#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

```
select column_name, data_type
from `annular-sky-381308.Target_SQL.INFORMATION_SCHEMA.COLUMNS`
where table_name = 'customers'
```

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

2. Time period for which the data is given

```
select min(order_purchase_timestamp) data_from,max(order_purchase_timestamp) data_to
from `Target_SQL.orders`
```

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

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

```
select c.customer_id,order_id,customer_city,customer_state
from `Target_SQL.customers`c
left join `Target_SQL.orders`o
on c.customer_id=o.customer_id
limit 10
```

Observation - no need to add condition for min and max of time period, because all corresponding data is in between the dates as mentioned in case 1.1

Row	customer_id	order_id	customer_city	customer_state
1	0735e7e4298a2ebbb46649346570476a	bf74f34eea55f16dd17b6212310074f8	acu	RN
2	903b3d86e3990db01619a4ebe3edef4e	667fc0af3acc404a6ef971908b1574b4	acu	RN
3	38c97666e962d4fea7fd6a83e69f20cd	9f738fc8b806bc3d86ccf78855e82eeb	acu	RN
4	77c2f46cf580f4874c9a5751c2d88474	9fd3d5bb20296499ef3fbcaa4db31c39	ico	CE
5	4d3ef4cfffb8ad4767c199c36a4cfee6	ecf6789fa93718435fc6279a4c051917	ico	CE
6	3000841b86e1fbe9493b523245d5c68d	9b41629ccbc3ae4be489cb815f3653f5	ico	CE
7	3c325415ccc7e622c66dec4bc9120030	5bb1e2f9ec792581a2209d429cfc1892	ico	CE
8	04f3a7b250e3be964f01bf22bccdc602	2e557b5d820cbc4f2f0e25b3867f8033	ico	CE
9	894202b8ef01f4719a4691e79dd24c17	d1ca60cfa5b276544043d344ad224285	ico	CE
10	9d715b9fb75a9d081c14126c09218b96	395522d74b4f19e9d229bacaab803c99	ico	CE

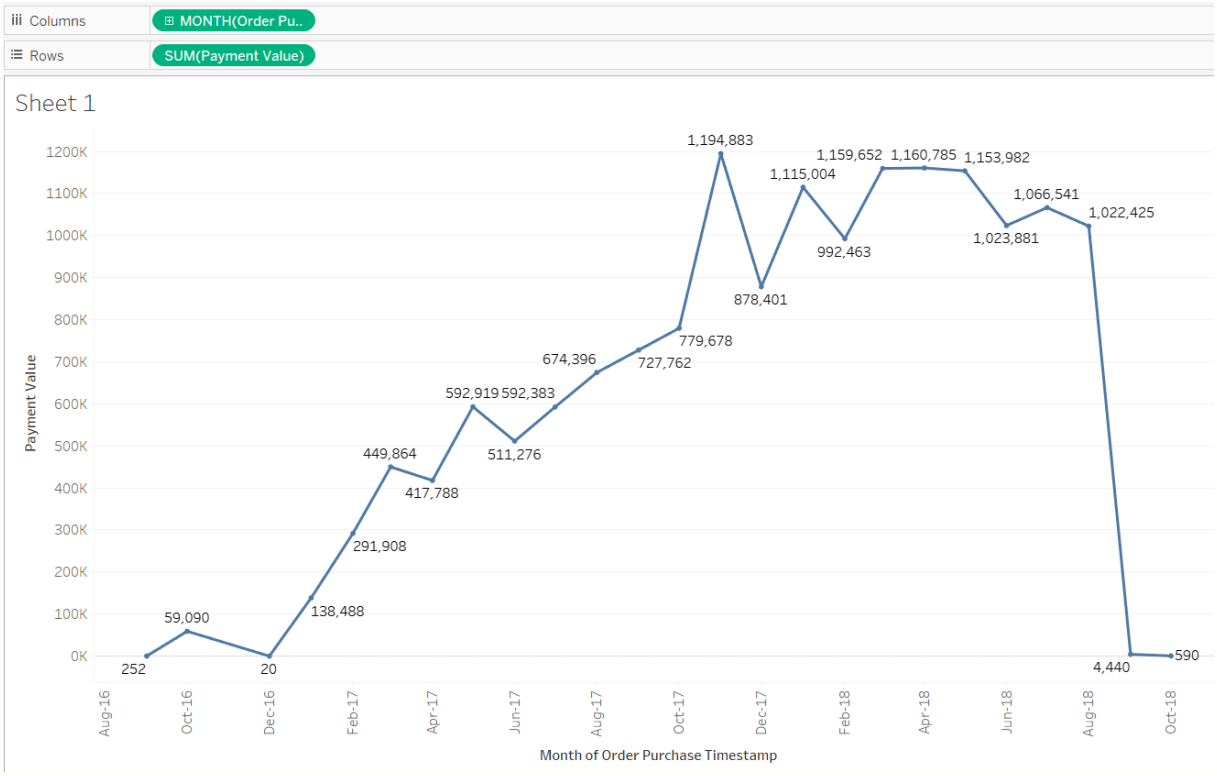
#2. In-depth Exploration:

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?

```
select*
from(
select extract(year from order_purchase_timestamp) year,
extract(month from order_purchase_timestamp) month,
count(o.order_id) total_orders,
round(sum(payment_value),1) total_sales
from `Target_SQL.orders`o
left join `Target_SQL.payments`p
on o.order_id=p.order_id
group by extract(year from order_purchase_timestamp),
extract(month from order_purchase_timestamp))
order by year, month
```

Observations/Insights –

1. yes,there is huge growth in brazilian e-commerce market from Jan 2017 to Aug 2018.
2. No. of orders along with sales had increase in this period
3. year 2017 is the profitable year among these years
4. max sales in Nov-2017



```
2. What time do Brazilian customers tend to buy
   (Dawn, Morning, Afternoon or Night)?
   0-6: Dawn
   7-12: Morning
   13-18: Afternoon
   19-23: Night

select time_part, count(order_purchase_timestamp) total_orders
from
(select *,
case
when (order_hour >= 0 and order_hour <= 6)
then 'Dawn (0-6)'
when (order_hour >= 7 and order_hour <= 12)
then 'Morning (7-12)'
when (order_hour >= 13 and order_hour <= 18)
then 'Afternoon (13-18)'
else 'Night (19-23)'
end time_part
from (select *,
extract(hour from order_purchase_timestamp) order_hour
from `Target_SQL.orders`
order by order_hour))
group by time_part
order by total_orders desc
```

Row	time_part	total_orders
1	Afternoon (13-18)	38135
2	Night (19-23)	28331
3	Morning (7-12)	27733
4	Dawn (0-6)	5242

Obervations/Insights –

1. Brazilian's tend to buy more in Afternoon, so keep all shops open in this time period,
2. And keep limited shops open in dawn, because sales is less

```
#3. Evolution of E-commerce orders in the Brazil region:
  1. Get month on month orders by states
  select *
  from(
  select customer_state,
  extract(year from order_purchase_timestamp) year,
  extract(month from order_purchase_timestamp) month,
  count(order_id) total_orders
  from `Target_SQL.orders`o
  left join `Target_SQL.customers`c
  on o.customer_id=c.customer_id
  group by customer_state,extract(year from order_purchase_timestamp),extract(month
  from order_purchase_timestamp))
  order by customer_state,year,month
  limit 30
```

Row	customer_state	year	month	total_orders
1	AC	2017	1	2
2	AC	2017	2	3
3	AC	2017	3	2
4	AC	2017	4	5
5	AC	2017	5	8
6	AC	2017	6	4
7	AC	2017	7	5
8	AC	2017	8	4
9	AC	2017	9	5
10	AC	2017	10	6
11	AC	2017	11	5
12	AC	2017	12	5
13	AC	2018	1	6
14	AC	2018	2	3
15	AC	2018	3	2

Observations/insights-
For example of state BA, month on month growth in orders can be seen

Row	customer_state	year	month	total_orders
79	BA	2016	10	4
80	BA	2017	1	25
81	BA	2017	2	59
82	BA	2017	3	91
83	BA	2017	4	93
84	BA	2017	5	127
85	BA	2017	6	106
86	BA	2017	7	155
87	BA	2017	8	158
88	BA	2017	9	170
89	BA	2017	10	166
90	BA	2017	11	250
91	BA	2017	12	192
92	BA	2018	1	239
93	BA	2018	2	214

