TARGET BRAZIL SALES ANALYSIS

Introduction:

- Target is a globally renowned brand and a generalised merchandise retailer & headquartered in Minneapolis, Minnesota. It is the 7th largest retailer in U.S.
- This brand has a wide variety of food & general merchandise from clothing to household goods to electronics and toys.
- Target offers both online & in-store shopping.
- The particular business case focuses on the operations of Target in Brazil and provides insightful information about 1,00,000 orders placed between 2016 and 2018.

Dataset Information:

- The dataset has information about the order_status, price, payment and freight performance, customer location, product attributes and customer reviews.
- Dataset Link: https://drive.google.com/drive/folders/1TGEc66YKbD443nslRi1bWgVd238gJCnb
- The data is available in 8 csv files:

1. customers.csv
2. sellers.csv
3. order_items.csv
4. geolocation.csv
5. payments.csv
6. reviews.csv
7. orders.csv
8. products.csv

- **The column description for these CSV files is given below:**

  The **customers.csv** contain following features:

  - customer_id: Id of the consumer who made the purchase.
  - customer_unique_id: Unique Id of the consumer.
  - customer_zip_code_prefix: Zip Code of the location of the - consumer.
  - customer_city: Name of the City from where order is made.
  - customer_state: State Code from where order is made (Ex- Sao paulo-SP).

  The **sellers.csv** contains following features:

  - seller_id: Unique Id of the seller registered
  - seller_zip_code_prefix: Zip Code of the location of the seller.
  - seller_city: Name of the City of the seller.
  - seller_state: State Code (Ex- Sao paulo-SP)

  The **order_items.csv** contain following features:

  - order_id: A unique id of order made by the consumers.
  - order_item_id: A Unique id given to each item ordered in the order.
  - product_id: A unique id given to each product available on the site.
  - seller_id: Unique Id of the seller registered in Target.
  - shipping_limit_date: The date before which shipping of the ordered product must be completed.
  - price: Actual price of the products ordered.
  - freight_value: Price rate at which a product is delivered from one point to another.

  The **geolocations.csv** contain following features:

  - geolocation_zip_code_prefix: first 5 digits of zip code
  - geolocation_lat: latitude
  - geolocation_lng: longitude
  - geolocation_city: city name
  - geolocation_state: state

  The **payments.csv** contain following features:

  - order_id: A unique id of order made by the consumers.
  - payment_sequential: sequences of the payments made in case of EMI.
  - payment_type: mode of payment used. (Ex-Credit Card)
  - payment_installments: number of installments in case of EMI purchase.
  - payment_value: Total amount paid for the purchase order.

  The **orders.csv** contain following features:

  - order_id: A unique id of order made by the consumers.
  - customer_id: Id of the consumer who made the purchase.
  - order_status: status of the order made i.e., delivered, shipped etc.
  - order_purchase_timestamp: Timestamp of the purchase.
  - order_delivered_carrier_date: delivery date at which carrier made the delivery.
  - order_delivered_customer_date: date at which customer got the product.

- order_estimated_delivery_date: estimated delivery date of the products.

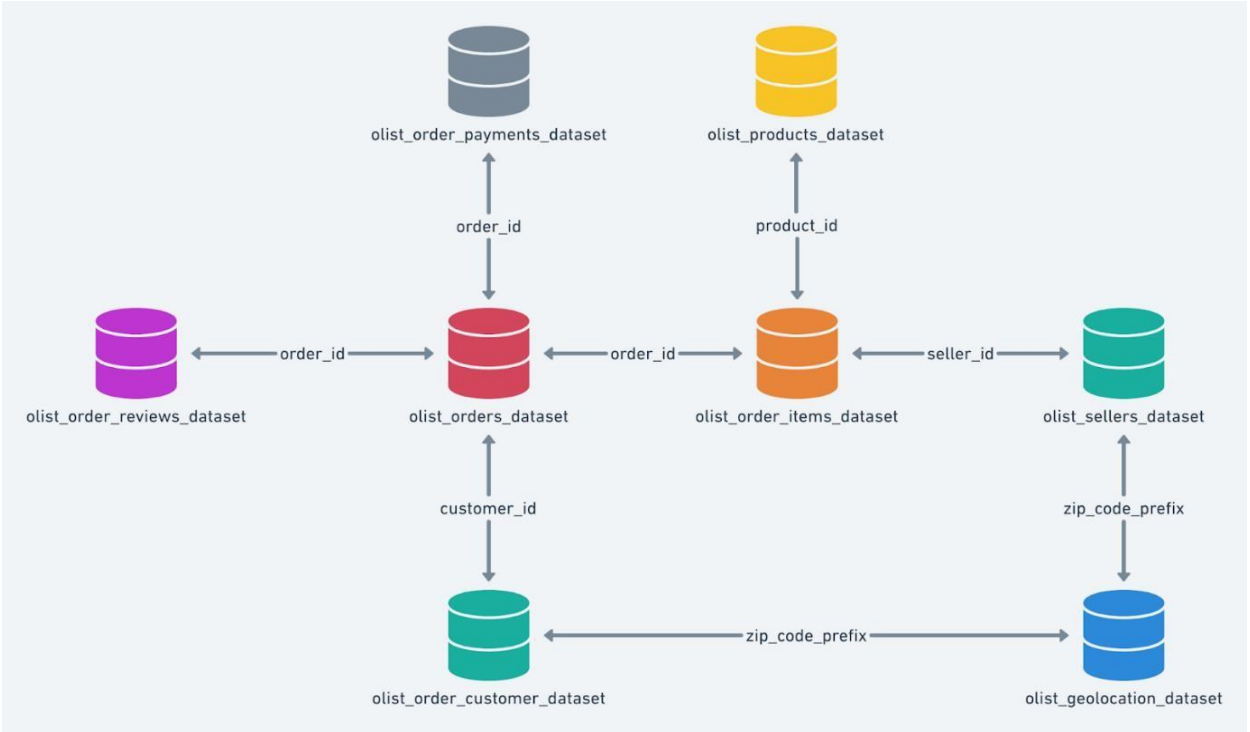
The **reviews.csv** contain following features:

- review_id: Id of the review given on the product ordered by the order id.
- order_id: A unique id of order made by the consumers.
- review_score: review score given by the customer for each order on the scale of 1–5.
- review_comment_title: Title of the review
- review_comment_message: Review comments posted by the consumer for each order.
- review_creation_date: Timestamp of the review when it is created.
- review_answer_timestamp: Timestamp of the review answered.

The **products.csv** contain following features:

- product_id: A unique identifier for the proposed project.
- product_category_name: Name of the product category
- product_name_length: length of the string which specifies the name given to the products ordered.
- product_description_length: length of the description written for each product ordered on the site.
- product_photos_qty: Number of photos of each product ordered available on the shopping portal.
- product_weight_g: Weight of the products ordered in grams.
- product_length_cm: Length of the products ordered in centimeters.
- product_height_cm: Height of the products ordered in centimeters.
- product_width_cm: width of the product ordered in centimeters.

Dataset Schema:



PRIMARY ANALYSIS: (EXTRA)

Customers:

Checking columns and data types:

```
SELECT
  *
FROM target-brazil-
sales.Target.INFORMATION_SCHEMA.COLUMNS WHERE table_name
= 'Customers';
```

table_catalog	table_schema	table_name	column_name	ordinal_position	is_nullable	data_type
target-brazil-sales	Target	Customers	customer_id	1	YES	STRING
target-brazil-sales	Target	Customers	customer_unique_id	2	YES	STRING
target-brazil-sales	Target	Customers	customer_zip_code_prefix	3	YES	INT64
target-brazil-sales	Target	Customers	customer_city	4	YES	STRING
target-brazil-sales	Target	Customers	customer_state	5	YES	STRING

Total number of customers:

```
SELECT COUNT(DISTINCT customer_id) AS customer_number FROM `Target.Customers`;
```

customer_number
99441

- There are 99441 customer_ids in the dataset.

Total number of unique customers:

```
SELECT COUNT(DISTINCT customer_unique_id) AS unique_customer_number FROM
`Target.Customers`;
```

unique_customer_number
96096

- There are 96096 unique customer_ids in the dataset.

Total number of customer locations:

```
SELECT COUNT(DISTINCT customer_zip_code_prefix) AS locations FROM
`Target.Customers`;
```

locations
14994

- There are 14994 locations in the dataset.

Total number of cities from which customers belong:

```
SELECT COUNT(DISTINCT customer_city) AS total_city FROM `Target.Customers`;
```

total_city
4119

- The customers are from 4119 cities.

```
# Total number of states from which customers belong:

SELECT COUNT(DISTINCT customer_state) AS total_state FROM `Target.Customers`;
```

total_state
27

- The customers are from 27 states.

