Second Year Project II

Frank Toth (i6225720)
Ramnarain Nair (i6221499)

24^{th} June 2021

Introduction

The case involves analysing shipment data from a shipment company, which allocates shipments to certain clusters. Our main aim is analyse the current clustering by conducting some data analysis using Econometrics and also identifying an optimal clustering allocation of locations using techniques from Operations Research.

In the Econometrics section, we first had to clean the data using excel by adding any missing information and aggregating the data on a daily basis for the volume, weight and number of shipments for our particular lane. This allowed us create a basis to start performing regressions on out series of interest. From theses regressions, we conducted tests to determine if we SARMA models. In addition to this, we conducted hypothesis tests to identify if we have significant daily effects. Finally, we forecasted our data to see how well our model performs.

In the Operations Research section, we first had to measure the current clustering provided in the shipment data by computing certain performance measures on distance, volume, weight and number of shipments for the whole data set. Then, we tried to improve the current clustering by attempting implement the well-known k-means clustering algorithm. Lastly, we suggested an alternative method of clustering using the weights of the shipments.

Question 1: Data (pre) processing

In this section of the report, the outcomes of data (pre) processing on the given shipment data is presented, which was essential to building the required time series. Firstly, using built-in functions in Excel, the 7 most frequent origin clusters were computed. This led to the results found in **Figure 1**.

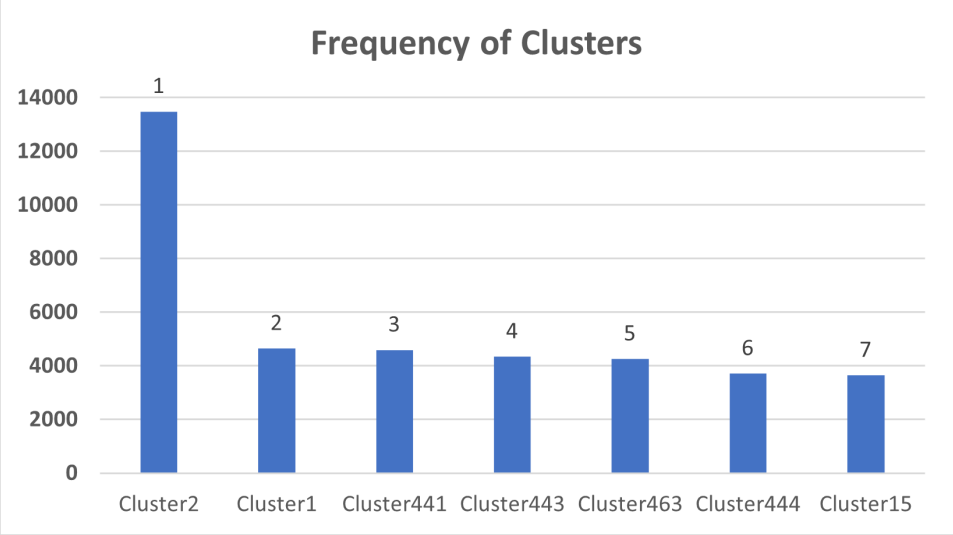


Figure 1: Ranking of 7 most frequent origin clusters

The origin cluster investigated was origin cluster 1. Through appropriate filtering in Excel, the most frequent destination for origin cluster 1 was destination cluster 15, with a frequency of 419. Hence, the lane found links cluster 1 with destination cluster 15. To avoid those trivial lanes, consisting of shipping goods to the other side of the street,

the data for the lane was filtered such that the distance between the origin and destination location was between 20 kilometers and 400 kilometers.

For the particular lane, the time series of interest were Gross Weight (kg), Nb of Ship Units, and Gross Volume (m3). To be able to build these time series, the shipment data for the lane had to be aggregated to only have one observation per day. The process involved identifying shipments with the same pick-up date and adding the associated values for Gross Weight (kg), Nb of Ship Units, and Gross Volume (m3) to obtain the aggregated daily values for each series from 2/01/2017 until 12/10/2017.

The lane being considered between origin cluster 1 and destination cluster 15 is the lane between two cities in Germany, namely, Hamburg and Bremen. In the process of data aggregation for the lane, shipments for the following dates were missing: 1/05/2017, 25/05/2017, 5/06/2017 and, 3/10/2017. The dates 1/05/2017 and 3/10/2017 were Labour Day and German Unity Day respectively. As those days were off, an empty zero line was added for those dates in the aggregated data. For the remaining two dates, it was assumed that no delivery was made on those dates, and therefore, zero's were included for Gross Weight (kg), Nb of Ship Units, and Gross Volume (m3). These were the considerations made before importing the aggregated data into EViews with a dated frequency of 5 days a week.

The plots of the series for Gross Weight (kg), Nb of Ship Units, and Gross Volume (m3) can be seen in **Figure 2**, **Figure 3**, and **Figure 4** respectively.

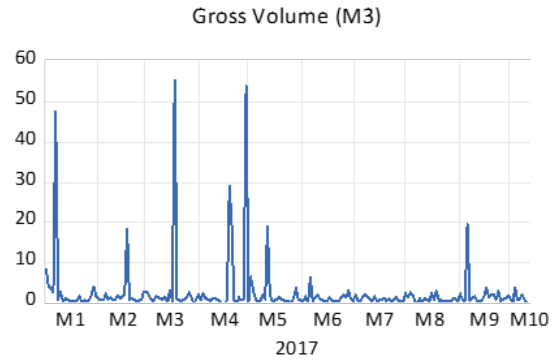


Figure 2: Plot of series Gross Volume (m^3)

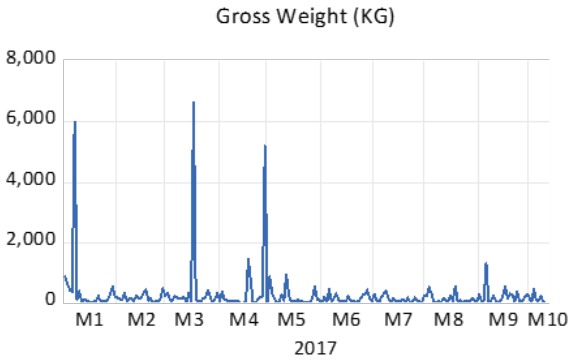


Figure 3: Plot of series Gross Volume (kg)

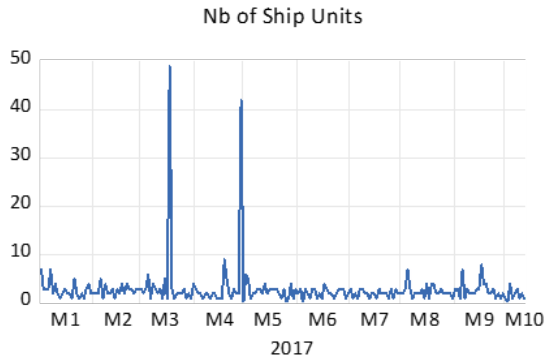


Figure 4: Plot of series Nb of Shipments

The extreme outliers for each series can be seen in **Table 1** below, which is useful for the next parts of the case.

	Series		
	Gross Volume	Gross Weight	Nb of Ship Units
Outliers	06/01/2017	06/01/2017	17/03/2017
	17/02/2017	17/03/2017	28/04/2017
	17/03/2017	28/04/2017	
	19/04/2017		
	28/04/2017		
	11/05/2017		
	06/09/2017		

Table 1: Represents extreme outliers for each series

Question 2: Daily Seasonality

1 Modelling

In this section of the report, daily seasonality will be discussed. It will be investigated whether seasonality patterns can be observed in the shipment data by looking at the time series. Firstly, the discussion will be about whether taking the levels of the variables is enough or we have to take log levels. Secondly, linear regressions will be made to check whether there is daily seasonality and check whether those variables are significant. Thirdly, we will be looking at the correlogram graphs from the LM test to identify if an ARMA(p,q)(P, Q) model exists. Lastly, various misspecification tests will be conducted on the models with their final specifications.

In **Figure 2**, **Figure 3**, and **Figure 4** the different series of Gross Volume (m^3), Gross Weight (kg), and the Number of Shipments can be observed. It can be observed that when removing outliers from the series, the data looks stationary. The mean seems to be around the same regardless of where we are in the series for each of the three series. So we conclude that by just looking at the graphs, that the series is stationary which means that it is not necessary to change the levels into log levels.

In the next part of the paper, linear regressions will be performed to determine whether there is a significant daily effect. The dependant variable for each of the three regressions will be Gross Volume (m^3), Gross Weight (kg), and

the Number of Shipments, in that order. The independent variables are going to be the "daily" dummy variables as well as the associated outliers for each series. The dummies for the outliers can be seen in **Table 1**.To construct such dummy variables, we create a new series with values for the dates equal to 0 everywhere except for the date of the outlier. A series is created for each dummy variable. The linear regressions will look as follows, where V, W and N correspond to Gross Volume, Gross Weight and Nb of Shipments at time t respectively:

$$V_t, W_t, N_t = \alpha_0 + \alpha_1 Tuesd + \alpha_2 Wed + \alpha_3 Thur + \alpha_4 Frid + dummies + \epsilon_t$$

The Eviews output for these regressions can be seen in **Figure A1**, **Figure A2** and **Figure A3** in the appendix.

