Business Case: Target SQL

Details:
Name: Rohan Anand
Registered email-id: rohan.btech.2021@gmail.com
Batch: Scaler DSML Dec 2022 (Beginner) MWF 9pm

Mindset:

1. Evaluation will be kept lenient, so make sure you attempt this case study.
2. It is understandable that you might struggle with getting started on this. Just brainstorm, discuss with peers, or get help from TAs.
3. Try to attempt this before it is discussed in the Live Case Discussion with the Instructor.
4. 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.

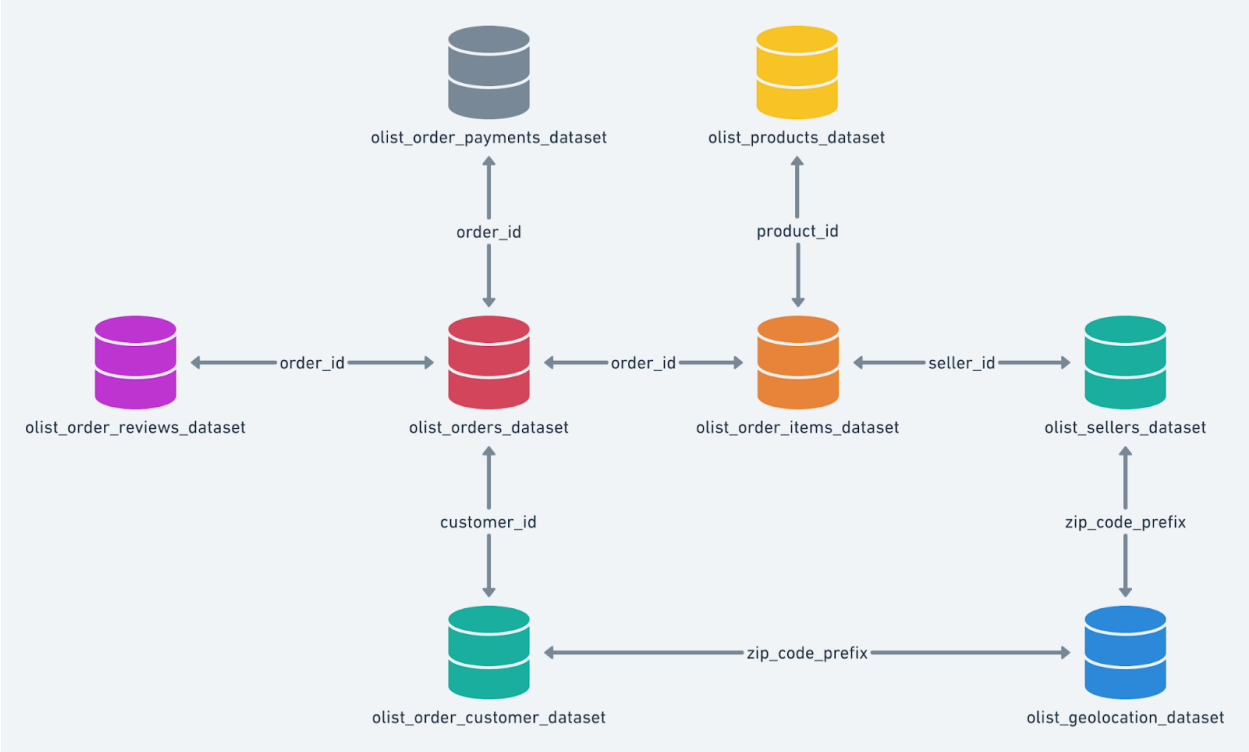
This business case has information of 100k orders from 2016 to 2018 made at Target in Brazil. Its features allows viewing an order from multiple dimensions: from order status, price, payment and freight performance to customer location, product attributes and finally reviews written by customers.

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

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:

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 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 columns in a table
    2. Time period for which the data is given
    3. Cities and States of customers ordered during the given period
2. In-depth Exploration:
    1. Is there a growing trend on e-commerce in Brazil? How can we describe a complete scenario? Can we see some seasonality with peaks at specific months?
    2. What time do Brazilian customers tend to buy (Dawn, Morning, Afternoon or Night)?
3. Evolution of E-commerce orders in the Brazil region:
    1. Get month on month orders by states
    2. Distribution of customers across the states in Brazil
4. Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others.
    1. Get % increase in cost of orders from 2017 to 2018 (include months between Jan to Aug only) - You can use "payment_value" column in payments table
    2. Mean & Sum of price and freight value by customer state

5. Analysis on sales, freight and delivery time

1. Calculate days between purchasing, delivering and estimated delivery
2. Find time_to_delivery & diff_estimated_delivery. Formula for the same given below:
    - time_to_delivery = order_purchase_timestamp-order_delivered_customer_date
    - diff_estimated_delivery = order_estimated_delivery_date-order_delivered_customer_date
3. Group data by state, take mean of freight_value, time_to_delivery, diff_estimated_delivery
4. Sort the data to get the following:
5. Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5
6. Top 5 states with highest/lowest average time to delivery
7. Top 5 states where delivery is really fast/ not so fast compared to estimated date

6. Payment type analysis:

1. Month over Month count of orders for different payment types
2. Count of orders based on the no. of payment installments

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 <IMP>:

- Use Word doc to paste your SQL queries along with the screenshot of the first 10 rows
- Convert your solutions notebook into PDF, and upload the same on the platform
- Optionally, you may add images/graphs in the text editor by taking screenshots
- After submitting, you will not be allowed to edit your submission

SOLUTION:

Q1)Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset.
1)Data type of columns in a table
2)Time period for which the data is given
3)Cities and States of customers ordered during the given period

1.1)Data type of columns in a table

Customers Table:

```
SELECT * FROM `business-case-study-sql.Target_dataset.customers` LIMIT 1000
```

Query results

SAVE RESULTS | EXPLORE DATA

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

Row	customer_id	customer_unique_id	customer_zip_c	customer_city	customer_state
1	0735e7e4298a2ebbb4664934...	fcb003b1bdc0df64b4d065d9b...	59650	acu	RN
2	903b3d86e3990db01619a4eb...	46824822b15da44e983b021d...	59650	acu	RN
3	38c97666e962d4fea7fd6a83e...	b6108acc674ae5c99e29adc10...	59650	acu	RN
4	77c2f46cf580f4874c9a5751c2...	402cce5c0509000eed9e77fec...	63430	ico	CE
5	4d3ef4cfffb8ad4767c199c36a...	6ba00666ab7eada5ceec279b2...	63430	ico	CE
6	3000841b86e1fbe9493b52324...	796a0b1a21f597704057184a1...	63430	ico	CE
7	3c325415ccc7e622c66dec4bc...	05d1d2d9f0161c5f397ce7fc77...	63430	ico	CE
8	04f3a7b250e3be964f01bf22bc...	c34585a0276ecc5e4fb03de75...	63430	ico	CE
9	894202b8ef01f4719a4691e79...	01a4fe5fc00bbdb0b0a4af5a53...	63430	ico	CE
10	9d715b9fb75a9d081c14126c0...	8f399f3b7ace8e6245422c9e1f...	63430	ico	CE

Load more

Results per page: 50 | 1 – 50 of 1000

GeoLocation Table:

```
SELECT * FROM `business-case-study-sql.Target_dataset.geolocation` LIMIT 1000
```

Query results

SAVE RESULTS | EXPLORE DATA

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

Row	geolocation_zip	geolocation_lat	geolocation_lng	geolocation_city	geolocation_state
1	49010	-10.910514...	-37.052400...	aracaju	SE
2	49047	-10.9268145	-37.071063...	aracaju	SE
3	49030	-10.970164...	-37.061643...	aracaju	SE
4	49048	-10.940183...	-37.070850...	aracaju	SE
5	49050	-10.927157...	-37.063078...	aracaju	SE
6	49015	-10.923370...	-37.045169...	aracaju	SE
7	49045	-10.930406...	-37.067178...	aracaju	SE
8	49052	-10.922973...	-37.057752...	aracaju	SE
9	49044	-10.992080...	-37.103470...	aracaju	SE
10	49048	-10.940235...	-37.071043...	aracaju	SE

Results per page: 50 | 1 – 50 of 1000

Order Items Table:

```
SELECT * FROM `business-case-study-sql.Target_dataset.order_items` LIMIT 1000
```

Query results

SAVE RESULTS | EXPLORE DATA

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

Row	order_id	order_item_id	product_id	seller_id	shipping_limit_date	price	freight_value
1	f09e36e258656850b92657ac5...	1	44d53f1240d6332232e4393c0...	b64d51f0435e884e8de603b16...	2018-07-09 13:31:36 UTC	3.0	12.79
2	f9ccaff7267fd0cf076e795b1fa...	1	44d53f1240d6332232e4393c0...	b64d51f0435e884e8de603b16...	2018-08-14 14:04:44 UTC	3.0	15.23
3	c79bdf061e22288609201ec60...	1	5304ff3fa35856a156e1170a60...	cf6f6bc4df3999b9c6440f124f...	2017-05-12 19:05:20 UTC	3.5	8.72
4	37193e64eb9a46b7f3197762f...	1	98224bfc1eaadb3a394ec334c...	ce616e1913288884e7742faac...	2018-06-28 01:30:49 UTC	3.5	7.39
5	95d6357ffe41aa6d2998852a7...	1	98224bfc1eaadb3a394ec334c...	ce616e1913288884e7742faac...	2018-06-12 19:15:14 UTC	3.5	18.23
6	95d6357ffe41aa6d2998852a7...	2	98224bfc1eaadb3a394ec334c...	ce616e1913288884e7742faac...	2018-06-12 19:15:14 UTC	3.5	18.23
7	95d6357ffe41aa6d2998852a7...	3	98224bfc1eaadb3a394ec334c...	ce616e1913288884e7742faac...	2018-06-12 19:15:14 UTC	3.5	18.23
8	95d6357ffe41aa6d2998852a7...	4	98224bfc1eaadb3a394ec334c...	ce616e1913288884e7742faac...	2018-06-12 19:15:14 UTC	3.5	18.23
9	95d6357ffe41aa6d2998852a7...	5	98224bfc1eaadb3a394ec334c...	ce616e1913288884e7742faac...	2018-06-12 19:15:14 UTC	3.5	18.23
10	dde867f83e689b0167785b684...	1	914323edd50192310dd03435...	2c9e548be18521d1c43cde1c5...	2017-10-20 14:50:12 UTC	4.5	11.85

Results per page: 50 | 1 – 50 of 1000

Orders Table:

```
SELECT * FROM `business-case-study-sql.Target_dataset.orders` LIMIT 1000
```

Query results

SAVE RESULTS | EXPLORE DATA

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

Row	order_id	customer_id	order_status	order_purchase_timestamp	order_approved_at	order_delivered_carr
1	7a4df5d8cff4090e541401a20a...	725e9c75605414b21fd8c8d5a...	created	2017-11-25 11:10:33 UTC	null	null
2	35de4050331c6c644cddc86f4...	4ee64f4bfc542546f422da0aeb...	created	2017-12-05 01:07:58 UTC	null	null
3	b5359909123fa03c50bdb0cfe...	438449d4af8980d107bf04571...	created	2017-12-05 01:07:52 UTC	null	null
4	dba5062fbda3af4fb6c33b1e04...	964a6df3d9bdf60fe3e7b8bb69...	created	2018-02-09 17:21:04 UTC	null	null
5	90ab3e7d52544ec7bc3363c82...	7d61b9f4f216052ba664f22e9c...	created	2017-11-06 13:12:34 UTC	null	null
6	fa65dad1b0e818e3ccc5cb0e3...	9af2372a1e49340278e7c1ef8...	shipped	2017-04-20 12:45:34 UTC	2017-04-22 09:10:13 UTC	2017-04-24 11:31:17
7	1df2775799eecdf9dd8502425...	1240c2e65c4601dd860e3a367...	shipped	2017-07-13 11:03:05 UTC	2017-07-13 11:10:22 UTC	2017-07-18 18:17:30
8	6190a94657e1012983a274b8...	5fc4c97dcb63903f996714524...	shipped	2017-07-11 13:36:30 UTC	2017-07-11 13:45:15 UTC	2017-07-13 17:55:46
9	58ce513a55c740a3a81e8c8b7...	530d41b47b9dda9bc6f31d856...	shipped	2017-07-29 18:05:07 UTC	2017-07-29 18:15:17 UTC	2017-07-31 16:41:59
10	088683f795a3d30bfd61152c4f...	58d89fd1f863819ff9b040734f...	shipped	2017-07-13 10:02:47 UTC	2017-07-14 02:25:54 UTC	2017-07-20 20:02:58

Results per page: 50 | 1 – 50 of 1000

Payments:

```
SELECT * FROM `business-case-study-sql.Target_dataset.payments` LIMIT 1000
```

Query results

SAVE RESULTS | EXPLORE DATA

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

Row	order_id	payment_sequei	payment_type	payment_installi	payment_value
1	1a5710839416 9c0b47d8f876a...	2	credit_card	0	129.94
2	744bade1fcf9ff3f31d860ace07...	2	credit_card	0	58.69
3	8bcbe01d44d147f901cd31926...	4	voucher	1	0.0
4	fa65dad1b0e818e3ccc5cb0e3...	14	voucher	1	0.0
5	6ccb433e00daae1283ccc9561...	4	voucher	1	0.0
6	4637ca194b6387e2d538dc89b...	1	not_defined	1	0.0
7	00b1cb0320190ca0daa2c88b3...	1	not_defined	1	0.0
8	45ed6e85398a87c253db47c2d...	3	voucher	1	0.0
9	fa65dad1b0e818e3ccc5cb0e3...	13	voucher	1	0.0
10	c8c528189310eaa44a745b8d9...	1	not_defined	1	0.0

Results per page: 50 | 1 – 50 of 1000

Products:

```
SELECT * FROM `business-case-study-sql.Target_dataset.products` LIMIT 1000
```

Query results

SAVE RESULTS | EXPLORE DATA

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

Row	product_id	product_category	product_name_l	product_descrip	product_photos	product_weight	product_length	product_height	product_width_c
1	5eb564652db742ff8f28759cd8...	null	null	null	null	null	null	null	null
2	09ff539a621711667c43eba6a...	babies	60	865	3	null	null	null	null
3	2f763ba79d9cd987b2034aac7...	electronics	45	1198	2	595	8	6	6
4	a69f15dfb803d485e8933e80b...	Watches present	53	506	6	150	11	16	6
5	e1cfc87f543782b8a78b59fc85...	Garden tools	39	524	4	369	26	7	7
6	106392145fca363410d287a81...	bed table bath	58	309	1	2083	12	2	7
7	7e33f4a1c59f89da30a335b2d...	electronics	51	381	3	1075	22	5	7
8	bc9cc914f974963c07be697fc...	HEALTH BEAUTY	55	435	1	75	14	9	7
9	5ae533eac9c0e93b3f89bc9ae...	computer accessories	58	1340	1	83	12	8	8
10	67d1a56495104e195338ec900...	pet Shop	20	2153	1	275	8	13	8

