

**Forecasting Assignment**

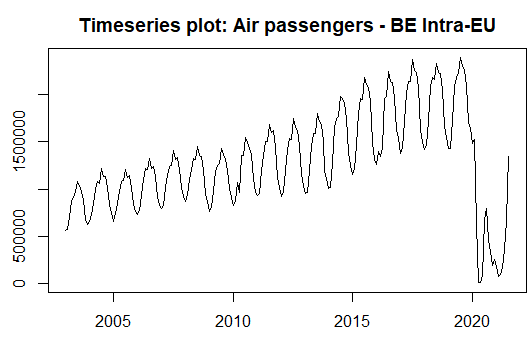
**Nour Azar**

**Exercise 1:**

This report analyses recent data on air transport in the European Union, more specifically Intra-EU transports which refers to transactions within the EU only. This data set takes into account the passengers that were carried on air transport between Belgium and other EU countries from January 2003 till October 2021.

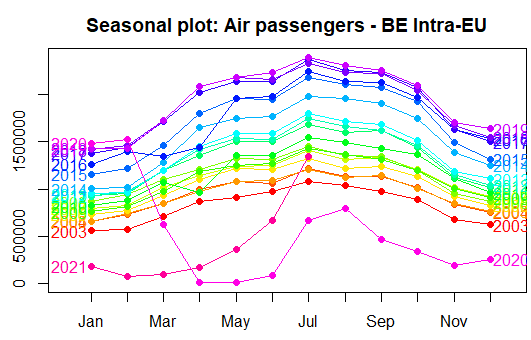
**Question 1:**

The graph below shows the time series plot of the whole data set before splitting it into test and train.



We can clearly see the upward trend in the data from 2003 till beginning of 2020.

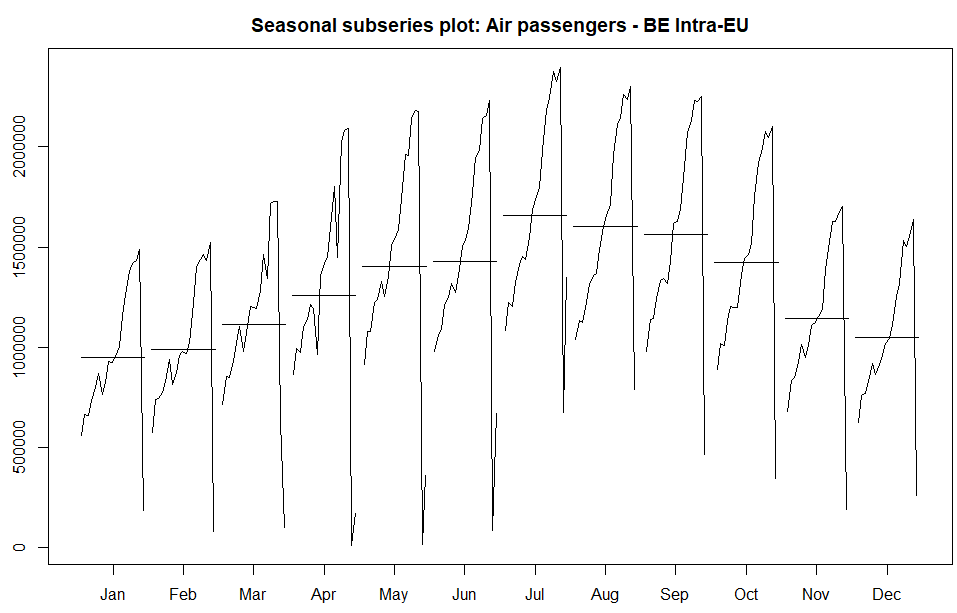
The graph below shows the seasonal pattern of our data, where a season is a month:



There is an obvious seasonal pattern in all years except for years 2020 and 2021. The number of passengers usually reaches its lowest point in January and then slightly increases and sometimes stays the same in February, and then we notice that after February numbers start rising in a sharper way until April, followed by a slight increase in May and another one in June, and then a steep increase from June till July where the number of passengers reaches its peak. Then the number of passengers start decreasing until December, with a steep decrease between October and November.

