Business Case Study

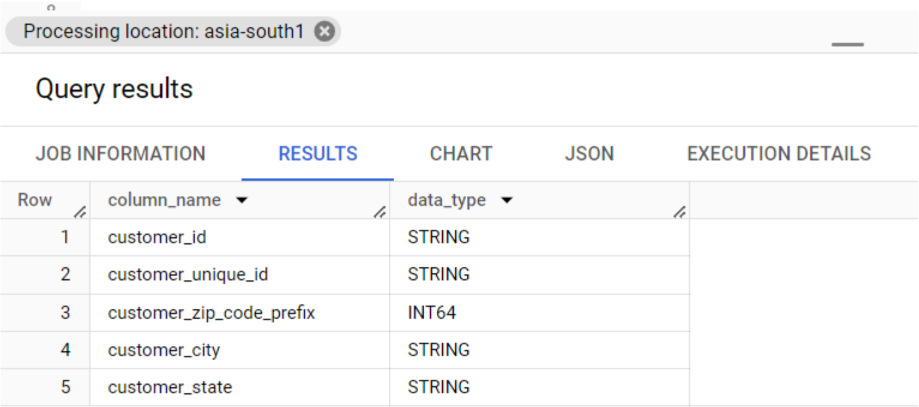
1. Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset:

Q1 . Data type of all columns in the "customers" table.

Code :

```
select column_name, data_type
from target_BCS_ecom.INFORMATION_SCHEMA.COLUMNS
WHERE table_name = 'customers'
```

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

INSIGHTS :

When we fetch the data-types of each column in the table it helps us for ensuring the understanding of data and table structure. Here customer_id and customer_unique_id are typically a unique identifier and either can be consider for primary key.

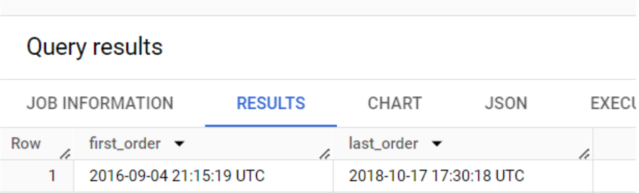
This unique identifier distinguishes one customer from another in a more granular way. A customer's unique ID is a more specific, individualized identifier assigned to them. Target can use this ID internally for data analysis, operational purposes, or as a reference in their databases.

Q2. Get the time range between which the orders were placed.

Query :

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

Output :



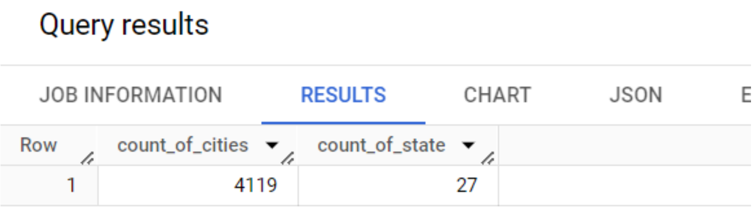
Row	first_order	last_order
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC

Q3 Count the Cities & States of customers who ordered during the given period.

Query :

```
select count(distinct customer_city) as count_of_cities ,
 count(distinct customer_state) as count_of_state
from `target_BCS_ecom.customers`
inner join `target_BCS_ecom.orders` using(customer_id)
```

Output:



Row	count_of_cities	count_of_state
1	4119	27

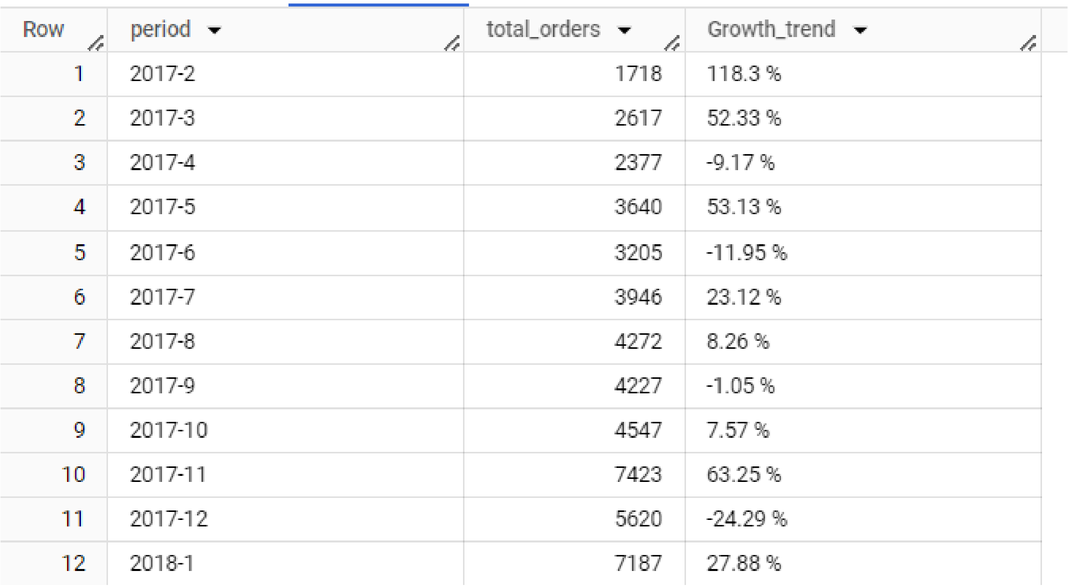
2. In-depth Exploration:

Q1 Is there a growing trend in the no. of orders placed over the past years?

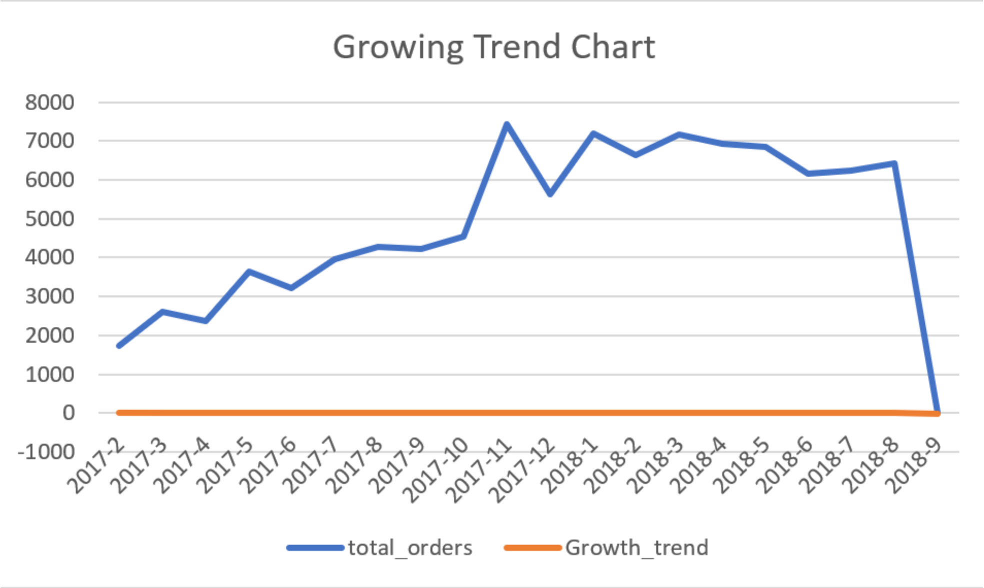
Code :

```
with cte as (
 select concat(year,'-',mnth) as period, sum(no_of_orders) as total_orders,
 lag(sum(no_of_orders)) over(order by year,mnth) as lagg
 from (
  select extract(year from order_purchase_timestamp) as year,
        extract(month from order_purchase_timestamp) as mnth, count(*) as no_of_orders
from `target_BCS_ecom.orders`
        where order_status not in ('canceled','unavailable') and
        extract(year from order_purchase_timestamp) >= 2017
  group by order_purchase_timestamp,year,mnth
  order by year,mnth
 )
 group by year, mnth
 order by year, mnth
)
select period, total_orders, Growth_trend
 from ( select * , concat(round(((total_orders-lagg) / lagg) * 100,2),' %') as Growth_trend
 from cte )
 where Growth_trend is not null;
```

