```
# This code makes the cell output to include every output, not just the last one.
from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = "all"


import pandas as pd
import numpy as np
import matplotlib.pyplot as plt



#Reading the Dataset
data_df = pd.read_csv(r"/content/train.csv")


#Data display
data_df.head(8)


data_df.shape
```

```
(550068, 12)
```

```
# Getting info on different data types
data_df.dtypes
```

```
User_ID                       int64
Product_ID                   object
Gender                       object
Age                          object
Occupation                    int64
City_Category                object
Stay_In_Current_City_Years   object
Marital_Status                int64
Product_Category_1            int64
Product_Category_2          float64
Product_Category_3          float64
Purchase                      int64
dtype: object
```

```
# Count of types of data type
data_df.dtypes.value_counts()
```

```
int64      5
object     5
float64    2
dtype: int64
```

```
# Using .info() to get the information about dataset
data_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 550068 entries, 0 to 550067
Data columns (total 12 columns):
 #   Column                      Non-Null Count   Dtype  
---  ------                      --------------   -----  
 0   User_ID                     550068 non-null  int64  
 1   Product_ID                  550068 non-null  object 
 2   Gender                      550068 non-null  object 
 3   Age                         550068 non-null  object 
 4   Occupation                  550068 non-null  int64  
 5   City_Category               550068 non-null  object 
 6   Stay_In_Current_City_Years  550068 non-null  object 
 7   Marital_Status              550068 non-null  int64  
 8   Product_Category_1          550068 non-null  int64  
 9   Product_Category_2          376430 non-null  float64
 10  Product_Category_3          166821 non-null  float64
 11  Purchase                    550068 non-null  int64  
dtypes: float64(2), int64(5), object(5)
memory usage: 50.4+ MB
```

```
#Findind out null values in each columns
data_df.isnull().sum()
```

```
User_ID                         0
Product_ID                      0
Gender                          0
Age                             0
```

```
Occupation                         0
City_Category                      0
Stay_In_Current_City_Years         0
Marital_Status                     0
Product_Category_1                 0
Product_Category_2            173638
Product_Category_3            383247
Purchase                           0
dtype: int64
```

```
# Calculating Null values in columns
null = pd.DataFrame({'Null Values' : data_df.isna().sum().sort_values(ascending=False),
                     'Percentage of Null Values' : (data_df.isna().sum().sort_values(ascending=False)) / (data_df.shape[0]) * (100)})
null[null['Null Values'] > 0]
```

| | Null Values | Percentage of Null Values |
|---|---|---|
| **Product_Category_3** | 383247 | 69.672659 |
| **Product_Category_2** | 173638 | 31.566643 |

```
#Checking for duplicates
data_df.duplicated().sum()
```

```
0
```

```
# Columns with "Object" Data type Category
# Using .discribe() to get info
cat_features=[col for col in data_df.columns if data_df[col].dtype=='O']
data_df[cat_features].head()
```

| | Product_ID | Gender | Age | City_Category | Stay_In_Current_City_Years |
|---|---|---|---|---|---|
| **0** | P00069042 | F | 0-17 | A | 2 |
| **1** | P00248942 | F | 0-17 | A | 2 |
| **2** | P00087842 | F | 0-17 | A | 2 |
| **3** | P00085442 | F | 0-17 | A | 2 |
| **4** | P00285442 | M | 55+ | C | 4+ |

```
data_df[cat_features].describe()
```

| | Product_ID | Gender | Age | City_Category | Stay_In_Current_City_Years |
|---|---|---|---|---|---|
| **count** | 550068 | 550068 | 550068 | 550068 | 550068 |
| **unique** | 3631 | 2 | 7 | 3 | 5 |
| **top** | P00265242 | M | 26-35 | B | 1 |
| **freq** | 1880 | 414259 | 219587 | 231173 | 193821 |

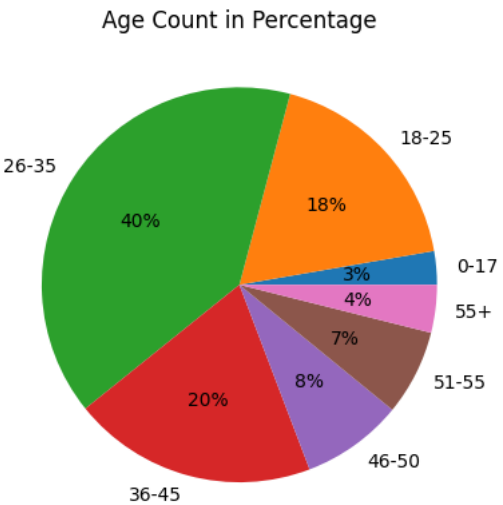
```
# Question 1: What age group has the highest and lowest purchases?

plt.hist(data_df['Age'])
```

```
(array([ 15102.,  21504.,      0., 219587.,      0.,  45701.,  38501.,
            0., 110013.,  99660.]),
 array([0. , 0.6, 1.2, 1.8, 2.4, 3. , 3.6, 4.2, 4.8, 5.4, 6. ]),
 <BarContainer object of 10 artists>)
```


```
#Pie chart for Age count
age_pie=data_df.groupby('Age')['Age'].agg('count')
display(age_pie.to_frame())
plt.title('Age Count in Percentage')
plt.pie(age_pie,labels=age_pie.index,radius=1.0,autopct='%.0f%%')
plt.show()
```

