```
!pip install names
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting names
  Downloading names-0.3.0.tar.gz (789 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 789.1/789.1 kB 13.3 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Building wheels for collected packages: names
  Building wheel for names (setup.py) ... done
  Created wheel for names: filename=names-0.3.0-py3-none-any.whl size=803698 sha256=b3d8f4b7a28e8f76a00223129d48307c13070038b121dc31dae5
  Stored in directory: /root/.cache/pip/wheels/fc/9a/6f/78f4282bbcaa2d8c678b73c54c0bb1b7a04009f0d7cec79fce
Successfully built names
Installing collected packages: names
Successfully installed names-0.3.0
```

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import random
import names
import os
```

+ Code + Text

```
from google.colab import files
uploaded = files.upload()
```

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving superstore.csv to superstore.csv

```
df = pd.read_csv("/content/drive/MyDrive/Datasets/superstore.csv" ,encoding = "latin")
```

```
df.shape
```

```
(51290, 24)
```

```
df.describe()
```

	Row ID	Postal Code	Sales	Quantity	Discount
count	51290.00000	9994.000000	51290.000000	51290.000000	51290.000000
mean	25645.50000	55190.379428	246.490581	3.476545	0.142908
std	14806.29199	32063.693350	487.565361	2.278766	0.212280
min	1.00000	1040.000000	0.444000	1.000000	0.000000
25%	12823.25000	23223.000000	30.758625	2.000000	0.000000
50%	25645.50000	56430.500000	85.053000	3.000000	0.000000

```
df.columns
```

```
Index(['Row ID', 'Order ID', 'Order Date', 'Ship Date', 'Ship Mode',
       'Customer ID', 'Customer Name', 'Segment', 'City', 'State', 'Country',
       'Postal Code', 'Market', 'Region', 'Product ID', 'Category',
       'Sub-Category', 'Product Name', 'Sales', 'Quantity', 'Discount',
       'Profit', 'Shipping Cost', 'Order Priority'],
      dtype='object')
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 51290 entries, 0 to 51289
Data columns (total 24 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Row ID          51290 non-null  int64
 1   Order ID        51290 non-null  object
 2   Order Date      51290 non-null  object
```

```
 3   Ship Date       51290 non-null  object
 4   Ship Mode       51290 non-null  object
 5   Customer ID     51290 non-null  object
 6   Customer Name   51290 non-null  object
 7   Segment         51290 non-null  object
 8   City            51290 non-null  object
 9   State           51290 non-null  object
 10  Country         51290 non-null  object
 11  Postal Code     9994 non-null   float64
 12  Market          51290 non-null  object
 13  Region          51290 non-null  object
 14  Product ID      51290 non-null  object
 15  Category        51290 non-null  object
 16  Sub-Category    51290 non-null  object
 17  Product Name    51290 non-null  object
 18  Sales           51290 non-null  float64
 19  Quantity        51290 non-null  int64
 20  Discount        51290 non-null  float64
 21  Profit          51290 non-null  float64
 22  Shipping Cost   51290 non-null  float64
 23  Order Priority  51290 non-null  object
dtypes: float64(5), int64(2), object(17)
memory usage: 9.4+ MB
```

```
df.head(2 )
```

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment
0	42433	AG-2011-2040	1/1/2011	6/1/2011	Standard Class	TB-11280	Toby Braunhardt	Consumer
1	22253	IN-2011-47883	1/1/2011	8/1/2011	Standard Class	JH-15985	Joseph Holt	Consumer

2 rows × 24 columns

```
df.tail(1)
```

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segme
51289	36388	CA-2014-156720	31-12-2014	4/1/2015	Standard Class	JM-15580	Jill Matthias	Consum

1 rows × 24 columns

