Section 1

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

1. Data type of columns in a table

Query used:

```
SELECT column_name, data_type
FROM `targetsql.INFORMATION_SCHEMA.COLUMNS`
WHERE table_name = 'geolocation';
```

Sample Results:

Row	column_name	data_type
1	geolocation_zip_code_prefix	INT64
2	geolocation_lat	FLOAT64
3	geolocation_lng	FLOAT64
4	geolocation_city	STRING
5	geolocation_state	STRING

Comments:

1. The Project was created in Big Query & the datasets shared were uploaded.
2. Similar queries were used to explore & check the data types of other tables.

Section 1

1. Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset
    2. Time period for which the data is given

Query used:

```
SELECT MIN(extract(YEAR FROM order_purchase_timestamp)) as min_year,
MAX(extract(YEAR FROM order_purchase_timestamp)) as max_year
FROM `targetsql.orders`;
```

Sample Results:

Row	min_year	max_year
1	2016	2018

Section 1

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

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

Query used:

```
select c.customer_city,c.customer_state
from `targetsql.orders` o
join `targetsql.customer` c on o.customer_id=c.customer_id;
```

Sample Results:

customer_city	customer_state
abadia dos dourados	MG
abadiania	GO
abaete	MG
abaetetuba	PA
abaiara	CE
abaira	BA
abare	BA
abatia	PR
abdon batista	SC

Section 2

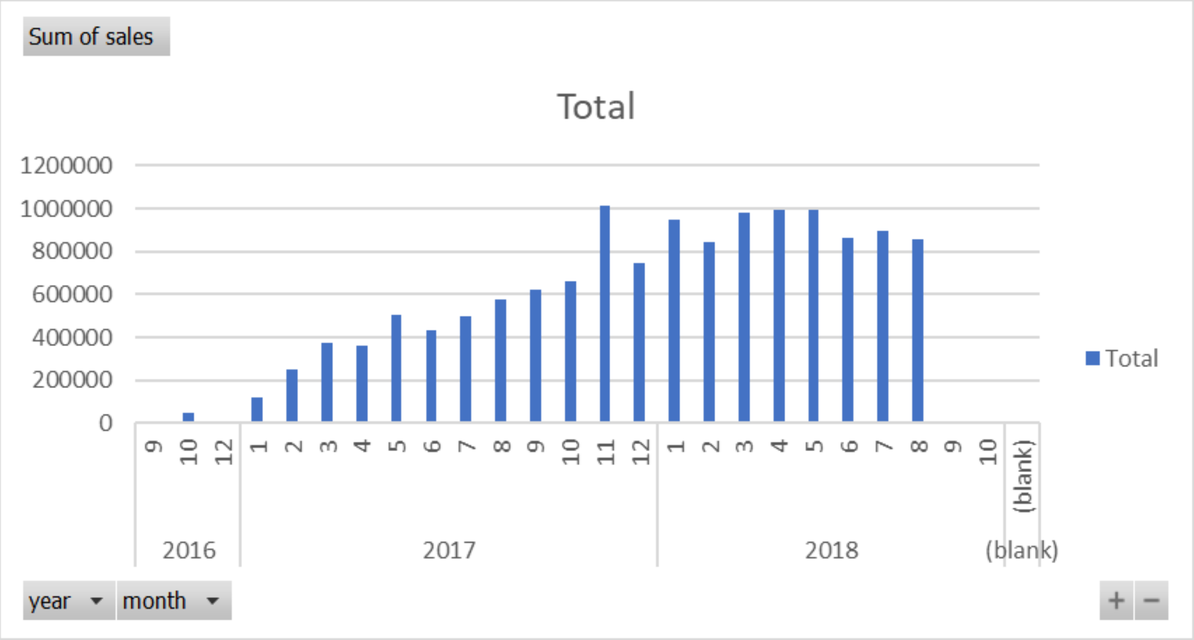
2. In-depth Exploration:
    1. Is there a growing trend on e-commerce in Brazil? How can we describe a complete scenario? Can we see some seasonality with peaks at specific months?

Query used:

```
select extract(year from date(o.order_purchase_timestamp)) as year,
extract(month from date(o.order_purchase_timestamp)) as month,
round(sum(oi.price),2) as sales
from `targetsql.orders` o
left join `targetsql.order_items`  oi on oi.order_id=o.order_id
group by year, month
order by year, month
```

Sample Results:

year	month	sales
2016	9	267.36
2016	10	49507.66
2016	12	10.9
2017	1	120312.9
2017	2	247303
2017	3	374344.3
2017	4	359927.2
2017	5	506071.1
2017	6	433038.6
2017	7	498031.5



Comments:

1. The trend does looks like it's growing.
    1. Specially through out the 2017.
    2. However, in 2018, the trend looks more or less stagnant
2. There is no indication of any seasonality.

Section 2

2. In-depth Exploration: (Approach 1)
    2. What time do Brazilian customers tend to buy (Dawn, Morning, Afternoon or Night)?

Query used:

```
select t1.timeslot, count(*) as order_count
from
(select *,
case
  when time(o.order_purchase_timestamp) between '00:00:00' and '05:59:59'
  then 'Dawn'
  when time(o.order_purchase_timestamp) between '06:00:00' and '11:59:59'
  then 'Morning'
  when time(o.order_purchase_timestamp) between '12:00:00' and '17:59:59'
  then 'Afternoon'
  when time(o.order_purchase_timestamp) between '18:00:00' and '23:59:59'
  then 'Night'
  else 'Invalid time slot'
end as timeslot
from `targetsql.orders` o)t1
group by t1.timeslot
```

Comments:

1. The question was solved using two approaches.
2. This is the first approach where total number for all years & month were found out

Sample Results:

Row	timeslot	order_count
1	Morning	22240
2	Dawn	4740
3	Afternoon	38361
4	Night	34100

Section 2

2. In-depth Exploration: (Approach 2)
    2. What time do Brazilian customers tend to buy (Dawn, Morning, Afternoon or Night)?

Query used:

```
select extract(year from date(t1.order_purchase_timestamp)) as year,
extract(month from date(t1.order_purchase_timestamp)) as month,
t1.timeslot, count(*) as order_count
from
(select *,
case
  when time(o.order_purchase_timestamp) between '00:00:00' and '05:59:59'
  then 'Dawn'
  when time(o.order_purchase_timestamp) between '06:00:00' and '11:59:59'
  then 'Morning'
  when time(o.order_purchase_timestamp) between '12:00:00' and '17:59:59'
  then 'Afternoon'
  when time(o.order_purchase_timestamp) between '18:00:00' and '23:59:59'
  then 'Night'
  else 'Invalid time slot'
end as timeslot
from `targetsql.orders` o
)t1
group by year, month, t1.timeslot
order by year, month, t1.timeslot
```

Comments:

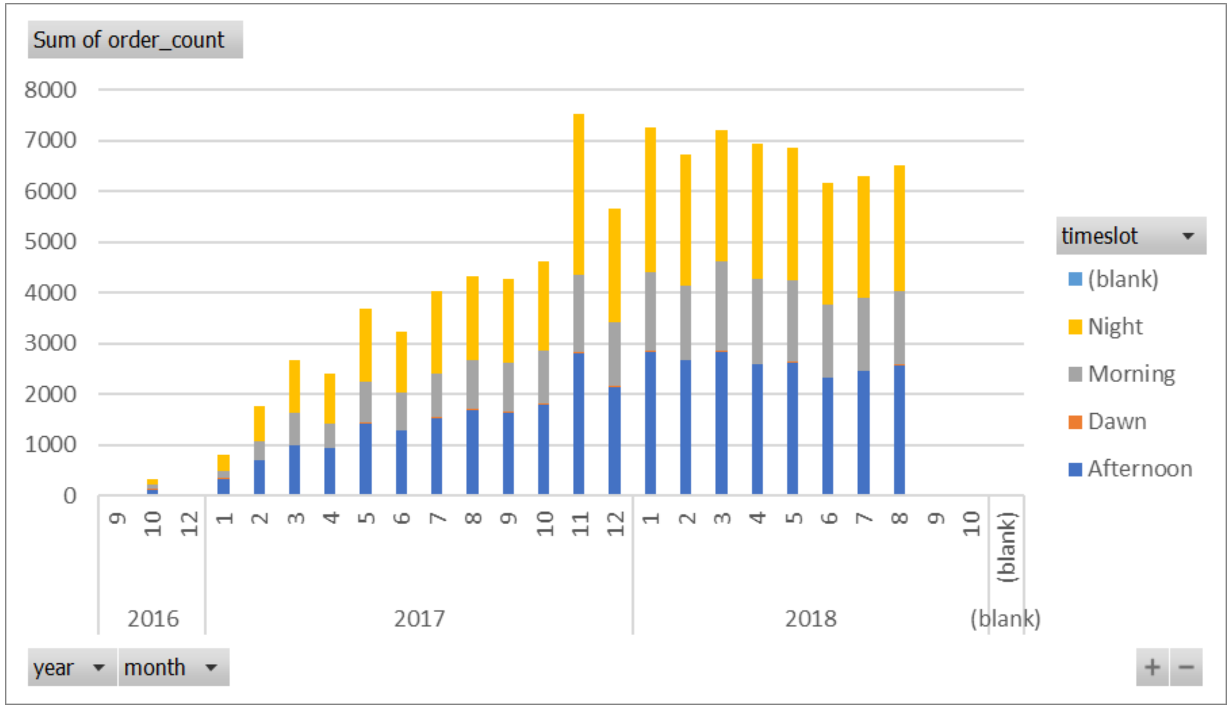
1. The question was solved using two approaches.
2. This is the second approach where total order count for each year & month were found out.
3. It seems that most customers tend to buy around night or afternoon.
4. However, most buying happens during after noon.

