

# ADIDAS DATASET ANALYSIS

In [1468]:

```
import pandas as pd
```

In [1469]:

```
df2=pd.read_csv('C:\Users\Leechita G\Desktop\Adidas\Adidas US Sales Datasets.csv') #imp
```

## OVERALL DATA

In [1470]:

```
df2.head()
```

Out[1470]:

	Retailer	Retailer ID	Invoice Date	Region	State	City	Product	Price per Unit	Units Sold	Total Sales	Open
0	Foot Locker	1185732	01-01-2020	Northeast	New York	New York	Men's Street Footwear	\$50.00	1200	\$600,000	\$30
1	Foot Locker	1185732	02-01-2020	Northeast	New York	New York	Men's Athletic Footwear	\$50.00	1000	\$500,000	\$15
2	Foot Locker	1185732	03-01-2020	Northeast	New York	New York	Women's Street Footwear	\$40.00	1000	\$400,000	\$14
3	Foot Locker	1185732	04-01-2020	Northeast	New York	New York	Women's Athletic Footwear	\$45.00	850	\$382,500	\$13
4	Foot Locker	1185732	05-01-2020	Northeast	New York	New York	Men's Apparel	\$60.00	900	\$540,000	\$16

In [1471]:

```
df2['Price per Unit'] = df2['Price per Unit'].str.replace('$', '')
df2['Price per Unit'] = df2['Price per Unit'].str.replace('.', '')
df2['Price per Unit'] = df2['Price per Unit'].map(lambda x: str(x)[-3])
df2['Price per Unit'] = df2['Price per Unit'].astype(int)
```

C:\Users\LEECHI~1\AppData\Local\Temp\ipykernel\_35736\1871374501.py:1: FutureWarning: The default value of regex will change from True to False in a future version. In addition, single character regular expressions will \*not\* be treated as literal strings when regex=True.

```
df2['Price per Unit'] = df2['Price per Unit'].str.replace('$', '')
```

C:\Users\LEECHI~1\AppData\Local\Temp\ipykernel\_35736\1871374501.py:2: FutureWarning: The default value of regex will change from True to False in a future version. In addition, single character regular expressions will \*not\* be treated as literal strings when regex=True.

```
df2['Price per Unit'] = df2['Price per Unit'].str.replace('.', '')
```

In [1472]:

```
df2.head()
```

Out[1472]:

	Retailer	Retailer ID	Invoice Date	Region	State	City	Product	Price per Unit	Units Sold	Total Sales	Operating Profit
0	Foot Locker	1185732	01-01-2020	Northeast	New York	New York	Men's Street Footwear	50	1200	\$600,000	\$300
1	Foot Locker	1185732	02-01-2020	Northeast	New York	New York	Men's Athletic Footwear	50	1000	\$500,000	\$150
2	Foot Locker	1185732	03-01-2020	Northeast	New York	New York	Women's Street Footwear	40	1000	\$400,000	\$140
3	Foot Locker	1185732	04-01-2020	Northeast	New York	New York	Women's Athletic Footwear	45	850	\$382,500	\$133
4	Foot Locker	1185732	05-01-2020	Northeast	New York	New York	Men's Apparel	60	900	\$540,000	\$162



In [1473]:

```
df2['Total Sales'] = df2['Total Sales'].str.replace('$', '')
df2['Total Sales'] = df2['Total Sales'].str.replace(',', '')
df2['Total Sales'] = df2['Total Sales'].str.replace('.', '')
```

C:\Users\LEECHI~1\AppData\Local\Temp\ipykernel\_35736\2592839297.py:1: FutureWarning: The default value of regex will change from True to False in a future version. In addition, single character regular expressions will \*not\* be treated as literal strings when regex=True.

```
df2['Total Sales'] = df2['Total Sales'].str.replace('$', '')
```

C:\Users\LEECHI~1\AppData\Local\Temp\ipykernel\_35736\2592839297.py:3: FutureWarning: The default value of regex will change from True to False in a future version. In addition, single character regular expressions will \*not\* be treated as literal strings when regex=True.

```
df2['Total Sales'] = df2['Total Sales'].str.replace('.', '')
```

In [1474]:

```
#object to int conversion
```

```
df2['Total Sales'] = df2['Total Sales'].astype(int)
```

In [1475]:

```
df2.head()
```

Out[1475]:

	Retailer	Retailer ID	Invoice Date	Region	State	City	Product	Price per Unit	Units Sold	Total Sales	Operati Pro
0	Foot Locker	1185732	01-01-2020	Northeast	New York	New York	Men's Street Footwear	50	1200	600000	\$300,0
1	Foot Locker	1185732	02-01-2020	Northeast	New York	New York	Men's Athletic Footwear	50	1000	500000	\$150,0
2	Foot Locker	1185732	03-01-2020	Northeast	New York	New York	Women's Street Footwear	40	1000	400000	\$140,0
3	Foot Locker	1185732	04-01-2020	Northeast	New York	New York	Women's Athletic Footwear	45	850	382500	\$133,8
4	Foot Locker	1185732	05-01-2020	Northeast	New York	New York	Men's Apparel	60	900	540000	\$162,0



In [1476]:

```
df2['Operating Profit'] = df2['Operating Profit'].str.replace('$', '')
df2['Operating Profit'] = df2['Operating Profit'].str.replace(',', '')
df2['Operating Profit'] = df2['Operating Profit'].astype(int)
```

C:\Users\LEECHI~1\AppData\Local\Temp\ipykernel\_35736\3563407418.py:1: FutureWarning: The default value of regex will change from True to False in a future version. In addition, single character regular expressions will \*not\* be treated as literal strings when regex=True.

```
df2['Operating Profit'] = df2['Operating Profit'].str.replace('$', '')
```

In [1477]:

```
df2.head()
```

Out[1477]:

	Retailer	Retailer ID	Invoice Date	Region	State	City	Product	Price per Unit	Units Sold	Total Sales	Operating Profit
0	Foot Locker	1185732	01-01-2020	Northeast	New York	New York	Men's Street Footwear	50	1200	600000	3000
1	Foot Locker	1185732	02-01-2020	Northeast	New York	New York	Men's Athletic Footwear	50	1000	500000	1500
2	Foot Locker	1185732	03-01-2020	Northeast	New York	New York	Women's Street Footwear	40	1000	400000	1400
3	Foot Locker	1185732	04-01-2020	Northeast	New York	New York	Women's Athletic Footwear	45	850	382500	1338
4	Foot Locker	1185732	05-01-2020	Northeast	New York	New York	Men's Apparel	60	900	540000	1620