Output :



Row	period	total_orders	Growth_trend
1	2017-2	1718	118.3 %
2	2017-3	2617	52.33 %
3	2017-4	2377	-9.17 %
4	2017-5	3640	53.13 %
5	2017-6	3205	-11.95 %
6	2017-7	3946	23.12 %
7	2017-8	4272	8.26 %
8	2017-9	4227	-1.05 %
9	2017-10	4547	7.57 %
10	2017-11	7423	63.25 %
11	2017-12	5620	-24.29 %
12	2018-1	7187	27.88 %



Insights :

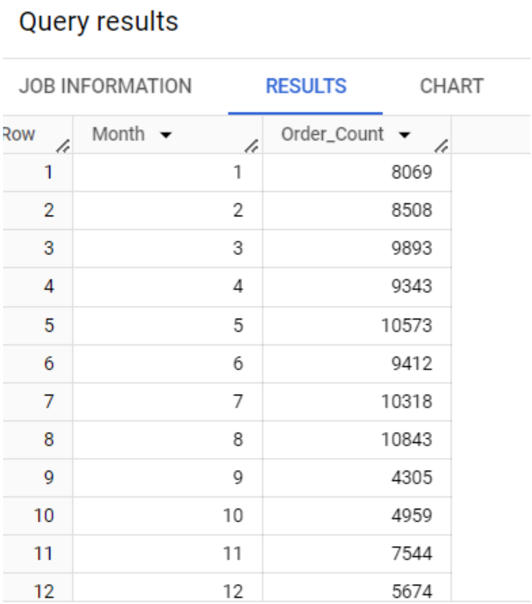
As a result, we can see a consistent and significant upward trend in the number of orders placed, while in terms of revenue, we see an upward trend over the Brazil region. From the chart analysis, we can see that Target's business in Brazil is performing well and customers are increasingly interested in its products, but we must work on revenue growth to gain an uptrend.

Q2 Can we see some kind of monthly seasonality in terms of the no. of orders being placed?

Code :

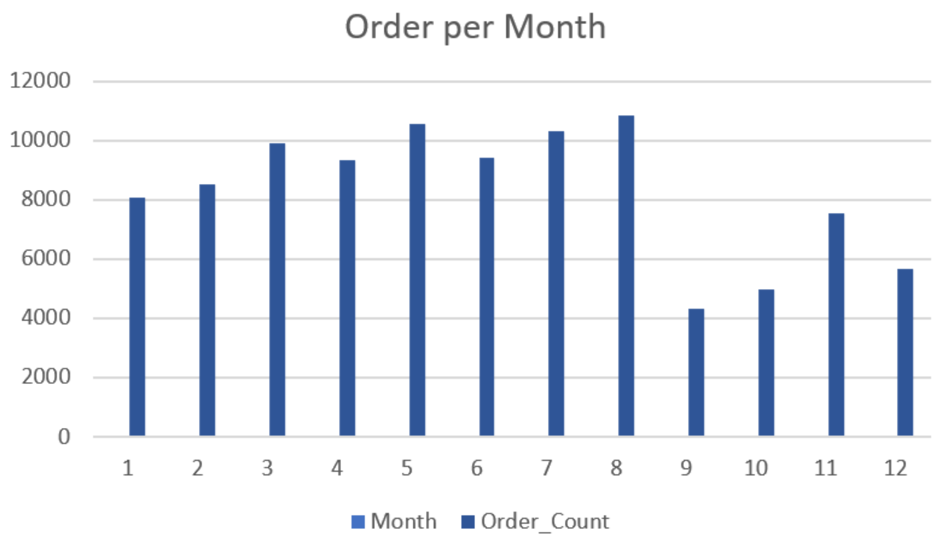
```
SELECT EXTRACT(MONTH FROM order_purchase_timestamp) AS `Month`,
    COUNT(order_id) AS Order_Count
FROM `target_BCS_ecom.orders`
GROUP BY EXTRACT(MONTH FROM order_purchase_timestamp)
ORDER BY Month ;
```

Output :



Query results

Row	Month	Order_Count
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
11	11	7544
12	12	5674



INSIGHTS :

The objective of the query was to calculate the total count of orders for each month over the years to get an understanding and manage the business order patterns.

From this we can analysis the order patterns for each month over the past years. We can infer that the peak is at 7th month i.e August shows the highest number of orders and September month shows the lowest number of orders. By using line graph, we can understand it easily.

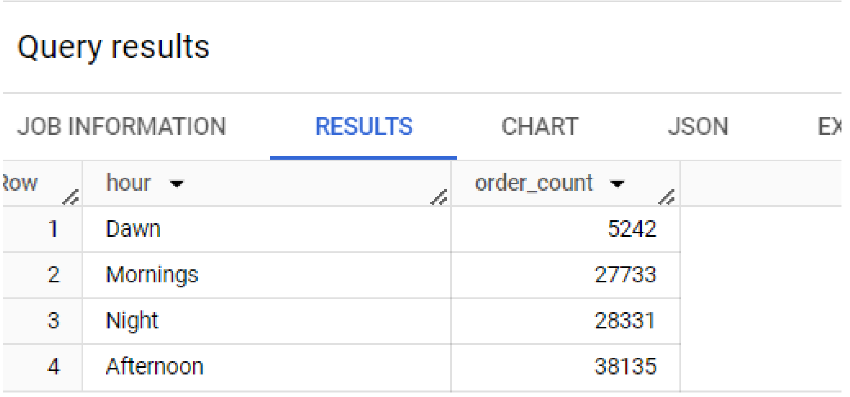
Q3 During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)

- 0-6 hrs : Dawn
- 7-12 hrs : Mornings
- 13-18 hrs : Afternoon
- 19-23 hrs : Night

