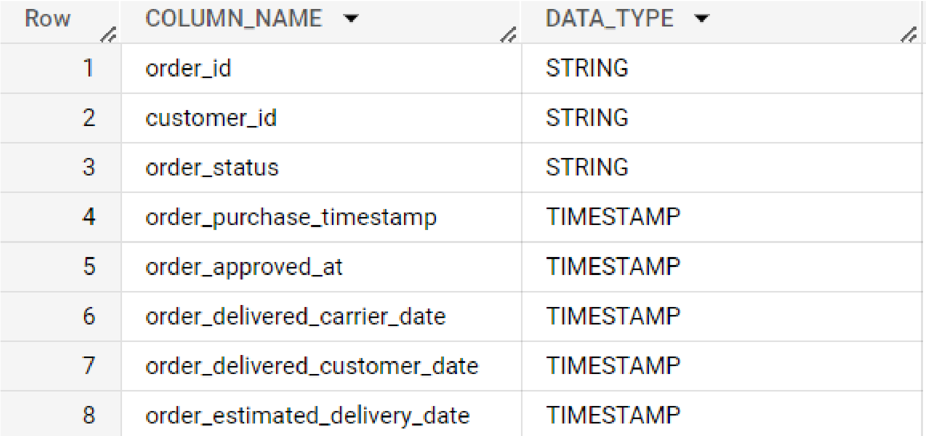
Usual exploratory analysis the structure & characteristics of the dataset

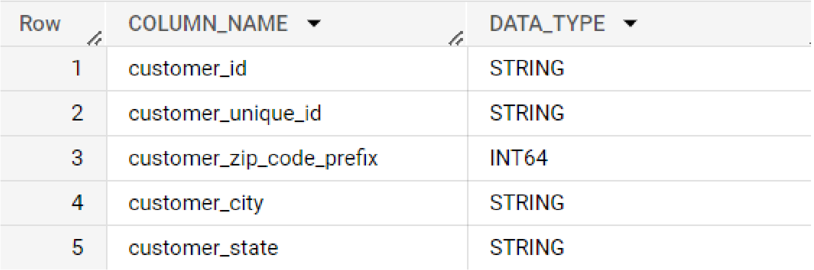
a. Data type of all columns in the "customers" table.

```
SELECT COLUMN_NAME,DATA_TYPE from TARGET_DATASET.INFORMATION_SCHEMA.COLUMNS
where table_name = 'ORDERS'
```



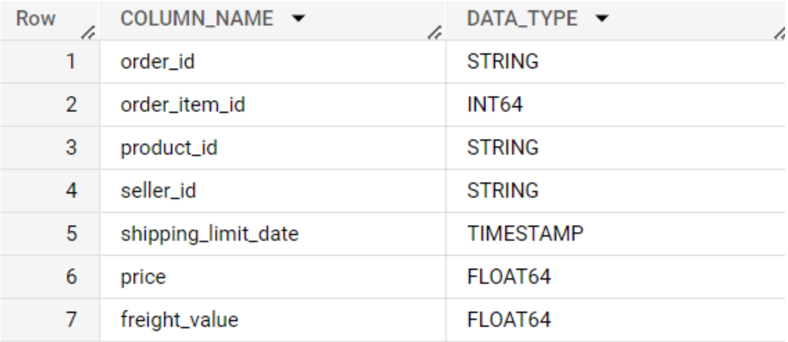
Row	COLUMN_NAME	DATA_TYPE
1	order_id	STRING
2	customer_id	STRING
3	order_status	STRING
4	order_purchase_timestamp	TIMESTAMP
5	order_approved_at	TIMESTAMP
6	order_delivered_carrier_date	TIMESTAMP
7	order_delivered_customer_date	TIMESTAMP
8	order_estimated_delivery_date	TIMESTAMP

```
SELECT COLUMN_NAME,DATA_TYPE from TARGET_DATASET.INFORMATION_SCHEMA.COLUMNS
where table_name = 'CUSTOMERS'
```



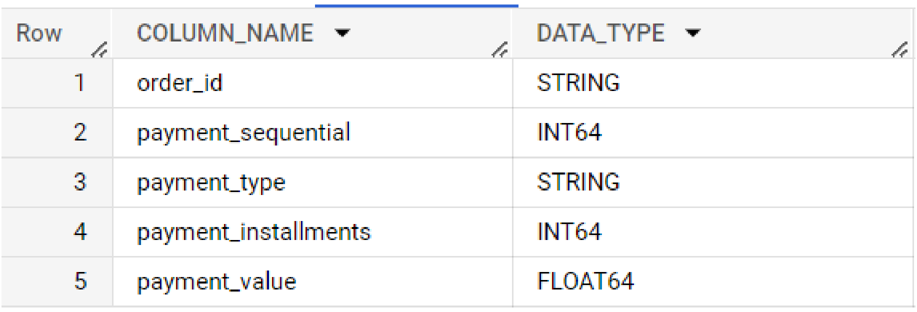
Row	COLUMN_NAME	DATA_TYPE
1	customer_id	STRING
2	customer_unique_id	STRING
3	customer_zip_code_prefix	INT64
4	customer_city	STRING
5	customer_state	STRING

```
SELECT COLUMN_NAME,DATA_TYPE from TARGET_DATASET.INFORMATION_SCHEMA.COLUMNS
where table_name = 'ORDER_ITEMS'
```