2. Distribution of customers across the states in Brazil

```
select customer_state,
count(c.customer_id) total_customers,
round(sum(payment_value),1) total_sales
from `Target_SQL.customers`c
join `Target_SQL.orders`o
on c.customer_id=o.customer_id
join `Target_SQL.payments`p
on p.order_id=o.order_id
group by customer_state
order by total_sales desc
```

Row	customer_state	total_customers	total_sales
1	SP	43622	5998227.0
2	RJ	13527	2144379.7
3	MG	12102	1872257.3
4	RS	5668	890898.5
5	PR	5262	811156.4
6	SC	3754	623086.4
7	BA	3610	616645.8
8	DF	2204	355141.1
9	GO	2112	350092.3
10	ES	2107	325967.6

Observations/insights –
SP state max numbers of customers along with max sales, if there is a plan to open a new shop then SP state is great option

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

1. 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

```
select *,
round(((total_order_cost-lag1)/lag1)*100) month_wise_order_cost_percet_increase
from(
select *,
lag(total_order_cost,1) over(order by year,month) lag1
from(
select extract(year from order_purchase_timestamp) year,
extract(month from order_purchase_timestamp) month,
round(sum(payment_value),1) total_order_cost
from `Target_SQL.payments`p
left join `Target_SQL.orders`o
on o.order_id=p.order_id
where (extract(year from order_purchase_timestamp)=2017
or extract(year from orde r_purchase_timestamp)=2018)
and extract(month from order_purchase_timestamp)>=1
and extract(month from order_purchase_timestamp)<=8
group by extract(year from order_purchase_timestamp),
extract(month from order_purchase_timestamp))
```

```
order by year,month)
```

Row	year	month	total_order_cost	lag1	month_wise_order_cost_percet_increase
1	2017	1	138488.0	*null*	*null*
2	2017	2	291908.0	138488.0	111.0
3	2017	3	449863.6	291908.0	54.0
4	2017	4	417788.0	449863.6	-7.0
5	2017	5	592918.8	417788.0	42.0
6	2017	6	511276.4	592918.8	-14.0
7	2017	7	592382.9	511276.4	16.0
8	2017	8	674396.3	592382.9	14.0
9	2018	1	1115004.2	674396.3	65.0
10	2018	2	992463.3	1115004.2	-11.0
11	2018	3	1159652.1	992463.3	17.0
12	2018	4	1160785.5	1159652.1	0.0
13	2018	5	1153982.2	1160785.5	-1.0
14	2018	6	1023880.5	1153982.2	-11.0
15	2018	7	1066540.8	1023880.5	4.0
16	2018	8	1022425.3	1066540.8	-4.0

Note - Question asked for % increase in cost of orders year wise, to have more granularity wrote query to have month wise % increase in cost of orders, year wise % increase in cost of orders can also be calculated easily

Observations/insights –
peak in % increase in cost of orders in Feb-2017

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

```
select *,
round((total_freight_value/sum_of_price)*100,2) freight_percet
from(
select customer_state,
round(sum(p.payment_value)/count(p.payment_value),1) mean_of_price,
round(sum(p.payment_value),1) sum_of_price,
round(sum(freight_value)) total_freight_value
from `Target_SQL.customers`c
left join `Target_SQL.orders`o
on c.customer_id=o.customer_id
join `Target_SQL.payments`p
on o.order_id=p.order_id
join `Target_SQL.order_items`oi
on oi.order_id=o.order_id
group by customer_state
order by sum_of_price
)
order by freight_percet desc
```

Row	customer_state	mean_of_price	sum_of_price	total_freight_value	freight_percet
1	TO	213.2	72281.2	13451.0	18.61
2	RR	239.7	12462.2	2235.0	17.93
3	RO	230.4	65886.0	11717.0	17.78
4	RN	204.3	116264.9	20074.0	17.27
5	PE	199.2	376377.3	61924.0	16.45
6	PI	238.7	136780.0	22481.0	16.44
7	SE	222.8	88437.5	14541.0	16.44
8	AM	203.2	34753.3	5657.0	16.28
9	MA	235.3	198566.3	32290.0	16.26
10	PB	283.2	180984.2	27642.0	15.27