Results per page: 50 | 1 – 50 of 1000

Sellers:

```
SELECT * FROM `business-case-study-sql.Target_dataset.sellers` LIMIT 1000
```

Query results

SAVE RESULTS | EXPLORE DATA

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

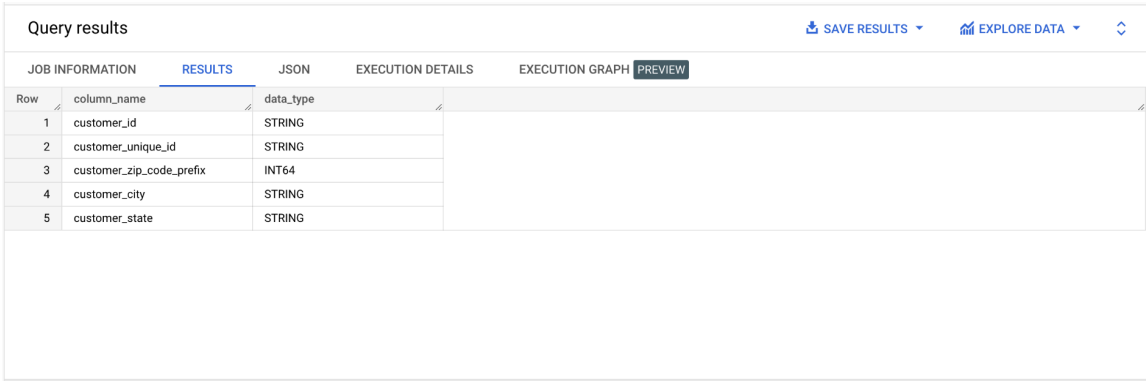
Row	seller_id	seller_zip_code	seller_city	seller_state
1	4be2e7f96b4fd749d52dff41f8...	69900	rio branco	AC
2	327b89b872c14d1c0be7235ef...	69005	manaus	AM
3	4221a7df464f1fe2955934e30f...	48602	bahia	BA
4	651530bf5c607240ccdd89a30...	44600	ipira	BA
5	2b402d5dc42554061f8ea98d1...	44900	irece	BA
6	d03698c2efd04a549382afa66...	45658	ilheus	BA
7	c72de06d72748d1a0dfb2125b...	46430	guanambi	BA
8	fc59392d66ef99377e50356ee...	40243	salvador	BA
9	b00af24704019bd2e1b335e70...	40130	salvador	BA
10	eb4a59a06b3948e851a7d7a83...	41820	salvador	BA

Results per page: 50 | 1 – 50 of 1000

1.1) Data type of columns in a table

Customers table:

```
SELECT column_name, data_type
FROM `business-case-study-sql.Target_dataset.INFORMATION_SCHEMA.COLUMNS`
WHERE table_name = 'customers'
```



Query results

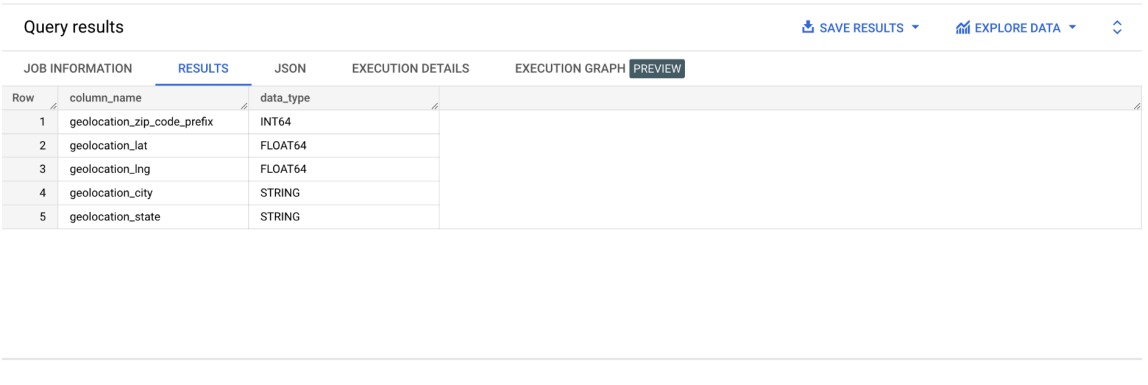
SAVE RESULTS | EXPLORE DATA

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

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

GeoLocation:

```
SELECT column_name, data_type
FROM `business-case-study-sql.Target_dataset.INFORMATION_SCHEMA.COLUMNS`
WHERE table_name = 'geolocation'
```



Query results

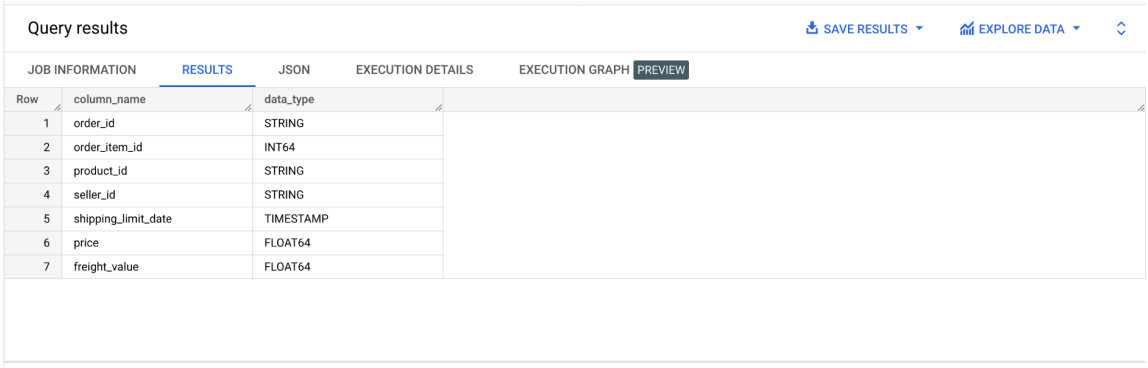
SAVE RESULTS | EXPLORE DATA

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

Row	column_name	data_type
1	geolocation_zip_code_prefix	INT64
2	geolocation_lat	FLOAT64
3	geolocation_lng	FLOAT64
4	geolocation_city	STRING
5	geolocation_state	STRING

Order_items:

```
SELECT column_name, data_type
FROM `business-case-study-sql.Target_dataset.INFORMATION_SCHEMA.COLUMNS`
WHERE table_name = 'order_items'
```



Query results

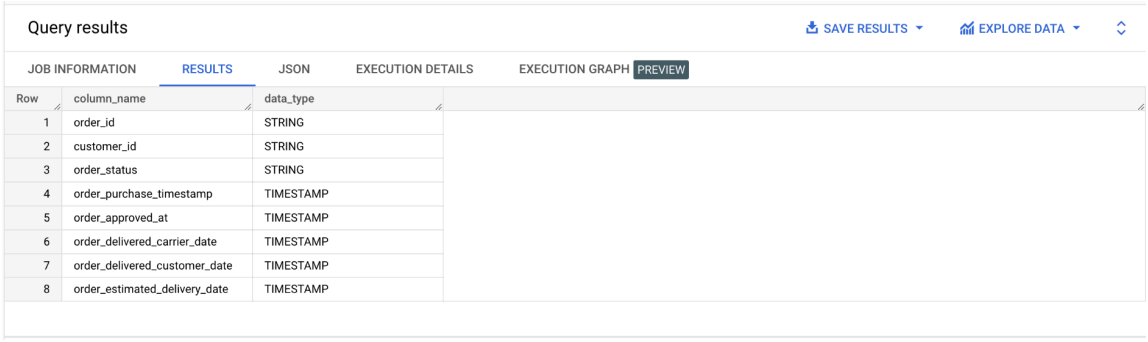
SAVE RESULTS | EXPLORE DATA

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

Row	column_name	data_type
1	order_id	STRING
2	order_item_id	INT64
3	product_id	STRING
4	seller_id	STRING
5	shipping_limit_date	TIMESTAMP
6	price	FLOAT64
7	freight_value	FLOAT64

Orders:

```
SELECT column_name, data_type
FROM `business-case-study-sql.Target_dataset.INFORMATION_SCHEMA.COLUMNS`
WHERE table_name = 'orders'
```



Query results

SAVE RESULTS | EXPLORE DATA

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

Row	column_name	data_type
1	order_id	STRING
2	customer_id	STRING
3	order_status	STRING
4	order_purchase_timestamp	TIMESTAMP
5	order_approved_at	TIMESTAMP
6	order_delivered_carrier_date	TIMESTAMP
7	order_delivered_customer_date	TIMESTAMP
8	order_estimated_delivery_date	TIMESTAMP

Payments:

```
SELECT column_name, data_type
FROM `business-case-study-sql.Target_dataset.INFORMATION_SCHEMA.COLUMNS`
WHERE table_name = 'payments'
```

Query results

SAVE RESULTS | EXPLORE DATA

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

Row	order_id	payment_seque	payment_type	payment_install	payment_value
1	1a57108394169c0b47d8f876a...	2	credit_card	0	129.94
2	744bade1fcf9ff3f31d860ace07...	2	credit_card	0	58.69
3	8bcbe01d44d147f901cd31926...	4	voucher	1	0.0
4	fa65dad1b0e818e3ccc5cb0e3...	14	voucher	1	0.0
5	6ccb433e00daae1283ccc9561...	4	voucher	1	0.0
6	4637ca194b6387e2d538dc89b...	1	not_defined	1	0.0
7	00b1cb0320190ca0daa2c88b3...	1	not_defined	1	0.0
8	45ed6e85398a87c253db47c2d...	3	voucher	1	0.0
9	fa65dad1b0e818e3ccc5cb0e3...	13	voucher	1	0.0
10	c8c528189310eaa44a745b8d9...	1	not_defined	1	0.0

Results per page: 50 | 1 – 50 of 1000

Products:

```
SELECT column_name, data_type
FROM `business-case-study-sql.Target_dataset.INFORMATION_SCHEMA.COLUMNS`
WHERE table_name = 'products'
```

Query results

SAVE RESULTS | EXPLORE DATA

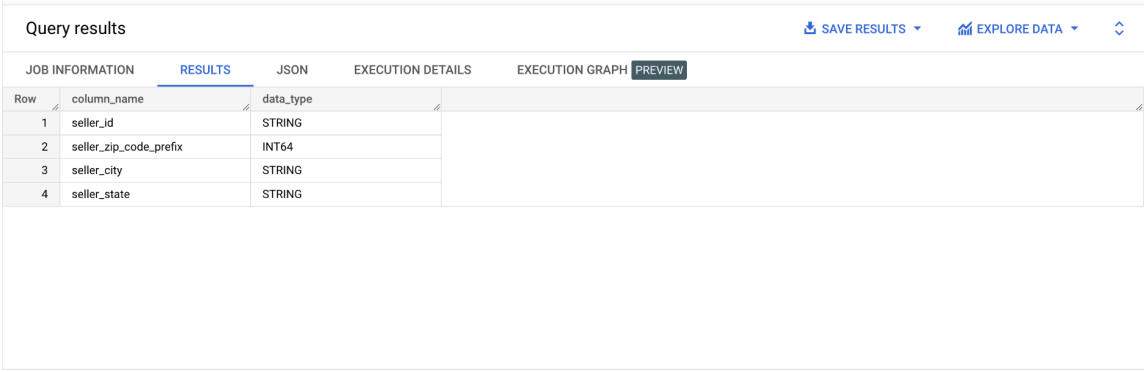
JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

Row	product_id	product_category	product_name_l	product_descrip	product_photos	product_weight	product_length	product_height	product_width_c
1	5eb564652db742ff8f28759cd8...	null	null	null	null	null	null	null	null
2	09ff539a621711667c43eba6a...	babies	60	865	3	null	null	null	null
3	2f763ba79d9cd987b2034aac7...	electronics	45	1198	2	595	8	6	6
4	a69f15dfb803d485e8933e80b...	Watches present	53	506	6	150	11	16	6
5	e1cfc87f543782b8a78b59fc85...	Garden tools	39	524	4	369	26	7	7
6	106392145fca363410d287a81...	bed table bath	58	309	1	2083	12	2	7
7	7e33f4a1c59f89da30a335b2d...	electronics	51	381	3	1075	22	5	7
8	bc9cc914f974963c07be697fc...	HEALTH BEAUTY	55	435	1	75	14	9	7
9	5ae533eac9c0e93b3f89bc9ae...	computer accessories	58	1340	1	83	12	8	8
10	67d1a56495104e195338ec900...	pet Shop	20	2153	1	275	8	13	8

Results per page: 50 | 1 – 50 of 1000

Sellers:

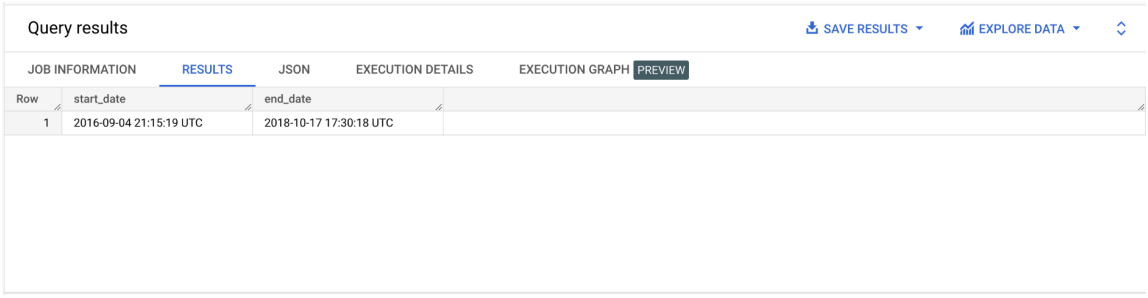
```
SELECT column_name, data_type
FROM `business-case-study-sql.Target_dataset.INFORMATION_SCHEMA.COLUMNS`
WHERE table_name = 'sellers'
```



Row	column_name	data_type
1	seller_id	STRING
2	seller_zip_code_prefix	INT64
3	seller_city	STRING
4	seller_state	STRING

1.2)Time period for which the data is given

```
SELECT
 MIN(order_purchase_timestamp) AS start_date,
 MAX(order_purchase_timestamp) AS end_date
FROM `business-case-study-sql.Target_dataset.orders`
```



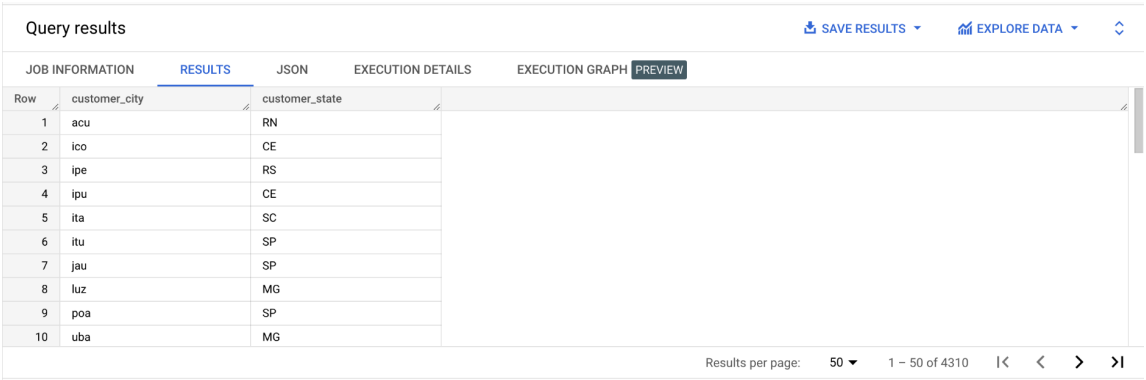
Row	start_date	end_date
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC

Insights:

Start date = 04-09-2016
End date = 17-10-2018
Time period = 2 years, 7 months, 14 days including the end date.

