In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import datetime as dt
import seaborn as sns
```

In [2]:
```
df = pd.read_csv('superstore_orders.csv') # First dataset
```

In [3]:
```
dfa = pd.read_excel('superstore_salesagents.xlsx') # Second dataset
```

Out[3]:
```
df.head()
```

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Sales Agent ID	Country/Region	City	State	Postal Code	Regi
0	1	CA-2017-152156	08/11/2017	11/11/2017	Second Class	CG-12520	Claire Gute	16	United States	Henderson	Kentucky	42420.0	Sou
1	2	CA-2017-152156	08/11/2017	11/11/2017	Second Class	CG-12520	Claire Gute	16	United States	Henderson	Kentucky	42420.0	Sou
2	3	CA-2017-138688	12/06/2017	16/06/2017	Second Class	DV-13045	Darrin Van Huff	4	United States	Los Angeles	California	90036.0	We
3	4	US-2017-108966	11/10/2018	18/10/2018	Standard Class	SO-20335	Sean O'Donnell	9	United States	Fort Lauderdale	Florida	33311.0	Sou
4	5	US-2017-108966	11/10/2018	18/10/2018	Standard Class	SO-20335	Sean O'Donnell	9	United States	Fort Lauderdale	Florida	33311.0	Sou

In [4]:
```
df.info()
```

Out[4]:
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7585 entries, 0 to 7584
Data columns (total 20 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Row ID        7585 non-null   int64
 1   Order ID      7585 non-null   object
 2   Order Date    7585 non-null   object
 3   Ship Date     7585 non-null   object
 4   Ship Mode     7585 non-null   object
 5   Customer ID   7585 non-null   object
 6   Customer Name 7585 non-null   object
 7   Sales Agent ID 7585 non-null  int64
 8   Country/Region 7585 non-null  object
 9   City          7585 non-null   object
 10  State         7585 non-null   object
 11  Postal Code   7585 non-null   int64
 12  Region        7585 non-null   object
 13  Product ID    7585 non-null   object
 14  Category      7585 non-null   object
 15  Sub-Category  7585 non-null   object
 16  Product Name  7585 non-null   object
 17  Sales         7585 non-null   float64
 18  Quantity      7585 non-null   int64
 19  Costs         7585 non-null   float64
dtype: float64(4), int64(6), object(10)
memory usage: 1.1+ MB
```

In [5]:
```
dfa.info()
```

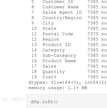
Out[5]:
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 65 entries, 0 to 65
Data columns (total 6 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Sales Agent ID 65 non-null    int64
 1   Manager       65 non-null     object
 2   Surname       65 non-null     object
 3   Level         65 non-null     object
 4   State         65 non-null     object
 5   age           65 non-null     int64
dtypes: int64(2), object(4)
memory usage: 2.9+ KB
```

In [6]:
```
Shipping = df['Ship Mode'].value_counts()
Shipping
```

Out[6]:
```
Standard Class    4466
Second Class      1427
First Class       1192
Same Day           499
Name: Ship Mode, dtype: int64
```

In [7]:
```
df['Ship Mode'].value_counts().plot.pie()
```

Out[7]:
```
<AxesSubplot:ylabel='Ship Mode'>
```



In [8]:
```
df.describe()
```

Out[8]:

	Row ID	Postal Code	Sales	Quantity	Costs
count	7585.000000	7585.000000	7585.000000	7585.000000	7585.000000
mean	4360.28389	54933.03377	233.39052	3.789094	0.204895
std	2913.97277	32034.170383	648.523613	2.216888	0.176531
min	1.000000	1040.000000	0.556000	1.000000	0.100000
25%	2436.000000	22304.000000	17.220000	2.000000	0.100000
50%	4979.000000	55407.000000	54.384000	3.000000	0.200000
75%	7024.000000	90004.000000	211.246000	5.000000	0.200000
max	9993.000000	99301.000000	22638.480000	14.000000	0.800000

In [9]:
```
print(df.isnull()).sum())
```

Out[9]:
```
Row ID          0
Order ID        0
Order Date      0
Ship Date       0
Ship Mode       0
Customer ID     0
Customer Name   0
Sales Agent ID  0
Country/Region  0
City            0
State           0
Postal Code     0
Region          0
Product ID      0
Category        0
Sub-Category    0
Product Name    0
Sales           0
Quantity        0
Costs           0
dtype: int64
```

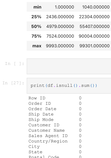
In [10]:
```
region_order_count = df['Region'].value_counts()
region_order_count
```

Out[10]:
```
West     2341
East     1848
Central  1571
South    1204
Name: Region, dtype: int64
```

In [11]:
```
```

In [12]:
```
region_gross_sales = df.groupby(['Region']).Sales.agg([sum]).sort_values(by='sum', ascending=False)
region_gross_sales.rename(columns={'sum':'Sales'})
region_gross_sales.round(2)
print(region_gross_sales)
plt.xlabel("Sales Record")
plt.ylabel("Region")
plt.title("Total Sales by Region")
```

Out[12]:
```
                 sum
Region
West    539710.0710
East    490981.0836
Central 391721.0604
South   293365.9630
Text(0.5, 1.0, 'Total Sales by Region')
```



In [13]:
```
```

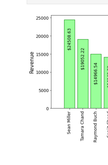
In [14]:
```
Top_customers = df.groupby(["Customer Name"]).sum().sort_values(["Sales"], ascending=False).head(20) # Sort the t
Top_customers = Top_customers[["Sales"]].round(2) # Round off the Sales Value up to 2 decimal places
Top_customers.reset_index(inplace=True) # Since we have used groupby, we will have to reset the index to add the t
```

In [15]:
```
plt.figure(figsize = (15,5)) # width and height of figure is defined in inches
plt.title("Most Valuable Customers (2017-2020)", fontsize=18)
plt.bar(Top_customers["Customer Name"], Top_customers["Sales"],color='#99FE99',edgecolor='green', linewidth =
plt.xlabel("Customers",fontsize=15) # x axis shows the customers
plt.ylabel("Revenue",fontsize=15) # y axis shows the Revenue
plt.xticks(fontsize=2, rotation=90)
plt.yticks(fontsize=2)
for k,v in Top_customers["Sales"].items():# Pts show the exact revenue generated in the figure
    plt.text(k,v+800,'%'+ str(v), fontsize=2,rotation=90,color='k', horizontalalignment='center');
```

In [16]:
```
```

In [17]:
```
Top_states = df.groupby(["State"]).sum().sort_values(["Sales"], ascending=False).head(20) # Sort the States as p
Top_states = Top_states[["Sales"]].round(2) # Round off the Sales Value up to 2 decimal places
Top_states.reset_index(inplace=True) # Since we have used groupby, we will have to reset the index to add the t
```

In [18]:
```
plt.figure(figsize = (15,5)) # width and height of figure is defined in inches
plt.title("States which generated Highest Revenue (2017-2020)", fontsize=18)
plt.bar(Top_states["State"], Top_states["Sales"],color='#FF8F01',edgecolor='Red', linewidth = 1)
plt.xlabel("States",fontsize=15) # x axis shows the States
plt.ylabel("Revenue",fontsize=15) # y axis shows the Revenue
plt.xticks(fontsize=2, rotation=90)
plt.yticks(fontsize=2)
for k,v in Top_states["Sales"].items():# Pts show the exact revenue generated in the figure
    if v==350000:
        plt.text(k,v+35000,'%'+ str(v), fontsize=2,rotation=90,color='k', horizontalalignment='center');
    else:
        plt.text(k,v+15000,'%'+ str(v), fontsize=2,rotation=90,color='k', horizontalalignment='center');
```



In [19]:
```
```

In [20]:
```
df1 = df['Category','Sales']]
df1.groupby(['Category'])
cor_mat = df1.corr()
mask = np.array(cor_mat)
mask[np.tril_indices_from(mask)]=False
fig = plt.gcf()
fig.set_size_inches(15,4)
sns.heatmap(data = cor_mat, mask = mask, square = True, annot = True, cbar = True);
```

In [21]:
```
```

In [22]:
```
data = df.groupby(['Ship Mode','Order Date')]['Sales'].mean()
L = [data[] for j in df['Ship Mode'].unique()]
la = list(df['Ship Mode'].unique())
fig0 = plt.figure(figsize=(1,5))
axes0 = fig0.add_axes([0.1,0.1,0.8,0.8])
for i,l in enumerate(la):
axes0.set_ylim('Avg daily sales per week for each mode of shipment',fontsize=16)
axes0.set_xlabel('Ship Mode',fontsize=13)
axes0.set_ylabel('Average daily sales per week',fontsize=13)
```

Out[22]:
```
Text(0, 0.5), 'Average daily sales per week')
```

In [23]:
```
col = df.columns
print(col)
```

Out[23]:
```
Index(['Row ID', 'Order ID', 'Order Date', 'Ship Date', 'Ship Mode',
       'Customer ID', 'Customer Name', 'Sales Agent ID', 'Country/Region',
       'City', 'State', 'Postal Code', 'Region', 'Product ID',
       'Sub-Category', 'Product Name', 'Sales', 'Quantity', 'Costs',
       dtype='object')
```

In [24]:
```
Correlation map
f,ax =plt.subplots(figsize=(8, 5))
sns.heatmap(df.corr(), annot=True, linewidth=.5, fmt= '.1f',ax=ax)
```

Out[24]:
```
<AxesSubplot:>
```