Q1 Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset.

1: Data type of all columns in the "customers" table.

Screenshot of Output:

Filter Enter property name or value

Field name	Type	Mode	Key	Collation	Default Value	Policy Tags	Description
customer_id	STRING	NULLABLE	-	-	-	-	-
customer_unique_id	STRING	NULLABLE	-	-	-	-	-
customer_zip_code_prefix	INTEGER	NULLABLE	-	-	-	-	-
customer_city	STRING	NULLABLE	-	-	-	-	-
customer_state	STRING	NULLABLE	-	-	-	-	-

Inference: There are 5 columns with their data type. The data type for customer_id , customer_unique_id ,customer_city and customer_state is **STRING** . And the data type for customer_zip_code_prefix is **INTEGER**.

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

Query:

```
SELECT
    MIN(DATE(order_purchase_timestamp)) AS first_date,
    MAX(DATE(order_purchase_timestamp)) AS last_date,
    TIME(MIN(order_purchase_timestamp)) AS first_time,
    TIME(MAX(order_purchase_timestamp)) AS last_time
from `Business_case.orders`
```

Screenshot of Output:

Row	first_date	last_date	first_time	last_time
1	2016-09-04	2018-10-17	21:15:19	17:30:18

Inference: From the output we can see that the first order date was 2016-09-04
The last order date was 2018-10-17
The first order time was 21:15:19
The last order time was 17:30:18

3: Count the Cities & States of customers who ordered during the given period.

Query:

```
SELECT count (distinct customer_city) as cities ,
count(distinct customer_state ) as states
from `Business_case.customers`
where customer_id in (select distinct customer_id from  `Business_case.orders`)
```

Screenshot of Output:

Row	cities	states
1	4119	27

Inference: From the result we can see that there are 4119 cities and 27 states of customers who ordered during the given period (2016 -2018).

Q2: In-depth Exploration:

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

Query:

```
select
extract(year from order_purchase_timestamp) as year,
extract(month from order_purchase_timestamp) as month,
count(order_id) as no_of_orders,
from `Business_case.orders`
group by 1,2
order by 1,2 ;
```

Screenshot of Output:

Row	year	month	no_of_orders
1	2016	9	4
2	2016	10	324
3	2016	12	1
4	2017	1	800
5	2017	2	1780
6	2017	3	2682
7	2017	4	2404
8	2017	5	3700
9	2017	6	3245
10	2017	7	4026
11	2017	8	4331
12	2017	9	4285
13	2017	10	4631
14	2017	11	7544
15	2017	12	5673

Inference:

From the output we can see that the no of orders for each month from 2016 to 2018. for the first 3 months of 2016 (months September, October, December) there was no consistent number of orders this is because the company was not established in Brazil and company was in starting phase of making their setup. Once the company well established there in the Brazil (from 2017 onwards), we can see there is growth in the number of orders and in November 2017 there is peak or company recorded highest number of orders (7544 orders) there can be multiple reasons why there was peak in November like there might be a Christmas sale, might be end of year sale, might be black Friday

sale etc. After then there is stable or consistency in the number of orders till august 2018, after that the number of orders fall the reason might be the company was wrapping up their operations. so, the answer is yes, there is a growing trend in the no. of orders placed over the past years.

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

Query:

```
select
extract(year from order_purchase_timestamp) as year,
extract(month from order_purchase_timestamp) as month,
count(order_id) as no_of_orders,
from `Business_case.orders`
group by 1,2
order by 1,2 ;
```

Screenshot of Output:

Row	year	month	no_of_orders
1	2016	9	4
2	2016	10	324
3	2016	12	1
4	2017	1	800
5	2017	2	1780
6	2017	3	2682
7	2017	4	2404
8	2017	5	3700
9	2017	6	3245
10	2017	7	4026
11	2017	8	4331
12	2017	9	4285
13	2017	10	4631
14	2017	11	7544
15	2017	12	5673

Inference:

Yes, we can see there is monthly seasonality in terms of the number of orders being placed, like when the company was setting up their operation there was no such a consistency in the number of orders, once company established or well settled in the market we can see the sudden growth in the

very first month 2017 and after that there is consistent growth in the number of orders for each month ,and in November 2017 we can see company recorded their highest numbers of order being placed there can be number of reasons behind it like Christmas sale, thanks giving sale etc. so company can predict there can be a more sale than usual months when there are some occasions like Christmas or any other. So, company should be ready with storage of more products, having more inventory , logistics, vendors, workers etc. after peak the next month there was slight drop in orders but then for next 5 months there was stable or constant number for the orders and then we can see the for the last two months of 2018 there was a huge drop in the orders as company was wrapping up their operations.

