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

    a. **Data type of columns in a table**

    customer.csv
    - Have columns with string and integer data types

    geolocation.csv
    - Have columns with string, float, integer data types

    order_items.csv
    - Have columns with string, integer, timestamp and float

    order_reviews.csv
    - Have columns with string, integer and timestamp

    orders.csv
    - Have columns with string and timestamp

    payments.csv
    - Have columns with string, integer and float

    products.csv
    - Have columns with string and integer

    sellers.csv
    - Have columns with string and integer

    b. **Time period for which the data is given**
    - Orders tables

    Query =>
    ```
    select order_purchase_timestamp from `target.orders` as o
    order by o.order_purchase_timestamp limit 1;
    ```

    ```
    select order_purchase_timestamp from `target.orders` as o order by
    o.order_purchase_timestamp desc limit 1;
    ```

    ```
    select count(*), timestamp_trunc(o.order_purchase_timestamp,YEAR)
    as order_date from `target.orders` as o group by order_date;
    ```

Row	order_purchase_timestamp
1	2018-10-17 17:30:18 UTC

Row	order_purchase_timestamp
1	2016-09-04 21:15:19 UTC

Row	f0_	order_date
1	45101	2017-01-01 00:00:00 UTC
2	54011	2018-01-01 00:00:00 UTC
3	329	2016-01-01 00:00:00 UTC

- Orders table have data between 04/09/16 and 17/10/18
- Total 329 orders was place in 2016, 45101 in 2017 and 54011 in 2018

c. **Cities and States of customers ordered during the given period**

Query=>
```
select count(*) as customer_count_per_state, c.customer_state from `target.customers` as c group by c.customer_state order by customer_count_per_state desc;

select count(*) as customer_count_per_city, c.customer_city from `target.customers` as c group by c.customer_city order by customer_count_per_city desc;
```

Row	customer_count	customer_state
1	41746	SP
2	12852	RJ
3	11635	MG
4	5466	RS

Results per page: 50 1 – 27 of 27

Row	customer_count_per_city	customer_city
1	15540	sao paulo
2	6882	rio de janeiro
3	2773	belo horizonte
4	2131	brasilia
5	1521	curitiba
6	1444	campinas

Results per page: 50 1 – 50 of 4119

- Customer from 4119 cities across 27 state

2. In-depth Exploration:
   a. **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?**

- Orders table

Query =>
```
select order_id, order_purchase_timestamp, payment_value from
(select o.order_id, o.order_purchase_timestamp, p.payment_value,
dense_rank() over(partition by extract(YEAR from
o.order_purchase_timestamp), extract(MONTH from
o.order_purchase_timestamp) ORDER BY p.payment_value desc) as rnk from
`target.orders` as o inner join `target.payments` as p on p.order_id =
o.order_id) as tmp where rnk = 1 order by order_purchase_timestamp;
```

Row	order_id	order_purchase_timestamp	payment_value
1	2e7a8482f6fb09756ca50c10d...	2016-09-04 21:15:19 UTC	136.23
2	bc0e0c28cbe995798d3afb7c7...	2016-10-08 01:28:14 UTC	1423.55
3	bd50a7fe9fd97ea4b7663031a...	2016-12-23 23:16:47 UTC	19.62
4	586992f50ed97898737b07970...	2017-01-31 17:01:46 UTC	3016.01
5	0812eb902a67711a1cb742b3c...	2017-02-12 20:37:36 UTC	6929.31
6	86c4eab1571921a6a6e248ed3...	2017-03-18 20:08:04 UTC	4016.91
12	03caa2c082116e1d31e67e9ae...	2017-09-29 15:24:52 UTC	13664.08

- Each month's highest sales.
- 29 sept. 2017 highest sales was made
- Market is growing and hit peak in some month