```
plt.figure(figsize=(15, 4))
df.boxplot()
```

```
<Axes: >
```


```
ids=["Postal Code"]
for i in df.columns:
  if "ID" in i:
    ids.append(i)
ids
```

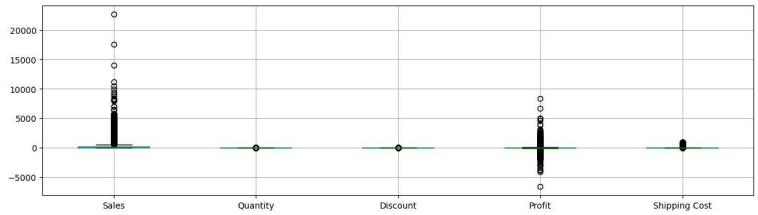
```
['Postal Code', 'Row ID', 'Order ID', 'Customer ID', 'Product ID']
```

```
df.drop(columns = ids, inplace = True)
df.columns
```

```
Index(['Order Date', 'Ship Date', 'Ship Mode', 'Customer Name', 'Segment',
       'City', 'State', 'Country', 'Market', 'Region', 'Category',
       'Sub-Category', 'Product Name', 'Sales', 'Quantity', 'Discount',
       'Profit', 'Shipping Cost', 'Order Priority'],
      dtype='object')
```

```
plt.figure(figsize=(15, 4))
df.boxplot()
```

```
<Axes: >
```

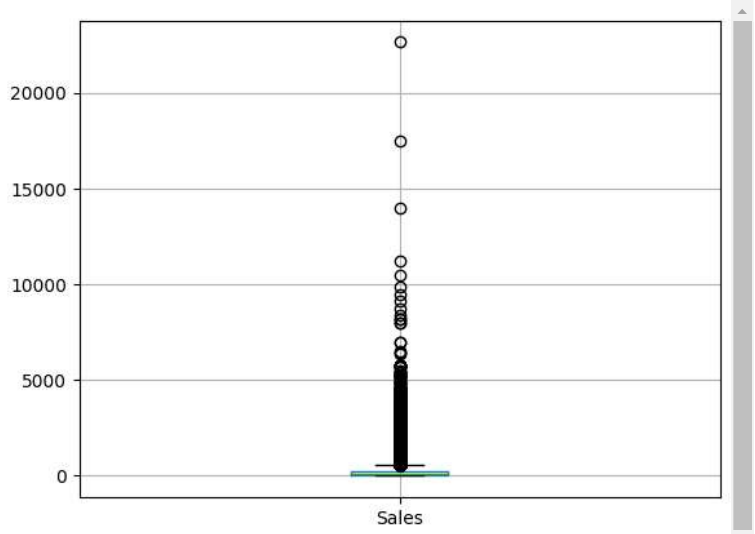


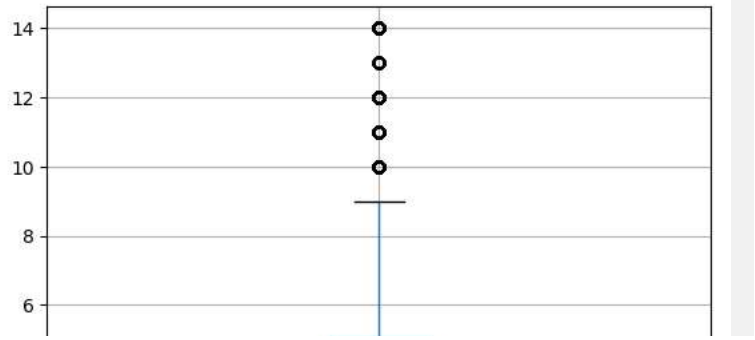

```
numeric_cols = []
for i in df.columns:
  if df[i].dtypes != "object":
    numeric_cols.append(i)

numeric_cols
```

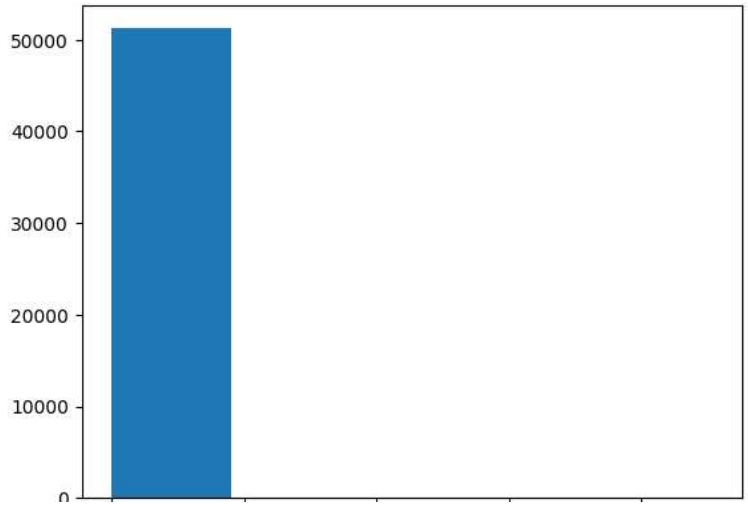
```
['Sales', 'Quantity', 'Discount', 'Profit', 'Shipping Cost']
```

```
for i in numeric_cols:
  df.boxplot(i)
  plt.show()
```

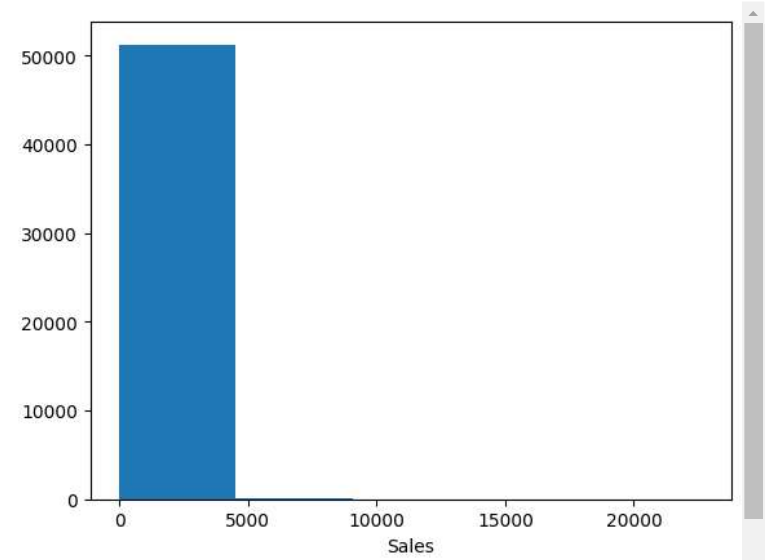


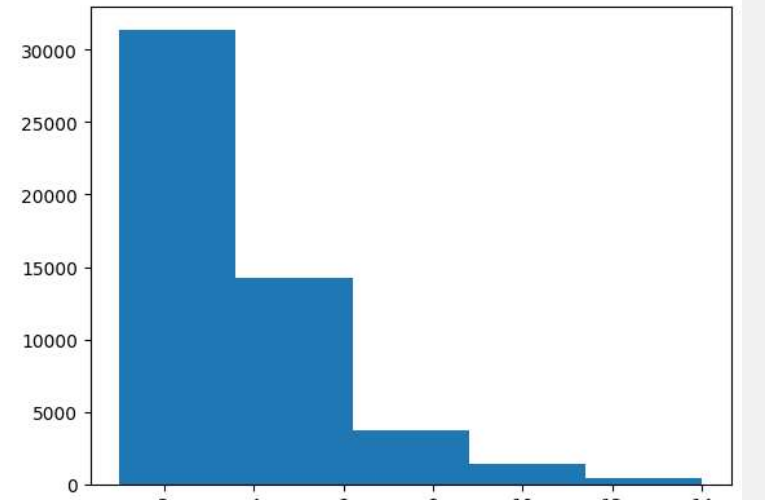



```
plt.hist(df['Sales'], bins = 5)
plt.show()
```




```
for i in numeric_cols:
  plt.hist(df[i], bins =5)
  plt.xlabel(i)
  plt.show()
```






