```
# 1.1 Data type of all columns in the "customers" table.

SELECT
  column_name,
  data_type
FROM
  `target-case-study-389213.Retail.INFORMATION_SCHEMA.COLUMNS`;
```

OUTPUT : Showing the data types of all the features which are present in the data

Row	column_name	data_type
1	seller_id	STRING
2	seller_zip_code_prefix	INT64
3	seller_city	STRING
4	seller_state	STRING
5	order_id	STRING
6	order_item_id	INT64
7	product_id	STRING
8	seller_id	STRING
9	shipping_limit_date	TIMESTAMP
10	price	FLOAT64

```
# 1.2 Get the time range between which the orders were placed.

SELECT min(order_purchase_timestamp) min_Date , max(order_purchase_timestamp)
as max_date from `Retail.Orders` ;
```

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

Insights:

The "min_Date" value represents the earliest recorded purchase timestamp in the dataset.
The "max_date" value represents the latest recorded purchase timestamp in the dataset.
By analyzing the time range between the minimum and maximum dates, you can determine the overall duration of the order records in the "Retail.Orders" table.

```
# 1.3 Count the number of Cities and States in our dataset.

WITH CITY_AND_STATES AS (

SELECT customer_city as CITY , customer_State as STATE from `Retail.Customers`

UNION ALL

select geolocation_city as CITY , geolocation_State as STATE from
`Retail.Geolocation`

UNION ALL

select seller_city as CITY , seller_state as STATE from `Retail.Sellers`

)

select count(DISTINCT CITY) as num_of_cities , count(DISTINCT STATE) as
num_of_states from CITY_AND_STATES ;
```

Row	num_of_cities	num_of_states
1	8126	27

Insights:

"num_of_cities" represents the count of distinct cities present in the combined dataset.
"num_of_states" represents the count of distinct states present in the combined dataset.
By analyzing these counts, you can gain insights into the diversity of cities and states across the customers, geolocation, and sellers data in the "Retail" database.

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

```
Select extract(year from order_purchase_timestamp) as YEAR ,

count(order_id) as Order_Count from `Retail.Orders`

group by YEAR

ORDER BY YEAR ;
```

Row	YEAR	Order_Count
1	2016	329
2	2017	45101
3	2018	54011

Insights:

The "YEAR" column represents the extracted year from the "order_purchase_timestamp."
The "Order_Count" column represents the count of orders for each year.
By analyzing the results, you can observe the distribution of order counts across different years, allowing you to identify any trends or patterns in the order volumes over time in the "Retail.Orders" table.

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

```
SELECT extract(year from order_purchase_timestamp) as YEAR , extract(month from order_purchase_timestamp) as MONTH ,

count(order_id) as Order_Count from `Retail.Orders`

group by YEAR , MONTH

order by Order_count DESC ;
```

Row	YEAR	MONTH	Order_Count
1	2017	11	7544
2	2018	1	7269
3	2018	3	7211
4	2018	4	6939
5	2018	5	6873
6	2018	2	6728
7	2018	8	6512
8	2018	7	6292
9	2018	6	6167

Monthly Order Volumes: The query provides a breakdown of order counts by year and month. By examining the results, you can identify which specific months had the highest and lowest order volumes. This insight can be helpful for understanding seasonal trends or identifying peak periods of customer activity.

Yearly Order Trends: The query allows for an analysis of order volumes over multiple years. By observing the order counts for each year, you can gain insights into overall growth or decline in sales over time. This information

can assist in strategic decision-making, such as identifying successful years or periods of expansion.

