ABOUT TARGET :-

Target is a globally renowned prominent retailer in United Sates, which offers outstanding value, inspiration, innovative experience to customers.

This Business case mainly focuses on operation in Brazil between 2016 and 2018.

Motive :-

To provide various information and insights of business such as pricing strategies, payment, shipping efficiency, Product characteristics

Analysis -1

Usual exploratory analysis

What does good look like?

1.a) What are datatypes of all columns in customers table?

```
1  SELECT column_name, data_type
2  FROM `spheric-keel-415014.Target_2016_2019.INFORMATION_SCHEMA.COLUMNS`
3  where table_name ='customer'
```

Output:-

Query results

JOB INFORMATION | RESULTS | CHART | JSON | EXECUTION D

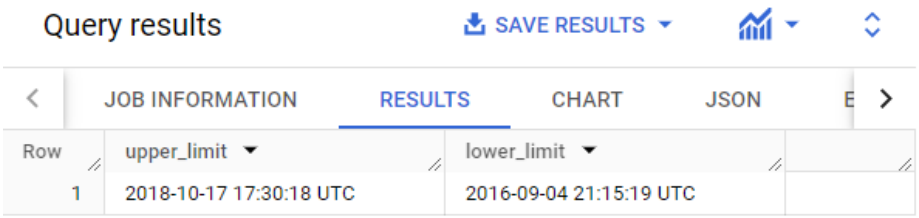
Row	column_name	data_type
1	customer_id	STRING
2	customer_unique_id	STRING
3	customer_zip_code_prefix	INT64
4	customer_city	STRING
5	customer_state	STRING

Majorly data_types are strings

1.b)Get the time range between which the orders are placed?

```
SELECT max(order_purchase_timestamp) as `upper_limit`, min
(order_purchase_timestamp) as `lower_limit`FROM
`spheric-keel-415014.Target_2016_2019.orders`
```

Output

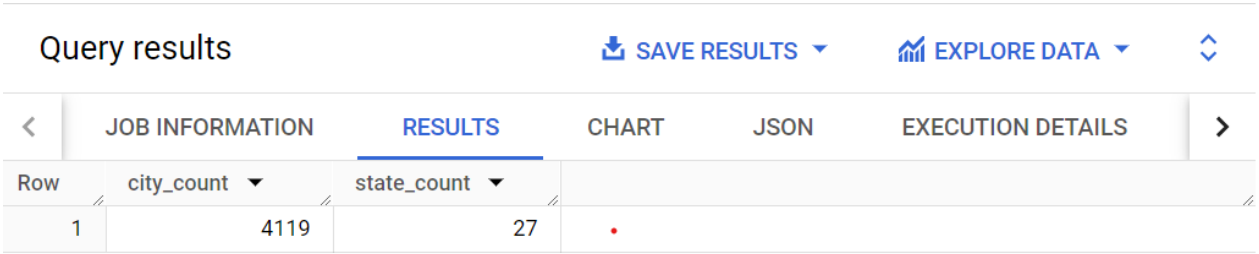


1.c) count the cities and states who ordered during given time period?

```
select count(distinct customer_city) as city_count,count(distinct customer_state) as
state_count from spheric-keel-415014.Target_2016_2019.customer c
join spheric-keel-415014.Target_2016_2019.orders o
on o.customer_id=c.customer_id
where extract(year from order_purchase_timestamp) in(2016,2017,2018)

