

# Sales Performance Analysis

December 6, 2024

```
[9]: #Importing Libraries
import pandas as pd
import glob
import os
import matplotlib.pyplot as plt
```

```
[38]: #loading files from folder
folder_path='Sales Performance/Data'
csv_files= glob.glob(os.path.join(folder_path, '*.csv'))
dataframes={}
for file in csv_files:
    file_name=os.path.basename(file).replace('.csv','')
    dataframes=pd.read_csv(file)
```

```
[40]: #Checking whether files are loaded succesfully

folder_path = 'Sales Performance/Data'

csv_files = glob.glob(os.path.join(folder_path, "*.csv"))

dataframes = {}
for file in csv_files:
    file_name = os.path.basename(file).replace('.csv', '')
    try:
        dataframes[file_name] = pd.read_csv(file)
        print(f"Loaded {file_name} successfully.")
    except Exception as e:
        print(f"Error loading {file_name}: {e}")
```

Loaded stores successfully.  
Loaded Features successfully.  
Loaded sales successfully.

```
[42]: print("Top 5 rows of Features:")
print(dataframes['Features'].head())
print("\n")
```

Top 5 rows of Features:

	Store	Date	Temperature	Fuel_Price	MarkDown1	MarkDown2	\
0	1	05/02/2010	42.31	2.572	NaN	NaN	
1	1	12/02/2010	38.51	2.548	NaN	NaN	
2	1	19/02/2010	39.93	2.514	NaN	NaN	
3	1	26/02/2010	46.63	2.561	NaN	NaN	
4	1	05/03/2010	46.50	2.625	NaN	NaN	

	MarkDown3	MarkDown4	MarkDown5	CPI	Unemployment	IsHoliday
0	NaN	NaN	NaN	211.096358	8.106	False
1	NaN	NaN	NaN	211.242170	8.106	True
2	NaN	NaN	NaN	211.289143	8.106	False
3	NaN	NaN	NaN	211.319643	8.106	False
4	NaN	NaN	NaN	211.350143	8.106	False

```
[44]: #Printing top 5 rows
print("Top 5 rows of Features:")
print(dataframes['Features'].head())
print("\n")
print("Top 5 rows of Sales:")
print(dataframes['sales'].head())
print("\n")
print("Top 5 rows of stores")
print(dataframes["stores"].head())
print("\n")
```

Top 5 rows of Features:

	Store	Date	Temperature	Fuel_Price	MarkDown1	MarkDown2	\
0	1	05/02/2010	42.31	2.572	NaN	NaN	
1	1	12/02/2010	38.51	2.548	NaN	NaN	
2	1	19/02/2010	39.93	2.514	NaN	NaN	
3	1	26/02/2010	46.63	2.561	NaN	NaN	
4	1	05/03/2010	46.50	2.625	NaN	NaN	

	MarkDown3	MarkDown4	MarkDown5	CPI	Unemployment	IsHoliday
0	NaN	NaN	NaN	211.096358	8.106	False
1	NaN	NaN	NaN	211.242170	8.106	True
2	NaN	NaN	NaN	211.289143	8.106	False
3	NaN	NaN	NaN	211.319643	8.106	False
4	NaN	NaN	NaN	211.350143	8.106	False

Top 5 rows of Sales:

	Store	Dept	Date	Weekly_Sales	IsHoliday
0	1	1	05/02/2010	24924.50	False
1	1	1	12/02/2010	46039.49	True
2	1	1	19/02/2010	41595.55	False

3	1	1	26/02/2010	19403.54	False
4	1	1	05/03/2010	21827.90	False

Top 5 rows of stores

	Store	Type	Size
0	1	A	151315
1	2	A	202307
2	3	B	37392
3	4	A	205863
4	5	B	34875

```
[46]: # Merging sales and feature df
sales_features = pd.merge(dataframes['sales'], dataframes['Features'],
    on=['Store', 'Date'])
```

```
[48]: full_data= pd.merge(sales_features,dataframes['stores'],on='Store')
```

```
[50]: print(full_data.head())
```

	Store	Dept	Date	Weekly_Sales	IsHoliday_x	Temperature \
0	1	1	05/02/2010	24924.50	False	42.31
1	1	2	05/02/2010	50605.27	False	42.31
2	1	3	05/02/2010	13740.12	False	42.31
3	1	4	05/02/2010	39954.04	False	42.31
4	1	5	05/02/2010	32229.38	False	42.31