Popular Months: By sorting the results in descending order based on order count, you can determine the most popular months in terms of order volumes. This knowledge can aid in planning marketing campaigns, allocating resources efficiently during busy periods, and ensuring sufficient inventory availability during peak months.

```
#2.3 During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)
/* 0-6 hrs : Dawn
7-12 hrs : Mornings
13-18 hrs : Afternoon
19-23 hrs : Night */

select *, CASE

WHEN time between 0 and 6 then 'Dawn'

WHEN time between 7 and 12 then 'Mornings'

WHEN time between 13 and 18 then 'Afternoon'

WHEN time between 19 and 23 then 'Night'

END as time_phase

from (select extract(Hour from order_purchase_timestamp) as time ,
count(order_id) as Orders_Count from `Retail.Orders`

group by extract(Hour from order_purchase_timestamp) )

order by time ;
```

Row	time	Orders_Count	time_phase
1	0	2394	Dawn
2	1	1170	Dawn
3	2	510	Dawn
4	3	272	Dawn
5	4	206	Dawn
6	5	188	Dawn
7	6	502	Dawn
8	7	1231	Mornings
12	11	6578	Mornings
13	12	5995	Mornings
14	13	6518	Afternoon
15	14	6569	Afternoon
16	15	6454	Afternoon
17	16	6675	Afternoon
18	17	6150	Afternoon
19	18	5769	Afternoon
20	19	5982	Night

By analyzing the results, you can observe the distribution of order counts across different hours of the day and identify which time phases have the highest or lowest order volumes. This information can be useful for understanding customer behavior patterns, optimizing staffing levels during peak times, and tailoring marketing strategies based on specific time phases.

```
#3.1 Get the month on month no. of orders placed in each state.

SELECT extract(MONTH from o.order_purchase_timestamp) as month ,
count(o.order_id) as Order_count ,c.customer_state
```

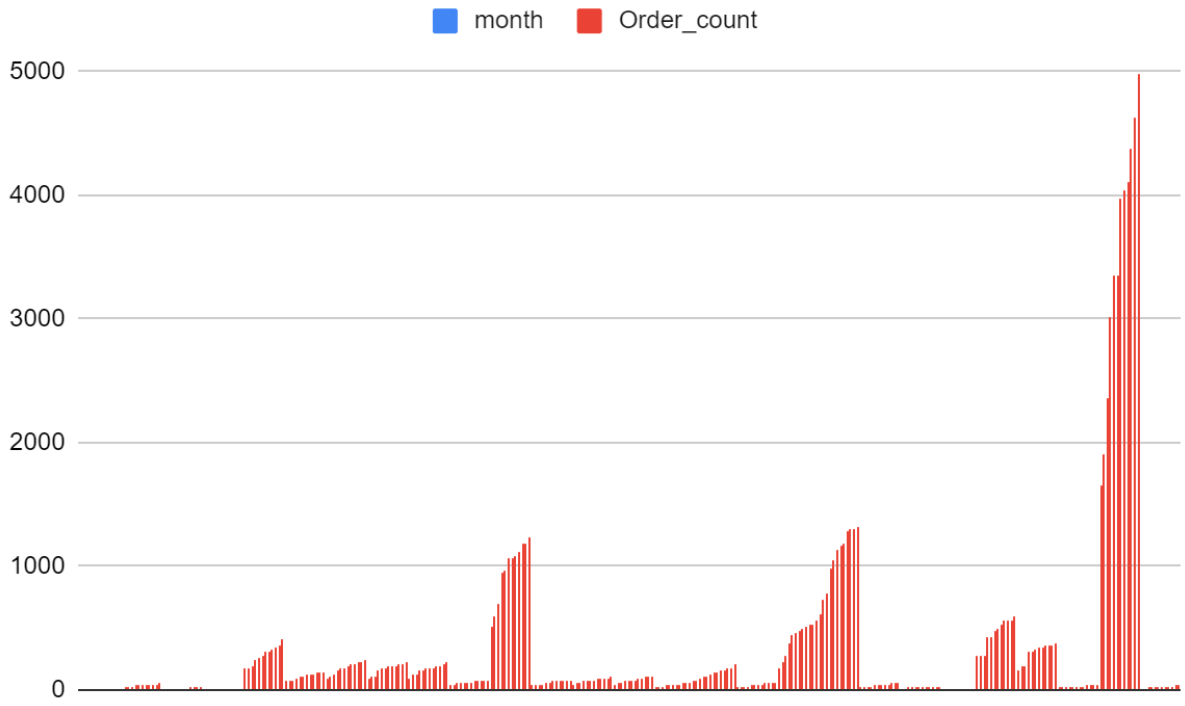
```
from `Retail.Orders` as o

JOIN `Retail.Customers` as c ON o.customer_id = c.customer_id

group by month , c.customer_state

order by c.customer_state , Order_count ;
```