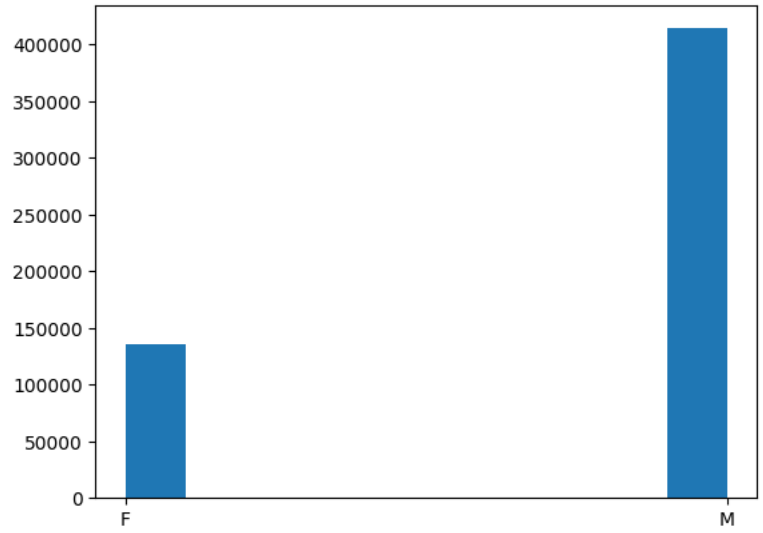
| Age | Age |
|---|---|
| **0-17** | 15102 |
| **18-25** | 99660 |
| **26-35** | 219587 |
| **36-45** | 110013 |
| **46-50** | 45701 |
| **51-55** | 38501 |
| **55+** | 21504 |

```
Text(0.5, 1.0, 'Age Count in Percentage')([<matplotlib.patches.Wedge at 0x7f90b497ec80>,
  <matplotlib.patches.Wedge at 0x7f90b48ef7c0>,
  <matplotlib.patches.Wedge at 0x7f90b497f9a0>,
  <matplotlib.patches.Wedge at 0x7f90b497ff40>,
  <matplotlib.patches.Wedge at 0x7f90b49b4610>,
  <matplotlib.patches.Wedge at 0x7f90b49b4ca0>,
  <matplotlib.patches.Wedge at 0x7f90b49b5330>],
 [Text(1.0959108846349965, 0.0947593422230203, '0-17'),
  Text(0.8110609466208436, 0.7430882456791397, '18-25'),
  Text(-0.9221569377090674, 0.5996887377923945, '26-35'),
  Text(-0.28804929537545926, -1.0616155629198838, '36-45'),
  Text(0.6430487042873294, -0.892461967769152, '46-50'),
  Text(0.9829467006757974, -0.49377705863128624, '51-55'),
  Text(1.0917144315273197, -0.134757560066147, '55+')],
 [Text(0.5977695734372708, 0.05168691393982924, '3%'),
  Text(0.44239687997500554, 0.40532086127953065, '18%'),
  Text(-0.5029946932958549, 0.32710294788676064, '40%'),
  Text(-0.15711779747752322, -0.5790630343199366, '20%'),
  Text(0.35075383870217963, -0.48679743696499195, '8%'),
  Text(0.5361527458231621, -0.26933294107161065, '7%'),
  Text(0.5954805990149016, -0.07350412367244381, '4%')])
```



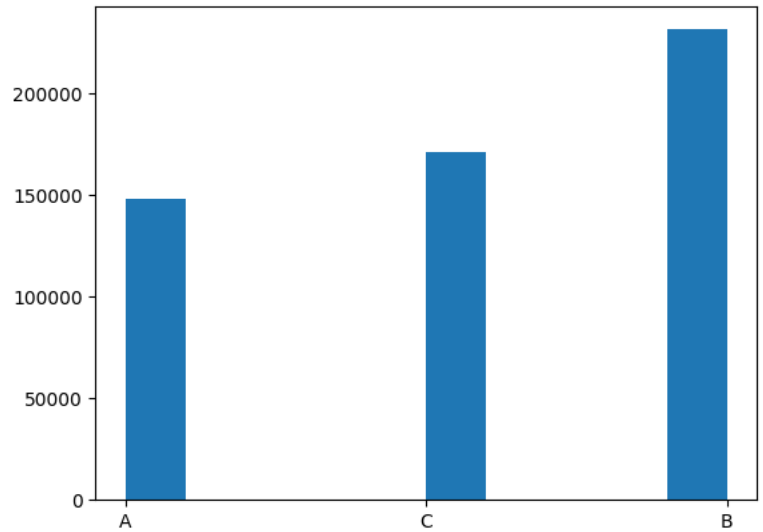
```
# Question 2: What gender is the top buyer on Black Friday?

plt.hist(data_df['Gender'])
```

```
(array([135809.,      0.,      0.,      0.,      0.,      0.,      0.,
             0.,      0., 414259.]),
 array([0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. ]),
 <BarContainer object of 10 artists>)
```




```
# Question 3: What city category accounts for the most and least purchases?

plt.hist(data_df['City_Category'])
```

