



Project 2: Business Challenge

Collaborators:

Ginosca Alejandro
Javier Dastas
Paola Rivera
Natanael Santiago



Introduction

Retail Data Analytics ¹

- Historical sales data for 45 stores located in different regions
- Each store contains a number of departments.
- Promotional markdown events throughout the year before prominent holidays.

Three tables:

- Features: 8,190 rows x 12 columns
- Sales: 421,570 rows x 5 columns
- Stores: 45 rows x 3 columns



Data Cleaning

To ensure data sets are prepared for analysis, we conducted several data cleaning steps:

1. Handling Missing Values

- a. Features Dataset:

- i. Replaced missing values in Markdown1-Markdown5 with 0, as these columns represent promotional markdowns that were not always applied.
 - ii. Applied forward-fill for missing values in CPI and Unemployment to maintain continuity in regional economic indicators.

- b. Sales and Stores Dataset:

- i. No missing values were detected.



Data Cleaning

To ensure data sets are prepared for analysis, we conducted several data cleaning steps:

2. Checking for Duplicates

- a. Verified that no duplicate rows existed in any of the datasets.

3. Data Type Conversions

- a. Converted the Date columns in the Features and Sales datasets to the appropriate datetime format to facilitate time-based analyses.

4. Data Consistency

- a. Ensure that all datasets align through primary and foreign keys:
 - i. Store column is consistent across all datasets
 - ii. Date column in Features and Sales datasets matches for proper integration.



Data Cleaning

To ensure data sets are prepared for analysis, we conducted several data cleaning steps:

5. Database Preparation

- a. Loaded the cleaned datasets into a SQL database (retail_data) to allow for structured querying and relational analysis.



Data Cleaning Summary

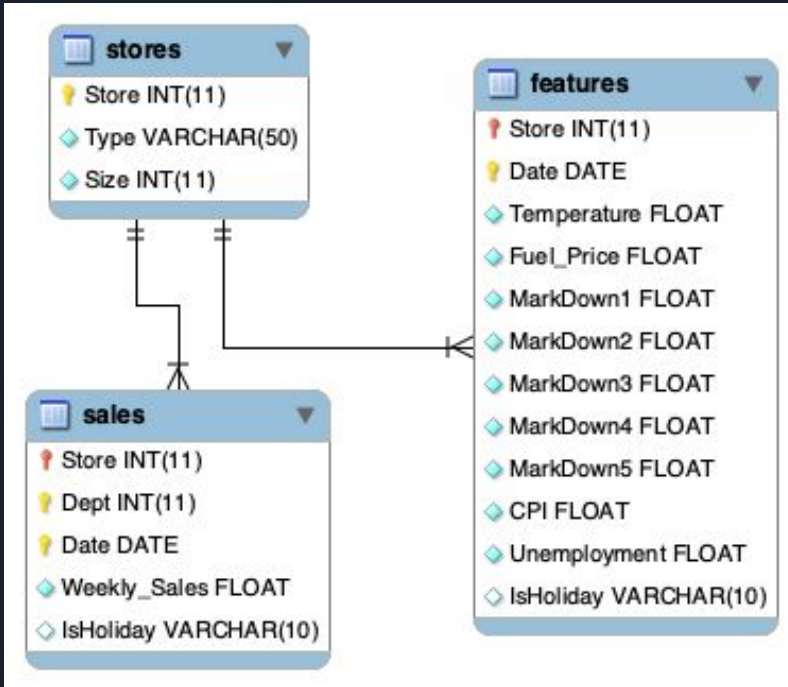
Dataset	Issue	Action Taken
Features	Missing values in Markdown1-Markdown5	Replaced with 0.
Features	Missing values in CPI and Unemployment	Applied forward-fill method.
Sales and Stores	No missing values	Verified completeness.
Features and Sales	Data format inconsistency	Converted to datetime format.
All datasets	Duplicates	Verified no duplicated.



Data Cleaning

- Data Consistency
 - Store column is consistent across all datasets
 - Date column in Features and Sales datasets matches for proper integration.
- Database Preparation
 - Loaded the cleaned datasets into a SQL database (retail_data) to allow for structured querying and relational analysis.

Database Schema



Relationships:

- The features table is related to the stores and sales tables through the Store field.

Data volume:

- features: 8,190 rows.
- sales: 421,570 rows.
- stores: 45 rows.

Business Questions



sales-data-set

Store	Dept	Date	Weekly_Sales	IsHoliday
1	1	05/02/2010	24924.5	FALSE
1	1	12/02/2010	46039.49	TRUE
1	1	19/02/2010	41595.55	FALSE
1	1	26/02/2010	19403.54	FALSE
1	1	05/03/2010	21827.9	FALSE
1	1	12/03/2010	21043.39	FALSE
1	1	19/03/2010	22136.64	FALSE
1	1	26/03/2010	26229.21	FALSE
1	1	02/04/2010	57258.43	FALSE
1	1	09/04/2010	42960.91	FALSE
1	1	16/04/2010	17596.96	FALSE
1	1	23/04/2010	16145.35	FALSE
1	1	30/04/2010	16555.11	FALSE
1	1	07/05/2010	17413.94	FALSE
1	1	14/05/2010	18926.74	FALSE
1	1	21/05/2010	14773.04	FALSE
1	1	28/05/2010	15580.43	FALSE
1	1	04/06/2010	17558.09	FALSE
1	1	11/06/2010	16637.62	FALSE
1	1	18/06/2010	16216.27	FALSE

