Target project-SQL (by Abhilash VA)

Q5 Analysis on sales, freight and delivery time

Q5.1--Calculate days between purchasing, delivering and estimated delivery

```
select order_id,order_status,
format_date('%d/%m/%Y',order_purchase_timestamp) as cust_purchased_on ,
FORMAT_DATE('%d/%m/%Y',order_estimated_delivery_date) as estimated_date ,
FORMAT_DATE('%d/%m/%Y',order_delivered_customer_date) as deliverd_on,
date_diff(order_delivered_customer_date,order_purchase_timestamp,day) as Actual_delivery,
date_diff(order_estimated_delivery_date,order_purchase_timestamp,day) as Estimated_delivery
  from
`oceanic-base-302317.Demo_target.Orders`
where order_status='delivered' ;
```

order_id	order_status	cust_purchased_on	estimated_date	deliverd_on	Actual_deliv...	Estimated_d...
cec8f5f7a13e5ab934a486ec9e...	delivered	17/03/2017	18/05/2017	07/04/2017	20	61
58527ee4726911bee84a0f42c...	delivered	20/03/2017	18/05/2017	30/03/2017	10	58
10ed5499d1623638ee810eff1...	delivered	21/03/2017	18/05/2017	18/04/2017	28	57
818996ea247803ddc123789f2...	delivered	20/08/2018	04/10/2018	29/08/2018	9	44
d195cac9ccaa1394ede717d38...	delivered	12/08/2018	04/10/2018	23/08/2018	10	52
64eeb35d3ade7fcdff9fbb1ca5...	delivered	16/08/2018	04/10/2018	23/08/2018	6	48
2691ae869f13b10f3d356461b...	delivered	22/08/2018	04/10/2018	29/08/2018	6	42
1cd147d1c0fe18f3b742a3533...	delivered	20/08/2018	04/10/2018	29/08/2018	8	44
b36d2e6b1781d380e140608a...	delivered	09/08/2018	04/10/2018	22/08/2018	12	55
88ab6b0ede7f19c65b5b71771...	delivered	13/08/2018	04/10/2018	29/08/2018	16	51
c15790c4480e97b6a152024a4...	delivered	20/08/2018	04/10/2018	30/08/2018	10	44
cc8068a058758e65bf6626652...	delivered	21/08/2018	04/10/2018	27/08/2018	6	43
44879a8f19c5e8a5e9278477b...	delivered	23/08/2018	04/10/2018	30/08/2018	6	41

Count of orders where (actual_delivery > estimated delivery)

```
select count(*) count_of_orders
from (select order_id,order_status,
format_date('%d/%m/%Y',order_purchase_timestamp) as cust_purchased_on ,
FORMAT_DATE('%d/%m/%Y',order_estimated_delivery_date) as estimated_date ,
FORMAT_DATE('%d/%m/%Y',order_delivered_customer_date) as deliverd_on,
date_diff(order_delivered_customer_date,order_purchase_timestamp,day) as Actual_delivery,
date_diff(order_estimated_delivery_date,order_purchase_timestamp,day) as Estimated_delivery
  from
`oceanic-base-302317.Demo_target.Orders`
where order_status='delivered' and
date_diff(order_delivered_customer_date,order_purchase_timestamp,day)>date_diff(order_estimated_delivery_
date,order_purchase_timestamp,day));
```

count_of_orders
7307

```
select order_id,order_status,
format_date('%d/%m/%Y',order_purchase_timestamp) as cust_purchased_on ,
FORMAT_DATE('%d/%m/%Y',order_estimated_delivery_date) as estimated_date ,
FORMAT_DATE('%d/%m/%Y',order_delivered_customer_date) as deliverd_on,
date_diff(order_delivered_customer_date,order_purchase_timestamp,day) as Actual_delivery,
date_diff(order_estimated_delivery_date,order_purchase_timestamp,day) as Estimated_delivery
  from
`oceanic-base-302317.Demo_target.Orders`
where order_status='delivered' and
date_diff(order_delivered_customer_date,order_purchase_timestamp,day)>date_diff(order_estimated_delivery_
date,order_purchase_timestamp,day)
```

;

_id	order_status	cust_purchased_on	estimated_date	deliverd_on	Actual_deliv...	Estimated_d...
9ec344d3bf029ff83a161c...	delivered	08/04/2017	18/05/2017	22/05/2017	43	39
8ab302de39ef3087f6a3a...	delivered	25/04/2018	06/06/2018	06/06/2018	42	41
2e3f591a1f8ab11d51f2ae...	delivered	28/12/2017	06/02/2018	20/02/2018	54	39
e1f2d1b360c34d90175e1...	delivered	31/01/2017	15/03/2017	30/03/2017	57	42
9503f2ebd9f53deba18716...	delivered	31/03/2017	10/05/2017	19/09/2017	172	39
3fbfb4e41c26246ea6097...	delivered	17/04/2018	29/05/2018	13/06/2018	56	41
6b80d080e344c3698d9d...	delivered	19/04/2018	29/05/2018	29/05/2018	40	39
15e915694747f4d7f7501...	delivered	18/04/2018	29/05/2018	08/06/2018	50	40

Actionable Insights

- This table gives the information about Actual days and Estimated days for a particular order.
- If the Actual day is lesser than the Estimated days then the Customer is satisfied .
- Note : We have considered only those orders for which order_status =’Delivered’
- We have 7307 orders for which Actual days is greater than Estimated days
- We have 96478 orders for status =’Delivered’
- Approx around 7.75 % of the total orders which has order_status= ‘Delivered’ have Actual days greater than Estimated days

Recommendations

- Make sure that customers gets there order in time or before to Estimated time.
- Analyse the difficulties faced for delivering the 7.75% of orders ,so that it could be taken care from next orders.