Sellers:

```
# Checking columns and data types:

SELECT
  *
FROM target-brazil-
sales.Target.INFORMATION_SCHEMA.COLUMNS WHERE table_name
= 'Sellers';
```

table_catalog	table_schema	table_name	column_name	ordinal_position	is_nullable	data_type
target-brazil-sales	Target	Sellers	seller_id	1	YES	STRING
target-brazil-sales	Target	Sellers	seller_zip_code_prefix	2	YES	INT64
target-brazil-sales	Target	Sellers	seller_city	3	YES	STRING
target-brazil-sales	Target	Sellers	seller_state	4	YES	STRING

```
# Total number of sellers:

SELECT COUNT(DISTINCT seller_id) AS seller_number FROM `Target.Sellers`;
```

seller_number
3095

- There are 3095 sellers.

```
# Total number of seller locations:
SELECT COUNT(DISTINCT seller_zip_code_prefix) AS seller_location FROM
`Target.Sellers`;
```

seller_location
2246

- The sellers are from 2246 locations.

```
# Total number of cities sellers belong to:
```

```
SELECT COUNT(DISTINCT seller_city) AS total_city FROM `Target.Sellers`;
```

total_city
611

- The sellers are from 611 cities.

```
# Total number of states sellers belong to:
```

```
SELECT COUNT(DISTINCT seller_state) AS total_state FROM `Target.Sellers`;
```

total_state
23

- The sellers are from 23 states.

Products:

```
# Checking columns and data types:
```

```
SELECT
  *
FROM target-brazil-
sales.Target.INFORMATION_SCHEMA.COLUMNS WHERE table_name
= 'Products';
```

table_catalog	table_schema	table_name	column_name	ordinal_position	is_nullable	data_type
target-brazil-sales	Target	Products	product_id	1	YES	STRING
target-brazil-sales	Target	Products	product_category	2	YES	STRING
target-brazil-sales	Target	Products	product_name_length	3	YES	INT64
target-brazil-sales	Target	Products	product_description_length	4	YES	INT64
target-brazil-sales	Target	Products	product_photos_qty	5	YES	INT64
target-brazil-sales	Target	Products	product_weight_g	6	YES	INT64
target-brazil-sales	Target	Products	product_length_cm	7	YES	INT64

```
# Total number of distinct products:
```

```
SELECT COUNT(DISTINCT product_id) AS number_of_products FROM `Target.Products`;
```

number_of_products
32951

- There are 32951 different products.

```
# Total number of distinct product_categories:
```

```
SELECT COUNT(DISTINCT product_category) AS total_product_category FROM
`Target.Products`;
```

total_product_category
73

- There are 73 different product_categories.

Name of the distinct product categories:

```
SELECT DISTINCT product_category FROM `Target.Products`;
```

product_category
null
babies
electronics
Watches present
Garden tools
bed table bath
HEALTH BEAUTY
computer accessories
pet Shop
telephony
Furniture Decoration
perfumery
climatization
stationary store
toys

Orders:

Checking columns and data types:

```
SELECT
  *
FROM target-brazil-
sales.Target.INFORMATION_SCHEMA.COLUMNS WHERE table_name
= 'Orders';
```

table_catalog	table_schema	table_name	column_name	ordinal_position	is_nullable	data_type
target-brazil-sales	Target	Orders	order_id	1	YES	STRING
target-brazil-sales	Target	Orders	customer_id	2	YES	STRING
target-brazil-sales	Target	Orders	order_status	3	YES	STRING
target-brazil-sales	Target	Orders	order_purchase_timestamp	4	YES	TIMESTAMP
target-brazil-sales	Target	Orders	order_approved_at	5	YES	TIMESTAMP
target-brazil-sales	Target	Orders	order_delivered_carrier_date	6	YES	TIMESTAMP
target-brazil-sales	Target	Orders	order_delivered_customer_date	7	YES	TIMESTAMP

Total number of orders:

```
SELECT COUNT(DISTINCT order_id) AS total_order FROM `Target.Orders`;
```

total_order
99441

- The total number of orders are 99441.

Different order_status:

```
SELECT DISTINCT order_status FROM `Target.Orders`;
```

order_status
created
shipped
approved
canceled
invoiced
delivered
processing
unavailable

Number of orders in each order_status:

```
SELECT
order_status,
COUNT(order_id) AS
no_of_orders FROM
`Target.Orders`
GROUP BY order_status
ORDER BY no_of_orders DESC;
```

order_status	no_of_orders
delivered	96478
shipped	1107
canceled	625
unavailable	609
invoiced	314
processing	301
created	5
approved	2

- Most of the orders are delivered. Some are shipped.
- There are many orders which were cancelled and some products are not available.
- No information about the return of the product.

Order_purchase time range in year,month:

```
SELECT
DATE_DIFF(MAX(DATE(order_purchase_timestamp)),
MIN(DATE(order_purchase_timestamp)), YEAR) AS total_year,
DATE_DIFF(MAX(DATE(order_purchase_timestamp)),
MIN(DATE(order_purchase_timestamp)), MONTH) AS total_month,
FROM `Target.Orders`;
```

total_year	total_month
2	25

- The sales data is available for 2 years (25 months).

Order_items:

```
# Checking columns and data types:

SELECT
  *
FROM target-brazil-
sales.Target.INFORMATION_SCHEMA.COLUMNS WHERE table_name
= 'Order_items';
```

table_catalog	table_schema	table_name	column_name	ordinal_position	is_nullable	data_type
target-brazil-sales	Target	Order_items	order_id	1	YES	STRING
target-brazil-sales	Target	Order_items	order_item_id	2	YES	INT64
target-brazil-sales	Target	Order_items	product_id	3	YES	STRING
target-brazil-sales	Target	Order_items	seller_id	4	YES	STRING
target-brazil-sales	Target	Order_items	shipping_limit_date	5	YES	TIMESTAMP
target-brazil-sales	Target	Order_items	price	6	YES	FLOAT64
target-brazil-sales	Target	Order_items	freight_value	7	YES	FLOAT64

```
# Total number of quantities ordered from each product:

SELECT
product_id,
COUNT(DISTINCT order_id)
qty_per_product FROM
`Target.Order_items`
GROUP BY product_id
ORDER BY qty_per_product DESC;
```

product_id	qty_per_product
99a4788cb24856965c36a24e3...	467
aca2eb7d00ea1a7b8ebd4e683...	431
422879e10f46682990de24d77...	352
d1c427060a0f73f6b889a5c7c...	323
389d119b48cf3043d311335e4...	311
53b36df67ebb7c41585e8d54d...	306
368c6c730842d78016ad82389...	291
53759a2ecddad2bb87a079a1f...	287
154e7e31ebfa092203795c972...	269
2b4609f8948be188744942034...	259

```
# Created a view having quantity, total_order_revenue
per order_id from Order_items table where
total_order_revenue = ( price * quantity ) + freight_value:

CREATE VIEW Target.Order_items_revenue AS (SELECT
*,
ROUND((price * quantity) + freight_value, 2) AS
total_order_revenue FROM (
  SELECT
  DISTINC
  T
  order_id,
  order_item_id,
  product_id,
  shipping_limit_date
  , price,
  freight_value,
  COUNT(product_id) OVER(PARTITION BY order_id) AS
  quantity FROM `Target.Order_items`) AS order_items_sales
  ORDER BY order_id);


# To see the data in Order_items_revenue:

SELECT * FROM `Target.Order_items_revenue`;
```

order_id	order_item_id	product_id	shipping_limit_date	price	freight_value	quantity	total_order_revenue
00010242fe8c5a6d1ba2dd792...	1	4244733e06e7ecb4970a6e268...	2017-09-19 09:45:00 UTC	58.9	13.29	1	72.19
00018f77f2f0320c557190d7a1...	1	e5f2d52b802189ee658865ca9...	2017-05-03 11:05:00 UTC	239.9	19.93	1	259.83
000229ec398224ef6ca0657da...	1	c777355d18b72b67abbeef9df...	2018-01-18 14:48:00 UTC	199.0	17.87	1	216.87
00024acbcdf0a6daa1e931b03...	1	7634da152a4610f1595efa32f1...	2018-08-15 10:10:00 UTC	12.99	12.79	1	25.78
00042b26cf59d7ce69dfabb4e...	1	ac6c3623068f30de03045865e...	2017-02-13 13:57:00 UTC	199.9	18.14	1	218.04
00048cc3ae777c65dbb7d2a06...	1	ef92defde845ab8450f9d70c52...	2017-05-23 03:55:00 UTC	21.9	12.69	1	34.59
00054e8431b9d7675808bcb8...	1	8d4f2bb7e93e6710a28f34fa83...	2017-12-14 12:10:00 UTC	19.9	11.85	1	31.75
000576fe39319847cbb9d288c...	1	557d850972a7d6f792fd18ae1...	2018-07-10 12:30:00 UTC	810.0	70.75	1	880.75
0005a1a1728c9d785b8e2b08b...	1	310ae3c140ff94b03219ad0ad...	2018-03-26 18:31:00 UTC	145.95	11.65	1	157.6
0005f50442cb953dcd1d21e1f...	1	4535b0e1091c278dfd193e5a1...	2018-07-06 14:10:00 UTC	53.99	11.4	1	65.39
00061f2a7bc09da83e415a52d...	1	d63c1011f49d98b976c352955...	2018-03-29 22:28:00 UTC	59.99	8.88	1	68.87

```
# Total revenue:

SELECT sum(total_order_revenue) AS total_revenue FROM `Target.Order_items_revenue`;
```

total_revenue
19482807.90999...