## NO OF DISTINCT COLUMNS

In [1478]:

```
#finding all unique columns
```

In [1479]:

```
uniqueval=df2.nunique()  
print(uniqueval)  
print(type(uniqueval))
```

```
Retailer          6  
Retailer ID       4  
Invoice Date      724  
Region            5  
State             50  
City              52  
Product           6  
Price per Unit    94  
Units Sold        361  
Total Sales       3138  
Operating Profit  4187  
Operating Margin   66  
Sales Method       3  
dtype: int64  
<class 'pandas.core.series.Series'>
```

## PRODUCT,REGION AND UNITS SOLD

In [1480]:

```
#group product,region and sum of units sold in each of them
ser1 = df2.groupby(by=['Product','Region'])['Units Sold'].sum()
print(ser1)
print(type(ser1))
```

Product	Region	
Men's Apparel	Midwest	45304
	Northeast	62031
	South	60641
	Southeast	54385
	West	84322
Men's Athletic Footwear	Midwest	65120
	Northeast	81474
	South	90079
	Southeast	71129
	West	127724
Men's Street Footwear	Midwest	109861
	Northeast	134252
	South	106545
	Southeast	91867
	West	150795
Women's Apparel	Midwest	69435
	Northeast	90048
	South	88740
	Southeast	68839
	West	116765
Women's Athletic Footwear	Midwest	44808
	Northeast	59464
	South	63998
	Southeast	55292
	West	93674
Women's Street Footwear	Midwest	56809
	Northeast	74010
	South	82257
	Southeast	65488
	West	113705

Name: Units Sold, dtype: int64  
<class 'pandas.core.series.Series'>

## PRODUCT,REGION WITH HIGHEST UNITS SOLD

In [1481]:

```
#converting series to Data Frame
ser1_df = ser1.reset_index()
print(ser1_df)
print(type(ser1_df))
```

	Product	Region	Units Sold
0	Men's Apparel	Midwest	45304
1	Men's Apparel	Northeast	62031
2	Men's Apparel	South	60641
3	Men's Apparel	Southeast	54385
4	Men's Apparel	West	84322
5	Men's Athletic Footwear	Midwest	65120
6	Men's Athletic Footwear	Northeast	81474
7	Men's Athletic Footwear	South	90079
8	Men's Athletic Footwear	Southeast	71129
9	Men's Athletic Footwear	West	127724
10	Men's Street Footwear	Midwest	109861
11	Men's Street Footwear	Northeast	134252
12	Men's Street Footwear	South	106545
13	Men's Street Footwear	Southeast	91867
14	Men's Street Footwear	West	150795
15	Women's Apparel	Midwest	69435
16	Women's Apparel	Northeast	90048
17	Women's Apparel	South	88740
18	Women's Apparel	Southeast	68839
19	Women's Apparel	West	116765
20	Women's Athletic Footwear	Midwest	44808
21	Women's Athletic Footwear	Northeast	59464
22	Women's Athletic Footwear	South	63998
23	Women's Athletic Footwear	Southeast	55292
24	Women's Athletic Footwear	West	93674
25	Women's Street Footwear	Midwest	56809
26	Women's Street Footwear	Northeast	74010
27	Women's Street Footwear	South	82257
28	Women's Street Footwear	Southeast	65488
29	Women's Street Footwear	West	113705

<class 'pandas.core.frame.DataFrame'>

In [1482]:

```
#max of units sold
max_units=ser1_df['Units Sold'].max()
print(max_units)
```

150795

In [1483]:

```
#Product and region with max of units sold
print(ser1_df.loc[ser1_df['Units Sold'] == ser1_df['Units Sold'].max()])
```

	Product	Region	Units Sold
14	Men's Street Footwear	West	150795

## PRODUCT AND PRICE/UNIT

In [1484]:

```
#group product and price/unit
ser2 = df2.groupby(by=['Product'])['Price per Unit'].unique()
print(ser2)
print(type(ser2))
```

```
Product
Men's Apparel          [60, 65, 40, 45, 55, 75, 50, 35, 70, 85, 80,
9...
Men's Athletic Footwear [50, 55, 35, 40, 45, 60, 65, 70, 30, 75, 80,
8...
Men's Street Footwear   [50, 60, 65, 25, 30, 40, 45, 55, 35, 20, 70,
7...
Women's Apparel         [50, 65, 70, 35, 40, 55, 60, 80, 45, 30, 75,
8...
Women's Athletic Footwear [45, 50, 55, 35, 40, 60, 65, 30, 70, 80, 75,
2...
Women's Street Footwear  [40, 50, 55, 35, 60, 65, 45, 30, 25, 70, 75,
8...
Name: Price per Unit, dtype: object
<class 'pandas.core.series.Series'>
```

In [1485]:

```
#converting series to Data Frame
ser2_df = ser2.reset_index()
print(ser2_df)
print(type(ser2_df))
```

```
      Product \
0      Men's Apparel
1  Men's Athletic Footwear
2      Men's Street Footwear
3      Women's Apparel
4  Women's Athletic Footwear
5      Women's Street Footwear

      Price per Unit
0  [60, 65, 40, 45, 55, 75, 50, 35, 70, 85, 80, 9...
1  [50, 55, 35, 40, 45, 60, 65, 70, 30, 75, 80, 8...
2  [50, 60, 65, 25, 30, 40, 45, 55, 35, 20, 70, 7...
3  [50, 65, 70, 35, 40, 55, 60, 80, 45, 30, 75, 8...
4  [45, 50, 55, 35, 40, 60, 65, 30, 70, 80, 75, 2...
5  [40, 50, 55, 35, 60, 65, 45, 30, 25, 70, 75, 8...
<class 'pandas.core.frame.DataFrame'>
```



