Target Analysis

1.a Data type of all columns in the "customers" table

```
select column_name,data_type
from target_analysis.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

1.b Get the time range between which the orders were placed

```
SELECT
  MIN(order_purchase_timestamp) AS mintime,
  MAX(order_purchase_timestamp) AS maxtime
FROM
  `target_analysis.orders` AS o
JOIN
  `target_analysis.customers` AS c
ON
  o.customer_id=c.customer_id
WHERE
  o.order_purchase_timestamp BETWEEN "2016-09-04"
  AND "2018-10-17"
```

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

Target Analysis

1.c Count the Cities & States of customers who ordered during the given period

```
SELECT
  COUNT(DISTINCT C.customer_city) AS NumberOfCities,
  COUNT(DISTINCT C.customer_state) AS NumberOfStates
FROM
  `target_analysis.customers` AS C
JOIN
  `target_analysis.orders` AS O
ON
  C.customer_id= O.customer_id
WHERE
  O.order_purchase_timestamp BETWEEN "2016-09-04"
  AND "2018-10-17"
```

Row	NumberOfCities	NumberOfStates
1	4119	27

In-depth Exploration:

2

a.Is there a growing trend in the no. of orders placed over the past years

```
SELECT
  EXTRACT(year
  FROM
    order_purchase_timestamp) AS year,
  COUNT(*) AS num_of_orders
FROM
  `target_analysis.orders`
GROUP BY
  year
```

Row	year	num_of_orders
1	2017	45101
2	2018	54011
3	2016	329

Target Analysis

* 2018 have more number of orders among the years
* orders was increasing compare to past years

b.Can we see some kind of monthly seasonality in terms of the no. of orders being placed

```
SELECT
  EXTRACT(month
  FROM
    order_purchase_timestamp) AS month,
  COUNT(*) AS num_of_orders
FROM
  `target_analysis.orders`
GROUP BY
  month
```

Row	month	num_of_orders
1	11	7544
2	12	5674
3	2	8508
4	4	9343
5	7	10318
6	5	10573
7	10	4959
8	1	8069
9	6	9412
10	9	4305
11	3	9893
12	8	10843

*** IN Month August ,May,July Have More Number Of Orders Was Placed**
*** Compare To All Months September Have Less Orders**

Target Analysis

C.During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)

- **0-6 hrs : Dawn**
- **7-12 hrs : Mornings**
- **13-18 hrs : Afternoon**
- **19-23 hrs : Night**

```
SELECT
    CASE
        WHEN EXTRACT(HOUR FROM order_purchase_timestamp) BETWEEN 0 AND 6 THEN
'Dawn'
        WHEN EXTRACT(HOUR FROM order_purchase_timestamp) BETWEEN 7 AND 12 THEN
'Mornings'
        WHEN EXTRACT(HOUR FROM order_purchase_timestamp) BETWEEN 13 AND 18 THEN
'Afternoon'
        WHEN EXTRACT(HOUR FROM order_purchase_timestamp) BETWEEN 19 AND 23 THEN
'Night'
        ELSE 'Other'
    END AS OrderTimeCategory,
    COUNT(*) AS NumberOfOrders
FROM
    `target_analysis.orders`
GROUP BY
    OrderTimeCategory
ORDER BY
    OrderTimeCategory;
```

Row	OrderTimeCategory	NumberOfOrders
1	Afternoon	38135
2	Dawn	5242
3	Mornings	27733
4	Night	28331

*** IN Brazil More Number Of Orders Was Placed in Afternoon And Night**

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

Target Analysis

A. Get the month on month no. of orders placed in each state

```
SELECT
  EXTRACT (month
  FROM
    o.order_purchase_timestamp) AS month,
  c.customer_state,
  COUNT(*) AS order_place
FROM
  `target_analysis.orders` AS o
INNER JOIN
  `target_analysis.customers` AS c
ON
  c.customer_id=o.customer_id
GROUP BY
  month,
  C.customer_state
```

Row	month	customer_state	order_place
1	11	RJ	1048
2	12	RS	283
3	12	SP	2357
4	2	DF	196
5	11	PR	378
6	4	MT	92
7	7	MA	79
8	7	AL	40
9	7	SP	4381
10	7	MT	85
11	7	MG	1111
12	5	MG	1190
13	5	SP	4632
14	5	PE	174
15	10	SP	1908

*** SP State Have More Orders Placed**

Target Analysis

B.How are the customers distributed across all the states

