Mindset:

Evaluation will be kept lenient, so make sure you attempt this case study.

It is understandable that you might struggle with getting started on this. Just brainstorm, discuss with peers, or get help from TAs.

Try to attempt this before it is discussed in the Live Case Discussion with the Instructor.

There is no right or wrong answer. We have to become comfortable dealing with uncertainty in business. This is exactly the skill we want to develop.

Context

Target is one of the world's most recognized brands and one of America's leading retailers. Target makes itself a preferred shopping destination by offering outstanding value, inspiration, innovation and an exceptional guest experience that no other retailer can deliver.

This business case has information of 100k orders from 2016 to 2018 made at Target in Brazil. Its features allows viewing an order from multiple dimensions: from order status, price, payment and freight performance to customer location, product attributes and finally reviews written by customers.

Dataset: https://drive.google.com/drive/folders/1TGEc66YKbD443nslRi1bWgVd238gJCnb (https://drive.google.com/drive/folders/1TGEc66YKbD443nslRi1bWgVd238gJCnb)

Data is available in 8 csv files:

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

Each feature or columns of different CSV files are described below:

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

Features and Description

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:

Features and Description

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:

Features and Description

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:

Features and Description

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:

Features and Description

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:

Features -> Description

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:

Features and Description

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:

Features and Description

product_id -> A unique identifier for the proposed project.

product_category_name -> Name of the product category

product_name_lenght -> length of the string which specifies the name given to the products ordered.

product_description_lenght -> 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.

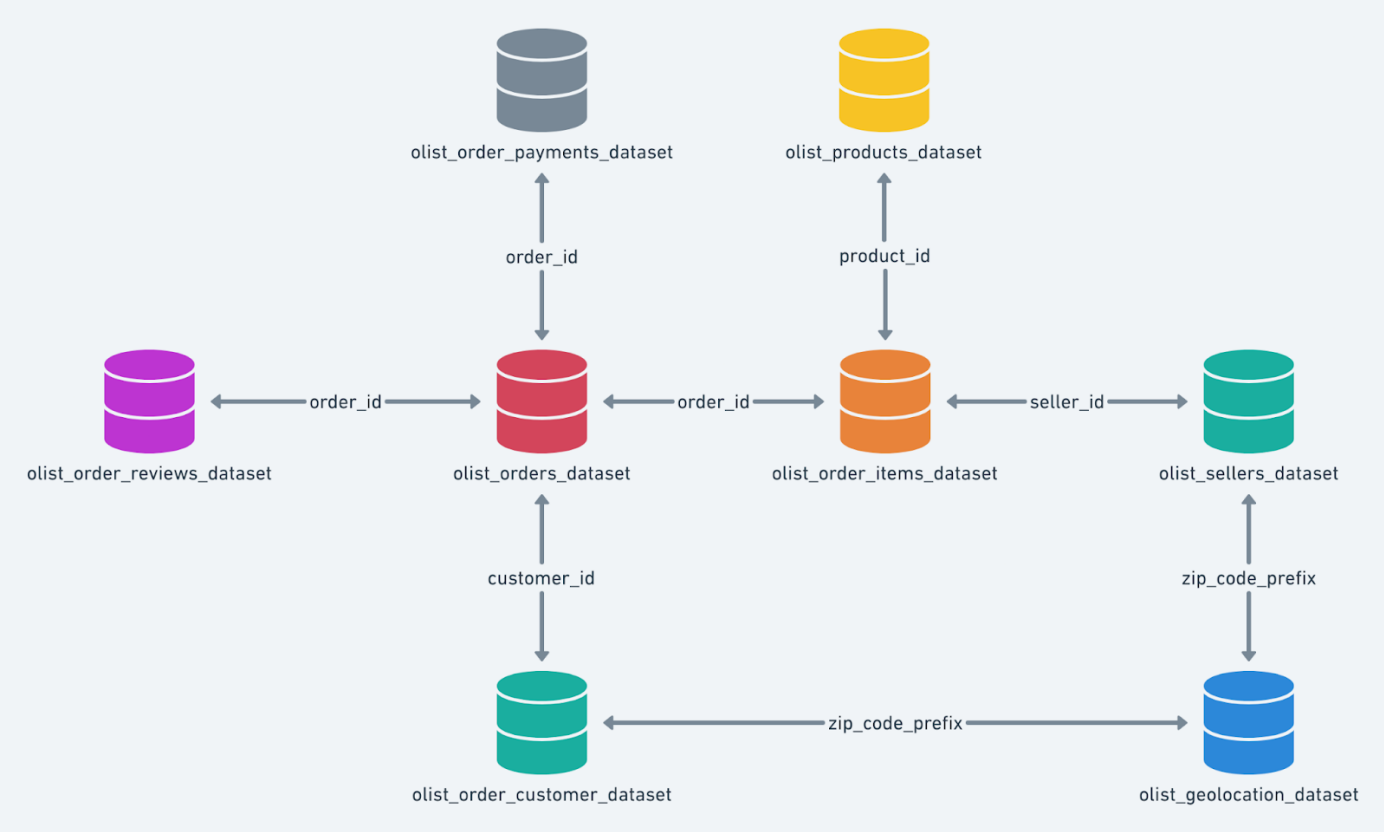
product_weight_g -> Weight of the products ordered in grams.

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.

High level overview of relationship between datasets:



Assume you are a data scientist at Target, and are given this data to analyze and provide some insights and recommendations from it.

What 'good' looks like?

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

```
I) Data type of columns in a table

II) Time period for which the data is given

III) Cities and States covered in the dataset
```

2. In-depth Exploration:

```
  I) 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?

  II) What time do Brazilian customers tend to buy (Dawn, Morning, Afternoon o
r Night)?
```

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

```
   I) Get month on month orders by region, states

   II) How are customers distributed in Brazil
```

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

```
   I) Get % increase in cost of orders from 2017 to 2018 (include months betwee
n Jan to Aug only)

   II) Mean & Sum of price and freight value by customer state
```

5. Analysis on sales, freight and delivery time

```
   I) Calculate days between purchasing, delivering and estimated delivery

   II) Create columns:

       * time_to_delivery = order_purchase_timestamp-order_delivered_customer_d
ate

       * diff_estimated_delivery = order_estimated_delivery_date-order_delivere
d_customer_date

   III) Group data by state, take mean of freight_value, time_to_delivery, diff
_estimated_delivery

   IV) Sort the data to get the following:

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

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

       * Top 5 states where delivery is really fast/ not so fast compared to es
timated date
```

6. Payment type analysis:

```
   I) Month over Month count of orders for different payment types

   II) Distribution of payment installments and count of orders
```

Evaluation Criteria (80 points)

1) Initial exploration of dataset like checking the characteristics of data (10 points)

2) In-depth Exploration (10 points)

3) Evolution of E-commerce orders in the Brazil region (10 points)

4) Impact on Economy (10 points)

5) Analysis on sales, freight and delivery time (10 points)

6) Payment type analysis (10 points)

7) Actionable Insights (10 points)

8) Recommendations (10 points)

Submission Process:

Type your insights and recommendations in the text editor Convert your solutions notebook into PDF, upload it on the dashboard

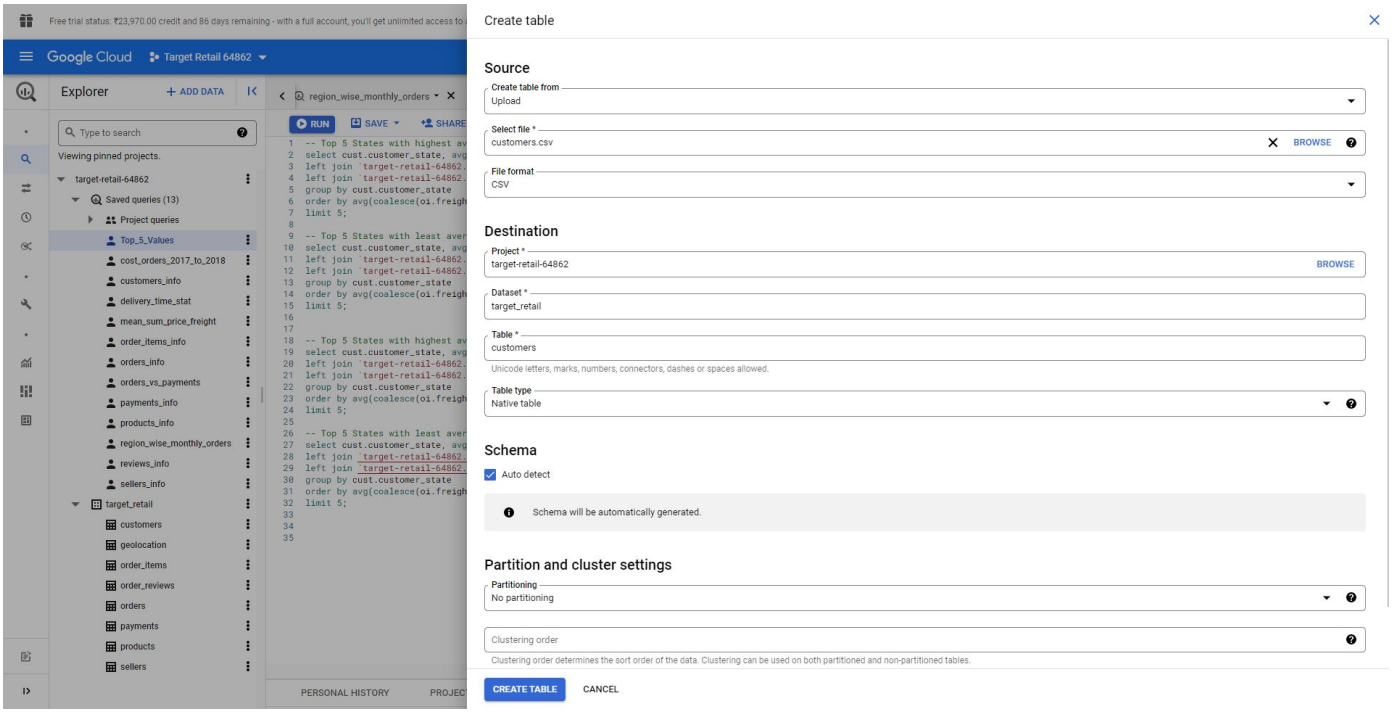
Optionally, you may add images/graphs in the text editor by taking screenshots

After submitting, you will not be allowed to edit your submission

Tool/Platform Used : Google Big Query

Importing the Dataset

All the datasets were uploaded one after the another onto Google Cloud for further analysis on Big Query



Schema of each and every table

Let us look at the datatype of each and every column in the tables available

Table: customers

customer_id : String

customer_unique_id : String

customer_zip_code_prefix : Integer

customer_city : String

customer_state : String

Table: geolocations

geolocation_zip_code_prefix : Integer

geolocation_lat : Float

geolocation_lng : Float

geolocation_city : String

geolocation_state : String

Table: order_items

order_id : String

order_item_id : Integer

product_id : String

seller_id : String

shipping_limit_date : Timestamp

price : Float

freight_value : Float

Table: order_reviews

review_id : String

order_id : String

review_score : Integer

review_comment_title : String

review_creation_date : Timestamp

review_answer_timestamp : Timstamp

Table: orders

order_id : String

customer_id : String

order_status : String

order_purchase_timestamp : Timestamp

order_delivered_carrier_date : Timestamp

order_delivered_customer_date : Timestamp

order_estimated_delivery_date : Timestamp

Table: payments

order_id : String

payment_sequential : Integer

payment_type : String

payment_installments : Integer

payment_value : Float

Table: products

product_id : String

product_category_name : String

product_name_lenght : Integer

product_description_lenght : Integer

product_photos_qty : Integer

product_weight_g : Integer

product_length_cm : Integer

product_height_cm : Integer

product_width_cm : Integer

Table: sellers

seller_id : String

seller_zip_code_prefix : Integer

seller_city : String

seller_state : String

Exploratory Data Analysis

Table: customers

There are 99441 records in the customers table

query -> select count(*) from `target-retail-64862.target_retail.customers` ;

All the 99441 customer_id's in the customers table are unique

query -> select count(distinct customer_id) from `target-retail-64862.target_retail.customers` ;

There are 4119 distinct Brazilian cities in the customers table

query -> select count(distinct(customer_city)) from `target-retail-64862.target_retail.customers` ;

There are 27 distinct Brazilian states in the customers table

query -> select count(distinct(customer_state)) from `target-retail-64862.target_retail.customers` ;

São Paulo (SP) has the highest and Curitiba (RR) has the least number of customers amongst states

query -> select customer_state,

count(*) as state_count from `target-retail-64862.target_retail.customers`

group by customer_state

order by count(*) desc;

Sao Paulo City has the highest number of customers amongst all cities

query -> select customer_city,

count(*) as city_count from `target-retail-64862.target_retail.customers`

group by customer_city

order by count(*) desc;

Belo Horizonte (MG) has the highest and Curitiba (RR) has the least number of cities in the customers table amongst all the states for Target Retail

query -> select customer_state,

count(distinct customer_city) as city_count from `target-retail-64862.target_retail.customers`

group by customer_state

order by count(distinct customer_city) desc;

Table: orders

There are 99441 records in the orders table

query -> select count(*) from `target-retail-64862.target_retail.orders` ;

There are 99441 distinct customer records in the orders table

query -> select count(distinct customer_id) from `target-retail-64862.target_retail.orders` ;

There are 99441 distinct order_ids in the orders table i.e. each customer has a unique order_id associated with them

query -> select count(distinct order_id) from `target-retail-64862.target_retail.orders` ;

All the unique 99441 records have order status associated with them in the orders table

query -> select count(order_status) from `target-retail-64862.target_retail.orders`

There are 8 Distinct order status across orders table - created, shipped, approved, canceled, invoiced, delivered, processing and unavailable

query -> select distinct(order_status) from `target-retail-64862.target_retail.orders` ;

Orders table has purchase records between - 4th September 2016 to 17th October 2018

query -> select min(order_purchase_timestamp) as first_order_date, max(order_purchase_timestamp) as latest_order_date from `target-retail-64862.target_retail.orders` ;

There are no missing records for purchase date in the orders table

query -> select order_purchase_timestamp from `target-retail-64862.target_retail.orders` where order_purchase_timestamp is null;