Row	month	Order_count	customer_state
5	10	6	AC
6	2	6	AC
7	8	7	AC
8	6	7	AC
9	1	8	AC
10	4	9	AC
11	7	9	AC
12	5	10	AC
13	12	14	AL

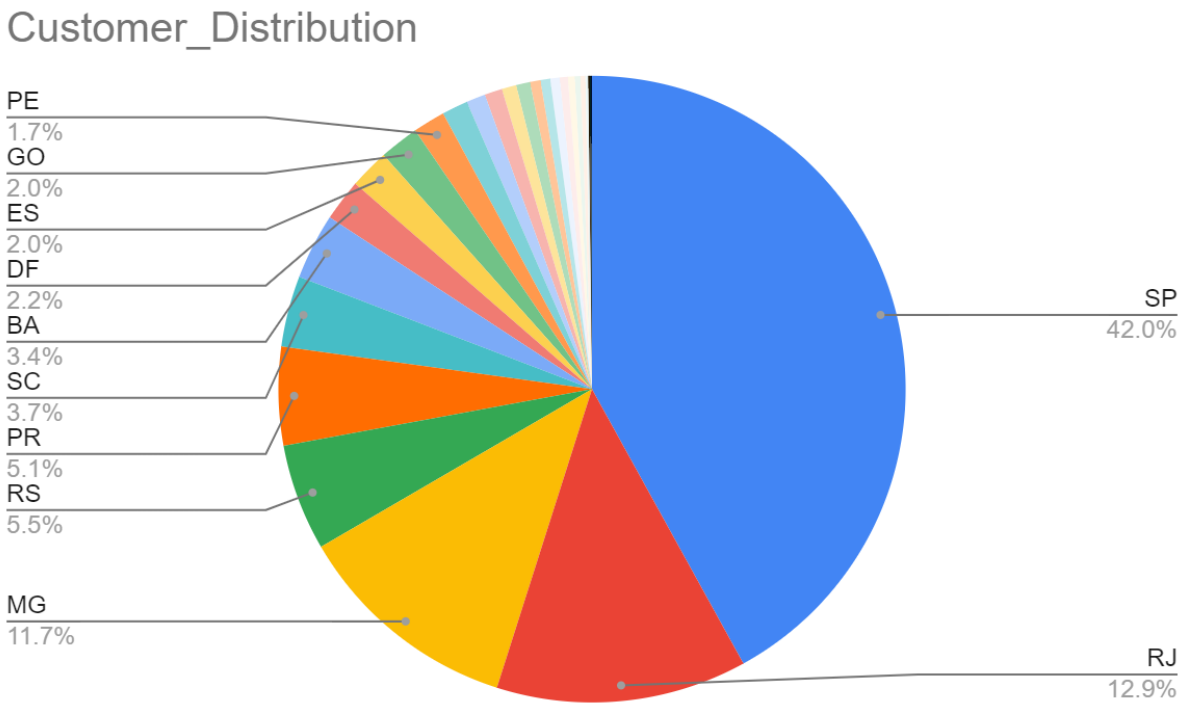


By analyzing the results, you can observe the order counts for each month and customer state, providing insights into the distribution of orders across different states and how it varies throughout the year. This information can help identify regional preferences, seasonal trends, and potential market opportunities within specific customer states.

```
#3.2 How are the customers distributed across all the states?
SELECT
    customer_state,

    SUM(COUNT(customer_id)) OVER(partition by customer_state) AS
Customer_Distribution
FROM
    `Retail.Customers`
GROUP BY
    customer_state
ORDER BY
    Customer_Distribution DESC;
```

Row	customer_state	Customer_Distributic
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
10	GO	2020



