The Sparks Foundation Internship

***Perform ‘Exploratory Data Analysis’ on dataset ‘SampleSuperstore’**

Table of Contents

1. Import Dataset

In [1]:

```
# Essential package loading
import pandas as pd
import numpy as np
%matplotlib inline
%config InlineBackend.figure_format = 'retina'
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_style('whitegrid')
plt.rcParams.update({'font.size': 18})
```

In [2]:

```
# Data loading
df_store = pd.read_csv("SampleSuperstore.csv")
```

Showing few lines:

In [3]:

```
df_store.head(5)
```

Out[3]:

	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sub-Category
0	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Bookcases
1	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Chairs
2	Second Class	Corporate	United States	Los Angeles	California	90036	West	Office Supplies	Labels
3	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Furniture	Tables
4	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Office Supplies	Storage

Checking for Duplicates :

In [4]:

```
sum(df_store.duplicated())
```

Out[4]:

17

Rows and Columns (Shape):

In [5]:

```
df_store.shape
```

Out[5]:

(9994, 13)

2. Exploratory Data Analysis

Correlation

In [6]:

```
df_store.corr()['Profit'].drop(['Profit'])
```

Out[6]:

```
Postal Code   -0.029961
Sales          0.479064
Quantity       0.066253
Discount      -0.219487
Name: Profit, dtype: float64
```

- **This shows that profit could be having some correlation with sales figure**

Heat Map

In [7]:

```
sns.heatmap(df_store.corr(),cmap='rocket_r',annot=True)
```

Out[7]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x215ce909f08>
```

Highest Selling By State

In [8]:

```
df_sales_state = df_store.groupby(['State']).sum().sort_values(by=['Sales'], ascending=
False).head(20)
df_sales_state.reset_index(drop=False, inplace=True)
df_sales_state[['State', 'Sales', 'Profit']]
```

Out[8]:

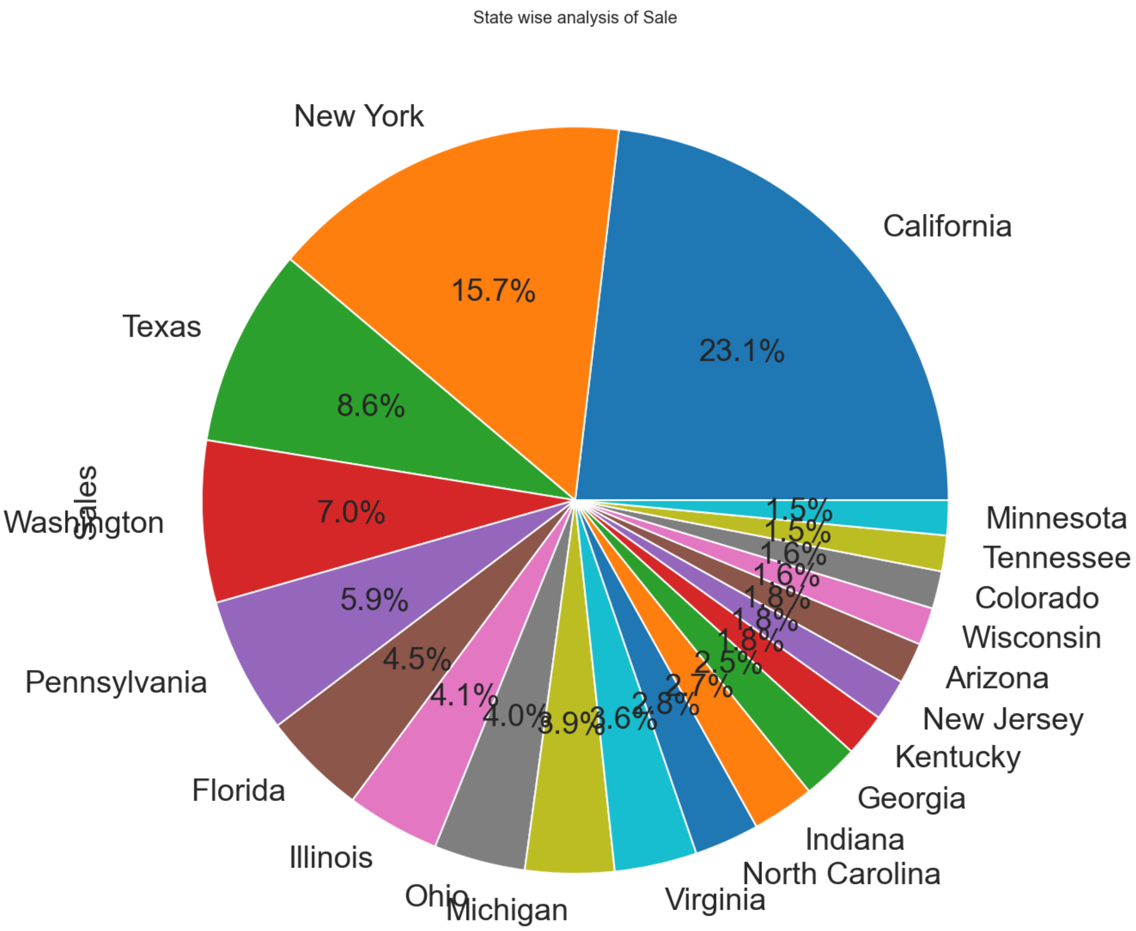
	State	Sales	Profit
0	California	457687.6315	76381.3871
1	New York	310876.2710	74038.5486
2	Texas	170188.0458	-25729.3563
3	Washington	138641.2700	33402.6517
4	Pennsylvania	116511.9140	-15559.9603
5	Florida	89473.7080	-3399.3017
6	Illinois	80166.1010	-12607.8870
7	Ohio	78258.1360	-16971.3766
8	Michigan	76269.6140	24463.1876
9	Virginia	70636.7200	18597.9504
10	North Carolina	55603.1640	-7490.9122
11	Indiana	53555.3600	18382.9363
12	Georgia	49095.8400	16250.0433
13	Kentucky	36591.7500	11199.6966
14	New Jersey	35764.3120	9772.9138
15	Arizona	35282.0010	-3427.9246
16	Wisconsin	32114.6100	8401.8004
17	Colorado	32108.1180	-6527.8579
18	Tennessee	30661.8730	-5341.6936
19	Minnesota	29863.1500	10823.1874

In [9]:

```
df_sales_state0 = df_sales_state[['State', 'Sales']]
df_sales_state0.set_index('State', inplace=True)
df_sales_state0['Sales'].plot(kind='pie',
                              figsize = (10,10),
                              autopct='%1.1f%%',
                              shadow=False)
plt.title('State wise analysis of Sale',fontsize=10)
```

Out[9]:

```
Text(0.5, 1.0, 'State wise analysis of Sale')
```



Highest Selling By City

In [10]:

```
df_sales_city = df_store.groupby(['City']).sum().sort_values(by=['Sales'], ascending=False).head(20)
df_sales_city.reset_index(drop=False, inplace=True)
df_sales_city[['City', 'Sales', 'Profit']]
```

Out[10]:

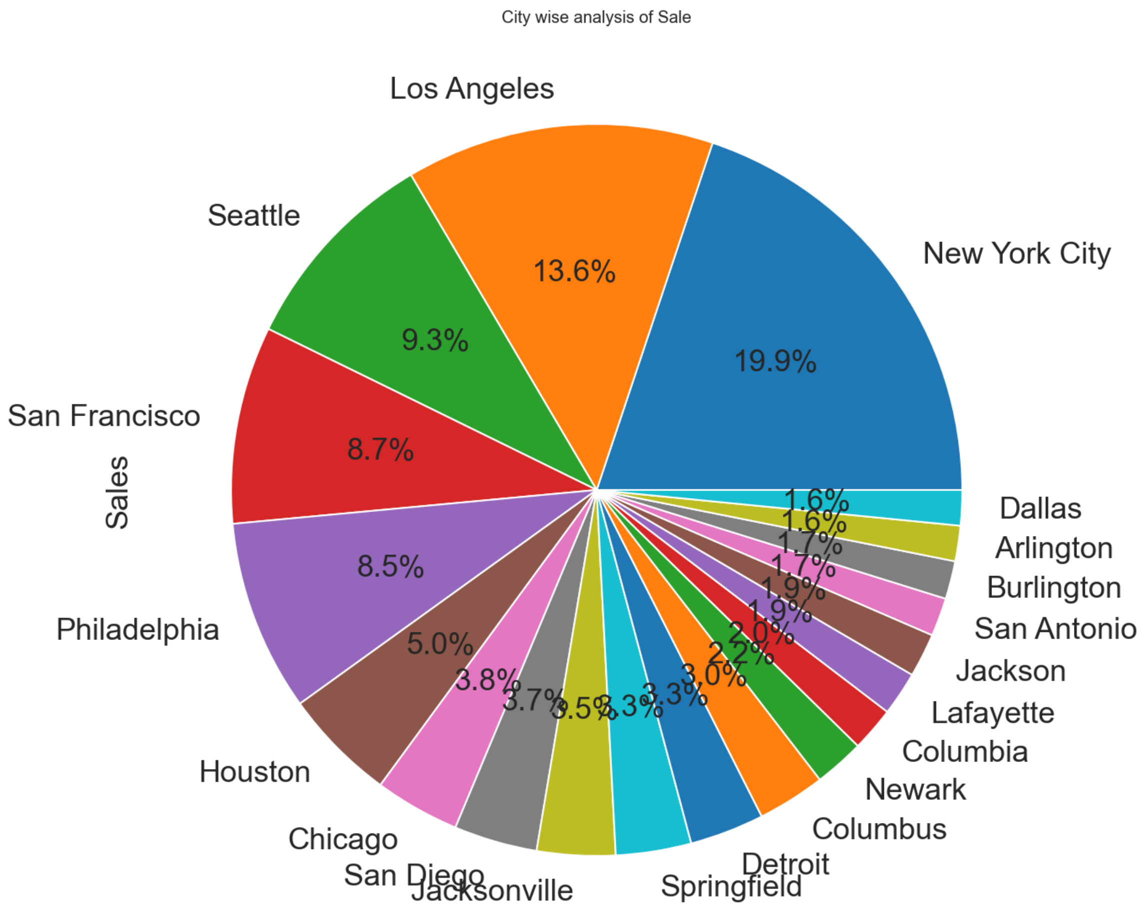
	City	Sales	Profit
0	New York City	256368.1610	62036.9837
1	Los Angeles	175851.3410	30440.7579
2	Seattle	119540.7420	29156.0967
3	San Francisco	112669.0920	17507.3854
4	Philadelphia	109077.0130	-13837.7674
5	Houston	64504.7604	-10153.5485
6	Chicago	48539.5410	-6654.5688
7	San Diego	47521.0290	6377.1960
8	Jacksonville	44713.1830	-2323.8350
9	Springfield	43054.3420	6200.6974
10	Detroit	42446.9440	13181.7908
11	Columbus	38706.2430	5897.1013
12	Newark	28576.1190	5793.7588
13	Columbia	25283.3240	5606.1167
14	Lafayette	25036.2000	10018.3876
15	Jackson	24963.8580	7581.6828
16	San Antonio	21843.5280	-7299.0502
17	Burlington	21668.0820	-3622.8772
18	Arlington	20214.5320	4169.6969
19	Dallas	20131.9322	-2846.5257

In [11]:

```
df_sales_city0 = df_sales_city[['City', 'Sales']]
df_sales_city0.set_index('City', inplace=True)
df_sales_city0['Sales'].plot(kind='pie',
                             figsize = (10,10),
                             autopct='%1.1f%%',
                             shadow=False)
plt.title('City wise analysis of Sale',fontsize=10)
```

Out[11]:

```
Text(0.5, 1.0, 'City wise analysis of Sale')
```



Lowest Profit where Profit < 0 (Count)

In [12]:

```
lowest = df_store.query('Profit < 0').sort_values(by=['Profit'])
lowest['Profit'].count()
```

Out[12]:

```
1871
```

In [13]:

```
#few lines:
lowest.head(5)
```

Out[13]:

	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sub Categor
7772	Standard Class	Consumer	United States	Lancaster	Ohio	43130	East	Technology	Machine
683	Same Day	Corporate	United States	Burlington	North Carolina	27217	South	Technology	Machine
9774	Standard Class	Consumer	United States	San Antonio	Texas	78207	Central	Office Supplies	Binder
3011	Standard Class	Home Office	United States	Louisville	Colorado	80027	West	Technology	Machine
4991	Standard Class	Corporate	United States	Chicago	Illinois	60653	Central	Office Supplies	Binder

Finding state with highest -ve Profit

In [14]:

```
df_state = df_store.groupby('State').sum().sort_values(by=['Profit']).head(20)
df_state.reset_index(drop=False, inplace=True)
df_state
```

Out[14]:

	State	Postal Code	Sales	Quantity	Discount	Profit
0	Texas	75747693	170188.0458	3724	364.64	-25729.3563
1	Ohio	20579836	78258.1360	1759	152.40	-16971.3766
2	Pennsylvania	11190565	116511.9140	2153	192.90	-15559.9603
3	Illinois	29873772	80166.1010	1845	191.90	-12607.8870
4	North Carolina	6994384	55603.1640	983	70.60	-7490.9122
5	Colorado	14613828	32108.1180	693	57.60	-6527.8579
6	Tennessee	6890574	30661.8730	681	53.30	-5341.6936
7	Arizona	19102126	35282.0010	862	68.00	-3427.9246
8	Florida	12640225	89473.7080	1379	114.65	-3399.3017
9	Oregon	12072125	17431.1500	499	35.80	-1190.4705
10	Wyoming	82001	1603.1360	4	0.20	100.1960
11	West Virginia	104012	1209.8240	18	0.30	185.9216
12	North Dakota	406721	919.9100	30	0.00	230.1497
13	South Dakota	686730	1315.5600	42	0.00	394.8283
14	Maine	34725	1270.5300	35	0.00	454.4862
15	Idaho	1752709	4382.4860	64	1.80	826.7231
16	Kansas	1603798	2914.3100	74	0.00	836.4435
17	District of Columbia	200160	2865.0200	40	0.00	1059.5893
18	New Mexico	3241556	4783.5220	151	2.20	1157.1161
19	Iowa	1537707	4579.7600	112	0.00	1183.8119

In [15]:

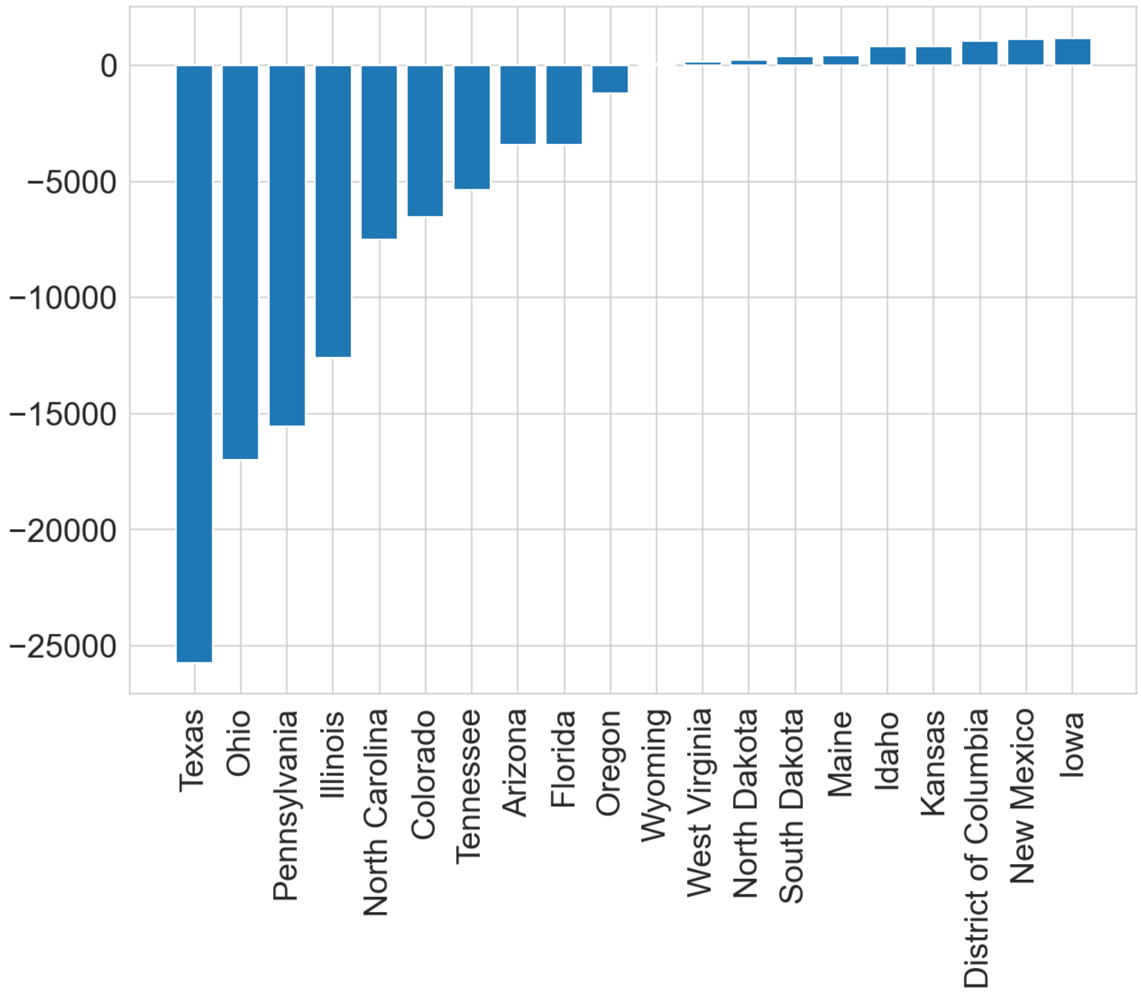
```
#top 20 states having heighest -ve/+ve profit

State = df_state['State']
profit = df_state['Profit']

# Figure Size
fig = plt.figure(figsize =(10, 7))

# Horizontal Bar Plot
plt.bar(State, profit)
plt.xticks(State, rotation='vertical')

# Show Plot
plt.show()
```



In [16]:

```
df_sales_state[['State', 'Sales', 'Profit']].query('Profit < 0')
```

Out[16]:

	State	Sales	Profit
2	Texas	170188.0458	-25729.3563
4	Pennsylvania	116511.9140	-15559.9603
5	Florida	89473.7080	-3399.3017
6	Illinois	80166.1010	-12607.8870
7	Ohio	78258.1360	-16971.3766
10	North Carolina	55603.1640	-7490.9122
15	Arizona	35282.0010	-3427.9246
17	Colorado	32108.1180	-6527.8579
18	Tennessee	30661.8730	-5341.6936

- **Above Data shows that despite having larger number of sales Texas, Pennsylvania, Florida ans other States are losing money**

Finding city with highest -ve Profit

In [17]:

```
df_store['City'].nunique()
```

Out[17]:

```
531
```

In [18]:

```
df_city = df_store.groupby('City').sum().sort_values(by=['Profit']).head(20)
df_city.reset_index(drop=False, inplace=True)
df_city
```

Out[18]:

	City	Postal Code	Sales	Quantity	Discount	Profit
0	Philadelphia	10275302	109077.0130	1981	175.50	-13837.7674
1	Houston	29052387	64504.7604	1466	143.14	-10153.5485
2	San Antonio	4614213	21843.5280	247	22.60	-7299.0502
3	Lancaster	1980720	9891.4640	171	14.50	-7239.0684
4	Chicago	19037248	48539.5410	1132	120.50	-6654.5688
5	Burlington	516678	21668.0820	105	3.40	-3622.8772
6	Dallas	11802703	20131.9322	555	56.30	-2846.5257
7	Phoenix	5356449	11000.2570	224	22.30	-2790.8832
8	Aurora	4777612	11656.4780	258	24.00	-2691.7386
9	Jacksonville	3843200	44713.1830	429	35.85	-2323.8350
10	Memphis	1143270	5942.3410	116	8.40	-1479.0400
11	Louisville	3247710	12345.8060	221	8.10	-1430.3129
12	Medina	398304	2477.7220	38	3.90	-1343.0446
13	Round Rock	550648	4854.0528	23	1.92	-1183.4313
14	Knoxville	910032	3928.1660	81	6.20	-1165.0755
15	Miami	1890806	8673.0745	215	18.65	-1150.3704
16	Rockford	672177	3166.2280	64	5.60	-1149.5078
17	Clarksville	259294	2217.7300	27	1.90	-1055.3532
18	Bethlehem	90090	1689.6340	18	1.90	-1003.0958
19	Colorado Springs	2022650	3694.0090	110	8.30	-956.6841

In [19]:

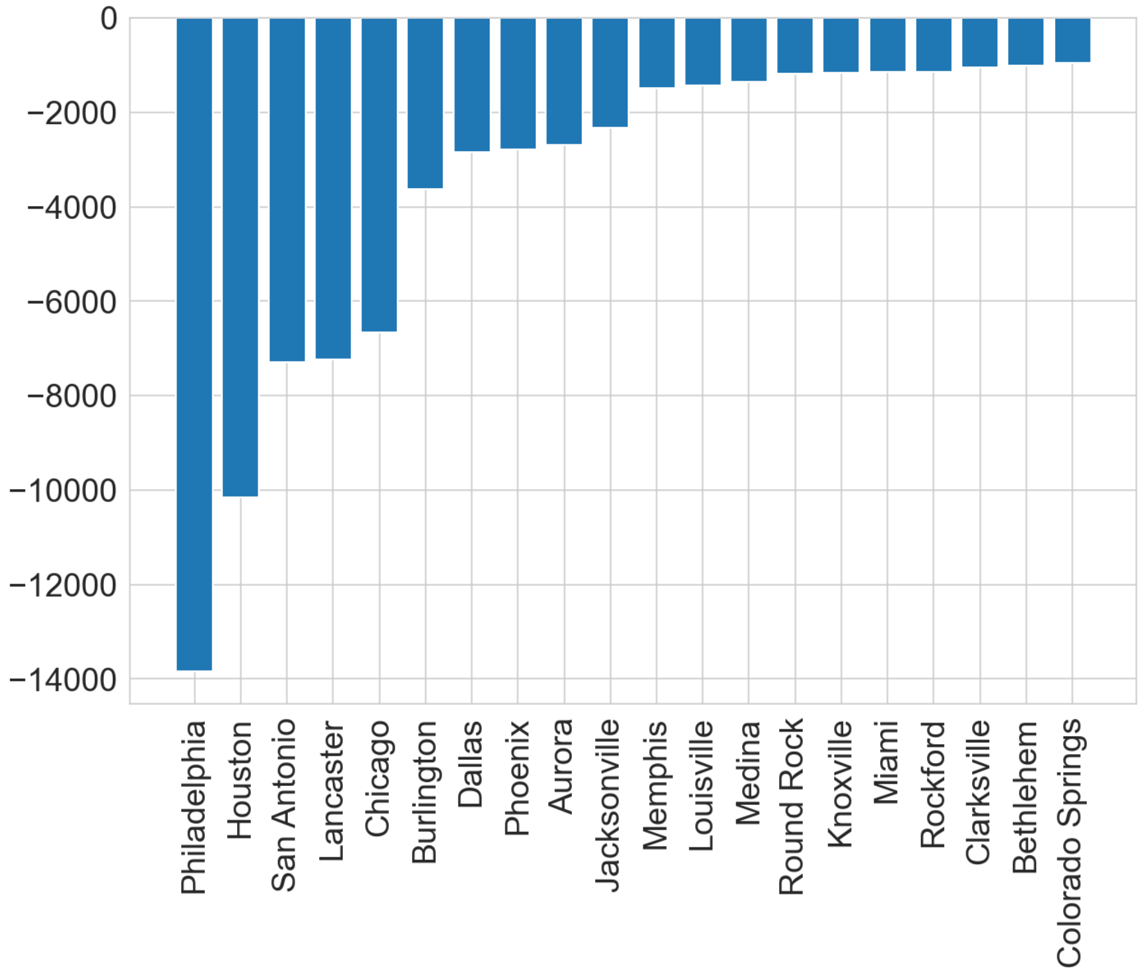
```
#top 20 cities having heighest -ve profit

city = df_city['City']
sales = df_sales_city['Sales']
profit = df_city['Profit']

