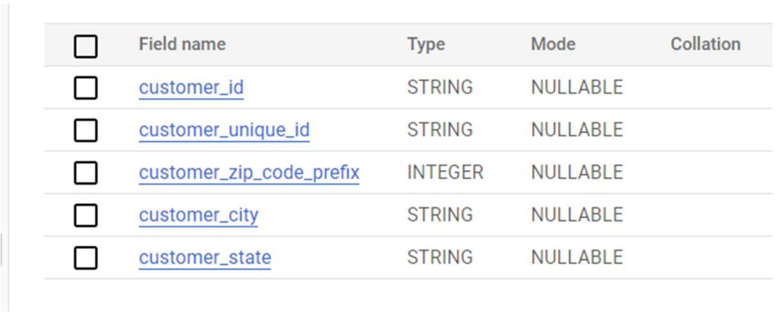
1. Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset

1. The structure & Datatype of the dataset:

- **Customers table**

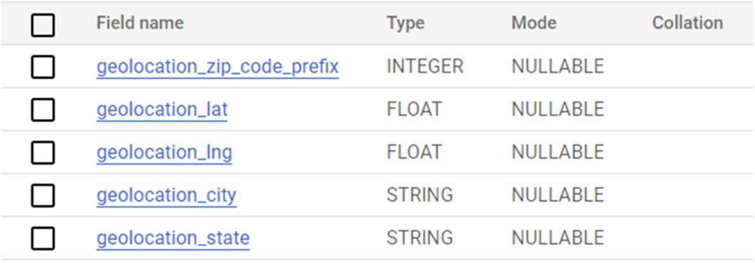
Structure and Datatype



□	Field name	Type	Mode	Collation
□	customer_id	STRING	NULLABLE	
□	customer_unique_id	STRING	NULLABLE	
□	customer_zip_code_prefix	INTEGER	NULLABLE	
□	customer_city	STRING	NULLABLE	
□	customer_state	STRING	NULLABLE	

Geolocation table

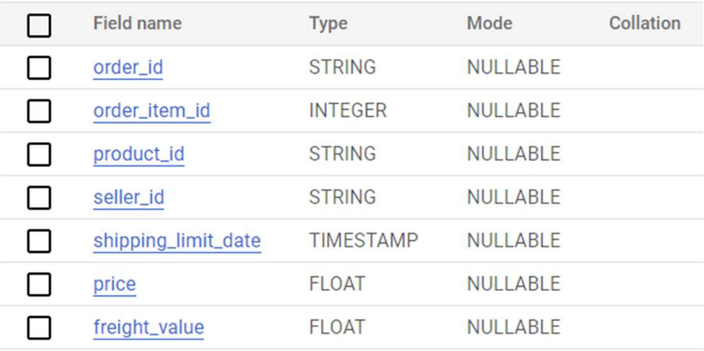
Structure and Datatype



□	Field name	Type	Mode	Collation
□	geolocation_zip_code_prefix	INTEGER	NULLABLE	
□	geolocation_lat	FLOAT	NULLABLE	
□	geolocation_lng	FLOAT	NULLABLE	
□	geolocation_city	STRING	NULLABLE	
□	geolocation_state	STRING	NULLABLE	

- **Order_items table**

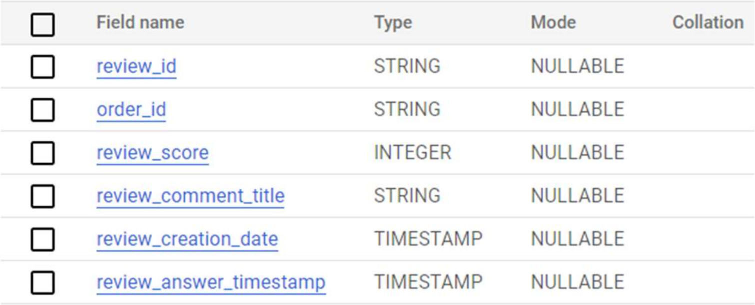
Structure and Datatype



□	Field name	Type	Mode	Collation
□	order_id	STRING	NULLABLE	
□	order_item_id	INTEGER	NULLABLE	
□	product_id	STRING	NULLABLE	
□	seller_id	STRING	NULLABLE	
□	shipping_limit_date	TIMESTAMP	NULLABLE	
□	price	FLOAT	NULLABLE	
□	freight_value	FLOAT	NULLABLE	

Order_reviews table

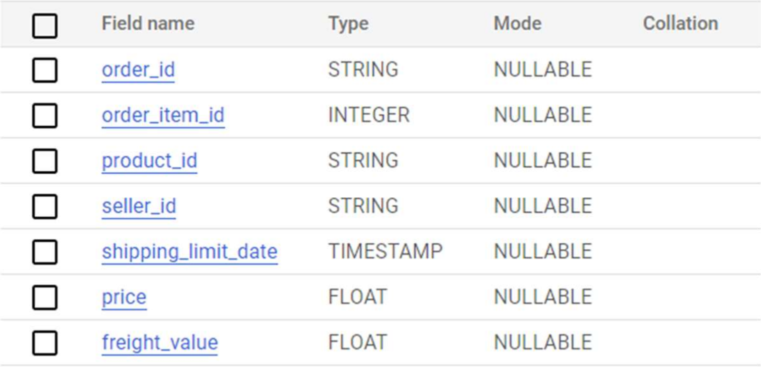
Structure and Datatype



□	Field name	Type	Mode	Collation
□	review_id	STRING	NULLABLE	
□	order_id	STRING	NULLABLE	
□	review_score	INTEGER	NULLABLE	
□	review_comment_title	STRING	NULLABLE	
□	review_creation_date	TIMESTAMP	NULLABLE	
□	review_answer_timestamp	TIMESTAMP	NULLABLE	

- ## Orders table

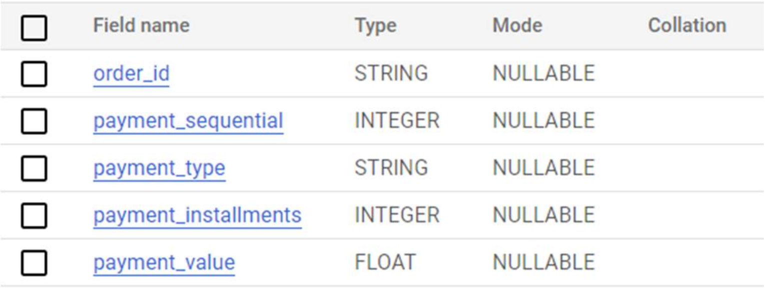
Structure and Datatype



□	Field name	Type	Mode	Collation
□	order_id	STRING	NULLABLE	
□	order_item_id	INTEGER	NULLABLE	
□	product_id	STRING	NULLABLE	
□	seller_id	STRING	NULLABLE	
□	shipping_limit_date	TIMESTAMP	NULLABLE	
□	price	FLOAT	NULLABLE	
□	freight_value	FLOAT	NULLABLE	

- ## Payments table

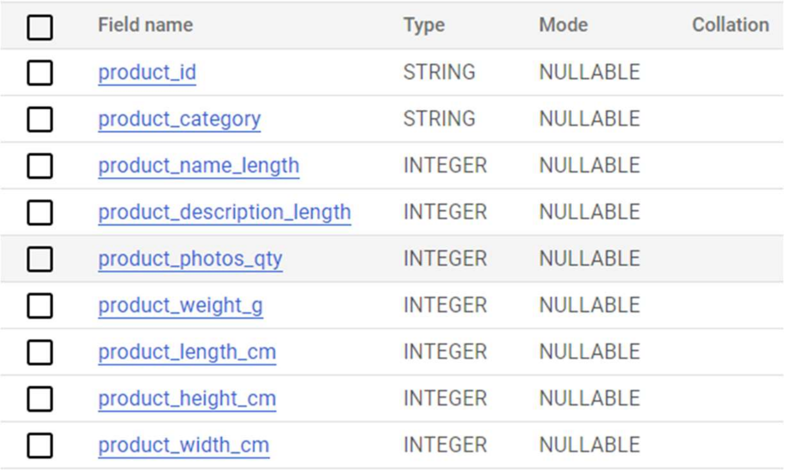
Structure and Datatype



□	Field name	Type	Mode	Collation
□	order_id	STRING	NULLABLE	
□	payment_sequential	INTEGER	NULLABLE	
□	payment_type	STRING	NULLABLE	
□	payment_installments	INTEGER	NULLABLE	
□	payment_value	FLOAT	NULLABLE	

