Us International Trade (Exports) analysis using Time Series Analytics

Venkata Sai Bala Krishna Batchu - im2250

Yaswant Bharadwaj Valluri – cq8996

Surya Sai Dinesh Addanki – ip2139

Bhargav Raj Veerla – xr2504

Summary

For this Project we extract export data from CENSUS.GOV has been used to predict quarterly beer production for the upcoming fiscal year. We began our investigation by looking through various time series data. Before running the R code, we performed a simple graphical depiction of the data in Excel. We have chosen US Exports data from census.gov after thoroughly reviewing the data (US Department of commerce-us census). To determine whether the data we chose is predictable or a random walk before moving on to apply various time series models, we first applied predictability approaches. Our data appears to be a random walk, but we go ahead and analyze it as a time series because there is some autocorrelation with the data's lag1 period and from the visualization we identified that data is having an upward trend. After that, we ran a few time series models, including:

1. Two-level model (regression model with linear trend and seasonality with MA for residuals
2. Two-level model (regression model with linear trend and seasonality with AR(1) for residuals
3. Holt-winter's model with automated selection of error/level, trend and seasonality
4. ARIMA model with automated selection of Autoregression, order of differencing and moving average

After running the forementioned models, we determined that the **Holt-Winter model was the best one, and we then compared the model's accuracy metrics to those of the Naive and Seasonal Naive forecasts. Even when compared to simple naive forecasts, the Holt-Winters model provides better predictions.**

To comparing the accuracy between models we consider measures such as,

- **MSE**: Mean Squared Error
- **RMSE**: Root Mean Squared Error
- **MAE**: Mean Absolute Error
- **MAPE**: Mean Absolute Percentage Error

Model evaluation was based on the RMSE and MAPE accuracy metrics.

Introduction

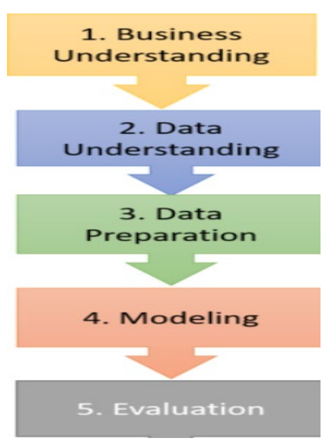
Data on US merchandise exports from the United States to all countries except Canada is compiled from Electronic Export Information (EEI) filed by the USPPI or their agents via the Automated Export System (AES). The EEI is distinct among Census Bureau data collection methods in that it is not sent to respondents soliciting responses, as surveys are. Each EEI represents a shipment of one or more types of merchandise on a single carrier from one exporter to one foreign importer. The Census Bureau's foreign trade statistics program is unique among its economic statistics programs in that the information is not gathered from forms sent to respondents soliciting responses, as in the case of surveys. Rather, the data is compiled from automated forms and reports initially filed with the United States Customs Service or, in some cases, directly with the Census Bureau for virtually all shipments leaving the country (exports).

The dataset we used here is US international trade in goods and services ranging from 1992 to 2022 from census.gov website. Export is defined as **an actual shipment or transmission of items out of the United States.** This includes standard physical movement of items across the border by truck, car, plane, rail, or hand-carry.

Time series analytics is an important aspect of predictive analytics concerned with producing predictions by applying time series forecasting. A time series is a collection of data points that have been collected in a timely manner. It is an uninterrupted group of subsequent data observations that have been organized in time at evenly spaced intervals, such as a day, month, or quarter.

Compared to time series, cross-sectional data recordings are collected at the same point of time or without respect to difference in time.

The scope of this project is to forecast the upcoming months exports data based on past time series data. This will greatly aid the US economy growth based on the demand forecasted early.



Eight Steps Involved in Time Series Forecasting

Step 1: Define Goal

Forecasting US export data for the future quarters of 2023 for the upcoming fiscal year is the aim of this research. The objective is to develop a time series forecast predictive model that will accurately forecast the target months while considering all relevant aspects of the past data. Each year's data is accessible in monthly format. Naturally, the model of preference will be the one with the maximum accuracy. The projections that are made as a result will be used to track the forecasting of US exports. The R programming language was used to create the forecasting models for this project.

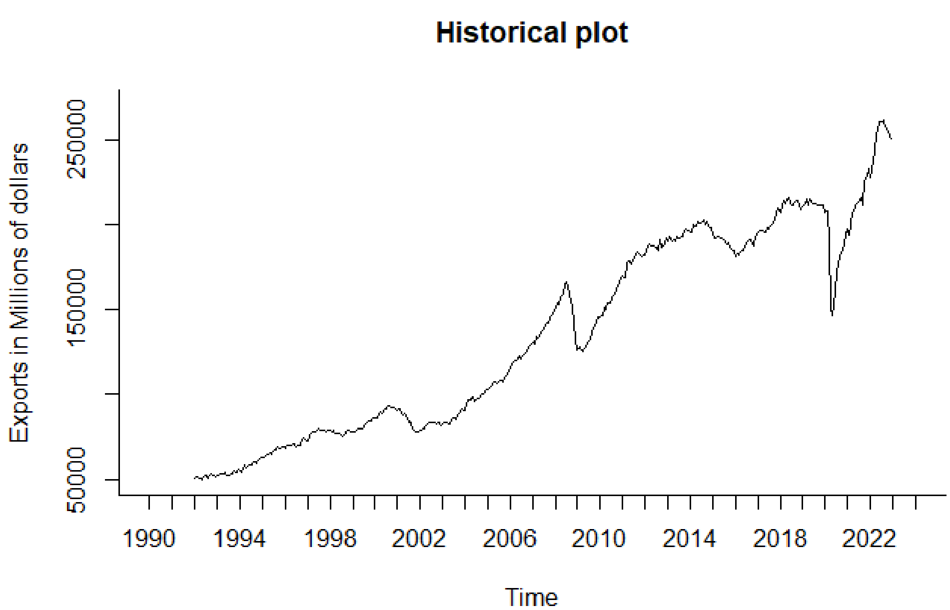
Step 2: Get data

In this project, we're using data that we pulled from the census.gov website (run by the US Department of Commerce-US Census) to get monthly figures in millions of dollars for US-International Trade from 1992 to 2022.

The dataset spans a time period of monthly data with 372 data points from 1992 January to 2022 December, and we'd like to anticipate data from 2023 January to 2023 December (for 12 months). Below the references section, a link to the data reference will be attached.

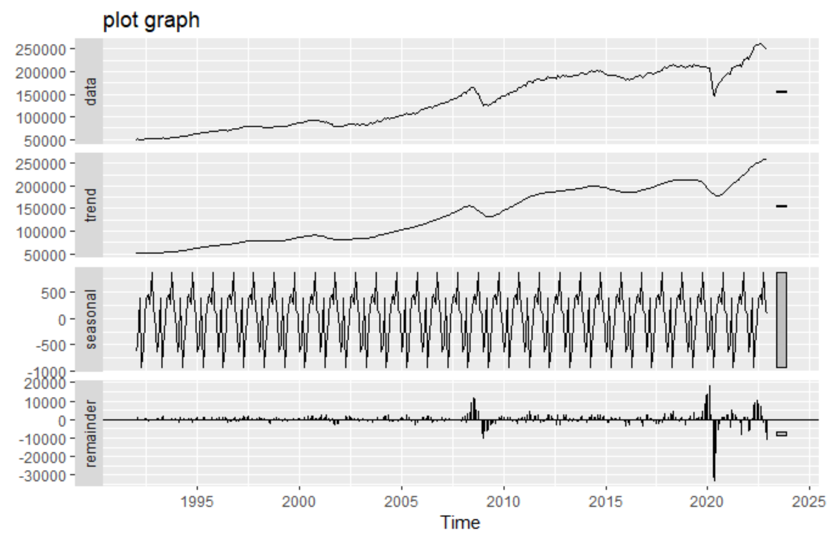
Step 3: Explore and Visualize Series

The data plot shown below shows how the historical data changed over time. The statistics in this time series seem to be trending upward. Yet, exports suffer significantly during recessions and terrible times like 2008 and 2020. however, it finally increased by a year.