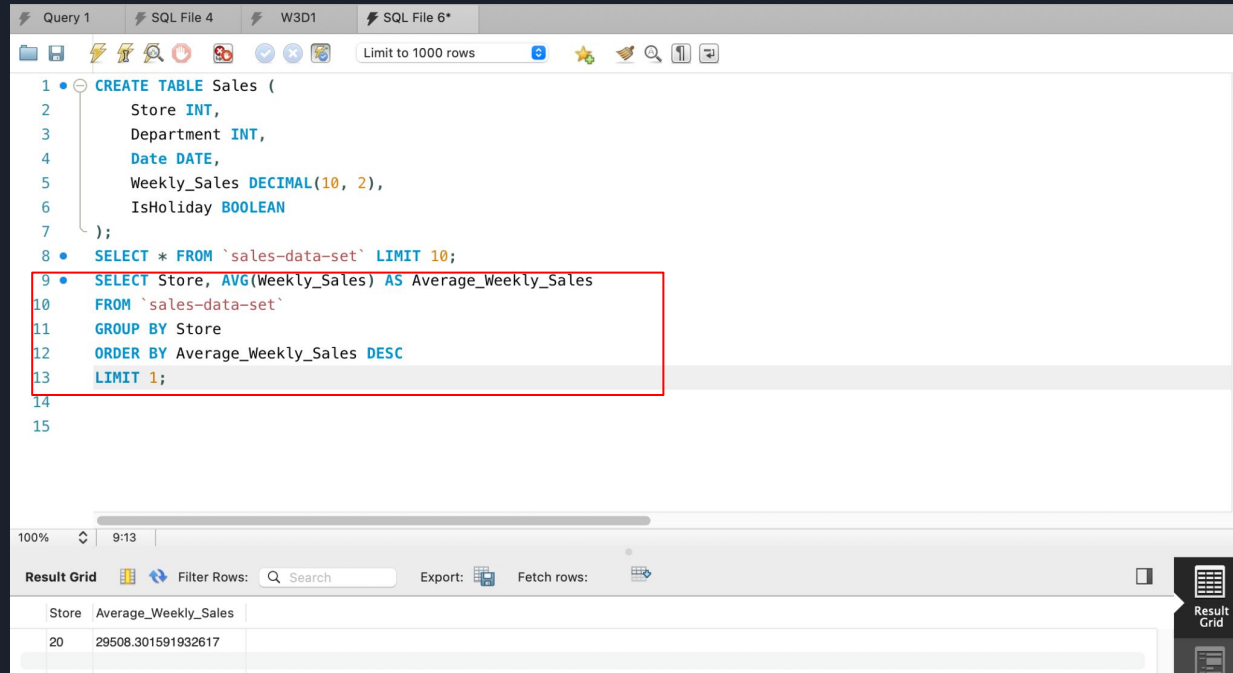
Sales:

Historical sales data, which covers to 2010-02-05 to 2012-11-01. Within this tab you will find the following fields:

- Store - the store number
- Dept - the department number
- Date - the week
- Weekly_Sales - sales for the given department in the given store
- IsHoliday - whether the week is a special holiday week

Overall Sales:

Which store has the highest average weekly sales?



The screenshot shows a SQL IDE interface with a query editor and a result grid. The query editor contains the following SQL code:

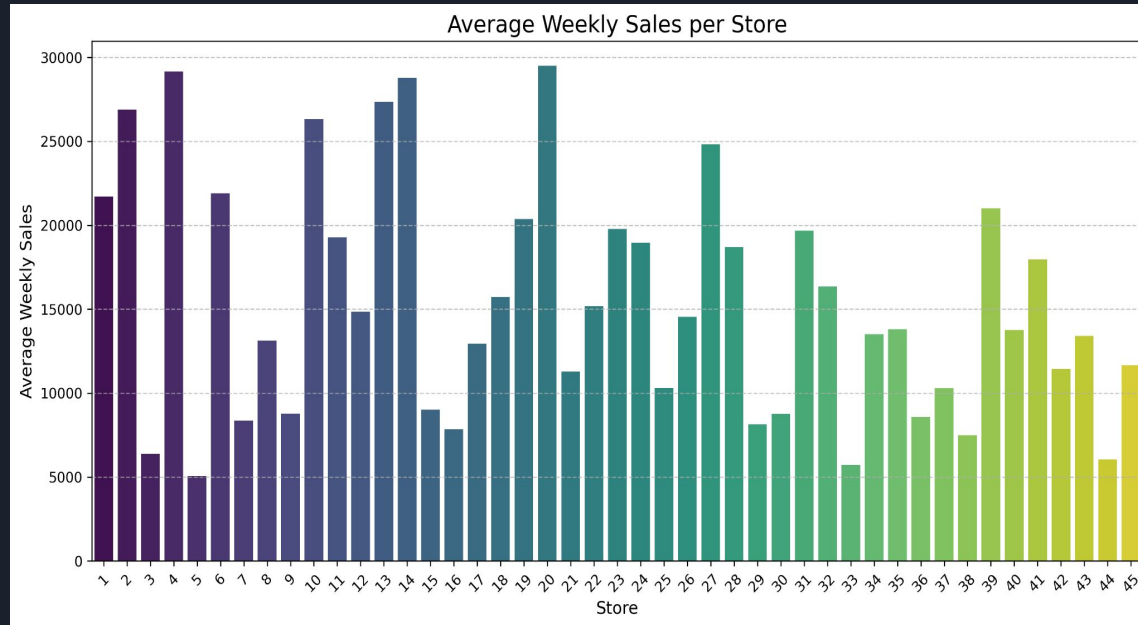
```
1 CREATE TABLE Sales (  
2   Store INT,  
3   Department INT,  
4   Date DATE,  
5   Weekly_Sales DECIMAL(10, 2),  
6   IsHoliday BOOLEAN  
7 );  
8 SELECT * FROM `sales-data-set` LIMIT 10;  
9 SELECT Store, AVG(Weekly_Sales) AS Average_Weekly_Sales  
10 FROM `sales-data-set`  
11 GROUP BY Store  
12 ORDER BY Average_Weekly_Sales DESC  
13 LIMIT 1;  
14  
15
```

