

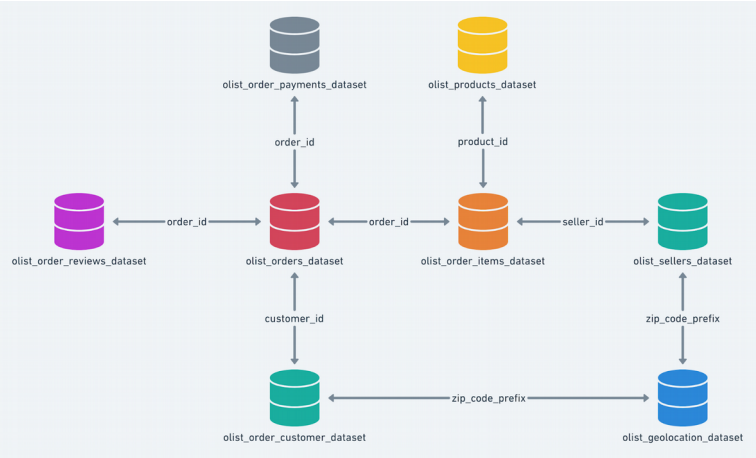
Case Study: SQL

Number of tables: 07

Names of tables:

- 1. customers
- 2. geolocation
- 3. order\_items
- 4. payments
- 5. reviews
- 6. orders
- 7. products
- 8. sellers

Schema: High level overview of relationship between datasets.



Problem statement: Explore the datasets using SQL queries and extract actionable insights. Provide valuable recommendations.

Solution:

Q1: 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
- 2.Time period for which the data is given
- 3.Cities and States covered in the dataset

Overall Tables View:

Below query shows all the details of tables in the Brazil schema and retail-360819 datasets

SELECT \* FROM `retail-360819.Brazil.INFORMATION\_SCHEMA.TABLES` ;

Query results									
JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS					
Row	table_catalog	table_schema	table_name	table_type	is_insertable_into	is_typed	creation_time	base	
1	retail-360819	Brazil	order_items	BASE TABLE	YES	NO	2022-08-28 20:26:02.693000 U...	null	
2	retail-360819	Brazil	sellers	BASE TABLE	YES	NO	2022-08-28 20:28:01.294000 U...	null	
3	retail-360819	Brazil	geolocation	BASE TABLE	YES	NO	2022-08-28 20:23:01.094000 U...	null	
4	retail-360819	Brazil	products	BASE TABLE	YES	NO	2022-08-28 20:30:54.543000 U...	null	

Overall Columns View:  
Below query shows all columns in all tables

SELECT \* FROM `retail-360819.Brazil.INFORMATION\_SCHEMA.COLUMNS`;

Query results

SAVE RESULTSEXPLORE DATA

JOB INFORMATION		RESULTS	JSON		EXECUTION DETAILS			
Row	table_catalog	table_schema	table_name	column_name	ordinal_posi...	is_nullable	data_type	is_generated
1	retail-360819	Brazil	order_items	order_id	1	YES	STRING	NEVER
2	retail-360819	Brazil	order_items	order_item_id	2	YES	INT64	NEVER
3	retail-360819	Brazil	order_items	product_id	3	YES	STRING	NEVER
4	retail-360819	Brazil	order_items	seller_id	4	YES	STRING	NEVER
5	retail-360819	Brazil	order_items	shipping_limit_date	5	YES	TIMESTAMP	NEVER
6	retail-360819	Brazil	order_items	price	6	YES	FLOAT64	NEVER
7	retail-360819	Brazil	order_items	freight_value	7	YES	FLOAT64	NEVER
8	retail-360819	Brazil	sellers	seller_id	1	YES	STRING	NEVER
9	retail-360819	Brazil	sellers	seller_zip_code_prefix	2	YES	INT64	NEVER
10	retail-360819	Brazil	sellers	seller_city	3	YES	STRING	NEVER

customers Table:  
Below query shows columns and data types in customers table

SELECT \* FROM `retail-360819.Brazil.INFORMATION\_SCHEMA.COLUMNS` where table\_name='customers';

Query results

SAVE RESULTSEXPLORE DATA

JOB INFORMATION		RESULTS	JSON		EXECUTION DETAILS			
Row	table_catalog	table_schema	table_name	column_name	ordinal_posi...	is_nullable	data_type	is_generated
1	retail-360819	Brazil	customers	customer_id	1	YES	STRING	NEVER
2	retail-360819	Brazil	customers	customer_unique_id	2	YES	STRING	NEVER
3	retail-360819	Brazil	customers	customer_zip_code_prefix	3	YES	INT64	NEVER
4	retail-360819	Brazil	customers	customer_city	4	YES	STRING	NEVER
5	retail-360819	Brazil	customers	customer_state	5	YES	STRING	NEVER

geolocation Table:  
Below query shows columns and data types in geolocation table

SELECT \* FROM `retail-360819.Brazil.INFORMATION\_SCHEMA.COLUMNS` where table\_name='geolocation';

Query results

SAVE RESULTSEXPLORE DATA

JOB INFORMATION		RESULTS	JSON		EXECUTION DETAILS			
Row	table_catalog	table_schema	table_name	column_name	ordinal_posi...	is_nullable	data_type	is_generated
1	retail-360819	Brazil	geolocation	geolocation_zip_code_prefix	1	YES	INT64	NEVER
2	retail-360819	Brazil	geolocation	geolocation_lat	2	YES	FLOAT64	NEVER
3	retail-360819	Brazil	geolocation	geolocation_lng	3	YES	FLOAT64	NEVER
4	retail-360819	Brazil	geolocation	geolocation_city	4	YES	STRING	NEVER
5	retail-360819	Brazil	geolocation	geolocation_state	5	YES	STRING	NEVER

order\_items Table:  
Below query shows columns and data types in order\_items table

SELECT \* FROM `retail-360819.Brazil.INFORMATION\_SCHEMA.COLUMNS` where table\_name='order\_items';

