# TARGET CASE STUDY

## Assumptions made:

1. The orders are counted on the order_id irrespective of the status of delivery,
2. Used custom bins for categorizing the time as dawn, morning, afternoon and night  and also Brazil local time is also not taken into account,
3. The customer_delivery_date is the default delivery_date.

**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

```
select
      column_name,
      data_type
from `target-383507.target`.INFORMATION_SCHEMA.COLUMNS
where table_name = 'customers';
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTIO

| 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 |

2. Time period for which the data is given

```
select  distinct extract(year from order_purchase_timestamp) as time_period
from `target-383507.target.orders`
order by 1 desc;
```

Query results

SAVE RESULTS

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

| Row | time_period |
|---|---|
| 1 | 2018 |
| 2 | 2017 |
| 3 | 2016 |

3. Cities and States of customers ordered during the given period

```
select
  extract(year from o.order_purchase_timestamp) as time_period,
  c.customer_city,
  c.customer_state
from `target-383507.target.customers` c
inner join `target-383507.target.orders` o
on c.customer_id = o.customer_id
order by 1 desc;
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECU

| Row | time_period | customer_city | customer_state |
|---|---|---|---|
| 1 | 2018 | acu | RN |
| 2 | 2018 | ico | CE |
| 3 | 2018 | ico | CE |
| 4 | 2018 | ico | CE |
| 5 | 2018 | ico | CE |
| 6 | 2018 | ipe | RS |
| 7 | 2018 | ipe | RS |
| 8 | 2018 | ipu | CE |
| 9 | 2018 | ipu | CE |
| 10 | 2018 | itu | SP |
| 11 | 2018 | itu | SP |

**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?

```
select
        extract(year from order_purchase_timestamp) AS Year,
        extract(month from order_purchase_timestamp) AS Month,
        count(order_id) as orders_count,
from `target-383507.target.orders`
group by Year, Month
order by Year, Month;
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTIO

| Row | Year | Month | orders_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 |
| 11 | 2017 | 8 | 4331 |

Results

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

```
select
  case
    when extract(hour from order_purchase_timestamp) between 3 and 6 then 'Daw
n'
    when extract(hour from order_purchase_timestamp) between 7 and 12 then 'Mor
ning'
    when extract(hour from order_purchase_timestamp) between 13 and 18 then 'Aft
ernoon'
  else 'Night'
  end as Purchase_time,
  count(customer_id) as customer_count
from `target-383507.target.orders`
group by Purchase_time
order by Purchase_time;
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

| Row | Purchase_time | customer_count |
|---|---|---|
| 1 | Afternoon | 38135 |
| 2 | Dawn | 1168 |
| 3 | Morning | 27733 |
| 4 | Night | 32405 |

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

1. Get month on month orders by states

```
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 orders
from `target-383507.target.customers` c
inner join `target-383507.target.orders` o
on c.customer_id = o.customer_id
group by 1,2,3
order by 1,2,3;
```

Query results

SAVE RESULTS

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

| Row | customer_state | year | month | orders |
|---|---|---|---|---|
| 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 |
| 11 | AC | 2017 | 11 | 5 |

2. Distribution of customers across the states in Brazil

```
select
    customer_state,
    customer_city,
    count(customer_id) as no_of_customers
from `target-383507.target.customers`
group by 1,2
order by 1,3 desc;
```

Query results

SAVE RESULTS

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

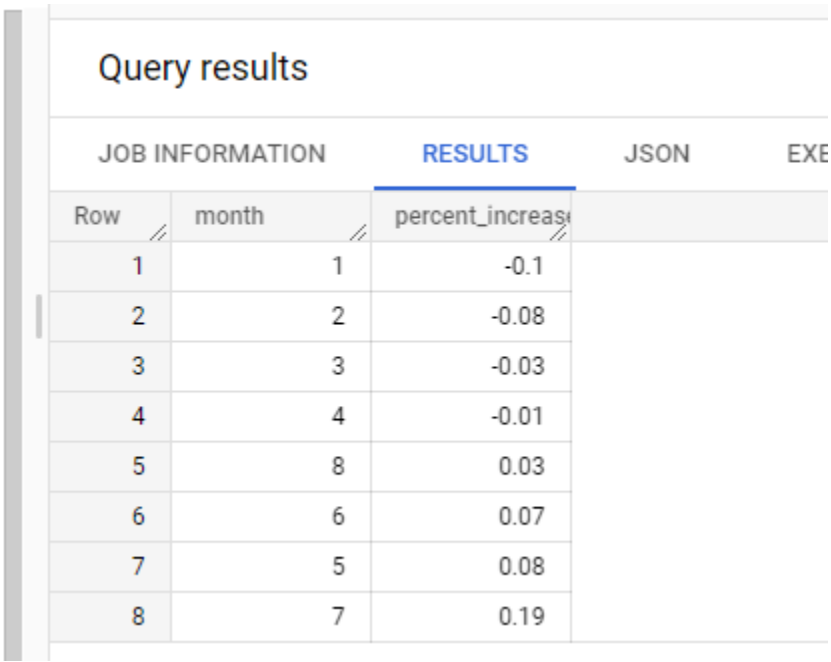
| Row | customer_state | customer_city | no_of_customers |
|---|---|---|---|
| 1 | AC | rio branco | 70 |
| 2 | AC | cruzeiro do sul | 3 |
| 3 | AC | xapuri | 2 |
| 4 | AC | senador guiomard | 2 |
| 5 | AC | brasileia | 1 |
| 6 | AC | porto acre | 1 |
| 7 | AC | manoel urbano | 1 |
| 8 | AC | epitaciolandia | 1 |
| 9 | AL | maceio | 247 |
| 10 | AL | arapiraca | 29 |
| 11 | AL | penedo | 8 |