In **Figure A1** the regression for Gross Volume (m^3) can be observed. The intercept is equal to 1.42 cubic meters. The interpretation of the coefficient of the intercept is the following: it is equal to the mean volume of shipments for Monday. As Monday is the reference group for the dummies, its value is equal to the intercept. It turns out that the mean volume of the shipments for Tuesday is approximately 0.04 more compared to Monday. For Wednesday it is approximately 0.122 less than the volume of shipments on Monday. For Thursday it is approximately 0.07 more than Monday and lastly, for Friday it is approximately 0.11 less than on Monday. Note that all the dummy variables for the outlier values are significant. The adjusted R-squared of this linear regression is equal to 0.93 which means that our model seems to explain the dependent variable well with the inclusion of the dummy outliers. The Schwarz criterion is equal to 4.18.

In **Figure A2** the regression for Gross Weight (kg) can be observed. The interpretations of the coefficient for both the intercept and the variables are the same. Meaning that for Monday, the mean Gross Weight (kg) being shipped is equal to 183.69 kilograms. Furthermore, it can also be seen that on Tuesday and Thursday, the mean Gross Weight (kg) is smaller by 9.28Kg and 10.19Kg respectively. Meanwhile, on Wednesdays and Fridays, the Gross Weight (kg) is larger by 26.00Kg and 1.79Kg respectively. The adjusted R-squared is equal to 0.918 and the Schwarz criterion is equal to 13.68.

In **Figure A3** the regression for Nb of Shipments can be observed. The same conclusions can be drawn from the coefficients. The mean Number of Shipments made on Mondays is equal to 2.82. Furthermore, for Wednesdays, the Number of Shipments is larger by 0.12 on average. While on Tuesdays, Thursdays, and Fridays the average Number of Shipments are smaller (by 0.95, 1.12, and 0.01 respectively). The adjusted R-squared is equal to 0.913 and the Schwarz criterion is equal to 3.54.

In the following part of the paper, the residuals will be investigated in detail to check for the presence of auto-correlation. The correlograms can be found in the appendix.

For the residuals of the Gross Volume (m3), no autocorrelation is present from the Q-test. The ACF and PACF result in the same conclusion of no autocorrelation. This can be seen in **Figure A4** in the appendix. In **Figure 5**, the results of the LM-test can be found up to 2 lags. The conclusion is that again no autocorrelation can be observed up to 2 lags.

Breusch-Godfrey Serial Correlation LM Test:
Null hypothesis: No serial correlation at up to 2 lags

F-statistic	0.209110	Prob. F(2,190)	0.8115
Obs*R-squared	0.448050	Prob. Chi-Square(2)	0.7993

Figure 5: LM-test on residuals of Gross Volume

Looking at the ACF and PACF in **Figure A4** in the appendix, it looks like the process is white noise. Both the ACF and PACF are very close to 0 at order 1. At order 5, 10, 15, and so on, there is no significant increase in the

values in the correlograms hence, there is no evidence of any seasonality. The conclusion about the Gross Volume (m3) of the shipments is that we have an ARMA(0,0)(0,0) model.

Next, the same analysis is performed on the residuals of the Gross Weight (kg). In the appendix, the Q-test results with the correlograms can be found for Gross Weight in **Figure A5**. There is evidence of autocorrelation at up to 1 lag as the p-value is equal to 0.045 (which is significant at a significance level of 0.05). This means that there is still some information in the residuals that can explain the dependent variable. Looking at the ACF and PACF, it is quite difficult to determine the type of ARMA model that we have at order 1 because they both follow a very similar pattern. However, at orders 5, 10, 15, and so on, the ACF and PACF do not show much evidence for seasonality. In **Figure 6** and **Figure 7**, the LM test can be seen at up to 1 and 2 lags. We have significant evidence for autocorrelation at up to 1 lag but no autocorrelation for 2 lags; this further demonstrates that there are remaining components in the residuals that need to be further investigated at up to 1 lag. To do so, we re-ran the linear regression with ARMA variables to check whether we have an ARMA model.

Breusch-Godfrey Serial Correlation LM Test:
Null hypothesis: No serial correlation at up to 1 lag

F-statistic	3.891757	Prob. F(1,195)	0.0499
Obs*R-squared	3.991711	Prob. Chi-Square(1)	0.0457

Figure 6: LM-test on residuals of Gross Weight at up to 1 lag

Breusch-Godfrey Serial Correlation LM Test:
Null hypothesis: No serial correlation at up to 2 lags

F-statistic	2.268935	Prob. F(2,194)	0.1062
Obs*R-squared	4.662715	Prob. Chi-Square(2)	0.0972

Figure 7: LM-test on residuals of Gross Weight at up to 2 lags

In **Figure A6** in the appendix, the output of the new regression with ARMA variables can be found. It can be seen that indeed when running the linear regression with a MA(1), the variable seems to be significant (p-value of 0.007). Hence, we conclude that we have an ARMA(0,1)(0,0) model for the Gross Weight (kg). This is further confirmed since now when running the Q-test with this model, we no longer find the presence of autocorrelation, which can be found in **Figure A7** appendix. When running the regression with AR(1) instead of MA(1), AR(1) also turns out to be significant but, the adjusted-R2 is slightly smaller and the information criterion slightly larger. For those reasons, the ARMA(0,1)(0,0) model was preferred over the ARMA(1,0)(0,0). The regression with AR(1) can be found in **Figure A8** of the appendix.

Lastly, we are going to investigate whether there is autocorrelation in the Number of Shipments residuals data. Similar conclusions can be drawn from Gross Volume (m3) for the Number of Shipments. When performing the Q-test, no proof of autocorrelation can be found as can be seen in **Figure A9** of the appendix. Furthermore, the same conclusions can be drawn from the LM-test up to 2 lags, with no signs of autocorrelation (Refer to **Figure 8**).

Breusch-Godfrey Serial Correlation LM Test:
Null hypothesis: No serial correlation at up to 2 lags

F-statistic	0.339862	Prob. F(2,195)	0.7123
Obs*R-squared	0.708625	Prob. Chi-Square(2)	0.7017

Figure 8: LM-test on residuals of Nb of Shipments at up to 2 lags

Looking at the ACF and PACF in **Figure A9** in the appendix, it looks like the process is white noise. Both the ACF and PACF are very close to 0 at order 1. At order 5, 10, 15, and so on, there is no significant increase in the values in the correlograms hence, there is no evidence of any seasonality. Hence, the conclusion is that we have an ARMA(0,0)(0,0) model.

2 Hypothesis Testing

In this part of the paper, we are going to investigate the significance of the daily dummy variables. For the Gross Volume (m3) of the shipments, it can be seen from **Figure A1** in the appendix the individual p-values of all the daily dummies are not a significant at a 5% significance level. Therefore, we fail to reject the null that the are individually equal to 0. The joint Wald test in **Figure 9** indicates that the daily dummies are jointly not significantly different from 0. Hence, the daily dummies are individually and jointly not significantly different from Monday.

Wald Test:
Equation: Untitled

Test Statistic	Value	df	Probability
F-statistic	0.108191	(4, 192)	0.9796
Chi-square	0.432765	4	0.9797

Figure 9: Joint Wald test on daily dummies for Gross Volume

For the Gross Weight (kg) of the shipments, it can be seen from **Figure A6** in the appendix the individual p-values of all the daily dummies are not a significant at a 5% significance level. Therefore, we fail to reject the null that the are individually equal to 0. The joint Wald test in **Figure 10** indicates that the daily dummies are jointly not significantly different from 0. Hence, the daily dummies are individually and jointly not significantly different from Monday.

Wald Test:
Equation: Untitled

Test Statistic	Value	df	Probability
F-statistic	0.257265	(4, 194)	0.9050
Chi-square	1.029059	4	0.9054

Figure 10: Joint Wald test on daily dummies for Gross Weight

For the Number of Shipments, we have individual significance for Tuesday and Thursday as it can be seen from **Figure A3** with p-values below 0.05. This means that there is a significant difference in the Number of Shipments compared to Monday for Tuesday and Thursday, in particular, by amounts -0.95 and -1.12 respectively. Furthermore, when performing the joint significance test, it can also be seen that the daily dummies are jointly significant from **Figure 11**. Hence, the daily dummies are jointly significantly different compared to Monday.

Wald Test:
Equation: Untitled

Test Statistic	Value	df	Probability
F-statistic	8.187043	(4, 197)	0.0000
Chi-square	32.74817	4	0.0000

Figure 11: Joint Wald test on daily dummies for Nb of Shipments

3 Misspecification Tests

Lastly for this question, we are going to investigate various misspecification tests for the data. In the appendix, the misspecification tests can be found including the test for normality, heteroskedasticity, and linearity for each of the three regressions.

For the Gross Volume (m3), the residuals do not follow a normally distributed pattern since the Jarque-Bera test statistic is equal to 30044 with a corresponding p-value of 0.00. This means that we reject the null hypothesis of normality. The test for heteroskedasticity using the White test yields a test statistic of 0.29 with a corresponding p-value of 0.98, hence we do not have enough evidence to reject the null of homoscedasticity. The test for linearity of the Ramsey reset test shows that we have linearity, with a p-value equal to 1. Please refer to **Figure A10, A11 and A12** in the appendix.