Query results

SAVE RESULTSEXPLORE DATA

JOB INFORMATION		RESULTS	JSON		EXECUTION DETAILS			
Row	table_catalog	table_schema	table_name	column_name	ordinal_posi...	is_nullable	data_type	is_generated
1	retail-360819	Brazil	order_items	order_id	1	YES	STRING	NEVER
2	retail-360819	Brazil	order_items	order_item_id	2	YES	INT64	NEVER
3	retail-360819	Brazil	order_items	product_id	3	YES	STRING	NEVER
4	retail-360819	Brazil	order_items	seller_id	4	YES	STRING	NEVER
5	retail-360819	Brazil	order_items	shipping_limit_date	5	YES	TIMESTAMP	NEVER
6	retail-360819	Brazil	order_items	price	6	YES	FLOAT64	NEVER
7	retail-360819	Brazil	order_items	freight_value	7	YES	FLOAT64	NEVER

**order\_reviews Table:**  
Below query shows columns and datatypes in order\_reviews table

SELECT \* FROM `retail-360819.Brazil.INFORMATION\_SCHEMA.COLUMNS` where table\_name='order\_reviews';

Query results

SAVE RESULTSEXPLORE DATA↕

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS					
Row	table_catalog		table_schema	table_name	column_name	ordinal_posi_	is_nullable	data_type	is_generated
1	retail-360819		Brazil	order_reviews	review_id	1	YES	STRING	NEVER
2	retail-360819		Brazil	order_reviews	order_id	2	YES	STRING	NEVER
3	retail-360819		Brazil	order_reviews	review_score	3	YES	INT64	NEVER
4	retail-360819		Brazil	order_reviews	review_comment_title	4	YES	STRING	NEVER
5	retail-360819		Brazil	order_reviews	review_creation_date	5	YES	TIMESTAMP	NEVER
6	retail-360819		Brazil	order_reviews	review_answer_timestamp	6	YES	TIMESTAMP	NEVER

**orders Table:**  
Below query shows columns and data types in orders table

SELECT \* FROM `retail-360819.Brazil.INFORMATION\_SCHEMA.COLUMNS` where table\_name='orders';

Query results

SAVE RESULTS

EXPLORE DATA

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS					
Row	table_catalog		table_schema	table_name	column_name	ordinal_posi_	is_nullable	data_type	is_generated
1	retail-360819		Brazil	orders	order_id	1	YES	STRING	NEVER
2	retail-360819		Brazil	orders	customer_id	2	YES	STRING	NEVER
3	retail-360819		Brazil	orders	order_status	3	YES	STRING	NEVER
4	retail-360819		Brazil	orders	order_purchase_timestamp	4	YES	TIMESTAMP	NEVER
5	retail-360819		Brazil	orders	order_approved_at	5	YES	TIMESTAMP	NEVER
6	retail-360819		Brazil	orders	order_delivered_carrier_date	6	YES	TIMESTAMP	NEVER
7	retail-360819		Brazil	orders	order_delivered_customer_date	7	YES	TIMESTAMP	NEVER
8	retail-360819		Brazil	orders	order_estimated_delivery_date	8	YES	TIMESTAMP	NEVER

**payments Table:**  
Below query shows columns and data types in payments table

SELECT \* FROM `retail-360819.Brazil.INFORMATION\_SCHEMA.COLUMNS` where table\_name='payments';

Query results

SAVE RESULTS

EXPLORE DATA

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS					
Row	table_catalog		table_schema	table_name	column_name	ordinal_posi_	is_nullable	data_type	is_generated
1	retail-360819		Brazil	payments	order_id	1	YES	STRING	NEVER
2	retail-360819		Brazil	payments	payment_sequential	2	YES	INT64	NEVER
3	retail-360819		Brazil	payments	payment_type	3	YES	STRING	NEVER
4	retail-360819		Brazil	payments	payment_installments	4	YES	INT64	NEVER
5	retail-360819		Brazil	payments	payment_value	5	YES	FLOAT64	NEVER

**products Table:**  
Below query shows columns and data types in products table

SELECT \* FROM `retail-360819.Brazil.INFORMATION\_SCHEMA.COLUMNS` where table\_name='products';

Query results

SAVE RESULTSEXPLORE DATA↕

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS					
Row	table_catalog		table_schema	table_name	column_name	ordinal_posi...	is_nullable	data_type	is_generated
1	retail-360819		Brazil	products	product_id	1	YES	STRING	NEVER
2	retail-360819		Brazil	products	product_category	2	YES	STRING	NEVER
3	retail-360819		Brazil	products	product_name_length	3	YES	INT64	NEVER
4	retail-360819		Brazil	products	product_description_length	4	YES	INT64	NEVER
5	retail-360819		Brazil	products	product_photos_qty	5	YES	INT64	NEVER
6	retail-360819		Brazil	products	product_weight_g	6	YES	INT64	NEVER
7	retail-360819		Brazil	products	product_length_cm	7	YES	INT64	NEVER
8	retail-360819		Brazil	products	product_height_cm	8	YES	INT64	NEVER
9	retail-360819		Brazil	products	product_width_cm	9	YES	INT64	NEVER

**sellers Table:**  
**Below query shows columns and data types in sellers table**

`SELECT * FROM `retail-360819.Brazil.INFORMATION_SCHEMA.COLUMNS` where table_name='sellers';`

Query results									
JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS					
Row	table_catalog		table_schema	table_name	column_name	ordinal_posi...	is_nullable	data_type	is_generated
1	retail-360819		Brazil	sellers	seller_id	1	YES	STRING	NEVER
2	retail-360819		Brazil	sellers	seller_zip_code_prefix	2	YES	INT64	NEVER
3	retail-360819		Brazil	sellers	seller_city	3	YES	STRING	NEVER
4	retail-360819		Brazil	sellers	seller_state	4	YES	STRING	NEVER

**Find out time period for which data is given:**  
**Below query shows start and end time of data given based on order status**

`SELECT`  
`order_status,`  
`MIN DATE (order_purchase_timestamp)) AS time_period_start,`  
`MAX DATE (order_purchase_timestamp)) AS time_period_end`  
`FROM`  
``retail-360819.Brazil.orders``  
`GROUP BY`  
`order_status;`