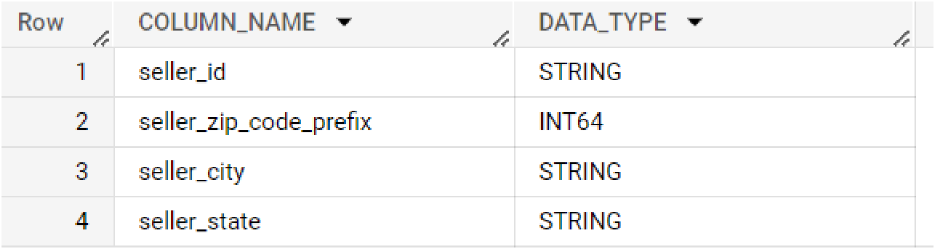
Row	COLUMN_NAME	DATA_TYPE
1	order_id	STRING
2	order_item_id	INT64
3	product_id	STRING
4	seller_id	STRING
5	shipping_limit_date	TIMESTAMP
6	price	FLOAT64
7	freight_value	FLOAT64

```
SELECT COLUMN_NAME,DATA_TYPE from TARGET_DATASET.INFORMATION_SCHEMA.COLUMNS
where table_name = 'PAYMENTS'
```



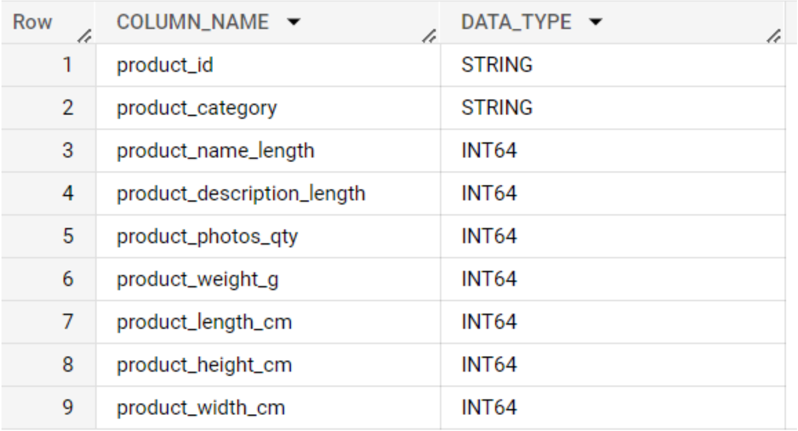
Row	COLUMN_NAME	DATA_TYPE
1	order_id	STRING
2	payment_sequential	INT64
3	payment_type	STRING
4	payment_installments	INT64
5	payment_value	FLOAT64

```
SELECT COLUMN_NAME,DATA_TYPE from TARGET_DATASET.INFORMATION_SCHEMA.COLUMNS
where table_name = 'SELLERS'
```



Row	COLUMN_NAME	DATA_TYPE
1	seller_id	STRING
2	seller_zip_code_prefix	INT64
3	seller_city	STRING
4	seller_state	STRING

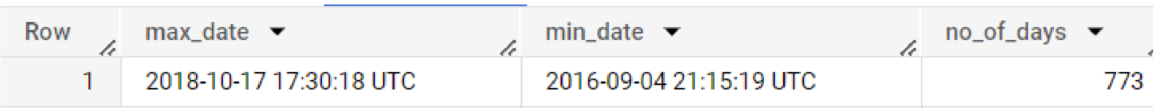
```
SELECT COLUMN_NAME,DATA_TYPE from TARGET_DATASET.INFORMATION_SCHEMA.COLUMNS
where table_name = 'PRODUCTS'
```



Row	COLUMN_NAME	DATA_TYPE
1	product_id	STRING
2	product_category	STRING
3	product_name_length	INT64
4	product_description_length	INT64
5	product_photos_qty	INT64
6	product_weight_g	INT64
7	product_length_cm	INT64
8	product_height_cm	INT64
9	product_width_cm	INT64

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

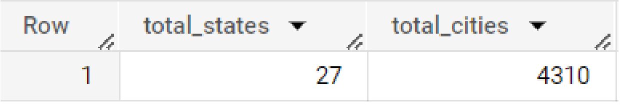
```
SELECT MAX(order_purchase_timestamp) as max_date,min(order_purchase_timestamp)as
min_date,date_diff(MAX(date(order_purchase_timestamp)),min(date(order_purchase_time
stamp)),day) as no_of_days
from TARGET_DATASET.ORDERS;
```



Row	max_date	min_date	no_of_days
1	2018-10-17 17:30:18 UTC	2016-09-04 21:15:19 UTC	773

c. Count the number of Cities and States in our dataset.

```
select count(States) as total_states,sum(no_of_cities) as total_cities
from
(select c.customer_state as States,count(distinct c.customer_city)as no_of_cities,
from TARGET_DATASET.ORDERS o inner join TARGET_DATASET.CUSTOMERS c
on o.customer_id = c.customer_id
group by c.customer_state
order by c.customer_state)a
```