1.3)Cities and States of customers ordered during the given period

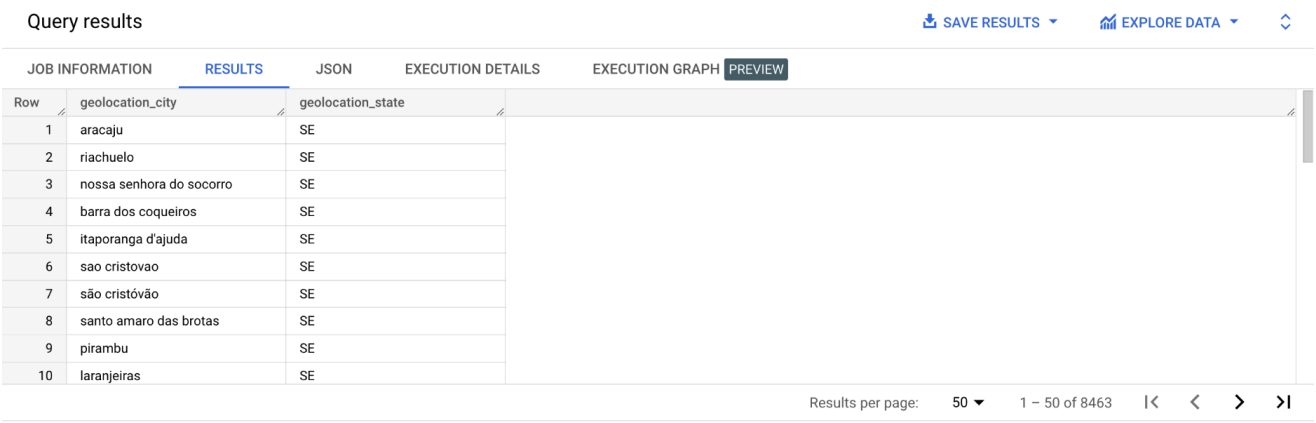
```
SELECT DISTINCT customer_city, customer_state
FROM `business-case-study-sql.Target_dataset.customers` AS c
JOIN `business-case-study-sql.Target_dataset.orders` AS o
ON c.customer_id = o.customer_id
```



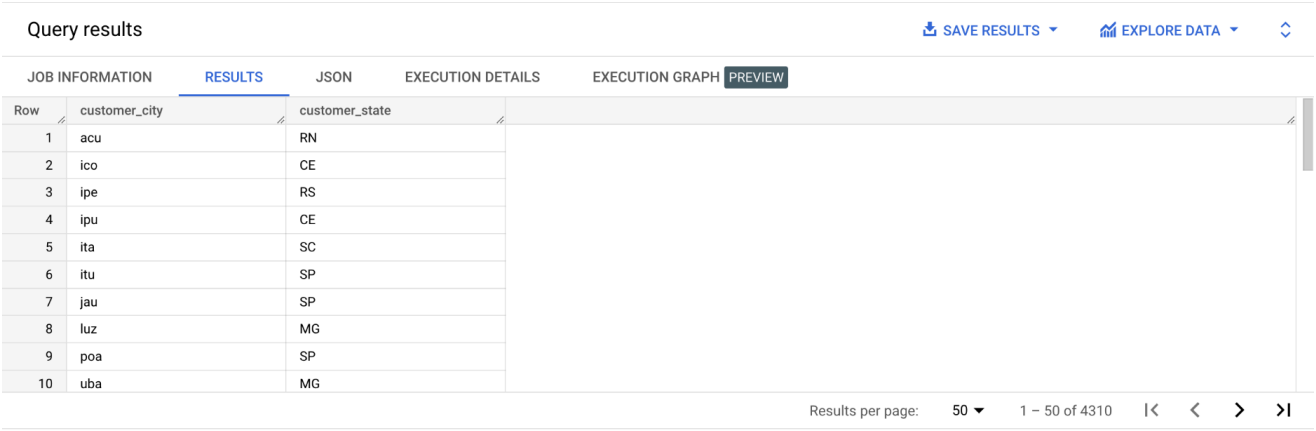
Insights:

There are total 4311 rows.
Count of unique cities = 4119
Count of unique states = 27

```
SELECT DISTINCT geolocation_city, geolocation_state
FROM `business-case-study-sql.Target_dataset.geolocation`
```



```sql
SELECT DISTINCT customer_city, customer_state
FROM `business-case-study-sql.Target_dataset.customers`
```



Query results

SAVE RESULTS | EXPLORE DATA

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

Row	customer_city	customer_state
1	acu	RN
2	ico	CE
3	ipe	RS
4	ipu	CE
5	ita	SC
6	itu	SP
7	jau	SP
8	luz	MG
9	poa	SP
10	uba	MG

Results per page: 50 | 1 – 50 of 4310

Insights:

There are total 8464 rows.
Count of unique cities = 4119
Count od unique states = 27

Q2) In-depth Exploration:
2.1)Is there a growing trend on e-commerce in Brazil?
How can we describe a complete scenario?
Can we see some seasonality with peaks at specific months?

```sql
SELECT
 EXTRACT(MONTH FROM order_purchase_timestamp) AS order_month,
 EXTRACT(YEAR FROM order_purchase_timestamp) AS order_year,
 COUNT(DISTINCT order_id) AS order_count,
 SUM(DISTINCT(order_id)) AS total_sales
FROM `business-case-study-sql.Target_dataset.orders`
GROUP BY order_month, order_year
ORDER BY order_year, order_month
```

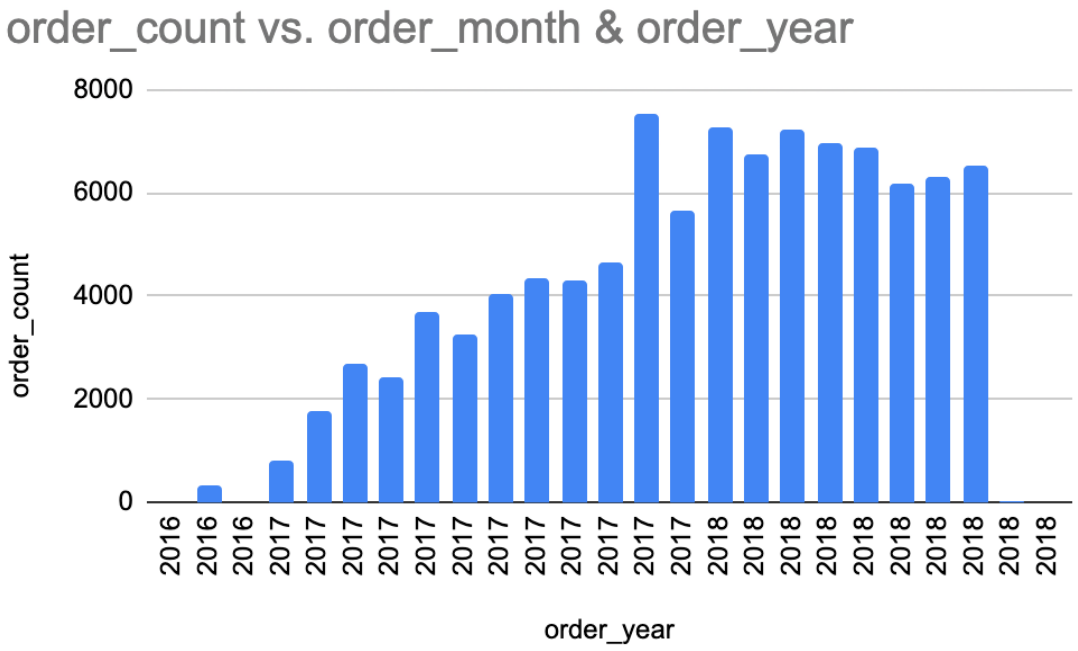
Query results

SAVE RESULTS | EXPLORE DATA

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

Row	order_month	order_year	order_count
1	9	2016	4
2	10	2016	324
3	12	2016	1
4	1	2017	800
5	2	2017	1780
6	3	2017	2682
7	4	2017	2404
8	5	2017	3700
9	6	2017	3245
10	7	2017	4026

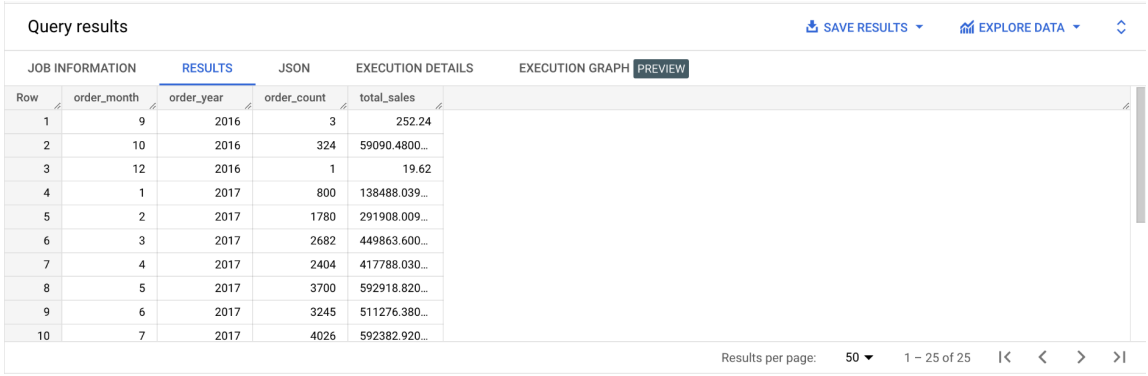
Results per page: 50 | 1 – 25 of 25

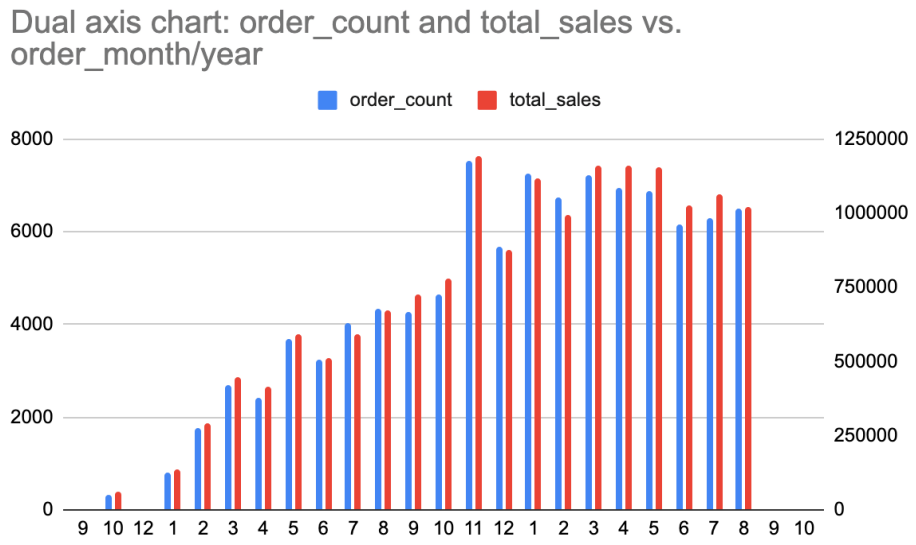


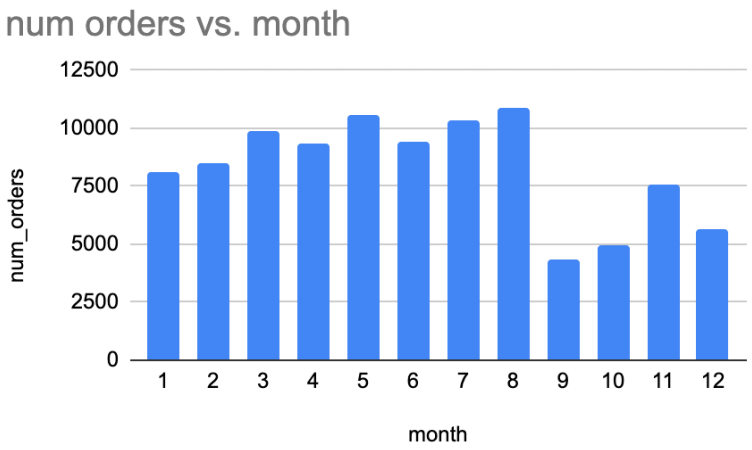
Insights:

1)We can clearly see that that there a growing trend on e-commerce in Brazil from 2016 to 2018.
2)Nov 2017 is the month when most of the orders were placed.
3)Outlier data points: (09 &12)in 2016 and (09 & 10) in 2018
Here, the number of order_count is very less than the median data. It seems the data was not captured correctly and looks corrupted. If these datapoints are not corrected then it may negatively impact the analysis of the entire dataset.

```sql
SELECT
 EXTRACT(MONTH FROM order_purchase_timestamp) AS order_month,
 EXTRACT(YEAR FROM order_purchase_timestamp) AS order_year,
 COUNT(DISTINCT c.order_id) AS order_count,
 SUM(p.payment_value) AS total_sales
FROM `business-case-study-sql.Target_dataset.orders` AS c
JOIN `business-case-study-sql.Target_dataset.payments` AS p
ON c.order_id = p.order_id
GROUP BY
 order_month, order_year
ORDER BY
 order_year, order_month
```







Insights:

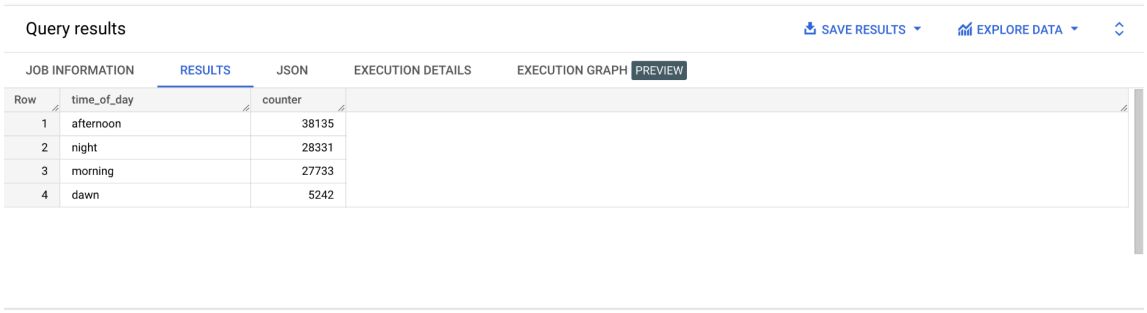
1)Seasonality chart (Month wise date added from Sept 2016 to Oct 2018)
1.1)August month has the highest number of orders.
(Maybe there is Big Sale Offer/Stock Clearance offer)
1.2)September month has the lowest number of orders.

