

Project Name - SUPERSTORE SALES

Author - SHIV CHETAN KUMAWAT

Import Python Libraries

```
In [102...]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [103...]: df = pd.read_csv('supermall.csv')
df.head()
```

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Count
0	1	CA-2017-152156	08/11/2017	11/11/2017	Second Class	CG-12520	Claire Gute	Consumer	United States
1	2	CA-2017-152156	08/11/2017	11/11/2017	Second Class	CG-12520	Claire Gute	Consumer	United States
2	3	CA-2017-138688	12/06/2017	16/06/2017	Second Class	DV-13045	Darrin Van Huff	Corporate	United States
3	4	US-2016-108966	11/10/2016	18/10/2016	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States
4	5	US-2016-108966	11/10/2016	18/10/2016	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States

```
In [104...]: df.shape
```

```
Out[104...]: (9800, 18)
```

In [105... df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9800 entries, 0 to 9799
Data columns (total 18 columns):
 #   Column            Non-Null Count  Dtype  
--- 
 0   Row ID             9800 non-null   int64  
 1   Order ID           9800 non-null   object  
 2   Order Date         9800 non-null   object  
 3   Ship Date          9800 non-null   object  
 4   Ship Mode          9800 non-null   object  
 5   Customer ID        9800 non-null   object  
 6   Customer Name      9800 non-null   object  
 7   Segment             9800 non-null   object  
 8   Country             9800 non-null   object  
 9   City                9800 non-null   object  
 10  State               9800 non-null   object  
 11  Postal Code        9789 non-null   float64 
 12  Region              9800 non-null   object  
 13  Product ID          9800 non-null   object  
 14  Category            9800 non-null   object  
 15  Sub-Category        9800 non-null   object  
 16  Product Name        9800 non-null   object  
 17  Sales               9800 non-null   float64 
dtypes: float64(2), int64(1), object(15)
memory usage: 1.3+ MB
```

In [106... df.isnull().sum()

```
Out[106... Row ID            0
Order ID           0
Order Date         0
Ship Date          0
Ship Mode          0
Customer ID        0
Customer Name      0
Segment             0
Country             0
City                0
State               0
Postal Code        11
Region              0
Product ID          0
Category            0
Sub-Category        0
Product Name        0
Sales               0
dtype: int64
```

In [107... df.dropna(inplace = True)

In [108... df.shape

```
Out[108... (9789, 18)
```

In [109... df.info()

```
<class 'pandas.core.frame.DataFrame'>
Index: 9789 entries, 0 to 9799
Data columns (total 18 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   Row ID            9789 non-null    int64  
 1   Order ID          9789 non-null    object  
 2   Order Date         9789 non-null    object  
 3   Ship Date          9789 non-null    object  
 4   Ship Mode          9789 non-null    object  
 5   Customer ID        9789 non-null    object  
 6   Customer Name      9789 non-null    object  
 7   Segment             9789 non-null    object  
 8   Country             9789 non-null    object  
 9   City                9789 non-null    object  
 10  State               9789 non-null    object  
 11  Postal Code        9789 non-null    float64 
 12  Region              9789 non-null    object  
 13  Product ID          9789 non-null    object  
 14  Category             9789 non-null    object  
 15  Sub-Category         9789 non-null    object  
 16  Product Name         9789 non-null    object  
 17  Sales                9789 non-null    float64 
dtypes: float64(2), int64(1), object(15)
memory usage: 1.4+ MB
```

In [110... df['Order Date']

```
Out[110... 0      08/11/2017
 1      08/11/2017
 2      12/06/2017
 3      11/10/2016
 4      11/10/2016
 ...
 9795  21/05/2017
 9796  12/01/2016
 9797  12/01/2016
 9798  12/01/2016
 9799  12/01/2016
Name: Order Date, Length: 9789, dtype: object
```

create new column

In [111... df['Order Year']=df['Order Date'].str[-4:]
df['Order Year']

```
Out[111... 0      2017
           1      2017
           2      2017
           3      2016
           4      2016
           ...
          9795    2017
          9796    2016
          9797    2016
          9798    2016
          9799    2016
Name: Order Year, Length: 9789, dtype: object
```

```
In [112... df.duplicated().sum()
```

```
Out[112... np.int64(0)
```

```
In [113... df.drop_duplicates(inplace = True)
```

```
In [114... df.head()
```

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Count
0	1	CA-2017-152156	08/11/2017	11/11/2017	Second Class	CG-12520	Claire Gute	Consumer	United States
1	2	CA-2017-152156	08/11/2017	11/11/2017	Second Class	CG-12520	Claire Gute	Consumer	United States
2	3	CA-2017-138688	12/06/2017	16/06/2017	Second Class	DV-13045	Darrin Van Huff	Corporate	United States
3	4	US-2016-108966	11/10/2016	18/10/2016	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States
4	5	US-2016-108966	11/10/2016	18/10/2016	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States

```
In [115... df['Sales'] = df['Sales'].round(2)
df.head()
```

Out[115...]

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Count
0	1	CA-2017-152156	08/11/2017	11/11/2017	Second Class	CG-12520	Claire Gute	Consumer	United States
1	2	CA-2017-152156	08/11/2017	11/11/2017	Second Class	CG-12520	Claire Gute	Consumer	United States
2	3	CA-2017-138688	12/06/2017	16/06/2017	Second Class	DV-13045	Darrin Van Huff	Corporate	United States
3	4	US-2016-108966	11/10/2016	18/10/2016	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States
4	5	US-2016-108966	11/10/2016	18/10/2016	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States

◀ ▶

Removing Outliers by the basis of Total Sales

In [116...]

```
Q1 = df['Sales'].quantile(0.25)
Q3 = df['Sales'].quantile(0.75)
```

In [117...]

Q1

Out[117...]

np.float64(17.25)

In [118...]

Q3

Out[118...]

np.float64(210.39)

In [119...]

```
IQR = Q3-Q1
IQR
```

Out[119...]

np.float64(193.14)

In [120...]

```
lower_bound = Q1-1.5*IQR
lower_bound
```

```
Out[120... np.float64(-272.46)
```

```
In [121... upper_bound = Q3 + 1.5*IQR  
upper_bound
```

```
Out[121... np.float64(500.0999999999997)
```

```
In [122... df_no_outliers = df[(df['Sales']>=lower_bound) & (df['Sales']<=upper_bound)]
```

```
In [123... print("Before:", df.shape)  
print("After:", df_no_outliers.shape)
```

Before: (9789, 19)

After: (8648, 19)

```
In [124... df=df_no_outliers
```

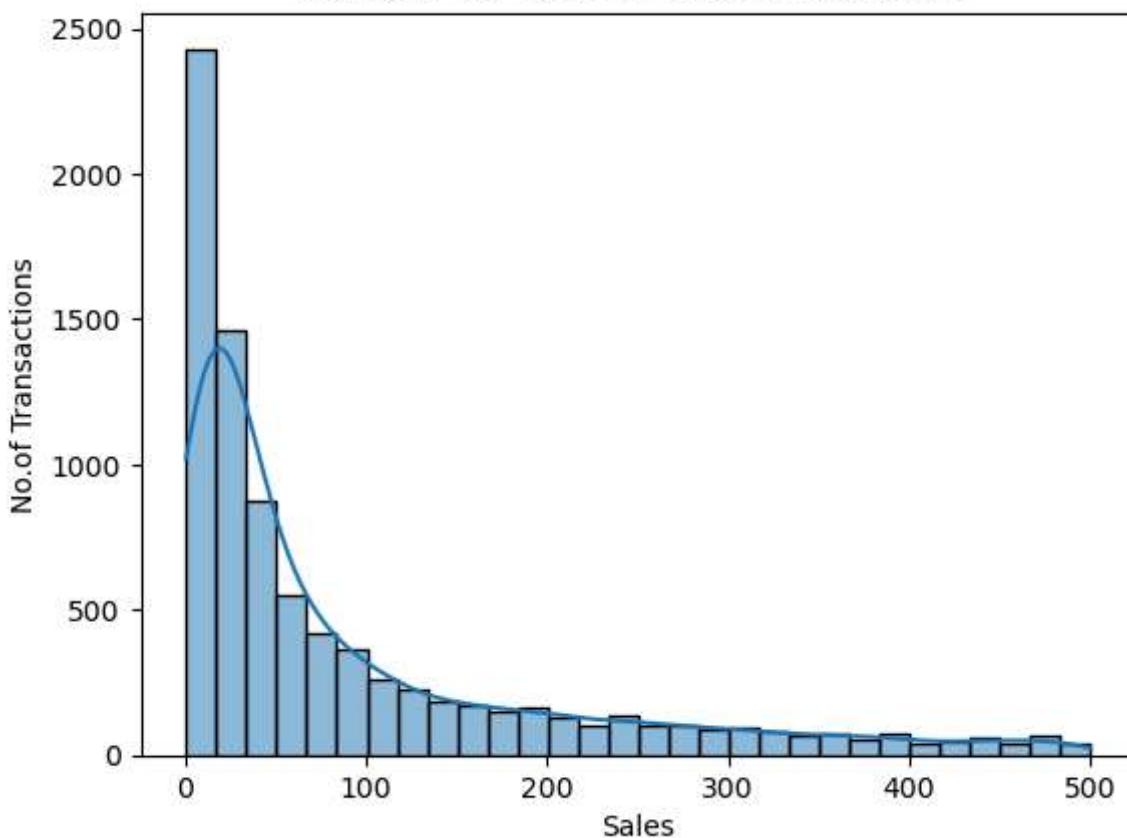
EXPLORATORY DATA ANALYSIS (EDA)

DATA ANALYSIS (YEAR - 2015, 2016, 2017, 2018)

1. SALES DISTRIBUTION

```
In [125... sns.histplot(df['Sales'], bins = 30,kde = True)  
plt.title("Chart of Sales Distribution ",fontsize = 18)  
plt.ylabel("No.of Transactions")  
plt.show()
```

Chart of Sales Distribution



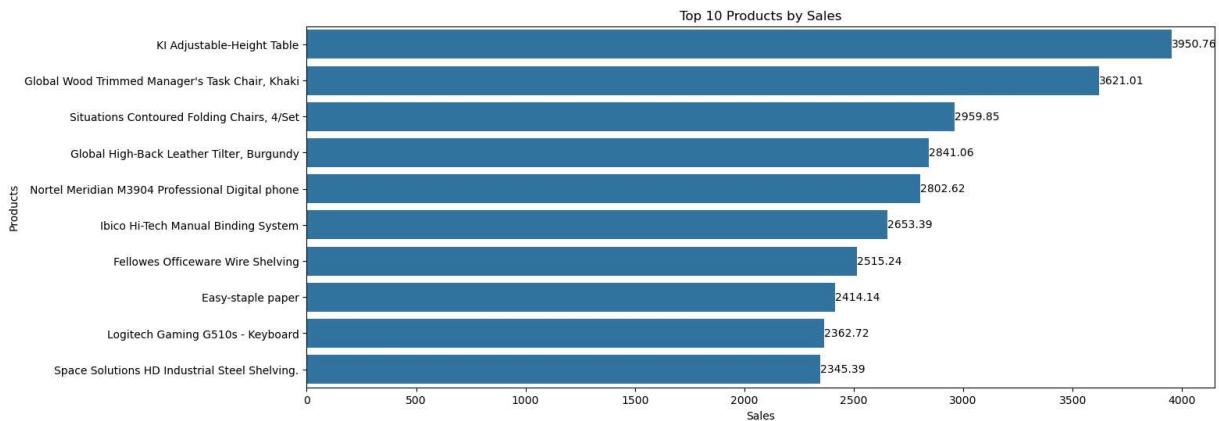
2. Top 10 Products by Sales

```
In [126]: product_sale = df.groupby(['Product Name'],as_index = False)[['Sales']].sum().sort_values('Sales', ascending=False)
```

```
Out[126]:
```

	Product Name	Sales
879	KI Adjustable-Height Table	3950.76
732	Global Wood Trimmed Manager's Task Chair, Khaki	3621.01
1363	Situations Contoured Folding Chairs, 4/Set	2959.85
715	Global High-Back Leather Tilter, Burgundy	2841.06
1099	Nortel Meridian M3904 Professional Digital phone	2802.62
835	Ibico Hi-Tech Manual Binding System	2653.39
615	Fellowes Officeware Wire Shelving	2515.24
507	Easy-staple paper	2414.14
956	Logitech Gaming G510s - Keyboard	2362.72
1387	Space Solutions HD Industrial Steel Shelving.	2345.39

```
In [127...]  
plt.figure(figsize =(15,6))  
ax = sns.barplot(data = product_sale , x = 'Sales',y = 'Product Name')  
for bars in ax.containers:  
    ax.bar_label(bars)  
plt.title("Top 10 Products by Sales")  
plt.ylabel("Products")  
plt.show()
```

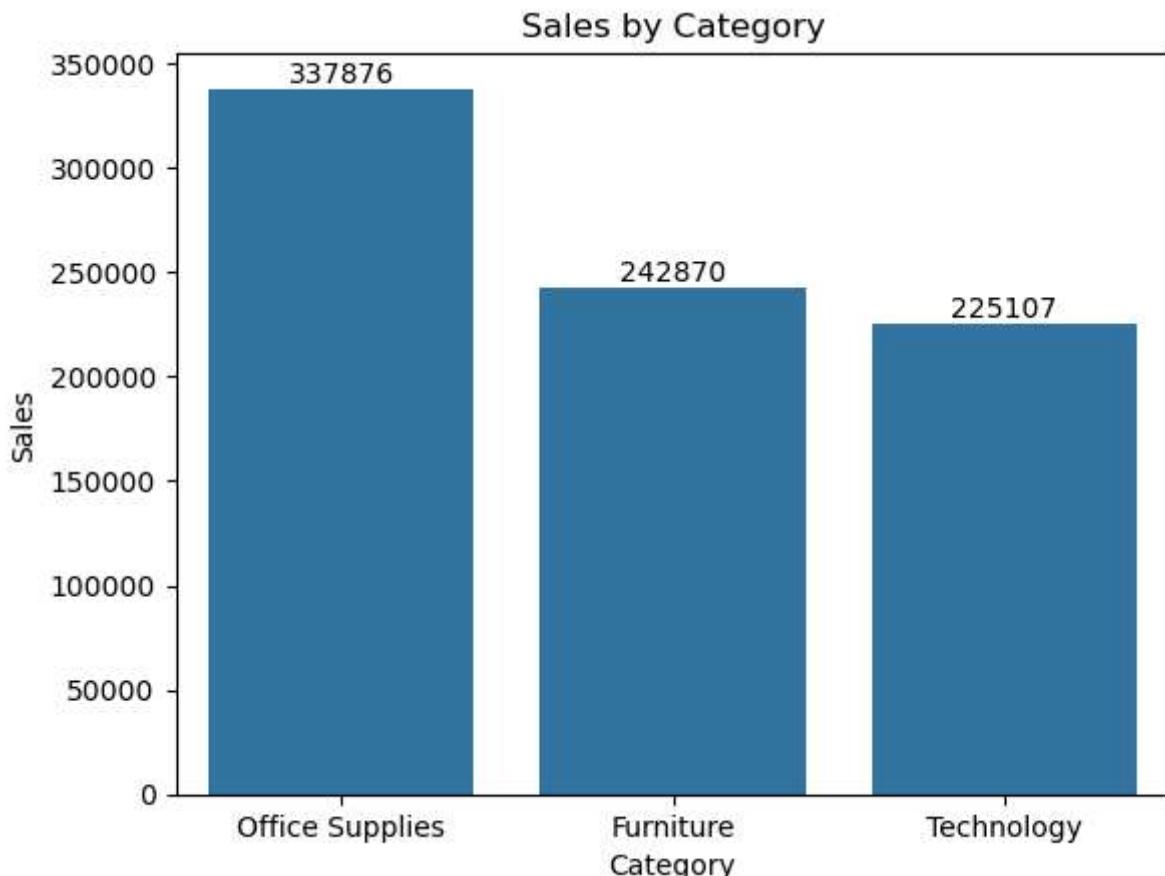


3. Sales by Category

```
In [128...]  
category_sale = df.groupby(['Category'],as_index = False)[['Sales']].sum().sort_values  
category_sale
```

	Category	Sales
1	Office Supplies	337876.32
0	Furniture	242870.18
2	Technology	225107.18

```
In [129...]  
ax = sns.barplot(data = category_sale, y = 'Sales',x = 'Category')  
for bars in ax.containers:  
    ax.bar_label(bars)  
plt.title("Sales by Category")  
plt.show()
```



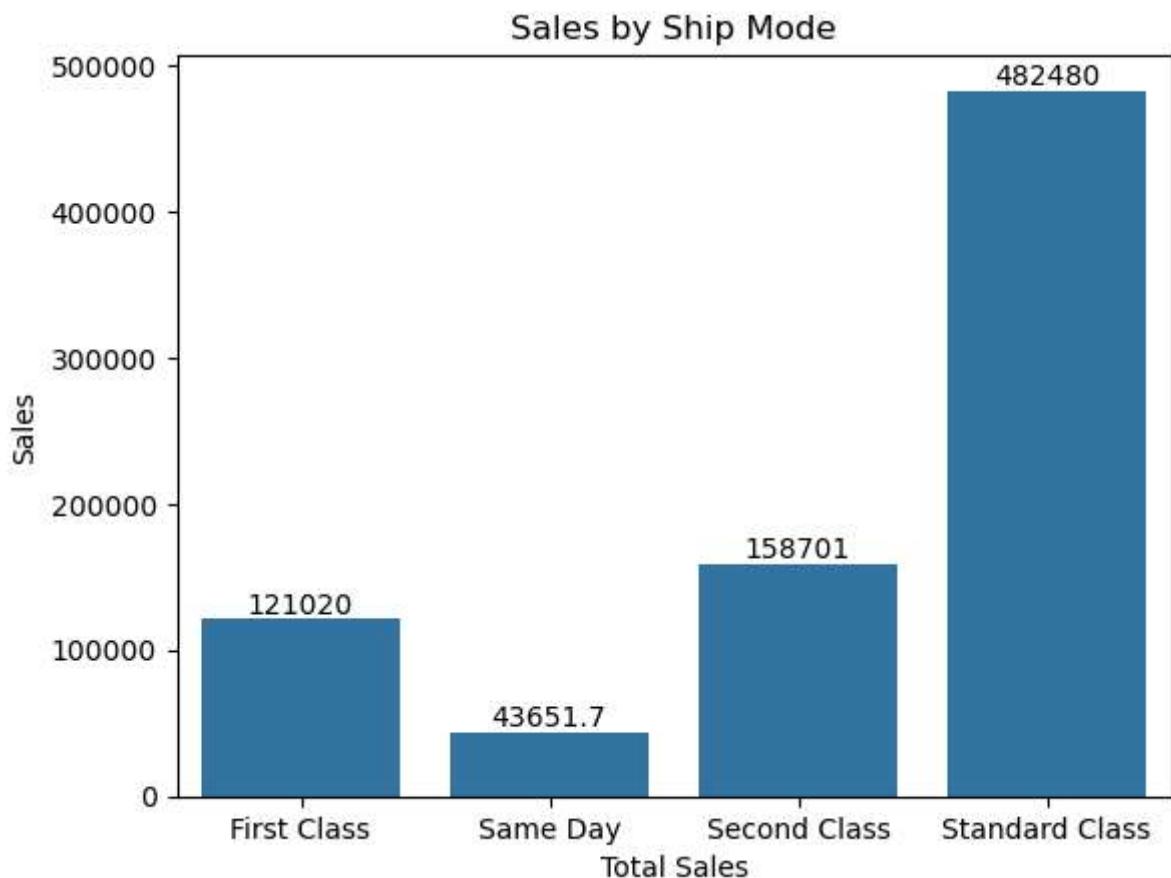
4. Sales by Ship Mode

```
In [130...]: shiping_sales = df.groupby(['Ship Mode'], as_index = False)[['Sales']].sum()  
shiping_sales
```

```
Out[130...]:
```

	Ship Mode	Sales
0	First Class	121020.31
1	Same Day	43651.67
2	Second Class	158701.46
3	Standard Class	482480.24

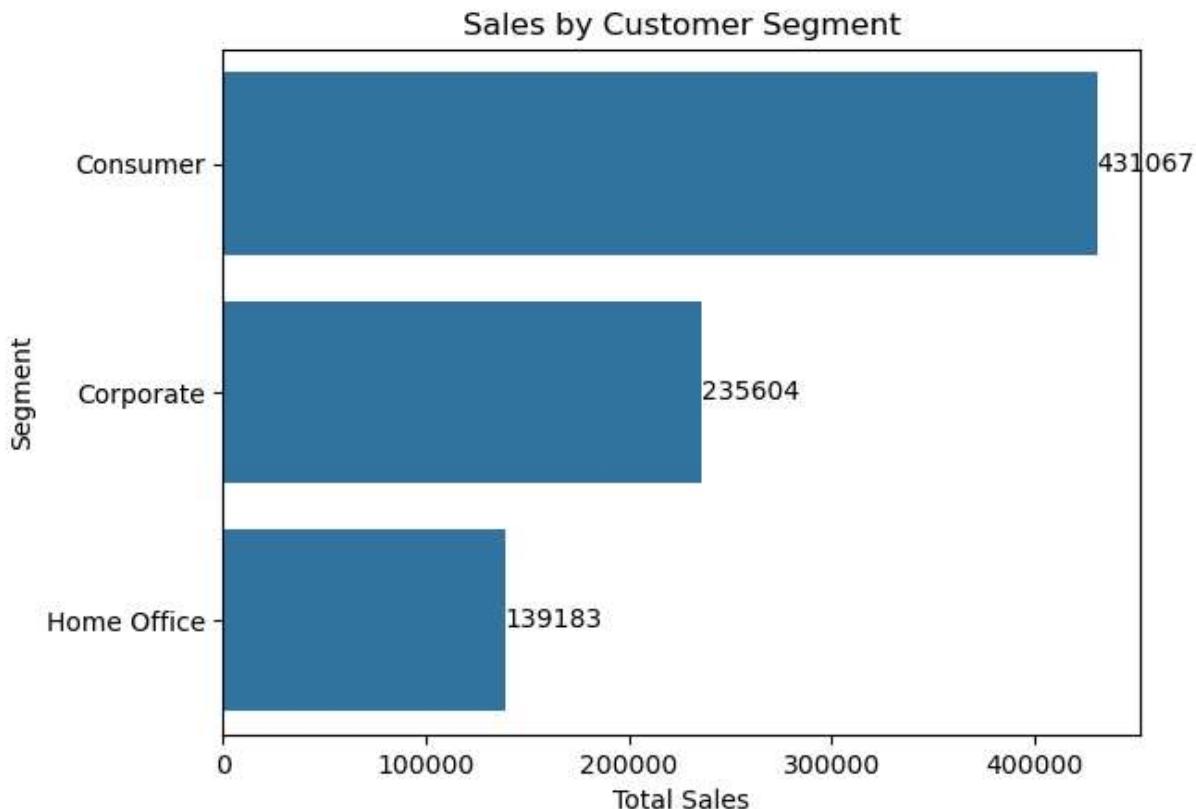
```
In [131...]: ax = sns.barplot(data = shiping_sales,x = 'Ship Mode',y = 'Sales')  
for bars in ax.containers:  
    ax.bar_label(bars)  
plt.title('Sales by Ship Mode')  
plt.xlabel('Total Sales')  
plt.show()
```



5. Sales by Customer Segment

In [132...]

```
segment_sales = df.groupby(['Segment'], as_index = False)[['Sales']].sum()
ax = sns.barplot(data = segment_sales, x = 'Sales', y = 'Segment')
for bars in ax.containers:
    ax.bar_label(bars)
plt.title('Sales by Customer Segment')
plt.xlabel('Total Sales')
plt.show()
```



6. Top 5 States by Sales

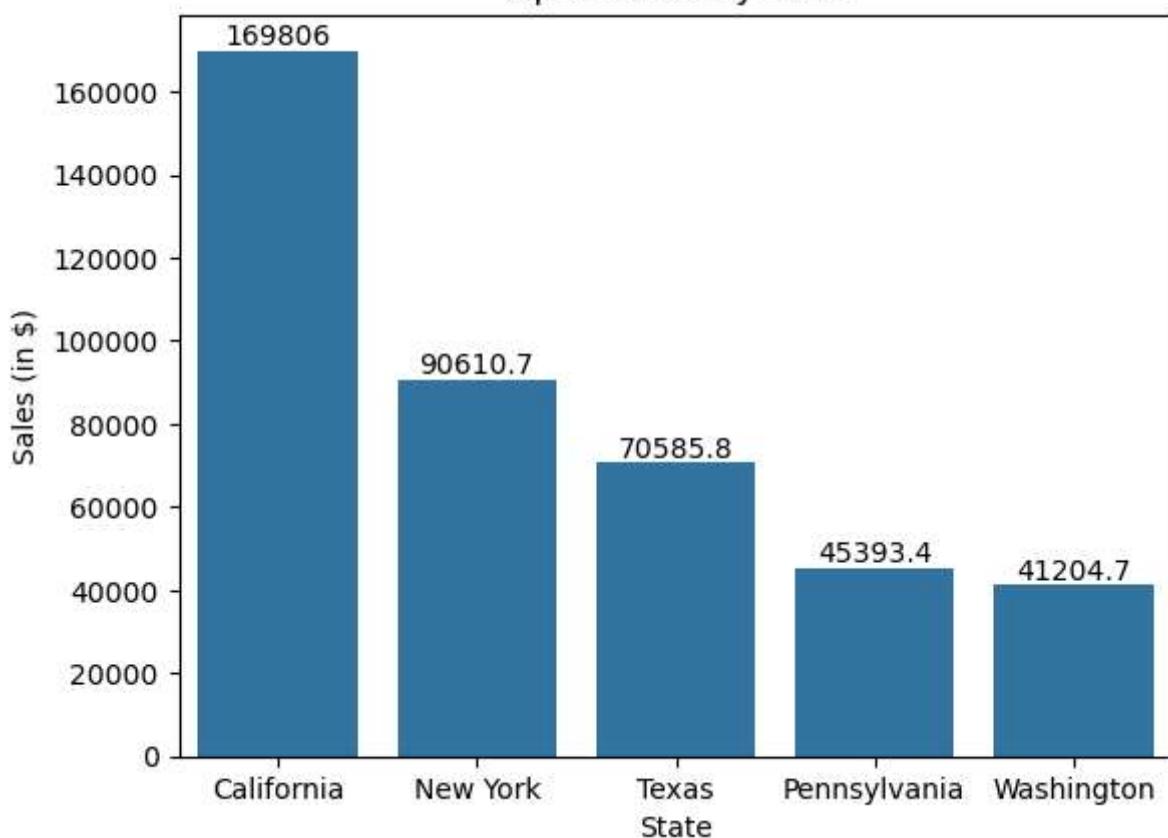
```
In [133]: state_sales = df.groupby(['State'], as_index = False)[['Sales']].sum().sort_values(by='Sales').head()
```

```
Out[133]:
```

	State	Sales
3	California	169805.89
30	New York	90610.73
41	Texas	70585.76
36	Pennsylvania	45393.43
44	Washington	41204.65

```
In [134]: ax = sns.barplot(data = state_sales.head(), x = 'State', y = 'Sales')
for bars in ax.containers:
    ax.bar_label(bars)
plt.title('Top 5 States by Sales')
plt.ylabel('Sales (in $)')
plt.show()
```

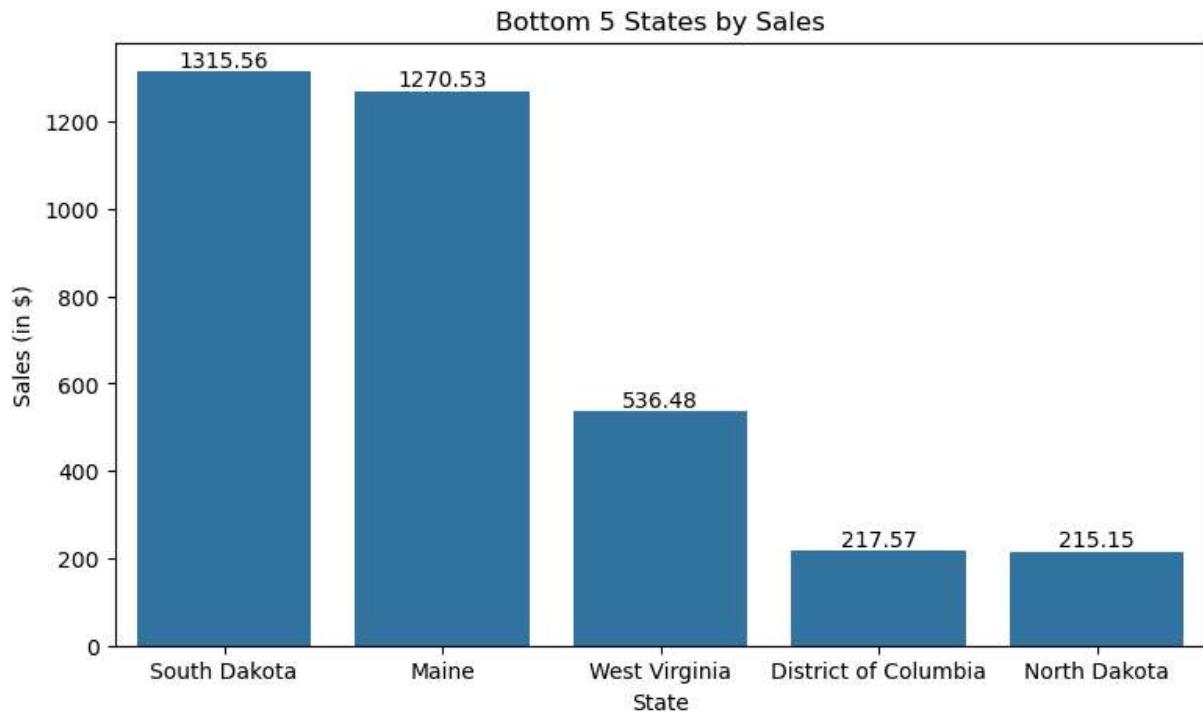
Top 5 States by Sales



7. Bottom 5 States by Sales

In [135...]

