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

```
df = pd.read_csv("/content/d2beiqkhq929f0.cloudfront.net_public_assets_assets_000_001_293_original_walmart_data.csv_1641285094.txt")
```

```
df.head()
```

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Status	Product_Category	Purch
0	1000001	P00069042	F	0-17	10	A	2	0	3	8
1	1000001	P00248942	F	0-17	10	A	2	0	1	15
2	1000001	P00087842	F	0-17	10	A	2	0	12	1

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 550068 entries, 0 to 550067
Data columns (total 10 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            550068 non-null  int64
 9   Purchase                    550068 non-null  int64
dtypes: int64(5), object(5)
memory usage: 42.0+ MB
```

```
cols = ['Occupation', 'Marital_Status', 'Product_Category']
df[cols] = df[cols].astype('object')
```

```
df.dtypes
```

```
User_ID                      int64
Product_ID                  object
Gender                      object
Age                         object
Occupation                  object
City_Category               object
Stay_In_Current_City_Years  object
Marital_Status              object
Product_Category            object
Purchase                     int64
dtype: object
```

```
df.memory_usage()
```

```
Index                          128
User_ID                    4400544
Product_ID                 4400544
Gender                     4400544
Age                        4400544
Occupation                 4400544
City_Category              4400544
Stay_In_Current_City_Years 4400544
Marital_Status             4400544
Product_Category           4400544
Purchase                   4400544
dtype: int64
```

```
df.describe()
```

	User_ID	Purchase
count	5.500680e+05	550068.000000
mean	1.003029e+06	9263.968713
std	1.727592e+03	5023.065394
min	1.000001e+06	12.000000
25%	1.001516e+06	5823.000000
50%	1.003077e+06	8047.000000

```
'''
    there are no missing value in the dataset
'''
'''
    purchase amount might have outliers
'''
```

```
    ' \n    purchase amount might have outliers\n'
```

```
# How many users are there in the dataset

df['User_ID'].nunique()
```

```
    5891
```

```
#How many products are there

df['Product_ID'].nunique()
```

```
    3631
```

```
# checking null values
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            0
    Purchase                    0
    dtype: int64
```

```
'''
Value_counts for the following:

Gender
Age
Occupation
City_Category
Stay_In_Current_City_Years
Marital_Status
Product_Category
'''
```