For the Gross Weight (kg), the normality test shows that we do not have normally distributed residuals as the test statistic is equal to 1621 with a p-value of 0.00. The White test for heteroskedasticity shows that we do have heteroskedasticity in our residuals as the p-value is eqaul to 0.00. Additionally, the RAMSEY reset test shows that we do not have linearity at a 5% significance level. Please refer to **Figure A13, A14 and A15** in the appendix.

Lastly, for the Number of Shipments, there is enough evidence to reject normality for the residuals as the corresponding p-value of the Jarque-Bera test is equal to 0.00 with a t-statistic of 246.6130. For the White test, We cannot reject the null of homoscedasticity (p-value equal to 0.75), and lastly, the RAMSEY rest test cannot be performed as we have perfect collinearity. Please refer to **Figure A16 and A17** in the appendix.

Question 3: Forecasting

This section of the report focuses on forecasting two weeks of data using both static and dynamic forecasts (**where applicable**) based on the final specifications of the regression models for each of the three series.

First, the three regression equations were reestimated without the last 10 observations and then using built-in commands in EViews, the last 10 observations were forecasted. The only model where it is possible to forecast using dynamic forecasting is for the regression model for the series Gross Weight (kg). The reason being that it is the only model with any dynamics at all, more specifically, a moving average of order 1 (**MA(1)**). For the series Gross Volume (m^3) and Nb of Ship Units, there is no difference between static and dynamic forecasts because they don't have any seasonality or dynamics. The comparisons of the forecasts of the last 10 observations with the actual values for each series can be seen in **Table 2**, **Table 3**, and **Table 4** below:

Dates	Actual	Forecasted (Static)
29/09/2017	1.938	1.298794
02/10/2017	1.372	1.406026
03/10/2017	0	1.519359
04/10/2017	3.963	1.265757
05/10/2017	0.988	1.552026
06/10/2017	0.97	1.298794
09/10/2017	2.256	1.4060126
10/10/2017	0.979	1.519359
11/10/2017	0.027	1.265757
12/10/2017	0.016	1.552026
RMSE		1.238058

Table 2: Forecasts for Gross Volume on last 10 observations

Dates	Actual	Forecasted (Static)	Forecasted (Dynamic)
29/09/2017	307	188.5761	188.5761
02/10/2017	237	200.2096	180.5513
03/10/2017	0	188.0021	181.8949
04/10/2017	455	177.4738	208.6821
05/10/2017	123	225.2666	179.1974
06/10/2017	86	169.0455	186.0216
09/10/2017	253	166.7658	180.5513
10/10/2017	57	196.2097	181.8949
11/10/2017	4	185.5733	208.6821
12/10/2017	2	149.0564	179.1974
RMSE		150.4492	147.7405

Table 3: Forecasts for Gross Weight on last 10 observations

Dates	Original	Forecasted (Static)
29/09/2017	2	2.86
02/10/2017	1	2.87
03/10/2017	0	1.95
04/10/2017	4	2.95
05/10/2017	1	1.74
06/10/2017	2	2.86
09/10/2017	3	2.87
10/10/2017	1	1.95
11/10/2017	2	2.95
12/10/2017	1	1.74
RMSE		1.13189

Table 4: Forecasts for Nb of Shipments on last 10 observations

Suppose that the actual value of the variable at time t is At and the forecasted value of the variable at time t is Ft

then, the Root Mean Squared Error is calculated as follows:

$$\sqrt{\frac{1}{n}\sum_{t=1}^{T}[(A_t - F_t)]^2}$$

It is the square root of the mean squared error, which is what lies inside the square root. Since observations of each series are being forecasted, in this case, the RMSE is a measure of the accuracy of forecasts compared to the actual values. In other words, it can be described as the difference between the actual values and the forecasted values on average.

It is important to note that the static forecasted values are equal to the average sales on a particular day and therefore, each time the same day appears the forecasted values are the same. These forecasted values can be obtained from the regression output. In this case, Monday is the reference group and so, the forecasted value for each date that falls on a Monday would be the value of the coefficient. Then, to obtain the forecasted values for the remainder of the days you would add the coefficient for Monday to that of the dummies for the remaining days. Hence, this might lead to inaccurate forecasts for the last 10 observations, which is evident from the actual and static forecasted values presented in the table. So, it is likely that other more significant variables need to be included in the regression models to more accurately forecast the last 10 observations.

From **Table 2**, it can be seen that the RMSE is equal to 1.238058, which means that the average difference between the forecasted and actual values is 1.238058. In **Table 3**, the RMSE for the static and dynamic forecasts are 150.4492 and 147.7405 respectively, which indicates the average difference between the forecasted and actual values. Similarly, the RMSE in **Table 4** implies that the average difference between the forecasted and actual values is 1.13189.

Next, the three regression equations were estimated for the whole, and then using built-in commands in EViews, future 10 observations were forecasted. For the same reasons as when we reestimated the last 10 observations, the only model where it is possible to forecast using dynamic forecasting is for the regression model for the series Gross Weight (kg). Once again, for the series Gross Volume (m3) and Nb of Ship Units, there is no difference between static and dynamic forecasts. The forecasts for the future 10 observations can be seen in **Table 5**, **Table 6**, and **Table 7** below:

Dates	Forecasted (Static)
13/10/2017	1.307417
16/10/2017	1.425927
17/10/2017	1.469122
18/10/2017	1.303154
19/10/2017	1.499525
20/10/2017	1.307417
23/10/2017	1.425927
24/10/2017	1.469122
25/10/2017	1.303154
26/10/2017	1.499525

Table 5: Forecasts for Gross Volume on future 10 observations

Dates	Forecasted (Static)	Forecasted (Dynamic)
13/10/2017	162.99	162.99
16/10/2017	183.7	183.7
17/10/2017	174.4	174.4
18/10/2017	209.7	209.7
19/10/2017	173.5	173.5
20/10/2017	185.9	185.9
23/10/2017	183.7	183.7
24/10/2017	174.41	174.41
25/10/2017	209.7	209.7
26/10/2017	173.5	173.5

Table 6: Forecasts for Gross Weight on future 10 observations

Dates	Forecasted (Static)
13/10/2017	2.816
16/10/2017	2.829
17/10/2017	1.878
18/10/2017	2.95
19/10/2017	1.707
20/10/2017	2.816
23/10/2017	2.829
24/10/2017	1.878
25/10/2017	2.95
26/10/2017	1.707

Table 7: Forecasts for Nb of Shipments on future 10 observations

Notice that in **Table 2**, **Table 3**, and **Table 4** the RMSE values exist whereas in Table 5, Table 6, and Table 7 the RMSE values do not exist. When the last 10 observations are forecasted, it is possible to compute the RMSE because the actual values exist and so, the formula above is applicable. However, when the future 10 observations are forecasted, the actual values do not exist as we are forecasting out-of-sample values, hence, the formula is not applicable anymore.

Once again, it is important to note that the static forecasted values are equal to the average sales on a particular day and therefore, each time the same day appears the forecasted values are the same. These forecasted values can be obtained from the regression output. In this case, Monday is the reference group and so, the forecasted value for each date that falls on a Monday would be the value of the coefficient. Then, to obtain the forecasted values for the remainder of the days you would add the coefficient for Monday to that of the dummies for the remaining days. The forecasted values for the future 10 observations are likely inaccurate because the Gross Volume, Gross Weight, and Nb of Shipments are not likely to be the same every time the date falls on a particular day. So, it is likely that other more significant variables need to be included in the regression models to more accurately forecast the future 10 observations.

Question 4: Clustering I

For this part of the report, the data structure of the shipments will be discussed that is going to allow further analysis of the clusters. Firstly, the actual data structure and the implementation in Java will be discussed. Secondly, the performance of the current clustering will be assessed.

The shipment list was given initially in an excel file with a lot of information. We have decided to only keep a portion of that. These include the city of Origin, which is where the shipment is originally from, the latitude and longitude of the origin of the shipments, the latitude and longitude of the assigned origin cluster, the weight, volume, and the number of the shipments, and finally the latitude and longitude of the destination cluster and the latitude and longitude of the destination. The weight corresponds to total weight being transported regardless of the number of shipments; the same assumption is made for the volume. This information is printed on a text file, which is then read through Java using a method to read the files. A class called shipment is created which will assign to each shipment from the text file to an object that we are going to call "Shipment", these will contain all the data including the shipments weights, volume, number and all the important coordinates but also the city name. As these data are homogenous, this means that we have an object which contains integers (the number of shipments and the cluster numbers), doubles (the various coordinates and the weight and volume) but also strings (the city name). Of course, appropriate setter-getter methods are created to have a flexible data structure which can allow us to later reassign the clusters.

