SELF PROJECTS

DATA ANALYTICS

By:
Bura Samshritha
22IM10012

Contents:

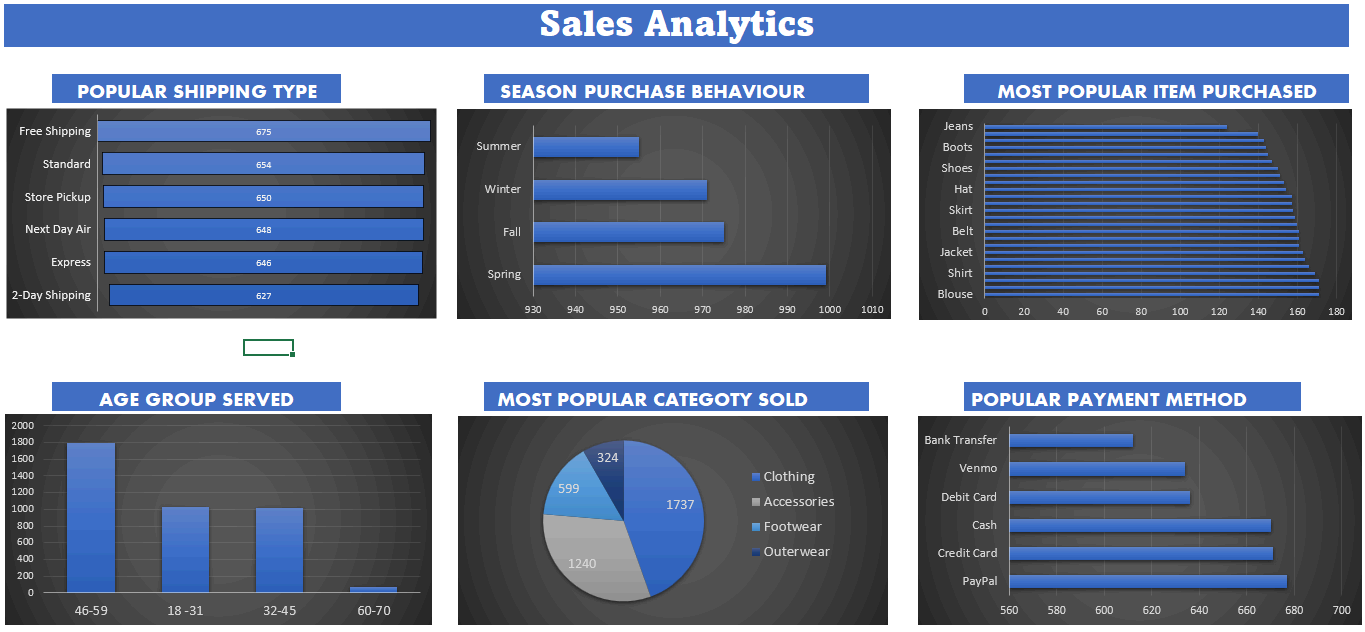
1. Retail Analytics

Tools Used: SQL, Excel
Rows: 3901 - clothing and fashion retail store

Columns:
1. Customer ID
2. Age
3. Gender
4. Item Purchased -
   Blouse
   Sweater
   Jeans
   Sandals
   Sneakers
   Shirt
   Shorts
   Coat
   Handbag
   Shoes
   Dress
   Skirt
   Sunglasses
   Pants
   Jacket
   Hoodie
   Jewelry
   T-shirt
   Scarf
   Hat
   Socks
   Backpack
   Belt
   Boots
   Gloves
5. Category -
   Clothing
   Footwear
   Accessories
6. Purchase Amount (usd)
7. Location
8.Size
9. Color
10. Season -

Winter
Spring
Summer
Fall

11. Review Rating
12. Subscription Status
13. Shipping Type
14. Discount Applied
15. Promo code used
16. Previous PurchasesFortnightly
    Weekly
    Annually
    Quarterly
    Bi-Weekly
    Monthly
    Every 3 Months
17. Payment MethodVenmo
    Cash
    Credit Card
    PayPal
    Bank Transfer
    Debit Card
18. Frequency of Purchases



