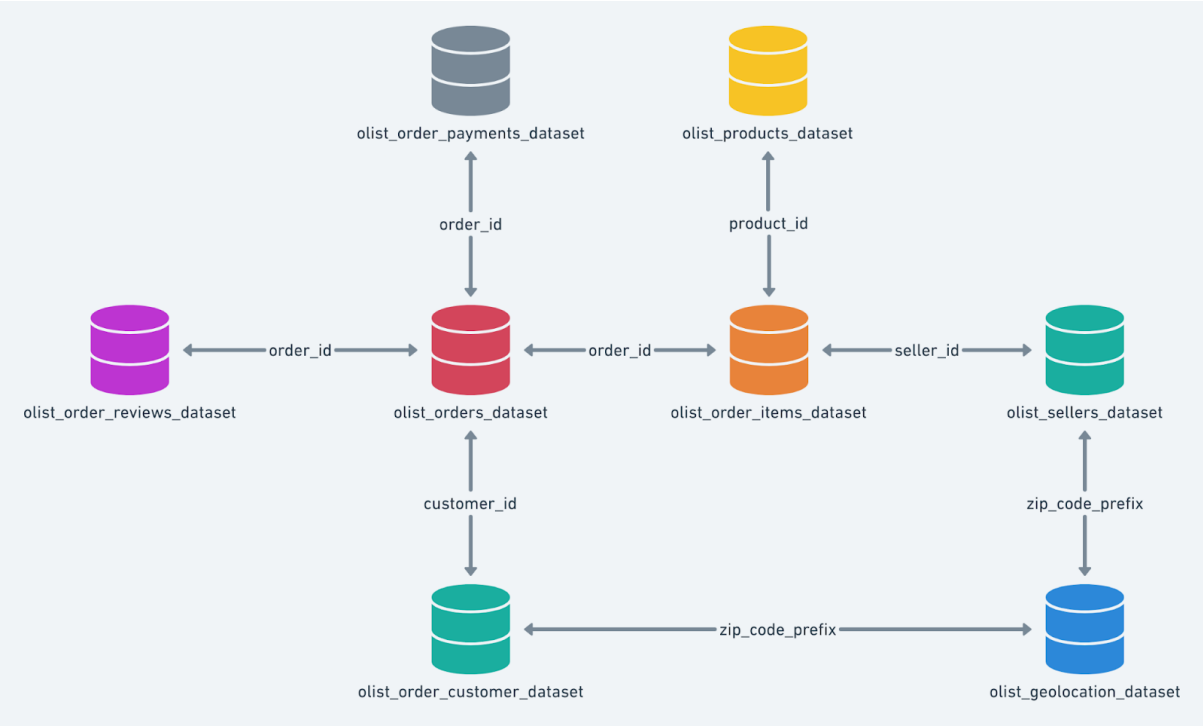
# Target

Target Corporation (doing business as Target and stylized as target) is an American big box department store chain headquartered in Minneapolis, Minnesota. It is the seventh largest retailer in the United States, and a component of the S&P 500 Index. Target was established as the discount division of Dayton's department store of Minneapolis in 1962.

## Problem Statement

The case study is more of an exploration, wherein we have to analyse patterns of customer and seller behaviour based on their geolocation, payment methods, reviews, timely delivery, etc.

### 1. Initial exploration of dataset like checking the characteristics of data




High Level Relationship between tables

#Customers

| Field name | Type | Mode |
|---|---|---|
| customer_id | STRING | NULLABLE |
| customer_unique_id | STRING | NULLABLE |
| customer_zip_code_prefix | INTEGER | NULLABLE |
| customer_city | STRING | NULLABLE |
| customer_state | STRING | NULLABLE |

| | |
|---|---|
| customer_id | ID of the consumer who made the purchase. |
| customer_unique_id | Unique ID of the customer. |
| customer_zip_code_prefix | Zip code of location of the customer. |
| customer_city | Name of the city where order is made. |
| customer_state | State Code from where order is made (Ex-Sao Paulo-SP) |

#Sellers

| Field name | Type | Mode |
|---|---|---|
| seller_id | STRING | NULLABLE |
| seller_zip_code_prefix | INTEGER | NULLABLE |
| seller_city | STRING | NULLABLE |
| seller_state | STRING | NULLABLE |

| | |
|---|---|
| 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) |

#Order_items

| Field name | Type | Mode |
|---|---|---|
| order_id | STRING | NULLABLE |
| order_item_id | INTEGER | NULLABLE |
| product_id | STRING | NULLABLE |
| seller_id | STRING | NULLABLE |
| shipping_limit_date | TIMESTAMP | NULLABLE |
| price | FLOAT | NULLABLE |
| freight_value | FLOAT | NULLABLE |

| | |
|---|---|
| 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. |

#Geolocations

| Field name | Type | Mode |
|---|---|---|
| geolocation_zip_code_prefix | INTEGER | NULLABLE |
| geolocation_lat | FLOAT | NULLABLE |
| geolocation_lng | FLOAT | NULLABLE |
| geolocation_city | STRING | NULLABLE |
| geolocation_state | STRING | NULLABLE |

| | |
|---|---|
| gelocation_zip_code_prefix | First 5 digits of zip code |
| gelocation_lat | Latitude |
| gelocation_lng | Longitude |
| gelocation_city | City name |
| gelocation_state | state |

#Payments

| Field name | Type | Mode |
|---|---|---|
| order_id | STRING | NULLABLE |
| payment_sequential | INTEGER | NULLABLE |
| payment_type | STRING | NULLABLE |
| payment_installments | INTEGER | NULLABLE |
| payment_value | FLOAT | NULLABLE |

| | |
|---|---|
| order_id | A unique id of order made by the consumers. |
| payment_sequential | Sequences of payments made in case of EMI. |
| payment_type | Mode of payment used (Ex-Credit card) |
| payment_installments | #installments in case of EMI Purchase |
| payment_value | Total amount paid for the purchase order. |

#Orders

| Field name | Type | Mode |
|---|---|---|
| order_id | STRING | NULLABLE |
| customer_id | STRING | NULLABLE |
| order_status | STRING | NULLABLE |
| order_purchase_timestamp | TIMESTAMP | NULLABLE |
| order_approved_at | TIMESTAMP | NULLABLE |
| order_delivered_carrier_date | TIMESTAMP | NULLABLE |
| order_delivered_customer_date | TIMESTAMP | NULLABLE |
| order_estimated_delivery_date | TIMESTAMP | NULLABLE |

| | |
|---|---|
| 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_approved_at | Date at which order was approved. |
| 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. |

