

TARGET BUSINESS CASE

1. Import the dataset and do usual exploratory analysis

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

```
SELECT column_name, data_type
FROM `target-bc-400714.Target.INFORMATION_SCHEMA.COLUMNS`
WHERE table_name = 'customers'
ORDER BY ordinal_position
```

JOB INFORMATION		RESULTS	JSON	EXECUTION DET
Row	column_name	data_type		
1	customer_id	STRING		
2	customer_unique_id	STRING		
3	customer_zip_code_prefix	INT64		
4	customer_city	STRING		
5	customer_state	STRING		

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

```
SELECT
MIN(order_purchase_timestamp) as first_order,
MAX(order_purchase_timestamp) as last_order,
FROM `Target.orders`
```

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

C) Count the Cities & States of customers who ordered during the given period.

```
SELECT count(DISTINCT customer_city) as citycount, count(DISTINCT customer_state) as
statecount FROM `Target.orders` o LEFT JOIN `Target.customers` c ON o.customer_id =
c.customer_id
```

Query results

JOB INFORMATION		RESULTS	JSOI
Row	citycount	statecount	
1	4119	27	

Insights:

- Customers are from 4119 cities and 27 states.
- To understand the time period covered, we determined the start and end date of the sales i.e, from 4th September 2016 to 17th October 2018.

2. In-depth Exploration:

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

```
WITH orderscount AS(  
SELECT  
  count(*) as ordersplaced,  
  EXTRACT(MONTH from order_purchase_timestamp) as Month,  
  EXTRACT(YEAR from order_purchase_timestamp) as year  
FROM `target-bc-400714.Target.orders`  
GROUP BY EXTRACT(MONTH from order_purchase_timestamp),EXTRACT(YEAR from  
order_purchase_timestamp)  
)
```

```
SELECT ordersplaced,Month,Year from orderscount  
ORDER BY Year,Month
```

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

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

```
WITH orderscount AS(  
SELECT  
  count(*) as ordersplaced,  
  EXTRACT(MONTH from order_purchase_timestamp) as Month  
FROM `target-bc-400714.Target.orders`  
GROUP BY EXTRACT(MONTH from order_purchase_timestamp)  
)  
  
SELECT ordersplaced,Month from orderscount ORDER BY ordersplaced DESC
```

Row	ordersplaced	Month
1	10843	8
2	10573	5
3	10318	7
4	9893	3
5	9412	6
6	9343	4
7	8508	2
8	8069	1
9	7544	11
10	5674	12

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

- i. 0-6 hrs : Dawn
- ii. 7-12 hrs : Mornings
- iii. 13-18 hrs : Afternoon
- iv. 19-23 hrs : Night

```

WITH orderscount AS(
SELECT
CASE WHEN EXTRACT(HOUR FROM order_purchase_timestamp) BETWEEN 0 and 6 THEN
'Dawn'
      WHEN EXTRACT(HOUR FROM order_purchase_timestamp) BETWEEN 7 and 12 THEN
'Mornings'
      WHEN EXTRACT(HOUR FROM order_purchase_timestamp) BETWEEN 13 and 18 THEN
'Afternoon'
      WHEN EXTRACT(HOUR FROM order_purchase_timestamp) BETWEEN 19 and 23 THEN
'Night'
      END as timeperiod
FROM `target-bc-400714.Target.orders`
)

```

```

SELECT Count(*) as count,timeperiod from orderscount
group by timeperiod
Order by count desc

```

JOB INFORMATION		RESULTS	JSON	EXECU
Row	count ▼	timeperiod ▼		
1	38135	Afternoon		
2	28331	Night		
3	27733	Mornings		
4	5242	Dawn		

Insights:

- Based on the orders count, it's observed that there's a growing trend.
- The count of orders is increasing from March to August.
- In the month of August shows a peak in order's count.
- The data indicates that more orders are placed during daytime, specifically in the afternoon. Dawn is the least preferred shopping timing according to the data.

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

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

```
WITH orderscount AS(  
SELECT  
    count(*) as count,  
    EXTRACT(MONTH from order_purchase_timestamp) as Month,  
    customer_state  
FROM `target-bc-400714.Target.orders` o  
LEFT JOIN `Target.customers` c  
ON o.customer_id = c.customer_id  
GROUP BY EXTRACT(MONTH from order_purchase_timestamp),customer_state  
)
```

```
SELECT customer_state, month, count as ordersplaced from orderscount  
Order by Month, customer_state
```

Row	customer_state	month	ordersplaced
1	AC	1	8
2	AL	1	39
3	AM	1	12
4	AP	1	11
5	BA	1	264
6	CE	1	99
7	DF	1	151
8	ES	1	159
9	GO	1	164
10	MA	1	66

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

```
SELECT  
    customer_state, count(*) as customercount  
FROM `Target.customers`  
GROUP BY customer_state  
ORDER by customercount desc, customer_state
```

Row	customer_state	customercount
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

Insights:

- Data reveals that in the month of August has more orders followed by May and July. Customers from SP placed more orders in August.

