BUSINESS CASE: TARGET SQL

Q 1.1 Data type of all columns in the "customers" table.

```
SELECT
    Column_name,
    DATA_TYPE FROM case_study1.INFORMATION_SCHEMA.COLUMNS
WHERE TABLE_NAME = 'customers';
```

```
1  --Q 1.1
2  --Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset:
3  --Data type of all columns in the "customers" table.
4
5  SELECT
6      column_name,
7      DATA_TYPE FROM case_study1.INFORMATION_SCHEMA.COLUMNS
8  WHERE TABLE_NAME = 'customers';
```

Press Alt+F1 for Accessibility Options.

Query results

SAVE RESULTS | EXPLORE DATA

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH

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 1.2 Get the time range between which the orders were placed.

```sql
SELECT
      MIN (order_purchase_timestamp) AS start_date,
      MAX (order_purchase_timestamp) AS end_date
FROM `case_study1.orders`;
```

```sql
--Q1.2
--Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset:
--Get the time range between which the orders were placed.

SELECT
      MIN(order_purchase_timestamp) AS start_date,
      MAX(order_purchase_timestamp) AS end_date
FROM `case_study1.orders`;
```

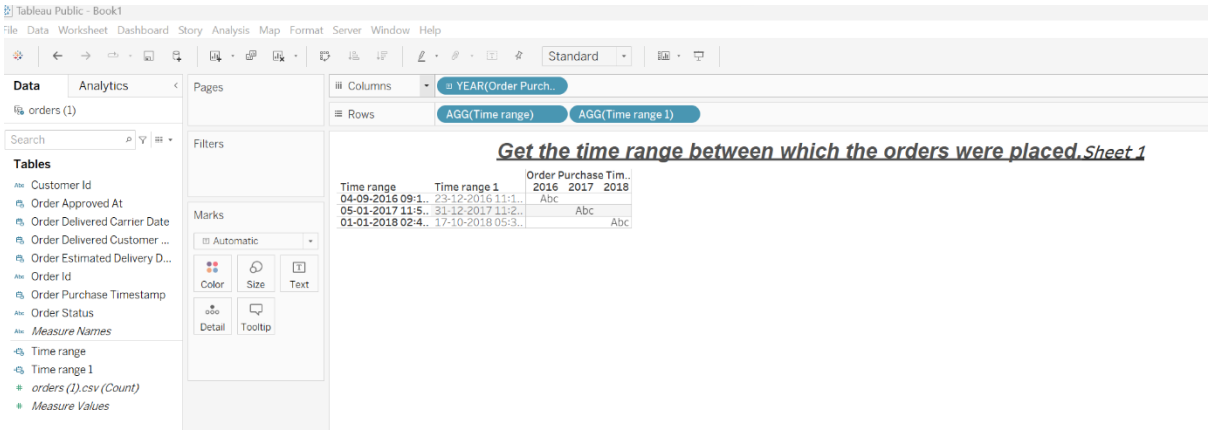
Press Alt+F1 for Accessibility Options.

Query results

SAVE RESULTS | EXPLORE DATA

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH

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



INSIGHTS :- *Orders were placed between Month of 4th September,2016 to Month of 17th october,2018.*

Q 1.3 Count the Cities & States of customers who ordered during the given period.

```
SELECT
    COUNT (DISTINCT c.customer_city) AS no_of_cities,
    COUNT (DISTINCT c.customer_state)AS no_of_states
FROM `case_study1.customers` c
JOIN `case_study1.orders` o
ON c.customer_id = o.customer_id;
```

```
1
2  --Q 1.3
3  --Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset:
4  --Count the Cities & States of customers who ordered during the given period.
5
6  SELECT
7      COUNT(DISTINCT c.customer_city) AS no_of_cities,
8      COUNT(DISTINCT c.customer_state)AS no_of_states
9  FROM `case_study1.customers` c
10 JOIN `case_study1.orders` o
11 ON c.customer_id = o.customer_id;
```

Press Alt+F1 for Accessibility Options.

Query results

SAVE RESULTS ▾ EXPLORE DATA ▾

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH

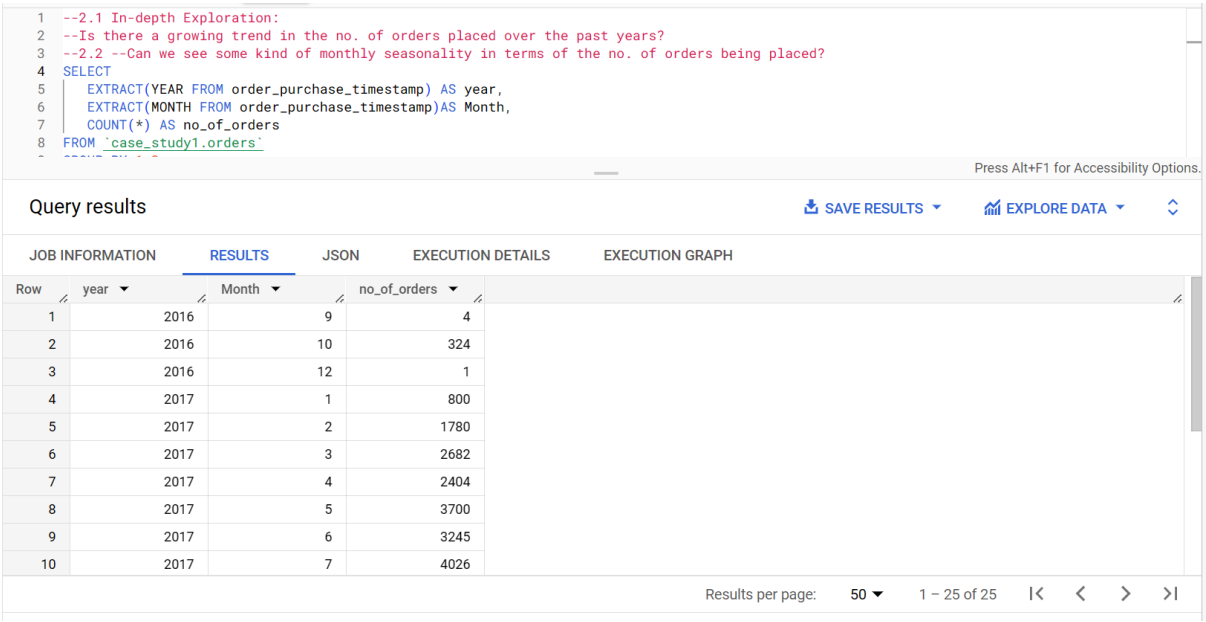
Row	no_of_cities	no_of_states
1	4119	27

INSIGHTS:- ***Sales in winter season is high especially in months of October, November, December and January among which month of November 2017 has the highest sale recorded.***

