

```
!pip install openpyxl

Requirement already satisfied: openpyxl in /usr/local/lib/python3.11/dist-packages (3.1.5)
Requirement already satisfied: et-xmlfile in /usr/local/lib/python3.11/dist-packages (from openpyxl) (2.0.0)

import pandas as pd
from matplotlib import pyplot as plt


!unzip /content/online+retail.zip

Archive: /content/online+retail.zip
  extracting: Online Retail.xlsx
```

Start coding or [generate](#) with AI.

```
df1=pd.read_excel('Online Retail.xlsx',dtype={'InvoiceNo':'string','StockCode':'string','Description':'string','InvoiceNo':'string','Country':'string'})

df1.head(10)
```




	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
5	536365	22752	SET 7 BABUSHKA NESTING BOXES	2	2010-12-01 08:26:00	7.65	17850.0	United Kingdom
6	536365	21730	GLASS STAR FROSTED T-LIGHT HOLDER	6	2010-12-01 08:26:00	4.25	17850.0	United Kingdom
7	536366	22633	HAND WARMER UNION JACK	6	2010-12-01 08:28:00	1.85	17850.0	United Kingdom
8	536366	22632	HAND WARMER RED POLKA DOT	6	2010-12-01 08:28:00	1.85	17850.0	United Kingdom
9	536367	84879	ASSORTED COLOUR BIRD ORNAMENT	32	2010-12-01 08:34:00	1.69	13047.0	United Kinadom

```
#for detailed info
df1.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 541909 entries, 0 to 541908
Data columns (total 8 columns):
#   Column          Non-Null Count  Dtype
---  -
0   InvoiceNo        541909 non-null string
1   StockCode       541909 non-null string
2   Description     540455 non-null string
3   Quantity        541909 non-null int64
4   InvoiceDate     541909 non-null datetime64[ns]
5   UnitPrice       541909 non-null float64
6   CustomerID     406829 non-null float64
7   Country         541909 non-null string
dtypes: datetime64[ns](1), float64(2), int64(1), string(4)
memory usage: 33.1 MB

#### data cleaning and missing value###
df1.isnull().sum()
# gives count of null values with column
#description have 1454 records as empty
```



	0
InvoiceNo	0
StockCode	0
Description	1454
Quantity	0
InvoiceDate	0
UnitPrice	0
CustomerID	135080
Country	0

```
df1[df1.Description.isnull()]
# gives only null values
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
622	536414	22139	<NA>	56	2010-12-01 11:52:00	0.0	NaN	United Kingdom
1970	536545	21134	<NA>	1	2010-12-01 14:32:00	0.0	NaN	United Kingdom
1971	536546	22145	<NA>	1	2010-12-01 14:33:00	0.0	NaN	United Kingdom
1972	536547	37509	<NA>	1	2010-12-01 14:33:00	0.0	NaN	United Kingdom
1987	536549	85226A	<NA>	1	2010-12-01 14:34:00	0.0	NaN	United Kingdom
...	...	...	...	...	...	...	...	...
535322	581199	84581	<NA>	-2	2011-12-07 18:26:00	0.0	NaN	United Kingdom
535326	581203	23406	<NA>	15	2011-12-07 18:31:00	0.0	NaN	United Kingdom
535332	581209	21620	<NA>	6	2011-12-07 18:35:00	0.0	NaN	United Kingdom
536981	581234	72817	<NA>	27	2011-12-08 10:33:00	0.0	NaN	United Kingdom
538554	581408	85175	<NA>	20	2011-12-08 14:06:00	0.0	NaN	United Kingdom

1454 rows x 8 columns

```
#check the description of stockcode=22139 with other records and may be the most occuring result is the description
#IF DESCRIPTION IS NULL THEN LOOK AT OTHER RECORDS WITH SAME STOCK CODE WHICH EVER THE DESCRPTION OCCURS FREQUENTLY OCCURED THEN CONSIDER
df1[df1.StockCode=='22139']
```



	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
106	536381	22139	RETROSPOT TEA SET CERAMIC 11 PC	23	2010-12-01 09:41:00	4.25	15311.0	United Kingdom
622	536414	22139	<NA>	56	2010-12-01 11:52:00	0.00	NaN	United Kingdom
6392	536942	22139	amazon	15	2010-12-03 12:08:00	0.00	NaN	United Kingdom
6885	536982	22139	RETROSPOT TEA SET CERAMIC 11 PC	10	2010-12-03 14:27:00	11.02	NaN	United Kingdom
7203	537011	22139	<NA>	-5	2010-12-03 15:38:00	0.00	NaN	United Kingdom
...	...	...	...	...	...	...	...	...
538411	581405	22139	RETROSPOT TEA SET CERAMIC 11 PC	1	2011-12-08 13:50:00	4.95	13521.0	United Kingdom
539531	581439	22139	RETROSPOT TEA SET CERAMIC 11 PC	1	2011-12-08 16:30:00	10.79	NaN	United Kingdom