By analyzing the results, you can observe the customer distribution across different states. The states with a higher customer distribution indicate a larger customer base, while those with a lower distribution suggest a smaller

customer presence. This information can be useful for understanding the market reach and potential customer segments in different states, aiding in targeted marketing efforts and business expansion strategies.

```
#4.1 Get the % increase in the cost of orders from year 2017 to 2018 (include
months between Jan to Aug only).
/* VIEW is CREATED
create view Retail.targets as

select
o.order_id, o.customer_id, o.order_status, o.order_purchase_timestamp,
o.order_approved_at,
o.order_delivered_carrier_date,
o.order_delivered_customer_date,
o.order_estimated_delivery_date,
p.payment_sequential,
p.payment_type,
p.payment_installments,
p.payment_value,
r.review_id,
r.review_score,
r.review_comment_title,
r.review_creation_date,
r.review_answer_timestamp,
t.order_item_id,
t.product_id,
t.seller_id,
t.shipping_limit_date,
t.price,
t.freight_value
 from `Retail.Orders` o
left join `Retail.Payments` p
on p.order_id=o.order_id
left join `Retail.Order_Reviews` r
on o.order_id = r.order_id
```

```
left join `Retail.Order_Items` t
on o.order_id = t.order_id;

*/

with cte as (
select extract(month from order_purchase_timestamp) as order_month ,
extract(year from order_purchase_timestamp) as order_year,
payment_value
from `Retail.targets`
where extract(month from order_purchase_timestamp) between 1 and 8 and
extract(year from order_purchase_timestamp) in (2017 , 2018)
)

, cte1 as (
    select sum(payment_value) as total_cost, order_year
  from cte
  group by order_year
)

select ((a.total_cost - b.total_cost)/b.total_cost*100) as percentage_increase
, a.order_year, a.total_cost, b.total_Cost, b.order_year
from cte1 a
 join cte1 b
on a.order_year > b.order_year
```

Row	percentage_increase	order_year	total_cost	total_Cost_1	order_year_1
1	143.3474169495...	2018	11162746.75999...	4587164.679999...	2017

By analyzing the results, you can observe the percentage increase in payment value between consecutive years, providing insights into the growth or decline of total costs over time. This information can be useful for assessing business performance and identifying trends in expenditure for the analyzed period.

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

```sql
WITH
 cte as (
  select  round(sum(price),4) as Total_price , avg(price) as avg_price,
cs.customer_state
  from `Retail.targets` a
  join `Retail.Customers` cs
  on a.customer_id=cs.customer_id
  where a.price is not null
  group by customer_state

)

select  Total_price, avg_price, customer_state
from cte
order by customer_state;
```

Row	Total_price	avg_price	customer_state
1	17059.44	179.5730526315...	AC
2	83314.32	180.3340259740...	AL
3	22865.26	132.9375581395...	AM
4	13654.3	162.5511904761...	AP
5	543243.99	133.5408038348...	BA
6	240095.13	154.3027827763...	CE
7	315122.29	126.0489159999...	DF
8	284771.3	121.1277328796...	ES

By analyzing the results, you can observe the total price and average price for each customer state. This information can provide insights into the spending patterns and average transaction values across different states in the analyzed data set.

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

```sql
with cte as (
  select  round(sum(freight_value),4) as Total_frieght_value ,
avg(freight_value) as avg_freight_price, cs.customer_state
  from `Retail.targets` a
  join `Retail.Customers` cs
  on a.customer_id =cs.customer_id
  where a.freight_value is not null
  group by cs.customer_state

)

select  Total_frieght_value, avg_freight_price, customer_state
from cte
order by customer_state;
```

Row	Total_frieght_value	avg_freight_price	customer_state
1	3802.06	40.02168421052…	AC
2	16467.38	35.64367965367…	AL
3	5683.78	33.04523255813…	AM
4	2863.09	34.08440476190…	AP
5	106976.39	26.29704768928…	BA
6	50504.46	32.45787917737…	CE
7	52627.86	21.051144	DF
8	51667.24	21.97670778392…	ES
9	55799.34	22.74738687321…	GO

The CTE named "cte" joins the "Retail.targets" table with the "Retail.Customers" table on the customer ID. It calculates the rounded sum of freight values as the total freight value and the average freight price per state. Null freight values are excluded from the calculation.

