Sales Analysis - Atharva Rodge

In [1]:
```
# ! pip install pyarrow
```

In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings

from warnings import filterwarnings
filterwarnings('ignore')
```

Data extraction

In [3]:
```
all_data = pd.read_feather(r'C:\Users\Atharva\Desktop\Data Analysis Course\Sales_data_analysis/Sales_data.ft
```

In [4]:
```
all_data.head()
```

Out[4]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
0	176558	USB-C Charging Cable	2	11.95	04/19/19 08:46	917 1st St, Dallas, TX 75001
1	None	None	None	None	None	None
2	176559	Bose SoundSport Headphones	1	99.99	04/07/19 22:30	682 Chestnut St, Boston, MA 02215
3	176560	Google Phone	1	600	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001
4	176560	Wired Headphones	1	11.99	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001

In [5]:
```
all_data.isnull().sum()
```

Out[5]:
```
Order ID            545
Product             545
Quantity Ordered    545
Price Each          545
Order Date          545
Purchase Address    545
dtype: int64
```

In [6]:
```
all_data = all_data.dropna(how = 'all')
```

- default value of dropna() is na we change it how all where it will only drop na where all the values in the rows/columns are na

In [7]:
```
all_data.isnull().sum()
```

Out[7]:
```
Order ID            0
Product             0
Quantity Ordered    0
Price Each          0
Order Date          0
Purchase Address    0
dtype: int64
```

In [8]:
```
all_data.duplicated()
```

Out[8]:
```
0         False
2         False
3         False
4         False
5         False
          ...
186845    False
186846    False
186847    False
186848    False
186849    False
Length: 186305, dtype: bool
```

In [9]:
```
all_data[all_data.duplicated()]
```

Out[9]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
31	176585	Bose SoundSport Headphones	1	99.99	04/07/19 11:31	823 Highland St, Boston, MA 02215
1149	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
1155	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
1302	177795	Apple Airpods Headphones	1	150	04/27/19 19:45	740 14th St, Seattle, WA 98101
1684	178158	USB-C Charging Cable	1	11.95	04/28/19 21:13	197 Center St, San Francisco, CA 94016
...	...	...	...	...	...	...
186563	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
186632	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
186738	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
186782	259296	Apple Airpods Headphones	1	150	09/28/19 16:48	894 6th St, Dallas, TX 75001
186785	259297	Lightning Charging Cable	1	14.95	09/15/19 18:54	138 Main St, Boston, MA 02215

618 rows × 6 columns

In [10]:
```
all_data = all_data.drop_duplicates()
```

In [11]:
```
all_data.shape
```

Out[11]:
```
(185687, 6)
```

In [12]:
```
all_data[all_data.duplicated()]
```

Out[12]:

Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address

Which is the best month for sale?

In [13]:
```
all_data.columns
```

Out[13]:
```
Index(['Order ID', 'Product', 'Quantity Ordered', 'Price Each', 'Order Date',
       'Purchase Address'],
      dtype='object')
```

In [14]:
```
all_data.dtypes
```

Out[14]:
```
Order ID            object
Product             object
Quantity Ordered    object
Price Each          object
Order Date          object
Purchase Address    object
dtype: object
```

In [15]:
```
all_data['Order Date'][0]
```

Out[15]:
```
'04/19/19 08:46'
```

In [16]:
```
all_data['Order Date'][0].split(' ')[0]
```

Out[16]:
```
'04/19/19'
```

In [17]:
```
all_data['Order Date'][0].split(' ')[0].split('/')[0]
```

Out[17]:
```
'04'
```

In [18]:
```
# extracting months using above apporach
def return_month(x):
    return x.split('/')[0]
```

In [19]:
```
all_data['Month'] = all_data['Order Date'].apply(return_month)
```

In [20]:
```
all_data.dtypes
```

Out[20]:
```
Order ID            object
Product             object
Quantity Ordered    object
Price Each          object
Order Date          object
Purchase Address    object
Month               object
dtype: object
```

In [21]:
```
all_data['Month'].astype(int)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[21], line 1
----> 1 all_data['Month'].astype(int)

File ~\anaconda3\lib\site-packages\pandas\core\generic.py:6240, in NDFrame.astype(self, dtype, copy, error
s)
   6233     results = [
   6234         self.iloc[:, i].astype(dtype, copy=copy)
   6235         for i in range(len(self.columns))
   6236     ]
   6238 else:
   6239     # else, only a single dtype is given
-> 6240     new_data = self._mgr.astype(dtype=dtype, copy=copy, errors=errors)
   6241     return self._constructor(new_data).__finalize__(self, method="astype")
   6243 # GH 33113: handle empty frame or series

File ~\anaconda3\lib\site-packages\pandas\core\internals\managers.py:448, in BaseBlockManager.astype(self,
dtype, copy, errors)
    447 def astype(self: T, dtype, copy: bool = False, errors: str = "raise") -> T:
--> 448     return self.apply("astype", dtype=dtype, copy=copy, errors=errors)

File ~\anaconda3\lib\site-packages\pandas\core\internals\managers.py:352, in BaseBlockManager.apply(self,
f, align_keys, ignore_failures, **kwargs)
    350         applied = b.apply(f, **kwargs)
    351     else:
--> 352         applied = getattr(b, f)(**kwargs)
    353 except (TypeError, NotImplementedError):
    354     if not ignore_failures:

File ~\anaconda3\lib\site-packages\pandas\core\internals\blocks.py:526, in Block.astype(self, dtype, copy,
errors)
    508 """