Now that we have the explanation of the data structure, we are going to measure the performance of the clustering that was given to us initially by the shipping company. To do so, we are going to get the average distance that needs to be travelled from the origin to the origin cluster, the average weight, volume, and number of shipments that need to be carried out based on this specific clustering, and the same for the destination cluster. The specific Java method for the origin and destination clusters are constructed in the following way: For the origin cluster, we need to go through the whole array list of shipments and select the shipment that belongs to a specific origin cluster. For example, we first get the information about origin cluster 1, we sum up all the distances to the origin, the gross weight, the gross volume, and the number of shipments of ALL the shipments that have cluster origin 1 using the appropriate getter methods. It is important to note that to calculate the distance between the origin and the origin cluster, we use a method that converts 2 coordinates given in latitude and longitude into a distance in kilometres, this method is called the haversine method. Now that we have all the aggregate data for cluster 1, we are going to do the same for cluster 2 up to the last origin cluster. Now that we have all the information for each cluster, we are just going to take the data for the average cluster: so, the sum of all the distance, weight, volume number of shipments of all the clusters divided by the number of clusters. The same procedure is performed for the destination cluster. The summary of the results can be found in **Table 8** below:

	Average Distance to Cluster per cluster (km)	Average Weight per cluster (kg)	Average Volume per cluster (cubic meters)	Average Number of shipments
Origin Cluster	443	45786	1033	230
Destination Cluster	351	45792	1033	230

Table 8: Average Performance of the initial clustering assignment

Additionally, the highest (and lowest) 3 values are given for the four categories. It can be found in **Table 9** and **Table 10**. Note that the values are listed from highest to lowest for the highest 3 values and lowest to highest for the lowest 3 values.

	Distance to the cluster (km)	Weight transported (kg)	Volume transported (cubic meters)	Number of shipments
Origin Cluster	373875 **(Cluster 2)** 13023 **(Cluster 630)** 12558 **(Cluster 463)**	6262102 **(Cluster 2)** 5195074 **(Cluster 901)** 3061947 **(Cluster 15)**	122109 **(Cluster 443)** 106330 **(Cluster 19)** 104512 **(Cluster 15)**	53341 **(Cluster 2)** 24849 **(Cluster 1)** 11964 **(Cluster 15)**
Destination Cluster	292411 **(Cluster 2)** 11626 **(Cluster 463)** 11286 **(Cluster 441)**	7161821 **(Cluster 2)** 4058932 **(Cluster 905)** 3552405 **(Cluster 1)**	163917 **(Cluster 2)** 100626 **(Cluster 446)** 82260 **(Cluster 284)**	49131 **(Cluster 2)** 31090 **(Cluster 1)** 18815 **(Cluster 15)**

Table 9: The highest values for the distance to cluster, the weight, the volume, and the number of shipments.

	Distance to the cluster (km)	Weight transported (kg)	Volume transported (cubic meters)	Number of shipments
Origin Cluster	0 0 0	0 0 0	0 0 0	0 0 0
Destination Cluster	0 0 0	0 0 0	0 0 0	0 0 0

Table 10: The lowest values for the distance to cluster, the weight, the volume, and the number of shipments.

It is important to note that for the lowest values, we have a lot of zeros. This can be explained by the fact that that for a few clusters, all the shipments are from the same origin coordinates, and it is assigned to the same coordinates for its origin cluster, making it a total distance of 0 from the origin to the origin cluster. For the other categories is because some shipments have 0 weight, volume, or number of shipments. So, we decided to consider the lowest values that are larger than 0 for each category. This is summarised in **Table 11** below:

	Distance to the cluster (km)	Weight transported (kg)	Volume transported (cubic meters)	Number of shipments
Origin Cluster	0.48 **(Cluster 558)** 0.67 **(Cluster 48)** 0.94 **(Cluster 195)**	0.22 **(Cluster 605)** 0.5 0.5	0.001 **(Cluster 195)** 0.002 **(Cluster 495)** 0.002 **(Cluster 998)**	1 1 1
Destination Cluster	0.497 **(Cluster 438)** 1.19 **(Cluster 772)** 1.648 **(Cluster 476)**	0.015 **(Cluster 521)** 0.25 **(Cluster 810)** 0.5 **(Cluster 92)**	0.001 **(Cluster 537)** 0.002 **(Cluster 810)** 0.003 **(Cluster 884)**	1 1 1

Table 11: The lowest values for the distance to cluster, the weight, the volume, and the number of shipments (larger than 0)

Note that in **Table 11** the weight for the origin cluster is 0.5 for both the 3rd lowest value and 2nd lowest value. The reason we did not add which cluster it belongs to is because there is a tie in weights for the origin cluster when the weight is 0.5. The clusters that are tied are cluster 304, 440 and 525. Similarly, the lowest 3 values for the number of shipments for both the origin and destination cluster are tied at 1. The clusters that are all tied at 1 are can be seen in the figure below:

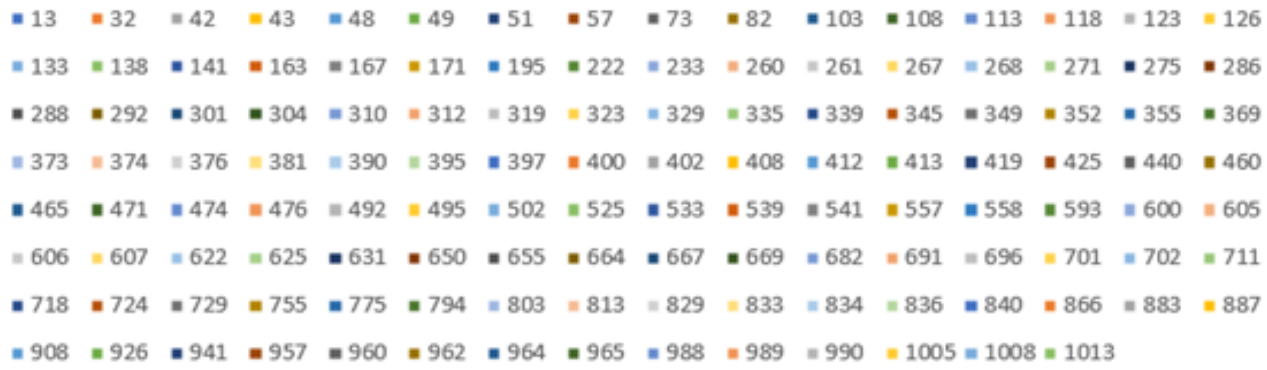


Figure 12: All clusters that have number of shipments equal to 1

Question 5: Clustering II

The first step was to create an object class that contained all the features of a centroid. These features included the latitude, longitude and cluster number of the centroid. Then, within this object class a method was introduced that computes the haversine distance between a particular centroid a shipment based on their respective latitude and longitude values. The same method was implemented twice, one for the origin clusters and the other for the destination clusters.

Then, using excel the maximum and minimum values for the latitude and longitude were obtained from all the locations in the data set to create bounds. The purpose of these bounds were for creating random latitude and longitude pairs within these bounds to represent the k number of centroids that the user wishes to input. The method that randomly generates these latitude and longitude pairs is called *kRandomClusters* and can be found in the class Test.java. It returns an array of size 2 where the element in the position 0 respresents the randomly generated latitude and the element in position 1 represents the randomly generated longitude.

The next step was to create another method called *centroidMaker* that returns a list which has a type of the object class centroid that was created. The purpose of this method is return k random centroids that have a cluster number associated with it. The method starts of with count variable assigned to a value 1. Then, a HashMap is created that allows us to assign cluster numbers to a particular centroid. The HashMap has a key of type Integer and value that is an array of type double. It looks as follows: HashMap<Integer, double[]>. The key represents the cluster number assigned to the centroid and the double[] represents the latitude and longitude of that particular centroid. After that, the method runs a for loop k number of times and in each iteration it adds to the HashMap the the variable count in the key position and calls the method *kRandomClusters* in the value position. Then, we create an instance of the centroid object class and for the parameters we include the current count value (associated cluster of that centroid), the latitude, and longitude of the centroid by getting the value of the key at the current count value. Lastly, we add this centroid to a list and increase the count value to ensure that we are moving to the next centroid. Doing this k times, we are able to generate k random centroids each belonging to one of the k clusters.

The final step was to create a method that actually assigns locations to clusters and keeps reassigning them to new clusters as long as the centroid changes positions. The method is called *clusterAssignmentOrigin* for the origin clusters and another method also exists called *clusterAssignmentDestination* for the destinations clusters, which is mostly the same. First we needed to initialise the locations to clusters so we call the method *centroidMaker* to create the initial k centroids. The method *centroidMaker* is assigned to a variable List which has a type of the object class

centroid that was created. Then, we create a for loop that runs as long as the number of locations isn't exceeded and get the first location from list of locations immediately after. After this, another for loop is created within the first that goes through the list containing the k number of centroids. The reason why the second for loop is nested within the first is because we want to compute the distance between current location and each of the k centroids. Next, we get the i^{th} centroid from the list of k centroids and compute the distance between the current location and the particular centroid. This same process is done k times and each time we update the minimum distance to be able to find which centroid the particular shipment is closest to. At the same time we keep updating the cluster assigned to the location only if the minimum distance changes. The next step of the initialisation process is setting the origin cluster of the particular shipment to the cluster associated with the minimum distance. This is done right after the for loop for the k centroids ends. Finally, the whole process runs for all the locations and the initial assignment of shipments of clusters is complete. Note that this is not the end of the method but just the initialisation step that assigns locations to clusters. The next steps are explained in the paragraph below.