The main query selects the total freight value, average freight price, and customer state from the CTE and orders the results by customer state in ascending order.

By analyzing the results, you can observe the total freight value and average freight price for each customer state. This information can provide insights into the freight cost patterns and average shipping expenses across different states in the analyzed dataset, aiding in logistics planning, cost analysis, and identifying potential regional variations in shipping costs.

```
 /*# 5.1 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,  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 `Retail.targets`
  where order_delivered_customer_date is not null
  order by order_id;
```

Row	order_id	time_to_deliver	diff_estimated_delive
1	00010242fe8c5a6d1ba2dd792...	7	8
2	00018f77f2f0320c557190d7a1...	16	2
3	000229ec398224ef6ca0657da...	7	13
4	00024acbcdf0a6daa1e931b03...	6	5
5	00042b26cf59d7ce69dfabb4e...	25	15
6	00048cc3ae777c65dbb7d2a06...	6	14
7	00054e8431b9d7675808bcb8...	8	16
8	000576fe39319847cbb9d288c...	5	15
9	0005a1a1728c9d785b8e2b08...	9	0

By analyzing the results, you can observe the delivery time for each order and the variance between the estimated and actual delivery dates. This information can provide insights into the efficiency of the delivery process, adherence to estimated delivery timelines, and potential areas for improvement in managing delivery schedules and customer expectations.

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

```
 with cte as(
    select avg(freight_value) as average_freight_value, c.customer_state as
state
    from `Retail.targets` a
    join `Retail.Customers` c
    on a.customer_id = c.customer_id
    group by c.customer_state
  )

  , cte1 as (select average_freight_value, state,
   row_number() over(order by average_freight_value asc) as top,
   row_number() over(order by average_freight_value desc) as lowest
  from cte
  )
```

```sql
select average_freight_value, state
from cte1
where top<=5 or lowest <=5
order by average_freight_value
```

Row	average_freight_valu	state
1	15.19151870049...	SP
2	20.58536740146...	PR
3	20.60969674879...	MG
4	21.051144	DF
5	21.09405445705...	RJ
6	39.67728613569...	TO
7	40.02168421052...	AC
8	40.97017482517...	RO

The CTE named "cte" joins the "Retail.targets" table with the "Retail.Customers" table on the customer ID. It calculates the average freight value per customer state by grouping the results by customer state.
The CTE named "cte1" uses the results from "cte" and assigns row numbers based on the ascending and descending order of the average freight value.
The main query selects the average freight value and state from "cte1" where the row number is within the top 5 (lowest average freight value) or lowest 5 (highest average freight value).

By analyzing the results, you can identify the top 5 states with the lowest and highest average freight values. This information can provide insights into regional variations in shipping costs and help identify states that may require specific attention or optimization in terms of freight management and cost control.

```
#5.3 Find out the top 5 states with the highest & lowest average delivery
time.

with cte as (
  select c.customer_state as state,
   avg( date_diff(a.order_delivered_customer_date, a.order_purchase_timestamp,
day)) as time_to_deliver
  from `Retail.targets` a
  join `Retail.Customers` c
  on a.customer_id = c.customer_id
  where order_delivered_customer_date is not null
  group by c.customer_state

)

, cte1 as(
  select state, time_to_deliver,
  row_number() over(order by time_to_deliver asc) as top,
  row_number() over(order by time_to_deliver desc) as lowest
  from cte
)
 select state, time_to_deliver
 from cte1
  where top <=5 or lowest <=5
  order by time_to_deliver
```

Row	state	time_to_deliver
1	SP	8.274159513224...
2	MG	11.49996295473...
3	PR	11.52307431286...
4	DF	12.50143032284...
5	SC	14.51344339622...
6	PA	23.26800364630...
7	AL	24.12808988764...
8	AM	26.06470588235...
9	AP	27.66265060240...
10	RR	27.82608695652...

By analyzing the results, you can identify the top 5 states with the fastest and slowest average delivery times. This information can provide insights into the efficiency of the delivery process across different states, highlight areas for improvement in logistics and fulfillment operations, and aid in identifying potential customer satisfaction issues related to delivery times.

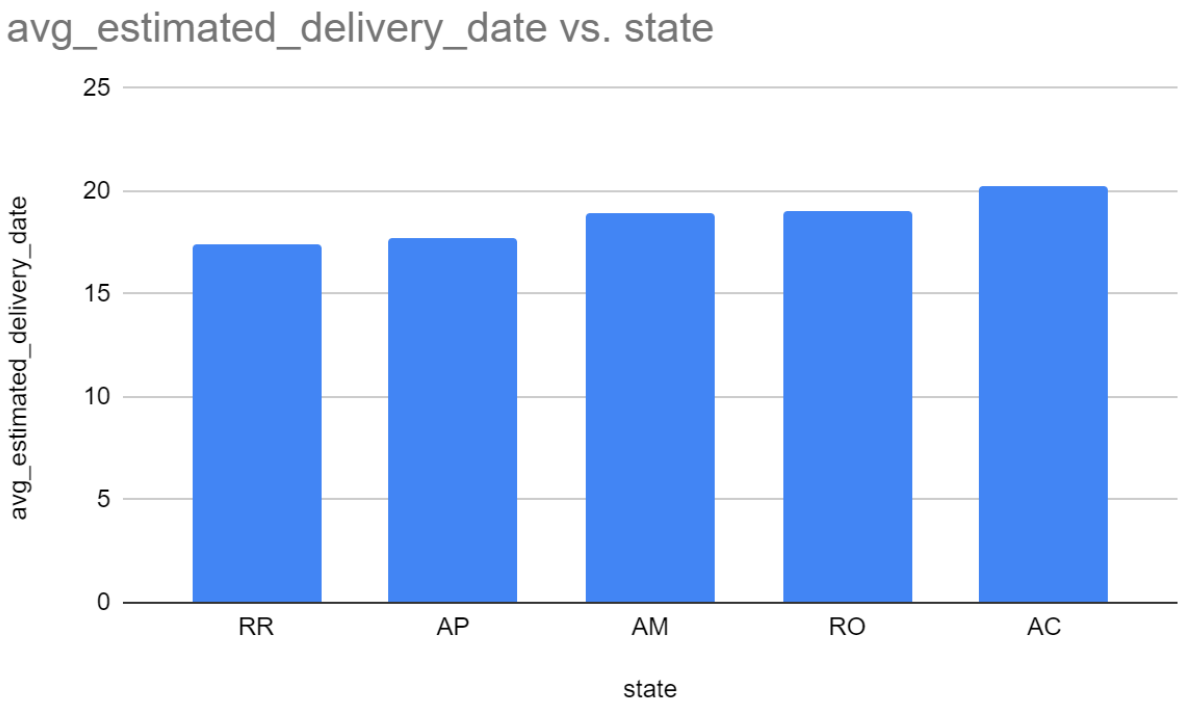
```
#5.4 Find out the top 5 states where the order delivery is really fast as
compared to the estimated date of delivery.
/* You can use the difference between the averages of actual & estimated
delivery date to figure out how fast the delivery was for each state.*/

with cte as (
  select c.customer_state as state,
   avg( date_diff(a.order_estimated_delivery_date,
a.order_delivered_customer_date, day)) as avg_estimated_delivery_date
  from  `Retail.targets` a
  join  `Retail.Customers` c
```

```
  on a.customer_id = c.customer_id
  where order_delivered_customer_date is not null
  group by c.customer_state

)
, cte1 as(
  select state, avg_estimated_delivery_date,
  row_number() over(order by avg_estimated_delivery_date desc) as top_deliver
  from cte
)
 select state, avg_estimated_delivery_date
 from cte1
  where  top_deliver <=5
  order by avg_estimated_delivery_date
```

Row	state	avg_estimated_deliv
1	RR	17.43478260869…
2	AP	17.69879518072…
3	AM	18.88823529411…
4	RO	19.02135231316…
5	AC	20.21276595744…