    509 Coerce to the new dtype.
    510
   (...)
    522 Block
    523 """
    524 values = self.values
--> 526 new_values = astype_array_safe(values, dtype, copy=copy, errors=errors)
    528 new_values = maybe_coerce_values(new_values)
    529 newb = self.make_block(new_values)

File ~\anaconda3\lib\site-packages\pandas\core\dtypes\astype.py:299, in astype_array_safe(values, dtype, co
py, errors)
    296     return values.copy()
    298 try:
--> 299     new_values = astype_array(values, dtype, copy=copy)
    300 except (ValueError, TypeError):
    301     # e.g. astype_nansafe can fail on object-dtype of strings
    302     #  trying to convert to float
    303     if errors == "ignore":

File ~\anaconda3\lib\site-packages\pandas\core\dtypes\astype.py:230, in astype_array(values, dtype, copy)
    227     values = values.astype(dtype, copy=copy)
    229 else:
--> 230     values = astype_nansafe(values, dtype, copy=copy)
    232 # in pandas we don't store numpy str dtypes, so convert to object
    233 if isinstance(dtype, np.dtype) and issubclass(values.dtype.type, str):

File ~\anaconda3\lib\site-packages\pandas\core\dtypes\astype.py:170, in astype_nansafe(arr, dtype, copy, sk
ipna)
    166     raise ValueError(msg)
    168 if copy or is_object_dtype(arr.dtype) or is_object_dtype(dtype):
    169     # Explicit copy, or required since NumPy can't view from / to object.
--> 170     return arr.astype(dtype, copy=True)
    172 return arr.astype(dtype, copy=copy)

ValueError: invalid literal for int() with base 10: 'Order Date'
```

we have a error in the above code chunk lets resolve it

In [22]:
```
all_data['Month'].unique()
```

Out[22]:
```
array(['04', '05', 'Order Date', '08', '09', '12', '01', '02', '03', '07',
       '06', '11', '10'], dtype=object)
```

- we can see that there is 'Order Date; object in our list because of which it is throwing an error lets manipulate our data to remove the error

In [23]:
```
filter1 = all_data['Month'] == 'Order Date'
```

In [24]:
```
all_data[filter1]
```

Out[24]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month
519	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Order Date

In [25]:
```
all_data = all_data[~filter1]
```

In [26]:
```
all_data['Month'] = all_data['Month'].astype(int)
```

In [27]:
```
all_data.dtypes
```

Out[27]:
```
Order ID            object
Product             object
Quantity Ordered    object
Price Each          object
Order Date          object
Purchase Address    object
Month                int32
dtype: object
```

In [28]:
```
all_data['Quantity Ordered'] = all_data['Quantity Ordered'].astype(int)
all_data['Price Each'] = all_data['Price Each'].astype(float)
```

In [29]:
```
all_data.dtypes
```

Out[29]:
```
Order ID             object
Product              object
Quantity Ordered      int32
Price Each          float64
Order Date           object
Purchase Address     object
Month                 int32
dtype: object
```

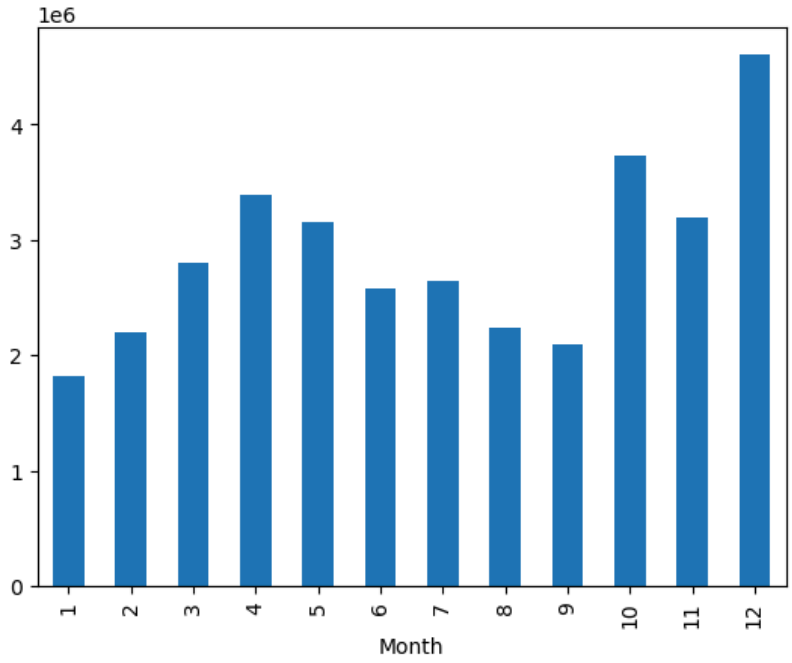
In [30]:
```
all_data['Sales'] = all_data['Quantity Ordered'] * all_data['Price Each']
```

In [31]:
```
all_data['Sales']
```

Out[31]:
```
0          23.90
2          99.99
3         600.00
4          11.99
5          11.99
             ...