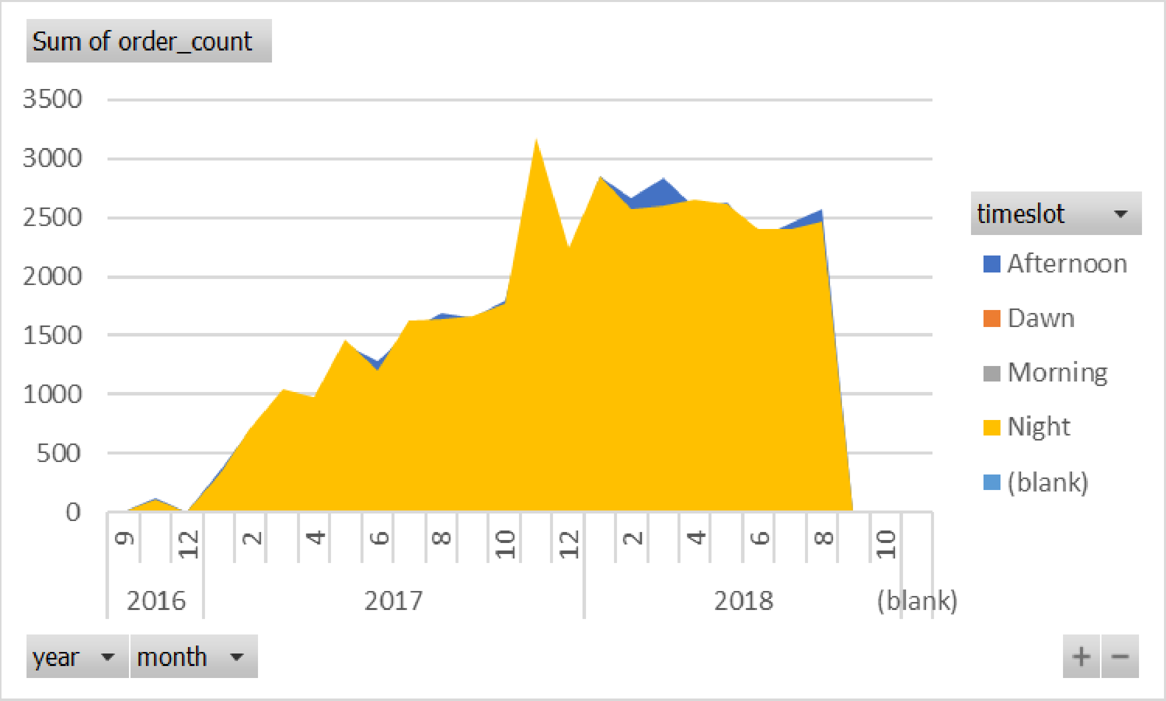
Section 2

2. In-depth Exploration: (Approach 2)
    2. What time do Brazilian customers tend to buy (Dawn, Morning, Afternoon or Night)?

Sample Results:

year	month	timeslot	order_count
2016	9	Afternoon	2
2016	9	Night	2
2016	10	Afternoon	125
2016	10	Dawn	2
2016	10	Morning	84
2016	10	Night	113
2016	12	Night	1
2017	1	Afternoon	339
2017	1	Dawn	3
2017	1	Morning	137





Section 3

3. Evolution of E-commerce orders in the Brazil region:
    1. Get month on month orders by states

Query used:

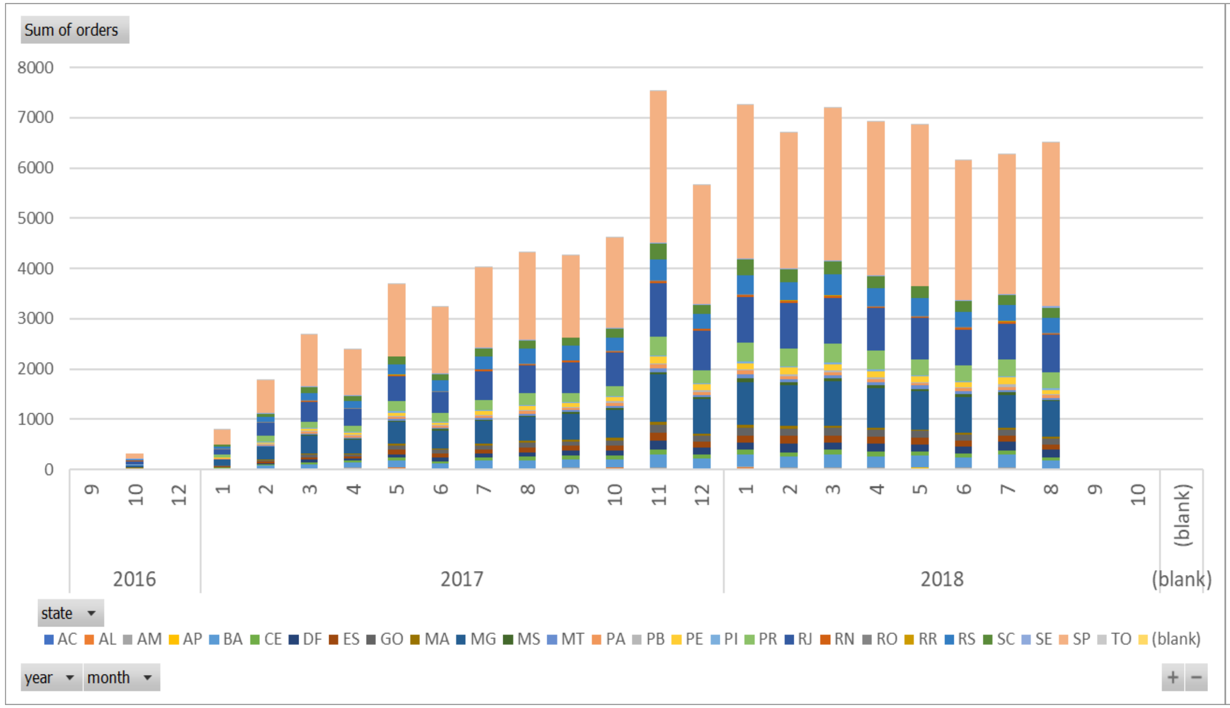
```
select extract(year from date(o.order_purchase_timestamp)) as year,
extract(month from date(o.order_purchase_timestamp)) as month,
c.customer_state as state,
count(distinct o.order_id) as orders

from `targetsql.orders` o
join `targetsql.customer` c on c.customer_id=o.customer_id

group by year, month, state
order by year, month, state
```

