Project:TARGET

Context:

Target is one of the world’s most recognized brands and one of America’s leading retailers. Target makes itself a preferred shopping destination by offering outstanding value, inspiration, innovation and an exceptional guest experience that no other retailer can deliver.

This business case has information of 100k orders from 2016 to 2018 made at Target in Brazil. Its features allows viewing an order from multiple dimensions: from order status, price, payment and freight performance to customer location, product attributes and finally reviews written by customers.

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

1.1Data type of columns in a table

1.2Time period for which the data is given

1.3Cities and States of customers ordered during the given period

Explanation of Dataset:

In this dataset, there are 8 tables given by name:

1. customers.csv

2. geolocation.csv

3. order_items.csv

4. payments.csv

5. reviews.csv

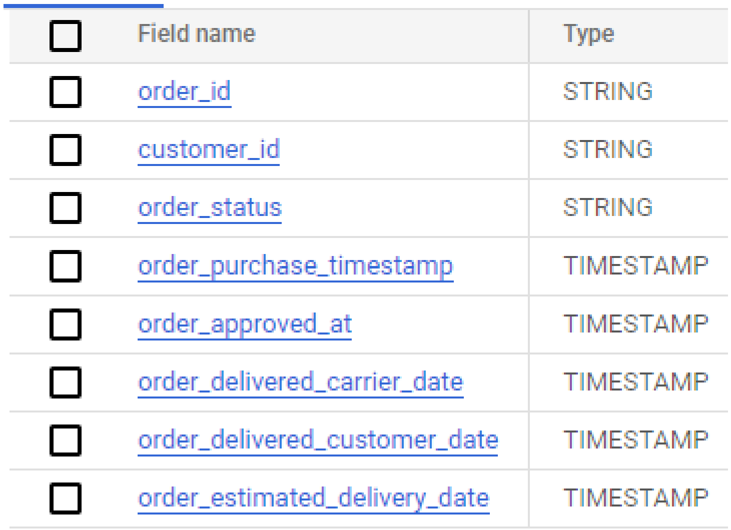
6. orders.csv

7. products.csv

8. sellers.csv

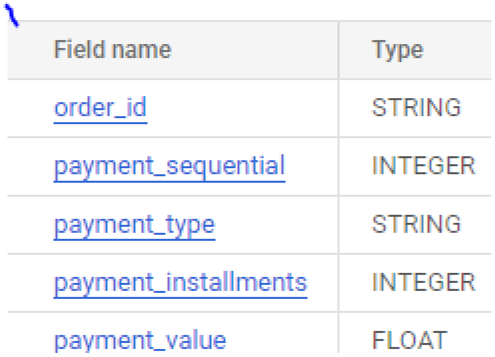
1.1Data type of columns in a table

Orders:



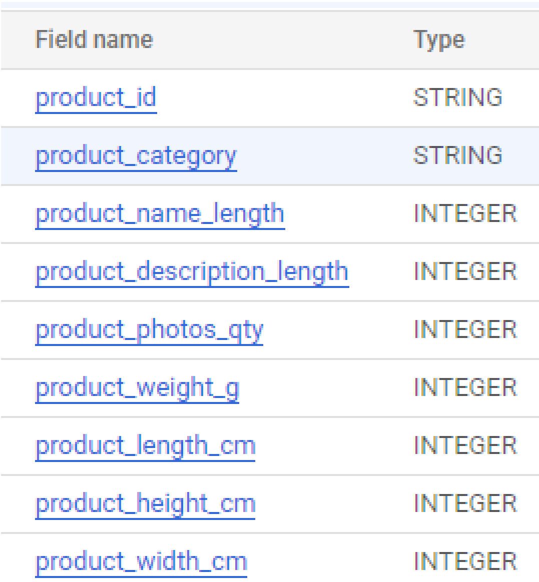
□	Field name	Type
□	order_id	STRING
□	customer_id	STRING
□	order_status	STRING
□	order_purchase_timestamp	TIMESTAMP
□	order_approved_at	TIMESTAMP
□	order_delivered_carrier_date	TIMESTAMP
□	order_delivered_customer_date	TIMESTAMP
□	order_estimated_delivery_date	TIMESTAMP

Payments:



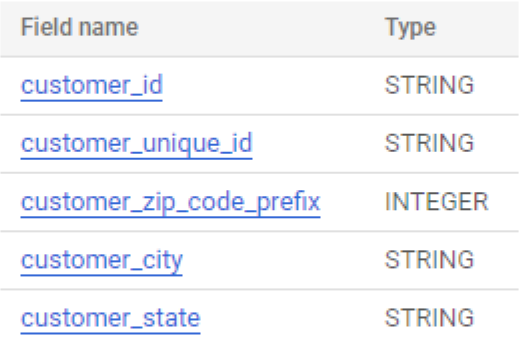
Field name	Type
order_id	STRING
payment_sequential	INTEGER
payment_type	STRING
payment_installments	INTEGER
payment_value	FLOAT

Products:



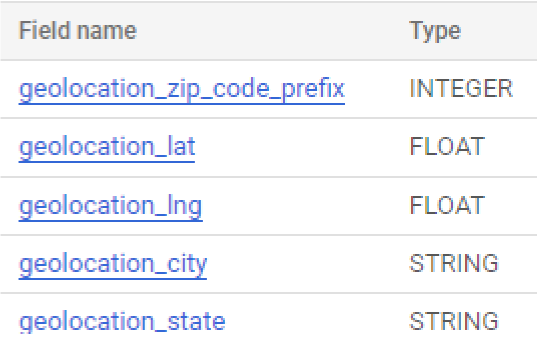
Field name	Type
product_id	STRING
product_category	STRING
product_name_length	INTEGER
product_description_length	INTEGER
product_photos_qty	INTEGER
product_weight_g	INTEGER
product_length_cm	INTEGER
product_height_cm	INTEGER
product_width_cm	INTEGER

Customers:



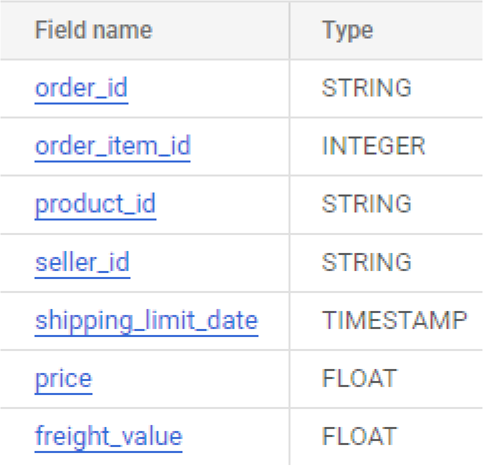
Field name	Type
customer_id	STRING
customer_unique_id	STRING
customer_zip_code_prefix	INTEGER
customer_city	STRING
customer_state	STRING

Geolocations:



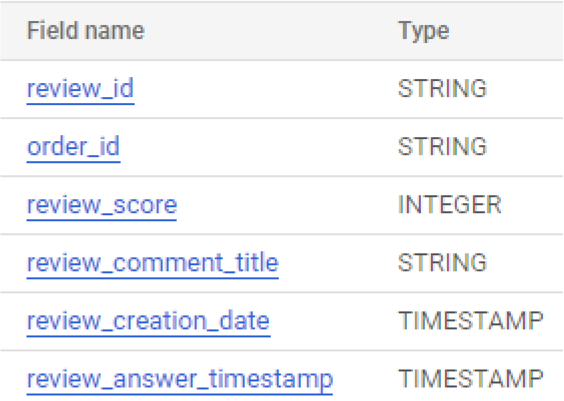
Field name	Type
geolocation_zip_code_prefix	INTEGER
geolocation_lat	FLOAT
geolocation_lng	FLOAT
geolocation_city	STRING
geolocation_state	STRING

Order_items:



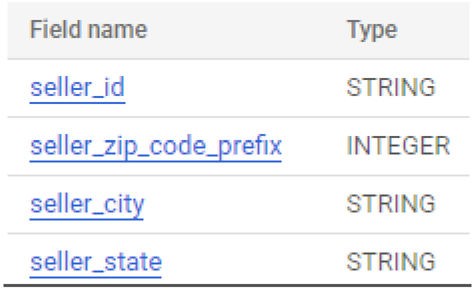
Field name	Type
order_id	STRING
order_item_id	INTEGER
product_id	STRING
seller_id	STRING
shipping_limit_date	TIMESTAMP
price	FLOAT
freight_value	FLOAT

Order_reviews:



Field name	Type
review_id	STRING
order_id	STRING
review_score	INTEGER
review_comment_title	STRING
review_creation_date	TIMESTAMP
review_answer_timestamp	TIMESTAMP

Sellers:

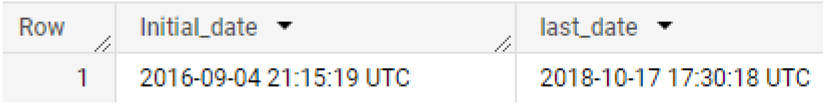


Field name	Type
seller_id	STRING
seller_zip_code_prefix	INTEGER
seller_city	STRING
seller_state	STRING

1.2 Time period for which the data is given

```
select
min(order_purchase_timestamp) as Initial_date,
max(order_purchase_timestamp) as last_date
from `target-project-388017.Target.Orders`
```

Output:



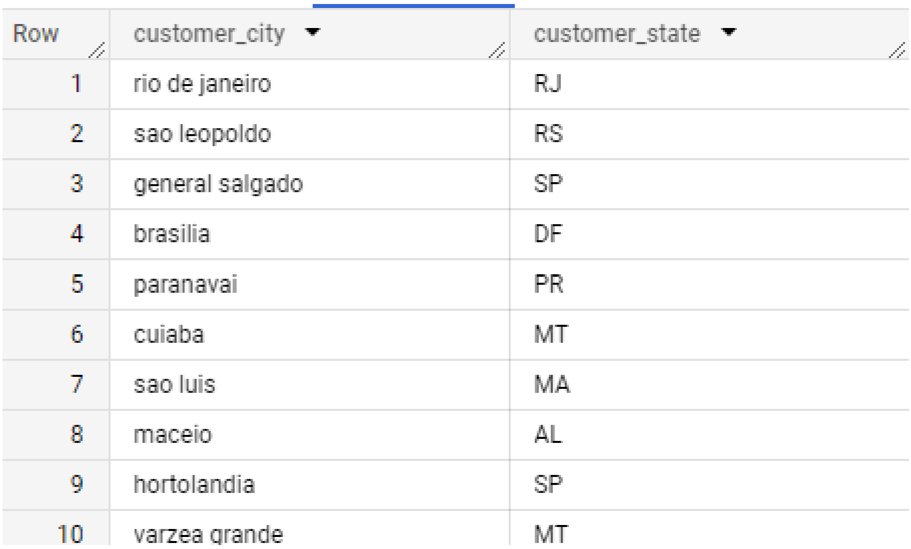
Row	Initial_date	last_date
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC

From the above table, we will get the start date and last date of the data range.

1.3 Cities and States of customers ordered during the given period

```
select distinct
c.customer_city,
c.customer_state
from `Target.Orders` as o
join `Target.customers` as c
on o.customer_id=c.customer_id
where o.order_purchase_timestamp between "2016-09-04" and "2018-10-17"
limit 10;
```

Output:



Row	customer_city	customer_state
1	rio de janeiro	RJ
2	sao leopoldo	RS
3	general salgado	SP
4	brasilia	DF
5	paranavai	PR
6	cuiaba	MT
7	sao luis	MA
8	maceio	AL
9	hortolandia	SP
10	varzea grande	MT

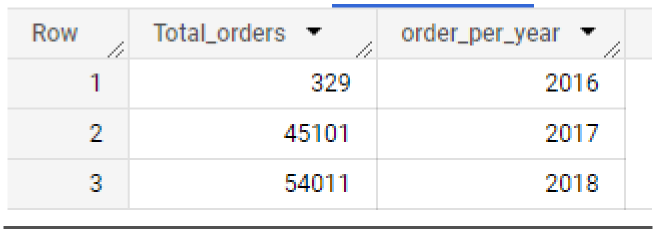
From the above table, cities and states we get the records of customers who ordered during the given time period. That time period is `"2016-09-04" and "2018-10-17".`

