Second Year Project Report

Introduction:

For this year’s second year project, we were given an excel file containing 81,520 entries of data. The file described the past shipments carried out by the company DHL LLP, and contained information such as the location of a shipment’s origin, origin cluster, destination, destination cluster, weight, volume, etc. The meaning of an origin cluster in this context is a location where all shipments of close proximity must gather in order to be shipped to another location as a group, which would thereby be called the destination cluster. From the destination cluster, the shipments would then be distributed to their nearby, individual destinations.

The purpose of this project was for us to answer the 6 questions given to us in adequate detail and thereby thoroughly analyze the data. However, the 6 questions given to us can be easily split into 2 categories. The first half (questions 1 to 3), focus more on forming an analysis from an econometrics perspective including conducting tests on potentially significant endogenous variables, conducting forecasts, etc. The second half (questions 4 to 6), are more focused on the ways this data can be analyzed and categorized using the algorithms found in Operations Research, such as clustering algorithms. Therefore, this report will be split into two major sections: an analysis from an econometrics perspective, and an analysis from an operations research perspective. Furthermore, in each of these major sections the report will be further split into question numbers. Once the report finishes explaining the analysis, it will then summarize everything in a concise business report consisting of 2 pages.

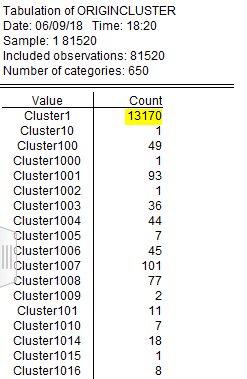
Part 1 (Econometric Analysis):

Question 1)

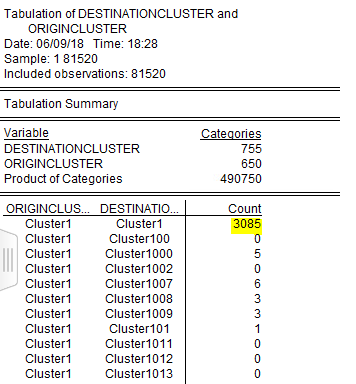
In this question, we are asked to pre-process the data so that we are left only with the data we need. This will help ensure that the tests we use later on yield efficient and accurate results.

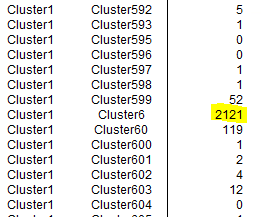
For this part of our analysis, the data we needed was a small part of the entire data set. In order to identify this part of the data, we had to first identify the most frequent origin cluster. This was done in Eviews using the One-Way Tabulation option (look at the Appendix: Annex A1 for the exact steps). The result of this was a long table showing the counts of each of the origin clusters. Once we had this, all we had to do was find the cluster with the highest count, as shown in Figure 1 below.

As can be seen from the figure, the cluster with the highest count was Cluster 1, with a count of 13,170 shipments. From the shipments which were described to have this origin cluster, we were then asked to find the ones with the first and second most frequent destination clusters. This was done using the N-Way Tabulation option, a process very similar to the first, on Eviews(steps found within Appendix: Annex A2). Within the group of shipments from Origin Cluster 1, we then found the 2 destination clusters with the highest counts. This can be seen in Figure 2.



*Figure 1: A screen-shot of the Cluster with highest count on Eviews*

**



*Figure 2: Two screen-shots of the Destination Clusters with highest count on Eviews*

As such we obtained 2 sets of shipments: one for origin cluster and destination cluster pair (1,1) and the other for the pair (1,6). From here on this report will refer to these two sets as Lane 1 and Lane 2 respectively.

It was clear from questions 2 and 3 that the entries Pick-Up Dates, Number of Shipments, Gross Volume, and Gross Weight from these two lanes were the only data we would need for the rest of our analysis. Therefore, using Excel, we discarded all the unnecessary data, leaving only the relevant entries for the shipments in these 2 groups. Furthermore, we divided that data into two separate sheets, one for Lane 1, and the other for Lane 2.

We then opened a new workfile in Eviews and imported the data from these 2 pages, leading to the creation of 9 time series: the Number of Shipments, Gross Volume, and Gross Weight in Lane 1, the same categories in Lane 2, the Pick-Up Dates, C (meaning the constant in a regression analysis), and Resid (meaning the residual over time). However, we still had to aggregate this data into a 5 days a week frequency as there were no shipments on weekends to begin with. To do this we created a new page with the frequency “5 days a week” and inserted our 6 series in this new page, thus helping create a concise and specific data set of aggregated time series (see Appendix: Annex A3 for steps).

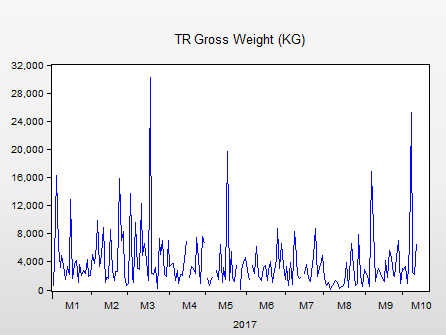
Question 2)

In this question, we are asked to analyze the time series we created in Question 1 using multiple tests for:

1. Determining the significance of dummy variables representing the days of the week
2. Determining the levels of autocorrelation in the series, and the methods to decrease them
3. Determining the presence of heteroskedasticity in the series and the methods to compensate for it

The goal of these tests was to find out which variables and coefficients to include when using our estimated equations to forecast future values of our 6 time series. However, before we could begin our tests, we first had to consider whether we would prefer to take the variables in levels, in log-levels, or in growth-rates. We chose to leave the variables as levels, primarily for 3 reasons: widely-distributed skewness, no permanent visibility of autocorrelation, and a convenient interpretation.

Theoretically, a logarithmic transformation is useful in mainly two instances: if what we are looking for is a trend in the elasticity of the data, or if the data happens to positively skewed. However, looking at our 6 time series, it was clear that the different variables hada wide range of skewness (0.0473, 12.1787), therefore no conclusion could be made for all of them. As for interpretation, since what we were looking at were time-series of physical quantities such as units shipped and their weights and volumes, information about changes in actual levels over time was more conclusive than percentages or elasticities.

As for a linear transformation using growth rates, looking at the graphs of the variables over time, there were no consistent indicators of autocorrelation. Therefore it was better to use levels instead. Also, for interpretations, it was more useful to see changes in levels and interpret the growth rates from there, rather than see growth rates directly. Figure 3 showsagraphof the gross weight in lane 1 over time to demonstrate the series’ lack of autocorrelation.

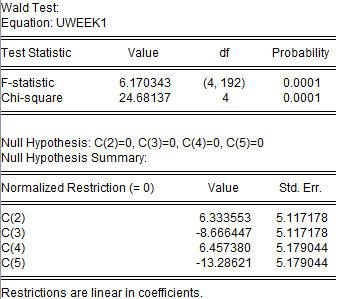
*Figure 3: A screen shot of the graph of Gross Weight (Lane 1) on Eviews*

Once we had decided to leave our variables as levels, it was time to start testing them. An important thing to note though is that, since there were multiple ways to test for autocorrelation and heteroskedasticity, every time there were different conclusions based on different tests, we chose to follow the test indicating the inclusion of more variables. This was because we were aware that the inclusion of irrelevant variables only leads to an increased variance in forecasts, but the exclusion of relevant variables can lead to both bias and inconsistency. Out of these, the latter seemed to be the worse choice so we chose to include variables as much as possible.

To test the significance of dummy variables representing different days of the week, we first had to estimate their values of their coefficients for each series using regression analysis. We did this onEviews using the Estimate Equation option (steps in Appendix: Annex A4). To clarify, the time series we were interested in were those describing the number of shipment units, the gross volume, and the gross weight, for both Lane 1 and Lane 2. Also, if we were to write the equation being regressed on paper, it would be written as.

Once we had the coefficients, we then used the Wald Test option to test the Null Hypothesis (steps in the Appendix: Annex A5) for each of our series’ dummy coefficients. The results we found can be summarized in the 6 figures below.

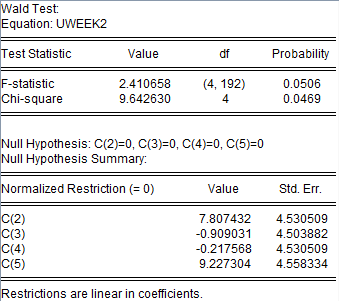
1. Number of Shipment Units (Lane 1): U(t) = 69.0811 + 6.3336\*(@WEEKDAY=2) - 8.6665\*(@WEEKDAY=3) + 6.4574\*(@WEEKDAY=4) - 13.2862\*(@WEEKDAY=5)



*Figure 4: A screen shot of the table showing coefficients for the days of the week dummy variables and the results of a Wald Test testing for no significance (in the series Number of Shipment Units in Lane 1)*

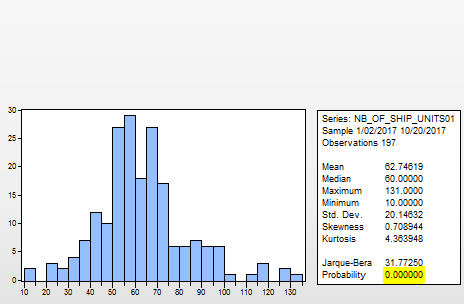
Looking at the p-values of both the F-statistic and Chi-square in this table, it can be concluded that the null of no seasonality should be rejected at 5% significance level (0.0001 < 0.05). Therefore, the days of the week do have a significant effect on the number of shipment units in Lane 1.

