

---

## SQL CASE STUDY- Mayank Khanduja

---

### 1. Exploratory analysis

1. Data type
2. Time Period for which data is given is 2016 to 2018.  
The data is present from 04-09-2016 to 17-10-2018

```
select extract(year from (min(order_purchase_timestamp))) as year1 ,
extract(year from (max(order_purchase_timestamp))) as year2
from `aa.orders`
```

### 3. Cities and States of customers ordered during the given period

The case study captures data from 5812 distinct cities and 27 distinct states which means the data capture order transactions from all over Brazil.

```
select geolocation_city as city, geolocation_state as state
from `aa.geolocation` g
inner join `aa.customers` cc
on g.geolocation_zip_code_prefix = cc.customer_zip_code_prefix
```

### 2. In – depth Exploration

#### 1. Growing Trend of e-commerce in Brazil – Complete Scenario

Number of Orders have increased every year and percentage of delivery rate is very high

year1	no_of_orders_placed	no_of_orders_delivered	percentage
2016	329	267	81.16
2017	45101	43428	96.29
2018	54011	52783	97.73

```
select tmp1.year1 , no_of_orders_placed , no_of_orders_delivered ,
round((no_of_orders_delivered/no_of_orders_placed)*100,2) as percentage
from
(select extract(year from order_purchase_timestamp) as year1,
count(order_id) as no_of_orders_placed
from `aa.orders`
group by year1) as tmp1
inner join
(select extract(year from order_purchase_timestamp) as year1,
count(order_id) as no_of_orders_delivered
from `aa.orders`
where order_status= 'delivered'
group by year1) as tmp2
on tmp1.year1 = tmp2.year1
order by year1
```

Sales has been increased by more than 20% from 2017 to 2018  
(Since we have only last quarter data of 2016, we can look and compare data of 2017 with 2018)

year	total_sales
2016	59362.0
2017	7249747.0
2018	8699763.0

```
select year , total_sales,
from
(select
extract(year from order_purchase_timestamp) as year,
round(sum(payment_value)) as total_sales
from `aa.payments` p
inner join `aa.orders` o
on p.order_id = o.order_id
group by year
order by year desc) as tmp
order by year
```

## 2. Growing Trend of e-commerce in Brazil – Seasonality

Considering the whole data for a particular month, we observed August and September has maximum sales and March and November has most positive percentage change. March month is linked to Carnival which is one of the most important festivals of Brazil whereas November is associated with “Black Friday” which is an annual shopping event for Brazil.

Row	month	total_sales	prev	percentage_change
1	1	1253492.0	<i>null</i>	<i>null</i>
2	2	1284371.0	1253492.0	2.46
3	3	1609516.0	1284371.0	25.32
4	4	1578574.0	1609516.0	-1.92
5	5	1746901.0	1578574.0	10.66
6	6	1535157.0	1746901.0	-12.12
7	7	1658924.0	1535157.0	8.06
8	8	1696822.0	1658924.0	2.28
9	9	732454.0	1696822.0	-56.83
10	10	839358.0	732454.0	14.6
11	11	1194883.0	839358.0	42.36
12	12	878421.0	1194883.0	-26.48

```

select * ,
lag(total_sales) over(order by month) as prev,
round(((total_sales - lag(total_sales) over(order by month))/lag(total_sales) over(ord
er by month))* 100,2) as percentage_change
from
(select
#extract(year from order_purchase_timestamp) as year,
extract(month from order_purchase_timestamp) as month,
round(sum(payment_value)) as total_sales,
from `aa.payments` p
inner join `aa.orders` o
on p.order_id = o.order_id
group by month
order by month) as tmp
order by tmp.month

```

### 3. Brazilian Time preference of shopping

They prefer to order more in night i.e., between 6PM to 4AM

Time Bifurcation is given as follows:

Dawn	4 AM – 6 AM
Morning	6AM – 12 PM
Afternoon	12 PM – 6 PM
Night	6 PM – 4 AM

preference	purchase_count
night	38446
afternoon	38361
morning	22240
dawn	394

```

select preference , count(*) as purchase_count
from
(select order_purchase_timestamp,
case when extract(time from order_purchase_timestamp) between '04:00:00' and '05:59:59'
then 'dawn'
when extract(time from order_purchase_timestamp) between '06:00:00' and '11:59:59' the
n 'morning'
when extract(time from order_purchase_timestamp) between '12:00:00' and '17:59:59' the
n 'afternoon'
else 'night'
end as preference
from `aa.orders` )
group by preference
order by purchase_count desc

```

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

#### 1. Month by Month Order by States

*Considering Month and State*

SP has maximum orders followed by RJ and MG

The maximum order count is 4982 which belongs to SP state for the month of August.

State	Months												Total
	1	2	3	4	5	6	7	8	9	10	11	12	
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		2	4	2		46
RS	427	473	569	488	559	526	565	599	279	276	422	283	5466
SC	345	316	362	351	379	321	356	365	157	189	303	193	3637
SE	24	27	43	27	19	37	42	43	16	25	27	20	350
SP	3351	3357	4047	3967	4632	4104	4381	4982	1648	1908	3012	2357	41746
TO	19	28	28	33	34	26	23	28	17	13	17	14	280
Total	8069	8508	9893	9343	10573	9412	10318	10843	4305	4959	7544	5674	99441

```

select month, customer_state , count(*) as order_count
from
(select *,
extract(month from order_purchase_timestamp) as month
from `aa.orders` oo
join
`aa.customers` cc
on oo.customer_id = cc.customer_id) as tmp
group by customer_state , month
order by order_count desc

```

### Considering Year, Month and State

Taking year into consideration, SP has maximum orders in the month of August in 2018 i.e., 3253 orders.

The table represents the counts of orders per month per year for all the states.

	Year and Month																													
	2016			2017												2018													Total	
Months	9	10	12	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3	4	5	6	7	8	9	10					
AC				2	3	2	5	8	4	5	4	5	6	5	5	6	3	2	4	2	3	4	3					81		
AL		2		2	12	10	23	27	10	17	18	20	28	26	14	37	27	30	28	19	24	23	16					413		
AM					8	5	13	10	1	5	5	9	3	10	6	12	8	9	6	9	7	18	4					148		
AP					2	3		5	2	1	3	2	3	4	4	11	2	5	5	6	2	6	2					68		
BA		4		25	59	91	93	127	106	155	158	170	166	250	192	239	214	249	225	241	201	250	165					3380		
CE		8		9	13	28	43	62	47	53	73	77	66	108	81	90	88	98	100	74	74	87	57					1336		
DF		6		13	24	57	35	64	70	77	87	97	98	168	131	138	172	150	148	144	150	166	145					2140		
ES		4		12	34	48	46	94	80	83	95	93	100	170	113	147	152	134	142	134	124	123	105					2033		
GO		9		18	27	53	41	87	79	77	93	88	108	157	127	146	149	146	136	139	105	115	120					2020		
MA		4		9	11	24	27	33	17	39	40	42	48	56	41	57	56	53	46	32	42	40	30					747		
MG		40		108	259	358	275	428	363	453	469	507	560	943	691	863	804	879	786	762	717	658	708	4				11635		
MS				1	11	20	15	29	27	25	24	33	34	46	36	70	64	59	43	45	49	49	35					715		
MT		3		11	17	16	27	37	25	38	38	35	52	74	50	85	67	55	65	67	58	47	40					907		
PA		4		12	25	36	36	35	38	39	60	41	54	70	58	70	58	73	71	40	54	57	44					975		
PB		1		2	12	16	20	18	23	27	16	29	30	30	37	31	35	39	31	29	28	52	30					536		
PE		7		9	21	45	40	68	46	73	85	76	80	126	103	104	125	108	114	106	94	137	85					1652		
PI		1		7	12	13	13	25	14	20	22	23	23	31	23	48	34	35	37	31	29	32	21		1			495		
PR		19	1	65	118	127	114	213	170	203	223	183	206	378	270	378	342	377	386	311	308	320	333					5045		
RJ		56		97	254	395	338	488	412	571	562	609	668	1048	783	893	922	907	834	833	716	717	745	3	1			12852		
RN		4		5	8	13	10	17	13	27	20	24	23	44	30	46	23	39	32	22	36	29	20					485		
RO				3	11	16	9	9	10	11	14	16	14	17	11	20	14	13	11	17	12	16	9					253		
RR	1	1			2	2	2	2	3	1		1	3	2		2	5	6	2	1	5	5						46		
RS	1	24		54	105	151	139	208	221	249	299	278	252	422	283	373	368	418	349	351	305	316	300					5466		
SC		11		31	59	110	105	152	116	158	159	156	178	303	193	314	257	252	246	227	205	198	206	1				3637		
SE		3		4	12	25	13	11	9	14	20	16	22	27	20	20	15	18	14	8	28	28	23					350		
SP	2	113		299	654	1010	908	1425	1331	1604	1729	1638	1793	3012	2357	3052	2703	3037	3059	3207	2773	2777	3253	8	2			41746		
TO				2	7	8	14	18	8	1	15	17	13	17	14	17	21	20	19	16	18	22	13					280		
Total	4	324	1	800	1780	2682	2404	3700	3245	4026	4331	4285	4631	7544	5673	7269	6728	7211	6939	6873	6167	6292	6512	16	4			99441		

```

select year , month, customer_state , count(*) as order_count
from
(select *,
extract(month from order_purchase_timestamp) as month,
extract(year from order_purchase_timestamp) as year
from `aa.orders` oo
join
`aa.customers` cc
on oo.customer_id = cc.customer_id) as tmp
group by customer_state , year , month
order by order_count desc