Query =>
```
select FORMAT_DATE('%A', o.order_purchase_timestamp) AS
week_day, round(SUM(p.payment_value), 2) AS total_sale, count(*) as
orders_count from `target.orders` as o inner join `target.payments` as p
on o.order_id = p.order_id  group by 1 order by total_sale desc;
```

Row	week_day	total_sale	orders_count
1	Monday	2622457.97	16875
2	Tuesday	2560743.03	16695
3	Wednesday	2493114.66	16274
4	Thursday	2384544.22	15470
5	Friday	2307128.2	14768
6	Sunday	1872456.36	12425
7	Saturday	1768427.68	11379

- On saturday's lowest sales were made.
- Monday is the most preferred day people buy.

b. What time do Brazilian customers tend to buy

Query=>
```
select count(*) as number_of_order ,
 case
  when  ordered_time > 1 and  ordered_time <= 5 then 'Dawn'
  when ordered_time > 5 and  ordered_time <= 12 then 'Morning'
  when  ordered_time > 12 and  ordered_time <= 18 then 'Afternoon'
  else 'Night'
 end as order_time,
from
(select * , cast(format_date('%k', o.order_purchase_timestamp) as
INT64) as ordered_time from`target.orders` as o) as tmp group by
order_time order by number_of_order desc
```

Row	number_of_orde	order_time
1	38135	Afternoon
2	31895	Night
3	28235	Morning
4	1176	Dawn

- Most brazilian's people tent to buy in afternoon and night
- Very few people buy at dawn time.

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

a. Get month on month orders by states

Query =>
```
select distinct c.customer_state,extract(YEAR from
o.order_purchase_timestamp) as year,extract(MONTH from
o.order_purchase_timestamp) as month, count(*) over(partition by
extract(YEAR from o.order_purchase_timestamp), extract(MONTH from
o.order_purchase_timestamp), c.customer_state) as sales from
`target.orders` as o inner join `target.customers` as c on o.customer_id
= c.customer_id order by  year, month;
```

Row	customer_state	year	month	sales
1	RR	2016	9	1
2	SP	2016	9	2
3	RS	2016	9	1
4	AL	2016	10	2
5	PI	2016	10	1
6	CE	2016	10	8
7	RS	2016	10	24
8	GO	2016	10	9

Query =>
```
select * from (select *, dense_rank() over(partition by year, month order by sales desc) as rnk from (select distinct c.customer_state,extract(YEAR from o.order_purchase_timestamp) as year,extract(MONTH from o.order_purchase_timestamp) as month, count(*) over(partition by extract(YEAR from o.order_purchase_timestamp), extract(MONTH from o.order_purchase_timestamp), c.customer_state) as sales from `target.orders` as o inner join `target.customers` as c on o.customer_id = c.customer_id)) where rnk < 4 order by  year, month;
```

Row	customer_state	year	month	sales	rnk
1	SP	2016	9	2	1
2	RS	2016	9	1	2
3	RR	2016	9	1	2
4	SP	2016	10	113	1
5	RJ	2016	10	56	2
6	MG	2016	10	40	3
7	PR	2016	12	1	1
8	SP	2017	1	299	1
9	MG	2017	1	108	2
10	RJ	2017	1	97	3
11	SP	2017	2	654	1
12	MG	2017	2	259	2
13	RJ	2017	2	254	3
14	SP	2017	3	1010	1
15	RJ	2017	3	395	2
16	MG	2017	3	358	3

b. Distribution of customers across the states in Brazil

Query =>
```
select *, round(100 * number_of_orders / sum(number_of_orders) over (),2) as order_percetage from (
select count(*) as number_of_orders, c.customer_state from `target.orders` as o left join `target.customers` as c on o.customer_id = c.customer_id
group by c.customer_state)
order by number_of_orders desc;
```

Query =>

```
select *, round(100 * number_of_orders / sum(number_of_orders) over (),2)
as order_percetage from (
select count(*) as number_of_orders, c.customer_city from `target.orders`
as o left join `target.customers` as c on o.customer_id = c.customer_id
group by c.customer_city)
order by number_of_orders desc;
```