In [1486]:

```
#max of price/unit
x=ser2_df['max'] = [max(x) for x in ser2_df['Price per Unit']]
print(ser2_df['max'])
print(max(x))
```

```
0    95
1    95
2    95
3   110
4    90
5    95
Name: max, dtype: int32
110
```

## PRODUCT WITH MAX PRICE/UNIT

In [1487]:

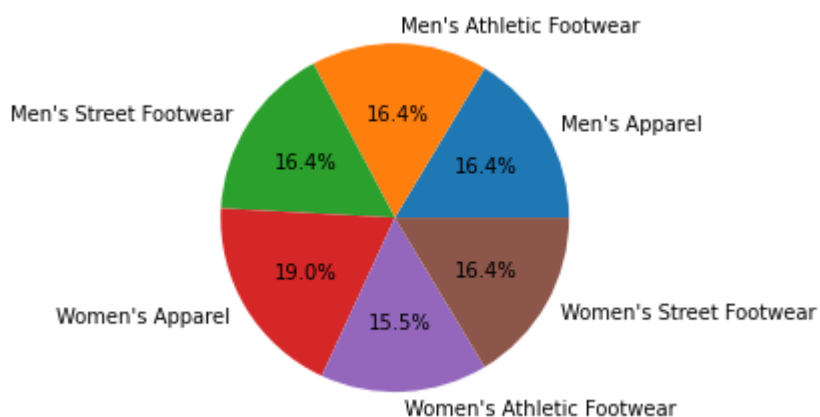
```
#costly product
ser2_df.iloc[[3]]
```

Out[1487]:

	Product	Price per Unit	max
3	Women's Apparel	[50, 65, 70, 35, 40, 55, 60, 80, 45, 30, 75, 8...	110

In [1488]:

```
import matplotlib.pyplot as plt
plt.pie(ser2_df['max'], labels = ser3_df["Product"],autopct='%0.1f%%')
plt.show()
```



## PRODUCT AND TOTAL SALES

In [1489]:

```
#group product and sum of total sales
ser3 = df2.groupby(by=['Product'])['Total Sales'].sum()
print(ser3)
print(type(ser3))
```

```
Product
Men's Apparel      123728632
Men's Athletic Footwear  153673680
Men's Street Footwear  208826244
Women's Apparel    179038860
Women's Athletic Footwear  106631896
Women's Street Footwear  128002813
Name: Total Sales, dtype: int32
<class 'pandas.core.series.Series'>
```

In [1490]:

```
#converting series to Data Frame
ser3_df = ser3.reset_index()
print(ser3_df)
print(type(ser3_df))
```

```
      Product  Total Sales
0  Men's Apparel  123728632
1  Men's Athletic Footwear  153673680
2  Men's Street Footwear  208826244
3  Women's Apparel  179038860
4  Women's Athletic Footwear  106631896
5  Women's Street Footwear  128002813
<class 'pandas.core.frame.DataFrame'>
```

In [1491]:

```
ser3_df.sort_values(by="Total Sales",ascending=True)
```

Out[1491]:

	Product	Total Sales
4	Women's Athletic Footwear	106631896
0	Men's Apparel	123728632
5	Women's Street Footwear	128002813
1	Men's Athletic Footwear	153673680
3	Women's Apparel	179038860
2	Men's Street Footwear	208826244

## PRODUCT WITH MAXIMUM TOTAL SALES

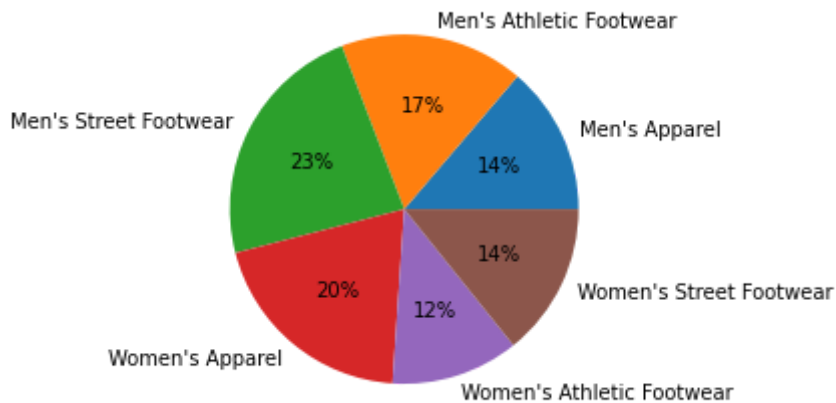
In [1492]:

```
#Product with max Total Sales #no flaws no issues,
print(ser3_df.loc[ser3_df['Total Sales'] == ser3_df['Total Sales'].max()])
```

```
Product Total Sales
2 Men's Street Footwear 208826244
```

In [1493]:

```
plt.pie(ser3_df["Total Sales"], labels = ser3_df["Product"], autopct='%1.0f%%')
plt.show()
```



## PRODUCT AND OPERATING PROFIT

In [1494]:

```
#group product and operating profit
ser4 = df2.groupby(by=['Product'])['Operating Profit'].sum()
print(ser4)
print(type(ser4))
```

```
Product
Men's Apparel      44763099
Men's Athletic Footwear  51846964
Men's Street Footwear  82802323
Women's Apparel    68650996
Women's Athletic Footwear  38975843
Women's Street Footwear  45095897
Name: Operating Profit, dtype: int32
<class 'pandas.core.series.Series'>
```

In [1495]:

```
#converting series to Data Frame
ser4_df = ser4.reset_index()
print(ser4_df)
print(type(ser4_df))
```

```

      Product  Operating Profit
0    Men's Apparel      44763099
1  Men's Athletic Footwear      51846964
2    Men's Street Footwear      82802323
3    Women's Apparel      68650996
4  Women's Athletic Footwear      38975843
5    Women's Street Footwear      45095897
<class 'pandas.core.frame.DataFrame'>
```

In [1496]:

```
ser4_df.sort_values(by="Operating Profit",ascending=True)
```

Out[1496]:

	Product	Operating Profit
4	Women's Athletic Footwear	38975843
0	Men's Apparel	44763099
5	Women's Street Footwear	45095897
1	Men's Athletic Footwear	51846964
3	Women's Apparel	68650996
2	Men's Street Footwear	82802323

