

Target Business Case Analysis

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

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

Sol.

Query:

```
SELECT COLUMN_NAME, data_type
FROM tensile-spirit-350306.target_sql.INFORMATION_SCHEMA.COLUMNS
WHERE table_name = 'customers';
```

Output:

JOB INFORMATION		RESULTS	JSON
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:

Most columns in customers table are STRING type.

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

Sol.

Query:

```
SELECT MIN(order_purchase_timestamp) AS order_start,
MAX(order_purchase_timestamp) AS order_end
FROM tensile-spirit-350306.target_sql.orders;
```

Output:

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	order_start ▼	order_end ▼		
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC		

Insights:

The Market in Brazil was operational for 2 years.

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

Sol.

Query:

```
SELECT COUNT(DISTINCT c.customer_city) AS Total_Cities,  
COUNT(DISTINCT c.customer_state) AS Total_States  
FROM tensile-spirit-350306.target_sql.customers AS c  
INNER JOIN tensile-spirit-350306.target_sql.orders AS o ON  
c.customer_id=o.customer_id;
```

Output:

JOB INFORMATION		RESULTS	JSON
Row	Total_Cities	Total_States	
1	4119	27	

Insights:

Target was operational in 4119 cities and 27 states across Brazil during the 2 years.

2. In-depth Exploration:

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

Sol.

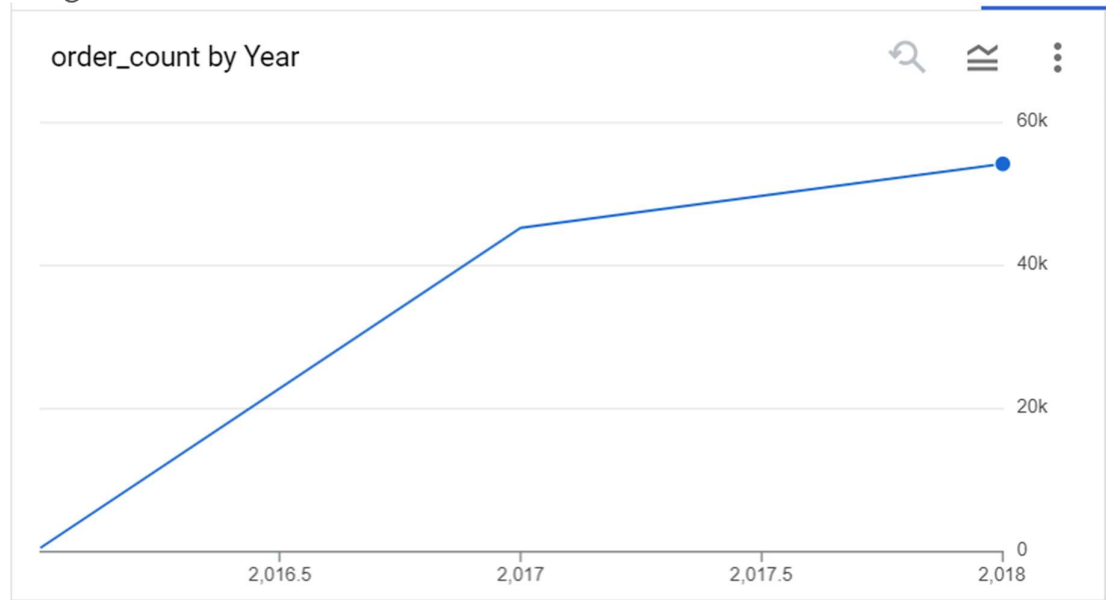
Query:

```
SELECT  
EXTRACT(Year FROM order_purchase_timestamp) AS Year,  
COUNT(order_id) AS order_count  
FROM tensile-spirit-350306.target_sql.orders  
GROUP BY Year  
ORDER BY Year;
```

Output:

JOB INFORMATION		RESULTS	JSON
Row	Year	order_count	
1	2016	329	
2	2017	45101	
3	2018	54011	

Insights:



Yes, there is an inclined trend in customer orders placed over the time period. From 2016 to 2017, no. of orders went too high in the trend however, after 2017, it slowed down a little.

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

Sol:

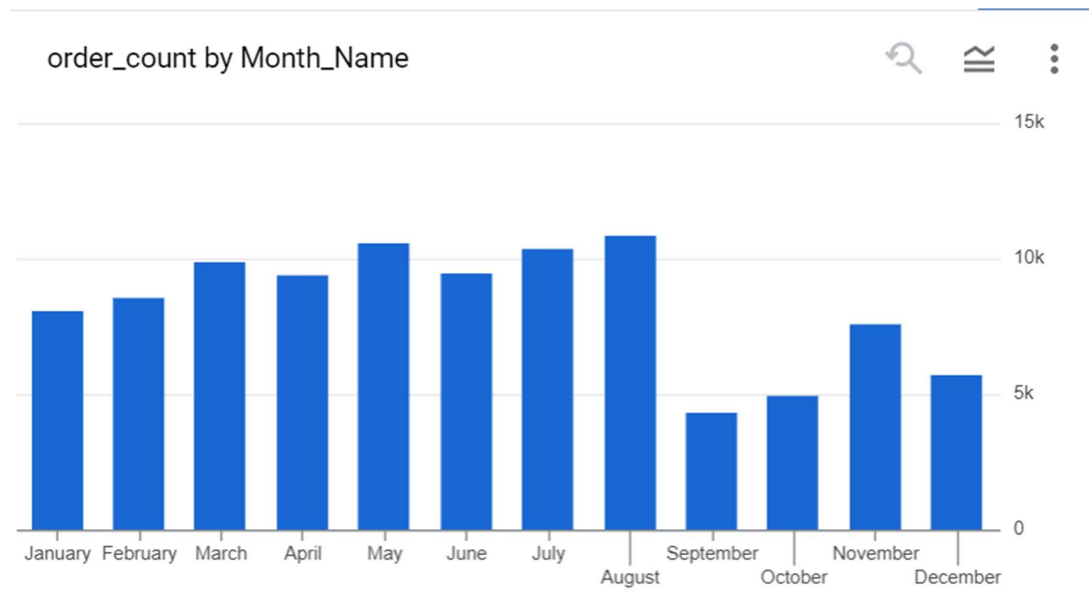
Query:

```
SELECT
EXTRACT(Month FROM order_purchase_timestamp) AS Month_Num,
FORMAT_TIMESTAMP('%B', order_purchase_timestamp) AS Month_Name,
COUNT(order_id) AS order_count
FROM tensile-spirit-350306.target_sql.orders
GROUP BY Month_Num, Month_Name
ORDER BY Month_Num ASC, Month_Name ASC;
```

Output:

JOB INFORMATION		RESULTS		JSON	EXECUTION DETAILS	
Row	Month_Num	Month_Name	order_count			
1	1	January	8069			
2	2	February	8508			
3	3	March	9893			
4	4	April	9343			
5	5	May	10573			
6	6	June	9412			
7	7	July	10318			
8	8	August	10843			
9	9	September	4305			
10	10	October	4959			

Insights:



From the above bar plot, we can see that yes there is some sort of seasonality in the no. of orders placed over the given period. From January to August, the orders were in inclined trend with a slight variation in between. However, the orders drastically fell in September and again went up in the trend.