Q5.2 -- Create columns:

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

```
select
order_id,customer_id,order_status,
format_date('%d/%m/%Y',order_purchase_timestamp) as cust_purchased_on ,
FORMAT_DATE('%d/%m/%Y',order_estimated_delivery_date) as estimated_date ,
FORMAT_DATE('%d/%m/%Y',order_delivered_customer_date) as deliverd_on,
date_diff(order_delivered_customer_date,order_purchase_timestamp,day) as time_to_delivery ,
date_diff(order_estimated_delivery_date,order_delivered_customer_date,day) as diff_estimated_delivery
 from
`oceanic-base-302317.Demo_target.Orders`
```

order_id	customer_id	order_status	cust_purchased_on	estimated_date	deliverd_on	time_to_delivery	diff_estimated_...
770d331c84e5b214b...	6c57e61193691...	canceled	07/10/2016	29/11/2016	14/10/2016	7	45
1950d777989f6a877...	1bccb206de9f0f...	canceled	19/02/2018	09/03/2018	21/03/2018	30	-12
2c45c33d2f9cb8ff8b...	de4caa97afa80c...	canceled	09/10/2016	08/12/2016	09/11/2016	30	28
dabf2b0e35b423f94...	5cdec0bb8cbdf5...	canceled	09/10/2016	30/11/2016	16/10/2016	7	44
8beb59392e21af5eb...	bf609b5741f716...	canceled	08/10/2016	30/11/2016	19/10/2016	10	41
b60b53ad0bb7dacac...	2f9902d85fcd93...	delivered	10/05/2017	18/05/2017	23/05/2017	12	-5
276e9ec344d3bf029f...	d33e520a99eb4...	delivered	08/04/2017	18/05/2017	22/05/2017	43	-4
1a0b31f08d0d7e879...	7e769bb9acb55...	delivered	11/04/2017	18/05/2017	18/04/2017	6	29

Actionable Insights

- If diff_estimated_delivery column is negative than (ie estimated_date < delivered_on) ,which implies customer waited longer than estimated date.
- If diff_estimated_delivery = -7 → Customer got his order late by 7 days compared to estimated date.
- If diff_estimated_delivery = 7 → Customer got his order early by 7 days compared to estimated date.

Q5.3--**Group data by state, take mean of freight_value, time_to_delivery, diff_estimated_delivery**

```
select coust.customer_state,ROUND(avg(order_items1.freight_value),2) as AVG_freight_value,
ROUND(avg(order1.time_to_delivery),2) AVG_time_to_delivery,
ROUND(avg(order1.diff_estimated_delivery),2) AS AVG_diff_estimated_delivery FROM
`oceanic-base-302317.Demo_target.Order_items` as order_items1
join

(select
order_id,customer_id,order_status,
format_date('%d/%m/%Y',order_purchase_timestamp) as cust_purchased_on ,
FORMAT_DATE('%d/%m/%Y',order_estimated_delivery_date) as estimated_date ,
FORMAT_DATE('%d/%m/%Y',order_delivered_customer_date) as deliverd_on,
date_diff(order_delivered_customer_date,order_purchase_timestamp,day) as time_to_delivery ,
date_diff(order_estimated_delivery_date,order_delivered_customer_date,day) as diff_estimated_delivery
 from
`oceanic-base-302317.Demo_target.Orders`) as order1
on order1.order_id=order_items1.order_id
join `oceanic-base-302317.Demo_target.Customers` as coust
on coust.customer_id=order1.customer_id
group by 1
```

customer_state	AVG_freight_value	AVG_time_to_delivery	AVG_diff_estimated_delivery
MT	28.17	17.51	13.64
MA	38.26	21.2	9.11
AL	35.84	23.99	7.98
SP	15.15	8.26	10.27
MG	20.63	11.52	12.4
PE	32.92	17.79	12.55
RJ	20.96	14.69	11.14
DF	21.04	12.5	11.27
RS	21.74	14.71	13.2
SE	36.65	20.98	9.17
PR	20.53	11.48	12.53
PA	35.83	23.3	13.37
BA	26.36	18.77	10.12

Actionable Insights

- Lesser the value of AVG_time_to_delivery for a state → implies that customers in this state gets orders faster compared to other states (Customer is happy)
- Greater the value of AVG_diff_estimated_delivery for a state → implies that customers in this state gets orders faster with respect to estimated date when compared to other state (Customer is happy)
- Lesser the value of AVG_freight_value for a state → implies that cost for delivery of orders within that state is cheap compared to other state.

Recommendations

- Try to decrease the AVG_freight (by decreasing freight_value per order by state) value for a state which has low AVG_time_to_delivery → since Avg_time_ro_delivery is less it is more likely that this state have strong delivery connection , If freight_value is decreased → then cost per order will become less → more customers.
- States which has large AVG_time_to_delivery and small AVG_diff_estimated_delivery →Give more importance to this because customers are not satisfied by the delivery time.

Q5.4--Sort the data to get the following

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

Highest freight value :

```
select coust.customer_state,ROUND(avg(order_items1.freight_value),2) as AVG_freight_value,
ROUND(avg(order1.time_to_delivery),2) AVG_time_to_delivery,
ROUND(avg(order1.diff_estimated_delivery),2) AS AVG_diff_estimated_delivery FROM
`oceanic-base-302317.Demo_target.Order_items` as order_items1
join

(select
order_id,customer_id,order_status,
format_date('%d/%m/%Y',order_purchase_timestamp) as cust_purchased_on ,
FORMAT_DATE('%d/%m/%Y',order_estimated_delivery_date) as estimated_date ,
FORMAT_DATE('%d/%m/%Y',order_delivered_customer_date) as deliverd_on,
date_diff(order_delivered_customer_date,order_purchase_timestamp,day) as time_to_delivery ,
date_diff(order_estimated_delivery_date,order_delivered_customer_date,day) as diff_estimated_delivery
 from
`oceanic-base-302317.Demo_target.Orders`) as order1
on order1.order_id=order_items1.order_id
join `oceanic-base-302317.Demo_target.Customers` as coust
on coust.customer_id=order1.customer_id
group by 1
order by 2 desc
limit 5
```

customer_state	AVG_freight...	AVG_time_t...	AVG_diff_es...
RR	42.98	27.83	17.43
PB	42.72	20.12	12.15
RO	41.07	19.28	19.08
AC	40.07	20.33	20.01
PI	39.15	18.93	10.68