Code :

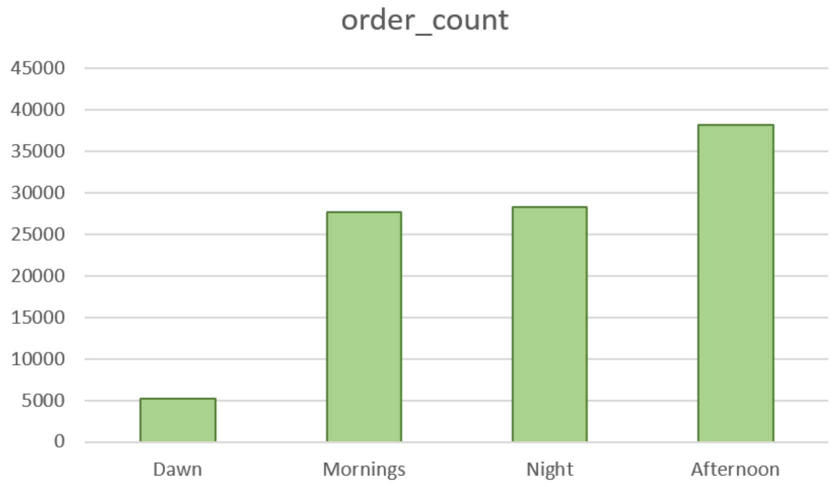
```
SELECT
  CASE
    WHEN EXTRACT(HOUR FROM ord.order_purchase_timestamp) <= 6 THEN 'Dawn'
    WHEN EXTRACT(HOUR FROM ord.order_purchase_timestamp) BETWEEN 7 AND 12 THEN 'Mornings'
    WHEN EXTRACT(HOUR FROM ord.order_purchase_timestamp) BETWEEN 13 AND 18 THEN 'Afternoon'
    WHEN EXTRACT(HOUR FROM ord.order_purchase_timestamp) BETWEEN 19 AND 23 THEN 'Night'
  END AS hour,
 COUNT(ord.order_id) AS order_count
 FROM `target_BCS_ecom.orders` ord
 JOIN `target_BCS_ecom.customers` cust
 ON ord.customer_id = cust.customer_id
 GROUP BY hour
 ORDER BY order_count
```

Output :



Query results

Row	hour	order_count
1	Dawn	5242
2	Mornings	27733
3	Night	28331
4	Afternoon	38135



INSIGHTS :

From the result we can see that the highest number of orders are at afternoon and the least are at dawn. This suggests that customers in Brazil are actively engaged in online shopping during the afternoon and night, possibly during leisure time, with a significant number of orders placed in the morning as well.

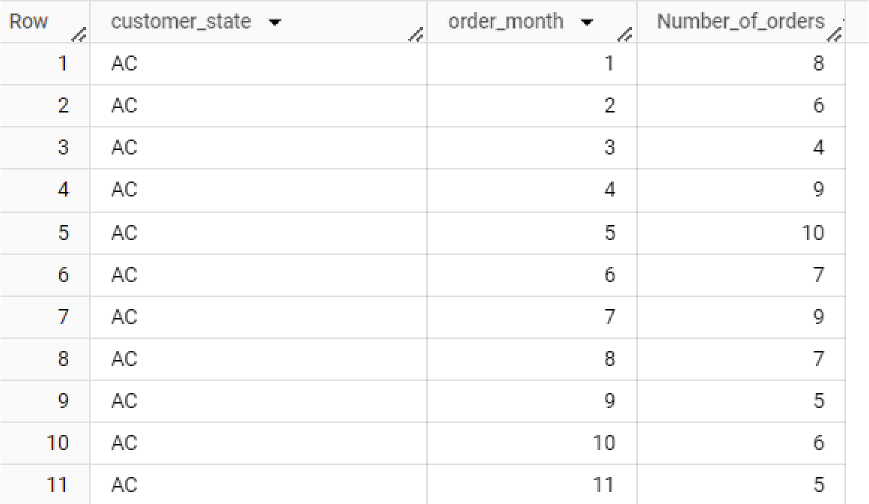
3 Evolution of E-commerce orders in the Brazil region

Q1 Get the month on month no. of orders placed in each state.

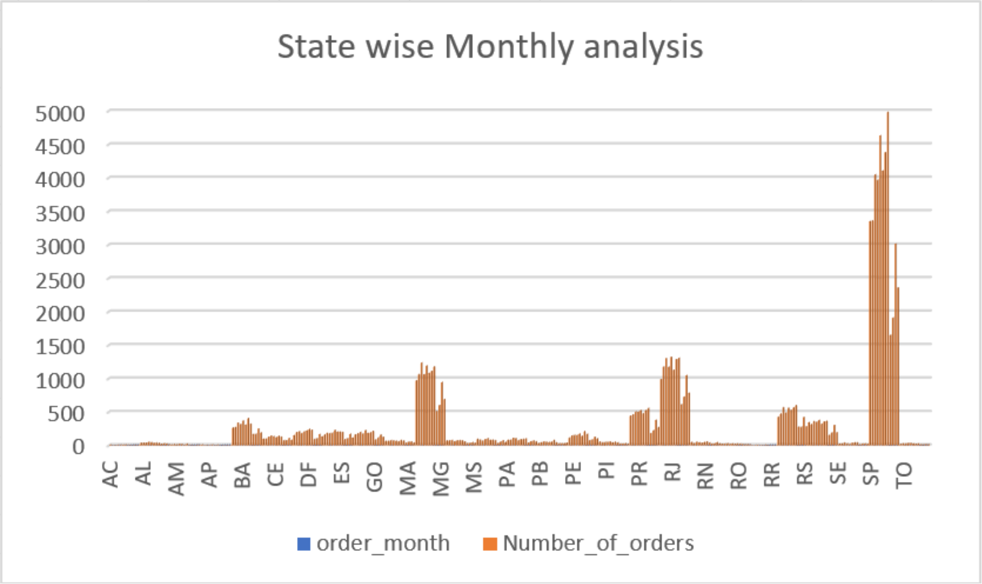
Code :

```
select c.customer_state , extract (Month from o.order_purchase_timestamp) as order_month ,
count(o.order_id) as Number_of_orders
from `target_BCS_ecom.orders` o join `target_BCS_ecom.customers` c on
o.customer_id=c.customer_id
  group by  c.customer_state,order_month
  order by c.customer_state,order_month
```

Output :



Row	customer_state	order_month	Number_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
8	AC	8	7
9	AC	9	5
10	AC	10	6
11	AC	11	5



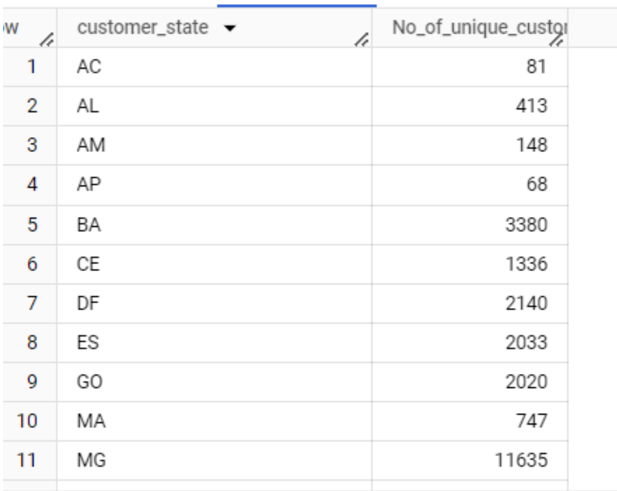
Insights :

By analysing this data, this enables data-driven decision-making, aiding in production planning, considering seasonality, state-wise performance, and growth trends, to optimize marketing, operations, and resource allocation strategies across different states throughout the Brazil country.