	Fuel_Price	Markdown1	Markdown2	Markdown3	Markdown4	Markdown5 \
0	2.572	NaN	NaN	NaN	NaN	NaN
1	2.572	NaN	NaN	NaN	NaN	NaN
2	2.572	NaN	NaN	NaN	NaN	NaN
3	2.572	NaN	NaN	NaN	NaN	NaN
4	2.572	NaN	NaN	NaN	NaN	NaN

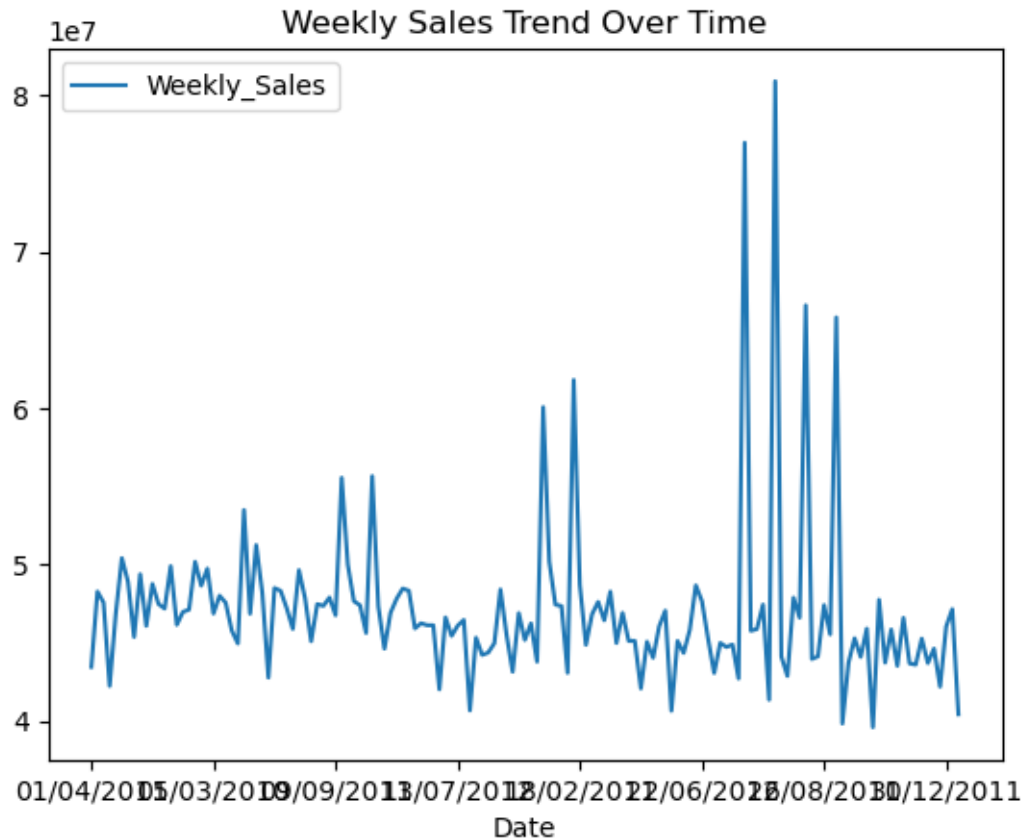
	CPI	Unemployment	IsHoliday_y	Type	Size
0	211.096358	8.106	False	A	151315
1	211.096358	8.106	False	A	151315
2	211.096358	8.106	False	A	151315
3	211.096358	8.106	False	A	151315
4	211.096358	8.106	False	A	151315

```
[52]: sales_trend = full_data.groupby('Date')['Weekly_Sales'].sum().reset_index()
```

```
[54]: # Plot sales trend over time (use matplotlib or seaborn for visualization)
```

```
sales_trend.plot(x='Date', y='Weekly_Sales', title="Weekly Sales Trend Over Time")
```