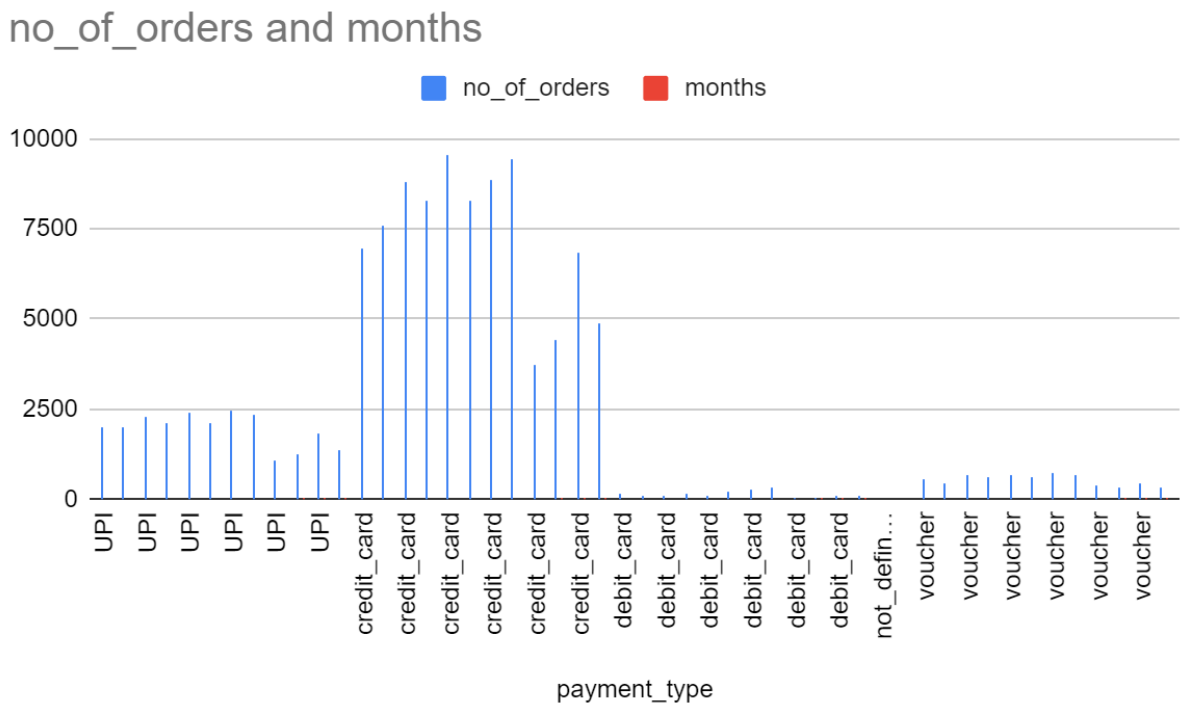


By analyzing the results, you can identify the top 5 states with the highest average difference between the estimated and actual delivery dates. This information can provide insights into potential delays or issues in meeting estimated delivery timelines for different states, enabling businesses to focus on improving their delivery performance and customer satisfaction in those areas.

```
#6.1 Find the month on month no. of orders placed using different payment types.

with cte as(
select extract(month from order_purchase_timestamp) as months,
order_id, payment_type
from `Retail.targets`
)
  select count(order_id) as no_of_orders, payment_type, months
  from cte
  where payment_type is not null
  group by payment_type, months
  order by payment_type, months
```

Row	no_of_orders	payment_type	months
1	2017	UPI	1
2	2027	UPI	2
3	2279	UPI	3
4	2102	UPI	4
5	2388	UPI	5
6	2090	UPI	6
7	2442	UPI	7
8	2363	UPI	8

