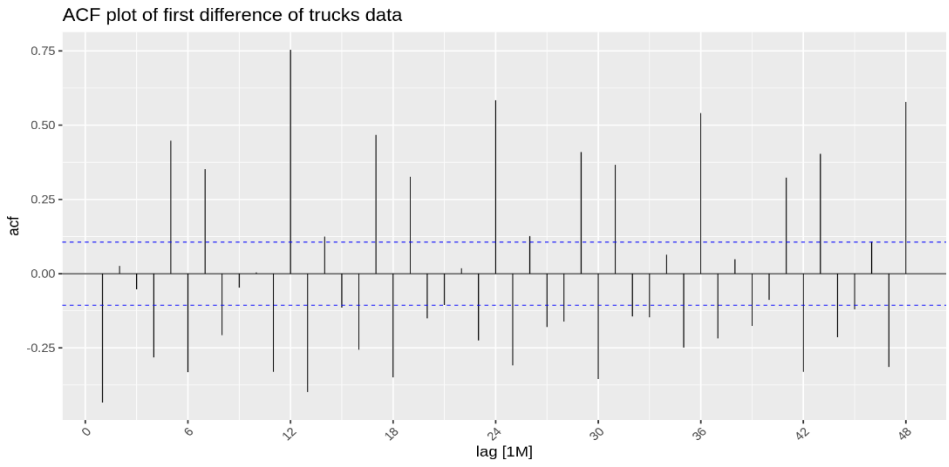
AutoRegressive (AR): Captures the relationship between current and past observations, represented by parameter p.

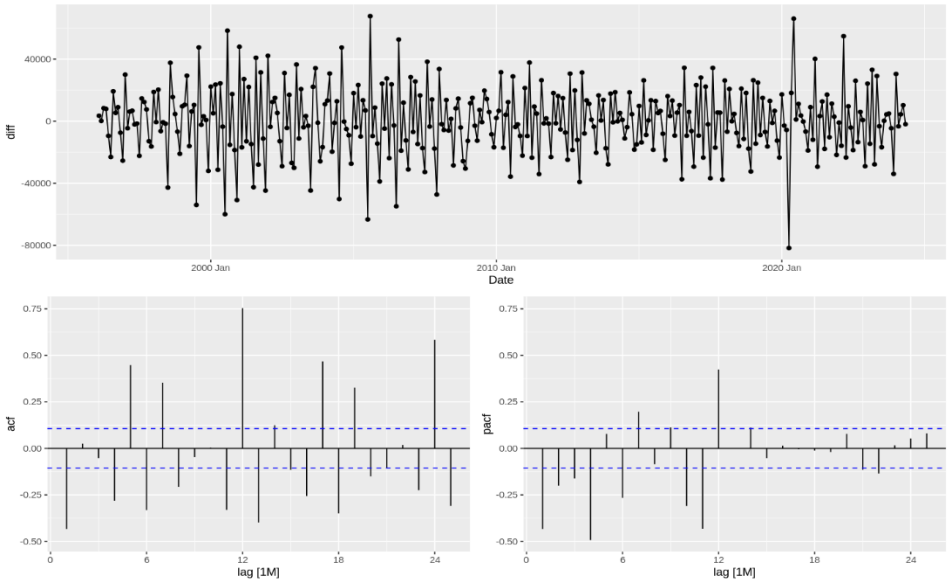
Integrated (I): Handles differencing to ensure the data is stationary, indicated by parameter d.

Moving Average (MA): Accounts for the impact of past forecast errors on current values, denoted by parameter q.

The ARIMA model is defined by the parameters (p,d,q), and it is effective for forecasting and analyzing time series data with trends and seasonal patterns. For our analysis, ARIMA helped forecast the future number of trucks crossing the borders trends which will help for better-informed decision making.

Before moving onto the next step, we checked the stationarity of the time series data earlier and we will now use it for modeling displaying it through ACF plot and the combined plot generated by the **gg_tsdisplay()** function from the **fable** package in R provides a comprehensive view of time series data through a set of plots like **time-series plot, ACF, and PACF plot** below.