The query starting at line 9 is highlighted with a red box. The result grid at the bottom shows the following data:

Store	Average_Weekly_Sales
20	29508.301591932617

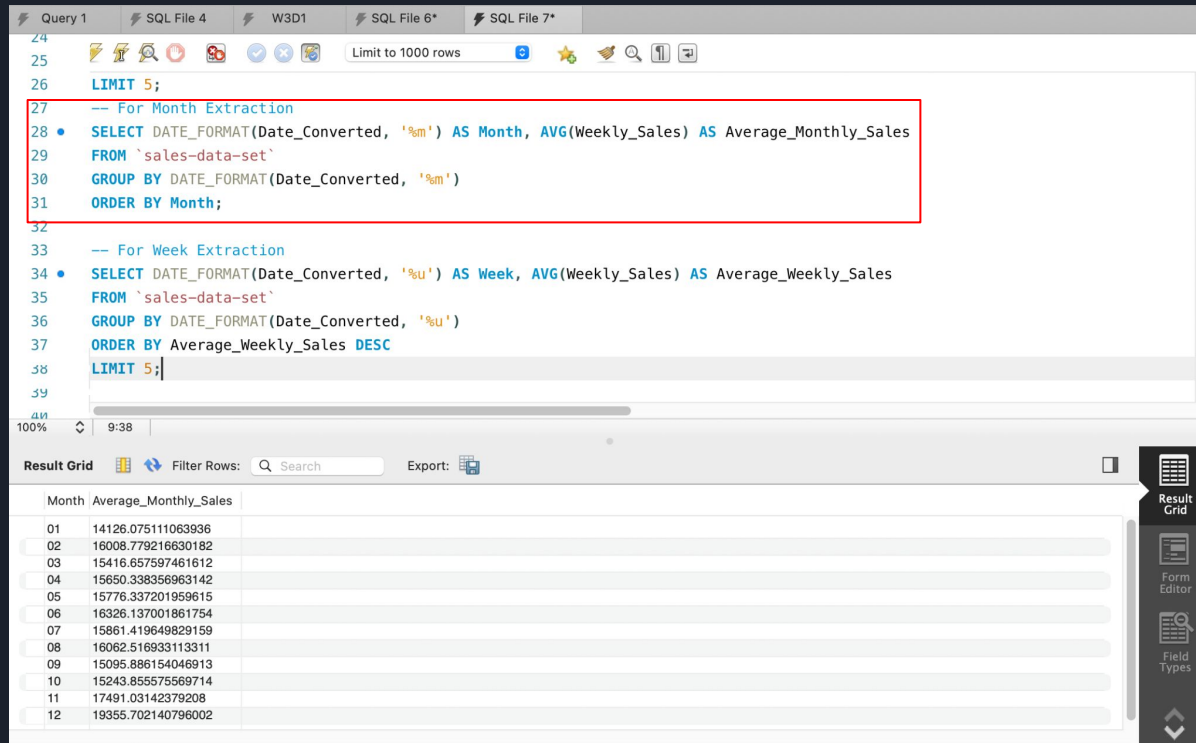
Overall Sales:

Which store has the highest average weekly sales?



Temporal Trends:

What is the average sales trend by month?



The screenshot displays a SQL IDE interface with a query editor and a result grid. The query editor shows two SQL queries. The first query, highlighted with a red box, is for monthly sales extraction. The second query is for weekly sales extraction. The result grid at the bottom shows the output of the first query, displaying 12 rows of data with columns for Month and Average_Monthly_Sales.

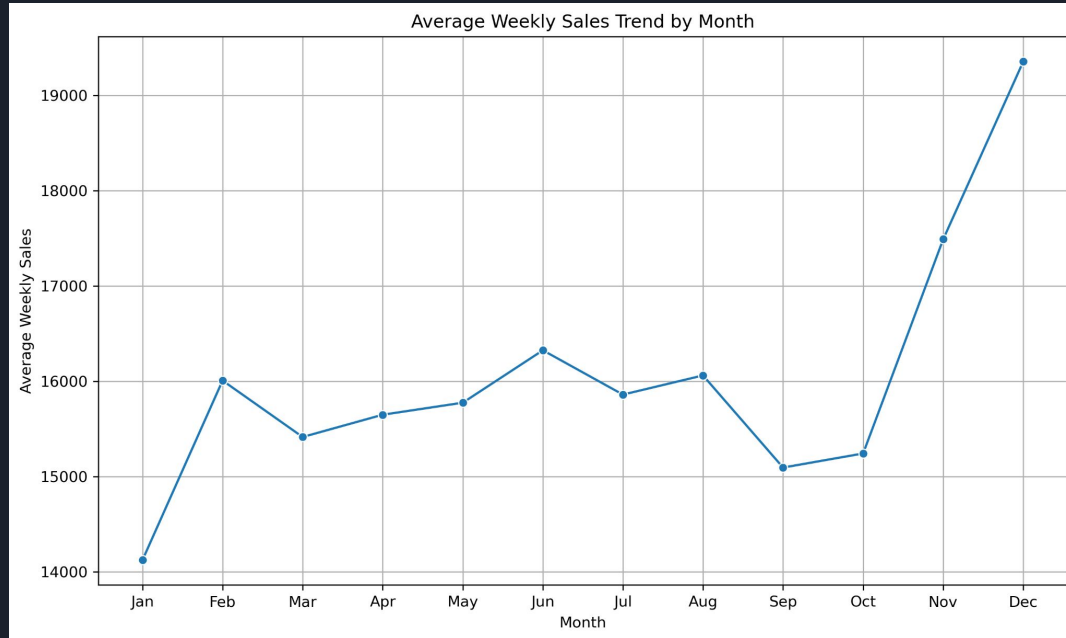
```
24
25
26 LIMIT 5;
27 -- For Month Extraction
28 • SELECT DATE_FORMAT(Date_Converted, '%m') AS Month, AVG(Weekly_Sales) AS Average_Monthly_Sales
29 FROM `sales-data-set`
30 GROUP BY DATE_FORMAT(Date_Converted, '%m')
31 ORDER BY Month;
32
33 -- For Week Extraction
34 • SELECT DATE_FORMAT(Date_Converted, '%u') AS Week, AVG(Weekly_Sales) AS Average_Weekly_Sales
35 FROM `sales-data-set`
36 GROUP BY DATE_FORMAT(Date_Converted, '%u')
37 ORDER BY Average_Weekly_Sales DESC
38 LIMIT 5;
39
```

