Q.1 PART-1 Data type of all columns in the "customers" table

```
SELECT COLUMN_NAME, DATA_TYPE
FROM target.INFORMATION_SCHEMA.COLUMNS
WHERE TABLE_NAME = 'customers'
```

```
ANSWER :
Row  COLUMN_NAME                DATA_TYPE
 1   customer_id                STRING
 2   customer_unique_id         STRING
 3   customer_zip_code_prefix   INT64
 4   customer_city              STRING
 5   customer_state             STRING
```

Row	COLUMN_NAME	DATA_TYPE
1	customer_id	STRING
2	customer_unique_id	STRING
3	customer_zip_code_prefix	INT64
4	customer_city	STRING
5	customer_state	STRING

Q.1 PART-2 Time range between which the orders were placed

```
SELECT MIN(order_purchase_timestamp) AS FIRST_ORDER_DATE,
MAX(order_purchase_timestamp) AS LATEST_ORDER_DATE
FROM `target.orders`
```

```
ANSWER : 2016-09-04 21:15:19 UTC to 2018-10-17 17:30:18 UTC
```

Q.1 PART-3 Count the number of Cities and States in our dataset

```
SELECT COUNT(DISTINCT customer_state) AS NUMBER_OF_STATES,
       COUNT(DISTINCT customer_city) AS NUMBER_OF_CITIES
FROM `target.customers`
```

```
ANSWER :
Row  NUMBER_OF_STATES     NUMBER_OF_CITIES
 1   27                   4119
```

Row	NUMBER_OF_STATES	NUMBER_OF_CITIES
1	27	4119

Q.2 PART-1 Is there a growing trend in the number of orders placed over the past years?

```
SELECT EXTRACT(YEAR FROM order_purchase_timestamp) AS YEAR,
       COUNT(*) AS NO_OF_ORDERS_PLACED
FROM `target.orders`

GROUP BY year

ORDER BY year
```

```
 ANSWERS :
 Row YEAR    NO_OF_ORDERS_PLACED
 1   2016           329
 2   2017          45101
 3   2018          54011
```

Row	YEAR	NO_OF_ORDERS_PLACED
1	2016	329
2	2017	45101
3	2018	54011

Yes. There is an uptrend in the number of orders placed over the past years.

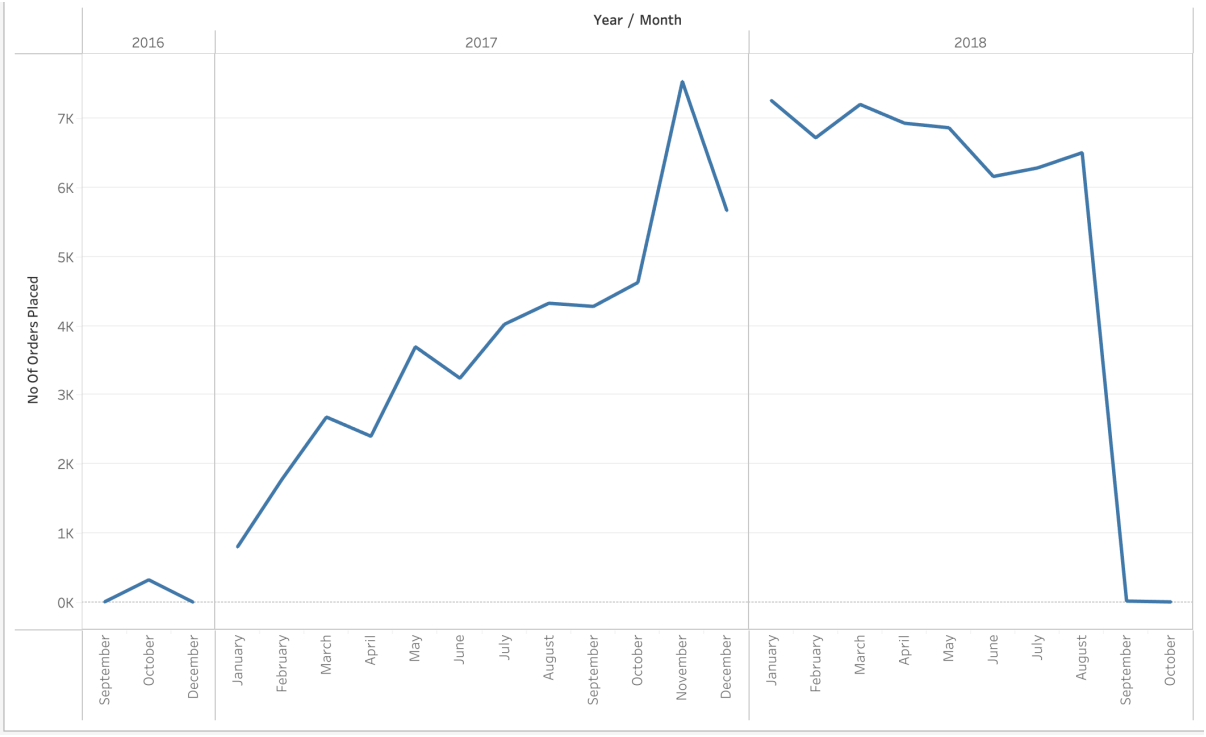
Q.2 PART-2 Can we see some kind of monthly seasonality in terms of the number of orders being placed?

```
SELECT EXTRACT(YEAR FROM order_purchase_timestamp) AS YEAR,
       format_datetime('%B', order_purchase_timestamp) AS MONTH,
       COUNT(*) AS NO_OF_ORDERS_PLACED
FROM `target.orders`

GROUP BY YEAR, MONTH
```

```
ORDER BY YEAR, MONTH
```