## PRODUCT WITH MAXIMUM OPERATING PROFIT

In [1497]:

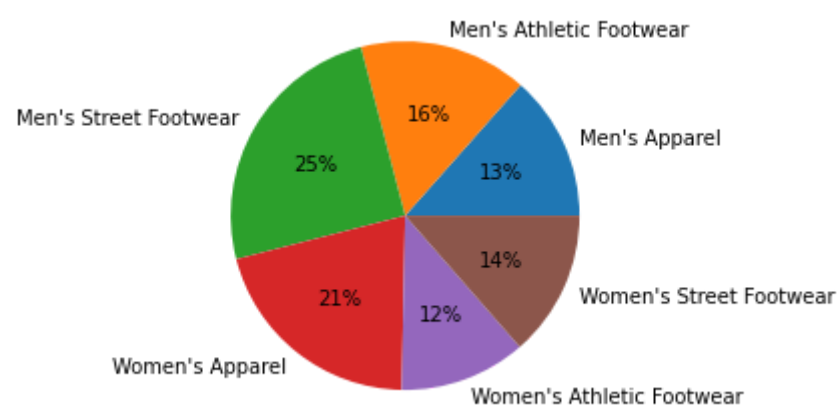
```
#Product with max Operating Profit                                #less price/unit, but more sales
print(ser4_df.loc[ser4_df['Operating Profit'] == ser4_df['Operating Profit'].max()])
```

```

      Product  Operating Profit
2  Men's Street Footwear      82802323
```

In [1498]:

```
plt.pie(ser4_df["Operating Profit"], labels = ser3_df["Product"],autopct='%1.0f%%')
plt.show()
```



# RETAILER,PRODUCT AND REGION WITH MAXIMUM TOTAL SALES

In [1499]:

```
#group retailer,region,product and sum of total sales
ser5 = df2.groupby(by=['Retailer','Region','Product'])['Total Sales'].sum()
print(ser5)
print(type(ser5))
```

Retailer	Region	Product	
Amazon	Midwest	Men's Apparel	2567061
		Men's Athletic Footwear	2770609
		Men's Street Footwear	4464402
		Women's Apparel	3268924
		Women's Athletic Footwear	1885400
...			
West Gear	West	Men's Athletic Footwear	25622089
		Men's Street Footwear	27016671
		Women's Apparel	26546128
		Women's Athletic Footwear	17876904
		Women's Street Footwear	21720603
Name: Total Sales, Length: 166, dtype: int32			
<class 'pandas.core.series.Series'>			

In [1500]:

```
#converting series to Data Frame
ser5_df = ser5.reset_index()
print(ser5_df)
print(type(ser5_df))
```

	Retailer	Region	Product	Total Sales
0	Amazon	Midwest	Men's Apparel	2567061
1	Amazon	Midwest	Men's Athletic Footwear	2770609
2	Amazon	Midwest	Men's Street Footwear	4464402
3	Amazon	Midwest	Women's Apparel	3268924
4	Amazon	Midwest	Women's Athletic Footwear	1885400
..	...	...	...	...
161	West Gear	West	Men's Athletic Footwear	25622089
162	West Gear	West	Men's Street Footwear	27016671
163	West Gear	West	Women's Apparel	26546128
164	West Gear	West	Women's Athletic Footwear	17876904
165	West Gear	West	Women's Street Footwear	21720603

[166 rows x 4 columns]  
<class 'pandas.core.frame.DataFrame'>

Type *Markdown* and LaTeX:  $\alpha^2$

In [1501]:

```
#Product with max Total Sales
print(ser5_df.loc[ser5_df['Total Sales'] == ser5_df['Total Sales'].max()])
```

	Retailer	Region	Product	Total Sales
162	West Gear	West	Men's Street Footwear	27016671

In [1502]:

```
#west region mostly covered by west gear with men's street footwear
```

## RETAILER AND OPERATING PROFIT

In [1503]:

```
#group retailer and sum of operating profit
ser6 = df2.groupby(by=['Retailer'])['Operating Profit'].sum()
print(ser6)
print(type(ser6))
```

Retailer	
Amazon	28818533
Foot Locker	80722234
Kohl's	36811302
Sports Direct	74333022
Walmart	25782063
West Gear	85667968

Name: Operating Profit, dtype: int32  
<class 'pandas.core.series.Series'>

In [1504]:

```
#converting series to Data Frame
ser6_df = ser6.reset_index()
print(ser6_df)
print(type(ser6_df))
```

```
      Retailer  Operating Profit
0      Amazon      28818533
1    Foot Locker      80722234
2      Kohl's      36811302
3 Sports Direct      74333022
4      Walmart      25782063
5    West Gear      85667968
<class 'pandas.core.frame.DataFrame'>
```

In [1505]:

```
ser6_df.sort_values(by="Operating Profit",ascending=True)
```

Out[1505]:

	Retailer	Operating Profit
4	Walmart	25782063
0	Amazon	28818533
2	Kohl's	36811302
3	Sports Direct	74333022
1	Foot Locker	80722234
5	West Gear	85667968

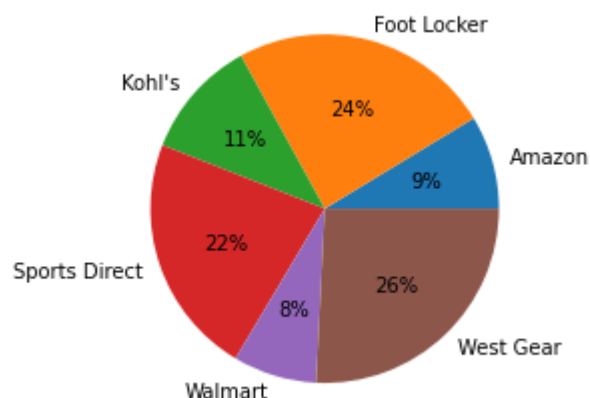
In [1506]:

```
#Retailer with max Operating Profit
print(ser6_df.loc[ser6_df['Operating Profit'] == ser6_df['Operating Profit'].max()])
```

```
      Retailer  Operating Profit
5    West Gear      85667968
```

