1. **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.**

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					

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

```
select
min(order_purchase_timestamp) as min_order_purchase_timestamp,
max(order_purchase_timestamp) as max_order_purchase_timestamp,
from Target_Buz_case.orders;
```

Row	min_order_purchase_timestamp	max_order_purchase_timestamp
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC

Insight:
Here, first order is placed at 2016-09-04 and the is placed at 2018-10-17 , This is basically purchasing start 2016-09-04 and the purchasing ended at 2018-10-17 orders were placed.

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

```
select
count(distinct c.customer_city) as No_of_city,
count(distinct c.customer_state) as No_of_state
from Target_Buz_case.customers c join Target_Buz_case.orders
using(customer_id)
```

Row	No_of_city	No_of_state
1	4119	27

Insight:

Here the Number of cities are 4119 and The Number of States are 27. This are the count of customer who ordered during the given period

2. **In-depth Exploration:**

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

```
select count(order_id) as count_order,
extract(year from order_purchase_timestamp) as year
from Target_Buz_case.orders
group by 2
order by 2
```

Row	count_order	year
1	329	2016
2	45101	2017
3	54011	2018

Insight:
Here we can see that the sudden orders are increase in 2017 and gradually Increase as well in 2018

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

```
select
extract(month from order_purchase_timestamp) as month,
count(order_id) as count_order
from Target_Buz_case.orders
group by 1
order by 1;
```

Row	month	count_order
1	1	8069
2	2	8508
3	3	9893
4	4	9343
5	5	10573
6	6	9412
7	7	10318
8	8	10843
9	9	4305
10	10	4959

Insight:

Slightly higher count in may,July and August.

3. **During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)**
    a. **0-6 hrs : Dawn**
    b. **7-12 hrs : Mornings**
    c. **13-18 hrs : Afternoon**
    d. **19-23 hrs : Night**

```
select
sum( case when extract(hour from order_purchase_timestamp) between 0 and 6
then 1 else 0
end) as DAWN,

sum( case when extract(hour from order_purchase_timestamp) between 7 and
12 then 1 else 0
end) as MORNING,

sum( case when extract(hour from order_purchase_timestamp) between 13 and
18 then 1 else 0
end) as AFTERNOON,

sum( case when extract(hour from order_purchase_timestamp) between 19 and
23 then 1 else 0
end) as NIGHT
from Target_Buz_case.orders
```

Row	DAWN	MORNING	AFTERNOON	NIGHT
1	5242	27733	38135	28331

Insight:

Here we can easily see that Brazilian mostly purchase between 13:00 to 18:0 i.e.

afternoon

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

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

```
select
extract(year from order_purchase_timestamp) as year,
extract(month from order_purchase_timestamp) as month,
customer_state,
count(order_id) as cnt_orders
from Target_Buz_case.orders inner join
Target_Buz_case.customers using(customer_id)
group by 1,2,3
order by 1,2
limit 10;
```

Row	year	month	customer_state	cnt_orders
1	2016	9	RS	1
2	2016	9	RR	1
3	2016	9	SP	2
4	2016	10	SP	113
5	2016	10	MG	40
6	2016	10	GO	9
7	2016	10	CE	8
8	2016	10	SC	11
9	2016	10	RJ	56
10	2016	10	RS	24

Insight:
Here we can see that the count of orders respective to their state and year and month

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

```
select customer_state,
count(distinct customer_id) distinct_customer
from Target_Buz_case.customers
group by 1
order by 2
```

Row	customer_state	distinct_customer
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

Insight:

Here Each state contain distinct customer and distinct_customer contain the count of customer.

4. **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).**
   You can use the "payment_value" column in the payments table to get the cost of orders.

```
with cte as(
select sum(p.payment_value) as cost_of_orders,extract(year from
order_purchase_timestamp) as year
from Target_Buz_case.orders o join Target_Buz_case.payments p
on o.order_id=p.order_id
where extract(month from order_purchase_timestamp) between 1 and 8
group by extract(year from order_purchase_timestamp)
)
select *,round(((cost_of_orders-pr_count)/pr_count)*100,2) as incr_perct from(
select year,cost_of_orders,
lag(year) over(order by cte.year) as pr_year,
lead(cost_of_orders) over(order by cost_of_orders desc) as pr_count
from cte) temp_t
where temp_t.pr_count is not null
order by cost_of_orders desc;
```

Row	year	cost_of_orders	pr_year	pr_count	incr_perct
1	2018	8694733.839999...	2017	3669022.119999...	136.98

Here the percentage increase is nearly 137 %

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

```
select
customer_state,
round(sum(price),0) as sum_price_value,
round(avg(price),0) as avg_price_value
from Target_Buz_case.orders o
join Target_Buz_case.order_items oi using(order_id)
join Target_Buz_case.customers c on o.customer_id=c.customer_id
group by 1
limit 10;
```

