

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 :

```
select column_name,
data_type
from Business_Case_Target_SQL.INFORMATION_SCHEMA.COLUMNS
where table_name='customers';
```

Output :

JOB INFORMATION		RESULTS	CHART	PREVIEW	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 : Customers table has more string datatype as compared to others. We explored information schema to get datatype

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

Answer :

```
select min(order_purchase_timestamp) as first_order_date,max(order_purchase_timestamp) as
last_order_date
from Business_Case_Target_SQL.orders;
```

Output :

Query results

JOB INFORMATION		RESULTS	CHART	PREVIEW	JSC
Row	first_order_date		last_order_date		
1	2016-09-04 21:15:19 UTC		2018-10-17 17:30:18 UTC		

Insights: Got first order purchase datetime (2016-09-04 21:15:19 UTC) and last order purchase datetime(2018-10-17 17:30:18 UTC) from orders table in between order were placed.

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

Answer:

```
with cte as (
select customer_city, customer_state, order_purchase_timestamp,
first_value(order_purchase_timestamp) over(order by order_purchase_timestamp )
as first_order_date,
last_value(order_purchase_timestamp) over(order by order_purchase_timestamp
range between unbounded preceding and unbounded following) as last_order_date
from Business_Case_Target_SQL.orders o JOIN Business_Case_Target_SQL.customers c
using(customer_id)
)
select count(distinct customer_city) as total_city, count(distinct
customer_state) as total_state
from cte
where order_purchase_timestamp between first_order_date and last_order_date;
```

Output :

Query results			
JOB INFORMATION		RESULTS	CHAR
Row	total_city	total_state	
1	4119	27	

Insights : Ordered placed by customer from 4119 cities and 27 states during period between 2016-09-04 21:15:19 UTC and 2018-10-17 17:30:18 UTC .

2) In-depth Exploration:

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

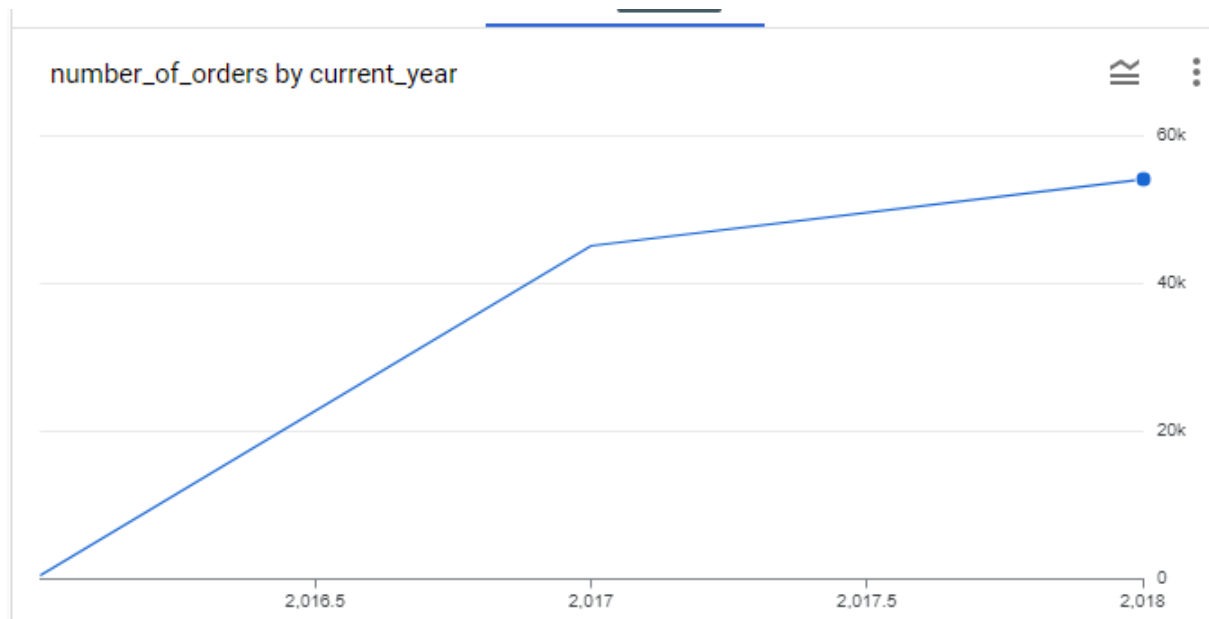
Answer:

```
select
extract(year from order_purchase_timestamp) as current_year,
count(order_id) as number_of_orders
from Business_Case_Target_SQL.orders
group by current_year
order by 1;
```

