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**

A) Code :

```
select column_name, data_type

from `arched-alpha-452814-b3.TARGET.INFORMATION_SCHEMA.COLUMNS`

where TABLE_NAME = 'customers';
```

B) Output :

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

C) Insights :

- Only the customer zip code has integer data type, rest all columns have string data type.
- Customer id is the primary key.

D) Recommendation :

- NA

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

A) Code :

```
select min(order_purchase_timestamp) as first_order_timestamp, max(order_purchase_timestamp) as last_order_timestamp, date_diff(max(order_purchase_timestamp),min(order_purchase_timestamp),day) as range_days

from `TARGET.orders`
```

B) Output :

Row	first_order_timestamp	last_order_timestamp	range_days
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC	772

C) Insights :

- The orders are placed between September 2016 to October 2018 with a range of 772 days.

D) Recommendation :

- NA

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

A) Query :

```
with table1 as(
select o.customer_id, c.customer_city,c.customer_state
from `TARGET.customers` c inner join `TARGET.orders` o
on c.customer_id = o.customer_id)
select count(distinct customer_city) as count_city, count(distinct customer_state) as count_state
from table1
from table1
```

B) Output :

Row	count_city	count_state
1	4119	27

C) Insights :

- There are 4119 cities and 27 states of customer base who ordered during the given period.

D) Recommendation :

- We can explore new cities and states to expand the business.

2. In-depth Exploration:

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

A) Query :

```
with table1 as(
select extract(year from order_purchase_timestamp) as year,extract(month from order_purchase_timestamp)
as month, count(order_id) as cnt
from `TARGET.orders`
group by 1,2
order by year)
select distinct year, round(avg(cnt) over(partition by year),2) as Avg_orders_per_Month
from table1
order by year
```

B) Output :

Row	year	Avg_orders_per_Month
1	2016	109.67
2	2017	3758.42
3	2018	5401.1

C) Insights :

- As we can see in the output table, the average number of orders per month is increasing from 2016 to 2017 by 3648 and from 2017 to 2018 by 1643. There is a growing trend in the number of orders placed over the past years.
- From year 2016 to 2017 shown 34x jump which shows better marketing, better products introduction, region expansion, better customer retention.
- From year 2017 to 2018 shows additional grown which represent sustained improvement in operations.

D) Recommendations :

- The growing trend in number of orders over the year is good sign for the business. Keep continuing the good work.
- Ensure infrastructure and inventory to fulfil future grown to improve customer satisfaction. Need to predict future growth to ensure sufficient facilities.

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

A) Query :

```
with table1 as(

select extract(year from order_purchase_timestamp) as year,extract(month from order_purchase_timestamp) as months, count(order_id) as cnt

from `TARGET.orders`

group by 1,2

order by year,months)


select t3.months, t2.cnt as Orders_2016,t3.cnt as Orders_2017,t4.cnt as Orders_2018

from (

 select months, cnt

 from table1

 where year = 2016

) t2 full outer join

(
```

```
 select months, cnt
 from table1
 where year = 2017
) t3
on t2.months = t3.months
full outer join
(
 select months, cnt
 from table1
 where year = 2018
) t4
on t3.months = t4.months
order by t3.months
```

B) Output :

Row	months	Orders_2016	Orders_2017	Orders_2018
1	1	null	800	7269
2	2	null	1780	6728
3	3	null	2682	7211
4	4	null	2404	6939
5	5	null	3700	6873
6	6	null	3245	6167
7	7	null	4026	6292
8	8	null	4331	6512
9	9	4	4285	16
10	10	324	4631	4
11	11	null	7544	null
12	12	1	5673	null

C) Insights:

- We can observe consistent growth in number of orders in 2017 and peaking November which extends up to first quarter of 2018, which shown holiday season sales and year end promotions.
- In 2018, first quarter numbers are high but it drops in the second and third quarter, which shows reduced market demand or external disruptions.
- Huge margin of increase in orders in 2017 and early 2018, need significant scaling in operations. Any inefficiency could result in steep decline in later time of 2018.