- ## Products table

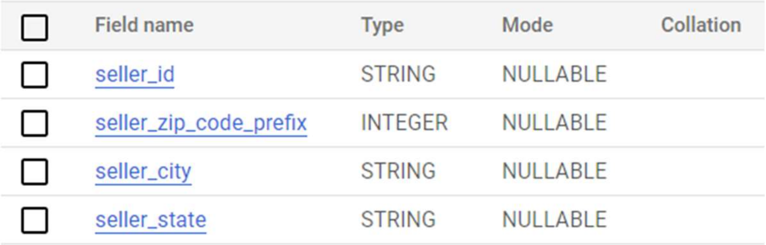
Structure and Datatype



□	Field name	Type	Mode	Collation
□	product_id	STRING	NULLABLE	
□	product_category	STRING	NULLABLE	
□	product_name_length	INTEGER	NULLABLE	
□	product_description_length	INTEGER	NULLABLE	
□	product_photos_qty	INTEGER	NULLABLE	
□	product_weight_g	INTEGER	NULLABLE	
□	product_length_cm	INTEGER	NULLABLE	
□	product_height_cm	INTEGER	NULLABLE	
□	product_width_cm	INTEGER	NULLABLE	

- ## Sellers table

Structure and Datatype

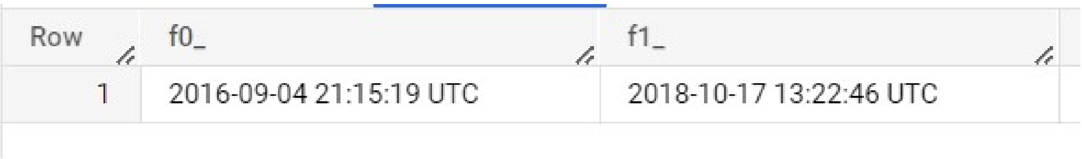


□	Field name	Type	Mode	Collation
□	seller_id	STRING	NULLABLE	
□	seller_zip_code_prefix	INTEGER	NULLABLE	
□	seller_city	STRING	NULLABLE	
□	seller_state	STRING	NULLABLE	

The time period for which the data is given, assuming there is a column in the dataset with a timestamp:

Query -

```
SELECT min(order_purchase_timestamp), max(order_delivered_customer_date)

FROM target-380817.Target123.orders;
```

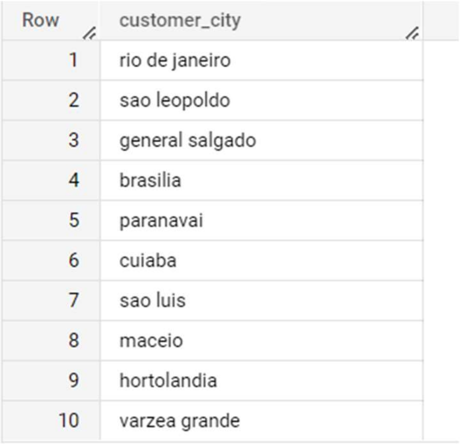


Row	f0_	f1_
1	2016-09-04 21:15:19 UTC	2018-10-17 13:22:46 UTC

The cities and states of customers who ordered during the given period:

Query(To see cities) - `SELECT distinct customer_city`

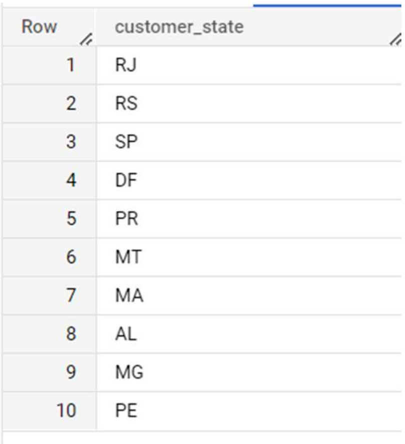
```
FROM target-380817.Target123.Customers as c inner join
 target-380817.Target123.orders as o
on c.customer_id = o.customer_id
where o.order_purchase_timestamp between '2016-09-04 21:15:19 UTC'and '2018-10-17 13:22:46 UTC'
```



Row	customer_city
1	rio de janeiro
2	sao leopoldo
3	general salgado
4	brasilia
5	paranavai
6	cuiaba
7	sao luis
8	maceio
9	hortolandia
10	varzea grande

Query (To see states) -

```
SELECT distinct customer_state
 FROM target-380817.Target123.Customers as c inner join target-380817.Target123.orders as o
on c.customer_id = o.customer_id
where o.order_purchase_timestamp between '2016-09-04 21:15:19 UTC'and '2018-10-17 13:22:46 UTC';
```



Row	customer_state
1	RJ
2	RS
3	SP
4	DF
5	PR
6	MT
7	MA
8	AL
9	MG
10	PE

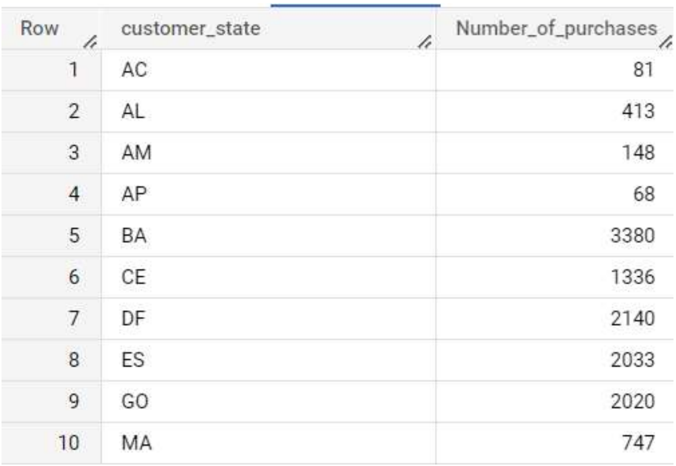
This command will provide us with a list of all unique city and states for customers who placed orders during the given time period.

2. In-depth Exploration:

Is there a growing trend on e-commerce in Brazil?