```
#TO FIND MOST FREQUENTLY OCCURED RECORD USE MODE() AS IT HELPS TO RETRIEVE THE EXACT RECORD
df1[df1.StockCode=='22139'].Description.mode()
```

Description

0 RETROSPOT TEA SET CERAMIC 11 PC

#valuecounts gives count of total occurrence of that description in the dataset  
most\_freq=df1[["StockCode","Description"]].value\_counts().reset\_index()  
most\_freq

	StockCode	Description	count
0	85123A	WHITE HANGING HEART T-LIGHT HOLDER	2302
1	22423	REGENCY CAKESTAND 3 TIER	2200
2	85099B	JUMBO BAG RED RETROSPOT	2159
3	47566	PARTY BUNTING	1727
4	20725	LUNCH BAG RED RETROSPOT	1638
...	...	...	...
4787	35954	counted	1
4788	35954	check	1
4789	35923	check	1
4790	35915C	damaged	1
4791	m	Manual	1

4792 rows x 3 columns

Next steps:

Generate code with most\_freq

View recommended plots

New interactive sheet

```
most_freq.columns = ["StockCode", "freqDescription", "count"]  
most_freq
```

	StockCode	freqDescription	count
0	85123A	WHITE HANGING HEART T-LIGHT HOLDER	2302
1	22423	REGENCY CAKESTAND 3 TIER	2200
2	85099B	JUMBO BAG RED RETROSPOT	2159
3	47566	PARTY BUNTING	1727
4	20725	LUNCH BAG RED RETROSPOT	1638
...	...	...	...
4787	35954	counted	1
4788	35954	check	1
4789	35923	check	1
4790	35915C	damaged	1
4791	m	Manual	1

4792 rows x 3 columns


Next steps:

Generate code with most\_freq


View recommended plots

New interactive sheet

```
most_freq = most_freq.groupby("StockCode").first().reset_index()  
  
df2 = df1.merge(most_freq, on="StockCode", how="left")  
df2
```



	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	freqDescription	count
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom	WHITE HANGING HEART T-LIGHT HOLDER	2302.0
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom	WHITE METAL LANTERN	328.0
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom	CREAM CUPID HEARTS COAT HANGER	293.0
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom	KNITTED UNION FLAG HOT WATER BOTTLE	473.0
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom	RED WOOLLY HOTTIE WHITE HEART.	449.0
...	...	...	...	...	...	...	...	...	...	...
541904	581587	22613	PACK OF 20 SPACEBOY NAPKINS	12	2011-12-09 12:50:00	0.85	12680.0	France	PACK OF 20 SPACEBOY NAPKINS	148.0
541905	581587	22899	CHILDREN'S APRON DOLLY GIRL	6	2011-12-09 12:50:00	2.10	12680.0	France	CHILDREN'S APRON DOLLY GIRL	320.0
541906	581587	23254	CHILDRENS CUTLERY DOLLY GIRL	4	2011-12-09 12:50:00	4.15	12680.0	France	CHILDRENS CUTLERY DOLLY GIRL	307.0




```
df2.isnull().sum()
```



	0
InvoiceNo	0
StockCode	0
Description	1454
Quantity	0
InvoiceDate	0
UnitPrice	0
CustomerID	135080
Country	0
freqDescription	112
count	112



```
df2['Description']=df2['freqDescription']
df2
```



	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	freqDescription	count
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom	WHITE HANGING HEART T-LIGHT HOLDER	2302.0
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom	WHITE METAL LANTERN	328.0
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom	CREAM CUPID HEARTS COAT HANGER	293.0
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom	KNITTED UNION FLAG HOT WATER BOTTLE	473.0
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom	RED WOOLLY HOTTIE WHITE HEART.	449.0
...	...	...	...	...	...	...	...	...	...	...
541904	581587	22613	PACK OF 20 SPACEBOY NAPKINS	12	2011-12-09 12:50:00	0.85	12680.0	France	PACK OF 20 SPACEBOY NAPKINS	148.0
541905	581587	22899	CHILDREN'S APRON DOLLY GIRL	6	2011-12-09 12:50:00	2.10	12680.0	France	CHILDREN'S APRON DOLLY GIRL	320.0
541906	581587	23254	CHILDRENS CUTLERY DOLLY GIRL	4	2011-12-09 12:50:00	4.15	12680.0	France	CHILDRENS CUTLERY DOLLY GIRL	307.0

