PROJECT-1

EXPLORATORY ANALYSIS

DATASET: Supermarket store branches sales analysis

DATASET-

In the dataset, You'll get data of different stores of a supermarket company as per their store IDs which for ease has been converted to positive integers.

1)Store ID: (Index) ID of the particular store.

2)Store Area: Physical Area of the store in yard square.

3)Items Available: Number of different items available in the corresponding store.

4)Daily Customer Count: Number of customers who visited to stores on an average over month.

5)Store Sales: Sales in (US $) that stores made.

PROBLEM:-

Analysing the performances of stores in the past on basis of which will try to rectify defects as well as to leverage the positives.

WHY I CHOOSE?

A supermarket is a self-service shop offering a wide variety of food, beverages and household products, organized into sections. This kind of store is larger and has a wider selection than earlier grocery stores, but is smaller and more limited in the range of merchandise than a hypermarket or big-box market. All things considered super market is easy and well categorised shop than some other road-side markets.

Data preparation :

As there are no null values present in my dataset.
There is no need to replace or re-organize any of my attribute.

```
In [7]: data.isnull().sum()
        #To check whether their are any null values present in the given dataset.

Out[7]: Store ID              0
        Store_Area            0
        Items_Available       0
        Daily_Customer_Count  0
        Store_Sales           0
        dtype: int64
```

Gathering the information of the attributes based on the data respectively

```
#Gives detailed information about the dataset
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 896 entries, 0 to 895
Data columns (total 5 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Store ID              896 non-null    int64
 1   Store_Area            896 non-null    int64
 2   Items_Available       896 non-null    int64
 3   Daily_Customer_Count  896 non-null    int64
 4   Store_Sales           896 non-null    int64
dtypes: int64(5)
memory usage: 35.1 KB
```

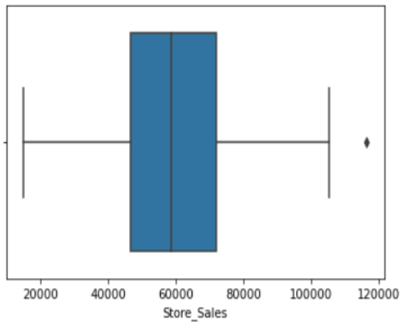
Doing the outlier analysis

Outlier analysis is carried forward by plotting a boxplot. Inspite of having the outliers it does not make any noticeable changes to the approach. So, we can even ignore the outliers of the attributes as it does not possess any significant change to the analysis.

```
#plotting the dataset for outliers
sns.boxplot(data['Store_Sales'])
```

```
C:\Users\Ponna Dhanush\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a k
eyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an exp
licit keyword will result in an error or misinterpretation.
  warnings.warn(
```

```
<AxesSubplot:xlabel='Store_Sales'>
```



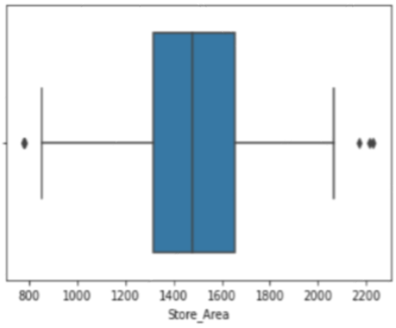

```
#finding the outliers
np.where(data['Store_Sales']>110000)
```

```
(array([649], dtype=int64),)
```

```
#plotting the dataset for outliers
sns.boxplot(data['Store_Area'])
```

```
C:\Users\Ponna Dhanush\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

```
<AxesSubplot:xlabel='Store_Area'>
```




```
#finding the outliers
np.where(data['Store_Area']<800)
```

```
(array([158, 865], dtype=int64),)
```

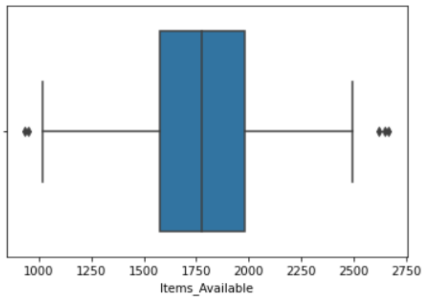
```
#finding the outliers
np.where(data['Store_Area']>2000)
```

```
(array([ 61,  91, 163, 243, 258, 311, 344, 398, 466, 469, 540, 550, 567,
        628, 784, 798, 849], dtype=int64),)
```

```
#plotting the dataset for outliers
sns.boxplot(data['Items_Available'])
```

```
C:\Users\Ponna Dhanush\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

```
<AxesSubplot:xlabel='Items_Available'>
```



```
#finding the outliers
np.where(data['Items_Available']<1000)
```

```
(array([158, 865], dtype=int64),)
```