From the ACF, and PACF plots, we can decipher the value of **AR(p),** and **MA(q),** since the value of differencing **I(d),** we have explained above that we took first differencing, and plot was visually stationary. For model selection, we decided several combinations of p, d, q to consider which we saw from the ACF and PACF plot. The range of p, d, q are mentioned below.

p = (0,1,2), d = (1), q = (1,2)

The above combination we tried to fit the AR1, AR2, AR3, AR4, and AR5 models to come up with the best accuracy, the models are shown below -

AR1: ARIMA(Value) with default parameters

AR2: ARIMA(Value ~ pdq(1,1,1)) where $p = 1, d = 1, q = 1$

AR3: ARIMA(Value ~ pdq(2,1,2)) where $p = 2, d = 1, q = 2$

AR4: ARIMA(Value ~ pdq(0,1,1)) where $p = 0, d = 1, q = 1$

AR5: ARIMA(Value ~ pdq(0,1,2)) where $p = 0, d = 1, q = 2$

Report:

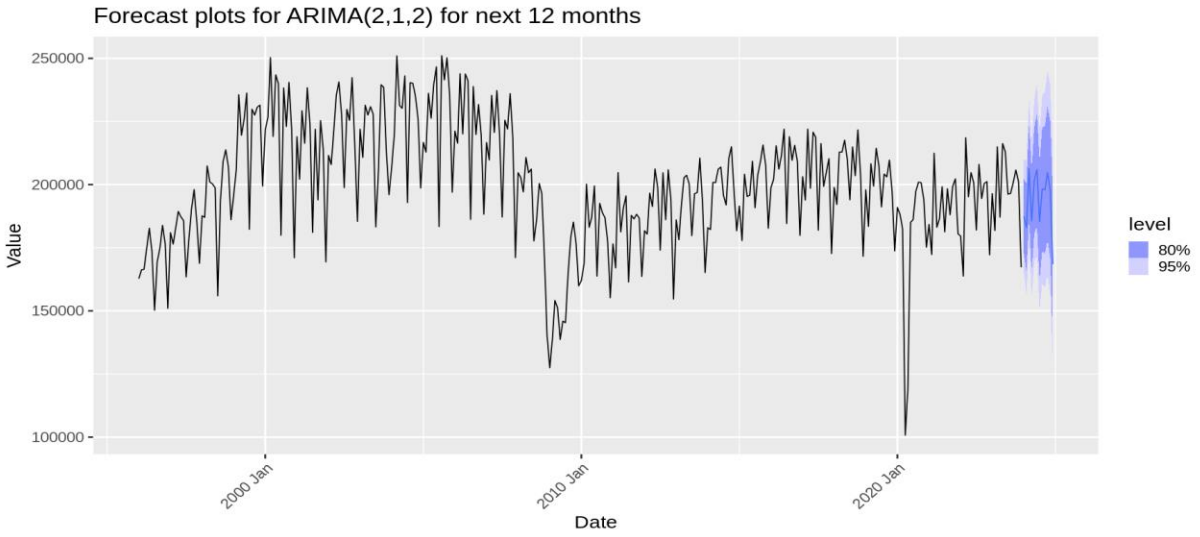
Model	ARIMA Order	Sigma^2 Estimate	Log Likelihood	AIC	AICc	BIC
AR1	(2,0,2) (0,1,2) [12]	128278747	-3487.68	6989.37	6989.72	7015.83
AR2	(1,1,1) (1,1,2) [12]	134130348	-3485.44	6982.88	6983.14	7005.54
AR3	(2,1,2) (0,1,2) [12]	131904468	-3481.38	6976.75	6977.11	7003.20
AR4	(0,1,1) (0,1,2) [12]	136006892	-3488.38	6984.76	6984.88	6999.87
AR5	(0,1,2) (1,1,2) [12]	133841366	-3485.06	6982.11	6982.38	7004.78

