

1. Mention Project back ground
2. Mention that used Big Query to solve this project
  1. Mention the project “Business Case - Target SQL” was created
  2. Under the above project “targetsql” dataset was created
  3. Under the above dataset all the tables (shared by client) were uploaded to perform the analysis



# Section 1

- 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

**Query used:**

```
SELECT column_name, data_type
FROM `Customer.INFORMATION_SCHEMA.COLUMNS`
WHERE table_name = "Customer_info";
```

**Sample Results:**

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

**Comments:**

We have imported the all the dataset to bigquery and we checked the structure of all the dataset and provide example of one snippet with one query

# Section 1

- 1.Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset
- 2. Time period for which the data is given

Query used:

```
SELECT MIN(extract(YEAR FROM order_purchase_timestamp)) as Start_year_data,MAX(extract(YEAR FROM order_purchase_timestamp)) as End_year_data FROM `first-business-case-study.Customer.orders`;
```

Sample Results:

Row	Start_year_data	End_year_data
1	2016	2018

# Section 1

- 1.Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset
- 3. Cities and States of customers ordered during the given period

## Query used:

```
select distinct customer.customer_city,customer.customer_state
from `first-business-case-study.Customer.orders` orders
join `first-business-case-study.Customer.Customer_info` customer on orders.customer_id=customer.customer_id
order by customer.customer_city,customer.customer_state;
```

## Sample Results:

Row	customer_city	customer_state
1	abadia dos dourados	MG
2	abadiania	GO
3	abaete	MG
4	abaetetuba	PA
5	abaiara	CE
6	abaira	BA
7	abare	BA

# Section 2

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?

**Query used:**

```
SELECT count(orders.order_id) as cnt_order,
extract(YEAR from orders.order_purchase_timestamp) as order_year,
extract(MONTH from orders.order_purchase_timestamp) as order_month
FROM `first-business-case-study.Customer.orders` orders
LEFT JOIN `first-business-case-study.Customer.Customer_info` customer
ON orders.customer_id = customer.customer_id
group by order_year,order_month;
```

**Sample Results:**

Row	cnt_order	order_year	order_month
1	4	2016	9
2	324	2016	10
3	1	2016	12
4	800	2017	1
5	1780	2017	2
6	2682	2017	3
7	2404	2017	4

**Comments:**

1. The trend does looks like it's growing.
1. Specially through out the 2017.
2. However, in 2018, the trend looks more or less stagnant
2. There is no indication of any seasonality.

# Section 2

2. In-depth Exploration: (Approach I)
2. What time do Brazilian customers tend to buy (Dawn, Morning, Afternoon or Night)?

## Query used:

```
select extract(year from date(order_time.order_purchase_timestamp)) as year,
extract(month from date(order_time.order_purchase_timestamp)) as month,
order_time.timeslot, count(*) as order_count
from
(select *,
case
when time(orders.order_purchase_timestamp) between '00:00:00' and '05:59:59'
then 'Dawn'
when time(orders.order_purchase_timestamp) between '06:00:00' and '11:59:59'
then 'Morning'
when time(orders.order_purchase_timestamp) between '12:00:00' and '17:59:59'
then 'Afternoon'
when time(orders.order_purchase_timestamp) between '18:00:00' and '23:59:59'
then 'Night'
end as timeslot
from `first-business-case-study.Customer.orders` orders
) order_time
group by year, month, order_time.timeslot
order by year, month, order_time.timeslot
```

## Sample Results:

Row	year	month	timeslot	order_count
1	2016	9	Afternoon	2
2	2016	9	Dawn	1
3	2016	9	Night	1
4	2016	10	Afternoon	125
5	2016	10	Dawn	15
6	2016	10	Morning	84
7	2016	10	Night	100

## Comments:

1. The question was solved using two approaches.
2. This is the first approach where total number for all years & month were found out

Row Labels	Afternoon	Dawn	Morning	Night	Grand Total
Grand Total	38.58%	4.77%	22.37%	34.29%	100.00%

