**PRASHANT KISHLAY**

**SEPTEMBER’ 22 MORNING**

1. Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset
   1. Data type of columns in a table
   2. Time period for which the data is given
   3. Cities and States of customers ordered during the given period

SOLUTION 1.1.

|  |  |
| --- | --- |
| Customers | |
| **Field name** | | **Type** | |
| customer\_id | | STRING | |
| customer\_unique\_id | | STRING | |
| customer\_zip\_code\_prefix | | INTEGER | |
| customer\_city | | STRING | |
| customer\_state | | STRING | |
|
| |  |  | | --- | --- | | geolocation | | | Field name | Type | | geolocation\_zip\_code\_prefix | INTEGER | | geolocation\_lat | FLOAT | | geolocation\_lng | FLOAT | | geolocation\_city | STRING | | geolocation\_state | STRING | | |
|  | |
|  | |
|  | |
|  | |

|  |  |
| --- | --- |
| order\_items | |
| Field name | Type |
| order\_id | STRING |
| order\_item\_id | INTEGER |
| product\_id | STRING |
| seller\_id | STRING |
| shipping\_limit\_date | TIMESTAMP |
| Price | FLOAT |
| freight\_value | FLOAT |

|  |  |
| --- | --- |
| order\_reviews | |
| Field name | Type |
| review\_id | STRING |
| order\_id | STRING |
| review\_score | INTEGER |
| review\_comment\_title | STRING |
| review\_creation\_date | TIMESTAMP |
| review\_answer\_timestamp | TIMESTAMP |

|  |  |
| --- | --- |
| orders | |
|  |  |
| Field name | 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 |

|  |  |
| --- | --- |
| payments | |
| Field name | Type |
| order\_id | STRING |
| payment\_sequential | INTEGER |
| payment\_type | STRING |
| payment\_installments | INTEGER |
| payment\_value | FLOAT |

|  |  |
| --- | --- |
| products | |
| Field name | Type |
| product\_id | STRING |
| product\_category | STRING |
| product\_name\_length | INTEGER |
| product\_description\_length | INTEGER |
| product\_photos\_qty | INTEGER |
| product\_weight\_g | INTEGER |
| product\_length\_cm | INTEGER |
| product\_height\_cm | INTEGER |
| product\_width\_cm | INTEGER |

|  |  |
| --- | --- |
| sellers | |
| Field name | Type |
| seller\_id | STRING |
| seller\_zip\_code\_prefix | INTEGER |
| seller\_city | STRING |
| seller\_state | STRING |

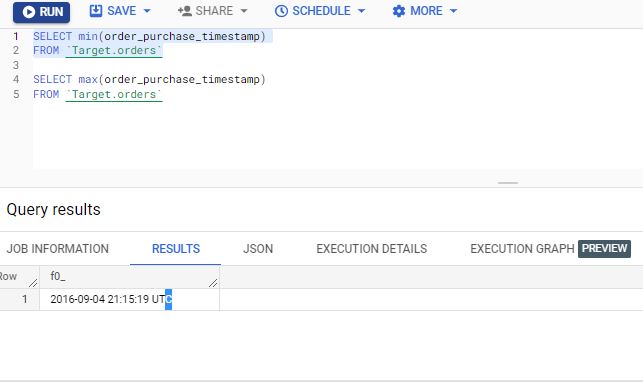
SOLUTION 1.2

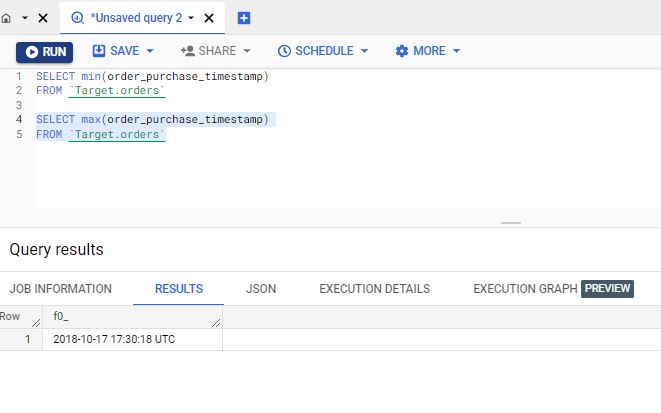
SELECT min(order\_purchase\_timestamp)