Row	customer_state	sum_price_value	avg_price_value
1	MT	156454.0	148.0
2	MA	119648.0	145.0
3	AL	80315.0	181.0
4	SP	5202955.0	110.0
5	MG	1585308.0	121.0
6	PE	262788.0	146.0
7	RJ	1824093.0	125.0
8	DF	302604.0	126.0
9	RS	750304.0	120.0
10	SE	58921.0	153.0

Insight:
Here we can see that, sum of price and average of price
According to each state and ‘Sp’ price value is in comparison with other is low and the Highest average price is ‘Al’

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

```
select
customer_state,
round(sum(freight_value),0) as sum_freight_value,
round(avg(freight_value),0) as avg_freight_value
from Target_Buz_case.orders o
join Target_Buz_case.order_items oi using(order_id)
join Target_Buz_case.customers c on o.customer_id=c.customer_id
group by 1
```

Row	customer_state	sum_freight_value	avg_freight_value
1	MT	29715.0	28.0
2	MA	31524.0	38.0
3	AL	15915.0	36.0
4	SP	718723.0	15.0
5	MG	270853.0	21.0
6	PE	59450.0	33.0
7	RJ	305589.0	21.0
8	DF	50625.0	21.0
9	RS	135523.0	22.0
10	SE	14111.0	37.0

Insight:

Here we can see that, sum of freight value and average of freight value According to each state

5.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.**
   Also, calculate the difference (in days) between the estimated & actual delivery date of an order.
   Do this in a single query.

   You can calculate the delivery time and the difference between the estimated & actual delivery date using the given formula:
   - **time_to_deliver = order_delivered_customer_date - order_purchase_timestamp**
   - **diff_estimated_delivery = order_estimated_delivery_date - order_delivered_customer_date**

```
select order_id,
order_purchase_timestamp,
order_delivered_customer_date,
order_estimated_delivery_date,
date_diff(order_delivered_customer_date,order_purchase_timestamp,day) as
time_to_deliver,
date_diff(order_estimated_delivery_date,order_delivered_customer_date,day)as
diff_estimated_delivery
from Target_Buz_case.orders
```

Row	order_id	order_purchase_timestamp	order_delivered_customer_date	order_estimated_delivery_date	time_to_deliver	diff_estimated_delive
1	1950d777989f6a877539f5379...	2018-02-19 19:48:52 UTC	2018-03-21 22:03:51 UTC	2018-03-09 00:00:00 UTC	30	-12
2	2c45c33d2f9cb8ff8b1c86cc28...	2016-10-09 15:39:56 UTC	2016-11-09 14:53:50 UTC	2016-12-08 00:00:00 UTC	30	28
3	65d1e226dfaeb8cdc42f66542...	2016-10-03 21:01:41 UTC	2016-11-08 10:58:34 UTC	2016-11-25 00:00:00 UTC	35	16
4	635c894d068ac37e6e03dc54e...	2017-04-15 15:37:38 UTC	2017-05-16 14:49:55 UTC	2017-05-18 00:00:00 UTC	30	1
5	3b97562c3aee8bdedcb5c2e45...	2017-04-14 22:21:54 UTC	2017-05-17 10:52:15 UTC	2017-05-18 00:00:00 UTC	32	0
6	68f47f50f04c4cb6774570cfde...	2017-04-16 14:56:13 UTC	2017-05-16 09:07:47 UTC	2017-05-18 00:00:00 UTC	29	1
7	276e9ec344d3bf029ff83a161c...	2017-04-08 21:20:24 UTC	2017-05-22 14:11:31 UTC	2017-05-18 00:00:00 UTC	43	-4
8	54e1a3c2b97fb0809da548a59...	2017-04-11 19:49:45 UTC	2017-05-22 16:18:42 UTC	2017-05-18 00:00:00 UTC	40	-4
9	fd04fa4105ee8045f6a0139ca5...	2017-04-12 12:17:08 UTC	2017-05-19 13:44:52 UTC	2017-05-18 00:00:00 UTC	37	-1
10	302bb8109d097a9fc6e9cefc5...	2017-04-19 22:52:59 UTC	2017-05-23 14:19:48 UTC	2017-05-18 00:00:00 UTC	33	-5

Here, time_to_deliver column shows the difference that taken to deliver each order from the order's purchase date as delivery time.

In diff_estimated_delivery ,
we can see some negative,
That mean of -ve difference shows that the customer received order late as per expected date,
Where as +ve difference shows that the customer received order before as per expected date.