Query results									
JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS					
Row	order_status		time_period...	time_period...					
1	created		2017-11-06	2018-02-09					
2	shipped		2016-09-04	2018-09-03					
3	approved		2017-02-06	2017-04-25					
4	canceled		2016-09-05	2018-10-17					
5	invoiced		2016-10-04	2018-08-14					
6	delivered		2016-09-15	2018-08-29					
7	processing		2016-10-05	2018-07-23					
8	unavailable		2016-10-05	2018-08-21					

**Find out cities and states covered in the datasets:**  
**Below query shows customer cities and states covered in the datasets:**

`SELECT distinct customer_city, customer_state FROM `retail-360819.Brazil.customers`;`

Query results									
JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS					
Row	customer_city		customer_state						
1	acu		RN						
2	ico		CE						
3	ipe		RS						
4	ipu		CE						
5	ita		SC						
6	itu		SP						
7	jau		SP						
8	luz		MG						
9	poa		SP						
10	uba		MG						
11	una		BA						

**Top 5 states with highest number of cities:**  
**Below query shows top 5 states with highest number of cities**

```
SELECT customer_state
,count(distinct customer_city)
FROM `retail-360819.Brazil.customers`
group by customer_state
order by count(distinct customer_city) desc
limit 5;
```

Query results

SAVE RESULTSEXPLORE DATA

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	customer_state	fo_		
1	MG		745	
2	SP		629	
3	RS		379	
4	PR		364	
5	BA		353	

**Find out cities and states in geolocation table:**  
**Below query shows cities and states in geolocation table**

```
SELECT distinct geolocation_city, geolocation_state
FROM `retail-360819.Brazil.geolocation`
limit 100;
```

Query results

SAVE RESULTSEXPLORE DATA

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	geolocation_city	geolocation_state		
1	aracaju	SE		
2	riachuelo	SE		
3	nossa senhora do socorro	SE		
4	barra dos coqueiros	SE		
5	itaporanga d'ajuda	SE		
6	sao cristovao	SE		
7	são cristóvão	SE		

**Find out top 5 geolocation states with highest number of cities:**  
**Below query shows top 5 states with highest number of cities**

```
SELECT geolocation_state
,count(distinct geolocation_city)
FROM `retail-360819.Brazil.geolocation`
group by geolocation_state
order by count(geolocation_city) desc
limit 5;
```

Query results

SAVE RESULTSEXPLORE DATA

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	geolocation_state	fo_		
1	SP		1048	
2	MG		1426	
3	RJ		245	
4	RS		691	
5	PR		651	

**Find out cites and states of sellers from sellers table:  
Below query shows cities and states from sellers table**

```
select distinct seller_city, seller_state from `retail-360819.Brazil.sellers` limit 100;
```

Query results SAVE RESULTS EXPLORE DATA

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	seller_city	seller_state		
1	rio branco	AC		
2	manaus	AM		
3	bahia	BA		
4	ipira	BA		
5	irece	BA		
6	ilheus	BA		

**Find out top 5 seller states with highest number seller cities:  
Below query shows top 5 seller states with highest number of seller cities**

```
select seller_state
,count(distinct seller_city)
from `retail-360819.Brazil.sellers`
group by seller_state
order by count(distinct seller_city) desc
limit 5;
```

Query results SAVE RESULTS EXPLORE DATA

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	seller_state	fo_		
1	SP	261		
2	MG	82		
3	PR	67		
4	SC	65		
5	RS	51		

**Q2: 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?  
2.What time do Brazilian customers tend to buy (Dawn, Morning, Afternoon or Night)?

**Find out if there is growing trend on e-commerce:  
Below query shows year wise count of orders and freight value for orders with delivered status**

```
select T.year,sum(T.no_of_orders) as no_of_orders,sum(T.freight_value) as freight_value from (
select distinct
extract(YEAR FROM o.order_purchase_timestamp) as year
,extract(MONTH FROM o.order_purchase_timestamp) as month
,extract(QUARTER FROM o.order_purchase_timestamp) as quarter
,count(o.order_id) over (partition by date(o.order_purchase_timestamp)) as no_of_orders
,round(sum(oi.freight_value) over (partition by date(o.order_purchase_timestamp))) as freight_value
from `retail-360819.Brazil.orders` o ,`retail-360819.Brazil.order_items` oi
where o.order_status='delivered'
and o.order_id = oi.order_id ) T
group by T.year order by T.year
```

Query results SAVE RESULTS EXPLORE DATA

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	year	no_of_orders	freight_value	
1	2016	317	6183.0	
2	2017	49556	958633.0	
3	2018	60324	1233457.0	

Inference:

- Increase in number of orders from 2017 to 2018: 21.72%
- Increase in freight value from 2017 to 2018: 28.66%

#### Seasonality Check:

Below query shows the peak months based on orders count

```
select T.year,T.month,sum(T.no_of_orders) as no_of_orders,sum(T.freight_value) as freight_value from (
select distinct
extract(YEAR FROM o.order_purchase_timestamp) as year
,extract(MONTH FROM o.order_purchase_timestamp) as month
,extract(QUARTER FROM o.order_purchase_timestamp) as quarter
,count(o.order_id) over (partition by date(o.order_purchase_timestamp)) as no_of_orders
,round(sum(oi.freight_value) over (partition by date(o.order_purchase_timestamp))) as freight_value
from `retail-360819.Brazil.orders` o ,`retail-360819.Brazil.order_items` oi
where o.order_status='delivered'
and o.order_id = oi.order_id ) T
where T.year = 2017
group by T.year,T.month order by no_of_orders desc
```

Query results						SAVE RESULTS	EXPLORE DATA	
JOB INFORMATION		RESULTS		JSON	EXECUTION DETAILS			
Row	year	month	no_of_orders	freight_value				
1	2017	11	8475	165600.0				
2	2017	12	6187	117047.0				
3	2017	10	5214	102867.0				
4	2017	8	4797	91130.0				
5	2017	9	4737	93678.0				

#### Inference:

- Peaks at specific months: 10, 11, 12 (It's holiday season, especially, Thanksgiving, CyberMonday, Christmas, Ester)
- 40.10% of total orders during 2017 were delivered during the month 10,11,12
- 40.21% of total freight\_value/sales was made during the month 10, 11 and 12

