

TARGET_SQL_PROJECT

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

A. Data type of all columns in the “customers” table.

ANSWER 1 A :

```
SELECT column_name,  
       data_type  
FROM `secure-racer-402417.target_sql.INFORMATION_SCHEMA.COLUMNS`  
where table_name = 'customers';
```

Query results

JOB INFORMATION		RESULTS	CHART	PREVIEW
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 : four columns are of string datatype and one column is of int datatype.

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

ANSWER 1 B

```
Select min(order_purchase_timestamp)as first_order,  
        max(order_purchase_timestamp) as last_order  
from `secure-racer-402417.target_sql.orders`;
```

Query results

JOB INFORMATION		RESULTS	CHART	PREVIEW	J
Row	first_order ▼	last_order ▼			
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC			

insights :1st order placed in 2016-09-04 21:15:19 UTC
Last order placed in 2018-10-17 17:30:18 UTC

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

ANSWER 1 C

```
select count(distinct customer_city )as count_city,  
       count(distinct customer_state)as count_state  
from `secure-racer-402417.target_sql.orders` o  
inner join `secure-racer-402417.target_sql.customers` c  
on o.customer_id = c.customer_id ;
```

Query results

JOB INFORMATION		RESULTS	CHART
Row	count_city ▼	count_state ▼	
1	4119	27	

Insights : count_city = 4119

count_state = 27

(2) In-depth Exploration:

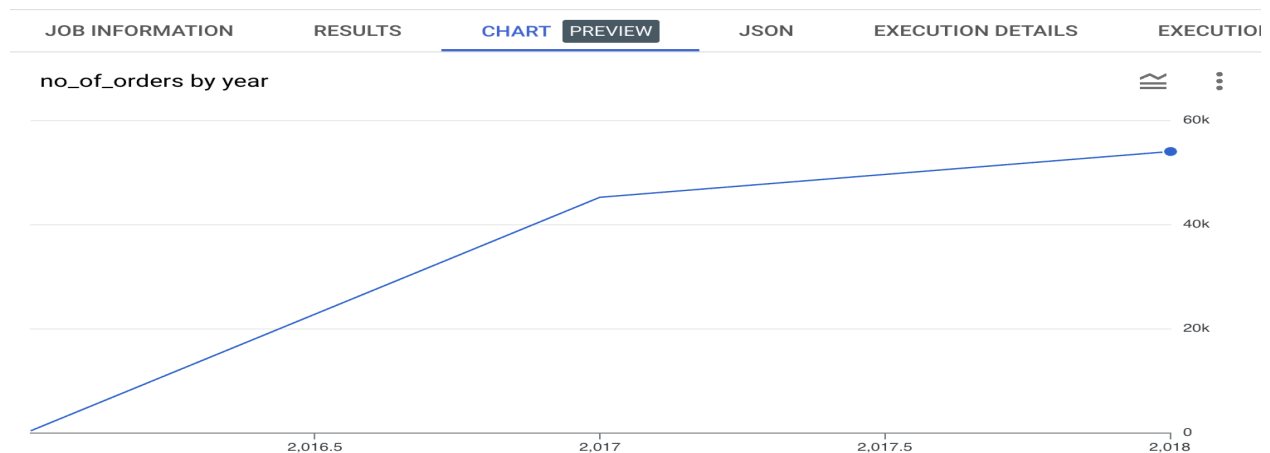
A. Is there a growing trend in the no. of orders placed over the past years?

ANSWER 2 A

```
select extract(year from order_purchase_timestamp)as year,
       count(*)as no_of_orders
from `secure-racer-402417.target_sql.orders`
group by 1
order by year ;
```

Query results

JOB INFORMATION		RESULTS		CHART
Row	year ▼		no_of_orders ▼	
1	2016		329	
2	2017		45101	
3	2018		54011	



Insights : over the last three years number of orders has been increasing.

