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

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

Answer:

```
SELECT column_name, data_type
FROM Target_SQL.INFORMATION_SCHEMA.COLUMNS
where table_name = 'customers'
```

JOB INFORMATION | RESULTS | CHART | PREVIEW

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:

We see that customer_id, customer_unique_id, customer_city, customer_state are "STRING" Data Type and customer_zip_code_prefix is of "INTEGER" Data Type.

1.B: Get the time range between which the orders were placed.

Answer:

```
select max(order_purchase_timestamp) as last_order,
       min(order_purchase_timestamp) as first_order
from `Target_SQL.orders`
```

JOB INFORMATION | RESULTS | CHART | PREVIEW | JSC

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

Insights:

From the data set, we see that the first order was made in 04.09.2016 and last order was made in 17.10.2018.

1.C. Count the Cities & States of customers who ordered during the given period.

Answer:

```
select count(distinct customer_city) as customer_city,
       count(distinct customer_state) as customer_state
from `Target_SQL.customers`
```

JOB INFORMATION | RESULTS | CHAR

Row	customer_city	customer_state
1	4119	27

Insights:

Customers who ordered are from 4119 cities in 27 states.

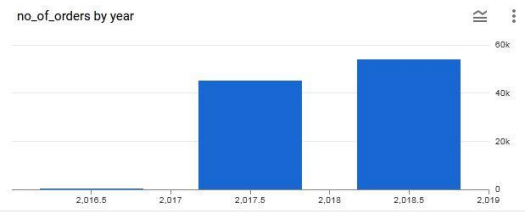
2. In-depth Exploration:

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

Answer:

```
select year,count(year) as no_of_orders
from( select *, extract(year from order_purchase_timestamp) as year
from `Target_SQL.orders`)a
group by year
order by year
```

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



Insights:

Yes, there a growing trend in the no. of orders placed over the past years.

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

Answer:

```
select month_name, month, year,count(month) as no_of_orders
from( select *, extract(month from order_purchase_timestamp) as month,
                extract(year from order_purchase_timestamp) as year,
                format_datetime('%b', order_purchase_timestamp) as month_name,
from `Target_SQL.orders`)a
group by 1,2,3
```

```
order by 2,3
```

Row	month_name	month	year	no_of_orders
1	Jan	1	2017	800
2	Jan	1	2018	7269
3	Feb	2	2017	1780
4	Feb	2	2018	6728
5	Mar	3	2017	2682
6	Mar	3	2018	7211
7	Apr	4	2017	2404
8	Apr	4	2018	6939

Insights:

Yes, we can see some kind of monthly seasonality in “Jan 2018 and March 2018” and “Feb 2018 and April 2018”

2.C. During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)
i) 0-6 hrs : Dawn ii) 7-12 hrs : Mornings iii) 13-18 hrs : Afternoon iv) 19-23 hrs : Night

Answer:

```
select purchase_time, count(purchase_time) as no_of_orders
from(select order_purchase_timestamp, case
                                when hour between 0 and 6 then 'Dawn'
                                when hour between 7 and 12 then 'Mornings'
                                when hour between 13 and 18 then 'Afternoon'
                                when hour between 19 and 23 then 'Night'
                                end as purchase_time
from(select order_purchase_timestamp, extract(hour from order_purchase_timestamp) as hour
from `Target_SQL.orders`)a)b
group by 1
order by 2 desc
```

purchase_time	no_of_orders
Afternoon	38135
Night	28331
Mornings	27733
Dawn	5242

Insights:

From the given data set we see Brazilian customers mostly place their orders in Afternoon then Night and Mornings. Very few only place their orders in Dawn.

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

3.A. Get the month on month no. of orders placed in each state.

Answer:

```
SELECT customer_state, extract (month from order_purchase_timestamp) as month,
        count(order_id) as no_of_orders
from `Target_SQL.customers` c
inner join `Target_SQL.orders` o
on c.customer_id = o.customer_id
group by 1,2
order by 1,2
```

Row	customer_state	month	no_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

Insights:

From the given data set we found the month on month orders placed in each state.

3.B. How are the customers distributed across all the states?

Answer:

```
select customer_state, count(c.customer_id) as no_of_customers
from `Target_SQL.customers` c
inner join `Target_SQL.orders` o
on c.customer_id = o.customer_id
group by 1
order by 2 desc
```

Row	customer_state	no_of_customers
1	SP	41746
2	RJ	12852
3	MG	11635
4	RS	5466
5	PR	5045
6	SC	3637
7	BA	3380
8	DF	2140
9	ES	2033

Insights:

From the given data set we found that maximum number of customers from SP state.

