

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

(A) Data type of all columns in the "customers" table.

Ans . output results for Data type of columns in customer table as follows

Filter Enter property name or value								
<input type="checkbox"/>	Field name	Type	Mode	Key	Collation	Default Value	Policy Tags	Description
<input type="checkbox"/>	customer_id	STRING	NULLABLE					
<input type="checkbox"/>	customer_unique_id	STRING	NULLABLE					
<input type="checkbox"/>	customer_zip_code_prefix	INTEGER	NULLABLE					
<input type="checkbox"/>	customer_city	STRING	NULLABLE					
<input type="checkbox"/>	customer_state	STRING	NULLABLE					

(B) Get the time range between which the orders were placed.

Ans. Query as follows for time range

```
with sorting as(select order_purchase_timestamp as a from
`casestudy-389916.Tables.orders` order by order_purchase_timestamp limit 1)
, sorting1 as(select order_purchase_timestamp as b from
`casestudy-389916.Tables.orders` order by order_purchase_timestamp desc
limit 1)
select a as time_range from sorting union all
select b from sorting1 order by time_range
```

Row	time_range
1	2016-09-04 21:15:19 UTC
2	2018-10-17 17:30:18 UTC

(C) Count the number of Cities and States in our dataset.

Ans. Query as follows for unique cities and states

```
select count(distinct geolocation_state) as unique_state,count(distinct
geolocation_city) as unique_city
from `casestudy-389916.Tables.geolocation`
```

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	EXECUTION GRAPH
Row	unique_state	unique_city			
1	27	8011			

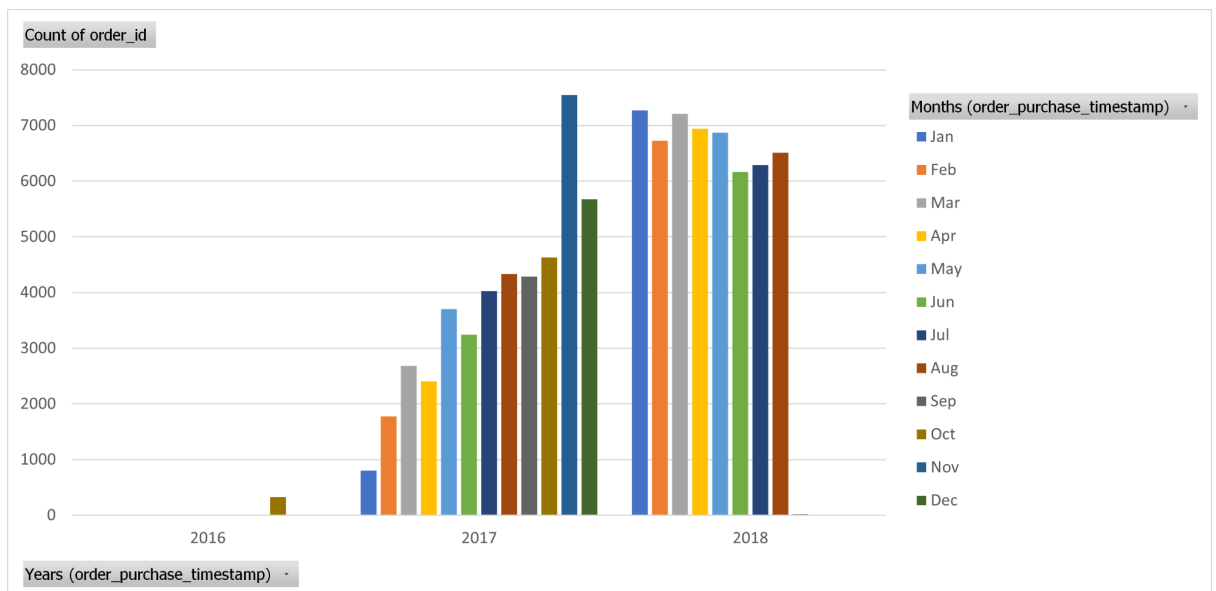
2. In-depth Exploration:

(A) Is there a growing trend in the no. of orders placed over the past years?