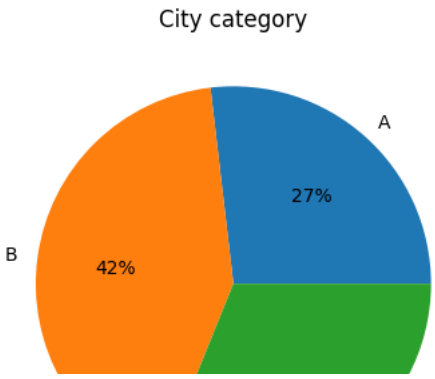
```
(array([147720.,      0.,      0.,      0.,      0., 171175.,      0.,
             0.,      0., 231173.]),
 array([0. , 0.2, 0.4, 0.6, 0.8, 1. , 1.2, 1.4, 1.6, 1.8, 2. ]),
 <BarContainer object of 10 artists>)
```




```
#Pie chart for City category

city_pie = data_df.groupby('City_Category')['City_Category'].agg('count')
display(city_pie.to_frame())
plt.title('City category')
plt.pie(city_pie,labels=city_pie.index,autopct='%.0f%%')
plt.show()
```

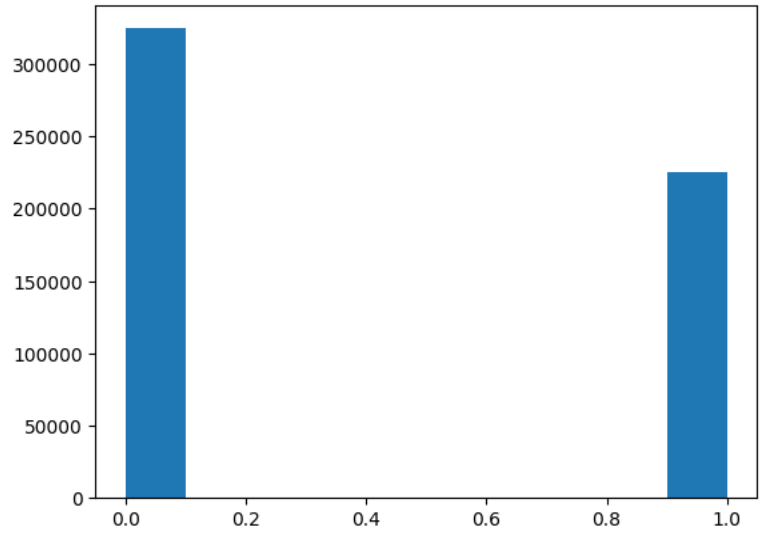
| City_Category | |
|---|---|
| **A** | 147720 |
| **B** | 231173 |
| **C** | 171175 |

```
Text(0.5, 1.0, 'City category')([<matplotlib.patches.Wedge at 0x7f90b471d120>,
  <matplotlib.patches.Wedge at 0x7f90b48a77c0>,
  <matplotlib.patches.Wedge at 0x7f90b471dd20>],
 [Text(0.7311977723577078, 0.8217967009541508, 'A'),
  Text(-1.0901451152986832, 0.14691367393956623, 'B'),
  Text(0.6148894666805983, -0.9120915215948724, 'C')],
 [Text(0.39883514855874963, 0.4482527459749913, '27%'),
  Text(-0.5946246083447362, 0.08013473123976338, '42%'),
  Text(0.33539425455305355, -0.4975044663244758, '31%')])
```




```
# Question 4: Does marital status have any impact on purchase?

plt.hist(data_df['Marital_Status'])
```

```
(array([324731.,      0.,      0.,      0.,      0.,      0.,      0.,
             0.,      0., 225337.]),
 array([0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. ]),
 <BarContainer object of 10 artists>)
```




```
# Question 5: Which occupation category has the most and least purchase?

plt.hist(data_df['Occupation'])
```

```
(array([117064.,  44238.,  84485.,  79488.,   7837.,  24516.,  38907.,
         39474.,  65414.,  48645.]),
 array([ 0.,  2.,  4.,  6.,  8., 10., 12., 14., 16., 18., 20.]),
 <BarContainer object of 10 artists>)
```