# Section 3

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

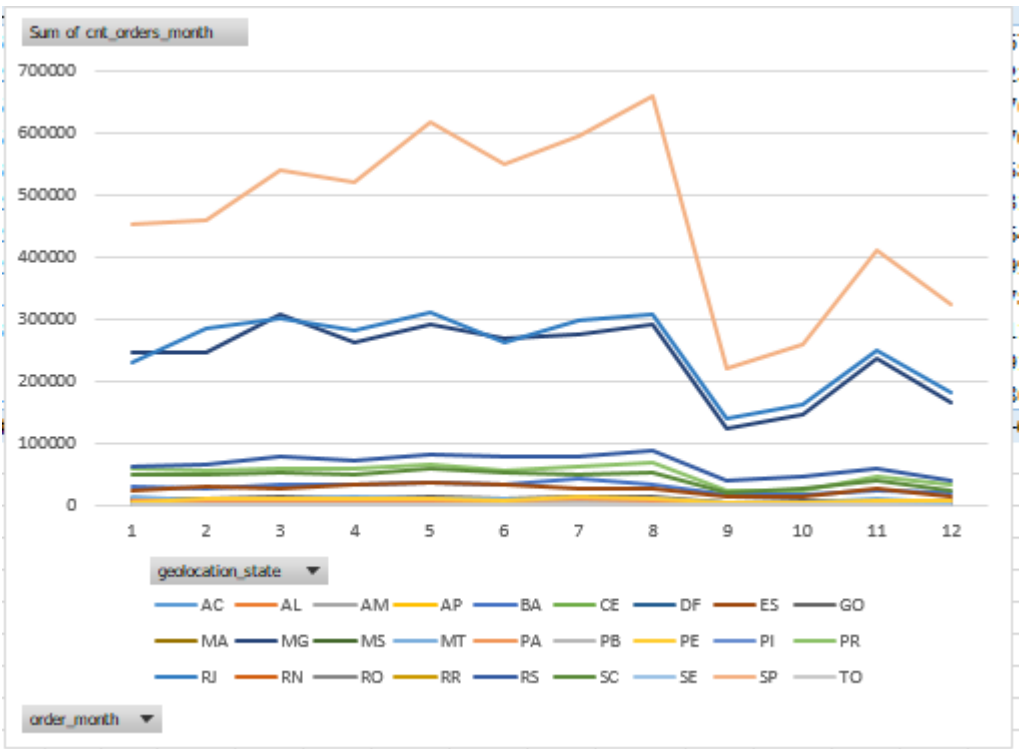
### 1. Get month on month orders by states

#### Query used:

```
SELECT extract(MONTH from orders.order_purchase_timestamp) as order_month, count(*) as cnt_orders_month, geolocation.geolocation_state FROM `first-business-case-study.Customer.orders` orders
INNER JOIN `first-business-case-study.Customer.Customer_info` cust
ON orders.customer_id = cust.customer_id
INNER JOIN `first-business-case-study.Customer.Geolocation` geolocation
ON cust.customer_zip_code_prefix = geolocation.geolocation_zip_code_prefix
group by order_month,geolocation.geolocation_state
order by order_month;
```

#### Sample Result :

Row	order_month	cnt_orders_mon	geolocation_state
1	1	2634	RN
2	1	453515	SP
3	1	246203	MG
4	1	32144	BA
5	1	230923	RJ
6	1	62867	RS
7	1	4008	MA