Results per page: 50

**Q4 -** Impact on Economy: Analyze the money movement 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) - You can use “payment_value” column in payments table

```sql
select
  x.month,
  x.percent_increase
from (
select
  v.year,
  v.month,
  round((v.lead_payment-v.avg_payment_value)/v.avg_payment_value,2) as percent_increase
from (
select
  s.*,
  lead(s.avg_payment_value) over(partition by s.month order by s.year,s.month) as lead_payment
from (
select distinct
  t.year,
  t.month,
  avg(t.payment_value) over(partition by t.year,t.month order by t.year,t.month) as avg_payment
_value
from
(
select
  extract(year from o.order_purchase_timestamp) as year,
  extract(month from o.order_purchase_timestamp) as month,
  p.payment_value
from `target-383507.target.orders` o
inner join `target-383507.target.payments` p on o.order_id = p.order_id
)t
where t.month between 1 and 8
)s
)v
)x
where x.percent_increase is not null
order by 2,1;
```



Query results

JOB INFORMATION | RESULTS | JSON | EXE

| Row | month | percent_increas |
|---|---|---|
| 1 | 1 | -0.1 |
| 2 | 2 | -0.08 |
| 3 | 3 | -0.03 |
| 4 | 4 | -0.01 |
| 5 | 8 | 0.03 |
| 6 | 6 | 0.07 |
| 7 | 5 | 0.08 |
| 8 | 7 | 0.19 |

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

```
select
  c.customer_state,
  round(sum(oi.price),0) as sum_price,
  round(avg(oi.price),0) as mean_price,
  round(sum(oi.freight_value),0) as sum_freight_value,
  round(avg(oi.freight_value),0) as mean_freight_value
from `target-383507.target.order_items` oi
inner join `target-383507.target.orders` o on oi.order_id = o.order_id
inner join `target-383507.target.customers` c on o.customer_id = c.customer_id
group by 1
order by 1;
```

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

| Row | customer_state | sum_price | mean_price | sum_freight_valu | mean_freight_va |
|---|---|---|---|---|---|
| 1 | AC | 15983.0 | 174.0 | 3687.0 | 40.0 |
| 2 | AL | 80315.0 | 181.0 | 15915.0 | 36.0 |
| 3 | AM | 22357.0 | 135.0 | 5479.0 | 33.0 |
| 4 | AP | 13474.0 | 164.0 | 2789.0 | 34.0 |
| 5 | BA | 511350.0 | 135.0 | 100157.0 | 26.0 |
| 6 | CE | 227255.0 | 154.0 | 48352.0 | 33.0 |
| 7 | DF | 302604.0 | 126.0 | 50625.0 | 21.0 |
| 8 | ES | 275037.0 | 122.0 | 49765.0 | 22.0 |
| 9 | GO | 294592.0 | 126.0 | 53115.0 | 23.0 |
| 10 | MA | 119648.0 | 145.0 | 31524.0 | 38.0 |
| 11 | MG | 1585308.0 | 121.0 | 270853.0 | 21.0 |

**Q5 -** Analysis on sales, freight and delivery time

1. Calculate days between purchasing, delivering and estimated delivery

```
select
  order_id,
  date_diff(order_delivered_customer_date,order_purchase_timestamp,day)
as purchase_delivery_gap,
  date_diff(order_estimated_delivery_date,order_purchase_timestamp,day)
as purchase_estimation_gap,
```

```
 date_diff(order_estimated_delivery_date,order_delivered_customer_date,d
ay) as estimation_delivery_gap
from `target-383507.target.orders`
order by 2 desc;
```