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

```
with cte as(
select
customer_state,
round(avg(freight_value),2)as average
from Target_Buz_case.orders o
join Target_Buz_case.order_items oi using(order_id)
join Target_Buz_case.customers c on o.customer_id=c.customer_id
group by 1
),
cte2 as(
select customer_state,average,
dense_rank()over(order by average desc ) as rnk_top_5,
dense_rank()over(order by average asc ) as rnk_lowest_5
from cte )

select  distinct a.customer_state,a.average,a.rnk_top_5,
            b.customer_state,b.average,b.rnk_lowest_5

from cte2 a join cte2 b
 on a.rnk_top_5=b.rnk_lowest_5
where a.rnk_top_5<=5 and b.rnk_lowest_5<=5
order by a.rnk_top_5 ,b.rnk_lowest_5;
```

Row	customer_state	average	rnk_top_5	customer_state_1	average_1	rnk_lowest_5
1	RR	42.98	1	SP	15.15	1
2	PB	42.72	2	PR	20.53	2
3	RO	41.07	3	MG	20.63	3
4	AC	40.07	4	RJ	20.96	4
5	PI	39.15	5	DF	21.04	5

I

Here is the Top 5 highest and lowest state with their Freight value.

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

```
with cte as (select customer_state,
round(avg(date_diff(order_delivered_customer_date,order_purchase_timestamp
,day)),2) avg_time_to_deliver
from Target_Buz_case.orders join Target_Buz_case.customers
using(customer_id)
group by customer_state),
cte2 as(
  select *,rank()over(order by avg_time_to_deliver desc ) as
top_5_avg_time_to_deliver,
         rank()over(order by avg_time_to_deliver asc ) as
lowest_5_avg_time_to_deliver from cte
  order by 3
)

select a.customer_state,a.avg_time_to_deliver,a.top_5_avg_time_to_deliver,
      b.customer_state,b.avg_time_to_deliver,b.lowest_5_avg_time_to_deliver

from cte2 a join cte2 b on
a.top_5_avg_time_to_deliver=b.lowest_5_avg_time_to_deliver
where a.top_5_avg_time_to_deliver<=5 and
b.lowest_5_avg_time_to_deliver<=5
order by 3,6
```

Row	customer_state	avg_time_to_deliver	top_5_avg_time_to_d	customer_state_1	avg_time_to_deliver	lowest_5_avg_time_t
1	RR	28.98	1	SP	8.3	1
2	AP	26.73	2	PR	11.53	2
3	AM	25.99	3	MG	11.54	3
4	AL	24.04	4	DF	12.51	4
5	PA	23.32	5	SC	14.48	5

Here is the top 5 highest and top 5 lowest state with their average time take to delver.

4. **Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.**
   You can use the difference between the averages of actual &

estimated delivery date to figure out how fast the delivery was for each state.

```
with cte as
(select customer_state,
round(avg(date_diff(order_delivered_customer_date,order_purchase_timestamp
,day)),2) as avg_delv_actual ,
round(avg(date_diff(order_estimated_delivery_date,order_delivered_customer_
date,day)),2) as avg_esti_dele
from Target_Buz_case.orders join Target_Buz_case.customers
using(customer_id)
group by customer_state
order by 2 desc ,3 desc
)

select customer_state,avg_delv_actual,avg_esti_dele,round((avg_delv_actual-
avg_esti_dele),2) diff_betweeen_actual_estimate
from cte
order by 4 desc
limit 5;
```

Row	customer_state	avg_delv_actual	avg_esti_dele	diff_betweeen_actua
1	AL	24.04	7.95	16.09
2	RR	28.98	16.41	12.57
3	MA	21.12	8.77	12.35
4	SE	21.03	9.17	11.86
5	CE	20.82	9.96	10.86

Insight:

Here we can easily see that top 5 states the order delivery is really fast as compared to the estimated date of delivery. The more difference that means the delivery done fast

6.Analysis based on the payments:

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

```
select count(order_id) as count_order,payment_type,
extract(year from order_purchase_timestamp) as year,
extract(month from order_purchase_timestamp) as month from
Target_Buz_case.payments join Target_Buz_case.orders using(order_id)
group by 2,3,4
order by 3,4
limit 10;
```

Row	count_order	payment_type	year	month
1	3	credit_card	2016	9
2	2	debit_card	2016	10
3	254	credit_card	2016	10
4	23	voucher	2016	10
5	63	UPI	2016	10
6	1	credit_card	2016	12
7	61	voucher	2017	1
8	197	UPI	2017	1
9	583	credit_card	2017	1
10	9	debit_card	2017	1

Insight:

Above, we can see that count of order using different payment type with month of each year

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

```
select count(order_id) as count_order,
payment_installments from Target_Buz_case.payments join
Target_Buz_case.orders using(order_id)
group by 2
having payment_installments>=1
limit 10;
```

Row	count_order	payment_installment
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
10	5328	10

Here the count of orders based on the number of installment and the count is slowly getting decrease