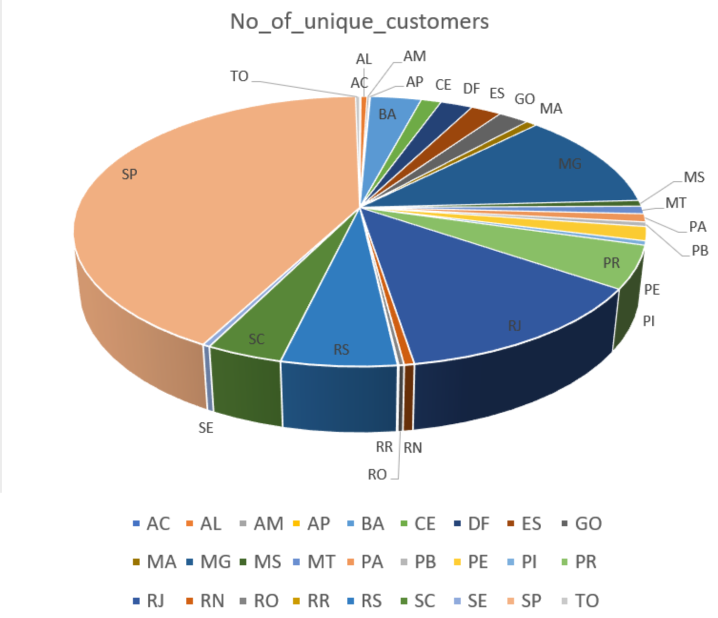
Q2 How are the customers distributed across all the states?
Code :

```
Select customer_state, count(customer_unique_id) AS No_of_unique_customers
from `target_BCS_ecom.customers`
group by customer_state
order by customer_state
```

Output :



w	customer_state	No_of_unique_custo
1	AC	81
2	AL	413
3	AM	148
4	AP	68
5	BA	3380
6	CE	1336
7	DF	2140
8	ES	2033
9	GO	2020
10	MA	747
11	MG	11635



Insight :

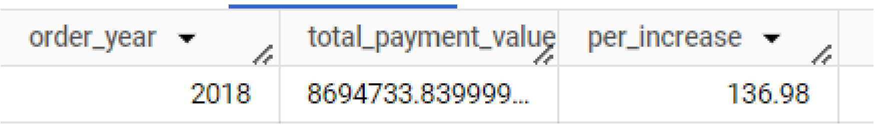
By examining customer distribution across states, regional trends and patterns can be identified. States like Sao Paulo (SP) and Rio de Janeiro (RJ) with a high count of customers and states with lower customer counts are Roraima (RR) and Tocantins (TO) with only 45 customers.
By analyzing this data, marketing campaigns can be targeted for specific regions, new markets can be expanded into, customer retention strategies can be improved, and investment opportunities can be identified.

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

Q1 Get the % increase in the cost of orders from year 2017 to 2018 (include months between Jan to Aug only).
Code :

```
WITH Total_Payments_by_year AS (
  SELECT  EXTRACT(YEAR FROM o.order_purchase_timestamp)  AS order_year,
  SUM(p.payment_value) AS total_payment_value
  FROM `target_BCS_ecom.orders` o JOIN `target_BCS_ecom.payments` p ON o.order_id = p.order_id
  WHERE EXTRACT(YEAR FROM o.order_purchase_timestamp)IN (2017, 2018) AND
  EXTRACT(MONTH FROM o.order_purchase_timestamp) BETWEEN  1 AND 8
  GROUP BY  EXTRACT(YEAR FROM o.order_purchase_timestamp)
),
Payment_Increase AS (
  SELECT  order_year,
  total_payment_value,
 LAG(total_payment_value) OVER (ORDER BY order_year) AS prev_year_payment_value
  FROM Total_Payments_by_year
)
SELECT order_year,total_payment_value,
 round(((total_payment_value - prev_year_payment_value) / prev_year_payment_value) * 100,2) AS
per_increase
FROM Payment_Increase
WHERE  prev_year_payment_value IS NOT NULL
ORDER BY order_year;
```

Output :



order_year	total_payment_value	per_increase
2018	8694733.839999...	136.98

Insight :

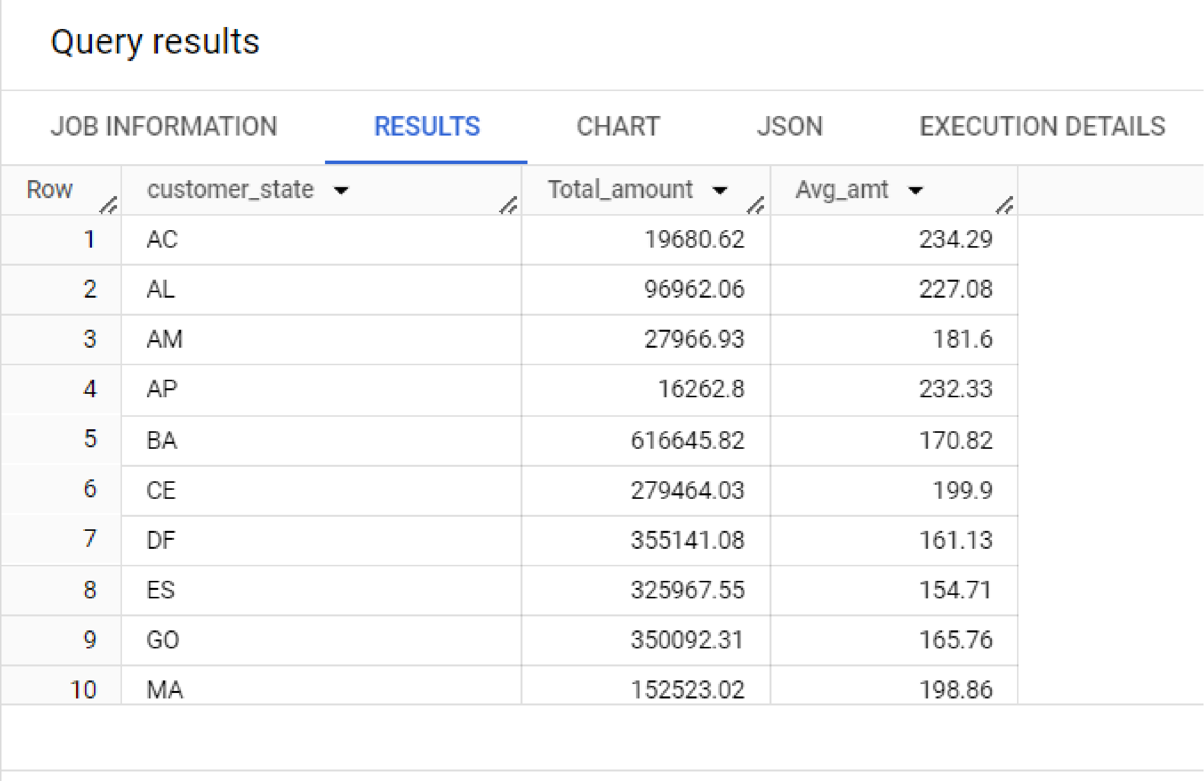
The value of orders has increased significantly from 2017 to 2018 by 136.98 %
This data shows a positive trend of increasing order amounts from 2017 to 2018, a substantial decrease in cost of the products makes us understand the economic growth of the country indicating potential growth and success in the business across Brazilian states.

Q2 Calculate the Total & Average value of order price for each state.

Code :

```
select customer_state ,
round(sum(payment_value),2) as Total_amount,
round(avg(payment_value),2) as Avg_amt
from `target_BCS_ecom.customers` cust
join `target_BCS_ecom.orders` ord
on cust.customer_id = ord.customer_id
join`target_BCS_ecom.payments` pay
on pay.order_id = ord.order_id
group by customer_state
order by customer_state;
```

Output :



Row	customer_state	Total_amount	Avg_amt
1	AC	19680.62	234.29
2	AL	96962.06	227.08
3	AM	27966.93	181.6
4	AP	16262.8	232.33
5	BA	616645.82	170.82
6	CE	279464.03	199.9
7	DF	355141.08	161.13
8	ES	325967.55	154.71
9	GO	350092.31	165.76
10	MA	152523.02	198.86

Insights :

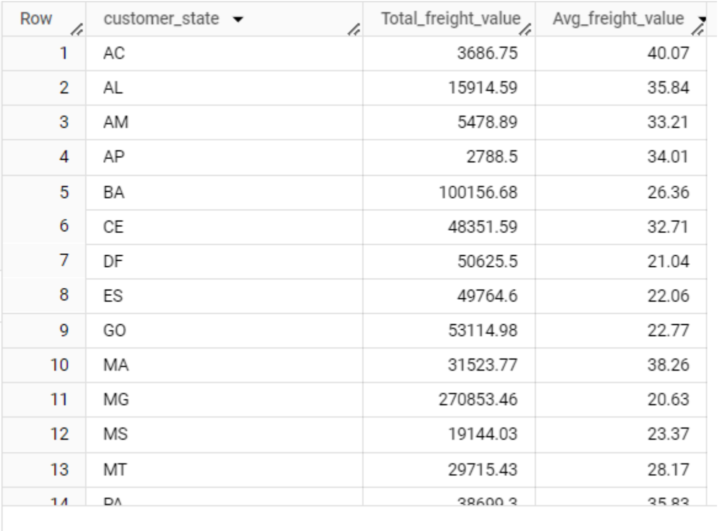
From this result we can analyse the marketing campaigns and provides insights into regional customer behaviour and preferences.

Q3 Calculate the Total & Average value of order freight for each state.

Code :

```
select customer_state ,
round(sum(freight_value),2) as Total_freight_value,
round(avg(freight_value),2) as Avg_freight_value
from `target_BCS_ecom.customers` cust
join `target_BCS_ecom.orders` ord on cust.customer_id=ord.customer_id
join`target_BCS_ecom.order_items` ord_item on ord_item.order_id = ord.order_id
group by customer_state
order by customer_state;
```

Output :



Row	customer_state	Total_freight_value	Avg_freight_value
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
10	MA	31523.77	38.26
11	MG	270853.46	20.63
12	MS	19144.03	23.37
13	MT	29715.43	28.17
14	PA	38699.3	35.83

Insight :

The data shows the total and average freight value for each state.
Which furthers down to understand how efficient is one state to other w.r.t cost involved in delivering the goods to customers. Required actions to be taken to improve freight cost of the states wherever its higher than expected.

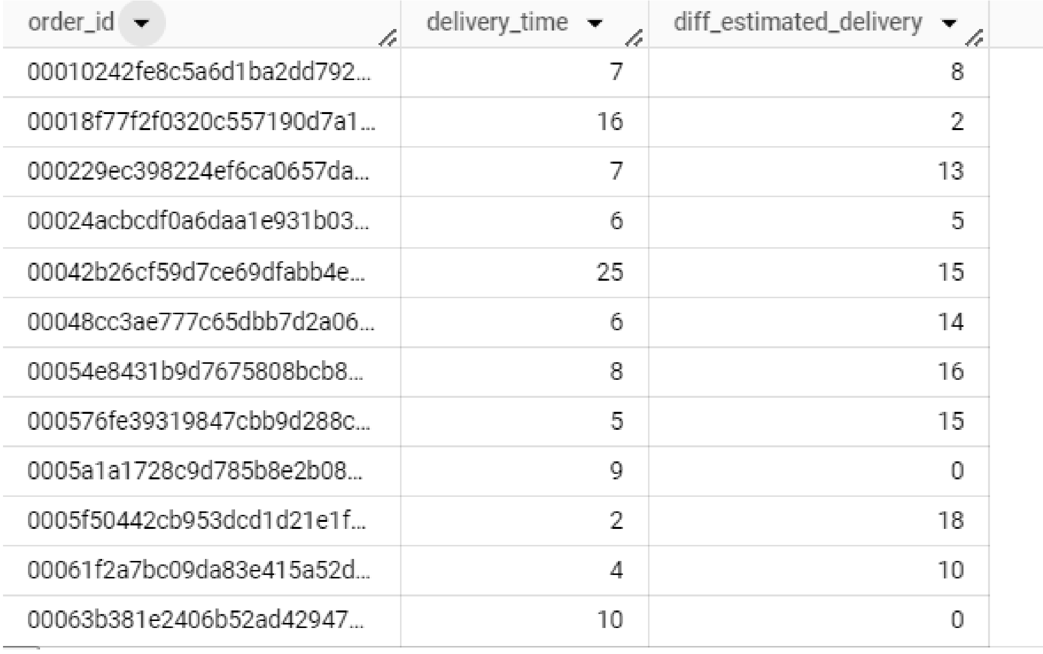
5 Analysis based on sales, freight and delivery time.

Q1 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.