Table: order_items

There are 112650 records in the order_items table

query -> select count(*) from `target-retail-64862.target_retail.order_items` ;

There are 98666 distinct orders in the order_items table. i.e. 775(99441-98666) customers have no corresponding record entries in order_items table

query -> select count(distinct order_id) from `target-retail-64862.target_retail.order_items` ;

There are no missing records in the order_item_id column

query -> select count(order_item_id) from `target-retail-64862.target_retail.order_items` ;

Count of customers by number of orders placed

query -> select ord_item.total_items,

count(*) as number_of_customers from

(select order_id, max(order_item_id) as total_items from `target-retail-64862.target_retail.order_items` group by order_id) as ord_item

group by ord_item.total_items

order by ord_item.total_items;

Count_of_Customers_by_Number_of_Items_ordered

Total Items	
1	88,863
2	7,516
3	1,322
4	505
5	204
6	198
7	22
8	8
9	3
10	8
11	4
12	5
13	1
14	2
15	2
20	2
21	1

There are 32951 distinct products in the order_items table

query -> select count(distinct product_id) from `target-retail-64862.target_retail.order_items` ;

There are 3095 distinct sellers in the order_items table

query -> select count(distinct seller_id) from `target-retail-64862.target_retail.order_items` ;

Range of shipping order dates - 19-09-2016 to 09-04-2020

query -> select max(shipping_limit_date), min(shipping_limit_date) from `target-retail-64862.target_retail.order_items` ;

No missing records for price or freight_value column in the order_items table

query -> select * from `target-retail-64862.target_retail.order_items` where price is null or freight_value is null;

Summary statistics of price in the order_items table

query -> select min(price) as min_price,

max(price) as max_price,

round(stddev(price),2) as stddev_price,

round(avg(price),2) as mean_price,

max(price)-min(price) as price_range from `target-retail-64862.target_retail.order_items` ;

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	min_price	max_price	stddev_price	mean_price	price_range
1	0.85	6735.0	183.63	120.65	6734.15

Summary statistics of freight_value in the order_items table (Insert Image)

query -> select min(freight_value) as min_freight,

max(freight_value) as max_freight,

round(stddev(freight_value),2) as stddev_freight,

round(avg(freight_value),2) as mean_freight,

max(freight_value)-min(freight_value) as freight_range from `target-retail-64862.target_retail.order_items` ;

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	min_freight_value	max_freight_value	stddev_freight_value	mean_freight_value	freight_value_range
1	0.0	409.68	15.81	19.99	409.68

Table: order_reviews

There are 99224 records in order_reviews table

query -> select count(*) from `target-retail-64862.target_retail.order_reviews` ;

There are 98673 distinct order_ids in order_reviews table

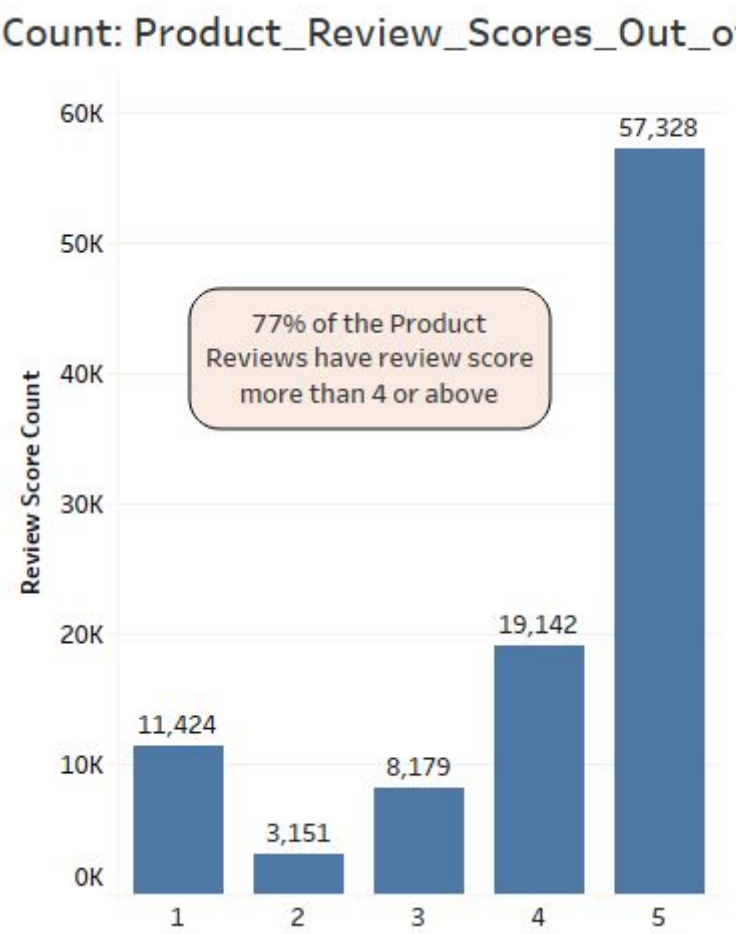
query -> select count(distinct order_id) from `target-retail-64862.target_retail.order_reviews` ;

There are 98410 distinct review_ids in order_reviews table

query -> select count(distinct review_id) from `target-retail-64862.target_retail.order_reviews` ;

Count of different review scores for different orders

query -> select review_score, count(*) as review_score_count from `target-retail-64862.target_retail.order_reviews` group by review_score order by review_score;



Let's look at the distribution of time taken (in hours) to submit a review

query -> select case when hour_dist.bracket = 1 then '25th Percentile' when hour_dist.bracket = 2 then '50th Percentile' when hour_dist.bracket = 3 then '75th Percentile' when hour_dist.bracket = 4 then '100th Percentile' else '' end as percentile_bracket, min(hour_dist.hours) as min_hours, max(hour_dist.hours) as max_hours, max(hour_dist.hours)-min(hour_dist.hours) as hours_range from (select *, timestamp_diff(review_answer_timestamp, review_creation_date, hour) as hours, ntile(4) over (order by timestamp_diff(review_answer_timestamp, review_creation_date, hour)) as bracket from `target-retail-64862.target_retail.order_reviews`) as hour_dist group by hour_dist.bracket order by hour_dist.bracket;

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	percentile_bracket	min_hours	max_hours	hours_range
1	25th Percentile	2	24	22
2	50th Percentile	24	40	16
3	75th Percentile	40	74	34
4	100th Percentile	74	12448	12374

From the above table we that Inter Quartile Range for time taken to submit a review is = 74 - 24 = 50 hours

So anything above 74 + 1.5 * 50 = 74 + 75 ~ 149 hours is a potential outlier

We see that 6305 records are outliers or have review completion time more than 149 hours or nearly 6 days

query -> select count(*) from `target-retail-64862.target_retail.order_reviews` where timestamp_diff(review_answer_timestamp, review_creation_date, hour) > 149;

Table: payments

There are 103886 records in the payments table

query -> select count(*) from `target-retail-64862.target_retail.payments` ;

There are 99440 distinct order_ids in the payments table

query -> select count(distinct order_id) from `target-retail-64862.target_retail.payments` ;

There is one missing record in the payments table for customer with id - 86dc2ffce2dfff336de2f386a786e574 with order_id - bfbd0f9bdef84302105ad712db648a6c

query -> select ord.order_id, ord.customer_id, pay.order_id from `target-retail-64862.target_retail.orders` ord left join `target-retail-64862.target_retail.payments` pay on pay.order_id = ord.order_id where pay.order_id is NULL;

There are 29 distinct sequences of payments

query -> select distinct payment_sequential from `target-retail-64862.target_retail.payments` ;

There are 5 different payment methods in the payments table

query -> select distinct payment_type from `target-retail-64862.target_retail.payments` ;

There are 24 distinct installments in the payments table

query -> select distinct payment_installments from `target-retail-64862.target_retail.payments` ;

Table: products

There are 32951 records in the products table

query -> select count(*) from `target-retail-64862.target_retail.products` ;

There are 32951 distinct product_id in the products table

query -> select count(distinct product_id) from `target-retail-64862.target_retail.products` ;

There are 74 distinct categories of product in the products table

query -> select product_category,

count(*) as product_category_count

from `target-retail-64862.target_retail.products`

group by product_category

order by count(*) desc;

Count of product by photos taken

query -> select product_photos_qty,

count(*) as photo_count

from `target-retail-64862.target_retail.products`

group by product_photos_qty

order by count(*) desc;

Showing count for only Top 10 -

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	product_category	product_category_count
1	bed table bath	3029
2	sport leisure	2867
3	Furniture Decoration	2657
4	HEALTH BEAUTY	2444
5	housewares	2335
6	automotive	1900
7	computer accessories	1639
8	toys	1411
9	Watches present	1329
10	telephony	1134

Let's look at the distribution of weight for different products

query -> select

case when weight_dist.bracket = 1 then '25th Percentile' when weight_dist.bracket = 2 then '50th Percentile' when weight_dist.bracket = 3 then '75th Percentile' when weight_dist.bracket = 4 then '100th Percentile' else '' end as weight_distribution,

min(weight_dist.product_weight_g) as min_weight,

max(weight_dist.product_weight_g) as max_weight,

max(weight_dist.product_weight_g)-min(weight_dist.product_weight_g) as weight_range from

(select *, ntile(4) over (order by product_weight_g) as bracket from `target-retail-64862.target_retail.products`) as weight_dist

group by weight_dist.bracket

order by weight_dist.bracket;

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	weight_distribution	min_weight	max_weight	weight_range
1	25th Percentile	0	300	300
2	50th Percentile	300	700	400
3	75th Percentile	700	1900	1200
4	100th Percentile	1900	40425	38525

From the above table we that Inter Quartile Range for weight distribution of products = 1900 - 300 = 1600 grams or 1.6 kilograms

So anything above 1900 + 1.5 * 1600 = 1900 + 2400 ~ 3300 grams or 3.3 kilograms is a potential outlier

We see that 5398 records/products are outliers or have product weight more than 3300 grams or 3.3 kgs

query -> select count(*) from `target-retail-64862.target_retail.products` where product_weight_g > 3300;

Let's look at the distribution of volume for different products

query -> select

case when vol_dist.bracket = 1 then '25th Percentile' when vol_dist.bracket = 2 then '50th Percentile' when vol_dist.bracket = 3 then '75th Percentile' when vol_dist.bracket = 4 then '100th Percentile' else '' end as volume_distribution,

min(vol_dist.product_volume) as min_vol,

max(vol_dist.product_volume) as max_vol,

max(vol_dist.product_volume)-min(vol_dist.product_volume) as vol_range from

(select *,

product_height_cm * product_length_cm * product_width_cm as product_volume,

ntile(4) over (order by product_height_cm * product_length_cm * product_width_cm)

as bracket from `target-retail-64862.target_retail.products`) as vol_dist

group by vol_dist.bracket

order by vol_dist.bracket;

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	volume_distribution	min_vol	max_vol	vol_range
1	25th Percentile	168	2880	2712
2	50th Percentile	2880	6840	3960
3	75th Percentile	6840	18480	11640
4	100th Percentile	18496	296208	277712

From the above table we that Inter Quartile Range for volume distribution of products = 18480 - 2880 = 15600 cubic cm

So anything above 18480 + 1.5 * 15600 = 18480 + 23400 ~ 41880 cubic cm is a potential outlier

We see that 3262 records/products are outliers or have product volume more than 41880 cubic cm

query -> select count(*) from `target-retail-64862.target_retail.products` where product_height_cm * product_length_cm * product_width_cm > 41880;

Table: sellers

There are 3095 records in the sellers table

query -> select count(*) from `target-retail-64862.target_retail.sellers` ;

There are 3095 distinct seller_ids in the sellers table

query -> select count(distinct seller_id) from `target-retail-64862.target_retail.sellers` ;

There are 611 distinct cities in the sellers table

query -> select count(distinct(seller_city)) from `target-retail-64862.target_retail.sellers` ;

There are 23 distinct states in the sellers table query -> select count(distinct(seller_state)) from `target-retail-64862.target_retail.sellers` ;

Sao Paolo (SP) state has the highest number of sellers

query -> select seller_state,

count(*) as state_count from `target-retail-64862.target_retail.sellers`

group by seller_state

order by count(*) desc;

Sao Paulo city has the highest number of sellers

query -> select seller_city,

count(*) as city_count from `target-retail-64862.target_retail.sellers`

group by seller_city

order by count(*) desc;

Sao Paolo (SP) state has the highest number of distinct cities where sellers are available

query -> select seller_state,

count(distinct seller_city) as city_count from `target-retail-64862.target_retail.sellers`

group by seller_state

order by count(distinct seller_city) desc;