ANSWER :



Yes. We can conclude that in 2017, number of orders placed was on uptrend with a peak in November.

Q.2 PART-3 During what time of the day, do the Brazilian customers mostly place their orders?

```
SELECT CASE WHEN EXTRACT(HOUR FROM order_purchase_timestamp) >= 0 AND EXTRACT(HOUR
FROM order_purchase_timestamp) < 7 THEN 'Dawn'
            WHEN EXTRACT(HOUR FROM order_purchase_timestamp) >= 7 AND EXTRACT(HOUR
FROM order_purchase_timestamp) < 13 THEN 'Mornings'
            WHEN EXTRACT(HOUR FROM order_purchase_timestamp) >= 13 AND EXTRACT(HOUR
FROM order_purchase_timestamp) < 19 THEN 'Afternoon'
            WHEN EXTRACT(HOUR FROM order_purchase_timestamp) >= 19 AND EXTRACT(HOUR
FROM order_purchase_timestamp) <= 23 THEN 'Night'
       END AS time_of_day,
       COUNT(*) AS ORDER_COUNT
FROM `target.orders`

GROUP BY time_of_day

ORDER BY ORDER_COUNT
```

ANSWER :

Row	time_of_day	ORDER_COUNT
1	Dawn	5242
2	Mornings	27733
3	Night	28331
4	Afternoon	38135

Brazilian customers place most of their orders in the AFTERNOON.

Q.3 PART-1 Get the month on month number of orders placed in each state.

```
WITH CTE AS
(
       SELECT EXTRACT(MONTH FROM o.order_purchase_timestamp) AS MONTH,
              c.customer_state, COUNT(o.order_id) AS MONTHLY_ORDERS
       FROM `target.customers` c
       INNER JOIN `target.orders` o

       ON c.customer_id = o.customer_id
       GROUP BY 1,2

)

SELECT *

FROM CTE

ORDER BY 1,3
```

ANSWER :