### Find out at what times do Brazilians customers tend to buy?

```
select T2.year
,T2.day_part
,sum(T2.no_of_orders) as no_of_orders
,sum(T2.freight_value) as freight_value
from
(select T1.year
,T1.hour
,CASE
when T1.hour >= 3 and T1.hour <= 4 then 'Dawn'
when T1.hour >= 5 and T1.hour <= 12 then 'Morning'
when T1.hour >= 13 and T1.hour <= 17 then 'Afternoon'
when T1.hour >= 18 and T1.hour <= 21 then 'Evening'
else 'Night'
END as day_part
,sum(T1.no_of_orders) as no_of_orders
,sum(T1.freight_value) as freight_value
from
(select distinct
extract(YEAR FROM o.order_purchase_timestamp) as year
,extract(MONTH FROM o.order_purchase_timestamp) as month
,extract(QUARTER FROM o.order_purchase_timestamp) as quarter
,extract(HOUR FROM o.order_purchase_timestamp) as hour
,count(o.order_id) over (partition by date(o.order_purchase_timestamp)) as no_of_orders
,round(sum(oi.freight_value) over (partition by date(o.order_purchase_timestamp))) as freight_value
from `retail-360819.Brazil.orders` o , `retail-360819.Brazil.order_items` oi
where o.order_status='delivered'
and o.order_id = oi.order_id) T1
group by T1.year,T1.hour) T2
group by T2.year, T2.day_part
order by sum(T2.no_of_orders) desc;
```

Query results						<a href="#">SAVE RESULTS</a>	<a href="#">EXPLORE DATA</a>	
JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS				
Row	year	day_part		no_of_orders	freight_value			
1	2018	Morning		414953	8482067.0			
2	2017	Morning		311820	6033056.0			
3	2018	Afternoon		301042	6155558.0			
4	2018	Night		272368	5566225.0			
5	2017	Afternoon		245894	4754806.0			
6	2018	Evening		241232	4932960.0			

#### Inference:

- Brazilian customers tend to buy during morning time.

### Q3: Evolution of E-commerce orders in the Brazil region:

- 1.Get month on month orders by region, states
- 2.How are customers distributed in Brazil



Find out month on month orders by region, states  
Below query shows the month on month orders %growth by state. Replace year and state as per the requirement

```
select
distinct T1.customer_state
,T1.year
,T1.month
,round(100*(T1.count - lag(T1.count, 1) over (order by T1.month)) / lag(T1.count, 1) over (order by T1.month),2) as percent_growth
from
(SELECT
c.customer_state
,extract(YEAR from o.order_purchase_timestamp) as year
,extract(MONTH from o.order_purchase_timestamp) as month
,count(order_id) as count
FROM
`retail-360819.Brazil.customers` c
,`retail-360819.Brazil.orders` o
where
c.customer_id = o.customer_id
and o.order_status = 'delivered'
and extract(YEAR from o.order_purchase_timestamp) = 2017
--and extract(YEAR from o.order_purchase_timestamp) = 2018
and c.customer_state = 'AC'
--other possible states (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)
group by
c.customer_state
,extract(YEAR from o.order_purchase_timestamp)
,extract(MONTH from o.order_purchase_timestamp)) T1
order by T1.customer_state, T1.year, T1.month
limit 100;
```

Query results

SAVE RESULTS

EXPLORE DATA

JOB INFORMATION		RESULTS	JSON		EXECUTION DETAILS	
Row	customer_state	year	month	percent_gro...		
1	AC	2017	1	null		
2	AC	2017	2	50.0		
3	AC	2017	3	-33.33		
4	AC	2017	4	150.0		
5	AC	2017	5	60.0		
6	AC	2017	6	-50.0		

**Find out how are customers distributed in Brazil:**

**Below query shows how customers are distributed in Brazil as per state, city and zipcode**

```
select c.customer_state
,c.customer_city
,c.customer_zip_code_prefix
,count(c.customer_unique_id) as total_customers
from `retail-360819.Brazil.customers` c
group by c.customer_state, c.customer_city, c.customer_zip_code_prefix
order by total_customers desc
limit 100;
```

Query results					SAVE RESULTS	EXPLORE DATA	
JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS			
Row	customer_state	customer_city	customer_zi...	total_custo...			
1	RJ	rio de janeiro	22790	142			
2	RJ	niteroi	24220	124			
3	RJ	rio de janeiro	22793	121			
4	RJ	niteroi	24230	117			
5	RJ	rio de janeiro	22775	110			

**Q4: Impact on Economy: Analyze 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)**

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

**Find out % Increase in cost of orders from 2017 to 2018 (Jan to Aug only):**

**Below query shows the percentage increase in cost of orders from 2017 to 2018 (span January to August)**

```
select T1.year
,round(100*(T1.total_orders - lag(T1.total_orders, 1) over (order by T1.year)) / lag(T1.total_orders, 1) over (order by T1.year),2) as
perc_order_Increase
,round(100*(T1.total_freight_value - lag(T1.total_freight_value, 1) over (order by T1.year)) / lag(T1.total_freight_value, 1) over (order by
T1.year),2) as perc_freight_Increase
from
(SELECT extract(year from o.order_purchase_timestamp) as year
,count(oi.order_item_id) as total_orders
,round(sum(oi.freight_value)) as total_freight_value
FROM `retail-360819.Brazil.orders` o, `retail-360819.Brazil.order_items` oi
where o.order_id = oi.order_id
and o.order_status = 'delivered'
and extract(year from o.order_purchase_timestamp) in (2017, 2018)
and extract(month from o.order_purchase_timestamp) in (1,2,3,4,5,6,7,8)
group by extract(year from o.order_purchase_timestamp)) T1
order by T1.year;
```

Query results					SAVE RESULTS	EXPLORE DATA	
JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS			
Row	year	perc_order_Increase	perc_freight_Increase				
1	2017	null	null				
2	2018	141.85	157.27				

Inference:

- %Increase on cost of orders = 157.26%
- %Increase on number of orders = 141.81%

**Find out what is the mean and sum of price and freight value by customer state:**  
**Below query shows the mean and sum of price and freight value by the customer state**

```
SELECT c.customer_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
`retail-360819.Brazil.orders` o
,`retail-360819.Brazil.order_items` oi
,`retail-360819.Brazil.customers` c
where o.order_id = oi.order_id
and o.customer_id = c.customer_id
and o.order_status = 'delivered'
and extract(year from o.order_purchase_timestamp) in (2017, 2018)
and extract(month from o.order_purchase_timestamp) in (1,2,3,4,5,6,7,8)
group by c.customer_state
order by c.customer_state
LIMIT 1000;
```

