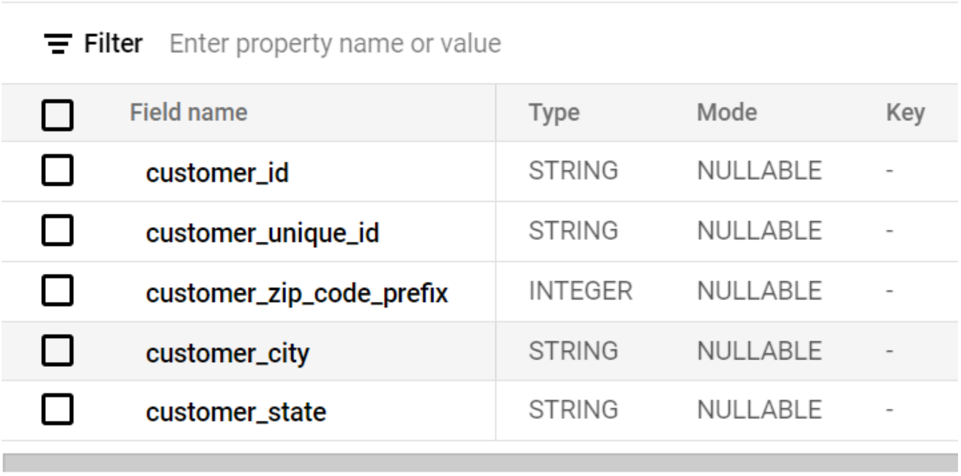
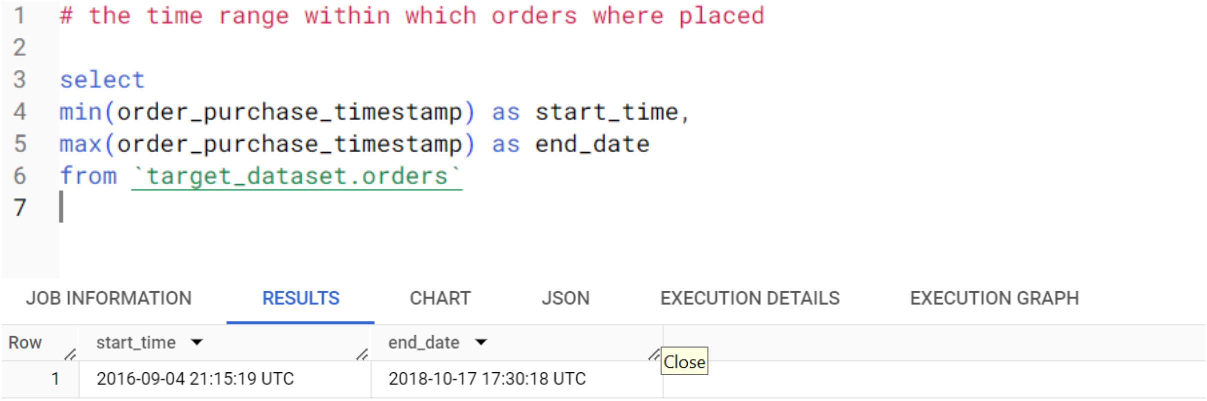
1. Data type of all columns in the "customers" table.



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

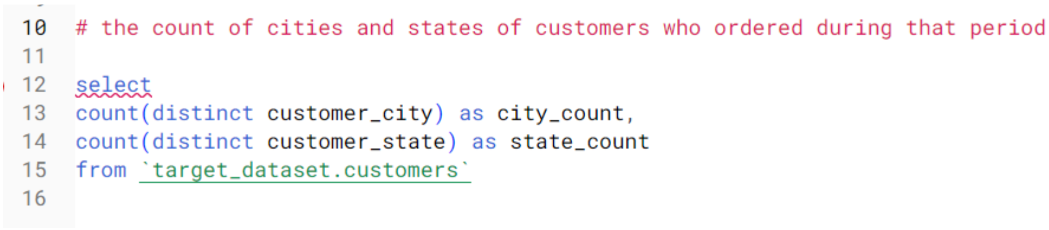
2. Get the time range between which the orders were placed.



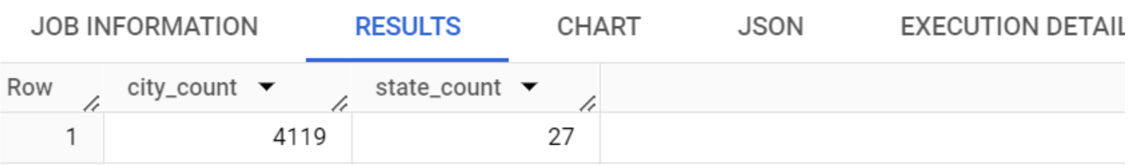
```sql
# the time range within which orders where placed

select
min(order_purchase_timestamp) as start_time,
max(order_purchase_timestamp) as end_date
from `target_dataset.orders`

```

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

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

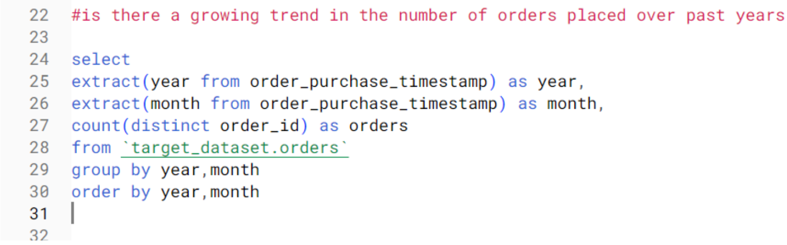


```sql
# the count of cities and states of customers who ordered during that period

select
count(distinct customer_city) as city_count,
count(distinct customer_state) as state_count
from `target_dataset.customers`

```

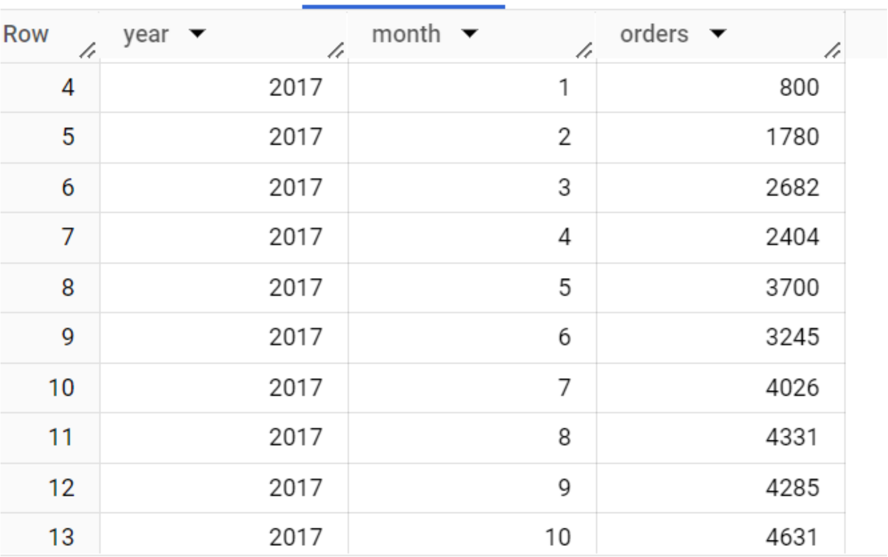


Row	city_count	state_count
1	4119	27

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



```
#is there a growing trend in the number of orders placed over past years

select
extract(year from order_purchase_timestamp) as year,
extract(month from order_purchase_timestamp) as month,
count(distinct order_id) as orders
from `target_dataset.orders`
group by year,month
order by year,month
```

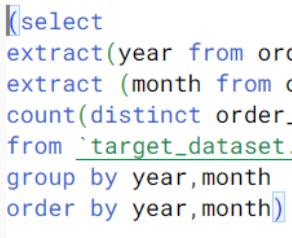


Row	year	month	orders
4	2017	1	800
5	2017	2	1780
6	2017	3	2682
7	2017	4	2404
8	2017	5	3700
9	2017	6	3245
10	2017	7	4026
11	2017	8	4331
12	2017	9	4285
13	2017	10	4631

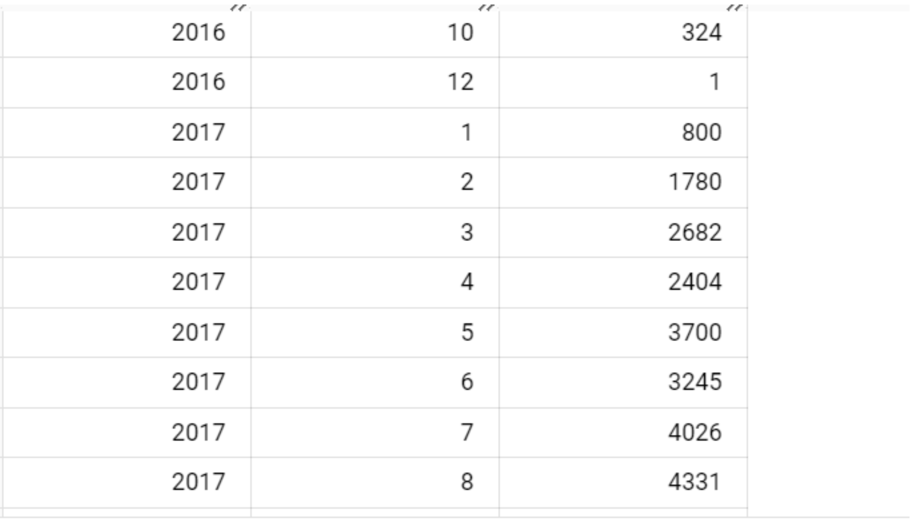
There is clearly a growing trend in the number of orders from April, 2017. The orders increased from 800 per month in January,2017 to 4285 in December,2017 that is almost 5 times in one year.

