Problem Statement:

Assuming you are a data analyst/ scientist at Target, you have been assigned the task of analyzing the given dataset to extract valuable insights and provide actionable recommendations.

What does 'good' look like?

1. Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset:
    1. Data type of all columns in the "customers" table.

```
SELECT
    column_name,
    data_type
FROM
    `case.INFORMATION_SCHEMA.COLUMNS`
WHERE
    table_name = 'customers';
```

Row	column_name	data_type
1	customer_id	STRING
2	customer_unique_id	STRING
3	customer_zip_code_prefix	INT64
4	customer_city	STRING
5	customer_state	STRING

    2. Get the time range between which the orders were placed.

```
SELECT
  MIN(order_purchase_timestamp) AS min_purchase_time,
  MAX(order_purchase_timestamp) AS max_purchase_time
FROM  `case.orders`;
```

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

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

```
SELECT
    COUNT(DISTINCT c.customer_city) AS count_of_cities,
    COUNT(DISTINCT c.customer_state) AS count_of_states
FROM
    `case.orders` o
JOIN
    `case.customers` c ON o.customer_id = c.customer_id;
```

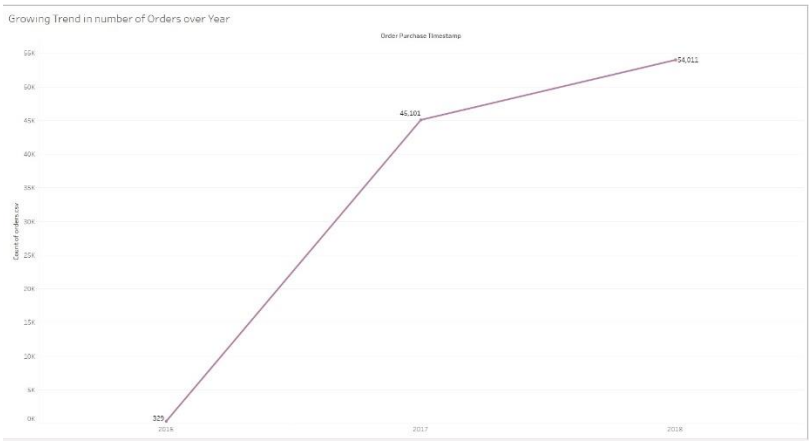
Row	count_of_cities	count_of_states
1	4119	27

2. In-depth Exploration:

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

```
SELECT
EXTRACT(YEAR FROM o.order_purchase_timestamp) AS year,
COUNT(DISTINCT o.order_id) AS count_of_orders
FROM
`       `case.orders` AS o
GROUP BY
        1
ORDER BY 1,2
```

Row	year	count_of_orders
1	2016	329
2	2017	45101
3	2018	54011



Yes,The data illustrates an upward trajectory in the number of orders placed over recent years. Specifically, there has been a marked increase from 329 orders in 2016 to 45,101 orders in 2017, and a subsequent rise to 54,011 orders in 2018. This consistent pattern of growth suggests an expanding customer base or heightened demand, indicating a positive trend in order volume.

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

```
SELECT
```

```
  EXTRACT(MONTH FROM o.order_purchase_timestamp) AS month,
  COUNT(DISTINCT o.order_id) AS count_of_orders
FROM
  `case.orders` AS o
GROUP BY
  1
ORDER BY 1
```

Row	month	count_of_orders
1	1	8069
2	2	8508
3	3	9893
4	4	9343
5	5	10573
6	6	9412
7	7	10318
8	8	10843
9	9	4305
10	10	4959

Yes , a monthly pattern emerges in the order placement, with peaks occurring in May, August, and July, contrasted by lower counts evident in September and October. This observed pattern hints at potential seasonal fluctuations or external factors that may influence consumer behaviour across different months throughout the year.