2.In-depth Exploration:

2.1 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?

```
select
count(order_id) as Total_orders,
extract(year from order_purchase_timestamp) as orders_per_year
from `Target.Orders`
group by orders_per_year
order by orders_per_year
```

Output:

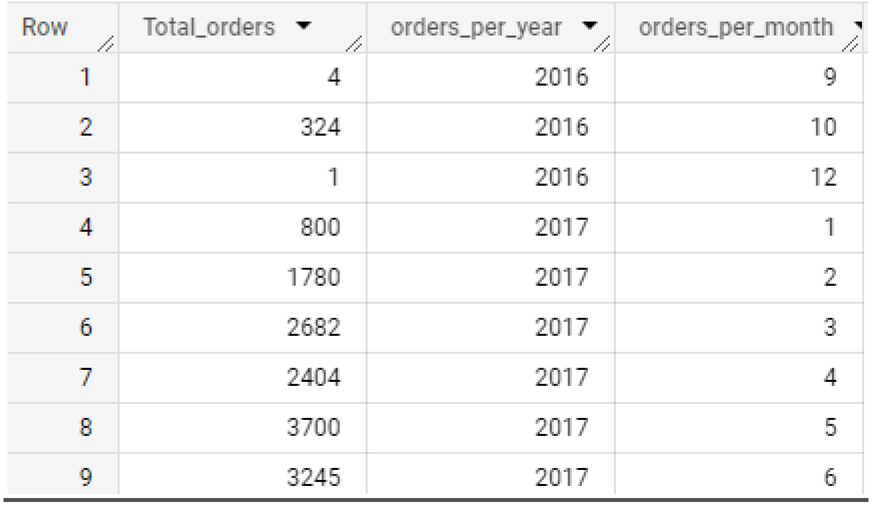


Row	Total_orders	order_per_year
1	329	2016
2	45101	2017
3	54011	2018

As we can see trend of e-commerce in Brazil is growing per year. Total orders in 2016 is 329, total orders in 2017 is 45101 and total orders in 2018 is 54011.

```
select
count(order_id) as Total_orders,
extract(year from order_purchase_timestamp) as orders_per_year,
extract(month from order_purchase_timestamp) as orders_per_month
from `Target.Orders`
group by orders_per_year,orders_per_month
order by orders_per_year,orders_per_month
```

Solution:



Row	Total_orders	orders_per_year	orders_per_month
1	4	2016	9
2	324	2016	10
3	1	2016	12
4	800	2017	1
5	1780	2017	2
6	2682	2017	3
7	2404	2017	4
8	3700	2017	5
9	3245	2017	6

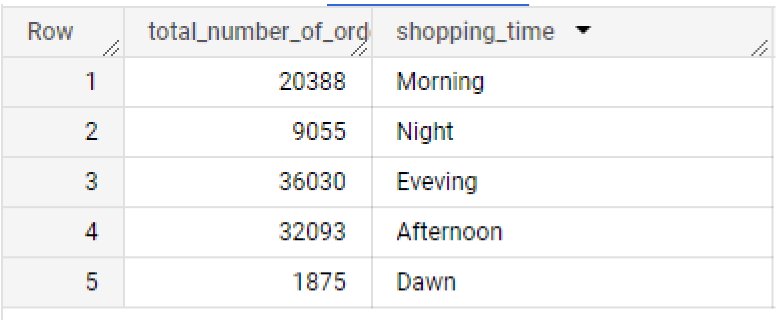
From the above table , we can see the month wise order details. Orders with respect to month in 2016 was highest in November(10th month) which is 324 and orders with respect to month in 2017 was highest in may(5th month) which is 3700.

Number of orders are increasing per year but in some months orders are decreasing and increasing. So there is upward downward range of orders month wise.

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

```
Select
count(order_id)as total_number_of_orders,
case
when
extract(time from order_purchase_timestamp) between "05:01:00" and "7:59:00"
Then "Dawn"
when
extract(time from order_purchase_timestamp) between "08:00:00" and "11:59:00"
Then "Morning"
when
extract(time from order_purchase_timestamp) between "12:00:00" and "16:59:00"
Then "Afternoon"
when
extract(time from order_purchase_timestamp) between "17:00:00" and "22:59:00"
Then "Eveving"
ELSE "Night"
end as shopping_time
from `Target.Orders`
group by shopping_time
```

Output:



Row	total_number_of_ord	shopping_time
1	20388	Morning
2	9055	Night
3	36030	Eveving
4	32093	Afternoon
5	1875	Dawn

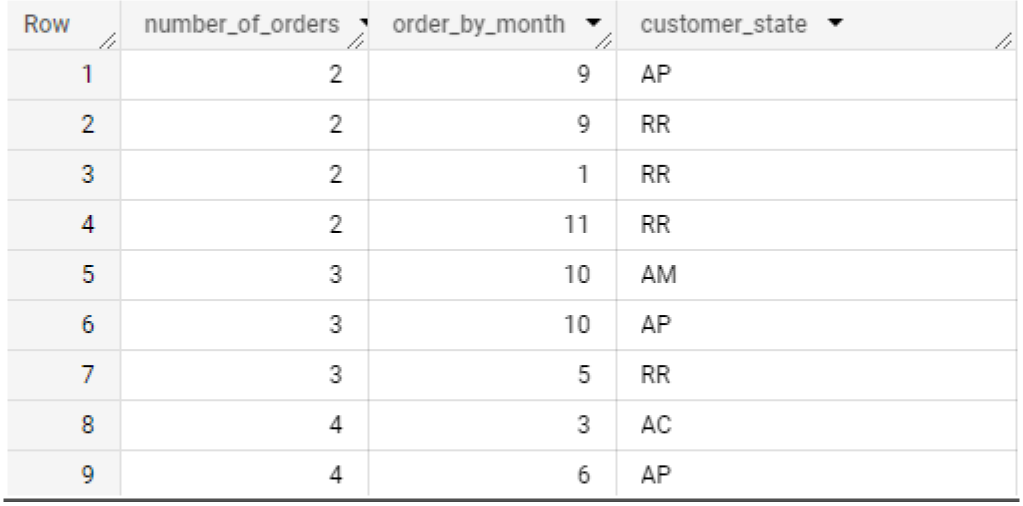
From the above table, customers of Brazil likes to shop more in afternoon.

