```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Loading the dataset

# Our dataset is a feather file

# Feather is a binary file format that is used for storing data .. Feather is a fast, lightweight, and easy-to-use binary file format for storing data.It shows high I/O speed, doesn’t take too much memory on the disk and doesn’t need any unpacking when loaded back into RAM. Feather has max I/O speed

pip install pyarrow
```

```
Requirement already satisfied: pyarrow in c:\users\manzoor\anaconda3\lib\site-packages (14.0.2)Note: you may need to restart the kernel to use updated packages.

Requirement already satisfied: numpy>=1.16.6 in c:\users\manzoor\anaconda3\lib\site-packages (from pyarrow) (1.26.4)
```

```
all_data = pd.read_feather(r"C:\Users\MANZOOR\Downloads/sales_data.ftr")
```

```
all_data.head(5)
```

```
  Order ID                     Product Quantity Ordered Price Each  \
0   176558        USB-C Charging Cable                2      11.95
1     None                        None             None       None
2   176559  Bose SoundSport Headphones                1      99.99
3   176560                Google Phone                1        600
4   176560            Wired Headphones                1      11.99

       Order Date                        Purchase Address
0  04/19/19 08:46          917 1st St, Dallas, TX 75001
1            None                                    None
2  04/07/19 22:30     682 Chestnut St, Boston, MA 02215
3  04/12/19 14:38  669 Spruce St, Los Angeles, CA 90001
4  04/12/19 14:38  669 Spruce St, Los Angeles, CA 90001
```

DATA CLEANING AND FORMATTING

```
all_data.isnull().sum() ## checking out total missing values we have
```

```
Order ID            545
Product             545
Quantity Ordered    545
Price Each          545
Order Date          545
Purchase Address    545
dtype: int64
```

```
### since there 545 observations where entire row have missing value ,
we can drop these 545 rows..

all_data = all_data.dropna(how="all")

all_data.isnull().sum()
```

```
Order ID            0
Product             0
Quantity Ordered    0
Price Each          0
Order Date          0
Purchase Address    0
dtype: int64
```

```
### check whether we have duplicate rows or not !

all_data.duplicated()
```

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

```
all_data[all_data.duplicated()] ## total 618 duplicate rows ..
```

```
        Order ID                     Product  Quantity Ordered  Price
Each  \
```

```
31        176585  Bose SoundSport Headphones                 1
99.99
1149    Order ID                     Product  Quantity Ordered  Price
Each
1155    Order ID                     Product  Quantity Ordered  Price
Each
1302      177795    Apple Airpods Headphones                 1
150
1684      178158        USB-C Charging Cable                 1
11.95
...          ...                          ...               ...
...
186563  Order ID                     Product  Quantity Ordered  Price
Each
186632  Order ID                     Product  Quantity Ordered  Price
Each
186738  Order ID                     Product  Quantity Ordered  Price
Each
186782    259296    Apple Airpods Headphones                 1
150
186785    259297    Lightning Charging Cable                 1
14.95

            Order Date                          Purchase Address
31      04/07/19 11:31       823 Highland St, Boston, MA 02215
1149        Order Date                          Purchase Address
1155        Order Date                          Purchase Address
1302    04/27/19 19:45          740 14th St, Seattle, WA 98101
1684    04/28/19 21:13  197 Center St, San Francisco, CA 94016
...                ...                                     ...
186563      Order Date                          Purchase Address
186632      Order Date                          Purchase Address
186738      Order Date                          Purchase Address
186782  09/28/19 16:48            894 6th St, Dallas, TX 75001
186785  09/15/19 18:54          138 Main St, Boston, MA 02215

[618 rows x 6 columns]
```

```
all_data = all_data.drop_duplicates() ## Dropping all the duplicate
rows ..
```

```
all_data[all_data.duplicated()]
```

```
Empty DataFrame
Columns: [Order ID, Product, Quantity Ordered, Price Each, Order Date,
Purchase Address]
Index: []
```