Observation/insights -
Customer state TO has maximum freight percentage charges as compare to other states

#5. Analysis on sales, freight and delivery time
1. Calculate days between purchasing, delivering and estimated delivery

```
select order_id,
order_purchase_timestamp,
order_delivered_customer_date,
order_estimated_delivery_date,
date_diff(order_delivered_customer_date,order_purchase_timestamp,day)
days_purchase_to_deliver,
date_diff(order_estimated_delivery_date,order_delivered_customer_date,day)
diff_estimate_and_deliver
from `Target_SQL.orders`
```

Row	order_id	order_purchase_timestamp	order_delivered_customer_date	order_estimated_delivery_date	days_purchase_to_deliver	diff_estimate_and_deliver
1	1950d777989f6a877539f5379...	2018-02-19 19:48:52 UTC	2018-03-21 22:03:51 UTC	2018-03-09 00:00:00 UTC	30	-12
2	2c45c33d2f9cb8ff8b1c86cc28...	2016-10-09 15:39:56 UTC	2016-11-09 14:53:50 UTC	2016-12-08 00:00:00 UTC	30	28
3	65d1e226dfaeb8cdc42f66542...	2016-10-03 21:01:41 UTC	2016-11-08 10:58:34 UTC	2016-11-25 00:00:00 UTC	35	16
4	635c894d068ac37e6e03dc54e...	2017-04-15 15:37:38 UTC	2017-05-16 14:49:55 UTC	2017-05-18 00:00:00 UTC	30	1
5	3b97562c3aee8bdedcb5c2e45...	2017-04-14 22:21:54 UTC	2017-05-17 10:52:15 UTC	2017-05-18 00:00:00 UTC	32	0
6	68f47f50f04c4cb6774570cfde...	2017-04-16 14:56:13 UTC	2017-05-16 09:07:47 UTC	2017-05-18 00:00:00 UTC	29	1
7	276e9ec344d3bf029ff83a161c...	2017-04-08 21:20:24 UTC	2017-05-22 14:11:31 UTC	2017-05-18 00:00:00 UTC	43	-4
8	54e1a3c2b97fb0809da548a59...	2017-04-11 19:49:45 UTC	2017-05-22 16:18:42 UTC	2017-05-18 00:00:00 UTC	40	-4
9	fd04fa4105ee8045f6a0139ca5...	2017-04-12 12:17:08 UTC	2017-05-19 13:44:52 UTC	2017-05-18 00:00:00 UTC	37	-1
10	302bb8109d097a9fc6e9cefc5...	2017-04-19 22:52:59 UTC	2017-05-23 14:19:48 UTC	2017-05-18 00:00:00 UTC	33	-5

Note:
Minus(-) in diff_estimate_and_deliver means order has been delivered after estimated date

2. 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

```
select order_id,
date_diff(order_delivered_customer_date,order_purchase_timestamp,day)
time_to_delivery,
date_diff(order_estimated_delivery_date,order_delivered_customer_date,day)
diff_estimated_delivery
from `Target_SQL.orders`
```

Row	order_id	time_to_delivery	diff_estimated_delivery
1	1950d777989f6a877539f53795b4c3c3	30	-12
2	2c45c33d2f9cb8ff8b1c86cc28c11c30	30	28
3	65d1e226dfaeb8cdc42f665422522d14	35	16
4	635c894d068ac37e6e03dc54eccb6189	30	1
5	3b97562c3aee8bdedcb5c2e45a50d5e1	32	0
6	68f47f50f04c4cb6774570cfde3a9aa7	29	1
7	276e9ec344d3bf029ff83a161c6b3ce9	43	-4
8	54e1a3c2b97fb0809da548a59f64c813	40	-4
9	fd04fa4105ee8045f6a0139ca5b49f27	37	-1
10	302bb8109d097a9fc6e9cefc5917d1f3	33	-5