```
    '\nValue_counts for the following:\n\nGender\nAge\nOccupation\nCity_Category\nStay_In_Current_City_Years\nMarital_Status\nProduct_
    Category\n'
```

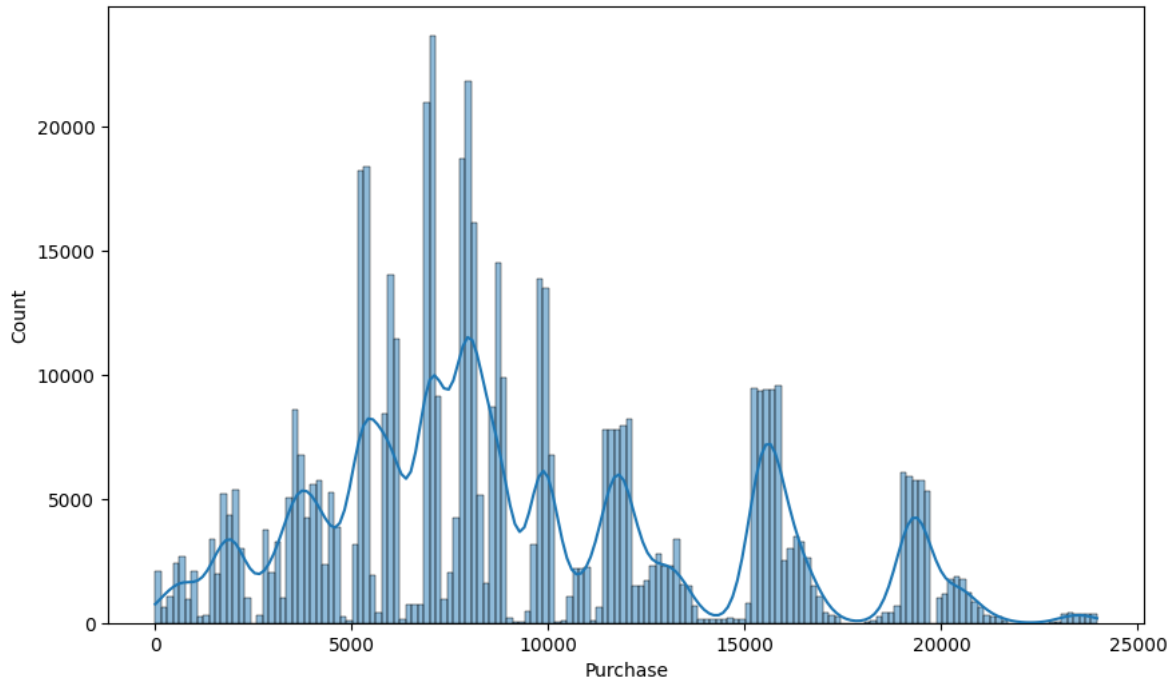
```
categorical_cols = ['Gender', 'Age', 'Occupation', 'City_Category', 'Stay_In_Current_City_Years', 'Marital_Status', 'Product_Category']
df[categorical_cols].melt().groupby(['variable', 'value'])[['value']].count()/len(df)
```

variable	value	value
Age	0-17	0.027455
	18-25	0.181178
	26-35	0.399200
	36-45	0.199999
	46-50	0.083082
	51-55	0.069993
	55+	0.039093
City_Category	A	0.268549
	B	0.420263
	C	0.311189
Gender	F	0.246895
	M	0.753105
Marital_Status	0	0.590347
	1	0.409653
Occupation	0	0.126599
	1	0.086218
	2	0.048336
	3	0.032087
	4	0.131453
	5	0.022137
	6	0.037005
	7	0.107501
	8	0.002811
	9	0.011437
	10	0.023506
	11	0.021063
	12	0.056682
	13	0.014049
	14	0.049647
	15	0.022115

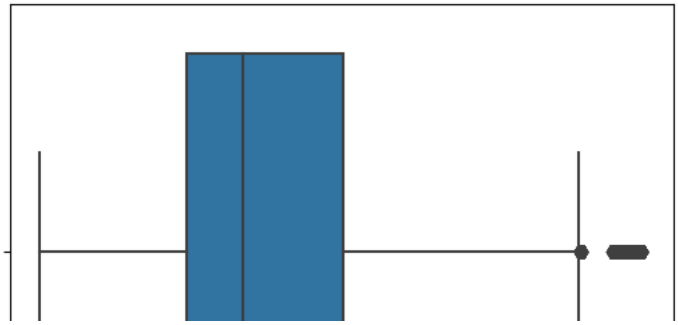
```
Observations
~ 80% of the users are between the age 18-50 (40%: 26-35, 18%: 18-25, 20%: 36-45)
75% of the users are Male and 25% are Female
60% Single, 40% Married
35% Staying in the city from 1 year, 18% from 2 years, 17% from 3 years
Total of 20 product categories are there
There are 20 differnent types of occupations in the city


'''
 Observation
 - 80% of the users are between the age 18- 50(40%: 26-35,18-25, 20%: 36-45)
 - 75% of the users are Male and 25% are Feamle
 - 60% single,40% married
 - 35% staying in the city from 1 year , 18% from 2 years, 17% from 3 year
 - total of 20 product categories are there
 - there are 20 differnent types odf occupation in the city
 '''
```

```
'\n Observation\n - 80% of the users are between the age 18- 50(40%: 26-35,18-25, 20%: 36-45)\n - 75% of the users are Male and 2
5% are Feamle\n - 60% single,40% married\n - 35% staying in the city from 1 year , 18% from 2 years, 17% from 3 year\n - total of
20 product categories are there \n - there are 20 differnent types odf occupation in the city\n '
```

```
from typing import Tuple
plt.figure(figsize=(10, 6))
sns.histplot(data=df, x ='Purchase', kde= True)
plt.show()
```