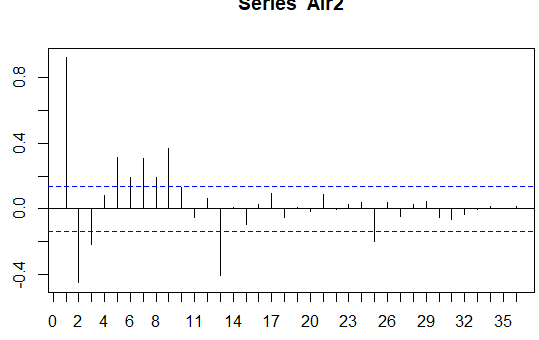
There is a clear substantial departure from the observed seasonal patterns in years 2020 and 2021, and the reason for that is most probably related to the travel restrictions due to covid.

The graph below shows the seasonal subseries of our data, helping us better visualize the changes in seasonality over time:



We notice that the same interpretation for the seasonal plot applies to this graph, since the only difference between the two is that the data of each season is aggregated. Note that the straight line represents the mean of the observations in that season.

**Partial autocorrelation graph**

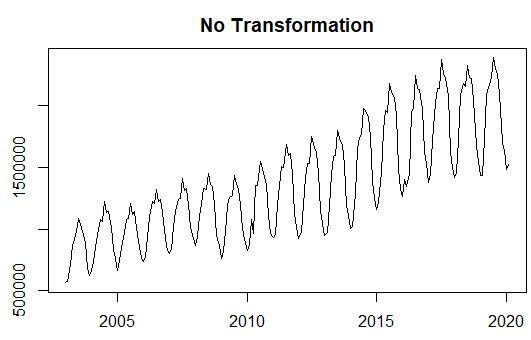
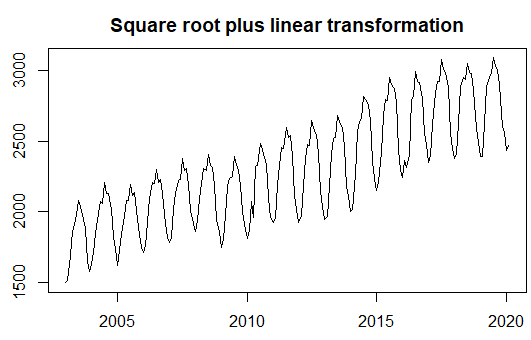


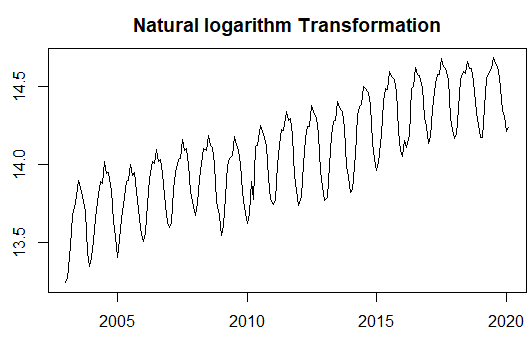
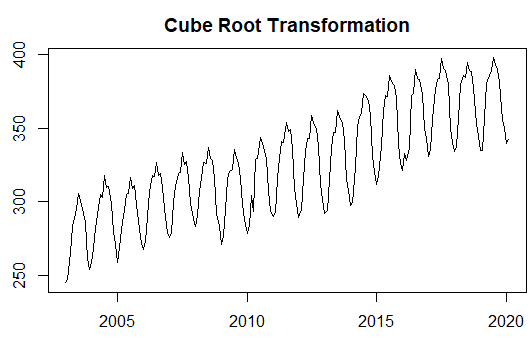
On lag 1 and lag 13 which represents the first month of the year, there is a clear autocorrelation of residuals.