#Reviews

| Field name | Type | Mode |
|---|---|---|
| review_id | STRING | NULLABLE |
| order_id | STRING | NULLABLE |
| review_score | INTEGER | NULLABLE |
| review_comment_title | STRING | NULLABLE |
| review_creation_date | TIMESTAMP | NULLABLE |
| review_answer_timestamp | TIMESTAMP | NULLABLE |

| | |
|---|---|
| review_id | ID of the review given on the product ordered by the order id. |
| order_id | 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_creation_date | Timestamp of the review when it is created. |
| review_answer_timestamp | Timestamp of the review answered. |

#Products

| Field name | Type | Mode |
|---|---|---|
| product_id | STRING | NULLABLE |
| product_category | STRING | NULLABLE |
| product_name_length | INTEGER | NULLABLE |
| product_description_length | INTEGER | NULLABLE |
| product_photos_qty | INTEGER | NULLABLE |
| product_weight_g | INTEGER | NULLABLE |
| product_length_cm | INTEGER | NULLABLE |
| product_height_cm | INTEGER | NULLABLE |
| product_width_cm | INTEGER | NULLABLE |

| | |
|---|---|
| product_id | A unique identifier for the purposed product. |
| 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 available on the shopping portal. |
| product_phots_qty | #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 in centimeters. |
| product_height_cm | Height of the products in centimeters. |
| product_width_cm | Width of the products in centimeters. |

### The time period for which the data is given

```
SELECT MIN(EXTRACT(YEAR FROM order_purchase_timestamp)) AS start_year,
MAX(EXTRACT(YEAR FROM order_purchase_timestamp)) AS end_year FROM
`target_dataset.orders`;
```

| start_year | end_year |
|---|---|
| 2016 | 2018 |

### Cities and States covered in the city

```
SELECT DISTINCT customer_city, customer_state FROM
  `target_dataset.customers` LIMIT 10;
```

| customer_city | customer_state |
|---|---|
| acu | RN |
| ico | CE |
| ipe | RS |
| ipu | CE |
| ita | SC |
| itu | SP |
| jau | SP |
| luz | MG |
| poa | SP |
| uba | MG |

**The most popular product categories in Target stores**

```
SELECT p.product_category, COUNT(*) as count_order FROM `target_dataset.products` p
WHERE p.product_category IS NOT NULL
GROUP BY p.product_category ORDER BY count_order DESC
LIMIT 10;
```

| product_category | count_order |
|---|---|
| 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 |

**Most frequently, states where orders are purchased.**

```
SELECT  c.customer_state, COUNT(*) AS count_orders FROM `target_dataset.customers` c
GROUP BY c.customer_state
ORDER BY count_orders DESC LIMIT 10;
```

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

**Frequent scores given by Target customers**

```
SELECT r.review_score, COUNT(*) AS review_score_count FROM
`target_dataset.order_reviews` r
GROUP BY r.review_score
ORDER BY review_score_count DESC;
```

| review_score | review_score_count |
|---|---|
| 5 | 57328 |
| 4 | 19142 |
| 1 | 11424 |
| 3 | 8179 |
| 2 | 3151 |

**Conclusions:**

1. The dataset consists of data from 2016 to 2018.
2. Different states and cities in Brazil are covered in the dataset.
3. bed table bath is the most popular product in Target, followed by sport leisure.
4. SP is the most frequent state where the orders are coming from, followed by RJ and MG.
5. Most of the customers have given a 5 star rating to a product, followed by 4 rating.

## 2. In-depth Exploration

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?

**Monthly analysis of average purchases by the customers.**

```
SELECT * FROM
(SELECT DISTINCT FORMAT_DATE("%b", o.order_purchase_timestamp) month,
ROUND(AVG(oi.price) OVER( PARTITION BY FORMAT_DATE("%b", o.order_purchase_timestamp)
ORDER BY FORMAT_DATE("%b", o.order_purchase_timestamp)), 2) AS average_purchases FROM
`target_dataset.orders` o
JOIN `target_dataset.order_items` oi ON o.order_id=oi.order_id) AS x
ORDER BY x.average_purchases DESC;
```

| month | average_purchases |
|---|---|
| Sep | 129.15 |
| Apr | 127.27 |
| Oct | 125.55 |
| May | 124.58 |
| Jun | 121.77 |
| Mar | 121.03 |
| Jul | 120.02 |
| Dec | 117.91 |
| Aug | 117.51 |
| Jan | 116.81 |
| Nov | 116.59 |
| Feb | 113.42 |

**Daily analysis of average purchases by the customers.**

```
SELECT * FROM
(SELECT DISTINCT EXTRACT(DAY FROM o.order_purchase_timestamp) day, ROUND(AVG(oi.price)
OVER( PARTITION BY EXTRACT(DAY FROM o.order_purchase_timestamp) ORDER BY EXTRACT(DAY
FROM o.order_purchase_timestamp)), 2) AS average_purchases FROM `target_dataset.orders`
o
JOIN `target_dataset.order_items` oi ON o.order_id=oi.order_id) AS x
ORDER BY x.average_purchases DESC;
```

| day | average_purchases |
|---|---|
| 1 | 128.53 |
| 4 | 127.19 |
| 18 | 127.12 |
| 31 | 126.99 |
| 29 | 126.91 |
| 11 | 126.54 |
| 9 | 125.43 |
| 10 | 125.08 |
| 28 | 124.87 |
| 6 | 124.35 |
| 7 | 123.71 |
| 3 | 123.4 |
| 12 | 122.84 |
| 16 | 121.14 |
| 2 | 120.93 |
| 5 | 120.65 |
| 20 | 120.31 |
| 15 | 119.75 |
| 14 | 119.5 |
| 13 | 119.32 |
| 17 | 118.71 |
| 30 | 118.53 |
| 24 | 117.16 |
| 19 | 116.48 |
| 25 | 116.41 |
| 8 | 116.28 |
| 27 | 116.23 |
| 23 | 114.42 |
| 26 | 113.76 |
| 21 | 112.05 |
| 22 | 110.13 |

**Weekly analysis of average purchases by the customers.**

```
SELECT * FROM
(SELECT DISTINCT FORMAT_DATE("%a", o.order_purchase_timestamp) week,
ROUND(AVG(oi.price) OVER( PARTITION BY FORMAT_DATE("%A", o.order_purchase_timestamp)
ORDER BY FORMAT_DATE("%a", o.order_purchase_timestamp)), 2) AS average_purchases FROM
`target_dataset.orders` o
JOIN `target_dataset.order_items` oi ON o.order_id=oi.order_id) AS x
ORDER BY x.average_purchases DESC;
```

| week | average_purchases |
|---|---|
| Sat | 123.6 |
| Fri | 122.35 |
| Mon | 121.29 |
| Thu | 120.18 |
| Wed | 120.1 |
| Tue | 119.13 |
| Sun | 118.46 |