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

Query:

```
SELECT
COUNT(order_id) AS no_of_orders,
      CASE
       WHEN hr BETWEEN 0 AND 6 THEN  'Dawn'
       WHEN hr BETWEEN  7 AND 12 THEN  'Morning'
       WHEN  hr BETWEEN 13 AND 18 THEN  'Afternoon'
       WHEN  hr BETWEEN 19 AND 23 THEN  'Night'
      END AS Time_Interval
FROM
    (SELECT
     order_id,
     EXTRACT(HOUR FROM order_purchase_timestamp) AS hr
   from `Business_case.orders`)
   group by Time_Interval ;
```

Screenshot of Output:

Row	no_of_orders	Time_Interval
1	27733	Morning
2	5242	Dawn
3	38135	Afternoon
4	28331	Night

Inference: The above table gives the number of orders during different time interval Brazilian people orders. Afternoon is the time interval when Brazilian people orders the most as a count of

38135 orders. The number of orders peaked during afternoon suggest people prefers afternoon time to order. After that Night is the time when orders reduce as compare to afternoon but has a decent number of orders as people might orders after work. Morning time also has moderate number of orders placed and number of orders placed in Dawn time is relatively very low as compare to other times of the day.so Afternoon is the time of the day when Brazilian people orders most and during Dawn they orders less

Q3: Evolution of E-commerce orders in the Brazil region:

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

Query:

```
select count(o.order_id) as no_of_orders,
c.customer_state,
extract (month from o.order_purchase_timestamp) as month
from `Business_case.orders` o
inner join `Business_case.customers` c
on o.customer_id=c.customer_id
group by c.customer_state,month
order by c.customer_state,month asc;
```

Screenshot of Output:

Row	no_of_orders	customer_state	month
1	8	AC	1
2	6	AC	2
3	4	AC	3
4	9	AC	4
5	10	AC	5
6	7	AC	6
7	9	AC	7
8	7	AC	8
9	5	AC	9

Row	no_of_orders	customer_state	month
10	6	AC	10
11	5	AC	11
12	5	AC	12
13	39	AL	1
14	39	AL	2
15	40	AL	3
16	51	AL	4
17	46	AL	5
18	34	AL	6

Inference: Above output shows the number of orders placed in each customer state for each month. Like for State AC there are number of order given for each 12 months then for AL there are also number of orders given for each 12 months and so on for every 27 states for each 12 months from this pattern we can get numbers of orders for each month for a particular state if we find out which state orders most in what month then we have to be prepared for next year by having more storage, more products, vendors, logistics so that we can deliver best to the customers and also for that state which orders least in a particular month , we can run some kind of discount on products or have some kind of sale so that customers might get attracted .Also we can get number of orders for each month to each state, from we can study pattern of customers behaviour and be prepared for coming years

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

Query:

```
select customer_state,
count(distinct customer_id) as no_of_unique_customers
from `Business_case.customers`
group by customer_state;
```

Screenshot of Output:

Row	customer_state	no_of_unique_customers
1	RN	485
2	CE	1336
3	RS	5466
4	SC	3637
5	SP	41746
6	MG	11635
7	BA	3380
8	RJ	12852

Row	customer_state	no_of_unique_customers
9	GO	2020
10	MA	747
11	PE	1652
12	PB	536
13	ES	2033
14	PR	5045
15	RO	253

Inference: Above table shows the number of unique customers present in each state. Like for state RN there are 485 customers present,1336 customers are shopping in the state named CE. SP is the State showing most number of customers having number equals to 41746. And RR is the state having least customers having number equals to 46.since we have least customers in state RR we have to take some measurable actions to grow our sale there by having some marketing and promotion, by finding out customers need at least customers states and adding that product In the store, enhancing customer experience by providing them exceptional customer service, offering incentives like loyalty benefits and reward ,improve product quality can help company to grow in the states having least customers.

