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

The target dataset has 8 tables with 99441 rows or orders with customer details, order details and payment details for all orders of target in Brazil from 2016 to 2018.

The data types used across various table in the target dataset are:

- **Customer table**
  - **String**
  - **Integer**
- **Geolocation**
  - **Integer**
  - **Float**
  - **String**
- **Order Item**
  - **String**
  - **Integer**
  - **Float**
  - **Timestamp**
- **Order reviews**
  - **String**
  - **Integer**
  - **Timestamp**
- **Orders**
  - **String**
  - **Timestamp**
- **Payment**
  - **String**
  - **Integer**
  - **Float**
- **Products**
  - **String**

- ➢ Integer

❖ Seller

- ➢ String
- ➢ Integer

1.2 Time period for which the data is given

Data is provided for all orders in Brazil from 2016-09-04 21:15:19 UTC to 2018-10-17 17:30:18 UTC

Query-

```
select min(order_purchase_timestamp) as minimum,
max(order_purchase_timestamp)as maximum

from `target_dataset.orders`
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

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

1.3 Cities and States of customers ordered during the given period

Customers who ordered during this time period are from the 4119 different cities and 27 states across Brazil:

Query-

```
select count(distinct(customer_city)) as City, count(distinct(customer_state))
as State
from `target_dataset.customers`
```

Row	city	state
1	4119	27

In-depth Exploration:

2.1 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 seems to be a growing trend on e-commerce in Brazil as the total orders in the year 2016(september-december)is 329 followed by 2017(january-december) is 45101 and 2018(january-october) is 54011, however the sales are high from november 2017 - january 2018 assuming the raise is due to the holiday season.

Query -

```
select
extract(year from order_purchase_timestamp) as Year,
extract(month from order_purchase_timestamp) as Month,
count(order_id) as order_count
from `target_dataset.orders`
group by Year,Month
order by Year,Month
```

JOB INFORMATION | RESULTS | JSON | E

Row	Year	Month	order_count
1	2016	9	4
2	2016	10	324
3	2016	12	1
4	2017	1	800
5	2017	2	1780
6	2017	3	2682
7	2017	4	2404
8	2017	5	3700
9	2017	6	3245
10	2017	7	4026

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

Dawn (3AM - 6AM) - 1168
Morning (7AM - 11AM) - 21738

Afternoon (12PM - 5PM) - 38361
Night (6PM - 12AM) - 36494

The Brazilian customers tend to buy at noon, however the most purchases are done between 1PM to 5PM , a similar trend is observed even at night.

Query-

```
select
extract(hour from order_purchase_timestamp) as Hour,
count(order_id) as order_count
from `target_dataset.orders`
where extract(hour from order_purchase_timestamp) between 12 and 17
group by Hour
order by Hour
```

JOB INFORMATION | RESULTS

Row	Hour	order_count
1	12	5995
2	13	6518
3	14	6569
4	15	6454
5	16	6675
6	17	6150

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

Query -

```
select c.customer_state,

extract(year from o.order_purchase_timestamp) as Year,

extract(month from o.order_purchase_timestamp) as Month,

count(o.order_id) as order_count

from `target_dataset.orders` as o join `target_dataset.customers` as c

on o.customer_id=c.customer_id
```

```
group by c.customer_state,Year,Month

order by c.customer_state,Year,Month
```

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTIC

Row	customer_state	Year	Month	order_count
1	AC	2017	1	2
2	AC	2017	2	3
3	AC	2017	3	2
4	AC	2017	4	5
5	AC	2017	5	8
6	AC	2017	6	4
7	AC	2017	7	5
8	AC	2017	8	4
9	AC	2017	9	5
10	AC	2017	10	6

3.2 Distribution of customers across the states in Brazil

Query -