Row	number_of_orders	customer_state	order_percetage
1	41746	SP	41.98
2	12852	RJ	12.92
3	11635	MG	11.7
4	5466	RS	5.5
5	5045	PR	5.07
6	3637	SC	3.66
7	3380	BA	3.4
8	2140	DF	2.15
9	2033	ES	2.04
10	2020	GO	2.03

Row	number_of_orde	customer_city	order_percetage
1	15540	sao paulo	15.63
2	6882	rio de janeiro	6.92
3	2773	belo horizonte	2.79
4	2131	brasilia	2.14
5	1521	curitiba	1.53
6	1444	campinas	1.45
7	1379	porto alegre	1.39
8	1245	salvador	1.25
9	1189	guarulhos	1.2
10	938	sao bernardo do campo	0.94

- Understanding state wise trends add rank to sales made by each state in a month.
- São Paulo both state and city sales, contributes highest in total sales, almost 42% state wise and 16% city wise.
- Every month of year 2017 and 2018 top sales are happening in states =>
    a. São Paulo

b. Rio de janeiro
c. Minas Gerais

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

Query =>
```
select month_name , total_orders as order_2017, orders_2018, round((orders_2018 - total_orders) / total_orders * 100,2) as percent_increase from(select *, lead(total_orders,8) over(order by year asc, month asc) as orders_2018, from (select cast(format_date('%m', o.order_purchase_timestamp) as int64) as month, cast(format_date('%y', o.order_purchase_timestamp) as int64) as year,format_date('%h', o.order_purchase_timestamp) as month_name, count(*) as total_orders from `target.orders` as o group by 3,2 ,1 having year=17 and  month < 9 or year = 18 and month < 9)) where orders_2018 is not null order by 2,1;
```

Row	month_name	order_2017	orders_2018	percent_increas
1	Jan	800	7269	808.63
2	Feb	1780	6728	277.98
3	Apr	2404	6939	188.64
4	Mar	2682	7211	168.87
5	Jun	3245	6167	90.05
6	May	3700	6873	85.76
7	Jul	4026	6292	56.28
8	Aug	4331	6512	50.36

- 800 orders were received in Jan 2017 and that increased by 808.63% in Jan 2018.
- It shows an increment in orders from 2017 to 2018.
- Minimum 50% hike in orders from 2017 to 2018 in Aug.
- Overall business increased in 2018 compared to 2017 Jan to Aug.

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

Query =>
```
select c.customer_state, round(sum(p.payment_value ), 2) as total_order_price, round(avg(p.payment_value), 2) as mean_order_price from `target.orders` as o left join `target.customers` as c on o.customer_id = c.customer_id left join `target.payments` as p on o.order_id = p.order_id group by c.customer_state order by total_order_price desc , mean_order_price desc;
```

Row	customer_state	total_order_price	mean_order_price
1	SP	5998226.96	137.5
2	RJ	2144379.69	158.53
3	MG	1872257.26	154.71
4	RS	890898.54	157.18
5	PR	811156.38	154.15
6	SC	623086.43	165.98
7	BA	616645.82	170.82
8	DF	355141.08	161.13
9	GO	350092.31	165.76
10	ES	325967.55	154.71
26	AP	16262.8	232.33
27	RR	10064.62	218.8

- States with the highest amount paid of orders have low mean price value.
- States with the lowest amount paid of orders have high mean price value.

