Abhishek Gupta

Intern : Bharat Intern 10 July 2023

Object :- Forecasting the sales of a supermarket

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

In [2]:
```
pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)
```

In [146]:
```
data = pd.read_csv('train.csv')
data.head(3)
```

Out[146]:

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	State	Postal Code	Region	Product ID	Categ
0	1	CA-2017-152156	08/11/2017	11/11/2017	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky	42420.0	South	FUR-BO-10001798	Furn
1	2	CA-2017-152156	08/11/2017	11/11/2017	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky	42420.0	South	FUR-CH-10000454	Furn
2	3	CA-2017-138688	12/06/2017	16/06/2017	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	California	90036.0	West	OFF-LA-10000240	O Sup

In [147]:
```
data.drop('Row ID',axis=1,inplace=True)
```

In [152]:
```
data.shape # The data set has 9800 rows and 17 columns
```

Out[152]:
```
(9800, 17)
```

In [153]:
```
data.info() #The data structure contains object, float64
```

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

In [10]:
```
data.isna().sum() # In the given dataset only Postal Code column contain null values.
```

Out[10]:
```
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 [409]:
```
cat_cols = data.dtypes=='object'
cat_cols = list[cat_cols[cat_cols].index]
cat_cols
```

Out[409]:
```
list[Index(['Order ID', 'Ship Mode', 'Customer ID', 'Customer Name', 'Segment',
       'Country', 'City', 'State', 'Region', 'Product ID', 'Category',
       'Sub-Category', 'Product Name'],
      dtype='object')]
```

In [410]:
```
cat_cols = data.dtypes!='object'
cat_cols = list[cat_cols[cat_cols].index]
cat_cols
```

Out[410]:
```
list[Index(['Order Date', 'Ship Date', 'Postal Code', 'Sales', 'Order_Year'], dtype='object')]
```

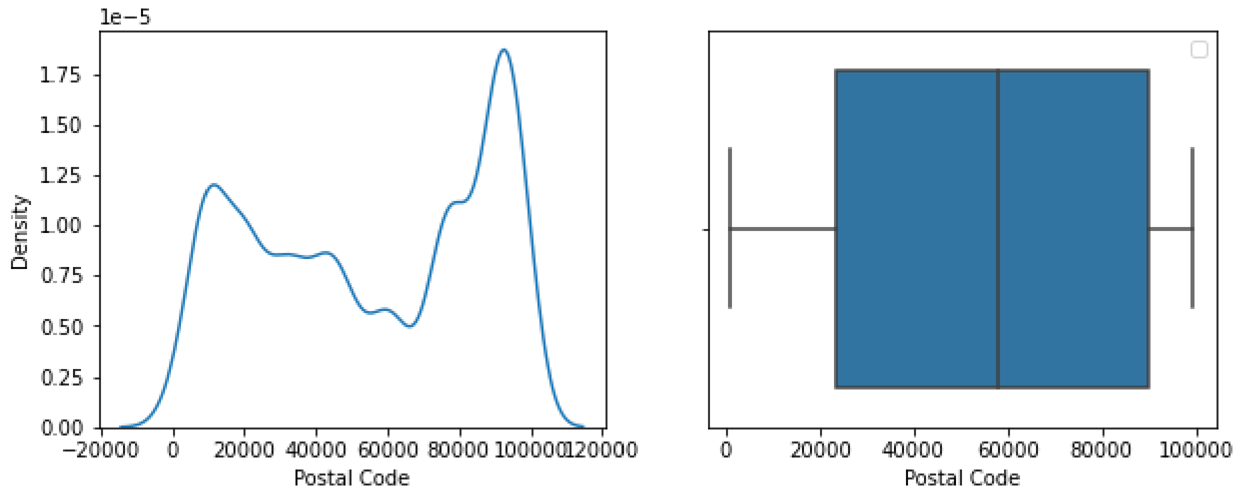
In [154]:
```
data.describe() # As we talk for postal code the mean is less than median there are high chance that,
                # there will be no outliers in this column.while in sales mean has higher value than median so outlier
                # be surely present.
```

Out[154]:

	Postal Code	Sales
count	9789.000000	9800.000000
mean	55273.322403	230.769059
std	32041.223413	626.651875
min	1040.000000	0.444000
25%	23223.000000	17.248000
50%	58103.000000	54.490000
75%	90008.000000	210.605000
max	99301.000000	22638.480000

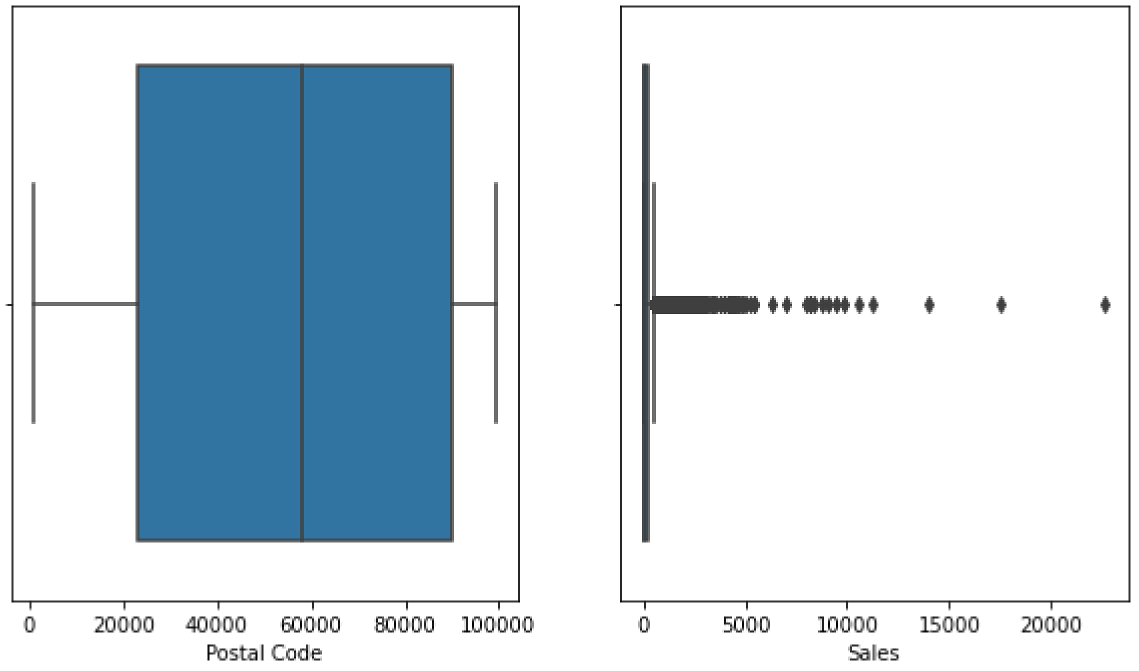
In [381]:
```
plt.figure(figsize=(10,8))

plt.subplot(2,2,1)
sns.kdeplot(data['Postal Code'])

plt.subplot(2,2,2)
sns.boxplot(data['Postal Code'])

plt.legend()
plt.show()
```

```
C:\Users\Dell\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a
keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments wit
hout an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
No artists with labels found to put in legend.  Note that artists whose label start with an underscore are ignored wh
en legend() is called with no argument.
```



