Analyzing a Large-Scale Retailer

1. Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset.

A. Datatype of all columns in the "customers" table.

Query:
```
SELECT column_name, data_type
 FROM
`target-394104.Target_brazil.INFORMATION_SCHEMA.COLUMNS`
 WHERE table_name = 'customers'
```

Output:

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	column_name	data_type
1	customer_id	STRING
2	customer_unique_id	STRING
3	customer_zip_code_prefix	INT64
4	customer_city	STRING
5	customer_state	STRING

Insights:

The above output indicates the data types of the customer column.

Understanding the characteristics play a vital role while processing the data and this can befetched from the information of data types displayed above.

The above information can be used to verify whether certain columns are joined with sametype in order to avoid further complications.

Certain columns have to be preprocessed before we analyze the
dataEx: conversion of text data to lower case.

The data types make us to ensure whether the numerical columns contain numerical data andcategorical columns contain categorical data.

The in detailed checking of data from the tables lets us know the column containing null values and act whether to exclude these null values while processing.

Recommendations:

The Datatypes of columns help us in letting know the type of data to be fitted onto that particular column which helps in Data Integrity.

2. Get the time range between which the orders were placed.

Query :

```
select
min(extract(datetime from order_purchase_timestamp at Time Zone "Asia/Kolkata")) as
min_time_range,
max(extract(datetime from order_purchase_timestamp at Time Zone "Asia/Kolkata")) as
max_time_range,
count(order_id) as order_count
from `target-394104.Target_brazil.orders`
```

Output:

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH

Row	min_time_range	max_time_range	order_count
1	2016-09-05T02:45:19	2018-10-17T23:00:18	99441

In Sights:

The minimum and maximum time range provides us the time period covered by the dataset.

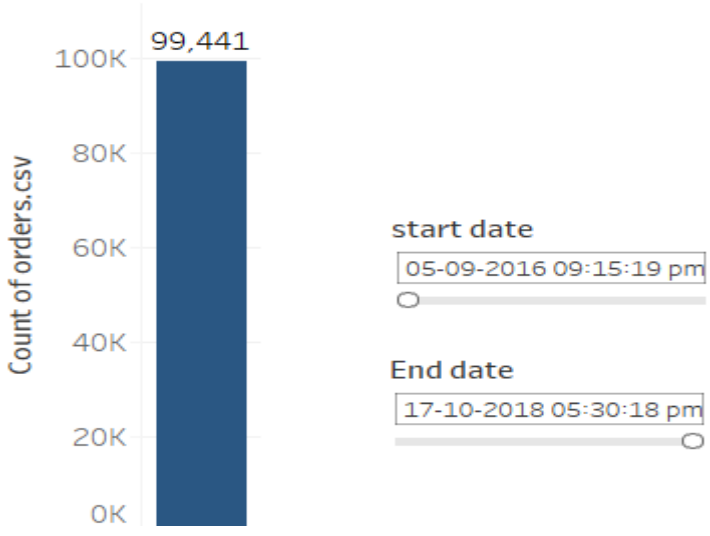
Assumptions:

The minimum time range and maximum time range are evaluated from the table named "orders"From the column name "order purchase timestamp"

The range helps us in maintaining Data Quality and in understanding Data Integrity.

Recommendations:

Since the maximum time range is much older than minimum time range, we can certainly usethis data to indicate the gaps or incomplete records. Reviewing this information can help in maintaining data integrity.



In the above graph, **Parameters** are used to display the min and max time range and the horizontal bar depicts the total count of orders during this period.

3. **Count the number of Cities and States in our dataset.**

Query:

```
select count(distinct customer_state) as unique_states,
count(distinct customer_city) as unique_cities
from `target-394104.Target_brazil.customers`
```

Output:

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH

Row	unique_states	unique_cities
1	27	4119

In sights:

Identifying Unique States and Cities enable us to understand the geographical distribution ofdata which helps in business decision making

Recommendations:

The results for the above query help us in understanding the region-specific data with respectto customer among different cities and states.

II. In-depth Exploration:

1. Is there a growing trend in the no. of orders placed over the past years?

Query:

```
select format_date("%B", order_purchase_timestamp) as month,
format_date("%Y", order_purchase_timestamp) as year ,count(distinct order_id) as
no_of_orders
from `target-394104.Target_brazil.orders`
group by month,year
order by year,month
```

Output:

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | CHART

Row	month	year	no_of_orders
1	December	2016	1
2	October	2016	324
3	September	2016	4
4	April	2017	2404
5	August	2017	4331
6	December	2017	5673
7	February	2017	1780
8	January	2017	800
9	July	2017	4026
10	June	2017	3245

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | CHART

Row	month	year	no_of_orders
16	April	2018	6939
17	August	2018	6512
18	February	2018	6728
19	January	2018	7269
20	July	2018	6292
21	June	2018	6167
22	March	2018	7211
23	May	2018	6873
24	October	2018	4
25	September	2018	16

Less no of orders

In Sights :

According to the results above, there has been a significant increase in the year 2018 exceptSeptember and October months in the year 2018.