After the initialisation, the centroids need to be updated as long as the latitude and longitude of the centroid changes. First, a boolean variable called *complete* is defined to check if the centroid before and after being updated is the same for all the clusters. A variable *counter* is also defined that basically counts the number of clusters whose centroids haven't changed before and after the update step. Then, a while loop is created, which exists to continuously update centroids and reassign shipments to new clusters as long as the centroids keep changing positions. The while loop executes as long as the boolean variable *complete* is false. Next, we proceed with the code in the while loop. So, first we define a for loop that goes through all the k clusters. We introduce temporary latitude and longitude variables that are assigned to the value of the centroids latitude and longitude before the update step for the k^{th} cluster. Within the first for loop another for loop is defined that goes through all the locations in the list of locations. The method then gets locations one at a time from the list and checks if the cluster associated to that location is equal to the current value of k in the first for loop. If that is the case, we add the latitude, longitude and also count the number of locations for that particular cluster. This step repeats for the number of locations in the list that are associated to the cluster k and finally exits the second for loop. This allows us to get the total latitude, total longitude and total number of locations for that cluster. Since we are still within the first for loop, we set the latitude of the centroid for cluster k to the total latitude of that cluster divided by the number of shipments of the cluster. Similarly, for the longitude of the centroid for cluster k, we set the longitude for the centroid to the total longitude of that cluster divided by the number of locations of the cluster. Immediately after, we check if the temporary latitude and longitude variables for the centroid before the update step are equal to the newly set latitude and longitude values for the same centroid. If so, we increase our *counter* variable by 1. After that, we reassign the locations to new clusters using the updated centroids. This step is done in exactly the same way as the initial assignment. Lastly, right before an iteration of the while loop ends, we check if the *counter* variable is equal to the number of clusters and if this is true, we set the boolean variable *complete = true*. We do this because it must mean that all the centroids for each cluster remain in the same position after the update step. If it isn't true, *complete = false*, and the method goes through the while loop again. The whole process explained above runs k number of times (**the number of clusters**).

Question 6: Clustering III

For this part of the paper, an alternative clustering will be proposed. This alternative idea consists of clustering the shipments based on their gross weight into various weight categories. The reason for this is because it is probably more useful to assign different sized transport vehicles to different weights of shipments. For example, using huge trucks for small shipments is probably not going to be cost-efficient since trucks of such sizes usually have higher variable costs such as the cost of petrol but also higher fixed costs, since they are more expensive to buy. Therefore, we decided to cluster the shipments into different weight groups to assess different vehicle sizes to different weight groups. The first weight group is going to be for courier vans which can carry weights between 0 and 500kg, then we have the transit vans which can carry weights between 501kg and 1500kg, then we have trucks which can carry weights between 1501kg and 9000kg, then we have trailers which carry between 9001kg and 40000kg and lastly, we

have cargo ships or planes which carry from 40001kg to 242000kg.

The implementation of this in Java can be done in the following way. Five new array lists of shipments are created, which will all correspond to one specific list of shipments belonging to a specific weight group. Then we need to go through the whole list of shipments and check in which weight group they belong by using a getter method to access the shipment's weight. After we have successfully identified into which weight group they belong, we simply just add this shipment to the new array list of the shipments belonging to that weight category. After performing all the above, the following results were found in **Table 12**:

	0kg-500kg: Courier Van	501kg-1500kg: Transit Van	1501kg-9000kg: Trucks	9001kg-40000kg: trailers	40001kg-242000kg: Cargo ships or planes
Number of Shipments	**65802**	**8443**	**6596**	**658**	**21**

Table 12: The number of shipments per weight category based on the alternative clustering

It is important to note that it is possible to go further with his clustering proposition, it is for example possible to implement the clustering algorithm that was discussed in question 5, to each specific weight group which can be used later for example to create new subclusters. This can help if we would like to save time and truck trips by aggregating shipments into the specific vehicle if they can fit within the weight constraints of the vehicle. This of course can save time, truck trips but also money since the vehicle will not have to go multiple times and use less gas.

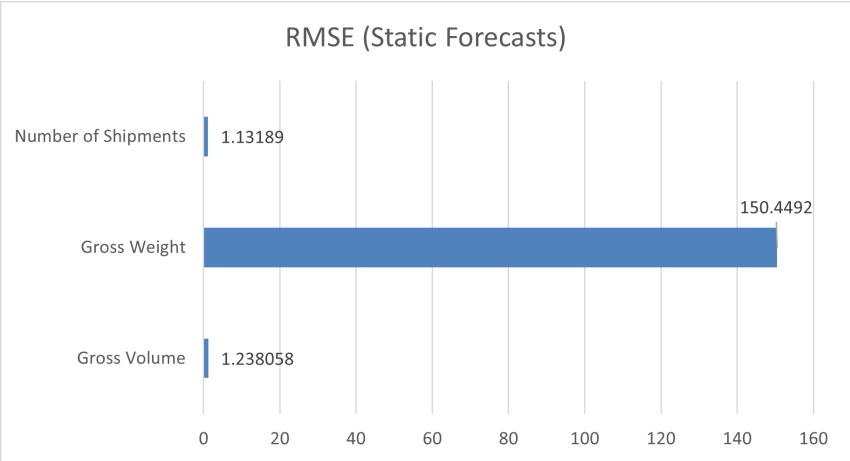
Business Summary

The origin cluster that was investigated is cluster 1 and its most frequent destination cluster was destination cluster 15. We performed data analysis on the shipments for the lane that links cluster 1 to cluster 15. The data for the volume, weight and number of shipments for each shipment was aggregated to have daily data.This formed the basis for building our models for the volume, weight and number of shipments. The main goal is capture the difference in daily effects on each of the series.

Conducting tests on the constructed models for Gross Volume and Gross Weight we find the following results:

- **Gross Volume Weight**: The results of the test indicate that on average the gross volume and gross weight of shipments for the lane from Tuesday to Friday are not significantly different from Monday. Hence, regardless of the day, the gross volume and gross weight of shipments is on average around the same.
- **Number of Shipments**: The results of the test indicate that on average the number of shipments on Tuesday and Thursday are significantly different from Monday. In particular, the number of shipments is smaller on Tuesday and Thursday in comparison to Monday. This could potentially be a cost-saving strategy because the shipping company could deploy less trucks to transport shipments on Tuesday and Thursday than on Monday. However, it is important to consider that the number of shipments is not easy to predict because it depends on the number of people who make orders, which could differ drastically in short periods of time.

In order to determine the performance of our model we forecasted the last 10 observations and compared it to the actual values using the Root Mean Squared Error (RMSE). It is a measure used to identify the difference between the actual values and forecasted values on average. The RMSE for the forecasts of the last 10 observations for each model can be seen in the bar graph below:



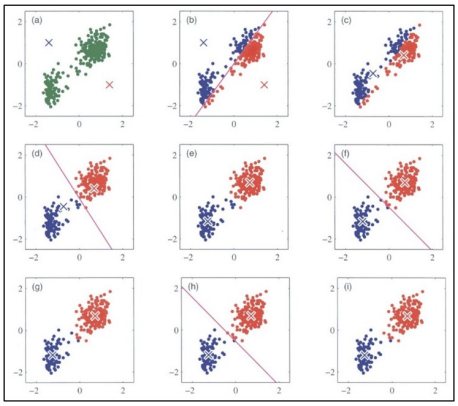
It is obvious from the graph that the RMSE values are not ideal. For example, the RMSE for the model with Gross Weight has a RMSE of 150.4492, which means that on average the difference between the actual weight and forecasted value is 150.4492. This could lead to misleading results when forecasting for the future, which will be costly. Similar conclusions can be made for Gross Volume and Number of Shipments. Hence, other significant factors need to be taken into consideration to improve the model and yield better forecasts.

In addition to performing data analysis, we have analysed the clustering performance in detail of the all locations in the data set. In the figure below, the various results can be found including the average performance per cluster and the clusters with the highest 3 values (**Ranked from highest to lowest**):

	Average Distance to Cluster per cluster (km)	Average Weight per cluster (kg)	Average Volume per cluster (cubic meters)	Average Number of shipments
Origin Cluster	443	45792	1033	230
Destination Cluster	351	45792	1033	230

	Distance to the cluster (km)	Weight transported (kg)	Volume transported (cubic meters)	Number of shipments
Origin Cluster	373875 (Cluster 2) 13023 (Cluster 630) 12558 (Cluster 463)	6262102 (Cluster 2) 5195074 (Cluster 901) 3061947 (Cluster 15)	122109 (Cluster 442) 106330 (Cluster 19) 104512 (Cluster 15)	53341 (Cluster 2) 24849 (Cluster 1) 11964 (Cluster 15)
Destination Cluster	292411 (Cluster 2) 11626 (Cluster 463) 11286 (Cluster 441)	7161821 (Cluster 2) 4058932 (Cluster 905) 3552405 (Cluster 1)	163917 (Cluster 2) 100626 (Cluster 446) 82260 (Cluster 284)	49131 (Cluster 2) 31090 (Cluster 1) 18815 (Cluster 15)

Our suggestion based on the results is that we would suggest making some changes to the clusters highlighted in red since they are the ones that have the highest values and are the most "important" clusters. Our results also show that the average distance to travel up to the origin cluster is equal to 443km and the average distance from the destination cluster to the destination is 351km. We would also like to suggest an alternative way of clustering the shipments. The method is called the k-means clustering algorithm, this is explained below:



First, we assign k number of random coordinates on the map (this can be seen on panel a), then we need to assign all the shipments (all the points in this case) to the nearest earlier assigned cluster coordinates (this can be seen on panel b). After this, new cluster coordinates are reassigned based on the average coordinates of all the shipments that belong to the cluster (this can be seen in panel c), and then once again we reassign the shipments based on these new coordinates (this can be seen in panel d). This last step is repeated until the clustering does not change anymore. Hence, obtaining the optimal clustering.

