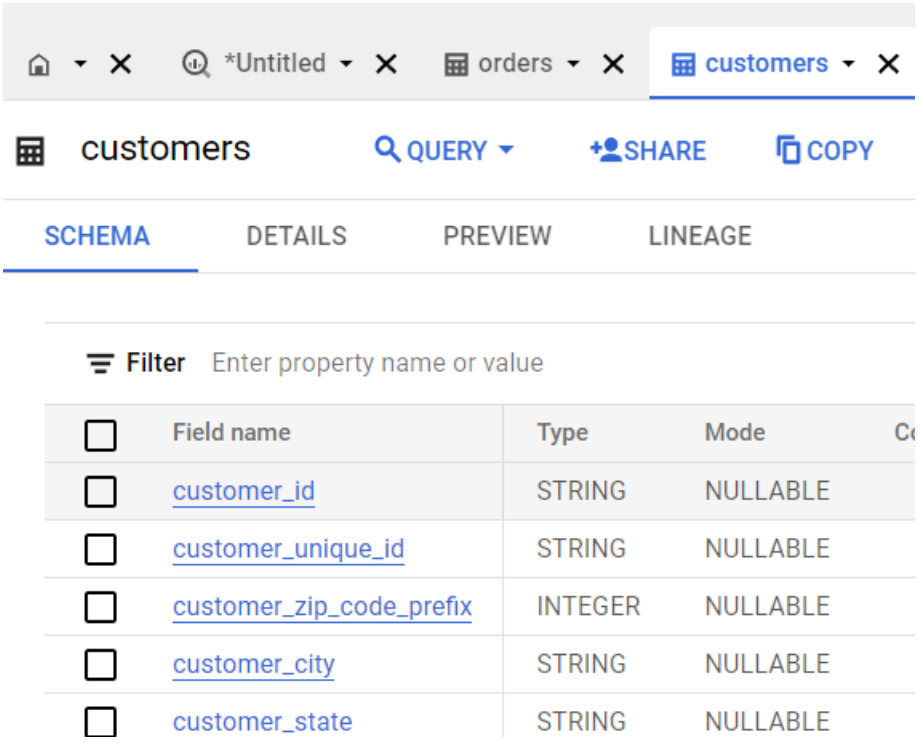
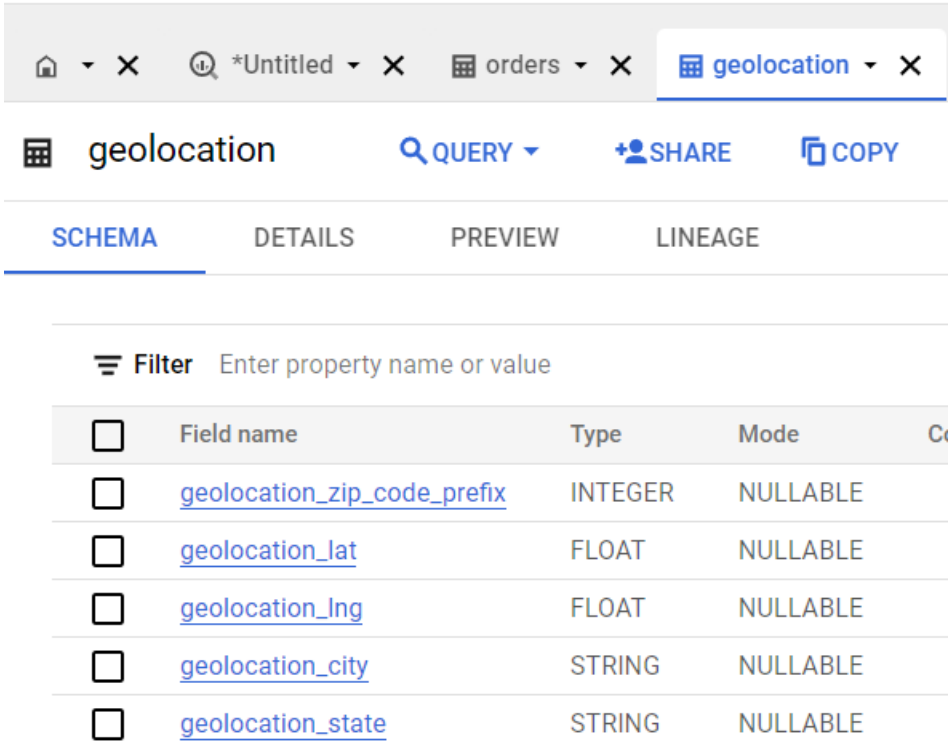
1) Exploratory analysis

1) Data type of columns in a table:

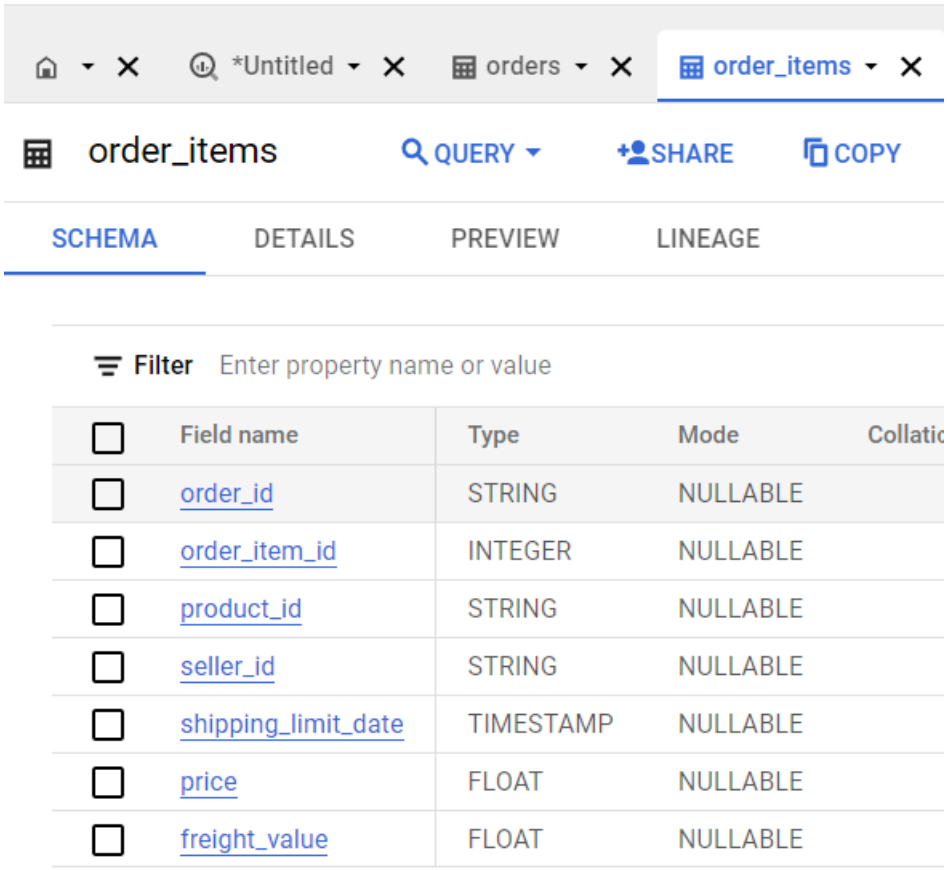
CUSTOMERS TABLE:



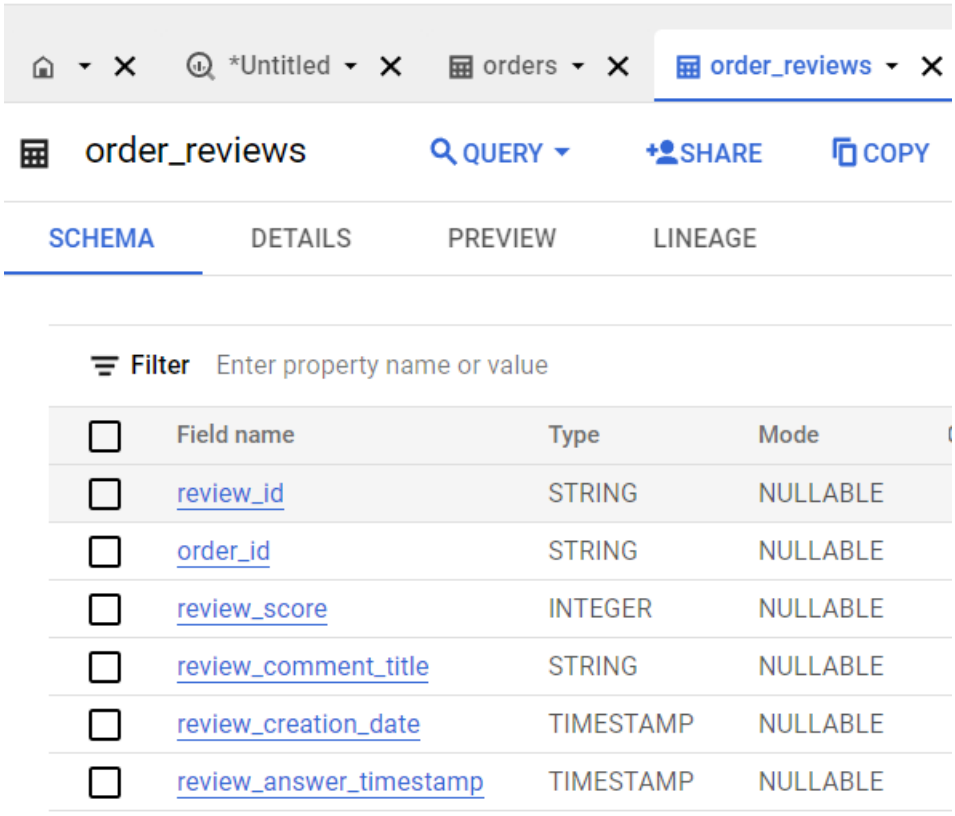
GEOLOCATION TABLE:



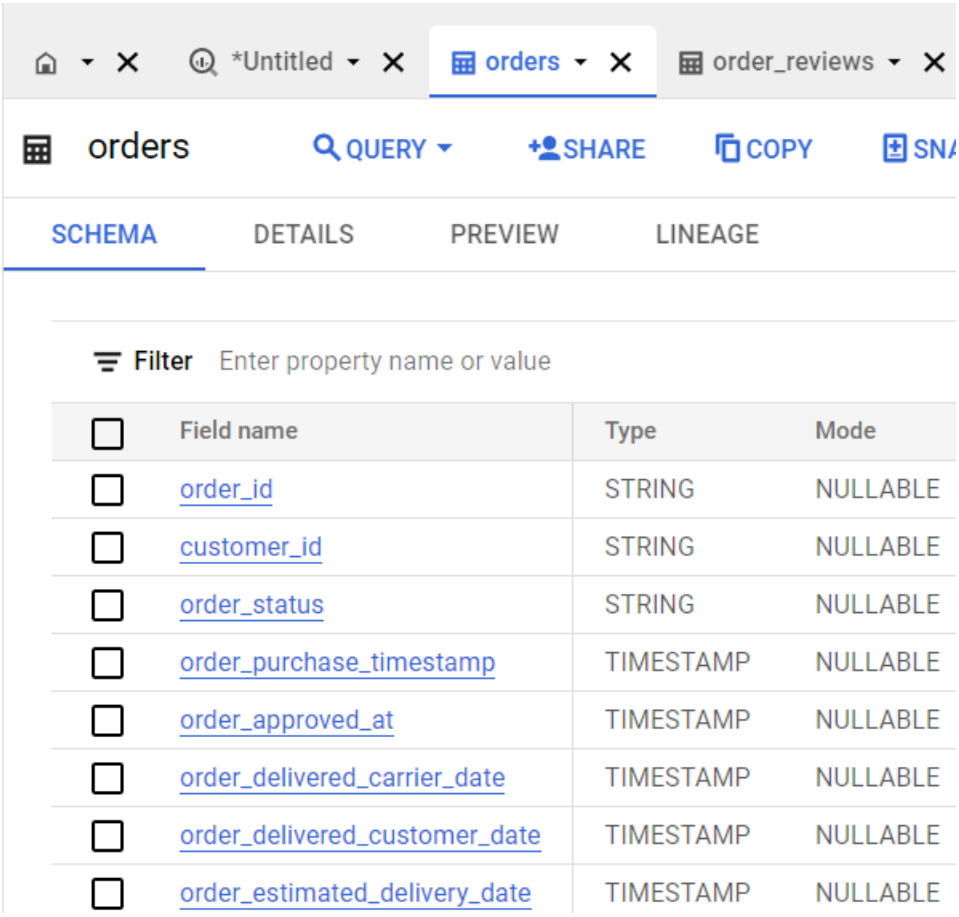
ORDERITEMS TABLE:



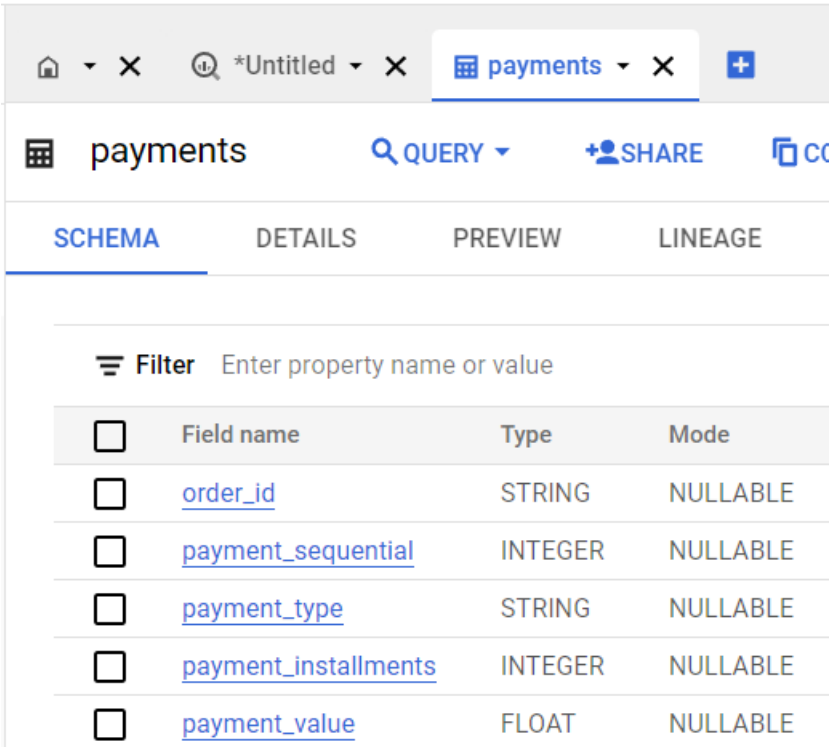
ORDER REVIEWS TABLE:



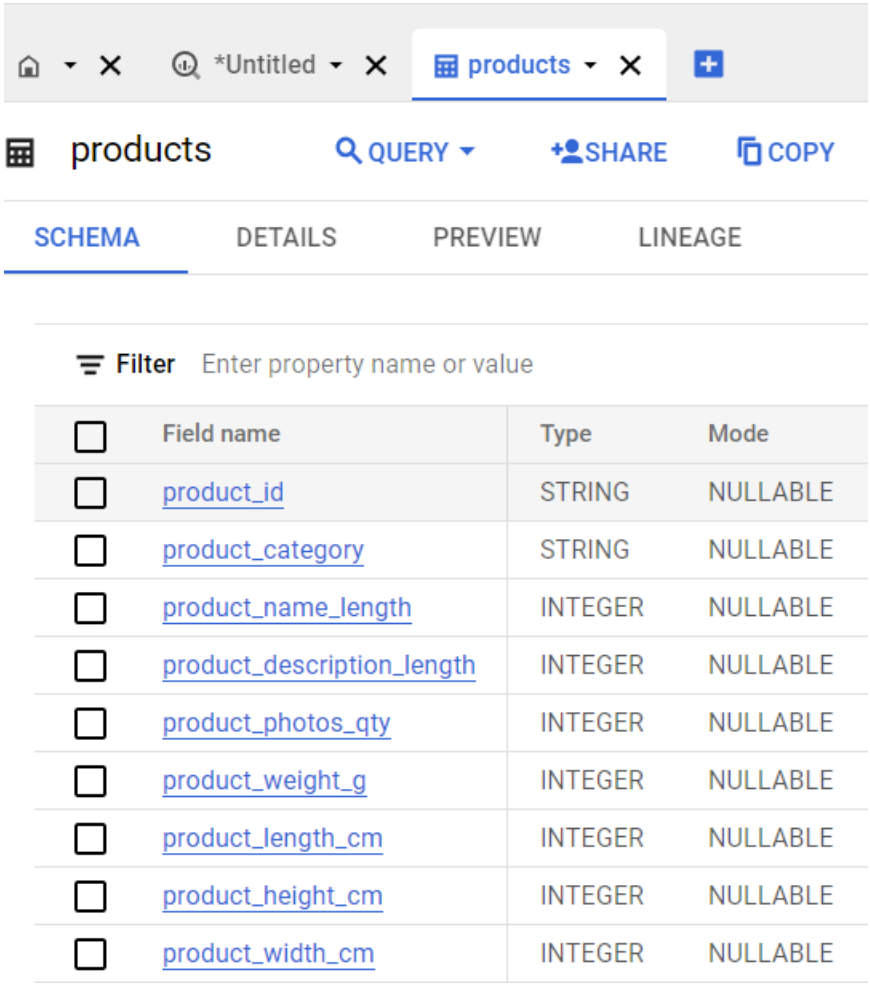
ORDERS:


*Untitled
orders
order_reviews
orders
QUERY
SHARE
COPY
SCHEMA
DETAILS
PREVIEW
LINEAGE
Filter
Enter property name or value
Field name
Type
Mode
order_id
STRING
NULLABLE
customer_id
STRING
NULLABLE
order_status
STRING
NULLABLE
order_purchase_timestamp
TIMESTAMP
NULLABLE
order_approved_at
TIMESTAMP
NULLABLE
order_delivered_carrier_date
TIMESTAMP
NULLABLE
order_delivered_customer_date
TIMESTAMP
NULLABLE
order_estimated_delivery_date
TIMESTAMP
NULLABLE

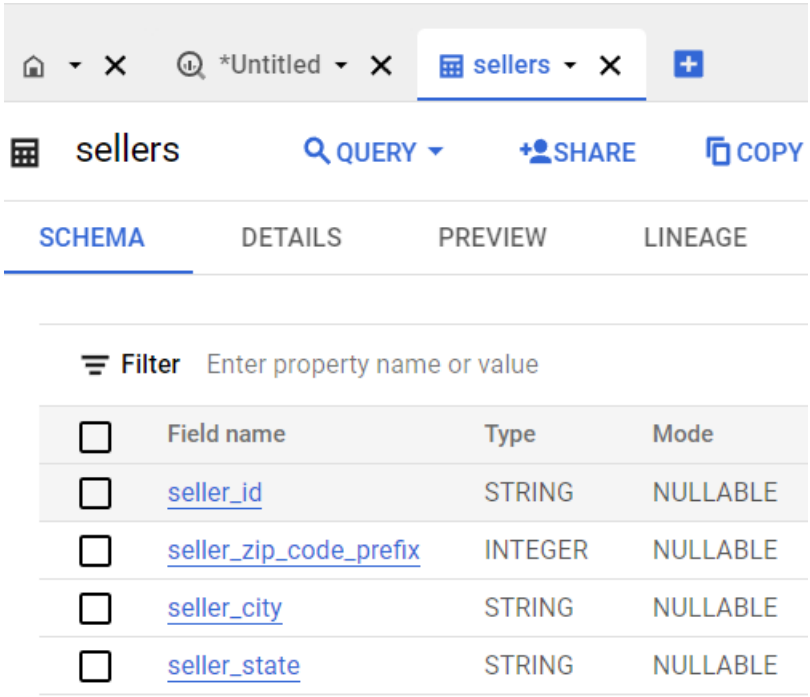
PAYMENTS TABLE:



PRODUCTS TABLE:



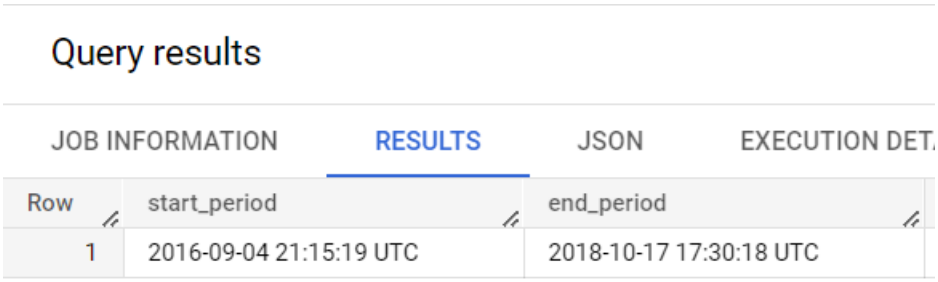
SELLERS TABLE:



2)Time period for which the data is given:

SQL :

```
select  min(order_purchase_timestamp) as start_period,max(order_purchase_timestamp) as
 end_period,from `Target_SQL.orders`
```



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

SQL:

```
SELECT distinct c.customer_city, c.customer_state
FROM
`Target_SQL.customers` c JOIN `Target_SQL.orders` o
on
c.customer_id = o.customer_id
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DET

Row	customer_city	customer_state
1	abadia dos dourados	MG
2	abadiania	GO
3	abaete	MG
4	abaetetuba	PA
5	abaiara	CE
6	abaira	BA
7	abare	BA
8	abatia	PR
9	abdon batista	SC
10	abelardo luz	SC
11	abrantes	BA
12	abre campo	MG
13	abreu e lima	PE
14	acaiaca	MG
15	acailandia	MA
16	acajutiba	BA
17	acarau	CE
18	acari	RN

Analysis:

Around 4310 cities and states are listed for the given period in Brazil.

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?

Month-wise:

SQL:

```
SELECT FORMAT_DATETIME("%B", DATETIME(order_purchase_timestamp)) as month_year, COUNT(
*) AS total_orders
 FROM `Target_SQL.orders`
 GROUP BY month_year
 ORDER BY 2 desc;
```

Query results

JOB INFORMATION | RESULTS | JSON

Row	month_year	total_orders
1	August	10843
2	May	10573
3	July	10318
4	March	9893
5	June	9412
6	April	9343
7	February	8508
8	January	8069
9	November	7544
10	December	5674
11	October	4959
12	September	4305

Analysis:-

August has the highest no of sales and September being the lowest.

Could increase sales in other months by analyzing the factor that contributes to increase in sales in August.

MONTH-YEAR-wise:-

SQL:

```
SELECT FORMAT_DATETIME("%B %Y", DATETIME(order_purchase_timestamp)) as month_year, COU
NT(*) AS total_orders
 FROM `Target_SQL.orders`
 GROUP BY month_year
 ORDER BY month_year;
```

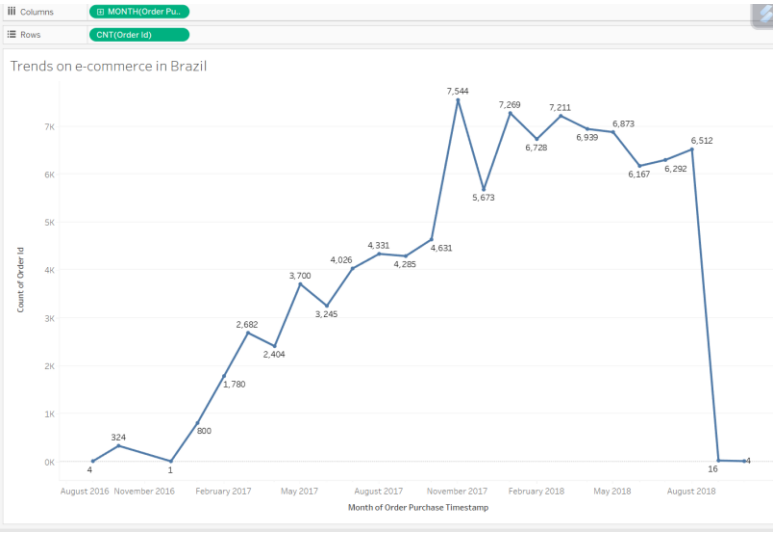
Query results

JOB INFORMATION | RESULTS | JSON

Row	month_year	total_orders
1	April 2017	2404
2	April 2018	6939
3	August 2017	4331
4	August 2018	6512
5	December 2016	1
6	December 2017	5673
7	February 2017	1780
8	February 2018	6728
9	January 2017	800
10	January 2018	7269
11	July 2017	4026
12	July 2018	6292
13	June 2017	3245
14	June 2018	6167
15	March 2017	2682
16	March 2018	7211
17	May 2017	3700
18	May 2018	6873
19	November 2017	7544

z

Tableau results :



Analysis:

We could clearly see that trend on e-commerce in Brazil is steadily increasing from June 2017 till November 2017 reaching its peak during November 2017 with 7544 orders and there is slight decrease in trend till August 2018 .

Although sales are poor after August 2018 which is same as during September 2016.

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

SQL:

```
select

SUM(CASE WHEN EXTRACT(time from order_purchase_timestamp) BETWEEN '00:00:00' and '06:0
0:00' THEN 1 ELSE 0 END) as dawn,
SUM(CASE WHEN EXTRACT(time from order_purchase_timestamp) BETWEEN '07:00:00' and '12:0
0:00' THEN 1 ELSE 0 END) as Morning,
```

```
SUM(CASE WHEN EXTRACT(time from order_purchase_timestamp) BETWEEN '13:00:00' and '18:0
0:00' THEN 1 ELSE 0 END) as Afternoon,
SUM(CASE WHEN EXTRACT(time from order_purchase_timestamp) BETWEEN '19:00:00' and '23:0
0:00' THEN 1 ELSE 0 END) as Night,

FROM `Target_SQL.orders`
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAI

Row	dawn	Morning	Afternoon	Night
1	4740	21738	32370	24209

Analysis:

Brazilian customers are ordering more during afternoon than any other part of the day.

So, citizens are purchasing after office hours in afternoon and Night .

