ANALYZING ORDERS DATA FOR MARKET EXPANSION IN THE BRAZIL REGION

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.

Information Schema:

https://cloud.google.com/bigquery/docs/information-schema-intro

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

a) Data type of columns in a table

SQL query:

```
select column_name,data_type
from targetsql.INFORMATION_SCHEMA.COLUMNS
where table_name="customers";


select column_name, data_type
from targetsql.INFORMATION_SCHEMA.COLUMNS
where table_name="orders";
```

Screen shot of output (only first 10 rows)

column_name	data_type
customer_id	STRING
customer_unique_id	STRING
customer_zip_code_prefix	INT64
customer_city	STRING
customer_state	STRING

column_name	data_type
order_id	STRING
customer_id	STRING
order_status	STRING
order_purchase_timestamp	TIMESTAMP
order_approved_at	TIMESTAMP
order_delivered_carrier_date	TIMESTAMP
order_delivered_customer_date	TIMESTAMP
order_estimated_delivery_date	TIMESTAMP

b) Time period for which the data is given

SQL query:

```
select min(order_purchase_timestamp) as min_date,
max(order_purchase_timestamp) as max_date
from `targetsql.orders`
```

Screen shot of output (only first 10 rows)

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

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

Insights:

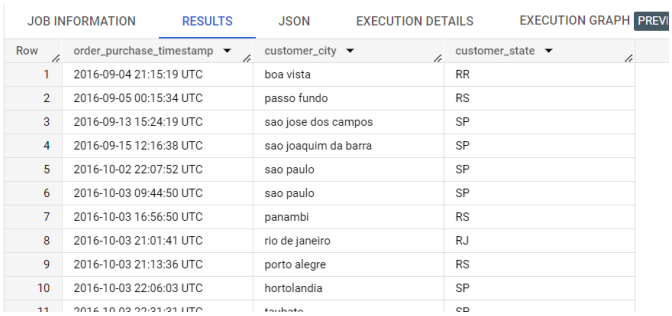
- The period for which the data is being stored will be considered from the first date where orders has been made till the date last order has been received.
- It has been seen that the data set is for the period of 2016–09–04 to 2018–10-17.

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

SQL Query:

```
select o.order_purchase_timestamp, c.customer_city, c.customer_state
from `targetsql.customers` c
left join `targetsql.orders` o
on c.customer_id = o.customer_id
order by o.order_purchase_timestamp
```

Screen shot of output (only first 10 rows)



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

Row	order_purchase_timestamp	customer_city	customer_state
1	2016-09-04 21:15:19 UTC	boa vista	RR
2	2016-09-05 00:15:34 UTC	passo fundo	RS
3	2016-09-13 15:24:19 UTC	sao jose dos campos	SP
4	2016-09-15 12:16:38 UTC	sao joaquim da barra	SP
5	2016-10-02 22:07:52 UTC	sao paulo	SP
6	2016-10-03 09:44:50 UTC	sao paulo	SP
7	2016-10-03 16:56:50 UTC	panambi	RS
8	2016-10-03 21:01:41 UTC	rio de janeiro	RJ
9	2016-10-03 21:13:36 UTC	porto alegre	RS
10	2016-10-03 22:06:03 UTC	hortolandia	SP

Insights:

- For the very first month of 2016-09 orders were scarce and was having only single order on particular day. In the month 2016-10 the orders have rose exponentially .
- The data suggest that SP state has major share of orders placed in the time period.

2) In-depth Exploration:
   a) 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 query:

```
select order_year, order_month, count(customer_id) as order_cnt
from (
select extract(year FROM order_purchase_timestamp) as order_year,
extract(month from order_purchase_timestamp) as order_month, customer_id
from `targetsql.orders`
group by extract(year FROM order_purchase_timestamp), extract(month FROM
order_purchase_timestamp), customer_id
)
group by order_year, order_month
order by order_year, order_month
```

Screen shot of output (only first 10 rows)

Row	order_year	order_month	order_cnt
1	2016	9	4
2	2016	10	324
3	2016	12	1
4	2017	1	800
5	2017	2	1780
6	2017	3	2682
7	2017	4	2404
8	2017	5	3700
9	2017	6	3245
10	2017	7	4026
11	2017	8	4331
12	2017	9	4285
13	2017	10	4631
14	2017	11	7544
15	2017	12	5673
16	2018	1	7269