```
df2.isnull().sum()
```



	0
InvoiceNo	0
StockCode	0
Description	112
Quantity	0
InvoiceDate	0
UnitPrice	0
CustomerID	135080
Country	0
freqDescription	112
count	112


```
#now drop the records if the value is "NA"  
df2.dropna(subset=['Description'],inplace=True)  
df2.isnull().sum()
```



	0
InvoiceNo	0
StockCode	0
Description	0
Quantity	0
InvoiceDate	0
UnitPrice	0
CustomerID	134968
Country	0
freqDescription	0
count	0




df2.shape

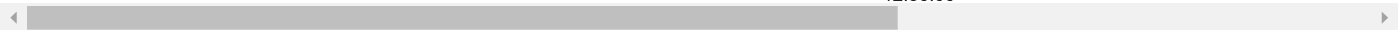


(541797, 10)


#now drop unwanted columns i.e freqdescription,count  
df2.drop(columns=['freqDescription','count'],inplace=True)  
df2




	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
...	...	...	...	...	...	...	...	...
541904	581587	22613	PACK OF 20 SPACEBOY NAPKINS	12	2011-12-09 12:50:00	0.85	12680.0	France
541905	581587	22899	CHILDREN'S APRON DOLLY GIRL	6	2011-12-09 12:50:00	2.10	12680.0	France



df2.describe()



	Quantity	InvoiceDate	UnitPrice	CustomerID
count	541797.000000	541797	541797.000000	406829.000000
mean	9.555919	2011-07-04 14:06:48.671255296	4.612067	15287.690570
min	-80995.000000	2010-12-01 08:26:00	-11062.060000	12346.000000
25%	1.000000	2011-03-28 11:36:00	1.250000	13953.000000
50%	3.000000	2011-07-20 08:59:00	2.080000	15152.000000
75%	10.000000	2011-10-19 11:41:00	4.130000	16791.000000
max	80995.000000	2011-12-09 12:50:00	38970.000000	18287.000000
std	218.103428	NaN	96.769831	1713.600303



df2[df2.Quantity<0]

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
141	C536379	D	Discount	-1	2010-12-01 09:41:00	27.50	14527.0	United Kingdom
154	C536383	35004C	SET OF 3 COLOURED FLYING DUCKS	-1	2010-12-01 09:49:00	4.65	15311.0	United Kingdom
235	C536391	22556	PLASTERS IN TIN CIRCUS PARADE	-12	2010-12-01 10:24:00	1.65	17548.0	United Kingdom
236	C536391	21984	PACK OF 12 PINK PAISLEY TISSUES	-24	2010-12-01 10:24:00	0.29	17548.0	United Kingdom
237	C536391	21983	PACK OF 12 BLUE PAISLEY TISSUES	-24	2010-12-01 10:24:00	0.29	17548.0	United Kingdom
...	...	...	...	...	...	...	...	...
540449	C581490	23144	ZINC T-LIGHT HOLDER STARS SMALL	-11	2011-12-09 09:57:00	0.83	14397.0	United Kingdom
541541	C581499	M	Manual	-1	2011-12-09 10:28:00	224.69	15498.0	United Kingdom

```
df2[df2.UnitPrice<0]
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
299983	A563186	B	Adjust bad debt	1	2011-08-12 14:51:00	-11062.06	NaN	United Kingdom
299984	A563187	B	Adiust bad debt	1	2011-08-12 14:52:00	-11062.06	NaN	United Kinadom

```
#now remove quantity<0 and unitprice<0 from the dataset
#solution
df3=df2[(df2.Quantity>0) & (df2.UnitPrice>0)]
df3.describe()
```

	Quantity	InvoiceDate	UnitPrice	CustomerID
count	530104.000000	530104	530104.000000	397884.000000
mean	10.542037	2011-07-04 20:16:05.225087744	3.907625	15294.423453
min	1.000000	2010-12-01 08:26:00	0.001000	12346.000000
25%	1.000000	2011-03-28 12:22:00	1.250000	13969.000000
50%	3.000000	2011-07-20 12:58:00	2.080000	15159.000000
75%	10.000000	2011-10-19 12:39:00	4.130000	16795.000000
max	80995.000000	2011-12-09 12:50:00	13541.330000	18287.000000
std	155.524124	NaN	35.915681	1713.141560

```
df3.Quantity.quantile(0.9999)
```

```
1439.87639999990188
```

```
#feature engineering to add new columns
# in the dataset we have quantity and unitprice so we can solve saleamount
copy=df3.copy()
```

```
copy['TotalSales'] = copy['Quantity'] * copy['UnitPrice']
copy
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	TotalSales	
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom	15.30	
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom	20.34	
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom	22.00	
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom	20.34	
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom	20.34	
...	...	...	...	...	...	...	...	...	...	
541904	581587	22613	PACK OF 20 SPACEBOY NAPKINS	12	2011-12-09 12:50:00	0.85	12680.0	France	10.20	
541905	581587	22899	CHILDREN'S APRON DOLLY GIRL	6	2011-12-09 12:50:00	2.10	12680.0	France	12.60	