186845      8.97
186846    700.00
186847    700.00
186848    379.99
186849     11.95
Name: Sales, Length: 185686, dtype: float64
```

In [32]:
```
all_data.groupby(['Month'])['Sales'].sum().plot(kind='bar')
```

Out[32]:
```
<Axes: xlabel='Month'>
```



Which city has maximum orders

In [33]:
```
all_data.head()
```

Out[33]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales
0	176558	USB-C Charging Cable	2	11.95	04/19/19 08:46	917 1st St, Dallas, TX 75001	4	23.90
2	176559	Bose SoundSport Headphones	1	99.99	04/07/19 22:30	682 Chestnut St, Boston, MA 02215	4	99.99
3	176560	Google Phone	1	600.00	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4	600.00
4	176560	Wired Headphones	1	11.99	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4	11.99
5	176561	Wired Headphones	1	11.99	04/30/19 09:27	333 8th St, Los Angeles, CA 90001	4	11.99

In [34]:
```
all_data['Purchase Address'][0].split(',')[1]
```

Out[34]:
```
' Dallas'
```

In [35]:
```
all_data['Purchase Address'].str.split(',').str.get(1)
```

Out[35]:
```
0                Dallas
2                Boston
3           Los Angeles
4           Los Angeles
5           Los Angeles
               ...
186845      Los Angeles
186846    San Francisco
186847    San Francisco
186848    San Francisco
186849    San Francisco
Name: Purchase Address, Length: 185686, dtype: object
```

In [36]:
```
# Using function
def return_city(x):
    return x.split(',')[1]
```

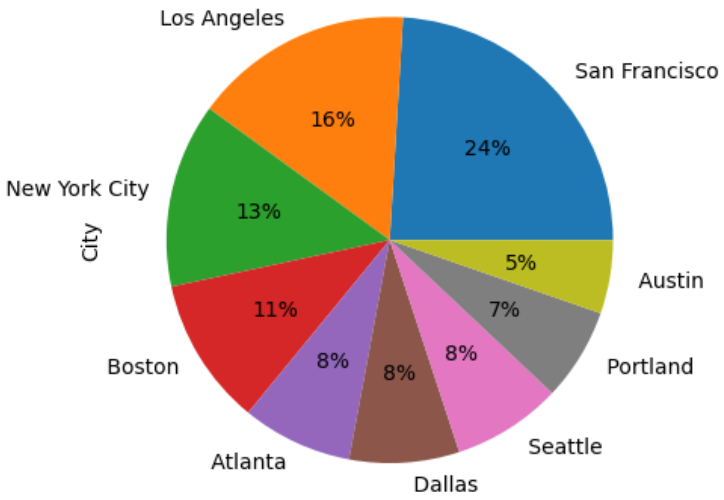
In [37]:
```
all_data['City'] = all_data['Purchase Address'].apply(return_city)
```

In [38]:
```
all_data['City']
```

Out[38]:
```
0                 Dallas
2                 Boston
3            Los Angeles
4            Los Angeles
5            Los Angeles
                ...
186845       Los Angeles
186846     San Francisco
186847     San Francisco
186848     San Francisco
186849     San Francisco
Name: City, Length: 185686, dtype: object
```

In [39]:
```
all_data['City'].value_counts().plot(kind ='pie', autopct = '%1.0f%%')
```

Out[39]:
```
<Axes: ylabel='City'>
```



Understanding which product sold the most and why?