```
sns.boxplot(data=df, x='Purchase', orient='h')
plt.show()
```



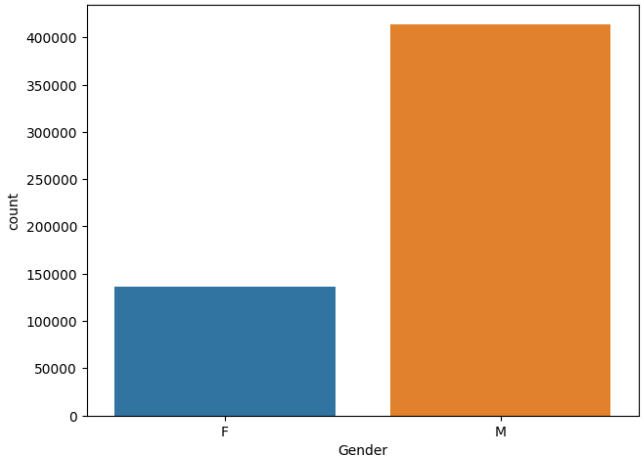
```
'''
observation
     -purchase is having outlier
'''
```

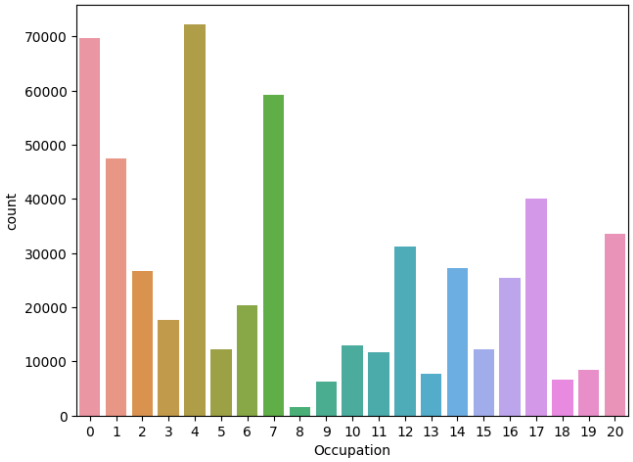
```
     '\nobservation\n     -purchase is having outlier\n'
```

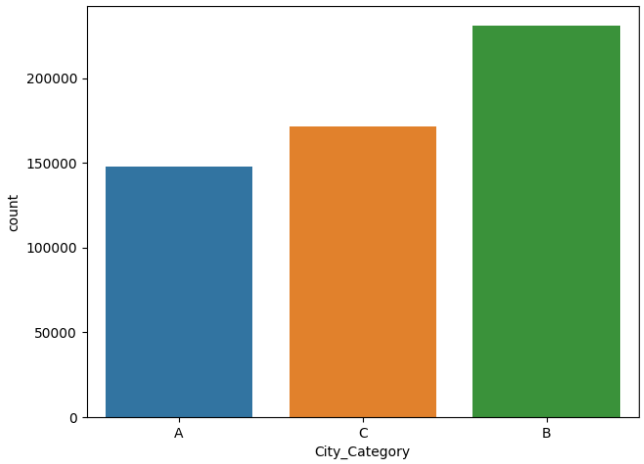
```
'''
Understanding the distribution of data for the categorical variables
Gender
Age
Occupation
City_Category
Stay_In_Current_City_Years
Marital_Status
Product_Category
'''
```

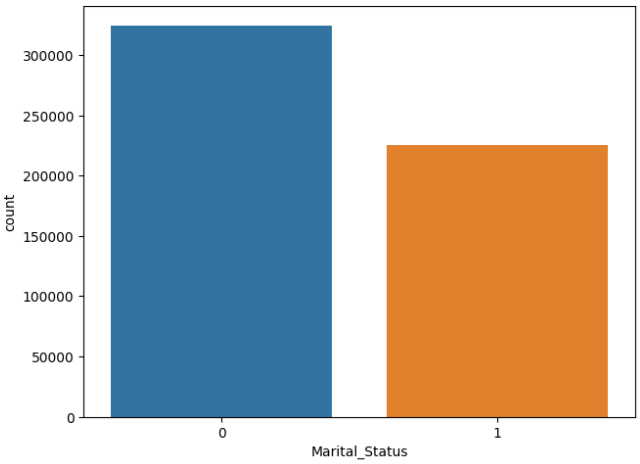
```
     '\nUnderstanding the distribution of data for the categorical variables\nGender\nAge\nOccupation\nCity_Category\nStay_In_Current_C
     ity_Years\nMarital_Status\nProduct_Category\n'
```

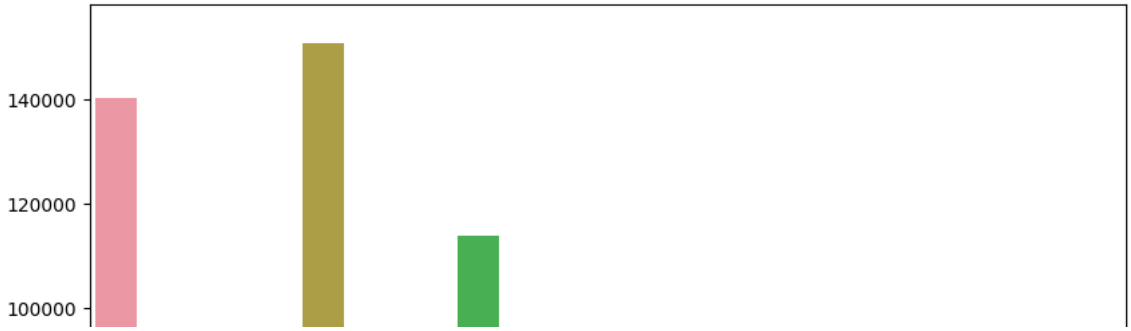
```
categorical_cols = ['Gender', 'Occupation','City_Category','Marital_Status','Product_Category']