```
# Extract the month
copy['Month'] = copy['InvoiceDate'].dt.month
copy
```

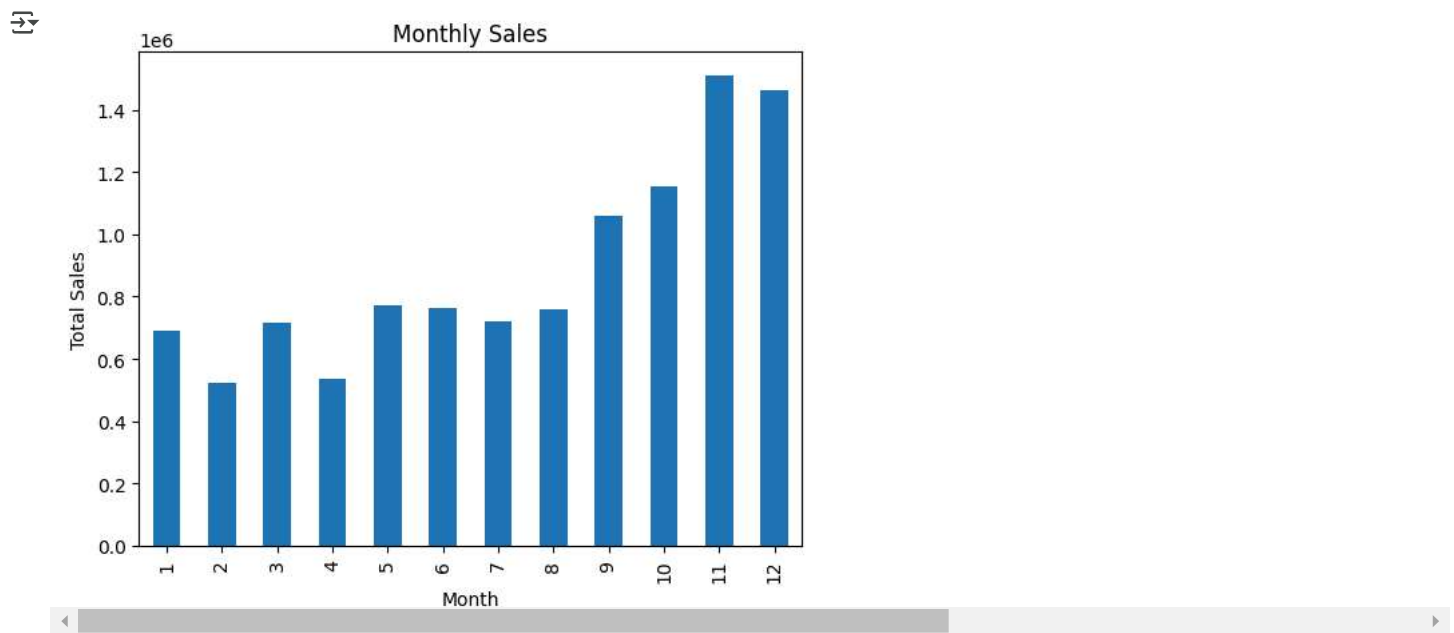
	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	TotalSales	Month
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom	15.30	12
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom	20.34	12
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom	22.00	12
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom	20.34	12
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom	20.34	12
...	...	...	...	...	...	...	...	...	...	...
541904	581587	22613	PACK OF 20 SPACEBOY NAPKINS	12	2011-12-09 12:50:00	0.85	12680.0	France	10.20	12
541905	581587	22899	CHILDREN'S APRON	6	2011-12-09 12:50:00	2.10	12680.0	France	12.60	12

## ✓ visualizattion and EDA

### 1.plot monthly sales