Query results

SAVE RESULTS

EXPLORE DATA

JOB INFORMATION

RESULTS

JSON

EXECUTION DETAILS

Row	customer_state	mean_price	sum_price	mean_freight_value	sum_freight_value
1	AC	167.29	11710.16	37.39	2617.4
2	AL	185.4	60995.85	35.08	11542.45
3	AM	128.78	17256.97	33.36	4470.87
4	AP	158.53	10304.58	33.56	2181.54
5	BA	138.45	389741.63	27.39	77115.15

#### Q5: Analysis on sales, freight and delivery time

1.Calculate days between purchasing, delivering and estimated delivery

2.Create columns:

•time\_to\_delivery = order\_purchase\_timestamp-order\_delivered\_customer\_date

•diff\_estimated\_delivery = order\_estimated\_delivery\_date-order\_delivered\_customer\_date

3.Group data by state, take mean of freight\_value, time\_to\_delivery, diff\_estimated\_delivery

4.Sort the data to get the following:

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

•Top 5 states with highest/lowest average time to delivery

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

#### Calculate days between purchasing, delivering and estimated delivery

```
SELECT order_purchase_timestamp
,order_delivered_customer_date
,order_estimated_delivery_date
,timestamp_diff(order_delivered_customer_date,order_purchase_timestamp,day) as no_of_days_took_to_deliver
,timestamp_diff(order_delivered_customer_date,order_estimated_delivery_date,day) as no_of_days_took_to_deliver_from_estimated_delivery
FROM
`retail-360819.Brazil.orders`
where order_status = 'delivered'
LIMIT 100;
```

Query results

SAVE RESULTS

EXPLORE DATA



JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS		
Row	order_purchase_timestamp	order_delivered_customer_date	order_estimated_delivery_date	no_of_days_took_to_deliver	no_of_days_took_to_deliver_from_estimated_delivery	
1	2017-04-14 22:06:32 UTC	2017-05-08 11:10:26 UTC	2017-05-18 00:00:00 UTC	23	-9	
2	2017-05-10 14:03:27 UTC	2017-05-23 13:12:27 UTC	2017-05-18 00:00:00 UTC	12	5	
3	2017-04-22 15:50:30 UTC	2017-05-05 13:27:50 UTC	2017-05-18 00:00:00 UTC	12	-12	
4	2017-05-09 17:42:45 UTC	2017-05-16 23:22:20 UTC	2017-05-18 00:00:00 UTC	7	-1	
5	2017-04-26 01:01:39 UTC	2017-05-08 08:54:36 UTC	2017-05-18 00:00:00 UTC	12	-9	

### Create columns time\_to\_deliver and diff\_estimated\_delivery

```
SELECT order_purchase_timestamp
,order_delivered_customer_date
,order_estimated_delivery_date
,timestamp_diff(order_delivered_customer_date,order_purchase_timestamp,day) as time_to_deliver
,timestamp_diff(order_estimated_delivery_date,order_delivered_customer_date,day) as diff_estimated_delivery
FROM
`retail-360819.Brazil.orders`
where order_status = 'delivered'
LIMIT 100;
```

Query results

SAVE RESULTS

EXPLORE DATA



JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS		
Row	order_purchase_timestamp	order_delivered_customer_date	order_estimated_delivery_date	time_to_deliver	diff_estimated_delivery	
1	2017-04-14 22:06:32 UTC	2017-05-08 11:10:26 UTC	2017-05-18 00:00:00 UTC	23	9	
2	2017-05-10 14:03:27 UTC	2017-05-23 13:12:27 UTC	2017-05-18 00:00:00 UTC	12	-5	
3	2017-04-22 15:50:30 UTC	2017-05-05 13:27:50 UTC	2017-05-18 00:00:00 UTC	12	12	
4	2017-05-09 17:42:45 UTC	2017-05-16 23:22:20 UTC	2017-05-18 00:00:00 UTC	7	1	
5	2017-04-26 01:01:39 UTC	2017-05-08 08:54:36 UTC	2017-05-18 00:00:00 UTC	12	9	

### Group data by state, take mean of freight\_value, time\_to\_deliver and diff\_estimated\_delivery

Below query shows mean freight\_value, mean time\_to\_deliver and mean diff\_estimated\_delivery as per the customer state

```
select T1.customer_state
,round(avg(T1.freight_value),2) as mean_freight_value
,round(avg(T1.time_to_deliver),2) as mean_time_to_deliver
,round(avg(T1.diff_estimated_delivery),2) as mean_diff_estimated_delivery
from
(SELECT c.customer_state
,o.order_purchase_timestamp
,o.order_delivered_customer_date
,o.order_estimated_delivery_date
,oi.freight_value
,timestamp_diff(o.order_delivered_customer_date,o.order_purchase_timestamp,day) as time_to_deliver
,timestamp_diff(o.order_estimated_delivery_date,o.order_delivered_customer_date,day) as diff_estimated_delivery
FROM
`retail-360819.Brazil.orders` o
,`retail-360819.Brazil.order_items` oi
,`retail-360819.Brazil.customers` c
where o.order_status = 'delivered'
and o.order_id = oi.order_id
and o.customer_id = c.customer_id
) T1
group by T1.customer_state
order by T1.customer_state;
```

Query results						<a href="#">SAVE RESULTS</a>	<a href="#">EXPLORE DATA</a>	
JOB INFORMATION		RESULTS	JSON		EXECUTION DETAILS			
Row	customer_state		mean_freight_value	mean_time_to_deliver	mean_diff_estimated_delivery			
1	AC		40.05	20.33	20.01			
2	AL		35.87	23.99	7.98			
3	AM		33.31	25.96	18.98			
4	AP		34.16	27.75	17.44			
5	BA		26.49	18.77	10.12			