Row Labe	AC	AL	AM	AP	BA	CE	DF	ES	GO	MA	MG	MS	MT	PA	PB	PE	PI	PR	RJ	RN	RO	RR	RS	SC	SE	SP	TO	Grand Total
1	8	39	12	11	264	99	151	159	164	66	971	71	96	82	33	113	55	443	990	51	23	2	427	345	24	3351	19	8069
2	6	39	16	4	273	101	196	186	176	67	1063	75	84	83	47	146	46	460	1176	31	25	7	473	316	27	3357	28	8508
3	4	40	14	8	340	126	207	182	199	77	1237	79	71	109	55	153	48	504	1302	52	29	8	569	362	43	4047	28	9893
4	9	51	19	5	318	143	183	188	177	73	1061	58	92	107	51	154	50	500	1172	42	20	4	488	351	27	3967	33	9343
5	10	46	19	11	368	136	208	228	226	65	1190	74	104	75	47	174	56	524	1321	39	26	3	559	379	19	4632	34	10573
6	7	34	8	4	307	121	220	204	184	59	1080	76	83	92	51	140	43	478	1128	49	22	8	526	321	37	4104	26	9412
7	9	40	23	7	405	140	243	206	192	79	1111	74	85	96	79	210	52	523	1288	56	27	6	565	356	42	4381	23	10318
8	7	34	9	5	323	130	232	200	213	70	1177	59	78	104	46	170	43	556	1307	40	23		599	365	43	4982	28	10843
9	5	20	9	2	170	77	97	93	88	42	511	33	35	41	29	76	23	183	612	24	16	2	279	157	16	1648	17	4305
10	6	30	3	3	170	74	104	104	117	52	600	34	55	58	31	87	25	225	725	27	14	4	276	189	25	1908	13	4959
11	5	26	10	4	250	108	168	170	157	56	943	46	74	70	30	126	31	378	1048	44	17	2	422	303	27	3012	17	7544
12	5	14	6	4	192	81	131	113	127	41	691	36	50	58	37	103	23	271	783	30	11		283	193	20	2357	14	5674
Grand Total	81	413	148	68	3380	1336	2140	2033	2020	747	11635	715	907	975	536	1652	495	5045	12852	485	253	46	5466	3637	350	41746	280	99441

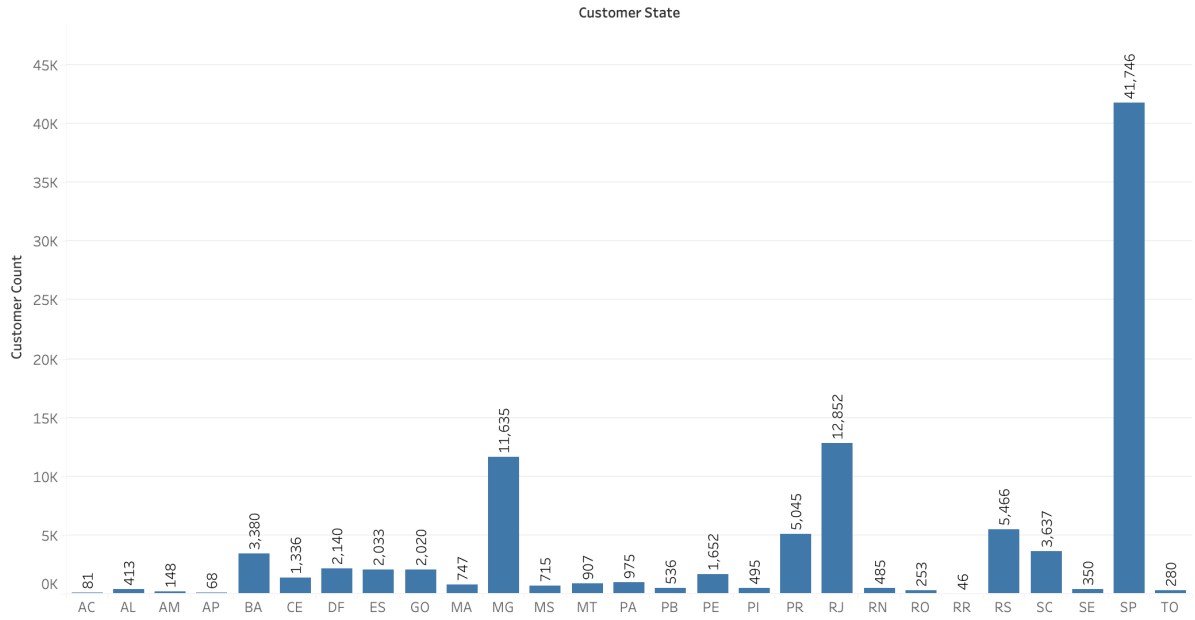
Q.3 PART-2 How are the customers distributed across all the states?

```
SELECT customer_state, COUNT(customer_id) AS customer_count

FROM `target.customers`

GROUP BY customer_state

ORDER BY customer_count DESC
```