# Section 3

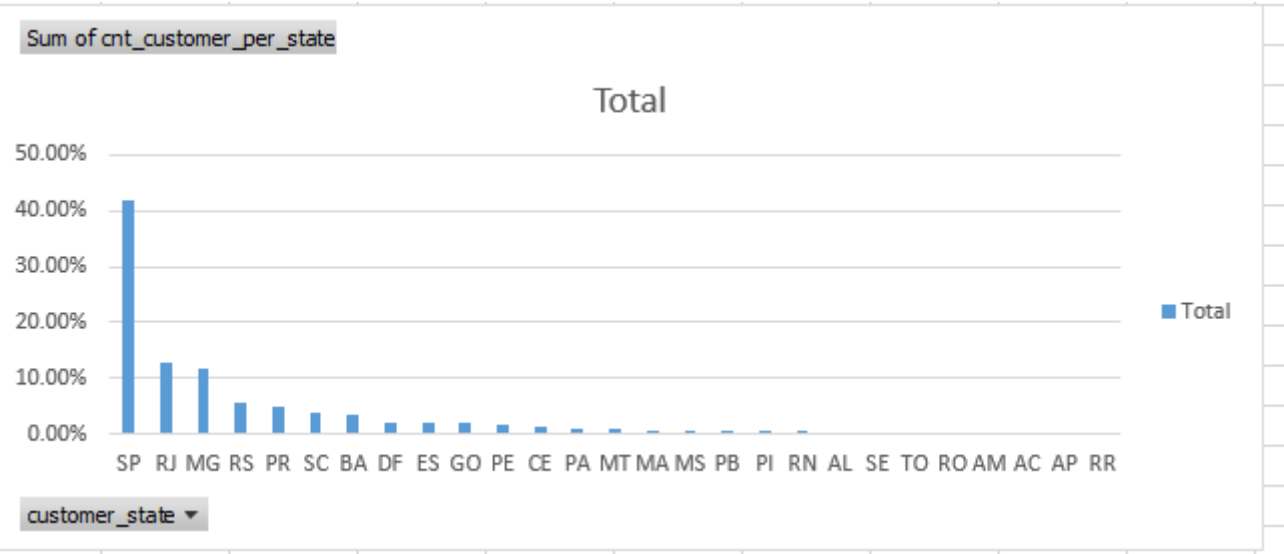
- 3. Evolution of E-commerce orders in the Brazil region:
  - 2. Distribution of customers across the states in Brazil

## Query used:

```
SELECT customer_state, ROUND((count(*)/(SELECT count(*)
FROM `first-business-case-study.Customer.Customer_info`))*100,2) as cnt_customer_per_state FROM `first-business-case-study.Customer.Customer_info` group by customer_state;
```

## Sample Results:

Row	customer_state	cnt_customer_per
1	RN	0.49
2	CE	1.34
3	RS	5.5
4	SC	3.66
5	SP	41.98
6	MG	11.7
7	BA	3.4





# Section 4

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

## Query used: (to obtain cost for each year)

```
WITH payment_2017 as
(SELECT extract(YEAR from orders.order_purchase_timestamp) as orders_year, extract(MONTH from orders.order_purchase_timestamp) as orders_month, ROUND(SUM(payment.payment_value),2) as payment_amount FROM `first-business-case-study.Customer.payments` payment
LEFT JOIN `first-business-case-study.Customer.orders` orders
ON payment.order_id = orders.order_id
where extract(MONTH from orders.order_purchase_timestamp) between 1 and 8 AND extract(YEAR from orders.order_purchase_timestamp)= 2017
group by orders_year, orders_month
order by orders_year, orders_month),
payment_2018 as
(SELECT extract(YEAR from orders.order_purchase_timestamp) as orders_year_next, extract(MONTH from orders.order_purchase_timestamp) as orders_month_next, ROUND(SUM(payment.payment_value),2)
as payment_amount_next FROM `first-business-case-study.Customer.payments` payment
LEFT JOIN `first-business-case-study.Customer.orders` orders
ON payment.order_id = orders.order_id
where extract(MONTH from orders.order_purchase_timestamp) between 1 and 8 AND extract(YEAR from orders.order_purchase_timestamp)= 2018
group by orders_year_next, orders_month_next
order by orders_year_next, orders_month_next)
select *,
ROUND(((payment_2018.payment_amount_next - payment_2017.payment_amount)/payment_2017.payment_amount)*100,2) as percentage_increase from payment_2017
INNER JOIN payment_2018
ON payment_2017.orders_month = payment_2018.orders_month_next
order by payment_2017.orders_month;
```