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

```
select customer_state,
round(sum(freight_value)/count(freight_value),2) freight_mean_value,
round(sum(date_diff(order_delivered_customer_date,order_purchase_timestamp,day))/
count(date_diff(order_delivered_customer_date,order_purchase_timestamp,day)),2)
time_to_delivery,
round(sum(date_diff(order_estimated_delivery_date,order_delivered_customer_date,
day))/
count(date_diff(order_estimated_delivery_date,order_delivered_customer_date,
day)),2) diff_estimated_delivery

from `Target_SQL.customers`c
join `Target_SQL.orders`o
on c.customer_id=o.customer_id
join `Target_SQL.order_items`oi
on o.order_id=oi.order_id
group by customer_state
order by time_to_delivery desc
```

4. Sort the data to get the following:
5. Top 5 states with highest/lowest average freight value
   - sort in desc/asc limit 5
   - Top 5

```
select customer_state,
round(sum(freight_value)/count(freight_value),2) freight_mean_value,
round(sum(date_diff(order_delivered_customer_date,order_purchase_timestamp,day))/
count(date_diff(order_delivered_customer_date,order_purchase_timestamp,day)),2) t
ime_to_delivery,
round(sum(date_diff(order_estimated_delivery_date,order_delivered_customer_date,d
ay))/
count(date_diff(order_estimated_delivery_date,order_delivered_customer_date,day))
,2) diff_estimated_delivery

from `Target_SQL.customers`c
join `Target_SQL.orders`o
on c.customer_id=o.customer_id
join `Target_SQL.order_items`oi
on o.order_id=oi.order_id
```

```
  group by customer_state
  order by freight_mean_value desc
  limit 5
```

Row	customer_state	freight_mean_value	time_to_delivery	diff_estimated_delivery
1	RR	42.98	27.83	17.43
2	PB	42.72	20.12	12.15
3	RO	41.07	19.28	19.08
4	AC	40.07	20.33	20.01
5	PI	39.15	18.93	10.68

Observations/insights –
Freight mean value of customer states RR, PB, RO, AC greater than 40, these states need to work on freight value optimization

```
  Bottom 5
  select customer_state,
  round(sum(freight_value)/count(freight_value),2) freight_mean_value,
  round(sum(date_diff(order_delivered_customer_date,order_purchase_timestamp,day))/
  count(date_diff(order_delivered_customer_date,order_purchase_timestamp,day)),2) t
ime_to_delivery,
  round(sum(date_diff(order_estimated_delivery_date,order_delivered_customer_date,d
ay))/
  count(date_diff(order_estimated_delivery_date,order_delivered_customer_date,day))
,2) diff_estimated_delivery

  from `Target_SQL.customers`c
  join `Target_SQL.orders`o
  on c.customer_id=o.customer_id
  join `Target_SQL.order_items`oi
  on o.order_id=oi.order_id
  group by customer_state
  order by freight_mean_value
  limit 5
```

Row	customer_state	freight_mean_va	time_to_delivery	diff_estimated_c
1	SP	15.15	8.26	10.27
2	PR	20.53	11.48	12.53
3	MG	20.63	11.52	12.4
4	RJ	20.96	14.69	11.14
5	DF	21.04	12.5	11.27

```
  6. Top 5 states with highest/lowest average time to delivery
  Top 5
  select customer_state,
  round(sum(freight_value)/count(freight_value),2) freight_mean_value,
  round(sum(date_diff(order_delivered_customer_date,order_purchase_timestamp,day))/
  count(date_diff(order_delivered_customer_date,order_purchase_timestamp,day)),2) t
ime_to_delivery,
  round(sum(date_diff(order_estimated_delivery_date,order_delivered_customer_date,d
ay))/
```

```
  count(date_diff(order_estimated_delivery_date,order_delivered_customer_date,day))
,2) diff_estimated_delivery

  from `Target_SQL.customers`c
  join `Target_SQL.orders`o
  on c.customer_id=o.customer_id
  join `Target_SQL.order_items`oi
  on o.order_id=oi.order_id
  group by customer_state
  order by time_to_delivery desc
  limit 5
```

Row	customer_state	freight_mean_value	time_to_delivery	diff_estimated_delivery
1	RR	42.98	27.83	17.43
2	AP	34.01	27.75	17.44
3	AM	33.21	25.96	18.98
4	AL	35.84	23.99	7.98
5	PA	35.83	23.3	13.37