2. Number of Shipment Units (Lane 2): U(t) = 59.5676 + 7.8074\*(@WEEKDAY=2) - 0.9090\*(@WEEKDAY=3) - 0.2176\*(@WEEKDAY=4) + 9.2273\*(@WEEKDAY=5)



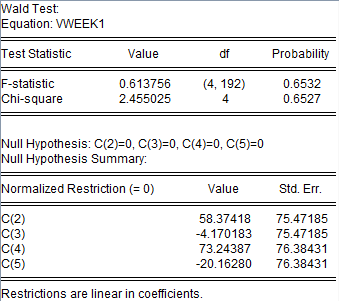
*Figure 5: A screen shot of the table showing coefficients for the days of the week dummy variables and the results of a Wald Test testing for no significance (in the series Number of Shipment Units in Lane 2)*

Since the F-statistic and Chi-square statistic in this table result in different conclusions based on a 5% significance level, we will have to test the series for normality and linearity. Should the series not accept the null hypothesis for either of these tests, we will use the result from the asymptotic chi-square distribution. If not, we will use the F-statistic.

For this series, according to the Jarque-Bera test, we should reject the null hypothesis of normality (check the Appendix: Annex A6 for steps). This can be seen in Figure 6 below. Therefore, in the table above, we will accept the chi-square statistic and therefore cannot accept the null hypothesis of the Wald test. As a result, the dummy coefficients can be considered significant for the Number of Shipment Units in Lane 2.

*Figure 6 : A screen shot showing the results of the JarqueBera test (in the series Number of Shipment Units in Lane 2)*

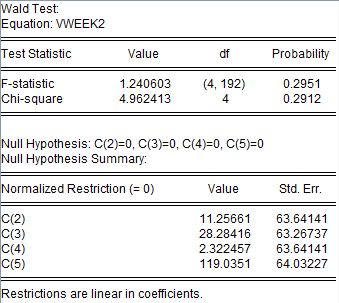
3. Gross Volume (Lane 1) : V(t) = 57.2526 + 58.3742\*(@WEEKDAY=2) - 4.1702\*(@WEEKDAY=3) + 73.2439\*(@WEEKDAY=4) - 20.1628\*(@WEEKDAY=5)



*Figure 7: A screen shot of the table showing coefficients for the days of the week dummy variables and the results of a Wald Test testing for no significance (in the series Gross Volume in Lane 1)*

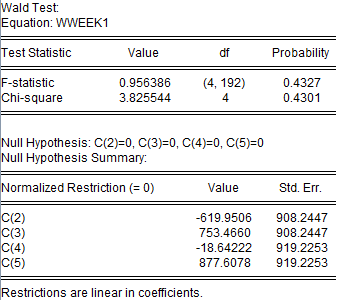
Based on the statistics in this table, we conclude that the null hypothesis should not be rejected at a 5% significance level (0.65>0.05). Weekdays do not have a significant effect on gross volume in Lane 1.

4. Gross Volume (Lane 2) : V(t) = 80.4066 + 11.2566\*(@WEEKDAY=2) + 28.2842\*(@WEEKDAY=3) + 2.3225\*(@WEEKDAY=4) + 119.0351\*(@WEEKDAY=5)



*Figure 8: A screen shot of the table showing coefficients for the days of the week dummy variables and the results of a Wald Test testing for no significance (in the series Gross Volume in Lane 2)*

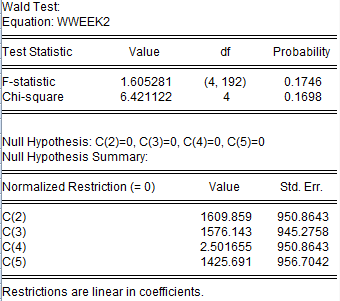
The statistics in this table indicate that we should not reject the null hypothesis at a 5% significance level (0.29>0.05) and conclude that the days of the week do not have a significant effect on the gross volume in Lane 2.

5. Gross Weight (Lane 1): W(t) = 3663.4284 - 619.9506\*(@WEEKDAY=2) + 753.4660\*(@WEEKDAY=3) - 18.6422\*(@WEEKDAY=4) + 877.6078\*(@WEEKDAY=5)

*Figure 9: A screen shot of the table showing coefficients for the days of the week dummy variables and the results of a Wald Test testing for no significance (in the series Gross Weight in Lane 1)*

Both the statistics in this table indicate that we should not reject the null hypothesis at a 5% significance level (0.43>0.05). Therefore, we conclude that the days of the week do not have a significant impact on the gross weight of shipments in Lane 1.

6. Gross Weight (Lane 2): 5139.2846 + 1609.8587\*(@WEEKDAY=2) + 1576.1432\*(@WEEKDAY=3) + 2.5017\*(@WEEKDAY=4) + 1425.6905\*(@WEEKDAY=5)



*Figure 10: A screen shot of the table showing coefficients for the days of the week dummy variables and the results of a Wald Test testing for no significance (in the series Gross Weight in Lane 2)*

Both the probabilities of the statistics in this table are larger than 0.05. As a result, we can conclude that, at a 5% significance level, the null hypothesis should not be rejected and the days of the week do not have a significant effect on the gross weight in Lane 2.

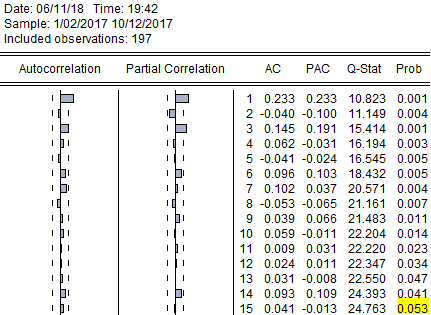
Based on the findings above, we concluded that the only series to have been significantly affected by the days of the week were the Number of Shipment Units in both lanes. Therefore, when forecasting, only they would include the coefficients of the days of the week dummy variables.

Next, we had to test the 6 series for autocorrelation and add a suitable number of lag variables to reduce the autocorrelation if it was significant. However, there are many ways to test for autocorrelation: by looking at the Durbin Watson Test, the LM Test, the Q statistics, ACF and PACF, and lastly by estimating the VAR and looking at corresponding lag specifications (steps to all these tests can be found in the Appendix: Annex A7-A10). It should be noted that the lag specification tests found when estimating the VAR are not meant to solely test autocorrelation. However, since their goal is to provide the optimal model, they can help to reduce autocorrelation to an insignificant level.

Since discussing the results of all these tests for each of the 6 series would probably lead to redundant results, we will only discuss the tests and series which yielded a significant amount of autocorrelation, as well as the number of lags needed to reduce such autocorrelation.

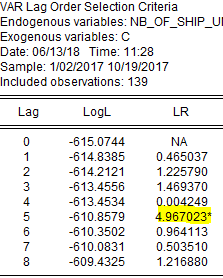
1. Number of Shipment Units (Lane 1): U(t) = 46.3948 + 0.3162\*U(t-1) - 0.1287\*U(t-2) + 0.1576\*U(t-3) - 0.0046\*U(t-4) - 0.1010\*U(t-5) + 0.1251\*U(t-6) + 0.1168\*U(t-7) - 0.1118\*U(t-8) + 0.0349\*U(t-9) + 0.0078\*U(t-10) - 0.0100\*U(t-11) + 0.0798\*U(t-12) - 0.1241\*U(t-13) + 0.1038\*U(t-14) - 0.0467\*U(t-15) + 1.4942\*(@WEEKDAY=2) - 21.4596\*(@WEEKDAY=3) + 1.0064\*(@WEEKDAY=4) - 20.0365\*(@WEEKDAY=5)

The Number of Shipment Units series in Lane 1 yielded a significant amount of autocorrelation and needed 15 lags to reduce it. This can be seen from the probability of the Q-statistics in Figure 10 (the null of the Q-statistic is no autocorrelation so a high probability is an indicator of insignificant autocorrelation). The equation above shows the results of regressing the series while taking account of the dummy variables for the days of the weeks and the 15 lags.