Time series components of Historical Data:

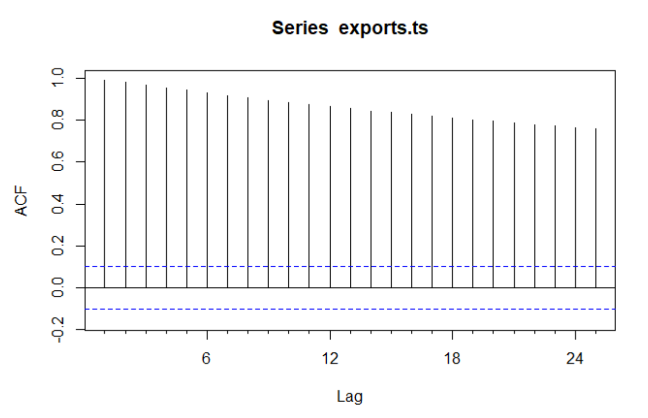
By looking at the above historical plot, we can observe that the data is linearly increasing over the time which tells us that there is trend component in this time series data. The NOISE OR REMAINDER IS ALSO ONCE DURING CERTAIN times like during unexpected scenarios.



We can infer from the above plot that there is an upward trend, additive seasonality, and that level component is still there. Except for the section of the data where there is a dramatic dip practically at the conclusion of the series, we can see that there is very little noise in the data.

Autocorrelation Plot:

All the lags' autocorrelation coefficients are significantly higher than the horizontal threshold (significantly greater than zero). As compared to the other lags in the series, lag 1's positive autocorrelation coefficient is higher than the horizontal threshold and is also thought to have the highest correlation, which suggests the presence of an upward trend component. When seasonal lags are considered, their coefficients are substantial, but not more so than the initial lag.



Step 4: Data Preprocessing

```
Values
  exports.ts        Time-Series [1:372] from 1992 to 2023: 50251 51682 50294 …
```

There are two columns: one lists the month and year, while the other contains information about exports valued in millions of dollars. We handled the comma-separated values in this Exports column in the code before using the "ts()" function to turn the data into time series data.

372 observations are contained in the time series data exports.ts file.

Checking the predictability of data

We have tested the dataset for the predictability check whether the dataset is a random walk or is it predictable?

->Approach1- Arima-Ar(1):

```
ar1 <- 0.9993
s.e. <- 0.0010
null_mean <- 1
alpha <- 0.05
z.stat <- (ar1-null_mean)/s.e.
z.stat
p.value <- pnorm(z.stat)
p.value
if (p.value<alpha) {
  "Reject null hypothesis"
} else {
  "Accept null hypothesis"
}
```

```
> ar1 <- 0.9993
> s.e. <- 0.0010
> null_mean <- 1
> alpha <- 0.05
> z.stat <- (ar1-null_mean)/s.e.
> z.stat
[1] -0.7
> p.value <- pnorm(z.stat)
> p.value
[1] 0.2419637
> if (p.value<alpha) {
+   "Reject null hypothesis"
+ } else {
+   "Accept null hypothesis"
+ }
[1] "Accept null hypothesis"
```

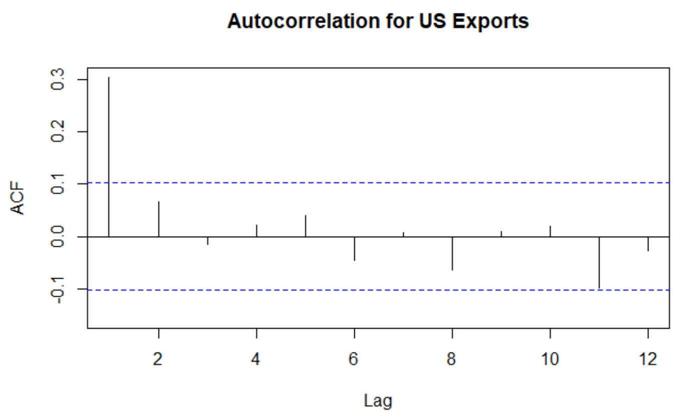
With this approach the P-value is around 0.241 and this results in

"Accept null hypothesis"

With this, we could say that the time series data we have is a random walk and is not predictable with this approach.

->Approach 2-ACF with differencing lag1

An upward trend component is shown by the positive autocorrelation coefficient in lag 1 being significantly higher than the horizontal threshold. As a result, it can be concluded that the data ca n be somewhat expected because there is an upward trend and a substantial value for lag1.



Step 5: Partition Series:

Out of the 372 data points we have, we receive 297.6 as 80% of the training data and 74.4 as 20% of the validation data when we divide the time series data. We divided the data into 288 records for the training period (24 years) and 84 records for the validation period since it is preferable to divide the yearly data into the proper ratio of years (7 years).

These partitioned data sets are:

Training data: train.ts
Validation data: valid.ts

train.ts	Time-Series [1:288] from 1992 to 2016:
valid.ts	Time-Series [1:84] from 2016 to 2023:

Step 6 & 7: Apply Forecasting & Comparing Performance

1) **Two level forecast (linear trend and seasonality with moving average for residuals)**

Two-level forecasting, which combines two forecasting models, may be used to apply the trailing MA in data with trend and/or seasonality:

Level1: Regression model with linear trend and/or seasonality. It can also be used to e liminate trends and/or seasonality from historical data (de-trending and/or de-seasonalizing)

Find residuals (errors): discrepancies between regression forecast and actual Exports for various time periods.

Level 2: The residuals (errors) of the regression model can be predicted using the trailing MA.

Regression model and trailing MA forecasts are combined (sum) to get the overall forecast utilized in predictions.

A trailing moving average was utilized to anticipate model residuals and enhance the linear trend and seasonality regression model. These elements were then brought together to develop a two-level model and a model over all the data that was used to forecast the following 12 months.

Model trained over training data

```
> summary(trend.seas)

Call:
tslm(formula = train.ts ~ trend + season)

Residuals:
   Min     1Q Median     3Q    Max
-27670  -9385   2881  10058  21554

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 33216.892   3074.002  10.806   <2e-16 ***
trend         556.561      9.619  57.858   <2e-16 ***
season2        39.189   3914.516   0.010    0.992
season3       932.544   3914.551   0.238    0.812
season4       663.233   3914.611   0.169    0.866
season5       833.713   3914.693   0.213    0.832
season6       783.027   3914.800   0.200    0.842
season7       705.674   3914.930   0.180    0.857
season8       482.779   3915.083   0.123    0.902
season9       282.009   3915.261   0.072    0.943
season10      572.656   3915.461   0.146    0.884
season11     -109.905   3915.686  -0.028    0.978
season12     -432.008   3915.934  -0.110    0.912
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 13560 on 275 degrees of freedom
Multiple R-squared:  0.9242,    Adjusted R-squared:  0.9209
F-statistic: 279.4 on 12 and 275 DF,  p-value: < 2.2e-16
```

Looking at Adjusted R-squared value 0.9209 (92%), we can conclude that the model is a good fit. Considering overall p-value and p-value for only trend component, this model is statistically significant. May be applied for time series forcasting

Model Equation: (Regression model with linear trend and seasonality)

$$y_t = \beta_0 + \beta_1 t + \beta_2 D_2 + \beta_3 D_3 + \ldots + \beta_{12} D_{12} + \varepsilon$$

In this case:

$$y_t = 33216.89 + 556.56 * t + 39.18 * D_2 + 932.54 * D_3 + 663.233 * D_4 + 833.713 * D_5 + 783.02 * D_6 + 705.67 * D_7 + 482.779 * D_8 + 282 * D_9 + 575.65 * D_{10} + (-109.90) * D_{11} + (-432) * D_{12}$$

where, $t = 1,2,3……n$ (n=number of time periods/trends)

D_2 = binary (1,0), it is 1 if Feb and 0 if otherwise

D_3 = binary (1,0), it is 1 if Mar and 0 if otherwise

.