Q 2.1 Is there a growing trend in the no. of orders placed over the past years?

```sql
SELECT
   EXTRACT (YEAR FROM order_purchase_timestamp) AS year,
   EXTRACT (MONTH FROM order_purchase_timestamp) AS Month,
   COUNT (*) AS no_of_orders
FROM `case_study1.orders`
GROUP BY 1,2
ORDER BY 1,2;
```

--Q 2.2: ***Sales in winter season is high especially in months of October, November, December and January among which month of November 2017 has the highest sale recorded.***



```sql
--2.1 In-depth Exploration:
--Is there a growing trend in the no. of orders placed over the past years?
--2.2 --Can we see some kind of monthly seasonality in terms of the no. of orders being placed?
SELECT
   EXTRACT(YEAR FROM order_purchase_timestamp) AS year,
   EXTRACT(MONTH FROM order_purchase_timestamp)AS Month,
   COUNT(*) AS no_of_orders
FROM `case_study1.orders`
```

Press Alt+F1 for Accessibility Options.

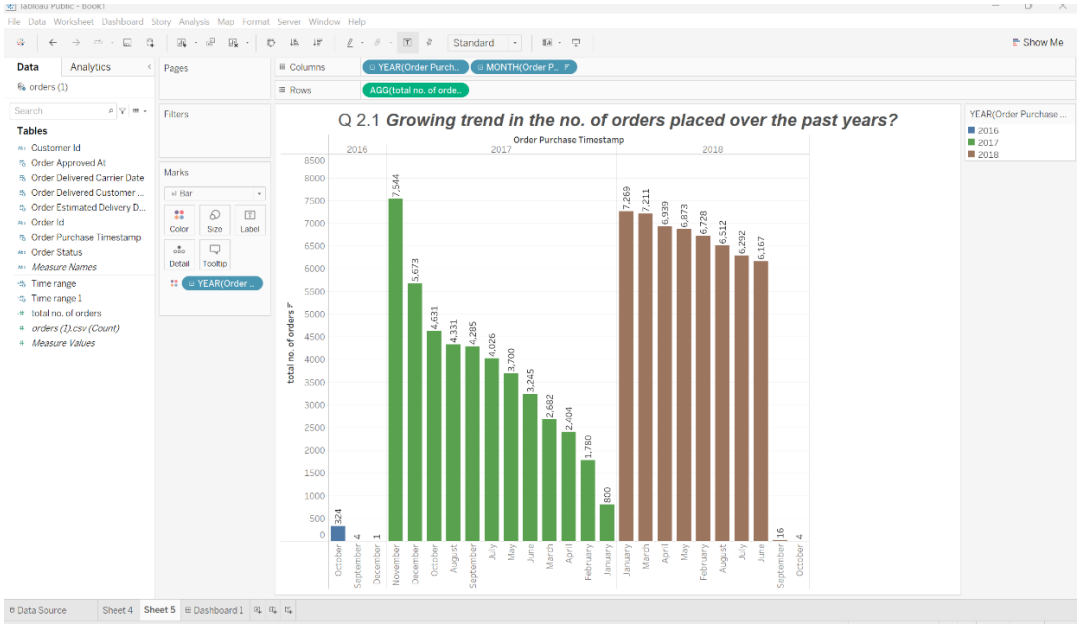
Query results

SAVE RESULTS | EXPLORE DATA

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH

Row	year	Month	no_of_orders
1	2016	9	4
2	2016	10	324
3	2016	12	1
4	2017	1	800
5	2017	2	1780
6	2017	3	2682
7	2017	4	2404
8	2017	5	3700
9	2017	6	3245
10	2017	7	4026

Results per page: 50 | 1 – 25 of 25



Q 2.3 During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)

```
select
  SUM (CASE WHEN hour BETWEEN 0 AND 6 THEN no_of_orders END) AS Dawn,
  SUM (CASE WHEN hour BETWEEN 7 AND 12 THEN no_of_orders END) AS Mornings,
  SUM (CASE WHEN hour BETWEEN 13 AND 18 THEN no_of_orders END) AS Afternoon,
  SUM (CASE WHEN hour BETWEEN 19 AND 23 THEN no_of_orders END) AS Night,
FROM (
SELECT EXTRACT (HOUR FROM order_purchase_timestamp) AS hour, COUNT (DISTINCT
order_id) AS no_of_orders,
FROM `case_study1.orders`
GROUP BY hour
ORDER BY hour)
```

```sql
--Q 2.3
--In-depth Exploration:
--During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)

select
  SUM(CASE WHEN hour BETWEEN 0 AND 6 THEN no_of_orders END)AS Dawn,
  SUM(CASE WHEN hour BETWEEN 7 AND 12 THEN no_of_orders END)AS Mornings,
  SUM(CASE WHEN hour BETWEEN 13 AND 18 THEN no_of_orders END)AS Afternoon,
  SUM(CASE WHEN hour BETWEEN 19 AND 23 THEN no_of_orders END)AS Night,
FROM(
SELECT EXTRACT(HOUR FROM order_purchase_timestamp) AS hour,COUNT(DISTINCT order_id) AS no_of_orders,
FROM `case_study1.orders`
GROUP BY hour
ORDER BY hour)
```