In [1507]:

```
plt.pie(ser6_df["Operating Profit"], labels = ser6_df["Retailer"], autopct='%1.0f%%') #we
plt.show()
```



## RETAILER AND UNITS SOLD

In [1508]:

```
#group retailer and sum of units sold
ser7 = df2.groupby(by=['Retailer'])['Units Sold'].sum()
print(ser7)
print(type(ser7))
```

```
Retailer
Amazon          197990
Foot Locker     604369
Kohl's          287375
Sports Direct   557640
Walmart        206225
West Gear       625262
Name: Units Sold, dtype: int64
<class 'pandas.core.series.Series'>
```

In [1509]:

```
#converting series to Data Frame
ser7_df = ser7.reset_index()
print(ser7_df)
print(type(ser7_df))
```

```
   Retailer  Units Sold
0    Amazon     197990
1  Foot Locker    604369
2    Kohl's     287375
3  Sports Direct  557640
4    Walmart     206225
5   West Gear    625262
<class 'pandas.core.frame.DataFrame'>
```



In [1510]:

```
ser7_df.sort_values(by="Units Sold",ascending=True)
```

Out[1510]:

	Retailer	Units Sold
0	Amazon	197990
4	Walmart	206225
2	Kohl's	287375
3	Sports Direct	557640
1	Foot Locker	604369
5	West Gear	625262

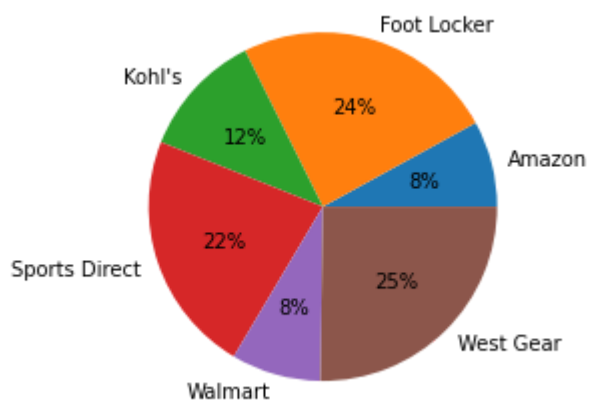
In [1511]:

```
#Retailer with max Units Sold  
print(ser7_df.loc[ser7_df['Units Sold'] == ser7_df['Units Sold'].max()])
```

	Retailer	Units Sold
5	West Gear	625262

In [1512]:

```
plt.pie(ser7_df["Units Sold"], labels = ser7_df["Retailer"],autopct='%1.0f%%')  
plt.show()
```



In [1513]:

```
#west gear has sold more units, thereby more profit
```

## REGION AND TOTAL SALES

In [1514]:

```
ser8 = df2.groupby(by=['Region'])['Total Sales'].sum()  
print(ser8)  
print(type(ser8))
```

```
Region  
Midwest      135800459  
Northeast    186324067  
South        144663181  
Southeast    163171236  
West         269943182  
Name: Total Sales, dtype: int32  
<class 'pandas.core.series.Series'>
```

In [1515]:

```
#converting series to Data Frame  
ser8_df = ser8.reset_index()  
print(ser8_df)  
print(type(ser8_df))
```

```
      Region  Total Sales  
0  Midwest    135800459  
1  Northeast   186324067  
2    South    144663181  
3  Southeast   163171236  
4    West     269943182  
<class 'pandas.core.frame.DataFrame'>
```

In [1516]:

```
ser8_df.sort_values(by="Total Sales",ascending=True)
```

Out[1516]:

	Region	Total Sales
0	Midwest	135800459
2	South	144663181
3	Southeast	163171236
1	Northeast	186324067
4	West	269943182

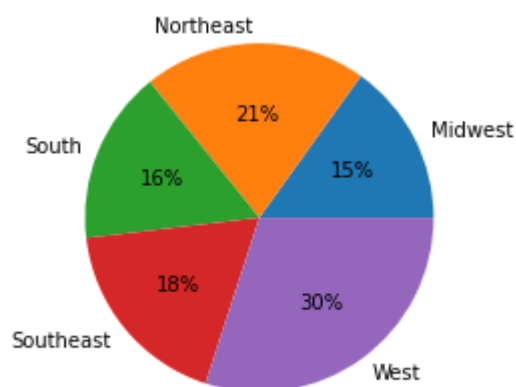
In [1517]:

```
#Region with max Total Sales  
print(ser8_df.loc[ser8_df['Total Sales'] == ser8_df['Total Sales'].max()])
```

```
      Region  Total Sales  
4    West     269943182
```

In [1518]:

```
plt.pie(ser8_df["Total Sales"], labels = ser8_df["Region"], autopct='%1.0f%%')  
plt.show()
```



In [1519]:

```
#more sales in west region
```

## SALES METHOD AND OPERATING PROFIT

In [1520]:

```
#groupsales method and sum of operating profit  
ser9 = df2.groupby(by=['Sales Method'])['Operating Profit'].sum()  
print(ser9)  
print(type(ser9))
```

```
Sales Method  
In-store    127591382  
Online      96555337  
Outlet      107988403  
Name: Operating Profit, dtype: int32  
<class 'pandas.core.series.Series'>
```

In [1521]:

```
#converting series to Data Frame  
ser9_df = ser9.reset_index()  
print(ser9_df)  
print(type(ser9_df))
```

```
   Sales Method  Operating Profit  
0      In-store    127591382  
1       Online     96555337  
2       Outlet    107988403  
<class 'pandas.core.frame.DataFrame'>
```

In [1522]:

```
ser9_df.sort_values(by="Operating Profit",ascending=True)
```

Out[1522]:

	Sales Method	Operating Profit
1	Online	96555337
2	Outlet	107988403
0	In-store	127591382

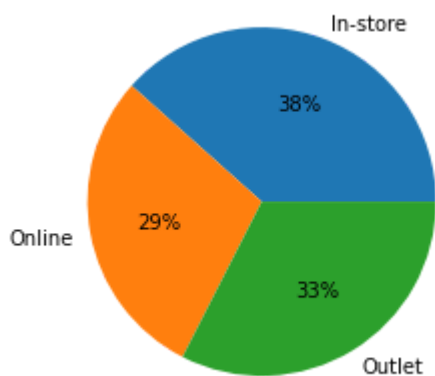
In [1523]:

```
#Region with max Total Sales  
print(ser9_df.loc[ser9_df['Operating Profit'] == ser9_df['Operating Profit'].max()])
```

```
   Sales Method  Operating Profit  
0      In-store      127591382
```