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

1)Get month on month orders by states:

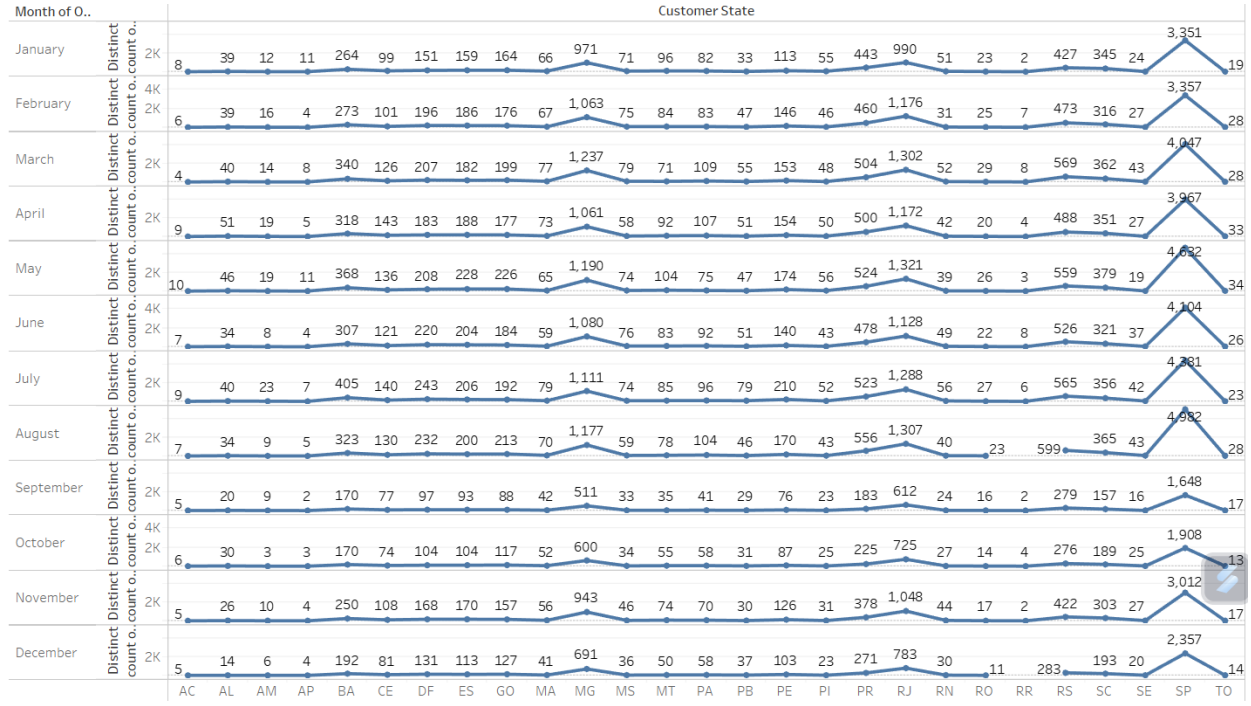
```
select c.customer_state,extract(month from o.order_purchase_timestamp) as month,
extract(year from o.order_purchase_timestamp) as year,count(o.order_id) as total_order
s
from
`Target_SQL.orders` o join `Target_SQL.customers`c
on o.customer_id = c.customer_id
group by c.customer_state,year,month
order by year,month;
```

Query results

SAVE RESULTS

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXEC

Row	customer_state	month	year	total_orders
1	RR	9	2016	1
2	RS	9	2016	1
3	SP	9	2016	2
4	SP	10	2016	113
5	RS	10	2016	24
6	RJ	10	2016	56
7	MT	10	2016	3
8	GO	10	2016	9
9	MG	10	2016	40
10	CE	10	2016	8
11	SC	10	2016	11
12	AL	10	2016	2
13	BA	10	2016	4
14	PE	10	2016	7
15	ES	10	2016	4
16	MA	10	2016	4



Analysis:

Above results shows that order counts are maximum in SP state in brazil among all other states.

All the other state should adapt strategies used in SP.

Orders are increasing during mid-months May,June,July in all state and Target should ensure supply of products.

2) Distribution of customers across the states in Brazil:

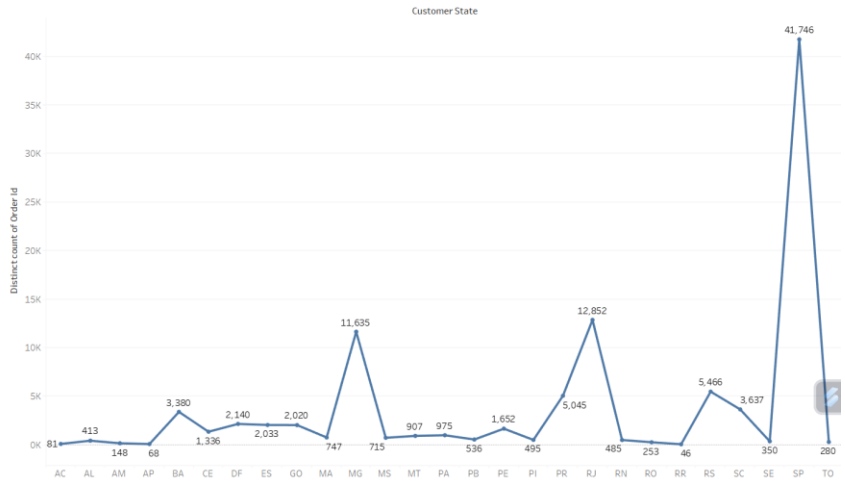
SQL:

```
select customer_state,count(distinct customer_id) as no_of_customers
from `Target_SQL.customers`
group by customer_state
order by no_of_customers desc
```

Query results

JOB INFORMATION | RESULTS | JSON

Row	customer_state	no_of_customer
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
13	PA	975
14	MT	907
15	MA	747
16	MS	715



Analysis:

State SP has maximum no of customers followed by RJ ,MG etc and Target should adapt strategies to improve and attract customers in other states as well by taking SP state into consideration.

4) Impact on Economy:

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

```
select round(((sum(b.cost_orders2018) - sum(a.cost_of_orders2017)) /sum(a.cost_of_orde
rs2017)* 100),2)  as percentage_increase from(


(select sum(p.payment_value) as cost_of_orders2017,extract(month from o.order_purchase
_timestamp) as month,extract(year from o.order_purchase_timestamp) as year
from
`Target_SQL.payments` p join `Target_SQL.orders` o
on p.order_id = o.order_id
group by 2,3
having month between 1 and 8 and year = 2017
order by month) a

inner join

(select sum(p.payment_value) as cost_orders2018,extract(month from o.order_purchase_ti
mestamp) as month,extract(year from o.order_purchase_timestamp) as year
from
`Target_SQL.payments` p join `Target_SQL.orders` o
on p.order_id = o.order_id
group by 2,3
having month between 1 and 8 and year = 2018
order by month)b
on a.month = b.month);
```

Query results

JOB INFORMATION RE

Row	percentage_increase
1	136.98

Analysis:

There is increase of 36.9 % in cost of orders from 2017 to 2018 in months between JAN and AUG.