Q4: Impact on Economy: Analyse 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:

```
select Total_2017_payment_value,Total_2018_payment_value,
round(((Total_2018_payment_value-Total_2017_payment_value)/ Total_2017_payment_value * 100
),2) as increase_percentage from
(select
round(sum(case when order_purchase_timestamp between '2017-01-01' and '2017-08-31' then
payment_value else 0 end  ),2) as  Total_2017_payment_value ,
round(sum (case when order_purchase_timestamp between '2018-01-01' and '2018-08-31' then
payment_value else 0 end  ),2) as Total_2018_payment_value,
FROM Business_case.payments p
INNER JOIN Business_case.orders o ON p.order_id = o.order_id) a
```

Screenshot of Output:

Row	Total_2017_payment_value	Total_2018_payment_value	increase_percentage
1	3645107.27	8694669.95	138.53

Inference: The table presents the cumulative payment values from January to August for the years 2017 and 2018, along with the percentage increase in payment value. The total payment value for 2017 amounts to approximately 3,645,107.27, while for 2018, it reaches approximately 8,694,669.95. And the increase in payment value from 2017 to 2018, goes approximately 138.53%

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

Query:

```
Select sum(p.payment_value) as Total_price ,
avg(p.payment_value) as Average_price ,
c.customer_state
from `Business_case.customers` c inner join
`Business_case.orders` o
on c.customer_id = o.customer_id
inner join `Business_case.payments` p
on o.order_id = p.order_id
group by c.customer_state;
```

Screenshot of Output:

Row	Total_price	Average_price	customer_state
1	2144379.689999...	158.5258882235...	RJ
2	890898.5399999...	157.1804057868...	RS
3	5998226.959999...	137.5046297739...	SP
4	355141.0800000...	161.1347912885...	DF
5	811156.3799999...	154.1536259977...	PR
6	187029.2900000...	195.2289039665...	MT
7	152523.0200000...	198.8566101694...	MA
8	96962.05999999...	227.0774238875...	AL

Row	Total_price	Average_price	customer_state
9	1872257.260000...	154.7064336473...	MG
10	324850.4400000...	187.9921527777...	PE
11	75246.25	208.4383656509...	SE
12	218295.8499999...	215.9207220573...	PA
13	616645.8200000...	170.8160166204...	BA
14	279464.0299999...	199.9027396280...	CE
15	350092.3100000...	165.7634043560...	GO

Inference: The above table has 3 columns names as Total price ,Average price, customer state. Total_price shows the total price of orders for each state and Average_price shows the average price of orders for each state. For above table, the state RJ total price is around 2144379 and average price is around 158. The state named SP recorded highest total price around 5998226 and state RR has lowest total price around 10064. State PB has highest Average price around 248 and state SP has lowest Average price around 137.

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

Query:

```
select round(sum(freight_value),2) as Total_freight_value,
round(avg(freight_value),2) as Average_freight_value,
customer_state
from  `Business_case.customers` c inner join
`Business_case.orders` o
on c.customer_id = o.customer_id
inner join `Business_case.order_items` i
on o.order_id=i.order_id
group by c.customer_state
```

Screenshot of Output:

Row	Total_freight_value	Average_freight_value	customer_state
1	18860.1	35.65	RN
2	48351.59	32.71	CE
3	135522.74	21.74	RS
4	89660.26	21.47	SC
5	718723.07	15.15	SP
6	270853.46	20.63	MG
7	100156.68	26.36	BA
8	305589.31	20.96	RJ
Row	Total_freight_value	Average_freight_value	customer_state
9	53114.98	22.77	GO
10	31523.77	38.26	MA
11	59449.66	32.92	PE
12	25719.73	42.72	PB
13	49764.6	22.06	ES
14	117851.68	20.53	PR
15	11417.38	41.07	RO

Inference : The above table has 3 columns names as Total_freight_value ,Average_freight_value and customer_state.The Total_freight_value shows the total freight value of orders for each state and Average_freight_value shows the average freight value of orders for each state. For above table, the state SP has highest total freight value which is around 718723 and the state RR has the lowest freight value around 2235. The state named RR has highest Average freight value around 42.98 and state SP has lowest freight value which is around 15.15.