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

Sol.

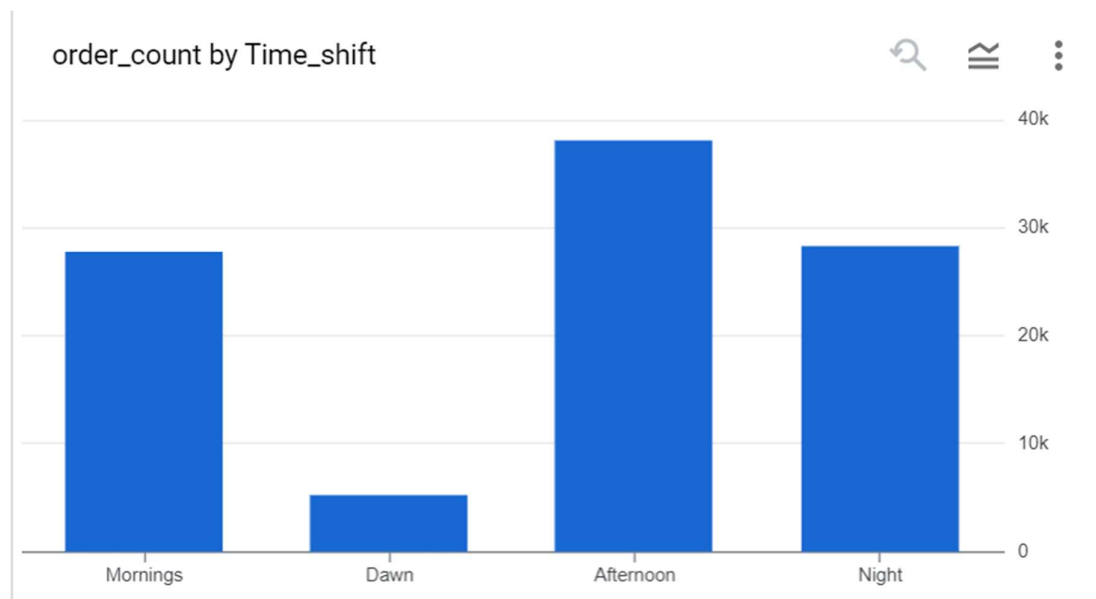
Query:

```
SELECT
CASE
WHEN EXTRACT(Hour FROM order_purchase_timestamp) BETWEEN 0 AND 6 THEN 'Dawn'
WHEN EXTRACT(Hour FROM order_purchase_timestamp) BETWEEN 7 AND 12 THEN
'Mornings'
WHEN EXTRACT(Hour FROM order_purchase_timestamp) BETWEEN 13 AND 18 THEN
'Afternoon'
WHEN EXTRACT(Hour FROM order_purchase_timestamp) BETWEEN 19 AND 23 THEN 'Night'
END AS Time_shift,
COUNT(order_id) AS order_count
FROM tensile-spirit-350306.target_sql.orders
GROUP BY Time_shift;
```

Output:

JOB INFORMATION		RESULTS	JSON	EXECUTION
Row	Time_shift ▼	order_count ▼		
1	Mornings	27733		
2	Dawn	5242		
3	Afternoon	38135		
4	Night	28331		

Insights:



From the plot, we can state that Brazillian customers mostly place their orders during **Afternoon**.

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

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

Sol.

Query:

```
SELECT
FORMAT_TIMESTAMP('%B', order_purchase_timestamp) AS Month_Name,
COUNT(order_id) AS order_count,
customer_state
FROM tensile-spirit-350306.target_sql.orders AS o inner join tensile-spirit-
350306.target_sql.customers AS c
ON o.customer_id = c.customer_id
GROUP BY Month_Name, customer_state;
```

Output:

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	Month_Name	order_count	customer_state	
1	November	1048	RJ	
2	December	283	RS	
3	December	2357	SP	
4	February	196	DF	
5	November	378	PR	
6	April	92	MT	
7	July	79	MA	
8	July	40	AL	
9	July	4381	SP	
10	July	85	MT	

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

Sol.

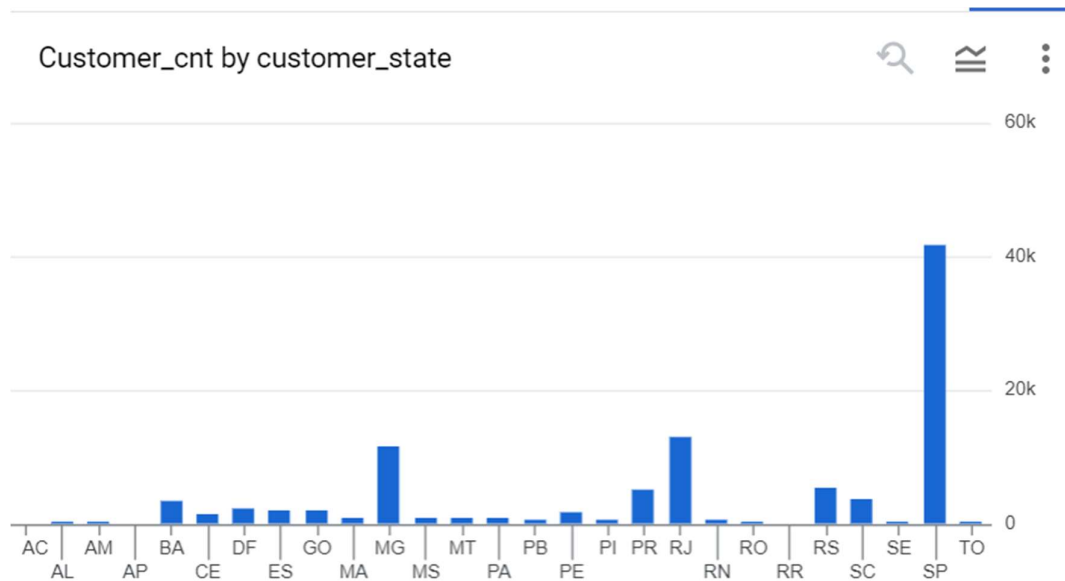
Query:

```
SELECT
customer_state,
COUNT(customer_id) AS Customer_cnt
FROM tensile-spirit-350306.target_sql.customers
GROUP BY customer_state
ORDER BY customer_state;
```

Output:

JOB INFORMATION		RESULTS	JSON	EXECUTIO
Row	customer_state	Customer_cnt		
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		

Insights:



From the above bar plot, we can state that Sao Paulo(SP) has the highest no. of customers while Roraima(RR) has the least no. of customers.