SQL queries

1. **Gender count and percentage:**

```
WITH total AS (
SELECT
Gender,
COUNT('Customer ID') as client_count
FROM retail_analytics.shopping_trends_updated
GROUP BY 1
ORDER BY 2 DESC
)
SELECT
sum(case when Gender = 'male' then client_count else 0 end) as male_cnt,
sum(case when gender = 'female' then client_count else 0 end) as female_cnt,
round(sum(case when gender = 'male' then client_count else 0 end) /
(sum(case when gender = 'male' then client_count else 0 end) + sum(case when gender =
'female' then client_count else 0 end))*100,2) as pct_male,
round(sum(case when gender = 'female' then client_count else 0 end )/
(sum(case when gender='male' then client_count else 0 end) + sum(case when gender='female'
then client_count else 0 end))*100,2) as pct_female
from total;
```

male_cnt	female_cnt	pct_male	pct_female
2652	1248	68.00	32.00

2. **Count of customer with age divided into bins:**

```
SELECT
    case
        when age between 18 and 31 then '18 -31'
        when age between 32 and 45 then '32-45'
        when age between 46 and 69 then '46-59'
        when age between 60 and 70 then '60-70'
    end as age_groups,
    COUNT('customer ID') as client_count
from retail_analytics.shopping_trends_updated
group by 1
Order by 2 desc;
```

	age_groups	client_count
▶	46-59	1791
	18 -31	1028
	32-45	1014
	60-70	67

3. **Popular category:**

```
select category, count('customer ID') AS client_count
FROM shopping_trends_updated
group by Category
order by 2 desc;
```

	category	client_count
▶	Clothing	1737
	Accessories	1240
	Footwear	599
	Outerwear	324

4. **No. of Distinct Items sold:**

```
SELECT count(distinct `item purchased`) as no_items FROM shopping_trends_updated;
```

	no_items
▶	25

5. **Popular Season:**

```
select season, count('customer ID') AS purchase_count
FROM shopping_trends_updated
group by season
order by 2 desc;
```

	season	purchase_count
▶	Spring	999
	Fall	975
	Winter	971
	Summer	955

6. **Popular Payment Method:**

```
select `Payment Method`, count(`customer ID`) AS purchase_count
FROM shopping_trends_updated
group by 1
order by 2 desc;
```

Payment Method	purchase_count
PayPal	677
Credit Card	671
Cash	670
Debit Card	636
Venmo	634
Bank Transfer	612

7. **Popular shipping type:**

```
select `shipping type`, count(`customer ID`) AS purchase_count
FROM shopping_trends_updated
group by 1
order by 2 desc;
```

shipping type	purchase_count
Free Shipping	675
Standard	654
Store Pickup	650
Next Day Air	648
Express	646
2-Day Shipping	627

8. **Popular payment method among popular age group:**

```
WITH AGE_GROUP_COUNT AS (
  SELECT
    CASE
      WHEN AGE BETWEEN 18 AND 31 THEN '18-30'
      WHEN AGE BETWEEN 32 AND 45 THEN '32-45'
      WHEN AGE BETWEEN 46 AND 59 THEN '46-59'
      WHEN AGE BETWEEN 60 AND 70 THEN '60-70'
    END AS AGE_GROUPS,
    `payment method`,
    COUNT(`CUSTOMER ID`) AS client_count
  FROM shopping_trends_updated
```

```
    GROUP BY AGE_GROUPS, `payment method`
),
MAX_GROUP AS (
    SELECT AGE_GROUPS
    FROM (
        SELECT
            CASE
                WHEN AGE BETWEEN 18 AND 31 THEN '18-30'
                WHEN AGE BETWEEN 32 AND 45 THEN '32-45'
                WHEN AGE BETWEEN 46 AND 59 THEN '46-59'
                WHEN AGE BETWEEN 60 AND 70 THEN '60-70'
            END AS AGE_GROUPS,
            COUNT(`CUSTOMER ID`) AS total_clients
        FROM shopping_trends_updated
        GROUP BY AGE_GROUPS
        ORDER BY total_clients DESC
        LIMIT 1
    ) AS top_group
)

SELECT ag.AGE_GROUPS, ag.`payment method`, ag.client_count
FROM AGE_GROUP_COUNT ag
JOIN MAX_GROUP mg ON ag.AGE_GROUPS = mg.AGE_GROUPS
ORDER BY ag.client_count DESC;
```