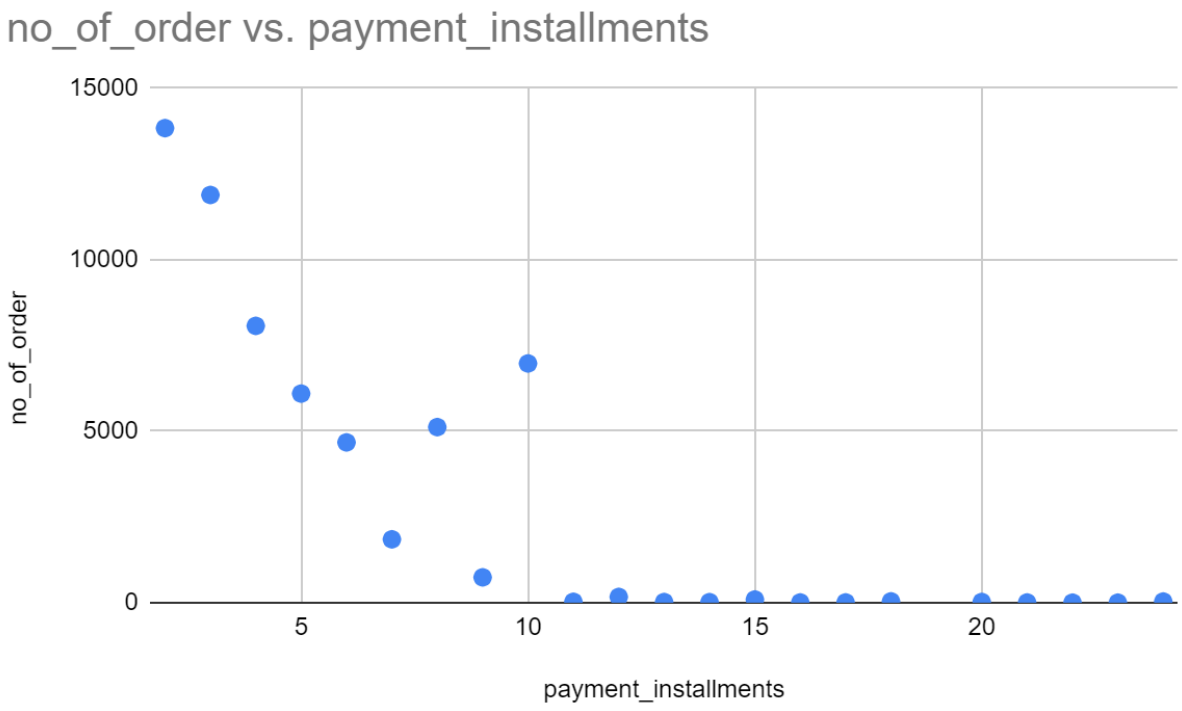


The results provide insights into the number of orders for each payment type and each month. By analyzing this data, you can identify patterns, trends, or seasonal variations in the number of orders based on payment types. This information can be valuable for understanding customer preferences, optimizing payment methods, and planning marketing or sales strategies accordingly.

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

```
select payment_installments, count(order_id) as no_of_order
from `Retail.targets`
where 1< payment_installments and payment_installments is not null
group by payment_installments
```

Row	payment_installment	no_of_order
1	7	1848
2	10	6976
3	6	4674
4	2	13838
5	4	8072
6	3	11889
7	8	5120
8	9	739
9	5	6097



This information can provide insights into customer preferences for payment installment options and help businesses understand the usage patterns of different payment terms.
Analyzing the results can assist in optimizing payment options, designing targeted marketing campaigns, and improving customer satisfaction by aligning payment installment offerings with customer preferences.

Overall Recommendations :

Growing trend in the number of orders: If there is a growing trend in the number of orders placed over the past years, it indicates a positive business outlook. To capitalize on this trend, the business can focus on improving its marketing and customer acquisition strategies to attract more customers and increase order volume.

Monthly seasonality: Analyzing the monthly seasonality in terms of the number of orders being placed can provide insights into customer behavior and

preferences. The business can leverage this information to optimize inventory management, marketing campaigns, and promotional activities based on seasonal trends.

Time of day for order placement: Understanding the preferred time of day for order placement by Brazilian customers can help optimize operations and customer service. By aligning staffing and logistics resources accordingly, the business can ensure efficient order processing and timely delivery, leading to improved customer satisfaction.

Customer distribution across states: Analyzing the distribution of customers across all states can provide valuable insights into geographic markets. The business can use this information to identify regions with a high concentration of customers and tailor marketing strategies and promotions to target those specific areas, thereby maximizing sales potential.