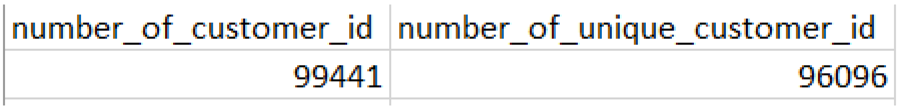
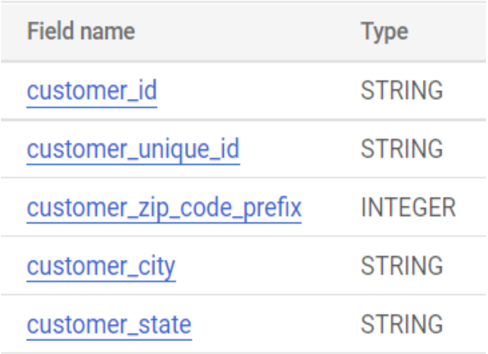
E-Commerce SQL

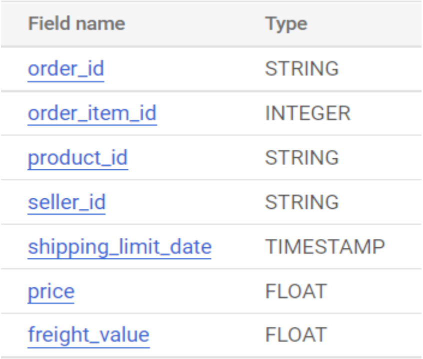
1. Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset
   1. Data type of columns in a table



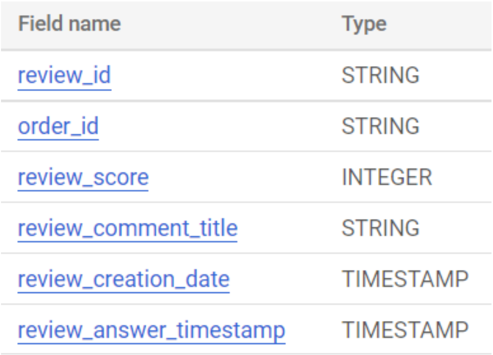
number_of_customer_id	number_of_unique_customer_id
99441	96096



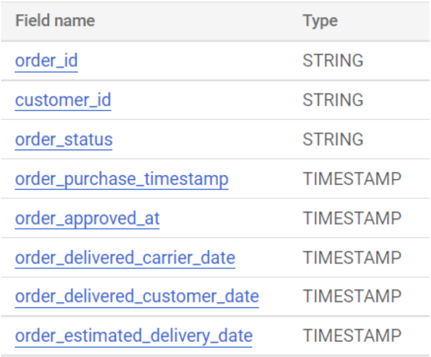
Field name	Type
customer_id	STRING
customer_unique_id	STRING
customer_zip_code_prefix	INTEGER
customer_city	STRING
customer_state	STRING



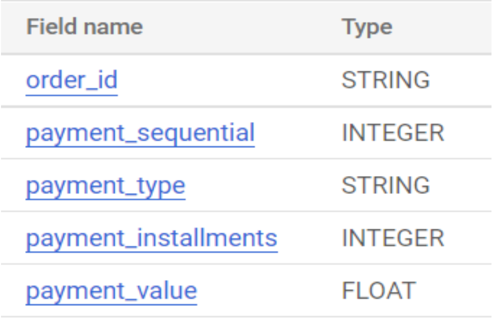
Field name	Type
order_id	STRING
order_item_id	INTEGER
product_id	STRING
seller_id	STRING
shipping_limit_date	TIMESTAMP
price	FLOAT
freight_value	FLOAT



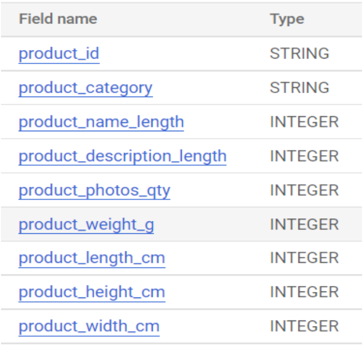
Field name	Type
review_id	STRING
order_id	STRING
review_score	INTEGER
review_comment_title	STRING
review_creation_date	TIMESTAMP
review_answer_timestamp	TIMESTAMP



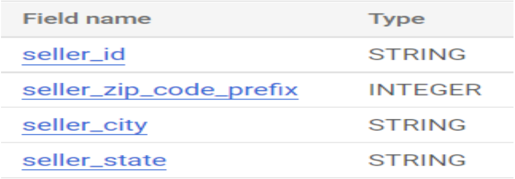
Field name	Type
order_id	STRING
customer_id	STRING
order_status	STRING
order_purchase_timestamp	TIMESTAMP
order_approved_at	TIMESTAMP
order_delivered_carrier_date	TIMESTAMP
order_delivered_customer_date	TIMESTAMP
order_estimated_delivery_date	TIMESTAMP



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

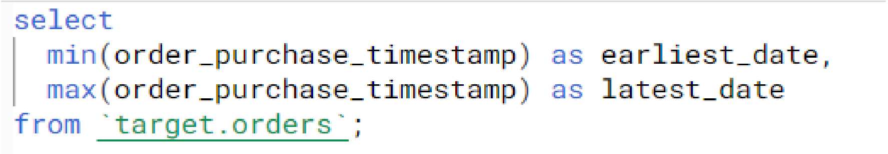


Field name	Type
product_id	STRING
product_category	STRING
product_name_length	INTEGER
product_description_length	INTEGER
product_photos_qty	INTEGER
product_weight_g	INTEGER
product_length_cm	INTEGER
product_height_cm	INTEGER
product_width_cm	INTEGER

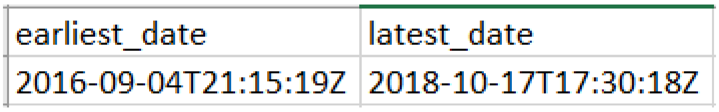


Field name	Type
seller_id	STRING
seller_zip_code_prefix	INTEGER
seller_city	STRING
seller_state	STRING

2. Time period for which the data is given

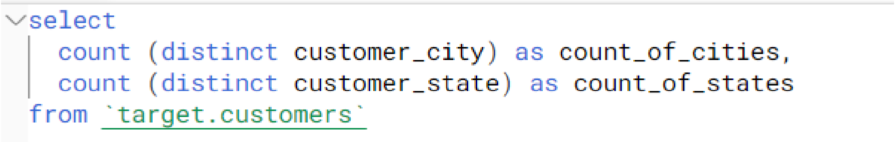


```
select
  min(order_purchase_timestamp) as earliest_date,
  max(order_purchase_timestamp) as latest_date
from `target.orders`;
```

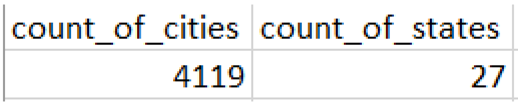


earliest_date	latest_date
2016-09-04T21:15:19Z	2018-10-17T17:30:18Z

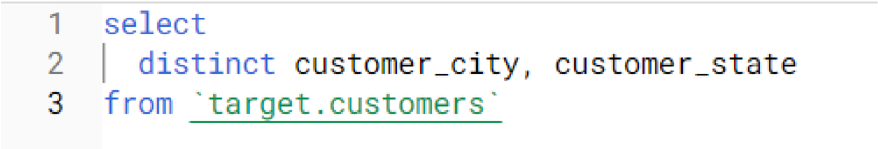
3. Cities and States covered in the dataset



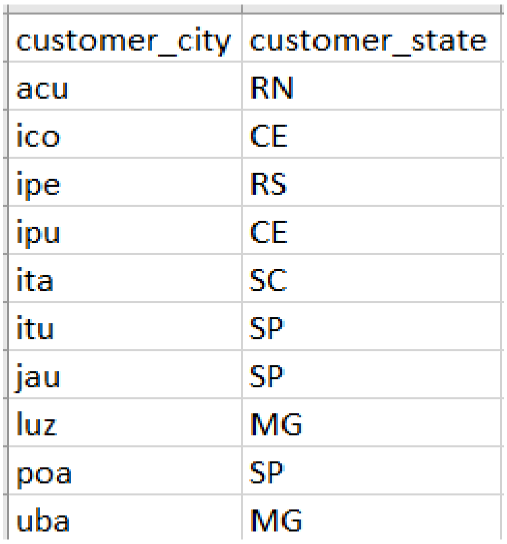
```
select
  count (distinct customer_city) as count_of_cities,
  count (distinct customer_state) as count_of_states
from `target.customers`
```



count_of_cities	count_of_states
4119	27

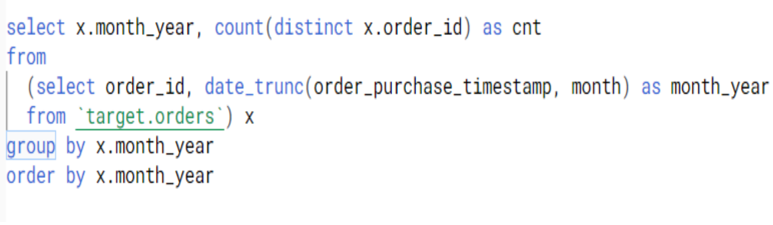


```
select
  distinct customer_city, customer_state
from `target.customers`
```

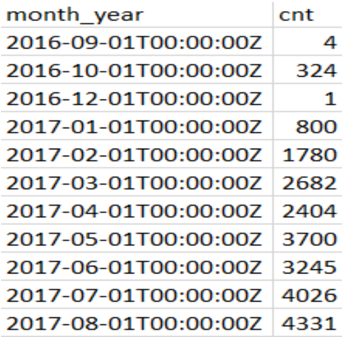


customer_city	customer_state
acu	RN
ico	CE
ipe	RS
ipu	CE
ita	SC
itu	SP
jau	SP
luz	MG
poa	SP
uba	MG