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

1. 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.

Sol.

Query:

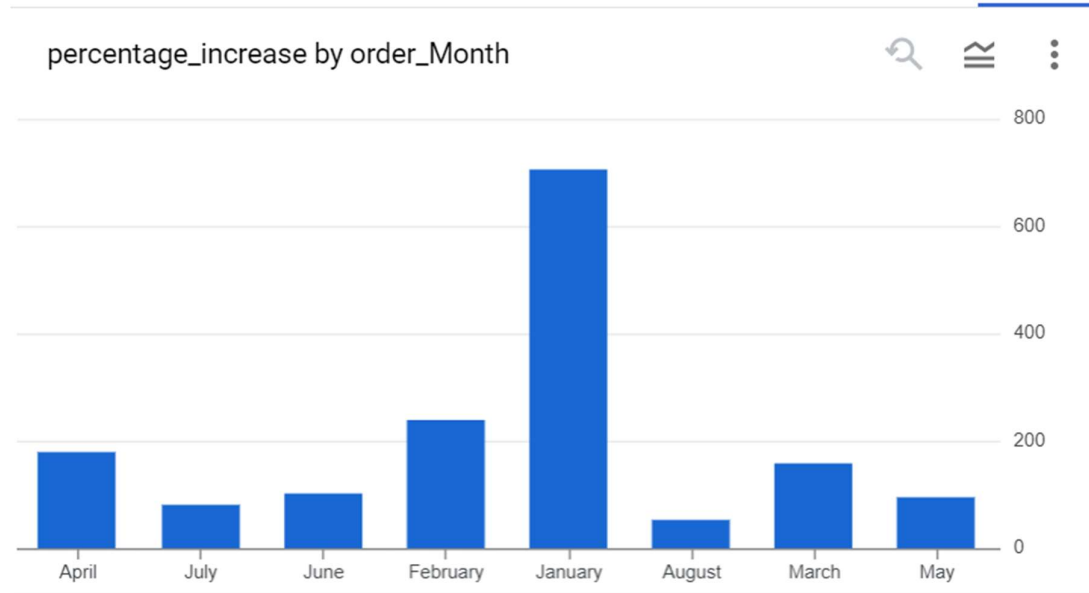
```
WITH t AS (SELECT
  EXTRACT(YEAR FROM o.order_purchase_timestamp) AS order_year,
  FORMAT_TIMESTAMP('%B', order_purchase_timestamp) AS order_Month,
  SUM(p.payment_value) AS total_payment
FROM tensile-spirit-350306.target_sql.orders AS o
  INNER JOIN tensile-spirit-350306.target_sql.payments AS p
  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),
  FORMAT_TIMESTAMP('%B', order_purchase_timestamp))
```

```
SELECT
  y_2017.order_Month,
  ((y_2018.total_payment - y_2017.total_payment) / y_2017.total_payment) * 100 AS
  percentage_increase
FROM t AS y_2017
  JOIN t AS y_2018
  ON y_2017.order_Month = y_2018.order_Month
  AND y_2017.order_year < y_2018.order_year;
```

Output:

JOB INFORMATION		RESULTS	JSON	EXECUTION I
Row	order_Month ▼	percentage_increase		
1	April	177.8407701149...		
2	July	80.04245463390...		
3	June	100.2596912456...		
4	February	239.9918145445...		
5	January	705.1266954171...		
6	August	51.60600520477...		
7	March	157.7786066709...		
8	May	94.62734375677...		

Insights:



From 2017 to 2018, the % increase in the cost of order was highest in January and lowest in August.

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

Sol.

Query:


```

With t AS (
SELECT
i.order_id,
o.customer_id,
sum(i.price) AS total_order_price
FROM tensile-spirit-350306.target_sql.order_items AS i inner join tensile-spirit-
350306.target_sql.orders AS o
on i.order_id=o.order_id
GROUP BY i.order_id, o.customer_id)

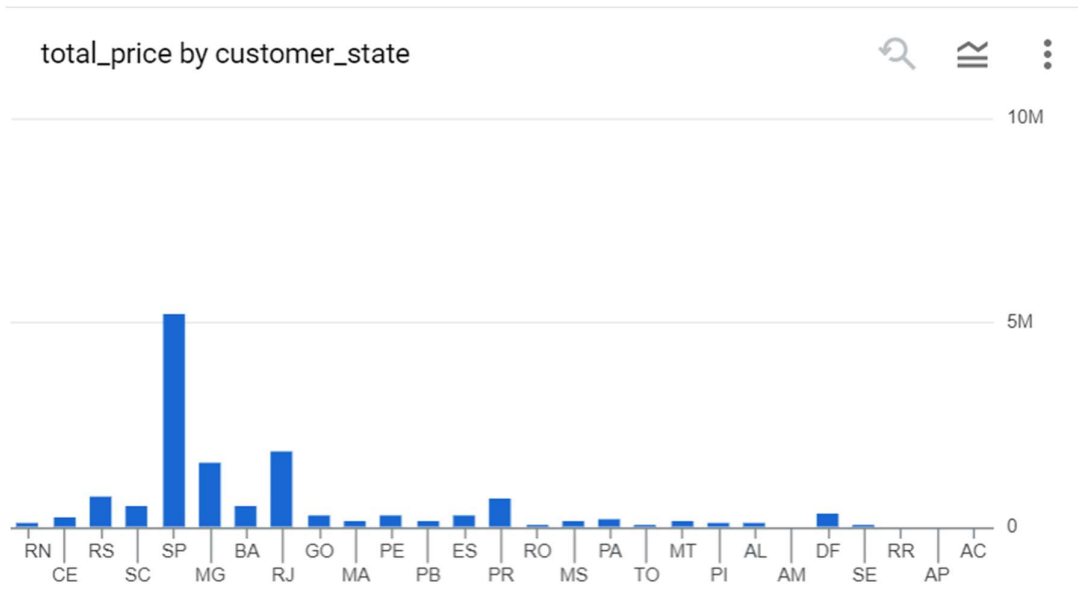
SELECT c.customer_state,
sum(t.total_order_price) AS total_price,
AVG(t.total_order_price) AS avg_price
FROM tensile-spirit-350306.target_sql.customers c INNER JOIN t on
c.customer_id = t.customer_id
GROUP BY customer_state;