```
plt.figure(figsize = (9,5))
ax = sns.barplot(data = state_sales.tail(),x = 'State',y = 'Sales')
for bars in ax.containers:
    ax.bar_label(bars)
plt.title('Bottom 5 States by Sales')
plt.ylabel('Sales (in $)')
plt.show()
```



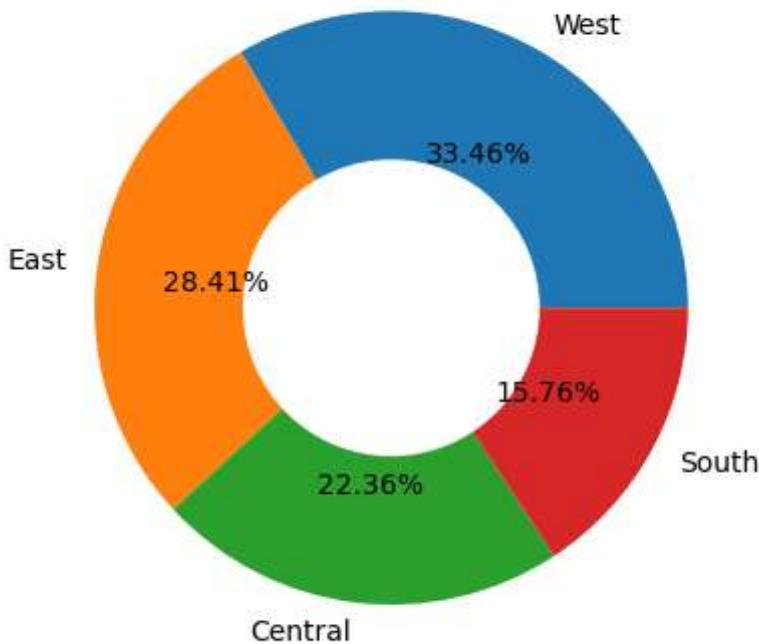
8. Sales by Region

```
In [136...]: sales_region = df.groupby(['Region'], as_index = False)[['Sales']].sum().sort_values(b...  
sales_region
```

```
Out[136...]:
```

	Region	Sales
3	West	269677.07
1	East	228944.71
0	Central	180207.32
2	South	127024.58

```
In [137...]: plt.pie(sales_region['Sales'], labels = sales_region['Region'], autopct = '%0.2f%%')  
plt.pie([1], radius = 0.5, colors = "w")  
plt.show()
```



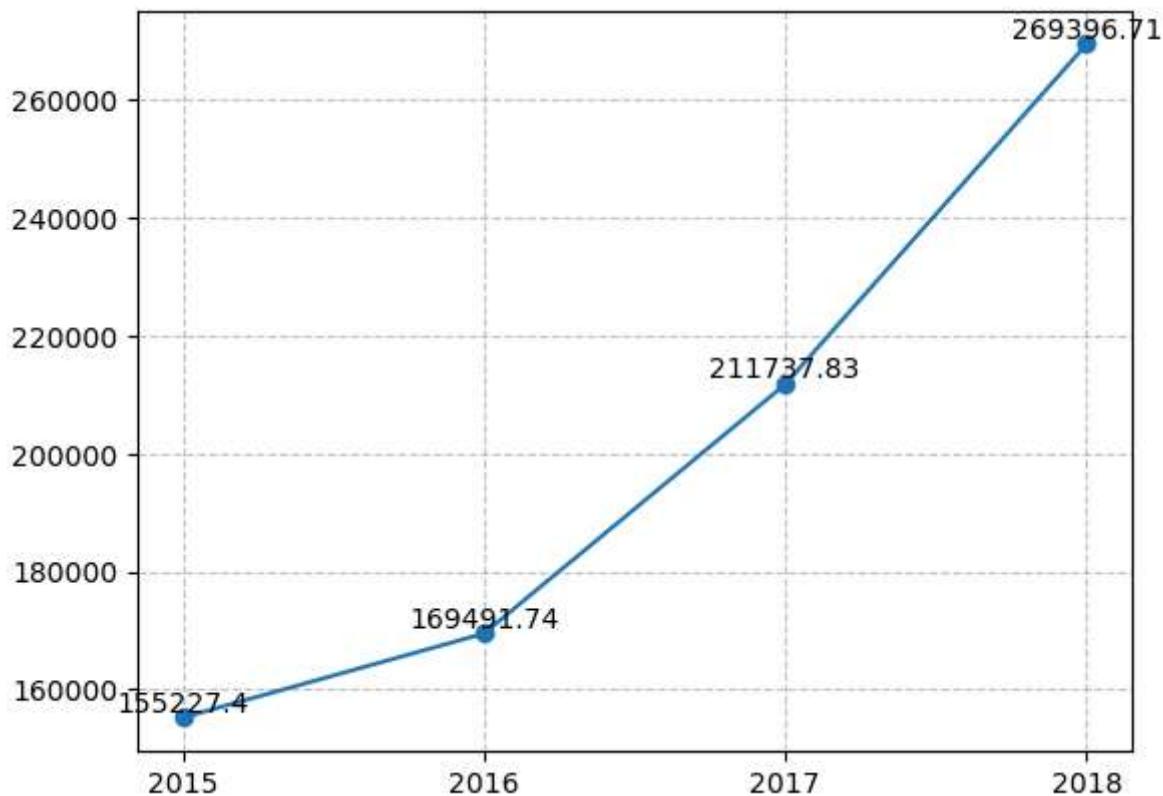
9. Sales by Year

```
In [138...]: sales_year = df.groupby(['Order Year'], as_index = False)['Sales'].sum().sort_values(sales_year)
```

Out[138...]:

	Order Year	Sales
0	2015	155227.40
1	2016	169491.74
2	2017	211737.83
3	2018	269396.71

```
In [139...]: x = sales_year['Order Year']
y = sales_year['Sales']
plt.plot(x,y,marker = 'o')
plt.grid(True, linestyle='--', alpha=0.7)
for i, (xi, yi) in enumerate(zip(x, y)):
    plt.text(xi, yi + 0.2, str(yi), ha='center', va='bottom', fontsize=10)
```



Separate data for year wise (2015, 2016, 2017, 2018)

```
In [140]: df_2017 = df[df['Order Year'] == '2017']
```

```
In [141]: df_2015 = df[df['Order Year'] == '2015']
```

```
In [142]: df_2016 = df[df['Order Year'] == '2016']
```

```
In [143]: df_2018 = df[df['Order Year'] == '2018']
```

```
In [144]: df_2017['Order Date'].head()
```

```
Out[144]: 0    08/11/2017
2    12/06/2017
13   05/12/2017
21   09/12/2017
22   09/12/2017
Name: Order Date, dtype: object
```

```
In [ ]:
```

Analysis - 2015

Month Column creation

In [145... df_2015['Month_no']=df_2015['Order Date'].str[-7:-5]

C:\Users\HP\AppData\Local\Temp\ipykernel_16268\3705148648.py:1: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

df_2015['Month_no']=df_2015['Order Date'].str[-7:-5]

In [146... df_2015['Month']=df_2015['Order Date'].str[-7:-5].replace({'01': 'Jan',

```
'02': 'Feb',
'03': 'Mar',
'04': 'Apr',
'05': 'May',
'06': 'Jun',
'07': 'Jul',
'08': 'Aug',
'09': 'Sep',
'10': 'Oct',
'11': 'Nov',
'12': 'Dec'})
```

C:\Users\HP\AppData\Local\Temp\ipykernel_16268\3806760072.py:1: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

df_2015['Month']=df_2015['Order Date'].str[-7:-5].replace({'01': 'Jan',

In [147... df_2015.head(1)

Out[147... Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Count
5	6 2015-115812	CA-09/06/2015	14/06/2015	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States

1 rows × 21 columns



Month wise Sales in 2015

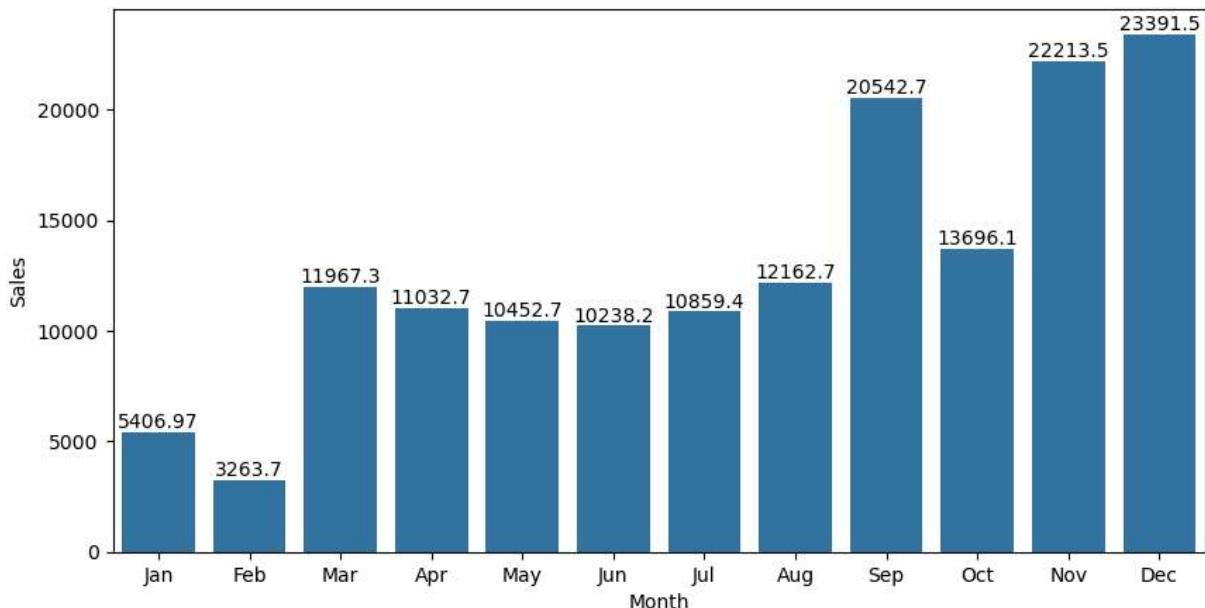
In [148... sales_month_2015 = df_2015.groupby(['Month','Month_no'],as_index = False)[['Sales']].sum()

Out[148...]

	Month	Month_no	Sales
4	Jan	01	5406.97
3	Feb	02	3263.70
7	Mar	03	11967.32
0	Apr	04	11032.71
8	May	05	10452.71
6	Jun	06	10238.16
5	Jul	07	10859.41
1	Aug	08	12162.68
11	Sep	09	20542.67
10	Oct	10	13696.08
9	Nov	11	22213.51
2	Dec	12	23391.48

In [149...]

```
plt.figure(figsize = (10,5))
ax = sns.barplot(data = sales_month_2015 , x = 'Month' , y = 'Sales')
for bars in ax.containers:
    ax.bar_label(bars)
plt.show()
```

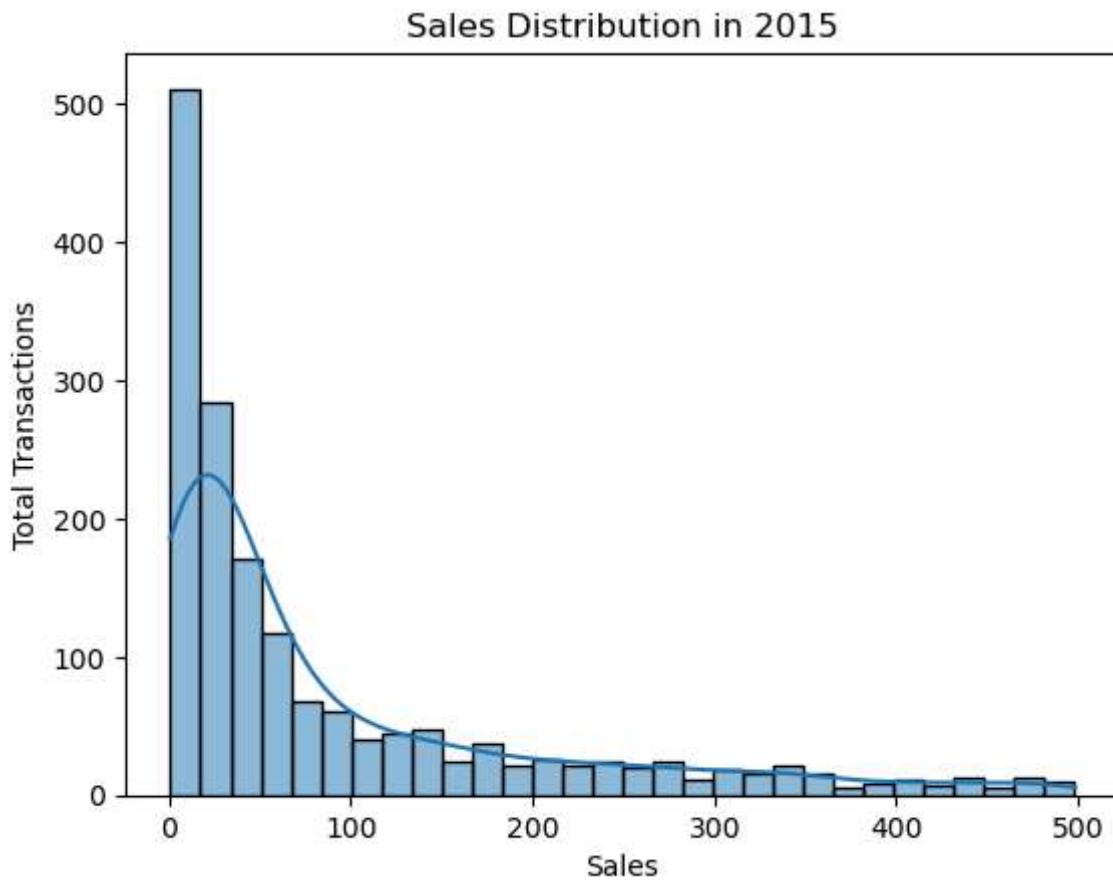


1. Sales Distribution in 2015

In [150...]

```
sns.histplot(df_2015['Sales'],kde = True,bins = 30)
plt.ylabel('Total Transactions')
```

```
plt.title('Sales Distribution in 2015')
plt.show()
```



2.Top 10 Products by Sales in 2015

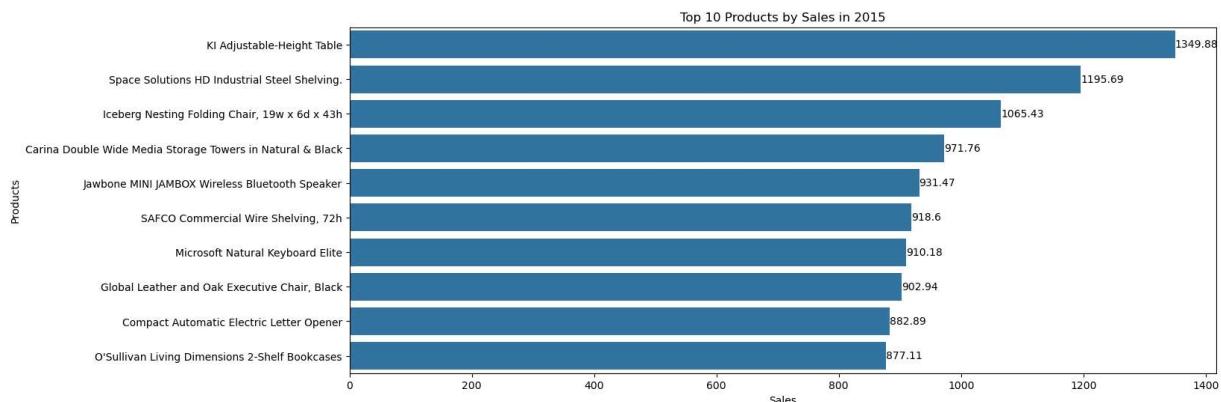
```
In [151]: product_sale_2015 = df_2015.groupby(['Product Name'],as_index = False)['Sales'].sum()
product_sale_2015
```

Out[151...]

	Product Name	Sales
538	KI Adjustable-Height Table	1349.88
849	Space Solutions HD Industrial Steel Shelving.	1195.69
517	Iceberg Nesting Folding Chair, 19w x 6d x 43h	1065.43
246	Carina Double Wide Media Storage Towers in Natural & Black	971.76
536	Jawbone MINI JAMBOX Wireless Bluetooth Speaker	931.47
781	SAFCO Commercial Wire Shelving, 72h	918.60
625	Microsoft Natural Keyboard Elite	910.18
452	Global Leather and Oak Executive Chair, Black	902.94
264	Compact Automatic Electric Letter Opener	882.89
679	O'Sullivan Living Dimensions 2-Shelf Bookcases	877.11

In [152...]

```
plt.figure(figsize =(15,6))
ax = sns.barplot(data = product_sale_2015 , x = 'Sales',y = 'Product Name')
for bars in ax.containers:
    ax.bar_label(bars)
plt.title("Top 10 Products by Sales in 2015")
plt.ylabel("Products")
plt.show()
```



3. Sales by Category in 2015

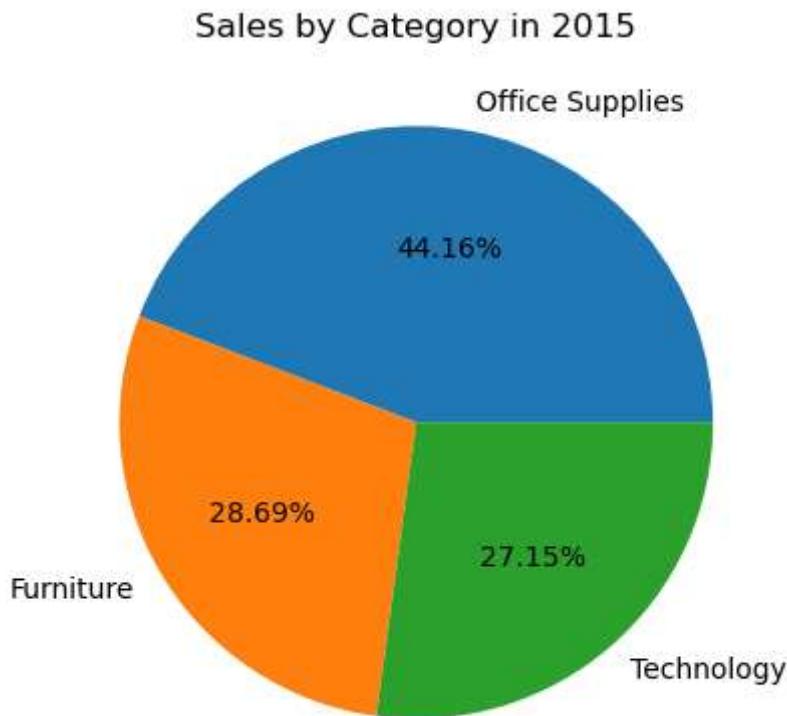
In [153...]

```
category_sale_2015 = df_2015.groupby(['Category'],as_index = False)[['Sales']].sum()
category_sale_2015
```

Out[153...]

	Category	Sales
1	Office Supplies	68549.83
0	Furniture	44532.93
2	Technology	42144.64

```
In [154... plt.pie(category_sale_2015['Sales'],labels = category_sale_2015['Category'],autopct=plt.title ("Sales by Category in 2015") plt.show()
```



4.Sales by Ship Mode in 2015

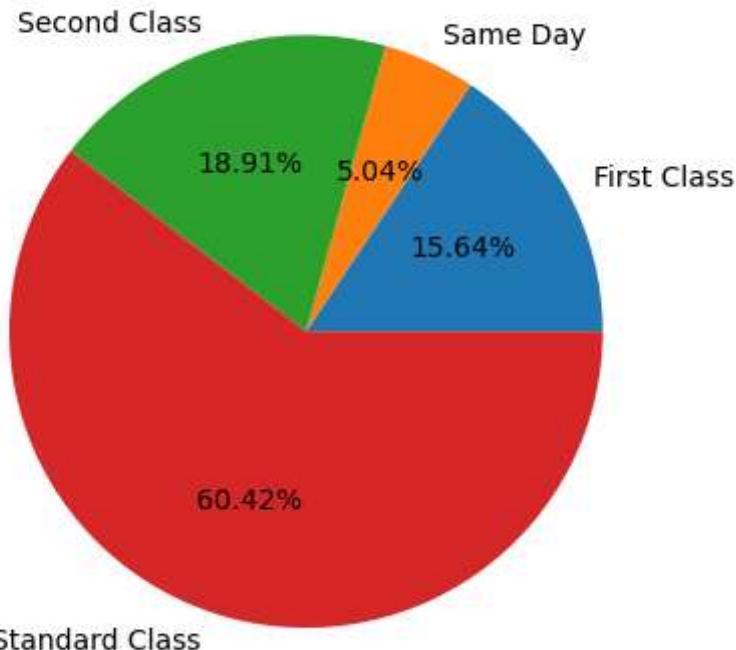
```
In [155... shiping_sales_2015 = df_2015.groupby(['Ship Mode'],as_index = False)['Sales'].sum() shiping_sales_2015
```

Out[155...]

	Ship Mode	Sales
0	First Class	24270.23
1	Same Day	7817.84
2	Second Class	29354.65
3	Standard Class	93784.68

```
In [156... plt.pie(shiping_sales_2015['Sales'],labels = shiping_sales_2015['Ship Mode'],autopct=plt.title('Sales by Ship Mode in 2015') plt.show()
```

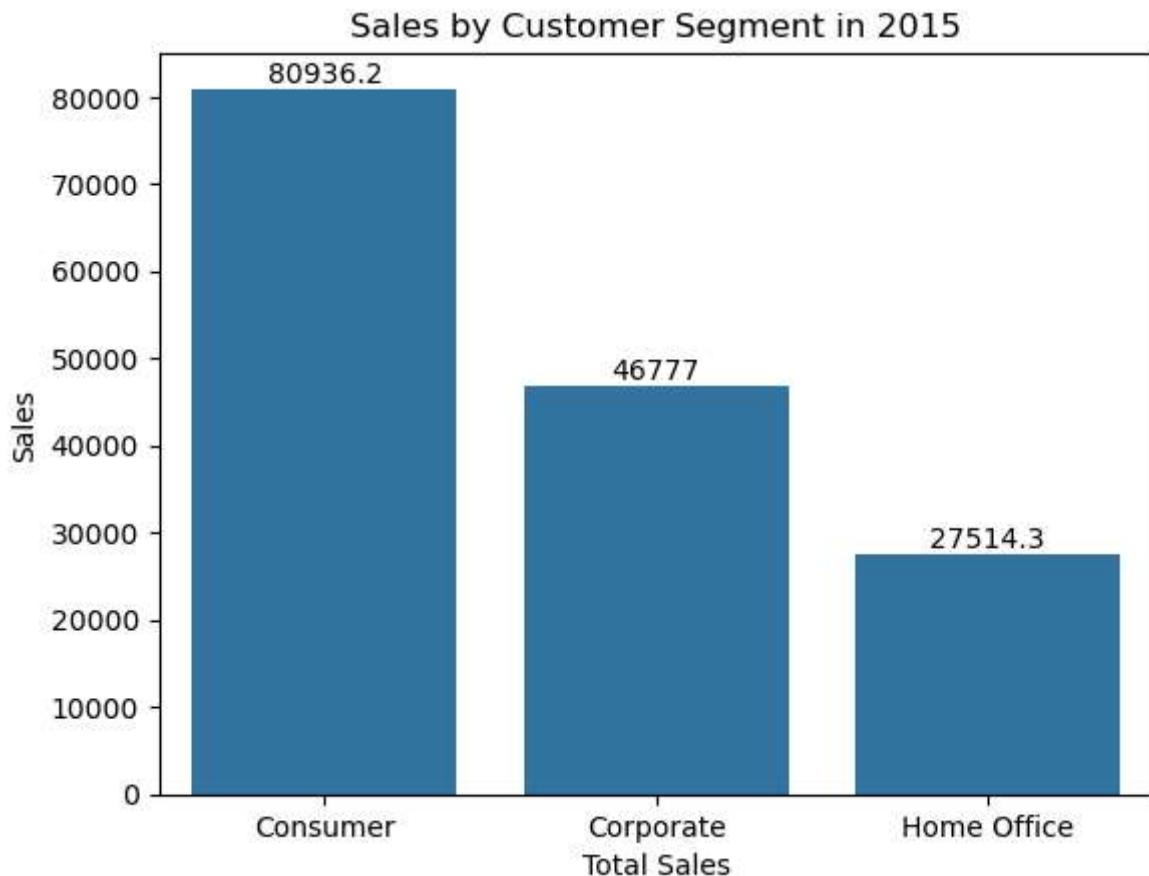
Sales by Ship Mode in 2015



5. Sales by Customer Segment in 2015

In [157...]