```
df.describe().T
```

	count	mean	std	min	25%	50%	
Sales	51290.0	246.490581	487.565361	0.444	30.758625	85.053	251.(
Quantity	51290.0	3.476545	2.278766	1.000	2.000000	3.000	5.(
Discount	51290.0	0.142908	0.212280	0.000	0.000000	0.000	0.2
Profit	51290.0	28.610982	174.340972	-6599.978	0.000000	9.240	36.8

```
df.describe(include = "all").T
```

	count	unique	top	freq	mean	std	
Order Date	51290	1430	18-06-2014	135	NaN	NaN	N
Ship Date	51290	1464	22-11-2014	130	NaN	NaN	N
Ship Mode	51290	4	Standard Class	30775	NaN	NaN	N
Customer Name	51290	795	Muhammed Yedwab	108	NaN	NaN	N
Segment	51290	3	Consumer	26518	NaN	NaN	N
City	51290	3636	New York City	915	NaN	NaN	N

```
df.nunique()
```

```
Order Date         1430
Ship Date          1464
Ship Mode             4
Customer Name       795
Segment               3
City               3636
State              1094
Country             147
Market                7
Region               13
Category              3
Sub-Category         17
Product Name       3788
Sales             22995
Quantity             14
Discount             27
Profit            24575
Shipping Cost     10037
Order Priority        4
dtype: int64
```

```
cat_cols = [ 'Ship Mode', 'Segment' ,'City' ,'State' ,'Country', 'Market' ,'Region', 'Category','Sub-Category' ]
cat_cols
```

```
['Ship Mode',
 'Segment',
 'City',
 'State',
 'Country',
 'Market',
 'Region',
 'Category',
 'Sub-Category']
```

```
for i in cat_cols:
  print(f"Unique Values in  {i} column are as below\n", df[i].unique())
  print(("\n"))
```

```
Unique Values in  Ship Mode column are as below
 ['Standard Class' 'Second Class' 'First Class' 'Same Day']


Unique Values in  Segment column are as below
 ['Consumer' 'Home Office' 'Corporate']


Unique Values in  City column are as below
 ['Constantine' 'Wagga Wagga' 'Budapest' ... 'Missoula' 'Lannion'
 'Deer Park']


Unique Values in  State column are as below
 ['Constantine' 'New South Wales' 'Budapest' ... 'Medea' 'Jizzakh'
 'Inhambane']


Unique Values in  Country column are as below
 ['Algeria' 'Australia' 'Hungary' 'Sweden' 'Bangladesh' 'United States'
 'Angola' 'China' 'Panama' 'Iran' 'France' 'Italy' 'Germany' 'Canada'
 'United Kingdom' 'Ukraine' 'Japan' 'Indonesia' 'Nigeria' 'South Korea'
 'Peru' 'Philippines' 'Colombia' 'Ireland' 'Nicaragua' 'Mexico' 'Brazil'
 'Turkey' 'Spain' 'Poland' 'India' 'Somalia' 'El Salvador' 'Sudan'
```

```
 'Slovakia' 'Egypt' 'Saudi Arabia' 'Democratic Republic of the Congo'
 'Norway' 'New Zealand' 'Kenya' 'Cuba' 'Venezuela' 'Singapore' 'Honduras'
 'Tanzania' 'Dominican Republic' 'Morocco' 'Albania' 'Belgium'
 'Afghanistan' 'Bolivia' 'Vietnam' 'Guatemala' 'Guinea-Bissau' 'Thailand'
 'Iraq' 'Myanmar (Burma)' 'Ecuador' 'Netherlands' 'Ghana' "Cote d'Ivoire"
 'Austria' 'Argentina' 'Madagascar' 'Russia' 'South Africa'
 'Bosnia and Herzegovina' 'Malaysia' 'Romania' 'Israel' 'Burundi'
 'Cameroon' 'Paraguay' 'Senegal' 'Georgia' 'Kazakhstan'
 'United Arab Emirates' 'Pakistan' 'Liberia' 'Czech Republic' 'Jamaica'
 'Benin' 'Taiwan' 'Rwanda' 'Switzerland' 'Denmark' 'Haiti' 'Qatar' 'Chile'
 'Bulgaria' 'Mozambique' 'Lebanon' 'Barbados' 'Uzbekistan' 'Moldova'
 'Cambodia' 'Guinea' 'Azerbaijan' 'Zambia' 'Uruguay' 'Portugal' 'Uganda'
 'Martinique' 'Togo' 'Zimbabwe' 'Finland' 'Belarus' 'Libya' 'Lithuania'
 'Republic of the Congo' 'Tunisia' 'Papua New Guinea' 'Turkmenistan'
 'Yemen' 'Trinidad and Tobago' 'Kyrgyzstan' 'Croatia' 'Nepal' 'Mali'
 'Namibia' 'Syria' 'Sierra Leone' 'Gabon' 'Mauritania' 'Guadeloupe'
 'Niger' 'Sri Lanka' 'Djibouti' 'Jordan' 'Equatorial Guinea' 'Hong Kong'
 'Mongolia' 'Eritrea' 'Slovenia' 'Ethiopia' 'Tajikistan' 'Montenegro'
 'Central African Republic' 'Lesotho' 'Chad' 'Armenia' 'Swaziland'
 'Estonia' 'South Sudan' 'Bahrain' 'Macedonia']


Unique Values in  Market column are as below
 ['Africa' 'APAC' 'EMEA' 'EU' 'US' 'LATAM' 'Canada']


Unique Values in  Region column are as below
 ['Africa' 'Oceania' 'EMEA' 'North' 'Central Asia' 'West' 'North Asia'
 'Central' 'South' 'Canada' 'Southeast Asia' 'East' 'Caribbean']


Unique Values in  Category column are as below
 ['Office Supplies' 'Furniture' 'Technology']
```

```
df.duplicated().sum()
```

```
0
```

```
df[df.duplicated()]
```

	Order Date	Ship Date	Ship Mode	Customer Name	Segment	City	State	Country	Market	Regio

```
df.drop_duplicates(inplace = True)
```

```
df.corr
```

```
<bound method DataFrame.corr of        Order Date Ship Date       Ship Mode     Customer Name      Segment  \
0        1/1/2011  6/1/2011  Standard Class   Toby Braunhardt     Consumer
1        1/1/2011  8/1/2011  Standard Class       Joseph Holt     Consumer
2        1/1/2011  5/1/2011    Second Class     Annie Thurman     Consumer
3        1/1/2011  5/1/2011    Second Class      Eugene Moren  Home Office
4        1/1/2011  8/1/2011  Standard Class       Joseph Holt     Consumer
...           ...       ...            ...               ...          ...
51285  31-12-2014  4/1/2015  Standard Class        Erica Bern    Corporate
51286  31-12-2014  5/1/2015  Standard Class         Liz Preis     Consumer
51287  31-12-2014  2/1/2015    Second Class  Charlotte Melton     Consumer
51288  31-12-2014  6/1/2015  Standard Class     Tamara Dahlen     Consumer
51289  31-12-2014  4/1/2015  Standard Class     Jill Matthias     Consumer

              City             State        Country  Market   Region  \
0      Constantine       Constantine        Algeria  Africa   Africa
1      Wagga Wagga   New South Wales      Australia    APAC  Oceania
2         Budapest          Budapest        Hungary    EMEA     EMEA
3        Stockholm         Stockholm         Sweden      EU    North
4      Wagga Wagga   New South Wales      Australia    APAC  Oceania
...            ...               ...            ...     ...      ...
51285    Fairfield        California  United States      US     West
51286       Agadir  Souss-Massa-Draâ        Morocco  Africa   Africa
51287      Managua           Managua      Nicaragua   LATAM  Central
51288       Juárez         Chihuahua         Mexico   LATAM    North
51289     Loveland          Colorado  United States      US     West

              Category Sub-Category  \
0      Office Supplies      Storage
1      Office Supplies     Supplies
```

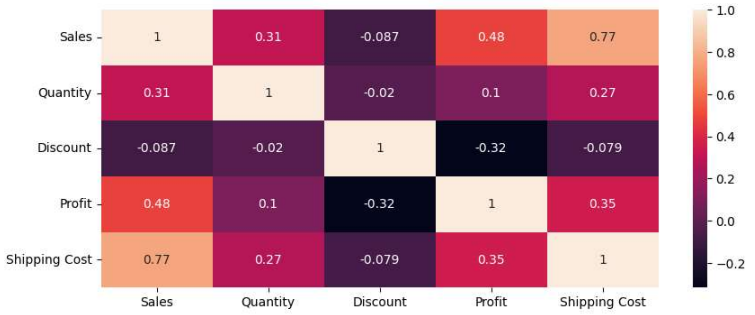
```
2      Office Supplies      Storage
3      Office Supplies        Paper
4            Furniture  Furnishings
...                ...          ...
51285  Office Supplies      Binders
51286  Office Supplies      Binders
51287  Office Supplies       Labels
51288  Office Supplies       Labels
51289  Office Supplies    Fasteners

                                      Product Name    Sales  Quantity  \
0                              Tenex Lockers, Blue  408.300         2
1                          Acme Trimmer, High Speed  120.366         3
2                            Tenex Box, Single Width   66.120         4
3                         Enermax Note Cards, Premium   44.865         3
4                          Eldon Light Bulb, Duo Pack  113.670         5
...                                              ...      ...       ...
51285  Cardinal Slant-D Ring Binder, Heavy Gauge Vinyl   13.904         2
51286          Wilson Jones Hole Reinforcements, Clear    3.990         1
51287           Hon Color Coded Labels, 5000 Label Set   26.400         3
51288           Hon Legal Exhibit Labels, Alphabetical    7.120         1
51289                              Bagged Rubber Bands    3.024         3

       Discount    Profit  Shipping Cost Order Priority
0           0.0  106.1400          35.46         Medium
1           0.1   36.0360           9.72         Medium
2           0.0   29.6400           8.17           High
3           0.5  -26.0550           4.82           High
```

```
plt.figure(figsize = (10,4))
sns.heatmap(df.corr(), annot = True);
```

```
<ipython-input-67-7a2682a176fe>:2: FutureWarning: The default value of num
  sns.heatmap(df.corr(), annot = True);
```

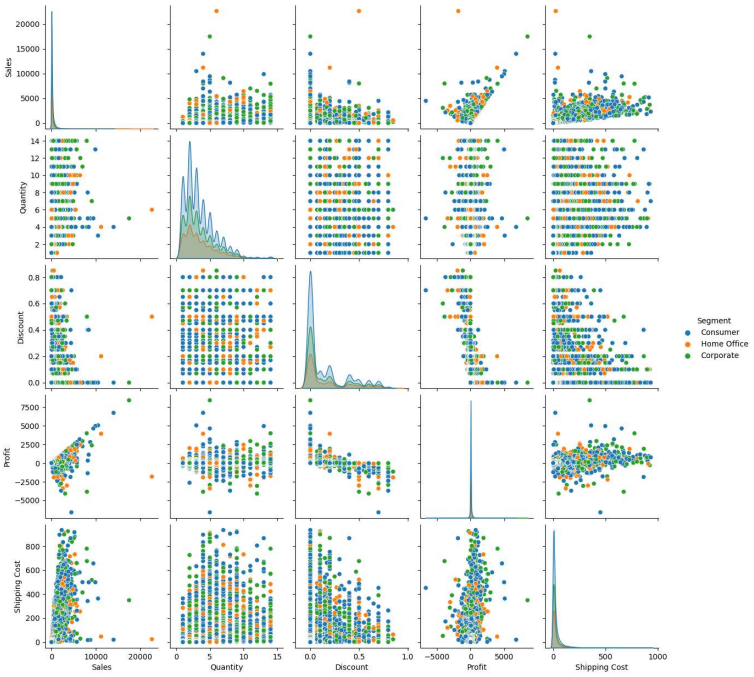