AGE_GROUPS	payment method	client_count
46-59	PayPal	207
46-59	Debit Card	180
46-59	Cash	177
46-59	Credit Card	174
46-59	Venmo	166
46-59	Bank Transfer	166

9. Most purchased items by the popular age group:

```
WITH AGE_GROUP AS(
                select  CASE
                WHEN AGE BETWEEN 18 AND 31 THEN '18-30'
        WHEN AGE BETWEEN 32 AND 45 THEN '32-45'
        WHEN AGE BETWEEN 46 AND 59 THEN '46-59'
        WHEN AGE BETWEEN 60 AND 70 THEN '60-70' END
         AS AGE_GROUPS,
```

```
                COUNT(`CUSTOMER ID`) AS client_count,
        `item purchased`
        FROM shopping_trends_updated
        GROUP BY 1,3
        ORDER BY 2 DESC
            )
    --  select count(age_groups)
--      from age_group;
    SELECT AGE_GROUPS, `item purchased`, client_count
    from AGE_GROUP
    WHERE AGE_GROUPS = '46-59';
```

AGE_GROUPS	item purchased	client_count
46-59	Shoes	55
46-59	Sandals	53
46-59	Coat	52
46-59	Shorts	51
46-59	Blouse	48
46-59	Handbag	48
46-59	Sunglasses	48
46-59	Boots	48
46-59	Pants	46
46-59	Skirt	46
46-59	Sweater	45
46-59	Sneakers	44
46-59	Dress	44
46-59	Socks	42
46-59	Shirt	42
46-59	Belt	40
46-59	Jacket	40
46-59	T-shirt	40
46-59	Scarf	39
46-59	Hat	39
46-59	Jewelry	38
46-59	Jeans	35
46-59	Gloves	34
46-59	Hoodie	27
46-59	Backpack	26

10. **Most purchased item by all age groups:**

```
select `item purchased`, count('customer ID') AS client_count
FROM shopping_trends_updated
group by `item purchased`
order by 2 desc
limit 20;
```

item purchased	client_count
Blouse	171
Pants	171
Jewelry	171
Shirt	169
Dress	166
Sweater	164
Jacket	163
Coat	161
Sunglasses	161
Belt	161
Sandals	160
Socks	159
Skirt	158
Shorts	157
Scarf	157
Hat	154
Handbag	153
Hoodie	151
Shoes	150
T-shirt	147

2.Customer Experience analysis

Tools Used - excel.
Performed :

- **1. Data cleaning and Preparation** - converted to table, changed some column types from general to number, decreased decimal for integers, Sorted date column, extracted year and month from Order date column, removed duplicates from whole data(i.e., selected all columns), checked for missing values(found none)
- **2. Formed pivot tables and inserted charts** for **a.** Sales by Category- funnel chart, **b.** Profit gained over time-line chart, **c.** Monthly Sales for Year - line chart, **d.** Top 5 customers making profit - bar chart, **e.** Sales by State - map chart, **f.** Customer count by year- bar chart
- **3. In the new sheet, created the dashboard** - **a.** in view offticked grid lines, formula bar, headings, **b.** Then cut pasted charts and put Heading over each, **c.** Created category and year slicers and reported connections to relevant charts and, **d.** finally made dashboard aesthetic ,eye catching and good looking using page layout, design and format

Sales Analysis - no of rows - 9994, 4 years range- 2014, 15, 16, 17, all 12 months
Columns:

1. Order Date
2. Customer Name
3. State - 50 of USA
4. Category - Furniture , Office Supplies , Technology
5. Sub-category -
   Paper
   Labels
   Storage
   Binders
   Art
   Chairs
   Phones
   Fasteners
   Furnishings
   Accessories
   Envelopes
   Bookcases
   Appliances
   Tables
   Supplies
   Machines
   Copiers
6. Product Name
7. Sales

