TARGET DATA ANALYSIS USING SQL

For Brazil

SCALER

14-11-2022

What ‘good’ looks 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

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

ilter 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

Filter 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

Filter 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

Filter Enter property name or value

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

Filter seller

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

2. Time period for which the data is given

```
1  SELECT order_purchase_timestamp FROM `target-dataset-368418.target.orders` order by order_purchase_timestamp ASC LIMIT 1
```

Query results

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

Row	order_purchase_timestamp
1	2016-09-04 21:15:19 UTC

```
1  SELECT order_purchase_timestamp FROM `target-dataset-368418.target.orders` order by order_purchase_timestamp DESC LIMIT 1
```

Query results

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

ow	order_purchase_timestamp
1	2018-10-17 17:30:18 UTC

3. Cities and States covered in dataset

```
1  SELECT count(distinct(customer_city)) city_count  FROM `target-dataset-368418.target.customers`
```

Query results

JOB INFORMATION | RESU

Row	city_count
1	4119

```
1  SELECT distinct(customer_city) city_count  FROM `target-dataset-368418.target.customers`
```

Query results

JOB INFORMATION | RESULTS

Row	city_count
1	acu
2	ico
3	ipe
4	ipu
5	ita
6	itu
7	jau
8	luz
9	poa
10	uba

```
1  SELECT distinct(customer_state) city_count  FROM `target-dataset-368418.target.customers`
```

Query results

JOB INFORMATION | RESULTS

Row	city_count
1	RN
2	CE
3	RS
4	SC
5	SP
6	MG
7	BA
8	RJ
9	GO
10	MA

```
1  SELECT COUNT(distinct(customer_state)) as number_of_States from `target.customers`
```

Query results

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

ow	number_of_...
1	27

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?

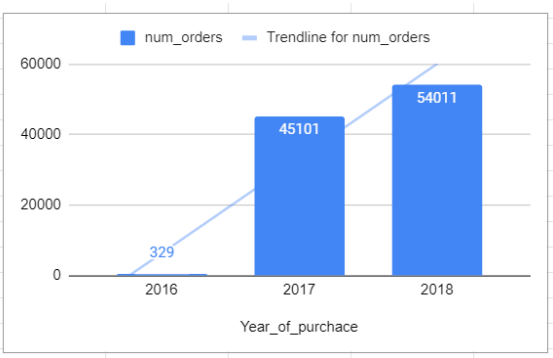
YOY growth

```
1  select
2  EXTRACT (YEAR FROM DATETIME (order_purchase_timestamp)) as Year_of_purchace,
3  count(*) as num_orders
4
5  FROM target-dataset-368418.target.orders
6
7  group by Year_of_purchace
8
9  order by Year_of_purchace asc
10
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GR

Row	Year_of_pur...	num_orders
1	2016	329
2	2017	45101
3	2018	54011



```sql
select

FORMAT_DATE('%B , %Y', datetime(order_purchase_timestamp)) AS month_year_of_purchace,
count(order_id) as num_orders,
EXTRACT(YEAR FROM DATETIME(order_purchase_timestamp)) as YEAR,
EXTRACT(MONTH FROM DATETIME(order_purchase_timestamp)) as Month