why? depends on various parameters like user rating or lower price etc which we will explore in this section

In [40]:
```
all_data.columns
```

Out[40]:
```
Index(['Order ID', 'Product', 'Quantity Ordered', 'Price Each', 'Order Date',
       'Purchase Address', 'Month', 'Sales', 'City'],
      dtype='object')
```

In [41]:
```
new_data = all_data.groupby(['Product']).agg({'Quantity Ordered':'sum', 'Price Each':'mean'})
```

In [42]:
```
new_data = new_data.reset_index()
```

In [43]:
```
new_data.head()
```

Out[43]:

	Product	Quantity Ordered	Price Each
0	20in Monitor	4126	109.99
1	27in 4K Gaming Monitor	6239	389.99
2	27in FHD Monitor	7541	149.99
3	34in Ultrawide Monitor	6192	379.99
4	AA Batteries (4-pack)	27615	3.84

In [44]:
```
new_data['Product'].values
```

Out[44]:
```
array(['20in Monitor', '27in 4K Gaming Monitor', '27in FHD Monitor',
       '34in Ultrawide Monitor', 'AA Batteries (4-pack)',
       'AAA Batteries (4-pack)', 'Apple Airpods Headphones',
       'Bose SoundSport Headphones', 'Flatscreen TV', 'Google Phone',
       'LG Dryer', 'LG Washing Machine', 'Lightning Charging Cable',
       'Macbook Pro Laptop', 'ThinkPad Laptop', 'USB-C Charging Cable',
       'Vareebadd Phone', 'Wired Headphones', 'iPhone'], dtype=object)
```

In [45]:
```
products = new_data['Product'].values
```

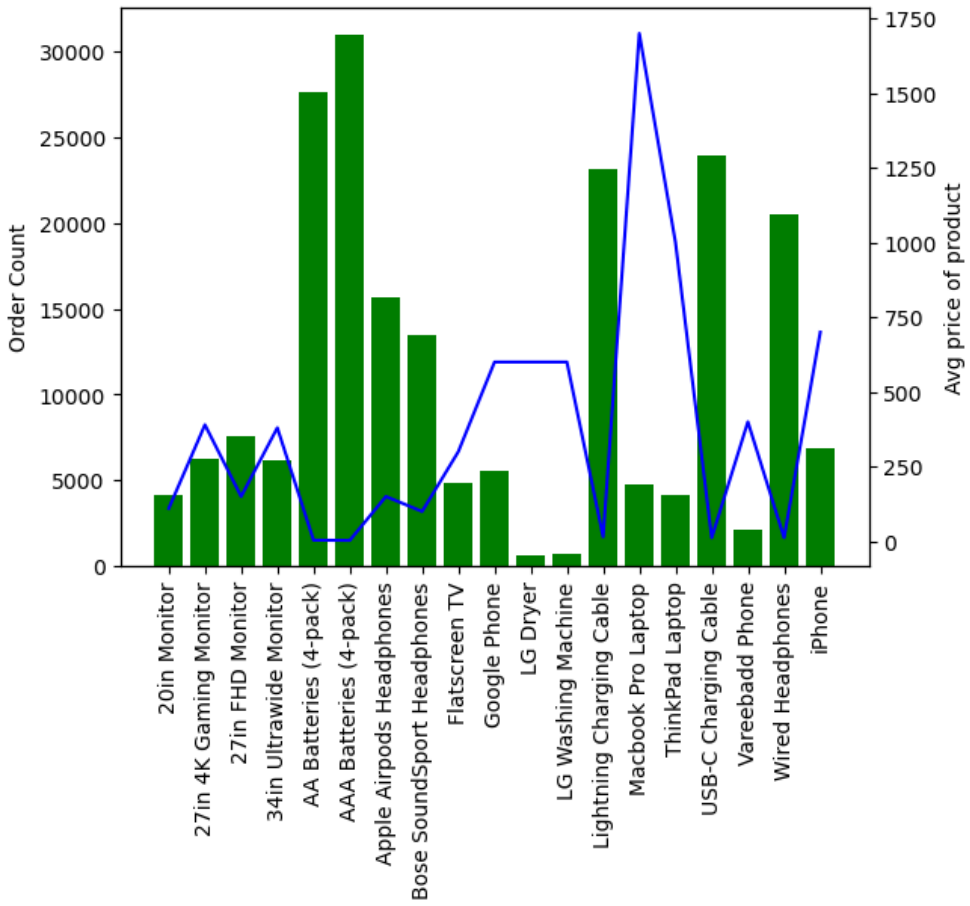
In [46]:
```
fig , ax1 = plt.subplots()

ax2 = ax1.twinx()

ax1.bar(new_data['Product'], new_data['Quantity Ordered'], color = 'g')
ax2.plot(new_data['Product'], new_data['Price Each'], color='b')
ax1.set_xticklabels(products  , rotation = 'vertical')

ax1.set_ylabel('Order Count')
ax2.set_ylabel('Avg price of product')
```

Out[46]:
```
Text(0, 0.5, 'Avg price of product')
```



Understanding the tren of the most sold product

In [47]:
```
all_data.columns
```

Out[47]:
```
Index(['Order ID', 'Product', 'Quantity Ordered', 'Price Each', 'Order Date',
       'Purchase Address', 'Month', 'Sales', 'City'],
      dtype='object')
```

In [48]:
```
all_data['Product'].value_counts()[0:5]
```

Out[48]:
```
USB-C Charging Cable        21859
Lightning Charging Cable    21610
AAA Batteries (4-pack)      20612
AA Batteries (4-pack)       20558
Wired Headphones            18849
Name: Product, dtype: int64
```

In [49]:
```
all_data['Product'].value_counts()[0:5].index
```

Out[49]:
```
Index(['USB-C Charging Cable', 'Lightning Charging Cable',
       'AAA Batteries (4-pack)', 'AA Batteries (4-pack)', 'Wired Headphones'],
      dtype='object')
```

In [50]:
```
most_sold_products = all_data['Product'].value_counts()[0:5].index
```

In [51]:
```