```

## 2. Distribution of customers across states in Brazil

SP and RJ are among the top state while AP and RR are among the lowest state in terms of usability.

customer_state	costumer_count	percentage
SP	41746	41.98
RJ	12852	12.92
MG	11635	11.7
RS	5466	5.5
PR	5045	5.07
SC	3637	3.66
BA	3380	3.4
DF	2140	2.15
ES	2033	2.04
GO	2020	2.03
PE	1652	1.66
CE	1336	1.34
PA	975	0.98
MT	907	0.91
MA	747	0.75
MS	715	0.72
PB	536	0.54
PI	495	0.5
RN	485	0.49
AL	413	0.42
SE	350	0.35
TO	280	0.28
RO	253	0.25
AM	148	0.15
AC	81	0.08
AP	68	0.07
RR	46	0.05

```
select
customer_state , count(*) as costumer_count,
round((count(*)/(select count(*) from `aa.customers`))*100,2) as percentage
from `aa.customers`
group by customer_state
order by percentage desc
```

#### 4. Impact on Economy

##### 1. % increase in cost of orders from 2017 to 2018

The order cost has been increased by 137% (Considering all filters given in the question)

Calculation and Code is shown below:

Total payment made in the year

year	sum
2017	3669022.12
2018	8694733.84

```
select year, round(sum(payment_value),2) as sum
from
(select payment_value , order_status ,customer_id , p.order_id,
extract(month from order_purchase_timestamp) as month,
extract(year from order_purchase_timestamp) as year,
from `aa.orders` o
inner join `aa.payments` p
on o.order_id = p.order_id
where (extract(year from order_purchase_timestamp)= 2017 or extract(year from order_purchase_timestamp)= 2018)
and extract(month from order_purchase_timestamp) between 1 and 8 )
group by year
order by year
```

Percentage change Calculation

year	sum	sum_2018	percentage
2017	3669022.12	8694733.84	136.98
2018	8694733.84	null	null

```
select year , sum ,
lead(sum) over(order by sum) as sum_2018,
round((lead(sum) over(order by sum) - sum )/sum * 100 ,2) as percentage
from
(select year, round(sum(payment_value),2) as sum
from
(select payment_value , order_status ,customer_id , p.order_id,
extract(month from order_purchase_timestamp) as month,
extract(year from order_purchase_timestamp) as year,
from `aa.orders` o
inner join `aa.payments` p
on o.order_id = p.order_id
where (extract(year from order_purchase_timestamp)= 2017 or extract(year from order_purchase_timestamp)= 2018)
and extract(month from order_purchase_timestamp) between 1 and 8 )
group by year
)
order by year
```

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

The sum of price is highest is SP and RJ since they have maximum number of orders if compared to other countries and there sum of freight cost is higher too because of same reason.