3. During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)
    - 0-6 hrs : Dawn
    - 7-12 hrs : Mornings

- 13-18 hrs : Afternoon
- 19-23 hrs : Night

```
SELECT
  CASE
    WHEN EXTRACT(HOUR FROM o.order_purchase_timestamp) BETWEEN 0 AND
6 THEN 'Dawn'
    WHEN EXTRACT(HOUR FROM o.order_purchase_timestamp) BETWEEN 7 AND
12 THEN 'Morning'
    WHEN EXTRACT(HOUR FROM o.order_purchase_timestamp) BETWEEN 13 AND
18 THEN 'Afternoon'
    WHEN EXTRACT(HOUR FROM o.order_purchase_timestamp) BETWEEN 19 AND
23 THEN 'Night'
  END AS hour,
  COUNT(o.order_id) AS order_count
FROM
  `case.orders` o
JOIN
  `case.customers` c
ON o.customer_id = c.customer_id
GROUP BY
  1
ORDER BY
  2 desc;
```

Row	hour	order_count
1	Afternoon	38135
2	Night	28331
3	Morning	27733
4	Dawn	5242

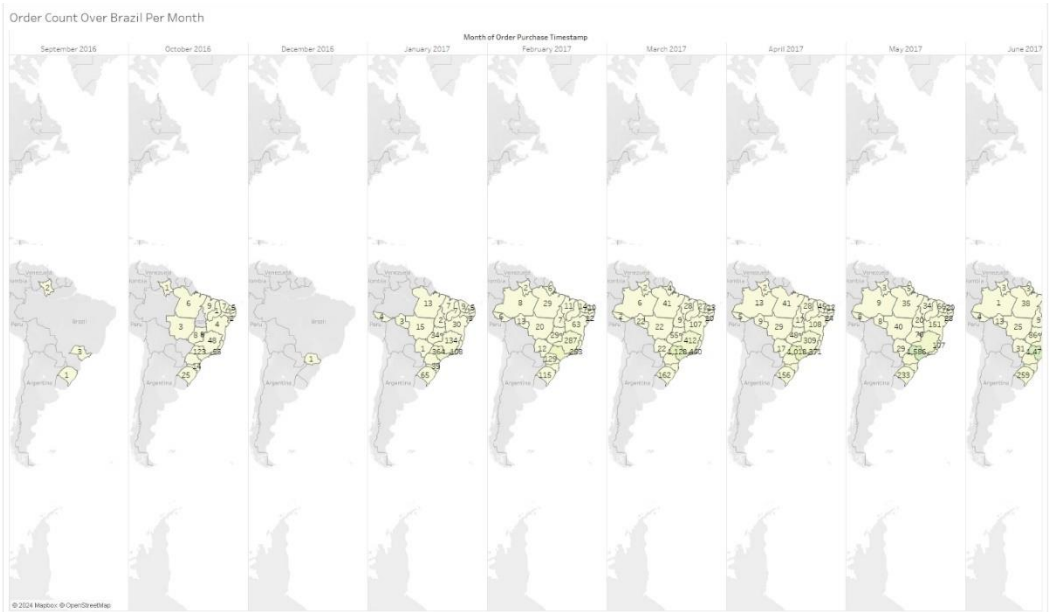
Brazilian customers mostly place their orders during the afternoon, with the highest order count, followed by the night, morning, and dawn hours, in descending order of order count.

3. Evolution of E-commerce orders in the Brazil region:
    1. Get the month on month no. of orders placed in each state.

```
SELECT
    c.customer_state,
    EXTRACT(YEAR FROM o.order_purchase_timestamp) AS year,
    EXTRACT(MONTH FROM o.order_purchase_timestamp) AS month,
    COUNT(o.order_id) AS num_orders
FROM
    case.orders o
JOIN
    case.customers c
ON
    o.customer_id = c.customer_id
GROUP BY
```

```
    1, 2, 3
ORDER BY
    1, 2, 3;
```

Row	customer_state	year	month	num_orders
1	AC	2017	1	2
2	AC	2017	2	3
3	AC	2017	3	2
4	AC	2017	4	5
5	AC	2017	5	8
6	AC	2017	6	4
7	AC	2017	7	5
8	AC	2017	8	4
9	AC	2017	9	5
10	AC	2017	10	6