Output :

Query results			
JOB INFORMATION		RESULTS	CHART
Row	current_year	number_of_orders	
1	2016	329	
2	2017	45101	
3	2018	54011	

Business Case : Target SQL



Insights :

We have observed that number of orders increased from year 2016 to 2018 and from year 2016 to 2017 ,number of orders has increased massively high compared to 2018.

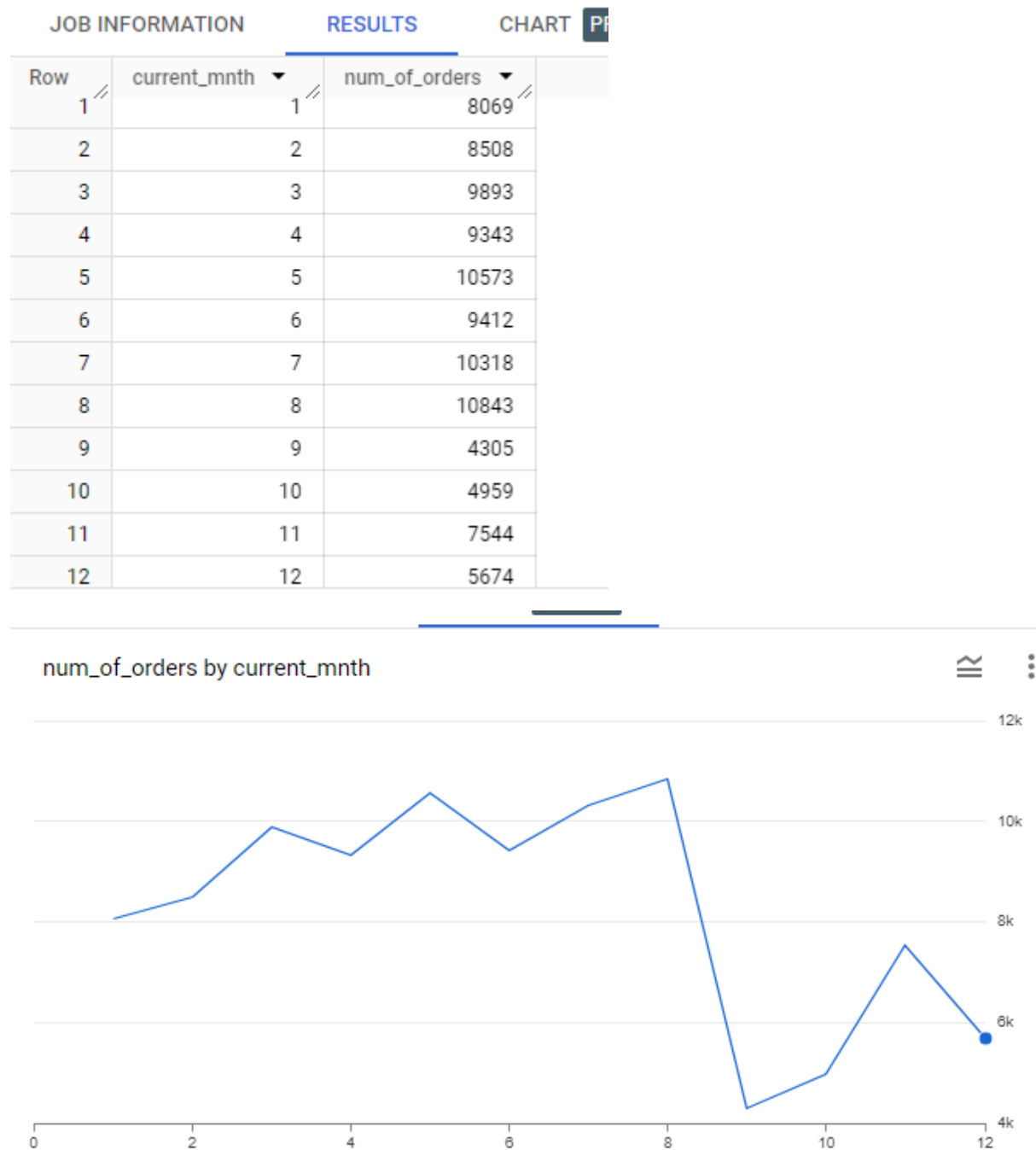
- b. Can we see some kind of monthly seasonality in terms of the no. of orders being placed?

