

## 1 : Import Sales Data and Basic Analysis

```
In [1]: # @title $Import \ \ Required \ \ Libraries$
import pandas as pd
import numpy as np
from IPython.display import display, Math

from matplotlib import pyplot as plt
import seaborn as sns
import plotly.express as px

import warnings
warnings.filterwarnings('ignore')

In [2]: # @title $Function \ \ for \ \ Initial \ \ Analysis$
def initial_analysis(df):
    print('first 5 records')
    display(df.head()); print('\n')

    print('last 5 records')
    display(df.tail()); print('\n')

    print('Shape of dataset')
    display(df.shape); print('\n')

    print('Data types')
    display(df.dtypes); print('\n')

    print('Dataset info')
    display(df.info()); print('\n')

    print('Dataset description')
    try:
        display(df.describe()); print()
    except:
        print('No numerical columns')
    try:
        display(df.describe(include='object')); print()
    except:
        print('No object columns')
    try:
        display(df.describe(include='boolean')); print()
    except:
        print('No boolean columns')

    print('\n')

    print('Duplicate Records')
    display(df.duplicated().sum()); print('\n')

    print('Missing Values')
    display(df.isnull().sum())

In [3]: # @title $Sales \ \ Data$
sales = pd.read_csv('sales_data-set.csv')
count=0
initial_analysis(sales)
```

first 5 records

Store	Dept	Date	Weekly_Sales	IsHoliday
0	1	1/10/2010	20094.19	False
1	1	2/4/2010	57258.43	False
2	1	2/7/2010	16333.14	False
3	1	3/9/2010	16241.78	False
4	1	3/12/2010	22517.56	False

last 5 records

Store	Dept	Date	Weekly_Sales	IsHoliday
430015	45	98 30/09/2011	727.05	False
430016	45	98 30/12/2011	553.21	True
430017	45	98 30/12/2011	553.21	True
430018	45	98 31/08/2012	346.04	False
430019	45	98 31/12/2010	74.55	True

Shape of dataset

(430020, 5)

## Data types

```
Store      int64
Dept      int64
Date      object
Weekly_Sales float64
IsHoliday  bool
dtype: object
```

## Dataset info

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 430020 entries, 0 to 430019
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype  
 ---  --          --          --      
 0   Store        430020 non-null   int64  
 1   Dept         430020 non-null   int64  
 2   Date         430020 non-null   object 
 3   Weekly_Sales 430020 non-null   float64 
 4   IsHoliday    430020 non-null   bool    
dtypes: bool(1), float64(1), int64(2), object(1)
memory usage: 13.5+ MB
None
```

## Dataset description

	Store	Dept	Weekly_Sales
<b>count</b>	430020.000000	430020.000000	430020.000000
<b>mean</b>	22.175103	44.122734	15855.891497
<b>std</b>	12.779362	30.326822	22580.083242
<b>min</b>	1.000000	1.000000	-4988.940000
<b>25%</b>	11.000000	18.000000	2111.652500
<b>50%</b>	22.000000	36.000000	7546.000000
<b>75%</b>	33.000000	72.000000	19965.532500
<b>max</b>	45.000000	99.000000	693099.360000

## Date

<b>count</b>	430020
<b>unique</b>	143
<b>top</b>	30/12/2011
<b>freq</b>	6080

## IsHoliday

<b>count</b>	430020
<b>unique</b>	2
<b>top</b>	False
<b>freq</b>	396942

Duplicate Records  
8430

## Missing Values

```
Store      0
Dept      0
Date      0
Weekly_Sales 0
IsHoliday  0
dtype: int64
```

- There are some negative entries in Weekly\_Sales column, so lets analyse them

```
In [4]: # @title $Store \ \ Wise \ \ Positive \ \ Negative \ \ Sales$
# Store wise positive sales
store_wise_positive_sales = sales[sales['Weekly_Sales'] >= 0].groupby(['Store'])['Weekly_Sales'].sum().sort_values(ascending=True)
store_wise_positive_sales = store_wise_positive_sales.reset_index()
store_wise_positive_sales.rename(columns={'Weekly_Sales':'Total_Positive_Sales'}, inplace=True)
#store_wise_positive_sales.head()

# Store wise negative sales
store_wise_negative_sales = sales[sales['Weekly_Sales'] < 0].groupby(['Store'])['Weekly_Sales'].sum().sort_values(ascending=True)
store_wise_negative_sales = store_wise_negative_sales.reset_index()
store_wise_negative_sales.rename(columns={'Weekly_Sales':'Total_Negative_Sales'}, inplace=True)
#store_wise_negative_sales.head()
```

```
# Store wise positive and negative sales
store_wise_sales = store_wise_positive_sales.merge(store_wise_negative_sales, on='Store', how='left')
store_wise_sales['neg_to_pos_%'] = store_wise_sales['Total_Negative_Sales']/store_wise_sales['Total_Positive_Sales']*100
display(store_wise_sales.sort_values(by='neg_to_pos_%', ascending=True).head())

print('\n')

neg_to_pos_perc = -1*store_wise_sales['Total_Negative_Sales'].sum()*100/store_wise_sales['Total_Positive_Sales'].sum()
print(f'Percentage Negative Sales : {neg_to_pos_perc}')
```

Store	Total_Positive_Sales	Total_Negative_Sales	neg_to_pos_%
20	35	1.339820e+08	-9349.73
8	16	7.523807e+07	-4309.70
29	28	1.917481e+08	-7827.45
11	42	8.002251e+07	-2311.24
24	12	1.469707e+08	-4004.41

Percentage Negative Sales : 0.001320548444576415

- Percentage of Negative Sales is extremely low (~ 0.001%), so we will replace negative sales values by zero

```
# @title $Remove \ \ Negative \ \ entries \ \ from \ \ Weekly\Sales \ \ Column$ {run:"auto", "form-width": "20%"}
delete_negative_records = True # @param {type: "boolean"}

if(count==0):
    original_sales_data = sales.copy()
    count+=1

if(delete_negative_records):
    sales['Weekly_Sales_New'] = sales['Weekly_Sales'].apply(lambda x: 0 if x < 0 else x)
    display(sales.describe())

print('\n')

print('Update Weekly_Sales Column')
sales.drop(columns=['Weekly_Sales'], inplace=True)
sales.rename(columns={'Weekly_Sales_New': 'Weekly_Sales'}, inplace=True)
else:
    sales = original_sales_data.copy()
    print('Original Dataset')
display(sales.describe())
```

	Store	Dept	Weekly_Sales	Weekly_Sales_New
count	430020.000000	430020.000000	430020.000000	430020.000000
mean	22.175103	44.122734	15855.891497	15856.100885
std	12.779362	30.326822	22580.083242	22579.932300
min	1.000000	1.000000	-4988.940000	0.000000
25%	11.000000	18.000000	2111.652500	2111.652500
50%	22.000000	36.000000	7546.000000	7546.000000
75%	33.000000	72.000000	19965.532500	19965.532500
max	45.000000	99.000000	693099.360000	693099.360000

Update Weekly\_Sales Column

	Store	Dept	Weekly_Sales
count	430020.000000	430020.000000	430020.000000
mean	22.175103	44.122734	15856.100885
std	12.779362	30.326822	22579.932300
min	1.000000	1.000000	0.000000
25%	11.000000	18.000000	2111.652500
50%	22.000000	36.000000	7546.000000
75%	33.000000	72.000000	19965.532500
max	45.000000	99.000000	693099.360000

```
# @title $Remove \ \ Duplicate \ \ Records$
print(f'{sales.duplicated().sum()} duplicate records are removed')
sales.drop_duplicates(inplace=True)
```

8430 duplicate records are removed

```
In [7]: # @title $Change \ \ type \ \ of \ \ Date \ \ column$  
sales['Date'] = pd.to_datetime(sales['Date'], format='%d/%m/%Y', errors='coerce')  
display(sales.dtypes)
```

Store	int64
Dept	int64
Date	datetime64[ns]
IsHoliday	bool
Weekly_Sales	float64
dtype:	object

```
In [8]: # @title $Total \ \ Weeks \ \ Data \ \ Expected$  
from datetime import date  
start_date = date(2010, 2, 5) # Friday  
end_date = date(2012, 11, 1) # Thursday  
  
total_days = (end_date - start_date).days + 1  
total_days/7
```

Out[8]: 143.0

```
In [9]: # @title $Available \ \ Data$  
print('Total Stores:', sales['Store'].unique().size)  
print('Stores:', sales['Store'].unique())  
print('\n')  
  
print('Total Departments:', sales['Dept'].unique().size)  
print('Departments:', sales['Dept'].unique())  
print('\n')  
  
print('No. of Weeks:', sales['Date'].unique().size)  
print('Weeks:', sales['Date'].unique())
```

Total Stores: 45  
Stores: [ 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24  
25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45]

Total Departments: 81  
Departments: [ 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25  
26 27 28 29 30 31 32 33 34 35 36 37 38 40 41 42 44 45 46 47 48 49 51 52  
54 55 56 58 59 60 67 71 72 74 77 78 79 80 81 82 83 85 87 90 91 92 93 94  
95 96 97 98 99 39 50 43 65]

No. of Weeks: 143  
Weeks: <DatetimeArray>  
['2010-10-01 00:00:00', '2010-04-02 00:00:00', '2010-07-02 00:00:00',  
'2010-09-03 00:00:00', '2010-12-03 00:00:00', '2010-06-04 00:00:00',  
'2010-02-05 00:00:00', '2010-03-05 00:00:00', '2010-11-05 00:00:00',  
'2010-08-06 00:00:00',  
...  
'2012-06-29 00:00:00', '2011-07-29 00:00:00', '2010-10-29 00:00:00',  
'2012-03-30 00:00:00', '2010-04-30 00:00:00', '2010-07-30 00:00:00',  
'2011-09-30 00:00:00', '2011-12-30 00:00:00', '2012-08-31 00:00:00',  
'2010-12-31 00:00:00']  
Length: 143, dtype: datetime64[ns]

- There are no missing weeks in the sales dataset
- Sales dataset contains weekly sales data for all 45 stores for all 143 Weeks from 2010-02-05 to 2012-11-01
- Also, sales data exist fro 81 departments

## 2 : Merge Stores and Sales Datasets

```
In [10]: # @title $Import \ \ Stores \ \ Data$  
stores = pd.read_csv('stores data-set.csv')
```

```
In [11]: # @title $Merge \ \ Stores \ \ & \ \ Sales \ \ on \ \ Store \ \ Column$  
sales = sales.merge(stores, on=['Store'], how='outer')  
print('Merged Dataset : sales')  
display(sales); print('\n')  
  
print('Null Entries')  
display(sales.isna().sum())
```

Merged Dataset : sales

	Store	Dept	Date	IsHoliday	Weekly_Sales	Type	Area
0	1	1	2010-10-01	False	20094.19	A	151315
1	1	1	2010-04-02	False	57258.43	A	151315
2	1	1	2010-07-02	False	16333.14	A	151315
3	1	1	2010-09-03	False	16241.78	A	151315
4	1	1	2010-12-03	False	22517.56	A	151315
...	...	...	...	...	...	...	...
421585	45	98	2010-07-30	False	5.50	B	118221
421586	45	98	2011-09-30	False	727.05	B	118221
421587	45	98	2011-12-30	True	553.21	B	118221
421588	45	98	2012-08-31	False	346.04	B	118221
421589	45	98	2010-12-31	True	74.55	B	118221

421590 rows × 7 columns

Null Entries

```
Store      0
Dept      0
Date      0
IsHoliday 0
Weekly_Sales 0
Type      0
Area      0
dtype: int64
```

### 3 : Analysis of Merged Dataset

```
# @title $Store \ \ wise \ \ unavailable \ \ Departments$ 
def not_in_depts(x, list_departments):
    not_in_list = []
    for i in list_departments:
        if i not in x:
            not_in_list.append(i)
    return not_in_list

def not_in_stores(x):
    not_in_list = []
    for i in range(1,46):
        if i not in x:
            not_in_list.append(i)
    return not_in_list

list_departments = sales['Dept'].unique()
list_stores = sales['Store'].unique()

# Store with the departments it does not have
store_not_in_depts = sales.groupby('Store')['Dept'].unique().apply(lambda x: not_in_depts(x, list_departments))
store_not_in_depts = store_not_in_depts.reset_index()
store_not_in_depts.rename(columns={'Dept': 'Not in following Depts'}, inplace=True)
store_not_in_depts['Total'] = store_not_in_depts['Not in following Depts'].apply(lambda x: len(x))
display(store_not_in_depts.head()); print('\n')

# Dept not in stores
dept_not_in_stores = sales.groupby('Dept')['Store'].unique().apply(lambda x: not_in_stores(x))
dept_not_in_stores = dept_not_in_stores.reset_index()
dept_not_in_stores.rename(columns={'Store': 'Not in following Stores'}, inplace=True)
dept_not_in_stores['Total'] = dept_not_in_stores['Not in following Stores'].apply(lambda x: len(x))
display(dept_not_in_stores.head())
```

	Store	Not in following Depts	Total
0	1	[39, 50, 43, 65]	4
1	2	[50, 43, 65]	3
2	3	[37, 48, 58, 93, 99, 39, 50, 43, 65]	9
3	4	[50, 43, 65]	3
4	5	[37, 48, 83, 93, 99, 39, 50, 43, 65]	9

Dept	Not in following Stores	Total
0	1	0
1	2	0
2	3	0
3	4	0
4	5	0

```
In [13]: # @title $Store \ \ n \ unavailable \ \ Departments$ {run: "auto", "form-width": "20%"}

store_no = 10 # @param {"type": "slider", "min": 1, "max": 45, "step": 1}
temp_list = list(store_not_in_depts[store_not_in_depts['Store'] == store_no]['Not in following Depts'])[0]

if(len(temp_list)!=0):
    print(f'Store {store_no} is not having following {len(temp_list)} Departments:')
    print(temp_list)
else:
    print(f'Store {store_no} is having all the Departments')
```

Store 10 is not having following 4 Departments:  
[99, 39, 43, 65]

```
In [14]: # @title $Stores \ \ not \ \ having \ \ Department \ \ n$ {run: "auto", "form-width": "20%"}

dept_no = 51 # @param {"type": "slider", "min": 1, "max": 99, "step": 1}

if(dept_no in list_departments):
    temp_list = list(dept_not_in_stores[dept_not_in_stores['Dept'] == dept_no]['Not in following Stores'])[0]

    if(len(temp_list)!=0):
        print(f'Department {dept_no} is not in following {len(temp_list)} Stores:')
        print(temp_list)
    else:
        print(f'Department {dept_no} is available in all the Stores')
else:
    print(f'Department {dept_no} is not available in the dataset')
```

Department 51 is not in following 8 Stores:  
[30, 33, 36, 37, 38, 42, 43, 44]

```
In [15]: # @title $Function \ \ to \ \ get \ \ Sales \ \ Plots$

def get_sales_plot(df, col):
    df = df.groupby(['Type', col])['Weekly_Sales'].sum().sort_values(ascending=False)
    df = df.reset_index()
    df = df.rename(columns={'Weekly_Sales': 'Total_Sales'})

    fig = px.bar(df, x=col, y='Total_Sales', color='Type', title='Total Sales by ' + col)
    fig.show()

    fig = px.scatter(df, x=col, y='Total_Sales', size='Total_Sales', color='Type')
    fig.show()

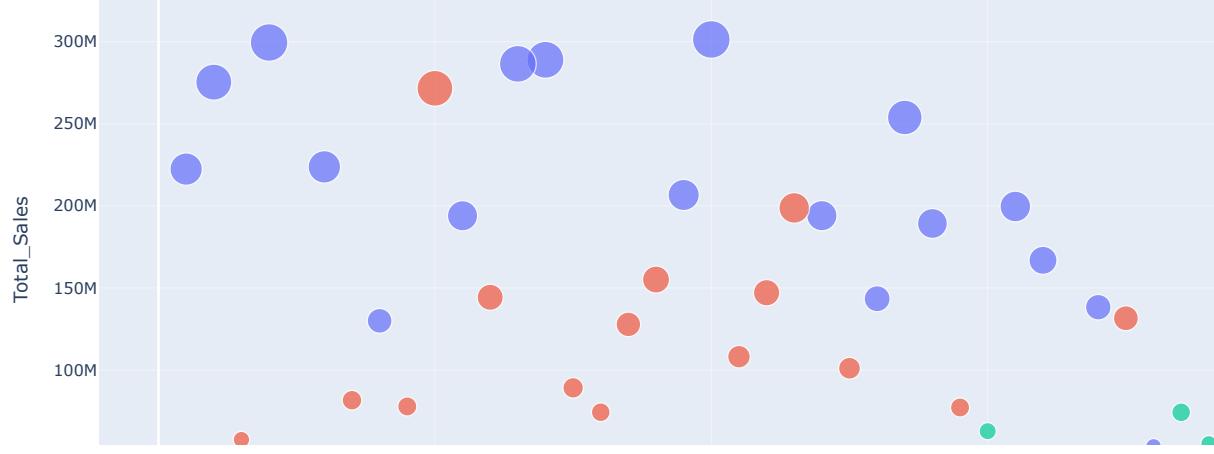
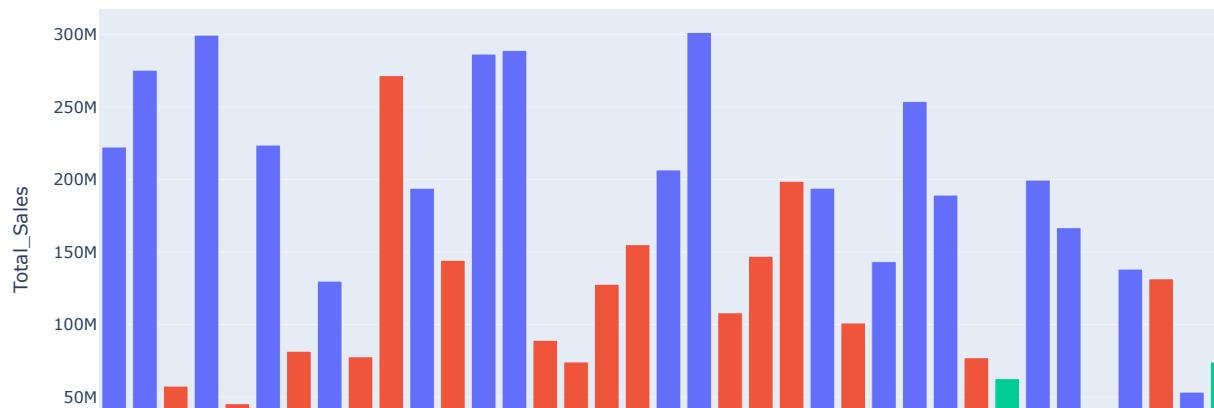
def get_pie_plot(df, col):
    df = df.groupby([col])['Weekly_Sales'].sum().sort_values(ascending=False)
    df = df.reset_index()
    df = df.rename(columns={'Weekly_Sales': 'Total_Sales'})

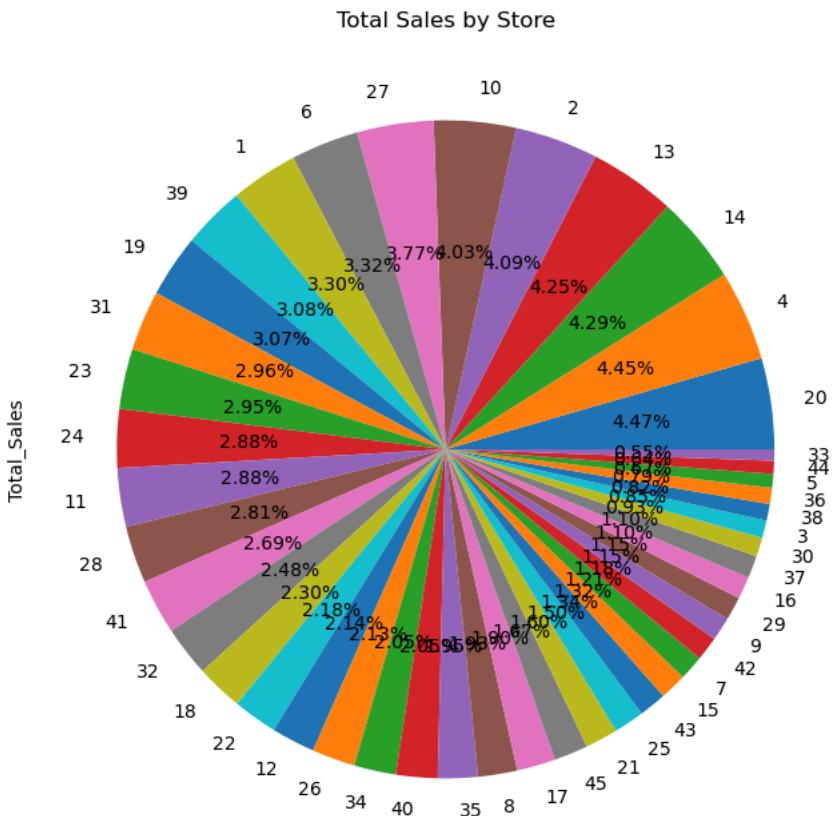
    plt.figure(figsize=(8, 8))
    df['Total_Sales'].plot.pie(labels=df[col], title='Total Sales by ' + col, autopct='%1.2f%')
    plt.show()
```

```
In [16]: # @title $Stores \ \ wise \ \ Total \ \ Sales$

get_sales_plot(sales, 'Store')
get_pie_plot(sales, 'Store')
```

## Total Sales by Store





## Observations:

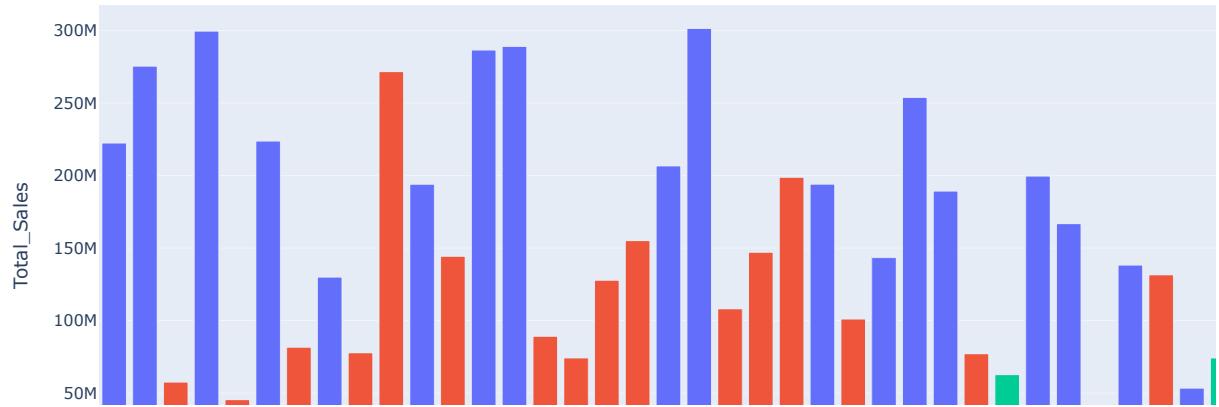
- Store 20 has generated most sales closely followed by store 4. Which is somewhat expected, as these two are of Type A and also some of largest stores.
  - Surprisingly, store 10 (Type B) has also generated large number of sales, even though it is a comparatively average size store.

```
In [17]: # @title $Store \ \ wise \ \ Analysis \ \ of \ \ Area \ \ and \ \ Sales$  
store_wise_total_sales = sales.groupby(['Store','Type','Area'])['Weekly_Sales'].sum().sort_values(ascending=False)  
store_wise_total_sales = store_wise_total_sales.reset_index()  
store_wise_total_sales = store_wise_total_sales.rename(columns={'Weekly_Sales':'Total_Sales'})  
store_wise_total_sales['Sales/Area'] = store_wise_total_sales['Total_Sales']/store_wise_total_sales['Area']  
store_wise_total_sales
```

Out[17]:	Store	Type	Area	Total_Sales	Sales/Area
0	20	A	203742	3.014014e+08	1479.328668
1	4	A	205863	2.995453e+08	1455.070942
2	14	A	200898	2.890019e+08	1438.550232
3	13	A	219622	2.865180e+08	1304.595853
4	2	A	202307	2.753872e+08	1361.233944
5	10	B	126512	2.716241e+08	2147.022491
6	27	A	204184	2.538572e+08	1243.276603
7	6	A	202505	2.237586e+08	1104.953400
8	1	A	151315	2.224068e+08	1469.826301
9	39	A	184109	2.074465e+08	1126.759117
10	19	A	203819	2.066377e+08	1013.829367
11	31	A	203750	1.996149e+08	979.705180
12	23	B	114533	1.987527e+08	1735.331698
13	24	A	203819	1.940166e+08	951.906332
14	11	A	207499	1.939644e+08	934.772659
15	28	A	206302	1.892715e+08	917.448731
16	41	A	196321	1.813429e+08	923.705848
17	32	A	203007	1.668208e+08	821.748886
18	18	B	120653	1.551158e+08	1285.635977
19	22	B	119557	1.470768e+08	1230.181177
20	12	B	112238	1.442911e+08	1285.581947
21	26	A	152513	1.434166e+08	940.356690
22	34	A	158114	1.382526e+08	874.385742
23	40	A	155083	1.378716e+08	889.018372
24	35	B	103681	1.315300e+08	1268.602628
25	8	A	155078	1.299515e+08	837.975146
26	17	B	93188	1.277832e+08	1371.240820
27	45	B	118221	1.123973e+08	950.739042
28	21	B	140167	1.081196e+08	771.362443
29	25	B	128107	1.010618e+08	788.886192
30	43	C	41062	9.056653e+07	2205.604525
31	15	B	123737	8.913487e+07	720.357438
32	7	B	70713	8.159950e+07	1153.953245
33	42	C	39690	7.956804e+07	2004.737810
34	9	B	125833	7.779117e+07	618.209574
35	29	B	93638	7.714283e+07	823.841032
36	16	B	57197	7.425674e+07	1298.262760
37	37	C	39910	7.420286e+07	1859.254837
38	30	C	42988	6.271701e+07	1458.942317
39	3	B	37392	5.758793e+07	1540.113550
40	38	C	39690	5.515990e+07	1389.768221
41	36	A	39910	5.341335e+07	1338.345104
42	5	B	34875	4.547612e+07	1303.974759
43	44	C	39910	4.329367e+07	1084.782549
44	33	A	39690	3.716033e+07	936.264182

```
In [18]: # @title $Stores \ \ and \ \ their \ \ Sales$  
fig = px.bar(store_wise_total_sales, x='Store', y='Total_Sales', color='Type', title='Sales by Store')  
fig.show()
```

### Sales by Store



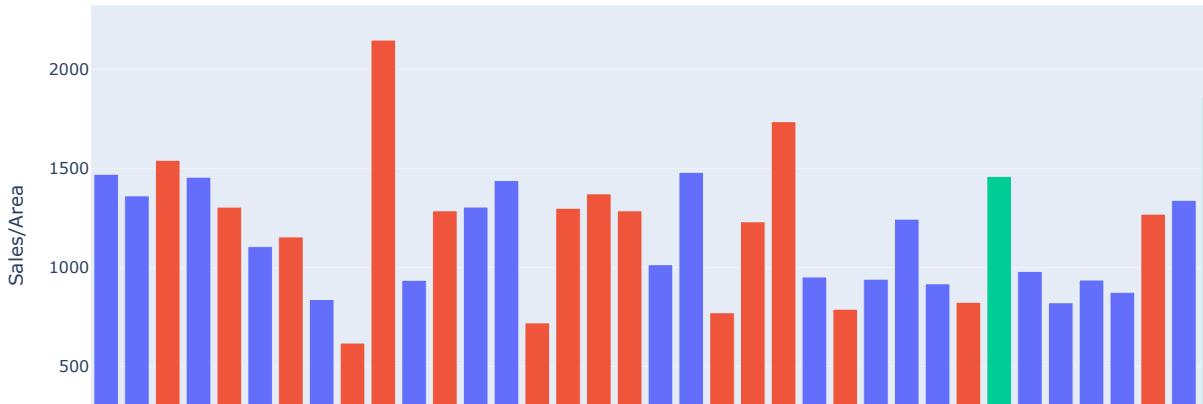
```
In [19]: # @title $Stores \ \ and \ \ their \ \ Areas$  
fig = px.bar(store_wise_total_sales, x='Store', y='Area', color='Type', title='Area by Store')  
fig.show()
```

### Area by Store



```
In [20]: # @title $Sales \ \ to \ \ Area \ \ Ratio$  
fig = px.bar(store_wise_total_sales, x='Store', y='Sales/Area', color='Type', title='Sales to Area Ratio')  
fig.show()
```

### Sales to Area Ratio

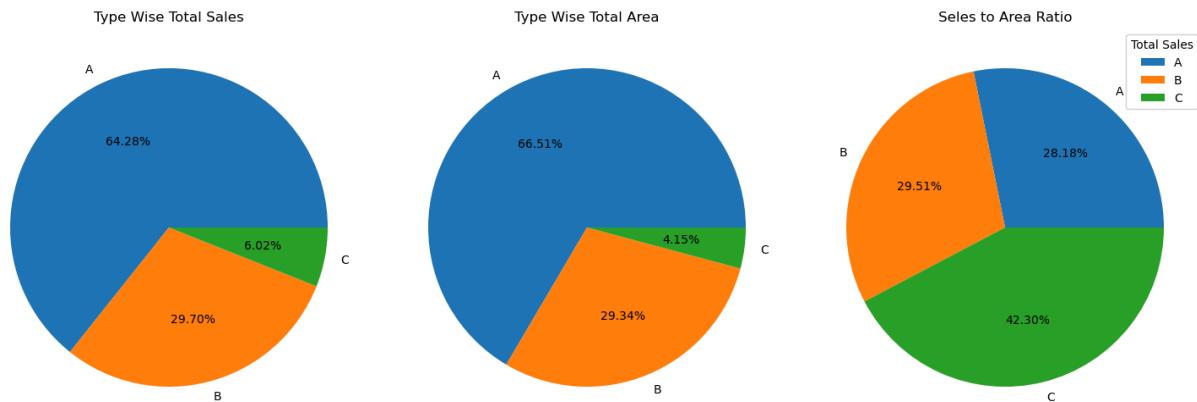


Observations:

- Store 10 (Type B) has highest Sales to Area ratio

```
# @title $Type \ \ wise \ \ total \ \ Sales \ \ and \ \ Area$  
type_wise_total_sales_area = store_wise_total_sales.groupby(['Type'])[['Total_Sales','Area']].sum()  
type_wise_total_sales_area.reset_index(inplace=True)  
type_wise_total_sales_area['Ratio']=type_wise_total_sales_area['Total_Sales']/type_wise_total_sales_area['Area']  
display(type_wise_total_sales_area); print('\n')  
  
fig, axes = plt.subplots(1, 3, figsize=(15, 5))  
axes[0].pie(type_wise_total_sales_area['Total_Sales'], labels=type_wise_total_sales_area['Type'], autopct='%.1f%%')  
axes[0].set_title("Type Wise Total Sales")  
  
axes[1].pie(type_wise_total_sales_area['Area'], labels=type_wise_total_sales_area['Type'], autopct='%.1f%%')  
axes[1].set_title("Type Wise Total Area")  
  
axes[2].pie(type_wise_total_sales_area['Ratio'], labels=type_wise_total_sales_area['Type'], autopct='%.1f%%')  
axes[2].set_title("Sales to Area Ratio")  
  
plt.tight_layout()  
plt.legend(title="Total Sales", loc="best")  
plt.show()
```

	Type	Total_Sales	Area	Ratio
0	A	4.331057e+09	3899450	1110.684194
1	B	2.000742e+09	1720242	1163.058249
2	C	4.055080e+08	243250	1667.042227

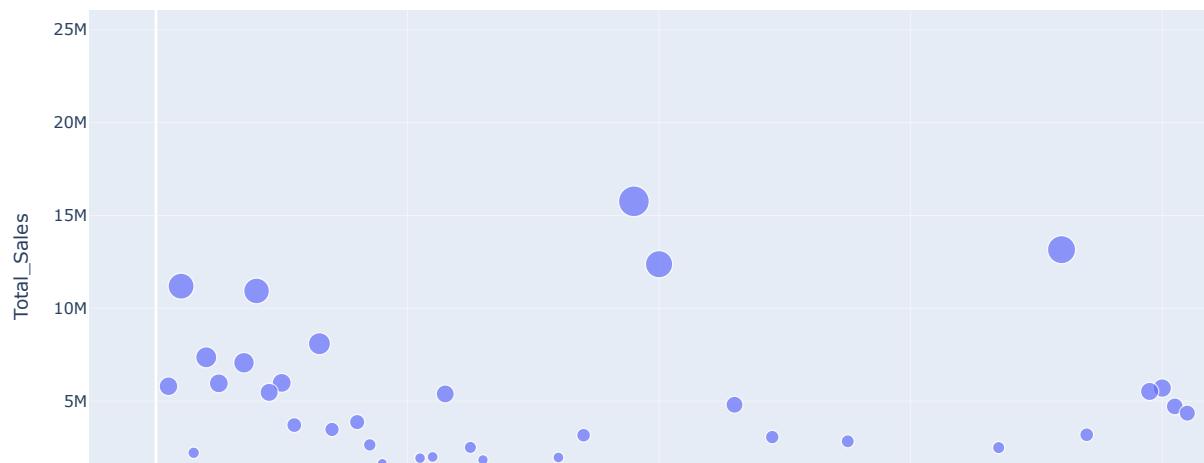


- This Pie Chart indicates that smaller stores have higher Sales to Area ratio

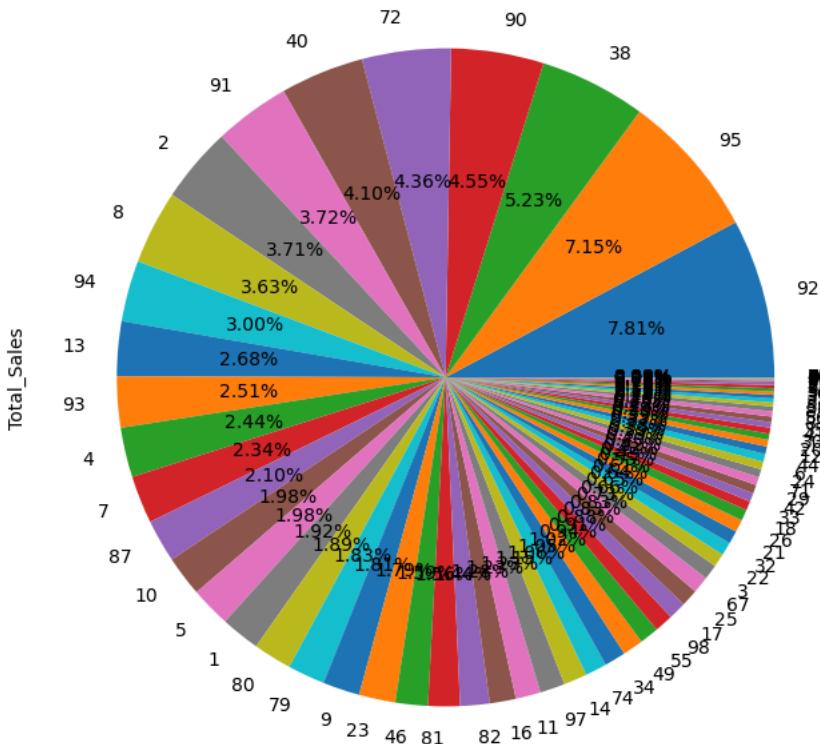
```
In [22]: # @title $Department \ \ wise \ \ Sales \ \ of \ \ Any \ \ Store$ {run: "auto", "form-width": "20%"}  
store_no = 20 # @param {"type": "slider", "min": 1, "max": 45, "step": 1}
```

```
df = sales[sales['Store']==store_no]
get_sales_plot(df,'Dept')
get_pie_plot(df,'Dept')
```

Total Sales by Dept



Total Sales by Dept

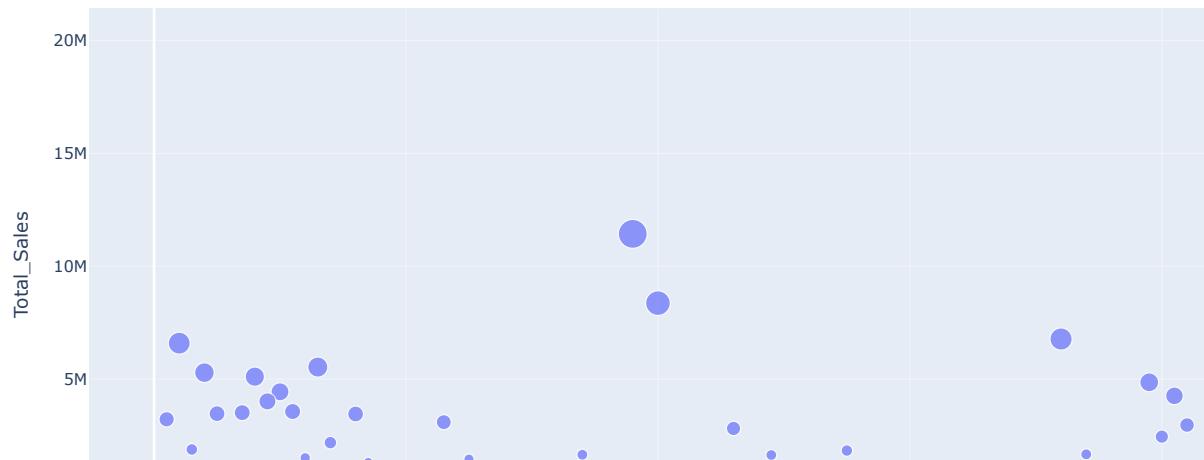


