STAT 515 Final Project
Team BALY

Norway Car Market Analysis

Amit Brahmbhatt Yufei Liu

Abstract:

Norway as a nation has leading number of hybrid and electric cars compared to all other countries of the world, with Oslo recognized as EV Capital of the world. It has the most number of per capita electric cars, as the numbers indicate that 135000 plug-in electric vehicles were registered in Norway by December 2016. Norway also leads in reduction of Avg. Co2 Emissions over the last decade. From the Relevant data, we found on Kaggle – supplied by www.ofvas.no – we intend to analyze details of the factors that might have contributed to growth of electric vehicles in Norway. We have used Linear Regression and Exponential smoothing methods to analyze the available data. Also, we used various plotting methods available with ggplot to understand popularity of different car makes and models.

Problem Description

Today, Norway is the country with the most number of electric car per capita. Out of 5.2 million populations in this country, more than 100,000 people have electric cars. Of all new car sales, the sales proportion of electric cars and hybrid cars constitutes 40.2%. In 2016, 12 out of the 15 most popular cars sold in Norway were electric cars and hybrid cars. Now, Norway has become the leading market in the development of electric cars and CO_2 emission reduction in the world. This amazing achievement of Norway motivated us to explore the development of electric cars.

Norway has been developing electric cars since 1970, and it helped to push the electric car revolution. In the beginning, Norway's government focused on developing their own national electric cars. Because of the electric cars' technological problems, the national electric car companies were acquired or merged by foreign car brands. For achieving its goals of CO_2 emission reduction and developing electric cars, Norway's government began to cooperate with foreign car companies by importing a great number of electric cars. Thus, the price of electric cars and hybrid cars declined sharply, and foreign brands access dramatically prompted the development of electric car market.

One important reason that we cannot ignore is Norway's government vigorous supportive policy. When people in Norway buy a new electric car or hybrid car, they don't need to pay tax and registration fees. People in Norway who have electric cars or hybrid cars also don't have to pay parking fees, road tolls, ferry fees, charge fees, and any other fees. By 2016, a total of 5600 charge stations are available in Norway. Electric cars also can be driven on the bus lane. However, bensin-fueled cars and diesel-fueled cars must pay the tax and all previously mentioned fees. These policies greatly have promoted people to buy and drive electric cars.

Fortunately, we found a dataset about the new car sales in Norway on Kaggle. We will use this dataset to explore the sales of new cars and electric cars by R studio. However, Norway is an oil producing and exporting country, so they do not need to rely on electric cars and hybrid cars due to their abundance of oil resources. We are planning to marge another dataset about Norway's oil export into the dataset of "Norway's new car sales by month". We will explore the information in the datasets by data visualization in Tableau and use multiple linear regression models in RStudio to predict different variables for the future, such as new car sales quantity, imported car sales quantity, electric car sales quantity, hybrid car sales quantity. We will also explore Norway's contribution of CO_2 emission reduction to the world.

Data Description

We will mainly use one dataset named "New Car Sales in Norway" to do our project, we will also use the dataset named "Exports of Oil and Gas in Norway" to help us supplement the first dataset. In the dataset "New Car Sales in Norway", we have 3 datasheets containing data from Norway New car sales. Originally this data comes from OFV , Norway (OFV, 'Information Council for the Road Traffic') (https://en.wikipedia.org/wiki/Opplysningsr%C3%A5det_for_Veitrafikken). We got the datasets from kaggle for this.

The data below clearly show the basic information and variables in each datasheet of this dataset:

1), First datasheet: Monthly sales of new passenger cars by make (manufacturer brand) - norway_new_car_sales_by_make.csv

- Year - year of sales
- Month - month of sales
- Make - car make (e.g. Volkswagen, Toyota, Tesla)
- Quantity - number of units sold
- Pct - percent share in monthly total

2), Second datasheet: Monthly summary of top-20 most popular models (by make and model) - norway_new_car_sales_by_model.csv

- Year - year of sales
- Month - month of sales

- Make - car make (e.g. Volkswagen, Toyota, Tesla)
- Model - car model (e.g. BMW-i3, Volkswagen Golf, Tesla S75)
- Quantity - number of units sold
- Pct - percent share in monthly total

3), Third datasheet: Summary stats for car sales in Norway by month - norway_new_car_sales_by_month.csv

- Year - year of sales
- Month - month of sales
- Quantity - total number of units sold
- Quantity_YoY - change YoY in units
- Import - total number of units imported (used cars)
- Import_YoY - change YoY in units
- Used - total number of units owner changes inside the country (data available from 2012)
- Used_YoY - change YoY in units
- Avg_CO2 - average CO_2 emission of all cars sold in a given month (in g/km)
- Bensin_CO2 - average CO_2 emission of bensin-fueled cars sold in a given month (in g/km)
- Diesel_CO2 - average CO_2 emission of diesel-fueled cars sold in a given month (in g/km)
- Quantity_Diesel - number of diesel-fueled cars sold in the country in a given month
- Diesel_Share - share of diesel cars in total sales (Quantity_Diesel / Quantity)
- Diesel_Share_LY - share of diesel cars in total sales a year ago
- Quantity_Hybrid - number of new hybrid cars sold in the country (both PHEV and BV)
- Quantity_Electric - number of new electric cars sold in the country (zero emission vehicles)
- Import_Electric - number of used electric cars imported to the country (zero emission vehicles)

Another dataset we are also planning to use comes from : http://www.norskpetroleum.no/en/production-and-exports/exports-of-oil-and-gas/

Our statistical model will mainly focus on data from New Car Sales. Other datasets would be used to supplement the findings.

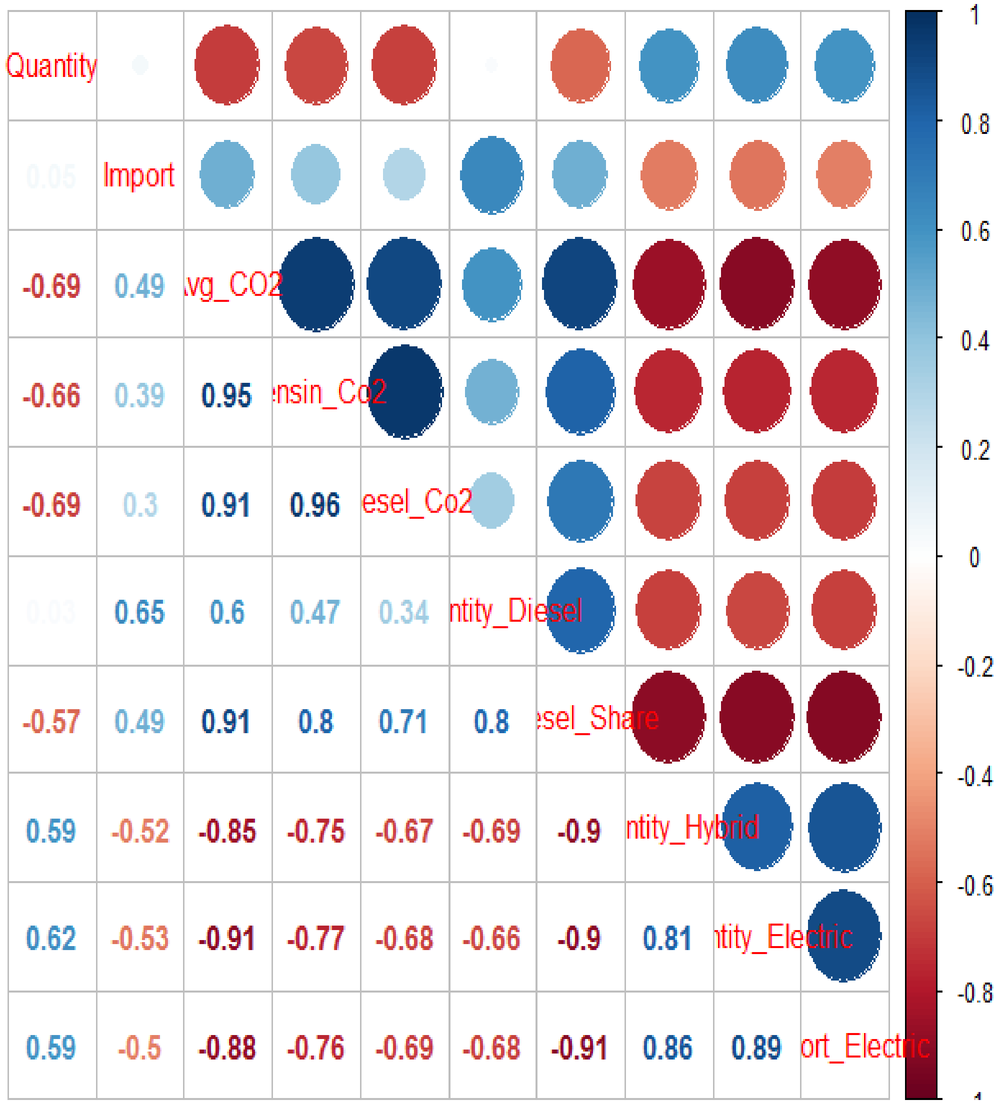
Data Collection:

We started with 3 csv files from Kaggle/ofvas.no and 1 excel sheet from http://www.norskpetroleum.no/en/production-and-exports/exports-of-oil-and-gas/.
However, after analyzing Gas and Oil production data and its correlation plots with car sheets, we decided to work with only original data sheets.

Data Cleaning and Data Preparation:

While preparing data for the analysis – we used statistical method to fill out the NA values in the sheets using a simple excel formula that retains patterns over 12-month period. After populating this values, we confirmed the pattern retention using various plots. Column populated with new data were: Import Electric, Quantity Electric, Quantity Hybrid and Used Cars. Used Cars numbers prior to 2011 December up until 2007 January were populated from data after 2012 January. This also helped populate values for Used YoY – From December 2012 to January 2008. These values however, were not used in the analysis, but confirmed the pattern retention. Similarly Import Electric values were populated for January 2007 to August 2012. Quantity Electric and quantity hybrid were produced from January 2007 to December 2010. Formulas Used for each one of these are part of Appendix A. These populated missing values in an ascending order for each year and month. After populating these values, we joined the Oil data table with our sheet and confirmed that no useful predictors were available in the oil sheet by plotting a scatter plot. We plotted scatter charts to confirm this. We plotted a correlation graph to find out which columns were positively or negatively correlated to which other columns. This helped us determine the direction to go in with existing data. Obvious choice for this analysis was Linear regression, as there were no categorical variables within the dataset. We then also determined that using exponential smoothing can help us plot trends of some of the dataset columns.

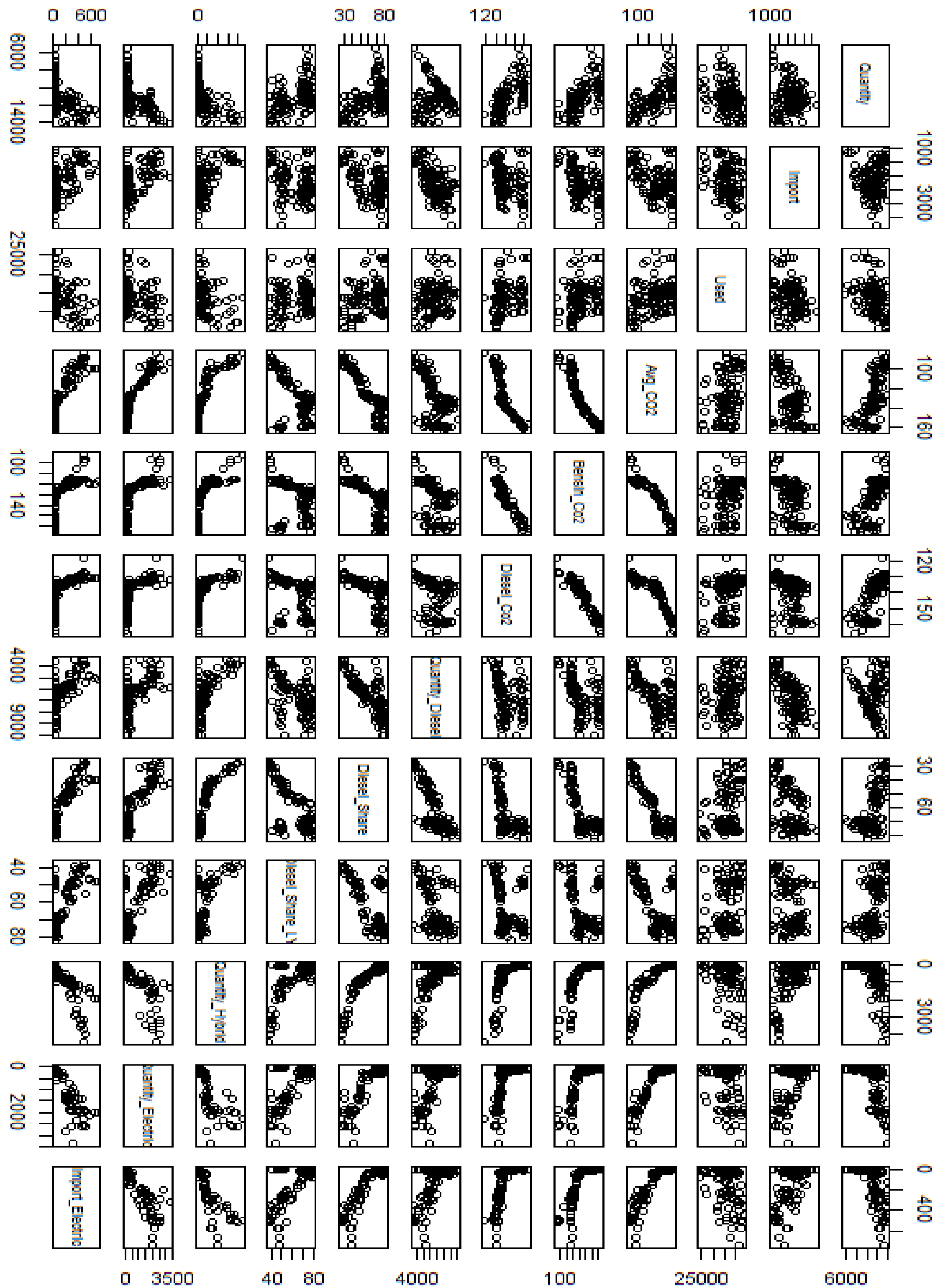
Correlation Plots with each predictor.



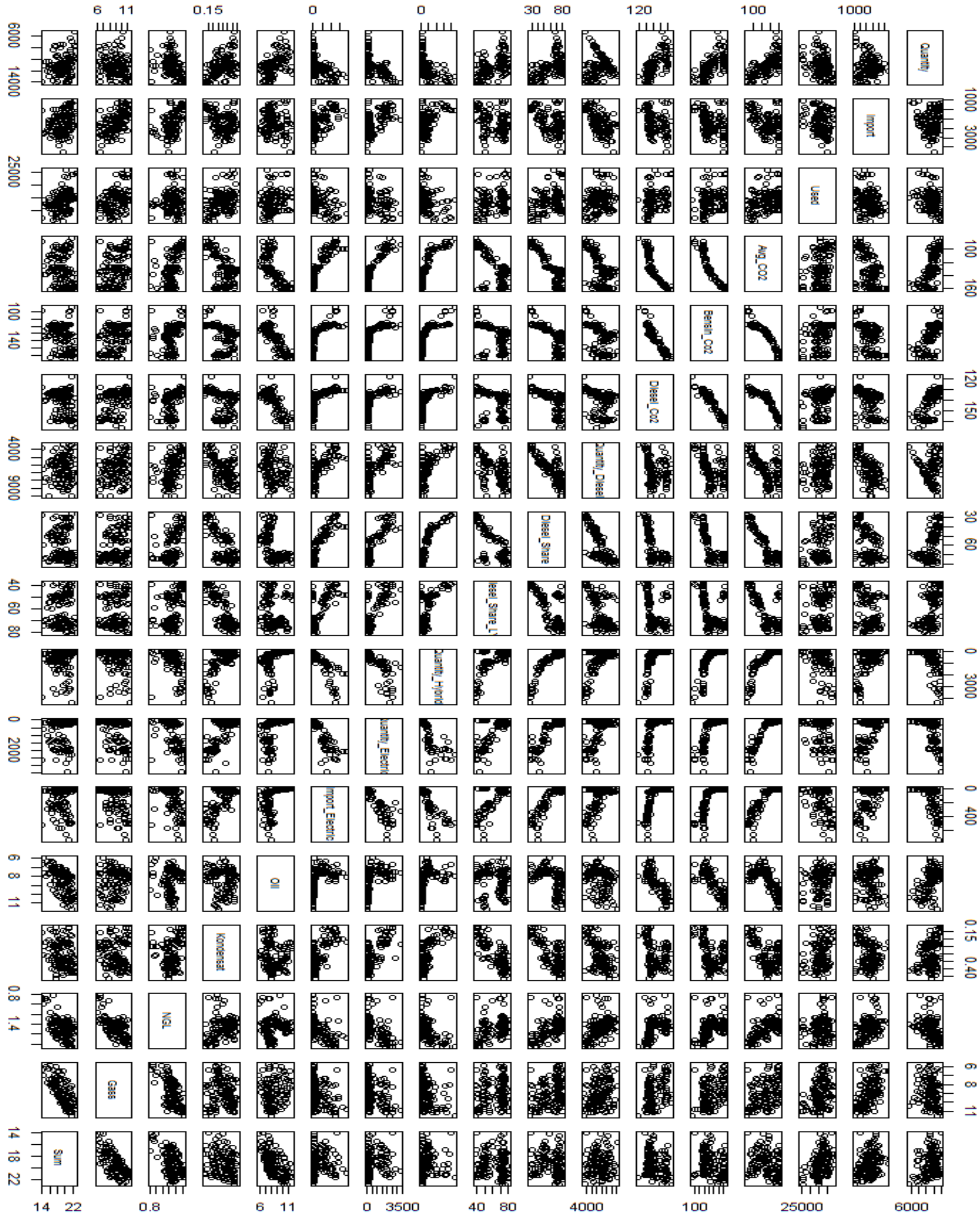
From the Correlation plot, above – we can infer few things such as:

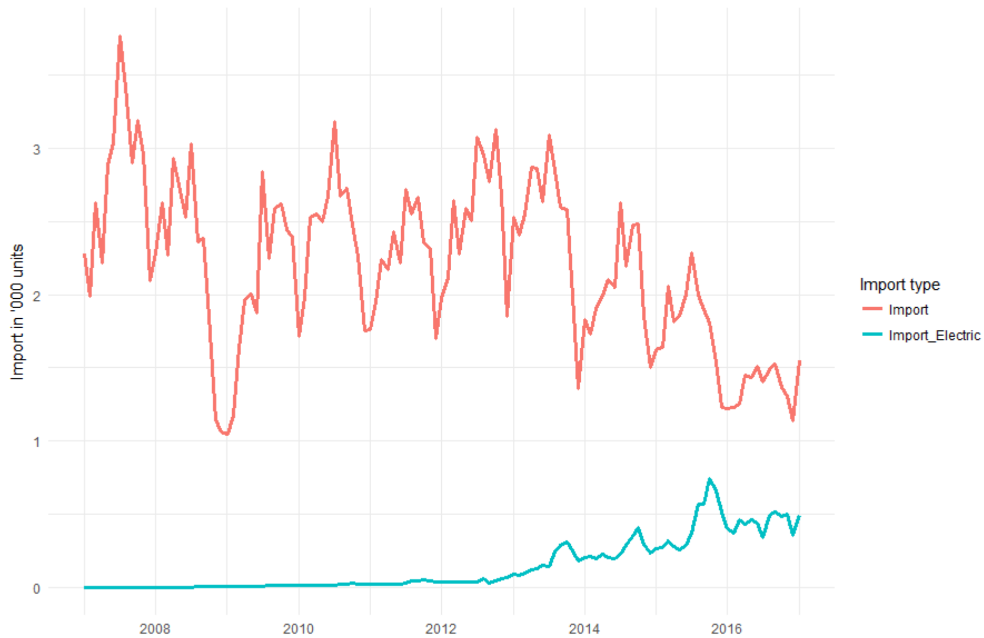
1. All Co2 columns are highly positively correlated to each other.
2. All Green Vehicles are also highly positively correlated to each other.
3. All Environmentally Friendly Vehicles are highly negatively correlated to Avg Co2.
4. Diesel Share values are highly positively correlated to Avg. Co2 values.

More information on this Correlation analysis is provided in the Appendix.

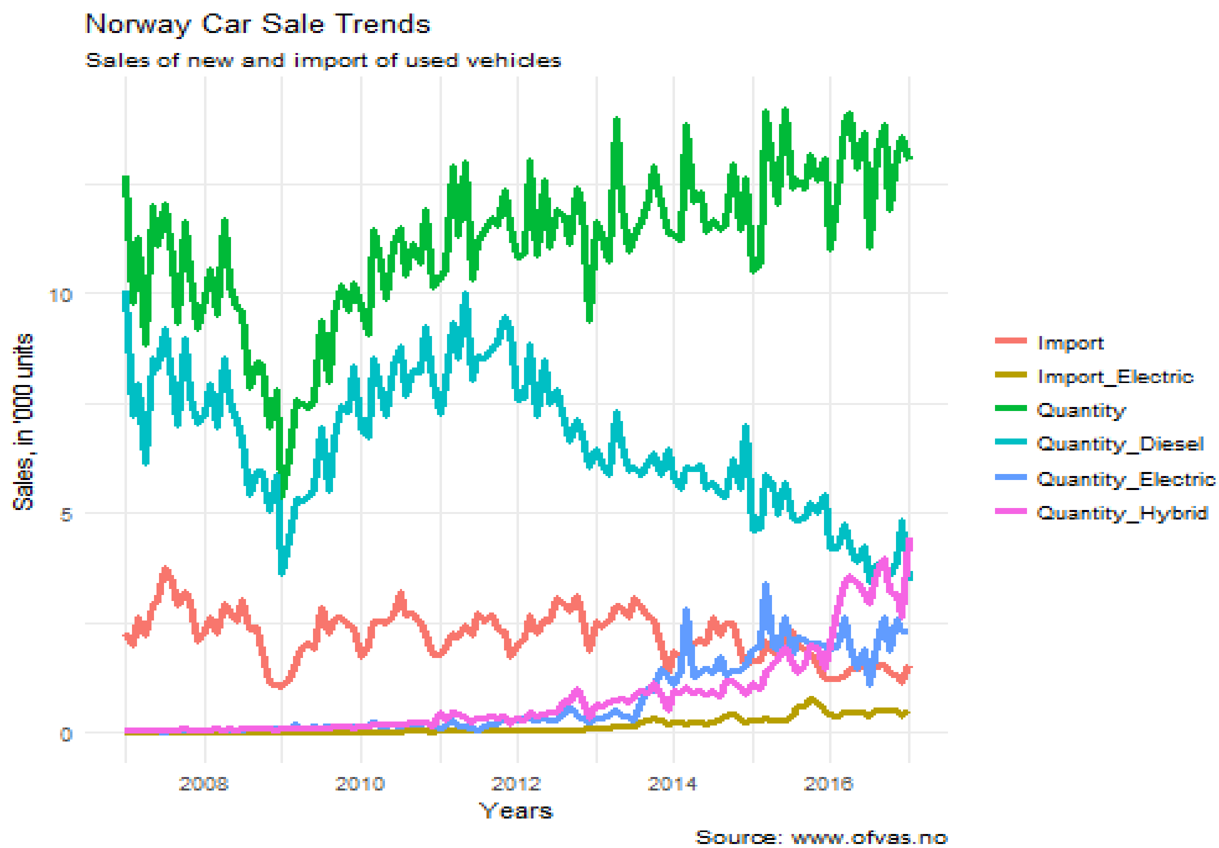


Correlation Plot with Oil Data addition shows that this data is not adding any additional value to the dataset we already have. Hence, we continued our analysis with original dataset without oil data.





This graph shows comparison between Imported cars and Imported Electric Cars. We can see a sudden hike in Import electric in recent years. Likewise, in the next graph we can see how green vehicles such as Import Electric, Electric and Hybrid cars are taking over the market.



We also notice a declining trend of Quantity Diesel vehicles where total Quantity of vehicles increase.

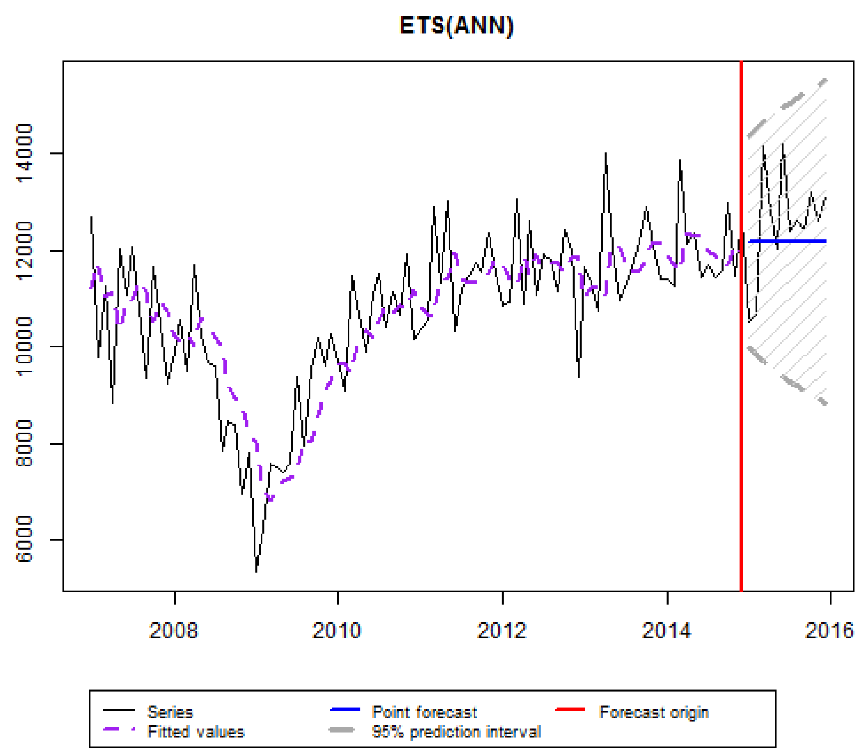
Time Series and Exponential Smoothing Analysis

For this part of our analysis we used these packages : tidyverse, forcats, smooth.

`library(tidyverse)` `contains` `and` will load the following core tidyverse packages:

ggplot2, for data visualisation, dplyr, for data manipulation, tidyr, for data tidying, readr, for data import, purrr, for functional programming, tibble, for tibbles, a modern re-imagining of data frames.

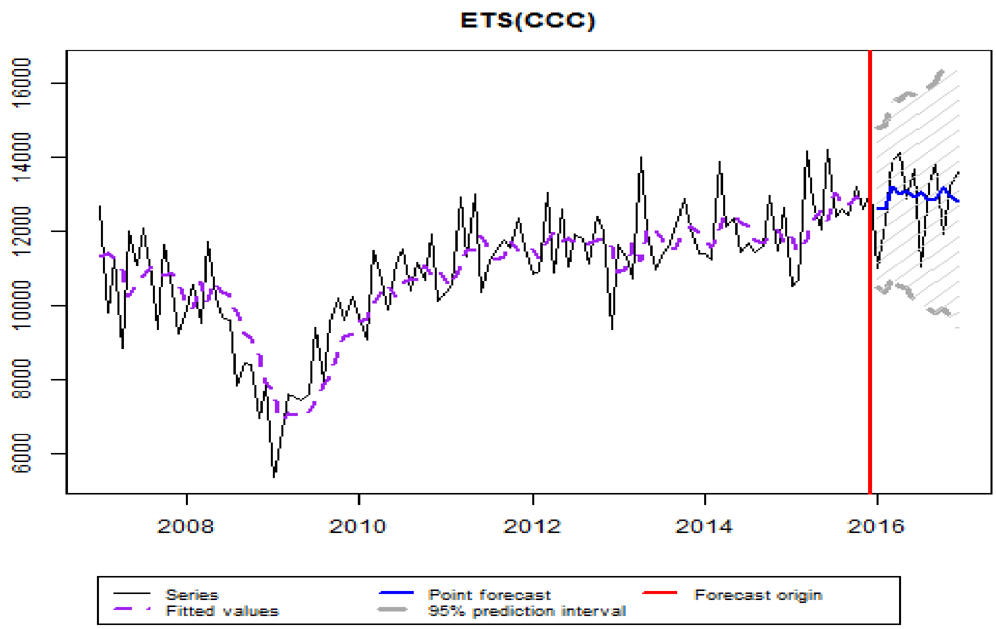
Library(forcats) is a package for categorical variables or factors. Factors help in visualization and modeling because they allow to control the order of the levels. Fct_reorder() reorders the factor levels by another variable. This is useful when mapping a categorical variable to position.
Library(smooth) is a package for generating exponential smoothing. It includes the set of smoothing functions used for time series analysis and in forecasting. Currently the package includes exponential smoothing models and SARIMA in state-space form + several simulation functions



The plot above is a time series from year 2007 to 2015. The Blue line after 2015 upto 2016 indicates a point forecast. The purple dotted line is from fitted values and the series itself is shown in continuous black line. This format will follow for rest of the analysis. After year 2015 we can see the grey area with 95% prediction interval. As indicated by the red vertical line, forecast origin indicates performance of the forecasting model. Above is an Additive None None Model. The first letter denotes the error type ("A", "M" or "Z"); the second letter denotes the trend type ("N","A","M" or "Z"); and the third letter denotes the season type ("N","A","M" or "Z"). In all cases, "N"=none, "A"=additive, "M"=multiplicative and "Z"=automatically selected. So, for example, "ANN" is simple exponential smoothing with additive errors.

