Group Exercise 2 (Ungraded)

Data Importing

In [1]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.preprocessing import StandardScaler
```

Importing datasets

In [3]:

```
retail_orders = pd.read_csv("Retail_Data_Orders_W23.csv") # Imported Retail_Data_Or
retail_data = pd.read_csv("Retail_Data_W23.csv") # Imported Retail_Data_W23.csv
retail_unlabelled_data = pd.read_csv("Retail_Unlabeled_Data.csv") # Imported Retail
store_data = pd.read_csv("Store.csv") # Imported Store.csv


retail_orders_df = pd.DataFrame(retail_orders) # Created a dataframe for retail_ord
retail_data_df = pd.DataFrame(retail_data) # Created a dataframe for retail_data.cs
retail_unlabelled_data_df = pd.DataFrame(retail_unlabelled_data) # Created a datafr
store_data_df = pd.DataFrame(store_data) # Created a dataframe for store_data.csv
```

Exploratory Data Analysis

In [4]:

```
retail_orders_df.head() # Top 5 rows of the dataframe
retail_orders_df.shape # Shape of the dataframe
```

Out[4]:

```
(651013, 3)
```

In [6]:

```
retail_data_df.head() # Top 5 rows of the dataframe
retail_data_df.shape # Shape of the dataframe
```

Out[6]:

```
(651013, 10)
```

In [7]:

```
retail_unlabelled_data_df.head() # Top 5 rows of the dataframe
retail_unlabelled_data_df.shape # Shape of the dataframe
```

Out[7]:

```
(162754, 10)
```

In [8]:

```
store_data_df.head() # Top 5 rows of the dataframe
store_data_df.shape # Shape of the dataframe
```

Out[8]:

```
(1115, 10)
```

Cleaning and Pre-processing

Our first step is to merge the three datasets into one and clean them.

Data sets to merge:
1. retail_orders_df
2. retail_data_df
3. store_data_df

In [9]:

```python
merged_1 = retail_data_df.merge(retail_orders_df, on="Id",how="inner")
```

In [10]:

```python
merged_1.head()
```

Out[10]:

	Unnamed: 0_x	Store	DayOfWeek	Date	Customers	Open	Promo	StateHoliday	SchoolHoliday
0	339662	516	5	2014-09-12	665	1	0	0	0
1	969929	665	2	2013-02-12	1213	1	0	0	0
2	499245	511	4	2014-04-10	685	1	0	0	1
3	581706	462	7	2014-01-26	0	0	0	0	0
4	618037	1113	3	2013-12-25	0	0	0	c	1

In [11]:

```python
merged_final = merged_1.merge(store_data_df,on="Store",how="left")
```

In [12]:

```python
merged_final.head()
```

Out[12]:

	Unnamed: 0_x	Store	DayOfWeek	Date	Customers	Open	Promo	StateHoliday	SchoolHoliday
0	339662	516	5	2014-09-12	665	1	0	0	0
1	969929	665	2	2013-02-12	1213	1	0	0	0
2	499245	511	4	2014-04-10	685	1	0	0	1
3	581706	462	7	2014-01-26	0	0	0	0	0
4	618037	1113	3	2013-12-25	0	0	0	c	1

5 rows × 21 columns

In [14]:

```python
merged_final.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 651013 entries, 0 to 651012
Data columns (total 21 columns):
 #   Column                     Non-Null Count   Dtype  
---  ------                     --------------   -----  
 0   Unnamed: 0_x               651013 non-null  int64  
 1   Store                      651013 non-null  int64  
 2   DayOfWeek                  651013 non-null  int64  
 3   Date                       651013 non-null  object 
 4   Customers                  651013 non-null  int64  
 5   Open                       651013 non-null  int64  
 6   Promo                      651013 non-null  int64  
 7   StateHoliday               651013 non-null  object 
 8   SchoolHoliday              651013 non-null  int64  
 9   Id                         651013 non-null  int64  
 10  Unnamed: 0_y               651013 non-null  int64  
 11  Orders                     651013 non-null  int64  
 12  StoreType                  651013 non-null  object 
 13  Assortment                 651013 non-null  object 
 14  CompetitionDistance        649312 non-null  float64
 15  CompetitionOpenSinceMonth  443851 non-null  float64
 16  CompetitionOpenSinceYear   443851 non-null  float64
 17  Promo2                     651013 non-null  int64  
 18  Promo2SinceWeek            325567 non-null  float64
 19  Promo2SinceYear            325567 non-null  float64
 20  PromoInterval              325567 non-null  object 
dtypes: float64(5), int64(11), object(5)
memory usage: 109.3+ MB
```

In [15]:
```
merged_final = merged_final.drop(['Unnamed: 0_y','Unnamed: 0_x'],axis=1)
```

In [17]:
```
merged_final.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 651013 entries, 0 to 651012
Data columns (total 19 columns):
 #   Column                     Non-Null Count   Dtype  
---  ------                     --------------   -----  
 0   Store                      651013 non-null  int64  
 1   DayOfWeek                  651013 non-null  int64  
 2   Date                       651013 non-null  object 
 3   Customers                  651013 non-null  int64  
 4   Open                       651013 non-null  int64  
 5   Promo                      651013 non-null  int64  
 6   StateHoliday               651013 non-null  object 
 7   SchoolHoliday              651013 non-null  int64  
 8   Id                         651013 non-null  int64  
 9   Orders                     651013 non-null  int64  
 10  StoreType                  651013 non-null  object 
 11  Assortment                 651013 non-null  object 
 12  CompetitionDistance        649312 non-null  float64
 13  CompetitionOpenSinceMonth  443851 non-null  float64
 14  CompetitionOpenSinceYear   443851 non-null  float64
 15  Promo2                     651013 non-null  int64  
 16  Promo2SinceWeek            325567 non-null  float64
 17  Promo2SinceYear            325567 non-null  float64
 18  PromoInterval              325567 non-null  object 
dtypes: float64(5), int64(9), object(5)
memory usage: 99.3+ MB
```

In [18]:
```
merged_final.isnull().sum()
```

Out[18]:

```
Store                             0
DayOfWeek                         0
Date                              0
Customers                         0
Open                              0
Promo                             0
StateHoliday                      0
SchoolHoliday                     0
Id                                0
Orders                            0
StoreType                         0
Assortment                        0
CompetitionDistance            1701
CompetitionOpenSinceMonth    207162
CompetitionOpenSinceYear     207162
Promo2                            0
Promo2SinceWeek              325446
Promo2SinceYear              325446
PromoInterval                325446
dtype: int64
```

In [19]:

```
merged_final.CompetitionDistance.describe()
```

Out[19]:

```
count    649312.000000
mean       5436.342390
std        7713.881629
min          20.000000
25%         710.000000
50%        2330.000000
75%        6890.000000
max       75860.000000
Name: CompetitionDistance, dtype: float64
```

In [20]:

```
merged_final['CompetitionDistance'].fillna(merged_final['CompetitionDistance'].medi
```

In [22]:

```
merged_final['CompetitionOpenSinceMonth'].fillna(0, inplace=True)
merged_final['CompetitionOpenSinceYear'].fillna(0, inplace=True)
```

In [24]:

```
merged_final['Promo2SinceWeek'].fillna(0, inplace=True)
merged_final['Promo2SinceYear'].fillna(0, inplace=True)
```

In [25]:

```
merged_final['Promo2SinceWeek'].fillna(0, inplace=True)
merged_final['Promo2SinceYear'].fillna(0, inplace=True)
```

In [26]:

```
merged_final['PromoInterval'].fillna('PromoUnavailable', inplace=True)
```

In [27]:

```
merged_final.isnull().sum()
```

Out[27]:

```
Store                        0
DayOfWeek                    0
Date                         0
Customers                    0
Open                         0
Promo                        0
StateHoliday                 0
SchoolHoliday                0
Id                           0
Orders                       0
StoreType                    0
Assortment                   0
CompetitionDistance          0
CompetitionOpenSinceMonth    0
CompetitionOpenSinceYear     0
Promo2                       0
Promo2SinceWeek              0
Promo2SinceYear              0
PromoInterval                0
dtype: int64
```

In [28]:

```
merged_final.describe()
```

Out[28]:

	Store	DayOfWeek	Customers	Open	Promo	SchoolHoliday
count	651013.000000	651013.000000	651013.000000	651013.000000	651013.000000	651013.000000
mean	558.645629	3.999336	632.851832	0.829619	0.381558	0.178927
std	321.905872	1.998260	464.857658	0.375967	0.485769	0.383292
min	1.000000	1.000000	0.000000	0.000000	0.000000	0.000000
25%	281.000000	2.000000	404.000000	1.000000	0.000000	0.000000
50%	558.000000	4.000000	609.000000	1.000000	0.000000	0.000000
75%	838.000000	6.000000	837.000000	1.000000	1.000000	0.000000
max	1115.000000	7.000000	5458.000000	1.000000	1.000000	1.000000

In [29]:

```
merged_final.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 651013 entries, 0 to 651012
Data columns (total 19 columns):
 #   Column                     Non-Null Count   Dtype  
---  ------                     --------------   -----  
 0   Store                      651013 non-null  int64  
 1   DayOfWeek                  651013 non-null  int64  
 2   Date                       651013 non-null  object 
 3   Customers                  651013 non-null  int64  
 4   Open                       651013 non-null  int64  
 5   Promo                      651013 non-null  int64  
 6   StateHoliday               651013 non-null  object 
 7   SchoolHoliday              651013 non-null  int64  
 8   Id                         651013 non-null  int64  
 9   Orders                     651013 non-null  int64  
 10  StoreType                  651013 non-null  object 
 11  Assortment                 651013 non-null  object 
 12  CompetitionDistance        651013 non-null  float64
 13  CompetitionOpenSinceMonth  651013 non-null  float64
 14  CompetitionOpenSinceYear   651013 non-null  float64
 15  Promo2                     651013 non-null  int64  
 16  Promo2SinceWeek            651013 non-null  float64
 17  Promo2SinceYear            651013 non-null  float64
 18  PromoInterval              651013 non-null  object 
dtypes: float64(5), int64(9), object(5)
memory usage: 99.3+ MB
```

In [32]:
```
merged_final.CompetitionDistance = merged_final.CompetitionDistance.astype('int64')
merged_final.CompetitionOpenSinceMonth = merged_final.CompetitionOpenSinceMonth.ast
merged_final.CompetitionOpenSinceYear = merged_final.CompetitionOpenSinceYear.astyp
merged_final.Promo2SinceWeek = merged_final.Promo2SinceWeek.astype('int').astype('o
merged_final.Promo2SinceYear = merged_final.Promo2SinceYear.astype('int').astype('o
merged_final.DayOfWeek = merged_final.DayOfWeek.astype('int').astype('object')
```

In [33]:
```
merged_final.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 651013 entries, 0 to 651012
Data columns (total 19 columns):
 #   Column                     Non-Null Count   Dtype 
---  ------                     --------------   ----- 
 0   Store                      651013 non-null  int64 
 1   DayOfWeek                  651013 non-null  object
 2   Date                       651013 non-null  object
 3   Customers                  651013 non-null  int64 
 4   Open                       651013 non-null  int64 
 5   Promo                      651013 non-null  int64 
 6   StateHoliday               651013 non-null  object
 7   SchoolHoliday              651013 non-null  int64 
 8   Id                         651013 non-null  int64 
 9   Orders                     651013 non-null  int64 
 10  StoreType                  651013 non-null  object
 11  Assortment                 651013 non-null  object
 12  CompetitionDistance        651013 non-null  int64 
 13  CompetitionOpenSinceMonth  651013 non-null  object
 14  CompetitionOpenSinceYear   651013 non-null  object
 15  Promo2                     651013 non-null  int64 
 16  Promo2SinceWeek            651013 non-null  object
 17  Promo2SinceYear            651013 non-null  object
 18  PromoInterval              651013 non-null  object
dtypes: int64(9), object(10)
memory usage: 99.3+ MB
```

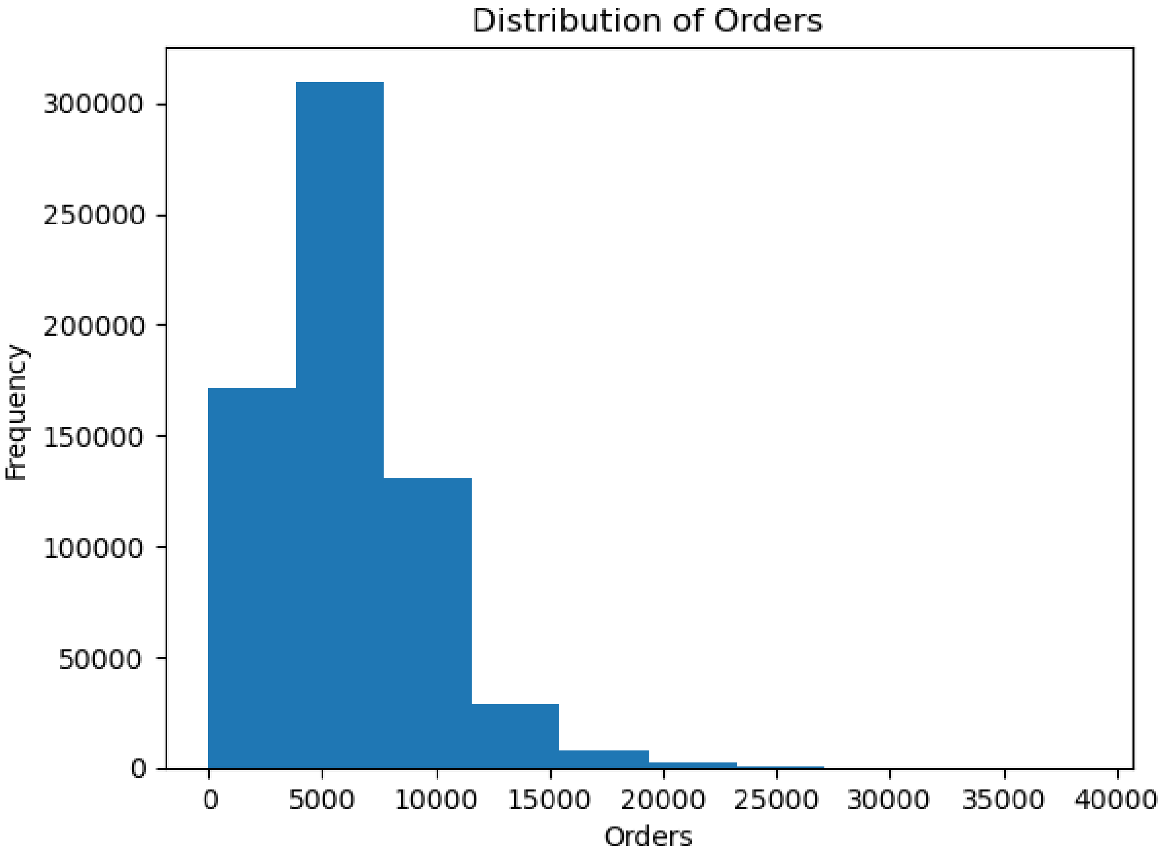
In [34]:

```
merged_final.drop(['Date'],axis = 1,inplace = True)
```

In [ ]:

In [35]:

```
plt.hist(merged_final['Orders'])
plt.xlabel('Orders')
plt.ylabel('Frequency')
plt.title('Distribution of Orders')
plt.show()
```

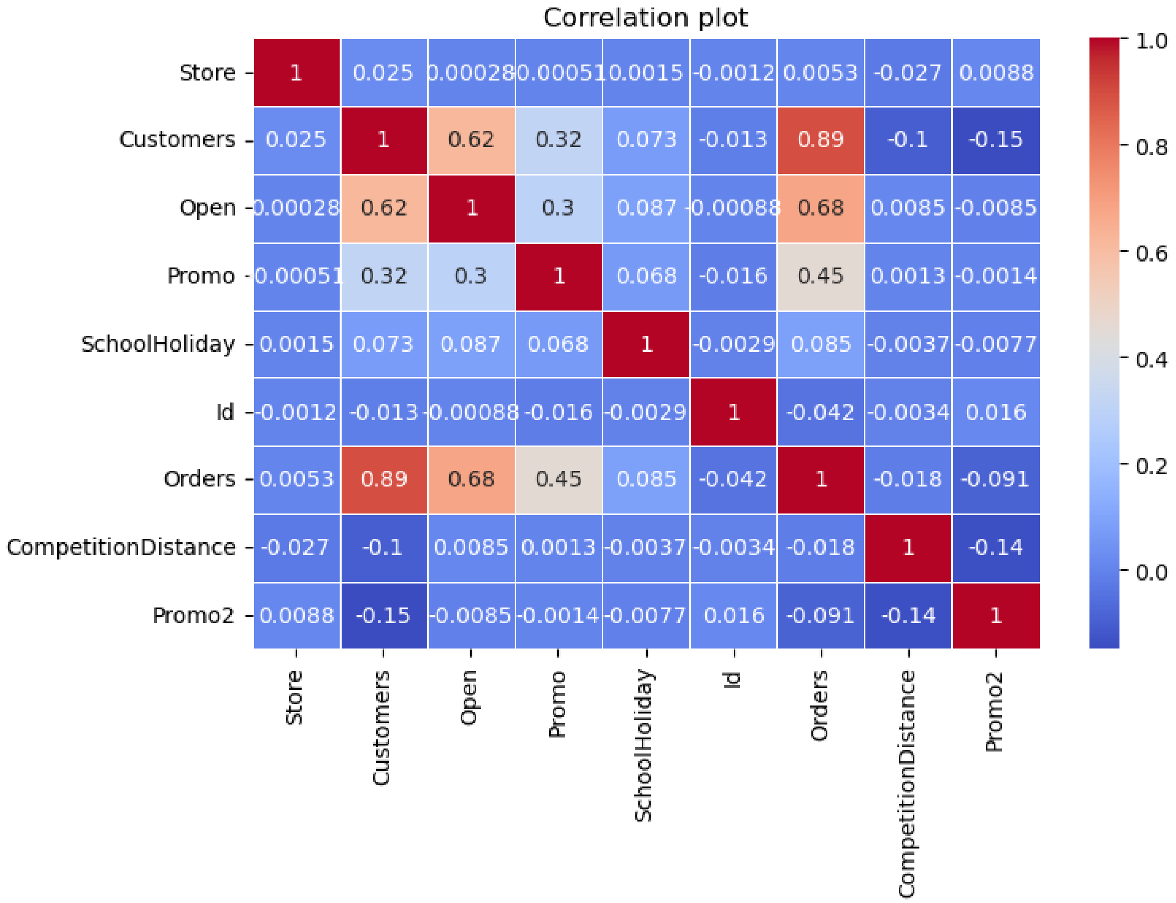


In [ ]:

In [36]:

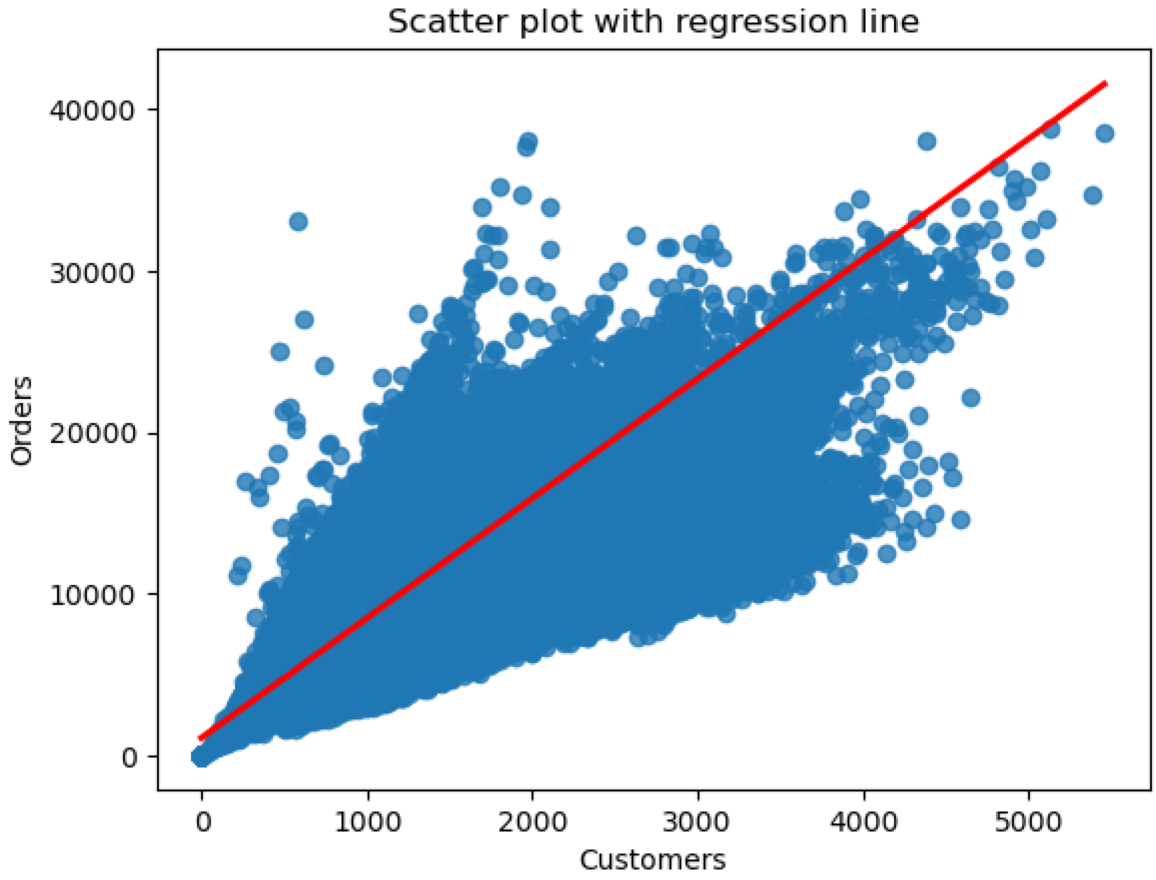
```
plt.figure(figsize=(8,5))
sns.heatmap(merged_final.corr(), annot=True, cmap='coolwarm', linewidths=0.5)
plt.title('Correlation plot')
plt.show()
```