.

D_{12} = binary (1,0), it is 1 if Dec and 0 if otherwise

If D_2 , D_3 ,…, D_{12} are 0 then it is Jan

<u>Selecting K (window width):</u>

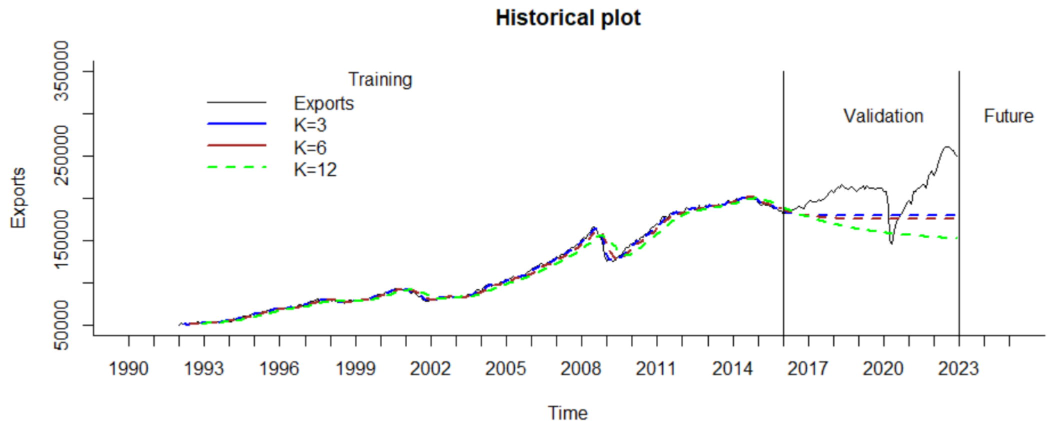
As there is little to no seasonality observed in the data, we'd like to select a narrower window width, which also makes it easier to see local trends. Yet, we want to use the trailing Moving Average approach for various widths to the training data and choose the ideal window width with statistical support (by comparing accuracy metrics for all chosen widths). This method will improve our forecast.

To select the width of moving average model for residuals, we compare the accuracy measures for various window widths,

```
> ma.trailing_3 <- rollmean(train.ts, k = 3, align = "right")
> ma.trailing_6 <- rollmean(train.ts, k = 6, align = "right")
> ma.trailing_12 <- rollmean(train.ts, k = 12, align = "right")
> round(accuracy(ma.trail_3.pred$mean, valid.ts), 3)
               ME     RMSE      MAE    MPE   MAPE  ACF1 Theil's U
Test set 28207.63 36892.59 30476.41 12.453 13.922 0.935     4.915
> round(accuracy(ma.trail_6.pred$mean, valid.ts), 3)
               ME     RMSE     MAE   MPE   MAPE  ACF1 Theil's U
Test set 31137.36 39734.51 33082.5 13.84 15.096 0.935     5.298
> round(accuracy(ma.trail_12.pred$mean, valid.ts), 3)
               ME     RMSE      MAE    MPE   MAPE  ACF1 Theil's U
Test set 43412.96 52788.83 44323.27 19.628 20.186 0.941     7.073
>
```

Least MAPE AND RMSE is for the model with **k=3 (for window width 3)**

<u>Window width graph vs historical data:</u>



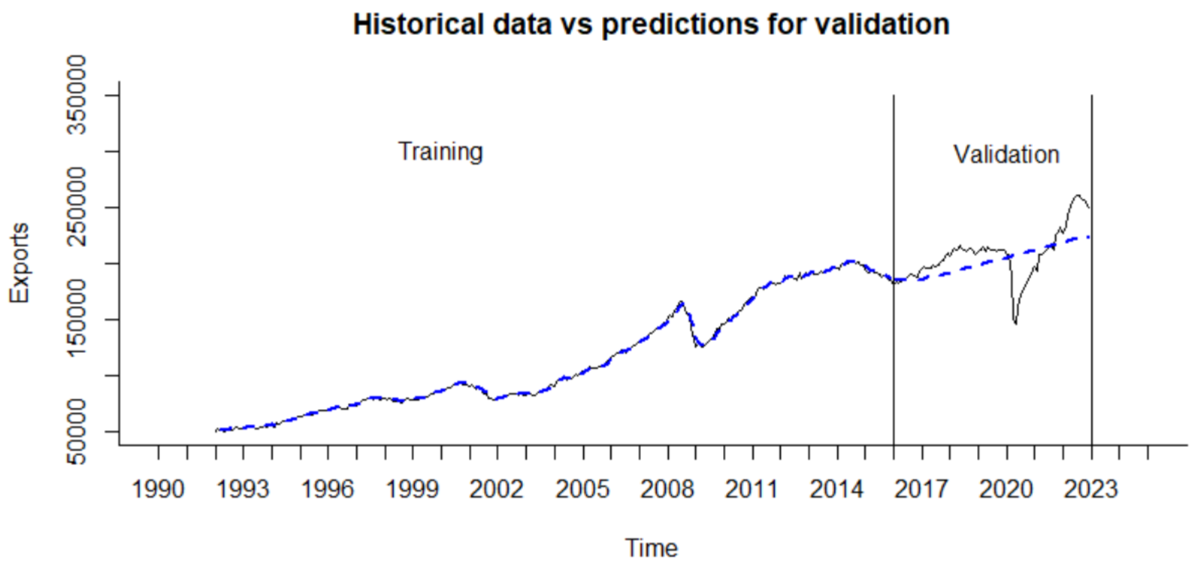
By looking at the above accuracy measures and graphical representation of various widths, we can conclude that window width 3 (k=3) is the best one to perform forecast for residuals/errors.

Two-level predictions = regression model with linear trend and seasonality + Trailing MA for residuals

<u>Two level forecasting for validation data</u>

```
> fst.2level
          Jan      Feb      Mar      Apr      May      Jun      Jul      Aug      Sep      Oct
2016 185489.6 184807.4 185195.2 184599.8 184593.4 184489.6 184462.2 184374.9 184380.7 184937.2
2017 185424.8 185881.8 187216.5 187407.9 188055.4 188495.1 188919.3 189207.3 189525.1 190340.7
2018 191371.6 191952.3 193389.7 193666.6 194385.0 194883.6 195356.9 195685.6 196037.3 196881.0
2019 197970.9 198565.0 200013.6 200299.7 201025.8 201530.9 202009.5 202342.6 202697.9 203544.8
2020 204641.0 205236.6 206686.4 206973.5 207700.4 208206.2 208685.3 209019.0 209374.7 210221.9
2021 211318.8 211914.5 213364.4 213651.7 214378.7 214884.6 215363.8 215697.4 216053.2 216900.4
2022 217997.4 218593.2 220043.1 220330.4 221057.4 221563.3 222042.5 222376.1 222731.9 223579.1
          Nov      Dec
2016 184569.5 184603.1
2017 190188.5 190401.1
2018 196752.2 196984.3
2019 203418.5 203652.7
2020 210095.8 210330.3
2021 216774.4 217008.9
2022 223453.1 223687.6
```

Prediction plot for validation:



To just perform a basic check whether the select regression model for level 1 of two level forecast is perfect, we applied a basic model "Regression model with linear trend" to training data and compared the accuracy measures of these both

Two level forecast of (linear trend) , moving average for residuals-Model trained over training data

```
> trend.reg <- tslm(train.ts ~ trend)
> summary(trend.reg)

Call:
tslm(formula = train.ts ~ trend)

Residuals:
   Min     1Q Median     3Q    Max 
-27232  -9464   3198   9761  21867 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 33623.116   1571.919   21.39   <2e-16 ***
trend         556.491      9.429   59.02   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 13300 on 286 degrees of freedom
Multiple R-squared:  0.9241,	Adjusted R-squared:  0.9239 
F-statistic:  3483 on 1 and 286 DF,  p-value: < 2.2e-16
```