2)The average number of orders for the first 8 months is higher than last 4 months.
Outlier data points: (09 &12)in 2016 and (09 & 10) in 2018
Here, the number of order_count is very less than the median data. It seems the data was not captured correctly and looks corrupted. If these datapoints are not corrected then it may negatively impact the analysis of the entire dataset.

3)Since Brazil lies in South Hemisphere, summer is December through March and winter June through September.
So, we observe that the number of orders are more in winter(June to September) than summer season(December to March).
Therefore, we observe that the number of orders depends on the seasons.

2.2) What time do Brazilian customers tend to buy
(Dawn - 12am-6am, Morning 6am-12pm, Afternoon - 12 noon to 6pm,or Night 6-pm - 12am)?

```sql
SELECT
CASE
WHEN EXTRACT(hour FROM timestamp(order_purchase_timestamp)) BETWEEN 0 AND 6
THEN 'dawn'
WHEN EXTRACT(hour FROM timestamp(order_purchase_timestamp)) BETWEEN 7 AND 12
THEN 'morning'
WHEN
EXTRACT(hour FROM timestamp(order_purchase_timestamp)) BETWEEN 13 AND 18
THEN 'afternoon'
WHEN
EXTRACT(hour FROM timestamp(order_purchase_timestamp)) BETWEEN 19 AND 23
THEN 'night'
END AS time_of_day,
COUNT(DISTINCT order_id) AS counter
FROM `business-case-study-sql.Target_dataset.orders`
GROUP BY 1
ORDER BY 2 DESC;
```



Row	time_of_day	counter
1	afternoon	38135
2	night	28331
3	morning	27733
4	dawn	5242

Insights:
Here the time slots are:
Dawn:- 12am-6am, Morning:- 7am-12pm, Afternoon:- 1pm to 6pm, Night:- 7pm - 11pm

```sql
SELECT
 CASE
   WHEN TIME(order_purchase_timestamp) BETWEEN '00:00:00' AND '06:59:59'
   THEN 'Dawn'
   WHEN TIME(order_purchase_timestamp) BETWEEN '07:00:00' AND '12:59:59'
   THEN 'Morning'
   WHEN TIME(order_purchase_timestamp) BETWEEN '13:00:00' AND '18:59:59'
   THEN 'Afternoon'
   ELSE 'Night'
   END AS TIME_OF_PURCHASE,
COUNT(DISTINCT order_id) AS num_orders
FROM `business-case-study-sql.Target_dataset.orders`
GROUP BY TIME_OF_PURCHASE
order by num_orders
```

Query results

SAVE RESULTS | EXPLORE DATA

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

Row	TIME_OF_PURCHASE	num_orders
1	Dawn	5242
2	Morning	27733
3	Night	28331
4	Afternoon	38135

Here the time slots are:
Dawn - 12:00:00 am-6:59:59 am, Morning 07:00:00 am-12:59:59 pm, Afternoon - 01:00:00 pm to 6:59:59 pm, or Night 07:00:00 pm - 11:59:59 pm

```
SELECT
CASE
 WHEN TIME(order_purchase_timestamp) BETWEEN '00:00:00' AND '05:59:59' THEN 'Dawn'
 WHEN TIME(order_purchase_timestamp) BETWEEN '06:00:00' AND '11:59:59' THEN 'Morning'
 WHEN TIME(order_purchase_timestamp) BETWEEN '12:00:00' AND '17:59:59' THEN 'Afternoon'
 ELSE 'Night'
END AS TIME_OF_PURCHASE,
COUNT(DISTINCT order_id) AS num_orders
FROM `business-case-study-sql.Target_dataset.orders`
GROUP BY TIME_OF_PURCHASE
order by num_orders
```

Query results

SAVE RESULTS | EXPLORE DATA

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

Row	TIME_OF_PURCHASE	num_orders
1	Dawn	4740
2	Morning	22240
3	Night	34100
4	Afternoon	38361

Here the time slots are:
Dawn - 12:00:00 am-5:59:59 am, Morning 06:00:00 am-11:59:59 pm, Afternoon - 12:00:00 pm to 5:59:59 pm, or Night 06:00:00 pm - 11:59:59 pm

Insights:

1)During the afternoon time (12pm-6pm), most of the customers place order.
Reason: People are awake during this time period.
So, most advertisements can be shown during this time period.

2)During morning and night, almost equal amount of orders are placed.

3)During dawn timing(12am-6am), very less people order.
Reason: People are mostly asleep during this time.

Q3)Evolution of E-commerce orders in the Brazil region:
3.1)Get month on month orders by states

```sql
SELECT
EXTRACT(month FROM timestamp(order_purchase_timestamp)) AS month,
g.geolocation_state,
COUNT(1) AS num_orders
FROM `business-case-study-sql.Target_dataset.orders` o
INNER JOIN `business-case-study-sql.Target_dataset.customers` c
ON o.customer_id = c.customer_id
INNER JOIN `business-case-study-sql.Target_dataset.geolocation` g
ON c.customer_zip_code_prefix = g.geolocation_zip_code_prefix
GROUP BY g.geolocation_state, month
ORDER BY geolocation_state DESC, month ASC
```

Query results

SAVE RESULTS | EXPLORE DATA

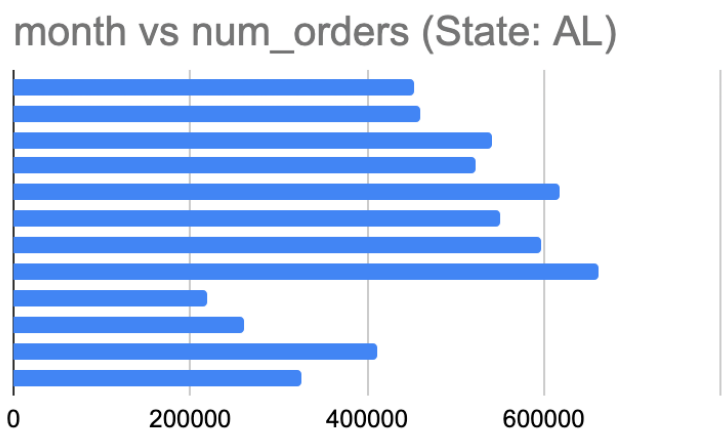
JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

Row	month	geolocation_state	num_orders
1	1	TO	1544
2	2	TO	1287
3	3	TO	1626
4	4	TO	2379
5	5	TO	2691
6	6	TO	1577
7	7	TO	743
8	8	TO	1603
9	9	TO	1236
10	10	TO	1010

Results per page: 50 1 – 50 of 322

Sheets Plot:

Most of the orders are from AL state.
Here, is the month vs num_orders plot for AL state.



Least of the orders are from SE state.
Here, is the month vs num_orders plot for SE state.

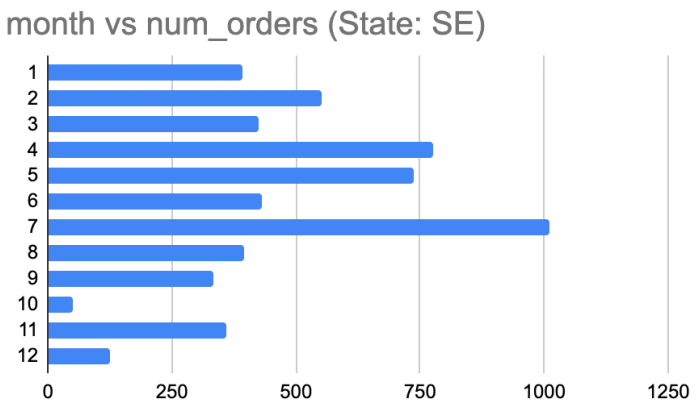
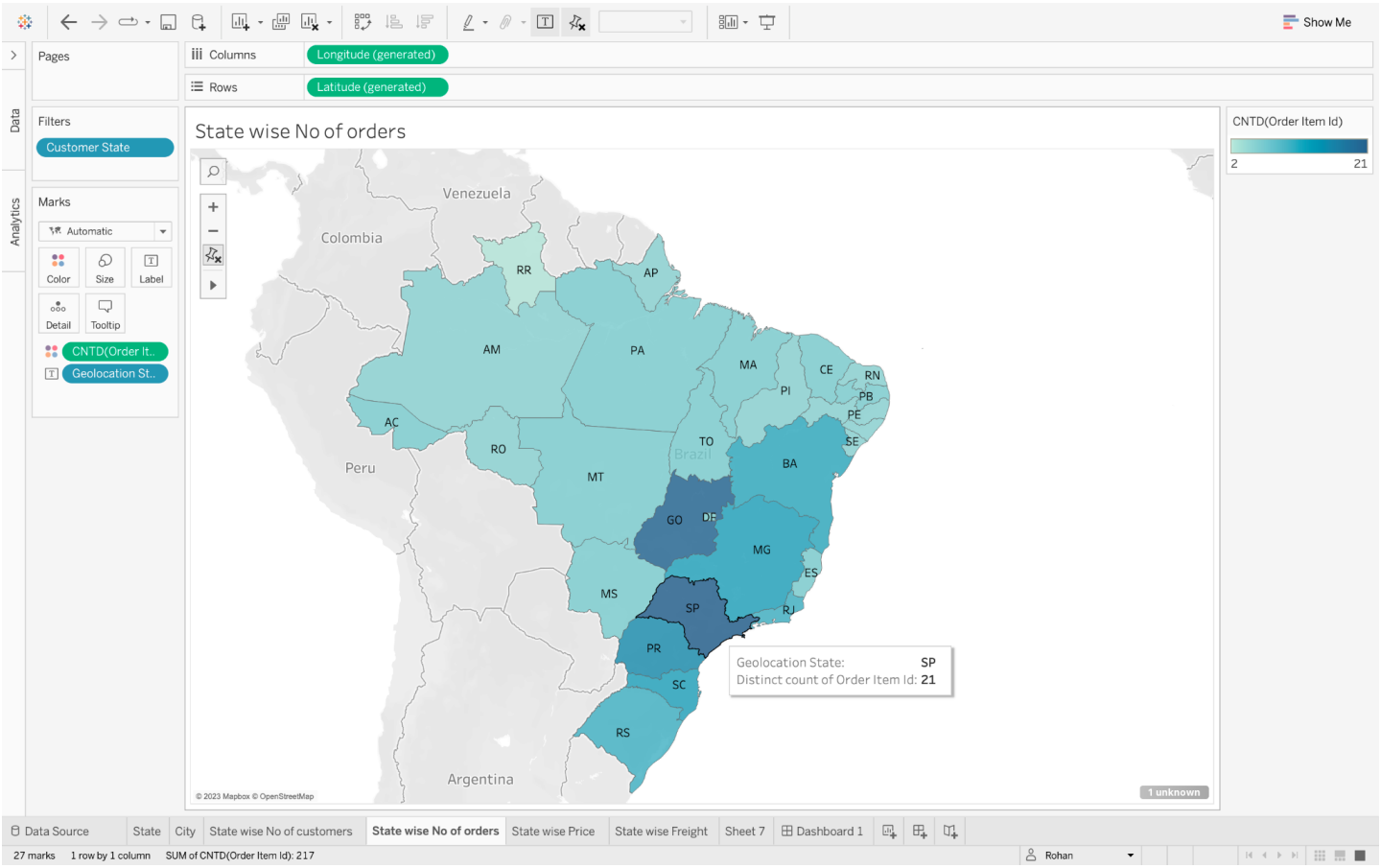


Tableau visualization:



3.2)Distribution of customers across the states in Brazil

```sql
SELECT g.geolocation_state, COUNT(DISTINCT (c.customer_unique_id)) AS num_customers
FROM `business-case-study-sql.Target_dataset.customers` c
INNER JOIN `business-case-study-sql.Target_dataset.geolocation` g
ON c.customer_zip_code_prefix = g.geolocation_zip_code_prefix
GROUP BY g.geolocation_state
ORDER BY num_customers DESC;
```

Query results

SAVE RESULTS | EXPLORE DATA

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

Row	geolocation_state	num_customers
1	SP	40287
2	RJ	12372
3	MG	11248
4	RS	5284
5	PR	4871
6	SC	3547
7	BA	3268
8	ES	1959
9	GO	1944
10	DF	1913

Results per page: 50 1 – 27 of 27

Sheets plot:

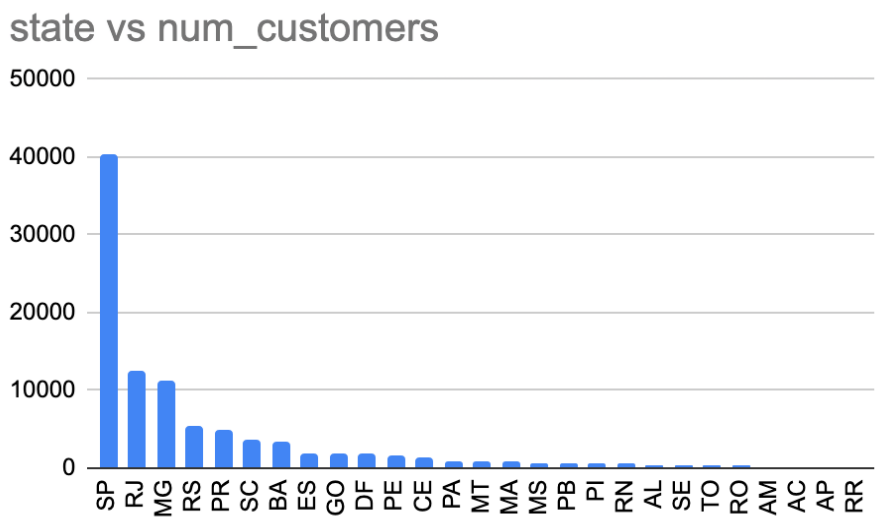
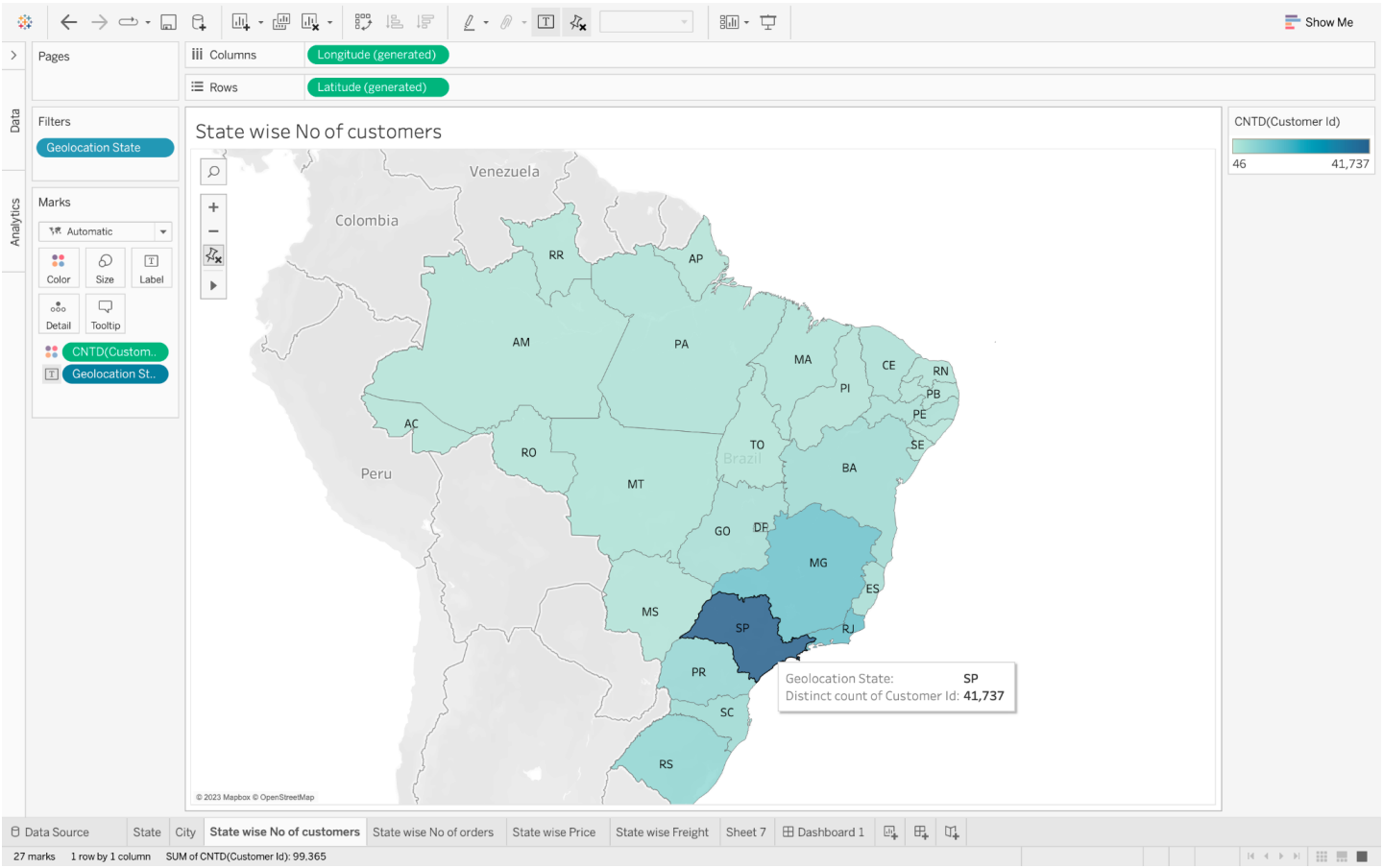


Tableau visualization:



Insights:

The Top most number of orders is placed from the SP state.
The 2nd most number of orders is placed from the RJ state.
The 3rd most number of orders is placed from the MG state.
We should further analyse the main reasons for the most number of orders.
Can we increase the number of orders even more ? If yes, then how?

The least number of orders is placed from the RR state.
We should further analyse the main reasons for the least number of orders .
What can we do to increase the number of orders?

We observe that:
The state SP has the maximum number of customers & the maximum number of orders.
Hence, the maximum number of sales.
The state GO has very less number of customers but has very large amount of order.

Q4)Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others (*charges*).
4.1)Get % increase in cost of orders (*refer to payment_value column in payment table*) from **2017 to 2018** (include months between **Jan to Aug only**(*month between 1 to 8*)) - You can use “payment_value” column in payments table

Approach:
Create CTE Table and new columns:
price_per_order = sum(price)/count(order_id)
freight_per_order= sum(freight_value)/count(order_id)
Group the data on yearly and monthly level

```
WITH
cte_table AS (
SELECT
EXTRACT(month FROM timestamp(o.order_purchase_timestamp)) AS month,
EXTRACT(year FROM timestamp(o.order_purchase_timestamp)) AS year,
(sum(price) / COUNT( distinct o.order_id)) AS price_per_order,
(sum(freight_value) / COUNT(distinct o.order_id)) AS freight_per_order
FROM `business-case-study-sql.Target_dataset.orders` o
INNER JOIN `business-case-study-sql.Target_dataset.order_items` i
ON o.order_id = i.order_id
GROUP BY year, month
)
SELECT (price_per_order), (freight_per_order), month, year
FROM cte_table
order by year asc, month asc ;
```

Query results

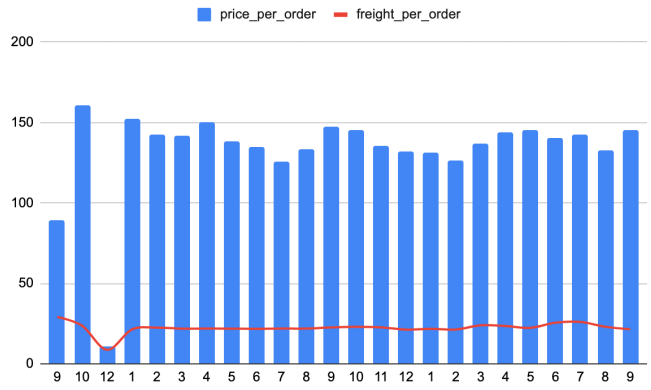
SAVE RESULTS | EXPLORE DATA

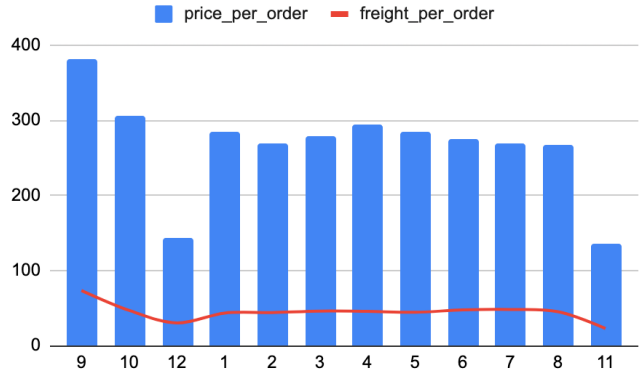
JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

Row	price_per_order	freight_per_orde	month	year
1	89.12	29.13	9	2016
2	160.739155...	23.7051298...	10	2016
3	10.9	8.72	12	2016
4	152.487794...	21.3886185...	1	2017
5	142.702261...	22.4914021...	2	2017
6	141.743392...	21.8494093...	3	2017
7	150.534182...	21.9552530...	4	2017
8	138.270803...	21.8906584...	5	2017
9	134.609449...	21.7359154...	6	2017
10	125.480342...	21.9047971...	7	2017

Results per page: 50 | 1 – 24 of 24

Sheets Plot:





Insights:

We observe that:

For Year wise plot

1)For price_per_order

From 2016 to 2017:- there is significant growth of

From 2017 to 2018:- there is slight decrease of

2) For freight_per_order

From 2016 to 2017:- there is increase of

From 2017 to 2018:- there is slight decrease of

For month on month plot:

For Year wise plot

1)For price_per_order

September is the highest

November & December is the lowest

2) For freight_per_order

September is the highest

November & December is the lowest

Note:

Freight transport, also referred as freight forwarding, is the physical process of transporting commodities and merchandise goods and cargo.

A freight rate is a price at which a certain cargo is delivered from one point to another.

The price depends on the form of the cargo, the mode of transport (truck, ship, train, aircraft), the weight of the cargo, and the distance to the delivery destination.

Here,
price_per_order = freight_per_order + other charges
Therefore,
other charges = price_per_order - freight_per_order

Total amount sold in 2017 between Jan to August (Jan to Aug because data is available starting 2017 01 to 2018 08) and we can only compare cycles with cycles
Compare YoY at a monthly level

```
WITH
cte_table AS (
 SELECT
 EXTRACT(month FROM timestamp(order_purchase_timestamp)) AS month,
 EXTRACT(year FROM timestamp(order_purchase_timestamp)) AS year,
 sum(price) AS total_price,
 sum(freight_value) AS total_freight
 FROM `business-case-study-sql.Target_dataset.orders` o
 INNER JOIN `business-case-study-sql.Target_dataset.order_items` i
 ON o.order_id = i.order_id
 GROUP BY year, month
 ORDER BY year ASC, month ASC
)
SELECT
 month,
 price_2017,
 price_2018,
 round((price_2018 - price_2017) / price_2017 * 100, 2) AS yoy
FROM
(
 SELECT
 month,
 sum(CASE WHEN year = 2017 THEN total_price ELSE 0 END) AS price_2017,
 sum(CASE WHEN year = 2018 THEN total_price ELSE 0 END) AS price_2018
-- sum(total_price) as total_transaction_amt
FROM cte_table
WHERE (year = 2017 OR year = 2018) AND month BETWEEN 1 AND 8
GROUP BY month
order by month
);
```

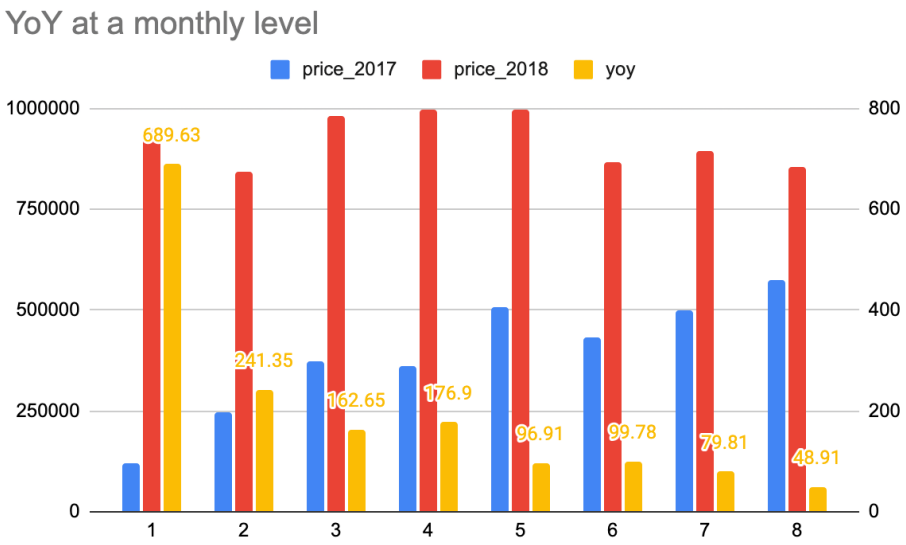
Query results

SAVE RESULTS | EXPLORE DATA

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

Row	month	price_2017	price_2018	yoy
1	1	120312.869...	950030.360...	689.63
2	2	247303.019...	844178.710...	241.35
3	3	374344.300...	983213.440...	162.65
4	4	359927.230...	996647.750...	176.9
5	5	506071.140...	996517.680...	96.91
6	6	433038.600...	865124.310...	99.78
7	7	498031.480...	895507.220...	79.81
8	8	573971.680...	854686.330...	48.91

Sheets plot:



Insights:

We observe that:
For YoY at amonthly level (from 2017 to 2018).
For January, the Total amount sold change is maximum with ~690% increase.
This is a huge growth. What are the main factors driving this massive growth?

In general, for all months there is increase in the Total amount sold (from 2017 to 2018).
This means that Brazil is a growing economy for e-commerce market.

MoM increase for year 2017

```sql
SELECT
 month, orders, lagger_orders,
 (orders - coalesce(lagger_orders, 0)) / coalesce(lagger_orders, 1) * 100 AS difference
FROM
(
 SELECT *, lag(orders, 1) OVER (ORDER BY month ASC) AS lagger_orders
 FROM
 (
   SELECT
   EXTRACT(month FROM timestamp(a.order_purchase_timestamp)) AS month,
   COUNT(DISTINCT a.order_id) AS orders,
   COUNT(DISTINCT b.customer_unique_id) AS customers
   FROM `business-case-study-sql.Target_dataset.orders` a
   LEFT JOIN `business-case-study-sql.Target_dataset.customers` b
   ON a.customer_id = b.customer_id
   WHERE EXTRACT(year FROM timestamp(a.order_purchase_timestamp)) = 2017
   GROUP BY 1
 ) base
) base_2
ORDER BY month ASC;
```

Query results

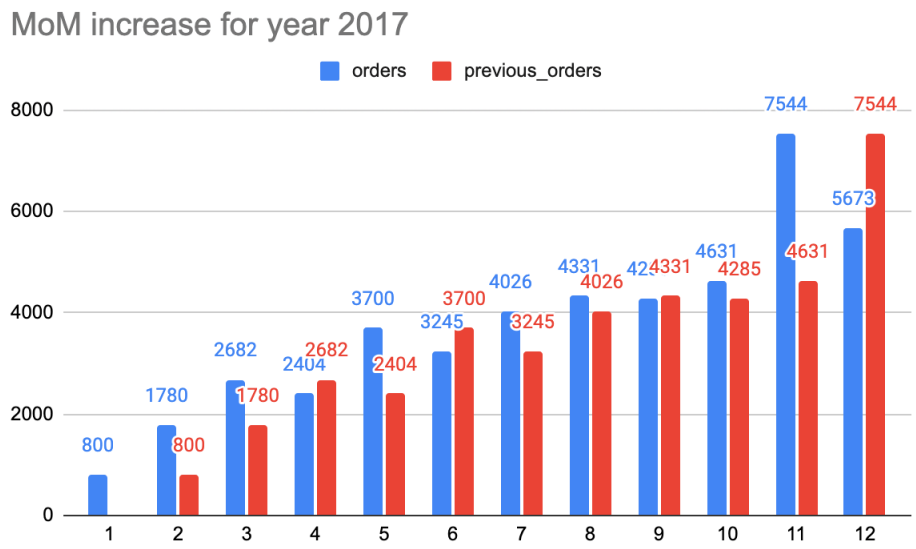
SAVE RESULTS | EXPLORE DATA

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

Row	month	orders	previous_orders	difference
1	1	800	null	80000.0
2	2	1780	800	122.500000...
3	3	2682	1780	50.6741573...
4	4	2404	2682	-10.365398...
5	5	3700	2404	53.9101497...
6	6	3245	3700	-12.297297...
7	7	4026	3245	24.0677966...
8	8	4331	4026	7.57575757...