Answer:

```
select
extract(month from order_purchase_timestamp) as current_mnth,
count(*) as num_of_orders
from Business_Case_Target_SQL.orders
group by 1
order by 1,2;
```

Output :

Business Case : Target SQL



Insights : Observed that there is sudden drop of orders from 9th month.

- 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 :

```
select
case
when extract(HOUR from order_purchase_timestamp) between 0 and 6 then 'Dawn'
```

Business Case : Target SQL

```
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 period,
count(*) as total_order
from Business_Case_Target_SQL.orders
group by 1
order by total_order desc;
```

Output :

JOB INFORMATION		RESULTS	CHART	PREVIEW	JSON
Row	period	total_order			
1	Afternoon	38135			
2	Night	28331			
3	Mornings	27733			
4	Dawn	5242			

Insights :

I have observe that Brazilian customer are ordering more in afternoon time and very less at dawn time as compared to others ,order placed in night and mornings are slightly similar.

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

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

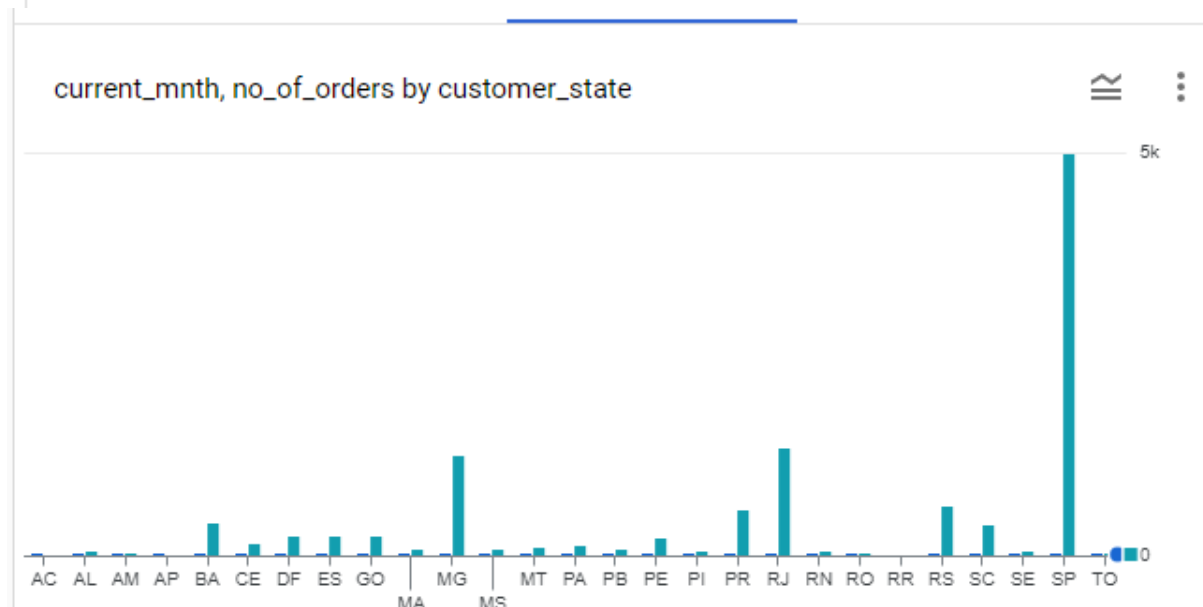
Answer:

```
select customer_state, extract(month from order_purchase_timestamp) as
current_mnth,
count(order_id) as no_of_orders
from Business_Case_Target_SQL.customers c LEFT JOIN Business_Case_Target_SQL.orders
o using(customer_id)
group by 1,2
order by 1,2;
```

Output:

Business Case : Target SQL

Row	customer_state	current_mnth	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



Insights:

Above graph shows that SP state has more number of order as compare to others, Whereas AC state has least .

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

Answer:

```
select customer_state, count(distinct customer_id) as total_cust
from Business_Case_Target_SQL.customers
group by 1
order by 2 desc;
```

Output :

Row	customer_state	total_cust
1	SP	41746
2	RJ	12852
3	MG	11635
4	RS	5466
5	PR	5045
6	SC	3637
7	BA	3380
8	DF	2140
9	ES	2033
10	GO	2020
11	PE	1652

Insights:

As per result, 41746 customer are from SP state which is higher compared to other states and state PE has the least number of customers.

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

Answer:

```
with cte_2017 as (
select
extract(year from order_purchase_timestamp) as year ,
sum(payment_value) as pur_1
from Business_Case_Target_SQL.payments p
JOIN Business_Case_Target_SQL.orders o using(order_id)
where extract(year from order_purchase_timestamp )= 2017 and
extract(month from order_purchase_timestamp) between 1 and 8
group by year
),
cte_2018 as (
select
extract(year from order_purchase_timestamp) as year ,
sum(payment_value) as pur_2
from Business_Case_Target_SQL.payments p
JOIN Business_Case_Target_SQL.orders o using(order_id)
where extract(year from order_purchase_timestamp )= 2018
and extract(month from order_purchase_timestamp) between 1 and 8
```

Business Case : Target SQL

```
group by year  
)
```

```
select round(100*(pur_2-pur_1)/pur_1,2) as percent_increase  
from cte_2017,cte_2018
```

Output :

JOB INFORMATION		RESULTS
Row	percent_increase	
1	136.98	

Insights : Observed that 2017 to 2018 has 139.98 percentage of increment happened in the cost of orders.

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

Answer:

```
with cte as(  
select customer_state,  
sum(price) as Total_value,  
count(distinct o.order_id) as num_orders  
from Business_Case_Target_SQL.customers c  
left join Business_Case_Target_SQL.orders o using(customer_id)  
left JOIN Business_Case_Target_SQL.order_items oi ON oi.order_id=o.order_id  
group by 1  
)  
select customer_state,Total_value,Total_value/num_orders as Average_value  
from cte  
order by 2 desc;
```

Output :

Row	customer_state	Total_value	Average_value
1	SP	5202955.050001...	124.6336187898...
2	RJ	1824092.669999...	141.9306465919...
3	MG	1585308.029999...	136.2533760206...
4	RS	750304.0200000...	137.2674753018...
5	PR	683083.7600000...	135.3981684836...
6	SC	520553.3400000...	143.1271212537...
7	BA	511349.9900000...	151.2869792899...
8	DF	302603.9399999...	141.4037102803...
9	GO	294591.9499999...	145.8375990098...
10	ES	275037.3099999...	135.2864289227...

Business Case : Target SQL

Insights :

SP state has highest order price of 5202955.0500015272

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

Answer :

```
with cte as(
select customer_state,
sum(freight_value) as Total_value,
count(distinct o.order_id) as num_orders
from Business_Case_Target_SQL.customers c
left join Business_Case_Target_SQL.orders o using(customer_id)
left JOIN Business_Case_Target_SQL.order_items oi ON oi.order_id=o.order_id
group by 1
)
select customer_state,Total_value,Total_value/num_orders as Average_value
from cte
order by 2 desc;
```

Output :

Row	customer_state	Total_value	Average_value
1	SP	718723.0699999...	17.21657332439...
2	RJ	305589.3100000...	23.77756847183...
3	MG	270853.4600000...	23.27919724967...
4	RS	135522.7400000...	24.79376875228...
5	PR	117851.6800000...	23.36009514370...
6	BA	100156.6799999...	29.63215384615...
7	SC	89660.26000000...	24.65225735496...
8	PE	59449.65999999...	35.98647699757...
9	GO	53114.97999999...	26.29454455445...
10	DF	50625.49999999...	23.65677570093...