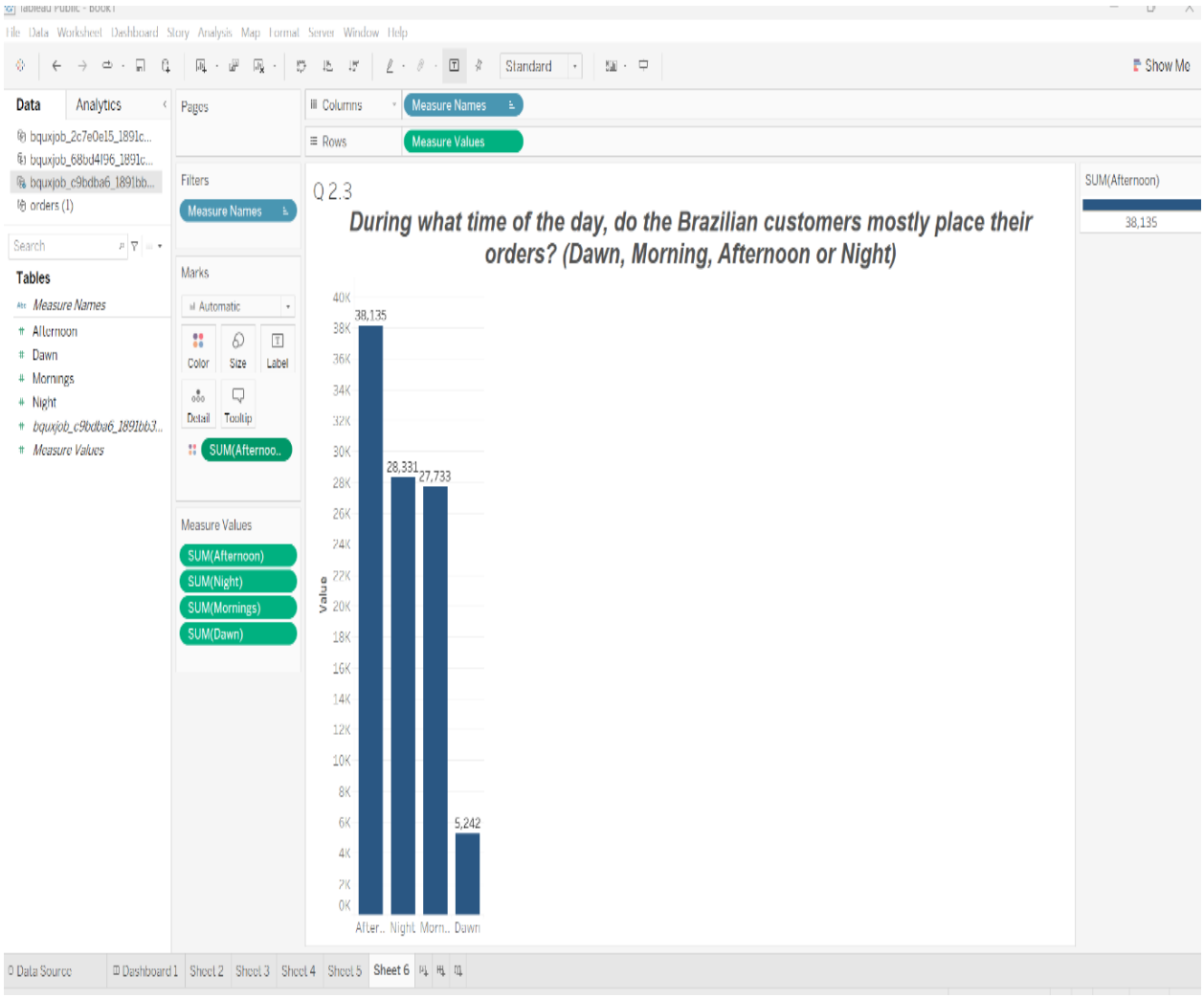
Press Alt+F1 for Accessibility Option

Query results

SAVE RESULTS EXPLORE DATA

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH

Row	Dawn	Mornings	Afternoon	Night
1	5242	27733	38135	28331



INSIGHTS:- *It is observed that Brazilians order during the day (in the afternoon). A Total of 38,135 orders were placed in the afternoon.*

Q 3.1 Get the month-on-month no. of orders placed in each state.

```
SELECT
    EXTRACT (MONTH FROM order_purchase_timestamp) AS month,
    COUNT (*) AS no_of_orders, c.customer_state AS state
FROM `case_study1.orders` AS o JOIN `case_study1.customers` c ON o.customer_id =
c.customer_id
GROUP BY 1,3
ORDER BY no_of_orders DESC;
```

```sql
--Q3.1
--Evolution of E-commerce orders in the Brazil region:
--Get the month on month no. of orders placed in each state.
SELECT
    EXTRACT(MONTH FROM order_purchase_timestamp)AS month,
    COUNT(*)AS no_of_orders,c.customer_state AS state
FROM `case_study1.orders` AS o JOIN `case_study1.customers` c ON o.customer_id = c.customer_id
GROUP BY 1,3
```

Press Alt+F1 for Accessibility Options.

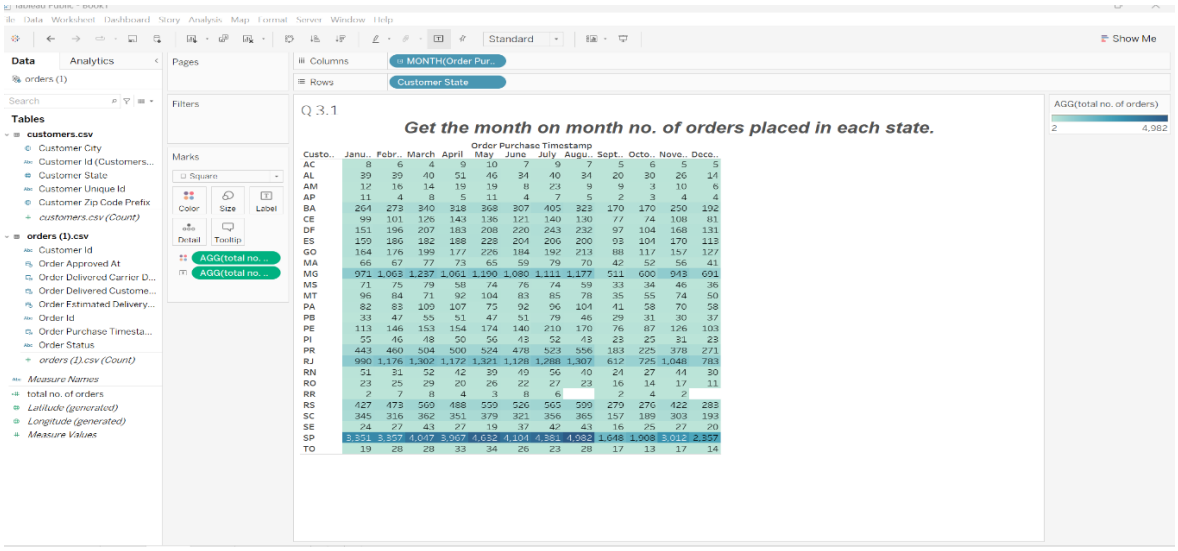
Query results

SAVE RESULTS | EXPLORE DATA

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH

Row	month	no_of_orders	state
1	8	4982	SP
2	5	4632	SP
3	7	4381	SP
4	6	4104	SP
5	3	4047	SP
6	4	3967	SP
7	2	3357	SP
8	1	3351	SP
9	11	3012	SP
10	12	2357	SP

Results per page: 50 | 1 – 50 of 322



INSIGHTS:- *It is observed that number of orders placed was highest in the state SP which is recorded as 4982 in the month of October.*