Model Equation:

$$y_t = \beta_0 + \beta_1 t + \varepsilon$$

In this case:

$$y_t = 33623.11 + 556.49 * t$$

where, t = 1,2,3……n (n=number of time periods/trends)

Forecast values:

```
> fst.2level.reg <- trend.reg.pred$mean + ma.trail.reg.pred$mean
> fst.2level.reg
          Jan      Feb      Mar      Apr      May      Jun      Jul      Aug      Sep      Oct
2016 185304.7 184422.6 183785.7 183352.0 183087.0 182961.9 182953.0 183040.3 183207.6 183441.2
2017 184436.1 184839.6 185269.2 185720.4 186189.5 186673.5 187169.9 187676.5 188191.6 188713.7
2018 190311.4 190851.7 191394.6 191939.9 192487.1 193035.9 193586.0 194137.2 194689.3 195242.1
2019 196904.0 197458.7 198013.8 198569.1 199124.6 199680.3 200236.1 200792.0 201348.0 201904.1
2020 203572.8 204129.1 204685.4 205241.8 205798.2 206354.6 206911.0 207467.4 208023.9 208580.3
2021 210249.7 210806.2 211362.7 211919.1 212475.6 213032.1 213588.6 214145.1 214701.6 215258.0
2022 216927.5 217484.0 218040.5 218597.0 219153.5 219710.0 220266.4 220822.9 221379.4 221935.9
          Nov      Dec
2016 183729.8 184064.0
2017 189241.7 189774.6
2018 195795.6 196349.6
2019 202460.3 203016.5
2020 209136.8 209693.3
2021 215814.5 216371.0
2022 222492.4 223048.9
```

```
> round(accuracy(trend.seas.pred$mean + ma.trail.res.pred$mean, valid.ts), 3)
               ME     RMSE      MAE   MPE  MAPE  ACF1 Theil's U
Test set 5775.454 19960.35 15541.77 1.882 7.617 0.938     3.124
> #twolevel accuracy using regression only tend,ma for residuals over training data
> round(accuracy(trend.reg.pred$mean+ ma.trail.reg.pred$mean, valid.ts), 3)
               ME     RMSE      MAE   MPE  MAPE  ACF1 Theil's U
Test set 7182.235 20398.98 16232.51 2.567 7.915 0.938     3.144
```

It is evident that **"Regression model with Linear trend and seasonality"** for level 1 performed better than simple regression model linear trend by looking at the measures.

Hence, we decided to use the former model for the entire dataset for level 1

Regression model(trend+seasonality),MA for residuals over Entire data:

```
> summary(tot.trend.seas)

Call:
tslm(formula = exports.ts ~ trend + season)

Residuals:
   Min     1Q Median     3Q    Max 
-70152  -7263   1624   8537  29927 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 35903.797   3052.800  11.761   <2e-16 ***
trend         528.825      7.382  71.633   <2e-16 ***
season2       219.626   3881.787   0.057    0.955    
season3      1183.220   3881.808   0.305    0.761    
season4      -165.573   3881.843  -0.043    0.966    
season5       168.150   3881.892   0.043    0.965    
season6       653.583   3881.955   0.168    0.866    
season7       918.919   3882.033   0.237    0.813    
season8       965.609   3882.124   0.249    0.804    
season9       747.816   3882.229   0.193    0.847    
season10     1464.088   3882.348   0.377    0.706    
season11      864.069   3882.482   0.223    0.824    
season12      928.534   3882.629   0.239    0.811    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 15280 on 359 degrees of freedom
Multiple R-squared:  0.9347,	Adjusted R-squared:  0.9325 
F-statistic: 428.2 on 12 and 359 DF,  p-value: < 2.2e-16
```

Looking at Adjusted R-squared value 0.9325 (93.2%), we can conclude that the model is a good fit. Considering overall p-value and p-value for only trend component, this model is statistically significant. May be applied for time series forecasting

Model Equation:

$$y_t = \beta_0 + \beta_1 t + \beta_2 D_2 + \beta_3 D_3 + \ldots + \beta_{12} D_{12} + \varepsilon$$

In this case:

$$y_t = 35903.79 + 528.82 * t + 219.62 * D_2 + 1183.22 * D_3 + (-165.57) * D_4 + 168.150 * D_5 + 653.58 * D_6 + 918.91 * D_7 + 965.609 * D_8 + 747.81 * D_9 + 1464.08 * D_{10} + 864.06 * D_{11} + 928.53 * D_{12}$$

where, $t = 1,2,3\ldots\ldots n$ (n=number of time periods/trends)

D_2 = binary (1,0), it is 1 if Feb and 0 if otherwise

D_3 = binary (1,0), it is 1 if Mar and 0 if otherwise

.

.

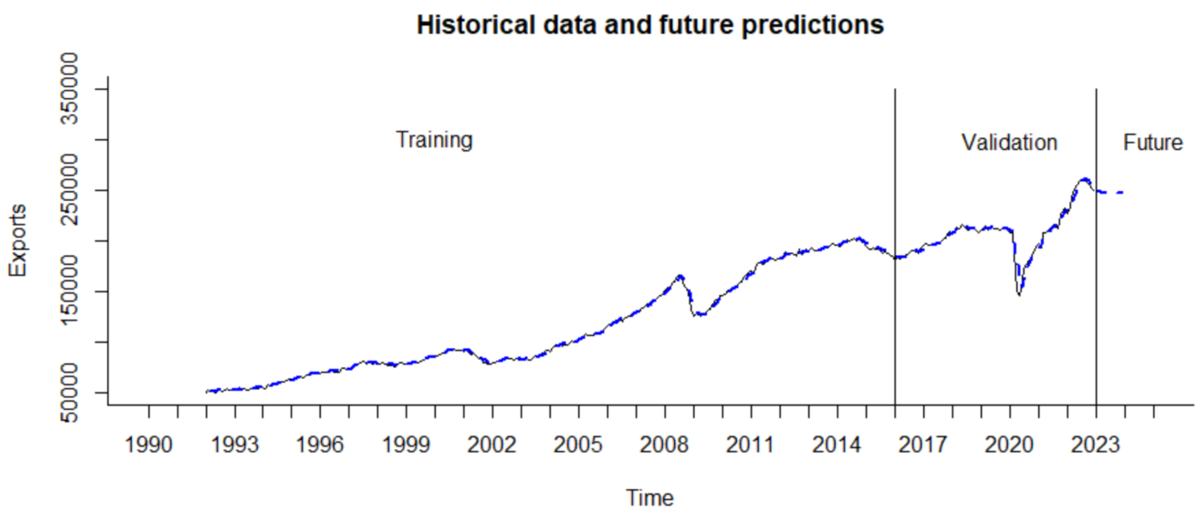
D_{12} = binary (1,0), it is 1 if Dec and 0 if otherwise

If D_2 , D_3 ,…, D_{12} are 0 then it is Jan

Forecast for 12 months in 2023

```
> # regression forecast and trailing MA for residuals for future
> # 12 periods.
> tot.fst.2level.train <- tot.trend.seas$fitted.values + tot.ma.trail.res
> tot.fst.2level <- tot.trend.seas.pred$mean + tot.ma.trail.res.pred$mean
> tot.fst.2level
          Jan      Feb      Mar      Apr      May      Jun      Jul      Aug      Sep      Oct
2023 250406.6 249090.1 248930.2 246788.0 246592.5 246760.2 246876.9 246910.3 246787.5 247685.5
          Nov      Dec
2023 247336.5 247707.6
>
```

Plot for the model built over entire data:



So the plot fits well when the model is built over entire data compared to when built only with training data.