```

Output:

Row	customer_state	total_price	avg_price
1	RN	83034.97999999...	172.2717427385...
2	CE	227254.70999999...	171.2544913338...
3	RS	750304.0200000...	138.1266605301...
4	SC	520553.3400000...	144.1177574750...
5	SP	5202955.050001...	125.7511794561...
6	MG	1585308.029999...	137.3274454261...
7	BA	511349.9900000...	152.2781387730...
8	RJ	1824092.669999...	142.9315679360...
9	GO	294591.9499999...	146.7822371699...
10	MA	119648.2199999...	161.6867837837...

Insights:



The total value of order price is highest in Sao Paulo(SP) and least in Roraima(RR) and Amapa(AP).



The average value of customer order price is highest in Paraiba(PB) and least in Sao Paulo(SP).

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

Sol.

Query:

```

With t AS (
SELECT
i.order_id,
o.customer_id,
sum(i.freight_value) AS total_freight
FROM tensile-spirit-350306.target_sql.order_items AS i inner join tensile-spirit-
350306.target_sql.orders AS o
on i.order_id=o.order_id
GROUP BY i.order_id, o.customer_id)

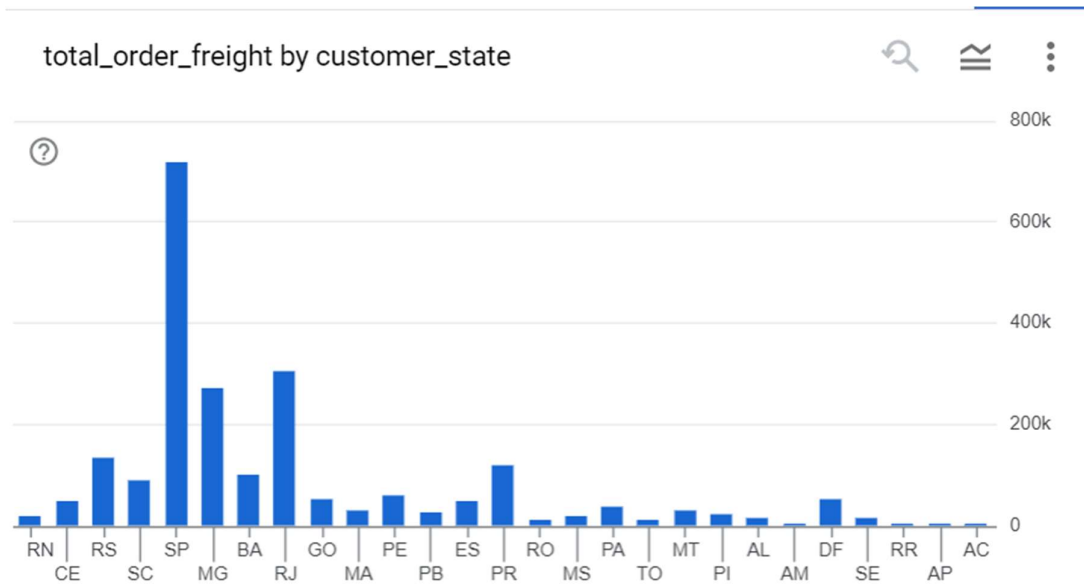
SELECT c.customer_state,
sum(t.total_freight) AS total_order_freight,
AVG(t.total_freight) AS avg_order_freight
FROM tensile-spirit-350306.target_sql.customers c INNER JOIN t on
c.customer_id = t.customer_id
GROUP BY customer_state;

```

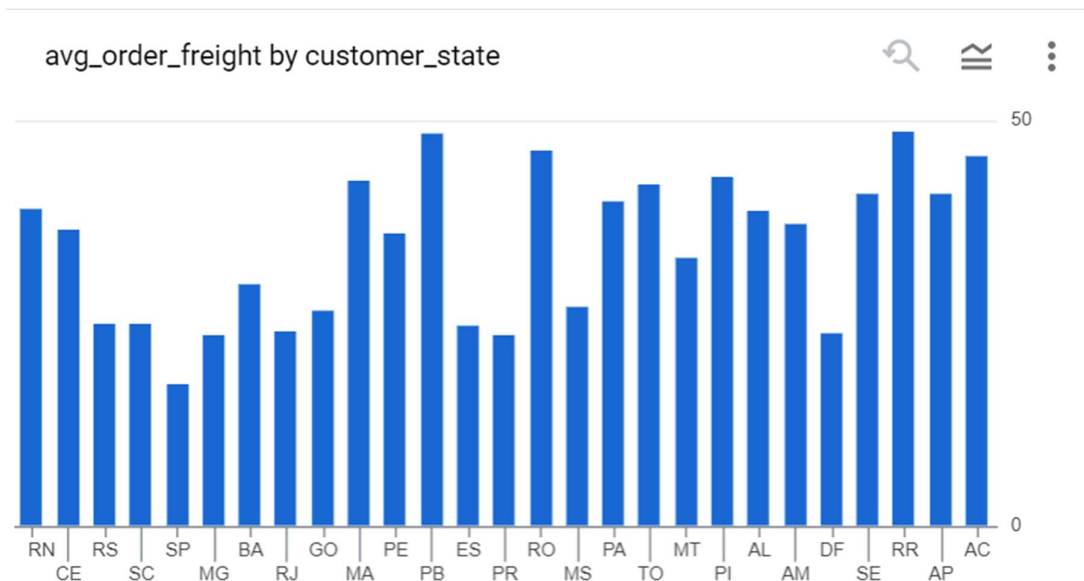
Output:

Row	customer_state	total_order_freight	avg_order_freight
1	RN	18860.10000000...	39.12883817427...
2	CE	48351.58999999...	36.43676714393...
3	RS	135522.74000000...	24.94895802650...
4	SC	89660.26000000...	24.82288482834...
5	SP	718723.06999999...	17.37095033232...
6	MG	270853.46000000...	23.46270443520...
7	BA	100156.67999999...	29.82628945801...
8	RJ	305589.31000000...	23.94525231155...
9	GO	53114.97999999...	26.46486297957...
10	MA	31523.77000000...	42.59968918918...