**Hourly analysis of average purchases by the customers**

```
SELECT * FROM
(SELECT DISTINCT FORMAT_DATE("%H", o.order_purchase_timestamp) hour,
ROUND(AVG(oi.price) OVER( PARTITION BY FORMAT_DATE("%H", o.order_purchase_timestamp)
ORDER BY FORMAT_DATE("%H", o.order_purchase_timestamp)), 2) AS average_purchases FROM
`target_dataset.orders` o
JOIN `target_dataset.order_items` oi ON o.order_id=oi.order_id) AS x
ORDER BY x.average_purchases DESC;
```

| hour | average_purchases |
|---|---|
| 18 | 125.39 |
| 14 | 125.35 |
| 09 | 124.77 |
| 12 | 124.24 |
| 20 | 124.2 |
| 19 | 122.61 |
| 15 | 122.56 |
| 16 | 122.21 |
| 21 | 121.14 |
| 22 | 120.38 |
| 17 | 119.67 |
| 10 | 118.76 |
| 11 | 117.8 |
| 13 | 117.75 |
| 00 | 116.61 |
| 01 | 116.38 |
| 08 | 115.94 |
| 23 | 113.54 |
| 03 | 113.14 |

| | |
|---|---|
| 07 | 111.05 |
| 06 | 102.9 |
| 05 | 102.73 |
| 04 | 99.51 |
| 02 | 93.89 |

**What time do Brazillian customers tend to buy (Afternoon, Morning, Evening, Night)**

```
SELECT
CASE
WHEN EXTRACT(HOUR FROM o.order_purchase_timestamp) BETWEEN 5 AND 11 THEN "Morning"
WHEN EXTRACT(HOUR FROM o.order_purchase_timestamp) BETWEEN 12 AND 16 THEN "Afternoon"
WHEN EXTRACT(HOUR FROM o.order_purchase_timestamp) BETWEEN 17 AND 21 THEN "Evening"
ELSE "Night"
END AS time_of_day, COUNT(*) AS items_purchased FROM `target_dataset.orders` o
GROUP BY time_of_day
ORDER BY items_purchased DESC;
```

| time_of_day | items_purchased |
|---|---|
| Afternoon | 32211 |
| Evening | 30311 |
| Morning | 22428 |
| Night | 14491 |

```
SELECT
CASE
WHEN FORMAT_DATE("%a", o.order_purchase_timestamp) IN ('Sat', 'Sun') THEN "Weekends"
ELSE "Weekday"
END AS week, COUNT(*) AS items_purchased FROM `target_dataset.orders` o
GROUP BY week
ORDER BY items_purchased DESC;
```

| week | items_purchased |
|---|---|
| Weekday | 76594 |
| Weekends | 22847 |

**Observations:**

1. We can see that September has the highest average spend, making it $129, followed by April and October, respectively.
2. November and February have the lowest average number of purchases.
3. On the 1st of every month, the highest average purchase is $128.53, followed by the 4th and 18th , respectively.
4. The average price in the 22nd and 21st districts is the lowest.
5. The highest average purchase is on Saturday, resulting in $123.6, followed by Friday and Monday.
6. The least average purchase is made on Sunday and Wednesday.
7. The highest average purchase is during the 18th hour, resulting in $125.39, followed by the 14th and 9th.
8. The lowest average purchases are during the 2nd and 4th hours, respectively.

9. The majority of the customers prefer to buy during the afternoon, while very few customers prefer to buy during the night.
10. Customers prefer to buy during the work week instead of the weekend.

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

### Month-on-month orders by states

```
SELECT *, ROUND(LAG(sales) OVER(PARTITION BY state ORDER BY month), 2) AS
prev_month_sales,
        ((x.sales - ROUND(LAG(sales) OVER(PARTITION BY state ORDER BY month), 2)) /
(ROUND(LAG(sales, 1) OVER(PARTITION BY state ORDER BY month), 2))) * 100 as
month_on_month_growth FROM
(SELECT c.customer_state AS state, EXTRACT(MONTH FROM o.order_purchase_timestamp) as
month, ROUND(SUM(oi.price), 2) AS sales,
        FROM `target_dataset.orders` o
        JOIN `target_dataset.customers` c ON o.customer_id=c.customer_id
        JOIN `target_dataset.order_items` oi ON o.order_id=oi.order_id
        GROUP BY EXTRACT(MONTH FROM o.order_purchase_timestamp), c.customer_state) x
        ORDER BY x.state ASC, x.month ASC LIMIT 24;
```

| state | month | sales | prev_month_sales | month_on_month_growth |
|---|---|---|---|---|
| AC | 1 | 1472.54 | | |
| AC | 2 | 652.23 | 1472.54 | -55.707145476523557 |
| AC | 3 | 553.79 | 652.23 | -15.092835349493287 |
| AC | 4 | 1585.36 | 553.79 | 186.27458061720145 |
| AC | 5 | 3459.28 | 1585.36 | 118.20154412877835 |
| AC | 6 | 813.68 | 3459.28 | -76.478342314007534 |
| AC | 7 | 2088.66 | 813.68 | 156.69304886441847 |
| AC | 8 | 1084.62 | 2088.66 | -48.071012036425273 |
| AC | 9 | 1786.89 | 1084.62 | 64.74802234884109 |
| AC | 10 | 735.2 | 1786.89 | -58.8558892824964 |
| AC | 11 | 649.92 | 735.2 | -11.59956474428728 |
| AC | 12 | 1100.78 | 649.92 | 69.371614967996067 |
| AL | 1 | 5528.05 | | |
| AL | 2 | 7209.1 | 5528.05 | 30.409457222709641 |
| AL | 3 | 8170.89 | 7209.1 | 13.341332482556767 |
| AL | 4 | 10824.25 | 8170.89 | 32.4733290987885 |
| AL | 5 | 8430.33 | 10824.25 | -22.11626671593875 |
| AL | 6 | 4682.25 | 8430.33 | -44.459469558131175 |
| AL | 7 | 6292.32 | 4682.25 | 34.3866730738427 |
| AL | 8 | 11225.35 | 6292.32 | 78.397633941058317 |
| AL | 9 | 4365.29 | 11225.35 | -61.112214763904916 |
| AL | 10 | 5426.85 | 4365.29 | 24.318201081715085 |
| AL | 11 | 5936.26 | 5426.85 | 9.3868450390189491 |
| AL | 12 | 2223.87 | 5936.26 | -62.537523625986736 |