FROM `Target.orders`

SELECT max(order\_purchase\_timestamp)

FROM `Target.orders`





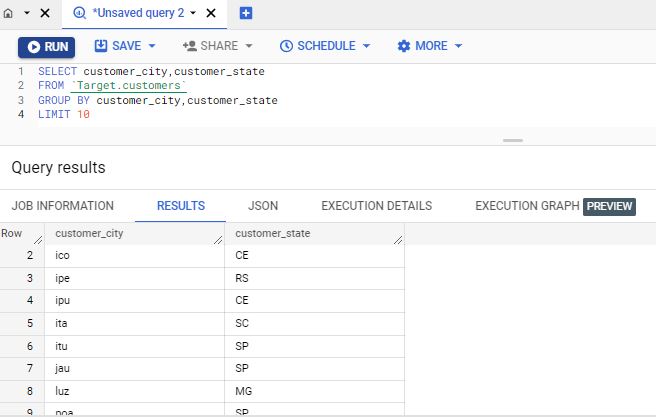
SOLUTION 1.3

SELECT customer\_city,customer\_state

FROM `Target.customers`

GROUP BY customer\_city,customer\_state

LIMIT 10



**INFERENCE:**

The data types of various tables and their columns are shown above.

The data is present starting from 4th September 2016 and is extended upto 17Th Oct 2018. Also the data is present cities and states that are all in Brazil .

**RECOMMENDATIONS:**

The data should be present for complete months not just starting from 4 th of any month and ending with any odd date of any month. The complete data for all 3 years would have made the analysis more clear.

1. In-depth Exploration:
   1. Is there a growing trend on e-commerce in Brazil? How can we describe a complete scenario? Can we see some seasonality with peaks at specific months?
   2. What time do Brazilian customers tend to buy (Dawn, Morning, Afternoon or Night)?

SOLUTION 2.1.

SELECT YEAR,SUM(payment\_value)

FROM

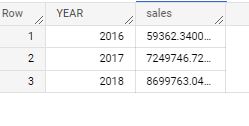
(SELECT EXTRACT(YEAR FROM(O.order\_purchase\_timestamp))AS YEAR,payment\_value

FROM `Target.orders` as O INNER JOIN `Target.payments` AS P

ON O.order\_id=P.order\_id) AS A

GROUP BY A.YEAR

ORDER BY YEAR



SELECT YEARMONTH,SUM(payment\_value) as sales

FROM

(SELECT CONCAT(EXTRACT(YEAR FROM(O.order\_purchase\_timestamp)),"-",EXTRACT(MONTH FROM (O.order\_purchase\_timestamp)))AS YEARMONTH,

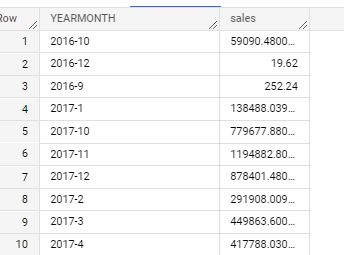
payment\_value

FROM `Target.orders` as O INNER JOIN `Target.payments` AS P

ON O.order\_id=P.order\_id) AS A

GROUP BY A.YEARMONTH

ORDER BY YEARMONTH



SOLUTION 2.2:

WITH order\_segmentation AS (SELECT \*,

case

 when extract(time FROM order\_purchase\_timestamp)>'00:00:00' AND extract(time FROM order\_purchase\_timestamp)<"06:00:00"

 THEN 'DAWN'

 WHEN extract(time FROM order\_purchase\_timestamp)>'06:00:00' AND extract(time FROM order\_purchase\_timestamp)<"12:00:00"

 THEN "MORNING"

 WHEN extract(time FROM order\_purchase\_timestamp)>'12:00:00' AND extract(time FROM order\_purchase\_timestamp)<"18:00:00"

 THEN "AFTERNOON"

 WHEN extract(time FROM order\_purchase\_timestamp)>"18:00:00" AND extract(time FROM order\_purchase\_timestamp)<="23:59:59"

 THEN "NIGHT"

END as Time

 from `Target.orders` )

 SELECT Time,count(distinct order\_id)

 FROM order\_segmentation

 GROUP BY Time



**INFERENCE :**

It is observed that the data represents **growing trend** is orders and sales from years 2016 to 2018 with **2018 receiving maximum sales** .