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

3.1 Get month on month orders by state Distribution of customers across the states in Brazil:

```
select
count(order_id) as number_of_orders,
extract(month from order_purchase_timestamp) as order_by_month,
c.customer_state
from `Target.orders` as o
join `Target.customers` as c
on o.customer_id = c.customer_id
group by c.customer_state,order_by_month
order by number_of_orders,customer_state;
```

Output:



Row	number_of_orders	order_by_month	customer_state
1	2	9	AP
2	2	9	RR
3	2	1	RR
4	2	11	RR
5	3	10	AM
6	3	10	AP
7	3	5	RR
8	4	3	AC
9	4	6	AP

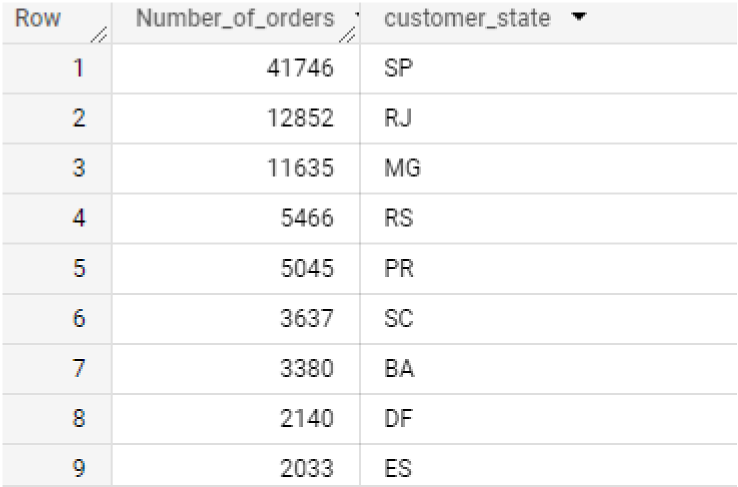
From the above table we can see the orders per month categorized by states. The number of orders varies from different states in different months. Few months show very less orders and this attributes to seasonal demand, market trend etc.

From this marketing campaigning can be done, can also set a target for coming years and information about loss giving prices can be evaluated to eradicate it.

3.2 Distribution of customers across the states in Brazil

```
select
count(order_id) as total_orders,
c.customer_state
from `Target.orders` as o
join `Target.customers` as c
on o.customer_id = c.customer_id
group by c.customer_state
order by total_orders desc;
```

Output:



Row	Number_of_orders	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

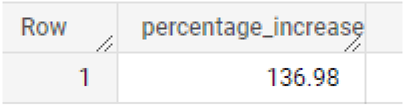
From the above table, total number of orders per state can be evaluated. Some states show higher number of orders as compared to others. This data can be helpful in evaluating market potential, expansion plans, Discount plans, investments plans and many others things.

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

4.1 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

```
with cost_table as
 (
select sum(p.payment_value) as total_cost,
extract(year from order_purchase_timestamp) as years
from `target-project-388017.Target.Payments` as p
join `target-project-388017.Target.orders` as o
 on p.order_id = o.order_id
where extract(year from order_purchase_timestamp) in (2017,2018)
and extract(month from order_purchase_timestamp) between 1 and 8
group by years)
SELECT
round((((select total_cost from cost_table where years = 2018) - total_cost)
/ total_cost * 100),2)AS percentage_increase
FROM
cost_table
WHERE years = 2017;
```

Output:



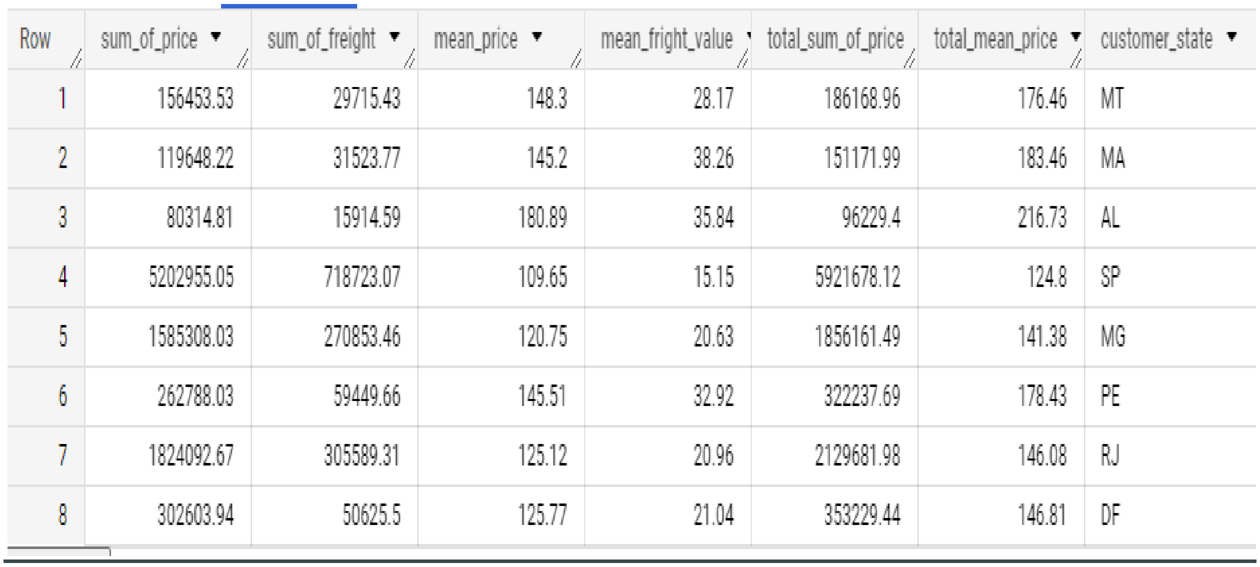
Row	percentage_increase
1	136.98

The percentage increase from 2017 to 2018 is 136.98 between January to august. This means average orders by customers increased.