Percentage_increase b/w 2016-2017 = 13,608.5% increase

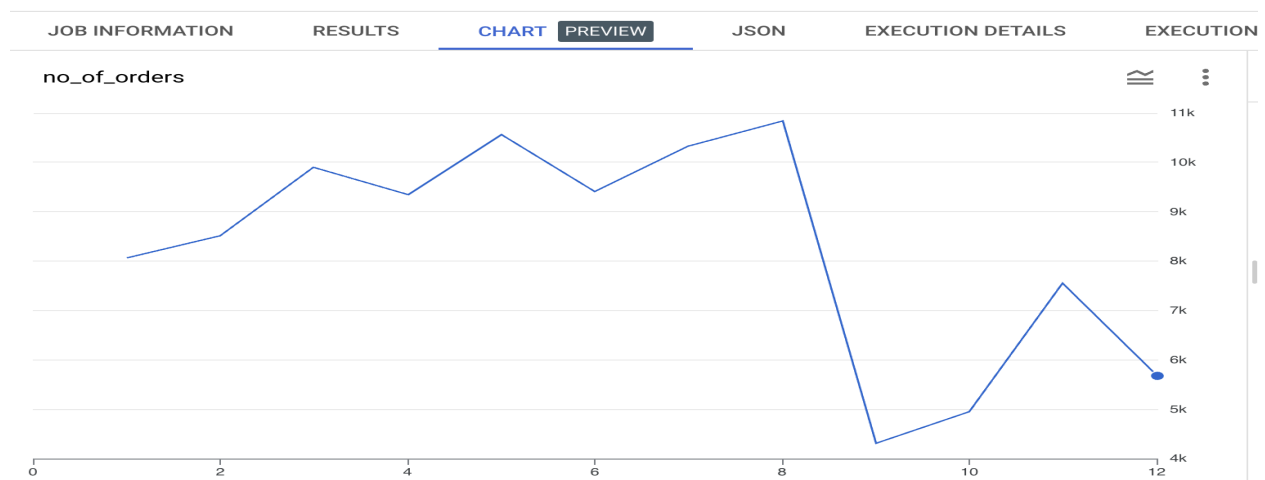
And b/w 2017-2018 = 19.7557% increase

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

ANSWER 2B

```
Select extract(month from order_purchase_timestamp) month,
       count(*) as no_of_orders
from `secure-racer-402417.target_sql.orders`
group by 1
order by 1
```

JOB INFORMATION		RESULTS		CHART	PREVIEW
Row	month		no_of_orders		
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: here the number of orders is greater than 10k for the month of 5,7,8 and after August there is a dip in the number of orders.

2(B)

2ND APPROACH MONTHLY SEASONALITY FOR INDIVIDUAL YEAR

```
Select extract(year from order_purchase_timestamp) year,  
extract(month from order_purchase_timestamp) month,  
count(*) as no_of_orders  
from `secure-racer-402417.target_sql.orders`  
group by 1,2  
order by 1,2
```

JOB INFORMATION		RESULTS	CHART	PREVIEW	JSOI
Row	year ▼	month ▼	no_of_orders ▼		
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		

2 C. 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

ANSWER 2 C

```
select
    case when extract(hour from o.order_purchase_timestamp ) between 0 and 6 then
        'Dawn'
        when extract(hour from o.order_purchase_timestamp ) between 7 and 12 then
        'Mornings'
        when extract(hour from o.order_purchase_timestamp ) between 13 and 18 then
        'Afternoon'
        when extract(hour from o.order_purchase_timestamp ) between 19 and 23 then
        'Night' end as time_of_the_day,
    count(*) as cnt
from `secure-racer-402417.target_sql.orders` o
inner join `secure-racer-402417.target_sql.customers` c
on c.customer_id=o.customer_id
where c.customer_state = 'BA'
group by 1
order by cnt;
```

Query results

JOB INFORMATION		RESULTS	CHART	PREVIEW
Row	time_of_the_day	cnt		
1	Dawn	207		
2	Mornings	895		
3	Night	1006		
4	Afternoon	1272		

Query results

[SAVI](#)

JOB INFORMATION

RESULTS

CHART

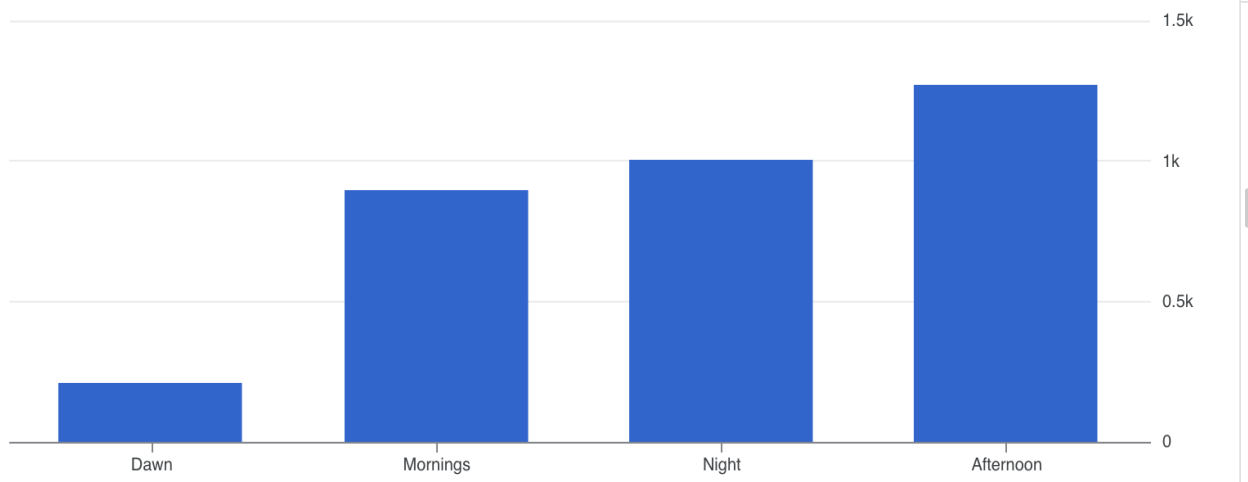
PREVIEW

JSON

EXECUTION DETAILS

EXECUTION

cnt by time_of_the_day



Insights: Brazilian customers mostly placed their orders in the afternoon and night .