FROM target-dataset-368418.target.orders

group by month_year_of_purchace, YEAR,Month


order by YEAR,Month asc
```

ery results

INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH | PREV

month_year_of_purchace	num_orders	YEAR	Month
September , 2016	4	2016	9
October , 2016	324	2016	10
December , 2016	1	2016	12
January , 2017	800	2017	1
February , 2017	1780	2017	2
March , 2017	2682	2017	3
April , 2017	2404	2017	4
May , 2017	3700	2017	5

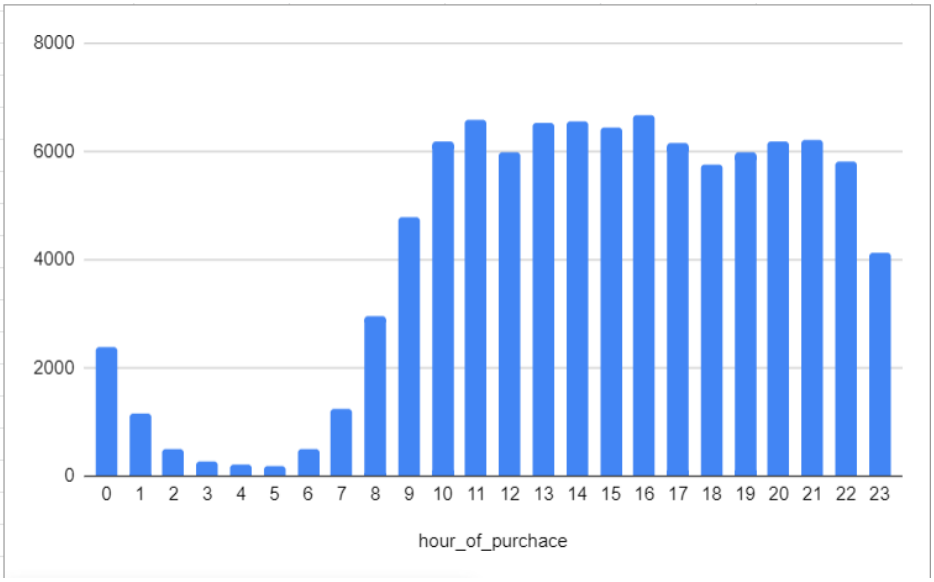
2. What time do Brazilian customers tend to buy (Dawn, Morning, Afternoon or Night)?

```sql
select
EXTRACT (HOUR FROM DATETIME (order_purchase_timestamp)) as hour_of_purchace,
count(*) as num_orders

FROM target-dataset-368418.target.orders

group by hour_of_purchace

order by hour_of_purchace asc

```

Query results

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

w	hour_of_pur...	num_orders
1	0	2394
2	1	1170
3	2	510
4	3	272
5	4	206
6	5	188
7	6	502
8	7	1231



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

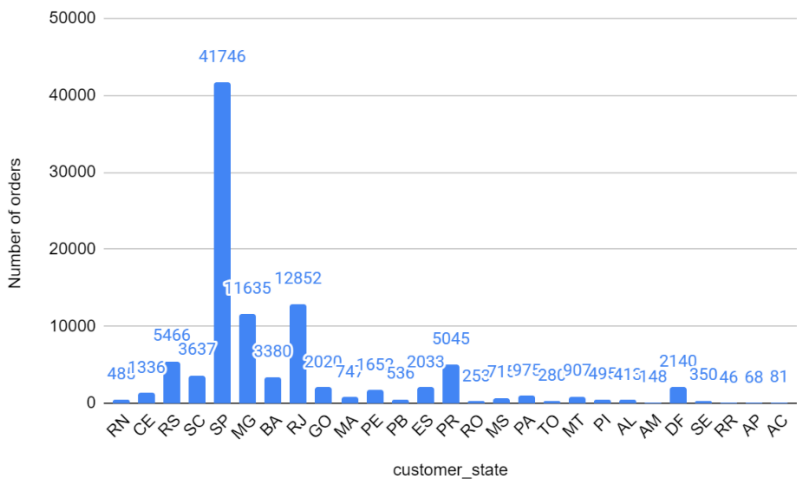
1. Get month on month orders by region, states

```sql
select
customers.customer_state,

FORMAT_DATE('%B , %Y', datetime(order_purchase_timestamp)) AS month_year_of_purchac,

count (orders.order_id) as num_orders,
EXTRACT (MONTH FROM DATETIME (order_purchase_timestamp)) as month_of_purchace,
EXTRACT (YEAR FROM DATETIME (order_purchase_timestamp)) as  year_of_purchace

FROM target-dataset-368418.target.customers


LEFT JOIN target-dataset-368418.target.orders ON customers.customer_id = orders.customer_ic

group by  month_of_purchace, month_year_of_purchac,customers.customer_state,year_of_purchac