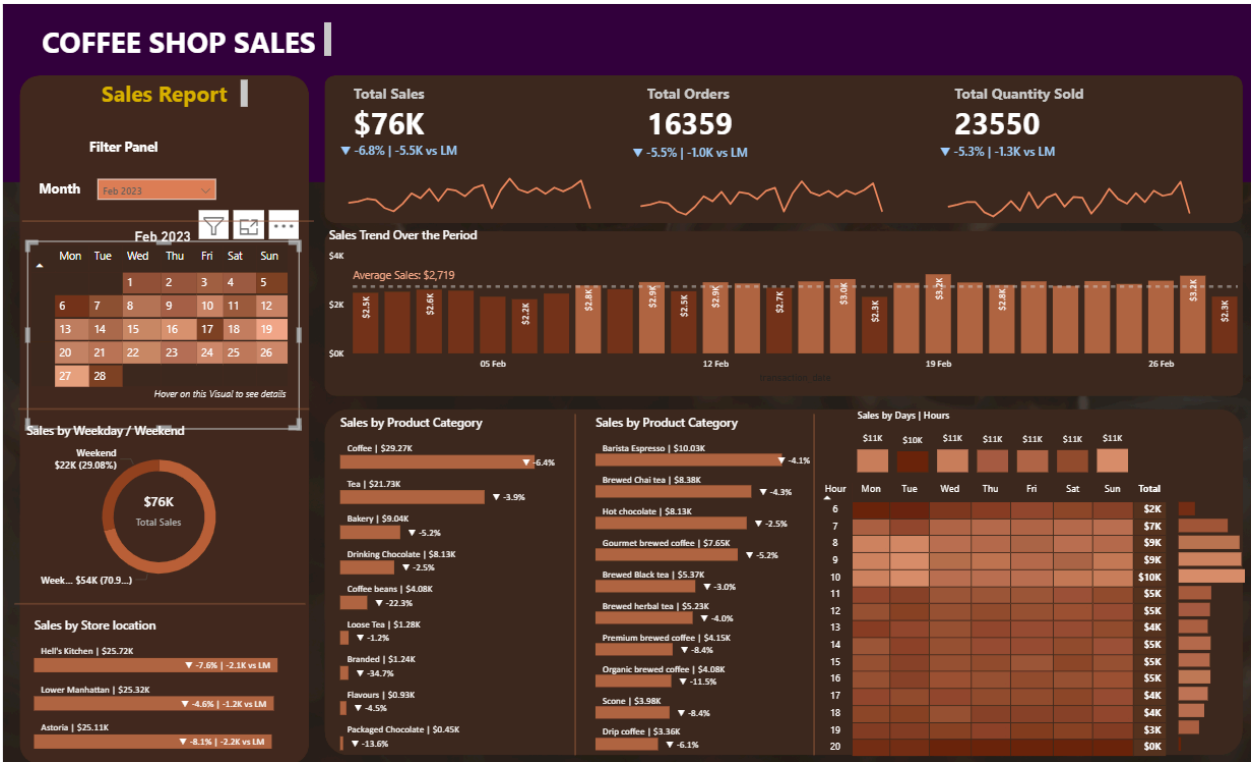
8. Quantity- range: 1 to 14
9. Profit (also includes -ve values, implying loss)

Dashboard:

Coffee Shop Sales Analysis

Tools Used: SQL, Power BI
Rows: 149116 , Time Range - 6 months of 2023 (1st Jan - 30th June)
Columns:

1. transaction_id
2. transaction_date
3. transaction_time
4. transaction_qty
5. store_id
6. store_location
7. product_id
8. unit_price
9. product_category
10. product_type
11. product_detail



Performed:

1. Created Database , imported the dataset into powerBI
2. **Cleaning the dataset** - a. Changed the datatypes of transaction_date & transaction_column from text to DATE and TIME datatype , b. No missing values
3. **Queries :**

DESCRIBE coffee_shop_sales;

