```
from google.colab import files
uploded = files.upload()
```

Choose files transaction_data.csv

- **transaction_data.csv**(text/csv) - 71167 bytes, last modified: 13/04/2024 - 100% done

```
Saving transaction_data.csv to transaction_data.csv
```

```
import pandas as pd
import numpy as np
```

```
df = pd.read_csv('transaction_data.csv')
```

```
df
```

	CustomerID	TransactionDate	ProductID	ProductCategor
0	C0098	2021-01-29	P006	Electronic
1	C0085	2021-03-06	P019	Home Decc
2	C0082	2020-02-11	P016	Clothin
3	C0060	2020-09-22	P003	Clothin
4	C0018	2021-06-07	P012	Home Decc
...	...	...	...	.
995	C0033	2021-03-25	P010	Clothin
996	C0074	2020-07-28	P005	Electronic
997	C0080	2021-08-23	P006	Book
998	C0019	2021-12-28	P018	Book
999	C0073	2020-02-20	P010	Clothin

1000 rows × 7 columns

Next steps: Generate code with df   View recommended plots

Check if any null is available or not in date set

```
df.isnull().sum()
```

```
CustomerID         0
TransactionDate    0
ProductID          0
ProductCategory    0
Quantity           0
UnitPrice          0
TotalPrice         0
dtype: int64
```

This define that how many unique customer are there in the data set

```
df['CustomerID'].unique()
```

```
    array(['C0098', 'C0085', 'C0082', 'C0060', 'C0018', 'C0011', 'C0010',
           'C0063', 'C0058', 'C0026', 'C0042', 'C0005', 'C0008', 'C0083',
           'C0046', 'C0035', 'C0041', 'C0091', 'C0099', 'C0066', 'C0022',
           'C0009', 'C0073', 'C0025', 'C0053', 'C0002', 'C0055', 'C0013',
           'C0034', 'C0057', 'C0054', 'C0039', 'C0021', 'C0065', 'C0027',
           'C0081', 'C0078', 'C0015', 'C0003', 'C0097', 'C0045', 'C0064',
           'C0052', 'C0047', 'C0075', 'C0020', 'C0084', 'C0071', 'C0089',
           'C0061', 'C0088', 'C0070', 'C0044', 'C0033', 'C0094', 'C0036',
           'C0048', 'C0043', 'C0037', 'C0014', 'C0086', 'C0049', 'C0040',
           'C0068', 'C0030', 'C0028', 'C0067', 'C0001', 'C0051', 'C0031',
           'C0029', 'C0007', 'C0093', 'C0012', 'C0017', 'C0038', 'C0077',
           'C0023', 'C0019', 'C0095', 'C0024', 'C0032', 'C0050', 'C0090',
           'C0080', 'C0056', 'C0062', 'C0072', 'C0092', 'C0079', 'C0059',
           'C0096', 'C0069', 'C0100', 'C0016', 'C0074', 'C0004', 'C0076',
           'C0087', 'C0006'], dtype=object)
```

Check all data types

```
df.dtypes
```

```
    CustomerID                 object
    TransactionDate    datetime64[ns]
    ProductID                  object
    ProductCategory            object
    Quantity                    int64
    UnitPrice                 float64
    TotalPrice                float64
    dtype: object
```

Conver Data type to datetime for better analisis

```
df['TransactionDate'] = pd.to_datetime(df['TransactionDate'])
```

```
df['TransactionDate'].dtype
```

```
    dtype('<M8[ns]')
```

```
df.shape
```

```
    (1000, 7)
```

```
df.info()
```

```
    <class 'pandas.core.frame.DataFrame'>
    RangeIndex: 1000 entries, 0 to 999
```

```
Data columns (total 7 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   CustomerID       1000 non-null   object
 1   TransactionDate  1000 non-null   datetime64[ns]
 2   ProductID        1000 non-null   object
 3   ProductCategory  1000 non-null   object
 4   Quantity         1000 non-null   int64
 5   UnitPrice        1000 non-null   float64
 6   TotalPrice       1000 non-null   float64
dtypes: datetime64[ns](1), float64(2), int64(1), object(3)
memory usage: 54.8+ KB
```

This specify that how many quantity that Customer purchase on different Category of product

```
pivot_table = pd.pivot_table(df,values='Quantity',
                              index=['CustomerID'],
                              columns='ProductCategory',
                              aggfunc=np.sum)
pivot_table
```