- **Sigma^2 Estimate:** This represents the estimated variance of the residuals (errors) in the ARIMA model, which indicates the spread of the error terms.
- **Log Likelihood:** A measure of how well the model fits the data. Higher values indicate a better fit.
- **AIC (Akaike Information Criterion):** A measure used to compare models, balancing model fit and complexity. Lower AIC values suggest a better model.
- **AICc (Corrected AIC):** An adjusted version of AIC that accounts for small sample sizes, helping to prevent overfitting.
- **BIC (Bayesian Information Criterion**): Like AIC but includes a stronger penalty for model complexity. Lower BIC values indicate a more parsimonious model.

Therefore, after the evaluation from the report, the best model for forecasting is AR3 (ARIMA (2,1,2) (0,1,2)[12]), as it has the lowest AIC (6976.75) and BIC (7003.20) values, which indicates the best balance between fit and complexity. Its log likelihood (-3481.38) also supports this selection, which makes it the most suitable model for accurate forecasting.

Forecasting Future Trends with Optimal AR3

After determining that AR3 (ARIMA (2,1,2) (0,1,2) [12]) is the best model for forecasting, we proceed with generating forecasts for the next 12 months. We use the forecast function on the best-fitting model and visualize the results using autoplot. The plot illustrates the forecast, which provides insights into the expected trends and shows the forecast is capturing similar pattern, but the prediction interval is still wide.



Model Evaluation Metrics for ARIMA (2,1,2)

Metric	Value
Mean Squared Error (MSE)	152324014.69501
Root Mean Squared Error (RMSE)	12341.9615416274

Mean Absolute Percentage Error (MAPE)	0.0569021159753049
MAPE (In percentage)	5.69021159753049
Mean Absolute Error (MAE)	10920.4957627266

Finding Best ARIMA Model Using Stepwise and Grid Search

Stepwise Selection: We used stepwise selection to find the optimal ARIMA model. This method involves iteratively adding or removing ARIMA parameters based on criteria such as AIC or BIC. The process starts with a simple model and progressively includes more parameters, selecting the one that best balances model fit and complexity.

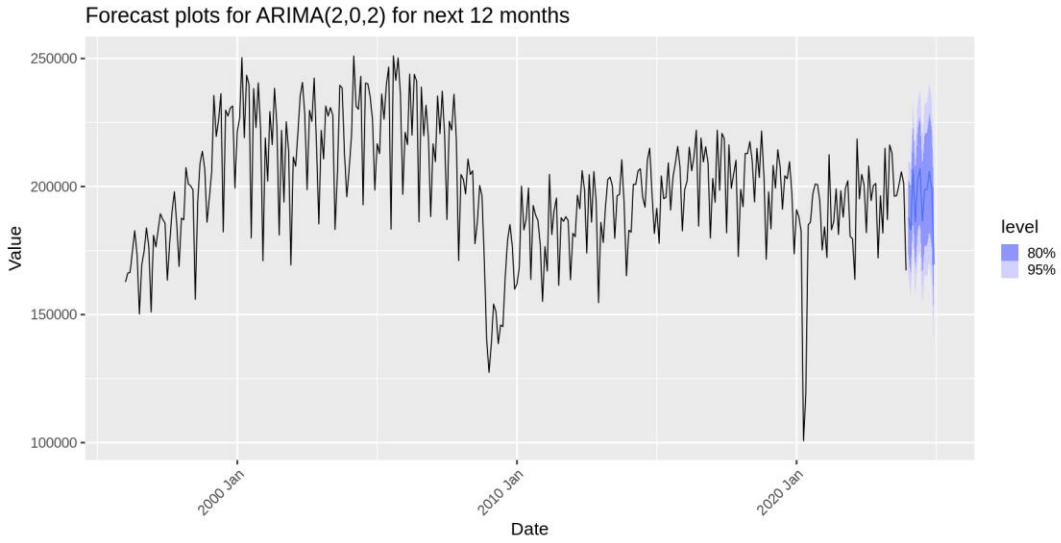
Grid Search: We also employed grid search to identify the best ARIMA model. This technique involves evaluating all possible combinations of ARIMA parameters (p, d, q) within a specified range. By systematically testing these combinations, we selected the model with the lowest AIC or BIC values.