```
select customer_state,
count(customer_id) as Customers
from `target_dataset.customers`
group by customer_state
order by count(customer_id) desc
```

JOB INFORMATION | RESULTS | JSON | EX

Row	customer_state	Customers
1	SP	41746
2	RJ	12852
3	MG	11635
4	RS	5466
5	PR	5045
6	SC	3637
7	BA	3380
8	DF	2140
9	ES	2033
10	GO	2020

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

Get % increase in cost of orders from 2017 to 2018 (include months between Jan to Aug only) - You can use the “payment_value” column in the payments table.

There is an increase of 136.98% in cost of orders from 2017 to 2018.

Query -

```
with cte as (

  select

  extract(year from o.order_purchase_timestamp) as Year,

  sum(p.payment_value) as cost

  from `target_dataset.payments` as p join `target_dataset.orders` as o

  on p.order_id =o.order_id

  where extract(month from o.order_purchase_timestamp) between 1 and 8

  group by 1

  order by 1

  ),

cte2 as (select c.cost as total from cte as c where c.Year=2017)



select Year,cost,

case

when Year=2017

then 0
```

```
else round(100*(cost-(select total from cte2))/(select total from cte2),2)

end as IncreasePercentage

from cte

order by 1
```

Query results

JOB INFORMATION | RESULTS | JSON | EXE

Row	Year	cost	IncreasePercent
1	2017	3669022.11...	0.0
2	2018	8694733.83...	136.98

4.2 Mean & Sum of price and freight value by customer state

Query -

```
select c.customer_state,

sum(i.price) as price_sum,

avg(i.price) as price_avg,

sum(i.freight_value) as freight_sum,

avg(i.freight_value) as freight_avg

from `target_dataset.customers` as c join `target_dataset.orders` as o

on c.customer_id = o.customer_id

join `target_dataset.order_item` as i

on i.order_id =o.order_id

group by c.customer_state

order by c.customer_state
```

Query results

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

Row	customer_state	price_sum	price_avg	freight_sum	freight_avg
1	AC	15982.9499...	173.727717...	3686.74999...	40.0733695...
2	AL	80314.8100...	180.889211...	15914.5899...	35.8436711...
3	AM	22356.8400...	135.495999...	5478.88999...	33.2053939...
4	AP	13474.2999...	164.320731...	2788.50000...	34.0060975...
5	BA	511349.990...	134.601208...	100156.679...	26.3639589...
6	CE	227254.709...	153.758261...	48351.5899...	32.7142016...
7	DF	302603.939...	125.770548...	50625.4999...	21.0413549...
8	ES	275037.309...	121.913701...	49764.5999...	22.0587765...
9	GO	294591.949...	126.271731...	53114.9799...	22.7668152...
10	MA	119648.219...	145.204150...	31523.7700...	38.2570024...

5.1 Analysis on sales, freight and delivery time

Calculate days between purchasing, delivering and estimated delivery

5.2 Find time_to_delivery & diff_estimated_delivery. Formula for the same given below:

- **time_to_delivery = order_purchase_timestamp-order_delivered_customer_date**
- **diff_estimated_delivery = order_estimated_delivery_date-order_delivered_customer_date**

Query-

```
select order_id,

ifnull(date_diff(order_delivered_customer_date,order_purchase_timestamp,day),0)
as time_to_delivery,

ifnull(date_diff(order_delivered_customer_date,order_estimated_delivery_date,da
y),0) as diff_estimated_delivery,

ifnull(date_diff(order_estimated_delivery_date,order_purchase_timestamp,day),0)
as diff_purchase_to_estimated_delivery

from `target_dataset.orders`
```

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTI

Row	order_id	time_to_delivery	diff_estimated_c	diff_purchase_tc
1	f88aac7ebccb37f19725a0753...	0	0	50
2	790cd37689193dca0d00d2feb...	0	0	6
3	49db7943d60b6805c3a41f547...	0	0	44
4	063b573b88fc80e516aba87df...	0	0	54
5	a68ce1686d536ca72bd2dadc4...	0	0	56
6	45973912e490866800c0aea8f...	0	0	54
7	cda873529ca7ab71f677d5ec1...	0	0	56
8	ead20687129da8f5d89d831bb...	0	0	41
9	6f028ccb7d612af251aa442a1f...	0	0	3
10	8733c8d440c173e524d2fab80...	0	0	3

5.3 Group data by state, take mean of freight_value, time_to_delivery, diff_estimated_delivery

Query -