order by customers.customer_state, month_of_purchace asc,year_of_purchace

```

Query results

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

w	customer_state	month_year_of_purchac	num_orders	month_of_p...	year_of_pur...
1	AC	January , 2017	2	1	2017
2	AC	January , 2018	6	1	2018
3	AC	February , 2017	3	2	2017
4	AC	February , 2018	3	2	2018
5	AC	March , 2017	2	3	2017
6	AC	March , 2018	2	3	2018
7	AC	April , 2017	5	4	2017
8	AC	April , 2018	4	4	2018
9	AC	May , 2017	8	5	2017
10	AC	May , 2018	2	5	2018



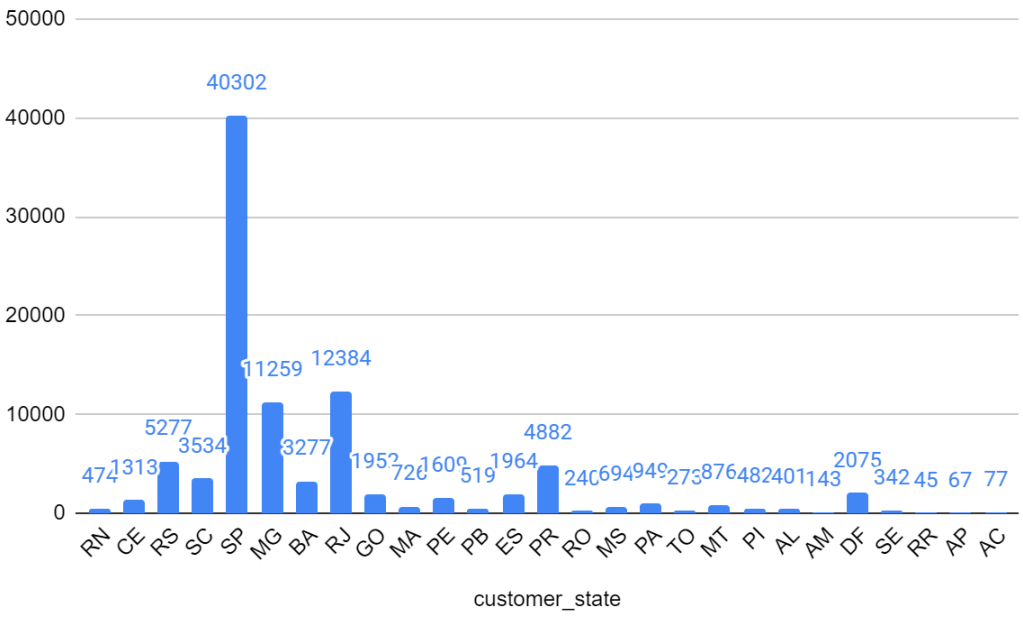
2. How are customers distributed in Brazil

```sql
SELECT
customer_state,
count(distinct(customer_unique_id)) as number_of_customers

from `target-dataset-368418.target.customers`

group by customer_state
order by number_of_customers desc
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

ow	customer_state	number_of_...
1	SP	40302
2	RJ	12384
3	MG	11259
4	RS	5277
5	PR	4882
6	SC	3534
7	BA	3277
8	DF	2075
9	ES	1964
10	GO	1952