```
df.columns
```

```
Index(['Order Date', 'Ship Date', 'Ship Mode', 'Customer Name', 'Segment',
       'City', 'State', 'Country', 'Market', 'Region', 'Category',
       'Sub-Category', 'Product Name', 'Sales', 'Quantity', 'Discount',
       'Profit', 'Shipping Cost', 'Order Priority'],
      dtype='object')
```

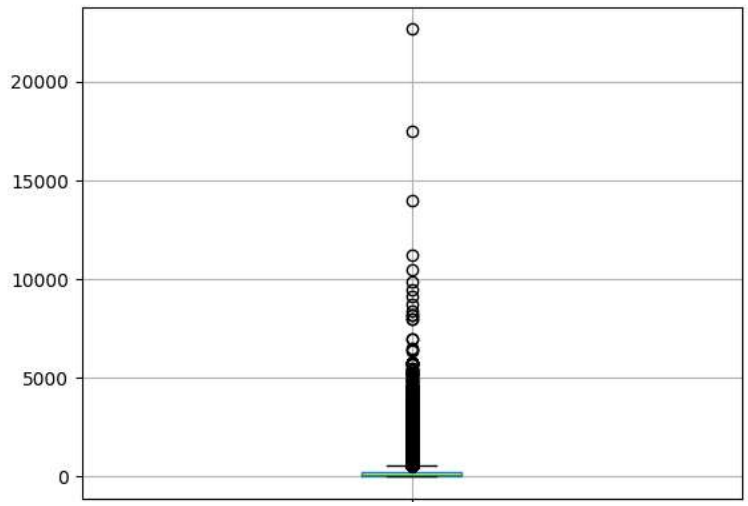
```
plt.figure(figsize= (10,10))
sns.pairplot(df, hue = "Segment");
```

```
<Figure size 1000x1000 with 0 Axes>
```




```
df.boxplot('Sales');
```

```
<Axes: >
```




```
df['Sales'].describe()
```

```
count    51290.000000
mean       246.490581
std        487.565361
min          0.444000
25%         30.758625
50%         85.053000
75%        251.053200
max      22638.480000
Name: Sales, dtype: float64
```

```
q1 = np.quantile(df['Sales'], 0.25)
q2 = np.quantile(df['Sales'], 0.5)
q3 = np.quantile(df['Sales'], 0.75)
```

```
print(q1, q2, q3)
```

```
30.758625000000002 85.053 251.0532
```

```
iqr = q3 - q1
uw = q3 + (1.5 * iqr)
lw = q1 - (1.5 * iqr)
```

```
print(lw, uw)
```

```
-299.6832375 581.4950625
```

```
len(df[df['Sales'] > uw])
```

```
5655
```

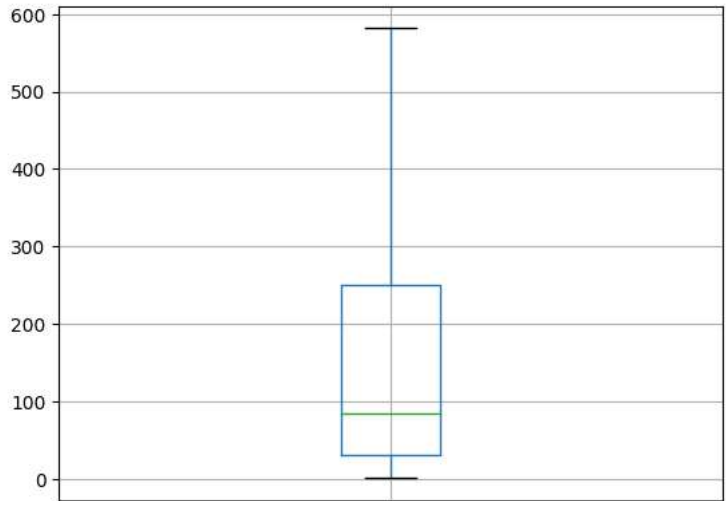
```
outliers = df[df['Sales'] > uw].index
```