Observations/insights –
RR state must work on time to deliver the order, has maximum delivery time as compared to other states

```
  Bottom 5
  select customer_state,
  round(sum(freight_value)/count(freight_value),2) freight_mean_value,
  round(sum(date_diff(order_delivered_customer_date,order_purchase_timestamp,day))/
  count(date_diff(order_delivered_customer_date,order_purchase_timestamp,day)),2) t
ime_to_delivery,
  round(sum(date_diff(order_estimated_delivery_date,order_delivered_customer_date,d
ay))/
  count(date_diff(order_estimated_delivery_date,order_delivered_customer_date,day))
,2) diff_estimated_delivery

  from `Target_SQL.customers`c
  join `Target_SQL.orders`o
  on c.customer_id=o.customer_id
  join `Target_SQL.order_items`oi
  on o.order_id=oi.order_id
  group by customer_state
  order by time_to_delivery
  limit 5
```

Row	customer_state	freight_mean_value	time_to_delivery	diff_estimated_delivery
1	SP	15.15	8.26	10.27
2	PR	20.53	11.48	12.53
3	MG	20.63	11.52	12.4
4	DF	21.04	12.5	11.27
5	SC	21.47	14.52	10.67

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

Top 5

```
select customer_state,
round(sum(diff_estimated_delivery)/count(diff_estimated_delivery),2)
delivery_compared_to_estimated_date
from(select *,
date_diff(order_estimated_delivery_date,order_delivered_customer_date,day)
diff_estimated_delivery
from `Target_SQL.customers`c
left join `Target_SQL.orders`o
on c.customer_id=o.customer_id)
group by customer_state
order by delivery_compared_to_estimated_date desc
limit 5
```

Row	customer_state	delivery_compared_to_estimated_date
1	AC	19.76
2	RO	19.13
3	AP	18.73
4	AM	18.61
5	RR	16.41

Bottom 5

```
select customer_state,
round(sum(diff_estimated_delivery)/count(diff_estimated_delivery),2)
delivery_compared_to_estimated_date
from(select *,
date_diff(order_estimated_delivery_date,order_delivered_customer_date,day)
diff_estimated_delivery
from `Target_SQL.customers`c
left join `Target_SQL.orders`o
on c.customer_id=o.customer_id)
group by customer_state
order by delivery_compared_to_estimated_date
limit 5
```

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

#6. Payment type analysis:

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

```
select *
from(
select extract(year from order_purchase_timestamp) year,
extract(month from order_purchase_timestamp) month,
```

```
payment_type,
count(o.order_id) order_count
from `Target_SQL.payments`p
left join `Target_SQL.orders`o
on p.order_id=o.order_id
group by extract(year from order_purchase_timestamp),
extract(month from order_purchase_timestamp),payment_type)
order by year,month,order_count desc
```

Row	year	month	payment_type	order_count
1	2016	9	credit_card	3
2	2016	10	credit_card	254
3	2016	10	UPI	63
4	2016	10	voucher	23
5	2016	10	debit_card	2
6	2016	12	credit_card	1
7	2017	1	credit_card	583
8	2017	1	UPI	197
9	2017	1	voucher	61
10	2017	1	debit_card	9
11	2017	2	credit_card	1356
12	2017	2	UPI	398
13	2017	2	voucher	119
14	2017	2	debit_card	13
15	2017	3	credit_card	2016

Observations –
Customer prefers credit card as top payment type

```
2.Count of orders based on the no. of payment installments
select payment_installments,count(order_id) total_orders
from `Target_SQL.payments`
group by payment_installments
```

Row	payment_installments	total_orders
1	0	2
2	1	52546
3	2	12413
4	3	10461
5	4	7098
6	5	5239
7	6	3920
8	7	1626
9	8	4268
10	9	644
11	10	5328
12	11	23
13	12	133
14	13	16
15	14	15

#7. Actionable Insights

- Brazilian's tend to buy more in Afternoon, so keep all shops open in this time period
- keep limited shops open in dawn, because sales is less, to avoid labour cost etc.
- Customer state TO has maximum freight percentage charges as compare to other states

#8. Recommendations

- SP state has maximum number of customers along with max sales, if there is a plan to open a new shop then SP state is great option
- RR state must work on time to deliver the order, has maximum delivery time as compared to other states
- Freight mean value of customer states RR, PB, RO, AC greater than 40, these states need to work on freight value optimization