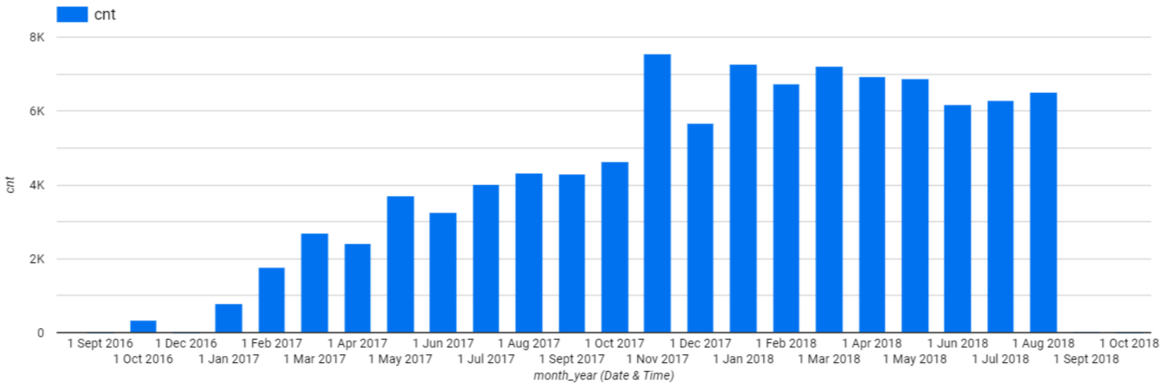
2. In-depth Exploration:
    1. Is there a growing trend on e-commerce in Brazil? How can we describe a complete scenario? Can we see some seasonality with peaks at specific months?
       - There is a clear growth in e-commerce since 2016 as shown in the graphs below
       - Nov’17 sees the highest number of orders, significantly higher than previous month suggesting high volume of orders placed during holiday season (Shopping for Christmas)

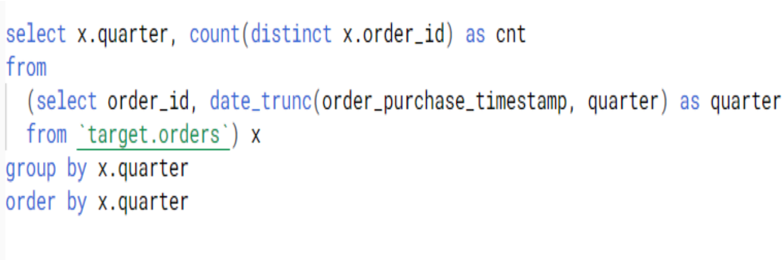


```
select x.month_year, count(distinct x.order_id) as cnt
from
  (select order_id, date_trunc(order_purchase_timestamp, month) as month_year
  from `target.orders`) x
group by x.month_year
order by x.month_year
```

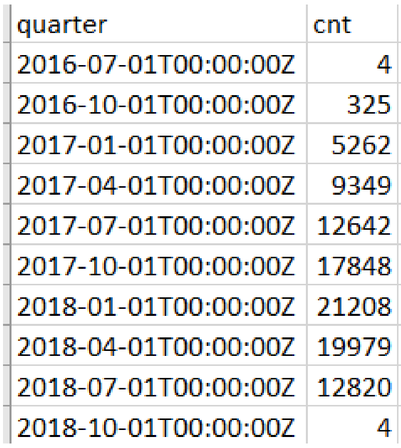


month_year	cnt
2016-09-01T00:00:00Z	4
2016-10-01T00:00:00Z	324
2016-12-01T00:00:00Z	1
2017-01-01T00:00:00Z	800
2017-02-01T00:00:00Z	1780
2017-03-01T00:00:00Z	2682
2017-04-01T00:00:00Z	2404
2017-05-01T00:00:00Z	3700
2017-06-01T00:00:00Z	3245
2017-07-01T00:00:00Z	4026
2017-08-01T00:00:00Z	4331

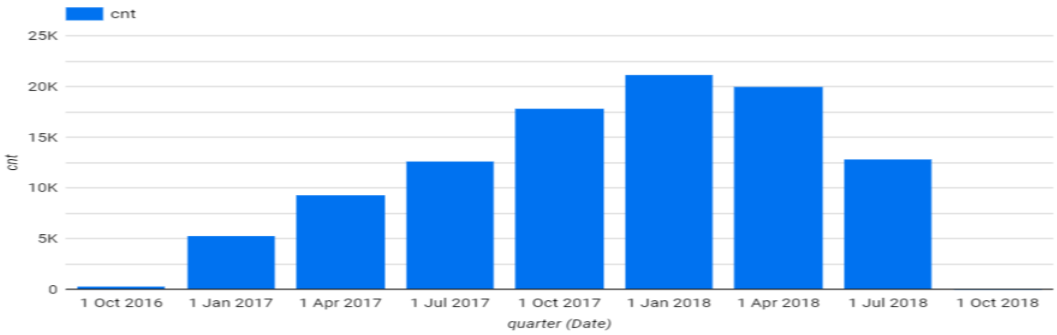


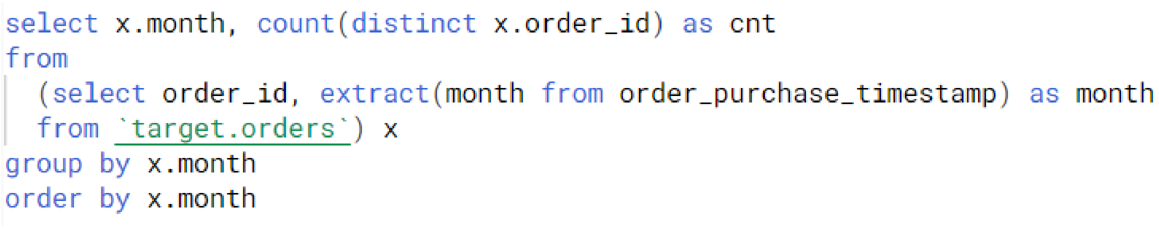


```
select x.quarter, count(distinct x.order_id) as cnt
from
  (select order_id, date_trunc(order_purchase_timestamp, quarter) as quarter
  from `target.orders`) x
group by x.quarter
order by x.quarter
```

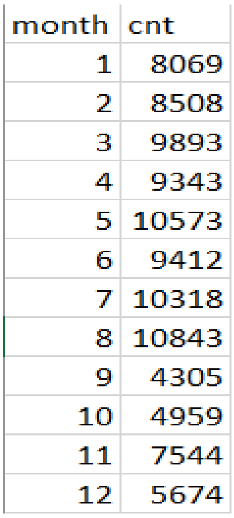


quarter	cnt
2016-07-01T00:00:00Z	4
2016-10-01T00:00:00Z	325
2017-01-01T00:00:00Z	5262
2017-04-01T00:00:00Z	9349
2017-07-01T00:00:00Z	12642
2017-10-01T00:00:00Z	17848
2018-01-01T00:00:00Z	21208
2018-04-01T00:00:00Z	19979
2018-07-01T00:00:00Z	12820
2018-10-01T00:00:00Z	4

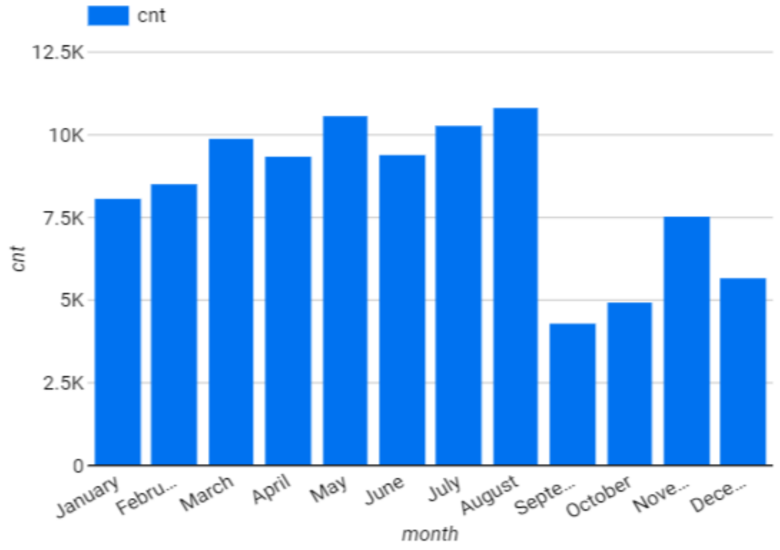




```
select x.month, count(distinct x.order_id) as cnt
from
  (select order_id, extract(month from order_purchase_timestamp) as month
  from `target.orders`) x
group by x.month
order by x.month
```

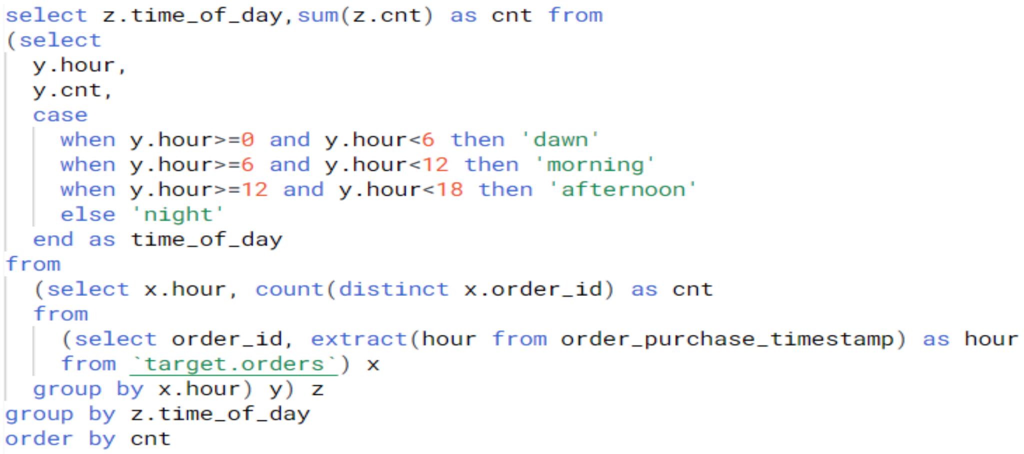


month	cnt
1	8069
2	8508
3	9893
4	9343
5	10573
6	9412
7	10318
8	10843
9	4305
10	4959
11	7544
12	5674

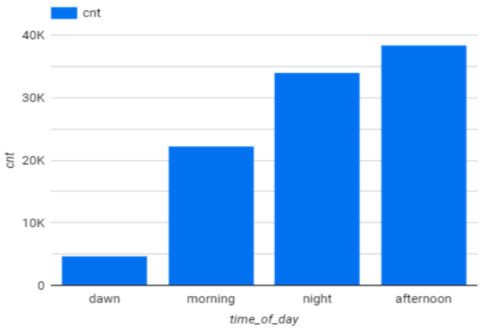


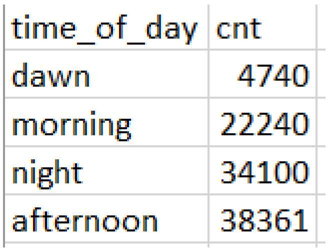
2. What time do Brazilian customers tend to buy (Dawn, Morning, Afternoon or Night)?

- Brazilian customers shop the most in the afternoon and on weekdays.