One last suggestion concerning the clustering, would be a new clustering which would assign the weights of the shipments based on different weight categories. This can be helpful when determining the appropriate vehicles that will be used to transport the shipment. This is useful since, smaller vehicles with smaller engines (or even electrical engines), cost less money to sustain than large trailer tucks with big fuel con summing engines. We also made it in such a way that it is possible to check the number of shipments per weight category, which can help in two ways: first one is that it is possible now to arrange the shipments in such a way that more than 1 shipment fit into the vehicle, which can help safe costs again. Secondly, by displaying the number of shipments per weight cluster, the management team can determine in advance the number of trucks that are needed per weight category.

	0kg-500kg: Currier Van	501kg-1500kg: Transit Van	1501kg-9000kg: Trucks	9001kg-40000kg: trailers	40001kg-242000kg: Cargo ships or planes
Number of Shipments	**65802**	**8443**	**6596**	**658**	**21**

Appendix

Dependent Variable: GROSS VOLUME M3
Method: Least Squares
Date: 06/18/21 Time: 16:47
Sample (adjusted): 1/02/2017 10/12/2017
Included observations: 204 after adjustments

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	1.425927	0.269743	5.286233	0.0000
DUM6JAN	46.18558	1.751025	26.37631	0.0000
DUM17FEB	17.17158	1.751025	9.806587	0.0000
DUM17MARCH	53.95658	1.751025	30.81428	0.0000
DUM19APRIL	27.78685	1.749204	15.88542	0.0000
DUM28APRIL	52.51358	1.751025	29.99019	0.0000
DUM11MAY	17.46548	1.748658	9.987933	0.0000
DUM6SEP	18.22185	1.749204	10.41722	0.0000
@WEEKDAY=2	0.043195	0.381475	0.113232	0.9100
@WEEKDAY=3	-0.122773	0.386335	-0.317789	0.7510
@WEEKDAY=4	0.073598	0.383852	0.191736	0.8482
@WEEKDAY=5	-0.118510	0.394498	-0.300407	0.7642

R-squared	0.941409	Mean dependent var	2.545123
Adjusted R-squared	0.938052	S.D. dependent var	6.939533
S.E. of regression	1.727201	Akaike info criterion	3.987904
Sum squared resid	572.7788	Schwarz criterion	4.183087
Log likelihood	-394.7662	Hannan-Quinn criter.	4.066859
F-statistic	280.4507	Durbin-Watson stat	1.832002
Prob(F-statistic)	0.000000		

Figure A1: Regression output Gross Volume

Dependent Variable: GROSS WEIGHT KG
Method: Least Squares
Date: 06/18/21 Time: 17:10
Sample: 1/02/2017 10/12/2017
Included observations: 204

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	183.6951	32.58967	5.636605	0.0000
DUM6JAN	5815.511	211.4769	27.49951	0.0000
DUM17MARCH	6459.511	211.4769	30.54476	0.0000
DUM28APRIL	5015.511	211.4769	23.71659	0.0000
@WEEKDAY=2	-9.282927	46.08876	-0.201414	0.8406
@WEEKDAY=3	26.00244	46.08876	0.564182	0.5733
@WEEKDAY=4	-10.19024	46.08876	-0.221100	0.8252
@WEEKDAY=5	1.794067	47.31801	0.037915	0.9698

R-squared	0.920833	Mean dependent var	270.1167
Adjusted R-squared	0.918006	S.D. dependent var	728.7542
S.E. of regression	208.6757	Akaike info criterion	13.55787
Sum squared resid	8534930.	Schwarz criterion	13.68799
Log likelihood	-1374.902	Hannan-Quinn criter.	13.61050
F-statistic	325.6847	Durbin-Watson stat	1.659687
Prob(F-statistic)	0.000000		

Figure A2: Regression output Gross Weight

Dependent Variable: NB OF SHIP UNITS
Method: Least Squares
Date: 06/18/21 Time: 20:21
Sample: 1/02/2017 10/12/2017
Included observations: 204

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	2.829268	0.206462	13.70356	0.0000
DUM17MARCH	46.18421	1.339286	34.48420	0.0000
DUM28APRIL	39.18421	1.339286	29.25754	0.0000
@WEEKDAY=2	-0.951220	0.291982	-3.257803	0.0013
@WEEKDAY=3	0.121951	0.291982	0.417667	0.6766
@WEEKDAY=4	-1.121951	0.291982	-3.842537	0.0002
@WEEKDAY=5	-0.013479	0.297689	-0.045278	0.9639

R-squared	0.916017	Mean dependent var	2.852941
Adjusted R-squared	0.913459	S.D. dependent var	4.493887
S.E. of regression	1.322004	Akaike info criterion	3.429886
Sum squared resid	344.2959	Schwarz criterion	3.543743
Log likelihood	-342.8484	Hannan-Quinn criter.	3.475943
F-statistic	358.1186	Durbin-Watson stat	1.837646
Prob(F-statistic)	0.000000		

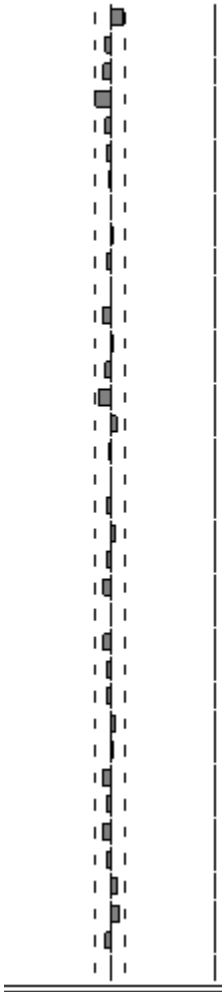
Figure A3: Regression output Nb of Shipments

Date: 06/18/21 Time: 16:48
Sample (adjusted): 1/02/2017 10/12/2017
Included observations: 204 after adjustments

Autocorrelation	Partial Correlation		AC	PAC	Q-Stat	Prob
		1	0.041	0.041	0.3451	0.557
		2	-0.021	-0.023	0.4355	0.804
		3	-0.038	-0.036	0.7359	0.865
		4	-0.094	-0.092	2.5885	0.629
		5	-0.027	-0.022	2.7444	0.739
		6	0.012	0.009	2.7758	0.836
		7	-0.064	-0.073	3.6465	0.819
		8	0.141	0.139	7.9175	0.442
		9	-0.015	-0.034	7.9635	0.538
		10	-0.036	-0.032	8.2505	0.604
		11	-0.050	-0.052	8.7993	0.640
		12	-0.014	0.009	8.8418	0.716
		13	0.006	0.006	8.8499	0.784
		14	0.005	-0.016	8.8551	0.840
		15	-0.064	-0.055	9.7580	0.835
		16	-0.039	-0.060	10.090	0.862
		17	-0.046	-0.043	10.573	0.878
		18	0.026	0.025	10.720	0.906
		19	0.041	0.037	11.104	0.920
		20	0.054	0.038	11.774	0.924
		21	-0.023	-0.040	11.891	0.943
		22	-0.033	-0.035	12.141	0.954
		23	-0.011	0.011	12.169	0.968
		24	-0.038	-0.029	12.512	0.973
		25	-0.020	-0.013	12.606	0.981
		26	0.060	0.041	13.470	0.979
		27	0.074	0.057	14.764	0.973
		28	-0.001	-0.031	14.764	0.981
		29	-0.060	-0.051	15.629	0.979
		30	-0.052	-0.025	16.289	0.980
		31	-0.016	-0.012	16.354	0.986
		32	-0.048	-0.060	16.909	0.987
		33	0.162	0.170	23.372	0.893
		34	-0.019	-0.052	23.461	0.913
		35	0.009	-0.001	23.479	0.931
		36	0.060	0.066	24.392	0.929

Figure A4: Q-test on residuals of Gross Volume regression

Date: 06/18/21 Time: 17:11
Sample: 1/02/2017 10/12/2017
Included observations: 204