Insights:



The total value of order freight is highest for Sao Paulo(SP) and lowest for Roraima(RR).



The average value of order freight is highest for Roraima(RR) and lowest for Sao Paulo(SP).

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

- 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

Sol.

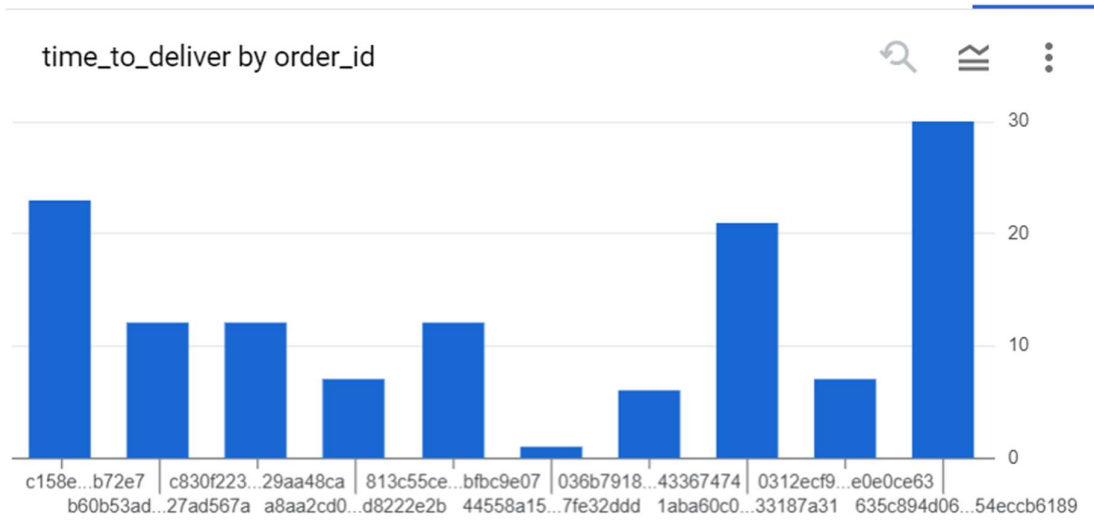
Query:

```
SELECT
order_id,
DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY) AS
time_to_deliver,
DATE_DIFF(order_estimated_delivery_date, order_delivered_customer_date, DAY) AS
diff_estimated_delivery
FROM tensile-spirit-350306.target_sql.orders
WHERE order_status = 'delivered'
LIMIT 10;
```

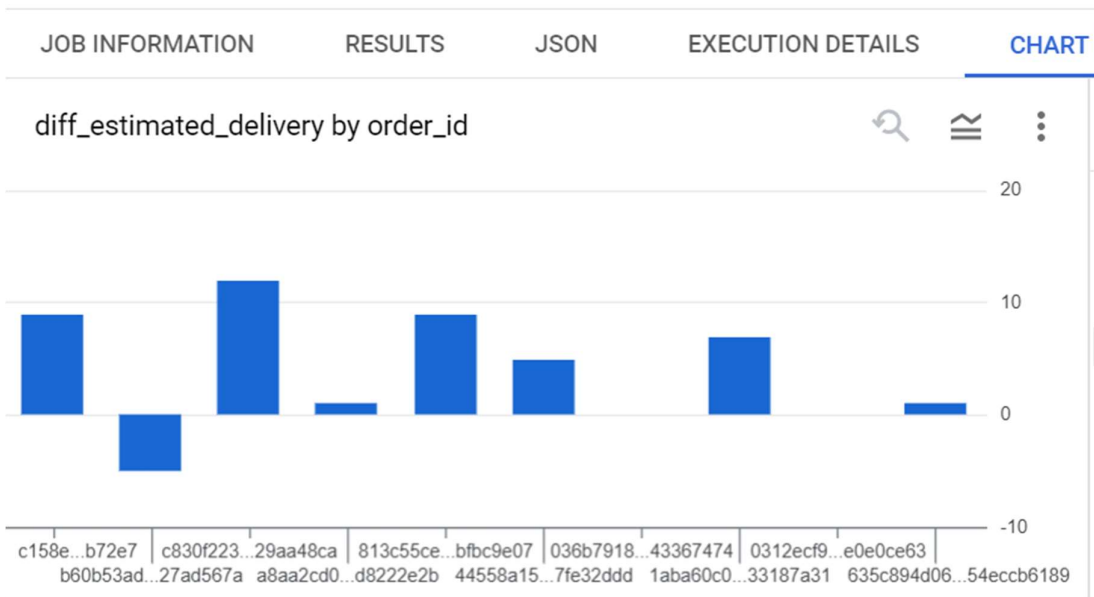
Output:

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	CH
Row	order_id	time_to_deliver	diff_estimated_delivery		
1	c158e9806f85a33877bdfd4f60...	23	9		
2	b60b53ad0bb7dacacf2989fe2...	12	-5		
3	c830f223aae08493ebecb52f2...	12	12		
4	a8aa2cd070eeac7e4368cae3d...	7	1		
5	813c55ce9b6baa8f879e064fbf...	12	9		
6	44558a1547e448b41c48c4087...	1	5		
7	036b791897847cdb8e39df794...	6	0		
8	1aba60c04110bdd421b250ea3...	21	7		
9	0312ecf90786def87f98aa19e0...	7	0		
10	635c894d068ac37e6e03dc54e...	30	1		

Insights:



From the above bar plot based on the 10 order ids, we can state that delivery time is highest for order id **635c894d068ac37e6e03dc54eccb6189**.



The difference between estimated delivery time and actual delivery date of order is highest for the order id **c830f223aae08493ebecb52f29aa48ca** and lowest for order id **b60b53ad0bb7dacacf2989fe27ad567a** because the order was actually delivered 5 days before the estimated delivery date.

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

Sol.