Also the sales are observed to peak by **around July, August month** and then a dip is seen Sep **To Dec Months**.

Most customers tend **to buy around afternoon and night time from 12 noon to 12 midnight**, the orders at morning and dawn are much lower compared to them.

**RECOMMENDATIONS:**

As there has been a dip seen in sales from months **Sep to Dec, stock clearance** sale should be announced at that time along with some offers to increase the sales those months.

As there **is lower sales at dawn and morning compared to that of afternoon and night times, some offers should be announced at those odd hours** to attract customers to shop at those times. There are people who tend to buy only during sale timings so it might prove to be a vital point of attraction for those customers .

* Evolution of E-commerce orders in the Brazil region:
  + Get month on month orders by states
  + Distribution of customers across the states in Brazil

SOLUTION 3.1:

WITH ppl\_state AS

(SELECT \*,EXTRACT(MONTH FROM order\_purchase\_timestamp) AS Month

FROM `Target.orders` O

INNER JOIN `Target.customers` C ON O.customer\_id=C.customer\_id

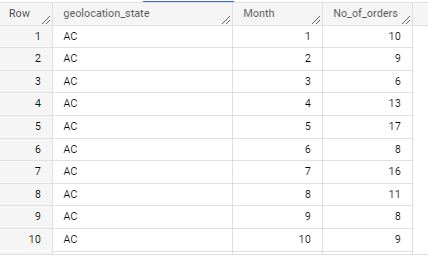
INNER JOIN `Target.geolocation` G ON C.customer\_zip\_code\_prefix=G.geolocation\_zip\_code\_prefix)

SELECT geolocation\_state,Month,COUNT(distinct Order\_id) as No\_of\_orders

FROM ppl\_state

GROUP BY geolocation\_state,Month

ORDER BY geolocation\_state,Month



SOLUTION 3.2:

WITH ppl\_state AS

(SELECT \*,EXTRACT(MONTH FROM order\_purchase\_timestamp) AS Month

FROM `Target.orders` O

INNER JOIN `Target.customers` C ON O.customer\_id=C.customer\_id

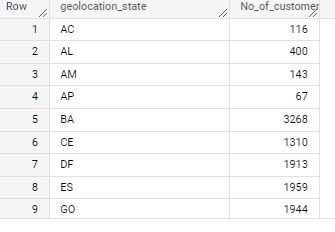
INNER JOIN `Target.geolocation` G ON C.customer\_zip\_code\_prefix=G.geolocation\_zip\_code\_prefix)

SELECT geolocation\_state,COUNT(distinct Customer\_unique\_id) as No\_of\_customers

FROM ppl\_state

GROUP BY geolocation\_state

ORDER BY geolocation\_state



**INFERENCE:**

With **minimum** orders and customers, the states such as **RR, AC, AL, AM** etc are lowest customer indulgence states,

With **maximum orders** and customers , the states such as **SP,RJ,MG,RS,PR** are the states having maximum customer indulgence .

**RECOMMENDATIONS:**

The states **such as SP,RJ,MG etc having maximum** orders and customers should be paid attention to so that **there is faster delivery** leading to **more boost** in sales as these are best performing states.

Also **various offers** should be introduced **in low performing states such as RR, AC, AL,AM so** that a considerable increase in sales can be seen over there.

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

* + 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
  + Mean & Sum of price and freight value by customer state

SOLUTION 4.1:

WITH pct\_cost AS

(SELECT \*,EXTRACT(YEAR FROM O.order\_purchase\_timestamp) Year FROM

`Target.orders` O INNER JOIN`Target.payments` P

On O.order\_id=P.order\_id

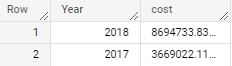
WHERE EXTRACT(YEAR FROM O.order\_purchase\_timestamp) IN (2017,2018)

AND Extract(MONTH FROM O.order\_purchase\_timestamp) IN (01,02,03,04,05,06,07,08))

SELECT Year,SUM(payment\_value) cost

FROM  pct\_cost

GROUP BY Year



SOLUTION 4.2:

WITH spf AS

(SELECT order\_id,s.seller\_state,price,freight\_value

FROM `Target.order\_items` AS oi

INNER JOIN `Target.sellers` AS s

ON oi.seller\_id=s.seller\_id)