```
monthliesales=copy.groupby('Month')['TotalSales'].sum()
monthliesales.plot(kind='bar',title='Monthly Sales')
plt.xlabel('Month')
plt.ylabel('Total Sales')
plt.show()
```

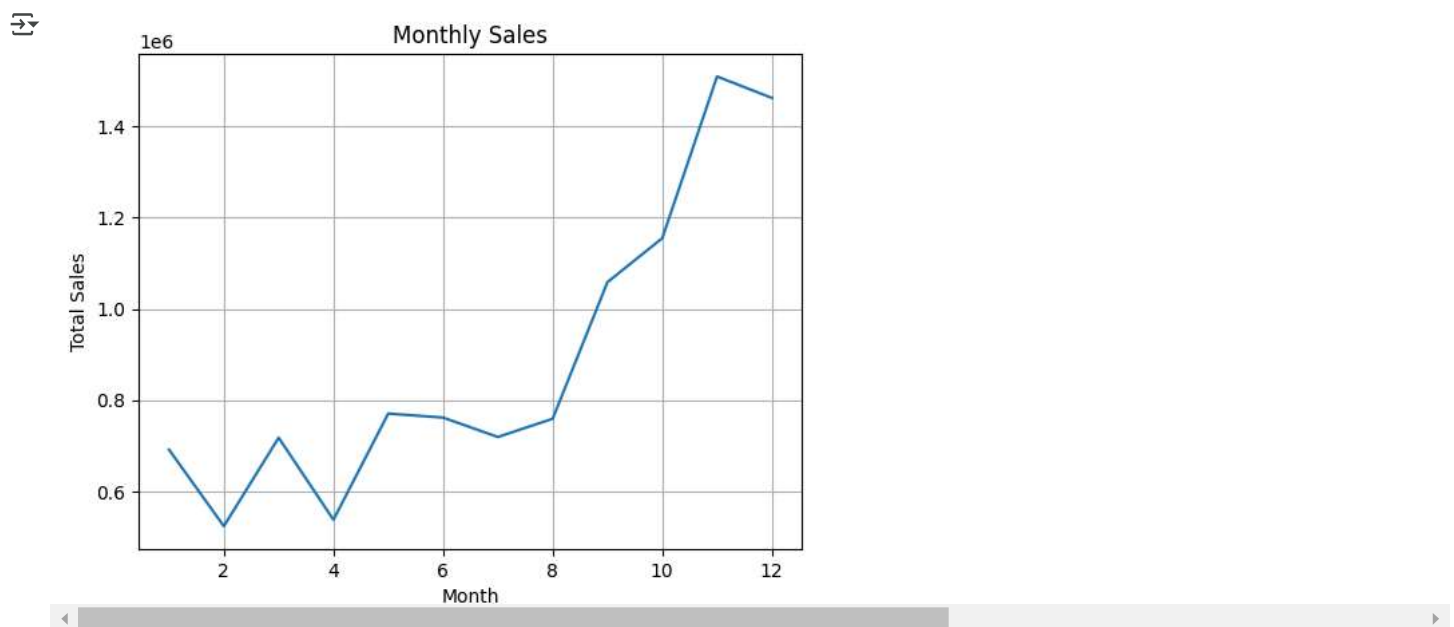