In [1524]:

```
plt.pie(ser9_df["Operating Profit"], labels = ser9_df["Sales Method"],autopct='%1.0f%%')  
plt.show()
```



In [1525]:

```
#profitable business via Instore purchase
```

## SALES METHOD AND UNITS SOLD

In [1526]:

```
#group sales method and sum of units sold
ser10 = df2.groupby(by=['Sales Method'])['Units Sold'].sum()
print(ser10)
print(type(ser10))
```

```
Sales Method
In-store    689990
Online      939093
Outlet      849778
Name: Units Sold, dtype: int64
<class 'pandas.core.series.Series'>
```

In [1527]:

```
#converting series to Data Frame
ser10_df = ser10.reset_index()
print(ser10_df)
print(type(ser10_df))
```

```
   Sales Method  Units Sold
0      In-store    689990
1       Online    939093
2       Outlet    849778
<class 'pandas.core.frame.DataFrame'>
```

In [1528]:

```
ser10_df.sort_values(by="Units Sold",ascending=True)
```

Out[1528]:

	Sales Method	Units Sold
0	In-store	689990
2	Outlet	849778
1	Online	939093

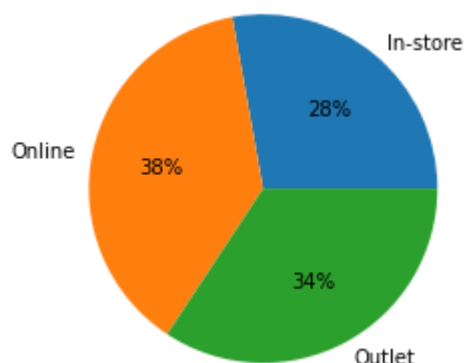
In [1529]:

```
#Sales Method with max Units Sold
print(ser10_df.loc[ser10_df['Units Sold'] == ser10_df['Units Sold'].max()])
```

```
   Sales Method  Units Sold
1       Online    939093
```

In [1530]:

```
plt.pie(ser10_df["Units Sold"], labels = ser10_df["Sales Method"], autopct='%1.0f%%')
plt.show()
```



In [1531]:

```
#Online purchase done more!
```

## REGION AND OPERATING PROFIT

In [1532]:

```
#group region and sum of operating profit
ser11 = df2.groupby(by=['Region'])['Operating Profit'].sum()
print(ser11)
print(type(ser11))
```

```
Region
Midwest      52811412
Northeast    68020688
South        61138044
Southeast    60555462
West         89609516
Name: Operating Profit, dtype: int32
<class 'pandas.core.series.Series'>
```

In [1533]:

```
#converting series to Data Frame
ser11_df = ser11.reset_index()
print(ser11_df)
print(type(ser11_df))
```

```
   Region  Operating Profit
0  Midwest      52811412
1  Northeast    68020688
2    South     61138044
3  Southeast    60555462
4    West      89609516
<class 'pandas.core.frame.DataFrame'>
```

In [1534]:

```
ser11_df.sort_values(by="Operating Profit",ascending=True)
```

Out[1534]:

	Region	Operating Profit
0	Midwest	52811412
3	Southeast	60555462
2	South	61138044
1	Northeast	68020688
4	West	89609516

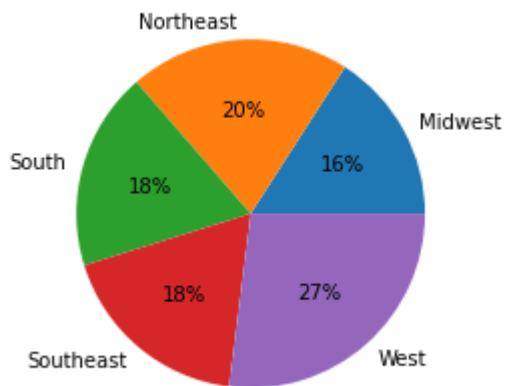
In [1535]:

```
#Region with max Operating Profit  
print(ser11_df.loc[ser11_df['Operating Profit'] == ser11_df['Operating Profit'].max()])
```

```
Region Operating Profit  
4 West 89609516
```

In [1536]:

```
plt.pie(ser11_df["Operating Profit"], labels = ser11_df["Region"],autopct='%1.0f%%')  
plt.show()
```



In [1537]:

```
#profit in west region
```

## REGION AND PRICE/UNIT

In [1538]:

```
#group region and price/unit
ser12 = df2.groupby(by=['Region'])['Price per Unit'].unique()
print(ser12)
print(type(ser12))
```

```
Region
Midwest      [55, 45, 35, 40, 50, 60, 65, 20, 30, 25, 15, 7...
Northeast    [50, 40, 45, 60, 55, 65, 70, 25, 35, 75, 30, 2...
South         [40, 35, 25, 30, 45, 50, 55, 20, 60, 65, 70, 1...
Southeast     [55, 45, 35, 40, 50, 60, 65, 70, 25, 30, 75, 8...
West          [55, 50, 40, 60, 65, 80, 70, 75, 35, 45, 30, 8...
Name: Price per Unit, dtype: object
<class 'pandas.core.series.Series'>
```

In [1539]:

```
#converting series to Data Frame
ser12_df = ser12.reset_index()
print(ser12_df)
print(type(ser12_df))
```

```
      Region      Price per Unit
0  Midwest  [55, 45, 35, 40, 50, 60, 65, 20, 30, 25, 15, 7...
1  Northeast  [50, 40, 45, 60, 55, 65, 70, 25, 35, 75, 30, 2...
2    South   [40, 35, 25, 30, 45, 50, 55, 20, 60, 65, 70, 1...
3  Southeast  [55, 45, 35, 40, 50, 60, 65, 70, 25, 30, 75, 8...
4    West    [55, 50, 40, 60, 65, 80, 70, 75, 35, 45, 30, 8...
<class 'pandas.core.frame.DataFrame'>
```

