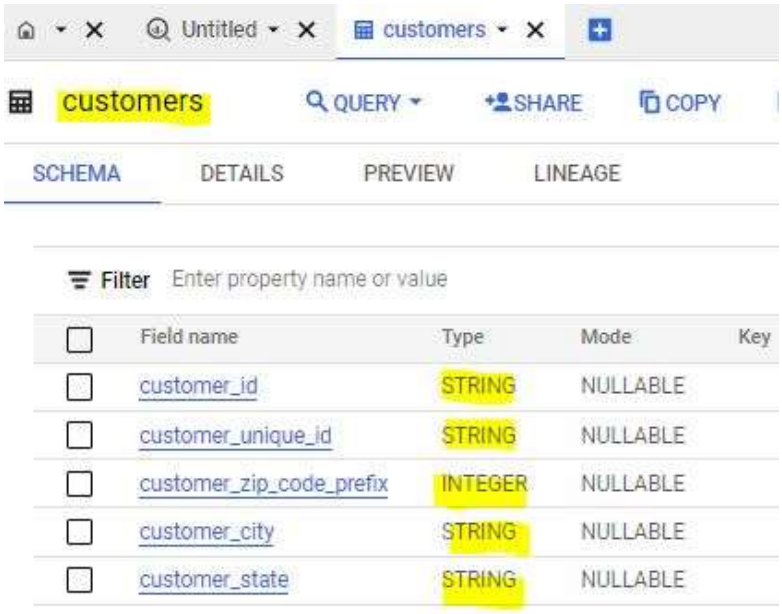
1. Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset:

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



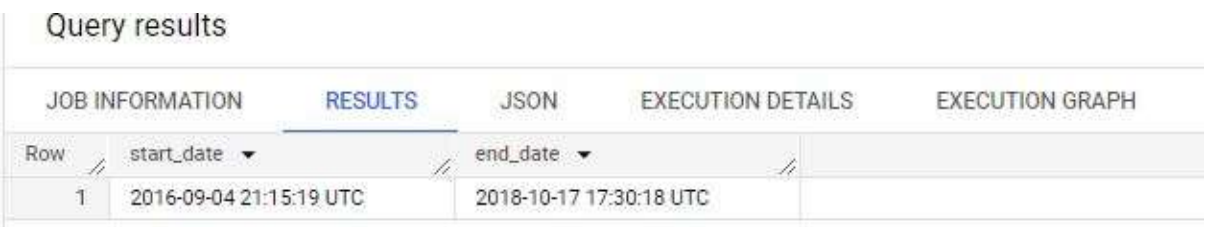
Insight:

The data type of all columns in the customers table is 'STRING', except for 'customer_zip_code_prefix', which is of the 'INTEGER' data type.

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

Query:

```
SELECT MIN(order_purchase_timestamp) AS start_date,
MAX(order_purchase_timestamp) AS end_date
FROM `Target.orders`
```



Insight:

Time period of given data is 04-09-2016 to 17-10-2018. i.e 2 Year 1 month 13 days.

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

Query:

```
SELECT COUNT(DISTINCT customer_city) AS total_cities, COUNT(DISTINCT customer_state) AS
total_states
FROM `Target.customers`
WHERE customer_id IN (
    SELECT customer_id
    FROM `Target.orders`
    WHERE order_purchase_timestamp >= '2016-09-04 21:15:19 UTC'
    AND order_purchase_timestamp <= '2018-10-17 17:30:18 UTC'
);
```

Row	total_cities	total_states
1	4119	27

Insight:

As per the data, during the given time period the total count of cities and states are 4119 and 27 respectively.

2. In-depth Exploration

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

Query:

```
SELECT EXTRACT(YEAR FROM order_purchase_timestamp) AS order_year, COUNT(*) AS total_orders
FROM `Target.orders`
GROUP BY order_year
ORDER BY order_year ASC;
```

Row	order_year	total_orders
1	2016	329
2	2017	45101
3	2018	54011

Insight:

It is clear that there is a growing trend in the no. of orders placed over the past two years.

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

Query:

```
SELECT
EXTRACT(MONTH FROM order_purchase_timestamp) AS month,
count(order_id) AS Number_of_orders
FROM `Target.orders`
GROUP BY month
ORDER BY Number_of_orders DESC
```

Query results

JOB INFORMATION | RESULTS | JSON

Row	month	Number_of_orders
1	8	10843
2	5	10573
3	7	10318
4	3	9893
5	6	9412
6	4	9343
7	2	8508
8	1	8069
9	11	7544
10	12	5674
11	10	4959
12	9	4305

Insight:

During the months of August, May, July, and March, the number of orders is high, while during the months of December, October, and September, the number of orders is low.