Autocorrelation	Partial Correlation		AC	PAC	Q-Stat	Prob
		1	0.139	0.139	4.0117	0.045
		2	-0.038	-0.058	4.3095	0.116
		3	-0.056	-0.043	4.9672	0.174
		4	-0.136	-0.127	8.8632	0.065
		5	-0.044	-0.012	9.2789	0.098
		6	-0.023	-0.030	9.3875	0.153
		7	-0.012	-0.019	9.4178	0.224
		8	0.010	-0.008	9.4372	0.307
		9	0.043	0.034	9.8439	0.363
		10	-0.022	-0.043	9.9505	0.445
		11	0.011	0.019	9.9761	0.533
		12	-0.074	-0.083	11.166	0.515
		13	0.043	0.076	11.565	0.564
		14	-0.045	-0.081	12.019	0.605
		15	-0.103	-0.085	14.393	0.496
		16	0.074	0.083	15.605	0.481
		17	-0.004	-0.032	15.609	0.552
		18	0.029	0.019	15.798	0.607
		19	-0.025	-0.056	15.938	0.661
		20	0.051	0.081	16.526	0.684
		21	-0.031	-0.057	16.750	0.726
		22	-0.056	-0.051	17.481	0.736
		23	0.029	0.053	17.672	0.775
		24	-0.050	-0.067	18.266	0.790
		25	-0.020	-0.014	18.360	0.827
		26	-0.022	-0.041	18.479	0.858
		27	0.060	0.064	19.340	0.857
		28	0.037	0.029	19.663	0.876
		29	-0.066	-0.130	20.715	0.869
		30	-0.029	0.021	20.918	0.890
		31	-0.064	-0.062	21.898	0.886
		32	-0.025	-0.005	22.050	0.906
		33	0.077	0.057	23.517	0.888
		34	0.098	0.056	25.909	0.839
		35	-0.048	-0.049	26.491	0.849
		36	0.012	-0.029	26.528	0.875

Figure A5: Q-test on residuals of Gross Weight regression

Dependent Variable: GROSS WEIGHT KG
Method: ARMA Maximum Likelihood (OPG - BHHH)
Date: 06/18/21 Time: 17:15
Sample: 1/02/2017 10/12/2017
Included observations: 204
Convergence achieved after 18 iterations
Coefficient covariance computed using outer product of gradients

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	183.6951	36.77167	4.995560	0.0000
DUM6JAN	5808.231	1676.995	3.463475	0.0007
DUM17MARCH	6490.503	3005.795	2.159330	0.0321
DUM28APRIL	5055.223	719.5220	7.025808	0.0000
@WEEKDAY=2	-9.282927	46.93701	-0.197774	0.8434
@WEEKDAY=3	26.00244	48.54391	0.535648	0.5928
@WEEKDAY=4	-10.19024	50.58156	-0.201462	0.8405
@WEEKDAY=5	2.210768	58.30304	0.037919	0.9698
MA(1)	0.161784	0.059980	2.697302	0.0076
SIGMASQ	40885.52	3277.635	12.47409	0.0000

R-squared	0.922636	Mean dependent var	270.1167
Adjusted R-squared	0.919046	S.D. dependent var	728.7542
S.E. of regression	207.3476	Akaike info criterion	13.55458
Sum squared resid	8340646.	Schwarz criterion	13.71723
Log likelihood	-1372.567	Hannan-Quinn criter.	13.62037
F-statistic	257.0679	Durbin-Watson stat	1.963267
Prob(F-statistic)	0.000000		

Inverted MA Roots	-.16

Figure A6: Gross Weight regression with MA(1)

Date: 06/18/21 Time: 17:23
Sample (adjusted): 1/02/2017 10/12/2017
Q-statistic probabilities adjusted for 1 ARMA term

Autocorrelation	Partial Correlation		AC	PAC	Q-Stat	Prob*
		1	-0.012	-0.012	0.0299	
		2	-0.034	-0.034	0.2695	0.604
		3	-0.029	-0.030	0.4514	0.798
		4	-0.127	-0.129	3.8377	0.280
		5	-0.020	-0.026	3.9182	0.417
		6	-0.020	-0.032	4.0041	0.549
		7	-0.017	-0.028	4.0649	0.668
		8	0.007	-0.014	4.0750	0.771
		9	0.045	0.036	4.5175	0.808
		10	-0.034	-0.043	4.7743	0.854
		11	0.031	0.026	4.9822	0.892
		12	-0.088	-0.092	6.6637	0.826
		13	0.062	0.071	7.5219	0.821
		14	-0.037	-0.053	7.8304	0.854
		15	-0.110	-0.106	10.507	0.724
		16	0.096	0.075	12.574	0.635
		17	-0.024	-0.022	12.698	0.695
		18	0.039	0.026	13.045	0.733
		19	-0.040	-0.065	13.400	0.767
		20	0.060	0.078	14.230	0.770
		21	-0.033	-0.037	14.481	0.805
		22	-0.056	-0.065	15.197	0.813
		23	0.043	0.051	15.622	0.834
		24	-0.059	-0.060	16.441	0.836
		25	-0.006	-0.015	16.449	0.871
		26	-0.028	-0.049	16.634	0.895
		27	0.055	0.044	17.347	0.898
		28	0.041	0.059	17.748	0.911
		29	-0.071	-0.127	18.961	0.899
		30	-0.009	0.011	18.983	0.922
		31	-0.055	-0.057	19.726	0.923
		32	-0.029	-0.024	19.934	0.937
		33	0.067	0.046	21.023	0.931
		34	0.097	0.070	23.364	0.893
		35	-0.068	-0.035	24.499	0.885
		36	0.025	-0.040	24.659	0.904

*Probabilities may not be valid for this equation specification.

Figure A7: Q-test on residuals of Gross Weight regression with MA(1)

Dependent Variable: GROSS WEIGHT KG
Method: ARMA Maximum Likelihood (OPG - BHHH)
Date: 06/23/21 Time: 11:35
Sample: 1/02/2017 10/12/2017
Included observations: 204
Convergence achieved after 17 iterations
Coefficient covariance computed using outer product of gradients

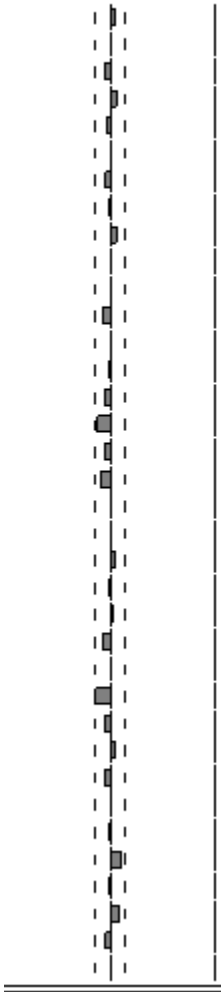
Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	183.6035	36.89592	4.976257	0.0000
DUM6JAN	5797.780	2320.043	2.498997	0.0133
DUM17MARCH	6483.997	3206.624	2.022064	0.0445
DUM28APRIL	5034.218	1086.307	4.634251	0.0000
@WEEKDAY=2	-9.196719	46.15475	-0.199258	0.8423
@WEEKDAY=3	26.14873	48.06417	0.544038	0.5870
@WEEKDAY=4	-9.717470	50.45998	-0.192578	0.8475
@WEEKDAY=5	3.237600	58.03258	0.055789	0.9556
AR(1)	0.148800	0.059956	2.481819	0.0139
SIGMASQ	40966.58	3323.493	12.32636	0.0000

R-squared	0.922482	Mean dependent var	270.1167
Adjusted R-squared	0.918886	S.D. dependent var	728.7542
S.E. of regression	207.5530	Akaike info criterion	13.55654
Sum squared resid	8357183.	Schwarz criterion	13.71919
Log likelihood	-1372.767	Hannan-Quinn criter.	13.62233
F-statistic	256.5166	Durbin-Watson stat	1.938325
Prob(F-statistic)	0.000000		

Inverted AR Roots	.15

Figure A8: Gross Weight regression with AR(1)

Date: 06/18/21 Time: 20:23
Sample: 1/02/2017 10/12/2017
Included observations: 204



Autocorrelation	Partial Correlation		AC	PAC	Q-Stat	Prob
		1	0.055	0.055	0.6305	0.427
		2	0.023	0.020	0.7448	0.689
		3	-0.033	-0.036	0.9733	0.808
		4	0.070	0.073	1.9900	0.738
		5	-0.030	-0.037	2.1823	0.823
		6	0.032	0.032	2.4019	0.879
		7	-0.039	-0.037	2.7347	0.908
		8	-0.007	-0.011	2.7453	0.949
		9	0.081	0.092	4.1724	0.900
		10	0.018	-0.001	4.2421	0.936
		11	0.030	0.033	4.4337	0.955
		12	-0.078	-0.081	5.7732	0.927
		13	0.025	0.025	5.9130	0.949
		14	-0.005	0.002	5.9181	0.969
		15	-0.040	-0.059	6.2762	0.975
		16	-0.117	-0.092	9.3614	0.898
		17	-0.037	-0.033	9.6626	0.917
		18	-0.098	-0.093	11.847	0.855
		19	0.026	0.030	11.996	0.886
		20	0.015	0.019	12.050	0.914
		21	0.056	0.064	12.779	0.916
		22	-0.007	-0.001	12.792	0.939
		23	0.033	0.022	13.045	0.951
		24	-0.073	-0.072	14.274	0.940
		25	0.001	0.015	14.275	0.957
		26	-0.124	-0.115	17.877	0.880
		27	-0.042	-0.024	18.286	0.895
		28	0.057	0.070	19.073	0.896
		29	-0.039	-0.061	19.433	0.910
		30	0.020	0.024	19.532	0.928
		31	-0.013	-0.023	19.572	0.945
		32	0.116	0.100	22.857	0.883
		33	-0.001	-0.002	22.858	0.907
		34	0.103	0.067	25.483	0.854
		35	-0.045	-0.012	25.994	0.865
		36	0.016	-0.007	26.054	0.889