```
segment_sales_2015 = df_2015.groupby(['Segment'],as_index = False)['Sales'].sum()
ax = sns.barplot(data = segment_sales_2015, y = 'Sales',x = 'Segment')
for bars in ax.containers:
    ax.bar_label(bars)
plt.title('Sales by Customer Segment in 2015')
plt.xlabel('Total Sales')
plt.show()
```



6. Sales by Region in 2015

```
In [158]: sales_region_2015 = df_2015.groupby(['Region'],as_index = False)['Sales'].sum().sort_values('Sales', ascending=False)
```

```
Out[158]:
```

	Region	Sales
3	West	57336.07
1	East	37601.03
0	Central	35080.48
2	South	25209.82

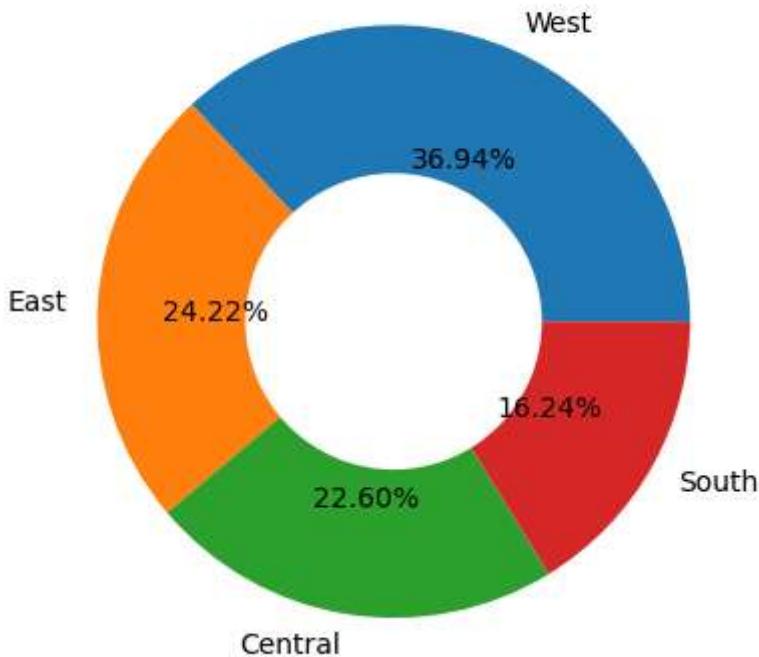
```
In [159]: plt.pie(sales_region_2015['Sales'],labels = sales_region_2015['Region'], autopct = '%.1f%%')
```

```
plt.pie([1],radius = 0.5,colors = "w")
```

```
plt.title ("Sales by Region in 2015")
```

```
plt.show()
```

Sales by Region in 2015



In []:

Analysis - 2016

In [160]:

```
df_2016['Month_no']=df_2016['Order Date'].str[-7:-5]
```

C:\Users\HP\AppData\Local\Temp\ipykernel_16268\570149014.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
df_2016['Month_no']=df_2016['Order Date'].str[-7:-5]
```

In [161]:

```
df_2016['Month']=df_2016['Order Date'].str[-7:-5].replace({'01': 'Jan',  
'02': 'Feb',  
'03': 'Mar',  
'04': 'Apr',  
'05': 'May',  
'06': 'Jun',  
'07': 'Jul',  
'08': 'Aug',  
'09': 'Sep',  
'10': 'Oct',  
'11': 'Nov',  
'12': 'Dec'})
```

```
C:\Users\HP\AppData\Local\Temp\ipykernel_16268\2521323480.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

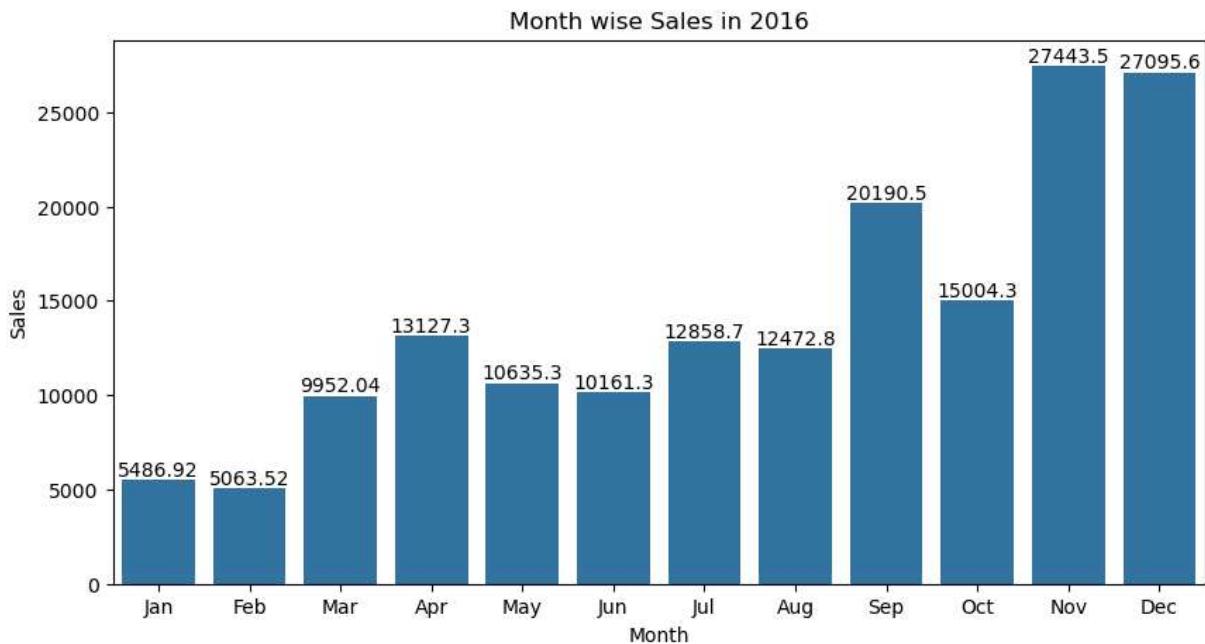
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
df_2016['Month']=df_2016['Order Date'].str[-7:-5].replace({'01': 'Jan',
```

In [162...]: sales_month_2016 = df_2016.groupby(['Month','Month_no'],as_index = False)[['Sales']].sum()

Out[162...]:

	Month	Month_no	Sales
4	Jan	01	5486.92
3	Feb	02	5063.52
7	Mar	03	9952.04
0	Apr	04	13127.28
8	May	05	10635.29
6	Jun	06	10161.30
5	Jul	07	12858.67
1	Aug	08	12472.80
11	Sep	09	20190.55
10	Oct	10	15004.27
9	Nov	11	27443.48
2	Dec	12	27095.62

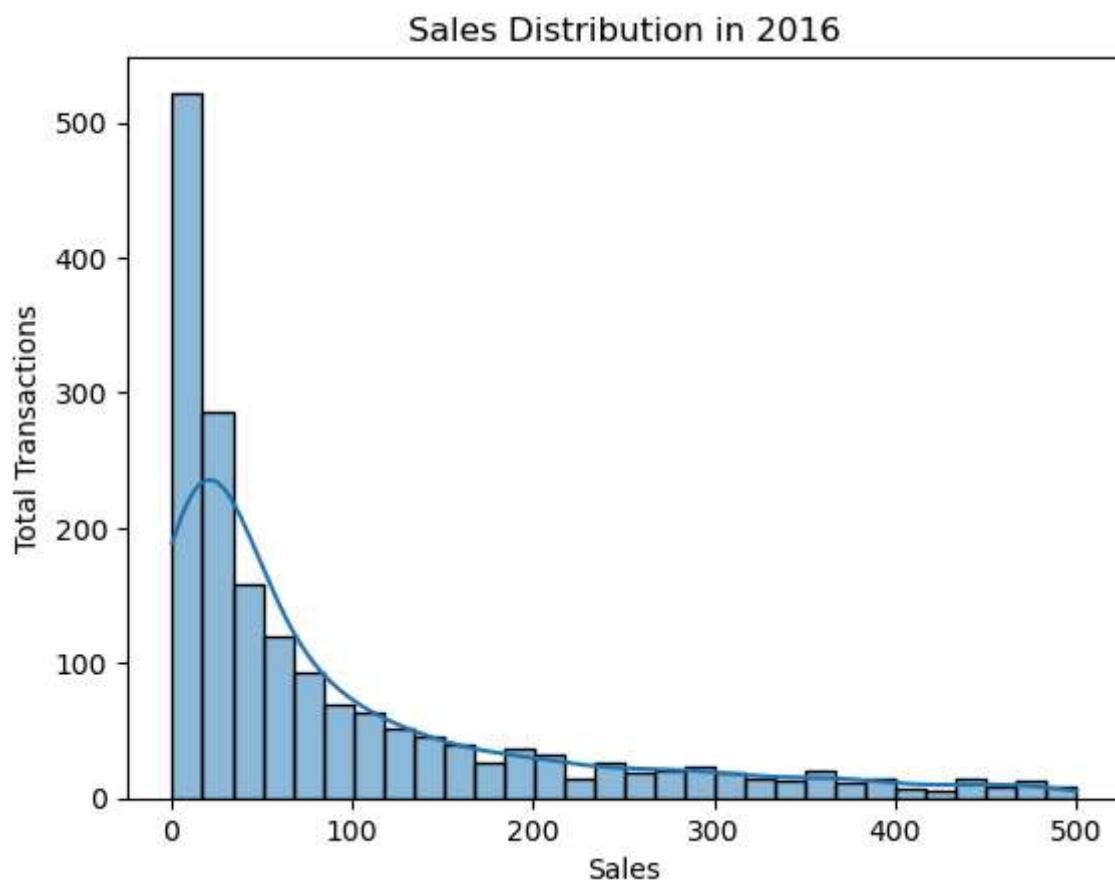
In [163...]: plt.figure(figsize = (10,5))
ax = sns.barplot(data = sales_month_2016 , x = 'Month' , y = 'Sales')
for bars in ax.containers:
 ax.bar_label(bars)
plt.title('Month wise Sales in 2016')
plt.show()



1. Sales Distribution in 2016

In [164]:

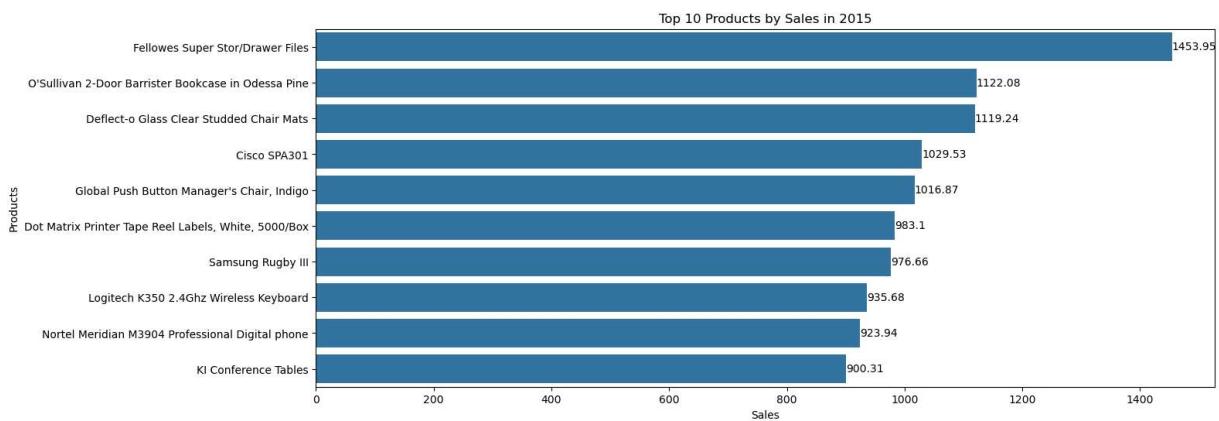
```
sns.histplot(df_2016['Sales'], kde = True, bins = 30)
plt.ylabel('Total Transactions')
plt.title('Sales Distribution in 2016')
plt.show()
```



2.Top 10 Products by Sales in 2016

In [165...]

```
product_sale_2016 = df_2016.groupby(['Product Name'],as_index = False)['Sales'].sum()
plt.figure(figsize =(15,6))
ax = sns.barplot(data = product_sale_2016 , x = 'Sales',y = 'Product Name')
for bars in ax.containers:
    ax.bar_label(bars)
plt.title("Top 10 Products by Sales in 2015")
plt.ylabel("Products")
plt.show()
```

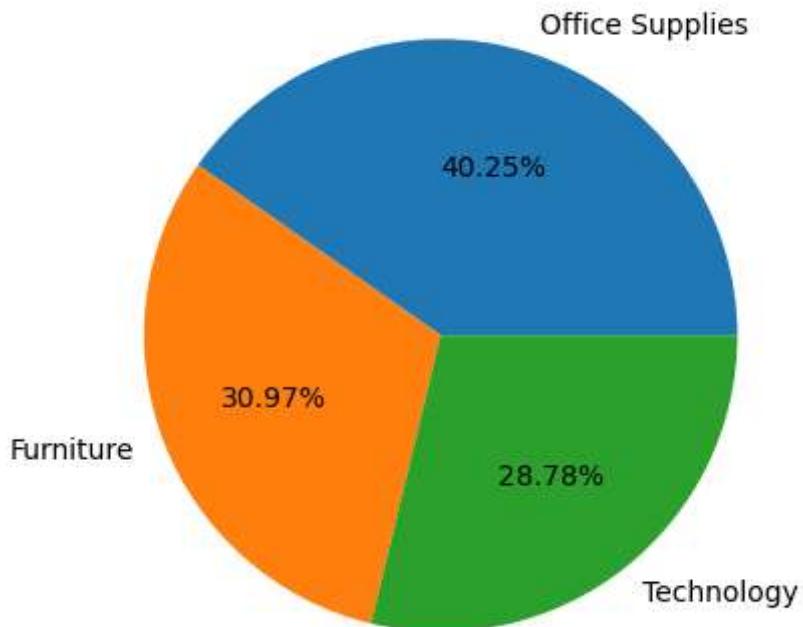


3. Sales by Category in 2016

In [166...]

```
category_sale_2016 = df_2016.groupby(['Category'],as_index = False)['Sales'].sum()
plt.pie(category_sale_2016['Sales'],labels = category_sale_2016['Category'], autopct = '%.1f%%')
plt.title ("Sales by Category in 2016")
plt.show()
```

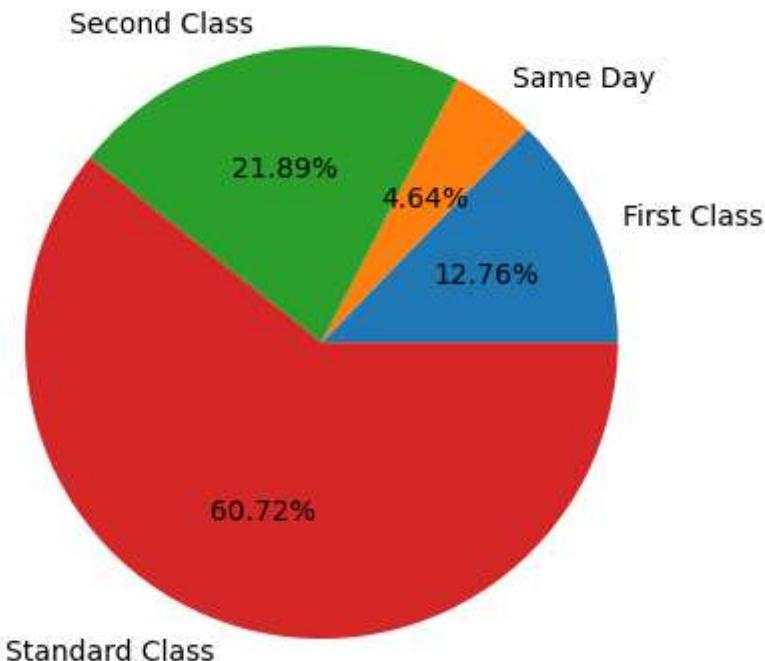
Sales by Category in 2016



4.Sales by Ship Mode in 2016

```
In [167]: shiping_sales_2016 = df_2016.groupby(['Ship Mode'],as_index = False)['Sales'].sum()
plt.pie(shiping_sales_2016['Sales'],labels = shiping_sales_2016['Ship Mode'],autopct = '%.2f%%')
plt.title('Sales by Ship Mode in 2016')
plt.show()
```