fig, axs = plt.subplots(nrows=2, ncols=2, figsize=(16, 12))
sns.countplot(data=df, x='Gender', ax=axs[0,0])
sns.countplot(data=df, x='Occupation', ax=axs[0,1])
sns.countplot(data=df, x='City_Category', ax=axs[1,0])
sns.countplot(data=df, x='Marital_Status', ax=axs[1,1])
plt.show()

plt.figure(figsize=(10, 8))
sns.countplot(data=df, x='Product_Category')
plt.show()
```







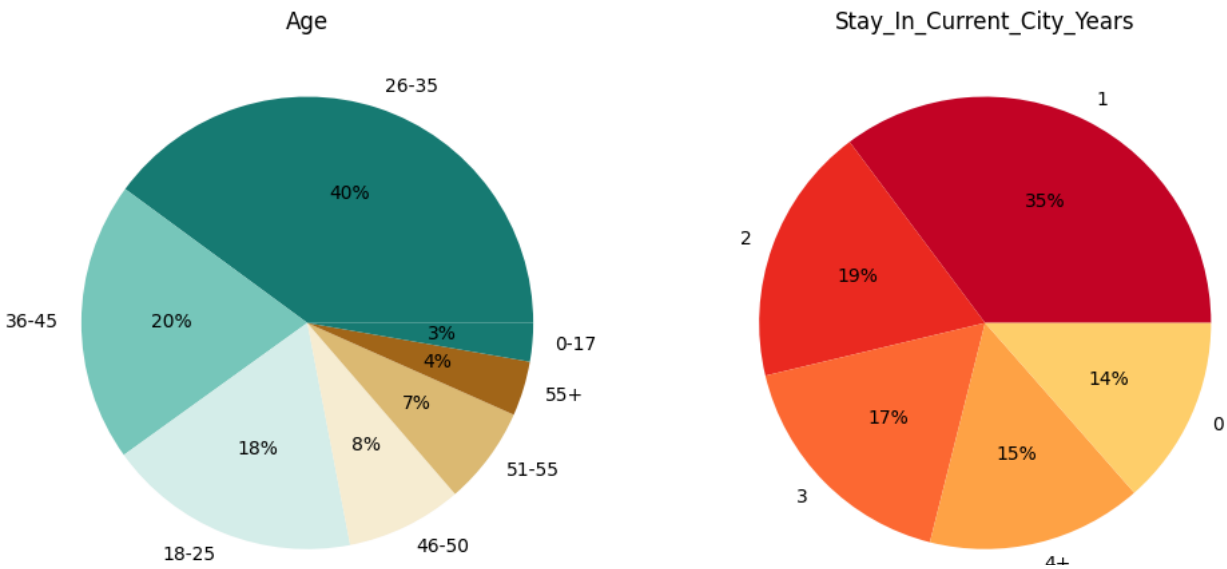




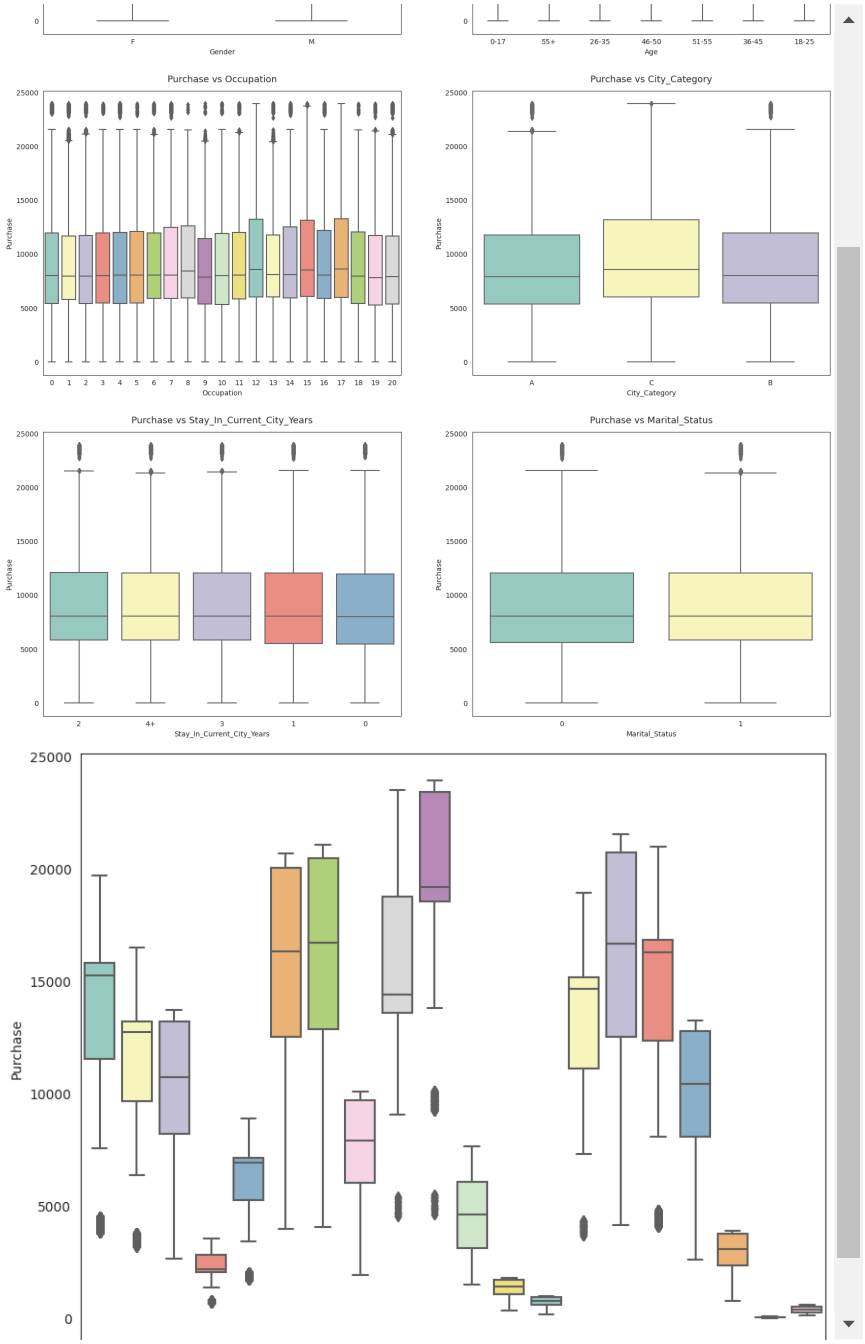
```
'''
Observations
-Most of the users are Male
-There are 20 different types of Occupation and Product_Category
-More users belong to B City_Category
-More users are Single as compare to Married
-Product_Category - 1, 5, 8, & 11 have highest purchasing frequency.
'''
```