Field	Type	Null	Key	Default	Extra
transaction_id	int	YES		NULL	
transaction_date	date	YES		NULL	
transaction_time	time	YES		NULL	
transaction_qty	int	YES		NULL	
store_id	int	YES		NULL	
store_location	text	YES		NULL	
product_id	int	YES		NULL	
unit_price	double	YES		NULL	
product_category	text	YES		NULL	
product_type	text	YES		NULL	
product_detail	text	YES		NULL	

SELECT *
FROM coffee_shop_sales;

transaction_id	transaction_date	transaction_time	transaction_qty	store_id	store_location	product_id	unit_price	product_category	product_type	product_detail
1	2023-01-01	07:06:11	2	5	Lower Manhattan	32	3	Coffee	Gourmet brewed coffee	Ethiopia Rg
2	2023-01-01	07:08:56	2	5	Lower Manhattan	57	3.1	Tea	Brewed Chai tea	Spicy Eye Opener Chai Lg
3	2023-01-01	07:14:04	2	5	Lower Manhattan	59	4.5	Drinking Chocolate	Hot chocolate	Dark chocolate Lg
4	2023-01-01	07:20:24	1	5	Lower Manhattan	22	2	Coffee	Drip coffee	Our Old Time Diner Blend Sm
5	2023-01-01	07:22:41	2	5	Lower Manhattan	57	3.1	Tea	Brewed Chai tea	Spicy Eye Opener Chai Lg
6	2023-01-01	07:22:41	1	5	Lower Manhattan	77	3	Bakery	Scone	Oatmeal Scone
7	2023-01-01	07:25:49	1	5	Lower Manhattan	22	2	Coffee	Drip coffee	Our Old Time Diner Blend Sm
8	2023-01-01	07:33:34	2	5	Lower Manhattan	28	2	Coffee	Gourmet brewed coffee	Columbian Medium Roast Sm
9	2023-01-01	07:39:13	1	5	Lower Manhattan	39	4.25	Coffee	Barista Espresso	Latte Rg
10	2023-01-01	07:39:34	2	5	Lower Manhattan	58	3.5	Drinking Chocolate	Hot chocolate	Dark chocolate Rg
11	2023-01-01	07:43:05	1	5	Lower Manhattan	56	2.55	Tea	Brewed Chai tea	Spicy Eye Opener Chai Rg
12	2023-01-01	07:44:35	2	5	Lower Manhattan	33	3.5	Coffee	Gourmet brewed coffee	Ethiopia Lg
13	2023-01-01	07:45:51	1	5	Lower Manhattan	51	3	Tea	Brewed Black tea	Earl Grey Lg
14	2023-01-01	07:48:19	1	5	Lower Manhattan	57	3.1	Tea	Brewed Chai tea	Spicy Eye Opener Chai Lg

#******Cleaning the data **********

SET SQL_SAFE_UPDATES = 0;
Desc coffee_shop_sales;

ALTER TABLE coffee_shop_sales
MODIFY COLUMN transaction_date DATE;

ALTER TABLE coffee_shop_sales
MODIFY COLUMN transaction_time TIME;

ALTER TABLE coffee_shop_sales
CHANGE COLUMN ï»¿transaction_id transaction_id INT ;

-- ******** Sales Analysis Month Wise**********

```
-- use of concat() and round() function!!
SELECT concat(ROUND(SUM(transaction_qty*unit_price)/1000),'k') as
total_sales
FROM coffee_shop_sales
WHERE MONTH(transaction_date) = 3; -- may month

-- actual full query

SELECT
month(transaction_date) as month_num,
SUM(transaction_qty*unit_price) as total_sales,
ROUND(
                (
                        sum(transaction_qty*unit_price)-
lag(sum(transaction_qty*unit_price))
        over ( order by month(transaction_date))
                )*100
      /
                (
                        lag(sum(transaction_qty*unit_price)) over(order by
month(transaction_date))
                ) , 2)
        as mom_increase_percentage
from coffee_shop_sales
where month(transaction_date) in (4,5)
group by month_nUM
order by month_nUM;
```

month_num	total_sales	mom_increase_percentage
4	118941.08000000106	NULL
5	156727.7600000045	31.77