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

I have taken 2 approaches to solve this problem. First we can see seasonality on the basis of the how the number of orders increased with each passing month till target was in business in brazil.

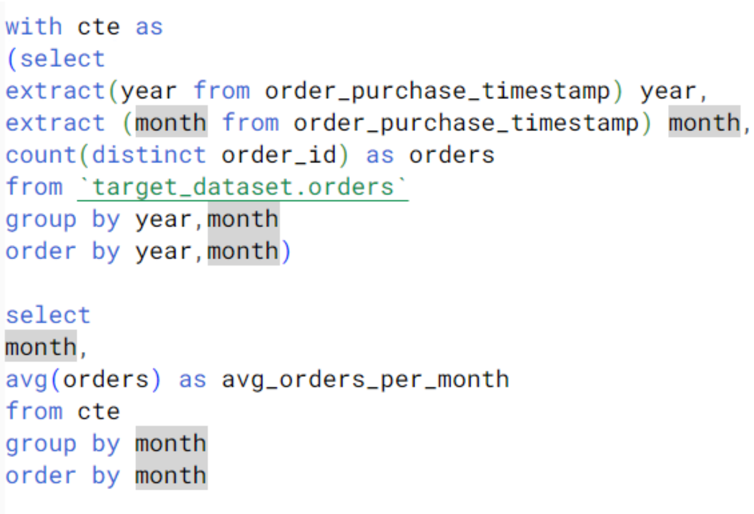


```
(select
extract(year from order_purchase_timestamp) year,
extract (month from order_purchase_timestamp) month,
count(distinct order_id) as orders
from `target_dataset.orders`
group by year,month
order by year,month)
```

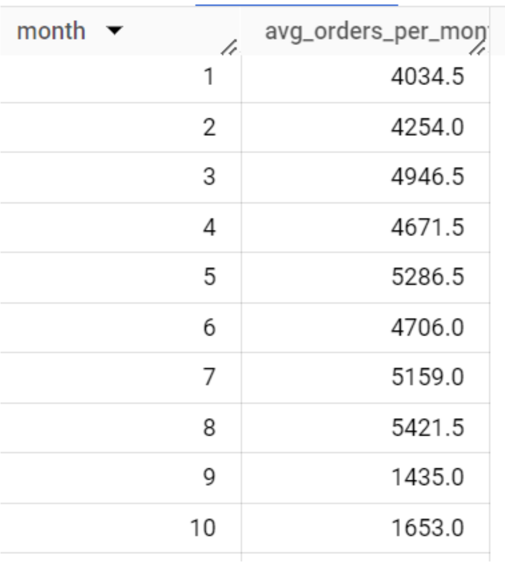
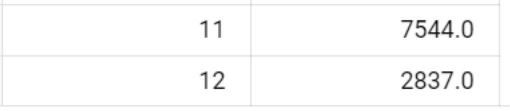


2016	10	324
2016	12	1
2017	1	800
2017	2	1780
2017	3	2682
2017	4	2404
2017	5	3700
2017	6	3245
2017	7	4026
2017	8	4331

My second approach is to come up with average number of orders that have been placed over the years based on each month. This will show how the average orders changed with each month.



```
with cte as
(select
extract(year from order_purchase_timestamp) year,
extract (month from order_purchase_timestamp) month,
count(distinct order_id) as orders
from `target_dataset.orders`
group by year,month
order by year,month)

select
month,
avg(orders) as avg_orders_per_month
from cte
group by month
order by month
```

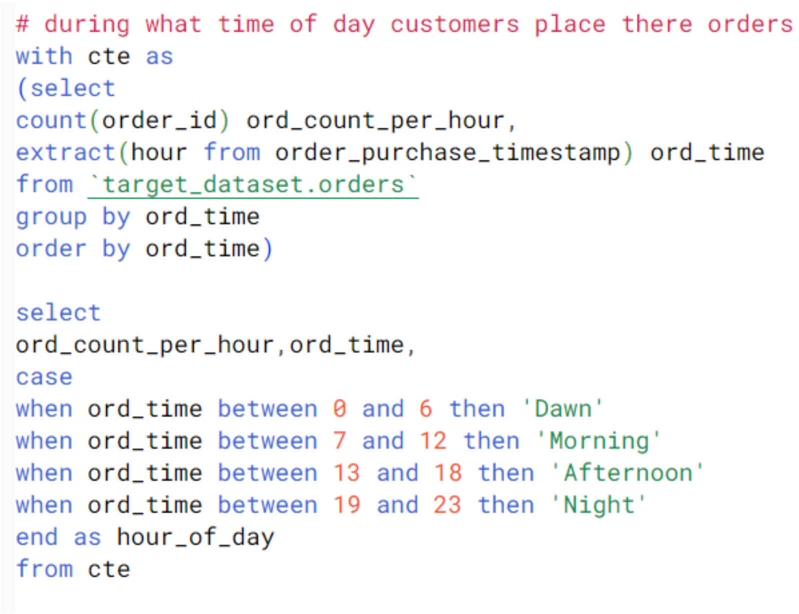
month	avg_orders_per_mon
1	4034.5
2	4254.0
3	4946.5
4	4671.5
5	5286.5
6	4706.0
7	5159.0
8	5421.5
9	1435.0
10	1653.0
11	7544.0
12	2837.0

Insight from data:

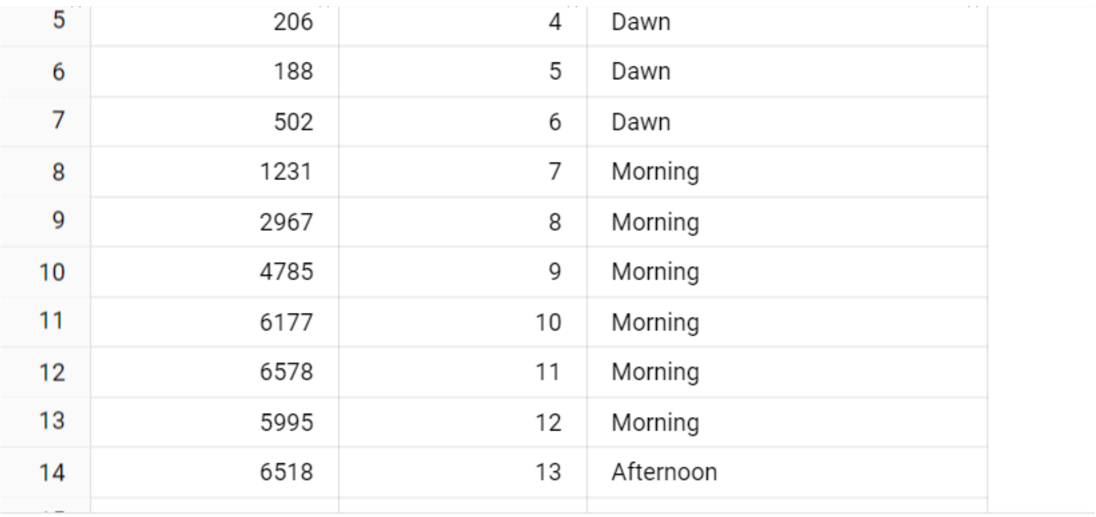
We can see with average orders per month over the years there are 3 months i.e. september, october and december when the orders were quite less as compared to the orders received every month.

We can see November clocked the most number of orders .