Insights:

- The data shows that the year 2017 and 2018 have seen a sharp rise in order count from the initial year of 2016. So, we can confirm that there is a growing trend on E-commerce in Brazil.
- We can describe this scenario as the E-commerce has started growing in early 2017, which then has grown to the highest orders in the month of November 2017.
- But the trend in 2018 has been seen a bit mixed. 2018, has experienced highest number of orders in January and March 2018.
- From this data, it has been seen that the months of Nov to March has the highest number of orders. And other months has average of 4400 orders.

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

SQL query:

```
SELECT
Case
 when order_time between "00:00:01" and "05:59:00"
 then "Dawn"
 when order_time between "06:00:00" and "11:59:00"
 then "Morning"
  when order_time between "12:00:00" and "17:59:00"
 then "Afternoon"
 else "Evening"
 END as day_time,
 count(customer_id) as order_cnt
FROM (
select extract(year FROM order_purchase_timestamp) as order_year,
extract(month from order_purchase_timestamp) as order_month,
extract(date from order_purchase_timestamp) as order_date, extract(time from
order_purchase_timestamp) as order_time
 , customer_id
from `targetsql.orders`
group by extract(year FROM order_purchase_timestamp), extract(month FROM
order_purchase_timestamp),extract(time from order_purchase_timestamp),
extract(date from order_purchase_timestamp), customer_id
)
group by day_time
order by day_time;
```

Screen shot of output (only first 10 rows)

day_time	order_cnt
Afternoon	38266
Dawn	4738
Evening	34316
Morning	22121

Insights:

- The data from the above query has shown that the Brazilian Customers tend to buy is in the afternoon and evening.
- Brazilian customers have placed 38000+ orders in the afternoons and 34000+ orders has been placed in the evening.
- The buying happens lowest in the dawn.

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

SQL query:

```
select order_year, order_month, count(customer_id) as order_cnt

from (
select extract(year FROM order_purchase_timestamp) as order_year,
format_date('%B', order_purchase_timestamp) as order_month, customer_id
from `targetsql.orders`
group by extract(year FROM order_purchase_timestamp), format_date('%B',
order_purchase_timestamp), customer_id
)
group by order_year, order_month
order by order_year, order_month;
```

Screen shot of output (only first 10 rows):

order_year	order_month	order_cnt
2016	December	1
2016	October	324
2016	September	4
2017	April	2404
2017	August	4331
2017	December	5673
2017	February	1780
2017	January	800
2017	July	4026
2017	June	3245
2017	March	2682

Insights:

- Data received from the query suggest that the count of orders has grown month on month from January 2017 and peaked at January 2018.
- The initial months i.e.., Sep to Dec 2016 has the lowest order counts.

   b) Distribution of customers across the states in Brazil

SQL query:

```
select c.customer_state, count(c.customer_id) as customer_cnt from
`targetsql.customers` c left join `targetsql.orders` o on c.customer_id =
o.customer_id

group by c.customer_state
order by count(c.customer_id) desc
```

Screen shot of output (only first 10 rows):

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

Insights:

- In Brazil, 27 states has E-commerce customers. The count of customers are not same in every state.
- SP state has the highest number of customers. The lowest number of customers are in RR and AP.

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

    SQL query:

```
select x.order_year, x.order_month, payment_value, cost_growth,
concat(round((cost_growth/payment_value*100),1),"%") as
percnt_growth
from (
```

```sql
select o.order_id, p.payment_value, (round(p.payment_value - LAG
(p.payment_value) OVER (ORDER BY (format_date('%B',
o.order_purchase_timestamp)) ASC),2)) AS cost_growth, extract(year
FROM o.order_purchase_timestamp) as order_year, format_date('%B',
o.order_purchase_timestamp) as order_month  from
`targetsql.orders` o join `targetsql.payments` p on o.order_id =
p.order_id
where p.payment_value != 0) as x
where x.order_year between 2017 and 2018
group by x.order_year, x.order_month, cost_growth, payment_value
order by x.order_year, x.order_month;
```

