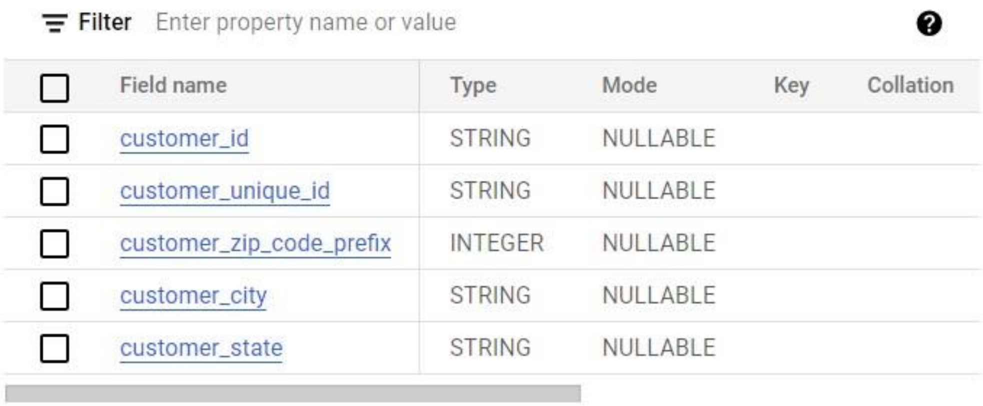
Business Case Study- Target Dataset

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

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



Filter Enter property name or value

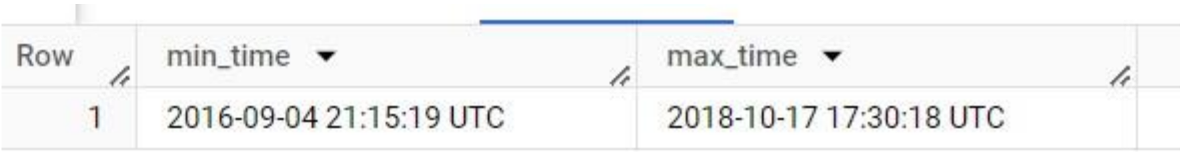
□	Field name	Type	Mode	Key	Collation
□	customer_id	STRING	NULLABLE		
□	customer_unique_id	STRING	NULLABLE		
□	customer_zip_code_prefix	INTEGER	NULLABLE		
□	customer_city	STRING	NULLABLE		
□	customer_state	STRING	NULLABLE		

The data types of customers table from the target dataset are as follows as shown in the above table.
Customer_id – String datatype,
Customer_unique_id- String
Customer_zip_code_prefix – Integer datatype,
Customer_city – String and
Customer_state - string datatype

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

```
select min(order_purchase_timestamp) min_time,
       max(order_purchase_timestamp) max_time
From `target.orders`
```

Output:



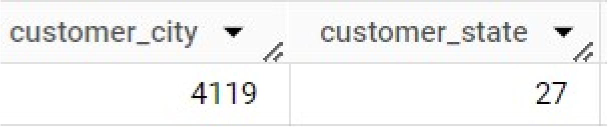
Row	min_time	max_time
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC

Insight:
The time range between which the orders were placed are 2016-09-04 to 2018-10-17. The orders purchases were happened between these two time periods.

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

```
select count(distinct c.customer_city) as customer_city,
       count(distinct c.customer_state) as customer_state
from `target.customers` c
join `target.orders` o
on c.customer_id = o.customer_id
where  o.order_purchase_timestamp  between  '2016-09-04  21:15:19  UTC'
and  '2018-10-17 17:30:18 UTC'
```

Output:



customer_city	customer_state
4119	27

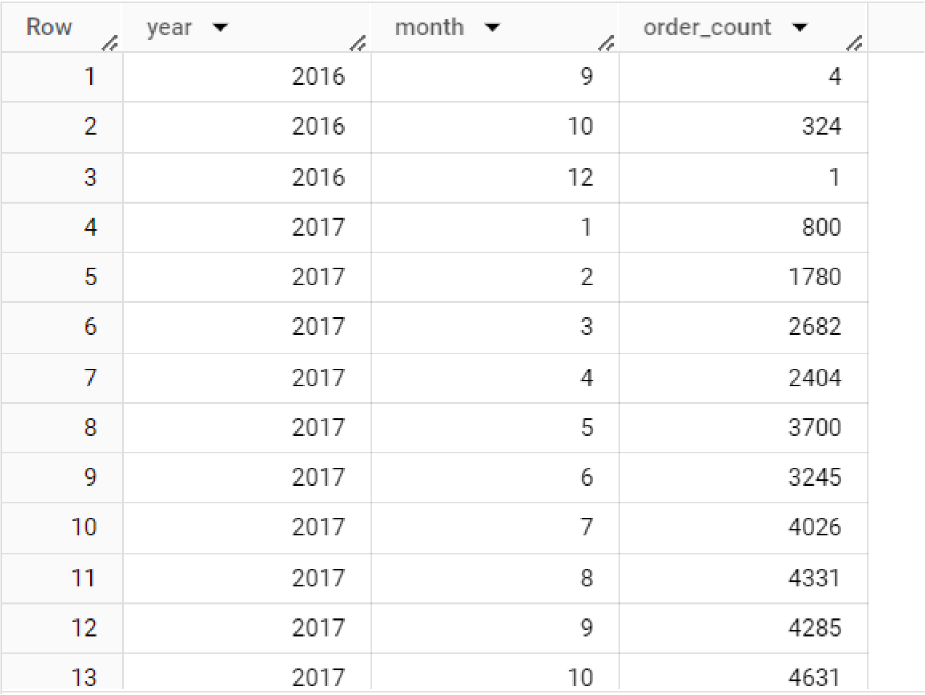
Insight: Then count of cities and states during the given period was found to be 4119 and 27 respectively

2.**In-depthExploration:**

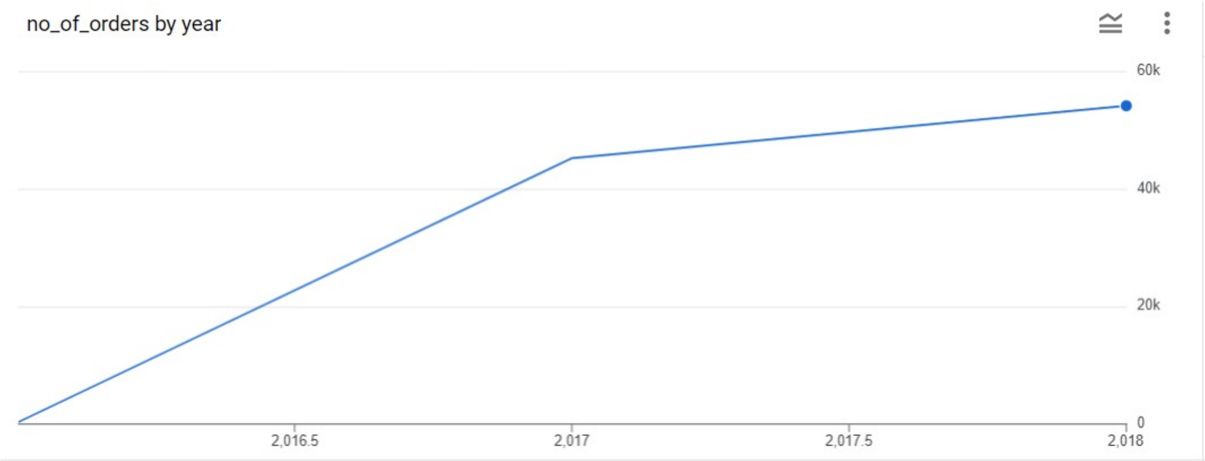
2.1Is there a growing trend in the no. of orders placed over the past years?

```
select distinct year, month,
       count(order_id) over(partition by year) as no_of_orders
from
      (select order_id, extract(year from order_purchase_timestamp) as year,
              extract(month from order_purchase_timestamp) as month
       from `target.orders`) x
order by year, month
```