```
In [382]: plt.figure(figsize=(10,12))

          plt.subplot(2,2,1)
          sns.boxplot(data['Postal Code'],orient='v')

          plt.subplot(2,2,2)
          sns.boxplot(data['Sales'])
          plt.show()
```

```
C:\Users\Dell\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a
keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments wit
hout an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
C:\Users\Dell\anaconda3\lib\site-packages\seaborn\_core.py:1326: UserWarning: Vertical orientation ignored with only
`x` specified.
  warnings.warn(single_var_warning.format("Vertical", "x"))
C:\Users\Dell\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a
keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments wit
hout an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



In [185]:
```
data.describe(include='object')
```

Out[185]:

	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	State	Region	Product ID	Category	Sub-Category
count	9800	9800	9800	9800	9800	9800	9800	9800	9800	9800	9800	9800	9800	9800
unique	4922	1230	1326	4	793	793	3	1	529	49	4	1861	3	17
top	CA-2018-100111	05/09/2017	26/09/2018	Standard Class	WB-21850	William Brown	Consumer	United States	New York City	California	West	OFF-PA-10001970	Office Supplies	Binders
freq	14	38	34	5859	35	35	5101	9800	891	1946	3140	19	5909	1492

In [356]:
```
data['Segment'].value_counts()
```

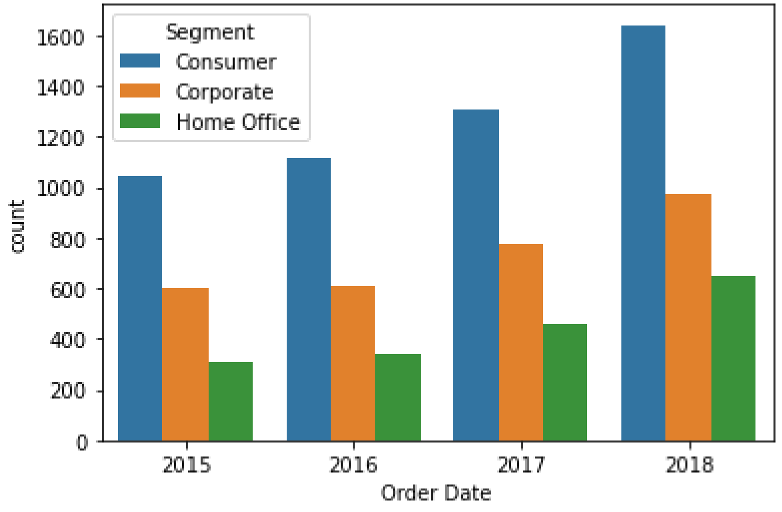
Out[356]:
```
Consumer       5101
Corporate      2953
Home Office    1746
Name: Segment, dtype: int64
```

In [398]:
```
data.groupby(data['Order Date'].dt.year)['Segment'].value_counts()
```

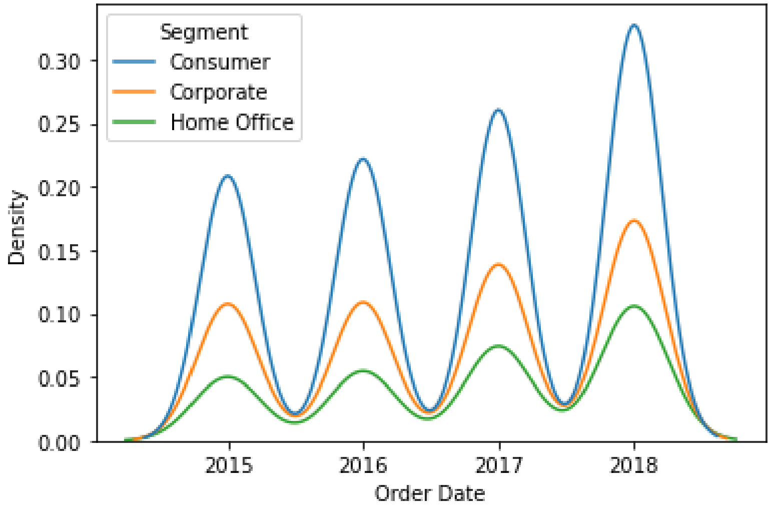
Out[398]:
```
Order Date  Segment
2015        Consumer       1045
            Corporate       601
            Home Office     307
2016        Consumer       1112
            Corporate       608
            Home Office     335
2017        Consumer       1304
            Corporate       775
            Home Office     455
2018        Consumer       1640
            Corporate       969
            Home Office     649
Name: Segment, dtype: int64
```

```
In [365]: sns.countplot(data['Order Date'].dt.year,hue=data['Segment'])
```

```
Out[365]: <AxesSubplot:xlabel='Order Date', ylabel='count'>
```