Lowest freight value :

```
select coust.customer_state,ROUND(avg(order_items1.freight_value),2) as AVG_freight_value,
ROUND(avg(order1.time_to_delivery),2) AVG_time_to_delivery,
ROUND(avg(order1.diff_estimated_delivery),2) AS AVG_diff_estimated_delivery FROM
`oceanic-base-302317.Demo_target.Order_items` as order_items1
join

(select
order_id,customer_id,order_status,
format_date('%d/%m/%Y',order_purchase_timestamp) as cust_purchased_on ,
FORMAT_DATE('%d/%m/%Y',order_estimated_delivery_date) as estimated_date ,
FORMAT_DATE('%d/%m/%Y',order_delivered_customer_date) as deliverd_on,
date_diff(order_delivered_customer_date,order_purchase_timestamp,day) as time_to_delivery ,
date_diff(order_estimated_delivery_date,order_delivered_customer_date,day) as diff_estimated_delivery
 from
`oceanic-base-302317.Demo_target.Orders`) as order1
on order1.order_id=order_items1.order_id
join `oceanic-base-302317.Demo_target.Customers` as coust
on coust.customer_id=order1.customer_id
group by 1
order by 2
limit 5
```

customer_state	AVG_freight...	AVG_time_t...	AVG_diff_es...
SP	15.15	8.26	10.27
PR	20.53	11.48	12.53
MG	20.63	11.52	12.4
RJ	20.96	14.69	11.14
DF	21.04	12.5	11.27

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

Highest average time to delivery :

```
select coust.customer_state,ROUND(avg(order_items1.freight_value),2) as AVG_freight_value,
ROUND(avg(order1.time_to_delivery),2) AVG_time_to_delivery,
ROUND(avg(order1.diff_estimated_delivery),2) AS AVG_diff_estimated_delivery FROM
`oceanic-base-302317.Demo_target.Order_items` as order_items1
join

(select
order_id,customer_id,order_status,
format_date('%d/%m/%Y',order_purchase_timestamp) as cust_purchased_on ,
FORMAT_DATE('%d/%m/%Y',order_estimated_delivery_date) as estimated_date ,
FORMAT_DATE('%d/%m/%Y',order_delivered_customer_date) as deliverd_on,
date_diff(order_delivered_customer_date,order_purchase_timestamp,day) as time_to_delivery ,
date_diff(order_estimated_delivery_date,order_delivered_customer_date,day) as diff_estimated_delivery
 from
`oceanic-base-302317.Demo_target.Orders`) as order1
on order1.order_id=order_items1.order_id
join `oceanic-base-302317.Demo_target.Customers` as coust
on coust.customer_id=order1.customer_id
group by 1
```

```
order by 3 desc
limit 5
```

customer_state	AVG_freight...	AVG_time_t...	AVG_diff_es...
RR	42.98	27.83	17.43
AP	34.01	27.75	17.44
AM	33.21	25.96	18.98
AL	35.84	23.99	7.98
PA	35.83	23.3	13.37

Lowest average time to delivery :

```
select coust.customer_state,ROUND(avg(order_items1.freight_value),2) as AVG_freight_value,
ROUND(avg(order1.time_to_delivery),2) AVG_time_to_delivery,
ROUND(avg(order1.diff_estimated_delivery),2) AS AVG_diff_estimated_delivery FROM
`oceanic-base-302317.Demo_target.Order_items` as order_items1
join

(select
order_id,customer_id,order_status,
format_date('%d/%m/%Y',order_purchase_timestamp) as cust_purchased_on ,
FORMAT_DATE('%d/%m/%Y',order_estimated_delivery_date) as estimated_date ,
FORMAT_DATE('%d/%m/%Y',order_delivered_customer_date) as deliverd_on,
date_diff(order_delivered_customer_date,order_purchase_timestamp,day) as time_to_delivery ,
date_diff(order_estimated_delivery_date,order_delivered_customer_date,day) as diff_estimated_delive
ry
from
`oceanic-base-302317.Demo_target.Orders`) as order1
on order1.order_id=order_items1.order_id
join `oceanic-base-302317.Demo_target.Customers` as coust
on coust.customer_id=order1.customer_id
group by 1
order by 3
limit 5
```

customer_state	AVG_freight...	AVG_time_t...	AVG_diff_es...
SP	15.15	8.26	10.27
PR	20.53	11.48	12.53
MG	20.63	11.52	12.4
DF	21.04	12.5	11.27
SC	21.47	14.52	10.67

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

```
select coust.customer_state,avg(order_items1.freight_value) as AVG_freight_value,
avg(order1.time_to_delivery) AVG_time_to_delivery,
avg(order1.diff_estimated_delivery) AS AVG_diff_estimated_delivery FROM
`oceanic-base-302317.Demo_target.Order_items` as order_items1
join

(select
order_id,customer_id,order_status,
format_date('%d/%m/%Y',order_purchase_timestamp) as cust_purchased_on ,
FORMAT_DATE('%d/%m/%Y',order_estimated_delivery_date) as estimated_date ,
FORMAT_DATE('%d/%m/%Y',order_delivered_customer_date) as deliverd_on,
date_diff(order_delivered_customer_date,order_purchase_timestamp,day) as time_to_delivery ,
date_diff(order_estimated_delivery_date,order_delivered_customer_date,day) as diff_estimated_delivery
 from
`oceanic-base-302317.Demo_target.Orders`) as order1
on order1.order_id=order_items1.order_id
join `oceanic-base-302317.Demo_target.Customers` as coust
on coust.customer_id=order1.customer_id
group by 1
having avg(order1.time_to_delivery) > avg(order1.diff_estimated_delivery)
order by 3
limit 5
```

customer_state	AVG_freight...	AVG_time_t...	AVG_diff_es...
DF	21.0413549...	12.5014861...	11.2747346...
SC	21.4703687...	14.5209858...	10.6688628...
RJ	20.9609239...	14.6893821...	11.1444931...
RS	21.7358043...	14.7082993...	13.2030001...
GO	22.7668152...	14.9481774...	11.3728590...