```

Output

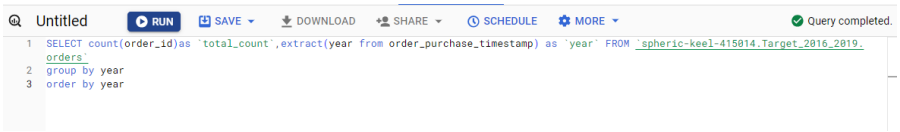


No.of cities -4119

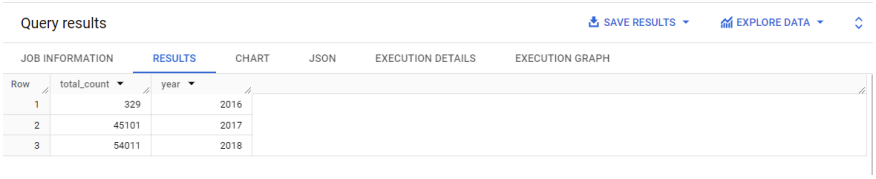
No.of state-27

<u>In-depth exploration</u>:

2.a) Is there a growing trend in no.of orders over past years?



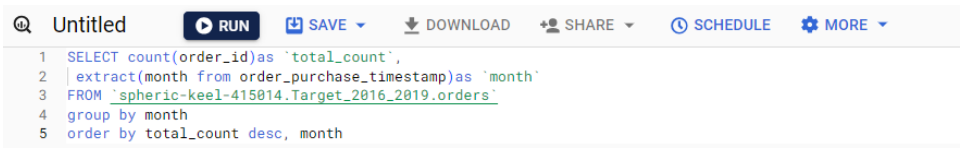
Output



Row	total_count	year
1	329	2016
2	45101	2017
3	54011	2018

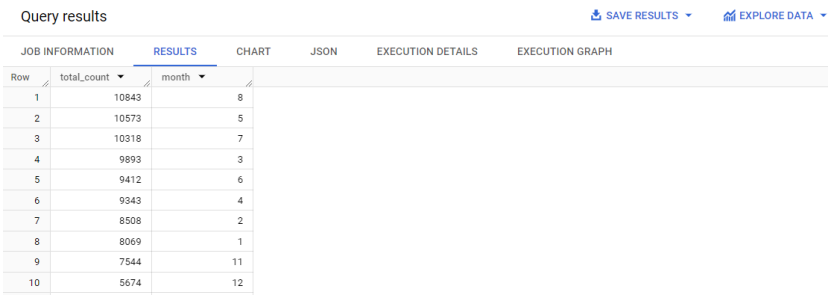
Insights:- Yes, there is a growing trend of orders over past years.

2.b) Can we see monthly seasonality in terms of no.of orders being placed?



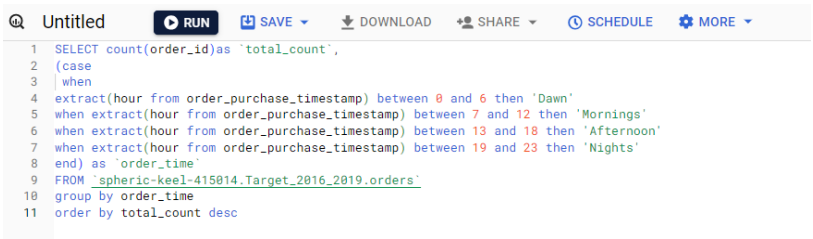

```sql
SELECT count(order_id)as `total_count`,
 extract(month from order_purchase_timestamp)as `month`
FROM `spheric-keel-415014.Target_2016_2019.orders`
group by month
order by total_count desc, month
```

Output:-



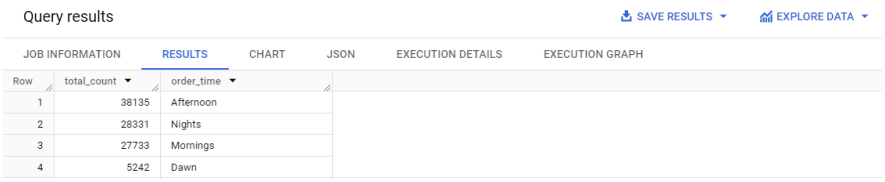
Row	total_count	month
1	10843	8
2	10573	5
3	10318	7
4	9893	3
5	9412	6
6	9343	4
7	8508	2
8	8069	1
9	7544	11
10	5674	12

2.c) During what time of day, do Brazilian customer mostly place their order?(Dawn, Mornings, Afternoon, Nights)




```sql
SELECT count(order_id)as `total_count`,
(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 'Nights'
end) as `order_time`
FROM `spheric-keel-415014.Target_2016_2019.orders`
group by order_time
order by total_count desc
```

Output



Row	total_count	order_time
1	38135	Afternoon
2	28331	Nights
3	27733	Mornings
4	5242	Dawn

Insights:- During afternoon most of Brazilians placed their orders.