Query-

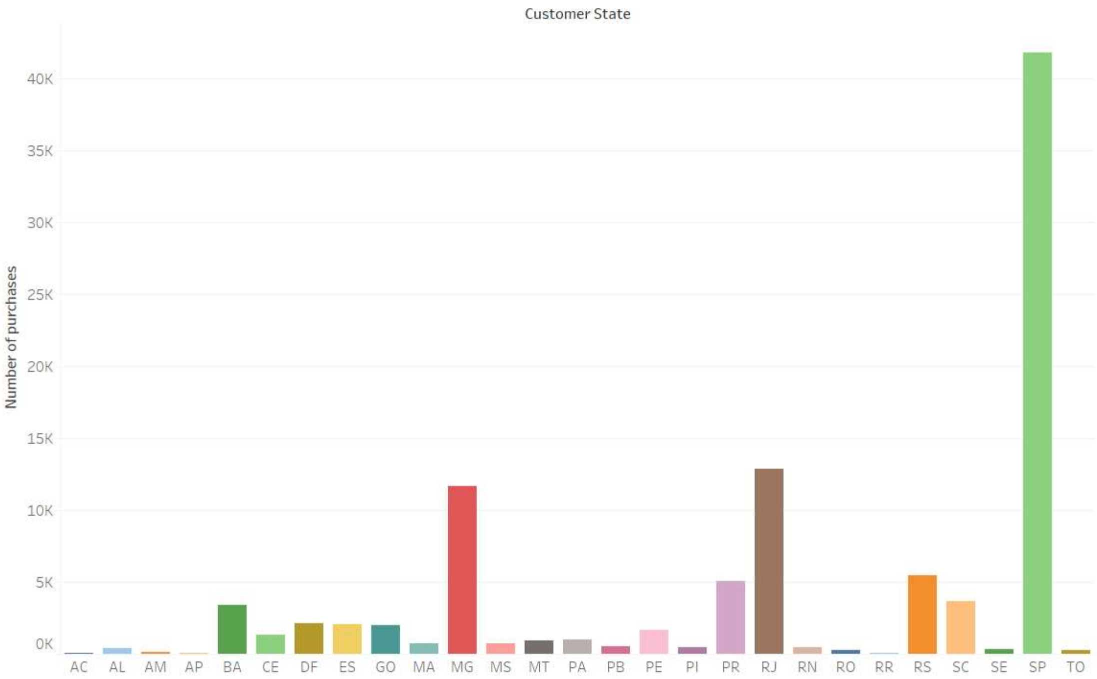
```
SELECT customer_state,count (customer_id) as  Number_of_purchases

FROM target-380817.Target123.Customers
group by customer_state
order by customer_state;
```



Row	customer_state	Number_of_purchases
1	AC	81
2	AL	413
3	AM	148
4	AP	68
5	BA	3380
6	CE	1336
7	DF	2140
8	ES	2033
9	GO	2020
10	MA	747

;

This shows us the purchases made from different states of Brazil.

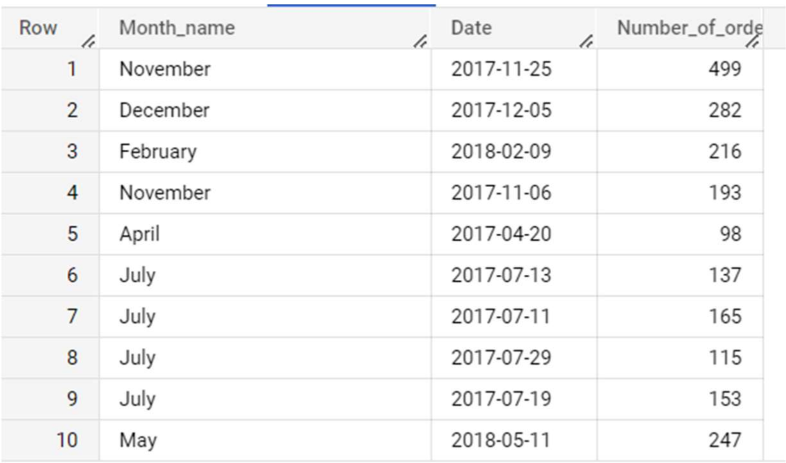


Hence, we can clearly see that most of the purchases are made from the most populous city of Brazil i.e **Sao Paulo (SP)**. In SA we can see an increasing trend of eCommerce. It seems like there is not much growing trend of eCommerce in most states but states like **Minas Gerais(MG)**, **Paraná(PR)**, **Rio de Janeiro(RJ)**, **Rio Grande do Sul (RS)**, Santa Catarina (SC) are some states making more purchases as compared to other states.

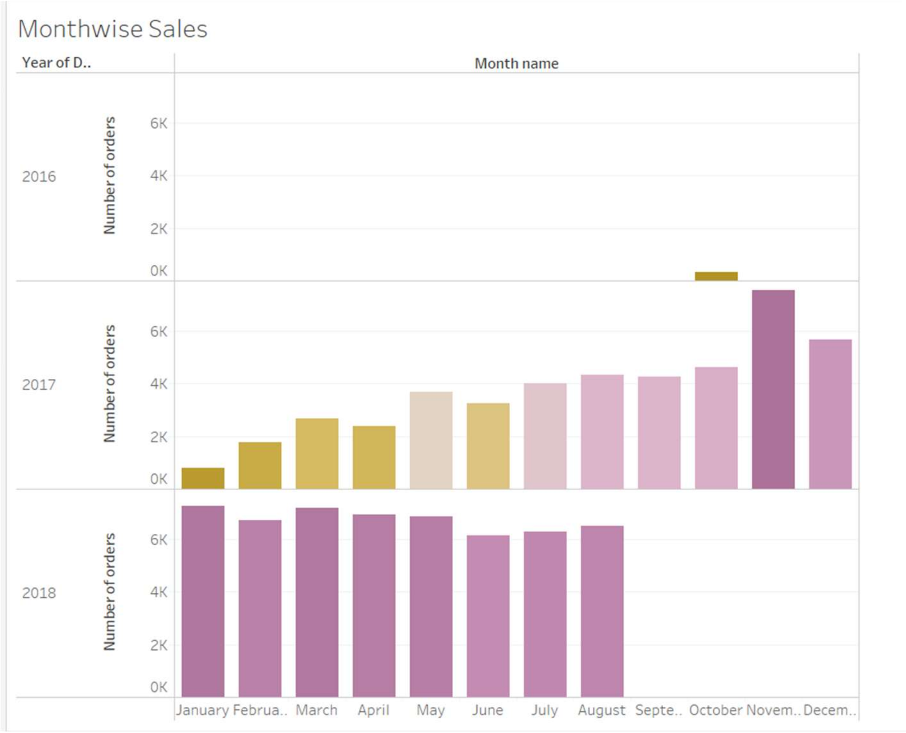
Can we see some seasonality with peaks at specific months?

Query -

```
Select FORMAT_DATE('%B', order_purchase_timestamp) AS Month_name

 ,extract( date from order_purchase_timestamp) AS Date ,count(order_id) as Nu
mber_of_orders from target-380817.Target123.orders
  group by  FORMAT_DATE('%B', order_purchase_timestamp), extract( date from o
rder_purchase_timestamp);
```



Row	Month_name	Date	Number_of_orde
1	November	2017-11-25	499
2	December	2017-12-05	282
3	February	2018-02-09	216
4	November	2017-11-06	193
5	April	2017-04-20	98
6	July	2017-07-13	137
7	July	2017-07-11	165
8	July	2017-07-29	115
9	July	2017-07-19	153
10	May	2018-05-11	247



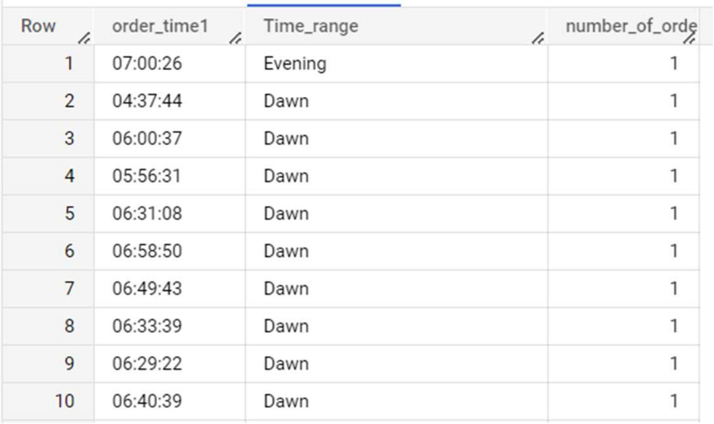
We can see that in both 2017 and 2018 Sales increased from February to March,
Sales declined from May to June and Sales go up again from June to July and July to August.