Insights :SP state has highest total freight order value with 718723.06999999518

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:

- i. **time_to_deliver** = order_delivered_customer_date - order_purchase_timestamp
- ii. **diff_estimated_delivery** = order_estimated_delivery_date - order_delivered_customer_date

Answer :

```
select
order_id,
timestamp_diff(order_delivered_customer_date,order_purchase_timestamp,DAY) as
time_to_del,
timestamp_diff(order_delivered_customer_date,order_estimated_delivery_date,DAY)
as diff_estimated_dil
from Business_Case_Target_SQL.orders
where order_status = 'delivered';
```

Output :

Row	order_id	time_to_del	diff_estimated_dil
1	635c894d068ac37e6e03dc54e...	30	-1
2	3b97562c3aee8bdedcb5c2e45...	32	0
3	68f47f50f04c4cb6774570cfde...	29	-1
4	276e9ec344d3bf029ff83a161c...	43	4
5	54e1a3c2b97fb0809da548a59...	40	4
6	fd04fa4105ee8045f6a0139ca5...	37	1
7	302bb8109d097a9fc6e9cefc5...	33	5
8	66057d37308e787052a32828...	38	6
9	19135c945c554eebfd7576c73...	36	2
10	4493e45e7ca1084efcd38ddeb...	34	0

Insights : Observed that diff_estimated_delivery are negative numbers that are late delivery with respective number days and positive numbers in diff_estimated_delivery are early delivered than estimated date.

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

Answer:

----TOP 5 STATE WITH HIGHEST AVERAGE FRIGHT VALUE BELOW--

```
with cte as(
select customer_state,
sum(freight_value) as Total_value,
count(distinct o.order_id) as num_orders
```

Business Case : Target SQL

```
from Business_Case_Target_SQL.customers c
left join Business_Case_Target_SQL.orders o using(customer_id)
left JOIN Business_Case_Target_SQL.order_items oi ON oi.order_id=o.order_id
group by 1
)

select customer_state,Total_value/num_orders as Average_value
from cte
order by Average_value desc
limit 5;
```

Output :

Row	customer_state	Average_value
1	RR	48.59108695652...
2	PB	47.98457089552...
3	AC	45.51543209876...
4	RO	45.12798418972...
5	PI	42.86505050505...

----TOP 5 STATE WITH LOWEST AVERAGE --- BELOW---

```
with cte as(
select customer_state,
sum(freight_value) as Total_value,
count(distinct o.order_id) as num_orders
from Business_Case_Target_SQL.customers c
left join Business_Case_Target_SQL.orders o using(customer_id)
left JOIN Business_Case_Target_SQL.order_items oi ON oi.order_id=o.order_id
group by 1
)

select customer_state,Total_value/num_orders as Average_value
from cte
order by Average_value
limit 5;
```

OUTPUT :

Row	customer_state	Average_value
1	SP	17.21657332439...
2	MG	23.27919724967...
3	PR	23.36009514370...
4	DF	23.65677570093...
5	RJ	23.77756847183...

Insights : Top 5 state with highest average value are : RR,PB,AC,RO,PI
and

Top 5 state with lowest average value are : SP,MG,DF,PR,RJ

Business Case : Target SQL

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

Answer:

```
with base as(
select customer_state,
sum(TIMESTAMP_DIFF(order_delivered_customer_date,order_purchase_timestamp,day)) as
total_time,
count(distinct o.order_id) as num_orders
from Business_Case_Target_SQL.customers c
left join Business_Case_Target_SQL.orders o using(customer_id)
left JOIN Business_Case_Target_SQL.order_items oi ON oi.order_id=o.order_id
group by 1
),highest_avg as
(
select customer_state as top_5_highest_avg_state,Total_time/num_orders as
high_Average_delivery_time,rank() over(order by Total_time/num_orders desc) as rnk1
from base
order by 2 DESC
limit 5
),lowest_avg as
(
select customer_state as top_5_lowest_avg_state,Total_time/num_orders as
low_Average_delivery_time, rank() over(order by Total_time/num_orders asc) as rnk2
from base
order by 2 asc
limit 5
)
select
top_5_highest_avg_state,high_Average_delivery_time,top_5_lowest_avg_state,low_Average_deliv
ery_time
from highest_avg h join lowest_avg l on h.rnk1=l.rnk2
;
```