2. WHICH IS THE BEST MONTH FOR SALE ?

```
## Lets checks the data type

all_data.dtypes
```

```
Order ID            object
Product             object
Quantity Ordered    object
Price Each          object
Order Date          object
Purchase Address    object
dtype: object
```

```
## change order-date data type to datetime

all_data['Order Date'] = pd.to_datetime(all_data['Order Date'],
errors='coerce')
```

```
C:\Users\MANZOOR\AppData\Local\Temp\ipykernel_27204\4061391744.py:1:
UserWarning: Could not infer format, so each element will be parsed
individually, falling back to `dateutil`. To ensure parsing is
consistent and as-expected, please specify a format.
  all_data['Order Date'] = pd.to_datetime(all_data['Order Date'],
errors='coerce')
```

```
all_data.dtypes
```

```
Order ID                    object
Product                     object
Quantity Ordered            object
Price Each                  object
Order Date          datetime64[ns]
Purchase Address            object
dtype: object
```

```
## Now we need extract month from the order date

all_data['Month'] = all_data['Order Date'].dt.month_name()

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

```
---------------------------------------------------------------------
-----
ValueError                                Traceback (most recent call
last)
Cell In[47], line 1
----> 1 all_data['Month'] = all_data['Month'].astype(int)

File ~\anaconda3\Lib\site-packages\pandas\core\generic.py:6643, in
NDFrame.astype(self, dtype, copy, errors)
   6637     results = [
```

```
   6638         ser.astype(dtype, copy=copy, errors=errors) for _, ser
in self.items()
   6639     ]
   6641 else:
   6642     # else, only a single dtype is given
-> 6643     new_data = self._mgr.astype(dtype=dtype, copy=copy,
errors=errors)
   6644     res = self._constructor_from_mgr(new_data,
axes=new_data.axes)
   6645     return res.__finalize__(self, method="astype")

File ~\anaconda3\Lib\site-packages\pandas\core\internals\
managers.py:430, in BaseBlockManager.astype(self, dtype, copy, errors)
    427 elif using_copy_on_write():
    428     copy = False
--> 430 return self.apply(
    431     "astype",
    432     dtype=dtype,
    433     copy=copy,
    434     errors=errors,
    435     using_cow=using_copy_on_write(),
    436 )

File ~\anaconda3\Lib\site-packages\pandas\core\internals\
managers.py:363, in BaseBlockManager.apply(self, f, align_keys,
**kwargs)
    361         applied = b.apply(f, **kwargs)
    362     else:
--> 363         applied = getattr(b, f)(**kwargs)
    364     result_blocks = extend_blocks(applied, result_blocks)
    366 out = type(self).from_blocks(result_blocks, self.axes)

File ~\anaconda3\Lib\site-packages\pandas\core\internals\
blocks.py:758, in Block.astype(self, dtype, copy, errors, using_cow,
squeeze)
    755         raise ValueError("Can not squeeze with more than one
column.")
    756     values = values[0, :]  # type: ignore[call-overload]
--> 758 new_values = astype_array_safe(values, dtype, copy=copy,
errors=errors)
    760 new_values = maybe_coerce_values(new_values)
    762 refs = None

File ~\anaconda3\Lib\site-packages\pandas\core\dtypes\astype.py:237,
in astype_array_safe(values, dtype, copy, errors)
    234     dtype = dtype.numpy_dtype
    236 try:
--> 237     new_values = astype_array(values, dtype, copy=copy)
    238 except (ValueError, TypeError):
    239     # e.g. _astype_nansafe can fail on object-dtype of strings
```

```
    240     #  trying to convert to float
    241     if errors == "ignore":

File ~\anaconda3\Lib\site-packages\pandas\core\dtypes\astype.py:182,
in astype_array(values, dtype, copy)
    179     values = values.astype(dtype, copy=copy)
    181 else:
--> 182     values = _astype_nansafe(values, dtype, copy=copy)
    184 # in pandas we don't store numpy str dtypes, so convert to
object
    185 if isinstance(dtype, np.dtype) and
issubclass(values.dtype.type, str):

File ~\anaconda3\Lib\site-packages\pandas\core\dtypes\astype.py:133,
in _astype_nansafe(arr, dtype, copy, skipna)
    129     raise ValueError(msg)
    131 if copy or arr.dtype == object or dtype == object:
    132     # Explicit copy, or required since NumPy can't view from /
to object.
--> 133     return arr.astype(dtype, copy=True)
    135 return arr.astype(dtype, copy=copy)