Actionable Insights

- Top 5 States which has high avg freight_value →Customers pay more amount for delivery
- Top 5 states which has high avg_time_to_delivery → Customers need to wait a bit longer period for their orders.

Recommendations

- The state which has high avg_time_to_delivery needs to work of their delivery people's network.
- The state which has high avg_freight_value ,needs to work on building up delivery connections within states ,so the cost of delivering from one place to another place decreases.

Q6. Payment type analysis:

Tableau Dashboard → https://public.tableau.com/app/profile/abhilash.v.a/viz/project_wk/Q6-DASHBOARD?publish=yes

Q6.1--Month over Month count of orders for different payment types

```
select y.year,y.month,sum(y.credit_card) as credit_card,sum(y.voucher) as voucher
,sum(y.not_defined) as not_defined ,sum(y.debit_card)as debit_card,sum(y.UPI) as UPI,
sum(y.credit_card)+sum(y.voucher)+sum(y.not_defined)+sum(y.debit_card)+sum(y.UPI) as total_
from (
select x.year,x.month,
case
when x.payment_type="credit_card"
then x.cnt
else 0
end as credit_card,
case
when x.payment_type="voucher"
then x.cnt
else 0
end as voucher,
case
when x.payment_type="not_defined"
then x.cnt
else 0
end as not_defined,
case
when x.payment_type="debit_card"
then x.cnt
else 0
end as debit_card,
case
when x.payment_type="UPI"
then x.cnt
else 0
end as UPI
from (
select extract(year from Ord.order_purchase_timestamp) as year,extract(month from Ord.order_purchase_time
stamp) as month ,pay.payment_type,count(*) as cnt from
`oceanic-base-302317.Demo_target.Orders` as Ord
left join `Demo_target.payments` as pay
on pay.order_id=Ord.order_id
group by 1,2,3
order by 1,2,3) as x) as y
group by  1,2
order by 1,2;
```

year	month	credit_card	voucher	not_defined	debit_card	UPI	total_
2016	9	3	0	0	0	0	3
2016	10	254	23	0	2	63	342
2016	12	1	0	0	0	0	1
2017	1	583	61	0	9	197	850
2017	2	1356	119	0	13	398	1886
2017	3	2016	200	0	31	590	2837
2017	4	1846	202	0	27	496	2571
2017	5	2853	289	0	30	772	3944
2017	6	2463	239	0	27	707	3436
2017	7	3086	364	0	22	845	4317
2017	8	3284	294	0	34	938	4550
2017	9	3283	287	0	43	903	4516
2017	10	3524	291	0	52	993	4860

Actionable Insights

year	month	credit_card	voucher	not_defined	debit_card	UPI	total_
2017	11	5897	387	0	70	1509	7863
2018	1	5520	416	0	109	1518	7563
2018	3	5691	391	0	78	1352	7512
2018	4	5455	370	0	97	1287	7209

- These are the top 4 months which has received highest number of payments.
- 2017 Nov month →Received number of payments[7893] → out of which [5897] are from credit cards.
- For 2017 Nov month → credit card = 74.99 % of total payment → upi =19.19%
- Usually most of the payments are done by credit cards.

Recommendations

- Since credit card is the most used payment type , providing offers on them (credit card) helps in customer attraction.
- UPI is the second most used payment , providing offers on them(UPI) helps in customer attraction.
- From the above data , we have observed that in the month of NOV ,JAN ,MARCH happens most of the payments , providing good deals on these months helps in customer attractions.
- And also make sure that payment server are stable on these months.

Q6.1 Distribution of payment installments and count of orders

```
with pre_1 as (select  payment_installments ,count(order_id) count_of_orders from `oceanic-base-
302317.Demo_target.payments`
group by 1),
pre_2 as (select sum(count_of_orders) from pre_1)
select payment_installments,count_of_orders,round(count_of_orders/(select * from pre_2)*100,2) as pert_sh
are from pre_1
order by 2 desc
```

payment_installments	count_of_orders	pert_share
1	52546	50.58
2	12413	11.95
3	10461	10.07
4	7098	6.83
10	5328	5.13
5	5239	5.04
8	4268	4.11
6	3920	3.77
7	1626	1.57
9	644	0.62
12	133	0.13
15	74	0.07
18	27	0.03

Actionable Insights

- **Most** of the orders are one month instalment i.e. about 50.5%
- 16,22,0,23,22-month payment instalments have 0% contribution
- Customers are most unlikely to pay amount in one shot, → 0 month payment instalment has 0% contribution

Recommendations

- Since payment instalment for 2,3 months are approx. 11 %, give more importance to this type. It will generate more profit (because of the interest and more time period)→ this payment instalment has more potential.
- Since 50% of customers choose 1 month instalment payment , Less interest on these will attract more customers .