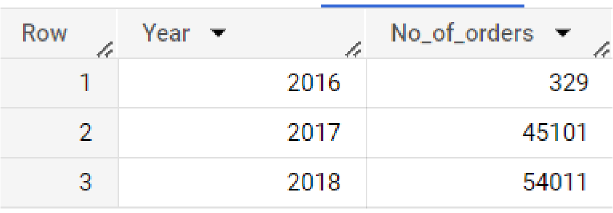


Row	total_states	total_cities
1	27	4310

In-depth Exploration

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

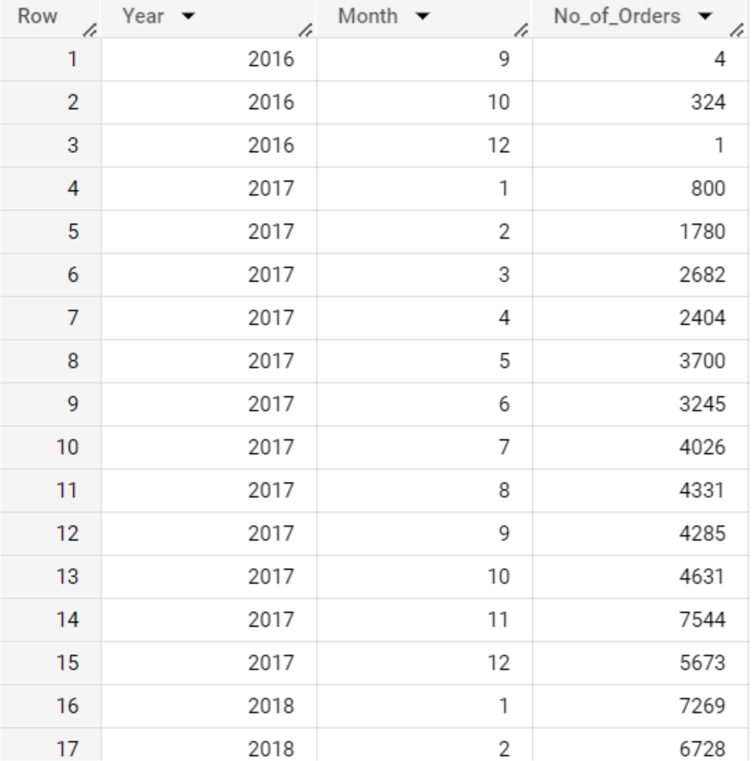
```
select
extract(YEAR from date(order_purchase_timestamp)) as Year,count(order_id) as
No_of_orders
from `TARGET_DATASET.ORDERS`
group by Year
order by Year
```



Row	Year	No_of_orders
1	2016	329
2	2017	45101
3	2018	54011

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

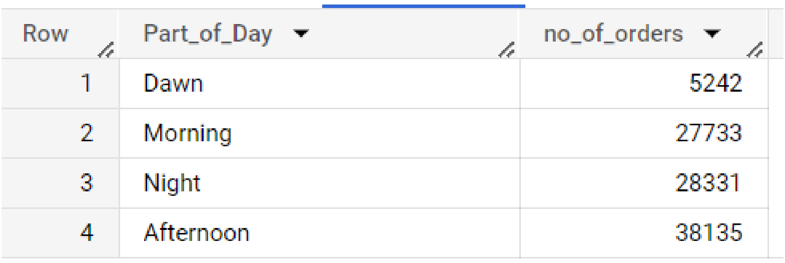
```
with dataset as
(select extract(year from date(order_purchase_timestamp)) as Year, extract(month
from date(order_purchase_timestamp)) as Month,count(distinct order_id) as
No_of_Orders from TARGET_DATASET.ORDERS
group by Year,Month
order by Year, Month)
select * from dataset
```



Row	Year	Month	No_of_Orders
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
16	2018	1	7269
17	2018	2	6728

c. During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)

```
with cte as (
SELECT
(CASE
WHEN hour BETWEEN 0 AND 6 THEN 'Dawn'
WHEN hour BETWEEN 7 AND 12 THEN 'Morning'
WHEN hour BETWEEN 13 AND  18 THEN 'Afternoon'
WHEN hour BETWEEN 19 AND 23 THEN 'Night'
end)as Part_of_Day,order_id from
(SELECT EXTRACT(HOUR FROM TIME (order_purchase_timestamp)) as hour,order_id from
TARGET_DATASET.ORDERS)a)
select Part_of_Day, count(order_id) as no_of_orders from cte
group by Part_of_Day
order by 2
```

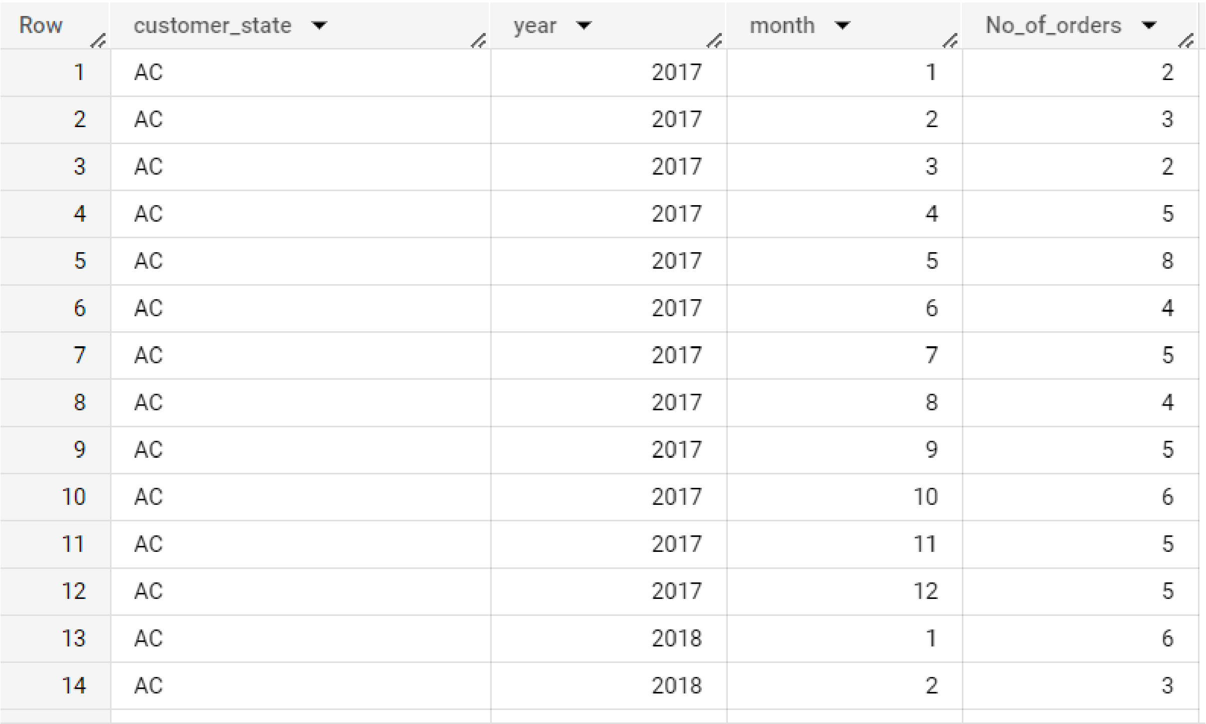


Row	Part_of_Day	no_of_orders
1	Dawn	5242
2	Morning	27733
3	Night	28331
4	Afternoon	38135