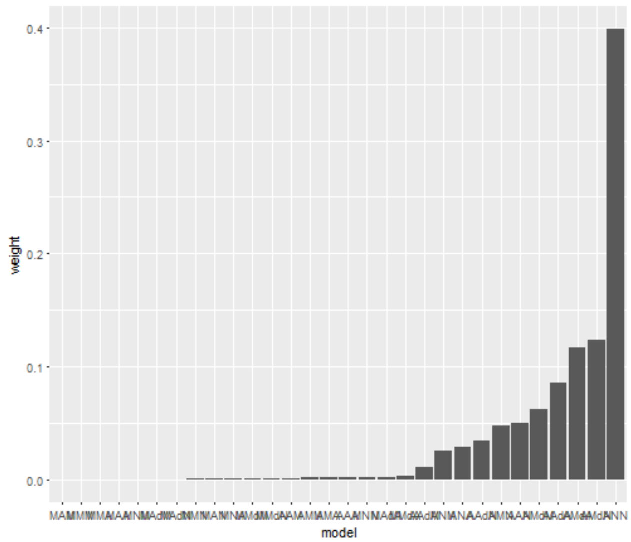
```
3 parameters were estimated in the process
```

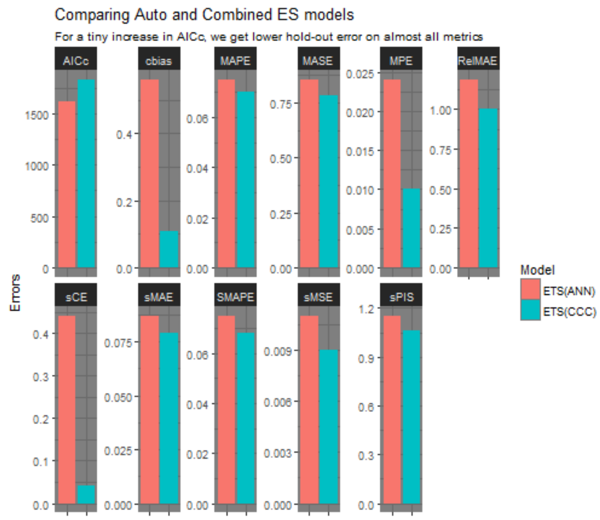
```
## Residuals standard deviation: 1102.852
## Cost function type: MSE; Cost function value: 1178274
##
## Information criteria:
## AIC AICc BIC
## 1620.474 1620.735 1628.167
## 95% parametric prediction intervals were constructed
## 100% of values are in the prediction interval
## Forecast errors:
## MPE: 2.4%; Bias: 55.9%; MAPE: 7.5%; SMAPE: 7.5%
## MASE: 0.854; sMAE: 8.7%; RelMAE: 1.186; sMSE: 1.1%
```



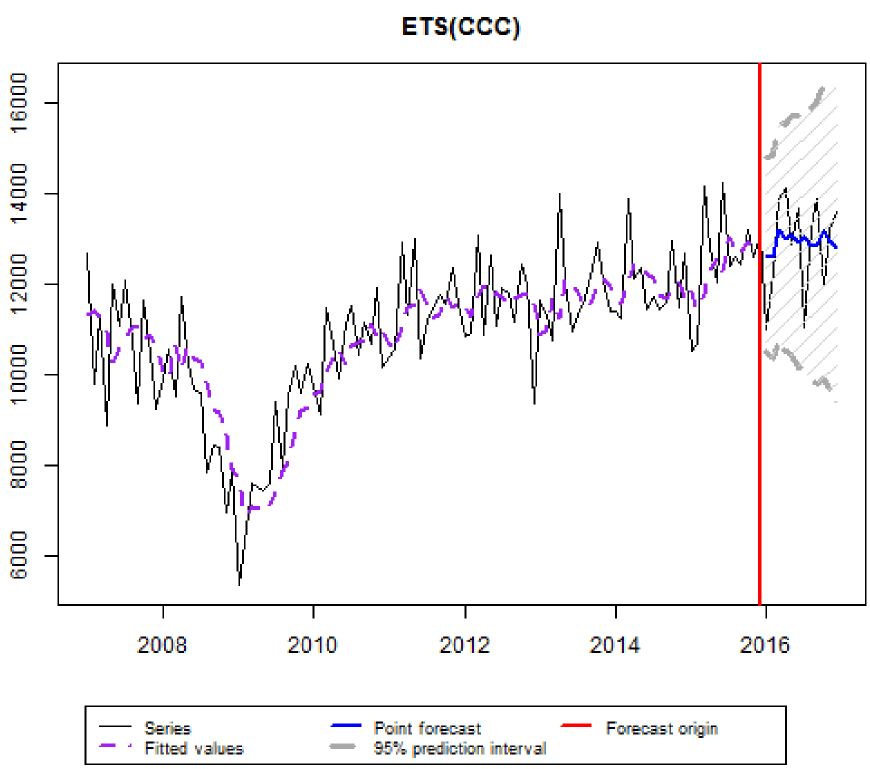
The above model is for Quantity values of Used cars again. It's a CCC Model with values from 2007 to 2016, forecasting for values after 2016.

Data plotted here are basically comparisons of different models. Es() generates 30 different models and 15 of them are quite unstable most of the times. This plot is generated by using enframe() for ICw values of the CCC model generated prior to this. We can see that ANN (Additive, None, None) has highest ICw weight value compared to other models making it the best pick for the analysis.



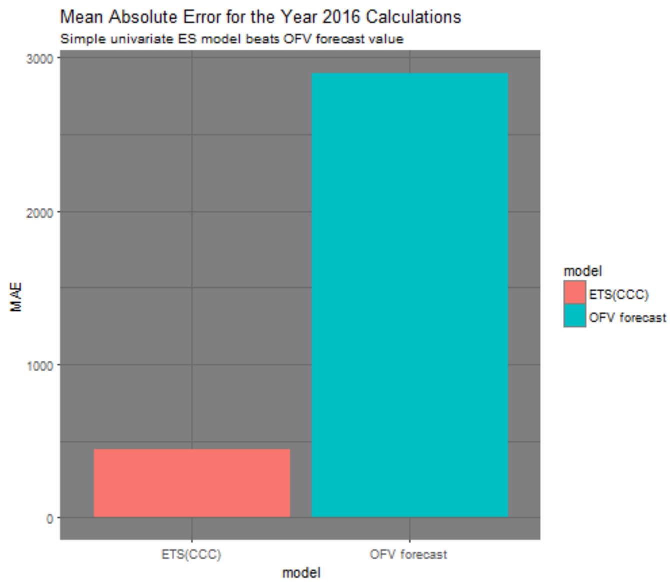


On the graph on right, we can see the AICc value of ETS(CCC) is little higher then AICc value for ETS(ANN). This results in lower error rates on all matrices. Every other ETC(CCC) Error values are lower then ETC(ANN) Model Error values.

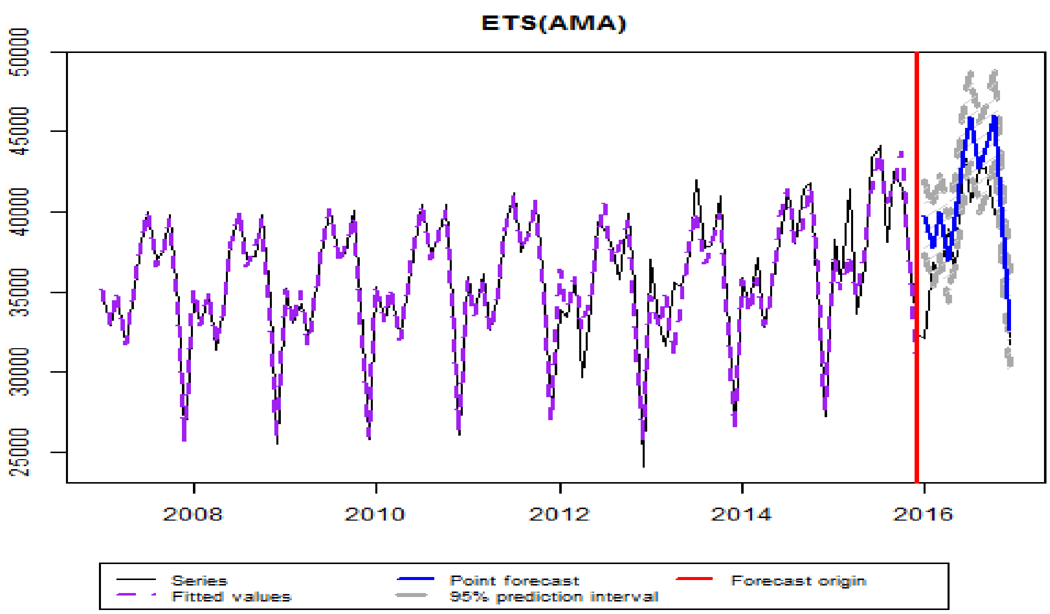



```
## Residuals standard deviation: 1003.013
## Cost function type: MSE
##
## Information criteria:
## Combined AICc
##       1826.874
## 95% parametric prediction intervals were constructed
## 100% of values are in the prediction interval
## Forecast errors:
## MPE: -1%; Bias: 10.8%; MAPE: 7%; SMAPE: 6.8%
## MASE: 0.786; sMAE: 7.9%; RelMAE: 1.006; sMSE: 0.9%
```

Norway Car Market Analysis



Since Year 2016 is being forecasted in two of the models built for comparison for value of Quantity – here we check for Mean Absolute error of the year 2016 calculations. OFV Forecast values for MAE are much higher than MAE for the ETC (CCC) model. Model summary are provided in the Appendix.



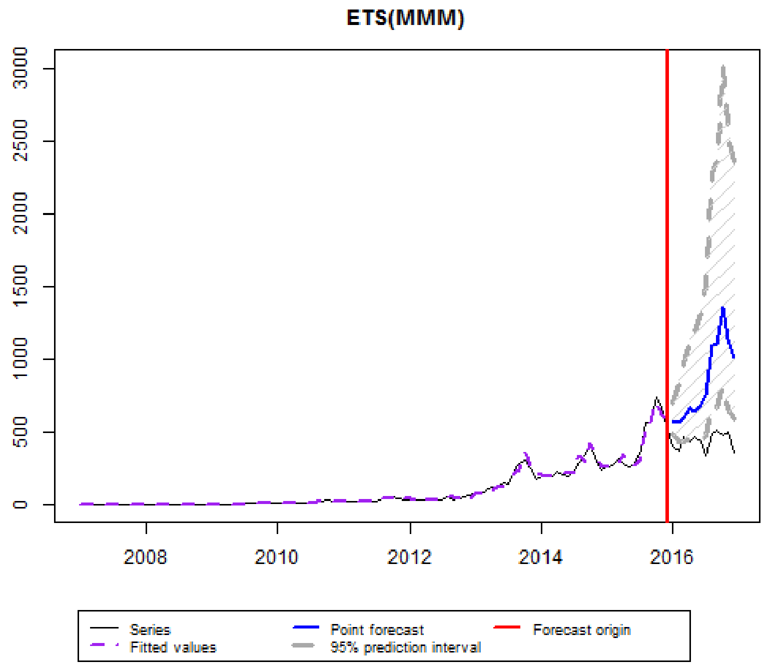
In the figure, above – we are using values of Used cars column with time series and es model is being built along with back casted values.

Cost function type: MSE; Cost function value: 1256517

Information criteria:

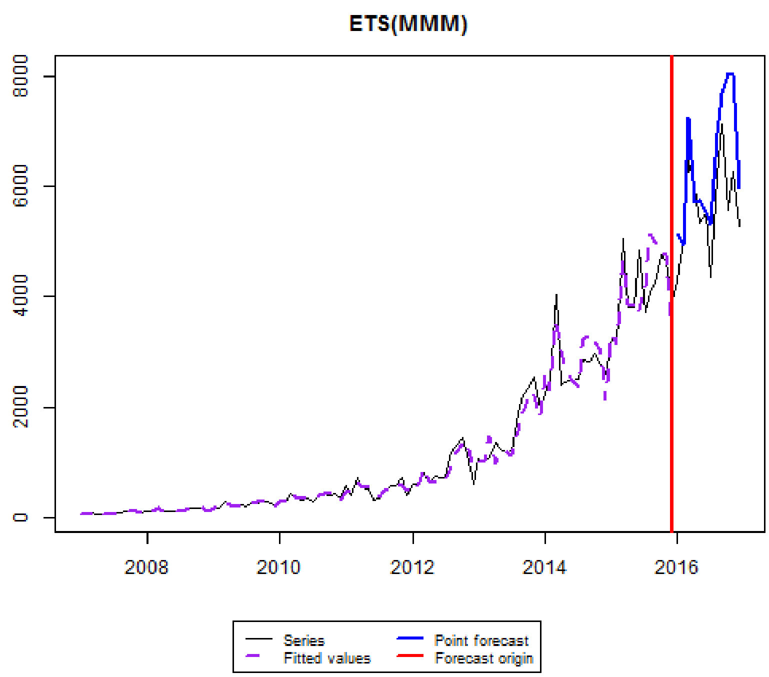
```
##      AIC     AICc      BIC
## 1859.227 1866.912 1907.505
```

We can see a similar pattern being held for back casted values indicating that it may help us with accurate predictions, and pattern for forecasted values which are slightly higher due to obvious reason of increase in used cars.



ES Object for Imported Vehicles gives this result: ## Residuals standard deviation: 0.128

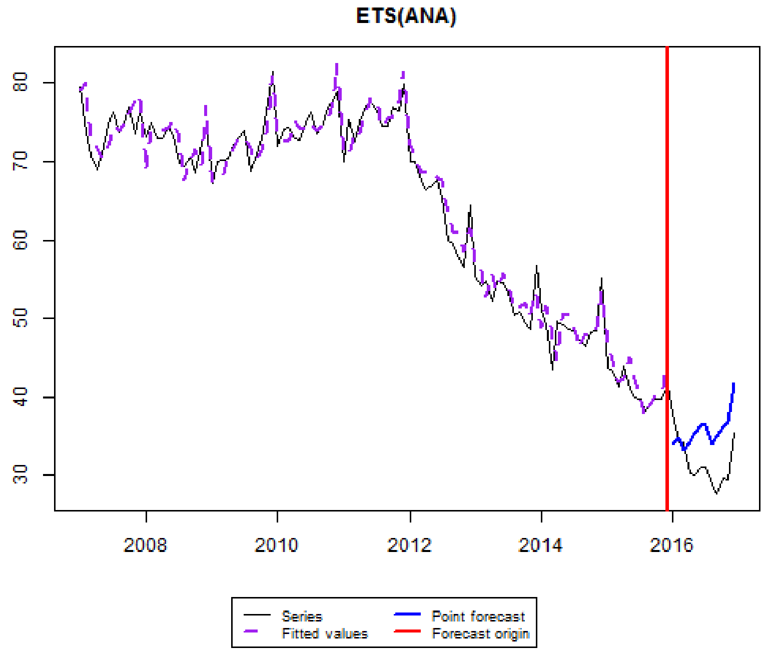
```
## Cost function type: MSE;
Cost function value: 14
##
## Information criteria:
##       AIC      AICc       BIC
## 627.5573 635.2427 675.8357
## 95% nonparametric
prediction intervals were
constructed
## 25% of values are in the
prediction interval
## Forecast errors:
## MPE: -92.9%; Bias: -100%;
MAPE: 92.9%; SMAPE: 59.3%
## MASE: 22.087; sMAE: 379.9%;
```




```
ES Object for Green Cars:
Hybrid, Electric and Import
Electric Combined:
## Residuals standard
deviation: 0.131
## Cost function type: MSE;
Cost function value: 4897
##
## Information criteria:
##       AIC      AICc       BIC
## 1260.104 1267.789 1308.382
## Forecast errors:
## MPE: -13.3%; Bias: -87.2%;
MAPE: 14.5%; SMAPE: 13%
## MASE: 4.082; sMAE: 67.6%;
RelMAE: 0.473; sMSE: 77.3%
```

Plot on the top is for Imported vehicles. es() function chooses MMM model for this . Its prediction values of point forecast are not very close to the actual values for 2016. Hence we see that its error values are higher compared to some of the other models.

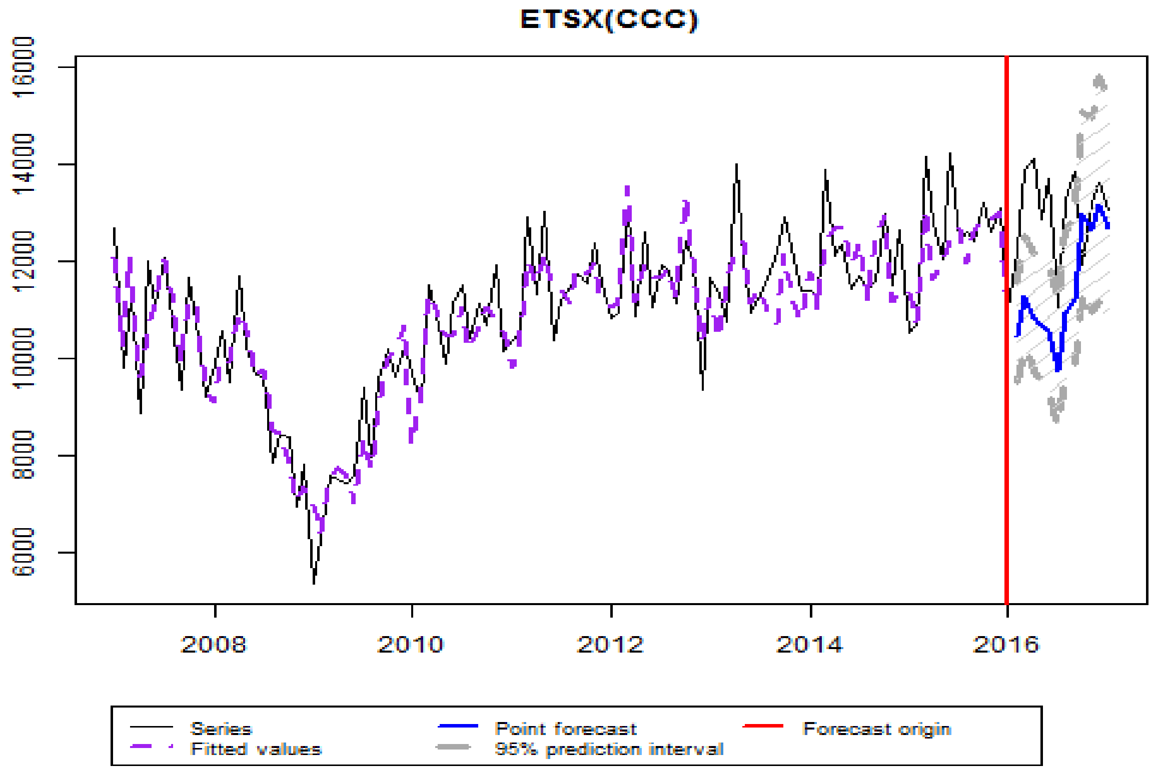
The Second plot is of green cars and its point forecast values are close to the series values. We see that the model's Error values are lower in this case.




```
## Residuals standard
deviation: 2.235
## Cost function type: MSE;
Cost function value: 4
##
## Information criteria:
##        AIC      AICc       BIC
## 494.8759 500.8539 537.7900
## Forecast errors:
## MPE: -13.5%; Bias: -83.4%;
MAPE: 15.7%; SMAPE: 14.4%
## MASE: 1.973; sMAE: 7.5%;
RelMAE: 0.492; sMSE: 0.7%
```

The plot above is of the Diesel share in Norway Market. As we could see in the plot comparing all different vehicles – Diesel share has been declining in last decade. It has gone down close to 25 percent from its 80 percent market share in 2007. The es() function plots it with ANA (Additive, None, Additive) Model with point forecast little higher than the series values. The Mean Square error and Mean Absolute Errors are quiet low for this case.

Our Intention in generating above Time series forecast with exponential smoothing is to be able to create a data frame which can be used to predict Quantity of vehicles using another Exponential smoothing model. The Model was not building correctly when we excluded data from 2016 & 2017. We wanted to see an effect of data prior to 2012 which was randomly populated in accordance with numbers available. When we included data from 2016, and re-ran the prediction model ETSX(CCC) Model was built. This Model Has Point forecast values that are not quite close to the Time Series, but since its 95% prediction interval and Forecast Errors are giving acceptable values – It could be used for the Analysis. We used same method to build model for Import Electric as well. Both results are shown in plots on the next page.

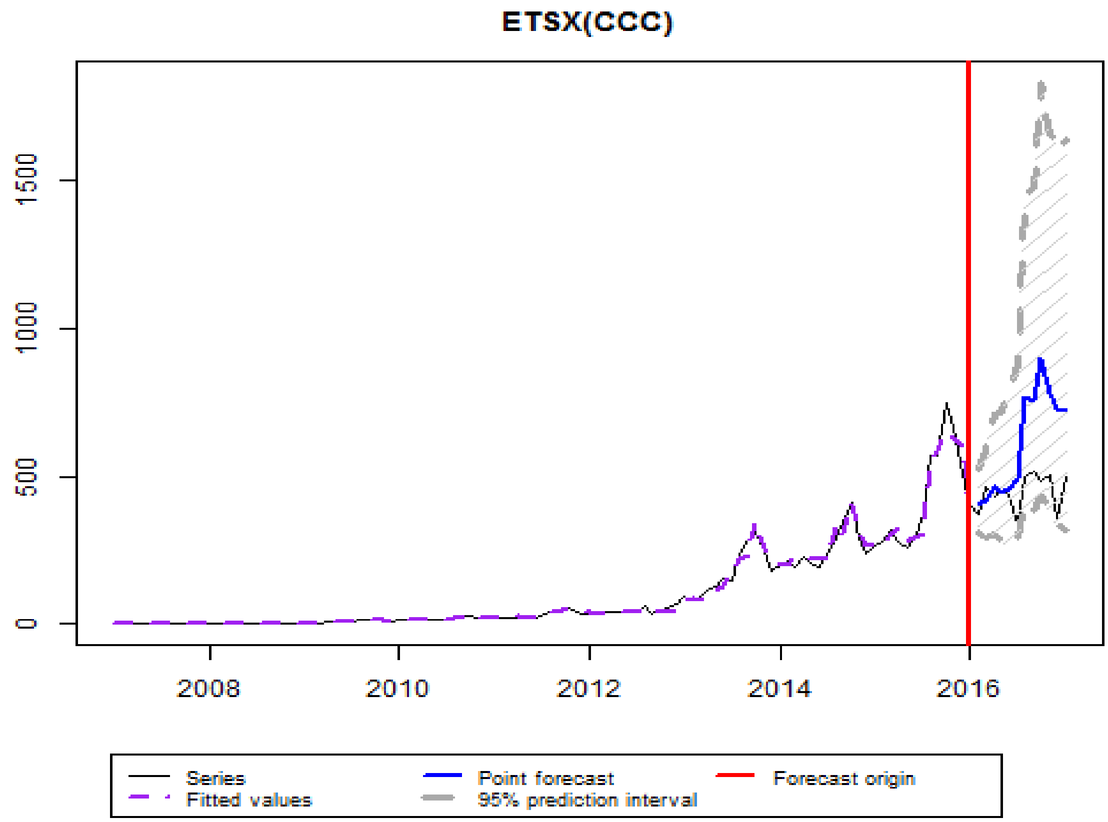


Here we could get a forecast model for ETSX – Prediction based on data frames of **Used, Import, Green and Diesel share ES objects.**

This model does have a little higher MSE then other models, but for us it was important to see the results.

We also used similar way to get prediction model of Electric Cars as shown in the next graph.

Residuals standard deviation: 670.727## Xreg coefficients were estimated in a normal style## Cost function type: MSE## ## Information criteria: ## Combined AICc ## 1793.125 ## 95% parametric rediction intervals were constructed## 42% of values are in the prediction interval## Forecast errors:
MPE: 12.2%; Bias: 91%; MAPE: 13.7%; SMAPE: 14.9%
MASE: 1.625; sMAE: 16.6%; RelMAE: 0.876; sMSE: 3.6%



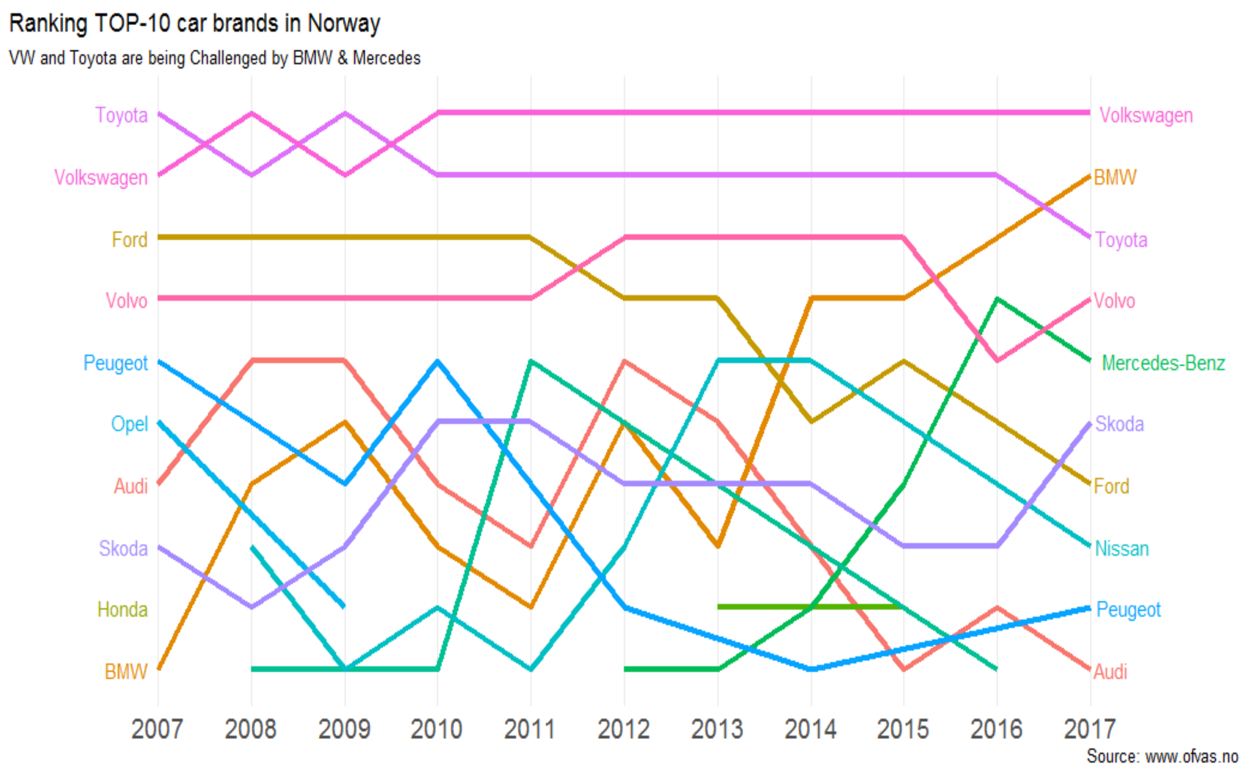

```
## Time elapsed: 11.51 seconds
## Model estimated: ETSX(CCC)
## Initial values were optimised.
## Residuals standard deviation: 20.353
## Xreg coefficients were estimated in a normal style
## Cost function type: MSE
##
## Information criteria:
## Combined AICc
##      657.1759
## 95% parametric prediction intervals were constructed
## 100% of values are in the prediction interval
## Forecast errors:
## MPE: -36.6%; Bias: -88.6%; MAPE: 39.2%; SMAPE: 30.1%
## MASE: 9.044; sMAE: 158.6%; RelMAE: 2.636; sMSE: 403.7%
```

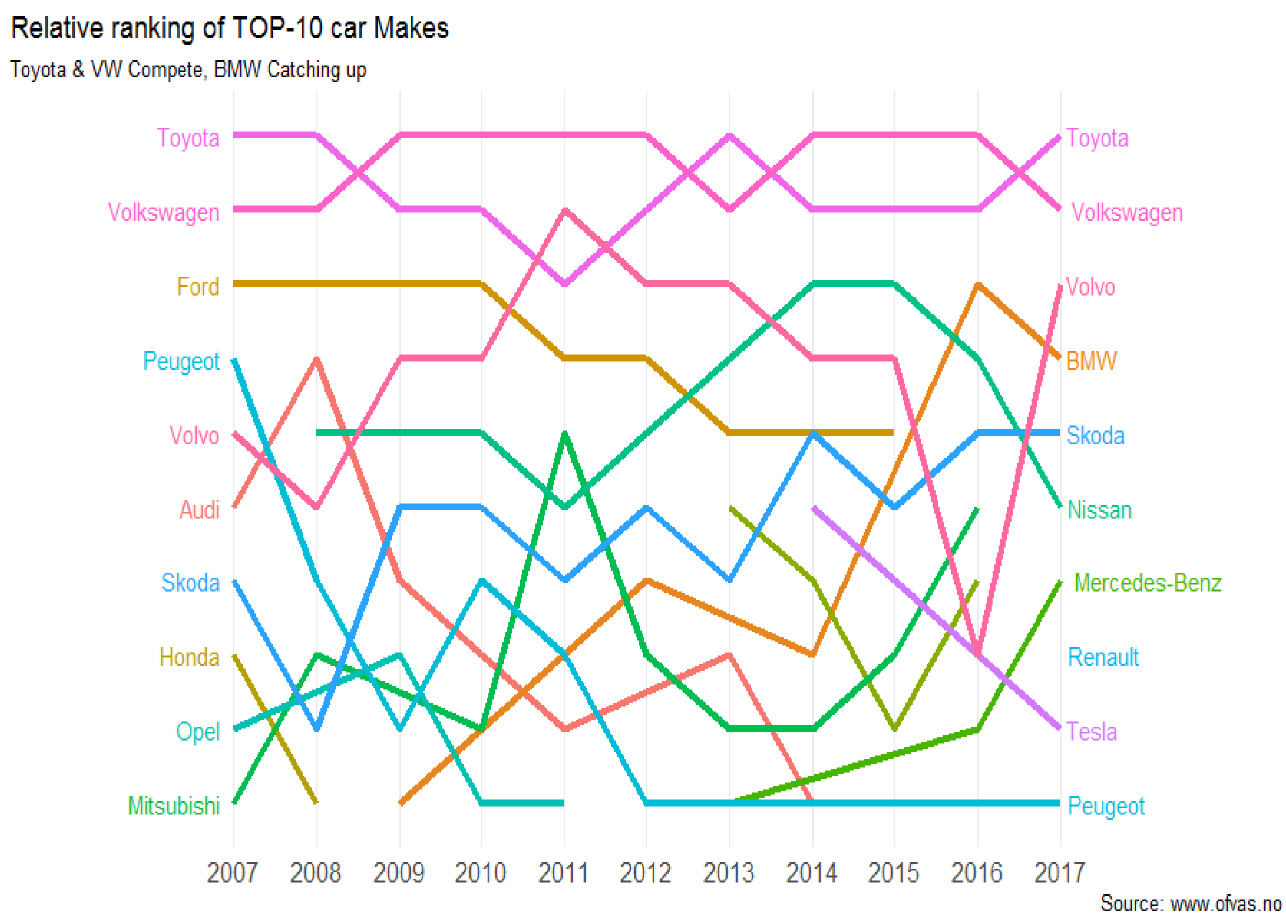
Above values and Model was produced by using same data frames used for creating prediction model for Quantity values. We used data frames from es objects of Used, Import, Greens and Diesel share. Green cars already included the Import Electric cars. However, this model produced a graph that point forecast that is close to the Series values. Error values are bit higher in this model compared to other forecast models, hence – it would not be a very accurate model for forecasting Import Electric Vehicles. Included it just to demonstrate the model we built.