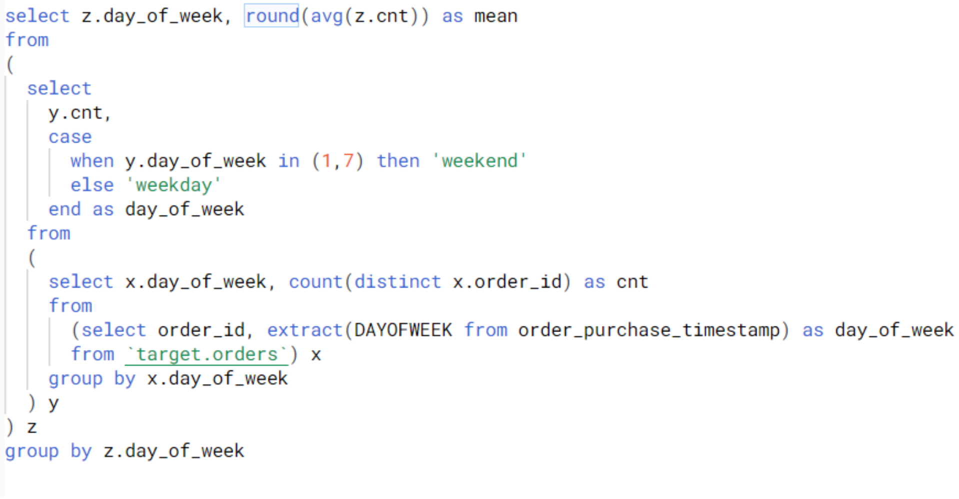


```
select z.time_of_day,sum(z.cnt) as cnt from
(select
  y.hour,
  y.cnt,
  case
    when y.hour>=0 and y.hour<6 then 'dawn'
    when y.hour>=6 and y.hour<12 then 'morning'
    when y.hour>=12 and y.hour<18 then 'afternoon'
    else 'night'
  end as time_of_day
from
  (select x.hour, count(distinct x.order_id) as cnt
  from
    (select order_id, extract(hour from order_purchase_timestamp) as hour
    from `target.orders`) x
  group by x.hour) y) z
group by z.time_of_day
order by cnt
```

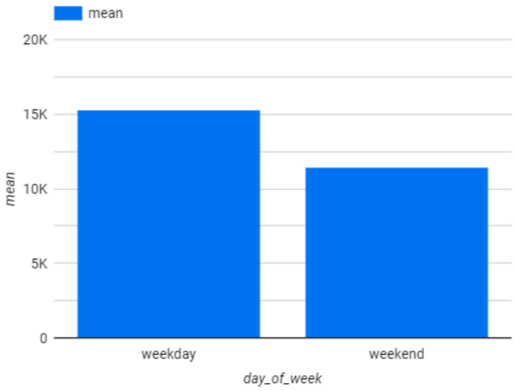


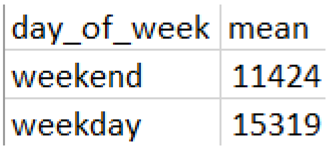


time_of_day	cnt
dawn	4740
morning	22240
night	34100
afternoon	38361



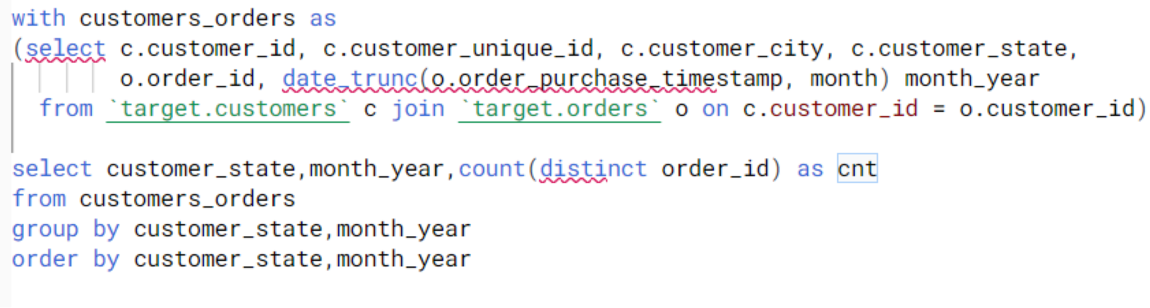
```
select z.day_of_week, round(avg(z.cnt)) as mean
from
(
  select
    y.cnt,
    case
      when y.day_of_week in (1,7) then 'weekend'
      else 'weekday'
    end as day_of_week
  from
  (
    select x.day_of_week, count(distinct x.order_id) as cnt
    from
      (select order_id, extract(DAYOFWEEK from order_purchase_timestamp) as day_of_week
      from `target.orders`) x
    group by x.day_of_week
  ) y
) z
group by z.day_of_week
```



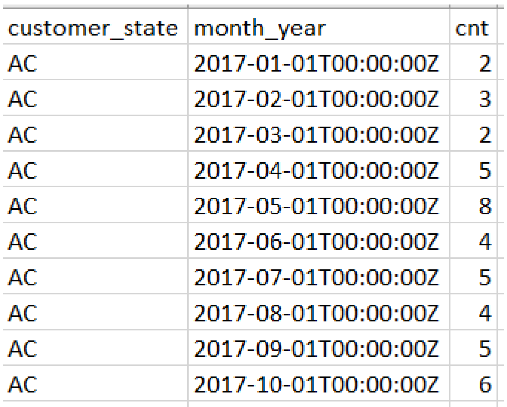


day_of_week	mean
weekend	11424
weekday	15319

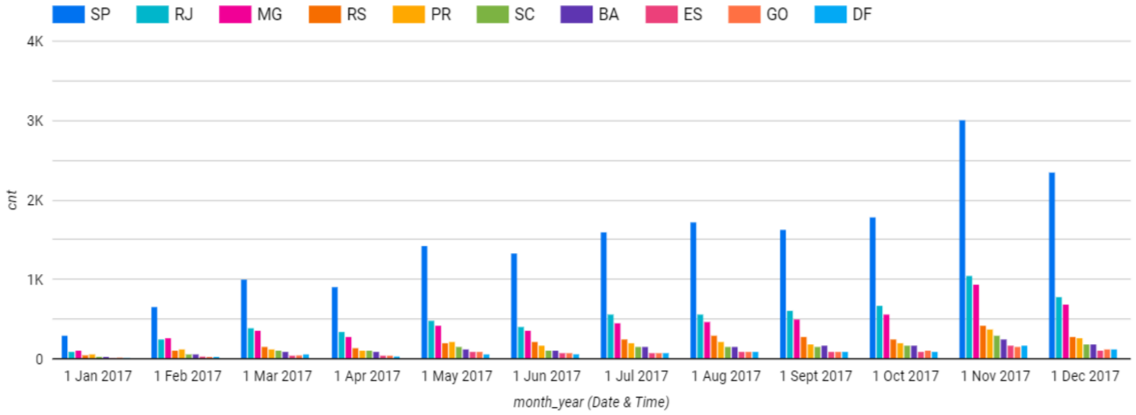
3. Evolution of E-commerce orders in the Brazil region:
    1. Get month on month orders by region, states

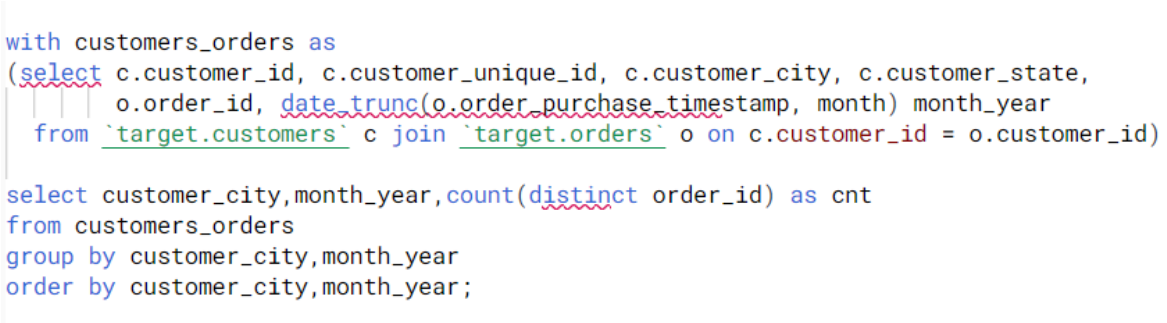


```
with customers_orders as
(select c.customer_id, c.customer_unique_id, c.customer_city, c.customer_state,
        o.order_id, date_trunc(o.order_purchase_timestamp, month) month_year
  from `target.customers` c join `target.orders` o on c.customer_id = o.customer_id)

select customer_state,month_year,count(distinct order_id) as cnt
from customers_orders
group by customer_state,month_year
order by customer_state,month_year
```

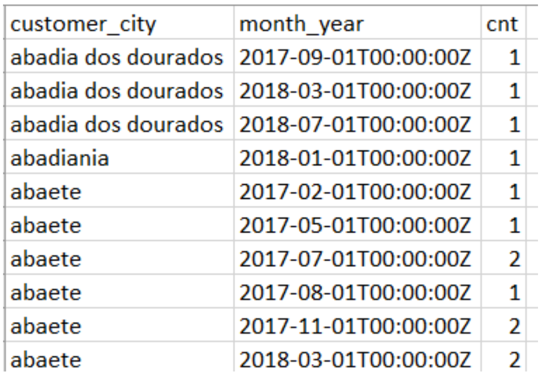


customer_state	month_year	cnt
AC	2017-01-01T00:00:00Z	2
AC	2017-02-01T00:00:00Z	3
AC	2017-03-01T00:00:00Z	2
AC	2017-04-01T00:00:00Z	5
AC	2017-05-01T00:00:00Z	8
AC	2017-06-01T00:00:00Z	4
AC	2017-07-01T00:00:00Z	5
AC	2017-08-01T00:00:00Z	4
AC	2017-09-01T00:00:00Z	5
AC	2017-10-01T00:00:00Z	6