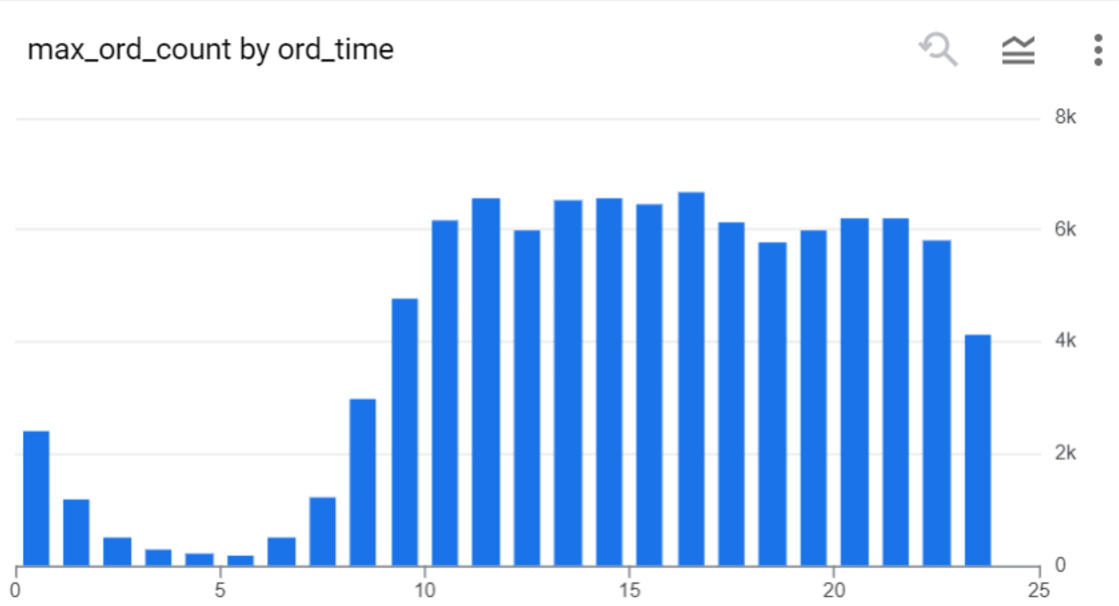
6.During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)



```
# during what time of day customers place there orders
with cte as
(select
count(order_id) ord_count_per_hour,
extract(hour from order_purchase_timestamp) ord_time
from `target_dataset.orders`
group by ord_time
order by ord_time)

select
ord_count_per_hour,ord_time,
case
when ord_time between 0 and 6 then 'Dawn'
when ord_time between 7 and 12 then 'Morning'
when ord_time between 13 and 18 then 'Afternoon'
when ord_time between 19 and 23 then 'Night'
end as hour_of_day
from cte
```



5	206	4	Dawn
6	188	5	Dawn
7	502	6	Dawn
8	1231	7	Morning
9	2967	8	Morning
10	4785	9	Morning
11	6177	10	Morning
12	6578	11	Morning
13	5995	12	Morning
14	6518	13	Afternoon

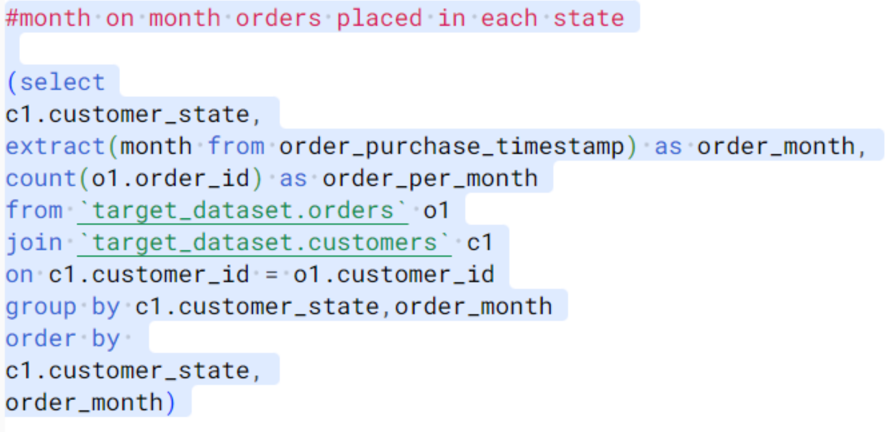


Insight from data:

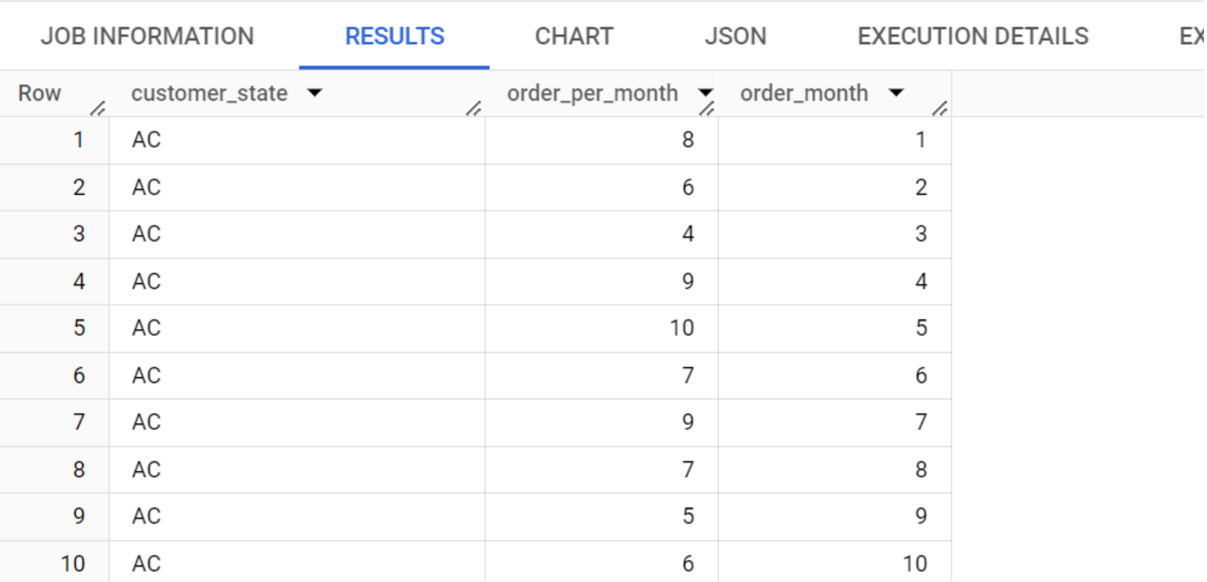
From the chart we can see that most number of orders were placed after 10:00 am to 10:00pm .

The morning hours are not very busy and we can see an increase in the number of orders after 10 that is usually the time when the business open and customers start to come in.

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



```
#month on month orders placed in each state

(select
c1.customer_state,
extract(month from order_purchase_timestamp) as order_month,
count(o1.order_id) as order_per_month
from `target_dataset.orders` o1
join `target_dataset.customers` c1
on c1.customer_id = o1.customer_id
group by c1.customer_state,order_month
order by
c1.customer_state,
order_month)
```

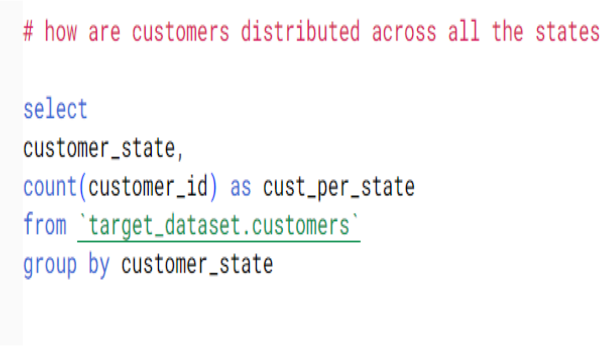


Row	customer_state	order_per_month	order_month
1	AC	8	1
2	AC	6	2
3	AC	4	3
4	AC	9	4
5	AC	10	5
6	AC	7	6
7	AC	9	7
8	AC	7	8
9	AC	5	9
10	AC	6	10

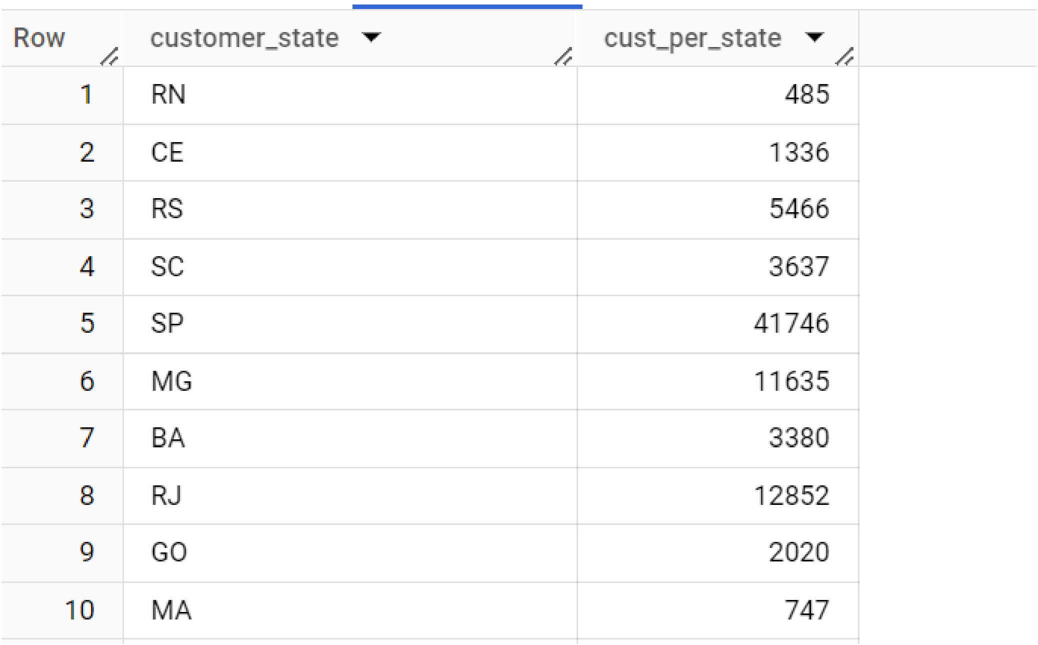
Insight from data:

The query shows how the orders are distributed on a month on month basis in each state.