STAT 515 Final Project
Team BALY

Norway Car Market Analysis

Amit Brahmbhatt Yufei Liu





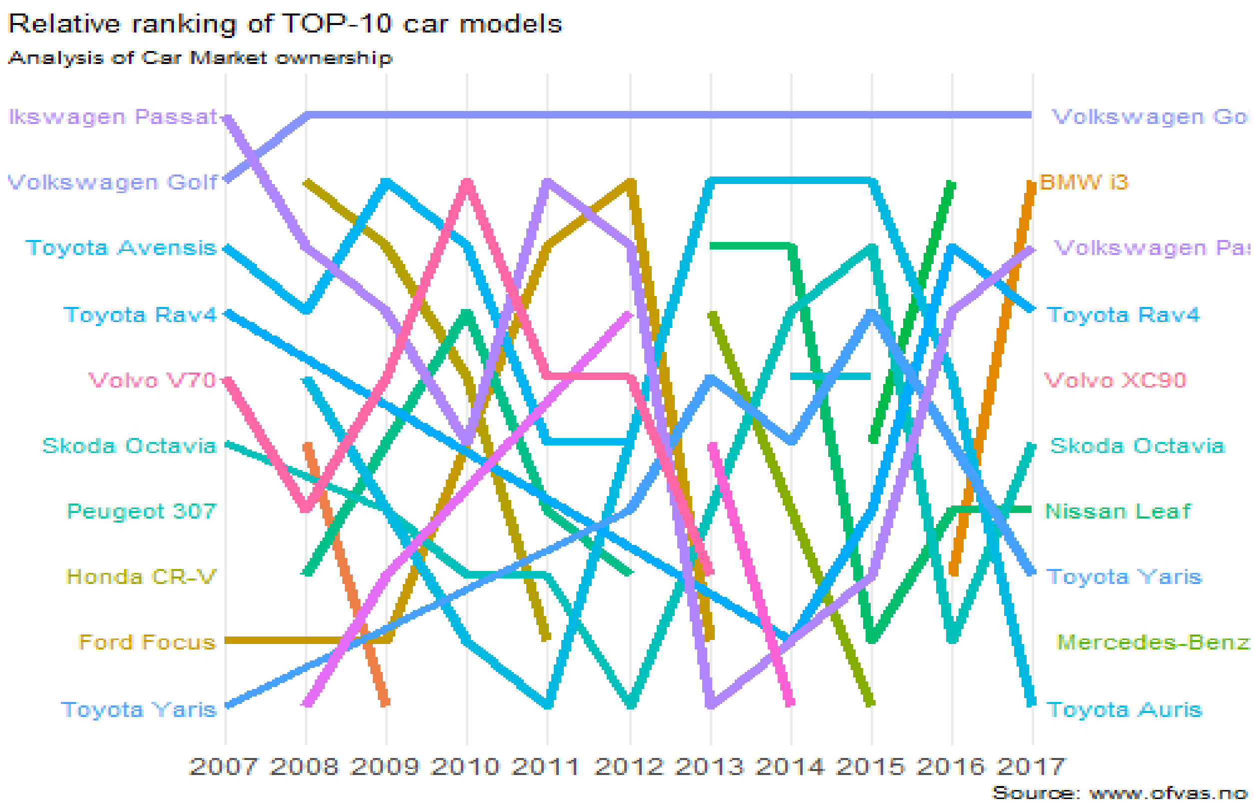
The Two plots above are basically providing comparison between two available sheets. The First graph is from the Car Makes sheet and the other graph is from the Car Models sheet. As the Car Models Sheet also contains data about car makes, we thought of utilizing it to compare accuracy of the first graph.

We Found out that:

1. Toyota and Volkswagen are the top two brands in Norway.
2. Ford was #3, but has been losing its market share in last 10 years.
3. Skoda has had a steady increase in Sales over the decade.
4. Mercedes and BMW are in fact the new competitors in Norway Car market.
5. Honda used to be in the Top-10 in 2007, but has lost significant market share.

From The next plot, we can infer that:

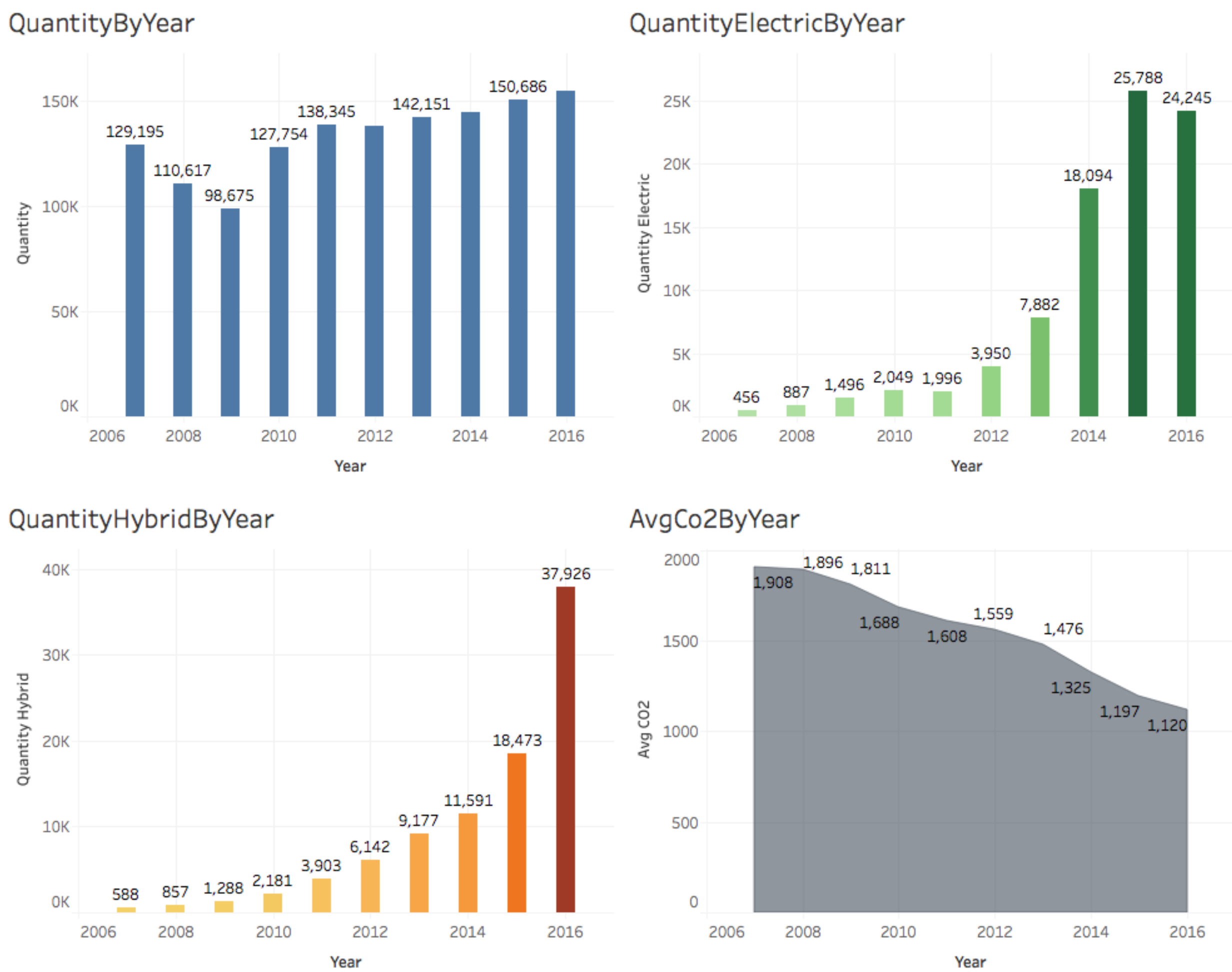
1. Volkswagen Golf is the most popular car model in Norway.
2. 3 Toyota cars have been popular in Last decade – however, the popular models have changed.
3. Skoda Octavia has had consistent sales and have maintained position in top ten throughout the years.
4. Volkswagen Passat was the most popular model that has seen ups and downs over the decade.
5. BMW i3 has come up to become a competitor to Volkswagen Golf just in a year.
6. Toyota Yaris and Skoda Octavia has retained their position in Top-10 popular models over the years.



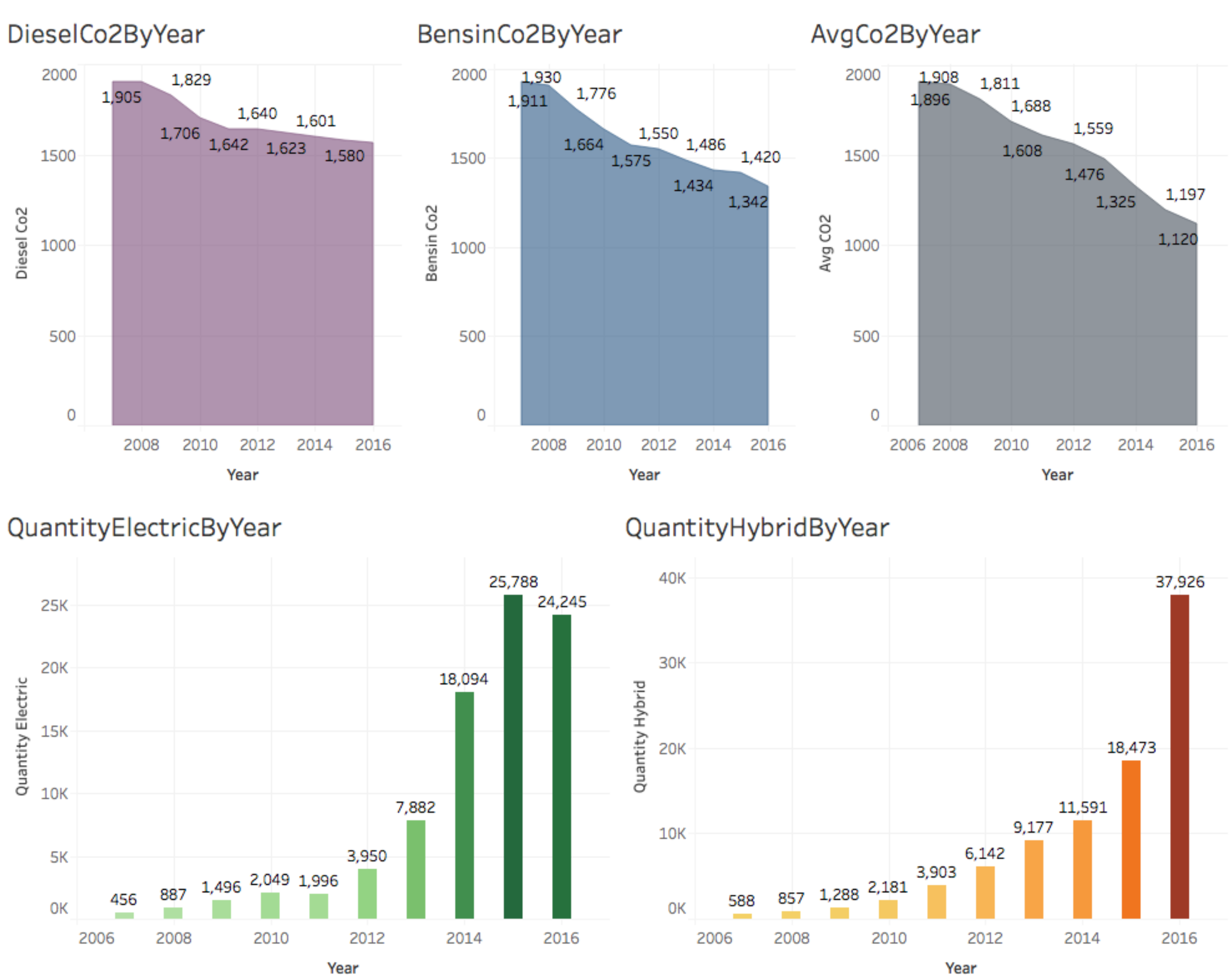
Several of the graphics were result of R code – as mentioned in the Appendix. Ggplot() has lot of great functionalities to display various statistical graphs. There are several Themes and Color pallets one can use to create visualization using R.

Additional Tableau Visualization and Linear Regression Analysis.

Data visualization is the presentation of data by different formats of graphs. Through visualizing the data, we can find different characteristics, trends, patterns, and interactions of the data. In our project, we chose Tableau as one of our visualization tool by bar chart and area graph. Bar chart shows comparisons and trends among related attributes by columns. Area graph displays graphically quantitative data and shows the trends of the attributes. I also made three dashboards for exploring the relationship among different attributes by comparing their different trends.



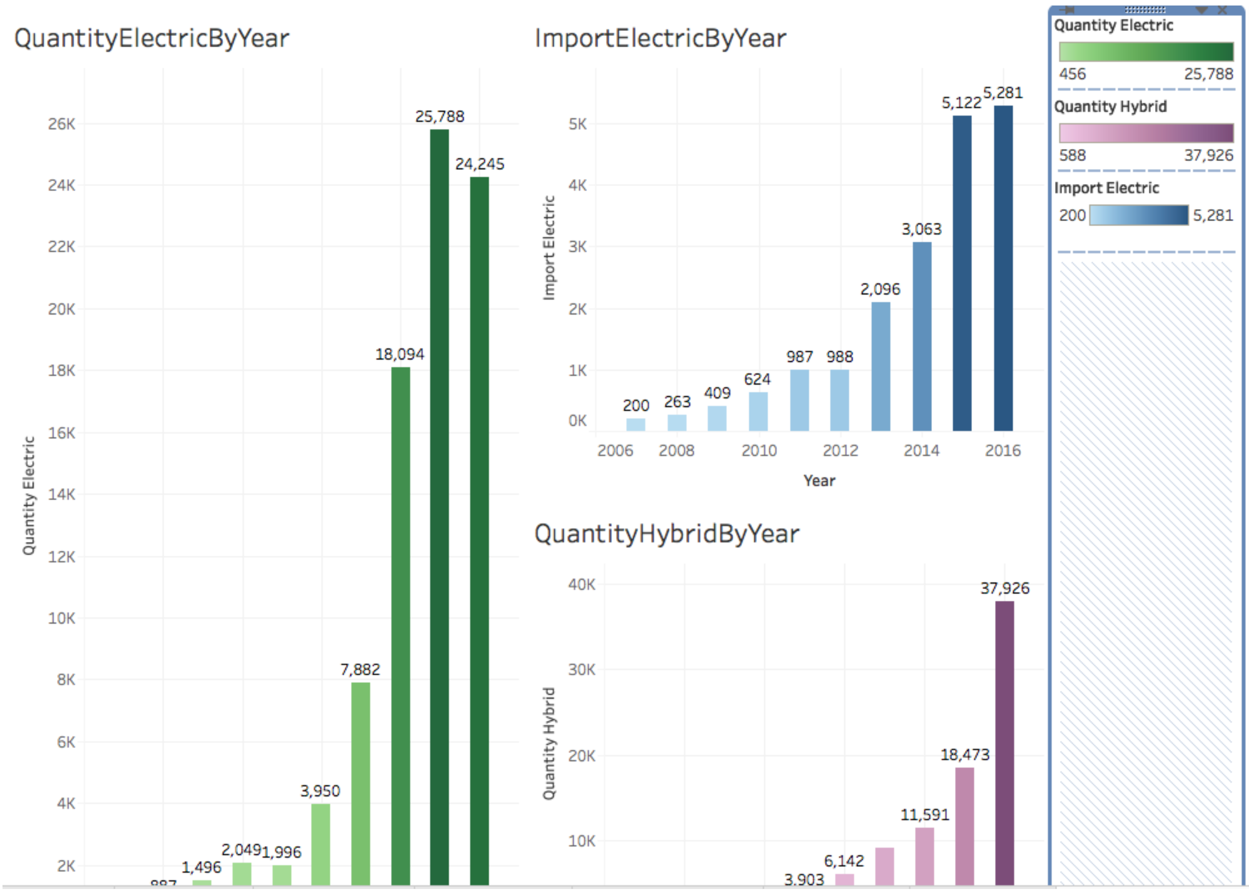
The above dashboard shows three bar charts and one area graph. The first bar chart shows the comparisons of different sale quantity from 2006 to 2016. Also, it shows the trend of quantity of Norway's new car sales by year. The overall trend of Norway's new car sales displays increasing. But it shows a decreasing trend from 2006 to 2009 because the influence of the global financial crisis in 2008. After the year of 2009, Norway's new sales quantity has maintained an increasing trend. The second bar chart shows an increasing trend of the quantity of Norway's electric new car sales from 2006 to 2016. Especially after the year of 2012, the quantity of Norway's new electric car sales increased extremely. The third bar chart shows that the quantity of Norway's Hybrid car sales increased slowly, but it shows an extreme increase from 2015 to 2016. The area graph shows the average CO_2 emission in Norway from 2006 to 2016, and it shows a decreasing trend. I gather the four charts into one dashboard for comparing the increase of the electric car sales and hybrid car sales with the decrease of CO_2 emission in Norway. We can clearly know Norway contributes a lot for CO_2 emission reduction by the increase of electric car sales and hybrid sales.



In this dashboard, there are three area graphs and two bar charts. The three area graphs show CO_2 emission of diesel cars, Bensin car, and average CO_2 in Norway, which all shows quickly decreasing trend. The two bar charts show the increasing trend of quantity of electric car sales and quantity of hybrid car sales in Norway. This dashboard indicates the comparison of the decreasing trends of different kinds of CO_2 emission and the increasing trend of electric car sales and hybrid car sales. Because more and more people in Norway choose to drive electric cars and hybrid cars, the CO_2 emission reduces a lot and very quickly.

Norway Car Market Analysis

Amit Brahmbhatt Yufei Liu



This dashboard shows the trends of electric car sales quantity, hybrid car sales, and imported electric car quantity. The three trends all indicate quick increase. In Norway, most of hybrid cars and electric cars driven by people are imported from other countries, so the quantity of imported cars increases along with the increase of electric car sales and hybrid car sales. We can know that Norway's government put a lot effort for improving environment and CO_2 emission reduction. The effort on improving environment quality and CO_2 emission of Norway 's government is very worth to learn for other countries all over the world.

Multiple Linear Regression Model

Linear Regression Model is simple supervised learning method, which is used for predicting quantitative output values. We choose to use linear regression model to predict different variables for exploring Norway's new car sales.

The below shows each detailed step and analysis for making the linear regression models:

First, we obtain the statistical summary in RStudio for the data that shows below:

```
   Quantity         Import         Avg_CO2        Bensin_Co2   
 Min.   : 5353   Min.   :1048   Min.   : 84.0   Min.   : 94.0  
 1st Qu.:10250   1st Qu.:1812   1st Qu.:110.0   1st Qu.:120.0  
 Median :11385   Median :2263   Median :132.0   Median :131.0  
 Mean   :11134   Mean   :2204   Mean   :129.5   Mean   :133.7  
 3rd Qu.:12337   3rd Qu.:2625   3rd Qu.:151.0   3rd Qu.:150.0  
 Max.   :14207   Max.   :3768   Max.   :162.0   Max.   :165.0  
  Diesel_Co2    Quantity_Diesel  Diesel_Share   Quantity_Hybrid
 Min.   :118.0   Min.   : 3422   Min.   :26.30   Min.   :  32.0  
 1st Qu.:133.0   1st Qu.: 5434   1st Qu.:48.50   1st Qu.: 107.0  
 Median :136.0   Median : 6583   Median :68.70   Median : 357.0  
 Mean   :141.4   Mean   : 6582   Mean   :60.51   Mean   : 797.9  
 3rd Qu.:151.0   3rd Qu.: 7808   3rd Qu.:73.70   3rd Qu.: 982.0  
 Max.   :166.0   Max.   :10072   Max.   :81.40   Max.   :4419.0  
 Quantity_Electric Import_Electric
 Min.   :  21.0    Min.   : 12.0  
 1st Qu.: 107.0    1st Qu.: 32.0  
 Median : 256.0    Median : 83.0  
 Mean   : 736.7    Mean   :161.4  
 3rd Qu.:1398.0    3rd Qu.:260.0  
 Max.   :3391.0    Max.   :746.0  
```

According to this summary, we can clearly know the values of minimum, fist quarter item, median, mean, third quarter item, and maximum for each variable. The values in this summary can make us clearly know about the values and distribution of different variables.

Then, we also get the correlation table in R, which shows below:

```
                    Quantity     Import    Avg_CO2 Bensin_Co2 Diesel_Co2
Quantity           1.0000000  0.0464554 -0.6933005 -0.6645810 -0.6855381
Import             0.0464554  1.0000000  0.4851788  0.3899292  0.2980778
Avg_CO2           -0.6933005  0.4851788  1.0000000  0.9452262  0.9077735
Bensin_Co2        -0.6645810  0.3899292  0.9452262  1.0000000  0.9608364
Diesel_Co2        -0.6855381  0.2980778  0.9077735  0.9608364  1.0000000
Quantity_Diesel    0.0286495  0.6495119  0.5985604  0.4745788  0.3426222
Diesel_Share      -0.5711528  0.4880610  0.9128388  0.8012680  0.7104250
Quantity_Hybrid    0.5928851 -0.5187403 -0.8522039 -0.7522281 -0.6703994
Quantity_Electric  0.6227751 -0.5338488 -0.9087182 -0.7669182 -0.6824586
Import_Electric    0.6045533 -0.4972720 -0.8850175 -0.7699789 -0.7141321
                  Quantity_Diesel Diesel_Share Quantity_Hybrid
Quantity                0.0286495   -0.5711528       0.5928851
Import                  0.6495119    0.4880610      -0.5187403
Avg_CO2                 0.5985604    0.9128388      -0.8522039
Bensin_Co2              0.4745788    0.8012680      -0.7522281
Diesel_Co2              0.3426222    0.7104250      -0.6703994
Quantity_Diesel         1.0000000    0.7983506      -0.6859401
Diesel_Share            0.7983506    1.0000000      -0.8990878
Quantity_Hybrid        -0.6859401   -0.8990878       1.0000000
Quantity_Electric      -0.6600227   -0.9023898       0.8149473
Import_Electric        -0.6509155   -0.8896409       0.8550463
                  Quantity_Electric Import_Electric
Quantity                  0.6227751       0.6045533
Import                   -0.5338488      -0.4972720
Avg_CO2                  -0.9087182      -0.8850175
Bensin_Co2               -0.7669182      -0.7699789
Diesel_Co2               -0.6824586      -0.7141321
Quantity_Diesel          -0.6600227      -0.6509155
Diesel_Share             -0.9023898      -0.8896409
Quantity_Hybrid           0.8149473       0.8550463
Quantity_Electric         1.0000000       0.8793047
Import_Electric           0.8793047       1.0000000
```

This table shows the the variables' correlation. When we predict "Quantity", the variables of "Quantity-Electric", "Quantity_Hybrid", and "Import_Electric" can be better predictors because they show the high correlation value with "Quantity". For predicting "Import", "Quantity_Diesel" shows the hightest correlation value. "Bensin_CO2" and "Diesel_CO2" with the high correlation value can be better predictors for predicting "Avg_CO2". For predicting "Quantity_Hybrid", the variables of "Quantity_Electric" and "Import_Electric" can be the best two predictors because of the higher correlation value for "Quantity_Hybrid". "Quantity_Hybrid" and "Import_Electric" can be the best predictors for predicting "Quantity_Electric". "Quantity_Hybrid" and "Quantity_Electric" have the higher correlation value with "Import_Electric" so that they can be the two best predictors for predicting "Import_Electric".

Next, we partitioned the data into training dataset (60%) and validation dataset (40%). We first used training dataset to make the linear regression models for predict the quantity of Norway's new car sales. We estimate our model by obtaining R-Square in training dataset and Mean Square Error value in validation dataset.

```
Call:
lm(formula = Quantity ~ Import + Avg_CO2 + Bensin_Co2 + Diesel_Co2 +
    Quantity_Diesel + Diesel_Share + Quantity_Hybrid + Quantity_Electric +
    Import_Electric, data = training)

Residuals:
    Min      1Q  Median      3Q     Max
-614.05  -79.42   -2.70   99.03  564.13

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)       9055.51544  823.02956  11.003 3.27e-16 ***
Import               0.03536    0.07369   0.480  0.63301
Avg_CO2             46.09429   13.42915   3.432  0.00107 **
Bensin_Co2         -10.92026    6.77152  -1.613  0.11189
Diesel_Co2         -23.69323   15.28263  -1.550  0.12615
Quantity_Diesel      1.47293    0.03736  39.422  < 2e-16 ***
Diesel_Share      -161.43532    8.36612 -19.296  < 2e-16 ***
Quantity_Hybrid      0.43787    0.05847   7.489 3.05e-10 ***
Quantity_Electric    0.72293    0.14212   5.087 3.62e-06 ***
Import_Electric      0.21100    0.30822   0.685  0.49615
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 192.5 on 62 degrees of freedom
Multiple R-squared:  0.9897,	Adjusted R-squared:  0.9882
F-statistic: 663.4 on 9 and 62 DF,  p-value: < 2.2e-16
> fitsummary=summary(trainfit)
> fitsummary$r.squared
[1] 0.9897218
> PredBase<-predict(trainfit, validation, se.fit=TRUE)
> y_1<-PredBase$fit
> y<-validation$Quantity
> MSE <- mean(y-y_1)^2
> MSE
[1] 10.03034
```

As the result, we got the formula of predicting the Norway's new car sales quantity

“Quantity ~ 1.000e+04+7.475e-02*Import +017.247e+01*Avg_CO2 -2.500e+01*Bensin_Co2 -4.294e+01*Diesel_Co2 + 1.420e*Quantity_Diesel + -1.569e+02*Diesel_Share + 5.117e-01*Quantity_Hybrid + 9.261e-01*Quantity_Electric -6.892e-02*Import_Electric”, which show the influence of each variables for predicting the value of Norway’s new car sales quantity.

Finally, we generated backward regression model. Also, we obtaind R-Square value and Mean Square Error value and compared them with R-Square value and MSE value of forward regression model.

```
Start:  AIC=766.72
Quantity ~ Import + Avg_CO2 + Bensin_Co2 + Diesel_Co2 + Quantity_Diesel +
    Diesel_Share + Quantity_Hybrid + Quantity_Electric + Import_Electric

                    Df Sum of Sq      RSS    AIC
- Import             1      8537  2307060 764.99
- Import_Electric    1     17375  2315898 765.26
<none>                            2298523 766.72
- Diesel_Co2         1     89107  2387630 767.46
- Bensin_Co2         1     96417  2394940 767.68
- Avg_CO2            1    436772  2735295 777.25
- Quantity_Electric  1    959249  3257773 789.83
- Quantity_Hybrid    1   2079394  4377917 811.11
- Diesel_Share       1  13804033 16102556 904.88
- Quantity_Diesel    1  57615095 59913619 999.49
```

```
Step:  AIC=764.99
Quantity ~ Avg_CO2 + Bensin_Co2 + Diesel_Co2 + Quantity_Diesel +
    Diesel_Share + Quantity_Hybrid + Quantity_Electric + Import_Electric

                    Df Sum of Sq      RSS     AIC
- Import_Electric    1     16352  2323412  763.50
<none>                            2307060  764.99
- Bensin_Co2         1     95368  2402429  765.90
- Diesel_Co2         1    109074  2416134  766.31
+ Import             1      8537  2298523  766.72
- Avg_CO2            1    522027  2829088  777.67
- Quantity_Electric  1    956803  3263864  787.97
- Quantity_Hybrid    1   2111986  4419046  809.78
- Diesel_Share       1  22345543 24652604  933.55
- Quantity_Diesel    1  97650056 99957116 1034.34
```

```
Step:  AIC=763.5
Quantity ~ Avg_CO2 + Bensin_Co2 + Diesel_Co2 + Quantity_Diesel +
    Diesel_Share + Quantity_Hybrid + Quantity_Electric

                    Df Sum of Sq       RSS     AIC
<none>                             2323412  763.50
- Bensin_Co2         1     82548   2405960  764.01
+ Import_Electric    1     16352   2307060  764.99
- Diesel_Co2         1    123162   2446573  765.21
+ Import             1      7514   2315898  765.26
- Avg_CO2            1    514730   2838142  775.90
- Quantity_Electric  1   1073794   3397205  788.85
- Quantity_Hybrid    1   2235202   4558614  810.02
- Diesel_Share       1  22443163  24766575  931.88
- Quantity_Diesel    1  97698634 100022046 1032.38
```

```
> coefficients(backward)
      (Intercept)           Avg_CO2        Bensin_Co2        Diesel_Co2
     9331.7243207        47.6540400        -9.8396690       -26.7920694
  Quantity_Diesel      Diesel_Share   Quantity_Hybrid Quantity_Electric
        1.4835410      -164.0524028         0.4385050         0.7438579
```

After using backward regression to reduce the number of predictors, we identified the best model that is the formula "Quantity ~ 47.65*Avg_CO2 – 9.83*Bensin_Co2 – 26.79*Diesel_Co2 + 1.48*Quantity_Diesel – 164.05* Diesel_Share + 0.43*Quantity_Hybrid + 0.74*Quantity_Electric".

```
> fitsummary=summary(backward)
> fitsummary$r.squared
[1] 0.9896105
> PredBase<-predict(backward, validation, se.fit=TRUE)
> y_1<-PredBase$fit
> y<-validation$Quantity
> MSE <- mean(y-y_1)^2
> MSE
[1] 9.371133
> fitsummary=summary(forward)
> fitsummary$r.squared
[1] 0.9896105
> PredBase<-predict(trainfit, validation, se.fit=TRUE)
> y_1<-PredBase$fit
> y<-validation$Quantity
> MSE <- mean(y-y_1)^2
> MSE
[1] 10.03034
```

Compared their R-Square value and MSE value, we find that the backward regression model is better that forward regression model because the better model has bigger R-Square and MSE value.

We also use the same way above to predict "Quantity_Electric", "Import_Electric", "Quantity_Hybrid", "Import", "Quantity_Diesel", "Avg_CO2", and "Quantity_Hybrid+Quantity_Electeic+Import_Electric" for the future.