-- ******** Order Analysis Month Wise**********

```
SELECT  month(transaction_date) as month,
                COUNT(transaction_id) as total_orders,
      ROUND(
                        (
                                COUNT(transaction_id) -
lag(COUNT(transaction_id),1)
            over(order by month(transaction_date))
```

```
                        )*100
        /
        lag(COUNT(transaction_id)) over (order by month(transaction_date))
        ,2)
        as mom_increase_percentage
FROM coffee_shop_sales
WHERE month(transaction_date) in (4,5)
GROUP BY month
order by month;
```

month	total_orders	mom_increase_percentage
4	25335	NULL
5	33527	32.33

```
-- ********** Quantity Analysis Month Wise **********
SELECT MONTH(transaction_date) as month_num,
        sum(transaction_qty) as total_quantity_sold,
        ROUND(
                ( sum(transaction_qty)-lag(sum(transaction_qty),1) over(order by
month(transaction_date)) )*100
      /
      lag(sum(transaction_qty),1) over(order by month(transaction_date))
      ,2)
      as mom_increase_percentage
from coffee_shop_sales
where month(transaction_date) in (4,5)  -- for april,may
group by month_num
order by month_num;
```

month_num	total_quantity_sold	mom_increase_percentage
4	36469	NULL
5	48233	32.26

```
-- **********Calender Heat Map hover over**********

-- in a day, date  of the month , total sales, total_quantity_sold and total orders
select concat(round(sum(transaction_qty*unit_price)/1000, 1),'K' ) as total_sales,
        concat(round(count(transaction_id)/1000,1),'K') as total_orders,
   concat(round(sum(transaction_qty)/1000,1),'K') as total_qty_sold
from coffee_shop_sales
where transaction_date = '2023-03-27';
```

total_sales	total_orders	total_qty_sold
3.7K	0.8K	1.2K

```
-- ************************Weekdays and Weekends sales in a month********************

SELECT SUM(case when dayofweek(transaction_date) in (1,7) then transaction_qty*unit_price else 0 end) as weekend_sales,
        sum(case when dayofweek(transaction_date) not in (1,7) then transaction_qty*unit_price else 0 end) as weekday_sales
from coffee_shop_sales
where month(transaction_date) = '5'; -- may month
-- -----------------or-----------------------
SELECT
        case when dayofweek(transaction_date) in (1,7) then 'weekends'
    else 'weekdays'
    end as day_type,
    concat(round(sum(transaction_qty*unit_price)/1000,1),'K') as total_sales
from coffee_shop_sales
where month(transaction_date) ='5' -- may month
group by
        case when dayofweek(transaction_date) in (1,7) then 'weekends'
    else 'weekdays'
    end ;
```

day_type	total_sales
weekdays	116.6K
weekends	40.1K

– **********sales analysis store wise*******************************

```
select store_location,
        concat(round(sum(transaction_qty*unit_price)/1000,1),'K') as total_sales
from coffee_shop_sales
where month(transaction_date) ='5' -- may month
group by store_location
order by total_sales desc;
```

	store_location	total_sales
▶	Hell's Kitchen	52.6K
	Astoria	52.4K
	Lower Manhattan	51.7K

-- **********Daily Sales ANALYSIS BY AVG LINE**********

```
SELECT day(transaction_date),
        sum(transaction_qty*unit_price) as total_sales,
        case
                when sum(transaction_qty*unit_price) > (select avg(daily_totals)
from
      (
                        select sum(transaction_qty*unit_price) as daily_totals
        from coffee_shop_sales
        where month(transaction_date) = 5
        group by day(transaction_date)
                )
      as daily_avg_sales)
      then 'above_avg' else 'below_avg' end as avg_index
from coffee_shop_sales
where month(transaction_date) = 5 -- may month
group by day(transaction_date)
order by day(transaction_date);

-- or first calculating average_sales
select avg(total_sales) as average_sales
from (
        select sum(transaction_qty*unit_price) as total_sales
   from coffee_shop_sales
   where month(transaction_date)= 5
   group by transaction_date
   ) as daily_sales;

-- daily sales
select
        day(transaction_date) as day_of_month,
   sum(transaction_qty*unit_price) as daily_total_sales
from coffee_shop_sales
where month(transaction_date)= 5
```

group by transaction_date ;-- or group by day(transaction_date) -- same answer

day_of_month	daily_total_sales
1	4731.449999999999
2	4625.499999999997
3	4714.599999999994
4	4589.699999999995
5	4700.999999999997
6	4205.149999999998
7	4542.699999999998
8	5604.209999999995
9	5100.969999999997
10	5256.329999999999
11	4850.059999999996
12	4681.1299999999965
13	5511.529999999999
14	5052.649999999999
15	5384.9800000000005
16	5542.129999999997
17	5418.000000000001
18	5583.470000000001
19	5657.880000000005
20	5519.280000000003
21	5370.810000000003
22	5541.16
23	5242.910000000001
24	5391.45
25	5230.8499999999985
26	5300.949999999998
27	5559.1500000000015
28	4338.649999999998
29	3959.499999999998
30	4835.479999999997
31	4684.129999999993

```
select
        day_of_month,
   total_sales,
   CASE
                WHEN total_sales > avg_sales THEN 'above average'
      WHEN total_sales < avg_sales THEN 'below_avg'
```

```
        ELSE 'equal to avg'
            END AS sales_status
FROM (
            SELECT
                        DAY(transaction_date) AS day_of_month,
        SUM(unit_price*transaction_qty) AS total_sales,
        AVG(SUM(unit_price*transaction_qty)) over() as avg_sales
            FROM
                        coffee_shop_sales
            where
                        month(transaction_date) = 5
            GROUP BY
                        day(transaction_date)
) as sales_data
ORDER BY
            day_of_month;
```

day_of_month	total_sales	sales_status
1	4731.449999999999	below_avg
2	4625.499999999997	below_avg
3	4714.599999999994	below_avg
4	4589.699999999995	below_avg
5	4700.999999999997	below_avg
6	4205.149999999998	below_avg
7	4542.699999999998	below_avg
8	5604.209999999995	above average
9	5100.969999999997	above average
10	5256.329999999999	above average
11	4850.059999999996	below_avg
12	4681.1299999999965	below_avg
13	5511.529999999999	above average
14	5052.649999999999	below_avg
15	5384.9800000000005	above average
16	5542.129999999997	above average
17	5418.000000000001	above average
18	5583.470000000001	above average
19	5657.880000000005	above average
20	5519.280000000003	above average
21	5370.810000000003	above average
22	5541.16	above average
23	5242.910000000001	above average
24	5391.45	above average
25	5230.8499999999985	above average
26	5300.949999999998	above average
27	5559.1500000000015	above average
28	4338.649999999998	below_avg
29	3959.499999999998	below_avg
30	4835.479999999997	below_avg
31	4684.129999999993	below_avg

-- ******** Sales Analysis by Product Category**********

```
SELECT
        product_category,
   sum(transaction_qty*unit_price) as total_sales
FROM coffee_shop_sales
```

```
WHERE MONTH(transaction_date) = 5
GROUP BY product_category
ORDER BY total_sales DESC;
```

product_category	total_sales
Coffee	60362.84999999928
Tea	44539.84999999951
Bakery	18565.519999999997
Drinking Chocolate	16319.75
Coffee beans	8768.949999999997
Branded	2889
Loose Tea	2395.1500000000005
Flavours	1905.5999999999476
Packaged Chocolate	981.0900000000009

-- ******** top 10 product by sales**********

```
SELECT
        product_type,
   sum(transaction_qty*unit_price) as total_sales
FROM coffee_shop_sales
WHERE MONTH(transaction_date) = 5 -- AND product_category = 'Coffee'
GROUP BY product_type
ORDER BY total_sales DESC
LIMIT 10;
```

product_type	total_sales
Barista Espresso	20423.749999999993
Brewed Chai tea	17427.350000000082
Hot chocolate	16319.75
Gourmet brewed coffee	15559.200000000008
Brewed herbal tea	10930
Brewed Black tea	10778
Premium brewed coffee	8739.199999999973
Organic brewed coffee	8350.199999999939
Scone	8305.2799999999
Drip coffee	7290.5