## Sample Results:

Row	orders_year	orders_month	payment_amount	orders_year_next	orders_month_next	payment_amount_next	percentage_increase
1	2017	1	138488.04	2018	1	1115004.18	705.13
2	2017	2	291908.01	2018	2	992463.34	239.99
3	2017	3	449863.6	2018	3	1159652.12	157.78
4	2017	4	417788.03	2018	4	1160785.48	177.84
5	2017	5	592918.82	2018	5	1153982.15	94.63
6	2017	6	511276.38	2018	6	1023880.5	100.26
7	2017	7	592382.92	2018	7	1066540.75	80.04
8	2017	8	674396.32	2018	8	1022425.32	51.61

# Section 4

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

## Query used: (to obtain cost for each year)

```
select round((cost_2018-cost_2017)*100/cost_2017,2) as pc_inc_in_cost
from
(select extract(year from date(o.order_purchase_timestamp)) as year,
round(sum(p.payment_value),2) as cost_2017
from `first-business-case-study.Customer.orders` o
join `first-business-case-study.Customer.payments` p on p.order_id=o.order_id
where extract(month from date(o.order_purchase_timestamp)) between 1 and 8
group by year
having year=2017)t1,

(select extract(year from date(o.order_purchase_timestamp)) as year,
round(sum(p.payment_value),2) as cost_2018
from `first-business-case-study.Customer..orders` o
join `first-business-case-study.Customer.payments` p on p.order_id=o.order_id
where extract(month from date(o.order_purchase_timestamp)) between 1 and 8
group by year
having year=2018)t2
```

## Sample Results:

Row	pc_inc_in_cost
1	136.98

# Section 4

4. Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others.
2. Mean & Sum of price and freight value by customer state

## Query used:

```
SELECT customer.customer_state, ROUND(avg(order_item.price),2) as mean_price, ROUND(sum(order_item.price),2) as total_price,
ROUND(avg(order_item.freight_value),2) as mean_freight_value, ROUND(sum(order_item.freight_value),2) as total_freight_value
FROM `first-business-case-study.Customer.order_items` order_item
INNER JOIN `first-business-case-study.Customer.orders` orders
ON order_item.order_id = orders.order_id
INNER JOIN `first-business-case-study.Customer.Customer_info` customer
ON orders.customer_id = customer.customer_id
group by customer.customer_state
order by customer.customer_state;
```

## Sample Results:

Row	customer_state	mean_price	total_price	mean_freight_va	total_freight_val
1	AC	173.73	15982.95	40.07	3686.75
2	AL	180.89	80314.81	35.84	15914.59
3	AM	135.5	22356.84	33.21	5478.89
4	AP	164.32	13474.3	34.01	2788.5
5	BA	134.6	511349.99	26.36	100156.68
6	CE	153.76	227254.71	32.71	48351.59
7	DF	125.77	302603.94	21.04	50625.5

# Section 5

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

## Query used:

```
SELECT order_id,  
DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, day) as purchase_delivery_day_diff,  
DATE_DIFF(order_estimated_delivery_date, order_delivered_customer_date, day) as estimated_delivery_day_diff,  
DATE_DIFF(order_estimated_delivery_date, order_purchase_timestamp, day) as estimated_purchase_day_diff  
FROM `first-business-case-study.Customer.orders`  
where order_status='delivered';
```

## Sample Results:

Row	order_id	purchase_delivery_day_diff	estimated_delivery_day_diff	estimated_purchase_day_diff
1	635c894d068ac37e6e03dc54e...	30	1	32
2	3b97562c3aee8bdedcb5c2e45...	32	0	33
3	68f47f50f04c4cb6774570cfde...	29	1	31
4	276e9ec344d3bf029ff83a161c...	43	-4	39
5	54e1a3c2b97fb0809da548a59...	40	-4	36
6	fd04fa4105ee8045f6a0139ca5...	37	-1	35
7	302bb8109d097a9fc6e9cefc5...	33	-5	28