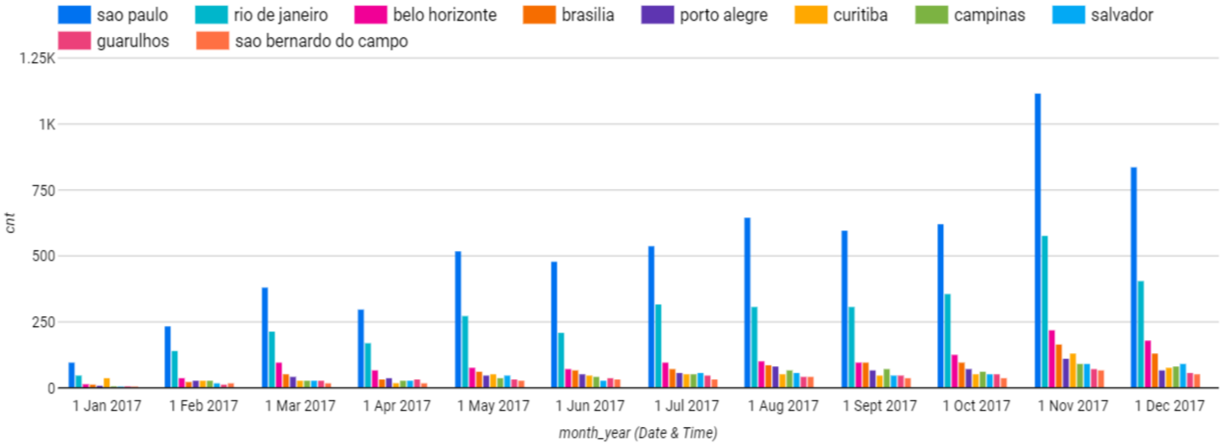




```
with customers_orders as
(select c.customer_id, c.customer_unique_id, c.customer_city, c.customer_state,
        o.order_id, date_trunc(o.order_purchase_timestamp, month) month_year
  from `target.customers` c join `target.orders` o on c.customer_id = o.customer_id)

select customer_city,month_year,count(distinct order_id) as cnt
from customers_orders
group by customer_city,month_year
order by customer_city,month_year;
```

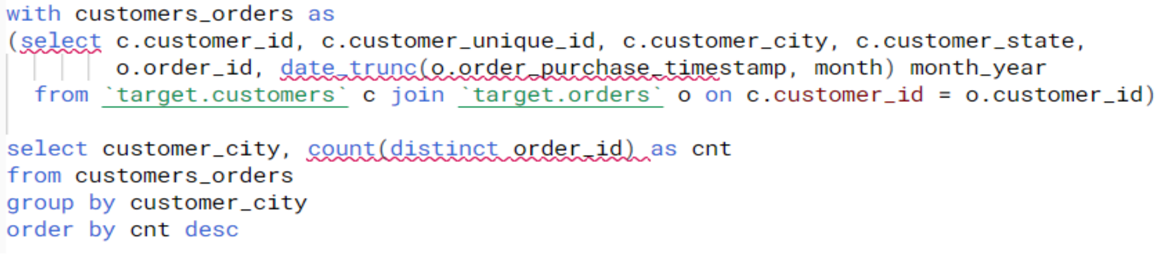


customer_city	month_year	cnt
abadia dos dourados	2017-09-01T00:00:00Z	1
abadia dos dourados	2018-03-01T00:00:00Z	1
abadia dos dourados	2018-07-01T00:00:00Z	1
abadiania	2018-01-01T00:00:00Z	1
abaete	2017-02-01T00:00:00Z	1
abaete	2017-05-01T00:00:00Z	1
abaete	2017-07-01T00:00:00Z	2
abaete	2017-08-01T00:00:00Z	1
abaete	2017-11-01T00:00:00Z	2
abaete	2018-03-01T00:00:00Z	2

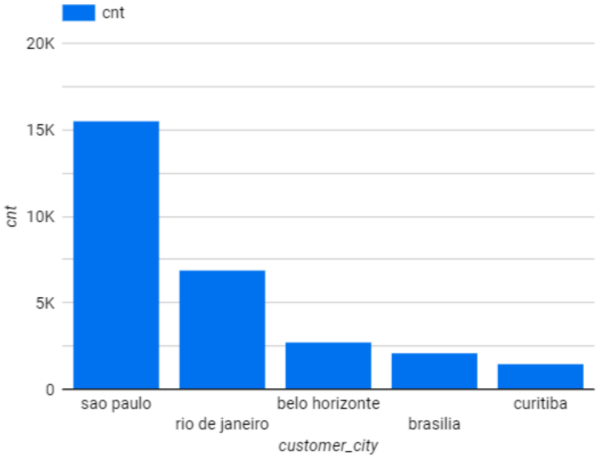


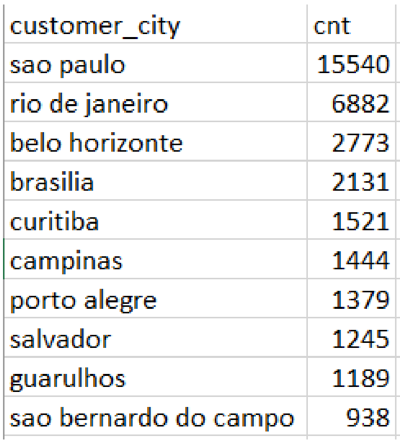
2. How are customers distributed in Brazil

- Sao Paulo (city) and SP(state) have the highest number of customers.

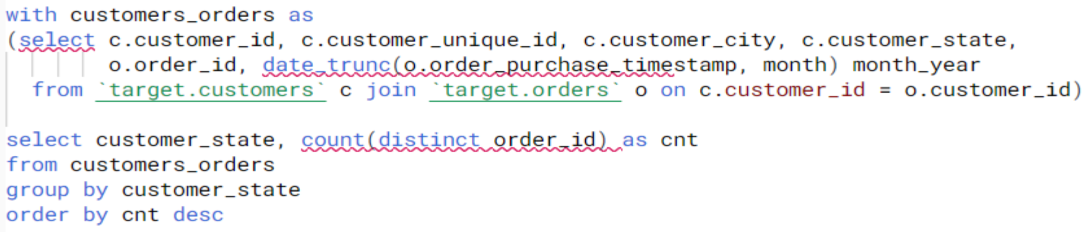


```
with customers_orders as
(select c.customer_id, c.customer_unique_id, c.customer_city, c.customer_state,
        o.order_id, date_trunc(o.order_purchase_timestamp, month) month_year
  from `target.customers` c join `target.orders` o on c.customer_id = o.customer_id)

select customer_city, count(distinct order_id) as cnt
from customers_orders
group by customer_city
order by cnt desc
```

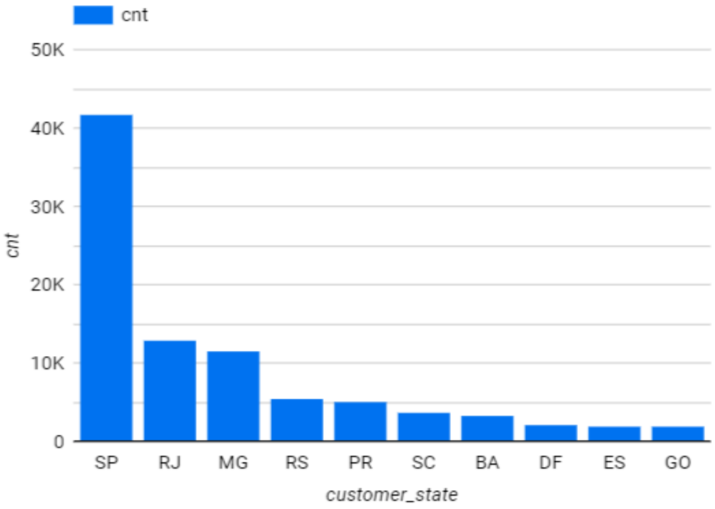


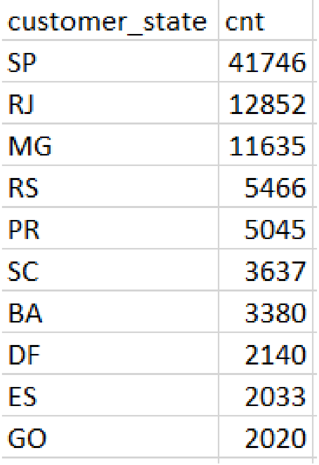


customer_city	cnt
sao paulo	15540
rio de janeiro	6882
belo horizonte	2773
brasilia	2131
curitiba	1521
campinas	1444
porto alegre	1379
salvador	1245
guarulhos	1189
sao bernardo do campo	938



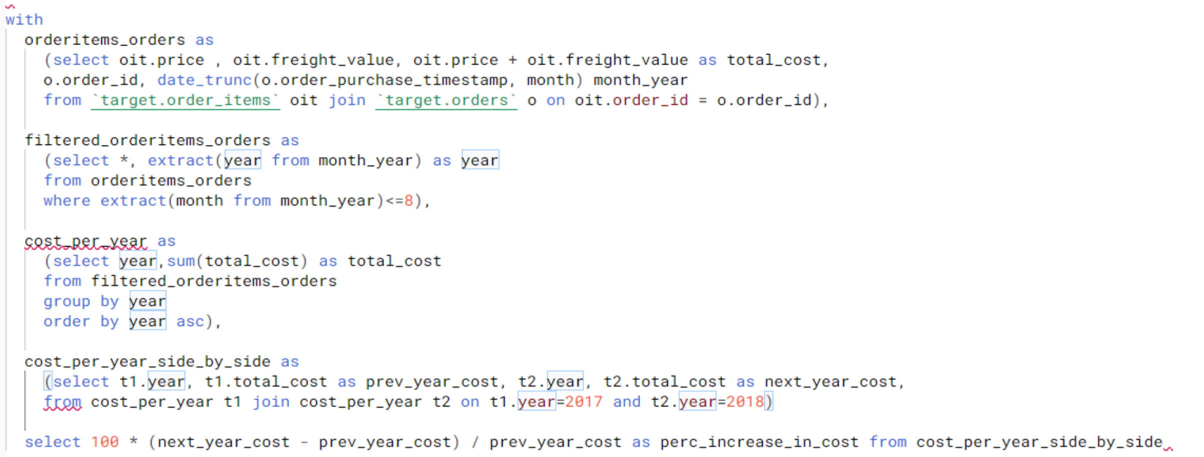
```
with customers_orders as
(select c.customer_id, c.customer_unique_id, c.customer_city, c.customer_state,
        o.order_id, date_trunc(o.order_purchase_timestamp, month) month_year
  from `target.customers` c join `target.orders` o on c.customer_id = o.customer_id)

select customer_state, count(distinct order_id) as cnt
from customers_orders
group by customer_state
order by cnt desc
```



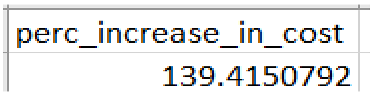


customer_state	cnt
SP	41746
RJ	12852
MG	11635
RS	5466
PR	5045
SC	3637
BA	3380
DF	2140
ES	2033
GO	2020