8. How are the customers distributed across all the states?



```
# how are customers distributed across all the states

select
customer_state,
count(customer_id) as cust_per_state
from `target_dataset.customers`
group by customer_state
```



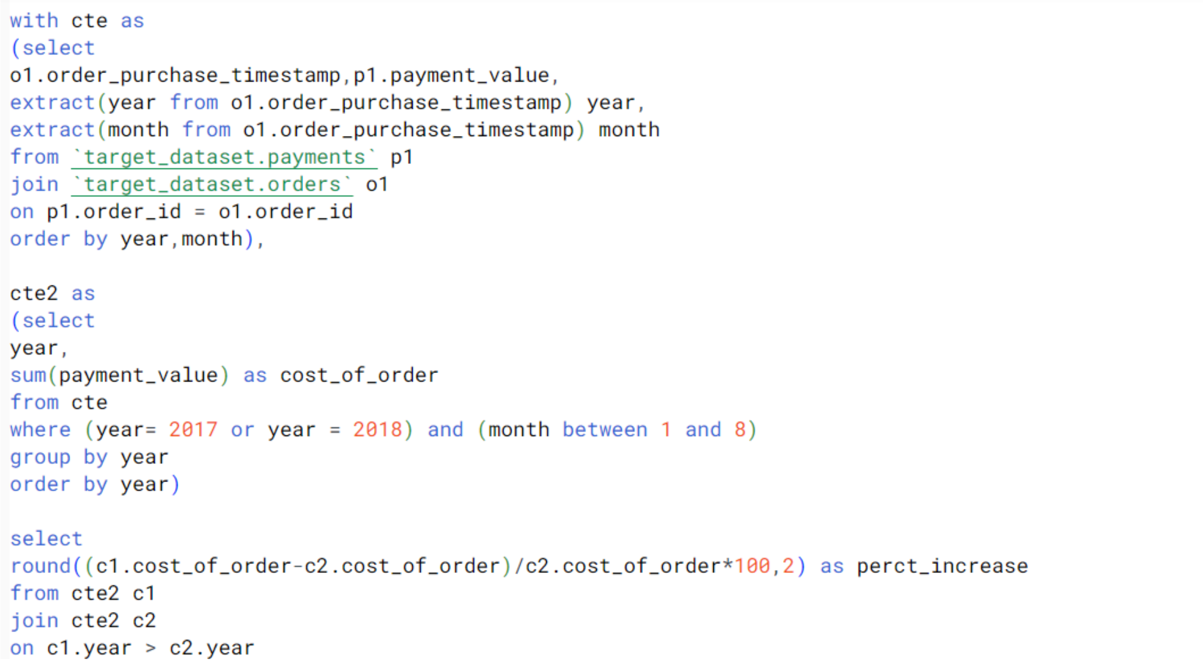
Row	customer_state	cust_per_state
1	RN	485
2	CE	1336
3	RS	5466
4	SC	3637
5	SP	41746
6	MG	11635
7	BA	3380
8	RJ	12852
9	GO	2020
10	MA	747

Insight from data:

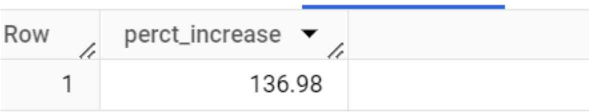
States SP,MG,RJ are the states where most people prefer target to shop. And RN and MA are the states where target is not so popular among the customers , being an analyst at target I would recommend that target should increase its marketing activities in these 2 states which have least number of customers.

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

You can use the "payment_value" column in the payments table to get the cost of orders.



```
with cte as
(select
o1.order_purchase_timestamp,p1.payment_value,
extract(year from o1.order_purchase_timestamp) year,
extract(month from o1.order_purchase_timestamp) month
from `target_dataset.payments` p1
join `target_dataset.orders` o1
on p1.order_id = o1.order_id
order by year,month),

cte2 as
(select
year,
sum(payment_value) as cost_of_order
from cte
where (year= 2017 or year = 2018) and (month between 1 and 8)
group by year
order by year)

select
round((c1.cost_of_order-c2.cost_of_order)/c2.cost_of_order*100,2) as perct_increase
from cte2 c1
join cte2 c2
on c1.year > c2.year
```

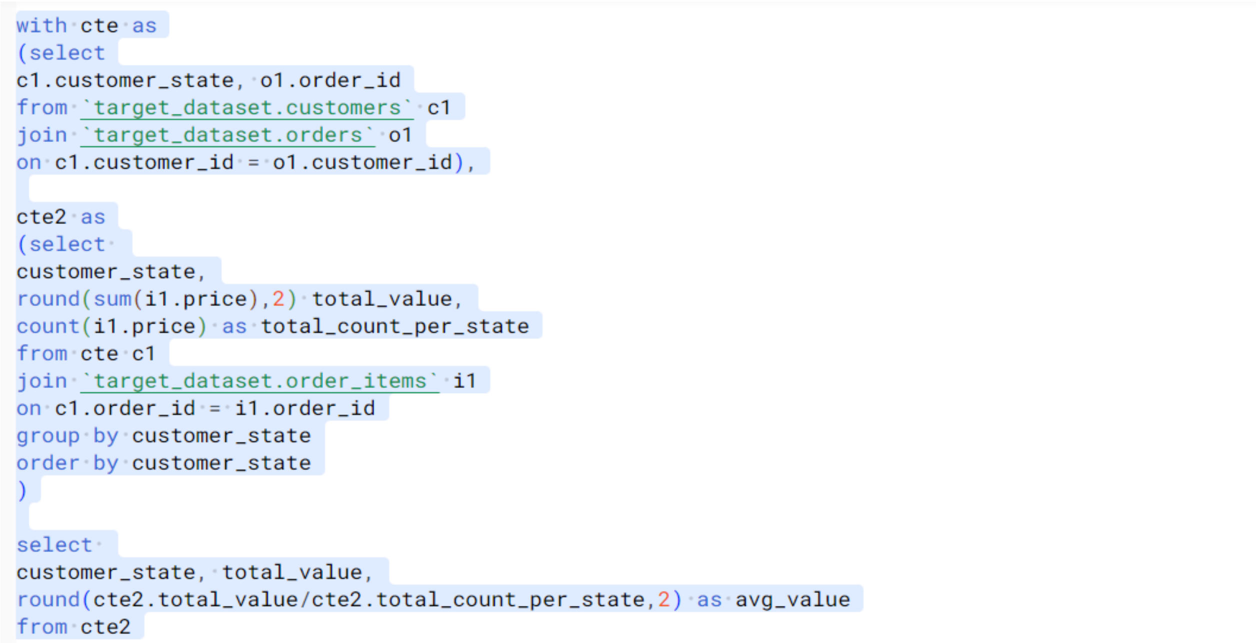


Row	perct_increase
1	136.98

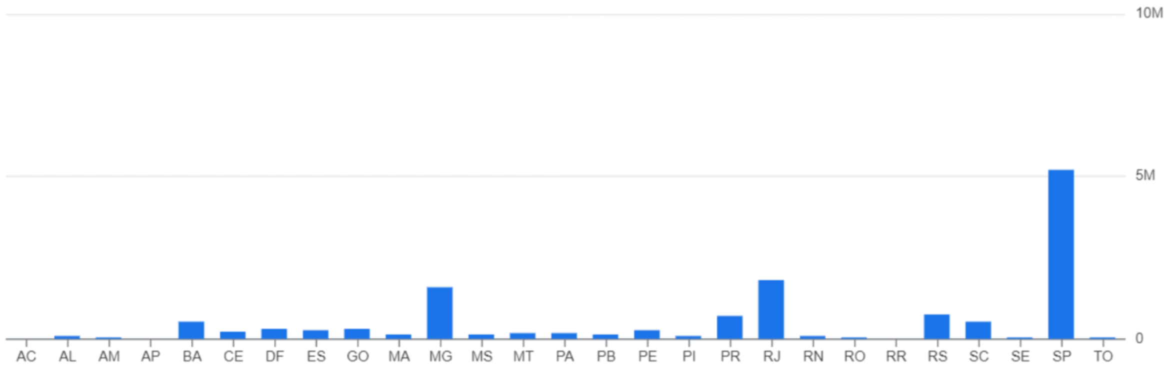
Insight from data

There has been an increase of 137% in the cost of orders from the year 2017 to 2018.