Query =>
```
select tmp.customer_state, round(sum(freight_value)) as sum_of_freight, round(avg(freight_value)) as mean_of_freight from (select od.freight_value, c.customer_state from `target.order_items` as od inner join `target.orders` as o on od.order_id = o.order_id inner join `target.customers` as c on o.customer_id = c.customer_id ) as tmp group by tmp.customer_state order by mean_of_freight asc;
```

Row	customer_state	sum_of_freight	mean_of_freight
1	SP	718723.0	15.0
2	MG	270853.0	21.0
3	RJ	305589.0	21.0
4	DF	50625.0	21.0
5	PR	117852.0	21.0

Row	customer_state	sum_of_freight	mean_of_freight
1	PB	25720.0	43.0
2	RR	2235.0	43.0
3	RO	11417.0	41.0
4	AC	3687.0	40.0
5	PI	21218.0	39.0

- States with the highest freight paid of orders have low mean freight value.
- States like RR , PB paid the highest freight of orders.

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

Query =>
```
select order_id, timestamp_diff(order_delivered_customer_date, order_purchase_timestamp, day) as days_bwtween_purchase_delivery, timestamp_diff(order_estimated_delivery_date, order_purchase_timestamp, day) as days_bwtween_purchase_estimated_delivery from `target.orders` where order_status in ('delivered');
```

Row	order_id	days_bwtween_purchase_delivery	days_bwtween_purchase_estimated_delivery
1	c158e9806f85a33877bdfd4f6...	23	33
2	b60b53ad0bb7dacacf2989fe2...	12	7
3	c830f223aae08493ebecb52f2...	12	25
4	a8aa2cd070eeac7e4368cae3...	7	8
5	813c55ce9b6baa8f879e064fb...	12	21

   b. **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, datetime_diff(order_purchase_timestamp, order_delivered_customer_date, 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_status in ('delivered');
```

Row	order_id	time_to_delivery	diff_estimated_delivery
1	635c894d068ac37e6e03dc54e...	-30	1
2	3b97562c3aee8bdedcb5c2e45...	-32	0
3	68f47f50f04c4cb6774570cfde...	-29	1
4	276e9ec344d3bf029ff83a161c...	-43	-4
5	54e1a3c2b97fb0809da548a59...	-40	-4
6	fd04fa4105ee8045f6a0139ca5...	-37	-1

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

Query =>
```
select tmp.customer_state, round(avg(freight_value)) as mean_of_freight, round(avg(time_to_delivery)) as mean_time_to_delivery, round(avg(diff_estimated_delivery)) as mean_diff_estimated_delivery, from (select o.*,od.*, c.* ,datetime_diff(order_purchase_timestamp, order_delivered_customer_date, day) as time_to_delivery, datetime_diff(order_estimated_delivery_date, order_delivered_customer_date, day) as diff_estimated_delivery from `target.order_items` as od inner join `target.orders` as o on od.order_id = o.order_id inner join `target.customers` as c on o.customer_id = c.customer_id ) as tmp group by tmp.customer_state;
```

Row	customer_state	mean_of_freight	mean_time_to_delivery	mean_diff_estimated_delivery
1	MT	28.0	-18.0	14.0
2	MA	38.0	-21.0	9.0
3	AL	36.0	-24.0	8.0
4	SP	15.0	-8.0	10.0
5	MG	21.0	-12.0	12.0

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

Query =>
```
select tmp.customer_state, round(avg(freight_value)) as mean_of_freight, round(avg(time_to_delivery)) as mean_time_to_delivery, round(avg(diff_estimated_delivery)) as mean_diff_estimated_delivery, from (select o.*,od.*, c.* ,datetime_diff(order_purchase_timestamp, order_delivered_customer_date, day) as time_to_delivery, datetime_diff(order_estimated_delivery_date, order_delivered_customer_date, day) as diff_estimated_delivery from `target.order_items` as od inner join `target.orders` as o on od.order_id = o.order_id inner join `target.customers` as c on o.customer_id = c.customer_id ) as tmp group by tmp.customer_state order by mean_of_freight asc/desc;
```

Row	customer_state	mean_of_freight	mean_time_to_delivery	mean_diff_estimated_delivery
1	SP	15.0	-8.0	10.0
2	RJ	21.0	-15.0	11.0
3	PR	21.0	-11.0	13.0
4	SC	21.0	-15.0	11.0
5	DF	21.0	-13.0	11.0

Row	customer_state	mean_of_freight	mean_time_to_delivery	mean_diff_estimated_delivery
1	PB	43.0	-20.0	12.0
2	RR	43.0	-28.0	17.0
3	RO	41.0	-19.0	19.0
4	AC	40.0	-20.0	20.0
5	PI	39.0	-19.0	11.0

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

Query =>