Sort mean values to get top 5 states:  
Below query shows top 5 states with highest mean\_freight\_value

```
select T1.customer_state
,round(avg(T1.freight_value),2) as mean_freight_value
,round(avg(T1.time_to_deliver),2) as mean_time_to_deliver
,round(avg(T1.diff_estimated_delivery),2) as mean_diff_estimated_delivery
from
(SELECT c.customer_state
,o.order_purchase_timestamp
,o.order_delivered_customer_date
,o.order_estimated_delivery_date
,oi.freight_value
,timestamp_diff(o.order_delivered_customer_date,o.order_purchase_timestamp,day) as time_to_deliver
,timestamp_diff(o.order_estimated_delivery_date,o.order_delivered_customer_date,day) as diff_estimated_delivery
FROM
`retail-360819.Brazil.orders` o
,`retail-360819.Brazil.order_items` oi
,`retail-360819.Brazil.customers` c
where o.order_status = 'delivered'
and o.order_id = oi.order_id
and o.customer_id = c.customer_id
) T1
group by T1.customer_state
order by round(avg(T1.freight_value),2) desc
limit 5;
```

Query results						<a href="#">SAVE RESULTS</a>	<a href="#">EXPLORE DATA</a>	
JOB INFORMATION		RESULTS	JSON		EXECUTION DETAILS			
Row	customer_state		mean_freigh...	mean_time_...	mean_diff_e...			
1	RR		43.09	27.83	17.43			
2	PB		43.09	20.12	12.15			
3	RO		41.33	19.28	19.08			
4	AC		40.05	20.33	20.01			
5	PI		39.12	18.93	10.68			

Below query shows top 5 states with lowest mean\_freight\_value

```
select T1.customer_state
,round(avg(T1.freight_value),2) as mean_freight_value
,round(avg(T1.time_to_deliver),2) as mean_time_to_deliver
,round(avg(T1.diff_estimated_delivery),2) as mean_diff_estimated_delivery
from
(SELECT c.customer_state
,o.order_purchase_timestamp
,o.order_delivered_customer_date
,o.order_estimated_delivery_date
,oi.freight_value
,timestamp_diff(o.order_delivered_customer_date,o.order_purchase_timestamp,day) as time_to_deliver
,timestamp_diff(o.order_estimated_delivery_date,o.order_delivered_customer_date,day) as diff_estimated_delivery
FROM
`retail-360819.Brazil.orders` o
,`retail-360819.Brazil.order_items` oi
,`retail-360819.Brazil.customers` c
where o.order_status = 'delivered'
and o.order_id = oi.order_id
and o.customer_id = c.customer_id
) T1
group by T1.customer_state
order by round(avg(T1.freight_value),2) asc
limit 5;
```

Query results SAVE RESULTS EXPLORE DATA

JOB INFORMATION		RESULTS	JSON		EXECUTION DETAILS	
Row	customer_state	mean_freigh...	mean_time_...	mean_diff_e...		
1	SP	15.12	8.26	10.26		
2	PR	20.47	11.48	12.53		
3	MG	20.63	11.51	12.4		
4	RJ	20.91	14.69	11.14		
5	DF	21.07	12.5	11.27		

Below query shows top 5 states with highest mean\_time\_to\_deliver

```
select T1.customer_state
,round(avg(T1.freight_value),2) as mean_freight_value
,round(avg(T1.time_to_deliver),2) as mean_time_to_deliver
,round(avg(T1.diff_estimated_delivery),2) as mean_diff_estimated_delivery
from
(SELECT c.customer_state
,o.order_purchase_timestamp
,o.order_delivered_customer_date
,o.order_estimated_delivery_date
,oi.freight_value
,timestamp_diff(o.order_delivered_customer_date,o.order_purchase_timestamp,day) as time_to_deliver
,timestamp_diff(o.order_estimated_delivery_date,o.order_delivered_customer_date,day) as diff_estimated_delivery
FROM
`retail-360819.Brazil.orders` o
,`retail-360819.Brazil.order_items` oi
,`retail-360819.Brazil.customers` c
where o.order_status = 'delivered'
and o.order_id = oi.order_id
and o.customer_id = c.customer_id
) T1
group by T1.customer_state
order by round(avg(T1.time_to_deliver),2) desc limit 5;
```

Query results					SAVE RESULTS	EXPLORE DATA	
JOB INFORMATION		RESULTS	JSON		EXECUTION DETAILS		
Row	customer_state	mean_freigh...	mean_time...	mean_diff_e...			
1	RR	43.09	27.83	17.43			
2	AP	34.16	27.75	17.44			
3	AM	33.31	25.96	18.98			
4	AL	35.87	23.99	7.98			
5	PA	35.63	23.3	13.37			

**Below query shows top 5 states with lowest mean time\_to\_deliver**

```

select T1.customer_state
,round(avg(T1.freight_value),2) as mean_freight_value
,round(avg(T1.time_to_deliver),2) as mean_time_to_deliver
,round(avg(T1.diff_estimated_delivery),2) as mean_diff_estimated_delivery
from
(SELECT c.customer_state
,o.order_purchase_timestamp
,o.order_delivered_customer_date
,o.order_estimated_delivery_date
,oi.freight_value
,timestamp_diff(o.order_delivered_customer_date,o.order_purchase_timestamp,day) as time_to_deliver
,timestamp_diff(o.order_estimated_delivery_date,o.order_delivered_customer_date,day) as diff_estimated_delivery
FROM
`retail-360819.Brazil.orders` o
,`retail-360819.Brazil.order_items` oi
,`retail-360819.Brazil.customers` c
where o.order_status = 'delivered'
and o.order_id = oi.order_id
and o.customer_id = c.customer_id
) T1
group by T1.customer_state
order by round(avg(T1.time_to_deliver),2) asc
limit 5;

```

Query results					SAVE RESULTS	EXPLORE DATA	
JOB INFORMATION		RESULTS	JSON		EXECUTION DETAILS		
Row	customer_state	mean_freigh...	mean_time...	mean_diff_e...			
1	SP	15.12	8.26	10.26			
2	PR	20.47	11.48	12.53			
3	MG	20.63	11.51	12.4			
4	DF	21.07	12.5	11.27			
5	SC	21.51	14.52	10.66			