Conclusion:

While working on this project we tried understanding the data available and discussed possibilities of statistical analysis that could be performed on it. Obvious choices included Multiple Linear Regression and Exponential smoothing. Each of the columns available in our data could be used for prediction with other columns as predictors. We received several interesting results in this manner.

Since there was data available for each month and year starting from January 2007 to Jan 2017 – Time Series analysis became an obvious option. For this we had to generated some of the missing data so that the final models can have enough data points to rely on. For this analysis, we studied various resources available online and used different methods to create es objects with time series. All the details are available in Appendix C of the document. As mentioned in the project proposal we have used additive methods to perform time series exponential smoothing. Exponential Smoothing methods have Trend Component and Seasonal Component. A stands for Additive, N stands for None and M stands for Multiplicative. Ad or Md are damped components.

So ANN model is Additive model with simple exponential smoothing without the seasonal component. es() builds us all 30 available models and gives a comparison on which one to pick. 15 of these models are not stable and not used for most of the analysis. Just like every other analysis we can say that minimizing AIC gives best model for prediction. Appendix lists out the R code used for this analysis and the results obtained for each of the model. Data Analysis and Generation for this model is displayed next to each plot.

We performed six different multi linear regression models for predicting the Norway's new car sales quantity, imported car sales quantity, imported electric car sales quantity, electric car sales quantity, hybrid car sales quantity, and average CO2 emission. All the detailed data can be found under Appendix B of the document. The below table clearly shows the output value for each model:

Target	Predictors	Coefficients	BIC	R-Square	MSE
New car sales quantity	Import + Avg_CO2 + Bensin_Co2 + Diesel_Co2 + Quantity_Diesel + Diesel_Share + Quantity_Hybrid +Quantity_Electric + Import_Electric	Estimate (Intercept) 9055.51544 Import 0.03536 Avg_CO2 46.09429 Bensin_Co2 -10.92026 Diesel_Co2 -23.69323 Quantity_Diesel 1.47293 Diesel_Share -161.43532 Quantity_Hybrid 0.43787 Quantity_Electric 0.72293 Import_Electric 0.21100	998.0905	0.9897	10.0303

Electric car sales quantity	Quantity + Import + Avg_CO2 + Bensin_Co2 + Diesel_Co2 + Quantity_Diesel + Diesel_Share + Quantity_Hybrid + Import_Electric	Estimate (Intercept) -4.421e+03 Quantity 4.073e-01 Import -1.977e-02 Avg_CO2 -7.049e+01 Bensin_Co2 5.967e+00 Diesel_Co2 6.250e+01 Quantity_Diesel -5.830e-01 Diesel_Share 6.901e+01 Quantity_Hybrid -2.832e-01 Import_Electric 2.682e-01	956.78	0.9734	169.6166
Imported electric car sales quantity	Quantity + Import + Avg_CO2 + Bensin_Co2 + Diesel_Co2 + Quantity_Diesel + Diesel_Share + Quantity_Hybrid + Quantity_Electric	Estimate (Intercept) 362.56084 Quantity 0.03556 Import -0.01170 Avg_CO2 -2.93518 Bensin_Co2 5.34304 Diesel_Co2 -6.17559 Quantity_Diesel -0.05262 Diesel_Share 4.07440 Quantity_Hybrid 0.01325 Quantity_Electric 0.08023	869.8753	0.8266	32.84913
Hybrid car sales quantity	Quantity + Import + Avg_CO2 + Bensin_Co2 + Diesel_Co2 + Quantity_Diesel + Diesel_Share + Quantity_Electric + Import_Electric	Estimate (Intercept) -8189.9013 Quantity 1.0847 Import -0.1719 Avg_CO2 -88.4528 Bensin_Co2 19.9184 Diesel_Co2 55.0783 Quantity_Diesel -1.5465 Diesel_Share 154.6262 Quantity_Electric -1.2451 Import_Electric 0.1948	1063.406	0.9232	28.68192
Imported car sales quantity	Quantity + Avg_CO2 + Bensin_Co2 + Diesel_Co2 + Quantity_Diesel + Diesel_Share + Quantity_Hybrid + Quantity_Electric + Import_Electric	Estimate (Intercept) 3099.3507 Quantity 0.1046 Avg_CO2 49.9998 Bensin_Co2 2.8744 Diesel_Co2 -47.0234 Quantity_Diesel 0.1657 Diesel_Share -51.3077 Quantity_Hybrid -0.2053 Quantity_Electric -0.1038 Import_Electric -0.2054	1076.199	0.6767	1576.242

average CO2 emission	Quantity + Import + Bensin_Co2 + Diesel_Co2 + Quantity_Diesel + Diesel_Share + Quantity_Hybrid + Quantity_Electric + Import_Electric	Estimate (Intercept) -5.092e+01 Quantity 3.464e-03 Import 1.270e-03 Bensin_Co2 1.336e-01 Diesel_Co2 8.281e-01 Quantity_Diesel -5.589e-03 Diesel_Share 8.293e-01 Quantity_Hybrid -2.683e-03 Quantity_Electric -9.403e-03 Import_Electric -1.309e-03	314.3812	0.9951	0.000315

Compared their BIC value, R-square value, and MSE value, we can find that the model of predicting average CO2 emission is the best because its MSE values and BIC are the smallest, and R-square is the biggest.

While performing analysis, we experimented with different R-codes to see if we can get values close to actual values provided in the spreadsheet. We also performed two additional Linear Regression Analysis for Green Cars and Avg Co2. For this We tried out few different formulas to find values at certain time. For Example – we used Predict function as follows:

```
>predict(AVGCO2FULL,data.frame(Quantity=13055,Import=1550,Used=6078,Quantity_Diesel=3433,Diesel_Share=26,Diesel_Share_LY=38,Quantity_Electric=2295,Quantity_Hybrid=4419,Import_Electric=494,Bensin_Co2=94,Diesel_Co2=118),interval='confidence')
     fit      lwr      upr
```

1 **75.6433 73.46871 77.8179 è Actual : 84**

```
predict(trainfit,data.frame(Quantity=13500,Quantity_Diesel=3433,Diesel_Share=26,Diesel_Share_LY=38,Import=1550,Used=36078,Avg_CO2=84,Bensin_Co2=94,Diesel_Co2=118),interval='confidence')
    fit      lwr      upr
```

1 **6094.05 5699.559 6488.542 è Actual: 7208.**

This gives us value of the Avg CO2 and Green Cars Total for the month of January 2017. This also gave an insight into why exponential smoothing and Time Series analysis would give better models to work with given data.

Plots and Data for these two analysis can be found under Appendix B.

Norway Car Market Analysis

Amit Brahmbhatt Yufei Liu

After learning statistics and R of one whole semester, we acquired good amount of knowledge about statistics so that we can analyze our data and models by using statistical concepts using RStudio. Also, we learnt how to make different models for classification or prediction. The models are useful and helpful for our future career because it can help us analyze all kinds of data and be applied in almost every domain to improve the development of different domains.

Apart from these two analysis, we also performed a simple analysis on popularity of car makes and models. This confirmed several outcomes stated next to respective graphs. For all of these analysis we used different ggplot() methods and themes with different parameters. Appendix shown below was generated with Rhtml and Knitr package. We generated html and pdf files using the same.

Appendix A: Preliminary Analysis

Used car value population Formula =ROUND (AVERAGE (G73, G85, G97,G109)-0.01*(AVERAGE(G73,G85,G97,G109)),0)
Quantity Electric Data generation Formula: =ROUND(AVERAGE(P61,P73,P85,P97,P109,P121)-0.85*(AVERAGE(P61,P73,P85,P97,P109,P121)),0)
Quantity Hybrid Data Generation Formula: =ROUND(AVERAGE(O61,O73,O85,O97,O109,O121)-0.85*(AVERAGE(O61,O73,O85,O97,O109,O121)),0)
Import Electric Data Generation Formula: =ROUND(AVERAGE(Q81,Q93,Q105,Q117)-0.85*(AVERAGE(Q81,Q93,Q105,Q117)),0)

Exponential Smoothing:
http://robjhyndman.com/talks/RevolutionR/5-ExponentialSmoothing.pdf
Rob J Hyndman is Professor of Statistics at Monash University, Australia, and Editor-in-Chief of the International Journal of Forecasting.

We Begin with Norway New Car csv file and perform **Multiple Linear Regression** below are the results we get after we perform this analysis

```
mydata <- read.csv("norway_new_car_sales_by_month_datab.csv",header = TRUE)

myvars = c(3,5,9:13,15:17)
mydata = mydata[myvars]

head(mydata)
```

```
##   Quantity Import Avg_CO2 Bensin_Co2 Diesel_Co2 Quantity_Diesel
## 1    12685   2276     152        155        152           10072
```

```
## 2     9793   1992     156        159        155            7222
## 3    11264   2626     159        161        158            7965
## 4     8854   2220     160        165        158            6116
## 5    12007   2881     160        163        159            8519
## 6    11083   3038     161        163        160            8290
##   Diesel_Share Quantity_Hybrid Quantity_Electric Import_Electric
## 1         79.4              42                25               2
## 2         73.7              35                27               2
## 3         70.7              48                36               2
## 4         69.1              46                30               2
## 5         71.0              47                27               2
## 6         74.8              41                27               2
```

```
summary(mydata)
```

```
##     Quantity         Import         Avg_CO2        Bensin_Co2
##  Min.   : 5353   Min.   :1048   Min.   : 84.0   Min.   : 94.0
##  1st Qu.:10250   1st Qu.:1812   1st Qu.:110.0   1st Qu.:120.0
##  Median :11385   Median :2263   Median :132.0   Median :131.0
##  Mean   :11134   Mean   :2204   Mean   :129.5   Mean   :133.7
##  3rd Qu.:12337   3rd Qu.:2625   3rd Qu.:151.0   3rd Qu.:150.0
##  Max.   :14207   Max.   :3768   Max.   :162.0   Max.   :165.0
##    Diesel_Co2    Quantity_Diesel  Diesel_Share   Quantity_Hybrid
##  Min.   :118.0   Min.   : 3422   Min.   :26.30   Min.   :  32.0
##  1st Qu.:133.0   1st Qu.: 5434   1st Qu.:48.50   1st Qu.: 107.0
##  Median :136.0   Median : 6583   Median :68.70   Median : 357.0
##  Mean   :141.4   Mean   : 6582   Mean   :60.51   Mean   : 797.9
##  3rd Qu.:151.0   3rd Qu.: 7808   3rd Qu.:73.70   3rd Qu.: 982.0
##  Max.   :166.0   Max.   :10072   Max.   :81.40   Max.   :4419.0
##  Quantity_Electric Import_Electric
##  Min.   :  21.0    Min.   :  2.0
##  1st Qu.: 107.0    1st Qu.:  9.0
##  Median : 256.0    Median : 40.0
##  Mean   : 736.7    Mean   :144.1
##  3rd Qu.:1398.0    3rd Qu.:260.0
##  Max.   :3391.0    Max.   :746.0
```

```
cor(mydata)
```

```
##                     Quantity     Import    Avg_CO2 Bensin_Co2 Diesel_Co2
## Quantity           1.0000000  0.0464554 -0.6933005 -0.6645810 -0.6855381
## Import             0.0464554  1.0000000  0.4851788  0.3899292  0.2980778
## Avg_CO2           -0.6933005  0.4851788  1.0000000  0.9452262  0.9077735
## Bensin_Co2        -0.6645810  0.3899292  0.9452262  1.0000000  0.9608364
## Diesel_Co2        -0.6855381  0.2980778  0.9077735  0.9608364  1.0000000
## Quantity_Diesel    0.0286495  0.6495119  0.5985604  0.4745788  0.3426222
## Diesel_Share      -0.5711528  0.4880610  0.9128388  0.8012680  0.7104250
## Quantity_Hybrid    0.5928851 -0.5187403 -0.8522039 -0.7522281 -0.6703994
## Quantity_Electric  0.6227751 -0.5338488 -0.9087182 -0.7669182 -0.6824586
## Import_Electric    0.5940507 -0.5010390 -0.8816827 -0.7585881 -0.6908066
##                   Quantity_Diesel Diesel_Share Quantity_Hybrid
## Quantity                0.0286495   -0.5711528       0.5928851
## Import                  0.6495119    0.4880610      -0.5187403
## Avg_CO2                 0.5985604    0.9128388      -0.8522039
## Bensin_Co2              0.4745788    0.8012680      -0.7522281
## Diesel_Co2              0.3426222    0.7104250      -0.6703994
## Quantity_Diesel         1.0000000    0.7983506      -0.6859401
## Diesel_Share            0.7983506    1.0000000      -0.8990878
## Quantity_Hybrid        -0.6859401   -0.8990878       1.0000000
## Quantity_Electric      -0.6600227   -0.9023898       0.8149473
```

```
## Import_Electric        -0.6837837   -0.9112012       0.8580814
##                   Quantity_Electric Import_Electric
## Quantity                  0.6227751       0.5940507
## Import                   -0.5338488      -0.5010390
## Avg_CO2                  -0.9087182      -0.8816827
## Bensin_Co2               -0.7669182      -0.7585881
## Diesel_Co2               -0.6824586      -0.6908066
## Quantity_Diesel          -0.6600227      -0.6837837
## Diesel_Share             -0.9023898      -0.9112012
## Quantity_Hybrid           0.8149473       0.8580814
## Quantity_Electric         1.0000000       0.8908605
## Import_Electric           0.8908605       1.0000000
```

```
pairs(mydata)
```


```
row<-nrow(mydata)
set.seed(12345)   ###### 60% for training data
trainindex <- sample(row,72, replace=FALSE)
training <- mydata[trainindex, ]
validation <- mydata[-trainindex, ]
```

Here is a Correlation chart for all the columns **Multiple Linear Regression** We can analyze and understand useful correlations.

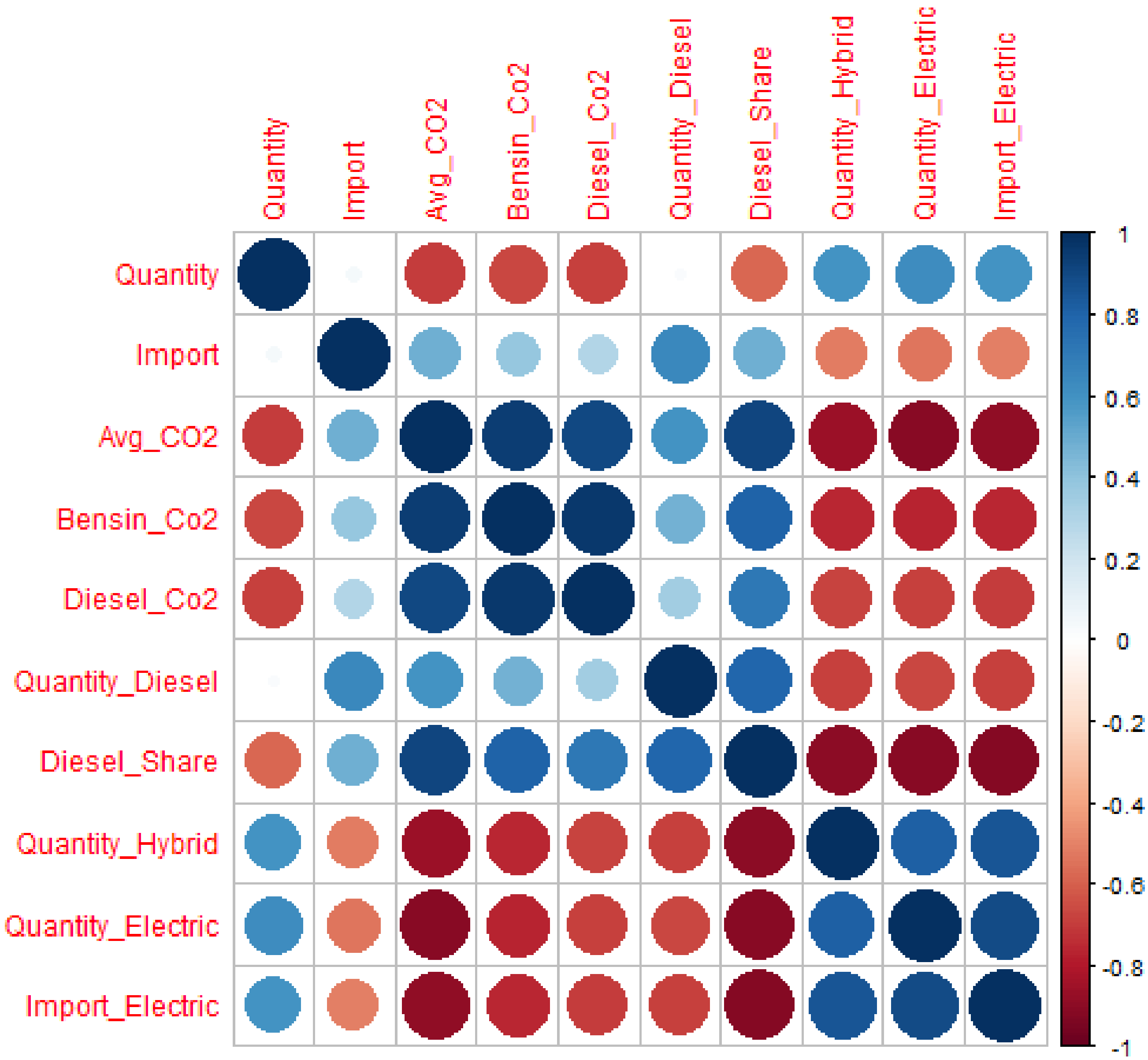
```
M=cor(mydata)
install.packages("corrplot")
```

```
## Installing package into 'C:/Users/BinalAmit/Documents/R/win-library/3.3'
## (as 'lib' is unspecified)
```

```
## package 'corrplot' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##     C:\Users\BinalAmit\AppData\Local\Temp\RtmpSe6Rfq\downloaded_packages
```

```
library(corrplot)
corrplot(M, method = "number")
```


```
corrplot(M, method = "circle")
```




```
corrplot.mixed(M)
```


Quantity
Import
vg_CO2
nsin_Co2
esel_Co2
ntity_Diesel
sel_Share
ntity_Hybrid
ntity_Electric
ort_Electric
0.05
-0.69
0.49
-0.66
0.39
0.95
-0.69
0.3
0.91
0.96
0.03
0.65
0.6
0.47
0.34
-0.57
0.49
0.91
0.8
0.71
0.8
0.59
-0.52
-0.85
-0.75
-0.67
-0.69
-0.9
0.62
-0.53
-0.91
-0.77
-0.68
-0.66
-0.9
0.81
0.59
-0.5
-0.88
-0.76
-0.69
-0.68
-0.91
0.86
0.89
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1

```
plot(mydata)
```

Appendix B: Multiple Linear Regression

Working with Quantity **Multiple Linear Regression** below are the results we get after we perform this analysis

```
###predict Quantity
```

```
trainfit<-lm(Quantity ~ Import + Avg_CO2 + Bensin_Co2 + Diesel_Co2 +
Quantity_Diesel + Diesel_Share + Quantity_Hybrid + Quantity_Electric +
Import_Electric, data = training)
summary(trainfit)
```

```
##
## Call:
## lm(formula = Quantity ~ Import + Avg_CO2 + Bensin_Co2 + Diesel_Co2 +
##     Quantity_Diesel + Diesel_Share + Quantity_Hybrid + Quantity_Electric +
##     Import_Electric, data = training)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -616.70  -76.33   -1.28  101.36  563.36
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)        9087.44302  816.30707  11.132  < 2e-16 ***
## Import                0.03600    0.07381   0.488   0.6275
## Avg_CO2              45.99104   13.43475   3.423   0.0011 **
## Bensin_Co2          -10.70878    6.73602  -1.590   0.1170
## Diesel_Co2          -24.15372   15.22686  -1.586   0.1178
## Quantity_Diesel       1.47214    0.03741  39.355  < 2e-16 ***
## Diesel_Share       -161.00991    8.44900 -19.057  < 2e-16 ***
## Quantity_Hybrid       0.44020    0.05814   7.572 2.19e-10 ***
## Quantity_Electric     0.72445    0.14241   5.087 3.61e-06 ***
## Import_Electric       0.19202    0.30864   0.622   0.5361
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 192.7 on 62 degrees of freedom
## Multiple R-squared:  0.9897,	Adjusted R-squared:  0.9882
## F-statistic: 662.5 on 9 and 62 DF,  p-value: < 2.2e-16
```

```
BIC(trainfit)
```

```
## [1] 998.1846
```

```
fitsummary=summary(trainfit)
fitsummary$r.squared
```

```
## [1] 0.9897083
```

```
PredBase<-predict(trainfit, validation, se.fit=TRUE)
y_1<-PredBase$fit
y<-validation$Quantity
MSE <- mean(y-y_1)^2
MSE
```

```
## [1] 14.26005
```

```
backward <- step(trainfit, direction = 'both')
```

```
## Start:  AIC=766.81
## Quantity ~ Import + Avg_CO2 + Bensin_Co2 + Diesel_Co2 + Quantity_Diesel +
##     Diesel_Share + Quantity_Hybrid + Quantity_Electric + Import_Electric
##
##                     Df Sum of Sq     RSS    AIC
## - Import             1      8828 2310358 765.09
## - Import_Electric    1     14368 2315898 765.26
## <none>                           2301530 766.81
## - Diesel_Co2         1     93406 2394935 767.68
## - Bensin_Co2         1     93821 2395351 767.69
## - Avg_CO2            1    435023 2736553 777.28
## - Quantity_Electric  1    960684 3262214 789.93
```

```
## - Quantity_Hybrid    1   2128226  4429756 811.96
## - Diesel_Share       1  13480931 15782461 903.44
## - Quantity_Diesel    1  57495336 59796866 999.35
##
## Step:  AIC=765.09
## Quantity ~ Avg_CO2 + Bensin_Co2 + Diesel_Co2 + Quantity_Diesel +
##     Diesel_Share + Quantity_Hybrid + Quantity_Electric + Import_Electric
##
##                     Df Sum of Sq       RSS     AIC
## - Import_Electric    1     13054   2323412  763.50
## <none>                             2310358  765.09
## - Bensin_Co2         1     92633   2402991  765.92
## - Diesel_Co2         1    114387   2424745  766.57
## + Import             1      8828   2301530  766.81
## - Avg_CO2            1    520647   2831005  777.72
## - Quantity_Electric  1    958928   3269286  788.09
## - Quantity_Hybrid    1   2162127   4472485  810.65
## - Diesel_Share       1  21858983  24169341  932.12
## - Quantity_Diesel    1  97711415 100021773 1034.38
##
## Step:  AIC=763.5
## Quantity ~ Avg_CO2 + Bensin_Co2 + Diesel_Co2 + Quantity_Diesel +
##     Diesel_Share + Quantity_Hybrid + Quantity_Electric
##
##                     Df Sum of Sq       RSS     AIC
## <none>                             2323412  763.50
## - Bensin_Co2         1     82548   2405960  764.01
## + Import_Electric    1     13054   2310358  765.09
## - Diesel_Co2         1    123162   2446573  765.21
## + Import             1      7514   2315898  765.26
## - Avg_CO2            1    514730   2838142  775.90
## - Quantity_Electric  1   1073794   3397205  788.85
## - Quantity_Hybrid    1   2235202   4558614  810.02
## - Diesel_Share       1  22443163  24766575  931.88
## - Quantity_Diesel    1  97698634 100022046 1032.38
```

```
coefficients(backward)
```

```
##       (Intercept)           Avg_CO2        Bensin_Co2        Diesel_Co2
##      9331.7243207        47.6540400        -9.8396690       -26.7920694
##   Quantity_Diesel      Diesel_Share   Quantity_Hybrid Quantity_Electric
##         1.4835410      -164.0524028         0.4385050         0.7438579
```

```
BIC(backward)
```

```
## [1] 990.3126
```

```
fitsummary=summary(backward)
fitsummary$r.squared
```

```
## [1] 0.9896105
```

```
PredBase<-predict(backward, validation, se.fit=TRUE)
y_1<-PredBase$fit
y<-validation$Quantity

MSE <- mean(y-y_1)^2
MSE
```

```
## [1] 9.371133
```

Working with Quantity Electric **Multiple Linear Regression** below are the results we get after we perform this analysis

```
###predict Quantity_Electric
trainfitQE<-lm(Quantity_Electric ~ Quantity + Import + Avg_CO2 + Bensin_Co2 +
Diesel_Co2 + Quantity_Diesel + Diesel_Share + Quantity_Hybrid +
Import_Electric, data = training)
summary(trainfitQE)
```

```
##
## Call:
## lm(formula = Quantity_Electric ~ Quantity + Import + Avg_CO2 +
##     Bensin_Co2 + Diesel_Co2 + Quantity_Diesel + Diesel_Share +
##     Quantity_Hybrid + Import_Electric, data = training)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -404.11  -51.52    1.25   47.60  715.87
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -4.396e+03  8.998e+02  -4.885 7.62e-06 ***
## Quantity         4.065e-01  7.991e-02   5.087 3.61e-06 ***
## Import          -1.831e-02  5.535e-02  -0.331    0.742
## Avg_CO2         -7.040e+01  6.362e+00 -11.066 2.57e-16 ***
## Bensin_Co2       6.043e+00  5.090e+00   1.187    0.240
## Diesel_Co2       6.197e+01  8.570e+00   7.231 8.55e-10 ***
## Quantity_Diesel -5.830e-01  1.221e-01  -4.774 1.14e-05 ***
## Diesel_Share     6.958e+01  1.402e+01   4.962 5.74e-06 ***
## Quantity_Hybrid -2.804e-01  4.881e-02  -5.745 2.99e-07 ***
## Import_Electric  2.843e-01  2.291e-01   1.241    0.219
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 144.3 on 62 degrees of freedom
## Multiple R-squared:  0.9735,	Adjusted R-squared:  0.9697
## F-statistic: 253.5 on 9 and 62 DF,  p-value: < 2.2e-16
```

```
backward <- step(trainfitQE, direction = 'both')
```

```
## Start:  AIC=725.21
## Quantity_Electric ~ Quantity + Import + Avg_CO2 + Bensin_Co2 +
##     Diesel_Co2 + Quantity_Diesel + Diesel_Share + Quantity_Hybrid +
##     Import_Electric
##
##                   Df Sum of Sq     RSS    AIC
## - Import           1      2281 1293687 723.34
## - Bensin_Co2       1     29358 1320764 724.83
## - Import_Electric  1     32075 1323481 724.98
## <none>                         1291406 725.21
## - Quantity_Diesel  1    474690 1766096 745.75
## - Diesel_Share     1    512918 1804324 747.29
## - Quantity         1    539047 1830454 748.33
## - Quantity_Hybrid  1    687478 1978884 753.94
## - Diesel_Co2       1   1089139 2380545 767.24
## - Avg_CO2          1   2550805 3842211 801.71
##
## Step:  AIC=723.34
## Quantity_Electric ~ Quantity + Avg_CO2 + Bensin_Co2 + Diesel_Co2 +
##     Quantity_Diesel + Diesel_Share + Quantity_Hybrid + Import_Electric
##
##                   Df Sum of Sq     RSS    AIC
```

```
## - Bensin_Co2       1     28935 1322622 722.93
## - Import_Electric  1     33540 1327227 723.18
## <none>                         1293687 723.34
## + Import           1      2281 1291406 725.21
## - Quantity_Diesel  1    486138 1779825 744.31
## - Quantity         1    536952 1830639 746.33
## - Diesel_Share     1    558643 1852330 747.18
## - Quantity_Hybrid  1    700464 1994151 752.49
## - Diesel_Co2       1   1272587 2566274 770.65
## - Avg_CO2          1   3471781 4765468 815.22
##
## Step:  AIC=722.93
## Quantity_Electric ~ Quantity + Avg_CO2 + Diesel_Co2 + Quantity_Diesel +
##     Diesel_Share + Quantity_Hybrid + Import_Electric
##
##                   Df Sum of Sq     RSS    AIC
## <none>                         1322622 722.93
## + Bensin_Co2       1     28935 1293687 723.34
## - Import_Electric  1     53164 1375785 723.77
## + Import           1      1857 1320764 724.83
## - Quantity_Diesel  1    460203 1782824 742.43
## - Quantity         1    512686 1835308 744.52
## - Diesel_Share     1    534608 1857229 745.37
## - Quantity_Hybrid  1    671628 1994249 750.50
## - Diesel_Co2       1   2422230 3744852 795.86
## - Avg_CO2          1   3675736 4998357 816.65
```

```
coefficients(backward)
```

```
##     (Intercept)        Quantity         Avg_CO2      Diesel_Co2
##   -4658.9215020       0.3921147     -69.2546347      68.7379318
## Quantity_Diesel    Diesel_Share Quantity_Hybrid Import_Electric
##      -0.5632204      68.4820623      -0.2663192       0.3543742
```

```
BIC(backward)
```

```
## [1] 949.7463
```

```
fitsummary=summary(backward)
fitsummary$r.squared
```

```
## [1] 0.9729068
```

```
PredBase<-predict(backward, validation, se.fit=TRUE)

y_1<-PredBase$fit

y<-validation$Quantity_Electric

MSE <- mean(y-y_1)^2
MSE
```

```
## [1] 152.4056
```