There can be many factors which is letting in decrease of orders during these months and fewfactors can be:

1. The next consecutive months will have Black Friday and Christmas which mightmake the customers to wait to shop during that period.
2. People of Brazil might have anticipated the sales upcoming during the Novemberwhich might lead to decrease during these months.

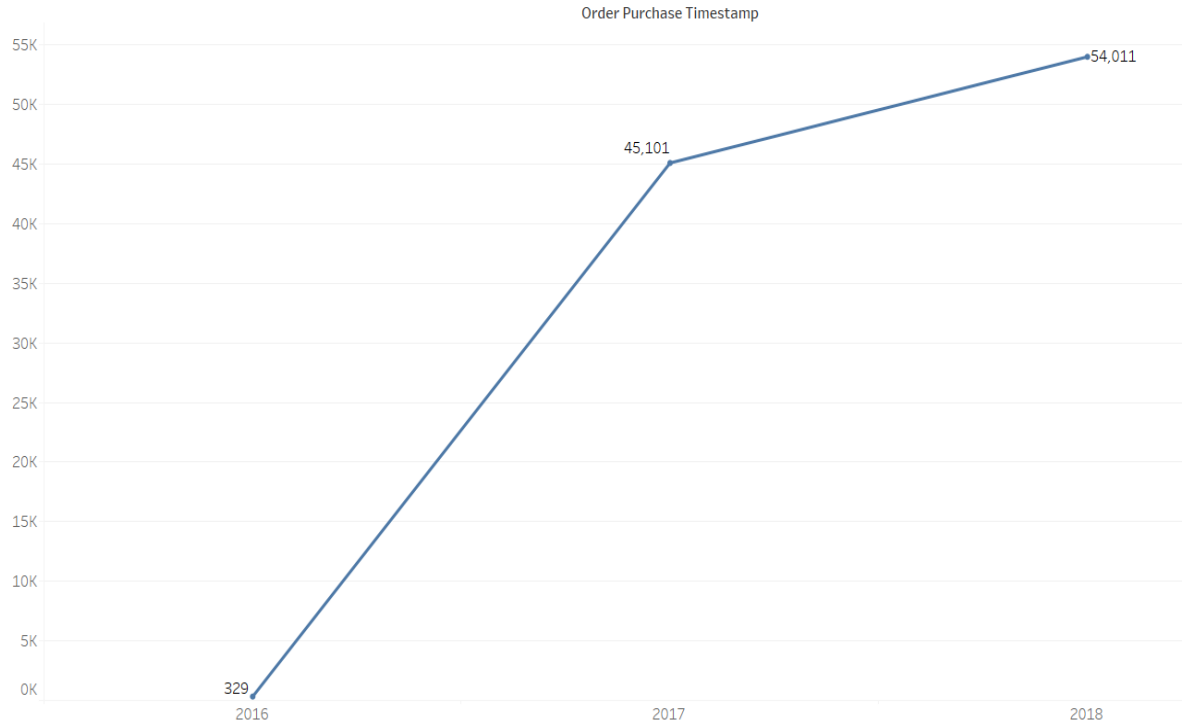
Recommendations:

By conducting this analysis, we can determine the growing trend and the factors that contributed towards this by getting into deeper insights of data.

Since the above data conveys that, the higher orders can be noticed during certain months hence the Ecommerce Retailers should increase their focus to attract people with discountsetc.

Assumptions:

Performed count on order id from orders table while considering the month and year from order purchase timestamp



The above line chart depicts the growth in terms of no of orders from the year 2016 to 2018

2. Can we see some kind of monthly seasonality in terms of the no. of orders being placed?

Query:

```
select monthyear , no_of_orders from

(select monthyear,db.no_of_orders as no_of_orders,
ntile(3) over( order by db.no_of_orders desc) as higher_orders
from
(select format_date("%B-%Y", order_purchase_timestamp) as monthyear ,
count(distinctorder_id) as no_of_orders
from `target-394104.Target_brazil.orders`
group by monthyear
order by no_of_orders)db)tb
where higher_orders = 1
order by tb.no_of_orders desc
```

Output:

Query results

JOB INFORMATION | RESULTS | JSON | EXECU

Row	monthyear	no_of_orders
1	November-2017	7544
2	January-2018	7269
3	March-2018	7211
4	April-2018	6939
5	May-2018	6873
6	February-2018	6728
7	August-2018	6512
8	July-2018	6292
9	June-2018	6167

In Sights

The above results show that the orders are at peak during certain months and the no of Orders declined in the year 2018(used NTILE to display the month year with highest orders)