Evolution of E-commerce orders in the Brazil region

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

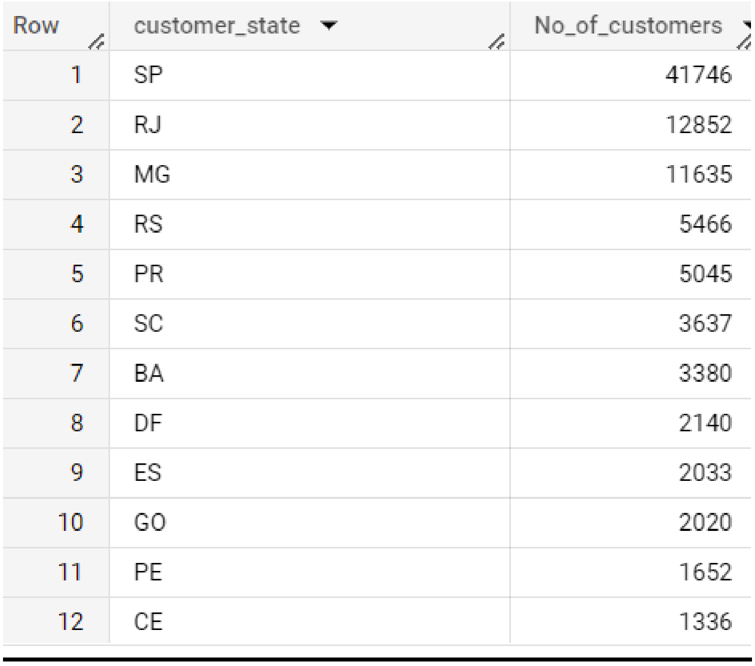
```
select
c.customer_state,
extract (year from date(o.order_purchase_timestamp)) as year,
extract(month from date(o.order_purchase_timestamp)) as month,
count(o.order_id) as No_of_orders
from TARGET_DATASET.ORDERS o
inner join TARGET_DATASET.CUSTOMERS c
on c.customer_id=o.customer_id
group by c.customer_state,year,month
order by c.customer_state,year,month
```



Row	customer_state	year	month	No_of_orders
1	AC	2017	1	2
2	AC	2017	2	3
3	AC	2017	3	2
4	AC	2017	4	5
5	AC	2017	5	8
6	AC	2017	6	4
7	AC	2017	7	5
8	AC	2017	8	4
9	AC	2017	9	5
10	AC	2017	10	6
11	AC	2017	11	5
12	AC	2017	12	5
13	AC	2018	1	6
14	AC	2018	2	3

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

```
select
c.customer_state,
count(distinct c.customer_id) as No_of_customers
from TARGET_DATASET.ORDERS o
inner join TARGET_DATASET.CUSTOMERS c
on c.customer_id=o.customer_id
group by c.customer_state
order by No_of_customers desc
```

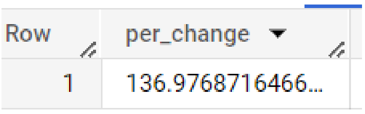


Row	customer_state	No_of_customers
1	SP	41746
2	RJ	12852
3	MG	11635
4	RS	5466
5	PR	5045
6	SC	3637
7	BA	3380
8	DF	2140
9	ES	2033
10	GO	2020
11	PE	1652
12	CE	1336

Impact on Economy

a. Get the % increase in the cost of orders from year 2017 to 2018 (include

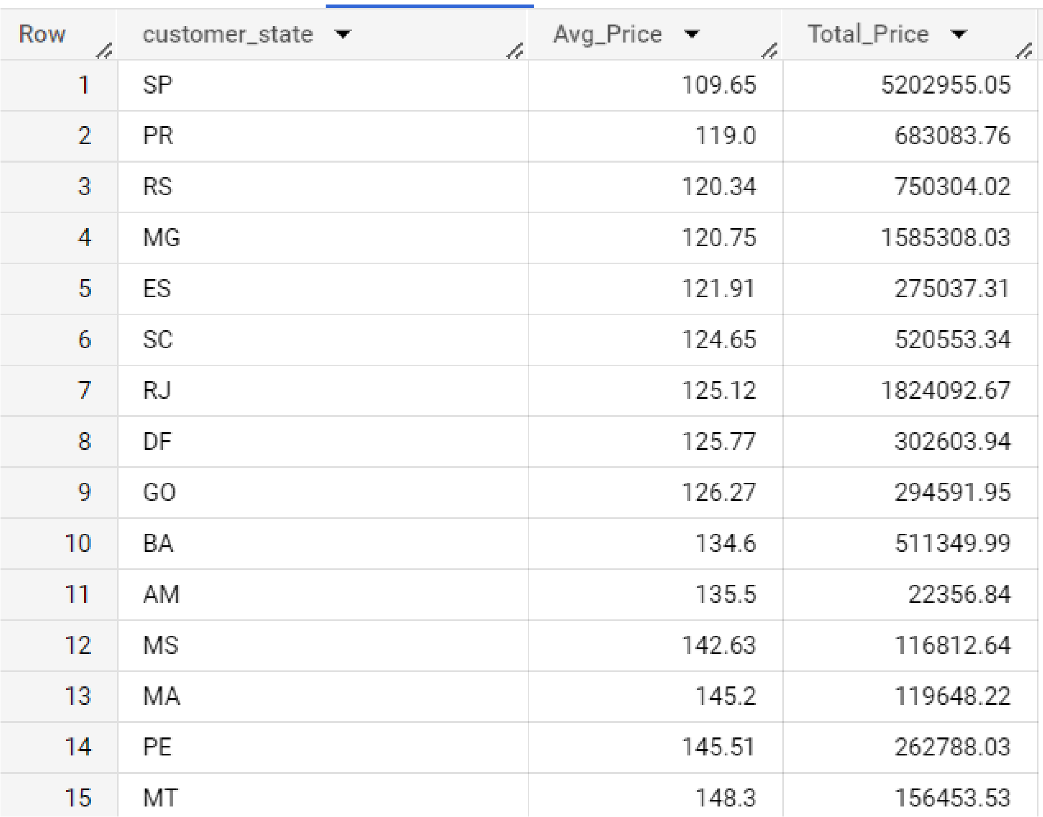
```
 with cte as(

select Year, Month,total_cost from
(select
extract(year from DATE(o.order_purchase_timestamp)) as Year,
extract(month from DATE(o.order_purchase_timestamp)) as Month,
sum(p.payment_value) as total_cost
from TARGET_DATASET.ORDERS o
INNER JOIN TARGET_DATASET.PAYMENTS p
on p.order_id=o.order_id
group by Year,Month
order by Year,Month
)a
where Year between 2017 and 2018
and Month between 1 and 8),
cte1 as
(select year,sum(total_cost) as summ
from cte
group by year
order by year),
cte2 as
(select year, summ ,lag(summ,1) over(order by year) as pv_value
from cte1)
select ((summ-pv_value)/pv_value)*100 as per_change
from cte2
where pv_value is not null
```