Figure A9: Q-test on residuals of Nb of Shipments

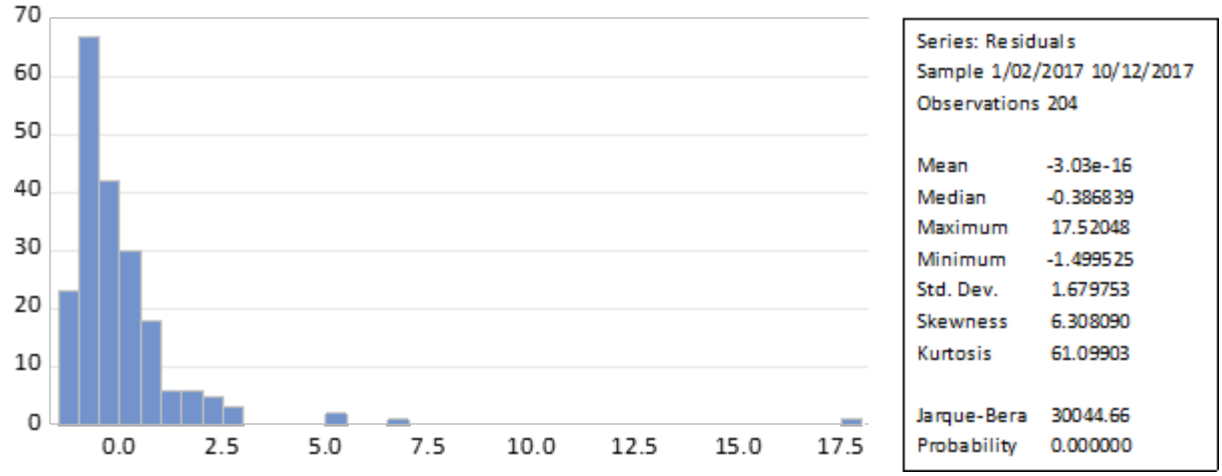


Figure A10: Normality test Gross Volume

Heteroskedasticity Test: White
Null hypothesis: Homoskedasticity

F-statistic	0.298600	Prob. F(11,192)	0.9856
Obs*R-squared	3.431185	Prob. Chi-Square(11)	0.9837
Scaled explained SS	91.33215	Prob. Chi-Square(11)	0.0000

Figure A11: Homoscedasticity test Gross Volume

Ramsey RESET Test
Equation: UNTITLED
Omitted Variables: Squares of fitted values
Specification: GROSS VOLUME M3 C @EXPAND(@WEEKDAY, @DROP(1)) DUM6JAN DUM17FEB DUM17MARCH DUM19APRIL DUM28APRIL DUM11MAY DUM6SEP

	Value	df	Probability
t-statistic	0.000000	191	1.0000
F-statistic	0.000000	(1, 191)	1.0000
Likelihood ratio	0.000000	1	NA

F-test summary:

	Sum of Sq.	df	Mean Squares
Test SSR	0.000000	1	0.000000
Restricted SSR	572.7788	192	2.983223
Unrestricted SSR	572.7788	191	2.998842

LR test summary:

	Value
Restricted LogL	-394.7662
Unrestricted LogL	-394.7662

Figure A12: Linearity test Gross Volume

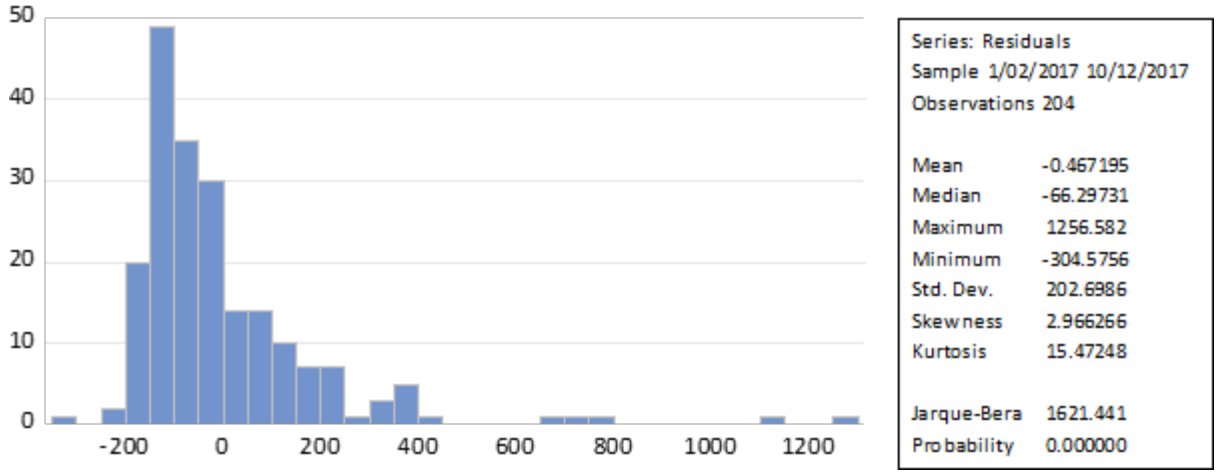


Figure A13: Normality test Gross Weight

Heteroskedasticity Test: White
Null hypothesis: Homoskedasticity

F-statistic	32414.24	Prob. F(10,193)	0.0000
Obs*R-squared	203.8786	Prob. Chi-Square(10)	0.0000
Scaled explained SS	1331.682	Prob. Chi-Square(10)	0.0000

Figure A14: Homoscedasticity test Gross Weight

Ramsey RESET Test
Equation: UNTITLED
Omitted Variables: Squares of fitted values
Specification: GROSS WEIGHT KG C @EXPAND(@WEEKDAY,@ DROP(1)) DUM6JAN DUM17MARCH DUM28APRIL MA(1)

	Value	df	Probability
t-statistic	3.710089	193	0.0003
F-statistic	13.76476	(1, 193)	0.0003
Likelihood ratio	13.02614	1	0.0003

WARNING: the MA backcasts differ for the original and test equation. Under the null hypothesis, the impact of this difference vanishes asymptotically.

F-test summary:

	Sum of Sq.	df	Mean Squares
Test SSR	555254.2	1	555254.2
Restricted SSR	8340646.	194	42993.02
Unrestricted SSR	7785392.	193	40338.82

LR test summary:

	Value
Restricted LogL	-1372.567
Unrestricted LogL	-1366.054

Figure A15: Linearity test Gross Weight

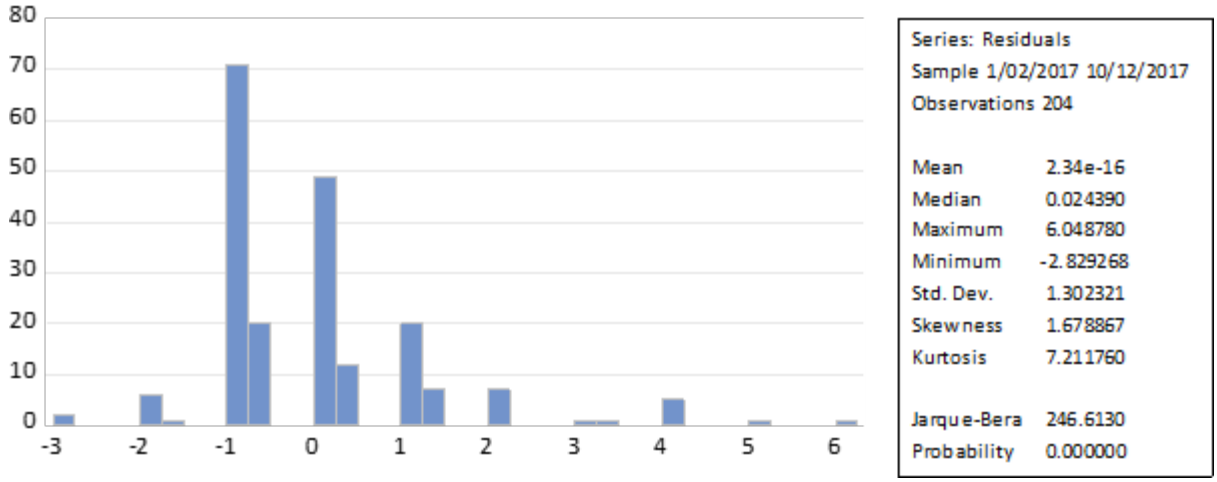


Figure A16: Normality test Nb of Shipments

Heteroskedasticity Test: White
Null hypothesis: Homoskedasticity

F-statistic	0.568873	Prob. F(6,197)	0.7548
Obs*R-squared	3.474322	Prob. Chi-Square(6)	0.7474
Scaled explained SS	10.06298	Prob. Chi-Square(6)	0.1220

Figure A17: Homoscedasticity test Nb of Shipments