Q 3.2 How are the customers distributed across all the states?

```
SELECT
   COUNT (customer_id) AS total_customer,customer_state AS state
FROM `case_study1.customers`
GROUP BY 2
ORDER BY 1 DESC;
```

```
1  --Q3.2
2  --Evolution of E-commerce orders in the Brazil region:
3  --How are the customers distributed across all the states?
4  SELECT
5      COUNT(customer_id) AS total_customer,customer_state AS state
6  FROM `case_study1.customers`
7  GROUP BY 2
8  ORDER BY 1 DESC;
```

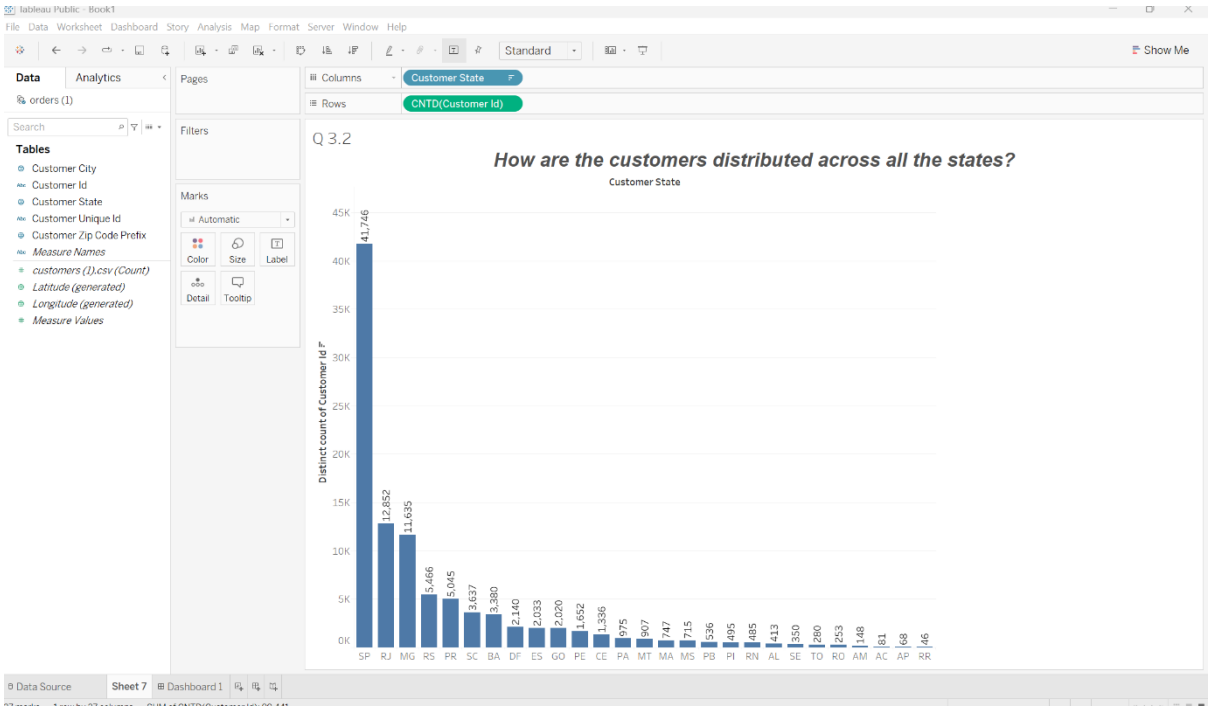
Press Alt+F1 for Accessibility Options

Query results

SAVE RESULTS ▾ EXPLORE DATA ▾

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH

Row	total_customer	state
1	41746	SP
2	12852	RJ
3	11635	MG
4	5466	RS
5	5045	PR
6	3637	SC
7	3380	BA
8	2140	DF
9	2033	ES
10	2020	GO



INSIGHTS:- *State SP has been recorded as highest total number of customers i.e,41746.*

Q 4.1 Get the % increase in the cost of orders from year 2017 to 2018 (include months between Jan to Aug only).

```
WITH yearly_revenue AS (
    SELECT
        EXTRACT (YEAR FROM o.order_purchase_timestamp) AS year,
ROUND(SUM(p.payment_value),2) AS total_revenue
        FROM `case_study1.orders` o JOIN `case_study1.payments` p ON o.order_id =
p.order_id
        WHERE EXTRACT (MONTH FROM o.order_purchase_timestamp)BETWEEN 0 AND 8
        GROUP BY year),
     previous_year_revenue AS(
      SELECT*,
        LAG (total_revenue) OVER (ORDER BY year) AS prev_year_revenue
        FROM yearly_revenue)
SELECT *,
ROUND((total_revenue-prev_year_revenue)/prev_year_revenue*100,2) AS
percent_increase
FROM previous_year_revenue;
```

```sql
--Q 4.1
--Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others.
--Get the % increase in the cost of orders from year 2017 to 2018 (include months between Jan to Aug only).
WITH yearly_revenue AS(
    SELECT
        EXTRACT(YEAR FROM o.order_purchase_timestamp) AS year,ROUND(SUM(p.payment_value),2) AS total_revenue
        FROM `case_study1.orders` o JOIN `case_study1.payments` p ON o.order_id = p.order_id
        WHERE EXTRACT(MONTH FROM o.order_purchase_timestamp)BETWEEN 0 AND 8
        GROUP BY year),
    previous_year_revenue AS(
     SELECT*,
        LAG(total_revenue)OVER(ORDER BY year) AS prev_year_revenue
        FROM yearly_revenue)
SELECT *,
ROUND((total_revenue-prev_year_revenue)/prev_year_revenue*100,2) AS percent_increase
FROM previous_year_revenue;
```