D) Recommendations:

- Need to investigate the later part of 2018 due to reduced order volumes based on, market trends, customer feedback, operational challenges etc.
- Given that sudden increase of orders in November, plan on seasonal marketing, inventory planning and efficient operation during high demand period.
- Boost the promotions in peak time like November and January.

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**

A) Query:

```
select case when extract(hour from order_purchase_timestamp) between 0 and 6 then "Dawn"
        when extract(hour from order_purchase_timestamp) between 7 and 12 then "Mornings"
        when extract(hour from order_purchase_timestamp) between 13 and 18 then "Afternoon"
        when extract(hour from order_purchase_timestamp) between 19 and 23 then "Night" end as time_of_day, count(order_id) as No_of_Orders
from `TARGET.orders`
group by 1
order by No_of_Orders desc
```

B) Output:

Row	time_of_day	No_of_Orders
1	Afternoon	38135
2	Night	28331
3	Mornings	27733
4	Dawn	5242

C) Insights:

- Brazilian customers are most active in the afternoon time (13 to 18 hrs), with 38135 orders much higher than other times of the day. Possibly due to lunch break free time or convenient shopping time.
- Night and Mornings are almost same in terms of no of orders, which shown people equally spend time for shopping whenever they find themselves free.
- Dawn time is least ordered time with 5242 orders, which makes sense, as most of the customers might be taking rest.

D) Recommendations:

- Focus advertisements, promotions offers and flash sales on the afternoon time that is 13 to 18 Hrs, to leverage maximum engagement of customers.
- Do experimental offers or deals in dawn time to test it on less number of customers.
- Plan some offers for morning and night time of the day, to increase the orders on those times so that through day, we can get consistent and high number of orders.

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

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

A) Query:

```
select c.customer_state, extract( month from o.order_purchase_timestamp) as month,  count(o.order_id) as Total_Orders

from TARGET.customers c inner join TARGET.orders o on

c.customer_id = o.customer_id

group by 1,2

order by customer_state, month
```

B) Output:

Row	customer_state	month	Total_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

C) Insights:

- Table shows monthly seasonal orders for each state, which shows whether the states are growing consistently which can be flagged for further regional investments.
- Table shows regional holiday effect for every state, so that to plan separately for each state season wise for inventory, marketing, and operations.
- Table shows performance of each state which helps to shift the focus time to time.

D) Recommendations:

- Based on monthly order patterns, identify high performing states to expand inventory and distribution centres for faster delivery.
- Target low performing states to find root cause of it.
- Use month on month data to find seasonal trend to plan for high demand months for better customer experience.

2) **How are the customers distributed across all the states?**

A) Query:

```
select customer_state, count(distinct customer_unique_id) as Total_Customers

from `TARGET.customers`

group by 1

order by 2 desc
```

B) Output:

Row	customer_state	Total_Customers
1	SP	40302
2	RJ	12384
3	MG	11259
4	RS	5277
5	PR	4882
6	SC	3534
7	BA	3277
8	DF	2075
9	ES	1964
10	GO	1952

C) Insights:

- SP state is leading with 40302 orders whereas RR is at the bottom with 45 orders which shows huge regional differences in e commerce adoption.
- High order states are generating highest orders means it already has strong distribution system and inventory. Whereas low order states has room for improvement in terms of marketing and awareness.

D) Recommendations:

- Conduct marketing campaigns and e commerce awareness in states with lower customer presence.
- Promote with additional discount in the lower customer presence states, to get people used to ecommerce services.
- Analyse customer buying preferences according to region, to give better product suggestions to particular customer.

4. 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).**

A) Query:

```
with table1 as(

select round(sum(p.payment_value),2) as Cost_2017
```

```
from `TARGET.orders` o inner join `TARGET.payments` p

on o.order_id = p.order_id

where extract(year from o.order_purchase_timestamp) = 2017 and extract(month from
o.order_purchase_timestamp) in (1,2,3,4,5,6,7,8)),

table2 as(

select round(sum(p.payment_value),2) as Cost_2018

from `TARGET.orders` o inner join `TARGET.payments` p

on o.order_id = p.order_id

where extract(year from o.order_purchase_timestamp) = 2018 and extract(month from
o.order_purchase_timestamp) in (1,2,3,4,5,6,7,8)

)

select *, round(100*(table2.Cost_2018-table1.Cost_2017)/table1.Cost_2017,2) as percent_increase

from table1,table2
```

B) Output:

Row	Cost_2017	Cost_2018	percent_increase
1	3669022.12	8694733.84	136.98

C) Insights:

- The total payment value for first 8 months of 2018 is 136.98% higher than in 2017, which shows strong business growth, high value purchases and high customer engagement.
- This growth in cost of orders is could be because of expanded customer base, new marketing strategies, improved customer service.