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

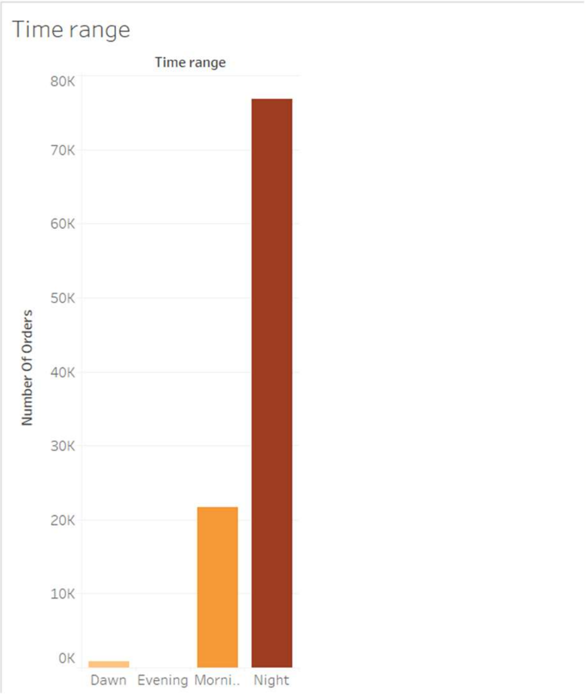
Query-

```
Select extract(time from order_purchase_timestamp) as order_time1, case

 when extract(time from order_purchase_timestamp) between "04:00:00" and "07:
00:00" then "Dawn"
 when extract(time from order_purchase_timestamp) between "07:01:00" and "12:
00:00" then "Morning"
 when extract(time from order_purchase_timestamp) between "12:01:00" and "04:
00:00" then "Afternoon"
```

```
 when extract(time from order_purchase_timestamp) between "04:01:00" and "08:
00:00" then "Evening"
 else "Night"
 End as Time_range
 ,count(order_id) as number_of_orders from target-380817.Target123.orders
 group by  order_purchase_timestamp ;
```



Row	order_time1	Time_range	number_of_orde
1	07:00:26	Evening	1
2	04:37:44	Dawn	1
3	06:00:37	Dawn	1
4	05:56:31	Dawn	1
5	06:31:08	Dawn	1
6	06:58:50	Dawn	1
7	06:49:43	Dawn	1
8	06:33:39	Dawn	1
9	06:29:22	Dawn	1
10	06:40:39	Dawn	1



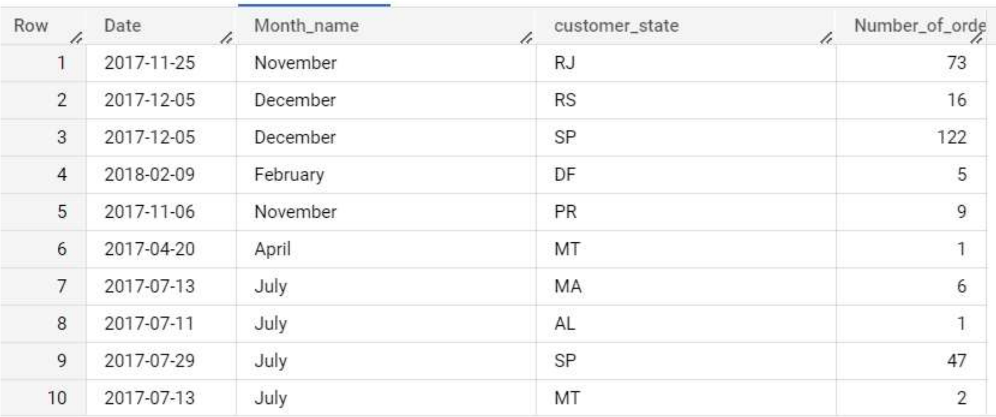
We can analyze that mostly purchases are made during Night.

3 Evolution of E-commerce orders in the Brazil region

1. Get month on month orders by states

Query -

```
Select extract( date from order_purchase_timestamp) AS Date,FORMAT_DATE('%B',
 order_purchase_timestamp) AS Month_name

 ,c.customer_state ,count(order_id) as Number_of_orders
  from target-380817.Target123.orders as o inner join target-
380817.Target123.Customers as c
  on o.customer_id= c.customer_id
  group by  FORMAT_DATE('%B', order_purchase_timestamp), extract( date from o
rder_purchase_timestamp), c.customer_state;
```

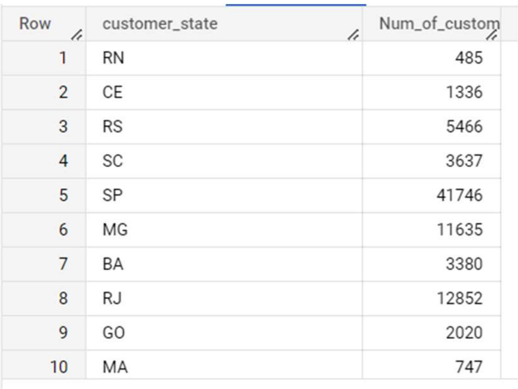


Row	Date	Month_name	customer_state	Number_of_orde
1	2017-11-25	November	RJ	73
2	2017-12-05	December	RS	16
3	2017-12-05	December	SP	122
4	2018-02-09	February	DF	5
5	2017-11-06	November	PR	9
6	2017-04-20	April	MT	1
7	2017-07-13	July	MA	6
8	2017-07-11	July	AL	1
9	2017-07-29	July	SP	47
10	2017-07-13	July	MT	2

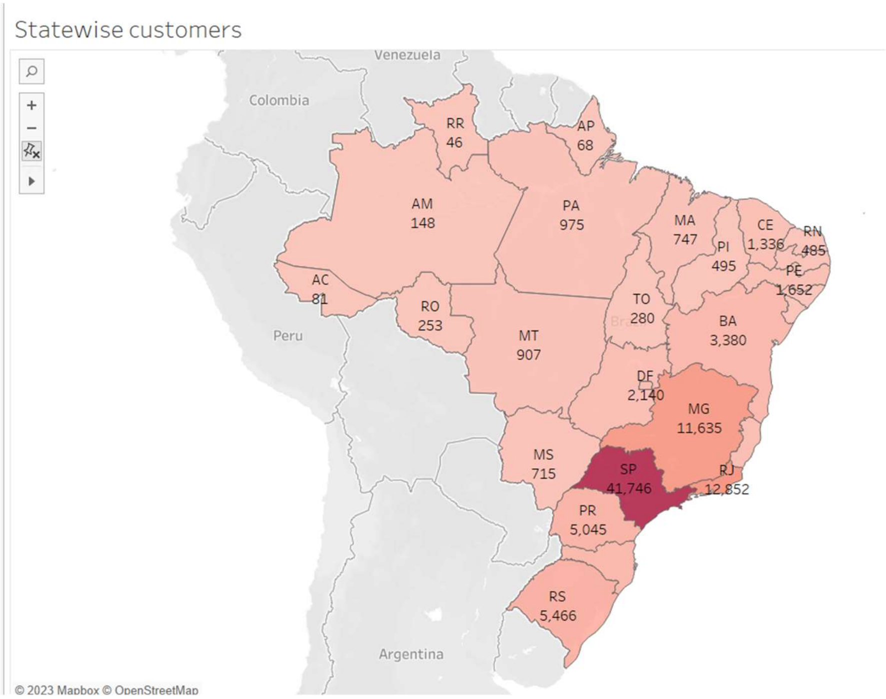
2. Distribution of customers across the states in Brazil

Query-

```
SELECT customer_state, count(customer_id) as Num_of_customers

  FROM `target-380817.Target123.Customers`
  Group by customer_state;
```



Row	customer_state	Num_of_custom
1	RN	485
2	CE	1336
3	RS	5466
4	SC	3637
5	SP	41746
6	MG	11635
7	BA	3380
8	RJ	12852
9	GO	2020
10	MA	747



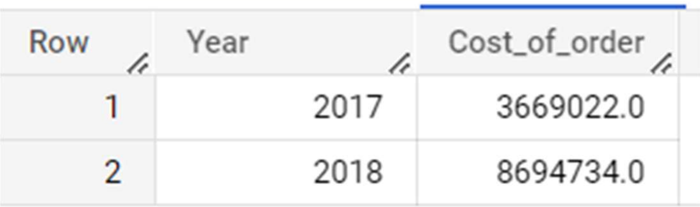
As we can see that most customers are in Sao Paulo(SA) and least customers are in Roraima(RR).