all_data['Product'].isin(most_sold_products)
```

Out[51]:
```
0          True
2         False
3         False
4          True
5          True
          ...
186845     True
186846    False
186847    False
186848    False
186849     True
Name: Product, Length: 185686, dtype: bool
```

In [52]:
```
all_data[all_data['Product'].isin(most_sold_products)]
```

Out[52]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City
0	176558	USB-C Charging Cable	2	11.95	04/19/19 08:46	917 1st St, Dallas, TX 75001	4	23.90	Dallas
4	176560	Wired Headphones	1	11.99	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4	11.99	Los Angeles
5	176561	Wired Headphones	1	11.99	04/30/19 09:27	333 8th St, Los Angeles, CA 90001	4	11.99	Los Angeles
6	176562	USB-C Charging Cable	1	11.95	04/29/19 13:03	381 Wilson St, San Francisco, CA 94016	4	11.95	San Francisco
8	176564	USB-C Charging Cable	1	11.95	04/12/19 10:58	790 Ridge St, Atlanta, GA 30301	4	11.95	Atlanta
...	...	...	...	...	...	...	...	...	...
186840	259349	AAA Batteries (4-pack)	1	2.99	09/01/19 22:14	911 River St, Dallas, TX 75001	9	2.99	Dallas
186842	259350	USB-C Charging Cable	1	11.95	09/30/19 13:49	519 Maple St, San Francisco, CA 94016	9	11.95	San Francisco
186844	259352	USB-C Charging Cable	1	11.95	09/07/19 15:49	976 Forest St, San Francisco, CA 94016	9	11.95	San Francisco
186845	259353	AAA Batteries (4-pack)	3	2.99	09/17/19 20:56	840 Highland St, Los Angeles, CA 90001	9	8.97	Los Angeles
186849	259357	USB-C Charging Cable	1	11.95	09/30/19 00:18	250 Meadow St, San Francisco, CA 94016	9	11.95	San Francisco

103488 rows × 9 columns

In [53]:
```
most_sold_producs_df = all_data[all_data['Product'].isin(most_sold_products)]
```

In [54]:
```
most_sold_producs_df.head()
```

Out[54]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City
0	176558	USB-C Charging Cable	2	11.95	04/19/19 08:46	917 1st St, Dallas, TX 75001	4	23.90	Dallas
4	176560	Wired Headphones	1	11.99	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4	11.99	Los Angeles
5	176561	Wired Headphones	1	11.99	04/30/19 09:27	333 8th St, Los Angeles, CA 90001	4	11.99	Los Angeles
6	176562	USB-C Charging Cable	1	11.95	04/29/19 13:03	381 Wilson St, San Francisco, CA 94016	4	11.95	San Francisco
8	176564	USB-C Charging Cable	1	11.95	04/12/19 10:58	790 Ridge St, Atlanta, GA 30301	4	11.95	Atlanta

In [55]:
```
most_sold_producs_df.groupby(['Month','Product']).size()
```

Out[55]:
```
Month  Product
1      AA Batteries (4-pack)       1037
       AAA Batteries (4-pack)      1084
       Lightning Charging Cable    1069
       USB-C Charging Cable        1171
       Wired Headphones            1004
