

# Snowflake Query Tracker Result

Demand Forecasting & Anomaly Detection

Rossmann Store Sales Data

Generated: 7/31/2025, 2:48:14 PM

Total Queries: 5

## Query #1

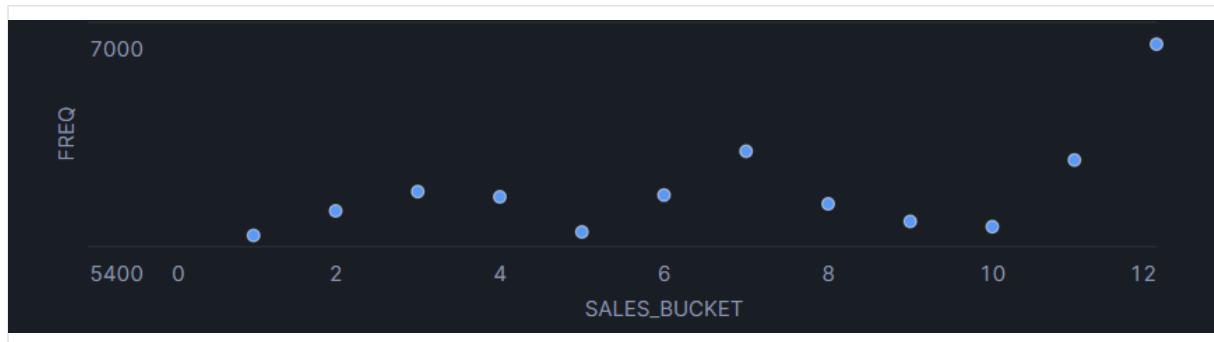
Topic: Exploratory Feature Analysis (Visual EDA)----->4.1: Check average sales by month

### SQL Query:

```
SELECT Month, ROUND(AVG(Sales), 2) AS AvgSales
FROM ROSSMANN_FEATURES
GROUP BY Month
ORDER BY Month;
```

### Result:

# MONTH	# AVGSALES
1	5465.40
2	5645.25
3	5784.58
4	5738.87
5	5489.64
6	5760.96
7	6064.92
8	5693.02
9	5570.25
10	5537.04
11	6008.11
12	6826.61



## Query #2

### Topic: 4.2: Promo vs Non-Promo

#### SQL Query:

```
SELECT Promo, ROUND(AVG(Sales), 2) AS AvgSales
FROM ROSSMANN_FEATURES
GROUP BY Promo;
```

#### Result:

# PROMO	# AVGSALES
1	7991.15
2	4406.05

## Query #3

### Topic: 4.3: School Holiday Effect

#### SQL Query:

```
SELECT SchoolHoliday, ROUND(AVG(Sales), 2) AS AvgSales
FROM ROSSMANN_FEATURES
GROUP BY SchoolHoliday;
```

#### Result:

# SCHOOLHOLIDAY	# AVGSALES
1	6476.52
2	5620.98

## Query #4

### Topic: 4.4: State Holiday Effect

#### SQL Query:

```
SELECT StateHolidayEncoded, ROUND(AVG(Sales), 2) AS AvgSales
FROM ROSSMANN_FEATURES
GROUP BY StateHolidayEncoded;
```

#### Result:

# STATEHOLIDAYENCODED	# AVGSALES
1	290.74
2	214.31
3	168.73
4	5947.48

## Query #5

### Topic: 4.5: Sales by Day of Week

#### SQL Query:

```
SELECT DayOfWeek, ROUND(AVG(Sales), 2) AS AvgSales
FROM ROSSMANN_FEATURES
GROUP BY DayOfWeek
ORDER BY DayOfWeek;
```

#### Result:

# DAYOFWEEK	# AVGSALES
1	7809.04
2	7005.24
3	6555.88
4	6247.58
5	6723.27
6	5847.56
7	204.18