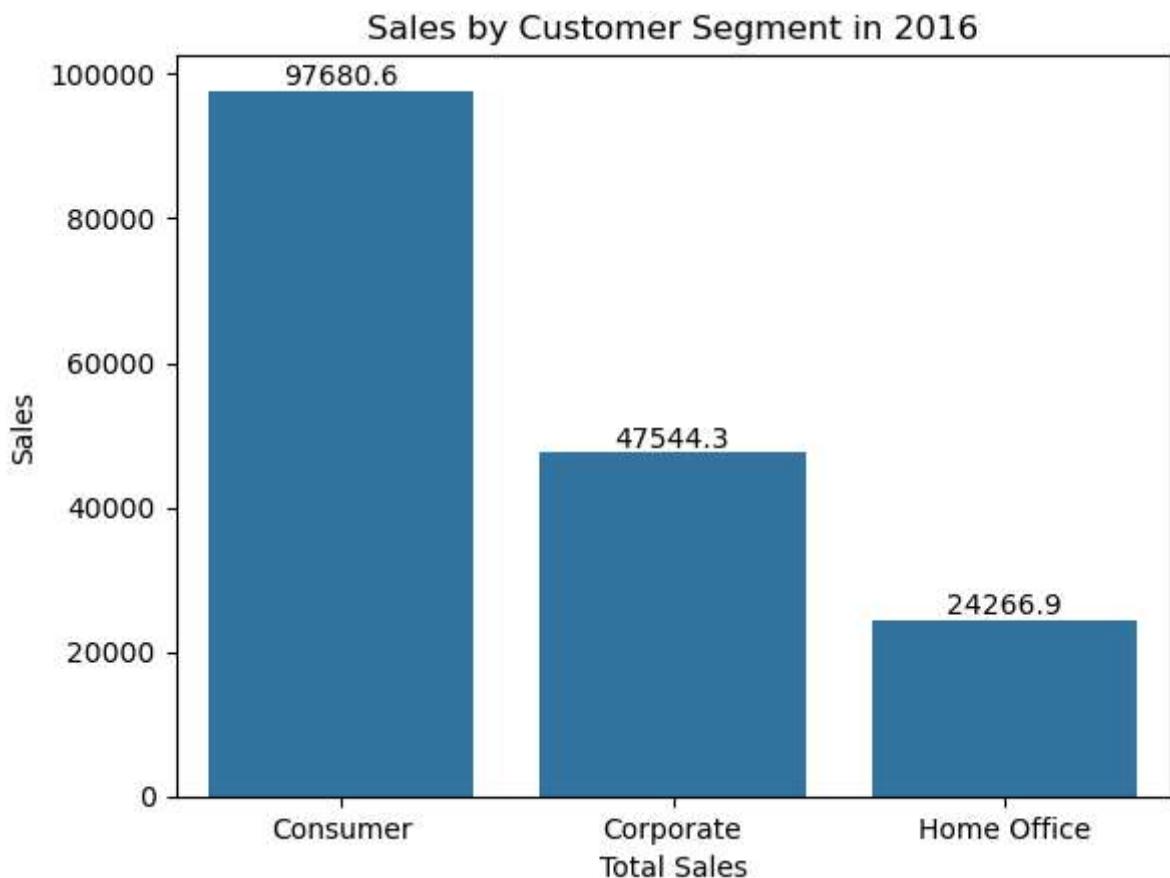
Sales by Ship Mode in 2016



5. Sales by Customer Segment in 2016

In [168...]

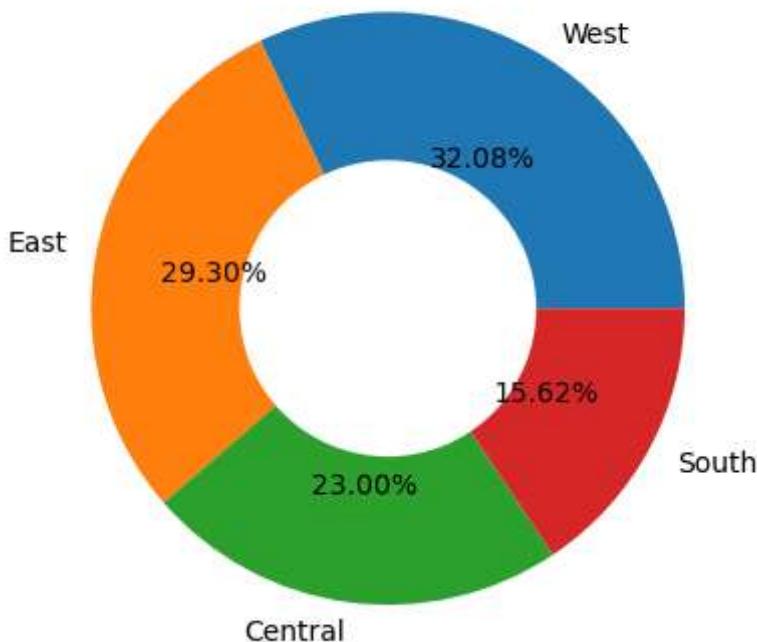
```
segment_sales_2016 = df_2016.groupby(['Segment'],as_index = False)['Sales'].sum()
ax = sns.barplot(data = segment_sales_2016, y = 'Sales',x = 'Segment')
for bars in ax.containers:
    ax.bar_label(bars)
plt.title('Sales by Customer Segment in 2016')
plt.xlabel('Total Sales')
plt.show()
```



6. Sales by Region in 2016

```
In [169]: sales_region_2016 = df_2016.groupby(['Region'], as_index = False)[['Sales']].sum().sort_values('Sales', ascending = False)
plt.pie(sales_region_2016['Sales'], labels = sales_region_2016['Region'], autopct = '%.1f%%')
plt.pie([1], radius = 0.5, colors = "w")
plt.title ("Sales by Region in 2016")
plt.show()
```

Sales by Region in 2016



In []:

Analysis - 2017

In [170...]

```
df_2017['Month_no']=df_2017['Order Date'].str[-7:-5]
df_2017['Month']=df_2017['Order Date'].str[-7:-5].replace({'01': 'Jan',
'02': 'Feb',
'03': 'Mar',
'04': 'Apr',
'05': 'May',
'06': 'Jun',
'07': 'Jul',
'08': 'Aug',
'09': 'Sep',
'10': 'Oct',
'11': 'Nov',
'12': 'Dec'})
sales_month_2017 = df_2017.groupby(['Month','Month_no'],as_index = False)[['Sales']].sum()
plt.figure(figsize = (10,5))
ax = sns.barplot(data = sales_month_2017 , x = 'Month', y = 'Sales')
for bars in ax.containers:
    ax.bar_label(bars)
plt.title('Month wise Sales in 2017')
plt.show()
```

```
C:\Users\HP\AppData\Local\Temp\ipykernel_16268\1190434489.py:1: SettingWithCopyWarning:
```

A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
df_2017['Month_no']=df_2017['Order Date'].str[-7:-5]
```

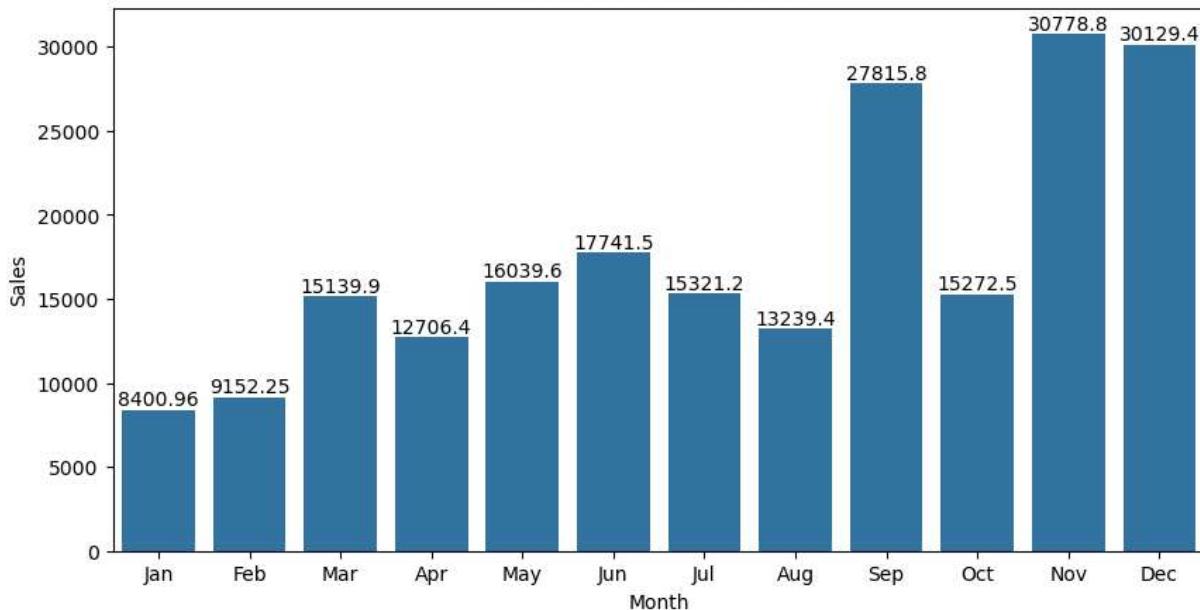
```
C:\Users\HP\AppData\Local\Temp\ipykernel_16268\1190434489.py:2: SettingWithCopyWarning:
```

A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
df_2017['Month']=df_2017['Order Date'].str[-7:-5].replace({'01': 'Jan',
```

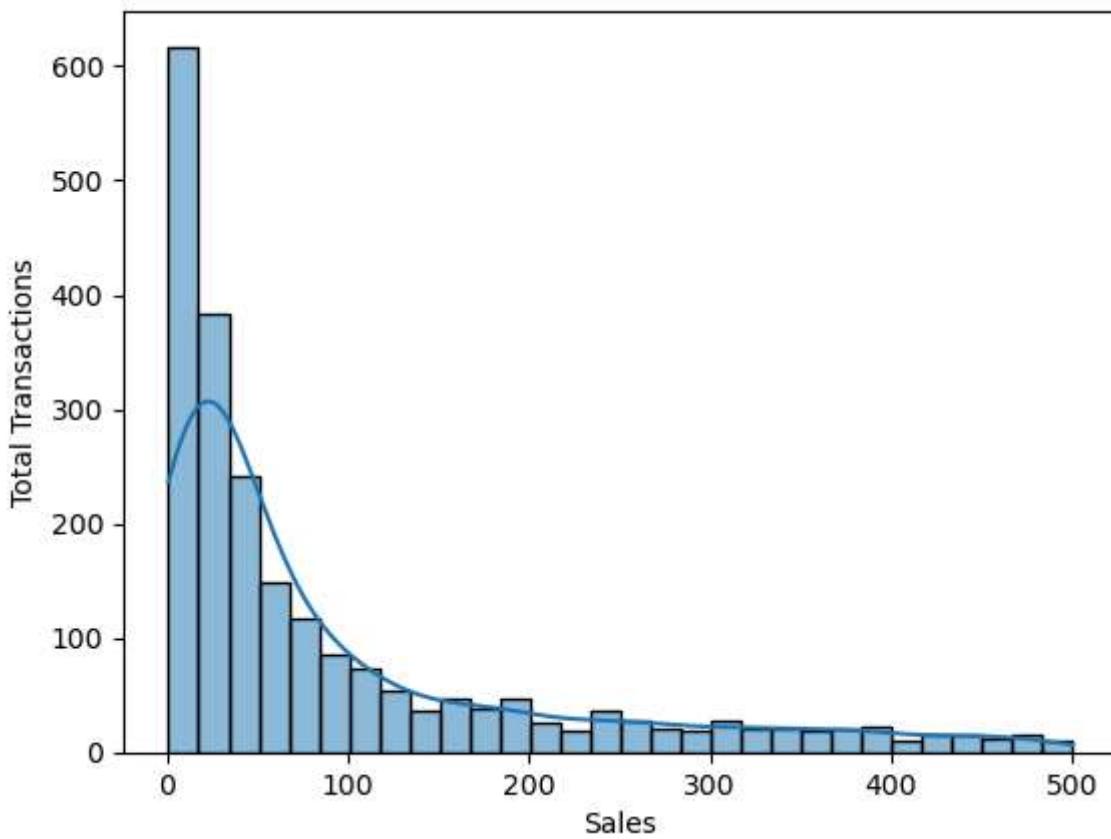
Month wise Sales in 2017



1. Sales Distribution in 2017

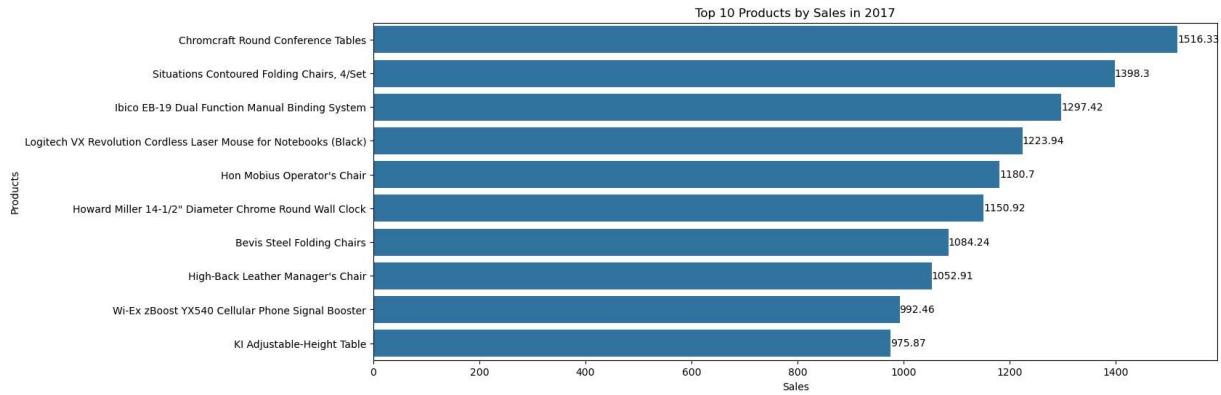
```
In [171]: sns.histplot(df_2017['Sales'],kde = True,bins = 30)
plt.ylabel('Total Transactions')
plt.title('Sales Distribution in 2017')
plt.show()
```