Screen shot of output (only first 10 rows):

order_year	order_month	payment_value	cost_growth	percnt_growth
2017	April	76.2	61.28	80.4%
2017	April	159.06	-47.38	-29.8%
2017	April	44.73	-24.17	-54%
2017	April	81.88	-235.36	-287.4%
2017	April	337.18	115.1	34.1%
2017	April	212.73	128.61	60.5%
2017	April	106.23	5.32	5%
2017	April	100.54	24.49	24.4%
2017	April	150.95	74.93	49.6%
2017	April	58.95	-192.81	-327.1%
2017	April	100.76	-39.27	-39%

Insights:

- The query has shown that the June 2017 has lowest percentage growth (0%).
- Highest percentage growth (99.9%) is seen in the months of August 2017, Feb 2018, March 2018, May 2018, June 2018

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

SQL Query:

```sql
select c.customer_state, round(avg(ot.price),2) as mean_price,
round(sum(ot.price),2) as sum_price, round(avg(ot.freight_value),2) as
mean_freight_value, round(sum(ot.freight_value),2) as
sum_freight_value  from `targetsql.customers` c left join
`targetsql.orders` o on c.customer_id = o.customer_id
```

```
left join `targetsql.order_items` ot on o.order_id = ot.order_id
group by c.customer_state
order by c.customer_state;
```

Screen shot of output (only first 10 rows):

customer_state	mean_price	sum_price	mean_freight_value	sum_freight_value
AC	173.73	15982.95	40.07	3686.75
AL	180.89	80314.81	35.84	15914.59
AM	135.5	22356.84	33.21	5478.89
AP	164.32	13474.3	34.01	2788.5
BA	134.6	511349.99	26.36	100156.68
CE	153.76	227254.71	32.71	48351.59
DF	125.77	302603.94	21.04	50625.5
ES	121.91	275037.31	22.06	49764.6
GO	126.27	294591.95	22.77	53114.98
MA	145.2	119648.22	38.26	31523.77
MG	120.75	1585308.03	20.63	270853.46

Insights:

- The mean price of the orders is 120 and mean freight value for the orders is around 20.
- Sum of price of all order is 13,591,643 and sum of freight value for all order is 22,519,909.
- Lowest sum of price and lowest sum of freight value is in the RR state.
- Highest sum of price and lowest sum of freight value is in the SP state.
- Lowest mean of price and lowest sum of freight value is in the SP state.
- Highest mean of price and lowest sum of freight value is in the PB state.

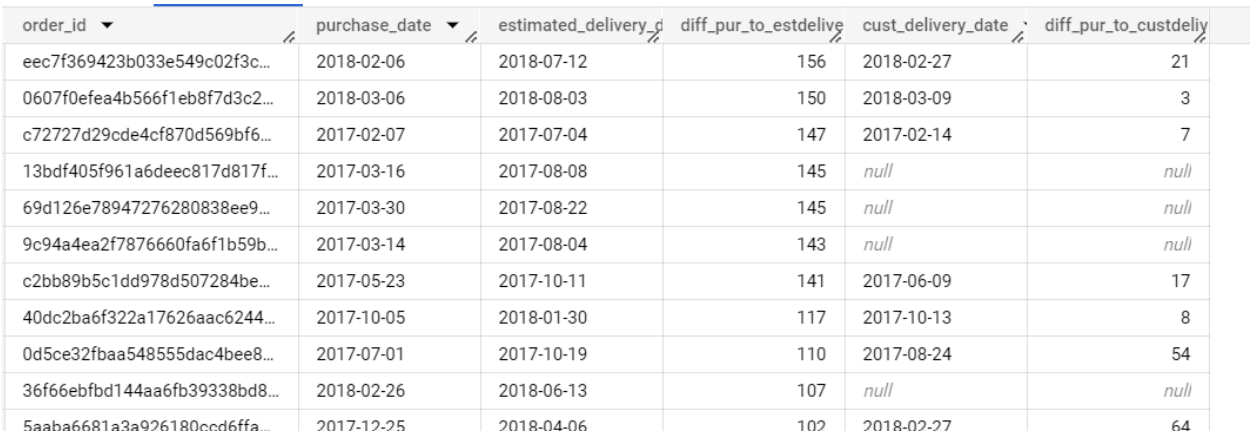
5. Analysis on sales, freight and delivery time