Row	per_change
1	136.9768716466...

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

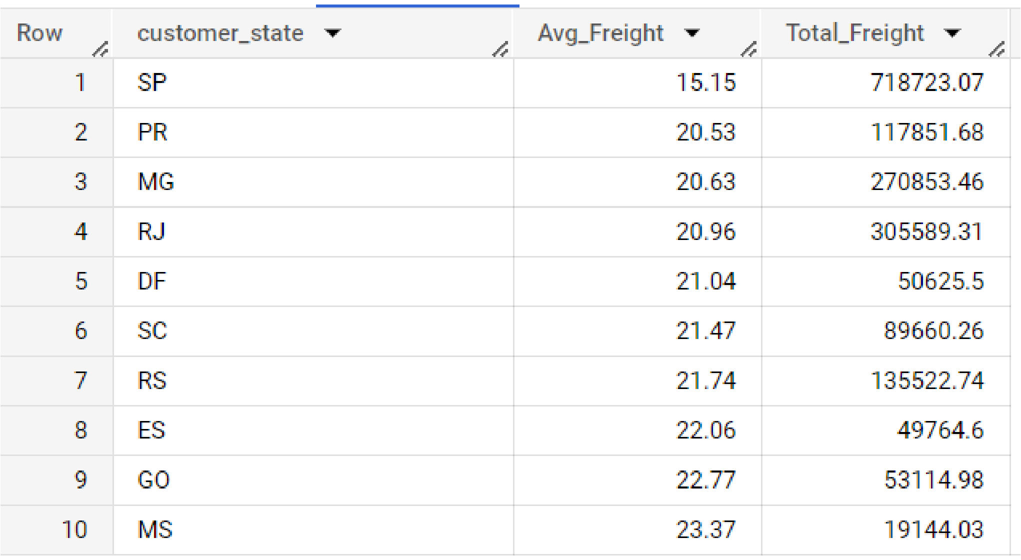
```
select c.customer_state,round(avg(oi.price),2) as Avg_Price,round(sum(oi.price),2)
as Total_Price
from TARGET_DATASET.ORDERS o
inner join
TARGET_DATASET.ORDER_ITEMS oi
on o.order_id=oi.order_id
inner join
TARGET_DATASET.CUSTOMERS c
on o.customer_id=c.customer_id
group by c.customer_state
order by Avg_Price,Total_Price
```



Row	customer_state	Avg_Price	Total_Price
1	SP	109.65	5202955.05
2	PR	119.0	683083.76
3	RS	120.34	750304.02
4	MG	120.75	1585308.03
5	ES	121.91	275037.31
6	SC	124.65	520553.34
7	RJ	125.12	1824092.67
8	DF	125.77	302603.94
9	GO	126.27	294591.95
10	BA	134.6	511349.99
11	AM	135.5	22356.84
12	MS	142.63	116812.64
13	MA	145.2	119648.22
14	PE	145.51	262788.03
15	MT	148.3	156453.53

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

```
select c.customer_state,round(avg(oi.freight_value),2) as
Avg_Freight,round(sum(oi.freight_value),2) as Total_Freight
from TARGET_DATASET.ORDERS o
inner join
TARGET_DATASET.ORDER_ITEMS oi
on o.order_id=oi.order_id
inner join
TARGET_DATASET.CUSTOMERS c
on o.customer_id=c.customer_id
group by c.customer_state
order by Avg_Freight,Total_Freight
limit 10
```