Output:



Row	year	month	order_count
1	2016	9	4
2	2016	10	324
3	2016	12	1
4	2017	1	800
5	2017	2	1780
6	2017	3	2682
7	2017	4	2404
8	2017	5	3700
9	2017	6	3245
10	2017	7	4026
11	2017	8	4331
12	2017	9	4285
13	2017	10	4631



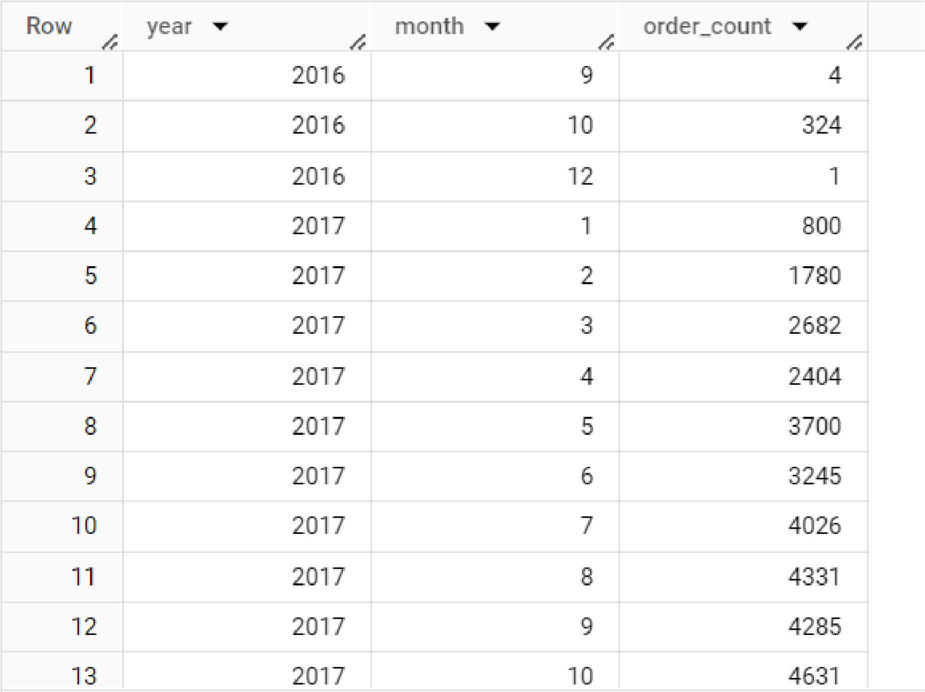
Insight: The no_of_orders from 2016, 2017 and 2018 in monthly order format has grouped in the above table. From the above graph, we can clearly say that the no_of_orders has increased over the years.

Recommendation: We can see a huge increase in sales year by year which indicates that the company has made huge number of customers over the years. They could maintain this growth for the coming years by focusing on the factors that could contribute to revenue generation through sales like customer satisfaction, speed delivery of orders, special offers, finding the right way to endorse their product and identifying their target audience.

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

```
select distinct year, month,

       count(order_id) over(partition by year) as no_of_orders
from
      (select order_id, extract(year from order_purchase_timestamp) as year,
              extract(month from order_purchase_timestamp) as month
       from `target.orders`) x
order by year, month
```

Output:



Row	year	month	order_count
1	2016	9	4
2	2016	10	324
3	2016	12	1
4	2017	1	800
5	2017	2	1780
6	2017	3	2682
7	2017	4	2404
8	2017	5	3700
9	2017	6	3245
10	2017	7	4026
11	2017	8	4331
12	2017	9	4285
13	2017	10	4631

Insight: The table represents the data of order_count of each month (Jan to Dec) from the years 2017 and 2018.

Recommendation: It seems to be the sales happened in an increasing order till the month of August. There is a fluctuation in the order count from September to December. We can bring harmony to the orders placed and increase the sales of products by introducing the launching seasonal sale offers, new ways of promoting, new offers, giving special coupons and discounts to the customers.

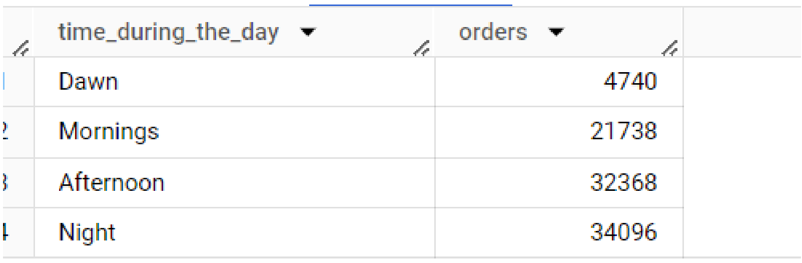
2.3 During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)

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

```
select time_during_the_day, count(order_id) as orders
from
(select order_id, case when hours between '00:00:00' and '06:00:00' then
"Dawn"
                       when hours between '06:00:00' and '07:00:00'
                       then "s"
                       when hours between '07:00:00' and '12:00:00' then
                       "Mornings"
                       when hours between '12:00:00' and '13:00:00' then "d"
                       when hours between '13:00:00' and '18:00:00' then
                        "Afternoon"
                       else "Night"
                       end as time_during_the_day
```

```
         from
               (select order_id, extract(time from order_purchase_timestamp)
as hours
                       from `target.orders`
                       ) a
) b
where time_during_the_day not in ("s","d")
group by time_during_the_day
order by orders asc
```

Output:



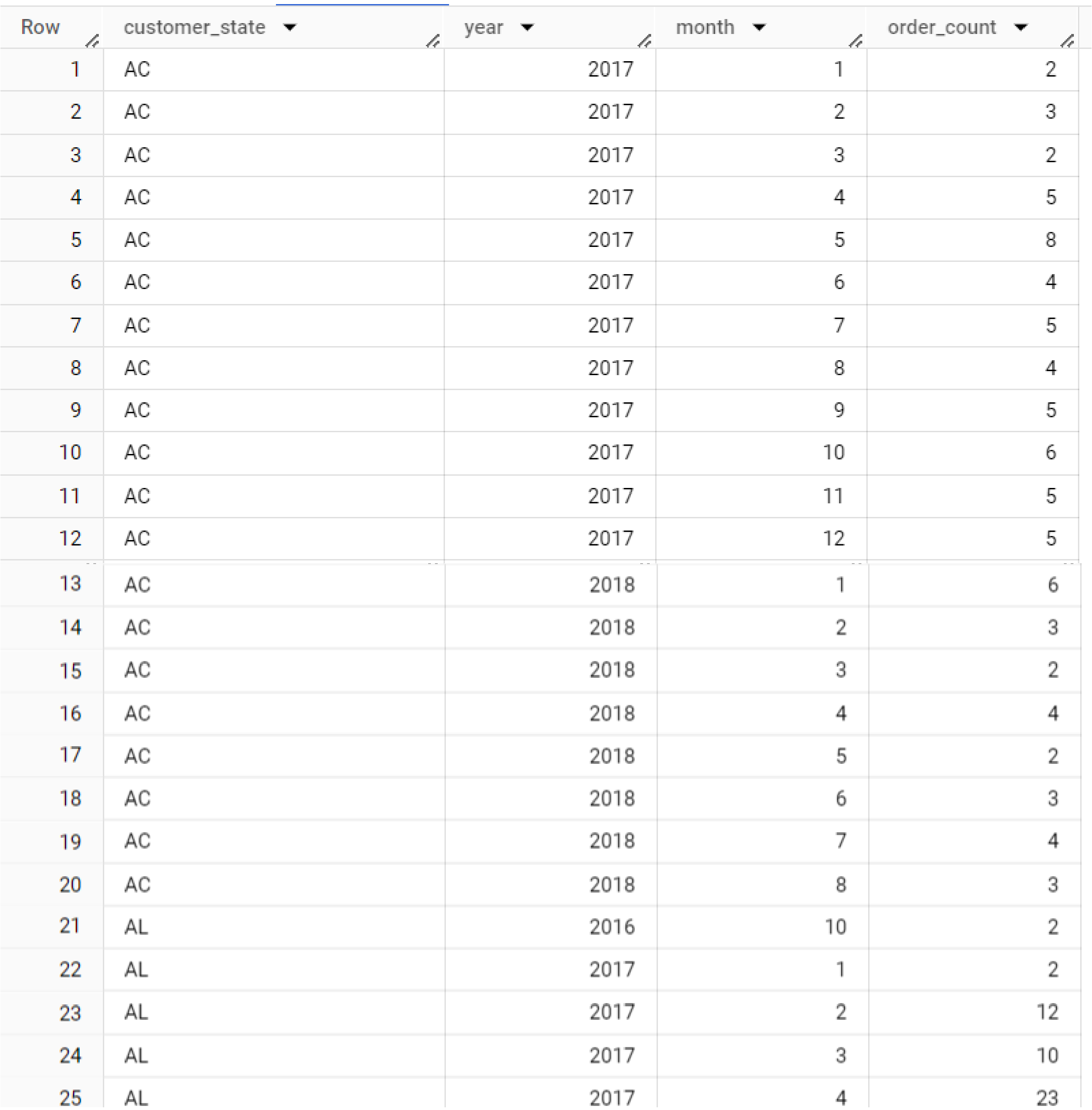
	time_during_the_day	orders
1	Dawn	4740
2	Mornings	21738
3	Afternoon	32368
4	Night	34096

Insight: The number of orders placed by Brazilian customers during the different time hours of days were shown in the table. There seem to be a lot of purchases made during the afternoon and night hours of the day.
Recommendation: Since there are a lot of sales happening during the second half of the day the company could introduce new offers like mid-day sales and price drops on the products for a few hours a day to increase sales. They can also focus on increasing the sales during Dawn and Morning hours by giving special discounts on the products which are sold more during those hours.

3. Evolution of E-commerce orders in the Brazil region:
3.1 Get the month-on-month no. of orders placed in each state.

```
select c.customer_state, extract(year from order_purchase_timestamp) as year,

extract(month from order_purchase_timestamp) as month,

count(o.order_id) order_count

from target.orders o
join target.customers c
on c.customer_id = o.customer_id
group by customer_state, year, month
order by customer_state, year, month
```

Output:



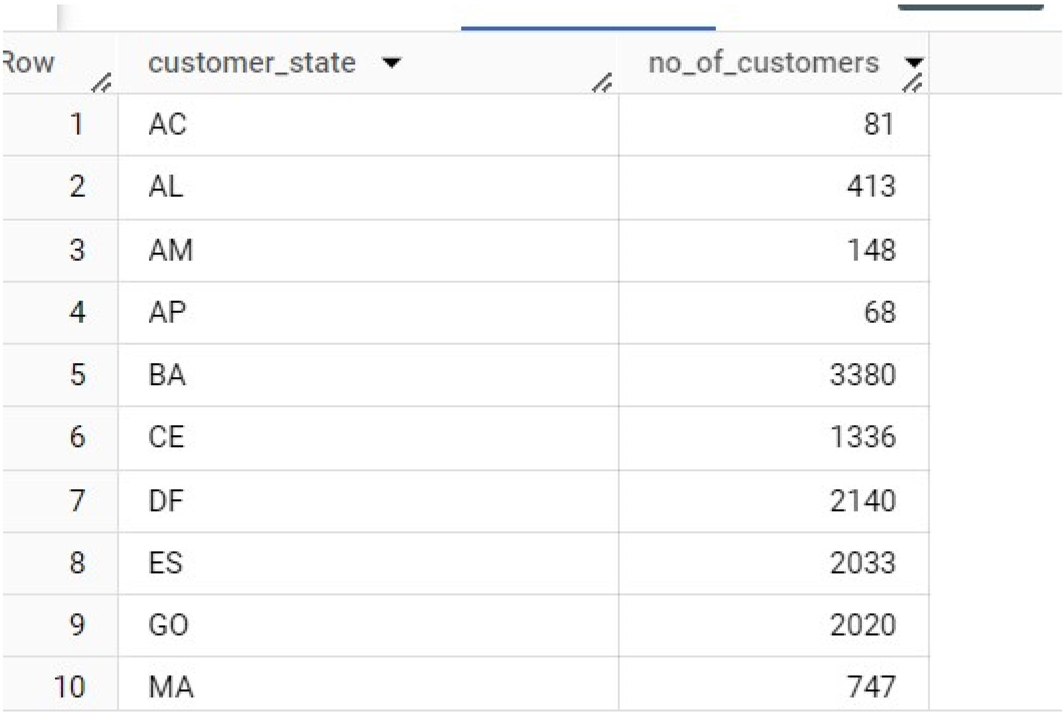
Row	customer_state	year	month	order_count
1	AC	2017	1	2
2	AC	2017	2	3
3	AC	2017	3	2
4	AC	2017	4	5
5	AC	2017	5	8
6	AC	2017	6	4
7	AC	2017	7	5
8	AC	2017	8	4
9	AC	2017	9	5
10	AC	2017	10	6
11	AC	2017	11	5
12	AC	2017	12	5
13	AC	2018	1	6
14	AC	2018	2	3
15	AC	2018	3	2
16	AC	2018	4	4
17	AC	2018	5	2
18	AC	2018	6	3
19	AC	2018	7	4
20	AC	2018	8	3
21	AL	2016	10	2
22	AL	2017	1	2
23	AL	2017	2	12
24	AL	2017	3	10
25	AL	2017	4	23

Insight: The numbers of orders placed month- on- month of each year of each state was found. The above table is a sample representation of the output.
Recommendation: From the results we can identify the order distribution pattern of all the months of each year. We can identify the trends in the order count

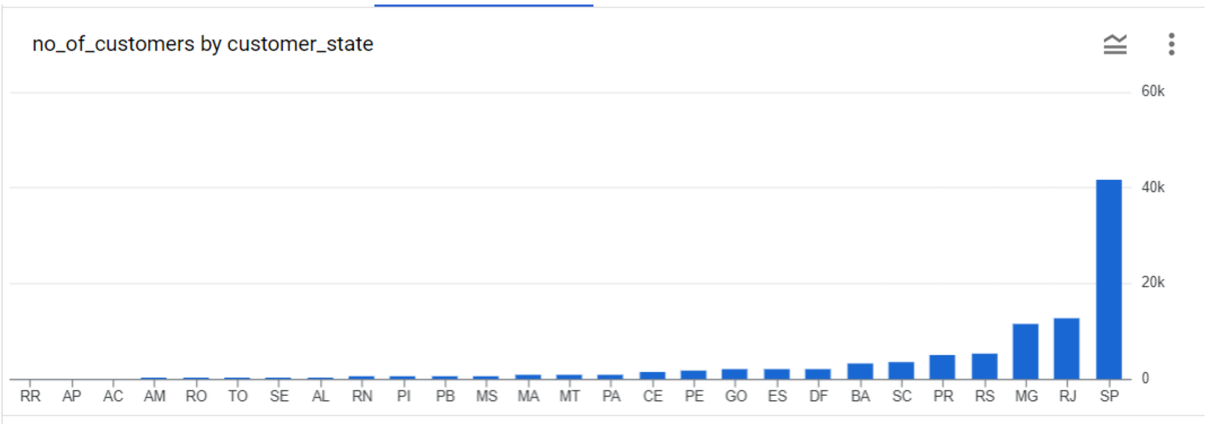
3.2 How are the customers distributed across all the states?

```
select customer_state,
       count(customer_id) as no_of_customers
from `target.customers`
group by customer_state
order by no_of_customers
```

Output:



Row	customer_state	no_of_customers
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



Insight: The table represents the number of customers from each state. From the above graph it is evident that the state SP and RR contain the highest and lowest number of customers from this table. From the data 3 different states have 1-100 customers, 12 different states have 100-1000 customers, 8 different states have 1000-5500 customers above 5500 we have 3 different states

Recommendation: The number of customers is found to be very low in the states like AP, AC, AM (according to the above table). So, they could focus on these states and develop new strategies like promoting their brand through various mediums like television, social media, campaigns, newspapers, etc. They can launch sales like clearance sales and festival sales by increasing the discounts on the products.

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

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

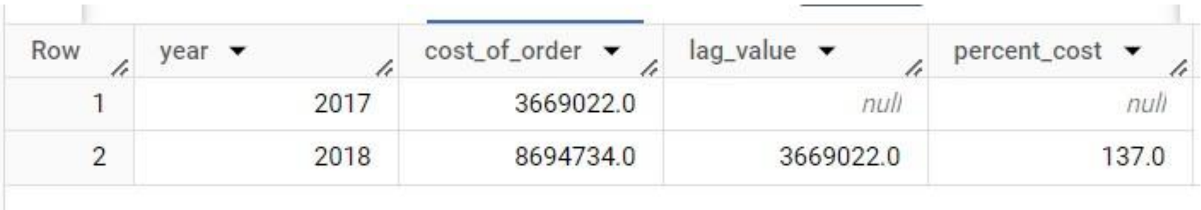
```
select b.year,
       b.cost_of_order,
       lag(b.cost_of_order)over(order by b.year) as
lag_value,   round((b.cost_of_order - (lag(b.cost_of_order)over(order by
b.year)))/(lag(b.cost_of_order)over(order by b.year))*100) as percent_cost

from
     (
       select a.year, round(sum(a.payment_value)) as cost_of_order,
       from
              (select extract(year from o.order_purchase_timestamp)as
               year,
                extract(month from o.order_purchase_timestamp) as month,
               p.payment_value as payment_value
               from `target.orders` o
               join `target.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
              ) a
group by a.year)b
order by b.year
```

Aliases: sum(payment_value) as cost_of_orders,

Lag(cost_of_orders)over(order by year) as lag_value

% Calculation: Percent_cost = cost_of_orders – lag_value/lag_value *100

Output:



Row	year	cost_of_order	lag_value	percent_cost
1	2017	3669022.0	null	null
2	2018	8694734.0	3669022.0	137.0

Insight: From the above table, we can observe that there is a huge increase in cost_of_orders.

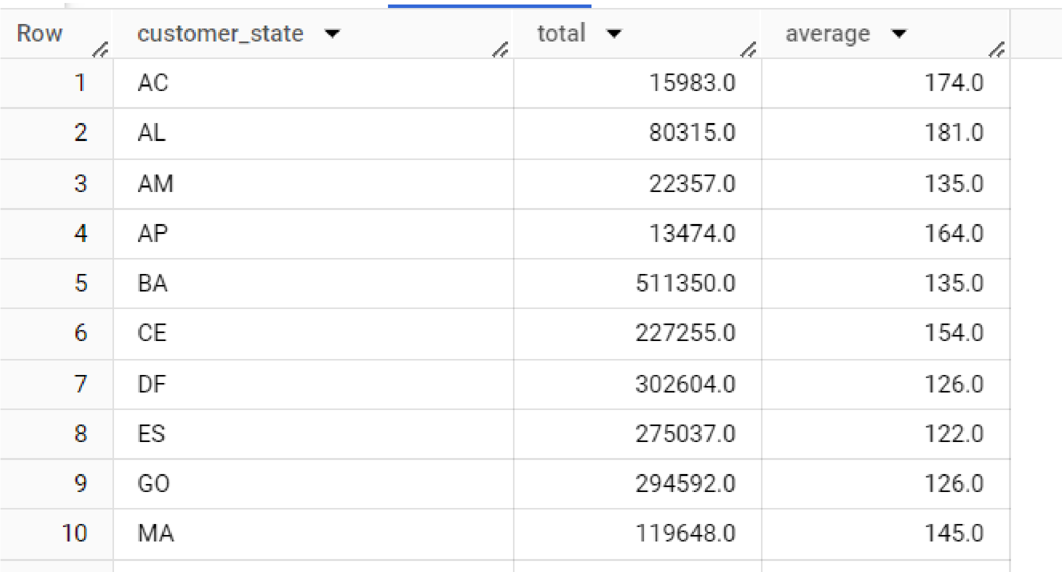
The % of cost increase was found to be 137%.

Recommendations: The cost of orders has taken a good leap over the years and it is evident that 2018's payment value is comparatively much high then the previous year (2017). We can make sure that the company maintains a similar or better payment value in the year to come by increasing the marketing strategies and promotions through ads and print and electronic media

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

```
select c.customer_state,
      round(sum(oi.price)) as total,
       round(avg(oi.price)) as average
from `target.customers` c
join `target.orders` o
on c.customer_id = o.customer_id
join `target.order_items` oi
on o.order_id = oi.order_id
group by customer_state
order by customer_state asc
```

Output:



Row	customer_state	total	average
1	AC	15983.0	174.0
2	AL	80315.0	181.0
3	AM	22357.0	135.0
4	AP	13474.0	164.0
5	BA	511350.0	135.0
6	CE	227255.0	154.0
7	DF	302604.0	126.0
8	ES	275037.0	122.0
9	GO	294592.0	126.0
10	MA	119648.0	145.0

Insight: The table provides information about the total order_price and average order_price of each state. From the above sample table, it is evident that BA and AP are the states with the highest and lowest order_price. AL and ES are the states with the highest and lowest Avg order_price.

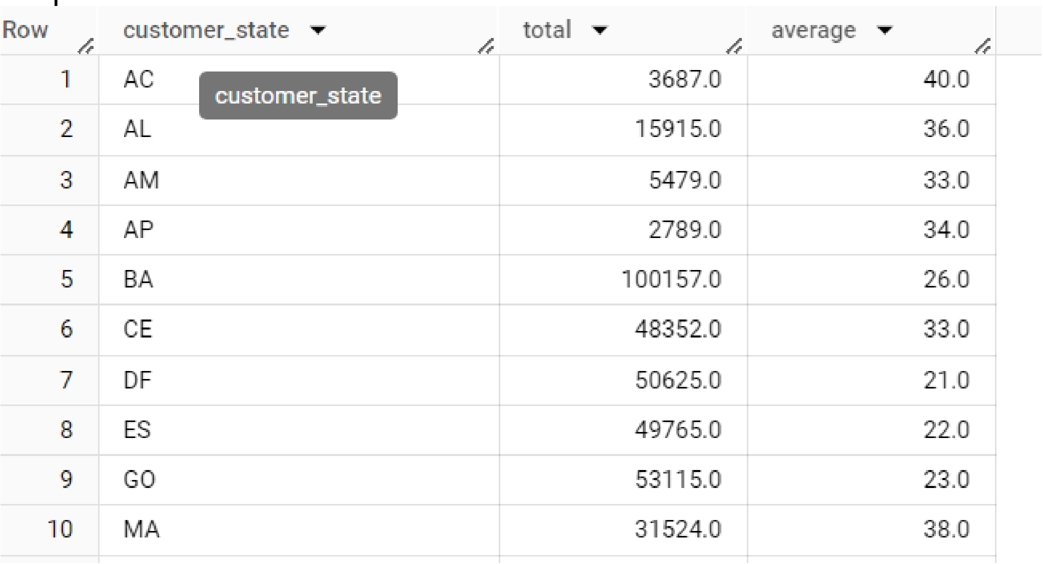
Recommendation: we can increase the sales of goods in the states with the low total order_price value to increase revenue. The company can focus on marketing in the low total order price states to increase the total orders_price revenue and also in they can promote more in the states with high sale values to maintain or to increase revenue. They can attract customers by providing special discounts and giving items, cashback.

The average price value represents the average price of goods in the market and helps in understanding the price fixation of their products. They can focus on the states with the high average price value since there is a lot of chance of getting high revenue generation in those states.

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

```
select c.customer_state,
       round(sum(oi.freight_value)) as total,
       round(avg(oi.freight_value)) as average
from `target.customers` c
join `target.orders` o
on c.customer_id = o.customer_id
join `target.order_items` oi
on o.order_id = oi.order_id
group by customer_state
order by customer_state
```

Output:



Row	customer_state	total	average
1	AC	3687.0	40.0
2	AL	15915.0	36.0
3	AM	5479.0	33.0
4	AP	2789.0	34.0
5	BA	100157.0	26.0
6	CE	48352.0	33.0
7	DF	50625.0	21.0
8	ES	49765.0	22.0
9	GO	53115.0	23.0
10	MA	31524.0	38.0

Insight: The total and average freight values are listed above. From the table we can get a conclusion that the state BA has the highest total freight values and the state AC has the highest freight value.

Recommendation: The highest freight value indicates that the shipping charges of these states are very high and need to be taken care of to increase the company's overall profits. It helps in understanding the excess shipping charges and can be cut down if not necessarily needed. The average freight values indicate how much the consumers are charged for the delivery of their order. Depending on the state they live and the distance the order has to be delivered the freight value can either be increased or decreased in a way that can benefit both the company and the consumers.

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

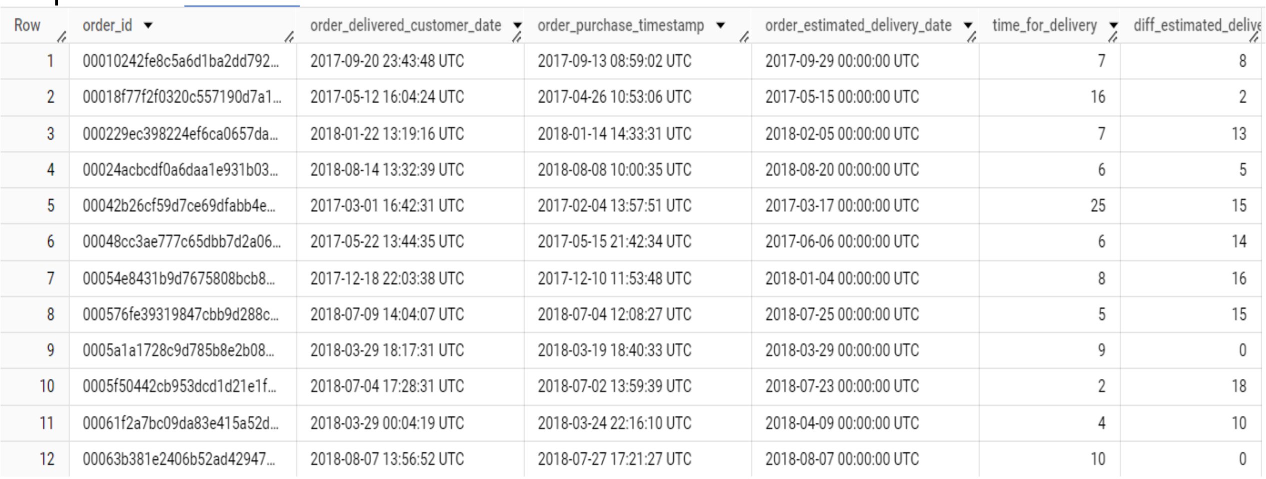
5.1 Find the no. of days taken to deliver each order from the order's purchase date as delivery time.
Also, calculate the difference (in days) between the estimated & actual delivery date of an order.
Do this in a single query.

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,
       order_delivered_customer_date,
       order_purchase_timestamp,
       order_estimated_delivery_date,
         date_diff(order_delivered_customer_date, order_purchase_timestamp,
day) as time_for_delivery,
         date_diff(order_estimated_delivery_date,order_delivered_customer_d
ate, day) as diff_estimated_delivery
from `target.orders`
where   date_diff(order_delivered_customer_date,   order_purchase_timestamp,
day) is not null
order by order_id
```

Output:



Row	order_id	order_delivered_customer_date	order_purchase_timestamp	order_estimated_delivery_date	time_for_delivery	diff_estimated_delive
1	00010242fe8c5a6d1ba2dd792...	2017-09-20 23:43:48 UTC	2017-09-13 08:59:02 UTC	2017-09-29 00:00:00 UTC	7	8
2	00018f77f2f0320c557190d7a1...	2017-05-12 16:04:24 UTC	2017-04-26 10:53:06 UTC	2017-05-15 00:00:00 UTC	16	2
3	000229ec398224ef6ca0657da...	2018-01-22 13:19:16 UTC	2018-01-14 14:33:31 UTC	2018-02-05 00:00:00 UTC	7	13
4	00024acbcdf0a6daa1e931b03...	2018-08-14 13:32:39 UTC	2018-08-08 10:00:35 UTC	2018-08-20 00:00:00 UTC	6	5
5	00042b26cf59d7ce69dfabb4e...	2017-03-01 16:42:31 UTC	2017-02-04 13:57:51 UTC	2017-03-17 00:00:00 UTC	25	15
6	00048cc3ae777c65dbb7d2a06...	2017-05-22 13:44:35 UTC	2017-05-15 21:42:34 UTC	2017-06-06 00:00:00 UTC	6	14
7	00054e8431b9d7675808bcb8...	2017-12-18 22:03:38 UTC	2017-12-10 11:53:48 UTC	2018-01-04 00:00:00 UTC	8	16
8	000576fe39319847cbb9d288c...	2018-07-09 14:04:07 UTC	2018-07-04 12:08:27 UTC	2018-07-25 00:00:00 UTC	5	15
9	0005a1a1728c9d785b8e2b08...	2018-03-29 18:17:31 UTC	2018-03-19 18:40:33 UTC	2018-03-29 00:00:00 UTC	9	0
10	0005f50442cb953dcd1d21e1f...	2018-07-04 17:28:31 UTC	2018-07-02 13:59:39 UTC	2018-07-23 00:00:00 UTC	2	18
11	00061f2a7bc09da83e415a52d...	2018-03-29 00:04:19 UTC	2018-03-24 22:16:10 UTC	2018-04-09 00:00:00 UTC	4	10
12	00063b381e2406b52ad42947...	2018-08-07 13:56:52 UTC	2018-07-27 17:21:27 UTC	2018-08-07 00:00:00 UTC	10	0

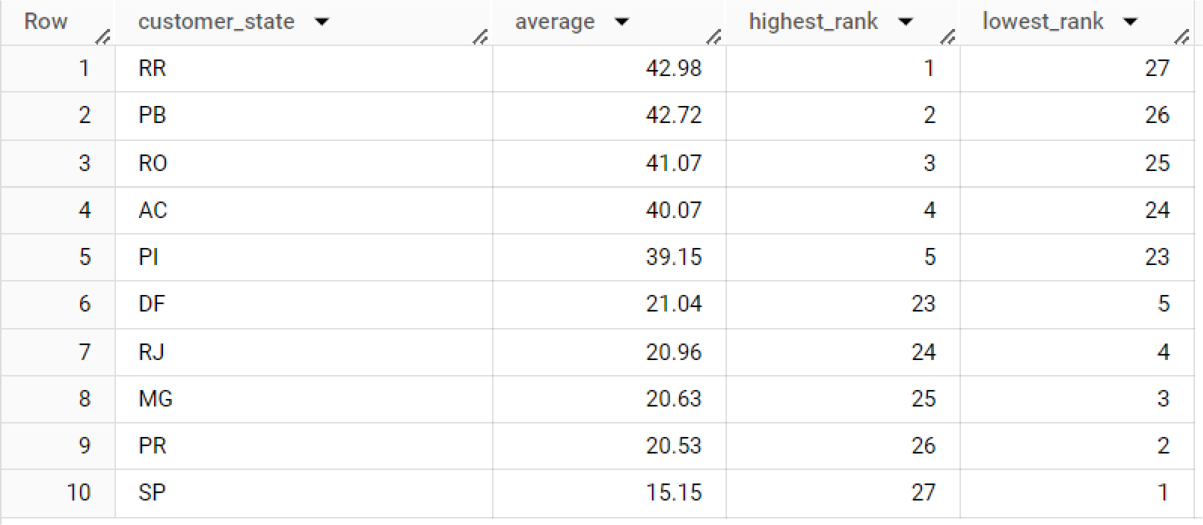
Insight: The table shows the data of the days difference between the estimated and the actual delivery time, the difference in the days between the purchase date and the delivery date. The higher the difference in days between the columns indicates the faster delivery.

Recommendation: The company can identify the orders that are delivered faster and their respective addresses. We can identify the reasons due to which the delivery is getting delayed. The company can hire extra man- power the deliver the products even faster to the addresses where the order delivery takes more time than usual.

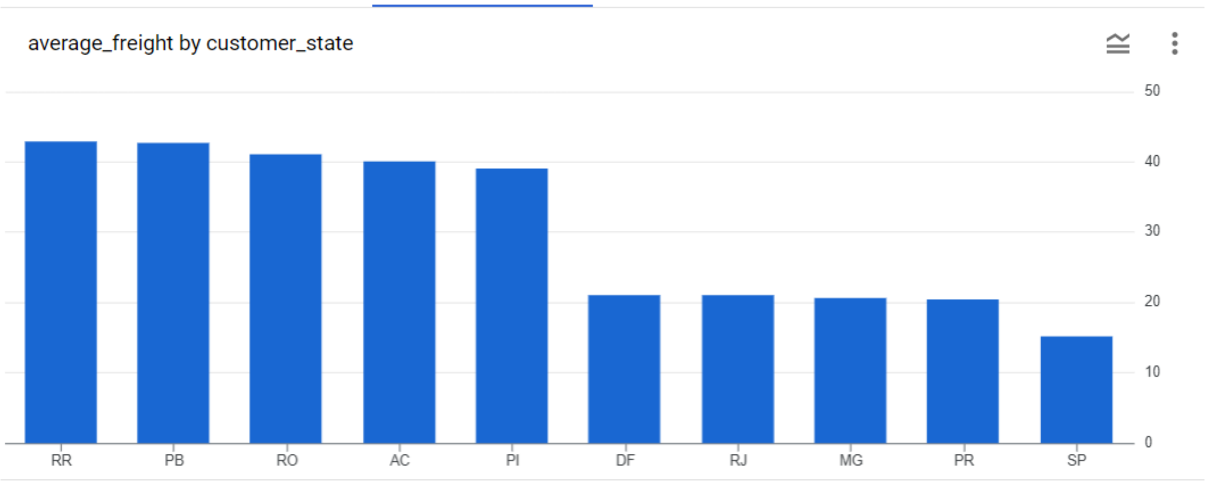
5.2Find out the top 5 states with the highest & lowest average freight value.