    1. Calculate days between purchasing, delivering and estimated delivery

        SQL Query:

```
select order_id, x.purchase_date, x.estimated_delivery_date,
datetime_diff(x.estimated_delivery_date, x.purchase_date, day) as
diff_pur_to_estdelivery, x.cust_delivery_date,
datetime_diff(x.cust_delivery_date, x.purchase_date, day) as
diff_pur_to_custdelivery
from
(
select
  order_id,
  extract(date FROM order_purchase_timestamp) as purchase_date,
  extract(date FROM order_estimated_delivery_date) as
estimated_delivery_date,
  extract(date FROM order_delivered_customer_date) as cust_delivery_date
from `targetsql.orders`
) as x
order by diff_pur_to_estdelivery desc
```

Screen shot of the output:



order_id	purchase_date	estimated_delivery_d	diff_pur_to_estdelive	cust_delivery_date	diff_pur_to_custdeliv
eec7f369423b033e549c02f3c...	2018-02-06	2018-07-12	156	2018-02-27	21
0607f0efea4b566f1eb8f7d3c2...	2018-03-06	2018-08-03	150	2018-03-09	3
c72727d29cde4cf870d569bf6...	2017-02-07	2017-07-04	147	2017-02-14	7
13bdf405f961a6deec817d817f...	2017-03-16	2017-08-08	145	*null*	*null*
69d126e78947276280838ee9...	2017-03-30	2017-08-22	145	*null*	*null*
9c94a4ea2f7876660fa6f1b59b...	2017-03-14	2017-08-04	143	*null*	*null*
c2bb89b5c1dd978d507284be...	2017-05-23	2017-10-11	141	2017-06-09	17
40dc2ba6f322a17626aac6244...	2017-10-05	2018-01-30	117	2017-10-13	8
0d5ce32fbaa548555dac4bee8...	2017-07-01	2017-10-19	110	2017-08-24	54
36f66ebfbd144aa6fb39338bd8...	2018-02-26	2018-06-13	107	*null*	*null*
5aaba6681a3a926180ccd6ffa...	2017-12-25	2018-04-06	102	2018-02-27	64

Insights:

- The maximum days for an estimated delivery of a order is 156 days and minimum days for an estimated delivery of a order is 2 days.
- The maximum days for customer delivery of a order is 210 days and minimum days for an customer delivery of a order is 2 days.

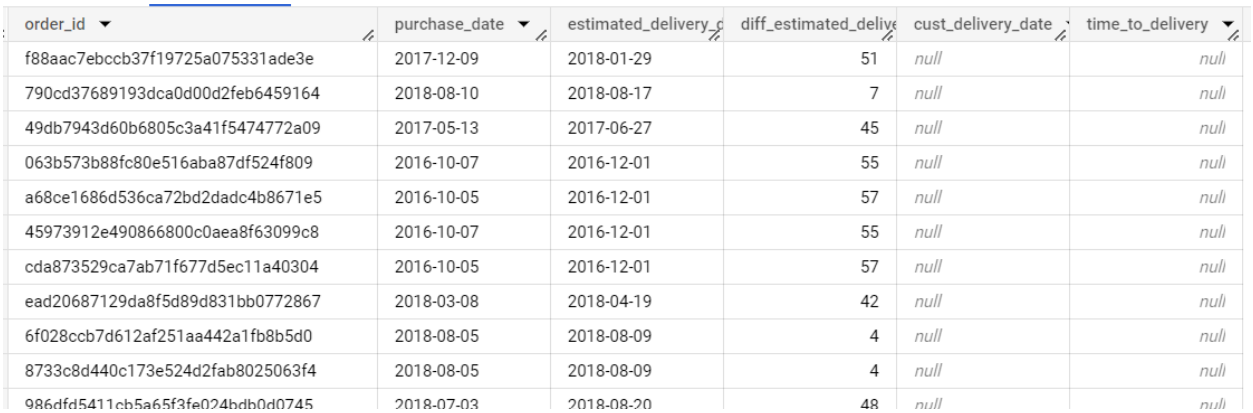
2. Find time_to_delivery & diff_estimated_delivery. Formula for the same given below:
    - time_to_delivery = order_delivered_customer_date- order_purchase_timestamp
    - diff_estimated_delivery = order_estimated_delivery_date- order_delivered_customer_date