### Month-on-month orders by region

```
SELECT *, ROUND(LAG(sales) OVER(PARTITION BY city ORDER BY month), 2) AS
prev_month_sales,
        ((x.sales - ROUND(LAG(sales) OVER(PARTITION BY city ORDER BY month), 2)) /
(ROUND(LAG(sales, 1) OVER(PARTITION BY city ORDER BY month), 2))) * 100 as
month_on_month_growth FROM
(SELECT c.customer_city AS city, EXTRACT(MONTH FROM o.order_purchase_timestamp) as
month, ROUND(SUM(oi.price), 2) AS sales,
        FROM `target_dataset.orders` o
        JOIN `target_dataset.customers` c ON o.customer_id=c.customer_id
        JOIN `target_dataset.order_items` oi ON o.order_id=oi.order_id
        GROUP BY EXTRACT(MONTH FROM o.order_purchase_timestamp), c.customer_city) x
        LIMIT 24;
```

| city | month | sales | prev_month_sales | month_on_month_growth |
|---|---|---|---|---|
| barreiras | 1 | 613.88 | | |
| barreiras | 3 | 2537.34 | 613.88 | 313.3283377858865 |
| barreiras | 4 | 1911.84 | 2537.34 | -24.651800704674979 |
| barreiras | 5 | 1296.97 | 1911.84 | -32.161164114151809 |
| barreiras | 6 | 1187.6 | 1296.97 | -8.4327316745954111 |
| barreiras | 7 | 574.49 | 1187.6 | -51.62596833950824 |
| barreiras | 8 | 1532.19 | 574.49 | 166.70438127730682 |
| barreiras | 9 | 678.58 | 1532.19 | -55.71175898550441 |
| barreiras | 10 | 433.5 | 678.58 | -36.116596421939938 |
| barreiras | 11 | 249.79 | 433.5 | -42.378316032295274 |
| barreiras | 12 | 27.75 | 249.79 | -88.890668161255462 |
| belmonte | 1 | 149.7 | | |
| belmonte | 3 | 37.99 | 149.7 | -74.622578490313956 |
| boa vista do sul | 4 | 27.99 | | |
| boituva | 1 | 477.3 | | |
| boituva | 2 | 1059.72 | 477.3 | 122.02388434946576 |
| boituva | 3 | 127.0 | 1059.72 | -88.015702260974592 |
| boituva | 4 | 324.0 | 127.0 | 155.11811023622047 |
| boituva | 5 | 1722.15 | 324.0 | 431.52777777777783 |
| boituva | 6 | 1155.79 | 1722.15 | -32.886798478645886 |
| boituva | 7 | 953.26 | 1155.79 | -17.523079452149609 |
| boituva | 8 | 436.0 | 953.26 | -54.26221597465539 |
| boituva | 10 | 315.19 | 436.0 | -27.708715596330276 |
| boituva | 11 | 561.2 | 315.19 | 78.051334115930089 |

### Year-On-Year growth orders by state

```
SELECT *, ROUND(LAG(sales) OVER(PARTITION BY state ORDER BY year), 2) AS
prev_year_sales,
        ROUND(((x.sales - ROUND(LAG(sales) OVER(PARTITION BY state ORDER BY year), 2))
/ (ROUND(LAG(sales, 1) OVER(PARTITION BY state ORDER BY year), 2))), 2) * 100 as
year_on_year_growth FROM
(SELECT c.customer_state AS state, EXTRACT(YEAR FROM o.order_purchase_timestamp) as
year, ROUND(SUM(oi.price), 2) AS sales,
        FROM `target_dataset.orders` o
        JOIN `target_dataset.customers` c ON o.customer_id=c.customer_id
        JOIN `target_dataset.order_items` oi ON o.order_id=oi.order_id
        GROUP BY EXTRACT(YEAR FROM o.order_purchase_timestamp), c.customer_state) x
        LIMIT 24;
```

| state | year | sales | prev_year_sales | year_on_year_growth |
|---|---|---|---|---|
| **RO** | 2017 | 24577.45 | | |
| **RO** | 2018 | 21563.19 | 24577.45 | -12.0 |
| **PE** | 2016 | 1369.1 | | |
| **PE** | 2017 | 123588.05 | 1369.1 | 8927.0 |
| **PE** | 2018 | 137830.88 | 123588.05 | 12.0 |
| **CE** | 2016 | 1689.38 | | |
| **CE** | 2017 | 112865.61 | 1689.38 | 6581.0 |
| **CE** | 2018 | 112699.72 | 112865.61 | -0.0 |
| **MT** | 2016 | 327.79 | | |
| **MT** | 2017 | 77650.22 | 327.79 | 23589.0 |
| **MT** | 2018 | 78475.52 | 77650.22 | 1.0 |
| **AM** | 2017 | 10494.49 | | |
| **AM** | 2018 | 11862.35 | 10494.49 | 13.0 |
| **RR** | 2016 | 112.59 | | |
| **RR** | 2017 | 1404.76 | 112.59 | 1148.0 |
| **RR** | 2018 | 6312.08 | 1404.76 | 349.0 |
| **GO** | 2016 | 984.39 | | |
| **GO** | 2017 | 138502.14 | 984.39 | 13969.999999999998 |
| **GO** | 2018 | 155105.42 | 138502.14 | 12.0 |
| **SC** | 2016 | 2373.06 | | |
| **SC** | 2017 | 233757.27 | 2373.06 | 9750.0 |
| **SC** | 2018 | 284423.01 | 233757.27 | 22.0 |
| **PB** | 2016 | 49.9 | | |
| **PB** | 2017 | 51903.37 | 49.9 | 103915.00000000001 |