Q.4 PART-1 Get the % increase in the cost of orders from year 2017 to 2018 (include months between Jan to Aug only).

```
WITH CTE AS
(
        SELECT ROUND(SUM(p.payment_value),2) AS YEARLY_COST,
               EXTRACT(YEAR FROM o.order_purchase_timestamp) AS YEAR

        FROM `target.payments` p INNER JOIN `target.orders` o
        ON p.order_id = o.order_id

        WHERE (o.order_purchase_timestamp >= '2017-01-01' AND
               o.order_purchase_timestamp <= '2017-08-31')
                OR
               (o.order_purchase_timestamp >= '2018-01-01' AND
               o.order_purchase_timestamp <= '2018-08-31')
```

```
        GROUP BY 2
        ORDER BY 2
)

SELECT CTE.YEAR,
        CTE.YEARLY_COST AS CURRENT_YEAR_COST,
        LAG(CTE.YEARLY_COST) OVER(ORDER BY CTE.YEAR) AS PREVIOUS_YEAR_COST,
        ROUND((CTE.YEARLY_COST - (LAG(CTE.YEARLY_COST) OVER(ORDER BY
CTE.YEAR)))/(LAG(CTE.YEARLY_COST) OVER(ORDER BY CTE.YEAR))*100,2) AS PERC_INC

FROM CTE

ORDER BY 1 DESC
LIMIT 1
```

ANSWER : 138.53% INCREASE

Row	YEAR	CURRENT_YEAR_COST	PREVIOUS_YEAR_COST	PERC_INC
1	2018	8694669.95	3645107.27	138.53

Q.4 PART-2 Calculate the Total & Average value of order price for each state.