4. Impact on Economy: Analyse the money movemented 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)

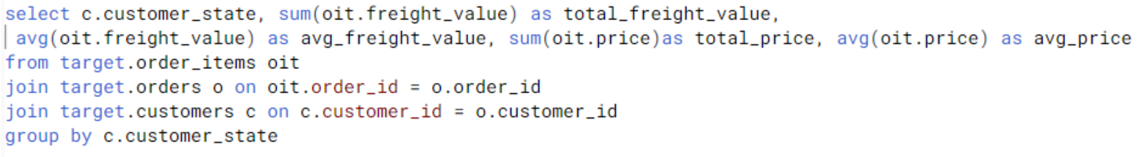


```
with
  orderitems_orders as
    (select oit.price , oit.freight_value, oit.price + oit.freight_value as total_cost,
    o.order_id, date_trunc(o.order_purchase_timestamp, month) month_year
    from `target.order_items` oit join `target.orders` o on oit.order_id = o.order_id),

  filtered_orderitems_orders as
    (select *, extract(year from month_year) as year
    from orderitems_orders
    where extract(month from month_year)<=8),

  cost_per_year as
    (select year,sum(total_cost) as total_cost
    from filtered_orderitems_orders
    group by year
    order by year asc),

  cost_per_year_side_by_side as
    (select t1.year, t1.total_cost as prev_year_cost, t2.year, t2.total_cost as next_year_cost,
    from cost_per_year t1 join cost_per_year t2 on t1.year=2017 and t2.year=2018)

  select 100 * (next_year_cost - prev_year_cost) / prev_year_cost as perc_increase_in_cost from cost_per_year_side_by_side
```

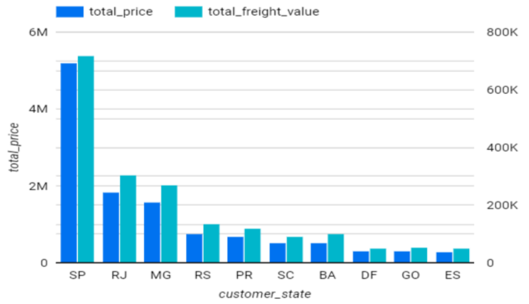


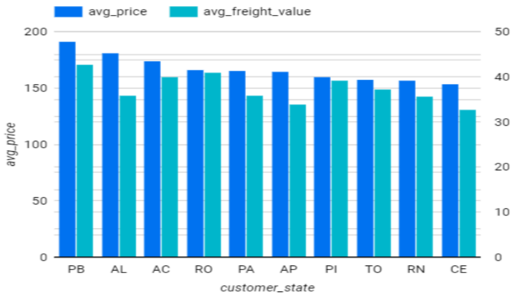
perc_increase_in_cost
139.4150792

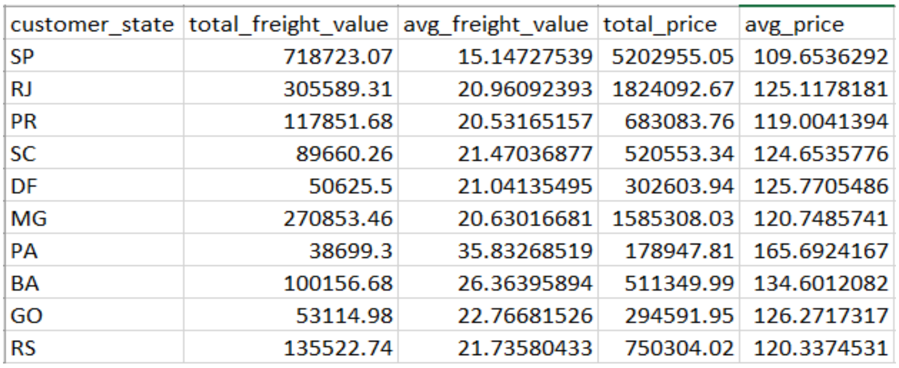
    2. Mean & Sum of price and freight value by customer state



```
select c.customer_state, sum(oit.freight_value) as total_freight_value,
 avg(oit.freight_value) as avg_freight_value, sum(oit.price)as total_price, avg(oit.price) as avg_price
from target.order_items oit
join target.orders o on oit.order_id = o.order_id
join target.customers c on c.customer_id = o.customer_id
group by c.customer_state
```



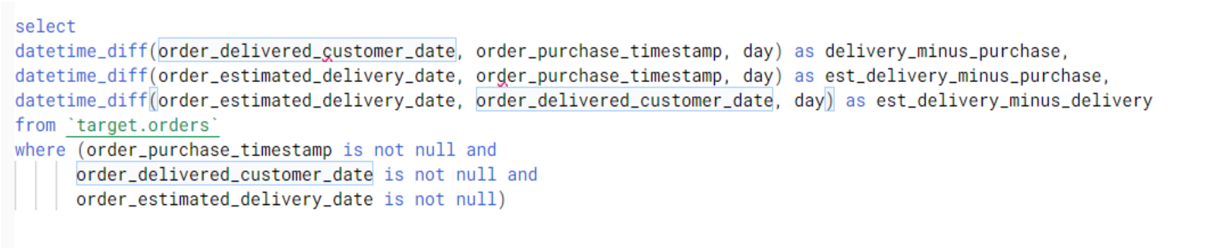




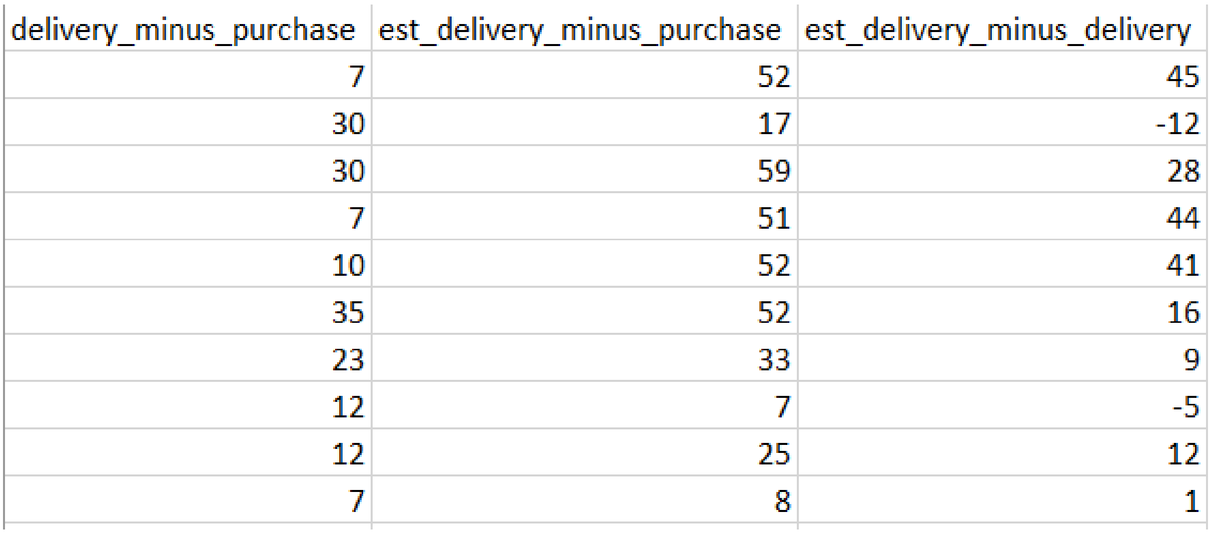
customer_state	total_freight_value	avg_freight_value	total_price	avg_price
SP	718723.07	15.14727539	5202955.05	109.6536292
RJ	305589.31	20.96092393	1824092.67	125.1178181
PR	117851.68	20.53165157	683083.76	119.0041394
SC	89660.26	21.47036877	520553.34	124.6535776
DF	50625.5	21.04135495	302603.94	125.7705486
MG	270853.46	20.63016681	1585308.03	120.7485741
PA	38699.3	35.83268519	178947.81	165.6924167
BA	100156.68	26.36395894	511349.99	134.6012082
GO	53114.98	22.76681526	294591.95	126.2717317
RS	135522.74	21.73580433	750304.02	120.3374531

5. Analysis on sales, freight and delivery time

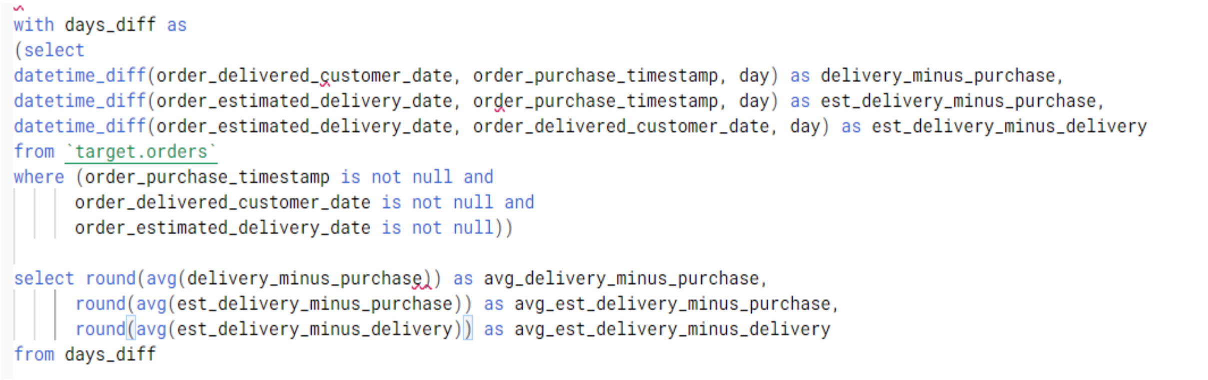
   1. Calculate days between purchasing, delivering and estimated delivery



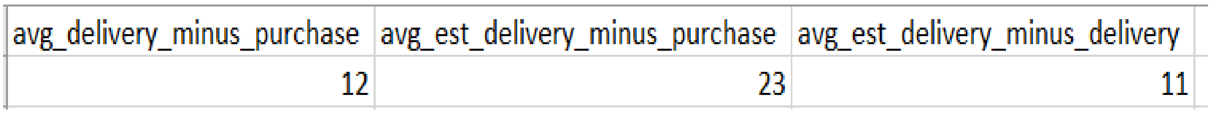
```
select
datetime_diff(order_delivered_customer_date, order_purchase_timestamp, day) as delivery_minus_purchase,
datetime_diff(order_estimated_delivery_date, order_purchase_timestamp, day) as est_delivery_minus_purchase,
datetime_diff(order_estimated_delivery_date, order_delivered_customer_date, day) as est_delivery_minus_delivery
from `target.orders`
where (order_purchase_timestamp is not null and
      order_delivered_customer_date is not null and
      order_estimated_delivery_date is not null)
```



delivery_minus_purchase	est_delivery_minus_purchase	est_delivery_minus_delivery
7	52	45
30	17	-12
30	59	28
7	51	44
10	52	41
35	52	16
23	33	9
12	7	-5
12	25	12
7	8	1