```
In [23]: # @title $Department wise Sales of ALL Stores
for store_no in range(1,46):
    print('Store:',store_no)
    df = sales[sales['Store']==store_no]
    get_sales_plot(df,'Dept')
    get_pie_plot(df,'Dept')
    print()
    print('*'*150)
    print()
```

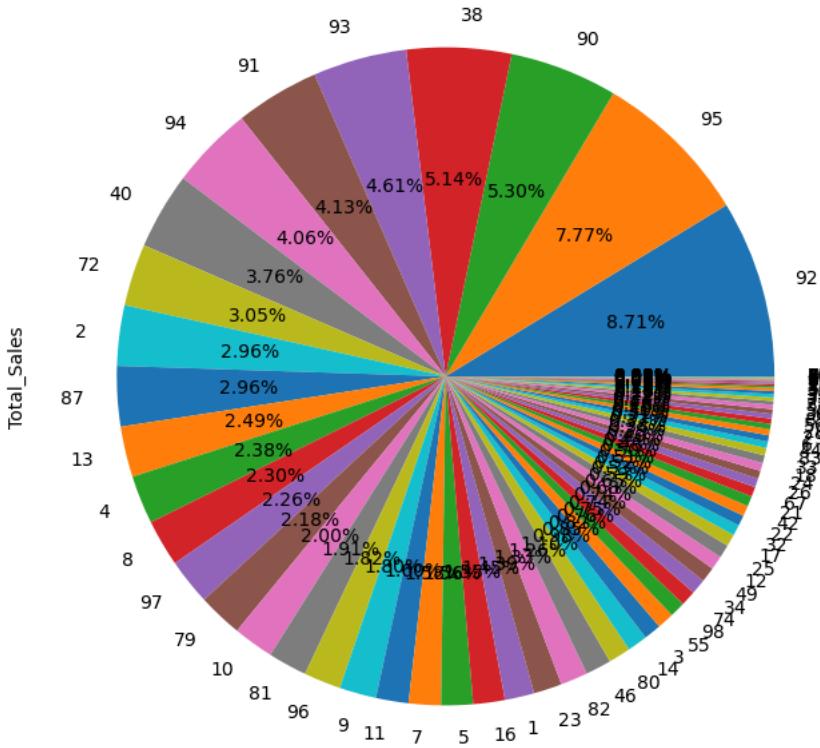
Store: 1

Total Sales by Dept





## Total Sales by Dept



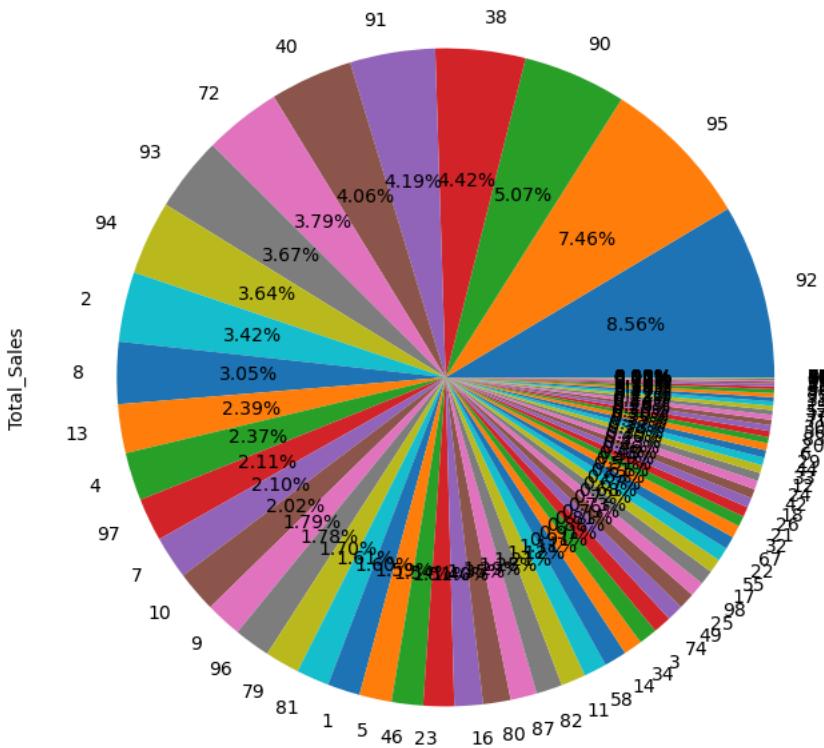
\*\*\*\*\*  
\*\*\*\*\*

Store: 2

## Total Sales by Dept

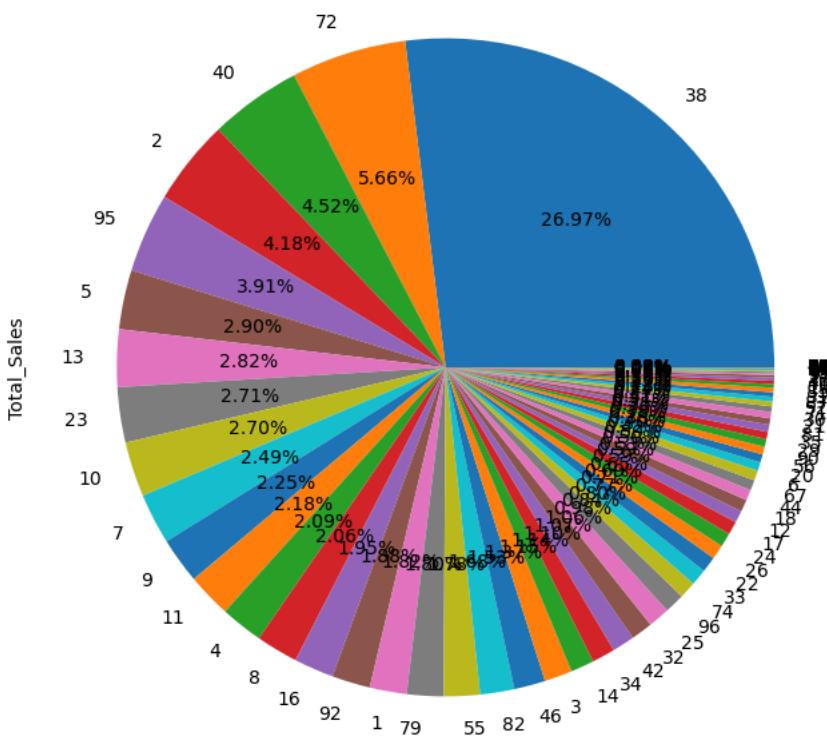


## Total Sales by Dept





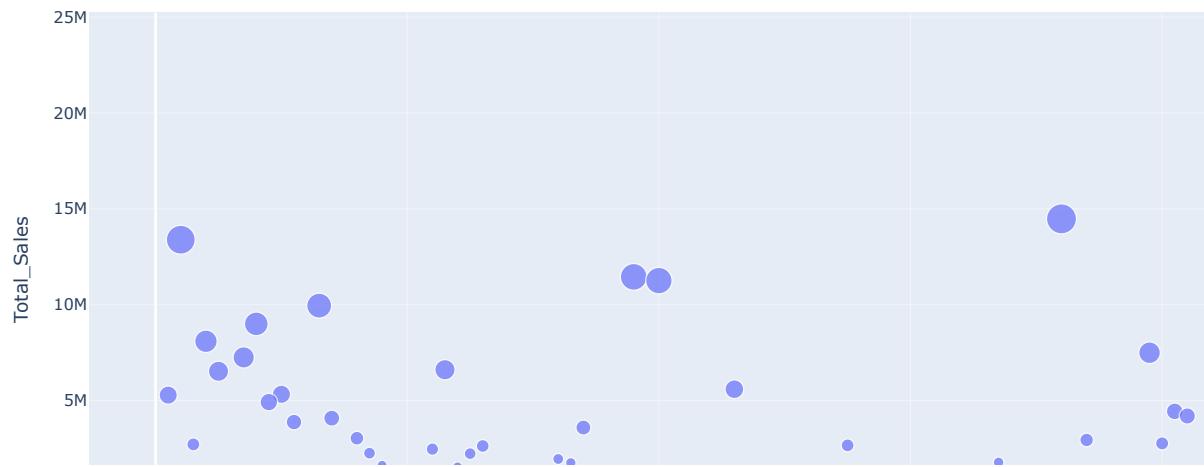
Total Sales by Dept



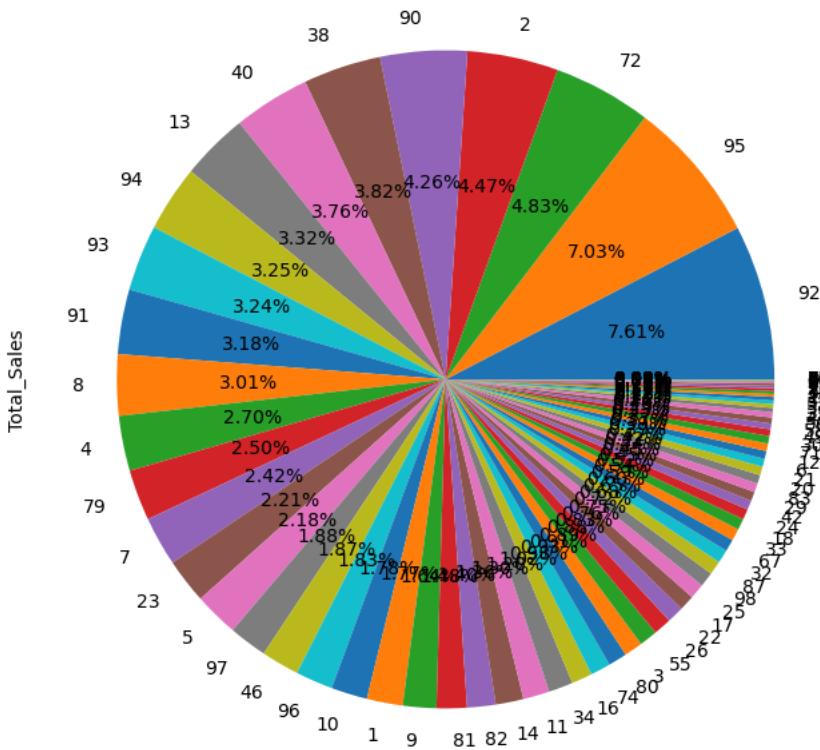
```
*****
*****
```

Store: 4

## Total Sales by Dept



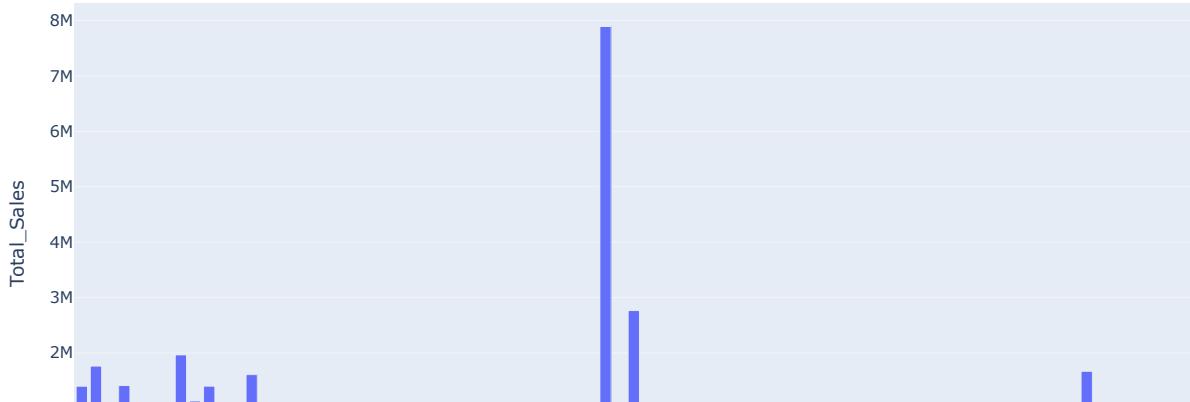
## Total Sales by Dept

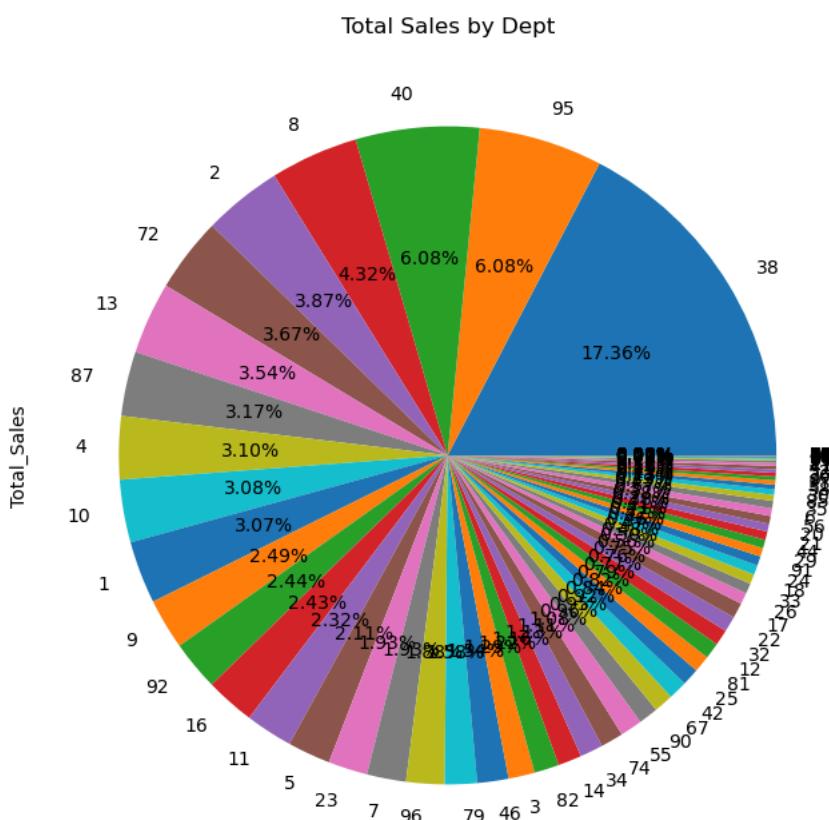
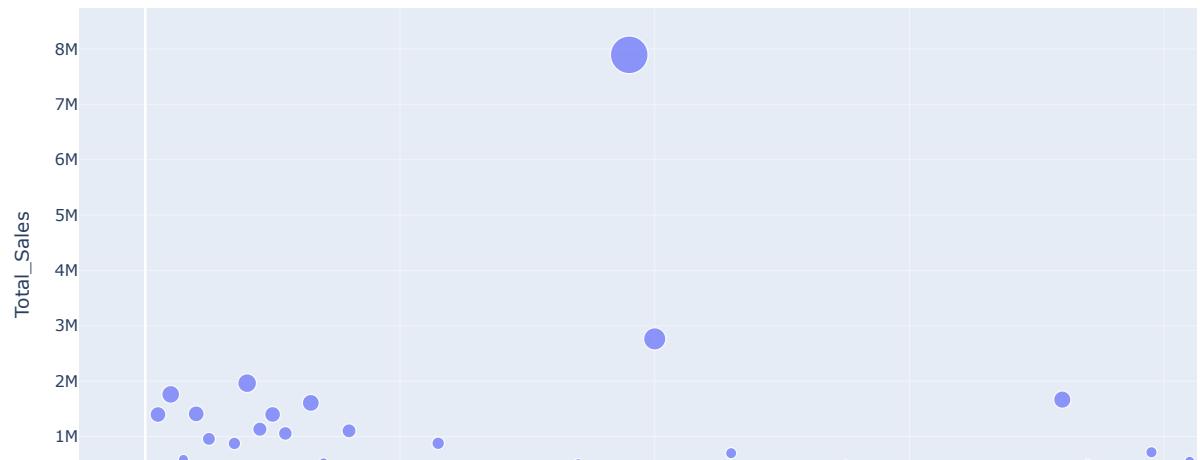


\*\*\*\*\*  
\*\*\*\*\*

Store: 5

## Total Sales by Dept

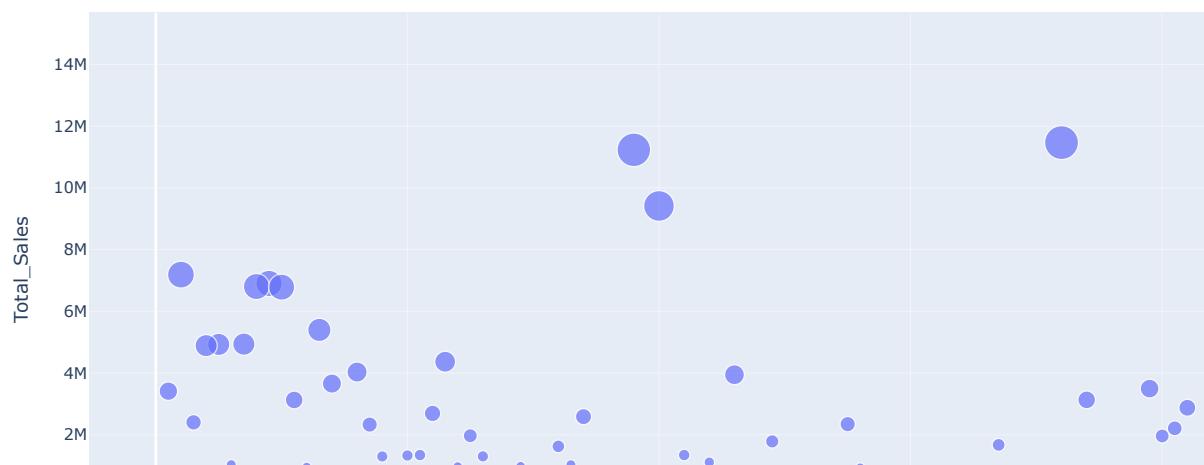
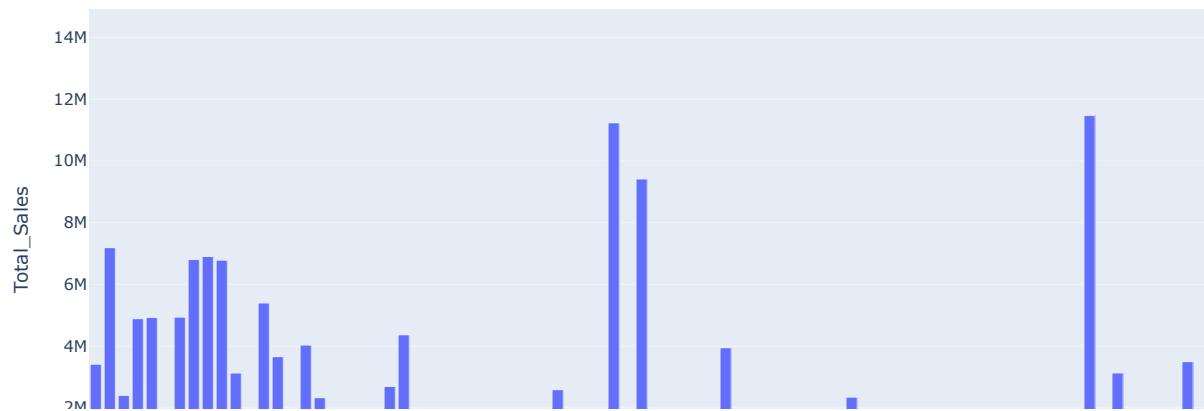




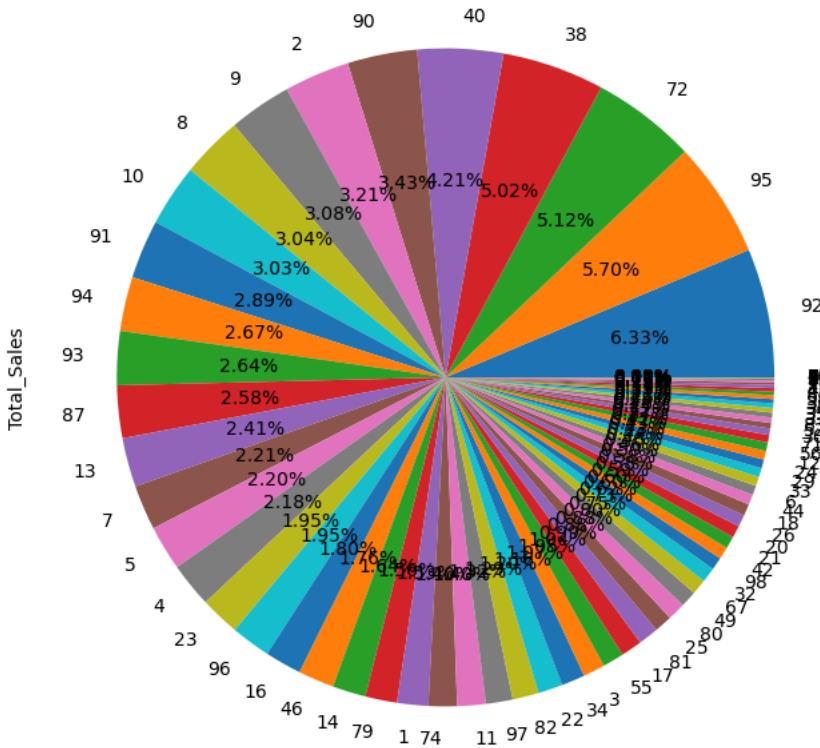
\*\*\*\*\*
\*\*\*\*\*  
\*\*\*\*\*

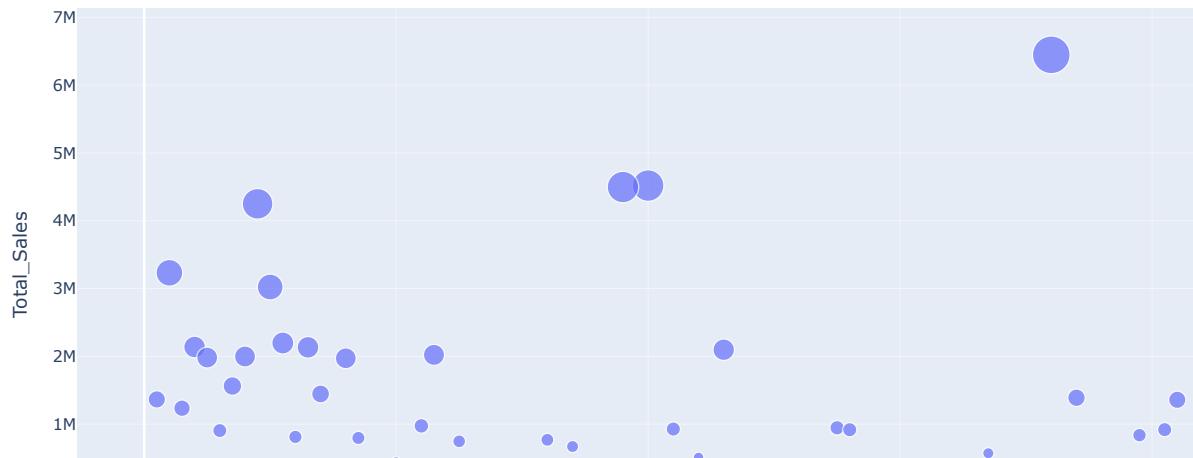
Store: 6

## Total Sales by Dept

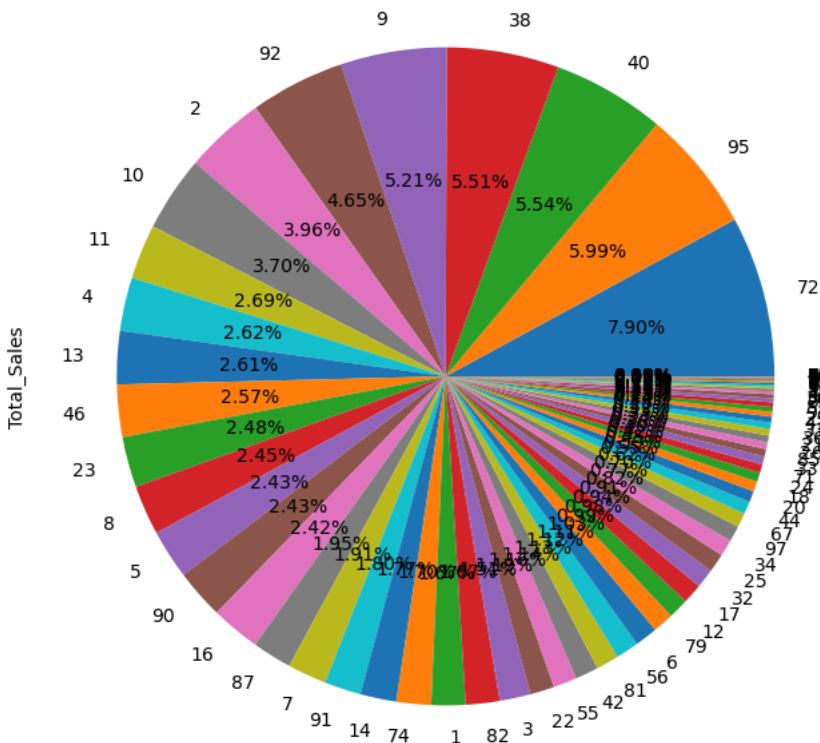


## Total Sales by Dept





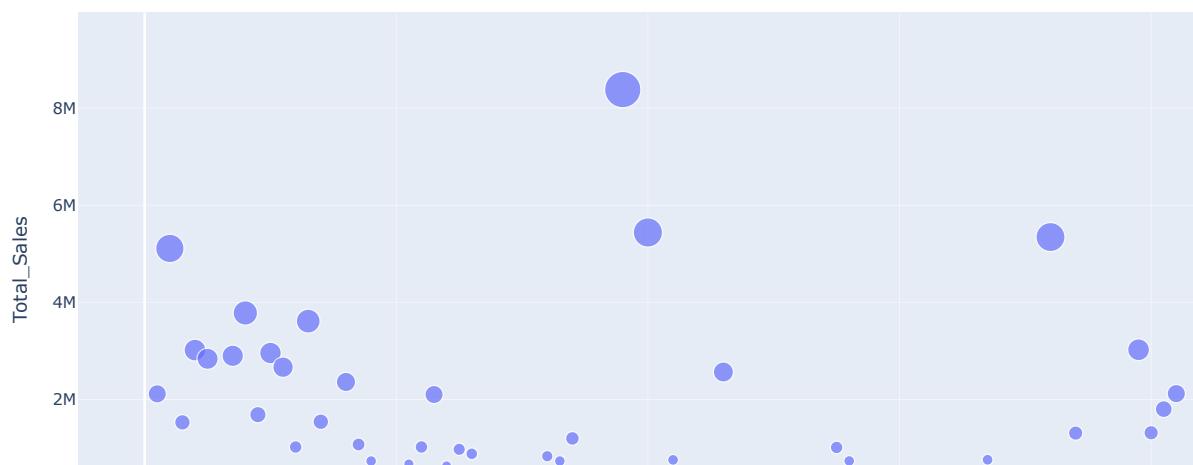
Total Sales by Dept



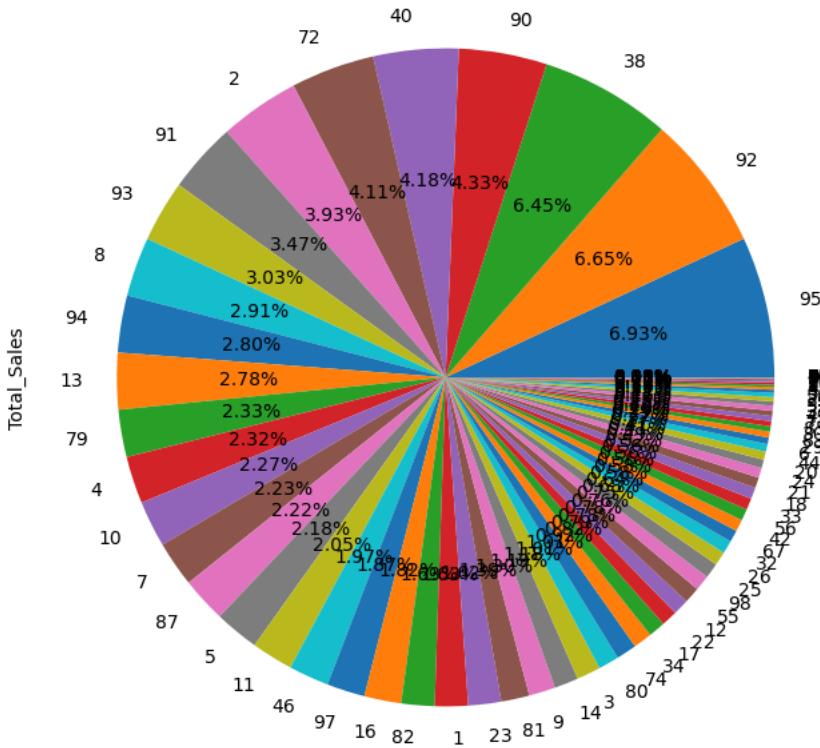
\*\*\*\*\*  
\*\*\*\*\*

Store: 8

## Total Sales by Dept



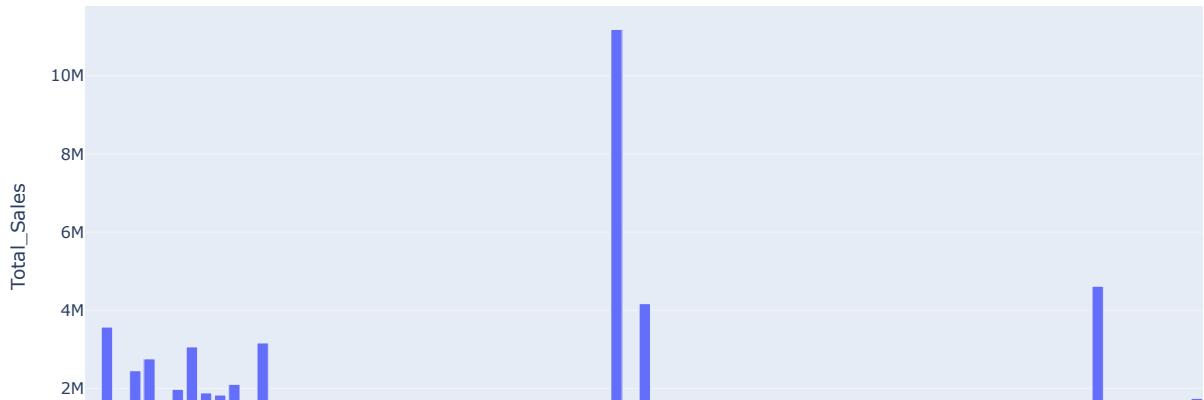
## Total Sales by Dept

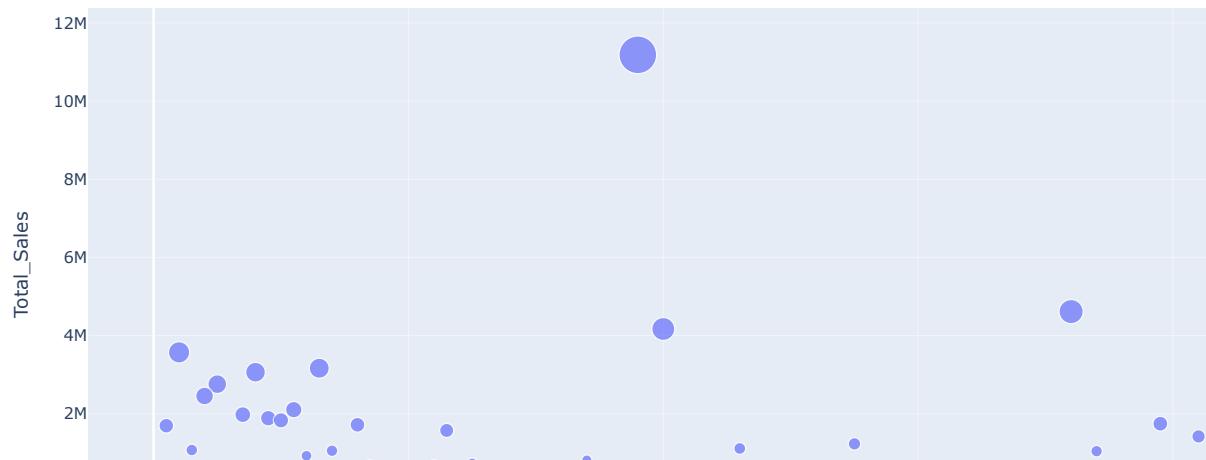


\*\*\*\*\*  
\*\*\*\*\*

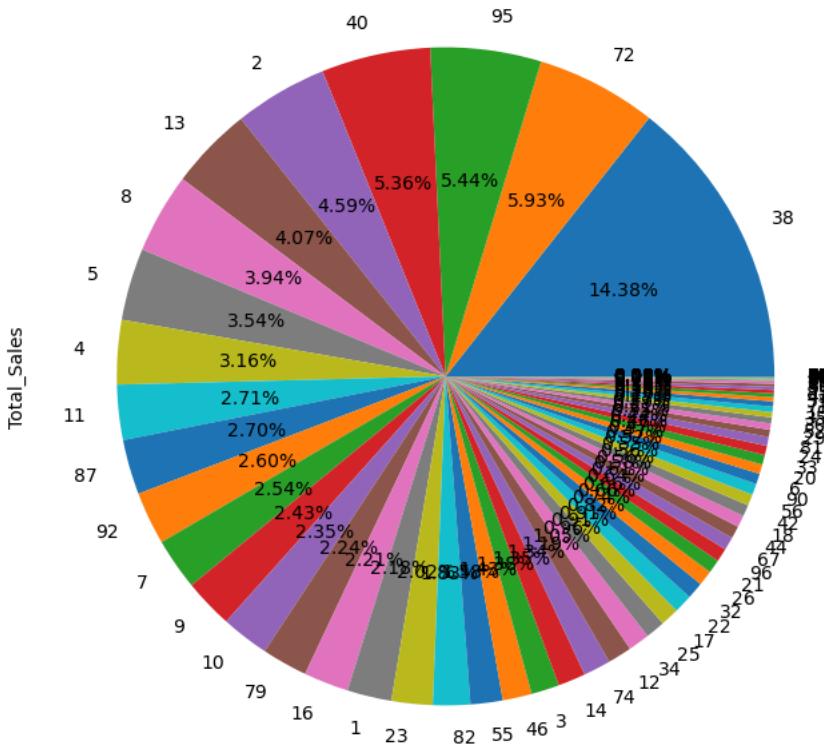
Store: 9

## Total Sales by Dept





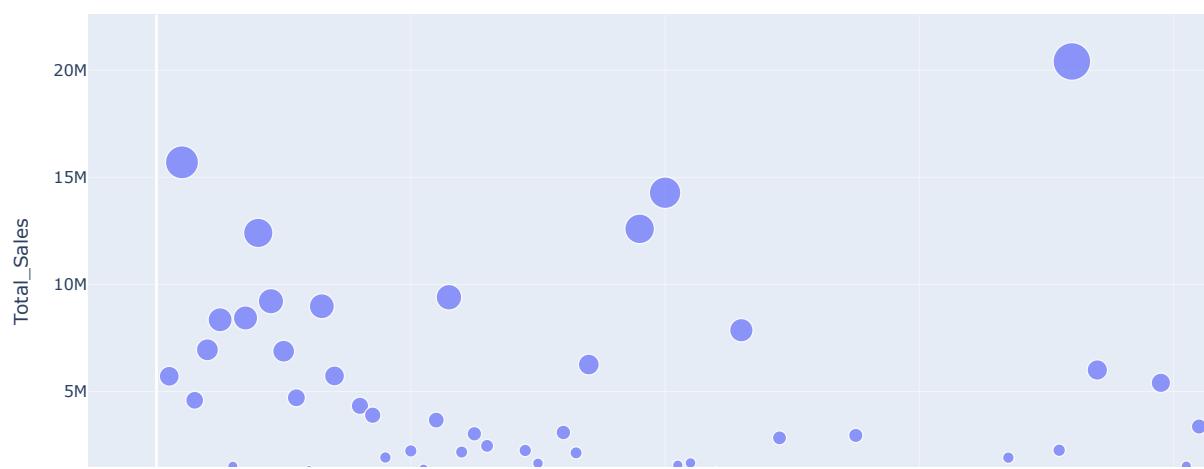
Total Sales by Dept



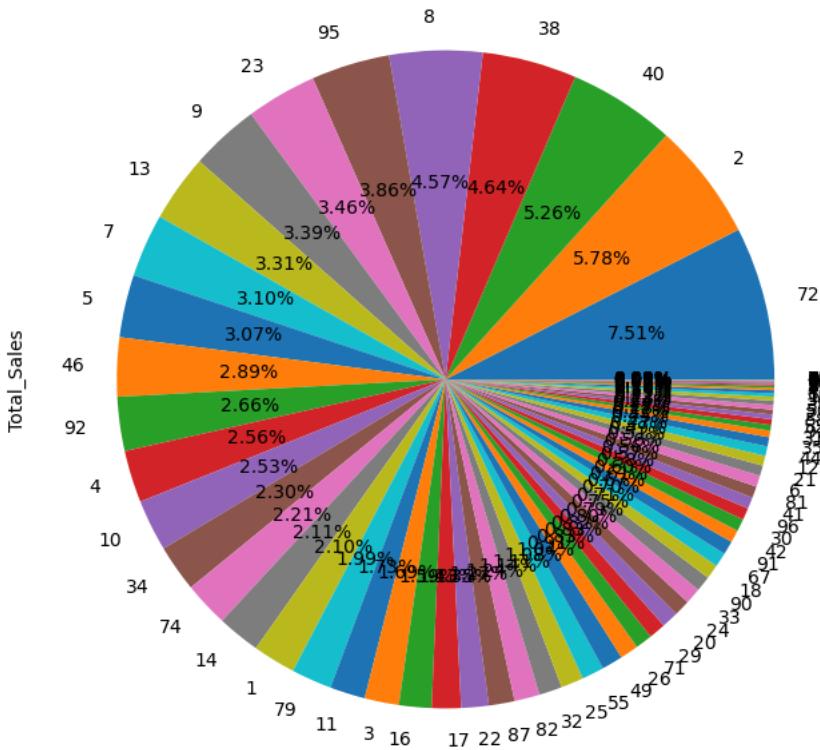
```
*****
*****
```

Store: 10

## Total Sales by Dept



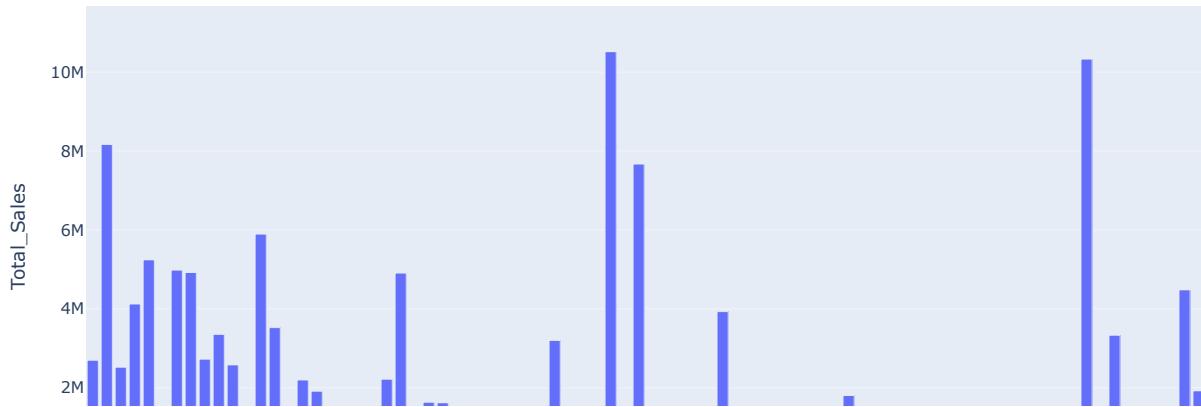
## Total Sales by Dept

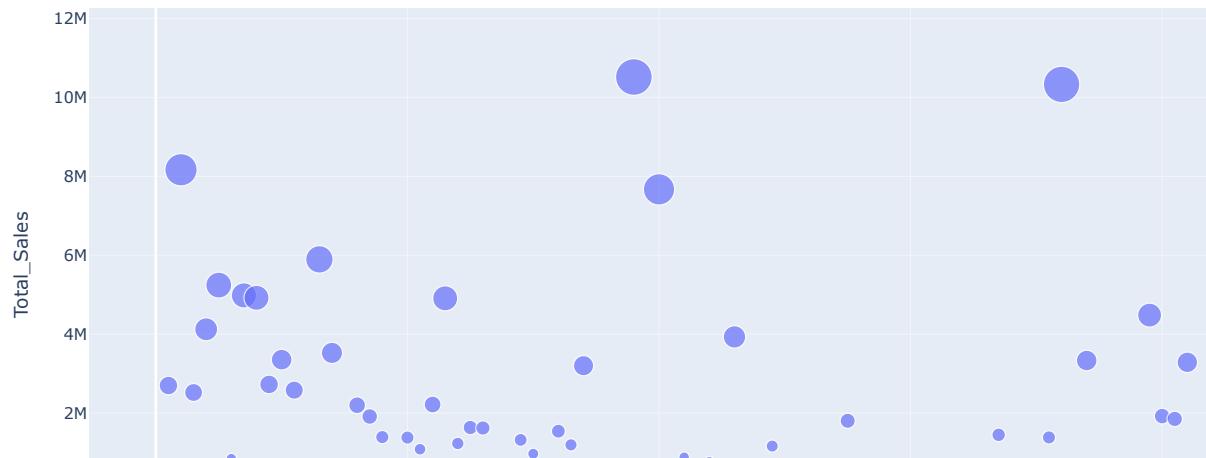


\*\*\*\*\*  
\*\*\*\*\*

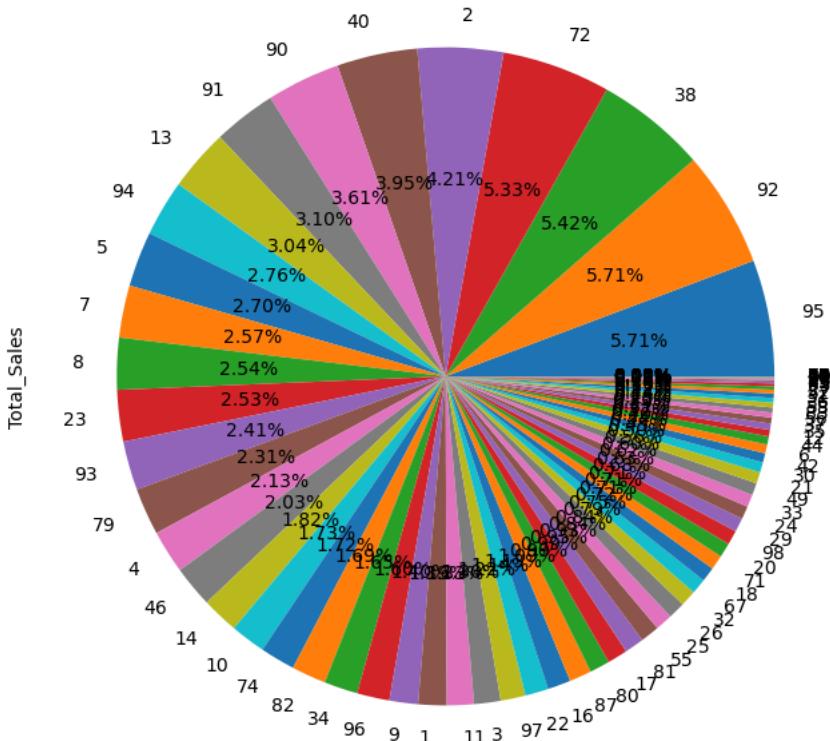
Store: 11

## Total Sales by Dept





Total Sales by Dept



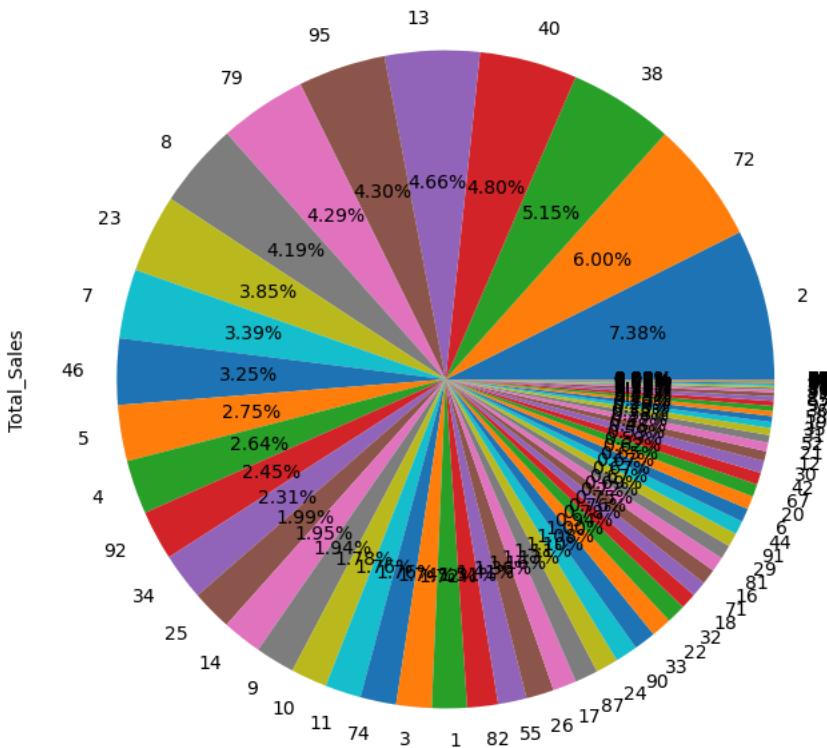
```
*****
*****
```

Store: 12

## Total Sales by Dept



## Total Sales by Dept



\*\*\*\*\*
\*\*\*\*\*

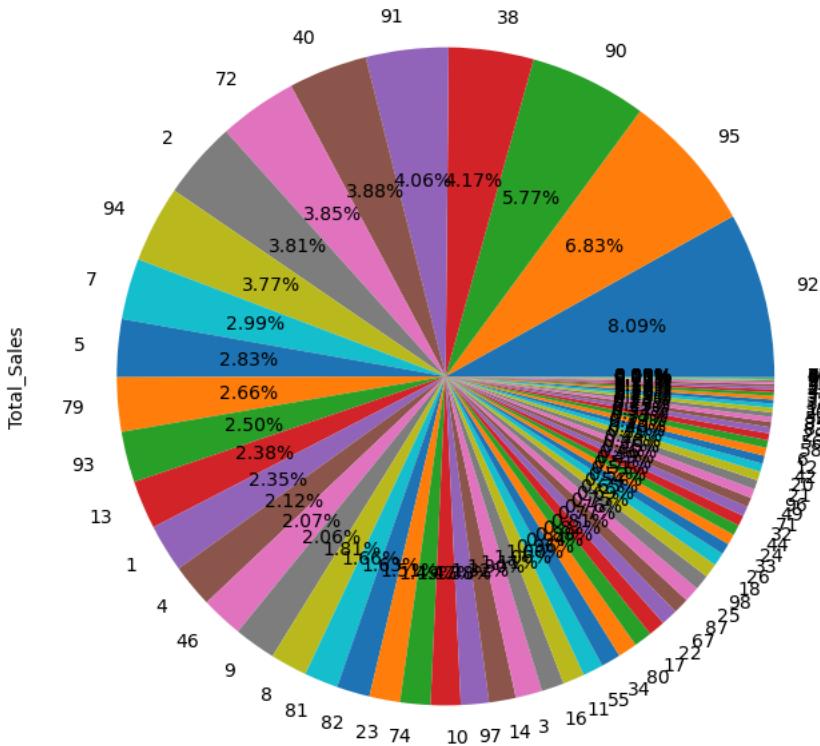
Store: 13

## Total Sales by Dept





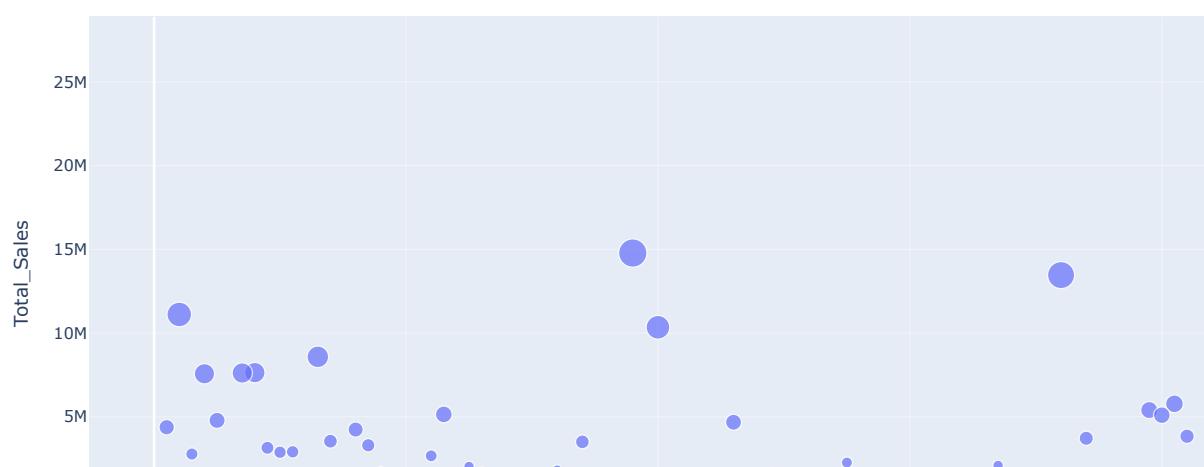
Total Sales by Dept



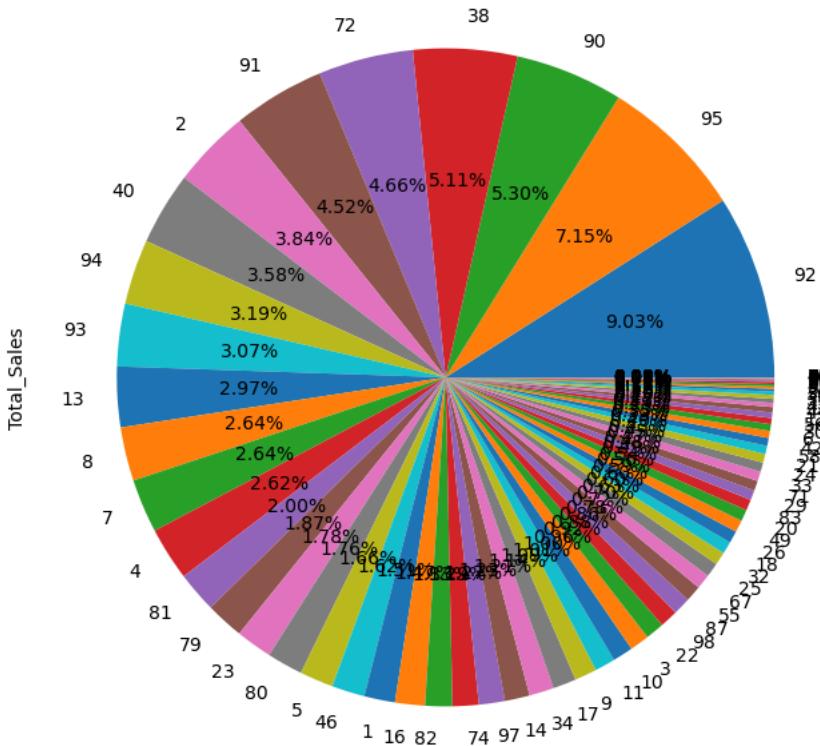
```
*****
*****
```

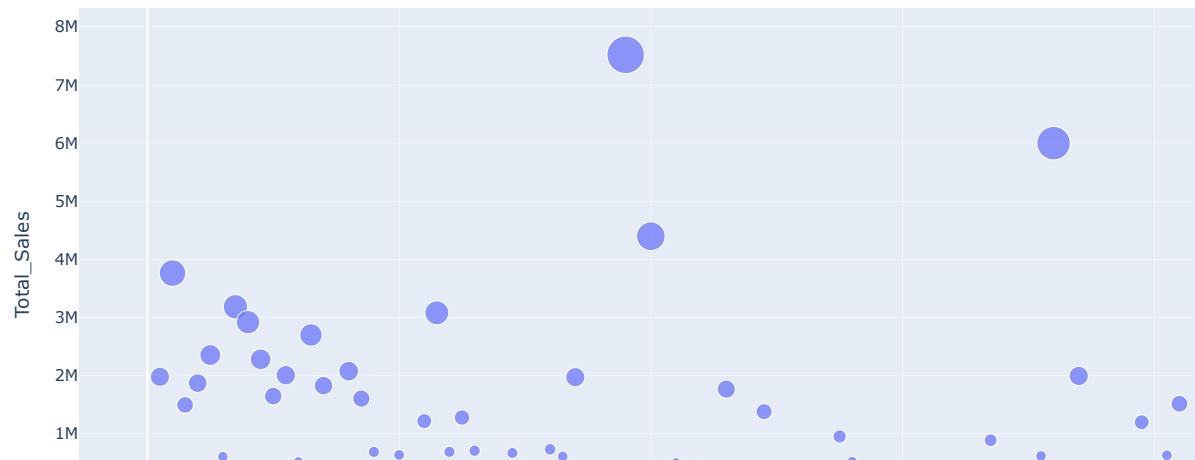
Store: 14

## Total Sales by Dept

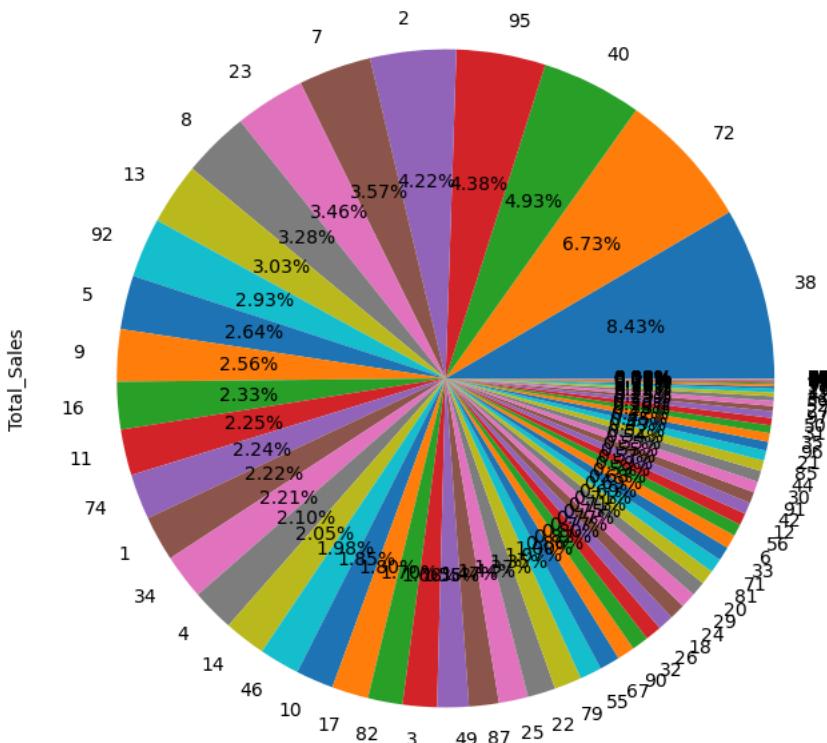


## Total Sales by Dept





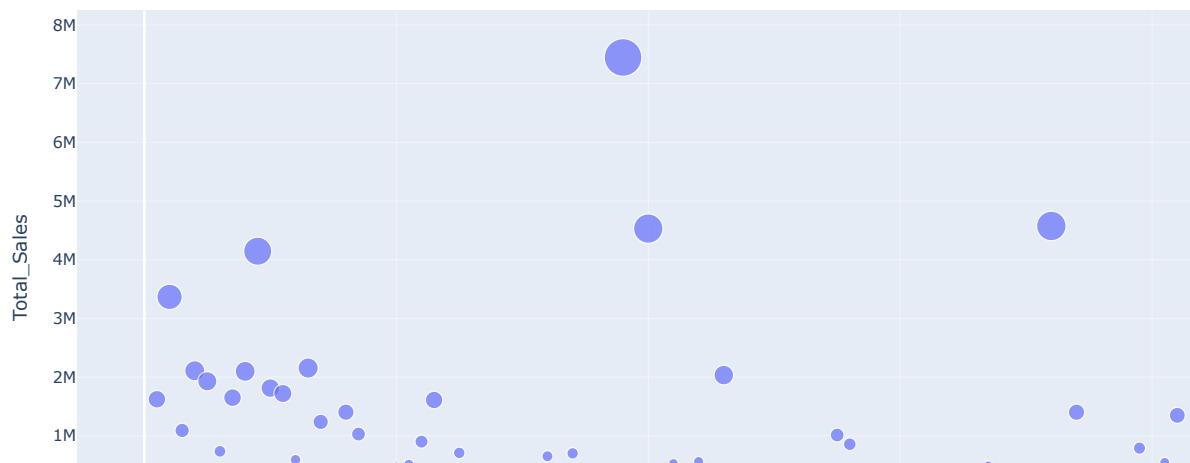
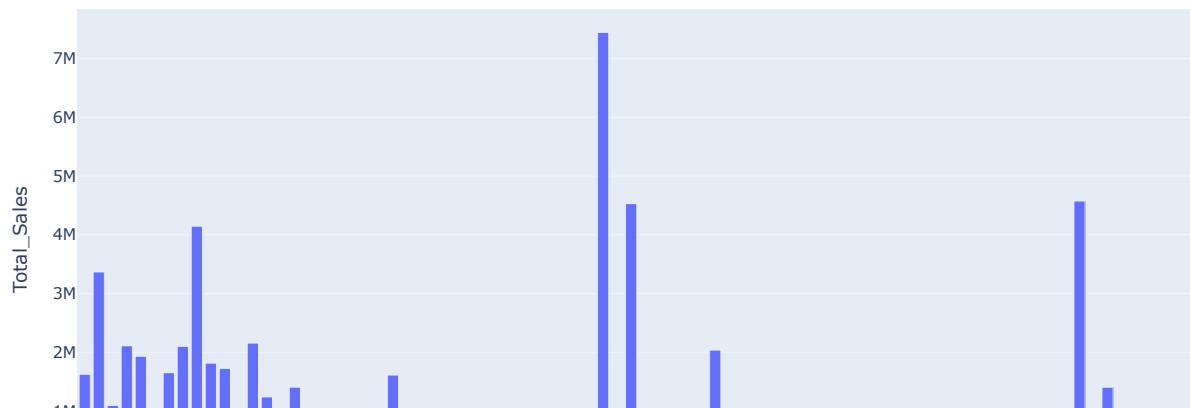
Total Sales by Dept



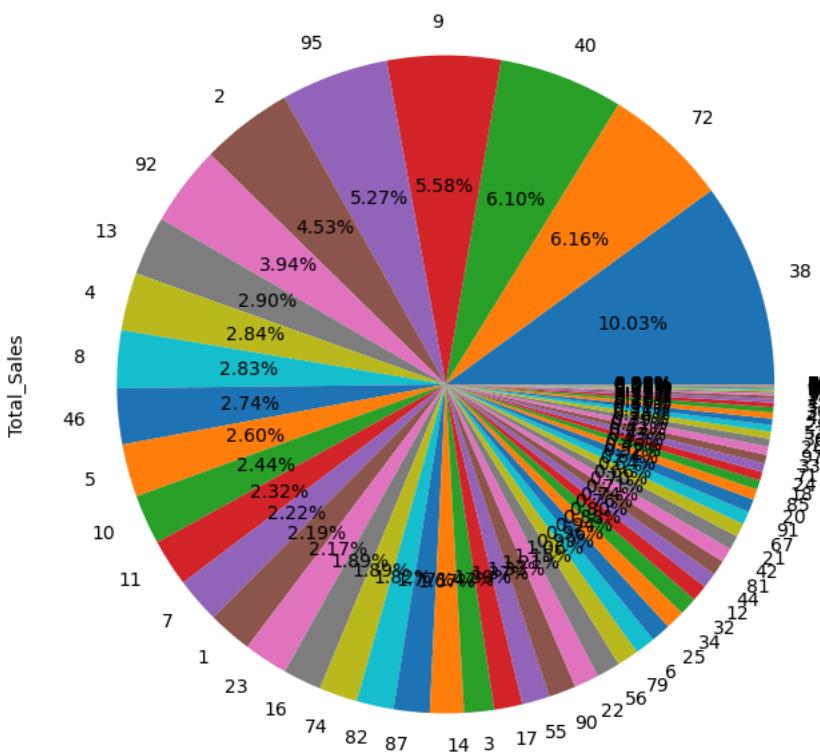
\*\*\*\*\*
\*\*\*\*\*  
\*\*\*\*\*

Store: 16

## Total Sales by Dept



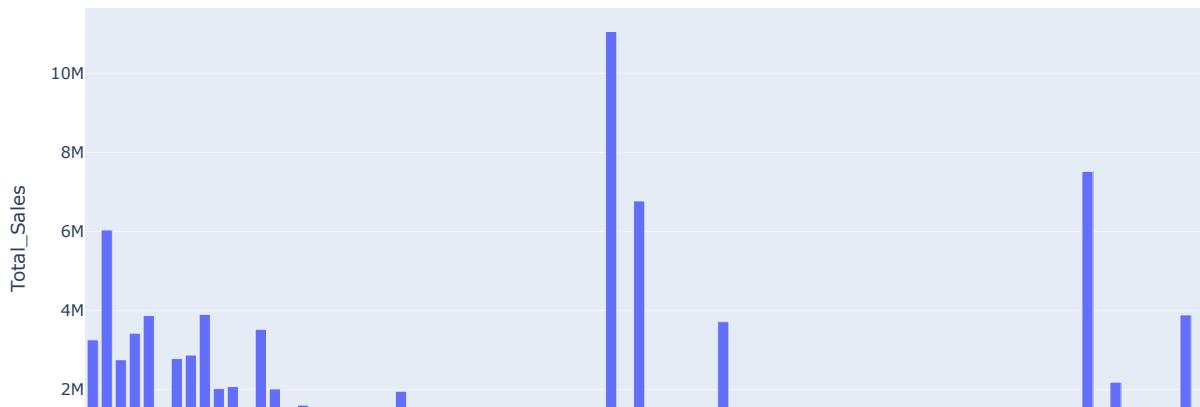
## Total Sales by Dept

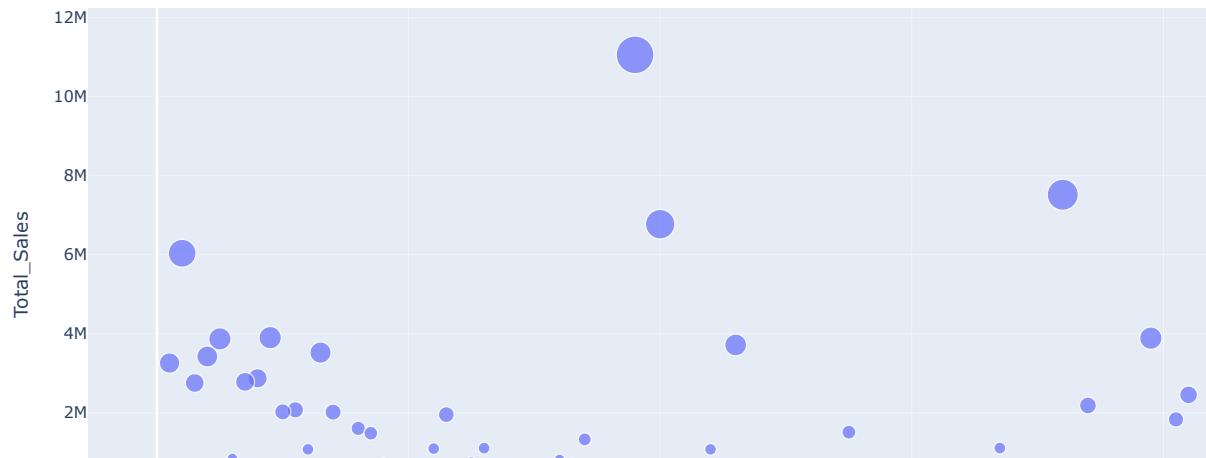


\*\*\*\*\*  
\*\*\*\*\*

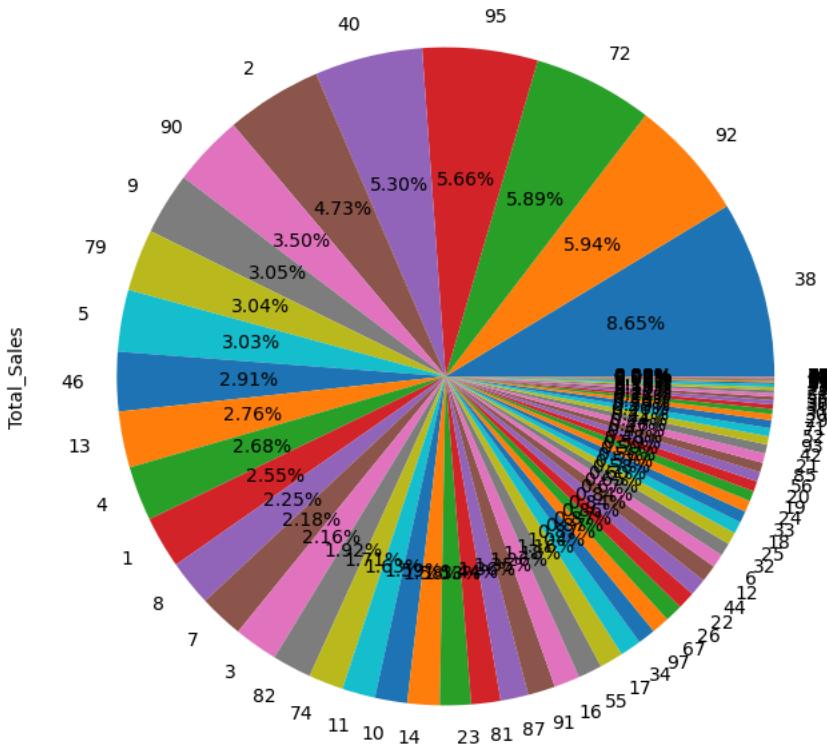
Store: 17

## Total Sales by Dept





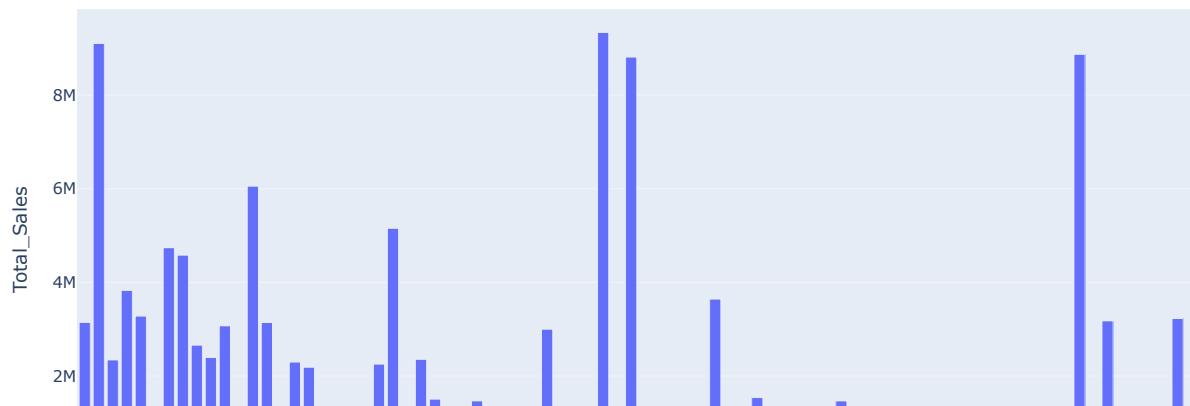
Total Sales by Dept



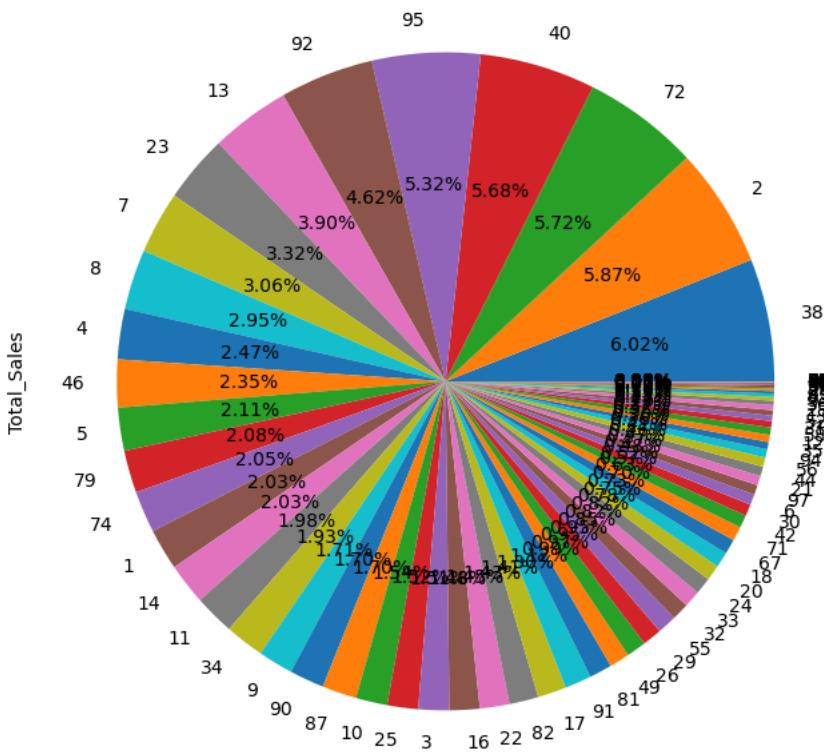
\*\*\*\*\*
\*\*\*\*\*  
\*\*\*\*\*

Store: 18

## Total Sales by Dept



## Total Sales by Dept

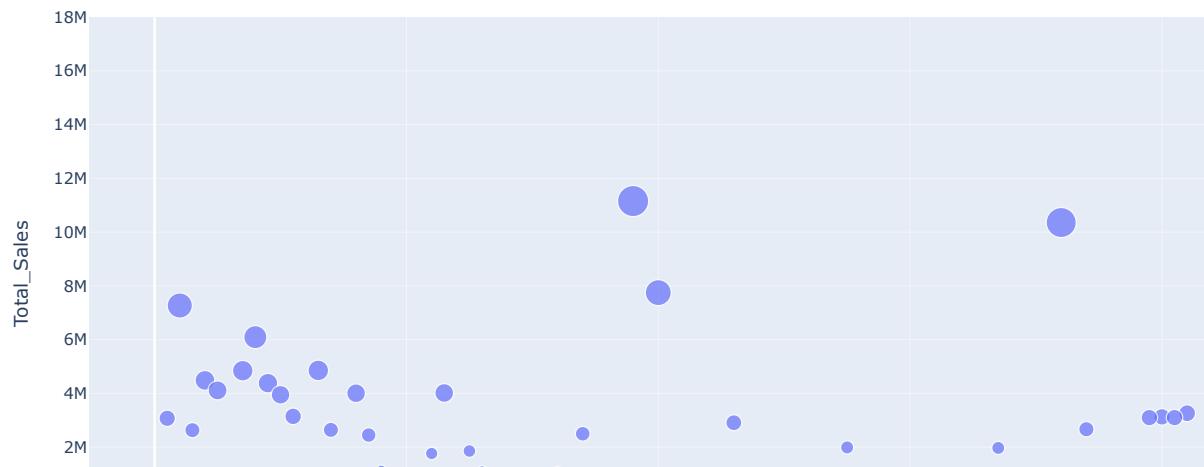


\*\*\*\*\*  
\*\*\*\*\*

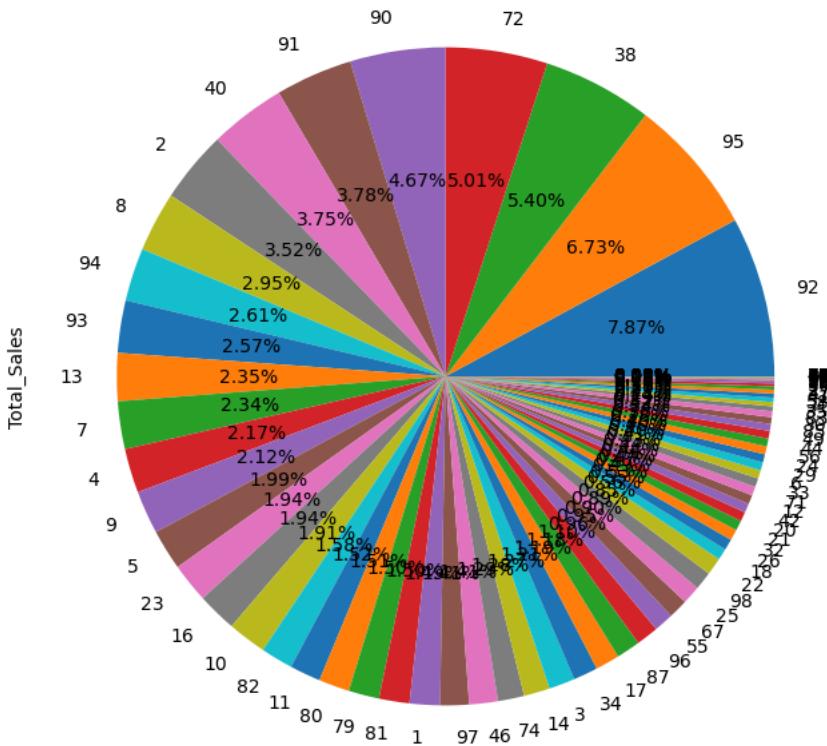
Store: 19

## Total Sales by Dept





## Total Sales by Dept



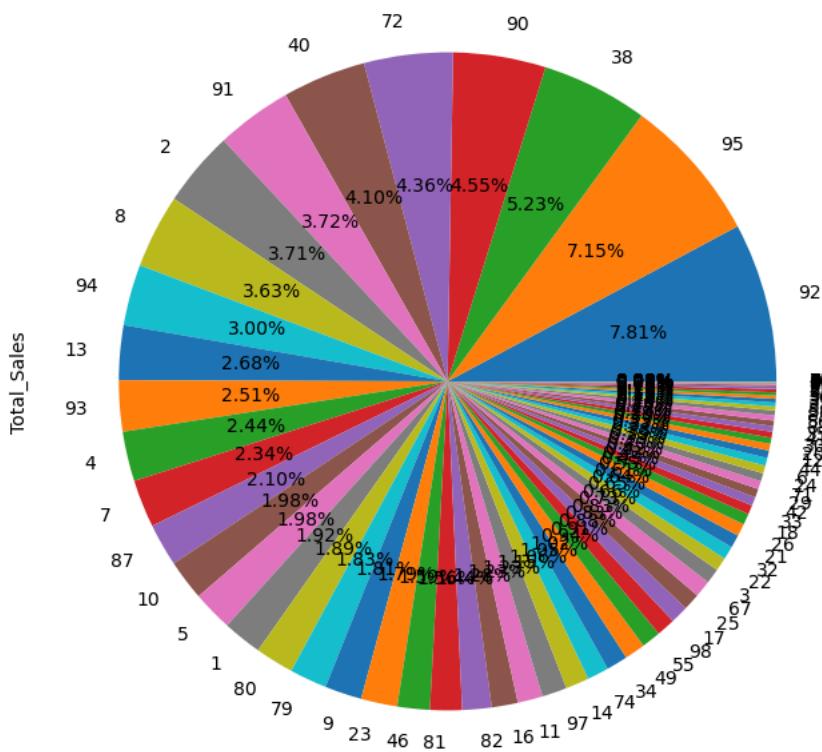
\*\*\*\*\*  
\*\*\*\*\*

Store: 20

## Total Sales by Dept



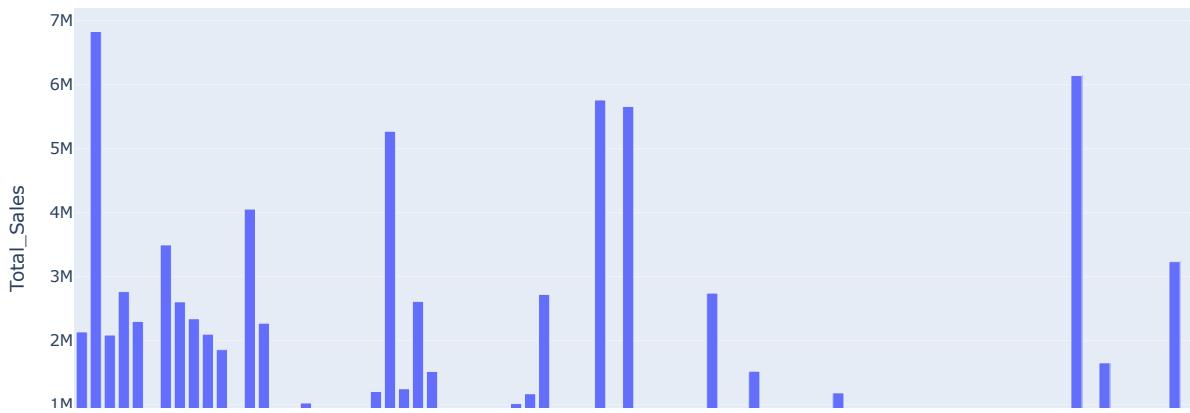
## Total Sales by Dept

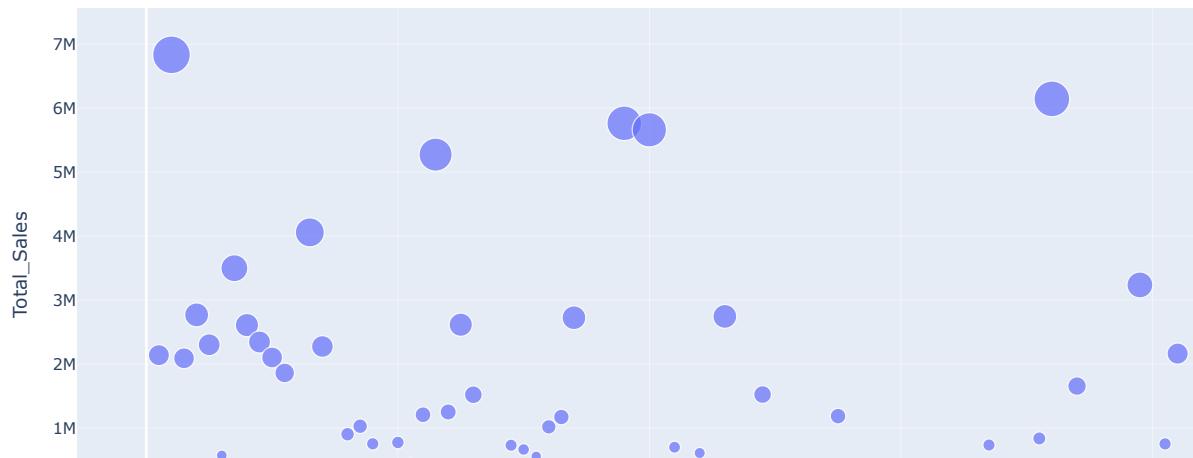


\*\*\*\*\*  
\*\*\*\*\*

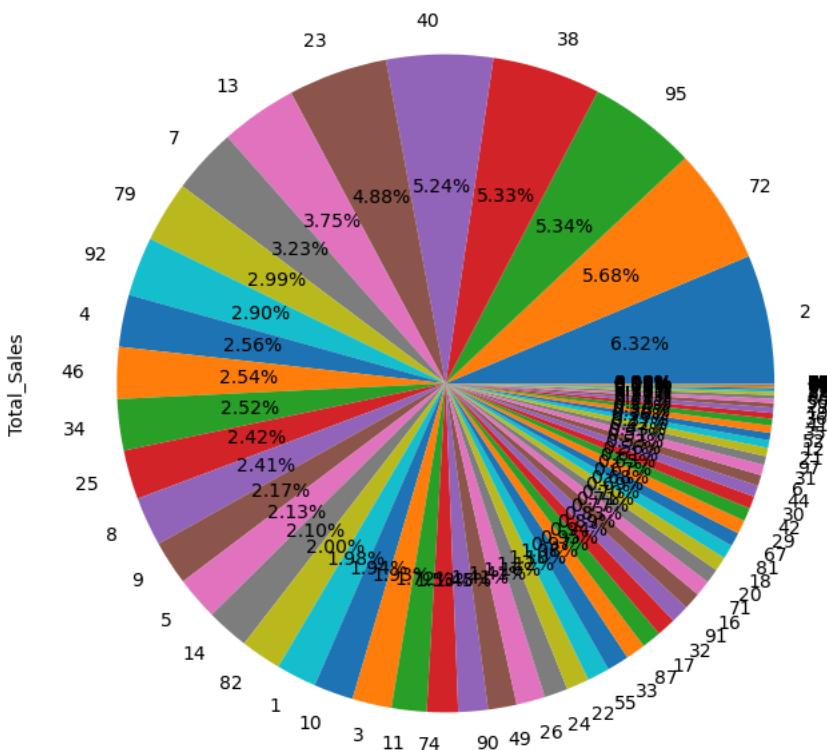
Store: 21

## Total Sales by Dept





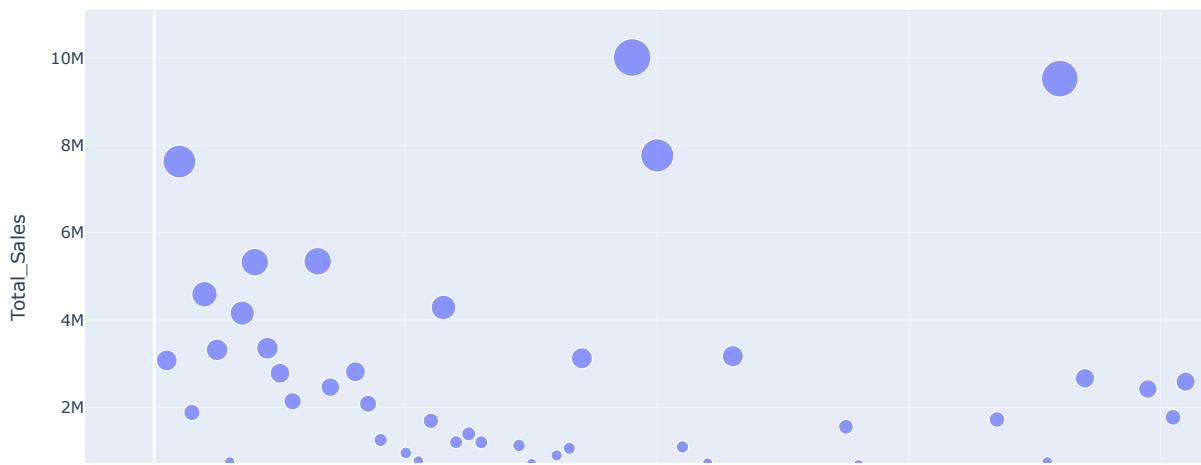
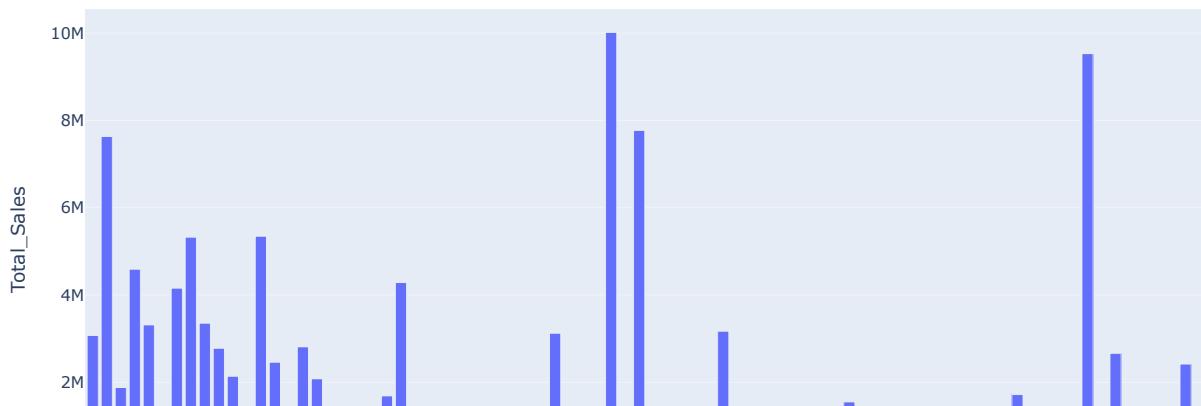
Total Sales by Dept



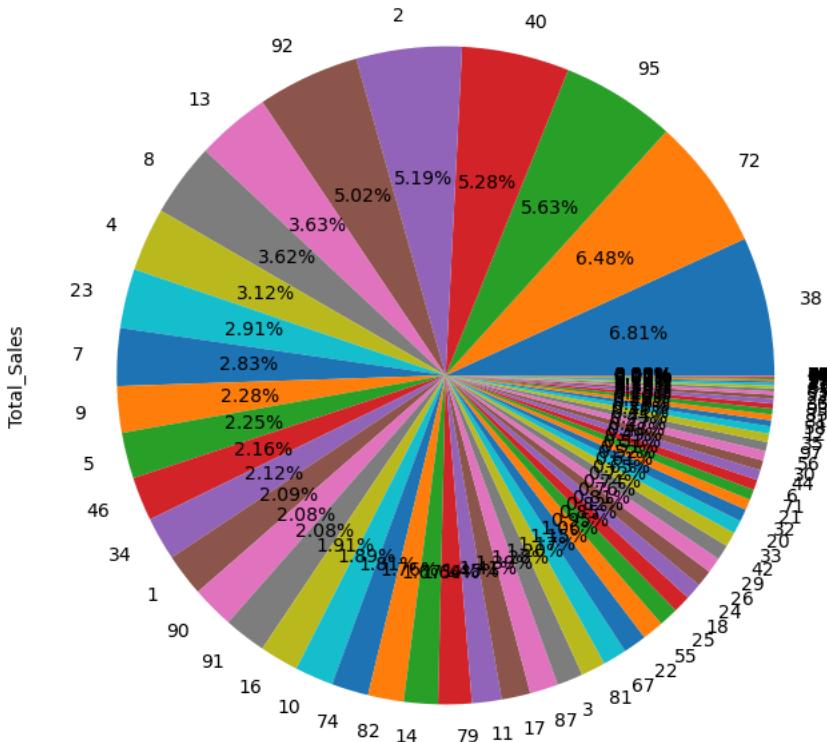
\*\*\*\*\*  
\*\*\*\*\*

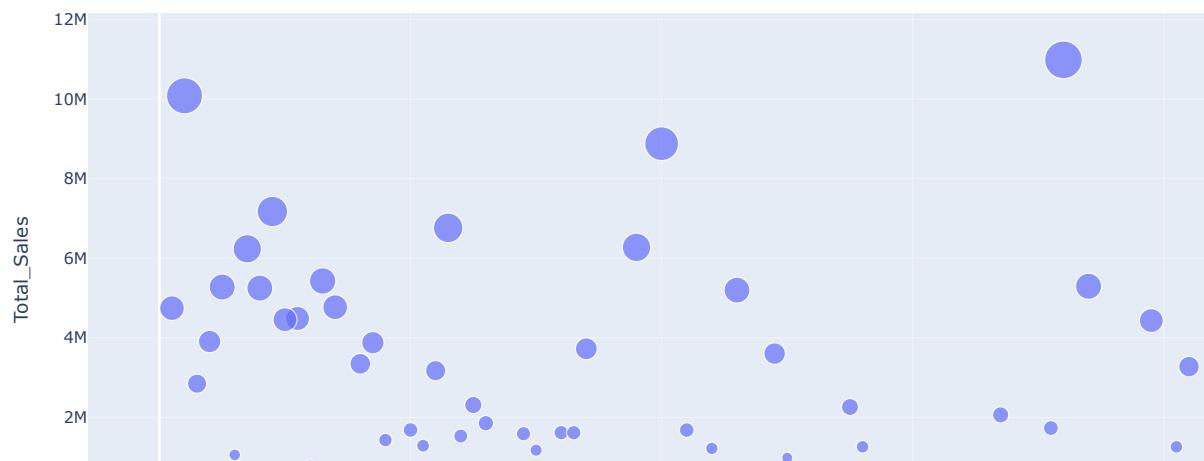
Store: 22

## Total Sales by Dept

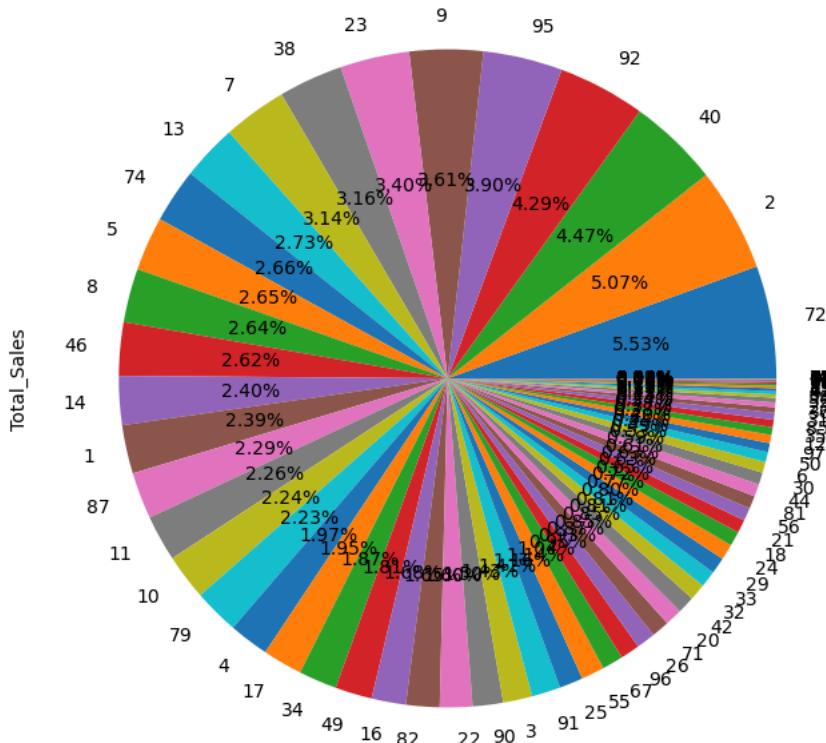


## Total Sales by Dept





Total Sales by Dept



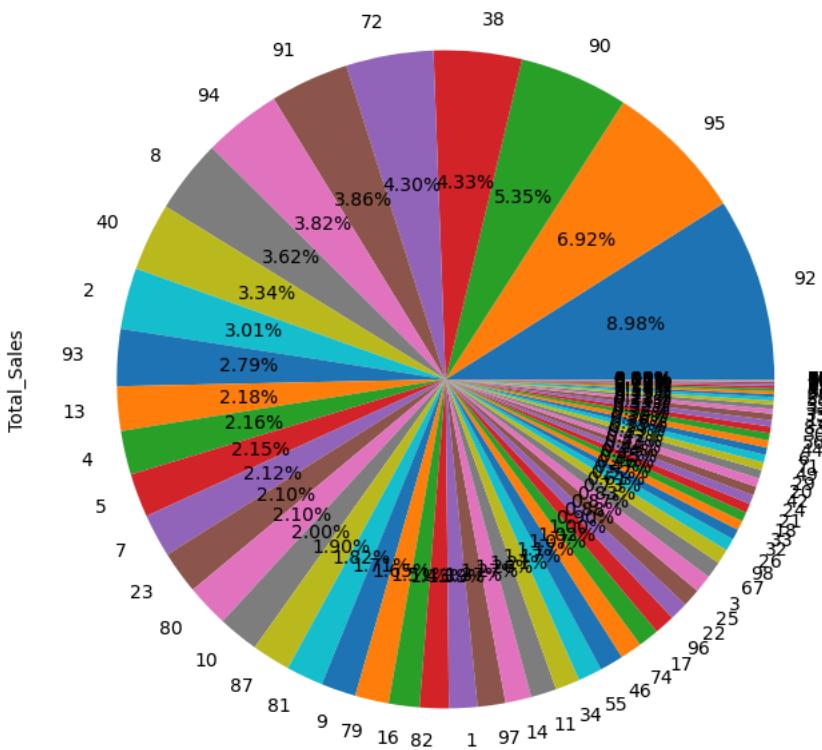
\*\*\*\*\*  
\*\*\*\*\*

Store: 24

## Total Sales by Dept



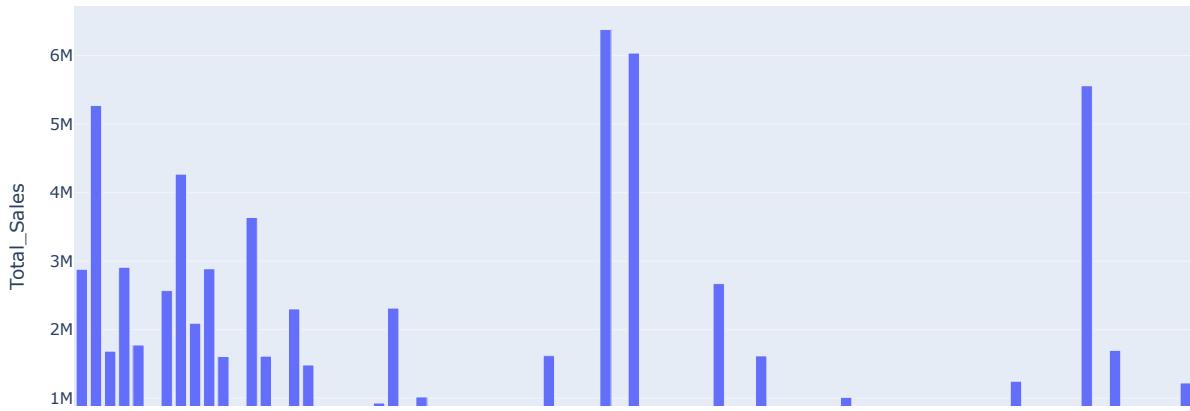
## Total Sales by Dept

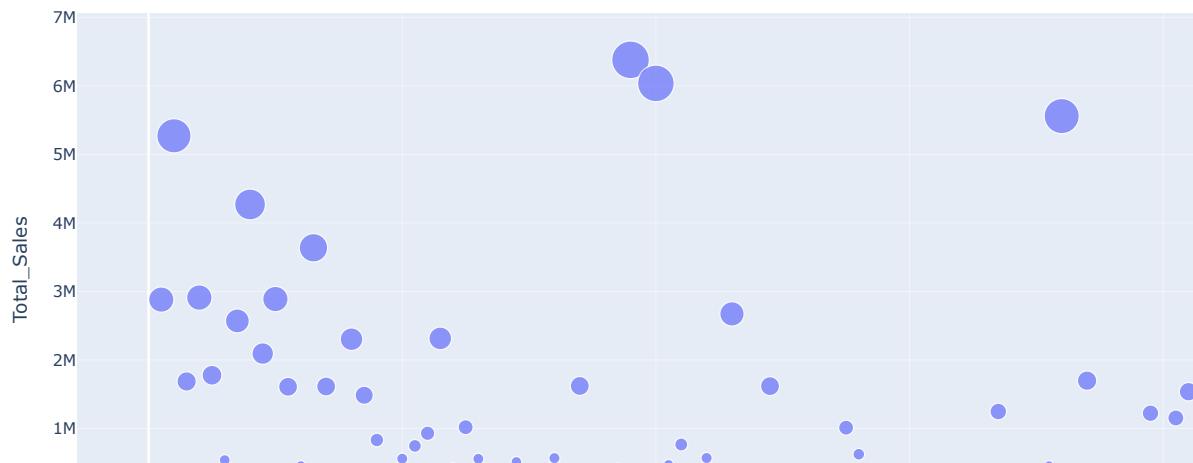


\*\*\*\*\*  
\*\*\*\*\*

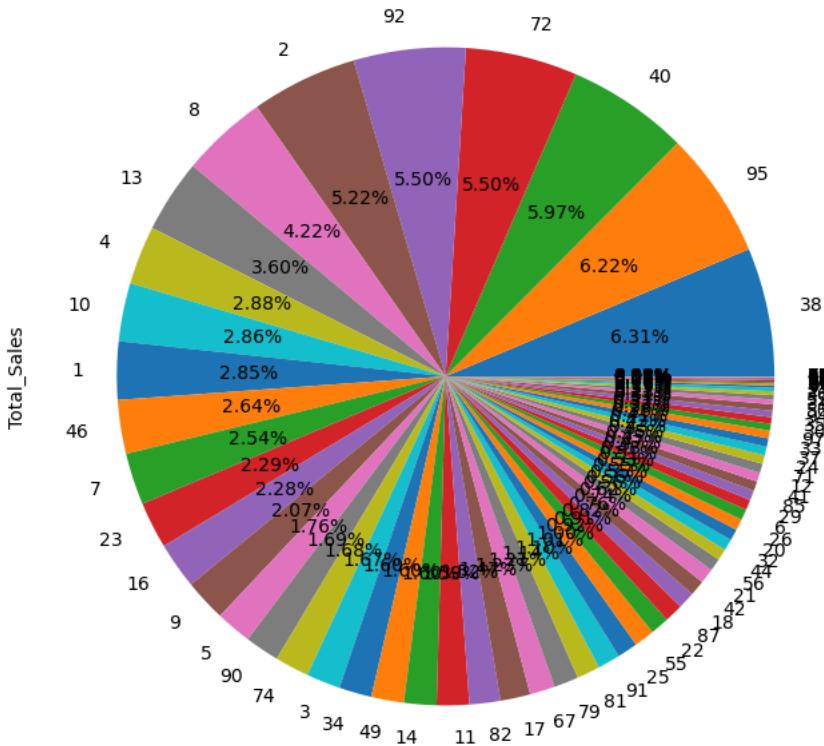
Store: 25

## Total Sales by Dept





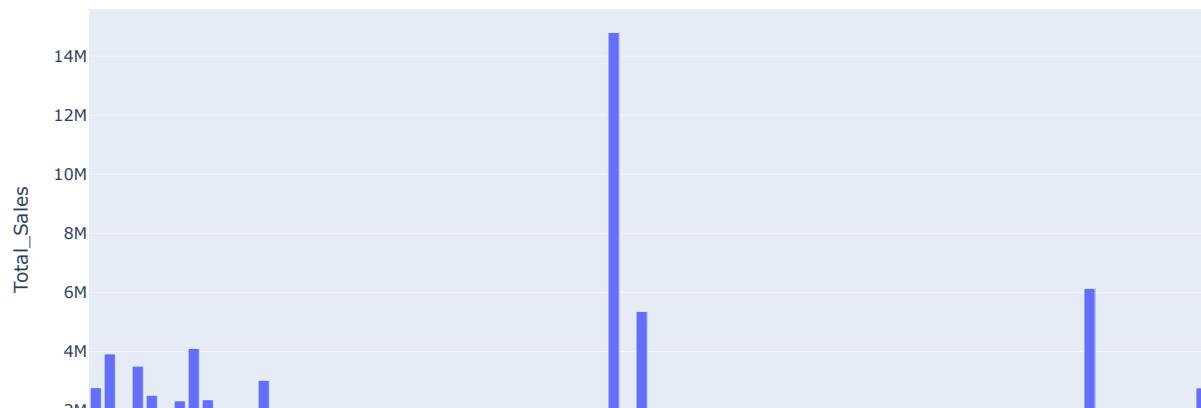
Total Sales by Dept



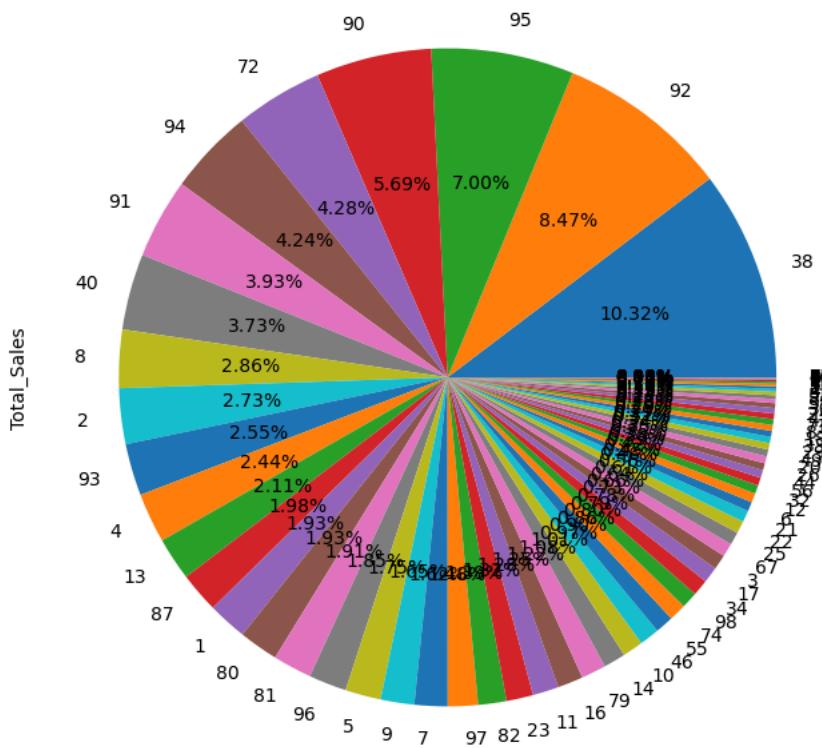
\*\*\*\*\*
\*\*\*\*\*  
\*\*\*\*\*

Store: 26

## Total Sales by Dept



## Total Sales by Dept



\*\*\*\*\*  
\*\*\*\*\*

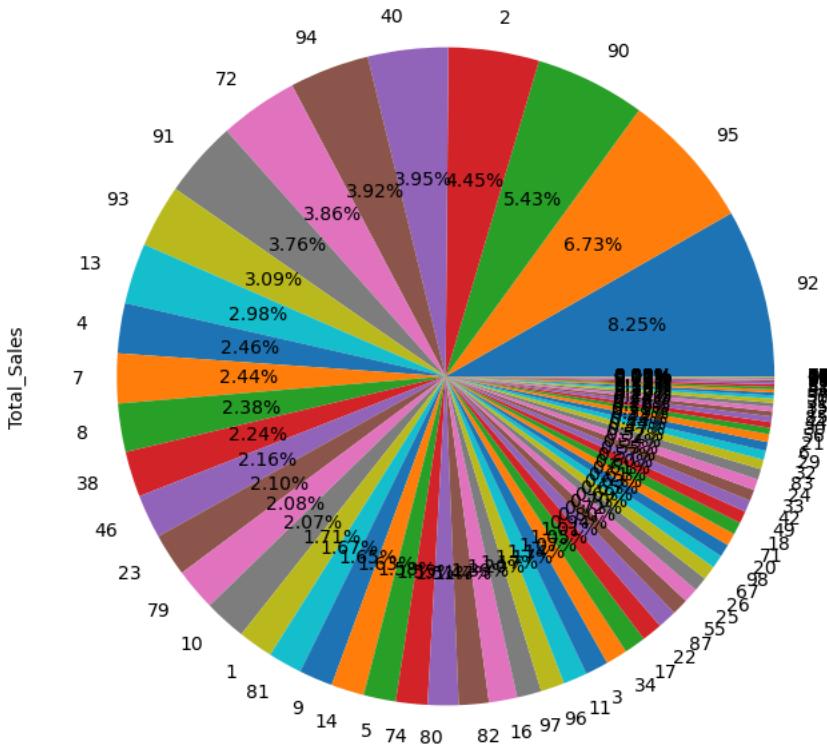
Store: 27

## Total Sales by Dept





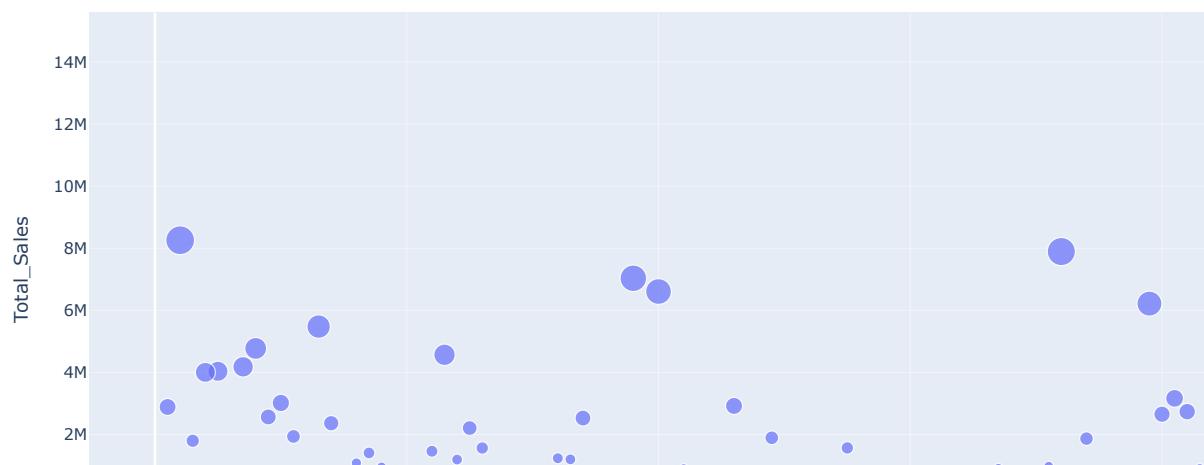
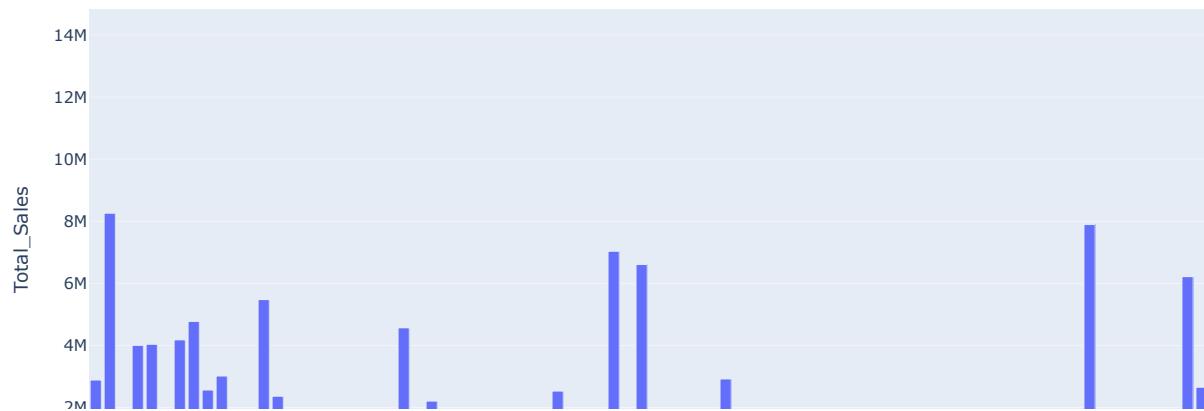
Total Sales by Dept



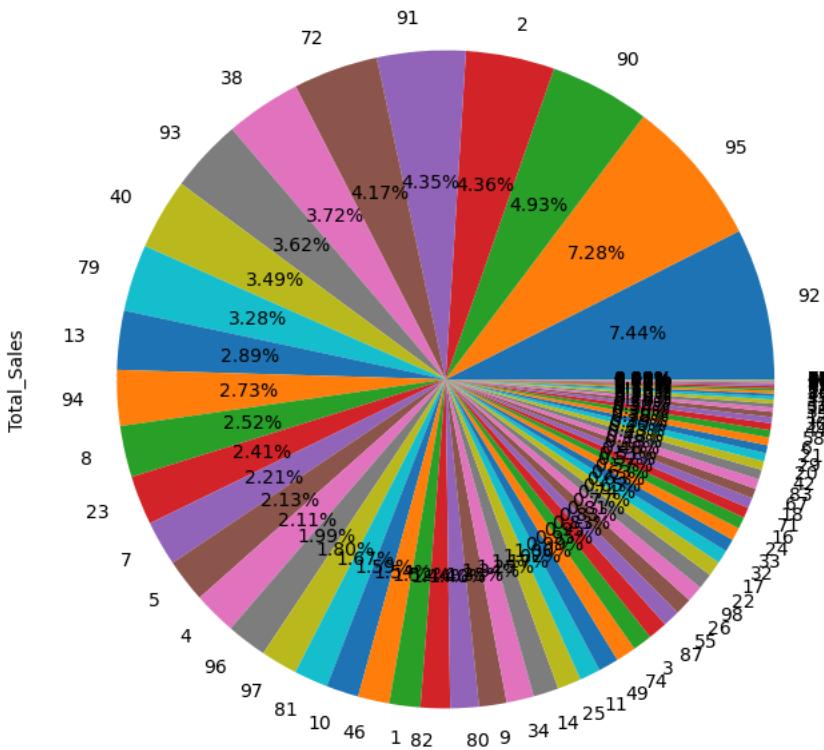
\*\*\*\*\*  
\*\*\*\*\*

Store: 28

## Total Sales by Dept



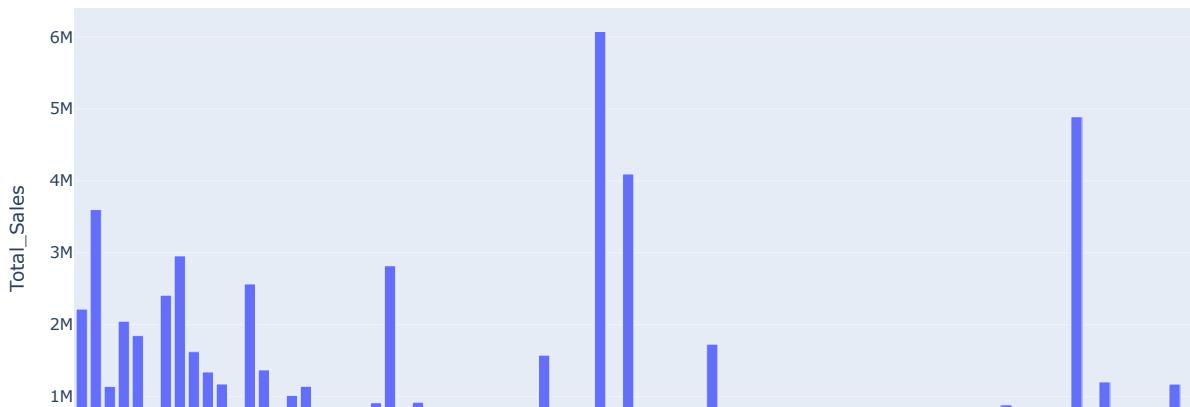
## Total Sales by Dept

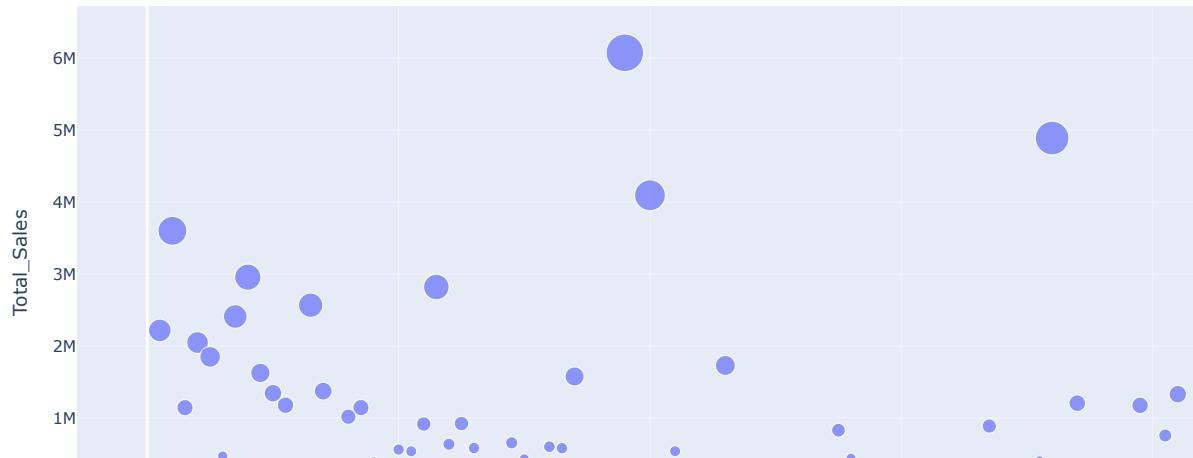


\*\*\*\*\*  
\*\*\*\*\*

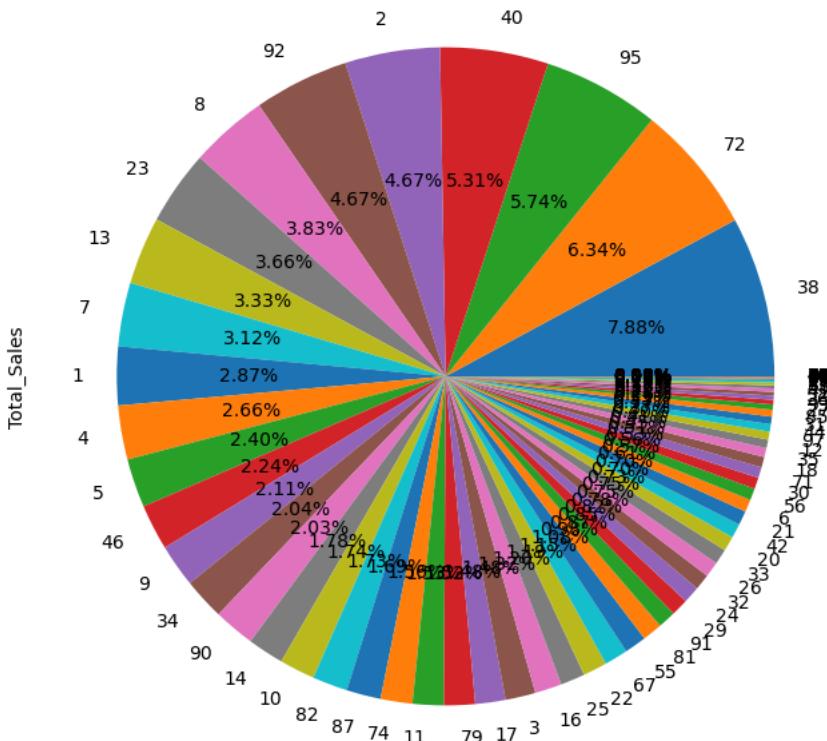
Store: 29

## Total Sales by Dept





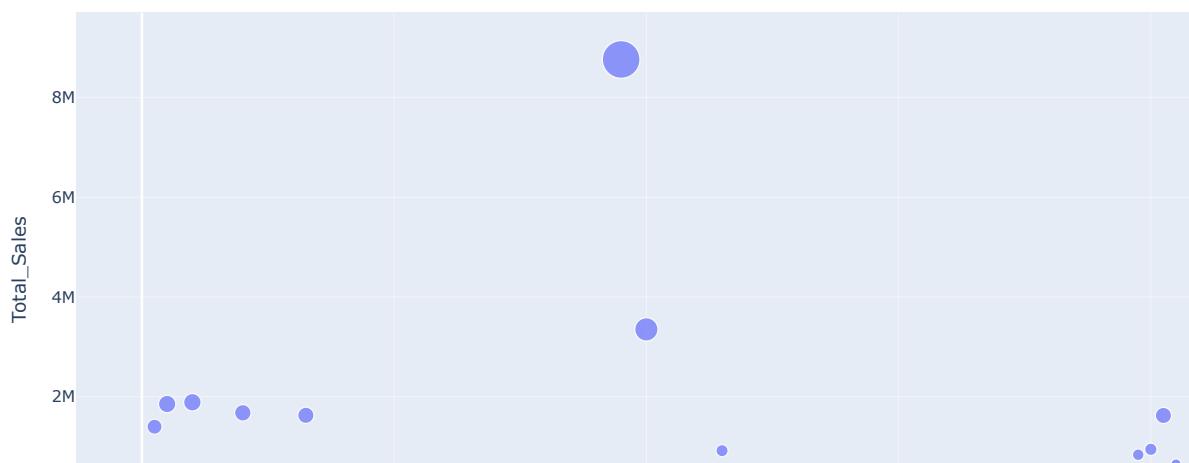
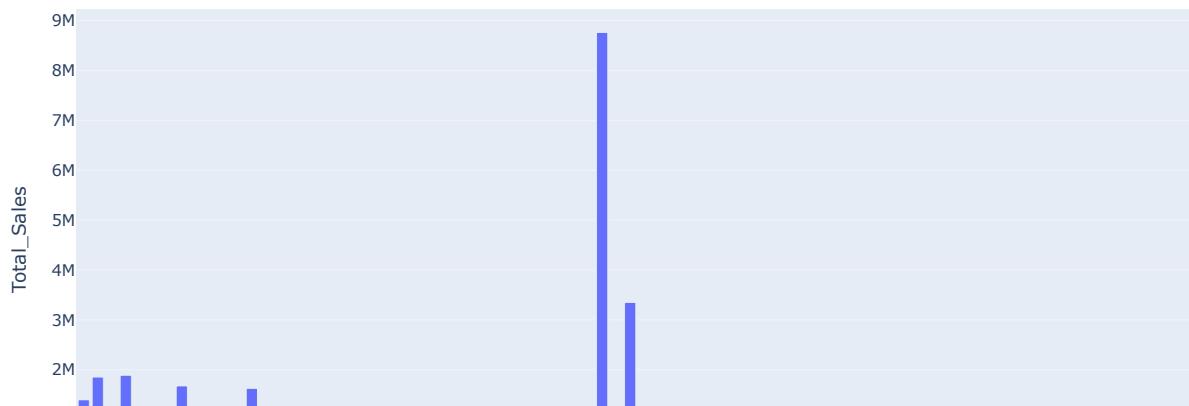
Total Sales by Dept



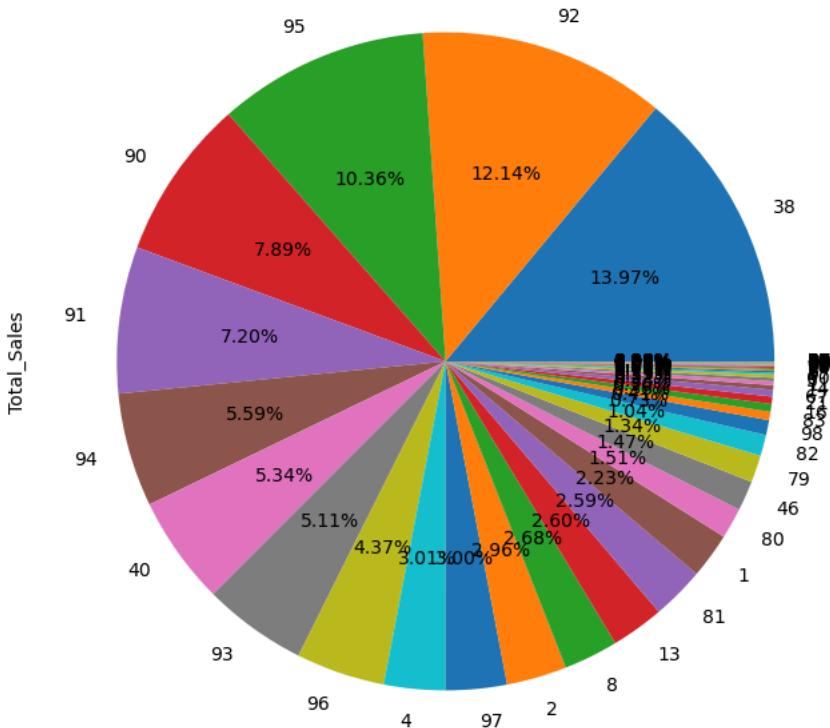
\*\*\*\*\*
\*\*\*\*\*  
\*\*\*\*\*

Store: 30

## Total Sales by Dept



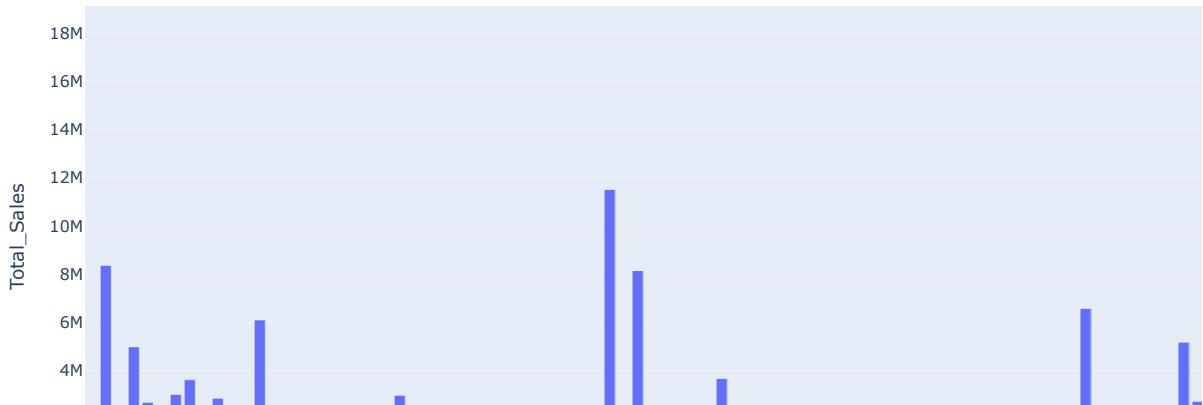
## Total Sales by Dept

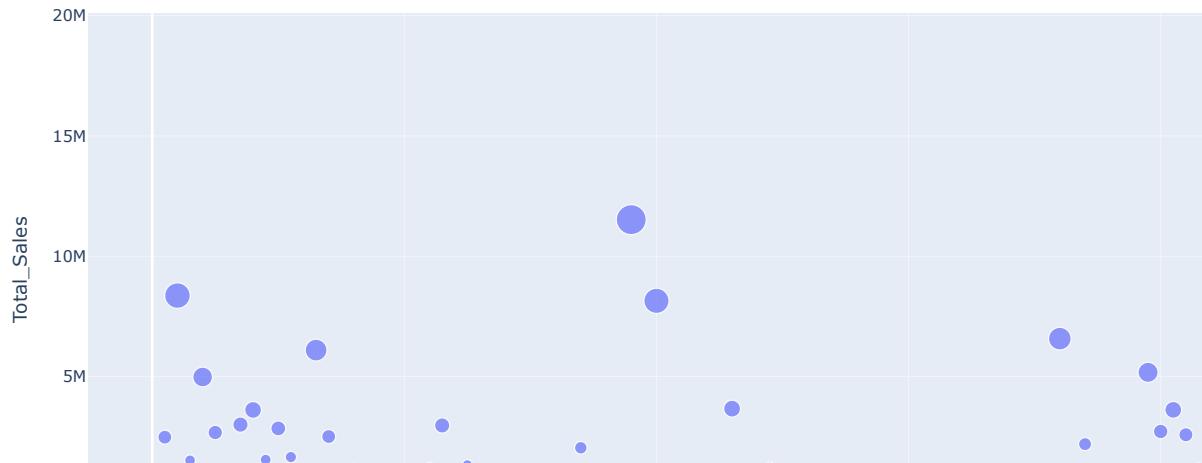


\*\*\*\*\*
\*\*\*\*\*

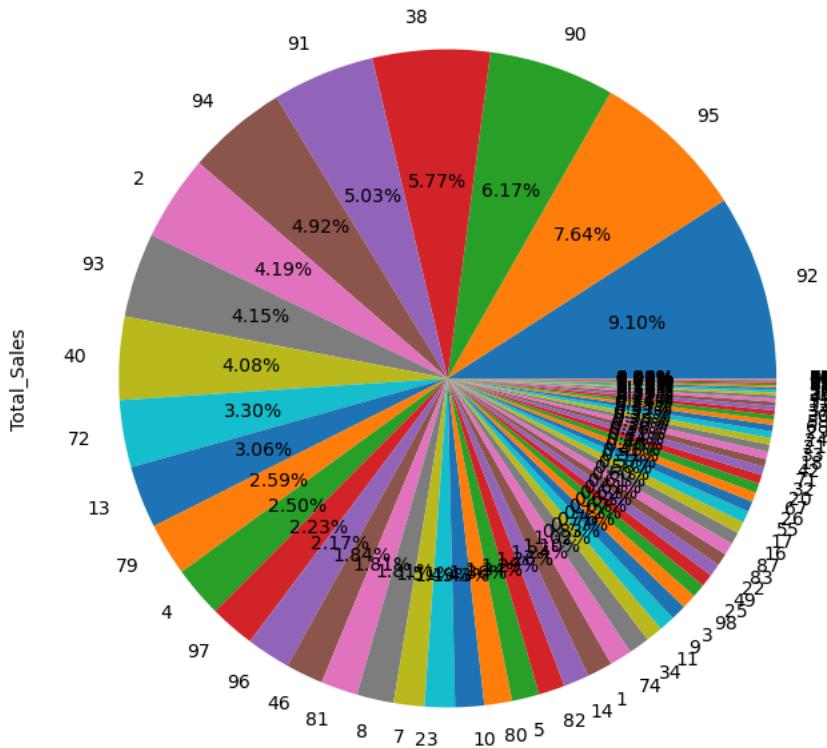
Store: 31

## Total Sales by Dept





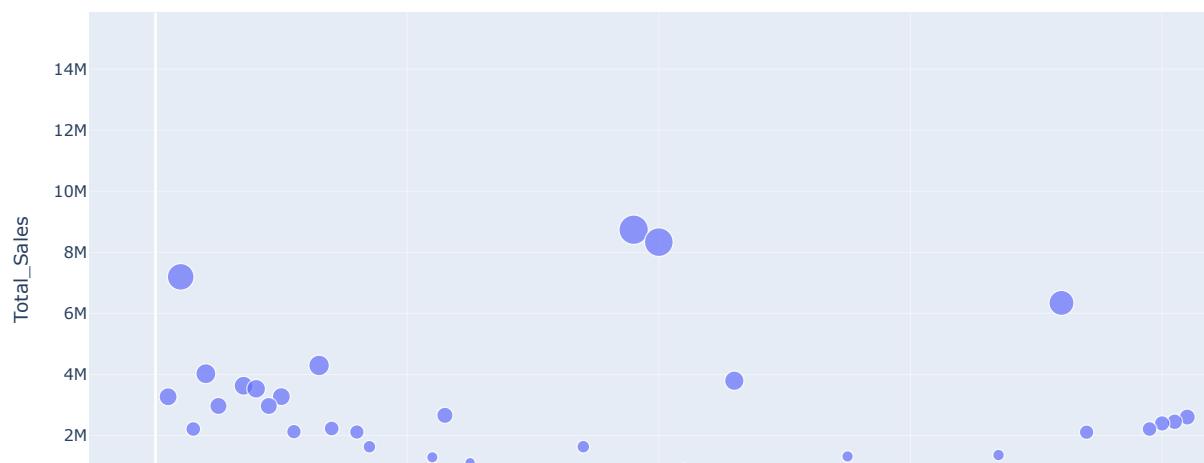
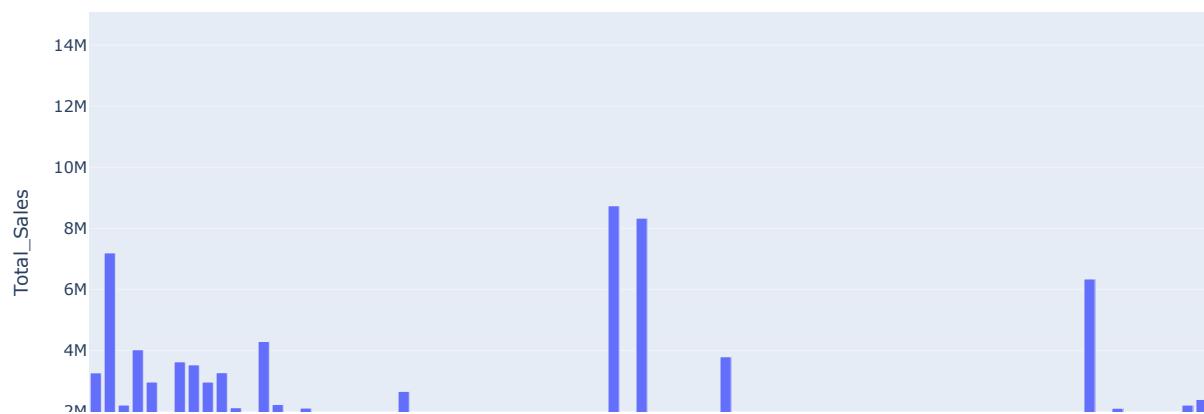
Total Sales by Dept



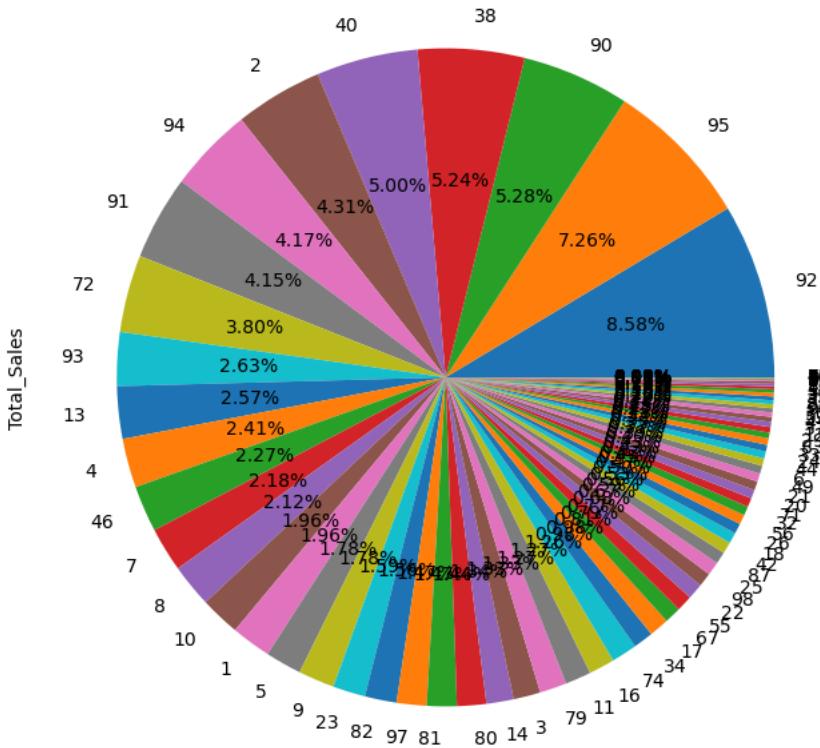
\*\*\*\*\*
\*\*\*\*\*  
\*\*\*\*\*

Store: 32

## Total Sales by Dept



## Total Sales by Dept



\*\*\*\*\*  
\*\*\*\*\*

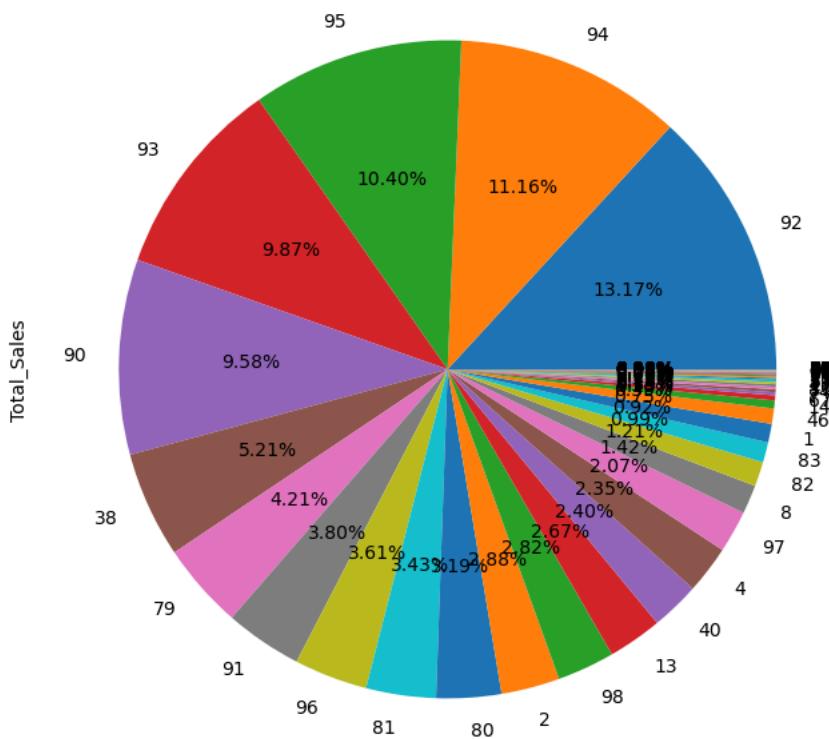
Store: 33

## Total Sales by Dept





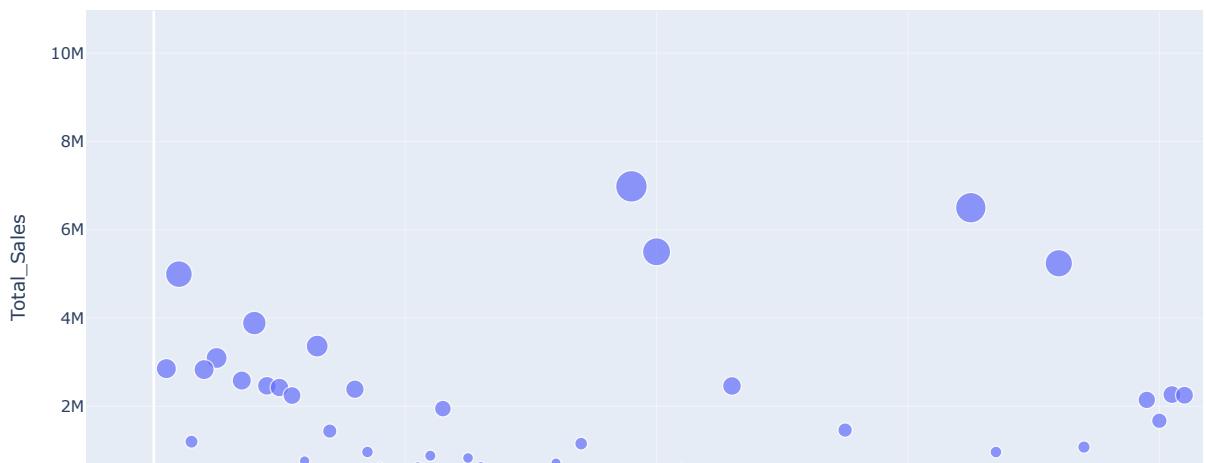
Total Sales by Dept



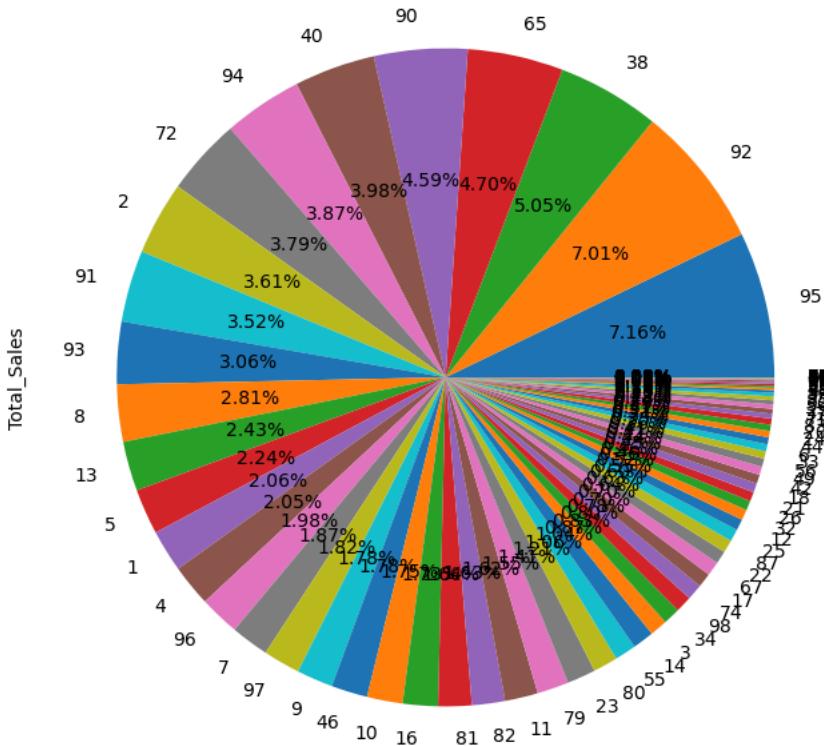
\*\*\*\*\*  
\*\*\*\*\*

Store: 34

## Total Sales by Dept



### Total Sales by Dept

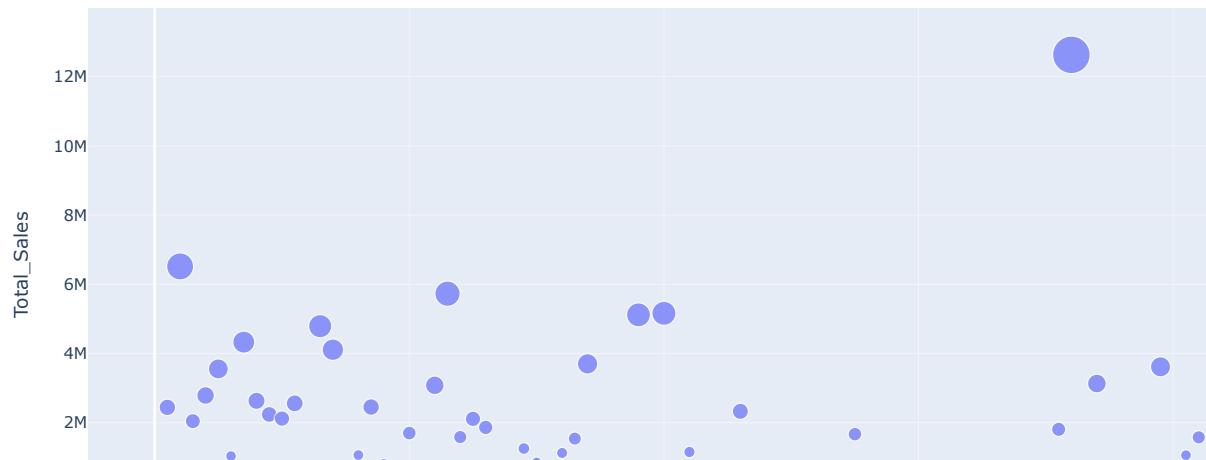


\*\*\*\*\*  
\*\*\*\*\*

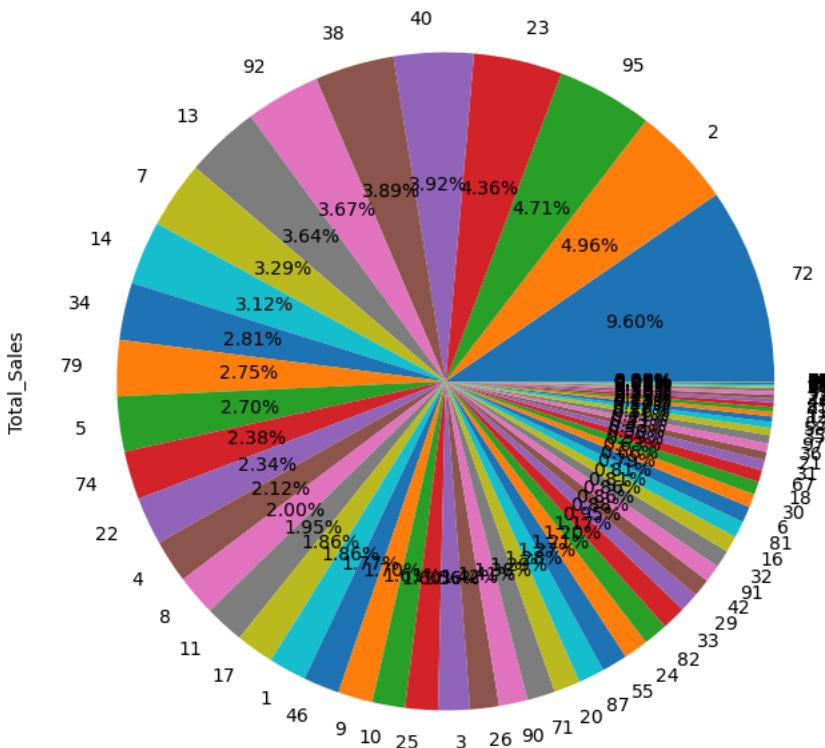
Store: 35

### Total Sales by Dept





Total Sales by Dept

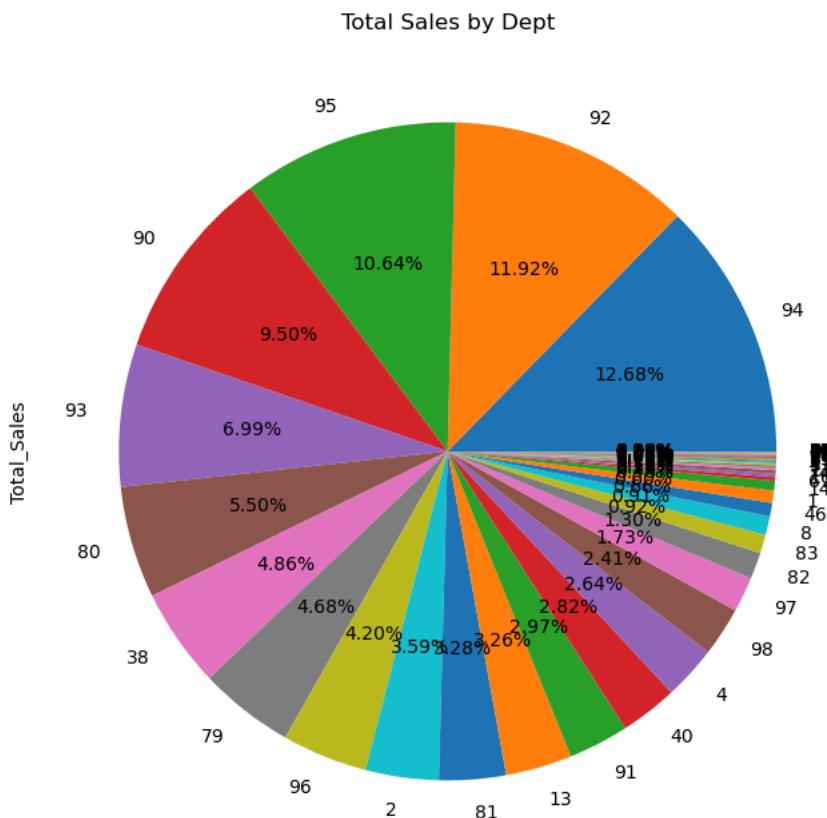


\*\*\*\*\*
\*\*\*\*\*  
\*\*\*\*\*

Store: 36

## Total Sales by Dept





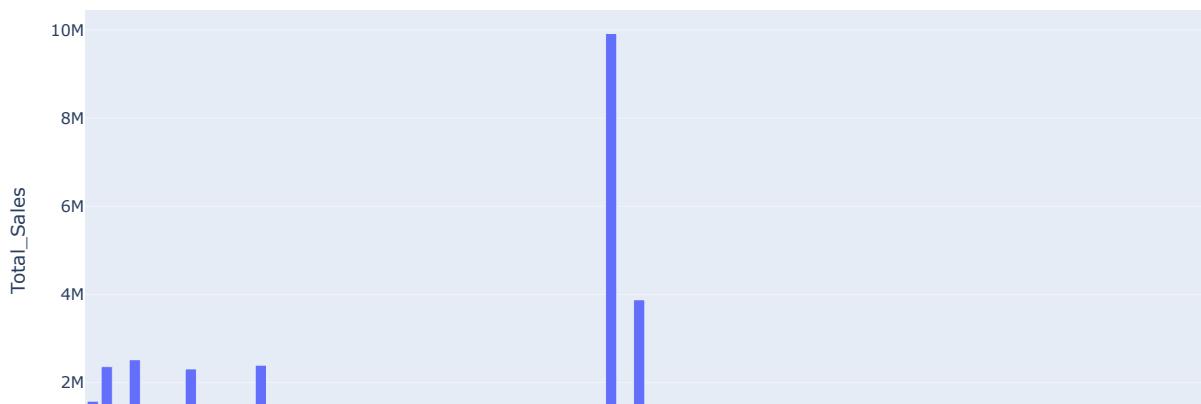

---

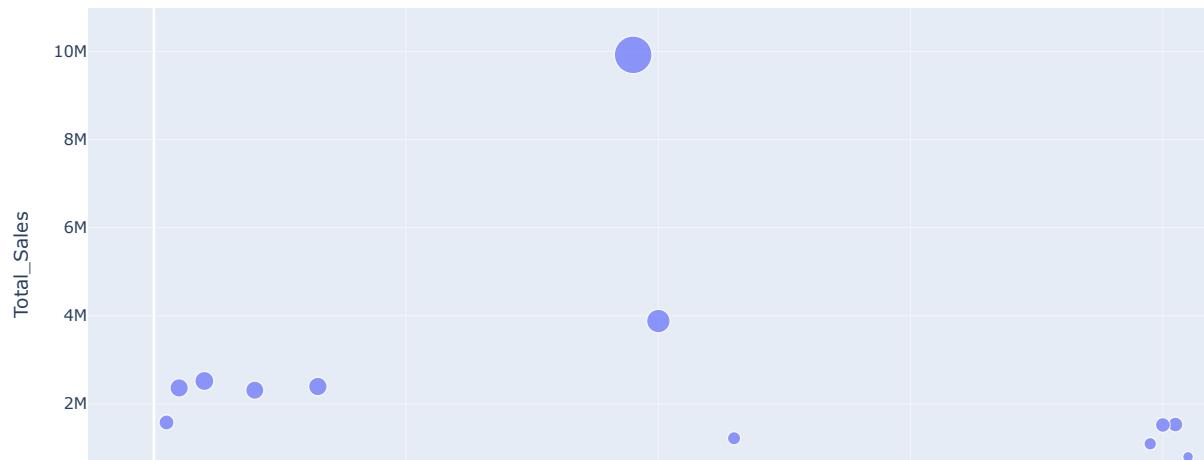


---

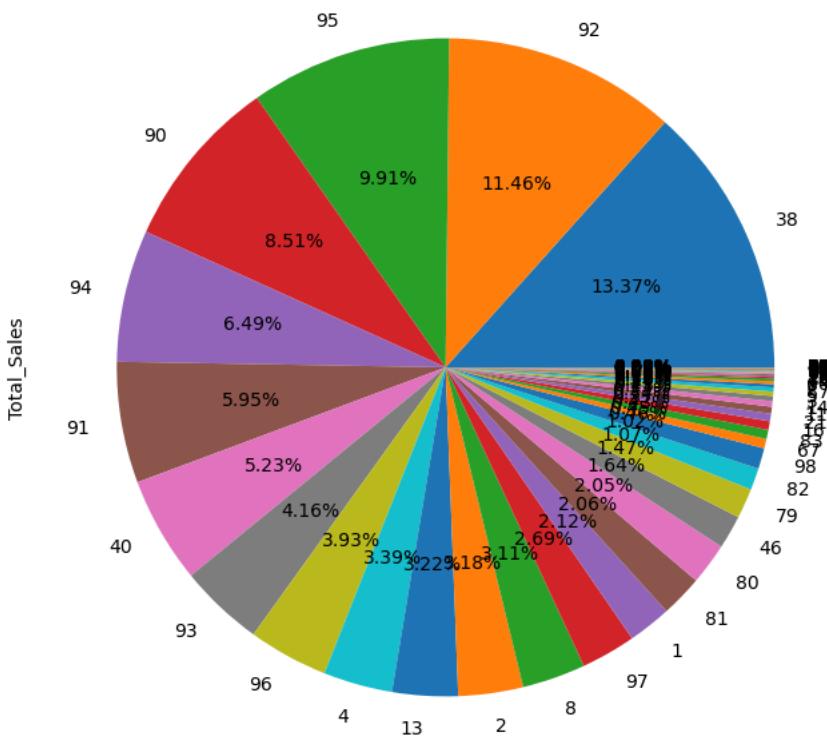
Store: 37

### Total Sales by Dept





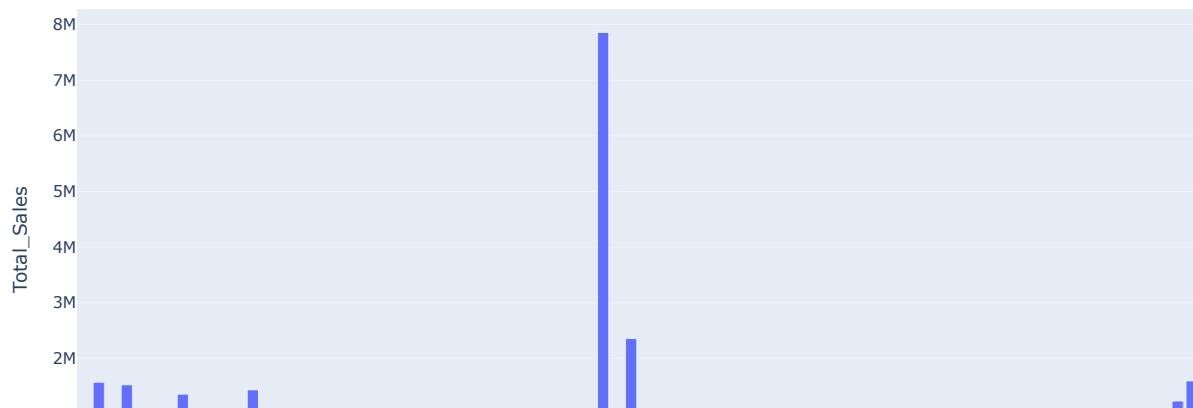
Total Sales by Dept



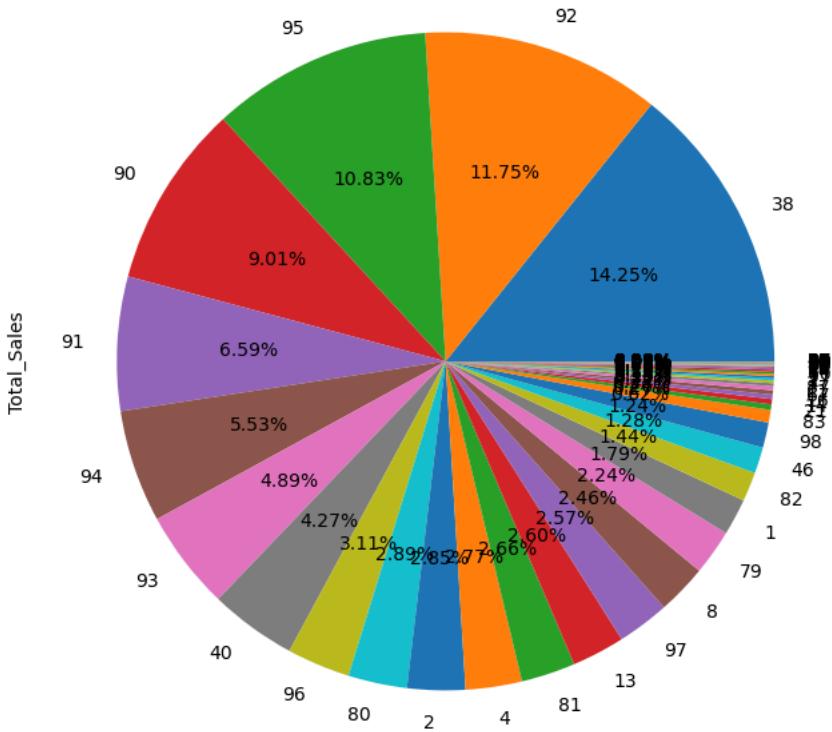
\*\*\*\*\*
\*\*\*\*\*  
\*\*\*\*\*

Store: 38

## Total Sales by Dept



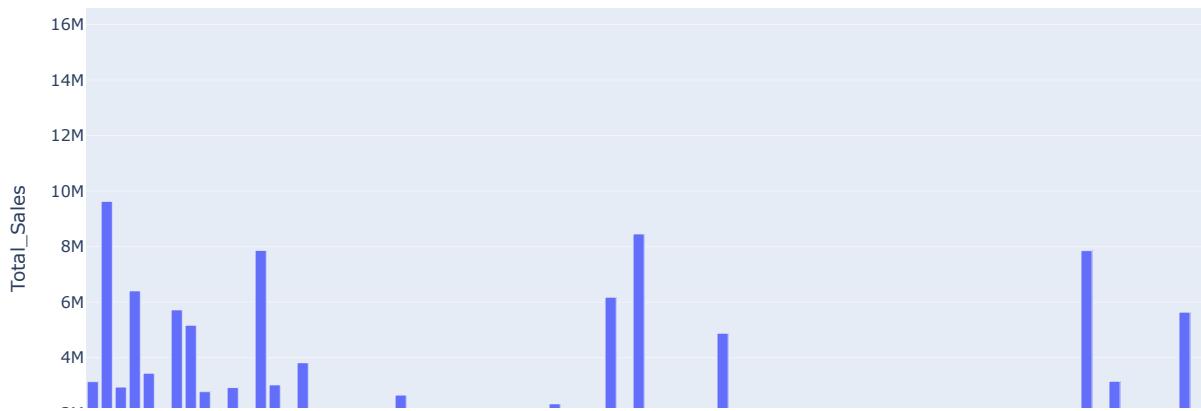
## Total Sales by Dept

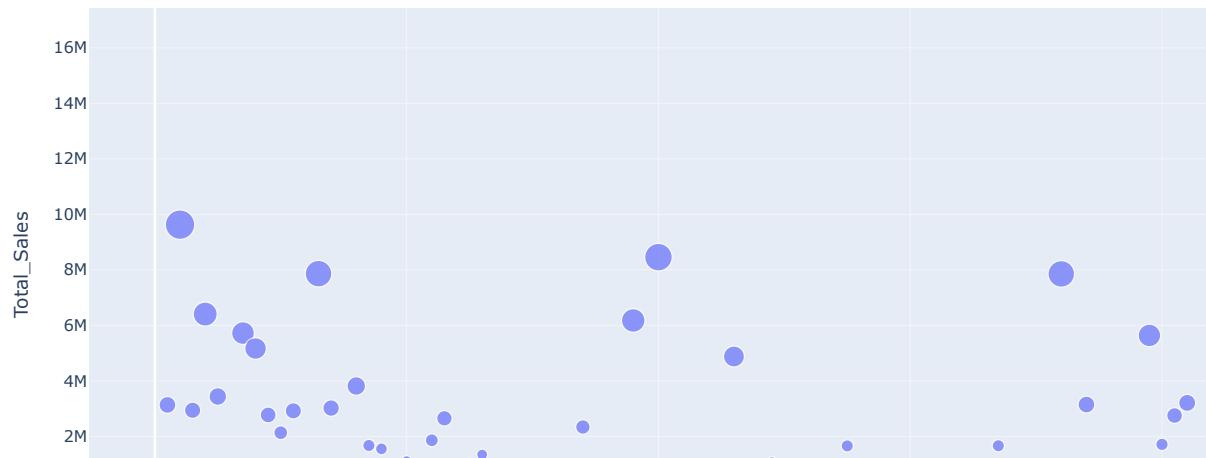


\*\*\*\*\*
\*\*\*\*\*

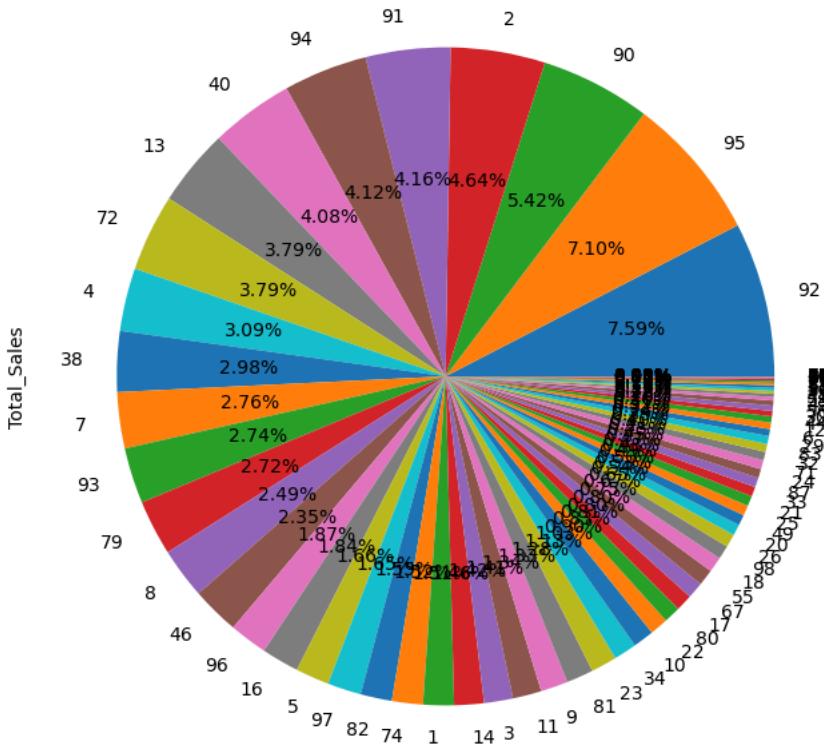
Store: 39

## Total Sales by Dept





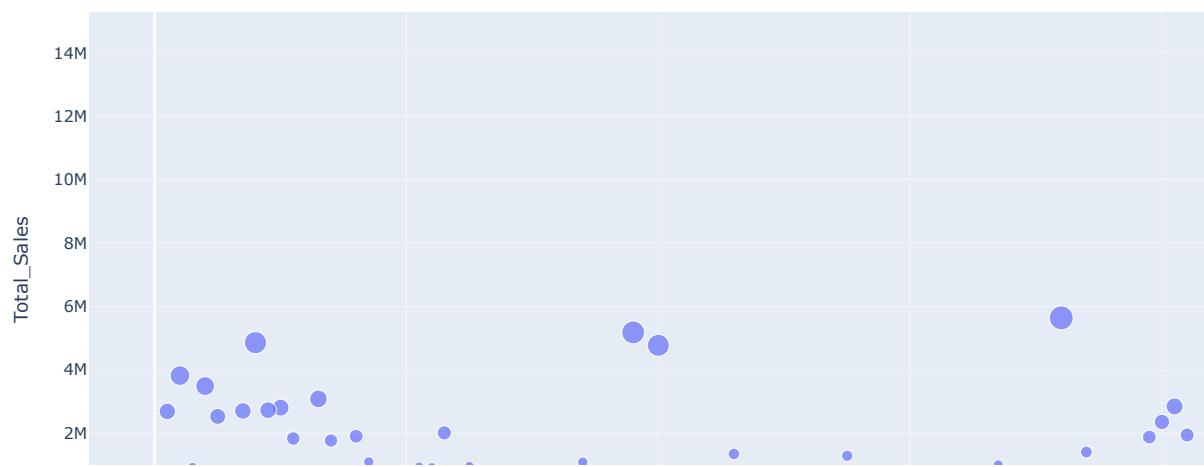
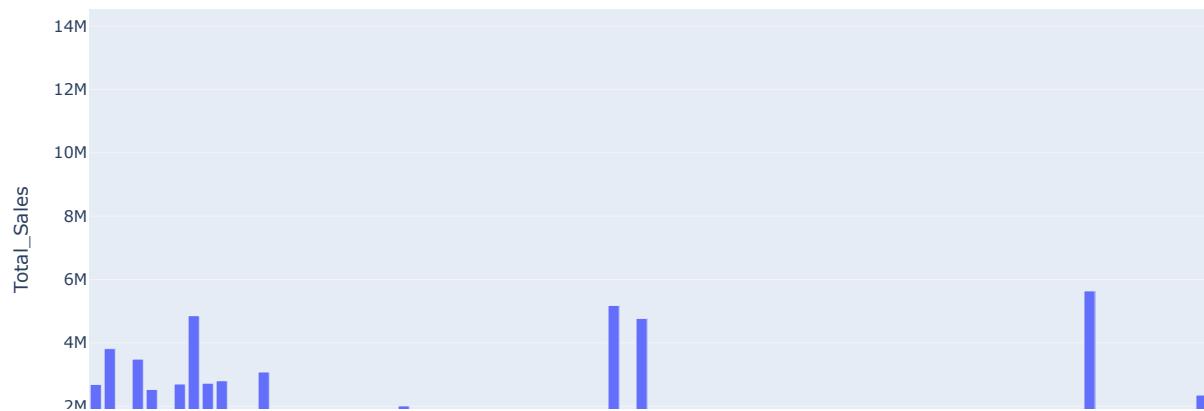
Total Sales by Dept



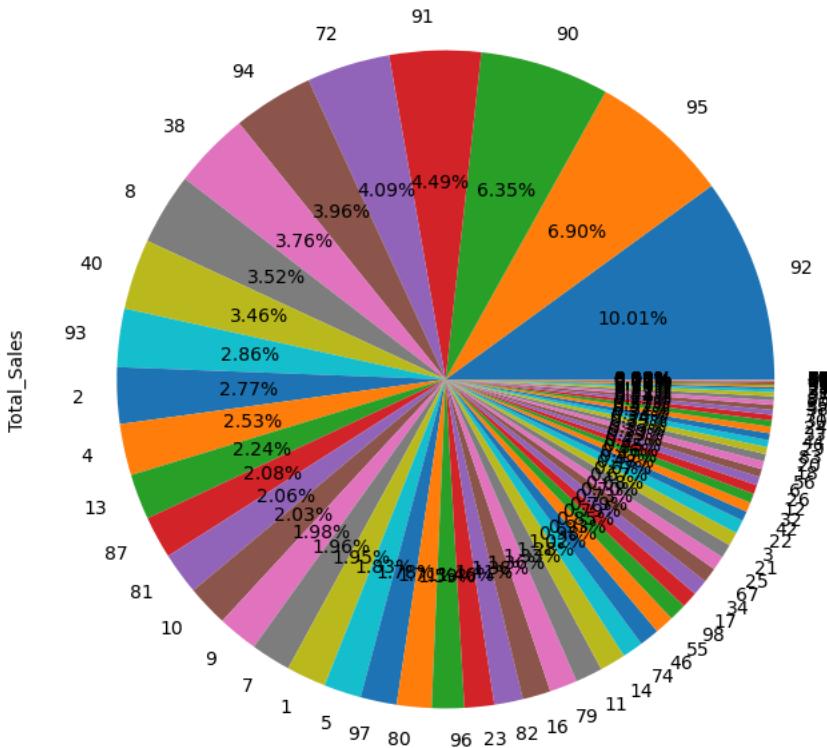
\*\*\*\*\*
\*\*\*\*\*  
\*\*\*\*\*

Store: 40

## Total Sales by Dept



## Total Sales by Dept

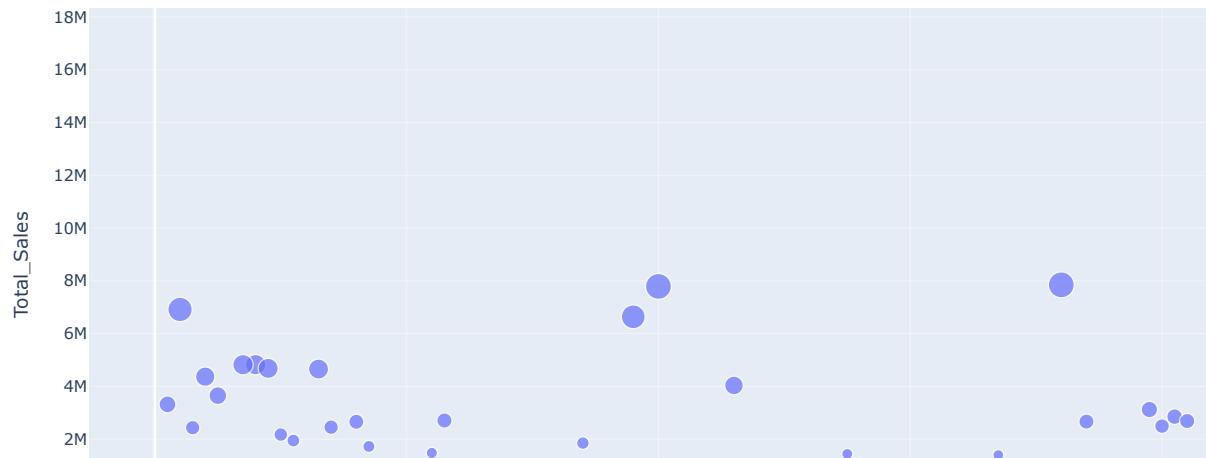


\*\*\*\*\*
\*\*\*\*\*  
\*\*\*\*\*

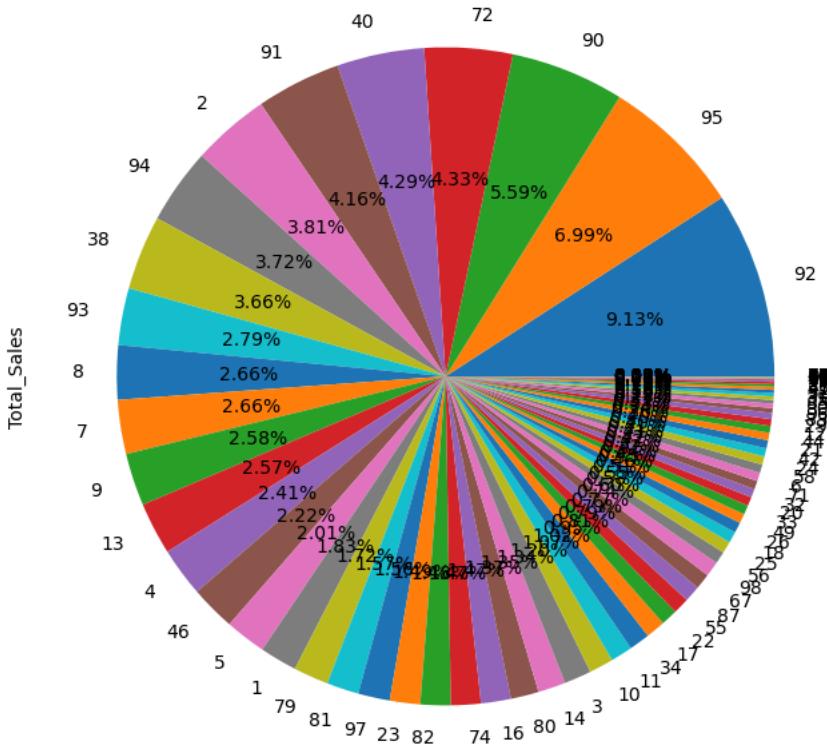
Store: 41

## Total Sales by Dept





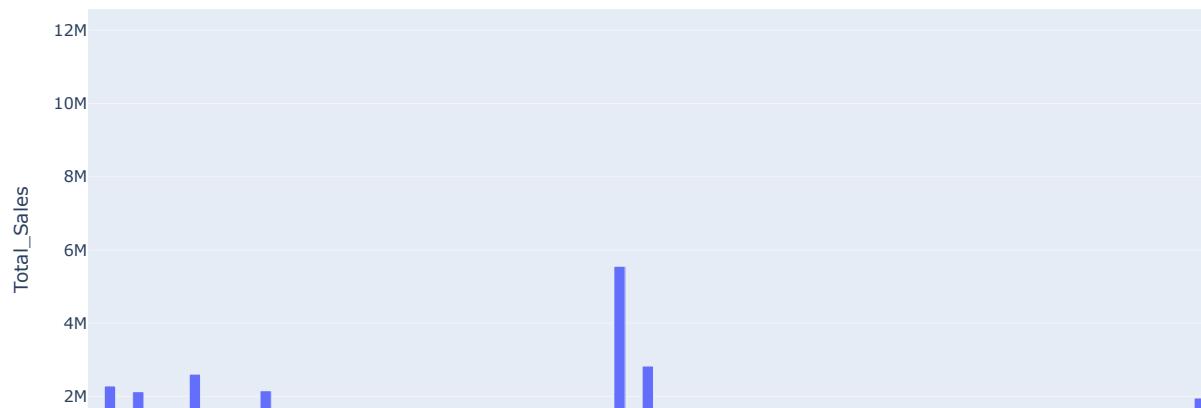
Total Sales by Dept



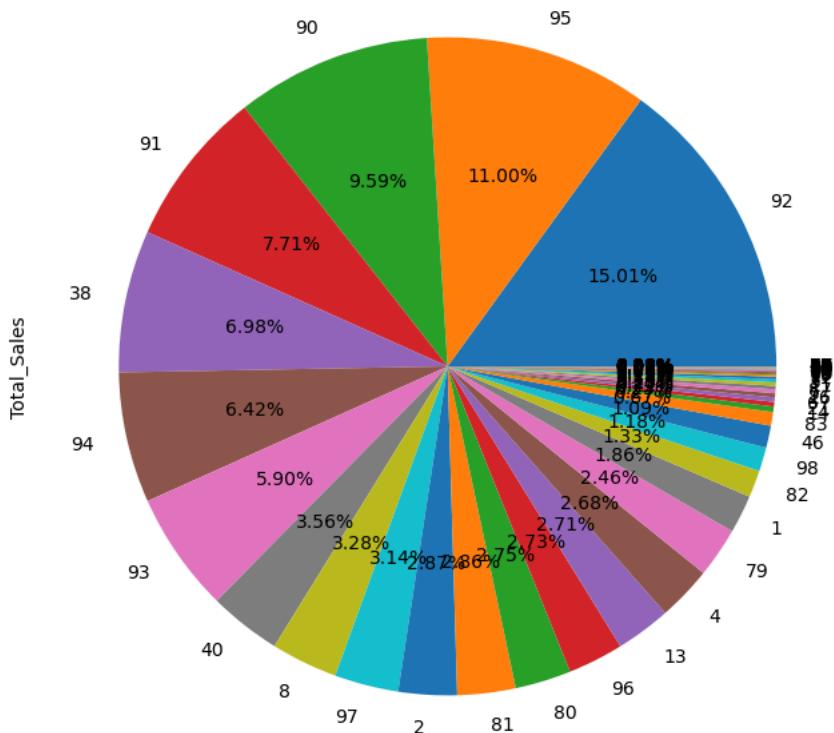
```
*****
*****
```

Store: 42

## Total Sales by Dept



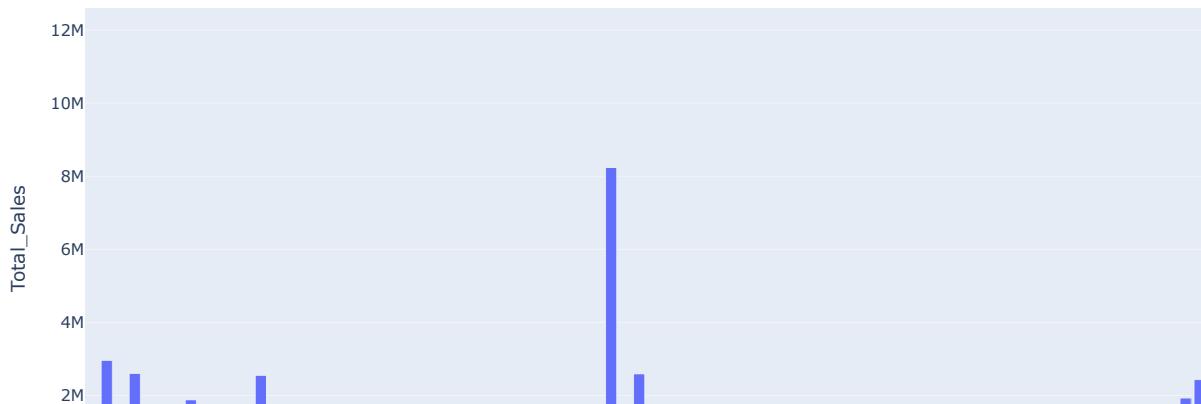
## Total Sales by Dept



\*\*\*\*\*  
\*\*\*\*\*

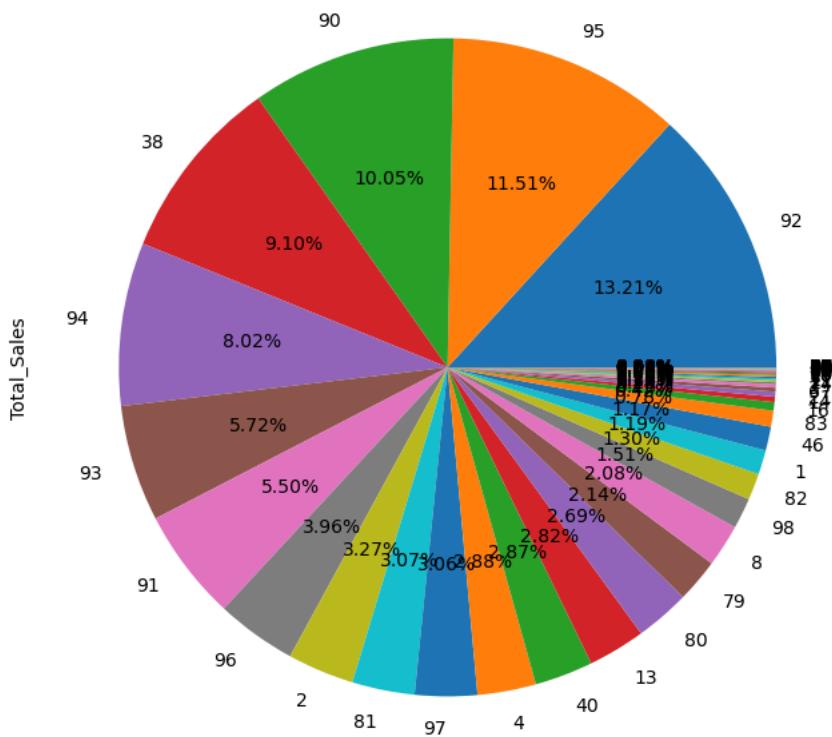
Store: 43

## Total Sales by Dept





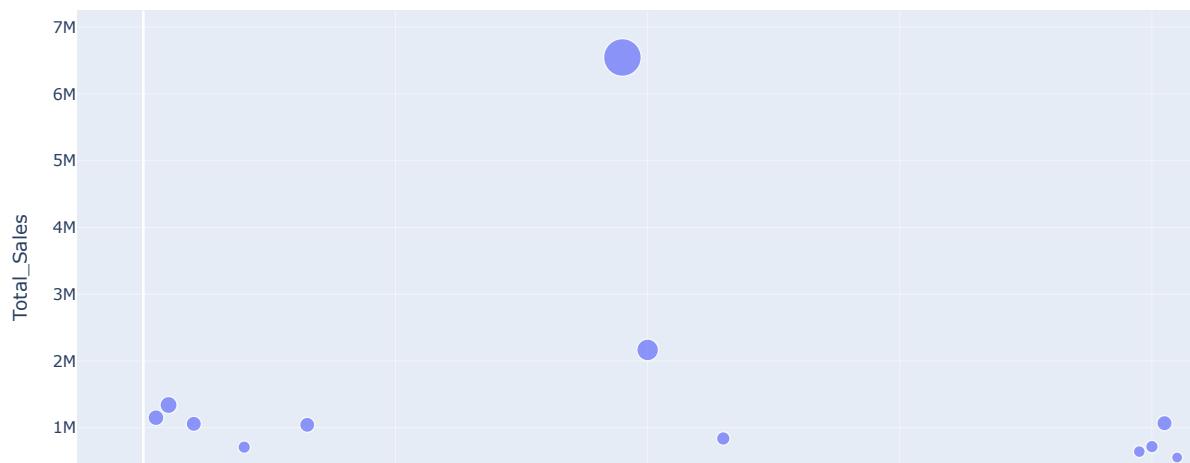
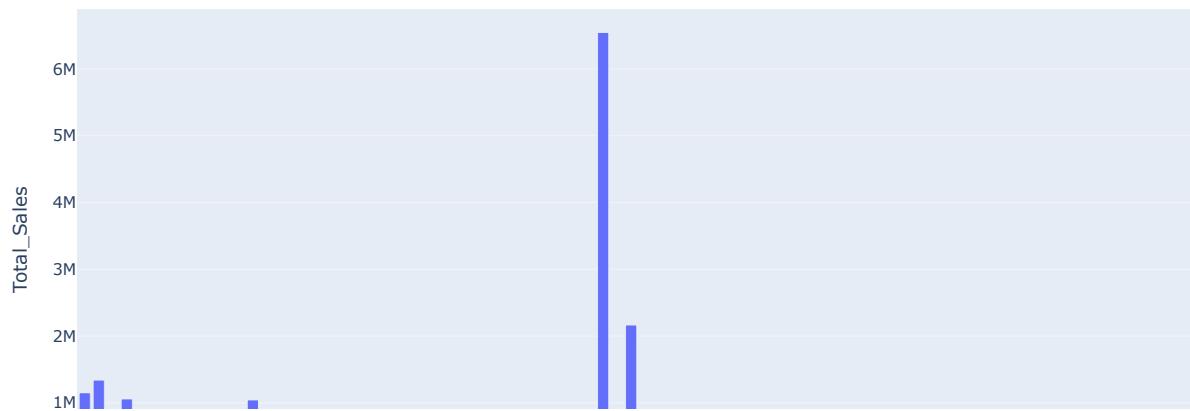
Total Sales by Dept



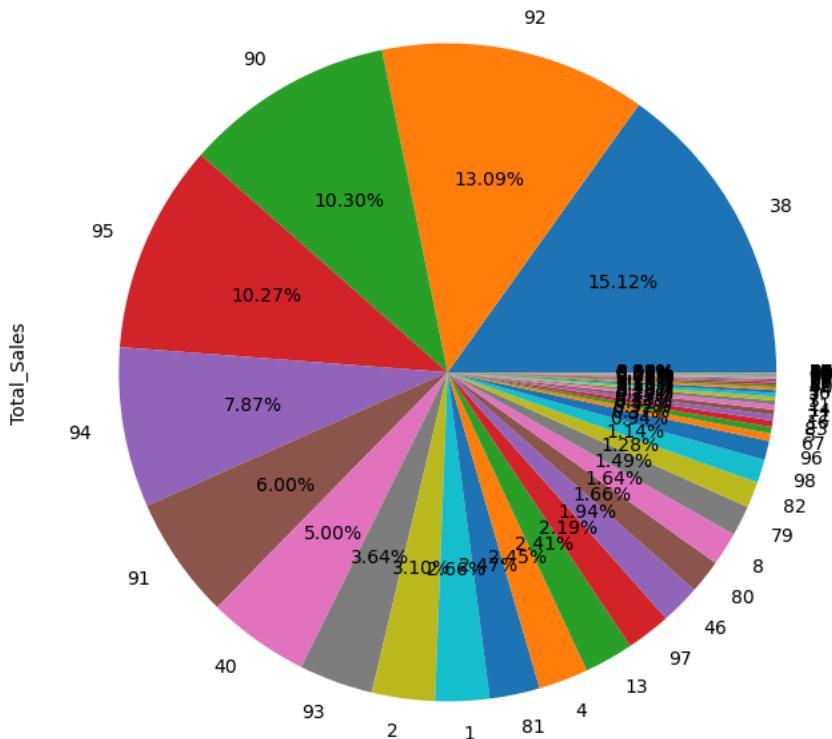
\*\*\*\*\*
\*\*\*\*\*  
\*\*\*\*\*

Store: 44

## Total Sales by Dept



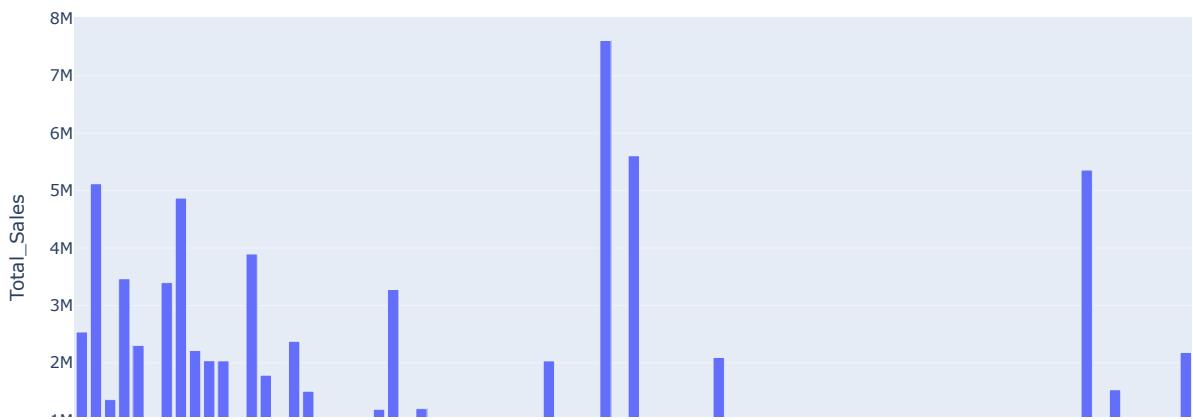
## Total Sales by Dept

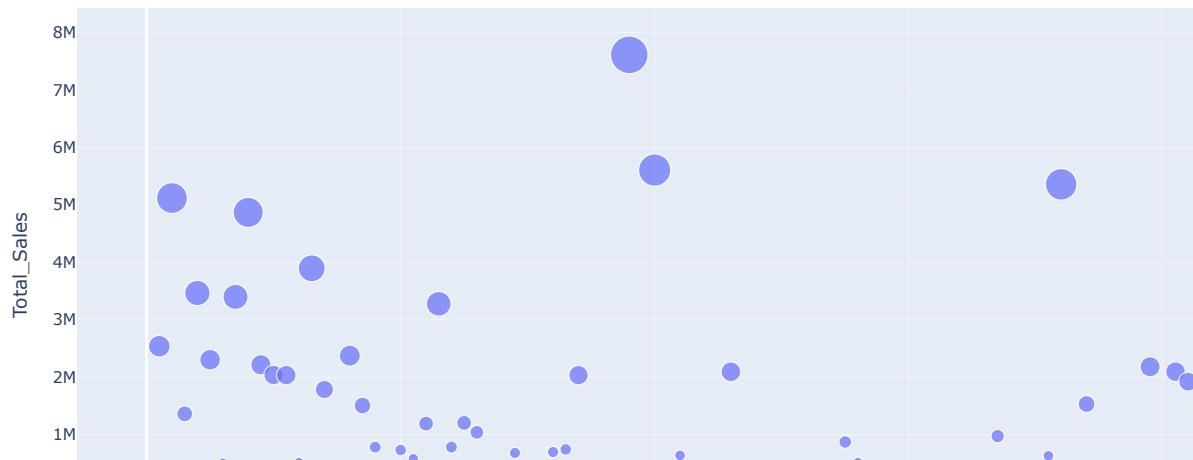


\*\*\*\*\*
\*\*\*\*\*

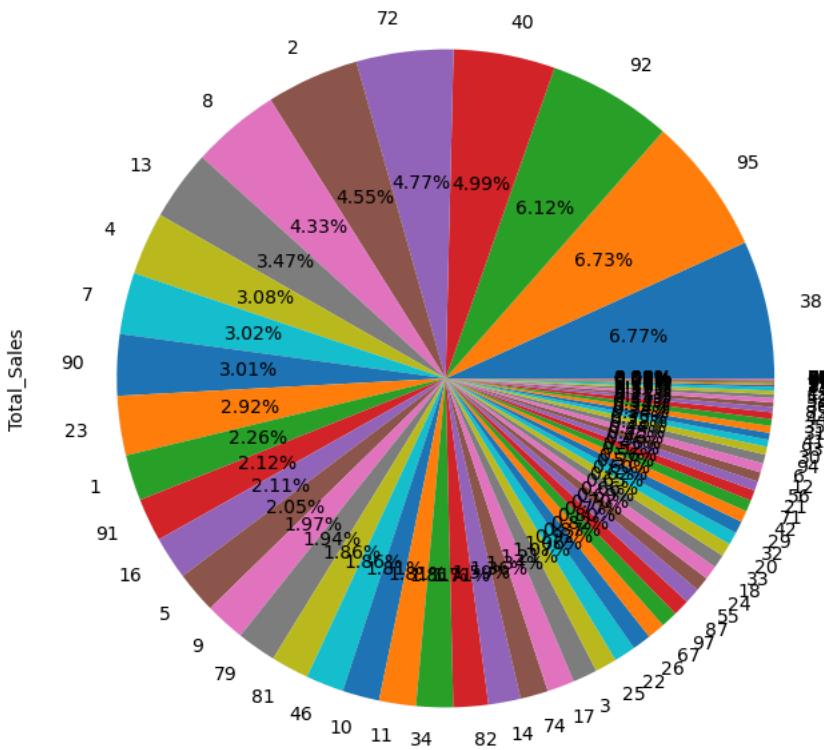
Store: 45

## Total Sales by Dept





Total Sales by Dept



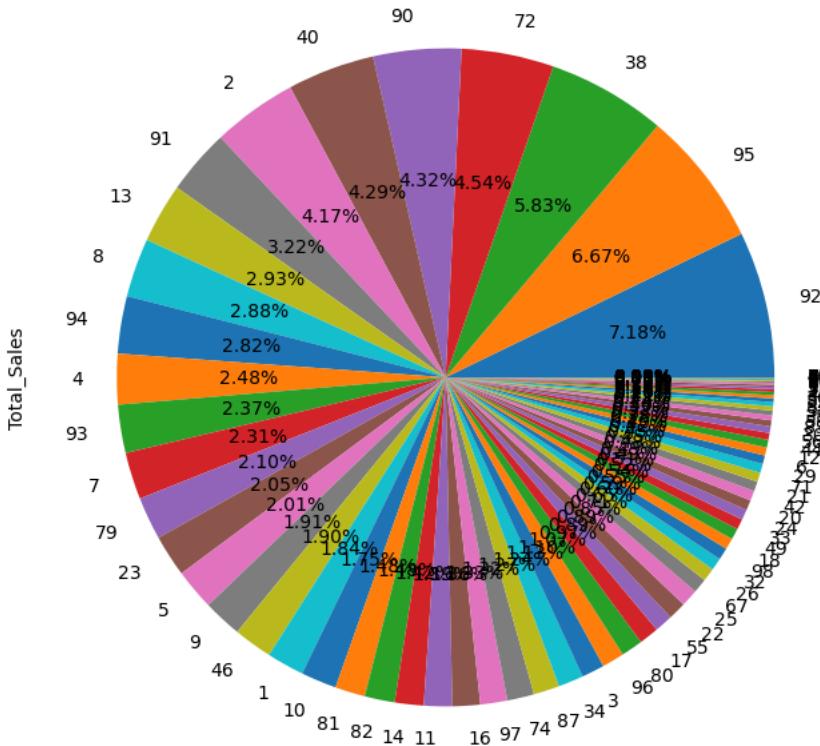
```
*****
*****
```