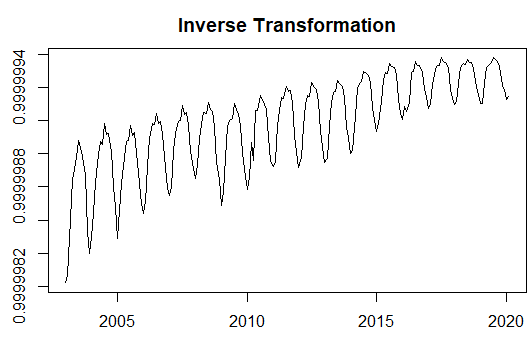
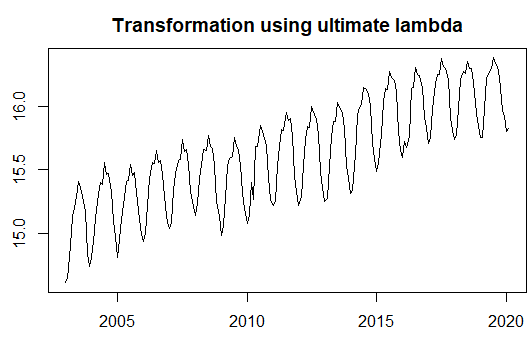
**Question 2:**

Transforming the data is useful when the data contains variations that increase and decrease across different levels of the timeseries. Many transformation methods can be used to reduce the variability of the data, like square root transformation, cube root transformation, inverse transformation, and natural logarithm transformation. In general, the logarithm transformation is preferred over the others because they are interpretable since changes in the original scale are relatively similar to the changes in the log scale.

The graphs below show the different outcomes from the different methods used:

We tranformed the data in R by using the BoxCox() function and changing the lambda value: 1 for no transformation, ½ for square root method, 1/3 for cube root method, -1 for inverse method, and 0 for log transformation. Finally, we used the function BoxCox.lambda() to find the ultimate lambda value, meaning “the value that makes the size of the seasonal variation about the same accross the whole series” (source: Professor Van Den Bossche slides), and we found that the ultimate lambda equals to 0.01461759 (plotted in the bottom right graph). We save the ultimate lambda in a variable, because when we want to forecast we want to change the data back to its original scale.

**Question 3:**

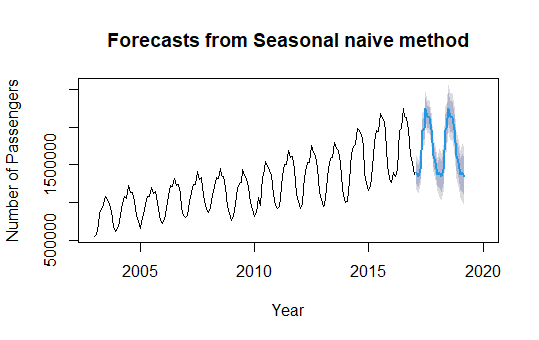
Now that we transformed our data, we will start forecasting with the seasonal naïve method where the forecasts are equal to the last observed value of the same season, as shown in the equation below:

Where:

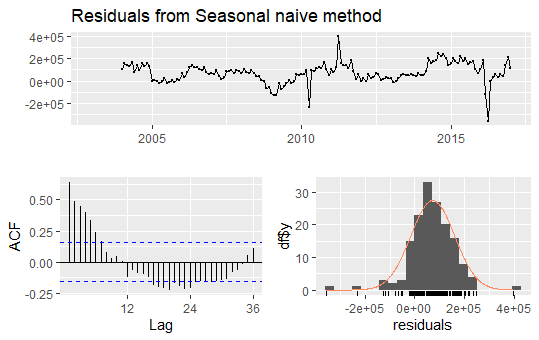
* m is the seasonal period
* h is the length of the period we want to forecast
* k equals to

This method is useful for highly seasonal data, which is the case of our data.