2      AA Batteries (4-pack)       1274
       AAA Batteries (4-pack)      1320
       Lightning Charging Cable    1393
       USB-C Charging Cable        1511
       Wired Headphones            1179
3      AA Batteries (4-pack)       1672
       AAA Batteries (4-pack)      1645
       Lightning Charging Cable    1749
       USB-C Charging Cable        1766
       Wired Headphones            1512
4      AA Batteries (4-pack)       2062
       AAA Batteries (4-pack)      1988
       Lightning Charging Cable    2197
       USB-C Charging Cable        2074
       Wired Headphones            1888
5      AA Batteries (4-pack)       1821
       AAA Batteries (4-pack)      1888
       Lightning Charging Cable    1929
       USB-C Charging Cable        1879
       Wired Headphones            1729
6      AA Batteries (4-pack)       1540
       AAA Batteries (4-pack)      1451
       Lightning Charging Cable    1560
       USB-C Charging Cable        1531
       Wired Headphones            1334
7      AA Batteries (4-pack)       1555
       AAA Batteries (4-pack)      1554
       Lightning Charging Cable    1690
       USB-C Charging Cable        1667
       Wired Headphones            1434
8      AA Batteries (4-pack)       1357
       AAA Batteries (4-pack)      1340
       Lightning Charging Cable    1354
       USB-C Charging Cable        1339
       Wired Headphones            1191
9      AA Batteries (4-pack)       1314
       AAA Batteries (4-pack)      1281
       Lightning Charging Cable    1324
       USB-C Charging Cable        1451
       Wired Headphones            1173
10     AA Batteries (4-pack)       2240
       AAA Batteries (4-pack)      2234
       Lightning Charging Cable    2414
       USB-C Charging Cable        2437
       Wired Headphones            2091
11     AA Batteries (4-pack)       1970
       AAA Batteries (4-pack)      1999
       Lightning Charging Cable    2044
       USB-C Charging Cable        2054
       Wired Headphones            1777
12     AA Batteries (4-pack)       2716
       AAA Batteries (4-pack)      2828
       Lightning Charging Cable    2887
       USB-C Charging Cable        2979
       Wired Headphones            2537
dtype: int64
```

In [56]:
```
pivot_df = most_sold_producs_df.groupby(['Month','Product']).size().unstack()
```

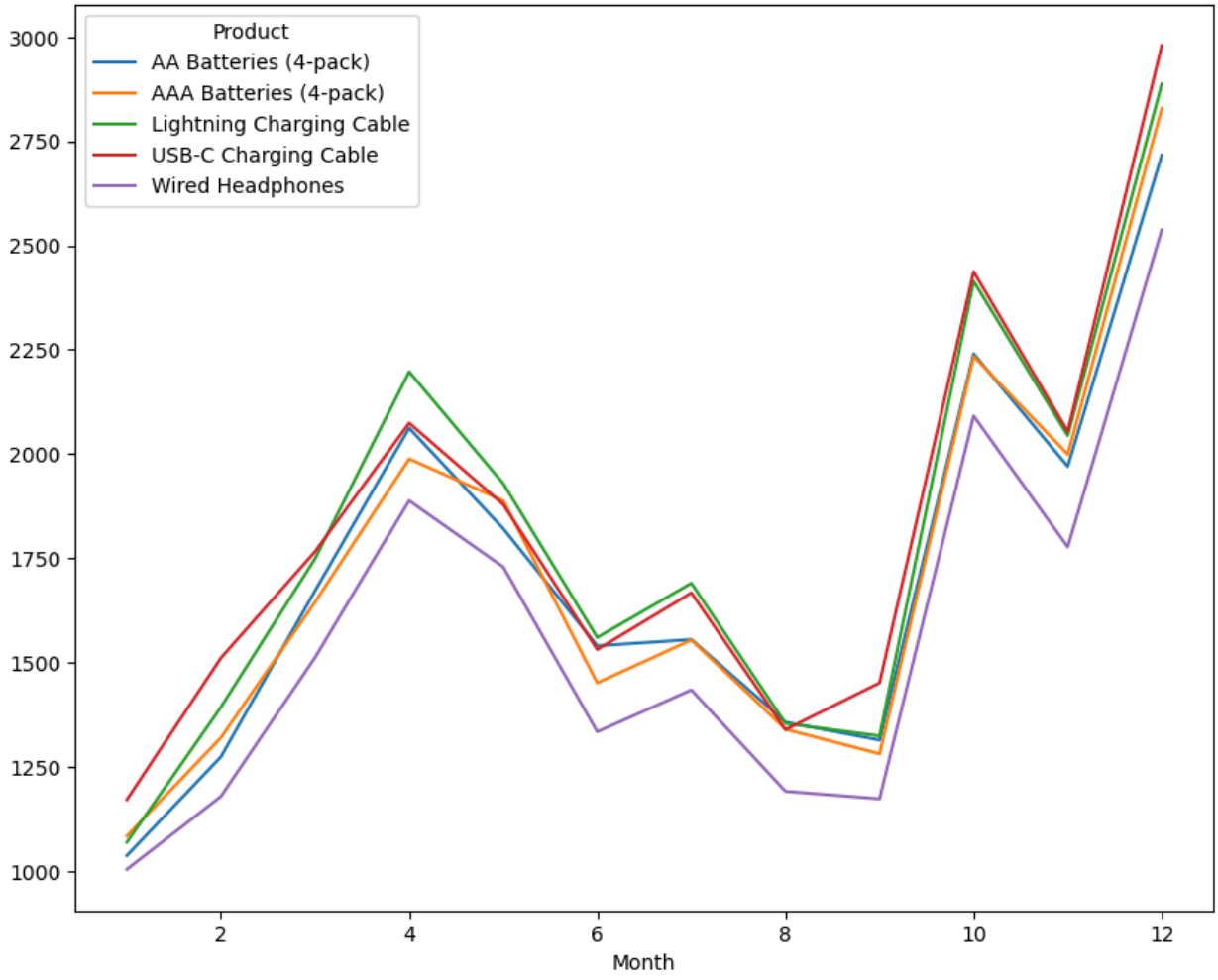
In [57]:
```
pivot_df
```

Out[57]:

Product Month	AA Batteries (4-pack)	AAA Batteries (4-pack)	Lightning Charging Cable	USB-C Charging Cable	Wired Headphones
1	1037	1084	1069	1171	1004
2	1274	1320	1393	1511	1179
3	1672	1645	1749	1766	1512
4	2062	1988	2197	2074	1888
5	1821	1888	1929	1879	1729