Trend of E-Commerce in Brazil

Let's analyze the customer orders month over month

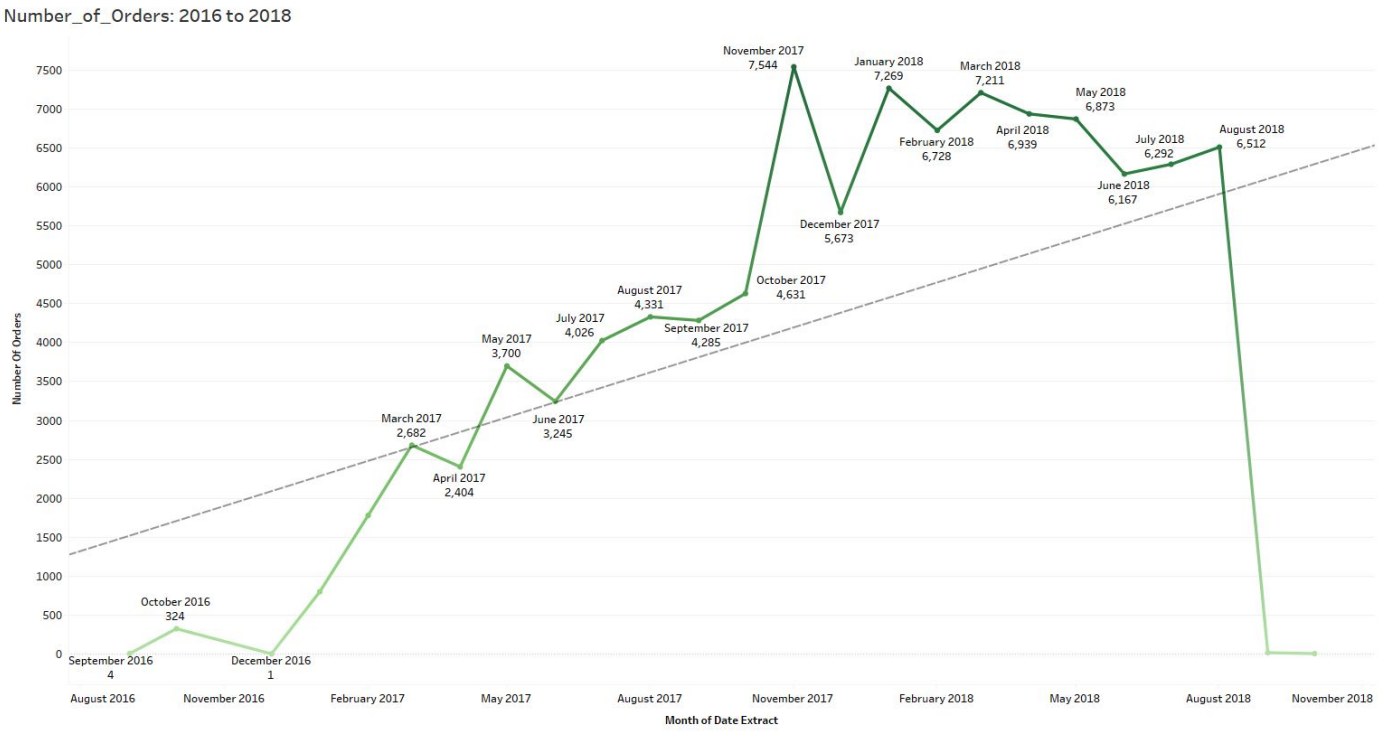
query -> select purc_ord.order_date,

count(*) as count_of_orders from

(select customer_id, FORMAT_TIMESTAMP("%Y-%m-%d", order_purchase_timestamp) as order_date from `target-retail-64862.target_retail.orders`) as purc_ord

group by purc_ord.order_date

order by purc_ord.order_date;



Trend: As we see from the figure above there is an increasing trend in number of orders between September 2016 to November 2017 and thereafter the number of orders have fluctuated between November 2017 to August 2018 with zero or no increase in the number of orders.

Seasonality: From the graph we see some random spikes but no repeat in trend over a period of time to give a definitive stand on seasonality of customer orders. We need to analyse data for more periods to come to a conclusion on seasonality of orders.

Let's analyze how Brazilians like to order over a period of 24 hours in a day

query -> select ord_week.Is_Weekend,

ord_week.hour_of_the_day,

count(*) as count_of_orders from

(select customer_id, order_purchase_timestamp, EXTRACT(hour from order_purchase_timestamp) as hour_of_the_day, EXTRACT(DAYOFWEEK FROM order_purchase_timestamp) as day_of_week,

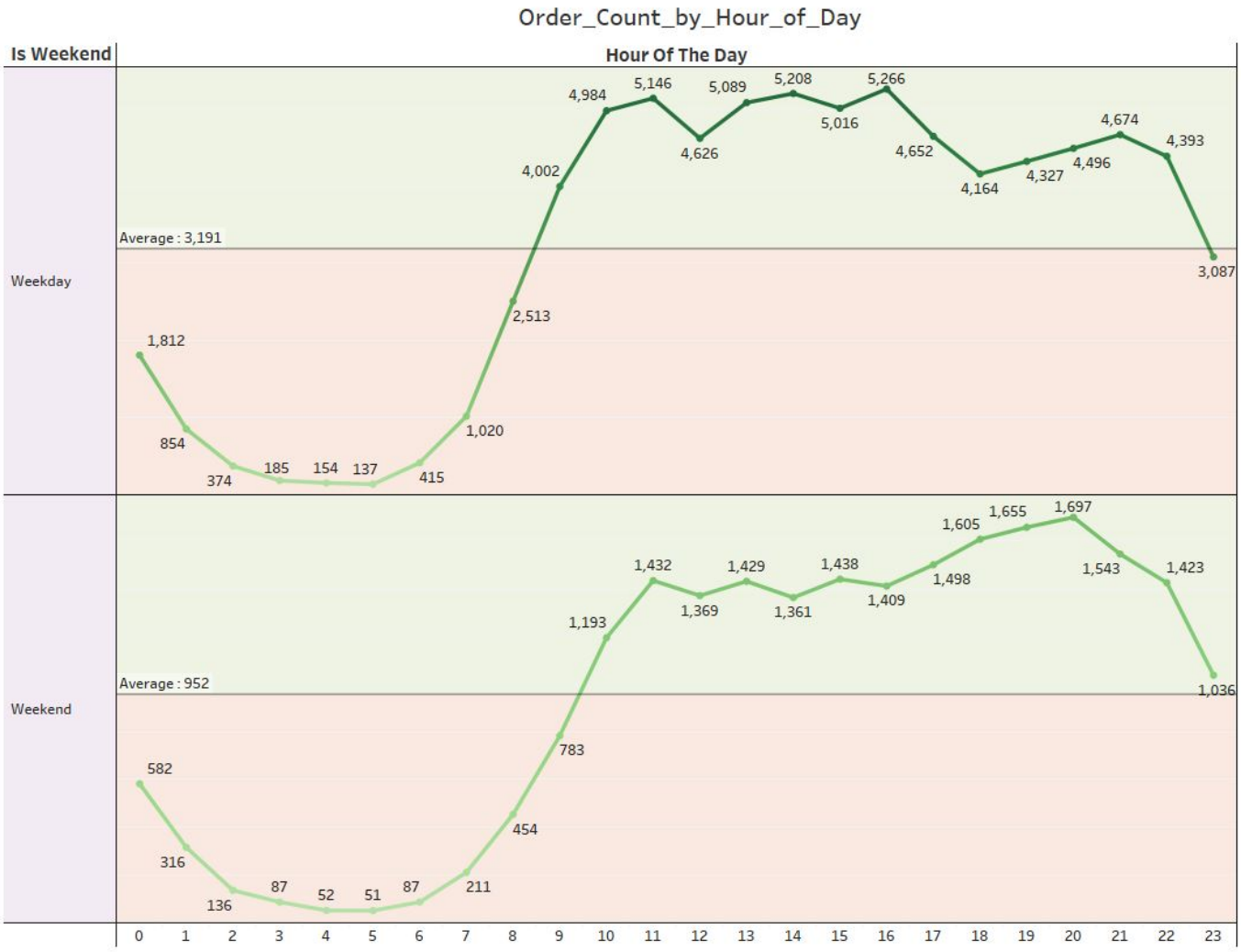
case when EXTRACT(DAYOFWEEK FROM order_purchase_timestamp) IN (2,3,4,5,6) THEN 'Weekday' when EXTRACT(DAYOFWEEK FROM order_purchase_timestamp) IN (7,1) THEN 'Weekend' else '' end as Is_Weekend

from `target-retail-64862.target_retail.orders`

order by order_purchase_timestamp) as ord_week

group by ord_week.Is_Weekend, ord_week.hour_of_the_day

order by ord_week.Is_Weekend, ord_week.hour_of_the_day;



Insights:

- The number of orders is above average between 10 a.m. to 11 p.m. in a day irrespective of whether it's a weekday or a weekend.

- During weekdays people like to shop between 10 a.m. to 4 p.m. and then there is a slight decrease in number of orders in the evening between 5 p.m. to 6 p.m. before increasing between 7 p.m. to 10 p.m.

- During weekends the number of orders in the morning and afternoon hours (10 a.m. to 3 p.m.) is relatively stable before increasing in the evening between 4 p.m. to 8 p.m. and then it decrease considerably after 9 p.m.

Recommendations:

- Target Retail should provide deals on products or run promotional campaigns only during evening hours (5 p.m to 7 p.m) or odd hours on weekdays (midnight to 8 a.m.) [Exception could be days on which there are festivals or carnivals]

- Target Retail should provide deals on products or run promotional campaigns only during odd hours on weekdays (midnight to 8 a.m.) [Exception could be days on which there are festivals or carnivals]

Evolution of E-commerce orders in the Brazil region

Let's analyze month on month orders by region

The whole of Brazil is divided into 5 regions:

```
North
NorthEast
CenterWest
South
SouthEast
```

Because we do not have regional data in any of the dataset, let's first map customers into different regions based on data available for different states.

Brazilian States and the Regions: https://brazil-help.com/brazilian_states.htm (https://brazil-help.com/brazilian_states.htm)

query -> select reg.region,

reg.year_extract,

reg.month_extract,

count(*) as count_of_orders from

(select cust.customer_state,

CASE WHEN cust.customer_state IN ('SP','RJ','MG','ES') THEN 'SouthEast'

WHEN cust.customer_state IN ('RS','PR','SC') THEN 'South'

WHEN cust.customer_state IN ('BA','PE','CE','MA','PB','PI','RN','AL','SE') THEN 'NorthEast'

WHEN cust.customer_state IN ('DF','GO','MT','MS') THEN 'CenterWest'

WHEN cust.customer_state IN ('PA','TO','RO','AM','AC','AP','RR') THEN 'North'

ELSE 'NA'

END AS Region,

EXTRACT(year from ord.order_purchase_timestamp) as year_extract,

EXTRACT(month from ord.order_purchase_timestamp) as month_extract from

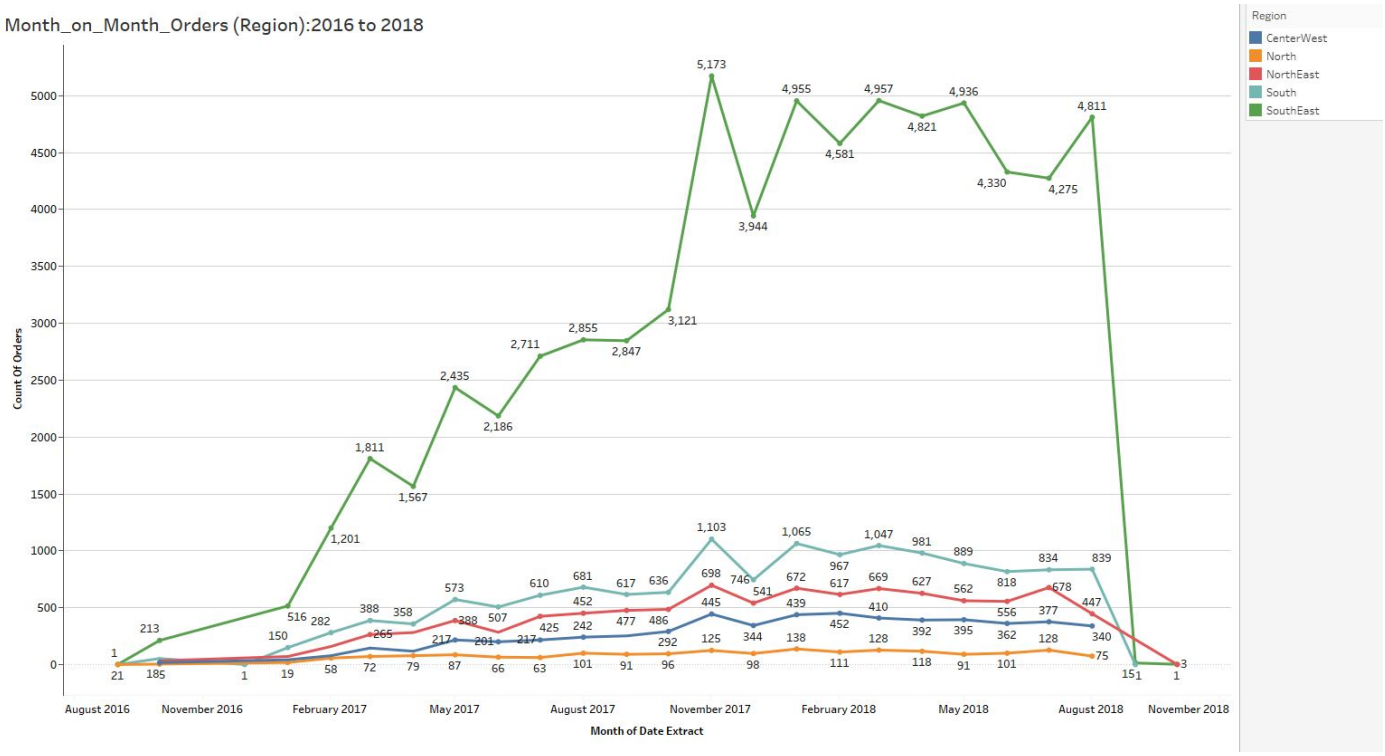
`target-retail-64862.target_retail.customers` cust

left join `target-retail-64862.target_retail.orders` ord

on ord.customer_id = cust.customer_id) as reg

group by reg.region, reg.year_extract, reg.month_extract

order by reg.region, reg.year_extract, reg.month_extract;



Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	region	count_of_orders	percent_total_orders
1	SouthEast	68266	68.65
2	South	14148	14.23
3	NorthEast	9394	9.45
4	CenterWest	5782	5.81
5	North	1851	1.86

Insights:

```
  - The SouthEast Region of Brazil comprises of nearly 69% of the total orders for
Target Retail. For this region,
  there is an increase in the volume of orders (month on month) from September 201
6 to November 2017 followed by
  dip in volume of orders (month on month) from November 2017 to August 2018 (Exce
ption being increase and decrease
  in orders for couple of months during that period of November 2017 to August 201
8).

  - South Region of Brazil comprises of nearly 14% of the total orders for Target
 retail. For this region,
  there is an increase in the volume of orders (month on month) from September 201
6 to November 2017 followed by
  dip in volume of orders (month on month) from November 2017 to August 2018 (Exce
ption being increase and decrease
  in orders for couple of months during that period of November 2017 to August 201
8).

  - NorthEast, CenterWest and North Region of Brazil comprises of rest of the 17%
 of the orders for Target Retail.
  The volume of increase in month on month orders for NorthEast, CenterWest and No
rth Region is relatively low
  when compared to SouthEast or South Region of Brazil.
```