Below query shows top 5 states with highest mean diff\_estimated\_delivery

```
select T1.customer_state
,round(avg(T1.freight_value),2) as mean_freight_value
,round(avg(T1.time_to_deliver),2) as mean_time_to_deliver
,round(avg(T1.diff_estimated_delivery),2) as mean_diff_estimated_delivery
from
(SELECT c.customer_state
,o.order_purchase_timestamp
,o.order_delivered_customer_date
,o.order_estimated_delivery_date
,oi.freight_value
,timestamp_diff(o.order_delivered_customer_date,o.order_purchase_timestamp,day) as time_to_deliver
,timestamp_diff(o.order_estimated_delivery_date,o.order_delivered_customer_date,day) as diff_estimated_delivery
FROM
`retail-360819.Brazil.orders` o
,`retail-360819.Brazil.order_items` oi
,`retail-360819.Brazil.customers` c
where o.order_status = 'delivered'
and o.order_id = oi.order_id
and o.customer_id = c.customer_id
) T1
group by T1.customer_state
order by round(avg(T1.diff_estimated_delivery),2) desc
limit 5;
```

Query results

[SAVE RESULTS](#) [EXPLORE DATA](#)

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	
Row	customer_state	mean_freigh...	mean_time_...	mean_diff_e...	
1	AC	40.05	20.33	20.01	
2	RO	41.33	19.28	19.08	
3	AM	33.31	25.96	18.98	
4	AP	34.16	27.75	17.44	
5	RR	43.09	27.83	17.43	



Below query shows top 5 states with lowest mean diff\_estimated\_delivery

```
select T1.customer_state
,round(avg(T1.freight_value),2) as mean_freight_value
,round(avg(T1.time_to_deliver),2) as mean_time_to_deliver
,round(avg(T1.diff_estimated_delivery),2) as mean_diff_estimated_delivery
from
(SELECT c.customer_state
,o.order_purchase_timestamp
,o.order_delivered_customer_date
,o.order_estimated_delivery_date
,oi.freight_value
,timestamp_diff(o.order_delivered_customer_date,o.order_purchase_timestamp,day) as time_to_deliver
,timestamp_diff(o.order_estimated_delivery_date,o.order_delivered_customer_date,day) as diff_estimated_delivery
FROM
`retail-360819.Brazil.orders` o
,`retail-360819.Brazil.order_items` oi
,`retail-360819.Brazil.customers` c
where o.order_status = 'delivered'
and o.order_id = oi.order_id
and o.customer_id = c.customer_id
) T1
group by T1.customer_state
order by round(avg(T1.diff_estimated_delivery),2) asc
limit 5;
```

Query results SAVE RESULTS EXPLORE DATA




JOB INFORMATION		RESULTS	JSON		EXECUTION DETAILS	
Row	customer_state		mean_freigh_	mean_time_	mean_diff_e_	
1	AL		35.87	23.99	7.98	
2	MA		38.49	21.2	9.11	
3	SE		36.57	20.98	9.17	
4	ES		22.03	15.19	9.77	
5	BA		26.49	18.77	10.12	

- Q6: Payment type analysis:
- 1.Month over Month count of orders for different payment types
  - 2.Distribution of payment installments and count of orders

What is month over month count of orders for different payment types?




Below query shows % month over moth increase in order count for different payment types (replace year and payment as per the requirement)

```
select
T1.payment_type
,T1.year
,T1.month
,round(100*(T1.count - lag(T1.count, 1) over (order by T1.month)) / lag(T1.count, 1) over (order by T1.month),2) as percent_growth
from (
SELECT
p.payment_type
,extract(year from o.order_purchase_timestamp) as year
,extract(month from o.order_purchase_timestamp) as month
,count(o.order_id) as count
FROM
`retail-360819.Brazil.payments` p
,`retail-360819.Brazil.orders` o
where o.order_id = p.order_id
and o.order_status = 'delivered'
and extract(year from o.order_purchase_timestamp) = 2017
--and extract(year from o.order_purchase_timestamp) = 2018
and p.payment_type = 'credit_card'
--and p.payment_type = 'voucher'
--and p.payment_type = 'not_defined'
--and p.payment_type = 'debit_card'
--and p.payment_type = 'UPI'
group by
p.payment_type
,extract(year from o.order_purchase_timestamp)
,extract(month from o.order_purchase_timestamp)
) T1
order by
T1.payment_type
,T1.year
,T1.month;
```

Query results						 SAVE RESULTS ▾	 EXPLORE DATA ▾	
JOB INFORMATION		RESULTS	JSON		EXECUTION DETAILS			
Row	payment_type	year	month	percent_gro...				
1	credit_card	2017	1	null				
2	credit_card	2017	2	131.92				
3	credit_card	2017	3	51.79				
4	credit_card	2017	4	-7.13				
5	credit_card	2017	5	54.23				
6	credit_card	2017	6	-13.17				
7	credit_card	2017	7	25.33				
8	credit_card	2017	8	7.13				
9	credit_card	2017	9	-0.09				
10	credit_card	2017	10	7.32				




**Distribution of payment installments and count of orders:**  
**Below query shows distribution of order counts as per number of payment installments**

```
SELECT
*
FROM (
SELECT
EXTRACT(year
FROM
o.order_purchase_timestamp) AS year,
p.payment_installments,
COUNT(o.order_id) AS order_count
FROM
`retail-360819.Brazil.payments` p,
`retail-360819.Brazil.orders` o
WHERE
o.order_id = p.order_id
AND o.order_status = 'delivered'
GROUP BY
EXTRACT(year
FROM
o.order_purchase_timestamp),
p.payment_installments)
WHERE
year IN (2017,
2018)
ORDER BY
year,
payment_installments;
```

Query results					 SAVE RESULTS ▾	 EXPLORE DATA ▾	
JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS			
Row	year	payment_in...	order_count				
1	2017	1	22344				
2	2017	2	5184				
3	2017	3	4748				
4	2017	4	3249				
5	2017	5	2481				

**Additional Analysis:**  
**Product category and their dimensional properties mean**

```
select product_category
,round(avg(product_name_length),2) as mean_product_name_length
,round(avg(product_description_length),2) as mean_product_description_length
,round(avg(product_photos_qty),2) as mean_product_photos_qty
,round(avg(product_weight_g),2) as mean_product_weight_g
,round(avg(product_length_cm),2) as mean_product_length_cm
,round(avg(product_height_cm),2) as mean_product_height_cm
,round(avg(product_width_cm),2) as mean_product_width_cm
FROM
`retail-360819.Brazil.products`
group by product_category;
```