We got the optimal combination of p, d, q for the ARIMA model after running the Stepwise and search grid as below.

stepwise	search
<model>	<model>
<ARIMA(2,0,2)(0,1,2)[12]>	<ARIMA(2,0,2)(0,1,2)[12]>

ARIMA (2,0,2) (0,1,2) [12] Model Summary and Forecast Plot

Metric	Value
Sigma^2 Estimate	128278747
Log Likelihood	-3487.68
AIC	6989.37
AICc	6989.72
BIC	7015.83



Interpretation

The ARIMA (2,0,2) (0,1,2) [12] model metrics indicate a good fit with a sigma^2 estimate of 128,278,747 and low AIC and BIC values. The forecast plot aligns well with these results, capturing almost similar trends. However, the prediction intervals remain wide, reflecting some uncertainty in the forecasts for the next 12 months.

Seasonal ARIMA Model

In our group project, we utilized the Seasonal ARIMA (SARIMA) model for forecasting the US-Canada trucks crossing the border. The SARIMA model extends the ARIMA framework by incorporating seasonal components, structured as follows:

AutoRegressive (AR): Captures the relationship between current and past observations, denoted by parameter p.

Integrated (I): Handles differencing to achieve stationarity, indicated by parameter d.

Moving Average (MA): Accounts for the impact of past forecast errors on current values, represented by parameter q.

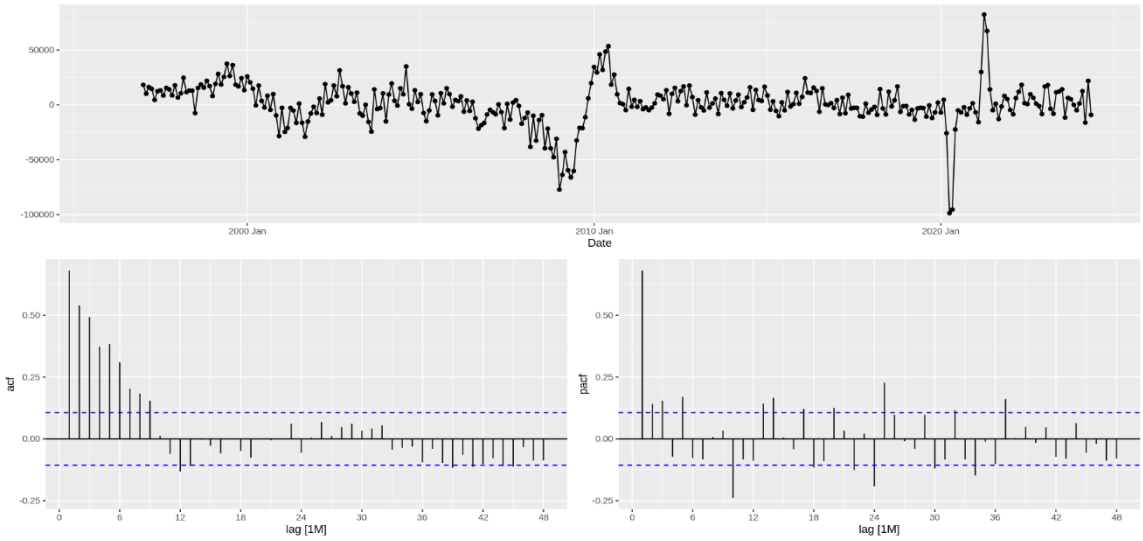
Seasonal Components (PDQ): Handles seasonal variations, with seasonal AR, differencing, and MA parameters denoted by P, D, and Q, respectively.

The SARIMA model is defined by the parameters (p, d, q) x (P, D, Q)s and is effective for forecasting time series data with seasonal patterns. For our analysis, SARIMA helped forecast future truck crossing trends, aiding in more informed decision-making.