Working with Import Electric **Multiple Linear Regression** below are the results we get after we perform this analysis

```
###predict Import_Electric
trainfitIE<-lm(Import_Electric ~ Quantity + Import + Avg_CO2 + Bensin_Co2 + Diesel_Co2 + Quantity_Diesel + Diesel_Share + Quantity_Hybrid + Quantity_Electric, data = training)
summary(trainfitIE)
```

```
##
## Call:
## lm(formula = Import_Electric ~ Quantity + Import + Avg_CO2 +
##     Bensin_Co2 + Diesel_Co2 + Quantity_Diesel + Diesel_Share +
##     Quantity_Hybrid + Quantity_Electric, data = training)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -265.11  -26.71   -4.67   11.86  329.29
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       294.341512 578.657979   0.509   0.6128
## Quantity            0.032310   0.051933   0.622   0.5361
## Import             -0.015921   0.030269  -0.526   0.6008
## Avg_CO2            -2.377862   6.001611  -0.396   0.6933
## Bensin_Co2          4.703181   2.754849   1.707   0.0928 .
## Diesel_Co2         -4.558259   6.345163  -0.718   0.4752
## Quantity_Diesel    -0.043740   0.078014  -0.561   0.5770
## Diesel_Share        1.167859   9.074444   0.129   0.8980
## Quantity_Hybrid     0.005448   0.033078   0.165   0.8697
## Quantity_Electric   0.085250   0.068699   1.241   0.2193
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 79.03 on 62 degrees of freedom
## Multiple R-squared:  0.8465,     Adjusted R-squared:  0.8243
## F-statistic:    38 on 9 and 62 DF,  p-value: < 2.2e-16
```

```
backward <- step(trainfitIE, direction = 'both')
```

```
## Start:  AIC=638.49
## Import_Electric ~ Quantity + Import + Avg_CO2 + Bensin_Co2 +
##     Diesel_Co2 + Quantity_Diesel + Diesel_Share + Quantity_Hybrid +
##     Quantity_Electric
##
##                     Df Sum of Sq    RSS    AIC
## - Diesel_Share       1     103.5 387365 636.51
## - Quantity_Hybrid    1     169.4 387431 636.53
## - Avg_CO2            1     980.5 388242 636.68
## - Import             1    1728.2 388990 636.81
## - Quantity_Diesel    1    1963.4 389225 636.86
## - Quantity           1    2417.6 389679 636.94
## - Diesel_Co2         1    3223.5 390485 637.09
## - Quantity_Electric  1    9618.5 396880 638.26
## <none>                           387262 638.49
## - Bensin_Co2         1   18205.4 405467 639.80
##
## Step:  AIC=636.51
## Import_Electric ~ Quantity + Import + Avg_CO2 + Bensin_Co2 +
##     Diesel_Co2 + Quantity_Diesel + Quantity_Hybrid + Quantity_Electric
##
##                     Df Sum of Sq    RSS    AIC
## - Quantity_Hybrid    1     521.4 387886 634.61
## - Avg_CO2            1     946.1 388311 634.69
## - Import             1    1937.7 389303 634.87
## - Diesel_Co2         1    4050.2 391415 635.26
## <none>                           387365 636.51
```

```
## - Quantity           1   11076.5 398442 636.54
## - Quantity_Electric  1   15204.0 402569 637.28
## - Quantity_Diesel    1   16373.2 403738 637.49
## - Bensin_Co2         1   18326.9 405692 637.84
## + Diesel_Share       1     103.5 387262 638.49
##
## Step:  AIC=634.61
## Import_Electric ~ Quantity + Import + Avg_CO2 + Bensin_Co2 +
##     Diesel_Co2 + Quantity_Diesel + Quantity_Electric
##
##                     Df Sum of Sq    RSS    AIC
## - Avg_CO2            1    1452.1 389339 632.88
## - Import             1    3123.1 391009 633.19
## - Diesel_Co2         1    3758.6 391645 633.30
## <none>                           387886 634.61
## - Quantity_Electric  1   15632.2 403519 635.45
## - Quantity           1   19922.6 407809 636.22
## - Bensin_Co2         1   20140.3 408027 636.25
## + Quantity_Hybrid    1     521.4 387365 636.51
## + Diesel_Share       1     455.4 387431 636.53
## - Quantity_Diesel    1   26665.5 414552 637.40
##
## Step:  AIC=632.88
## Import_Electric ~ Quantity + Import + Bensin_Co2 + Diesel_Co2 +
##     Quantity_Diesel + Quantity_Electric
##
##                     Df Sum of Sq    RSS    AIC
## - Import             1      5168 394506 631.83
## <none>                           389339 632.88
## - Bensin_Co2         1     18871 408209 634.29
## + Avg_CO2            1      1452 387886 634.61
## - Diesel_Co2         1     20937 410275 634.65
## + Quantity_Hybrid    1      1027 388311 634.69
## + Diesel_Share       1        31 389308 634.87
## - Quantity           1     43245 432584 638.46
## - Quantity_Electric  1     53009 442348 640.07
## - Quantity_Diesel    1     71501 460839 643.02
##
## Step:  AIC=631.83
## Import_Electric ~ Quantity + Bensin_Co2 + Diesel_Co2 + Quantity_Diesel +
##     Quantity_Electric
##
##                     Df Sum of Sq    RSS    AIC
## <none>                           394506 631.83
## + Import             1      5168 389339 632.88
## - Bensin_Co2         1     18004 412511 633.04
## + Quantity_Hybrid    1      3596 390910 633.17
## + Avg_CO2            1      3497 391009 633.19
## - Diesel_Co2         1     21464 415970 633.64
## + Diesel_Share       1       491 394016 633.74
## - Quantity           1     38266 432773 636.49
## - Quantity_Diesel    1     70911 465417 641.73
## - Quantity_Electric  1     84481 478988 643.80
```

```
coefficients(backward)
```

```
##       (Intercept)          Quantity        Bensin_Co2        Diesel_Co2
##      403.49845785        0.02929057        4.44532147       -6.92873595
```

```
##   Quantity_Diesel Quantity_Electric
##       -0.04313930        0.11595169
```

```
BIC(backward)
```

```
## [1] 854.092
```

```
fitsummary=summary(backward)
fitsummary$r.squared
```

```
## [1] 0.8436755
```

```
PredBase<-predict(backward, validation, se.fit=TRUE)
y_1<-PredBase$fit
y<-validation$Import_Electric
MSE <- mean(y-y_1)^2
MSE
```

```
## [1] 19.47549
```

Working with Quantity Hybrid **Multiple Linear Regression** below are the results we get after we perform this analysis

```
###predict Quantity_Hybrid
trainfitQH<-lm(Quantity_Hybrid ~ Quantity + Import + Avg_CO2 + Bensin_Co2 +
Diesel_Co2 + Quantity_Diesel + Diesel_Share + Quantity_Electric +
Import_Electric, data = training)
summary(trainfitQH)
```

```
##
## Call:
## lm(formula = Quantity_Hybrid ~ Quantity + Import + Avg_CO2 +
##     Bensin_Co2 + Diesel_Co2 + Quantity_Diesel + Diesel_Share +
##     Quantity_Electric + Import_Electric, data = training)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -767.31 -104.33  -22.02   92.01 1778.04
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -8.160e+03  1.970e+03  -4.143 0.000106 ***
## Quantity           1.091e+00  1.441e-01   7.572 2.19e-10 ***
## Import            -1.733e-01  1.143e-01  -1.515 0.134739
## Avg_CO2           -8.903e+01  2.011e+01  -4.428 3.94e-05 ***
## Bensin_Co2         2.063e+01  1.050e+01   1.965 0.053932 .
## Diesel_Co2         5.436e+01  2.346e+01   2.317 0.023841 *
## Quantity_Diesel   -1.557e+00  2.260e-01  -6.889 3.35e-09 ***
## Diesel_Share       1.557e+02  2.869e+01   5.426 1.01e-06 ***
## Quantity_Electric -1.239e+00  2.157e-01  -5.745 2.99e-07 ***
## Import_Electric    8.027e-02  4.874e-01   0.165 0.869720
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 303.4 on 62 degrees of freedom
## Multiple R-squared:  0.9231,       Adjusted R-squared:  0.912
## F-statistic: 82.71 on 9 and 62 DF,  p-value: < 2.2e-16
```

```
backward <- step(trainfitQH, direction = 'both')
```

```
## Start:  AIC=832.19
## Quantity_Hybrid ~ Quantity + Import + Avg_CO2 + Bensin_Co2 +
##     Diesel_Co2 + Quantity_Diesel + Diesel_Share + Quantity_Electric +
```

```
##     Import_Electric
##
##                      Df Sum of Sq      RSS    AIC
## - Import_Electric    1      2496  5708822 830.22
## <none>                            5706326 832.19
## - Import             1    211374  5917699 832.81
## - Bensin_Co2         1    355278  6061604 834.54
## - Diesel_Co2         1    493959  6200285 836.17
## - Avg_CO2            1   1804387  7510713 849.97
## - Diesel_Share       1   2710098  8416424 858.17
## - Quantity_Electric  1   3037752  8744078 860.92
## - Quantity_Diesel    1   4367547 10073873 871.11
## - Quantity           1   5276644 10982970 877.33
##
## Step:  AIC=830.22
## Quantity_Hybrid ~ Quantity + Import + Avg_CO2 + Bensin_Co2 +
##     Diesel_Co2 + Quantity_Diesel + Diesel_Share + Quantity_Electric
##
##                      Df Sum of Sq      RSS    AIC
## <none>                            5708822 830.22
## - Import             1    215807  5924629 830.89
## + Import_Electric    1      2496  5706326 832.19
## - Bensin_Co2         1    388141  6096963 832.96
## - Diesel_Co2         1    491615  6200437 834.17
## - Avg_CO2            1   1822504  7531326 848.17
## - Diesel_Share       1   2718931  8427753 856.27
## - Quantity_Electric  1   3104725  8813547 859.49
## - Quantity_Diesel    1   4447592 10156414 869.70
## - Quantity           1   5396775 11105597 876.13
```

```
coefficients(backward)
```

```
##       (Intercept)          Quantity            Import           Avg_CO2
##     -8140.2804051         1.0944848        -0.1746465       -89.2551531
##        Bensin_Co2        Diesel_Co2   Quantity_Diesel      Diesel_Share
##        21.0137065        54.0145700        -1.5607422       155.8224878
## Quantity_Electric
##        -1.2326807
```

```
BIC(backward)
```

```
## [1] 1059.316
```

```
fitsummary=summary(backward)
fitsummary$r.squared
```

```
## [1] 0.9230837
```

```
PredBase<-predict(backward, validation, se.fit=TRUE)
y_1<-PredBase$fit
y<-validation$Quantity_Hybrid

MSE <- mean(y-y_1)^2
MSE
```

```
## [1] 18.0609
```

Working with Import **Multiple Linear Regression** below are the results we get after we perform this analysis

```
###predict Import
```

```
trainfitIm<-lm(Import ~ Quantity + Avg_CO2 + Bensin_Co2 + Diesel_Co2 + Quantity_Diesel + Diesel_Share + Quantity_Hybrid + Quantity_Electric + Import_Electric, data = training)
summary(trainfitIm)
```

```
##
## Call:
## lm(formula = Import ~ Quantity + Avg_CO2 + Bensin_Co2 + Diesel_Co2 +
##     Quantity_Diesel + Diesel_Share + Quantity_Hybrid + Quantity_Electric +
##     Import_Electric, data = training)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -719.52 -212.71   24.48  220.21  595.08
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3100.83990 2395.41872   1.294   0.2003
## Quantity             0.10615    0.21768   0.488   0.6275
## Avg_CO2             49.83550   24.34809   2.047   0.0449 *
## Bensin_Co2           3.08590   11.79451   0.262   0.7945
## Diesel_Co2         -46.93349   25.99955  -1.805   0.0759 .
## Quantity_Diesel      0.16393    0.32677   0.502   0.6177
## Diesel_Share       -51.71182   37.42291  -1.382   0.1720
## Quantity_Hybrid     -0.20612    0.13601  -1.515   0.1347
## Quantity_Electric   -0.09626    0.29090  -0.331   0.7418
## Import_Electric     -0.27904    0.53050  -0.526   0.6008
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 330.9 on 62 degrees of freedom
## Multiple R-squared:  0.6774,       Adjusted R-squared:  0.6305
## F-statistic: 14.46 on 9 and 62 DF,  p-value: 2.886e-12
```

```
backward <- step(trainfitIm, direction = 'both')
```

```
## Start:  AIC=844.68
## Import ~ Quantity + Avg_CO2 + Bensin_Co2 + Diesel_Co2 + Quantity_Diesel +
##     Diesel_Share + Quantity_Hybrid + Quantity_Electric + Import_Electric
##
##                     Df Sum of Sq     RSS    AIC
## - Bensin_Co2         1      7494 6794789 842.76
## - Quantity_Electric  1     11986 6799282 842.81
## - Quantity           1     26034 6813330 842.96
## - Quantity_Diesel    1     27551 6814846 842.97
## - Import_Electric    1     30289 6817585 843.00
## <none>                           6787295 844.68
## - Diesel_Share       1    209030 6996326 844.86
## - Quantity_Hybrid    1    251415 7038710 845.30
## - Diesel_Co2         1    356730 7144025 846.37
## - Avg_CO2            1    458620 7245915 847.39
##
## Step:  AIC=842.76
## Import ~ Quantity + Avg_CO2 + Diesel_Co2 + Quantity_Diesel +
##     Diesel_Share + Quantity_Hybrid + Quantity_Electric + Import_Electric
##
##                     Df Sum of Sq     RSS    AIC
## - Quantity_Electric  1      9555 6804345 840.86
## - Quantity           1     21674 6816463 840.99
```

```
## - Import_Electric    1     25408 6820197 841.03
## - Quantity_Diesel    1     35862 6830652 841.14
## <none>                           6794789 842.76
## - Diesel_Share       1    233210 7028000 843.19
## - Quantity_Hybrid    1    245141 7039931 843.31
## - Diesel_Co2         1    362153 7156942 844.50
## + Bensin_Co2         1      7494 6787295 844.68
## - Avg_CO2            1    542893 7337683 846.29
##
## Step:  AIC=840.86
## Import ~ Quantity + Avg_CO2 + Diesel_Co2 + Quantity_Diesel +
##     Diesel_Share + Quantity_Hybrid + Import_Electric
##
##                     Df Sum of Sq     RSS    AIC
## - Quantity           1     12670 6817015 839.00
## - Import_Electric    1     33186 6837530 839.21
## - Quantity_Diesel    1     77021 6881366 839.67
## <none>                           6804345 840.86
## - Quantity_Hybrid    1    289777 7094122 841.86
## + Quantity_Electric  1      9555 6794789 842.76
## + Bensin_Co2         1      5063 6799282 842.81
## - Diesel_Share       1    402466 7206810 843.00
## - Diesel_Co2         1   1310805 8115150 851.55
## - Avg_CO2            1   2545054 9349399 861.74
##
## Step:  AIC=839
## Import ~ Avg_CO2 + Diesel_Co2 + Quantity_Diesel + Diesel_Share +
##     Quantity_Hybrid + Import_Electric
##
##                     Df Sum of Sq      RSS    AIC
## - Import_Electric    1     27455  6844470 837.28
## <none>                            6817015 839.00
## - Quantity_Hybrid    1    322329  7139344 840.32
## + Quantity           1     12670  6804345 840.86
## + Bensin_Co2         1      2994  6814021 840.96
## + Quantity_Electric  1       552  6816463 840.99
## - Diesel_Co2         1   1326861  8143876 849.80
## - Avg_CO2            1   2542175  9359190 859.81
## - Diesel_Share       1   3986690 10803705 870.15
## - Quantity_Diesel    1   4788822 11605837 875.31
##
## Step:  AIC=837.28
## Import ~ Avg_CO2 + Diesel_Co2 + Quantity_Diesel + Diesel_Share +
##     Quantity_Hybrid
##
##                     Df Sum of Sq      RSS    AIC
## <none>                            6844470 837.28
## - Quantity_Hybrid    1    337668  7182138 838.75
## + Import_Electric    1     27455  6817015 839.00
## + Quantity           1      6940  6837530 839.21
## + Quantity_Electric  1      4191  6840279 839.24
## + Bensin_Co2         1       295  6844175 839.28
## - Diesel_Co2         1   1596953  8441423 850.38
## - Avg_CO2            1   3206173 10050643 862.95
## - Diesel_Share       1   3964152 10808622 868.18
## - Quantity_Diesel    1   4764035 11608505 873.32
```

```
coefficients(backward)
```

```
##     (Intercept)         Avg_CO2      Diesel_Co2 Quantity_Diesel
##    3950.5961220      59.3371397     -51.5419725       0.3211110
##    Diesel_Share Quantity_Hybrid
##     -68.5358491      -0.1583739
```

```
BIC(backward)
```

```
## [1] 1059.548
```

```
fitsummary=summary(backward)
fitsummary$r.squared
```

```
## [1] 0.6746608
```

```
PredBase<-predict(backward, validation, se.fit=TRUE)
y_1<-PredBase$fit
y<-validation$Import
MSE <- mean(y-y_1)^2
MSE
```

```
## [1] 1604.715
```

Working with Quantity Diesel **Multiple Linear Regression** below are the results we get after we perform this analysis

```
###predict Quantity_Diesel
trainfitQD<-lm(Quantity_Diesel ~ Quantity + Import + Avg_CO2 + Bensin_Co2 +
Diesel_Co2 + Diesel_Share + Quantity_Hybrid + Quantity_Electric +
Import_Electric, data = training)
summary(trainfit)
```

```
##
## Call:
## lm(formula = Quantity ~ Import + Avg_CO2 + Bensin_Co2 + Diesel_Co2 +
##     Quantity_Diesel + Diesel_Share + Quantity_Hybrid + Quantity_Electric +
##     Import_Electric, data = training)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -616.70  -76.33   -1.28  101.36  563.36
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       9087.44302  816.30707  11.132  < 2e-16 ***
## Import               0.03600    0.07381   0.488   0.6275
## Avg_CO2             45.99104   13.43475   3.423   0.0011 **
## Bensin_Co2         -10.70878    6.73602  -1.590   0.1170
## Diesel_Co2         -24.15372   15.22686  -1.586   0.1178
## Quantity_Diesel      1.47214    0.03741  39.355  < 2e-16 ***
## Diesel_Share      -161.00991    8.44900 -19.057  < 2e-16 ***
## Quantity_Hybrid      0.44020    0.05814   7.572 2.19e-10 ***
## Quantity_Electric    0.72445    0.14241   5.087 3.61e-06 ***
## Import_Electric      0.19202    0.30864   0.622   0.5361
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 192.7 on 62 degrees of freedom
## Multiple R-squared:  0.9897,	Adjusted R-squared:  0.9882
## F-statistic: 662.5 on 9 and 62 DF,  p-value: < 2.2e-16
```

```
backward <- step(trainfitQD, direction = 'both')
```

```
## Start:  AIC=708.3
```

```
## Quantity_Diesel ~ Quantity + Import + Avg_CO2 + Bensin_Co2 +
##     Diesel_Co2 + Diesel_Share + Quantity_Hybrid + Quantity_Electric +
##     Import_Electric
##
##                     Df Sum of Sq      RSS    AIC
## - Import             1      4145  1025254 706.59
## - Import_Electric    1      5177  1026287 706.66
## <none>                            1021110 708.30
## - Diesel_Co2         1     40489  1061599 709.10
## - Bensin_Co2         1     49505  1070615 709.71
## - Avg_CO2            1    229373  1250482 720.89
## - Quantity_Electric  1    375335  1396445 728.84
## - Quantity_Hybrid    1    781544  1802653 747.22
## - Diesel_Share       1  12511061 13532171 892.36
## - Quantity           1  25508699 26529808 940.83
##
## Step:  AIC=706.59
## Quantity_Diesel ~ Quantity + Avg_CO2 + Bensin_Co2 + Diesel_Co2 +
##     Diesel_Share + Quantity_Hybrid + Quantity_Electric + Import_Electric
##
##                     Df Sum of Sq      RSS    AIC
## - Import_Electric    1      5891  1031145 705.01
## <none>                            1025254 706.59
## - Diesel_Co2         1     36806  1062061 707.13
## - Bensin_Co2         1     51011  1076266 708.09
## + Import             1      4145  1021110 708.30
## - Avg_CO2            1    228425  1253679 719.07
## - Quantity_Electric  1    385268  1410522 727.56
## - Quantity_Hybrid    1    902150  1927404 750.04
## - Diesel_Share       1  14780612 15805866 901.54
## - Quantity           1  43360837 44386092 975.89
##
## Step:  AIC=705.01
## Quantity_Diesel ~ Quantity + Avg_CO2 + Bensin_Co2 + Diesel_Co2 +
##     Diesel_Share + Quantity_Hybrid + Quantity_Electric
##
##                     Df Sum of Sq      RSS    AIC
## <none>                            1031145 705.01
## - Diesel_Co2         1     39983  1071129 705.74
## - Bensin_Co2         1     46172  1077317 706.16
## + Import_Electric    1      5891  1025254 706.59
## + Import             1      4859  1026287 706.66
## - Avg_CO2            1    225796  1256941 717.26
## - Quantity_Electric  1    431342  1462488 728.17
## - Quantity_Hybrid    1    932468  1963613 749.38
## - Diesel_Share       1  15361487 16392632 902.17
## - Quantity           1  43359286 44390431 973.89
```

```
coefficients(backward)
```

```
##       (Intercept)          Quantity           Avg_CO2        Bensin_Co2
##     -5952.9942960         0.6584051       -31.5954559         7.3262902
##        Diesel_Co2      Diesel_Share   Quantity_Hybrid Quantity_Electric
##        15.3696161       111.1380149        -0.2874871        -0.4786877
```

```
BIC(backward)
```

```
## [1] 931.8222
```

```
fitsummary=summary(backward)
fitsummary$r.squared
```

```
## [1] 0.9942721
```

```
PredBase<-predict(backward, validation, se.fit=TRUE)

y_1<-PredBase$fit

y<-validation$Quantity_Diesel

MSE <- mean(y-y_1)^2
MSE
```

```
## [1] 1.725701
```

Working with Green Cars **Multiple Linear Regression** below are the results we get after we perform this analysis

```
###predict Green = Import Electric +Quantity Electric + Hybrid Vehicles
trainfitGreen<-lm( Quantity_Hybrid + Quantity_Electric + Import_Electric ~
Import + Avg_CO2 + Bensin_Co2 + Diesel_Co2 + Quantity_Diesel + Diesel_Share
+Quantity, data = training)
summary(trainfitGreen)
```

```
##
## Call:
## lm(formula = Quantity_Hybrid + Quantity_Electric + Import_Electric ~
##     Import + Avg_CO2 + Bensin_Co2 + Diesel_Co2 + Quantity_Diesel +
##         Diesel_Share + Quantity, data = training)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -794.60 -123.75  -11.76  122.71 1732.60
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -7392.2795  1891.5601  -3.908 0.000227 ***
## Import             -0.1978     0.1167  -1.695 0.095012 .
## Avg_CO2           -80.8540    13.0547  -6.193 4.72e-08 ***
## Bensin_Co2         25.4606    10.4910   2.427 0.018055 *
## Diesel_Co2         38.6266    18.1212   2.132 0.036888 *
## Quantity_Diesel    -1.5739     0.2238  -7.034 1.62e-09 ***
## Diesel_Share      151.3862    27.9805   5.410 1.00e-06 ***
## Quantity            1.1058     0.1417   7.804 7.13e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 311.4 on 64 degrees of freedom
## Multiple R-squared:  0.9769,	Adjusted R-squared:  0.9743
## F-statistic: 385.9 on 7 and 64 DF,  p-value: < 2.2e-16
```

```
backward <- step(trainfitGreen, direction = 'both')
```

```
## Start:  AIC=834.25
## Quantity_Hybrid + Quantity_Electric + Import_Electric ~ Import +
##     Avg_CO2 + Bensin_Co2 + Diesel_Co2 + Quantity_Diesel + Diesel_Share +
##     Quantity
##
##                   Df Sum of Sq     RSS    AIC
## <none>                          6207003 834.25
## - Import           1    278508  6485511 835.41
## - Diesel_Co2       1    440656  6647659 837.18
```

```
## - Bensin_Co2       1    571219  6778222 838.58
## - Diesel_Share     1   2838991  9045994 859.36
## - Avg_CO2          1   3720268  9927271 866.06
## - Quantity_Diesel  1   4798578 11005581 873.48
## - Quantity         1   5907067 12114070 880.39
```

```
coefficients(backward)
```

```
##     (Intercept)          Import         Avg_CO2      Bensin_Co2
##   -7392.2795114      -0.1978261     -80.8539821      25.4605654
##      Diesel_Co2 Quantity_Diesel    Diesel_Share        Quantity
##      38.6265961      -1.5738992     151.3861910       1.1058103
```

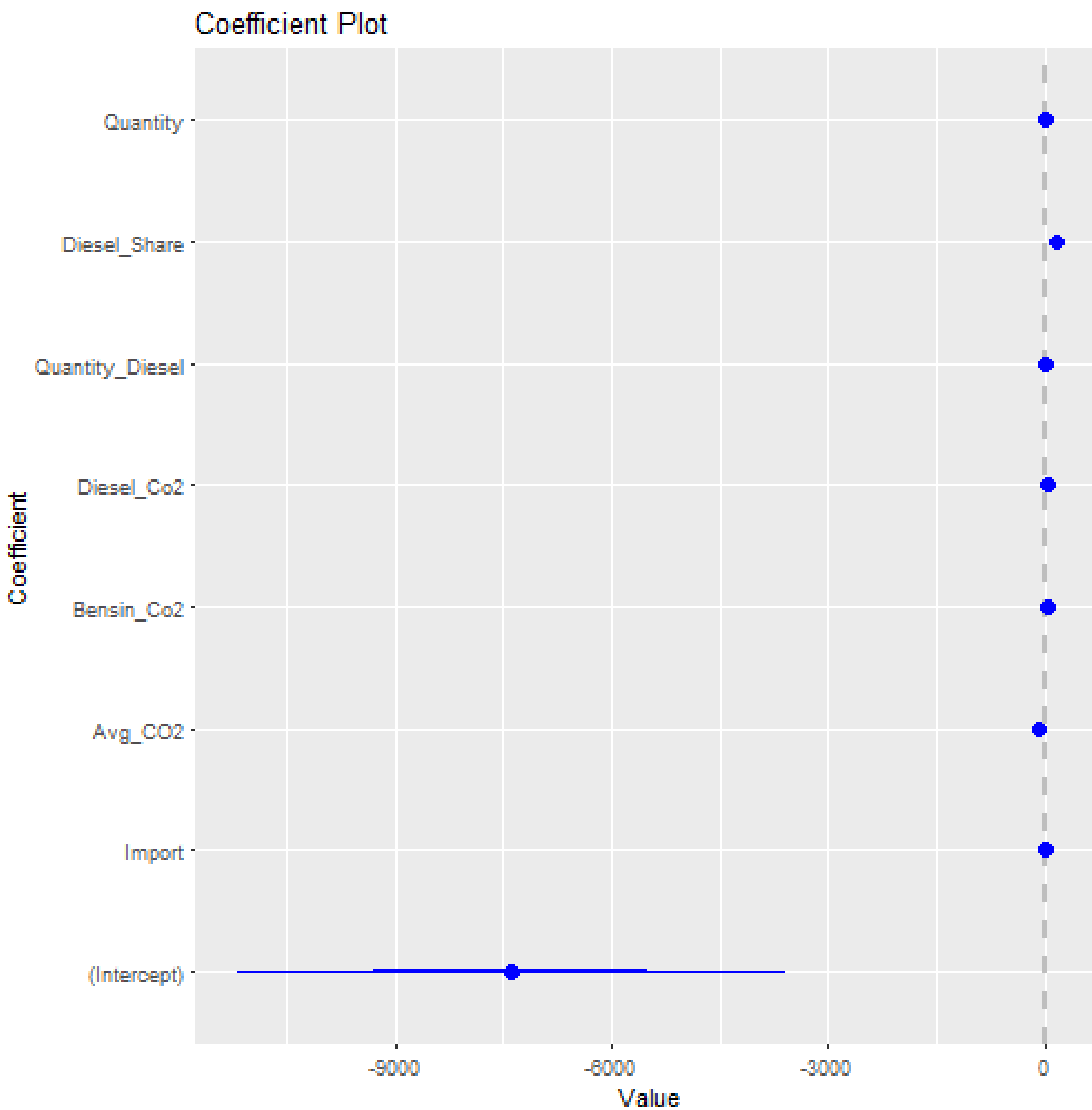
```
require(coefplot)
```

```
## Loading required package: coefplot
```

```
## Loading required package: ggplot2
```

```
coefplot(trainfitGreen)
```

```
## Warning: Ignoring unknown aesthetics: xmin, xmax
```




```
head(fortify(trainfitGreen))
```

```
##     Quantity_Hybrid + Quantity_Electric + Import_Electric Import Avg_CO2
## 88                                                   2408   1997     114
## 106                                                  4770   1808      99
## 91                                                   2523   2625     111
## 105                                                  4256   1899     100
## 54                                                    324   2220     135
## 20                                                    164   2358     157
##     Bensin_Co2 Diesel_Co2 Quantity_Diesel Diesel_Share Quantity        .hat
## 88         121        134            6021         49.7    12115 0.03196792
## 106        118        131            5239         39.7    13197 0.06058584
## 91         119        133            5670         48.5    11690 0.10190232
```

```
## 105        119        131            4881         39.3    12421 0.06373811
## 54         134        137            8024         77.5    10354 0.11458780
## 20         159        157            5434         69.4     7833 0.10388349
##       .sigma      .cooksd   .fitted      .resid   .stdresid
## 88  311.7060 3.655524e-03 2696.3390 -288.33900 -0.94103877
## 106 313.6032 9.261841e-04 4667.6904  102.30956  0.33895079
## 91  313.5867 1.724964e-03 2625.9242 -102.92421 -0.34874239
## 105 313.8774 2.680589e-05 4239.0875   16.91254  0.05612537
## 54  312.8973 6.506386e-03  509.8413 -185.84129 -0.63418864
## 20  313.8816 2.099980e-05  175.2226  -11.22263 -0.03806813
```

```
BIC(backward)
```

```
## [1] 1061.063
```

```
fitsummary=summary(backward)
fitsummary$r.squared
```

```
## [1] 0.9768535
```

```
PredBase<-predict(backward, validation, se.fit=TRUE)

y_1<-PredBase$fit

y<-
(validation$Import_Electric+validation$Quantity_Hybrid+validation$Quantity_El
ectric)

MSE <- mean(y-y_1)^2
MSE
```

```
## [1] 1.923739
```