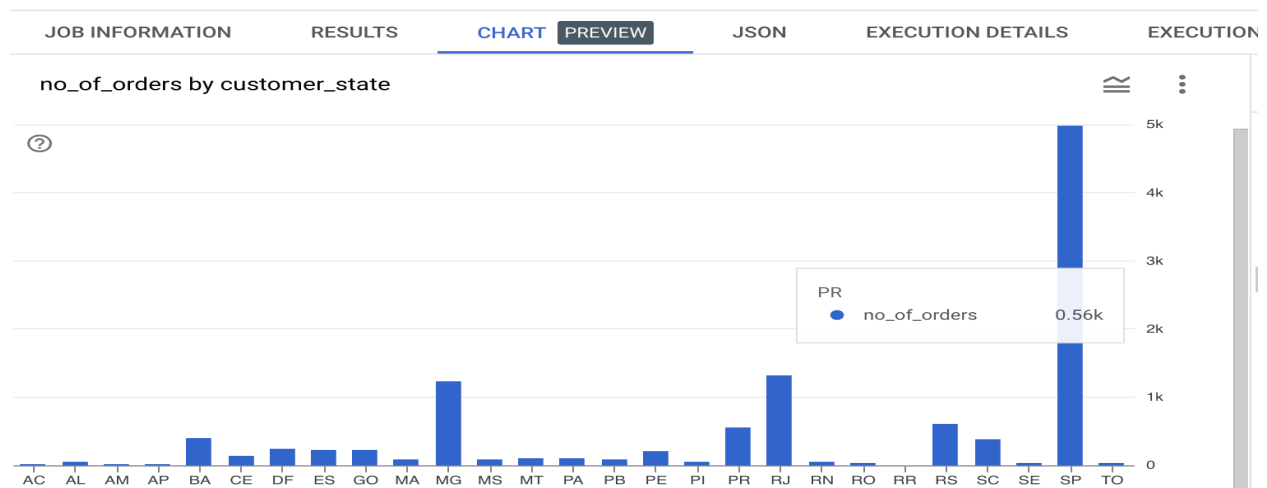
3. Evolution of E-commerce orders in the Brazil region:

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

ANSWER 3A

```
select c.customer_state,
       extract(month from o.order_purchase_timestamp ) as month,
       count(o.order_id) as no_of_orders
from `secure-racer-402417.target_sql.customers` c
     inner join `secure-racer-402417.target_sql.orders` o
     on c.customer_id=o.customer_id
group by 1,2
order by 1,2 ;
```

JOB INFORMATION		RESULTS	CHART	PREVIEW	JSON	E
Row	customer_state	month	no_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			
12	AC	12	5			



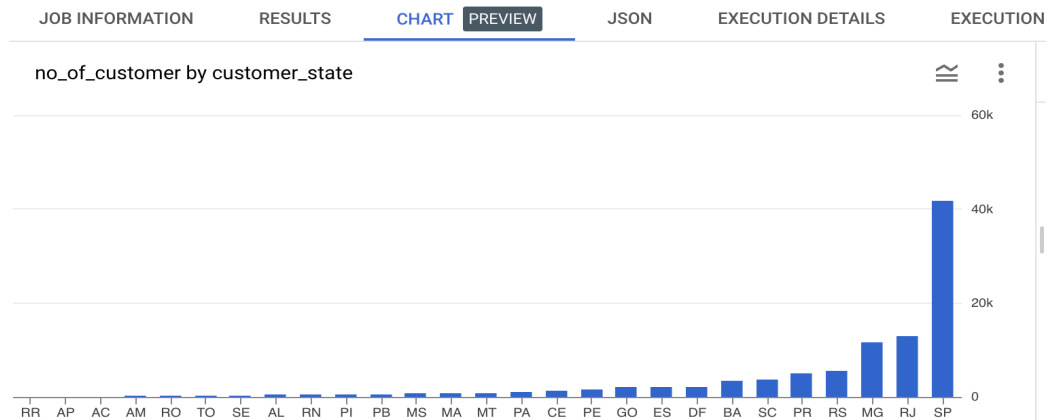
Insights: The number of orders every month is highest in SP state.

3B. How are the customers distributed across all the states?

ANSWER B ;

```
select c.customer_state,
       count(distinct c.customer_id) as no_of_customer
from `secure-racer-402417.target_sql.customers` c
group by 1
order by 1;
```

Query results			
JOB INFORMATION		RESULTS	CHART PREVIEW
Row	customer_state	no_of_customer	
1	RR	46	
2	AP	68	
3	AC	81	
4	AM	148	
5	RO	253	
6	TO	280	
7	SE	350	
8	AL	413	
9	RN	485	
10	PI	495	



Insights:for state SP no_of_customer is 41746. which is 'highest' and for state RR no_of_customer is 46.which is 'lowest'.

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

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

ANSWER (A):

```
with final AS
(
select distinct extract(year from o.order_purchase_timestamp) as year,
sum(p.payment_value) over (partition by extract(year from
o.order_purchase_timestamp) order by extract(year from
o.order_purchase_timestamp)) as total_payment
from `secure-racer-402417.target_sql.orders` o
inner join `secure-racer-402417.target_sql.payments` p
on o.order_id = p.order_id
where extract(month from o.order_purchase_timestamp) between 1 and 8 and
extract(year from o.order_purchase_timestamp) between 2017 and 2018
ORDER BY year
)
Select l.year as start_yr,
r.year as end_yr,
l.total_payment as start_total_payment,
r.total_payment as end_total_payment,
((r.total_payment - l.total_payment) / l.total_payment) * 100 as
percentage_increase
from final l
inner join final r
on l.year < r.year ;
```

Query results

JOB INFORMATION		RESULTS	CHART	PREVIEW	JSON	EXECUTION DETAILS
Row	start_yr	end_yr	start_total_payment	end_total_payment	percentage_increase	
1	2017	2018	3669022.12	8694733.84	136.9768716466...	