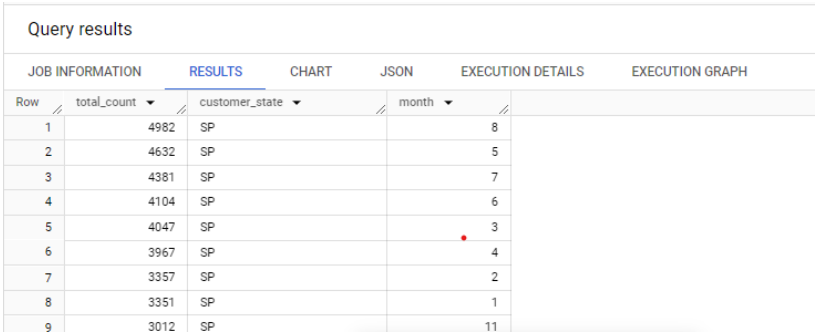
Recommendation:- Giving popup messages of discounts or flat sales will help to place more orders.

Evolution of E-commerce orders in the Brazil region

3.a)Get month on month no.of orders placed in each state?

```
select count(o.order_id) as `total_count`, customer_state, extract(month from order_purchase_timestamp) as `month`
from spheric-keel-415014.Target_2016_2019.customer c
join spheric-keel-415014.Target_2016_2019.orders o
on o.customer_id=c.customer_id
group by customer_state, month
order by total_count desc
```

Output:-



Query results

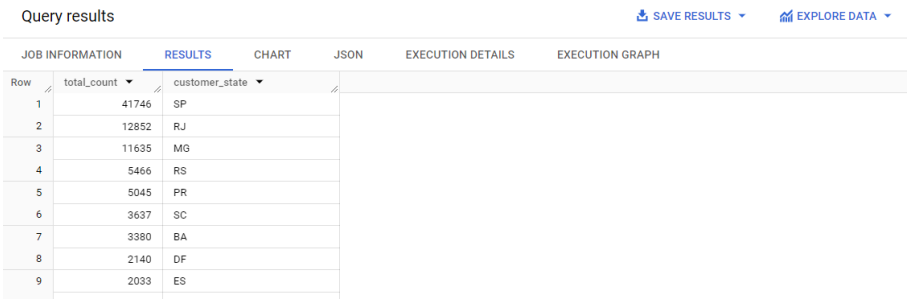
JOB INFORMATION | RESULTS | CHART | JSON | EXECUTION DETAILS | EXECUTION GRAPH

Row	total_count	customer_state	month
1	4982	SP	8
2	4632	SP	5
3	4381	SP	7
4	4104	SP	6
5	4047	SP	3
6	3967	SP	4
7	3357	SP	2
8	3351	SP	1
9	3012	SP	11

3.b) How are customers distributed across all states?

```
select count(*) as `total_count`, customer_state from spheric-keel-415014.Target_2016_2019.customer
group by customer_state
order by `total_count` desc
```

Output:-



Query results

SAVE RESULTS | EXPLORE DATA

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

Row	total_count	customer_state
1	41746	SP
2	12852	RJ
3	11635	MG
4	5466	RS
5	5045	PR
6	3637	SC
7	3380	BA
8	2140	DF
9	2033	ES