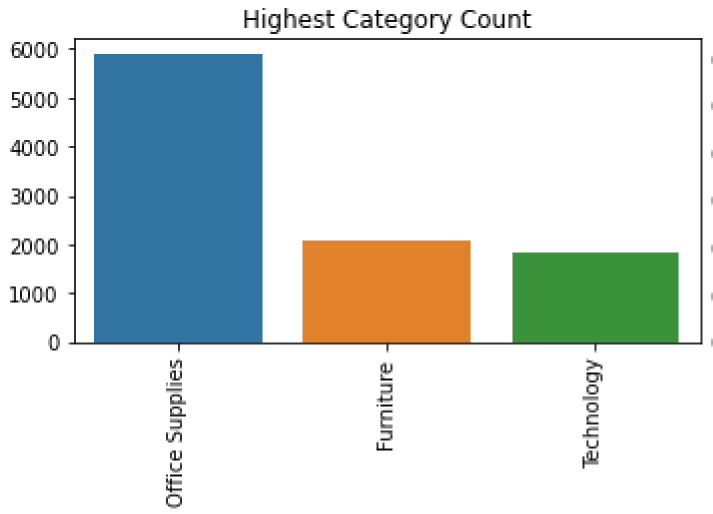


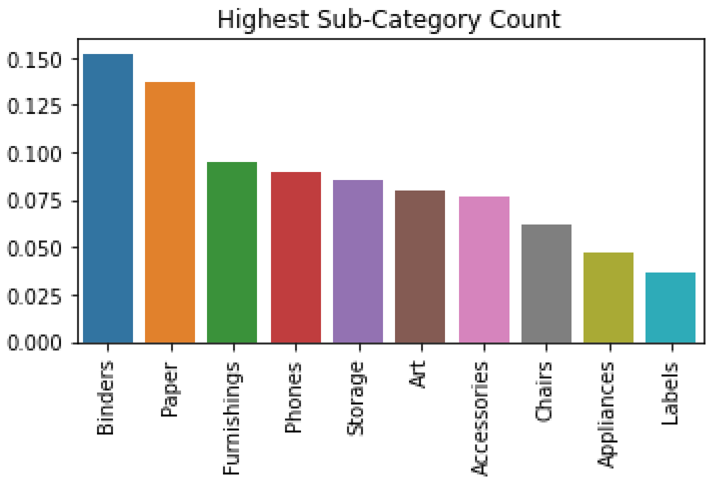
```
In [373]: sns.kdeplot(data['Order Date'].dt.year,hue=data['Segment'])   # ese kya predit kar sakte hain
          plt.show()
```

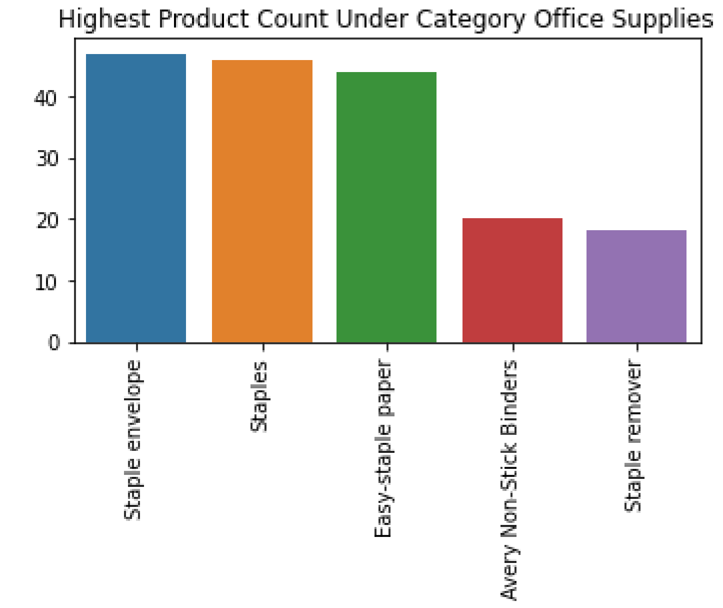


In [399]:
```
plt.figure(figsize=(10,8))
plt.subplot(2,2,1)
plt.title('Highest Category Count')
sns.barplot(data['Category'].value_counts().head().index, data['Category'].value_counts().head().values)
plt.xticks(rotation=90)

plt.subplot(2,2,2)
plt.title('Highest Sub-Category Count')
sns.barplot(data['Sub-Category'].value_counts().head(10).index, data['Sub-Category'].value_counts(5).head(10).values)
plt.xticks(rotation=90)

plt.subplot(2,2,3)
plt.title('Highest Product Count Under Category Office Supplies')
sns.barplot(data[data['Category']=='Office Supplies']["Product Name"].value_counts().head().index, data[data['Category
plt.xticks(rotation=90)

plt.tight_layout()
plt.show()
```

```
C:\Users\Dell\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as k
eyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
C:\Users\Dell\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as k
eyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
C:\Users\Dell\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as k
eyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```


Highest Category Count
6000
5000
4000
3000
2000
1000
0
Office Supplies
Furniture
Technology

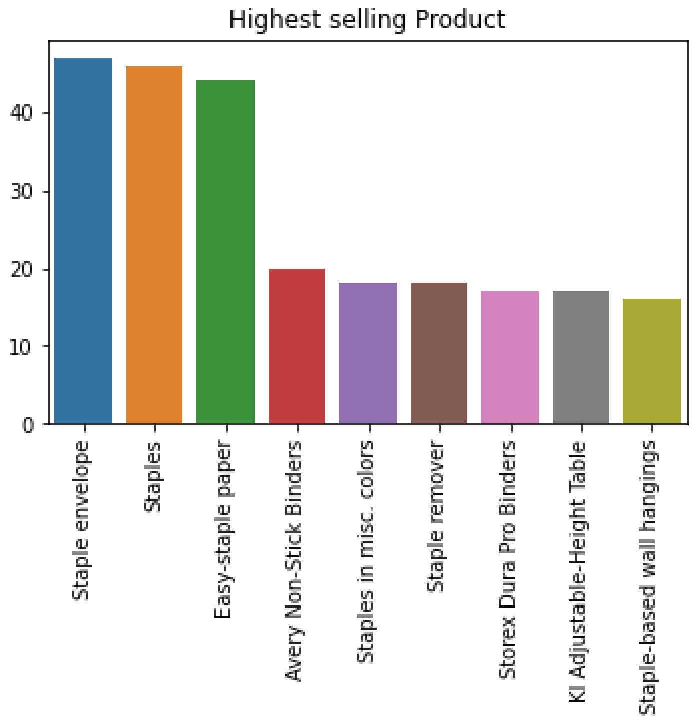

Highest Sub-Category Count
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Binders
Paper
Furnishings
Phones
Storage
Art
Accessories
Chairs
Appliances
Labels

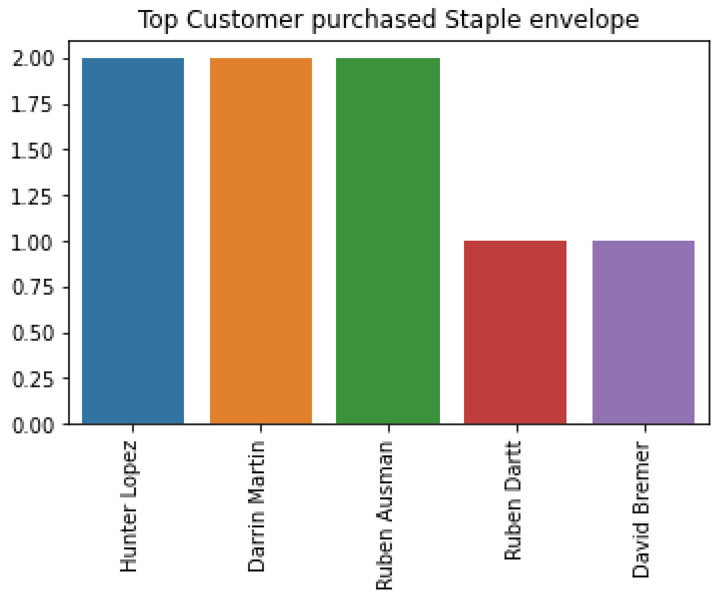

Highest Product Count Under Category Office Supplies
40
30
20
10
0
Staple envelope
Staples
Easy-staple paper
Avery Non-Stick Binders
Staple remover