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

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

Firstly to calculate cost of orders for the years(2017 and 2018)

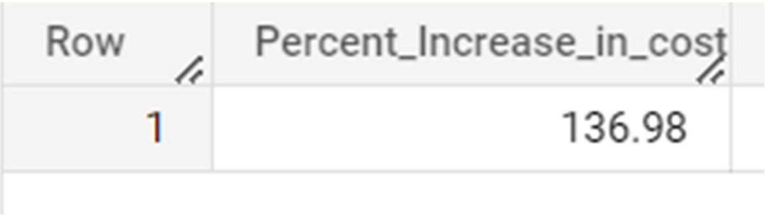
Query -
```
Select * from ( Select
  extract(ISOYEAR from (extract ( date from order_purchase_timestamp))) AS Ye
ar ,
Round(sum(p.payment_value), 0) as Cost_of_order,
 from target-380817.Target123.orders as o inner join target-
380817.Target123.Payments as p
 on o.order_id= p.order_id
 where extract(ISOYEAR from (extract ( date from order_purchase_timestamp)))
in (2017, 2018)   and
FORMAT_DATE('%B', order_purchase_timestamp) in ("January","February", "March"
,"April","May", "June","July","August")
  group by extract(ISOYEAR from (extract ( date from order_purchase_timestamp
)))
  )
 order by Year asc;
```



Row	Year	Cost_of_order
1	2017	3669022.0
2	2018	8694734.0

Then to calculate % increase in cost of orders from 2017 to 2018

Query-
```
Select Round((SAFE_SUBTRACT(8694734.0, 3669022.0)/3669022.0)*100,2) as Percen
t_Increase_in_cost;
```

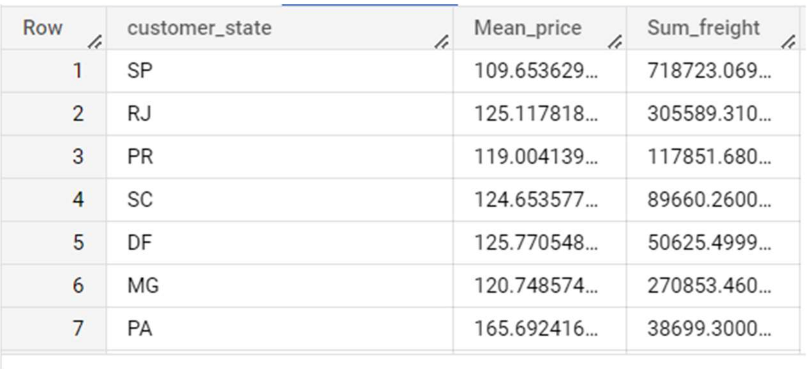


Row	Percent_Increase_in_cost
1	136.98

Mean & Sum of price and freight value by customer state

Query-

```
SELECT customer_state, Avg(i.price) as Mean_price, Sum(i.freight_value) as Su
m_freight

FROM target-380817.Target123.order_items as i inner join target-
380817.Target123.orders as o
on i.order_id = o.order_id
inner join target-380817.Target123.Customers as c
on o.customer_id = c.customer_id
group by customer_state;
```



Row	customer_state	Mean_price	Sum_freight
1	SP	109.653629...	718723.069...
2	RJ	125.117818...	305589.310...
3	PR	119.004139...	117851.680...
4	SC	124.653577...	89660.2600...
5	DF	125.770548...	50625.4999...
6	MG	120.748574...	270853.460...
7	PA	165.692416...	38699.3000...

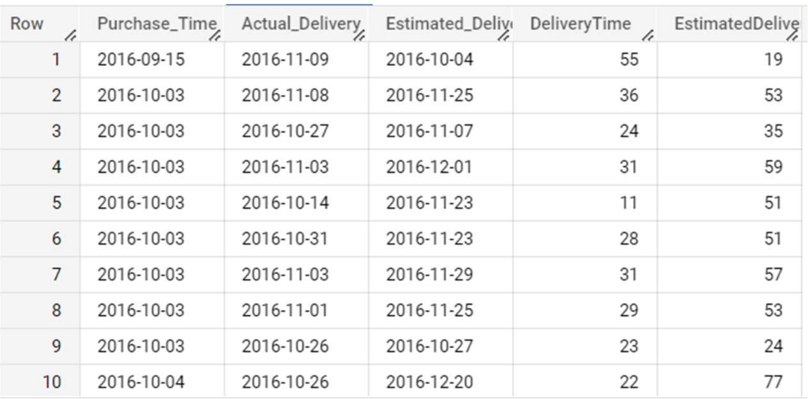
5. Analysis on sales, freight and delivery time

Calculate days between purchasing, delivering and estimated delivery

Query-

```
Select Purchase_Time, Actual_Delivery, Estimated_Delivery, DATE_DIFF(Actual_D
elivery,Purchase_Time, Day) As DeliveryTime,

DATE_DIFF(Estimated_Delivery,Purchase_Time, Day) As EstimatedDeliveryTime,
 from
(
SELECT extract( Date from order_purchase_timestamp) As Purchase_Time, extract
( date from order_delivered_customer_date) As Actual_Delivery
, extract( Date from order_estimated_delivery_date) As Estimated_Delivery FRO
M `target-380817.Target123.orders`
where order_delivered_customer_date is not null
) as X
```

```
order by Purchase_Time;
```

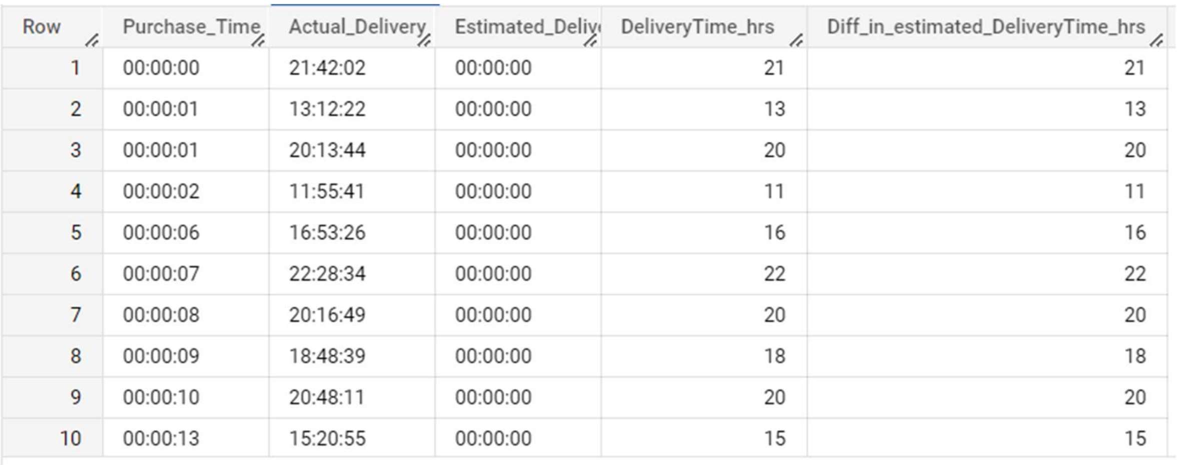


Row	Purchase_Time	Actual_Delivery	Estimated_Deliv	DeliveryTime	EstimatedDelive
1	2016-09-15	2016-11-09	2016-10-04	55	19
2	2016-10-03	2016-11-08	2016-11-25	36	53
3	2016-10-03	2016-10-27	2016-11-07	24	35
4	2016-10-03	2016-11-03	2016-12-01	31	59
5	2016-10-03	2016-10-14	2016-11-23	11	51
6	2016-10-03	2016-10-31	2016-11-23	28	51
7	2016-10-03	2016-11-03	2016-11-29	31	57
8	2016-10-03	2016-11-01	2016-11-25	29	53
9	2016-10-03	2016-10-26	2016-10-27	23	24
10	2016-10-04	2016-10-26	2016-12-20	22	77

Find time_to_delivery & diff_estimated_delivery.