```
In [24]: # @title $Department \ \ wise \ \ Total \ \ Sales$  
get_sales_plot(sales,'Dept')  
get_pie_plot(sales,'Dept')
```

## Total Sales by Dept



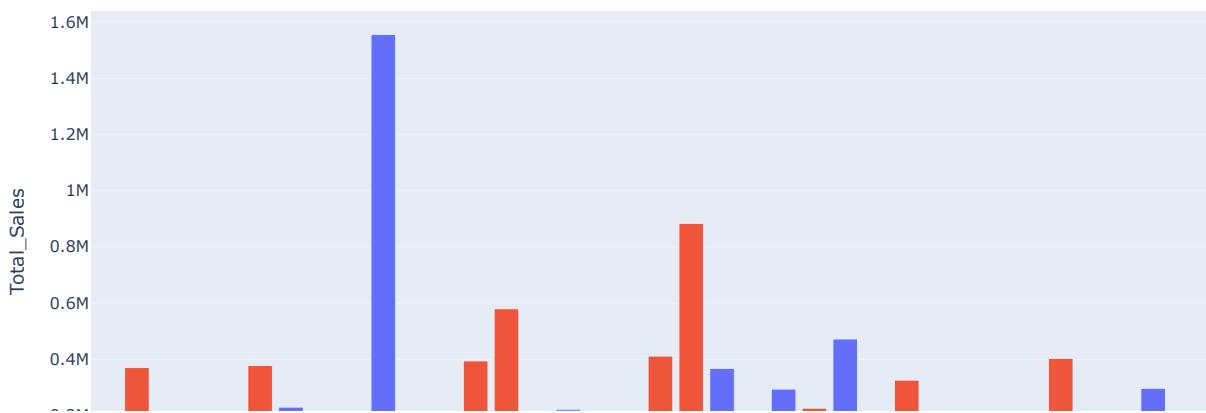
## Total Sales by Dept

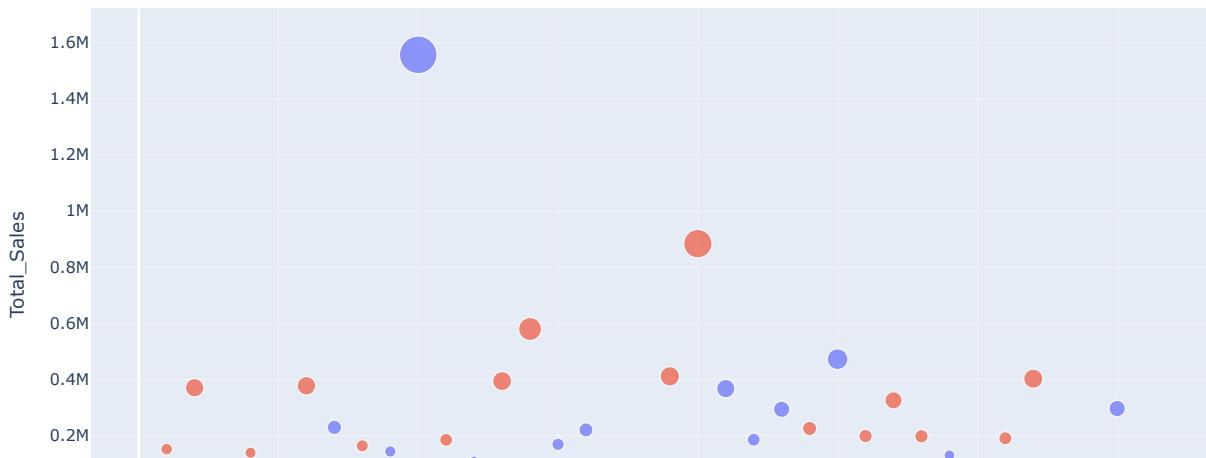


```
In [25]: # @title $Store \ \ wise \ \ sales \ \ of \ \ Department \ \ n$ {run: "auto", "form-width": "20%"}
dept_no = 41 # @param {"type": "slider", "min": 1, "max": 99, "step": 1}