SQL Query:

```
select order_id, x.purchase_date, x.estimated_delivery_date,
datetime_diff(x.estimated_delivery_date, x.purchase_date, day) as
diff_estimated_delivery, x.cust_delivery_date,
datetime_diff(x.cust_delivery_date, x.purchase_date, day) as
time_to_delivery

from
(
select
  order_id,
  extract(date FROM order_purchase_timestamp) as purchase_date,
  extract(date FROM order_estimated_delivery_date) as
estimated_delivery_date,
  extract(date FROM order_delivered_customer_date) as cust_delivery_date
from `targetsql.orders`
) as x
```

Screen shot of the output:



order_id	purchase_date	estimated_delivery_d	diff_estimated_delive	cust_delivery_date	time_to_delivery
f88aac7ebccb37f19725a075331ade3e	2017-12-09	2018-01-29	51	null	null
790cd37689193dca0d00d2feb6459164	2018-08-10	2018-08-17	7	null	null
49db7943d60b6805c3a41f5474772a09	2017-05-13	2017-06-27	45	null	null
063b573b88fc80e516aba87df524f809	2016-10-07	2016-12-01	55	null	null
a68ce1686d536ca72bd2dadc4b8671e5	2016-10-05	2016-12-01	57	null	null
45973912e490866800c0aea8f63099c8	2016-10-07	2016-12-01	55	null	null
cda873529ca7ab71f677d5ec11a40304	2016-10-05	2016-12-01	57	null	null
ead20687129da8f5d89d831bb0772867	2018-03-08	2018-04-19	42	null	null
6f028ccb7d612af251aa442a1fb8b5d0	2018-08-05	2018-08-09	4	null	null
8733c8d440c173e524d2fab8025063f4	2018-08-05	2018-08-09	4	null	null
986dfd5411cb5a65f3fe024bdb0d0745	2018-07-03	2018-08-20	48	null	null

Insights:

- The time_to_delivery is maximum 210 days.
- The diff_estimated_delivery is maximum 156 days.

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

SQL Query:

```
select x.customer_state, round(avg(x.freight_value) over (partition by
x.customer_state),2) as mean_freight_value,
datetime_diff(x.estimated_delivery_date, x.purchase_date, day) as
```

```
diff_estimated_delivery, datetime_diff(x.cust_delivery_date,
x.purchase_date, day) as time_to_delivery

from
(
select
  ot.freight_value, c.customer_state,
  extract(date FROM o.order_purchase_timestamp) as purchase_date,
  extract(date FROM o.order_estimated_delivery_date) as
estimated_delivery_date,
  extract(date FROM o.order_delivered_customer_date) as
cust_delivery_date
from `targetsql.customers` c left outer join `targetsql.orders` o on
c.customer_id = o.customer_id right outer join `targetsql.order_items` ot
on o.order_id = ot.order_id
) as x
order by time_to_delivery desc;
```

Screeen shot of the output:

customer_state	mean_freight_value	diff_estimated_delive	time_to_delivery
AM	33.21	42	32
AM	33.21	44	27
AM	33.21	53	24
AM	33.21	43	26
AM	33.21	57	29
AM	33.21	57	29
AM	33.21	57	29
AM	33.21	52	23
AM	33.21	43	31
AM	33.21	62	20

4. Sort the data to get the following:

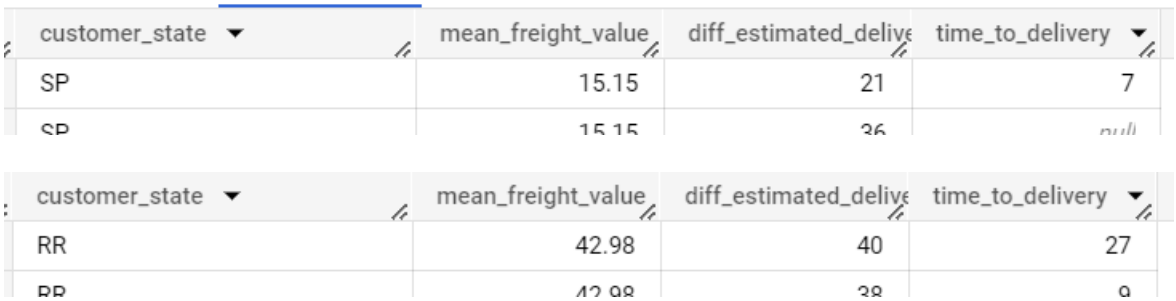
   Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5