Result Grid

	Month	Average_Monthly_Sales
01	14126.075111063936	
02	16008.779216630182	
03	15416.657597461612	
04	15650.338356963142	
05	15776.337201959615	
06	16326.137001861754	
07	15861.419649829159	
08	16062.516933113311	
09	15095.886154046913	
10	15243.855575569714	
11	17491.03142379208	
12	19355.702140796002	

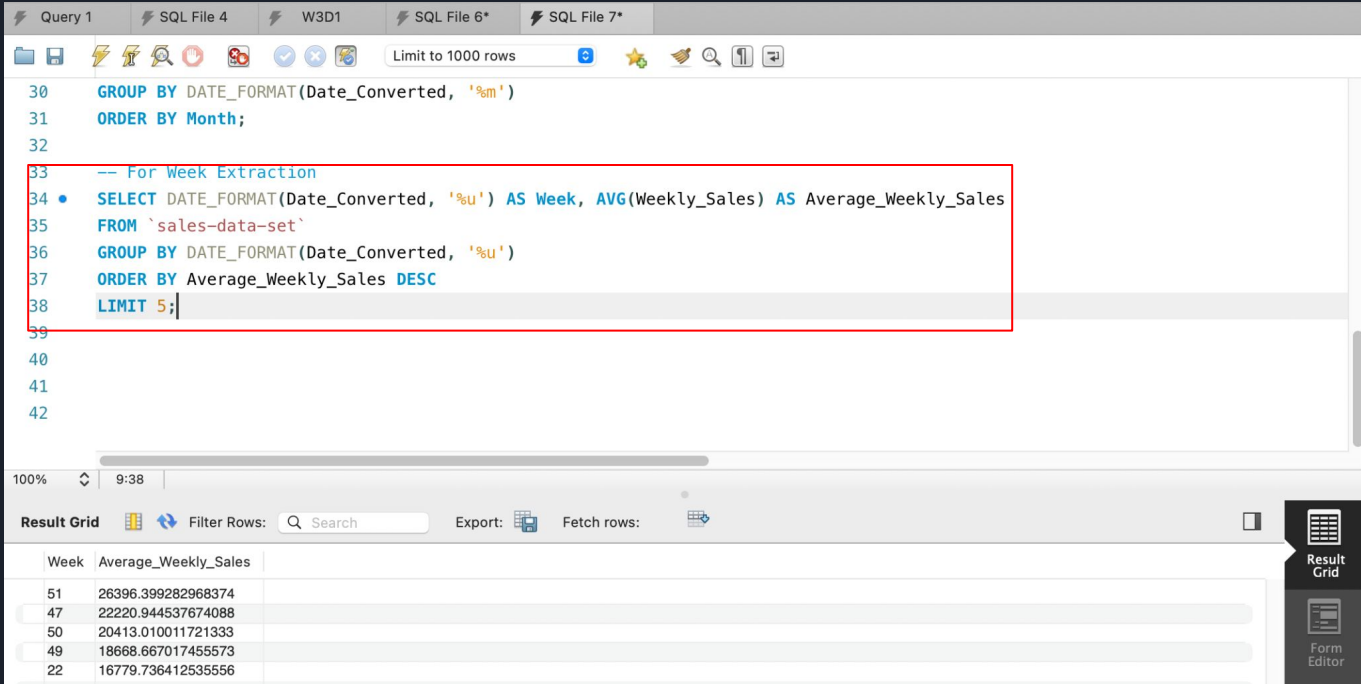
Temporal Trends:

What is the average sales trend by month?



Seasonal Patterns:

Which weeks of the year have the highest average sales?



The screenshot shows a SQL IDE interface with a query editor and a result grid. The query is designed to extract the top 5 weeks with the highest average weekly sales from a dataset named 'sales-data-set'.