Insights: % increase in the cost of orders from year 2017 to 2018 is 136.977 %

Second approach =>>> with lead()function

```
with final AS
(
select distinct extract(year from o.order_purchase_timestamp) as year,
       sum(p.payment_value) over (partition by extract(year from
o.order_purchase_timestamp) order by extract(year from
       o.order_purchase_timestamp)) as total_payment
from `secure-racer-402417.target_sql.orders` o
inner join `secure-racer-402417.target_sql.payments` p
       on o.order_id = p.order_id
where extract(month from o.order_purchase_timestamp) between 1 and 8 and
       extract(year from o.order_purchase_timestamp) between 2017 and 2018
ORDER BY year

)
select *,
       lead(total_payment)over(order by year) as lead_total_payment,
       ((lead(total_payment)over(order by year) - total_payment)/total_payment)*100 as
percentage_increase
from final
order by final.year ;
```

JOB INFORMATION		RESULTS	CHART	PREVIEW	JSON	EXECUTED
Row	year ▼	total_payment ▼	lead_total_payment	percentages_increase		
1	2017	3669022.12	8694733.84	136.9768716466...		
2	2018	8694733.84	null	null		

Insights: % increase in the cost of orders from year 2017 to 2018 is = 136.977 %

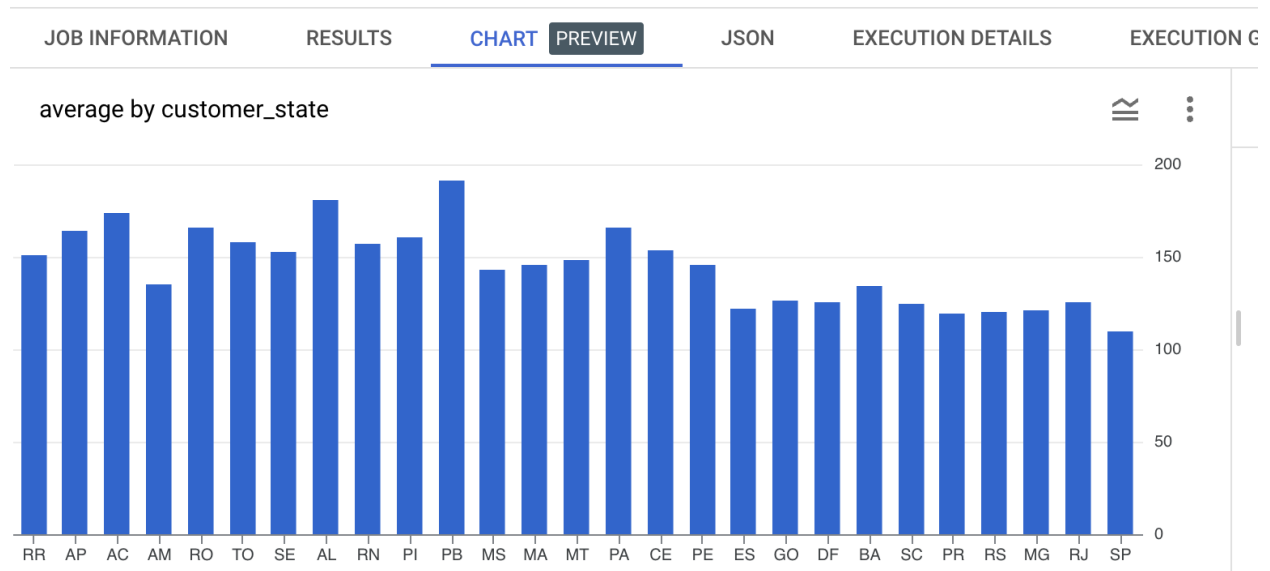
4 B . Calculate the Total & Average value of order price for each state.

ANSWER ;

```
select distinct A.customer_state,
               sum(A.price)over(partition by A.customer_state) as total_price,
               avg(A.price)over(partition by A.customer_state) as average_price
from(
  select c.customer_state , oi.price
  from `secure-racer-402417.target_sql.order_items` oi
    join `secure-racer-402417.target_sql.orders` o
    on oi.order_id = o.order_id
    join `secure-racer-402417.target_sql.customers` c
    on o.customer_id = c.customer_id) A
order by total_price ,average_price desc ;
```

Query results

JOB INFORMATION		RESULTS	CHART	PREVIEW	JSON	EXECUTE
Row	customer_state ▼	total_price ▼	average_price ▼			
1	RR	7829.43	150.5659615384...			
2	AP	13474.3	164.3207317073...			
3	AC	15982.95	173.7277173913...			
4	AM	22356.84	135.496			
5	RO	46140.64	165.9735251798...			
6	TO	49621.74	157.5293333333...			
7	SE	58920.85	153.0411688311...			
8	AL	80314.81	180.8892117117...			
9	RN	83034.98	156.9659357277...			
10	PI	86914.08	160.3580811808...			