# Section 5

- 5. Analysis on sales, freight and delivery time
  - 2. Find time\_to\_delivery & diff\_estimated\_delivery. Formula for the same given below:
    - 1.  $\text{time\_to\_delivery} = \text{order\_purchase\_timestamp} - \text{order\_delivered\_customer\_date}$
    - 2.  $\text{diff\_estimated\_delivery} = \text{order\_estimated\_delivery\_date} - \text{order\_delivered\_customer\_date}$

```
SELECT DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp,day) as time_to_delivery,
DATE_DIFF(order_estimated_delivery_date, order_delivered_customer_date, day) as diff_estimated_delivery
FROM `first-business-case-study.Customer.orders`
where (order_delivered_customer_date -
order_purchase_timestamp) is not null or (order_estimated_delivery_date -
order_delivered_customer_date) is not null;
```

Row	time_to_delivery	diff_estimated_delivery
1	30	-12
2	30	28
3	35	16
4	30	1
5	32	0
6	29	1
7	43	-4

# Section 5

- 5. Analysis on sales, freight and delivery time
  - 3. Group data by state, take mean of freight\_value, time\_to\_delivery, diff\_estimated\_delivery

```
SELECT customer.customer_state,  
ROUND(AVG(order_items.freight_value),2) as mean_freight_value,  
ROUND(AVG(DATE_DIFF(orders.order_delivered_customer_date, orders.order_purchase_timestamp,d  
ay)),2) as time_to_delivery,  
ROUND(AVG(DATE_DIFF(orders.order_estimated_delivery_date, orders.order_delivered_customer_dat  
e, day)),2) as diff_estimated_delivery  
FROM `first-business-case-study.Customer.orders` orders  
LEFT JOIN `first-business-case-study.Customer.order_items` order_items  
ON orders.order_id = order_items.order_id  
LEFT JOIN `first-business-case-study.Customer.Customer_info` customer_info  
ON orders.customer_id = customer.customer_id  
group by customer.customer_state  
order by customer.customer_state;
```

Row	customer_state	mean_freight_va	time_to_delivery	diff_estimated_c
1	AC	40.07	20.33	20.01
2	AL	35.84	23.99	7.98
3	AM	33.21	25.96	18.98
4	AP	34.01	27.75	17.44
5	BA	26.36	18.77	10.12
6	CE	32.71	20.54	10.26
7	DF	21.04	12.5	11.27