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

| Row | order_id | purchase_delive | purchase_estim | estimation_deliy |
|---|---|---|---|---|
| 1 | ca07593549f1816d26a572e06... | 209 | 28 | -181 |
| 2 | 1b3190b2dfa9d789e1f14c05b... | 208 | 19 | -188 |
| 3 | 440d0d17af552815d15a9e41a... | 195 | 30 | -165 |
| 4 | 0f4519c5f1c541ddec9f21b3bd... | 194 | 32 | -161 |
| 5 | 285ab9426d6982034523a855f... | 194 | 28 | -166 |
| 6 | 2fb597c2f772eca01b1f5c561b... | 194 | 39 | -155 |
| 7 | 47b40429ed8cce3aee9199792... | 191 | 15 | -175 |
| 8 | 2fe324febf907e3ea3f2aa9650... | 189 | 22 | -167 |
| 9 | 2d7561026d542c8dbd8f0daea... | 188 | 28 | -159 |
| 10 | 437222e3fd1b07396f1d9ba8c... | 187 | 42 | -144 |
| 11 | c27815f7e3dd0b926b5855262... | 187 | 25 | -162 |

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

```
select
  order_id,
  date_diff(order_delivered_customer_date,order_purchase_timestamp,day)
 as time_to_delivery,
  date_diff(order_estimated_delivery_date,order_delivered_customer_date,day)
 as diff_estimated_delivery
from `target-383507.target.orders`
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUT

| Row | order_id | time_to_delivery | diff_estimated_c |
|---|---|---|---|
| 1 | 1950d777989f6a877539f5379... | 30 | -12 |
| 2 | 2c45c33d2f9cb8ff8b1c86cc28... | 30 | 28 |
| 3 | 65d1e226dfaeb8cdc42f66542... | 35 | 16 |
| 4 | 635c894d068ac37e6e03dc54e... | 30 | 1 |
| 5 | 3b97562c3aee8bdedcb5c2e45... | 32 | 0 |
| 6 | 68f47f50f04c4cb6774570cfde... | 29 | 1 |
| 7 | 276e9ec344d3bf029ff83a161c... | 43 | -4 |
| 8 | 54e1a3c2b97fb0809da548a59... | 40 | -4 |
| 9 | fd04fa4105ee8045f6a0139ca5... | 37 | -1 |
| 10 | 302bb8109d097a9fc6e9cefc5... | 33 | -5 |
| 11 | 66057d37308e787052a32828... | 38 | -6 |

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

```
select
  c.customer_state,
  round(avg(oi.freight_value),2) as avg_freight_value,
  ceil(avg(date_diff(o.order_delivered_customer_date,o.order_purchase_timestamp
,day))) as avg_time_to_delivery,
  ceil(avg(date_diff(o.order_estimated_delivery_date,o.order_delivered_customer_
date,day))) as avg_diff_estimated_delivery
from `target-383507.target.orders` o
inner join `target-383507.target.customers` c on o.customer_id = c.customer_id
inner join `target-383507.target.order_items` oi on o.order_id = oi.order_id
group by 1
order by 1;
```

Query results

SAVE RESULTS

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

| Row | customer_state | avg_freight_valu | avg_time_to_del | avg_diff_estimat |
|---|---|---|---|---|
| 1 | AC | 40.07 | 21.0 | 21.0 |
| 2 | AL | 35.84 | 24.0 | 8.0 |
| 3 | AM | 33.21 | 26.0 | 19.0 |
| 4 | AP | 34.01 | 28.0 | 18.0 |
| 5 | BA | 26.36 | 19.0 | 11.0 |
| 6 | CE | 32.71 | 21.0 | 11.0 |
| 7 | DF | 21.04 | 13.0 | 12.0 |
| 8 | ES | 22.06 | 16.0 | 10.0 |
| 9 | GO | 22.77 | 15.0 | 12.0 |
| 10 | MA | 38.26 | 22.0 | 10.0 |
| 11 | MG | 20.63 | 12.0 | 13.0 |

4. Sort the data to get the following:

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

```
select
  c.customer_state,
  round(avg(oi.freight_value),2) as highest_avg_freight_value,
from `target-383507.target.orders` o
inner join `target-383507.target.customers` c on o.customer_id = c.customer_id
inner join `target-383507.target.order_items` oi on o.order_id = oi.order_id
group by 1
order by 2 desc
limit 5;
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION D

| Row | customer_state | highest_avg_frei |
|---|---|---|
| 1 | RR | 42.98 |
| 2 | PB | 42.72 |
| 3 | RO | 41.07 |
| 4 | AC | 40.07 |
| 5 | PI | 39.15 |

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

```
select
  c.customer_state,
  ceil(avg(date_diff(o.order_delivered_customer_date,o.order_purchase_time
stamp,day))) as avg_time_to_delivery,
from `target-383507.target.orders` o
inner join `target-
383507.target.customers` c on o.customer_id = c.customer_id
group by 1
order by 2 desc
limit 5;
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION

| Row | customer_state | avg_time_to_del |
|---|---|---|
| 1 | RR | 29.0 |
| 2 | AP | 27.0 |
| 3 | AM | 26.0 |
| 4 | AL | 25.0 |
| 5 | PA | 24.0 |

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