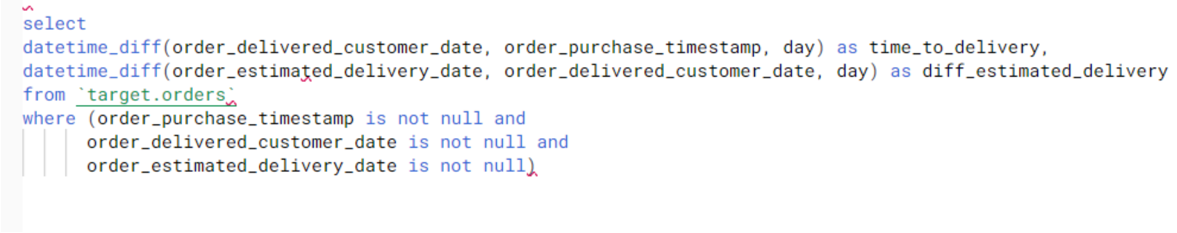


```
with days_diff as
(select
datetime_diff(order_delivered_customer_date, order_purchase_timestamp, day) as delivery_minus_purchase,
datetime_diff(order_estimated_delivery_date, order_purchase_timestamp, day) as est_delivery_minus_purchase,
datetime_diff(order_estimated_delivery_date, order_delivered_customer_date, day) as est_delivery_minus_delivery
from `target.orders`
where (order_purchase_timestamp is not null and
      order_delivered_customer_date is not null and
      order_estimated_delivery_date is not null))

select round(avg(delivery_minus_purchase)) as avg_delivery_minus_purchase,
      round(avg(est_delivery_minus_purchase)) as avg_est_delivery_minus_purchase,
      round(avg(est_delivery_minus_delivery)) as avg_est_delivery_minus_delivery
from days_diff
```



avg_delivery_minus_purchase	avg_est_delivery_minus_purchase	avg_est_delivery_minus_delivery
12	23	11

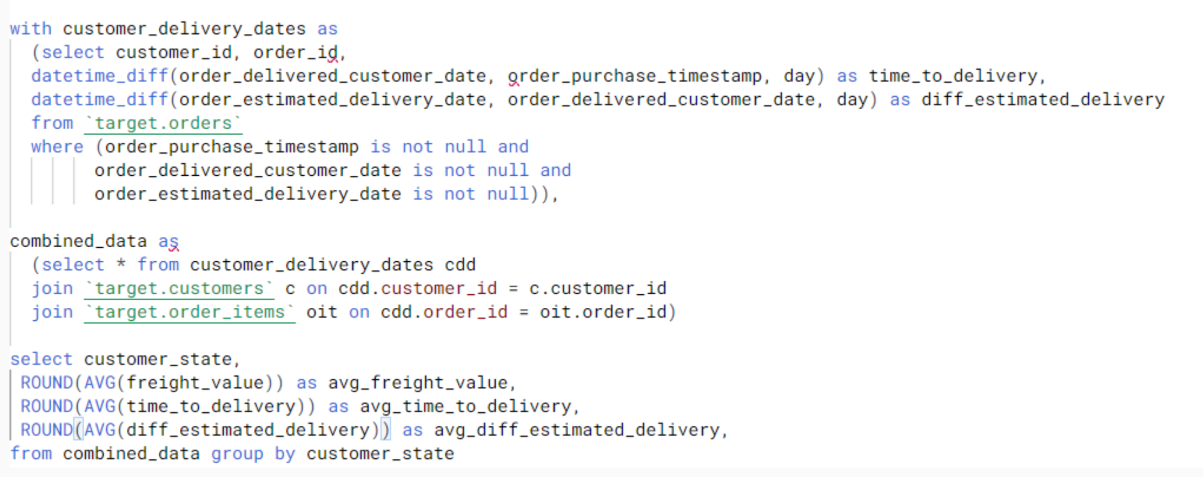
2. Create columns:
1. time_to_delivery = order_purchase_timestamp-order_delivered_customer_date
2. diff_estimated_delivery = order_estimated_delivery_date-order_delivered_customer_date



```
select
datetime_diff(order_delivered_customer_date, order_purchase_timestamp, day) as time_to_delivery,
datetime_diff(order_estimated_delivery_date, order_delivered_customer_date, day) as diff_estimated_delivery
from `target.orders`
where (order_purchase_timestamp is not null and
      order_delivered_customer_date is not null and
      order_estimated_delivery_date is not null)
```

time_to_delivery	diff_estimated_delivery
7	45
30	-12
30	28
7	44
10	41
35	16
23	9
12	-5
12	12
7	1

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

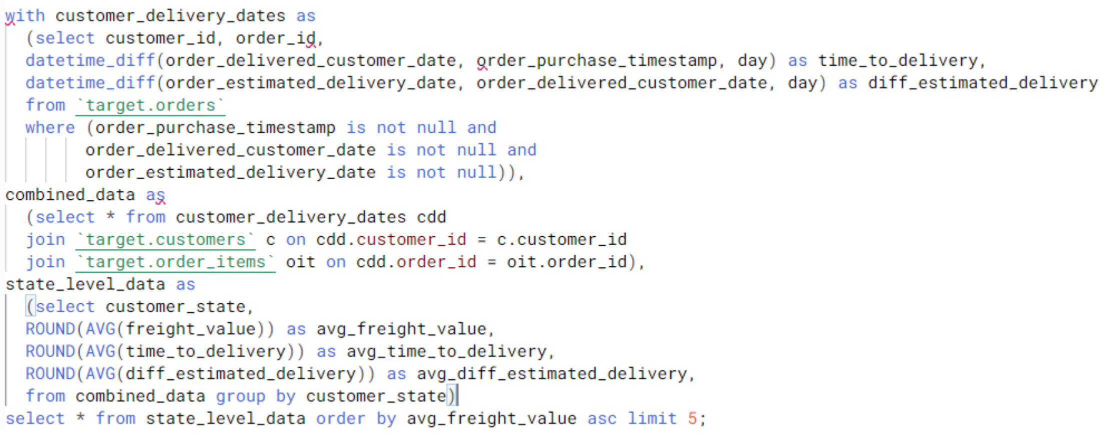


```
with customer_delivery_dates as
  (select customer_id, order_id,
  datetime_diff(order_delivered_customer_date, order_purchase_timestamp, day) as time_to_delivery,
  datetime_diff(order_estimated_delivery_date, order_delivered_customer_date, day) as diff_estimated_delivery
  from `target.orders`
  where (order_purchase_timestamp is not null and
        order_delivered_customer_date is not null and
        order_estimated_delivery_date is not null)),

combined_data as
  (select * from customer_delivery_dates cdd
  join `target.customers` c on cdd.customer_id = c.customer_id
  join `target.order_items` oit on cdd.order_id = oit.order_id)

select customer_state,
 ROUND(AVG(freight_value)) as avg_freight_value,
 ROUND(AVG(time_to_delivery)) as avg_time_to_delivery,
 ROUND(AVG(diff_estimated_delivery)) as avg_diff_estimated_delivery,
from combined_data group by customer_state
```

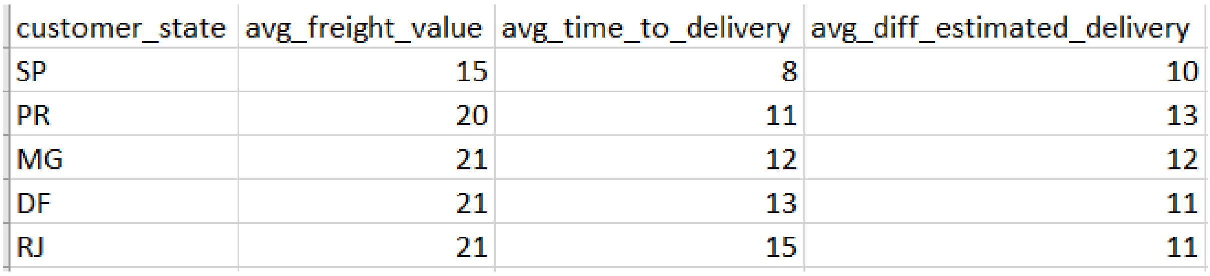
customer_state	avg_freight_value	avg_time_to_delivery	avg_diff_estimated_delivery
RJ	21	15	11
MG	21	12	12
SC	22	15	11
SP	15	8	10
GO	23	15	11
RS	22	15	13
BA	26	19	10
MT	28	18	14
SE	37	21	9
PE	33	18	13

4. Sort the data to get the following:

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

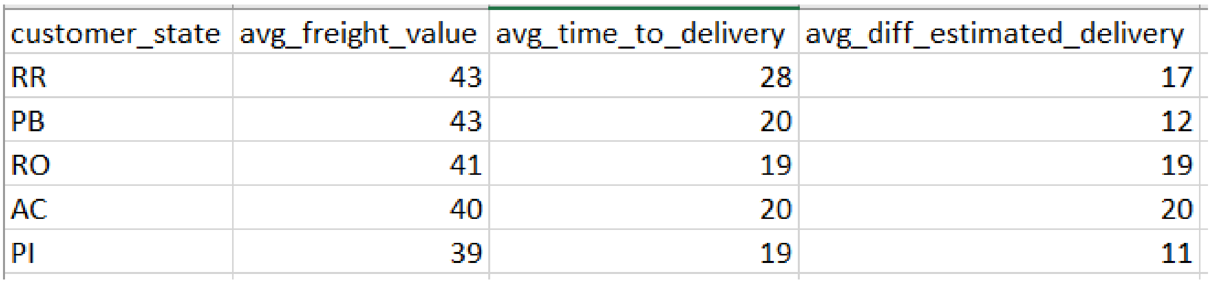


```
with customer_delivery_dates as
  (select customer_id, order_id,
  datetime_diff(order_delivered_customer_date, order_purchase_timestamp, day) as time_to_delivery,
  datetime_diff(order_estimated_delivery_date, order_delivered_customer_date, day) as diff_estimated_delivery
  from `target.orders`
  where (order_purchase_timestamp is not null and
        order_delivered_customer_date is not null and
        order_estimated_delivery_date is not null)),
combined_data as
  (select * from customer_delivery_dates cdd
  join `target.customers` c on cdd.customer_id = c.customer_id
  join `target.order_items` oit on cdd.order_id = oit.order_id),
state_level_data as
  (select customer_state,
  ROUND(AVG(freight_value)) as avg_freight_value,
  ROUND(AVG(time_to_delivery)) as avg_time_to_delivery,
  ROUND(AVG(diff_estimated_delivery)) as avg_diff_estimated_delivery,
  from combined_data group by customer_state)
select * from state_level_data order by avg_freight_value asc limit 5;
```