```

monthlsales = copy.groupby('Month')['TotalSales'].sum()
monthlsales.plot(kind='line', title='Monthly Sales') # Changed to line plot
plt.xlabel('Month')
plt.ylabel('Total Sales')
plt.grid(True)
plt.show()

```



## insights

total sales rising up from august and having a peak in november. most likely due to holiday season at end of year.

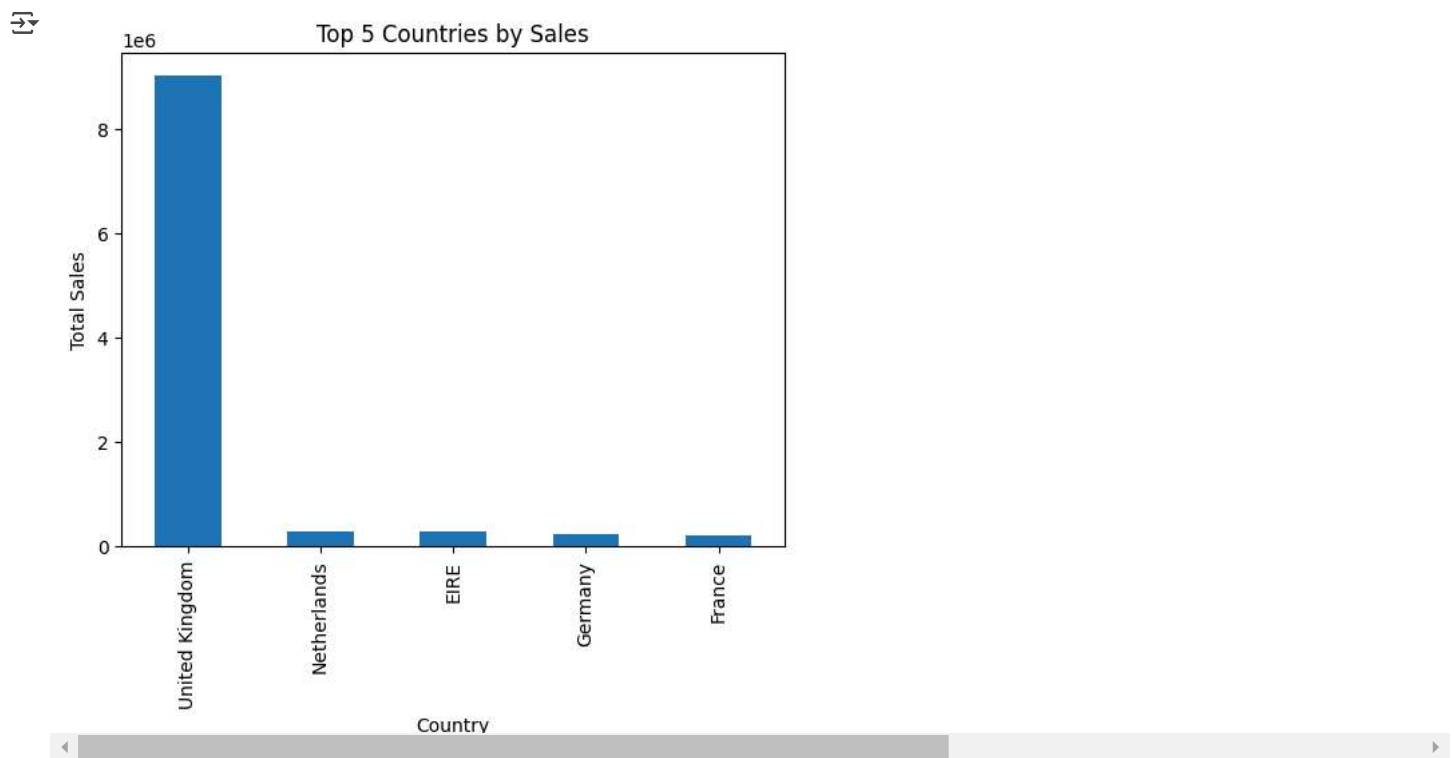
## Top 5 countries by sales

```

top_5_countries = copy.groupby('Country')['TotalSales'].sum().nlargest(5)
top_5_countries.plot(kind='bar', title='Top 5 Countries by Sales')
plt.xlabel('Country')
plt.ylabel('Total Sales')