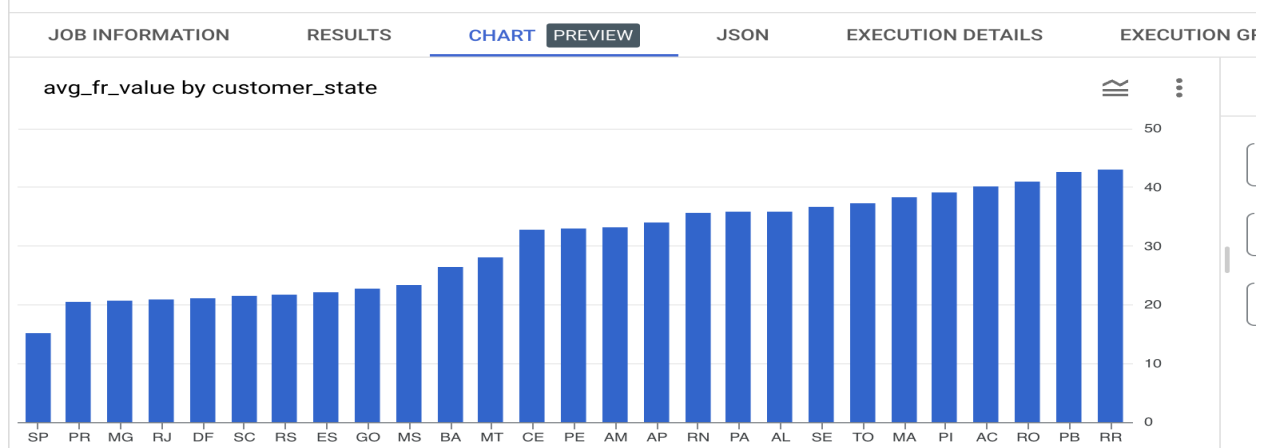
Insights: for customer_state SP average_price is lowest and for Customer_state PB average_price is highest.

4C. Calculate the Total & Average value of order freight for each state.

```
select distinct A.customer_state,
    sum(A.freight_value)over(partition by A.customer_state order by A.customer_state)
    as total_fr_value,
    avg(A.freight_value)over(partition by A.customer_state order by A.customer_state)
    as avg_fr_value
From (select c.customer_state , oi.freight_value
    from `secure-racer-402417.target_sql.order_items` oi
    join `secure-racer-402417.target_sql.orders` o
    on oi.order_id = o.order_id
    join `secure-racer-402417.target_sql.customers` c
    on o.customer_id = c.customer_id) A
Order by 3 ;
```

Query results

JOB INFORMATION		RESULTS	CHART	PREVIEW	JSON	EXEC
Row	customer_state	total_fr_value	avg_fr_value			
1	AC	3686.75	40.07336956521...			
2	AL	15914.59	35.84367117117...			
3	AM	5478.89	33.20539393939...			
4	AP	2788.5	34.00609756097...			
5	BA	100156.68	26.36395893656...			
6	CE	48351.59	32.71420162381...			
7	DF	50625.5	21.04135494596...			
8	ES	49764.6	22.05877659574...			
9	GO	53114.98	22.76681525932...			
10	MA	31523.77	38.25700242718...			



Insights: for state RR avg_fr_value is highest and for state SP avg_fr_value is lowest.

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

A. 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

ANSWER ;

```
select order_id,  
       date_diff(order_delivered_customer_date, order_purchase_timestamp, day)  
AS time_to_deliver,  
       date_diff(order_delivered_customer_date, order_estimated_delivery_date,  
day) AS diff_estimated_delivery  
FROM `secure-racer-402417.target_sql.orders`;
```

JOB INFORMATION		RESULTS	CHART	PREVIEW	JSON	EXEC
Row	order_id	time_to_deliver	diff_estimated_delivery			
1	1950d777989f6a877539f5379...	30	-12			
2	2c45c33d2f9cb8ff8b1c86cc28...	30	28			
3	65d1e226dfaeb8cdc42f66542...	35	16			
4	635c894d068ac37e6e03dc54e...	30	1			
5	3b97562c3aee8bdedcb5c2e45...	32	0			
6	68f47f50f04c4cb6774570cfde...	29	1			
7	276e9ec344d3bf029ff83a161c...	43	-4			
8	54e1a3c2b97fb0809da548a59...	40	-4			
9	fd04fa4105ee8045f6a0139ca5...	37	-1			
10	302bb8109d097a9fc6e9cefc5...	33	-5			

Insights: negative value means some orders are taking more days than estimated delivery date.