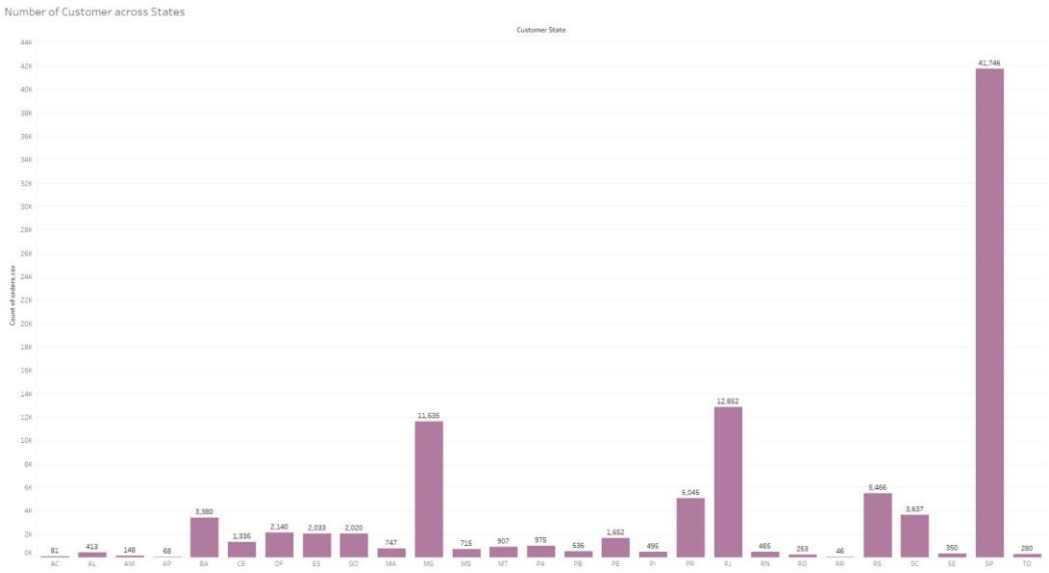
The data depicts the monthly fluctuations in the number of orders placed within each state across several years. These fluctuations suggest dynamic activity levels that may be influenced by factors such as seasonal changes or economic conditions.

2. How are the customers distributed across all the states?

```
SELECT
    customer_state,
    COUNT(customer_id) AS customer_count
FROM
    `case.customers`
GROUP BY
    1
```

```
ORDER BY
    2 DESC
```

Row	customer_state	customer_count
1	SP	41746
2	RJ	12852
3	MG	11635
4	RS	5466
5	PR	5045
6	SC	3637
7	BA	3380
8	DF	2140
9	ES	2033
10	GO	2020



The majority of customers are dispersed across states in Brazil, with São Paulo (SP) boasting the highest customer count, trailed by Rio de Janeiro (RJ) and Minas Gerais (MG). This distribution gradually declines across other states, illustrating differing levels of customer presence across various regions within the country.

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

```sql
SELECT
  EXTRACT(MONTH FROM o.order_purchase_timestamp) AS month,
       ROUND(
  (
    (
      SUM(CASE WHEN EXTRACT(YEAR FROM o.order_purchase_timestamp) =
2018 AND
      EXTRACT(MONTH FROM o.order_purchase_timestamp) BETWEEN 1 AND 8
THEN
      p.payment_value END)
      -
      SUM(CASE WHEN EXTRACT(YEAR FROM o.order_purchase_timestamp) =
2017 AND
      EXTRACT(MONTH FROM o.order_purchase_timestamp) BETWEEN 1 AND 8
THEN
      p.payment_value END)
    )
    /
    SUM(CASE WHEN EXTRACT(YEAR FROM o.order_purchase_timestamp) =
2017 AND
    EXTRACT(MONTH FROM o.order_purchase_timestamp) BETWEEN 1 AND 8
THEN
    p.payment_value END)
  )*100,2) AS percent_increase
FROM
  `case.orders` o
JOIN
  `case.payments` p ON o.order_id = p.order_id
WHERE
  EXTRACT(YEAR FROM o.order_purchase_timestamp) IN (2017, 2018) AND
  EXTRACT(MONTH FROM o.order_purchase_timestamp) BETWEEN 1 AND 8
GROUP BY 1
ORDER BY 1;
```

Row	month	percent_increase
1	1	705.1266954171...
2	2	239.9918145445...
3	3	157.7786066709...
4	4	177.8407701149...
5	5	94.62734375677...
6	6	100.2596912456...
7	7	80.04245463390...
8	8	51.60600520477...