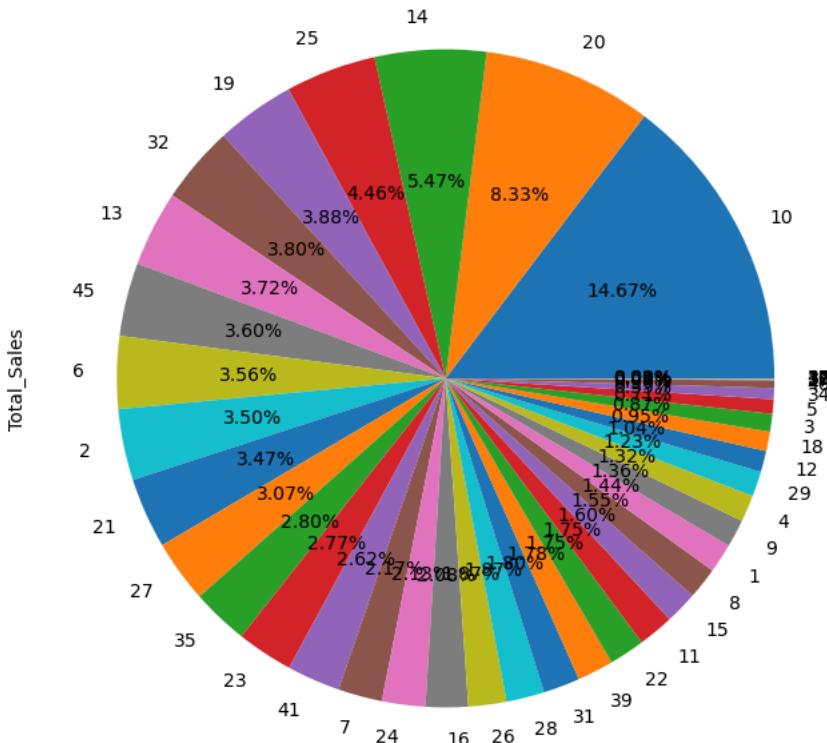
if(dept_no in list_departments):
    df = sales[sales['Dept']==dept_no]
    get_sales_plot(df,'Store')
    get_pie_plot(df, 'Store')
else:
    print(f'Department {dept_no} is not available in the dataset')
```

## Total Sales by Store





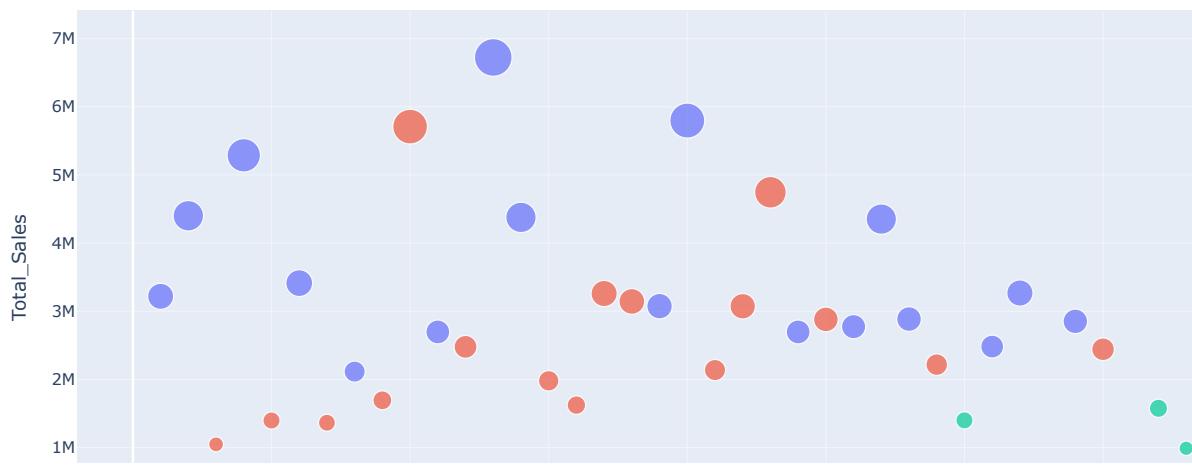
Total Sales by Store



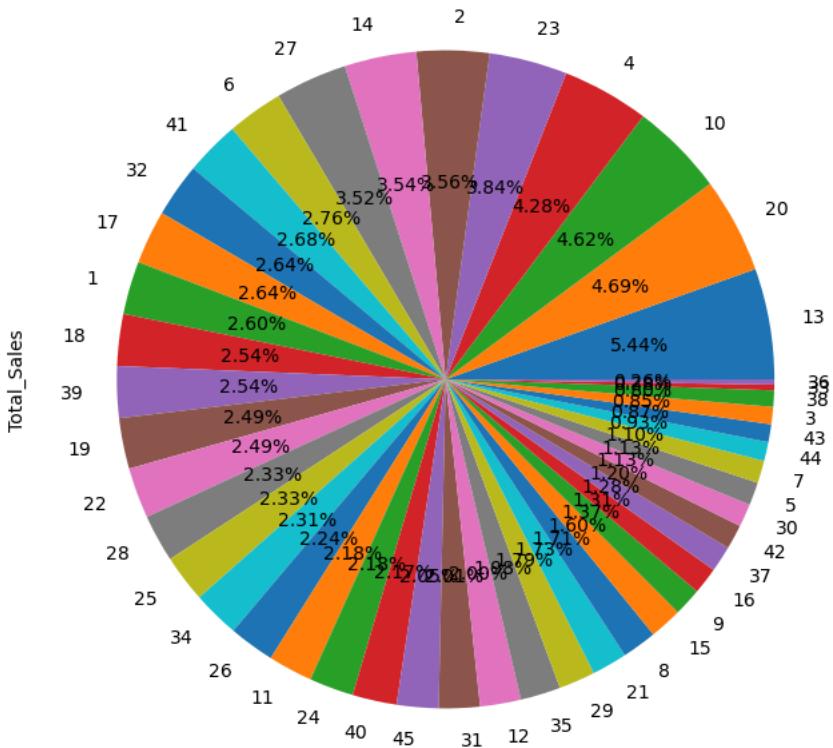
```
In [26]: # @title $Store \ \ wise \ \ Sales \ \ of \ \ ALL \ \ Departments$  
for dept_no in list_departments:  
    print('Dept:',dept_no)  
    df = sales[sales['Dept']==dept_no]  
    get_sales_plot(df,'Store')  
    get_pie_plot(df, 'Store')  
    print()  
    print('*'*150)  
    print()
```

Dept: 1

## Total Sales by Store



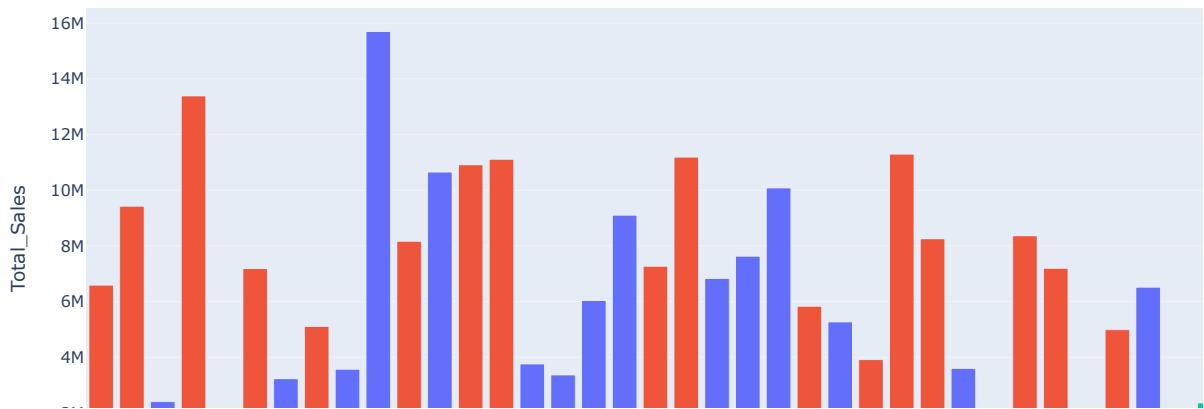
## Total Sales by Store

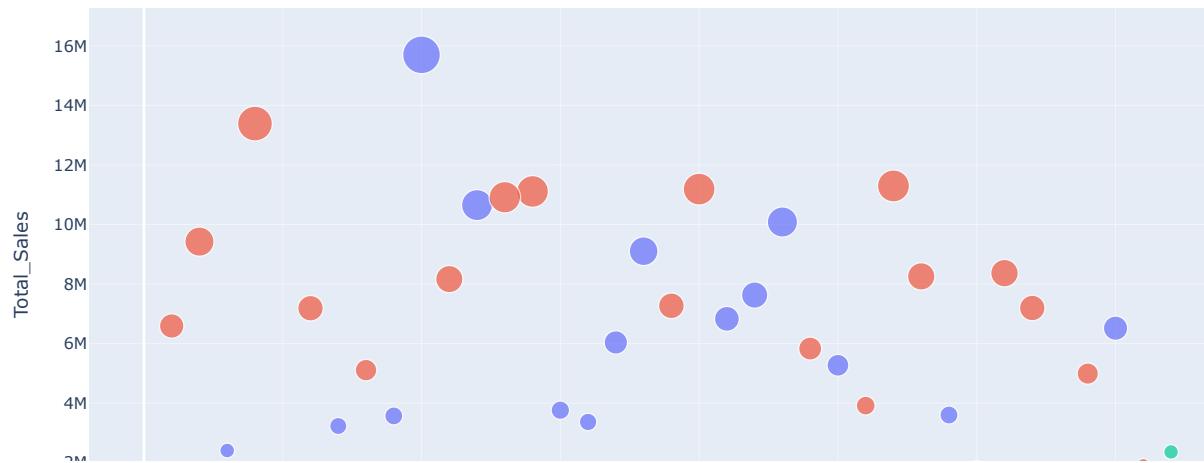


\*\*\*\*\*
\*\*\*\*\*

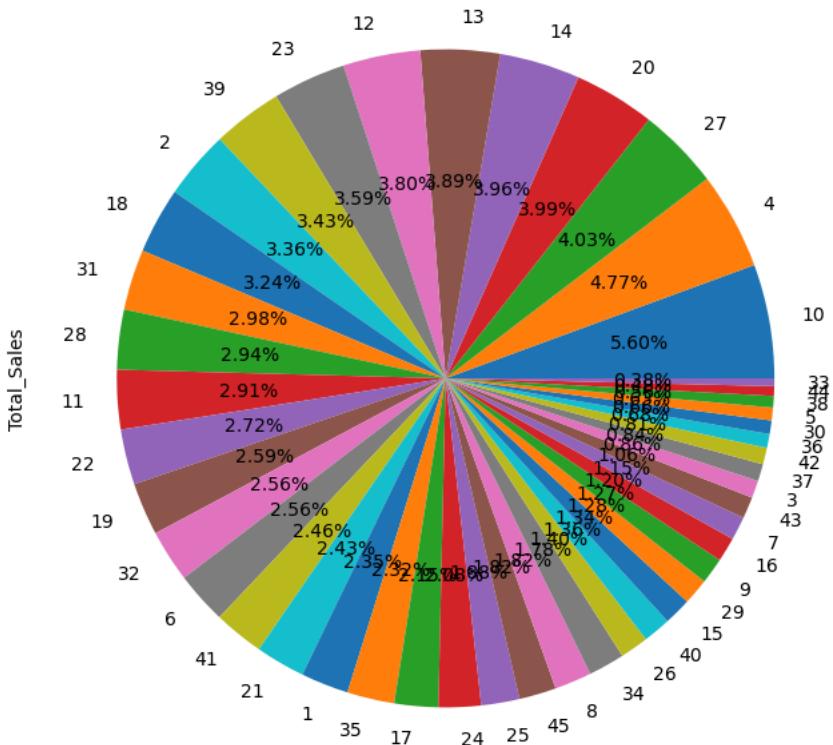
Dept: 2

## Total Sales by Store





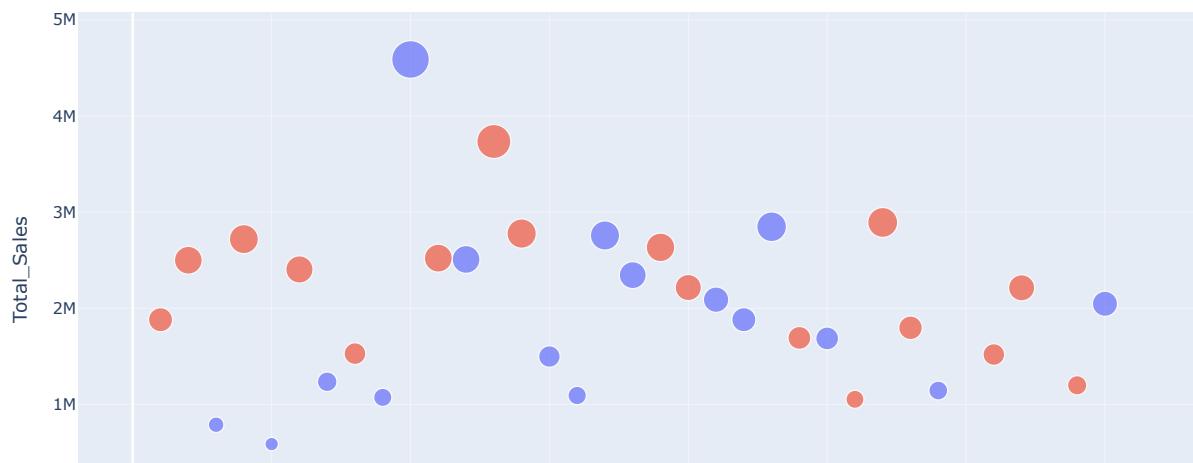
Total Sales by Store



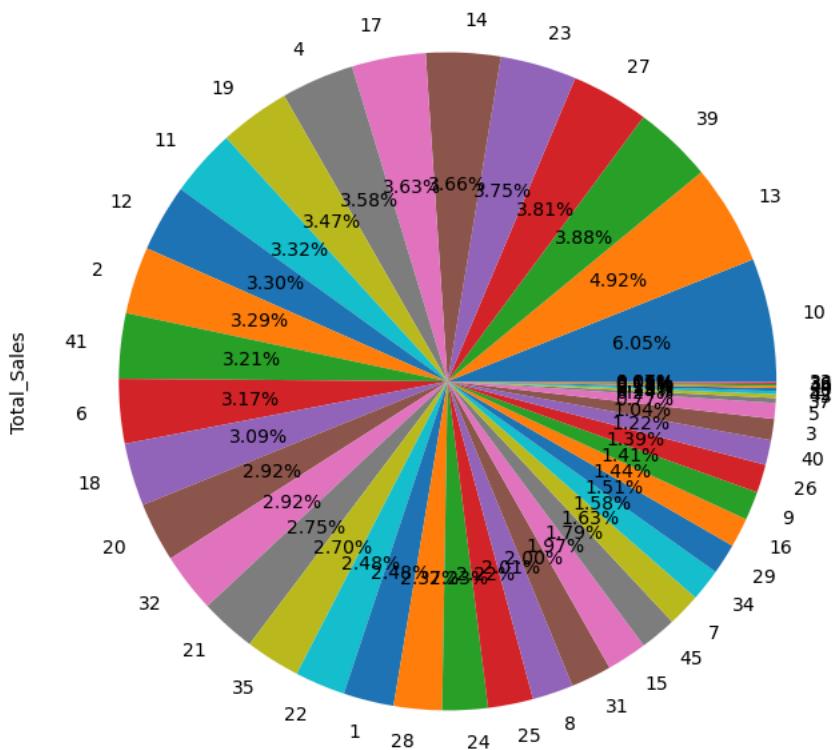
\*\*\*\*\*
\*\*\*\*\*  
\*\*\*\*\*

Dept: 3

## Total Sales by Store



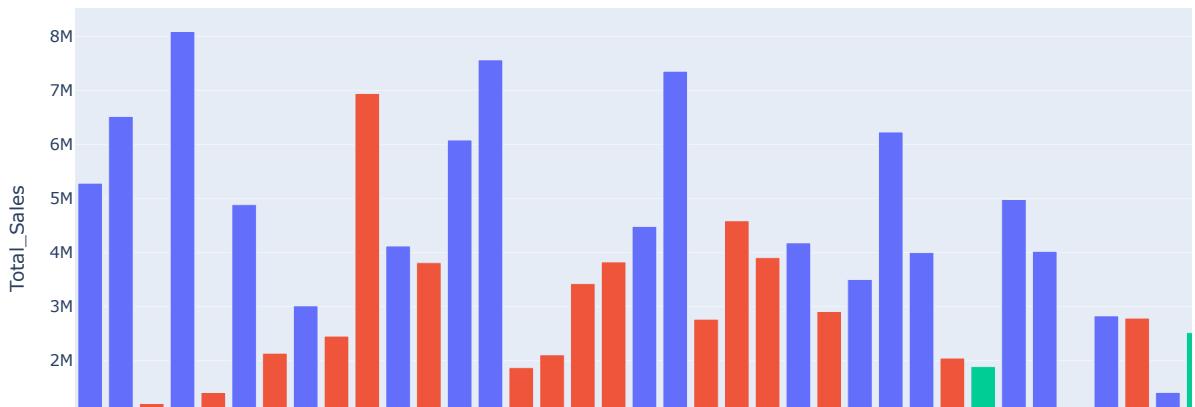
## Total Sales by Store



\*\*\*\*\*
\*\*\*\*\*

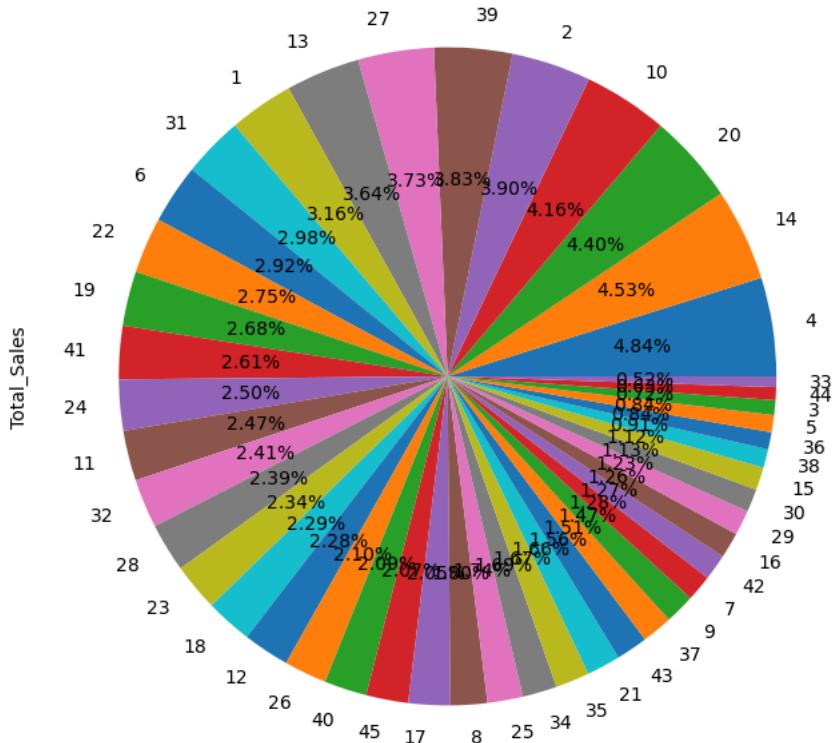
Dept: 4

## Total Sales by Store





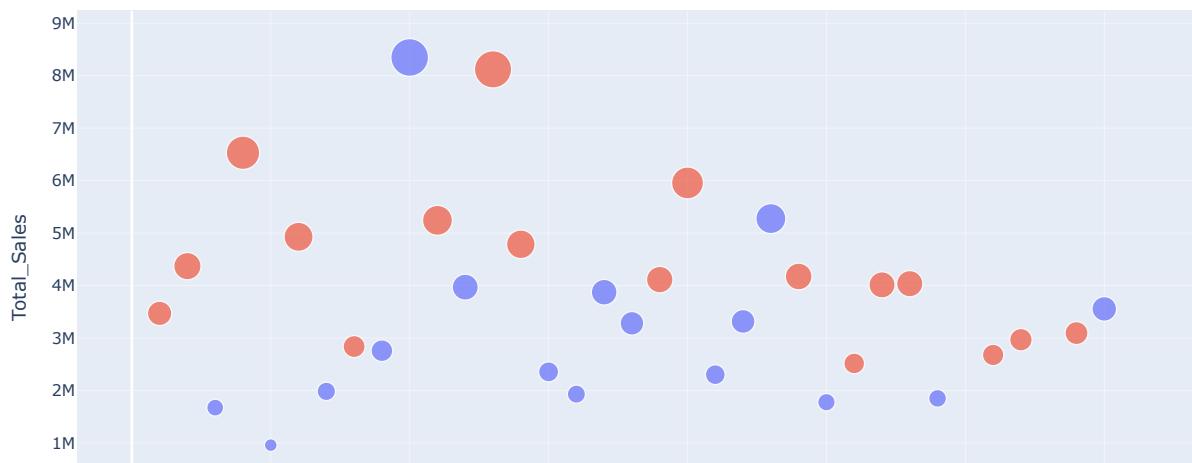
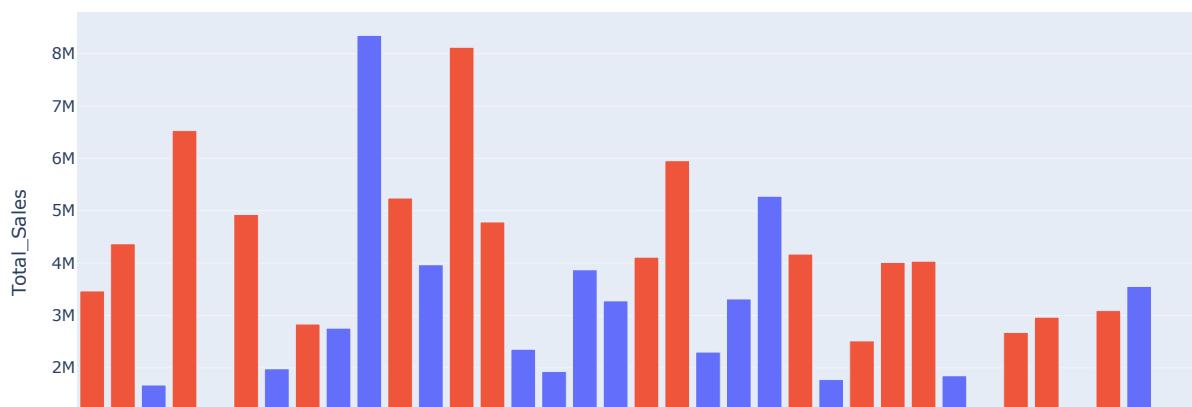
Total Sales by Store



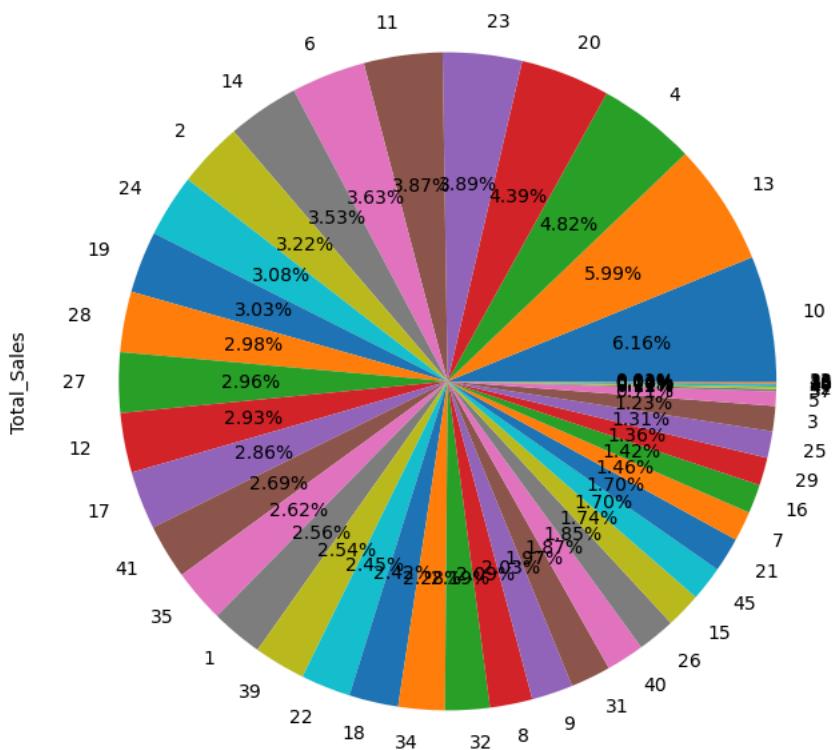
\*\*\*\*\*
\*\*\*\*\*  
\*\*\*\*\*

Dept: 5

## Total Sales by Store



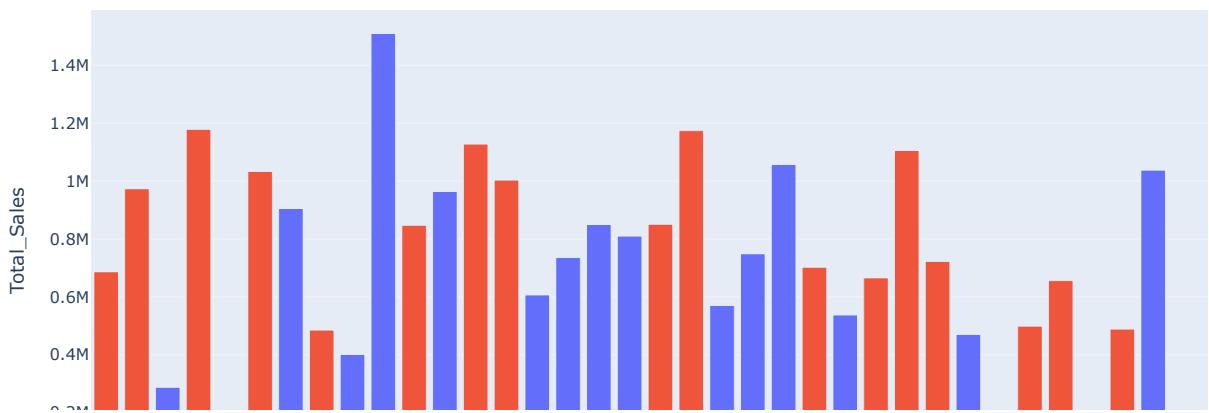
## Total Sales by Store



\*\*\*\*\*
\*\*\*\*\*

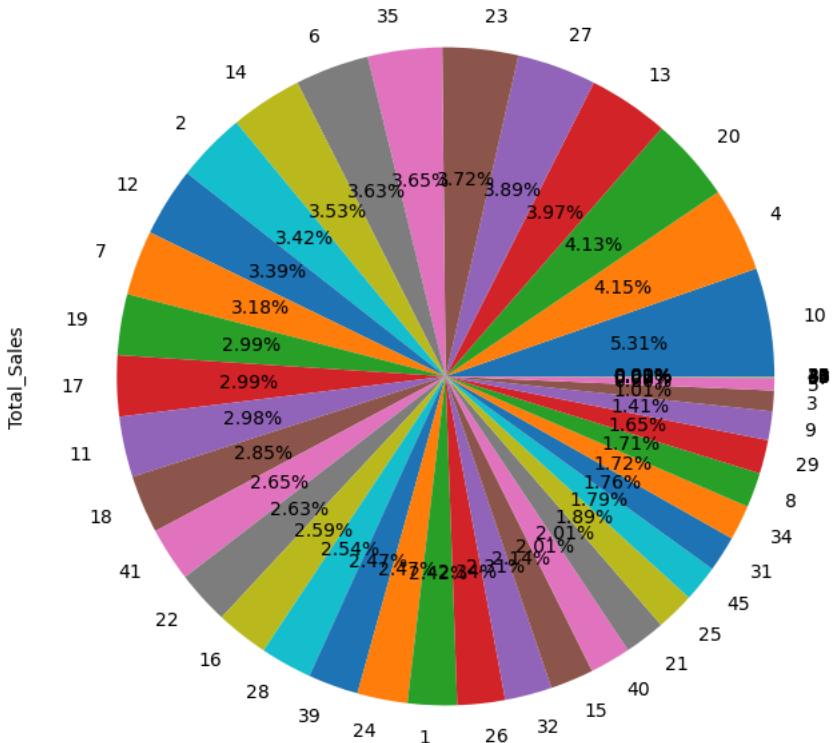
Dept: 6

## Total Sales by Store





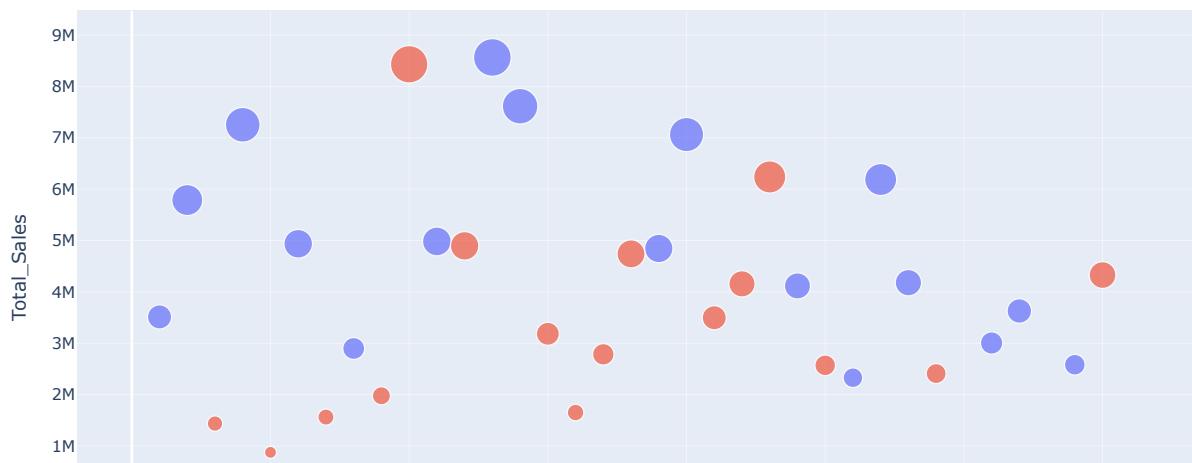
Total Sales by Store

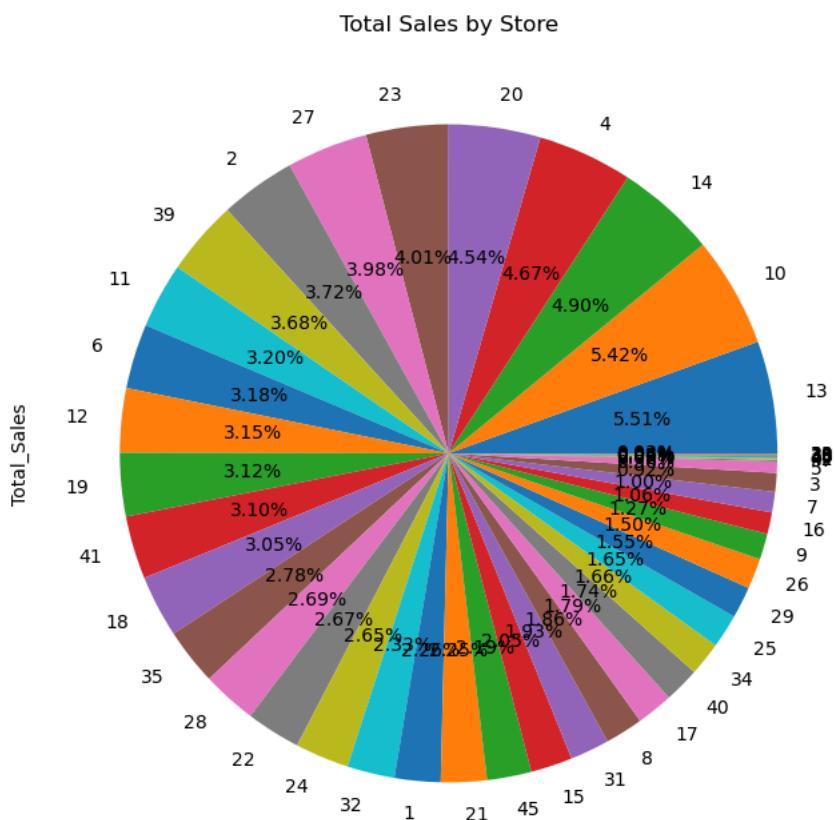