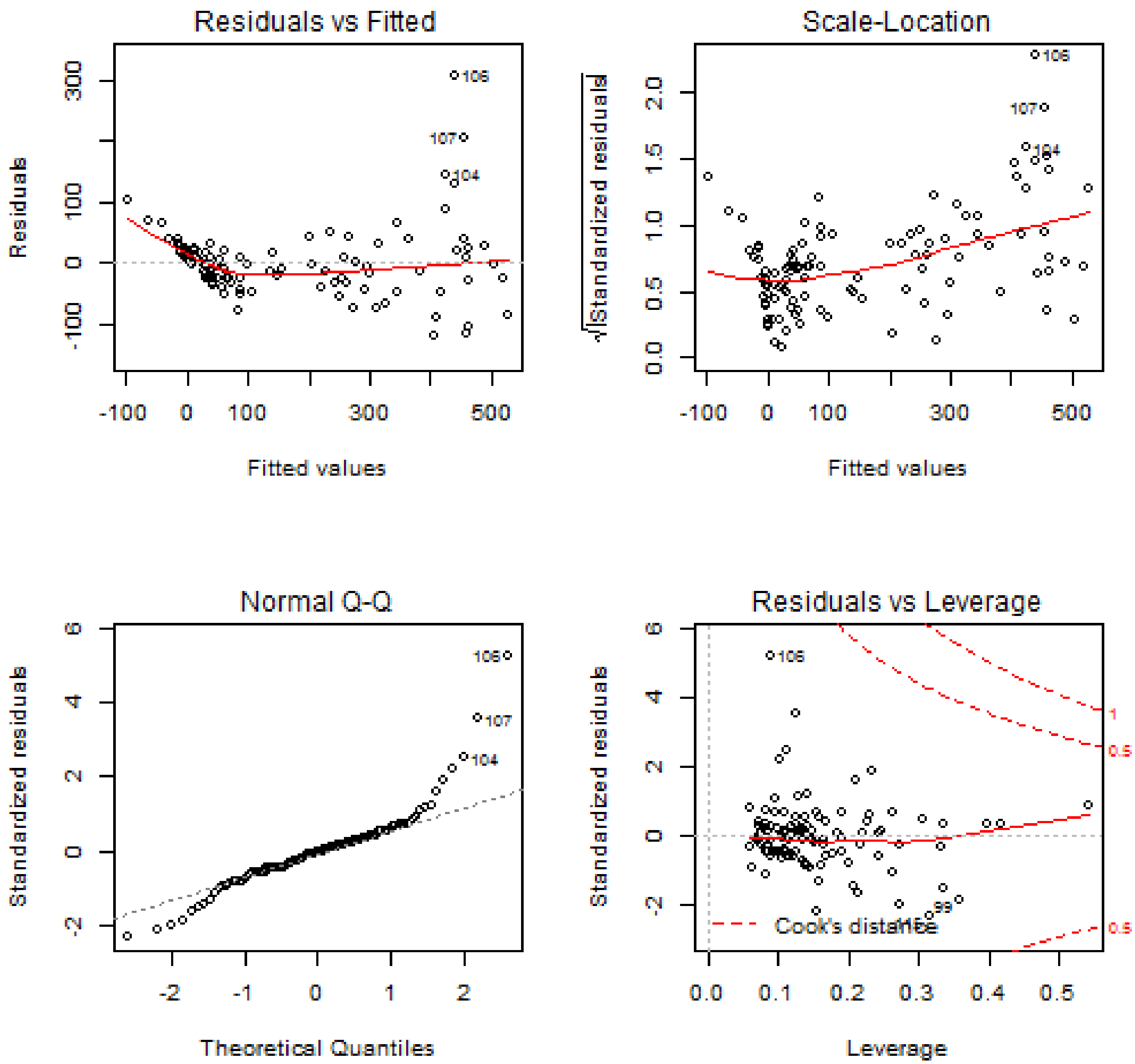
Working with Import Electric **Multiple Linear Regression** below are the results we get after we perform this analysis

```
ncbm<-lm(Import_Electric~.,na.action = na.omit,data=carsbymonth)
summary(ncbm)
```

```
##
## Call:
## lm(formula = Import_Electric ~ ., data = carsbymonth, na.action = na.omit)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -121.257  -27.502   -1.706   19.413  306.746
##
## Coefficients: (1 not defined because of singularities)
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -4.729e+04  4.252e+04  -1.112 0.268956
## Year              2.382e+01  2.094e+01   1.137 0.258307
## Month             1.132e+01  2.007e+00   5.640 1.85e-07 ***
## Quantity          1.332e-02  3.541e-02   0.376 0.707712
## Quantity_YoY     -5.954e-03  7.106e-03  -0.838 0.404282
## Import           -6.614e-02  2.971e-02  -2.226 0.028457 *
## Import_YoY        2.242e-02  2.147e-02   1.044 0.299034
## Used              3.482e-03  2.432e-03   1.431 0.155679
## Used_YoY         -1.645e-03  3.768e-03  -0.437 0.663473
## Avg_CO2           7.225e+00  4.463e+00   1.619 0.108922
## Bensin_Co2        5.439e+00  1.834e+00   2.966 0.003842 **
## Diesel_Co2       -1.183e+01  4.244e+00  -2.787 0.006460 **
## Quantity_Diesel  -1.479e-03  5.269e-02  -0.028 0.977673
```

```
## Diesel_Share      -4.896e+00  6.347e+00  -0.771 0.442472
## Diesel_Share_LY   -6.264e+00  1.790e+00  -3.499 0.000723 ***
## Quantity_Hybrid    3.335e-03  2.371e-02   0.141 0.888481
## Quantity_Electric  6.529e-02  4.511e-02   1.447 0.151249
## MonthNumber               NA          NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 61.39 on 92 degrees of freedom
##   (12 observations deleted due to missingness)
## Multiple R-squared:  0.9054,	Adjusted R-squared:  0.889
## F-statistic: 55.05 on 16 and 92 DF,  p-value: < 2.2e-16
```

```
layout(matrix(c(1,2,3,4),2,2))
plot(ncbm)
```



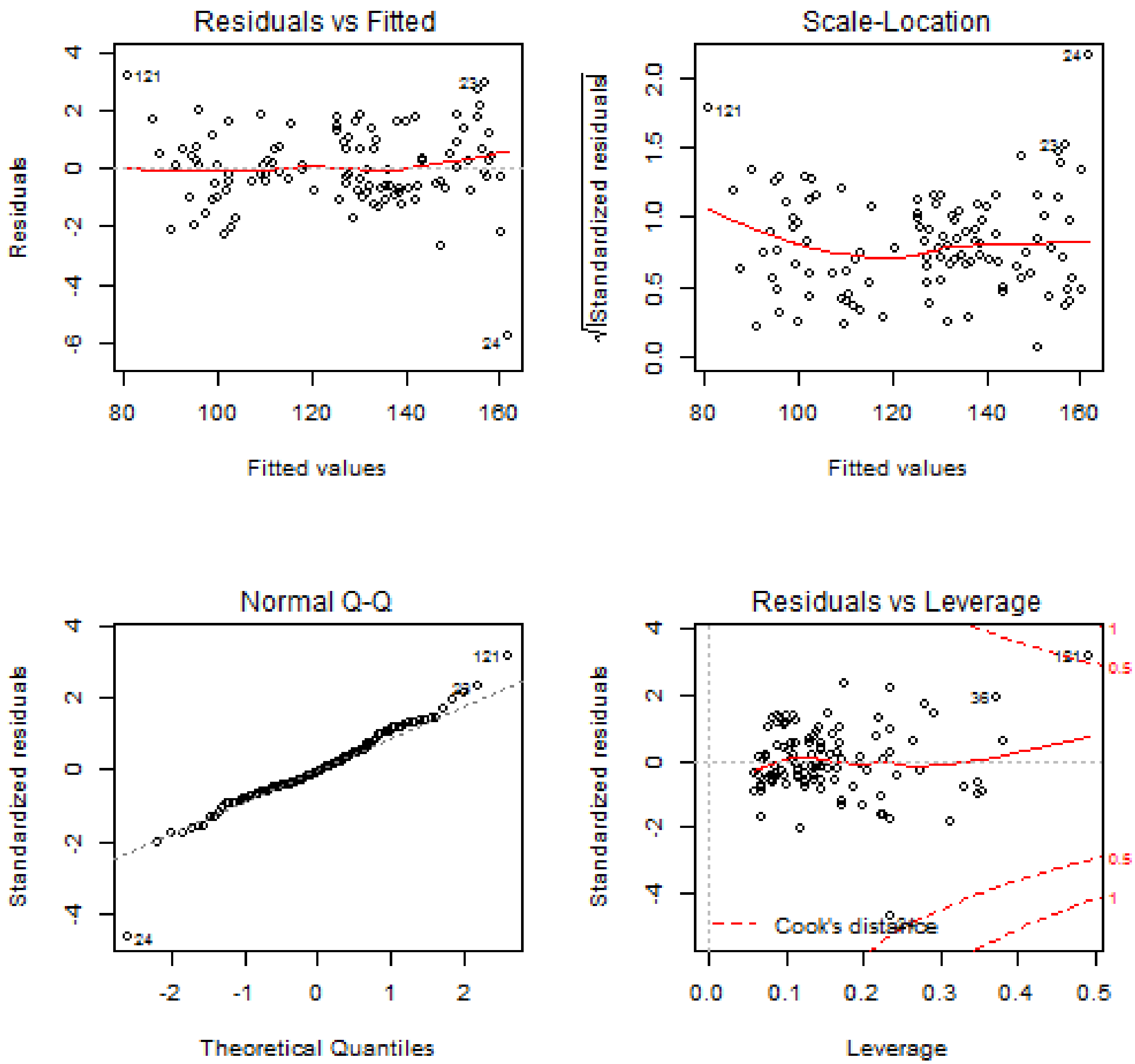
Working with Avg Co2 **Multiple Linear Regression** below are the results we get after we perform this analysis

```
ncbm_Avgco2<-lm(Avg_CO2~.,na.action = na.omit,data = carsbymonth)
summary(ncbm_Avgco2)
```

```
##
## Call:
## lm(formula = Avg_CO2 ~ ., data = carsbymonth, na.action = na.omit)
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -5.7791 -0.7665 -0.1049  0.7151  3.1880
##
## Coefficients: (1 not defined because of singularities)
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        5.276e+03  8.182e+02   6.449 5.15e-09 ***
## Year              -2.617e+00  4.017e-01  -6.515 3.82e-09 ***
## Month             -2.013e-01  4.934e-02  -4.079 9.62e-05 ***
## Quantity           2.604e-03  7.697e-04   3.383 0.001055 **
## Quantity_YoY      -2.079e-04  1.629e-04  -1.276 0.205002
## Import             1.393e-03  6.873e-04   2.027 0.045604 *
## Import_YoY         2.453e-04  4.967e-04   0.494 0.622630
## Used              -1.175e-04  5.530e-05  -2.125 0.036271 *
## Used_YoY           1.387e-04  8.567e-05   1.619 0.108773
## Bensin_Co2        -2.472e-02  4.413e-02  -0.560 0.576767
## Diesel_Co2         6.563e-01  7.538e-02   8.706 1.21e-13 ***
## Quantity_Diesel   -4.014e-03  1.139e-03  -3.523 0.000666 ***
## Diesel_Share       5.275e-01  1.360e-01   3.880 0.000196 ***
## Diesel_Share_LY    5.577e-02  4.351e-02   1.282 0.203077
## Quantity_Hybrid   -1.976e-03  5.059e-04  -3.906 0.000179 ***
## Quantity_Electric -8.325e-03  5.924e-04 -14.053  < 2e-16 ***
## Import_Electric    3.833e-03  2.368e-03   1.619 0.108922
## MonthNumber               NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.414 on 92 degrees of freedom
##   (12 observations deleted due to missingness)
## Multiple R-squared:  0.9963,	Adjusted R-squared:  0.9957
## F-statistic:  1566 on 16 and 92 DF,  p-value: < 2.2e-16
```

```
layout(matrix(c(1,2,3,4),2,2))
plot(ncbm_Avgco2)
```



Appendix C :Exponential Smoothing and Time Series

For this analysis we are using **Exponential Smoothing** below are the packages we need to install in order to perform this analysis

```
install.packages("tidyverse")
## Installing package into 'C:/Users/BinalAmit/Documents/R/win-library/3.3'
## (as 'lib' is unspecified)
```

```
## package 'tidyverse' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##      C:\Users\BinalAmit\AppData\Local\Temp\RtmpSe6Rfq\downloaded_packages
```

```
install.packages("forcats")
```

```
## Installing package into 'C:/Users/BinalAmit/Documents/R/win-library/3.3'
## (as 'lib' is unspecified)
```

```
## package 'forcats' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##      C:\Users\BinalAmit\AppData\Local\Temp\RtmpSe6Rfq\downloaded_packages
```

```
install.packages("smooth")
```

```
## Installing package into 'C:/Users/BinalAmit/Documents/R/win-library/3.3'
## (as 'lib' is unspecified)
```

```
## package 'smooth' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##      C:\Users\BinalAmit\AppData\Local\Temp\RtmpSe6Rfq\downloaded_packages
```

```
library(tidyverse)
```

```
## Loading tidyverse: tibble
## Loading tidyverse: tidyr
## Loading tidyverse: readr
## Loading tidyverse: purrr
## Loading tidyverse: dplyr
```

```
## Conflicts with tidy packages -----------------------------------------------
```

```
## filter(): dplyr, stats
## lag():    dplyr, stats
```

```
library(forcats)
library.path <- cat(.libPaths())
```

```
## C:/Users/BinalAmit/Documents/R/win-library/3.3 C:/Program Files/Microsoft/R
Open/library
```

```
library(smooth, lib.loc = library.path)
```

```
## This is package "smooth", v1.7.0
```

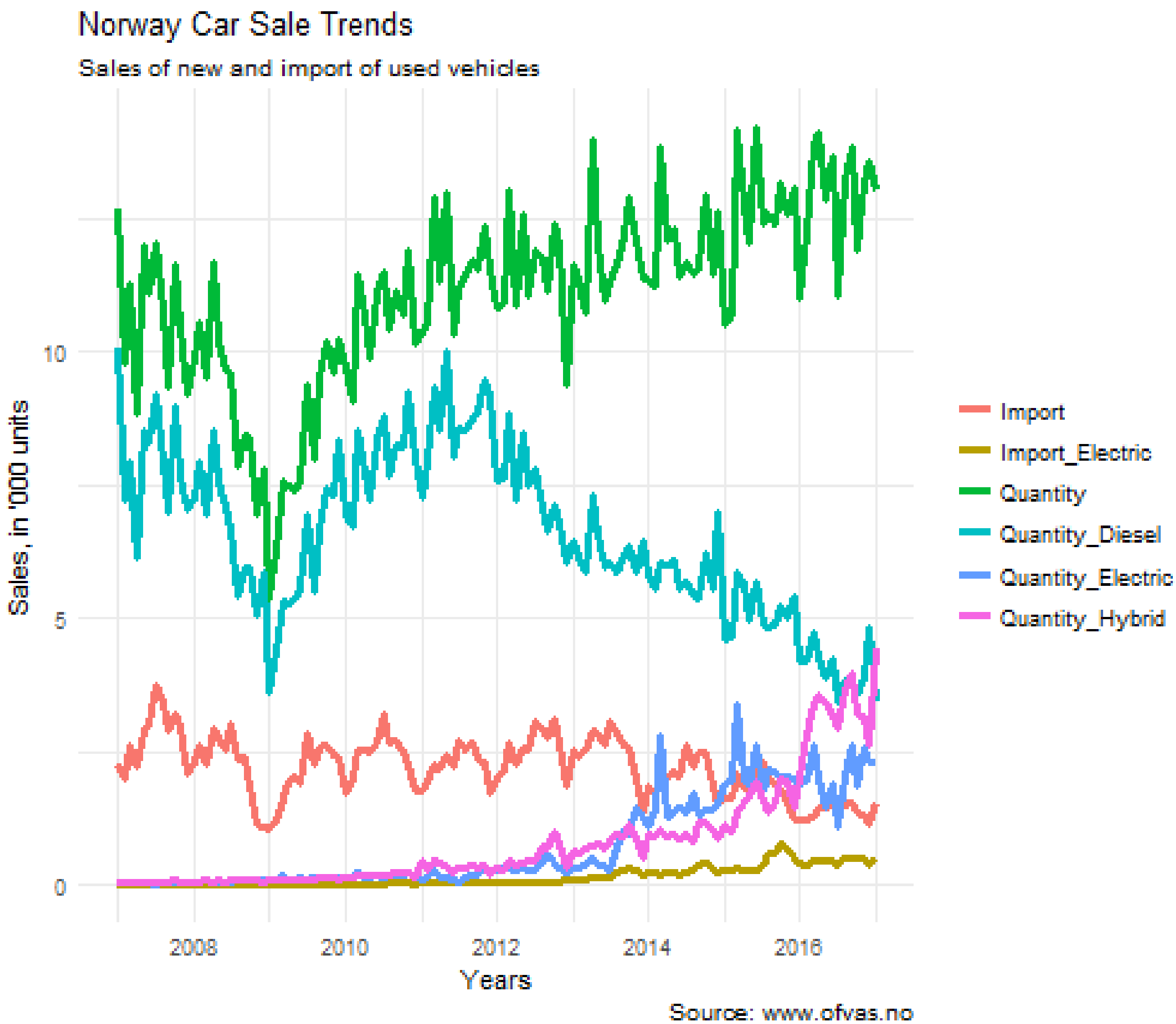
Following code **imports the file and plots** all relevant columns given that Working Directory has been set up to be same as R code folder:

```
carsbymonth<-read.csv("norway_new_car_sales_by_month_datab.csv",header=TRUE)
glimpse(carsbymonth)
```

```
## Observations: 121
## Variables: 18
## $ Year              <int> 2007, 2007, 2007, 2007, 2007, 2007, 2007, 20...
## $ Month             <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 1, 2,...
## $ Quantity          <int> 12685, 9793, 11264, 8854, 12007, 11083, 1206...
## $ Quantity_YoY      <int> 5227, 2448, 1445, 504, 1592, 1545, 1908, 199...
## $ Import            <int> 2276, 1992, 2626, 2220, 2881, 3038, 3768, 34...
## $ Import_YoY        <int> 257, -89, 45, -130, 7, 23, 137, 260, -28, 59...
## $ Used              <int> 34976, 32952, 34684, 31834, 34328, 38085, 40...
## $ Used_YoY          <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ...
## $ Avg_CO2           <int> 152, 156, 159, 160, 160, 161, 159, 160, 160,...
## $ Bensin_Co2        <int> 155, 159, 161, 165, 163, 163, 161, 160, 160,...
## $ Diesel_Co2        <int> 152, 155, 158, 158, 159, 160, 158, 160, 160,...
## $ Quantity_Diesel   <int> 10072, 7222, 7965, 6116, 8519, 8290, 9203, 7...
## $ Diesel_Share      <dbl> 79.4, 73.7, 70.7, 69.1, 71.0, 74.8, 76.3, 73...
```

```
## $ Diesel_Share_LY   <dbl> 52.5, 47.4, 48.1, 48.4, 49.1, 49.5, 50.1, 50...
## $ Quantity_Hybrid   <int> 42, 35, 48, 46, 47, 41, 47, 58, 59, 73, 60, ...
## $ Quantity_Electric <int> 25, 27, 36, 30, 27, 27, 21, 42, 56, 48, 63, ...
## $ Import_Electric   <int> 2, 2, 2, 2, 2, 2, 2, 3, 3, 4, 3, 3, 3, 3, 3,...
## $ MonthNumber       <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 1...
```

```
carsbymonth %>% mutate(Date=as.Date(paste(Year, Month, "1", sep="-"))) %>%
  select(Date,                          Quantity,Quantity_Electric,Quantity_Hybrid,
Quantity_Diesel,Import,Import_Electric) %>%
  gather(key=type, value=value, -Date) %>%
  ggplot()+
  geom_line(mapping = aes(x=Date, y=value/10^3, color=type), size=1.1)+
  theme_minimal()+
  labs(y="Sales, in '000 units",
       x="Years",
       color= NULL,
       title="Norway Car Sale Trends",
       subtitle="Sales of new and import of used vehicles",
       caption="Source: www.ofvas.no")
```



Following code checks for all NA Values

```
apply(carsbymonth,1,function(x) sum(is.na(x)))
```

```
##   [1] 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [36] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [71] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [106] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

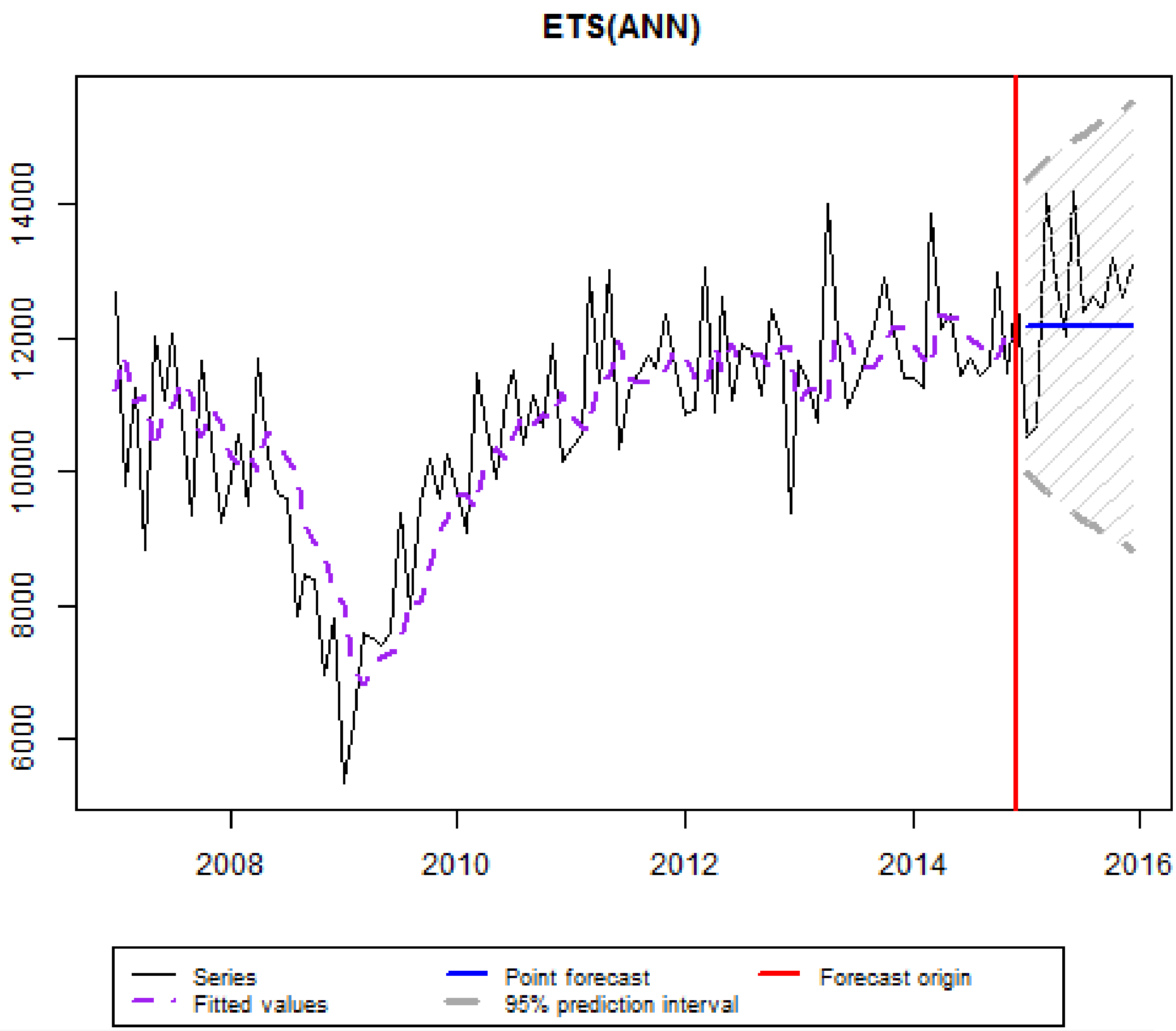
In the code that follows we Install the ES Package - Exponential Smoothing. Then, we perform creation of Exponential Smoothing object for Quantity using ZZZ Model Summary of the Model created is displayed and plotted:

```
install.packages("ES")
```

```
## Installing package into 'C:/Users/BinalAmit/Documents/R/win-library/3.3'
## (as 'lib' is unspecified)
```

```
## package 'ES' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##     C:\Users\BinalAmit\AppData\Local\Temp\RtmpSe6Rfq\downloaded_packages
```

```
library(smooth)
library(ES)
es_obj_ZZZ <- carsbymonth %>% filter(Year<=2015) %>%
  select(Quantity) %>% unlist() %>%
  ts(start=2007, f=12) %>%
  es(model="ZZZ", h=12, holdout=T, intervals=T, silent="output")
```

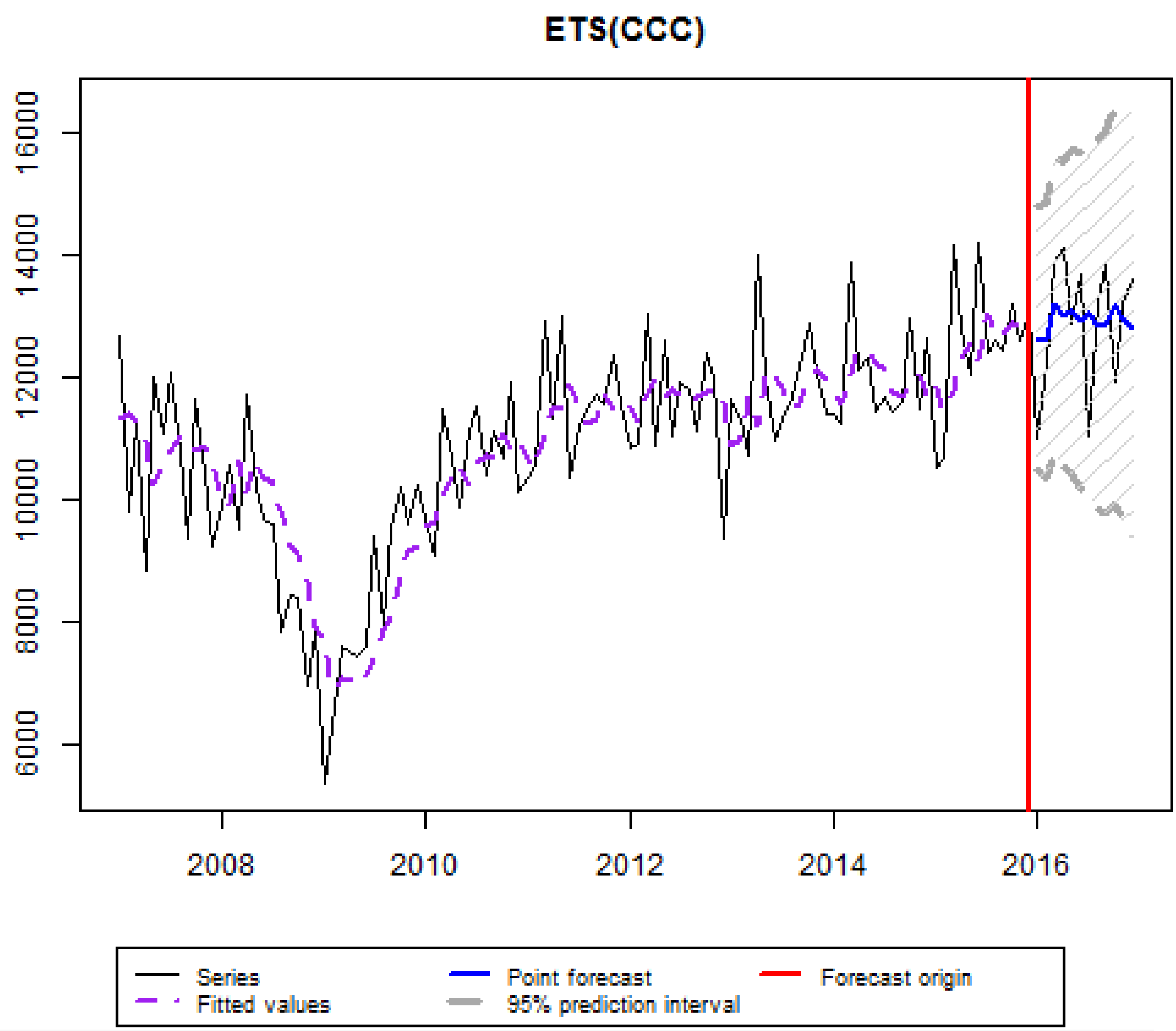



```
summary(es_obj_ZZZ)
## Time elapsed: 0.33 seconds
## Model estimated: ETS(ANN)
## Persistence vector g:
## alpha
## 0.348
## Initial values were optimised.
## 3 parameters were estimated in the process
## Residuals standard deviation: 1102.852
## Cost function type: MSE; Cost function value: 1178274
##
## Information criteria:
##      AIC     AICc      BIC
## 1620.474 1620.735 1628.167
## 95% parametric prediction intervals were constructed
## 100% of values are in the prediction interval
## Forecast errors:
## MPE: 2.4%; Bias: 55.9%; MAPE: 7.5%; SMAPE: 7.5%
```

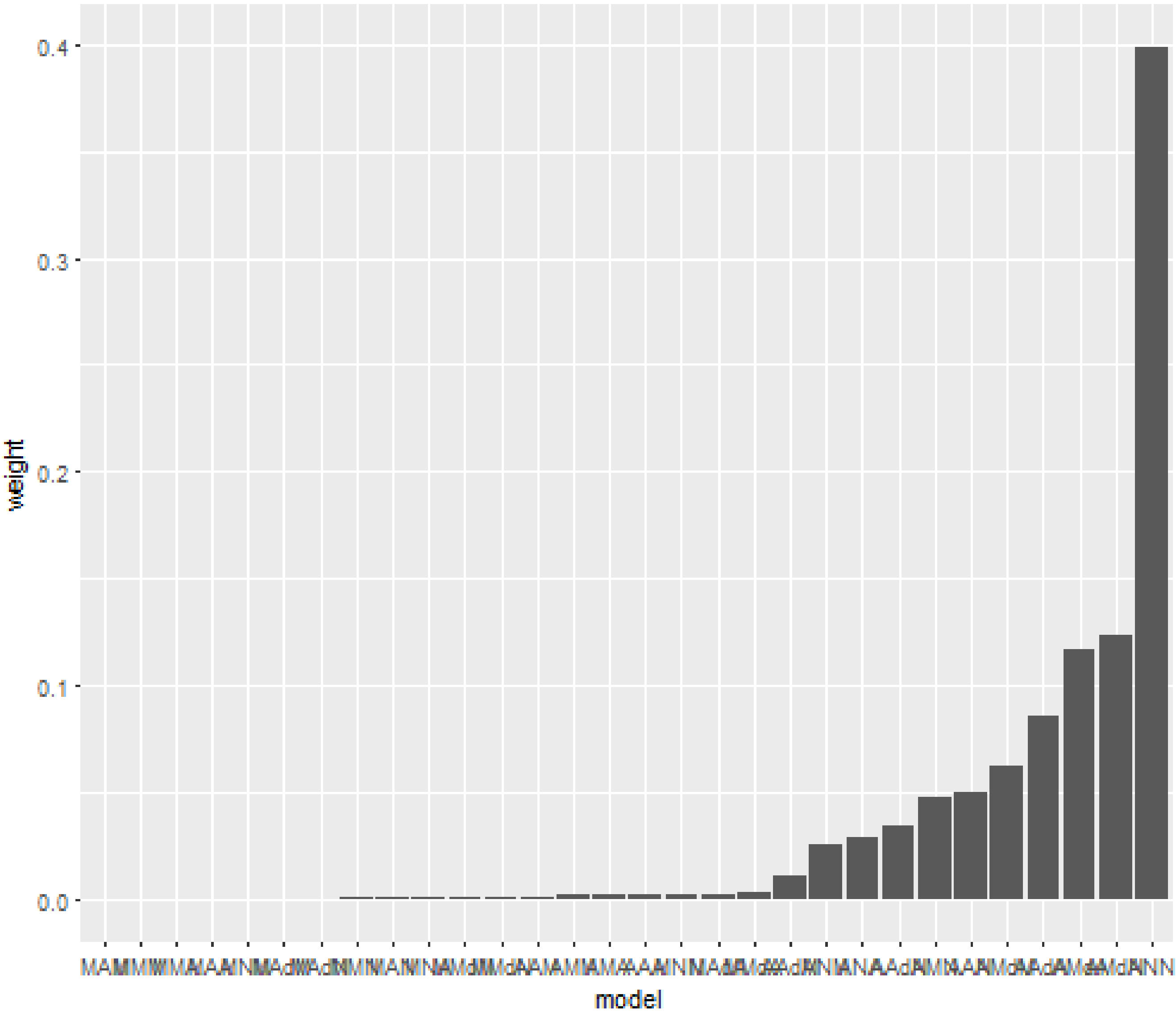
```
## MASE: 0.854; sMAE: 8.7%; RelMAE: 1.186; sMSE: 1.1%
```

Following code produces TS and ES for CCC Model, Also es builds us the 30 known models and suggests the best model to be ANN (Additive, None, None)