Query:

Output:

```
With t AS (  
SELECT  
i.order_id,  
o.customer_id,  
sum(i.freight_value) AS total_freight  
FROM tensile-spirit-350306.target_sql.order_items AS i inner join tensile-spirit-  
350306.target_sql.orders AS o  
on i.order_id=o.order_id  
GROUP BY i.order_id, o.customer_id)
```

```
SELECT c.customer_state,  
AVG(t.total_freight) AS highest_avg_order_freight  
FROM tensile-spirit-350306.target_sql.customers c INNER JOIN t on  
c.customer_id = t.customer_id  
GROUP BY customer_state  
ORDER BY highest_avg_order_freight DESC  
LIMIT 5;
```

###-----##

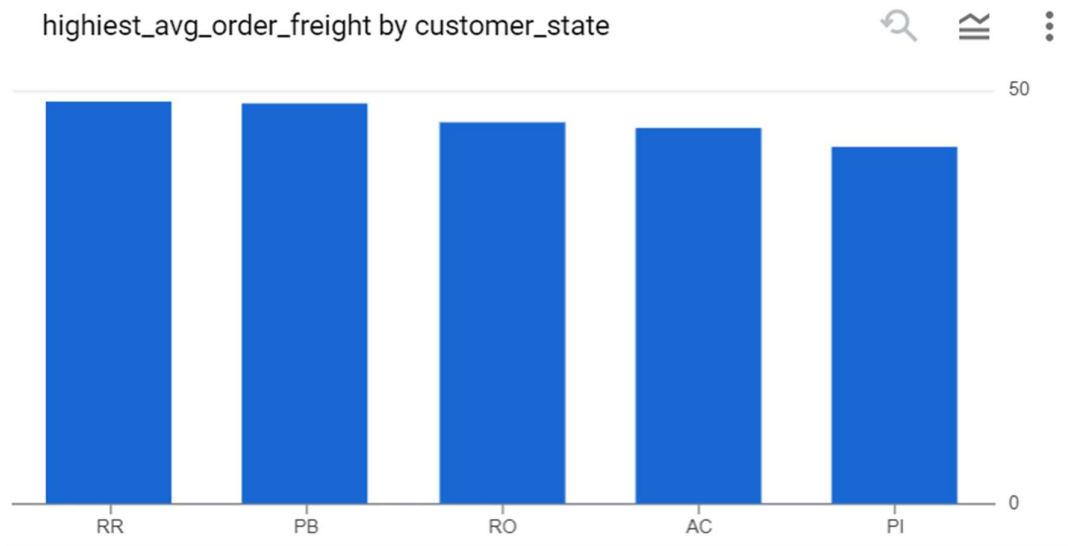
```
With t AS (  
SELECT  
i.order_id,  
o.customer_id,  
sum(i.freight_value) AS total_freight  
FROM tensile-spirit-350306.target_sql.order_items AS i inner join tensile-spirit-  
350306.target_sql.orders AS o  
on i.order_id=o.order_id  
GROUP BY i.order_id, o.customer_id)
```

```
SELECT c.customer_state,  
AVG(t.total_freight) AS lowest_avg_order_freight  
FROM tensile-spirit-350306.target_sql.customers c INNER JOIN t on  
c.customer_id = t.customer_id  
GROUP BY customer_state  
ORDER BY lowest_avg_order_freight ASC  
LIMIT 5;
```

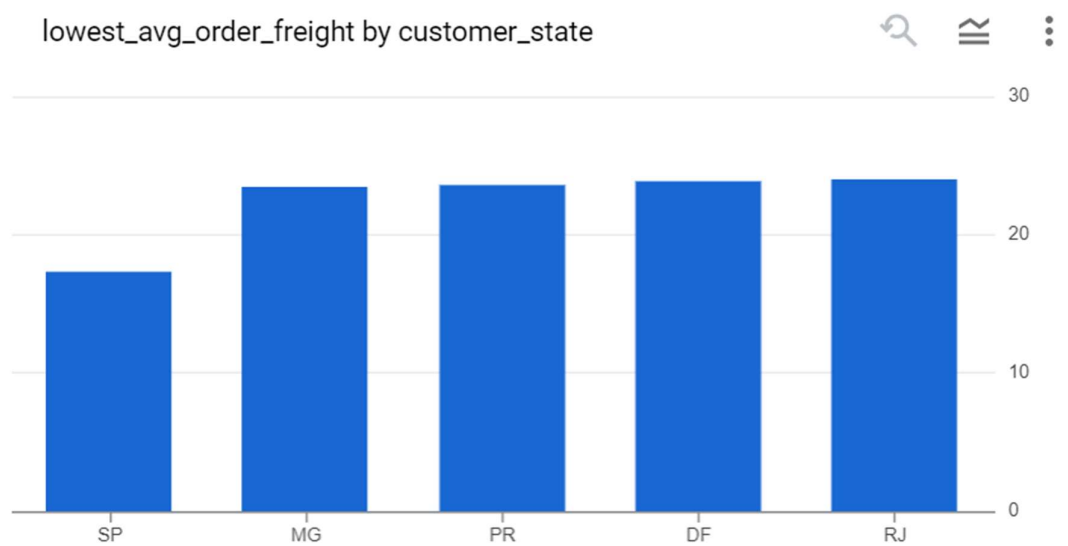
Output:

Row	customer_state	highest_avg_order_freight	Row	customer_state	lowest_avg_order_freight
1	RR	48.591086956521742	1	SP	17.370950332326089
2	PB	48.3453571428572	2	MG	23.46270443520438
3	RO	46.224210526315773	3	PR	23.579767907162864
4	AC	45.515432098765444	4	DF	23.823764705882333
5	PI	43.038945233265778	5	RJ	23.945252311550078

Insights:



Out of 27 states of Brazil based on highest average order freight, the top 5 states are Roraima(RR), Paraiba(PB), Rondonia (RO), Acre(AC) and Piaui(PI).



The top 5 states in Brazil Market, based on lowest average order freight are Sao Paulo(SP), Minas Gerais(MG), Parana(PR), Distrito Federal(DF) and Rio De Janerio(RJ).

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

Sol.