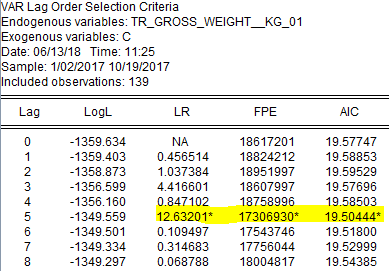
*Figure 11: A screen shot of the table showing the Q-Statistics and their corresponding p-values (in the series Number of Shipment Units in Lane 1)*

2. Number of Shipment Units (Lane 2): 56.8348 - 0.03671\*U(t-1) - 0.0837\*U(t- 2) - 0.0340U(t-3) + 0.0232\*U(t-4) + 0.2077\*U(t-5) + 5.7853\*(@WEEKDAY=2) - 2.1201\*(@WEEKDAY=3) - 1.7219\*(@WEEKDAY=4) + 4.9803\*(@WEEKDAY=5)

 According to the VAR estimation and lag specification, using a likelihood ratio test at the 5% significance level, the Number of Shipment Units in Lane 2 optimally needs 5 lag variables to decrease it’s autocorrelation to a non-significant level. This can be seen in Figure 11 below. The equation above shows the results of regressing the equation taking these new lag variables into account.

*Figure 12: A screen shot of the table showing the Likelihood Ratio test results and the ideal number of lag variables (in the series Number of Shipment Units in Lane 2)*

3. Gross Weight (Lane 2): W(t) = 3908.2041 + 0.0918\*W(t-1) - 0.1536\*W(t-2) + 0.2163\*W(t-3) - 0.0568\*W(t-4) + 0.2710\*W(t-5)

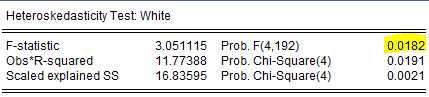
 According to the Q-statistics, although the autocorrelation is initially at an insignificant level, it quickly increases as the lags increase. Therefore, looking at the lag specifications for the VAR estimation of this series (likelihood ratio, final prediction error, and akaike information criterion), we find that 5 lags is the optimal number. This can be seen in Figure 12. The equation above shows the corresponding values we obtained through regression analysis.

*Figure 13: A screen shot of the table showing the ideal number of lag variables (in the series Gross Weight in Lane 2)*

Lastly, we had to test for heteroskedasticity. Within Eviews there is a list of tests for this, including: the Breusch-Pagan-Godfrey test, the Harvey test, the Glesjer test, the ARCH test, and the White test (the steps for these are in the Appendix: Annex A11). However, since the White test can be used for most models, with no restrictions, unlike many of the other tests, we will focus our analysis on the White test. Below we will state any series found to have heteroskedasticity according to the White Test. If we find a series to be heteroskedastic, we will then estimate its coefficients again, using the Robust Least Squares method instead (steps are in the Appendix: Annex A12).

1. The Number of Shipment Units (Lane 2): U(t) = 57.0012 - 0.0218\*U(t-1) - 0.0357U(t- 2) + 0.0010\*U(t-3) + 0.0438\*U(t-4) + 0.0439U(t-5) + 2.2656\*(@WEEKDAY=2) - 1.3117\*(@WEEKDAY=3) + 0.9816\*(@WEEKDAY=4) + 4.9891\*(@WEEKDAY=5)

Figure 13 below shows that the p-value for the F statistic of the White Test is below 0.05 for this series, meaning that there is heteroskedacity. This means that we have to run the regression for this series again, this time using robust least square, instead of least squares. The result of this is the equation above.



*Figure 14: A screen shot of the results of the White test for heteroskedasticity (in the series Number of Shipment Units in Lane 2)*

Question 3)

The last question of this section asks us to use the results of the tests in question 2 to find forecasts for the 6 series for a week after the last entry in the given data. For 3 of the series this means a forecast using a combination of dummy variables and extra lag variables. However, the other 3 series, Gross Volume in both lanes, and Gross Weight in Lane 1, required no lag variables, were homoskedastic, and had insignificant dummy variables. This means that the equations to forecast them only consist of a constant variable each. While this is what the tests indicate, it is highly unlikely for the forecasts to be the same every day of the week. Therefore, in order to avoid bias and inconsistency, we chose to include the dummy variables for the forecasts of these series, regardless of the results found during question 2. As a result, we came up with the table of forecasts below (steps are in the Appendix: Annex A13).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Date | Number of Shipments (1) | Number of Shipments (2) | Gross Volume (1) | Gross Volume (2) | Gross Weight (1) | Gross Weight (2) |
| 10/16/2017 | 76.1074 | 59.6025 | 57.2526 | 80.4066 | 3663.428 | 5695.253 |
| 10/17/2017 | 76.3347 | 66.9714 | 115.6268 | 91.6632 | 3043.478 | 5400.956 |
| 10/18/2017 | 51.3366 | 60.3709 | 53.0824 | 108.6907 | 4416.894 | 5898.426 |
| 10/19/2017 | 72.4791 | 61.3272 | 130.4965 | 82.7290 | 3644.786 | 5678.467 |
| 10/20/2017 | 59.7489 | 67.2448 | 37.0898 | 199.4417 | 4541.036 | 6020.416 |

*Table 1: A table of forecasts for the 6 series over the first week after the data ends.*

Part 2 (Operations Research):

In this section, we are tasked to analyze and evaluate the performance of the current clustering in the DHL data. To reduce computational time, we aggregated the four main metrics in the clustering, namely weight, volume, and number of shipments, for a given origin-destination pair over the given time period. To carry out this pre-processing step, we had to neglect pick up dates. After pre-processing, we reduced the dataset to 4011 instances.

Question 4)

In this question, we are asked to develop a flexible structure that allowsus to analyze the current clustering. To identify such a structure, our first step was feature selection; we identified the most important variables that are of relevance to our clustering analysis. After careful consideration, we chose the following variables:

* Origin Country
* Origin City
* OriginCluster
* OriginClusterLat
* OriginClusterLong
* OriginLat
* OriginLong
* Dest Country
* Dest City
* DestinationCluster
* DestinationClusterLat
* DestinationClusterLong
* DestLat DestLong
* TR Gross Weight (KG)
* Nb of Ship Units TR Gross Volume (M3)

In determining the classes to be created in Java, we noted that several highly related variables (in terms of their definitions) could be grouped together in the same class. For example, Origin Country, Origin City, OriginLat, and OriginLongall describe the geographic location (of origin) of a certain shipment; hence, a class ‘Location’ with a country, city, latitude and longitude as private variables would be appropriate.

By the same token, we identified groups of variables that summarize the aggregated dataset, and created corresponding classes for them. The classes that we created in Java are as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Class Name** | **(Private) Variables** | **Important Methods** | **Interpretation** |
| Shipment | **Metrics:**  weight  quantity  volume  **Clusters:**  originCluster  destCluster  **Location:**  originLocation  destLocation  **Distance between location and cluster:**  originDistance  destDistance | Constructor  Getters  Setters | The class represents a particular shipment and its data.  The distance between a location and its respective cluster is calculated using the *Haversine formula*, which calculates the distance between two points on a sphere given their longitudes and latitudes.[[1]](#footnote-1) The Haversine formula is an approximate distance (in kilometres) between two specified locations on Earth. The Haversine distance is computed using the HaversineDistance class. |
| Cluster | ID  latitude  longitude | Constructor  Getters  Setters | The class represents a particular cluster, which can be either an origin cluster or a destination cluster. A Cluster object contains a cluster ID, latitude, and longitude. |
| Location | country  city  latitude  longitude | Constructor  Getters  Setters | The class represents a shipment location, which can be either an origin location (where the shipment departs from) or a destination location (where the shipment is headed). a Location object includes a country, city, latitude, and longitude. |
| ShipmentList | **Shipment dataset:**  shipmentList  **Lists of unique clusters:**  originClusters  destClusters  **Total weight:**  originWeights  destWeights  **Total distance between location and cluster:**  originDistances  destDistances  **Total volume:**  originVolumes  destVolumes  **Total quantity:**  originQuantity  destQuantity | Constructor  Getters  Setters  Average cluster  Top 3 clusters  Bottom 3 clusters  k-Means Clustering | This class summarizes the dataset. A ShipmentList object includes all shipments that have been read into our Java program, as well as the metrics associated with the currently applied clustering. These metrics are automatically updated each time a shipment is added to (or removed) from the shipment list.  In addition, this class contains the k-Means Clustering algorithm, which is important in answering question 5 and 6. |
| HaversineDistance | EARTH\_RADIUS  startLat (latitude of starting point)  startLong (longitude of starting point)  endLat (latitude of ending point)  endLong (longtitude of ending point | distance  haversin | This class consists of two methods, one of which calculates the Haversine distance between two points on a sphere (i.e the Earth), using the input variables stated.  This class is used in conjunction with other classes, such as the ShipmentList class and the Shipment class. |
| Clustering |  | main  readData | This class includes the actual implementation of any algorithms created (i.e k-Means Clustering) as well as the evaluation of the original clustering. |

*Table 2: A table of Java classes and their associated methods*

The Haversine formula is:

where

* is the distance between the two points along a great circle of the sphere,
* is the radius of the sphere,
* and are the latitude of point 1 and latitude of point 2 respectively, in radians, and
* and are the longitude of point 1 and longitude of point 2 respectively, in radians.