```
es_obj_CCC <- carsbymonth %>% filter(Year<=2016) %>%
  select(Quantity) %>% unlist() %>%
  ts(start=2007, f=12) %>%
  es(model="CCC", h=12, holdout=T, intervals=T, silent="output")
```



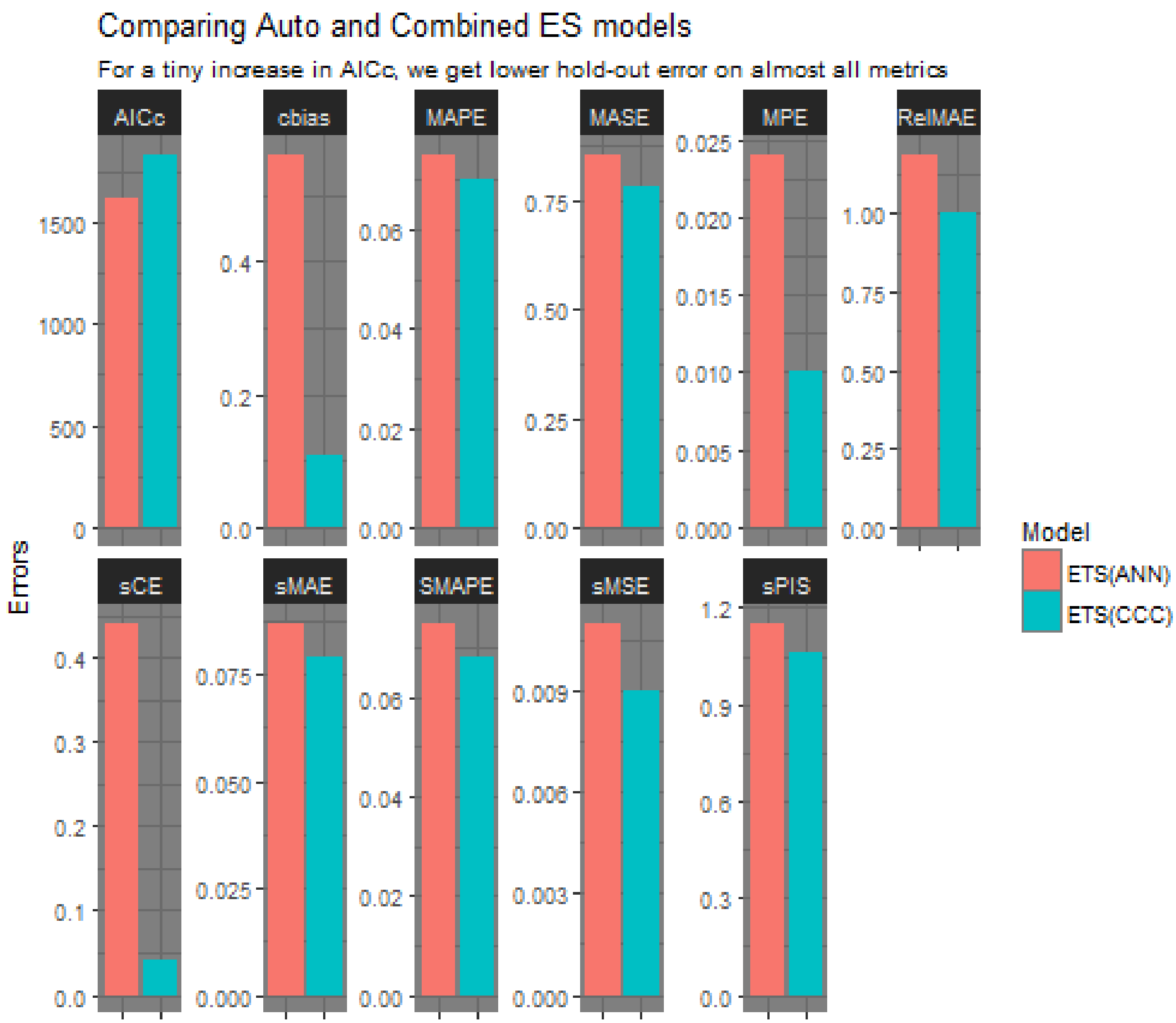

```
enframe(es_obj_CCC$ICw) %>% ggplot()+
  geom_col(mapping=aes(x=fct_reorder(name, value), y=value))+
  labs(x="model", y="weight")
```




```
summary(es_obj_CCC)
## Time elapsed: 8.88 seconds
## Model estimated: ETS(CCC)
## Initial values were optimised.
## Residuals standard deviation: 1003.013
## Cost function type: MSE
##
## Information criteria:
## Combined AICc
##      1826.874
## 95% parametric prediction intervals were constructed
## 100% of values are in the prediction interval
## Forecast errors:
## MPE: -1%; Bias: 10.8%; MAPE: 7%; SMAPE: 6.8%
## MASE: 0.786; sMAE: 7.9%; RelMAE: 1.006; sMSE: 0.9%
```

Comparisons of Errors for Models ZZZ and CCC using their AIC Values:

```
res <- rbind(c(es_obj_ZZZ$model, es_obj_ZZZ$accuracy, es_obj_ZZZ$ICs["AICc"]),
             c(es_obj_CCC$model,  es_obj_CCC$accuracy,  es_obj_CCC$ICs))  %>%
as_tibble()
names(res)[1] <- "model"
res %>% gather(key="metric", value="value", -model) %>%
  mutate(value=abs(as.numeric(value))) %>%
  ggplot()+geom_col(mapping=aes(x=model, y=value, fill=model))+
  facet_wrap(~metric, scales = "free_y", nrow=2)+
  theme_dark()+
  theme(axis.text.x = element_blank())+
  labs(x=NULL, y="Errors", fill="Model",
       title="Comparing Auto and Combined ES models",
       subtitle="For a tiny increase in AICc, we get lower hold-out error on
almost all metrics")
```

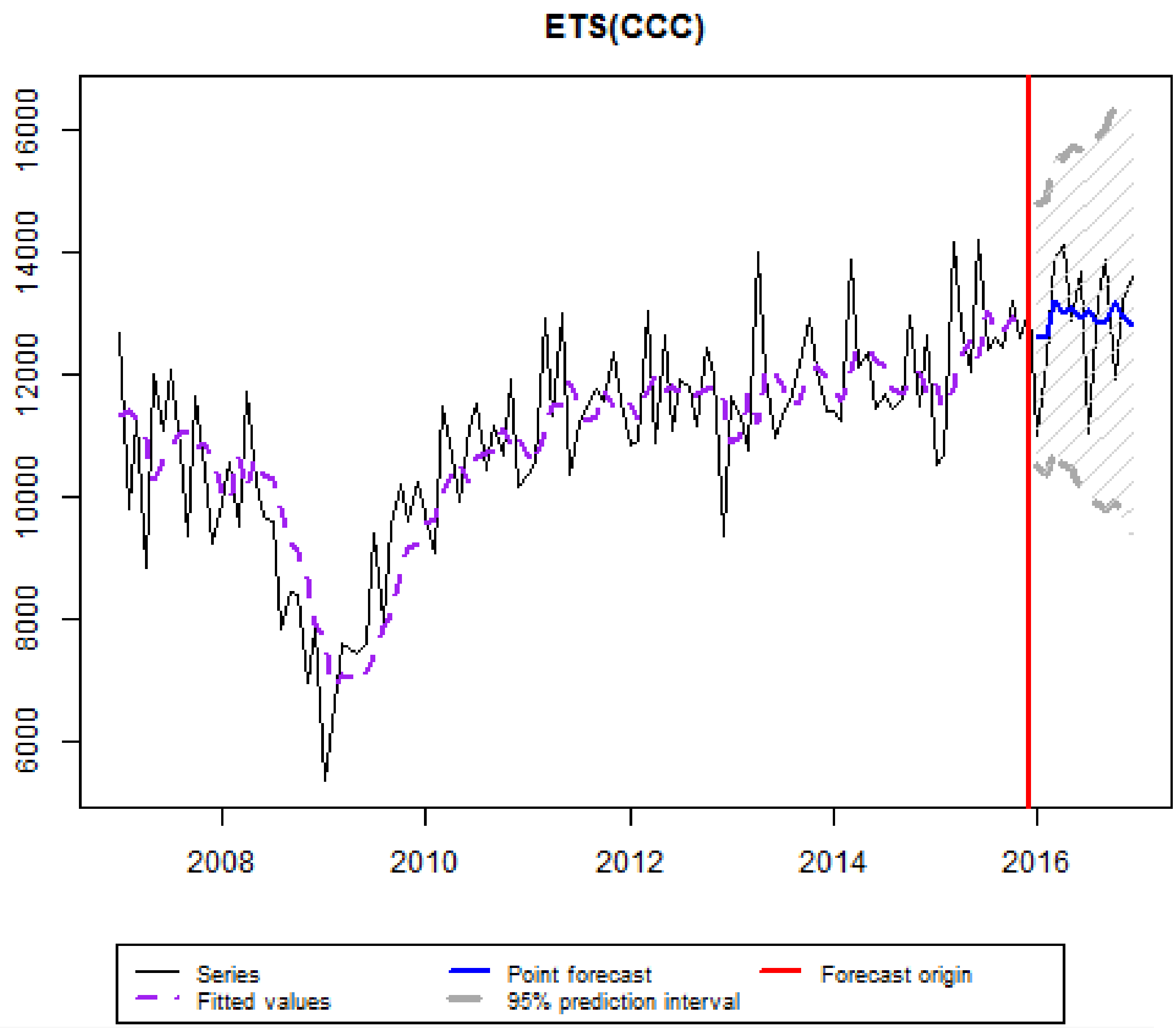


Building es object for Quantity:

```
es_obj <- carsbymonth %>% filter(Year<=2016) %>%
```

```
  select(Quantity) %>% unlist() %>%
  ts(start=2007, f=12) %>%
  es(model="CCC", h=12, holdout=T, intervals=T)#, silent="output")
```

```
## Estimation progress:
3%7%10%13%17%20%23%27%30%33%37%40%43%47%50%53%57%60%63%67%70%73%77%80%83%87%9
0%93%97%100%... Done!
```



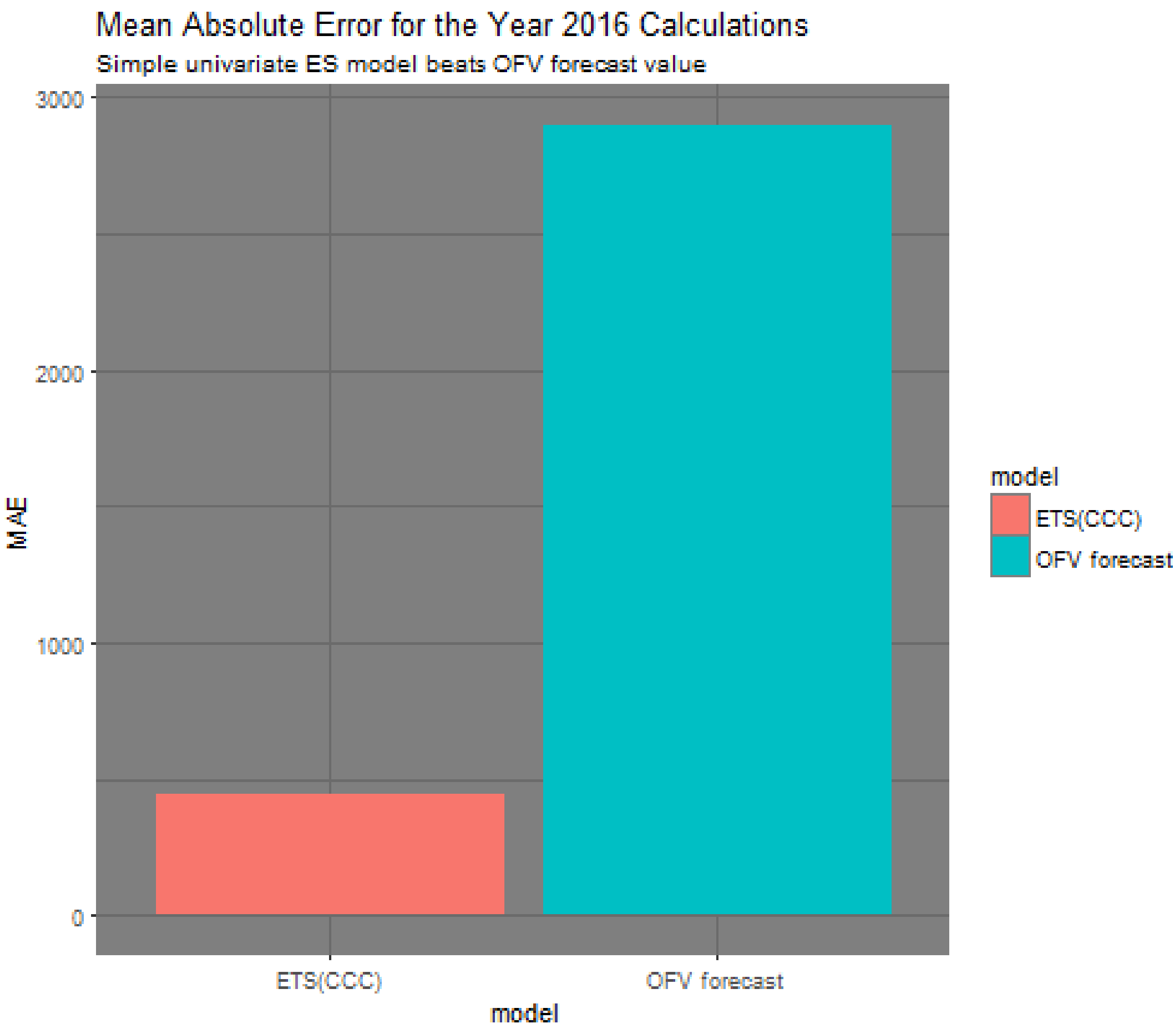

```
summary(es_obj)
```

```
## Time elapsed: 8.38 seconds
## Model estimated: ETS(CCC)
## Initial values were optimised.
## Residuals standard deviation: 1003.013
## Cost function type: MSE
##
## Information criteria:
## Combined AICc
##      1826.874
## 95% parametric prediction intervals were constructed
## 100% of values are in the prediction interval
```

```
## Forecast errors:
## MPE: -1%; Bias: 10.8%; MAPE: 7%; SMAPE: 6.8%
## MASE: 0.786; sMAE: 7.9%; RelMAE: 1.006; sMSE: 0.9%
```

Comparision for Mean Absolute Error for Year 2016 after forecast

```
rbind(
  tibble(model="OFV forecast", MAE=157500-sum(es_obj$holdout)),
  tibble(model=es_obj$model,    MAE=sum(es_obj$forecast)-sum(es_obj$holdout)))
%>%
  mutate(MAE=abs(MAE)) %>%
  ggplot()+
  geom_col(mapping=aes(x=model, y=MAE, fill=model))+
  theme_dark()+
  labs(title="Mean Absolute Error for the Year 2016 Calculations",
       subtitle="Simple univariate ES model beats OFV forecast value")
```



Building ES Object for the year 2017 with CCC model

```
es_obj_2017 <- carsbymonth %>% filter(Year<=2016) %>%
  select(Quantity) %>% unlist() %>%
  ts(start=2007, f=12) %>%
  es(model="CCC", h=12, holdout=F, intervals=T, silent=T)

summary(es_obj_2017)
```

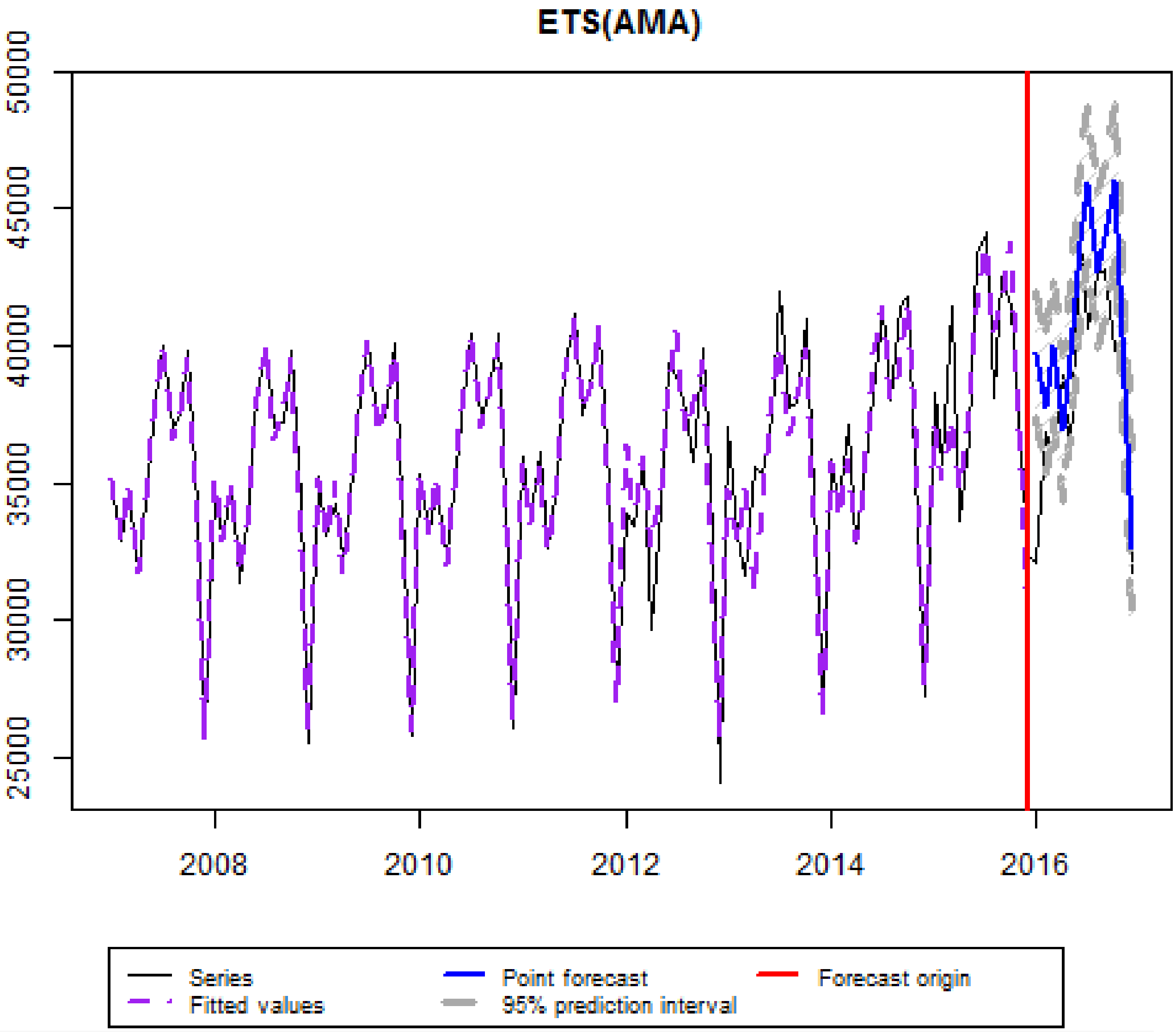
```
## Time elapsed: 8.66 seconds
## Model estimated: ETS(CCC)
## Initial values were optimised.
## Residuals standard deviation: 1012.227
## Cost function type: MSE
##
## Information criteria:
## Combined AICc
##      2031.068
## 95% parametric prediction intervals were constructed
```

Building ES Object for February and Used Cars.

```
es_obj_feb <- carsbymonth %>% select(Quantity) %>% unlist() %>%
  ts(start=2007, f=12) %>%
  es(model="CCC", h=1, holdout=F, intervals="np", silent=T)
summary(es_obj_feb)
```

```
## Time elapsed: 5.9 seconds
## Model estimated: ETS(CCC)
## Initial values were optimised.
## Residuals standard deviation: 1008.588
## Cost function type: MSE
##
## Information criteria:
## Combined AICc
##      2046.898
## 95% nonparametric prediction intervals were constructed
```

```
es_obj_Used <-carsbymonth %>% filter(Year>=2007, Year<=2016) %>%
  select(Used) %>% unlist() %>%
  ts(start=2007, f=12) %>%
  es(model="ZZZ", h=12, holdout=T, intervals="np", silent="output")
```

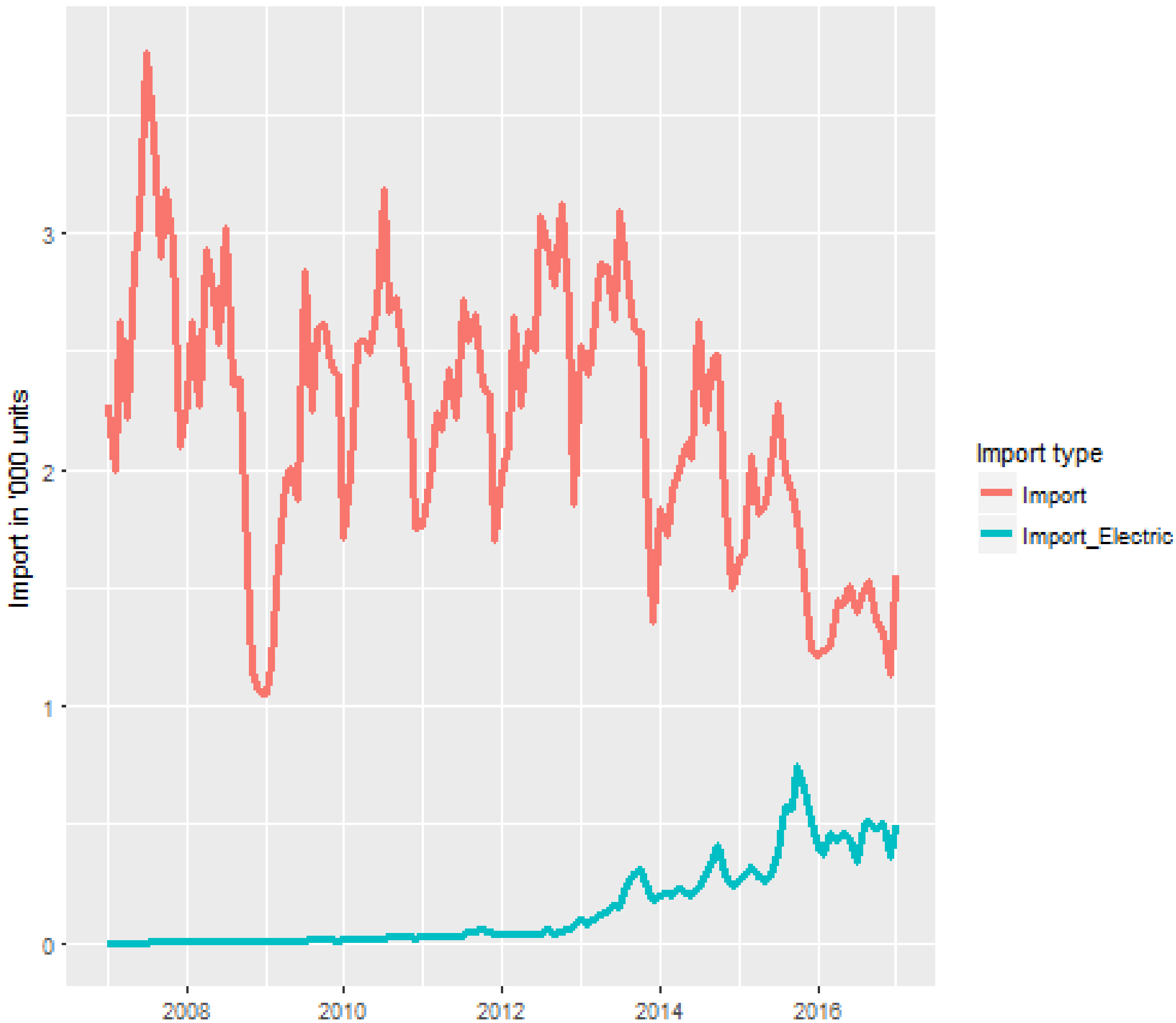



```
summary(es_obj_Used)
## Time elapsed: 2.18 seconds
## Model estimated: ETS(AMA)
## Persistence vector g:
## alpha  beta gamma
## 0.111 0.020 0.000
## Initial values were optimised.
## 18 parameters were estimated in the process
## Residuals standard deviation: 1227.933
## Cost function type: MSE; Cost function value: 1256517
##
## Information criteria:
##      AIC     AICc      BIC
## 1859.227 1866.912 1907.505
## 95% nonparametric prediction intervals were constructed
## 58% of values are in the prediction interval
## Forecast errors:
## MPE: -6.4%; Bias: -85.1%; MAPE: 7.3%; SMAPE: 6.8%
```

```
## MASE: 0.683; sMAE: 7.5%; RelMAE: 0.43; sMSE: 1%
```

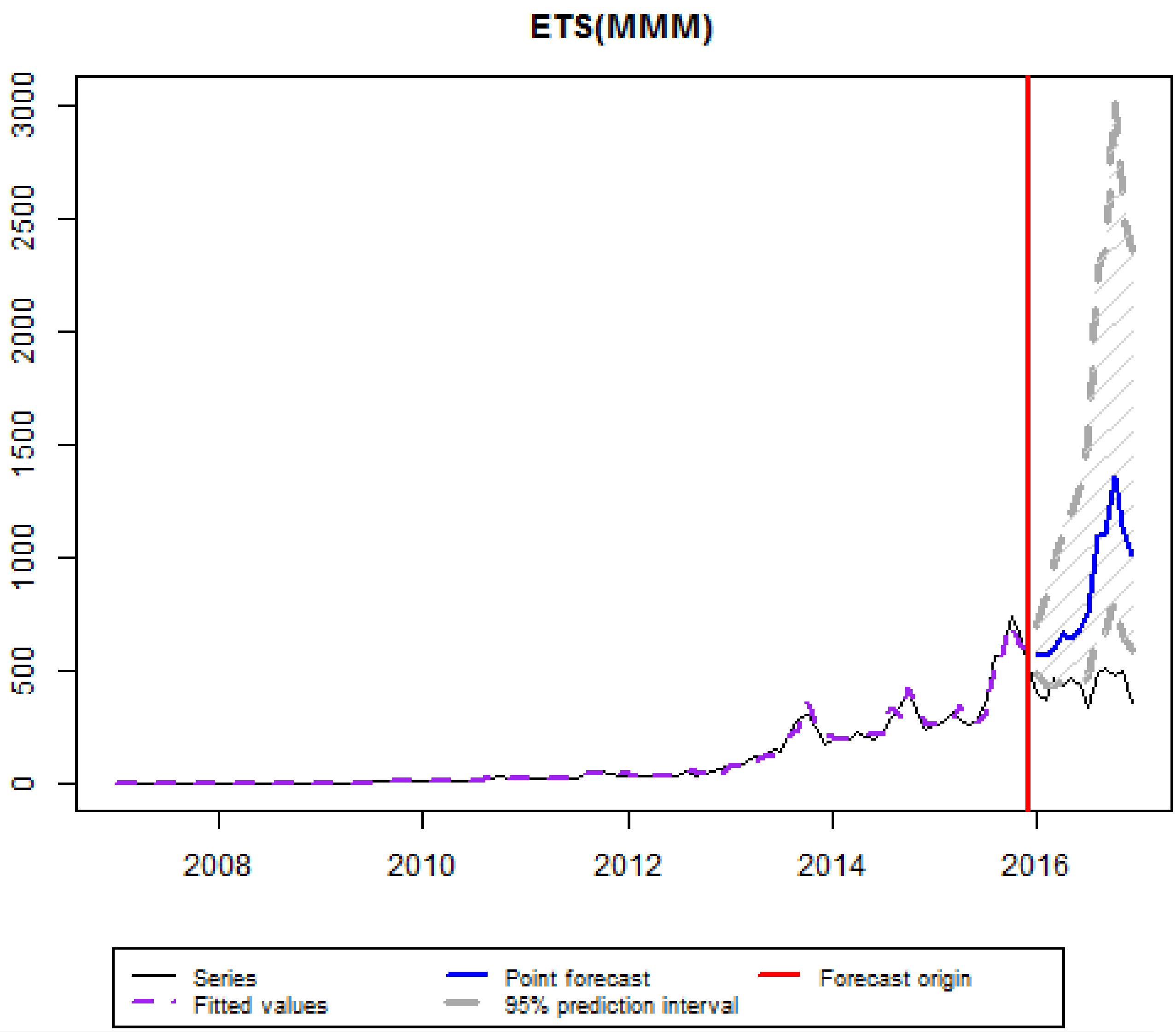
Plotting **Imports and Import of Electric Vehicles** (With Populated Values)