```
In [171]: data['Product Name'].value_counts().head()

Out[171]: Staple envelope            47
          Staples                    46
          Easy-staple paper          44
          Avery Non-Stick Binders    20
          Staples in misc. colors    18
          Name: Product Name, dtype: int64
```

In [346]:
```
plt.figure(figsize=(10,8))

plt.subplot(2,2,1)
plt.title('Highest selling Product')
sns.barplot(data['Product Name'].value_counts().head(9).index, data['Product Name'].value_counts().head(9).values)
plt.xticks(rotation=90)

plt.subplot(2,2,2)
plt.title('Top Customer purchased Staple envelope')
sns.barplot(data[data['Product Name']=='Staple envelope']['Customer Name'].value_counts().head().index, data[data['Pro
plt.xticks(rotation=90)

plt.tight_layout()
plt.show()
```

```
C:\Users\Dell\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as k
eyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
C:\Users\Dell\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as k
eyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



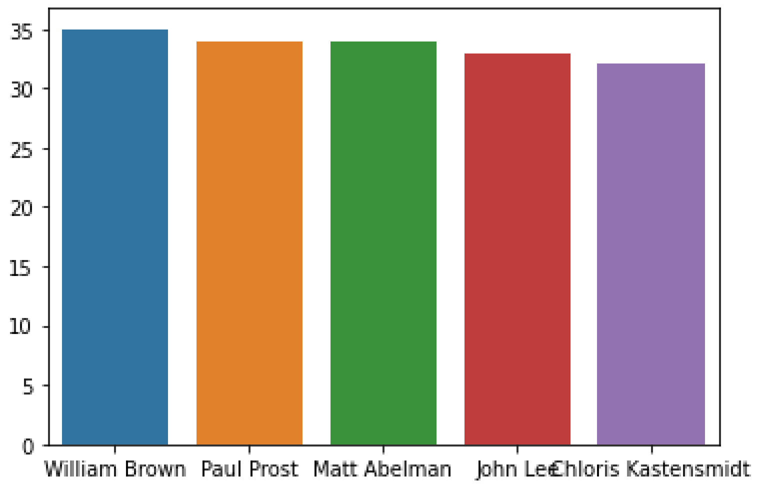


```
In [389]: x = data['Customer Name'].value_counts().sort_values(ascending=False).head().index
          y = data['Customer Name'].value_counts().sort_values(ascending=False).head().values
```

In [390]:
```
sns.barplot(x,y)
```

```
C:\Users\Dell\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as k
eyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[390]:
```
<AxesSubplot:>
```



In [349]:
```
data[data['Customer Name']=='William Brown']['City'].value_counts()
```

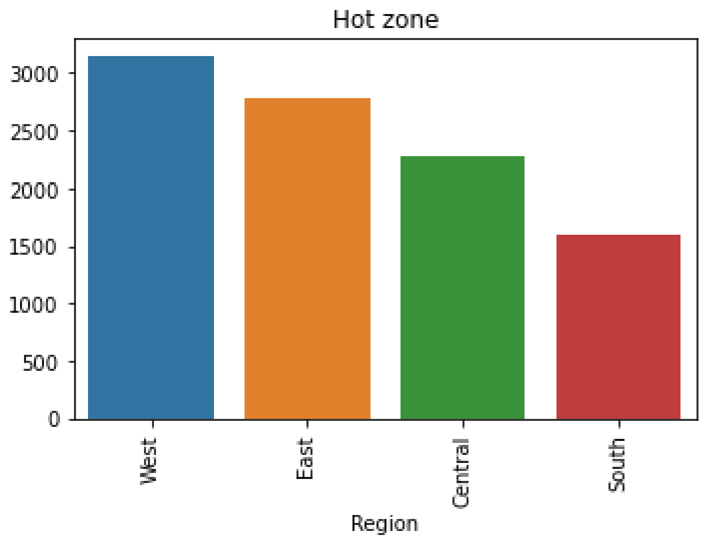
Out[349]:
```
Anaheim          11
Los Angeles       9
Philadelphia      5
New York City     3
Redmond           3
Grand Prairie     2
Concord           1
Urbandale         1
Name: City, dtype: int64
```

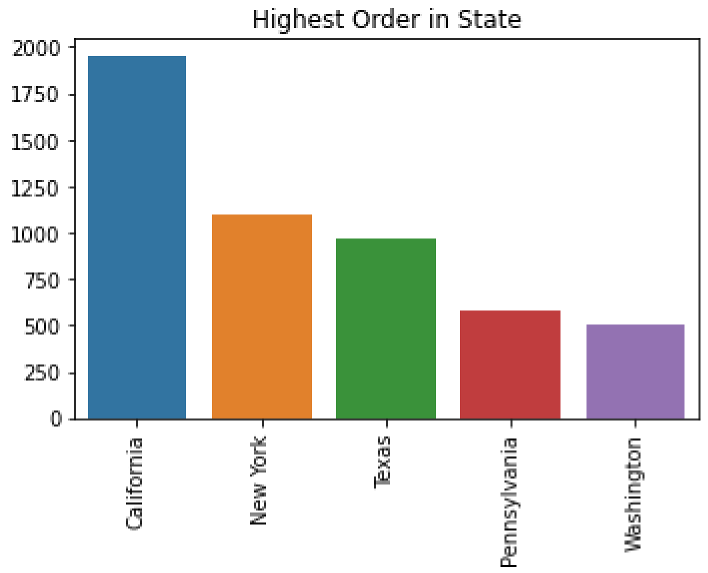
```
In [353]: data[data['Customer Name']=='William Brown'][['Product Name','Category']].value_counts().head()
```

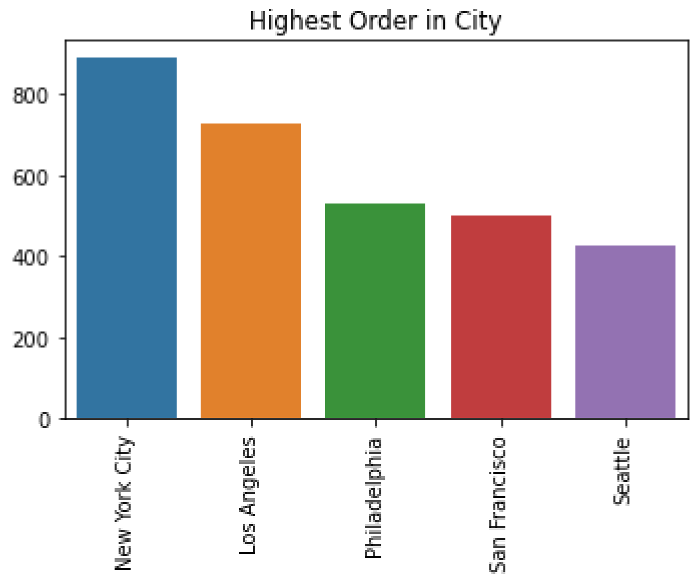
```
Out[353]: Product Name                                          Category
          Fellowes 8 Outlet Superior Workstation Surge Protector  Office Supplies    2
          #10 Gummed Flap White Envelopes, 100/Box              Office Supplies    1
          Polycom SoundPoint Pro SE-225 Corded phone            Technology         1
          Logitech Desktop MK120 Mouse and keyboard Combo       Technology         1
          Microsoft Natural Ergonomic Keyboard 4000             Technology         1
          dtype: int64
```

In [400]:
```
plt.figure(figsize=(10,8))