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

```
all_data.head(5)
```

```
  Order ID                     Product Quantity Ordered Price Each  \
0   176558        USB-C Charging Cable                2      11.95
2   176559  Bose SoundSport Headphones                1      99.99
3   176560                Google Phone                1        600
4   176560            Wired Headphones                1      11.99
5   176561            Wired Headphones                1      11.99

           Order Date                       Purchase Address  Month
0 2019-04-19 08:46:00          917 1st St, Dallas, TX 75001  April
2 2019-04-07 22:30:00     682 Chestnut St, Boston, MA 02215  April
3 2019-04-12 14:38:00  669 Spruce St, Los Angeles, CA 90001  April
4 2019-04-12 14:38:00  669 Spruce St, Los Angeles, CA 90001  April
5 2019-04-30 09:27:00     333 8th St, Los Angeles, CA 90001  April
```

```
# Change the data type of Quantity ordered and Price Each

all_data['Quantity Ordered'] = pd.to_numeric(all_data['Quantity
Ordered'], errors='coerce')
all_data['Price Each'] = pd.to_numeric(all_data['Price Each'],
errors='coerce')

all_data.dtypes
```

```
Order ID                     object
Product                      object
Quantity Ordered            float64
Price Each                  float64
Order Date           datetime64[ns]
Purchase Address             object
Month                        object
dtype: object
```

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

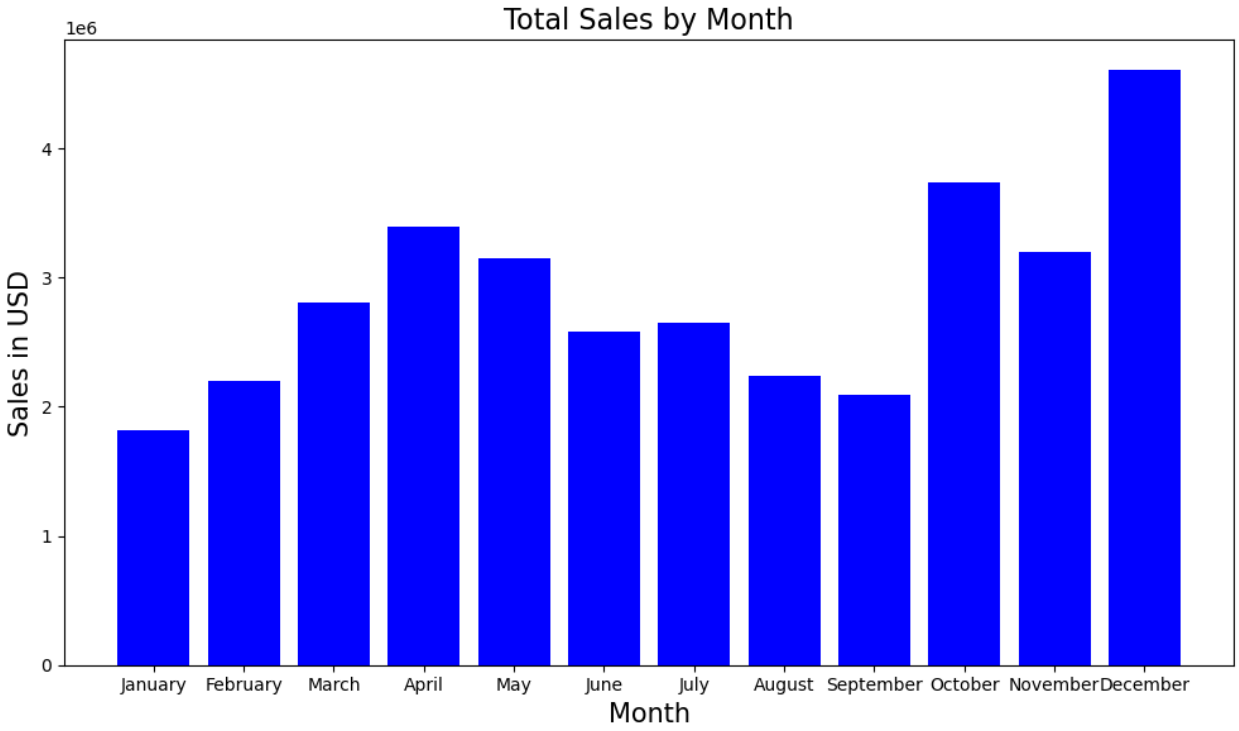
```
all_data.head(5)
```

```
  Order ID                     Product  Quantity Ordered  Price
Each  \
0   176558        USB-C Charging Cable               2.0       11.95

2   176559  Bose SoundSport Headphones               1.0       99.99

3   176560                Google Phone               1.0      600.00

4   176560            Wired Headphones               1.0       11.99

5   176561            Wired Headphones               1.0       11.99


           Order Date                       Purchase Address  Month
Sales
0 2019-04-19 08:46:00          917 1st St, Dallas, TX 75001  April
23.90
2 2019-04-07 22:30:00     682 Chestnut St, Boston, MA 02215  April
99.99
3 2019-04-12 14:38:00  669 Spruce St, Los Angeles, CA 90001  April
600.00
4 2019-04-12 14:38:00  669 Spruce St, Los Angeles, CA 90001  April
11.99
5 2019-04-30 09:27:00     333 8th St, Los Angeles, CA 90001  April
11.99
```

```
monthly_sales = all_data.groupby('Month').sum(numeric_only=True)
['Sales']

month_order = ["January", "February", "March", "April", "May", "June",
"July", "August", "September", "October", "November", "December"]
monthly_sales = monthly_sales.reindex(month_order)

best_month = monthly_sales.sort_values(ascending=False)



print(best_month)
```

```
Month
December     4608295.70
October      3734777.86
April        3389217.98
November     3197875.05
May          3150616.23
March        2804973.35
July         2646461.32
June         2576280.15
August       2241083.37
February     2200078.08
September    2094465.69
January      1821413.16
Name: Sales, dtype: float64
```

```
# Bar Chart for Visualisation

plt.figure(figsize=(10, 6))  # Set figure size
plt.bar(monthly_sales.index, monthly_sales.values, color='blue')

# Set the title and labels for the chart
plt.title('Total Sales by Month', fontsize=16)
plt.xlabel('Month', fontsize=15)
plt.ylabel('Sales in USD', fontsize=15)
plt.tight_layout()  # Adjust layout to fit everything nicely
plt.show()
```