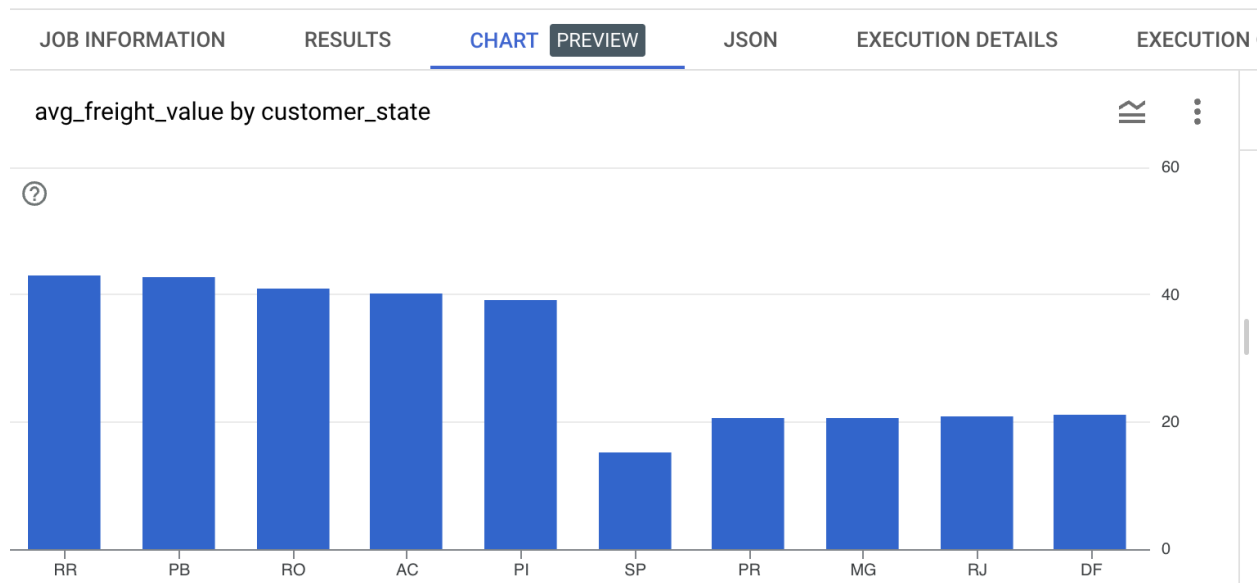
Recommendations ; to fix this it is necessary to place the order by the expected date. Therefore there is a need to work in this area.

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

ANSWER;

```
with final as (  
    select c.customer_state,  
           o.order_id,  
           oi.freight_value  
    from `secure-racer-402417.target_sql.orders` o  
    inner join `secure-racer-402417.target_sql.order_items` oi  
    on o.order_id = oi.order_id  
    inner join `secure-racer-402417.target_sql.customers` c  
    on c.customer_id = o.customer_id )  
(select customer_state,  
        avg(freight_value)as avg_freight_value  
from final  
group by customer_state  
order by avg_freight_value desc  
limit 5)  
  
union all  
  
(select customer_state,  
        avg(freight_value)as avg_freight_value  
from final  
group by customer_state  
order by avg_freight_value asc  
Limit 5);
```

JOB INFORMATION		RESULTS	CHART	PREVIEW
Row	customer_state	avg_freight_value		
1	RR	42.98442307692...		
2	PB	42.72380398671...		
3	RO	41.06971223021...		
4	AC	40.07336956521...		
5	PI	39.14797047970...		
6	SP	15.14727539041...		
7	PR	20.53165156794...		
8	MG	20.63016680630...		
9	RJ	20.96092393168...		
10	DF	21.04135494596...		



Insights: top 5 state with highest freight value

=RR,PB,RO,AC,PI

top 5 state with lowest freight value = SP,PR,MG,RJ,DF

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

ANSWER ;

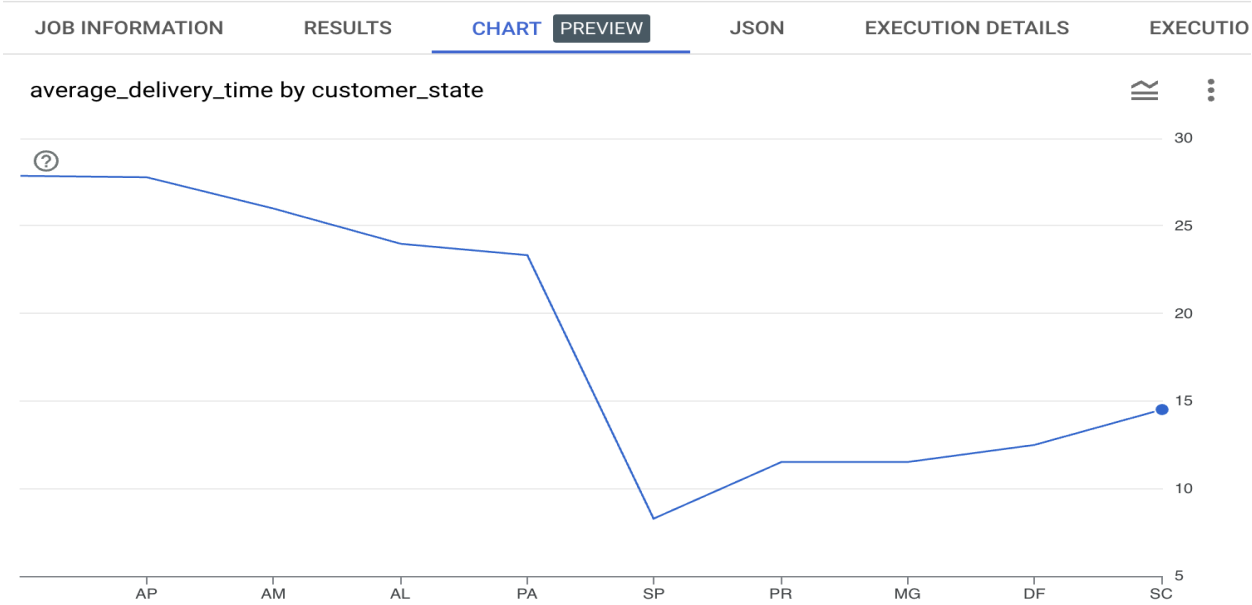
```
with final as
(
select c.customer_state,
       o.order_purchase_timestamp,
       o.order_delivered_customer_date,
       oi.shipping_limit_date
from `secure-racer-402417.target_sql.orders` o
inner join `secure-racer-402417.target_sql.order_items` oi
on o.order_id = oi.order_id
inner join `secure-racer-402417.target_sql.customers` c
on c.customer_id = o.customer_id
)
(select customer_state,
       avg(date_diff(order_delivered_customer_date, order_purchase_timestamp, day)) AS
average_delivery_time
from final
group by customer_state
order by average_delivery_time desc
limit 5 )

union all

(select customer_state,
       avg(date_diff(order_delivered_customer_date, order_purchase_timestamp, day)) AS
average_delivery_time
from final
group by customer_state
order by average_delivery_time asc
limit 5 );
```