Let's analyze month on month orders by top 5 states

query -> select cust_ord.customer_state,

cust_ord.year_extract,

cust_ord.month_extract,

count(*) as count_of_orders from

(select cust.customer_state,

EXTRACT(year from ord.order_purchase_timestamp) as year_extract,

EXTRACT(month from ord.order_purchase_timestamp) as month_extract

from `target-retail-64862.target_retail.customers` cust

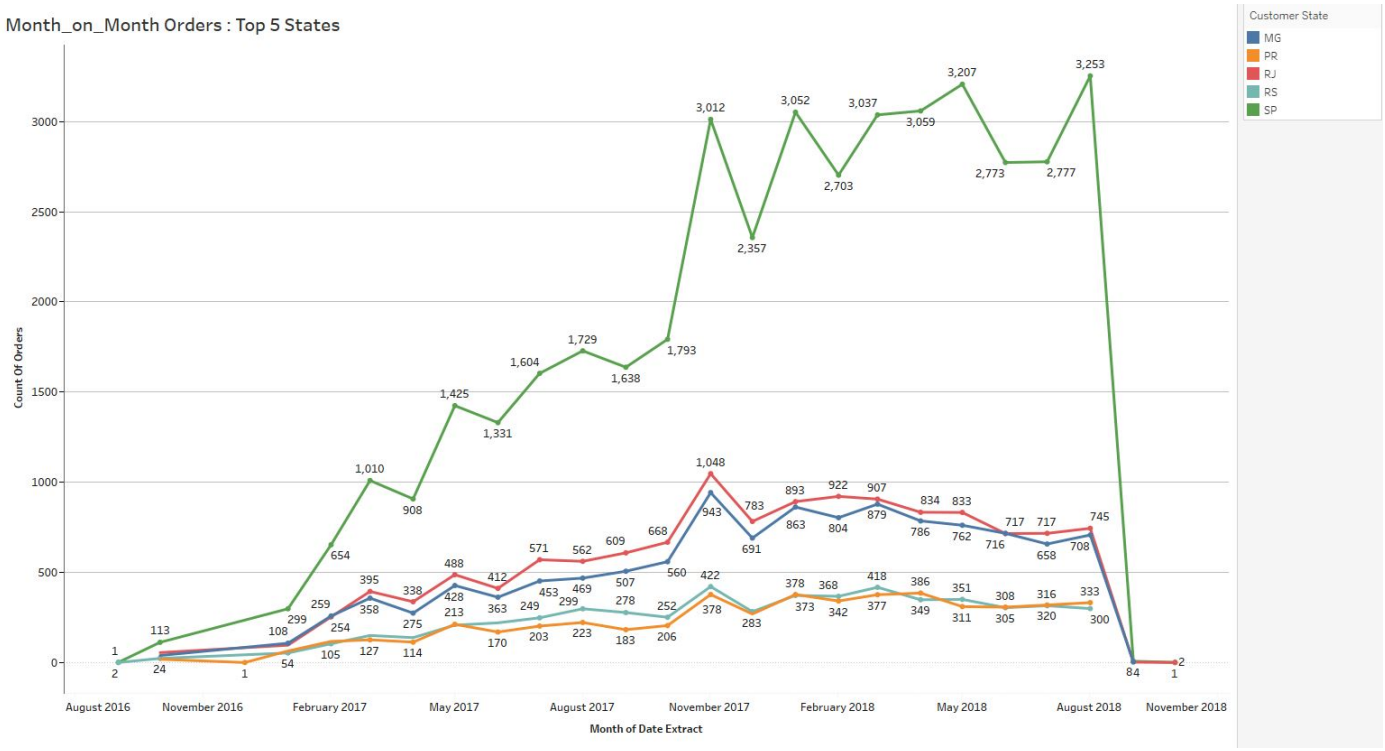
left join `target-retail-64862.target_retail.orders` ord

on ord.customer_id = cust.customer_id

where cust.customer_state IN ('SP','RJ','MG','RS','PR')) as cust_ord

group by cust_ord.customer_state, cust_ord.year_extract, cust_ord.month_extract

order by cust_ord.customer_state, cust_ord.year_extract, cust_ord.month_extract;



Insights:

- Amongst the top 5 states in Brazil, Sao Paolo (SP) had the highest number of orders placed by customers,
followed by Rio de Janeiro (RJ), Belo Horizonte (MG), Porto Alegre (RS) and Curitiba (PR).

- All the states had month on month increase in volume of orders until November 2017 but the growth has either
stagnated or decreased slightly thereafter for all the states until end of August 2018.

- All the customers who purchase/shop/order products from Target are unique over the period of September 2016 to August 2018,
which means none of the existing customer who brought items within the period of September 2016 to August 2018
have come back and made a purchase again with Target Retail.

Recommendations:

- Target has to work on customer retention policy as sales/volume of orders can increase month on month only if
there is a customer retention policy/plan in place for the existing customers. If there is one already in place,
it has to be looked at or tweaked for better sales volume.

- Loyalty Plans can be created based on customers with rich/good history to further stimulate/encourage customers to make
future purchase with Target Retail.

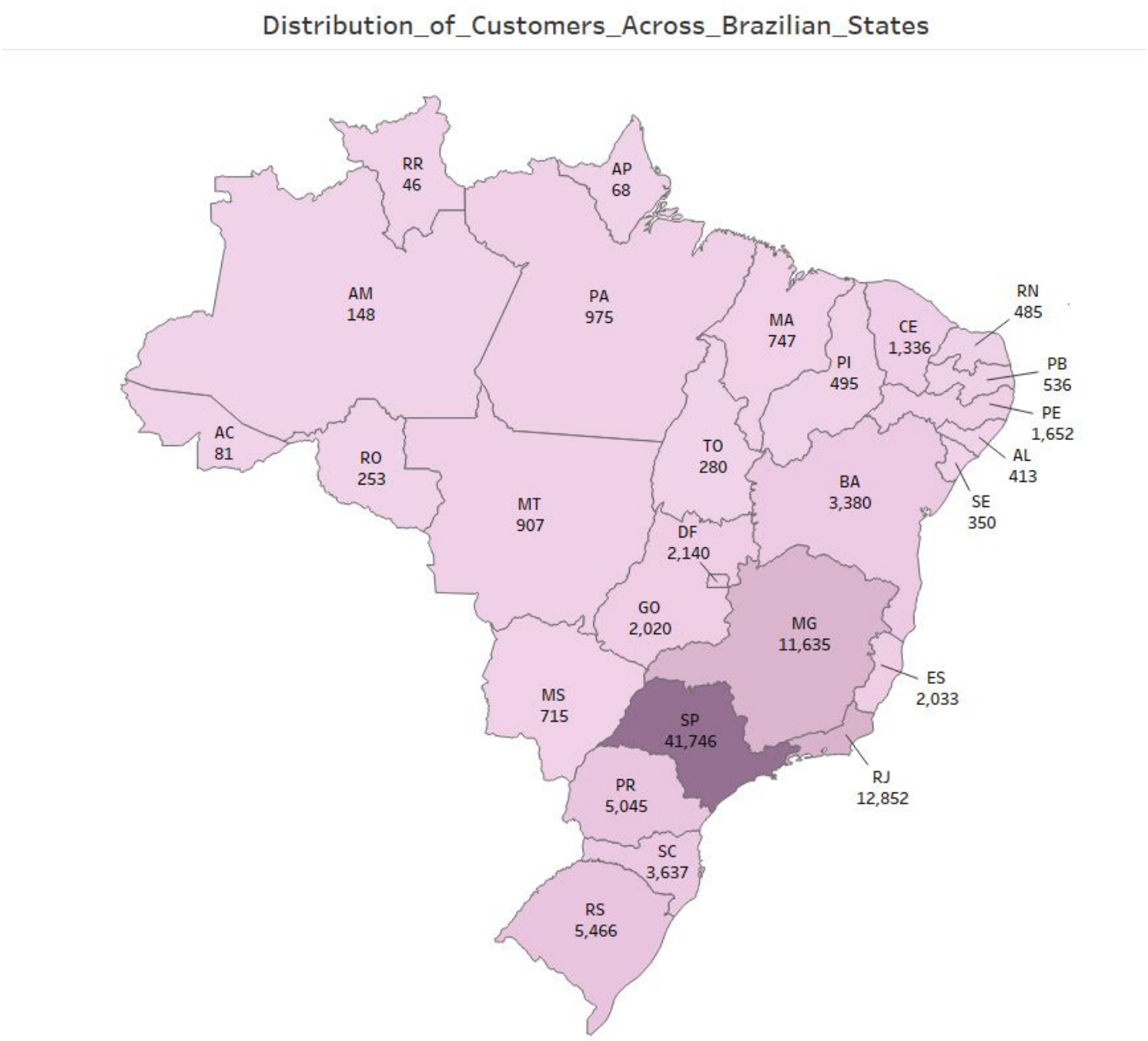
Let's analyze the ditribution of customers in Brazil

query -> select customer_state,

count(*) as state_count from `target-retail-64862.target_retail.customers`

group by customer_state

order by count(*) desc;



Insight:

- Majority of the customers of Target Retail in Brazil are in the South or South East Region.

Impact on Economy: Money movement by e-commerce

by order prices, freight and others.

Observation: In our Initial EDA, we found that there are 98666 distinct orders in order_items table. i.e. 775(99441-98666) customers have no corresponding record entries in order_items table. Hence, we need to observe the mean and the distribution of price and freight_value to impute these missing values.

Imputing of Missing Values for cost of order:

```
 - Let's find the distribution of price in the data and average price to observe
what could be a good measure of
 imputation of missing price values for these 775 customers.

 - We will be dividing the price available for 98666 customers into 4 quartiles t
o see how the price for customers
 are distributed.
```

query -> select

case when bracket.ntile_price = 1 then '1st_Quartile'

when bracket.ntile_price = 2 then '2nd_Quartile'

when bracket.ntile_price = 3 then '3rd_Quartile'

when bracket.ntile_price = 4 then '4th_Quartile'

else ''

end as price_quartile,

min(bracket.price) as min_price,

max(bracket.price) as max_price,

round(max(bracket.price)-min(bracket.price),2) as price_range from

(select price_items.order_id, price_items.price, ntile(4) over (order by price_items.price) as ntile_price from

(select ord.order_id, sum(ord_it.price) as price from `target-retail-64862.target_retail.orders` ord

left join `target-retail-64862.target_retail.order_items` ord_it

on ord_it.order_id = ord.order_id

where ord_it.price is not null

group by ord.order_id) as price_items) as bracket

group by bracket.ntile_price

order by bracket.ntile_price;

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	price_quartile	min_price	max_price	price_range
1	1st_Quartile(Q1)	0.85	45.9	45.05
2	2nd_Quartile(Q2)	45.9	86.9	41.0
3	3rd_Quartile(Q3)	86.9	149.9	63.0
4	4th_Quartile(Q4)	149.9	13440.0	13290.1

So from the above table we see that Inter Quartile Range (IQR) = 149.9 - 45.9 = 104.

So anything beyond - Q3(max_price) + 1.5 * IQR is an outlier

```
= 149.9 + 1.5 * 104
= 149.9 + 156
= 305.9
```

So as per the available price range, anything above 305.9 can be considered an outlier

Count of outliers

query -> select count(*) from

(select ord.order_id, sum(ord_it.price) as price from `target-retail-64862.target_retail.orders` ord

left join `target-retail-64862.target_retail.order_items` ord_it

on ord_it.order_id = ord.order_id

where ord_it.price is not null

group by ord.order_id

having sum(ord_it.price) > 305.9);

So, nearly 7913 customers or nearly 8% of the customers have brought products which are too expensive or not within distribution range of price available in the dataset.

Calculating mean price

query -> select avg(price_items.price) as mean_price from

(select ord.order_id, sum(ord_it.price) as price from `target-retail-64862.target_retail.orders` ord

left join `target-retail-64862.target_retail.order_items` ord_it

on ord_it.order_id = ord.order_id

group by ord.order_id) as price_items;

Mean Price is 137.75

Conclusion: So, the median price i.e 86.9 is a better replacement/imputer for null values in this case as the

ditribution of price for customers is heavily right skewed [Mean (137.75) > Median (86.9)], which large number of outliers (7913 customers) to the far right.

Imputing of Missing Values for freight value:

```
  - Let's find the distribution of freight value along with average freight value
 to observe what could be a good measure of
  imputation of missing freight values for these 775 customers.

  - We will be dividing the freight value available for 98666 customers into 4 qua
rtiles to see how the freight value
  for customers are distributed.
```

query -> select

case when bracket.ntile_freight_value = 1 then '1st_Quartile'

when bracket.ntile_freight_value = 2 then '2nd_Quartile'

when bracket.ntile_freight_value = 3 then '3rd_Quartile'

when bracket.ntile_freight_value = 4 then '4th_Quartile'

else ''

end as freight_value_quartile,

round(min(bracket.freight_value),2) as min_freight_value,

round(max(bracket.freight_value),2) as max_freight_value,

round(max(bracket.freight_value)-min(bracket.freight_value),2) as freight_value_range from

(select freight_value_items.order_id,

freight_value_items.freight_value,

ntile(4) over (order by freight_value_items.freight_value) as

ntile_freight_value from

(select ord.order_id,

sum(ord_it.freight_value) as freight_value from target-retail-64862.target_retail.orders ord

left join target-retail-64862.target_retail.order_items ord_it on ord_it.order_id = ord.order_id

where ord_it.freight_value is not null

group by ord.order_id) as freight_value_items) as bracket

group by bracket.ntile_freight_value

order by bracket.ntile_freight_value;

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	freight_value_quartile	min_freight_value	max_freight_value	freight_value_range
1	1st_Quartile	0.0	13.85	13.85
2	2nd_Quartile	13.85	17.17	3.32
3	3rd_Quartile	17.17	24.04	6.87
4	4th_Quartile	24.04	1794.96	1770.92

So from the above table we see that Inter Quartile Range (IQR) = 24.04 - 13.85 = 10.19.

