Data Analysis

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

1. Data type of columns in a table

Customers Table Insights: We see customer id is a string or varchar which indicates that it is a combination of letters and numbers. Also, there is another customer unique id which may be generated when the customer made a purchase or it may be their name which is kept private.

Geolocation Table: Geolocation zip code prefix column is common in customers and geolocation which is integer data type. Contains latitude, longitude in float format. City and State in string format.

Order Items: There is order id a string data type which is a unique id for each order placed and order item id an integer data type given to each item within an order. Product id and seller id which are string data type, shipping limit date is timestamp or date time data type and price, fright are float data type.

Order Reviews: There's a review id a string data type which is unique and generated for each review provided, order id for which order the review is provided, review score an int data type, review creation date a time stamp which seems to be in a wrong format (0001-04-17 00:00:00 UTC), review answer timestamp which is also in the wrong format (0001-04-17 00:00:00 UTC).

Orders : contains order id, customer id, order status in string format. Order purchase, order approved at, delivered to carrier, delivered to customer and estimated delivery timestamp data type columns.

Payments: Order id, payment type are string data type, payment sequential and payment instalments are integers and payment value is a float.

Products: There are few integer data types which corresponds to product description like length, width, height in cm, weight in kg, number of photos of the product in the website and string data types like product id, product category etc

Sellers: There is seller id a string column, seller zip code an integer, seller city and state string data type.

2. Time period for which the data is given
   Query:

```
SELECT extract(date from Min(order_purchase_timestamp)) as first_order_date,
extract(date from max(order_purchase_timestamp)) most_recent_order_date ,
count(order_id)  total_number_orders
FROM `target-dataset123.target_market.orders`
```

   Result:

Row	first_order_date	most_recent_order_date	total_number_orders
1	2016-09-04	2018-10-17	99441

   Insights: From the above result it's known that the given data is between 2016 and 2018 with a total close to 100k orders from Brazil during this period.

3. Cities and States of customers ordered during the given period
   Query:

```
select count(distinct customer_state) total_number_of_states from `target-
dataset123.target_market.customers`
```

   Result:

Row	total_number_of_states
1	27

   Insights: Orders came from all the 27 different states in Brazil.

   Query:

```
SELECT distinct customer_state,count(distinct customer_city) no_of_cities
FROM `target-dataset123.target_market.customers`
group by customer_state
order by no_of_cities desc
```

   Result:

Row	customer_state	no_of_cities
1	MG	745
2	SP	629
3	RS	379
4	PR	364
5	BA	353
6	SC	240
7	GO	178
8	CE	161
9	PE	152
10	RJ	149

   Insights: These are the total number of different cities in each state where the customers are from.

2. In-depth Exploration:

1. Is there a growing trend on e-commerce in Brazil? How can we describe a complete scenario? Can we see some seasonality with peaks at specific months?

Query:

```
select *,(number_of_orders-lag(number_of_orders,1)over(order by Year))/
lag(number_of_orders,1)
over(order by Year) groth_rate
from (SELECT extract(year from order_purchase_timestamp) as Year,
count(order_id) number_of_orders
FROM `target-dataset123.target_market.orders`
group by Year) a
order by Year
```

Result:

Row	Year	number_of_orde	groth_rate
1	2016	329	*null*
2	2017	45101	136.085106...
3	2018	54011	0.19755659...