Load more

Results per page: 50 | 1 – 12 of 12



Insights:

We observe that:
For the year 2017, there is month on month increase in Total sales for all the months.
Moreover, the amount of increase per month is also increasing across the month (in general).

4.2)Mean(avg) & Sum of price and freight value by customer state

Sum and mean price by customer state
It's very interesting to see how some states have a high total amount sold and a low price per order.
If we look at SP (São Paulo) for example, it's possible to see that it
is the state with most valuable state for e-commerce (5202955 sold)
but it is also where customers pay less per order (125.75 per order)

```
with cte_table as (
 select
 c.customer_state as state,
 sum(price) as total_price,
 count(distinct(o.order_id)) as num_orders
 from `business-case-study-sql.Target_dataset.orders` o
 inner join `business-case-study-sql.Target_dataset.order_items` i
 on o.order_id= i.order_id
 inner join `business-case-study-sql.Target_dataset.customers` c
 on o.customer_id=c.customer_id
 group by state
)
select state, total_price, num_orders,(total_price/num_orders) as
avg_price
from cte_table
order by total_price desc;
```

Query results

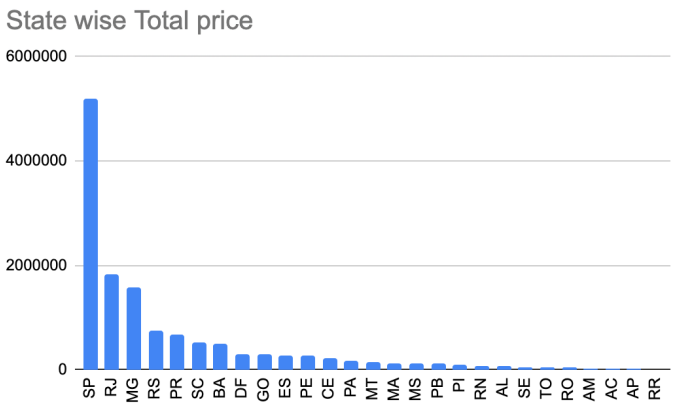
SAVE RESULTS | EXPLORE DATA

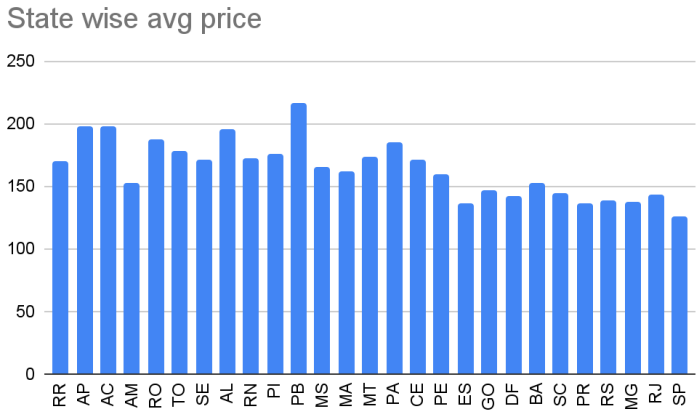
JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

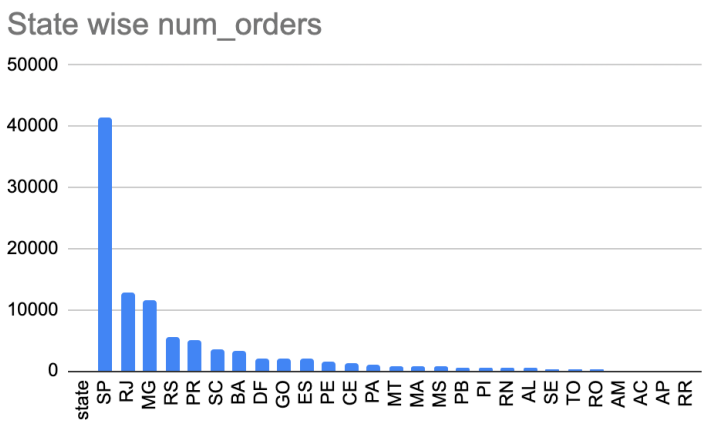
Row	state	total_price	num_orders	avg_price
1	SP	5202955.05...	41375	125.751179...
2	RJ	1824092.66...	12762	142.931567...
3	MG	1585308.02...	11544	137.327445...
4	RS	750304.020...	5432	138.126660...
5	PR	683083.760...	4998	136.671420...
6	SC	520553.340...	3612	144.117757...
7	BA	511349.990...	3358	152.278138...
8	DF	302603.939...	2125	142.401854...
9	GO	294591.949...	2007	146.782237...
10	ES	275037.309...	2025	135.820893...

Results per page: 50 1 – 27 of 27

Sheets plot:







Insights:

We observe that:

1)For state wise total price

SP is the highest

RR is the lowest

2)For state wise avg price

PB is the highest

SP is the lowest

3)For state wise num_orders

SP is the highest

RR is the lowest

So, for SP num_orders is the highest & avg_price is lowest.

Hence, Total price is maximum.

This means customers prefer more number of lower price products.

Q5) Analysis on sales, freight and delivery time

1. **Calculate days between purchasing, delivering and estimated delivery:**

   Find time_to_delivery & diff_estimated_delivery. Formula for the same given below:

   - time_to_delivery = order_purchase_timestamp-order_delivered_customer_date
   - diff_estimated_delivery = order_estimated_delivery_date-order_delivered_customer_date

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

   Sort the data to get the following:

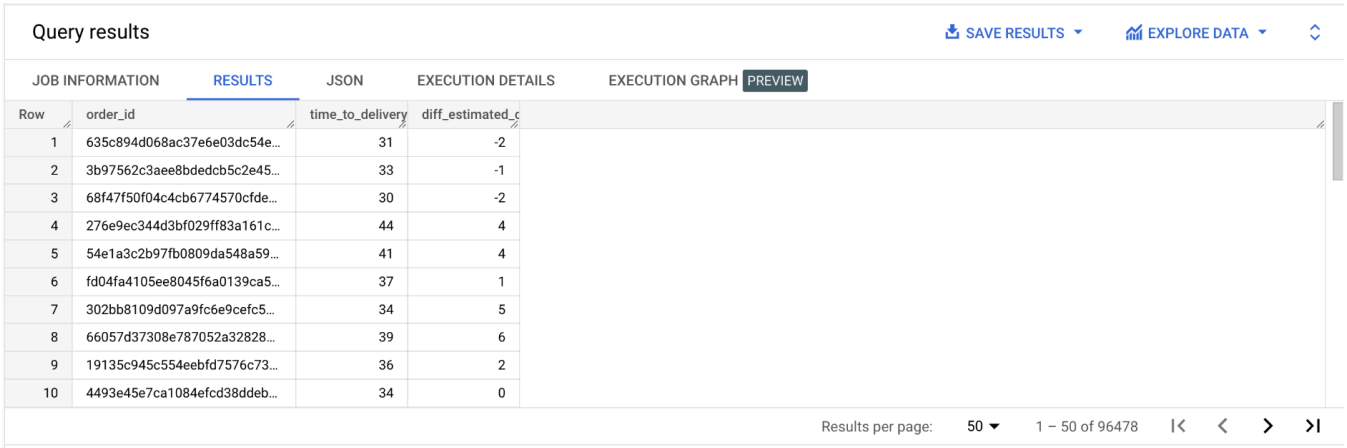
3. Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5
4. Top 5 states with highest/lowest average time to delivery

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

5.1)Calculate days between purchasing, delivering and estimated delivery:

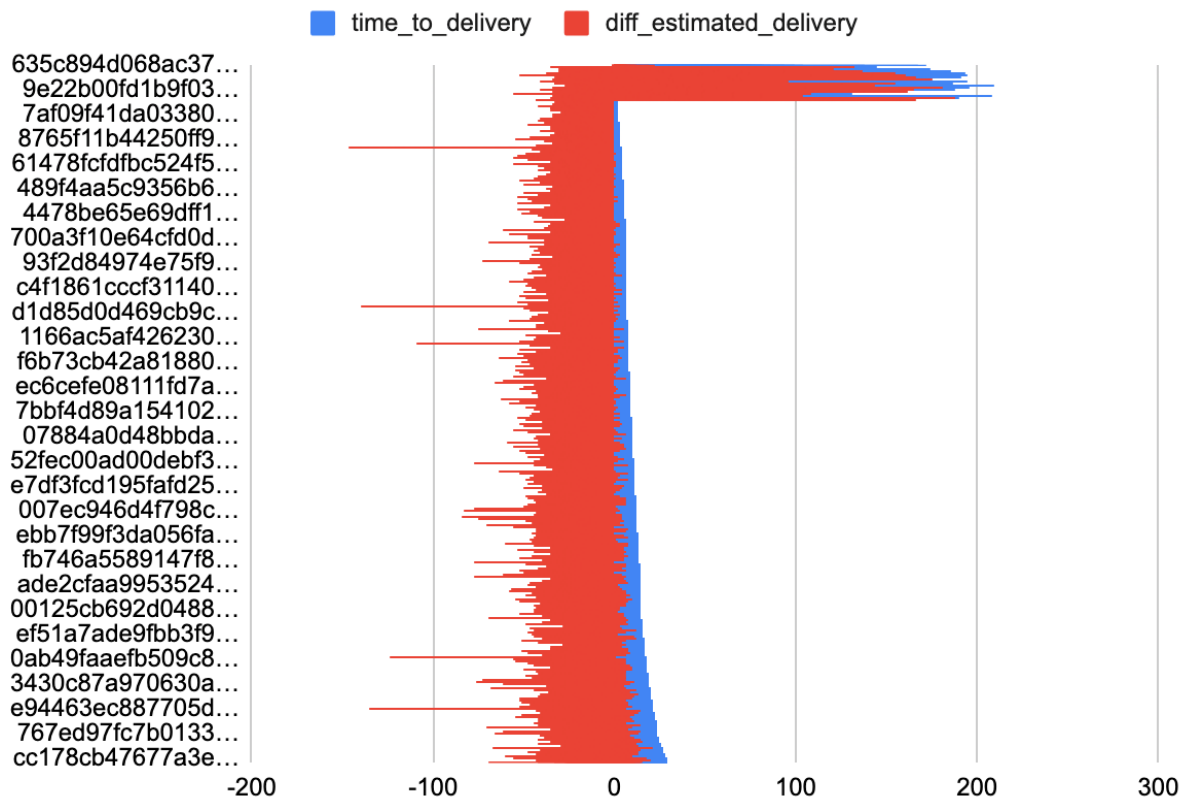
Approach:
create new columns for time to delivery and difference in estimated vs actual delivery

```
SELECT
order_id,
date_DIFF(date(order_delivered_customer_date),date(order_purchase_timestamp),DAY) AS
time_to_delivery,
date_DIFF(date(order_delivered_customer_date),date(order_estimated_delivery_date),DAY) AS
diff_estimated_delivery
FROM `business-case-study-sql.Target_dataset.orders`
WHERE order_status = 'delivered';
```



Row	order_id	time_to_delivery	diff_estimated_c
1	635c894d068ac37e6e03dc54e...	31	-2
2	3b97562c3aee8bdedcb5c2e45...	33	-1
3	68f47f50f04c4cb6774570cfde...	30	-2
4	276e9ec344d3bf029ff83a161c...	44	4
5	54e1a3c2b97fb0809da548a59...	41	4
6	fd04fa4105ee8045f6a0139ca5...	37	1
7	302bb8109d097a9fc6e9cefc5...	34	5
8	66057d37308e787052a32828...	39	6
9	19135c945c554eebfd7576c73...	36	2
10	4493e45e7ca1084efcd38ddeb...	34	0

Sheets plot:



Using basic statistics to get the insights from data:

	Time_to_delivery (days)	Diff_estimated_delivery (days)
Maximum	210	188
Minimum	0	-56
Average	12.49	-11.87
Median	10	-6
90th percentile	23	-2
95th percentile	29	3
99th percentile	83	18
99.9th percentile	83	55
99.99th percentile	187.35	156.41

Insights:

We can clearly observe the maximum, minimum, average, median time to delivery & difference in estimated vs actual delivery (in days).
According to Median value, for delivery it takes 10 days.
And difference in estimated vs actual delivery (in days) is 6 days before actual delivery date.

The delivery team shold work on the 95th percentile (~29 days) to 99.9th percentile(~83 days) range.
The logistics team should identify the reason for such huge delay in delivery time (29 days-83 days) and
Find relevant ways to reduce this huge delivery time.

The delivery team shold work on the 99th percentile (~18 days) to 99.9th percentile(~55 days) range.
The logistics team should identify the reason for such huge delay in difference in estimated vs actual delivery time (18 days-55 days) and
Find relevant ways to reduce this huge delivery time.

The maximum delivery time(~210 days) can be considered as an outlier.
The maximum difference in estimated vs actual delivery time(~188 days) can be considered as an outlier.

Faster delivery will result in customer satisfaction.
This leads to more number of orders per customers.
Therefore, the total sales will increase.