```
select c.customer_state,

avg(ifnull(date_diff(order_delivered_customer_date,order_purchase_timestamp,day),0)) as time_to_delivery,

avg(ifnull(date_diff(order_delivered_customer_date,order_estimated_delivery_date,day),0)) as diff_estimated_delivery,

avg(i.freight_value) as freight_avg

from `target_dataset.customers` as c join `target_dataset.orders` as o

on c.customer_id = o.customer_id

join `target_dataset.order_item` as i

on i.order_id =o.order_id

group by c.customer_state
```

order by c.customer_state

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECU

Row	customer_state	time_to_delivery	diff_estimated_d	freight_avg
1	AC	20.1086956...	-19.793478...	40.0733695...
2	AL	23.0743243...	-7.6711711...	35.8436711...
3	AM	25.6484848...	-18.745454...	33.2053939...
4	AP	27.4146341...	-17.231707...	34.0060975...
5	BA	18.2013687...	-9.8104764...	26.3639589...
6	CE	19.8146143...	-9.8958051...	32.7142016...
7	DF	12.2364921...	-11.035743...	21.0413549...
8	ES	14.9840425...	-9.6343085...	22.0587765...
9	GO	14.5893699...	-11.099871...	22.7668152...
10	MA	20.5861650...	-8.8446601...	38.2570024...

5.4 Sort the data to get the following:

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

5 Highest freight value

Query -

```
select c.customer_state, avg(i.freight_value) as higest_avg_freight

from `target_dataset.order_item` as i join `target_dataset.orders` as o

on i.order_id =o.order_id

join `target_dataset.customers` as c

on c.customer_id =o.customer_id

group by c.customer_state

order by avg(i.freight_value) desc
```

```
limit 5
```

JOB INFORMATION | RESULTS | JSON

Row	customer_state	higest_avg_freig
1	RR	42.9844230...
2	PB	42.7238039...
3	RO	41.0697122...
4	AC	40.0733695...
5	PI	39.1479704...

5 Lowest freight value -

```
select c.customer_state, avg(i.freight_value) as lowest_avg_freight

from `target_dataset.order_item` as i join `target_dataset.orders` as o

on i.order_id =o.order_id

join `target_dataset.customers` as c

on c.customer_id =o.customer_id

group by c.customer_state

order by avg(i.freight_value)

limit 5
```

JOB INFORMATION | RESULTS | JSON

Row	customer_state	lowest_avg_freig
1	SP	15.1472753...
2	PR	20.5316515...
3	MG	20.6301668...
4	RJ	20.9609239...
5	DF	21.0413549...

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

Highest average time to delivery -

```
select c.customer_state,

avg(ifnull(date_diff(order_delivered_customer_date,order_purchase_timestamp,day
),0)) as time_to_delivery,

from `target_dataset.customers` as c join `target_dataset.orders` as o

on c.customer_id = o.customer_id

join `target_dataset.order_item` as i

on i.order_id =o.order_id

group by c.customer_state

order by
avg(ifnull(date_diff(order_delivered_customer_date,order_purchase_timestamp,day
),0)) desc

limit 5
```

JOB INFORMATION | RESULTS | JSON

Row	customer_state	time_to_delivery
1	AP	27.4146341...
2	AM	25.6484848...
3	RR	24.6153846...
4	AL	23.0743243...
5	PA	22.7407407...

Lowest average time to delivery -

Query -