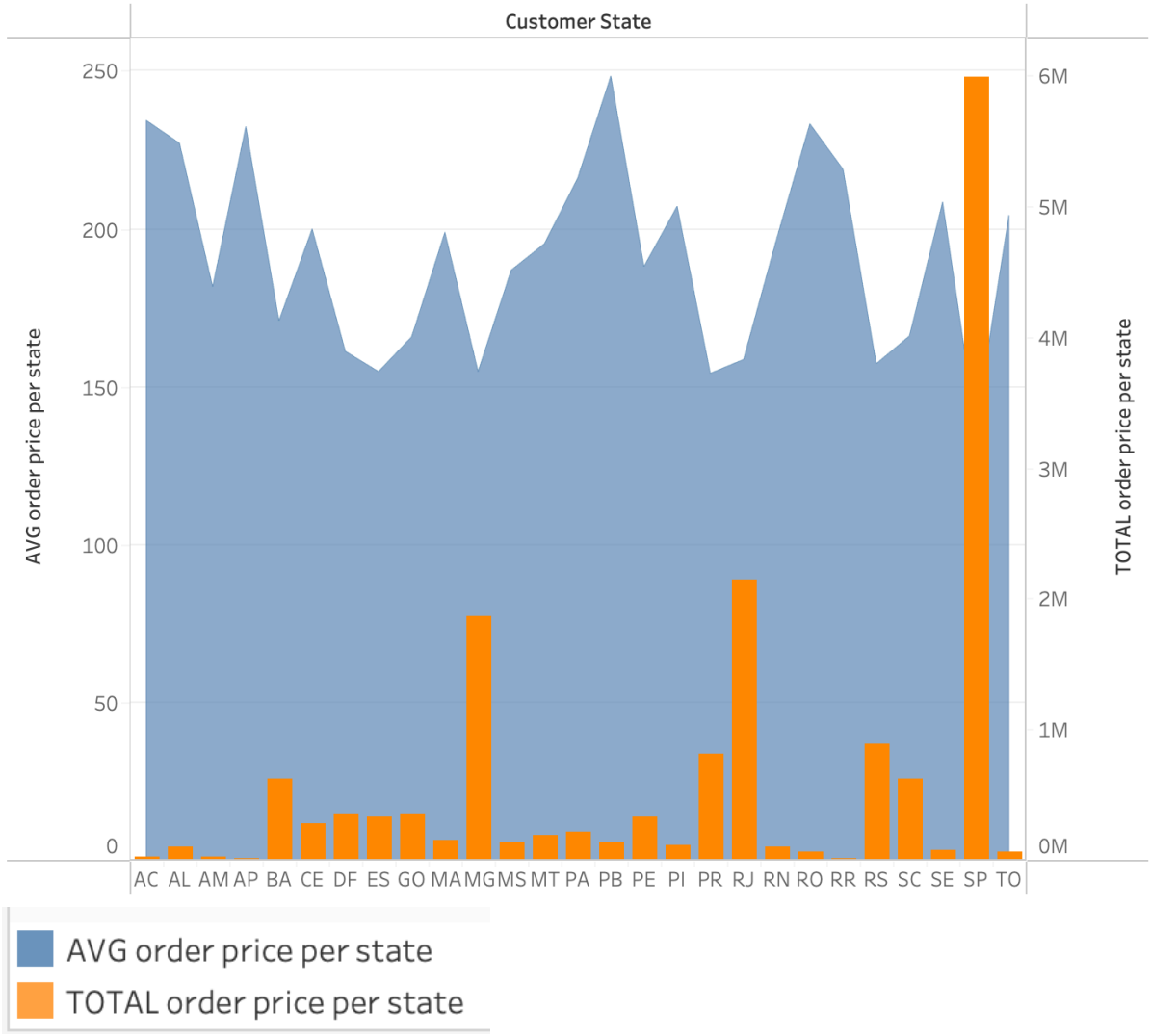
```
SELECT c.customer_state, round(SUM(p.payment_value),2) AS
TOTAL_order_price_per_state,
        round(AVG(p.payment_value),2) AS AVG_order_price_per_state

FROM `target.customers` c
INNER JOIN `target.orders` o

ON c.customer_id = o.customer_id

INNER JOIN `target.payments` p
ON p.order_id = o.order_id

GROUP BY 1
ORDER BY 1
```



Q.4 PART-3 Calculate the Total & Average value of order freight for each state.