D) Recommendations:

- With growing customer base and order value, need to improve service quality, streamlined shipping, inventory, faster distribution is required.
- Expand marketing strategies and distribution regions and customer retention strategies.
- Need to investigate the seasonal trend to plan for high demand in orders for a particular time.

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

A) Query:

```
select c.customer_state, round(sum(ot.price),2) as total_Order_Price, round(avg(ot.price),2) as
avg_Order_Price

from `TARGET.customers` c inner join `TARGET.orders` o

on c.customer_id = o.customer_id

inner join `TARGET.order_items` ot

on o.order_id = ot.order_id

group by 1

order by total_Order_Price desc
```

B) Output:

Row	customer_state	total_Order_Price	avg_Order_Price
1	SP	5202955.05	109.65
2	RJ	1824092.67	125.12
3	MG	1585308.03	120.75
4	RS	750304.02	120.34
5	PR	683083.76	119.0
6	SC	520553.34	124.65
7	BA	511349.99	134.6
8	DF	302603.94	125.77
9	GO	294591.95	126.27
10	ES	275037.31	121.91

c) Insights:

- State SP showing the highest overall total order price with 5202955.05, which shows high purchasing power of customer of state or customer engagement. Whereas state PB is having highest average order price 191.48, which shows, this region might prefer higher value purchases compared to other.
- States like RR, AP, AC having lowest total order price and average order price, shows these regions facing lower customer engagement or economical constrains affecting purchase power.

D) Recommendations:

- Need to focus on low total and average order price states to improve customer engagement by personalized marketing, additional discount to increase order prices.
- Improving the services and offering premium products to states having higher order price to retain the customer with high purchase power to further increase the order price.

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

A) Query:

```
select c.customer_state, round(sum(ot.freight_value),2) as total_freight_price, round(avg(ot.freight_value),2) as avg_freight_price

from `TARGET.customers` c inner join `TARGET.orders` o

on c.customer_id = o.customer_id

inner join `TARGET.order_items` ot

on o.order_id = ot.order_id

group by 1

order by total_freight_price desc
```

B) Output:

Row	customer_state	total_freight_price	avg_freight_price
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

C) Insights:

- State SP leads with highest total freight price of 718723 with average freight price of only 15 which shows higher order volumes.
- States such as RR, AC, RO, PB have higher average freight price (>40), which shows lower delivery density or long delivery routes.

D) Recommendations:

- In case of RR, AC, RO, PB, look for different option of logistics and carrier partnership to reduce freight expenses.
- States with lowest total freight prices, need to be relooked to improve customer engagement, might need to run localised promotional offers and discounts.

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_delivered_customer_date - order_estimated_delivery_date**

A) 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 `TARGET.orders`

where order_delivered_customer_date is not null

order by time_to_deliver desc
```

B) Output:

Row	order_id	time_to_deliver	diff_estimated_deliv
1	ca07593549f1816d26a572e06...	209	181
2	1b3190b2dfa9d789e1f14c05b...	208	188
3	440d0d17af552815d15a9e41a...	195	165
4	285ab9426d6982034523a855f...	194	166
5	0f4519c5f1c541ddec9f21b3bd...	194	161
6	2fb597c2f772eca01b1f5c561b...	194	155
7	47b40429ed8cce3aee9199792...	191	175
8	2fe324febf907e3ea3f2aa9650...	189	167
9	2d7561026d542c8dbd8f0daea...	188	159
10	c27815f7e3dd0b926b5855262...	187	162

C) Insights:

- Highest delivery time for an order in 209 days where as very few orders have delayed the delivery by more than 100 days, which signifies significant logistic disruptions.
- The system is predicting over optimistic delivery time and failed to consider real world delays.

D) Recommendations:

- Need to refine estimated delivery time prediction with considering all real world constrains.
- Analyse logistic and shipment performance at the locations of significant delays to improve the process.
- Need to examine patterns behind extreme delays and also enhance customer communication to reduce customer dissatisfaction.

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

B) Query:

```
with table1 as(

select customer_state, rank() over(order by avg_frt_value desc) as top_rank, rank() over(order by avg_frt_value) as bottom_rank,avg_frt_value

from (select distinct c.customer_state, avg(ot.freight_value) over(partition by c.customer_state) as avg_frt_value

from `TARGET.customers` c inner join `TARGET.orders` o

on c.customer_id = o.customer_id

inner join `TARGET.order_items` ot

on o.order_id = ot.order_id) as x),

table2 as(

select customer_state, row_number() over(order by top_rank) as row_number_2, avg_frt_value

from table1
```

```
where top_rank <= 5

),

table3 as(

select customer_state, row_number() over(order by bottom_rank) as row_number_3,avg_frt_value

from table1

where bottom_rank <= 5

)


select distinct t2.customer_state as Highest_avg_freight_States,round(t2.avg_frt_value,2) as
Highest_avg_freight_value, t3.customer_state as  Lowest_avg_freight_States,round(t3.avg_frt_value,2) as
Lowest_avg_freight_values

from table2 t2 inner join table3 t3

on t2.row_number_2 = t3.row_number_3
```

B) Output:

Row	Highest_avg_freight_States	Highest_avg_freight_value	Lowest_avg_freight_States	Lowest_avg_freight_values
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

C) Insights:

- RR, PB, RO, AC and PI are the top 5 states with highest average freight rates, which signifies the logistic difficulties such as long delivery distance and sparse infrastructure.
- SP, PR, MG,RJ and DF are top 5 states with lowest average freight rates, which signifies likely to have higher order volumes and better developed logistics network, leading to more efficient shipping operations.

D) Recommendations:

- Optimize logistics at the highest freight rate states by establishing regional distribution centres to reduce shipping distances and overall freight costs
- Maximize advantage in states, SP, PR, MG, RJ and DF by increasing inventory and shipping throughput.

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

A) Query:

```
with table1 as(
```

```
select *, rank() over(order by avg_del_time desc) as top_rank, rank() over(order by avg_del_time) as bottom_rank

from(

select distinct c.customer_state,
avg(date_diff(o.order_delivered_customer_date,o.order_purchase_timestamp,day)) over(partition by c.customer_state) as avg_del_time

from `TARGET.customers` c inner join `TARGET.orders` o

on c.customer_id = o.customer_id

where o.order_status = "delivered"

order by avg_del_time desc) as x)

select t2.customer_state as Highest_Delivery_time_State, round(t2.Average_delivery_time,2) as Highest_Avg_delivery_time, t3.customer_state as Lowest_Delivery_time_State,
round(t3.Average_delivery_time,2) as Lowest_Avg_delivery_state

from(

(select customer_state,avg_del_time as Average_delivery_time, row_number() over(order by top_rank) as row_number1

from table1

where top_rank <= 5

order by top_rank) t2

inner join

(select customer_state,avg_del_time as Average_delivery_time, row_number() over(order by bottom_rank) as row_number1

from table1

where bottom_rank <= 5

order by bottom_rank) t3

on t2.row_number1 = t3.row_number1)
```

B) Output:

Row	Highest_Delivery_time_State	Highest_Avg_delivery_time	Lowest_Delivery_time_State	Lowest_Avg_delivery_state
1	RR	28.98	SP	8.3
2	AP	26.73	PR	11.53
3	AM	25.99	MG	11.54
4	AL	24.04	DF	12.51
5	PA	23.32	SC	14.48

C) Insights:

- The highest delivery time states are, RR, AP, AM, AL and PA, these regions are likely affected by logistic challenges such as long shipping distance, limited infrastructure, lower order volumes etc.
- The lowest delivery time states are, SP, PR, MG, DF, SC, these states benefit from strong logistic system, higher order volumes, leading to faster deliveries.

D) Recommendations:

- Invest in local shipment centres in regions like RR and AP to improve the logistic challenges to reduce the delivery distance.
- Leverage efficient regions like SP and PR for faster expansion by keeping them as high distribution hubs.
- To encourage purchases in slow delivery time states, introduce discounts on shipping charges.

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.

A) Query:

```
select distinct customer_state, round(avg(date_diff_1) over(partition by customer_state),2) as avg_fastest_delivery

from(

select c.customer_state, date_diff(o.order_estimated_delivery_date,o.order_delivered_customer_date,day) as date_diff_1, o.order_estimated_delivery_date,o.order_delivered_customer_date

from `TARGET.customers` c inner join `TARGET.orders` o

on c.customer_id = o.customer_id) as x

order by avg_fastest_delivery desc

limit 5
```

B) Output:

Row	customer_state	avg_fastest_delivery
1	AC	19.76
2	RO	19.13
3	AP	18.73
4	AM	18.61
5	RR	16.41

C) Insights:

- The states AC, RO and AP have early delivery by almost 19 days as compared to estimated delivery, which signifies the delivery time prediction system is over conservative and the orders are getting delivered much sooner than estimated.
- There are consistent early deliveries in regions, like, AC, RO, AP, AM, RR which signifies the logistic efficiencies outperforming initial expectations.

D) Recommendations:

- Need to refine estimated delivery prediction based on the efficiency of logistics and historic delivery trends, to align more closely to actual delivery dates.
- Advertise the faster deliveries in the region listed above to attract the customers.

- Need to provide more accurate real time status of order updates to let the customer know and plan accordingly.

6. Analysis based on the payments:

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

A) Query:

```
select distinct format_timestamp("%Y-%m",o.order_purchase_timestamp) as orderMonth, p.payment_type,
count(o.order_id) over(partition by format_timestamp("%Y-
%m",o.order_purchase_timestamp),p.payment_type) as Total_orders

from `TARGET.orders` o inner join `TARGET.payments` p

on o.order_id = p.order_id

order by 1,3 desc
```

B) Output:

Row	orderMonth	payment_type	Total_orders
1	2016-09	credit_card	3
2	2016-10	credit_card	254
3	2016-10	UPI	63
4	2016-10	voucher	23
5	2016-10	debit_card	2
6	2016-12	credit_card	1
7	2017-01	credit_card	583
8	2017-01	UPI	197
9	2017-01	voucher	61
10	2017-01	debit_card	9

C) Insights:

- The data shows that for almost every month, the most preferred payment type for orders in credit card and then followed by UPI, with peak orders in November 2017 using credit card.
- Voucher is usage is lower side but it is consistent almost every month
- Debit card shows lowest transaction volumes among all payment types.
- There a little growth in orders with payment type UPI month on month.
- A significant increase in overall orders across all payment types observed in November 2017.

D) Recommendations:

- Focus on credit card marketing and collaborate with credit card companies to offer exclusive discounts, cashback, and rewards to strengthen customer engagement.
- As there is growing market for UPI payment type, boost the adotion to UPI for future expansion of customer base.

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

A) Query:

```
select distinct p.payment_installments, count(o.order_id) over(partition by p.payment_installments) as Total_orders

from `TARGET.orders` o inner join `TARGET.payments` p

on o.order_id = p.order_id

order by 2 desc
```

B) Output:

Row	payment_installment	Total_orders
1	1	52546
2	2	12413
3	3	10461
4	4	7098
5	10	5328
6	5	5239
7	8	4268
8	6	3920
9	7	1626
10	9	644

C) Insights:

- Customers prefer to pay upfront rather paying in instalments which shown in table as 52546 orders for 1 instalment.
- As the number instalment increases, the number of order goes on reducing.
- Payment instalments of 2 to 4 are still popular as compared higher number of instalments.

D) Recommendations:

- Give discounts or offers for upfront payers to gain more customer base, as this is the most preferred payment type.
- Develop targeted promotion for 2 to 4 instalment range customers, such as no cost EMI which is no interest EMI, to increase customer engagement and increase the number of orders.
- Track the price and product based EMI preferences by customer, to attract future customer by promoting those customer choices for specific products.