Q4 Impact on Economy: Analyze the money movemented by e-commerce by looking at order prices, freight and other

Tableau dashboard--https://public.tableau.com/app/profile/abhilash.v.a/viz/project_wk/q4-Dashboard?publish=yes

Q4.1 Get % increase in cost of orders from 2017 to 2018 (include months between Jan to Aug only)

```
with pre_comp as
(
select extract(year from o.order_purchase_timestamp) as year,
extract(month from o.order_purchase_timestamp) as month,
count(oi.order_item_id) as no_items,
avg(oi.freight_value) as total_freght,
avg(oi.price) as price
from `oceanic-base-302317.Demo_target.Order_items` as oi
join `oceanic-base-302317.Demo_target.Orders` as o
on o.order_id=oi.order_id
group by  1,2
having  (year=2017 or year=2018) and (month in (1,2,3,4,5,6,7,8))
order by 1,2),
date_17 as (
  select * from pre_comp where year=2017
),
date_18 as (
  select * from pre_comp where year=2018

),
calcu_all as (
select d17.month,d17.no_items as no_items_2017,
d18.no_items as no_items_2018,
d17.total_freght as total_freght_2017,
d18.total_freght as total_freght_2018,
d17.price as price_2017,
d18.price as price_2018
from date_17 as d17
join date_18 as d18
on d17.month=d18.month),
pre_21 as (
select *,
total_freght_2017+price_2017 as cost_in_2017,
total_freght_2018+price_2018 as cost_in_2018,
round((no_items_2018-no_items_2017)/no_items_2017*100,2) as perct_change_no_items,
round((total_freght_2018-total_freght_2017)/total_freght_2017*100,2)   as perct_change_in_freght,
round((price_2018-price_2017)/price_2017*100,2 )as perct_price
from calcu_all
order by 1)
select *,round((cost_in_2018-cost_in_2017)/cost_in_2017*100,2) as perct_change_in_cost from pre_21
```

month	no_items...	no_item...	total_freght...	total_freght...	price_2017	price_2018	cost_in_2017	cost_in_2018	perct_change_...	perct_chang...	perct_price	perct_change_in_cost
1	955	8208	17.6708062...	19.1607614...	125.982062...	115.744439...	143.652869...	134.905201...	759.48	8.43	-8.13	-6.09
2	1951	7672	19.9782675...	18.6040471...	126.757057...	110.033721...	146.735325...	128.637768...	293.23	-6.88	-13.19	-12.33
3	3000	8217	19.2347633...	20.9216721...	124.781433...	119.656010...	144.016196...	140.577682...	173.9	8.77	-4.11	-2.39
4	2684	7975	19.5584985...	20.4451774...	134.101054...	124.971504...	153.659552...	145.416682...	197.13	4.53	-6.81	-5.36
5	4136	7925	19.3713273...	19.3393236...	122.357625...	125.743555...	141.728953...	145.082879...	91.61	-0.17	2.77	2.37
6	3583	7078	19.5156126...	22.2595083...	120.859224...	122.227226...	140.374836...	144.486734...	97.54	14.06	1.13	2.93
7	4519	7092	19.2388006...	23.0147786...	110.208338...	126.270053...	129.447138...	149.284832...	56.94	19.63	14.57	15.32
8	4910	7248	19.1920407...	20.5052621...	116.898509...	117.920299...	136.090549...	138.425561...	47.62	6.84	0.87	1.72

Actionable Insights

- The no_itmes (count of orders) in 2017 for (jan-aug) months has increased in 2018 (jan-aug)→ which implies that sales are good in 2018 when compared to 2017
- 759 % increase in orders (2017 jan month → 2018 jan month)
- Price has been decreased in 2018 when compared to 2017 (prect_price are in negative)[for Jan ,feb ,march ,April]
- -13% drop-in price in 2018 feb compared to 2107 feb
- Freght valye has increased in 2018 when compared to 2017 (jan ,march ,April ,jun ,jul ,aug)
- Cost=freght value+price (corelated to each other liner relation)
- Highest cost percentage →Jul month
- Lowest cost percentage →Feb month

Recommendations

- JUL and AUG month doesn't have 2x growth rate on orders, check what is reason behind this .
- Jul month has most increase in price ie 14% and 19% change in freight vale , This will impact the customers since it increase the cost of the product.

Q4.2 Mean & Sum of price and freight value by customer state

```
select cust.customer_state,avg(orderitem.freight_value) AS freight_value_AVG ,
sum(orderitem.freight_value) as freight_value_SUM,avg(orderitem.price) as price_AVG ,
sum(orderitem.price) as price_SUM
from
`oceanic-base-302317.Demo_target.Orders` as order1
join `oceanic-base-302317.Demo_target.Order_items` as orderitem
on orderitem.order_id=order1.order_id
join `oceanic-base-302317.Demo_target.Customers` as cust
```

```
on cust.customer_id=order1.customer_id
group by cust.customer_state
```

customer_state	freight_value_AVG	freight_value_SUM	price_AVG	price_SUM
MT	28.1662843601896	29715.43000000...	148.297184834...	156453.52999...
MA	38.257002427184...	31523.77000000...	145.204150485...	119648.21999...
AL	35.843671171171...	15914.58999999...	180.889211711...	80314.81
SP	15.147275390419...	718723.0699999...	109.653629159...	5202955.0500...
MG	20.630166806306...	270853.4600000...	120.748574148...	1585308.0299...
PE	32.917862679955...	59449.6599999999	145.508322259...	262788.02999...
RJ	20.960923931682...	305589.3100000...	125.117818094...	1824092.6699...
DF	21.041354945968...	50625.49999999...	125.770548628...	302603.93999...
RS	21.735804330392...	135522.7400000...	120.337453087...	750304.02000...
SE	36.653168831168...	14111.46999999...	153.041168831...	58920.850000...
PR	20.531651567944...	117851.6800000...	119.004139372...	683083.76000...
PA	35.832685185185...	38699.30000000...	165.692416666...	178947.80999...
BA	26.363958936562...	100156.6799999...	134.601208212...	511349.99000...

Actionable Insights

customer_state	freight_valu...	freight_valu...	price_AVG	price_SUM
PB	42.7238039...	25719.7300...	191.475215...	115268.079...
AL	35.8436711...	15914.5899...	180.889211...	80314.81
AC	40.0733695...	3686.74999...	173.727717...	15982.9499...

- Top 3 states which has highest avg price and sum price

customer_state	freight_valu...	freight_valu...	price_AVG	price_SUM
RR	42.9844230...	2235.19	150.565961...	7829.42999...
PB	42.7238039...	25719.7300...	191.475215...	115268.079...
RO	41.0697122...	11417.3799...	165.973525...	46140.6400...

- Top 3 states which has highest avg freight value and sum freight value

Recommendations

- Since PB ,AL , AC has high price value → implies that customers in this state buys more or buys products which is expensive . recommending costly products for these customers helps increase in sales(and visa versa for low price value states)

- Since RR,PB,RO has high freight value → implies that in these states the transportation cost for delivering from one place to another is more →Look into the connection or network of delivery peoples.

Q3 Evolution of E-commerce orders in the Brazil region

Tableau Dashboard

https://public.tableau.com/app/profile/abhilash.v.a/viz/project_wk/Q3-dASHBOARD?publish=yes

Q3.1 Get month on month orders by region, states

```
with pre_comp as (select customer_state,customer_city,extract(month from order_purchase_timestamp) as mon
th,count(order_id) as count1
from (
select * from
`oceanic-base-302317.Demo_target.Orders` as ord
join `oceanic-base-302317.Demo_target.Customers` as cust
on cust.customer_id=ord.customer_id)
group by 1,2,3
order by 1,2,3),
cpm1 as (
select customer_state,customer_city,
case
when month=1
then count1
else 0
end as JAN,
case
when month=2
then count1
else 0
end as FEB,
case
when month=3
then count1
else 0
end as MAR,
case
when month=4
then count1
else 0
end as APRIL,
case
when month=5
then count1
else 0
end as MAY,
case
when month=6
then count1
else 0
end as JUN,
case
when month=7
then count1
```

```
else 0
end as JUL,
case
when month=8
then count1
else 0
end as AUG,
case
when month=9
then count1
else 0
end as SEP,
case
when month=10
then count1
else 0
end as OCT,
case
when month=11
then count1
else 0
end as NOV,
case
when month=12
then count1
else 0
end as DEC,


from pre_comp


),
pre2 as (

SELECT customer_state,customer_city,SUM(JAN) AS JAN,
SUM(FEB) AS FEB,
SUM(MAR) AS MAR,
SUM(APRIL) AS APRIL,
SUM(MAY) AS MAY,
SUM(JUN) AS JUN,
SUM(JUL) AS JUL,
SUM(AUG) AS AUG,
SUM(SEP) AS SEP,
SUM(OCT) AS OCT,
SUM(NOV) AS NOV,
SUM(DEC) AS DEC,
SUM(JAN+FEB+MAR+APRIL+MAY+JUN+JUL+AUG+SEP+OCT+NOV+DEC) AS TOTAL_COUNT
from cpm1
GROUP BY 1,2
ORDER BY 1,2)
select * from pre2
```

customer_state	customer_city	JAN	FEB	MAR	APRIL	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	TOTAL_COUNT
AC	brasileia	0	1	0	0	0	0	0	0	0	0	0	0	1
AC	cruzeiro do sul	0	0	0	0	1	0	0	0	0	0	0	2	3
AC	epitaciolandia	0	0	0	0	0	0	0	0	0	1	0	0	1
AC	manoel urbano	0	0	0	0	0	0	0	0	1	0	0	0	1
AC	porto acre	0	0	0	1	0	0	0	0	0	0	0	0	1
AC	rio branco	8	5	4	7	9	7	9	7	3	5	4	2	70
AC	senador guiomard	0	0	0	0	0	0	0	0	1	0	0	1	2
AC	xapuri	0	0	0	1	0	0	0	0	0	0	1	0	2
AL	agua branca	0	1	0	0	0	0	0	0	0	0	0	0	1
AL	anadia	1	0	0	0	1	0	0	0	0	0	0	0	2
AL	arapiraca	3	3	2	2	6	2	1	2	2	2	3	1	29
AL	atalaia	0	0	0	0	0	0	0	0	0	0	1	0	1
AL	barra de santo a...	0	0	0	0	0	0	0	2	0	0	0	0	2

Satate wise analysis

customer_state	JAN	FEB	MAR	APRIL	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	TOTAL_COUNT
AC	8	6	4	9	10	7	9	7	5	6	5	5	81
AL	39	39	40	51	46	34	40	34	20	30	26	14	413
AM	12	16	14	19	19	8	23	9	9	3	10	6	148
AP	11	4	8	5	11	4	7	5	2	3	4	4	68
BA	264	273	340	318	368	307	405	323	170	170	250	192	3380
CE	99	101	126	143	136	121	140	130	77	74	108	81	1336
DF	151	196	207	183	208	220	243	232	97	104	168	131	2140
ES	159	186	182	188	228	204	206	200	93	104	170	113	2033
GO	164	176	199	177	226	184	192	213	88	117	157	127	2020
MA	66	67	77	73	65	59	79	70	42	52	56	41	747
MG	971	1063	1237	1061	1190	1080	1111	1177	511	600	943	691	11635
MS	71	75	79	58	74	76	74	59	33	34	46	36	715
MT	96	84	71	92	104	83	85	78	35	55	74	50	907
PA	82	83	109	107	75	92	96	104	41	58	70	58	975
PB	33	47	55	51	47	51	79	46	29	31	30	37	536
PE	113	146	153	154	174	140	210	170	76	87	126	103	1652
PI	55	46	48	50	56	43	52	43	23	25	31	23	495
PR	443	460	504	500	524	478	523	556	183	225	378	271	5045
RJ	990	1176	1302	1172	1321	1128	1288	1307	612	725	1048	783	12852
RN	51	31	52	42	39	49	56	40	24	27	44	30	485
RO	23	25	29	20	26	22	27	23	16	14	17	11	253
RR	2	7	8	4	3	8	6	0	2	4	2	0	46

City wise analysis

customer_city	JAN	FEB	MAR	APRIL	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	TOTAL_COU...
abadia dos dourados	0	0	1	0	0	0	1	0	1	0	0	0	3
abadiania	1	0	0	0	0	0	0	0	0	0	0	0	1
abaete	0	1	2	0	1	2	2	2	0	0	2	0	12
abaetetuba	0	0	2	2	1	1	1	2	1	0	0	1	11
abaiara	0	0	0	0	1	0	0	0	0	0	1	0	2
abaira	0	0	0	0	1	0	0	1	0	0	0	0	2
abare	0	0	0	0	1	0	0	1	0	0	0	0	2
abatia	0	0	0	1	0	0	0	0	0	1	1	0	3
abdon batista	0	0	0	1	0	0	0	0	0	0	0	0	1
abelardo luz	0	0	0	0	1	1	0	1	1	0	2	0	6
abrantes	0	0	1	0	0	0	0	0	0	1	0	0	2
abre campo	0	0	0	0	0	1	0	0	1	0	1	3	6
abreu e lima	1	3	0	0	2	2	1	0	0	0	0	2	11
acaiaca	0	0	0	0	0	0	1	1	0	0	0	0	2
acailandia	0	0	0	2	0	2	1	0	1	0	1	0	7
acajutiba	0	0	0	0	1	0	0	0	0	0	0	0	1
acarau	2	0	2	0	0	0	0	2	0	1	1	0	8

Actionable Insights

customer_state
SP
RJ
MG
RS
PR

- Top 5 states which has max orders

customer_city
sao paulo
rio de janeiro
belo horizonte
brasilia
curitiba

- Top 5 cities which has max orders

customer_state	customer_city
SP	sao paulo
RJ	rio de janeiro
MG	belo horizonte
DF	brasilia

- Top 4 cities by states which has high orders .

customer_city
baguari
boquim
muliterno
itacurussa
dores de guanhaes

- Top 5 cities which has lowest orders approx 1 order

customer_state
RR
AP
AC
AM
RO

- Top 5 states which has lowest orders

Recommendations

- States which has lowest order rate → needs more advertising and offers.
- Cities which has lowest order rate → needs more advertising and offers .
- Find out the reason behind these (why these states or cites has least order rate).

Q3.2 How are customers distributed in Brazil

State wise :

```
with pre as (select customer_state,count(distinct customer_id) as count_1 from
`oceanic-base-302317.Demo_target.Customers`
group by customer_state),
pre1 as (select sum(count_1) from pre)
select customer_state,count_1,
 round((count_1/(select * from pre1))*100,2) as cust_contribution
 from pre
```

customer_state	count_1	cust_contrib...
RN	485	0.49
CE	1336	1.34
RS	5466	5.5
SC	3637	3.66
SP	41746	41.98
MG	11635	11.7
BA	3380	3.4
RJ	12852	12.92
GO	2020	2.03
MA	747	0.75
PE	1652	1.66
PB	536	0.54
ES	2033	2.04
PR	5045	5.07
RO	253	0.25
MS	715	0.72
PA	975	0.98

Cities wise :

```
with pre as (select customer_city,count(distinct customer_id) as count_1 from
`oceanic-base-302317.Demo_target.Customers`
group by customer_city),
pre1 as (select sum(count_1) from pre)
select customer_city,count_1,
round((count_1/(select * from pre1))*100,2) as cust_contribution
from pre
```

customer_city	count_1	cust_contrib...
acu	3	0.0
ico	8	0.01
ipe	2	0.0
ipu	4	0.0
ita	3	0.0
itu	136	0.14
jau	74	0.07
luz	2	0.0
poa	85	0.09
uba	53	0.05
una	5	0.01
anta	4	0.0
avai	1	0.0
bage	65	0.07
bodo	1	0.0
bora	1	0.0
buri	10	0.01

City wise for each state :

```
with pre1 as (select customer_state,customer_city,min(c1) as c1,max(c2) as c2 from (
select customer_state,customer_city,
count(customer_id) over(partition by customer_state ) as c1,
count(customer_id) over(partition by customer_state ,customer_city) as c2
 from `oceanic-base-302317.Demo_target.Customers`)
 group by 1,2),
 pre_3 as (
 select customer_state,customer_city,c1,c2,round(c2/c1*100,1) as prect_share  from pre1)
 select * from pre_3
 order by 1,2
```

customer_state	customer_city	c1	c2	prect_share
AC	brasileia	81	1	1.2
AC	cruzeiro do sul	81	3	3.7
AC	epitaciolandia	81	1	1.2
AC	manoel urbano	81	1	1.2
AC	porto acre	81	1	1.2
AC	rio branco	81	70	86.4
AC	senador guiomard	81	2	2.5
AC	xapuri	81	2	2.5
AL	agua branca	413	1	0.2
AL	anadia	413	2	0.5
AL	arapiraca	413	29	7.0
AL	atalaia	413	1	0.2
AL	barra de santo antonio	413	2	0.5
AL	barra de sao miguel	413	2	0.5
AL	batalha	413	3	0.7
AL	belem	413	3	0.7
AL	boca da mata	413	2	0.5

Actionable Insights

customer_state	count_1	cust_contrib...
SP	41746	41.98
RJ	12852	12.92
MG	11635	11.7
RS	5466	5.5
PR	5045	5.07

- Top 5 states with max no of customers

customer_state	count_1	cust_contrib...
RR	46	0.05
AP	68	0.07
AC	81	0.08
AM	148	0.15
RO	253	0.25

- Top 5 states with lowest customers

customer_city	count_1	cust_contrib...
sao paulo	15540	15.63
rio de janeiro	6882	6.92
belo horizonte	2773	2.79
brasilia	2131	2.14
curitiba	1521	1.53

- Top 5 cities with highest customers

customer_city	count_1	cust_contrib...
acu	3	0.0
ipe	2	0.0
ipu	4	0.0
ita	3	0.0
luz	2	0.0

- Top 5 cities with lowest customers (approx. 0% contribution)

Recommendations

- We can observe that number of customers is directly proportional to the orders
- Advertising needs to be done for lowest customer base (state or cities)
- More offers needed to be given to attract them .