4.2 Mean & Sum of price and freight value by customer state

```
    select
     round(sum(oi.price),2) as sum_of_price,
    round(sum(oi.freight_value),2) as sum_of_freight,
    round(avg(oi.price),2) as mean_price,
    round(avg(oi.freight_value),2) as mean_fright_value,
    round(sum(oi.price + oi.freight_value),2) as total_sum_of_price,
    round(Avg(oi.price + oi.freight_value),2) as total_mean_price,
    c.customer_state
    from `target-project-388017.Target.customers` as c
   join `target-project-388017.Target.Orders` as o on
    c.customer_id = o.customer_id
    join `target-project-388017.Target.order_items` as oi
     on oi.order_id = o.order_id
    group by c.customer_state
```

Output:



Row	sum_of_price	sum_of_freight	mean_price	mean_fright_value	total_sum_of_price	total_mean_price	customer_state
1	156453.53	29715.43	148.3	28.17	186168.96	176.46	MT
2	119648.22	31523.77	145.2	38.26	151171.99	183.46	MA
3	80314.81	15914.59	180.89	35.84	96229.4	216.73	AL
4	5202955.05	718723.07	109.65	15.15	5921678.12	124.8	SP
5	1585308.03	270853.46	120.75	20.63	1856161.49	141.38	MG
6	262788.03	59449.66	145.51	32.92	322237.69	178.43	PE
7	1824092.67	305589.31	125.12	20.96	2129681.98	146.08	RJ
8	302603.94	50625.5	125.77	21.04	353229.44	146.81	DF

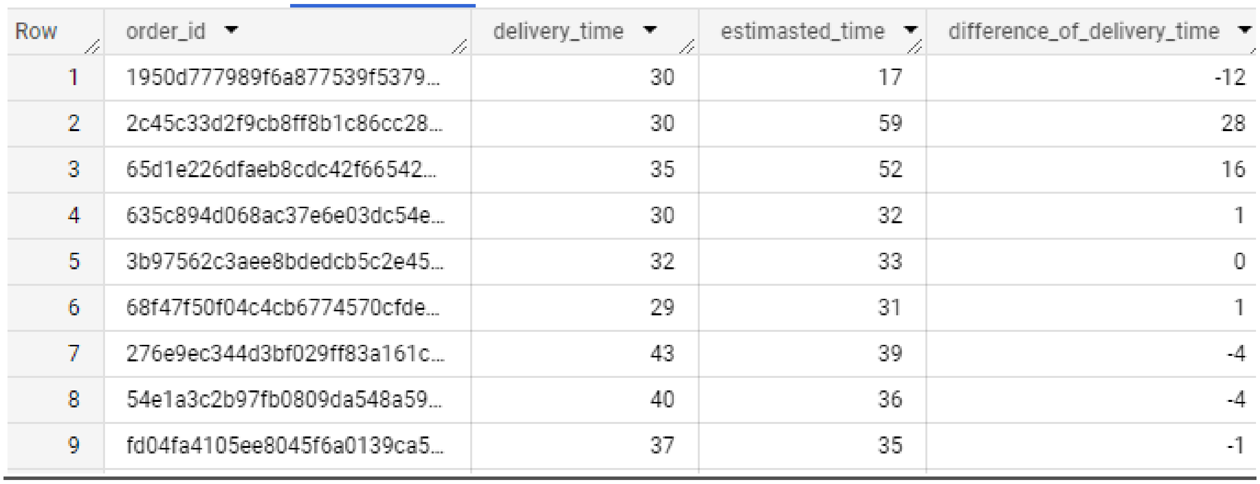
From the above table , sum of price, sum of freight, mean price, mean freight, total sum of price and freight and total mean of price and freight according to states can be calculated.

5 Analysis on sales, freight and delivery time

5.2Calculate days between purchasing, delivering and estimated delivery

```
select
order_id,
date_diff(order_delivered_customer_date,order_purchase_timestamp,day) as delivery_time,
date_diff(order_estimated_delivery_date,order_purchase_timestamp,day) as
estimasted_time,
date_diff(order_estimated_delivery_date,order_delivered_customer_date,day) as
difference_of_delivery_time
from `target-project-388017.Target.Orders`
where date_diff(order_delivered_customer_date,order_purchase_timestamp,day) is not null
```

Output:



Row	order_id	delivery_time	estimasted_time	difference_of_delivery_time
1	1950d777989f6a877539f5379...	30	17	-12
2	2c45c33d2f9cb8ff8b1c86cc28...	30	59	28
3	65d1e226dfaeb8cdc42f66542...	35	52	16
4	635c894d068ac37e6e03dc54e...	30	32	1
5	3b97562c3aee8bdedcb5c2e45...	32	33	0
6	68f47f50f04c4cb6774570cfde...	29	31	1
7	276e9ec344d3bf029ff83a161c...	43	39	-4
8	54e1a3c2b97fb0809da548a59...	40	36	-4
9	fd04fa4105ee8045f6a0139ca5...	37	35	-1

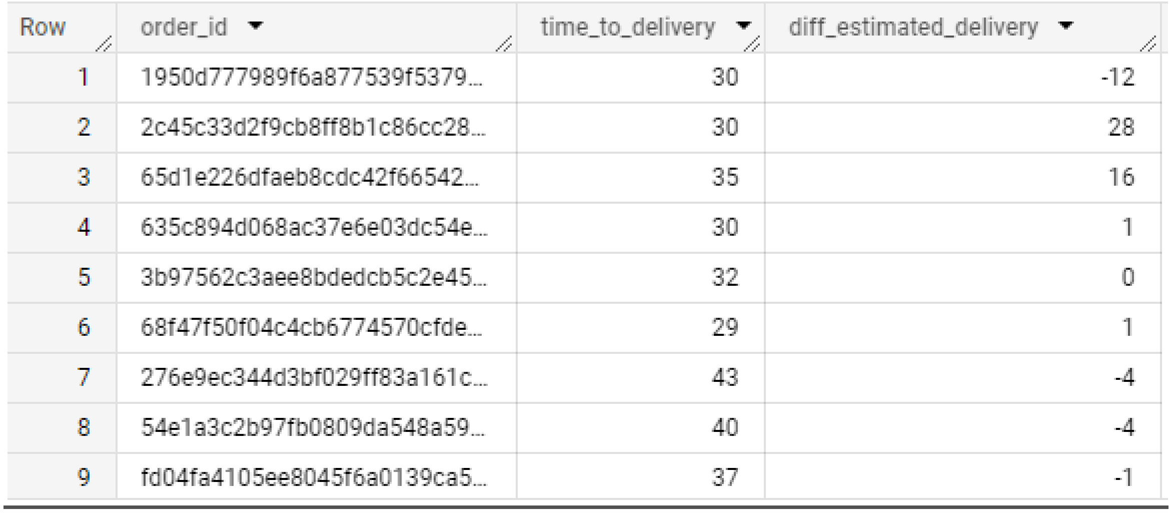
From the above table information about delivery date time ,estimated delivery time, and difference of delivery time has been calculated.