```
select tmp.customer_state, round(avg(freight_value)) as mean_of_freight, round(avg(time_to_delivery)) as mean_time_to_delivery, round(avg(diff_estimated_delivery)) as mean_diff_estimated_delivery, from (select o.*,od.*, c.* ,datetime_diff(order_purchase_timestamp, order_delivered_customer_date, day) as time_to_delivery, datetime_diff(order_estimated_delivery_date, order_delivered_customer_date, day) as diff_estimated_delivery from `target.order_items` as od inner join `target.orders` as o on od.order_id = o.order_id inner join `target.customers` as c on o.customer_id = c.customer_id ) as tmp group by tmp.customer_state order by mean_time_to_delivery desc/asc;
```

Row	customer_state	mean_of_freight	mean_time_to_delivery	mean_diff_estimated_delivery
1	SP	15.0	-8.0	10.0
2	PR	21.0	-11.0	13.0
3	MG	21.0	-12.0	12.0
4	DF	21.0	-13.0	11.0
5	RJ	21.0	-15.0	11.0

Row	customer_state	mean_of_freight	mean_time_to_delivery	mean_diff_estimated_delivery
1	RR	43.0	-28.0	17.0
2	AP	34.0	-28.0	17.0
3	AM	33.0	-26.0	19.0
4	AL	36.0	-24.0	8.0
5	PA	36.0	-23.0	13.0

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

Query =>
```
select tmp.customer_state, round(avg(freight_value)) as mean_of_freight, round(avg(time_to_delivery)) as mean_time_to_delivery, round(avg(diff_estimated_delivery)) as mean_diff_estimated_delivery, from (select o.*,od.*, c.* ,datetime_diff(order_purchase_timestamp, order_delivered_customer_date, day) as time_to_delivery, datetime_diff(order_estimated_delivery_date, order_delivered_customer_date, day) as diff_estimated_delivery from `target.order_items` as od inner join `target.orders` as o on od.order_id = o.order_id inner join `target.customers` as c on o.customer_id = c.customer_id ) as tmp group by tmp.customer_state order by mean_diff_estimated_delivery desc/asc;
```

Row	customer_state	mean_of_freight	mean_time_to_delivery	mean_diff_estimated_delivery
1	AC	40.0	-20.0	20.0
2	AM	33.0	-26.0	19.0
3	RO	41.0	-19.0	19.0
4	RR	43.0	-28.0	17.0
5	AP	34.0	-28.0	17.0

Row	customer_state	mean_of_freight	mean_time_to_delivery	mean_diff_estimated_delivery
1	AL	36.0	-24.0	8.0
2	MA	38.0	-21.0	9.0
3	SE	37.0	-21.0	9.0
4	SP	15.0	-8.0	10.0
5	BA	26.0	-19.0	10.0

6. Payment type analysis:
   a. **Month over Month count of orders for different payment types**

Query =>
```
select payment_type, number_of_tran, round(100 * number_of_tran / sum(number_of_tran) over(), 3) as total_tran_percentage from (select payment_type, count(*) as number_of_tran from `target.payments` group by payment_type) order by number_of_tran desc;
```

Row	payment_type	number_of_tran	total_tran_percentage
1	credit_card	76795	73.922
2	UPI	19784	19.044
3	voucher	5775	5.559
4	debit_card	1529	1.472
5	not_defined	3	0.003

Query =>
```
select distinct p.payment_type ,extract(YEAR from o.order_purchase_timestamp) as year,extract(MONTH from o.order_purchase_timestamp) as month, count(*) over(partition by extract(YEAR from o.order_purchase_timestamp), extract(MONTH from o.order_purchase_timestamp), p.payment_type) as order_per_month from `target.orders` as o inner join `target.customers` as c on o.customer_id = c.customer_id  inner join `target.payments` as p on o.order_id = p.order_id order by year, month;
```

Row	payment_type	year	month	order_per_month
1	credit_card	2016	9	3
2	credit_card	2016	10	254
3	UPI	2016	10	63
4	voucher	2016	10	23
5	debit_card	2016	10	2