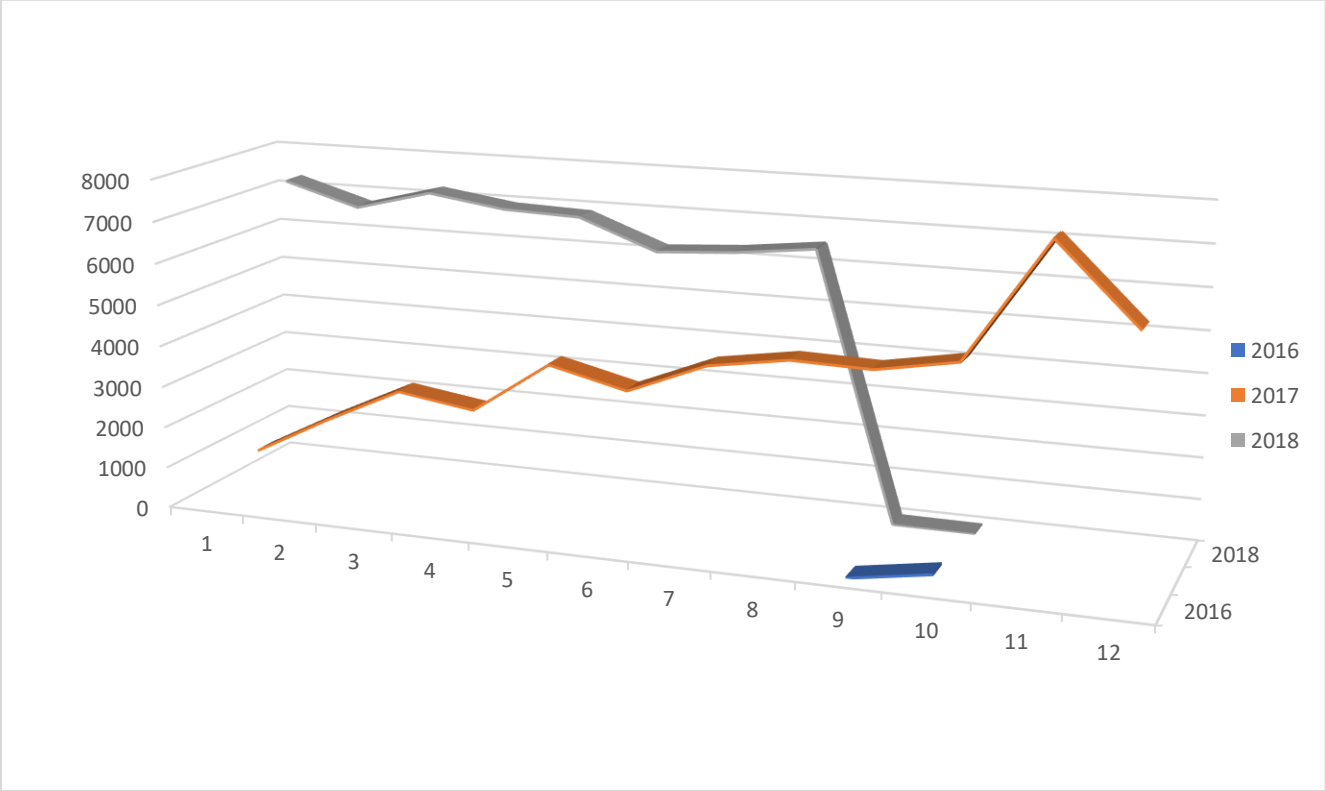
Insights: As the growth rate is positive, we can say that there is a growing trend on e-commerce in Brazil.

Query:

```
SELECT extract(year from order_purchase_timestamp) as Year,
extract(month from order_purchase_timestamp) as Month,
count(order_id) number_of_orders
FROM `target-dataset123.target_market.orders`
group by Year,Month
order by Year,Month
```

Result:

Row	Year	Month	number_of_orde
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
11	2017	8	4331
12	2017	9	4285
13	2017	10	4631
14	2017	11	7544
15	2017	12	5673



Insights: As we have only 3-months of data for the year 2016, only during October the sale was at its peak for that year. In 2017 we see almost a linear growth till the month of October and a sudden exponential raise in the month of November to account for its peak growth for that year. Then during December 2017 there was a decrease in trend but not by much. In the year 2018 the trend stayed almost constant at peak level until the month of August and there was a huge exponential decline in the trend in the month of September and October.

2. What time do Brazilian customers tend to buy (Dawn, Morning, Afternoon or Night)?

Query:

```
with new_table as

(select order_id,Time,case
when Time between ("04:00:00") and ("06:00:00") then "Dawn"
when Time between ("6:00:00" ) and ("12:00:00" )then "Morning"
when Time between ("12:00:00") and ("16:00:00") then "Afternoon"
when Time between ("16:00:00") and ("21:00:00") then "Evening"
else "Night"
end as time_of_day
from (select order_id,extract(time from order_purchase_timestamp ) Time
from `target-dataset123.target_market.orders`) a)

select n1.time_of_day,count(n2.time_of_day) total_orders
from new_table n1
join new_table n2
on n1.order_id = n2.order_id
group by n1.time_of_day order by total_orders desc
```

Result:

time_of_day	total_orders
Evening	30768
Afternoon	25537
Morning	22240
Night	20502
Dawn	394

Insights: From the above result we see that 30K (approx. 30%) of the orders were made in the Evening(4 pm UTC to 9 pm UTC), about 25% in the Afternoon(12 pm UTC to 4 pm UTC),22% in the Morning(between 6 am to 12 pm UTC) and approximately 0.4% in the Dawn(9 pm to 4 am UTC).

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

1. Get month on month orders by states

Query:

```
SELECT c.customer_state, extract(year from o.order_purchase_timestamp) Year,
extract(month from o.order_purchase_timestamp) Month,
count(o.order_id) no_of_orders
FROM `target-dataset123.target_market.orders` o
left join `target-dataset123.target_market.customers`c
on o.customer_id = c.customer_id
group by c.customer_state,Year,Month
having Year = 2016
order by no_of_orders desc
```

Result:

Row	customer_state	Year	Month	no_of_orders
1	SP	2016	10	113
2	RJ	2016	10	56
3	MG	2016	10	40
4	RS	2016	10	24
5	PR	2016	10	19
6	SC	2016	10	11
7	GO	2016	10	9
8	CE	2016	10	8
9	PE	2016	10	7
10	DF	2016	10	6
11	BA	2016	10	4

Insights: São Paulo(SP) had the highest number of orders in October 2016 with 113 orders followed by Rio de Janeiro(RJ) with 56 orders and so on.

Query:

```
SELECT c.customer_state, extract(year from o.order_purchase_timestamp) Year,
extract(month from o.order_purchase_timestamp) Month,
count(o.order_id) no_of_orders
FROM `target-dataset123.target_market.orders` o
left join `target-dataset123.target_market.customers`c
on o.customer_id = c.customer_id
group by c.customer_state,Year,Month
having Year = 2017
order by no_of_orders desc
```

Result:

Row	customer_state	Year	Month	no_of_orders
1	SP	2017	11	3012
2	SP	2017	12	2357
3	SP	2017	10	1793
4	SP	2017	8	1729
5	SP	2017	9	1638
6	SP	2017	7	1604
7	SP	2017	5	1425
8	SP	2017	6	1331
9	RJ	2017	11	1048
10	SP	2017	3	1010
11	MG	2017	11	943
12	SP	2017	4	908
13	RJ	2017	12	783
14	MG	2017	12	691
15	RJ	2017	10	668

Insights: In 2017 São Paulo(SP) had the highest orders for 8 consecutive months with peak of 3012 orders in November 2017.

Query:

```
SELECT c.customer_state, extract(year from o.order_purchase_timestamp) Year,
extract(month from o.order_purchase_timestamp) Month,
count(o.order_id) no_of_orders
FROM `target-dataset123.target_market.orders` o
left join `target-dataset123.target_market.customers`c
on o.customer_id = c.customer_id
group by c.customer_state,Year,Month
having Year = 2018
order by no_of_orders desc
```

Result:

Row	customer_state	Year	Month	no_of_orders
1	SP	2018	8	3253
2	SP	2018	5	3207
3	SP	2018	4	3059
4	SP	2018	1	3052
5	SP	2018	3	3037
6	SP	2018	7	2777
7	SP	2018	6	2773
8	SP	2018	2	2703
9	RJ	2018	2	922
10	RJ	2018	3	907

Insights: Again in the year 2018 São Paulo(SP) had the highest orders for 8 consecutive months with peak orders during August 2018 and May 2018 in the second at just 46 orders less than peak