6	1540	1451	1560	1531	1334
7	1555	1554	1690	1667	1434
8	1357	1340	1354	1339	1191
9	1314	1281	1324	1451	1173
10	2240	2234	2414	2437	2091
11	1970	1999	2044	2054	1777
12	2716	2828	2887	2979	2537

In [58]:
```
pivot_df.plot(figsize = (10,8))
```

Out[58]:
```
<Axes: xlabel='Month'>
```



What products are most often sold together?

In [59]:
```
all_data.head()
```

Out[59]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City
0	176558	USB-C Charging Cable	2	11.95	04/19/19 08:46	917 1st St, Dallas, TX 75001	4	23.90	Dallas
2	176559	Bose SoundSport Headphones	1	99.99	04/07/19 22:30	682 Chestnut St, Boston, MA 02215	4	99.99	Boston
3	176560	Google Phone	1	600.00	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4	600.00	Los Angeles
4	176560	Wired Headphones	1	11.99	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4	11.99	Los Angeles
5	176561	Wired Headphones	1	11.99	04/30/19 09:27	333 8th St, Los Angeles, CA 90001	4	11.99	Los Angeles

In [60]:
```
all_data.columns
```

Out[60]:
```
Index(['Order ID', 'Product', 'Quantity Ordered', 'Price Each', 'Order Date',
       'Purchase Address', 'Month', 'Sales', 'City'],
      dtype='object')
```

In [61]:
```
all_data['Order ID']
```

Out[61]:
```
0         176558
2         176559
3         176560
4         176560
5         176561
           ...
186845    259353
186846    259354
186847    259355
186848    259356
186849    259357
Name: Order ID, Length: 185686, dtype: object
```

In [62]:
```
all_data['Order ID'].duplicated(keep = False) # Kepp the duplicated rows because the products are bought tog
```

Out[62]:
```
0         False
2         False
3          True
4          True
5         False
          ...
186845    False
186846    False
186847    False
186848    False
186849    False
Name: Order ID, Length: 185686, dtype: bool
```

In [63]:
```
df_duplicated = all_data[all_data['Order ID'].duplicated(keep = False)]
```

In [64]:
```
df_duplicated # data frame with duplicated order id
```

Out[64]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City
3	176560	Google Phone	1	600.00	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4	600.00	Los Angeles
4	176560	Wired Headphones	1	11.99	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4	11.99	Los Angeles
18	176574	Google Phone	1	600.00	04/03/19 19:42	20 Hill St, Los Angeles, CA 90001	4	600.00	Los Angeles
19	176574	USB-C Charging Cable	1	11.95	04/03/19 19:42	20 Hill St, Los Angeles, CA 90001	4	11.95	Los Angeles
32	176586	AAA Batteries (4-pack)	2	2.99	04/10/19 17:00	365 Center St, San Francisco, CA 94016	4	5.98	San Francisco
...	...	...	...	...	...	...	...	...	...