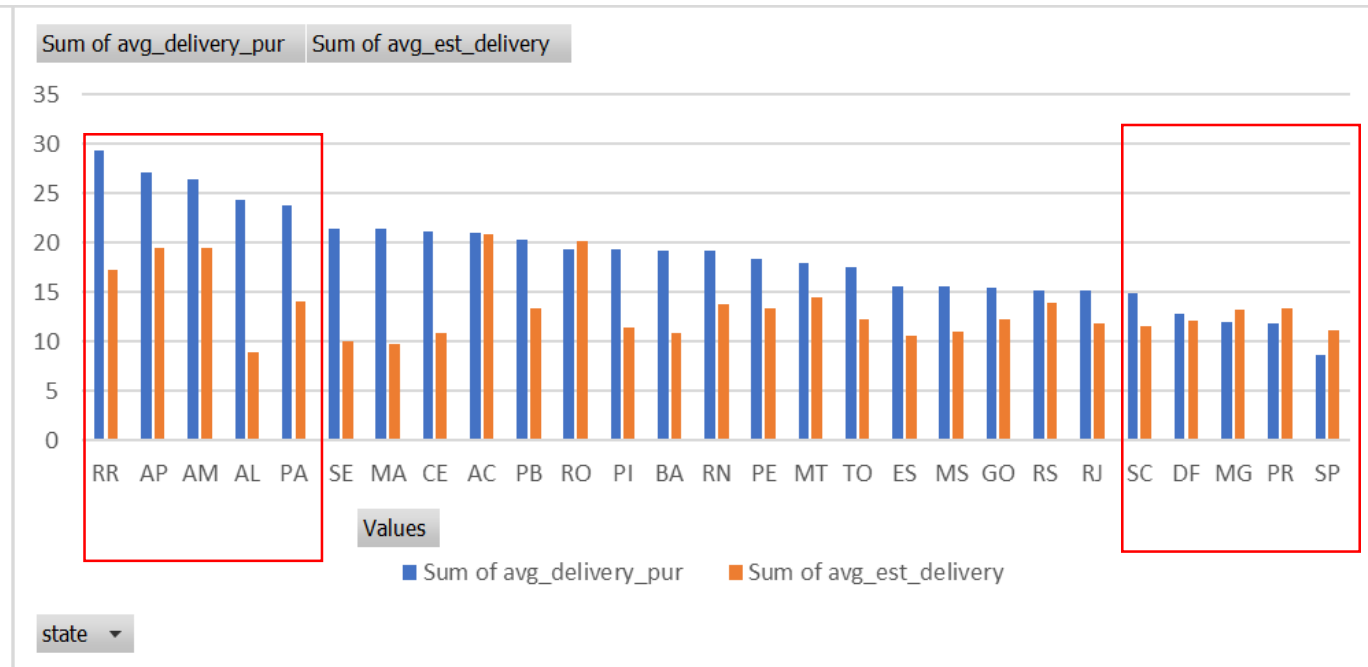
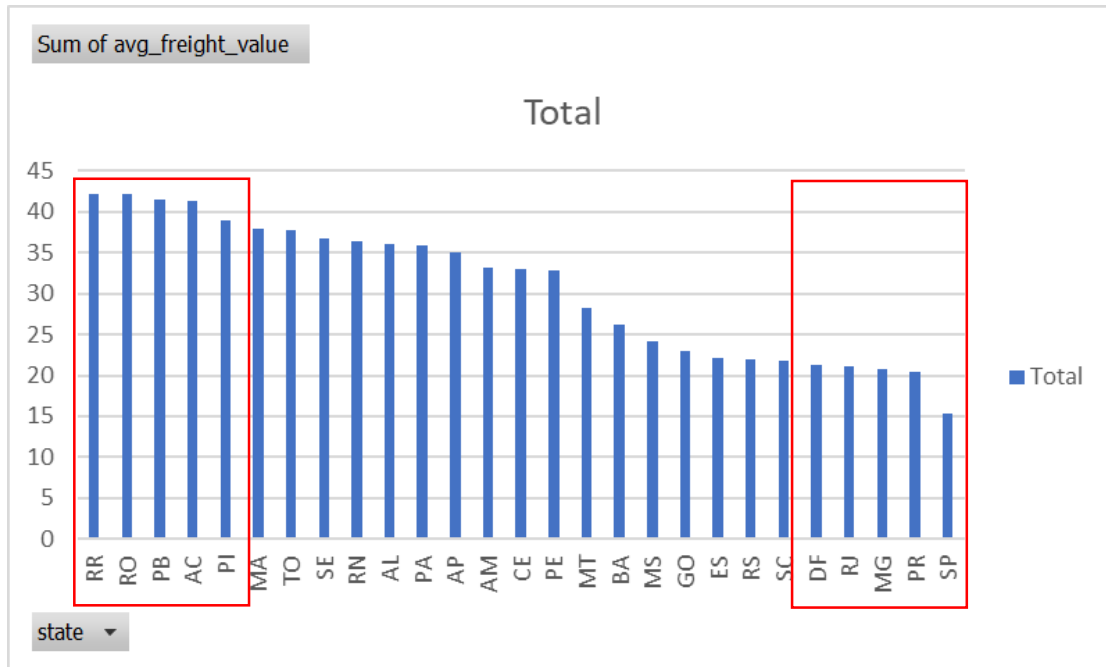
# Section 5

- 5. Analysis on sales, freight and delivery time
  - 4. Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5

state	avg_freight_value	avg_delivery_pur	avg_est_delivery
AC	41.3	20.98	20.89
AL	36.15	24.31	8.87
AM	33.17	26.37	19.54
AP	34.98	27.15	19.53
BA	26.27	19.24	10.89
CE	33.02	21.12	10.92
DF	21.32	12.82	12.16
ES	22.11	15.64	10.58
GO	23.05	15.45	12.24