To derive the aforementioned performance measures, such as the total distance of all shipments from the origin location to the origin cluster, we increment the appropriate metric in our program, with each shipment added. For example, after obtaining the list of unique origin clusters in the current clustering, for each shipment whose origin cluster is, for example, Cluster 1, we incremented the performance metrics for Cluster 1 accordingly. In this way, we arrived at the total values for the performance metrics for each cluster.

The performance measures for the initial clustering are as follows:

|  |  |
| --- | --- |
| **Origin** | **Destination** |
| Total distance from locations to clusters: 5428.961960619172 | Total distance from locations to clusters: 4874.904502514321 |
| Average cluster wrt total distance: Cluster902   * Value of average cluster: 8.313240107222832 * Average value: 8.390976755207383   Average cluster wrt total weight: Cluster76   * Value of average cluster: 76387.0 * Average value: 72602.05645131374   Average cluster wrt total volume: Cluster550   * Value of average cluster: 1706.556 * Average value: 1638.9395069551786   Average cluster wrt total quantity: Cluster996   * Value of average cluster: 364 * Average value: 364.548686244204 | Average cluster wrt total distance: Cluster684   * Value of average cluster: 6.123313380001216 * Average value: 6.298326230638661   Average cluster wrt total weight: Cluster547   * Value of average cluster: 61364.0 * Average value: 60689.31592248065   Average cluster wrt total volume: Cluster10   * Value of average cluster: 1464.9469999999994 * Average value: 1370.0179082687332   Average cluster wrt total quantity: Cluster547   * Value of average cluster: 310 * Average value: 304.7325581395349 |
| Top 3 total origin distances:   1. Cluster2 (3342.5071723061615) 2. Cluster8 (503.9226266971474) 3. Cluster3 (278.97624112065745)   Top 3 total origin weights:   1. Cluster2 (6262006.481000002) 2. Cluster902 (5195074.104) 3. Cluster15 (3061947.3500000006)   Top 3 total origin volumes:   1. Cluster443 (122109.77900000085) 2. Cluster19 (106330.67500000008) 3. Cluster15 (104512.2429999997)   Top 3 total origin quantities:   1. Cluster2 (53327) 2. Cluster1 (24848) 3. Cluster15 (11964) | Top 3 total destination distances:   1. Cluster2 (2328.4887185208154) 2. Cluster8 (667.1922833921053) 3. Cluster441 (199.28990420469611)   Top 3 total destination weights:   1. Cluster2 (7966273.728) 2. Cluster906 (5519990.334) 3. Cluster546 (3219875.7010000004)   Top 3 total destination volumes:   1. Cluster2 (244201.52699999948) 2. Cluster443 (149982.4760000009) 3. Cluster19 (106808.47200000002)   Top 3 total destination quantities:   1. Cluster2 (56656) 2. Cluster1 (39947) 3. Cluster546 (22109) |
| Bottom 3 total origin distances:   1. Cluster827 (0.0) 2. Cluster637 (0.0) 3. Cluster596 (0.0)   Bottom 3 total origin weights:   1. Cluster606 (0.22) 2. Cluster304 (0.5) 3. Cluster526 (0.5)   Bottom 3 total origin volumes:   1. Cluster886 (0.0) 2. Cluster526 (0.0) 3. Cluster195 (0.001)   Bottom 3 total origin quantities:   1. Cluster719 (1) 2. Cluster601 (1) 3. Cluster607 (1) | Bottom 3 total destination distances:   1. Cluster19 (0.0) 2. Cluster15 (0.0) 3. Cluster17 (0.0)   Bottom 3 total destination weights:   1. Cluster719 (0.25) 2. Cluster586 (0.5) 3. Cluster1024 (0.5)   Bottom 3 total destination volumes:   1. Cluster586 (0.0) 2. Cluster856 (0.001) 3. Cluster771 (0.001)   Bottom 3 total destination quantities:   1. Cluster748 (1) 2. Cluster858 (1) 3. Cluster812 (1) |

*Table 3: Performance measures for the initial clustering*

In the top/bottom 3 lists, the values in the round brackets indicate the values of the respective metrics for the clusters in question.

Question 5)

In this question, we are asked to develop and implement an algorithm that can find a different assignment of the locations toclusters, so that the total distance of all shipments from the location to the assigned cluster(origin and destination) is small (but not necessarily minimum). The algorithm should take asinput the number of clusters that can be used.

We implemented the -Means Clustering algorithm, which is named ‘kMeans’ in the ShipmentList class.kMeans partitionsthe 4011 shipments into clusters in which each shipment belongs to the cluster that is closest to it, based on the Haversine distance between them. The algorithm hence takes into account the latitudes and longitudes of the (origin and destination) locations of the shipments, as well as those for the (origin and destination) clusters that are constantly updated at each iteration. The algorithm stops when the cluster assignment no longer changes.

More specifically, consider the assignment of origin clusters. The first step of a general k-Means Clustering implementation is randomly assign the data instances to k clusters. In our case, to reduce the computational complexity of randomizing the cluster ID of the shipments, while still achieving the same intended result, we computed the number of shipments required per cluster: . Then, we sequentially assigned the shipments to the clusters; for example, the first shipments belong to Cluster 1, while the next shipments belong to Cluster 2, and so on. In the next step, we derived the so-called ‘cluster centroids’, which are basically representative clusters for the current cluster assignment. These centroids are still instances of the Cluster class, so they each have an ID (which is the same as the ID of their respective clusters), latitude, and longitude. The latitudes and longitudes of the centroid of a cluster were computed as the average latitudes and longitudes, respectively, of all the shipments that belong the said cluster. The next step is the reassignment step: we reassigned the shipments to the centroids that are closest in Haversine distance to them. The Haversine distances were computed using the latitudes and longitudes of the shipment locations and the centroids respectively. The algorithm determined the minimum of such distances, and assigned the shipment to the corresponding cluster. The recursion step follows, in which we recursively computed new centroid locations, and reassigned the shipments. We identified the stopping point as the iteration in which there is no changes to the cluster assignment. The algorithm serves to minimize location-cluster distances.

We ran the -Means Clustering algorithm using several different values of . The results below are for the case when

|  |  |
| --- | --- |
| **Origin**  7 iterations | **Destination**  14 iterations |
| Total distance from locations to clusters: 720861.7421654129 | Total distance from locations to clusters: 911800.9561343524 |
| Average cluster wrt total distance: Cluster5   * Value of average cluster: 176770.9821887278 * Average value: 144172.3484330838   Average cluster wrt total weight: Cluster1   * Value of average cluster: 8008764.550999998 * Average value: 9394706.104799997   Average cluster wrt total volume: Cluster2   * Value of average cluster: 214631.3479999999 * Average value: 212078.77220000018   Average cluster wrt total quantity: Cluster5   * Value of average cluster: 48128 * Average value: 47172.6 | Average cluster wrt total distance: Cluster4   * Value of average cluster: 124199.13636514782 * Average value: 182360.19122687136   Average cluster wrt total weight: Cluster5   * Value of average cluster: 9055716.059999991 * Average value: 9394706.104799997   Average cluster wrt total volume: Cluster3   * Value of average cluster: 271218.9849999994 * Average value: 212078.77219999986   Average cluster wrt total quantity: Cluster5   * Value of average cluster: 52412 * Average value: 47172.6 |
| Top 3 total origin distances:   1. Cluster2 (317677.7193741632) 2. Cluster5 (176770.9821887278) 3. Cluster3 (95935.87426902015)   Top 3 total origin weights:   1. Cluster2 (1.3557637145999987E7) 2. Cluster3 (1.1492750517000003E7) 3. Cluster5 (1.1262108227000002E7)   Top 3 total origin volumes:   1. Cluster5 (304956.18100000056) 2. Cluster3 (227236.82100000008) 3. Cluster2 (214631.3479999999)   Top 3 total origin quantities:   1. Cluster1 (65989) 2. Cluster3 (53603) 3. Cluster2 (53362) | Top 3 total destination distances:   1. Cluster3 (379173.010131012) 2. Cluster5 (254185.10479771908) 3. Cluster4 (124199.13636514782)   Top 3 total destination weights:   1. Cluster3 (1.2176985528999997E7) 2. Cluster1 (1.0318968314E7) 3. Cluster4 (1.0224966147999993E7)   Top 3 total destination volumes:   1. Cluster4 (289872.5709999992) 2. Cluster5 (273259.75800000085) 3. Cluster3 (271218.9849999994)   Top 3 total destination quantities:   1. Cluster3 (77436) 2. Cluster4 (65699) 3. Cluster5 (52412) |
| Bottom 3 total origin distances:   1. Cluster4 (59900.50136906431) 2. Cluster1 (70576.66496444347) 3. Cluster3 (95935.87426902015)   Bottom 3 total origin weights:   1. Cluster4 (2652270.0829999996) 2. Cluster1 (8008764.550999998) 3. Cluster5 (1.1262108227000002E7)   Bottom 3 total origin volumes:   1. Cluster4 (117644.48600000028) 2. Cluster1 (195925.02499999997) 3. Cluster2 (214631.3479999999)   Bottom 3 total origin quantities:   1. Cluster4 (14781) 2. Cluster5 (48128) 3. Cluster2 (53362) | Bottom 3 total destination distances:   1. Cluster1 (38239.0301350532) 2. Cluster2 (116004.67470542467) 3. Cluster4 (124199.13636514782)   Bottom 3 total destination weights:   1. Cluster2 (5196894.473000001) 2. Cluster5 (9055716.059999991) 3. Cluster4 (1.0224966147999993E7)   Bottom 3 total destination volumes:   1. Cluster1 (94958.93699999989) 2. Cluster2 (131083.60999999996) 3. Cluster3 (271218.9849999994)   Bottom 3 total destination quantities:   1. Cluster1 (10778) 2. Cluster2 (29538) 3. Cluster5 (52412) |