SELECT seller\_state as state,

SUM(price)/count(order\_id) as price\_mean,

SUM(freight\_value)/COUNT(order\_id) as freight\_mean,

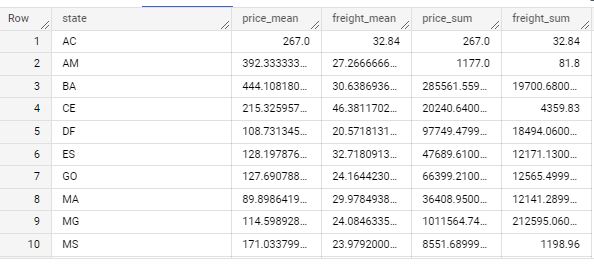
SUM(price) as price\_sum,

SUM(freight\_value) as freight\_sum

FROM spf

GROUP BY seller\_state

ORDER BY state



**INFERENCE:**

By using the tables orders and payments , it is derived that there has been **136.97% increase in cost of orders** from **the year 2017 to 2018** for the months Jan to August.

The states having lower mean price and freight mean can be viewed using above query.

**RECOMMENDATIONS:**

**The strategy used in states for lower price and lower freight mean should be adopted in states of higher price.**

Analysis on sales, freight and delivery time

* Calculate days between purchasing, delivering and estimated delivery
* Find time\_to\_delivery & diff\_estimated\_delivery. Formula for the same given below:
  + time\_to\_delivery = order\_purchase\_timestamp-order\_delivered\_customer\_date
  + diff\_estimated\_delivery = order\_estimated\_delivery\_date-order\_delivered\_customer\_date
* Group data by state, take mean of freight\_value, time\_to\_delivery, diff\_estimated\_delivery
* Sort the data to get the following:
* Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5
* Top 5 states with highest/lowest average time to delivery
* Top 5 states where delivery is really fast/ not so fast compared to estimated date

SOLUTION:

WITH avg as

(SELECT o.order\_id,

order\_purchase\_timestamp as pd,

o.order\_delivered\_customer\_date as dd,

date\_diff(o.order\_delivered\_customer\_date,order\_purchase\_timestamp,DAY) as time\_to\_delivery,

date\_diff(o.order\_estimated\_delivery\_date,o.order\_delivered\_customer\_date,DAY) as diff\_estimated\_delivery,

o.order\_estimated\_delivery\_date as ed,

S.seller\_state,

oi.price,

oi.freight\_value

FROM `Target.orders` o

INNER JOIN `Target.order\_items` as oi

ON o.order\_id=oi.order\_id

INNER JOIN `Target.sellers` as S

ON oi.seller\_id=S.seller\_id

ORDER BY dd DESC)

SELECT seller\_state,

AVG(freight\_value) as average\_freight\_value,

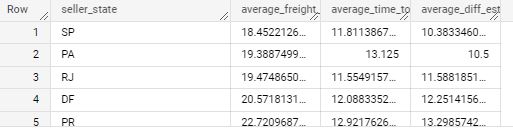
AVG(time\_to\_delivery) as average\_time\_to\_delivery,

AVG(avg.diff\_estimated\_delivery) as average\_diff\_estimated\_value

FROM avg

GROUP BY seller\_state

ORDER BY average\_freight\_value ASC



WITH avg as

(SELECT o.order\_id,

order\_purchase\_timestamp as pd,

o.order\_delivered\_customer\_date as dd,

date\_diff(o.order\_delivered\_customer\_date,order\_purchase\_timestamp,DAY) as time\_to\_delivery,

date\_diff(o.order\_estimated\_delivery\_date,o.order\_delivered\_customer\_date,DAY) as diff\_estimated\_delivery,

o.order\_estimated\_delivery\_date as ed,

S.seller\_state,

oi.price,

oi.freight\_value

FROM `Target.orders` o

INNER JOIN `Target.order\_items` as oi

ON o.order\_id=oi.order\_id

INNER JOIN `Target.sellers` as S

ON oi.seller\_id=S.seller\_id

ORDER BY dd DESC)