```
C:\Users\balde\AppData\Local\Temp\ipykernel_15332\2851571332.py:2: FutureWarning:
The default value of numeric_only in DataFrame.corr is deprecated. In a future ver
sion, it will default to False. Select only valid columns or specify the value of
numeric_only to silence this warning.
  sns.heatmap(merged_final.corr(), annot=True, cmap='coolwarm', linewidths=0.5)
```



Scatter plot comparison between Orders and Customers

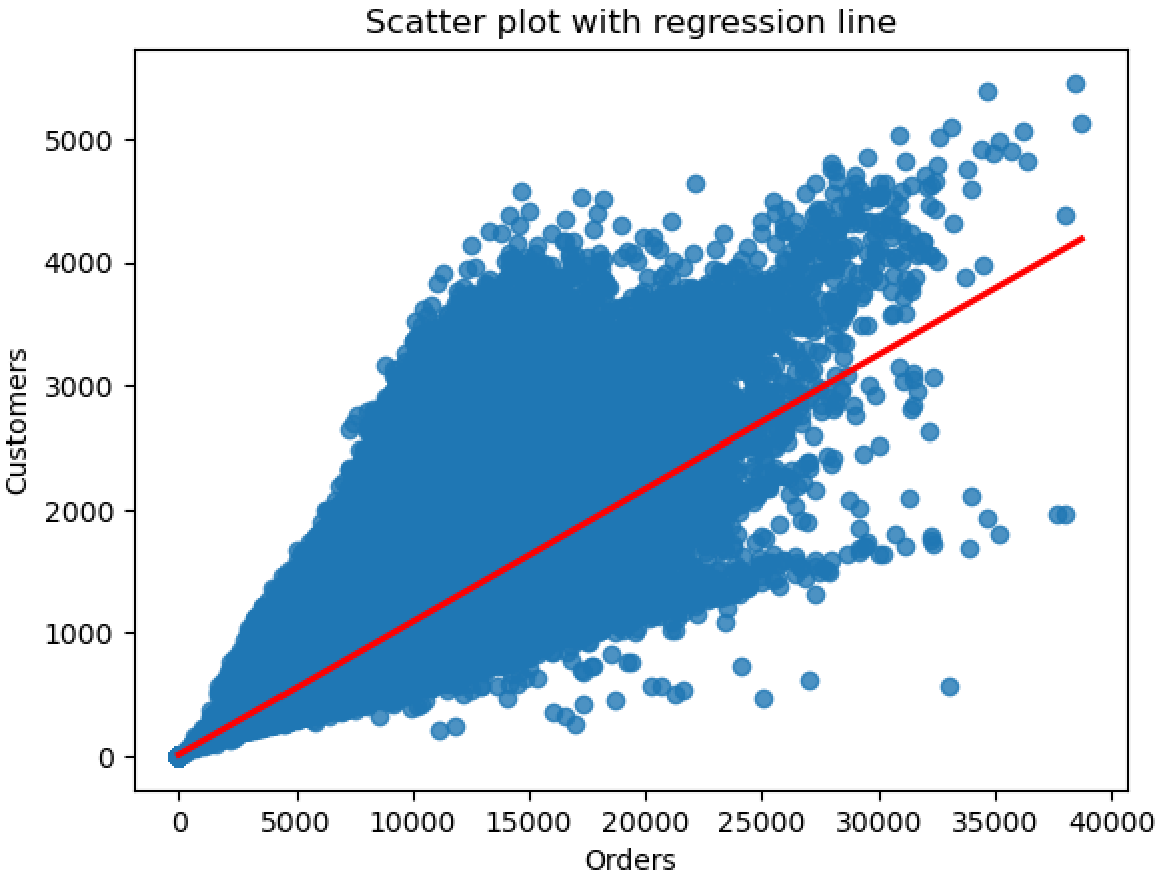
In [55]:
```
sns.regplot(data=merged_final, x = 'Customers', y = 'Orders', line_kws=dict(color='
plt.title('Scatter plot with regression line')
plt.show()
```



Scatter plot comparison between Customers and Orders

In [56]:

```
sns.regplot(data=merged_final, x = 'Orders', y = 'Customers',line_kws=dict(color="r
plt.title('Scatter plot with regression line')
plt.show()
```



In [ ]:

```
```

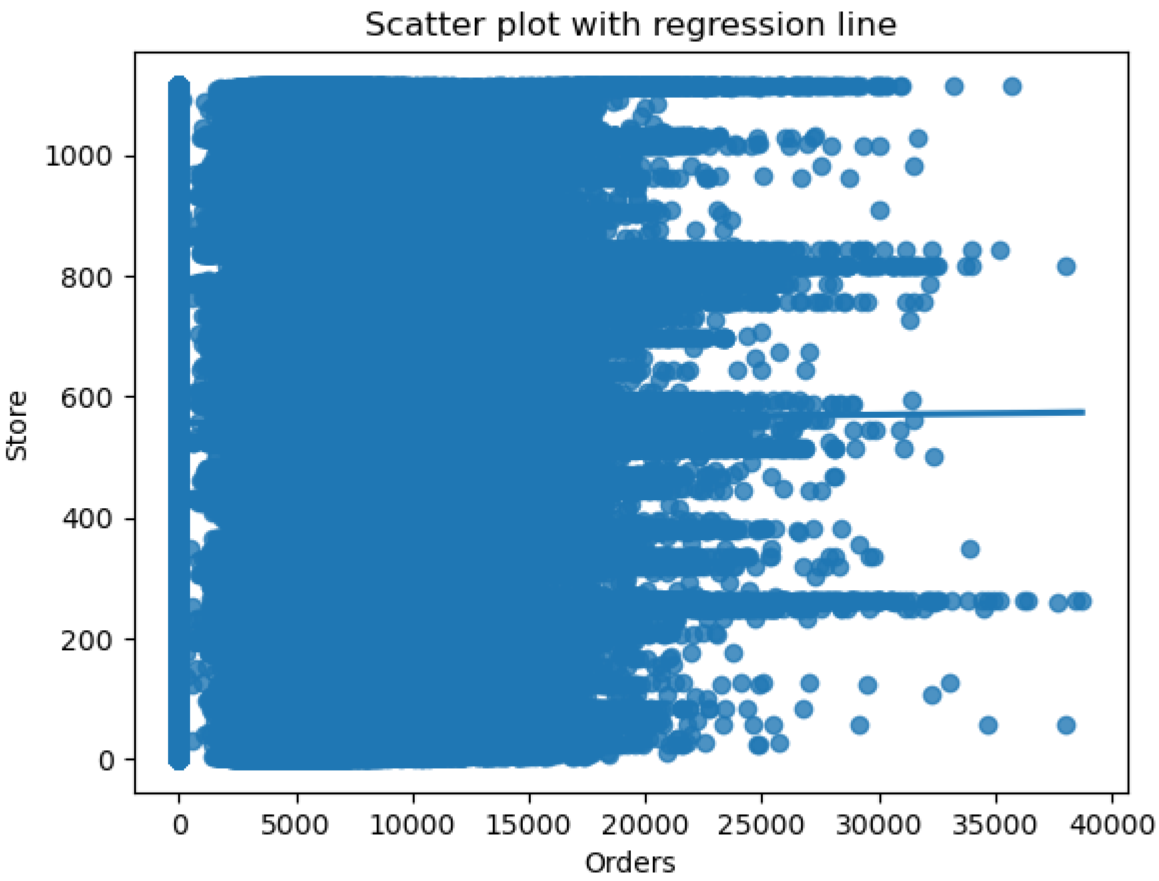
In [ ]:

```
```

Scatter plot comparison between Store and Orders

In [42]:

```
sns.regplot(data=merged_final, x = 'Orders', y = 'Store')
plt.title('Scatter plot with regression line')
plt.show()
```



In [ ]:

In [43]:

```
merged_final.select_dtypes(include='object').columns
```

Out[43]:

```
Index(['DayOfWeek', 'StateHoliday', 'StoreType', 'Assortment',
       'CompetitionOpenSinceMonth', 'CompetitionOpenSinceYear',
       'Promo2SinceWeek', 'Promo2SinceYear', 'PromoInterval'],
      dtype='object')
```

In [44]:

```
df_encoded = pd.get_dummies(merged_final, columns = ['StateHoliday','StoreType','As
                                                'Promo2SinceWeek','Promo2SinceYear'
df_encoded.columns
```

```
C:\Users\balde\AppData\Local\Temp\ipykernel_15332\475200570.py:1: FutureWarning: I
n a future version, the Index constructor will not infer numeric dtypes when passe
d object-dtype sequences (matching Series behavior)
  df_encoded = pd.get_dummies(merged_final, columns = ['StateHoliday','StoreTyp
e','Assortment','CompetitionOpenSinceMonth','CompetitionOpenSinceYear',
C:\Users\balde\AppData\Local\Temp\ipykernel_15332\475200570.py:1: FutureWarning: I
n a future version, the Index constructor will not infer numeric dtypes when passe
d object-dtype sequences (matching Series behavior)
  df_encoded = pd.get_dummies(merged_final, columns = ['StateHoliday','StoreTyp
e','Assortment','CompetitionOpenSinceMonth','CompetitionOpenSinceYear',
C:\Users\balde\AppData\Local\Temp\ipykernel_15332\475200570.py:1: FutureWarning: I
n a future version, the Index constructor will not infer numeric dtypes when passe
d object-dtype sequences (matching Series behavior)
  df_encoded = pd.get_dummies(merged_final, columns = ['StateHoliday','StoreTyp
e','Assortment','CompetitionOpenSinceMonth','CompetitionOpenSinceYear',
C:\Users\balde\AppData\Local\Temp\ipykernel_15332\475200570.py:1: FutureWarning: I
n a future version, the Index constructor will not infer numeric dtypes when passe
d object-dtype sequences (matching Series behavior)
  df_encoded = pd.get_dummies(merged_final, columns = ['StateHoliday','StoreTyp
e','Assortment','CompetitionOpenSinceMonth','CompetitionOpenSinceYear',
C:\Users\balde\AppData\Local\Temp\ipykernel_15332\475200570.py:1: FutureWarning: I
n a future version, the Index constructor will not infer numeric dtypes when passe
d object-dtype sequences (matching Series behavior)
  df_encoded = pd.get_dummies(merged_final, columns = ['StateHoliday','StoreTyp
e','Assortment','CompetitionOpenSinceMonth','CompetitionOpenSinceYear',
```

Out[44]:

```
Index(['Store', 'Customers', 'Open', 'Promo', 'SchoolHoliday', 'Id', 'Orders',
       'CompetitionDistance', 'Promo2', 'StateHoliday_0',
       ...
       'PromoInterval_Jan,Apr,Jul,Oct', 'PromoInterval_Mar,Jun,Sept,Dec',
       'PromoInterval_PromoUnavailable', 'DayOfWeek_1', 'DayOfWeek_2',
       'DayOfWeek_3', 'DayOfWeek_4', 'DayOfWeek_5', 'DayOfWeek_6',
       'DayOfWeek_7'],
      dtype='object', length=101)
```

In [45]:

```
# Model fitting and splitting of dataset

target_column = 'Orders'
X = df_encoded.drop(['Orders'],axis = 1)
y = df_encoded['Orders']



# Train Test Split
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.75, test_siz



# Scaling is done after train test split to prevent data leakage
scaler = StandardScaler()
X_train_sc = scaler.fit_transform(X_train)
X_test_sc = scaler.transform(X_test)

model = LinearRegression()
model.fit(X_train_sc, y_train)

# Make predictions on the scaled test data
ypred = model.predict(X_test_sc)
```

In [46]:

```
# Evaluating Mean squared error and R2 scores

mse = mean_squared_error(y_test, ypred)
r2 = r2_score(y_test, ypred)

# Print evaluation metrics
```

```
print(f"Mean Squared Error : {mse}")
print(f"R-squared : {r2}")
```

```
Mean Squared Error : 1356489.415316727
R-squared : 0.9082723261670267
```

In [ ]:

In [ ]:

Build an OLS model in Excel/Python on select features of the Retail dataset without using in-built OLS functions

In [47]:

```
import pandas as pd
import numpy as np
```

In [48]:

```
def ordinary_least_squares(X, Y):
    # Step 1: Calculate the means of X and Y
    mean_X = np.mean(X)
    mean_Y = np.mean(Y)