Ans. Yes there is growing trend in month over past years , i have attached both SQL query and graph

```
with uni as (select distinct format_date('%b-%Y',order_purchase_timestamp)
as monthly ,count(order_id) over(partition by format_date('%b-%Y' ,
order_purchase_timestamp) ) as unique_order from
`casestudy-389916.Tables.orders` order by monthly desc)
select * from uni
```

Row	monthly	unique_order
1	Sep-2018	16
2	Sep-2017	4285
3	Sep-2016	4
4	Oct-2018	4
5	Oct-2017	4631
6	Oct-2016	324
7	Nov-2017	7544
8	May-2018	6873
9	May-2017	3700
10	Mar-2018	7211



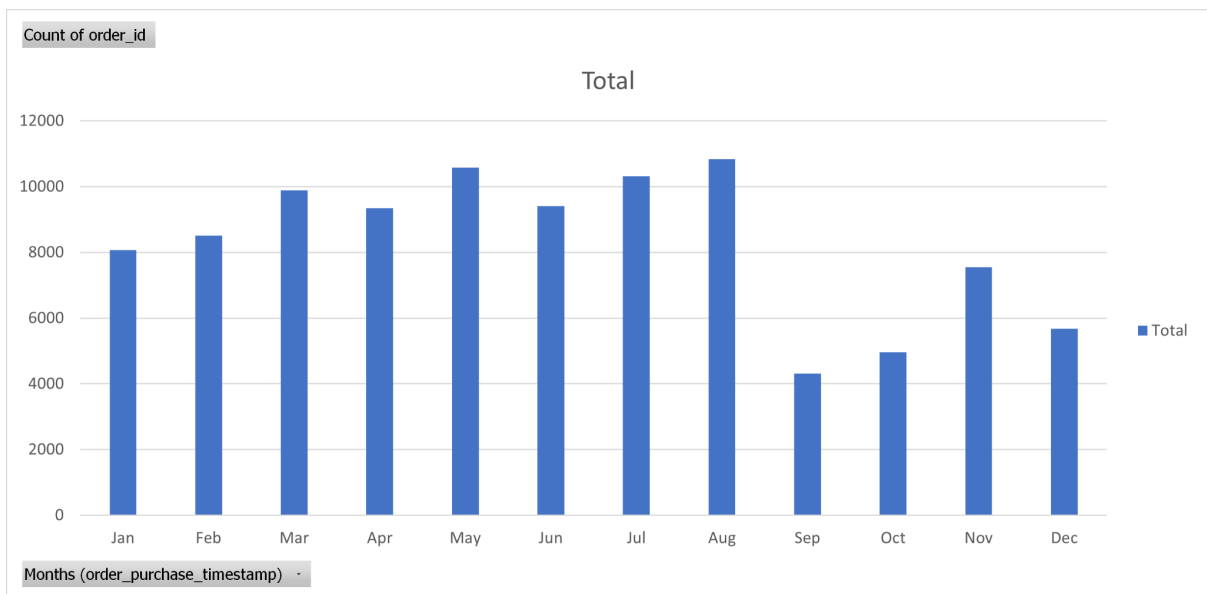
(B) Can we see some kind of monthly seasonality in terms of the no. of orders being placed?

Ans. There is peak in month of August where Customers are placed order rapidly

SQL query and graph as follows.

```
with uni as (select distinct format_date('%b',order_purchase_timestamp) as
monthly ,count(order_id) over(partition by format_date('%b' ,
order_purchase_timestamp) ) as unique_order
from `casestudy-389916.Tables.orders` order by unique_order desc)
select * from uni
```

Row	monthly	unique_order
1	Aug	10843
2	May	10573
3	Jul	10318
4	Mar	9893
5	Jun	9412
6	Apr	9343
7	Feb	8508
8	Jan	8069
9	Nov	7544
10	Dec	5674
11	Oct	4959



(C) During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)

- 0-6 hrs : Dawn
- 7-12 hrs : Mornings
- 13-18 hrs : Afternoon
- 19-23 hrs : Night

Ans. Brazilian customers mostly order during afternoon period of day  
SQL query as follows

```
with hour_stamp as (select order_id,extract(hour from
order_purchase_timestamp) as hourr from `casestudy-389916.Tables.orders`
order by hourr desc ),
hour_new as (select order_id,case when hour_stamp.hourr between 0 and 6
then 'Dawn' when hour_stamp.hourr between 7 and 12 then 'Mornings' when
hour_stamp.hourr between 13 and 18 then 'Afternoon' when hour_stamp.hourr
between 19 and 23 then 'Night' end as hour_count from hour_stamp)
select distinct count(order_id) over(partition by hour_count) as
order_count,hour_count from hour_new
```

Row	order_count	hour_count
1	27733	Mornings
2	5242	Dawn
3	38135	Afternoon
4	28331	Night

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

(A) Get the month on month no. of orders placed in each state.