The mean price is higher in AL and PB whereas mean freight value is highest in PB and RR

SP which has maximum orders has minimum mean freight value (15) and minimum mean price too (100) which means products are quite affordable there.

customer_state	sum_price	sum_freight	mean_price	mean_freight
AC	15983	3687	174	40
AL	80315	15915	181	36
AM	22357	5479	135	33
AP	13474	2789	164	34
BA	511350	100157	135	26
CE	227255	48352	154	33
DF	302604	50625	126	21
ES	275037	49765	122	22
GO	294592	53115	126	23
MA	119648	31524	145	38
MG	1585308	270853	121	21
MS	116813	19144	143	23
MT	156454	29715	148	28
PA	178948	38699	166	36
PB	115268	25720	191	43
PE	262788	59450	146	33
PI	86914	21218	160	39
PR	683084	117852	119	21
RJ	1824093	305589	125	21
RN	83035	18860	157	36
RO	46141	11417	166	41
RR	7829	2235	151	43
RS	750304	135523	120	22
SC	520553	89660	125	21
SE	58921	14111	153	37
SP	5202955	718723	110	15
TO	49622	11733	158	37

```
select customer_state,
round(sum(price)) as sum_price,
round(sum(freight_value)) as sum_freight,
round(avg(price)) as mean_price,
round(avg(freight_value)) as mean_freight
from `aa.customers` c
inner join `aa.orders` o
on c.customer_id = o.customer_id
inner join `aa.order_items` i
on o.order_id = i.order_id
group by customer_state
order by customer_state
```



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

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

We consider 3 dates here:

1. order\_purchase\_timestamp - Timestamp of the purchase
2. order\_estimated\_delivery\_date - estimated delivery date of the products
3. order\_delivered\_customer\_date - date at which customer got the product.

The Max difference between purchase date and delivery date is 209 days.

This is an automotive type of a product ordered from montanha (ES) and seller belongs to belo horizonte (MG)

```
select product_category , seller_city , seller_state , customer_city , customer_state
from `aa.orders` o
inner join `aa.customers` c
on o.customer_id = c.customer_id
inner join `aa.order_items` i
on o.order_id = i.order_id
inner join `aa.products` p
on i.product_id = p.product_id
inner join `aa.sellers` s
on i.seller_id = s.seller_id
where
timestamp_diff(order_delivered_customer_date,order_purchase_timestamp,day)= 209
```

The Min difference between purchase date and delivery date is 0 days i.e., order is delivered on the same day.

The Max difference between purchase date and estimated delivery date is 155 days which was a furniture decoration product where seller belongs to sao paulo and customer belongs to sao Gabriel.

The Min difference between purchase date and estimated delivery date is 2 days

```
Select
max(timestamp_diff(order_estimated_delivery_date,order_purchase_timestamp,day)) as max
1,
max(timestamp_diff(order_delivered_customer_date,order_purchase_timestamp,day)) as max
2,
min(timestamp_diff(order_estimated_delivery_date,order_purchase_timestamp,day)) as min
1,
min(timestamp_diff(order_delivered_customer_date,order_purchase_timestamp,day)) as min
2
from `aa.orders`
where order_delivered_customer_date is not null
```

## 2. Time\_to\_delivery & diff\_estimated\_delivery

Time to Delivery = order\_delivered\_customer\_date-order\_purchase\_timestamp

➤ Already covered above.