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

4.A. 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.

Answer:

```
with final as
(select *, lead(total_cost) over(order by year ) as prev_year
from(select extract(year from order_purchase_timestamp) as year,
        round(sum(payment_value),2) as total_cost
from `Target_SQL.orders` o
inner join `Target_SQL.payments` p
on p.order_id = o.order_id
where order_purchase_timestamp between '2017-01-01' and '2017-08-31' or
        order_purchase_timestamp between '2018-01-01' and '2018-08-31'
group by 1)a
order by year)

select year_2017, year_2018, round(((year_2018 - year_2017)/ year_2017)*100,2) as
percentage
from(select sum(case when year = 2017 then total_cost end) as year_2017,
           sum(case when year = 2018 then total_cost end) as year_2018
           from final)b
```

Row	year_2017	year_2018	percentage
1	3645107.27	8694669.95	138.53

Insights:

From the given data set we found that 138% increase in the cost of orders from year 2017 to 2018.

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

Answer:

```
select customer_state, round(sum(price),2) as total_price, round(avg(price),2) as
average_price
from `Target_SQL.customers` c
inner join `Target_SQL.orders` ord
on ord.customer_id = c.customer_id
inner join `Target_SQL.order_items` o
on o.order_id = ord.order_id
group by 1
order by 1
```

Row	customer_state	total_price	average_price
1	AC	15982.95	173.73
2	AL	80314.81	180.89
3	AM	22356.84	135.5
4	AP	13474.3	164.32
5	BA	511349.99	134.6
6	CE	227254.71	153.76
7	DF	302603.94	125.77
8	ES	275037.31	121.91
9	GO	294591.95	126.27
10	MA	119648.22	145.2

Insights:

From the given data set we found Total price and Average price of product for each state.

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

Answer:

```
select distinct customer_state,
round(sum(freight_value) over(partition by customer_state),2) as total_freight_price,
round(avg(freight_value) over(partition by customer_state),2) as average_freight_price
from `Target_SQL.customers` c
inner join `Target_SQL.orders` ord
on ord.customer_id = c.customer_id
inner join `Target_SQL.order_items` o
on o.order_id = ord.order_id
order by 1
```

Row	customer_state	total_freight_price	average_freight_pric
1	AC	3686.75	40.07
2	AL	15914.59	35.84
3	AM	5478.89	33.21
4	AP	2788.5	34.01
5	BA	100156.68	26.36
6	CE	48351.59	32.71
7	DF	50625.5	21.04
8	ES	49764.6	22.06
9	GO	53114.98	22.77

Insights:

From the given data set we found Total Freight Price and Average Freight Price of product for each state.

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

5.A. 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

Answer:

```
select timestamp_diff(order_delivered_customer_date, order_purchase_timestamp, day) as
time_to_deliver,
       timestamp_diff(order_estimated_delivery_date, order_delivered_customer_date, day)
as diff_estimated_delivery
from `Target_SQL.orders`
```

Row	time_to_deliver	diff_estimated_delive
1	30	-12
2	30	28
3	35	16
4	30	1
5	32	0
6	29	1
7	43	-4
8	40	-4
9	37	-1
10	33	-5

Insights:

From the given data set we found Delivered time and Difference in estimated delivery time.

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

Answer:

```
select (a.customer_state) as highest_avg_freight_state, a.highest_avg_freight_value,
       (b.customer_state) as lowest_avg_freight_state, b.lowest_avg_freight_value
from (select customer_state, round(avg(freight_value),2) as highest_avg_freight_value,
    row_number() over(order by round(avg(freight_value),2) asc) as rnk
from `Target_SQL.customers` c
inner join `Target_SQL.orders` ord
on ord.customer_id = c.customer_id
inner join `Target_SQL.order_items` o
on o.order_id = ord.order_id
group by 1
order by highest_avg_freight_value desc
limit 5) a
```

```
inner join
(select customer_state, round(avg(freight_value),2) as lowest_avg_freight_value,
      row_number() over(order by round(avg(freight_value),2) desc) rnk
from `Target_SQL.customers` c
inner join `Target_SQL.orders` ord
on ord.customer_id = c.customer_id
inner join `Target_SQL.order_items` o
on o.order_id = ord.order_id
group by 1
order by lowest_avg_freight_value
limit 5) b
on a.rnk = b.rnk
```

Row	highest_avg_freight_state	highest_avg_freight_	lowest_avg_freight_state	lowest_avg_freight_v
1	RR	42.98	SP	15.15
2	PB	42.72	PR	20.53
3	RO	41.07	MG	20.63
4	AC	40.07	RJ	20.96
5	PI	39.15	DF	21.04

Insights:

From the given data set we found Five Highest Avg Freight State and Five Lowest Avg Freight State.