The graph below plots the Seasonal Naïve forecast on our data using R:



In order to check the residuals, we ran the function checkresiduals() on the model and got the following graphs:



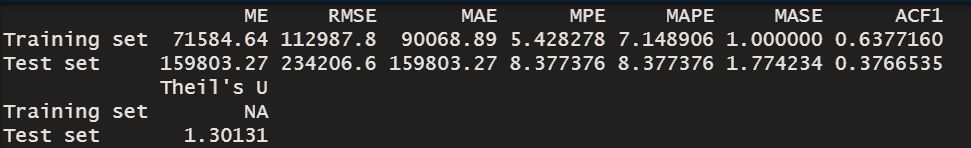
Residuals are the difference between the observations and their fitted value, so the less the residuals the better the model. Moreover, we want the absolute autocorrelation coefficient of the residuals to be low, implying that the residuals are white noise. However, in the ACF graph we see that the autocorrelation is high since it goes beyond the blue line which represents the significance threshold, meaning that the residuals are not white noise and our model is not fully adequate.

In addition to plotting the residuals graph, we did the Ljung-Box test on the residuals, which tests the overall randomness based on the number of lags (source: <https://en.wikipedia.org/wiki/Ljung%E2%80%93Box_test>). Moreover, the Ljung-Box test uses the two hypothesis according to statology.org (<https://www.statology.org/ljung-box-test/> ). The first, , hypothesizes that the residuals are independently distributed, and the second hypothesizes that the residuals are not independently distributed and they exhibit serial correlation. If the p-value is less than 0.05 then hypothesis applies, otherwise applies. When we ran the Ljung-Box test on the residuals of our model we got the following output:



This indicates that applies here since the p-value is less than 0.05, which confirms our interpretation of the ACF graph above that the residuals are not whote noise.

We also checked the forecast accuracy by using the function accuracy() in R on the test set, and we got the following output:



We will interpret the forecast accuracy by looking at the MASE score, but first we will give a brief explanation of what MASE score indicates.

According to Towardsdatascience (<https://towardsdatascience.com/time-series-forecast-error-metrics-you-should-know-cc88b8c67f27>), the MASE is calculated by taking the MAE (mean average error) and dividing it by the MAE of an in-sample (training set) naive benchmark, and values of MASE greater than 1 indicate that the forecasts are worse, on average, than in-sample one-step forecasts from the naive model (Hyndman and Koehler, 2006). In our case the MASE of the test set is greater than 1, meaning our forecasts are not good.

**Question 4:**

After finding that the seasonal naïve model is not adequate for our data, we now explore new models using STL decomposition: naive, rwdrift, ets and arima.

After running these models, we got the following accuracy and residual scores:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Accuracy scores** | | | | **Residuals** | | |
|  | **RMSE** | **MAE** | **MAPE** | **MASE** | **Q\*** | **df** | **p-value** |
| **Seasonal naive** | 234206.6 | 159803.3 | 8.377376 | 1.774234 | 263.4990082 | 24 | 0.000000000 |
| **Naïve** | 87801.32 | 66011.48 | 3.214178 | 0.7329 | 36.50716928 | 24 | 0.048968091 |
| **rwdrift** | 271836 | 242415.2 | 12.76524 | 2.691442 | 36.50716928 | 23 | 0.036589185 |
| **ets** | 234193.1 | 204316.2 | 10.69065 | 2.268444 | 18.45886235 | 20 | 0.557208465 |
| **arima** | 202454.6 | 172468.4 | 8.957144 | 1.914849 | 14.22877152 | 21 | 0.85957339 |

The best model is the Naïve model since it’s MASE score is the closest to zero, however in this model we reject the null hypothesis since the p-value of the residuals is lower than 0.05, although the p-value is not so bad since it is only slightly lower than 0.05.