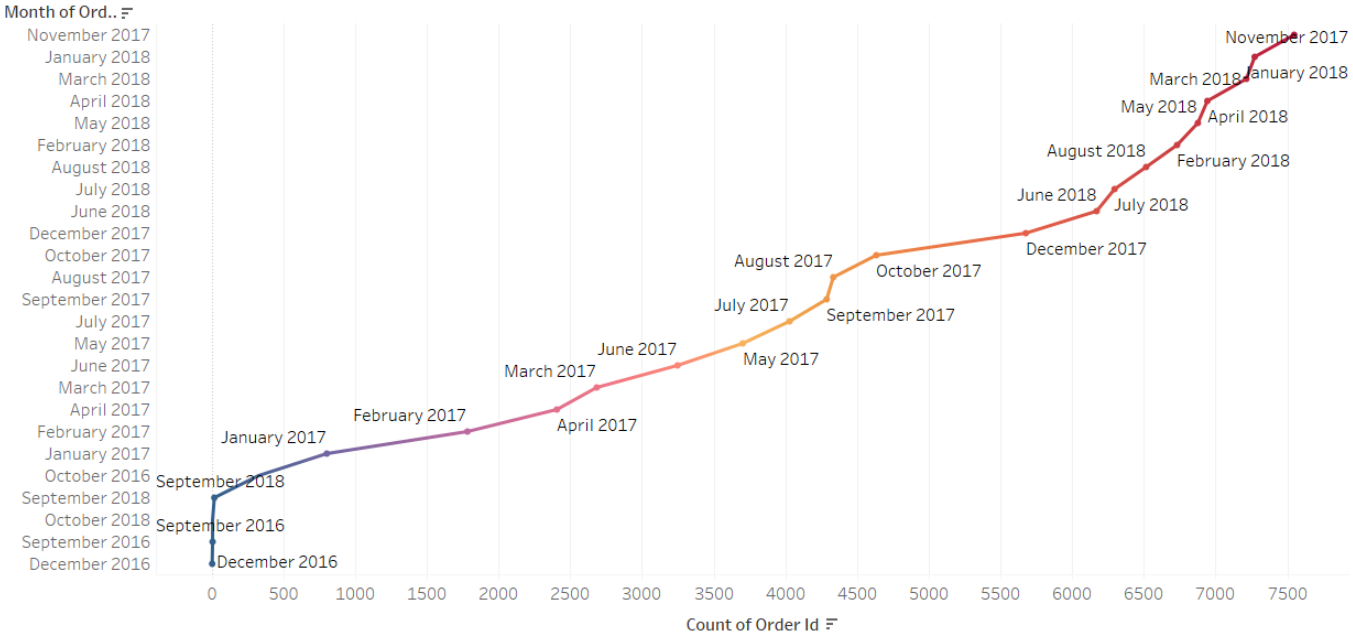
Recommendations

Analyzing monthly seasonality data in terms of orders being placed will let us know the various factors involved for seasonal variations so that the business can be placed accordinglyfor peak time

Assumptions

The count on order_id will give us no of orders with distinct keyword associated with dateusing extract keyword.

Monthly seasonality in terms of No Of Orders placed



From the above graph, Nov 2017 has higher no of orders in comparison to other months.

In the year 2018, January has highest no of orders in comparison to other months.

3. During what time of the day, do the Brazilian customers mostly place their orders ?(Dawn, Morning, Afternoon, Night)

Query:

```
with cte as
(
select extract(hour from order_purchase_timestamp) as hours
from `target-394104.Target_brazil.orders`
),
```

```
display as(
select hours,
case when hours between 0 and 6
      then 'Dawn'
    when hours between 7 and 12
      then 'Mornings'
    when hours between 13 and 18
      then 'Afternoon'
      else
       'Night'
      End as orders_high
      from cte
)
select orders_high,count(orders_high) as total_count
from display
group by orders_high
order by total_count desc
```

Output

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION

Row	orders_high	total_count
1	Afternoon	38135
2	Night	28331
3	Mornings	27733
4	Dawn	5242

Insights

From the results, it is evident that the orders are high during Afternoons then Nights later.It can be a case where people in the Brazil might get some time during the work days which might make them to order the products.

Recommendations

Since the orders are high during Afternoon's, identifying the potential customers and offeringAfternoon deals or discounts would make an add on and entice more no of customers duringthis time. People would love deals and promotions, so leveraging this can attract more customers.

Assumptions:

I extracted the hours from the order purchase timestamp since these hours indicate the hours where the orders are purchased.

III. Evolution of E-commerce orders in the Brazil region:

1. Get the month-on-month no. of orders placed in each state.

Query:

```
select c.customer_state as state,extract(month from order_purchase_timestamp) as
month,count(c.customer_id) as count_of_orders
from `target-394104.Target_brazil.customers` c
join `target-394104.Target_brazil.orders` o
on c.customer_id = o.customer_id
group by c.customer_state, month
order by state,month,count_of_orders
```

Output:

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	state	month	count_of_orders
1	AC	1	8
2	AC	2	6
3	AC	3	4
4	AC	4	9
5	AC	5	10
6	AC	6	7
7	AC	7	9
8	AC	8	7
9	AC	9	5
10	AC	10	6