`

4. Impact on Economy: Analyze the money movemented 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)

```sql
WITH
  TEMP AS (
  SELECT
    EXTRACT (YEAR
    FROM
      DATETIME(shipping_limit_date)) AS Year_of_purchace,
    EXTRACT (MONTH
    FROM
      DATETIME(shipping_limit_date)) AS Month_of_purchace,
    SUM(price) AS cost_of_sales
  FROM
    target.order_items
  GROUP BY
    Year_of_purchace,
    Month_of_purchace
  HAVING
    Month_of_purchace<9
    AND Year_of_purchace BETWEEN 2016
    AND 2019
  ORDER BY
    Year_of_purchace,
    Month_of_purchace)
SELECT
  ((b.cost_of_sales-a.cost_of_sales)/a.cost_of_sales)*100 AS Percentage_change,
  a.Month_of_purchace,
FROM
  TEMP a
JOIN
  TEMP b
ON
  a.Month_of_purchace = b.Month_of_purchace
WHERE
  a.Year_of_purchace < b.Year_of_purchace
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DE

Row	Percentage_change	Month_of_purchace
1	74.331245496013949	7
2	91.423040309671691	8
3	85.657759188388113	6
4	200.23308295650372	3
5	211.12605124702762	4
6	114.4461270130449	5
7	954.95403791645685	1
8	233.04421327399828	2

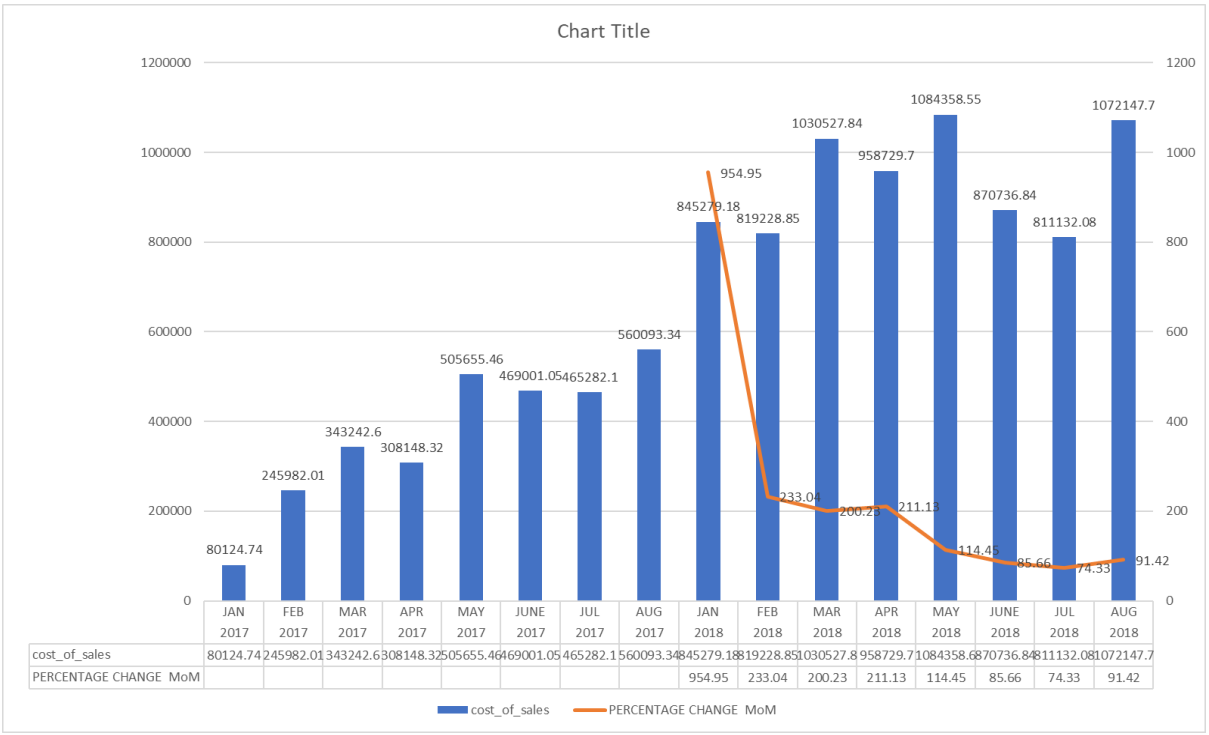


Chart Title
1200000
1000000
800000
600000
400000
200000
0
1200
1000
800
600
400
200
0
80124.74
245982.01
343242.6
308148.32
505655.46
469001.05
465282.1
560093.34
845279.18
819228.85
1030527.84
958729.7
1084358.55
870736.84
811132.08
1072147.7
954.95
233.04
200.23
211.13
114.45
85.66
74.33
91.42
JAN 2017
FEB 2017
MAR 2017
APR 2017
MAY 2017
JUNE 2017
JUL 2017
AUG 2017
JAN 2018
FEB 2018
MAR 2018
APR 2018
MAY 2018
JUNE 2018
JUL 2018
AUG 2018
cost_of_sales
PERCENTAGE CHANGE MoM

2. Mean & Sum of price and freight value by a customer state

```sql
SELECT
  c.customer_state,
  AVG(op.price) mean_price,
  SUM(op.price) sum_price,
  AVG(op.freight_value) mean_freight_charges,
  SUM(op.freight_value) sum_fright_charges
FROM
  target.customers c
JOIN
  target.orders o
ON
  c.customer_id =o.customer_id
JOIN
  `target.order_items` op
ON
  o.order_id= op.order_id
GROUP BY
  c.customer_state
```

Query results

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

ow	customer_state	mean_price	sum_price	mean_freigh...	sum_fright_...
1	MT	148.297184...	156453.529...	28.1662843...	29715.4300...
2	MA	145.204150...	119648.219...	38.2570024...	31523.7700...
3	AL	180.889211...	80314.81	35.8436711...	15914.5899...
4	SP	109.653629...	5202955.05...	15.1472753...	718723.069...
5	MG	120.748574...	1585308.02...	20.6301668...	270853.460...
6	PE	145.508322...	262788.029...	32.9178626...	59449.6599...
7	RJ	125.117818...	1824092.66...	20.9609239...	305589.310...
8	DF	125.770548...	302603.939...	21.0413549...	50625.4999...
9	RS	120.337453...	750304.020...	21.7358043...	135522.740...
10	SE	153.041168...	58920.8500...	36.6531688...	14111.4699...

5. Analysis on sales, freight and delivery time

1. Calculate days between purchasing, delivering and estimated delivery