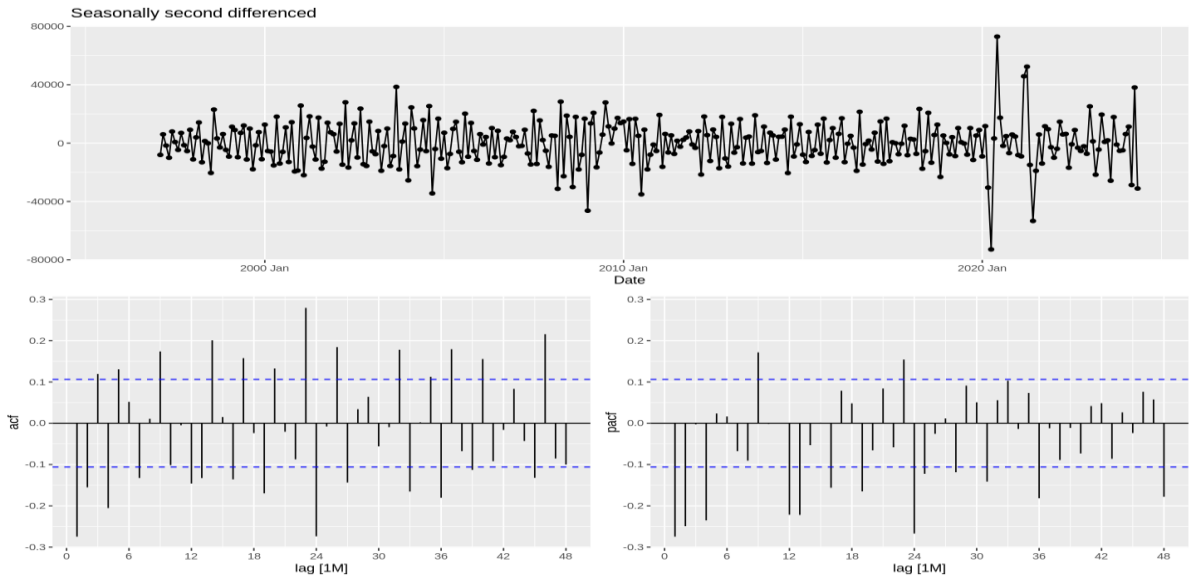
Before moving to the next step, we seasonally difference our data to smoothen so that we can make it seasonally stationary.

Seasonal Differencing Analysis:

Seasonal First Difference ACF: The combined plot of the seasonally first differenced data (lag 12) shows almost stationary pattern but still with few spikes. The plot below shows it clearly.



Seasonal Second Difference ACF: The combined plot after applying a second seasonal difference confirms stationarity and far better than the first differencing.



Here we can see that the second difference of seasonal first difference looks stationary and both PACF and ACF have controlled spikes.

Seasonal MA (2) since we two significant spikes at ACF plot at 12 and 24 months

Non-seasonal difference, we can visually see in ACF that we have the first two spikes going down and in PACF we have the first 3 significant spikes showing downward trend. so, we choose non-seasonal AR(2) or AR (3).

Model Selection and Evaluation:

We evaluated several SARIMA models using several combinations of p, d, q and P, D, Q.

SARIMA Model Report:

Model	ARIMA Order	Sigma^2 Estimate	Log Likelihood	AIC	AICc	BIC
SAR2	ARIMA(0,1,2)(0,1,2)[12]	133,457,853	-3485.14	6980.28	6980.47	6999.17
SAR3	ARIMA(2,1,0)(0,1,2)[12]	135,946,706	-3487.51	6985.02	6985.21	7003.91
SAR4	ARIMA(2,0,2)(0,1,2)[12]	128,278,747	-3487.68	6989.37	6989.72	7015.83

Interpretation:

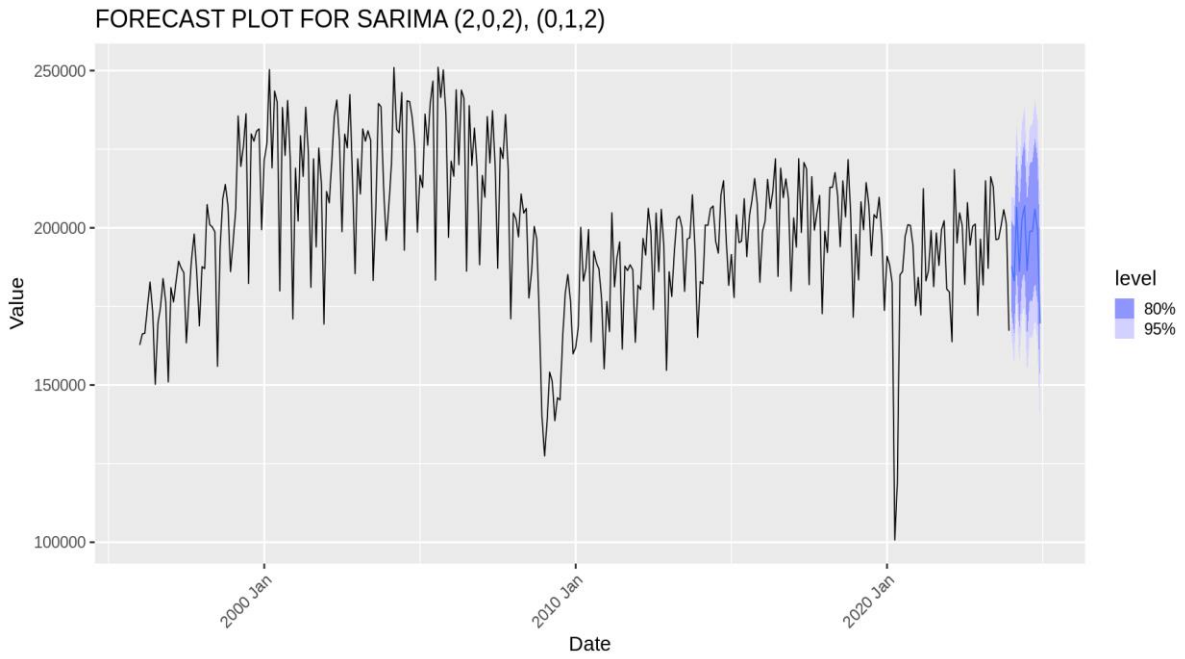
- SAR2 (ARIMA(0,1,2)(0,1,2)[12]) and SAR3 (ARIMA(2,1,0)(0,1,2)[12]) are considered based on their AIC and BIC values.
- SAR4 (ARIMA(2,0,2)(0,1,2)[12]) was selected as the best model for forecasting due to its lowest AIC and BIC values, which indicates a good balance between model fit and complexity.

Forecasting Future Trends with optimal SARIMA (2,0,2) (0,1,2) Using Stepwise and Grid Search

After selecting the SARIMA(2,0,2) (0,1,2) [12] model we forecasted and the report we can see below by comparing the matrices. Also, the forecast plot is visually clear to check how the pattern is being captured despite the wide prediction interval.

SARIMA (2,0,2) (0,1,2) Model Summary and Forecast Plot

Metric	Value
Mean Squared Error (MSE)	141,439,777.87
Root Mean Squared Error (RMSE)	11,892.85
Mean Absolute Percentage Error (MAPE)	0.0496
MAPE (In percentage)	4.96%
Mean Absolute Error (MAE)	9,266.66



Summary

MODEL	RMSE	MAE
ETS	9555.302	8955.878
Multiple Linear	5378.155	2919.457
ARIMA	12508.005	11063.411
SARIMA	12341.961	10920.49

- ARIMA AND SARIMA show similar accuracy but overall Multiple Linear Regression shows the best accuracy in terms of RMSE and MAE.
- We recommend use of Linear regression since it's easy to implement with less computational costs.