5.2 Find time_to_delivery & diff_estimated_delivery. Formula for the same given below:

- **time_to_delivery = order_delivered_customer_date-order_purchase_timestamp**
- **diff_estimated_delivery = order_estimated_delivery_date-order_delivered_customer_date**

```
select
order_id,
date_diff(order_delivered_customer_date,order_purchase_timestamp,day) as time_to_delivery,
date_diff(order_estimated_delivery_date,order_delivered_customer_date,day) as
diff_estimated_delivery
 from `target-project-388017.Target.Orders`
 where date_diff(order_delivered_customer_date,order_purchase_timestamp,day) is not null
```

Output:

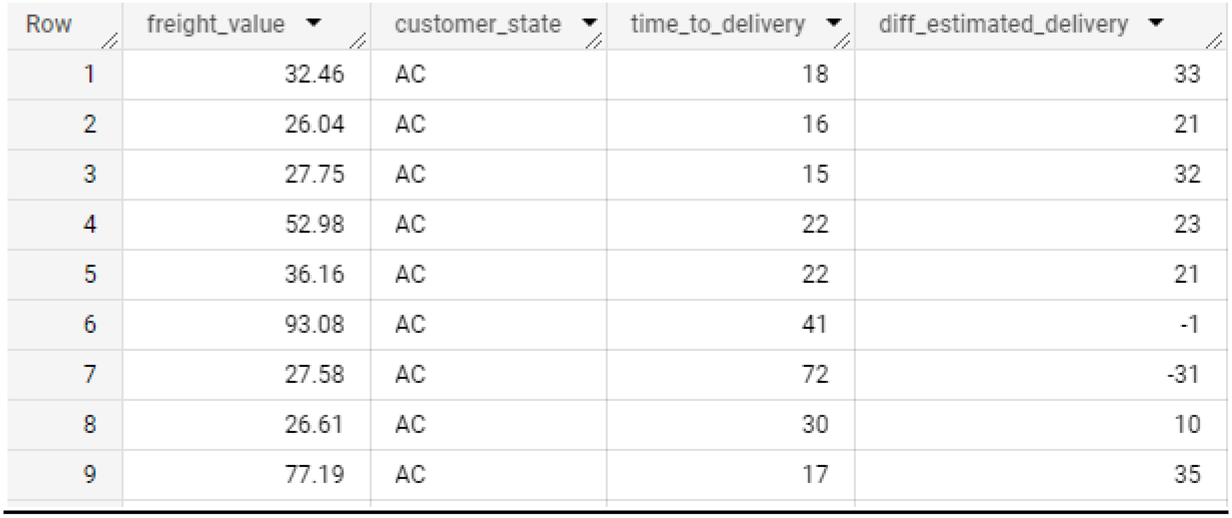


Row	order_id	time_to_delivery	diff_estimated_delivery
1	1950d777989f6a877539f5379...	30	-12
2	2c45c33d2f9cb8ff8b1c86cc28...	30	28
3	65d1e226dfaeb8cdc42f66542...	35	16
4	635c894d068ac37e6e03dc54e...	30	1
5	3b97562c3aee8bdedcb5c2e45...	32	0
6	68f47f50f04c4cb6774570cfde...	29	1
7	276e9ec344d3bf029ff83a161c...	43	-4
8	54e1a3c2b97fb0809da548a59...	40	-4
9	fd04fa4105ee8045f6a0139ca5...	37	-1

5.3 Group data by state, take mean of freight_value, time_to_delivery, diff_estimated_delivery

```
 select
round(avg(freight_value),2) as freight_value,
c.customer_state,
date_diff(order_delivered_customer_date,order_purchase_timestamp,day) as time_to_delivery,
date_diff(order_estimated_delivery_date,order_delivered_customer_date,day) as
diff_estimated_delivery
from `Target.customers` as c
join `target-project-388017.Target.Orders` as o
on c.customer_id=o.customer_id
join `target-project-388017.Target.order_items` as oo
on oo.order_id=o.order_id
 where date_diff(order_delivered_customer_date,order_purchase_timestamp,day) is not null
group by c.customer_state, time_to_delivery, diff_estimated_delivery
 order by c.customer_state
```

Output:



Row	freight_value	customer_state	time_to_delivery	diff_estimated_delivery
1	32.46	AC	18	33
2	26.04	AC	16	21
3	27.75	AC	15	32
4	52.98	AC	22	23
5	36.16	AC	22	21
6	93.08	AC	41	-1
7	27.58	AC	72	-31
8	26.61	AC	30	10
9	77.19	AC	17	35

From the above table we can analyse the average freight value, time of delivery and difference between the delivery time and estimated delivery time. All these are calculated with respect to state.

Positive values in diff_estimated_delivery indicate delivery time before the estimated time.

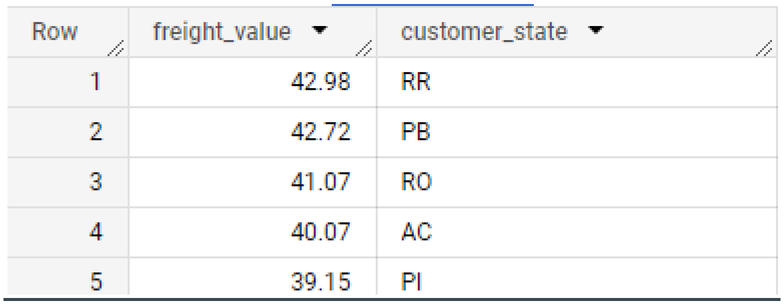
Negative values in diff_estimated_delivery indicate delivery time after the estimated time.