plt.subplot(2,2,1)
plt.title('Hot zone')
sns.barplot(data.groupby('Region')['Sales'].count().sort_values(ascending=False).index, data.groupby('Region')['Sales'
plt.xticks(rotation=90)

plt.subplot(2,2,2)
plt.title('Highest Order in State')
sns.barplot(data['State'].value_counts().head().index, data['State'].value_counts().head().values)
plt.xticks(rotation=90)

plt.subplot(2,2,3)
plt.title('Highest Order in City')
sns.barplot(data['City'].value_counts().head().index, data['City'].value_counts().head().values)
plt.xticks(rotation=90)

plt.tight_layout()
plt.show()
```

```
C:\Users\Dell\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as k
eyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
C:\Users\Dell\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as k
eyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
C:\Users\Dell\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as k
eyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```


Hot zone
3000
2500
2000
1500
1000
500
0
West
East
Central
South
Region


Highest Order in State
2000
1750
1500
1250
1000
750
500
250
0
California
New York
Texas
Pennsylvania
Washington

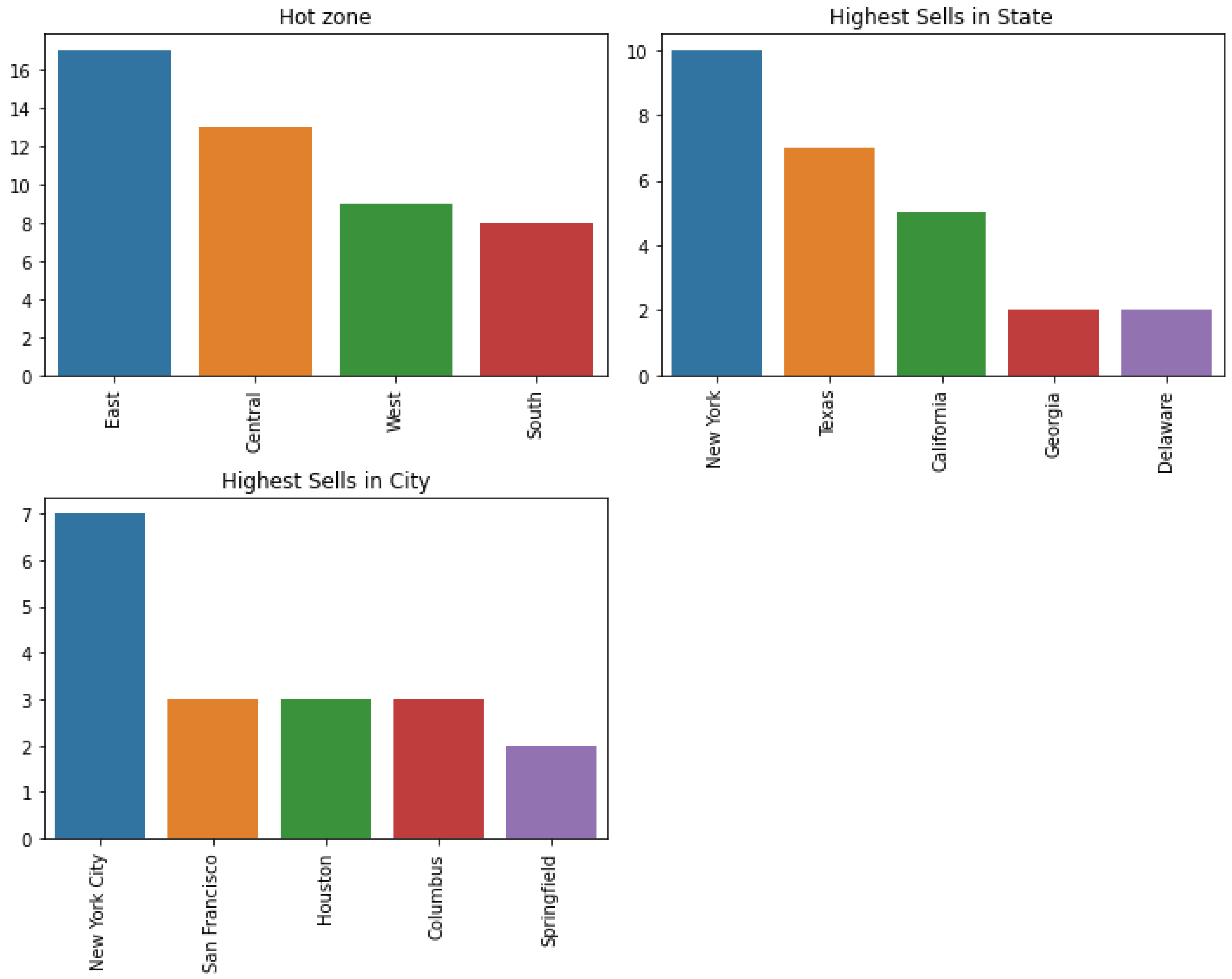

Highest Order in City
800
600
400
200
0
New York City
Los Angeles
Philadelphia
San Francisco
Seattle

In [396]:
```
data.groupby('City')['Sales'].count().sort_values(ascending=False)
```

Out[396]:
```
City
New York City       891
Los Angeles         728
Philadelphia        532
San Francisco       500
Seattle             426
Houston             374
Chicago             308
Columbus            221
San Diego           170
Springfield         161
Dallas              156
Jacksonville        125
Detroit             115
Newark               92
Jackson              82
Richmond             81
Columbia             81
Aurora               68
```

In [220]:
```
plt.figure(figsize=(10,8))

plt.subplot(2,2,1)
plt.title('Hot zone')
sns.barplot(data[data['Product Name']=='Staple envelope']['Region'].value_counts().head().index, data[data['Product Na
plt.xticks(rotation=90)

plt.subplot(2,2,2)
plt.title('Highest Sells in State')
sns.barplot(data[data['Product Name']=='Staple envelope']['State'].value_counts().head().index, data[data['Product Nam
plt.xticks(rotation=90)

plt.subplot(2,2,3)
plt.title('Highest Sells in City')
sns.barplot(data[data['Product Name']=='Staple envelope']['City'].value_counts().head().index, data[data['Product Name
plt.xticks(rotation=90)

plt.tight_layout()
plt.show()
```

```
C:\Users\Dell\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as k
eyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
C:\Users\Dell\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as k
eyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
C:\Users\Dell\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as k
eyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```


Hot zone
0
2
4
6
8
10
12
14
16
East
Central
West
South
Highest Sells in State
0
2
4
6
8
10
New York
Texas
California
Georgia
Delaware
Highest Sells in City
0
1
2
3
4
5
6
7
New York City
San Francisco
Houston
Columbus
Springfield

In [238]:
```
data['Order_Year'].value_counts()
```

Out[238]:
```
2018    3258
2017    2534
2016    2055
2015    1953
Name: Order_Year, dtype: int64
```

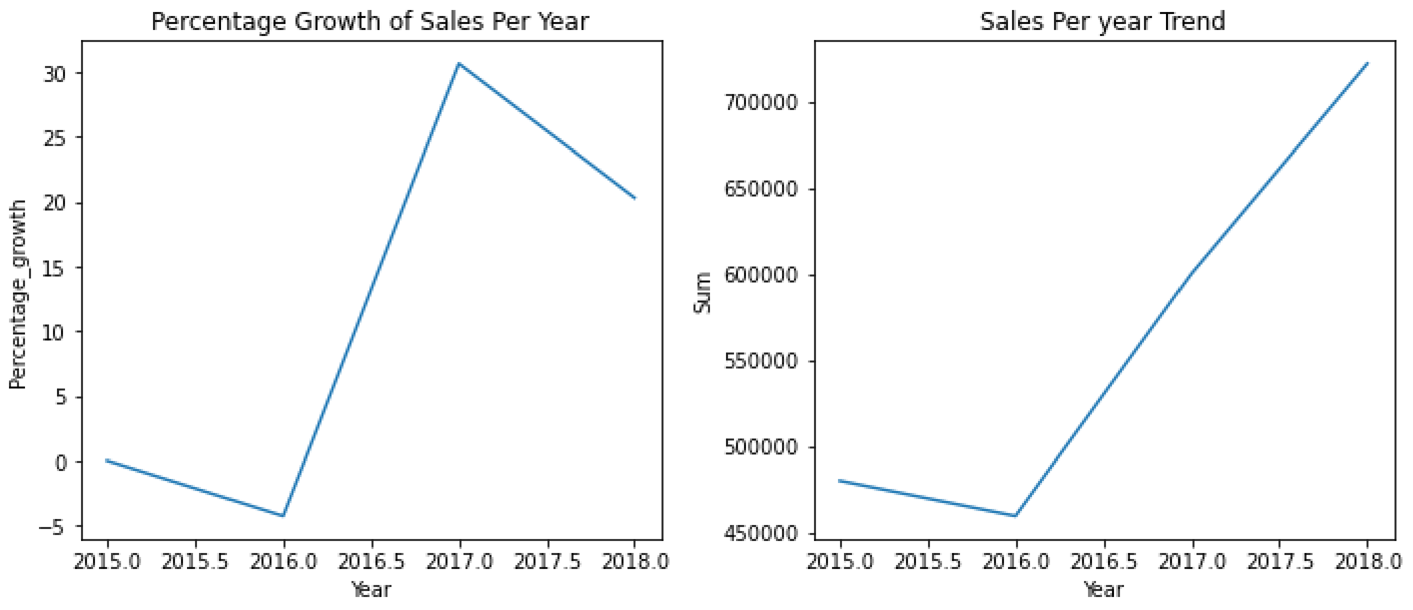
In [233]:
```
year_sum = pd.DataFrame(data.groupby('Year')['Sales'].sum()).reset_index()
year_sum['Percentage_growth'] = round(year_sum['Sales'].pct_change() * 100, 2)
year_sum['Percentage_growth'] = year_sum['Percentage_growth'].fillna(0)
year_sum
```

Out[233]:

	Year	Sales	Percentage_growth
0	2015	479856.2081	0.00
1	2016	459436.0054	-4.26
2	2017	600192.5500	30.64
3	2018	722052.0192	20.30

In [234]:
```
plt.figure(figsize=(10,8))
plt.subplot(2,2,1)
plt.title('Percentage Growth of Sales Per Year')
sns.lineplot(year_sum['Year'],year_sum['Percentage_growth'])

plt.subplot(2,2,2)
plt.title('Sales Per year Trend')
plt.ylabel('Sum')
sns.lineplot(data.groupby('Year')['Sales'].sum().index, data.groupby('Year')['Sales'].sum().values)

plt.tight_layout()
plt.show()
```

```
C:\Users\Dell\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as k
eyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
C:\Users\Dell\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as k
eyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



```
In [246]: data['Ship Mode'].value_counts()
```

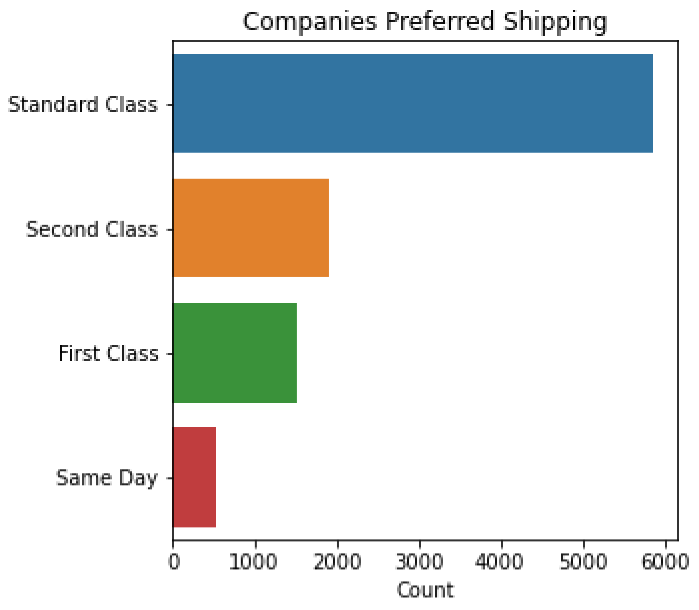
```
Out[246]: Standard Class    5859
          Second Class      1902
          First Class       1501
          Same Day           538
          Name: Ship Mode, dtype: int64
```