```
SELECT
  order_id,
  customer_id,
  order_status,
  order_purchase_timestamp,
  order_delivered_carrier_date,
  order_estimated_delivery_date,
  DATE_DIFF( order_delivered_carrier_date,order_purchase_timestamp,DAY) AS actual_delevery_days,
  DATE_DIFF( order_estimated_delivery_date,order_purchase_timestamp,DAY) AS esimated_delvery_date
FROM
  `target.orders`
WHERE
  order_delivered_carrier_date IS NOT NULL
```

Press Alt+F1

Query results

SAVE RESULTS | EXPLO

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

	order_id	customer_id	order_...	order_purchase_ti...	order_delivered_carri...	order_estimated_delivery_...	actual_delev...	esimated_d...
1	3385c99ff53af3e0f111...	5b1939524fc66b1be29160f5d...	shipped	2017-02-04 12:58:...	2017-03-03 12:36:00 ...	2017-03-15 00:00:00 UTC	26	38
2	52cb9b4d5ee3ce7d1e...	a5c8228ef32a5a250903b18c0...	shipped	2018-07-11 20:24:...	2018-07-31 14:10:00 ...	2018-08-01 00:00:00 UTC	19	20
3	9981f1a6ab88e70da56...	c8c5ec1b4ad39f4b5d6dc83f1...	shipped	2018-03-22 20:12:...	2018-04-05 15:42:21 ...	2018-04-27 00:00:00 UTC	13	35
4	d56ec35ab3257ad373...	c19d9a703628a4464d27d3af8...	shipped	2018-01-19 18:32:...	2018-02-03 08:38:38 ...	2018-02-22 00:00:00 UTC	14	33
5	063b573b88fc80e516a...	285195a5b585842e25bd1ef90...	shipped	2016-10-07 19:17:...	2016-10-30 10:23:36 ...	2016-12-01 00:00:00 UTC	22	54
6	a68ce1686d536ca72b...	d7bed5fac093a4136216072ab...	shipped	2016-10-05 01:47:...	2016-11-07 16:37:37 ...	2016-12-01 00:00:00 UTC	33	56
7	45973912e490866800...	912f108a7026f25f99240a5c4c...	shipped	2016-10-07 22:45:...	2016-10-26 13:18:16 ...	2016-12-01 00:00:00 UTC	18	54
8	cda873529ca7ab71f67...	76c74aaff2f3f7355f46d9818a...	shipped	2016-10-05 16:57:...	2016-11-14 11:14:39 ...	2016-12-01 00:00:00 UTC	39	56
9	8e1d33832d930d453a...	5079d74c713c4f881925b5170...	shipped	2018-03-22 12:47:...	2018-04-06 23:52:48 ...	2018-04-19 00:00:00 UTC	15	27
10	38541d08d4eb7d571b...	937682b4176cf634c443714be...	shipped	2017-11-24 21:36:...	2018-01-04 21:07:51 ...	2017-12-20 00:00:00 UTC	40	25

2. Create columns:

1. time_to_delivery = order_purchase_timestamp-order_delivered_customer_date
2. diff_estimated_delivery = order_estimated_delivery_date-order_delivered_customer_date