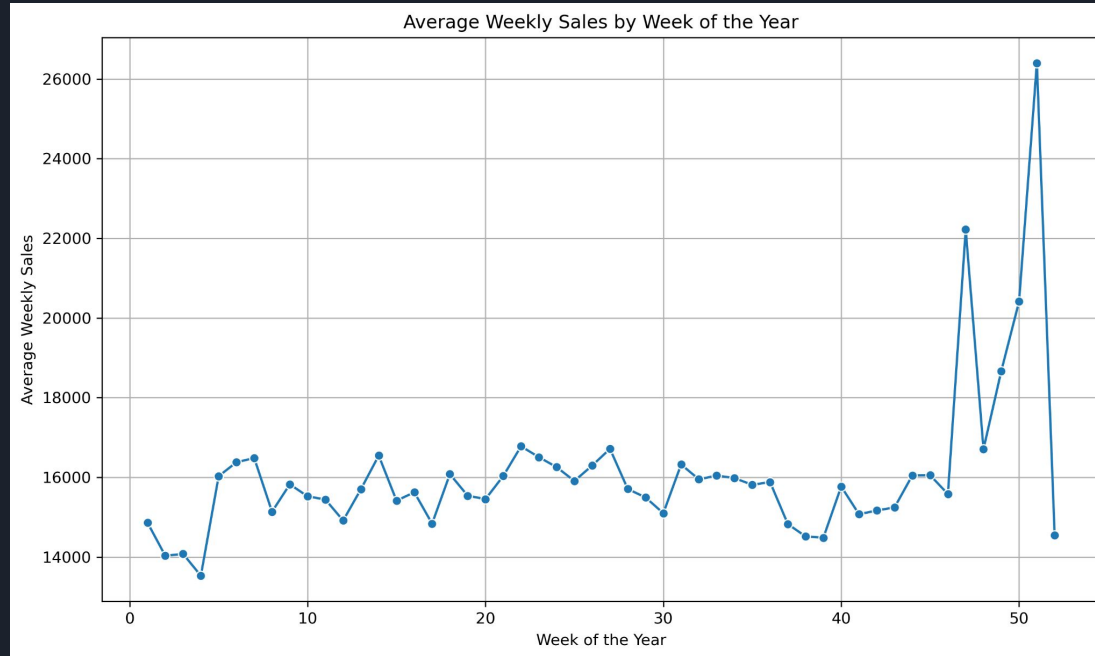
```
30 GROUP BY DATE_FORMAT(Date_Converted, '%m')
31 ORDER BY Month;
32
33 -- For Week Extraction
34 • SELECT DATE_FORMAT(Date_Converted, '%u') AS Week, AVG(Weekly_Sales) AS Average_Weekly_Sales
35 FROM `sales-data-set`
36 GROUP BY DATE_FORMAT(Date_Converted, '%u')
37 ORDER BY Average_Weekly_Sales DESC
38 LIMIT 5;
39
40
41
42
```

The result grid displays the following data:

Week	Average_Weekly_Sales
51	26396.399282968374
47	22220.944537674088
50	20413.010011721333
49	18668.667017455573
22	16779.736412535556

Seasonal Patterns:

Which weeks of the year have the highest average sales?



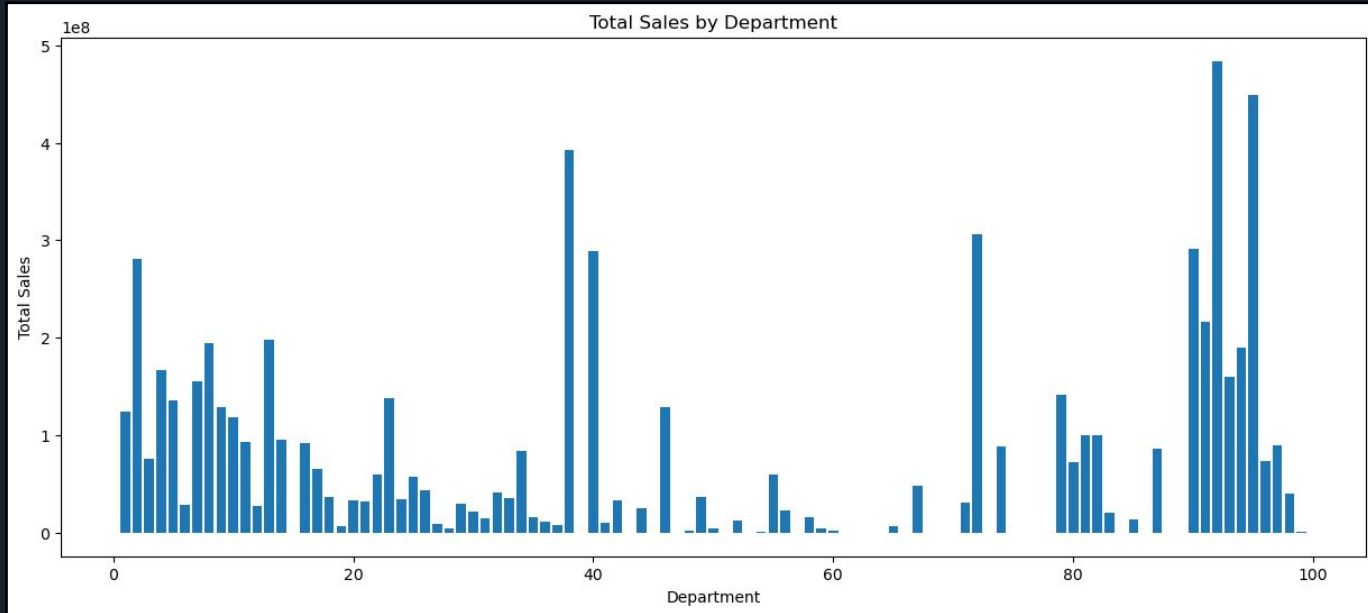


Which departments have the highest total sales across all stores?

```
SELECT
    dept AS Department,
    sum(Weekly_Sales) AS TotalSales
FROM
    stores s
    INNER JOIN sales ON s.Store =
sales.Store
GROUP BY dept
ORDER BY TotalSales desc;
```

Top 5 Depts	
Dept	Total Sales
92	\$ 483,943,341.73
95	\$ 449,320,162.45
38	\$ 393,118,136.80
72	\$ 305,725,152.19
90	\$ 291,068,463.56

Which departments have the highest total sales across all stores?



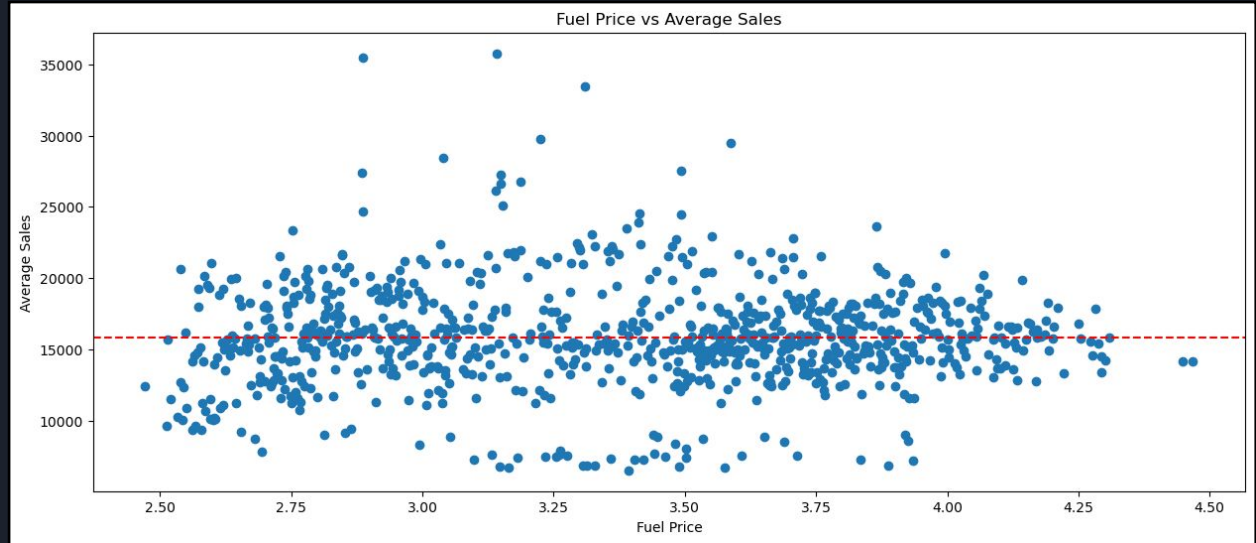
How does the price of fuel affect average sales per store?

```
SELECT fuel_price,  
       AVG(weekly_sales)  
FROM  
    features f  
    JOIN sales ON (sales.date,  
sales.store) = (f.date, f.store)  
GROUP BY fuel_price  
ORDER BY Fuel_Price;
```

Result Grid			Filter Rows:
	fuel_price	avg(weekly_sales)	
▶	2.472	12375.81670718444	
	2.513	9654.919520823161	
	2.514	15685.86072568796	
	2.52	11474.857366953142	
	2.533	10268.915508605185	
	2.539	12690.910837115469	
	2.54	20638.25391660418	
	2.542	10044.915615500344	
	2.545	12303.861575754065	
	2.548	16153.450585956858	
	2.55	10871.698950323389	
	2.561	14155.74561434549	
	2.562	9364.789343259552	
	2.565	14553.18865474011	

How does the price of fuel affect average sales per store?

	Fuel price	Weekly sales
Fuel price	1.00000	0.01581
Weekly Sales	0.01581	1.00000





Holiday Impact:

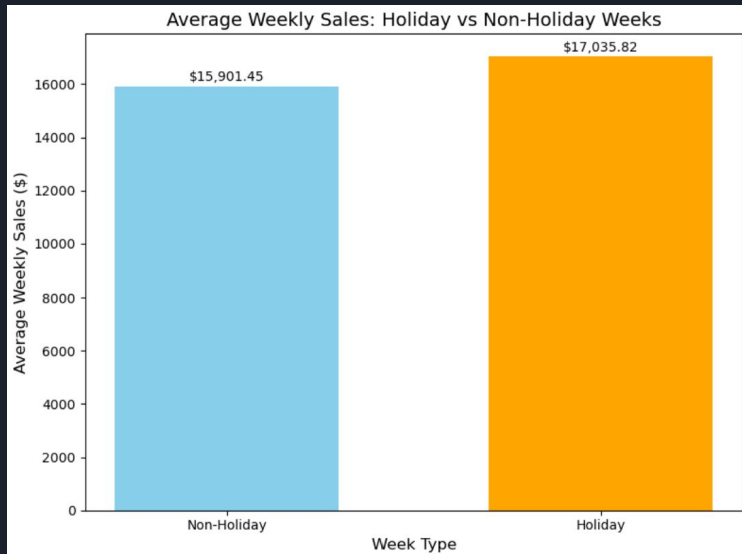
How do average weekly sales differ between holiday and non holidays weeks?

```
SELECT IsHoliday,  
       AVG(Weekly_Sales) AS  
       Avg_Weekly_Sales  
FROM Sales  
GROUP BY IsHoliday;
```

Is Holiday	Average Weekly Sales
False	\$15,901.45
True	\$17,035.82

Holiday Impact:

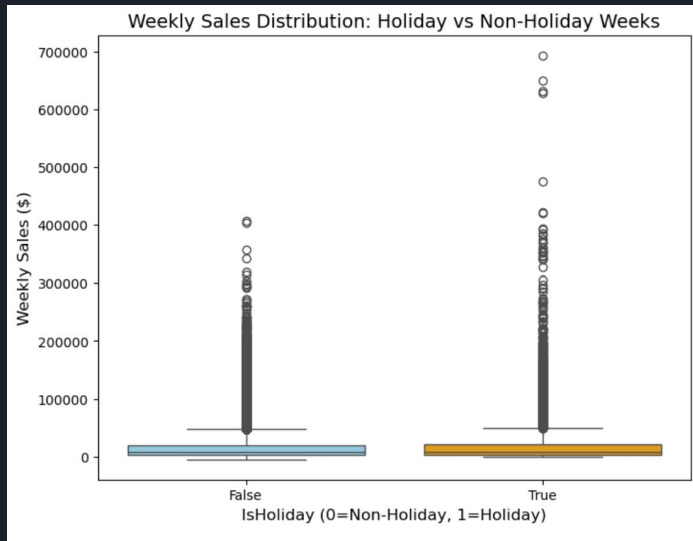
How do average weekly sales differ between holiday and non holidays weeks?



- Holiday weeks generate higher average weekly sales than non-holiday weeks, with sales being approximately 7% higher on average..
- Sales surges during holidays are likely driven by promotions, markdowns, and heightened consumer activity.

Holiday Impact:

How do average weekly sales differ between holiday and non holiday weeks?



- **Higher median:** Holiday weeks have slightly higher median weekly sales, supporting the findings from the bar chart.
- **Great Variability:** Sales during holidays exhibit more variability due to events like Super Bowl, Labor Day, Thanksgiving and Christmas.
- **Outliers:** Holidays weeks show fewer but larger outliers, indicating extreme sales spikes are likely driven by major promotions.



Holiday Impact:

How does markdown activity differ between holiday and non-holiday weeks?

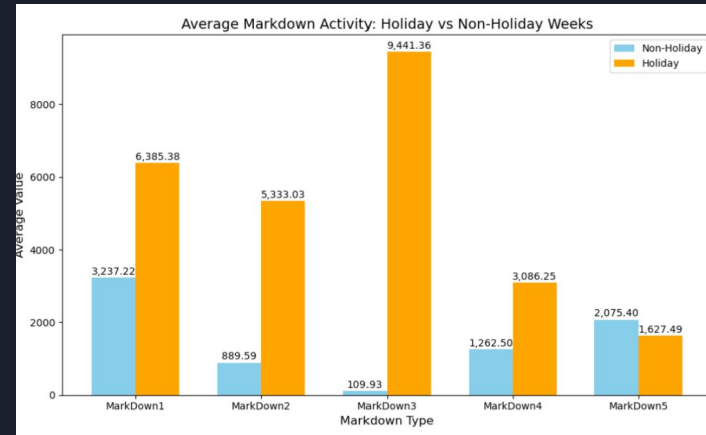
```
SELECT IsHoliday,  
       AVG(MarkDown1) AS avg_markdown1,  
       AVG(MarkDown2) AS avg_markdown2,  
       AVG(MarkDown3) AS avg_markdown3,  
       AVG(MarkDown4) AS avg_markdown4,  
       AVG(MarkDown5) AS avg_markdown5  
  
FROM Features  
  
GROUP BY IsHoliday;
```

Is Holiday	Average Markdown 1	Average Markdown 2	Average Markdown 3	Average Markdown 4	Average Markdown 5
False	\$3,237.22	\$889.59	\$109.93	\$1,262.50	\$2,075.40
True	\$6,385.38	\$5,333.03	\$9,441.36	\$3,086.25	\$1,627.49

Holiday Impact:

How does markdown activity differ between holiday and non-holiday weeks?

- **Higher Markdown Activity During Holidays:** Significant increases in categories like Markdown 1 and Markdown 3 during holiday weeks.
- **Exception in Markdown 5:** Slightly lower markdown values during holidays weeks compared to non-holiday weeks.
- **Business Implications:**
 - Prioritize markdown strategies in high-impact categories like Markdown 1 and Markdown 3 during holidays.
 - Refine Markdown 5 to optimize year-round sales.





Store Types:

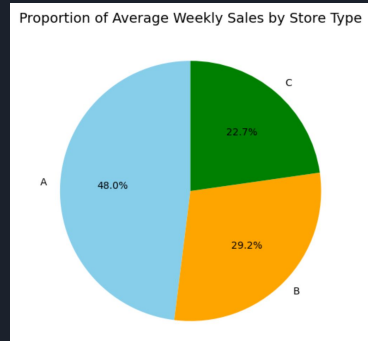
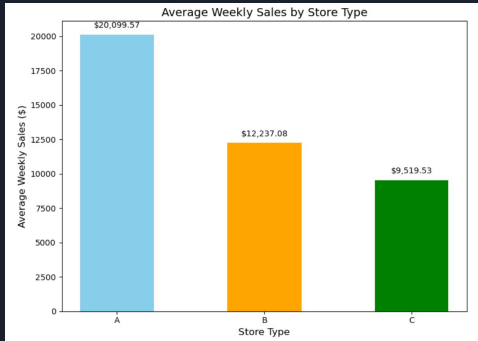
Which store type (A, B, C) generates the highest average sales?

```
SELECT st.Type AS Store_Type,  
       AVG(s.Weekly_Sales) AS  
avg_weekly_sales  
FROM Sales s  
JOIN Stores st ON s.Store =  
st.Store  
GROUP BY st.Type  
ORDER BY avg_weekly_sales DESC;
```

Store Type	Average Weekly Sales
A	\$20,099.57
B	\$12,237.08
C	\$9,519.53

Store Types:

Which store type (A, B, C) generates the highest average sales?



- **Store Type A dominates:** Accounts for 48% of total sales and generates significantly higher weekly sales compared to types B and C.
- **Performance Gap:** Store type A's average weekly sales are 65% higher than type B and over 110% higher than type C.
- **Stores Types B and C:** Type B contributes 29.2% of total sales, outperforming type C at 22.7%.
- **Business Implications:**
 - **Focus on Type A:** Prioritize investments in inventory, promotions, and customer experience for type A stores.
 - **Opportunities for Type B and C:** Improve performance with tailored product offering and targeted marketing.



Store Size and Sales:

Do larger stores have higher average sales?

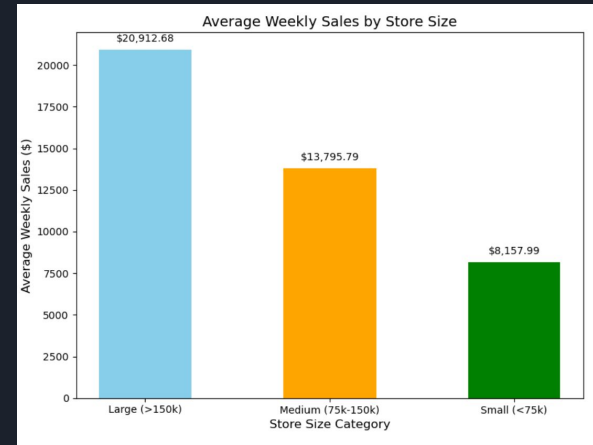
```
SELECT
  CASE
    WHEN st.Size < 75000 THEN 'Small
(<75k) '
    WHEN st.Size BETWEEN 75000 AND
150000 THEN 'Medium (75k-150k) '
    ELSE 'Large (>150k) '
  END AS Store_Size_Category,
  AVG(s.Weekly_Sales) AS
avg_weekly_sales
FROM Sales s
JOIN Stores st ON s.Store = st.Store
GROUP BY Store_Size_Category
ORDER BY avg_weekly_sales DESC;
```

Store Size Category	Average Weekly Sales
Large (greater than 150k)	\$20,912.68
Medium (between 75k and 150k)	\$13,795.79
Small (less than 75k)	\$8,157.99

Store Size and Sales:

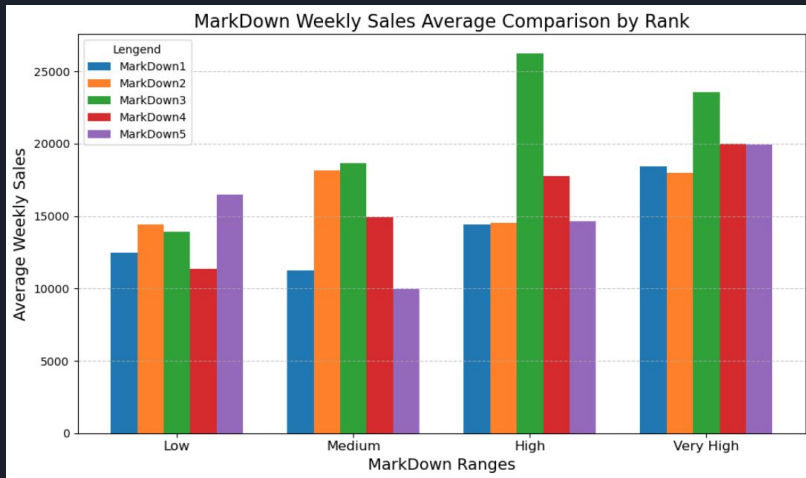
Do larger stores have higher average sales?

- **Larger Stores Dominate:** Larger stores generate the highest average weekly sales, about 51% more than medium stores and 156% more than small stores.
- **Clear Size-to-Sales Trend:** Averages sales consistently increase with store size, highlighting a strong relationship between store capacity and revenue potential.
- **Business Implications:**
 - Focus on optimizing inventory, promotions, and staffing in larger stores to maximize their revenue potential.
 - Develop tailored strategies for medium and small stores, such as niche product offering or localized marketing, to boost their performance.



Markdown Impact:

Do weeks with higher Markdown values have higher average sales?



Relationship between Markdown values and average sales:

- MarkDown1 and MarkDown5 have a strong positive correlation with higher average sales.
- MarkDown2, MarkDown3, and MarkDown4 show inconsistent patterns with no clear relationship to average sales.

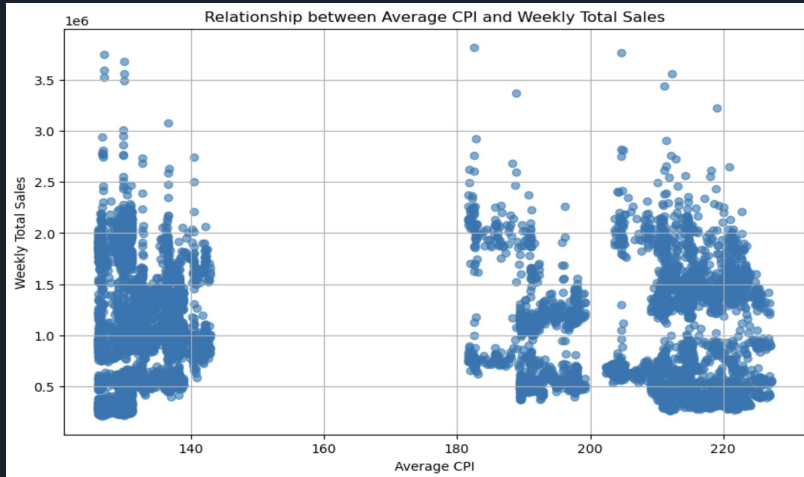
Business implications:

- Discounts associated with MarkDown1 and MarkDown5 should be prioritized as they effectively boost sales.
- Variables MarkDown2, MarkDown3, and MarkDown4 require re-evaluation to determine their potential impact.

```
SELECT f.MarkDown1,f.MarkDown2,f.MarkDown3,f.MarkDown4,f.MarkDown5,s.Weekly_Sales
FROM features f
JOIN sales s ON f.Store = s.Store AND f.Date = s.Date
WHERE f.Date >= '2011/11/01';
```

External Factors:

Is there a relationship between the Consumer Price Index (CPI) and weekly sales?



The correlation between Average CPI and Weekly Total Sales is: **-0.07**

Relationship between CPI and sales: The Consumer Price Index (CPI) has an insignificant impact on total weekly sales, as evidenced by a weak correlation coefficient of -0.07 and a dispersed distribution in the scatter plot.

Business implications:

- Changes in CPI do not significantly affect sales.
- Other external factors may have a greater influence on sales trends.

```
SELECT f.Store, f.Date, AVG(f.CPI) AS Avg_CPI, SUM(s.Weekly_Sales) AS Total_Weekly_Sales
FROM Features f JOIN Sales s ON f.Store = s.Store AND f.Date = s.Date
GROUP BY f.Store, f.Date
ORDER BY f.Date;
```



THANK YOU!