```
SELECT
  customer_state,
  COUNT(DISTINCT customer_id) AS num_of_customers
FROM
  `target_analysis.customers`
GROUP BY
  Customer_state
```

Row	customer_state	num_of_customers
1	RN	485
2	CE	1336
3	RS	5466
4	SC	3637
5	SP	41746
6	MG	11635
7	BA	3380
8	RJ	12852
9	GO	2020
10	MA	747
11	PE	1652
12	PB	536
13	ES	2033
14	PR	5045
15	RO	253

*** In SP State Have More Customers And RO State Have Less Customers**

Target Analysis

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

A. Get the % increase in the cost of orders from year 2017 to 2018 (include months between Jan to Aug only).
You can use the "payment_value" column in the payments table to get the cost of orders.

```
WITH OrderCosts AS (
    SELECT
        EXTRACT(YEAR FROM o.order_purchase_timestamp) AS OrderYear,
        EXTRACT(MONTH FROM o.order_purchase_timestamp) AS OrderMonth,
        SUM(p.payment_value) AS TotalPayment
    FROM
        `target_analysis.orders` o
    JOIN
        `target_analysis.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) BETWEEN 1 AND 8
    GROUP BY
        OrderYear,
        OrderMonth
)

SELECT
    oc1.OrderYear AS Year2017,
    oc1.OrderMonth AS Month,
    oc1.TotalPayment AS Cost2017,
    oc2.TotalPayment AS Cost2018,
    ((oc2.TotalPayment - oc1.TotalPayment) / oc1.TotalPayment) * 100 AS
PercentageIncrease
FROM
    OrderCosts oc1
JOIN
    OrderCosts oc2 ON oc1.OrderMonth = oc2.OrderMonth AND oc1.OrderYear = 2017 AND
oc2.OrderYear = 2018
ORDER BY
    oc1.OrderMonth;
```

Target Analysis

Row	Year2017	Month	Cost2017	Cost2018	PercentageIncrease
1	2017	1	138488.0399999...	1115004.180000...	705.1266954171...
2	2017	2	291908.0099999...	992463.3400000...	239.9918145445...
3	2017	3	449863.6000000...	1159652.119999...	157.7786066709...
4	2017	4	417788.0300000...	1160785.479999...	177.8407701149...
5	2017	5	592918.8200000...	1153982.149999...	94.62734375677...
6	2017	6	511276.3800000...	1023880.499999...	100.2596912456...
7	2017	7	592382.9200000...	1066540.750000...	80.04245463390...
8	2017	8	674396.3200000...	1022425.320000...	51.60600520477...

*** January Have More Percentage Increased In 2017 and 2018**

B.Calculate the Total & Average value of order price for each state

```
SELECT
  SUM(price) AS totalprice,
  AVG(price) AS avgprice,
  c.customer_state
FROM
  `target_analysis.customers` AS c
JOIN
  `target_analysis.orders` AS o
ON
  c.customer_id=o.customer_id
JOIN
  `target_analysis.orders_items` AS ol
ON
  ol.order_id=o.order_id
group by c.customer_state
order by c.customer_state
```

Target Analysis

Row	totalprice	avgprice	customer_state
1	15982.94999999...	173.7277173913...	AC
2	80314.80999999...	180.8892117117...	AL
3	22356.84000000...	135.4959999999...	AM
4	13474.29999999...	164.3207317073...	AP
5	511349.9900000...	134.6012082126...	BA
6	227254.7099999...	153.7582611637...	CE
7	302603.9399999...	125.7705486284...	DF
8	275037.3099999...	121.9137012411...	ES
9	294591.9499999...	126.2717316759...	GO
10	119648.2199999...	145.2041504854...	MA
11	1585308.029999...	120.7485741488...	MG
12	116812.6399999...	142.6283760683...	MS
13	156453.5299999...	148.2971848341...	MT
14	178947.8099999...	165.6924166666...	PA
15	115268.0799999...	191.4752159468...	PB

*** State MG have High TotalPrice And Less AvgPrice**

*** State PB And AL Have High AvgPrice and Less TotalPrice**

C.Calculate the Total & Average value of order freight for each state.