Accuracy measures of this two level model when compared with seasonal naïve, naïve

As per the below accuracy means we can depict that the naivemodel has less mape and rmse values compared to our model.

```
> round(accuracy(tot.trend.seas.pred$fitted + tot.ma.trail.res.pred$fitted, exports.ts), 3)
            ME     RMSE      MAE    MPE  MAPE  ACF1 Theil's U
Test set 6.073 4212.474 2260.854 -0.102 1.646 0.535     1.167
> round(accuracy((snaive(exports.ts))$fitted, exports.ts), 3)
               ME     RMSE      MAE   MPE  MAPE  ACF1 Theil's U
Test set 6646.806 16902.12 12282.27 4.681 8.755 0.948     4.773
> round(accuracy((naive(exports.ts))$fitted, exports.ts), 3)
              ME     RMSE      MAE   MPE  MAPE  ACF1 Theil's U
Test set 538.817 3525.773 2000.299 0.406 1.488 0.303         1
```

So, its better to choose the any other model for this data for forecasting.

2) Two level Model-using AR(1)(trend,seasonality,AR(1) for residuals

For level 1, Regression model with linear trend and seasonality was already performed for training and entire dataset in the former model (point 1).

For level 2, in this model we use AR(1) model for residuals from regression model with linear trend and seasonality form level 1 predictions as performed below,

Autoregressive model (AR) idea: apply the autocorrelation directly in regression model using past observations as predictors.

Similar to linear regression models, the predictors are the past values of the time series

AR model of order 1, AR(1): Yt = a + b1Yt-1 + et

AR model of order 2, AR(2): Yt = a + b1Yt-1 + b2Yt-2 + et

Two approaches in using AR models in time series forecasting.

Two-level forecasting modeling with AR model for residuals (errors):

Level 1: Use any method to generate forecasts (In this case-regression model)

Level 2: Examine forecast residual series for autocorrelation by utilizing time plot of forecast residuals and ACF function plot

If autocorrelation of residuals significant, fit AR model to forecast residual series, To improve the regression model with linear trend and seasonality, a AR(1) was used to forecast residuals from the model. These components were then combined to create a two-level model which was used to predict the next 12 months.

In this case, we use AR(1) for residuals instead of Trailing MA

```
Series: trend.seas$residuals
ARIMA(1,0,0) with non-zero mean

Coefficients:
         ar1      mean
      0.9881  1489.900
s.e.  0.0081  8115.847

sigma^2 = 4223986:  log likelihood = -2606.43
AIC=5218.86   AICc=5218.94   BIC=5229.85

Training set error measures:
                    ME     RMSE      MAE      MPE     MAPE      MASE      ACF1
Training set -96.21947 2048.085 1475.236 6.237678 53.13992 0.1931062 0.1741509
>
```

<u>AR(1) model equation for errors/residuals:</u>

$$e_t = \alpha + \beta_1 e_{t-1} + \varepsilon_t$$

where, α = mean

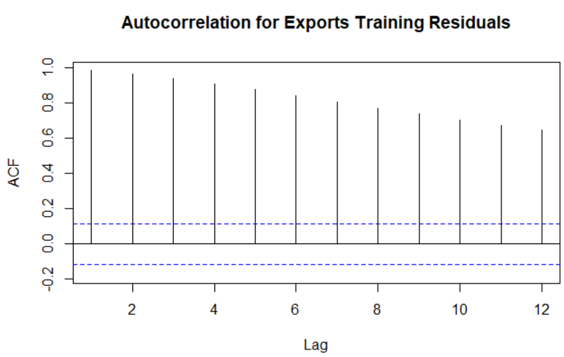
β_1 = ar1 coefficient

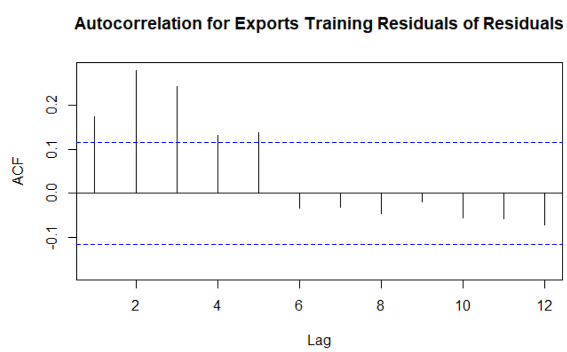
e_{t-1} = error forecast for lag 1

<u>In this case:</u>

$$e_t = 1489.9 + 0.9881 * e_{t-1}$$

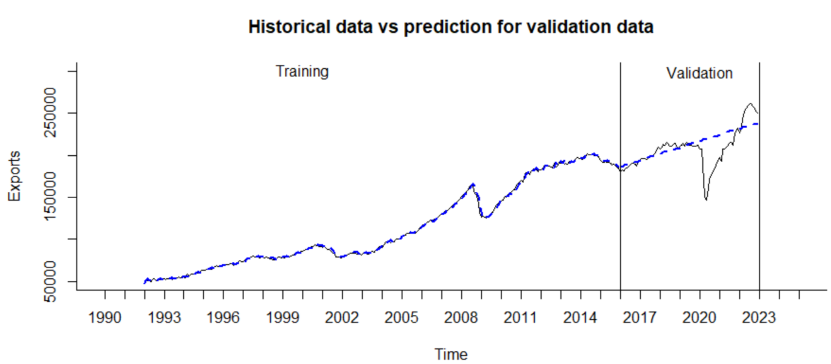
We Use Acf() function to identify autocorrelation for the training residual of residuals and plot a utocorrelation for different lags (up to maximum of 12) as below:





The left graph represents training residuals from regression model, which depicts that there is some information eft in the residuals to train the data. The right graph represents residuals pf residuals, which depicts that the information has been grasped from the AR(1) model for training residuals.

The model fits as per the below picture



Accuracy measure of this model compared with naïve,seasonal naïve:

```
> round(accuracy(trend.seas.pred$mean + res.ar1.pred$mean, valid.ts), 3)
                 ME     RMSE      MAE    MPE  MAPE  ACF1 Theil's U
Test set -5017.313 19806.12 12339.77 -3.306 6.393 0.943     3.412
> round(accuracy((snaive(valid.ts))$fitted, valid.ts), 3)
                ME     RMSE      MAE   MPE   MAPE  ACF1 Theil's U
Test set 10679.26 28701.74 22365.26 3.873 10.798 0.938     4.024
> round(accuracy((naive(valid.ts))$fitted, valid.ts), 3)
               ME     RMSE      MAE   MPE  MAPE  ACF1 Theil's U
Test set 833.747 6310.161 3658.301 0.328 1.875 0.354         1
> |
```

As per the accuracy measure for models using Training data set, we can observe that naïve forecasting is still the better with accuracy when compared with our current Two level model (Regression with trend,seasonality+Ar(1) for residuals).

Model built over Entire Dataset:

```
> residual.ar1 <- Arima(tot.trend.seas$residuals, order = c(1,0,0))
>
> # Use summary() to identify parameters of AR(1) model.
> summary(residual.ar1)
Series: tot.trend.seas$residuals
ARIMA(1,0,0) with non-zero mean

Coefficients:
         ar1      mean
      0.9739  2046.725
s.e.  0.0112  6187.436

sigma^2 = 11644949:  log likelihood = -3554.62
AIC=7115.23   AICc=7115.3   BIC=7126.99

Training set error measures:
                    ME     RMSE      MAE       MPE     MAPE      MASE      ACF1
Training set -39.86079 3403.284 1914.664 -24.32881 76.61612 0.1859703 0.3377239
```