So anything beyond - Q3(max_freight_value) + 1.5 * IQR is an outlier

```
= 24.04 + 1.5 * 10.19
= 24.04 + 15.285
= 39.325
```

So as per the available freight value range, anything above 39.325 can be considered an outlier

Count of outliers

query -> select count(*) from

(select ord.order_id, sum(ord_it.freight_value) as freight_value from `target-retail-64862.target_retail.orders` ord

left join `target-retail-64862.target_retail.order_items` ord_it

on ord_it.order_id = ord.order_id

where ord_it.freight_value is not null

group by ord.order_id

having sum(ord_it.freight_value) > 39.325);

So, nearly 9941 customers or nearly 10% of the customer orders have freight value that can be considered as outliers

Calculating mean freight value

query -> select avg(freight_value_items.freight_value) as mean_freight_value_items from

(select ord.order_id, sum(ord_it.freight_value) as freight_value from `target-retail-64862.target_retail.orders` ord

left join `target-retail-64862.target_retail.order_items` ord_it

on ord_it.order_id = ord.order_id

group by ord.order_id) as freight_value_items;

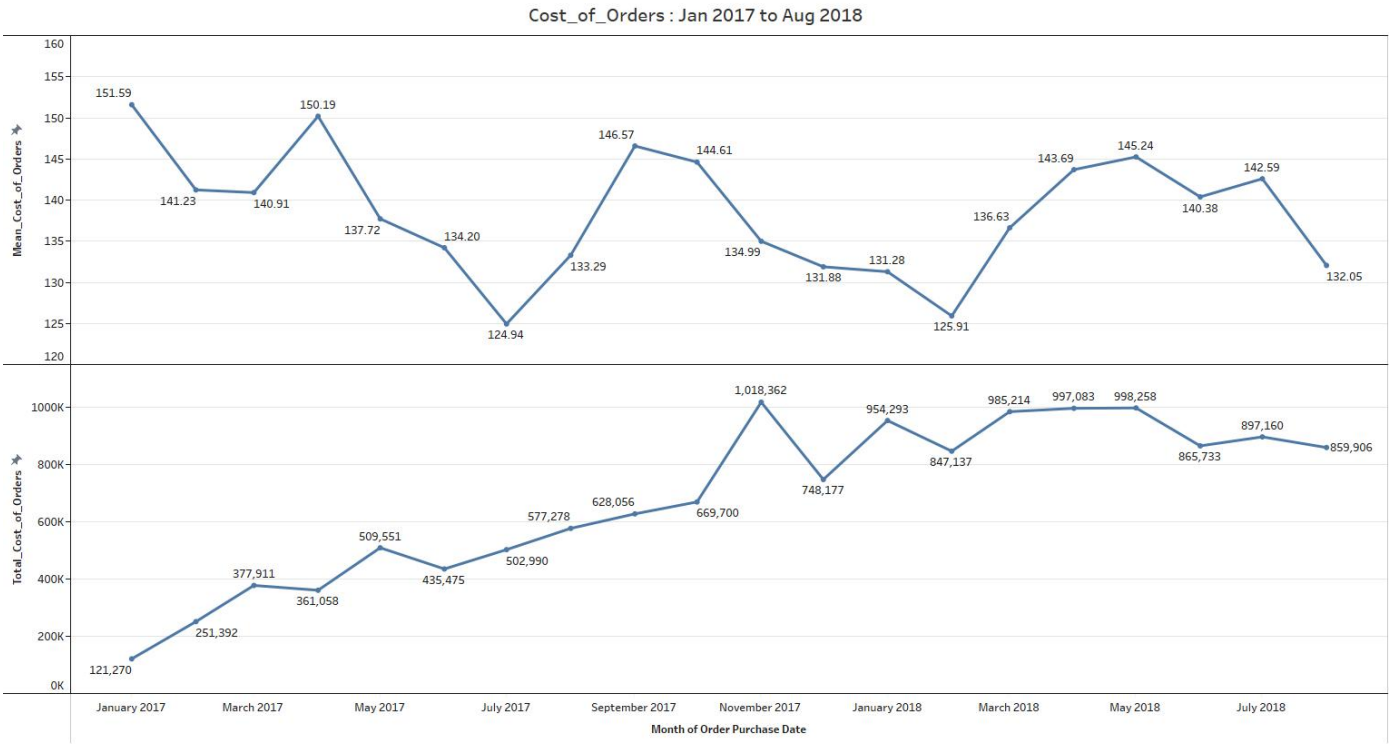
Mean Freight Value is 22.82

Conclusion: So, again the median freight value i.e 17.17 is a better replacement/imputer for null values in this case as the distribution of freight value for orders is right skewed [Mean (22.82) > Median (17.17)], with larger of outliers to the right.

Percentage increase in cost of orders from 2017 to 2018 (Jan 2017 to Aug 2018 only)

query ->

```
select cost_of_orders.year,

cost_of_orders.month,

avg(cost_of_orders.price) as avg_price_per_month,

sum(cost_of_orders.price) as sum_price_per_month from

(select ord.order_id, max(FORMAT_TIMESTAMP("%Y-%m-%d", ord.order_purchase_timestamp)) as order_purchase_date, max(EXTRACT(MONTH from ord.order_purchase_timestamp)) as month, max(EXTRACT(YEAR from ord.order_purchase_timestamp)) as year, sum(coalesce(ord_it.price, 87)) as price

from target-retail-64862.target_retail.orders ord

left join target-retail-64862.target_retail.order_items ord_it

on ord_it.order_id = ord.order_id

where FORMAT_TIMESTAMP("%Y-%m-%d", ord.order_purchase_timestamp) between '2017-01-01' AND '2018-08-31' group by ord.order_id) as cost_of_orders

group by cost_of_orders.year, cost_of_orders.month

order by cost_of_orders.year, cost_of_orders.month;
```



Insights:

- The average cost of orders has gone down between January 2017 (151.59) to August 2018 (132.05) by 13%.

- The total cost of orders has increased considerably between January 2017 to November 2017 by 40%.

- Post November 2017, the total cost of orders has gone down by 15% between November 2017 to August 2018.
Although there were few random fluctuations in cost of orders between November 2017 to August 2018,
but mostly the numbers have been below the cost of orders mark in November 2017.

Let's Analyze Mean & Sum of price and freight value by customer state

query -> select cust.customer_state,

round(avg(coalesce(oi.price,87)),2) as mean_price,

round(sum(coalesce(oi.price,87)),2) as sum_price,

round(avg(coalesce(oi.freight_value,87)),2) as mean_freight,

round(sum(coalesce(oi.freight_value,17.17)),2) as sum_freight

from `target-retail-64862.target_retail.customers` cust

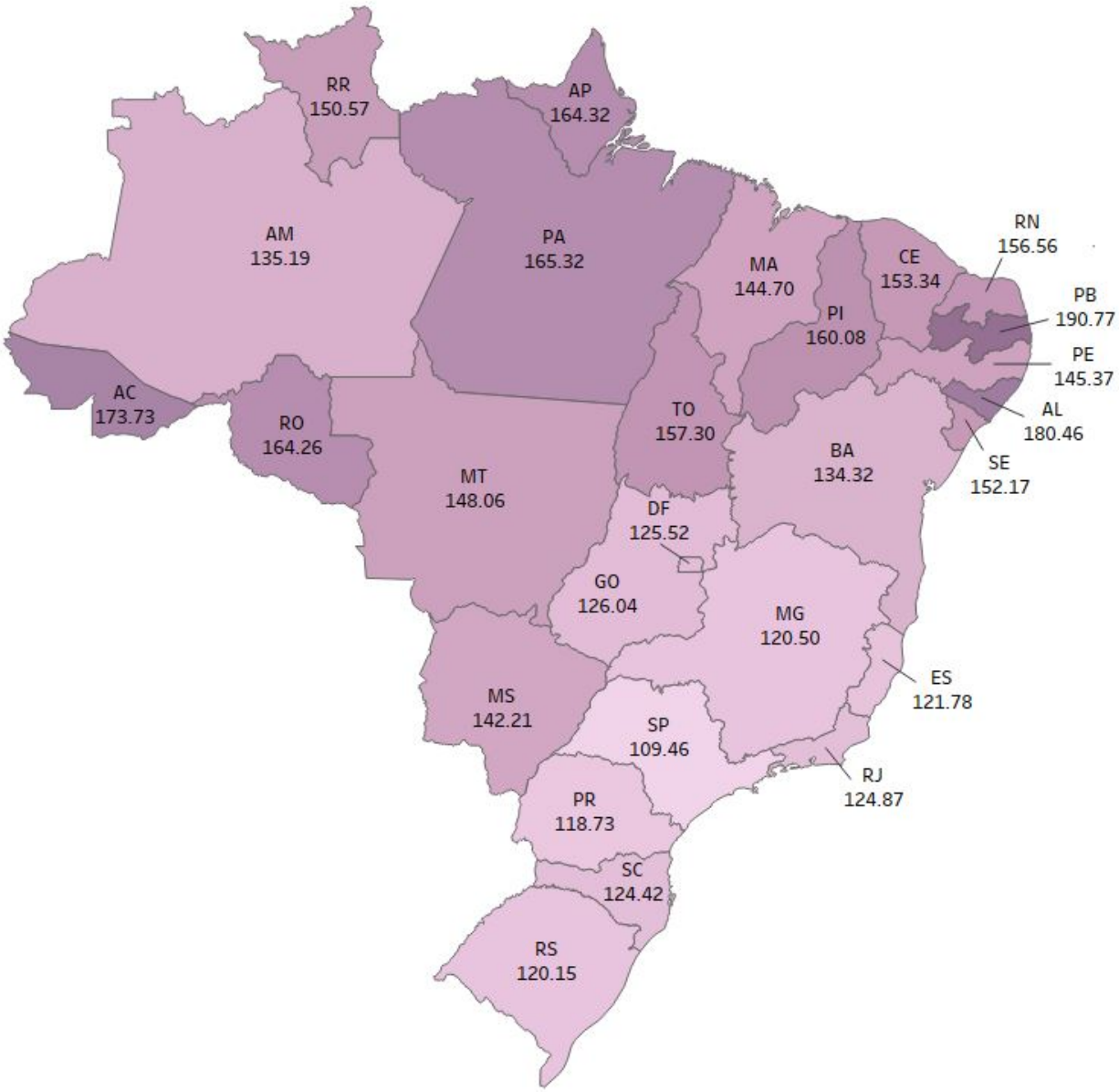
left join `target-retail-64862.target_retail.orders` ord on ord.customer_id = cust.customer_id

left join `target-retail-64862.target_retail.order_items` oi on oi.order_id = ord.order_id

group by cust.customer_state;

Mean Price of Orders Per State

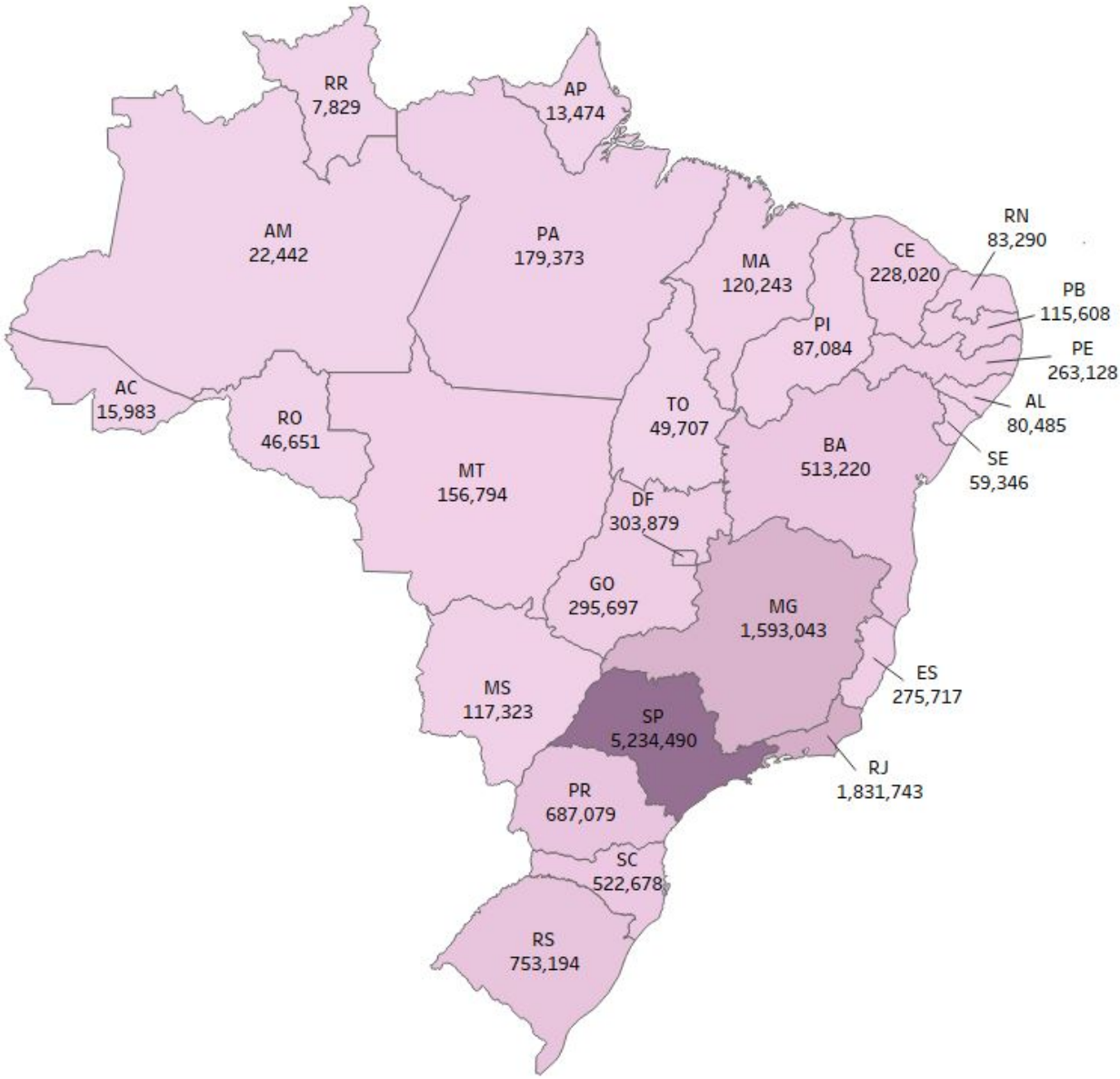
Average_Price_of_Orders_Per_State



Insight:

- Even though the number of orders are less in the North/NorthEast/West Region of Brazil as compared to the states
in the South/SouthEast Region of Brazil, the mean price of orders in the North/NorthEast/West Region of Brazil
is much higher than the mean price of orders in the South/SouthEast Region of Brazil.