186792	259303	AA Batteries (4-pack)	1	3.84	09/20/19 20:18	106 7th St, Atlanta, GA 30301	9	3.84	Atlanta

- as we can see that the the products bought together

In [65]:
```
dup_products = df_duplicated.groupby(['Order ID'])['Product'].apply(lambda x : ','.join(x)).reset_index().re
```

In [66]:
```
dup_products
```

Out[66]:

	Order ID	grouped_products
0	141275	USB-C Charging Cable,Wired Headphones
1	141290	Apple Airpods Headphones,AA Batteries (4-pack)
2	141365	Vareebadd Phone,Wired Headphones
3	141384	Google Phone,USB-C Charging Cable
4	141450	Google Phone,Bose SoundSport Headphones
...	...	...
6874	319536	Macbook Pro Laptop,Wired Headphones
6875	319556	Google Phone,Wired Headphones
6876	319584	iPhone,Wired Headphones
6877	319596	iPhone,Lightning Charging Cable
6878	319631	34in Ultrawide Monitor,Lightning Charging Cable

6879 rows × 2 columns

In [67]:
```
duplicated_products_df = df_duplicated.merge(dup_products, how = 'left', on = 'Order ID')
```

In [68]:
```
duplicated_products_df
```

Out[68]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City	grouped_products
0	176560	Google Phone	1	600.00	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4	600.00	Los Angeles	Google Phone,Wired Headphones
1	176560	Wired Headphones	1	11.99	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4	11.99	Los Angeles	Google Phone,Wired Headphones
2	176574	Google Phone	1	600.00	04/03/19 19:42	20 Hill St, Los Angeles, CA 90001	4	600.00	Los Angeles	Google Phone,USB-C Charging Cable
3	176574	USB-C Charging Cable	1	11.95	04/03/19 19:42	20 Hill St, Los Angeles, CA 90001	4	11.95	Los Angeles	Google Phone,USB-C Charging Cable
4	176586	AAA Batteries (4-pack)	2	2.99	04/10/19 17:00	365 Center St, San Francisco, CA 94016	4	5.98	San Francisco	AAA Batteries (4-pack),Google Phone

as we have two order id we will now drop the duplicates

In [69]:
```
no_dup_df = duplicated_products_df.drop_duplicates(subset = ['Order ID'])
```

In [70]:
```
no_dup_df.shape
```

Out[70]:
```
(6879, 10)
```

In [71]:
```
no_dup_df
```

Out[71]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City	grouped_products
0	176560	Google Phone	1	600.00	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001	4	600.00	Los Angeles	Google Phone,Wired Headphones
2	176574	Google Phone	1	600.00	04/03/19 19:42	20 Hill St, Los Angeles, CA 90001	4	600.00	Los Angeles	Google Phone,USB-C Charging Cable
4	176586	AAA Batteries (4-pack)	2	2.99	04/10/19 17:00	365 Center St, San Francisco, CA 94016	4	5.98	San Francisco	AAA Batteries (4-pack),Google Phone
6	176672	Lightning Charging Cable	1	14.95	04/12/19 11:07	778 Maple St, New York City, NY 10001	4	14.95	New York City	Lightning Charging Cable,USB-C Charging Cable
8	176681	Apple Airpods Headphones	1	150.00	04/20/19 10:39	331 Cherry St, Seattle, WA 98101	4	150.00	Seattle	Apple Airpods Headphones,ThinkPad Laptop
...	...	...	...	...	...	...	...	...	...	...
14118	259277	iPhone	1	700.00	09/28/19 13:07	795 Willow St, New York City, NY 10001	9	700.00	New York City	iPhone,Wired Headphones
14120	259297	iPhone	1	700.00	09/15/19 18:54	138 Main St, Boston, MA 02215	9	700.00	Boston	iPhone,Lightning Charging Cable
14122	259303	34in Ultrawide Monitor	1	379.99	09/20/19 20:18	106 7th St, Atlanta, GA 30301	9	379.99	Atlanta	34in Ultrawide Monitor,AA Batteries (4-pack)
14124	259314	Wired Headphones	1	11.99	09/16/19 00:25	241 Highland St, Atlanta, GA 30301	9	11.99	Atlanta	Wired Headphones,AAA Batteries (4-pack)
14126	259350	Google Phone	1	600.00	09/30/19 13:49	519 Maple St, San Francisco, CA 94016	9	600.00	San Francisco	Google Phone,USB-C Charging Cable

6879 rows × 10 columns

In [73]:
```
no_dup_df['grouped_products'][0:5].value_counts().plot.pie()
```

Out[73]:
```
<Axes: ylabel='grouped_products'>
```