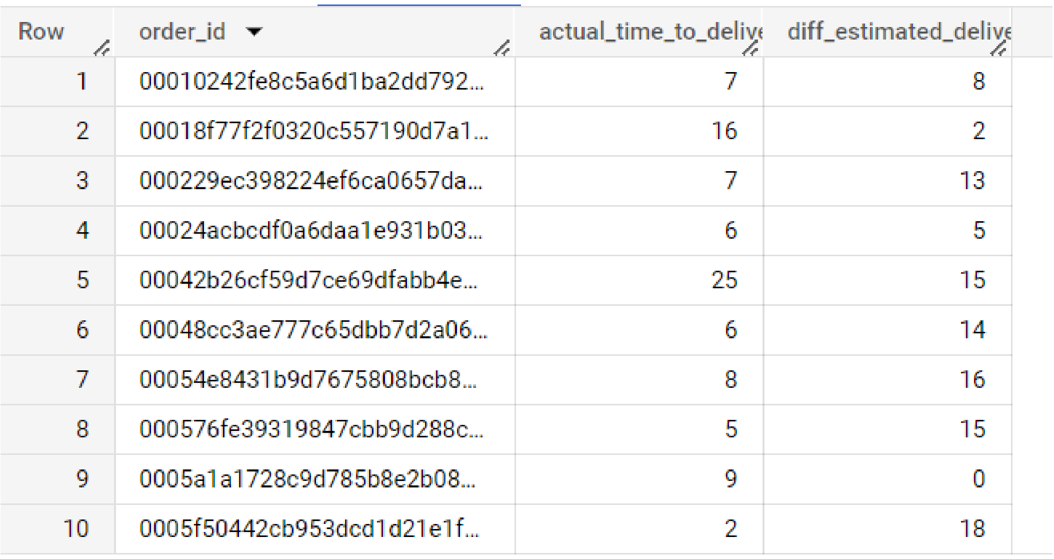


Row	customer_state	Avg_Freight	Total_Freight
1	SP	15.15	718723.07
2	PR	20.53	117851.68
3	MG	20.63	270853.46
4	RJ	20.96	305589.31
5	DF	21.04	50625.5
6	SC	21.47	89660.26
7	RS	21.74	135522.74
8	ES	22.06	49764.6
9	GO	22.77	53114.98
10	MS	23.37	19144.03

Analysis based on sales, freight and delivery time

a. Find the no. of days taken to deliver each order from the order's purchase date as delivery time.

```
select distinct
order_id,DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp,day)
as
actual_time_to_deliver,DATE_DIFF(order_estimated_delivery_date,order_delivered_c
ustomer_date,day) as diff_estimated_delivery
from TARGET_DATASET.ORDERS
order by order_id
limit 10
```

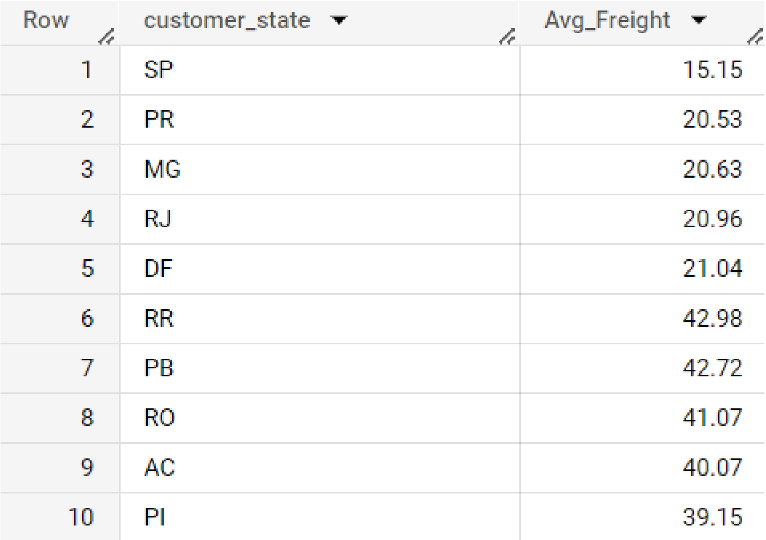


Row	order_id	actual_time_to_delive	diff_estimated_delive
1	00010242fe8c5a6d1ba2dd792...	7	8
2	00018f77f2f0320c557190d7a1...	16	2
3	000229ec398224ef6ca0657da...	7	13
4	00024acbcdf0a6daa1e931b03...	6	5
5	00042b26cf59d7ce69dfabb4e...	25	15
6	00048cc3ae777c65dbb7d2a06...	6	14
7	00054e8431b9d7675808bcb8...	8	16
8	000576fe39319847cbb9d288c...	5	15
9	0005a1a1728c9d785b8e2b08...	9	0
10	0005f50442cb953dcd1d21e1f...	2	18

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

Here first 5 are lowest freight values and next 5 are highest freight value

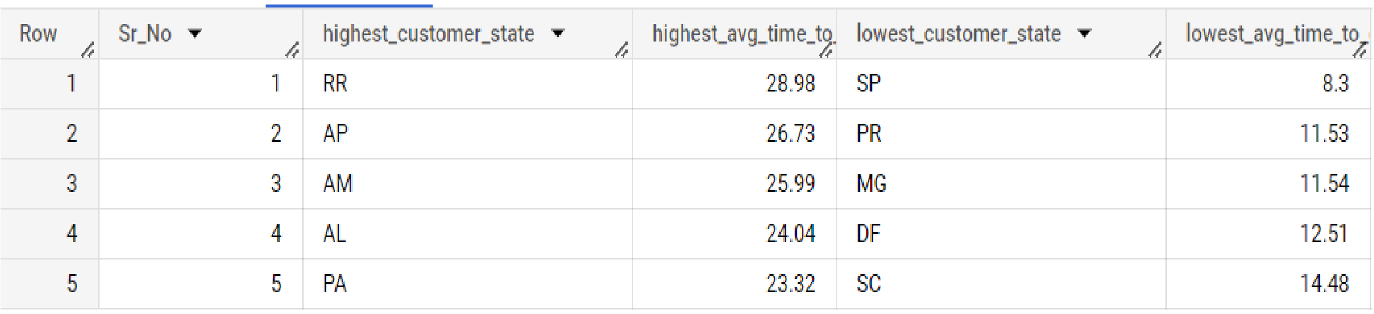
```
(select c.customer_state,round(avg(oi.freight_value),2) as Avg_Freight

from TARGET_DATASET.ORDERS o
inner join
TARGET_DATASET.ORDER_ITEMS oi
on o.order_id=oi.order_id
inner join
TARGET_DATASET.CUSTOMERS c
on o.customer_id=c.customer_id
group by c.customer_state
order by Avg_Freight asc
limit 5)
union all
(select c.customer_state,round(avg(oi.freight_value),2) as Avg_Freight
from TARGET_DATASET.ORDERS o
inner join
TARGET_DATASET.ORDER_ITEMS oi
on o.order_id=oi.order_id
inner join
TARGET_DATASET.CUSTOMERS c
on o.customer_id=c.customer_id
group by c.customer_state
order by Avg_Freight desc
limit 5)
```



Row	customer_state	Avg_Freight
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