Diff\_estimated\_delivery:

(order\_estimated\_delivery\_date) - (order\_delivered\_customer\_date)

- This estimates the time difference between order estimated delivery date and order delivery date.
- A positive difference indicates that the order is delivered before time and a negative difference indicates that the order is late and is not delivered within estimated time.
- Approx 93% of orders were delivered before time and only approx. 7% orders got delayed.

positive_count //	neagative_count //	total_count //	percent_positive //	percent_negativ //
89941	6535	96476	93.23	6.77

```
select
count(case when diff1 >= 0 then 1 end) as positive_count,
count(case when diff1 < 0 then 1 end) as neagative_count,
count(*) as total_count,
round((count(case when diff1 >= 0 then 1 end)/count(*)* 100,2) as percent_positive,
round((count(case when diff1 < 0 then 1 end)/count(*)* 100,2) as percent_negative
from
(select
order_estimated_delivery_date , order_delivered_customer_date,
timestamp_diff(order_estimated_delivery_date,order_delivered_customer_date,day) as diff1
from `aa.orders`)
where diff1 is not null
```

- The maximum difference is observed to be +146 and minimum difference is observed to be -188.
- The details for max and min are given below:  
Min

product_category	order_estimated_delivery_date	order_delivered_customer_date	seller_city	customer_city
Cool Stuff	2018-03-15 00:00:00.000000 UTC	2018-09-19 23:24:07.000000 UTC	sao paulo	rio de janeiro

Max

product_category	order_estimated_delivery_date	order_delivered_customer_date	seller_city	customer_city
Furniture Decoration	2018-08-03 00:00:00.000000 UTC	2018-03-09 23:36:47.000000 UTC	sao paulo	sumare
Furniture Decoration	2018-08-03 00:00:00.000000 UTC	2018-03-09 23:36:47.000000 UTC	sao paulo	sumare

```

# finding max and min values
select
max(timestamp_diff(order_estimated_delivery_date,order_delivered_customer_date,day)) as max1,
min(timestamp_diff(order_estimated_delivery_date,order_delivered_customer_date,day)) as min1,
from `aa.orders`
where order_delivered_customer_date is not null

#product level info for such values
select product_category , order_estimated_delivery_date , order_delivered_customer_date , seller_city, customer_city
from `aa.orders` o
inner join `aa.customers` c
on o.customer_id = c.customer_id
inner join `aa.order_items` i
on o.order_id = i.order_id
inner join `aa.products` p
on i.product_id = p.product_id
inner join `aa.sellers` s
on i.seller_id = s.seller_id
where
timestamp_diff(order_estimated_delivery_date,order_delivered_customer_date,day) = 146

```

- Group data by state, take mean of freight\_value, time\_to\_delivery, diff\_estimated\_delivery

customer_state	avg_freight	time_to_delivery	diff_estimated_delivery
RJ	20.91	14.69	11.14
MG	20.63	11.52	12.4
SC	21.51	14.52	10.67
SP	15.11	8.26	10.27
GO	22.56	14.95	11.37
RS	21.61	14.71	13.2
BA	26.49	18.77	10.12
MT	28	17.51	13.64
SE	36.57	20.98	9.17
PE	32.69	17.79	12.55
TO	37.44	17	11.46
CE	32.73	20.54	10.26
PR	20.47	11.48	12.53
PA	35.63	23.3	13.37
MS	23.35	15.11	10.34
ES	22.03	15.19	9.77
RN	35.72	18.87	13.06
AL	35.87	23.99	7.98
RO	41.33	19.28	19.08
AM	33.31	25.96	18.98
MA	38.49	21.2	9.11
DF	21.07	12.5	11.27
PI	39.12	18.93	10.68
PB	43.09	20.12	12.15
AP	34.16	27.75	17.44
AC	40.05	20.33	20.01
RR	43.09	27.83	17.43