Sales Distribution in 2017



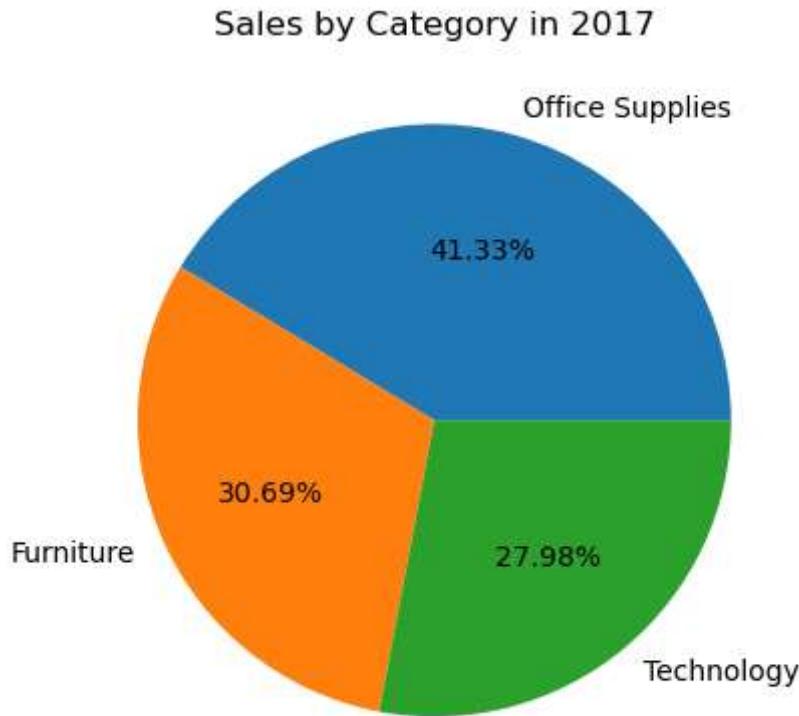
2. Top 10 Products by Sales in 2017

```
In [172]: product_sale_2017 = df_2017.groupby(['Product Name'],as_index = False)['Sales'].sum()
plt.figure(figsize =(15,6))
ax = sns.barplot(data = product_sale_2017 , x = 'Sales',y = 'Product Name')
for bars in ax.containers:
    ax.bar_label(bars)
plt.title("Top 10 Products by Sales in 2017")
plt.ylabel("Products")
plt.show()
```



3. Sales by Category in 2017

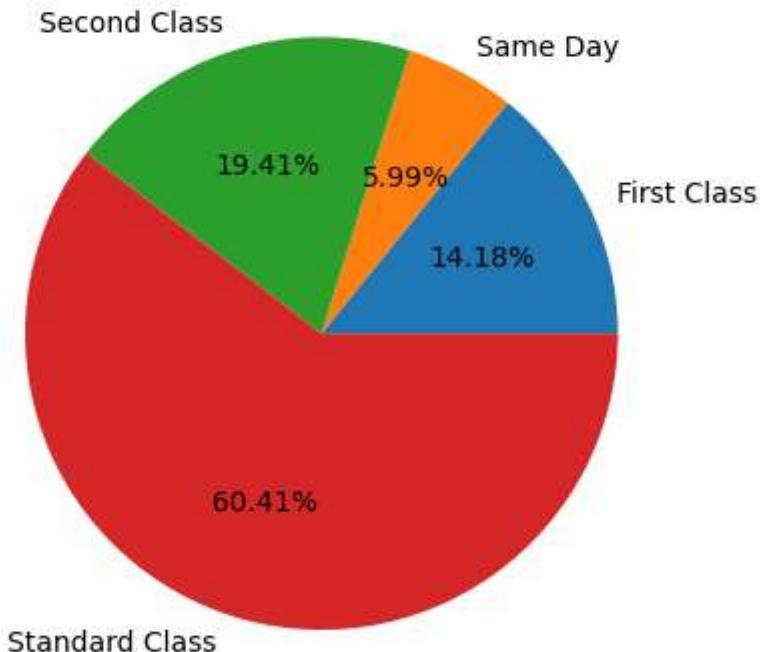
```
In [173...]: category_sale_2017 = df_2017.groupby(['Category'],as_index = False)['Sales'].sum()
plt.pie(category_sale_2017['Sales'],labels = category_sale_2017['Category'], autopct = '%.2f%%')
plt.title ("Sales by Category in 2017")
plt.show()
```



4.Sales by Ship Mode in 2017

```
In [174...]: shiping_sales_2017 = df_2017.groupby(['Ship Mode'],as_index = False)['Sales'].sum()
plt.pie(shiping_sales_2017['Sales'],labels = shiping_sales_2017['Ship Mode'], autopct = '%.2f%%')
plt.title('Sales by Ship Mode in 2017')
plt.show()
```

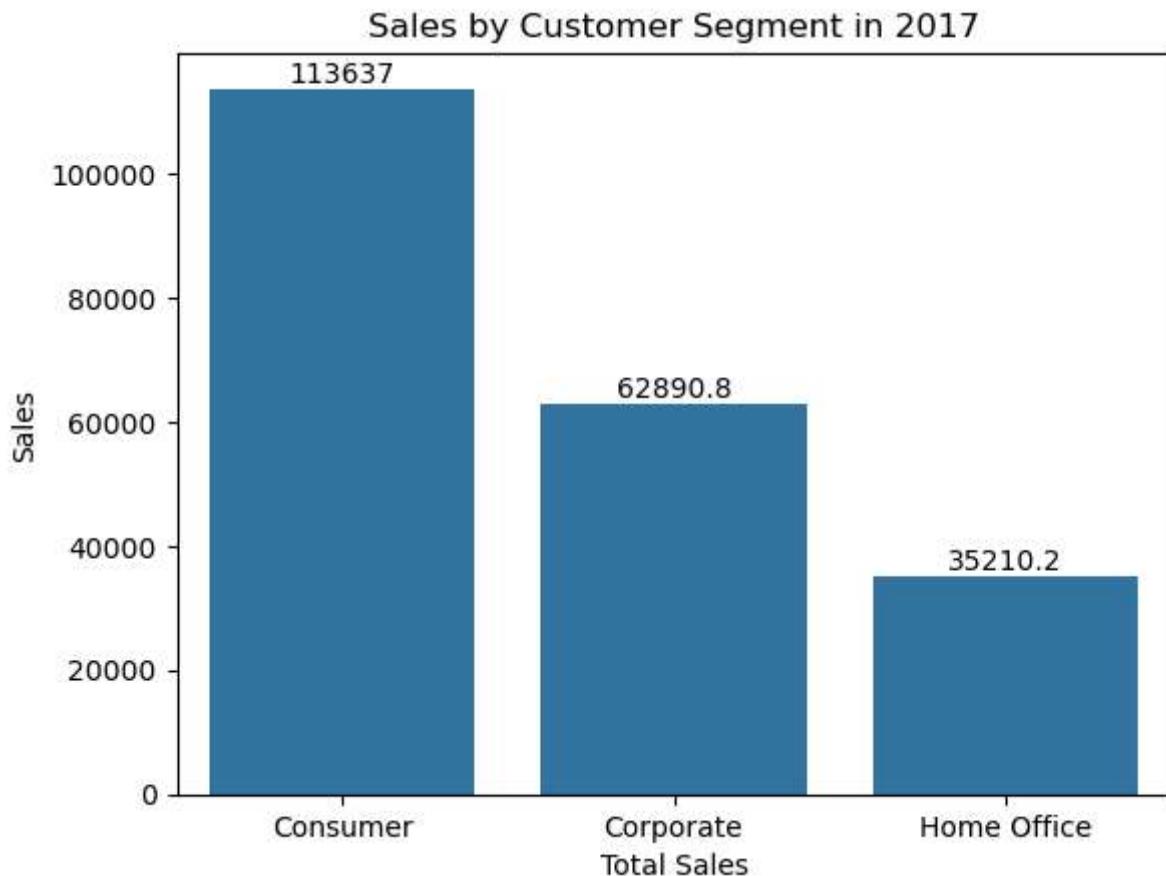
Sales by Ship Mode in 2017



5. Sales by Customer Segment in 2017

In [175...]

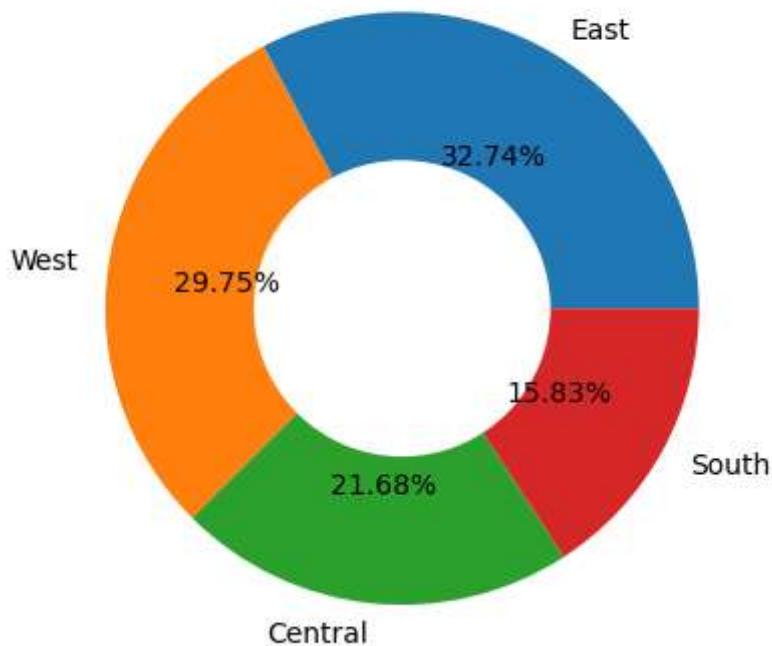
```
segment_sales_2017 = df_2017.groupby(['Segment'],as_index = False)['Sales'].sum()
ax = sns.barplot(data = segment_sales_2017, y = 'Sales',x = 'Segment')
for bars in ax.containers:
    ax.bar_label(bars)
plt.title('Sales by Customer Segment in 2017')
plt.xlabel('Total Sales')
plt.show()
```



6. Sales by Region in 2017

```
In [176]: sales_region_2017 = df_2017.groupby(['Region'], as_index = False)[['Sales']].sum().sort_values('Sales', ascending = False)
plt.pie(sales_region_2017['Sales'], labels = sales_region_2017['Region'], autopct = '%.1f%%')
plt.pie([1], radius = 0.5, colors = "w")
plt.title ("Sales by Region in 2017")
plt.show()
```

Sales by Region in 2017



In []:

Analysis - 2018

In [177...]

```
df_2018['Month_no']=df_2018['Order Date'].str[-7:-5]
df_2018['Month']=df_2018['Order Date'].str[-7:-5].replace({'01': 'Jan',
'02': 'Feb',
'03': 'Mar',
'04': 'Apr',
'05': 'May',
'06': 'Jun',
'07': 'Jul',
'08': 'Aug',
'09': 'Sep',
'10': 'Oct',
'11': 'Nov',
'12': 'Dec'})
sales_month_2018 = df_2018.groupby(['Month','Month_no'],as_index = False)[['Sales']].sum()
plt.figure(figsize = (10,5))
ax = sns.barplot(data = sales_month_2018 , x = 'Month', y = 'Sales')
for bars in ax.containers:
    ax.bar_label(bars)
plt.show()
```

```
C:\Users\HP\AppData\Local\Temp\ipykernel_16268\1107112528.py:1: SettingWithCopyWarning:
```

A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

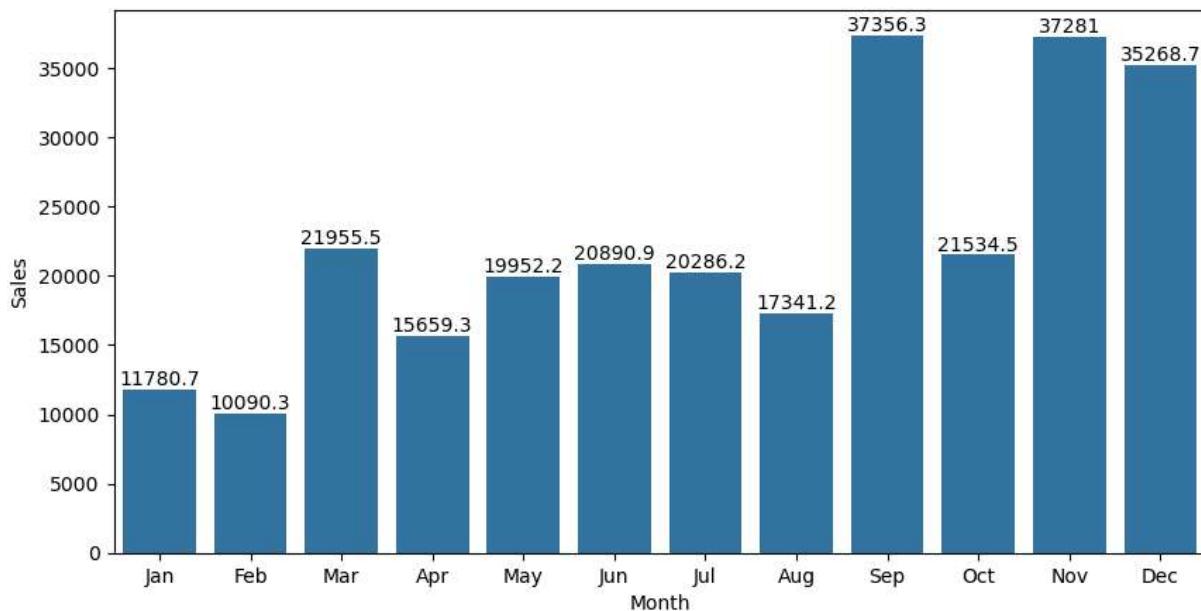
```
df_2018['Month_no']=df_2018['Order Date'].str[-7:-5]
```

```
C:\Users\HP\AppData\Local\Temp\ipykernel_16268\1107112528.py:2: SettingWithCopyWarning:
```