3.WHICH CITY HAS MAXIMUM ORDER ?

```
all_data.head()
```

```
  Order ID                     Product  Quantity Ordered  Price
Each  \
0   176558        USB-C Charging Cable               2.0       11.95

2   176559  Bose SoundSport Headphones               1.0       99.99

3   176560                Google Phone               1.0      600.00

4   176560            Wired Headphones               1.0       11.99

5   176561            Wired Headphones               1.0       11.99


           Order Date                       Purchase Address  Month
Sales
0 2019-04-19 08:46:00          917 1st St, Dallas, TX 75001  April
23.90
2 2019-04-07 22:30:00     682 Chestnut St, Boston, MA 02215  April
99.99
3 2019-04-12 14:38:00  669 Spruce St, Los Angeles, CA 90001  April
600.00
4 2019-04-12 14:38:00  669 Spruce St, Los Angeles, CA 90001  April
11.99
5 2019-04-30 09:27:00     333 8th St, Los Angeles, CA 90001  April
11.99
```

```
print("Missing values in Month column:",
all_data['Month'].isnull().sum())
```

```
Missing values in Month column: 1
```

```
all_data.dropna(subset=['Month'], inplace=True)
```

```
print("Missing values in Month column:",
all_data['Month'].isnull().sum())
```

```
Missing values in Month column: 0
```

```
all_data['Purchase Address'][0]
```

```
'917 1st St, Dallas, TX 75001'
```

```
## Need to extract the city name from the address
```

```
#all_data['Purchase Address'][0].split(',')[1] ## extracting city from
"Purchase Address"
```

```
' Dallas'
```

```
all_data['City'] = all_data['Purchase
Address'].str.split(',').str.get(1)
```

```
all_data['City']
```

```
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
```

```
all_data.head(5)
```

```
  Order ID                      Product  Quantity Ordered  Price
Each  \
0   176558        USB-C Charging Cable               2.0       11.95

2   176559  Bose SoundSport Headphones               1.0       99.99

3   176560                Google Phone               1.0      600.00

4   176560            Wired Headphones               1.0       11.99

5   176561            Wired Headphones               1.0       11.99


            Order Date                       Purchase Address  Month
Sales  \
0 2019-04-19 08:46:00          917 1st St, Dallas, TX 75001  April
23.90
2 2019-04-07 22:30:00     682 Chestnut St, Boston, MA 02215  April
99.99
3 2019-04-12 14:38:00  669 Spruce St, Los Angeles, CA 90001  April
600.00
```

```
4 2019-04-12 14:38:00  669 Spruce St, Los Angeles, CA 90001  April
11.99
5 2019-04-30 09:27:00     333 8th St, Los Angeles, CA 90001  April
11.99
```

```
           city          City
0        Dallas        Dallas
2        Boston        Boston
3   Los Angeles   Los Angeles
4   Los Angeles   Los Angeles
5   Los Angeles   Los Angeles
```