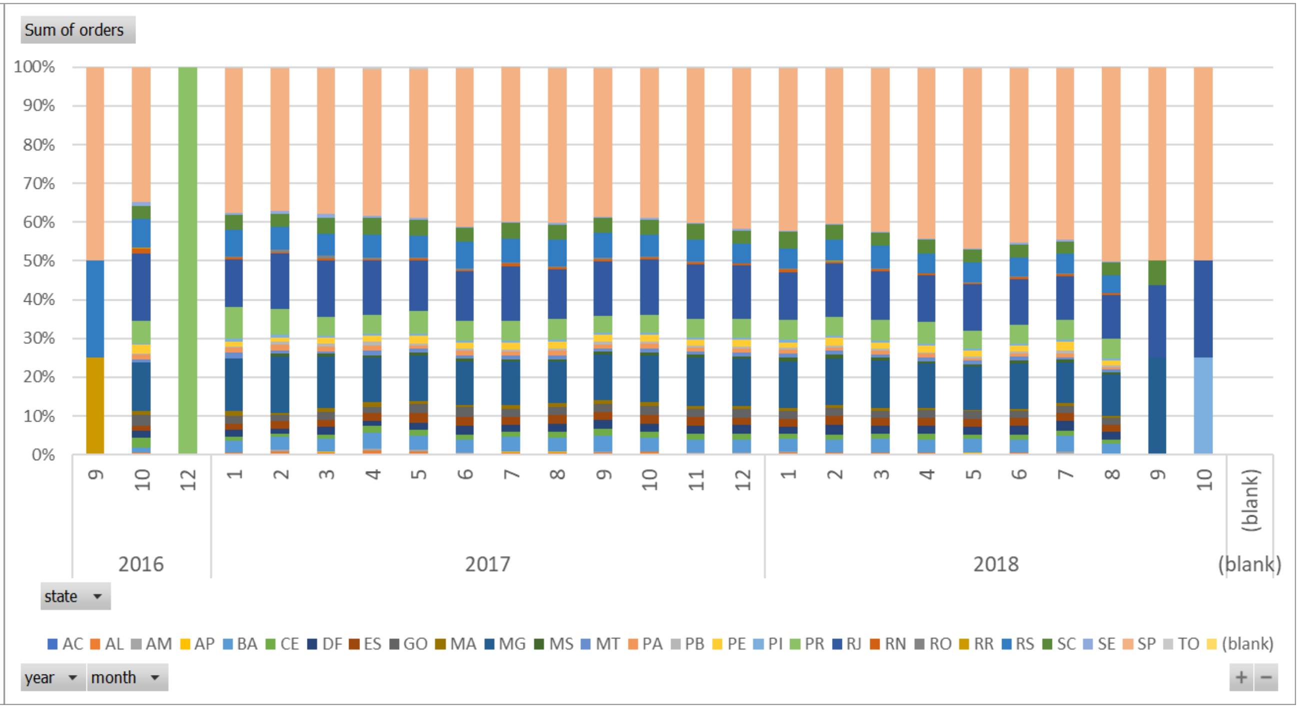
Section 3

3. Evolution of E-commerce orders in the Brazil region:
    1. Get month on month orders by states

Sample Results:

year	month	state	orders
2016	9	RR	1
2016	9	RS	1
2016	9	SP	2
2016	10	AL	2
2016	10	BA	4
2016	10	CE	8
2016	10	DF	6
2016	10	ES	4
2016	10	GO	9
2016	10	MA	4





Section 3

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

    2. Distribution of customers across the states in Brazil

Query used:

```
select customer_state as state, count(distinct customer_id) as customer_count
from `targetsql.customer`
group by customer_state
```

Sample Results:

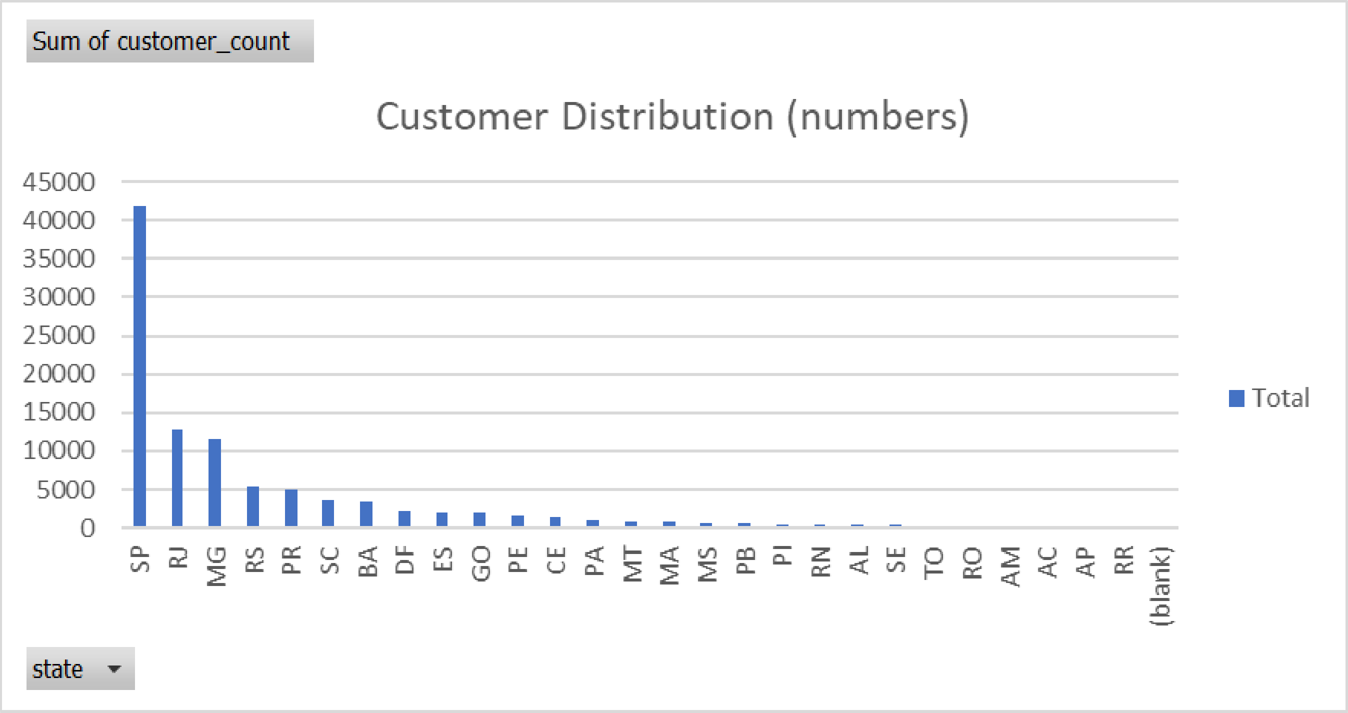
state	customer_count
SP	41746
RJ	12852
MG	11635
RS	5466
PR	5045
SC	3637
BA	3380
DF	2140
ES	2033
GO	2020

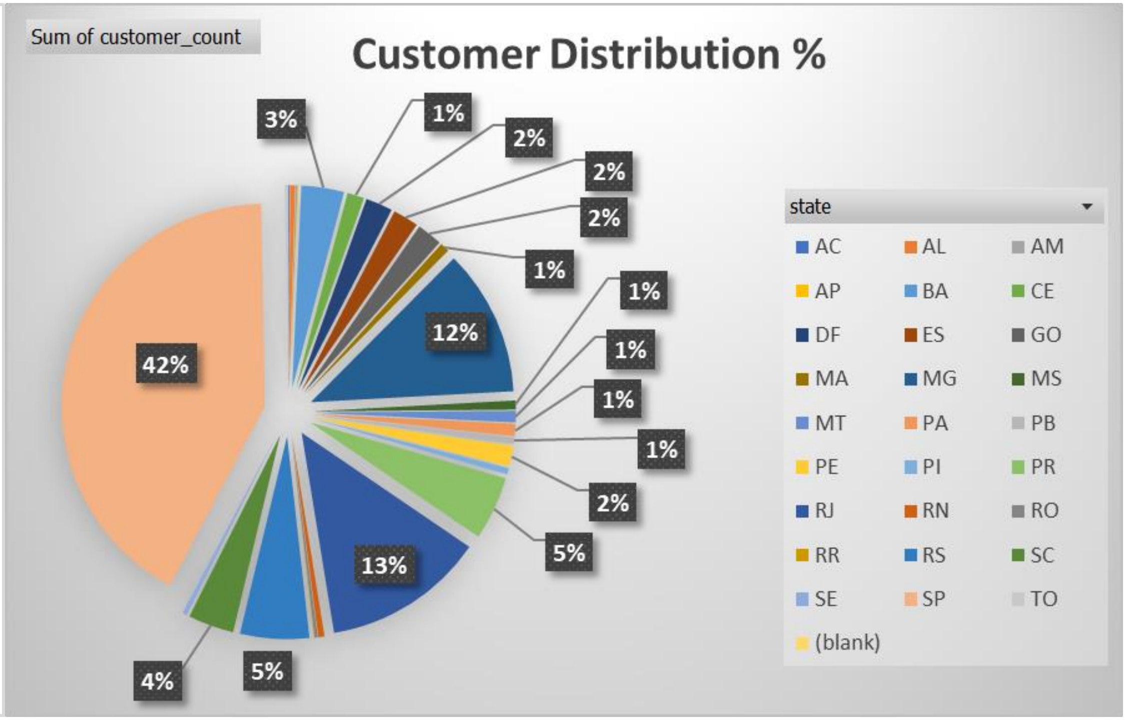
Section 3

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

    2. Distribution of customers across the states in Brazil

Sample Results:





Section 4

4. Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others.
   1. Get % increase in cost of orders from 2017 to 2018 (include months between Jan to Aug only) - You can use "payment_value" column in payments table

Query used: (to obtain cost for each year)

```
select extract(year from date(o.order_purchase_timestamp)) as year,
round(sum(p.payment_value),2) as cost
from `targetsql.orders` o
join `targetsql.payments` p on p.order_id=o.order_id
where extract(month from date(o.order_purchase_timestamp)) between 1 and 8
group by year
```

Sample Results:

year	cost
2018	8694733.84
2017	3669022.12

Section 4

4. Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others.
   1. Get % increase in cost of orders from 2017 to 2018 (include months between Jan to Aug only) - You can use "payment_value" column in payments table

Query used: (to obtain % increase)

```
select round((cost_2018-cost_2017)*100/cost_2017,2) as pc_inc_in_cost
from
(select extract(year from date(o.order_purchase_timestamp)) as year,
round(sum(p.payment_value),2) as cost_2017
from `targetsql.orders` o
join `targetsql.payments` p on p.order_id=o.order_id
where extract(month from date(o.order_purchase_timestamp)) between 1 and 8
group by year
having year=2017)t1,

(select extract(year from date(o.order_purchase_timestamp)) as year,
round(sum(p.payment_value),2) as cost_2018
from `targetsql.orders` o
join `targetsql.payments` p on p.order_id=o.order_id
where extract(month from date(o.order_purchase_timestamp)) between 1 and 8
group by year
having year=2018)t2
```

Sample Results:

pc_inc_in_cost
136.98

Section 4

4. Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others.
    2. Mean & Sum of price and freight value by customer state

Query used:

```
select c.customer_state as state,
round(avg(oi.price),2) as mean_price,
round(sum(oi.price),2) as sum_price,
round(avg(oi.freight_value),2) as mean_freight_value,
round(sum(oi.freight_value),2) as sum_freight_value,
from `targetsql.orders` o
join `targetsql.order_items` oi on oi.order_id=o.order_id
join `targetsql.customer`c on c.customer_id=o.customer_id
group by state
order by state
```

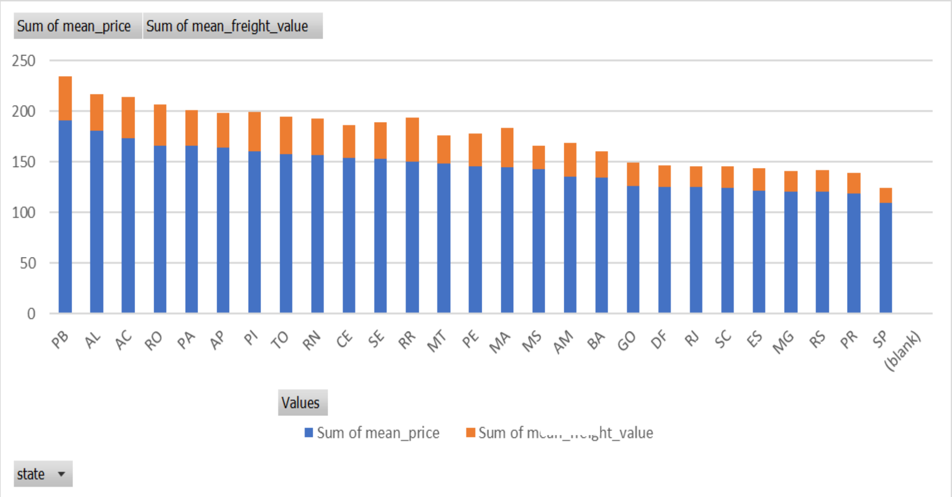
Sample Results:

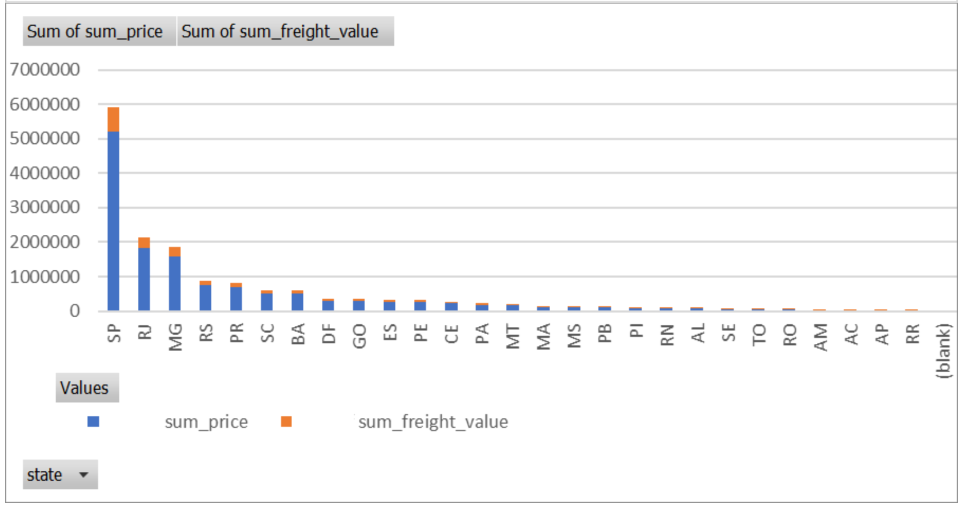
state	mean_price	sum_price	mean_freight_value	sum_freight_value
AC	173.73	15982.95	40.07	3686.75
AL	180.89	80314.81	35.84	15914.59
AM	135.5	22356.84	33.21	5478.89
AP	164.32	13474.3	34.01	2788.5
BA	134.6	511349.99	26.36	100156.68
CE	153.76	227254.71	32.71	48351.59
DF	125.77	302603.94	21.04	50625.5
ES	121.91	275037.31	22.06	49764.6
GO	126.27	294591.95	22.77	53114.98
MA	145.2	119648.22	38.26	31523.77

Section 4

4. Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others.
    2. Mean & Sum of price and freight value by customer state

Sample Results:





Section 5

5. Analysis on sales, freight and delivery time
    1.Calculate days between purchasing, delivering and estimated delivery

Query used:

```
select
order_id,
t1.purchasing,
t1.delivering,
t1.estimated_delivery,
date_diff(delivering,purchasing,day) as delivery_pur,
date_diff(estimated_delivery,delivering,day) as est_delivery
from
(
select
distinct o.order_id,
date(o.order_purchase_timestamp) as purchasing,
date(o.order_delivered_customer_date) as delivering,
date(o.order_estimated_delivery_date) as estimated_delivery
from `targetsql.orders` o)t1
order by delivery_pur desc,est_delivery desc
```

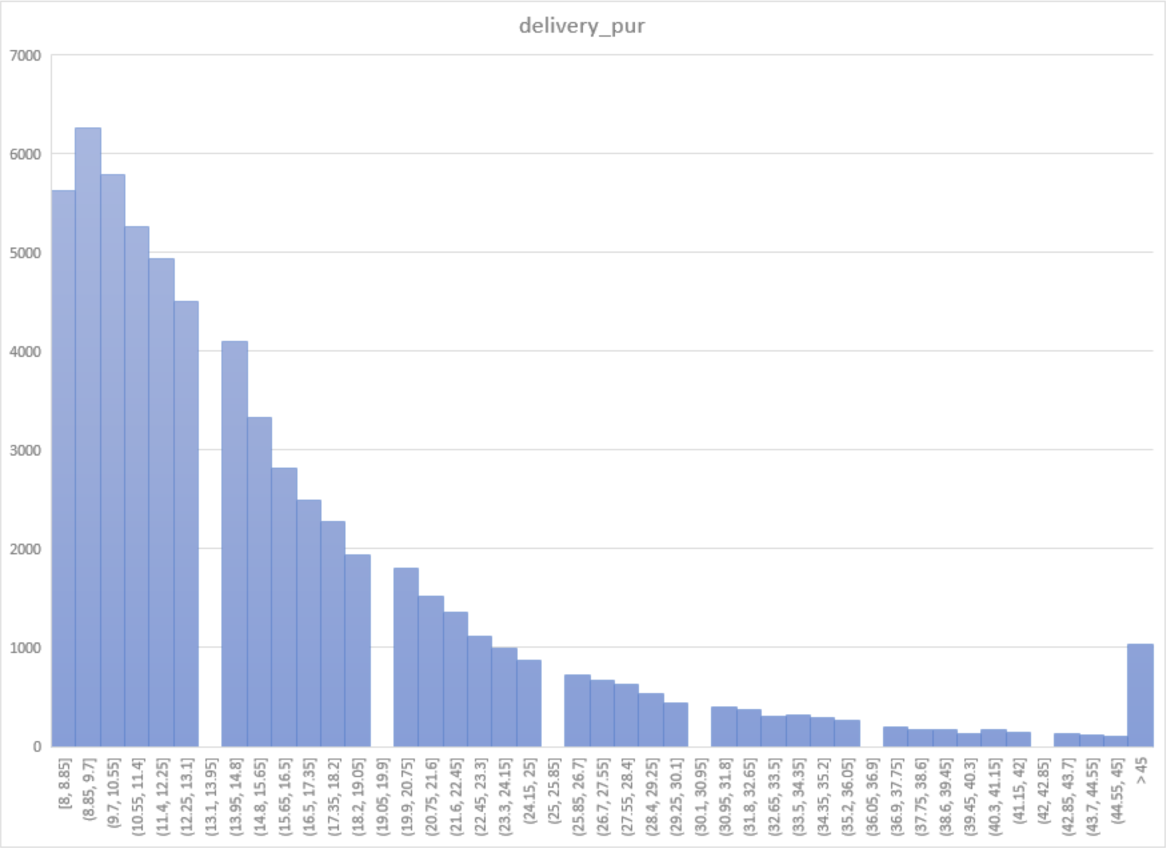
Sample Results:

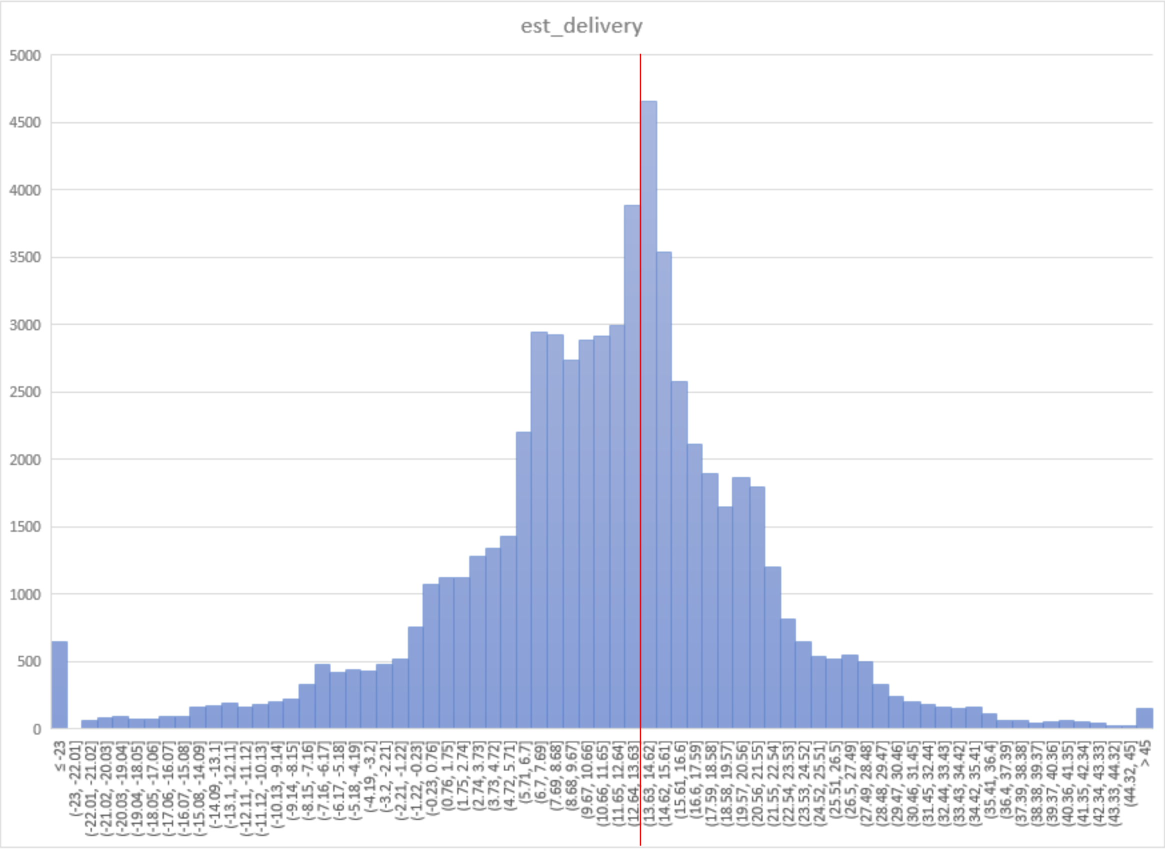
order_id	purchasing	delivering	estimated_delivery	delivery_pur	est_delivery
ca07593549f1816d26a572e06dc1eab6	2/21/2017	9/19/2017	3/22/2017	210	-181
1b3190b2dfa9d789e1f14c05b647a14a	2/23/2018	9/19/2018	3/15/2018	208	-188
440d0d17af552815d15a9e41abe49359	3/7/2017	9/19/2017	4/7/2017	196	-165
2fb597c2f772eca01b1f5c561bf6cc7b	3/8/2017	9/19/2017	4/17/2017	195	-155
285ab9426d6982034523a855f55a885e	3/8/2017	9/19/2017	4/6/2017	195	-166
0f4519c5f1c541ddec9f21b3bddd533a	3/9/2017	9/19/2017	4/11/2017	194	-161
47b40429ed8cce3aee9199792275433f	1/3/2018	7/13/2018	1/19/2018	191	-175
2fe324febf907e3ea3f2aa9650869fa5	3/13/2017	9/19/2017	4/5/2017	190	-167
2d7561026d542c8dbd8f0daeadf67a43	3/15/2017	9/19/2017	4/13/2017	188	-159
c27815f7e3dd0b926b58552628481575	3/15/2017	9/19/2017	4/10/2017	188	-162

Section 5

5. Analysis on sales, freight and delivery time
    2. Find time_to_delivery & diff_estimated_delivery. Formula for the same given below:
        1. time_to_delivery = order_purchase_timestamp - order_delivered_customer_date
        2. diff_estimated_delivery = order_estimated_delivery_date - order_delivered_customer_date

Sample Results:





Section 5

5. Analysis on sales, freight and delivery time
    3. Group data by state, take mean of freight_value, time_to_delivery, diff_estimated_delivery
    4. Sort the data to get the following:
    5. Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5
    6. Top 5 states with highest/lowest average time to delivery
    7. Top 5 states where delivery is really fast/ not so fast compared to estimated date

```
with t1 as
(select distinct o.order_id,c.customer_state as state,
oi.freight_value,
date(o.order_purchase_timestamp) as purchasing,
date(o.order_delivered_customer_date) as delivering,
date(o.order_estimated_delivery_date) as estimated_delivery
from `targetsql.orders` o
left join `targetsql.order_items` oi on oi.order_id=o.order_id
left join `targetsql.customer` c on c.customer_id=o.customer_id)

select t1.state,
round(avg(t1.freight_value),2) as avg_freight_value,
round(avg(date_diff(delivering,purchasing,day)),2) as avg_delivery_pur,
round(avg(date_diff(estimated_delivery,delivering,day)),2) as avg_est_delivery
from t1
group by t1.state
order by t1.state
```

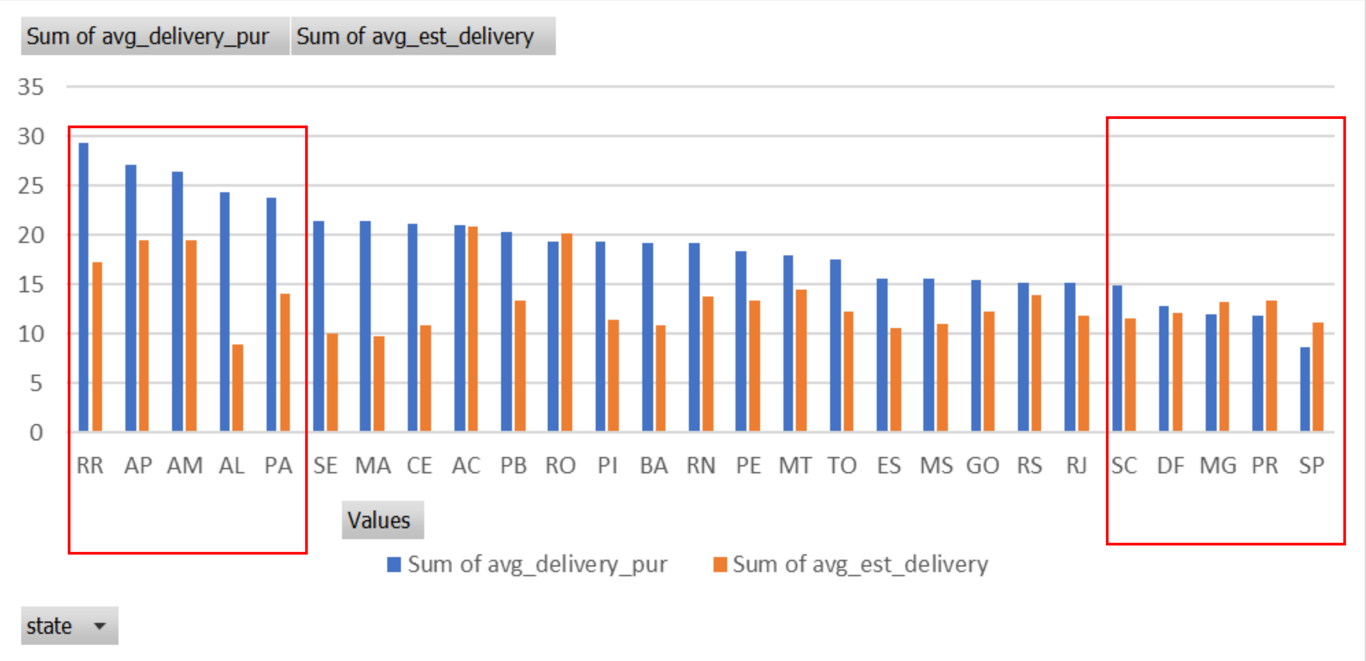
Section 5

5. Analysis on sales, freight and delivery time
    3. Group data by state, take mean of freight_value, time_to_delivery, diff_estimated_delivery
    4. Sort the data to get the following:
    5. Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5
    6. Top 5 states with highest/lowest average time to delivery
    7. Top 5 states where delivery is really fast/ not so fast compared to estimated date

state	avg_freight_value	avg_delivery_pur	avg_est_delivery
AC	41.3	20.98	20.89
AL	36.15	24.31	8.87
AM	33.17	26.37	19.54
AP	34.98	27.15	19.53
BA	26.27	19.24	10.89
CE	33.02	21.12	10.92
DF	21.32	12.82	12.16
ES	22.11	15.64	10.58
GO	23.05	15.45	12.24