Query results									 SAVE RESULTS	 EXPLORE DATA	
JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS							
Row	product_category	mean_product_name_length	mean_product_description_length	mean_product_photos_qty	mean_product_weight_g	mean_product_length_cm	mean_product_height_cm	mean_product_width_cm			
1	null	null	null	null	2250.77	28.72	15.81	22.57			
2	babies	46.06	824.26	2.35	3655.2	37.15	2.62	28.72			
3	electronics	47.44	709.16	1.95	1275.09	25.24	11.74	18.47			
4	Watches present	49.8	611.98	2.23	509.29	19.22	10.29	15.27			
5	Garden tools	48.59	815.35	2.42	3103.78	30.94	19.2	23.5			
6	bed table bath	51.73	464.48	1.39	2456.41	36.77	14.09	30.59			

### Top 5 product categories with largest mean size volume in cm cube

```
select product_category, round((mean_product_length_cm * mean_product_height_cm * mean_product_width_cm), 2) as mean_size_volume
from (
select product_category
, round(avg(product_name_length), 2) as mean_product_name_length
, round(avg(product_description_length), 2) as mean_product_description_length
, round(avg(product_photos_qty), 2) as mean_product_photos_qty
, round(avg(product_weight_g), 2) as mean_product_weight_g
, round(avg(product_length_cm), 2) as mean_product_length_cm
, round(avg(product_height_cm), 2) as mean_product_height_cm
, round(avg(product_width_cm), 2) as mean_product_width_cm
FROM
`retail-360819.Brazil.products`
group by product_category)
order by round((mean_product_length_cm * mean_product_height_cm * mean_product_width_cm), 2) desc
limit 5;
```

Query results

SAVE RESULTS

EXPLORE DATA

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	product_category	mean_size_volume		
1	Furniture office	88303.23		
2	Furniture Kitchen Service Area ...	74123.38		
3	CITTE AND UPHACK FURNITURE	65779.34		
4	Furniture	63497.03		
5	ELECTRICES 2	53534.92		

### Top 5 states with Highest number of sellers

```
select seller_state,count(seller_id) as seller_count
FROM `retail-360819.Brazil.sellers`
group by seller_state
order by count(seller_id) desc
limit 5;
```

Query results

SAVE RESULTS

EXPLORE DATA

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	seller_state	seller_count		
1	SP	1849		
2	PR	349		
3	MG	244		
4	SC	190		
5	RJ	171		

Observations	Inference	Action Items
Time span of data given for delivered orders is Sept 2016 to Aug 2018	Useful range of time for data analysis is delivered orders which were created during Aug 2017 to Aug 2018	N/A
Top 5 customer states with highest number of cities are MG, SP, RS, PR, BA	All these 5 customer states have at least 350 cities from where orders were placed	Business should consider these 5 states while making any policy change decisions
Top 5 geolocation states with highest number of cities irrespective of order creation are SP, MG, RJ, RS, PR	All these 5 geolocation states have at least 650 cities within it	Business should take a look at state RJ as it is not top 5 states as per the order creation metric
Top 5 states with highest number of seller cities are SP, MG, PR, SC, RS	All these 5 states have at least 50 cities where sellers are located	Business should consider having more sellers in the states with large number of customers
There is growing trend on e-commerce looking at the orders and sales numbers in 2017 and 2018. Order count is increased by 21.72% and freight value is increased by 28.66%	There is growing trend on e-commerce year by year by more than 20%	Looking at the growing trend so rapidly, business should shift focus on online business and make necessary business decisions to invest more in warehouse management and prompt ship to home deliveries
Seasonality check shows that Oct, Nov and December are the peak season months where 40% of the total business occurs.	40% of the total business occurs during the month of Oct, Nov and Dec. This period seems to be the holiday season (especially, Thanksgiving, Cyber Monday, Christman, Ester)	Business should do accurate inventory management calculations and allocation more resources on logistic to deliver orders to the customers on time
Most of the Brazilian customers tend to do shopping during morning hours i.e. around 33% of total shopping in 2018	Nearly 33% of total shopping occurred during morning hours (5 am to 12 pm) in 2018. That means morning hours are having more online traffic of customers	Business should allocate more resources to website during morning hours and can potentially target ads if any during this time on the site and monitor the conversion rate
% increase in order count and freight value for span of Jan to Aug while comparing year 2018 with 2017. There is 157.26% increase in count of orders and 141.81% increase in freight value.	While comparing the data of 8 months (January to August) from 2017 and 2018. The results are stunning. There is more than 140% increase in both orders count and freight value. That means there is a very strong growing trend online purchase	Business need to invest more and need to make radical decisions to give justice to the huge growing trend of e-comm shopping. Customers need seamless experience and growing trend will create a very strong competitive environment.
Top 5 states with highest mean_freight_value are PR, PB, RO, AC, PI	The mean freight value of all these 5 states ranges between 39 to 43	Business can use this information in their decision making
Top 5 states with lowest mean_freight_value are SP, PR, MG, RJ, DF	The mean freight value of these 5 states ranges between 15 to 21	Business can use this information to improve the mean freight value
Top 5 states with highest mean_time_to_deliver are RR, AP, AM, AL, PA	The mean time_to_deliver of all these 5 states ranges between 23 and 27	Business can work in improving on these 5 states and reduce the time to deliver.
Top 5 states with lowest mean time_to_deliver are SP, PR, MG, DF, SC	The mean time_to_deliver of all these 5 states ranges from 8 to 14	Business can look at these states if there is anything specific here that other states can imitate
Top 5 states with highest mean diff_estimated_delivery are AC, RO, AM, AP, RR	The mean diff_estimated_delivery of all these 5 states ranges from 17 to 20	Please compare these values with actual time_to_deliver and work out the formula to improve
Top 5 states with lowest mean diff_estimated_delivery are AL, MA, SE, ES, BA	The mean diff_estimated_delivery of all these 5 states ranges from 7 to 10	Please compare these values with actual time_to_deliver and work out the formula to improve