# Figure Size
fig = plt.figure(figsize =(10, 7))

# Horizontal Bar Plot
plt.bar(city, profit)
plt.xticks(city, rotation='vertical')

# Show Plot
plt.show()
```



In [20]:

```
df_sales_city[['City', 'Sales', 'Profit']].query('Profit < 0')
```

Out[20]:

	City	Sales	Profit
4	Philadelphia	109077.0130	-13837.7674
5	Houston	64504.7604	-10153.5485
6	Chicago	48539.5410	-6654.5688
8	Jacksonville	44713.1830	-2323.8350
16	San Antonio	21843.5280	-7299.0502
17	Burlington	21668.0820	-3622.8772
19	Dallas	20131.9322	-2846.5257

- **Above Data shows that despite having larger number of sales Philadelphia, Houston, Chicago and other Cities are losing money**

Finding sub-category with highest -ve Profit

In [21]:

```
df_store.groupby(['Category', 'Sub-Category']).sum().sort_values(by=['Profit'])
```

Out[21]:

Category	Sub-Category	Postal Code	Sales	Quantity	Discount	Profit
Furniture	**Tables**	18607828	206965.5320	1241	83.35	-17725.4811
	Bookcases	12771539	114879.9963	868	48.14	-3472.5560
Office Supplies	**Supplies**	10633558	46673.5380	647	14.60	-1189.0995
	Fasteners	12506063	3024.2800	914	17.80	949.5182
Technology	**Machines**	6364668	189238.6310	440	35.20	3384.7569
Office Supplies	**Labels**	19552985	12486.3120	1400	25.00	5546.2540
	Art	43329658	27118.7920	3000	59.60	6527.7870
	Envelopes	13325731	16476.4020	906	20.40	6964.1767
Furniture	**Furnishings**	51880430	91705.1640	3563	132.40	13059.1436
Office Supplies	**Appliances**	25250538	107532.1610	1729	77.60	18138.0054
	Storage	46248720	223843.6080	3158	63.20	21278.8264
Furniture	**Chairs**	34936229	328449.1030	2356	105.00	26590.1663
Office Supplies	**Binders**	83626398	203412.7330	5974	567.00	30221.7633
	Paper	76299221	78479.2060	5178	102.60	34053.5693
Technology	**Accessories**	44468434	167380.3180	2976	60.80	41936.6357
	Phones	47897175	330007.0540	3289	137.40	44515.7306
	Copiers	3873477	149528.0300	234	11.00	55617.8249

In [22]:

```
df_category = df_store.groupby(['Sub-Category']).sum().sort_values(by=['Profit']).head(
20)
df_category.reset_index(drop=False, inplace=True)
df_category
```

Out[22]:

	Sub-Category	Postal Code	Sales	Quantity	Discount	Profit
0	Tables	18607828	206965.5320	1241	83.35	-17725.4811
1	Bookcases	12771539	114879.9963	868	48.14	-3472.5560
2	Supplies	10633558	46673.5380	647	14.60	-1189.0995
3	Fasteners	12506063	3024.2800	914	17.80	949.5182
4	Machines	6364668	189238.6310	440	35.20	3384.7569
5	Labels	19552985	12486.3120	1400	25.00	5546.2540
6	Art	43329658	27118.7920	3000	59.60	6527.7870
7	Envelopes	13325731	16476.4020	906	20.40	6964.1767
8	Furnishings	51880430	91705.1640	3563	132.40	13059.1436
9	Appliances	25250538	107532.1610	1729	77.60	18138.0054
10	Storage	46248720	223843.6080	3158	63.20	21278.8264
11	Chairs	34936229	328449.1030	2356	105.00	26590.1663
12	Binders	83626398	203412.7330	5974	567.00	30221.7633
13	Paper	76299221	78479.2060	5178	102.60	34053.5693
14	Accessories	44468434	167380.3180	2976	60.80	41936.6357
15	Phones	47897175	330007.0540	3289	137.40	44515.7306
16	Copiers	3873477	149528.0300	234	11.00	55617.8249

In [23]:

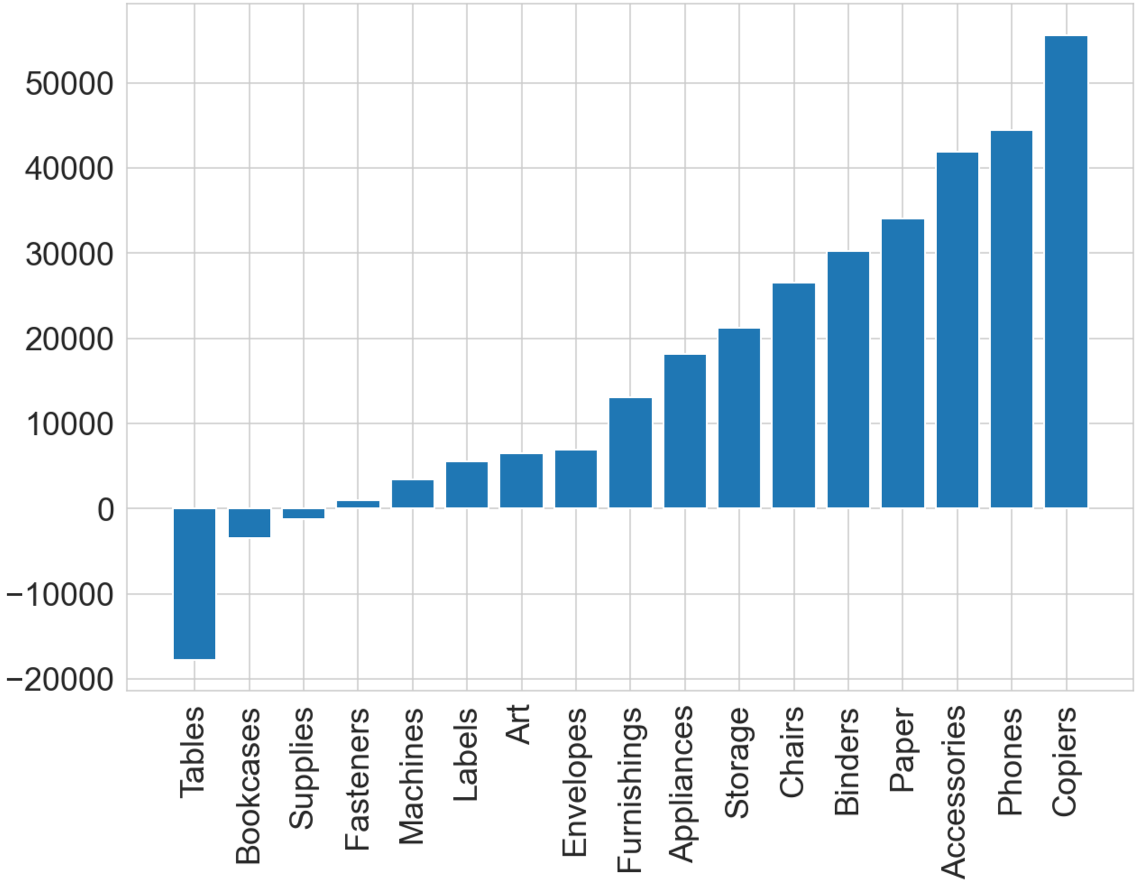
```
#top 20 sub-categories having heighest -ve/+ve profit