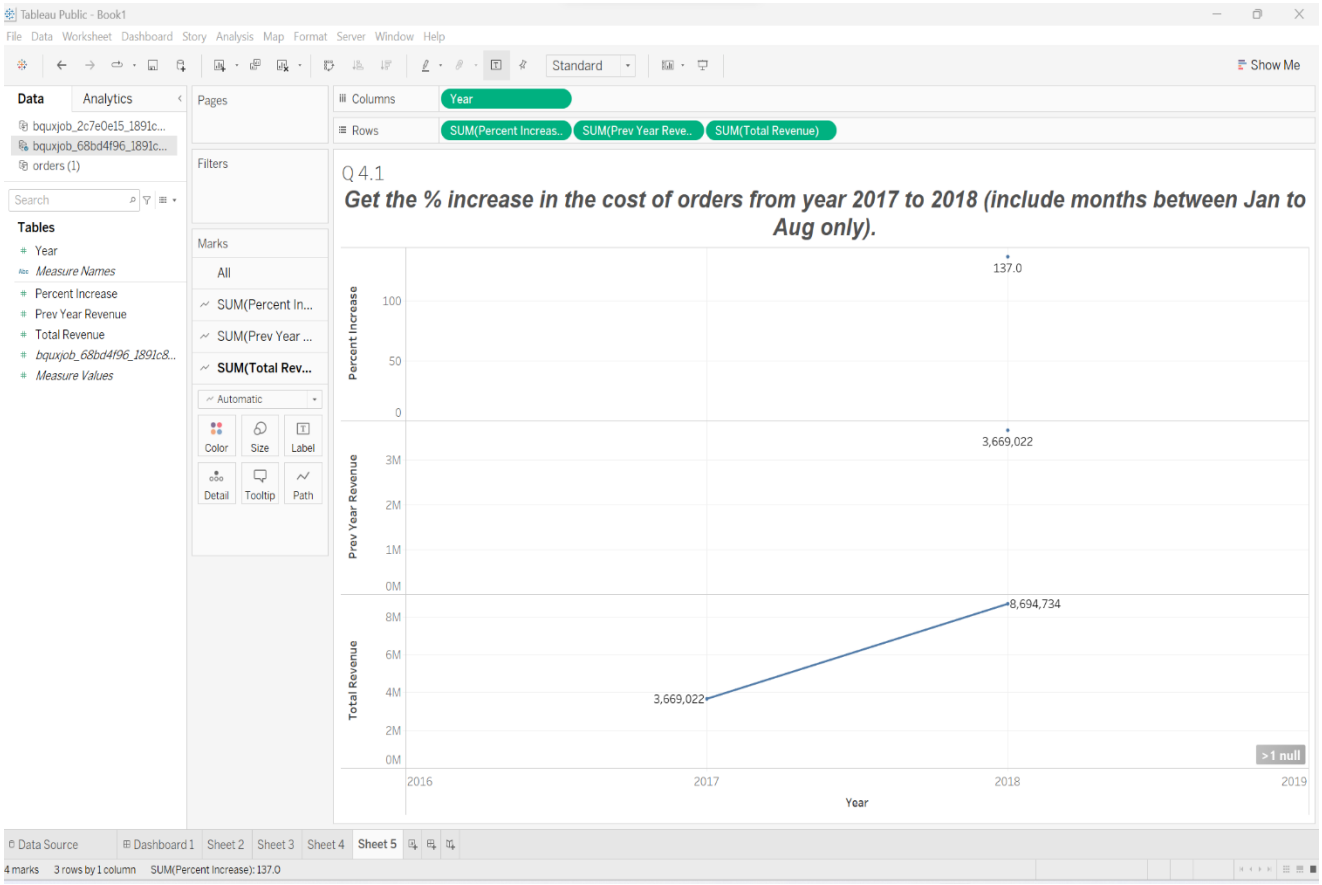
Press Alt+F1 for Accessibility Options.

Query results

SAVE RESULTS ▾ EXPLORE DATA ▾

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH

Row	year	total_revenue	prev_year_revenue	percent_increase
1	2018	8694733.84	3669022.12	136.98
2	2017	3669022.12	null	null



INSIGHTS:- *We observed a year over year, total_ revenue growth of 136.98%, which indicates 1.36 times growth in total_ revenue over last year.*

Q 4.2 Calculate the Total & Average value of order price for each state.

```
WITH order_details AS (
  SELECT ROUND(SUM (od.price),2) AS total_price,COUNT(DISTINCT o.order_id) AS
total_orders,c.customer_state AS state
FROM `case_study1.customers` c JOIN `case_study1.orders` o ON c.customer_id =
o.customer_id
JOIN `case_study1.order_items` od ON o.order_id = od.order_id
GROUP BY state)
SELECT total_price,total_orders, ROUND (total_price/total_orders,2) AS
avg_price,state
FROM order_details
ORDER BY avg_price DESC;
```

```
1  --Q 4.2
2  --Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others.
3  --Calculate the Total & Average value of order price for each state.
4  WITH order_details AS(
5    SELECT ROUND(SUM(od.price),2) AS total_price,COUNT(DISTINCT o.order_id) AS total_orders,c.customer_state AS state
6  FROM `case_study1.customers` c JOIN `case_study1.orders` o ON c.customer_id = o.customer_id
7  JOIN `case_study1.order_items` od ON o.order_id = od.order_id
8  GROUP BY state)
```

Press Alt+F1 for Accessibility Options.