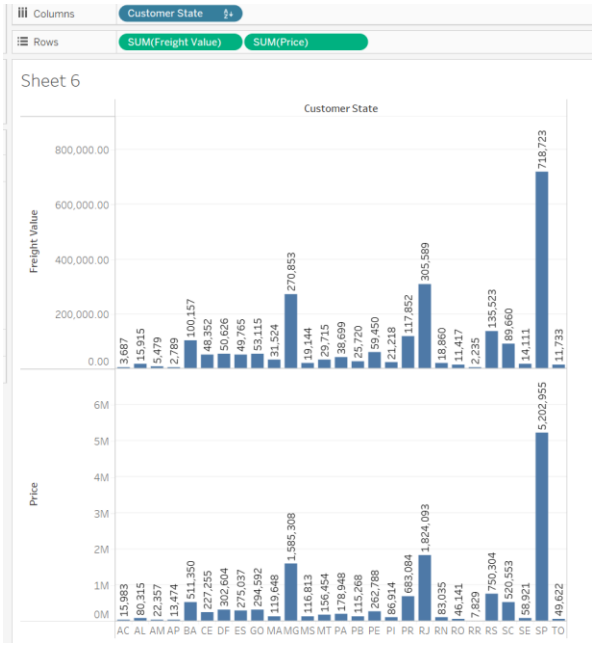
2)Mean & Sum of price and freight value by customer state

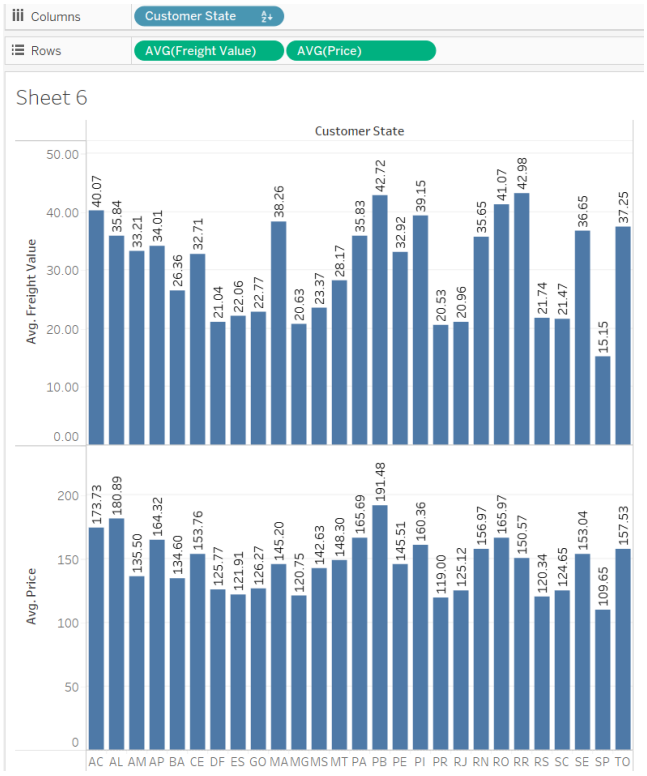
```
select c.customer_state,
round(sum(oi.price),2)  as sum_price,
round(avg(oi.price),2)  as mean_price,
round(sum(oi.freight_value),2) as sum_fght_value,
round(avg(oi.freight_value),2)  as avg_fght_value
from
`Target_SQL.orders` o join `Target_SQL.order_items` oi
on o.order_id = oi.order_id
join `Target_SQL.customers` c
on o.customer_id = c.customer_id
group by 1
order by 1
```

Query results

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

Row	customer_state	sum_price	mean_price	sum_fght_value	avg_fght_value
1	AC	15982.95	173.73	3686.75	40.07
2	AL	80314.81	180.89	15914.59	35.84
3	AM	22356.84	135.5	5478.89	33.21
4	AP	13474.3	164.32	2788.5	34.01
5	BA	511349.99	134.6	100156.68	26.36
6	CE	227254.71	153.76	48351.59	32.71
7	DF	302603.94	125.77	50625.5	21.04
8	ES	275037.31	121.91	49764.6	22.06
9	GO	294591.95	126.27	53114.98	22.77
10	MA	119648.22	145.2	31523.77	38.26
11	MG	1585308.03	120.75	270853.46	20.63
12	MS	116812.64	142.63	19144.03	23.37
13	MT	156453.53	148.3	29715.43	28.17
14	PA	178947.81	165.69	38699.3	35.83
15	PB	115268.08	191.48	25719.73	42.72
16	PE	262788.03	145.51	59449.66	32.92
17	PI	86914.08	160.36	21218.2	39.15
18	PR	683083.76	119.0	117851.68	20.53
19	RJ	1824092.67	125.12	305589.31	20.96


Columns
Customer State
Rows
SUM(Freight Value)
SUM(Price)
Sheet 6
Customer State
Freight Value
800,000.00
600,000.00
400,000.00
200,000.00
0.00
3,687
15,915
5,479
2,789
100,157
48,352
50,626
49,765
53,115
31,524
270,853
19,144
29,715
38,699
25,720
59,450
21,218
117,852
305,589
18,860
11,417
2,235
135,523
89,660
14,111
718,723
11,733
Price
6M
5M
4M
3M
2M
1M
0M
15,983
80,315
22,357
13,474
511,350
227,255
302,604
275,037
294,592
119,648
1,585,308
116,813
156,454
178,948
115,268
262,788
86,914
683,084
1,824,093
83,035
46,141
7,829
750,304
520,553
58,921
5,202,955
49,622
AC AL AM AP BA CE DF ES GO MA MG MS MT PA PB PE PI PR RJ RN RO RR RS SC SE SP TO


Columns
Customer State
Rows
AVG(Freight Value)
AVG(Price)
Sheet 6
Customer State
Avg. Freight Value
50.00
40.00
30.00
20.00
10.00
0.00
40.07
35.84
33.21
34.01
26.36
32.71
21.04
22.06
22.77
38.26
20.63
23.37
28.17
35.83
42.72
32.92
39.15
20.53
20.96
35.65
41.07
42.98
21.74
21.47
36.65
15.15
37.25
Avg. Price
200
150
100
50
0
173.73
180.89
135.50
164.32
134.60
153.76
125.77
121.91
126.27
145.20
120.75
142.63
148.30
165.69
191.48
145.51
160.36
119.00
125.12
156.97
165.97
150.57
120.34
124.65
153.04
109.65
157.53
AC AL AM AP BA CE DF ES GO MA MG MS MT PA PB PE PI PR RJ RN RO RR RS SC SE SP TO

Analysis:

Average price of orders in PB is the highest with 191.4 and lowest in SP state.

Average freight value is lowest in SP and highest in RR.

Sum of price and freight value is highest in SP state.

This implies that cost of orders and shipment are less in SP state and so people are ordering more order items and other states must follow this strategy.