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

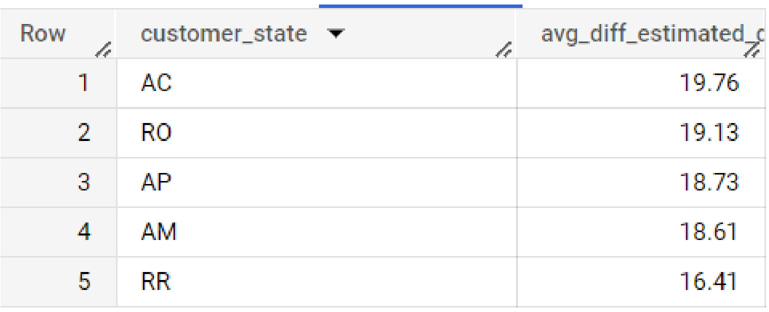
```
with cte_highest as
(select row_number() over(order by highest_avg_time_to_deliver desc) as
Sr_No,highest_customer_state,highest_avg_time_to_deliver
from(
  select * from
(select c.customer_state as
highest_customer_state,round(avg(DATE_DIFF(o.order_delivered_customer_date,o.order_
purchase_timestamp,day)),2) as highest_avg_time_to_deliver
from TARGET_DATASET.ORDERS o
inner join TARGET_DATASET.CUSTOMERS c
on c.customer_id=o.customer_id
group by c.customer_state
)
order by highest_avg_time_to_deliver desc
limit 5)),
cte_lowest as
(select row_number() over(order by lowest_avg_time_to_deliver asc) as
Sr_No,lowest_customer_state,lowest_avg_time_to_deliver
from(
  select * from
(select c.customer_state as
lowest_customer_state,round(avg(DATE_DIFF(o.order_delivered_customer_date,o.order_p
urchase_timestamp,day)),2) as lowest_avg_time_to_deliver
from TARGET_DATASET.ORDERS o
inner join TARGET_DATASET.CUSTOMERS c
on c.customer_id=o.customer_id
group by c.customer_state
)
order by lowest_avg_time_to_deliver asc
limit 5))
select
ch.Sr_No,ch.highest_customer_state,ch.highest_avg_time_to_deliver,cl.lowest_custome
r_state,cl.lowest_avg_time_to_deliver
from cte_highest ch
inner join cte_lowest cl
on ch.Sr_No= cl.Sr_No
```



Row	Sr_No ▼	highest_customer_state ▼	highest_avg_time_to	lowest_customer_state ▼	lowest_avg_time_to
1	1	RR	28.98	SP	8.3
2	2	AP	26.73	PR	11.53
3	3	AM	25.99	MG	11.54
4	4	AL	24.04	DF	12.51
5	5	PA	23.32	SC	14.48

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

```
select * from
(select c.customer_state,
round(avg(DATE_DIFF(order_estimated_delivery_date,order_delivered_customer_date,day
)),2) as avg_diff_estimated_delivery
from TARGET_DATASET.ORDERS o
inner join TARGET_DATASET.CUSTOMERS c
on c.customer_id=o.customer_id
group by c.customer_state
)
order by avg_diff_estimated_delivery  desc
limit 5
```

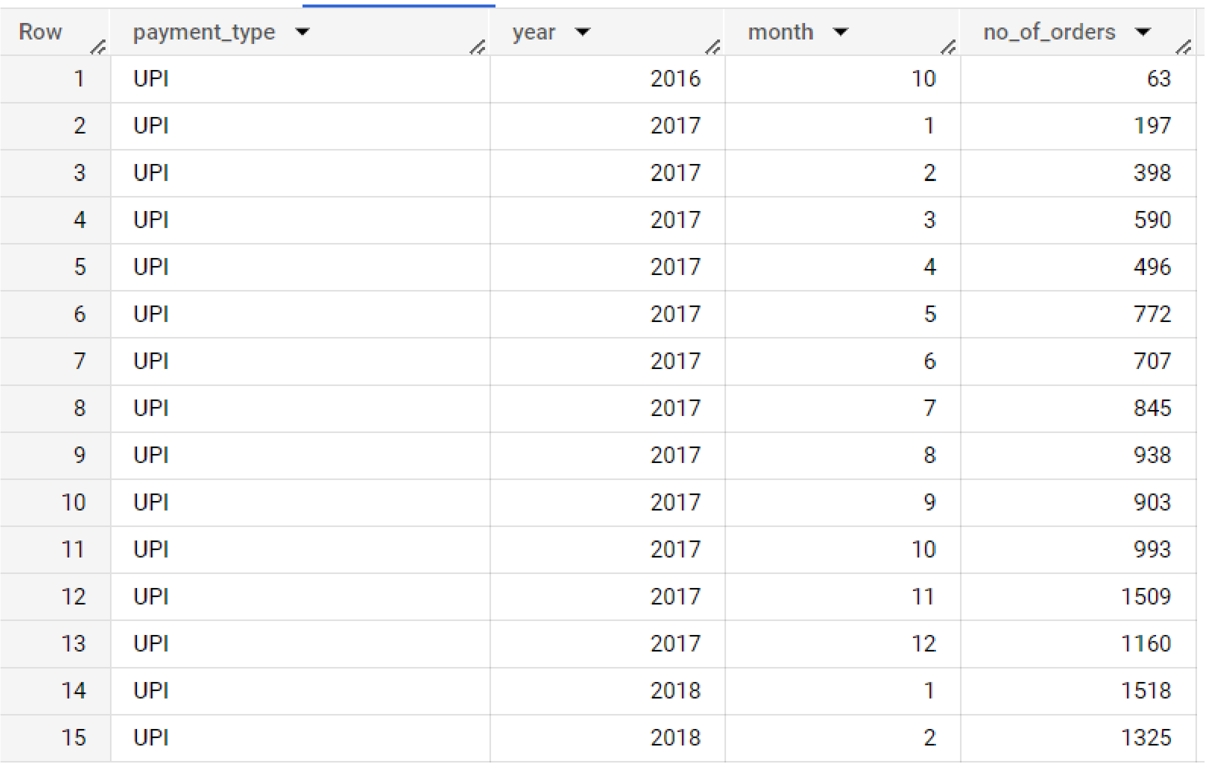


Row	customer_state	avg_diff_estimated_c
1	AC	19.76
2	RO	19.13
3	AP	18.73
4	AM	18.61
5	RR	16.41