sub_category = df_category['Sub-Category']
profit = df_category['Profit']

# Figure Size
fig = plt.figure(figsize =(10, 7))

# Horizontal Bar Plot
plt.bar(sub_category, profit)
plt.xticks(sub_category, rotation='vertical')

# Show Plot
plt.show()
```



Finding Postal Code with highest -ve Profit

In [24]:

```
df_store.groupby(['Postal Code', 'State', 'City']).sum().sort_values(by=['Profit']).hea
d(20)
```

Out[24]:

			Sales	Quantity	Discount	Profit
Postal Code	**State**	**City**				
78207	**Texas**	**San Antonio**	21843.5280	247	22.60	-7299.0502
43130	**Ohio**	**Lancaster**	8202.6250	121	9.70	-7149.6180
27217	**North Carolina**	**Burlington**	12681.2820	46	3.40	-5894.5269
60653	**Illinois**	**Chicago**	16012.5060	358	40.70	-5678.7982
19140	**Pennsylvania**	**Philadelphia**	22752.9140	564	48.60	-5168.3905
77095	**Texas**	**Houston**	11077.4192	381	40.32	-4447.3323
19143	**Pennsylvania**	**Philadelphia**	20641.2880	424	37.40	-3830.7458
19134	**Pennsylvania**	**Philadelphia**	39390.2930	585	52.60	-3745.8552
80027	**Colorado**	**Louisville**	5070.4160	90	8.10	-3406.2095
85023	**Arizona**	**Phoenix**	11000.2570	224	22.30	-2790.8832
32216	**Florida**	**Jacksonville**	39133.3280	248	21.35	-2445.6608
77036	**Texas**	**Houston**	18522.7876	272	22.88	-2411.4332
77041	**Texas**	**Houston**	16556.9592	443	44.60	-2302.4956
43055	**Ohio**	**Newark**	8128.0690	137	9.80	-2292.4127
60505	**Illinois**	**Aurora**	7572.9680	128	13.40	-1894.7196
75217	**Texas**	**Dallas**	9720.6520	200	17.94	-1864.8163
28027	**North Carolina**	**Concord**	5111.8440	44	1.80	-1788.6868
38109	**Tennessee**	**Memphis**	5942.3410	116	8.40	-1479.0400
45503	**Ohio**	**Springfield**	5613.1670	137	12.20	-1420.1620
44256	**Ohio**	**Medina**	2477.7220	38	3.90	-1343.0446

Texas

In [38]:

```
df_texas = df_store.groupby(['State', 'Category', 'Sub-Category']).sum()
df_texas.query('State.str.contains("Texas")').sort_values(by=['Profit'], ascending=False)
```

Out[38]:

State	Category	Sub-Category	Postal Code	Sales	Quantity	Discount	Profit
Texas	Office Supplies	Binders	11765034	9042.6760	626	122.40	-14705.0738
		Appliances	3613797	2407.8140	159	37.60	-6147.2225
	Furniture	Furnishings	6218925	3766.7240	308	48.60	-3312.6786
	Technology	Machines	1003402	19546.2240	47	5.20	-2666.8434
	Furniture	Chairs	4725503	26572.4480	235	18.30	-2515.6490
		Bookcases	2074306	14493.4588	105	8.64	-2391.1377
		Tables	2532267	15760.6610	118	9.90	-2216.6766
	Office Supplies	Supplies	1452046	4516.7600	65	3.80	-837.2795
		Storage	6380516	15723.5840	309	16.60	-763.7054
		Fasteners	1850215	332.4640	108	4.80	80.7357
		Labels	2297999	583.6000	96	6.00	200.4020
		Art	5464328	2369.5280	259	14.20	316.3538
		Envelopes	2299005	2530.6480	105	6.00	848.1760
	Technology	Accessories	6228252	11328.5600	281	16.20	1105.8501
		Copiers	386259	5639.8720	16	1.00	1629.9615
	Office Supplies	Paper	11306185	6983.4560	572	29.40	2422.9703
	Technology	Phones	6149654	28589.5680	315	16.00	3222.4608

Ohio

In [39]:

```
df_ohio = df_store.groupby(['State', 'Category', 'Sub-Category']).sum()
df_ohio.query('State.str.contains("Ohio")').sort_values(by=['Profit'], ascending=False)
```

Out[39]:

			Postal Code	Sales	Quantity	Discount	Profit
State	**Category**	**Sub-Category**					
Ohio	**Technology**	**Machines**	348474	8978.238	39	5.6	-11770.9447
		Phones	2065875	14634.948	179	18.8	-2778.8578
	Furniture	**Tables**	704515	7887.114	49	6.4	-2715.3345
	Office Supplies	**Binders**	3213985	1917.087	292	51.1	-1400.6681
	Furniture	**Bookcases**	353029	2077.705	32	4.0	-1359.0516
		Chairs	1005912	10145.702	79	6.9	-649.3542
	Office Supplies	**Storage**	1620294	7264.440	123	7.4	-276.3364
		Supplies	303693	478.808	22	1.4	-82.9029
		Labels	570152	161.840	43	2.6	55.5222
		Fasteners	610362	204.896	83	2.8	61.1197
		Art	1710946	840.104	140	7.8	103.2376
		Envelopes	525452	562.000	38	2.4	194.6051
	Technology	**Copiers**	175091	3839.934	11	1.6	446.9923
	Office Supplies	**Appliances**	1010194	4807.536	96	4.6	486.2553
	Furniture	**Furnishings**	2015276	4088.624	184	9.2	517.4191
	Office Supplies	**Paper**	2464117	2146.288	194	11.2	744.0522
	Technology	**Accessories**	1882469	8222.872	155	8.6	1452.8701

Pennsylvania

In [42]:

```
df_pennsylvania = df_store.groupby(['State', 'Category', 'Sub-Category']).sum()
df_pennsylvania.query('State.str.contains("Pennsylvania")').sort_values(by=['Profit'], ascending=False)
```

Out[42]:

State	Category	Sub-Category	Postal Code	Sales	Quantity	Discount	Profit
Pennsylvania	**Technology**	**Copiers**	95656	13079.868	22	2.0	1735.9829
		Accessories	859990	7299.280	141	9.0	898.9687
	Office Supplies	**Paper**	1258588	2378.304	236	13.2	813.0179
		Appliances	438988	4663.280	77	4.6	694.2150
		Envelopes	344749	1234.064	67	3.6	417.1003
	Furniture	**Furnishings**	1221628	7347.816	225	12.8	282.2120
	Office Supplies	**Labels**	439844	598.304	106	4.6	201.4208
		Art	761496	1152.160	155	8.0	137.7581
		Fasteners	267868	154.712	73	2.8	29.3222
		Storage	933768	11784.624	169	9.8	-1434.3118
		Supplies	226943	6710.208	34	2.4	-1459.5663
	Furniture	**Chairs**	684284	18724.174	153	10.8	-1993.4180
	Technology	**Machines**	133951	2133.717	20	4.9	-2219.2456
	Furniture	**Tables**	287020	8052.186	65	6.0	-2588.7538
		Bookcases	190676	5230.755	44	5.0	-2896.7601
	Technology	**Phones**	1183027	19702.404	235	24.8	-3606.9276
	Office Supplies	**Binders**	1862089	6266.058	331	68.6	-4570.9750

Conclusions

- **Overall profit is positive but still there are some areas where work could be done to make profit**

    1. We could minimize the sales of table.
    2. Texas is making loses but there are some Sub-Categories where profit has been made like Phones, Paper, Copiers, Accessories, etc. We should focus more on that.
    3. Ohio is also making loses but there are some Sub-Categories where profit has been made like Accessories, Paper, Furnishings, Appliances, etc. We should focus more on that.
    4. Pennsylvania is also making loses but there are some Sub-Categories where profit has been made like Copiers, Accessories, Paper, Envelopes, etc. We should focus more on that. These could be applied to every loss having state or cities
    5. Increase sales more in the east as profit is more.
    6. We should concentrate on the states like 'California' and 'New York' to make more profits.
    7. Discounts and Profit are in negatively Correlated.