Output :

Row	top_5_highest_avg_state	high_Average_deliver	top_5_lowest_avg_state	low_Average_deliver
1	AP	33.05882352941...	SP	9.188928280553...
2	AM	28.59459459459...	MG	12.78435754189...
3	RR	27.82608695652...	PR	12.85530227948...
4	PA	25.18974358974...	DF	13.75747663551...
5	AL	24.80629539951...	RJ	16.16837846249...

Insights :

Here Top 5 state with high average are: AP,AM,RR,PA,AL and

Top 5 state with low average are : SP,MG,PR,DF,RJ

- d. 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 :

```
SELECT
customer_state AS state,
ROUND(SUM(TIMESTAMP_DIFF(order_delivered_customer_date,
order_purchase_timestamp,
DAY))/COUNT(ORDER_ID), 2) AS average_time_for_del,
ROUND(SUM(TIMESTAMP_DIFF(order_estimated_delivery_date,
order_purchase_timestamp,
DAY))/COUNT(ORDER_ID), 2) AS average_est_dil_time,
FROM `Business_Case_Target_SQL.orders` o
INNER JOIN `Business_Case_Target_SQL.customers` c
ON o.customer_id=c.customer_id
WHERE order_status='delivered'
GROUP BY customer_state
ORDER BY average_time_for_del - average_est_dil_time;
```

Output :

Row	state	average_time_for_del	average_est_dil_time
1	AC	20.64	40.73
2	RO	18.91	38.39
3	AP	26.73	45.87
4	AM	25.99	44.92
5	RR	28.98	45.63
6	MT	17.59	31.37
7	PA	23.32	36.79
8	RS	14.82	28.16
9	RN	18.82	31.87
10	PE	17.97	30.69

Insights : AC state has faster delivery among all and these 5 state has faster than other states.

6) Analysis based on the payments:

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

Answer:

```
with cte as (
select order_id,payment_type,order_purchase_timestamp,extract(month from
order_purchase_timestamp) as month_no,format_date('%B-%Y',
order_purchase_timestamp) as month_name
from Business_Case_Target_SQL.payments p JOIN Business_Case_Target_SQL.orders o
using(order_id)
)
select month_no, month_name ,payment_type, count(order_id) as total_no_of_orders
from cte
group by 1,2,3
order by 1,2,3
;
```

Output :

JOB INFORMATION		RESULTS	CHART	PREVIEW	JSON	EXECUTION DETAILS	EXPLAIN
Row	month_no	month_name	payment_type	total_no_of_orders			
1	1	January-2017	UPI	197			
2	1	January-2017	credit_card	583			
3	1	January-2017	debit_card	9			
4	1	January-2017	voucher	61			
5	1	January-2018	UPI	1518			
6	1	January-2018	credit_card	5520			
7	1	January-2018	debit_card	109			
8	1	January-2018	voucher	416			
9	2	February-2017	UPI	398			
10	2	February-2017	credit_card	1356			
11	2	February-2017	debit_card	12			

Insights : Credit card and UPI payment are more compared to other payment type.

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

Answer:

```
select payment_installments ,count(order_id) as total_no_of_orders
from Business_Case_Target_SQL.payments
where payment_installments >=1
group by 1
order by 2 desc;
```

Output:

Business Case : Target SQL

Row	payment_installment	total_no_of_orders
1	1	52546
2	2	12413
3	3	10461
4	4	7098
5	10	5328
6	5	5239
7	8	4268
8	6	3920
9	7	1626
10	9	644
11	12	133

Insights : payment instalment 1 has highest 52546 number of orders.