Q2 In-depth Exploration: Tableau Dashboard

https://public.tableau.com/app/profile/abhilash.v.a/viz/project_wk/Q2-DASHBOARD?publish=yes

Q2.1 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?

```
select *,
lag(no_of_orders,1) over(partition by year order by month) as prev_month_orders,
round((no_of_orders-
lag(no_of_orders,1) over(partition by year order by month)) /lag(no_of_orders,1) over(partition by year o
rder by month)*100,2) as perct_change
from(


select extract(year from order_purchase_timestamp) as year,extract(month from order_purchase_timestamp) a
s month,count(order_id) as no_of_orders
from `oceanic-base-302317.Demo_target.Orders`
group by 1,2
order by 1,2)
order by 1
```

Percent change in orders partitioned by year

year	month	no_of_orders	prev_month...	perct_change
2016	9	4	*null*	*null*
2016	10	324	4	8000.0
2016	12	1	324	-99.69
2017	1	800	*null*	*null*
2017	2	1780	800	122.5
2017	3	2682	1780	50.67
2017	4	2404	2682	-10.37
2017	5	3700	2404	53.91
2017	6	3245	3700	-12.3
2017	7	4026	3245	24.07
2017	8	4331	4026	7.58
2017	9	4285	4331	-1.06
2017	10	4631	4285	8.07

Year to year analysis (for different order status)

month	created	shipped	delivered	canceled	unavailable	processing	approved	invoiced	total_count_...
2017	4	530	43428	265	457	240	2	175	45101
2018	1	568	52783	334	145	59	0	121	54011
2016	0	9	267	26	7	2	0	18	329

Month to month analysis (for different order status) perct_change_deliered → this column describes the change in order count for order status =”delivered”

month	created	shipped	delivered	cancled	unavailable	processing	approved	invoiced	perct_change_diliverd
1	0	90	7819	37	58	38	0	27	null
2	1	78	8208	90	75	38	1	17	4.98
3	0	178	9549	59	49	32	32	26	16.34
4	0	148	9101	33	14	18	18	28	-4.69
5	0	109	10295	53	47	29	29	40	13.12
6	0	90	9234	34	28	12	0	14	-10.31
7	0	116	10031	69	70	12	0	20	8.63
8	0	88	10544	111	39	18	0	43	5.11
9	0	40	4151	37	38	22	0	17	-60.63
10	0	41	4743	54	65	22	0	34	14.26
11	2	72	7289	37	84	25	0	35	53.68
12	2	57	5514	11	42	35	0	13	-24.35

Month-month (for all years)

```
select *,
lag(no_of_orders,1) over(order by month) as prev_month_orders,
round((no_of_orders-
lag(no_of_orders,1) over(order by month)) /lag(no_of_orders,1) over(order by month)*100,2) as perct_chang
e
from(
select extract(month from order_purchase_timestamp) as month,count(order_id) as no_of_orders
from `oceanic-base-302317.Demo_target.Orders`
group by 1
order by 1)
order by 1
```

month	no_of_orders	prev_month...	perct_change
1	8069	null	null
2	8508	8069	5.44
3	9893	8508	16.28
4	9343	9893	-5.56
5	10573	9343	13.16
6	9412	10573	-10.98
7	10318	9412	9.63
8	10843	10318	5.09
9	4305	10843	-60.3
10	4959	4305	15.19
11	7544	4959	52.13
12	5674	7544	-24.79

Actionable Insights

- 11th month ie November we see raise in order compared to prev month.
- 12th month ie Dec we see a fall in orders .
- Over all 5,7,8 ie (may ,jul ,aug) month has highest orders.
-

Recommendations

- Since Nov,May,Jul ,Aug month is more likely to have more orders , please make sure that all things are going smooth
- After Nov (ie highest orders) → but the exact next month ie Dec we see fall in the orders (customer retention are not happening).
- Make sure that in the dec month customers get good deals .

Q2.1 What time do Brazilian customers tend to buy (Dawn, Morning, Afternoon or Night)?

```
select
sum(
case
when hour between 4 and 11
then no_of_orders
end )as Morning,
sum(
case
when hour between 12 and 15
then no_of_orders
end )as Afternoon,
sum(
case
when hour between  16 and 22
then no_of_orders
end )as Evening,
sum(
case
when hour =23 or hour between 0 and 3
then no_of_orders
end )as Night,

from (
select extract(hour from order_purchase_timestamp) as hour,count(order_id) as no_of_orders
from `oceanic-base-302317.Demo_target.Orders`
group by 1
order by 1)
```

Morning	Afternoon	Evening	Night
22634	25536	42802	8469

Actionable Insights

- Morning 4 – 11
- Afternoon 12-15
- Evening 16 -22
- Night 23 – 3
- Evening has highest orders ie from 16 hours to 22 hours

Recommendations

- Provide customers good deals on evening because they are most likely to buy .
- Increase the service at night.

Q1 Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset

Q1.1 Data type of columns in a table

- **Customers 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

- **Order Items Table**

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

- **Order Review Table**

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

- **Orders Table**

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

- **Payments Table**

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

- **Products Table**

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

- **Seller Table**

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

- **Geo-location Table**

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

Q1.2 Time period for which the data is given

```
SELECT min(order_purchase_timestamp) as start_date ,max(order_purchase_timestamp) end_date  FROM `oceanic
-base-302317.Demo_target.Orders`
```

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

(purchasetimestamp from order table is used)

Q1.3 Cities and States covered in the dataset

States :

```
SELECT distinct customer_state FROM `oceanic-base-302317.Demo_target.Customers`
```

There are total 27 states

1. RN
2. CE
3. RS
4. SC
5. SP
6. MG
7. BA
8. RJ
9. GO
10. MA
11. PE
12. PB
13. ES
14. PR
15. RO
16. MS
17. PA
18. TO
19. MT
20. PI
21. AL
22. AM
23. DF
24. SE
25. RR
26. AP
27. AC

Cities :

```
SELECT distinct customer_city FROM `oceanic-base-302317.Demo_target.Customers`
```

There are total 4119 cities

customer_city
acu
ico
ipe
ipu
ita
itu
jau
luz
poa