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

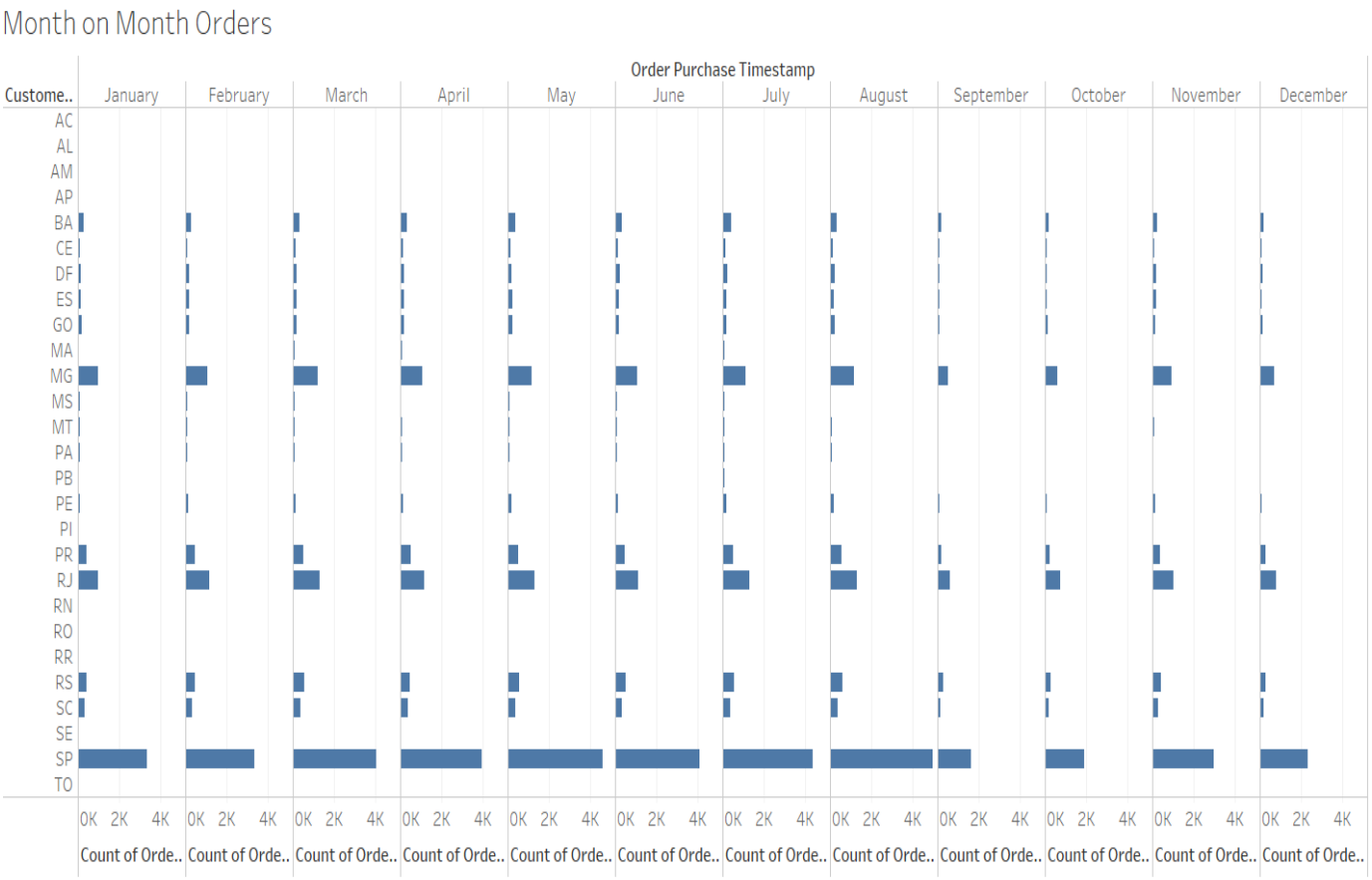
Row	state	month	count_of_orders
11	AC	11	5
12	AC	12	5
13	AL	1	39
14	AL	2	39
15	AL	3	40
16	AL	4	51
17	AL	5	46
18	AL	6	34
19	AL	7	40
20	AL	8	34

Insights:
The data above shows the no of orders for each month for each state.
Certain months have higher number of orders while certain months have lower orders.

The reason for lower orders can be due to multiple factors and the weather conditions alsoplay a role here.

Recommendations:
The higher number of orders during certain months might be due to Holidays,
Events and Weather conditions and the focus should be on increase in the demand during these periods.

Assumptions:
I considered the customer_id col in this output since customer_id is the one who has placed the orders.



2. How are the customers distributed across all the states?

Query:

```
select c.customer_state as state,count( c.customer_unique_id) as unique_customers
from `target-394104.Target_brazil.customers` c
group by state
order by unique_customers
```

Output:

Query results

JOB INFORMATION | RESULTS | JSON | EX

Row	state	unique_customers
1	RR	46
2	AP	68
3	AC	81
4	AM	148
5	RO	253
6	TO	280
7	SE	350
8	AL	413
9	RN	485
10	PI	495

Insights:
The above data depicts the number of unique customers for each state.

Recommendations:

Identifying the high customers will let a business concentrate on increase of high marketingStrategies and promotions to region specific

Assumptions:
In order to consider the no of unique customers we have to consider the unique_customer_id

IV Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others.