plt.show()

```

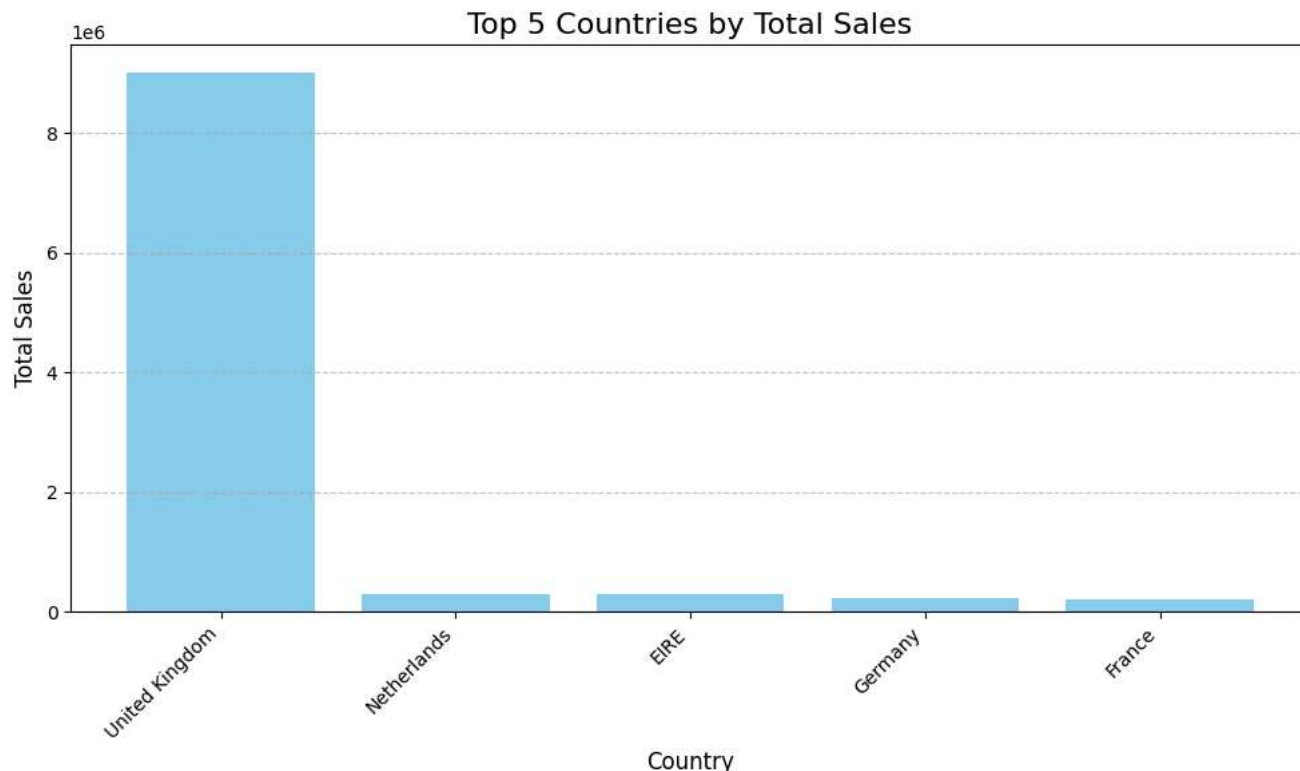


```
top_5_countries = copy.groupby('Country')['TotalSales'].sum().nlargest(5)
top_5_countries
```

	TotalSales
Country	
United Kingdom	9025222.084
Netherlands	285446.340
EIRE	283453.960
Germany	228867.140
France	209715.110

```
top_5_countries = copy.groupby('Country')['TotalSales'].sum().nlargest(5)

plt.figure(figsize=(10, 6))
plt.bar(top_5_countries.index, top_5_countries.values, color='skyblue')
plt.title('Top 5 Countries by Total Sales', fontsize=16)
plt.xlabel('Country', fontsize=12)
plt.ylabel('Total Sales', fontsize=12)
plt.xticks(rotation=45, ha='right')
plt.yticks(fontsize=10)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```



Generate

get the percentage of top\_5\_countries and plot it in horizontal bar graph with data label as on and result in the percentage



Close

< 1 of 1 > [Undo Changes](#) [Use code with caution](#)

```
import matplotlib.pyplot as plt
total_sales = top_5_countries.sum()
percentages = (top_5_countries / total_sales) * 100
plt.figure(figsize=(10, 6))
bars = plt.barh(percentages.index, percentages.values, color='skyblue')
for bar, percentage in zip(bars, percentages):
    plt.text(bar.get_width() + 0.5, bar.get_y() + bar.get_height()/2, f'{percentage:.1f}%', va='center')

plt.title('Top 5 Countries by Total Sales Percentage', fontsize=16)
plt.xlabel('Percentage of Total Sales', fontsize=12)
plt.ylabel('Country', fontsize=12)
plt.xlim(0, percentages.max() * 1.1)
plt.tight_layout()
plt.show()
```



Top 5 Countries by Total Sales Percentage



Generate

get the same with stockcode and plot the horizontal bar graph with percentage of overall contribution to total