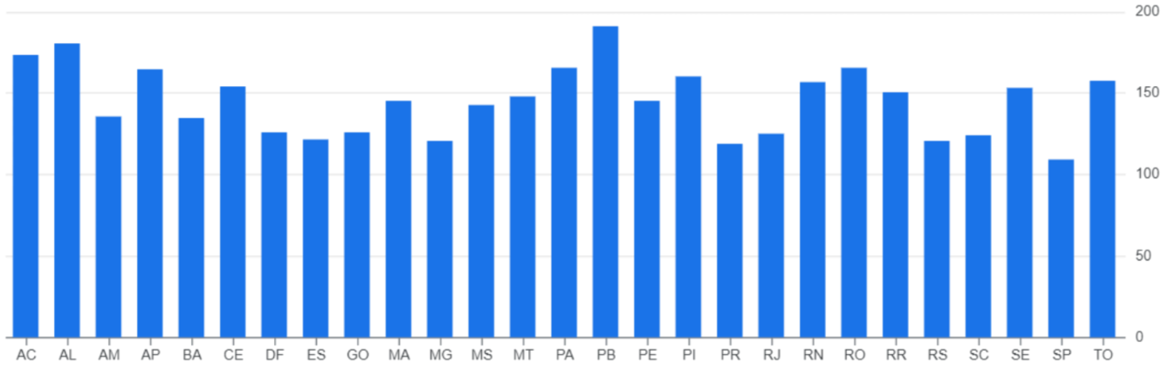
10. Calculate the Total & Average value of order price for each state.



```
with cte as
(select
c1.customer_state, o1.order_id
from `target_dataset.customers` c1
join `target_dataset.orders` o1
on c1.customer_id = o1.customer_id),

cte2 as
(select
customer_state,
round(sum(i1.price),2) total_value,
count(i1.price) as total_count_per_state
from cte c1
join `target_dataset.order_items` i1
on c1.order_id = i1.order_id
group by customer_state
order by customer_state
)

select
customer_state, total_value,
round(cte2.total_value/cte2.total_count_per_state,2) as avg_value
from cte2
```



total_value by customer_state

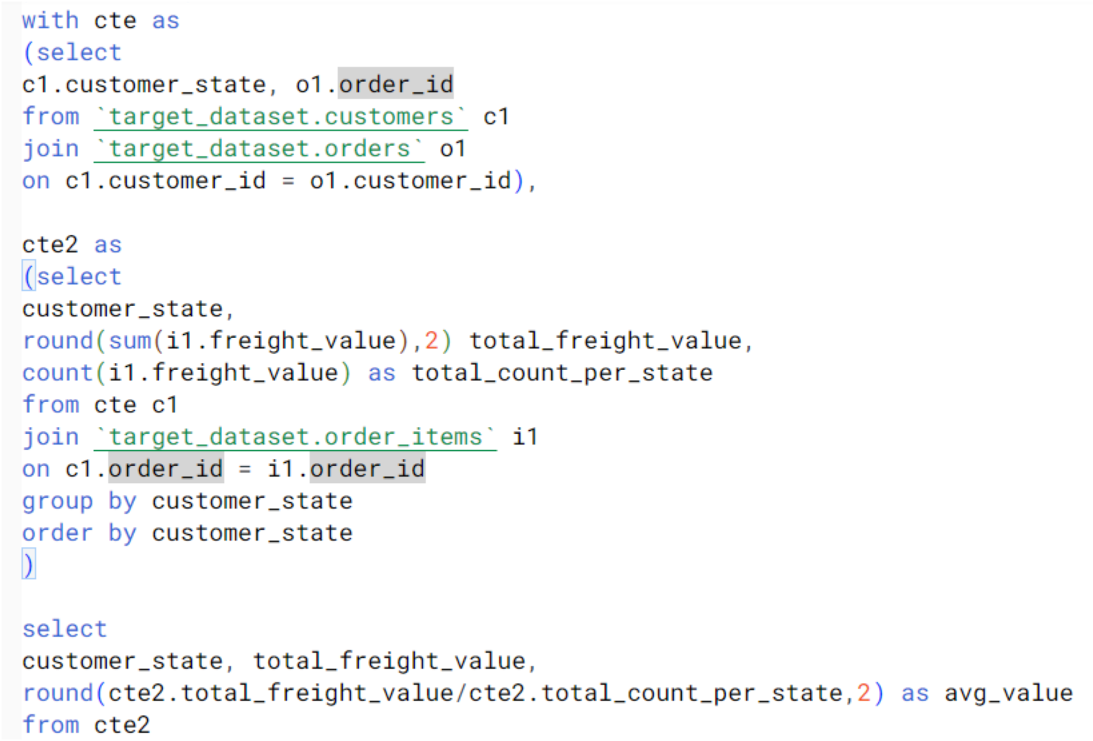


avg_value by customer_state

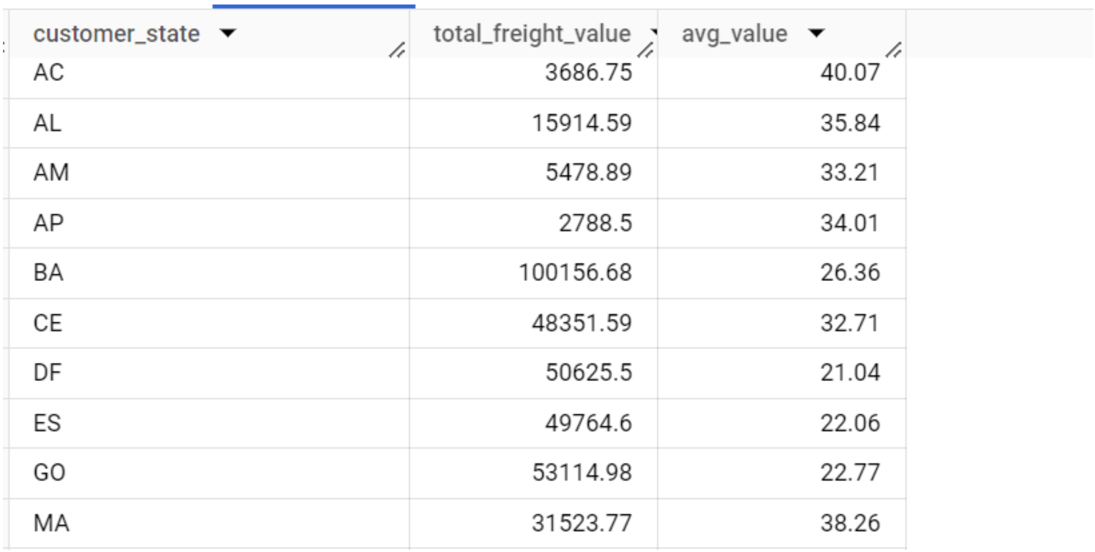
Insight from data:

SP is the state that has the highest value of orders that has been received, and from the previous analysis also we know that SP has the highest number of customers as well. Total value of the orders from SP are around 5million which is way greater than the average value of orders that is around 100 thousand to 150 thousand dollars. This figure is more clear from the avg_value by customer chart .

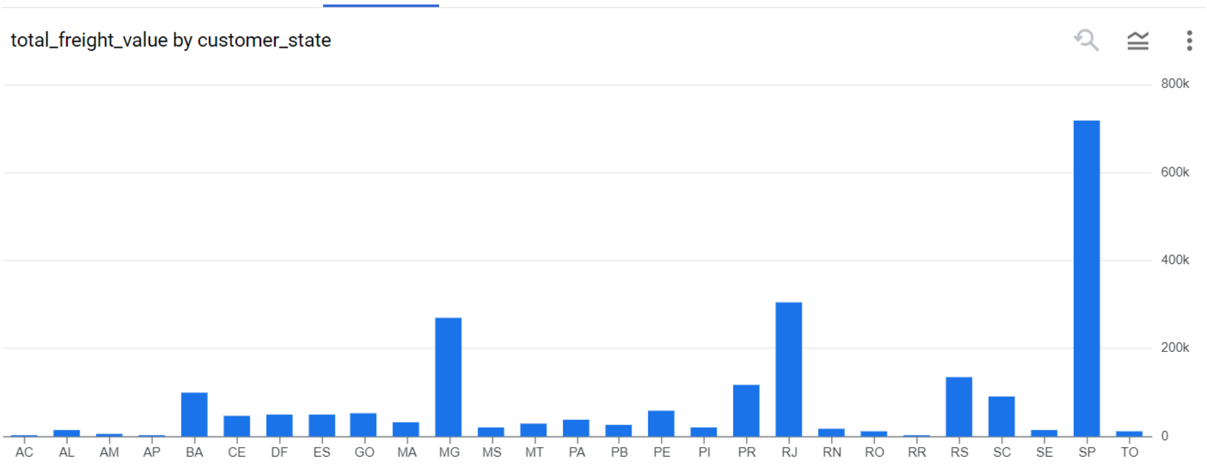
11. Calculate the Total & Average value of order freight for each state.