```
'\nObservations\n-Most of the users are Male\n-There are 20 different types of Occupation and Product_Category\n-More users belong
to B City_Category\n-More users are Single as compare to Married\n-Product_Category - 1, 5, 8, & 11 have highest purchasing freque
ncy.\n'
```

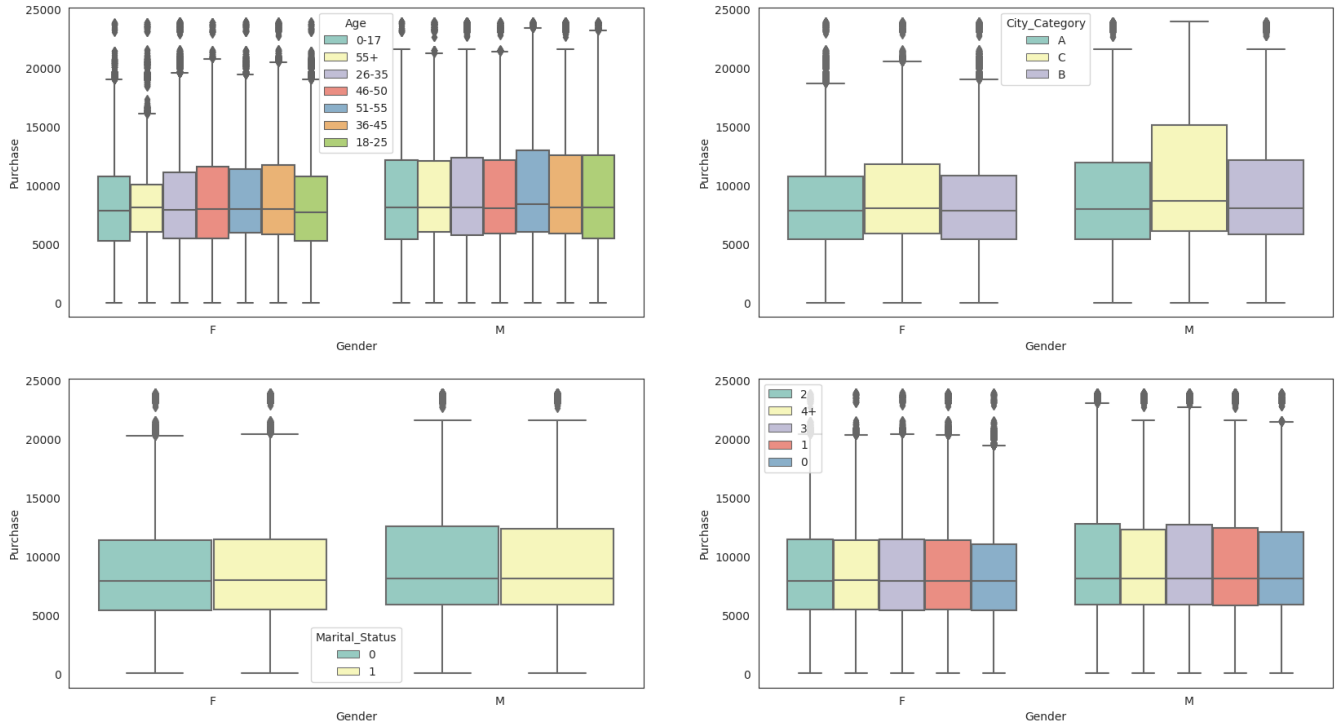
```
fig, axs = plt.subplots(nrows=1, ncols=2, figsize=(12, 8))