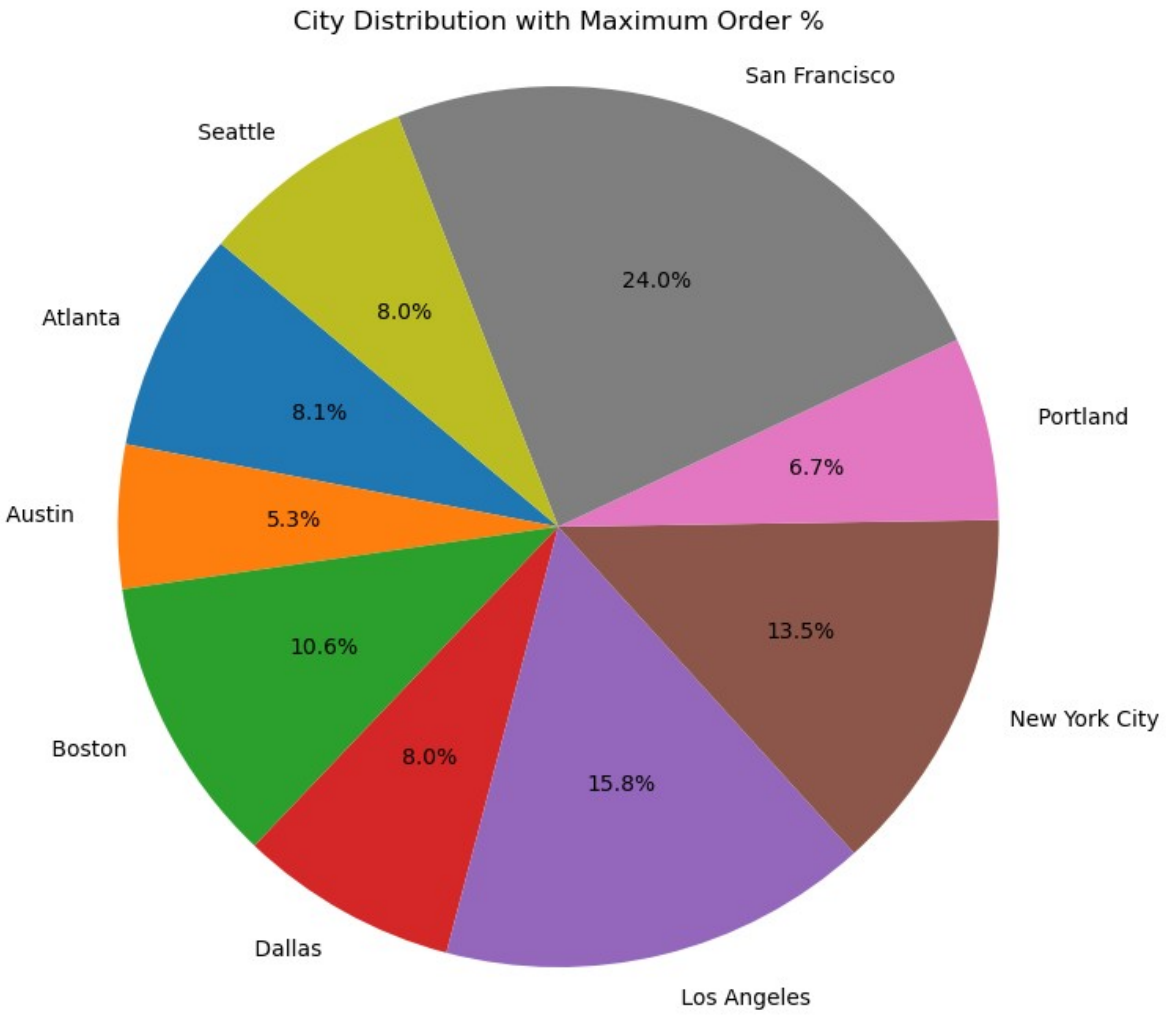
```
pd.value_counts(all_data['city'])
```

```
C:\Users\MANZOOR\AppData\Local\Temp\ipykernel_27204\2119930960.py:1:
FutureWarning: pandas.value_counts is deprecated and will be removed
in a future version. Use pd.Series(obj).value_counts() instead.
  pd.value_counts(all_data['city'])
```

```
city
San Francisco    44662
Los Angeles      29564
New York City    24847
Boston           19901
Atlanta          14863
Dallas           14797
Seattle          14713
Portland         12449
Austin            9890
Name: count, dtype: int64
```

```
City_Sales = all_data.groupby('City').sum(numeric_only=True)['Sales']
```

```
plt.figure(figsize=(10, 8))  # Set figure size
plt.pie(City_Sales, labels=City_Sales.index, autopct='%1.1f%%',
startangle=140)
plt.title('City Distribution with Maximum Order %')
plt.axis('equal')  # Equal aspect ratio ensures that pie chart is
circular
plt.show()
```




```
## New York , Los Angeles , San Francisco are the Top 3 cities which
has max order
```