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

Query:

```
SELECT DISTINCT
time_of_the_day,
count(time_of_the_day) OVER(PARTITION BY
time_of_the_day) as count_time_of_day
FROM
(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 "Morning"
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_of_the_day
FROM `Target.orders`)
ORDER BY count_time_of_day DESC;
```

Query results

JOB INFORMATION | RESULTS | JSON | EXEC

Row	time_of_the_day	count_time_of_day
1	Afternoon	38135
2	Night	28331
3	Morning	27733
4	Dawn	5242

Insight:

It can be clearly observed that during Afternoon Brazilian customers tend to buy more compared to any other time of the day.

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

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

Query:

```
SELECT DISTINCT
cus.customer_state,
EXTRACT(MONTH FROM ord.order_purchase_timestamp) AS
month,
count(ord.order_id) AS Number_of_orders
FROM `Target.customers` AS cus
```

```
JOIN `Target.orders` AS ord
ON cus.customer_id=ord.customer_id
GROUP BY customer_state, month
ORDER BY customer_state, month
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	customer_state	month	Number_of_orders
1	SP	8	4982
2	SP	5	4632
3	SP	7	4381
4	SP	6	4104
5	SP	3	4047
6	SP	4	3967
7	SP	2	3357
8	SP	1	3351
9	SP	11	3012
10	SP	12	2357
11	SP	10	1908
12	SP	9	1648

Insight:

From the above result, we can analyse the monthly sales for each state. It is evident that the state 'SP' has the highest number of orders compared to all other states. Additionally, the month of August (8th month) exhibits the highest sales.

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

Query:

```
SELECT DISTINCT
customer_state,
count(customer_id) AS Number_of_people
FROM `Target.customers`
GROUP BY customer_state
ORDER BY Number_of_people DESC
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION

Row	customer_state	Number_of_people
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
12	CE	1336

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DET

Row	customer_state	Number_of_people
16	MS	715
17	PB	536
18	PI	495
19	RN	485
20	AL	413
21	SE	350
22	TO	280
23	RO	253
24	AM	148
25	AC	81
26	AP	68
27	RR	46

Insight:

Upon examining the results, it becomes clear that there is a distinct customer distribution across the various states of Brazil. Specifically, the state of 'SP' stands out with the highest number of customers, totalling 41,746 individuals, highlighting a substantial customer base in that region. In contrast, the state of 'RR' has the lowest number of customers, with only 46 individuals recorded in the dataset.

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.

Query:

```
SELECT DISTINCT
Year,
ROUND(((SUM(pay.payment_value)) -
(LAG(SUM(pay.payment_value)) OVER(ORDER BY
year)))/(LAG(SUM(pay.payment_value)) OVER(ORDER BY
year))*100,2) AS percentage_cost_increase
FROM
(SELECT
order_id,
EXTRACT (YEAR FROM order_purchase_timestamp) AS year,
EXTRACT (MONTH FROM order_purchase_timestamp) AS month
FROM `Target.orders`) AS ord
JOIN `Target.payments` As pay
ON ord.order_id=pay.order_id
WHERE ord.year BETWEEN 2017 AND 2018 AND ord.month
BETWEEN 1 AND 8
GROUP BY ord.year
ORDER BY ord.year
```

Query results

JOB INFORMATION | RESULTS | JSON

Row	Year	percentage_cost_inc
1	2017	null
2	2018	136.98

Insight:

The cost of orders experienced a substantial surge of 136.98% from 2017 to 2018, specifically during the months spanning January to August.

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

Query:

```
SELECT cus.customer_state,
       ROUND(AVG(ordi.price), 2) AS avg_price,
       ROUND(SUM(ordi.price), 2) AS total_price
FROM `Target.customers` AS cus
JOIN `Target.orders` AS ord ON cus.customer_id = ord.customer_id
```

```
JOIN `Target.order_items` AS ordi ON ord.order_id = ordi.order_id
GROUP BY cus.customer_state
ORDER BY cus.customer_state asc;
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	customer_state	avg_price	total_price
1	AC	173.73	15982.95
2	AL	180.89	80314.81
3	AM	135.5	22356.84
4	AP	164.32	13474.3
5	BA	134.6	511349.99
6	CE	153.76	227254.71
7	DF	125.77	302603.94
8	ES	121.91	275037.31
9	GO	126.27	294591.95
10	MA	145.2	119648.22
11	MG	120.75	1585308.03
12	MS	142.63	116812.64

Insight:

The average and total prices for each state are presented in the table above, arranged in ascending order by state.

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

Query:

```
SELECT DISTINCT
cus.customer_state,
ROUND(AVG (ordi.price) OVER(PARTITION BY
cus.customer_state),2) as mean_price,
ROUND(AVG (ordi.freight_value) OVER(PARTITION BY
cus.customer_state),2) as mean_freight_value,
ROUND(SUM (ordi.price) OVER(PARTITION BY
cus.customer_state),2) as price_sum,
ROUND(SUM (ordi.freight_value) OVER(PARTITION BY
cus.customer_state),2) as freight_value_sum
FROM `Target.customers` AS cus
JOIN `Target.orders` AS ord
ON cus.customer_id =ord.customer_id
```

```
JOIN `Target.order_items` AS ordi
ON ord.order_id = ordi.order_id
ORDER BY cus.customer_state
```

Query results

SAVE RESULTS

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH

Row	customer_state	mean_price	mean_freight_value	price_sum	freight_value_sum
1	AC	173.73	40.07	15982.95	3686.75
2	AL	180.89	35.84	80314.81	15914.59
3	AM	135.5	33.21	22356.84	5478.89
4	AP	164.32	34.01	13474.3	2788.5
5	BA	134.6	26.36	511349.99	100156.68
6	CE	153.76	32.71	227254.71	48351.59
7	DF	125.77	21.04	302603.94	50625.5
8	ES	121.91	22.06	275037.31	49764.6
9	GO	126.27	22.77	294591.95	53114.98
10	MA	145.2	38.26	119648.22	31523.77
11	MG	120.75	20.63	1585308.03	270853.46
12	MS	142.63	23.37	116812.64	19144.03

Insight:

The Sum and Average of price and freight value is arranged state wise in the above table.

5. Analysis based on sales, freight and delivery time

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

Query:

```
SELECT
  order_id,
  TIMESTAMP_DIFF(DATE(order_delivered_customer_date), DATE(order_purchase_timestamp), DAY)
AS time_to_delivery,
  TIMESTAMP_DIFF(DATE(order_estimated_delivery_date), DATE(order_delivered_customer_date),
DAY) AS diff_estimated_delivery
FROM
  `Target.orders`
ORDER BY
  order_id;
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION G

Row	order_id	time_to_delivery	diff_estimated_deliv
1	00010242fe8c5a6d1ba2dd792...	7	9
2	00018f77f2f0320c557190d7a1...	16	3
3	000229ec398224ef6ca0657da...	8	14
4	00024acbcdf0a6daa1e931b03...	6	6
5	00042b26cf59d7ce69dfabb4e...	25	16
6	00048cc3ae777c65dbb7d2a06...	7	15
7	00054e8431b9d7675808bcb8...	8	17
8	000576fe39319847cbb9d288c...	5	16
9	0005a1a1728c9d785b8e2b08...	10	0
10	0005f50442cb953dcd1d21e1f...	2	19
11	00061f2a7bc09da83e415a52d...	5	11
12	00063b381e2406b52ad42947...	11	0

Insight:

The `time_to_delivery` and `diff_estimated_delivery` columns for each order provide the time taken for actual delivery and the time difference between the estimated and actual delivery in terms of hours. The negative sign in the `diff_estimated_delivery` attribute indicates that the order was delivered earlier than the estimated delivery time.

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

Query:

Highest average freight value

```
SELECT
  cus.customer_state,
  ROUND(AVG(ordi.freight_value), 3) AS Highest_avg_freight_value
FROM
  `Target.customers` AS cus
JOIN
  `Target.orders` AS ord ON cus.customer_id = ord.customer_id
JOIN
  `Target.order_items` AS ordi ON ord.order_id = ordi.order_id
GROUP BY
  cus.customer_state
ORDER BY
  Highest_avg_freight_value DESC
LIMIT 5;
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	customer_state	Highest_avg_freight_value
1	RR	42.984
2	PB	42.724
3	RO	41.07
4	AC	40.073
5	PI	39.148

Insight:

States RR, PB, RO, AC, PI are the top 5 States that have the highest average freight values.

Lowest average freight value

```
SELECT
  cus.customer_state,
  ROUND(AVG(ordi.freight_value), 3) AS Lowest_avg_freight_value
FROM
  `Target.customers` AS cus
JOIN
  `Target.orders` AS ord ON cus.customer_id = ord.customer_id
```

```
JOIN
  `Target.order_items` AS ordi ON ord.order_id = ordi.order_id
GROUP BY
  cus.customer_state
ORDER BY
  Lowest_avg_freight_value ASC
LIMIT 5;
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	customer_state	Lowest_avg_freight_value
1	SP	15.147
2	PR	20.532
3	MG	20.63
4	RJ	20.961
5	DF	21.041

Insight:

States SP, PR, MG, RJ, DF are the top 5 States that have the lowest average freight values.

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

Query:

Highest average delivery time

```
SELECT
  cus.customer_state,
  ROUND(AVG(ordi.freight_value), 3) AS avg_freight_value,
  ROUND(AVG(TIMESTAMP_DIFF(ord.order_delivered_customer_date, ord.order_purchase_timestamp,
HOUR)), 3) AS avg_time_to_delivery,
  ROUND(AVG(TIMESTAMP_DIFF(ord.order_estimated_delivery_date,
ord.order_delivered_customer_date, HOUR)), 3) AS avg_diff_estimated_delivery
FROM
  `Target.customers` AS cus
JOIN
  `Target.orders` AS ord ON cus.customer_id = ord.customer_id
JOIN
  `Target.order_items` AS ordi ON ord.order_id = ordi.order_id
GROUP BY
  cus.customer_state
ORDER BY
  avg_time_to_delivery DESC
```

```
LIMIT 5;
```

Query results

SAVE RES

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH

Row	customer_state	avg_freight_value	avg_time_to_delivery	avg_diff_estimated_delivery
1	RR	42.984	676.978	422.435
2	AP	34.006	676.457	426.012
3	AM	33.205	632.847	460.969
4	AL	35.844	587.227	193.133
5	PA	35.833	569.601	325.289

Insight:

States RR, AP, AM, AL, PA are the top 5 States that have the highest average time to delivery.

Lowest average delivery time

```
SELECT
  cus.customer_state,
  ROUND(AVG(ordi.freight_value), 3) AS avg_freight_value,
  ROUND(AVG(TIMESTAMP_DIFF(ord.order_delivered_customer_date, ord.order_purchase_timestamp,
HOUR)), 3) AS avg_time_to_delivery,
  ROUND(AVG(TIMESTAMP_DIFF(ord.order_estimated_delivery_date,
ord.order_delivered_customer_date, HOUR)), 3) AS avg_diff_estimated_delivery
FROM
  `Target.customers` AS cus
JOIN
  `Target.orders` AS ord ON cus.customer_id = ord.customer_id
JOIN
  `Target.order_items` AS ordi ON ord.order_id = ordi.order_id
GROUP BY
  cus.customer_state
ORDER BY
  avg_time_to_delivery ASC
LIMIT 5;
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH

Row	customer_state	avg_freight_value	avg_time_to_delivery	avg_diff_estimated_d
1	SP	15.147	208.869	251.894
2	PR	20.532	286.24	306.574
3	MG	20.63	287.112	302.913
4	DF	21.041	310.518	275.42
5	SC	21.47	359.526	260.55

Insight:

States SP, PR, MG, DF, SC are the top 5 States that have the lowest average time to delivery.

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.

Query:

```
SELECT
  cus.customer_state,
  ROUND(AVG(ordi.freight_value), 3) AS avg_freight_value,
  ROUND(AVG(TIMESTAMP_DIFF(ord.order_delivered_customer_date, ord.order_purchase_timestamp,
HOUR)), 3) AS avg_actual_delivery_time,
  ROUND(AVG(TIMESTAMP_DIFF(ord.order_estimated_delivery_date, ord.order_purchase_timestamp,
HOUR)), 3) AS avg_estimated_delivery_time,
  ROUND(AVG(TIMESTAMP_DIFF(ord.order_delivered_customer_date,
ord.order_estimated_delivery_date, HOUR)), 3) AS avg_delivery_time_difference
FROM
  `Target.customers` AS cus
JOIN
  `Target.orders` AS ord ON cus.customer_id = ord.customer_id
JOIN
  `Target.order_items` AS ordi ON ord.order_id = ordi.order_id
GROUP BY
  cus.customer_state
HAVING avg_delivery_time_difference < 0
ORDER BY
  avg_delivery_time_difference ASC
LIMIT 5;
```

Query results

SAVE RESULTS | EXPLORE D

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH

Row	customer_state	avg_freight_value	avg_actual_delivery_time	avg_estimated_delivery_time	avg_delivery_time_difference
1	AC	40.073	496.67	985.543	-487.593
2	RO	41.07	473.271	936.439	-463.769
3	AM	33.205	632.847	1092.758	-460.969
4	AP	34.006	676.457	1099.78	-426.012
5	RR	42.984	676.978	1111.192	-422.435

Note: average_time_to_delivery and average_diff_estimated_delivery has been represented in terms of hours.

Insight:

States AC, RO, AM, AP, RR are the top 5 States that have the fastest delivery.

6. Analysis based on the payments:

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

Query:

```
SELECT DISTINCT
pay.payment_type,
EXTRACT (MONTH FROM ord.order_purchase_timestamp) AS
month,
count(ord.order_id) AS order_count
FROM `Target.orders` as ord
JOIN `Target.payments` as pay
ON ord.order_id = pay.order_id
GROUP BY month, pay.payment_type
ORDER BY month,pay.payment_type;
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	payment_type	month	order_count
1	UPI	1	1715
2	credit_card	1	6103
3	debit_card	1	118
4	voucher	1	477
5	UPI	2	1723
6	credit_card	2	6609
7	debit_card	2	82
8	voucher	2	424
9	UPI	3	1942
10	credit_card	3	7707
11	debit_card	3	109
12	voucher	3	591

Insight:

The month wise count of different payment methods are shown in the above result.

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

Query:

```
SELECT DISTINCT
pay.payment_installments,
count(ord.order_id) AS order_count
FROM `Target.orders` as ord
JOIN `Target.payments` as pay
ON ord.order_id = pay.order_id
GROUP BY pay.payment_installments
ORDER BY pay.payment_installments
```

Query results

JOB INFORMATION | RESULTS | JSON | EXI

Row	payment_installments	order_count
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
11	10	5328
12	11	23

Insight:

From the results, it can be observed that the majority of items were purchased with full payment. However, there is a significant number of items that were purchased with 2 to 4 installments. Additionally, there are a few items that were bought with the maximum installment count of 24.

Summary of Insights

From the above findings, we can come to the conclusion that:

1. The dataset covers the time period from 04-09-2016 to 17-10-2018, with attributes of different data types including String, Integer, Float, and Timestamp.

2. Various cities and States from which the customers have placed orders could also be determined.

3. It is evident that Brazilian customers make more purchases during the afternoon compared to other times of the day. Additionally, the months of August, May, July, and March show a higher number of orders, while December, October, and September exhibit a lower number of orders.

4. The rise and fall of sales for each State month wise can be observed.

5. Similarly, the distribution of customers across different states of Brazil can be identified.

6. A massive increase by 136.98% in the cost of orders can be observed when comparing from 2017 to 2018.

7. Also, State wise Sum and Average of price and freight value tabulated is seen.

8. The fastness of delivery and the gap between estimated delivery and actual delivery time can be observed from the table showing the difference between purchase, delivery and estimated delivery days.

9. From the data collected, it can be ascertained that in some cases the products were delivered sooner than the estimated delivery date.

10. With respect to different States the average of freight value, delivery time, and the difference between estimated and actual delivery is analysed.

11. Top five States with the highest and lowest mean freight value, and mean delivery time are evaluated. Similarly top five States with the fastest and slowest delivery are also evaluated.

12. Significant changes in the usage of various methods of payment per month can be observed.

13. It is observed that the mode of payment exhibits a consistent pattern across all months.

14. Number of orders based on the Payment Installments can be observed in the table generated.

Recommendations

1. As it is ascertained that in certain months the sales are higher, we need to make sure that proper sales and marketing techniques must be adopted so as to ensure a constant stream of sales in all the months, thereby increasing the profit.

2. There is a massive difference in distribution of customers when it comes to different states. Quality advertisements and customer satisfaction packages must be introduced in the states with low amount of customers. By creating customer oriented management, one can attract huge amount of customers.

3. Significant increase in the cost of orders can be used as an advantage by the companies to invest in areas that require more managerial attention.

4. The organization must ensure quick delivery of products and must also develop a quality interface so that there is not much difference between the estimated and actual delivery time. Doing this will ensure customer loyalty as well as helps in attracting new customers.

5. States with the highest mean freight value and delivery time warrant careful attention for reviewing and redesigning organizational policies to enhance efficiency. This proactive approach can help lagging states catch up while enabling high-performing states to further improve their performance.

6. In the digital age, staying ahead of technological advancements is crucial for any organization. As evidenced by the diverse range of payment methods being used, prioritizing seamless and accessible internet banking experiences can enhance

customer retention and drive sales. Embracing hassle-free transaction options is the key to success in the evolving marketplace.

By implementing these recommendations and demonstrating attentiveness to consumers' needs while leveraging organizational capabilities, businesses can achieve optimal levels of customer satisfaction and profitability.