```
carsbymonth %>% mutate(Date=as.Date(paste(Year, Month, "1", sep="-"))) %>%
  select(Date, Import, Import_Electric) %>%
  gather(key=Import_type, value=Quantity, -Date) %>%
  ggplot()+
  geom_line(mapping=aes(x=Date,      y=Quantity/10^3,      color=Import_type),
size=1.1)+
  theme_grey()+
  labs(x=NULL, y="Import in '000 units",
       color="Import type")
```



Building ES Object for Imported vehicles

```
es_obj_Import <-carsbymonth %>% filter(Year<=2016) %>% select(Import_Electric)
%>% unlist() %>%
  ts(start=2007, f=12) %>%
  es(model="ZZZ", h=12, holdout=T, intervals="np", silent="output")
```



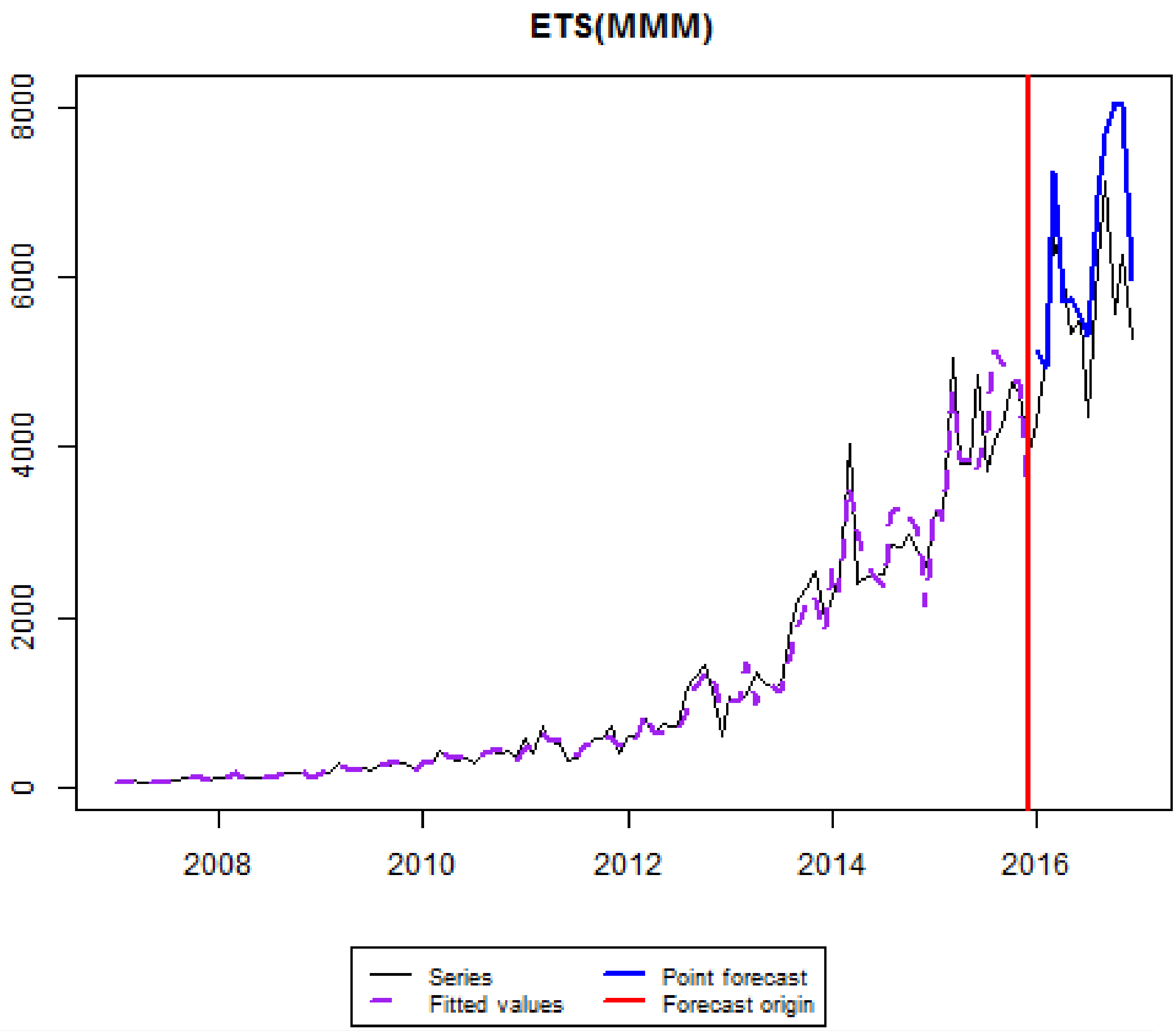

```
summary(es_obj_Import)
## Time elapsed: 2.32 seconds
## Model estimated: ETS(MMM)
## Persistence vector g:
## alpha  beta gamma
## 0.941 0.006 0.000
## Initial values were optimised.
## 18 parameters were estimated in the process
## Residuals standard deviation: 0.128
## Cost function type: MSE; Cost function value: 14
##
## Information criteria:
##      AIC     AICc      BIC
## 627.5573 635.2427 675.8357
```

```
## 95% nonparametric prediction intervals were constructed
## 25% of values are in the prediction interval
## Forecast errors:
## MPE: -92.9%; Bias: -100%; MAPE: 92.9%; SMAPE: 59.3%
## MASE: 22.087; sMAE: 379.9%; RelMAE: 5.574; sMSE: 1927%
```

Building ES model for Environmentally Friendly Vehicles

```
es_obj_EFV <- carsbymonth %>%  filter(Year>=2007, Year<=2016) %>%
  transmute(EFV=Import_Electric+Quantity_Hybrid+Quantity_Electric) %>%
  select(EFV) %>%  unlist() %>%
  ts(start=2007, f=12) %>%
  es(model="ZZZ", h=12, holdout=T)
```

```
## Forming the pool of models based on... ANN, ANA, ANM, AAM, Estimation
progress:    45%55%64%73%82%91%100%... Done!
```



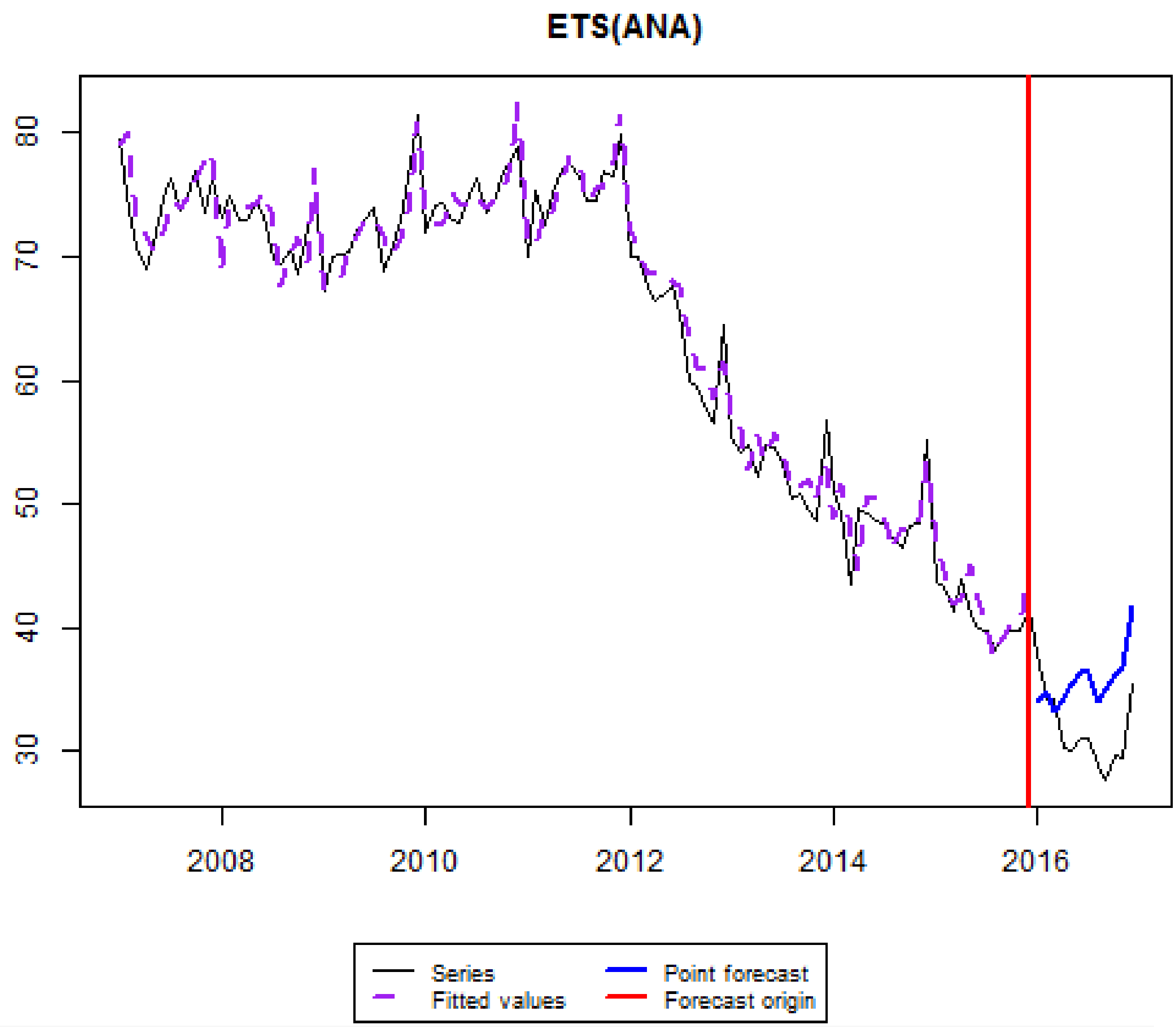

```
summary(es_obj_EFV)
```

```
## Time elapsed: 2.34 seconds
```

```
## Model estimated: ETS(MMM)
## Persistence vector g:
## alpha  beta gamma
## 0.591 0.001 0.001
## Initial values were optimised.
## 18 parameters were estimated in the process
## Residuals standard deviation: 0.131
## Cost function type: MSE; Cost function value: 4897
##
## Information criteria:
##      AIC     AICc      BIC
## 1260.104 1267.789 1308.382
## Forecast errors:
## MPE: -13.3%; Bias: -87.2%; MAPE: 14.5%; SMAPE: 13%
## MASE: 4.082; sMAE: 67.6%; RelMAE: 0.473; sMSE: 77.3%
```

Building ES Object for Diesel Vehicles

```
es_obj_Diesel <- carsbymonth %>%  filter(Year<=2016) %>%
  select(Diesel_Share) %>%  unlist() %>%
  ts(start=2007, f=12) %>%
  es(model="ZZZ", h=12, holdout=T, silent="output")
```




```
summary(es_obj_Diesel)
## Time elapsed: 0.78 seconds
## Model estimated: ETS(ANA)
## Persistence vector g:
## alpha gamma
## 0.945 0.000
## Initial values were optimised.
## 16 parameters were estimated in the process
## Residuals standard deviation: 2.235
## Cost function type: MSE; Cost function value: 4
##
## Information criteria:
##      AIC     AICc      BIC
## 494.8759 500.8539 537.7900
## Forecast errors:
## MPE: -13.5%; Bias: -83.4%; MAPE: 15.7%; SMAPE: 14.4%
## MASE: 1.973; sMAE: 7.5%; RelMAE: 0.492; sMSE: 0.7%
```

Using Models Built above to Predict Quantity **It Works !** Better model compared to the one on Kaggle

```
xregs_hist <- carsbymonth %>%
  select(Import_Electric,   Quantity_Hybrid,   Quantity_Electric,   Used)   %>%
rowSums(na.rm=T) %>%
  as_tibble()        %>%        bind_cols(carsbymonth,        .)        %>%
setNames(.,c(names(carsbymonth), "Greens")) %>%
  filter(Year>=2007, Year<=2015) %>% select(Used, Import, Diesel_Share,Greens)

xreg_pred <- data.frame(Used=as.vector(es_obj_Used$forecast),
                        Import=as.vector(es_obj_Import$forecast),
                        Greens=as.vector(es_obj_EFV$forecast),
                        Diesel_Share=as.vector(es_obj_Diesel$forecast),
                        stringsAsFactors = F)

xregs <- rbind(xregs_hist, xreg_pred)

es_obj_X2016 <- carsbymonth %>% filter(Year>=2007, Year<=2017) %>%
  select(Quantity) %>% unlist() %>%
  ts(start=2007, f=12) %>%
  es(model="CCC", xreg=xregs, h=12, holdout=T, intervals=T)
```

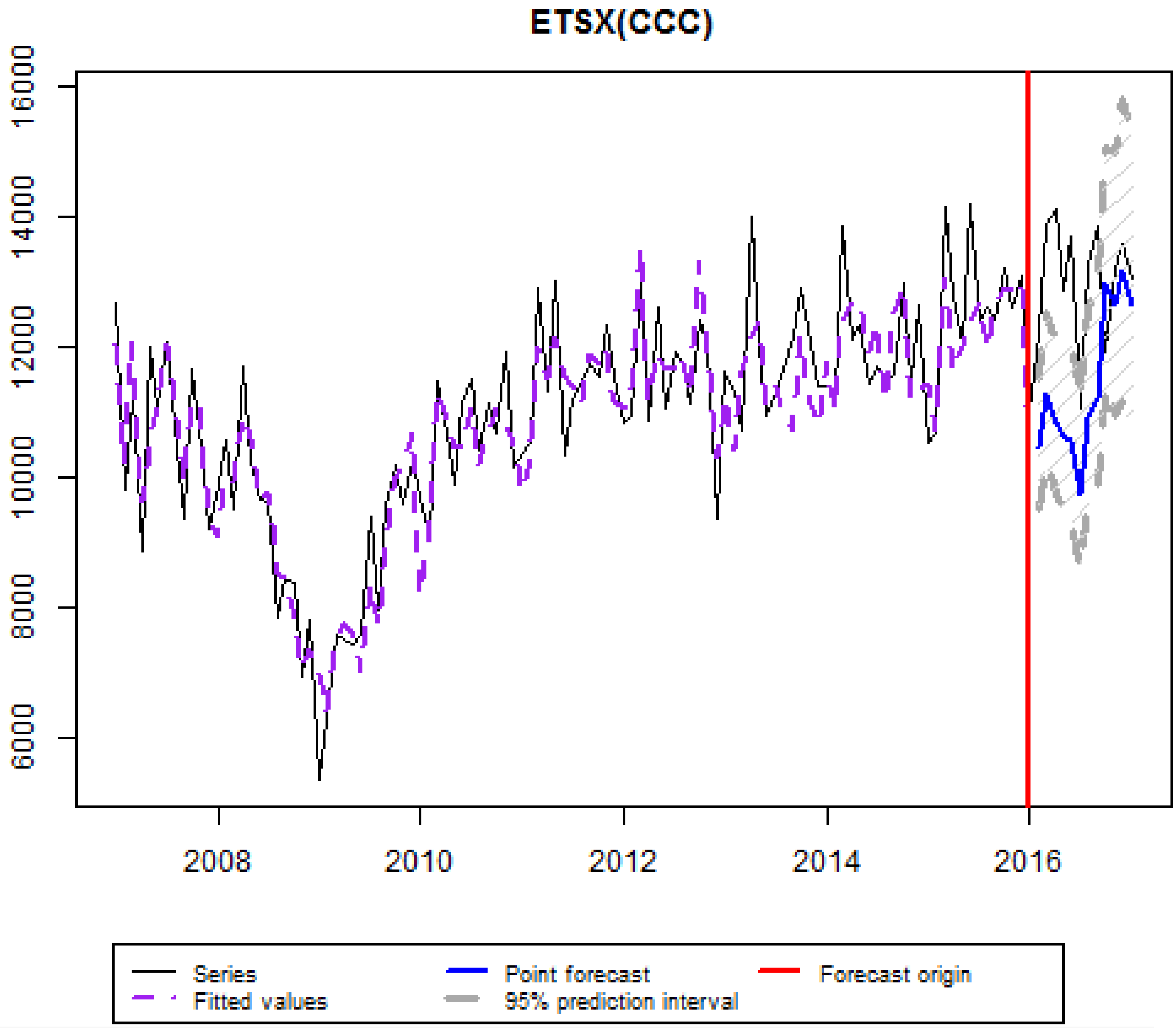
```
## Warning: xreg did not contain values for the holdout, so we had to predict
## missing values.
```

```
## Producing forecasts for xreg variable...
```

```
## 25%50%75%100%Done!
##                              Estimation                              progress:
3%7%10%13%17%20%23%27%30%33%37%40%43%47%50%53%57%60%63%67%70%73%77%80%83%87%9
0%93%97%100%... Done!
```

```
## Warning: Something went wrong during the optimisation and NAs were
## produced!
```

```
## Warning: Please check the input and report this error to the maintainer if
## it persists.
```




```
summary(es_obj_X2016)
## Time elapsed: 11.94 seconds
## Model estimated: ETSX(CCC)
## Initial values were optimised.
## Residuals standard deviation: 670.727
## Xreg coefficients were estimated in a normal style
## Cost function type: MSE
##
## Information criteria:
## Combined AICc
##      1793.125
## 95% parametric prediction intervals were constructed
## 42% of values are in the prediction interval
## Forecast errors:
## MPE: 12.2%; Bias: 91%; MAPE: 13.7%; SMAPE: 14.9%
## MASE: 1.625; sMAE: 16.6%; RelMAE: 0.876; sMSE: 3.6%
```

Import Car Prediction Model

```
es_obj_Y2016 <- carsbymonth %>% filter(Year>=2007, Year<=2017) %>%
  select(Import_Electric) %>% unlist() %>%
  ts(start=2007, f=12) %>%
  es(model="CCC", xreg=xregs, h=12, holdout=T, intervals=T)
```

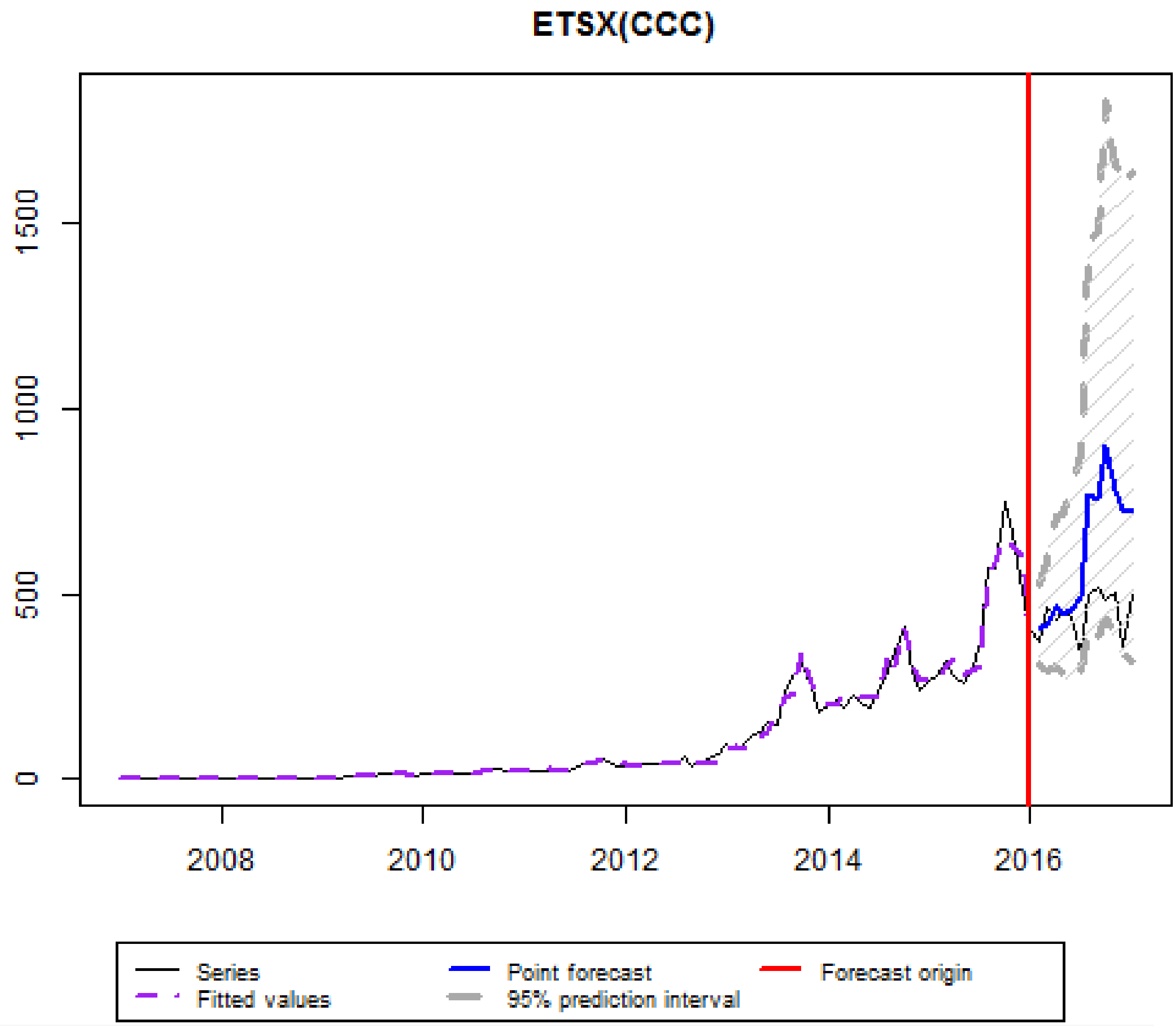
```
## Warning: xreg did not contain values for the holdout, so we had to predict
## missing values.
```

```
## Producing forecasts for xreg variable...
```

```
## 25%50%75%100%Done!
##                              Estimation                              progress:
3%7%10%13%17%20%23%27%30%33%37%40%43%47%50%53%57%60%63%67%70%73%77%80%83%87%9
0%93%97%100%... Done!
```

```
## Warning: Negative values produced in state vector of model MAA.
## Please, use a different model.
```

```
## Warning: Negative values produced in state vector of model MAdA.
## Please, use a different model.
```




```
summary(es_obj_Y2016)
```

```
## Time elapsed: 11.51 seconds
## Model estimated: ETSX(CCC)
```

```
## Initial values were optimised.
## Residuals standard deviation: 20.353
## Xreg coefficients were estimated in a normal style
## Cost function type: MSE
##
## Information criteria:
## Combined AICc
##      657.1759
## 95% parametric prediction intervals were constructed
## 100% of values are in the prediction interval
## Forecast errors:
## MPE: -36.6%; Bias: -88.6%; MAPE: 39.2%; SMAPE: 30.1%
## MASE: 9.044; sMAE: 158.6%; RelMAE: 2.636; sMSE: 403.7%
```

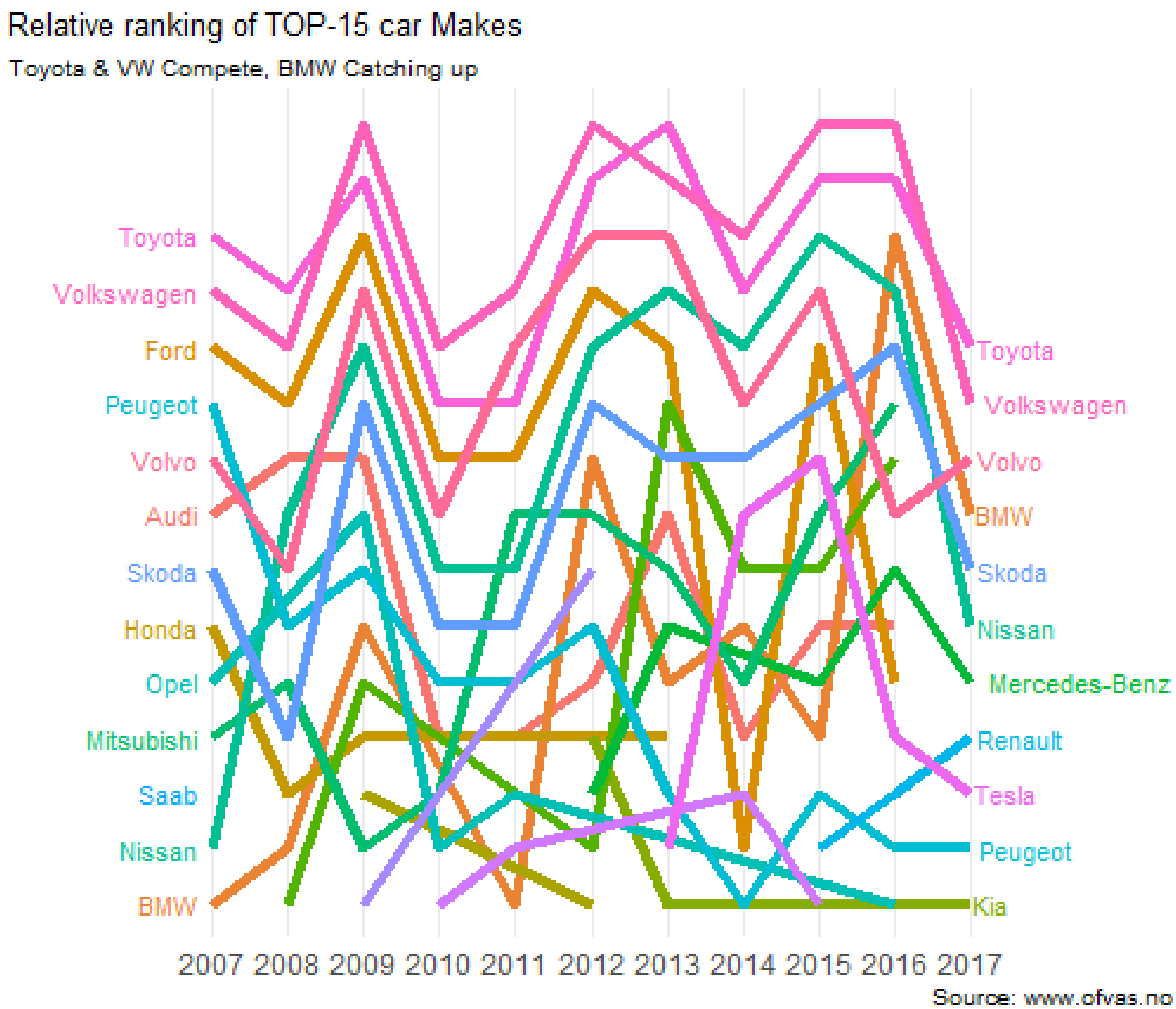
Comparision of Data from Two Sheets for Makes and Models Popularity in Norway

```
by_make <- read.csv("norway_new_car_sales_by_make.csv",header=TRUE)
by_model <- read.csv("norway_new_car_sales_by_model.csv",header=TRUE)

by_make %>% group_by(Year, Make) %>% summarise(sum_Quantity=sum(Quantity)) %>%
arrange(Year, desc(sum_Quantity)) %>%
  top_n(15, sum_Quantity) %>% mutate(Relative_rank=rank(sum_Quantity)) %>%
  ggplot(mapping=aes(x=Year, y=Relative_rank,
                     group = Make, colour = Make, label = Make)) +
  geom_line(size=1.5) +
  geom_text(data = . %>% filter(Year== 2017), mapping=aes(x = Year, hjust = -
0.1))+
  geom_text(data = . %>% filter(Year== 2007), mapping=aes(x = Year-0.1, hjust
= "right"))+
  labs(title="Relative ranking of TOP-15 car brands",
       subtitle="Market traditionally owned by VW and Toyota has gotten a
challenger",
       caption="Source: www.ofvas.no")+
  scale_y_discrete(breaks=NULL) + expand_limits(x=2005:2019) +
scale_x_continuous(breaks=2007:2017)+
  theme_bw() + theme(legend.position = "none", panel.border = element_blank(),
                     panel.grid.minor.x = element_blank(), axis.ticks =
element_blank(),
                     axis.text=element_text(size=12, family = "sans")
  ) + xlab(NULL) + ylab(NULL)
```


```
by_model %>% group_by(Year, Make) %>% summarise(sum_Quantity=sum(Quantity)) %>%
arrange(Year, desc(sum_Quantity)) %>%
  top_n(15, sum_Quantity) %>% mutate(Relative_rank=rank(sum_Quantity)) %>%
  ggplot(mapping=aes(x=Year, y=Relative_rank,
                     group = Make, colour = Make, label = Make)) +
  geom_line(size=1.5) +
  geom_text(data = . %>% filter(Year== 2017), mapping=aes(x = Year, hjust = -
0.1))+
  geom_text(data = . %>% filter(Year== 2007), mapping=aes(x = Year-0.1, hjust
= "right"))+
  labs(title="Relative ranking of TOP-15 car Makes",
       subtitle="Toyota & VW Compete, BMW Catching up",
       caption="Source: www.ofvas.no")+
  scale_y_discrete(breaks=NULL)      +      expand_limits(x=2005:2019)      +
scale_x_continuous(breaks=2007:2017)+
  theme_bw() + theme(legend.position = "none", panel.border = element_blank(),
                     panel.grid.minor.x  =  element_blank(),  axis.ticks  =
element_blank(),
```

```
                    axis.text=element_text(size=12, family = "sans")
  ) + xlab(NULL) + ylab(NULL)
```



Top - 15 car Models

```
by_model %>% group_by(Year, Model) %>% summarise(sum_Quantity=sum(Quantity))
%>% arrange(Year, desc(sum_Quantity)) %>%
  top_n(15, sum_Quantity) %>% mutate(Relative_rank=rank(sum_Quantity)) %>%
  ggplot(mapping=aes(x=Year, y=Relative_rank,
                    group = Model, colour = Model, label = Model)) +
  geom_line(size=1.5) +
  geom_text(data = . %>% filter(Year== 2017), mapping=aes(x = Year, hjust = -
0.1))+
  geom_text(data = . %>% filter(Year== 2007), mapping=aes(x = Year-0.1, hjust
= "right"))+
  labs(title="Relative ranking of TOP-15 car models",
       subtitle="Analysis of Car Market ownership",
       caption="Source: www.ofvas.no")+
```

```
  scale_y_discrete(breaks=NULL)      +      expand_limits(x=2005:2019)      +
scale_x_continuous(breaks=2007:2017)+
  theme_bw() + theme(legend.position = "none", panel.border = element_blank(),
                     panel.grid.minor.x   =   element_blank(),   axis.ticks   =
element_blank(),
                     axis.text=element_text(size=12, family = "sans")
  ) + xlab(NULL) + ylab(NULL)
```