Code :

```
Select order_id,
datetime_diff(order_delivered_customer_date,order_purchase_timestamp,day) as delivery_time,
datetime_diff(order_estimated_delivery_date,order_delivered_customer_date,day) as
diff_estimated_delivery
from
`target_BCS_ecom.orders`
order by order_id
```

Output :



order_id	delivery_time	diff_estimated_delivery
00010242fe8c5a6d1ba2dd792...	7	8
00018f77f2f0320c557190d7a1...	16	2
000229ec398224ef6ca0657da...	7	13
00024acbcdf0a6daa1e931b03...	6	5
00042b26cf59d7ce69dfabb4e...	25	15
00048cc3ae777c65dbb7d2a06...	6	14
00054e8431b9d7675808bcb8...	8	16
000576fe39319847cbb9d288c...	5	15
0005a1a1728c9d785b8e2b08...	9	0
0005f50442cb953dcd1d21e1f...	2	18
00061f2a7bc09da83e415a52d...	4	10
00063b381e2406b52ad42947...	10	0

Insight :

By analyzing this data, we can identify orders that took longer to deliver and compare each delivery time with the average delivery timeline to assess delivery efficiency. Negative values in the "diff_estimated_delivery" column indicate delayed deliveries, while positive values indicate early deliveries.

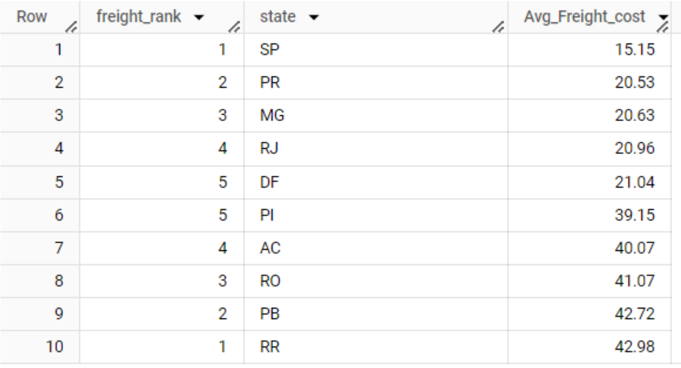
Digging deeper, the reasons for these variances can help improve delivery timelines and reduce the difference between estimated and actual delivery dates, leading to enhanced logistics and delivery processes.

Q2 Find out the top 5 states with the highest & lowest average freight value.

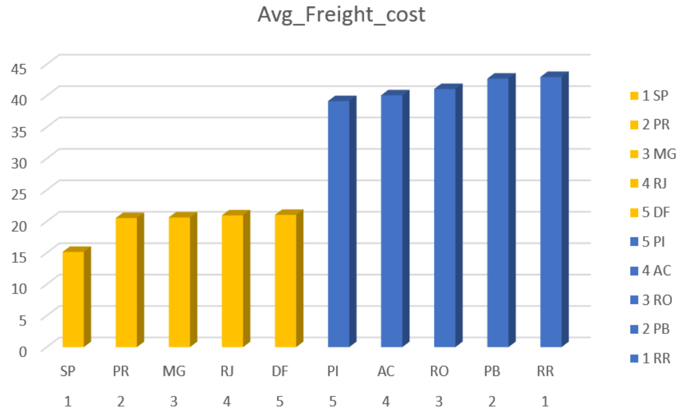
Code :

```
with cte as (
select State,avg_freight_value, dense_rank() over (order by avg_freight_value desc) as avg_rank
from (
select customer_state as state,round(avg(freight_value),2) as avg_freight_value
from `target_BCS_ecom.customers` as c
join `target_BCS_ecom.orders` as o on c.customer_id = o.customer_id
join `target_BCS_ecom.order_items` as p on o.order_id = p.order_id
group by customer_state
) nt1
union all
select state,avg_freight_value, dense_rank() over (order by avg_freight_value asc) as avg_rank
from (
select customer_state as state,round(avg(freight_value),2) as avg_freight_value
from `target_BCS_ecom.customers` as c
join `target_BCS_ecom.orders` as o on c.customer_id = o.customer_id
join `target_BCS_ecom.order_items` as p on o.order_id = p.order_id
group by customer_state
) nt2 )
select avg_rank as freight_rank,state,cte.avg_freight_value as Avg_Freight_cost from cte
where avg_rank<=5
order by Avg_Freight_cost ;
```

output :



Row	freight_rank	state	Avg_Freight_cost
1	1	SP	15.15
2	2	PR	20.53
3	3	MG	20.63
4	4	RJ	20.96
5	5	DF	21.04
6	5	PI	39.15
7	4	AC	40.07
8	3	RO	41.07
9	2	PB	42.72
10	1	RR	42.98



Insight :

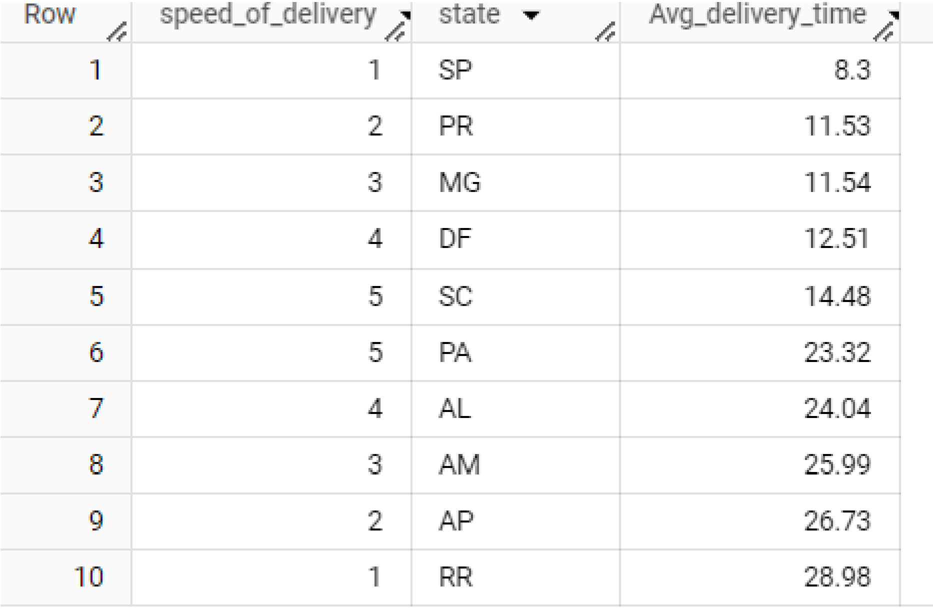
Certain states exhibit higher average freight values, indicating specific characteristic or logistical challenges that result in increased freight costs. Some states demonstrate lower average freight values, suggesting more favourable logistics infrastructure or other factors contributing to reduced freight costs. With average freight costs , we can't hat there is a notable difference (Significant Variation) in the average freight value among different states.

Q3 Find out the top 5 states with the highest & lowest average delivery time.