   SQL Query:

```
select x.customer_state, round(avg(x.freight_value) over (partition by
x.customer_state),2) as mean_freight_value,
datetime_diff(x.estimated_delivery_date, x.purchase_date, day) as
```

```
diff_estimated_delivery, datetime_diff(x.cust_delivery_date,
x.purchase_date, day) as time_to_delivery
from
(
select
  ot.freight_value, c.customer_state,
  extract(date FROM o.order_purchase_timestamp) as purchase_date,
  extract(date FROM o.order_estimated_delivery_date) as
estimated_delivery_date,
  extract(date FROM o.order_delivered_customer_date) as
cust_delivery_date
from `targetsql.customers` c left outer join `targetsql.orders` o on
c.customer_id = o.customer_id right outer join `targetsql.order_items` ot
on o.order_id = ot.order_id
) as x
 order by mean_freight_value asc/desc
```

Screeen shot of the output:



customer_state	mean_freight_value	diff_estimated_delive	time_to_delivery
SP	15.15	21	7
SP	15.15	36	null

customer_state	mean_freight_value	diff_estimated_delive	time_to_delivery
RR	42.98	40	27
RR	42.98	38	9

Insights:

- RR state has the highest mean freight value of 42.98 and SP state has the lowest mean freight value of 15.15.

Top 5 states with highest/lowest average time to delivery

SQL Query:

```
select x.customer_state, round(avg(x.freight_value) over (partition by
x.customer_state),2) as mean_freight_value,
datetime_diff(x.estimated_delivery_date, x.purchase_date, day) as
diff_estimated_delivery, datetime_diff(x.cust_delivery_date,
x.purchase_date, day) as time_to_delivery
from
(
select
  ot.freight_value, c.customer_state,
  extract(date FROM o.order_purchase_timestamp) as purchase_date,
  extract(date FROM o.order_estimated_delivery_date) as
estimated_delivery_date,
```

```
  extract(date FROM o.order_delivered_customer_date) as
cust_delivery_date
from `targetsql.customers` c left outer join `targetsql.orders` o on
c.customer_id = o.customer_id right outer join `targetsql.order_items` ot
on o.order_id = ot.order_id
) as x
 order by time_to_delivery desc /asc;
```

Screeen shot of the output:

customer_state	mean_freight_value	diff_estimated_delive	time_to_delivery
ES	22.06	29	210
RJ	20.96	20	208
PA	35.83	31	196
SE	36.65	29	195
PI	39.15	40	195
PI	39.15	33	194
SP	15.15	16	191
SP	15.15	23	190
SE	36.65	29	188
MG	20.63	26	188

Insights:

- Maximum days for customer delivery is 210 which is in ES state.

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

SQL Query:

```
select x.customer_state, round(avg(x.freight_value) over (partition by
x.customer_state),2) as mean_freight_value,
datetime_diff(x.estimated_delivery_date, x.purchase_date, day) as
diff_estimated_delivery, datetime_diff(x.cust_delivery_date,
x.purchase_date, day) as time_to_delivery
from
(
select
  ot.freight_value, c.customer_state,
  extract(date FROM o.order_purchase_timestamp) as purchase_date,
```

```
  extract(date FROM o.order_estimated_delivery_date) as
estimated_delivery_date,
  extract(date FROM o.order_delivered_customer_date) as
cust_delivery_date
from `targetsql.customers` c left outer join `targetsql.orders` o on
c.customer_id = o.customer_id right outer join `targetsql.order_items` ot
on o.order_id = ot.order_id
) as x
 order by diff_estimated_delivery desc /asc
```

Screeen shot of the output:

customer_state	mean_freight_value	diff_estimated_delive	time_to_delivery
RS	21.74	156	21
SP	15.15	150	3
SP	15.15	150	3
MA	38.26	147	7
RJ	20.96	145	null
SP	15.15	145	null
DF	21.04	143	null
SP	15.15	141	17
SP	15.15	141	17
RJ	20.96	117	8
RN	35.65	110	54

Insights:

- Minimum days for estimated delivery is in RS state i.e., 3 and maximum days for estimated delivery is 156 which is in RS state.

