E-Food Assignment

Vaggelis Stathis

July 2022

1 Part 1

```

--CREATE VIEW main_assessment.cities_with_many_orders AS
--SELECT *
--FROM ‘efood2022-357409.main_assessment.orders‘ WHERE CITY IN (select city  FROM ‘efood2022-357409.main_assessment.orders‘ group by city  having count(order_id)  >1000 );

--CREATE VIEW main_assessment.cities_with_many_breakfast_orders AS
--SELECT *
--FROM ‘efood2022-357409.main_assessment.orders‘ WHERE CITY IN (select city  FROM ‘efood2022-357409.main_assessment.orders‘ group by city  having count(order_id)  >1000 ) and cuisine = ’Breakfast’;



--Query #1

with br as (
  SELECT table_agg.city,
  num_of_orders,
  round((table_br_agg.total_amount_br / table_br_agg.num_of_orders_br),2) as avg_breakfast_basket,
  round((table_agg.total_amount / table_agg.num_of_orders),2) as avg_efood_basket,
  round((table_br_agg.num_of_orders_br / table_br_agg.num_of_users_br),2) as avg_breakfast_frequency_per_brfst_user,
  round((table_br_agg.num_of_orders_br / table_agg.num_of_users),2) as avg_breakfast_frequency_per_user,
  round((table_agg.num_of_orders / table_agg.num_of_users),2) as avg_order_frequency,
  concat(round((table_br_agg_3.number_of_users_3_br_orders / table_br_agg.num_of_users_br * 100),2), ’%’)as breakfast_frequent_users_perc_of_breakfast_total_users,
  concat(round((table_agg_3.number_of_users_3_orders / table_agg.num_of_users * 100),2), ’%’) as frequent_users_perc_of_total_users,
FROM
(select city, sum(amount) as total_amount, count(order_id) as num_of_orders, count(distinct user_id) num_of_users FROM main_assessment.cities_with_many_orders group by city) as table_agg,
(select city, sum(amount) as total_amount_br, count(order_id) as num_of_orders_br, count(distinct user_id) num_of_users_br FROM main_assessment.cities_with_many_breakfast_orders group by city) as table_br_agg,
(select city, count(m) as number_of_users_3_orders from (select city, count(user_id) as m  FROM main_assessment.cities_with_many_orders group by city, user_id having count(user_id)  >3 order by city) group by city) as table_agg_3,
(select city, count(m) as number_of_users_3_br_orders from (select city,  count(user_id) as m  FROM main_assessment.cities_with_many_breakfast_orders  group by city, user_id having count(user_id)  >3 order by city) group by city)  as
table_br_agg_3
where table_agg.city = table_br_agg.city
and table_agg.city = table_agg_3.city
and table_agg.city = table_br_agg_3.city
)
select city,
avg_breakfast_basket,
avg_efood_basket,
concat(round((avg_breakfast_basket / avg_efood_basket *100),2), '%') breakfast_basket_perc_of_total_basket,
avg_breakfast_frequency_per_brfst_user,
avg_breakfast_frequency_per_user,
avg_order_frequency,
concat(round((avg_breakfast_frequency_per_user / avg_order_frequency *100),2), '%') breakfast_freq_perc_of_total_order_freq,
breakfast_frequent_users_perc_of_breakfast_total_users,
frequent_users_perc_of_total_users from br order by num_of_orders desc limit 5;
```

Listing 1: First Query

```

--CREATE VIEW main_assessment.city_best_customers AS
--(
  --  SELECT *, ROW_NUMBER()
   -- over (
    --    PARTITION BY city
    --    order by m desc
   -- ) AS RowNo
  --  FROM (select city,user_id,   count(user_id) as m  FROM main_assessment.cities_with_many_orders group by city, user_id order by city, m desc)
--);

--CREATE VIEW main_assessment.city_best_customers_am AS
--(
   -- SELECT *, ROW_NUMBER()
   -- over (
   --     PARTITION BY city
    --    order by m desc
   -- ) AS RowNo
--FROM (select city,user_id,   count(user_id) as m,  round(sum(amount),2) as amount  FROM main_assessment.cities_with_many_orders group by city, user_id  order by city, m desc)
--);

SELECT top10.city,
round((top10.orders_sum_top_10 / total.orders_sum *100),2) orders_percentage_top_10_customers
FROM
(SELECT city, sum(m) as orders_sum_top_10 FROM main_assessment.city_best_customers WHERE RowNo <= 10 group by city) as top10,
(SELECT city, sum(m) as orders_sum FROM main_assessment.city_best_customers  group by city) as total
where top10.city = total.city order by 2 desc;
```