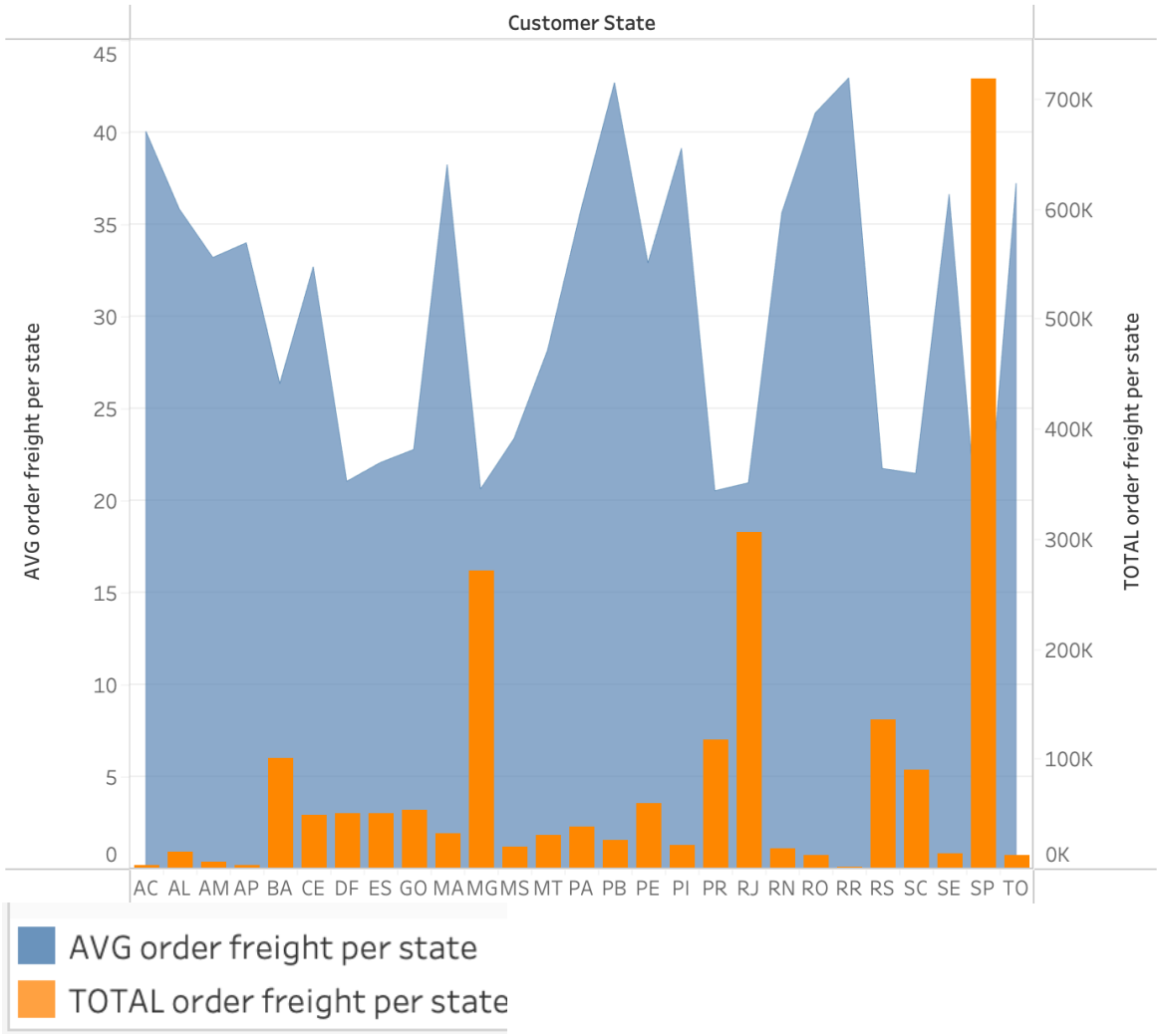
```sql
SELECT c.customer_state, round(SUM(oi.freight_value),2) AS
TOTAL_order_freight_per_state,
       round(AVG(oi.freight_value),2) AS AVG_order_freight_per_state

FROM `target.customers` c
INNER JOIN `target.orders` o

ON c.customer_id = o.customer_id

INNER JOIN `target.order_items` oi
ON oi.order_id = o.order_id

GROUP BY 1
ORDER BY 1
```



Q.5 PART-1 Find the no. of days taken to deliver each order from the order's purchase date as delivery time. Also, calculate the difference (in days) between the estimated & actual delivery date of an order.

```
SELECT order_id,
       DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY) AS
time_to_deliver,
       DATE_DIFF(order_estimated_delivery_date, order_delivered_customer_date, DAY)
AS diff_estimated_delivery

FROM `target.orders`

WHERE (order_delivered_customer_date IS NOT NULL) AND (order_purchase_timestamp IS
NOT NULL)


ORDER BY 2, 3
```

Row	order_id	time_to_deliver	diff_estimated_deliv
1	d5fbeedc85190ba88580d6f82...	0	7
2	79e324907160caea526fd8b94...	0	8
3	e65f1eeee1f52024ad1dcd034...	0	9
4	b70a8d75313560b4acf607739...	0	9
5	1d893dd7ca5f77ebf5f59f0d20...	0	10
6	d3ca7b82c922817b06e5ca211...	0	11
7	f3c6775ba3d2d9fe2826f93b71...	0	11
8	21a8ffca665bc7a1087d31751...	0	11
9	f349cdb62f69c3fae5c4d7d3f3...	0	12
10	38c1e3d4ed6a13cd0cf612d4c...	0	16