Query-

```
Select Purchase_Time, Actual_Delivery, Estimated_Delivery, Time_diff(Actual_Delivery,Purchase_Time,Hour) As DeliveryTime_hrs,

TIME_DIFF(Actual_Delivery,Estimated_Delivery,Hour) As Diff_in_estimated_DeliveryTime_hrs,
 from
(
SELECT extract( time from order_purchase_timestamp) As Purchase_Time, extract( time from order_delivered_customer_date) As Actual_Delivery
, extract( time from order_estimated_delivery_date) As Estimated_Delivery FROM `target-380817.Target123.orders`
where order_delivered_customer_date is not null
) as X
order by Purchase_Time;
```

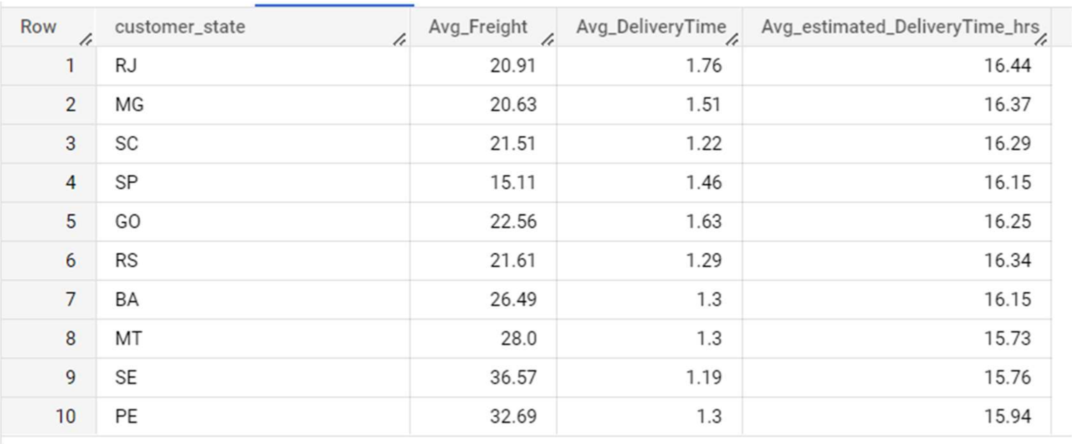


Row	Purchase_Time	Actual_Delivery	Estimated_Deliv	DeliveryTime_hrs	Diff_in_estimated_DeliveryTime_hrs
1	00:00:00	21:42:02	00:00:00	21	21
2	00:00:01	13:12:22	00:00:00	13	13
3	00:00:01	20:13:44	00:00:00	20	20
4	00:00:02	11:55:41	00:00:00	11	11
5	00:00:06	16:53:26	00:00:00	16	16
6	00:00:07	22:28:34	00:00:00	22	22
7	00:00:08	20:16:49	00:00:00	20	20
8	00:00:09	18:48:39	00:00:00	18	18
9	00:00:10	20:48:11	00:00:00	20	20
10	00:00:13	15:20:55	00:00:00	15	15

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

Query-

```
Select customer_state,Round(Avg(Freight),2) as Avg_Freight, Round(Avg(Time_diff(Actual_Delivery,Purchase_Time,Hour)),2) As Avg_DeliveryTime,

Round(Avg(TIME_DIFF(Actual_Delivery,Estimated_Delivery,Hour)),2) As Avg_estimated_DeliveryTime_hrs,
 from
(
SELECT c.customer_state, i.freight_value as Freight, extract( time from order_purchase_timestamp) As Purchase_Time,
extract( time from order_delivered_customer_date) As Actual_Delivery
, extract( time from order_estimated_delivery_date) As Estimated_Delivery
FROM `target-380817.Target123.orders` as o inner join `target-380817.Target123.Customers` as c
on o.customer_id = c.customer_id
inner join `target-380817.Target123.order_items` as i
on o.order_id = i.order_id
where order_delivered_customer_date is not null
) as X
group by customer_state;
```



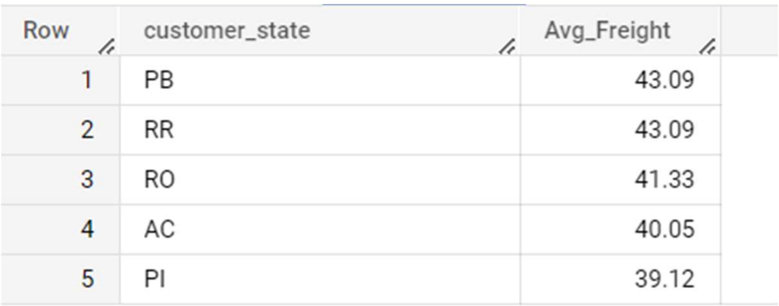
Row	customer_state	Avg_Freight	Avg_DeliveryTime	Avg_estimated_DeliveryTime_hrs
1	RJ	20.91	1.76	16.44
2	MG	20.63	1.51	16.37
3	SC	21.51	1.22	16.29
4	SP	15.11	1.46	16.15
5	GO	22.56	1.63	16.25
6	RS	21.61	1.29	16.34
7	BA	26.49	1.3	16.15
8	MT	28.0	1.3	15.73
9	SE	36.57	1.19	15.76
10	PE	32.69	1.3	15.94

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

Top 5 states with highest average freight value

Query-

```
Select customer_state,Round(Avg(Freight),2) as Avg_Freight
 from
(
SELECT c.customer_state, i.freight_value as Freight
FROM `target-380817.Target123.orders` as o inner join `target-
380817.Target123.Customers` as c
on o.customer_id = c.customer_id
inner join `target-380817.Target123.order_items` as i
on o.order_id = i.order_id
where order_delivered_customer_date is not null
) as X
group by customer_state
order by Avg_Freight desc
limit 5;
```

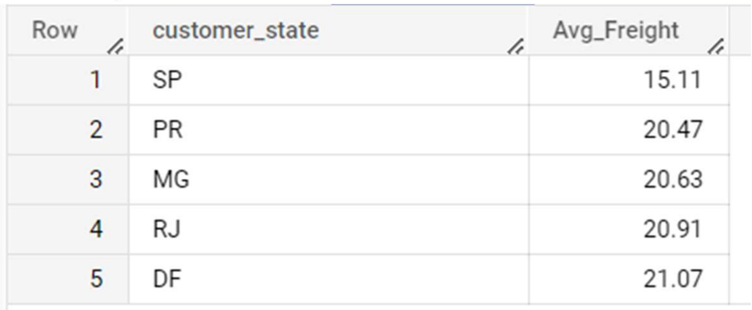


Row	customer_state	Avg_Freight
1	PB	43.09
2	RR	43.09
3	RO	41.33
4	AC	40.05
5	PI	39.12

Top 5 states with lowest average freight value

Query-

```
Select customer_state,Round(Avg(Freight),2) as Avg_Freight

 from
(
SELECT c.customer_state, i.freight_value as Freight
FROM `target-380817.Target123.orders` as o inner join `target-
380817.Target123.Customers` as c
on o.customer_id = c.customer_id
inner join `target-380817.Target123.order_items` as i
on o.order_id = i.order_id
where order_delivered_customer_date is not null
) as X
group by customer_state
order by Avg_Freight asc
limit 5;
```