Q5: 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 order_id,
DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp,day) as time_to_deliver,
DATE_DIFF(order_delivered_customer_date, order_estimated_delivery_date,day) as
diff_estimated_delivery
from `Business_case.orders`
```

Screenshot of Output:

Row	order_id	time_to_deliver	diff_estimated_delivery
1	1950d777989f6a877539f53795b4c3c3	30	12
2	2c45c33d2f9cb8ff8b1c86cc28c11c30	30	-28
3	65d1e226dfaeb8cdc42f665422522d14	35	-16
4	635c894d068ac37e6e03dc54eccb6189	30	-1
5	3b97562c3aee8bdedcb5c2e45a50d5e1	32	0
6	68f47f50f04c4cb6774570cfde3a9aa7	29	-1
7	276e9ec344d3bf029ff83a161c6b3ce9	43	4
8	54e1a3c2b97fb0809da548a59f64c813	40	4
9	fd04fa4105ee8045f6a0139ca5b49f27	37	1

Row	order_id	time_to_deliver	diff_estimated_delivery
10	302bb8109d097a9fc6e9cefc5917d1f3	33	5
11	66057d37308e787052a32828cd007e58	38	6
12	19135c945c554eebfd7576c733d5ebdd	36	2
13	4493e45e7ca1084efcd38ddebf174dda	34	0
14	70c77e51e0f179d75a64a614135afb6a	42	11
15	d7918e406132d7c81f1b845276b03a3b	35	3
16	43f6604e77ce6433e7d68dd86db73b45	32	7
17	37073d851c3f30deebe598e5a586bdbd	31	9

Inference : Above table shows time to deliver and difference of estimated delivery for each order id. Time to deliver is the number of days taken to deliver each order to the customer from the order's purchase date and difference of estimated delivery is number of days between the estimated & actual delivery date of an order to the customer. We can see that there are many orders with

delivery date more than 30 days which is not good for any company as this will impact business, we have to reduce that delivery time. We should have as low as delivery date. We should take actions to reduce delivery time like maintain inventory to prevent stockouts, use of automation to handle orders quickly, partner up with logistics and various transportation, build distribution canters or warehouses to distinct locations etc. Also, we have to strictly avoid difference estimated date i.e. we have to avoid delay in delivery of product, this too can achieved by above actions , if we have given estimated date of delivery, we have to get that product delivered on or before the estimated delivery date to maintain growth in business. Also, we can see there are some negative values like order id 2c45c33d2f9cb8ff8b1c86cc28c11c30 (row 2) has diff of estimated delivery value as -28 this means the product was delivered 28 days before the estimated delivery date to the customer.

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

Query:

```
WITH cte AS (
    SELECT
        c.customer_state,
        ROUND(AVG(freight_value), 2) AS avg_freight
    FROM
        Business_case.customers c
    INNER JOIN
        Business_case.orders o ON c.customer_id = o.customer_id
    INNER JOIN
        Business_case.order_items i ON o.order_id = i.order_id
    GROUP BY
        c.customer_state
)

(SELECT
    customer_state  ,
    row_number()over( order by avg_freight ) as rank
    from cte
    limit 5)
union all
(SELECT
    customer_state,
    row_number()over( order by avg_freight desc) as rank
    from cte
    limit 5) ;
```

Screenshot of Output:

Row	customer_state	rank
1	RR	1
2	PB	2
3	RO	3
4	AC	4
5	PI	5
6	SP	1
7	PR	2
8	MG	3
9	RJ	4
10	DF	5

Inference : The above table gives the customer state where ranking is given to each state according to average freight value. First five states have highest average freight value the states are RR, PB, RO, AC, PI and next five states have lowest average freight value the states are SP, PR, MG ,RJ,DF.

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

Query:

```
with cte as
(select c.customer_state,
round(avg(DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp,day)),2) as
avg_delivery_days
from `Business_case.customers` c inner join
`Business_case.orders` o  on c.customer_id=o.customer_id
group by c.customer_state)