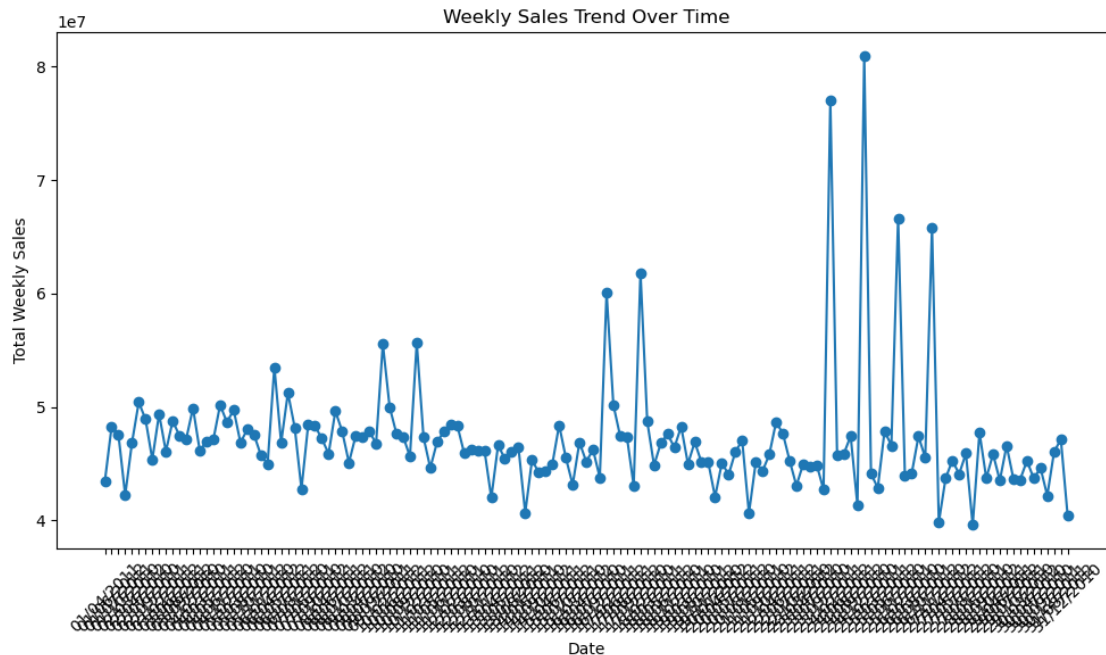
[54]: <Axes: title={'center': 'Weekly Sales Trend Over Time'}, xlabel='Date'>



```
[56]: # Plot the sales trend
plt.figure(figsize=(10, 6)) # Set figure size
plt.plot(sales_trend['Date'], sales_trend['Weekly_Sales'], marker='o') # Plot with markers for better visibility
plt.title("Weekly Sales Trend Over Time")
plt.xlabel("Date")
plt.ylabel("Total Weekly Sales")
plt.xticks(rotation=45)
plt.tight_layout() # Adjust layout for better spacing

# Save the plot as a PNG file
plt.savefig("weekly_sales_trend.png")

# Show the plot (optional)
plt.show()
```



```
[58]: print(full_data.head())
```

	Store	Dept	Date	Weekly_Sales	IsHoliday_x	Temperature	\
0	1	1	05/02/2010	24924.50	False	42.31	
1	1	2	05/02/2010	50605.27	False	42.31	
2	1	3	05/02/2010	13740.12	False	42.31	
3	1	4	05/02/2010	39954.04	False	42.31	
4	1	5	05/02/2010	32229.38	False	42.31	

	Fuel_Price	Markdown1	Markdown2	Markdown3	Markdown4	Markdown5	\
0	2.572	NaN	NaN	NaN	NaN	NaN	
1	2.572	NaN	NaN	NaN	NaN	NaN	
2	2.572	NaN	NaN	NaN	NaN	NaN	
3	2.572	NaN	NaN	NaN	NaN	NaN	
4	2.572	NaN	NaN	NaN	NaN	NaN	

	CPI	Unemployment	IsHoliday_y	Type	Size
0	211.096358	8.106	False	A	151315
1	211.096358	8.106	False	A	151315
2	211.096358	8.106	False	A	151315
3	211.096358	8.106	False	A	151315
4	211.096358	8.106	False	A	151315

```
[60]: # Group by holiday status and calculate average sales
holiday_sales = full_data.groupby('IsHoliday_x')['Weekly_Sales'].mean()
print("Average Sales - Holiday vs Non-Holiday Weeks:")
```

Average Sales - Holiday vs Non-Holiday Weeks:

```
[62]: print(holiday_sales)
```

```
IsHoliday_x
False      15901.445069
True       17035.823187
Name: Weekly_Sales, dtype: float64
```

```
[64]: # sales by store type
store_type_sales= full_data.groupby("Type")["Weekly_Sales"].mean()
print("Sales by store type")
```