1.* Get the % increase in the cost of orders from year 2017 to 2018 *(include months between Jan to Aug only)

Query:

```
with cte2 as
(
select round(sum(case when extract(year from o.order_purchase_timestamp) = 2017 AND
extract(month from o.order_purchase_timestamp) BETWEEN 1 AND 8
then    payment_value else 0 end),2) as c_2017,
round(sum(case when extract(year from o.order_purchase_timestamp) = 2018 AND
extract(month from o.order_purchase_timestamp) BETWEEN 1 AND 8 then payment_value
else 0 end),2) as c_2018
```

```
from `target-394104.Target_brazil.orders` AS o JOIN `target-
394104.Target_brazil.payments` AS p
ON o.order_id = p.order_id
)
select c_2017 as cost_of_orders_2017,c_2018 as cost_of_orders_2018,((c_2018-
c_2017)/c_2017)*100 as per_of_increment
from cte2
```

Output:

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	cost_of_orders_2017	cost_of_orders_2018	per_of_increment
1	3669022.12	8694733.84	136.9768716466...

Insights:

The cost_of_orders are higher in 2018 than 2017 which indicates that the demand for the company's products or services has increased from 2017 to 2018.This suggests that the company's sales has been profitable in comparison to 2017

Recommendations:

Analyzing customer behaviour and preferences during 2018 to determine if there were any changes that led to larger orders.

2. Calculate the Total & Average value of order price for each state.

Query:

```
select c.customer_state as  customer_state,round(sum(price),2) as Total_price,round(avg(price),2) as
AVG_price
from `target-394104.Target_brazil.order_items` oi
join `target-394104.Target_brazil.orders` o
on p.order_id = o.order_id
join `target-394104.Target_brazil.customers` c
on o.customer_id = c.customer_id
group by c.customer_state
order by customer_state
```

Output:

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	customer_state	total_price	AVG
1	AC	19680.62	234.29
2	AL	96962.06	227.08
3	AM	27966.93	181.6
4	AP	16262.8	232.33
5	BA	616645.82	170.82
6	CE	279464.03	199.9
7	DF	355141.08	161.13
8	ES	325967.55	154.71

Insights:

Comparing total and average prices across various states helps us in understanding the regional price variations. Understanding the price differences helps us to adjust the market demands considering the local cost of living and preferences.

Recommendations:

Higher prices in certain states might indicate stronger demand for certain products in thoseregions. Business can be expanded in these regions.

3. Calculate the Total & Average value of order freight for each state.

Query:

```
select c.customer_state as  customer_state,round(sum(freight_value),2)  as
total_freight_value,round(avg(freight_value),2) as AVG
from `target-394104.Target_brazil.orders` o
join `target-394104.Target_brazil.order_items` oi
on o.order_id = oi.order_id
join `target-394104.Target_brazil.customers` c
on o.customer_id = c.customer_id
group by c.customer_state
order by 2 desc
```

Output:

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	customer_state	total_freight_value	AVG
1	SP	718723.07	15.15
2	RJ	305589.31	20.96
3	MG	270853.46	20.63
4	RS	135522.74	21.74
5	PR	117851.68	20.53
6	BA	100156.68	26.36
7	SC	89660.26	21.47
8	PE	59449.66	32.92
9	GO	53114.98	22.77
10	DF	50625.5	21.04

Insights:

A decrease in Total freight value can indicate the measures are taken within particular statesfor handling the freight charges.

A higher Freight values indicate can be due to many factors like Transportation charges,distance, demand etc.

Recommendations:

A lower freight values may also indicate that the demand for certain products might have reduced and this can be due to seasonal changes etc. The deeper insights have to be done to getthese relevant details.

Assumptions:

Considering the order details table to get the summation of freight values.

V. Analysis based on sales, freight and delivery time.

1. Find the no. of days taken to deliver each order from the order's purchase date as delivery time.

Query:

```
select distinct order_id,
date_diff(extract(date from order_delivered_customer_date),extract(date from
order_purchase_timestamp),day) as time_to_deliver,
abs(date_diff(extract(date from order_estimated_delivery_date),extract(date from
order_delivered_customer_date),day)) as diff_estimated_delivery
from `Target_brazil.orders`
where order_delivered_customer_date is not null
order by order_id
```

Output:

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	order_id	time_to_deliver	diff_estimated_delive
1	00010242fe8c5a6d1ba2dd792...	7	9
2	00018f77f2f0320c557190d7a1...	16	3
3	000229ec398224ef6ca0657da...	8	14
4	00024acbcdf0a6daa1e931b03...	6	6
5	00042b26cf59d7ce69dfabb4e...	25	16
6	00048cc3ae777c65dbb7d2a06...	7	15
7	00054e8431b9d7675808bcb8...	8	17
8	000576fe39319847cbb9d288c...	5	16
9	0005a1a1728c9d785b8e2b08...	10	0
10	0005f50442cb953dcd1d21e1f...	2	19

In Sights:

The Time to Deliver section dictates the no of days taken to deliver the product from the purchase date.

The Diff Estimated Deliver displays the no of days the product has reached before the estimated date.

Recommendations:

Higher transit time helps us in understanding the network including the distribution centers, carriers etc. However, the longer transit time may also occur due to destination place being Remote hence choosing an Advanced routing system can help.

In addition to this, reaching the products to the customers before the given estimated datewould positively impact the overall customer delivery experience.