```
SELECT
  order_id,
  customer_id,
  order_status,
  order_purchase_timestamp,
  order_delivered_carrier_date,
  order_estimated_delivery_date,
  DATE_DIFF( order_delivered_carrier_date,order_purchase_timestamp,DAY) AS time_to_delivery_in_days,
  DATE_DIFF( order_estimated_delivery_date,order_purchase_timestamp,DAY) AS diff_estimated_delivery_in_days
FROM
  `target.orders`
WHERE
  order_delivered_carrier_date IS NOT NULL
```

Press Alt+F1 for Accessibility Opti

ery results

SAVE RESULTS | EXPLORE DATA

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

	order_id	customer_id	order_st...	order_purchase_times...	order_delivered_carri...	order_estimated_delivery_date	time_to_delivery_in_days	diff_estimated_delivery_...
	3385c99ff53af3e0f111...	5b1939524fc66b1be29160...	shipped	2017-02-04 12:58:55 U...	2017-03-03 12:36:00 ...	2017-03-15 00:00:00 UTC	26	38
	52cb9b4d5ee3ce7d1e2...	a5c8228ef32a5a250903b1...	shipped	2018-07-11 20:24:49 U...	2018-07-31 14:10:00 ...	2018-08-01 00:00:00 UTC	19	20
	9981f1a6ab88e70da56...	c8c5ec1b4ad39f4b5d6dc8...	shipped	2018-03-22 20:12:15 U...	2018-04-05 15:42:21 ...	2018-04-27 00:00:00 UTC	13	35
	d56ec35ab3257ad373b...	c19d9a703628a4464d27d3...	shipped	2018-01-19 18:32:44 U...	2018-02-03 08:38:38 ...	2018-02-22 00:00:00 UTC	14	33
	063b573b88fc80e516a...	285195a5b585842e25bd1e...	shipped	2016-10-07 19:17:00 U...	2016-10-30 10:23:36 ...	2016-12-01 00:00:00 UTC	22	54
	a68ce1686d536ca72bd...	d7bed5fac093a413621607...	shipped	2016-10-05 01:47:40 U...	2016-11-07 16:37:37 ...	2016-12-01 00:00:00 UTC	33	56
	45973912e490866800c...	912f108a7026f25f99240a5...	shipped	2016-10-07 22:45:28 U...	2016-10-26 13:18:16 ...	2016-12-01 00:00:00 UTC	18	54
	cda873529ca7ab71f67...	76c74aaff2f3f7355f46d98...	shipped	2016-10-05 16:57:30 U...	2016-11-14 11:14:39 ...	2016-12-01 00:00:00 UTC	39	56
	8e1d33832d930d453a3...	5079d74c713c4f881925b5...	shipped	2018-03-22 12:47:22 U...	2018-04-06 23:52:48 ...	2018-04-19 00:00:00 UTC	15	27

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

```sql
SELECT
  customer_state,
  AVG(DATE_DIFF( o.order_delivered_carrier_date,o.order_purchase_timestamp,DAY)) AS mean_time_to_delivery,
  AVG(DATE_DIFF( o.order_estimated_delivery_date,order_purchase_timestamp,DAY)) AS mean_diff_estimated_delivery,
  AVG(ot.freight_value) AS mean_freight_value
FROM
  target.customers c
LEFT JOIN
  target.orders o
ON
  c.customer_id = o.customer_id
JOIN
  target.order_items ot
ON
  o.order_id = ot.order_id
GROUP BY
  customer_state
order by mean_time_to_delivery asc ,mean_diff_estimated_delivery desc,mean_freight_value
```

Query results

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

ow	customer_state	mean_time_to_delivery	mean_diff_estimated_...	mean_freigh...
1	AM	2.29090909090909	45.206060606060618	33.2053939...
2	RO	2.3369963369963385	38.651079136690626	41.0697122...
3	GO	2.6150519031141854	26.623231890270041	22.7668152...
4	MT	2.720797720797727	31.521327014218059	28.1662843...
5	SP	2.7224963157557447	18.898290796434139	15.1472753...
6	MS	2.7248157248157283	25.700854700854684	23.3748840...
7	PI	2.7654784240150079	29.922509225092249	39.1479704...
8	MG	2.7894291429450639	24.308401249143134	20.6301668...
9	PE	2.7946278679350898	30.8106312292359	32.9178626...
10	RS	2.7996112101085453	28.30906174819571	21.7358043...

4. Sort the data to get the following:

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

```
(SELECT
  customer_state ,
  AVG(ot.freight_value) AS lowest_avg_freight_value
FROM
  target.customers c
LEFT JOIN
  target.orders o
ON
  c.customer_id = o.customer_id
JOIN
  target.order_items ot
ON
  o.order_id = ot.order_id
GROUP BY
  customer_state
ORDER BY
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION

low	customer_state	lowest_avg_...
1	SP	15.1472753...
2	PR	20.5316515...
3	MG	20.6301668...
4	RJ	20.9609239...
5	DF	21.0413549...

```sql
SELECT
  customer_state ,
  AVG(ot.freight_value) AS top_avg_freight_value
FROM
  target.customers c
LEFT JOIN
  target.orders o
ON
  c.customer_id = o.customer_id
JOIN
  target.order_items ot
ON
  o.order_id = ot.order_id
GROUP BY
  customer_state
ORDER BY
  top_avg_freight_value DESC
  LIMIT 5
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTIO

Row	customer_state	top_avg_frei...
1	RR	42.9844230...
2	PB	42.7238039...
3	RO	41.0697122...
4	AC	40.0733695...
5	PI	39.1479704...

2. Top 5 states with highest/lowest average time to delivery