Sales by store type

```
[66]: print(store_type_sales)
```

```
Type
A      20099.568043
B      12237.075977
C       9519.532538
Name: Weekly_Sales, dtype: float64
```

```
[68]: pd.options.display.float_format = '{:,.0f}'.format
```

```
[71]: # dept-wise Sales
dept_sales=full_data.groupby("Dept")["Weekly_Sales"].sum()
print("Sales by Dept")
```

Sales by Dept

```
[74]: print(dept_sales)
```

```
Dept
1      123,638,777
2      280,611,174
3       75,892,450
4      167,146,746
5      135,607,359
...
95     449,320,163
96      73,833,916
97      89,496,512
98      39,828,919
99       358,150
Name: Weekly_Sales, Length: 81, dtype: float64
```

```
[76]: top_depts = dept_sales.sort_values(ascending=False).head(10)
print("Top 10 Departments by Sales:")
```

Top 10 Departments by Sales:

```
[78]: print(top_depts)
```

```
Dept
92    483,943,342
95    449,320,163
38    393,118,137
72    305,725,152
90    291,068,464
40    288,936,022
2     280,611,174
91    216,781,706
13    197,321,570
8     194,280,781
Name: Weekly_Sales, dtype: float64
```

```
[80]: # Example: Using rolling average to forecast next week's sales (simple forecast)
full_data['Weekly_Sales_Rolling'] = full_data['Weekly_Sales'].rolling(window=4).
    .mean()
print(full_data[['Date', 'Weekly_Sales', 'Weekly_Sales_Rolling']].tail(10))
```

	Date	Weekly_Sales	Weekly_Sales_Rolling
421560	26/10/2012	1,689	8,615
421561	26/10/2012	8,188	6,989
421562	26/10/2012	25,352	8,987
421563	26/10/2012	16,331	12,890
421564	26/10/2012	54,609	26,120
421565	26/10/2012	2,488	24,695
421566	26/10/2012	5,203	19,658
421567	26/10/2012	56,017	29,579
421568	26/10/2012	6,817	17,632
421569	26/10/2012	1,077	17,279

```
[82]: # Export each DataFrame to a separate CSV file
sales_trend.to_csv('sales_trend.csv', index=True)
holiday_sales.to_csv('holiday_sales_impact.csv', index=True)
store_type_sales.to_csv('store_type_performance.csv', index=True)
dept_sales.to_csv('department_sales.csv', index=True)
```

```
[84]: # Step 2: Export to a single Excel file with multiple sheets
with pd.ExcelWriter('sales_performance_report.xlsx') as writer:
    sales_trend.to_excel(writer, sheet_name='Weekly Sales Trend', index=True)
    holiday_sales.to_excel(writer, sheet_name='Holiday Sales', index=True)
```

```
store_type_sales.to_excel(writer, sheet_name='Store Type Sales', index=True)
dept_sales.to_excel(writer, sheet_name='Top Departments', index=True)

print("Sales performance report saved as 'sales_performance_report.xlsx'.")
```

Sales performance report saved as 'sales\_performance\_report.xlsx'.

```
[86]: print(full_data.describe())
```

	Store	Dept	Weekly_Sales	Temperature	Fuel_Price	Markdown1	\
count	421,570	421,570	421,570	421,570	421,570	150,681	
mean	22	44	15,981	60	3	7,246	
std	13	30	22,711	18	0	8,291	
min	1	1	-4,989	-2	2	0	
25%	11	18	2,080	47	3	2,240	
50%	22	37	7,612	62	3	5,347	
75%	33	74	20,206	74	4	9,211	
max	45	99	693,099	100	4	88,647	

	Markdown2	Markdown3	Markdown4	Markdown5	CPI	Unemployment	\
count	111,248	137,091	134,967	151,432	421,570	421,570	
mean	3,335	1,439	3,383	4,629	171	8	
std	9,475	9,623	6,292	5,963	39	2	
min	-266	-29	0	135	126	4	
25%	42	5	504	1,878	132	7	
50%	192	25	1,481	3,359	182	8	
75%	1,927	104	3,595	5,564	212	9	
max	104,520	141,631	67,475	108,519	227	14	

	Size	Weekly_Sales_Rolling
count	421,570	421,567
mean	136,728	15,981
std	60,981	16,297
min	34,875	-45
25%	93,638	4,927
50%	140,167	11,122
75%	202,505	21,497
max	219,622	205,132

```
[ ]:
```