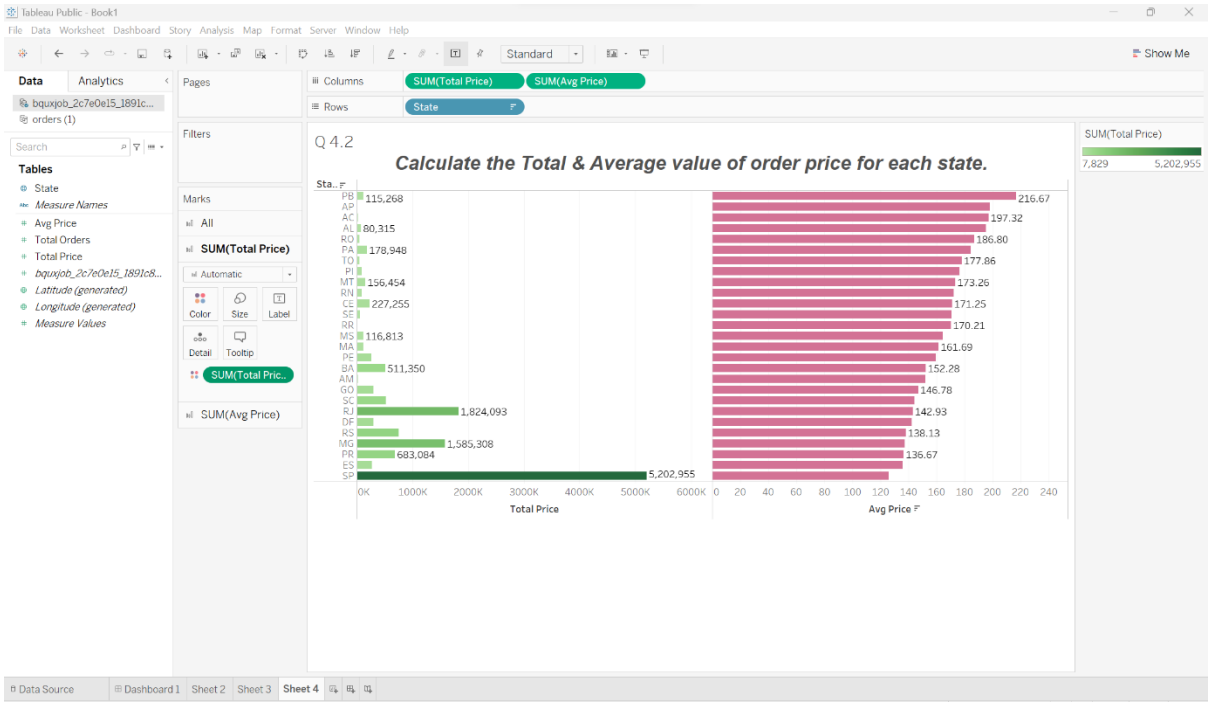
Query results

SAVE RESULTS ▾ EXPLORE DATA ▾

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH

Row	total_price	total_orders	avg_price	state
1	115268.08	532	216.67	PB
2	13474.3	68	198.15	AP
3	15982.95	81	197.32	AC
4	80314.81	411	195.41	AL
5	46140.64	247	186.8	RO
6	178947.81	970	184.48	PA
7	49621.74	279	177.86	TO
8	86914.08	493	176.3	PI
9	156453.53	903	173.26	MT
10	83034.98	482	172.27	RN

Results per page: 50 ▾ 1 – 27 of 27



INSIGHTS:- ***State PB has been recorded with highest avg_ price i.e., 216.67 with total_ orders as 532.***

Q 4.3 Calculate the Total & Average value of order freight for each state.

```
SELECT
    ROUND(SUM (od.freight_value),2) AS
total_freight_value,ROUND(AVG(od.freight_value),2) AS
mean_freight_value,c.customer_state AS state
FROM `case_study1.customers` c JOIN `case_study1.orders` o ON c.customer_id =
o.customer_id
    JOIN `case_study1.order_items` od ON o.order_id = od.order_id
GROUP BY state
ORDER BY 2 DESC;
```

```sql
--Q 4.3
--Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and other
--Calculate the Total & Average value of order freight for each state.
SELECT
    ROUND(SUM(od.freight_value),2) AS total_freight_value,ROUND(AVG(od.freight_value),2) AS mean_freight_value,c.customer_state AS state
FROM `case_study1.customers` c JOIN `case_study1.orders` o ON c.customer_id = o.customer_id
    JOIN `case_study1.order_items` od ON o.order_id = od.order_id
GROUP BY state
```

Press Alt+F1 for Accessibility Options.

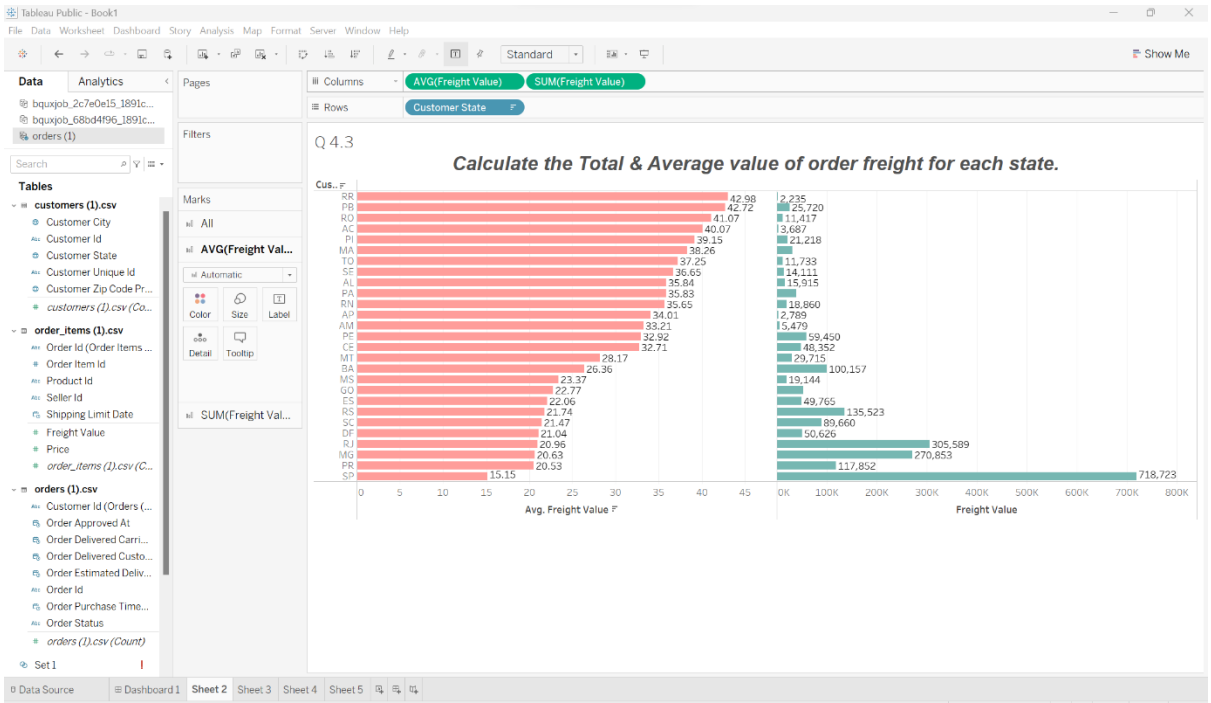
Query results

SAVE RESULTS | EXPLORE DATA

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH

Row	total_freight_value	mean_freight_value	state
1	2235.19	42.98	RR
2	25719.73	42.72	PB
3	11417.38	41.07	RO
4	3686.75	40.07	AC
5	21218.2	39.15	PI
6	31523.77	38.26	MA
7	11732.68	37.25	TO
8	14111.47	36.65	SE
9	15914.59	35.84	AL
10	38699.3	35.83	PA

Results per page: 50 | 1 – 27 of 27



INSIGHTS:- *State RR has highest mean_ freight_ value as 42.98 though total_ freight _value is 2235.19.*

Q 5.1 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.
Do this in a single query.

```
SELECT 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 `case_study1.orders` o
ORDER BY diff_estimated_delivery DESC;
```

```sql
--Q 5.1
--Analysis based on sales, freight and delivery time.
--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.
SELECT 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 `case_study1.orders` o
```

Press Alt+F1 for Accessibility Options.