4. Impact on Economy:

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

You can use the "payment_value" column in the payments table to get the cost of orders.

```
WITH getpercent2017 AS(
  SELECT SUM(payment_value) as payment_value,
  EXTRACT(MONTH FROM o.order_purchase_timestamp) as Month from `Target.payments` p
  LEFT JOIN `Target.orders` o
  ON p.order_id = o.order_id
  WHERE EXTRACT(YEAR FROM o.order_purchase_timestamp) = 2017
  AND EXTRACT(MONTH FROM o.order_purchase_timestamp) BETWEEN 1 and 8
  group by Month
),getpercent2018 AS(
```

```
  SELECT SUM(payment_value) as payment_value,
  EXTRACT(MONTH FROM o.order_purchase_timestamp) as Month from `Target.payments` p
  LEFT JOIN `Target.orders` o
  ON p.order_id = o.order_id
  WHERE EXTRACT(YEAR FROM o.order_purchase_timestamp) = 2018
```

```

AND EXTRACT(MONTH FROM o.order_purchase_timestamp) BETWEEN 1 and 8
group by Month
)

```

```

select ROUND(((b.payment_value - a.payment_value)/a.payment_value)* 100,2) as percent,
a.Month from getpercent2017 a
JOIN getpercent2018 b
ON a.Month = b.Month
order by a.Month

```

Row	percent	Month
1	705.13	1
2	239.99	2
3	157.78	3
4	177.84	4
5	94.63	5
6	100.26	6
7	80.04	7
8	51.61	8

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

```

SELECT customer_state,ROUND(SUM(price),2) as Total,ROUND(AVG(price),2) as Average from
`Target.orders` o
LEFT JOIN `Target.order_items` i ON o.order_id = i.order_id
LEFT JOIN `Target.customers` c ON o.customer_id = c.customer_id
group by customer_state
order by customer_state

```

Row	customer_state ▼	Total ▼	Average ▼
1	AC	15982.95	173.73
2	AL	80314.81	180.89
3	AM	22356.84	135.5
4	AP	13474.3	164.32
5	BA	511349.99	134.6
6	CE	227254.71	153.76
7	DF	302603.94	125.77
8	ES	275037.31	121.91
9	GO	294591.95	126.27
10	MA	119648.22	145.2

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

```
SELECT customer_state,ROUND(SUM(freight_value),2) as Total,ROUND(AVG(freight_value),2)
as Average from `Target.orders` o
LEFT JOIN `Target.order_items` i ON o.order_id = i.order_id
LEFT JOIN `Target.customers` c ON o.customer_id = c.customer_id
group by customer_state
order by customer_state
```

Row	customer_state ▼	Total ▼	Average ▼
1	AC	3686.75	40.07
2	AL	15914.59	35.84
3	AM	5478.89	33.21
4	AP	2788.5	34.01
5	BA	100156.68	26.36
6	CE	48351.59	32.71
7	DF	50625.5	21.04
8	ES	49764.6	22.06
9	GO	53114.98	22.77
10	MA	31523.77	38.26

Insights:

- By analyzing the percentage increase for each month January shows the highest % of increase followed by February and April.
- Data reveals that SP has the highest Total order price and RR has the lowest total order price. But if we check the average order price then SP has the lowest.
- Based on the data analysis it reveals that RR has the highest average freight followed by PB and RO. Meanwhile SP has the lowest average freight followed by PR and MG.

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.

```
SELECT order_id, 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 `Target.orders`
ORDER BY order_id
```

Row	order_id	time_to_deliver	diff_estimated_delivery
1	00010242fe8c5a6d1ba2dd792...	7	8
2	00018f77f2f0320c557190d7a1...	16	2
3	000229ec398224ef6ca0657da...	7	13
4	00024acbcd0a6daa1e931b03...	6	5
5	00042b26cf59d7ce69dfabb4e...	25	15
6	00048cc3ae777c65dbb7d2a06...	6	14
7	00054e8431b9d7675808bcb8...	8	16
8	000576fe39319847cbb9d288c...	5	15
9	0005a1a1728c9d785b8e2b08...	9	0
10	0005f50442cb953dcd1d21e1f...	2	18

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

```
WITH highestState AS (  
  SELECT  
    customer_state,  
    ROUND(AVG(freight_value), 2) AS highAverage,  
    ROW_NUMBER() OVER (ORDER BY AVG(freight_value) DESC) AS row_num  
  FROM `Target.orders` o  
  LEFT JOIN `Target.order_items` i ON o.order_id = i.order_id  
  LEFT JOIN `Target.customers` c ON o.customer_id = c.customer_id  
  GROUP BY customer_state  
  ORDER BY highAverage DESC  
  LIMIT 5  
,  
lowestState AS (  
  SELECT  
    customer_state,  
    ROUND(AVG(freight_value), 2) AS lowAverage,  
    ROW_NUMBER() OVER (ORDER BY AVG(freight_value)) AS row_num  
  FROM `Target.orders` o  
  LEFT JOIN `Target.order_items` i ON o.order_id = i.order_id  
  LEFT JOIN `Target.customers` c ON o.customer_id = c.customer_id  
  GROUP BY customer_state  
  ORDER BY lowAverage  
  LIMIT 5  
)  
  
SELECT  
  high.customer_state AS top_5_highest_state,  
  low.customer_state AS top_5_lowest_state  
FROM highestState high  
JOIN lowestState low  
ON high.row_num = low.row_num;
```