```
*****
*****
```

Dept: 7

## Total Sales by Store





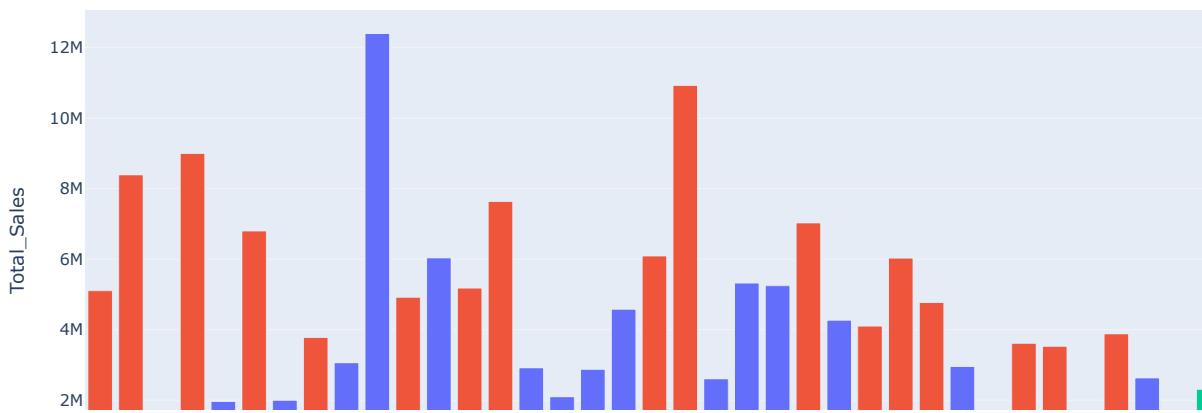

---

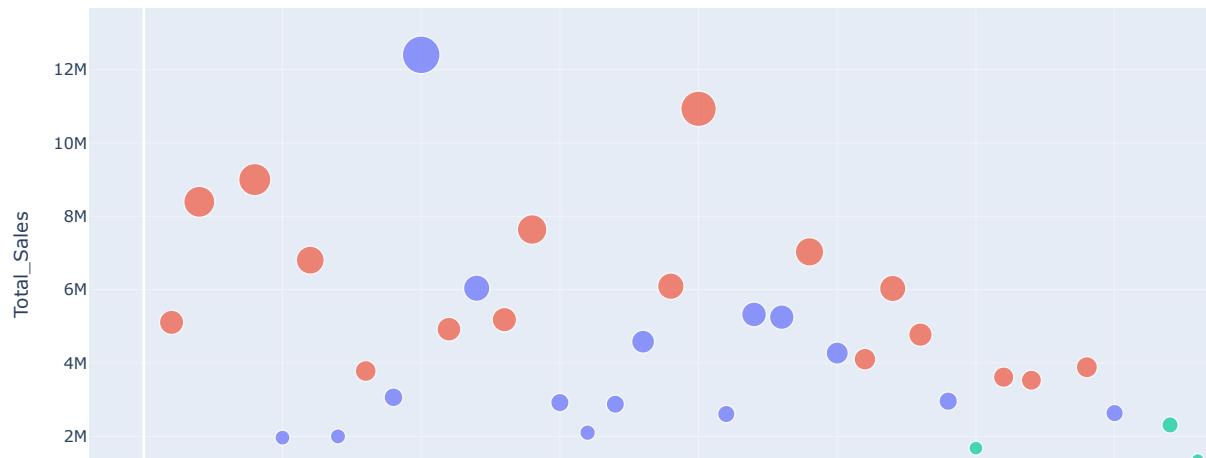


---

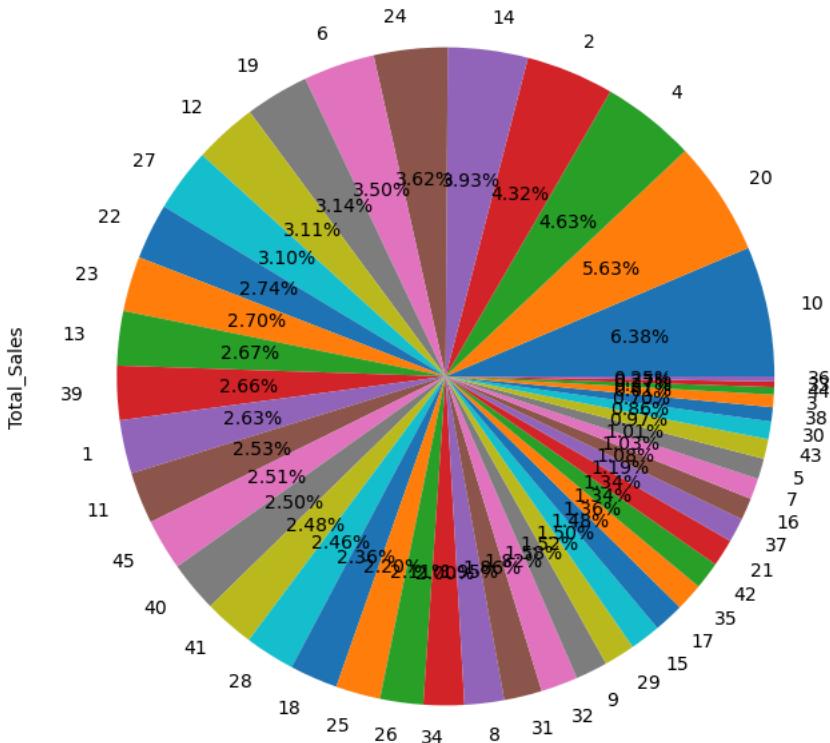
Dept: 8

### Total Sales by Store





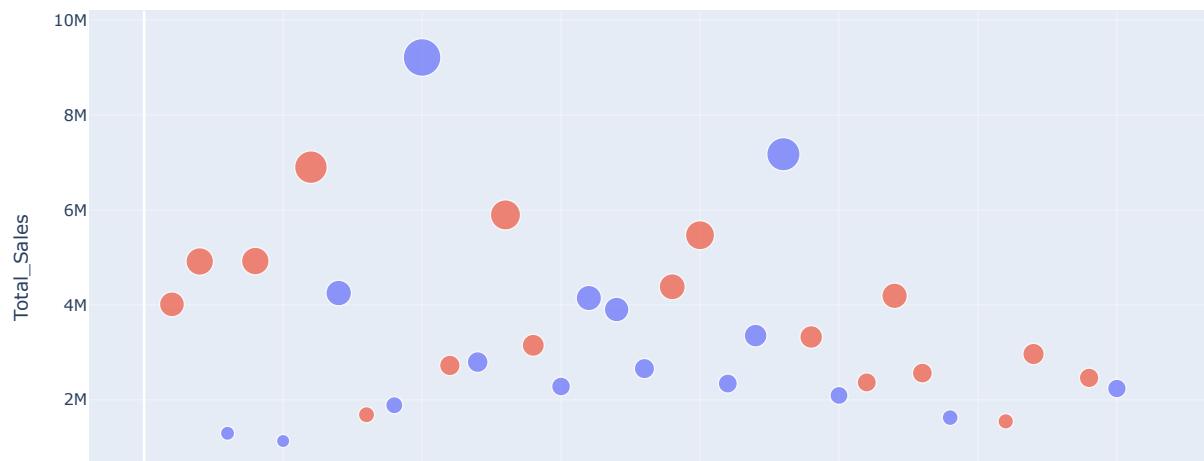
Total Sales by Store



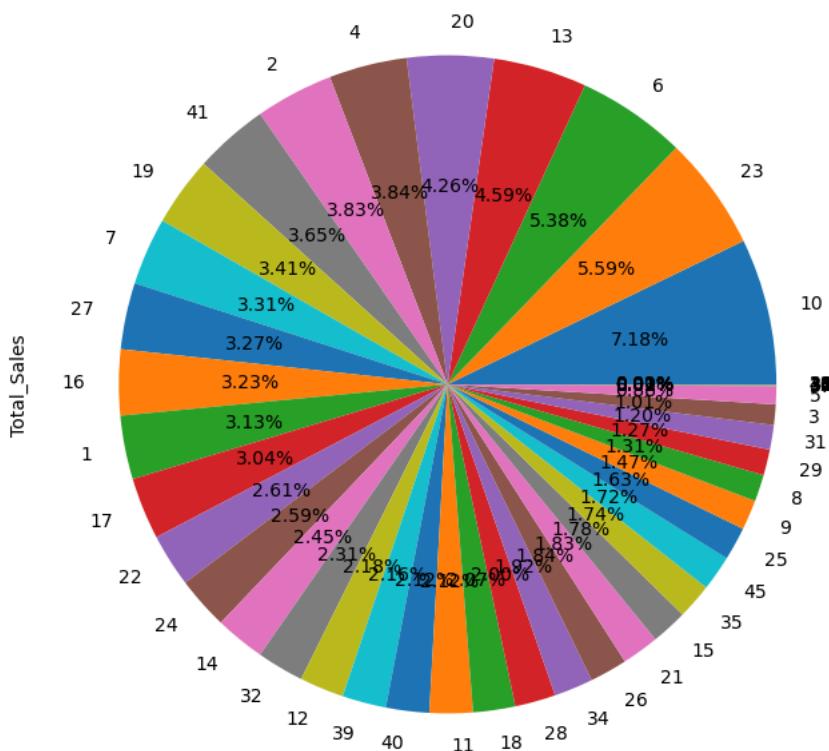
```
*****
*****
```

Dept: 9

## Total Sales by Store



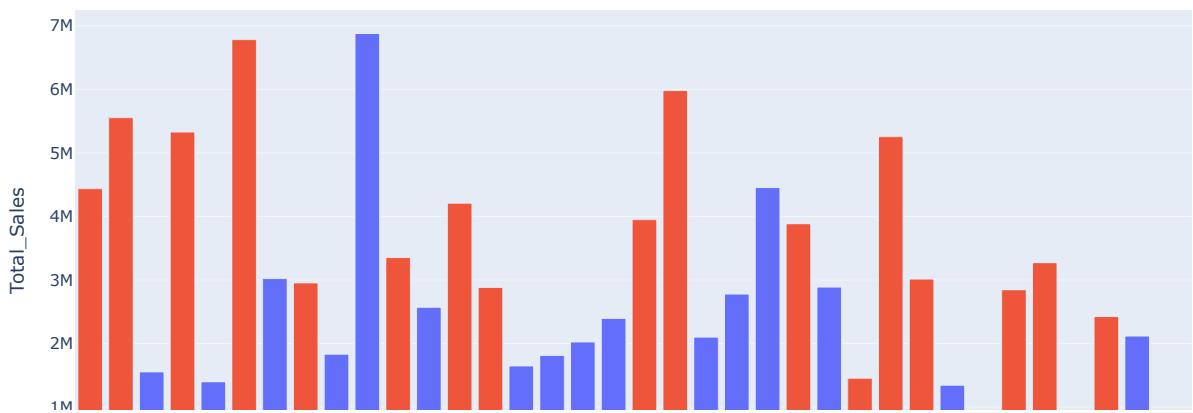
## Total Sales by Store



\*\*\*\*\*
\*\*\*\*\*

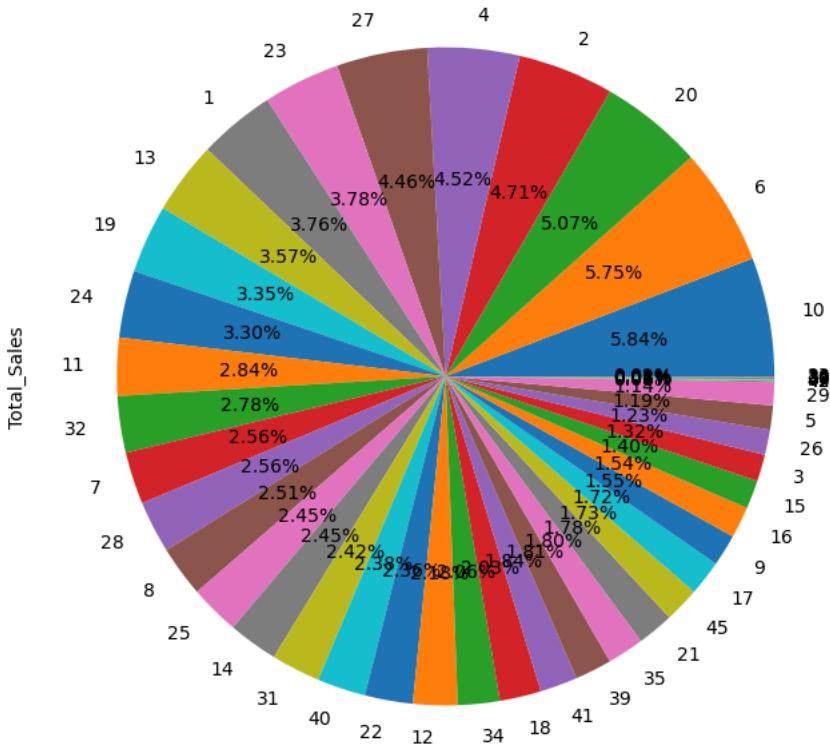
Dept: 10

## Total Sales by Store





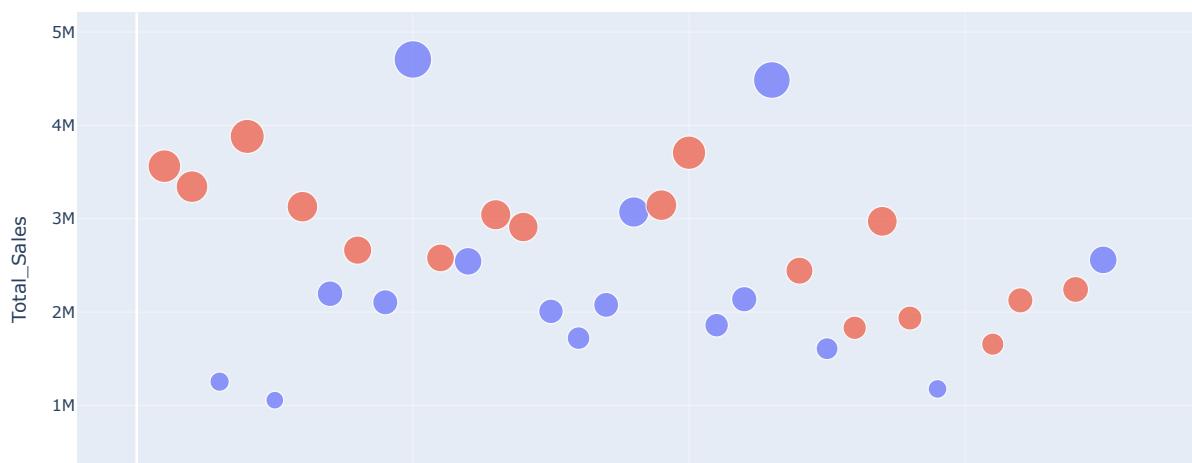
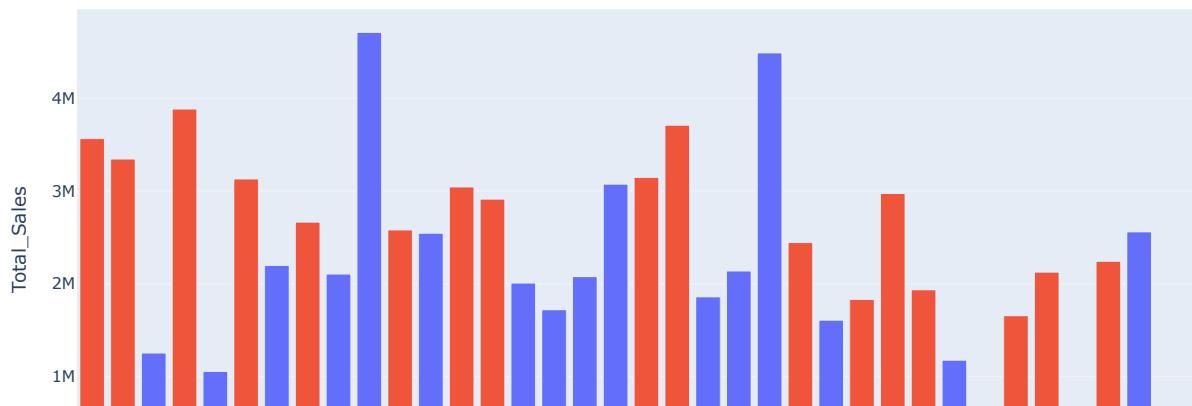
Total Sales by Store



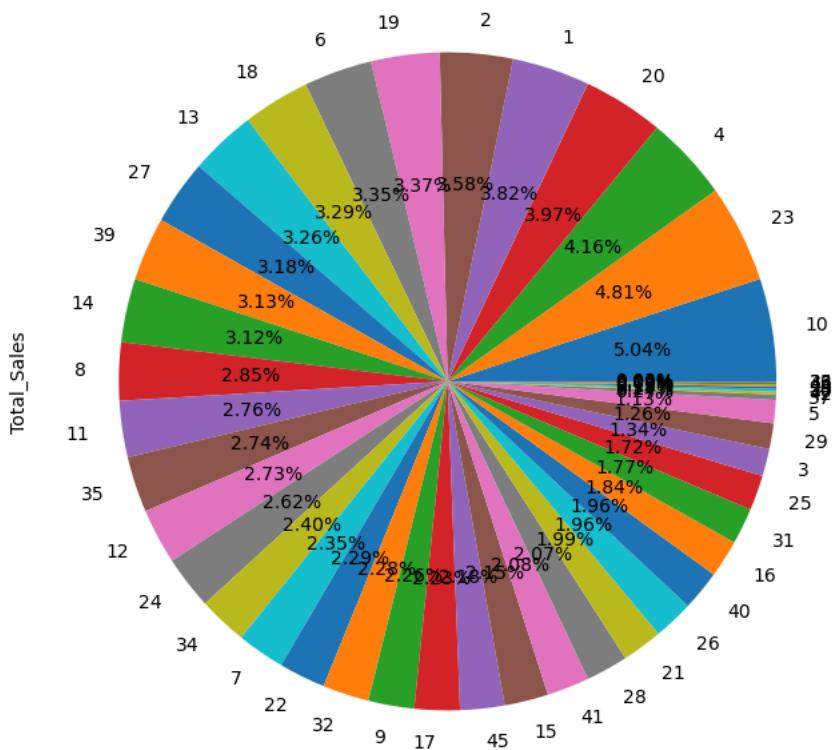
\*\*\*\*\*
\*\*\*\*\*  
\*\*\*\*\*

Dept: 11

## Total Sales by Store



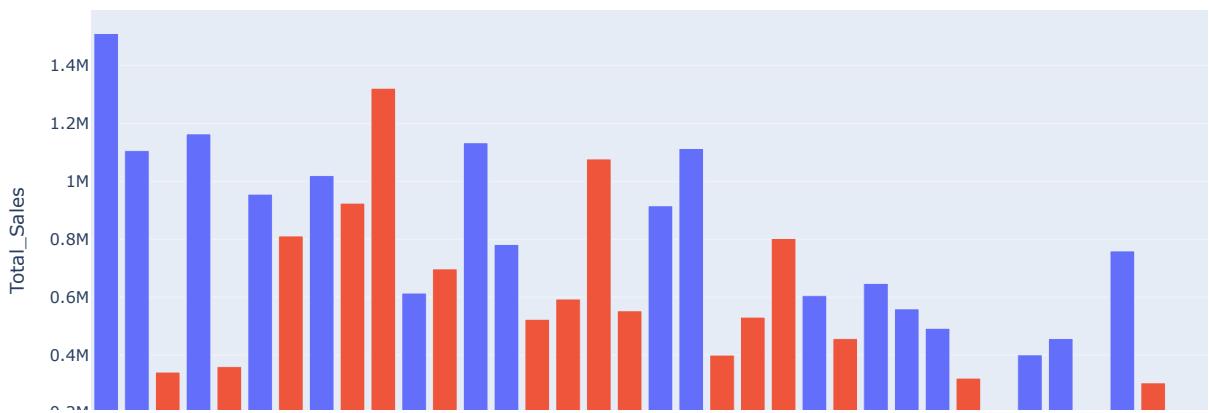
## Total Sales by Store

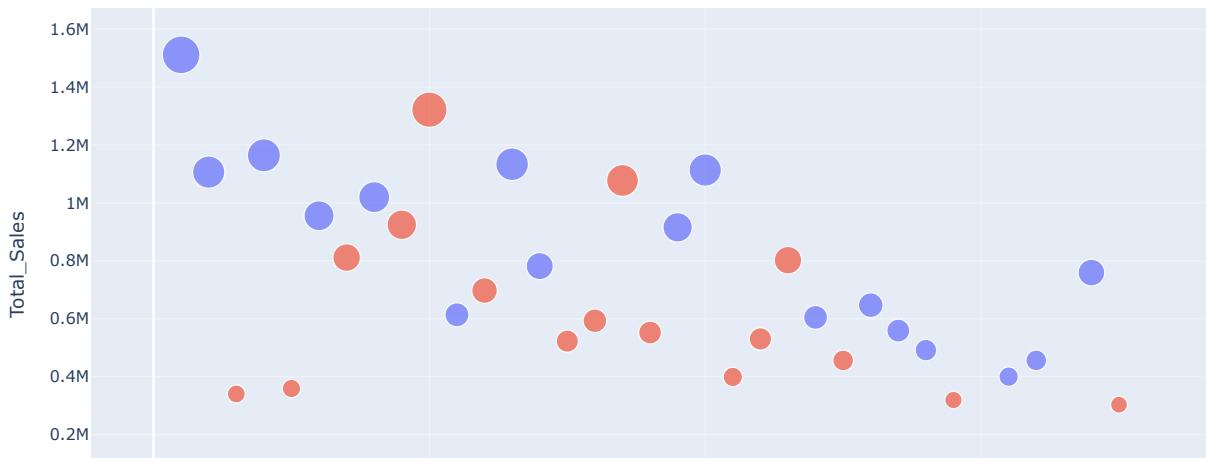


\*\*\*\*\*
\*\*\*\*\*

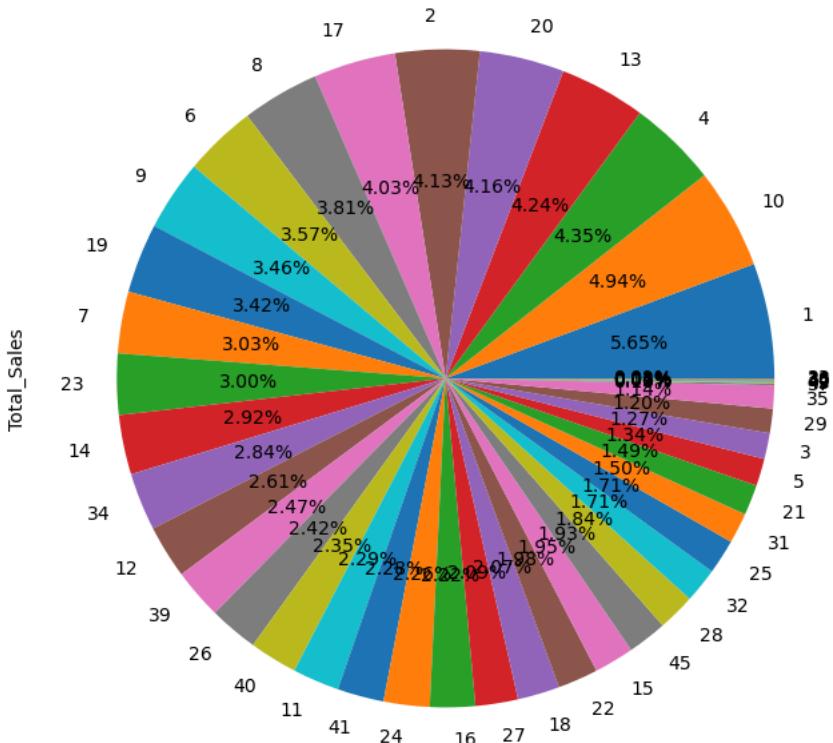
Dept: 12

## Total Sales by Store





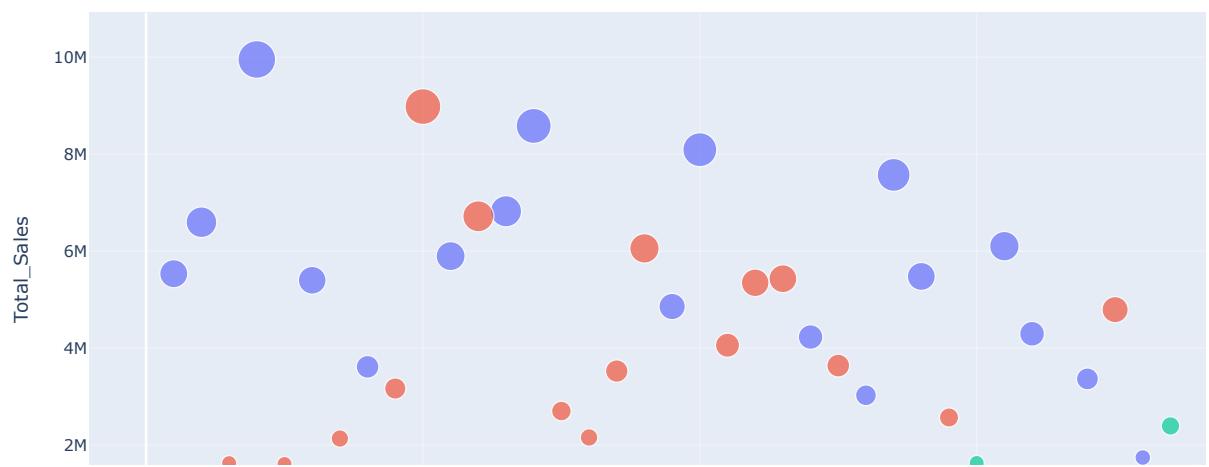
Total Sales by Store



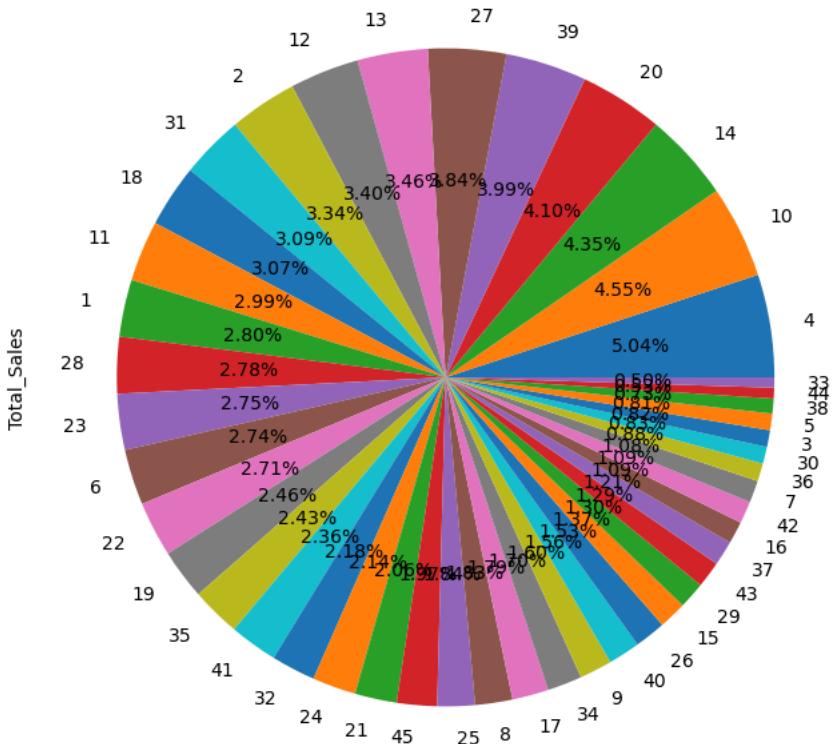
```
*****
*****
```

Dept: 13

## Total Sales by Store



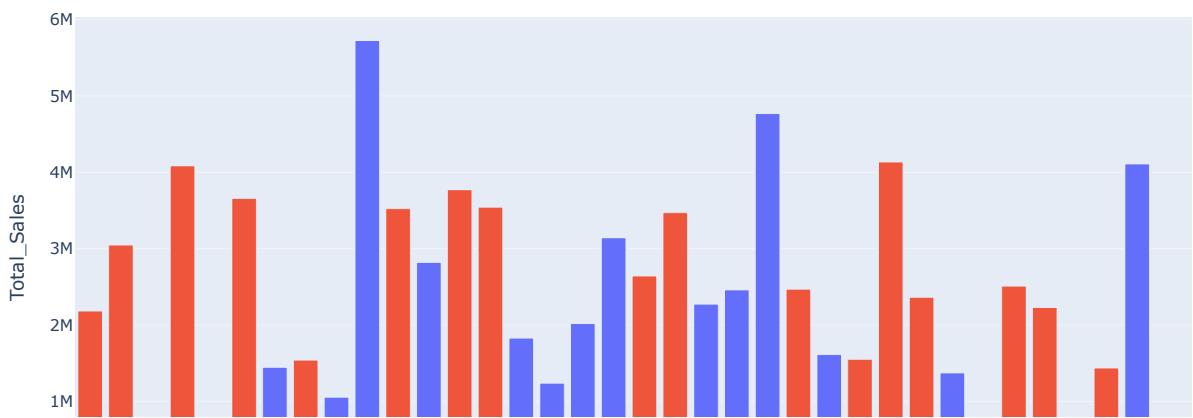
## Total Sales by Store



\*\*\*\*\*
\*\*\*\*\*

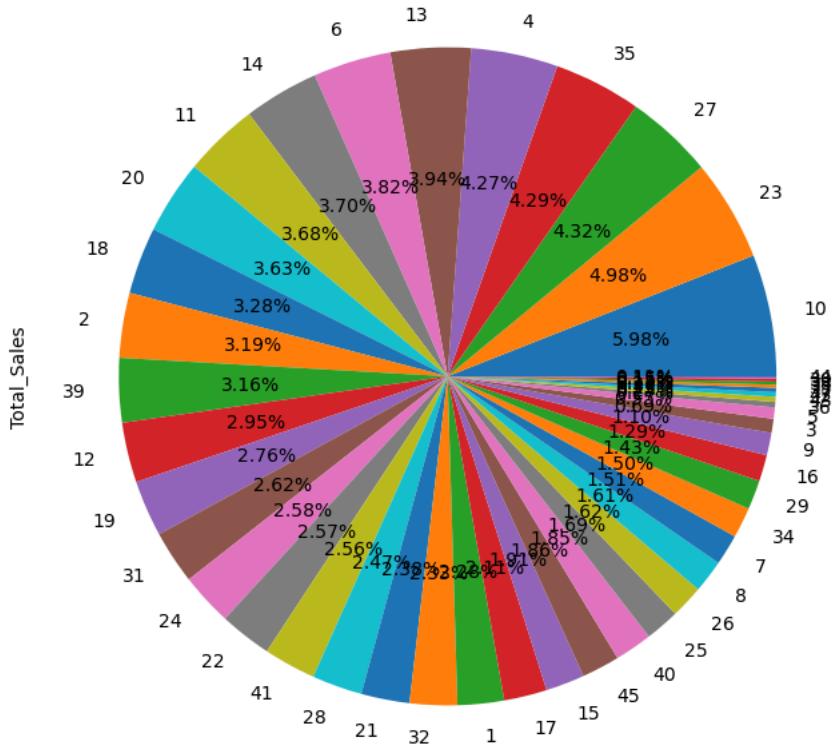
Dept: 14

## Total Sales by Store





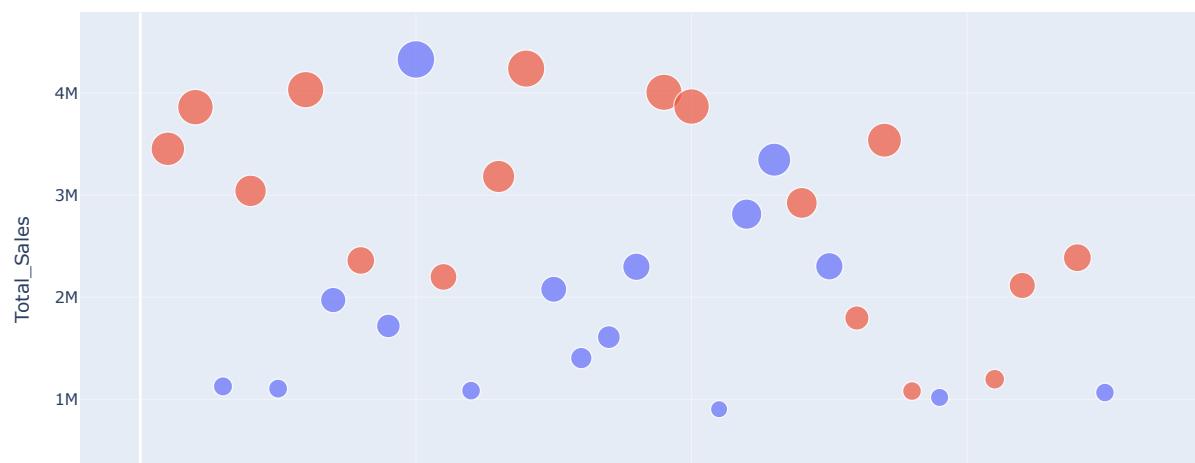
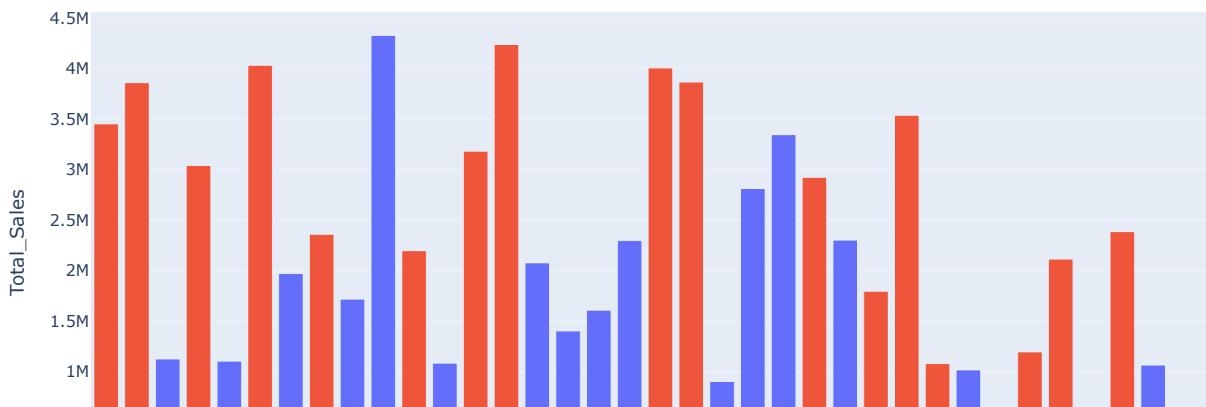
Total Sales by Store



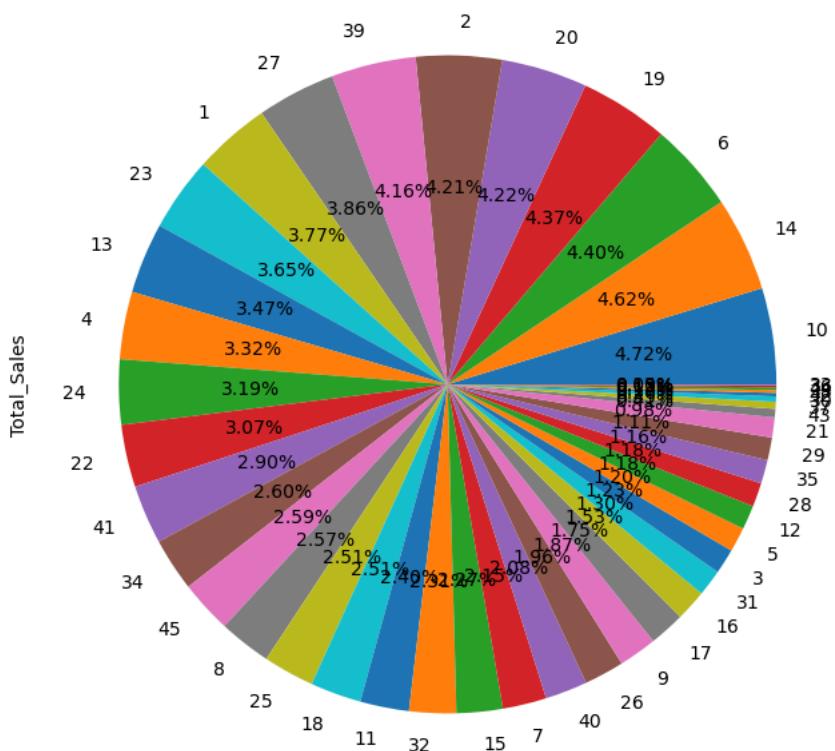
\*\*\*\*\*
\*\*\*\*\*

Dept: 16

## Total Sales by Store



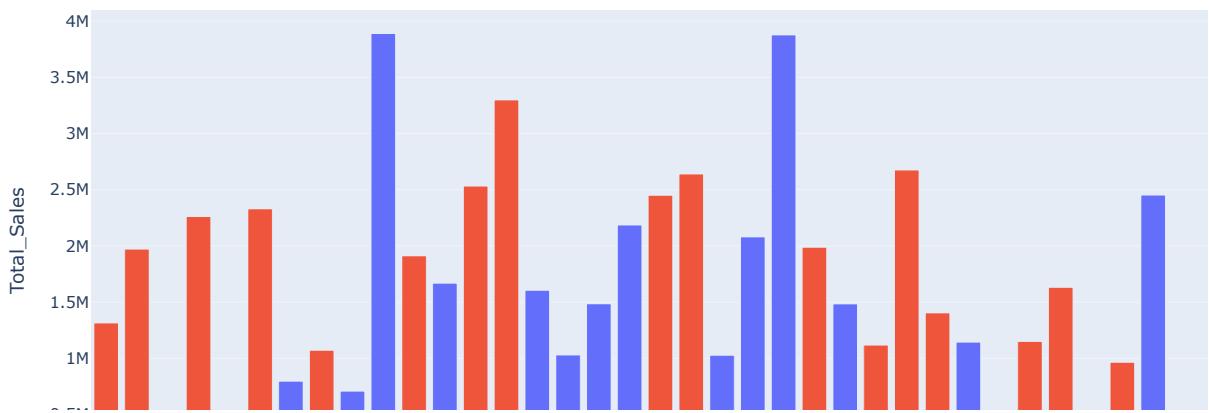
## Total Sales by Store



\*\*\*\*\*
\*\*\*\*\*

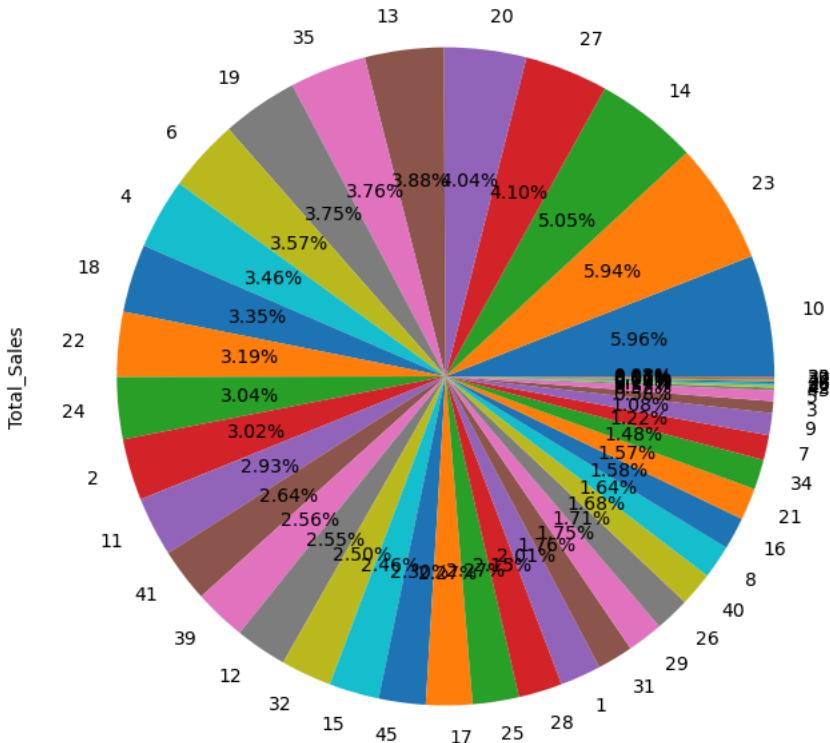
Dept: 17

## Total Sales by Store





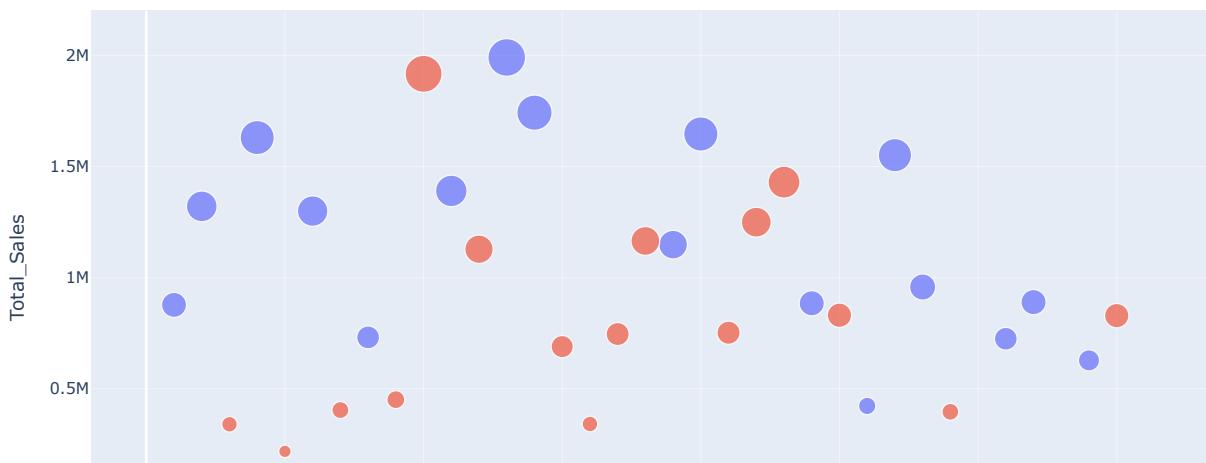
Total Sales by Store



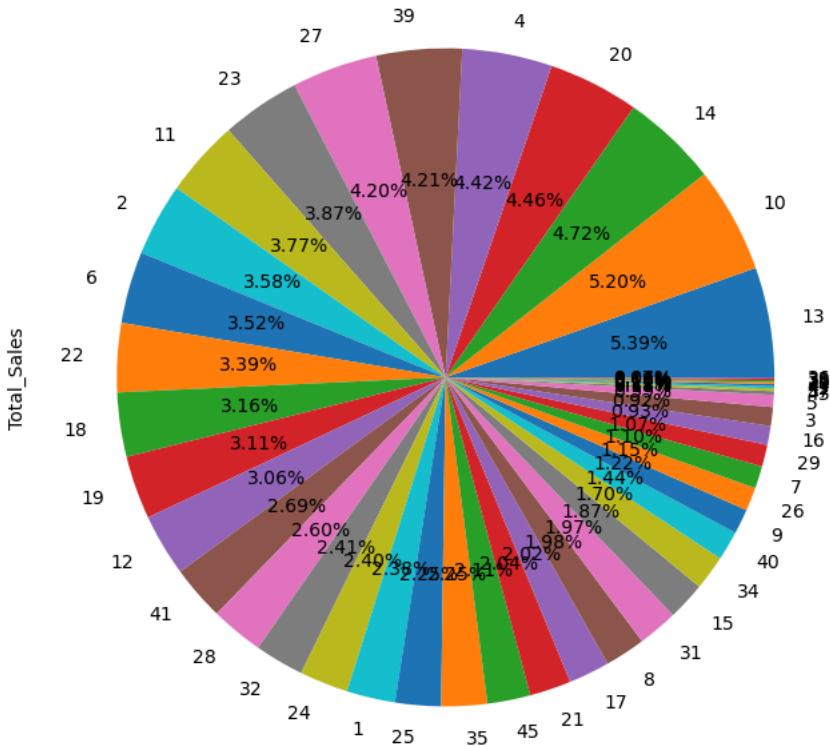
```
*****
*****
```

Dept: 18

## Total Sales by Store



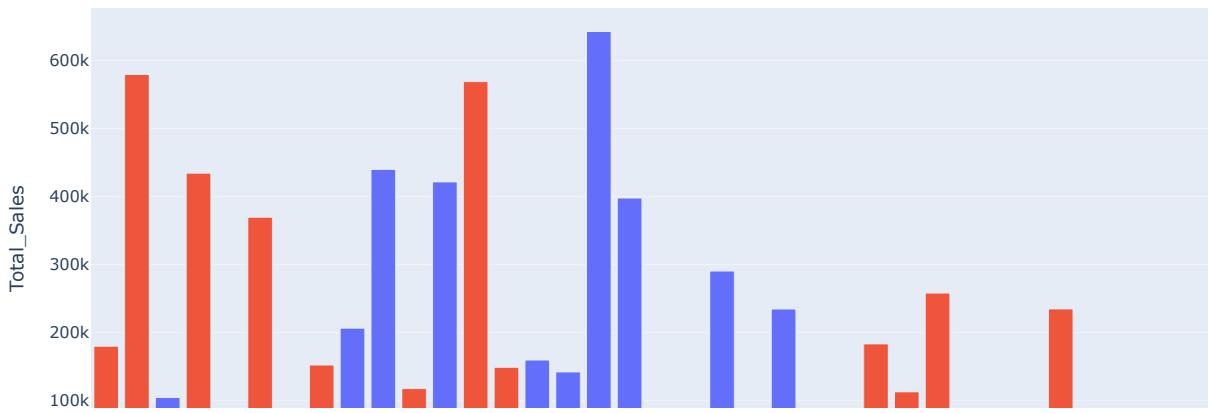
## Total Sales by Store



\*\*\*\*\*
\*\*\*\*\*

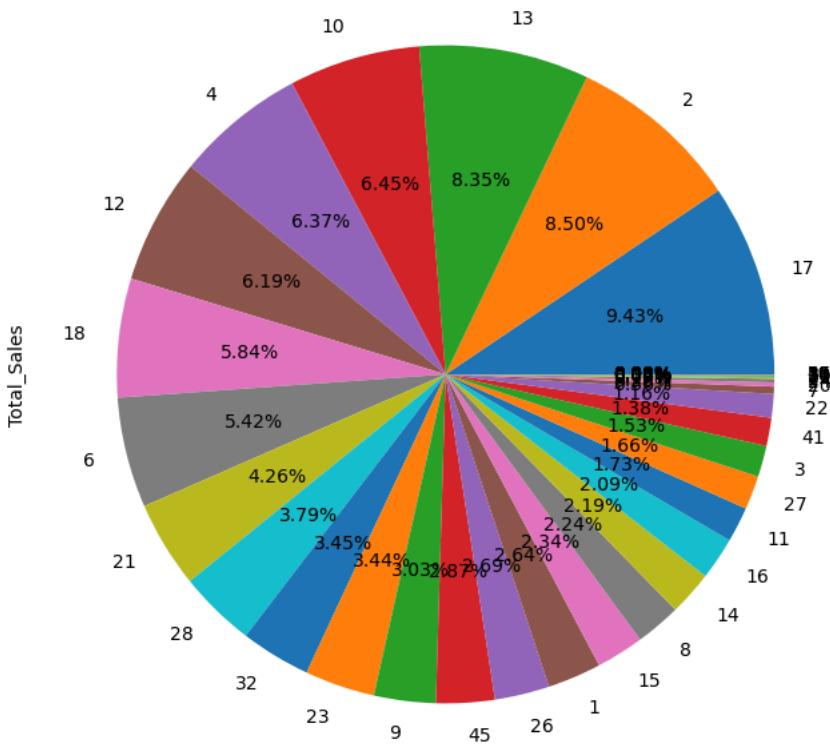
Dept: 19

## Total Sales by Store





Total Sales by Store



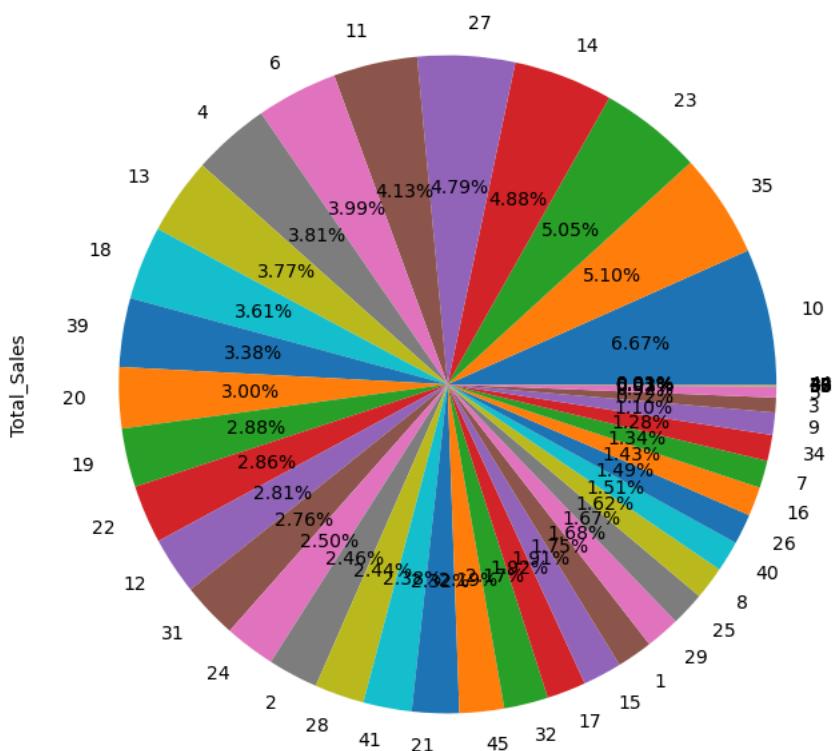
\*\*\*\*\*
\*\*\*\*\*  
\*\*\*\*\*

Dept: 20

## Total Sales by Store



## Total Sales by Store

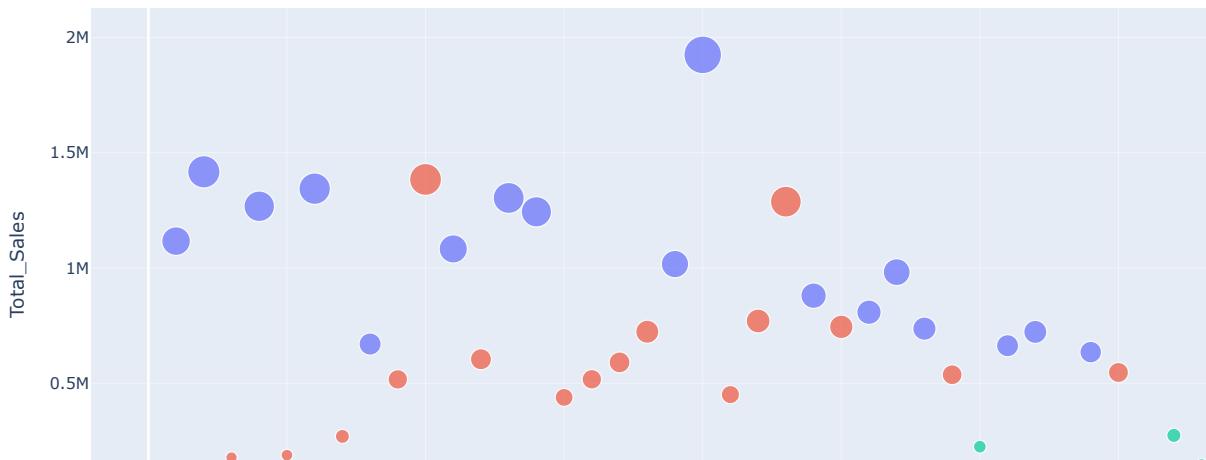


\*\*\*\*\*
\*\*\*\*\*

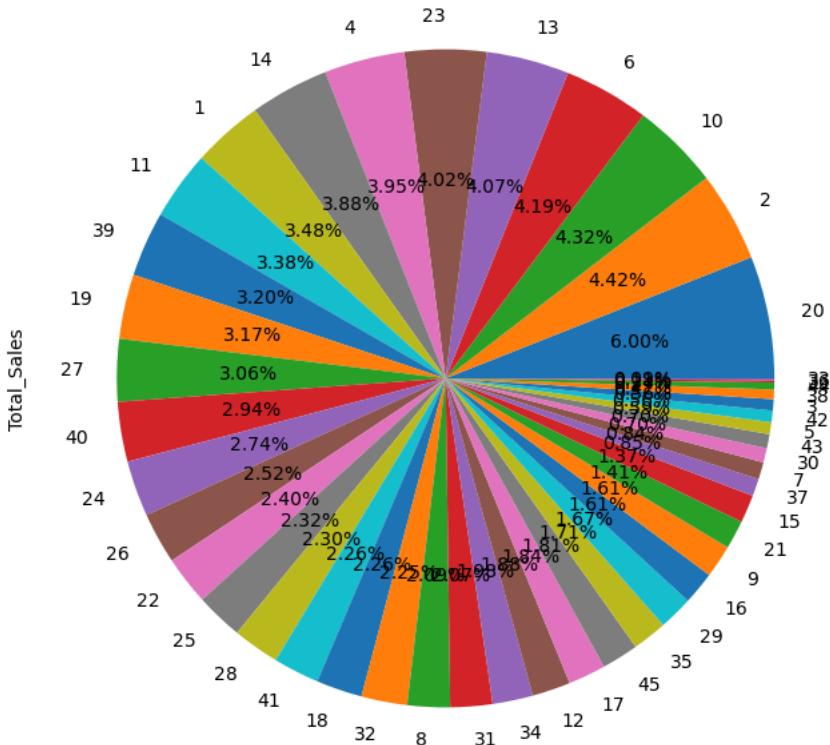
Dept: 21

## Total Sales by Store





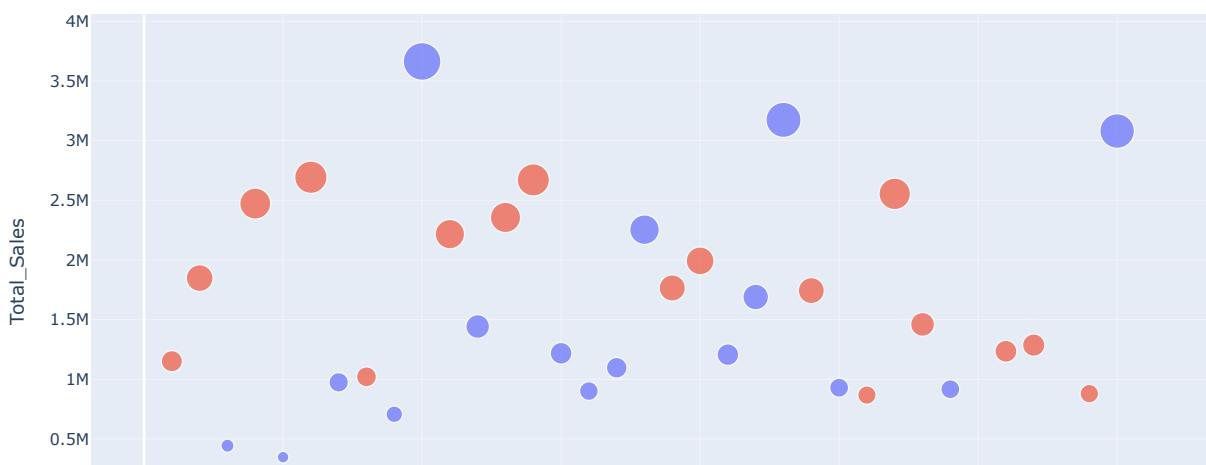
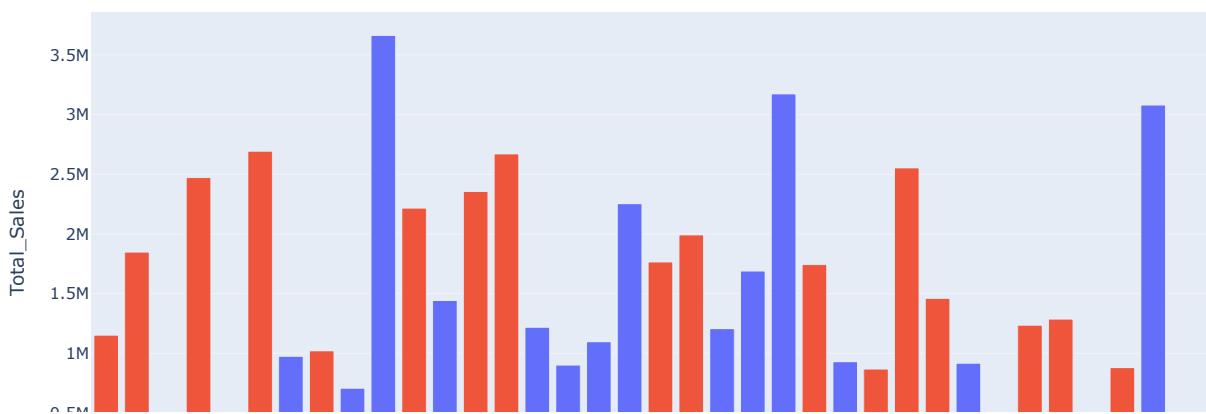
Total Sales by Store



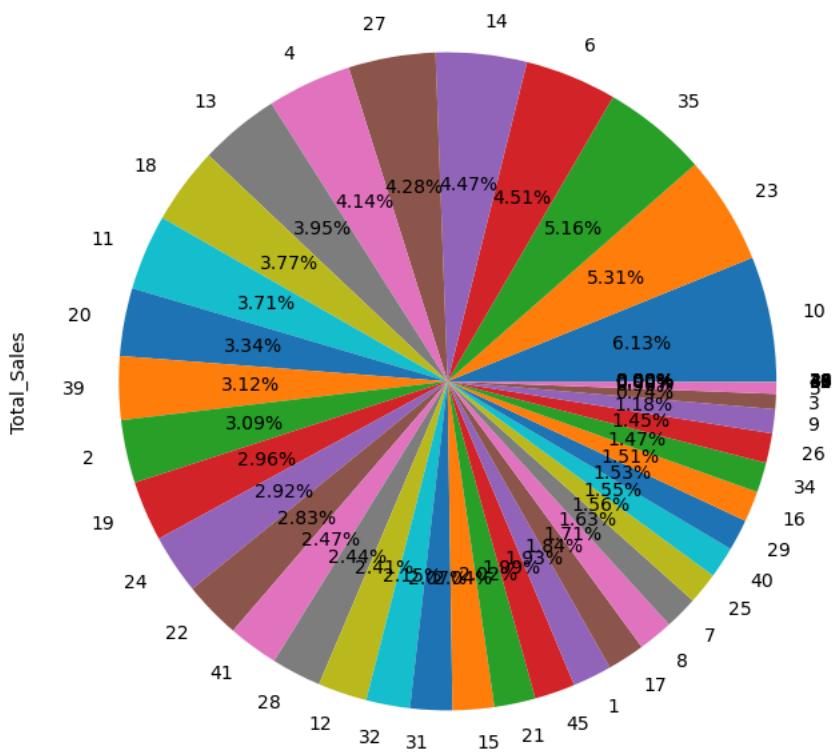
\*\*\*\*\*
\*\*\*\*\*  
\*\*\*\*\*

Dept: 22

## Total Sales by Store



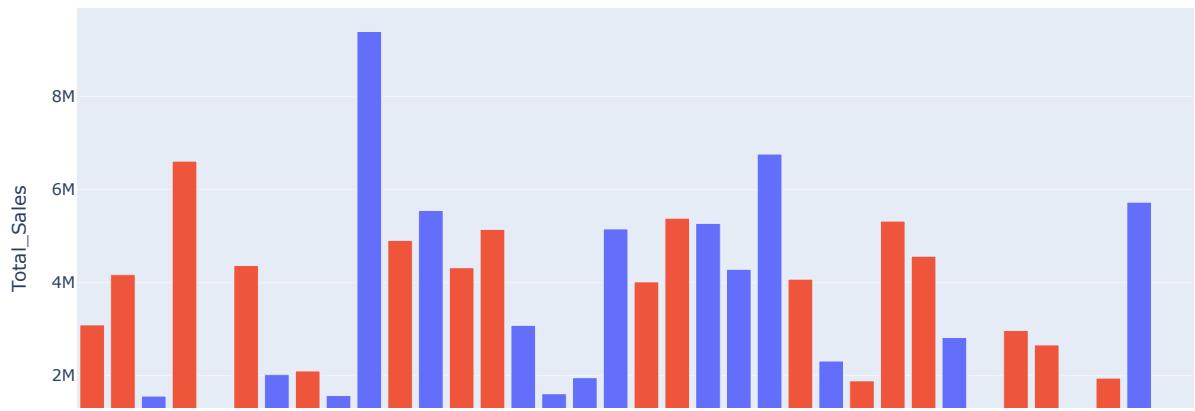
## Total Sales by Store

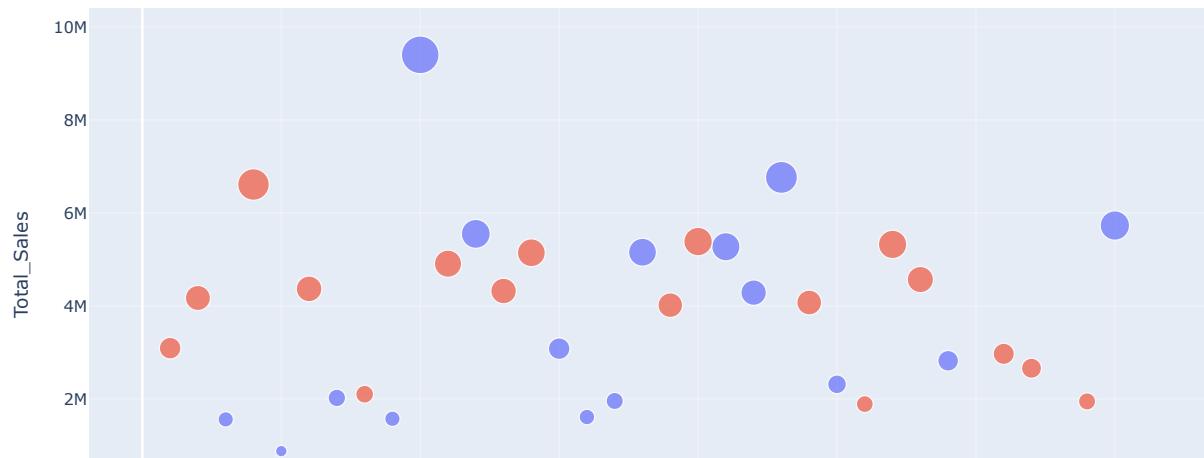


\*\*\*\*\*
\*\*\*\*\*

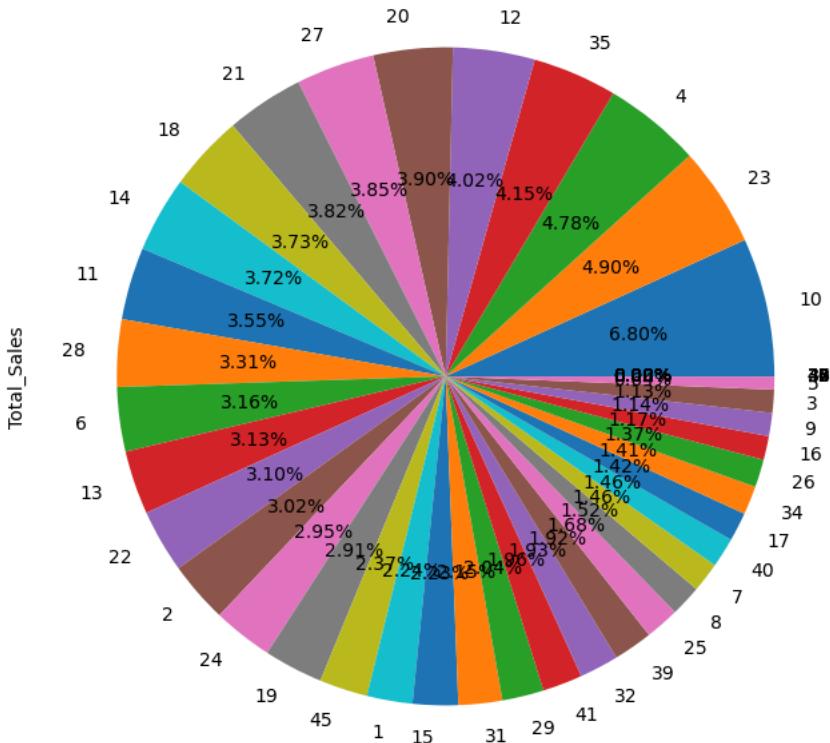
Dept: 23

## Total Sales by Store





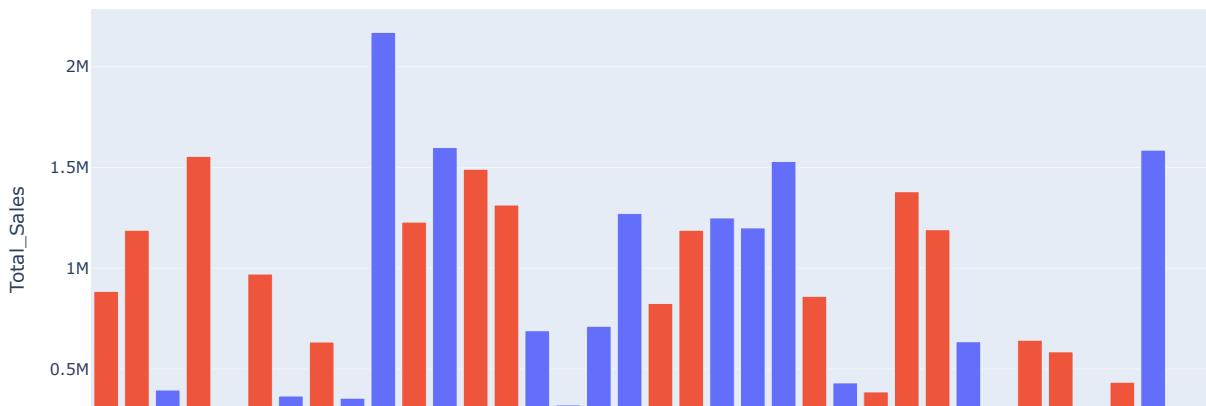
Total Sales by Store



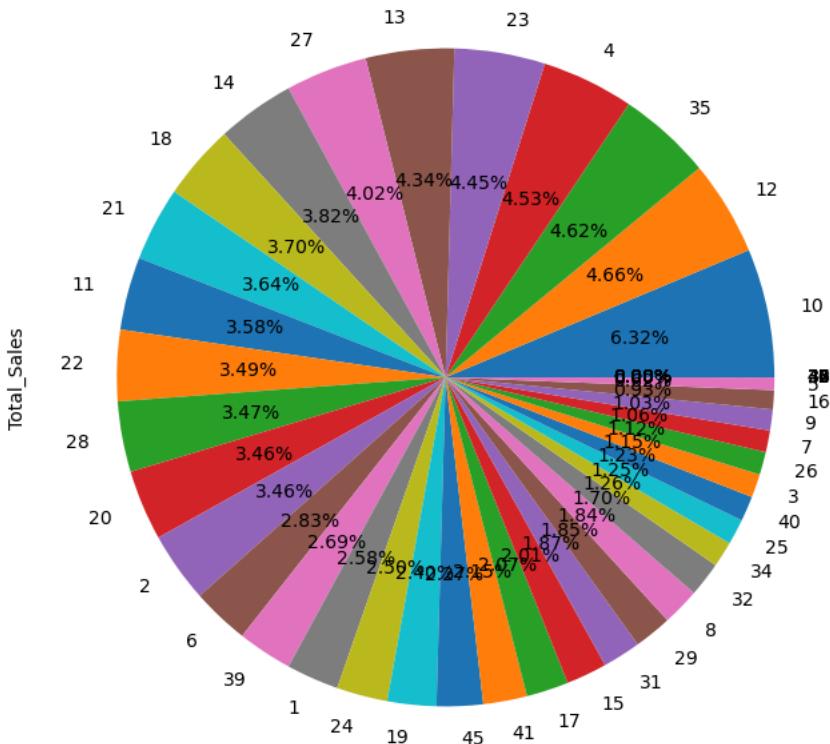
\*\*\*\*\*
\*\*\*\*\*  
\*\*\*\*\*

Dept: 24

## Total Sales by Store



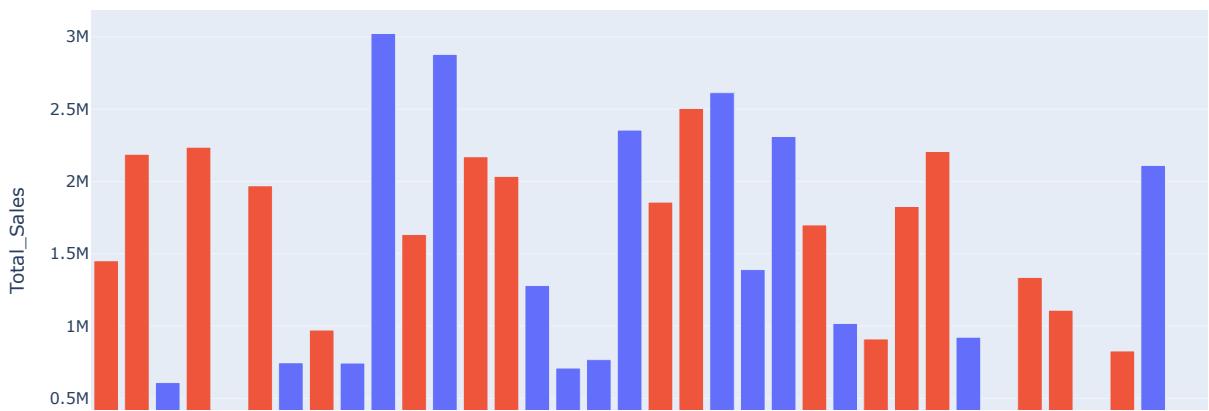
## Total Sales by Store

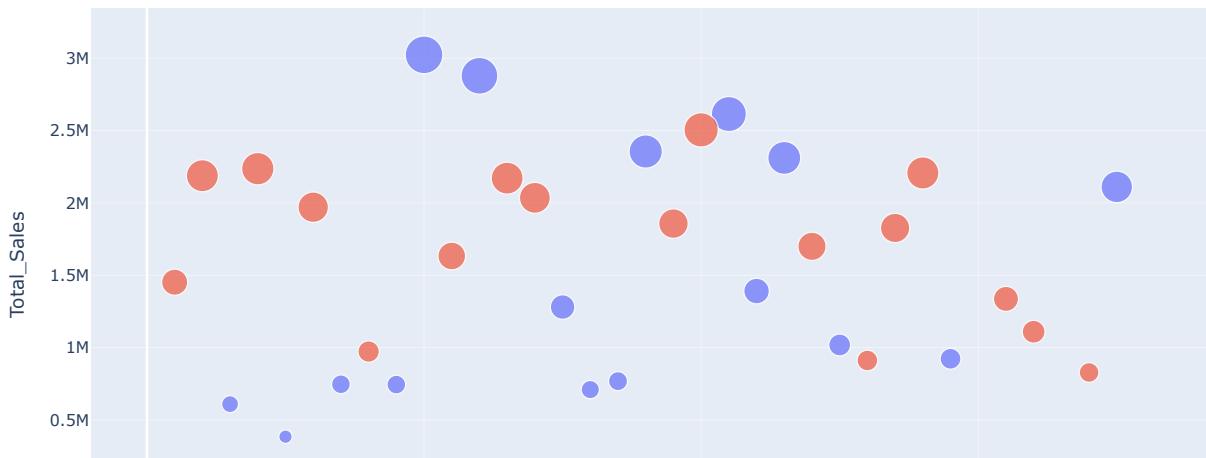


\*\*\*\*\*
\*\*\*\*\*

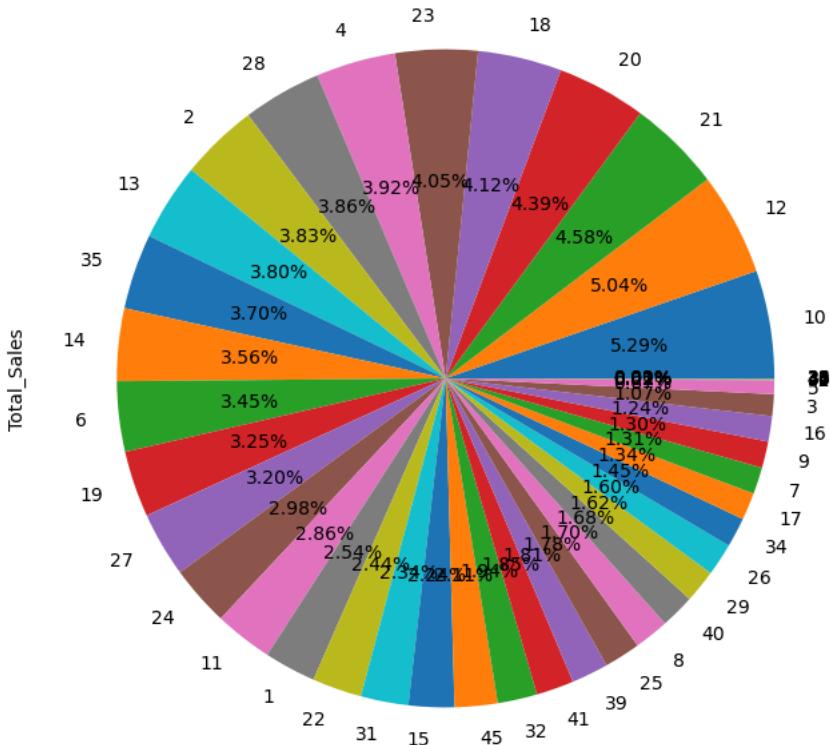
Dept: 25

## Total Sales by Store





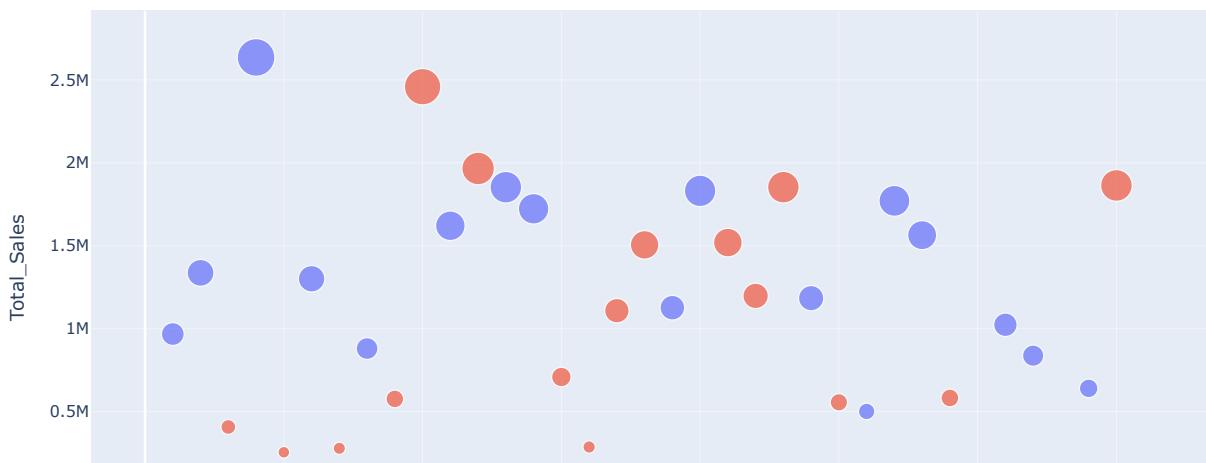
Total Sales by Store



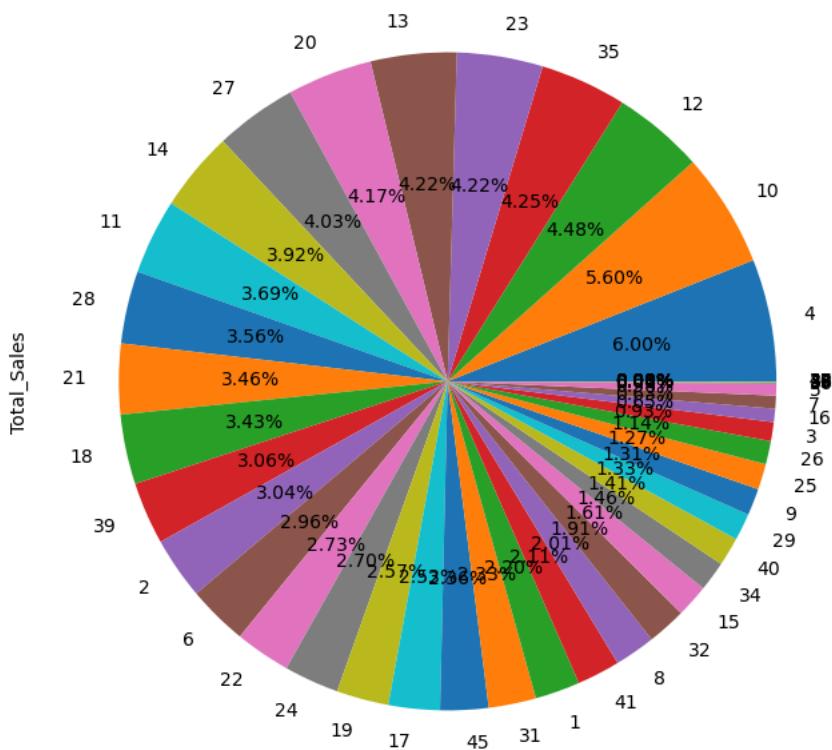
\*\*\*\*\*
\*\*\*\*\*  
\*\*\*\*\*

Dept: 26

## Total Sales by Store



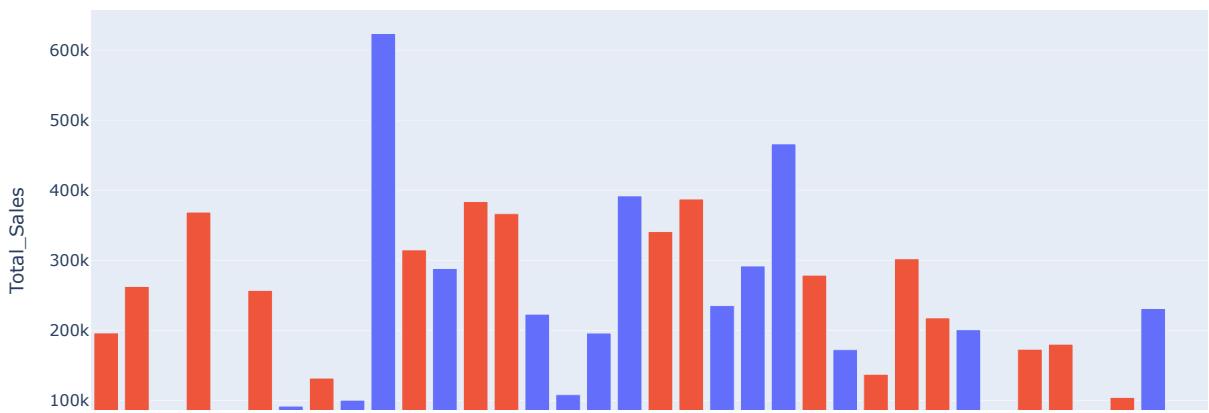
## Total Sales by Store

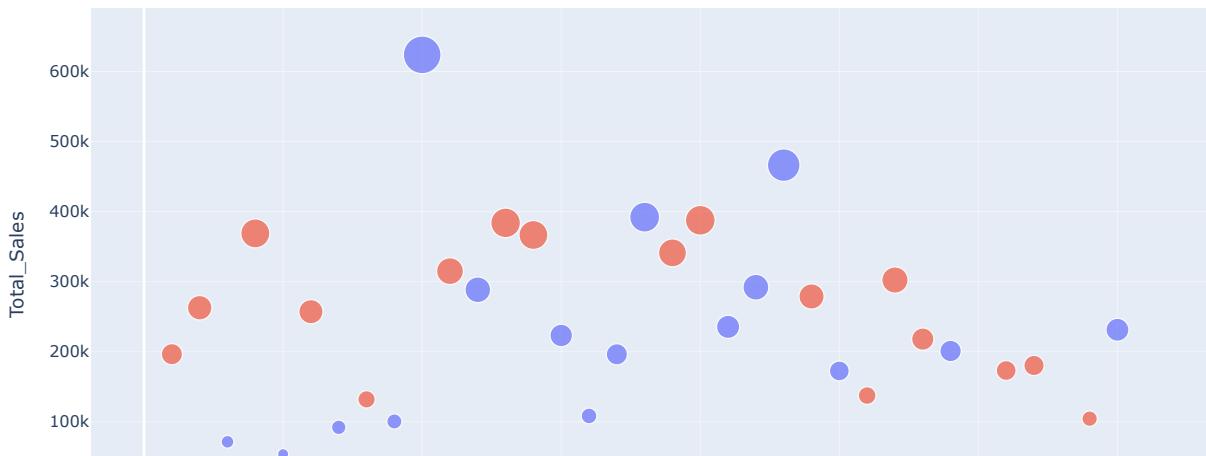


\*\*\*\*\*
\*\*\*\*\*

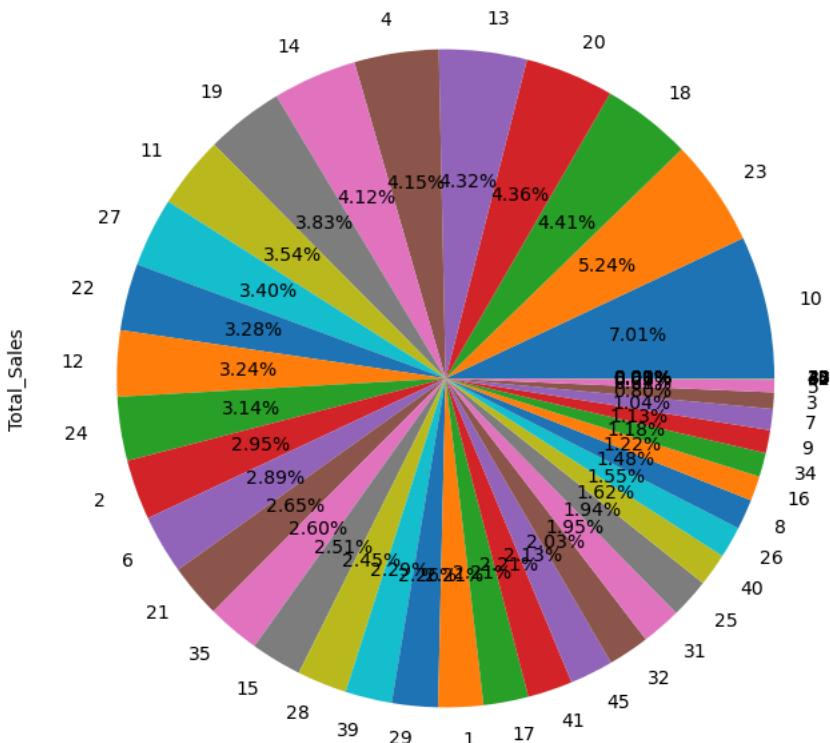
Dept: 27

## Total Sales by Store





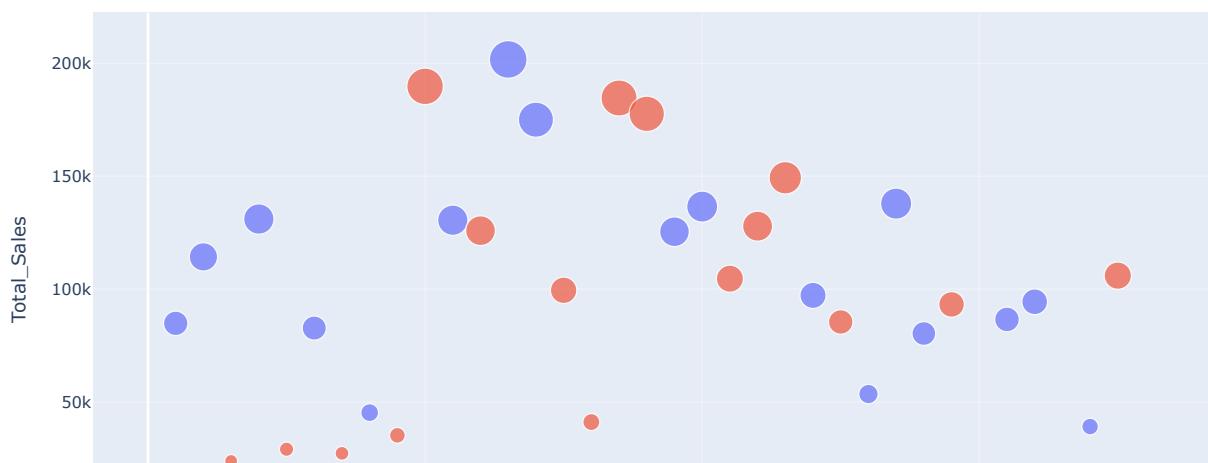
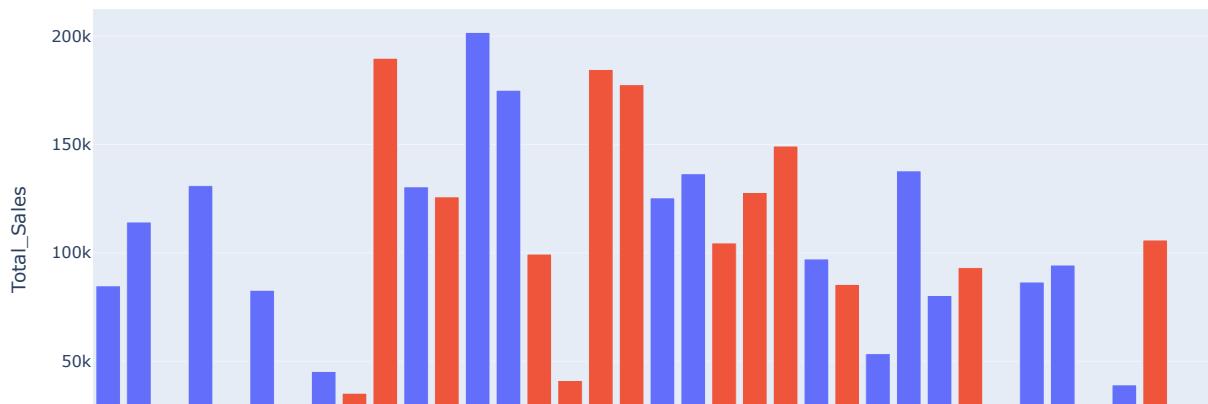
Total Sales by Store



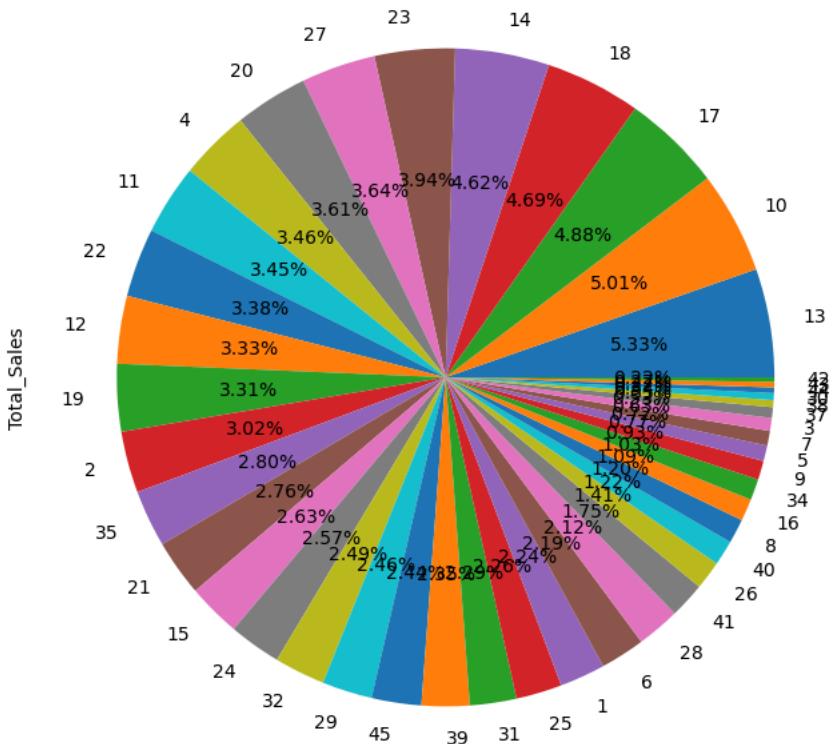
\*\*\*\*\*
\*\*\*\*\*  
\*\*\*\*\*

Dept: 28

## Total Sales by Store



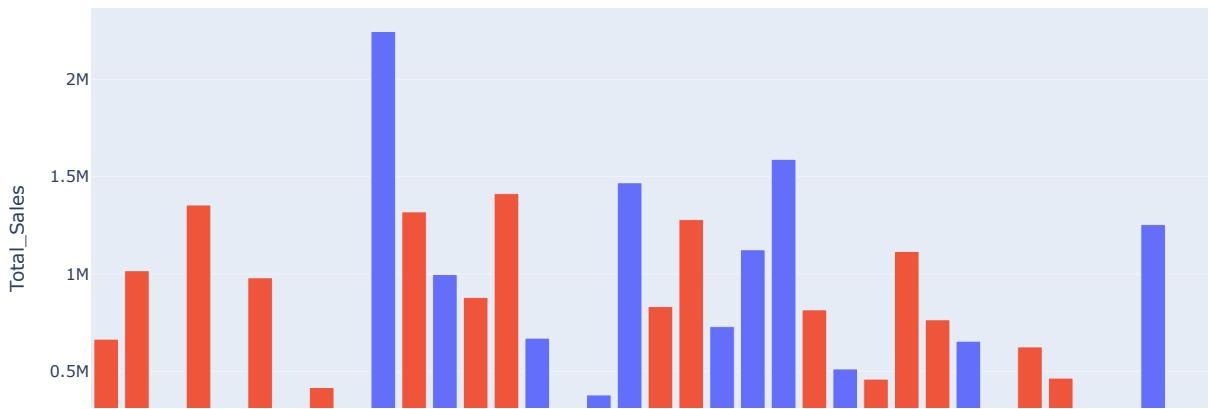
## Total Sales by Store

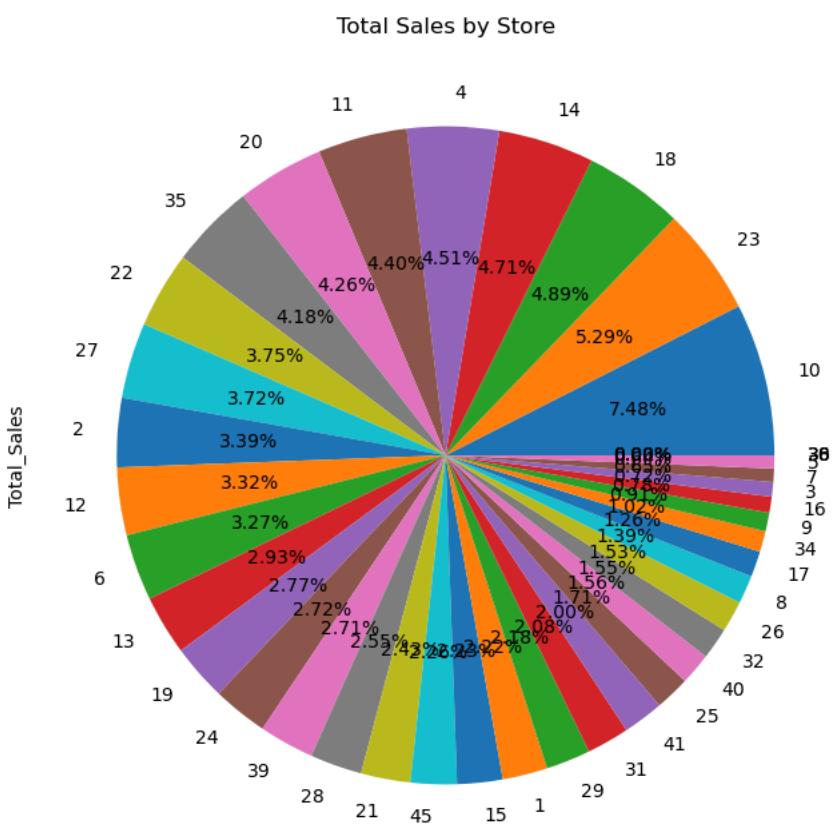
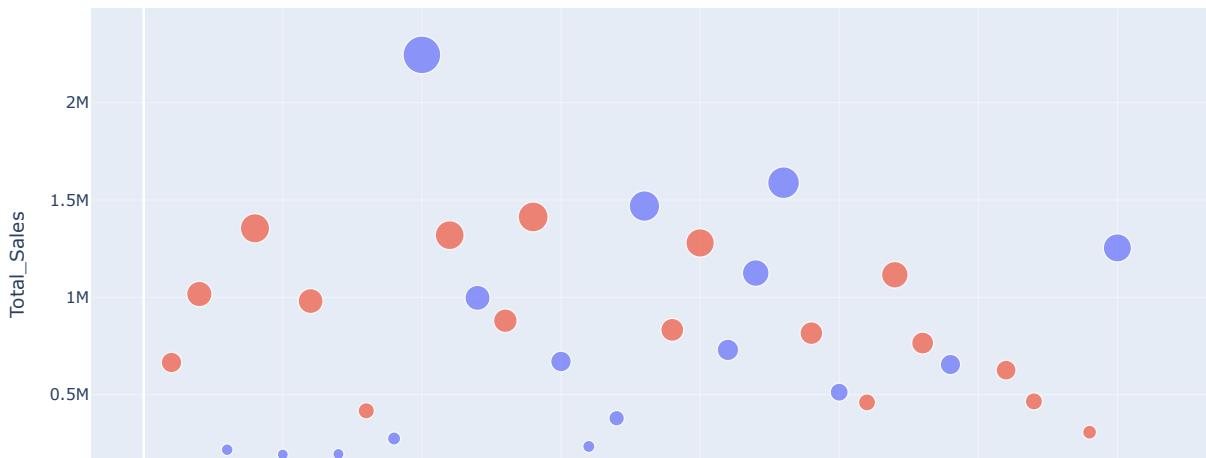


\*\*\*\*\*
\*\*\*\*\*

Dept: 29

## Total Sales by Store





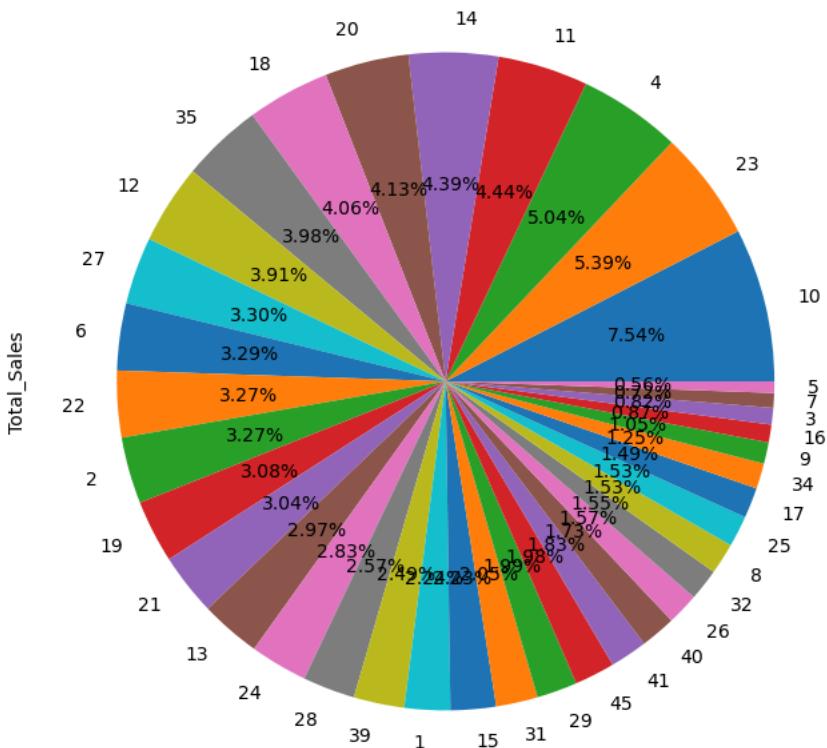
\*\*\*\*\*
\*\*\*\*\*

Dept: 30

## Total Sales by Store



## Total Sales by Store

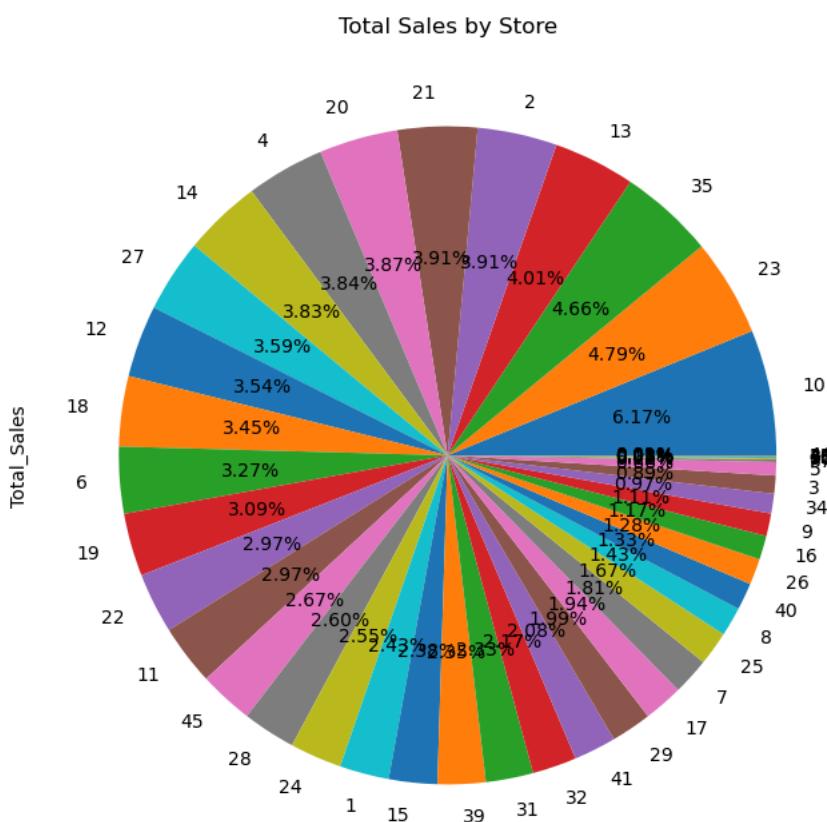
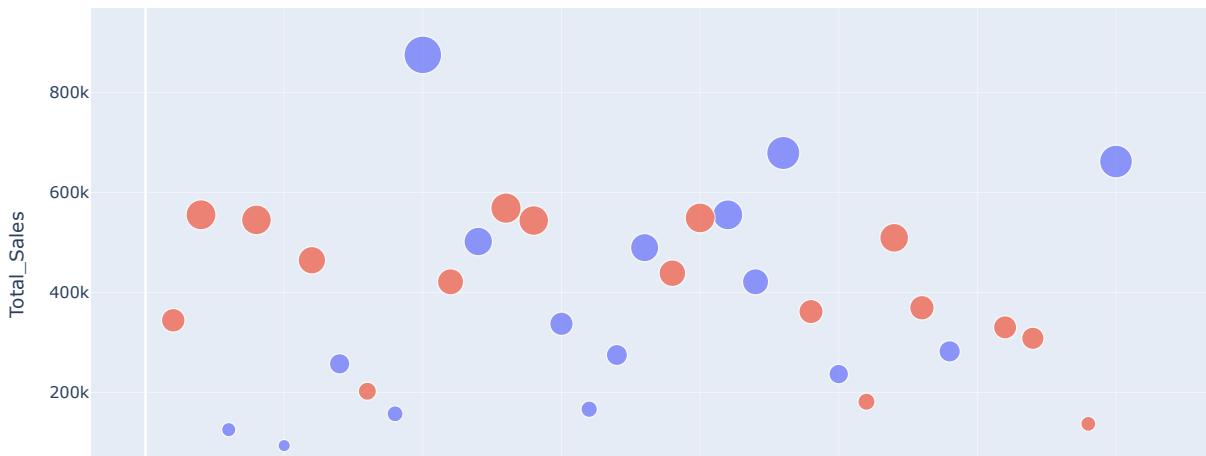


\*\*\*\*\*
\*\*\*\*\*

Dept: 31

## Total Sales by Store





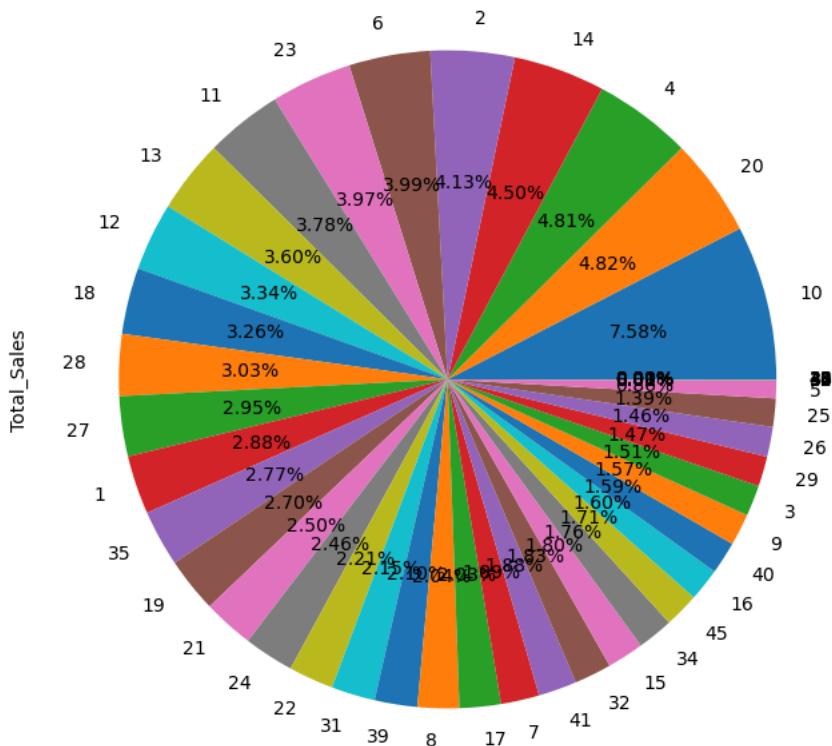
\*\*\*\*\*
\*\*\*\*\*  
\*\*\*\*\*

Dept: 32

## Total Sales by Store



## Total Sales by Store

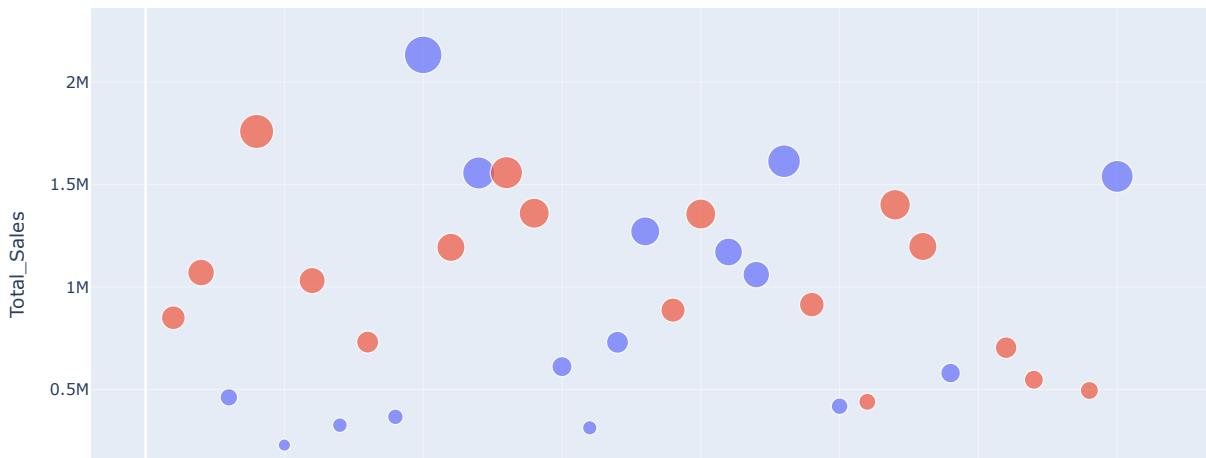


\*\*\*\*\*  
\*\*\*\*\*

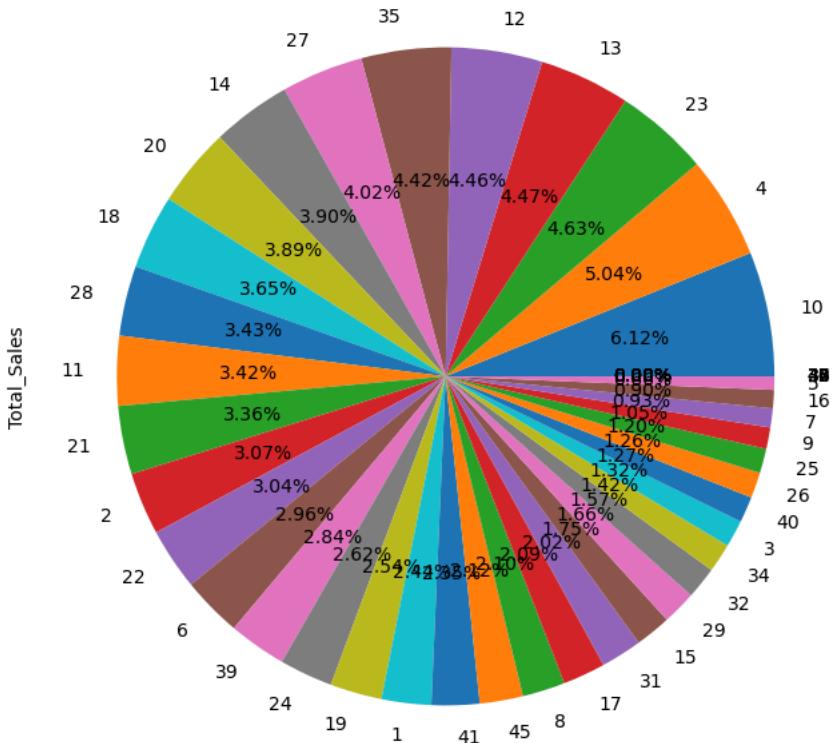
Dept: 33

## Total Sales by Store





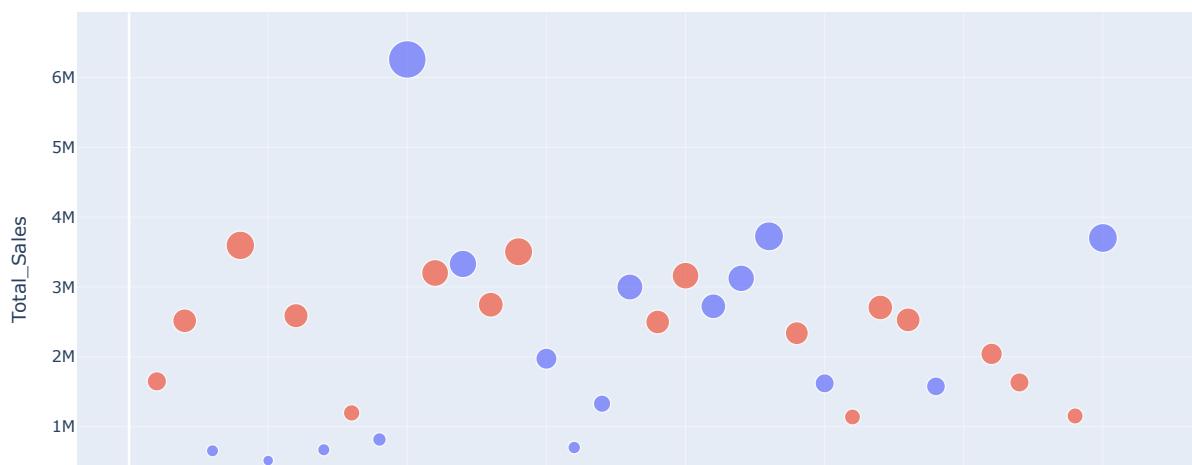
Total Sales by Store



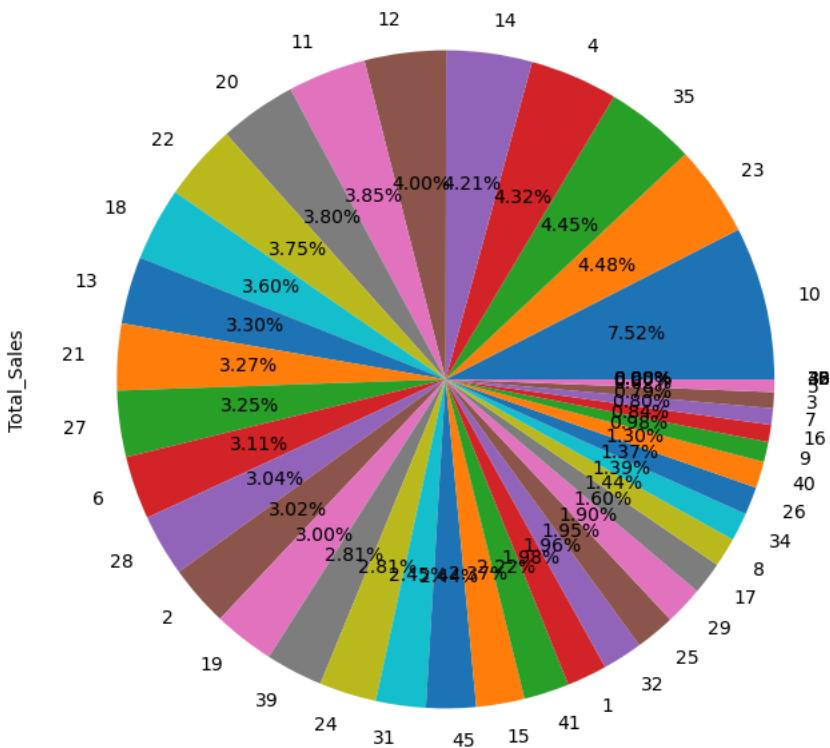
\*\*\*\*\*
\*\*\*\*\*  
\*\*\*\*\*

Dept: 34

## Total Sales by Store



## Total Sales by Store

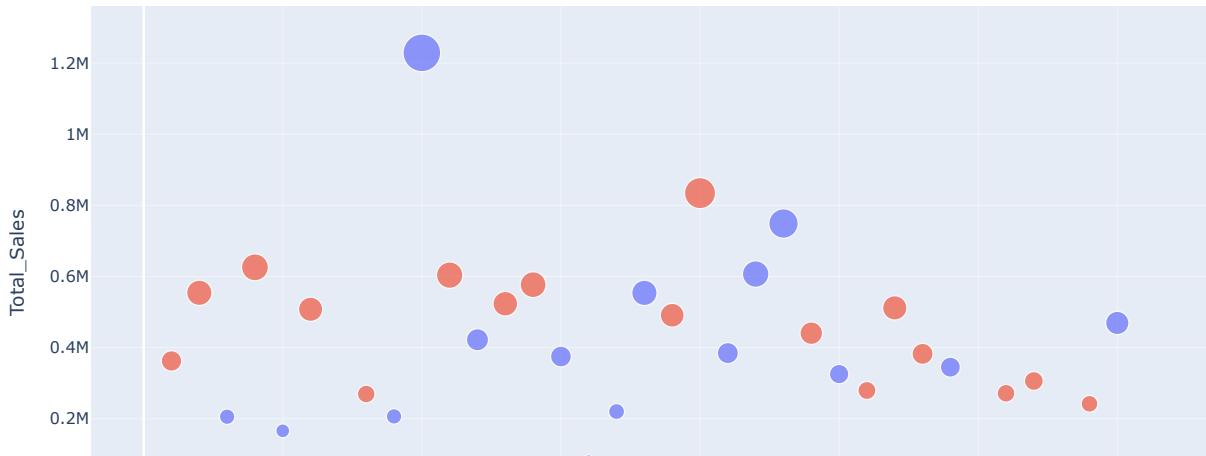


\*\*\*\*\*
\*\*\*\*\*

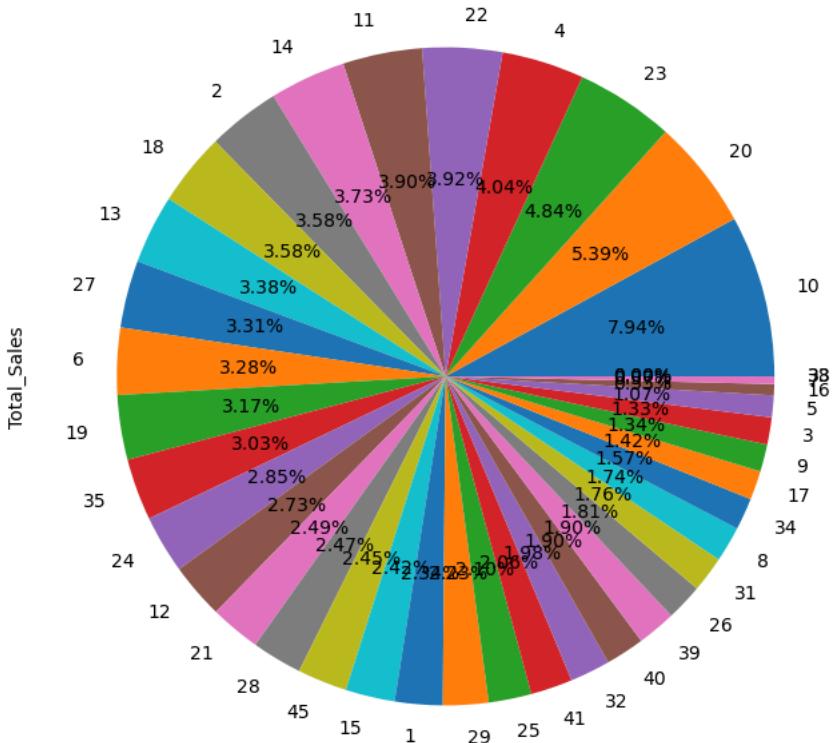
Dept: 35

## Total Sales by Store





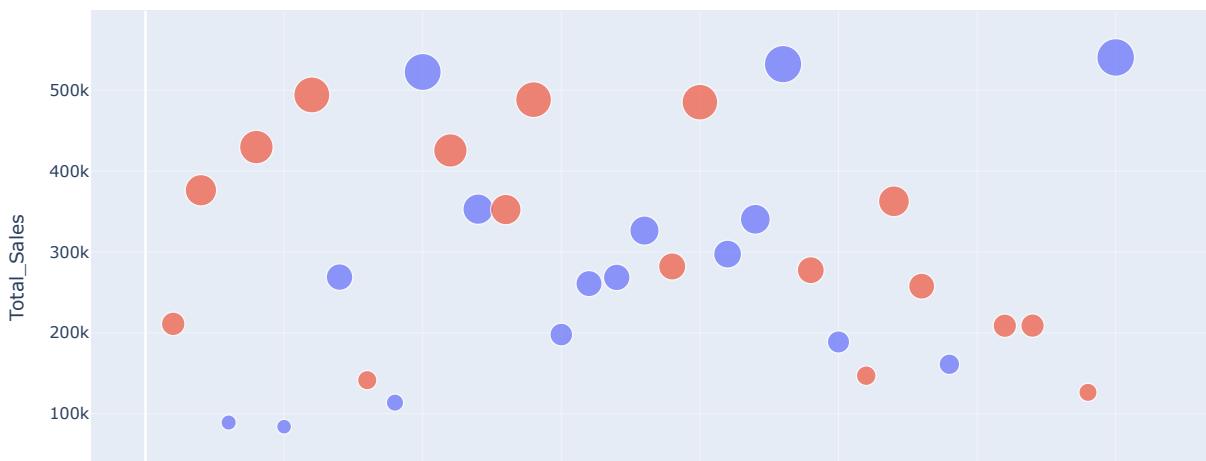
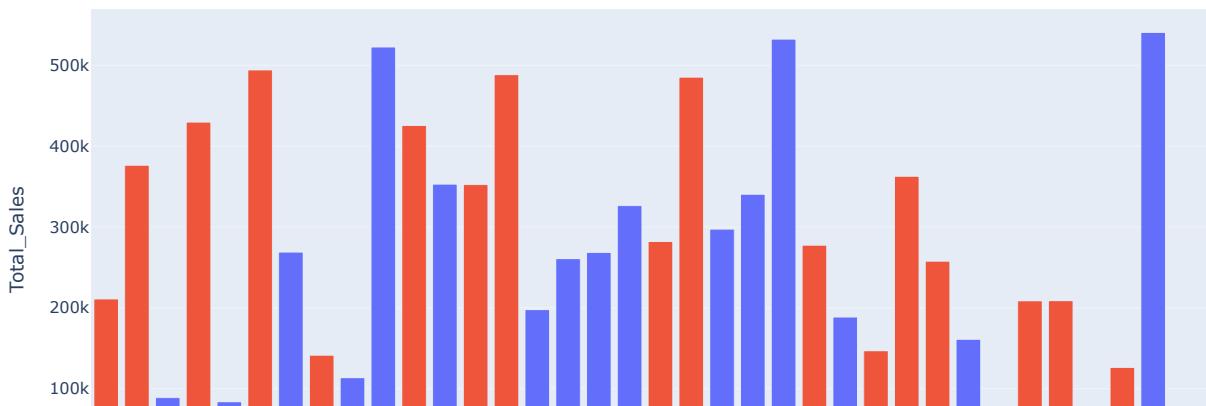
Total Sales by Store

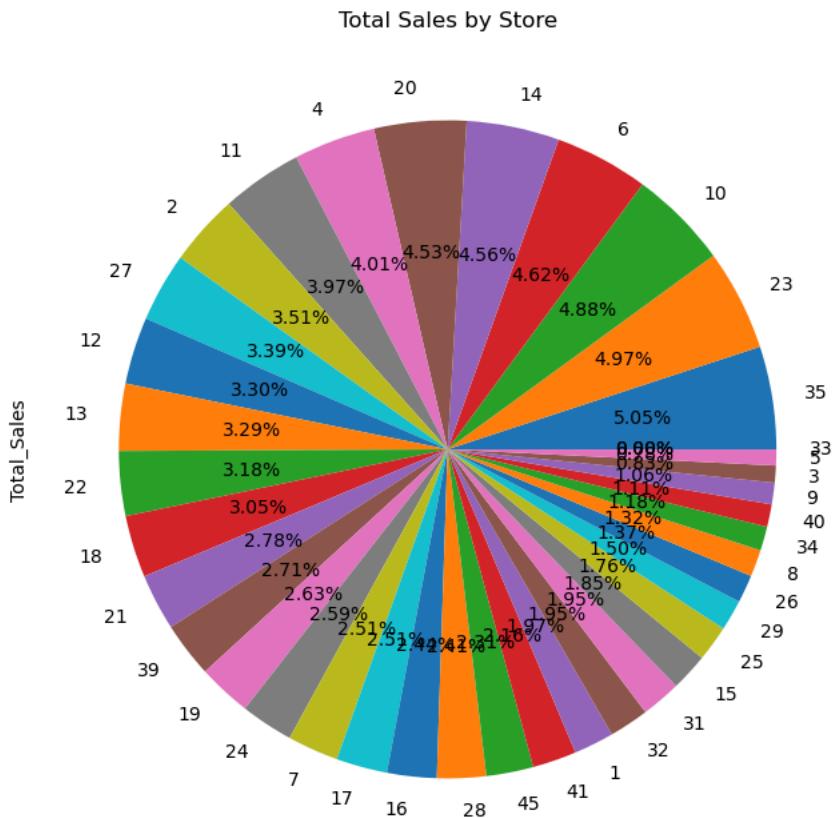


\*\*\*\*\*
\*\*\*\*\*  
\*\*\*\*\*

Dept: 36

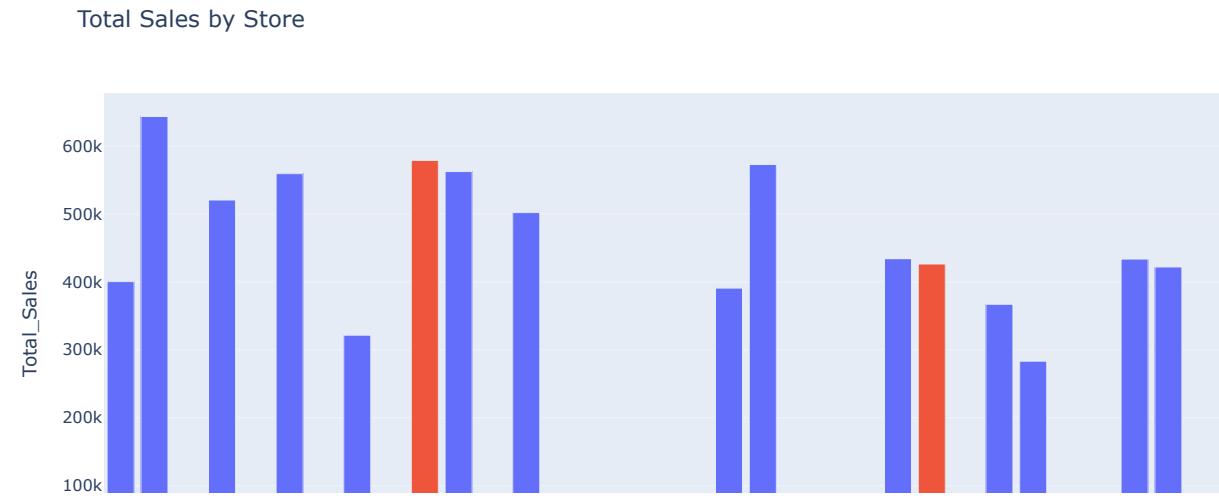
## Total Sales by Store

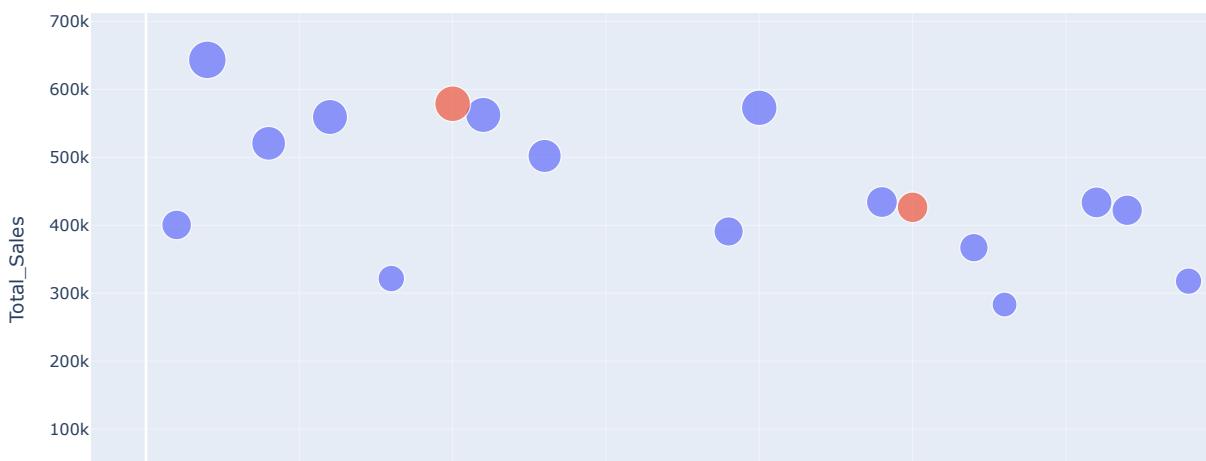




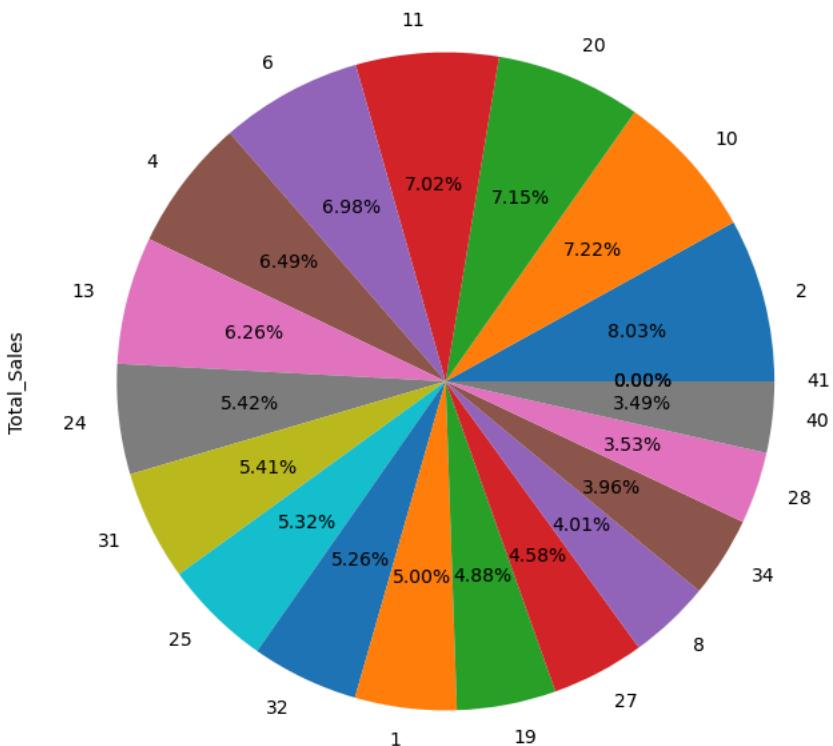
\*\*\*\*\*  
\*\*\*\*\*

Dept: 37





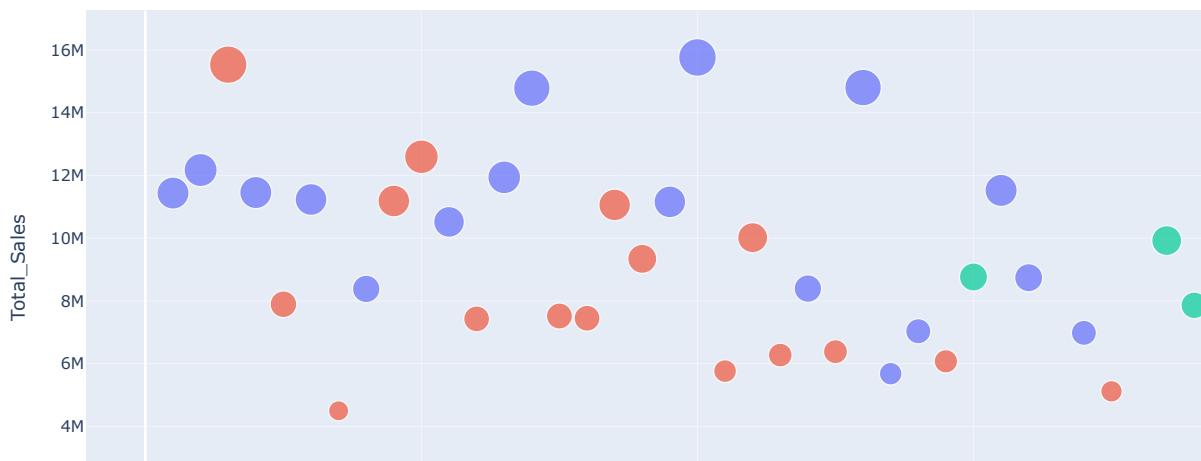
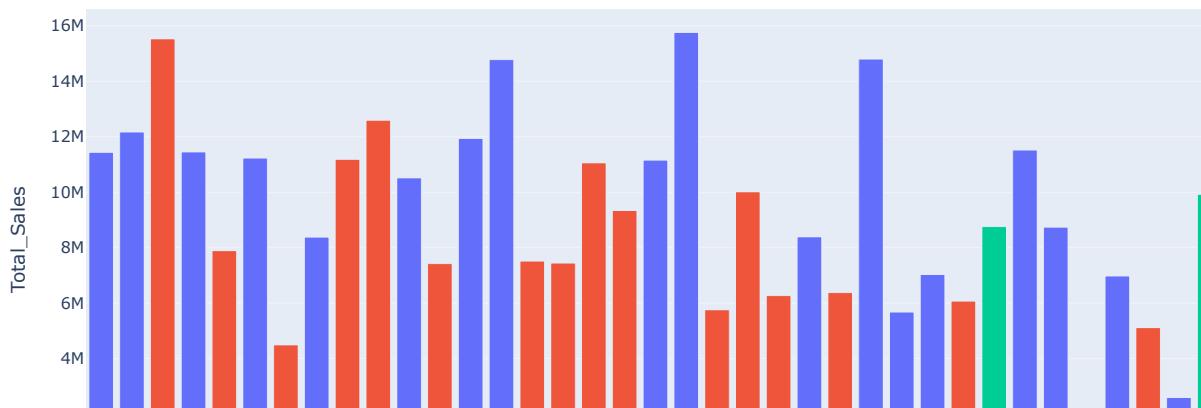
Total Sales by Store



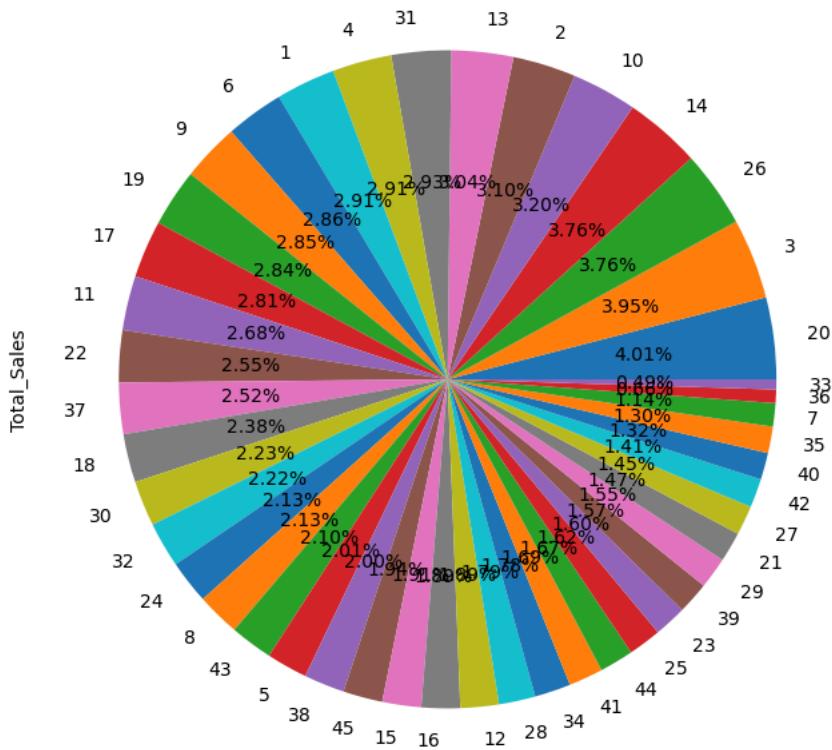
```
*****
*****
```

Dept: 38

## Total Sales by Store



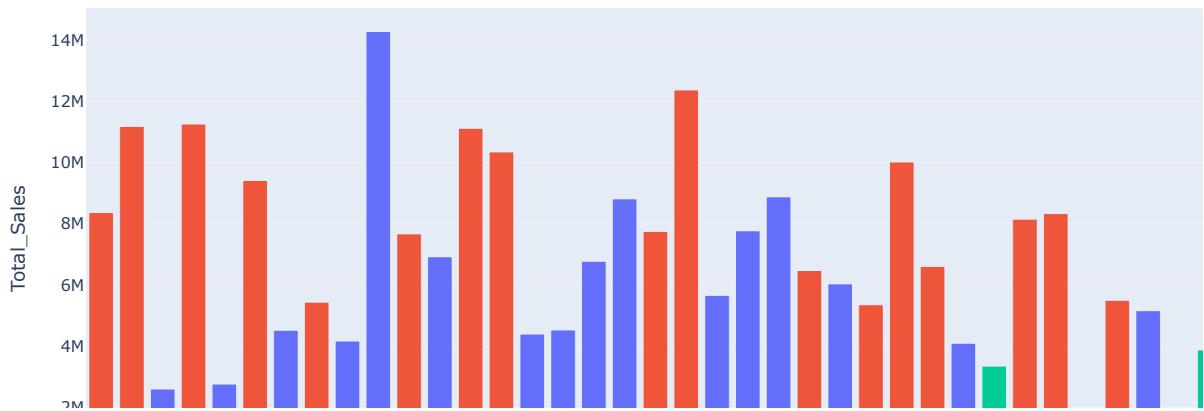
## Total Sales by Store

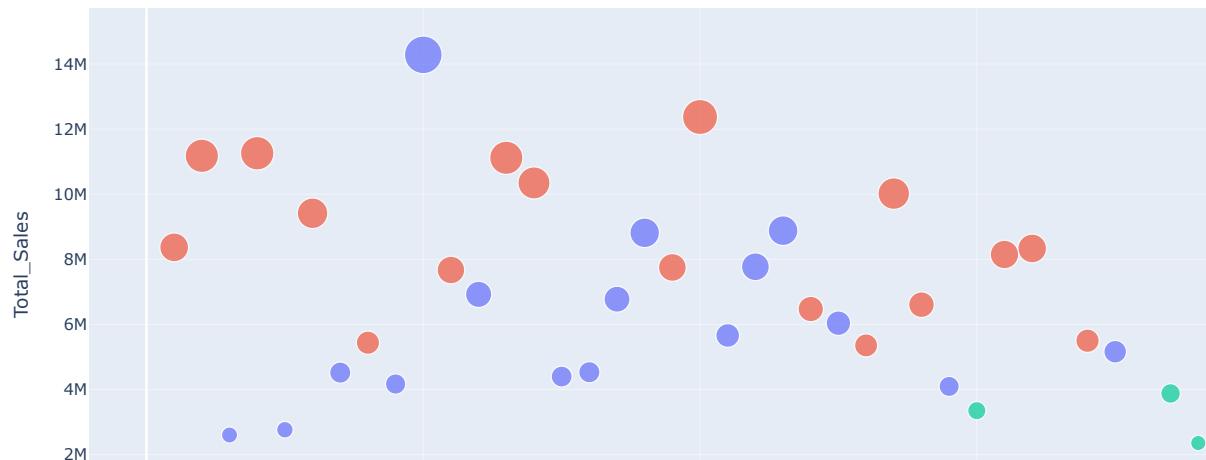


\*\*\*\*\*  
\*\*\*\*\*

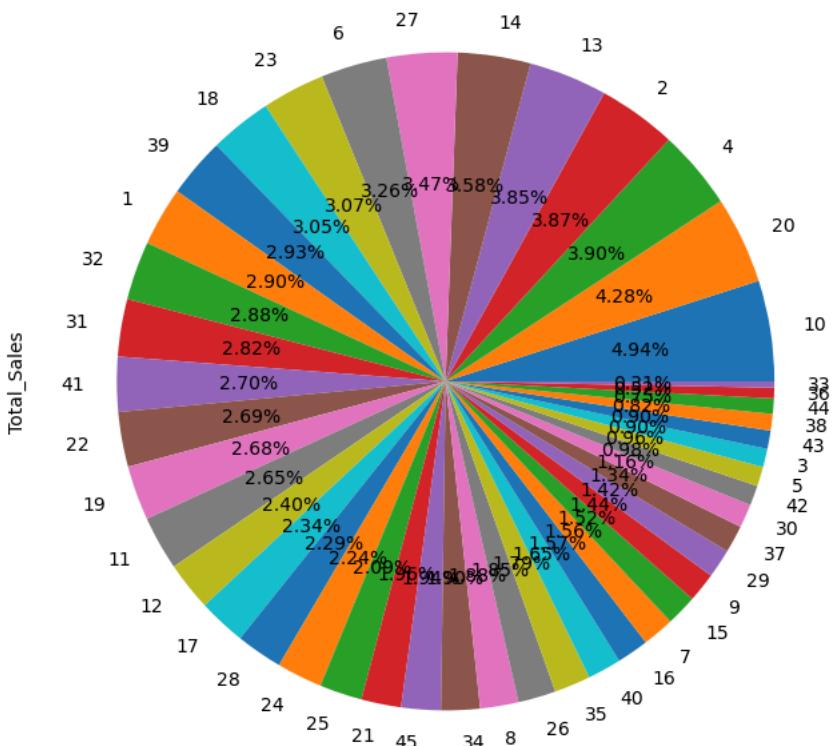
Dept: 40

## Total Sales by Store





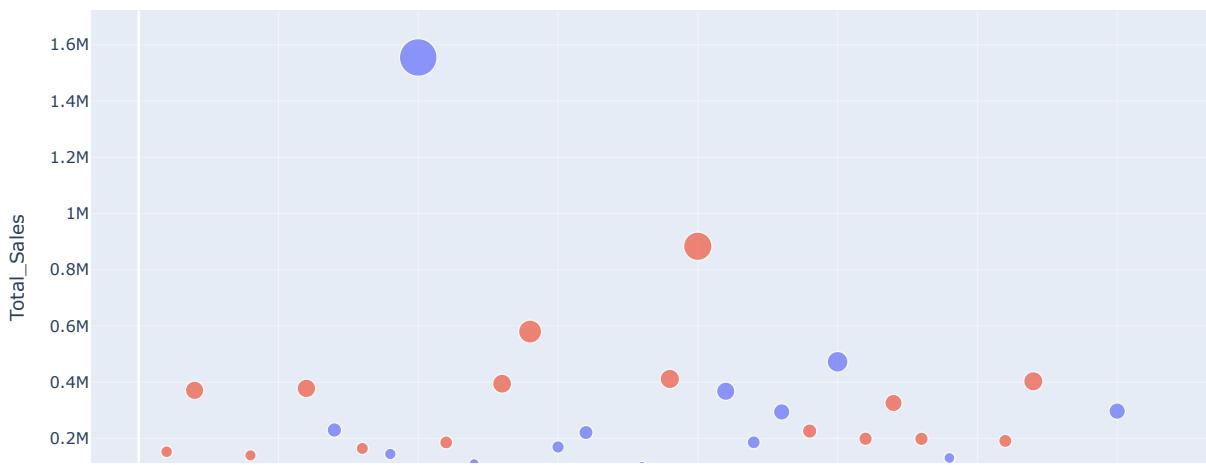
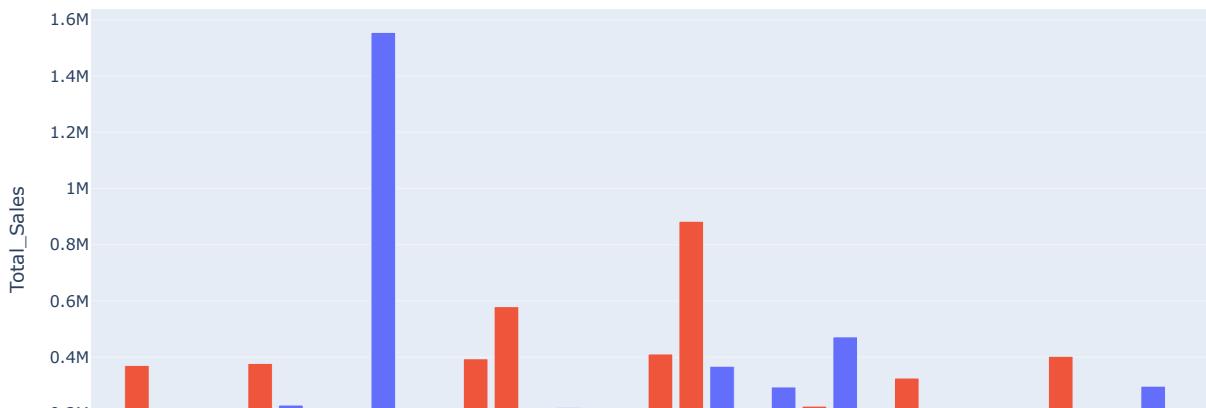
Total Sales by Store



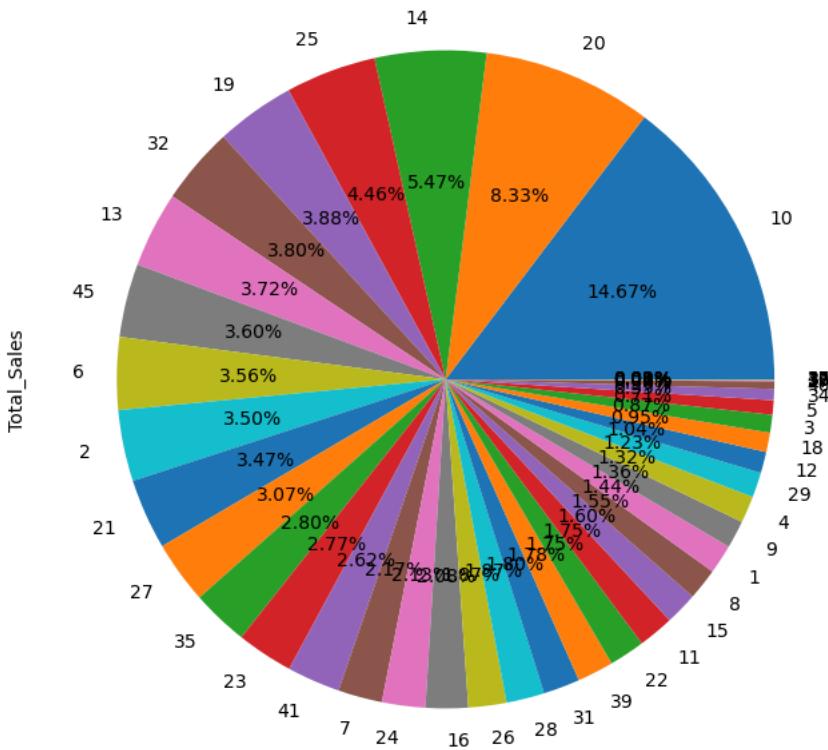
\*\*\*\*\*
\*\*\*\*\*  
\*\*\*\*\*

Dept: 41

## Total Sales by Store



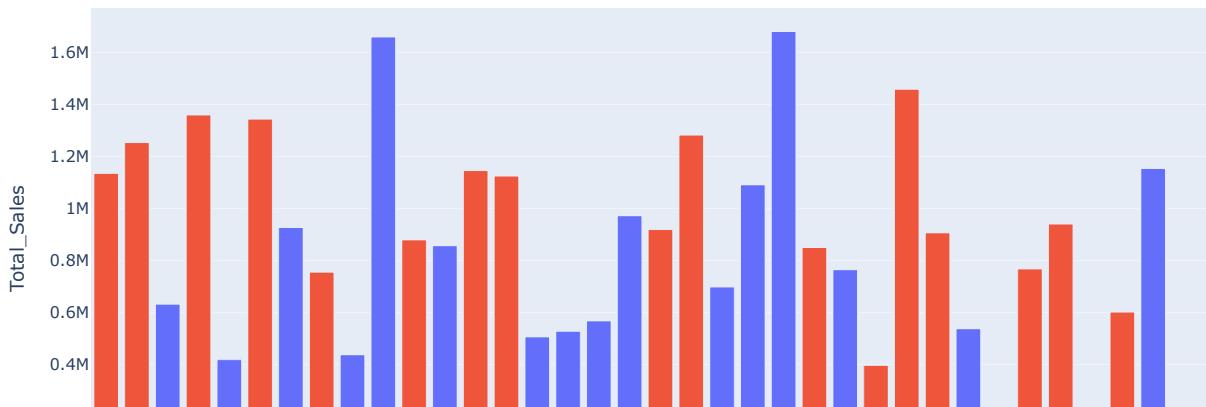
## Total Sales by Store

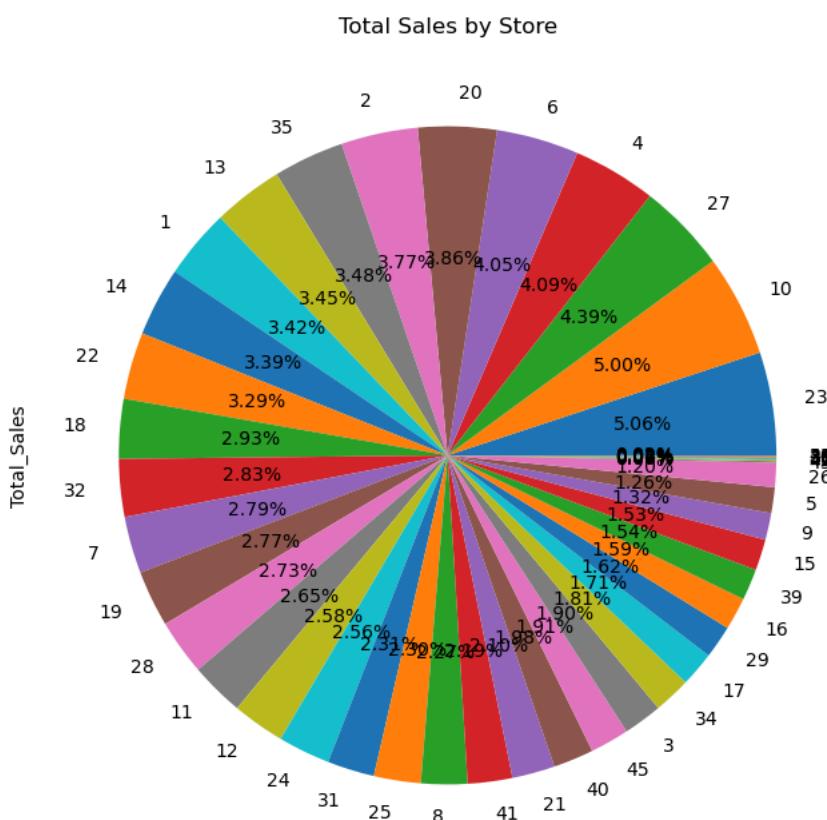
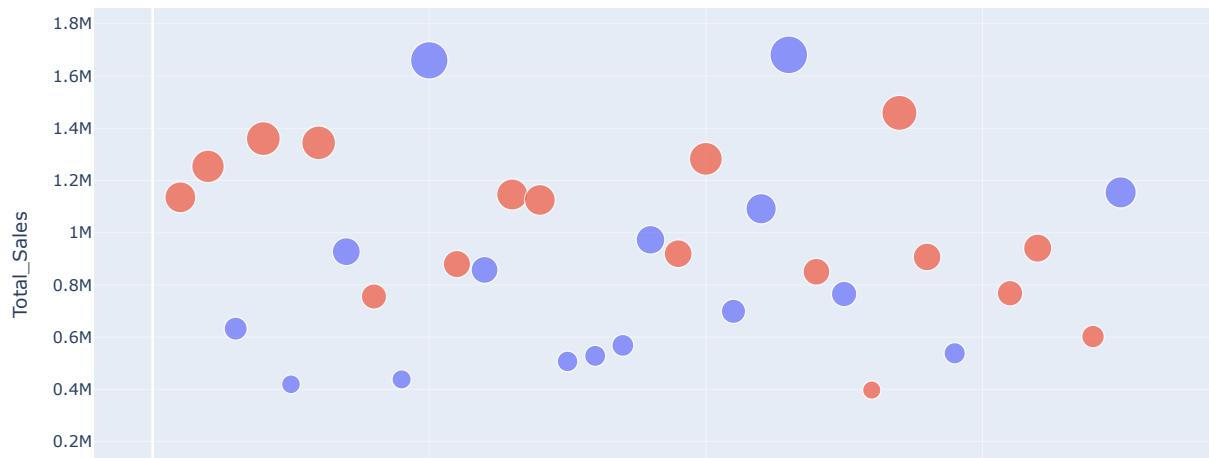


\*\*\*\*\*  
\*\*\*\*\*

Dept: 42

## Total Sales by Store

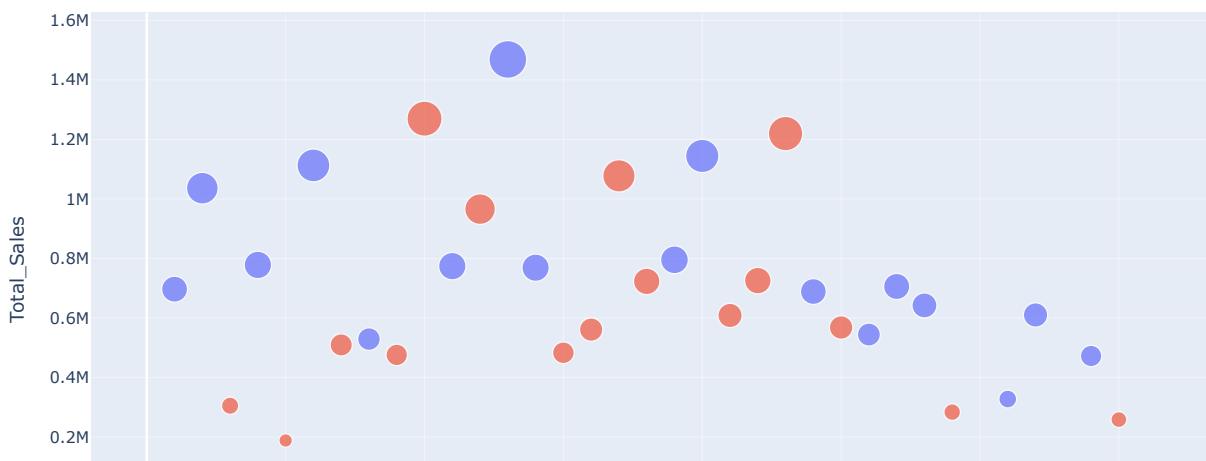
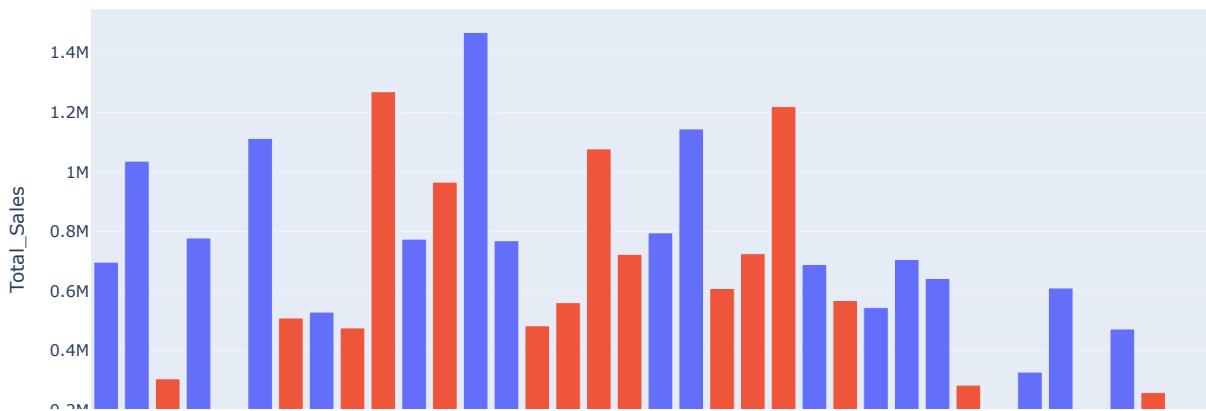




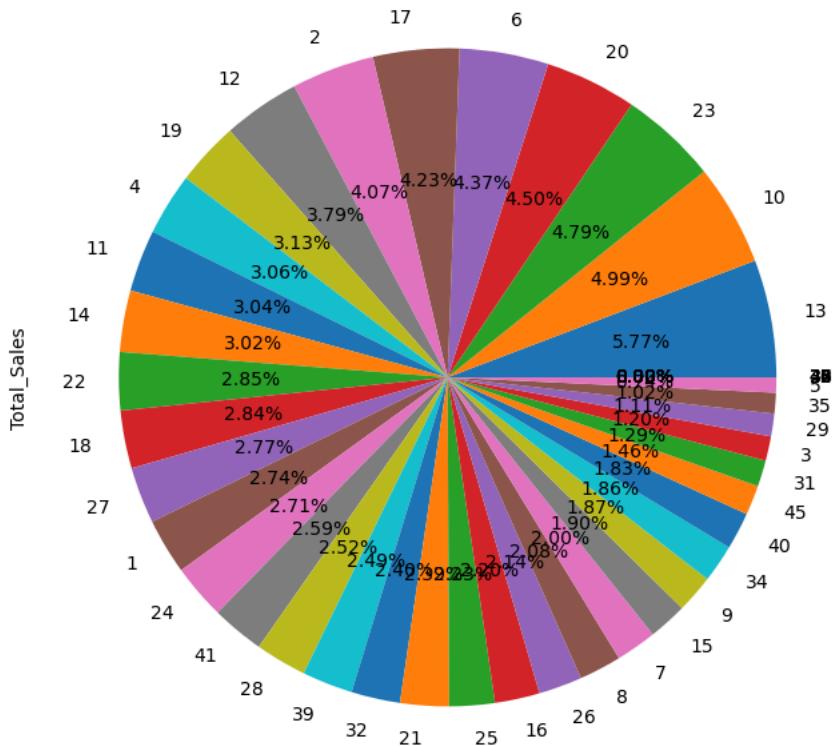
\*\*\*\*\*
\*\*\*\*\*

Dept: 44

## Total Sales by Store



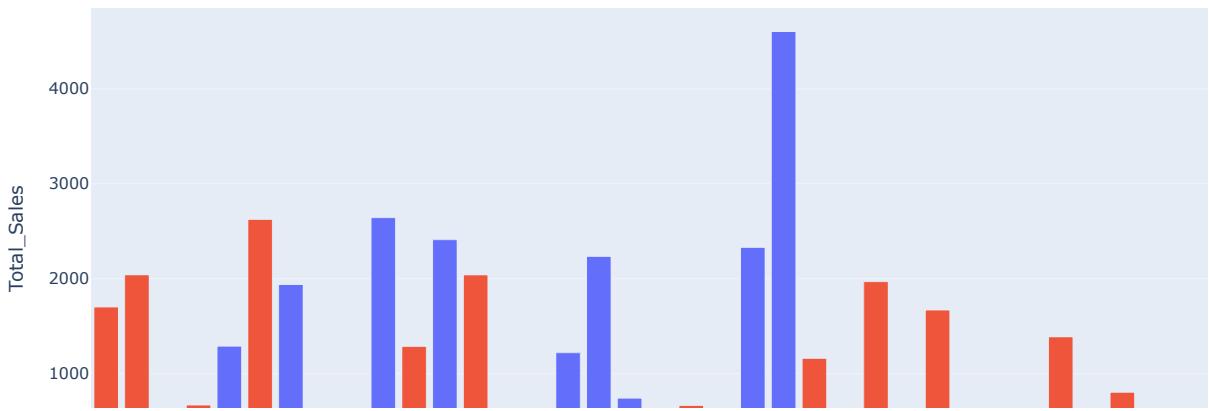
## Total Sales by Store



\*\*\*\*\*
\*\*\*\*\*

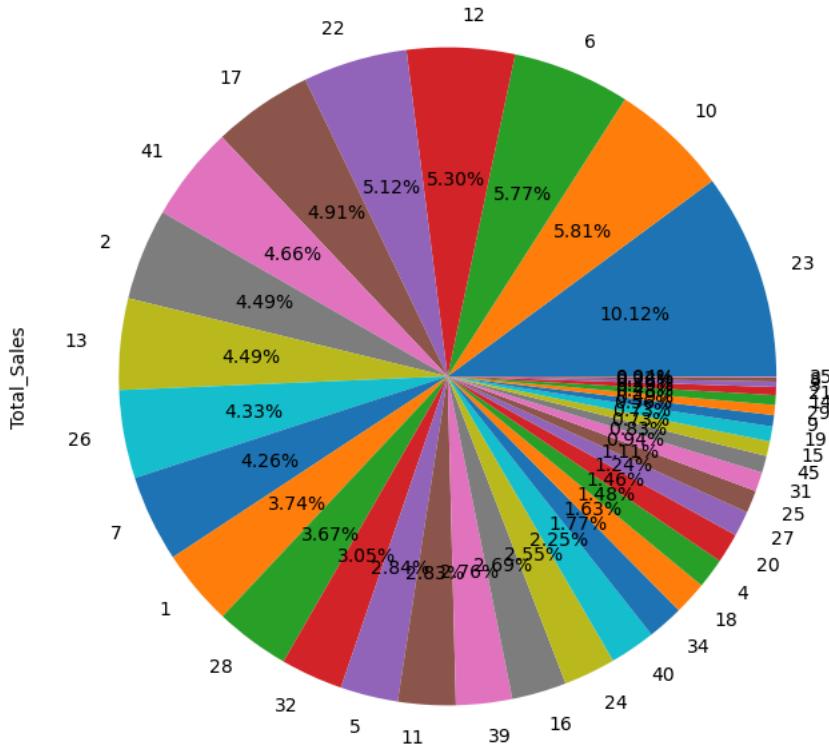
Dept: 45

## Total Sales by Store





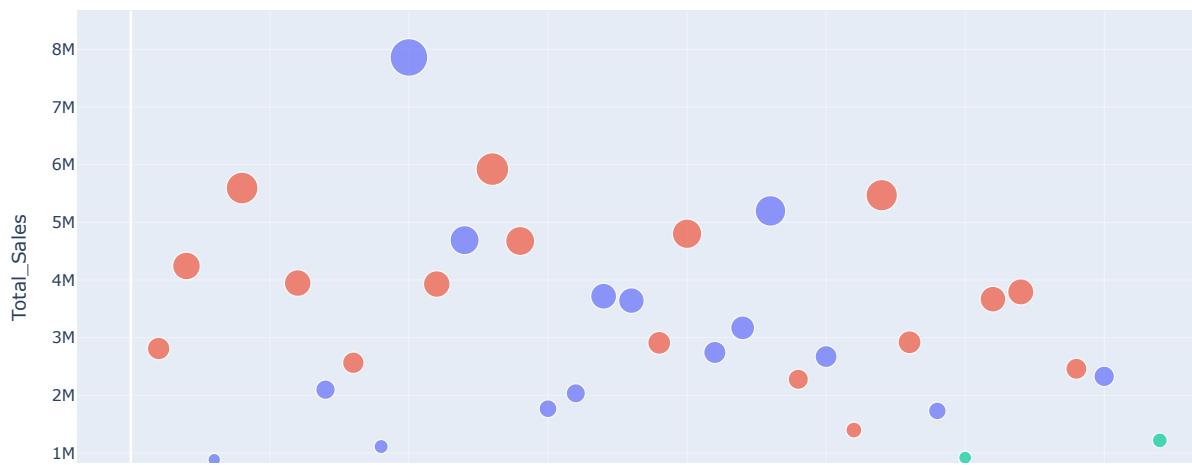
Total Sales by Store



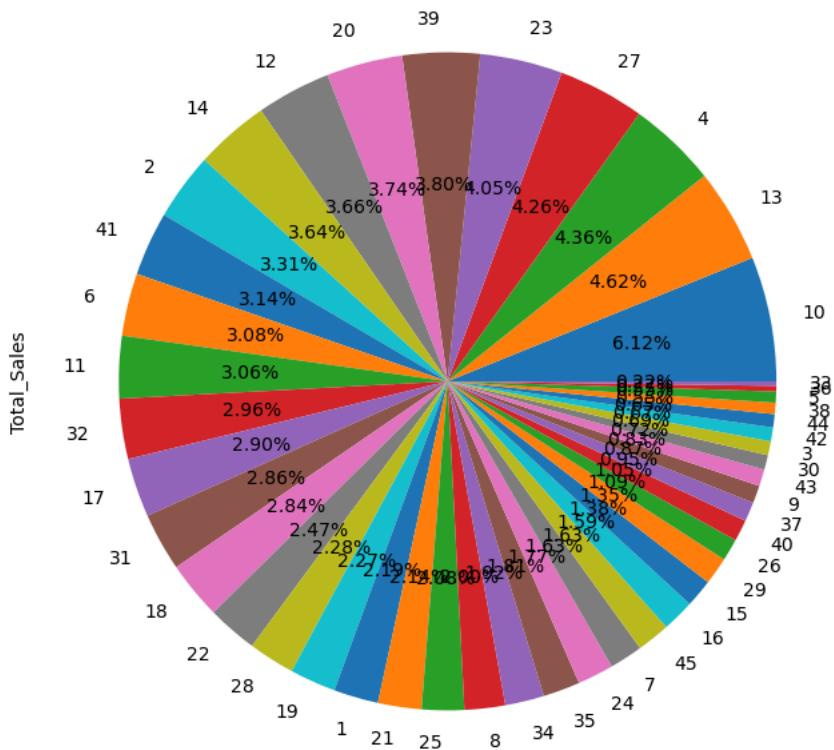
\*\*\*\*\*
\*\*\*\*\*

Dept: 46

## Total Sales by Store



## Total Sales by Store



\*\*\*\*\*  
\*\*\*\*\*

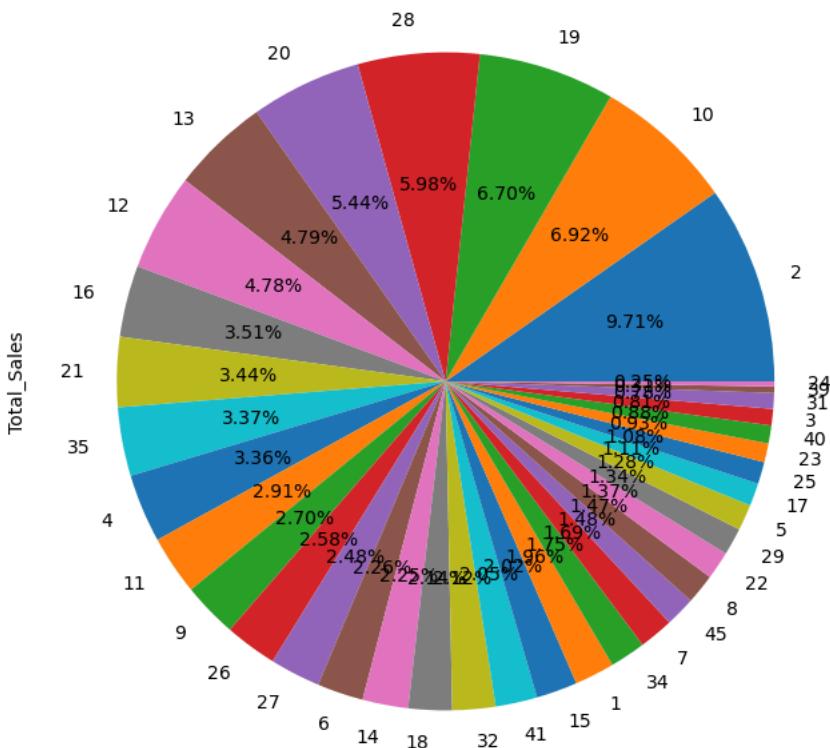
Dept: 47

## Total Sales by Store





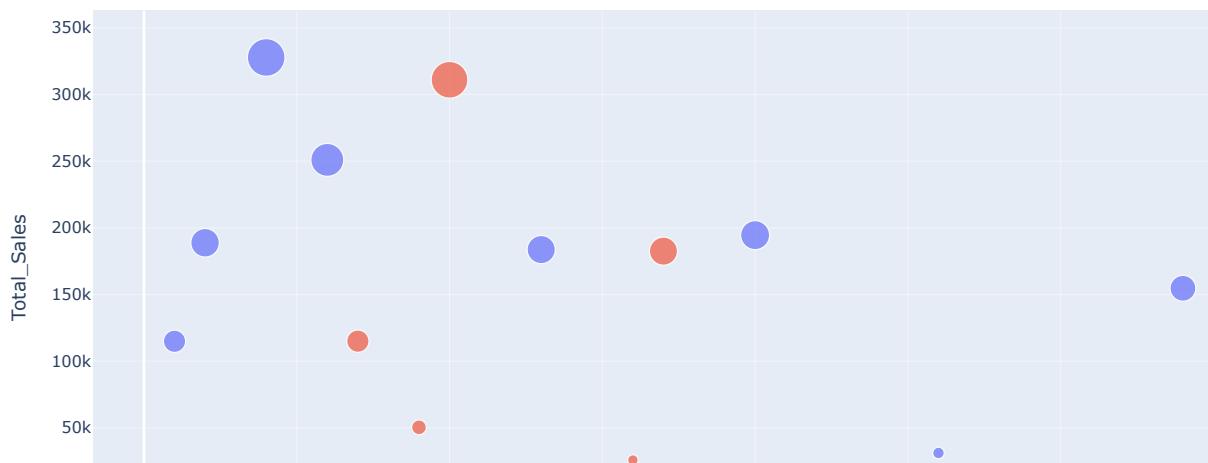
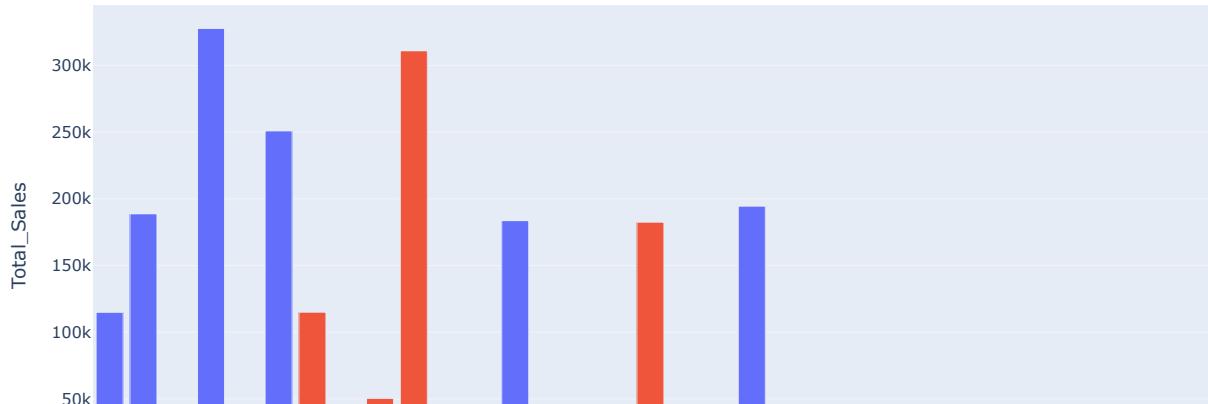
Total Sales by Store



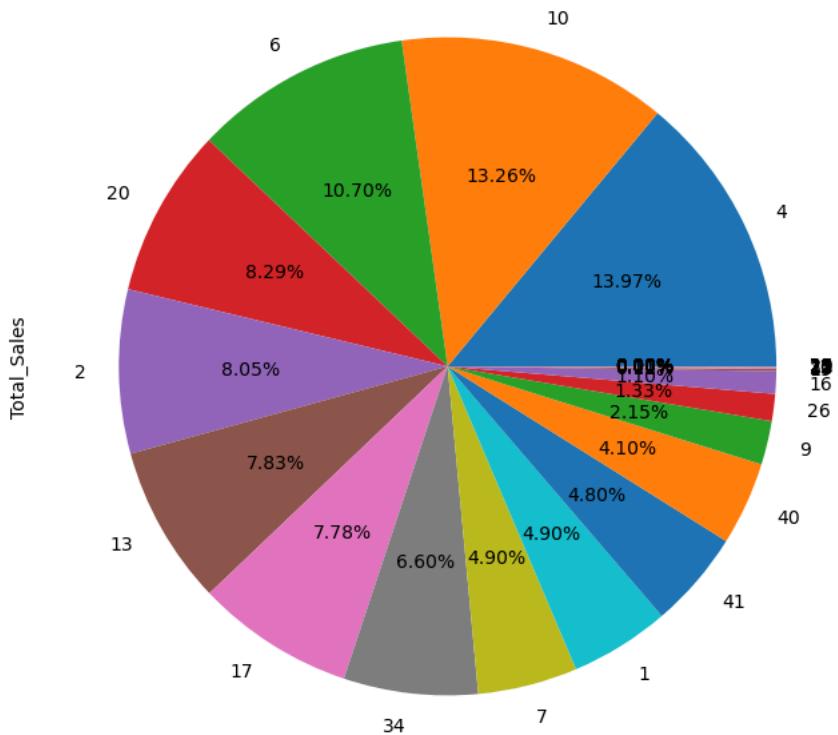
\*\*\*\*\*
\*\*\*\*\*  
\*\*\*\*\*

Dept: 48

## Total Sales by Store



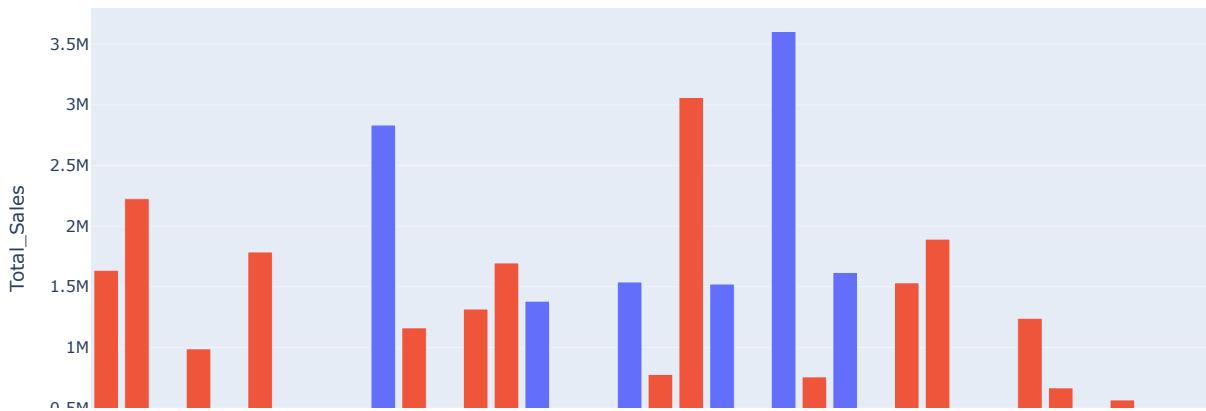
## Total Sales by Store



\*\*\*\*\*  
\*\*\*\*\*

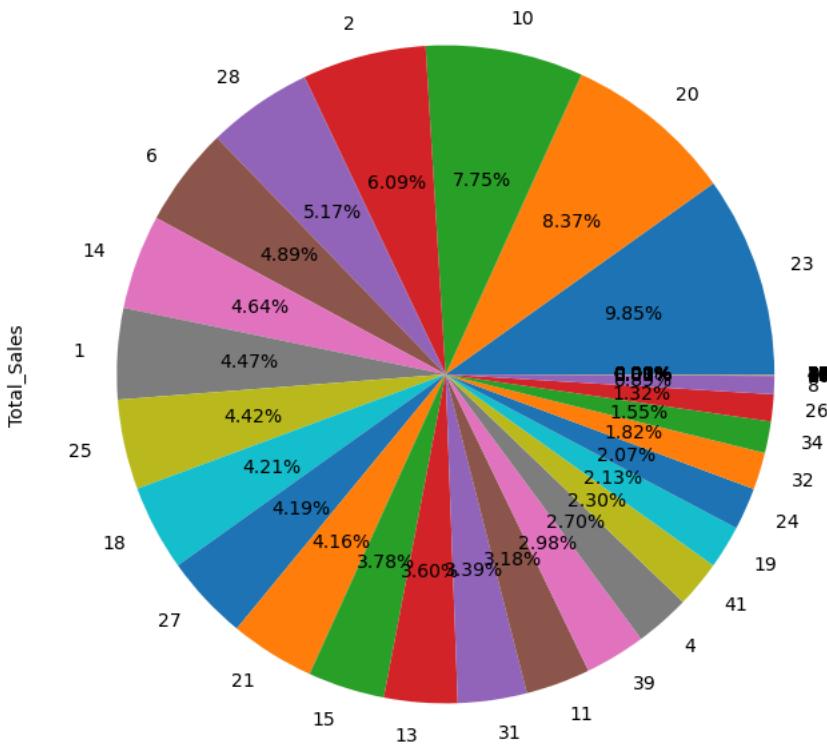
Dept: 49

## Total Sales by Store





Total Sales by Store



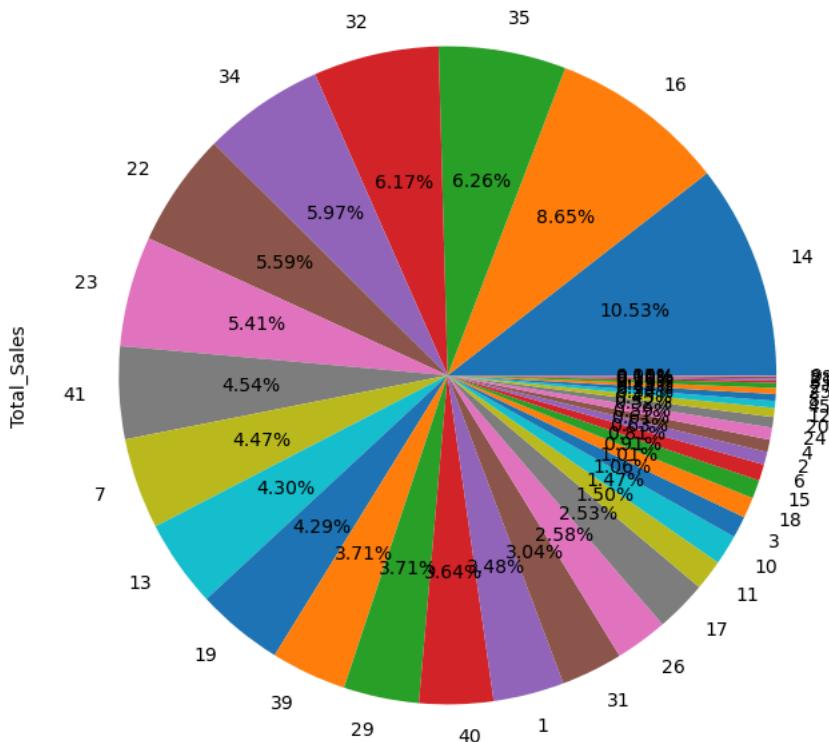
\*\*\*\*\*
\*\*\*\*\*  
\*\*\*\*\*

Dept: 51

## Total Sales by Store



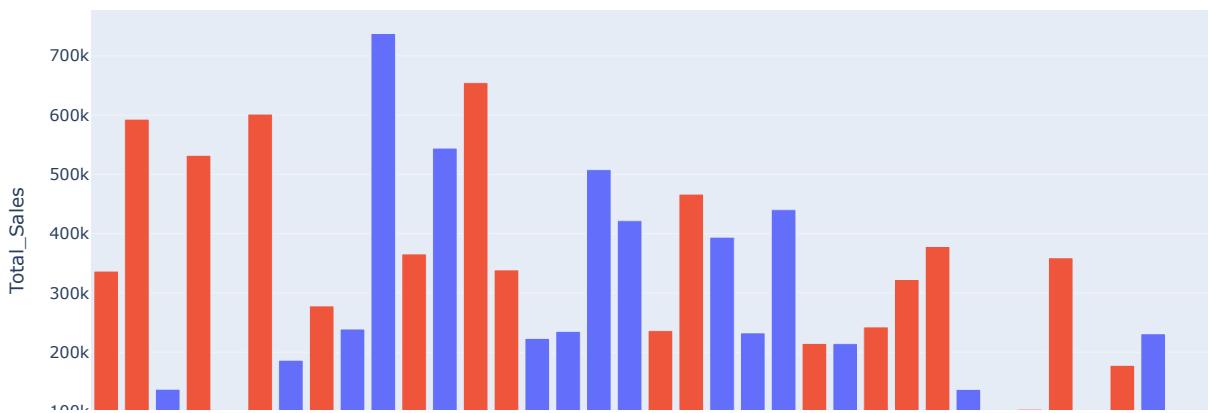
## Total Sales by Store

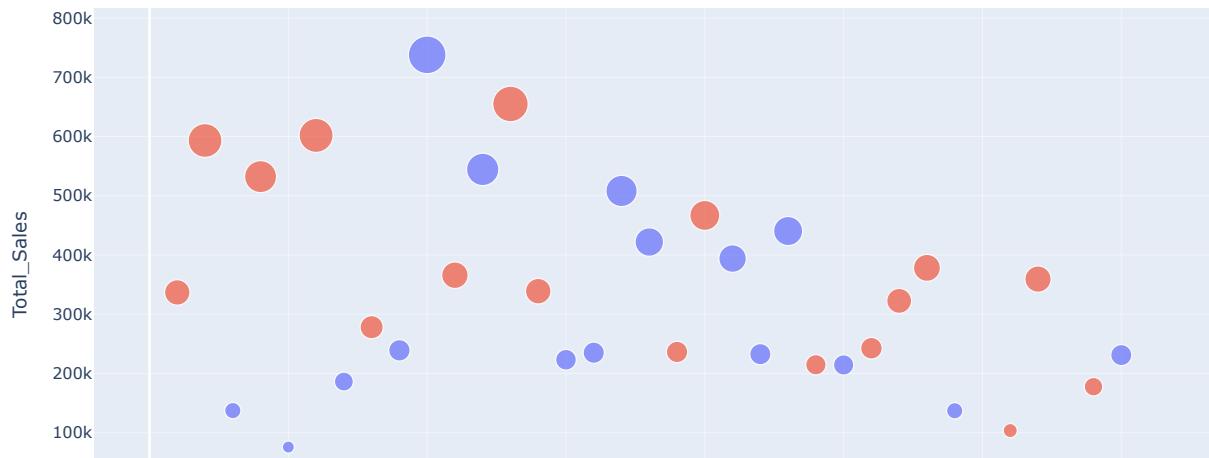


\*\*\*\*\*
\*\*\*\*\*

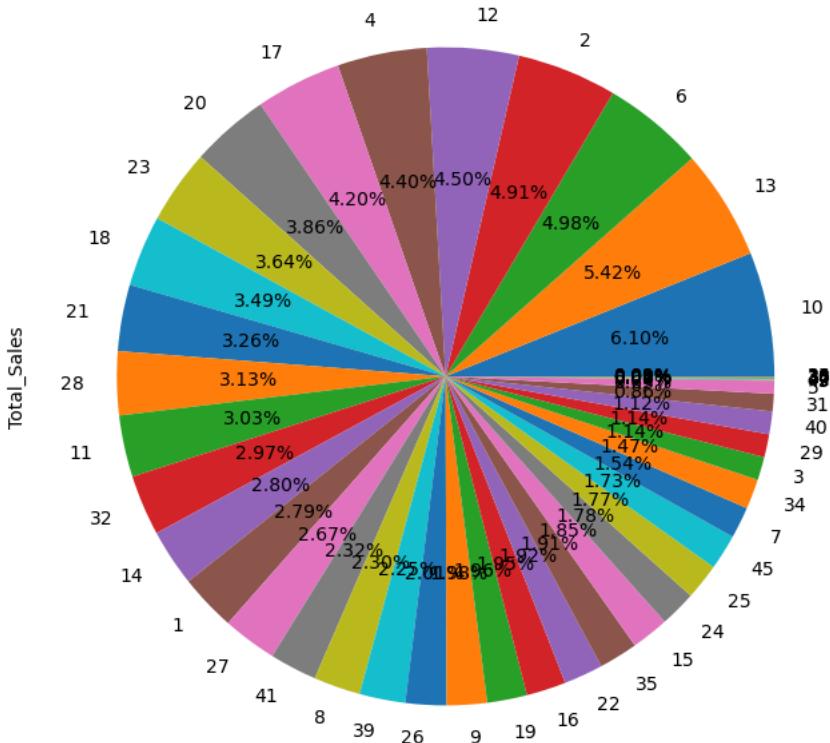
Dept: 52

## Total Sales by Store





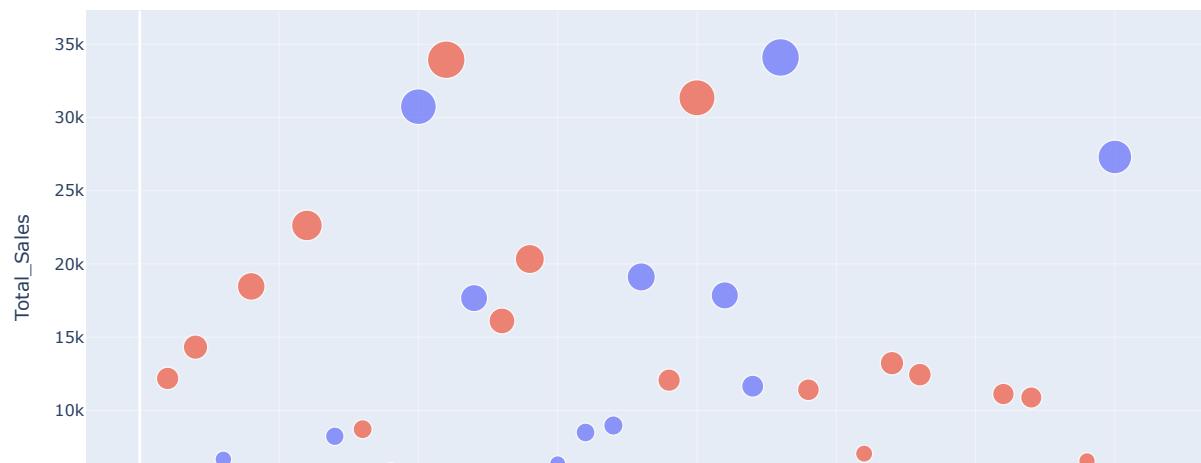
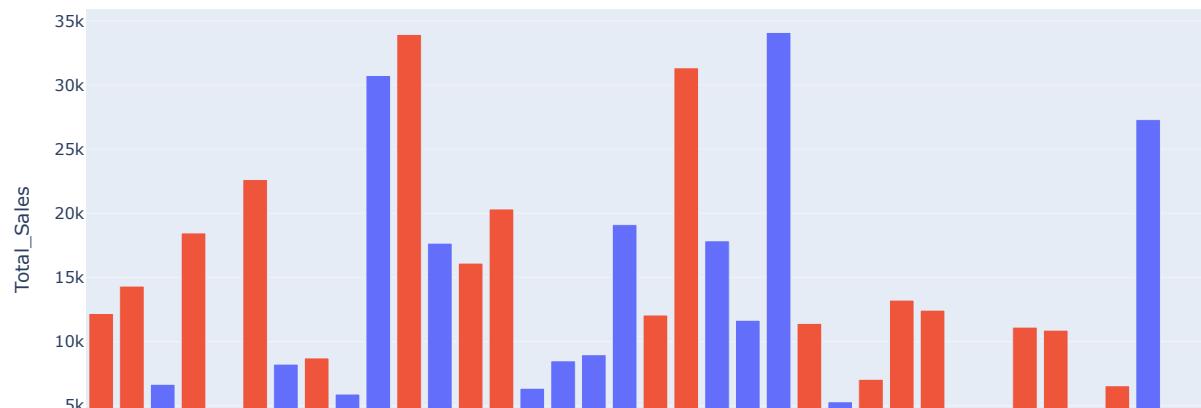
Total Sales by Store



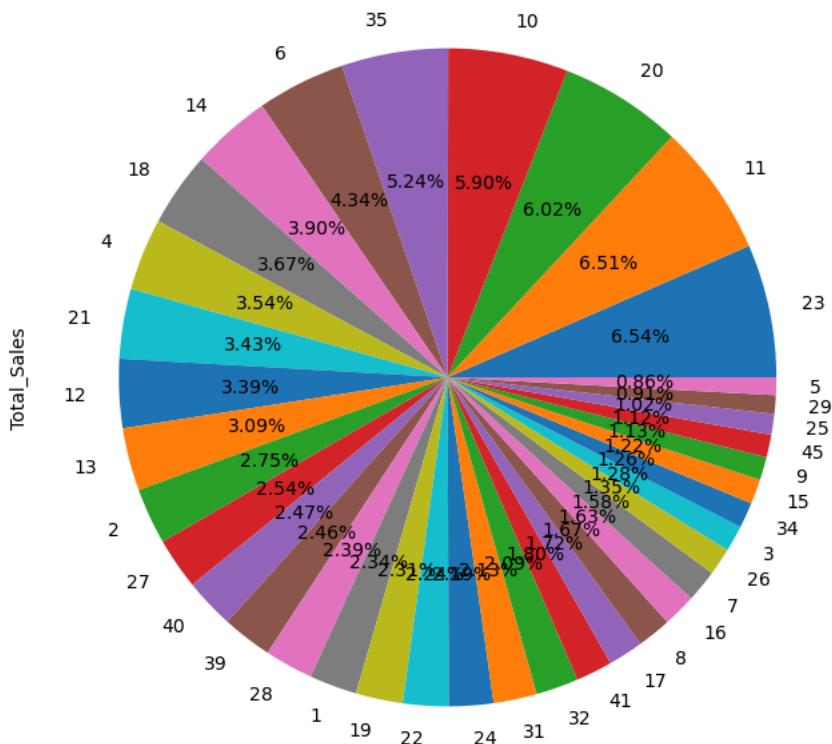
```
*****
*****
```

Dept: 54

## Total Sales by Store



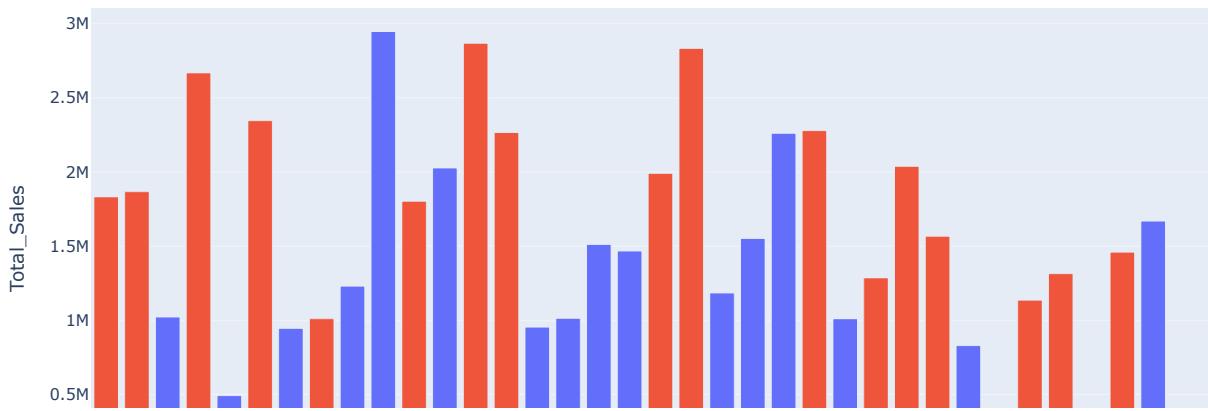
## Total Sales by Store

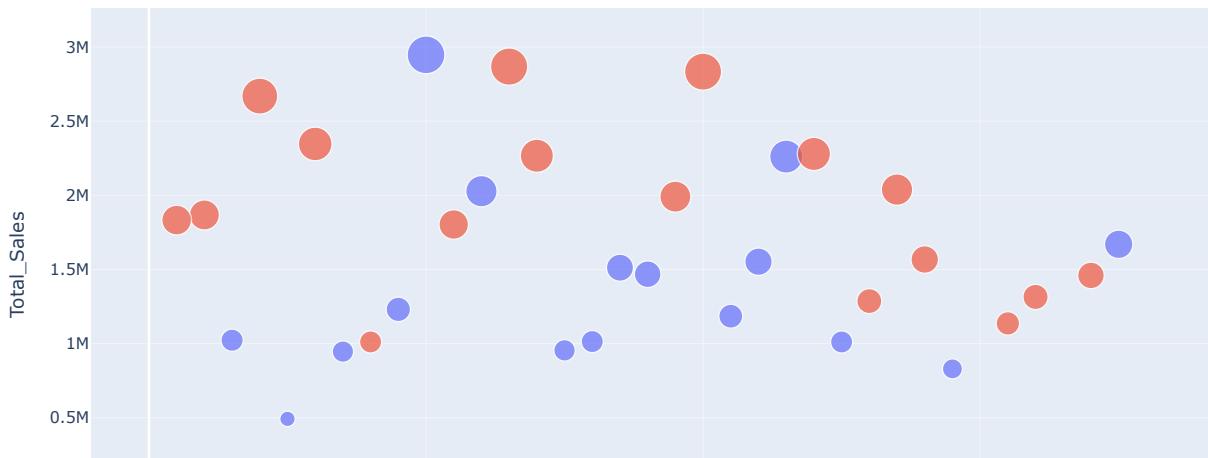


\*\*\*\*\*
\*\*\*\*\*

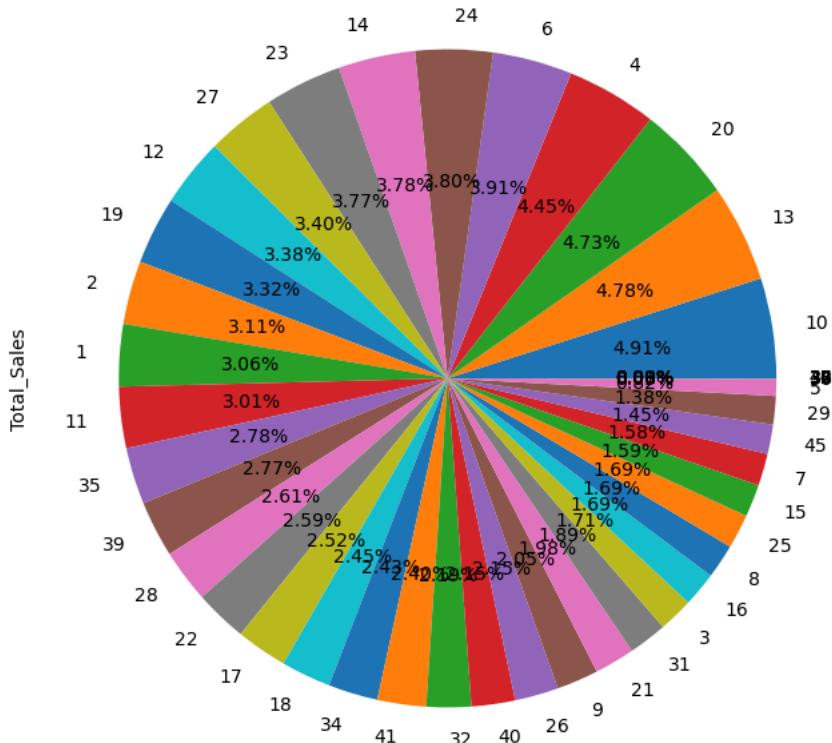
Dept: 55

## Total Sales by Store





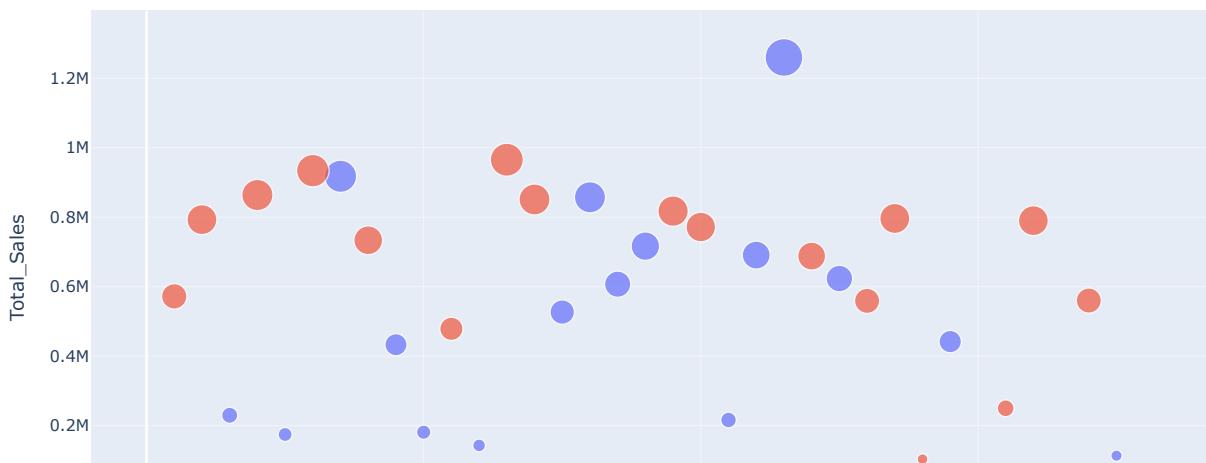
Total Sales by Store



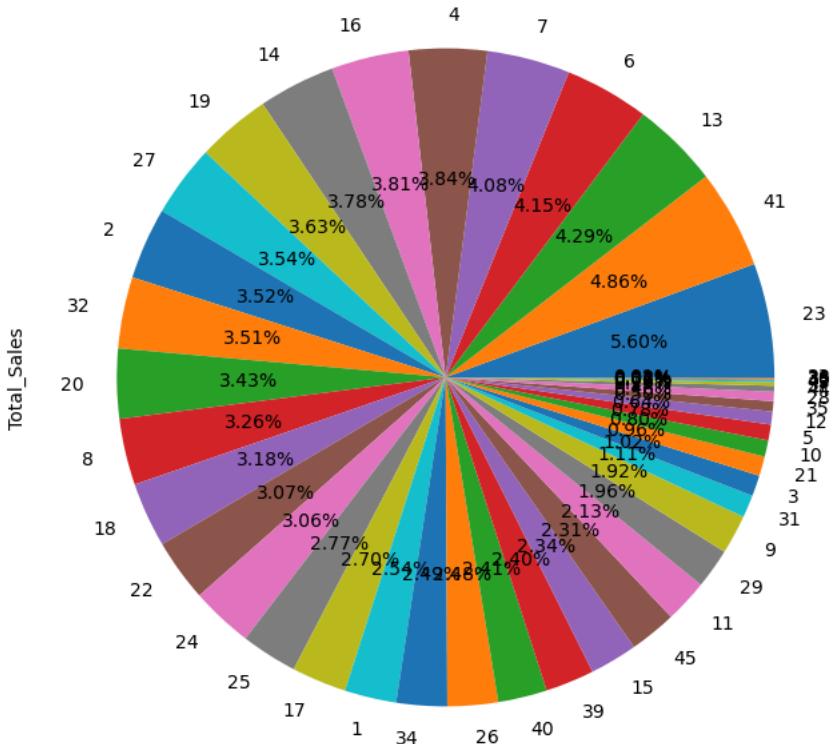
```
*****
*****
```

Dept: 56

## Total Sales by Store



## Total Sales by Store

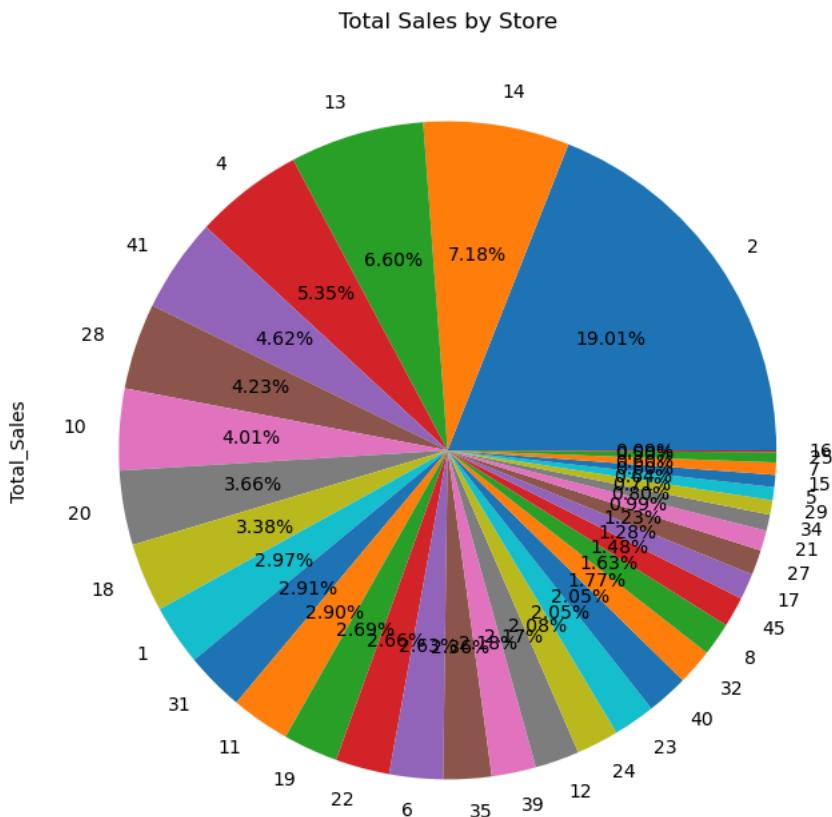
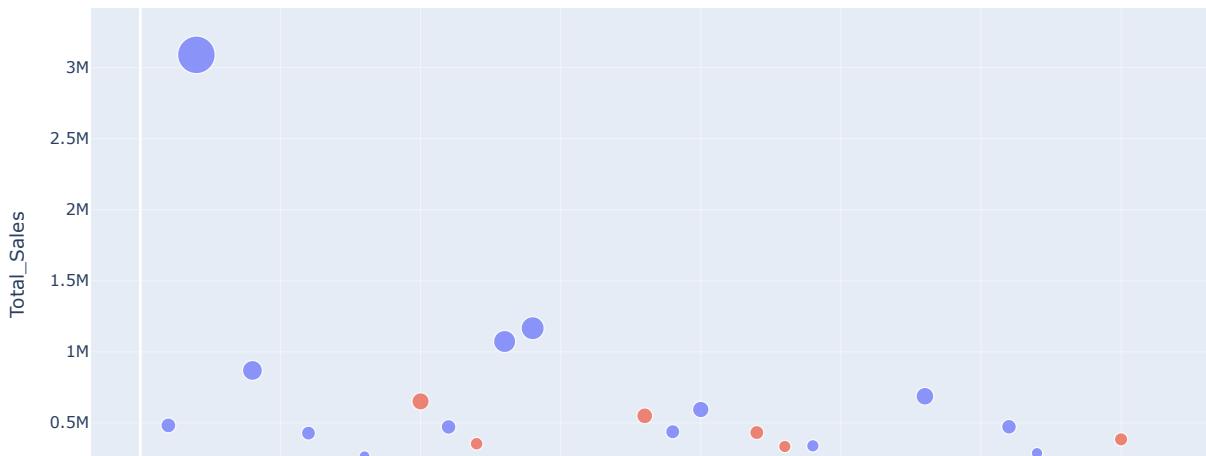


\*\*\*\*\*
\*\*\*\*\*

Dept: 58

## Total Sales by Store

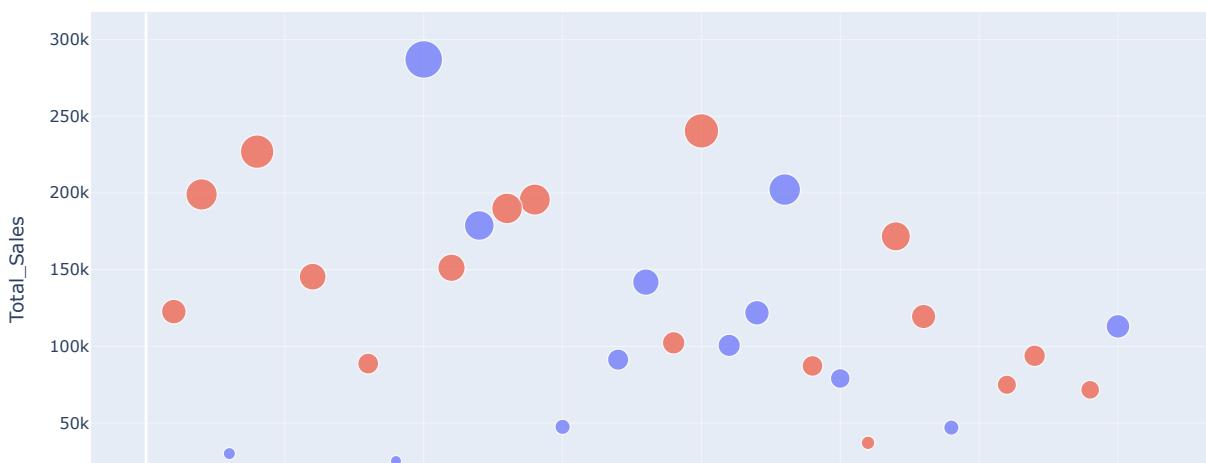
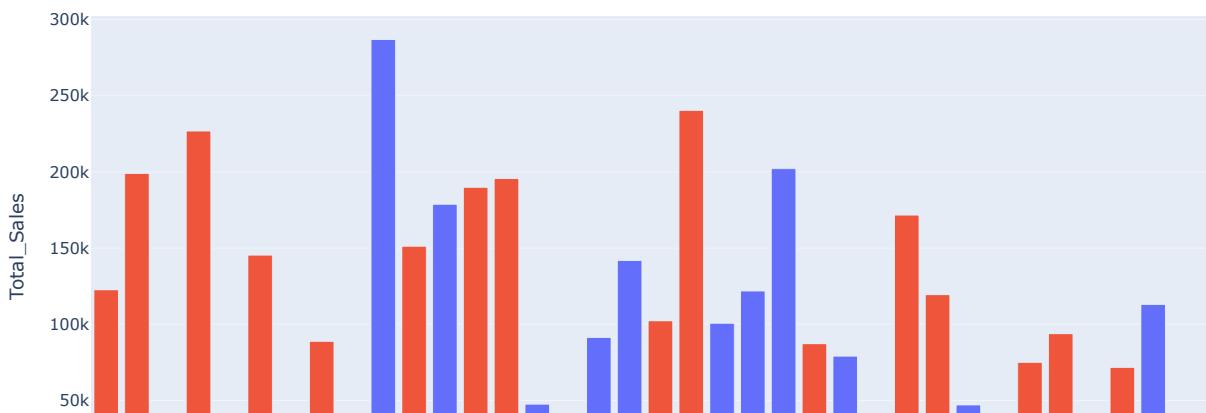




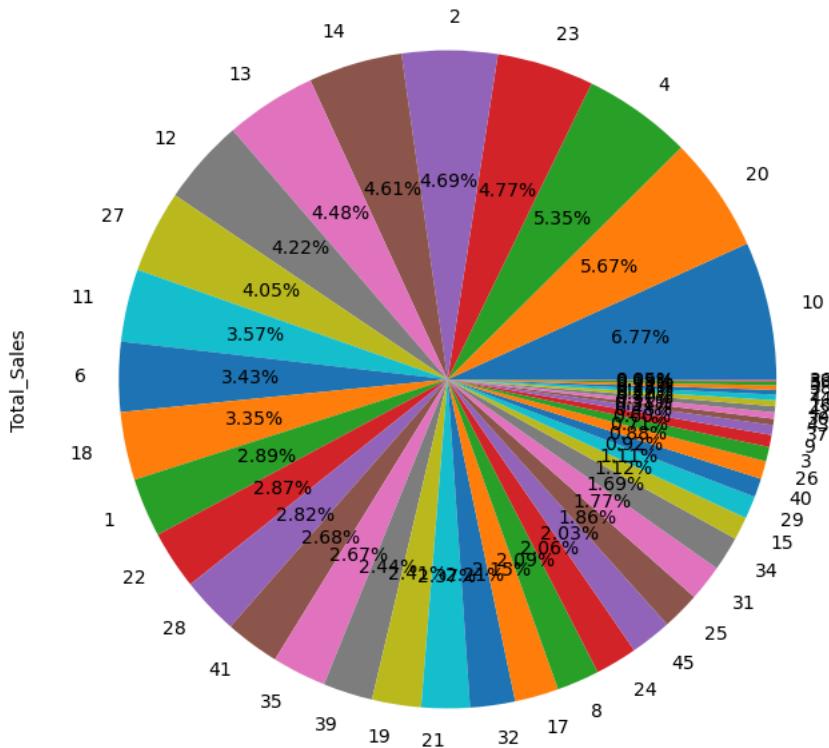
```
*****
*****
```

Dept: 59

## Total Sales by Store



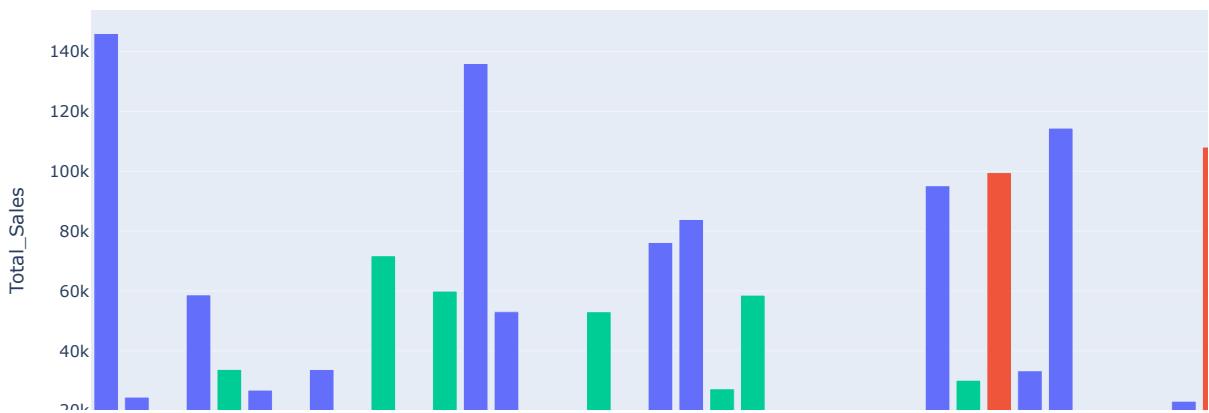
## Total Sales by Store



\*\*\*\*\*
\*\*\*\*\*

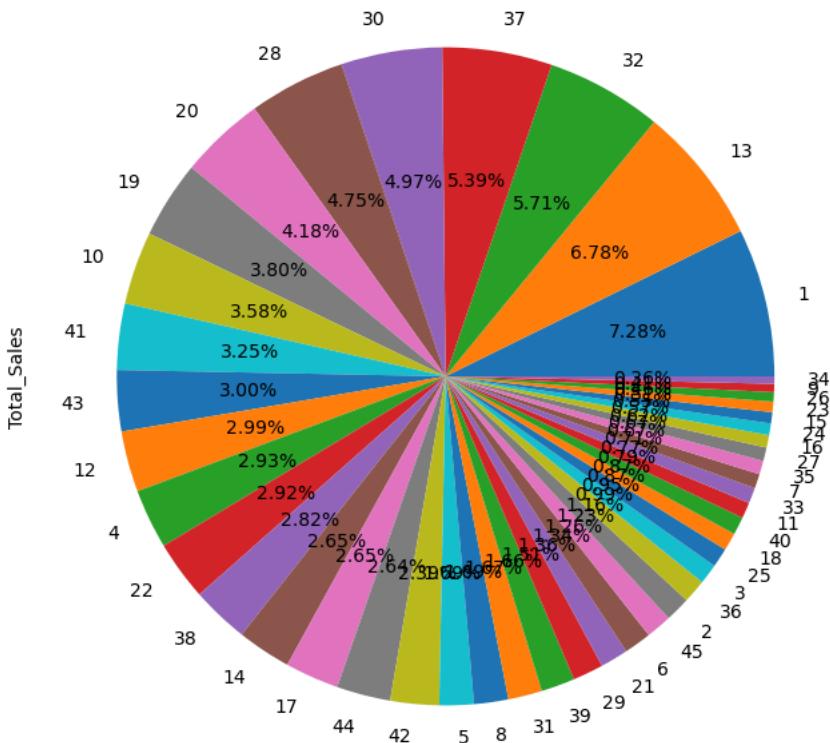
Dept: 60

## Total Sales by Store





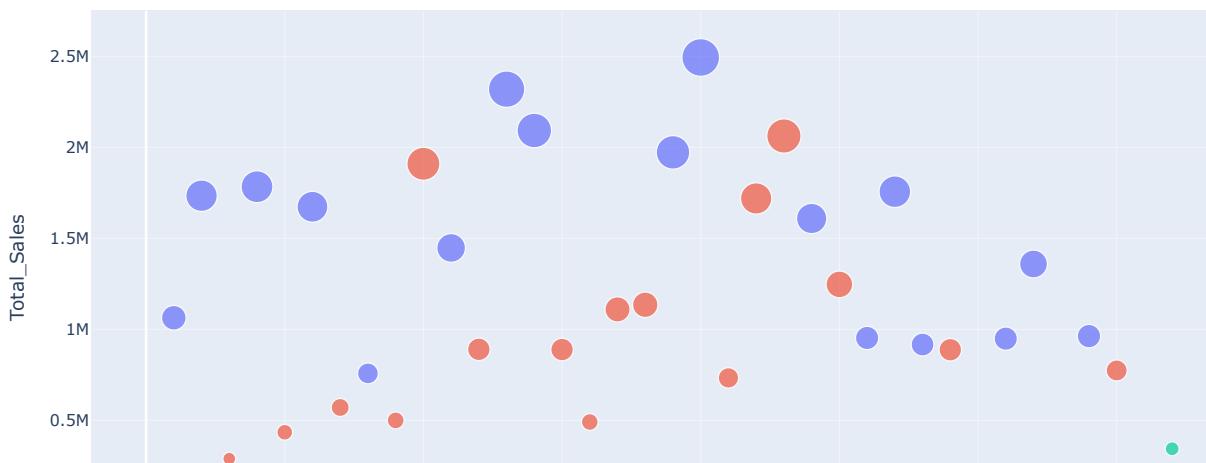
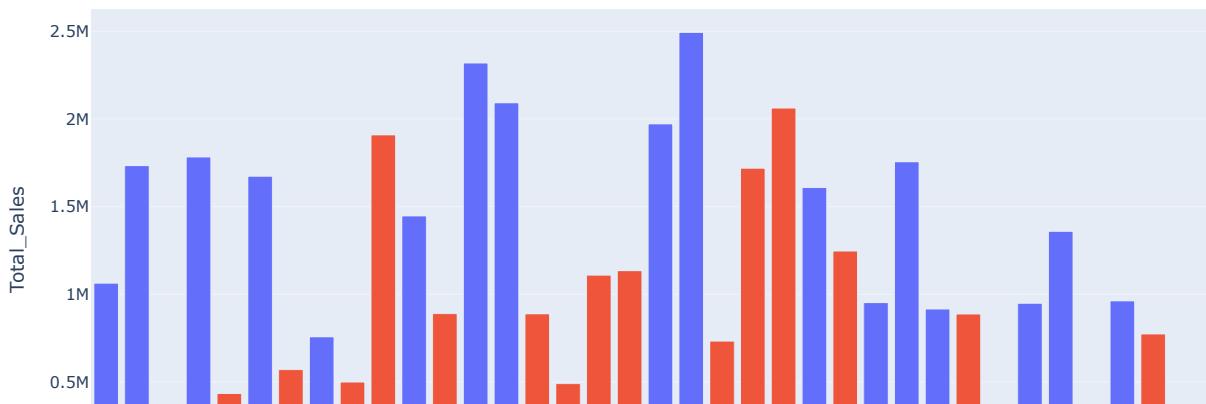
Total Sales by Store



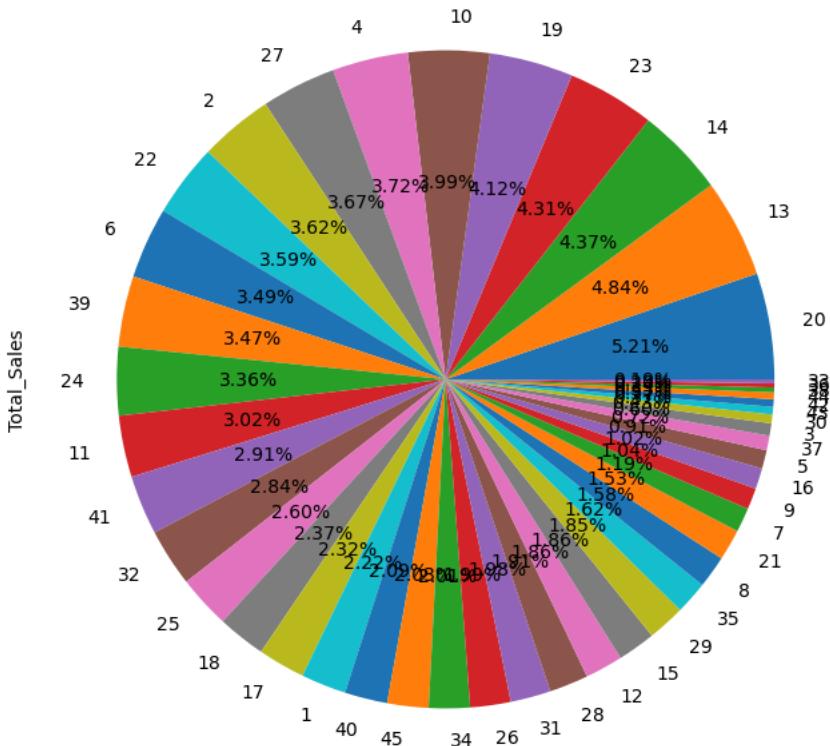
```
*****
*****
```

Dept: 67

## Total Sales by Store



## Total Sales by Store

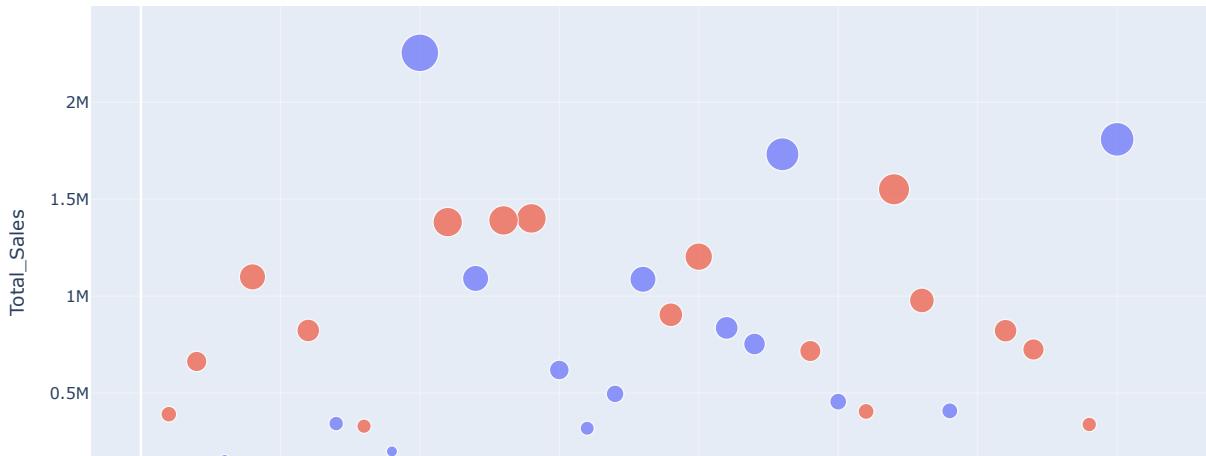


\*\*\*\*\*
\*\*\*\*\*

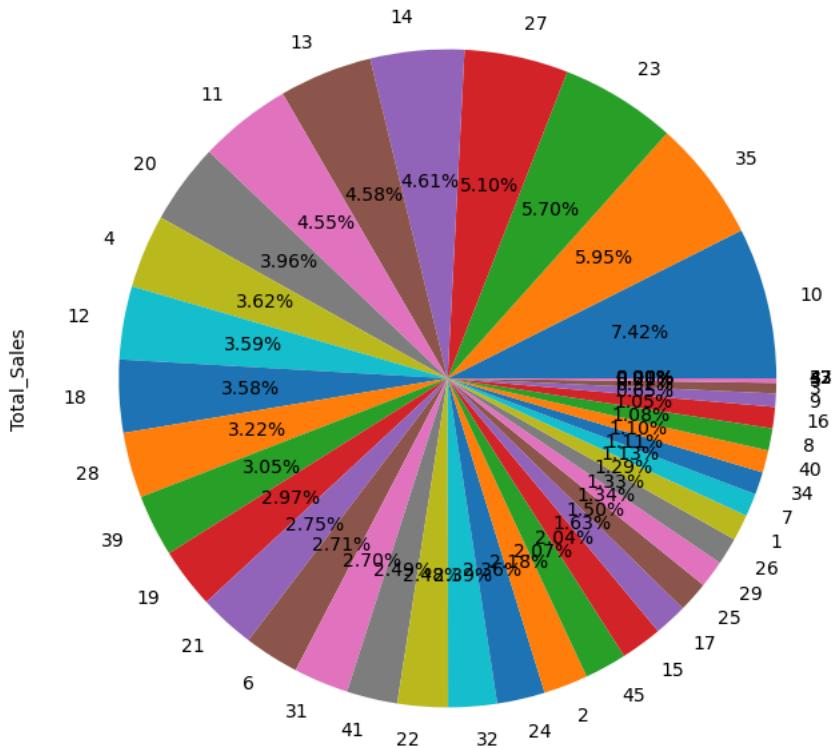
Dept: 71

## Total Sales by Store





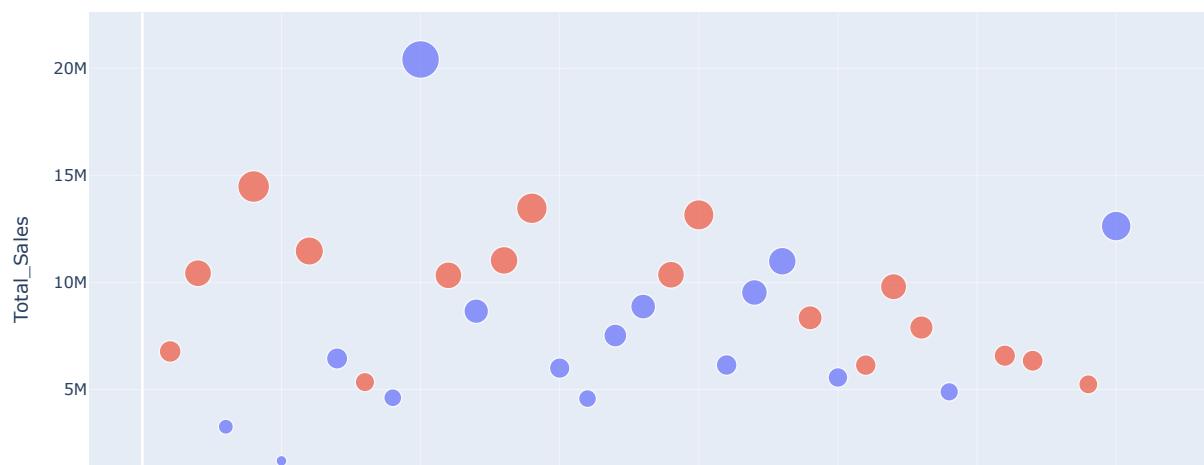
Total Sales by Store



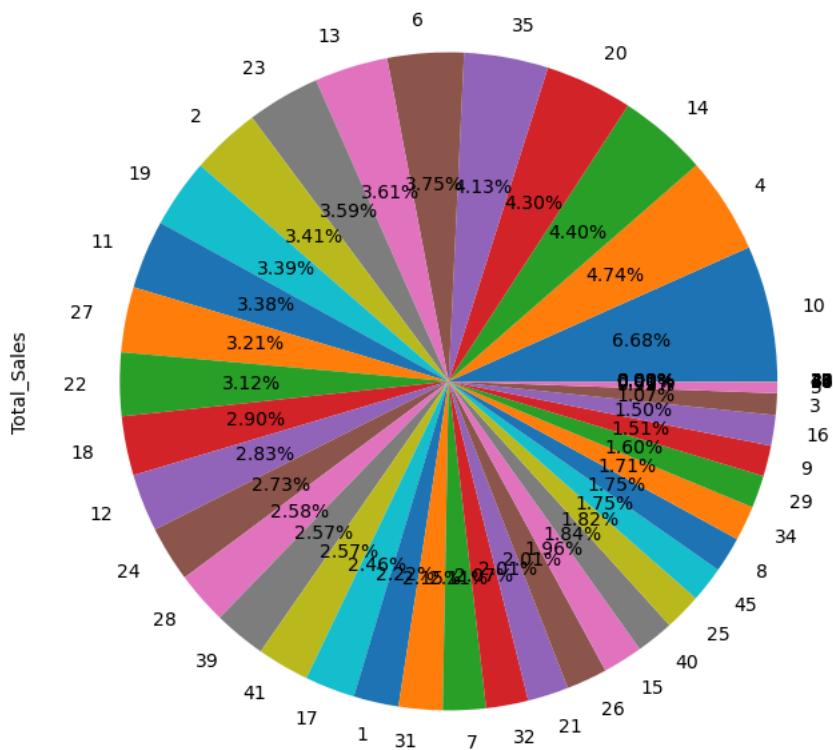
\*\*\*\*\*
\*\*\*\*\*  
\*\*\*\*\*

Dept: 72

## Total Sales by Store



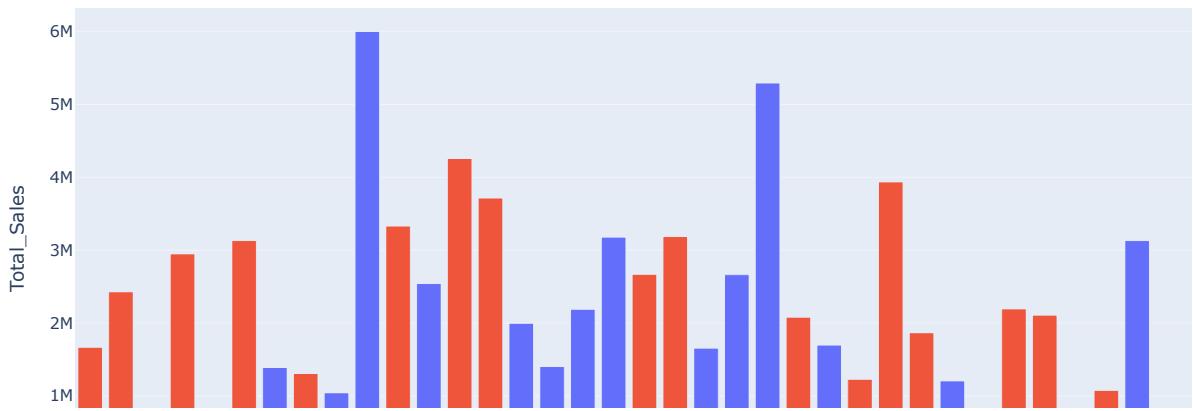
## Total Sales by Store

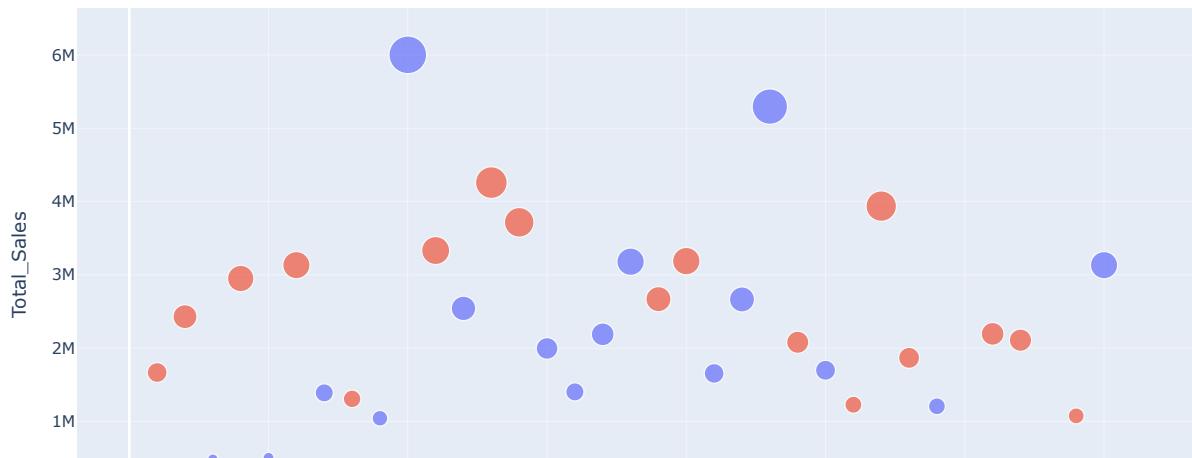


\*\*\*\*\*
\*\*\*\*\*

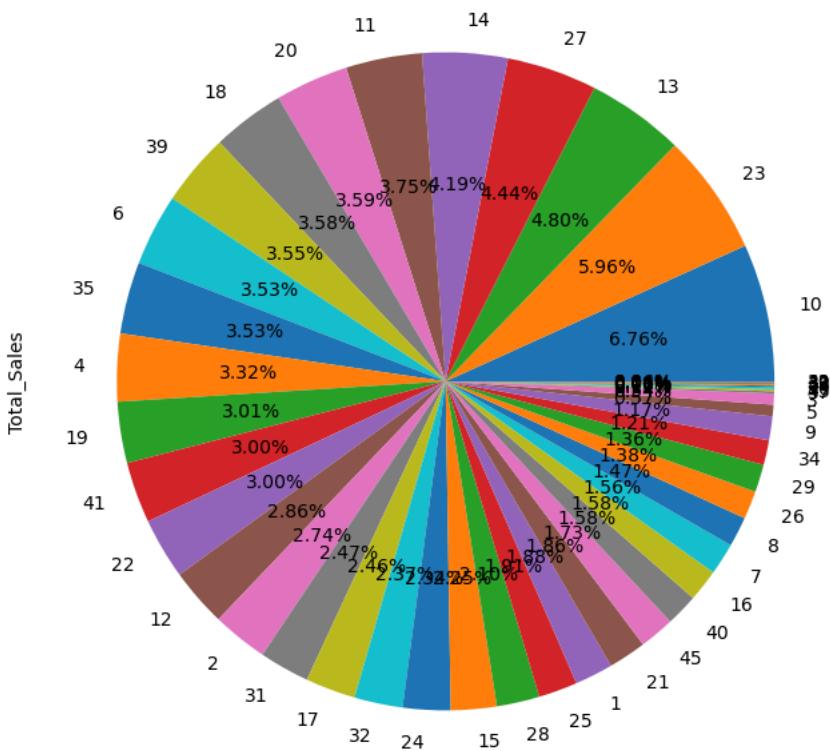
Dept: 74

## Total Sales by Store





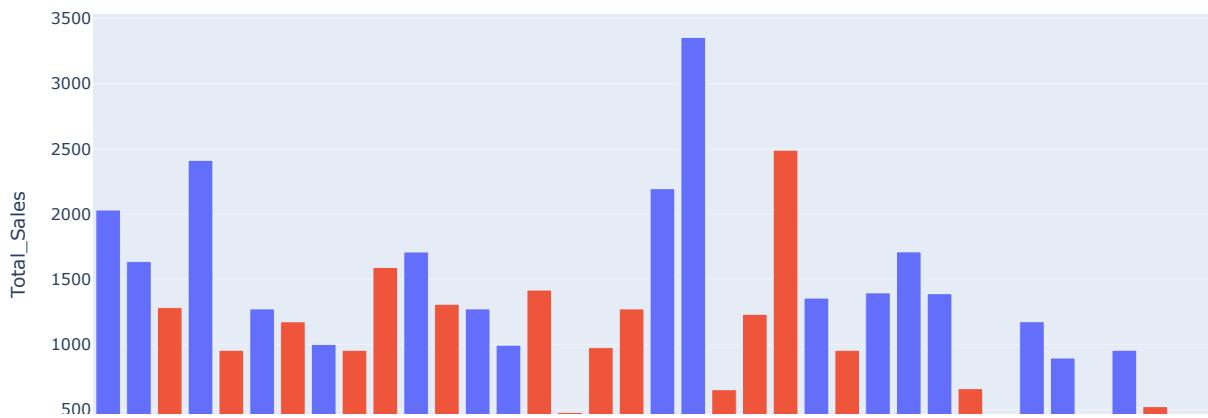
Total Sales by Store



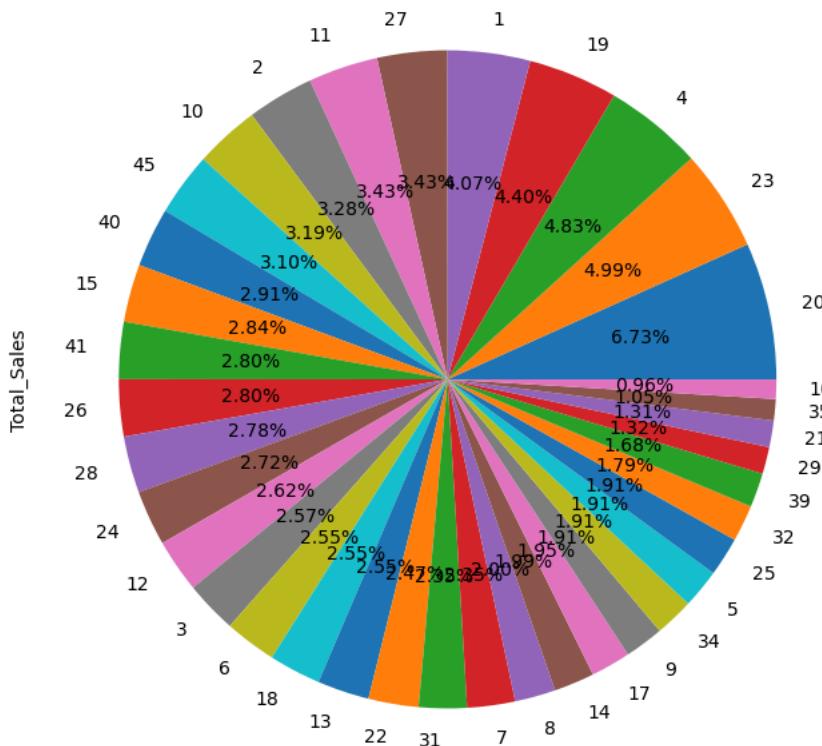
\*\*\*\*\*
\*\*\*\*\*  
\*\*\*\*\*

Dept: 77

## Total Sales by Store



## Total Sales by Store

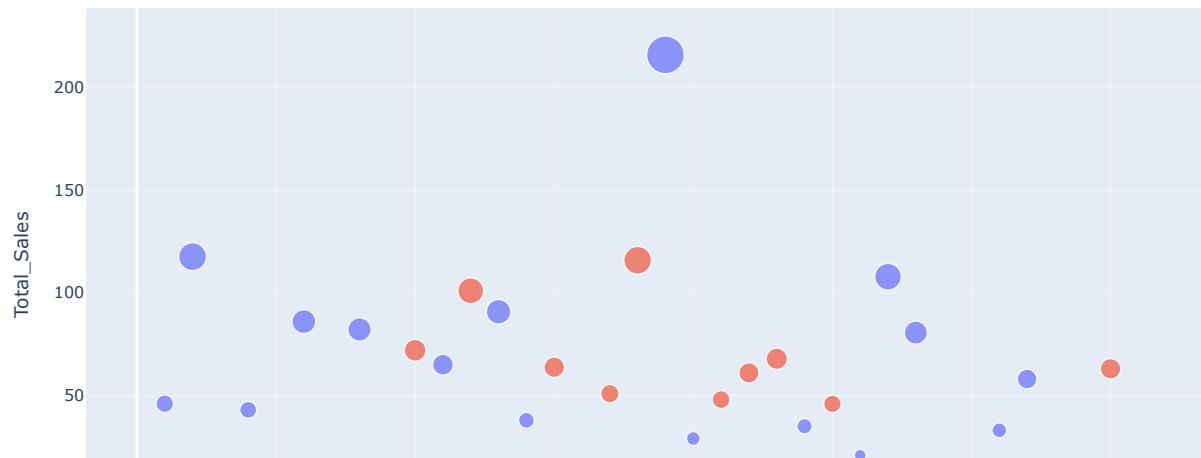


\*\*\*\*\*
\*\*\*\*\*

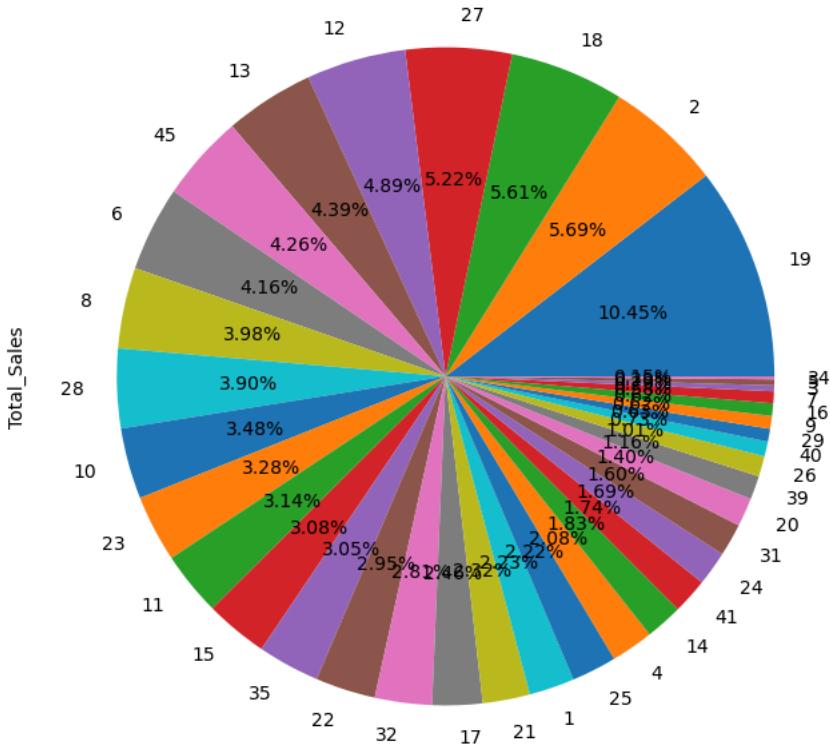
Dept: 78

## Total Sales by Store





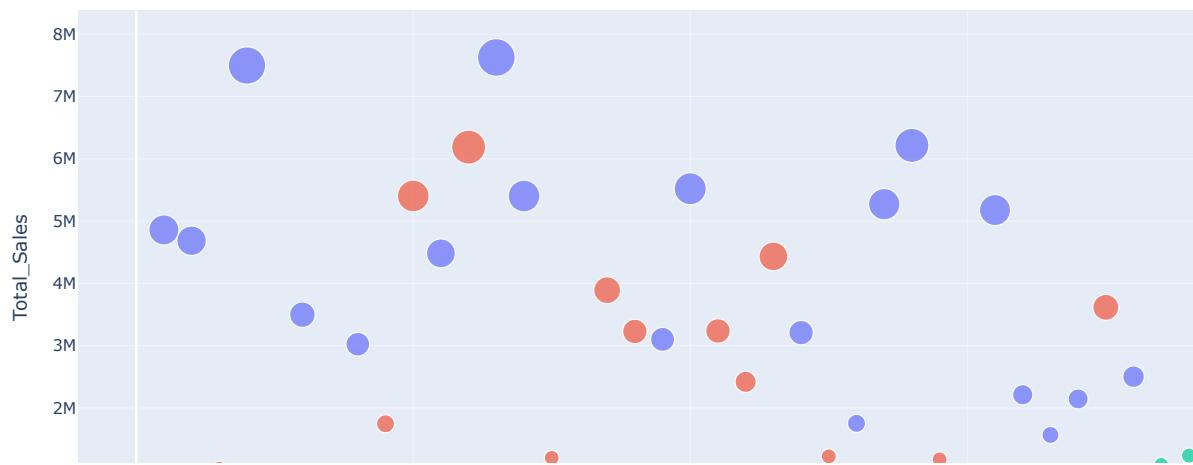
Total Sales by Store



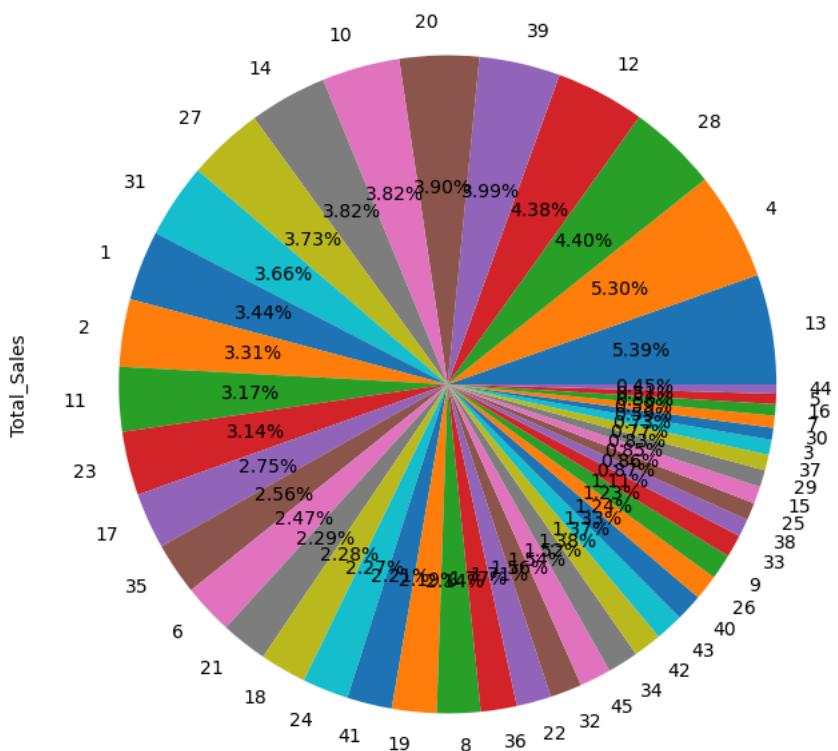
\*\*\*\*\*
\*\*\*\*\*  
\*\*\*\*\*

Dept: 79

## Total Sales by Store



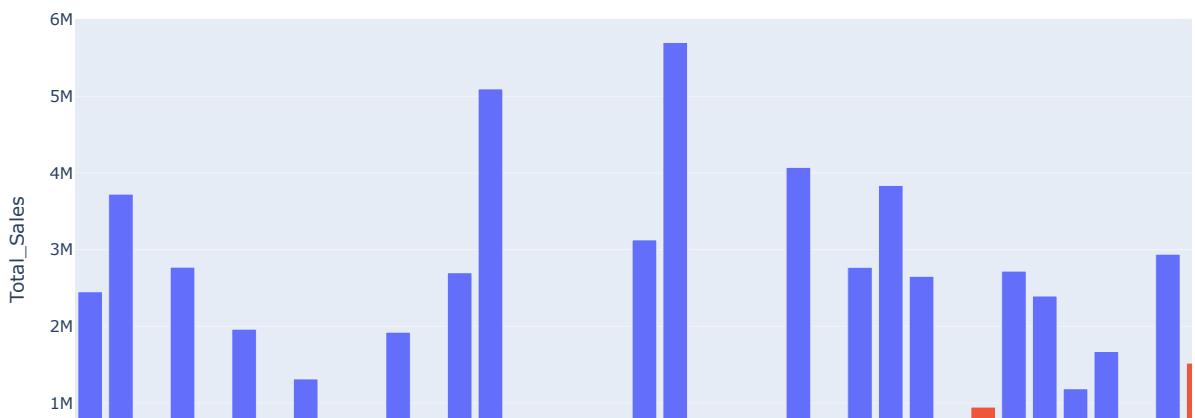
## Total Sales by Store



\*\*\*\*\*  
\*\*\*\*\*

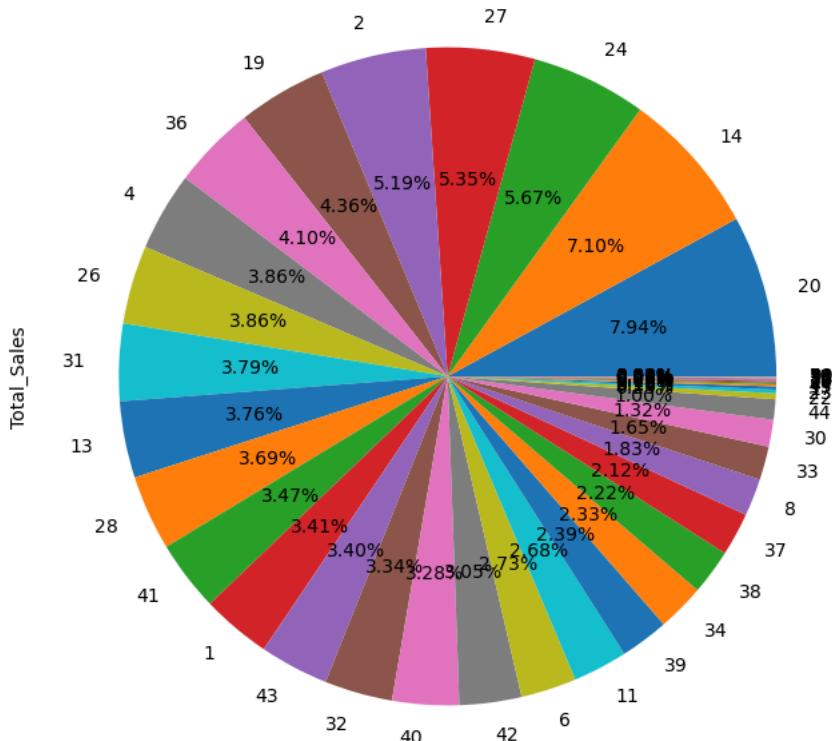
Dept: 80

## Total Sales by Store





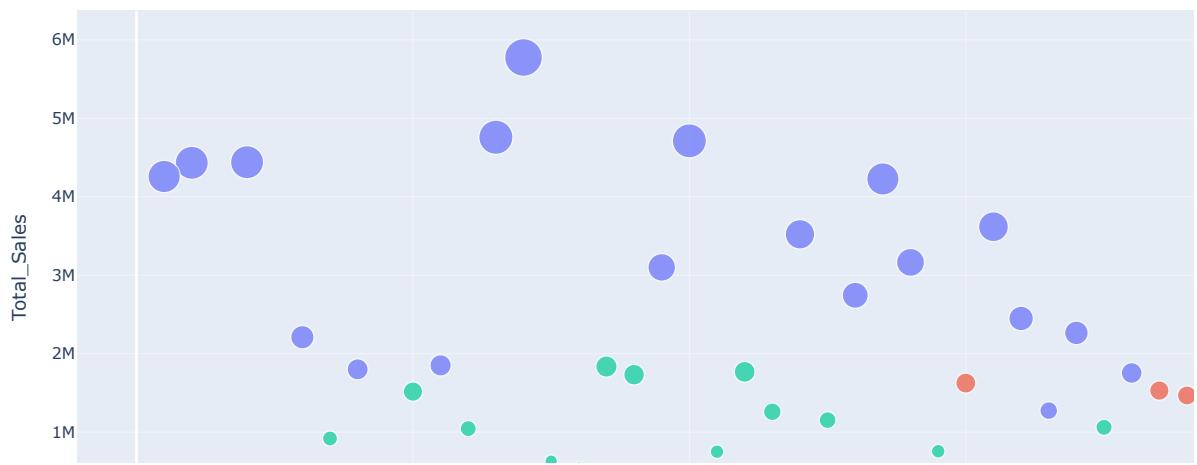
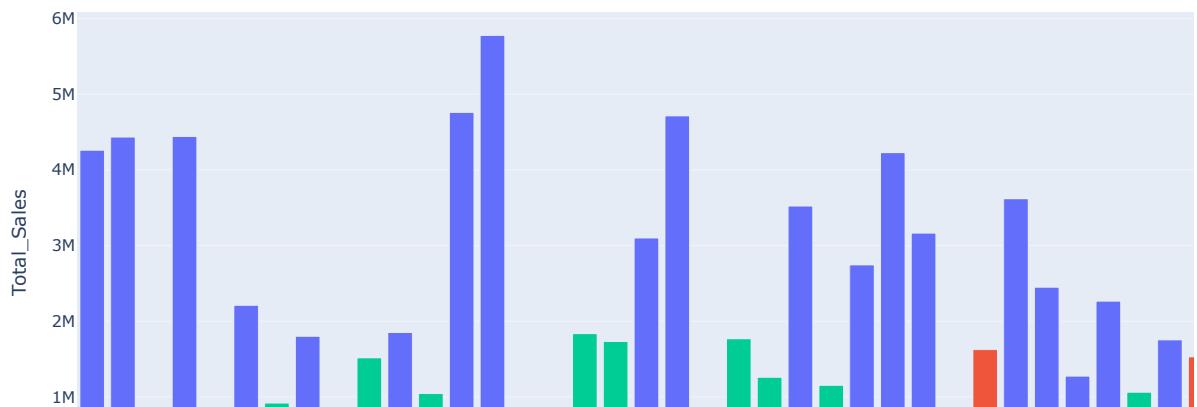
Total Sales by Store



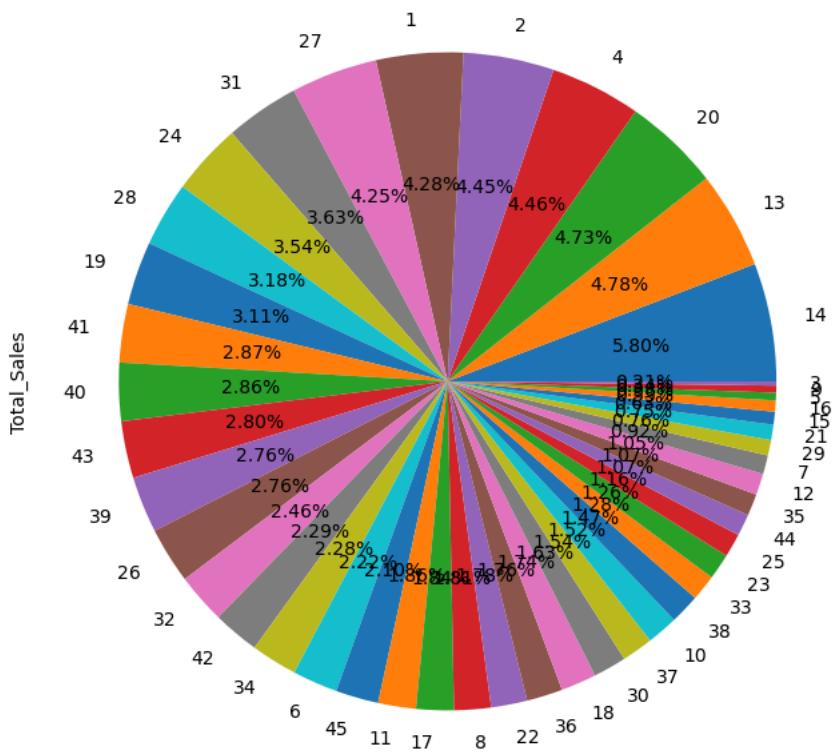
```
*****
*****
```

Dept: 81

## Total Sales by Store



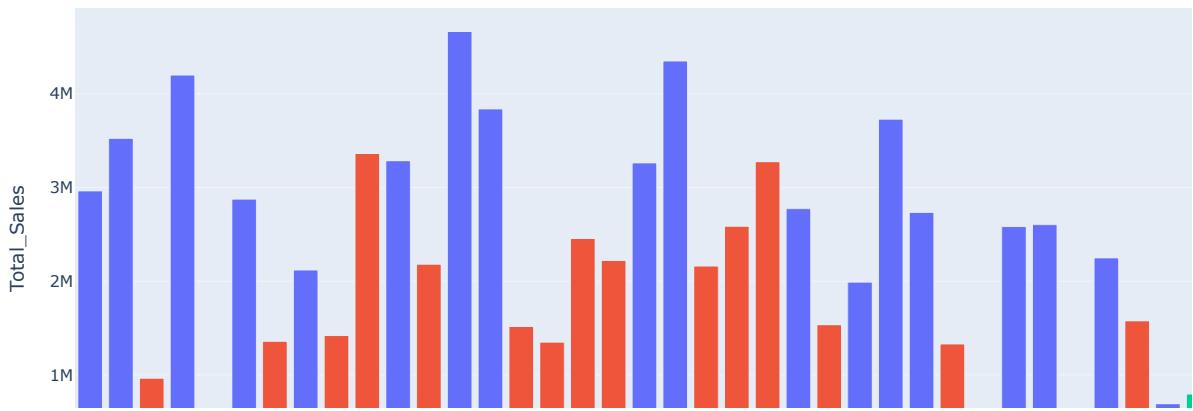
### Total Sales by Store

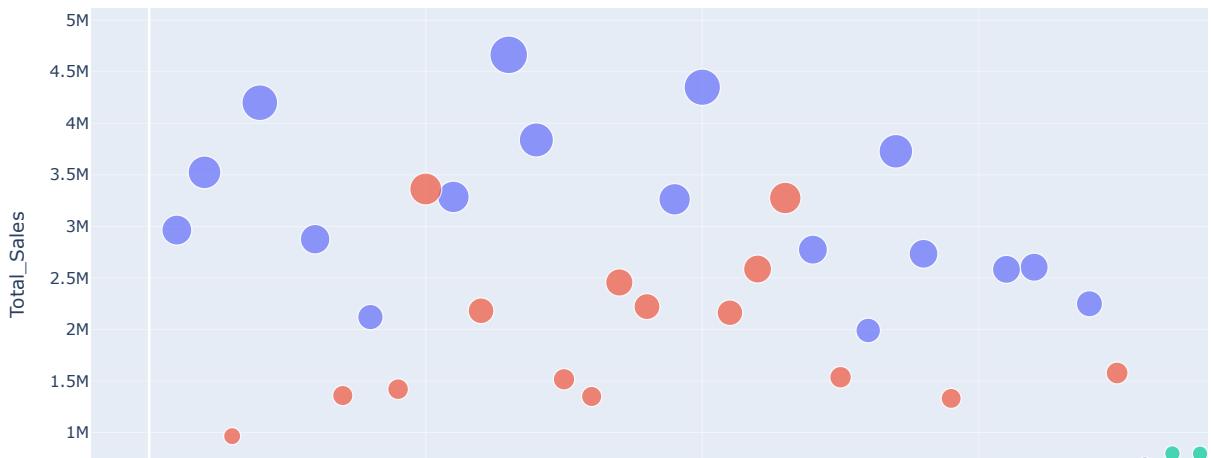


\*\*\*\*\*
\*\*\*\*\*

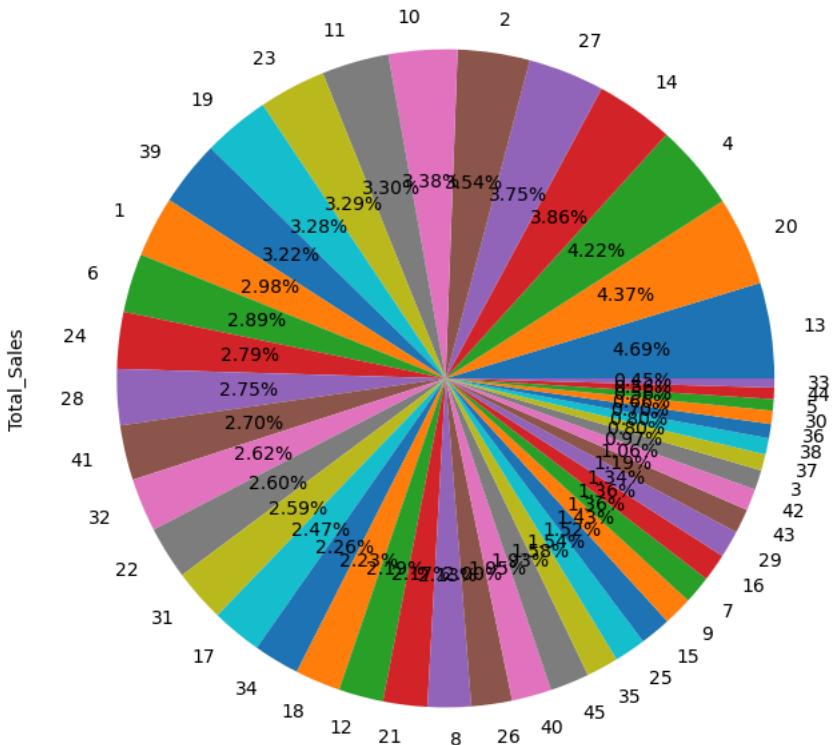
Dept: 82

### Total Sales by Store





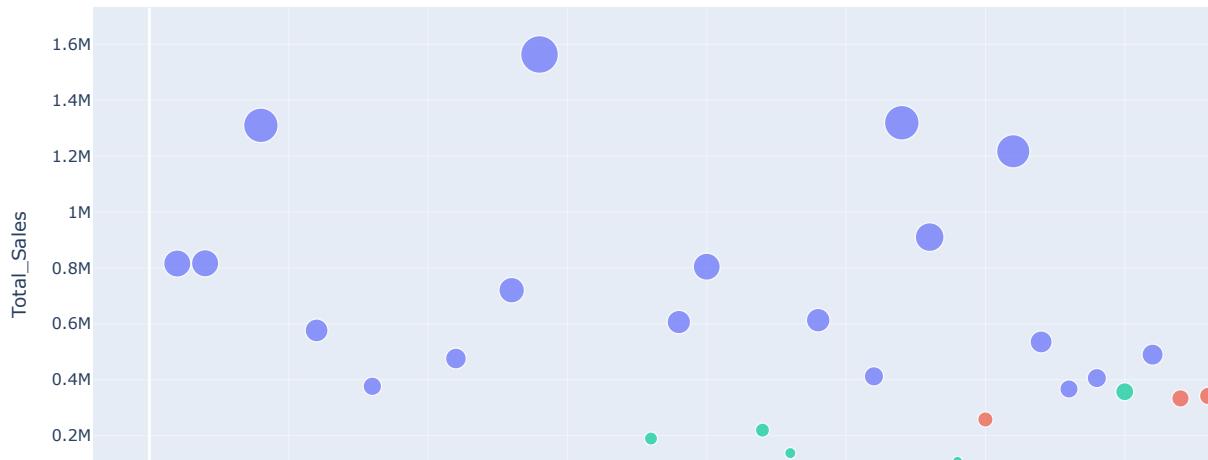
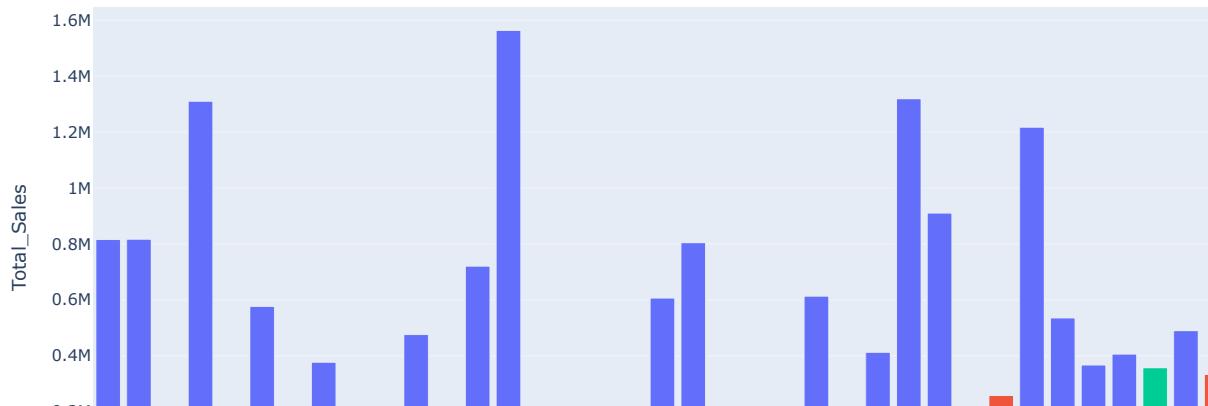
Total Sales by Store

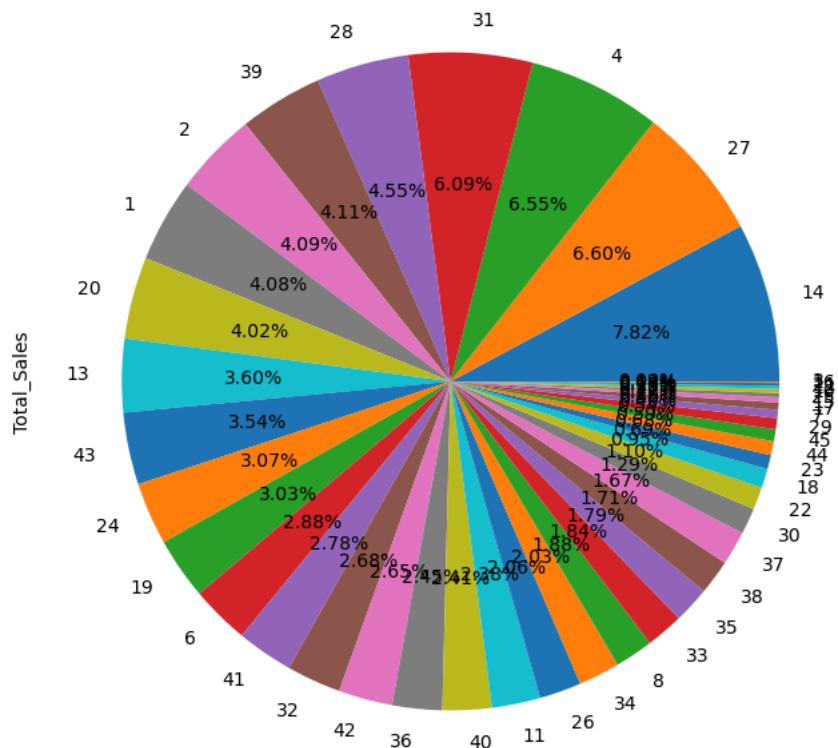


\*\*\*\*\*
\*\*\*\*\*  
\*\*\*\*\*

Dept: 83

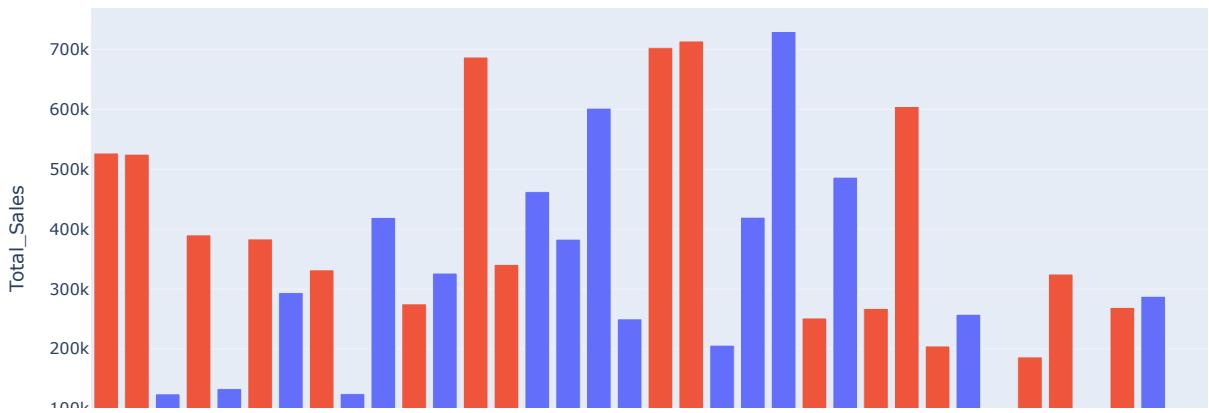
## Total Sales by Store

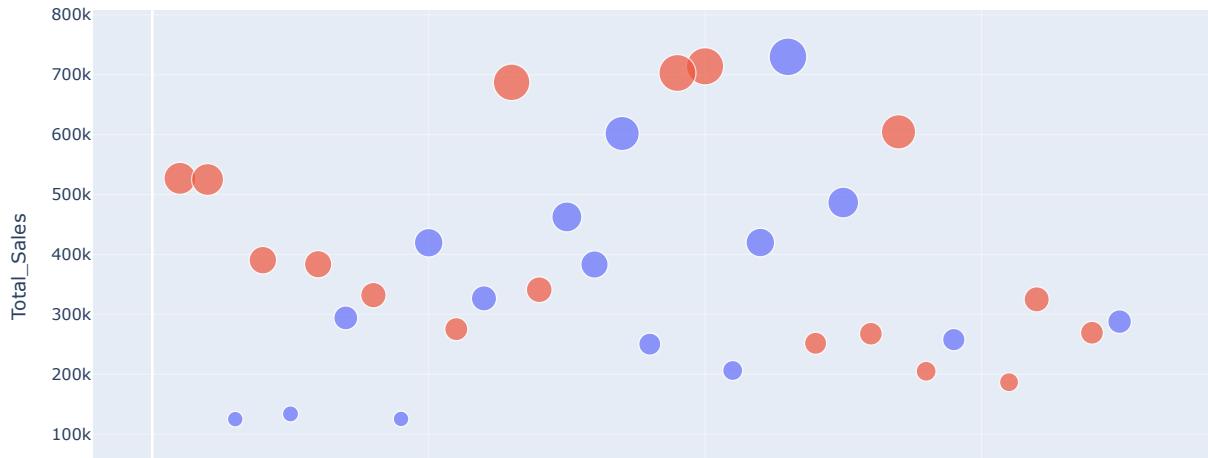


**Total Sales by Store**

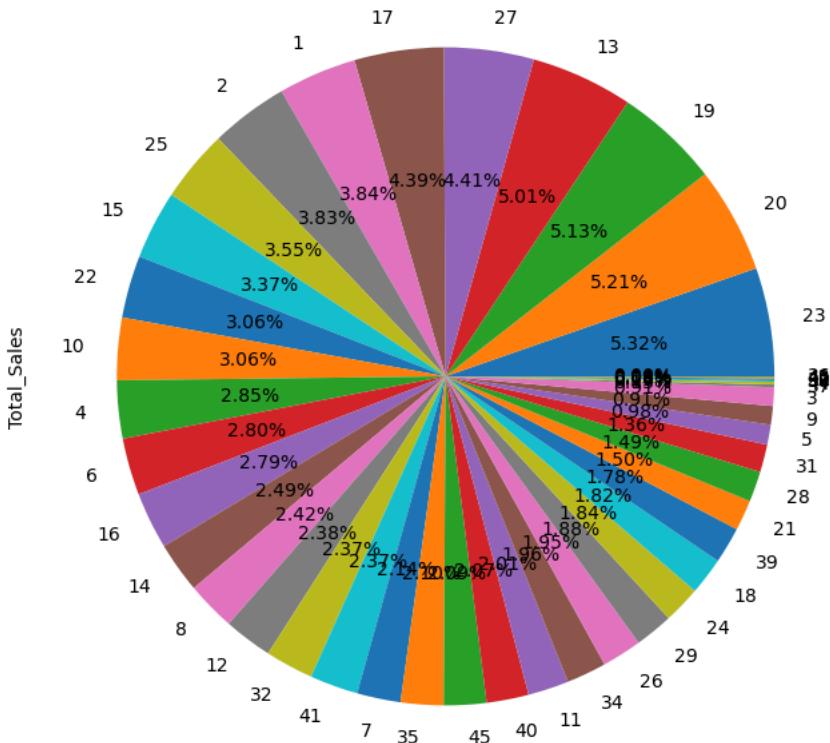
\*\*\*\*\*
\*\*\*\*\*

Dept: 85

**Total Sales by Store**



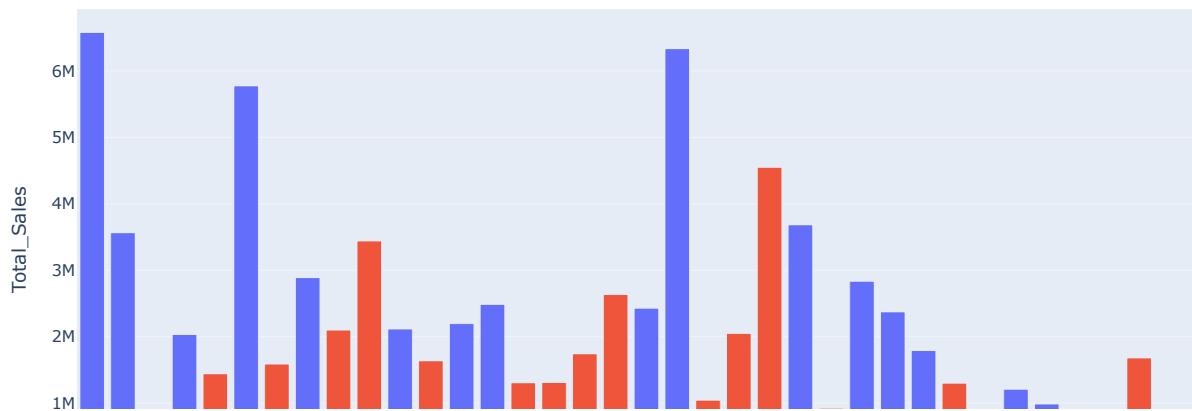
Total Sales by Store

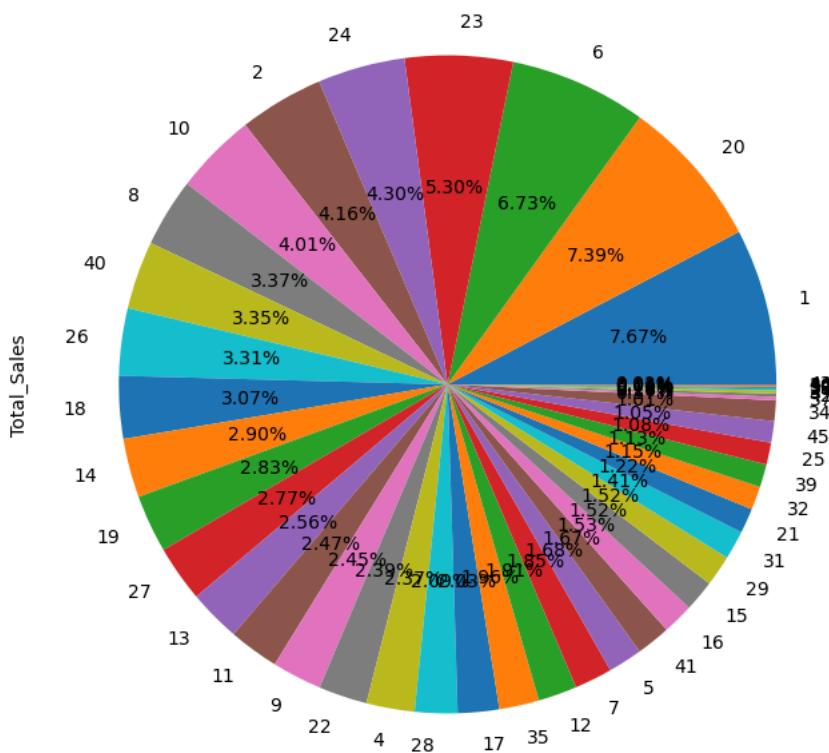