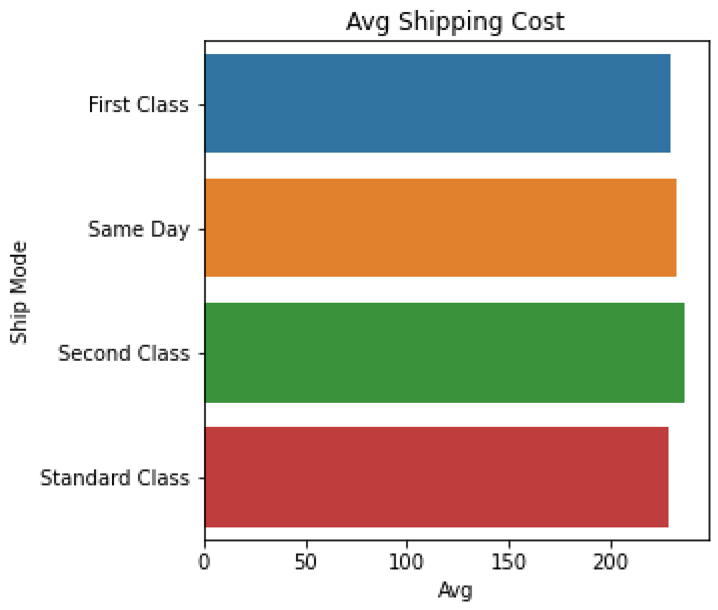
```
In [328]: data.groupby('Ship Mode')['Sales'].mean()
```

```
Out[328]: Ship Mode
          First Class       230.228020
          Same Day          232.749143
          Second Class      236.547939
          Standard Class    228.849856
          Name: Sales, dtype: float64
```

```
In [260]: plt.figure(figsize=(10,8))
          plt.subplot(2,2,1)
          plt.title('Companies Preferred Shipping')
          sns.barplot(data['Ship Mode'].value_counts().head().values, data['Ship Mode'].value_counts().head().index)
          plt.xlabel('Count')

          plt.subplot(2,2,2)
          plt.title('Avg Shipping Cost')
          sns.barplot(data.groupby('Ship Mode')['Sales'].mean().values,data.groupby('Ship Mode')['Sales'].mean().index)
          plt.xlabel('Avg')

          plt.tight_layout()
          plt.show()
```

```
C:\Users\Dell\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as k
eyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
C:\Users\Dell\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as k
eyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```





In [412]:
```
data.head(3)
```

Out[412]:

	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	State	Postal Code	Region	Product ID	Category	Sub-Category
0	CA-2017-152156	2017-08-11	2017-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky	42420.0	South	FUR-BO-10001798	Furniture	Bookcases
1	CA-2017-152156	2017-08-11	2017-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky	42420.0	South	FUR-CH-10000454	Furniture	Chairs
2	CA-2017-138688	2017-12-06	2017-06-16	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	California	90036.0	West	OFF-LA-10000240	Office Supplies	Labels

In [419]:
```
data['Order_month'] = data['Order Date'].dt.month
data['Year_month'] = data['Order Date'].dt.year
```

In [420]:
```
data.drop('Year_month',axis=1,inplace=True)
```

```
In [421]: data.head(3)
```

Out[421]:

	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	State	Postal Code	Region	Product ID	Category	Sub-Category
0	CA-2017-152156	2017-08-11	2017-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky	42420.0	South	FUR-BO-10001798	Furniture	Bookcases
1	CA-2017-152156	2017-08-11	2017-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky	42420.0	South	FUR-CH-10000454	Furniture	Chairs
2	CA-2017-138688	2017-12-06	2017-06-16	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	California	90036.0	West	OFF-LA-10000240	Office Supplies	Labels

```
In [427]: data.groupby('Order_Year')['Sales'].count()
```

```
Out[427]: Order_Year
2015    1953
2016    2055
2017    2534
2018    3258
Name: Sales, dtype: int64
```

```
In [425]: data.groupby(['Order_Year','Order_month'])['Sales'].count()
```

```
Out[425]: Order_Year  Order_month
          2015        1              126
                      2               84
                      3              161
                      4              121
                      5              146
                      6              135
                      7              154
                      8              145
                      9              238
                      10             145
                      11             258
                      12             240
          2016        1               86
                      2              102
                      3              144
                      4              159
                      5              174
                      6              149
                      7              128
                      8              170
                      9              271
                      10             153
                      11             279
                      12             240
          2017        1              154
                      2              122
                      3              190
                      4              186
                      5              257
                      6              180
                      7              201
                      8              236
                      9              225
                      10             231
                      11             288
                      12             264
          2018        1              209
                      2              228
                      3              302
                      4              225
```

```
                        5               241
                        6               220
                        7               241
                        8               248
                        9               382
                        10              272
                        11              369
                        12              321
            Name: Sales, dtype: int64
```

```
In [426]: moth = pd.DataFrame(data.groupby(['Order_Year','Order_month'])['Sales'].count()).reset_index()
          moth['% increase on ']
```

Out[426]:

	Order_Year	Order_month	Sales
0	2015	1	126
1	2015	2	84
2	2015	3	161
3	2015	4	121
4	2015	5	146
5	2015	6	135
6	2015	7	154
7	2015	8	145
8	2015	9	238
9	2015	10	145
10	2015	11	258
11	2015	12	240
12	2016	1	86
13	2016	2	102
14	2016	3	144
15	2016	4	159
16	2016	5	174
17	2016	6	149
18	2016	7	128
19	2016	8	170
20	2016	9	271
21	2016	10	153
22	2016	11	279
23	2016	12	240
24	2017	1	154
25	2017	2	122

	Order_Year	Order_month	Sales
26	2017	3	190
27	2017	4	186
28	2017	5	257
29	2017	6	180
30	2017	7	201
31	2017	8	236
32	2017	9	225
33	2017	10	231
34	2017	11	288
35	2017	12	264
36	2018	1	209
37	2018	2	228
38	2018	3	302
39	2018	4	225
40	2018	5	241
41	2018	6	220
42	2018	7	241
43	2018	8	248
44	2018	9	382
45	2018	10	272
46	2018	11	369
47	2018	12	321