AR(1) model equation for errors/residuals:

$$e_t = \alpha + \beta_1 e_{t-1} + \varepsilon_t$$

where, α = mean

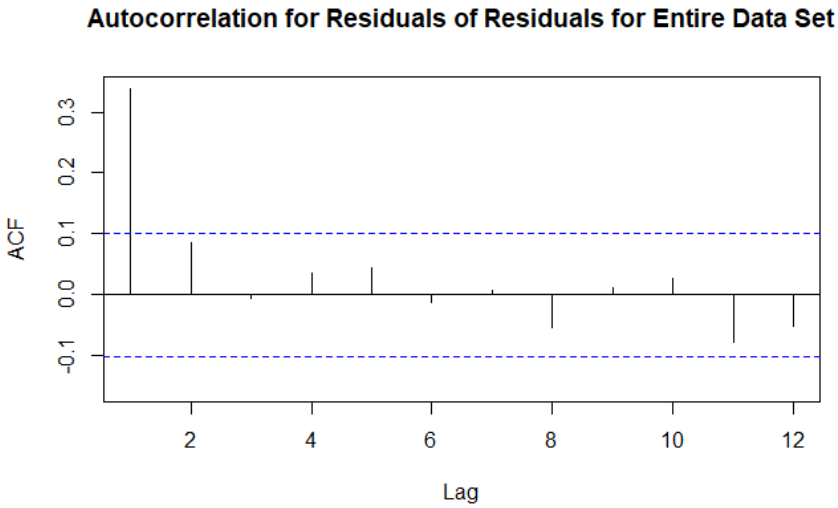
β_1 = ar1 coefficient

e_{t-1} = error forecast for lag 1

In this case:

$$e_t = 2046.72 + 0.9739 * e_{t-1}$$

Autocorrelation for Residuals of Residuals over entire data

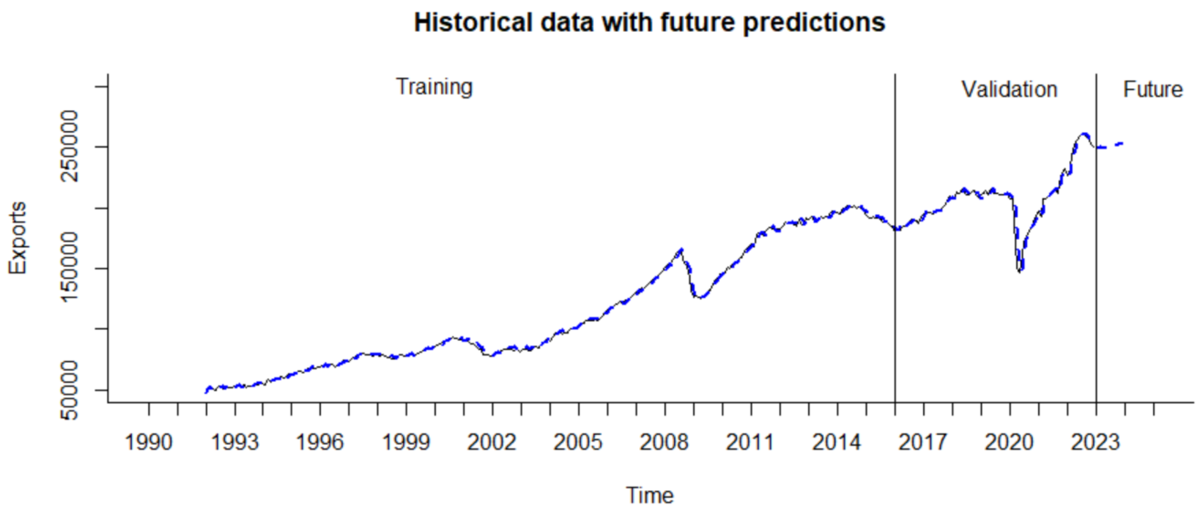


Lag 1 is significant for the residuals

Forecasting using Two level (Regression,ar(1)

```
> lin.season.ar1.pred <- tot.trend.seas.pred$mean + residual.ar1.pred$mean
> lin.season.ar1.pred
          Jan      Feb      Mar      Apr      May      Jun      Jul      Aug      Sep      Oct
2023 249372.0 249750.1 250881.9 249710.7 250231.1 250912.3 251382.0 251641.6 251645.0 252590.4
          Nov      Dec
2023 252227.4 252536.6
> 
```

Plot



Accuracy

```
> round(accuracy(tot.trend.seas.pred$fitted + residual.ar1.pred$fitted, exports.ts), 3)
              ME     RMSE      MAE    MPE  MAPE  ACF1 Theil's U
Test set -39.861 3403.284 1914.664 -0.131 1.465 0.338     0.973
> round(accuracy((snaive(exports.ts))$fitted, exports.ts), 3)
               ME     RMSE      MAE   MPE  MAPE  ACF1 Theil's U
Test set 6646.806 16902.12 12282.27 4.681 8.755 0.948     4.773
> round(accuracy((naive(exports.ts))$fitted, exports.ts), 3)
              ME     RMSE      MAE   MPE  MAPE  ACF1 Theil's U
Test set 538.817 3525.773 2000.299 0.406 1.488 0.303         1
```

So as per the accuracy measures when compare the model with seasonal naïve, navie its observed that the model is performed better with least mape-1.465, rmse-3525.7

3) <u>Holt-winter's</u>

The next technique utilized for the time series analysis is advanced exponential smoothing, more specifically the Holt-Winters model. Holt-Winter's (HW) or simply Winter's model is used for time series that contains trend and seasonality -Idea is to augment Holt's model by capturing a seasonal component. This model is ideal since it considers both the trend and seasonality components when creating forecasts.

Advanced Exponential Smoothing

The next technique utilized for the time series analysis is advanced exponential smoothing, more specifically the Holt-Winters model. This model is ideal since it considers both the trend and seasonality components when creating forecasts. Prior to running the model on the entire data set, it was first evaluated using the training and validation partitions.

Automated Holt-Winter Model (Z, Z, Z)

Ets() function uses model=ZZZ and chooses the best possible parameters for alpha,beta,gamma.

where: α = smoothing constant for exponential smoothing

β = smoothing constant for trend estimate

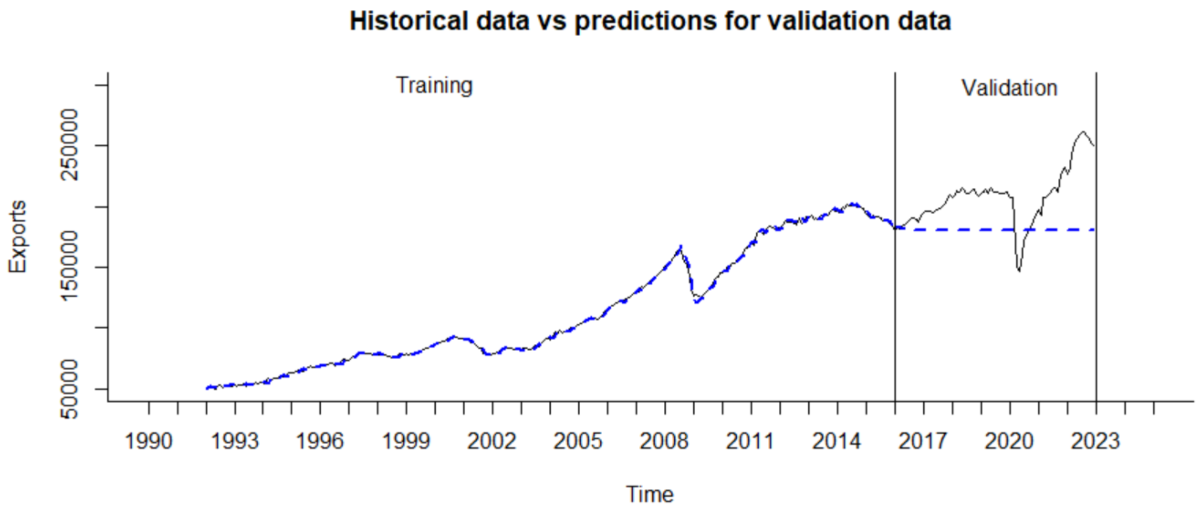
γ = smoothing constant for seasonality estimate

k = periods to be forecasted into future

M = number of seasons

```
> # Use ets() function with model = "ZZZ", to identify the best HW option
> # and optimal alpha, beta, & gamma to fit HW for the training data period.
> HW.ZZZ <- ets(train.ts, model = "ZZZ")
> HW.ZZZ
ETS(M,Ad,N)

Call:
 ets(y = train.ts, model = "ZZZ")

  Smoothing parameters:
    alpha = 0.6883
    beta  = 0.357
    phi   = 0.8

  Initial states:
    l = 49986.7929
    b = 235.2446

  sigma:  0.0173

     AIC     AICc      BIC
5954.203 5954.502 5976.181
```