data = df['Age'].value_counts(normalize=True)*100
palette_color = sns.color_palette('BrBG_r')
axs[0].pie(x=data.values, labels=data.index, autopct='%.0f%%', colors=palette_color)
axs[0].set_title("Age")

data = df['Stay_In_Current_City_Years'].value_counts(normalize=True)*100
palette_color = sns.color_palette('YlOrRd_r')
axs[1].pie(x=data.values, labels=data.index, autopct='%.0f%%', colors=palette_color)
axs[1].set_title("Stay_In_Current_City_Years")


plt.show()
```




```
attrs = ['Gender', 'Age', 'Occupation', 'City_Category', 'Stay_In_Current_City_Years', 'Marital_Status', 'Product_Category']
sns.set_style("white")

fig, axs = plt.subplots(nrows=3, ncols=2, figsize=(20, 16))
fig.subplots_adjust(top=1.3)
count = 0
for row in range(3):
    for col in range(2):
        sns.boxplot(data=df, y='Purchase', x=attrs[count], ax=axs[row, col], palette='Set3')
        axs[row,col].set_title(f"Purchase vs {attrs[count]}", pad=12, fontsize=13)
        count += 1
plt.show()

plt.figure(figsize=(10, 8))
sns.boxplot(data=df, y='Purchase', x=attrs[-1], palette='Set3')
plt.show()
```




```
fig, axs = plt.subplots(nrows=2, ncols=2, figsize=(20, 6))
fig.subplots_adjust(top=1.5)
sns.boxplot(data=df, y='Purchase', x='Gender', hue='Age', palette='Set3', ax=axs[0,0])
sns.boxplot(data=df, y='Purchase', x='Gender', hue='City_Category', palette='Set3', ax=axs[0,1])

sns.boxplot(data=df, y='Purchase', x='Gender', hue='Marital_Status', palette='Set3', ax=axs[1,0])
sns.boxplot(data=df, y='Purchase', x='Gender', hue='Stay_In_Current_City_Years', palette='Set3', ax=axs[1,1])
axs[1,1].legend(loc='upper left')

plt.show()
```




```
df.head(10)
```

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Status	Product_Category	Purch
0	1000001	P00069042	F	0-17	10	A	2	0	3	8
1	1000001	P00248942	F	0-17	10	A	2	0	1	15
2	1000001	P00087842	F	0-17	10	A	2	0	12	1
3	1000001	P00085442	F	0-17	10	A	2	0	12	1
4	1000002	P00285442	M	55+	16	C	4+	0	8	7
5	1000003	P00193542	M	26-35	15	A	3	0	1	15
6	1000004	P00184942	M	46-	7	B	2	1	1	19

Average amount spend per customer for Male and Female