What products are most often sold together ?

```
all_data.head()
```

```
  Order ID                     Product  Quantity Ordered  Price
Each  \
0   176558        USB-C Charging Cable               2.0       11.95
```

```
2   176559  Bose SoundSport Headphones                1.0        99.99

3   176560                Google Phone                1.0       600.00

4   176560             Wired Headphones               1.0        11.99

5   176561             Wired Headphones               1.0        11.99
```

```
           Order Date                       Purchase Address  Month
Sales  \
0 2019-04-19 08:46:00          917 1st St, Dallas, TX 75001  April
23.90
2 2019-04-07 22:30:00     682 Chestnut St, Boston, MA 02215  April
99.99
3 2019-04-12 14:38:00  669 Spruce St, Los Angeles, CA 90001  April
600.00
4 2019-04-12 14:38:00  669 Spruce St, Los Angeles, CA 90001  April
11.99
5 2019-04-30 09:27:00     333 8th St, Los Angeles, CA 90001  April
11.99
```

```
           city          City
0        Dallas        Dallas
2        Boston        Boston
3   Los Angeles   Los Angeles
4   Los Angeles   Los Angeles
5   Los Angeles   Los Angeles
```

```
all_data.columns
```

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

```
 ## dataframe in which we have those Order Ids who have purchased more
products !
```

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

```
df_duplicated
```

```
       Order ID                  Product  Quantity Ordered  Price Each
\
3        176560             Google Phone               1.0      600.00

4        176560        Wired Headphones               1.0       11.99

18       176574             Google Phone               1.0      600.00
```

```
19       176574    USB-C Charging Cable               1.0       11.95
32       176586  AAA Batteries (4-pack)               2.0        2.99
...         ...                     ...               ...         ...
186792   259303   AA Batteries (4-pack)               1.0        3.84
186803   259314        Wired Headphones               1.0       11.99
186804   259314  AAA Batteries (4-pack)               2.0        2.99
186841   259350            Google Phone               1.0      600.00
186842   259350    USB-C Charging Cable               1.0       11.95

                 Order Date                        Purchase Address
Month  \
3      2019-04-12 14:38:00    669 Spruce St, Los Angeles, CA 90001
April
4      2019-04-12 14:38:00    669 Spruce St, Los Angeles, CA 90001
April
18     2019-04-03 19:42:00       20 Hill St, Los Angeles, CA 90001
April
19     2019-04-03 19:42:00       20 Hill St, Los Angeles, CA 90001
April
32     2019-04-10 17:00:00  365 Center St, San Francisco, CA 94016
April
...                    ...                                     ...
...
186792 2019-09-20 20:18:00           106 7th St, Atlanta, GA 30301
September
186803 2019-09-16 00:25:00      241 Highland St, Atlanta, GA 30301
September
186804 2019-09-16 00:25:00      241 Highland St, Atlanta, GA 30301
September
186841 2019-09-30 13:49:00   519 Maple St, San Francisco, CA 94016
September
186842 2019-09-30 13:49:00   519 Maple St, San Francisco, CA 94016
September

         Sales            city            City
3       600.00     Los Angeles     Los Angeles
4        11.99     Los Angeles     Los Angeles
18      600.00     Los Angeles     Los Angeles
19       11.95     Los Angeles     Los Angeles
32        5.98   San Francisco   San Francisco
...        ...             ...             ...
186792    3.84         Atlanta         Atlanta
```

```
186803   11.99         Atlanta         Atlanta
186804    5.98         Atlanta         Atlanta
186841  600.00   San Francisco   San Francisco
186842   11.95   San Francisco   San Francisco

[14128 rows x 10 columns]
```

```
dup_products = df_duplicated.groupby(['Order ID'])
['Product'].apply(lambda x :
','.join(x)).reset_index().rename(columns={'Product':'grouped_products
'})

dup_products
```

```
     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 x 2 columns]
```

```
dup_products_df = df_duplicated.merge(dup_products , how='left' ,
on='Order ID') ## merge dataframes
```

```
dup_products_df
```

```
      Order ID                  Product  Quantity Ordered  Price
Each  \
0       176560             Google Phone               1.0      600.00

1       176560        Wired Headphones               1.0       11.99

2       176574             Google Phone               1.0      600.00

3       176574    USB-C Charging Cable               1.0       11.95
```