### Year-On-Year growth orders by region

```
SELECT *, ROUND(LAG(sales) OVER(PARTITION BY city ORDER BY year), 2) AS
prev_year_sales,
        ROUND(((x.sales - ROUND(LAG(sales) OVER(PARTITION BY city ORDER BY year), 2)) /
(ROUND(LAG(sales, 1) OVER(PARTITION BY city ORDER BY year), 2))), 2) * 100 || "%" as
year_on_year_growth FROM
(SELECT c.customer_city AS city, EXTRACT(YEAR FROM o.order_purchase_timestamp) as year,
ROUND(SUM(oi.price), 2) AS sales,
        FROM `target_dataset.orders` o
        JOIN `target_dataset.customers` c ON o.customer_id=c.customer_id
        JOIN `target_dataset.order_items` oi ON o.order_id=oi.order_id
        GROUP BY EXTRACT(YEAR FROM o.order_purchase_timestamp), c.customer_city) x
        LIMIT 10;
```

| city | year | sales | prev_year_sales | year_on_year_growth |
|---|---|---|---|---|
| **agua limpa** | 2018 | 388.0 | | |
| **aiuruoca** | 2017 | 49.2 | | |
| **aiuruoca** | 2018 | 231.89 | 49.2 | 371% |
| **altamira** | 2017 | 2129.97 | | |
| **altamira** | 2018 | 3038.18 | 2129.97 | 43% |
| **altinopolis** | 2017 | 1319.0 | | |
| **altinopolis** | 2018 | 109.8 | 1319.0 | -92% |
| **anchieta** | 2017 | 1707.51 | | |
| **anchieta** | 2018 | 578.99 | 1707.51 | -66% |
| **bastos** | 2017 | 614.44 | | |

### How are customers distributed in Brazil

### Segregation by state

```
SELECT c.customer_state, ROUND(100 * COUNT(*) / (SELECT COUNT(c.customer_state) FROM
`target_dataset.customers` c), 2) AS customer_count FROM `target_dataset.customers` c
GROUP BY c.customer_state ORDER BY customer_count DESC LIMIT 10;
```

| customer_state | customer_count |
|---|---|
| SP | 41.98 |
| RJ | 12.92 |
| MG | 11.7 |
| RS | 5.5 |
| PR | 5.07 |
| SC | 3.66 |
| BA | 3.4 |
| DF | 2.15 |
| ES | 2.04 |
| GO | 2.03 |

**Segregation by City**

```
SELECT c.customer_city, ROUND(100 * COUNT(*) / (SELECT COUNT(c.customer_city) FROM
`target_dataset.customers` c), 2) AS customer_count FROM `target_dataset.customers` c
GROUP BY c.customer_city ORDER BY customer_count DESC LIMIT 10;
```

| customer_city | customer_count |
|---|---|
| sao paulo | 15.63 |
| rio de janeiro | 6.92 |
| belo horizonte | 2.79 |
| brasilia | 2.14 |
| curitiba | 1.53 |
| campinas | 1.45 |
| porto alegre | 1.39 |
| salvador | 1.25 |
| guarulhos | 1.2 |
| sao bernardo do campo | 0.94 |

## Conclusions:

1. Month-on-Month sales can be seen fluctuating for different states for every month.
2. Month-on-month sales for different regions can be seen fluctuating for all the months.
3. Year-On-Year sales can be seen increasing from 2016 to 2017, but decreasing from 2017 to 2018 for different states.
4. Year-On-Year sales can be seen increasing from 2017 to 2018 for different cities.
5. The majority of the customers are residing in SP and RJ states.
6. The vast majority of customers are from So Paulo and Rio de Janeiro.

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

### Get % increase in cost of orders from 2017 to 2018 (include months between Jan to Aug only)

```
SELECT *, ROUND(LAG(sales) OVER(ORDER BY year), 2) AS prev_year_sales,
        ROUND(((x.sales - ROUND(LAG(sales) OVER(ORDER BY year), 2)) / (ROUND(LAG(sales,
1) OVER(ORDER BY year), 2))), 2) * 100 || "%"
        AS percent_increase_cost_order
 FROM
(SELECT EXTRACT(YEAR FROM o.order_purchase_timestamp) year, ROUND(SUM(oi.price), 2) as
sales
FROM `target_dataset.orders` o JOIN `target_dataset.order_items` oi ON
o.order_id=oi.order_id
WHERE (EXTRACT(MONTH FROM o.order_purchase_timestamp) BETWEEN 1 AND 8) AND
(EXTRACT(YEAR FROM o.order_purchase_timestamp) BETWEEN 2017 AND 2018)
GROUP BY EXTRACT(YEAR FROM o.order_purchase_timestamp)) x
ORDER BY x.year;
```

| year | sales | prev_year_sales | percent_increase_cost_order |
|---|---|---|---|
| 2017 | 3113000.32 | | |
| 2018 | 7385905.8 | 3113000.32 | 137% |

### Mean & Sum of price and freight value by customer state

```
SELECT c.customer_state AS state, ROUND(AVG(oi.price), 2) AS average_price,
ROUND(AVG(oi.freight_value), 2) AS average_freight,
ROUND(SUM(oi.price), 2) AS total_price, ROUND(SUM(oi.freight_value), 2) AS
total_freight
        FROM `target_dataset.orders` o
        JOIN `target_dataset.customers` c ON o.customer_id=c.customer_id
        JOIN `target_dataset.order_items` oi ON o.order_id=oi.order_id
        GROUP BY c.customer_state
        ORDER BY c.customer_state ASC, average_price, average_freight, total_price,
total_freight DESC;
```

| state | average_price | average_freight | total_price | total_freight |
|---|---|---|---|---|
| AC | 173.73 | 40.07 | 15982.95 | 3686.75 |
| AL | 180.89 | 35.84 | 80314.81 | 15914.59 |
| AM | 135.5 | 33.21 | 22356.84 | 5478.89 |
| AP | 164.32 | 34.01 | 13474.3 | 2788.5 |
| BA | 134.6 | 26.36 | 511349.99 | 100156.68 |
| CE | 153.76 | 32.71 | 227254.71 | 48351.59 |
| DF | 125.77 | 21.04 | 302603.94 | 50625.5 |
| ES | 121.91 | 22.06 | 275037.31 | 49764.6 |
| GO | 126.27 | 22.77 | 294591.95 | 53114.98 |
| MA | 145.2 | 38.26 | 119648.22 | 31523.77 |
| MG | 120.75 | 20.63 | 1585308.03 | 270853.46 |
| MS | 142.63 | 23.37 | 116812.64 | 19144.03 |
| MT | 148.3 | 28.17 | 156453.53 | 29715.43 |
| PA | 165.69 | 35.83 | 178947.81 | 38699.3 |
| PB | 191.48 | 42.72 | 115268.08 | 25719.73 |
| PE | 145.51 | 32.92 | 262788.03 | 59449.66 |
| PI | 160.36 | 39.15 | 86914.08 | 21218.2 |
| PR | 119.0 | 20.53 | 683083.76 | 117851.68 |
| RJ | 125.12 | 20.96 | 1824092.67 | 305589.31 |
| RN | 156.97 | 35.65 | 83034.98 | 18860.1 |
| RO | 165.97 | 41.07 | 46140.64 | 11417.38 |
| RR | 150.57 | 42.98 | 7829.43 | 2235.19 |
| RS | 120.34 | 21.74 | 750304.02 | 135522.74 |
| SC | 124.65 | 21.47 | 520553.34 | 89660.26 |
| SE | 153.04 | 36.65 | 58920.85 | 14111.47 |
| SP | 109.65 | 15.15 | 5202955.05 | 718723.07 |
| TO | 157.53 | 37.25 | 49621.74 | 11732.68 |