SELECT seller\_state,

AVG(freight\_value) as average\_freight\_value,

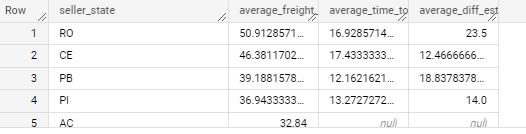
AVG(time\_to\_delivery) as average\_time\_to\_delivery,

AVG(avg.diff\_estimated\_delivery) as average\_diff\_estimated\_value

FROM avg

GROUP BY seller\_state

ORDER BY average\_freight\_value DESC



WITH avg as

(SELECT o.order\_id,

order\_purchase\_timestamp as pd,

o.order\_delivered\_customer\_date as dd,

date\_diff(o.order\_delivered\_customer\_date,order\_purchase\_timestamp,DAY) as time\_to\_delivery,

date\_diff(o.order\_estimated\_delivery\_date,o.order\_delivered\_customer\_date,DAY) as diff\_estimated\_delivery,

o.order\_estimated\_delivery\_date as ed,

S.seller\_state,

oi.price,

oi.freight\_value

FROM `Target.orders` o

INNER JOIN `Target.order\_items` as oi

ON o.order\_id=oi.order\_id

INNER JOIN `Target.sellers` as S

ON oi.seller\_id=S.seller\_id

ORDER BY dd DESC)

SELECT seller\_state,

AVG(freight\_value) as average\_freight\_value,

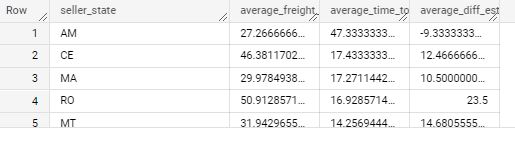
AVG(time\_to\_delivery) as average\_time\_to\_delivery,

AVG(avg.diff\_estimated\_delivery) as average\_diff\_estimated\_value

FROM avg

GROUP BY seller\_state

ORDER BY average\_time\_to\_delivery DESC



WITH avg as

(SELECT o.order\_id,

order\_purchase\_timestamp as pd,

o.order\_delivered\_customer\_date as dd,

date\_diff(o.order\_delivered\_customer\_date,order\_purchase\_timestamp,DAY) as time\_to\_delivery,

date\_diff(o.order\_estimated\_delivery\_date,o.order\_delivered\_customer\_date,DAY) as diff\_estimated\_delivery,

o.order\_estimated\_delivery\_date as ed,

S.seller\_state,

oi.price,

oi.freight\_value

FROM `Target.orders` o

INNER JOIN `Target.order\_items` as oi

ON o.order\_id=oi.order\_id

INNER JOIN `Target.sellers` as S

ON oi.seller\_id=S.seller\_id

ORDER BY dd DESC)

SELECT seller\_state,

AVG(freight\_value) as average\_freight\_value,

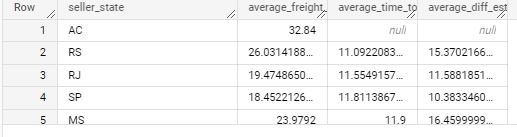
AVG(time\_to\_delivery) as average\_time\_to\_delivery,

AVG(avg.diff\_estimated\_delivery) as average\_diff\_estimated\_value

FROM avg

GROUP BY seller\_state

ORDER BY average\_time\_to\_delivery ASC



WITH avg as

(SELECT o.order\_id,

order\_purchase\_timestamp as pd,

o.order\_delivered\_customer\_date as dd,

date\_diff(o.order\_delivered\_customer\_date,order\_purchase\_timestamp,DAY) as time\_to\_delivery,

date\_diff(o.order\_estimated\_delivery\_date,o.order\_delivered\_customer\_date,DAY) as diff\_estimated\_delivery,

o.order\_estimated\_delivery\_date as ed,

S.seller\_state,

oi.price,

oi.freight\_value

FROM `Target.orders` o

INNER JOIN `Target.order\_items` as oi

ON o.order\_id=oi.order\_id

INNER JOIN `Target.sellers` as S

ON oi.seller\_id=S.seller\_id

ORDER BY dd DESC)

SELECT seller\_state,

AVG(freight\_value) as average\_freight\_value,

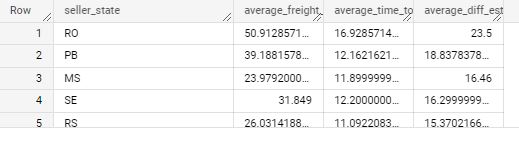
AVG(time\_to\_delivery) as average\_time\_to\_delivery,