*Table 3: Performance measures for the -Means clustering*

The results for other cases of can be found in the appendix (Annex B4).

In general, the total distances evaluated using -Means Clustering are higher than those derived from the initial clustering, which suggests that the -Means Clustering algorithm we created is not optimal. Nonetheless, the algorithm is a solution to clustering the shipments based on their locations.

Question 6)

In this question, we are asked to suggest alternative ways of clustering, based on the specific profiles of the shipments. For example, we could consider the direction of the shipping flow, or origin-destination distances.

Given the data available, we noted that there are four origin (as well as destination) countries: Spain, France, Germany, and United Kingdom (UK). Hence, instead of clustering the entire dataset based on the locations of all shipments relative to the others, we segmented the data based on countries. Specifically, we derived 5 subsets of data:

* Set 1: All shipments whose origin city and destination city are both Germany. (938 shipments)
* Set 2: All shipments whose origin city and destination city are both France. (1053 shipments)
* Set 3: All shipments whose origin city and destination city are both Spain. (414 shipments)
* Set 4: All shipments whose origin city and destination city are both UK. (57 shipments)
* Set 5: All shipments whose origin city and destination city differ. (1549 shipments)

Then, we performed -Means Clustering on these data subsets individually. The input number of cluster is taken as a proportion of the total number of clusters had we performed -Means Clustering on the entire dataset. For example, since the first data subset (Germany) contain 938out of the total 4011 shipments, suppose we take the total number of clusters as the number of unique clusters in the initial clustering (i.e 1027 clusters), our input for the algorithm implemented on dataset subset 1 would be clusters. Doing the same for the rest of the data subsets, we thus identified clusters that specifically cover the geographical connection between the origin and destination locations of a shipment, as well as indirectly addressing the distance between them. Indeed, shipments that depart from and to the same country should have clusters dedicated to them.

The results of the -Means Clustering on the UK data subset, with , are as follows:

|  |  |
| --- | --- |
| **Origin**  10 iterations | **Destination**  3 iterations |
| Total distance from locations to clusters: 1013.7604796507624 | Total distance from locations to clusters: 844.954111198685 |
| Average cluster wrt total distance: Cluster3   * Value of average cluster: 185.06928577639258 * Average value: 202.75209593015242   Average cluster wrt total weightCluster15   * Value of average cluster: 975966.5 * Average value: 1712652.7748000002   Average cluster wrt total volume: Cluster7   * Value of average cluster: 17992.71999999995 * Average value: 14660.814599999987   Average cluster wrt total quantity: Cluster3   * Value of average cluster: 306 * Average value: 320.0 | Average cluster wrt total distance: Cluster11   * Value of average cluster: 66.62841125738865 * Average value: 93.88379013318729   Average cluster wrt total weight: Cluster2   * Value of average cluster: 1039348.19 * Average value: 951473.7637777776   Average cluster wrt total volume: Cluster2   * Value of average cluster: 10680.853000000003 * Average value: 8144.896999999991   Average cluster wrt total quantity: Cluster2   * Value of average cluster: 214 * Average value: 177.77777777777777 |
| Top 3 total origin distances:   1. Cluster15 (592.2530950676421) 2. Cluster3 (185.06928577639258) 3. Cluster7 (101.14592017802013)   Top 3 total origin weights:   1. Cluster3 (4785804.414000001) 2. Cluster7 (2720900.16) 3. Cluster15 (975966.5)   Top 3 total origin volumes:   1. Cluster3 (45266.255999999965) 2. Cluster7 (17992.71999999995) 3. Cluster15 (9382.143000000007)   Top 3 total origin quantities:   1. Cluster7 (1127) 2. Cluster3 (306) 3. Cluster15 (84) | Top 3 total destination distances:   1. Cluster12 (211.6153983450417) 2. Cluster1 (157.85900598407815) 3. Cluster8 (129.561669605354)   Top 3 total destination weights:   1. Cluster14 (4931863.864) 2. Cluster12 (1962420.2) 3. Cluster2 (1039348.19)   Top 3 total destination volumes:   1. Cluster14 (46199.12799999996) 2. Cluster12 (11227.787999999964) 3. Cluster2 (10680.853000000003)   Top 3 total destination quantities:   1. Cluster12 (648) 2. Cluster14 (346) 3. Cluster1 (315) |
| Bottom 3 total origin distances:   1. Cluster13 (40.34456547603622) 2. Cluster1 (94.94761315267118) 3. Cluster7 (101.14592017802013)   Bottom 3 total origin weights:   1. Cluster1 (67.0) 2. Cluster13 (80525.8) 3. Cluster15 (975966.5)   Bottom 3 total origin volumes:   1. Cluster1 (1.371) 2. Cluster13 (661.5830000000002) 3. Cluster15 (9382.143000000007)   Bottom 3 total origin quantities:   1. Cluster1 (4) 2. Cluster13 (79) 3. Cluster15 (84) | Bottom 3 total destination distances:   1. Cluster5 (0.0) 2. Cluster6 (39.51043977864824) 3. Cluster14 (50.57381557790219)   Bottom 3 total destination weights:   1. Cluster5 (49.0) 2. Cluster10 (889.5) 3. Cluster11 (1183.87)   Bottom 3 total destination volumes:   1. Cluster5 (0.254) 2. Cluster10 (6.601000000000001) 3. Cluster11 (15.677999999999999)   Bottom 3 total destination quantities:   1. Cluster5 (1) 2. Cluster10 (17) 3. Cluster11 (17) |

*Table 3: Performance measures for the -Means clustering for UK*

The results of the algorithm for other data subsets can be found in Annex B5.

In general, there are improvements in the total distances as compared to applying -Means Clustering on the entire dataset, these metrics are still higher than those derived from the initial clustering. Hence, this suggests that this alternative is yet to be optimal, and further consideration is warranted.

Business Summary

Part 1 (Econometric Analysis):

Based on the data given by the company DHL LLP, it can be said that being able to accurately forecast future shipments and their specifics is an extremely important key to the company’s future success. This is because by doing so they will be able to prepare well and avoid unnecessary costs such as hiring last minute vehicles. They will also minimize the number of customers who may be dissatisfied due to a late or returned shipment.

Therefore, in order to create these accurate forecasts, the scientific report uses Econometric methods to test the given data and recognize the factors which would affect future forecasts of the number of shipments, and their gross volumes and weights. These factors are then combined to create equations that can be used to model the specifics of shipments over time. Although the report focuses upon the two most popular routes of Shipments: the ones within Hamburg, Germany, and those going from Hamburg, Germany to Brive La Gaillarde, France, the analysis used within the report could easily be expanded to include more.

The report first describes the procedures used to remove any unnecessary entries from the data file and prepare the remaining entries for the analysis to be done later on. This was done using the software Excel and means that any entries other than those describing the time, volume, weight, or number of shipped units, were removed. Furthermore, all weekend entries were removed as well since no shipments are picked up on the weekends.

The report then goes on to describe the various methods through which the data was analyzed, using a software called Eviews. These include testing whether the day of the week a shipment takes place has a significant effect on the specifics of the shipment, such as the gross volume of the shipment. Furthermore, the tests then go on to calculate whether a shipment’s specifics are related to those of past shipments. So, for example, whether the volume of today’s shipment depends on the shipments that have been taking place across the same route for the past month. If the specifics of a set of shipments are shown to be related those of past shipments, the equations used to model the data are then modified in order to take these relations into consideration. Lastly, the tests gauge whether the shipment’s specifics stay within certain ranges over time or whether they have no such boundaries. If the latter case is true, the equations used to model the data are modified once again to take this into account.