-- ********** Sales ANalysis by Days and Hours Het Map Hover**********

```
SELECT
        sum(unit_price * transaction_qty) AS total_sales,
    SUM(transaction_qty) as total_qty_sold,
    COUNT(*) as total_orders
FROM coffee_shop_sales
WHERE MONTH(transaction_date) = 5
AND DAYOFWEEK(transaction_date) = 1 -- sun
AND hour(transaction_time) = 20 ;-- hour no. 14
```

total_sales	total_qty_sold	total_orders
313.83000000000004	83	58

```
-- ********************SUM of sales in a hour of a month*****************************

SELECT
        hour( transaction_time),
    sum(unit_price * transaction_qty) AS total_sales
FROM coffee_shop_sales
WHERE MONTH(transaction_date) = 5
GROUP BY hour( transaction_time)
ORDER BY hour( transaction_time);
```

hour(transaction_time)	total_sales
6	4912.930000000001
7	14350.680000000037
8	18822.31000000003
9	19145.270000000022
10	19639.13000000001
11	10312.160000000014
12	8869.790000000008
13	9379.210000000008
14	9057.660000000007
15	9525.15000000002
16	9154.310000000012
17	8966.850000000013
18	7679.909999999997
19	6256.469999999997
20	655.9300000000002

-- *********Sum of Sales in day_of_weekof a month**********

```
SELECT
        dayofweek(transaction_date),
   sum(unit_price * transaction_qty) AS total_sales
FROM coffee_shop_sales
WHERE MONTH(transaction_date) = 5
GROUP BY dayofweek(transaction_date)
ORDER BY dayofweek(transaction_date); -- where 1 = sun and 7 = saturday
```

dayofweek(transaction_date)	total_sales
1	19304.810000000056
2	25221.300000000036
3	25346.99000000002
4	25464.509999999966
5	20254.080000000034
6	20340.96
7	20795.109999999975

-- so to get monday name and at first

```
SELECT
        CASE
                WHEN dayofweek(transaction_date) =2 THEN 'Monday'
      WHEN dayofweek(transaction_date) =3 THEN 'Tuesday'
      WHEN dayofweek(transaction_date) =4 THEN 'Wednesday'
      WHEN dayofweek(transaction_date) =5 THEN 'Thursday'
      WHEN dayofweek(transaction_date) =6 THEN 'Friday'
      WHEN dayofweek(transaction_date) =7 THEN 'Saturday'
      ELSE 'Sunday'
        END as day_of_week,
   ROUND(SUM(transaction_qty * unit_price)) as Total_sales
   FROM coffee_shop_sales
   WHERE MONTH(transaction_date) = 6 -- filter for may
   GROUP BY day_of_week -- or (
--                 CASE
--                 WHEN dayofweek(transaction_date) =2 THEN 'Monday'
--       WHEN dayofweek(transaction_date) =3 THEN 'Tuesday'
--       WHEN dayofweek(transaction_date) =4 THEN 'Wednesday'
--       WHEN dayofweek(transaction_date) =5 THEN 'Thursday'
--       WHEN dayofweek(transaction_date) =6 THEN 'Friday'
```

```
--         WHEN dayofweek(transaction_date) =7 THEN 'Saturday'
--         ELSE 'Sunday' END);
           -- to get monday first always add this order by , otherwise the day it first encounters while groupng that comes first
-- ORDER BY
--   CASE DAYOFWEEK(transaction_date)
--     WHEN 2 THEN 1  -- Monday
--     WHEN 3 THEN 2
--     WHEN 4 THEN 3
--     WHEN 5 THEN 4
--     WHEN 6 THEN 5
--     WHEN 7 THEN 6
--     ELSE 7          -- Sunday
--   END;
```

day_of_week	Total_sales
Thursday	27251
Friday	28198
Saturday	22817
Sunday	22185
Monday	22520
Tuesday	22259
Wednesday	21257