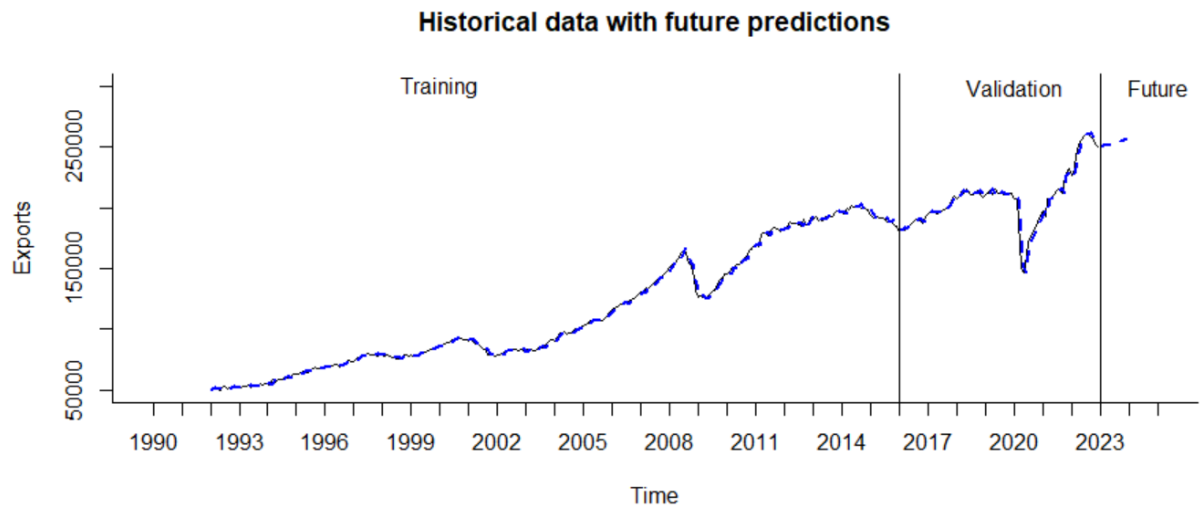
Plot for training and validation data



Holts winter Model built over entire data

```
> HW.ZZZ.entire
ETS(M,A,N)

Call:
 ets(y = exports.ts, model = "ZZZ")

  Smoothing parameters:
    alpha = 0.9999
    beta  = 0.0035

  Initial states:
    l = 49986.8268
    b = 234.5089

  sigma:  0.0221

     AIC     AICc      BIC
8081.034 8081.198 8100.628
```

Plot of holts winter model over entire data



Accuracy measure when compared Holts winter model with naïve, seasonal naïve over entire data set

```
> round(accuracy((snaive(exports.ts))$fitted, exports.ts), 3)
               ME     RMSE      MAE   MPE  MAPE  ACF1 Theil's U
Test set 6646.806 16902.12 12282.27 4.681 8.755 0.948     4.773
> round(accuracy((naive(exports.ts))$fitted, exports.ts), 3)
              ME     RMSE      MAE   MPE  MAPE  ACF1 Theil's U
Test set 538.817 3525.773 2000.299 0.406 1.488 0.303         1
> round(accuracy(HW.ZZZ.entire.pred$fitted, exports.ts), 3)
              ME     RMSE      MAE   MPE  MAPE  ACF1 Theil's U
Test set 201.165 3489.024 1945.499 0.129 1.444 0.302     0.982
```

So, for the entire data Holt's winter model gives the least MAPE and RMSE values which means the Holt's winter model gives us the best model as of now.

4) Auto.arima() for training data:

auto.arima() function in R is used to automatically identify ARIMA model and its respective (p, d, q)

where: p, The number of lag observations included in the model d, The degree of differencing q, The size of the moving average window.

AR= (p) value: Autoregressive and it works with linear series of variables' past values. • MA= (q) value: Moving Average and it works with a linear series of previous forecast errors. • I= (d) value: Integrated, is the differencing error between AR and MA.

Does not require to input any of these parameters into the function.

Identifies ARIMA model that is close to optimal, or, actually optimal, in terms of accuracy measures

```
Series: train.ts
ARIMA(2,1,2) with drift

Coefficients:
         ar1      ar2      ma1    ma2     drift
      1.1680  -0.4526  -1.1067  0.610  455.3430
s.e.  0.1715   0.1583   0.1517  0.122  200.1734

sigma^2 = 3771039:  log likelihood = -2577.9
AIC=5167.81   AICc=5168.11   BIC=5189.77

Training set error measures:
                   ME     RMSE      MAE         MPE     MAPE      MASE         ACF1
Training set 2.844471 1921.582 1386.397 -0.02088622 1.268683 0.1396326 -0.001426255
```

<u>ARIMA Equation:</u> (for order 1 differencing of season 12)

$$y_t - y_{t-1} = \beta_0 + \beta_1(y_{t-1} - y_{t-2}) + \beta_2(y_{t-2} - y_{t-3}) + \ldots + \varepsilon_t + \ldots + \theta_1(y_{t-1} - y_{t-13}) + \theta_2(y_{t-2} - y_{t-14}) + \ldots + \rho_{t-1} + \rho_{t-2} + \ldots$$

for seasonal ARIMA, the coefficients repeat with seasonal parameters (P,D,Q)
(p, d, q)(P, D, Q)[m]

<u>m=seasonality</u>

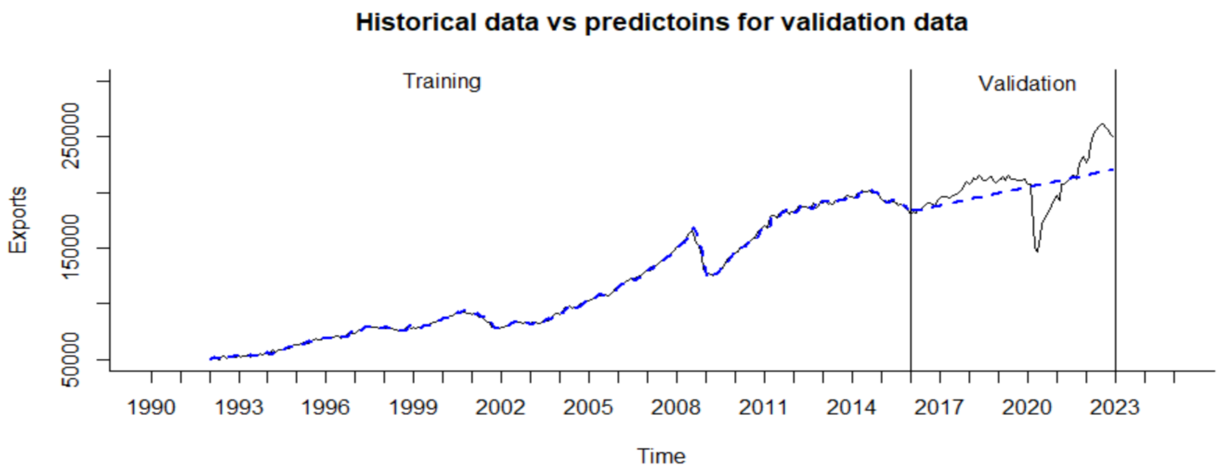
<u>In this model:</u>

ARIMA (2,1,2)

p = 2, order 2 autoregressive model AR(2)
d = 1, order 1 differencing to remove linear trend
q = 2, order 2 moving average MA(2) for error lags

$$y_t - y_{t-1} = 1.16 * (y_{t-1} - y_{t-2}) + (-0.45) * (y_{t-1} - y_{t-2}) + (-1.106) * \varepsilon_{t-1} + 0.61\ \varepsilon_{t-2} + 455.34$$