Results per page: 50 1 – 50 of 96476

Q.5 PART-2 Find out the top 5 states with the highest & lowest average freight value.

```
(SELECT c.customer_state, round(AVG(oi.freight_value),2) AS
AVG_order_freight_per_state

FROM `target.customers` c
INNER JOIN `target.orders` o

ON c.customer_id = o.customer_id

INNER JOIN `target.order_items` oi
ON oi.order_id = o.order_id

GROUP BY 1
ORDER BY 2 DESC
LIMIT 5)

UNION ALL

(SELECT c.customer_state, round(AVG(oi.freight_value),2) AS
AVG_order_freight_per_state

FROM `target.customers` c
INNER JOIN `target.orders` o

ON c.customer_id = o.customer_id

INNER JOIN `target.order_items` oi
ON oi.order_id = o.order_id

GROUP BY 1
ORDER BY 2 ASC
LIMIT 5)
```

Row	customer_state	AVG_order_freight_per_state
1	RR	42.98
2	PB	42.72
3	RO	41.07
4	AC	40.07
5	PI	39.15
6	SP	15.15
7	PR	20.53
8	MG	20.63
9	RJ	20.96
10	DF	21.04

HIGHEST

LOWEST

Q.5 PART-3 Find out the top 5 states with the highest & lowest average delivery time

```
(SELECT c.customer_state, ROUND(AVG(DATE_DIFF(order_delivered_customer_date,
order_purchase_timestamp, DAY)),2) AS AVG_delivery_time_IN_DAYS

FROM `target.customers` c
INNER JOIN `target.orders` o

ON c.customer_id = o.customer_id


GROUP BY 1
ORDER BY 2 DESC
LIMIT 5)


UNION ALL


(SELECT c.customer_state, ROUND(AVG(DATE_DIFF(order_delivered_customer_date,
order_purchase_timestamp, DAY)),2) AS AVG_delivery_time_IN_DAYS

FROM `target.customers` c
INNER JOIN `target.orders` o

ON c.customer_id = o.customer_id
```

```
GROUP BY 1
ORDER BY 2 ASC
LIMIT 5)
```

Row	customer_state	AVG_delivery_time_IN_DAYS
1	RR	28.98
2	AP	26.73
3	AM	25.99
4	AL	24.04
5	PA	23.32
6	SP	8.3
7	PR	11.53
8	MG	11.54
9	DF	12.51
10	SC	14.48

HIGHEST

LOWEST

Q.5 PART-4 Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.

```
SELECT c.customer_state, ROUND(AVG(DATE_DIFF(order_estimated_delivery_date,
order_delivered_customer_date, DAY)),2) AS AVG_delivery_diff_IN_DAYS

FROM `target.customers` c
INNER JOIN `target.orders` o

ON c.customer_id = o.customer_id

WHERE (order_estimated_delivery_date IS NOT NULL) AND
(order_delivered_customer_date IS NOT NULL)
GROUP BY 1
ORDER BY 2 ASC
LIMIT 5
```

Row	customer_state	AVG_delivery_diff_IN_DAYS
1	AL	7.95
2	MA	8.77
3	SE	9.17
4	ES	9.62
5	BA	9.93

Q.6 PART-1 Find the month on month no. of orders placed using different payment type.

```
SELECT EXTRACT(MONTH FROM o.order_purchase_timestamp),
       p.payment_type as payment_method,
       COUNT(DISTINCT o.order_id) AS no_of_orders_placed


FROM `target.payments` p
INNER JOIN `target.orders` o

ON o.order_id = p.order_id

GROUP BY p.payment_type, EXTRACT(MONTH FROM o.order_purchase_timestamp)

ORDER BY 1,2
```