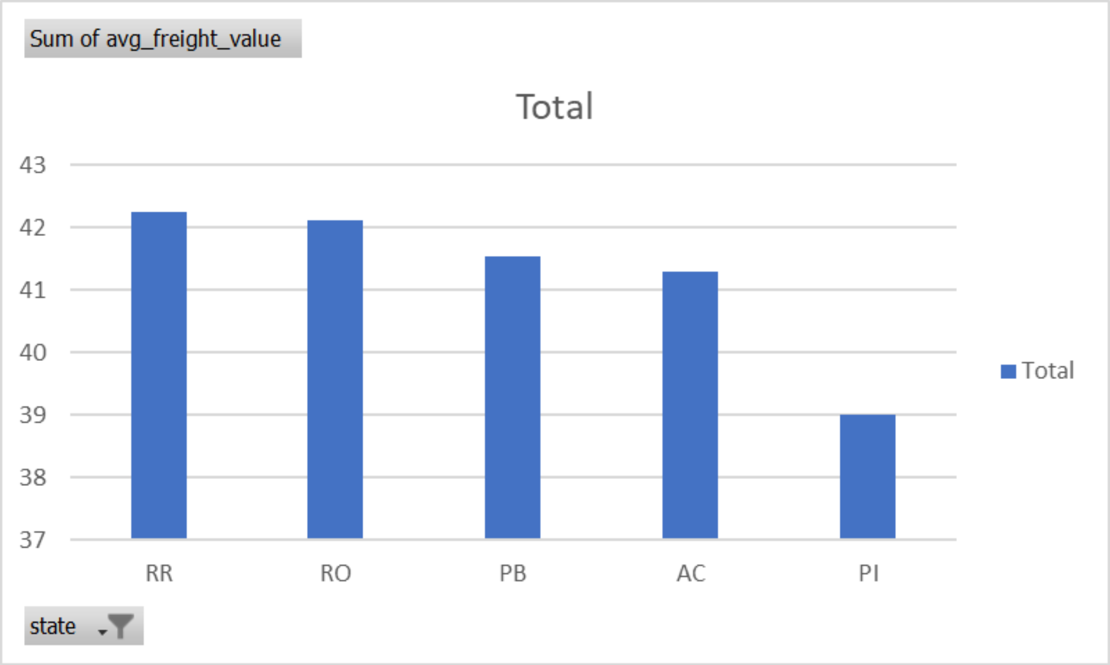
Section 5

5. Analysis on sales, freight and delivery time
    3. Group data by state, take mean of freight_value, time_to_delivery, diff_estimated_delivery
    4. Sort the data to get the following:
    5. Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5
    6. Top 5 states with highest/lowest average time to delivery
    7. Top 5 states where delivery is really fast/ not so fast compared to estimated date



Section 5

5. Analysis on sales, freight and delivery time
    3. Group data by state, take mean of freight_value, time_to_delivery, diff_estimated_delivery
    4. Sort the data to get the following:
    5. Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5
    6. Top 5 states with highest/lowest average time to delivery
    7. Top 5 states where delivery is really fast/ not so fast compared to estimated date



Row Labels	Sum of avg_freight_value
RR	42.26
RO	42.13
PB	41.55
AC	41.3
PI	39.01

Section 5

5. Analysis on sales, freight and delivery time
    3. Group data by state, take mean of freight_value, time_to_delivery, diff_estimated_delivery
    4. Sort the data to get the following:
    5. Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5
    6. Top 5 states with highest/lowest average time to delivery
    7. Top 5 states where delivery is really fast/ not so fast compared to estimated date

Row Labels	Sum of avg_delivery_pur	Sum of avg_est_delivery
RR	29.34	17.29
AP	27.15	19.53
AM	26.37	19.54
AL	24.31	8.87
PA	23.79	14.08

Section 6

6. Payment type analysis:

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

Query used:

```
select extract(year from date(order_purchase_timestamp)) as year,
extract(month from date(order_purchase_timestamp))as month,
p.payment_type,
count(o.order_id) as order_count
from `targetsql.orders` o
left join `targetsql.payments` p on p.order_id=o.order_id
group by year, month, p.payment_type
order by year, month, p.payment_type
```

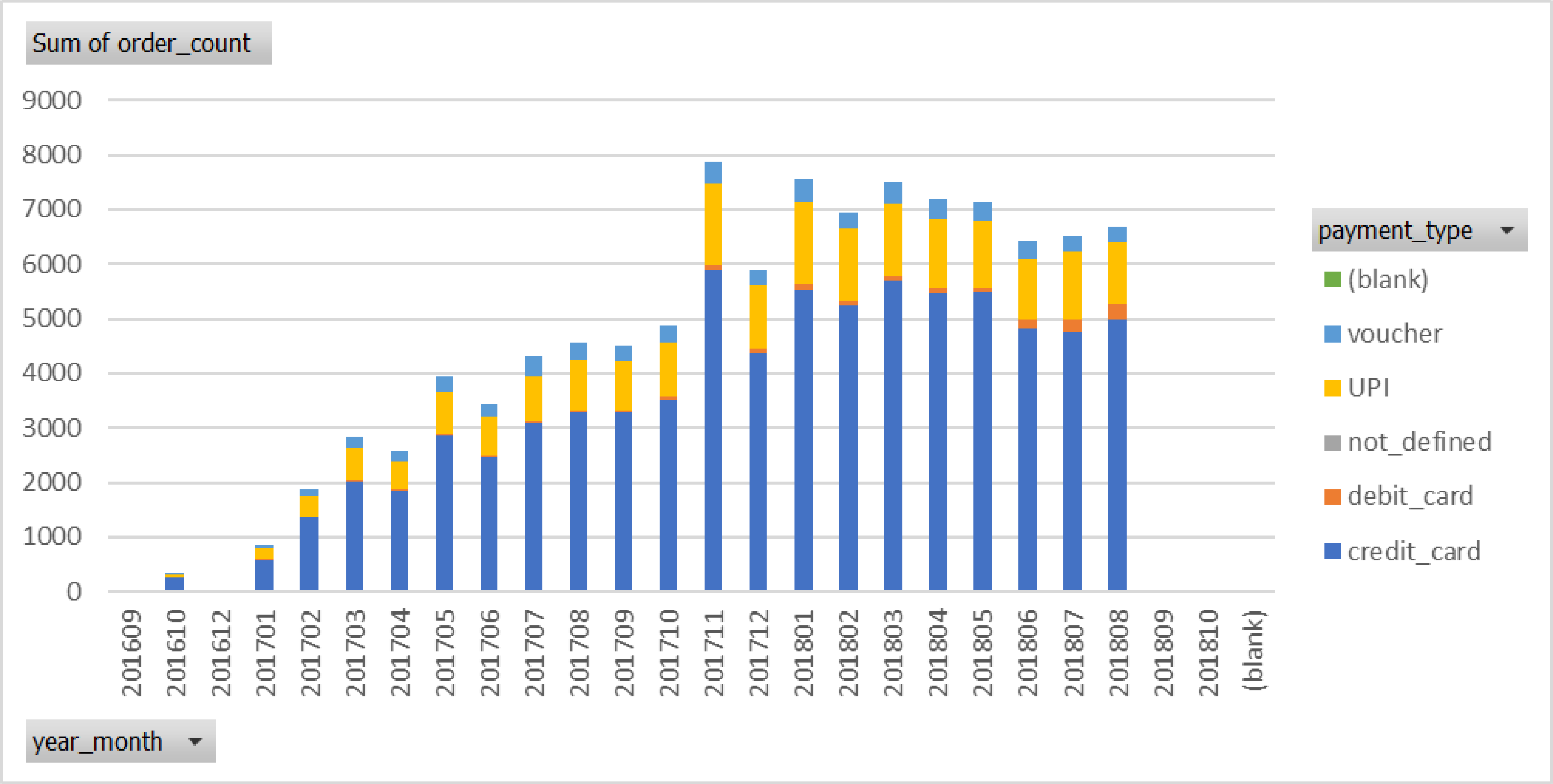
Sample Results:

year	month	payment_type	order_count
2016	9		1
2016	9	credit_card	3
2016	10	UPI	63
2016	10	credit_card	254
2016	10	debit_card	2
2016	10	voucher	23
2016	12	credit_card	1
2017	1	UPI	197
2017	1	credit_card	583
2017	1	debit_card	9