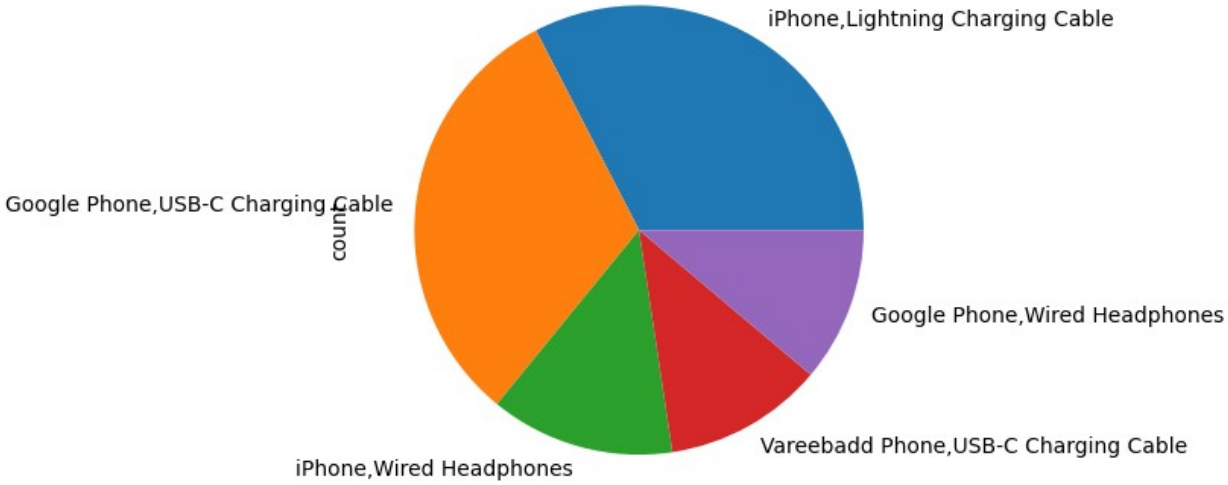
```
4       176586  AAA Batteries (4-pack)               2.0        2.99

...        ...                     ...               ...         ...

14123   259303   AA Batteries (4-pack)               1.0        3.84

14124   259314        Wired Headphones               1.0       11.99

14125   259314  AAA Batteries (4-pack)               2.0        2.99

14126   259350            Google Phone               1.0      600.00

14127   259350    USB-C Charging Cable               1.0       11.95


                Order Date                        Purchase Address
Month  \
0     2019-04-12 14:38:00    669 Spruce St, Los Angeles, CA 90001
April
1     2019-04-12 14:38:00    669 Spruce St, Los Angeles, CA 90001
April
2     2019-04-03 19:42:00       20 Hill St, Los Angeles, CA 90001
April
3     2019-04-03 19:42:00       20 Hill St, Los Angeles, CA 90001
April
4     2019-04-10 17:00:00  365 Center St, San Francisco, CA 94016
April
...                   ...                                    ...
...
14123 2019-09-20 20:18:00           106 7th St, Atlanta, GA 30301
September
14124 2019-09-16 00:25:00      241 Highland St, Atlanta, GA 30301
September
14125 2019-09-16 00:25:00      241 Highland St, Atlanta, GA 30301
September
14126 2019-09-30 13:49:00   519 Maple St, San Francisco, CA 94016
September
14127 2019-09-30 13:49:00   519 Maple St, San Francisco, CA 94016
September

        Sales            city            City  \
0      600.00     Los Angeles     Los Angeles
1       11.99     Los Angeles     Los Angeles
2      600.00     Los Angeles     Los Angeles
3       11.95     Los Angeles     Los Angeles
4        5.98   San Francisco   San Francisco
...       ...             ...             ...
14123    3.84         Atlanta         Atlanta
14124   11.99         Atlanta         Atlanta
14125    5.98         Atlanta         Atlanta
```

```
14126  600.00   San Francisco   San Francisco
14127   11.95   San Francisco   San Francisco

                                  grouped_products
0                    Google Phone,Wired Headphones
1                    Google Phone,Wired Headphones
2                  Google Phone,USB-C Charging Cable
3                  Google Phone,USB-C Charging Cable
4                AAA Batteries (4-pack),Google Phone
...                                             ...
14123  34in Ultrawide Monitor,AA Batteries (4-pack)
14124       Wired Headphones,AAA Batteries (4-pack)
14125       Wired Headphones,AAA Batteries (4-pack)
14126             Google Phone,USB-C Charging Cable
14127             Google Phone,USB-C Charging Cable

[14128 rows x 11 columns]



no_dup_df = dup_products_df.drop_duplicates(subset=['Order ID']) #
lets drop out all duplicate Order ID

no_dup_df

      Order ID                   Product  Quantity Ordered  Price Each
\
0       176560               Google Phone               1.0      600.00

2       176574               Google Phone               1.0      600.00

4       176586    AAA Batteries (4-pack)               2.0        2.99