```sql
SELECT
  customer_state,
  AVG(DATE_DIFF( o.order_delivered_carrier_date,o.order_purchase_timestamp,DAY)) AS lowest_avg_time_of_delevery,
FROM
  target.customers c
LEFT JOIN
  target.orders o
ON
  c.customer_id = o.customer_id
JOIN
  target.order_items ot
ON
  o.order_id = ot.order_id
GROUP BY
  customer_state
ORDER BY
  lowest_avg_time_of_delevery ASC
LIMIT
  5
```

uery results

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

	customer_state	lowest_avg_time_of_del...
1	AM	2.29090909090909
2	RO	2.3369963369963385
3	GO	2.6150519031141854
4	MT	2.720797720797727
5	SP	2.7224963157557447

```sql
SELECT
  customer_state ,
  avg(DATE_DIFF( o.order_delivered_carrier_date,o.order_purchase_timestamp,DAY)) AS higest_avg_time_of_delevery,
FROM
  target.customers c
LEFT JOIN
  target.orders o
ON
  c.customer_id = o.customer_id
JOIN
  target.order_items ot
ON
  o.order_id = ot.order_id
GROUP BY
  customer_state
  order by higest_avg_time_of_delevery desc

  LIMIT 5
```

Query results

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

	customer_state	higest_avg_...
1	RR	4.62745098...
2	MA	3.39682539...
3	SE	3.24999999...
4	RN	3.20075757...
5	AP	3.14814814...

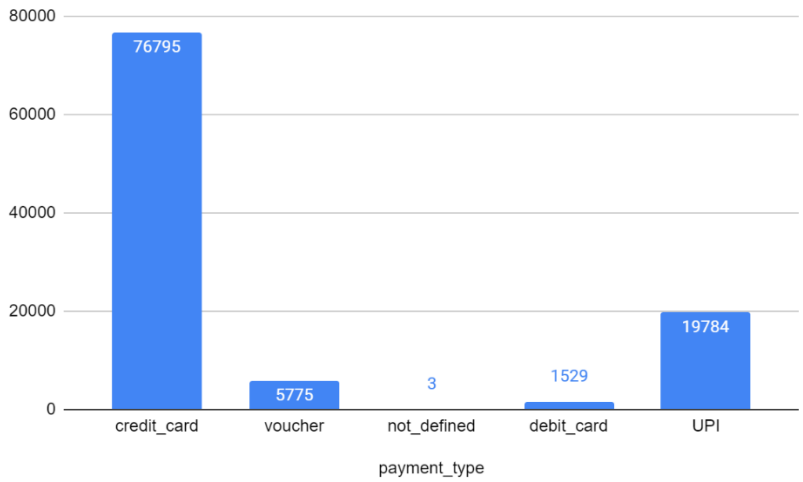
3. Top 5 states where delivery is really fast/ not so fast compared to estimated date

6. Payment type analysis:

1. Month over Month count of orders for different payment types