    # Step 2: Calculate the deviations from the means
    dev_X = X - mean_X
    dev_Y = Y - mean_Y

    # Step 3: Calculate the slope (m)
    m = np.sum(dev_X * dev_Y) / np.sum(dev_X ** 2)

    # Step 4: Calculate the intercept (b)
    b = mean_Y - m * mean_X

    # Step 5: Return the coefficients
    return m, b
```

In [49]:

```
def calculate_r_squared(X, Y, slope, intercept):
    # Step 1: Calculate the predicted values (Y_pred) using the linear regression m
    Y_pred = slope * X + intercept

    # Step 2: Calculate the total sum of squares (TSS)
    mean_Y = np.mean(Y)
    tss = np.sum((Y - mean_Y) ** 2)

    # Step 3: Calculate the residual sum of squares (RSS)
    rss = np.sum((Y - Y_pred) ** 2)

    # Step 4: Calculate R-squared (R²)
    r_squared = 1 - (rss / tss)

    return r_squared
```

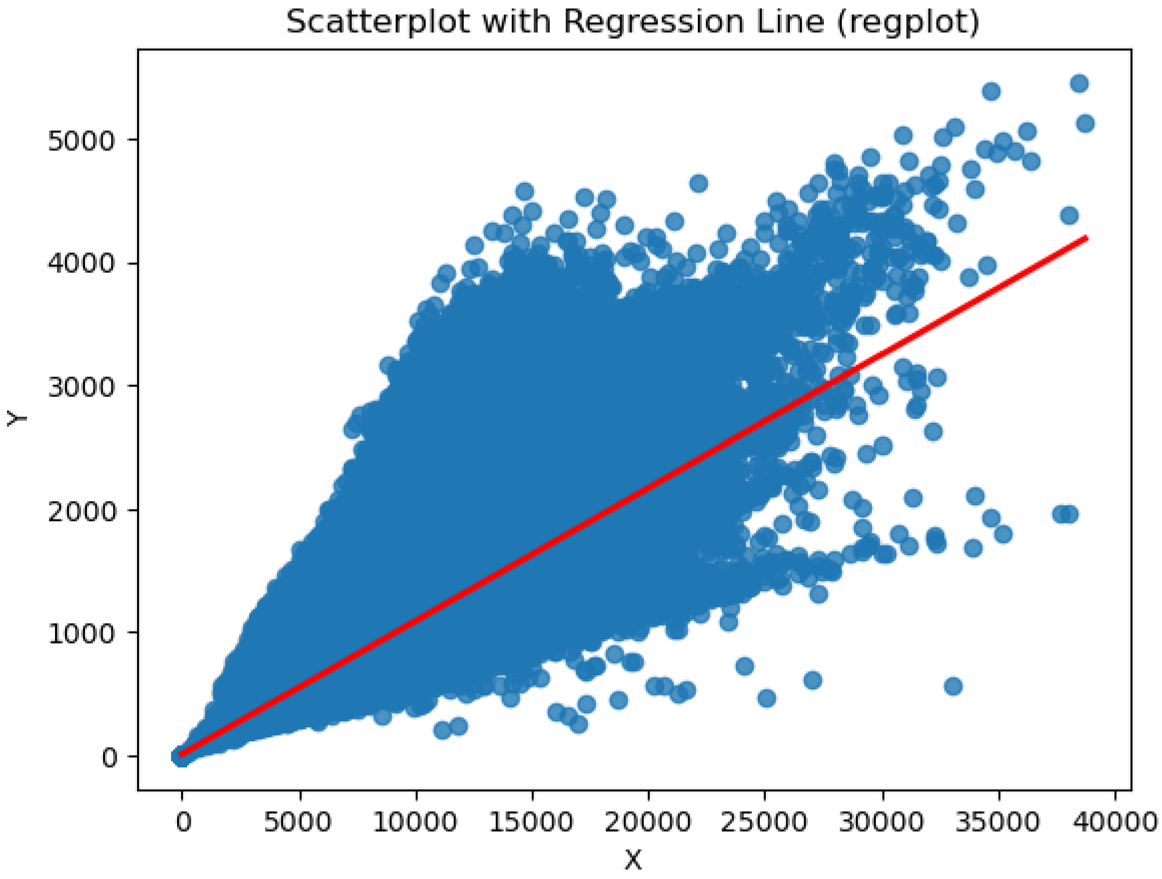
In [53]:

```
x = merged_final['Orders'].values
y = merged_final['Customers'].values

# Step 6: Use the function to get coefficients for your data

slope, intercept = ordinary_least_squares(x, y)
r_squared = calculate_r_squared(x, y, slope, intercept)

# Step 7: Output the results
print("Slope (m):", slope)
print("Intercept (b):", intercept)
print("R-squared (R²):", r_squared)
```

```
Slope (m): 0.107955007903139
Intercept (b): 9.8951915907096
R-squared (R²): 0.7998921458551951
```

In [57]:

```
import seaborn as sns
import matplotlib.pyplot as plt
sns.regplot(x=x, y=y, ci=None,line_kws=dict(color="r"))

# Show the plot
plt.xlabel('X')
plt.ylabel('Y')
plt.title('Scatterplot with Regression Line (regplot)')
plt.show()
```



In [ ]: