```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

#load the dataset
df = pd.read_csv('/Global_Superstore.csv')

# display first few rows
print("first 5 rows of the dataset:")
display(df.head())

#basic info abt dataset
print("\ndataset information:")
df.info()

print("\nstatistical summary:")
display(df.describe())

#check for duplicates
duplicates = df.duplicated().sum()
print(f"number of duplicate rows:{duplicates}")

#remove duplicates
df = df.drop_duplicates()

#handle missing values
print(f"missing values before cleaning:\n{df.isnull().sum()}")

#fill missing numerical values with column scan
# Select only numeric columns for calculating the mean
numeric_df = df.select_dtypes(include=['number'])
df[numeric_df.columns] = numeric_df.fillna(numeric_df.mean())

print(f"missing values after cleaning:\n{df.isnull().sum()}")

#convert date column to datetime format
df['date'] = pd.to_datetime(df['Order Date'])

#verify the changes
print("\ndata after cleaning:")
display(df.head())

#plot sales trend over time
plt.figure(figsize=(10,8))
df.groupby('date')['Sales'].sum().plot(kind='line', color='pink') # 'sales' should be
plt.title('sales trend over time')
plt.xlabel('date')
plt.ylabel('total sales')
plt.show()

#scatter plot profit vs discount
plt.figure(figsize=(10,8))
sns.scatterplot(x='Discount',y='Profit',data=df, color='blue') # 'discount', 'profit'
plt.title('profit vs discount')
plt.xlabel('discount')
plt.ylabel('profit')
plt.show()

#sales distribution by region
plt.figure(figsize=(10,8))
region_sales = df.groupby('Region')['Sales'].sum() # 'region', 'sales' should be 'Reg
region_sales.plot(kind='bar', color='yellow')
plt.title('sales by region')
plt.ylabel('total sales')
plt.show()

# heatmap for corelations
plt.figure(figsize=(10,8))
# Calculate correlation only for numeric columns
numeric_df = df.select_dtypes(include=['number'])
sns.heatmap(numeric_df.corr(), annot=True, cmap='coolwarm')
plt.title('correlation matrix')
plt.show()


from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
```

```
# select featured and target
x = df[['Profit', 'Discount']] # 'profit', 'discount' should be 'Profit', 'Discount'
y = df['Sales'] # 'sales' should be 'Sales'

#split the dataset into training and test sets
x_train, x_test, y_train, y_test = train_test_split(x,y, test_size=0.2, random_state=
model = LinearRegression()
model.fit(x_train, y_train)

#make predictions
y_pred = model.predict(x_test)

#evaluate the model
print (f"mean squared error: {mean_squared_error(y_test,y_pred):.2f}")
print(f"r-squared score: {r2_score(y_test,y_pred):.2f}")
```

first 5 rows of the dataset:

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segm
0	40098	CA-2014-AB10015140-41954	11/11/2014	11/13/2014	First Class	AB-100151402	Aaron Bergman	Consu
1	26341	IN-2014-JR162107-41675	2/5/2014	2/7/2014	Second Class	JR-162107	Justin Ritter	Corpo
2	25330	IN-2014-CR127307-41929	10/17/2014	10/18/2014	First Class	CR-127307	Craig Reiter	Consu
3	13524	ES-2014-KM1637548-41667	1/28/2014	1/30/2014	First Class	KM-1637548	Katherine Murray	H O
4	47221	SG-2014-RH9495111-41948	11/5/2014	11/6/2014	Same Day	RH-9495111	Rick Hansen	Consu

5 rows × 24 columns

```
dataset information:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 24 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Row ID          1000 non-null   int64
 1   Order ID        1000 non-null   object
 2   Order Date      1000 non-null   object
 3   Ship Date       1000 non-null   object
 4   Ship Mode       1000 non-null   object
 5   Customer ID     1000 non-null   object
 6   Customer Name   1000 non-null   object
 7   Segment         1000 non-null   object
 8   Postal Code     194 non-null    float64
 9   City            1000 non-null   object
 10  State           1000 non-null   object
 11  Country         1000 non-null   object
 12  Region          1000 non-null   object
 13  Market          1000 non-null   object
 14  Product ID      1000 non-null   object
 15  Category        1000 non-null   object
 16  Sub-Category    1000 non-null   object
 17  Product Name    1000 non-null   object
 18  Sales           1000 non-null   float64
 19  Quantity        1000 non-null   int64
 20  Discount        1000 non-null   float64
 21  Profit          1000 non-null   float64
 22  Shipping Cost   1000 non-null   float64
 23  Order Priority  1000 non-null   object
dtypes: float64(5), int64(2), object(17)
memory usage: 187.6+ KB
```

statistical summary:

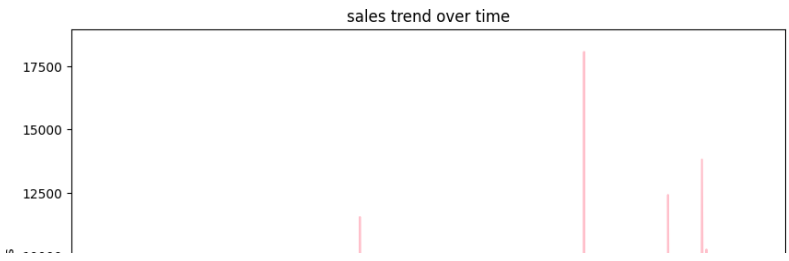
	Row ID	Postal Code	Sales	Quantity	Discount	Pro
count	1000.000000	194.000000	1000.000000	1000.00000	1000.000000	1000.000
mean	25079.328000	53966.170103	1710.971470	5.55800	0.092840	288.920
std	12897.726632	33734.306466	1259.239238	2.71846	0.148666	574.504
min	58.000000	2920.000000	1.910000	1.00000	0.000000	-3059.820
25%	15118.750000	19134.000000	826.907500	4.00000	0.000000	10.037
50%	25084.500000	60564.000000	1585.115000	5.00000	0.000000	190.685
75%	34524.000000	88187.500000	2477.812500	7.00000	0.150000	518.872
max	51284.000000	98198.000000	9892.740000	14.00000	0.800000	4946.370

```
number of duplicate rows:0
missing values before cleaning:
Row ID              0
Order ID            0
Order Date          0
Ship Date           0
```

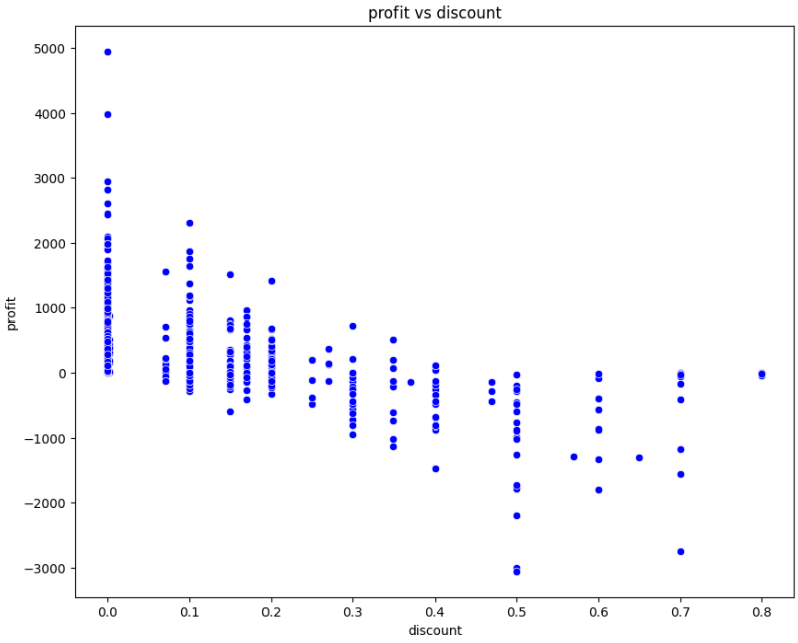
```
Ship Mode           0
Customer ID         0
Customer Name       0
Segment             0
Postal Code       806
City                0
State               0
Country             0
Region              0
Market              0
Product ID          0
Category            0
Sub-Category        0
Product Name        0
Sales               0
Quantity            0
Discount            0
Profit              0
Shipping Cost       0
Order Priority      0
dtype: int64
missing values after cleaning:
Row ID            0
Order ID          0
Order Date        0
Ship Date         0
Ship Mode         0
Customer ID       0
Customer Name     0
Segment           0
Postal Code       0
City              0
State             0
Country           0
Region            0
Market            0
Product ID        0
Category          0
Sub-Category      0
Product Name      0
Sales             0
Quantity          0
Discount          0
Profit            0
Shipping Cost     0
Order Priority    0
dtype: int64

data after cleaning:
```

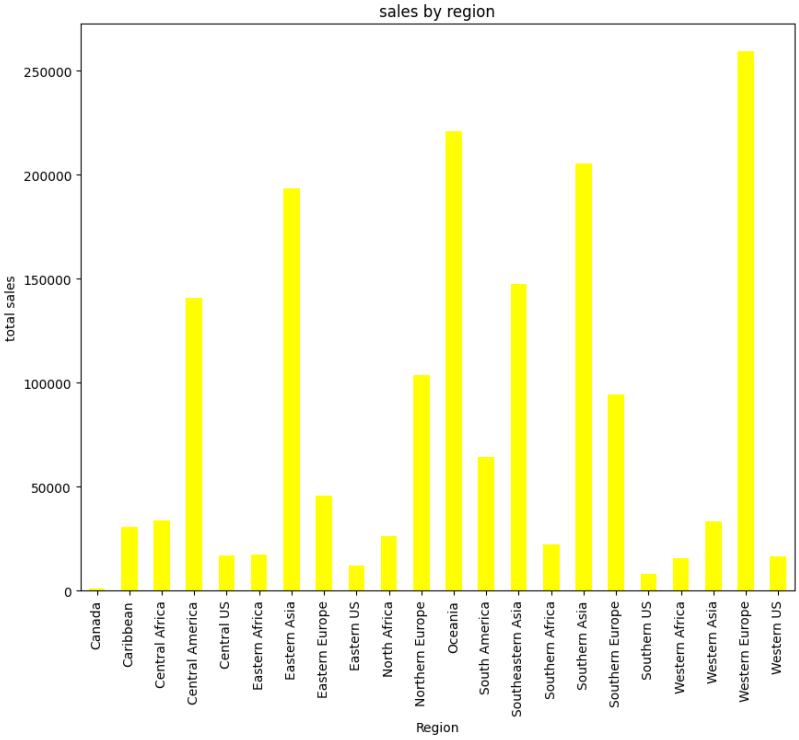
	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segm
0	40098	CA-2014-AB10015140-41954	11/11/2014	11/13/2014	First Class	AB-100151402	Aaron Bergman	Consu
1	26341	IN-2014-JR162107-41675	2/5/2014	2/7/2014	Second Class	JR-162107	Justin Ritter	Corpo
2	25330	IN-2014-CR127307-41929	10/17/2014	10/18/2014	First Class	CR-127307	Craig Reiter	Consu
3	13524	ES-2014-KM1637548-41667	1/28/2014	1/30/2014	First Class	KM-1637548	Katherine Murray	H O
4	47221	SG-2014-RH9495111-41948	11/5/2014	11/6/2014	Same Day	RH-9495111	Rick Hansen	Consu

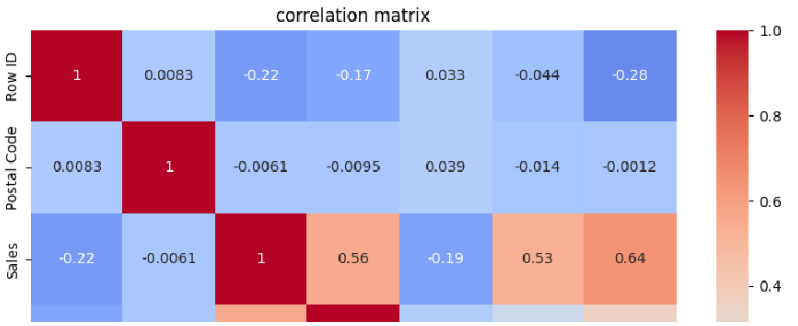
5 rows × 25 columns




total sale
10000
7500
5000
2500
0
2012-01
2012-07
2013-01
2013-07
2014-01
2014-07
2015-01
2015-07
2016-01
date


profit vs discount
profit
5000
4000
3000
2000
1000
0
−1000
−2000
−3000
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
discount


sales by region
total sales
250000
200000
150000
100000
50000
0
Canada
Caribbean
Central Africa
Central America
Central US
Eastern Africa
Eastern Asia
Eastern Europe
Eastern US
North Africa
Northern Europe
Oceania
South America
Southeastern Asia
Southern Africa
Southern Asia
Southern Europe
Southern US
Western Africa
Western Asia
Western Europe
Western US
Region


correlation matrix
Row ID
Postal Code
Sales
1
0.0083
-0.22
-0.17
0.033
-0.044
-0.28
0.0083
1
-0.0061
-0.0095
0.039
-0.014
-0.0012
-0.22
-0.0061
1
0.56
-0.19
0.53
0.64
1.0
0.8
0.6
0.4


Quantity
Discount
Profit
Shipping Cost
-0.17
-0.0095
0.56
1
0.034
0.15
0.36
0.033
0.039
-0.19
0.034
1
-0.51
-0.17
-0.044
-0.014
0.53
0.15
-0.51
1
0.29
-0.28
-0.0012
0.64
0.36
-0.17
0.29
1
Row ID
Postal Code
Sales
Quantity
Discount
Profit
Shipping Cost
0.2
0.0
−0.2
−0.4