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

Answer:

```
select (a.customer_state) as highest_avg_time_deliver_state,
a.highest_average_time_deliver,
        (b.customer_state) as lowest_avg_time_deliver_state, b.lowest_average_time_deliver
from(select customer_state, round(avg(time_to_deliver),2) as highest_average_time_deliver,
row_number() over(order by round(avg(time_to_deliver),2) desc) as rnk
from(select customer_state, timestamp_diff(order_delivered_customer_date,
order_purchase_timestamp, day) as time_to_deliver
from `Target_SQL.customers` c
inner join `Target_SQL.orders` ord
on ord.customer_id = c.customer_id
inner join `Target_SQL.order_items` o
on o.order_id = ord.order_id)a
group by 1
order by 2 desc
limit 5)a
inner join
(select customer_state, round(avg(time_to_deliver),2) as lowest_average_time_deliver,
row_number() over(order by round(avg(time_to_deliver),2) asc) as rnk
from(select customer_state, timestamp_diff(order_delivered_customer_date,
order_purchase_timestamp, day) as time_to_deliver
from `Target_SQL.customers` c
inner join `Target_SQL.orders` ord
on ord.customer_id = c.customer_id
inner join `Target_SQL.order_items` o
on o.order_id = ord.order_id)a
```

```
group by 1
order by 2
limit 5)b
on a.rnk =b.rnk
```

Row	highest_avg_time_deliver_state	highest_average_tim	lowest_avg_time_deliver_state	lowest_average_time
1	RR	27.83	SP	8.26
2	AP	27.75	PR	11.48
3	AM	25.96	MG	11.52
4	AL	23.99	DF	12.5
5	PA	23.3	SC	14.52

Insights:

From the given data set we found Five Highest Avg Time Delivery State and Five Lowest Avg Time Delivery State.

5.D. 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.

Answer:

```
with final as
(select
c.customer_state,o.order_id,ord.order_estimated_delivery_date,ord.order_delivered_customer_
date,
from `Target_SQL.customers` c
inner join `Target_SQL.orders` ord
on ord.customer_id = c.customer_id
inner join `Target_SQL.order_items` o
on o.order_id = ord.order_id )

select customer_state,
round(avg(date_diff(order_estimated_delivery_date, order_delivered_customer_date,day)),2)
as fast_delivery_state
from final
group by 1
order by 2 desc
limit 5
```

Row	customer_state	fast_delivery
1	AC	20.01
2	RO	19.08
3	AM	18.98
4	AP	17.44
5	RR	17.43

Insights:

From the given data set we found Five Fastest Delivery State.

6. Analysis based on the payments:

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

Answer:

```
SELECT p.payment_type, extract (month from order_purchase_timestamp) as month,
extract (year from order_purchase_timestamp) as year, count(o.order_id) as no_of_orders
from `Target_SQL.orders` o
inner join `Target_SQL.payments` p
on o.order_id = p.order_id
group by 1,2,3
order by 2
```

Row	payment_type	month	year	no_of_orders
1	credit_card	1	2018	5520
2	credit_card	1	2017	583
3	UPI	1	2018	1518
4	voucher	1	2018	416
5	UPI	1	2017	197
6	debit_card	1	2018	109
7	voucher	1	2017	61
8	debit_card	1	2017	9
9	UPI	2	2018	1325

Insights:

From the given data set we found most of the purchase was done by Credit card only.

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

Answer:

```
select payment_installments, count(o.order_id) as no_of_orders
from `Target_SQL.orders` o
inner join `Target_SQL.payments` p
on o.order_id = p.order_id
where payment_installments >0
group by 1
order by 1
```

Row	payment_installment	no_of_orders
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
10	10	5328

Insights:

From the given data set we found the installments that have been paid.