Row	top_5_highest_state	top_5_lowest_state
1	RR	SP
2	PB	PR
3	RO	MG
4	AC	RJ
5	PI	DF

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

```

WITH highestState AS (
  SELECT
    customer_state,
    ROUND(AVG(date_diff( order_delivered_customer_date, order_purchase_timestamp,day))),
  2) AS highAverage,
    ROW_NUMBER() OVER (ORDER BY AVG(date_diff( order_delivered_customer_date,
order_purchase_timestamp,day)) DESC) AS row_num
  FROM `Target.orders` o
  LEFT JOIN `Target.order_items` i ON o.order_id = i.order_id
  LEFT JOIN `Target.customers` c ON o.customer_id = c.customer_id
  GROUP BY customer_state
  ORDER BY highAverage DESC
  LIMIT 5
),
lowestState AS (
  SELECT
    customer_state,
    ROUND(AVG(date_diff( order_delivered_customer_date, order_purchase_timestamp,day))),
  2) AS lowAverage,
    ROW_NUMBER() OVER (ORDER BY AVG(date_diff( order_delivered_customer_date,
order_purchase_timestamp,day))) AS row_num
  FROM `Target.orders` o
  LEFT JOIN `Target.order_items` i ON o.order_id = i.order_id
  LEFT JOIN `Target.customers` c ON o.customer_id = c.customer_id
  GROUP BY customer_state
  ORDER BY lowAverage
  LIMIT 5
)

```

```

SELECT
  high.customer_state AS top_5_highest_state,
  low.customer_state AS top_5_lowest_state
FROM highestState high
JOIN lowestState low
ON high.row_num = low.row_num;

```

Row	top_5_highest_state	top_5_lowest_state
1	RR	SP
2	AP	PR
3	AM	MG
4	AL	DF
5	PA	SC

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

```

WITH top5state AS (
  SELECT
    customer_state,
    ROUND(AVG(date_diff( order_delivered_customer_date,
order_estimated_delivery_date,day)), 2) AS fastestDelivery,
  FROM `Target.orders` o
  LEFT JOIN `Target.customers` c ON o.customer_id = c.customer_id
  GROUP BY customer_state
)

```

```

SELECT customer_state
FROM top5state
order by fastestDelivery
LIMIT 5

```

Row	customer_state
1	AC
2	RO
3	AP
4	AM
5	RR

Insights:

- From the given data it reveals that SP has the lowest average freight value, while RR has the highest average freight value.
- Same as average freight value, in average time to delivery also SP has the lowest average time to delivery, and RR has the highest average time to delivery.
- AC,RO,AP,AM and RR are the top 5 fastest delivery states.

6. Analysis based on the payments

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

```
SELECT count(DISTINCT o.order_id) as ordercount,  
EXTRACT(MONTH from order_purchase_timestamp) as Month,payment_type  
FROM `target-bc-400714.Target.orders` o  
INNER JOIN `Target.payments` p  
ON o.order_id = p.order_id  
GROUP BY Month,payment_type  
ORDER BY payment_type,Month,ordercount
```

Row	ordercount	Month	payment_type
1	1715	1	UPI
2	1723	2	UPI
3	1942	3	UPI
4	1783	4	UPI
5	2035	5	UPI
6	1807	6	UPI
7	2074	7	UPI
8	2077	8	UPI
9	903	9	UPI
10	1056	10	UPI

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

```
SELECT p.payment_installments,count(DISTINCT o.order_id) as ordercount
```

```
FROM `target-bc-400714.Target.orders` o
INNER JOIN `Target.payments` p
ON o.order_id = p.order_id
where o.order_status != 'canceled'
group by payment_installments
ORDER BY payment_installments
```

Row	payment_installment	ordercount
1	0	2
2	1	48732
3	2	12329
4	3	10374
5	4	7046
6	5	5204
7	6	3894
8	7	1617
9	8	4224
10	9	638

Insights

- Based on the given data it reveals that credit card transactions are the most preferred transaction in Brazil followed by UPI.
- Most preferred payment installments are single payment installment, followed by two and three month installments.

Recommendations

- SP state has more number of orders compared to other states, by increasing the Average Freight Value and Average Delivery time can help to increase the revenue.
- In the month of November and December sales are decreasing compared to other months, promoting more sale offers and other promotions can increase the sales in these months.

- Promoting more social media advertisements and running other campaigns at night time can increase the sales in night time compared to the daytime.