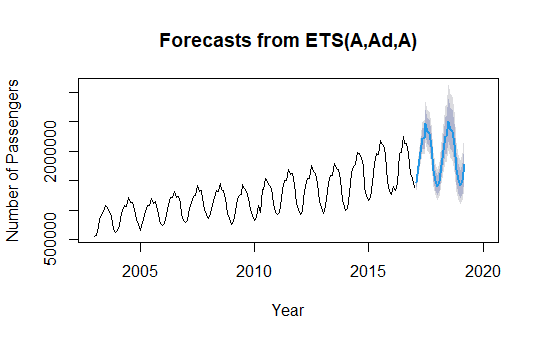
**Question 5:**

In addition to the STL decomposition models, we ran new models using ETS. We selected 4 ETS models, each one with different parameters and we got the following accuracy and residuals scores:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Accuracy** | | | | **Residuals scores** | | |
|  | **RMSE** | **MAE** | **MAPE** | **MASE** | **Q\*** | **df** | **p-value** |
| **AAN** | 1627677 | 1590550 | 85.72481 | 17.65926 | 230.3991 | 20 | 0.00000 |
| **AAdN** | 633118.6 | 539203.3 | 26.83195 | 5.986565 | 765.2903 | 19 | 0.00000 |
| **AAA** | 172591.3 | 148480.8 | 7.783813 | 1.648525 | 36.17085 | 8 | 0.00000 |
| **AAdA** | 90432.64 | 72662.85 | 3.648737 | 0.806747 | 12.73963 | 7 | 0.078712 |

We also ran the model without specifying any parameters to know what is the ultimate model, and it turned out that the ultimate model is ‘AAdA’, with a MASE that is less than 1 and residuals p-value above 0.05.

We got the following result when we plotted the forecast for model AAdA:



**Question 6**

Now we will run ARIMA models, each one with different paramters. The table below illustratesour results:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Accuracy** | | | | **Residuals scores** | | |
|  | **RMSE** | **MAE** | **MAPE** | **MASE** | **Q\*** | **df** | **p-value** |
| **Arima1** | 101305.7 | 76213.72 | 4.256482 | 0.846171 | 14.1148 | 19 | 0.776958 |
| **Arima2** | 169714.9 | 145650.9 | 7.640211 | 1.617105 | 4.777818 | 18 | 0.999165 |
| **Arima3** | 122421.7 | 106686.1 | 6.134022 | 1.184494 | 19.77007 | 21 | 0.535864 |
| **Arima4** | 257250.9 | 233700.6 | 12.39897 | 2.594687 | 10.98092 | 20 | 0.946716 |

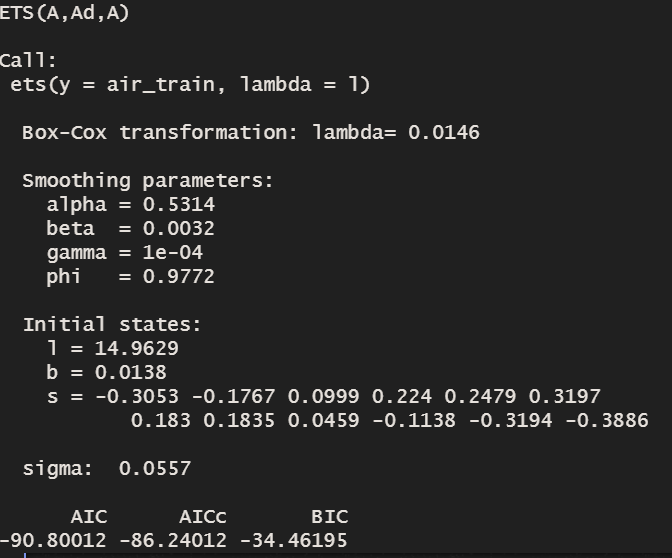
The results shown in the graph show that the first Arima model is the best, which is the model in which we did not specify the paramters: lambda, d and D. The MASE value of the first one is lowest with a p-value greater than 0.05 indicating that the residuals are white noise.

**Question 7:**

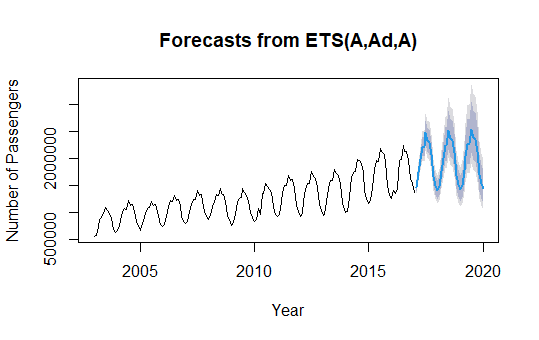
The following table shows all the results of all the models that we ran so far:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Accuracy Scores** | | | | **Residuals scores** | |
|  | **RMSE** | **MAE** | **MAPE** | **MASE** | **Q\*** | **p-value** |
| **Seasonal naive** | 234206.6 | 159803.3 | 8.377376 | 1.774234 | 263.499 | 0 |
| **stl\_Naïve** | 87801.32 | 66011.48 | 3.214178 | 0.7329 | 36.50717 | 0.048968 |
| **stl\_rwdrift** | 271836 | 242415.2 | 12.76524 | 2.691442 | 36.50717 | 0.036589 |
| **stl\_ets** | 234193.1 | 204316.2 | 10.69065 | 2.268444 | 18.45886 | 0.557208 |
| **stl\_arima** | 202454.6 | 172468.4 | 8.957144 | 1.914849 | 14.22877 | 0.859573 |
| **ets\_AAN** | 1627677 | 1590550 | 85.72481 | 17.65926 | 230.3991 | 0 |
| **ets\_AADN** | 633118.6 | 539203.3 | 26.83195 | 5.986565 | 765.2903 | 0 |
| **ets\_AAA** | 172591.3 | 148480.8 | 7.783813 | 1.648525 | 36.17085 | 1.63E-05 |
| **ets\_AADA** | 90432.64 | 72662.85 | 3.648737 | 0.806747 | 12.73963 | 0.078712 |
| **auto\_ets** | 80952.6 | 65071.37 | 3.271523 | 0.722462 | 12.73963 | 0.078712 |
| **arima1** | 101305.7 | 76213.72 | 4.256482 | 0.846171 | 14.1148 | 0.776958 |
| **arima2** | 169714.9 | 145650.9 | 7.640211 | 1.617105 | 4.777818 | 0.999165 |
| **arima3** | 122421.7 | 106686.1 | 6.134022 | 1.184494 | 19.77007 | 0.535864 |
| **arima4** | 257250.9 | 233700.6 | 12.39897 | 2.594687 | 10.98092 | 0.946716 |

After studying carefully the results of every model, we conclude that the best model in terms of MASE and p-value is the auto\_ETS model, in which we only specified one parameter, lambda, and let the model find the ultimate parameters for a best performing model. Below is the details of the models we chose:

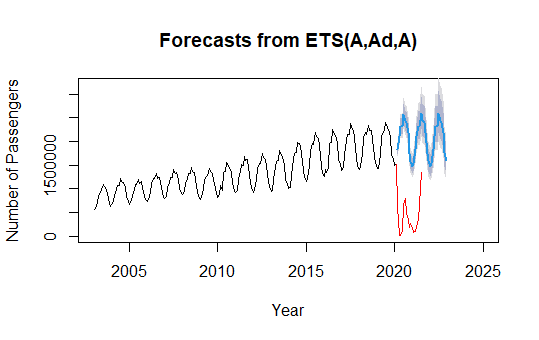


**Question 8:**



The graph above presents the generated out of sample forecasts up to December 2022, where the blue part represents the forecast from March 2020, and the grey shadows presents the confidence interval.

**Question 9:**



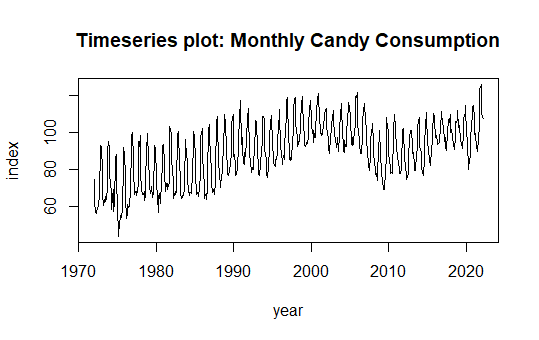
Thanks to our best model, we are now able to capture the genuine pattern of our data, which is the main purpose of forecasting. However sometimes unexpected events happen causing drastic differences between the expected and actual outcomes. In our case, the unexpexted event is the Covid pandamic that resulted in numerous travel restrictions in the EU and consequently in a drop of air passengers in the beginning of year 2020. However we can see that the red line is getting closer with time to the blue line, which indicates that perhaps, the data will again follow the observed patterns shown by our model.

**Exercise 2:**

For the second exercise, we chose a dataset that lists the production index of candy for each month, from January 1972 till February 2022. Industries could use this forecast to anticipate the high demand and prepare adequate candies stock accordingly.

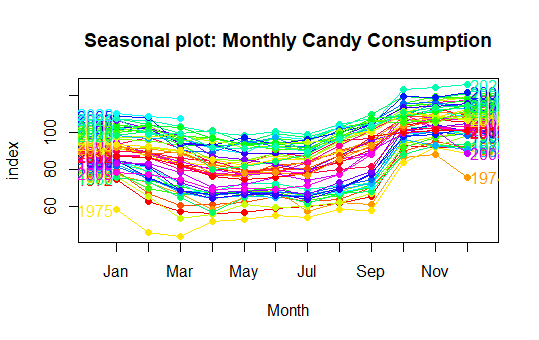
**Data Exploration:**

The graph below shows the time series plot of the whole data set before splitting it into test and train:



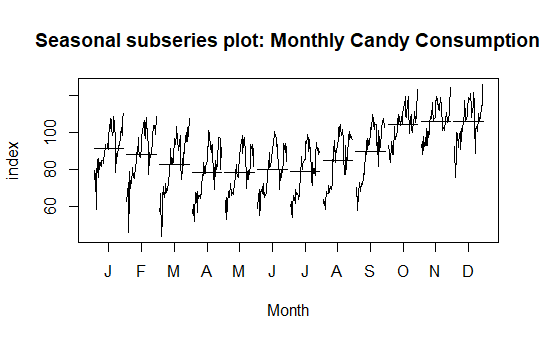
We can clearly see a horizontal trend in the data from 1972 till beginning of 2022.

The graph below shows the seasonal pattern of our data, where a season is a month:



There is an obvious seasonal pattern in our data. The index usually reaches its lowest points between April and July and its highest points in October, November, and December.

The graph below shows the seasonal subseries of our data, helping us better visualize the changes in seasonality over time:



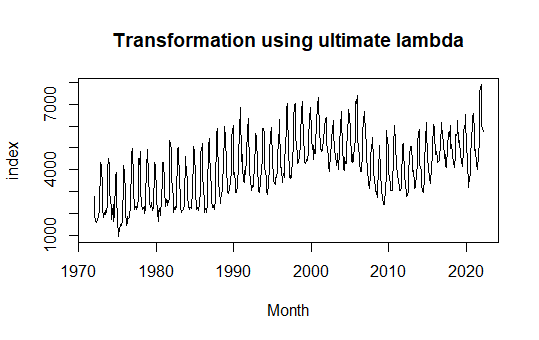
We notice that the same interpretation for the seasonal plot applies to this graph, since the only difference between the two is that the data of each season is aggregated.

**Splitting the data**

We split the data into train and test, the train set till 2014 and the test set from 2015.

**Data Transformation**

We calculated the ultimate lambda using the BocCox.lambda() function on the whole set and we got a value of 1,999924. The following graph illustrates our data after transforming it:



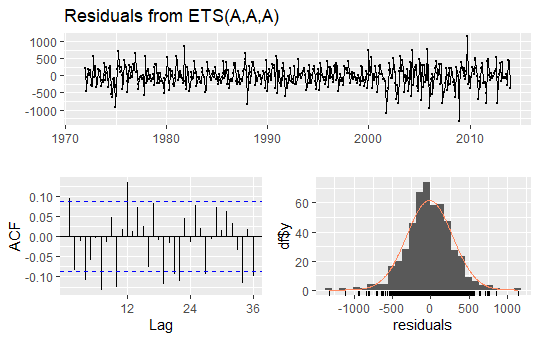
**Modeling:**

We will now start forecasting using the same models as exercise 1, and finally we will pick our best model.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **RMSE** | **MAE** | **MAPE** | **MASE** | **Q\*** | **df** | **p-value** |
| **Seasonal naive** | 9.692289 | 8.614038 | 8.753245 | 1.908283 | 1011.328 | 24 | 0.000000 |
| **stl\_Naïve** | 11.8688 | 10.77946 | 10.96948 | 2.387994 | 245.5086 | 24 | 0.000000 |
| **stl\_rwdrift** | 10.16474 | 9.060766 | 9.221743 | 2.007247 | 245.5086 | 23 | 0.000000 |
| **stl\_ets** | 9.228871 | 8.131843 | 8.254642 | 1.801461 | 118.1015 | 22 | 0.000000 |
| **stl\_arima** | 9.229185 | 8.132141 | 8.254948 | 1.801527 | 117.9685 | 23 | 0.000000 |
| **ets\_AAN** | 8.189521 | 6.591587 | 6.754303 | 1.460246 | 1329.29 | 20 | 0.000000 |
| **ets\_AADN** | 10.86796 | 8.91048 | 8.670539 | 1.973954 | 1332.664 | 19 | 0.000000 |
| **ets\_AAA** | 5.939931 | 4.949291 | 5.065048 | 1.096425 | 74.24543 | 8 | 0.000000 |
| **ets\_AADA** | 9.621043 | 8.537812 | 8.671168 | 1.891396 | 74.02664 | 7 | 0.000000 |
| **ets\_auto** | 8.197825 | 7.096491 | 7.198588 | 1.572098 | 84.16622 | 10 | 0.000000 |
| **arima1** | 9.542993 | 8.524938 | 8.648561 | 1.888545 | 47.06907 | 19 | 0.000349 |
| **arima2** | 9.392893 | 8.338402 | 8.461988 | 1.847221 | 52.54309 | 19 | 0.000055 |
| **arima3** | 12.7833 | 11.88388 | 12.04034 | 2.632658 | 60.79203 | 19 | 0.000003 |
| **arima4** | 14.08692 | 13.08692 | 13.3061 | 2.899168 | 56.55479 | 19 | 0.000013 |

The model that resulted in the lowest MASE score is **ets\_AAA,** however the p-value of the residuals is below 0.05 meaning the residuals are not white noise. We will pick this model since also the p-values of the rest of the models are also below 0.05.

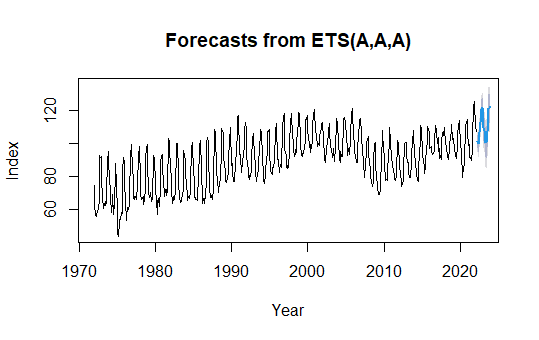
We ran the function checkresiduals() on our preferred model and got the following graphs:



In the ACF graph we see that the autocorrelation is high since it goes beyond the blue line which represents the significance threshold, meaning that the residuals are not white noise and our model is not fully adequate. However since as menntioned, the MASE score is relatively low therefore among the other models, this model is the best one.

**Forecast**

We forecasted until end of 2022 using our preferred model ETS(A,A,A) and got the following results.

This

Our forecast will help industries anticipate the high demand of candies so that the candy stocks are enough to fulfill people’s demand.

**REFERENCES**

<https://en.wikipedia.org/wiki/Ljung%E2%80%93Box_test>

<https://www.statology.org/ljung-box-test/>

<https://towardsdatascience.com/time-series-forecast-error-metrics-you-should-know-cc88b8c67f27>

Van de Bossche, F. (2022). Forecasting. [Course].Lille: IESEG Management School. MSc in Big Data  
Analytics.