5.2) Group data by state, take mean of freight_value, time_to_delivery, diff_estimated_delivery

```
SELECT
 c.customer_state AS state,
 AVG(it.freight_value) AS avg_freight_value,
 AVG(date_DIFF(date(o.order_delivered_customer_date), date(o.order_purchase_timestamp), DAY)) AS
avg_time_to_delivery,
 AVG(date_DIFF(date(o.order_estimated_delivery_date), date(o.order_delivered_customer_date), DAY))
AS avg_diff_estimated_delivery
FROM `business-case-study-sql.Target_dataset.orders` AS o
INNER JOIN `business-case-study-sql.Target_dataset.customers` AS c
ON o.customer_id = c.customer_id
INNER JOIN `business-case-study-sql.Target_dataset.order_items` AS it
ON o.order_id = it.order_id
WHERE order_status = 'delivered'
GROUP BY state;
```

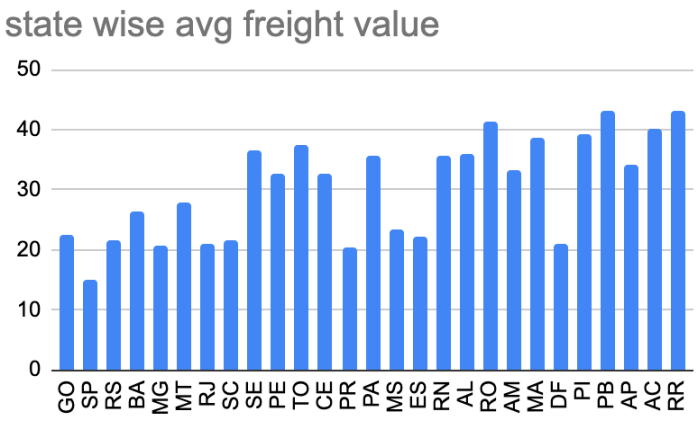
Query results

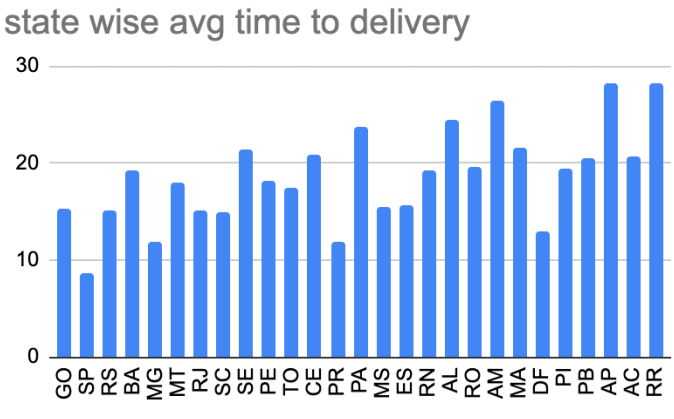
SAVE RESULTS | EXPLORE DATA

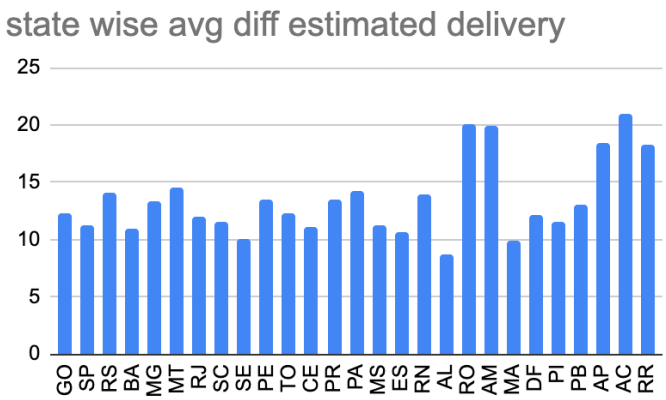
JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

Row	state	avg_freight_valu	avg_time_to_del	avg_diff_estimat
1	GO	22.5628678...	15.3355292...	12.2942468...
2	SP	15.1151823...	8.66232423...	11.2064555...
3	RS	21.6131920...	15.1345181...	14.1341920...
4	BA	26.4875563...	19.1925061...	10.9826228...
5	MG	20.6263425...	11.9193248...	13.3446113...
6	MT	27.9969141...	17.9074252...	14.5718418...
7	RJ	20.9114360...	15.0741709...	12.0098988...
8	SC	21.5073590...	14.9463021...	11.5684647...
9	SE	36.5731733...	21.4186666...	10.0026666...

Results per page: 50 1 – 27 of 27







Insights:

We observe that:
1)For state wise avg freight value
Maximum value = state RR
Minimum value = state SP

2)For state wise avg time to delivery
Maximum value = RR (~28 days)
Minimum value = SP (~8 days)

3)For state wise difference in estimated vs actual delivery time
Maximum value = AC (~22 days)
Minimum value = AL (~8 days)

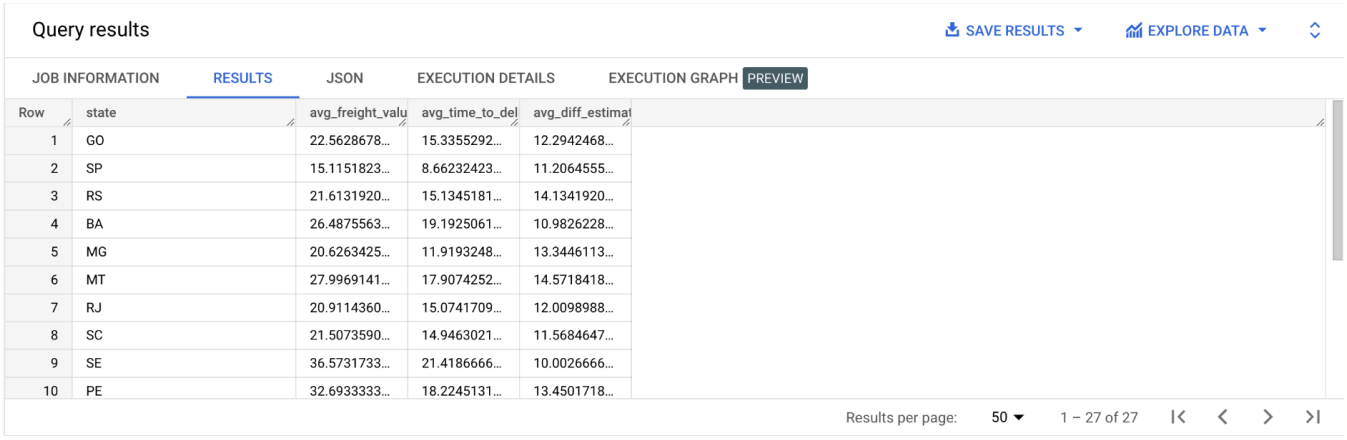
Therefore, we can infer that
due to maximum avg freight value and maximum avg time to delivery time the state RR has lowest sales.
due to minimum avg freight value and minimum avg time to delivery time the state SP has lowest sales.

Hence, to increase the total sales the avg freight value shold be minimized and avg time to delivery time should be minimized.

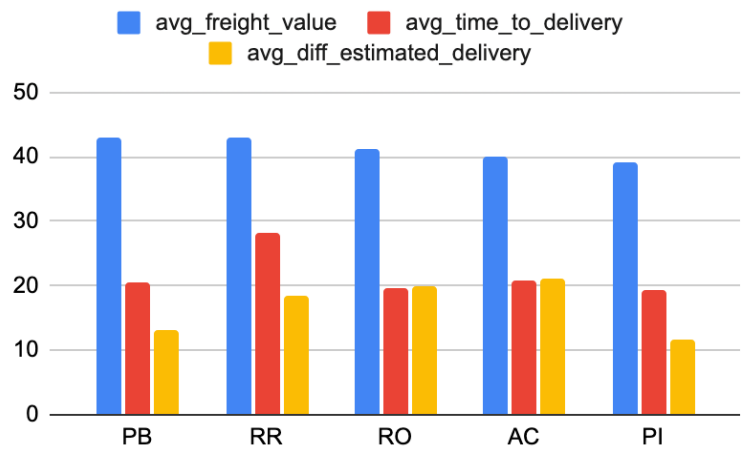
5.3) Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5

```sql
--ORDER BY avg_freight_value DESC

SELECT
 c.customer_state AS state,
 AVG(it.freight_value) AS avg_freight_value,
 AVG(date_DIFF(date(o.order_delivered_customer_date), date(o.order_purchase_timestamp), DAY)) AS
avg_time_to_delivery,
 AVG(date_DIFF(date(o.order_estimated_delivery_date), date(o.order_delivered_customer_date), DAY))
AS avg_diff_estimated_delivery
FROM `business-case-study-sql.Target_dataset.orders` AS o
INNER JOIN `business-case-study-sql.Target_dataset.customers` AS c
ON o.customer_id = c.customer_id
INNER JOIN `business-case-study-sql.Target_dataset.order_items` AS it
ON o.order_id = it.order_id
WHERE order_status = 'delivered'
GROUP BY state
ORDER BY avg_freight_value DESC
LIMIT 5;
```



Sheets plot:



Insights:

We observe that:
Maximum avg freight value = RR & PB state (~42 units)
Maximum avg time to delivery = RR state (~28 days)
Maximum difference in estimated vs actual delivery = AC state (~21 days)

```
--ORDER BY avg_freight_value ASC
SELECT
 c.customer_state AS state,
 AVG(it.freight_value) AS avg_freight_value,
 AVG(date_DIFF(date(o.order_delivered_customer_date), date(o.order_purchase_timestamp), DAY)) AS
avg_time_to_delivery,
 AVG(date_DIFF(date(o.order_estimated_delivery_date), date(o.order_delivered_customer_date), DAY))
AS avg_diff_estimated_delivery
FROM `business-case-study-sql.Target_dataset.orders` AS o
INNER JOIN `business-case-study-sql.Target_dataset.customers` AS c
ON o.customer_id = c.customer_id
INNER JOIN `business-case-study-sql.Target_dataset.order_items` AS it
ON o.order_id = it.order_id
WHERE order_status = 'delivered'
GROUP BY state
ORDER BY avg_freight_value ASC
LIMIT 5;
```

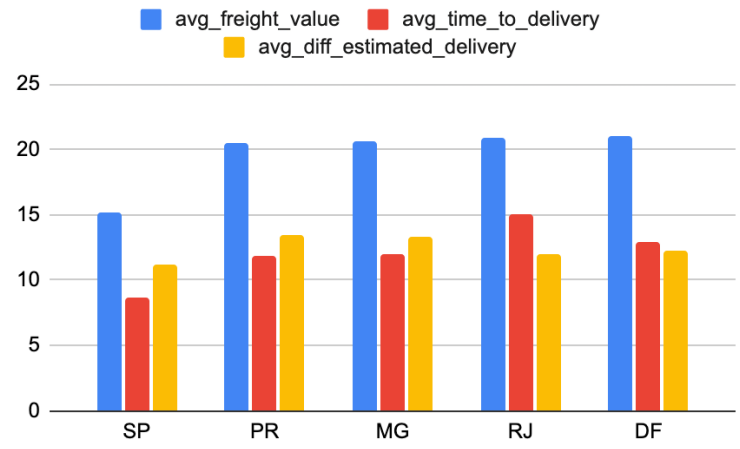
Query results

SAVE RESULTS | EXPLORE DATA

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

Row	state	avg_freight_valu	avg_time_to_del	avg_diff_estimat
1	SP	15.1151823...	8.66232423...	11.2064555...
2	PR	20.4718162...	11.8930784...	13.4861037...
3	MG	20.6263425...	11.9193248...	13.3446113...
4	RJ	20.9114360...	15.0741709...	12.0098988...
5	DF	21.0721613...	12.8938428...	12.2004246...

Sheets Plot:



Insights:

We observe that:
Minimum avg freight value = SP state (~15 units)
Minimum avg time to delivery = SP state (~8 days)
Minimum difference in estimated vs actual delivery = SP state (~12 days)

5.4) Top 5 states with highest/lowest average time to delivery

```
--highest delivery time

SELECT
 g.geolocation_state AS state,
 SUM(TIMESTAMP_DIFF(TIMESTAMP(order_estimated_delivery_date), TIMESTAMP(order_purchase_timestamp),
DAY)) / COUNT(ORDER_ID) AS avg_del_time
FROM `business-case-study-sql.Target_dataset.orders` o
INNER JOIN `business-case-study-sql.Target_dataset.customers` c ON o.customer_id = c.customer_id
INNER JOIN `business-case-study-sql.Target_dataset.geolocation` g ON c.customer_zip_code_prefix =
g.geolocation_zip_code_prefix
WHERE order_status = 'delivered'
GROUP BY g.geolocation_state
ORDER BY avg_del_time DESC
```

Query results

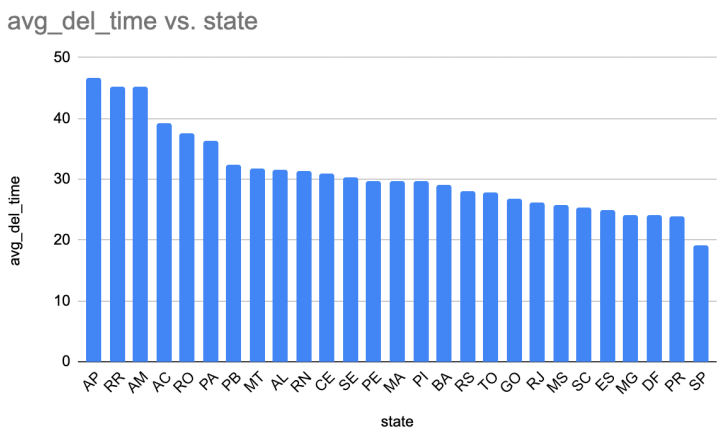
SAVE RESULTS | EXPLORE DATA

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

Row	state	avg_del_time
1	AP	46.5684144...
2	RR	45.2594654...
3	AM	45.1333820...
4	AC	39.2102604...
5	RO	37.6369070...
6	PA	36.4100072...
7	PB	32.4802731...
8	MT	31.8519264...
9	AL	31.4986354...
10	RN	31.4127235...

Results per page: 50 1 – 27 of 27

Sheets Plot:



Insights:

We observe that:
The highest delivery time = AP, RR, AM state (~46 days)

```sql
--lowest delivery time

SELECT
 g.geolocation_state AS state,
 SUM(TIMESTAMP_DIFF(TIMESTAMP(order_estimated_delivery_date), TIMESTAMP(order_purchase_timestamp),
DAY)) / COUNT(ORDER_ID) AS avg_del_time
FROM `business-case-study-sql.Target_dataset.orders` o
INNER JOIN `business-case-study-sql.Target_dataset.customers` c ON o.customer_id = c.customer_id
INNER JOIN `business-case-study-sql.Target_dataset.geolocation` g ON c.customer_zip_code_prefix =
g.geolocation_zip_code_prefix
WHERE order_status = 'delivered'
GROUP BY g.geolocation_state
ORDER BY avg_del_time ASC
```

Query results

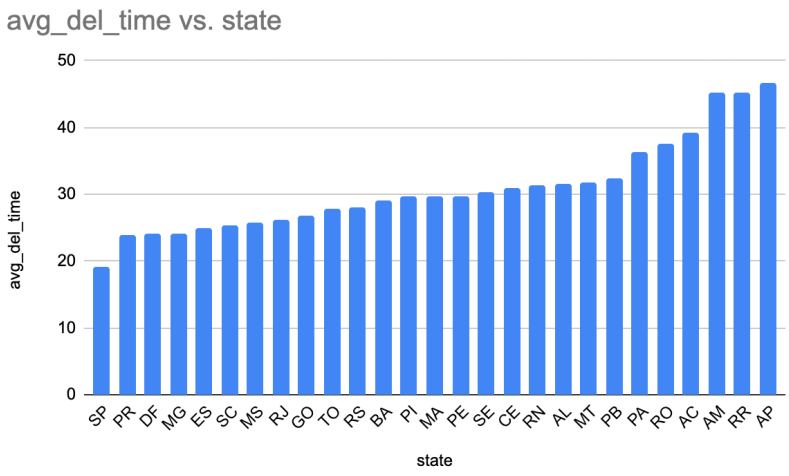
SAVE RESULTS | EXPLORE DATA

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

Row	state	avg_del_time
1	SP	19.0483124...
2	PR	23.9669328...
3	DF	24.0838005...
4	MG	24.1111076...
5	ES	25.0071975...
6	SC	25.3494907...
7	MS	25.7513439...
8	RJ	26.0842963...
9	GO	26.8784227...
10	TO	27.9042927...