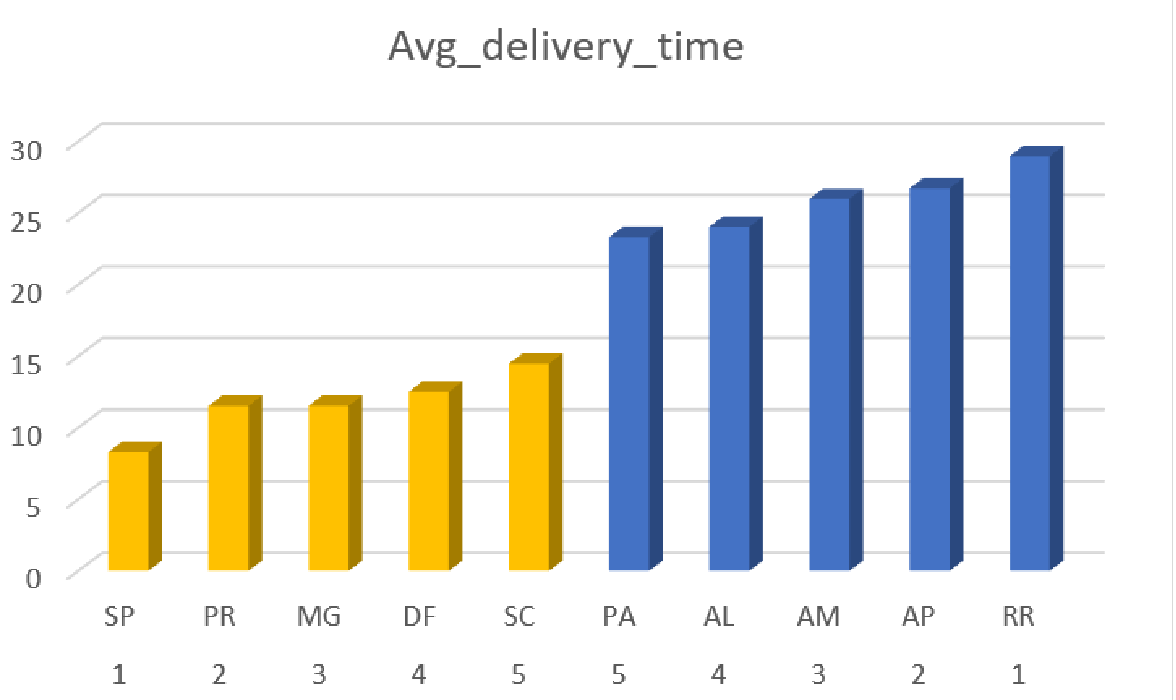
Code :

```
with cte as
(
select state,avg(delivery_time) as avg_delivery_time,
dense_rank() over (order by avg(delivery_time) desc) as rnk
from
(
select customer_state as state,
datetime_diff(order_delivered_customer_date,order_purchase_timestamp,day) as delivery_time,
from `target_BCS_ecom.customers` as c
join `target_BCS_ecom.orders` as o on c.customer_id = o.customer_id
group by state,order_delivered_customer_date,order_purchase_timestamp,delivery_time
) nt1
group by state
union all
select state,avg(delivery_time) as avg_delivery_time,
dense_rank() over (order by avg(delivery_time) asc) as rnk
from
(
select customer_state as state,
datetime_diff(order_delivered_customer_date,order_purchase_timestamp,day) as delivery_time,
from `target_BCS_ecom.customers` as c
join `target_BCS_ecom.orders` as o on c.customer_id = o.customer_id
group by state,order_delivered_customer_date,order_purchase_timestamp,delivery_time
) nt2
group by state
)
select rnk as speed_of_delivery,state,
round(avg_delivery_time,2)as Avg_delivery_time
from cte
where rnk<=5
order by Avg_delivery_time;
```

Output :



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



Insight :

The data shows the difference between actual delivery time versus estimated delivery time of an order. The states with high average delivery time are the ones which need attention w.r.t resources and planning in those states. If not improved can negatively impact customer experience.

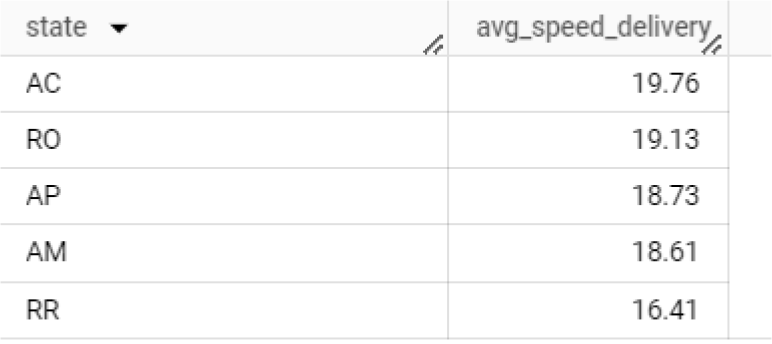
The states with low average delivery time can be the model to those with high delivery time. Strategy can be devised to further bring it down w.r.t best in the industry.

Q4 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.

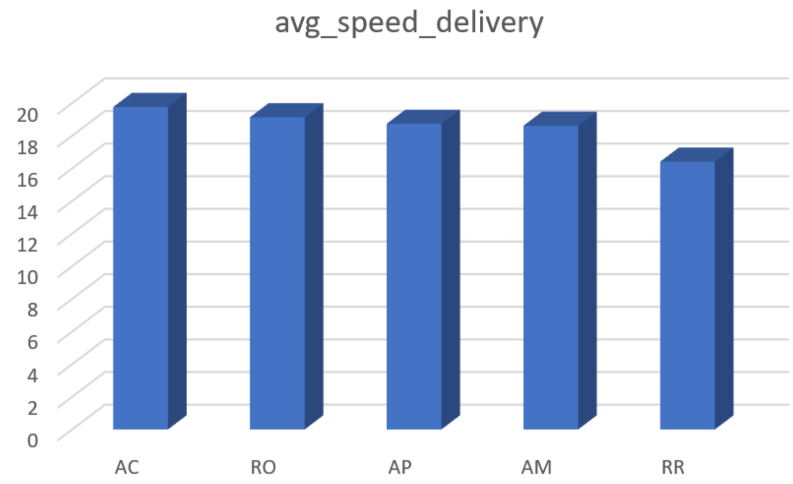
Code :

```
select customer_state as state,
round(avg(date_diff(o.order_estimated_delivery_date,o.order_delivered_customer_date,day)),2)as
avg_speed_delivery
from `target_BCS_ecom.customers` as c
join `target_BCS_ecom.orders` as o on c.customer_id = o.customer_id
where o.order_status="delivered"and o.order_delivered_customer_date is not null
group by state
order by avg_speed_delivery desc
limit 5
```

Output :



state	avg_speed_delivery
AC	19.76
RO	19.13
AP	18.73
AM	18.61
RR	16.41



Insight :

Businesses can leverage this information to highlight their effective logistics operations and gain a competitive advantage. It also helps identify areas for improving delivery efficiency by comparing with states that have longer delivery times.