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

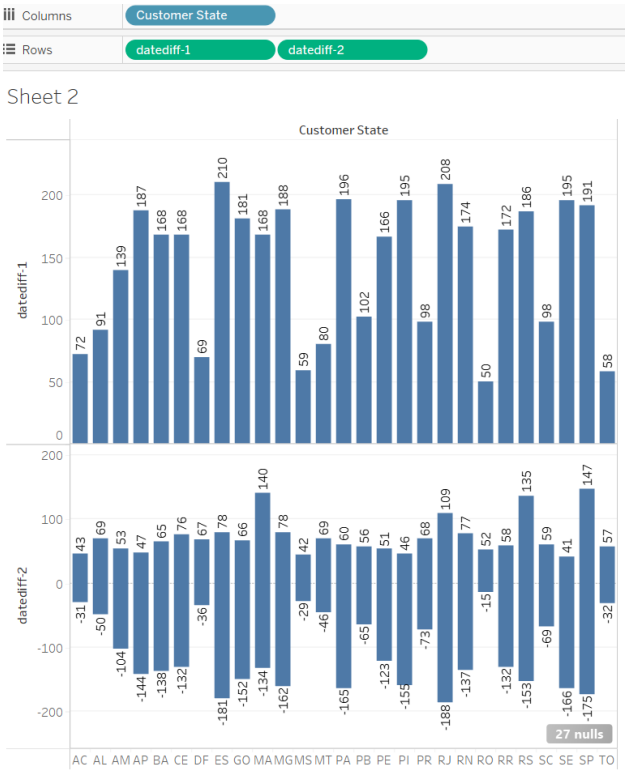
diff_estimated_delivery = order_estimated_delivery_date-order_delivered_customer_date

```
select order_id,order_purchase_timestamp,nullif(date_diff(order_delivered_customer_dat
e, order_purchase_timestamp, day),0) as time_to_delivery,
nullif(date_diff(order_estimated_delivery_date,order_delivered_customer_date,day),0) a
s diff_estimated_delivery
from
`Target_SQL.orders`
order by order_id,order_purchase_timestamp
```

Query results

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

Row	order_id	order_purchase_timestamp	time_to_delivery	diff_estimated_
1	00010242fe8c5a6d1ba2dd792...	2017-09-13 08:59:02 UTC	7	8
2	00018f77f2f0320c557190d7a1...	2017-04-26 10:53:06 UTC	16	2
3	000229ec398224ef6ca0657da...	2018-01-14 14:33:31 UTC	7	13
4	00024acbcdf0a6daa1e931b03...	2018-08-08 10:00:35 UTC	6	5
5	00042b26cf59d7ce69dfabb4e...	2017-02-04 13:57:51 UTC	25	15
6	00048cc3ae777c65dbb7d2a06...	2017-05-15 21:42:34 UTC	6	14
7	00054e8431b9d7675808bcb8...	2017-12-10 11:53:48 UTC	8	16
8	000576fe39319847cbb9d288c...	2018-07-04 12:08:27 UTC	5	15
9	0005a1a1728c9d785b8e2b08...	2018-03-19 18:40:33 UTC	9	null
10	0005f50442cb953dcd1d21e1f...	2018-07-02 13:59:39 UTC	2	18
11	00061f2a7bc09da83e415a52d...	2018-03-24 22:16:10 UTC	4	10
12	00063b381e2406b52ad42947...	2018-07-27 17:21:27 UTC	10	null
13	0006ec9db01a64e59a68b2c34...	2018-07-24 17:04:17 UTC	6	21
14	0008288aa423d2a3f00fcb17c...	2018-02-13 22:10:21 UTC	12	7



Analysis:

It is found that states RO,MS,TO,DF are taking very less days to deliver orders than other states .So, it can be said those states have higher chances of purchases .ES ,RJ,SE,PA states takes too long to deliver and their difference in actual delivery date and estimated delivery date is very high in negative.

So, they need to set up more warehouses in places where most orders comes from and avoid the delay in time of delivery.

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

```
select c.customer_state,round(avg(date_diff(o.order_delivered_customer_date, o.order_p
urchase_timestamp, day)),2) as time_to_delivery,
round(avg(date_diff(o.order_estimated_delivery_date,o.order_delivered_customer_date,da
y)),2) as diff_estimated_delivery,round(avg(oi.freight_value),2) as mean_avg_freight_v
alue
from
`Target_SQL.orders` o join `Target_SQL.order_items` oi
on o.order_id = oi.order_id
join  `Target_SQL.customers` c
on o.customer_id = c.customer_id
group by 1
```

Row	customer_state	time_to_delivery	diff_estimated_c	mean_avg_freigh
1	MT	17.51	13.64	28.17
2	MA	21.2	9.11	38.26
3	AL	23.99	7.98	35.84
4	SP	8.26	10.27	15.15
5	MG	11.52	12.4	20.63
6	PE	17.79	12.55	32.92
7	RJ	14.69	11.14	20.96
8	DF	12.5	11.27	21.04
9	RS	14.71	13.2	21.74
10	SE	20.98	9.17	36.65
11	PR	11.48	12.53	20.53
12	PA	23.3	13.37	35.83
13	BA	18.77	10.12	26.36

Analysis:

RR state takes more average time to deliver which means there is high chances that target could lose customers and it also has high freight value .so it needs to change its carrier company.

AC state takes more average difference in estimated delivery time which means it delivers products way before than estimated time.

It is also observed that SP state delivers products in very less time than other states, so more customers.

4)

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

Highest average freight value:

```
select c.customer_state,round(avg(oi.freight_value),2) as mean_avg_freight_value
from
`Target_SQL.orders` o join `Target_SQL.order_items` oi
on o.order_id = oi.order_id
join  `Target_SQL.customers` c
on o.customer_id = c.customer_id
group by 1
order by 2 desc
limit 5
```

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

Lowest average freight value:

```
select c.customer_state,round(avg(oi.freight_value),2) as mean_avg_freight_value
from
`Target_SQL.orders` o join `Target_SQL.order_items` oi
on o.order_id = oi.order_id
join  `Target_SQL.customers` c
on o.customer_id = c.customer_id
group by 1
order by 2 asc
limit 5
```

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

Analysis:

RR state takes more time to deliver so needs to change its carrier company or it will lose its customers.

On contrary, SP takes very less time to deliver.

b) Top 5 states with highest/lowest average time to delivery:

highest avg:

```
select c.customer_state,round(avg(date_diff(o.order_delivered_customer_date, o.order_p
urchase_timestamp, day)),2) as time_to_delivery
from
`Target_SQL.orders` o join `Target_SQL.order_items` oi
on o.order_id = oi.order_id
join  `Target_SQL.customers` c
on o.customer_id = c.customer_id
group by 1
order by 2 DESC
limit 5
```

Row	customer_state	time_to_delivery
1	RR	27.83
2	AP	27.75
3	AM	25.96
4	AL	23.99
5	PA	23.3

lowest avg:

```
select c.customer_state,round(avg(date_diff(o.order_delivered_customer_date, o.order_p
urchase_timestamp, day)),2) as time_to_delivery
from
`Target_SQL.orders` o join `Target_SQL.order_items` oi
on o.order_id = oi.order_id
join  `Target_SQL.customers` c
on o.customer_id = c.customer_id
group by 1
order by 2 asc
limit 5
```

Query results

JOB INFORMATION | RESULTS | JSON

Row	customer_state	time_to_delivery
1	SP	8.26
2	PR	11.48
3	MG	11.52
4	DF	12.5
5	SC	14.52

Analysis:

As said before RR takes more time and SP takes very less time to deliver.