```
select c.customer_state,

avg(ifnull(date_diff(order_delivered_customer_date,order_purchase_timestamp,day
),0)) as time_to_delivery,

from `target_dataset.customers` as c join `target_dataset.orders` as o
```

```
on c.customer_id = o.customer_id

join `target_dataset.order_item` as i

on i.order_id =o.order_id

group by c.customer_state

order by
avg(ifnull(date_diff(order_delivered_customer_date,order_purchase_timestamp,day
),0))

limit 5
```

JOB INFORMATION | RESULTS | JSON

Row	customer_state	time_to_delivery
1	SP	8.08449071...
2	PR	11.2987804...
3	MG	11.3295757...
4	DF	12.2364921...
5	SC	14.2497605...

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

Query -

```
 select c.customer_state,

avg(ifnull(date_diff(order_delivered_customer_date,order_purchase_timestamp,day
),0)) as time_to_delivery,

avg(ifnull(date_diff(order_estimated_delivery_date,order_purchase_timestamp,day
),0)) as diff_estimated_delivery

from `target_dataset.customers` as c join `target_dataset.orders` as o

on c.customer_id = o.customer_id

join `target_dataset.order_item` as i

on i.order_id =o.order_id
```

```
where
(ifnull(date_diff(order_estimated_delivery_date,order_purchase_timestamp,day),0
))<(ifnull(date_diff(order_delivered_customer_date,order_purchase_timestamp,day
),0))

group by c.customer_state

order by
avg(ifnull(date_diff(order_delivered_customer_date,order_purchase_timestamp,day
),0))

limit 5
```

JOB INFORMATION | RESULTS | JSON | EXECUTION DET

Row	customer_state	time_to_delivery	diff_estimated_d
1	SP	22.7239819...	15.2114356...
2	MS	29.0476190...	21.4404761...
3	DF	29.3251533...	21.8404907...
4	MG	29.4654377...	21.5130568...
5	TO	30.2702702...	24.3243243...

6.1 Payment type analysis:

Month over Month count of orders for different payment types

Query -

```
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 number_of_orders

from `target_dataset.orders` as o join `target_dataset.payments` as p

on o.order_id=p.order_id

group by payment_type,Year,Month
```

```
order by payment_type,Year,Month
```

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAF

Row	payment_type	Year	Month	number_of_orde
1	UPI	2016	10	63
2	UPI	2017	1	197
3	UPI	2017	2	398
4	UPI	2017	3	590
5	UPI	2017	4	496
6	UPI	2017	5	772
7	UPI	2017	6	707
8	UPI	2017	7	845
9	UPI	2017	8	938
10	UPI	2017	9	903

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

Query -

```
select p.payment_installments,
 count(o.order_id) as number_of_orders
 from `target_dataset.orders` as o join `target_dataset.payments` as p
 on o.order_id=p.order_id
 group by p.payment_installments
 order by  p.payment_installments
```

JOB INFORMATION | RESULTS

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

Actionable Insights -

- **Target dataset has 8 tables and 99441 records from 27 states and 4119 cities of Brazil from September 2016 to October 2018.**
- **There seems to be a growing trend on e-commerce in Brazil as the total orders in the year 2016(september-december)is 329 followed by 2017(january-december) is 45101 and 2018(january-october) is 54011, however the sales are high from november 2017 - january 2018 assuming the raise is due to the holiday season.**
- **The Brazilian customers tend to buy at noon, however the most purchases are done between 1PM to 5PM , a similar trend is observed even at night.**
- **SP State has the highest number of customers of 41746 customers followed by RJ State.**
- **There is 137% increase in cost of orders from 2017-2018 without considering the months with high sales in November and December.**
- **RR state has the highest freight value and SP state has the lowest freight value.**

- **AP state has the highest average time to delivery and SP has the lowest average time to delivery.**
- **Majority of the customers prefer one time installments for their payments.**

Recommendations -

- **Measures must be taken to decrease the delivery time. Can charge a nominal fee for a speedy delivery**
- **Launch offers to increase sales from February to September.**
- **Marketing to increase the awareness of e-commerce among customers.**
- **Create loyalty programs to increase sales by reward points.**
- **Generate user specific ads based on previous purchases.**
- **Offer free delivery to increase sales.**
- **Can provide different modes of payment apart from credit cards and vouchers.**