ProductCategory	Books	Clothing	Electronics	Home Decor
CustomerID				
C0001	6.0	9.0	8.0	11.0
C0002	14.0	NaN	3.0	10.0
C0003	7.0	6.0	8.0	9.0
C0004	3.0	4.0	5.0	7.0
C0005	3.0	13.0	NaN	10.0
...	...	...	...	...
C0096	8.0	2.0	8.0	7.0
C0097	5.0	NaN	8.0	24.0
C0098	7.0	15.0	7.0	NaN
C0099	8.0	20.0	10.0	13.0
C0100	9.0	4.0	12.0	1.0

100 rows × 4 columns

Next steps: Generate code with pivot_table   View recommended plots

**This specify that how much amount Customer spend on different Category of product **

```
pivot_table = pd.pivot_table(df,values='TotalPrice',
                              index=['CustomerID'],
                              columns='ProductCategory',
                              aggfunc=np.sum)
pivot_table
```

ProductCategory	Books	Clothing	Electronics	H De
CustomerID				
C0001	372.465430	450.140194	278.888286	475.209
C0002	603.992737	NaN	103.826635	319.130
C0003	327.394379	256.562822	599.558798	377.139
C0004	97.226784	182.114063	347.579551	407.055
C0005	235.016467	1038.304271	NaN	384.650
...	...	...	...	
C0096	266.861212	76.744642	379.405203	586.373
C0097	228.093947	NaN	435.539516	1227.766
C0098	569.825846	626.285320	234.017439	
C0099	206.239940	963.684857	510.840306	931.784
C0100	545.646717	277.696405	738.469511	68.810

Next steps: Generate code with pivot_table View recommended plots

```
import datetime
df['Date_month'] = df['TransactionDate'].dt.month
```

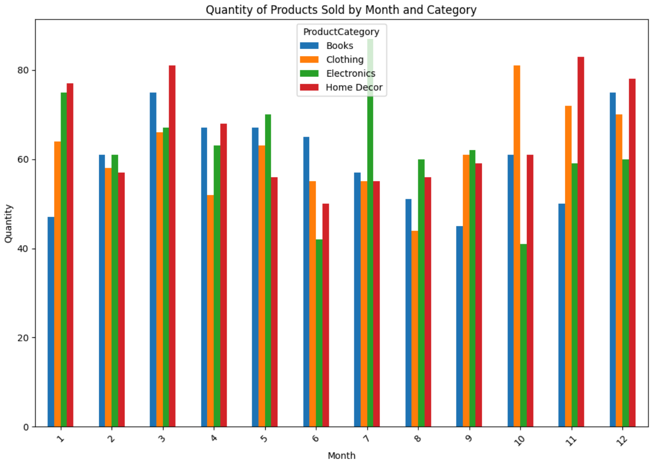
```
df['Date_month']
```

```
0       1
1       3
2       2
3       9
4       6
       ..
995     3
996     7
997     8
998    12
999     2
Name: Date_month, Length: 1000, dtype: int32
```

```
import matplotlib.pyplot as plt
```

This define that which category have how much sales in different months

```
grouped_data = df.groupby(['Date_month', 'ProductCategory'])['Quantity'].sum().unstack()

fig, ax = plt.subplots(figsize=(12, 8))
grouped_data.plot.bar(ax=ax)

ax.set_xlabel('Month')
ax.set_ylabel('Quantity')
ax.set_title('Quantity of Products Sold by Month and Category')

plt.xticks(rotation=45)

plt.show()
```



this define that how a customer spend total amount on different category

```
pivot_table_Customer = pd.pivot_table(df,values='TotalPrice',
                                     index=['ProductCategory'],
                                     columns='CustomerID',
                                     aggfunc=np.sum)
pivot_table_Customer
```

CustomerID	C0001	C0002	C0003	C0004
ProductCategory				
Books	372.465430	603.992737	327.394379	97.226784
Clothing	450.140194	NaN	256.562822	182.114063
Electronics	278.888286	103.826635	599.558798	347.579551
Home Decor	475.209927	319.130542	377.139363	407.055247