```
df.iloc[outliers].replace(df['Sales'], uw)
```

Order Date	Ship Date	Ship Mode	Customer Name	Segment	City
		Standard	Todd		

```
df['Sales'] = np.where(df['Sales'] > uw, uw, df['Sales'])
```

```
df.boxplot('Sales');
```




```
df['Order Date'].dtypes
```

```
dtype('O')
```

```
df['Order Date'] = pd.DatetimeIndex(df['Order Date'])
```

```
<ipython-input-94-259cacc9b45b>:1: UserWarning: Parsing dates in DD/MM/YYYY format when dayfirst=False (the default) was specified. This
  df['Order Date'] = pd.DatetimeIndex(df['Order Date'])
```

```
df['Month'] = pd.DatetimeIndex(df['Order Date']).month_name()
```

```
df['Year'] = pd.DatetimeIndex(df['Order Date']).year
```

```
df.head()
```

	Order Date	Ship Date	Ship Mode	Customer Name	Segment	City	State	C
0	2011-01-01	6/1/2011	Standard Class	Toby Braunhardt	Consumer	Constantine	Constantine	

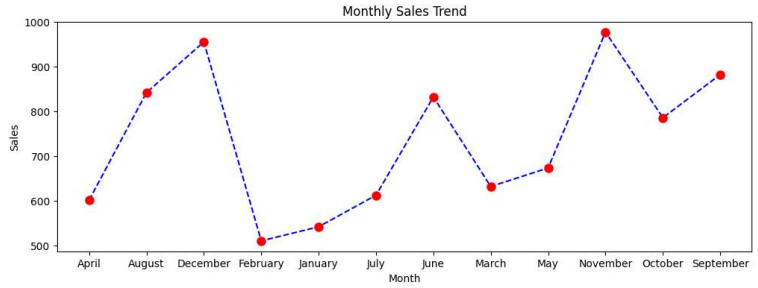
```
df['Year'].unique()
```

```
array([2011, 2012, 2013, 2014])
```

```
monthlytrend = df.groupby("Month", as_index = False) ['Sales'].sum()
monthlytrend["Sales (in $k' s)"] = round(monthlytrend['Sales']/1000, 2)
monthlytrend
```

	Month	Sales	Sales (in $k' s)
0	April	600766.079078	600.77
1	August	841630.000670	841.63
2	December	954811.328498	954.81
3	February	509859.535230	509.86
4	January	541232.058245	541.23
5	July	611912.063505	611.91
6	June	831725.856035	831.73
7	March	631433.544325	631.43
8	May	672920.768565	672.92
9	November	976904.728160	976.90
10	October	785286.253508	785.29

```
plt.figure(figsize = (12, 4))
plt.title("Monthly Sales Trend")
plt.plot(monthlytrend['Month'], monthlytrend["Sales (in $k' s)"], "b--o" , mec = "r", mfc = 'r' , ms = 8)
plt.xlabel("Month")
plt.ylabel("Sales")
plt.show()
```




```
yearlytrend = df.groupby("Year", as_index = False) ['Sales'].sum()
yearlytrend["Sales (in $k' s)"] = round(yearlytrend['Sales']/1000, 2)
yearlytrend
```

	Year	Sales	Sales (in $k' s)
0	2011	1.556334e+06	1556.33

```
plt.figure(figsize = (12, 4))
plt.title("Yearly Sales Trend")
plt.bar(yearlytrend['Year'], yearlytrend["Sales (in $k' s)"], color = [ 'r', 'g', 'b' ,'m'])
plt.xticks(list(df['Year'].unique()))
plt.xlabel("Years")
plt.ylabel("Sales")
plt.show()
```