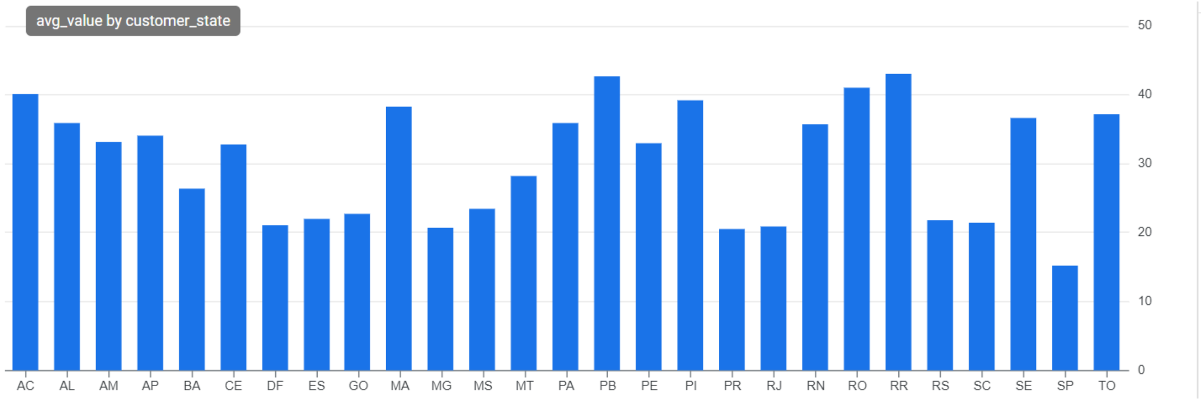


```
with cte as
(select
c1.customer_state, o1.order_id
from `target_dataset.customers` c1
join `target_dataset.orders` o1
on c1.customer_id = o1.customer_id),

cte2 as
(select
customer_state,
round(sum(i1.freight_value),2) total_freight_value,
count(i1.freight_value) as total_count_per_state
from cte c1
join `target_dataset.order_items` i1
on c1.order_id = i1.order_id
group by customer_state
order by customer_state
)

select
customer_state, total_freight_value,
round(cte2.total_freight_value/cte2.total_count_per_state,2) as avg_value
from cte2
```



customer_state	total_freight_value	avg_value
AC	3686.75	40.07
AL	15914.59	35.84
AM	5478.89	33.21
AP	2788.5	34.01
BA	100156.68	26.36
CE	48351.59	32.71
DF	50625.5	21.04
ES	49764.6	22.06
GO	53114.98	22.77
MA	31523.77	38.26

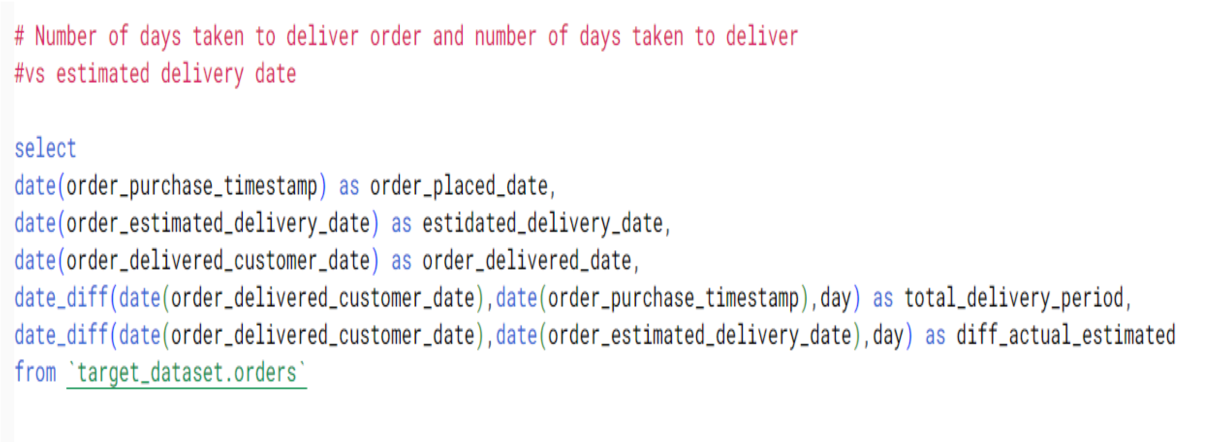




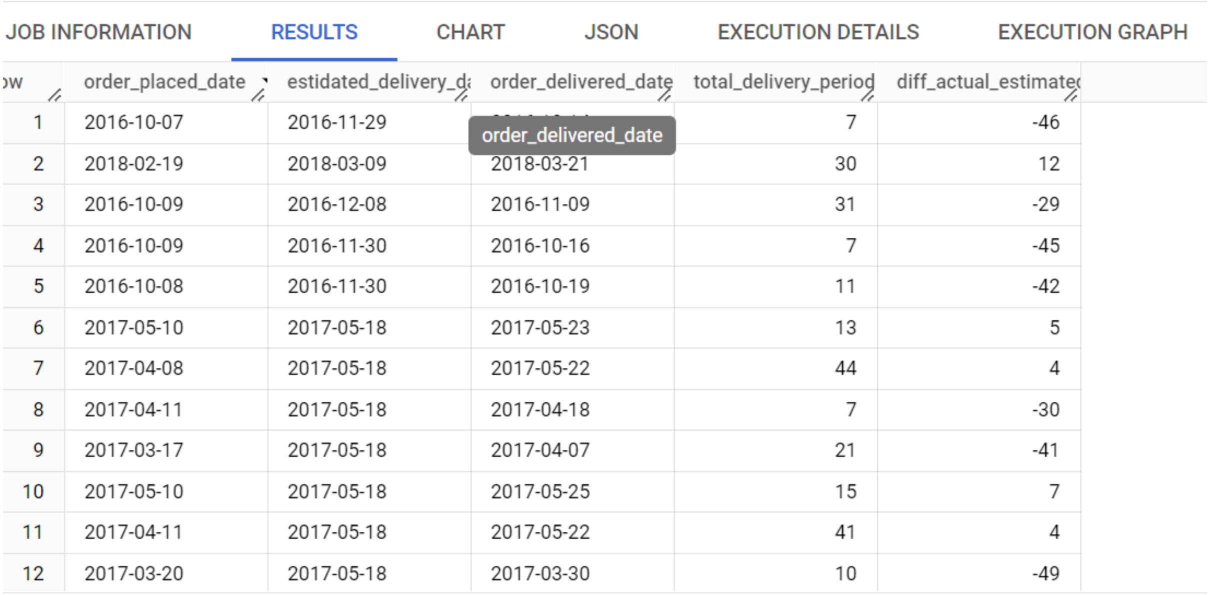
Insight from data:

States with more number of orders have low average freight value. So in order to reduce the freight cost company should either increase the order value per customer or open up some new warehouses near the states which have high average freight value , otherwise the profitability may decline in those states.

12.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.



```
# Number of days taken to deliver order and number of days taken to deliver
#vs estimated delivery date

select
date(order_purchase_timestamp) as order_placed_date,
date(order_estimated_delivery_date) as estidated_delivery_date,
date(order_delivered_customer_date) as order_delivered_date,
date_diff(date(order_delivered_customer_date),date(order_purchase_timestamp),day) as total_delivery_period,
date_diff(date(order_delivered_customer_date),date(order_estimated_delivery_date),day) as diff_actual_estimated
from `target_dataset.orders`
```

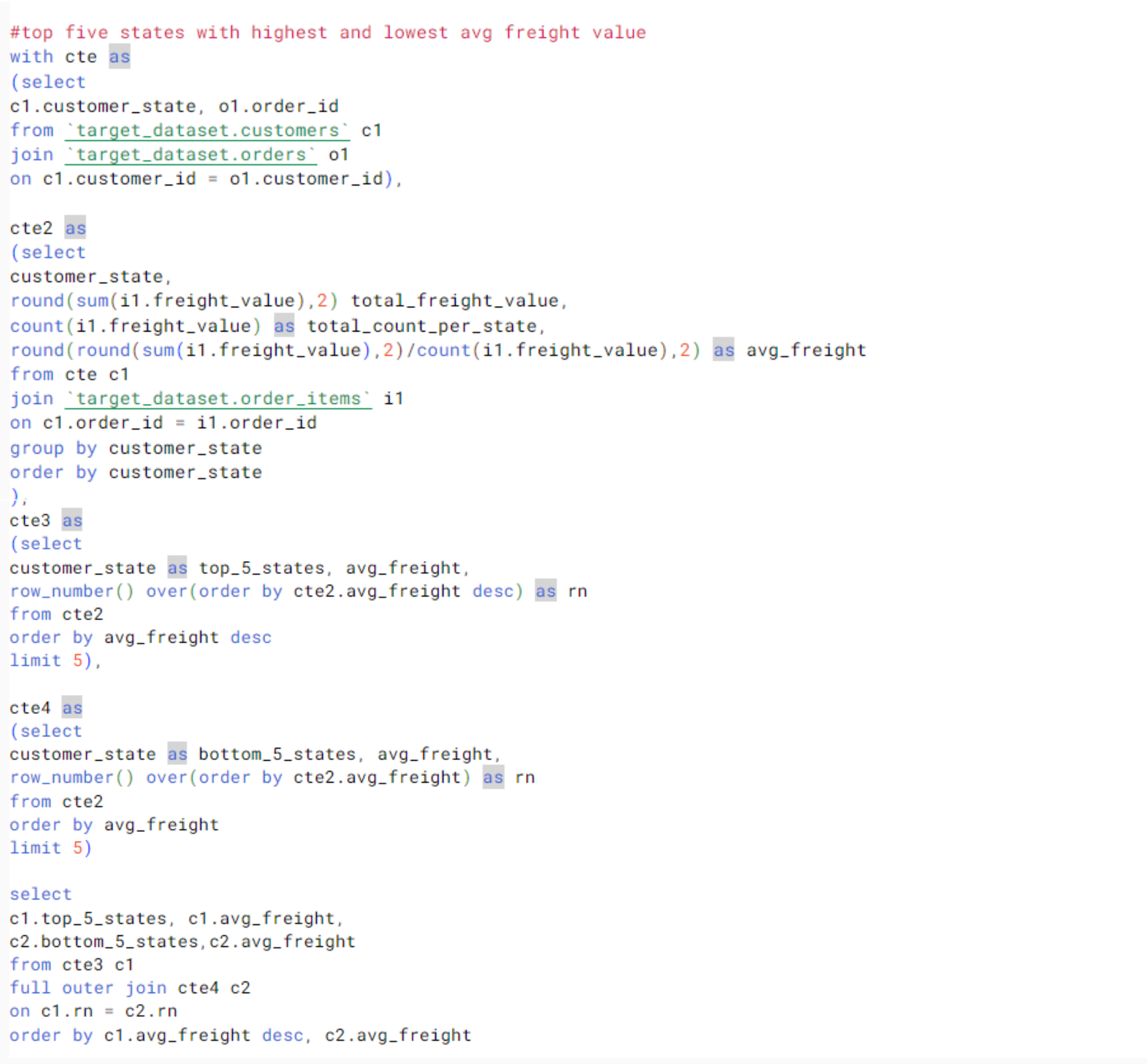


ow	order_placed_date	estidated_delivery_da	order_delivered_date	total_delivery_period	diff_actual_estimatec
1	2016-10-07	2016-11-29	[illegible]	7	-46
2	2018-02-19	2018-03-09	2018-03-21	30	12
3	2016-10-09	2016-12-08	2016-11-09	31	-29
4	2016-10-09	2016-11-30	2016-10-16	7	-45
5	2016-10-08	2016-11-30	2016-10-19	11	-42
6	2017-05-10	2017-05-18	2017-05-23	13	5
7	2017-04-08	2017-05-18	2017-05-22	44	4
8	2017-04-11	2017-05-18	2017-04-18	7	-30
9	2017-03-17	2017-05-18	2017-04-07	21	-41
10	2017-05-10	2017-05-18	2017-05-25	15	7
11	2017-04-11	2017-05-18	2017-05-22	41	4
12	2017-03-20	2017-05-18	2017-03-30	10	-49

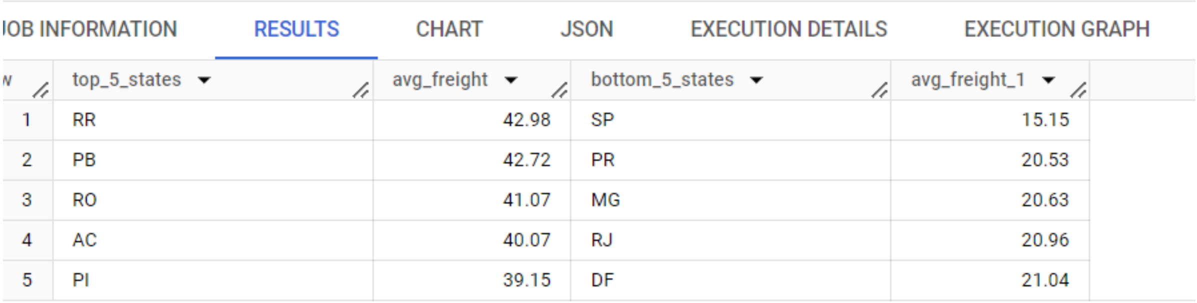
Insight from data:

Total delivery period shows how much time it took from the order placed date to actually reach the buyer, and the next column in result table shows whether the actual delivery quicker or took more time than what was the tentative delivery date that the buyer was given at the time of making the order. Negative values shows the delivery was quick and positive value shows that delivery was late.

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



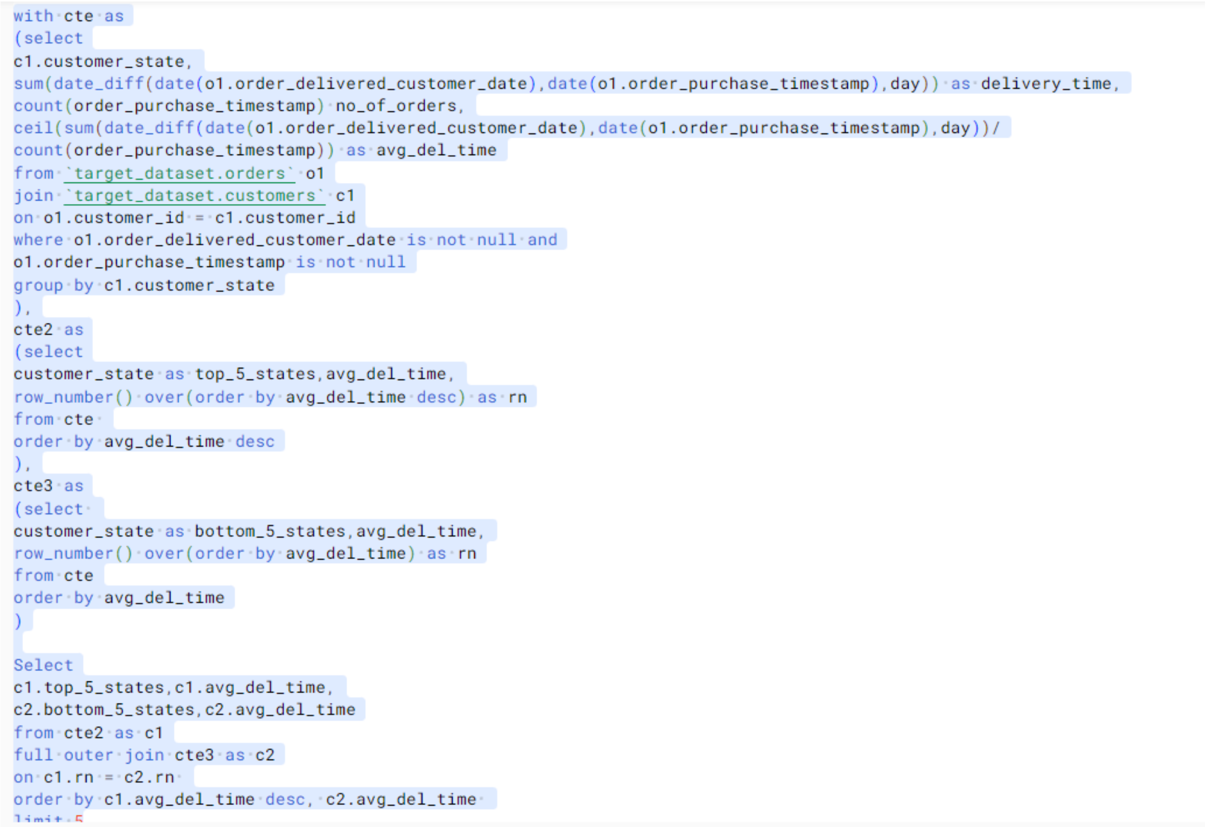
```
#top five states with highest and lowest avg freight value
with cte as
(select
c1.customer_state, o1.order_id
from `target_dataset.customers` c1
join `target_dataset.orders` o1
on c1.customer_id = o1.customer_id),

cte2 as
(select
customer_state,
round(sum(i1.freight_value),2) total_freight_value,
count(i1.freight_value) as total_count_per_state,
round(round(sum(i1.freight_value),2)/count(i1.freight_value),2) as avg_freight
from cte c1
join `target_dataset.order_items` i1
on c1.order_id = i1.order_id
group by customer_state
order by customer_state
),
cte3 as
(select
customer_state as top_5_states, avg_freight,
row_number() over(order by cte2.avg_freight desc) as rn
from cte2
order by avg_freight desc
limit 5),

cte4 as
(select
customer_state as bottom_5_states, avg_freight,
row_number() over(order by cte2.avg_freight) as rn
from cte2
order by avg_freight
limit 5)

select
c1.top_5_states, c1.avg_freight,
c2.bottom_5_states,c2.avg_freight
from cte3 c1
full outer join cte4 c2
on c1.rn = c2.rn
order by c1.avg_freight desc, c2.avg_freight
```



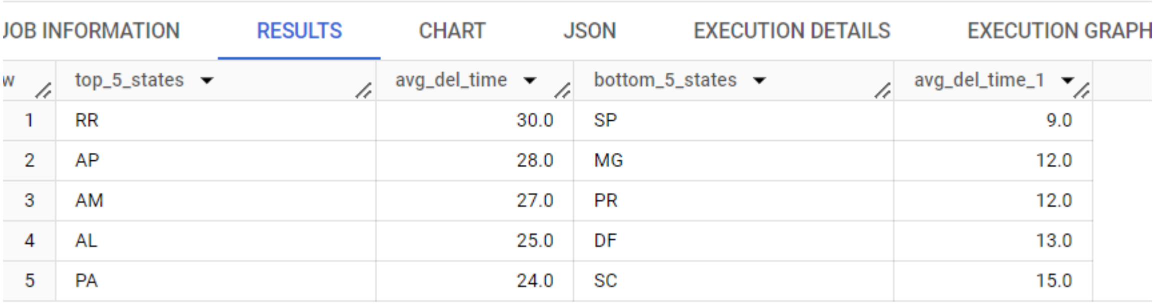
IOB INFORMATION | RESULTS | CHART | JSON | EXECUTION DETAILS | EXECUTION GRAPH

w	top_5_states	avg_freight	bottom_5_states	avg_freight_1
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