4 rows × 100 columns

Explain About Which month they spent high with data of product categorty

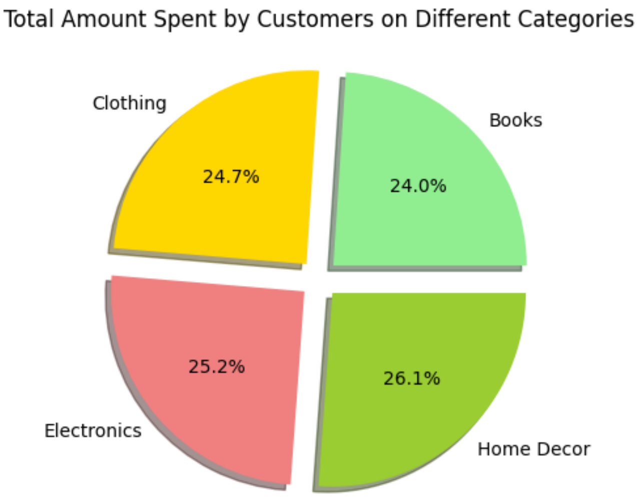
```
pivot_table_Customer_Month = pd.pivot_table(df,values='TotalPrice',
                                     index=['ProductCategory'],
                                     columns='Date_month',
                                     aggfunc=np.sum)
pivot_table_Customer
```

Date_month	1	2	3	
ProductCategory				
Books	2913.648824	3987.538709	3874.569293	3750.814
Clothing	3656.426753	3115.263090	3636.942081	2607.491
Electronics	4304.237628	3834.474992	3349.767684	3718.340
Home Decor	4319.701047	3869.547553	4459.128765	3897.005

Next steps: Generate code with `pivot_table_Customer` View recommended plots

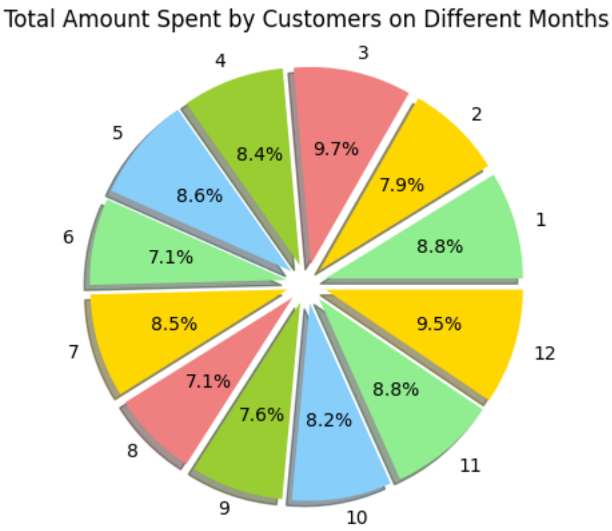
this define that how a customer spend total amount on different category in easy visual format

```
labels = pivot_table_Customer.index.to_list()
colors = ['lightgreen', 'gold', 'lightcoral', 'yellowgreen', 'lightskyblue']

plt.pie(pivot_table_Customer.sum(axis=1), explode=[0.1] * len(labels), labels=labels, autopct='%
plt.title("Total Amount Spent by Customers on Different Categories")
plt.show()
```



this define that how a customer spend total amount on different Months in Visual format

```
labels = grouped_data.index.to_list()
colors = ['lightgreen', 'gold', 'lightcoral', 'yellowgreen', 'lightskyblue']

plt.pie(grouped_data.sum(axis=1), explode=[0.1] * len(labels), labels=labels, autopct='%1.1f%%',
plt.title("Total Amount Spent by Customers on Different Months")
plt.show()
```



Customer By Details That How Many Quantity Of product that they Purchase

```
grouped_data_quantity = df.groupby(['CustomerID', 'ProductCategory'])['Quantity'].sum()
grouped_data_quantity
```

```
CustomerID  ProductCategory
C0001       Books               6
            Clothing            9
            Electronics         8
            Home Decor         11
C0002       Books              14
                               ..
C0099       Home Decor         13
C0100       Books               9
            Clothing            4
            Electronics        12
            Home Decor          1
Name: Quantity, Length: 361, dtype: int64
```

brief summary of findings and recommendations :

- **Here we can see the data of customer that have a higher purchase on Home Decore products**
- **We find that a Customer spend a more amount In March and December**
- **By the Visual Data of Bar char we find that our Home Decore Product Are less sales in Summer Seasion**
- **Our home Decore Product are sells good in 11Th and 12Th month than summer time**
- **Our marketing team need to focus more on Electronic and clothing market for better result**