**Observations:**

1. The order count has been increased by 137% from 2017 to 2018.

## 5. Analysis on sales, freight and delivery time

### Calculate days between purchasing, delivering and estimated delivery

```
SELECT DATE_DIFF(EXTRACT(DATE FROM o.order_delivered_customer_date), EXTRACT(DATE FROM
o.order_purchase_timestamp), DAY) AS  time_to_delivery,
        DATE_DIFF(EXTRACT(DATE FROM o.order_estimated_delivery_date), EXTRACT(DATE FROM
o.order_delivered_customer_date), DAY) AS  diff_estimated_delivery,
        FROM `my-project-5366-212807.target_dataset.orders`o ORDER BY time_to_delivery
DESC LIMIT 5;
```

| time_to_delivery | diff_estimated_delivery |
|---|---|
| 210 | -181 |
| 208 | -188 |
| 196 | -165 |
| 195 | -166 |
| 195 | -155 |

### Group data by state, take mean of freight_value, time_to_delivery, diff_estimated_delivery

```
SELECT  c.customer_state, ROUND(AVG(DATE_DIFF(EXTRACT(DATE FROM
o.order_delivered_customer_date), EXTRACT(DATE FROM o.order_purchase_timestamp), DAY)),
2) AS  time_to_delivery,
        ROUND(AVG(DATE_DIFF(EXTRACT(DATE FROM o.order_estimated_delivery_date),
EXTRACT(DATE FROM o.order_delivered_customer_date), DAY)), 2) AS
diff_estimated_delivery,
        ROUND(AVG(oi.freight_value), 2) AS average_freight
        FROM `my-project-5366-212807.target_dataset.orders`o JOIN
`target_dataset.customers` c ON o.customer_id=c.customer_id
        JOIN `target_dataset.order_items` oi ON o.order_id=oi.order_id
        GROUP BY c.customer_state
        ORDER BY c.customer_state ASC;
```

| customer_state | time_to_delivery | diff_estimated_delivery | average_freight |
|---|---|---|---|
| AC | 20.68 | 20.98 | 40.07 |
| AL | 24.45 | 8.74 | 35.84 |
| AM | 26.34 | 19.93 | 33.21 |
| AP | 28.22 | 18.4 | 34.01 |
| BA | 19.19 | 10.98 | 26.36 |
| CE | 20.92 | 11.1 | 32.71 |
| DF | 12.89 | 12.2 | 21.04 |
| ES | 15.59 | 10.65 | 22.06 |
| GO | 15.34 | 12.29 | 22.77 |
| MA | 21.59 | 9.91 | 38.26 |

| | | | |
|---|---|---|---|
| MG | 11.92 | 13.34 | 20.63 |
| MS | 15.46 | 11.23 | 23.37 |
| MT | 17.91 | 14.57 | 28.17 |
| PA | 23.7 | 14.25 | 35.83 |
| PB | 20.55 | 13.04 | 42.72 |
| PE | 18.22 | 13.45 | 32.92 |
| PI | 19.32 | 11.53 | 39.15 |
| PR | 11.89 | 13.49 | 20.53 |
| RJ | 15.07 | 12.01 | 20.96 |
| RN | 19.27 | 13.95 | 35.65 |
| RO | 19.66 | 20.04 | 41.07 |
| RR | 28.17 | 18.33 | 42.98 |
| RS | 15.13 | 14.13 | 21.74 |
| SC | 14.95 | 11.57 | 21.47 |
| SE | 21.42 | 10.0 | 36.65 |
| SP | 8.66 | 11.21 | 15.15 |
| TO | 17.4 | 12.34 | 37.25 |
| SC | 14.95 | 11.57 | 21.47 |
| SE | 21.42 | 10.0 | 36.65 |
| SP | 8.66 | 11.21 | 15.15 |
| TO | 17.4 | 12.34 | 37.25 |

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

```
SELECT  c.customer_state, ROUND(AVG(DATE_DIFF(EXTRACT(DATE FROM
o.order_delivered_customer_date), EXTRACT(DATE FROM o.order_purchase_timestamp), DAY)),
2) AS  time_to_delivery,
        ROUND(AVG(DATE_DIFF(EXTRACT(DATE FROM o.order_estimated_delivery_date),
EXTRACT(DATE FROM o.order_delivered_customer_date), DAY)), 2) AS
diff_estimated_delivery,
        ROUND(AVG(oi.freight_value), 2) AS average_freight
        FROM `my-project-5366-212807.target_dataset.orders`o JOIN
`target_dataset.customers` c ON o.customer_id=c.customer_id
        JOIN `target_dataset.order_items` oi ON o.order_id=oi.order_id
        GROUP BY c.customer_state
        ORDER BY average_freight DESC LIMIT 5;
```

| customer_state | time_to_delivery | diff_estimated_delivery | average_freight |
|---|---|---|---|
| **RR** | 28.17 | 18.33 | 42.98 |
| **PB** | 20.55 | 13.04 | 42.72 |
| **RO** | 19.66 | 20.04 | 41.07 |
| **AC** | 20.68 | 20.98 | 40.07 |
| **PI** | 19.32 | 11.53 | 39.15 |

```sql
SELECT c.customer_state, ROUND(AVG(DATE_DIFF(EXTRACT(DATE FROM
o.order_delivered_customer_date), EXTRACT(DATE FROM o.order_purchase_timestamp), DAY)),
2) AS  time_to_delivery,
        ROUND(AVG(DATE_DIFF(EXTRACT(DATE FROM o.order_estimated_delivery_date),
EXTRACT(DATE FROM o.order_delivered_customer_date), DAY)), 2) AS
diff_estimated_delivery,
        ROUND(AVG(oi.freight_value), 2) AS average_freight
        FROM `my-project-5366-212807.target_dataset.orders`o JOIN
`target_dataset.customers` c ON o.customer_id=c.customer_id
        JOIN `target_dataset.order_items` oi ON o.order_id=oi.order_id
        GROUP BY c.customer_state
        ORDER BY average_freight ASC LIMIT 5;
```

| customer_state | time_to_delivery | diff_estimated_delivery | average_freight |
|---|---|---|---|
| **SP** | 8.66 | 11.21 | 15.15 |
| **PR** | 11.89 | 13.49 | 20.53 |
| **MG** | 11.92 | 13.34 | 20.63 |
| **RJ** | 15.07 | 12.01 | 20.96 |
| **DF** | 12.89 | 12.2 | 21.04 |