Section 6

6. Payment type analysis:

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



Section 6

6. Payment type analysis:

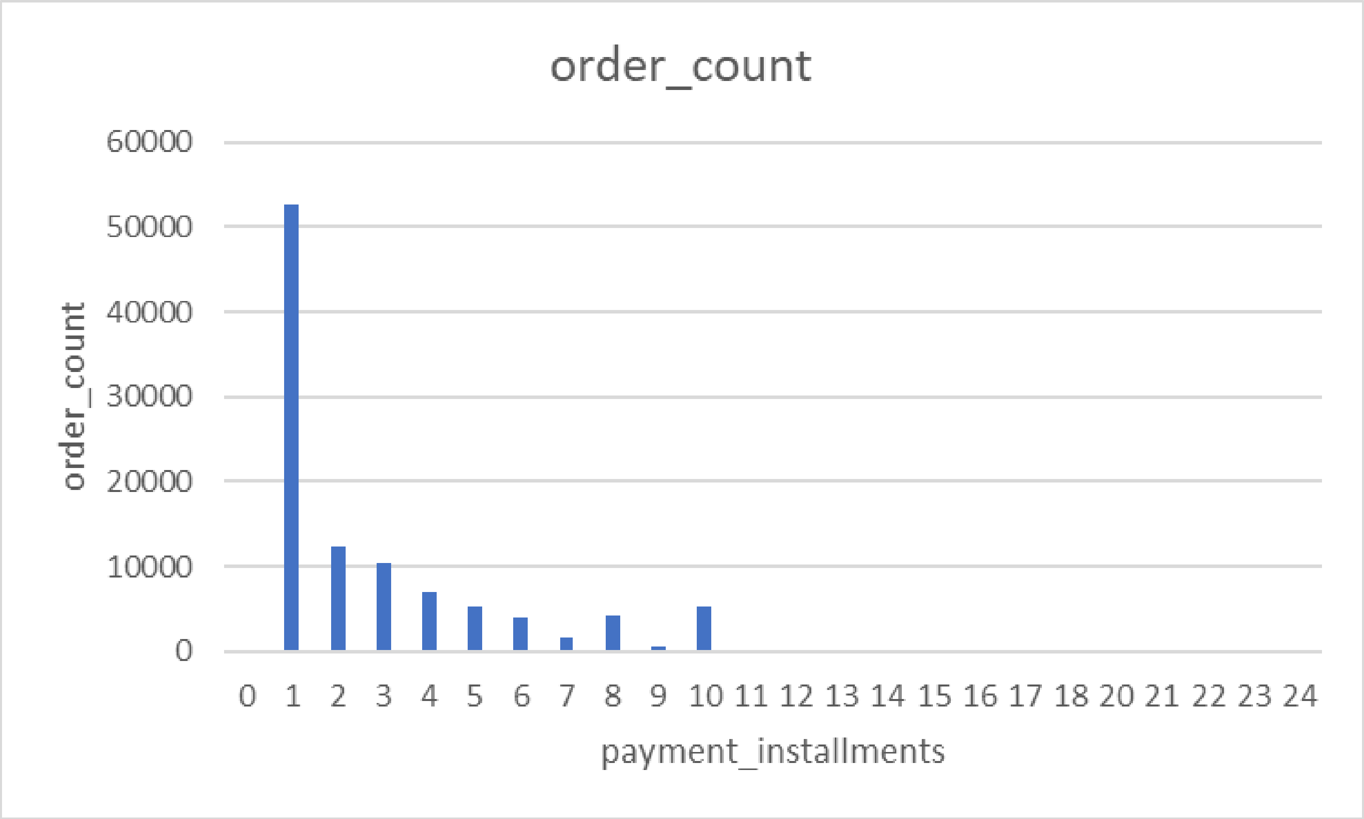
   2. Count of orders based on the no. of payment installments

Query used:

```
select payment_installments, count(order_id) as order_count
from `targetsql.payments`
group by payment_installments
order by payment_installments
```

Sample Results:

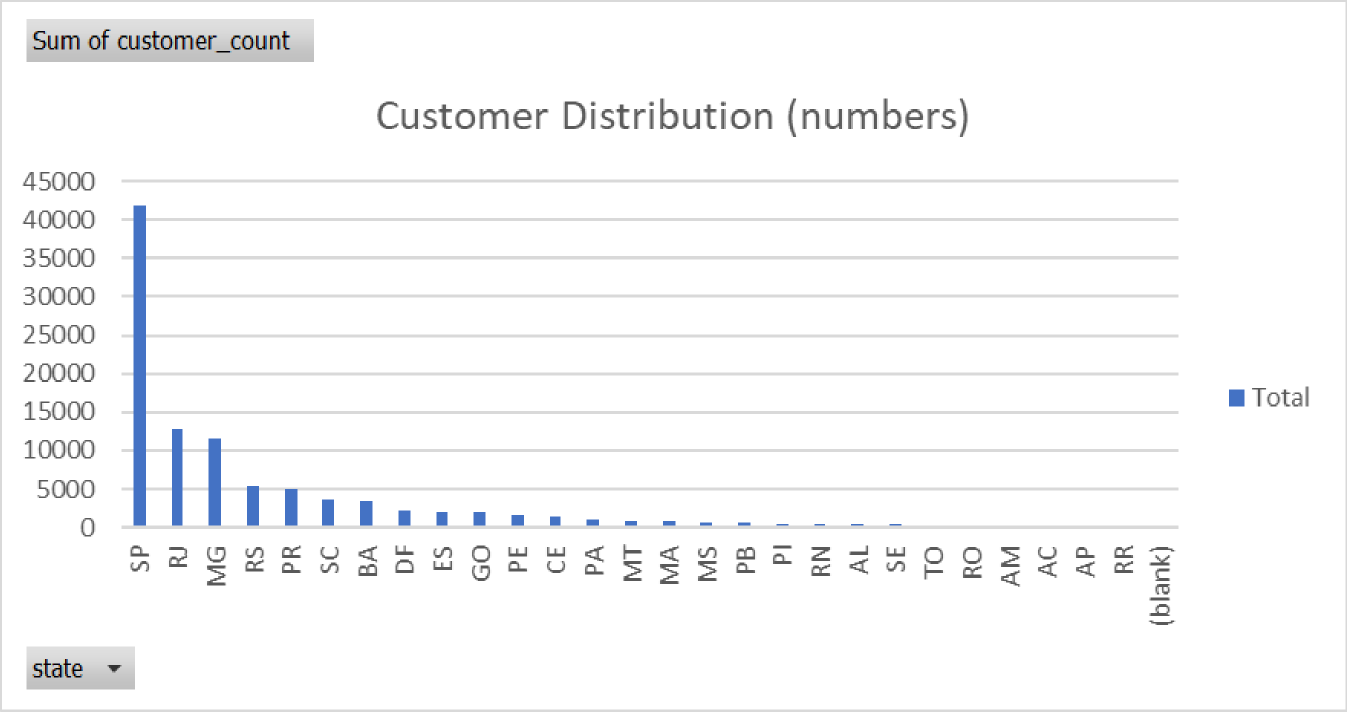
payment_installments	order_count
0	2
1	52546
2	12413
3	10461
4	7098
5	5239
6	3920
7	1626
8	4268