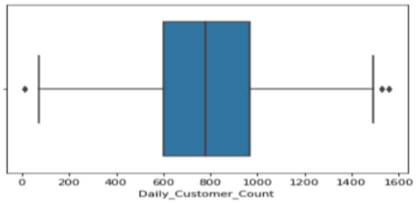
```
#finding the outliers
np.where(data['Items_Available']>2500)
```

```
(array([ 91, 466, 540], dtype=int64),)
```

```
#plotting the dataset for outliers
sns.boxplot(data['Daily_Customer_Count'])
```

```
C:\Users\Ponna Dhanush\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a k
eyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an exp
licit keyword will result in an error or misinterpretation.
  warnings.warn(
```

```
<AxesSubplot:xlabel='Daily_Customer_Count'>
```




```
#finding the outliers
np.where(data["Daily_Customer_Count"]<50)
```

```
(array([39], dtype=int64),)
```

```
#finding the outliers
np.where(data['Daily_Customer_Count']>1500)
```

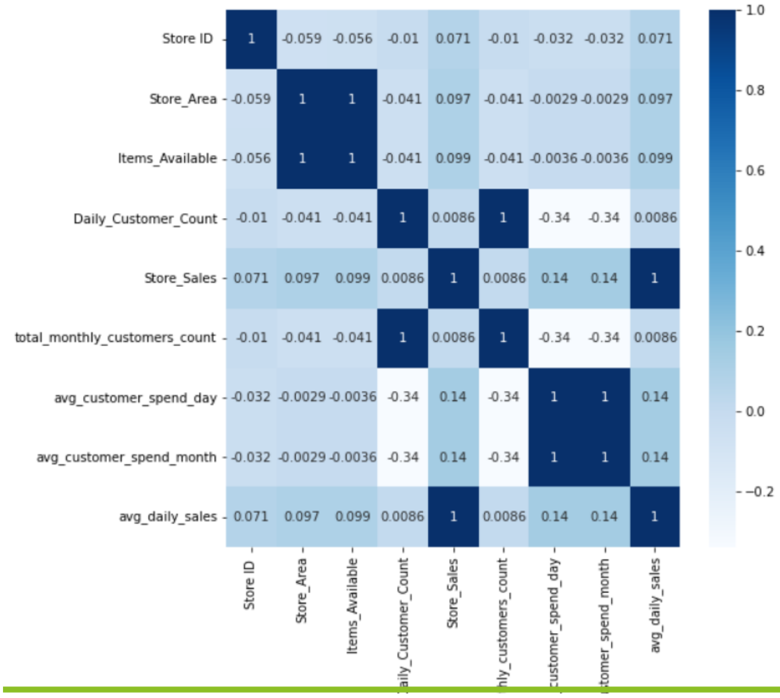
```
(array([349, 848], dtype=int64),)
```

Exploratory analysis:

Plotting a heatmap showing the correlation between the attributes.

```
plt.figure(figsize=(8,8))
sns.heatmap(data.corr(), annot=True, cmap='Blues')
```

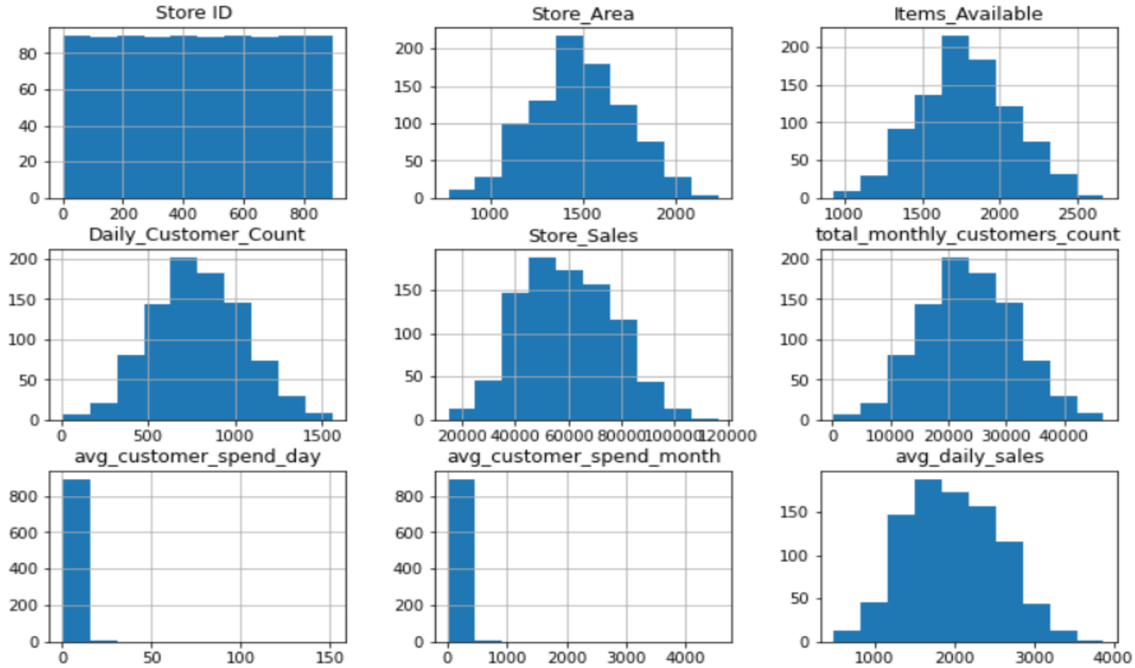
```
<AxesSubplot:>
```



We see high correlation between Items Available and Store Area, monthly customer count with Daily Customer Count, Store Sales and Avg Daily Sales, Avg Customer Spend Day with Avg Customer Spend Month (i.e., it’s almost about 1)