Query results

SAVE RESULTS ▾ EXPLORE DATA ▾

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH

Row	time_to_deliver	diff_estimated_delivery
1	3	146
2	6	139
3	20	134
4	16	123
5	7	108
6	12	83
7	11	82
8	11	77
9	12	77
10	13	77

Results per page: 50 ▾ 1 – 50 of 99441

INSIGHTS:- *In most cases there is a difference of almost 4 months between order_ estimated_ delivery_ date and order_ delivered_ customer_ date because of which e. commerce may lose its potential customer resulting in drop of number orders.*

Q 5.2 Find out the top 5 states with the highest & lowest average freight value.

```sql
(SELECT c.customer_state,AVG(od.freight_value) AS average_freight_value
FROM `case_study1.order_items` od JOIN `case_study1.orders` o ON od.order_id =
o.order_id JOIN `case_study1.customers` c ON
c.customer_id = o.customer_id
GROUP BY customer_state
ORDER BY average_freight_value DESC
LIMIT 5)
UNION ALL
(SELECT c.customer_state,AVG(od.freight_value) AS average_freight_value
```

```
FROM `case_study1.order_items` od JOIN `case_study1.orders` o ON od.order_id =
o.order_id JOIN `case_study1.customers` c ON
c.customer_id = o.customer_id
GROUP BY customer_state
ORDER BY average_freight_value
LIMIT 5);
```

```
1  --Q 5.2
2  --Analysis based on sales, freight and delivery time.
3  --Find out the top 5 states with the highest & lowest average freight value.
4  (SELECT c.customer_state AS state,ROUND(AVG(od.freight_value),2) AS average_freight_value
5  FROM `case_study1.order_items` od JOIN `case_study1.orders` o ON od.order_id = o.order_id JOIN `case_study1.customers` c ON
6  c.customer_id = o.customer_id
7  GROUP BY customer_state
8  ORDER BY average_freight_value DESC
9  LIMIT 5)
10 UNION ALL
```

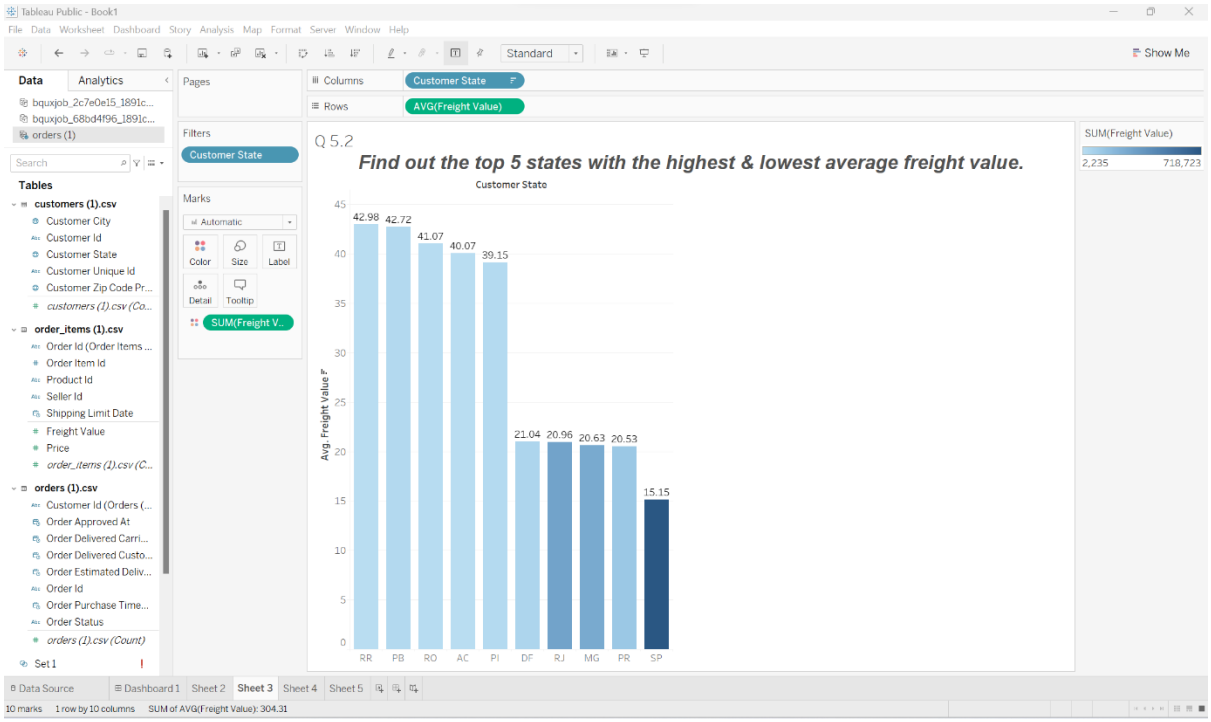
Press Alt+F1 for Accessibility Options

Query results

SAVE RESULTS ▾ EXPLORE DATA ▾

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH

Row	state	average_freight_value
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



Q 5.3 Find out the top 5 states with the highest & lowest average delivery time.

```
(SELECT c.customer_state AS
state,ROUND(AVG(DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp,DA
Y)),2) AS time_to_deliver
FROM `case_study1.orders` o JOIN `case_study1.customers` c ON
c.customer_id = o.customer_id
GROUP BY customer_state
ORDER BY time_to_deliver DESC
LIMIT 5)
UNION ALL
(SELECT c.customer_state AS
state,ROUND(AVG(DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp,DA
Y)),2) AS time_to_deliver
FROM `case_study1.orders` o JOIN `case_study1.customers` c ON
c.customer_id = o.customer_id
GROUP BY customer_state
ORDER BY time_to_deliver
LIMIT 5);
```

```sql
--Q 5.3
--Analysis based on sales, freight and delivery time.
--Find out the top 5 states with the highest & lowest average delivery time.
(SELECT c.customer_state AS state,ROUND(AVG(DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp,DAY)),2) AS time_to_deliver
FROM `case_study1.orders` o JOIN `case_study1.customers` c ON
c.customer_id = o.customer_id
GROUP BY customer_state
ORDER BY time_to_deliver DESC
LIMIT 5)
UNION ALL
```

Press Alt+F1 for Accessibility Options.

Query results

SAVE RESULTS ▾ EXPLORE DATA ▾

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH

Row	state	time_to_deliver
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

Q 5.4 Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.

```sql
SELECT
    c.customer_state,
    ROUND(AVG(DATE_DIFF(order_estimated_delivery_date,order_delivered_customer_date
,DAY)),2) AS diff_estimated_delivery
FROM `case_study1.orders` o JOIN `case_study1.customers` c
    ON c.customer_id = o.customer_id
GROUP BY customer_state
ORDER BY diff_estimated_delivery
LIMIT 5
```

```sql

----Q 5.4
--Analysis based on sales, freight and delivery time.
--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 diff_estimated_delivery
FROM `case_study1.orders` o JOIN `case_study1.customers` c
    ON c.customer_id = o.customer_id
GROUP BY customer_state
ORDER BY diff_estimated_delivery
LIMIT 5
```

Press Alt+F1 for Accessibility Options.