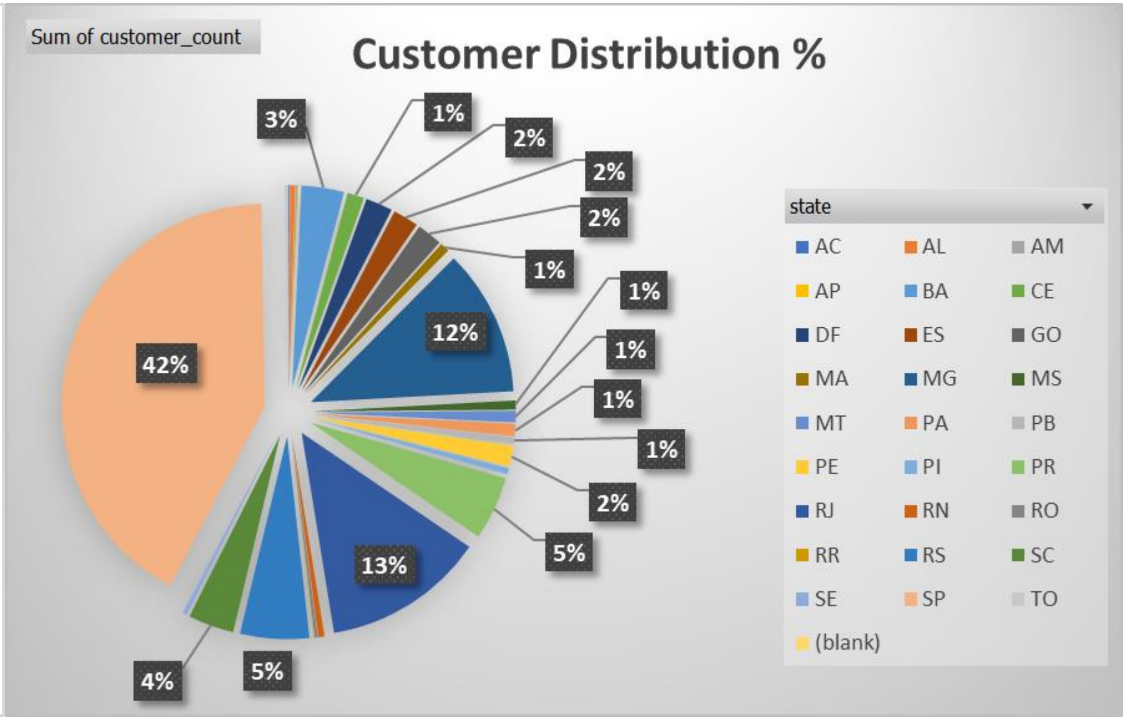


Section 7 - Actionable Insights

1. Need to improve the customer base in other states.

The customer distribution chart (shown below from Q.3.2.), clearly indicates that the customer base in other states (except for top 3 – i.e.. SP, RJ, MG) is very small.



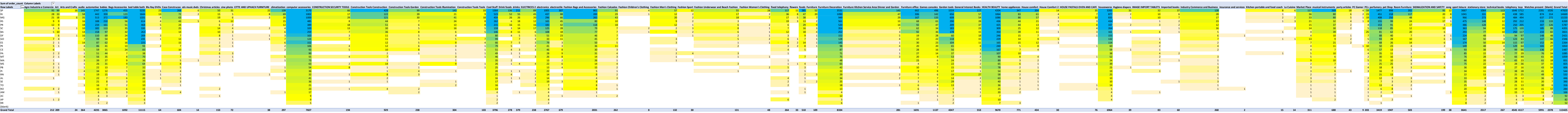


Section 7 - Actionable Insights

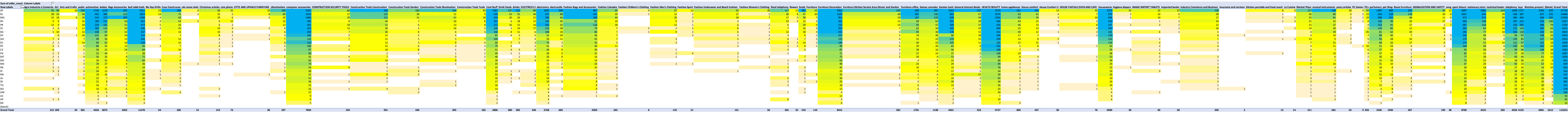
2. Need more seller in other states

I plotted count of orders & count of sellers with product category in the horizontal axis & states in the vertical axis .
The color signature clearly indicates that the count of orders is low in other states due to lower count of sellers present in the other states. (Queries used in the next slide)

Count of orders



Count of sellers



Product categories

States

Section 7 - Actionable Insights

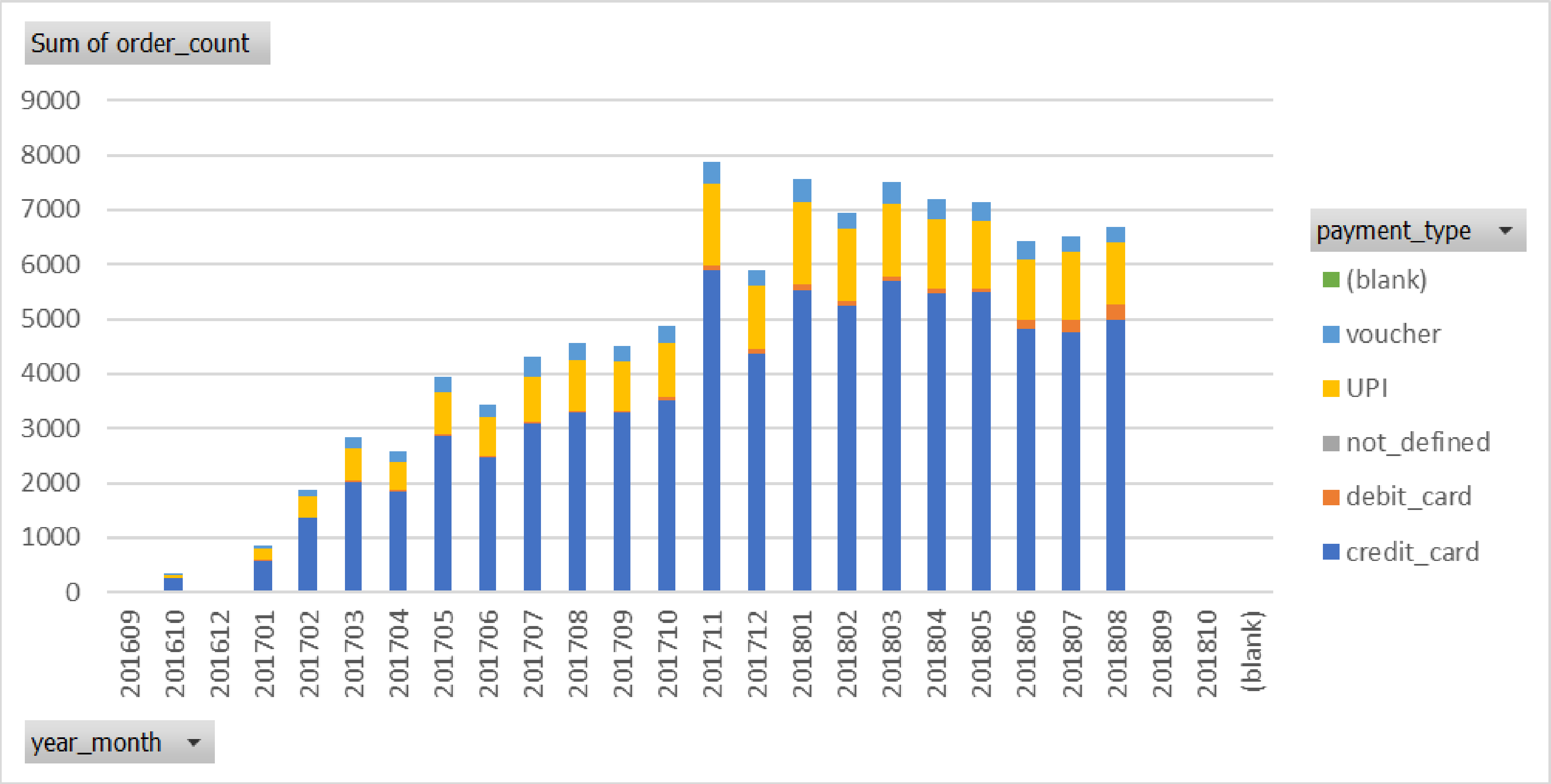
2. Need more seller in other states

```
select c.customer_state, p.product_category, count(s.seller_id) as seller_count
from `targetsql.orders` o
left join `targetsql.order_items` oi on oi.order_id=o.order_id
left join `targetsql.products` p on p.product_id=oi.product_id
left join `targetsql.customer` c on c.customer_id=o.customer_id
left join `targetsql.order_reviews` o_r on o_r.order_id=o.order_id
left join `targetsql.sellers` s on s.seller_id=oi.seller_id
group by c.customer_state, p.product_category
order by c.customer_state, p.product_category

select c.customer_state, p.product_category, count(o.order_id) as order_count
from `targetsql.orders` o
left join `targetsql.order_items` oi on oi.order_id=o.order_id
left join `targetsql.products` p on p.product_id=oi.product_id
left join `targetsql.customer` c on c.customer_id=o.customer_id
group by c.customer_state, p.product_category
order by c.customer_state, p.product_category
```

Section 7 - Actionable Insights

3. Need to use more UPI

The chart below clearly shows that UPI option is being underused a lot specially when compared to the credit card.



Section 8 - Recommendation

Drawing from the insights pointed out in the section 7 following are the recommendations:

1. Improve the customer base in other states.
2. Improve the sellers in other states.
3. Encourage among the sellers the usage of UPI as payment method