- The total revenue is 1,94,82,807.9 dollars.

Payments:

```
# Checking columns and data types:

SELECT
  *
FROM target-brazil-
sales.Target.INFORMATION_SCHEMA.COLUMNS
WHERE table_name = 'Payments';
```

table_catalog	table_schema	table_name	column_name	ordinal_position	is_nullable	data_type
target-brazil-sales	Target	Payments	order_id	1	YES	STRING
target-brazil-sales	Target	Payments	payment_sequential	2	YES	INT64
target-brazil-sales	Target	Payments	payment_type	3	YES	STRING
target-brazil-sales	Target	Payments	payment_installments	4	YES	INT64
target-brazil-sales	Target	Payments	payment_value	5	YES	FLOAT64

```
# Different Payment_type:

SELECT DISTINCT payment_type FROM `Target.Payments`;
```

payment_type
credit_card
voucher
not_defined
debit_card
UPI

- There are 5 modes for payment.

```
# Total payment_value:

SELECT SUM(payment_value) AS total_payment FROM `Target.Payments`;
```

total_payment
16008872.11999...

- The payment_value is 1,60,08,872.11 dollars.

Order_reviews:

```
# Checking columns and data types:

SELECT
  *
FROM target-brazil-
sales.Target.INFORMATION_SCHEMA.COLUMNS WHERE table_name
= 'Order_reviews';
```

table_catalog	table_schema	table_name	column_name	ordinal_position	is_nullable	data_type
target-brazil-sales	Target	Order_reviews	review_id	1	YES	STRING
target-brazil-sales	Target	Order_reviews	order_id	2	YES	STRING
target-brazil-sales	Target	Order_reviews	review_score	3	YES	INT64
target-brazil-sales	Target	Order_reviews	review_comment_title	4	YES	STRING
target-brazil-sales	Target	Order_reviews	review_creation_date	5	YES	TIMESTAMP
target-brazil-sales	Target	Order_reviews	review_answer_timestamp	6	YES	TIMESTAMP

```
# Different review_score/rating:
```

```
SELECT DISTINCT review_score FROM `Target.Order_reviews`;
```

review_score
1
2
3
4
5

```
# Count of ratings:

SELECT
review_score,
COUNT(review_score) AS
rating_count FROM
`Target.Order_reviews`
GROUP BY review_score
ORDER BY rating_count DESC;
```

review_score	rating_count
5	57328
4	19142
1	11424
3	8179
2	3151

- Around 57% products got 5-star rating, 20% got 4-star rating. Yet 23% needs improvement.

```
# Total number of reviews:

SELECT COUNT(review_id) AS total_reviews FROM `Target.Order_reviews`;
```

total_reviews
99224

- Almost all the customers have given review rating.

```
# Total number of comments:

SELECT COUNT(review_comment_title) AS number_of_comments FROM
`Target.Order_reviews`;
```

number_of_commen
11549

- Only 11549 customers have commented.

Geolocation:

Checking columns and data types:

```
SELECT
  *
FROM target-brazil-
sales.Target.INFORMATION_SCHEMA.COLUMNS WHERE table_name
= 'Geolocation';
```

table_catalog	table_schema	table_name	column_name	ordinal_position	is_nullable	data_type
target-brazil-sales	Target	Geolocation	geolocation_zip_code_prefix	1	YES	INT64
target-brazil-sales	Target	Geolocation	geolocation_lat	2	YES	FLOAT64
target-brazil-sales	Target	Geolocation	geolocation_lng	3	YES	FLOAT64
target-brazil-sales	Target	Geolocation	geolocation_city	4	YES	STRING
target-brazil-sales	Target	Geolocation	geolocation_state	5	YES	STRING

Total number of cities:

```
SELECT COUNT(DISTINCT geolocation_city) AS total_city FROM `Target.Geolocation`;
```

total_city
8011

- There are 8011 cities in the dataset.

Total number of states:

```
SELECT  COUNT(DISTINCT geolocation_state) AS total_state FROM `Target.Geolocation` ;
```

total_state
27

- There are 27 states in the dataset.

IN-DEPTH ANALYSIS (KEY-HIGHLIGHTS):

01. checking the structure & characteristics of the data:

 Data type of all columns in the "customers" table:

```
SELECT
  *
FROM target-brazil-
sales.Target.INFORMATION_SCHEMA.COLUMNS WHERE table_name
= 'Customers';
```

table_catalog	table_schema	table_name	column_name	ordinal_position	is_nullable	data_type
target-brazil-sales	Target	Customers	customer_id	1	YES	STRING
target-brazil-sales	Target	Customers	customer_unique_id	2	YES	STRING
target-brazil-sales	Target	Customers	customer_zip_code_prefix	3	YES	INT64
target-brazil-sales	Target	Customers	customer_city	4	YES	STRING
target-brazil-sales	Target	Customers	customer_state	5	YES	STRING

```
SELECT
COLUMN_NAME
, DATA_TYPE
FROM Target.INFORMATION_SCHEMA.COLUMNS
WHERE table_name = 'Customers';
```

COLUMN_NAME	DATA_TYPE
customer_id	STRING
customer_unique_id	STRING
customer_zip_code_prefix	INT64
customer_city	STRING
customer_state	STRING

The time range between which the orders were placed:

```
SELECT
FORMAT_DATETIME('%Y-%m-%d', MIN(order_purchase_timestamp)) AS
min_order_date, FORMAT_DATETIME('%Y-%m-%d', MAX(order_purchase_timestamp))
AS max_order_date FROM `Target.Orders` ;
```

min_order_date	max_order_date
2016-09-04	2018-10-17

```
SELECT
DATE_DIFF(MAX(DATE(order_purchase_timestamp)),
MIN(DATE(order_purchase_timestamp)), MONTH) AS time_range_in_months
FROM `Target.Orders`;
```

time_range_in_months
25

- The orders have been placed between 4th September 2016 and 17th October 2018.

Count of the number of Cities and States in our dataset:

```
SELECT
COUNT(DISTINCT geolocation_city) AS total_city_count,
COUNT(DISTINCT geolocation_state) AS
total_state_count FROM `Target.Geolocation` ;
```

total_city_count	total_state_count
8011	27

- There are 8011 cities and 27 states in our dataset.

02. In-depth Exploration:

To Check if there is a growing trend in the no. of orders placed over the past years:

Average Orders in each year:

```
WITH cte AS(SELECT
order_id,
EXTRACT(YEAR FROM order_purchase_timestamp) AS
year FROM `Target.Orders`)

SELECT
c.year,
COUNT(c.order_id) AS order_count_per_year,
SUM(r.total_order_revenue) AS
total_revenue_per_year FROM cte c
JOIN `Target.Order_items_revenue` r ON c.order_id =
r.order_id GROUP BY c.year
ORDER BY c.year;
```

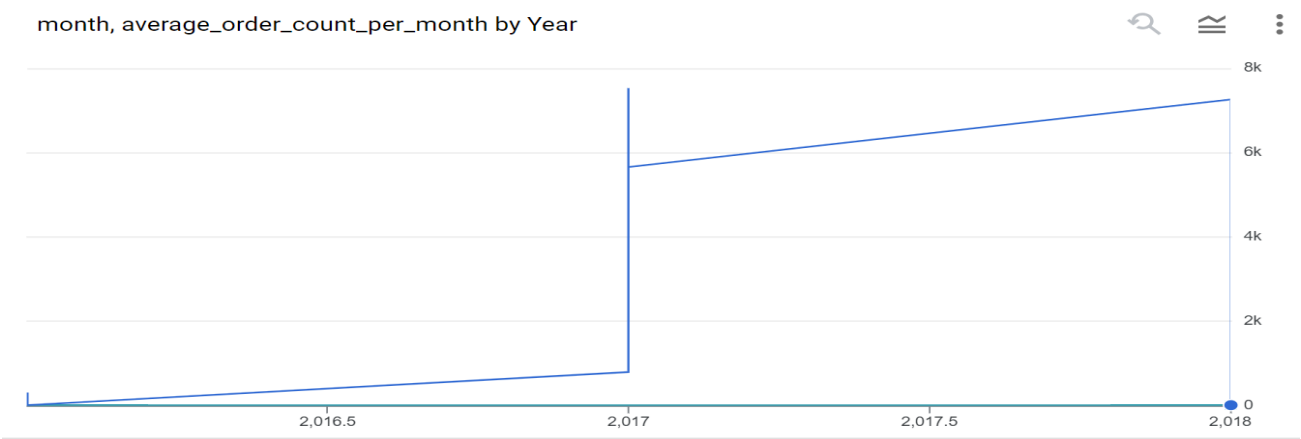
year	order_count_per_year	total_revenue_per_year
2016	370	70514.58
2017	50864	8802865.4699996244
2018	61416	10609427.860000007

- The number of orders has been increased by 136% from 2016 to 2017 and by 20% from 2017 to 2018.
- The revenue has been increased by 123% from 2016 to 2017 and by 20% from 2017 to 2018.