5.4 Sort the data to get the following:

5.5 Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5

```
select
round(avg(freight_value),2) as freight_value,
c.customer_state
from `Target.customers` as c
join `target-project-388017.Target.Orders` as o
on c.customer_id=o.customer_id
join `target-project-388017.Target.order_items` as oo
on oo.order_id=o.order_id
group by c.customer_state
order by freight_value desc
limit 5;
```

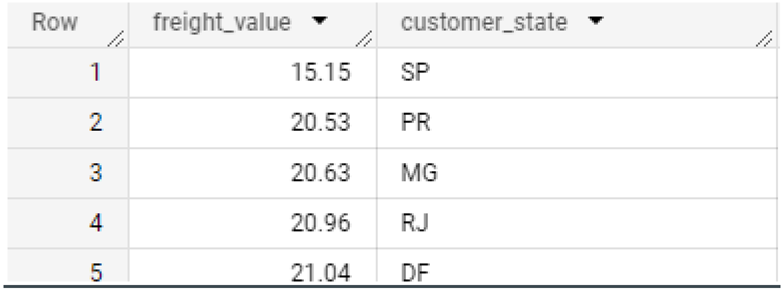
Output:



Row	freight_value	customer_state
1	42.98	RR
2	42.72	PB
3	41.07	RO
4	40.07	AC
5	39.15	PI

```
select
round(avg(freight_value),2) as freight_value,
c.customer_state
from `Target.customers` as c
join `target-project-388017.Target.Orders` as o
on c.customer_id=o.customer_id
join `target-project-388017.Target.order_items` as oo
on oo.order_id=o.order_id
group by c.customer_state
order by freight_value desc asc
limit 5;
```

Output:



Row	freight_value	customer_state
1	15.15	SP
2	20.53	PR
3	20.63	MG
4	20.96	RJ
5	21.04	DF

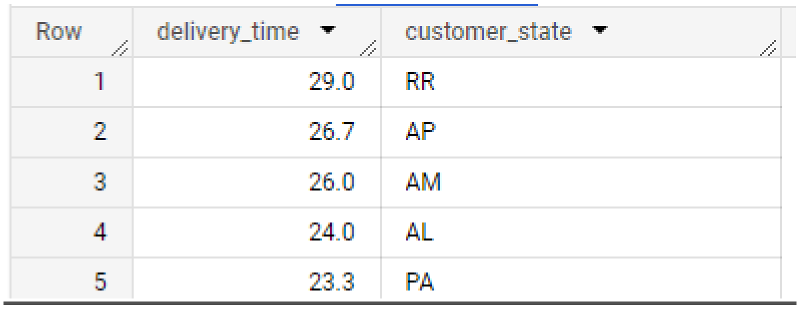
From the above data, average freight value has been calculated. Average freight value is value associated with every person. Also this freight value is associated with every state. We can analyise the variation in freight cost from different states.

5.5 Top 5 states with highest/lowest average time to delivery

```
select
round(avg(date_diff(o.order_delivered_customer_date,o.order_purchase_timestamp,day)),1) as
delivery_time,
c.customer_state
```

```
from `Target.customers` as c
join `target-project-388017.Target.Orders` as o
on c.customer_id=o.customer_id
group by c.customer_state
order by delivery_time desc
limit 5;
```

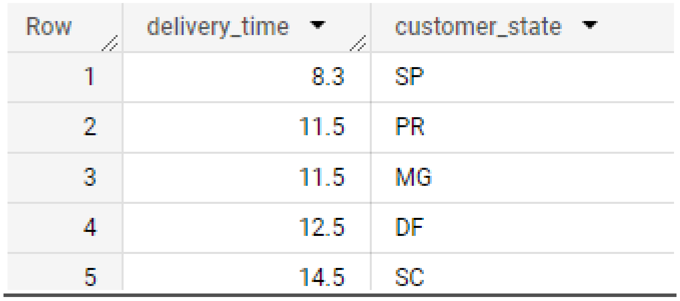
Output:



Row	delivery_time	customer_state
1	29.0	RR
2	26.7	AP
3	26.0	AM
4	24.0	AL
5	23.3	PA

```
select
round(avg(date_diff(o.order_delivered_customer_date,o.order_purchase_timestamp,day)),1) as
delivery_time,
c.customer_state
from `Target.customers` as c
join `target-project-388017.Target.Orders` as o
on c.customer_id=o.customer_id
group by c.customer_state
order by delivery_time asc
limit 5;
```

Output:



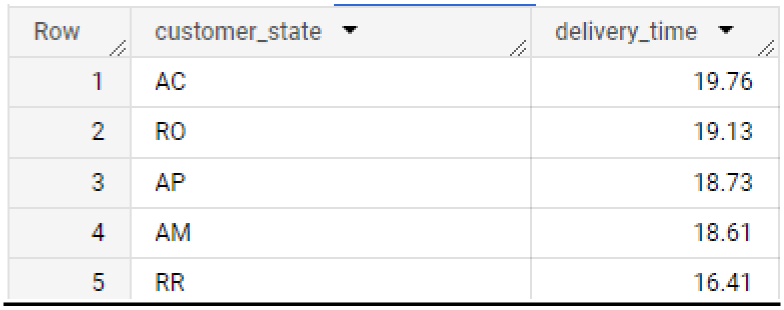
Row	delivery_time	customer_state
1	8.3	SP
2	11.5	PR
3	11.5	MG
4	12.5	DF
5	14.5	SC

From this table, we can check the states with different delivery time.Some states have average delivery time very high and some have average delivery time very less. The major obstacles of late delivery can be eradicated by multiple factors like proactive communication,customer feedback, network optimization, maintain high level services etc.