AVG(avg.diff\_estimated\_delivery) as average\_diff\_estimated\_value

FROM avg

GROUP BY seller\_state

ORDER BY average\_diff\_estimated\_value DESC



WITH avg as

(SELECT o.order\_id,

order\_purchase\_timestamp as pd,

o.order\_delivered\_customer\_date as dd,

date\_diff(o.order\_delivered\_customer\_date,order\_purchase\_timestamp,DAY) as time\_to\_delivery,

date\_diff(o.order\_estimated\_delivery\_date,o.order\_delivered\_customer\_date,DAY) as diff\_estimated\_delivery,

o.order\_estimated\_delivery\_date as ed,

S.seller\_state,

oi.price,

oi.freight\_value

FROM `Target.orders` o

INNER JOIN `Target.order\_items` as oi

ON o.order\_id=oi.order\_id

INNER JOIN `Target.sellers` as S

ON oi.seller\_id=S.seller\_id

ORDER BY dd DESC)

SELECT seller\_state,

AVG(freight\_value) as average\_freight\_value,

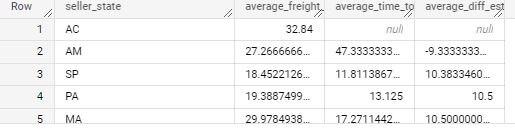
AVG(time\_to\_delivery) as average\_time\_to\_delivery,

AVG(avg.diff\_estimated\_delivery) as average\_diff\_estimated\_value

FROM avg

GROUP BY seller\_state

ORDER BY average\_diff\_estimated\_value ASC



**INFERENCE:**

All the states having lowest and highest freight value ,average time to delivery and average difference in estimated time and delivery are all listed above using the query that is written above.

**RECOMMENDATION :**

Special priority to logistics of states that take higher time to delivery should be provided by the company and also strategies of state having lower average time to delivery should be adopted.

6. Payment type analysis:

* Month over Month count of orders for different payment types
* Count of orders based on the no. of payment installments

SOLUTION :

WITH CTE AS

(SELECT CONCAT(EXTRACT(YEAR FROM order\_purchase\_timestamp),'-',EXTRACT(MONTH FROM order\_purchase\_timestamp)) as Year\_month,

payment\_type,

O.order\_id

FROM `Target.orders` as O INNER JOIN `Target.payments` as P

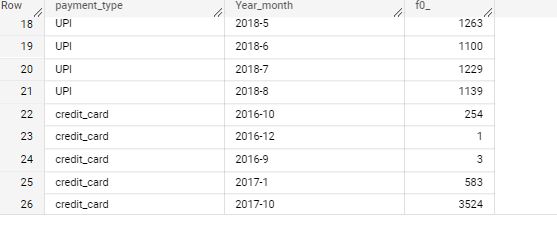
ON O.order\_id=P.order\_id)

SELECT payment\_type,Year\_month,Count(order\_id)

FROM CTE

GROUP BY payment\_type,Year\_month

ORDER BY payment\_type,Year\_month



WITH CTE AS

(SELECT payment\_installments,

O.order\_id

FROM `Target.orders` as O INNER JOIN `Target.payments` as P

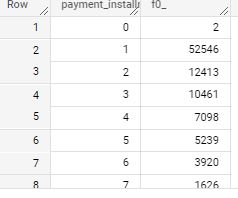
ON O.order\_id=P.order\_id)

SELECT payment\_installments,Count(order\_id)

FROM CTE

GROUP BY payment\_installments

ORDER BY payment\_installments



**INFERENCE:**

There are multiple payment types such as Credit\_card, UPI,Debit Card,Vouchers. Out of all of them , it is observed that **most order count has come with credit card as payment type followed by UPI**.

There has been installments upto 24 times starting with 1 installment time. It is observed that **most people prefer 1 installment** for their expenses and the **least people have gone for 22 or 23** installments.

**RECOMMENDATIONS:**

It can be one of the ideas to **partnership with Credit card companies for some offers to shop at Target and also some cashback should be offered when paying by UPI.**

The above recommendation might increase the sales or orders over months as it has been observed that **people choose credit cards and Upi mostly for their payments.**