Query:


```

With cte AS
(
SELECT
order_id,
customer_id,
SUM(DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY)) AS
time_to_deliver
FROM tensile-spirit-350306.target_sql.orders
WHERE order_status = 'delivered'
GROUP BY order_id,customer_id
)

```

```

SELECT
c.customer_state,
AVG(cte.time_to_deliver) AS highest_avg_delivery_time
FROM tensile-spirit-350306.target_sql.customers AS c INNER JOIN cte on c.customer_id=
cte.customer_id
GROUP BY c.customer_state
ORDER BY highest_avg_delivery_time DESC
LIMIT 5;

```

##-----

```

With cte AS
(
SELECT
order_id,
customer_id,
SUM(DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY)) AS
time_to_deliver
FROM tensile-spirit-350306.target_sql.orders
WHERE order_status = 'delivered'
GROUP BY order_id,customer_id
)

```

```

SELECT
c.customer_state,
AVG(cte.time_to_deliver) AS lowest_avg_delivery_time
FROM tensile-spirit-350306.target_sql.customers AS c INNER JOIN cte on c.customer_id=
cte.customer_id
GROUP BY c.customer_state
ORDER BY lowest_avg_delivery_time ASC
LIMIT 5;

```

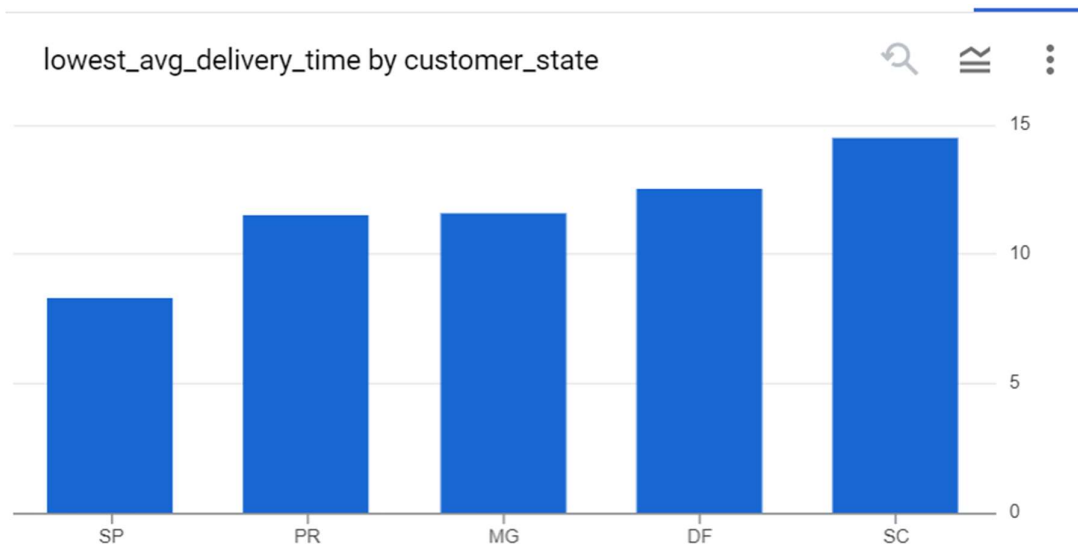
Output:

Row	customer_state	highest_avg_delivery_time	Row	customer_state	lowest_avg_delivery_time
1	RR	28.975609756097562	1	SP	8.2980935447227253
2	AP	26.731343283582088	2	PR	11.526711354864963
3	AM	25.986206896551735	3	MG	11.54218775233405
4	AL	24.040302267002509	4	DF	12.509134615384614
5	PA	23.316067653276953	5	SC	14.475183305132523

Insights:



The top 5 states in Brazil Market based on highest average delivery time are Roraima(RR), Amapa(AP), Amazonas(AM), Alagoas(AL) and Para(PA).



The top 5 states based on lowest average delivery time are Sao Paulo(SP), PR(Parana), Minas Gerais(MG), Distrito Federal (DF) and Santa Catarina(SC).

- 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.

Sol.

Query:

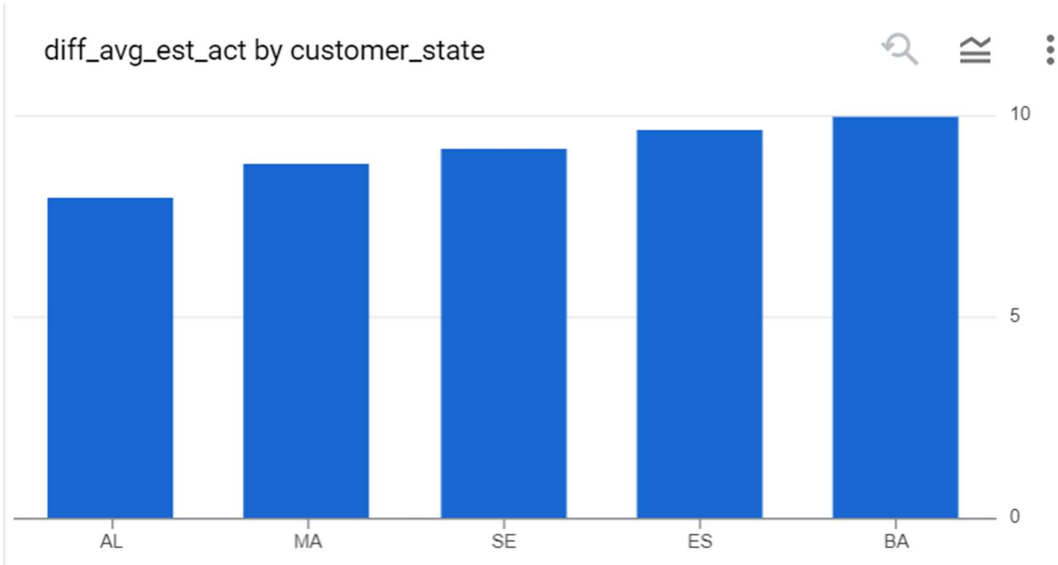
```
WITH t AS
(
  SELECT order_id,
  customer_id,
  SUM(DATE_DIFF(order_estimated_delivery_date, order_delivered_customer_date, DAY)) AS
diff_estimated_delivery
FROM tensile-spirit-350306.target_sql.orders
WHERE order_status = 'delivered'
GROUP BY order_id,customer_id
)

SELECT
c.customer_state,
AVG(t.diff_estimated_delivery) AS diff_avg_est_act
FROM tensile-spirit-350306.target_sql.customers AS c INNER JOIN t on t.customer_id =
c.customer_id
GROUP BY c.customer_state
ORDER BY diff_avg_est_act ASC
LIMIT 5;
```

Output:

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	customer_state	diff_avg_est_act		
1	AL	7.9471032745592		
2	MA	8.768479776847...		
3	SE	9.173134328358...		
4	ES	9.618546365914...		
5	BA	9.934889434889...		

Insights:



The top 5 states in Brazil Market where order delivery is as fast as compared to the estimated date of delivery are Alagoas(AL), Maranhao(MA), Sergipe(SE), Espirito Santo(ES) and Bahia(BA).

6. Analysis based on the payments:

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

Sol.

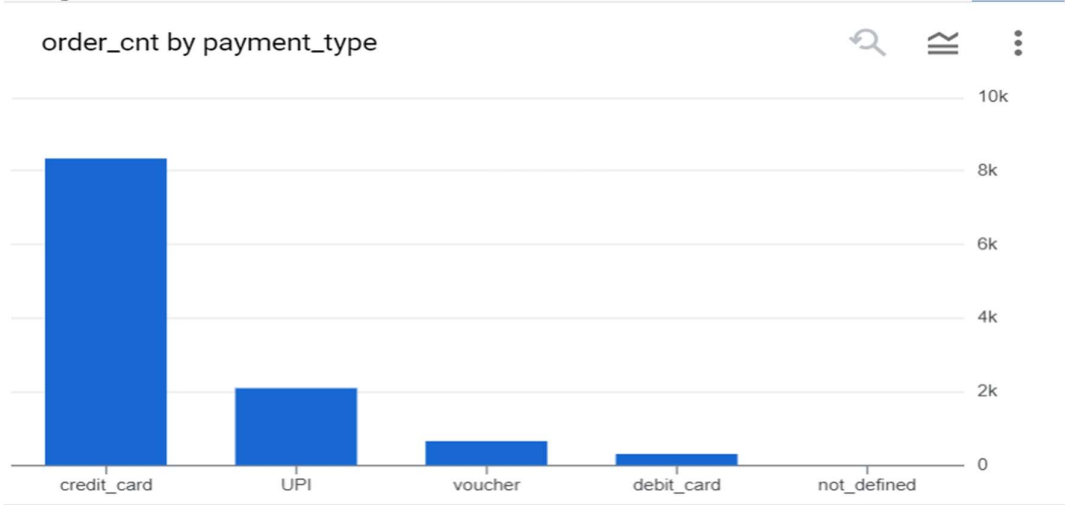
Query:

```
SELECT
EXTRACT(Month FROM o.order_purchase_timestamp) AS Month_Num,
FORMAT_TIMESTAMP('%B', o.order_purchase_timestamp) AS Month_Name,
COUNT(o.order_id) AS order_cnt,
p.payment_type
FROM tensile-spirit-350306.target_sql.orders AS o INNER JOIN tensile-spirit-350306.target_sql.payments AS p
ON o.order_id = p.order_id
GROUP BY Month_Num,Month_Name,payment_type
ORDER BY Month_Num;
```

Output:

JOB INFORMATION		RESULTS		JSON	EXECUTION DETAILS	CHART	PREVIEW	EXI
Row	Month_Num	Month_Name	order_cnt	payment_type				
1	1	January	6103	credit_card				
2	1	January	1715	UPI				
3	1	January	477	voucher				
4	1	January	118	debit_card				
5	2	February	1723	UPI				
6	2	February	6609	credit_card				
7	2	February	424	voucher				
8	2	February	82	debit_card				
9	3	March	7707	credit_card				
10	3	March	1942	UPI				

Insights:



From the plot, we can state that, maximum number of orders placed by Brazilian customers are through credit card.

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

Sol.

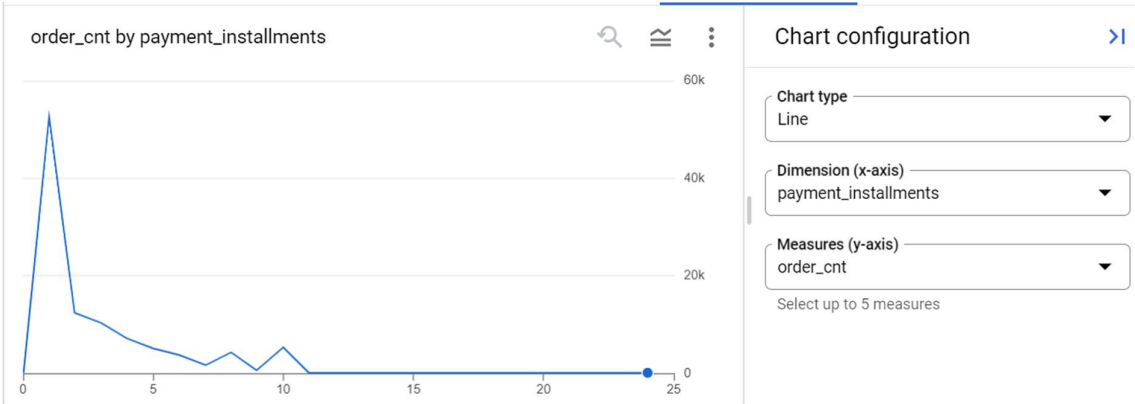
Query:

```
SELECT
p.payment_installments,
COUNT(o.order_id) AS order_cnt
FROM tensile-spirit-350306.target_sql.orders AS o INNER JOIN tensile-spirit-350306.target_sql.payments AS p
ON o.order_id=p.order_id
GROUP BY p.payment_installments
ORDER BY p.payment_installments;
```

Output:

JOB INFORMATION		RESULTS	JSON
Row	payment_installment	order_cnt	
1	0	2	
2	1	52546	
3	2	12413	
4	3	10461	
5	4	7098	
6	5	5239	
7	6	3920	
8	7	1626	
9	8	4268	
10	9	644	

Insights:



From the above line plot, we can state that the number of orders placed by customers are highest when the instalment is 1 and gradually with the increase in no. of instalments, the no. of orders declined drastically.

Recommendations:

Target's operation in Brazil lasted for 2 years, spanning across 27 states which were driven by customer numbers, shipping efficiency, payment efficiency and customer satisfaction levels to name a few.

Based on above analysis, Target should have focused on below recommendations in order to make a turnaround in its operations.

- Out of 27 states, Sao Paulo was having the highest number of customers and Roraima was the lowest. Surprisingly, the average time of delivery to customers was highest in Roraima and lowest in Sao Paulo. This could have been the cause of lack of resources or geographical positioning of customers in the state of Roraima.
- States like Roraima, Paraiba, Rondonia had highest average order freight which is another factor for declined customer satisfaction level.
- Credit card was the most popular payment type for customers so analysts at Target should have focused on other payments types like UPI, debit card and should have provided some incentives or perks for using other payment types.
- It was observed that, number of customer orders were maximum when the No. of payment instalments were limited to 1,2 or 3. So, Target should have focused on customer order of products with lesser no. of instalments and should have added some benefits from customer perspective for products with more no. of instalments.