Assumptions:

The order_delivered_customer_date includes NULLS hence filtered them in the where clause.

2. Find out the top 5 states with the highest & lowest average freight value.

Query: Solved this using TWO approaches

```
select customer_state,avg,high_rnk from
(select c.customer_state as customer_state,
round(avg(freight_value),2) as avg,
dense_rank() over(order by avg(freight_value) desc) high_rnk,
dense_rank() over(order by avg(freight_value) asc) low_rnk
from `target-394104.Target_brazil.orders` o
```

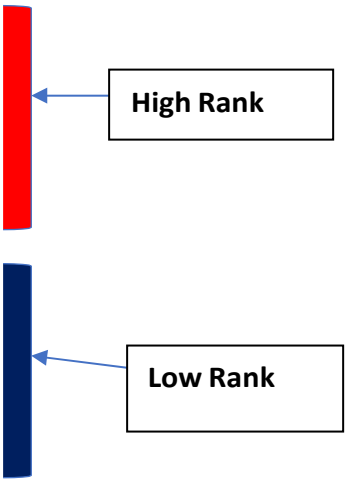
```
join `target-394104.Target_brazil.order_items` oi
on o.order_id = oi.order_id
join `target-394104.Target_brazil.customers` c
on o.customer_id = c.customer_id
group by customer_state)db
where high_rnk <= 5 or low_rnk <= 5
order by high_rnk,low_rnk
```

Output:

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	customer_state	avg	high_rnk
1	RR	42.98	1
2	PB	42.72	2
3	RO	41.07	3
4	AC	40.07	4
5	PI	39.15	5
6	DF	21.04	23
7	RJ	20.96	24
8	MG	20.63	25
9	PR	20.53	26
10	SP	15.15	27

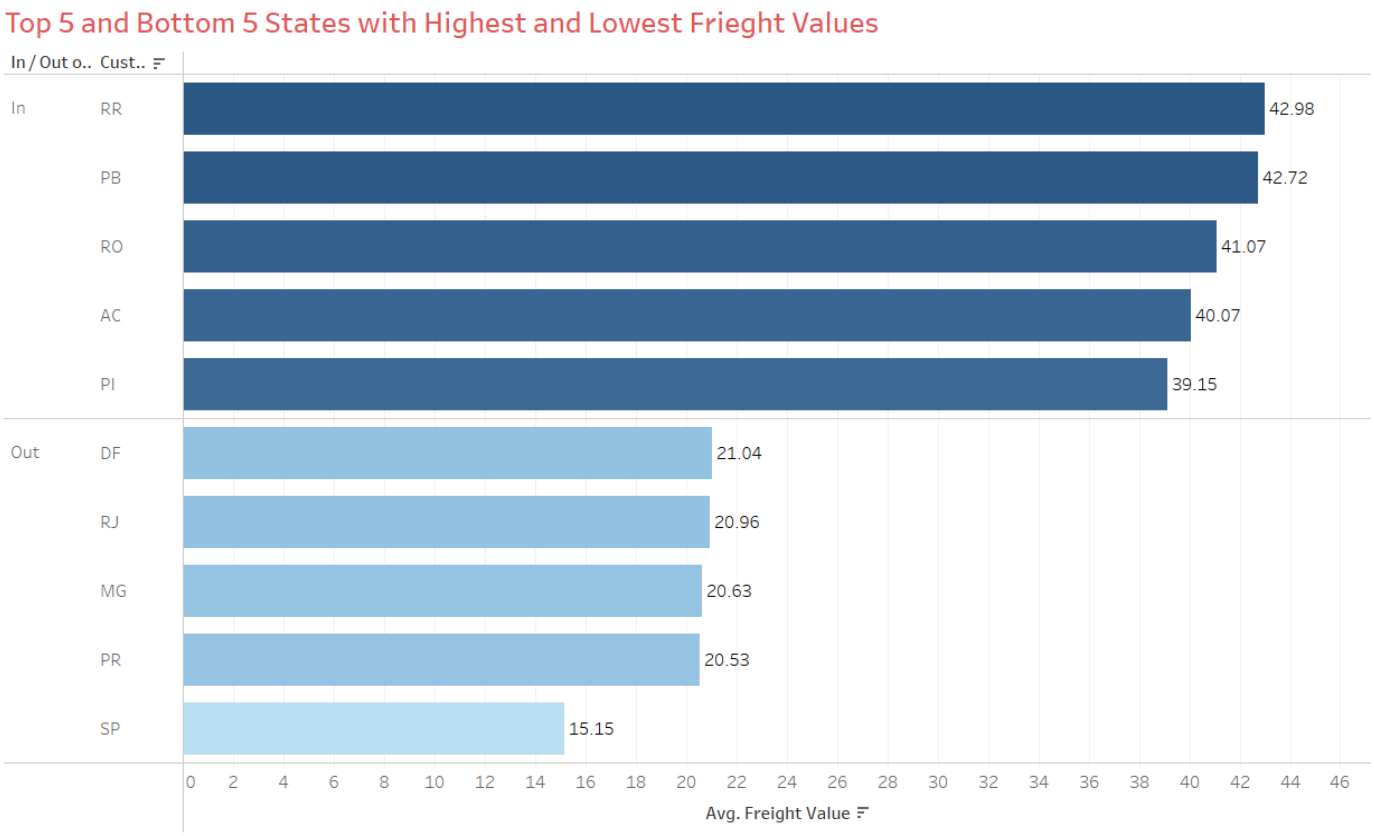


Insights:

The states with high Rank have Highest Freight values and the bottom states indicate States with lowest freight values.
The states with highest freight values indicate the well-developed transportation facility.