Total Price of Orders Per State

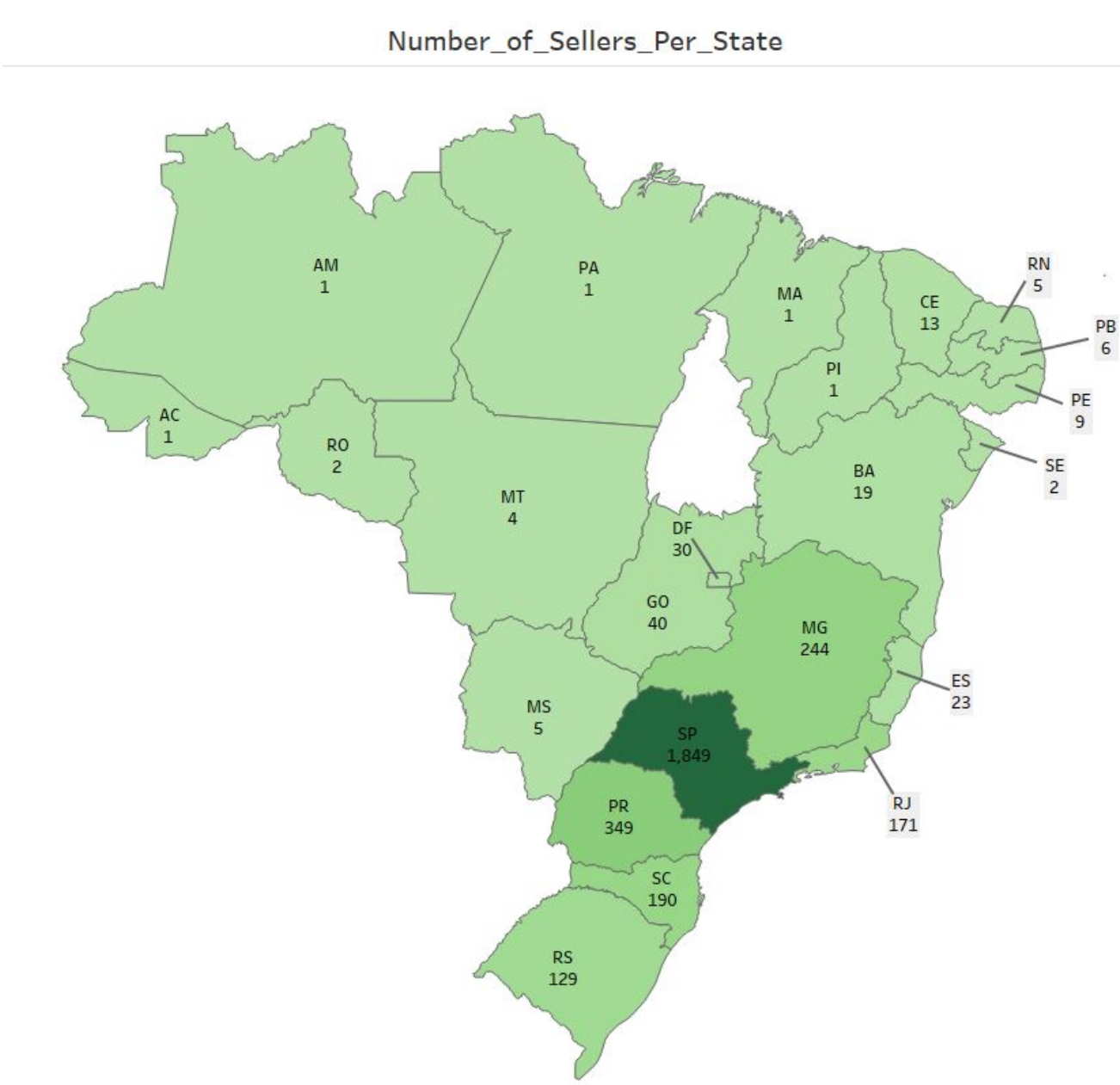


Insights:

- Even though the mean price of orders in the North/NorthEast/West Region of Brazil is much higher than the mean price of orders in the South/SouthEast Region of Brazil, the total price of orders per state in the South/SouthEast Region of Brazil is very high when compared to the total price of orders in the South/SouthEast Region of Brazil.

- The total price of orders in the South/SouthEast Region of Brazil is higher due to the large volume of orders emanating from the South/SouthEast Region.

Number of Sellers Per State



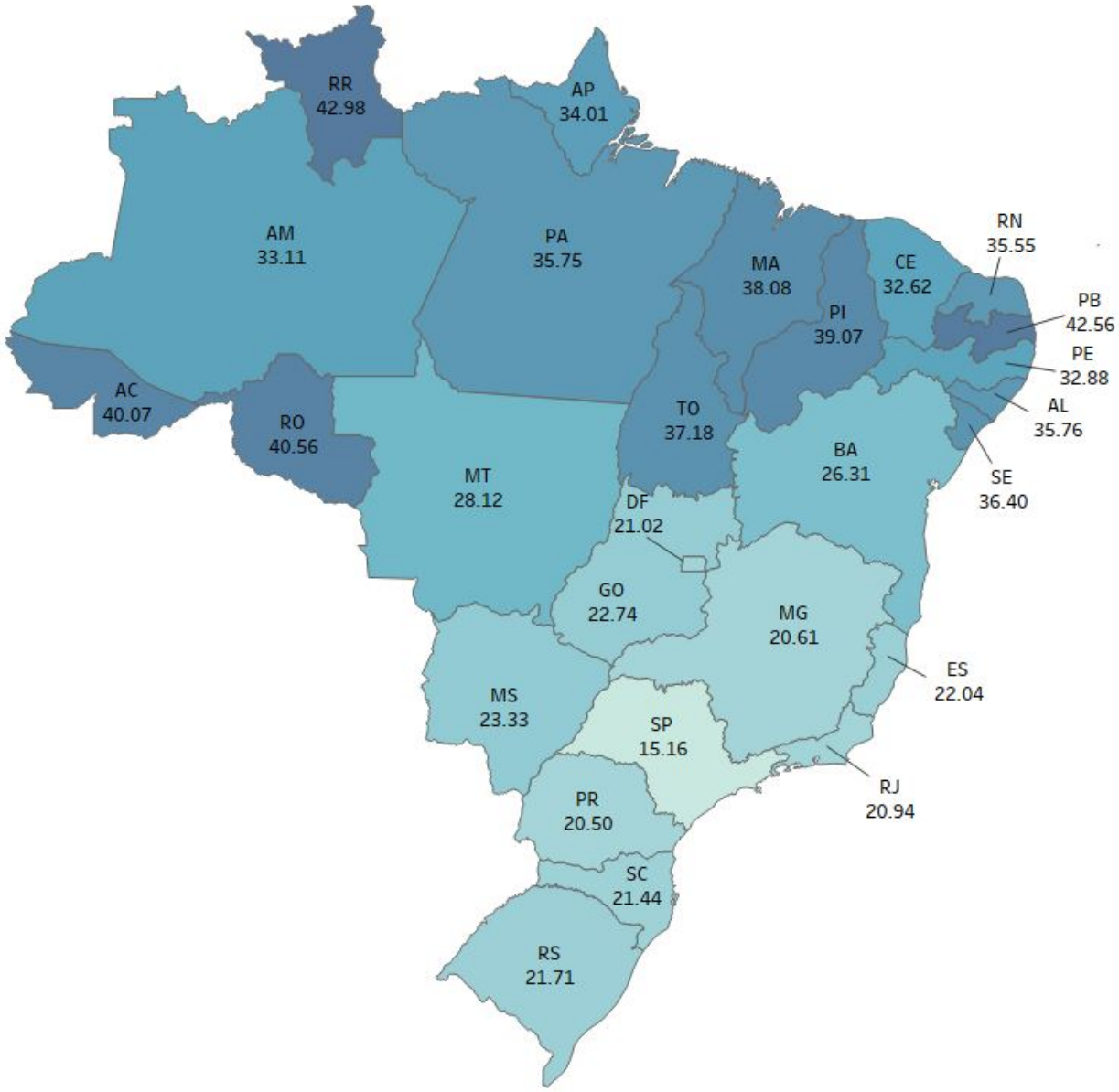
Insights:

- Most of the sellers are located in the South/SouthEast Region of Brazil.

- Fewer sellers are located in the North/West/CenterWest Region of Brazil.

Average Freight Value of Orders Per State

Average_Freight_Value_of_Orders_Per_State



Insights:

- The Average Freight Value of orders in the North/West/CenterWest Region of Brazil is very high (nearly twice)
when compared to the Average Freight Value of orders in the South/SouthEast Region of Brazil.

- The Average Freight Value of orders in the North/West/CenterWest Region of Brazil could be higher due to the
large number of sellers (nearly 90%) located in the South/SouthEast Region of Brazil, which increase
the shipping cost of orders to customers in the North/West/CenterWest Region of Brazil and reduces the shipping cost to
customers in the proximity i.e. (South/SouthEast Region)

- That is also testament to the fact that the average freight cost is higher in the North/West/CenterWest Region of Brazil
in the following order:

- Top 5 (Highest Average Freight Cost) in entire Brazil:

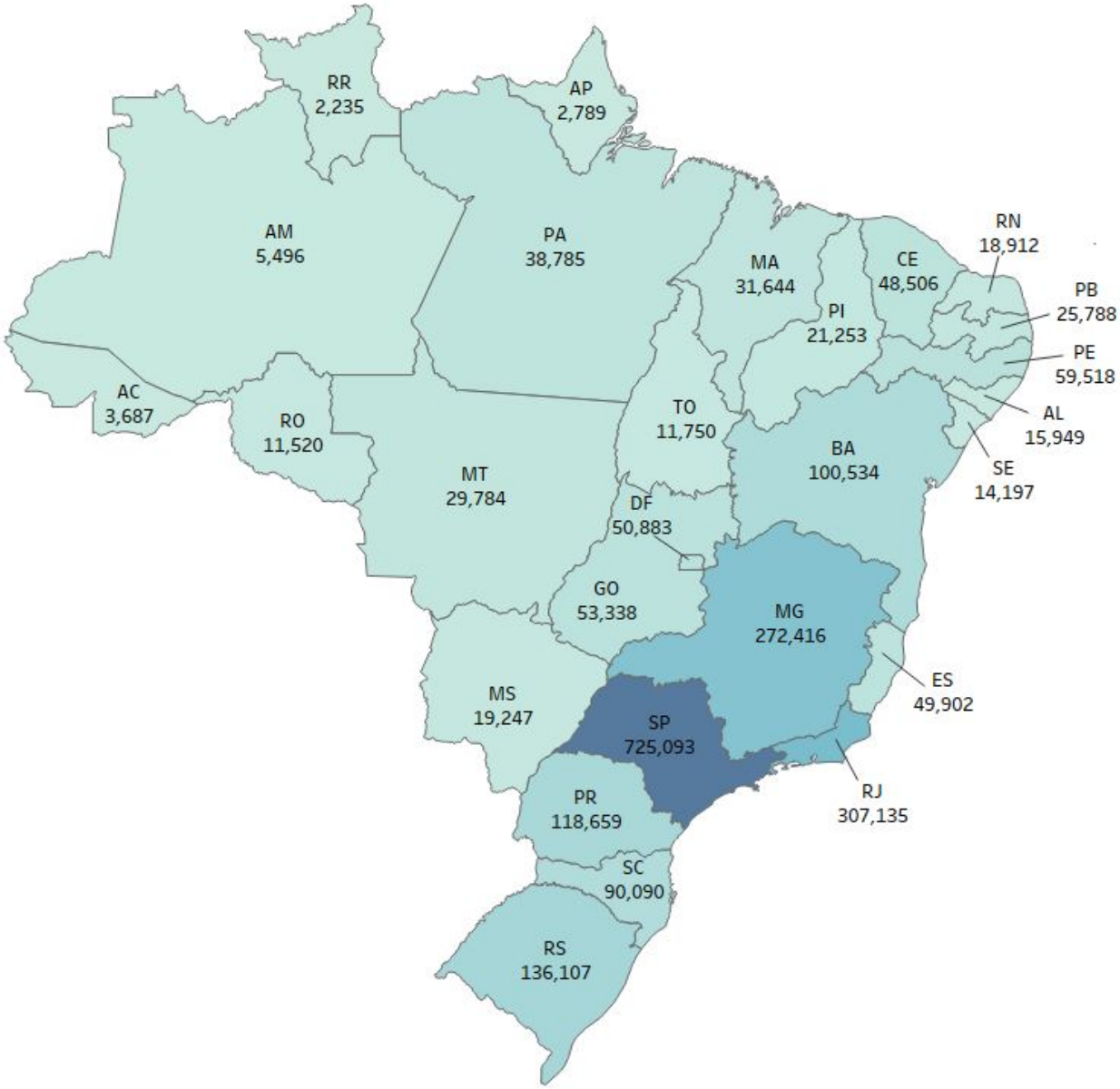
1. RR (Boa Vista) - 42.98
2. PB (João Pessoa) - 42.56
3. RO (Porto Velho) - 40.56
4. AC (Rio Branco) - 40.07
5. PI (Teresina) - 39.07

- Top 5 (Lowest Average Freight Cost) in entire Brazil:

1. SP (Sao Paolo) - 15.16
2. PR (Curitiba) - 20.50
3. MG (Belo Horizonte) - 20.61
4. RJ (Rio de Janeiro) - 20.96
5. DF (Brasília) - 21.02

Total Freight Value of Orders Per State

Average_Freight_Value_of_Orders_Per_State



Insights:

- The total freight value of orders per state in the South/SouthEast Region of Brazil is very high when compared to the total freight value of orders in the North/West/CenterWest Region of Brazil.

- The total freight value of orders in the South/SouthEast Region of Brazil is higher due to the large volume of orders emanating from the South/SouthEast Region.