Listing 2: Second Query

2 Part 2

I performed an FM (Frequency, Monetary) analysis according to the instructions. In order to do that, I had to create multiple score variables.

FScore: is the score for the frequency of the total number of orders that a user made. It takes values from 1 to 4.

FBRScore: is the score for the frequency of the number of breakfast orders that a user made. It takes values from 1 to 5.

MScore: is the score for the total amount that a user paid. It takes values from 1 to 4.

FWBRScore: is the score for the frequency of the total number of orders that a user made, if we subtract the breakfast orders. It takes values from 1 to 5.

I created the last score variable (FWBRScore) because 'FScore' was not enough to describe a client since a great number of breakfast orders (FBRScore = 5) implies a great Fscore (FScore = 4). Then, i concatenated the FWBRScore and the Mscore, to create the final segments which where stored in a new variable MFWBRScores. The latest, has 20 values ('1 1', '1 2', ..., '4 4', '5 4'). Finally, i created a pie-chart to see the segments that form the group of users with FBRScore = 5. In this graph it is clear that segment '5 4' is by far the biggest and thus, this is the one that should be targeted for the campaign. The results are similar if we produce the same chart for both FBRScore = 4 and FBRScore = 5.

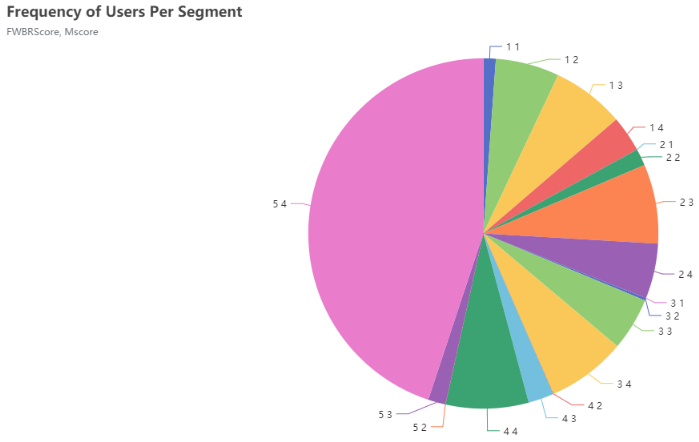
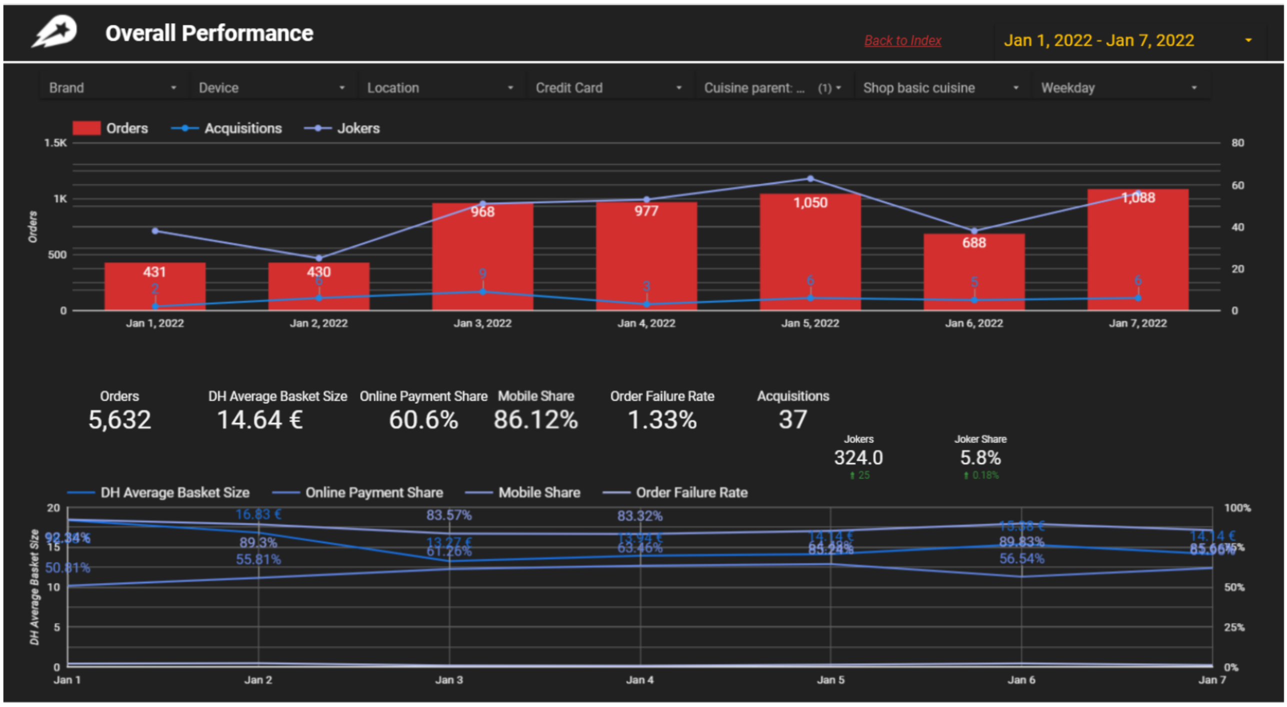