6. Payment type analysis:

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

    SQL Query:

```
SELECT x.order_year, x.order_month, x.payment_type, x.order_count
```

```
FROM
(
  select p.payment_type, count(o.order_id) as order_count, extract(year
FROM o.order_purchase_timestamp) as order_year, format_date('%B',
o.order_purchase_timestamp) as order_month
  from  `targetsql.orders` o join `targetsql.payments` p on o.order_id =
p.order_id
  group by p.payment_type, extract(year FROM o.order_purchase_timestamp),
format_date('%B', o.order_purchase_timestamp)
) as x
order by x.order_year, x.order_month;
```

Screen shot of the output:

order_year	order_month	payment_type	order_count
2016	December	credit_card	1
2016	October	credit_card	254
2016	October	UPI	63
2016	October	voucher	23
2016	October	debit_card	2
2016	September	credit_card	3
2017	April	voucher	202
2017	April	credit_card	1846
2017	April	UPI	496
2017	April	debit_card	27
2017	August	credit_card	3284

Insights:

- For UPI payment type, October 2016 has lowest count of orders i.e., 63 and January 2018 has highest count of orders i.e., 1518.
- For Credit Card payment type, December 2016 has lowest count of orders i.e., 1 and November 2017 has highest count of orders i.e., 5897.
- For Voucher payment type, October 2018 has lowest count of orders i.e., 4 and January 2018 has highest count of orders i.e., 416.
- For the not defined payment type, August 2018 has 2 orders and September 2018 has 1 order.

2. Count of orders based on the no. of payment installments.

SQL Query:

```
SELECT x.payment_installments, x.order_count
FROM
```

```
(
  select p.payment_installments, count(o.order_id) as order_count
  from  `targetsql.orders` o join `targetsql.payments` p on o.order_id =
p.order_id
  group by p.payment_installments
) as x
order by x.payment_installments, x.order_count;
```

Screen shot of the output:

payment_installment	order_count
0	2
1	52546
2	12413
3	10461
4	7098
5	5239
6	3920
7	1626
8	4268
9	644
10	5328

Insights:

- Payment installments varies from 1 installment to 24 installments.
- The maximum number of orders i.e., 52546 are having a single installment and minimum number of orders i.e., 1 is having 22 and 23 installments.
- Most orders are having single payment installment to 4 installments.

7. Actionable Insights & Recommendations (10 points)
    - Actionable Insights:
        - In Brazil, E-commerce has grown exponentially in 2017 and 2018. Brazil has experienced highest number of orders in January and March 2018.

    - The data suggest that the Brazilian Customers tend to buy is in the afternoon and evening.
    - The count of orders has grown month on month from January 2017 and peaked at January 2018.
    - In Brazil, SP state has the highest number of customers. The lowest number of customers are in RR and AP.
    - The highest percentage growth (99.9%) is seen in the months of August 2017, Feb 2018, March 2018, May 2018, June 2018.
    - The maximum days for customer delivery of a order is 210 days and minimum days for an customer delivery of a order is 2 days.
    - RR state has the highest mean freight value of 42.98 and SP state has the lowest mean freight value of 15.15.
    - In Brazil, the customers has been seen using Credit Card as payment type used maximum time while ordering. After that UPI payment type is used and lowest number of payments has been done using voucher payment type.
    - Also, it has been seen that the maximum number of orders are done using a single payment installment followed by 2 and 3 payment installments.
- Recommendations:
    - It should be suggested that the seasonal growth has been seen in Jan to March months. We can target these months to run marketing campaign to get the maximum orders.
    - Also, it is of concern that RR and AP are the states that has lowest number of customer. Steps should be taken to grow the customers in these states.
    - Also, we should focus on customers and orders with 1 to 3 installments.
    - Also, as freight value in SP state is lowest, it can be seen as opportunity to grow the customers and orders by using lowest freight value to provide discounts.