```
select
  c.customer_state,
  ceil(avg(date_diff(o.order_estimated_delivery_date,o.order_delivered_customer_
date,day))) as fast_delivery
from `target-383507.target.orders` o
inner join `target-383507.target.customers` c on o.customer_id = c.customer_id
group by 1
order by 2 desc
limit 5;
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

| Row | customer_state | fast_delivery |
|---|---|---|
| 1 | AC | 20.0 |
| 2 | RO | 20.0 |
| 3 | AP | 19.0 |
| 4 | AM | 19.0 |
| 5 | RR | 17.0 |

6. Payment type analysis:

1. Month over Month count of orders for different payment types

```
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 orders
from `target-383507.target.orders` o
inner join `target-383507.target.payments` p  on  o.order_id = p.order_id
group by 1,2,3
order by 1,2,3;
```

Query results

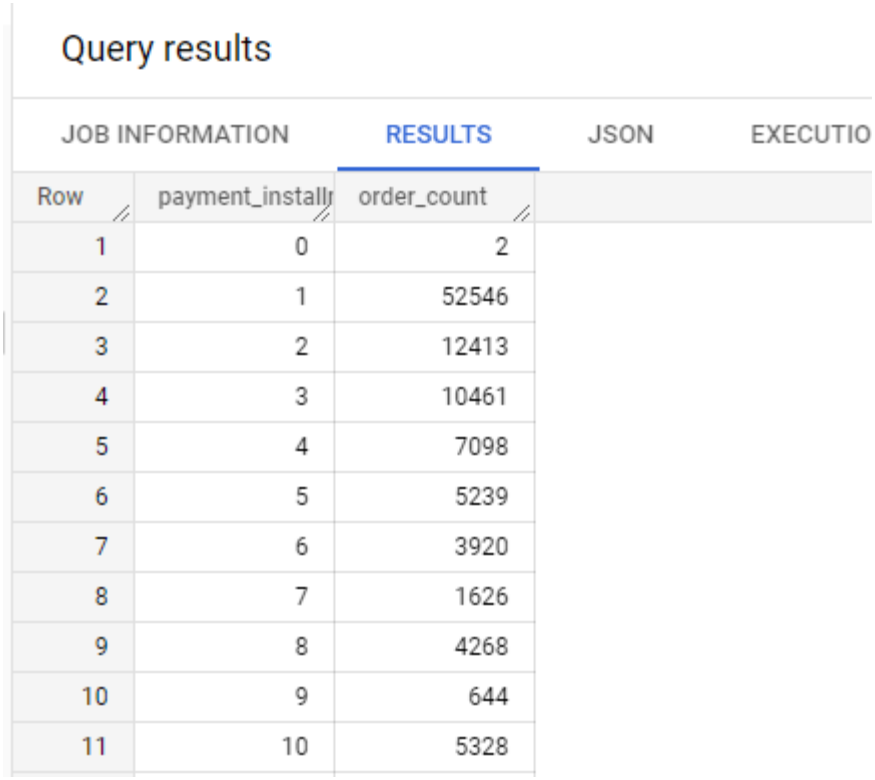
SAVE RE

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

| Row | payment_type | year | month | orders |
|---|---|---|---|---|
| 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 |
| 11 | UPI | 2017 | 10 | 993 |

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

```
select
  payment_installments,
  count(order_id) as order_count
from `target-383507.target.payments`
group by 1
order by 1;
```



Query results

JOB INFORMATION | RESULTS | JSON | EXECUTIO

| Row | payment_installr | order_count |
|---|---|---|
| 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 |
| 11 | 10 | 5328 |

# **Q7 -** Actionable Insights

- Not exactly, there is a growing trend in orders as it initially (2016) and then later (2018) we see a sharp dip in orders. Initially we see 2017 March. We get to see a peak in 2017 March. Also, we can find peaks during 2017 Nov and 2018 Jan, March. With given data we find peak in March repeated, indicating seasonality in the region.

- But when we do see rise in payment value between 2016 to 2017 only considering Jan to Aug month on month even with oscillating growth percentages.

- From the result obtained, it is found that the maximum customer distribution is from the state named “SP” in Brazil.

- We can see a sizeable number of undelivered orders while getting the delivery date differences.

- We see customers prefer credit card payments more than UPI and debit card.

- For those customers who pay by installments, we see high number for less than 10 installments and few for more than 10 installments.

- Customers prefer afternoons and nights to carry out purchases.

## **Q8 -** RECOMMENDATION

- For having limited time sales for goods that are less purchased, we can use the evening and night to attract more sales. Performing this on March month would be adding more success to this

- Since people prefer credit cards for payment mode, we can use cashbacks and offers for improving sales