Results per page: 50 1 – 27 of 27

Sheets Plot:



Insights:

We observe that regarding the delivery time that SP state is delivering the fastest (~19 days)
This may be the reason for people ordering the most this state (as already analysed in 4.2).
The slowest delivery time is for AP state (~46 days).
The customers will lose interest to order because of this huge delay in delivery.
The delivery partners need to find some ways to decrease the delivery time.

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

```sql
--Top 5 states where delivery is really fast

SELECT
 g.geolocation_state AS state,
 AVG(TIMESTAMP_DIFF(TIMESTAMP(order_estimated_delivery_date),
TIMESTAMP(order_delivered_customer_date), DAY)) AS diff_estimated_delivery
FROM `business-case-study-sql.Target_dataset.orders` o
INNER JOIN `business-case-study-sql.Target_dataset.customers` c ON o.customer_id = c.customer_id
INNER JOIN `business-case-study-sql.Target_dataset.geolocation` g ON c.customer_zip_code_prefix =
g.geolocation_zip_code_prefix
WHERE order_status = 'delivered'
GROUP BY g.geolocation_state
ORDER BY diff_estimated_delivery ASC
LIMIT 5;
```

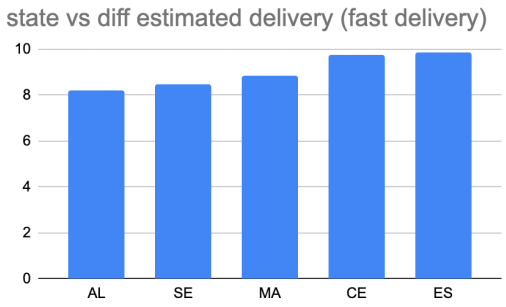
Query results

SAVE RESULTS | EXPLORE DATA

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

Row	state	diff_estimated_c
1	AL	8.20063385...
2	SE	8.48572389...
3	MA	8.84281466...
4	CE	9.72013087...
5	ES	9.85501162...

Sheets Plot:



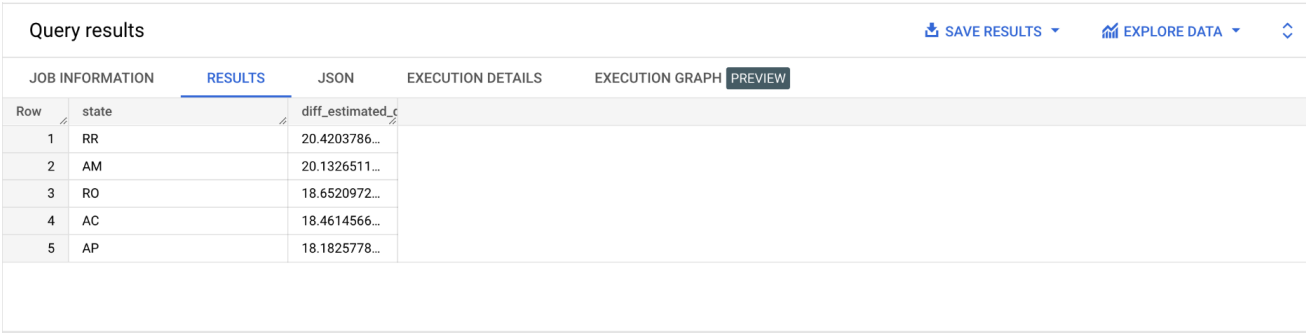
Insights:

We observe that

The fastest delivery is for AL state (~8 days).

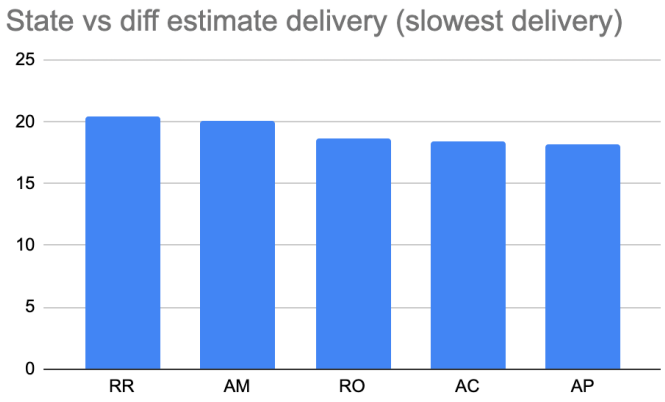
```sql
--Top 5 states where delivery is not so fast compared to estimated date

SELECT
 g.geolocation_state AS state,
 AVG(TIMESTAMP_DIFF(TIMESTAMP(order_estimated_delivery_date),
TIMESTAMP(order_delivered_customer_date), DAY)) AS diff_estimated_delivery
FROM `business-case-study-sql.Target_dataset.orders` o
INNER JOIN `business-case-study-sql.Target_dataset.customers` c ON o.customer_id = c.customer_id
INNER JOIN `business-case-study-sql.Target_dataset.geolocation` g ON c.customer_zip_code_prefix =
g.geolocation_zip_code_prefix
WHERE order_status = 'delivered'
GROUP BY g.geolocation_state
ORDER BY diff_estimated_delivery DESC
LIMIT 5;
```



Row	state	diff_estimated_d
1	RR	20.4203786...
2	AM	20.1326511...
3	RO	18.6520972...
4	AC	18.4614566...
5	AP	18.1825778...

Sheets Plot:



Insights:

We observe that
The slowest delivery is for RR state (~21days).

Q6) Payment type analysis:

1. Month over Month **count of orders** for different payment types
2. Count of orders based on the no. of payment installments

6.1)Month over Month **count of orders** for different payment types

```
WITH
cte_table AS (
SELECT
EXTRACT(month FROM timestamp(o.order_purchase_timestamp)) AS month,
EXTRACT(year FROM timestamp(o.order_purchase_timestamp)) AS year,
(sum(price) / COUNT( distinct o.order_id)) AS price_per_order,
(sum(freight_value) / COUNT(distinct o.order_id)) AS freight_per_order
FROM `business-case-study-sql.Target_dataset.orders` o
INNER JOIN `business-case-study-sql.Target_dataset.order_items` i
ON o.order_id = i.order_id
GROUP BY year, month
)
SELECT (price_per_order), (freight_per_order), month, year
FROM cte_table
order by payment_type, year asc, month asc ;
```

Query results

SAVE RESULTS | EXPLORE DATA

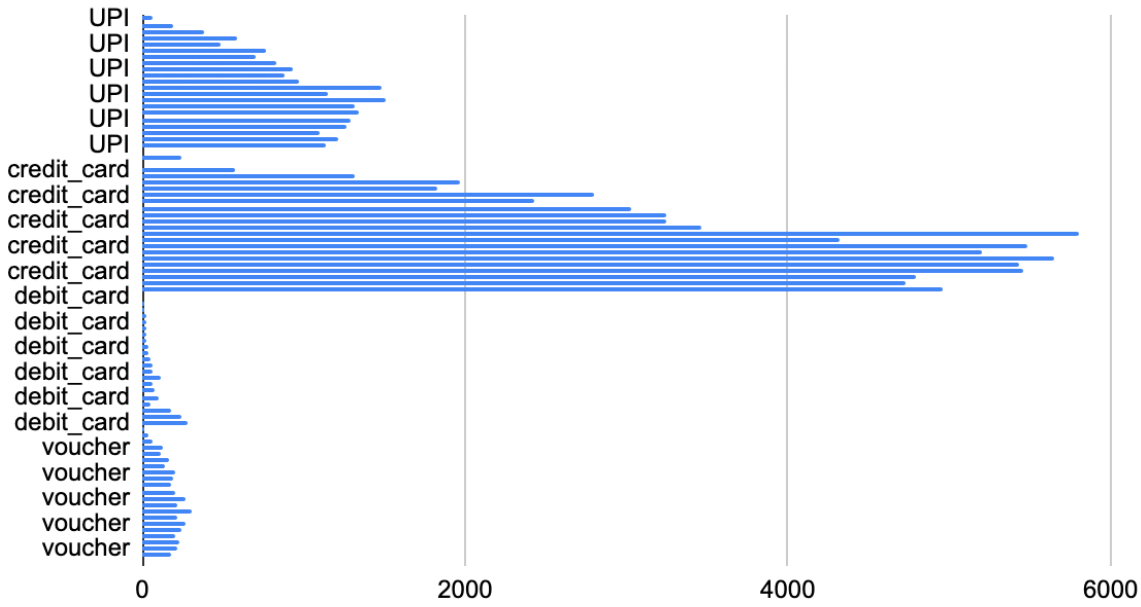
JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

Row	payment_type	count_of_orders	month	year
1	UPI	58	10	2016
2	UPI	193	1	2017
3	UPI	385	2	2017
4	UPI	584	3	2017
5	UPI	491	4	2017
6	UPI	761	5	2017
7	UPI	703	6	2017
8	UPI	833	7	2017
9	UPI	928	8	2017
10	UPI	888	9	2017

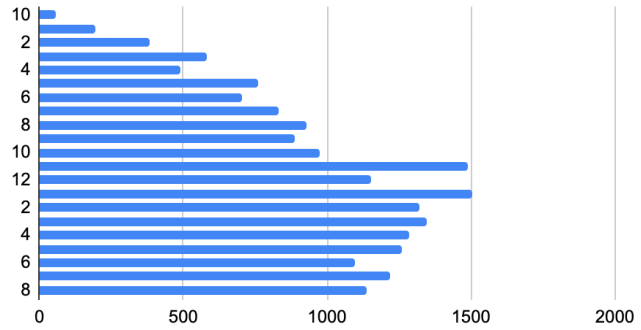
Results per page: 50 1 – 50 of 87

Sheets Plot:

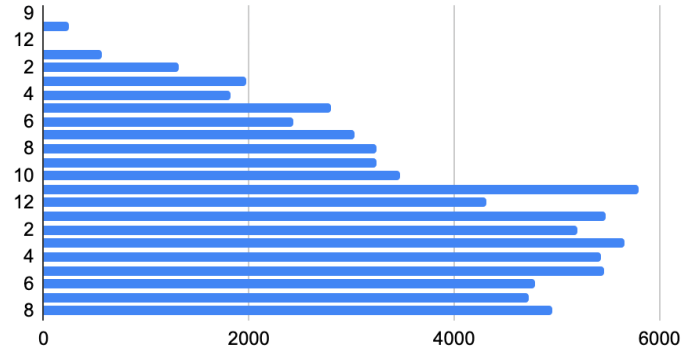
payment_type vs count_of_orders

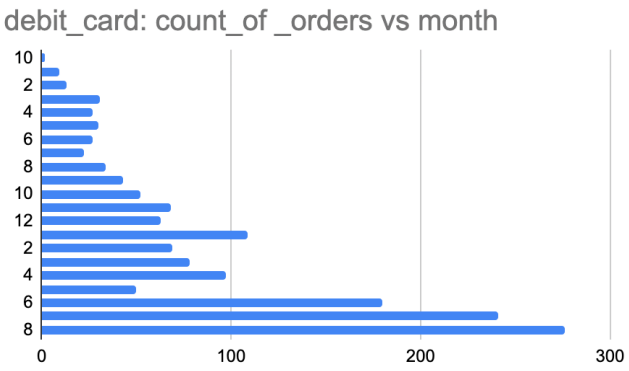


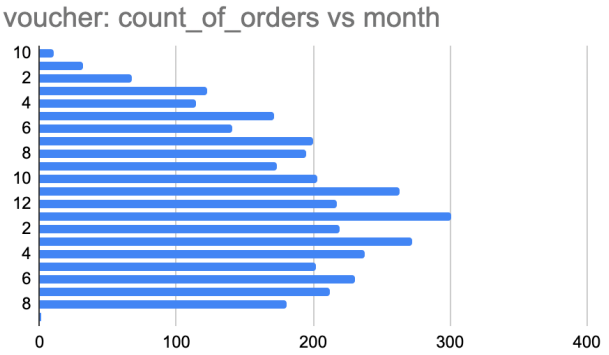
UPI: count_of_orders vs month



credit_card: count_of_orders vs month







Insights:

We observe that most of the payments are done using Credit Card.
The second most popular method of payment is UPI.
Then vouchers are used for payments.
ANd Debit cards are used the least for payments.
Credit card companies can add some offers/coupons for purchasing anything.
Customer should maintain a good credit score to avail best offers and loans for their needs.
UPI merchants should also give some rewards or cash backs to promote UPI payments.
As UPI payments are Down Payments.
The volume of payments is increasing over the months from 2016 to 2018.

6.2)Count of orders based on the no. of payment installments

```
SELECT
 payment_installments,
 COUNT(DISTINCT order_id) AS count_of_orders
FROM
 `business-case-study-sql.Target_dataset.payments`
GROUP BY
 payment_installments
```

Query results

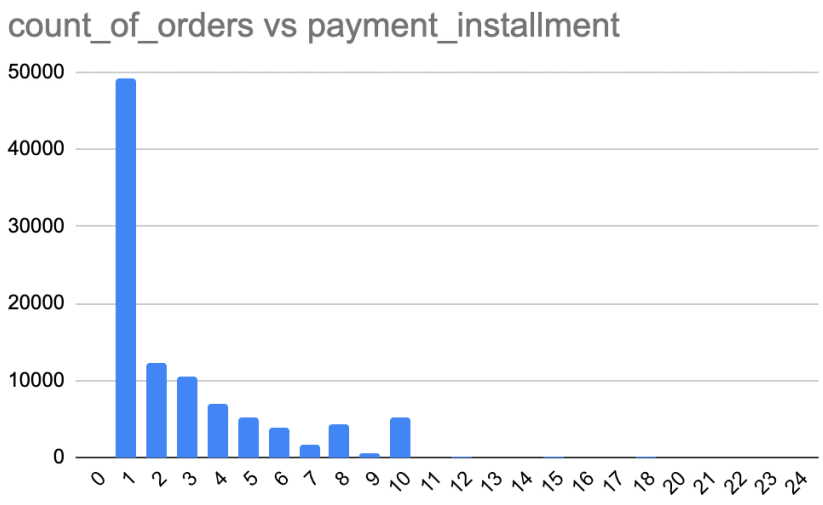
SAVE RESULTS | EXPLORE DATA

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

Row	payment_installments	count_of_orders
1	0	2
2	1	49060
3	2	12389
4	3	10443
5	4	7088
6	5	5234
7	6	3916
8	7	1623
9	8	4253
10	9	644

Results per page: 50 | 1 – 24 of 24

Sheets Plot:



Insights:

We observe that most of the payments are done at once.
Then 2 to 10 instalments are used for payments.
And 11-24 instalments are very less used for payments.