At the end of the first section, all the tests’ findings were used to create the following table of future forecasts (based on the last entry of the given data) for the shipments’ specifics.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Date | Number of Shipments (1) | Number of Shipments (2) | Gross Volume (1) | Gross Volume (2) | Gross Weight (1) | Gross Weight (2) |
| 10/16/2017 | 76.1074 | 59.6025 | 57.2526 | 80.4066 | 3663.428 | 5695.253 |
| 10/17/2017 | 76.3347 | 66.9714 | 115.6268 | 91.6632 | 3043.478 | 5400.956 |
| 10/18/2017 | 51.3366 | 60.3709 | 53.0824 | 108.6907 | 4416.894 | 5898.426 |
| 10/19/2017 | 72.4791 | 61.3272 | 130.4965 | 82.7290 | 3644.786 | 5678.467 |
| 10/20/2017 | 59.7489 | 67.2448 | 37.0898 | 199.4417 | 4541.036 | 6020.416 |

Part 2 (Operations Research):

In this part of the report, we analysed the performance of the current shipment process within Europe (Germany, Spain, France, and the UK). We calculated the total distance covered by the shipments, as well as their specifics such as total weight. We also projected possible locations where DHL can establish shipment centers (i.e places where shipments within close proximity are gathered to be sent or received in bulk), in order to minimize cost and the total distance covered by the shipments.

The algorithm that we used to locate ideal shipment centers can be of great importance to the company because it is flexible and scalable. By repeatedly using the algorithm while varying the allowed number of shipment centers under consideration, DHL can identify the optimal number of such centers as well as the ideal locations for them. Not only will this help them expand their current shipment network efficiently, they will also be able to assess the effectiveness of the current network. Furthermore, since the algorithm produces aggregate estimates of the specifics of the shipments at different locations, it helps the company obtain an overall view on their current and also hypothetical shipment networks.

The said algorithm, called -Means Clustering, first randomly allocates the available shipments to a prespecified number of groups. Then, it goes on to pinpoint the central location within the locations of the shipments that belong to the same group, by averaging the latitudes and longitudes of such locations. Such central locations then constitute the ideal arrangement of shipment centers. By repeatedly testing out different possible locations of shipment centers and thus optimizing the distance between the available shipments and such centers,the algorithm returns the final arrangement that it deems optimal.

The output that we obtained from using this algorithm, however, is still not as good as the preset arrangement, in terms of minimizing the total distances. This might be because the preset arrangement was created using more complicated models that consider other aspects of the shipments, such as pickup dates (which we have assumed to be irrelevant in our analysis). As such, the preset arrangement might have been created in a more holistic way. Nonetheless, our algorithm can be leveraged upon, such as by utilizing shipment segmentation. Specifically, we divided the shipments into groups based on their origin and destination countries, such that there are two types of shipments: shipments within the same countries, and shipments that cross borders. Then, we implemented the algorithm on these subsets, and arrived at the arrangement that is more specific to the geographical aspect of the available shipments. This modification demonstrates the flexibility and scalability of the algorithm.

An example of another algorithm that can be used besides -Means is Hierarchical Clustering. Hierarchical clustering involves creating groups that have a predetermined ordering from top to bottom. For example, all files and folders on the hard disk are organized in a hierarchy. This could be implemented on the available shipment data, to find a hierarchy within the shipments, such as one based on the distances they cover.

In conclusion, this analysis gives an overall evaluation of the performance of the company. Nonetheless, there are various ways to extend it, subject to the short-term and long-term priorities of DHL.

Appendix

**ANNEX A: ECONOMETRICS ANALYSIS**

Annex A1 (One-Way Tabulation):

1. Import the data from the given excel file to e-views
2. Click on the Origin Cluster series to open it
3. Use the One-way Tabulation option under Views to obtain a table of the counts for each cluster
4. Identify the Cluster with the most views (Cluster 1)

Annex A2 (N-Way Tabulation):

1. Click on the Origin and Destination Cluster
2. Click on the option One Window under Open Selected in the tab View
3. Click as a Group
4. Use the N-way Tabulation option under Views to obtain a table of the counts for each pair of clusters
5. Identify the 2 pairs with the highest frequencies

Annex A3 (Aggregating data on a 5 Days a Week Frequency):

1. Click on New Page at the bottom of the screen
2. Click on Specify by Frequency/Range
3. Click on the 5 Days a Week option under Frequency
4. Click to select all relevant series on the original page
5. Drag series to the new page

Annex A4 (Estimate Equation using Least Squares):

1. Click on the Estimate Equation option under the Quick tab
2. Type the formula “Nb\_of\_ship\_units\_ c @expand (@weekday, @dropfirst)” in the equation box
3. Save the resultant equation and name it
4. Repeat steps 1-3 for 5 other series but replace the name in of the variable in Step 2, based on the series being analyzed

Annex A5 (Wald Test):

1. Click on the equation including the series and coefficients you want to test
2. Click on the Coefficient Diagnostics option under View
3. Click on Wald Test-Coefficient Restrictions
4. Type in “c(2)=c(3)=c(4)=c(5)=0” and click OK
5. Analyze the results

Annex A6 (Jarque-Bera Test):

1. Click on the series you want to test
2. Click on the Descriptive Statistics and Tests option under the View Tab.
3. Click on the Histogram and Stats option
4. Read and analyze the probability of the Jarque-Bera test

Annex A7 (Durbin-Watson Test):

1. Click on the equation you want to test for autocorrelation
2. Click on the Estimation Output option under View
3. The Durbin-Watson statistic should be listed under the coefficients

Annex A8 (LM Test):

1. Click on the equation you want to test for autocorrelation
2. Click on the Residual Diagnostics option under the View tab
3. Click on the Serial Correlation LM Test option
4. Specify the number of lags you want to test autocorrelation with
5. Repeat while increasing/decreasing the number of lags depending on the outcome of Step 4

Annex A9 (Q-Statistic/ACF/PACF):

1. Click on the equation you want to test for autocorrelation
2. Click on the Residual Diagnostics option under the View tab
3. Click on the Correlogram-Q Statistics option
4. Specify the total number of lag variables you want to analyze
5. View and analyze the results shown in regards to ACF, PACF, and Q-statistics

Annex A10 (Estimate VAR/Likelihood Ratio Test):

1. Click on the Estimate VAR option under the Quick tab
2. Type in the endogenous variable you want to test for autocorrelation and click OK
3. In the resultant variable, click on the option Lag Structure under View
4. Click on the option Lag Length Criteria
5. Check to see which number of lags has the asterix next to it (is the optimum)

Annex A11 (White/Heteroskedasticity Test):

1. Click on the equation you want to test for heteroskedasticity
2. Click on the Residual Diagnostics option under the View tab
3. Click on the Heteroskedasticity tests option
4. Choose the test you want to use
5. Analyze the results accordingly

Annex A12 (Estimate Equation using Robust Least Squares):

1. Click on the Estimate Equation option under Quick
2. In the Method option, click ROBUSTLS
3. Write the equation you want to analyze again using Robust Errors, like “Nb\_of\_ship\_units\_ c @expand (@weekday, @dropfirst)”
4. Analyze the resultant coefficients

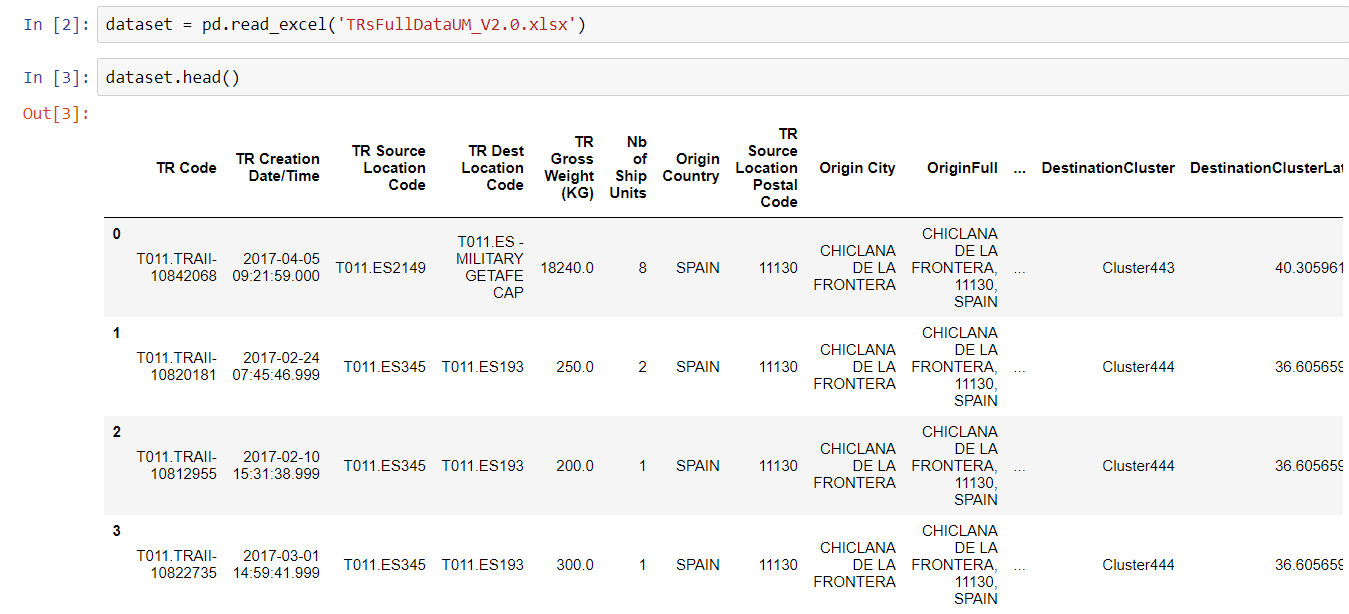
Annex A13 (Create Forecasts):