- SP has the minimum average freight value while PB and RR has maximum.
- Average time\_to\_delivery is highest in RR while minimum in SP
- Average diff\_estimated\_delivery is highest in AC and minimum in AL

```
select customer_state,
round(avg(freight_value),2) as avg_freight,
round(avg(timestamp_diff(order_delivered_customer_date,order_purchase_timestamp,day)),
2) as time_to_delivery,
round(avg(timestamp_diff(order_estimated_delivery_date,order_delivered_customer_date,d
ay)),2) as diff_estimated_delivery
from `aa.orders` o
inner join
`aa.customers` c
on o.customer_id = c.customer_id
inner join
`aa.order_items` oi
on oi.order_id = o.order_id
where order_delivered_customer_date is not null
group by customer_state
```

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

##### Top 5 states with Highest Average freight value

customer_state	avg_freight
RR	42.98
PB	42.72
RO	41.07
AC	40.07
PI	39.15

```
select customer_state,
round(avg(freight_value),2) as avg_freight
from `aa.orders` o
inner join
`aa.customers` c
on o.customer_id = c.customer_id
inner join
`aa.order_items` oi
on oi.order_id = o.order_id
group by customer_state
order by avg_freight desc limit 5
```

## Top 5 states with lowest Average freight value

customer_state	avg_freight
SP	15.15
PR	20.53
MG	20.63
RJ	20.96
DF	21.04

```
select customer_state,
round(avg(freight_value),2) as avg_freight
from `aa.orders` o
inner join
`aa.customers` c
on o.customer_id = c.customer_id
inner join
`aa.order_items` oi
on oi.order_id = o.order_id
group by customer_state
order by avg_freight limit 5
```

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

customer_state	time_to_delivery
SP	8.26
PR	11.48
MG	11.52
DF	12.5
SC	14.52

```
select customer_state,
round(avg(timestamp_diff(order_delivered_customer_date,order_purchase_timestamp,day)),
2) as time_to_delivery,
from `aa.orders` o
inner join
`aa.customers` c
on o.customer_id = c.customer_id
inner join
`aa.order_items` oi
on oi.order_id = o.order_id
where order_delivered_customer_date is not null
group by customer_state
order by time_to_delivery limit 5
```

## Top 5 states with high average time to delivery

customer_state	time_to_delivery
RR	27.83
AP	27.75
AM	25.96
AL	23.99
PA	23.3

```
select customer_state,
round(avg(timestamp_diff(order_delivered_customer_date,order_purchase_timestamp,day)),
2) as time_to_delivery,
from `aa.orders` o
inner join
`aa.customers` c
on o.customer_id = c.customer_id
inner join
`aa.order_items` oi
on oi.order_id = o.order_id
where order_delivered_customer_date is not null
group by customer_state
order by time_to_delivery desc limit 5
```

6. Top 5 states where delivery is really fast/ not so fast compared to estimated date  
Comparing the overall data grouped by state and compare

## Top 5 states where delivery is FAST.

customer_state	diff_estimated_c
AL	7.98
MA	9.11
SE	9.17
ES	9.77
BA	10.12

## Top 5 states where delivery is NOT FAST

customer_state	diff_estimated_c
AC	20.01
RO	19.08
AM	18.98
AP	17.44
RR	17.43

```

# ASC
select customer_state,
round(avg(timestamp_diff(order_estimated_delivery_date,order_delivered_customer_date,d
ay)),2) as diff_estimated_delivery
from `aa.orders` o
inner join
`aa.customers` c
on o.customer_id = c.customer_id
inner join
`aa.order_items` oi
on oi.order_id = o.order_id
where order_delivered_customer_date is not null
group by customer_state
order by diff_estimated_delivery limit 5

# DESC
select customer_state,
round(avg(timestamp_diff(order_estimated_delivery_date,order_delivered_customer_date,d
ay)),2) as diff_estimated_delivery
from `aa.orders` o
inner join
`aa.customers` c
on o.customer_id = c.customer_id
inner join
`aa.order_items` oi
on oi.order_id = o.order_id
where order_delivered_customer_date is not null
group by customer_state
order by diff_estimated_delivery desc limit 5

