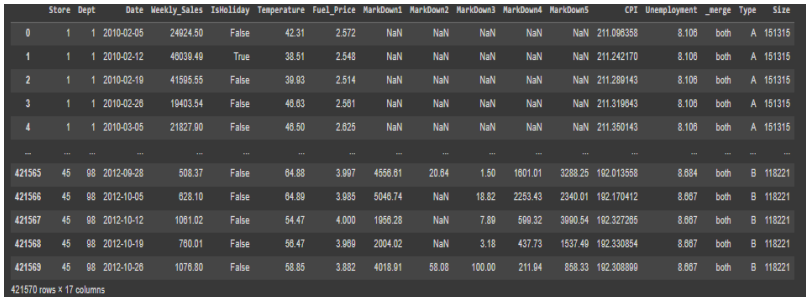


Data Collection and Preprocessing Phase

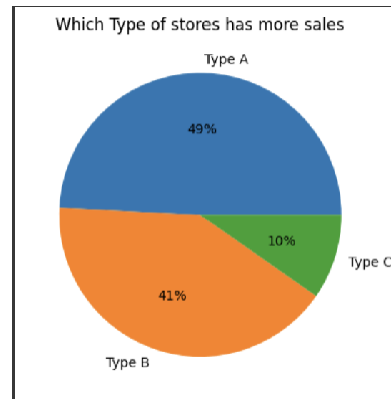
Date	10-july-2024
Team ID	739969
Project Title	Walmart Sales Analysis For Retail Industry
Maximum Marks	6 Marks

Data Exploration and Preprocessing Report

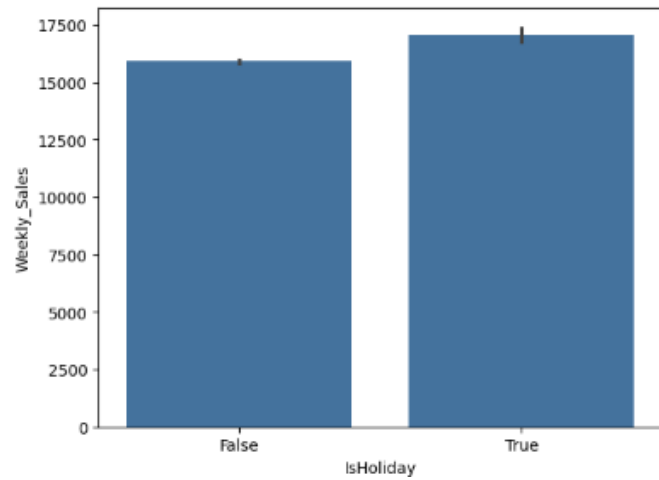
Dataset variables will be statistically analyzed to identify patterns and outliers, with Python employed for preprocessing tasks like normalization and feature engineering. Data cleaning will address missing values and outliers, ensuring quality for subsequent analysis and modeling, and forming a strong foundation for insights and predictions.

Section	Description
Data Overview	<p><u>Dimension:</u> 421570rows x 17 Columns</p> <p><u>Descriptive statistics:</u></p>  <p>421570 rows x 17 columns</p>

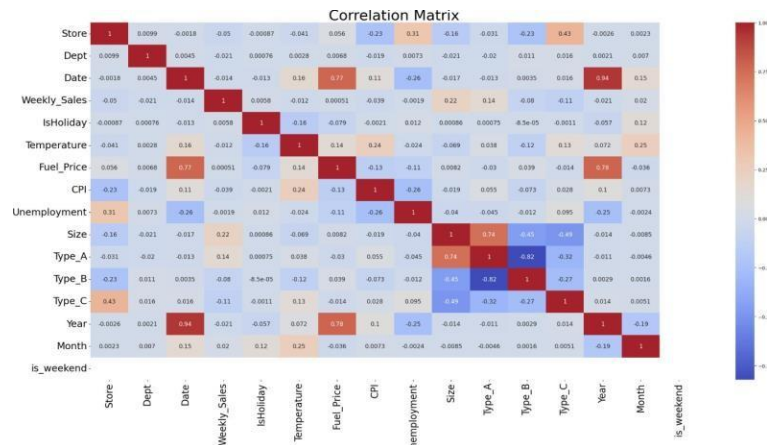
Univariate analysis



Bivariate analysis



Multivariate analysis



Outliers and Anomalies

-

Data Preprocessing Code Screenshots

Loading Data

```
[ ] train= pd.read_csv('/content/train.csv (1).zip')
store= pd.read_csv('/content/stores.csv')
features= pd.read_csv('/content/features.csv (1).zip')
```

train.head()

	Store	Dept	Date	Weekly_Sales	IsHoliday
0	1	1	2010-02-05	24924.50	False
1	1	1	2010-02-12	46039.49	True
2	1	1	2010-02-19	41595.55	False
3	1	1	2010-02-26	19403.54	False
4	1	1	2010-03-05	21827.90	False

store.head()

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

[] features.head()

	Store	Date	Temperature	Fuel_Price	MarkDown1	MarkDown2	MarkDown3	MarkDown4	MarkDown5	CPI	Unemployment	IsHoliday
0	1	2010-02-05	42.31	2.572	NaN	NaN	NaN	NaN	NaN	211.096358	8.106	False
1	1	2010-02-12	38.51	2.548	NaN	NaN	NaN	NaN	NaN	211.242170	8.106	True
2	1	2010-02-19	39.93	2.514	NaN	NaN	NaN	NaN	NaN	211.289143	8.106	False
3	1	2010-02-26	46.63	2.561	NaN	NaN	NaN	NaN	NaN	211.319643	8.106	False
4	1	2010-03-05	46.50	2.625	NaN	NaN	NaN	NaN	NaN	211.350143	8.106	False

data2.describe()

	Store	Dept	Weekly_Sales	Temperature	Fuel_Price	CPI	Unemployment	Size
count	421570.000000	421570.000000	421570.000000	421570.000000	421570.000000	421570.000000	421570.000000	421570.000000
mean	22.200546	44.260317	15981.258123	60.090069	3.361027	171.201947	7.960289	136727.915739
std	12.785297	30.492054	22711.183519	18.447931	0.458515	39.159276	1.863296	60980.583328
min	1.000000	1.000000	-4988.940000	-2.060000	2.472000	126.064000	3.879000	34875.000000
25%	11.000000	18.000000	2079.650000	46.680000	2.933000	132.022667	6.891000	93638.000000
50%	22.000000	37.000000	7612.030000	62.090000	3.452000	182.318780	7.866000	140167.000000
75%	33.000000	74.000000	20205.852500	74.280000	3.738000	212.416993	8.572000	202505.000000
max	45.000000	99.000000	693099.360000	100.140000	4.468000	227.232807	14.313000	219622.000000

[] data3=data2.loc[data2['Weekly_Sales']>=0]
data3.describe()

	Store	Dept	Weekly_Sales	Temperature	Fuel_Price	CPI	Unemployment	Size
count	420285.000000	420285.000000	420285.000000	420285.000000	420285.000000	420285.000000	420285.000000	420285.000000
mean	22.195477	44.242771	16030.329773	60.090474	3.360888	171.212152	7.960077	136749.569176
std	12.787213	30.507197	22728.500149	18.448260	0.458523	39.162280	1.863873	60992.688568
min	1.000000	1.000000	0.000000	-2.060000	2.472000	126.064000	3.879000	34875.000000
25%	11.000000	18.000000	2117.560000	46.680000	2.933000	132.022667	6.891000	93638.000000
50%	22.000000	37.000000	7659.090000	62.090000	3.452000	182.350989	7.866000	140167.000000
75%	33.000000	74.000000	20268.380000	74.280000	3.738000	212.445487	8.567000	202505.000000
max	45.000000	99.000000	693099.360000	100.140000	4.468000	227.232807	14.313000	219622.000000

Handling Negative Data

Data Transformation

```
[ ] if 'Type' in data9.columns:
    data9 = pd.get_dummies(data9, columns=['Type'])
else:
    print("Column 'Type' does not exist. It might have been already one-hot encoded.")
```

```
[ ] data9['Date'] = pd.to_datetime(data9['Date'])
```

```
[ ] data9['Year'] = data9['Date'].dt.year
    data9['Month'] = data9['Date'].dt.month
```

```
data9[['Date', 'Month', 'Year']].head()
```

	Date	Month	Year
0	2010-02-05	2	2010
3	2010-02-26	2	2010
4	2010-03-05	3	2010
5	2010-03-12	3	2010
6	2010-03-19	3	2010

```
[ ] data9['Dayofweek_name'] = data9['Date'].dt.day_name()
    data9[['Date', 'Dayofweek_name']].head()
```

	Date	Dayofweek_name
0	2010-02-05	Friday
3	2010-02-26	Friday
4	2010-03-05	Friday
5	2010-03-12	Friday
6	2010-03-19	Friday

```
[ ] data9['is_weekend'] = np.where(data9['Dayofweek_name'].isin(['Saturday', 'Sunday']), 1, 0)
```

```
[ ] data9['IsHoliday'] = data9['IsHoliday'].astype(int)
    del data9['Dayofweek_name']
```

```
[ ] data9['Type_A'] = data9['Type_A'].astype(int)
    data9['Type_B'] = data9['Type_B'].astype(int)
    data9['Type_C'] = data9['Type_C'].astype(int)
```

```
print(data9.head())
```

	Store	Dept	Date	Weekly_Sales	IsHoliday	Temperature	Fuel_Price	\
0	1	1	2010-02-05	24924.50	0	42.31	2.572	
3	1	1	2010-02-26	19403.54	0	46.63	2.561	
4	1	1	2010-03-05	21827.90	0	46.50	2.625	
5	1	1	2010-03-12	21043.39	0	57.79	2.667	
6	1	1	2010-03-19	22136.64	0	54.58	2.720	

	CPI	Unemployment	Size	Type_A	Type_B	Type_C	Year	Month	\
0	211.096358	8.106	151315	1	0	0	2010	2	
3	211.319643	8.106	151315	1	0	0	2010	2	
4	211.350143	8.106	151315	1	0	0	2010	3	
5	211.380643	8.106	151315	1	0	0	2010	3	
6	211.215635	8.106	151315	1	0	0	2010	3	

	is_weekend
0	0
3	0
4	0
5	0
6	0

Feature Engineering

Attached the codes in final submission.

Save Processed Data

-