Recommendations:

```
      - As the average price of orders in the North/West/CenterWest Region of Braz
il is much higher than the
      mean price of orders in the South/SouthEast Region of Brazil, hence increasi
ng the blueprint of sellers
      in the North/West/CenterWest Region of Brazil can contribute to the overall
 increase in volume of orders for
      Target Retail.

      - Target Retail can figure out the best selling products for the customers i
n the North/West/CenterWest Region of Brazil
      and accordingly recommend sellers to stock their shelf with similar/associat
ed products for customers in the area.
```

Operational Efficiency: Timeliness Delivery of Customer Orders

Calculate days between purchasing, delivering and estimated delivery

Creating columns to measure delivery performance indicators:

```
    - time_to_delivery = order_purchase_timestamp - order_delivered_customer_date

    - diff_estimated_delivery = order_estimated_delivery_date - order_delivered_cu
stomer_date
```

Observation: There are 2965 orders for which customer delivery date is missing. As delivery date is purely dependent on operational efficiency we need to treat these entries separately and see what could be a good measure of time_to_delivery and diff_estimated_delivery based on entries where we have data for time_to_delivery and diff_estimated_delivery

query -> select count(*) from `target-retail-64862.target_retail.customers` cust

left join `target-retail-64862.target_retail.orders` ord on ord.customer_id = cust.customer_id

where FORMAT_TIMESTAMP("%Y-%m-%d", ord.order_delivered_customer_date) is null;

Let's first analyze actual delivery time to customer

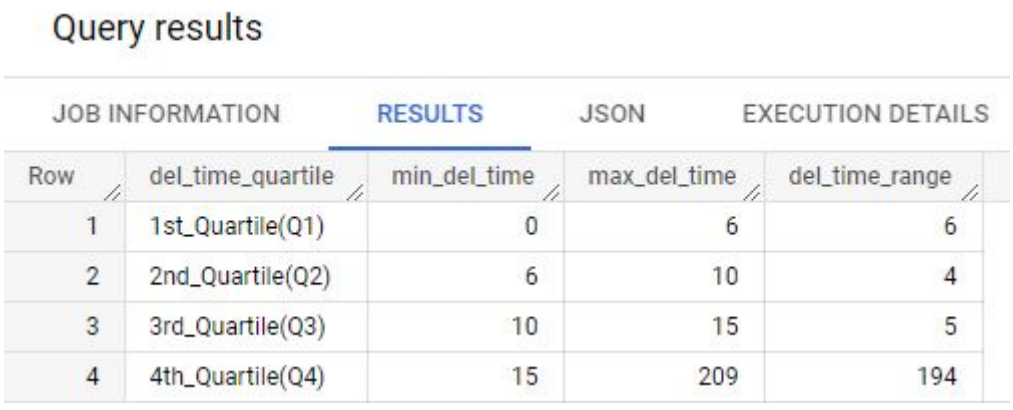
Average delivery time is 12.09 ~ 12 days

query -> select avg(timestamp_diff(order_delivered_customer_date, order_purchase_timestamp, day)) as avg_delivery_time from `target-retail-64862.target_retail.orders` where FORMAT_TIMESTAMP("%Y-%m-%d", order_delivered_customer_date) is not null;

Distribution of actual delivery time:

query -> select

```
case when ntile_data.ntile_del_time = 1 then '1st_Quartile(Q1)'

when ntile_data.ntile_del_time = 2 then '2nd_Quartile(Q2)'

when ntile_data.ntile_del_time = 3 then '3rd_Quartile(Q3)'

when ntile_data.ntile_del_time = 4 then '4th_Quartile(Q4)'

else ''

end as del_time_quartile,

min(ntile_data.time_to_delivery) as min_del_time,

max(ntile_data.time_to_delivery) as max_del_time,

max(ntile_data.time_to_delivery)-min(ntile_data.time_to_delivery) as del_time_range from

(select date_data.customer_state, date_data.time_to_delivery, ntile(4) over(order by
date_data.time_to_delivery) as ntile_del_time from

(select cust.customer_state, FORMAT_TIMESTAMP("%Y-%m-%d", ord.order_purchase_timestamp) as
ord_purch_ts, FORMAT_TIMESTAMP("%Y-%m-%d", ord.order_delivered_customer_date) as ord_del_cust_dt,
timestamp_diff(order_delivered_customer_date, order_purchase_timestamp, day) as time_to_delivery from
 target-retail-64862.target_retail.customers  cust

left join  target-retail-64862.target_retail.orders  ord on ord.customer_id = cust.customer_id

where FORMAT_TIMESTAMP("%Y-%m-%d", ord.order_delivered_customer_date) is not null) as date_data) as
ntile_data

group by ntile_data.ntile_del_time

order by ntile_data.ntile_del_time;
```



Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	del_time_quartile	min_del_time	max_del_time	del_time_range
1	1st_Quartile(Q1)	0	6	6
2	2nd_Quartile(Q2)	6	10	4
3	3rd_Quartile(Q3)	10	15	5
4	4th_Quartile(Q4)	15	209	194

So, the median time to delivery is 10 days

Conclusion: Lesser the delivery time to customers better the operational efficiency, hence let's consider median time to deliver (10 days) as the measure to impute/treat the missing value when compared to the mean value (12 days).

Let's now analyze difference in estimated delivery time to actual delivery time

Average difference in estimated delivery time is -11 days or the order is delivered 11 days before the estimated delivery date

query -> select round(avg(timestamp_diff(order_delivered_customer_date, order_estimated_delivery_date, day)),2) as avg_diff_estimated_delivery

from `target-retail-64862.target_retail.orders`

where FORMAT_TIMESTAMP("%Y-%m-%d", order_delivered_customer_date) is not null;

Distribution of difference in estimated delivery time:

query -> select

case when ntile_data.ntile_est_del_time = 1 then '1st_Quartile(Q1)'

when ntile_data.ntile_est_del_time = 2 then '2nd_Quartile(Q2)'

when ntile_data.ntile_est_del_time = 3 then '3rd_Quartile(Q3)'

when ntile_data.ntile_est_del_time = 4 then '4th_Quartile(Q4)'

else ''

end as del_time_quartile,

min(ntile_data.diff_estimated_delivery) as min_del_time,

max(ntile_data.diff_estimated_delivery) as max_del_time,

max(ntile_data.diff_estimated_delivery)-min(ntile_data.diff_estimated_delivery) as del_time_range from

(select date_data.customer_state, date_data.diff_estimated_delivery, ntile(4) over(order by date_data.diff_estimated_delivery) as ntile_est_del_time from

(select cust.customer_state, FORMAT_TIMESTAMP("%Y-%m-%d", ord.order_estimated_delivery_date) as ord_est_del_dt, FORMAT_TIMESTAMP("%Y-%m-%d", ord.order_delivered_customer_date) as ord_del_cust_dt, timestamp_diff(order_delivered_customer_date, order_estimated_delivery_date, day) as diff_estimated_delivery from `target-retail-64862.target_retail.customers` cust

left join `target-retail-64862.target_retail.orders` ord on ord.customer_id = cust.customer_id

where FORMAT_TIMESTAMP("%Y-%m-%d", ord.order_delivered_customer_date) is not null) as date_data) as ntile_data

group by ntile_data.ntile_est_del_time

order by ntile_data.ntile_est_del_time;

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	diff_est_del_time_quartile	min_del_time	max_del_time	del_time_range
1	1st_Quartile(Q1)	-146	-16	130
2	2nd_Quartile(Q2)	-16	-11	5
3	3rd_Quartile(Q3)	-11	-6	5
4	4th_Quartile(Q4)	-6	188	194

So, the median difference in estimated delivery time is -11 days or the order is delivered 11 days before the estimated delivery date

Conclusion: Since, the mean is equal to median, so -11 days is a good measure to impute/treat the missing value.

Top 5 states with fastest average time to delivery

query -> select cust.customer_state,

round(avg(coalesce(timestamp_diff(order_delivered_customer_date, order_purchase_timestamp, day),10)),2) as avg_time_to_delivery

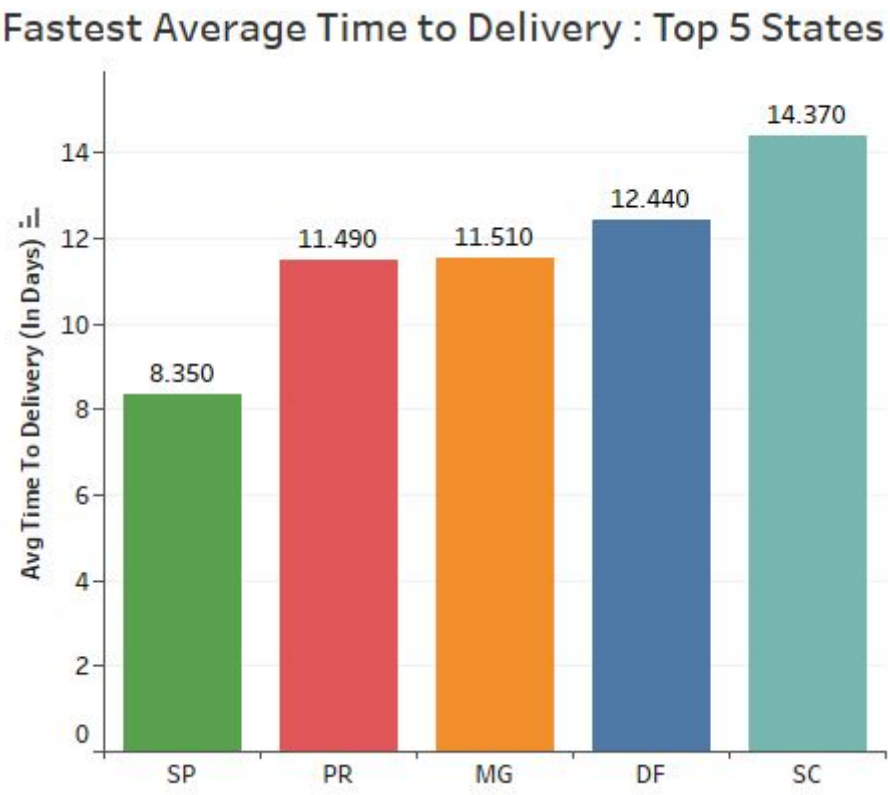
from `target-retail-64862.target_retail.customers` cust

left join `target-retail-64862.target_retail.orders` ord on ord.customer_id = cust.customer_id

group by cust.customer_state

order by avg(coalesce(timestamp_diff(order_delivered_customer_date, order_purchase_timestamp, day),10))

limit 5;



Top 5 states with slowest average time to delivery

query -> select cust.customer_state,

round(avg(coalesce(timestamp_diff(order_delivered_customer_date, order_purchase_timestamp, day),10)),2) as avg_time_to_delivery

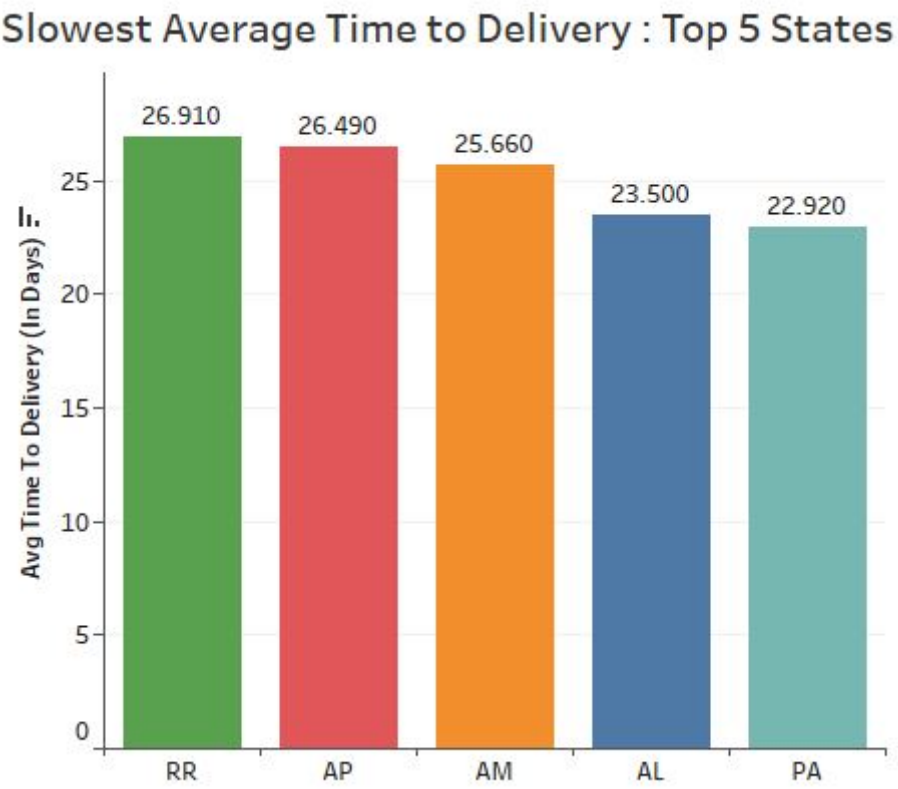
from target-retail-64862.target_retail.customers cust

left join target-retail-64862.target_retail.orders ord on ord.customer_id = cust.customer_id

group by cust.customer_state

order by avg(coalesce(timestamp_diff(order_delivered_customer_date, order_purchase_timestamp, day),10)) desc

limit 5;



Insights:

- The delivery time is faster in South/SouthEast Region of Brazil for Target Retail.

- The delivery time is relatively slower in the North/West/CenterWest Region of Brazil for Target Retail.

- The difference in Average Delivery Time could be due to the presence of large number of retailers in the South/SouthEast Region of Brazil.