```
amt_df = df.groupby(['User_ID', 'Gender'])[['Purchase']].sum()
amt_df = amt_df.reset_index()
amt_df
```

	User_ID	Gender	Purchase
0	1000001	F	334093

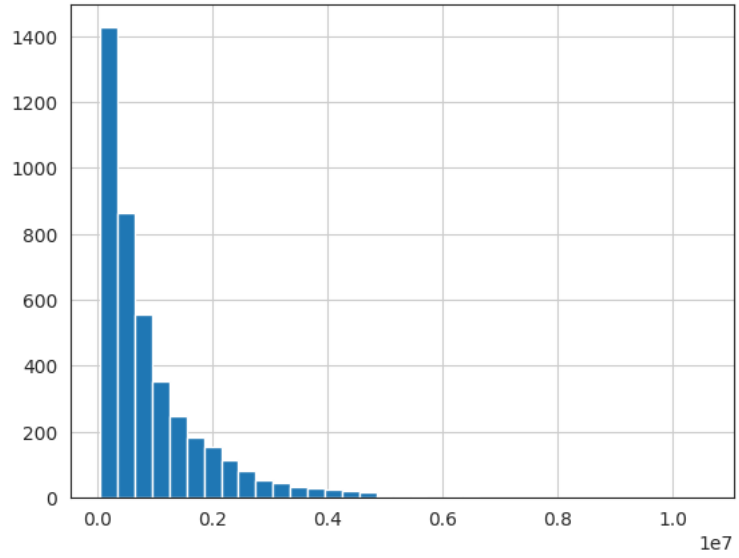
```
# Gender wise value counts in avg_amt_df
avg_amt_df['Gender'].value_counts()
```

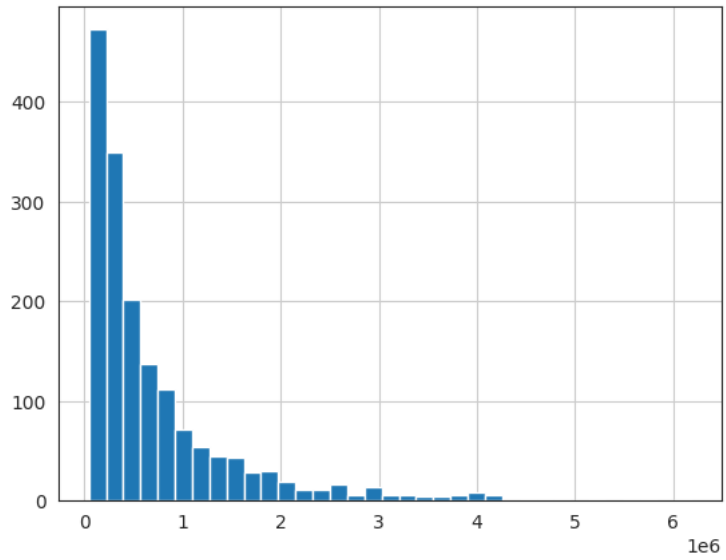
```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-47-8507529e438d> in <cell line: 2>()
      1 # Gender wise value counts in avg_amt_df
----> 2 avg_amt_df['Gender'].value_counts()

NameError: name 'avg_amt_df' is not defined
```

SEARCH STACK OVERFLOW

5889	1006039	F	590319

```
# histogram of average amount spend for each customer - Male & Female
amt_df[amt_df['Gender']=='M']['Purchase'].hist(bins=35)
plt.show()

amt_df[amt_df['Gender']=='F']['Purchase'].hist(bins=35)
plt.show()
```






```
male_avg = amt_df[amt_df['Gender']=='M']['Purchase'].mean()
female_avg = amt_df[amt_df['Gender']=='F']['Purchase'].mean()

print("Average amount spend by Male customers: {:.2f}".format(male_avg))
print("Average amount spend by Female customers: {:.2f}".format(female_avg))
```

```
Average amount spend by Male customers: 925344.40
Average amount spend by Female customers: 712024.39
```