<u>Plot for training data:</u>



Auto.arima() for entire data:

```
Series: exports.ts
ARIMA(0,1,2) with drift

Coefficients:
         ma1     ma2     drift
      0.3134  0.0891  538.0686
s.e.  0.0522  0.0537  241.3155

sigma^2 = 11092663:  log likelihood = -3534.11
AIC=7076.23   AICc=7076.34   BIC=7091.89

Training set error measures:
                    ME     RMSE      MAE         MPE     MAPE      MASE         ACF1
Training set -1.181833 3312.61 1926.956 -0.05982969 1.460302 0.1568892 -0.001720371
```

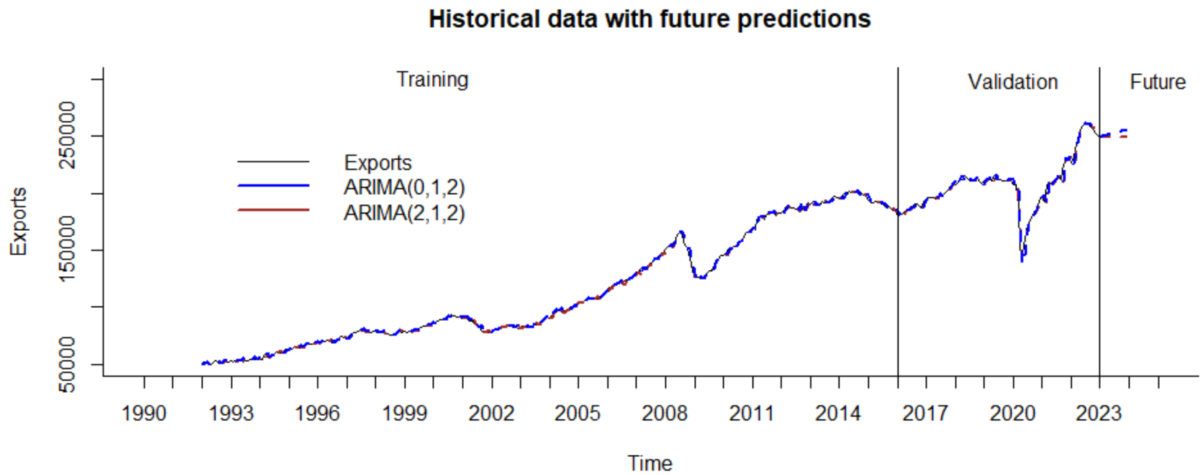
In this model:

ARIMA (0,1,2)

p = 0, no AR() for this
d = 1, order 1 differencing to remove linear trend
q = 2, order 2 moving average MA(2) for error lags

$$y_t - y_{t-1} = 0.313 * \varepsilon_{t-1} + 0.089\ \varepsilon_{t-2} + 538.06$$

Plot for future predictions:



Step 8: Implement Forecast

```
> round(accuracy(tot.trend.seas.pred$fitted + tot.ma.trail.res.pred$fitted, exports.ts), 3)
            ME      RMSE      MAE    MPE  MAPE  ACF1 Theil's U
Test set 6.073 4212.474 2260.854 -0.102 1.646 0.535     1.167
>
> round(accuracy(tot.trend.seas.pred$fitted + residual.ar1.pred$fitted, exports.ts), 3)
              ME     RMSE      MAE    MPE  MAPE  ACF1 Theil's U
Test set -39.861 3403.284 1914.664 -0.131 1.465 0.338     0.973
>
> round(accuracy(HW.ZZZ.entire.pred$fitted, exports.ts), 3)
              ME     RMSE      MAE   MPE  MAPE  ACF1 Theil's U
Test set 201.165 3489.024 1945.499 0.129 1.444 0.302     0.982
>
> round(accuracy(entire.auto.arima.pred$fitted, exports.ts), 3)
             ME     RMSE      MAE   MPE MAPE   ACF1 Theil's U
Test set -1.182 3312.61 1926.956 -0.06 1.46 -0.002     0.966
> round(accuracy(entire.212.arima.pred$fitted, exports.ts), 3)
              ME     RMSE      MAE   MPE  MAPE   ACF1 Theil's U
Test set 384.943 3333.499 1960.983 0.294 1.491 -0.012     0.981
>
> round(accuracy((snaive(exports.ts))$fitted, exports.ts), 3)
               ME     RMSE      MAE   MPE  MAPE  ACF1 Theil's U
Test set 6646.806 16902.12 12282.27 4.681 8.755 0.948     4.773
> round(accuracy((naive(exports.ts))$fitted, exports.ts), 3)
              ME     RMSE      MAE   MPE  MAPE  ACF1 Theil's U
Test set 538.817 3525.773 2000.299 0.406 1.488 0.303         1
```

Below is the table comprising of the RMSE AND MAPE Values of various models built.

Model Name	MAPE	RMSE
Twolevel(trend,seasonality),Ma trail	**1.646**	**4241.474**
Twolevel(trend,seasonality),Ar(1) for residuals	**1.465**	**3403.284**
Holts winter models	**1.444**	**3489.024**
Auto Arima	**1.46**	**3312.62**
Arima(2,1,2)	**1.491**	**3333.499**
Seasonal naive	**8.755**	**16902.12**
naive	**1.488**	**3525.773**

The best model with least MAPE(1.444) gives the Holts winter model and then AutoArima() as below:

```
> forecast.best<-forecast(HW.ZZZ.entire, h = 12 , level = 0)
> forecast.best
         Point Forecast     Lo 0     Hi 0
Jan 2023       250645.2 250645.2 250645.2
Feb 2023       251138.1 251138.1 251138.1
Mar 2023       251631.0 251631.0 251631.0
Apr 2023       252123.9 252123.9 252123.9
May 2023       252616.8 252616.8 252616.8
Jun 2023       253109.7 253109.7 253109.7
Jul 2023       253602.6 253602.6 253602.6
Aug 2023       254095.5 254095.5 254095.5
Sep 2023       254588.4 254588.4 254588.4
Oct 2023       255081.3 255081.3 255081.3
Nov 2023       255574.2 255574.2 255574.2
Dec 2023       256067.1 256067.1 256067.1
> |
```

Forecast future 12 months using AutoArima()

```
> forecast.autoarima<-forecast(entire.auto.arima, h = 12 , level = 0)
> forecast.autoarima
         Point Forecast     Lo 0     Hi 0
Jan 2023       249896.0 249896.0 249896.0
Feb 2023       250312.4 250312.4 250312.4
Mar 2023       250850.5 250850.5 250850.5
Apr 2023       251388.6 251388.6 251388.6
May 2023       251926.6 251926.6 251926.6
Jun 2023       252464.7 252464.7 252464.7
Jul 2023       253002.8 253002.8 253002.8
Aug 2023       253540.8 253540.8 253540.8
Sep 2023       254078.9 254078.9 254078.9
Oct 2023       254617.0 254617.0 254617.0
Nov 2023       255155.0 255155.0 255155.0
Dec 2023       255693.1 255693.1 255693.1
```

Conclusion

In this project we applied several techniques related to data analytics (Time series) to gain some insights on USA Exports data, We applied a various models from Regression(two-level),Holts winter, Auto Arima to study its relationship with data. This final analysis demonstrated that the ARIMA models that have minimum MAPE, RMSE are Holts winter model which comes first and second comes with the Auto.Arima() model, And Forecast for the coming 12 months of 2023 using Holts winter model finally.

Limitations of the Study:

- During this study, we identified the following limitation that need to be taken into consideration for assessment of the results and should be considered by future work.
- Limited knowledge in econometric methods and theories.
- Limited knowledge of trade theories and factors that impact trade between countries.
- This analysis is only based on historical trade data and doesn't include any other factors that impact trade between countries such as pandemics, geopolitical situations, wars, and any other unforeseen factors.

Bibliography:

https://www.census.gov/econ/currentdata/datasets/?programCode=FTD&startYear=1992&endYear=2023&categories[]=BOPGS&dataType=BAL&geoLevel=US&adjusted=1&notAdjusted=0&errorData=0

Appendices (Used for Reference)

https://www.analyticsvidhya.com/blog/2015/12/complete-tutorial-time-series-modeling/