Figure 1: Pie Chart for FBRScore = 5

3 Part 3

3.1 The Problems



Overall Performance

Back to Index

Jan 1, 2022 - Jan 7, 2022

Brand | Device | Location | Credit Card | Cuisine parent: ... (1) | Shop basic cuisine | Weekday

Orders 5,632

DH Average Basket Size 14.64 €

Online Payment Share 60.6%

Mobile Share 86.12%

Order Failure Rate 1.33%

Acquisitions 37

Jokers 324.0 ⬆ 25

Joker Share 5.8% ⬆ 0.18%

Figure 2: page 1

The colours of the lines are similar. Mobile Share and Order Failure Rate are very hard to differentiate. Solution: add more distinct colours.

The Jokers line in the first graph has no labels to declare the size. Since there are two x-axis it is not a good tactic. Solution: add labels.

There are extra horizontal lines, that do not add anything in the graph. Solution: Should be deleted.

It would be nice to show days names except of dates, so that the user understands which days have the lowest orders. Probably it's Saturday and Sunday.

Although i am a big of the dark themes, i believe that a bright theme should be applied. Should a stakeholder be outdoors, it will be impossible to see the screen.

Jokers and Joker Share, could be in the same hight with the rest of the KPIs. In addition, it would be nice to see how the rest of the KPIs differ from last week.

The labels in the second graph overlap with the y-axis (both left and right) and with each other. In addition, the 'order failure rate' it is very difficult to be distinguished from the x-axis. Solution: add space between both y-axis and the first and last date. Furthermore, maybe we could add an option for the user to choose which lines to project. This way 'order failure rate' would be shown properly.

In the second graph, there is a null value. It should be clarified.

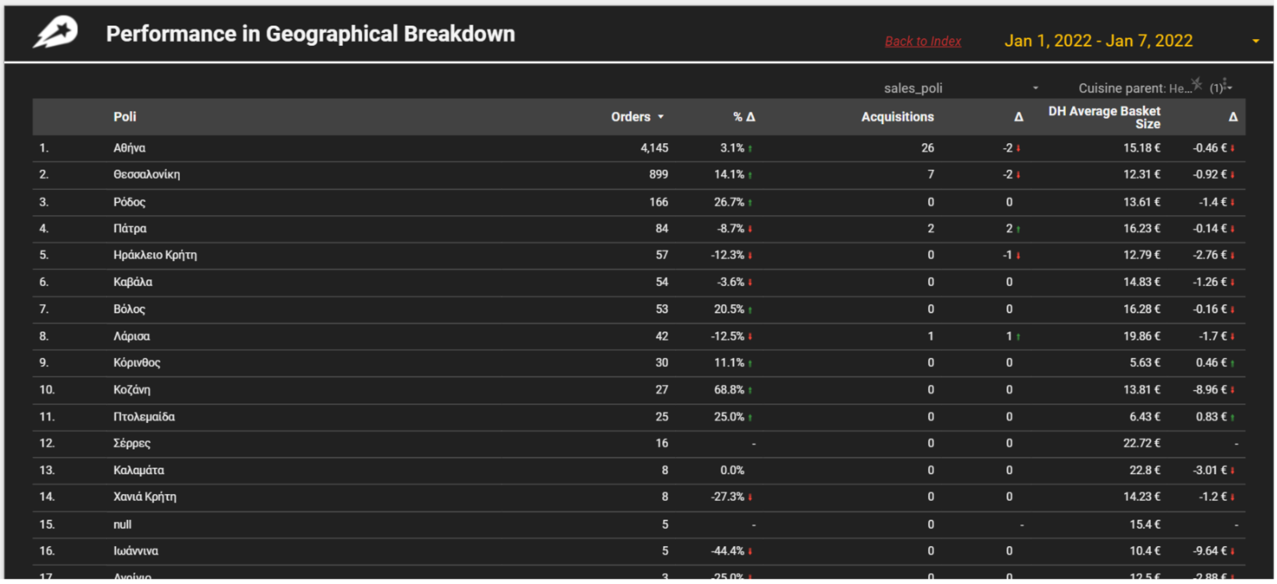
If it was in my hand i would also change 'Poli' to 'City' in the second graph.

3.2 Findings

The Jokers line follow the number of orders bars while the acquisitions don't.

In the second graph, the total orders rate is increased (probably compared with the previous week) while the rate of acquisitions and the average basket size has decreased.

I believe that the rest of the information is very easy for someone to see.



Performance in Geographical Breakdown
Back to Index
Jan 1, 2022 - Jan 7, 2022
sales_poli
Cuisine parent: He... (1)

	Poli	Orders ▾	% Δ	Acquisitions	Δ	DH Average Basket Size	Δ
1.	Αθήνα	4,145	3.1% ↑	26	-2 ↓	15.18 €	-0.46 € ↓
2.	Θεσσαλονίκη	899	14.1% ↑	7	-2 ↓	12.31 €	-0.92 € ↓
3.	Ρόδος	166	26.7% ↑	0	0	13.61 €	-1.4 € ↓
4.	Πάτρα	84	-8.7% ↓	2	2 ↑	16.23 €	-0.14 € ↓
5.	Ηράκλειο Κρήτη	57	-12.3% ↓	0	-1 ↓	12.79 €	-2.76 € ↓
6.	Καβάλα	54	-3.6% ↓	0	0	14.83 €	-1.26 € ↓
7.	Βόλος	53	20.5% ↑	0	0	16.28 €	-0.16 € ↓
8.	Λάρισα	42	-12.5% ↓	1	1 ↑	19.86 €	-1.7 € ↓
9.	Κόρινθος	30	11.1% ↑	0	0	5.63 €	0.46 € ↑
10.	Κοζάνη	27	68.8% ↑	0	0	13.81 €	-8.96 € ↓
11.	Πτολεμαίδα	25	25.0% ↑	0	0	6.43 €	0.83 € ↑
12.	Σέρρες	16	-	0	0	22.72 €	-
13.	Καλαμάτα	8	0.0%	0	0	22.8 €	-3.01 € ↓
14.	Χανιά Κρήτη	8	-27.3% ↓	0	0	14.23 €	-1.2 € ↓
15.	null	5	-	0	-	15.4 €	-
16.	Ιωάννινα	5	-44.4% ↓	0	0	10.4 €	-9.64 € ↓
17.	Αγρίνιο	3	-25.0% ↓	0	0	12.5 €	-2.88 € ↓

Figure 3: page 2