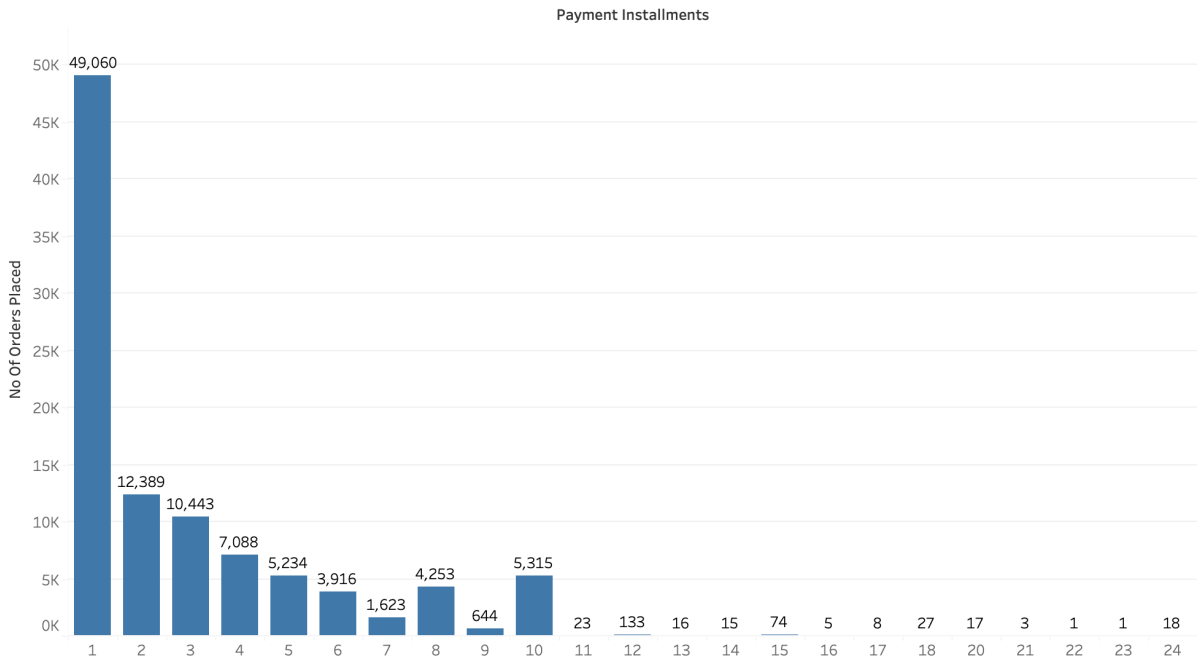
Row	f0_	payment_method	no_of_orders_placed
1	1	UPI	1715
2	1	credit_card	6093
3	1	debit_card	118
4	1	voucher	337
5	2	UPI	1723
6	2	credit_card	6582
7	2	debit_card	82
8	2	voucher	288
9	3	UPI	1942
10	3	credit_card	7682

Q.6 PART-2 Find the no. of orders placed on the basis of the payment instalments that have been paid.

```
SELECT COUNT(DISTINCT order_id) AS no_of_orders_placed

FROM `target.payments`

WHERE payment_installments > 0
GROUP BY payment_installments
```



ACTIONABLE INSIGHTS & RECOMMENDATIONS

1. As there is an uptrend in the number of orders placed over the past years, we can direct our efforts in maintaining these customers while simultaneously looking for growth. To do so, we can run loyalty-programs and special member-only prices.

    a. As we see a peak in number of orders placed in the month of November, we can run intense ad programs showcasing our best offers and top-selling products.

    b. We can invite influencers to our stores in October.

    c. We can also hire temporary staff to manage the store better.

    d. We can increase inventory of our top-selling products

2. As we see a peak in the number of orders placed in the Afternoon, we can have a multicuisine in-store restaurant.

3. Along with increase in number of orders placed, there is a 138% increase in the cost of orders too which is a very healthy sign. It's a good time to up-sell and cross-sell products.

4. As credit card is the preferred mode of payment for most of our customers, we can launch our own credit card by tying-up with a bank offering unlimited 5% cashback like Flipkart and Amazon already have.

5. As most orders are placed with up to 10 instalments, we can charge higher ROI on short-term EMIs.