(select customer_state,
avg_delivery_days,
       row_number() over(order by cte.avg_delivery_days  DESC) as top_5_Bottom_5_rank
       from cte
       limit 5)
union all
(select customer_state,
avg_delivery_days ,
       row_number() over(order by cte.avg_delivery_days  ASC) as top_5_Bottom_5_rank
       from cte
       limit 5)
```

Screenshot of Output:

Row	customer_state	avg_delivery_days	top_5_Bottom_5_rank
1	SP	8.3	1
2	PR	11.53	2
3	MG	11.54	3
4	DF	12.51	4
5	SC	14.48	5
6	RR	28.98	1
7	AP	26.73	2
8	AM	25.99	3
9	AL	24.04	4
10	PA	23.32	5

Inference : The above table gives the view of top 5 states with lowest average delivery time and top 5 highest average delivery time. SP is the state leading as top state with lowest average delivery time of about almost 8 days. Customers in state SP gets the order delivered in almost 8 days whereas PA state customers have to wait of almost 23 days to get their order delivered.
States like SP, PR, MG, DF, SC have lowest delivery time. Customers in these states likely to receive their order within less days contributing higher satisfaction level. Also, Business operations in these states might have optimized logistics and delivery operations.
Whereas, States like RR, AP, AM, AL, PA have the highest delivery time. Customers in these states might experience longer wait times for their orders, and leading to decreased satisfaction and loyalty. This can impact business in such area, companies need to investigate causes of delay, such as logistical challenges, transportation issue etc. To improve delivery times in such states we need to be more focused towards improving logistics route, warehouse management or partner up with efficient delivery services etc.

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

Query:

```
select  c.customer_state,
avg(DATE_DIFF(order_delivered_customer_date, order_estimated_delivery_date,day))  as
order_delivery
from `Business_case.customers` c  inner join `Business_case.orders` o  on
c.customer_id=o.customer_id
group by c.customer_state
order by order_delivery
limit 5 ;
```

Screenshot of Output:

Row	customer_state	order_delivery
1	AC	-19.7625
2	RO	-19.1316872427...
3	AP	-18.7313432835...
4	AM	-18.6068965517...
5	RR	-16.4146341463...

Inference : Above table gives top 5 states where order delivery is really fast as compared to estimated delivery date. In sate AC order delivery is average 19 days before estimated delivery date and RR state customers get their product delivered 16 days before estimated delivery date.

Q6: Analysis based on the payments:

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

Query:

```
select count(o.order_id) as no_of_orders,p.payment_type,
extract (month from order_purchase_timestamp ) as month
from `Business_case.payments` p inner join
`Business_case.orders` o on p.order_id=o.order_id
group by 2,3
order by 3 asc;
```

Screenshot of Output:

Row	no_of_orders	payment_type	month
1	477	voucher	1
2	6103	credit_card	1
3	118	debit_card	1
4	1715	UPI	1
5	6609	credit_card	2
6	424	voucher	2
7	1723	UPI	2
8	82	debit_card	2
9	591	voucher	3
10	7707	credit_card	3
11	1942	UPI	3
12	109	debit_card	3
13	7301	credit_card	4
14	572	voucher	4
15	124	debit_card	4
16	1783	UPI	4
17	8350	credit_card	5
18	613	voucher	5

Inference : Above table shows the number of orders placed using different payment methods in each month . Like for 1st month i.e. January 477 orders were placed using voucher as a payment type

,6103 orders were placed using credit card,118 orders placed using debit card ,1715 orders placed using UPI as a payment type. From observation we can see that, most customers are using credit card as a payment type so we can offer them benefits like reward like giving points to the customers which they can use as discount etc. also we can see debit card is least used as payment type. To make customers use debit card more often we can give them benefits like cashback on using debit card, referrals like customers to refer friends and family to shopping using debit card, early bird discount offers to debit card users etc.

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

Query:

```
select count(order_id) no_of_orders,
            payment_installments
from `Business_case.payments`
where payment_installments > 0
group by payment_installments
```

Screenshot of Output:

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

Row	no_of_orders	payment_installments
10	5328	10
11	23	11
12	133	12
13	16	13
14	15	14
15	74	15
16	5	16
17	8	17
18	27	18

Inference : The table above shows the number of orders according to the number of payment installments completed. It indicates that there are 52,545 orders where a single installment has been completed, 12,413 orders with two installments completed, and so on.