Recommendations:

As the states with highest freight values indicate the well-developed transportation facility hence these states shed light on employment opportunities related to transportation, warehousing etc.



The above graph depicts the Top 5 States with Highest and Lowest Freight Values. Information on freight values is integral to effective business planning, cost management, and the overall success of supply chain and logistics operations. It enables businesses to make informed decisions, improve efficiency, and stay competitive in a dynamic market.

3. Find out the top 5 states with the highest & lowest average delivery time.

Query:

```
with cte as
(
select
customer_state,round(avg(date_diff(order_delivered_customer_date,order_purchase_timesta
mp,day)),0) avge
```

```
from `target-394104.Target_brazil.orders` o
join `target-394104.Target_brazil.customers` c
on o.customer_id  = c.customer_id
group by customer_state
),
rank_avg as
(
select customer_state,avge,
row_number() over(order by avge desc) high_avg_rank,
row_number() over(order by avge ) low_avg_rank
from cte)
select customer_state,avge,low_avg_rank
from rank_avg
where high_avg_rank <= 5 or low_avg_rank <= 5
order by avge ,low_avg_rank
```

Output:

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	customer_state	avge	low_avg_rank
1	SP	8.0	1
2	MG	12.0	2
3	PR	12.0	3
4	DF	13.0	4
5	SC	14.0	5
6	PA	23.0	23
7	AL	24.0	24
8	AM	26.0	25
9	AP	27.0	26
10	RR	29.0	27

[1-5] Top Ranks for lesser no of avge

[23-27] Bottom ranks for higher avge

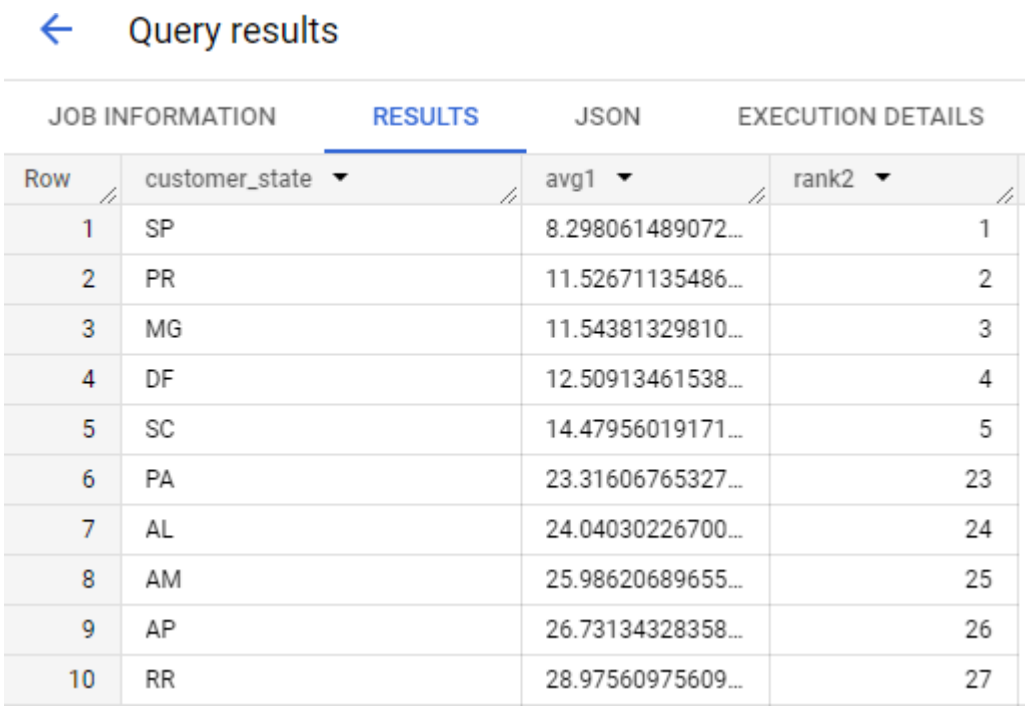
Another Approach with Dense Rank:

Query:

```
with cte as
(
select
customer_state,avg(date_diff(order_delivered_customer_date,order_purchase_timestamp,day
)) avg1
from `target-394104.Target_brazil.orders` o
```

```
join `target-394104.Target_brazil.customers` c
on o.customer_id  = c.customer_id
group by customer_state
),
rank_avg as
(
select customer_state,avg1,
dense_rank() over(order by avg1 desc) rank1,
dense_rank() over(order by avg1 ) rank2
from cte
)
select customer_state,avg1,rank2
from rank_avg
where rank1 <= 5 or rank2 <= 5
order by avg1 ,rank2
```