# Section 5

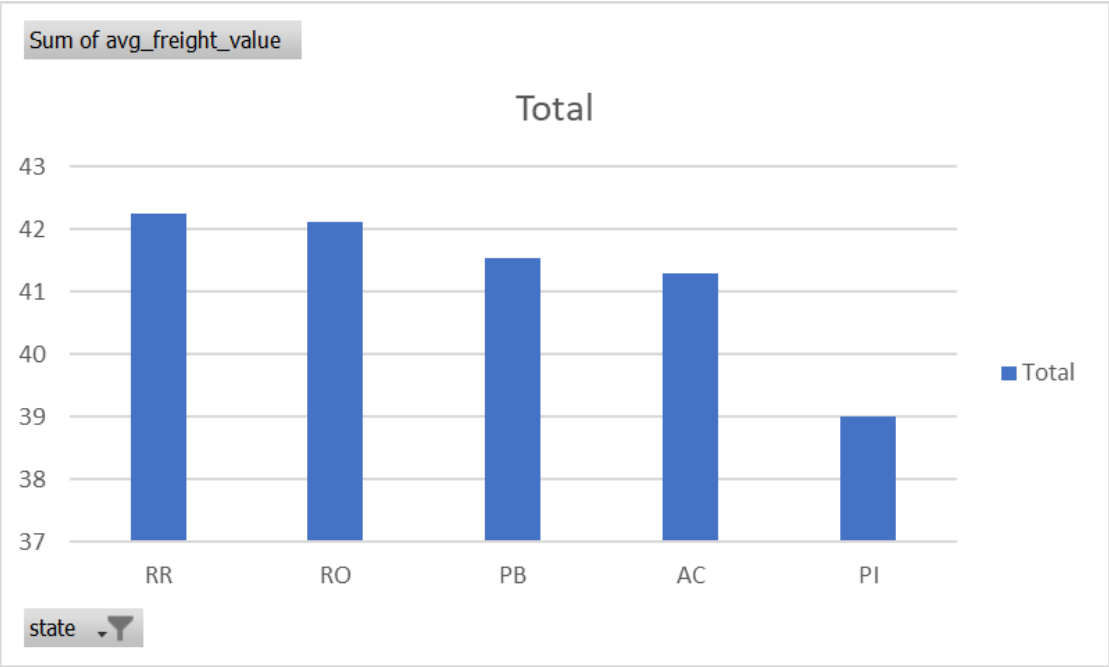
5. Analysis on sales, freight and delivery time
  3. Group data by state, take mean of freight\_value, time\_to\_delivery, diff\_estimated\_delivery
  4. Sort the data to get the following:
  5. Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5
  6. Top 5 states with highest/lowest average time to delivery
  7. Top 5 states where delivery is really fast/ not so fast compared to estimated date





# Section 5

- 5. Analysis on sales, freight and delivery time
  - 3. Group data by state, take mean of freight\_value, time\_to\_delivery, diff\_estimated\_delivery
  - 4. Sort the data to get the following:
  - 5. Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5
  - 6. Top 5 states with highest/lowest average time to delivery
  - 7. Top 5 states where delivery is really fast/ not so fast compared to estimated date



Row Labels	Sum of avg_freight_value
RR	42.26
RO	42.13
PB	41.55
AC	41.3
PI	39.01

# Section 5

- 5. Analysis on sales, freight and delivery time
  - 3. Group data by state, take mean of freight\_value, time\_to\_delivery, diff\_estimated\_delivery
  - 4. Sort the data to get the following:
  - 5. Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5
  - 6. Top 5 states with highest/lowest average time to delivery
  - 7. Top 5 states where delivery is really fast/ not so fast compared to estimated date



Row Labels	Sum of avg_delivery_pur	Sum of avg_est_delivery
RR	29.34	17.29
AP	27.15	19.53
AM	26.37	19.54
AL	24.31	8.87
PA	23.79	14.08