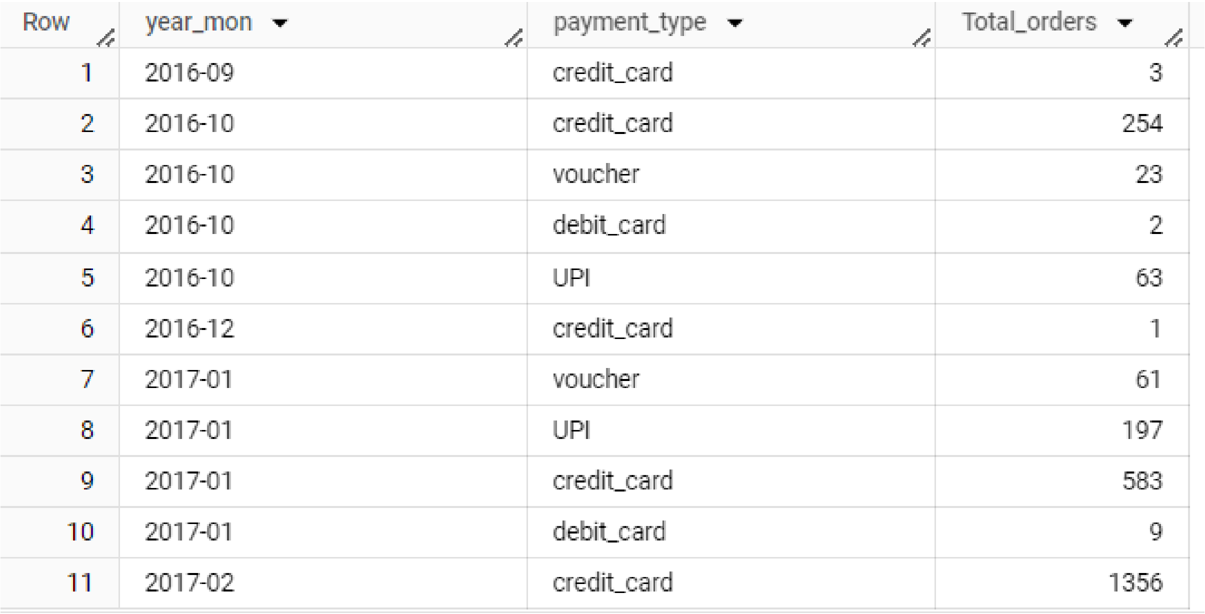
6 Analysis based on the payments:

Q1 Find the month on month no. of orders placed using different payment types

Code :

```
select year_mon,
payment_type,
sum(cnt)as Total_orders
from
(
select
format_timestamp("%Y-%m",order_purchase_timestamp) as year_mon,
payment_type,
count(p.order_id)as cnt
from `target_BCS_ecom.payments` p
join `target_BCS_ecom.orders` o
on p.order_id=o.order_id
group by order_purchase_timestamp,payment_type
)
where payment_type is not null
group by payment_type,year_mon
order by year_mon
```

output :



Row	year_mon	payment_type	Total_orders
1	2016-09	credit_card	3
2	2016-10	credit_card	254
3	2016-10	voucher	23
4	2016-10	debit_card	2
5	2016-10	UPI	63
6	2016-12	credit_card	1
7	2017-01	voucher	61
8	2017-01	UPI	197
9	2017-01	credit_card	583
10	2017-01	debit_card	9
11	2017-02	credit_card	1356

Insights :

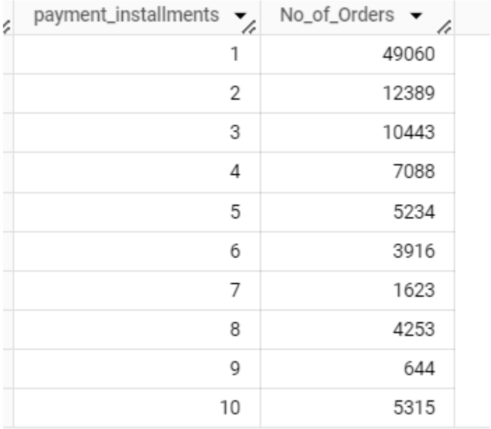
The query provides insights into customer payment preferences by analyzing the monthly distribution of payment types. Identifying trends in payment methods from the monthly distribution of payment types can help businesses gain insight into customer payment preferences.

Q2 Find the no. of orders placed on the basis of the payment instalments that have been paid.

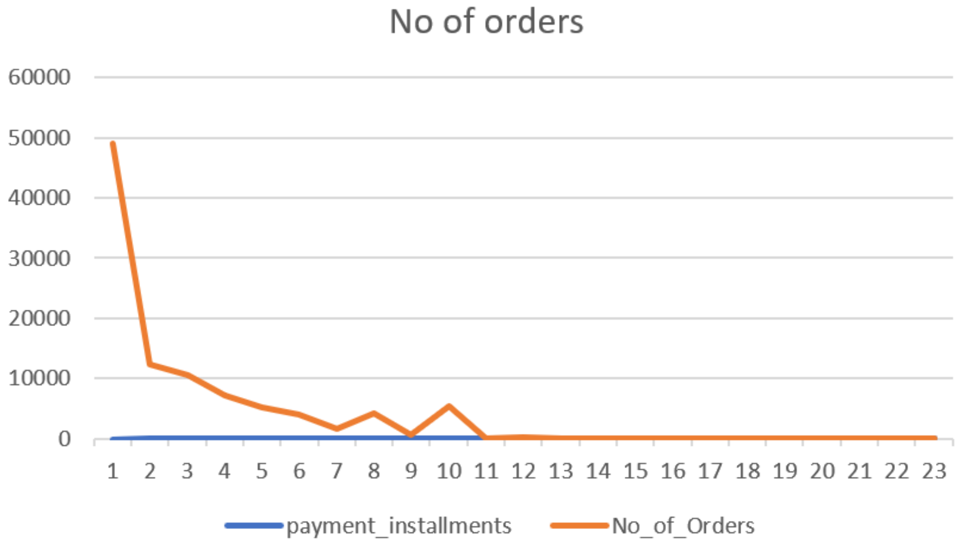
Code :

```
select payment_installments,
count(distinct order_id) as No_of_Orders
from `target_BCS_ecom.payments`
where payment_installments <> 0
group by 1
order by 1
```

Output :



payment_installments	No_of_Orders
1	49060
2	12389
3	10443
4	7088
5	5234
6	3916
7	1623
8	4253
9	644
10	5315



Insight :

As a result of our queries, we see the number of distinct orders for each payment instalment option. We see the distribution of customers who choose different instalment plans for their payments. The most preferred instalment is 1, or at least one out of several has been paid. Based on the data, there is a distinct customer preference for instalments between 9 and 10.