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

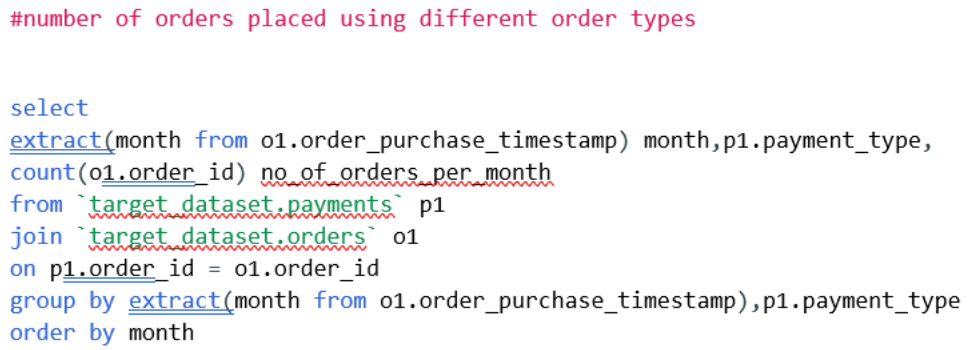


```
with cte as
(select
c1.customer_state,
sum(date_diff(date(o1.order_delivered_customer_date),date(o1.order_purchase_timestamp),day)) as delivery_time,
count(order_purchase_timestamp) no_of_orders,
ceil(sum(date_diff(date(o1.order_delivered_customer_date),date(o1.order_purchase_timestamp),day))/
count(order_purchase_timestamp)) as avg_del_time
from `target_dataset.orders` o1
join `target_dataset.customers` c1
on o1.customer_id = c1.customer_id
where o1.order_delivered_customer_date is not null and
o1.order_purchase_timestamp is not null
group by c1.customer_state
),
cte2 as
(select
customer_state as top_5_states,avg_del_time,
row_number() over(order by avg_del_time desc) as rn
from cte
order by avg_del_time desc
),
cte3 as
(select
customer_state as bottom_5_states,avg_del_time,
row_number() over(order by avg_del_time) as rn
from cte
order by avg_del_time
)

Select
c1.top_5_states,c1.avg_del_time,
c2.bottom_5_states,c2.avg_del_time
from cte2 as c1
full outer join cte3 as c2
on c1.rn = c2.rn
order by c1.avg_del_time desc, c2.avg_del_time
limit 5
```

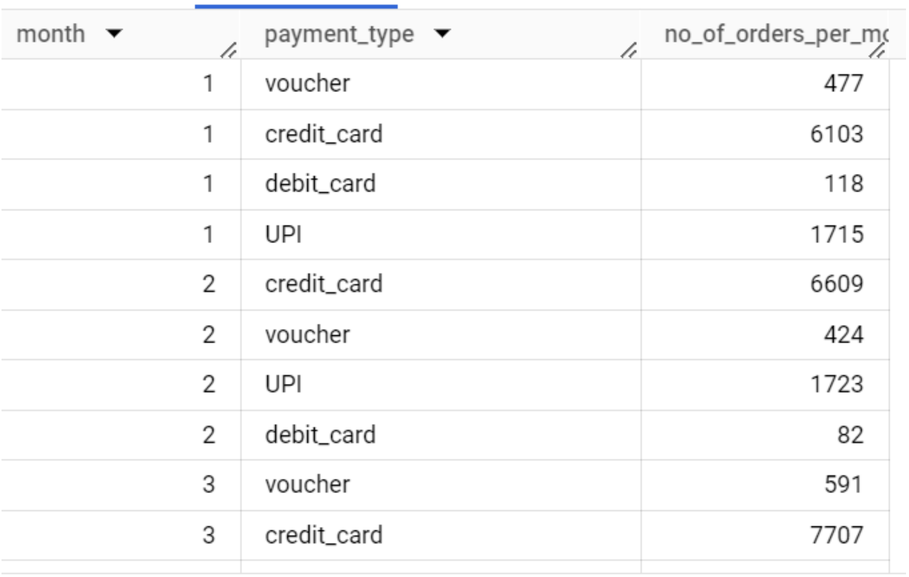


w	top_5_states	avg_del_time	bottom_5_states	avg_del_time_1
1	RR	30.0	SP	9.0
2	AP	28.0	MG	12.0
3	AM	27.0	PR	12.0
4	AL	25.0	DF	13.0
5	PA	24.0	SC	15.0

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



```
#number of orders placed using different order types


select
extract(month from o1.order_purchase_timestamp) month,p1.payment_type,
count(o1.order_id) no_of_orders_per_month
from `target_dataset.payments` p1
join `target_dataset.orders` o1
on p1.order_id = o1.order_id
group by extract(month from o1.order_purchase_timestamp),p1.payment_type
order by month
```

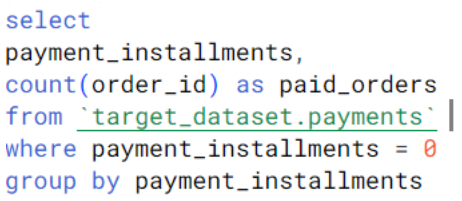


month	payment_type	no_of_orders_per_mo
1	voucher	477
1	credit_card	6103
1	debit_card	118
1	UPI	1715
2	credit_card	6609
2	voucher	424
2	UPI	1723
2	debit_card	82
3	voucher	591
3	credit_card	7707

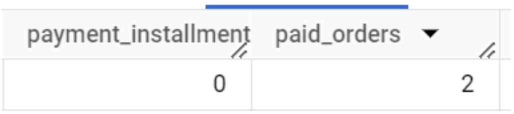
Insight from data:

There are majorly 4 payment types being used for placing orders i.e. vouchers, credit_card , debit_card and UPI. Result shows how the number of orders are differentiated based on the payment type.

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



```
select
payment_installments,
count(order_id) as paid_orders
from `target_dataset.payments`
where payment_installments = 0
group by payment_installments
```

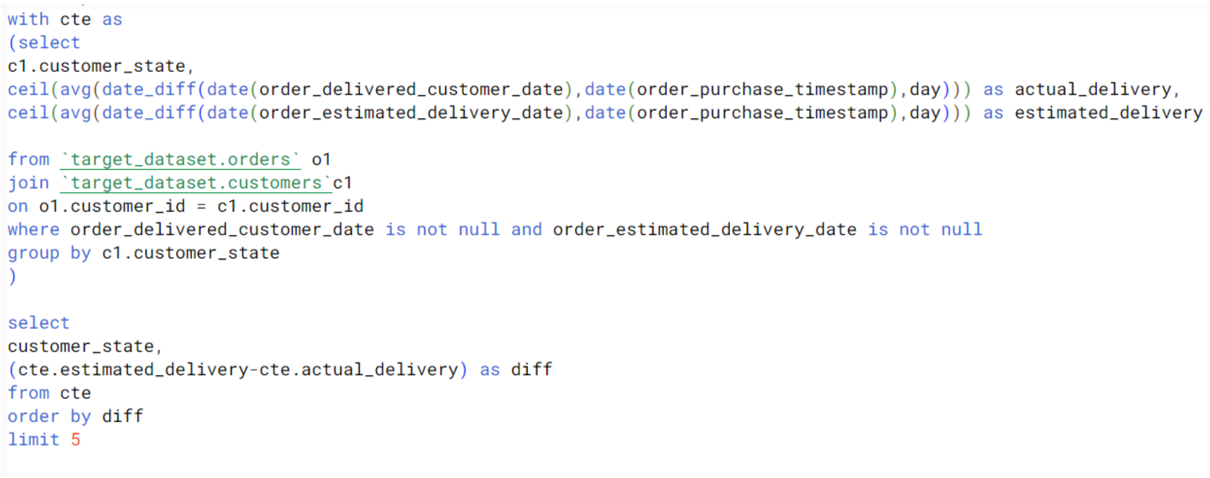


payment_installment	paid_orders
0	2

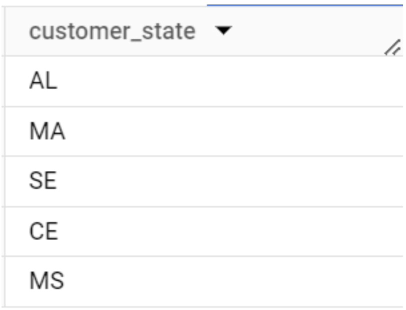
Insight from data:

There are only 2 orders that have been paid.

17. Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.



```
with cte as
(select
c1.customer_state,
ceil(avg(date_diff(date(order_delivered_customer_date),date(order_purchase_timestamp),day))) as actual_delivery,
ceil(avg(date_diff(date(order_estimated_delivery_date),date(order_purchase_timestamp),day))) as estimated_delivery

from `target_dataset.orders` o1
join `target_dataset.customers`c1
on o1.customer_id = c1.customer_id
where order_delivered_customer_date is not null and order_estimated_delivery_date is not null
group by c1.customer_state
)

select
customer_state,
(cte.estimated_delivery-cte.actual_delivery) as diff
from cte
order by diff
limit 5
```



customer_state
AL
MA
SE
CE
MS

Insight from data:

As per the average delivery time and average estimated delivery period these are the top five states with fastest delivery serive.