Close

< 1 of 1 > [Undo Changes](#) [Use code with caution](#)

```
top_products = copy.groupby('StockCode')['TotalSales'].sum().nlargest(5)
total_sales = top_products.sum()
percentages = (top_products / total_sales) * 100

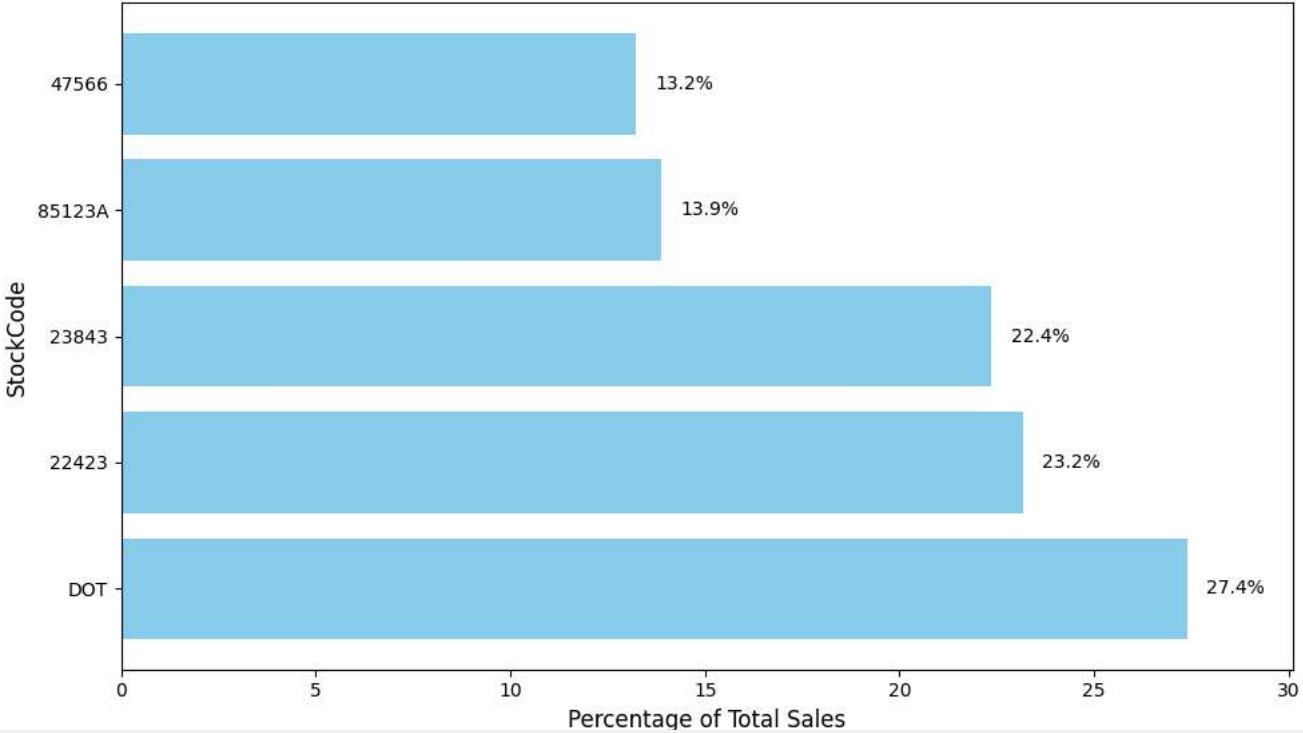
plt.figure(figsize=(10, 6))
bars = plt.barh(percentages.index, percentages.values, color='skyblue')

for bar, percentage in zip(bars, percentages):
    plt.text(bar.get_width() + 0.5, bar.get_y() + bar.get_height()/2, f'{percentage:.1f}%', va='center')

plt.title('Top 5 Products by Total Sales Percentage', fontsize=16)
plt.xlabel('Percentage of Total Sales', fontsize=12)
plt.ylabel('StockCode', fontsize=12)
plt.xlim(0, percentages.max() * 1.1)
plt.tight_layout()
plt.show()
```



Top 5 Products by Total Sales Percentage



Start coding or [generate](#) with AI.

4.RFM ANALYSIS

```
current_dt =copy[' InvoiceDate'].max()+pd.DateOffset(days=1)
current_dt
```



Timestamp('2011-12-10 12:50:00')

```
rfm=copy.groupby("CustomerID").agg({
    "InvoiceDate": lambda date: (current_dt - date.max()).days,
    "InvoiceNo": "count",
    "TotalSales": "sum"
})
rfm.columns = ['Recency', 'frequency', 'monetary']
rfm
```

	Recency	frequency	monetary	
CustomerID				
12346.0	326	1	77183.60	
12347.0	2	182	4310.00	
12348.0	75	31	1797.24	
12349.0	19	73	1757.55	
12350.0	310	17	334.40	
...	...	...	...	
18280.0	278	10	180.60	
18281.0	181	7	80.82	
18282.0	8	12	178.05	
18283.0	4	756	2094.88	
18287.0	43	70	1837.28	

4338 rows x 3 columns

Next steps:

Generate code with rfm

View recommended plots

New interactive sheet

```
copy[copy.CustomerID==12349]
#for the customerid=12349 the recent transaction was 19days ago
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	TotalSales	Month	
485502	577609	23112	PARISIENNE CURIO CABINET	2	2011-11-21 09:51:00	7.50	12349.0	Italy	15.00	11	
485503	577609	23460	SWEETHEART WALL TIDY	2	2011-11-21 09:51:00	9.95	12349.0	Italy	19.90	11	
485504	577609	21564	PINK HEART SHAPE LOVE BUCKET	6	2011-11-21 09:51:00	2.95	12349.0	Italy	17.70	11	
485505	577609	21411	GINGHAM HEART DOORSTOP RED	3	2011-11-21 09:51:00	4.25	12349.0	Italy	12.75	11	
485506	577609	21563	RED HEART SHAPE LOVE BUCKET	6	2011-11-21 09:51:00	2.95	12349.0	Italy	17.70	11	
...	...	...	...	...	...	...	...	...	...	...	
485570	577609	22064	PINK DOUGHNUT TRINKET POT	12	2011-11-21 09:51:00	1.65	12349.0	Italy	19.80	11	
485571	577609	37448	CERAMIC CAKE DESIGN SPOTTED MUG	12	2011-11-21 09:51:00	1.49	12349.0	Italy	17.88	11	

```
copy[copy.CustomerID==12349]["TotalSales"].sum()
```