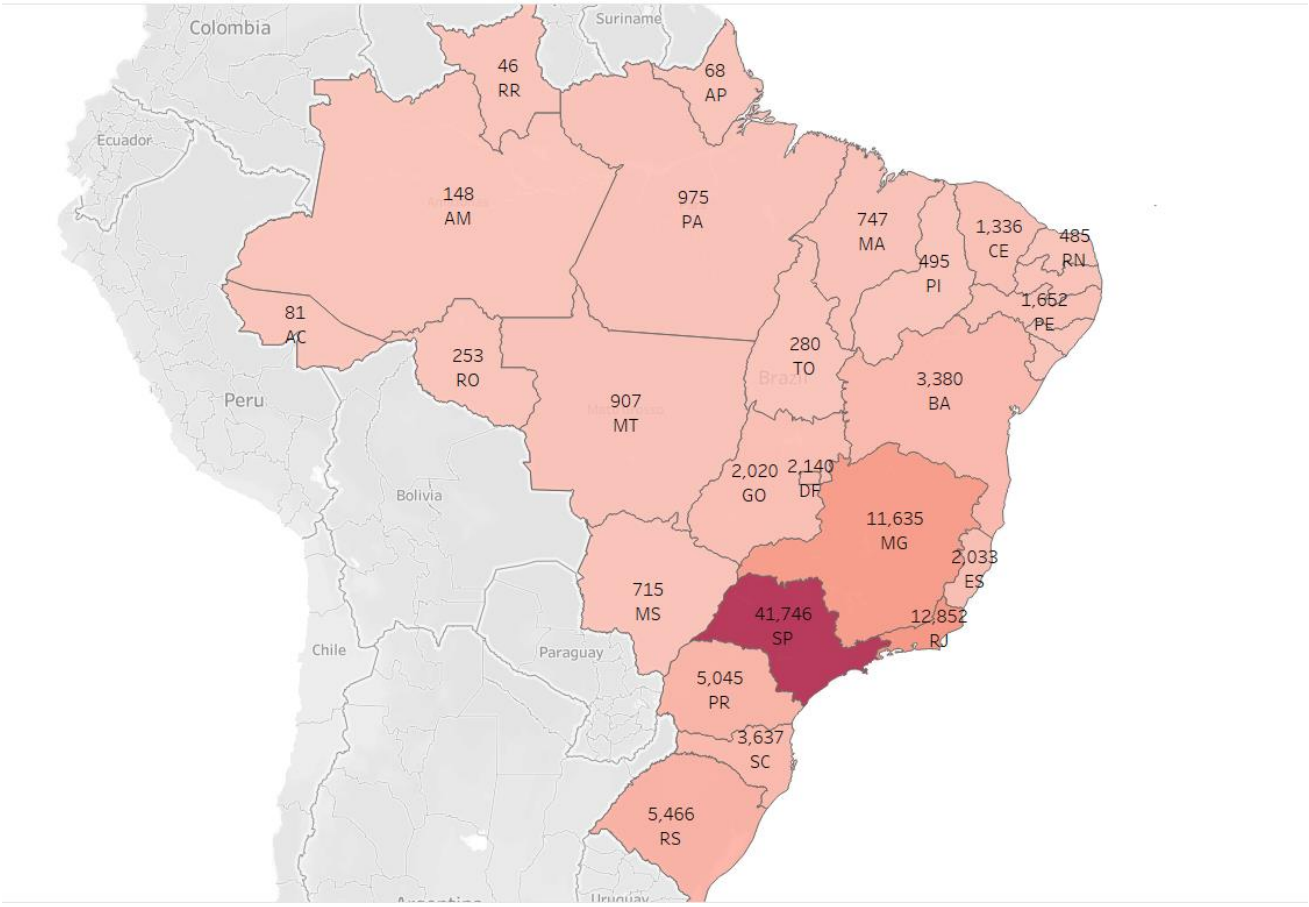
2. Distribution of customers across the states in Brazil

Query:

```
SELECT distinct customer_state,count(distinct customer_id) no_of_cust
FROM `target-dataset123.target_market.customers`
group by  customer_state
order by no_of_cust desc
```

Result:

Row	customer_state	no_of_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



Insights: SP has most number of customers [41746] that's the reason why it has more number of orders and RJ with 12,852 customers is in the second spot.

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

1. Get % increase in cost of orders from 2017 to 2018 (include months between Jan to Aug only) - You can use "payment value" column in payments table

   Query:

```
with table1 as
(SELECT extract(date from o.order_purchase_timestamp) Date_of_purchase,p.payment_value FRO
M `target-dataset123.target_market.payments` p
left join `target_market.orders` o on p.order_id = o.order_id
where extract(month from o.order_purchase_timestamp) between 1 and 8),

table2 as
(select extract(year from Date_of_purchase) as Year, sum(payment_value) as cost_of_orders
from table1
group by Year
order by Year)

select *,(cost_of_orders -lag(cost_of_orders)over(order by Year asc))/
lag(cost_of_orders)over(order by Year asc )*100 as precent_increase_cost
from table2
```

Result:

Row	Year	cost_of_orders	precent_increas
1	2017	3669022.12...	null
2	2018	8694733.83...	136.976871...

Insights: There is a significant increase of 137 percent in cost of orders from 2017 to 2018 considering data for the month between Jan to Aug only.

2. Mean & Sum of price and freight value by customer state

Query:

```
with table1 as
(SELECT oi.order_id,o.customer_id,price,freight_value FROM `target-
dataset123.target_market.order_items` oi
left join `target_market.orders` o on o.order_id = oi.order_id )

select distinct c.customer_state,avg(t1.freight_value)as mean_freight,
sum(t1.freight_value) as sum_freight ,
avg(t1.price) as mean_price ,
sum(t1.price) as sum_price
from table1 t1
join target_market.customers c on
t1.customer_id = c.customer_id
group by c.customer_state
order by c.customer_state
```

Result:

Row	customer_state	mean_freight	sum_freight	mean_price	sum_price
1	AC	40.0733695...	3686.75000...	173.727717...	15982.9499...
2	AL	35.8436711...	15914.5899...	180.889211...	80314.8099...
3	AM	33.2053939...	5478.89000...	135.495999...	22356.8400...
4	AP	34.0060975...	2788.50000...	164.320731...	13474.2999...
5	BA	26.3639589...	100156.679...	134.601208...	511349.990...
6	CE	32.7142016...	48351.5899...	153.758261...	227254.709...
7	DF	21.0413549...	50625.4999...	125.770548...	302603.939...
8	ES	22.0587765...	49764.5999...	121.913701...	275037.309...
9	GO	22.7668152...	53114.9799...	126.271731...	294591.949...
10	MA	38.2570024...	31523.7700...	145.204150...	119648.219...