```sql
SELECT
  FORMAT_DATETIME("%B,%Y", DATETIME(o.order_purchase_timestamp)) AS Month_Year,
  payment_type,
  COUNT(payment_type) AS number_of_transaction,
  EXTRACT(month
  FROM (o.order_purchase_timestamp)) AS month,
  EXTRACT(YEAR
  FROM (o.order_purchase_timestamp)) AS Year,
FROM
  target.orders o
JOIN
  TARGET-dataset-368418.target.payments p
ON
  o.order_id= p.order_id
GROUP BY
  month,
  Year,
  Month_Year,
  payment_type
ORDER BY
  Year,
  month ASC
```

Row	Month_Year	payment_type	number_of_transaction
1	September,2016	credit_card	3
2	October,2016	credit_card	254
3	October,2016	UPI	63
4	October,2016	voucher	23
5	October,2016	debit_card	2
6	December,2016	credit_card	1
7	January,2017	credit_card	583
8	January,2017	UPI	197
9	January,2017	voucher	61
10	January,2017	debit_card	9



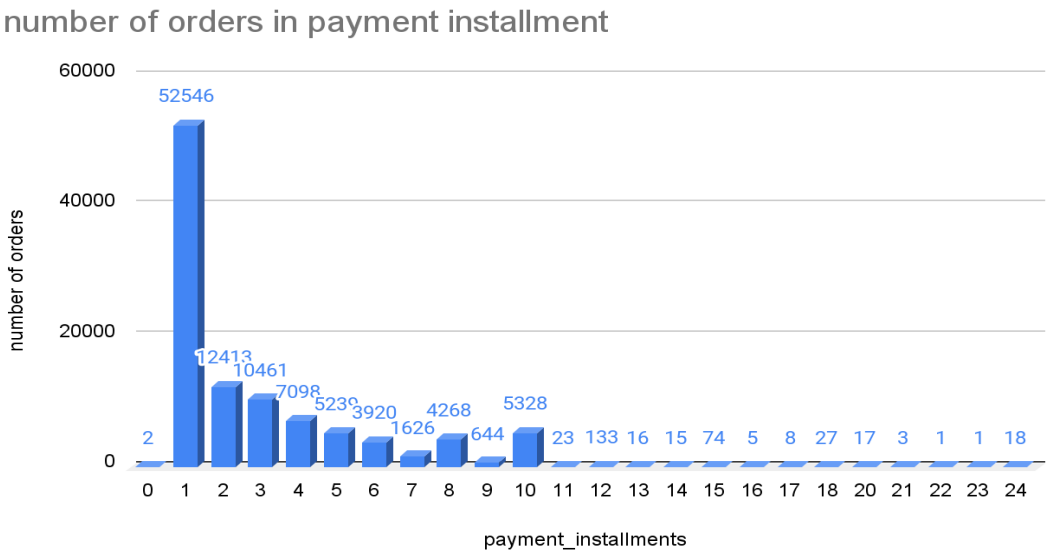
2. Distribution of payment installments and count of orders

```sql
SELECT
  payment_installments,
  COUNT(p.order_id) AS Number_of_orders
FROM
  target.orders o
JOIN
  TARGET-dataset-368418.target.payments p
ON
  o.order_id= p.order_id
GROUP BY
  payment_installments
ORDER BY
  payment_installments ASC
```

Query results

JOB INFORMATION | RESULTS | JSON | E

w	payment_installments	Number_of_orders
1	0	2
2	1	52546
3	2	12413
4	3	10461
5	4	7098
6	5	5239
7	6	3920
8	7	1626
9	8	4268
10	9	644



Actionable Insights:

1. Target data set is based in Brazil

2. time period of data is from 04-09-2016 to 17-10-2018

3.there are total 4119 cities and 27 states of Brazil

4.There is a year on year growth in number of customers , we can see a rise from 2016 to 2018,

As we can observe growth from 329 customer during 2016 to 54011 customers in 2018

Also, there is month on month growth of 954 % in Jan 2018 and a minimum of 91 % growth every month in 2018

5. we cannot observer any seasonal spike apart from November 2017, but that ais an independent data and cannot be compared to next year November

6.brazilian customer tends buy between 9 am to 11 pm in which between 11 am to 6PM is peak hour.

7.maximum customers are from SP, RJ, and MG

8. maximum orders are generated from SP

9. we have observed that 92 % of order reached before the estimated time of delivery

10. mean fastest time to delivery is in 2.29 days in AM state

with mean fright price of 33.20

11. lowest fright price are in SP of 15.14 highest in RR at 42.98 followed by pb with 42.72

12. Top mode of payment is credit card with total of 76795 transaction.

13. customer prefer to pay in 1 instalment using credit. Total of 52546 payments are done for 1 instalment using credit card.

Recommendations:

1. Re-Calculation estimated time of delivery – As site estimates 30 to 40 days to deliver but products are getting delivered much earlier like 2 to 5 days. As so much gap may increase order cancelation.

2. **Re-calculate fright price-** states like AM with shortest time for delivery but average fright price is high at 33.

3. maximum customers are concentrated in few sates like SP, MG and RJ . So some Survey need to be conducted to understand need of the customers of other states.

4. Sales growth should be maintained.