In [1540]:

```
#max of Price/Unit
x=ser12_df['max'] = [max(x) for x in ser12_df['Price per Unit']]
print(ser12_df['max'])
print(max(x))
```

```
0      80
1      80
2     110
3     110
4     105
Name: max, dtype: int32
110
```

In [1541]:

```
#Region with maximum price/unit
ser12_df.iloc[[2,3]]
```

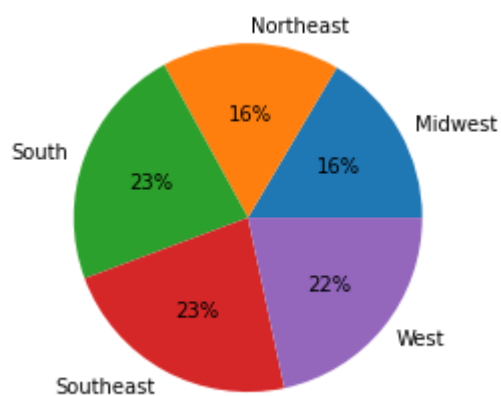
Out[1541]:

	Region	Price per Unit	max
2	South	[40, 35, 25, 30, 45, 50, 55, 20, 60, 65, 70, 1...	110
3	Southeast	[55, 45, 35, 40, 50, 60, 65, 70, 25, 30, 75, 8...	110



In [1542]:

```
plt.pie(scr12_df["max"], labels = scr12_df["Region"],autopct='%1.0f%%')  
plt.show()
```



In [1543]:

```
#south and southeast region with more price/unit but west region profits by 1%
```

## CITY AND OPERATING PROFIT

In [1544]:

```
#group city and sum of operating profit  
ser13 = df2.groupby(by=['City'])['Operating Profit'].sum()  
print(ser13)  
print(type(ser13))
```

City	
Albany	9429864
Albuquerque	6738070
Anchorage	4498359
Atlanta	6893299
Baltimore	2757648
Billings	6232040
Birmingham	9147581
Boise	8121123
Boston	3353884
Burlington	5785973
Charleston	15607205
Charlotte	9756425
Cheyenne	6544076
Chicago	3920377
Columbus	7528843
Dallas	8843074
Denver	7713562
Des Moines	2655220
Detroit	8135902
Fargo	2688518
Hartford	4152327
Honolulu	5849802
Houston	9845140
Indianapolis	3379262
Jackson	6369102
Knoxville	8493670
Las Vegas	7060660
Little Rock	4915703
Los Angeles	9044931
Louisville	3935833
Manchester	5786209
Miami	12168628
Milwaukee	2800835
Minneapolis	2670235
New Orleans	9417239
New York	13899981
Newark	3657247
Oklahoma City	4106535
Omaha	2439482
Orlando	8757591
Philadelphia	4156749
Phoenix	5344678
Portland	10760813
Providence	3045828
Richmond	7719439
Salt Lake City	3873440
San Francisco	10256252
Seattle	6991412
Sioux Falls	2943322
St. Louis	3907217
Wichita	3510159
Wilmington	4524358

Name: Operating Profit, dtype: int32  
<class 'pandas.core.series.Series'>

In [1545]:

```
#converting series to Data Frame  
ser13_df = ser13.reset_index()  
print(ser13_df)  
print(type(ser13_df))
```

	City	Operating Profit
0	Albany	9429864
1	Albuquerque	6738070
2	Anchorage	4498359
3	Atlanta	6893299
4	Baltimore	2757648
5	Billings	6232040
6	Birmingham	9147581
7	Boise	8121123
8	Boston	3353884
9	Burlington	5785973
10	Charleston	15607205
11	Charlotte	9756425
12	Cheyenne	6544076
13	Chicago	3920377
14	Columbus	7528843
15	Dallas	8843074
16	Denver	7713562
17	Des Moines	2655220
18	Detroit	8135902
19	Fargo	2688518
20	Hartford	4152327
21	Honolulu	5849802
22	Houston	9845140
23	Indianapolis	3379262
24	Jackson	6369102
25	Knoxville	8493670
26	Las Vegas	7060660
27	Little Rock	4915703
28	Los Angeles	9044931
29	Louisville	3935833
30	Manchester	5786209
31	Miami	12168628
32	Milwaukee	2800835
33	Minneapolis	2670235
34	New Orleans	9417239
35	New York	13899981
36	Newark	3657247
37	Oklahoma City	4106535
38	Omaha	2439482
39	Orlando	8757591
40	Philadelphia	4156749
41	Phoenix	5344678
42	Portland	10760813
43	Providence	3045828
44	Richmond	7719439
45	Salt Lake City	3873440
46	San Francisco	10256252
47	Seattle	6991412
48	Sioux Falls	2943322
49	St. Louis	3907217
50	Wichita	3510159
51	Wilmington	4524358

<class 'pandas.core.frame.DataFrame'>

In [1546]:

```
ser13_df.sort_values(by="Operating Profit",ascending=True)
```

Out[1546]:

	City	Operating Profit
38	Omaha	2439482
17	Des Moines	2655220
33	Minneapolis	2670235
19	Fargo	2688518
4	Baltimore	2757648
32	Milwaukee	2800835
48	Sioux Falls	2943322
43	Providence	3045828
8	Boston	3353884
23	Indianapolis	3379262
50	Wichita	3510159
36	Newark	3657247
45	Salt Lake City	3873440
49	St. Louis	3907217
13	Chicago	3920377
29	Louisville	3935833
37	Oklahoma City	4106535
20	Hartford	4152327
40	Philadelphia	4156749
2	Anchorage	4498359
51	Wilmington	4524358
27	Little Rock	4915703
41	Phoenix	5344678
9	Burlington	5785973
30	Manchester	5786209
21	Honolulu	5849802
5	Billings	6232040
24	Jackson	6369102
12	Cheyenne	6544076
1	Albuquerque	6738070
3	Atlanta	6893299
47	Seattle	6991412
26	Las Vegas	7060660
14	Columbus	7528843
16	Denver	7713562
44	Richmond	7719439
7	Boise	8121123