```
In [ ]:
```

In [417]:
```
data.groupby(data['Order Date'].dt.month)['Sales'].count()
```

Out[417]:
```
Order Date
1      575
2      536
3      797
4      691
5      818
6      684
7      724
8      799
9     1116
10     801
11    1194
12    1065
Name: Sales, dtype: int64
```

In [332]:
```
data[data['Ship Mode']=='Standard Class'].groupby(data['Order Date'].dt.year)['Sales'].count()
```

Out[332]:
```
Order Date
2015    1207
2016    1269
2017    1517
2018    1866
Name: Sales, dtype: int64
```

In [333]:
```
data[data['Ship Mode']=='Standard Class'].groupby(data['Ship Date'].dt.year)['Sales'].count()
```

Out[333]:
```
Ship Date
2015    1173
2016    1285
2017    1506
2018    1859
2019      36
Name: Sales, dtype: int64
```

In [277]:
```
ship_year=pd.DataFrame(data.groupby(data['Ship Date'].dt.year)['Sales'].count()).reset_index()
ship_year
```

Out[277]:

	Ship Date	Sales
0	2015	1902
1	2016	2083
2	2017	2524
3	2018	3249
4	2019	42

In [282]:
```
Order_year=pd.DataFrame(data.groupby(data['Order Date'].dt.year)['Sales'].count()).reset_index()
Order_year
```

Out[282]:

	Order Date	Sales
0	2015	1953
1	2016	2055
2	2017	2534
3	2018	3258

In [306]:
```
Order_year['Percentage_increase'] = round(Order_year['Sales'].pct_change()*100,2)
Order_year['Percentage_increase'] = Order_year['Percentage_increase'].fillna(0)
Order_year.rename(columns={'Percentage_increase':'Order_Percentage_increase'})
Order_year.rename(columns={'Order Date':'Year'},inplace=True)
```

In [309]:
```
Order_year.drop('Percentage_increase',axis=1,inplace=True)
```

In [310]:
```
Order_year
```

Out[310]:

	Year	Sales	Order_Percentage_increase
0	2015	1953	0.00
1	2016	2055	5.22
2	2017	2534	23.31
3	2018	3258	28.57

In [304]:
```
ship_year['Percentage_increase'] = round(ship_year['Sales'].pct_change()*100,2)
ship_year['Percentage_increase'] = ship_year['Percentage_increase'].fillna(0)
ship_year.rename(columns={'Percentage_increase':'Ship_Percentage_increase'})
ship_year.rename(columns={'Ship Date':'Year'},inplace=True)
```

In [305]:
```
ship_year
```

Out[305]:

	Year	Sales	Percentage_increase
0	2015	1902	0.00
1	2016	2083	9.52
2	2017	2524	21.17
3	2018	3249	28.72
4	2019	42	-98.71

In [313]:
```
final_increase = pd.merge(Order_year,ship_year,how='inner',on='Year')
final_increase['Failed_ship'] = final_increase['Sales_x']-final_increase['Sales_y']
final_increase
```

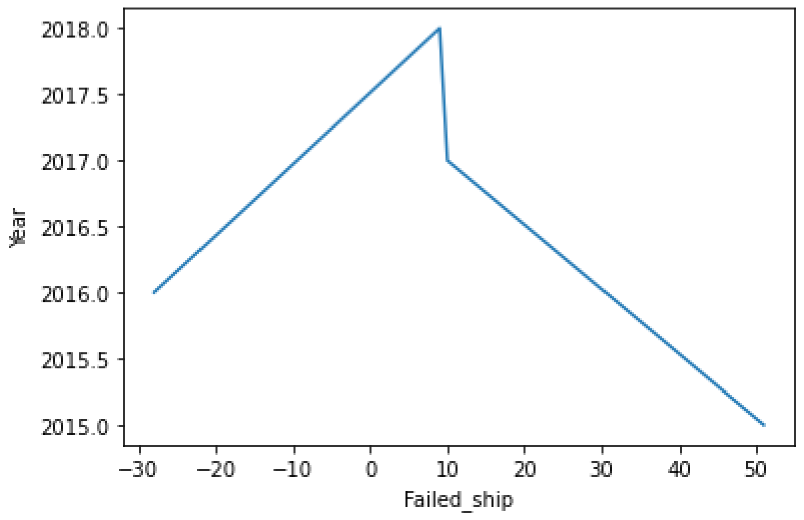
Out[313]:

	Year	Sales_x	Order_Percentage_increase	Sales_y	Percentage_increase	Failed_ship
0	2015	1953	0.00	1902	0.00	51
1	2016	2055	5.22	2083	9.52	-28
2	2017	2534	23.31	2524	21.17	10
3	2018	3258	28.57	3249	28.72	9

In [317]:
```
sns.lineplot(final_increase['Failed_ship'],final_increase['Year'])
```

```
C:\Users\Dell\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as k
eyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[317]:
```
<AxesSubplot:xlabel='Failed_ship', ylabel='Year'>
```



In [ ]:

```
1. The most frequently used segment is "Consumer," followed by "Corporate" and "Home Office." Over the years,
   all segments have shown a consistent increase in counts, with distributions following a binomial pattern.

2. The category with the highest Sells count is "Office Supplies," and the sub-category with the highest count is "Bin
   Among the most commonly ordered Office Supplies products, "Staple Envelope" emerges as the highest in demand.

3. Staple envelope is the highest demanding product which comes under Office Supplies Category.Hunter Lopez & Darrin M
   are the customer who order max time staple enelope.Highest demanding city and zone for staple envelope product is N
   San Francisco,Houston,Columbus and East follwed by Central.

4. William Brown ranks as the top customer with the highest number of orders. Among the maximum orders,
   Anaheim and Los Angeles stand out. The most commonly ordered Category is Office Supplies, with the Fellowes 8 Outle
   Superior Workstation Surge Protector being the top product.

5. California and New York have the highest purchase orders, while New York City and Los Angeles exhibit the highest,
   ordering habits. Maximum sales are concentrated in the West, East, Central, and South zones.

6. The percentage yearly growth experienced its highest peak in 2017 at 30% and its lowest point in 2016. Notably,
   the trend of percentage growth follows a pattern of initial decline, subsequent increase, and then another decline.

7. Shipping through Standard Class is a prevalent choice, and companies often prefer it for their shipments.

8. In 2015, there were 51 shipping failures, but in 2016, 28 pending orders were successfully cleared. The year 2018 w
   the highest successful shipping deliveries, with a minimal number of failed deliveries (only 9).
```