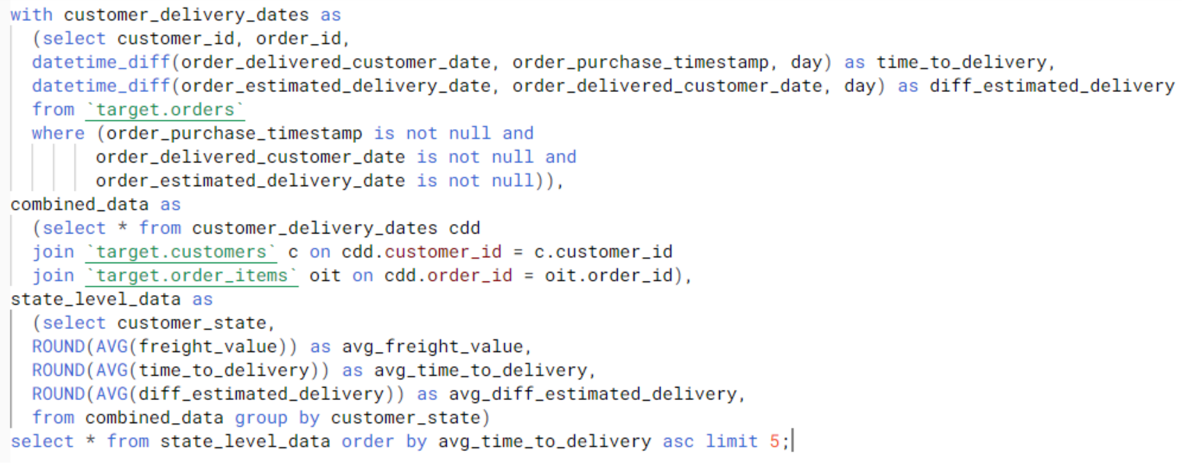
customer_state	avg_freight_value	avg_time_to_delivery	avg_diff_estimated_delivery
SP	15	8	10
PR	20	11	13
MG	21	12	12
DF	21	13	11
RJ	21	15	11

Setting **avg_freight_value by desc** in the above query :

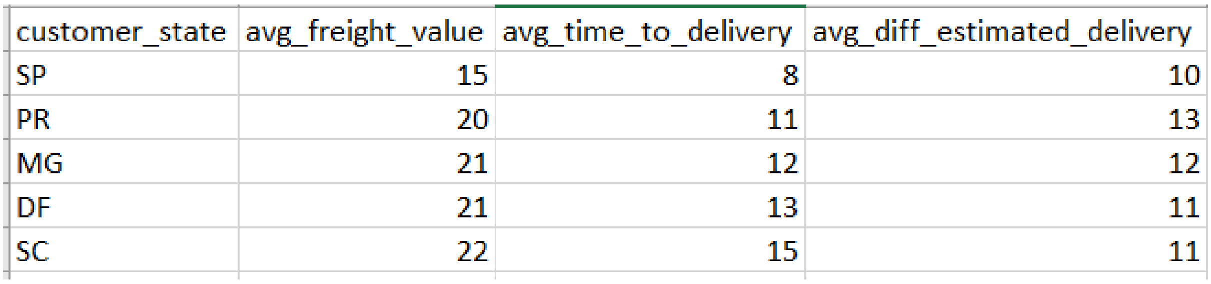


customer_state	avg_freight_value	avg_time_to_delivery	avg_diff_estimated_delivery
RR	43	28	17
PB	43	20	12
RO	41	19	19
AC	40	20	20
PI	39	19	11

2. Top 5 states with highest/lowest average time to delivery

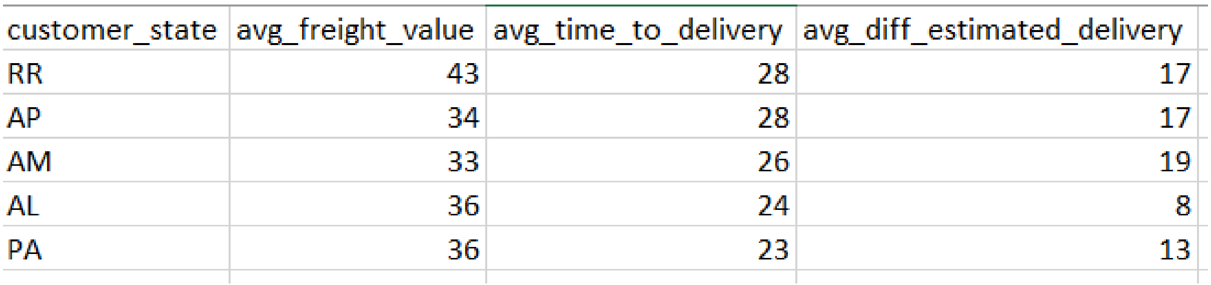


```
with customer_delivery_dates as
  (select customer_id, order_id,
  datetime_diff(order_delivered_customer_date, order_purchase_timestamp, day) as time_to_delivery,
  datetime_diff(order_estimated_delivery_date, order_delivered_customer_date, day) as diff_estimated_delivery
  from `target.orders`
  where (order_purchase_timestamp is not null and
        order_delivered_customer_date is not null and
        order_estimated_delivery_date is not null)),
combined_data as
  (select * from customer_delivery_dates cdd
  join `target.customers` c on cdd.customer_id = c.customer_id
  join `target.order_items` oit on cdd.order_id = oit.order_id),
state_level_data as
  (select customer_state,
  ROUND(AVG(freight_value)) as avg_freight_value,
  ROUND(AVG(time_to_delivery)) as avg_time_to_delivery,
  ROUND(AVG(diff_estimated_delivery)) as avg_diff_estimated_delivery,
  from combined_data group by customer_state)
select * from state_level_data order by avg_time_to_delivery asc limit 5;
```



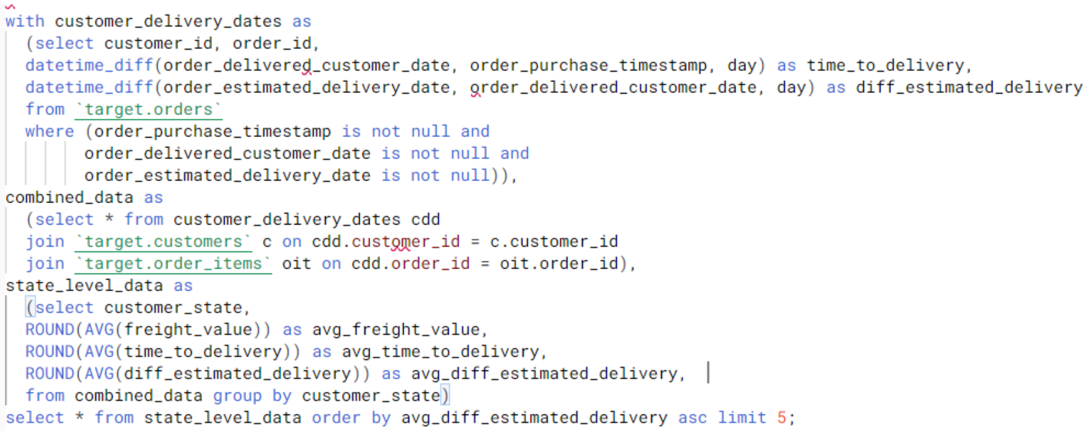
customer_state	avg_freight_value	avg_time_to_delivery	avg_diff_estimated_delivery
SP	15	8	10
PR	20	11	13
MG	21	12	12
DF	21	13	11
SC	22	15	11

Setting **avg_time_to_delivery by desc** in the above query :

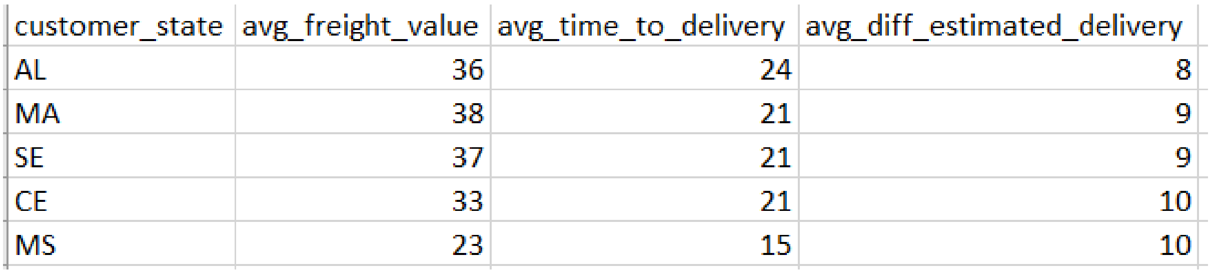


customer_state	avg_freight_value	avg_time_to_delivery	avg_diff_estimated_delivery
RR	43	28	17
AP	34	28	17
AM	33	26	19
AL	36	24	8
PA	36	23	13

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

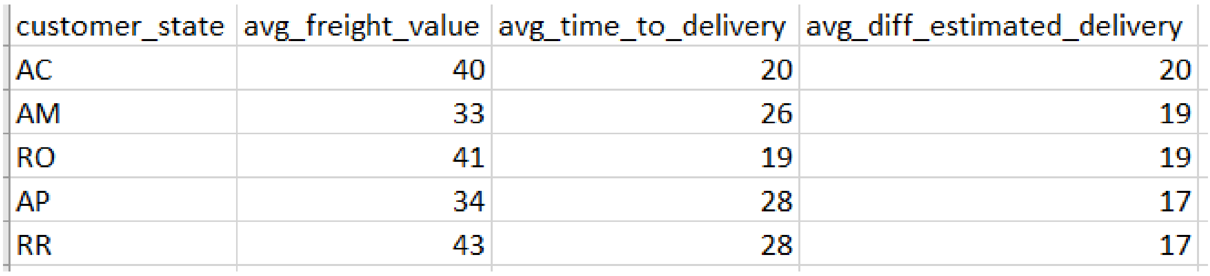


```
with customer_delivery_dates as
  (select customer_id, order_id,
  datetime_diff(order_delivered_customer_date, order_purchase_timestamp, day) as time_to_delivery,
  datetime_diff(order_estimated_delivery_date, order_delivered_customer_date, day) as diff_estimated_delivery
  from `target.orders`
  where (order_purchase_timestamp is not null and
        order_delivered_customer_date is not null and
        order_estimated_delivery_date is not null)),
combined_data as
  (select * from customer_delivery_dates cdd
  join `target.customers` c on cdd.customer_id = c.customer_id
  join `target.order_items` oit on cdd.order_id = oit.order_id),
state_level_data as
  (select customer_state,
  ROUND(AVG(freight_value)) as avg_freight_value,
  ROUND(AVG(time_to_delivery)) as avg_time_to_delivery,
  ROUND(AVG(diff_estimated_delivery)) as avg_diff_estimated_delivery,
  from combined_data group by customer_state)
select * from state_level_data order by avg_diff_estimated_delivery asc limit 5;
```



customer_state	avg_freight_value	avg_time_to_delivery	avg_diff_estimated_delivery
AL	36	24	8
MA	38	21	9
SE	37	21	9
CE	33	21	10
MS	23	15	10