6       176672  Lightning Charging Cable               1.0       14.95

8       176681  Apple Airpods Headphones               1.0      150.00

...        ...                       ...               ...         ...

14118   259277                    iPhone               1.0      700.00

14120   259297                    iPhone               1.0      700.00

14122   259303    34in Ultrawide Monitor               1.0      379.99

14124   259314          Wired Headphones               1.0       11.99

14126   259350               Google Phone               1.0      600.00


               Order Date                         Purchase Address
Month  \
```

```
0     2019-04-12 14:38:00    669 Spruce St, Los Angeles, CA 90001
April
2     2019-04-03 19:42:00       20 Hill St, Los Angeles, CA 90001
April
4     2019-04-10 17:00:00  365 Center St, San Francisco, CA 94016
April
6     2019-04-12 11:07:00   778 Maple St, New York City, NY 10001
April
8     2019-04-20 10:39:00        331 Cherry St, Seattle, WA 98101
April
...                   ...                                     ...
...
14118 2019-09-28 13:07:00  795 Willow St, New York City, NY 10001
September
14120 2019-09-15 18:54:00           138 Main St, Boston, MA 02215
September
14122 2019-09-20 20:18:00           106 7th St, Atlanta, GA 30301
September
14124 2019-09-16 00:25:00      241 Highland St, Atlanta, GA 30301
September
14126 2019-09-30 13:49:00   519 Maple St, San Francisco, CA 94016
September

        Sales            city            City  \
0      600.00     Los Angeles     Los Angeles
2      600.00     Los Angeles     Los Angeles
4        5.98   San Francisco   San Francisco
6       14.95   New York City   New York City
8      150.00         Seattle         Seattle
...       ...             ...             ...
14118  700.00   New York City   New York City
14120  700.00          Boston          Boston
14122  379.99         Atlanta         Atlanta
14124   11.99         Atlanta         Atlanta
14126  600.00   San Francisco   San Francisco

                                    grouped_products
0                     Google Phone,Wired Headphones
2                   Google Phone,USB-C Charging Cable
4                 AAA Batteries (4-pack),Google Phone
6      Lightning Charging Cable,USB-C Charging Cable
8           Apple Airpods Headphones,ThinkPad Laptop
...                                              ...
14118                         iPhone,Wired Headphones
14120                iPhone,Lightning Charging Cable
14122   34in Ultrawide Monitor,AA Batteries (4-pack)
14124        Wired Headphones,AAA Batteries (4-pack)
14126              Google Phone,USB-C Charging Cable

[6879 rows x 11 columns]
```

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

```
<Axes: ylabel='count'>
```




```
## ie as soon as any Person will bought Iphone , we can recommend him charging cable , wired headphones
## ie as soon as any Person will bought Google phone , we can recommend him USB-c charging cable


## This is a very important insight if someone is building recommendation system ..
```