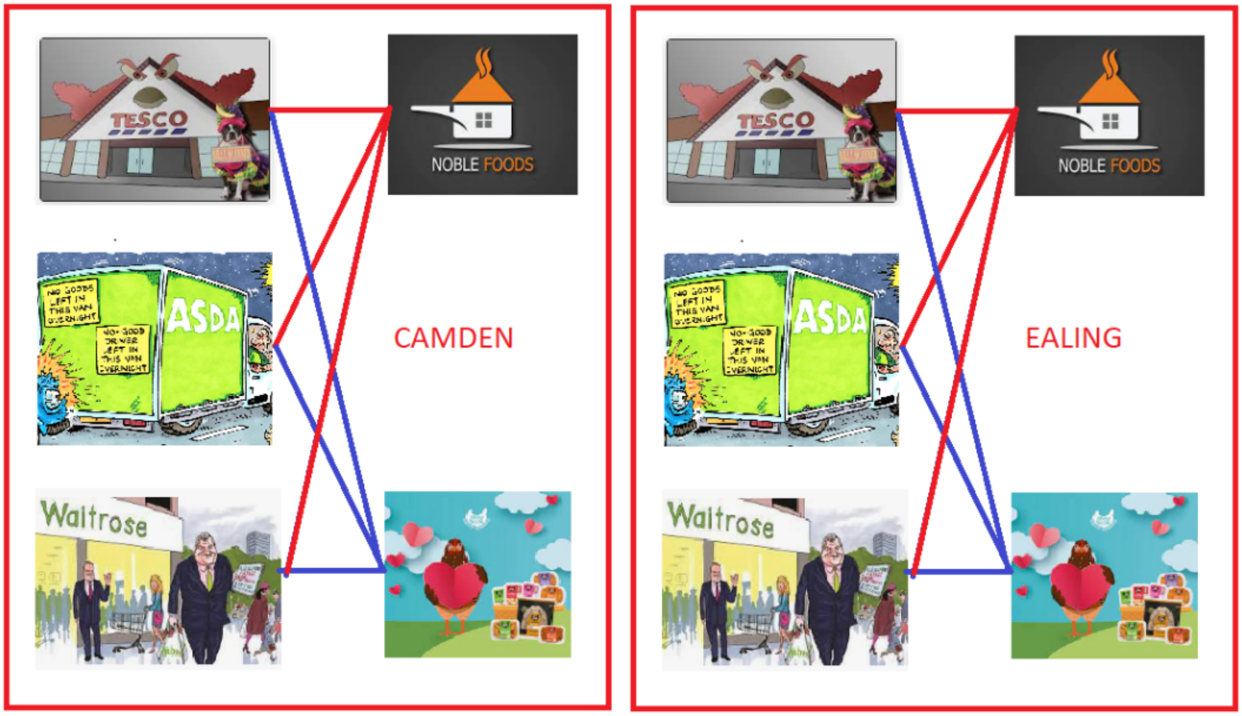
Resource Allocation based on DHondt Method and BigQuery SHort

1. Problem statement:

We gave three supermarket brands (for example, Lidl, Tesco, and Asda) we denote them Shop 1, Shop 2, and Shop 3. They have branches in Camden, Ealing, Greenwich, Hounslow, Richmond upon Thames, Hammersmith and Fulham, Kensington and Chelsea, and the City of Westminster. They use two suppliers that have branches in the same London Boroughs, we have a historical probability of items being in the given supplier branch of London Borough. We know how many items each supplier can deliver per day.

We are given a total number of items demanded by every supermarket brand and information about when (date) and where (London Borough) the given supermarket can accept part of this total demand but the exact value is not provided. The supplier can deliver items to supermarkets from the same Borough only. We have weights or a probability that the total supply for the given location is going to a particular supermarket. The task is to find the date, location, and number of items that should be delivered to all three supermarkets.



We are given Demand Weights that could be weights of the objective function that are used in linear programming:

Table 1 Demand Weights defined by Optimization Objective in Linear Programming

Demand Weight		
Shop 1	Shop 2	Shop 3
0.2	0.3	0.5

The table below shows how many items are required by every supermarket in all considered locations during two days:

Table 2 Total Number of Demanded Items per Time Interval (2 days in our case)

Shop_Demand		
Shop 1	Shop 2	Shop 3
150	500	300

Information about when (date) and where (London Borough) the given supermarket is available to accept part of the above total demand (see the table called Shop_Demand above) but how much it will accept it depends on the supply ib the given Borough and

Demand Weights.

If it is "+" (True) then the supermarket is available to accept the items on the particular date and London Borough (location) and it does not accept otherwise (when it is "-" or False:

Table 3 Shop Demand Availability or Schedule when Shops update their Stocks

N	Date	London Borough	Shop Demand Availability		
			Shop 1	Shop 2	Shop 3
1	11/06/2022	Camden	-	-	-
2	11/06/2022	Ealing	-	-	+
3	11/06/2022	Greenwich	-	+	-
4	11/06/2022	Hounslow	-	+	+
5	11/06/2022	Richmond upon Thames	+	-	-
6	11/06/2022	Hammersmith and Fulham	+	-	+
7	11/06/2022	Kensington and Chelsea	+	+	-
8	11/06/2022	City of Westminster	+	+	+
9	12/06/2022	Camden	-	-	-
10	12/06/2022	Ealing	-	-	+
11	12/06/2022	Greenwich	-	+	-
12	12/06/2022	Hounslow	-	+	+
13	12/06/2022	Richmond upon Thames	+	-	-
14	12/06/2022	Hammersmith and Fulham	+	-	+
15	12/06/2022	Kensington and Chelsea	+	+	-
16	12/06/2022	City of Westminster	+	+	+
17	Total	-	150	500	300

The information about Supply includes the probability of supply over different London Borough (location) and the daily total amount of the supplied items (see Table below). The probability could be different for the different suppliers but in our example, it is the same for simplicity of presentation:

Table 4 Supply Items Availability

N	Date	London Borough	Supply			
			Supplier 1		Supplier 2	
			Probability	Daily Total	Probability	Daily Total
1	11/06/2022	Camden	0.118		0.118	
2	11/06/2022	Ealing	0.157		0.157	
3	11/06/2022	Greenwich	0.118		0.118	
4	11/06/2022	Hounslow	0.220	255	0.220	255
5	11/06/2022	Richmond upon Thames	0.078		0.078	
6	11/06/2022	Hammersmith and Fulham	0.192		0.192	
7	11/06/2022	Kensington and Chelsea	0.039		0.039	
8	11/06/2022	City of Westminster	0.078		0.078	
9	12/06/2022	Camden	0.118		0.118	
10	12/06/2022	Ealing	0.157		0.157	
11	12/06/2022	Greenwich	0.118		0.118	
12	12/06/2022	Hounslow	0.220	510	0.220	255
13	12/06/2022	Richmond upon Thames	0.078		0.078	
14	12/06/2022	Hammersmith and Fulham	0.192		0.192	
15	12/06/2022	Kensington and Chelsea	0.039		0.039	
16	12/06/2022	City of Westminster	0.078		0.078	
17	Total		2	765	2	510

The algorithm consists of 3 major steps:

STEP 1. Computation of the number of supplied items per day, Borough, and supplier given the probability of supply over different Boroughs and the total number of supplied items per day and supplier. In the simplest case, we just multiply the probability of items distributed to a given Borough by the total number of supplied items per day, and supplier. But sometimes we get no integer numbers after this multiplication that is why we apply DHondt method instead. For example, given date 11/06/2022 and supplier 1 (see Table 4) we have the total amount of items available equal to 255 and the following vector of probabilities: `[0.118, 0.157, 0.118, 0.220, 0.078, 0.192, 0.039, 0.078]` over London Boroughs. This information is required by DHondt method to find the number of items that can be supplied on 11/06/2022 by Supplier 1 to each London Borough. Therefore, for every combination of date and supply we need to run DHondt method or 4 times in our case (see Quantity column in Table 5).

Table 5 Supply Information with estimated “Quantity” and “ALL” Columns

N	Date	London Borough	Supply						
			Supplier 1			Supplier 2			ALL
			Probability	Quantity	Total	Probability	Quantity	Total	
1	11/06/2022	Camden	0.118	30	255	0.118	30	255	60
2	11/06/2022	Ealing	0.157	40		0.157	40		80
3	11/06/2022	Greenwich	0.118	30		0.118	30		60
4	11/06/2022	Hounslow	0.220	56		0.220	56		112
5	11/06/2022	Richmond upon Thames	0.078	20		0.078	20		40
6	11/06/2022	Hammersmith and Fulham	0.192	49		0.192	49		98
7	11/06/2022	Kensington and Chelsea	0.039	10		0.039	10		20
8	11/06/2022	City of Westminster	0.078	20		0.078	20		40
9	12/06/2022	Camden	0.118	60	510	0.118	30	255	90
10	12/06/2022	Ealing	0.157	80		0.157	40		120
11	12/06/2022	Greenwich	0.118	60		0.118	30		90
12	12/06/2022	Hounslow	0.220	112		0.220	56		168
13	12/06/2022	Richmond upon Thames	0.078	40		0.078	20		60
14	12/06/2022	Hammersmith and Fulham	0.192	98		0.192	49		147
15	12/06/2022	Kensington and Chelsea	0.039	20		0.039	10		30
16	12/06/2022	City of Westminster	0.078	40		0.078	20		60
17	Total			765	765		510	510	1275

STEP 2. Calculation of the maximum demand that can be satisfied or supplied. Using demand weights, demand availability, and the total possible number of supplied items per day and Borough (see 1)) we compute how many items can be supplied given day, Borough, and shop. Therefore, we find the maximum number of items that the shop located in the corresponding London Borough can get on the given date. In order to find it we first multiply every row of Shop Demand Availability (see Table 3, where “+” can be interpreted as 1 and “-” as 0) by Demand Weights and normalise every row of the result in such a way that weights in the row sum to 1. Then using this row normalised weights and the total possible supplied in the given London Borough and date we calculate the number of the items that can be delivered to the corresponding shop on the given date and London Borough. We can apply DHondt method again - we need to run it 16 times for every combination of date and London Borough, the output of the DHondt in this case is the number of items (in our case, the output is a vector of three values - one value per shop) for the corresponding shop given date and London Borough (see Table 6). Then we aggregate the number of items over all dates and London Borough for every shop. Therefore, we get the number of items that suppliers can provide to the corresponding entire supermarket network for all location and dates. then we compare these numbers with Shop Demand (see Table 2) and for every shop choose the smallest number (see Table 7). Therefore, we find the number of items that are required by the shops and can be delivered by suppliers subject to constrain (see Table 1-4) over all days (in our case two days) and London Borough (we selected eight London Boroughs).

Table 6 Distribution of Number of Available for Supply Items over Shops given Date and London Borough (see Supply AS Demand in red colour)

N	Date	London Borough	Shop Demand Availability			Demand Weight			Row Normalised Weight			Supply AS Demand		
			Shop 1	Shop 2	Shop 3	Shop 1	Shop 2	Shop 3	Shop 1	Shop 2	Shop 3	Shop 1	Shop 2	Shop 3
1	11/06/2022	Camden	-	-	-	0	0	0	0	0	0	0	0	0
2	11/06/2022	Ealing	-	-	+	0	0	0.5	0	0	1	0	0	80
3	11/06/2022	Greenwich	-	+	-	0	0.3	0	0	1	0	0	60	0
4	11/06/2022	Hounslow	-	+	+	0	0.3	0.5	0	0.375	0.625	0	42	70
5	11/06/2022	Richmond upon Thames	+	-	-	0.2	0	0	1	0	0	40	0	0
6	11/06/2022	Hammersmith and Fulham	+	-	+	0.2	0	0.5	0.285714	0	0.714286	28	0	70
7	11/06/2022	Kensington and Chelsea	+	+	-	0.2	0.3	0	0.4	0.6	0	8	12	0
8	11/06/2022	City of Westminster	+	+	+	0.2	0.3	0.5	0.2	0.3	0.5	8	12	20
9	12/06/2022	Camden	-	-	-	0	0	0	0	0	0	0	0	0
10	12/06/2022	Ealing	-	-	+	0	0	0.5	0	0	1	0	0	120
11	12/06/2022	Greenwich	-	+	-	0	0.3	0	0	1	0	0	90	0
12	12/06/2022	Hounslow	-	+	+	0	0.3	0.5	0	0.375	0.625	0	63	105
13	12/06/2022	Richmond upon Thames	+	-	-	0.2	0	0	1	0	0	60	0	0
14	12/06/2022	Hammersmith and Fulham	+	-	+	0.2	0	0.5	0.285714	0	0.714286	42	0	105
15	12/06/2022	Kensington and Chelsea	+	+	-	0.2	0.3	0	0.4	0.6	0	12	18	0
16	12/06/2022	City of Westminster	+	+	+	0.2	0.3	0.5	0.2	0.3	0.5	12	18	30
17	Total	-	150	500	300	-						210	315	600

Table 7 Available Demand Calculation

Super Market	Demand	Supply	Available Demand
Shop 1	150	210	150
Shop 2	500	315	315
Shop 3	300	600	300

STEP 3. Find weights for each shop using the corresponding column of Supply AS Demand (see Table 6) that is divided the corresponding total (for example, the total equal to 210, 315 and 600 for shop 1, 2 and 3, respectively, see Table 6). As a result we get “Column Normalise Weight” (see Table 8). After this we can multiply every column of “Column Normalised Weight” (see Table 8) that corresponds to the given shop by the corresponding value of “Available Demand” (see Table 7) to get the final allocation. As we mentioned before this approach give some float numbers therefore instead we apply DHondt method to every column of “Column Normalised Weight” (see Table 8) using the corresponding value of Available Demand (see Table 7). In our case we have three shops therefore we apply DHondt method three times and we get “Final Allocation” (see Table 8). Please note that, the following implementation in BigQuery gives slightly different numbers than what we get in Table 8 but it can be explained by difference in Python and BigQuery DHondt method implementations.

Table 8

N	Date	London Borough	Supply AS Demand			Column Normalised Weight			Final Allocation		
			Shop 1	Shop 2	Shop 3	Shop 1	Shop 2	Shop 3	Shop 1	Shop 2	Shop 3
1	11/06/2022	Camden	0	0	0	0	0	0	0	0	0
2	11/06/2022	Ealing	0	0	80	0	0	0.133333	0	0	40
3	11/06/2022	Greenwich	0	60	0	0	0.190476	0	0	60	0
4	11/06/2022	Hounslow	0	42	70	0	0.133333	0.116667	0	42	35
5	11/06/2022	Richmond upon Thames	40	0	0	0.190476	0	0	29	0	0
6	11/06/2022	Hammersmith and Fulham	28	0	70	0.133333	0	0.116667	20	0	35
7	11/06/2022	Kensington and Chelsea	8	12	0	0.038095	0.038095	0	5	12	0
8	11/06/2022	City of Westminster	8	12	20	0.038095	0.038095	0.033333	5	12	10
9	11/06/2022	Camden	0	0	0	0	0	0	0	0	0
10	12/06/2022	Ealing	0	0	120	0	0	0.2	0	0	60
11	12/06/2022	Greenwich	0	90	0	0	0.285714	0	0	90	0
12	12/06/2022	Hounslow	0	63	105	0	0.2	0.175	0	63	53
13	12/06/2022	Richmond upon Thames	60	0	0	0.285714	0	0	44	0	0
14	12/06/2022	Hammersmith and Fulham	42	0	105	0.2	0	0.175	31	0	52
15	12/06/2022	Kensington and Chelsea	12	18	0	0.057143	0.057143	0	8	18	0
15	12/06/2022	City of Westminster	12	18	30	0.057143	0.057143	0.05	8	18	15
17	Total	-	210	315	600	1	1	1	150	315	300