```
*****
*****
```

Dept: 87

## Total Sales by Store



**Total Sales by Store**

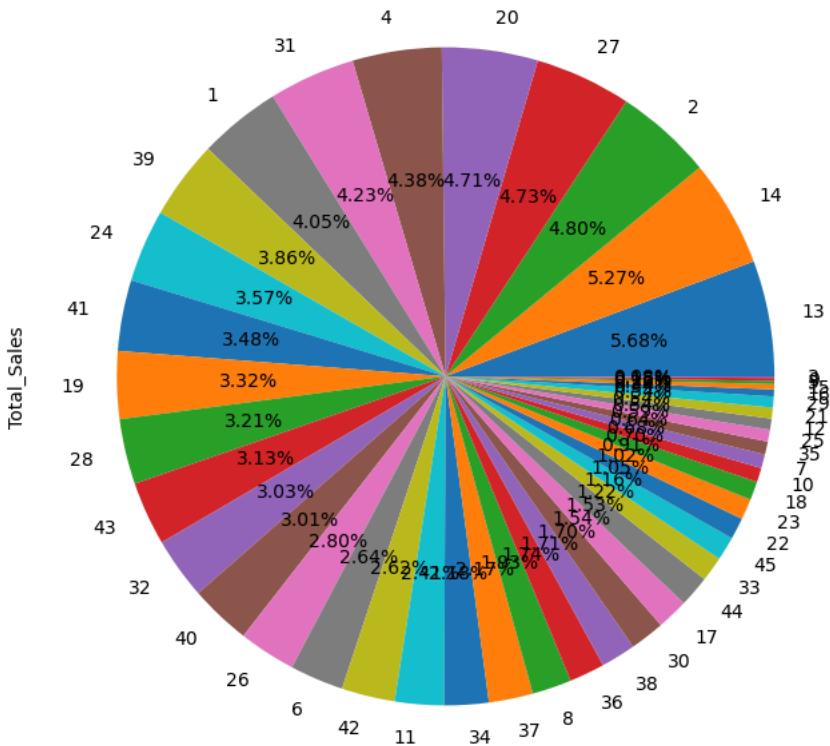
\*\*\*\*\*
\*\*\*\*\*

Dept: 90

**Total Sales by Store**



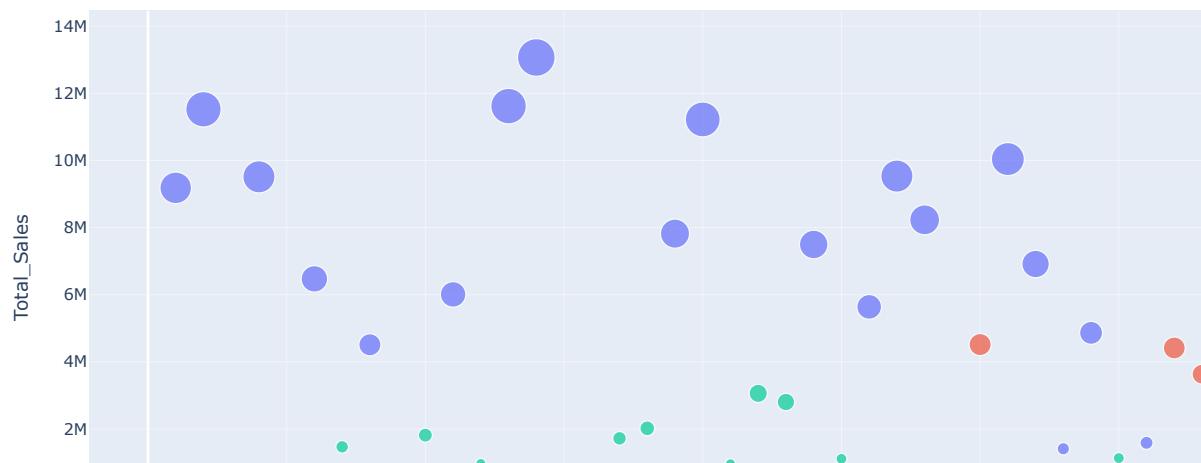
Total Sales by Store



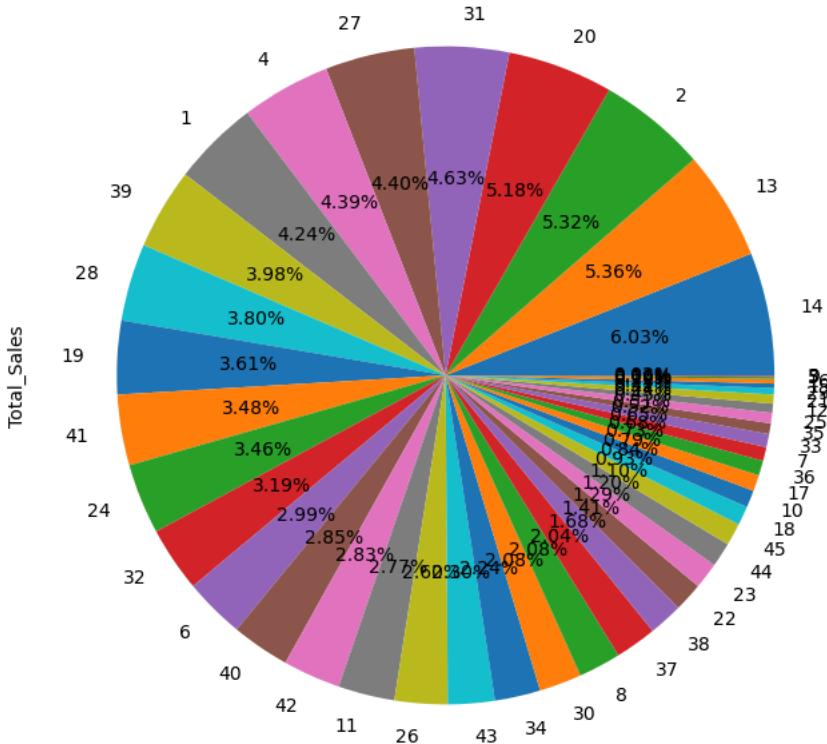
\*\*\*\*\*
\*\*\*\*\*  
\*\*\*\*\*

Dept: 91

## Total Sales by Store



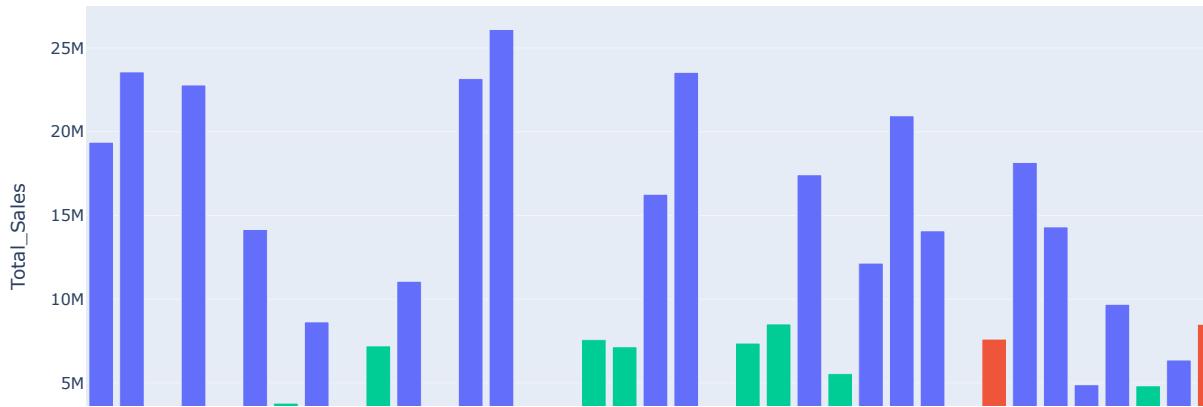
## Total Sales by Store



\*\*\*\*\*  
\*\*\*\*\*

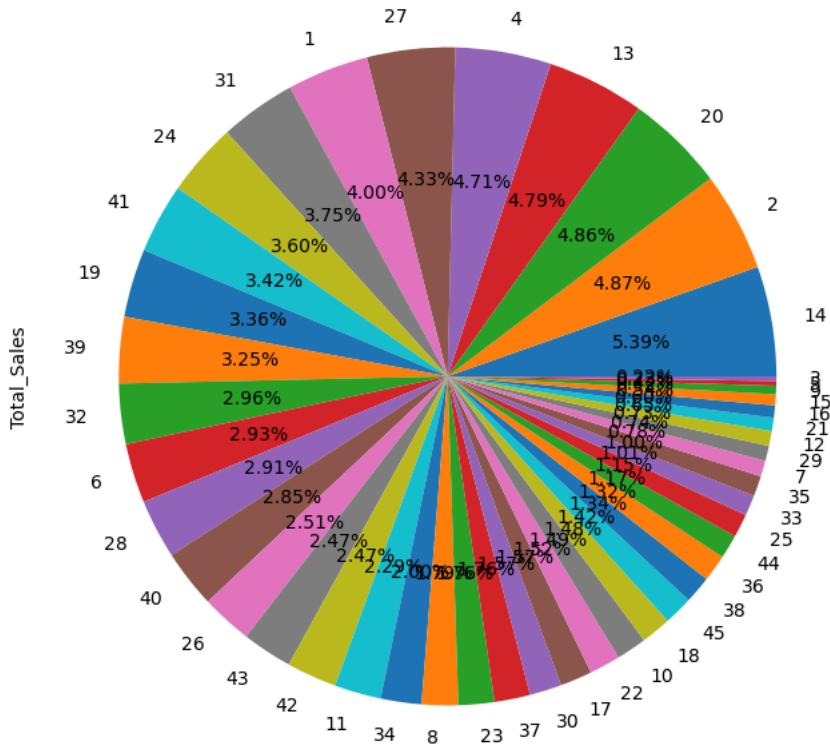
Dept: 92

## Total Sales by Store





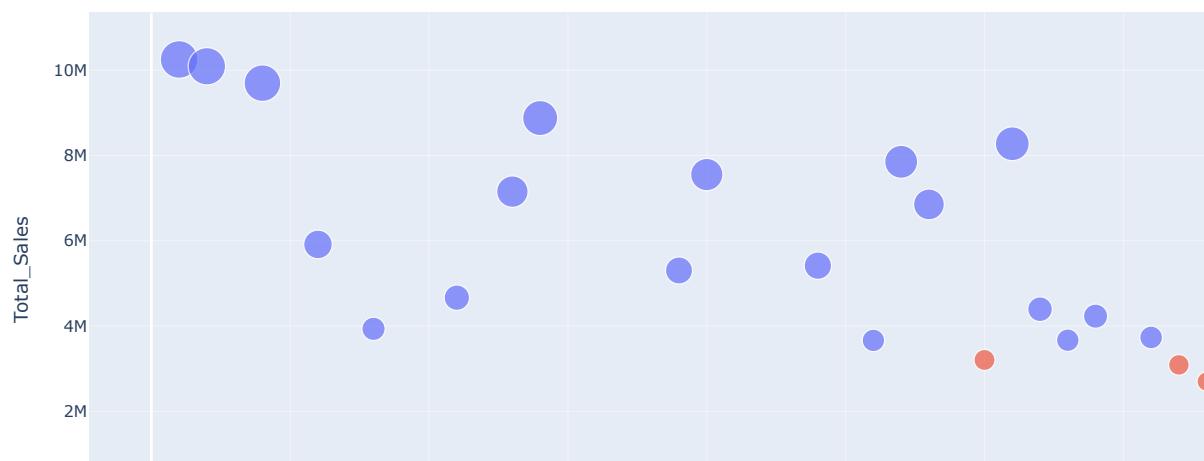
Total Sales by Store



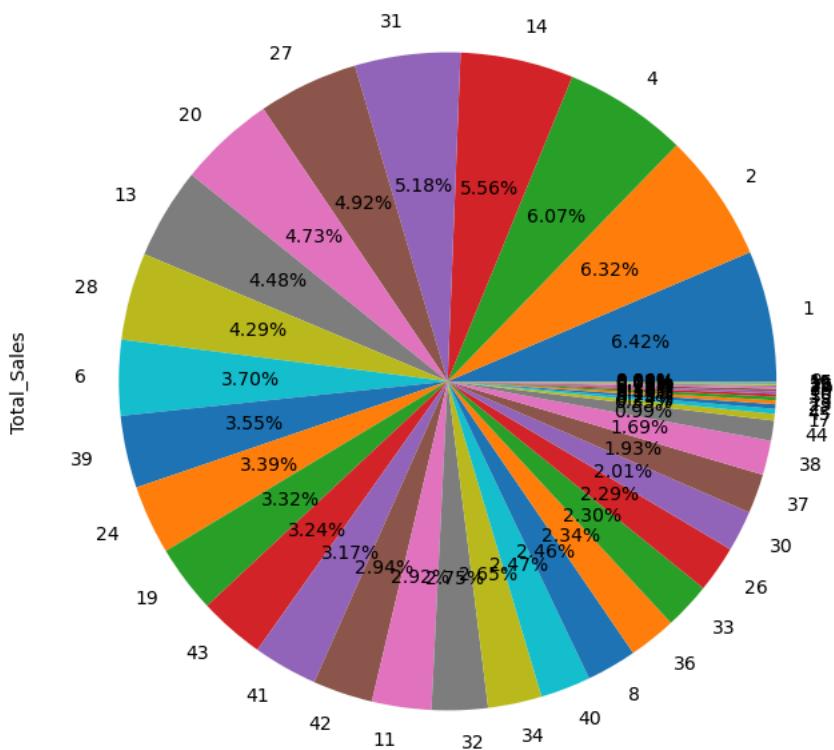
\*\*\*\*\*
\*\*\*\*\*  
\*\*\*\*\*

Dept: 93

## Total Sales by Store



## Total Sales by Store

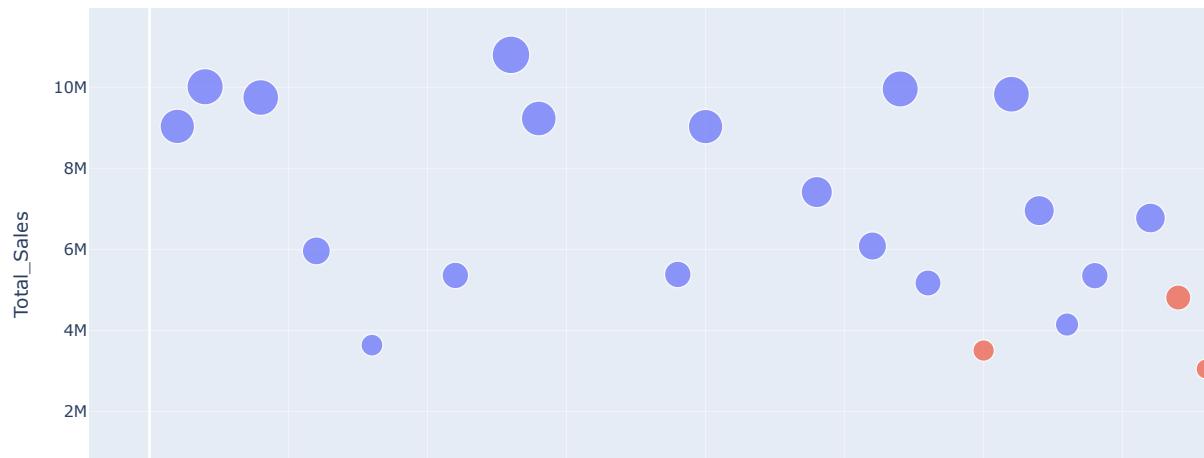


\*\*\*\*\*
\*\*\*\*\*

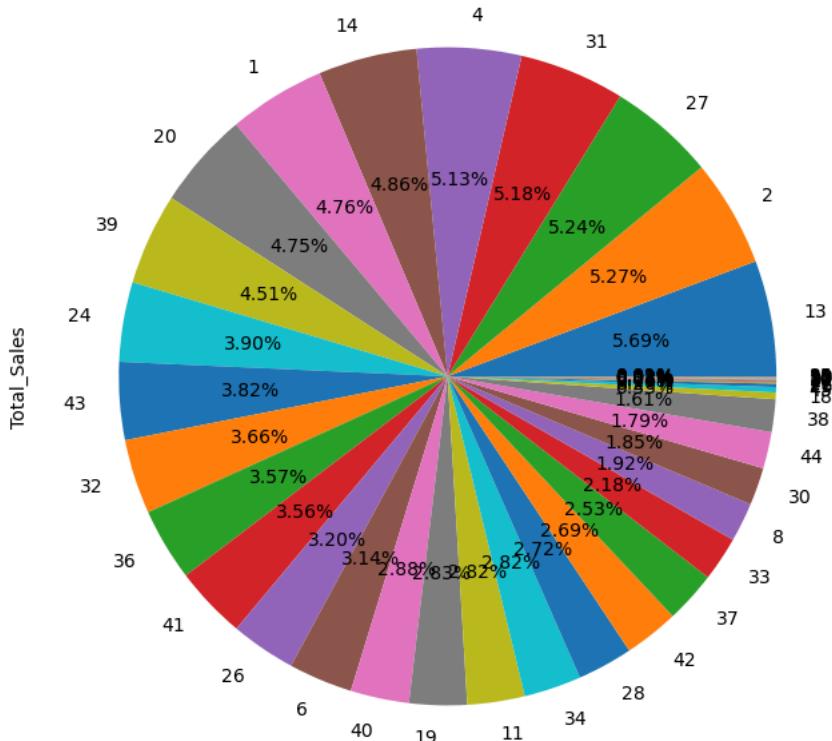
Dept: 94

## Total Sales by Store





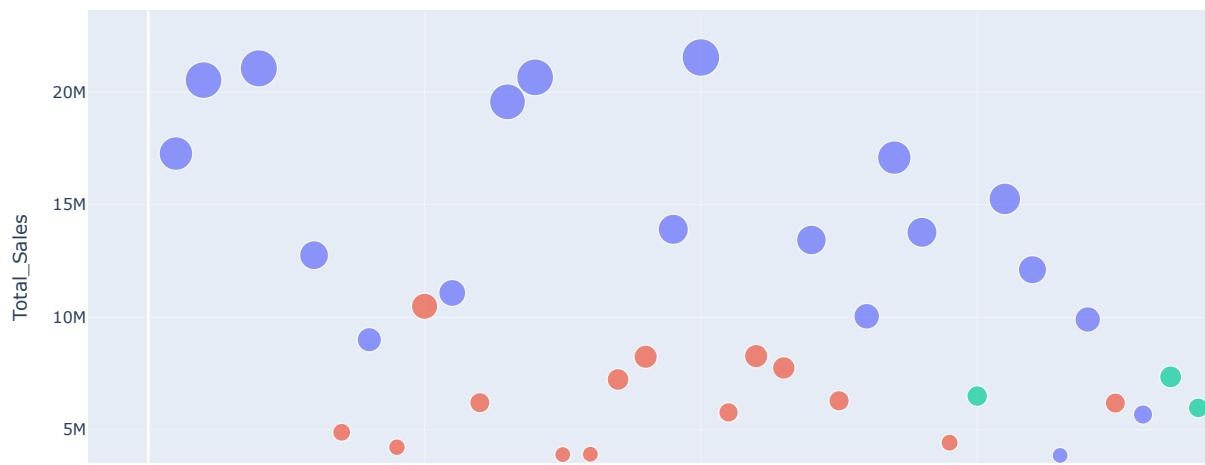
Total Sales by Store



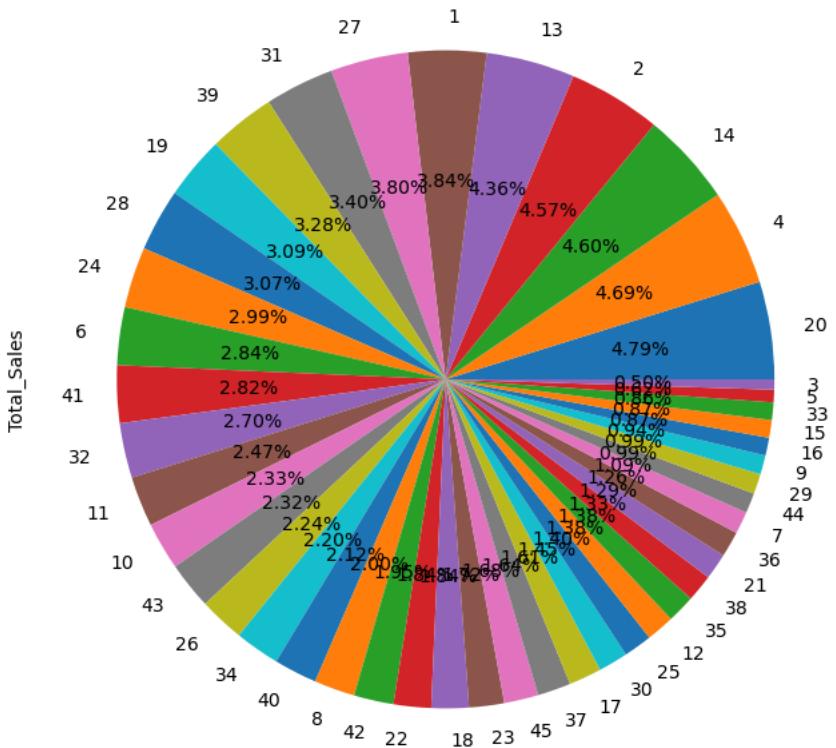
\*\*\*\*\*
\*\*\*\*\*  
\*\*\*\*\*

Dept: 95

## Total Sales by Store



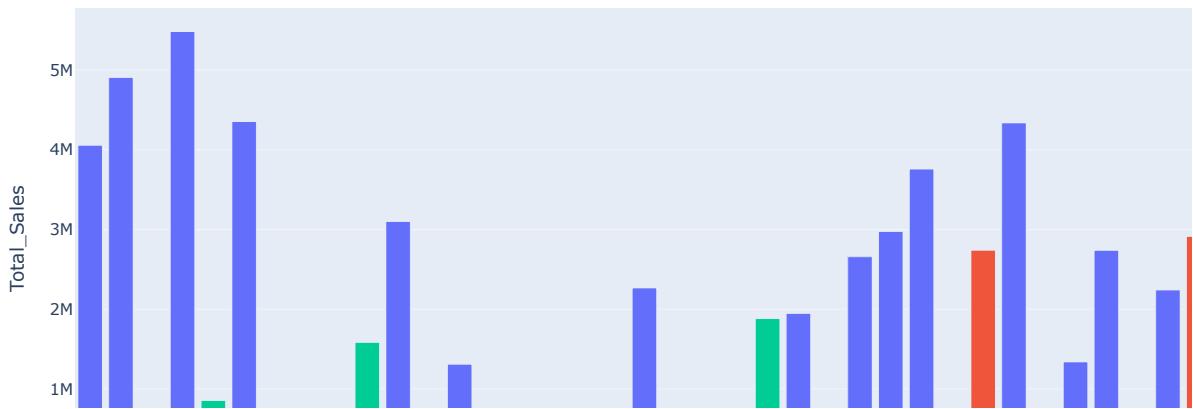
## Total Sales by Store

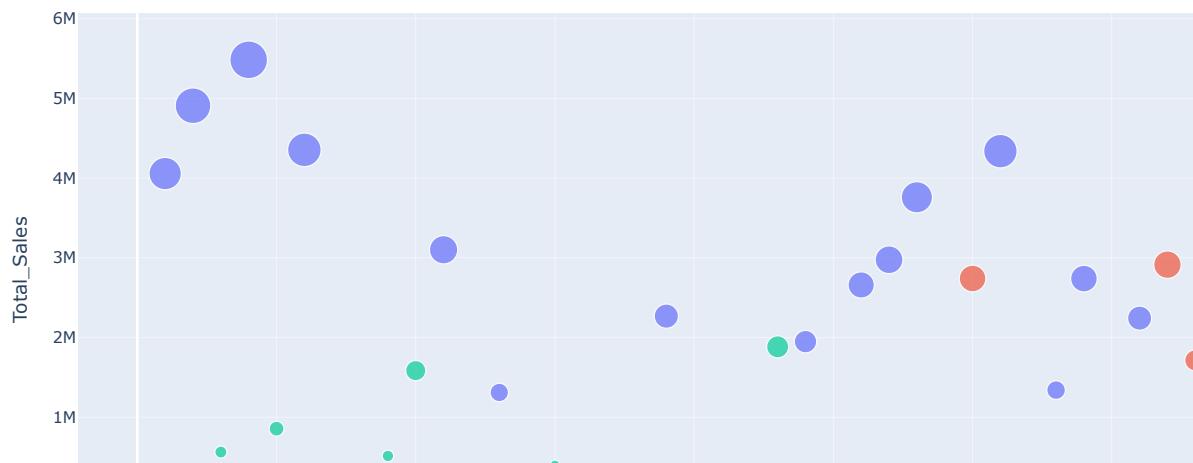


\*\*\*\*\*  
\*\*\*\*\*

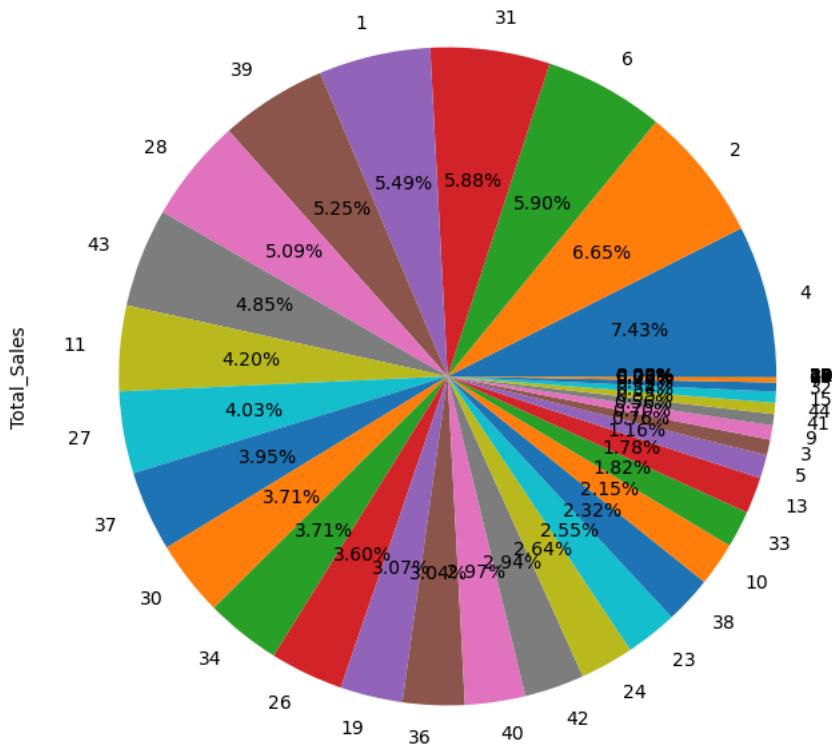
Dept: 96

## Total Sales by Store





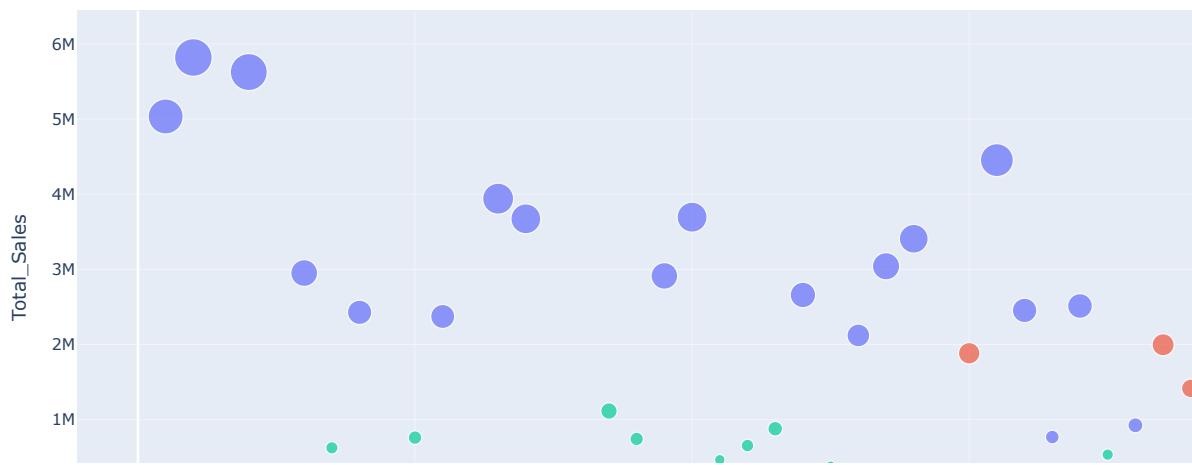
Total Sales by Store

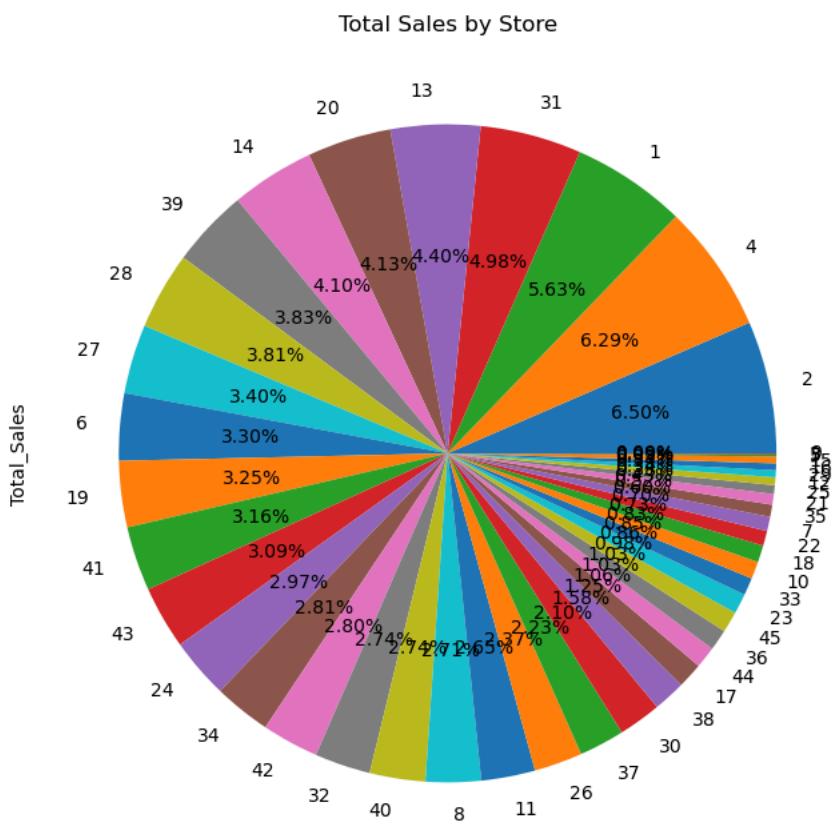


\*\*\*\*\*
\*\*\*\*\*  
\*\*\*\*\*

Dept: 97

## Total Sales by Store



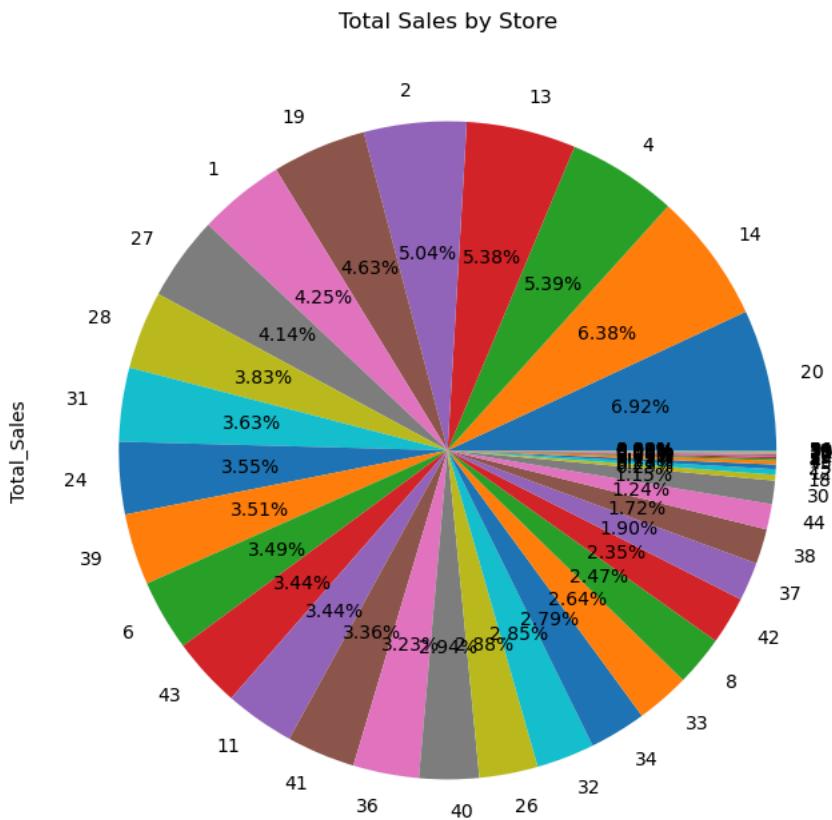
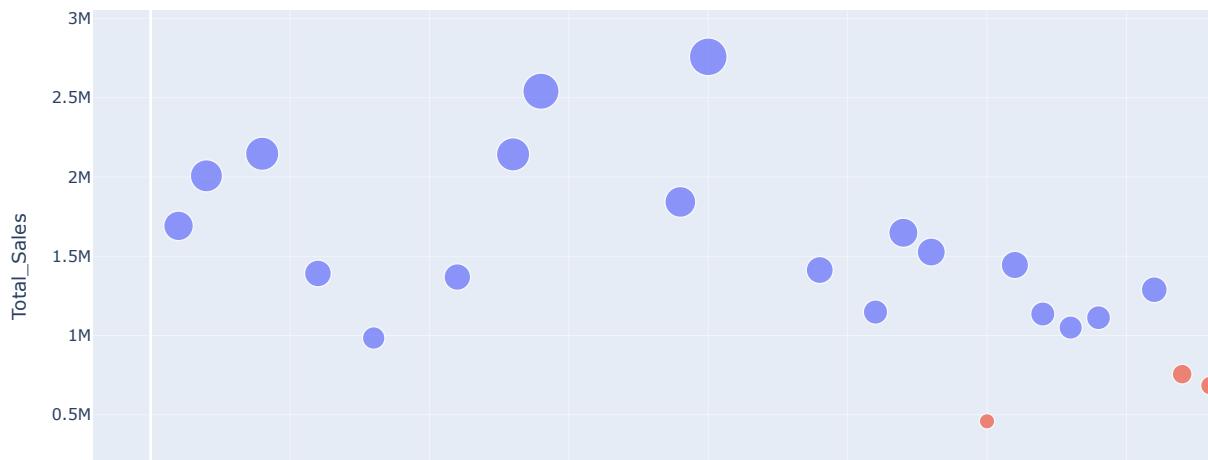


\*\*\*\*\*  
\*\*\*\*\*

Dept: 98

**Total Sales by Store**

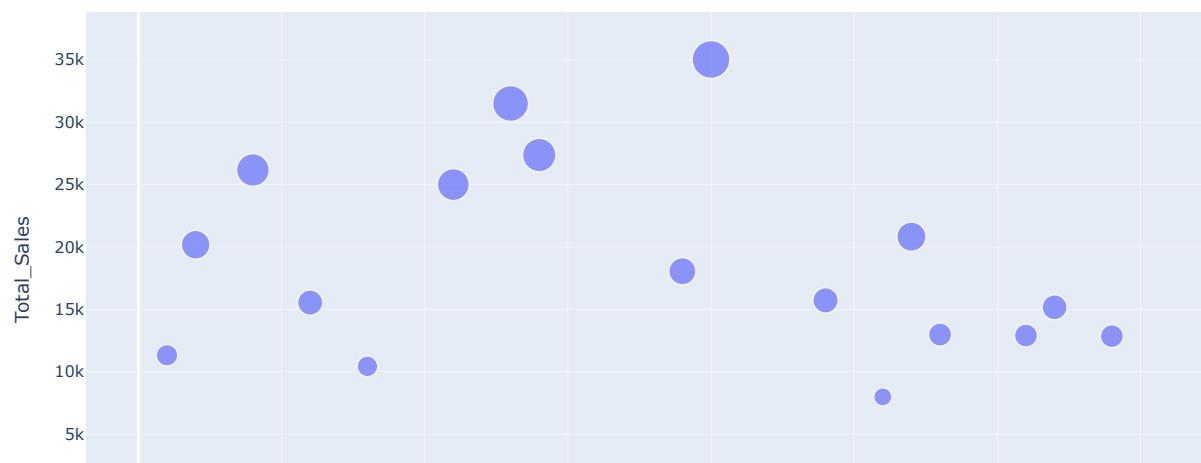




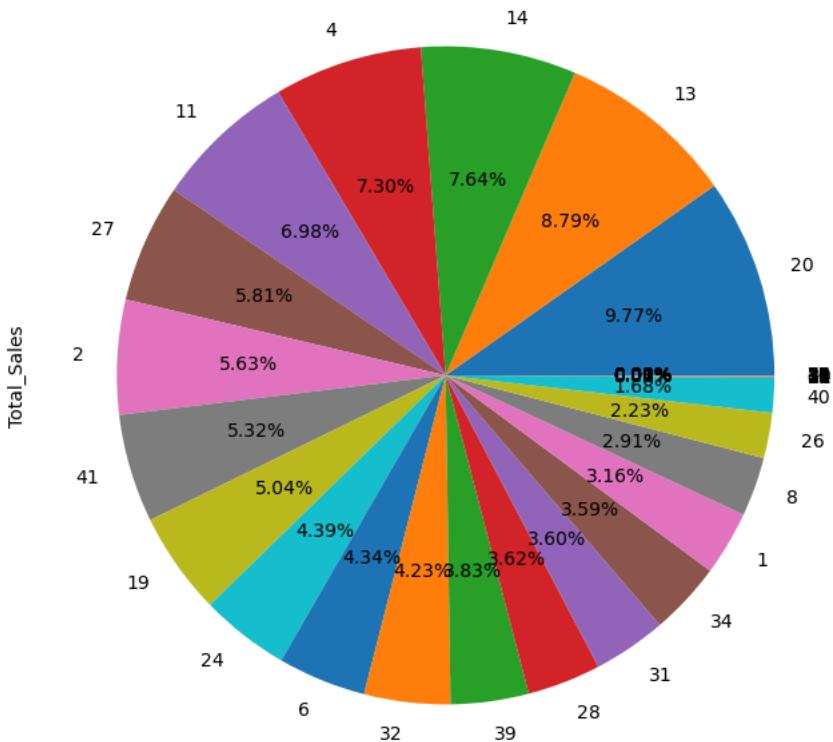
\*\*\*\*\*
\*\*\*\*\*

Dept: 99

## Total Sales by Store



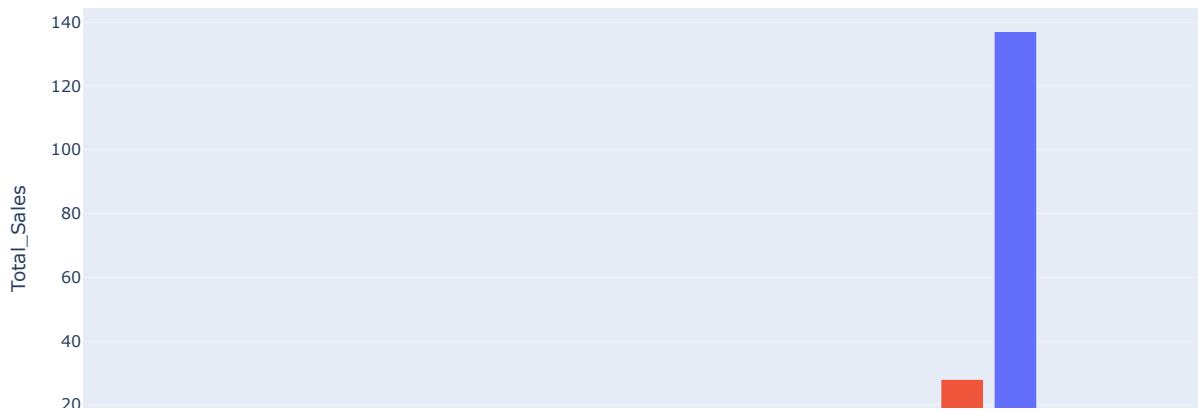
## Total Sales by Store

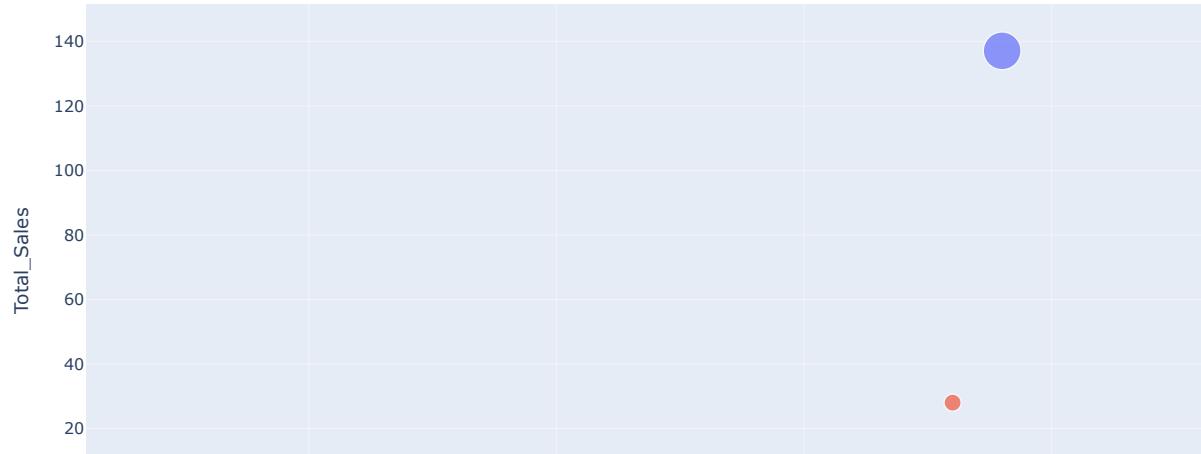


\*\*\*\*\*
\*\*\*\*\*

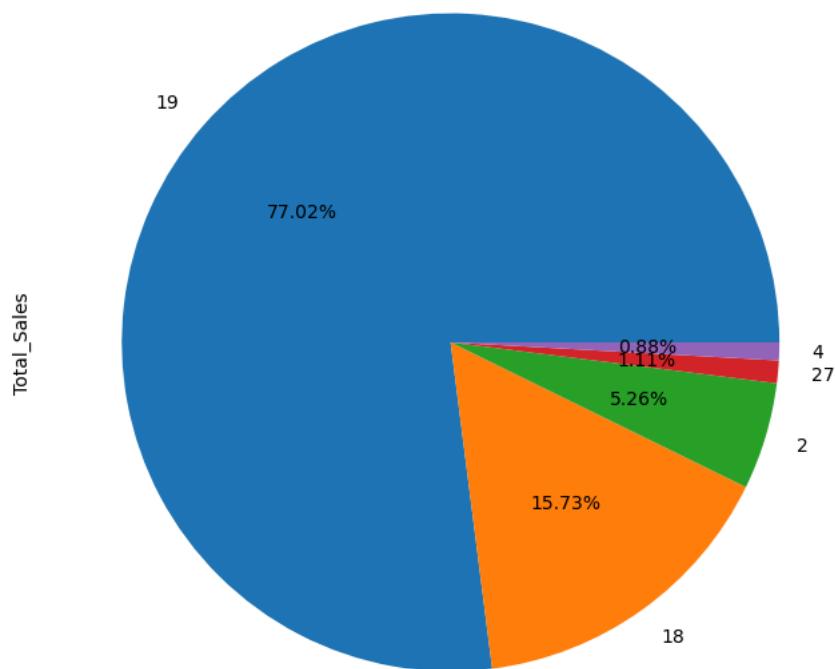
Dept: 39

## Total Sales by Store





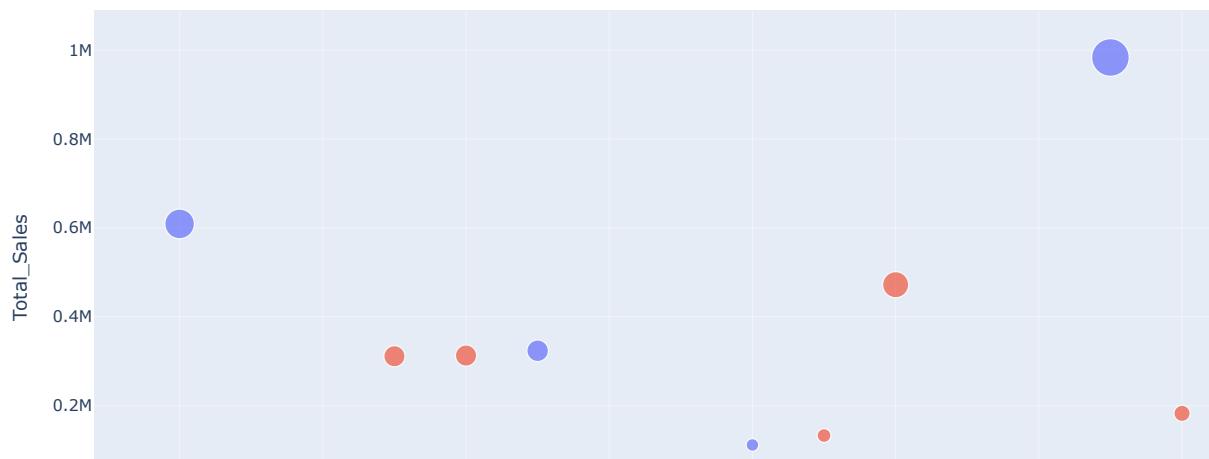
Total Sales by Store

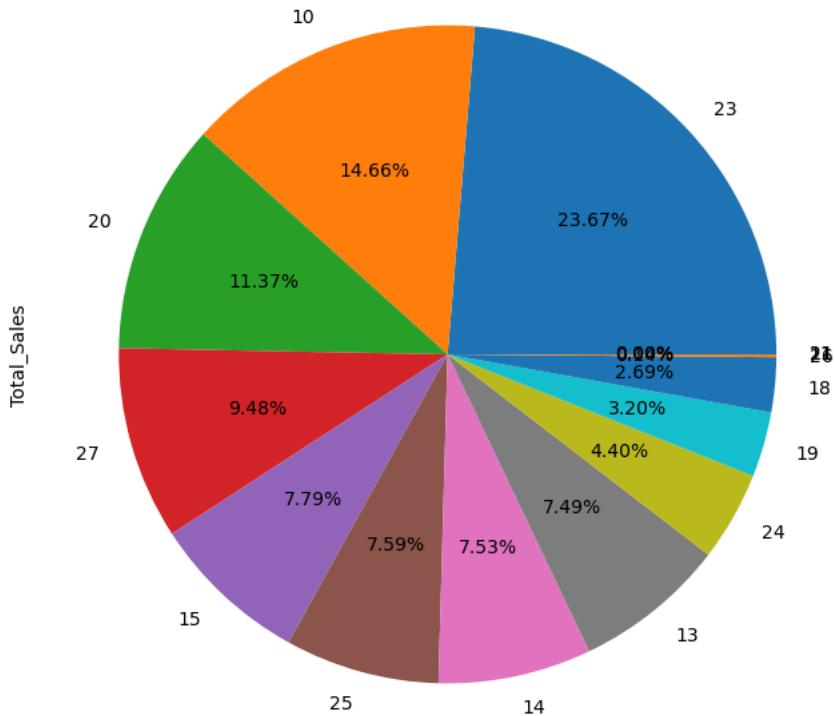


\*\*\*\*\*  
\*\*\*\*\*

Dept: 50

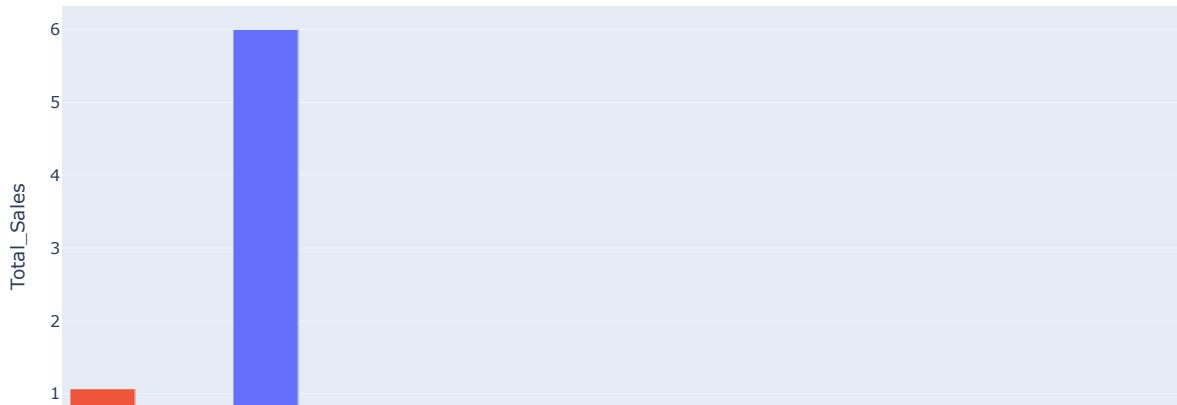
## Total Sales by Store

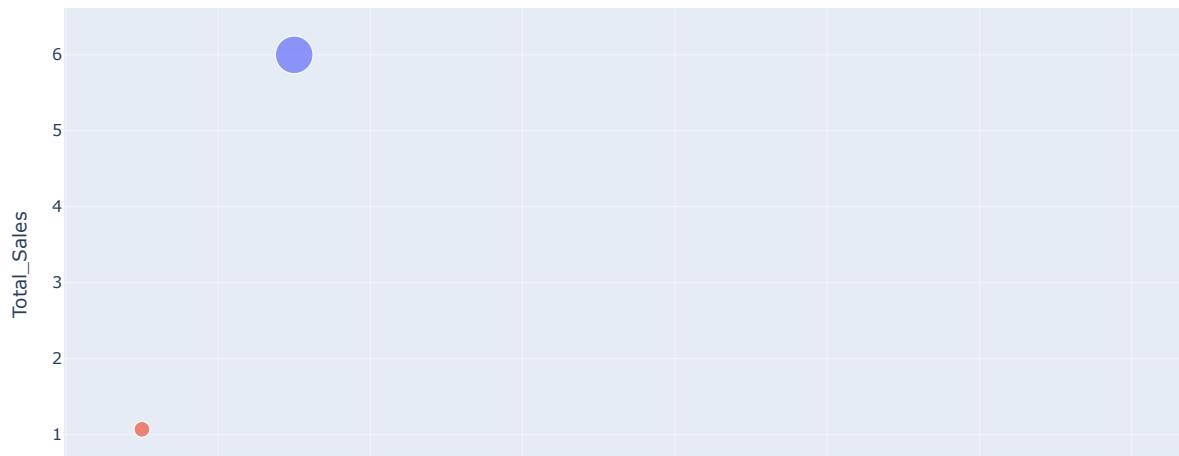


**Total Sales by Store**

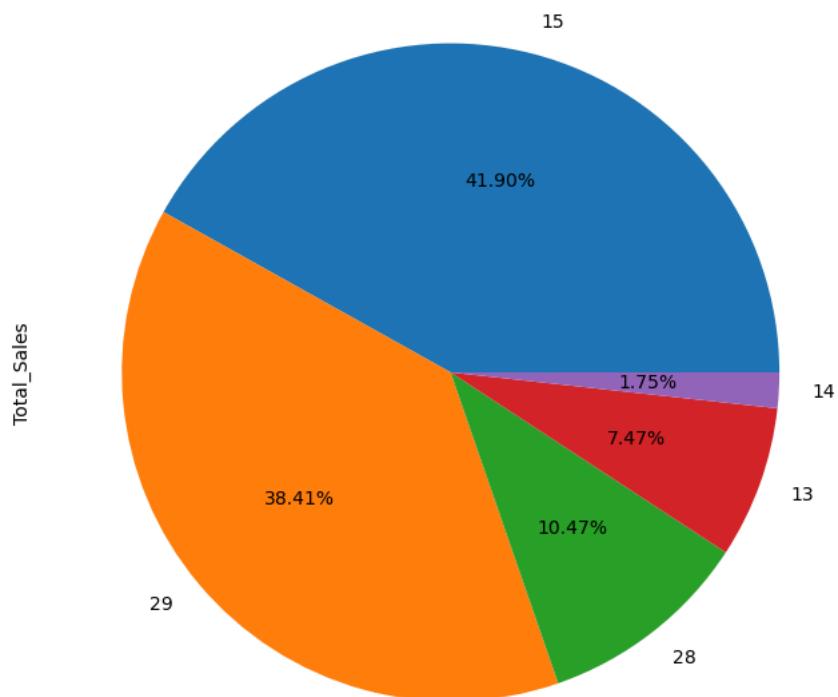
\*\*\*\*\*  
\*\*\*\*\*

Dept: 43

**Total Sales by Store**



Total Sales by Store



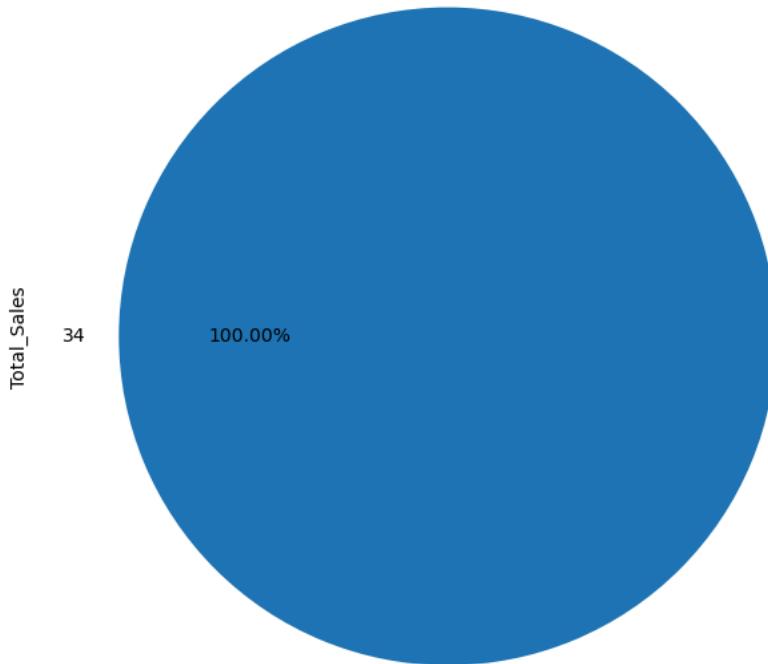
\*\*\*\*\*  
\*\*\*\*\*

Dept: 65

## Total Sales by Store



## Total Sales by Store



```
*****
*****
```

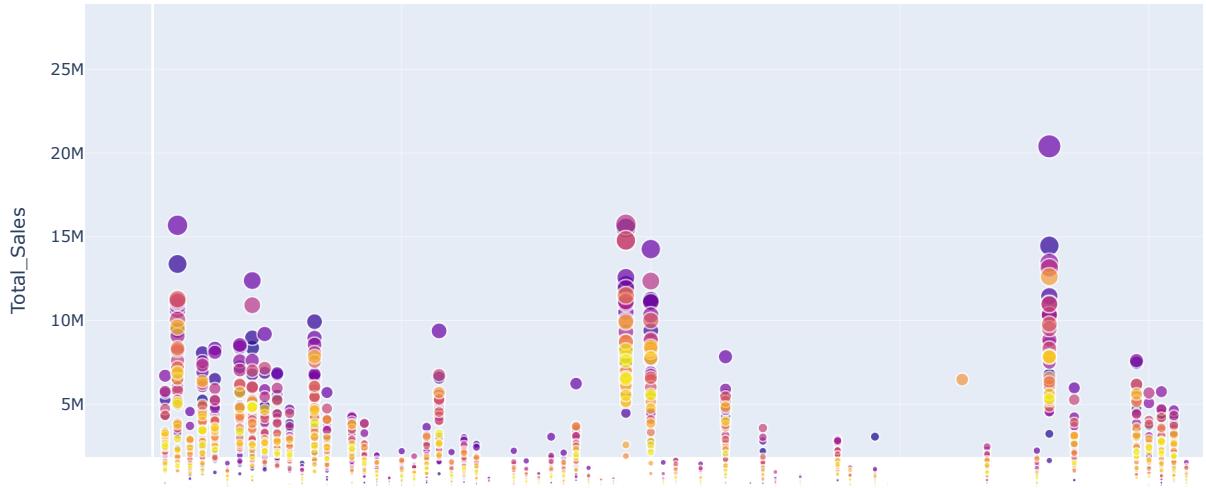
```
In [27]: # @title $Store \ \ \& \ \ Department \ \ wise \ \ sales
store_dept_wise_sales = sales.groupby(['Store','Dept'])['Weekly_Sales'].sum()
store_dept_wise_sales = store_dept_wise_sales.reset_index()
#store_dept_wise_total_sales["Store"] = store_dept_wise_total_sales["Store"].astype(str)
#store_dept_wise_total_sales["Dept"] = store_dept_wise_total_sales["Dept"].astype(str)
store_dept_wise_sales.rename(columns={'Weekly_Sales':'Total_Sales'}, inplace=True)
display(store_dept_wise_sales)