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

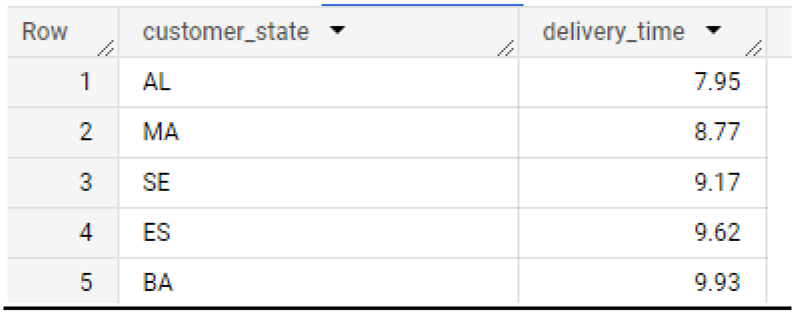
```
select
c.customer_state,
round(avg(date_diff(o.order_estimated_delivery_date,order_delivered_customer_date,day)),2)
as delivery_time
from `Target.customers` as c
join `target-project-388017.Target.Orders` as o
on c.customer_id=o.customer_id
group by c.customer_state
order by delivery_time desc
limit 5;
```

Output:



Row	customer_state	delivery_time
1	AC	19.76
2	RO	19.13
3	AP	18.73
4	AM	18.61
5	RR	16.41

```
select
c.customer_state,
round(avg(date_diff(o.order_estimated_delivery_date,order_delivered_customer_date,day)),2) as
delivery_time
from `Target.customers` as c
join `target-project-388017.Target.Orders` as o
on c.customer_id=o.customer_id
group by c.customer_state
order by delivery_time asc
limit 5;
```



Row	customer_state	delivery_time
1	AL	7.95
2	MA	8.77
3	SE	9.17
4	ES	9.62
5	BA	9.93

From this table we get the difference between the estimated delivery time and how early or how late they delivered an order with respect to state.

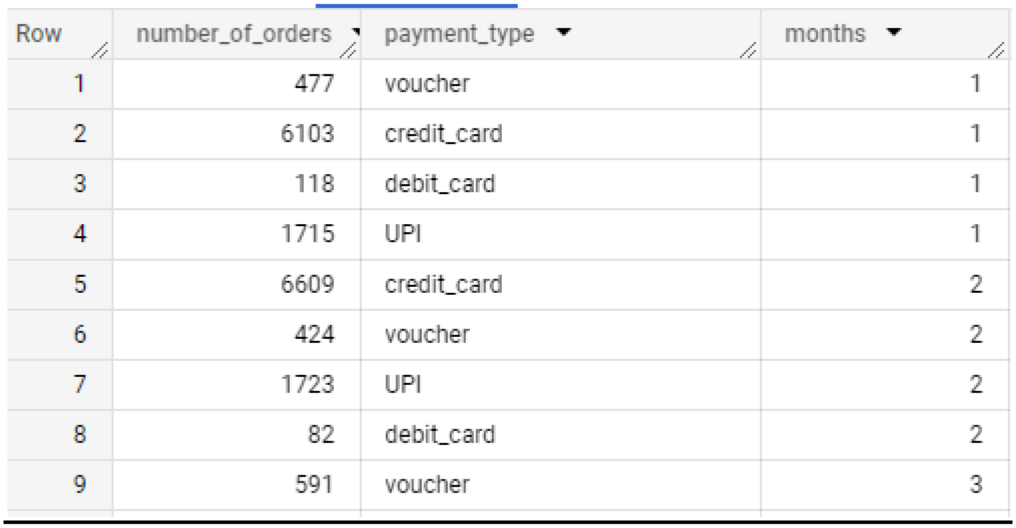
Some states show faster delivery whereas some shows slower delivery. From this relevant steps can be taken to increase the delivery time for the convenience of customers. Challenges such as remote locations,operational ineffeciences can be identified and covered up.

6 Payment type analysis:

6.1 Month over Month count of orders for different payment types

```
select
count(o.order_id) as number_of_orders,
p.payment_type,
extract(month from o.order_purchase_timestamp)  as months
from `target-project-388017.Target.Payments` as p
join `target-project-388017.Target.Orders` as o
on p.order_id = o.order_id
group by months,p.payment_type
order by months
```

Output:



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

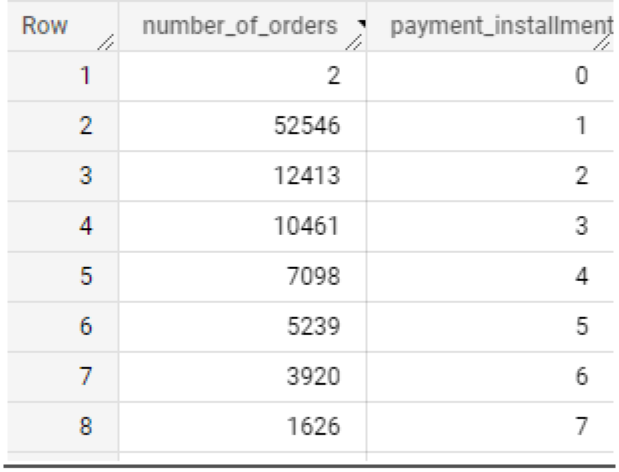
From the above table, we get the payment type of customers they follow.

Most of the people tend to follow credit card for their payment. Very less customers uses debit cards for their payment. We can also get a detailed information month wise for payment type.From this, we can analyse voucher strategies, monitor card usages, and occasional peaks in any payment modes.

6.2 Count of orders based on the no. of payment installments

```
select
count(o.order_id) as number_of_orders,
p.payment_installments
from `target-project-388017.Target.Payments` as p
join `target-project-388017.Target.Orders` as o
on p.order_id = o.order_id
group by p.payment_installments
order by  p.payment_installments
```

Output:



Row	number_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

From the above table, we can understand most people tend to pay in single installment, few like to pay in 2 to 3 installment.

Recommendations:

From the above project, we have analysed that trend of e-commerce is increasing every year. Customers are facing some problem like late deliveries in remote areas, due to network issues etc which can be exterminated by various methods.

However some states are having a very good experience like they have a facility of tracking their orders, change delivery time and dates accordingly. These positive outcomes and happy customers motivate us and their feedback can also a major factor to enlarge our efficiencies.

Positive steps take can be taken are:

- ***Continous monitoring***
- *Customer Satisfaction*
- *Delivery Efficiency*

- *Regional Delivery Performance*
- *Negotiate Shipping rates*
- *Identify Best Practices*
- *Proactive issue resolution*
- *Maintain high-level services*
- *Customer feedback*
- *Discount facility*

***In this performance of Target can be enhanced*.**