Analysis based on the payments

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

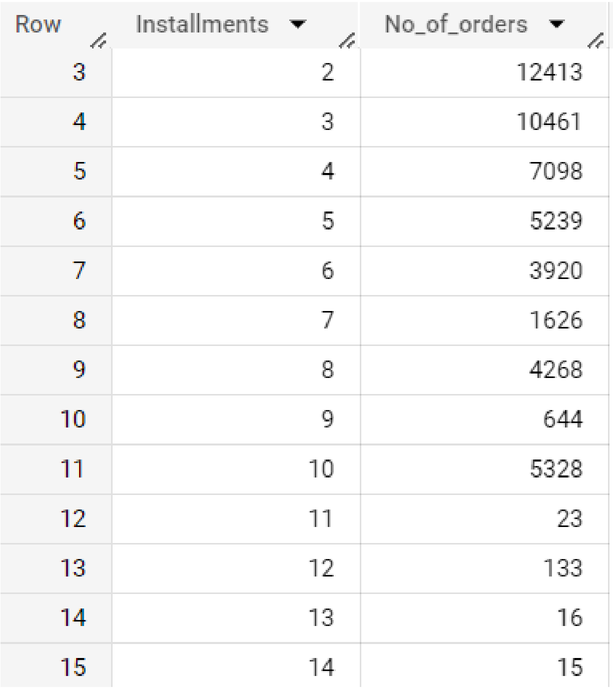
```
select p.payment_type,
extract(year from date(o.order_purchase_timestamp)) as year,
extract(month from date(o.order_purchase_timestamp)) as month,
count(o.order_id) as no_of_orders
from
TARGET_DATASET.ORDERS o
INNER JOIN TARGET_DATASET.PAYMENTS p
ON o.order_id=p.order_id
group by p.payment_type,year,month
order by p.payment_type,year,month
limit 15
```



Row	payment_type	year	month	no_of_orders
1	UPI	2016	10	63
2	UPI	2017	1	197
3	UPI	2017	2	398
4	UPI	2017	3	590
5	UPI	2017	4	496
6	UPI	2017	5	772
7	UPI	2017	6	707
8	UPI	2017	7	845
9	UPI	2017	8	938
10	UPI	2017	9	903
11	UPI	2017	10	993
12	UPI	2017	11	1509
13	UPI	2017	12	1160
14	UPI	2018	1	1518
15	UPI	2018	2	1325

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

```
SELECT * FROM
(select  p.payment_installments as Installments,count(o.order_id) as No_of_orders
from
TARGET_DATASET.ORDERS o
inner join
TARGET_DATASET.PAYMENTS p
on o.order_id = p.order_id
group by Installments
order by Installments)
limit 15
```



Row	Installments	No_of_orders
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
13	12	133
14	13	16
15	14	15

Actionable Insights & Recommendations

Insights

Checking the structure & characteristics of the dataset

1. It is visible that all the columns are having the right datatype and there is no mismatch
2. The orders table tells us that the spread is over **773 days .**
3. IN totality there are 27 states and 4310 cities in which **TARGET** operates
4. **IN Minas Gerais-MG ,** the state has highest no of cities &
   IN SAO-PAULO – SP , the state has second highest no of cities , very important regions

In-depth Exploration

1. As we can see there is a growing trend of no of order placed **YEAR OVER YEAR**
2. IN 2017 , THE NO OF ORDERS were increasing in the beginning till MAY, THEN TOOK A DIP
3. There was a peak in **NOVEMBER,2017**
4. Usually as visible , the orders **increases continuously** in **JUNE , JULY , AUGUST** in both the years 2017 and 2018
5. Also as per the data, most Brazilians buy/order at **AFTERNOON** followed by **NIGHT**

Evolution of E-commerce orders

1. Most our customers come from **SP > RJ > MG so these are** TOP PRIORITY STATES

Impact on Economy

1. The percentage_cost of orders increased 136 percent in 2018 over 2017 ,which is good for growth.
2. It is visible that lowest avg price is in **SP,** followed by **PR**
3. It is visible that highest total price is In **SP** , followed by **RJ**
4. It is visible that lowest avg freight is in **SP,** followed by **PR .**
5. It is visible that highest total freight is In **SP,** followed by **RJ**

Analysis based on sales, freight and delivery time

1. State **RR** has the highest avg freight value and state **SP** has the lowest freight value .
2. One can see that states with lowest avg freight values also have less avg delivery time like **SP, MG , PR.**
3. One can also see that state with highest avg freight value i.e. **RR ,** has highest avg delivery time

4. AC, RO, AP AM are the states where delivery is faster than estimated delivery time as in these states deliveries are done in way advance of estimated delivery dates.

Analysis based on the payments

1. In Case of UPI , the orders are almost increasing every month in 2017 until November
2. For credit card, the orders increases continuously in months of June to November in 2017
3. IT is clear that as instalments increase no of orders decrease , and there are more orders when instalment ranges from 1-3 .

Recommendations

1. Give more credit card offers on sales in the months of June to November
2. Keep 2-4 instalments and give more offers on products thereby increasing sales and revenue
3. Bringing down the freight will definitely increase the total sales.
4. Try to create ad campaigns wherein messages, promotional mails are sent at afternoon time
5. Try to give more discounts and consolidate the position in states **SP , RJ , MG as these are our most important states RR, AP , AM , AL, PA** as reduced delivery time leads to more orders and revenue
6. Similarly reducing average freight in states like **RR, PB RO , A C , PI** will lead to more revenue
7. In states with low average price, more revenue is generated , therefore we should work on bringing low priced items to increase sales in states lagging behind .