Average Orders per month in each year:

```
WITH cte AS(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.Orders`
GROUP BY year, month
ORDER BY year,
month)

SELECT
Year,
month
,
AVG(order_count) AS
average_order_count_per_month, FROM cte
GROUP BY year,month
ORDER BY
year,month;
```

Year	month	average_order_count_per_month
2016	9	4.0
2016	10	324.0
2016	12	1.0
2017	1	800.0
2017	2	1780.0
2017	3	2682.0
2017	4	2404.0
2017	5	3700.0
2017	6	3245.0
2017	7	4026.0
2017	8	4331.0
2017	9	4285.0



- The number of orders has increased in 2017 and in November sale was highest and then a slightly fall in December and again rise in Jan 2018 and sustains with a slight less sales in other months of 2018.

To check some kind of monthly seasonality in terms of the no. of orders being placed:

Average Orders per month:

```
WITH cte AS(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.Orders`
GROUP BY year, month
ORDER BY year,
month)

SELECT
month,
AVG(order_count) AS
average_order_count_per_month, FROM cte
GROUP BY month
ORDER BY average_order_count_per_month DESC;
```

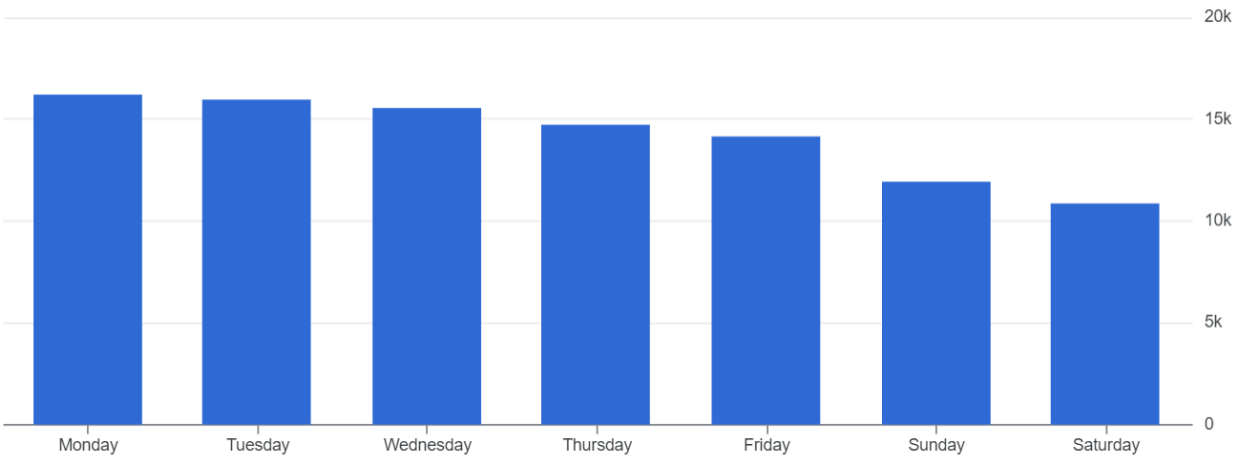
month	average_order_count_per_month
11	7544.0
8	5421.5
5	5286.5
7	5159.0
3	4946.5
6	4706.0
4	4671.5
2	4254.0
1	4034.5
12	2837.0
10	1653.0
9	1435.0

- The average number of orders are higher during the months of November.
- During September, October, December the average number of orders are low.
- In the month of May, July, August the average number of orders are also high compared to other months.

- To check during what time of the day, do the Brazilian customers mostly place their orders (Dawn, Morning, Afternoon or Night):

```
# Total number of orders per day of a week:

WITH cte AS(SELECT
order_id,
FORMAT_DATE('%A', order_purchase_timestamp) AS
day_of_week FROM `Target.Orders`)

SELECT
day_of_week,
COUNT(order_id) AS
total_number_of_orders FROM cte
GROUP BY day_of_week
ORDER BY total_number_of_orders DESC;
```

day_of_week	total_number_of_orders
Monday	16196
Tuesday	15963
Wednesday	15552
Thursday	14761
Friday	14122
Sunday	11960
Saturday	10887



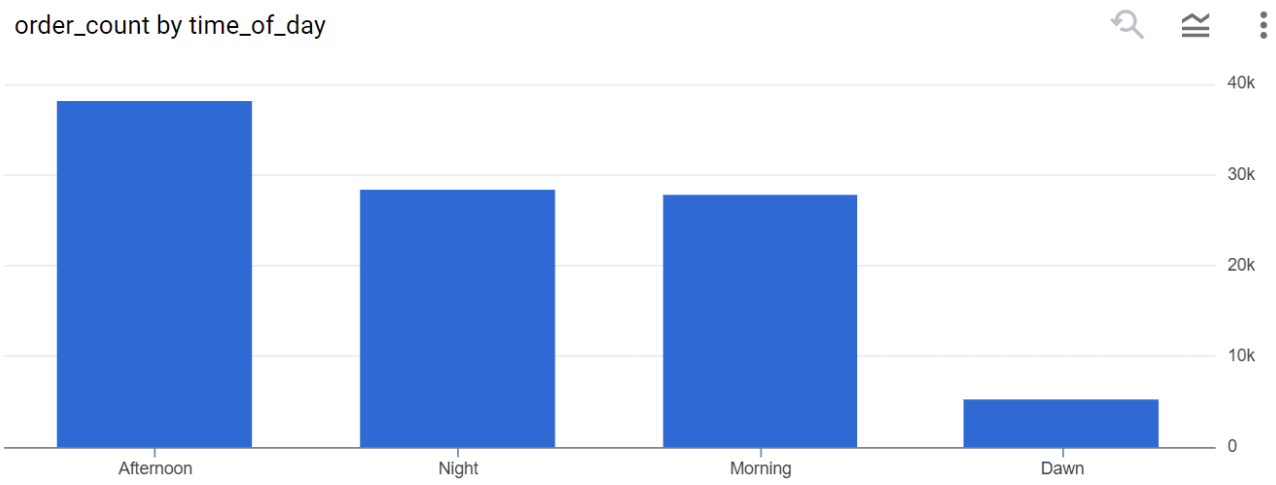
- The orders are high on Monday, Tuesday and Wednesday compared to other days of week.

```
# Total number of orders per hours of a day:

WITH cte AS(SELECT
order_id,
EXTRACT(HOUR FROM order_purchase_timestamp) AS
hour_of_day FROM `Target.Orders`)

SELECT
hour_of_day,
COUNT(order_id) AS
total_number_of_orders, (CASE
WHEN hour_of_day BETWEEN 0 AND 6 THEN 'Dawn'
WHEN hour_of_day BETWEEN 7 AND 12 THEN
'Morning'
WHEN hour_of_day BETWEEN 13 AND 18 THEN
'Afternoon' ELSE 'Night' END) AS time_of_day
FROM cte
GROUP BY hour_of_day
ORDER BY total_number_of_orders DESC;
```

hour_of_day	total_number_of_orders	time_of_day
16	6675	Afternoon
11	6578	Morning
14	6569	Afternoon
13	6518	Afternoon
15	6454	Afternoon
21	6217	Night
20	6193	Night
10	6177	Morning
17	6150	Afternoon
12	5995	Morning
19	5982	Night
22	5816	Night
18	5769	Afternoon

```
# Total number of orders per time_of_day(Dawn, Morning, Afternoon
or Night):

WITH cte AS(SELECT
hour_of_day,
COUNT(order_id) AS
total_number_of_orders, (CASE
WHEN hour_of_day BETWEEN 0 AND 6 THEN 'Dawn'
WHEN hour_of_day BETWEEN 7 AND 12 THEN
'Morning'
WHEN hour_of_day BETWEEN 13 AND 18 THEN
'Afternoon' ELSE 'Night' END) AS time_of_day,
FROM
(SELEC
T
order_id,
EXTRACT(HOUR FROM order_purchase_timestamp) AS
hour_of_day FROM `Target.Orders`) tbl
GROUP BY hour_of_day)

SELECT
time_of_day,
SUM(total_number_of_orders) AS
order_count FROM cte
GROUP BY time_of_day
ORDER BY order_count DESC;
```

time_of_day	order_count
Afternoon	38135
Night	28331
Morning	27733
Dawn	5242



- The customers are purchasing mostly in afternoons.
- The orders are low from 12am. To 6 am.

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

- To get the month on month no. of orders placed in each state:

```
WITH cte AS(SELECT
c.customer_state,
EXTRACT(MONTH FROM o.order_purchase_timestamp) AS
month, COUNT(o.order_id) AS order_count,
FROM `Target.Customers` c
JOIN `Target.Orders` o ON c.customer_id =
o.customer_id GROUP BY c.customer_state, month
ORDER BY c.customer_state, month)

SELECT
customer_state
, month,
order_count,
(order_count - LAG(order_count, 1) OVER(PARTITION BY customer_state ORDER
BY customer_state, month)) AS growth,

CONCAT(ROUND(100 * (order_count - LAG(order_count, 1) OVER(PARTITION BY
customer_state ORDER BY customer_state, month))/LAG(order_count, 1)
OVER(PARTITION BY customer_state ORDER BY customer_state, month),2),'%') AS
growth_percentage FROM cte
ORDER BY customer_state,month;
```

customer_state	month	order_count	growth	growth_percentage
AC	1	8	*null*	*null*
AC	2	6	-2	-25%
AC	3	4	-2	-33.33%
AC	4	9	5	125%
AC	5	10	1	11.11%
AC	6	7	-3	-30%
AC	7	9	2	28.57%
AC	8	7	-2	-22.22%
AC	9	5	-2	-28.57%
AC	10	6	1	20%
AC	11	5	-1	-16.67%
AC	12	5	0	0%
AL	1	39	*null*	*null*
AL	2	39	0	0%
AL	3	40	1	2.56%

- Distribution of customers across all the states:

```
WITH cte1 AS(SELECT
```

```
customer_state,
COUNT(customer_id) AS
customer_count, FROM
`Target.Customers` c
GROUP BY customer_state
ORDER BY customer_count DESC),

cte2 AS(SELECT
SUM(customer_count)
total_customer FROM cte1)

SELECT
cte1.customer_state
,
cte1.customer_count
,
CONCAT(ROUND((100 * cte1.customer_count)/cte2.total_customer, 2),'%')
AS customer_count_percentage
FROM cte1, cte2
ORDER BY cte1.customer_count DESC, customer_count_percentage DESC;
```

customer_state	customer_count	customer_count_percentage
SP	41746	41.98%
RJ	12852	12.92%
MG	11635	11.7%
RS	5466	5.5%
PR	5045	5.07%
SC	3637	3.66%
BA	3380	3.4%
DF	2140	2.15%
ES	2033	2.04%
GO	2020	2.03%
PE	1652	1.66%

- The orders are high on Monday, Tuesday and Wednesday compared to other days of week.

04. Impact on Economy(The money movement by e- commerce by looking at order prices, freight and others):

- To get the % increase in the cost of orders during the months between Jan to Aug from year 2017 to 2018:

```
WITH cte AS(
  SELECT
  EXTRACT(YEAR FROM o.order_purchase_timestamp) AS year,
  EXTRACT(MONTH FROM o.order_purchase_timestamp) AS month,
  SUM(r.total_order_revenue) AS monthly_revenue
  FROM `Target.Orders` o
```

```
  JOIN `Target.Order_items_revenue` r ON o.order_id = r.order_id
  WHERE EXTRACT(YEAR FROM o.order_purchase_timestamp) BETWEEN 2017 AND 2018
        AND
        EXTRACT(MONTH FROM o.order_purchase_timestamp) BETWEEN 1 AND 8
  GROUP BY month, year
  ORDER BY year,month
)

SELECT
year,
month,
monthly_revenue,
(monthly_revenue - LAG (monthly_revenue) OVER (ORDER BY year,month ASC)) AS
revenue_growth,
CONCAT(ROUND(100 * (monthly_revenue - LAG (monthly_revenue) OVER (ORDER BY year,month
ASC))/LAG (monthly_revenue) OVER (ORDER BY year,month ASC),2),'%') AS
revenue_growth_percentage,
LEAD (monthly_revenue,8) OVER (ORDER BY year,month ASC) AS next_year_revenue FROM cte
ORDER BY year,month;
```

year	month	monthly_revenue	revenue_growth	revenue_growth_percentage	next_year_revenue
2017	1	181083.3999999...	null	null	1353555.220000...
2017	2	331990.3099999...	150906.9099999...	83.34%	1245069.260000...
2017	3	507090.5600000...	175100.2500000...	52.74%	1406067.579999...
2017	4	491132.0699999...	-15958.4900000...	-3.15%	1430421.429999...
2017	5	701050.3400000...	209918.2700000...	42.74%	1448603.919999...
2017	6	578996.0800000...	-122054.259999...	-17.41%	1242017.169999...
2017	7	706656.7500000...	127660.6700000...	22.05%	1292943.529999...
2017	8	833081.1700000...	126424.4199999...	17.89%	1190583.289999...
2018	1	1353555.220000...	520474.0500000...	62.48%	null
2018	2	1245069.260000...	-108485.960000...	-8.01%	null
2018	3	1406067.579999...	160998.3199999...	12.93%	null
2018	4	1430421.429999...	24353.84999999...	1.73%	null
2018	5	1448603.919999...	18182.49000000...	1.27%	null

- There is an increase in the cost of orders in Feb, Mar of 2017 and Jan of 2018 during the months from January and August.

The Total & Average value of order price for each state:

```
SELECT
c.customer_state,
ROUND(SUM(r.total_order_revenue),4) AS revenue_per_state,
ROUND(SUM(r.total_order_revenue)/COUNT(DISTINCT o.order_id), 4) AS avg_revenue_per_state
FROM `Target.Customers` c
JOIN  `Target.Orders` o ON c.customer_id = o.customer_id
JOIN `Target.Order_items_revenue` r ON o.order_id = r.order_id GROUP BY c.customer_state
ORDER BY revenue_per_state DESC;
```

customer_state	revenue_per_state	avg_revenue_per_state
SP	7323236.68	176.9967
RJ	2659817.56	208.417
MG	2234841.42	193.5933
RS	1094796.33	201.5457
PR	1009851.35	202.0511
BA	756212.85	225.1974
SC	750427.35	207.7595
GO	484354.09	241.3324
DF	417409.24	196.4279

- The total and average revenues are high in the states SP(Sao Paulo) , RJ(Rio De Janeiro) and MG(Minas Gerais), RS (Rio Grande Do Sul) and PR(Parana) compared to other states.
- These states are in the south and southeast region of Brazil.

The Total & Average value of order freight for each state:

```
SELECT
c.customer_state,
ROUND(SUM(freight_value), 4) AS total_freight_per_state,
ROUND(SUM(freight_value)/COUNT(DISTINCT o.order_id), 4) AS avg_freight_per_state
FROM `Target.Customers` c
JOIN  `Target.Orders` o ON c.customer_id = o.customer_id
JOIN `Target.Order_items_revenue` r ON o.order_id = r.order_id GROUP BY
c.customer_state
ORDER BY avg_freight_per_state DESC;
```

customer_state	total_freight_per_state	avg_freight_per_state
RR	2235.19	48.5911
PB	25719.73	48.3454
RO	11417.38	46.2242
AC	3686.75	45.5154
PI	21218.2	43.0389
MA	31523.77	42.5997
TO	11732.68	42.0526
AP	2788.5	41.0074
SE	14111.47	40.9028
PA	38699.3	39.8962
RN	18860.1	39.1288

- The total and average freight_values are higher in the states RR, PB, RO compared to other states.

- These states are in the northeast and north region of the Brazil.

05. Analysis based on sales, freight and delivery time:

- The no. of days taken to deliver each order from the order's purchase date as delivery time & The difference (in days) between the estimated & actual delivery date of an order:

```
SELECT
order_id,
DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY)
AS delivery_time,
DATE_DIFF(order_estimated_delivery_date, order_delivered_customer_date, DAY)
AS days_between_estimated_and_actual_delivery
FROM `Target.Orders`
WHERE order_status = 'delivered';
```

order_id	delivery_time	days_between_estimated_and_actual_delivery
635c894d068ac37e6e03dc54e...	30	1
3b97562c3aee8bdedcb5c2e45...	32	0
68f47f50f04c4cb6774570cfde...	29	1
276e9ec344d3bf029ff83a161c...	43	-4
54e1a3c2b97fb0809da548a59...	40	-4
fd04fa4105ee8045f6a0139ca5...	37	-1
302bb8109d097a9fc6e9cefc5...	33	-5
66057d37308e787052a32828...	38	-6

Orders and their delivery_status:

```
WITH cte AS(
     SELECT
     order_id,
     DATE_DIFF(order_estimated_delivery_date, order_purchase_timestamp,
     DAY) AS estimated_delivery_time,
     DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp,
     DAY) AS delivery_time,
     DATE_DIFF(order_estimated_delivery_date, order_delivered_customer_date,
     DAY) AS days_between_estimated_and_actual_delivery
     FROM `Target.Orders`
     WHERE order_status = 'delivered'
)

SELECT
*,
 (CASE
 WHEN days_between_estimated_and_actual_delivery < 0 THEN
 'delayed' ELSE 'on_time delivery'
 END
) AS
```

```
delivery_status
FROM cte;
```

order_id	estimated_delivery_time	delivery_time	days_between_estimated_and_actual_delivery	delivery_status
635c894d068ac37e6e03dc54e...	32	30	1	on_time delivery
3b97562c3aee8bdedcb5c2e45...	33	32	0	on_time delivery
68f47f50f04c4cb6774570cfde...	31	29	1	on_time delivery
276e9ec344d3bf029ff83a161c...	39	43	-4	delayed
54e1a3c2b97fb0809da548a59...	36	40	-4	delayed
fd04fa4105ee8045f6a0139ca5...	35	37	-1	delayed
302bb8109d097a9fc6e9cefc5...	28	33	-5	delayed
66057d37308e787052a32828...	32	38	-6	delayed
19135c945c554eebfd7576c73...	33	36	-2	delayed
4493e45e7ca1084efcd38ddebf...	33	34	0	on_time delivery
70c77e51e0f179d75a64a6141...	31	42	-11	delayed
d7918e406132d7c81f1b84527...	31	35	-3	delayed

order_count & delivery_status:

```
WITH CTE AS(

SELECT
*,
(CAS
E
WHEN order_delivered_customer_date IS NULL THEN 'not_yet_delivered'
WHEN (order_delivered_customer_date < order_estimated_delivery_date)
THEN 'on_time_delivery'
ELSE 'delayed'
END) AS
delivery_status FROM
`Target.Orders`
)

SELECT
delivery_status,
COUNT(*) AS
order_count FROM CTE
GROUP BY delivery_status
ORDER BY order_count
DESC;
```

delivery_status	order_count
on_time_delivery	88649
delayed	7827
not_yet_delivered	2965

- 89% of orders are delivered on time and 8% got delayed and 3% are not yet delivered.