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

```sql
SELECT  c.customer_state, ROUND(AVG(DATE_DIFF(EXTRACT(DATE FROM
o.order_delivered_customer_date), EXTRACT(DATE FROM o.order_purchase_timestamp), DAY)),
2) AS  time_to_delivery,
        ROUND(AVG(DATE_DIFF(EXTRACT(DATE FROM o.order_estimated_delivery_date),
EXTRACT(DATE FROM o.order_delivered_customer_date), DAY)), 2) AS
diff_estimated_delivery,
        ROUND(AVG(oi.freight_value), 2) AS average_freight
        FROM `my-project-5366-212807.target_dataset.orders`o JOIN
`target_dataset.customers` c ON o.customer_id=c.customer_id
        JOIN `target_dataset.order_items` oi ON o.order_id=oi.order_id
        GROUP BY c.customer_state
        ORDER BY time_to_delivery DESC LIMIT 5;
```

| customer_state | time_to_delivery | diff_estimated_delivery | average_freight |
|---|---|---|---|
| AP | 28.22 | 18.4 | 34.01 |
| RR | 28.17 | 18.33 | 42.98 |
| AM | 26.34 | 19.93 | 33.21 |
| AL | 24.45 | 8.74 | 35.84 |
| PA | 23.7 | 14.25 | 35.83 |

```
SELECT  c.customer_state, ROUND(AVG(DATE_DIFF(EXTRACT(DATE FROM
o.order_delivered_customer_date), EXTRACT(DATE FROM o.order_purchase_timestamp), DAY)),
2) AS  time_to_delivery,
        ROUND(AVG(DATE_DIFF(EXTRACT(DATE FROM o.order_estimated_delivery_date),
EXTRACT(DATE FROM o.order_delivered_customer_date), DAY)), 2) AS
diff_estimated_delivery,
        ROUND(AVG(oi.freight_value), 2) AS average_freight
        FROM `my-project-5366-212807.target_dataset.orders`o JOIN
`target_dataset.customers` c ON o.customer_id=c.customer_id
        JOIN `target_dataset.order_items` oi ON o.order_id=oi.order_id
        GROUP BY c.customer_state
        ORDER BY time_to_delivery ASC LIMIT 5;
```

| customer_state | time_to_delivery | diff_estimated_delivery | average_freight |
|---|---|---|---|
| SP | 8.66 | 11.21 | 15.15 |
| PR | 11.89 | 13.49 | 20.53 |
| MG | 11.92 | 13.34 | 20.63 |
| DF | 12.89 | 12.2 | 21.04 |
| SC | 14.95 | 11.57 | 21.47 |

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

```
SELECT  DISTINCT c.customer_state, DATE_DIFF(EXTRACT(DATE FROM
o.order_delivered_customer_date), EXTRACT(DATE FROM o.order_purchase_timestamp), DAY)
AS time_to_delivery
        FROM `target_dataset.orders`o JOIN `target_dataset.customers` c ON
o.customer_id=c.customer_id
        JOIN `target_dataset.order_items` oi ON o.order_id=oi.order_id
        WHERE DATE_DIFF(EXTRACT(DATE FROM o.order_delivered_customer_date),
EXTRACT(DATE FROM o.order_purchase_timestamp), DAY) IS NOT NULL
        ORDER BY time_to_delivery ASC
        LIMIT 6;
```

| customer_state | time_to_delivery |
|---|---|
| RJ | 0 |
| RJ | 1 |
| SP | 1 |
| RS | 1 |
| RN | 1 |
| DF | 1 |

**Observations:**

1. Maximum distance taken between purchasing and delivery is 210 days, followed by 208 days.
2. The State RR and PB have the highest average freight value.
3. The states with the lowest average freight value are SP and PR.
4. The states with the longest delivery times are AP and RR.

5. State PP and PR have the lowest delivery times.
6. Delivery is fast in RJ and SP states.

## 6.Payment type analysis

### Month over Month count of orders for different payment types

```
SELECT *,ROUND(LAG(orders_count) OVER(PARTITION BY payment_type ORDER BY month), 2) AS
prev_count_sales,
        ROUND(((x.orders_count - ROUND(LAG(orders_count) OVER(PARTITION BY payment_type
ORDER BY month), 2)) / (ROUND(LAG(orders_count, 1) OVER(PARTITION BY payment_type ORDER
BY month), 2))), 2) * 100 || "%" as month_on_month_sales FROM
(SELECT p.payment_type, EXTRACT(MONTH FROM o.order_purchase_timestamp) AS month,
COUNT(*) as orders_count
FROM `target_dataset.orders` o
JOIN `target_dataset.payments` p ON o.order_id=p.order_id
GROUP BY p.payment_type, EXTRACT(MONTH FROM o.order_purchase_timestamp)) x;
```

| payment_type | month | orders_count | prev_count_sales | month_on_month_sales |
|---|---|---|---|---|
| debit_card | 1 | 118 | | |
| debit_card | 2 | 82 | 118.0 | -31% |
| debit_card | 3 | 109 | 82.0 | 33% |
| debit_card | 4 | 124 | 109.0 | 14.000000000000002% |
| debit_card | 5 | 81 | 124.0 | -35% |
| debit_card | 6 | 209 | 81.0 | 158% |
| debit_card | 7 | 264 | 209.0 | 26% |
| debit_card | 8 | 311 | 264.0 | 18% |
| debit_card | 9 | 43 | 311.0 | -86% |
| debit_card | 10 | 54 | 43.0 | 26% |
| debit_card | 11 | 70 | 54.0 | 30% |
| debit_card | 12 | 64 | 70.0 | -9% |
| voucher | 1 | 477 | | |
| voucher | 2 | 424 | 477.0 | -11% |
| voucher | 3 | 591 | 424.0 | 39% |
| voucher | 4 | 572 | 591.0 | -3% |
| voucher | 5 | 613 | 572.0 | 7.0000000000000009% |
| voucher | 6 | 563 | 613.0 | -8% |
| voucher | 7 | 645 | 563.0 | 15% |
| voucher | 8 | 589 | 645.0 | -9% |
| voucher | 9 | 302 | 589.0 | -49% |
| voucher | 10 | 318 | 302.0 | 5% |
| voucher | 11 | 387 | 318.0 | 22% |
| voucher | 12 | 294 | 387.0 | -24% |
| not_defined | 8 | 2 | | |
| not_defined | 9 | 1 | 2.0 | -50% |
| UPI | 1 | 1715 | | |
| UPI | 2 | 1723 | 1715.0 | 0% |
| UPI | 3 | 1942 | 1723.0 | 13% |