Output:



Row	customer_state	avg1	rank2
1	SP	8.298061489072...	1
2	PR	11.52671135486...	2
3	MG	11.54381329810...	3
4	DF	12.50913461538...	4
5	SC	14.47956019171...	5
6	PA	23.31606765327...	23
7	AL	24.04030226700...	24
8	AM	25.98620689655...	25
9	AP	26.73134328358...	26
10	RR	28.97560975609...	27

Less average time with top ranks and higher average time with bottom ranks

Insights:

The column with lesser average time are assigned with Top ranks and higherValues are given the bottom ranks. The same problem can be done using dense rank Function where the states with multiple ranks would be displayed but according to theQuestion as we are expected to display only top 5 hence Row number is used to display top 5

Recommendations:

The states with higher average delivery time indicate can lead to decreases customer satisfaction which can impact on business negatively.
In addition to this, the states with longer delivery time can be a loss in competition to States with least delivery time.

4. Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.

Query:

```
with cte as
(
select customer_state,o.order_status,
round(avg(date_diff(order_estimated_delivery_date,order_delivered_customer_date     ,day)),0)
as no_of_days_earlier
from `target-394104.Target_brazil.orders` o
join `target-394104.Target_brazil.customers` c
on o.customer_id = c.customer_id
where order_status = 'delivered'
group  by customer_state,order_status
),
ranking as
(
select customer_state,no_of_days_earlier,order_status,
dense_rank() over(order by no_of_days_earlier) Top_Rank
from cte
)
select customer_state,order_status,
no_of_days_earlier,
Top_Rank  from ranking
where Top_Rank <= 5
order by Top_Rank
```

Output:

Query results

SAVE RESI

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | CHART | PREVIEW | EX

Row	customer_state	order_status	no_of_days_earlier	Top_Rank
1	AL	delivered	8.0	1
2	MA	delivered	9.0	2
3	SE	delivered	9.0	2
4	SP	delivered	10.0	3
5	BA	delivered	10.0	3
6	ES	delivered	10.0	3
7	PI	delivered	10.0	3
8	CE	delivered	10.0	3
9	MS	delivered	10.0	3
10	GO	delivered	11.0	4

11	SC	delivered	11.0	4
12	RJ	delivered	11.0	4
13	DF	delivered	11.0	4
14	TO	delivered	11.0	4
15	PB	delivered	12.0	5
16	PE	delivered	12.0	5
17	PR	delivered	12.0	5
18	MG	delivered	12.0	5

Insights:

The above are the top 5 states that have a count of less days in terms of delivery time. Dense Rank would display all the states with equal ranks for the same value, whereas order by desc and limit would eliminate certain rows with similar values.

Recommendations:

Analyzing delivery performance during certain seasons or peak periods can shed light on thefactors like holidays etc. where companies can plan effectively during those periods.

If the delivery difference is more, it indicates that companies have to look onto the transportation network and get the deeper insights which are leading to customer dissatisfaction.

Assumptions:

Considered only the products where the status of the product is “delivered”

VI. Analysis based on the payments:

1. Find the month-on-month no. of orders placed using different payment types. Query:

```
select payment_type,
format_date("%B-%Y", order_purchase_timestamp) as monthyear,
count(p.order_id) as no_of_orders
from `target-394104.Target_brazil.payments` p
join `target-394104.Target_brazil.orders` o
on p.order_id = o.order_id
group by payment_type,monthyear
order by 2,3 desc
```

Output:

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | CHART

Row	payment_type	monthyear	no_of_orders
1	credit_card	April-2017	1846
2	UPI	April-2017	496
3	voucher	April-2017	202
4	debit_card	April-2017	27
5	credit_card	April-2018	5455
6	UPI	April-2018	1287
7	voucher	April-2018	370
8	debit_card	April-2018	97
9	credit_card	August-2017	3284
10	UPI	August-2017	938

Insights:

The above output dictates that the usage of credit card is more in comparison to other payment typesOver each month.

Recommendations:

Assessing the information behind certain payment types to be high would imply the customer behaviour. For e.g.: Do customers who use credit cards for payment tend to go for larger purchases? Different payment types over different months can be used to help identify the trends and changes incustomer preferences.

B. Find the no. of orders placed on the basis of the payment installments that have been paid.

Query:

```
select payment_installments,count(order_id) as no_of_orders_placed
from `target-394104.Target_brazil.payments`
where payment_installments <> 0
group by payment_installments
order by 1
```

Output:

Query results

JOB INFORMATION | RESULTS | JSON

Row	payment_installment	no_of_orders_placed
1	1	52546
2	2	12413
3	3	10461
4	4	7098
5	5	5239
6	6	3920
7	7	1626
8	8	4268
9	9	644

In Sights:

The payment_installement refers to a method of payment where the customer chooses to pay the entire amount with EMI options rather than paying the entire amount upfront.

Recommendations:

By analyzing the above data, we can get the customer preferences for flexible paymentoptions which in turn helps in boosting the sales.

Assumptions:

In order to get the above output, I considered the no of orders where payment installments are greater than 1