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

```sql
SELECT
  c.customer_state,
  ROUND(AVG(i.price), 2) AS mean_price,
  ROUND(SUM(i.price), 2) AS total_price,
```

```
  FROM
  `case.orders` o
JOIN
  `case.order_items` i ON o.order_id = i.order_id
JOIN
  `case.customers` c ON o.customer_id = c.customer_id
GROUP BY
  c.customer_state;
```

Row	customer_state	mean_price	total_price
1	SP	109.65	5202955.05
2	RJ	125.12	1824092.67
3	PR	119.0	683083.76
4	SC	124.65	520553.34
5	DF	125.77	302603.94
6	MG	120.75	1585308.03
7	PA	165.69	178947.81
8	BA	134.6	511349.99
9	GO	126.27	294591.95
10	RS	120.34	750304.02

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

```
SELECT
  c.customer_state,
  ROUND(AVG(i.freight_value), 2) AS mean_freight_value,
  ROUND(SUM(i.freight_value), 2) AS total_freight_value
  FROM
  `case.orders` o
JOIN
  `case.order_items` i ON o.order_id = i.order_id
JOIN
  `case.customers` c ON o.customer_id = c.customer_id
GROUP BY
  c.customer_state;
```

Row	customer_state	mean_freight_value	total_freight_value
1	MT	28.17	29715.43
2	MA	38.26	31523.77
3	AL	35.84	15914.59
4	SP	15.15	718723.07
5	MG	20.63	270853.46
6	PE	32.92	59449.66
7	RJ	20.96	305589.31
8	DF	21.04	50625.5
9	RS	21.74	135522.74
10	SE	36.65	14111.47

5. Analysis based on sales, freight and delivery time.
    1. Find the no. of days taken to deliver each order from the order's purchase date as delivery time.
    Also, calculate the difference (in days) between the estimated & actual delivery date of an order.
    Do this in a single query.

    You can calculate the delivery time and the difference between the estimated & actual delivery date using the given formula:
        - time_to_deliver = order_delivered_customer_date - order_purchase_timestamp
        - diff_estimated_delivery = order_delivered_customer_date - order_estimated_delivery_date

```
SELECT
  order_id,
  DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY)
  AS delivery,
  DATE_DIFF(order_estimated_delivery_date, order_delivered_customer_date,
DAY)
  AS diffrence_in_days
FROM
  `case.orders`
WHERE
  DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY)
IS NOT NULL
ORDER BY
    1;
```

Row	order_id	delivery	diffrence_in_days
1	00010242fe8c5a6d1ba2dd792...	7	8
2	00018f77f2f0320c557190d7a1...	16	2
3	000229ec398224ef6ca0657da...	7	13
4	00024acbcdf0a6daa1e931b03...	6	5
5	00042b26cf59d7ce69dfabb4e...	25	15
6	00048cc3ae777c65dbb7d2a06...	6	14
7	00054e8431b9d7675808bcb8...	8	16
8	000576fe39319847cbb9d288c...	5	15
9	0005a1a1728c9d785b8e2b08...	9	0

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

```
WITH high_average AS (
  SELECT
    customer_state,
    AVG(freight_value) AS avg_freight
  FROM
    `case.customers` c
  JOIN
    `case.orders` o ON c.customer_id = o.customer_id
  JOIN
    `case.order_items` i ON o.order_id = i.order_id
  GROUP BY
    customer_state
  ORDER BY
    avg_freight DESC
  LIMIT 5
),
low_average AS (
  SELECT
    customer_state,
    AVG(freight_value) AS avg_freight
  FROM
    `case.customers` c
  JOIN
    `case.orders` o ON c.customer_id = o.customer_id
  JOIN
    `case.order_items` i ON o.order_id = i.order_id
  GROUP BY
    customer_state
  ORDER BY
    avg_freight ASC
  LIMIT 5
)

SELECT
  customer_state,
  avg_freight,
  'highest avg freight' AS type
```

```
FROM
  high_average

UNION ALL

SELECT
  customer_state,
  avg_freight,
  'lowest avg freight' AS type
FROM
  low_average;
```