```
'''
```

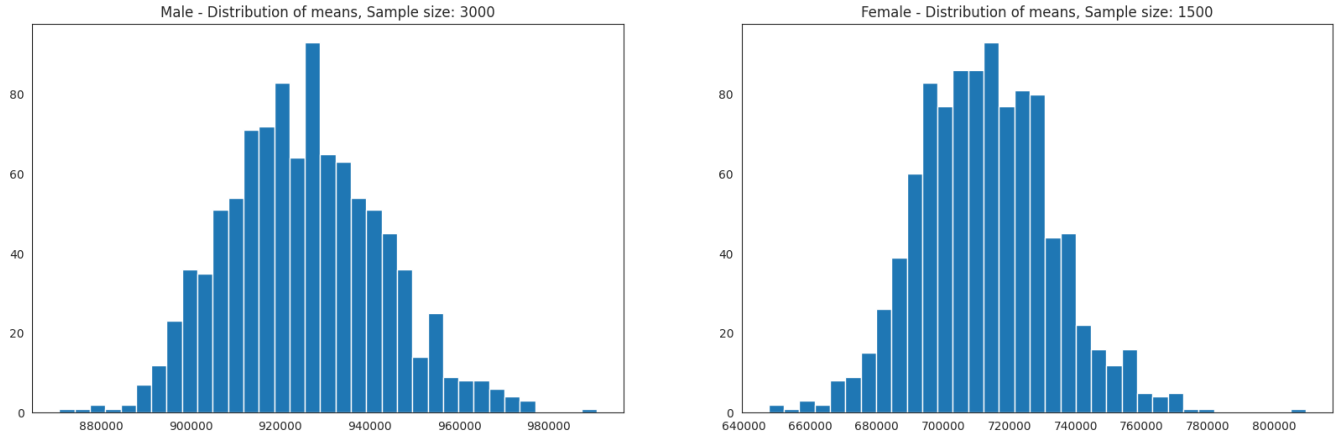
```
Observation

Male customers spend more money than female customers
'''
```

```
    '\nObservation\n\nMale customers spend more money than female customers\n'
```

```
male_df = amt_df[amt_df['Gender']=='M']
female_df = amt_df[amt_df['Gender']=='F']
```

```
genders = ["M", "F"]

male_sample_size = 3000
female_sample_size = 1500
num_repitions = 1000
male_means = []
female_means = []

for _ in range(num_repitions):
    male_mean = male_df.sample(male_sample_size, replace=True)['Purchase'].mean()
    female_mean = female_df.sample(female_sample_size, replace=True)['Purchase'].mean()

    male_means.append(male_mean)
    female_means.append(female_mean)
```

```
fig, axis = plt.subplots(nrows=1, ncols=2, figsize=(20, 6))

axis[0].hist(male_means, bins=35)
axis[1].hist(female_means, bins=35)
axis[0].set_title("Male - Distribution of means, Sample size: 3000")
axis[1].set_title("Female - Distribution of means, Sample size: 1500")

plt.show()
```



Doing the same activity for married vs unmarried

```
amt_df
```

	User_ID	Gender	Purchase
0	1000001	F	334093

```
amt_df = df.groupby(['User_ID', 'Marital_Status'])[['Purchase']].sum()
amt_df = amt_df.reset_index()
amt_df
```

	User_ID	Marital_Status	Purchase
0	1000001	0	334093
1	1000002	0	810472
2	1000003	0	341635
3	1000004	1	206468
4	1000005	1	821001
...	...	...	...
5886	1006036	1	4116058
5887	1006037	0	1119538
5888	1006038	0	90034
5889	1006039	1	590319
5890	1006040	0	1653299

5891 rows × 3 columns

```
amt_df['Marital_Status'].value_counts()
```

```
0    3417
1    2474
Name: Marital_Status, dtype: int64
```

```
#Calculating the average amount spent by Age
```

```
amt_df = df.groupby(['User_ID', 'Age'])[['Purchase']].sum()
amt_df = amt_df.reset_index()
amt_df
```

	User_ID	Age	Purchase
0	1000001	0-17	334093
1	1000002	55+	810472
2	1000003	26-35	341635
3	1000004	46-50	206468
4	1000005	26-35	821001
...	...	...	...
5886	1006036	26-35	4116058
5887	1006037	46-50	1119538
5888	1006038	55+	90034
5889	1006039	46-50	590319
5890	1006040	26-35	1653299

5891 rows × 3 columns

```
amt_df['Age'].value_counts()
```

```
26-35    2053
36-45    1167
18-25    1069
46-50     531
51-55     481
55+       372
0-17      218
Name: Age, dtype: int64
```