C) 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,o.order_de
livered_customer_date,day)),2) as diff_estimated_delivery
from
`Target_SQL.orders` o join `Target_SQL.order_items` oi
on o.order_id = oi.order_id
join  `Target_SQL.customers` c
on o.customer_id = c.customer_id
group by 1
order by 2 DESC
limit 5
```

Row	customer_state	diff_estimated_
1	AC	20.01
2	RO	19.08
3	AM	18.98
4	AP	17.44
5	RR	17.43

```
select c.customer_state,round(avg(date_diff(o.order_estimated_delivery_date,o.order_de
livered_customer_date,day)),2) as diff_estimated_delivery
from
`Target_SQL.orders` o join `Target_SQL.order_items` oi
on o.order_id = oi.order_id
join  `Target_SQL.customers` c
on o.customer_id = c.customer_id
group by 1
order by 2
limit 5
```

Query results

JOB INFORMATION | RESULTS | JSON

Row	customer_state	diff_estimated_c
1	AL	7.98
2	MA	9.11
3	SE	9.17
4	ES	9.77
5	BA	10.12

Analysis:

AC has happy customers since average diff in estimated date is high so very fast delivery where as AL has low average diff in estimated delivery so more time to deliver.

6. Payment type analysis:

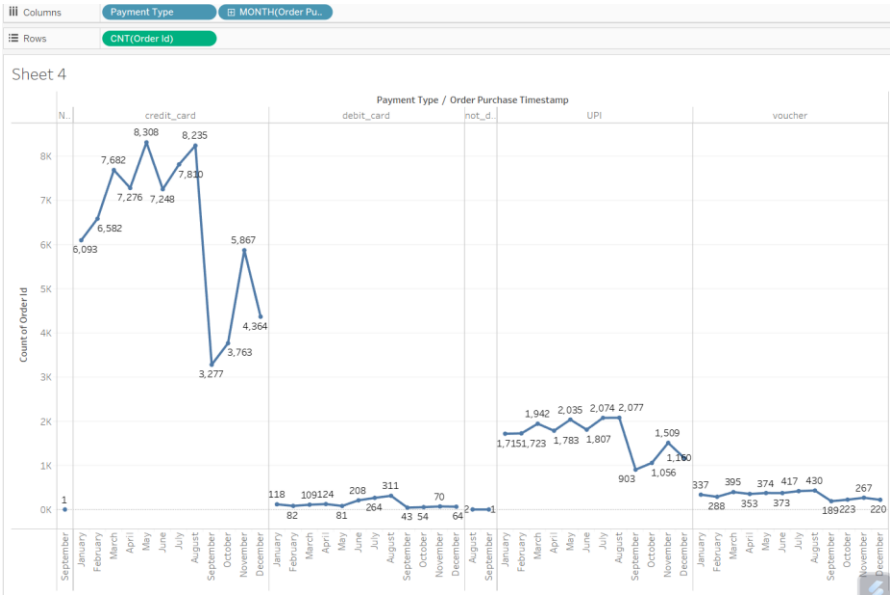
1) Month over Month count of orders for different payment types:

```
SELECT distinct p.payment_type,extract(month from o.order_purchase_timestamp) as month
,count(o.order_purchase_timestamp) as count_orders
FROM
`Target_SQL.orders` o JOIN `Target_SQL.payments` p
on o.order_id = p.order_id
group by 1,2
order by 1,2
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DET.

Row	payment_type	month	count_orders
1	UPI	1	1715
2	UPI	2	1723
3	UPI	3	1942
4	UPI	4	1783
5	UPI	5	2035
6	UPI	6	1807
7	UPI	7	2074
8	UPI	8	2077
9	UPI	9	903
10	UPI	10	1056
11	UPI	11	1509
12	UPI	12	1160
13	credit_card	1	6103
14	credit_card	2	6609
15	credit_card	3	7707
16	credit_card	4	7301
17	credit_card	5	8350



Analysis:

Credit card is in top of other payment methods followed by UPI , Voucher,debit_card.

It is found that no of purchases is topped in the month of August and is increasing from first half of year and is decreasing from September for every payment methods.

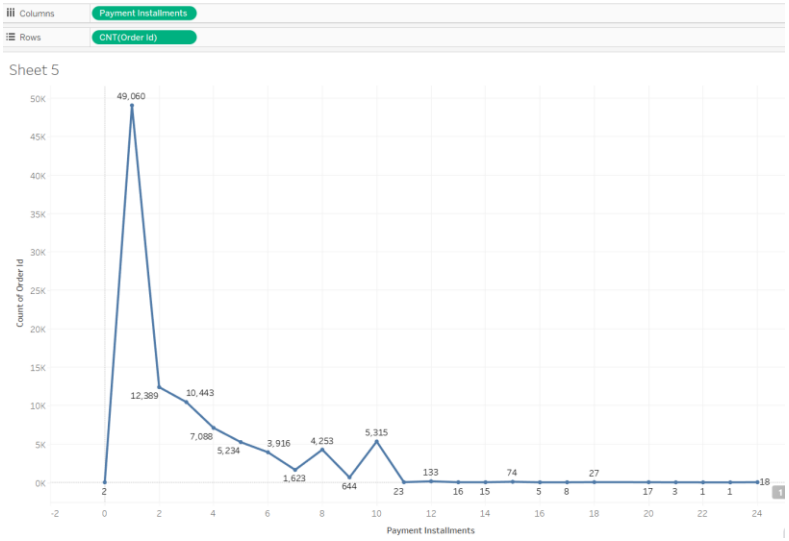
2) Count of orders based on the no. of payment installments:

```
SELECT p.payment_installments,count(o.order_purchase_timestamp) as count_orders
FROM
`Target_SQL.orders` o JOIN `Target_SQL.payments` p
on o.order_id = p.order_id
group by 1
order by 1,2
```

Query results

JOB INFORMATION | RESULTS

Row	payment_installments	count_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
11	10	5328
12	11	23
13	12	133
14	13	16
15	14	15
16	15	74



Analysis:

With increase in no of payment installments , count of orders is decreasing and becomes stable after 23 rd installment.So, people in Brazil are settling payments for order in 2nd installment and after that no of orders is drastically reduced.