	City	Operating Profit
18	Detroit	8135902
25	Knoxville	8493670
39	Orlando	8757591
15	Dallas	8843074
28	Los Angeles	9044931
6	Birmingham	9147581
34	New Orleans	9417239
0	Albany	9429864
11	Charlotte	9756425
22	Houston	9845140
46	San Francisco	10256252
42	Portland	10760813
31	Miami	12168628
35	New York	13899981
10	Charleston	15607205

In [1547]:

```
#City with max Total Sales
print(ser13_df.loc[ser13_df['Operating Profit'] == ser13_df['Operating Profit'].max()])
```

```
      City  Operating Profit
10  Charleston      15607205
```

In [1548]:

```
#Charleston in west region has more profit
```

## CITY AND PRICE/UNIT



In [1549]:

```
#group city,region and price/unit
ser14 = df2.groupby(by=['City', 'Region'])['Price per Unit'].unique()

#converting series to Data Frame
ser14_df = ser14.reset_index()
print(ser14_df)
print(type(ser14_df))
#max of Price/unit
x=ser14_df['max'] = [max(x) for x in ser14_df['Price per Unit']]
print(ser14_df['max'])
print(max(x))
```

```
   City      Region \
0   Albany  Northeast
1  Albuquerque    West
2   Anchorage    West
3   Atlanta   Southeast
4   Baltimore  Northeast
5   Billings   Midwest
6  Birmingham    South
7    Boise     West
8   Boston   Northeast
9  Burlington  Northeast
10  Charleston  Northeast
11  Charleston  Southeast
12  Charlotte   Southeast
13  Cheyenne     West
14  Chicago    Midwest
15  Columbus    Midwest
16   Dallas     South
17  Denver     West
18  Denver     West
```

In [1550]:

```
ser14_df.iloc[[40]]
```

Out[1550]:

	City	Region	Price per Unit	max
40	Orlando	Southeast	[50, 55, 60, 65, 70, 75, 85, 95, 110, 90, 80, ...	110

In [1551]:

```
#Orlando in southeast has max price/unit
```

## MONTH AND PROFIT

In [1552]:

```
df2['Invoice Date'] = df2['Invoice Date'].str.replace('-', '')
```

In [1553]:

```
df2.head()
```

Out[1553]:

	Retailer	Retailer ID	Invoice Date	Region	State	City	Product	Price per Unit	Units Sold	Total Sales	Operati F
0	Foot Locker	1185732	01012020	Northeast	New York	New York	Men's Street Footwear	50	1200	600000	30
1	Foot Locker	1185732	02012020	Northeast	New York	New York	Men's Athletic Footwear	50	1000	500000	15
2	Foot Locker	1185732	03012020	Northeast	New York	New York	Women's Street Footwear	40	1000	400000	14
3	Foot Locker	1185732	04012020	Northeast	New York	New York	Women's Athletic Footwear	45	850	382500	13
4	Foot Locker	1185732	05012020	Northeast	New York	New York	Men's Apparel	60	900	540000	16

In [1554]:

```
df2['Invoice Date'] = df2['Invoice Date'].map(lambda x: str(x)[2:4])
```

In [1555]:

```
df2.head()
```

Out[1555]:

	Retailer	Retailer ID	Invoice Date	Region	State	City	Product	Price per Unit	Units Sold	Total Sales	Operati Prc
0	Foot Locker	1185732	01	Northeast	New York	New York	Men's Street Footwear	50	1200	600000	3000
1	Foot Locker	1185732	01	Northeast	New York	New York	Men's Athletic Footwear	50	1000	500000	1500
2	Foot Locker	1185732	01	Northeast	New York	New York	Women's Street Footwear	40	1000	400000	1400
3	Foot Locker	1185732	01	Northeast	New York	New York	Women's Athletic Footwear	45	850	382500	1338
4	Foot Locker	1185732	01	Northeast	New York	New York	Men's Apparel	60	900	540000	1620

In [1556]:

```
#group invoice date and sum of operating profit
ser15 = df2.groupby(by=['Invoice Date'])['Operating Profit'].sum()
print(ser15)
print(type(ser15))
```

```
Invoice Date
01    25141970
02    21392765
03    20439808
04    27559278
05    29946287
06    26714734
07    34054930
08    34451469
09    31009630
10    25078465
11    24755550
12    31590236
Name: Operating Profit, dtype: int32
<class 'pandas.core.series.Series'>
```

In [1557]:

```
#converting series to Data Frame
ser15_df = ser15.reset_index()
print(ser15_df)
print(type(ser15_df))
```

```
Invoice Date  Operating Profit
0           01         25141970
1           02         21392765
2           03         20439808
3           04         27559278
4           05         29946287
5           06         26714734
6           07         34054930
7           08         34451469
8           09         31009630
9           10         25078465
10          11         24755550
11          12         31590236
<class 'pandas.core.frame.DataFrame'>
```

In [1558]:

```
ser15_df.sort_values(by="Operating Profit",ascending=True)
```

Out[1558]:

	Invoice Date	Operating Profit
2	03	20439808
1	02	21392765
10	11	24755550
9	10	25078465
0	01	25141970
5	06	26714734
3	04	27559278
4	05	29946287
8	09	31009630
11	12	31590236
6	07	34054930
7	08	34451469

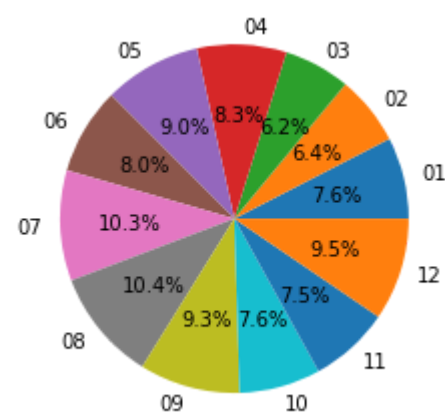
In [1559]:

```
#Invoice Date with max Operating Profit
print(ser15_df.loc[ser15_df['Operating Profit'] == ser15_df['Operating Profit'].max()])
```

	Invoice Date	Operating Profit
7	08	34451469

In [1560]:

```
plt.pie(ser15_df["Operating Profit"], labels = ser15_df["Invoice Date"],autopct='%0.1f%%',
plt.show()
```



In [1561]:

```
#maximum profit only in August
```