Query results

SAVE RESULTS ▾ EXPLORE DATA ▾

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH

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

Q 6.1 Find the month-on-month no. of orders placed using different payment types.

```sql
SELECT
    EXTRACT (YEAR FROM order_purchase_timestamp) AS year,
    EXTRACT (MONTH FROM order_purchase_timestamp) AS month,
    COUNT(p.order_id)AS no_of_orders,p.payment_type
FROM `case_study1.orders` AS o JOIN `case_study1.payments` AS p ON o.order_id =
p.order_id
GROUP BY year,month,p.payment_type
ORDER BY year,month,p.payment_type;
```

```sql
--Q 6.1
--Analysis based on the payments:
--Find the month on month no. of orders placed using different payment types.
SELECT EXTRACT(YEAR FROM order_purchase_timestamp) AS year,EXTRACT(MONTH FROM order_purchase_timestamp)AS month,
COUNT(p.order_id)AS no_of_orders,p.payment_type
FROM `case_study1.orders` AS o JOIN `case_study1.payments` AS p ON o.order_id = p.order_id
GROUP BY year,month,p.payment_type
ORDER BY year,month,p.payment_type;
```

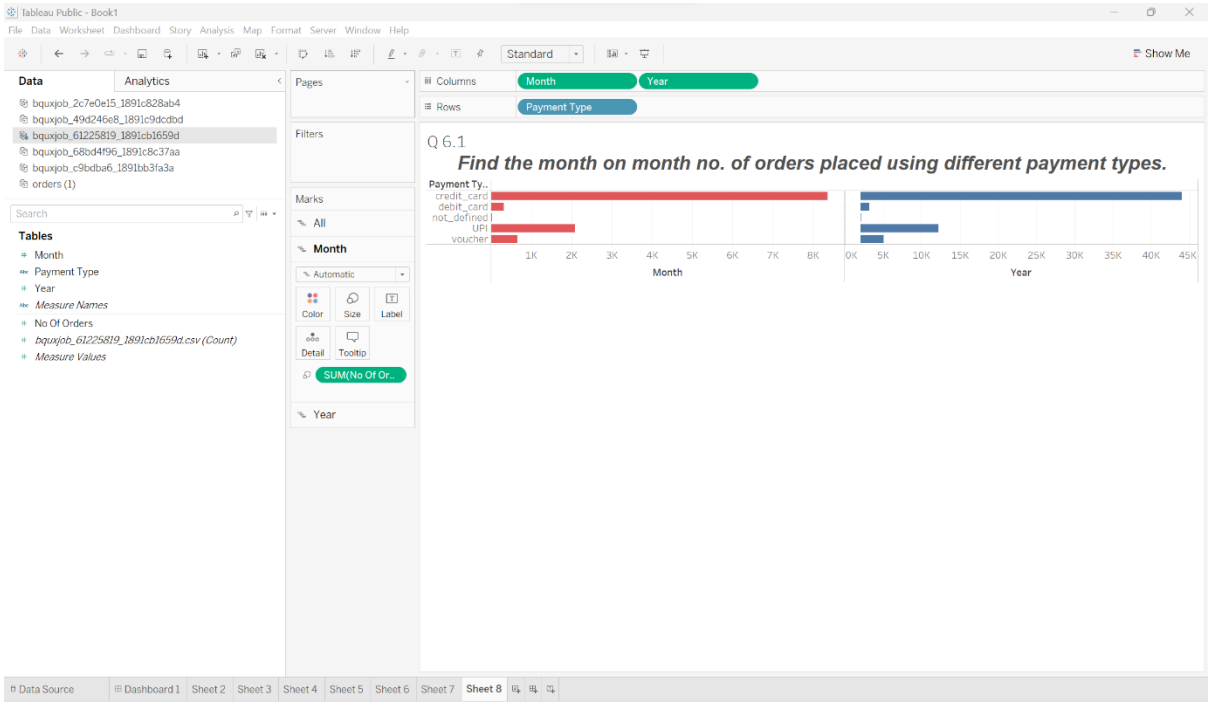
Press Alt+F1 for Accessibility Options

Query results

SAVE RESULTS ▾ EXPLORE DATA ▾

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH

Row	year	month	no_of_orders	payment_type
1	2016	9	3	credit_card
2	2016	10	63	UPI
3	2016	10	254	credit_card
4	2016	10	2	debit_card
5	2016	10	23	voucher
6	2016	12	1	credit_card
7	2017	1	197	UPI
8	2017	1	583	credit_card
9	2017	1	9	debit_card
10	2017	1	61	voucher



INSIGHTS:- *With above query results it clearly shows that customers are using different modes of payment among which maximum payment type mode used by the customer is credit_ card.*

Q 6.2 Find the no. of orders placed on the basis of the payment installments that have been paid.

```
SELECT
   COUNT(o.order_id) AS order_count,
   p.payment_installments
   FROM `case_study1.orders` o JOIN `case_study1.payments` p
   ON o.order_id = p.order_id
 GROUP BY p.payment_installments
 ORDER BY p.payment_installments;
```

```
1  --Q 6.2
2  --Analysis based on the payments:
3  --Find the no. of orders placed on the basis of the payment installments that have been paid.
4  SELECT COUNT(o.order_id) AS order_count,p.payment_installments FROM `case_study1.orders` o JOIN `case_study1.payments` p
5   ON o.order_id = p.order_id
6   GROUP BY p.payment_installments
```

Press Alt+F1 for Accessibility Options

Query results

SAVE RESULTS ▾ EXPLORE DATA ▾

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH

Row	order_count	payment_installment
1	2	0
2	52546	1
3	12413	2
4	10461	3
5	7098	4
6	5239	5
7	3920	6
8	1626	7
9	4268	8
10	644	9
11	5328	10

Results per page: 50 ▾ 1 – 24 of 24

INSIGHTS:- ***Number of orders placed is more on the basis of payment_ instalments 1.***

RECCOMENDATION: -

- From all the data analysis carried out, it has been observed that the expected delivery dates for most orders is big which needs to be reduced.
- Also, it has been observed that there is a difference between actual and expected delivery date which needs to be minimized in order to maintain potential customer.
- Maximum sales were observed in the season of winter so during these period sales should be boosted by offering discounts, promotions and packaging with the help of social media, Advertisements.
- Hiring more delivery agent, Making free shipping.
- Maximum ordered was order during afternoon.
- From above Data Analysis, it shows that customers are using different mode of payment so multiple payment options should be available with instalment facility.