1. (If the dates you want to forecast using your equations are beyond your current data) Click on the Structure/Resize Current Page option under the Proc tab and include the extra dates
2. Click on the equation you want to forecast
3. Click on the forecast option at the top of the Equation window
4. Specify the dates you want to forecast (beware any missing values in the data as you may need these as lag variables in the forecast’s equations)

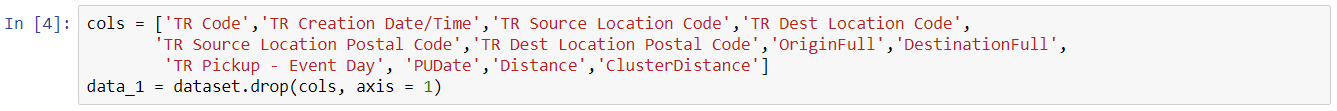
**ANNEX B: OPERATIONS RESEARCH**

Annex B1 (Data Aggregation):

1. Import the data from the given excel file to Python (Jupyter Notebook) using the Pandas package



1. Remove redundant variables.



1. Group the data entries by location-cluster pairs.
2. Export the data to CSV (to be read in Java)



Annex B2 (Method to read the shipment data into Java):

**public static void** readData(ShipmentListshipmentList, String filePath, String condition) {  
BufferedReader reader = **null**;  
  
**try** {  
 reader = **new** BufferedReader(**new** FileReader(filePath));  
 String line = **null**;  
reader.readLine();  
**while** ((line = reader.readLine()) != **null**) {  
 String[] words = line.split(**","**);  
 String OriginCountry = words[0];  
 String OriginCity = words[1];  
 String OriginCluster = words[2];  
**double** OriginClusterLat = Double.*parseDouble*(words[3]);  
**double** OriginClusterLong = Double.*parseDouble*(words[4]);  
**double** OriginLat = Double.*parseDouble*(words[5]);  
**double** OriginLong = Double.*parseDouble*(words[6]);  
 String DestCountry = words[7];  
 String DestCity = words[8];  
 String DestinationCluster = words[9];  
**double** DestinationClusterLat = Double.*parseDouble*(words[10]);  
**double** DestinationClusterLong = Double.*parseDouble*(words[11]);  
**double** DestLat = Double.*parseDouble*(words[12]);  
**double** DestLong = Double.*parseDouble*(words[13]);  
**double** weight = Double.*parseDouble*(words[14]);  
**int**quantity = Integer.*parseInt*(words[15]);  
**double** volume = Double.*parseDouble*(words[16]);  
  
 Shipment newShipment = **new** Shipment(weight, quantity, volume, **new** Cluster(OriginCluster, OriginClusterLat, OriginClusterLong),  
**new** Cluster(DestinationCluster, DestinationClusterLat, DestinationClusterLong),  
**new** Location(OriginCountry,OriginCity,OriginLat,OriginLong), **new** Location(DestCountry,DestCity,DestLat,DestLong));  
  
**if** (condition.equals(**"All"**)) shipmentList.addShipment(newShipment);  
**else if** (condition.equals(**"Different origin and destination"**)) {  
**if** (!OriginCountry.equals(DestCountry)) shipmentList.addShipment(newShipment);  
 }  
**else** {  
**if** ((OriginCountry.equals(DestCountry)) && (OriginCountry.equals(condition))) shipmentList.addShipment(newShipment);  
 }  
 }  
 } **catch** (FileNotFoundException e) {  
e.printStackTrace();  
 } **catch** (IOException e) {  
e.printStackTrace();  
 } **finally** {  
**if** (reader != **null**) {  
**try** {  
reader.close();  
 } **catch** (IOException e) {  
e.printStackTrace();  
 }  
 }  
 }  
}

Here, if the condition is ‘All’, all the data is read. If the condition is ‘Different origin and origin’, only the shipments that have different origins and destinations are read. If the condition is a country name (such as ‘GERMANY’), only the shipments whose origin and destination are both that country are read into Java.

Annex B3: (Haversine Distance class):

**public class** HaversineDistance {  
  
**private static final int*EARTH\_RADIUS*** = 6371; *// Approx Earth radius in KM***public static double** distance(**double** startLat, **double** startLong,  
**double** endLat, **double** endLong) {  
  
**double** dLat = Math.*toRadians*((endLat - startLat));  
**double** dLong = Math.*toRadians*((endLong - startLong));  
  
startLat = Math.*toRadians*(startLat);  
endLat = Math.*toRadians*(endLat);  
  
**double** a = *haversin*(dLat) + Math.*cos*(startLat) \* Math.*cos*(endLat) \* *haversin*(dLong);  
**double** c = 2 \* Math.*atan2*(Math.*sqrt*(a), Math.*sqrt*(1 - a));  
  
**return *EARTH\_RADIUS*** \* c; *// <-- d*}  
  
**public static double** haversin(**double** value) {  
**return** Math.*pow*(Math.*sin*(value / 2), 2);  
 }  
}

Here, the radius of the Earth is taken to be 6371 kilometers.

Annex B4 (Question 5 -Means Clustering results – Other values of ):

**:**

|  |  |
| --- | --- |
| **Origin**  14 iterations | **Destination**  11 iterations |
| Total distance from locations to clusters: 77356.07676505513 | Total distance from locations to clusters: 118320.77583305737 |
| Average cluster wrt total distance from origin location to origin cluster: Cluster10  Value of average cluster: 1122.376815341619  Average value: 1105.0868109293442  Average cluster wrt total weight from origin location to origin cluster: Cluster53  Value of average cluster: 637963.71  Average value: 671050.4360571429  Average cluster wrt total volume from origin location to origin cluster: Cluster88  Value of average cluster: 14726.874999999985  Average value: 15148.483728571433  Average cluster wrt total quantity from destination location to destination cluster: Cluster32  Value of average cluster: 3313  Average value: 3369.4714285714285 | Average cluster wrt total distance from destination location to destination cluster: Cluster58  Value of average cluster: 1590.739063690564  Average value: 1620.8325456583357  Average cluster wrt total weight from destination location to destination cluster: Cluster43  Value of average cluster: 436016.7  Average value: 643473.0208767126  Average cluster wrt total volume from destination location to destination cluster: Cluster76  Value of average cluster: 14790.657000000005  Average value: 14525.943301369856  Average cluster wrt total quantity from destination location to destination cluster: Cluster13  Value of average cluster: 3336  Average value: 3231.0 |
| Top 3 total origin distances:  1: Cluster21 (8181.916611036634)  2: Cluster95 (7218.665617381495)  3: Cluster46 (6849.443477691635)  Top 3 total origin weights:  1: Cluster14 (6202350.741)  2: Cluster94 (5228299.830000002)  3: Cluster79 (5168122.1899999995)  Top 3 total origin volumes:  1: Cluster81 (139083.1690000008)  2: Cluster46 (114852.26000000014)  3: Cluster50 (105832.82899999969)  Top 3 total origin quantities:  1: Cluster14 (50565)  2: Cluster57 (27969)  3: Cluster21 (21700) | Top 3 total destination distances:  1: Cluster46 (13823.595929851275)  2: Cluster13 (12788.308706972615)  3: Cluster92 (12100.22195627136)  Top 3 total destination weights:  1: Cluster7 (8552623.767999995)  2: Cluster52 (8440544.904)  3: Cluster70 (4992409.416)  Top 3 total destination volumes:  1: Cluster7 (250597.9159999995)  2: Cluster77 (160409.3820000009)  3: Cluster46 (122153.49600000012)  Top 3 total destination quantities:  1: Cluster7 (59535)  2: Cluster59 (44498)  3: Cluster80 (27617) |
| Bottom 3 total origin distances:  1: Cluster43 (0.0)  2: Cluster37 (0.0)  3: Cluster35 (0.0)  Bottom 3 total origin weights:  1: Cluster28 (65.34)  2: Cluster43 (70.0)  3: Cluster63 (96.25999999999999)  Bottom 3 total origin volumes:  1: Cluster43 (0.18)  2: Cluster28 (0.374)  3: Cluster63 (0.9700000000000002)  Bottom 3 total origin quantities:  1: Cluster28 (4)  2: Cluster43 (6)  3: Cluster83 (6) | Bottom 3 total destination distances:  1: Cluster14 (0.0)  2: Cluster86 (0.0)  3: Cluster37 (0.0)  Bottom 3 total destination weights:  1: Cluster65 (1.0)  2: Cluster37 (5.0)  3: Cluster20 (12.0)  Bottom 3 total destination volumes:  1: Cluster65 (0.001)  2: Cluster37 (0.032)  3: Cluster20 (0.067)  Bottom 3 total destination quantities:  1: Cluster37 (1)  2: Cluster44 (1)  3: Cluster65 (1) |

**:**

|  |  |
| --- | --- |
| **Origin**  12 iterations | **Destination**  3 iterations |
| Total distance from locations to clusters: 38235.48005693789 | Total distance from locations to clusters: 2823713.778543799 |
| Average cluster wrt total distance from origin location to origin cluster: Cluster688  Value of average cluster: 144.0753916829532  Average value: 145.93694678220564  Average cluster wrt total weight from origin location to origin cluster: Cluster226  Value of average cluster: 171109.55999999997  Average value: 179288.2844427481  Average cluster wrt total volume from origin location to origin cluster: Cluster501  Value of average cluster: 4296.881000000003  Average value: 4047.3048129771028  Average cluster wrt total quantity from destination location to destination cluster: Cluster260  Value of average cluster: 880  Average value: 900.2404580152672 | Average cluster wrt total distance from destination location to destination cluster: Cluster1  Value of average cluster: 2823713.778543799  Average value: 2823713.778543799  Average cluster wrt total weight from destination location to destination cluster: Cluster1  Value of average cluster: 4.697353052400006E7  Average value: 4.697353052400006E7  Average cluster wrt total volume from destination location to destination cluster: Cluster1  Value of average cluster: 1060393.8609999989  Average value: 1060393.8609999989  Average cluster wrt total quantity from destination location to destination cluster: Cluster1  Value of average cluster: 235863  Average value: 235863.0 |
| Top 3 total origin distances:  1: Cluster383 (9638.093408490631)  2: Cluster573 (4943.348595755281)  3: Cluster333 (3964.8155865984763)  Top 3 total origin weights:  1: Cluster333 (5234706.490000001)  2: Cluster1022 (5168302.449999999)  3: Cluster383 (4009927.1440000003)  Top 3 total origin volumes:  1: Cluster1018 (139047.7480000008)  2: Cluster592 (106451.85600000009)  3: Cluster648 (104649.6179999997)  Top 3 total origin quantities:  1: Cluster170 (27013)  2: Cluster740 (24848)  3: Cluster1010 (22204) | Top 3 total destination distances:  1: Cluster1 (2823713.778543799)  Top 3 total destination weights:  1: Cluster1 (4.697353052400006E7)  Top 3 total destination volumes:  1: Cluster1 (1060393.8609999989)  Top 3 total destination quantities:  1: Cluster1 (235863) |
| Bottom 3 total origin distances:  1: Cluster2 (0.0)  2: Cluster454 (0.0)  3: Cluster421 (0.0)  Bottom 3 total origin weights:  1: Cluster93 (0.22)  2: Cluster168 (1.0)  3: Cluster576 (1.0)  Bottom 3 total origin volumes:  1: Cluster332 (0.0)  2: Cluster168 (0.003)  3: Cluster694 (0.009)  Bottom 3 total origin quantities:  1: Cluster454 (1)  2: Cluster168 (1)  3: Cluster76 (1) | Bottom 3 total destination distances:  1: Cluster1 (2823713.778543799)  Bottom 3 total destination weights:  1: Cluster1 (4.697353052400006E7)  Bottom 3 total destination volumes:  1: Cluster1 (1060393.8609999989)  Bottom 3 total destination quantities:  1: Cluster1 (235863) |

Annex B4 (Question 6 -Means Clustering results):

(Only the average clusters are shown below)

**Germany**

Number of shipments for Germany: 938

Performed k-means clustering of origin clusters with k = 240

Number of iterations: 10

Performed k-means clustering of destination clusters with k = 240

Number of iterations: 3

Summary of k-means clustering:

Total distance from origin location to origin cluster: 6157.795243362836

Average cluster wrt total distance from origin location to origin cluster: Cluster192

Value of average cluster: 66.67100801295891

Average value: 68.41994714847584

Average cluster wrt total weight from origin location to origin cluster: Cluster2

Value of average cluster: 120468.15

Average value: 93604.71741111112

Average cluster wrt total volume from origin location to origin cluster: Cluster87

Value of average cluster: 3175.565000000003

Average value: 2314.57678888889

Average cluster wrt total quantity from destination location to destination cluster: Cluster179

Value of average cluster: 349

Average value: 376.26666666666665

Total distance from destination location to destination cluster: 248209.49474799467

Average cluster wrt total distance from destination location to destination cluster: Cluster1

Value of average cluster: 248209.49474799467

Average value: 248209.49474799467

Average cluster wrt total weight from destination location to destination cluster: Cluster1

Value of average cluster: 8424424.567000004

Average value: 8424424.567000004

Average cluster wrt total volume from destination location to destination cluster: Cluster1

Value of average cluster: 208311.91100000025

Average value: 208311.91100000025

Average cluster wrt total quantity from destination location to destination cluster: Cluster1

Value of average cluster: 33864

Average value: 33864.0

**France**

Number of shipments for France: 1053

Performed k-means clustering of origin clusters with k = 270

Number of iterations: 3

Performed k-means clustering of destination clusters with k = 270

Number of iterations: 7

Summary of k-means clustering:

Total distance from origin location to origin cluster: 306312.8611600129

Average cluster wrt total distance from origin location to origin cluster: Cluster1

Value of average cluster: 306312.8611600129

Average value: 306312.8611600129

Average cluster wrt total weight from origin location to origin cluster: Cluster1

Value of average cluster: 7477013.352999998

Average value: 7477013.352999998

Average cluster wrt total volume from origin location to origin cluster: Cluster1

Value of average cluster: 255506.0990000001

Average value: 255506.0990000001

Average cluster wrt total quantity from destination location to destination cluster: Cluster1

Value of average cluster: 56179

Average value: 56179.0

Total distance from destination location to destination cluster: 13337.480263023066

Average cluster wrt total distance from destination location to destination cluster: Cluster55

Value of average cluster: 118.962695866237

Average value: 122.36220424791814

Average cluster wrt total weight from destination location to destination cluster: Cluster14

Value of average cluster: 65428.46

Average value: 68596.45277981652

Average cluster wrt total volume from destination location to destination cluster: Cluster68

Value of average cluster: 2146.675

Average value: 2344.0926513761465

Average cluster wrt total quantity from destination location to destination cluster: Cluster162

Value of average cluster: 476

Average value: 515.4036697247707

**Spain**

Number of shipments for Spain: 414

Performed k-means clustering of origin clusters with k = 106

Number of iterations: 3

Performed k-means clustering of destination clusters with k = 106

Number of iterations: 4

Summary of k-means clustering:

Total distance from origin location to origin cluster: 134135.58381468552

Average cluster wrt total distance from origin location to origin cluster: Cluster1

Value of average cluster: 134135.58381468552

Average value: 134135.58381468552

Average cluster wrt total weight from origin location to origin cluster: Cluster1

Value of average cluster: 7645390.634999998

Average value: 7645390.634999998

Average cluster wrt total volume from origin location to origin cluster: Cluster1

Value of average cluster: 256550.7840000008

Average value: 256550.7840000008

Average cluster wrt total quantity from destination location to destination cluster: Cluster1

Value of average cluster: 37102

Average value: 37102.0

Total distance from destination location to destination cluster: 3675.1005384659024

Average cluster wrt total distance from destination location to destination cluster: Cluster24

Value of average cluster: 124.54623389036485

Average value: 131.25359065949655

Average cluster wrt total weight from destination location to destination cluster: Cluster1

Value of average cluster: 159243.0

Average value: 273049.6655357142

Average cluster wrt total volume from destination location to destination cluster: Cluster66

Value of average cluster: 7804.5880000000025

Average value: 9162.528000000028

Average cluster wrt total quantity from destination location to destination cluster: Cluster61

Value of average cluster: 1373

Average value: 1325.0714285714287

1. Haversine formula. (2018, June 10). Retrieved June 14, 2018, from <https://en.wikipedia.org/wiki/Haversine_formula> [↑](#footnote-ref-1)