Top 5 states where delivery is really fast as compared to estimated date

query -> select cust.customer_state,

round(avg(coalesce(timestamp_diff(order_delivered_customer_date,

order_estimated_delivery_date, day),-11)),2) as avg_diff_estimated_delivery from `target-retail-64862.target_retail.customers` cust

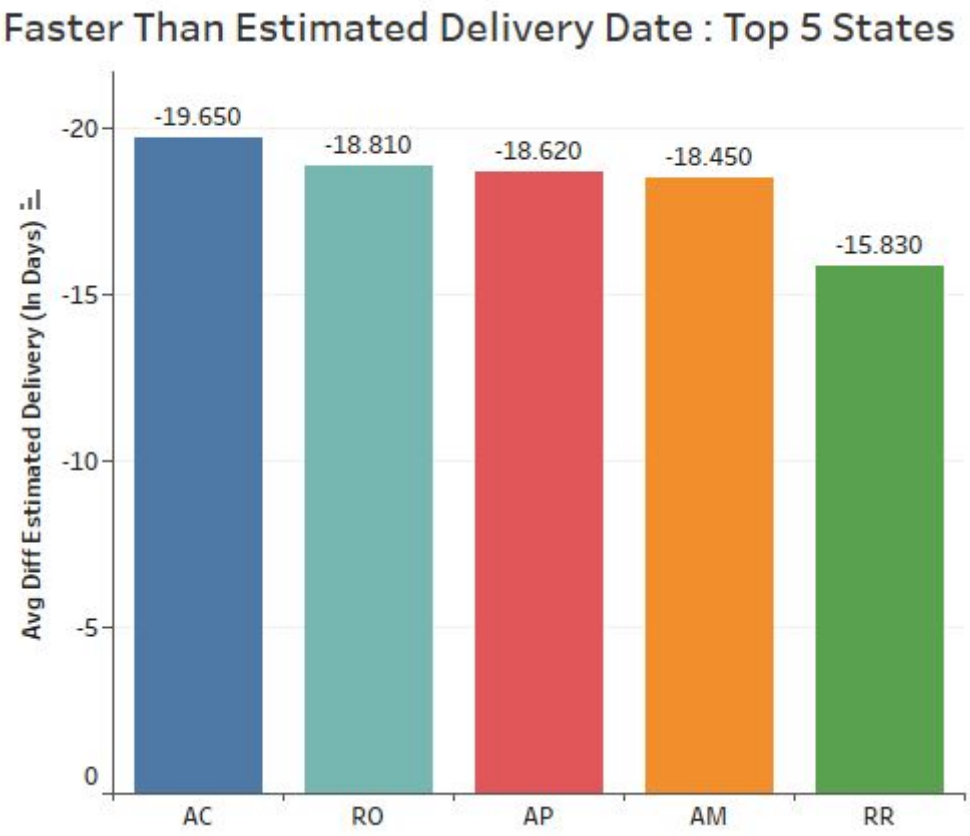
left join `target-retail-64862.target_retail.orders` ord on ord.customer_id = cust.customer_id

group by cust.customer_state

order by avg(coalesce(timestamp_diff(order_delivered_customer_date,

order_estimated_delivery_date, day),-11))

limit 5;



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

query -> select cust.customer_state,

round(avg(coalesce(timestamp_diff(order_delivered_customer_date,

order_estimated_delivery_date, day),-11)),2) as avg_diff_estimated_delivery from `target-retail-64862.target_retail.customers` cust

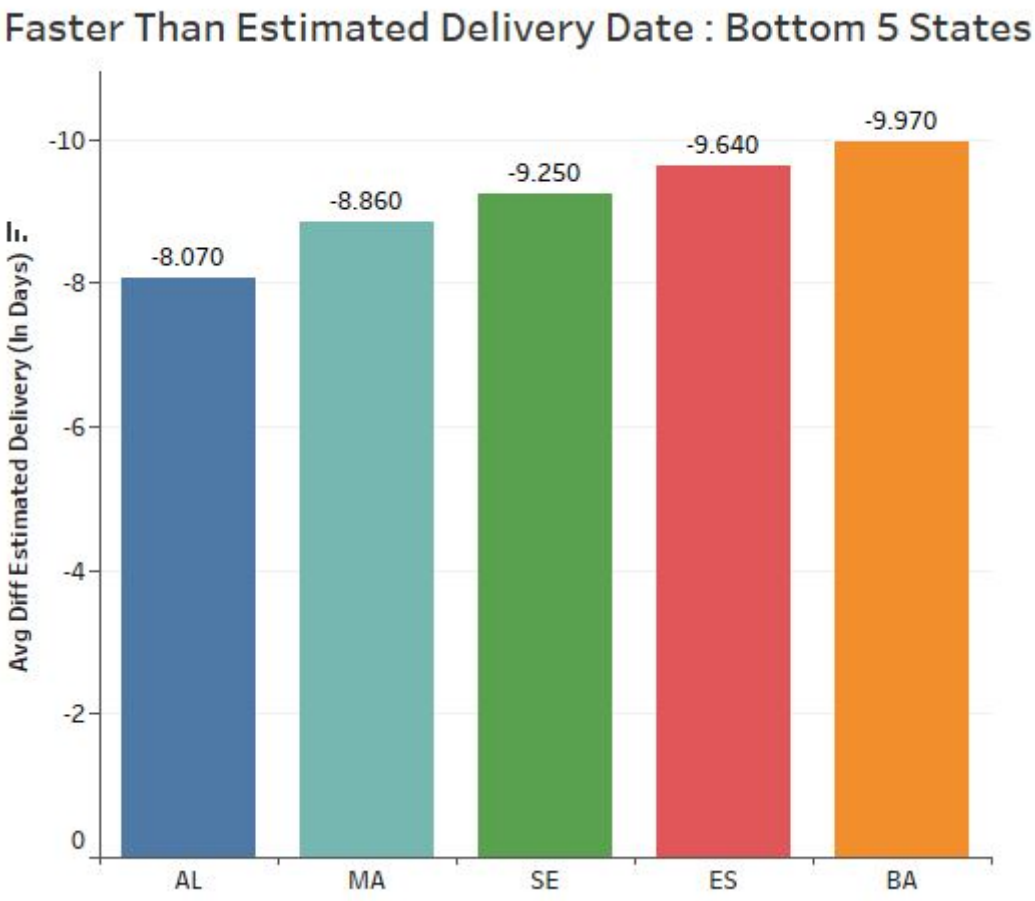
left join `target-retail-64862.target_retail.orders` ord on ord.customer_id = cust.customer_id

group by cust.customer_state

order by avg(coalesce(timestamp_diff(order_delivered_customer_date,

order_estimated_delivery_date, day),-11)) desc

limit 5;



Insight:

- The estimated delivery date is much higher in the North/West/CenterWest Region of Brazil due to fewer retailers in the North/West/CenterWest Region. Hence the fulfillment period for North/West/CenterWest Region of Brazil is also better (in negative) as compared to South/SouthEast Region of Brazil, where the difference between expected delivery date and actual delivery date is much lower (in negative).

Recommendations:

```
      - Average delivery time is very high for most of the states. The best averag
e turn-around time for customer delivery
      is 8.350 days or nearly 8 days for Sao Paolo state. The average turn-around
 time should be very less for customers
      where there are large of retailers in their area (especially South/SouthEast
region of Brazil).

      - 24-hour delivery service should be available for customers buying best sel
ling products in areas where the retailers
      are located in their vicinity.

      - As less or miniscule amount of retailers are present in the North/West/Cen
terWest Region of Brazil,
      the turn-around time for delivery or the estimated delivery time should be l
ess than 7 days in such areas.

      - Pick-up-service for best selling products can be arranged from locations/w
arehouses in the West/North/CenterWest
      region of Brazil where order volumes are less. This will minimize delivery c
ost and increase sales for certain products,
      there by decreasing the turn-around time for delivery of customer orders.
```

Payment Cycles and Preferred Mode of Payments

Let's analyze month over month count of orders for different payment types

Observation: There are certain orders (00bd50cdd31bd22e9081e6e2d5b3577b, 00b4a910f64f24dbcac04fe54088a443, 00c405bd71187154a7846862f585a9d4, 009ac365164f8e06f59d18a08045f6c4...) which are paid through different payment modes and hence the sum of count of orders across different payment types will never be 99440

query -> select order_id,

payment_type,

payment_installments from `target-retail-64862.target_retail.payments`

where order_id in ('00bd50cdd31bd22e9081e6e2d5b3577b','00b4a910f64f24dbcac04fe54088a443','00c405bd71187154a7846862f

order by order_id;

Count of Orders by Different Payment Types

query -> select ord_pay.payment_type,

ord_pay.year, ord_pay.month, count(distinct(ord_pay.order_id)) as count_of_orders from

(select pay.payment_type,

EXTRACT(YEAR from ord.order_purchase_timestamp) as year,

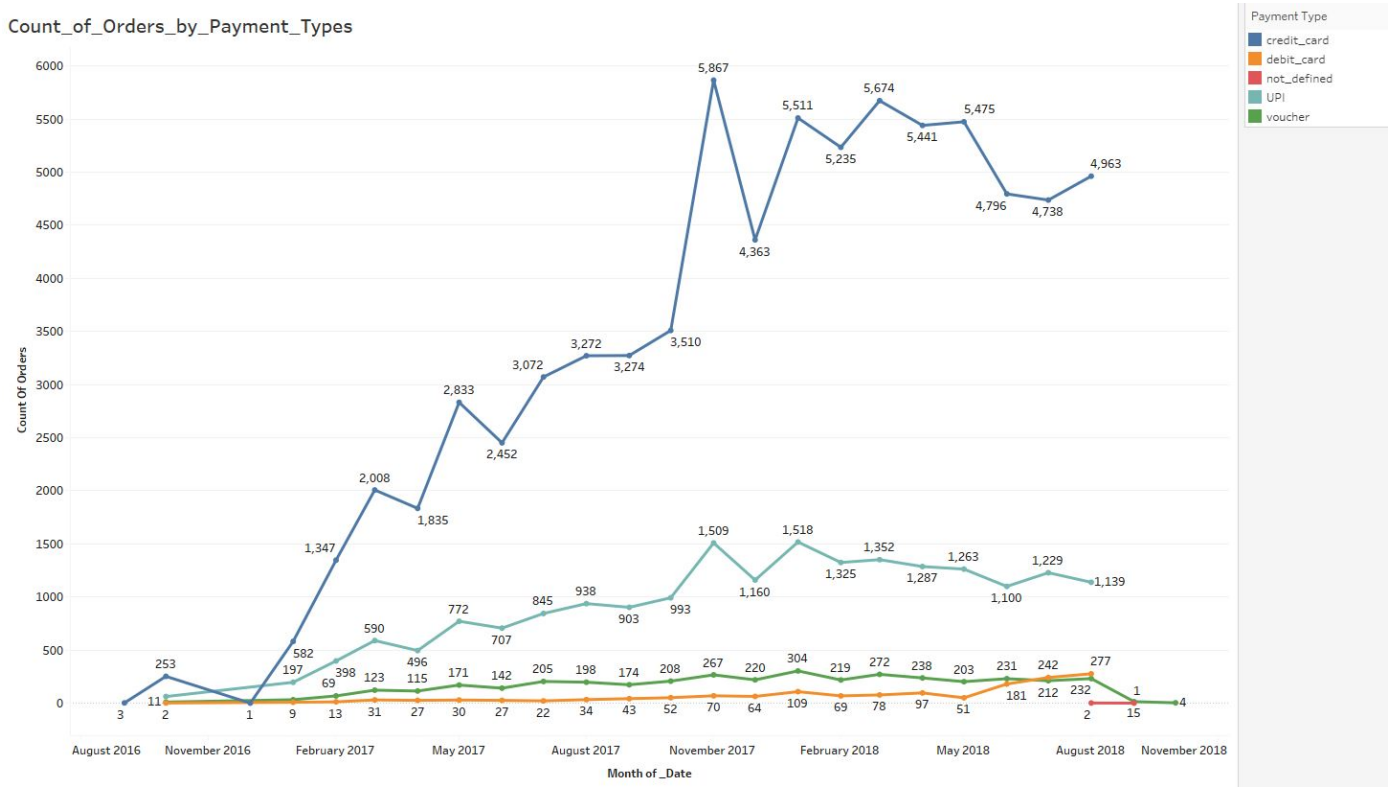
EXTRACT(MONTH from ord.order_purchase_timestamp) as month,

ord.order_id from `target-retail-64862.target_retail.orders` ord

inner join `target-retail-64862.target_retail.payments` pay on pay.order_id = ord.order_id) as ord_pay

group by ord_pay.payment_type, ord_pay.year, ord_pay.month

order by ord_pay.payment_type, ord_pay.year, ord_pay.month;



Insights:

- The most preferred mode of payments is credit card, followed by UPI, voucher and debit card in that order.

- The month on month increase in transactions has increased for almost all modes of payments between January 2017 to
  November 2017 but thereafter there has been no growth in terms of number of payments for any payment mode.
  Infact the number of transactions has decreased from November 2017 to August 2018 across all payment modes
  (exceptions being few random increase in number of transactions for certain months).

Distribution of payment installments and count of orders

query -> select installment.installment_no,

count(*) as installment_counts from

(select order_id, max(payment_installments) as installment_no

from `target-retail-64862.target_retail.payments`

group by order_id

order by order_id) as installment

group by installment.installment_no

order by installment.installment_no;

Distribution of Installments by Count

Installment No	
0	2
1	48,268
2	12,363
3	10,429
4	7,070
5	5,227
6	3,908
7	1,622
8	4,251
9	644
10	5,315
11	23
12	133
13	16
14	15
15	74
16	5
17	8
18	27
20	17
21	3
22	1
23	1
24	18

Payments of Installments by Different Payment Methods

Query ->

select installment.installment_no,

installment.payment_type,

count(*) as installment_counts from

(select order_id, payment_type, payment_installments as installment_no from `target-retail-64862.target_retail.payments` group by order_id, payment_type, payment_installments) as installment

group by installment.installment_no, installment.payment_type

order by installment.installment_no, installment.payment_type;

Preferred Mode of Payment vs Installments

Payment Type	Installment No	
debit_card	1	1,528
not_defined	1	3
UPI	1	19,784
voucher	1	3,866
credit_card	1	25,407
	2	12,389
	3	10,443
	4	7,088
	5	5,234
	6	3,916
	7	1,623
	8	4,253
	9	644
	10	5,315
	11	23
	12	133
	13	16
	14	15
	15	74
	16	5
	17	8
	18	27
	20	17
	21	3
	22	1
	23	1
	24	18

Insights:

- As is evident from the figure above that the preferred mode of payment for customers is credit card when the number of installments is more than equal to 2.

- The number of payments made through debit card is the least when compared to voucher, UPI or credit card payment.

Recommendations:

- Reward points or cashback should be provided for customers if they make transactions through debit card or any online mode of transaction. This may increase the chance of customer coming back and buying products from Target Retail in the future to make use of the accumulated reward point or cashback.

- In addition to that, point of sales options should be provided to customers via their mobile wallets, QR codes, mobile banking or net banking. Providing customers with multiple payment methods can increase revenue for Target Retail.