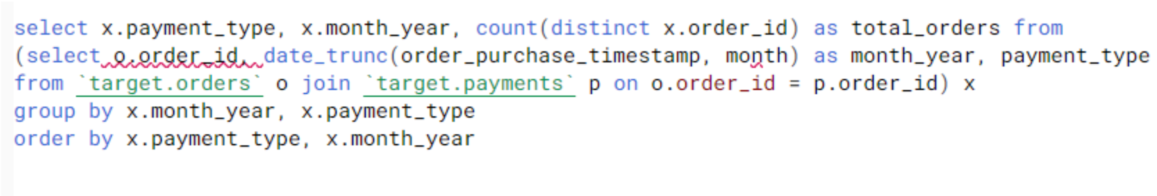
Setting **avg_diff_estimated_delivery by desc** in the above query :



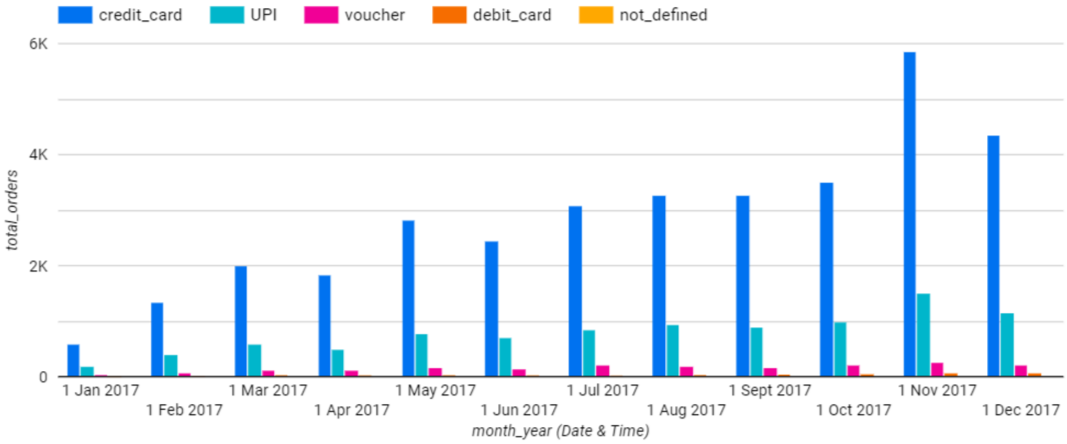
customer_state	avg_freight_value	avg_time_to_delivery	avg_diff_estimated_delivery
AC	40	20	20
AM	33	26	19
RO	41	19	19
AP	34	28	17
RR	43	28	17

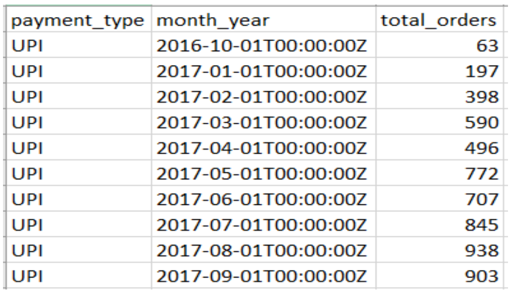
6. Payment type analysis:

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



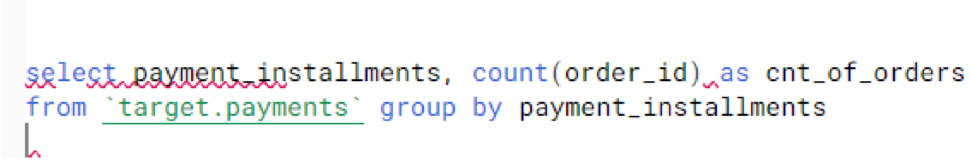
```
select x.payment_type, x.month_year, count(distinct x.order_id) as total_orders from
(select o.order_id, date_trunc(order_purchase_timestamp, month) as month_year, payment_type
from `target.orders` o join `target.payments` p on o.order_id = p.order_id) x
group by x.month_year, x.payment_type
order by x.payment_type, x.month_year
```



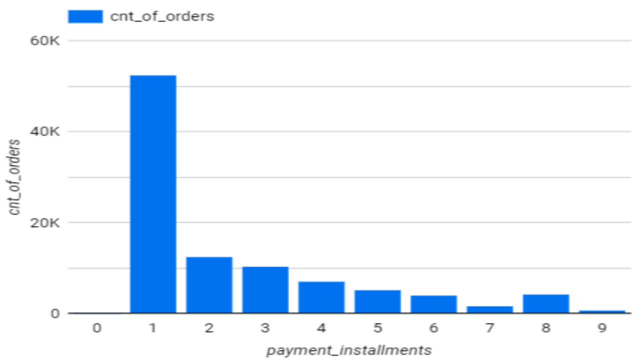


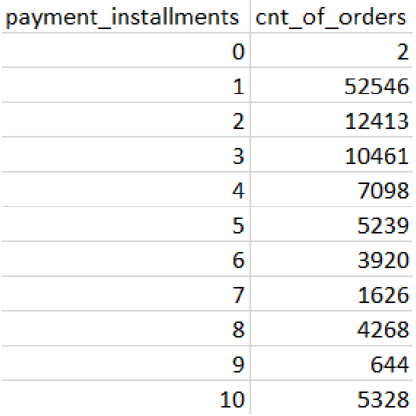
payment_type	month_year	total_orders
UPI	2016-10-01T00:00:00Z	63
UPI	2017-01-01T00:00:00Z	197
UPI	2017-02-01T00:00:00Z	398
UPI	2017-03-01T00:00:00Z	590
UPI	2017-04-01T00:00:00Z	496
UPI	2017-05-01T00:00:00Z	772
UPI	2017-06-01T00:00:00Z	707
UPI	2017-07-01T00:00:00Z	845
UPI	2017-08-01T00:00:00Z	938
UPI	2017-09-01T00:00:00Z	903

2. Distribution of payment instalments and count of orders



```
select payment_installments, count(order_id) as cnt_of_orders
from `target.payments` group by payment_installments
```





payment_installments	cnt_of_orders
0	2
1	52546
2	12413
3	10461
4	7098
5	5239
6	3920
7	1626
8	4268
9	644
10	5328

7. Actionable Insights
    1. The dataset consists of **~1 Lakh customers** who have made orders from **Sep’16 to Oct’18** coming from **27 different states** of Brazil.
    2. Brazil has seen a **steep increase in e-commerce activity** since 2016, showing **strong month on month growth** till 2018.
    3. The e-commerce growth fits **inline with global trends** which was made **possible in the 2010s** due to the widespread availability of **smartphones, internet and social media**.
    4. **Nov’17** saw the **highest volume,** making a significant increase from the previous month (Oct’17). This can be explained by the **festive purchases made by Brazillians for Christmas.**
    5. Brazillians shop the **most on afternoons** (12PM-6PM) and **least in dawn** (12AM-6AM)
    6. Brazillians shop **more on weekdays than on weekends** on average.
    7. The **3 top performing states across all months** in terms of order volume : **SP, RJ, MG**
    8. The **3 top performing cities across all months** in terms of order volume : **Sao Paulo, Rio de Janeiro, Belo Horizonte**
    9. The **% increase in cost of orders** from 2017 to 2018 was a whopping **139%**
    10. State with **highest total order price** is **SP (~5.2M)**
    11. State with **highest average order price** is **PB (~175)**
    12. **Avg time to delivery** across the country is **12 days**
    13. **Avg difference between estimated and delivery date** is **11 days**
    14. **Average freight value :**
        Highest 5 States : RR, PB, RO, AC, PI
        Lowest 5 States : SP, PR, MG, DF, RJ
    15. **Average time to delivery** :
        Highest 5 States : RR, AP, AM, AL, PA
        Lowest 5 States : SP, PR, MG, DF, SC
    16. **Average delivery time compared to estimated date :**
        Highest 5 States : AC, AM, RO, AP, RR
        Lowest 5 States : AL, MA, SE, CE, MS
    17. Brazillians **choice of payments** is dominated by **credit cards** followed by **UPIs** as a distant second.
8. Recommendations
    1. **Festive season** is marked by **increased consumer spending** across the globe. And so is seen in **Brazil (Nov’17)**. I would recommend the e-commerce vendor to come up with **lucrative festival deals, special discounts days, credit card offers** in months leading up to **Christmas (Oct, Nov, Dec)** and back it up with a **strong marketing campaign** especially in states with **lower market penetration** like **RS, PR, SC.**
    2. Since the e-commerce vendor’s order volume is **low on weekends and dawn**,it can provide special offers like **midnight weekends discounts**. They can also **prioritise non-festive months** for this scheme to keep the order volume high throughout the year.

3. Even though **PB has the highest average order price (~175)** i.e. **most premium customers,** the **time to deliver** to this state is **not even in the top 5**. The e-commerce vendor can look into **increasing its distribution channels** into such states with premium customers.
4. The **average difference between estimated and delivery date** is **11 days.** Although this tells that the e-commerce vendor has a **dense and robust distribution channel,** it is **not able to give a good estimate** to the consumer on when to expect the delivery. This can **drive away those customers** who need their **orders delivered urgently** and are unaware of the vendor's optimistic estimates.The vendor should look into this and **calibrate its estimated delivery date by ~11 days** accordingly.
5. **Average time to deliver** in states like **RR and AP** are **28 days!** The vendor must i**ncrease its distribution channel efficiency** in these states if they want to increase market cap. If delivery is an issue due to remote location, the e-commerce vendor can even think of **building new godowns** near these remote states.
6. Credit cards are the primary choice of payment taken by customers. **To capture a bigger market** including those **who do not have a credit card** or **a good CIBIL score to get one**, the e-commerce vendor can offer **special discounts on UPI transactions.**