print('\n')

fig = px.scatter(store_dept_wise_sales, x='Dept', y='Total_Sales', color='Store', size='Total_Sales')
fig.show()
```

Store	Dept	Total_Sales
0	1	3219405.18
1	1	6592598.93
2	1	1880518.36
3	1	5285874.09
4	1	3468885.58
...	...	...
3326	45	494496.46
3327	45	7564151.83
3328	45	96 5.94
3329	45	924775.55
3330	45	75767.27

3331 rows × 3 columns

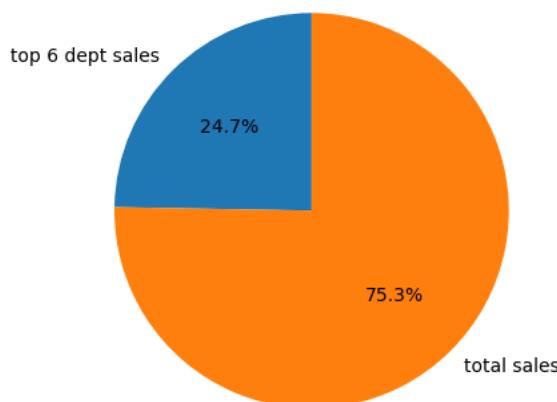


```
In [28]: # @title $Top \ \ n \ \ Departments \ \ Sales$ {run: "auto", "form-width": "20%"}
top_n_depts = 6 # @param {"type": "slider", "min": 1, "max": 98, "step": 1}
top_performing_depts = list(store_dept_wise_sales.groupby(['Dept'])['Total_Sales'].sum().sort_values(ascending=False).index)

top_n_dept_sales = store_dept_wise_sales[store_dept_wise_sales['Dept'].isin(top_performing_depts[:top_n_depts])]['Total_Sales'].sum()

total_sales = store_dept_wise_sales['Total_Sales'].sum()

plt.pie([top_n_dept_sales, total_sales], labels = ['top '+str(top_n_depts)+ ' dept sales', 'total sales'], autopct='%.1f%%', startangle=90)
plt.show()
```



- Department 92 generates highest sales (~6.7%).
- Top 6 departments generates almost quarter (24.7%) of total sales.

## Merge all three datasets

```
In [29]: # @title $Import \ \ Features \ \ Dataset$ {run: "auto", "form-width": "20%"}
features = pd.read_csv('Features.csv')
```

```
In [30]: # @title $Change \ \ type \ \ of \ \ Date \ \ column$ {run: "auto", "form-width": "20%"}
features['Date'] = pd.to_datetime(features['Date'], format='%d/%m/%Y', errors='coerce')
display(features.dtypes)
```

```

Store          int64
Date      datetime64[ns]
Temperature    float64
Fuel_Price     float64
MarkDown1      float64
MarkDown2      float64
MarkDown3      float64
MarkDown4      float64
MarkDown5      float64
CPI           float64
Unemployment   float64
IsHoliday      bool
dtype: object

```

```
In [31]: # @title $Initial \ \ Analysis$
data = sales.merge(features, on=['Store', 'Date', 'IsHoliday'], how='left')
initial_analysis(data)
```

first 5 records

Store	Dept	Date	IsHoliday	Weekly_Sales	Type	Area	Temperature	Fuel_Price	MarkDown1	MarkDown2	MarkDown3	MarkDown4
0	1	1 2010-10-01	False	20094.19	A 151315		71.89	2.603	NaN	NaN	NaN	NaN
1	1	1 2010-04-02	False	57258.43	A 151315		62.27	2.719	NaN	NaN	NaN	NaN
2	1	1 2010-07-02	False	16333.14	A 151315		80.91	2.669	NaN	NaN	NaN	NaN
3	1	1 2010-09-03	False	16241.78	A 151315		81.21	2.577	NaN	NaN	NaN	NaN
4	1	1 2010-12-03	False	22517.56	A 151315		49.27	2.708	NaN	NaN	NaN	NaN

last 5 records

Store	Dept	Date	IsHoliday	Weekly_Sales	Type	Area	Temperature	Fuel_Price	MarkDown1	MarkDown2	MarkDown3	MarkDo
421585	45	98 2010-07-30	False	5.50	B 118221		79.78	2.781	NaN	NaN	NaN	
421586	45	98 2011-09-30	False	727.05	B 118221		70.66	3.645	NaN	NaN	NaN	
421587	45	98 2011-12-30	True	553.21	B 118221		37.79	3.389	3604.35	43941.56	287.45	35
421588	45	98 2012-08-31	False	346.04	B 118221		75.09	3.867	23641.30	6.00	92.93	698
421589	45	98 2010-12-31	True	74.55	B 118221		29.67	3.179	NaN	NaN	NaN	

Shape of dataset  
(421590, 16)

```

Data types
Store          int64
Dept           int64
Date      datetime64[ns]
IsHoliday      bool
Weekly_Sales   float64
Type          object
Area           int64
Temperature    float64
Fuel_Price     float64
MarkDown1      float64
MarkDown2      float64
MarkDown3      float64
MarkDown4      float64
MarkDown5      float64
CPI            float64
Unemployment   float64
dtype: object

```

```
Dataset info
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 421590 entries, 0 to 421589
Data columns (total 16 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   Store        421590 non-null   int64  
 1   Dept         421590 non-null   int64  
 2   Date         421590 non-null   datetime64[ns]
 3   IsHoliday    421590 non-null   bool   
 4   Weekly_Sales 421590 non-null   float64
 5   Type         421590 non-null   object  
 6   Area          421590 non-null   int64  
 7   Temperature  421590 non-null   float64
 8   Fuel_Price   421590 non-null   float64
 9   MarkDown1   150681 non-null   float64
 10  MarkDown2   111248 non-null   float64
 11  MarkDown3   137091 non-null   float64
 12  MarkDown4   134967 non-null   float64
 13  MarkDown5   151432 non-null   float64
 14  CPI          421590 non-null   float64
 15  Unemployment 421590 non-null   float64
dtypes: bool(1), datetime64[ns](1), float64(10), int64(3), object(1)
memory usage: 48.6+ MB
None
```

## Dataset description

	Store	Dept	Date	Weekly_Sales	Area	Temperature	Fuel_Price	MarkDown1	MarkDo
<b>count</b>	421590.000000	421590.000000		421590	421590.000000	421590.000000	421590.000000	150681.000000	111248.0000
<b>mean</b>	22.201485	44.262108	2011-06-18 08:07:00.533693696	15980.709098	136723.312313	60.090600	3.36101	7246.420196	3334.628
<b>min</b>	1.000000	1.000000	2010-02-05 00:00:00	0.000000	34875.000000	-2.060000	2.47200	0.270000	-265.760
<b>25%</b>	11.000000	18.000000	2010-10-08 00:00:00	2079.790000	93638.000000	46.680000	2.93300	2240.270000	41.600
<b>50%</b>	22.000000	37.000000	2011-06-17 00:00:00	7611.210000	140167.000000	62.090000	3.45200	5347.450000	192.000
<b>75%</b>	33.000000	74.000000	2012-02-24 00:00:00	20205.640000	202505.000000	74.280000	3.73800	9210.900000	1926.940
<b>max</b>	45.000000	99.000000	2012-10-26 00:00:00	693099.360000	219622.000000	100.140000	4.46800	88646.760000	104519.540
<b>std</b>	12.785721	30.492439		NaN	22710.738191	60982.799374	18.447929	0.45851	8291.221345

◀ ▶

## Type

<b>count</b>	421590
<b>unique</b>	3
<b>top</b>	A
<b>freq</b>	215478

## IsHoliday

<b>count</b>	421590
<b>unique</b>	2
<b>top</b>	False
<b>freq</b>	391927

Duplicate Records  
0

Missing Values

```
Store          0
Dept          0
Date          0
IsHoliday     0
Weekly_Sales  0
Type          0
Area          0
Temperature   0
Fuel_Price    0
MarkDown1    270909
MarkDown2    310342
MarkDown3    284499
MarkDown4    286623
MarkDown5    270158
CPI          0
Unemployment 0
dtype: int64
```

```
In [32]: # @title $Date \ wise \ total \ weekly \ sale$
date_wise_weekly_sales = data.groupby(['Date'])['Weekly_Sales'].sum()
date_wise_weekly_sales = date_wise_weekly_sales.reset_index()
display(date_wise_weekly_sales)
```

	Date	Weekly_Sales
0	2010-02-05	49750875.98
1	2010-02-12	48336800.10
2	2010-02-19	48277902.33
3	2010-02-26	43970440.65
4	2010-03-05	46872715.16
...	...	...
138	2012-09-28	43735713.56
139	2012-10-05	47567053.49
140	2012-10-12	46128909.98
141	2012-10-19	45122443.96
142	2012-10-26	45544379.53

143 rows × 2 columns

```
In [33]: # @title $Function \ to \ Plot \ Time-Series \ and \ Moving \ Averages$
import plotly.graph_objs as go
from plotly.subplots import make_subplots

def time_series_plot(df, col):
    df = df.groupby(['Date'])[col].sum()
    df = df.reset_index()
    #display(df)
    fig = px.line(df, x="Date", y=col, title=col)
    fig.show()

def store_dept_time_series(df, title1 = 'Time_Series'):
    fig = make_subplots(rows=5, cols=1, shared_xaxes=True,
                        subplot_titles=('Weekly_Sales', 'Temperature', 'Fuel_Price', 'CPI', 'Unemployment'),
                        vertical_spacing=0.02)

    for i, column in enumerate(['Weekly_Sales', 'Temperature', 'Fuel_Price', 'CPI', 'Unemployment']):
        if(column == 'Weekly_Sales'):
            df1 = df.groupby(['Date'])[column].sum()
            df1 = df1.reset_index()
        else:
            df1 = df.groupby(['Date'])[column].mean()
            df1 = df1.reset_index()

        df1 = df1.reset_index()
        fig.add_trace(go.Scatter(x=df1["Date"], y=df1[column]), row=i+1, col=1)

    fig.update_layout(height=1200, width=1400, title_text=title1, showlegend=False)
    fig.show()

def moving_averages(df,column1,col_no, window_size, title='Moving Averages of Time Series'):
    df1 = df[df[column1] == col_no].groupby(['Date'])['Weekly_Sales'].sum()
    df1 = df1.reset_index()

    fig = make_subplots(rows=4, cols=1, shared_xaxes=True,
                        subplot_titles=["1 week", "2 week", "4 week", str(window_size)+" week"],
                        vertical_spacing=0.1)

    for i, column in enumerate(["1 week", "2 week", "4 week", str(window_size)+" week"]):
        fig.add_trace(go.Scatter(x=df1["Date"], y=df1['Weekly_Sales'].rolling(window=i+1).mean()), row=i+1, col=1)
```

```
fig.update_layout(height=1000, width=1400, title_text=title, showlegend=False)
fig.show()

In [34]: # @title $Time-Series \ \ Total \ \ Weekly \ \ Sales$
columns_to_explore = ['Weekly_Sales','Temperature','Fuel_Price','CPI','Unemployment']
for col in columns_to_explore:
    time_series_plot(data, col)
```

Weekly\_Sales



Temperature



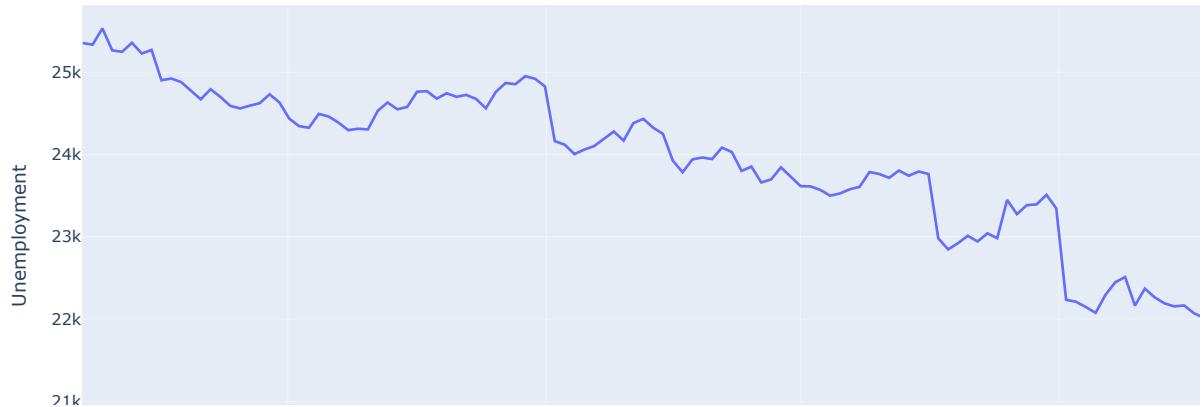
Fuel\_Price



CPI

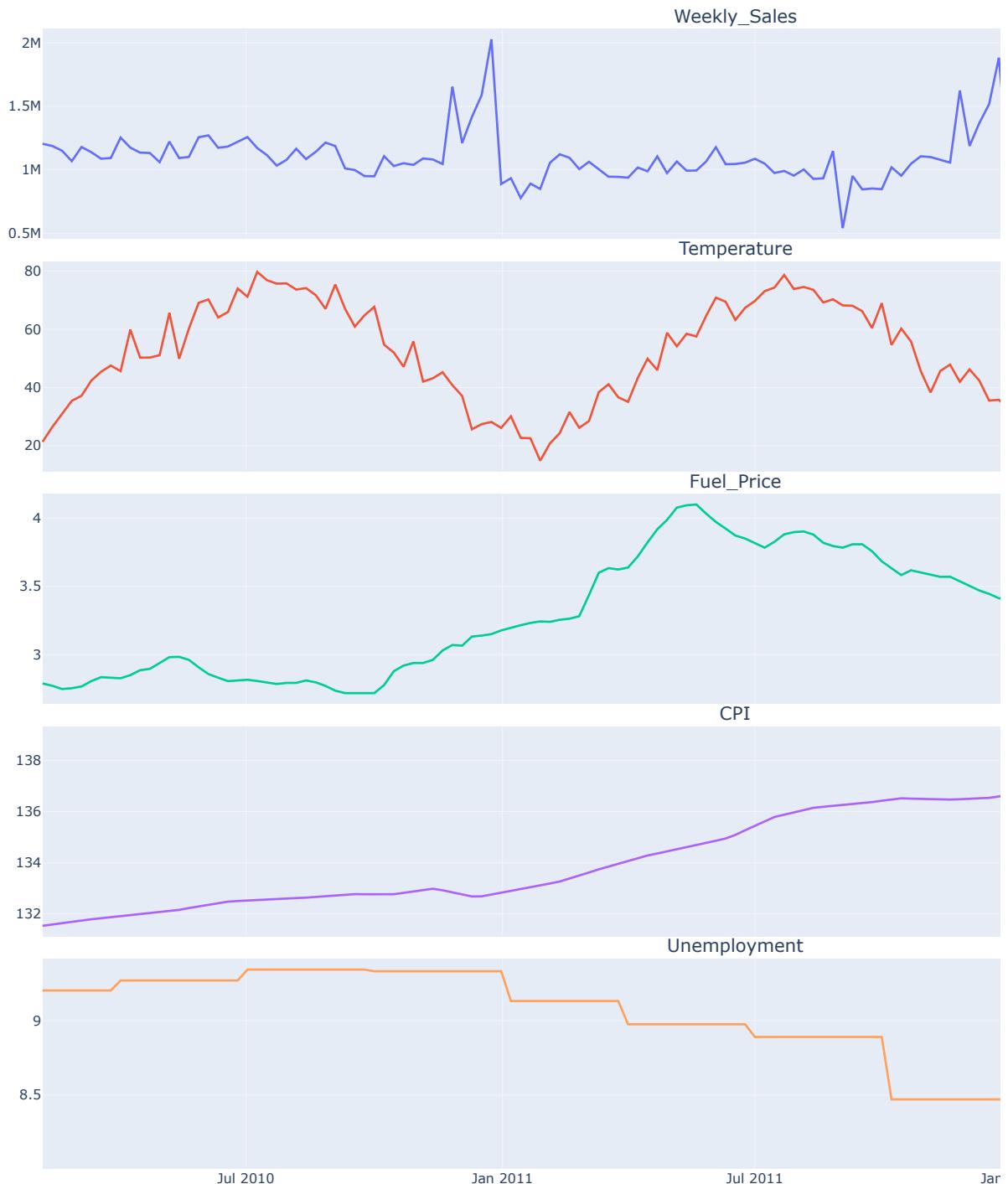


### Unemployment



```
In [35]: # @title $Store \ \ wise \ \ Time \ \ Series$ {run: "auto", "form-width": "20%"}  
store_no = 18 # @param {"type": "slider", "min": 1, "max": 45, "step": 1}  
  
data_store = data[data['Store'] == store_no]  
store_dept_time_series(data_store, title1 = 'Store ' + str(store_no) + ' Time Series')
```

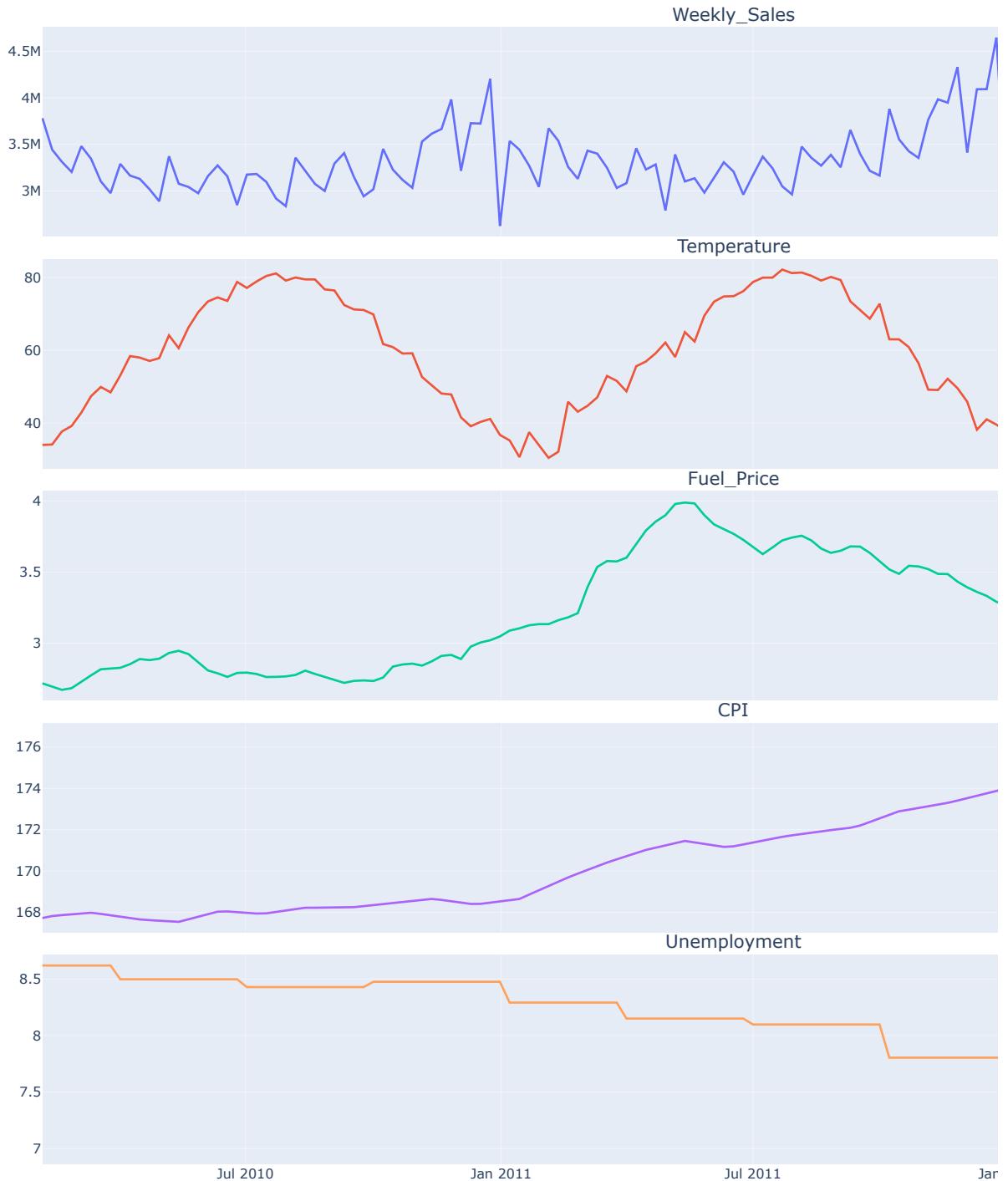
## Store 18 Time Series



```
In [36]: # @title $Department \ \ wise \ \ Time \ \ Series$ {run: "auto", "form-width": "20%"}
dept_no = 92 # @param {"type": "slider", "min": 1, "max": 99, "step": 1}

if(dept_no in list_departments):
    data_dept = data[data['Dept'] == dept_no]
    store_dept_time_series(data_dept, title1 = 'Dept ' + str(dept_no) + ' Time Series')
else:
    print(f'Department {dept_no} is not available in the dataset')
```

## Dept 92 Time Series

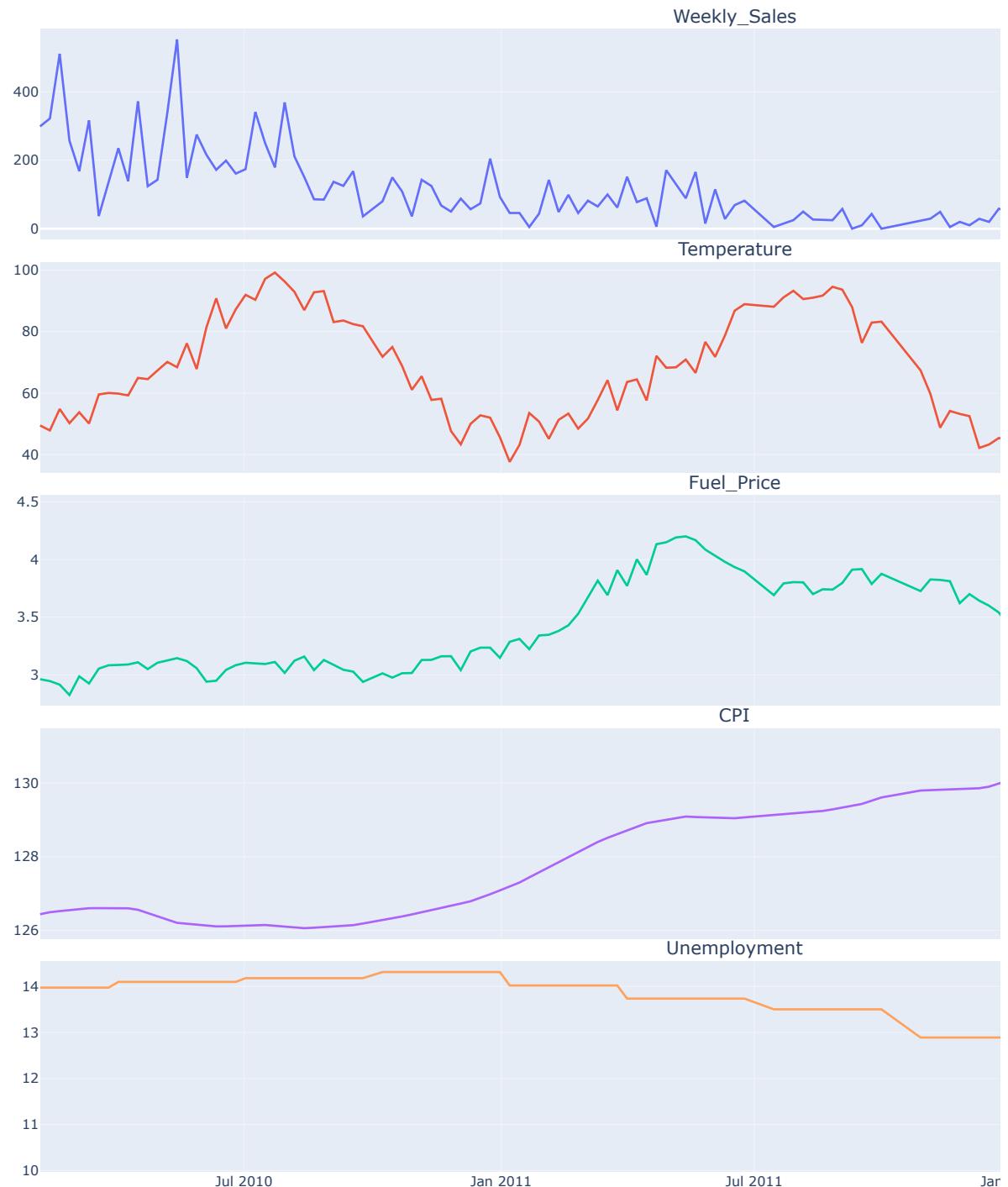


```
In [37]: # @title $Department \ \ of \ \ Store \ \ wise \ \ Time \ \ Series$ {run: "auto", "form-width": "20%"}
dept_no = 54 #@param {type:"slider", min:1, max:99, step:1}
store_no = 28 #@param {type:"slider", min:1, max:45, step:1}

if(dept_no in list_departments):
    temp_list = list(dept_not_in_stores[dept_not_in_stores['Dept'] == dept_no]['Not in following Stores'])[0]

    if(store_no in temp_list):
        print(f'Department {dept_no} is not available in Store {store_no}')
    else:
        dept_store_data = data[(data['Dept']==dept_no) & (data['Store']==store_no)]
        store_dept_time_series(dept_store_data, title1 = 'Dept '+str(dept_no)+ ' of Store '+str(store_no)+ ' Time Series')
else:
    print(f'Department {dept_no} is not available in the dataset')
```

## Dept 54 of Store 28 Time Series



```
In [38]: # @title $Store \ \ wise \ \ Moving \ \ Averages$ {run: "auto", "form-width": "20%"}
store_no = 24 # @param {"type": "slider", "min": 1, "max": 45, "step": 1}
window_size = 12 # @param {"type": "integer", "placeholder": "12"}
```

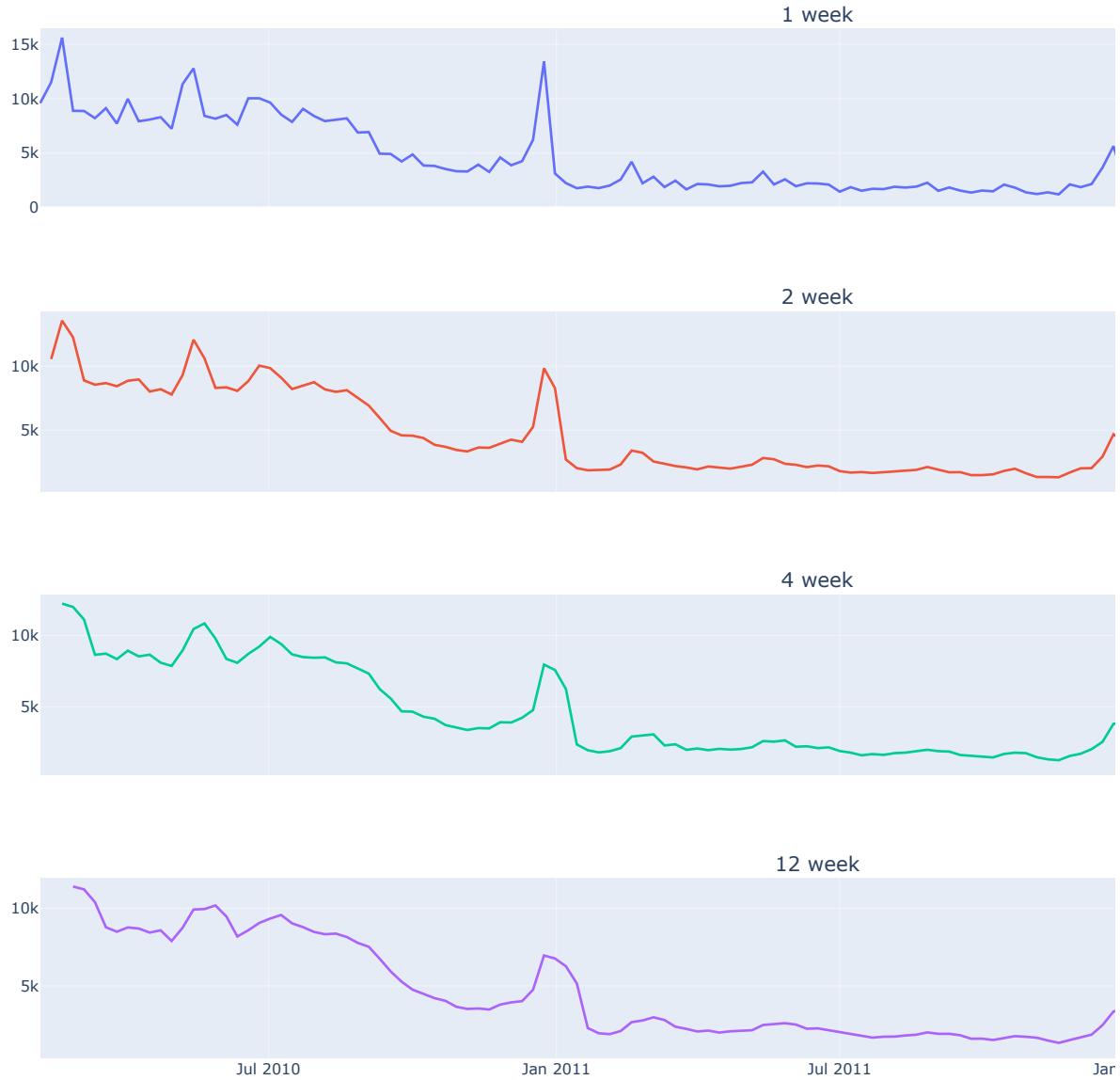
## Moving Averages : Store 24



```
In [39]: # @title $Department \ \ wise \ \ Moving \ \ Averages$ {run:"auto", "form-width":"20%"}
dept_no = 54 # @param {"type":"slider", "min":1, "max":98, "step":1}
window_size = 12 # @param {"type":"integer", "placeholder":"12"}

if(dept_no in list_departments):
    moving_averages(data,'Dept',dept_no, window_size, title = 'Moving Averages : Department '+str(dept_no))
else:
    print(f'Department {dept_no} is not available in the dataset')
```

## Moving Averages : Department 54



```
In [40]: # @title $Department \ \ of \ \ Store-Wise \ \ Moving \ \ Averages$ {run: "auto", "form-width": "20%"}
dept_no = 54 #@param {type:"slider", min:1, max:99, step:1}
store_no = 20 #@param {type:"slider", min:1, max:45, step:1}
window_size = 12 #@param {"type":"integer", "placeholder": "12"}

if(dept_no in list_departments):
    temp_list = list(dept_not_in_stores[dept_not_in_stores['Dept'] == dept_no]['Not in following Stores'])[0]

    if(store_no in temp_list):
        print(f'Department {dept_no} is not available in Store {store_no}')
    else:
        store_data = data[data['Store']==store_no]
        #store_dept_time_series(dept_store_data, title1 = 'Dept '+str(dept_no)+ ' of Store '+str(store_no)+' Time Series')
        moving_averages(store_data,'Dept',dept_no, window_size, title = 'Moving Averages : Department '+str(dept_no)+ ' of Store '+str(store_no))

else:
    print(f'Department {dept_no} is not available in the dataset')
```

## Moving Averages : Department 54 of Store 20

*Seasonal Decomposition*

```
In [41]: # @title $Function \ \ for \ \ Seasonal \ \ Decomposition$
from statsmodels.tsa.seasonal import seasonal_decompose

def seasonal_decomposition(df,column1,col_no, model_type, period, title1 = 'Seasonal Decomposition'):
    df1 = df[df[column1]==col_no].groupby(['Date'])['Weekly_Sales'].sum()
    df1 = df1.reset_index()

    tsm_decompose = seasonal_decompose(np.array(df1['Weekly_Sales']), model=model_type, period=period)

    trend = tsm_decompose.trend
    seasonal = tsm_decompose.seasonal
    residual = tsm_decompose.resid
    observed = tsm_decompose.observed

    fig = make_subplots(rows=4, cols=1, shared_xaxes=True, subplot_titles=("Observed", "Trend", "Seasonal", "Residual"), vertical
```

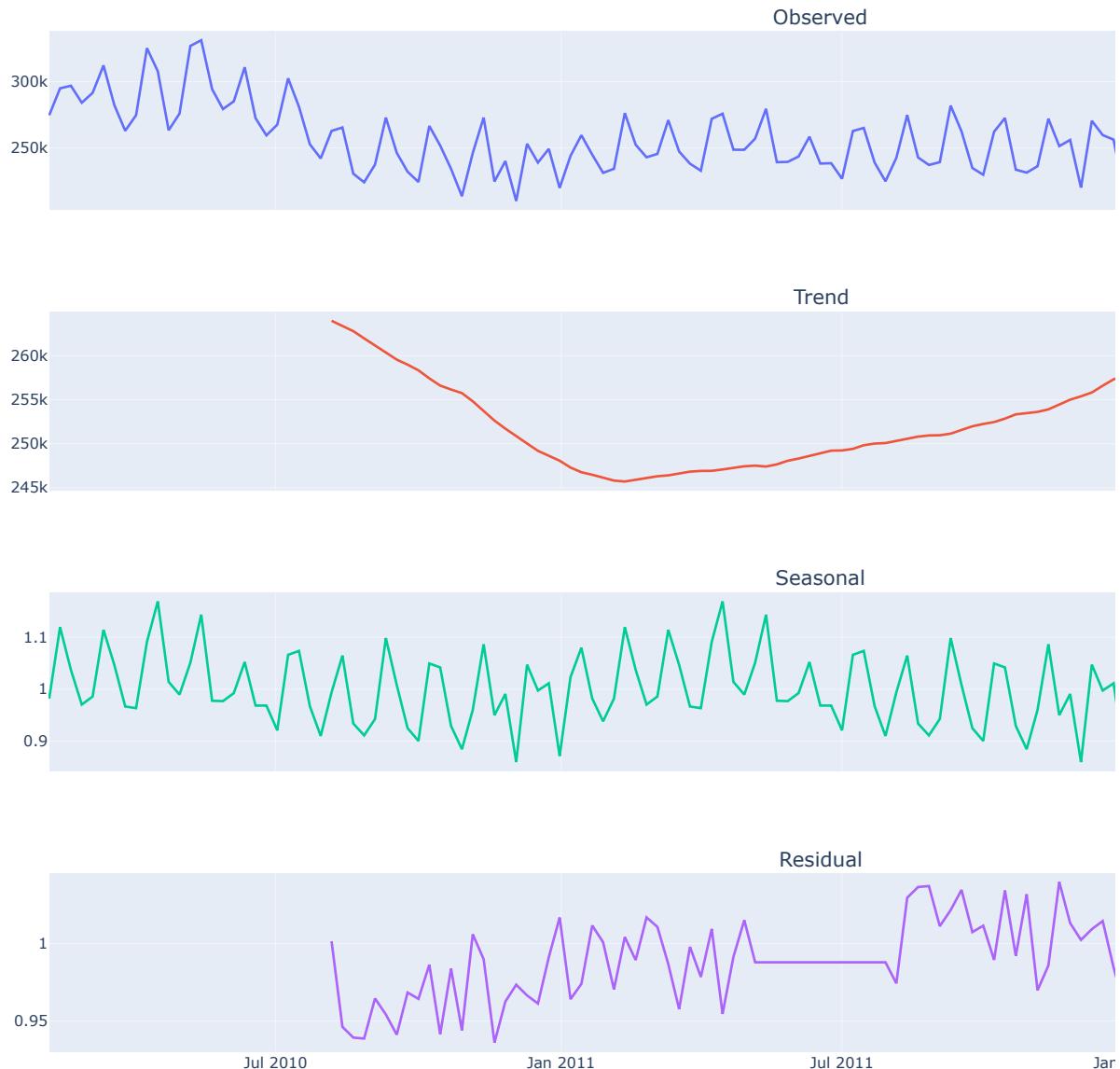
```
fig.update_layout(height=1000, width=1400, title_text=title1, showlegend=False)
fig.show()

In [42]: # @title $Store-Wise \\ Seasonal \\ Decomposition$ {run:"auto", "form-width":"20%"}

store_no = 33 #@param {type:"slider", min:1, max:45, step:1}
period = 52 #@param {type:"slider", min:1, max:52, step:1}
model_type = "multiplicative" # @param ["additive","multiplicative"]

seasonal_decomposition(data, 'Store', store_no, model_type, period, title1 = 'Seasonal Decomposition of Store '+str(store_no))
```

Seasonal Decomposition of Store 33



```
In [43]: # @title $Department-Wise \\ Seasonal \\ Decomposition$ {run:"auto", "form-width":"20%"}

dept_no = 81 # @param {"type":"slider","min":1,"max":99,"step":1}
period = 52 #@param {type:"slider", min:1, max:52, step:1}
model_type = "multiplicative" # @param ["additive","multiplicative"]

if(dept_no in list_departments):
    if(data[data['Dept']] == dept_no].groupby(['Date'])['Weekly_Sales'].sum().count() >= 2*period):
        seasonal_decomposition(data, 'Dept', dept_no, model_type, period, title1 = 'Seasonal Decomposition of Department '+str(dept_no))
    else:
        print('Not Enough Data for Seasonal Decomposition Kindly reduce the period')
else:
    print(f'Department {dept_no} is not available in the dataset')
```

## Seasonal Decomposition of Department 81



```
In [44]: # @title $Department \\ of \\ Store-wise \\ Seasonal \\ Decompose$ {run: "auto", "form-width": "20%"}
dept_no = 54 #@param {type:"slider", min:1, max:99, step:1}
store_no = 20 #@param {type:"slider", min:1, max:45, step:1}
period = 52 #@param {type:"slider", min:1, max:52, step:1}
model_type = "additive" # @param ["additive", "multiplicative"]

if(dept_no in list_departments):
    temp_list = list(dept_not_in_stores[dept_not_in_stores['Dept'] == dept_no]['Not in following Stores'])[0]

    if(store_no in temp_list):
        print(f'Department {dept_no} is not available in Store {store_no}')
    else:
        store_data = data[data['Store']==store_no]
        #store_dept_time_series(dept_store_data, title1 = 'Dept '+str(dept_no)+ ' of Store '+str(store_no)+ ' Time Series')
        seasonal_decomposition(store_data, 'Dept', dept_no, model_type, period, title1 = 'Seasonal Decomposition : Department '+str(dept_no))

else:
    print(f'Department {dept_no} is not available in the dataset')
```

## Seasonal Decomposition : Department 54 of Store 20