Row	customer_state	mean_freight	sum_freight
1	SP	15.1472753...	718723.069...
2	RJ	20.9609239...	305589.310...
3	MG	20.6301668...	270853.460...
4	RS	21.7358043...	135522.740...
5	PR	20.5316515...	117851.680...
6	BA	26.3639589...	100156.679...
7	SC	21.4703687...	89660.2600...
8	PE	32.9178626...	59449.6599...
23	RO	41.0697122...	11417.3800...
24	AM	33.2053939...	5478.89000...
25	AC	40.0733695...	3686.75000...
26	AP	34.0060975...	2788.50000...
27	RR	42.9844230...	2235.19000...

Insights: SP might have highest total freight value but it has the lowest average fright value. This may be due to the highest number of orders from that state. RR(Roraima) has the lowest sum of freight with highest average due to low number of orders.

Row	customer_state	mean_price	sum_price
1	SP	109.653629...	5202955.05...
2	RJ	125.117818...	1824092.66...
3	MG	120.748574...	1585308.02...
4	RS	120.337453...	750304.020...
5	PR	119.004139...	683083.760...
6	SC	124.653577...	520553.340...
7	BA	134.601208...	511349.990...
8	DF	125.770548...	302603.939...
9	GO	126.271731...	294591.949...
10	ES	121.913701...	275037.309...
17	PB	191.475215...	115268.079...
22	TO	157.529333...	49621.7400...
23	RO	165.973525...	46140.6400...
24	AM	135.495999...	22356.8400...
25	AC	173.727717...	15982.9499...
26	AP	164.320731...	13474.2999...
27	RR	150.565961...	7829.42999...

Insights: Highest sum of prices was seen in SP as it has most number of orders and customers but highest mean sum price was from PB(Paraíba).

5. Analysis on sales, freight and delivery time

1. Calculate days between purchasing, delivering and estimated delivery
2. Find time_to_delivery & diff_estimated_delivery. Formula for the same given below:
    - time_to_delivery = order_purchase_timestamp-order_delivered_customer_date
    - diff_estimated_delivery = order_estimated_delivery_date-order_delivered_customer_date

Query:

```
SELECT order_id,order_purchase_timestamp,order_delivered_customer_date,
date_diff(order_delivered_customer_date, order_purchase_timestamp, day) time_to_delivery ,
date_diff(order_estimated_delivery_date, order_delivered_customer_date, day) diff_estimate
d_delivery
FROM `target-dataset123.target_market.orders`
where order_delivered_customer_date is not null and order_status = "delivered"
order by time_to_delivery desc
```

3. Group data by state, take mean of freight_value, time_to_delivery, diff_estimated_delivery

```
with table1 as
(SELECT o.order_id,o.order_purchase_timestamp,order_delivered_customer_date,
date_diff(order_delivered_customer_date, order_purchase_timestamp, day) time_to_delivery ,
date_diff(order_estimated_delivery_date, order_delivered_customer_date, day) diff_estimate
d_delivery ,customer_state,freight_value
FROM `target-dataset123.target_market.orders` o
left join `target_market.order_items` oi
on o.order_id = oi.order_id
left join `target_market.customers`c on
o.customer_id = c.customer_id
where order_delivered_customer_date is not null and order_status = "delivered"
order by time_to_delivery desc)

select customer_state,avg(freight_value) mean_freight,avg(table1.time_to_delivery)mean_tim
e_to_delivery,avg(table1.diff_estimated_delivery)mean_estimated_delivery from table1
group by customer_state
order by customer_state
```

Row	customer_state	mean_freight	mean_time_to_d	mean_estimated
1	AC	40.0479120...	20.3296703...	20.0109890...
2	AL	35.8706557...	23.9929742...	7.97658079...
3	AM	33.3106134...	25.9631901...	18.9754601...
4	AP	34.1604938...	27.7530864...	17.4444444...
5	BA	26.4875563...	18.7746402...	10.1194678...
6	CE	32.7344950...	20.5371669...	10.2566619...
7	DF	21.0721613...	12.5014861...	11.2747346...
8	ES	22.0289797...	15.1928089...	9.76853932...
9	GO	22.5628678...	14.9481774...	11.3728590...
10	MA	38.4927125...	21.2037500...	9.10999999...
11	MG	20.6263425...	11.5140910...	12.3990399...
12	MS	23.3509001...	15.1072749...	10.3378545...

4. Sort the data to get the following:
5. Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5

Top 5 highest freight value in descending order

Row	customer_state	mean_freight	mean_time_to_d	mean_estimated
1	PB	43.0916894...	20.1194539...	12.1501706...
2	RR	43.0880434...	27.8260869...	17.4347826...
3	RO	41.3305494...	19.2820512...	19.0805860...
4	AC	40.0479120...	20.3296703...	20.0109890...
5	PI	39.1150860...	18.9311663...	10.6826003...

Top 5 lowest freight value in ascending order

Row	customer_state	mean_freight	mean_time_to_d	mean_estimated
1	SP	15.1149899...	8.25966279...	10.2641415...
2	PR	20.4718162...	11.4807930...	12.5338998...
3	MG	20.6263425...	11.5140910...	12.3990399...
4	RJ	20.9114360...	14.6888213...	11.1396450...
5	DF	21.0721613...	12.5014861...	11.2747346...

Insight: PB and RR has the highest freight value. Some states like PB, RR, RO, AC and PI etc. need to evaluate their method of transportation. They could find a cheaper way of transport. Ship more products at a time, less often. SP has the lowest mean freight charges which is comparatively good which may be due to high number of orders.

6. Top 5 states with highest/lowest average time to delivery

```
Query: Highest average time.
with table1 as
(SELECT o.order_id,o.order_purchase_timestamp,order_delivered_customer_date,
date_diff(order_delivered_customer_date,order_purchase_timestamp,day)time_to_delivery,
date_diff(order_estimated_delivery_date,order_delivered_customer_date,day)
diff_estimated_delivery,customer_state,freight_value
FROM `target-dataset123.target_market.orders` o
left join `target_market.order_items` oi
on o.order_id = oi.order_id
left join `target_market.customers`c on
o.customer_id = c.customer_id
where order_delivered_customer_date is not null and order_status = "delivered"
order by time_to_delivery desc)

select customer_state,avg(freight_value) mean_freight,avg(table1.time_to_delivery)mean_tim
e_to_delivery,avg(table1.diff_estimated_delivery)mean_estimated_delivery from table1
group by customer_state
order by mean_time_to_delivery desc
limit 5

Result:
```

Row	customer_state	mean_freight	mean_time_to_d	mean_estimated_delivery
1	RR	43.0880434...	27.8260869...	17.434782608695652
2	AP	34.1604938...	27.7530864...	17.444444444444443
3	AM	33.3106134...	25.9631901...	18.975460122699381
4	AL	35.8706557...	23.9929742...	7.9765807962529349
5	PA	35.6290132...	23.3017077...	13.37476280834913

Insights: RR has the highest mean time to delivery is the highest along with AP. This may be because these places are located very far from the product shipping station or it might be a remote place as most parts of RR and AP are covered by Amazon Rain Forest. This can be overcome by setting up small intermediate shipping facilities in these states.

Query: lowest mean time to delivery

```
order by mean_time_to_delivery asc
limit 5
```

Row	customer_state	mean_freight	mean_time_to_d	mean_estimated
1	SP	15.1149899...	8.25966279...	10.2641415...
2	PR	20.4718162...	11.4807930...	12.5338998...
3	MG	20.6263425...	11.5140910...	12.3990399...
4	DF	21.0721613...	12.5014861...	11.2747346...
5	SC	21.5073590...	14.5172077...	10.6646326...

Insights: SP is the state where mean delivery time is faster than any other state. This is the state which has most number of orders, customers and contributes to most part of the company's revenue.

7. Top 5 states where delivery is really fast/ not so fast compared to estimated date

Query: Delivery really fast

```
with table1 as
(SELECT o.order_id,o.order_purchase_timestamp,order_delivered_customer_date,
date_diff(order_delivered_customer_date,order_purchase_timestamp,day)time_to_delivery,
date_diff(order_estimated_delivery_date,order_delivered_customer_date,day)
diff_estimated_delivery,customer_state,freight_value
FROM `target-dataset123.target_market.orders` o
left join `target_market.order_items` oi
on o.order_id = oi.order_id
left join `target_market.customers`c on
o.customer_id = c.customer_id
where order_delivered_customer_date is not null and order_status = "delivered"
order by time_to_delivery desc)

select customer_state,avg(freight_value) mean_freight,avg(table1.time_to_delivery)mean_tim
e_to_delivery,avg(table1.diff_estimated_delivery)mean_estimated_delivery,avg(table1.time_t
o_delivery)-avg(table1.diff_estimated_delivery) delivery_compared_to_estimate from table1
group by customer_state
order by delivery_compared_to_estimate asc
limit 5
```

Row	customer_state	mean_freight	mean_time_to_d	mean_estimated	delivery_compar
1	SP	15.1149899...	8.25966279...	10.2641415...	-2.00447880...
2	PR	20.4718162...	11.4807930...	12.5338998...	-1.05310674...
3	MG	20.6263425...	11.5140910...	12.3990399...	-0.88494890...
4	RO	41.3305494...	19.2820512...	19.0805860...	0.20146520...
5	AC	40.0479120...	20.3296703...	20.0109890...	0.31868131...

Insights: In SP the products were delivered 2 days before the estimated time. Which is good considering the sales in that state. Followed by PR with one day before the estimate, MG, RO and AC with no delay from the estimated date.

Query: Delivery not so fast compared to estimate

```
order by delivery_compared_to_estimate desc
limit 5
```

Row	customer_state	mean_freight	mean_time_to_d	mean_estimated	delivery_compar
1	AL	35.8706557...	23.9929742...	7.97658079...	16.0163934...
2	MA	38.4927125...	21.2037500...	9.10999999...	12.0937500...
3	SE	36.5731733...	20.9786666...	9.16533333...	11.8133333...
4	RR	43.0880434...	27.8260869...	17.4347826...	10.3913043...
5	AP	34.1604938...	27.7530864...	17.4444444...	10.3086419...

Insights: AL delays there delivery by an average of 16 days.

6. Payment type analysis:

1. Month over Month count of orders for different payment types

Query:

```
with table1 as
(SELECT p.*,extract(date from order_purchase_timestamp) order_date FROM `target-
dataset123.target_market.payments` p
left join `target_market.orders`o
on p.order_id=o.order_id)
select distinct payment_type,extract(month from order_date)Month,
count(order_id) over(partition by payment_type
order by extract(month from order_date)) number_of_customers
from table1
where extract(year from order_date)=2016
order by month asc, number_of_customers desc
```

Result:

Row	payment_type	Month	number_of_cust
1	credit_card	9	3
2	credit_card	10	257
3	UPI	10	63
4	voucher	10	23
5	debit_card	10	2
6	credit_card	12	258

Insights: For the year 2016 all 3 customers used credit card as their mode of payment in September. In October 257 used credit card and 63 used UPI (or it might be some kind of online payment, as UPI is payment mode of India) and few used vouchers. In December all transaction were done by using credit card.

Query:

```
where extract(year from order_date)=2017
order by month asc, number_of_customers desc
```

Result:

Row	payment_type	Month	number_of_cust
1	credit_card	1	583
2	UPI	1	197
3	voucher	1	61
4	debit_card	1	9
5	credit_card	2	1939
6	UPI	2	595
7	voucher	2	180
8	debit_card	2	22
9	credit_card	3	3955
10	UPI	3	1185
11	voucher	3	380
12	debit_card	3	53

Query:

```
where extract(year from order_date)=2018
order by month asc, number_of_customers desc
```

Result:

Row	payment_type	Month	number_of_cust
1	credit_card	1	5520
2	UPI	1	1518
3	voucher	1	416
4	debit_card	1	109
5	credit_card	2	10773
6	UPI	2	2843
7	voucher	2	721
8	debit_card	2	178
9	credit_card	3	16464
10	UPI	3	4195
11	voucher	3	1112
12	debit_card	3	256

Insights: In each year and in almost every month the most used payment type is credit card as it is the most convenient way of payment as it is easy and secure. Second most preferred payment mode was UPI? Or online payment. Least used payment mode turns out to be debit card.

2.Count of orders based on the no. of payment instalments

Query:

```
SELECT distinct payment_installments,count(distinct order_id)
over(partition by payment_installments) no_of_orders
FROM `target-dataset123.target_market.payments`
order by no_of_orders desc
```

Result:

Row	payment_installr	no_of_orders
1	1	49060
2	2	12389
3	3	10443
4	4	7088
5	10	5315
6	5	5234
7	8	4253
8	6	3916
9	7	1623
10	9	644
11	12	133
12	15	74
13	18	27
14	11	23
15	24	18
16	20	17
17	13	16
18	14	15
19	17	8

Insights: Most orders were paid in a single instalment. Highest number of instalments taken was 24. By this we can assume that most of the customers do not prefer to take loan or they make only affordable purchases.

7.Product category analysis:

1. Total number of different product category and number of orders in each category.

Query:

```
select count(distinct product_category) no_of_categories
from `target_market.products`
```

Result:

Row	no_of_categorie
1	73

Insights: There are 73 different product categories sold on the website.

Query:

```
SELECT product_category,
count(distinct order_id) no_of_orders
FROM `target-dataset123.target_market.order_items` o
left join `target_market.products`p
on o.product_id = p.product_id
group by product_category
order by no_of_orders desc
```

Result: Top 10

Row	product_category	no_of_orders
1	bed table bath	9417
2	HEALTH BEAUTY	8836
3	sport leisure	7720
4	computer accessories	6689
5	Furniture Decoration	6449
6	housewares	5884
7	Watches present	5624
8	telephony	4199
9	automotive	3897
10	toys	3886

Bottom 10

Row	product_category	no_of_orders
1	insurance and services	2
2	PC Gamer	8
3	Fashion Children's Clothing	8
4	cds music dvds	12
5	La Cuisine	13
6	Kitchen portable and food coach	14
7	Arts and Crafts	23
8	House Comfort 2	24
9	Hygiene diapers	27
10	Fashion Sport	27

Insights: Bed, table and bath category has the highest order count followed by beauty, sport leisure and so on. Insurance and service , PC games and Children's clothing are the least placed orders.

2. Top 5 and bottom 5 rated products.

Query: Top 5

```
SELECT product_category,
round(avg( orev.review_score),2) mean_review_score
FROM `target-dataset123.target_market.order_items` oi
join `target_market.products`p
on oi.product_id = p.product_id
join `target_market.order_reviews`orev
on oi.order_id = orev.order_id
group by product_category
order by mean_review_score desc
limit 5
```

Result

Row	product_category	mean_review_sc
1	cds music dvds	4.64
2	Fashion Children's Clothing	4.5
3	General Interest Books	4.45
4	Construction Tools Tools	4.44
5	flowers	4.42

Insights: We can see cds music dvds have been rated and children's clothing are among the top 5 highest rated products because very few people who purchased those products and most of them have been satisfied that's the reason they have good rating.

Query:

```
order by mean_review_score asc
limit 5
```

Result:

Row	product_category	mean_review_sc
1	insurance and services	2.5
2	Hygiene diapers	3.26
3	Kitchen portable and food coach	3.27
4	PC Gamer	3.33
5	Furniture office	3.49

Insights: Insurance and services is the lowest rated even though only 2 orders were placed in this category. This might be one of the reasons for their poor sales.

8. Seller Analysis:
   1. Total number of sellers and the Seller who have had highest sales.

      Query:

```
select count(distinct seller_id) no_of_sellers
from `target_market.sellers`
```

      Result:

Row	no_of_sellers
1	3095

      Query:

```
SELECT oi.seller_id,count(distinct order_id) no_of_orders
FROM `target-dataset123.target_market.order_items` oi
join `target_market.sellers` s
on oi.seller_id = s.seller_id
group by oi.seller_id
order by no_of_orders desc
```

      Result:

Row	seller_id	no_of_orders
1	6560211a19b47992c3666cc44...	1854
2	4a3ca9315b744ce9f8e937436...	1806
3	cc419e0650a3c5ba77189a188...	1706
4	1f50f920176fa81dab994f9023...	1404
5	da8622b14eb17ae2831f4ac5b...	1314
6	955fee9216a65b617aa5c0531...	1287
7	7a67c85e85bb2ce8582c35f22...	1160
8	ea8482cd71df3c1969d7b9473...	1146
9	4869f7a5dfa277a7dca6462dcf...	1132
10	3d871de0142ce09b7081e2b9d...	1080
11	7c67e1448b00f6e969d365cea...	982

   Insights: As there's no name of the seller these are the sellers who have highest number of orders.

Final Insights:

As this is a vast data set we can do numerous Exploratory data analysis. By doing the necessary analysis we can conclude that state SP is the best performing state in terms of sales and acquiring/ maintaining customer base with minimal freight cost and lowest time to delivery. Whereas State like RR and few other states have least contribution towards sale and have highest freight cost and poor judgement of estimated delivery time.

The poor performing states like RR, AP, AL must consider revising their estimated delivery date and minimize freight charges by shipping more products, less frequently. More freight charges does not mean they are not doing a great job, it may be that these places are too far from where the products are shipped, So, establishing intermediate shipping facilities closer to these far away places might reduce the time to delivery and also the freight cost. This may also result in more order and customers in the future.