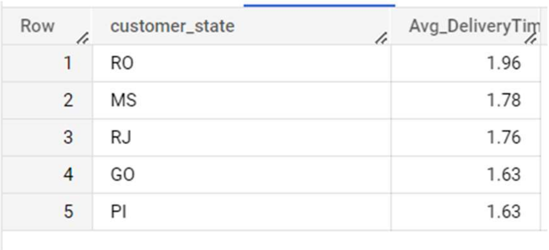
Row	customer_state	Avg_Freight
1	SP	15.11
2	PR	20.47
3	MG	20.63
4	RJ	20.91
5	DF	21.07

Top 5 states with highest/lowest average time to delivery

Top 5 states with highest average Time to delivery(in hours)

Query-

```
Select customer_state, Round(Avg(Time_diff(Actual_Delivery,Purchase_Time,Hour
)),2) As Avg_DeliveryTime

 from
(
SELECT c.customer_state, extract( time from order_purchase_timestamp) As Purc
hase_Time,
extract( time from order_delivered_customer_date) As Actual_Delivery
FROM `target-380817.Target123.orders` as o inner join `target-
380817.Target123.Customers` as c
on o.customer_id = c.customer_id
inner join `target-380817.Target123.order_items` as i
on o.order_id = i.order_id
where order_delivered_customer_date is not null
) as X
group by customer_state
order by Avg_DeliveryTime desc
limit 5;
```

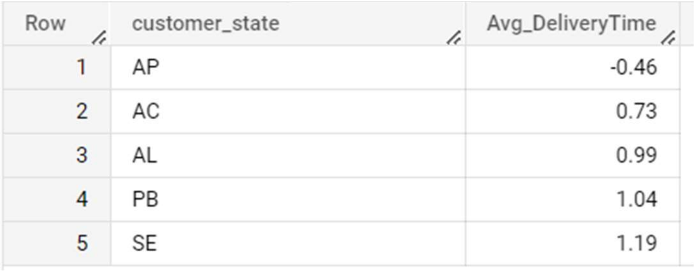


Row	customer_state	Avg_DeliveryTim
1	RO	1.96
2	MS	1.78
3	RJ	1.76
4	GO	1.63
5	PI	1.63

Top 5 states with lowest average Time to delivery(in hours)

Query-

```
Select customer_state, Round(Avg(Time_diff(Actual_Delivery,Purchase_Time,Hour
)),2) As Avg_DeliveryTime

 from
(
SELECT c.customer_state, extract( time from order_purchase_timestamp) As Purc
hase_Time,
extract( time from order_delivered_customer_date) As Actual_Delivery
FROM `target-380817.Target123.orders` as o inner join `target-
380817.Target123.Customers` as c
on o.customer_id = c.customer_id
inner join `target-380817.Target123.order_items` as i
on o.order_id = i.order_id
where order_delivered_customer_date is not null
) as X
group by customer_state
order by Avg_DeliveryTime desc
limit 5;
```

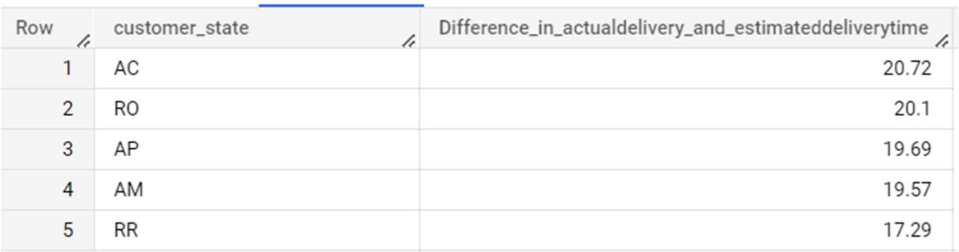


Row	customer_state	Avg_DeliveryTime
1	AP	-0.46
2	AC	0.73
3	AL	0.99
4	PB	1.04
5	SE	1.19

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

Top 5 states with Fastest delivery(in days)

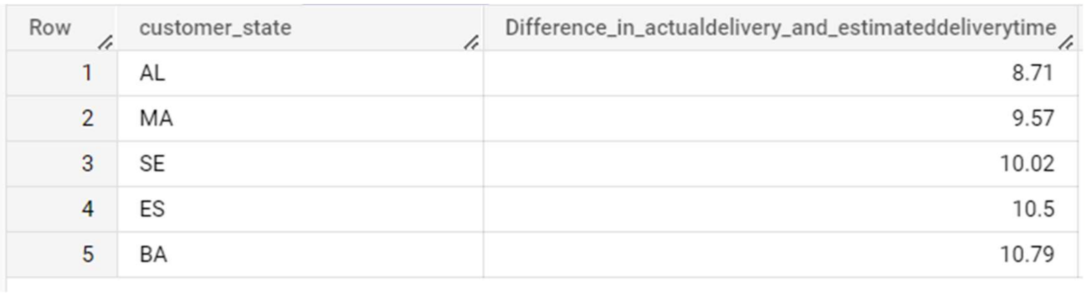
Query-
```
Select customer_state ,
Round(Avg(DATE_DIFF(Estimated_Delivery, Actual_Delivery, Day)),2) As Difference_in_actualdelivery_and_estimateddeliverytime
 from
(
SELECT c.customer_state ,extract( Date from order_purchase_timestamp) As Purchase_Time, extract( date from order_delivered_customer_date) As Actual_Delivery
, extract( Date from order_estimated_delivery_date) As Estimated_Delivery
 FROM `target-380817.Target123.orders` as o inner join `target-380817.Target123.Customers` as c
on o.customer_id = c.customer_id
where order_delivered_customer_date is not null
) as X
group by customer_state
order by Difference_in_actualdelivery_and_estimateddeliverytime desc
limit 5;
```



Row	customer_state	Difference_in_actualdelivery_and_estimateddeliverytime
1	AC	20.72
2	RO	20.1
3	AP	19.69
4	AM	19.57
5	RR	17.29

Top 5 states with slowest delivery(in days)

Query-
```
Select customer_state ,
Round(Avg(DATE_DIFF(Estimated_Delivery, Actual_Delivery, Day)),2) As Difference_in_actualdelivery_and_estimateddeliverytime
 from
(
SELECT c.customer_state ,extract( Date from order_purchase_timestamp) As Purchase_Time, extract( date from order_delivered_customer_date) As Actual_Delivery
, extract( Date from order_estimated_delivery_date) As Estimated_Delivery
 FROM `target-380817.Target123.orders` as o inner join `target-380817.Target123.Customers` as c
on o.customer_id = c.customer_id
where order_delivered_customer_date is not null
) as X
group by customer_state
order by Difference_in_actualdelivery_and_estimateddeliverytime asc
limit 5;
```

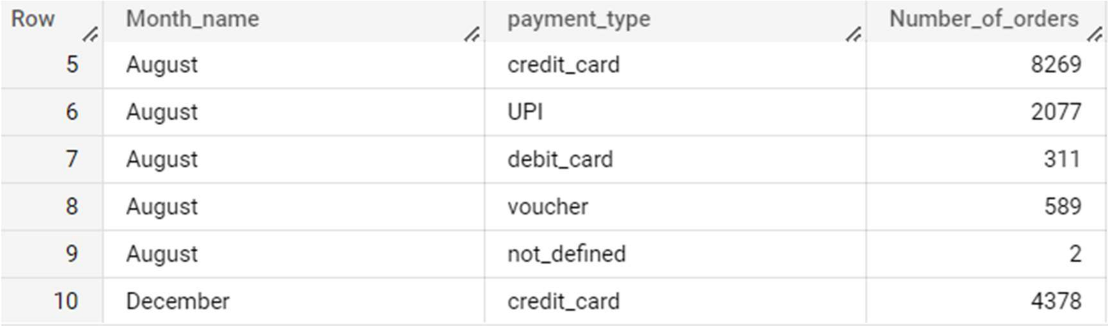


Row	customer_state	Difference_in_actualdelivery_and_estimateddeliverytime
1	AL	8.71
2	MA	9.57
3	SE	10.02
4	ES	10.5
5	BA	10.79