4.a) Get the % increase in the cost of orders from year 2017 to 2018(include months between jan to aug only) Use payment value column to get cost of orders.

```
SELECT (SUM(CASE WHEN EXTRACT(YEAR FROM O.ORDER_PURCHASE_TIMESTAMP) =2018 THEN P.PAYMENT_VALUE ELSE 0 END)) AS `2018_SUM`,
(SUM(CASE WHEN EXTRACT(YEAR FROM O.ORDER_PURCHASE_TIMESTAMP)=2017 THEN P.PAYMENT_VALUE ELSE 0 END))  AS `2017_SUM`,

((SUM(CASE WHEN EXTRACT(YEAR FROM O.ORDER_PURCHASE_TIMESTAMP) =2018 THEN P.PAYMENT_VALUE ELSE 0 END)
-SUM(CASE WHEN EXTRACT(YEAR FROM O.ORDER_PURCHASE_TIMESTAMP)=2017 THEN P.PAYMENT_VALUE ELSE 0 END))
/SUM(CASE WHEN EXTRACT(YEAR FROM O.ORDER_PURCHASE_TIMESTAMP)=2017 THEN P.PAYMENT_VALUE ELSE 0 END)) * 100 AS PERCENTAGE_INCREASE
FROM spheric-keel-415014.Target_2016_2019.orders O
JOIN spheric-keel-415014.Target_2016_2019.payments 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
```

OUTPUT

Query results

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

Row	2018_SUM	2017_SUM	PERCENTAGE_INCRE
1	8694733.839999...	3669022.119999...	136.9768716466...

Insights:- Yes there is 136.9% increase in cost of orders from 2017 to 2018

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

```
select c.customer_state, round(sum(oi.price),2) as `total_price`, round(avg(oi.
price),2) as `avg_price`
from spheric-keel-415014.Target_2016_2019.customer c
join spheric-keel-415014.Target_2016_2019.orders o
on o.customer_id=c.customer_id
join spheric-keel-415014.Target_2016_2019.order_items oi
on oi.order_id=o.order_id
group by c.customer_state
order by c.customer_state
```

output

Query results SAVE RESULTS EXPLORE DATA

JOB INFORMATION | RESULTS | CHART | JSON | EXECUTION DETAILS

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

4.c) Calculate the Total & Average value of freight value for each state

```
select c.customer_state, round(sum(oi.freight_value),2) as `total_price`, round(avg
(oi.freight_value),2) as `avg_price`
from spheric-keel-415014.Target_2016_2019.customer c
join spheric-keel-415014.Target_2016_2019.orders o
on o.customer_id=c.customer_id
join spheric-keel-415014.Target_2016_2019.order_items oi
on oi.order_id=o.order_id
group by c.customer_state
order by total_price, avg_price
```

Output

Query results SAVE RESULTS EXPLORE DATA

JOB INFORMATION | RESULTS | CHART | JSON | EXECUTION DETAILS

Row	customer_state	total_price	avg_price
1	RR	2235.19	42.98
2	AP	2788.5	34.01
3	AC	3686.75	40.07
4	AM	5478.89	33.21
5	RO	11417.38	41.07
6	TO	11732.68	37.25
7	SE	14111.47	36.65
8	AL	15914.59	35.84
9	RN	18860.1	35.65
10	MS	19144.03	23.37

5.a) Find Number of days taken to deliver each order from the orders purchase date as delivery_time,

Also calculate diff in dates between estimated and actual delivery time

```
select distinct 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 `Target_2016_2019.orders`
```

Output:-

Query results

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

Row	order_id	time_to_deliver	diff_estimated_deliv
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
11	66057d37308e787052a32828...	38	6
12	19135c945c554eebfd7576c73...	36	2

5.b) Find Out Top 5 states with the highest and lowest average freight value.

Top5 states with lowest and highest average freight value

```
(select c.customer_state, round(avg(oi.freight_value),2)
as `avg_price`
from spheric-keel-415014.Target_2016_2019.customer c
join spheric-keel-415014.Target_2016_2019.orders o
on o.customer_id=c.customer_id
join spheric-keel-415014.Target_2016_2019.order_items oi
on oi.order_id=o.order_id
group by c.customer_state
order by avg_price
limit 5
)

union all


(select c.customer_state, round(avg(oi.freight_value),2)
as `avg_price`
from spheric-keel-415014.Target_2016_2019.customer c
join spheric-keel-415014.Target_2016_2019.orders o
on o.customer_id=c.customer_id
join spheric-keel-415014.Target_2016_2019.order_items oi
on oi.order_id=o.order_id
group by c.customer_state
order by avg_price desc
limit 5
```

Output

Query results

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

Row	customer_state	avg_price
1	SP	15.15
2	PR	20.53
3	MG	20.63
4	RJ	20.96
5	DF	21.04
6	RR	42.98
7	PB	42.72
8	RO	41.07
9	AC	40.07
10	PI	39.15

5.c)Top 5 states with highest and lowest average delivery time

```
(select c.customer_state, avg(date_diff(o.order_delivered_customer_date,o.order_purchase_timestamp,day))
as `avg_time`
from spheric-keel-415014.Target_2016_2019.customer c
join spheric-keel-415014.Target_2016_2019.orders o
on o.customer_id=c.customer_id
group by c.customer_state
order by avg_time
limit 5
)

union all


(select c.customer_state, avg(date_diff(o.order_delivered_customer_date,o.order_purchase_timestamp,day))
as `avg_time`
from spheric-keel-415014.Target_2016_2019.customer c
join spheric-keel-415014.Target_2016_2019.orders o
on o.customer_id=c.customer_id
group by c.customer_state
order by avg_time desc
limit 5
)
```

Output:-

Query results

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

Row	customer_state	avg_time
1	RR	28.97560975609...
2	AP	26.73134328358...
3	AM	25.98620689655...
4	AL	24.04030226700...
5	PA	23.31606765327...
6	SP	8.298061489072...
7	PR	11.52671135486...
8	MG	11.54381329810...
9	DF	12.50913461538...
10	SC	14.47956019171...

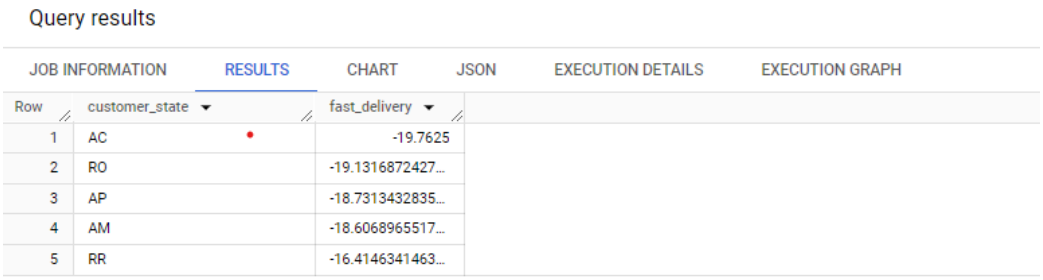
First five states (RR, AP,AM,AL,PA)have highest average delivery time

Last 5 states(SP,PR,MG,DF,SC) have lowest average delivery time

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

```
select c.customer_state,
avg(date_diff(order_delivered_customer_date,order_estimated_delivery_date,day)) as fast_delivery
from spheric-keel-415014.Target_2016_2019.customer c
join spheric-keel-415014.Target_2016_2019.orders o
on o.customer_id=c.customer_id
group by c.customer_state
order by fast_delivery asc
limit 5
```

Output



Row	customer_state	fast_delivery
1	AC	-19.7625
2	RO	-19.1316872427...
3	AP	-18.7313432835...
4	AM	-18.6068965517...
5	RR	-16.4146341463...

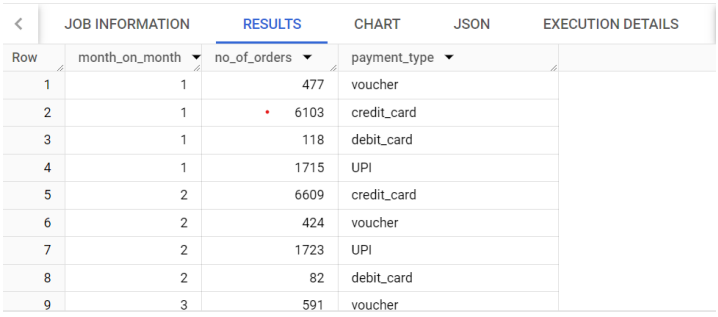
Here (-) represents order delivered earlier than expected days.

These are top 5 (AC,RO,AP,AM,RR) states the delivery of products happened fast than estimated time.

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

```
select extract(month FROM o.order_purchase_timestamp) as `month_on_month`,count(o.
order_id) as `no_of_orders`,p.payment_type
from spheric-keel-415014.Target_2016_2019.orders o
join spheric-keel-415014.Target_2016_2019.payments p on
p.order_id = o.order_id
group by month_on_month,payment_type
order by month_on_month
```

Output

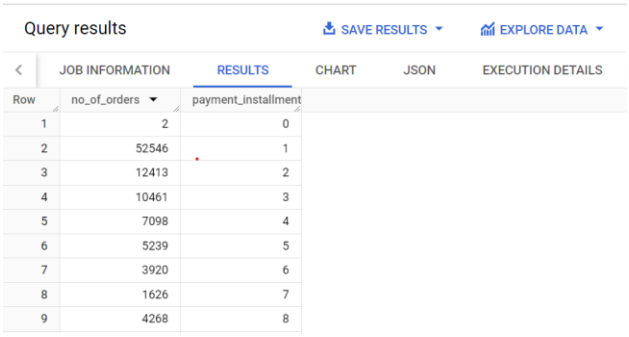


Row	month_on_month	no_of_orders	payment_type
1	1	477	voucher
2	1	6103	credit_card
3	1	118	debit_card
4	1	1715	UPI
5	2	6609	credit_card
6	2	424	voucher
7	2	1723	UPI
8	2	82	debit_card
9	3	591	voucher

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

```
select count(distinct order_id) as no_of_orders, payment_installments from spheric-keel-415014.Target_2016_2019.payments
group by payment_installments
```

Output



Row	no_of_orders	payment_installment
1	2	0
2	52546	1
3	12413	2
4	10461	3
5	7098	4
6	5239	5
7	3920	6
8	1626	7
9	4268	8

Overall Insights: -

We are doing our analysis on brazil operation, from year 2016 to 2018

Where

1) We have overall 27 states and 4119 cities in Brazil
2) No.of orders have been increasing from year to year 2016 have less orders compared to 2017, and 2018(2016<<2017<<2018)
3) In month of may and august more no of orders have been placed.
4) Most of the orders have been placed during afternoon time.
5) Most of the customers were from state **SP**
6) Just considering from months (Jan to Aug) from 2017 to 2018, we have 136.9% increase in cost of orders.
7) State RR (42.98)has highest freight and average delivery time value, SP (15.15)has lowest freight and average delivery time value
8) Most of the customer were using credit card as payment option
9) On analyzing most of them paid only one installment by the time data is driven.

Recommendations:-

1) Keeping more inventory/stocks in warehouse in months of may and august may help company to deliver the products as soon as possible to customers, which improves customer relationship with company.
2) Delivering the products on time to low average and freight states might help us in increasing the sales in those particular states.
3) Giving offers on certain set of credit cards might help us in improving then sales.

4) Giving huge discount (clearance sales) will definitely increase the no.of orders and cost of orders
5) As we observe most of the orders were placed during afternoon. Giving offers on UPI or credit card payments, or, 1+1 offers in other slots(morning, night) helps us in increasing the sales.