| | | | | |
|---|---|---|---|---|
| UPI | 4 | 1783 | 1942.0 | -8% |
| UPI | 5 | 2035 | 1783.0 | 14.000000000000002% |
| UPI | 6 | 1807 | 2035.0 | -11% |
| UPI | 7 | 2074 | 1807.0 | 15% |
| UPI | 8 | 2077 | 2074.0 | 0% |
| UPI | 9 | 903 | 2077.0 | -56.999999999999993% |
| UPI | 10 | 1056 | 903.0 | 17% |
| UPI | 11 | 1509 | 1056.0 | 43% |
| UPI | 12 | 1160 | 1509.0 | -23% |
| credit_card | 1 | 6103 | | |
| credit_card | 2 | 6609 | 6103.0 | 8% |
| credit_card | 3 | 7707 | 6609.0 | 17% |
| credit_card | 4 | 7301 | 7707.0 | -5% |
| credit_card | 5 | 8350 | 7301.0 | 14.000000000000002% |
| credit_card | 6 | 7276 | 8350.0 | -13% |
| credit_card | 7 | 7841 | 7276.0 | 8% |
| credit_card | 8 | 8269 | 7841.0 | 5% |
| credit_card | 9 | 3286 | 8269.0 | -60% |
| credit_card | 10 | 3778 | 3286.0 | 15% |
| credit_card | 11 | 5897 | 3778.0 | 56.000000000000007% |
| credit_card | 12 | 4378 | 5897.0 | -26% |

### Distribution of payment installments and count of orders

```
SELECT p.payment_installments, ROUND(100 * COUNT(*) / (SELECT COUNT(o.order_id) FROM
`target_dataset.orders` o), 4) || "%" AS order_count
FROM `target_dataset.orders` o
JOIN `target_dataset.payments` p ON o.order_id=p.order_id
GROUP BY p.payment_installments;
```

| payment_installments | order_count |
|---|---|
| 0 | 0.002% |
| 1 | 52.8414% |
| 2 | 12.4828% |
| 3 | 10.5198% |
| 4 | 7.1379% |
| 5 | 5.2685% |
| 6 | 3.942% |
| 7 | 1.6351% |
| 8 | 4.292% |
| 9 | 0.6476% |
| 10 | 5.358% |
| 11 | 0.0231% |
| 12 | 0.1337% |
| 13 | 0.0161% |
| 14 | 0.0151% |
| 15 | 0.0744% |
| 16 | 0.005% |
| 17 | 0.008% |

| | |
|---|---|
| 18 | 0.0272% |
| 20 | 0.0171% |
| 21 | 0.003% |
| 22 | 0.001% |
| 23 | 0.001% |
| 24 | 0.0181% |

**Observations:**

1. Utilization of debit cards significantly increased from May to June by 158%. Utilization of debit cards decreased to 88% from August to September.
2. Usage of vouchers increased from February to March by 39%. Usage of vouchers decreased from August to September by 49%.
3. Utilization of UPI increased from October to November by 43%. Utilization of UPI decreased from August to September by 57%.
4. Utilization of credit cards increased from October to November by 56%. Utilization of credit cards decreased from August to September by 60%.
5. Payment instalments of 1 month contribute to 53% of sales, followed by 2 and 3 months.

**Insights:**

1. The dataset consists of data from 2016 to 2018. Different states and cities in Brazil are covered in the dataset.
2. Target's most popular product category is bed & bath, followed by sports & leisure.
3. SP is the most frequent state where the orders are coming from, followed by RJ and MG.
4. Most of the customers have given a 5 star rating to a product, followed by a 4 star rating.
5. We can see that September has the highest average spending, making it $129, followed by April and October, respectively. November and February have the lowest average number of purchases.
6. On the $1^{st}$ of every month, the highest average purchase is $128.53, followed by the $4^{th}$ and $18^{th}$ , respectively. The average price in the 22nd and 21st districts is the lowest.
7. The highest average purchase is on Saturday, resulting in $123.6, followed by Friday and Monday. The least average purchase is made on Sunday and Wednesday.
8. The highest average purchase is during the $18^{th}$ hour, resulting in $125.39, followed by the $14^{th}$ and $9^{th}$. The lowest average purchases are during the $2^{nd}$ and $4^{th}$ hours, respectively.
9. The majority of the customers prefer to buy during the afternoon, while very few customers prefer to buy during the night. Customers prefer to buy during the work week instead of the weekend.
10. Month-on-month sales can be seen fluctuating in different states every month. Month-on-month sales for different regions can be seen fluctuating for all the months. Year-on-year sales can be seen increasing from 2016 to 2017, but decreasing from 2017 to 2018 for

different states. Year-On-Year sales can be seen increasing from 2017 to 2018 in different cities.

11. Utilization of debit cards significantly increased from May to June by 158%. Utilization of debit cards decreased to 88% from August to September.
12. Usage of vouchers increased from February to March by 39%. Usage of vouchers decreased from August to September by 49%.
13. Utilization of UPI increased from October to November by 43%. Utilization of UPI decreased from August to September by 57%.
14. Utilization of credit cards increased from October to November by 56%. Utilization of credit cards decreased from August to September by 60%.
15. Payment instalments of 1 month contribute to 53% of sales, followed by 2 and 3 months.

## Recommendations:

1. Target should open multiple stores in different states, including AC, AM, MA, etc.
2. Target should introduce more vouchers, UPI, and credit card payment types to increase month-on-month usage by providing discounts on particular occasions such as Women's Day, Mother's Day, Christmas, New Year, etc.
3. The majority of the customers are opting for 1, 2, or 3 payment instalments. Target should be promoting more payment instalments, resulting in more interest from the customers.
4. Month-on-month and year-on-year are very fluctuating for different regions/states. Target should introduce monthly offers as a way to increase their sales based on states and cities.
5. During the weekends, sales at Target are relatively low. Target should introduce discounts during the weekends to attract more customers.