```

## 6. Payment type Analysis

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

Credit card has highest usage followed by UPI.

```

select extract(month from o.order_purchase_timestamp) as month,
p.payment_type , count(*) as order_count
from `aa.orders` o
inner join `aa.payments` p
on o.order_id = p.order_id
group by month,payment_type
order by month,payment_type

```

month	payment_type	order_count
1	UPI	1715
1	credit_card	6103
1	debit_card	118
1	voucher	477
2	UPI	1723
2	credit_card	6609
2	debit_card	82
2	voucher	424
3	UPI	1942
3	credit_card	7707
3	debit_card	109
3	voucher	591
4	UPI	1783
4	credit_card	7301
4	debit_card	124
4	voucher	572
5	UPI	2035
5	credit_card	8350
5	debit_card	81
5	voucher	613
6	UPI	1807
6	credit_card	7276
6	debit_card	209
6	voucher	563
7	UPI	2074
7	credit_card	7841
7	debit_card	264
7	voucher	645
8	UPI	2077
8	credit_card	8269
8	debit_card	311
8	not_defined	2
8	voucher	589
9	UPI	903
9	credit_card	3286
9	debit_card	43
9	not_defined	1
9	voucher	302
10	UPI	1056
10	credit_card	3778
10	debit_card	54
10	voucher	318
11	UPI	1509
11	credit_card	5897
11	debit_card	70
11	voucher	387
12	UPI	1160
12	credit_card	4378
12	debit_card	64
12	voucher	294

- Count of orders based on the no. of payment installments.  
Most customers prefer to pay the amount in 1 or 2 installments.

```

select
p.payment_installments , count(*) as no_of_orders
from `aa.orders` o
inner join `aa.payments` p
on o.order_id = p.order_id
group by payment_installments
order by payment_installments

```



payment_in	no_of_order
0	2
1	52546
2	12413
3	10461
4	7098
5	5239
6	3920
7	1626
8	4268
9	644
10	5328
11	23
12	133
13	16
14	15
15	74
16	5
17	8
18	27
20	17
21	3
22	1
23	1
24	18

## 7. Insights & Recommendations

### 1. Average Order value per product category

Top 5 product categories which are driving high value orders are shown below. A more focus on these products can be helpful in generating more sales.

product_category	AOV
PCs	1542.11
fixed telephony	953.96
HOUSE PASTALS OVEN AND CAFE	683.06
Agro Industria e Comercio	652.37
ELECTRICES 2	532.32

Calculation:

```

select product_category , round(sum(payment_value)/count(distinct p.order_id),2) as AOV
V
from `aa.orders` o
inner join `aa.payments` p
on o.order_id = p.order_id
inner join `aa.order_items` oi
on o.order_id = oi.order_id
inner join `aa.products` pr
on oi.product_id = pr.product_id
group by product_category
order by AOV desc

```

## 2. Most and least bought products

Least bought products - Target can increase the advertisement of these products in targeted cities.

product_category	order_count
insurance and services	2
Fashion Children's Clothing	8
PC Gamer	10
cds music dvds	14
Kitchen portable and food coach	15

### Most Bought products –

product_category	order_count
bed table bath	11823
HEALTH BEAUTY	9972
sport leisure	8945
Furniture Decoration	8744
computer accessories	8082

Calculation:

```

select product_category , count(o.order_id) as order_count
from `aa.orders` o
inner join `aa.payments` p
on o.order_id = p.order_id
inner join `aa.order_items` oi
on o.order_id = oi.order_id
inner join `aa.products` pr
on oi.product_id = pr.product_id
group by product_category
order by order_count desc limit 5

```

### 3. Net Promoter Score (NPS)

The data achieved a Net promoter score of 62.38 which means majority of products have positive perception and have capability to increase business by recommending it to others.

Calculation:

```
select  
round((count(case when review_score = 4 or review_score= 5 then 1 end)/ count(*))*100  
- (count(case when review_score = 1 or review_score= 2 then 1 end)/ count(*))*100,2) as  
nps_score  
from `aa.order_reviews`
```