The top 5 states with the highest & lowest average freight value:

Top 5 states with highest average freight value:

```
SELECT
c.customer_state,
```

```
AVG(r.freight_value) AS
avg_freight_value FROM
`Target.Customers` c
JOIN  `Target.Orders` o ON c.customer_id = o.customer_id
JOIN `Target.Order_items_revenue` r ON o.order_id =
r.order_id GROUP BY c.customer_state
ORDER BY avg_freight_value
DESC LIMIT 5;
```

customer_state	avg_freight_value
RR	42.984423076923093
PB	42.723803986710955
RO	41.069712230215835
AC	40.073369565217405
PI	39.14797047970476

- The states RR, PB, RO, AC, PI are having highest average freight values and these states also produce low revenues.

Top 5 states with lowest average freight value:

```
SELECT
c.customer_state,
AVG(r.freight_value) AS
avg_freight_value FROM
`Target.Customers` c
JOIN  `Target.Orders` o ON c.customer_id = o.customer_id
JOIN `Target.Order_items_revenue` r ON o.order_id =
r.order_id GROUP BY c.customer_state
ORDER BY avg_freight_value
ASC LIMIT 5;
```

customer_state	avg_freight_value
SP	15.14727539041...
PR	20.53165156794...
MG	20.63016680630...
RJ	20.96092393168...
DF	21.04135494596...

- The states SP, PR, MG, RJ, DF are having lowest average freight values. And these states also produce higher revenues.

The top 5 states with the highest & lowest average delivery time:

Top 5 states with highest average delivery time:

```
SELECT
c.customer_state,
ROUND(AVG(DATE_DIFF(o.order_delivered_customer_date, o.order_purchase_timestamp,
DAY)),2) AS avg_delivery_time
FROM `Target.Customers` c
JOIN  `Target.Orders` o ON c.customer_id = o.customer_id
GROUP BY c.customer_state
ORDER BY avg_delivery_time DESC
LIMIT 5;
```

customer_state	avg_delivery_time
RR	28.98
AP	26.73
AM	25.99
AL	24.04
PA	23.32

- The states RR, AP, AM, AL, PA are having highest delivery time. And these states also produce lower revenues.

Top 5 states with lowest average delivery time:

```
SELECT
c.customer_state,
ROUND(AVG(DATE_DIFF(o.order_delivered_customer_date, o.order_purchase_timestamp,
DAY)), 2) AS avg_delivery_time
FROM `Target.Customers` c
JOIN  `Target.Orders` o ON c.customer_id = o.customer_id
GROUP BY c.customer_state
ORDER BY avg_delivery_time ASC
LIMIT 5;
```

customer_state	avg_delivery_time
SP	8.3
PR	11.53
MG	11.54
DF	12.51
SC	14.48

- The states SP, PR, MG, DF, SC are having lower average delivery time. And these states also produce higher revenues.

The top 5 states where the order delivery is really fast as compared to the estimated date of delivery:

Considering all deliveries:

```
SELECT
c.customer_state,
ROUND(AVG(DATE_DIFF(o.order_estimated_delivery_date,
o.order_delivered_customer_date, DAY)), 2) AS
avg_days_between_estimated_and_actual_delivery
FROM `Target.Customers` c
JOIN  `Target.Orders` o ON c.customer_id = o.customer_id
GROUP BY c.customer_state
ORDER BY avg_days_between_estimated_and_actual_delivery DESC
LIMIT 5;
```

customer_state	avg_days_between_estimated_and_actual_delivery
AC	19.76
RO	19.13
AP	18.73
AM	18.61
RR	16.41

Considering only on_time deliveries:

```
SELECT
c.customer_state,
ROUND(AVG(DATE_DIFF(o.order_estimated_delivery_date,
o.order_delivered_customer_date, DAY)), 2) AS
avg_days_between_estimated_and_actual_delivery
FROM `Target.Customers` c
JOIN  `Target.Orders` o ON c.customer_id = o.customer_id
WHERE o.order_estimated_delivery_date >= o.order_delivered_customer_date
GROUP BY c.customer_state
ORDER BY avg_days_between_estimated_and_actual_delivery DESC
LIMIT 5;
```

customer_state	avg_days_between_estimated_and_actual_delivery
RR	23.75
AP	21.88
AC	21.26
AM	20.28
RO	19.86

06. Analysis based on the payments:

- The month on month no. of orders placed using different payment types:

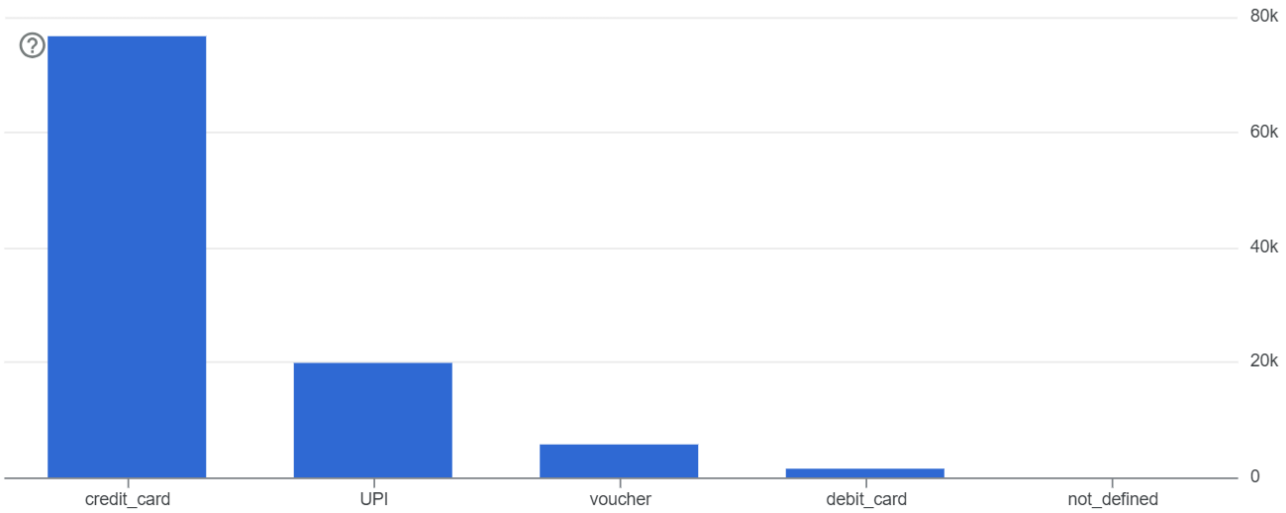
```
WITH cte AS(
  SELECT
  p.payment_type,
  EXTRACT(MONTH FROM o.order_purchase_timestamp) AS month,
  EXTRACT(YEAR FROM o.order_purchase_timestamp) AS year,
  COUNT(o.order_id) AS order_count
  FROM `Target.Orders` o
  RIGHT JOIN `Target.Payments` p ON o.order_id = p.order_id
  GROUP BY p.payment_type, month, year
  ORDER BY p.payment_type, month, year
)

SELECT
payment_type,
month,
order_count,
(order_count - LAG(order_count, 1) OVER(PARTITION BY payment_type ORDER BY payment_type,
month)) AS growth,
CONCAT(ROUND(100 * (order_count - LAG(order_count, 1) OVER(PARTITION BY payment_type
ORDER BY payment_type, month))/LAG(order_count, 1) OVER(ORDER BY payment_type,
month),2),'%') AS growth_percentage
FROM cte
ORDER BY payment_type, month, year;
```

payment_type	month	order_count	growth	growth_percentage
UPI	1	197	-1321	-87.02%
UPI	1	1518	*null*	*null*
UPI	2	398	-927	-69.96%
UPI	2	1325	1128	572.59%
UPI	3	590	-762	-56.36%
UPI	3	1352	954	239.7%
UPI	4	496	-791	-61.46%
UPI	4	1287	697	118.14%
UPI	5	772	276	55.65%
UPI	5	1263	491	63.6%

Orders per Payment_type:

```
SELECT
p.payment_type,
COUNT(o.order_id) AS order_count
FROM `Target.Orders` o
JOIN `Target.Payments` p ON o.order_id = p.order_id
GROUP BY p.payment_type
ORDER BY order_count DESC;
```

payment_type	order_count
credit_card	76795
UPI	19784
voucher	5775
debit_card	1529
not_defined	3

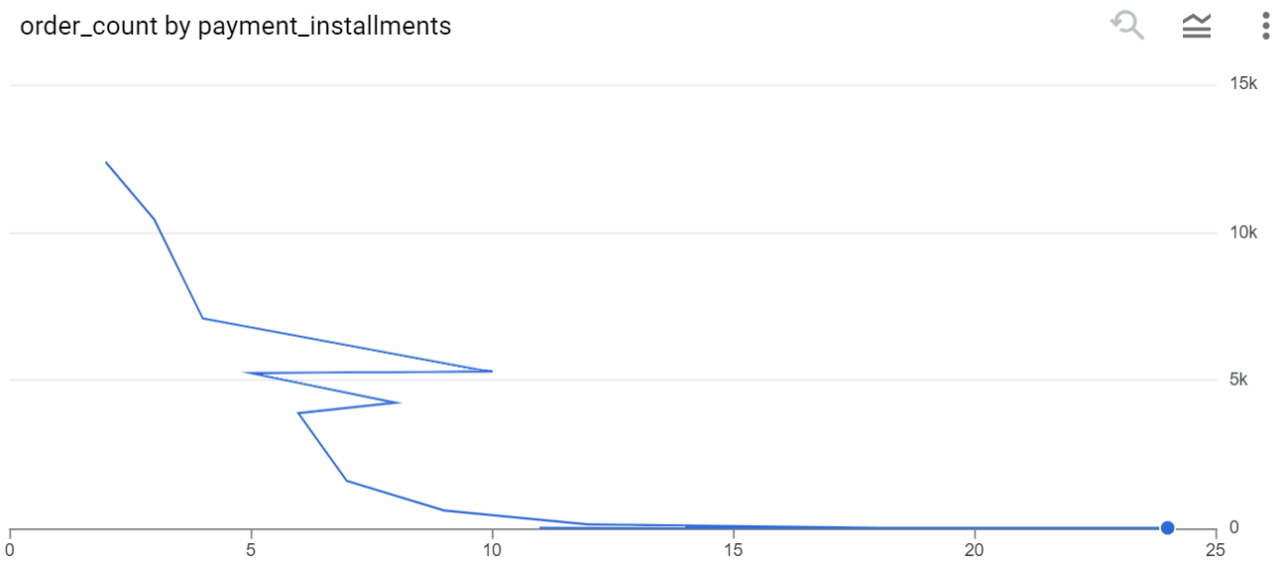


- Highest payment have been done through credit card and then UPI.

The no. of orders placed on the basis of the payment installments that have been paid:

```
SELECT
p.payment_installments,
COUNT(o.order_id) AS order_count
FROM `Target.Orders` o
JOIN `Target.Payments` p ON o.order_id = p.order_id
WHERE p.payment_installments > 1
GROUP BY p.payment_installments
ORDER BY order_count DESC;
```

payment_installments	order_count
2	12413
3	10461
4	7098
10	5328
5	5239
8	4268
6	3920
7	1626
9	644
12	133



Total orders w.r.t Payment_installments:

```
SELECT
SUM(order_count) AS
total_number_of_orders FROM
(SELECT
p.payment_installments,
COUNT(o.order_id) AS
order_count FROM
`Target.Orders` o
JOIN `Target.Payments` p ON o.order_id =
p.order_id GROUP BY p.payment_installments
ORDER BY order_count DESC) tbl;
```

total_number_of_orders
103886

SOME EXTRA ANALYSIS:

Number of customers per state:

```
SELECT
customer_state,
COUNT(customer_id) as
customer_number FROM
`Target.Customers`
GROUP BY customer_state
ORDER BY customer_number DESC;
```

customer_state	customer_number
SP	41746
RJ	12852
MG	11635
RS	5466
PR	5045
SC	3637
BA	3380
DF	2140
ES	2033
GO	2020

- Maximum number of customers belong to the state SP.

Number of orders & revenue in each state:

```
SELECT
c.customer_state,
COUNT(o.order_id) AS order_count_per_state,
SUM(r.total_order_revenue) AS
revenue_per_state FROM `Target.Customers`     c
JOIN `Target.Orders` as o on c.customer_id= o.customer_id
JOIN `Target.Order_items_revenue` r ON o.order_id =
r.order_id GROUP BY c.customer_state
ORDER BY order_count_per_state DESC;
```

customer_state	order_count_per_state	revenue_per_state
SP	47449	7323236.67999987
RJ	14579	2659817.5599999819
MG	13129	2234841.4199999948
RS	6235	1094796.3299999952
PR	5740	1009851.3499999925
SC	4176	750427.35000000068
BA	3799	756212.85000000382
DF	2406	417409.24000000104
GO	2333	484354.08999999991

- The orders and revenues are high in the state SP(Sao Paulo) , RJ(Rio De Janeiro) and MG(Minas Gerais) compared to other states.

Number of products per product_category:

```
SELECT
product_category,
COUNT(DISTINCT product_id) AS
product_count FROM `Target.Products`
GROUP BY product_category
ORDER BY product_count
DESC;
```

product_category	product_count
bed table bath	3029
sport leisure	2867
Furniture Decoration	2657
HEALTH BEAUTY	2444
housewares	2335
automotive	1900
computer accessories	1639
toys	1411
Watches present	1329
telephony	1134
babies	919
perfumery	868
stationary store	849

Top selling product_categories:

```
SELECT
p.p roduct_category,
COUNT(DISTINCT oi.order_id)
qty_per_product_category FROM `Target.Order_items`
oi
JOIN `Target.Products` p ON oi.product_id =
p.product_id GROUP BY p.product_category
ORDER BY qty_per_product_category DESC;
```

product_category	qty_per_product_category
bed table bath	9417
HEALTH BEAUTY	8836
sport leisure	7720
computer accessories	6689
Furniture Decoration	6449
housewares	5884
Watches present	5624
telephony	4199
automotive	3897
toys	3886

- The bed table bath, HEALTH BEAUTY, sport leisure, computer accessories, Furniture Decoration , housewares are the top selling product categories.

Top rating product_ids:

```
SELECT
re.review_score
, oi.product_id
FROM `Target.Order_reviews` re
JOIN `Target.Order_items` oi ON re.order_id =
oi.order_id JOIN `Target.Products` p ON oi.product_id =
p.product_id GROUP BY
p.product_category,re.review_score,oi.product_id ORDER
BY re.review_score DESC;
```

review_score	product_id
5	247fa5b4e2f524a22b21ef256f...
5	3e0f398b664b26e3224c79dbf...
5	870bcc6c58e03ca658cfdd13d...
5	67638c5b9e1d3f0f3c76cd414...
5	3a7c9b0413d7b9cc7f4a18318...
5	a45a0a8751b5e6c64c920b637...
5	8d98dedc3c19de17f1a090195...
5	c19d5ba108c34b8255cf9a524...
5	28b5fef7b6d63771e9784bc68...
5	8ef13bdd5d6da4bc50df1aaa4...

Average month on month running orders:

```
WITH cte AS(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.Orders`
GROUP BY year,month
ORDER BY
year,month)

SELECT
year,
month,
order_count
,
AVG(order_count) OVER(ORDER BY year,month ROWS BETWEEN 1 PRECEDING AND CURRENT ROW)
AS
running_avg_order_count
FROM cte
ORDER BY year,month;
```

year	month	order_count	running_avg_order_count
2016	9	4	4.0
2016	10	324	164.0
2016	12	1	162.5
2017	1	800	400.5
2017	2	1780	1290.0
2017	3	2682	2231.0
2017	4	2404	2543.0
2017	5	3700	3052.0
2017	6	3245	3472.5
2017	7	4026	3635.5

Average month on month running revenue:

```
WITH cte AS(SELECT
EXTRACT(YEAR FROM o.order_purchase_timestamp) AS year,
EXTRACT(MONTH FROM o.order_purchase_timestamp) AS
month, SUM(r.total_order_revenue) AS revenue
FROM `Target.Orders` o
JOIN `Target.Order_items_revenue` r ON o.order_id =
r.order_id GROUP BY year,month
ORDER BY year,month)

SELECT
year,
month,
revenue
,
AVG(revenue) OVER(ORDER BY year,month ROWS BETWEEN 1 PRECEDING AND CURRENT ROW) AS
running_avg_revenu
e FROM cte
ORDER BY year,month;
```

year	month	revenue	running_avg_revenue
2016	9	697.5799999999...	697.579999999999...
2016	10	69797.38000000...	35247.48000000001
2016	12	19.62	34908.5000000000...
2017	1	181083.3999999...	90551.5099999997...
2017	2	331990.3099999...	256536.854999999...
2017	3	507090.5600000...	419540.435000000...
2017	4	491132.0699999...	499111.315000000...
2017	5	701050.3400000...	596091.205000000...
2017	6	578996.0800000...	640023.210000002
2017	7	706656.7500000...	642826.4150000033

The Total & Average value of order price/revenue for each city:

```
SELECT
c.customer_city,
ROUND(SUM(r.total_order_revenue),4) AS revenue_per_city,
ROUND(AVG(r.total_order_revenue), 4) AS
avg_revenue_per_city FROM `Target.Customers` c
JOIN  `Target.Orders` o ON c.customer_id = o.customer_id
JOIN `Target.Order_items_revenue` r ON o.order_id =
r.order_id GROUP BY c.customer_city
ORDER BY revenue_per_city DESC;
```

customer_city	revenue_per_city	avg_revenue_per_city
sao paulo	2735967.85	153.637
rio de janeiro	1504511.23	191.9754
belo horizonte	480279.39	152.7606
brasilia	415737.94	173.8035
curitiba	313871.39	179.2526
porto alegre	301491.67	187.0296
salvador	275842.93	195.3562
campinas	257997.15	155.9838
goiania	201477.75	246.6068
guarulhos	198402.97	149.2874

- The city Sao Paulo is producing higher revenue.

Min_product_price, Max_product_price, Avg_product_price in each product_category:

```
SELECT
p.product_category,
MIN(oi.price)
min_price,
MAX(oi.price)
max_price,
ROUND(AVG(oi.price),2) avg_price,
ROUND(AVG(oi.freight_value),2)
avg_freight_value FROM `Target.Order_items` oi
JOIN `Target.Products` p ON oi.product_id =
p.product_id GROUP BY p.product_category
ORDER BY avg_price DESC, avg_freight_value DESC;
```

product_category	min_price	max_price	avg_price	avg_freight_value
PCs	34.5	6729.0	1098.34	48.45
HOUSE PASTALS OVEN AND C...	10.19	2899.0	624.29	36.16
ELECTRICES 2	13.9	2350.0	476.12	44.54
Agro Industria e Comercio	12.99	2990.0	342.12	27.56
musical instruments	4.9	4399.87	281.62	27.41
electrostile	6.5	4799.0	280.78	23.59
Kitchen portable and food coach	17.42	1099.0	264.57	20.65
fixed telephony	6.0	1790.0	225.69	17.57
CONSTRUCTION SECURITY TO...	8.9	3099.9	208.99	20.2
Watches present	8.99	3999.9	201.14	16.78

- PCs , HOUSE PASTALS OVEN AND CAFÉ, Argo industry and ELECTRICS 2 are having highest average product price categories.

The time taken to approve the order & to reach at the delivery carrier after approval:

```
SELECT
order_id,
DATE_DIFF(order_approved_at,order_purchase_timestamp,DAY) AS
time_to_approve, DATE_DIFF(order_delivered_carrier_date,
order_approved_at,DAY) AS time_taken_to_reach_at_delivery,
DATE_DIFF(order_estimated_delivery_date, order_delivered_customer_date, DAY)
AS delivery_time
FROM `Target.Orders`;
```

order_id	time_to_approve	time_taken_to_reach_at_delivery	delivery_time
770d331c84e5b214bd9dc70a1...	0	4	45
1950d777989f6a877539f5379...	0	0	-12
2c45c33d2f9cb8ff8b1c86cc28...	0	4	28
dabf2b0e35b423f94618bf965f...	0	4	44
8beb59392e21af5eb9547ae1a...	0	5	41
b60b53ad0bb7dacacf2989fe2...	0	1	-5
276e9ec344d3bf029ff83a161c...	0	16	-4
1a0b31f08d0d7e87935b819ed...	0	2	29
cec8f5f7a13e5ab934a486ec9e...	0	17	40
2d846c03073b1a424c1be1a77...	1	3	-7

Average delivery_time per customer_state:

```
SELECT
c.customer_state,
AVG(DATE_DIFF(o.order_approved_at, o.order_purchase_timestamp, DAY))
AS avg_approval_time,
AVG(DATE_DIFF(o.order_delivered_carrier_date, o.order_approved_at, DAY))
AS avg_time_taken_to_reach_at_delivery,
AVG(DATE_DIFF(o.order_delivered_customer_date, o.order_purchase_timestamp, DAY))
AS avg_delivery_time,
AVG(DATE_DIFF(o.order_estimated_delivery_date,
o.order_delivered_customer_date, DAY)) AS
avg_days_between_estimated_and_actual_delivery,
AVG(oi.freight_value) AS
avg_fright_value FROM `Target.Customers`
AS c
JOIN `Target.Orders` AS o ON c.customer_id = o.customer_id
JOIN `Target.Order_items` AS oi ON oi.order_id =
o.order_id GROUP BY c.customer_state
ORDER BY avg_delivery_time DESC;
```

customer_state	avg_approval_time	avg_time_taken_to_reach_at_delivery	avg_delivery_time	avg_days_between_estimated_and_actual_delivery	avg_fright_value
RR	1.480769230769...	2.9019607843137263	27.82608695652...	17.434782608695649	42.98442307692...
AP	0.536585365853...	2.3209876543209877	27.75308641975...	17.444444444444446	34.00609756097...
AM	0.218181818181...	1.9151515151515155	25.96319018404...	18.975460122699378	33.20539393939...
AL	0.351351351351...	2.6353211009174307	23.99297423887...	7.9765807962529287	35.84367117117...
PA	0.323148148148...	2.4874418604651156	23.30170777988...	13.374762808349129	35.83268518518...
MA	0.391251518833...	2.7921760391198029	21.20374999999...	9.1099999999999941	38.25700242718...
SE	0.259740259740...	2.8229166666666674	20.97866666666...	9.1653333333333311	36.65316883116...
CE	0.316858496953...	2.434219495569 19	20.53716690042...	10.256661991584842	32.71420162381...
AC	0.402173913043...	2.3152173913043481	20.32967032967...	20.010989010989011	40.07336956521...
PB	0.413621262458...	2.5494137353433826	20.11945392491...	12.150170648464169	42.72380398671...
RO	0.320143884892...	1.8388278388278387	19.28205128205...	19.080586080586091	41.06971223021...
PI	0.261992619926...	2.3114446529080683	18.93116634799...	10.682600382409174	39.14797047970...

- The average delivery time and freight value are directly proportional to each other.

INSIGHTS:

- We have 99441 customers in our dataset.
- We have 96096 number of Unique Customers ids.
- Customers belong to 14994 different locations.
- Customers are from different 4119 cities and 27 states from Brazil.
- There are 8011 cities and 27 states in our dataset.
- Total sellers are 3095, who are from 611 different cities and 23 states in Brazil and from 2246 different locations as per zip-code.

- Time range for which the data is given is 25 months i.e., from 2016 to 2018.
- compare to 2017, revenue has increased in 2018 by 20%.
- Average number of orders are higher during November, average orders are comparatively low in September and October month. May, July and August have higher average orders compare to other months.
- Monday, Tuesday and Wednesdays have comparatively higher number of orders.
- Most of the customers prefer to order during afternoon followed by night .
- The orders are low from 12am. To 6 am.

- we can observe the trend of increasing orders with time and also for revenue.
- The number of orders has been increased by 136% from 2016 to 2017 and by 20% from 2017 to 2018.
- The revenue has been increased by 123% from 2016 to 2017 and by 20% from 2017 to 2018.

- The growth rate for November is highest!
- The growth rate for July and August in 2017 and 2018 is comparatively very low!
- In the month of May, July, August the average number of orders are also high compared to other months.
- There is an increase in the cost of orders in Feb, Mar of 2017 and Jan of 2018 during the months from January and August.

- There are 32951 different products and 73 different product_categories.
- PCs , HOUSE PASTALS OVEN AND CAFÉ, Argo industry and ELECTRICS 2 are having highest average product price categories.
- The bed table bath, HEALTH BEAUTY, sport leisure, computer accessories, Furniture Decoration , housewares are the top selling product categories.
- The orders and revenues are high in the state SP(Sao Paulo) , RJ(Rio De Janeiro) and MG(Minas Gerais) compared to other states.
- The states RR, PB, RO, AC, PI are having highest average freight values. And these states also produce low revenues.
- The states SP, PR, MG, RJ, DF are having lowest average freight values. And these states also produce higher revenues.
- The states RR, AP, AM, AL, PA are having highest delivery time. And these states also produce lower revenues.
- The states SP, PR, MG, DF, SC are having lower average delivery time. And these states also produce higher revenues.

- The average delivery time and freight value are directly proportional to each other.
- Highest payment have been done through credit card and then UPI.

RECOMMENDATIONS:

- Though there is an increasing trend in the orders and revenue over the time period, but to increase the sales during low selling months like Jan, Apr, June, Sept and Oct, company can provide some kind of discounts .

- The average delivery time should be reduced to increase the number of orders . The delivery time can be reduced by reducing the approval time .

- The north and northeast region of Brazil have high freight value and high delivery time. It should be reduced.

- Adding more products in the top selling categories can increase the revenue.