Ans. order placed by Customer in each state by month

Query as follows.

```
select distinct count(ord.customer_id) over(partition by
custo.customer_state,format_date('%b',ord.order_purchase_timestamp)) as
order_by_state,
custo.customer_state, format_date('%b',ord.order_purchase_timestamp) as
month
from `casestudy-389916.Tables.customers` custo right join
`casestudy-389916.Tables.orders` ord on custo.customer_id=ord.customer_id
order by month
```

Row	order_by_state	customer_state	month
1	11	AP	Jan
2	427	RS	Jan
3	159	ES	Jan
4	55	PI	Jan
5	24	SE	Jan
6	71	MS	Jan
7	96	MT	Jan
8	39	AL	Jan
9	2	RR	Jan
10	82	PA	Jan

(B) How are the customers distributed across all the states?

Ans. Query as follows for unique customer in each state with output result

```
select customer_state,count(distinct customer_unique_id) as
customer_by_state from `casestudy-389916.Tables.customers` group by
customer_state order by customer_by_state
```

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	EXECUTION GRAPH
Row	customer_state	customer_by_state			
1	RR	45			
2	AP	67			
3	AC	77			
4	AM	143			
5	RO	240			
6	TO	273			
7	SE	342			
8	AL	401			
9	RN	474			
10	PI	482			

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

(A) Get the % increase in the cost of orders from year 2017 to 2018 (include months between Jan to Aug only).

Ans. In 2018 there is growth in payment\_value approx 136% compare to 2017 SQL as follows .

```
with per as (select distinct sum(p.payment_value) over(partition by
format_date('%Y',o.order_purchase_timestamp)) as Total_payment,
format_date('%Y',o.order_purchase_timestamp) as mon
from `casestudy-389916.Tables.payments` p join `casestudy-389916.Tables.orders` o
on p.order_id=o.order_id
where format_date('%m-%y',o.order_purchase_timestamp) between '01-17' and '08-18'
order by mon)
select mon,Total_payment, (nth_value(Total_payment,2) over(order by Total_payment)-
nth_value(Total_payment,1) over(order by Total_payment))/
nth_value(Total_payment,1) over(order by Total_payment)*100 as percent_growth from
per
```

Row	mon	Total_payment	percent_growth
1	2017	3669022.12	null
2	2018	8694733.84	136.9768716466...

(B) Calculate the Total & Average value of order price for each state.

Ans. Total price and avg price by each state SQL query as follows

```
select distinct sum(p.payment_value) over(partition by c.customer_state ) as
Total_by_state,
avg(p.payment_value) over(partition by c.customer_state) as AVG_by_state,
c.customer_state from `casestudy-389916.Tables.payments` p join
`casestudy-389916.Tables.orders` o on p.order_id=o.order_id
join `casestudy-389916.Tables.customers` c on o.customer_id=c.customer_id order by
c.customer_state
```

Row	Total_by_state	AVG_by_state	customer_state
1	19680.62	234.2930952380...	AC
2	96962.06	227.0774238875...	AL
3	27966.93	181.6034415584...	AM
4	16262.8	232.3257142857...	AP
5	616645.82	170.8160166204...	BA
6	279464.03	199.9027396280...	CE
7	355141.08	161.1347912885...	DF
8	325967.55	154.7069530137...	ES
9	350092.31	165.7634043560...	GO
10	152523.02	198.8566101694...	MA

(C) Calculate the Total & Average value of order freight for each state.

Ans. Total&Avg freight for each state as follows.

SQL query as follows.

```
select distinct sum(p.freight_value) over(partition by c.customer_state ) as
freight_total_by_state,
avg(p.freight_value) over(partition by c.customer_state) as freight_AVG_by_state,
c.customer_state from `casestudy-389916.Tables.order_items` p join
`casestudy-389916.Tables.orders` o on p.order_id=o.order_id
join `casestudy-389916.Tables.customers` c on o.customer_id=c.customer_id order by
c.customer_state
```

Row	freight_total_by_state	freight_AVG_by_state	customer_state
1	3686.75	40.0733695652173...	AC
2	15914.59	35.8436711711711...	AL
3	5478.89	33.2053939393939...	AM
4	2788.5	34.0060975609756...	AP
5	100156.68	26.3639589365622...	BA
6	48351.59	32.7142016238159...	CE
7	50625.5	21.0413549459684...	DF
8	49764.6	22.0587765957446...	ES
9	53114.98	22.7668152593227...	GO
10	31523.77	38.2570024271844...	MA

## 5. Analysis based on sales, freight and delivery time.

(A) Find the no. of days taken to deliver each order from the order's purchase date as delivery time.

Also, calculate the difference (in days) between the estimated & actual delivery date of an order.

Do this in a single query.

**Ans.** time to deliver the product to customer and difference in actual and estimate date **where negative value indicate product delivery get late in no of days** SQL query as follows.

```

select
order_purchase_timestamp,order_delivered_customer_date,order_estimated_delivery_date
,date_diff(order_delivered_customer_date,order_purchase_timestamp,day) as
time_to_deliver,
date_diff(order_estimated_delivery_date,order_delivered_customer_date,day)
as diff_estimated_delivery from `casestudy-389916.Tables.orders` where
order_delivered_customer_date is not null
order by time_to_deliver,diff_estimated_delivery

```

Row	order_purchase_timestamp	order_delivered_customer_date	order_estimated_delivery_date	time_to_deliver	diff_estimated_delivery
1	2017-05-15 11:50:53 UTC	2017-05-16 10:21:52 UTC	2017-05-24 00:00:00 UTC	0	7
2	2018-06-18 12:59:42 UTC	2018-06-19 12:43:27 UTC	2018-06-28 00:00:00 UTC	0	8
3	2018-05-18 15:03:19 UTC	2018-05-19 12:28:30 UTC	2018-05-29 00:00:00 UTC	0	9
4	2018-05-14 12:20:06 UTC	2018-05-15 12:17:46 UTC	2018-05-25 00:00:00 UTC	0	9
5	2017-06-19 08:19:45 UTC	2017-06-19 21:07:52 UTC	2017-06-30 00:00:00 UTC	0	10
6	2017-11-16 13:54:08 UTC	2017-11-17 13:49:40 UTC	2017-11-29 00:00:00 UTC	0	11
7	2017-07-04 11:37:47 UTC	2017-07-05 08:09:26 UTC	2017-07-17 00:00:00 UTC	0	11
8	2017-05-31 12:00:35 UTC	2017-06-01 10:28:24 UTC	2017-06-13 00:00:00 UTC	0	11
9	2018-06-28 14:34:48 UTC	2018-06-29 14:12:18 UTC	2018-07-12 00:00:00 UTC	0	12
10	2018-02-02 15:26:38 UTC	2018-02-03 15:05:56 UTC	2018-02-20 00:00:00 UTC	0	16

(B) Find out the top 5 states with the highest & lowest average freight value.

Ans. Top 5 and bottom 5 States are here with highest and lowest freight value

```

with fre as (select distinct avg(p.freight_value) over(partition by
c.customer_state ) as Highest_freight_Top5_state ,c.customer_state from
`casestudy-389916.Tables.order_items` p join `casestudy-389916.Tables.orders` o on
p.order_id=o.order_id
join `casestudy-389916.Tables.customers` c on o.customer_id=c.customer_id order by
Highest_freight_Top5_state limit 5 ),
fre1 as (select distinct avg(p.freight_value) over(partition by c.customer_state
) as Highest_freight_Top5_state ,c.customer_state from
`casestudy-389916.Tables.order_items` p join `casestudy-389916.Tables.orders` o on
p.order_id=o.order_id
join `casestudy-389916.Tables.customers` c on o.customer_id=c.customer_id order by
Highest_freight_Top5_state desc limit 5 )
select Highest_freight_Top5_state, customer_state
from fre
union all
select Highest_freight_Top5_state,customer_state
from fre1
order by Highest_freight_Top5_state

```

Row	Highest_freight_Top5_state	customer_state
1	15.147275390419187	SP
2	20.531651567944252	PR
3	20.630166806306651	MG
4	20.960923931682558	RJ
5	21.041354945968411	DF
6	39.147970479704796	PI
7	40.073369565217391	AC
8	41.069712230215828	RO
9	42.723803986710962	PB
10	42.984423076923079	RR

(C) Find out the top 5 states with the highest & lowest average delivery time.

Ans. Top 5 and bottom 5 States are here with highest and lowest average delivery time

```

with top as(select distinct avg(date_diff
(o.order_estimated_delivery_date,o.order_purchase_timestamp,day)) over(partition by
c.customer_state) as Top_avg_delivery,c.customer_state
from `casestudy-389916.Tables.orders` o join `casestudy-389916.Tables.customers` c
on o.customer_id=c.customer_id
where o.order_delivered_customer_date is not null
order by Top_avg_delivery
limit 5),
low as (select distinct avg(date_diff
(o.order_estimated_delivery_date,o.order_purchase_timestamp,day)) over(partition by
c.customer_state) as Bottom_avg_delivery,c.customer_state
from `casestudy-389916.Tables.orders` o join `casestudy-389916.Tables.customers` c
on o.customer_id=c.customer_id
where o.order_delivered_customer_date is not null
order by Bottom_avg_delivery desc
limit 5)
select Top_avg_delivery,t.customer_state from top t
union all
select Bottom_avg_delivery,customer_state from
low l
order by Top_avg_delivery

```



Row	Top_avg_delivery	customer_state
1	18.77691072971...	SP
2	23.94711538461...	DF
3	24.18731836195...	MG
4	24.25248832013...	PR
5	25.22005012531...	ES
6	38.38683127572...	RO
7	40.725	AC
8	44.92413793103...	AM
9	45.63414634146...	RR
10	45.86567164179...	AP

(D) Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.

Ans. Here is Top 5 states with fast delivery Avg.

SQL query as follows.

```
select
avg(date_diff(o.order_estimated_delivery_date,o.order_delivered_customer_date,day))
as fast_delivery_time,c.customer_state from `casestudy-389916.Tables.orders` o
join `casestudy-389916.Tables.customers` c on o.customer_id=c.customer_id
where order_delivered_customer_date is not null
group by c.customer_state
order by fast_delivery_time desc
limit 5
```

Row	fast_delivery_time	customer_state
1	19.7625000000000...	AC
2	19.13168724279836	RO
3	18.7313432835820...	AP
4	18.60689655172413	AM
5	16.4146341463414...	RR

## 6. Analysis based on the payments:

(A) Find the month on month no. of orders placed using different payment types.

Ans. There is various type of payment sort the order by month over past years

SQL query as follows

```
select distinct count(p.order_id) over(partition by
format_date('%m',o.order_purchase_timestamp),p.payment_type) as by_payment ,
p.payment_type ,format_date('%b',o.order_purchase_timestamp) as month
from `casestudy-389916.Tables.payments` p join `casestudy-389916.Tables.orders` o
on p.order_id=o.order_id
order by month
```

Row	by_payment ▼	payment_type ▼	month ▼ ↑
1	1783	UPI	Apr
2	124	debit_card	Apr
3	7301	credit_card	Apr
4	572	voucher	Apr
5	2077	UPI	Aug
6	8269	credit_card	Aug
7	589	voucher	Aug
8	311	debit_card	Aug
9	2	not_defined	Aug
10	64	debit_card	Dec

(B) Find the no. of orders placed on the basis of the payment installments that have been paid.

Ans.

```
select count(order_id) as order_by_installments, payment_installments
from `casestudy-389916.Tables.payments`
where payment_sequential >=1
group by payment_installments
order by payment_installments
```

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

### Recommendations

**On Freight** there is some states where we have to decrease freight for increasing in sales

**In Sales** : sales are increasing per year but we have to keep growing as in month of august sales are on high

**Seller** we have to only seller in 23 states we have to increase seller as well in other states

**Delivery Time** we have to decrease the delivery time customers are getting delay their product it's effect on sales as well

**Expected delivery time** : we have to decrease in expected date as well because some product are getting very delay after expected date

**Customers Reviews** : we have to keep motivated to customers for review because there is total 1 lakh order and we get approx 11 k reviews

Reviews can improve our sales and product quality and services

**Order cancellation** : many customers are facing cancellation from our end just because of delay delivery time we have to decrease cancellation for customers