- **Plotting and visualizing the data in the form of a histogram.**

```
data.hist(figsize=(12,8));
plt.grid(False)
#plotiing histograms with the help of grid lines for different attributes of the dataset.
```

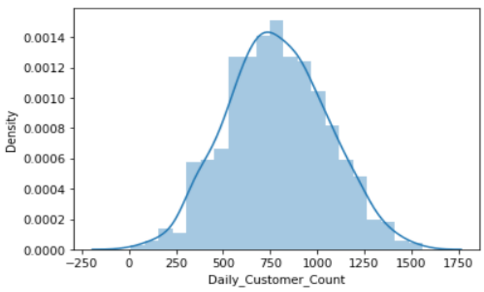


- We get to observe that daily customer count is mostly lies between (600-800) heads.

```
sns.distplot(data['Daily_Customer_Count'])
#we are visually representing the attribute Daily_Customer_Count.
```

```
C:\Users\Ponna Dhanush\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated fun
ction and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with sim
ilar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

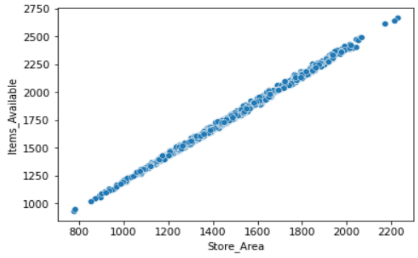
```
<AxesSubplot:xlabel='Daily_Customer_Count', ylabel='Density'>
```



- As we can see that the graph between Store Area and Items Available is steadily increasing, which is clearly evident that store area plays a major role in items available for any particular branch.

```python
sns.scatterplot(x='Store_Area',y='Items_Available',data=data)
#we are finding the relation between two attributes "Store_Area " and "Items_Available" of the dataset using scatterplot.
```

```
<AxesSubplot:xlabel='Store_Area', ylabel='Items_Available'>
```

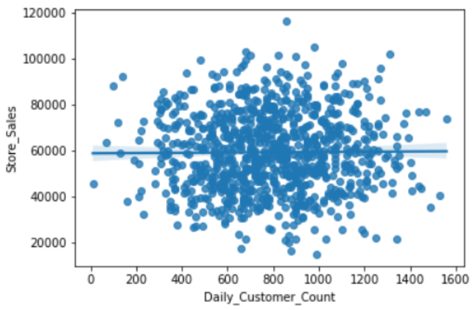


- It is clear that even the daily customer count increases ,store sales remain consistent (i.e. it’s about 60,000$)

```python
sns.regplot("Daily_Customer_Count", "Store_Sales",data=data)
#here we are identifying how "Daily_Customer_Count" is effecting the "Store_Sales".
```

```
C:\Users\Ponna Dhanush\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as ke
yword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an
explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

```
<AxesSubplot:xlabel='Daily_Customer_Count', ylabel='Store_Sales'>
```



- Here we are building some statistical logics using "Daily Customer Count" and "Store Sales" to find "Average customer spend month" and "avg daily sales".

```
data['total_monthly_customers_count'] = data['Daily_Customer_Count'] * 30
data['avg_customer_spend_day'] = data['Store_Sales'] / data['total_monthly_customers_count']
data['avg_customer_spend_month'] = data['Store_Sales'] / data['Daily_Customer_Count']
data['avg_daily_sales'] = data['Store_Sales'] / 30

#here we are building some logics using "Daily_Custoomer_Count" and "Store_Sales" to find "Average_customer_spend_month" and "avg
```

```
correlation = data.corr()
print(correlation['Store_Sales'].sort_values(ascending = False),'\n')
#here we are finding the correlation of attribute "Store_Sales".
```

```
avg_daily_sales                1.000000
Store_Sales                    1.000000
avg_customer_spend_day         0.139546
avg_customer_spend_month       0.139546
Items_Available                0.098849
Store_Area                     0.097474
Store ID                       0.071486
Daily_Customer_Count           0.008629
total_monthly_customers_count  0.008629
Name: Store_Sales, dtype: float64
```

- **Finally, I want to conclude that even there is raise in daily customer count the total store sales are consistent.**

```
Yfac = data["Store_Sales"]
Xfac = data["Daily_Customer_Count"]
plt.figure(figsize=(6,6))
plt.scatter(Xfac,Yfac,s=15)
plt.xlabel('Count of Customers')
plt.ylabel('Sales in $')
plt.title('Customers & Sales of All Stores')
plt.xlim(200, 2000)
plt.yscale("linear")
plt.grid(True)
```

At last, we can draw this analysis by saying that most of the customers are just going to the store and returning with empty hands

which leads to increase in daily customer count and consistent store sales which are bought by regular customers.

Advice: -

We should introduce a new innovative idea which leads to high store sales like “it is mandatory for every customer to buy at least one product from store.