6	credit_card	2016	12	1
7	credit_card	2017	1	583
8	UPI	2017	1	197
9	voucher	2017	1	61
10	debit_card	2017	1	9

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

Query =>
```
select *, round(100 * no_of_orders / sum(no_of_orders) over(), 2) as percent_orders from (select payment_installments, count(*) as no_of_orders from `target.payments` group by payment_installments) order by no_of_orders desc;
```

Row	payment_installments	no_of_orders	percent_orders
1	1	52546	50.58
2	2	12413	11.95
3	3	10461	10.07
4	4	7098	6.83
5	10	5328	5.13
6	5	5239	5.04
7	8	4268	4.11
8	6	3920	3.77
9	7	1626	1.57
10	9	644	0.62

7. Actionable Insights
    - Orders table have data between 04/09/16 and 17/10/18
    - Total 329 orders was place in 2016, 45101 in 2017 and 54011 in 2018
    - Customer from 4119 cities across 27 state
    - 29 sept. 2017 highest sales was made
    - Market is growing and hit peak in some month
    - On saturday's lowest sales were made.
    - Monday is the most preferred day people buy.
    - Most brazilian's people tent to buy in afternoon and night
    - Very few people buy at dawn time.
    - Understanding state wise trends add rank to sales made by each state in a month.
    - São Paulo both state and city sales, contributes highest in total sales, almost 42% state wise and 16% city wise.
    - Every month of year 2017 and 2018 top sales are happening in states =>
        - São Paulo
        - Rio de janeiro
        - Minas Gerais

- 800 orders were received in Jan 2017 and that increased by 808.63% in Jan 2018.
- It shows an increment in orders from 2017 to 2018.
- Minimum 50% hike in orders from 2017 to 2018 in Aug.
- Overall business increased in 2018 compared to 2017 Jan to Aug.
- States with the highest amount paid of orders have low mean price value.
- States with the lowest amount paid of orders have high mean price value.
- States with the highest freight paid of orders have low mean freight value.
- States like RR , PB paid the highest freight of orders
- Top 5 states with lowest average time to delivery
    - São Paulo (SA)
    - Parana (PR)
    - Minas Gerais (MG)
    - DF
    - RJ
- Top 5 states with highest average time to delivery
    - Roraima (RR)
    - Ampa (AP)
    - AM
    - AL
    - PA
- Top 5 states where delivery is really fast compared to estimated date
    - AC
    - AM
    - RO
    - RR
    - AP

- Top 5 states where delivery is not so fast compared to estimated date
    - AL
    - MA
    - SE
    - SP
    - BA
- Almost 74% payments are made using Credit Card
- More credit card use can be seen every month wise orders
- Almost 50% orders pay in single installments
- People also prefer EMI option till 10 installments

8. Recommendations
    - 14 states and 3800 cities of brazil with then 1000 customers can be targeted for advertisement and offers to attract more customers
    - People in brazil use almost 73% time credit card for shopping this can be use attract more customers with credit card offers
    - Brazil's online shopping market is expected to grow more in future as it shows a minimum 50% increment in orders from 2017 to 2018.
    - People in Brazil buy online most in the afternoon and night. So best time for social media promotions
    - As orders increase in states with highest fright. Fright value will decrease. We have seen this in states with most orders.