A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

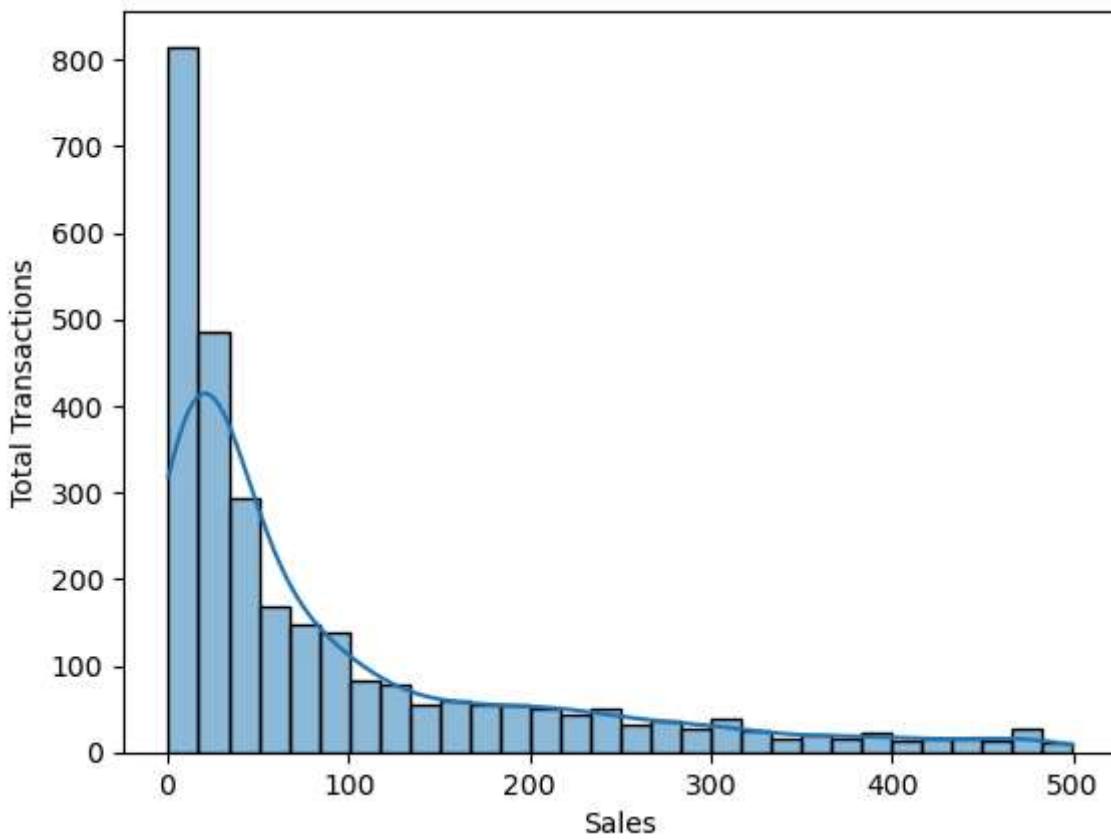
```
df_2018['Month']=df_2018['Order Date'].str[-7:-5].replace({'01': 'Jan',
```



1. Sales Distribution in 2018

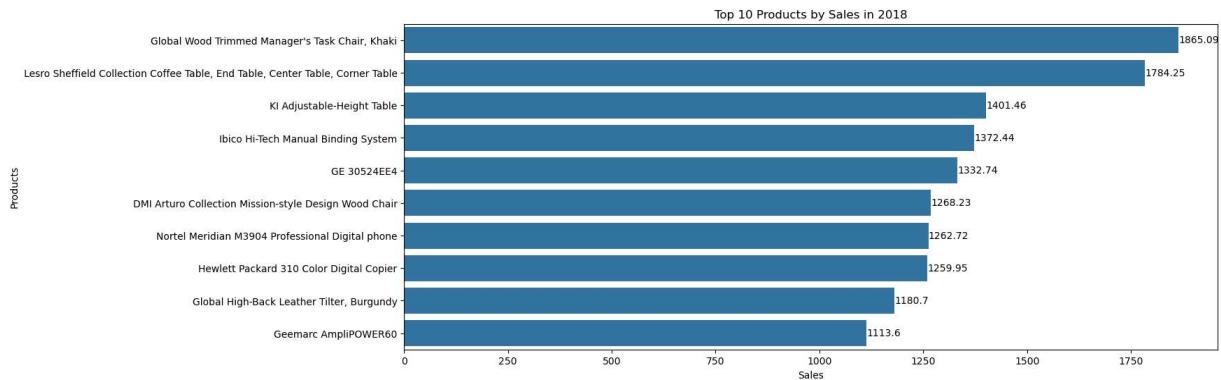
```
In [178...]: sns.histplot(df_2018['Sales'], kde = True, bins = 30)
plt.ylabel('Total Transactions')
plt.title('Sales Distribution in 2018')
plt.show()
```

Sales Distribution in 2018



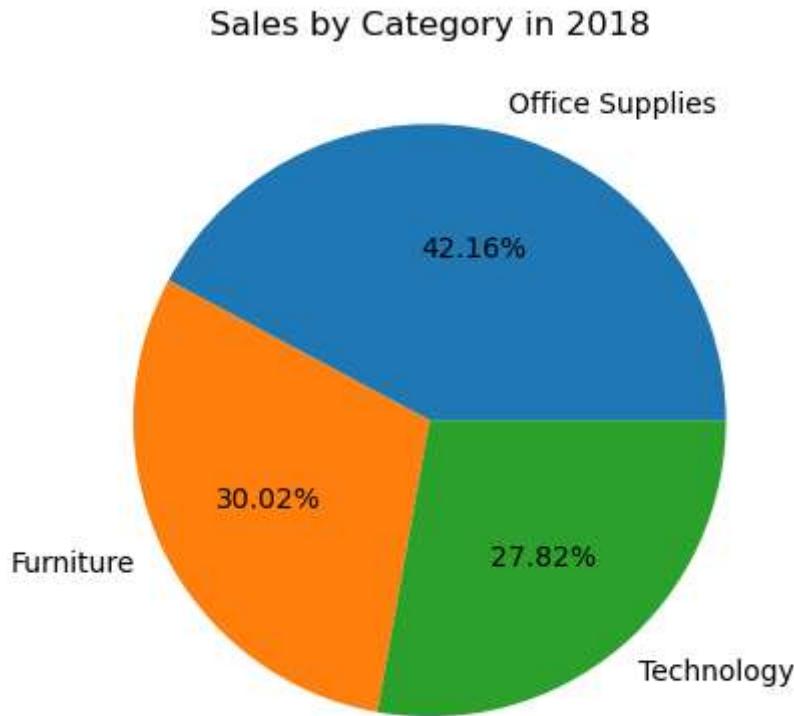
2. Top 10 Products by Sales in 2018

```
In [179]: product_sale_2018 = df_2018.groupby(['Product Name'],as_index = False)['Sales'].sum()
plt.figure(figsize =(15,6))
ax = sns.barplot(data = product_sale_2018 , x = 'Sales',y = 'Product Name')
for bars in ax.containers:
    ax.bar_label(bars)
plt.title("Top 10 Products by Sales in 2018")
plt.ylabel("Products")
plt.show()
```



3. Sales by Category in 2018

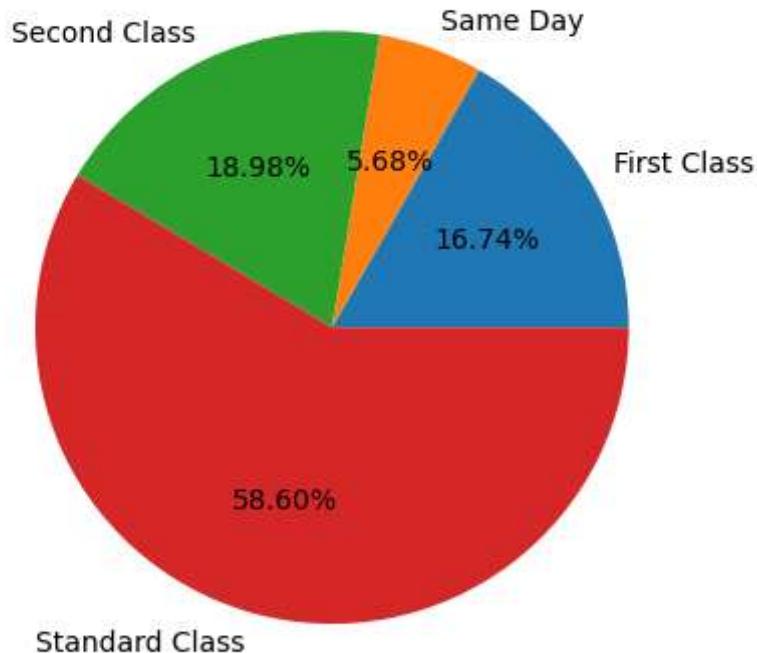
```
In [180...]: category_sale_2018 = df_2018.groupby(['Category'],as_index = False)['Sales'].sum()
plt.pie(category_sale_2018['Sales'],labels = category_sale_2018['Category'],autopct = '%.2f%%')
plt.title ("Sales by Category in 2018")
plt.show()
```



4.Sales by Ship Mode in 2018

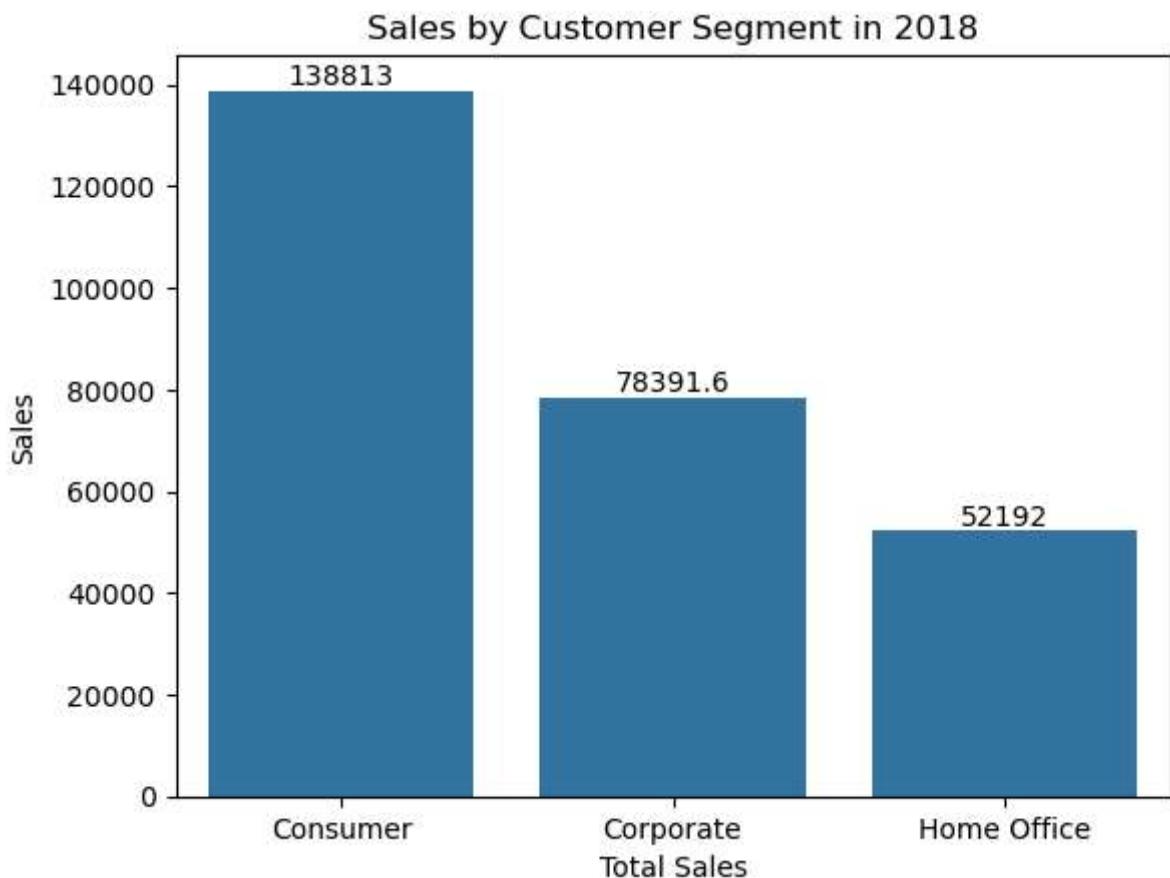
```
In [181...]: shiping_sales_2018 = df_2018.groupby(['Ship Mode'],as_index = False)['Sales'].sum()
plt.pie(shiping_sales_2018['Sales'],labels = shiping_sales_2018['Ship Mode'],autopct = '%.2f%%')
plt.title('Sales by Ship Mode in 2018')
plt.show()
```

Sales by Ship Mode in 2018



5. Sales by Customer Segment in 2018

```
In [182...]:  
segment_sales_2018 = df_2018.groupby(['Segment'], as_index = False)[['Sales']].sum()  
ax = sns.barplot(data = segment_sales_2018, y = 'Sales', x = 'Segment')  
for bars in ax.containers:  
    ax.bar_label(bars)  
plt.title('Sales by Customer Segment in 2018')  
plt.xlabel('Total Sales')  
plt.show()
```



6. Sales by Region in 2018

```
In [183]: sales_region_2018 = df_2018.groupby(['Region'], as_index = False)[['Sales']].sum().sort_values(['Sales'], ascending = False)
plt.pie(sales_region_2018['Sales'], labels = sales_region_2018['Region'], autopct = '%.1f%%')
plt.pie([1], radius = 0.5, colors = "w")
plt.title ("Sales by Region in 2018")
plt.show()
```

Sales by Region in 2018