Row	customer_state	avg_freight	type
1	RR	42.98442307692...	highest avg freight
2	PB	42.72380398671...	highest avg freight
3	RO	41.06971223021...	highest avg freight
4	AC	40.07336956521...	highest avg freight
5	PI	39.14797047970...	highest avg freight
6	SP	15.14727539041...	lowest avg freight
7	PR	20.53165156794...	lowest avg freight
8	MG	20.63016680630...	lowest avg freight
9	RJ	20.96092393168...	lowest avg freight

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

```
WITH high_avg AS (
  SELECT
    customer_state,
    ROUND(AVG(DATE_DIFF(order_delivered_customer_date,
order_purchase_timestamp, DAY)), 2) AS avg_d_time
  FROM
    `case.customers` c
  JOIN
    `case.orders` o ON c.customer_id = o.customer_id
  JOIN
    `case.order_items` i ON o.order_id = i.order_id
  GROUP BY
    customer_state
  ORDER BY
    avg_d_time DESC
  LIMIT 5
),
low_avg AS (
  SELECT
    customer_state,
    ROUND(AVG(DATE_DIFF(order_delivered_customer_date,
order_purchase_timestamp, DAY)), 2) AS avg_d_time
  FROM
    `case.customers` c
  JOIN
    `case.orders` o ON c.customer_id = o.customer_id
```

```
  JOIN
    `case.order_items` i ON o.order_id = i.order_id
  GROUP BY
    customer_state
  ORDER BY
    avg_d_time ASC
  LIMIT 5
)

SELECT
  customer_state,
  avg_d_time,
  'Highest Average Delivery' AS type
FROM
  high_avg

UNION ALL

SELECT
  customer_state,
    avg_d_time,
  'Lowest Average Delivery' AS type
FROM
  low_avg;
```

Row	customer_state	avg_d_time	type
1	RR	27.83	Highest Average Delivery
2	AP	27.75	Highest Average Delivery
3	AM	25.96	Highest Average Delivery
4	AL	23.99	Highest Average Delivery
5	PA	23.3	Highest Average Delivery
6	SP	8.26	Lowest Average Delivery
7	PR	11.48	Lowest Average Delivery
8	MG	11.52	Lowest Average Delivery
9	DF	12.5	Lowest Average Delivery
10	SC	14.52	Lowest Average Delivery

4. Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.
   You can use the difference between the averages of actual & estimated delivery date to figure out how fast the delivery was for each state.

```
SELECT
    customer_state,
    round(avg(DATE_DIFF(order_estimated_delivery_date,
order_delivered_customer_date, DAY)))
  AS diffrence_in_days
  FROM
    `case.customers` c
```

```
  JOIN
    `case.orders` o ON c.customer_id = o.customer_id
  JOIN
    `case.order_items` i ON o.order_id = i.order_id
    WHERE
  DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp,
DAY) IS NOT NULL
  GROUP BY
    customer_state
  LIMIT 5
```

Row	customer_state	diffrence_in_days
1	RJ	11.0
2	MG	12.0
3	SC	11.0
4	SP	10.0
5	GO	11.0

6. Analysis based on the payments:
    1. Find the month on month no. of orders placed using different payment types.

```
SELECT
  p.payment_type,
  EXTRACT(MONTH FROM o.order_purchase_timestamp) AS month,
  COUNT(DISTINCT o.order_id) AS count_of_order
FROM
  `case.orders` o
JOIN
  `case.payments` p
ON
  o.order_id = p.order_id
GROUP BY
  1, 2
ORDER BY
  1, 2;
```

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

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

```sql
SELECT
  p.payment_installments,
  COUNT(o.order_id) AS count_order
FROM
  `case.orders` o
JOIN
  `case.payments` p
ON
  o.order_id = p.order_id
WHERE
  o.order_status != 'canceled'
GROUP BY
  1
ORDER BY
  2 DESC;
```

Row	payment_installment	order_count
1	1	52184
2	2	12353
3	3	10392
4	4	7056
5	10	5292
6	5	5209
7	8	4239
8	6	3898
9	7	1620
10	9	638