```
select *
from
(select customer_state, average_freight,
      dense_rank()over(order by average_freight desc) as highest_rank,
      dense_rank()over (order by x.average_freight) as lowest_rank
from
(select c.customer_state,
       round(avg(oi.freight_value), 2) as average_freight
       from `target.customers` c
       join `target.orders` o
       on c.customer_id = o.customer_id
       join `target.order_items` oi
       on o.order_id = oi.order_id
       group by c.customer_state) x )y
where highest_rank <= 5 or lowest_rank <= 5
order by highest_rank, lowest_rank
```

Output:



Row	customer_state	average	highest_rank	lowest_rank
1	RR	42.98	1	27
2	PB	42.72	2	26
3	RO	41.07	3	25
4	AC	40.07	4	24
5	PI	39.15	5	23
6	DF	21.04	23	5
7	RJ	20.96	24	4
8	MG	20.63	25	3
9	PR	20.53	26	2
10	SP	15.15	27	1



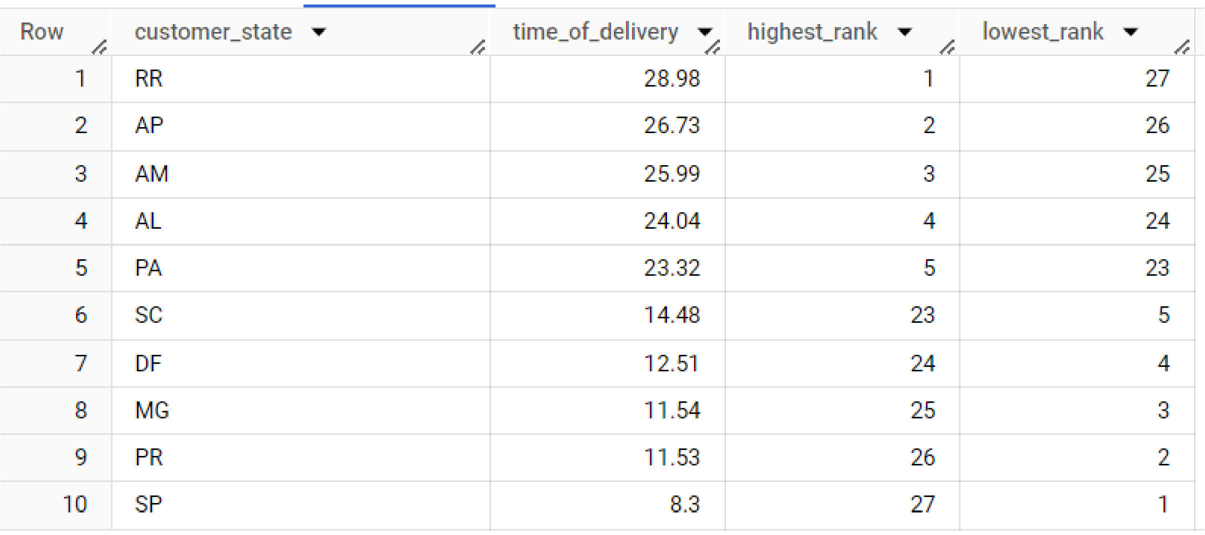
Insight: These are states with the highest and lowest average freight value. The freight values vary from state to state depending on the distance to deliver the order. Highest freight value indicates the high transportation costs of goods may be due to the amount od distance they have to be delivered.

Recommendation: We can estimate the average delivery price for each state. We can find the ways to reduce the cost of the delivery charges to customers with high freight values to provide customer satisfaction. Even if we cannot reduce the charges due to the distance to be covered to deliver the product, we can give the customers bonus coins, special offers on their credit cards, and coupons. We can also get in touch with the competitive logistics companies to the current logistic partner who can provide us the better price deals and better performances.

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

```
select *
from
(select customer_state, time_of_delivery, dense_rank()over(order by
x.time_of_delivery desc) as highest_rank,
       dense_rank()over(order by x.time_of_delivery) as lowest_rank
from(
      select c.customer_state,
       round(avg(date_diff(o.order_delivered_customer_date,
o.order_purchase_timestamp, day)),2) as time_of_delivery
        from `target.orders` o
          join `target.customers` c
           on o.customer_id = c.customer_id
          where o.order_delivered_customer_date is not null
          group by c.customer_state) x ) y
  where highest_rank <= 5 or lowest_rank <= 5
order by highest_rank, lowest_rank
```

Output:



Row	customer_state	time_of_delivery	highest_rank	lowest_rank
1	RR	28.98	1	27
2	AP	26.73	2	26
3	AM	25.99	3	25
4	AL	24.04	4	24
5	PA	23.32	5	23
6	SC	14.48	23	5
7	DF	12.51	24	4
8	MG	11.54	25	3
9	PR	11.53	26	2
10	SP	8.3	27	1

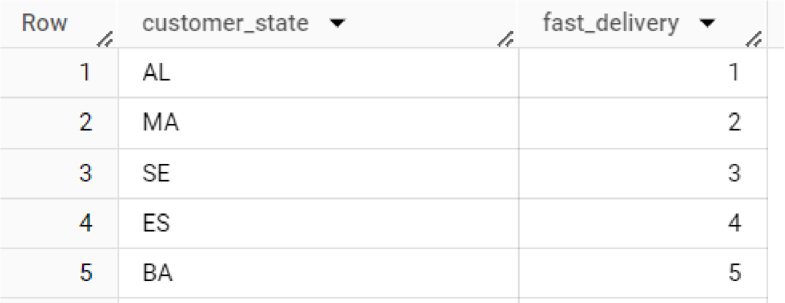
Insight: The output from the query to get the top 5 highest and lowest time_delivery was displayed in the tabular format above.
The states with the lowest value in the highest_rank column represent the fastest delivery of the orders. RR is the state where delivery seems to happen very fast.
The states with the lowest value in the lowest_rank column represent the slowest delivery of the orders. SP is the state with slowest delivery time.
Recommendation: In the states with the slowest delivery, we can reduce the delivery time by increasing the number of delivery boys in those states. Getting in

touch with the logistics department and encouraging them to work more hours in a day by giving them extra salaries.

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.