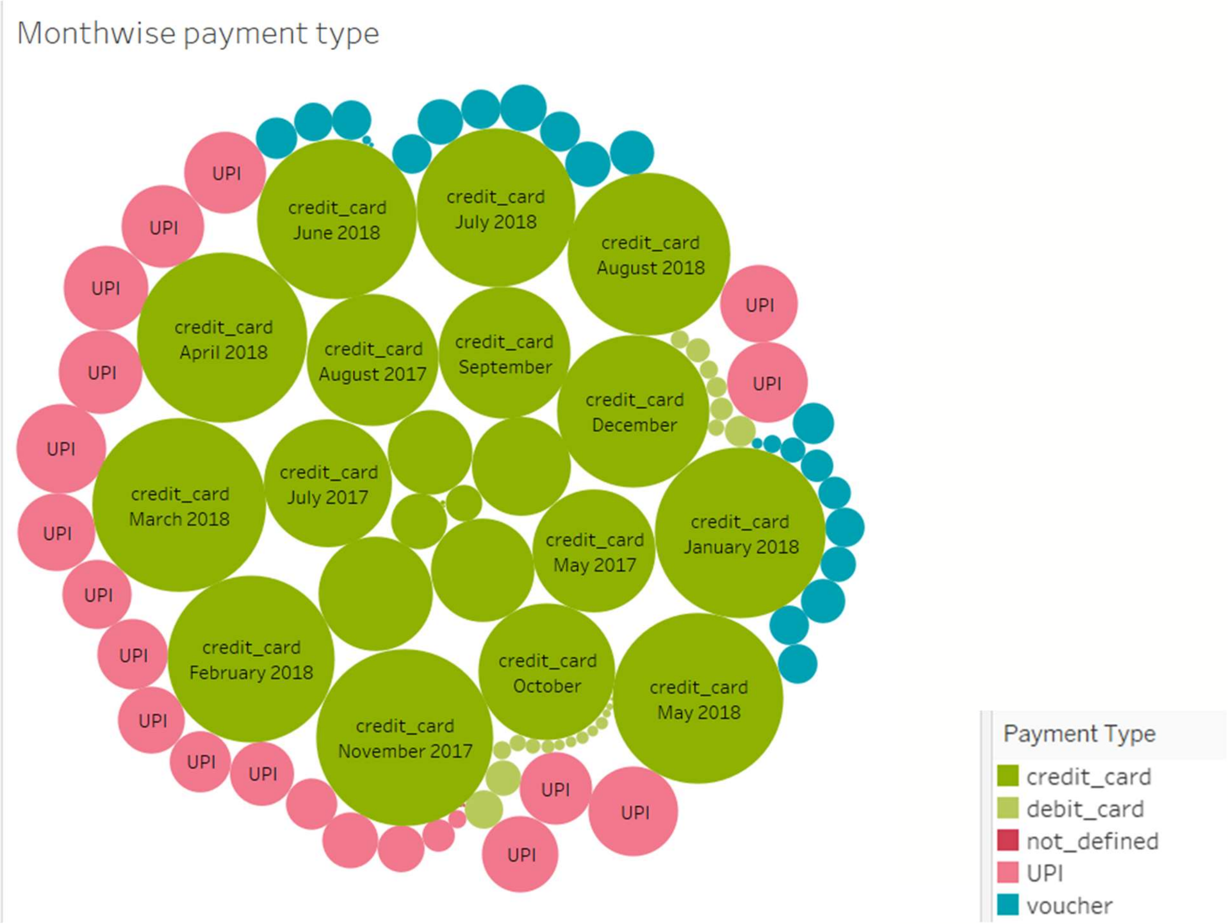
6. Payment type analysis

Month over Month count of orders for different payment types

Query-
```
Select * from
 (
 Select FORMAT_DATE('%B', order_purchase_timestamp) AS Month_name ,p.payment_type, count(p.order_id) as Number_of_orders
 from target-380817.Target123.orders as o inner join target-
380817.Target123.Payments as p
 on o.order_id = p.order_id
  group by FORMAT_DATE('%B', order_purchase_timestamp), p.payment_type
  )
  order by Month_name;
```



Row	Month_name	payment_type	Number_of_orders
5	August	credit_card	8269
6	August	UPI	2077
7	August	debit_card	311
8	August	voucher	589
9	August	not_defined	2
10	December	credit_card	4378

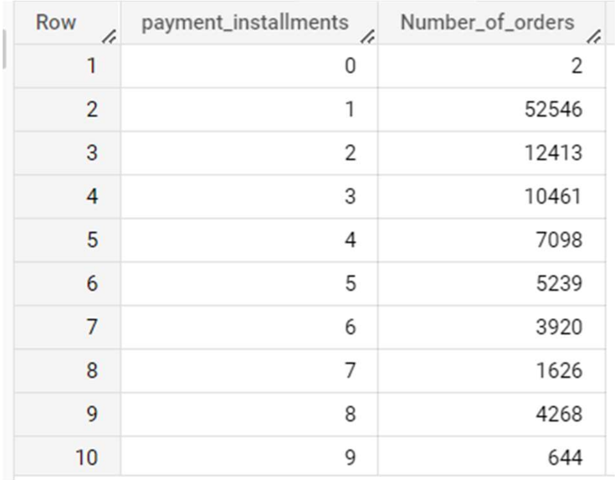


Hence we can analyze that mostly payments were done using credit cards.

Count of orders based on the no. of payment installments

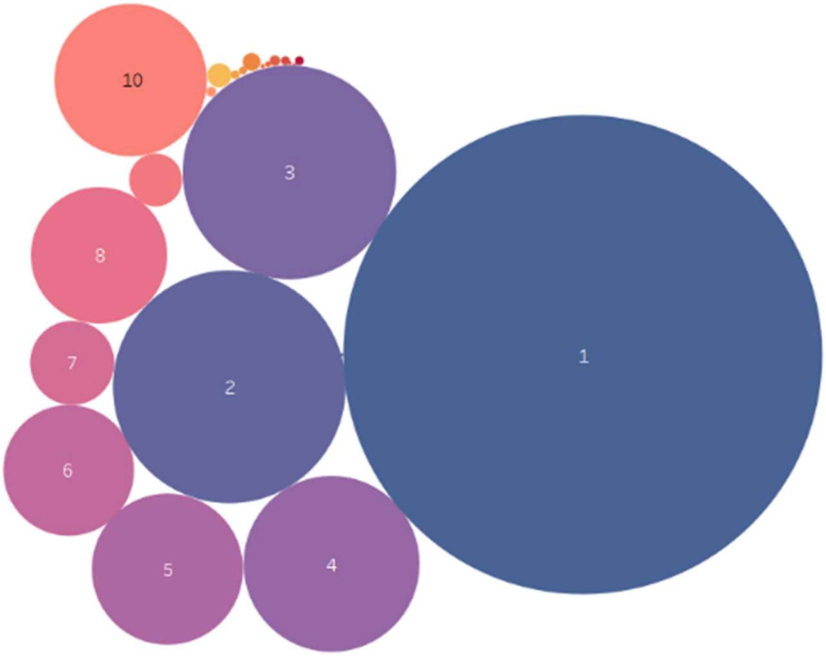
Query-

```
SELECT payment_installments, count(order_id) as Number_of_orders FROM `target
-380817.Target123.Payments`

group by payment_installments
order by payment_installments;
```



Row	payment_installments	Number_of_orders
1	0	2
2	1	52546
3	2	12413
4	3	10461
5	4	7098
6	5	5239
7	6	3920
8	7	1626
9	8	4268
10	9	644

Payment Installment


10
3
8
1
7
2
6
5
4