```
SELECT
  SUM(freight_value) AS totalfreight,
  AVG(freight_value) AS avgfreight,
  c.customer_state
FROM
  `target_analysis.customers` AS c
JOIN
  `target_analysis.orders` AS o
ON
  c.customer_id=o.customer_id
JOIN
  `target_analysis.orders_items` AS ol
ON
  ol.order_id=o.order_id
group by c.customer_state
order by c.customer_state
```

Target Analysis

Row	totalfreight	avgfreight	customer_state
1	3686.749999999...	40.07336956521...	AC
2	15914.58999999...	35.84367117117...	AL
3	5478.889999999...	33.20539393939...	AM
4	2788.500000000...	34.00609756097...	AP
5	100156.6799999...	26.36395893656...	BA
6	48351.58999999...	32.71420162381...	CE
7	50625.49999999...	21.04135494596...	DF
8	49764.59999999...	22.05877659574...	ES
9	53114.97999999...	22.76681525932...	GO
10	31523.77000000...	38.25700242718...	MA
11	270853.4600000...	20.63016680630...	MG
12	19144.03000000...	23.37488400488...	MS
13	29715.43000000...	28.16628436018...	MT
14	38699.30000000...	35.83268518518...	PA
15	25719.72999999...	42.72380398671...	PB

Target Analysis

5.Analysis based on sales, freight and delivery time.

A.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.
Do this in a single query.

You can calculate the delivery time and the difference between the estimated & actual delivery date using the given formula:

- **time_to_deliver = order_delivered_customer_date - order_purchase_timestamp**
- **diff_estimated_delivery = order_estimated_delivery_date - order_delivered_customer_date**

```
SELECT
    order_id,
    order_purchase_timestamp,
    order_delivered_customer_date,
    order_estimated_delivery_date,
    DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp,DAY) AS
delivery_time,
    DATE_DIFF(order_estimated_delivery_date, order_delivered_customer_date,DAY) AS
diff_estimated_delivery
FROM
    `target_analysis.orders`
WHERE
    order_delivered_customer_date IS NOT NULL
    AND order_estimated_delivery_date IS NOT NULL;
```

Target Analysis

Row	order_id	order_purchase_timestamp	order_delivered_customer_date	order_estimated_delivery_date	delivery_time	diff_estimated_deliv
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
11	66057d37308e787052a32828...	2017-04-15 19:22:06 UTC	2017-05-24 08:11:57 UTC	2017-05-18 00:00:00 UTC	38	-6
12	19135c945c554eebfd7576c73...	2017-07-11 14:09:37 UTC	2017-08-16 20:19:32 UTC	2017-08-14 00:00:00 UTC	36	-2
13	4493e45e7ca1084efcd38ddeb...	2017-07-11 20:56:34 UTC	2017-08-14 21:37:08 UTC	2017-08-14 00:00:00 UTC	34	0
14	70c77e51e0f179d75a64a6141...	2017-07-13 21:03:44 UTC	2017-08-25 19:41:53 UTC	2017-08-14 00:00:00 UTC	42	-11
15	d7918e406132d7c81f1b84527...	2017-07-13 17:54:53 UTC	2017-08-17 18:35:38 UTC	2017-08-14 00:00:00 UTC	35	-3

B.Find out the top 5 states with the highest & lowest average freight value

```
-- Top 5 states with the highest average freight value


select c.customer_state,
avg(freight_value) as avg_freight
from `target_analysis.orders` as o
join `target_analysis.customers` as c
on o.customer_id=c.customer_id
join `target_analysis.orders_items` as ol
on o.order_id=ol.order_id
group by c.customer_state
order by avg_freight desc
limit 5
```

Row	customer_state	avg_freight
1	RR	42.98442307692...
2	PB	42.72380398671...
3	RO	41.06971223021...
4	AC	40.07336956521...
5	PI	39.14797047970...

*** RR State Have High Average Freight**

Target Analysis

```sql
-- Top 5 states with the lowest  average freight value
select
 c.customer_state,
avg(freight_value) as avg_freight
from `target_analysis.orders` as o
join `target_analysis.customers` as c
on o.customer_id=c.customer_id
join `target_analysis.orders_items` as ol
on o.order_id=ol.order_id
group by c.customer_state
order by avg_freight asc
limit 5
```

Row	customer_state	avg_freight
1	SP	15.14727539041...
2	PR	20.53165156794...
3	MG	20.63016680630...
4	RJ	20.96092393168...
5	DF	21.04135494596...

*** SP State Lowest Average Freight**

C.Find out the top 5 states with the highest & lowest average delivery time

```sql
-- Top 5 states with the highest average delivery time
SELECT
  c.customer_state,
  AVG(DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY)) AS
AverageDeliveryTime
FROM
  `target_analysis.orders` AS o
JOIN
  `target_analysis.customers` AS c
ON
  o.customer_id=c.customer_id
WHERE
  order_delivered_customer_date IS NOT NULL
GROUP BY
  c.customer_state
ORDER BY
  AverageDeliveryTime DESC
```

Target Analysis

```
LIMIT
  5
```

Row	customer_state	AverageDeliveryTime
1	RR	28.97560975609...
2	AP	26.73134328358...
3	AM	25.98620689655...
4	AL	24.04030226700...
5	PA	23.31606765327...

*** RR State Have High Average Delivery Time**

```
-- Top 5 states with the lowest average delivery time

SELECT
  c.customer_state,
  AVG(DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY)) AS
AverageDeliveryTime
FROM
  `target_analysis.orders` AS o
JOIN
  `target_analysis.customers` AS c
ON
  o.customer_id=c.customer_id
WHERE
  order_delivered_customer_date IS NOT NULL
GROUP BY
  c.customer_state
ORDER BY

  AverageDeliveryTime asc
LIMIT
  5
```

Target Analysis

Row	customer_state	AverageDeliveryTime
1	SP	8.298061489072...
2	PR	11.52671135486...
3	MG	11.54381329810...
4	DF	12.50913461538...
5	SC	14.47956019171...

*** SP State Have Lowest Average Delivery Time**

D.Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.
You can use the difference between the averages of actual & estimated delivery date to figure out how fast the delivery was for each state.

```
WITH DeliveryTimeDifference AS (
    SELECT
        customer_state,
        AVG(DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp,DAY))
AS AverageActualDelivery,
        AVG(DATE_DIFF(order_estimated_delivery_date, order_purchase_timestamp,DAY))
AS AverageEstimatedDelivery
    FROM
        `target_analysis.orders` as o
        join `target_analysis.customers` as c
        on o.customer_id=c.customer_id
    WHERE
        order_delivered_customer_date IS NOT NULL
        AND order_estimated_delivery_date IS NOT NULL
    GROUP BY
       c.customer_state
)

SELECT
    customer_state,
    AverageActualDelivery,
    AverageEstimatedDelivery,
    AverageActualDelivery - AverageEstimatedDelivery AS DeliveryTimeDifference
FROM
    DeliveryTimeDifference
```

Target Analysis

```
ORDER BY
    DeliveryTimeDifference DESC
LIMIT 5;
```

Row	customer_state	AverageActualDelive	AverageEstimatedDe	DeliveryTimeDifferen
1	AL	24.04030226700...	32.20906801007...	-8.16876574307...
2	MA	21.11715481171...	30.08368200836...	-8.96652719665...
3	SE	21.02985074626...	30.48358208955...	-9.45373134328...
4	ES	15.33182957393...	25.22005012531...	-9.88822055137...
5	CE	20.817826426896	31.00469116497...	-10.1868647380...

6.Analysis based on the payments:

A.Find the month on month no. of orders placed using different payment types

```
SELECT
  EXTRACT(month
  FROM
    order_purchase_timestamp) AS month,
  p.payment_type,
  COUNT(*) AS num_of_orders
FROM
  `target_analysis.orders` AS o
JOIN
  `target_analysis.payments` AS p
ON
  o.order_id=p.order_id
GROUP BY
  p.payment_type,
  month
```

Target Analysis

Row	month	payment_type	num_of_orders
1	5	credit_card	8350
2	4	credit_card	7301
3	1	voucher	477
4	4	voucher	572
5	10	voucher	318
6	9	not_defined	1
7	8	not_defined	2
8	6	voucher	563
9	5	voucher	613
10	3	voucher	591
11	2	credit_card	6609
12	8	credit_card	8269
13	11	credit_card	5897
14	3	credit_card	7707
15	7	voucher	645

*** Customers Using More (Payment Type) Credit_Cards**

B.Find the no. of orders placed on the basis of the payment instalments that have been paid

```
SELECT
  payment_installments,
  COUNT(*) AS orders
FROM
  `target_analysis.payments`
GROUP BY
  Payment_installments
```

Target Analysis

Row	payment_installment	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

Recommendations :

1. Year by Year Orders increasing With High percentage .

2.Sp State Has More Orders and More Customers We Have To Focus On Reducing Delivery Time .

3.RO state Have Less Customers We Have To Communicate With Customers And Give Offers And Easy Payments Type .