```
select customer_state,
       dense_rank()over(order by avg_delivery) as fast_delivery
from
 (select c.customer_state,
 round(avg(date_diff(order_delivered_customer_date,
order_estimated_delivery_date, day)),2) as avg_delivery
from `target.orders` o
join target.customers  c
on o.customer_id = c.customer_id
where o.order_delivered_customer_date is not null
group by customer_state) x
order by fast_delivery
```

Output:



Row	customer_state	fast_delivery
1	AL	1
2	MA	2
3	SE	3
4	ES	4
5	BA	5

The listed cities are the ones where the delivery of the orders happened very fast compared to the other states. AL, MA, SE, ES, BA these are the top 5 states with fastest delivery time

Recommendation: Based on this data, we can estimate the delivery time in the other states are not so good compared to this. So, we can increase the logistics services and increase the shipping stuff as per the requirement in the other states to increase customer satisfaction so that they can shop again. If at all, there is a very slow delivery then we can encourage the customers by giving them coupons and vouchers as compensation.

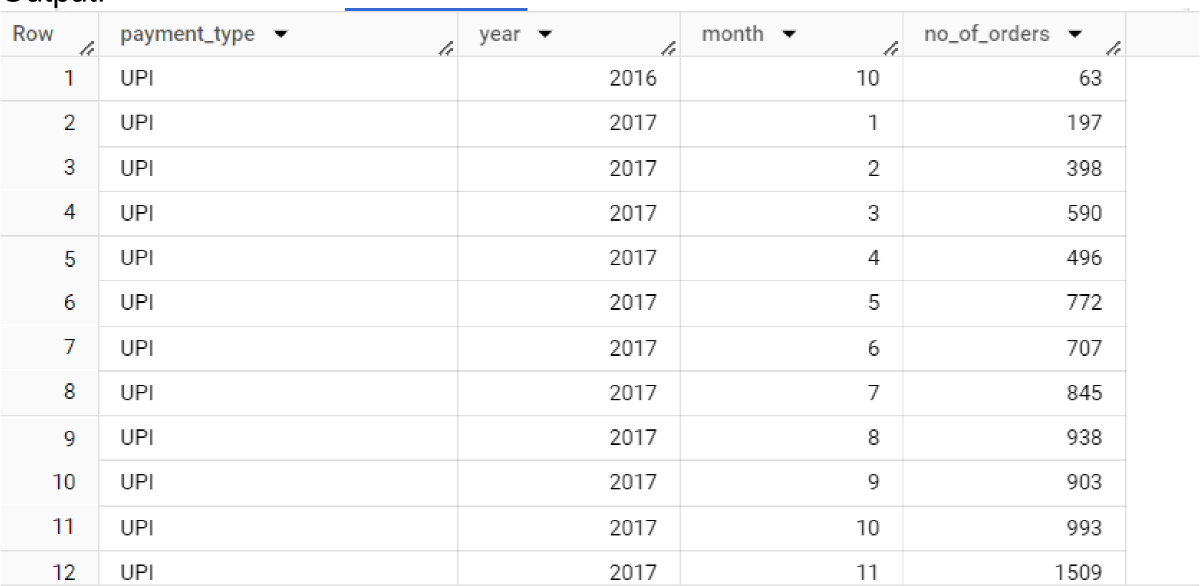
6.Analysis based on the payments:

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

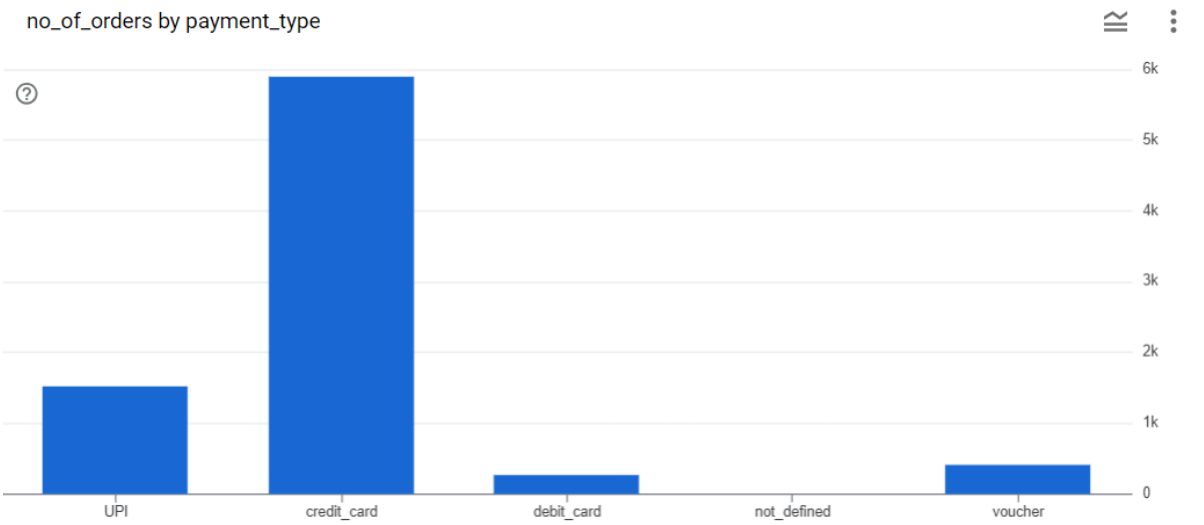
```
select p.payment_type,
   extract(year from o.order_purchase_timestamp) as year,
   extract(month from o.order_purchase_timestamp) as month,
count(o.order_id) as no_of_orders
from `target.orders` o
join target.payments p
```

```
on o.order_id = p.order_id
group by payment_type, year, month
order by payment_type, year, month
```

Output:



Row	payment_type	year	month	no_of_orders
1	UPI	2016	10	63
2	UPI	2017	1	197
3	UPI	2017	2	398
4	UPI	2017	3	590
5	UPI	2017	4	496
6	UPI	2017	5	772
7	UPI	2017	6	707
8	UPI	2017	7	845
9	UPI	2017	8	938
10	UPI	2017	9	903
11	UPI	2017	10	993
12	UPI	2017	11	1509



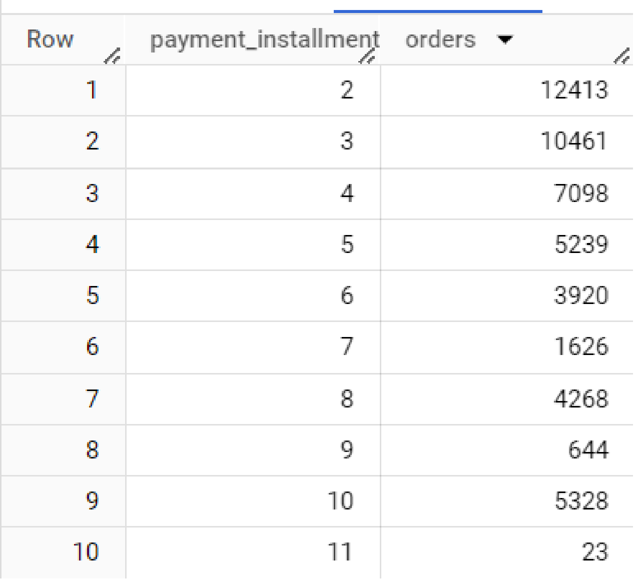
Insight: The above table represents the sample data of the output. From the table, we can observe the payments that happened using UPI for different months in the year 2017 and 2016. The graph represents the no.of orders placed using different payment methods. We can make a conclusion that a greater number of orders were placed using the credit cards and UPI stands second to it.

Recommendation: From the above analysis, the company can understand the business revenue throw various payment modes based on customer preference. They can categorize their consumers based on payment type and provide them beneficial offers on shopping like cashback offers, movie ticket coupons, food coupons, etc. Since, the company has a lot customers suing credit cards, they can tie up with some credit card company and provide some beneficial offers to the customers like extra discounts, instant cashbacks.

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

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

Output:



Row	payment_installment	orders
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	11	23

Insight: The highest number of orders were placed by customers who were paying the bill amount in 2 instalments. We are not considering the payment_installments 0 and 1 since they paid the bill amount all at once and cannot be considered as installment.

Recommendation: we can encourage other customers by launching new payment methods, no-cost EMI's, special offers on credit cards. Reducing the rate of interest, providing gifts and vouchers to the customers.