Query results

JOB INFORMATION		RESULTS	CHART	PREVIEW
Row	customer_state	average_delivery_tim		
1	RR	27.82608695652...		
2	AP	27.75308641975...		
3	AM	25.96319018404...		
4	AL	23.99297423887...		
5	PA	23.30170777988...		
6	SP	8.259608552419...		
7	PR	11.48079306071...		
8	MG	11.51552218007...		
9	DF	12.50148619957...		
10	SC	14.52098584675...		



Insights: 5 states with the highest average delivery time = RR,AP,AM,AL,PA

5 states with the highest average delivery time = SP,PR,MG,DF,SC

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

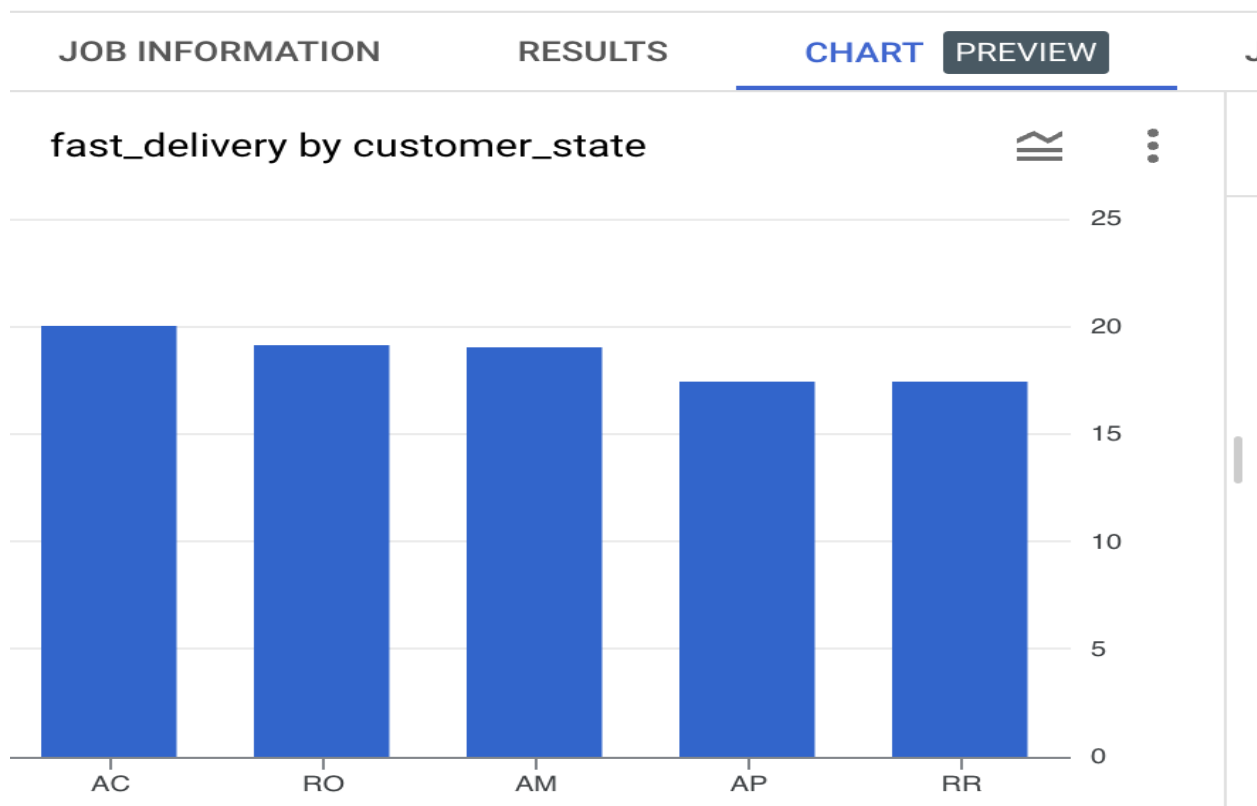
You can use the difference between the averages of actual & estimated delivery date to figure out how fast the delivery was for each state.

ANSWER ;

```
with final as
(
  select c.customer_state,o.order_id,
         o.order_estimated_delivery_date,
         o.order_delivered_customer_date,
  from `secure-racer-402417.target_sql.order_items` oi
  inner join `secure-racer-402417.target_sql.orders` o
    on o.order_id = oi.order_id
  inner join `secure-racer-402417.target_sql.customers` c
    on c.customer_id = o.customer_id
)
select customer_state,
avg(date_diff(order_estimated_delivery_date, order_delivered_customer_date,day) ) as
fast_delivery
from final
group by customer_state
order by fast_delivery desc
limit 5 ;
```

Query results

JOB INFORMATION		RESULTS	CHART	PREVIEW
Row	customer_state	fast_delivery		
1	AC	20.01098901098...		
2	RO	19.08058608058...		
3	AM	18.97546012269...		
4	AP	17.444444444444...		
5	RR	17.43478260869...		



Insights: IN ABOVE TABLE FAST DELIVERY IS HAPPENING IN STATE 'AC'

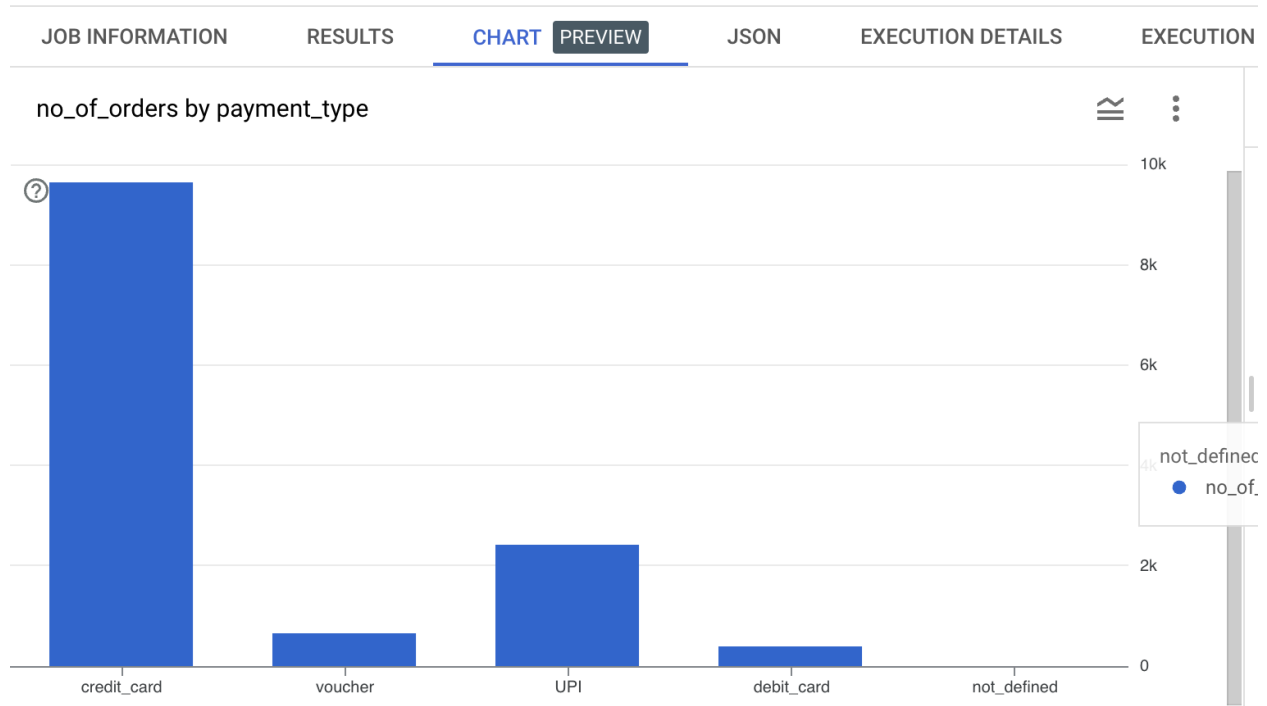
6. Analysis based on the payments:

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

ANSWER ;

```
with final as(
    select o.order_delivered_customer_date,
           p.payment_type,
           o.order_id
    from `secure-racer-402417.target_sql.payments` p
    inner join `secure-racer-402417.target_sql.orders` o
    on p.order_id = o.order_id
)
Select extract(month from order_delivered_customer_date) as month ,
       payment_type,
       count(order_id) as no_of_orders
from final
group by 1,2
order by 1 desc ;
```

JOB INFORMATION		RESULTS	CHART	PREVIEW	JSON	EXI
Row	month	payment_type	no_of_orders			
1	12	credit_card	5649			
2	12	voucher	325			
3	12	UPI	1440			
4	12	debit_card	64			
5	11	voucher	292			
6	11	credit_card	3643			
7	11	UPI	977			
8	11	debit_card	45			
9	10	voucher	293			
10	10	credit_card	3607			



Insights:most of the number of order placed by using credit_card payment type.

Credit_card > UPI > voucher > debit_card

6B. Find the no. of orders placed on the basis of the payment installments that have been paid.

ANSWER ;

```
select
    count(*)as no_of_order
from `secure-racer-402417.target_sql.payments`
where payment_installments > 1 ;
```

Query results

JOB INFORMATION		RESULTS	C
Row	no_of_order		
1	51338		

Insights : 51338 orders placed on the basis of the payment installments that have been paid.