In [1]:
```
import pandas as pd
x=pd.read_excel(r"C:\Users\SNIGDHA\Downloads\Adidas US Sales Datasets (1).xlsx")
x
```

Out[1]:

	Unnamed: 0	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5	Unnamed: 6	Unnamed: 7	Unnamed: 8	Unnamed: 9	Unnamed: 10	Unnamed: 11	Unnamed: 12	Unnamed: 13
0	NaN	NaN	Adidas Sales Database	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	NaN	Retailer	Retailer ID	Invoice Date	Region	State	City	Product	Price per Unit	Units Sold	Total Sales	Operating Profit	Operating Margin	Sales Method
4	NaN	Foot Locker	1185732	2020-01-01 00:00:00	Northeast	New York	New York	Men's Street Footwear	50	1200	600000	300000	0.5	In-store
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
9647	NaN	Foot Locker	1185732	2021-01-24 00:00:00	Northeast	New Hampshire	Manchester	Men's Apparel	50	64	3200	896.0	0.28	Outlet
9648	NaN	Foot Locker	1185732	2021-01-24 00:00:00	Northeast	New Hampshire	Manchester	Women's Apparel	41	105	4305	1377.6	0.32	Outlet
9649	NaN	Foot Locker	1185732	2021-02-22 00:00:00	Northeast	New Hampshire	Manchester	Men's Street Footwear	41	184	7544	2791.28	0.37	Outlet
9650	NaN	Foot Locker	1185732	2021-02-22 00:00:00	Northeast	New Hampshire	Manchester	Men's Athletic Footwear	42	70	2940	1234.8	0.42	Outlet
9651	NaN	Foot Locker	1185732	2021-02-22 00:00:00	Northeast	New Hampshire	Manchester	Women's Street Footwear	29	83	2407	649.89	0.27	Outlet

9652 rows × 14 columns

In [2]:
```
file=r"C:\Users\SNIGDHA\Downloads\Adidas US Sales Datasets (1).xlsx"
y=pd.read_excel(file, header=4)
y
```

Out[2]:

	Unnamed: 0	Retailer	Retailer ID	Invoice Date	Region	State	City	Product	Price per Unit	Units Sold	Total Sales	Operating Profit	Operating Margin	Sales Method
0	NaN	Foot Locker	1185732	2020-01-01	Northeast	New York	New York	Men's Street Footwear	50.0	1200	600000.0	300000.00	0.50	In-store
1	NaN	Foot Locker	1185732	2020-01-02	Northeast	New York	New York	Men's Athletic Footwear	50.0	1000	500000.0	150000.00	0.30	In-store
2	NaN	Foot Locker	1185732	2020-01-03	Northeast	New York	New York	Women's Street Footwear	40.0	1000	400000.0	140000.00	0.35	In-store
3	NaN	Foot Locker	1185732	2020-01-04	Northeast	New York	New York	Women's Athletic Footwear	45.0	850	382500.0	133875.00	0.35	In-store
4	NaN	Foot Locker	1185732	2020-01-05	Northeast	New York	New York	Men's Apparel	60.0	900	540000.0	162000.00	0.30	In-store
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
9643	NaN	Foot Locker	1185732	2021-01-24	Northeast	New Hampshire	Manchester	Men's Apparel	50.0	64	3200.0	896.00	0.28	Outlet
9644	NaN	Foot Locker	1185732	2021-01-24	Northeast	New Hampshire	Manchester	Women's Apparel	41.0	105	4305.0	1377.60	0.32	Outlet
9645	NaN	Foot Locker	1185732	2021-02-22	Northeast	New Hampshire	Manchester	Men's Street Footwear	41.0	184	7544.0	2791.28	0.37	Outlet
9646	NaN	Foot Locker	1185732	2021-02-22	Northeast	New Hampshire	Manchester	Men's Athletic Footwear	42.0	70	2940.0	1234.80	0.42	Outlet
9647	NaN	Foot Locker	1185732	2021-02-22	Northeast	New Hampshire	Manchester	Women's Street Footwear	29.0	83	2407.0	649.89	0.27	Outlet

9648 rows × 14 columns

In [3]:
```
y.drop("Unnamed: 0", axis=1)
```

Out[3]:

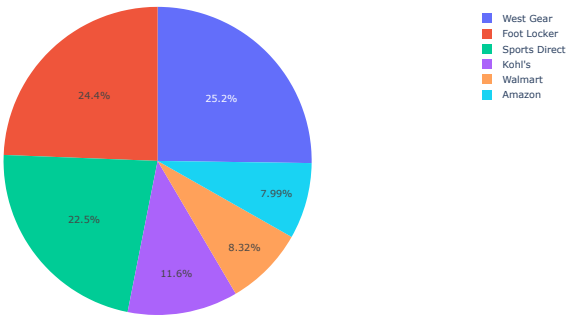
	Retailer	Retailer ID	Invoice Date	Region	State	City	Product	Price per Unit	Units Sold	Total Sales	Operating Profit	Operating Margin	Sales Method
0	Foot Locker	1185732	2020-01-01	Northeast	New York	New York	Men's Street Footwear	50.0	1200	600000.0	300000.00	0.50	In-store
1	Foot Locker	1185732	2020-01-02	Northeast	New York	New York	Men's Athletic Footwear	50.0	1000	500000.0	150000.00	0.30	In-store
2	Foot Locker	1185732	2020-01-03	Northeast	New York	New York	Women's Street Footwear	40.0	1000	400000.0	140000.00	0.35	In-store
3	Foot Locker	1185732	2020-01-04	Northeast	New York	New York	Women's Athletic Footwear	45.0	850	382500.0	133875.00	0.35	In-store
4	Foot Locker	1185732	2020-01-05	Northeast	New York	New York	Men's Apparel	60.0	900	540000.0	162000.00	0.30	In-store
...	...	...	...	...	...	...	...	...	...	...	...	...	...
9643	Foot Locker	1185732	2021-01-24	Northeast	New Hampshire	Manchester	Men's Apparel	50.0	64	3200.0	896.00	0.28	Outlet
9644	Foot Locker	1185732	2021-01-24	Northeast	New Hampshire	Manchester	Women's Apparel	41.0	105	4305.0	1377.60	0.32	Outlet
9645	Foot Locker	1185732	2021-02-22	Northeast	New Hampshire	Manchester	Men's Street Footwear	41.0	184	7544.0	2791.28	0.37	Outlet
9646	Foot Locker	1185732	2021-02-22	Northeast	New Hampshire	Manchester	Men's Athletic Footwear	42.0	70	2940.0	1234.80	0.42	Outlet
9647	Foot Locker	1185732	2021-02-22	Northeast	New Hampshire	Manchester	Women's Street Footwear	29.0	83	2407.0	649.89	0.27	Outlet

9648 rows × 13 columns

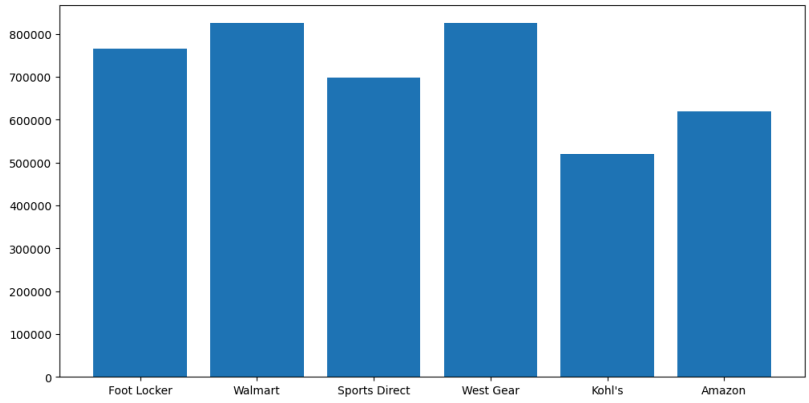
In [4]:
```
y.isnull().sum()
```

Out[4]:
```
Unnamed: 0          9648
Retailer               0
Retailer ID            0
Invoice Date           0
Region                 0
State                  0
City                   0
Product                0
Price per Unit         0
Units Sold             0
Total Sales            0
Operating Profit       0
Operating Margin       0
Sales Method           0
dtype: int64
```

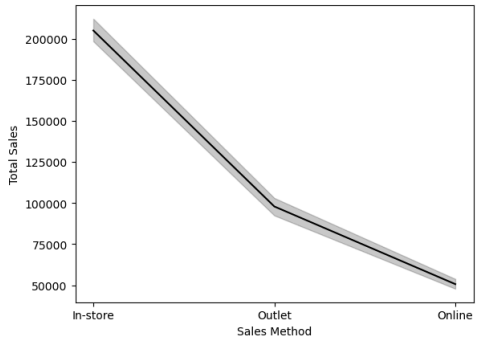
In [24]:
```
# To identify the number of units sold by different retailers and in different regions

import plotly.express as px

graph = px.pie(y, names="Retailer", values="Units Sold", hover_data=["Region"])
graph.show()
```



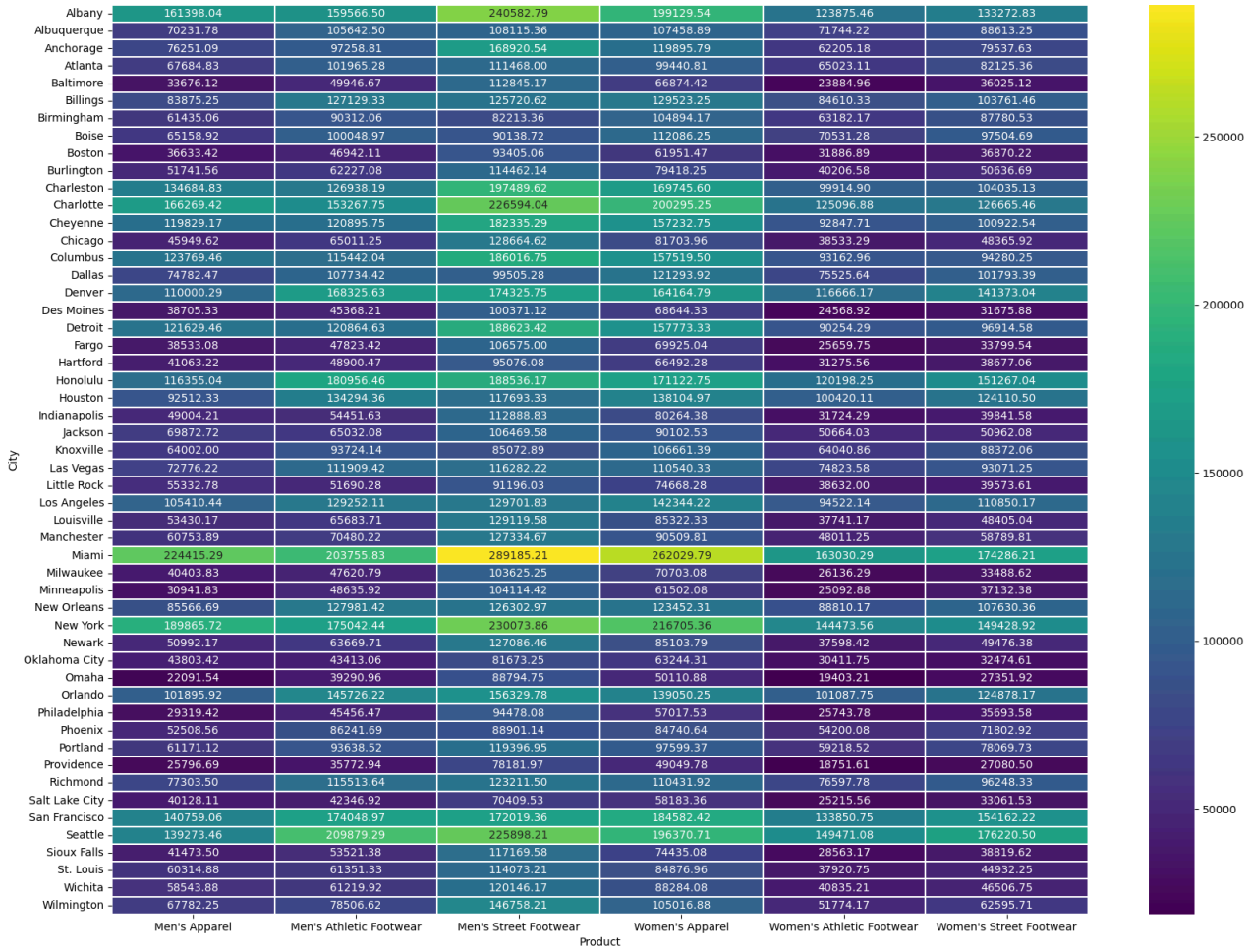
In [13]:
```
# To identify the sales acquired through different retailers

from matplotlib import pyplot as plt
plt.figure(figsize=(12,6))
plt.bar(y["Retailer"], y["Total Sales"])
plt.show()
```



In [14]:
```
# To identify the total sales acquired through different sales method

import seaborn as sns
from matplotlib import pyplot as plt

sns.lineplot(x="Sales Method", y="Total Sales", color="black", data=y)
plt.show()
```

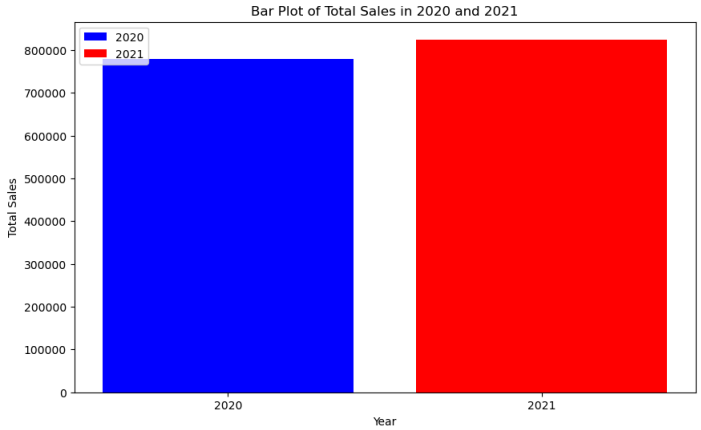


In [30]:
```
# To understand how the total sales of different products is affected by different cities

import seaborn as sns
import pandas as pd

plt.figure(figsize=(20,15))
data_df = pd.DataFrame(y)
heatmap_data = data_df.pivot_table(index='City', columns='Product', values='Total Sales')
sns.heatmap(heatmap_data, annot=True, fmt="0.2f", cmap="viridis", linewidths=0.20)
plt.show()
```

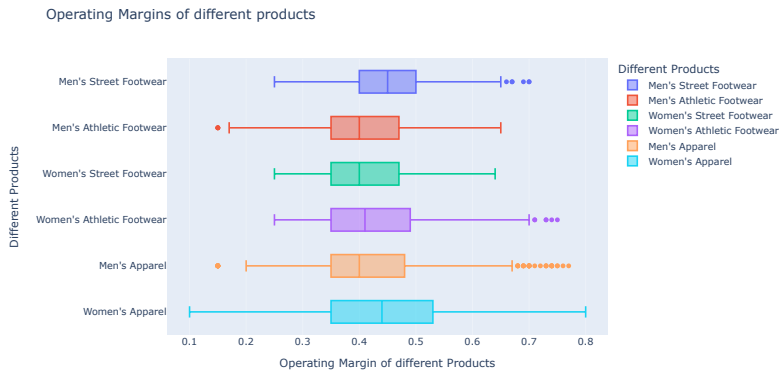


City	Men's Apparel	Men's Athletic Footwear	Men's Street Footwear	Women's Apparel	Women's Athletic Footwear	Women's Street Footwear
Albany	161398.04	159566.50	240582.79	199129.54	123875.46	133272.83
Albuquerque	70231.78	105642.50	108115.36	107458.89	71744.22	88613.25
Anchorage	76251.09	97258.81	168920.54	119895.79	62205.18	79537.63
Atlanta	67684.83	101965.28	111468.00	99440.81	65023.11	82125.36
Baltimore	33676.12	49946.67	112845.17	66874.42	23884.96	36025.12
Billings	83875.25	127129.33	125720.62	129523.25	84610.33	103761.46
Birmingham	61435.06	90312.06	82213.36	104894.17	63182.17	87780.53
Boise	65158.92	100048.97	90138.72	112086.25	70531.28	97504.69
Boston	36633.42	46942.11	93405.06	61951.47	31886.89	36870.22
Burlington	51741.56	62227.08	114462.14	79418.25	40206.58	50636.69
Charleston	134684.83	126938.19	197489.62	169745.60	99914.90	104035.13
Charlotte	166269.42	153267.75	226594.04	200295.25	125096.88	126665.46
Cheyenne	119829.17	120895.75	182335.29	157232.75	92847.71	100922.54
Chicago	45949.62	65011.25	128664.62	81703.96	38533.29	48365.92
Columbus	123769.46	115442.04	186016.75	157519.50	93162.96	94280.25
Dallas	74782.47	107734.42	99505.28	121293.92	75525.64	101793.39
Denver	110000.29	168325.63	174325.75	164164.79	116666.17	141373.04
Des Moines	38705.33	45368.21	100371.12	68644.33	24568.92	31675.88
Detroit	121629.46	120864.63	188623.42	157773.33	90254.29	96914.58
Fargo	38533.08	47823.42	106575.00	69925.04	25659.75	33799.54
Hartford	41063.22	48900.47	95076.08	66492.28	31275.56	38677.06
Honolulu	116355.04	180956.46	188536.17	171122.75	120198.25	151267.04
Houston	92512.33	134294.36	117693.33	138104.97	100420.11	124110.50
Indianapolis	49004.21	54451.63	112888.83	80264.38	31724.29	39841.58
Jackson	69872.72	65032.08	106469.58	90102.53	50664.03	50962.08
Knoxville	64002.00	93724.14	85072.89	106661.39	64040.86	88372.06
Las Vegas	72776.22	111909.42	116282.22	110540.33	74823.58	93071.25
Little Rock	55332.78	51690.28	91196.03	74668.28	38632.00	39573.61
Los Angeles	105410.44	129252.11	129701.83	142344.22	94522.14	110850.17
Louisville	53430.17	65683.71	129119.58	85322.33	37741.17	48405.04
Manchester	60753.89	70480.22	127334.67	90509.81	48011.25	58789.81
Miami	224415.29	203755.83	289185.21	262029.79	163030.29	174286.21
Milwaukee	40403.83	47620.79	103625.25	70703.08	26136.29	33488.62
Minneapolis	30941.83	48635.92	104114.42	61502.08	25092.88	37132.38
New Orleans	85566.69	127981.42	126302.97	123452.31	88810.17	107630.36
New York	189865.72	175042.44	230073.86	216705.36	144473.56	149428.92
Newark	50992.17	63669.71	127086.46	85103.79	37598.42	49476.38
Oklahoma City	43803.42	43413.06	81673.25	63244.31	30411.75	32474.61
Omaha	22091.54	39290.96	88794.75	50110.88	19403.21	27351.92
Orlando	101895.92	145726.22	156329.78	139050.25	101087.75	124878.17
Philadelphia	29319.42	45456.47	94478.08	57017.53	25743.78	35693.58
Phoenix	52508.56	86241.69	88901.14	84740.64	54200.08	71802.92
Portland	61171.12	93638.52	119396.95	97599.37	59218.52	78069.73
Providence	25796.69	35772.94	78181.97	49049.78	18751.61	27080.50
Richmond	77303.50	115513.64	123211.50	110431.92	76597.78	96248.33
Salt Lake City	40128.11	42346.92	70409.53	58183.36	25215.56	33061.53
San Francisco	140759.06	174048.97	172019.36	184582.42	133850.75	154162.22
Seattle	139273.46	209879.29	225898.21	196370.71	149471.08	176220.50
Sioux Falls	41473.50	53521.38	117169.58	74435.08	28563.17	38819.62
St. Louis	60314.88	61351.33	114073.21	84876.96	37920.75	44932.25
Wichita	58543.88	61219.92	120146.17	88284.08	40835.21	46506.75
Wilmington	67782.25	78506.62	146758.21	105016.88	51774.17	62595.71

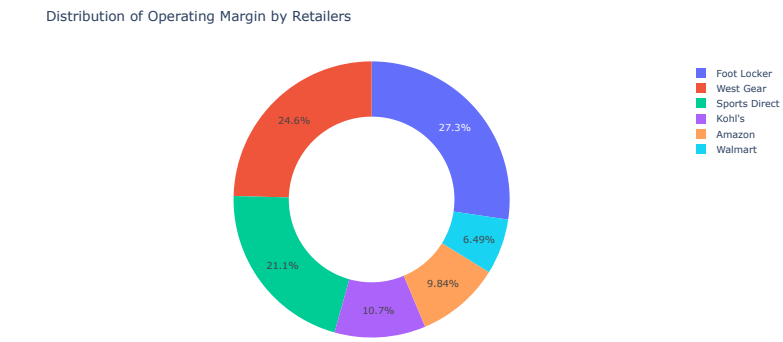
In [74]:
```
# To identify the total sales in 2020 and 2021

import pandas as pd
import matplotlib.pyplot as plt

y['Invoice Date'] = pd.to_datetime(y['Invoice Date'])
year = y['Invoice Date'].dt.year

plt.figure(figsize=(10, 6))

plt.bar(x=year[y['Invoice Date'].dt.year == 2020], height=y['Total Sales'][y['Invoice Date'].dt.year == 2020], color="blue", label='2020')

plt.bar(x=year[y['Invoice Date'].dt.year == 2021], height=y['Total Sales'][y['Invoice Date'].dt.year == 2021], color="red", label='2021')

plt.xlabel('Year')
plt.ylabel('Total Sales')
plt.title('Bar Plot of Total Sales in 2020 and 2021')
plt.legend()

plt.xticks([2020, 2021])

plt.show()
```



In [17]:
```
# To measure the operating margin of different products

import plotly.express as px
box=px.box(y, x="Operating Margin", y="Product", color="Product",
                   hover_data=["Units Sold"],
                   title="Operating Margins of different products",
                   labels={"Product":"Different Products", "Operating Margin":"Operating Margin of different Products"})

box.show()
```



In [23]:
```
# To measure the distribution of operating margin by different retailers

import plotly.express as px

donut_chart = px.pie(y, names="Retailer", title="Distribution of Operating Margin by Retailers",
                     hole=0.6,
                     labels={"Retailer":"Retailer", "Operating Margin":"Operating Margin of different retailers"},
                     hover_data=["Units Sold"])
donut_chart.show()
```



In [ ]: