Business Case: Walmart - Confidence Interval and CLT

Description Walmart

Walmart is an American multinational retail corporation that operates a chain of supercenters, discount departmental stores, and grocery stores from the United States. Walmart has more than 100 million customers worldwide.

Walmart Inc. is an American multinational retail corporation that operates a chain of hypermarkets, discount department stores, and grocery stores in the United States, headquartered in Bentonville, Arkansas.

In [7]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from scipy.stats import norm
```

READING A FILE

In [8]:
```
df=pd.read_csv('WalMart.csv')
df
```

Out[8]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	M
0	1000001	P00069042	F	0-17	10	A	2	
1	1000001	P00248942	F	0-17	10	A	2	
2	1000001	P00087842	F	0-17	10	A	2	
3	1000001	P00085442	F	0-17	10	A	2	
4	1000002	P00285442	M	55+	16	C	4+	
...	...	...	...	...	...	...	...	
550063	1006033	P00372445	M	51-55	13	B	1	
550064	1006035	P00375436	F	26-35	1	C	3	
550065	1006036	P00375436	F	26-35	15	B	4+	
550066	1006038	P00375436	F	55+	1	C	2	
550067	1006039	P00371644	F	46-50	0	B	4+	

550068 rows × 10 columns

HEAD OF DATA

In [3]:
```
df.head()
```

Out[3]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_
0	1000001	P00069042	F	0-17	10	A	2	
1	1000001	P00248942	F	0-17	10	A	2	
2	1000001	P00087842	F	0-17	10	A	2	
3	1000001	P00085442	F	0-17	10	A	2	
4	1000002	P00285442	M	55+	16	C	4+	

TAIL OF DATA

In [4]:
```
df.tail()
```

Out[4]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	M
550063	1006033	P00372445	M	51-55	13	B	1	
550064	1006035	P00375436	F	26-35	1	C	3	
550065	1006036	P00375436	F	26-35	15	B	4+	
550066	1006038	P00375436	F	55+	1	C	2	
550067	1006039	P00371644	F	46-50	0	B	4+	

To check the first and last 5 rows together.

In [157…
```
df[['User_ID','Product_ID']].head().append(df[['User_ID','Product_ID']].tail())
```

```
C:\Users\HP\AppData\Local\Temp\ipykernel_4688\2954946926.py:1: FutureWarning: The fra
me.append method is deprecated and will be removed from pandas in a future version. U
se pandas.concat instead.
  df[['User_ID','Product_ID']].head().append(df[['User_ID','Product_ID']].tail())
```

Out[157]:

	User_ID	Product_ID
0	1000001	P00069042
1	1000001	P00248942
2	1000001	P00087842
3	1000001	P00085442
4	1000002	P00285442
550063	1006033	P00372445
550064	1006035	P00375436
550065	1006036	P00375436
550066	1006038	P00375436
550067	1006039	P00371644

LENGTH OF DATA

In [5]:
```
len(df)
```

Out[5]:
```
550068
```

SHAPE OF DATA

In [6]:
```
df.shape
```

Out[6]:
```
(550068, 10)
```

SIZE OF DATA

In [7]:
```
df.size
```

Out[7]:
```
5500680
```

ANALYTICAL DESCRIPTION OF DATA

In [3]:
```
df.describe()
```

Out[3]:

	User_ID	Occupation	Marital_Status	Product_Category	Purchase
count	5.500680e+05	550068.000000	550068.000000	550068.000000	550068.000000
mean	1.003029e+06	8.076707	0.409653	5.404270	9263.968713
std	1.727592e+03	6.522660	0.491770	3.936211	5023.065394
min	1.000001e+06	0.000000	0.000000	1.000000	12.000000
25%	1.001516e+06	2.000000	0.000000	1.000000	5823.000000
50%	1.003077e+06	7.000000	0.000000	5.000000	8047.000000
75%	1.004478e+06	14.000000	1.000000	8.000000	12054.000000
max	1.006040e+06	20.000000	1.000000	20.000000	23961.000000

Statement:The table presents key statistics for various attributes in a dataset, likely related to retail transactions. The dataset comprises 550,068 entries with User IDs ranging from 1,000,001 to 1,006,040. On average, customers have an occupation code of approximately 8, and a little over 40% are married. The most common product category is 5, and the average purchase amount is $9,263.97, with a significant standard deviation of $5,023.07. These statistics provide insights into customer demographics and their purchasing behaviors, aiding businesses in targeting and pricing strategies.

ANALYTICAL DESCRIPTION OF DATA WITH OBJECT DATA TYPE

In [9]:
```
df.describe(include="object")
```

Out[9]:

	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

Statement: The table summarizes information related to product purchases. It includes data on 550,068 transactions involving 3,631 unique product IDs. The majority of customers are males in the 26-35 age group, residing in City Category B, and have stayed in their current city for 1 year. The most frequently purchased product has ID P00265242, with 1,880 occurrences. This data provides insights into the most common customer demographics and product preferences.

CHECKING NULL VALUES

In [10]:
```
df.isna().sum()
```

Out[10]:
```
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
```

INFORMATION OF DATA

In [11]:
```
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
```

COLUMNS INCLUDES

In [12]:
```
df.keys()
```

Out[12]:
```
Index(['User_ID', 'Product_ID', 'Gender', 'Age', 'Occupation', 'City_Category',
       'Stay_In_Current_City_Years', 'Marital_Status', 'Product_Category',
       'Purchase'],
      dtype='object')
```

In [13]:
```
columns =['Age','Gender','City_Category','Stay_In_Current_City_Years','Marital_Status'
df[columns].melt().groupby(['variable','value'])[['value']].count()/len(df)
```

Out[13]:

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
Stay_In_Current_City_Years	**0**	0.135252
	1	0.352358
	2	0.185137
	3	0.173224
	4+	0.154028

Statement:The provided dataset offers insights into the purchasing behavior of a population. Notably, the majority of shoppers are male (75.31%), with a significant portion falling into the 26-35 age group (39.92%). Shoppers from City Category B (42.03%) dominate, while most customers are unmarried (59.03%). Interestingly, a substantial proportion of customers have been staying in their current city for less than a year (13.53%). These demographics can inform marketing and product strategies for businesses targeting this customer base.

In [14]:
```
df[:3]
```

Out[14]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_
0	1000001	P00069042	F	0-17	10	A	2	
1	1000001	P00248942	F	0-17	10	A	2	
2	1000001	P00087842	F	0-17	10	A	2	

Observing the users data with categories

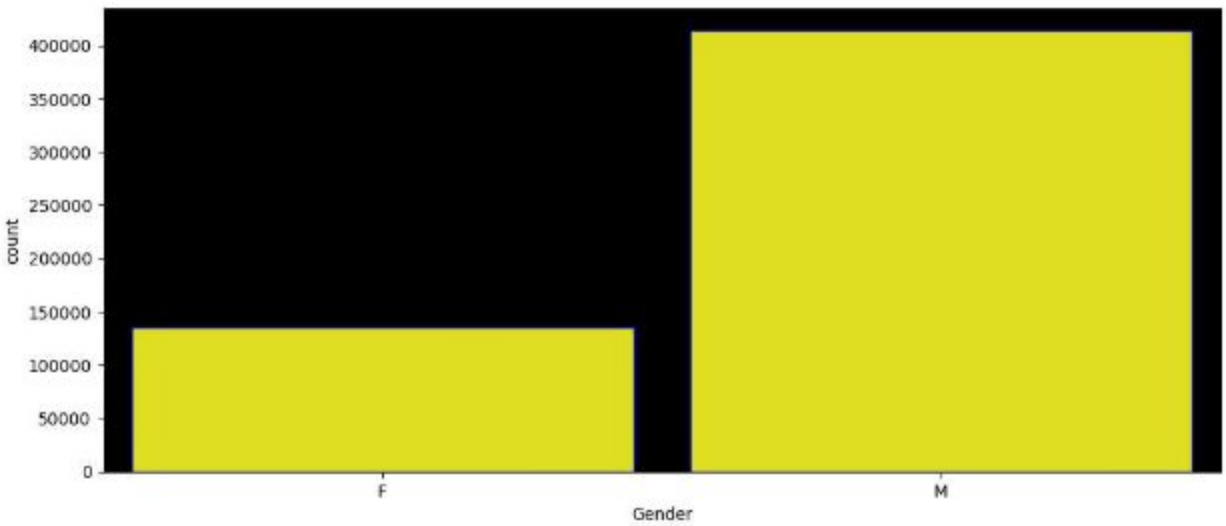
Gender Contribution

In [4]:
```
Genders = df.groupby(by=['User_ID'])['Gender'].unique()
Genders.value_counts()/len(Genders)*100
```

Out[4]:
```
[M]    71.719572
[F]    28.280428
Name: Gender, dtype: float64
```

Statement: The provided data shows that the majority of customers in the dataset are male (71.72%), while females represent a smaller portion (28.28%). This gender distribution can inform marketing and product strategies to better cater to the predominantly male customer base.

In [15]:
```
plt.figure(figsize=(12,5), dpi=100)
bar = sns.countplot(data=df, x='Gender', color="yellow",ec="b",width=0.9)
ax=plt.gca()
ax.set_facecolor('black')
plt.show()
```



Age Contribution

In [5]:
```
Age = df.groupby(by=['User_ID'])['Age'].unique()
Age.value_counts()/len(Age)*100
```

Out[5]:
```
[26-35]    34.849771
[36-45]    19.809879
[18-25]    18.146325
[46-50]     9.013750
[51-55]     8.164997
[55+]       6.314717
[0-17]      3.700560
Name: Age, dtype: float64
```

Statement: The data reveals that the largest customer age group is 26-35, comprising 34.85% of the dataset, followed by 36-45 at 19.81%. Younger age groups (18-25 and 0-17) make up a smaller portion, while older groups (46-55 and 55+) collectively represent a substantial portion, highlighting the diverse age demographics among customers.

Marital Status contribution

In [18]:
```
Marital_Status = df.groupby(by=['User_ID'])['Marital_Status'].unique()
Marital_Status.value_counts()/len(Marital_Status)
```

Out[18]:
```
[0]    0.580037
[1]    0.419963
Name: Marital_Status, dtype: float64
```

Statement: The data indicates that the majority of customers have a Marital_Status of 0 (unmarried), accounting for 58.00%, while 41.99% are married (Marital_Status 1). This shows a higher prevalence of unmarried individuals in the dataset and can guide marketing strategies and product offerings accordingly.

Cities Behaviour

In [6]:
```
city= df.groupby(by=['User_ID'])['City_Category'].unique()
city.value_counts()/len(city)*100
```

Out[6]:
```
[C]    53.284672
[B]    28.976405
[A]    17.738924
Name: City_Category, dtype: float64
```

Statement: The table displays three values: [C] at 0.532847, [B] at 0.289764, and [A] at 0.177389. These values represent proportions or percentages of something, with [C] being the highest and [A] the lowest, indicating varying degrees of contribution or importance.

Genders contributing Purchase

In [20]:
```
purchase_sample = pd.DataFrame(df.groupby(by=['Gender'])['Purchase'].sum())
purchase_sample
```

Out[20]:

Gender	Purchase
F	1186232642
M	3909580100

Statement: In purchase data, males (M) outspent females (F) significantly, with males contributing $3,909,580,100 compared to females' 1,186,232,642, indicating a gender-based spending disparity favoring males.

In [21]:
```
purchase_sample['percentage'] = purchase_sample['Purchase']/purchase_sample['Purchase'
purchase_sample
```

Out[21]:

Gender	Purchase	percentage
F	1186232642	23.278576
M	3909580100	76.721424

Statement:The table illustrates gender-based purchase percentages. Males (M) dominate, accounting for 76.72% of total purchases, significantly surpassing females (F) who contribute only 23.28%. This highlights a substantial gender disparity in purchasing behavior, with males being the primary consumers.

Average Contribution of Male and Female in Purchase

In [22]:
```
female = df[df['Gender']=="F"]['Purchase'].mean()
print("Female Average spending:",female)
male = df[df['Gender']=="M"]['Purchase'].mean()
print("Male Average Spending:",male)
```

```
Female Average spending: 8734.565765155476
Male Average Spending: 9437.526040472265
```

Statement: On average, male customers spend more (9437.53) compared to female customers (8734.57). This suggests that tailoring marketing strategies and product offerings to male preferences could potentially boost revenue and profitability.

Insights of Cities with Ages

In [23]:
```
pd.crosstab(index=df['City_Category'], columns=df['Age'],margins=True,normalize=True)
```

Out[23]:

City_Category \ Age	0-17	18-25	26-35	36-45	46-50	51-55	55+	All
A	0.004625	0.050057	0.134065	0.048389	0.013829	0.011088	0.006496	0.268549
B	0.009881	0.078621	0.166496	0.086531	0.037097	0.032252	0.009384	0.420263
C	0.012949	0.052499	0.098639	0.065079	0.032156	0.026653	0.023213	0.311189
All	0.027455	0.181178	0.399200	0.199999	0.083082	0.069993	0.039093	1.000000

Statement: The table presents age and city category-wise proportions of a population. In City Category A, the highest percentage of individuals falls in the 55+ age group (26.85%), while in Category B, the 18-25 age group is the largest (42.03%). In Category C, the 26-35 age group dominates (39.92%). Overall, the 26-35 age group is the most prevalent (39.92%) across all city categories, with Category B having the highest overall population share (42.03%).

Insights of Genders with Ages

In [24]:
```
pd.crosstab(index=df['Gender'],columns=df['Age'],normalize=True,margins=True)
```

Out[24]:

Gender \ Age	0-17	18-25	26-35	36-45	46-50	51-55	55+	All
F	0.009241	0.044773	0.092265	0.049394	0.023995	0.017987	0.009241	0.246895
M	0.018214	0.136405	0.306935	0.150605	0.059087	0.052006	0.029853	0.753105
All	0.027455	0.181178	0.399200	0.199999	0.083082	0.069993	0.039093	1.000000

Statement: This table displays the distribution of age groups by gender. Among females (F), the 18-25 age group is the largest (24.69%), while for males (M), the 26-35 age group dominates (30.69%). Overall, the 26-35 age group is the most prominent (39.92%) across both genders. The majority of the population is in the 18-25 and 26-35 age ranges, with males having a notably higher representation in the 26-35 category.

Contribution of segment Age with purchase value

In [25]:
```
Ages = pd.DataFrame(df.groupby('Age')['Purchase'].sum())
Ages
```

Out[25]:

Age	Purchase
0-17	134913183
18-25	913848675
26-35	2031770578
36-45	1026569884
46-50	420843403
51-55	367099644
55+	200767375

Statement: This table shows purchase amounts by age groups. The 26-35 age group has the highest spending, totaling $2,031,770,578, while the 0-17 and 55+ age groups exhibit the lowest spending, with 134,913,183 and $200,767,375, respectively.

In [26]:
```
Ages['percentage'] = Ages['Purchase']/Ages['Purchase'].sum()*100
Ages
```

Out[26]:

Age	Purchase	percentage
0-17	134913183	2.647530
18-25	913848675	17.933325
26-35	2031770578	39.871374
36-45	1026569884	20.145361
46-50	420843403	8.258612
51-55	367099644	7.203947
55+	200767375	3.939850

Statement: This table displays purchase percentages across various age groups. The 26-35 age group makes the largest contribution to purchases, accounting for 39.87% of total spending. Notably, the 18-25 age range also has a substantial share at 17.93%. Older age groups, such as 36-45, 46-50, and 51-55, exhibit moderate buying behavior. The 0-17 and 55+ age groups have the smallest purchasing shares at 2.65% and 3.94%, respectively, indicating varying spending patterns among different age demographics.

Contribution of segment Marital_status with purchase value

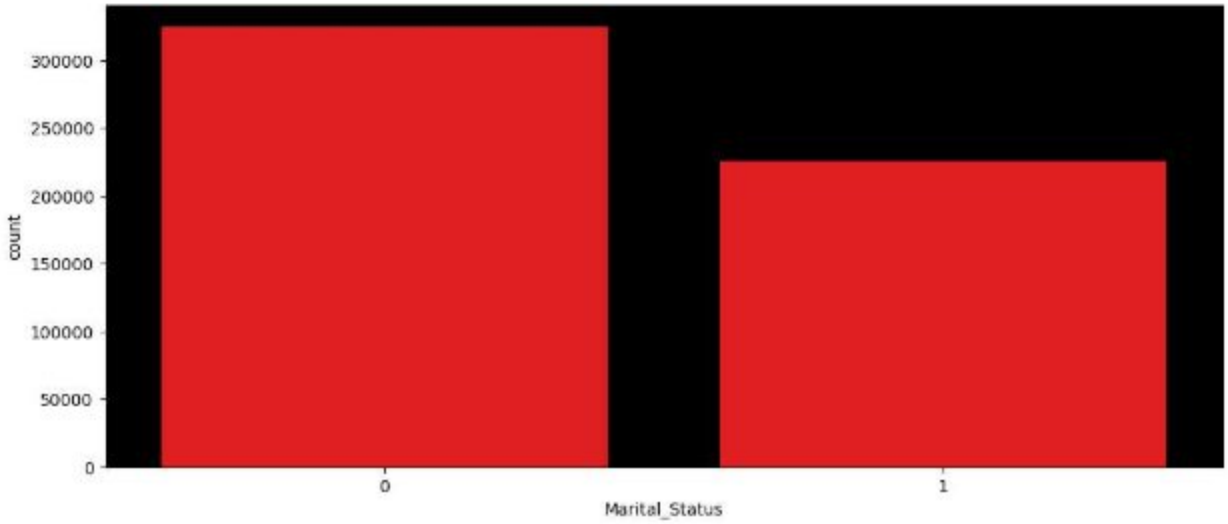
In [27]:
```
Marital_status = pd.DataFrame(df.groupby('Marital_Status')['Purchase'].sum())
Marital_status['percenatge'] = Marital_status['Purchase']/Marital_status['Purchase'].s
Marital_status
```

Out[27]:

Marital_Status	Purchase	percenatge
0	3008927447	59.047057
1	2086885295	40.952943

Statement: From a business perspective, this data suggests that the majority of purchases come from individuals with a marital status of 0 (single or unmarried), contributing 59.05% of total spending. This insight can help businesses tailor marketing and product strategies to effectively target and engage both single and married customers, optimizing revenue streams.

In [28]:
```
plt.figure(figsize=(12,5))
sns.countplot(data=df, x='Marital_Status', color="Red")
ax=plt.gca()
ax.set_facecolor('black')
plt.show()
```



Contribution of segment City Segments with purchase value

In [13]:
```
city = pd.DataFrame(df.groupby('City_Category')['Purchase'].sum())
city['percenatge'] = city['Purchase']/city['Purchase'].sum()*100
city["Average"] = city['Purchase'].mean()
city
```

Out[13]:

City_Category	Purchase	percenatge	Average
A	1316471661	25.834381	1.698604e+09
B	2115533605	41.515136	1.698604e+09
C	1663807476	32.650483	1.698604e+09

statement: In a business context, this data reveals significant variations in purchasing behavior across different city categories. Category B represents the highest purchasing power,

contributing 41.52% of total spending, making it a lucrative market for product offerings and marketing efforts. Meanwhile, Category C follows closely with 32.65%, suggesting a sizable customer base. Category A, at 25.83%, still holds potential for growth and market penetration strategies.

Top userID with maximum purchases

In [30]:
```
df.groupby(by=['User_ID'])['Purchase'].sum().sort_values(ascending=False)[:10].reset_
```

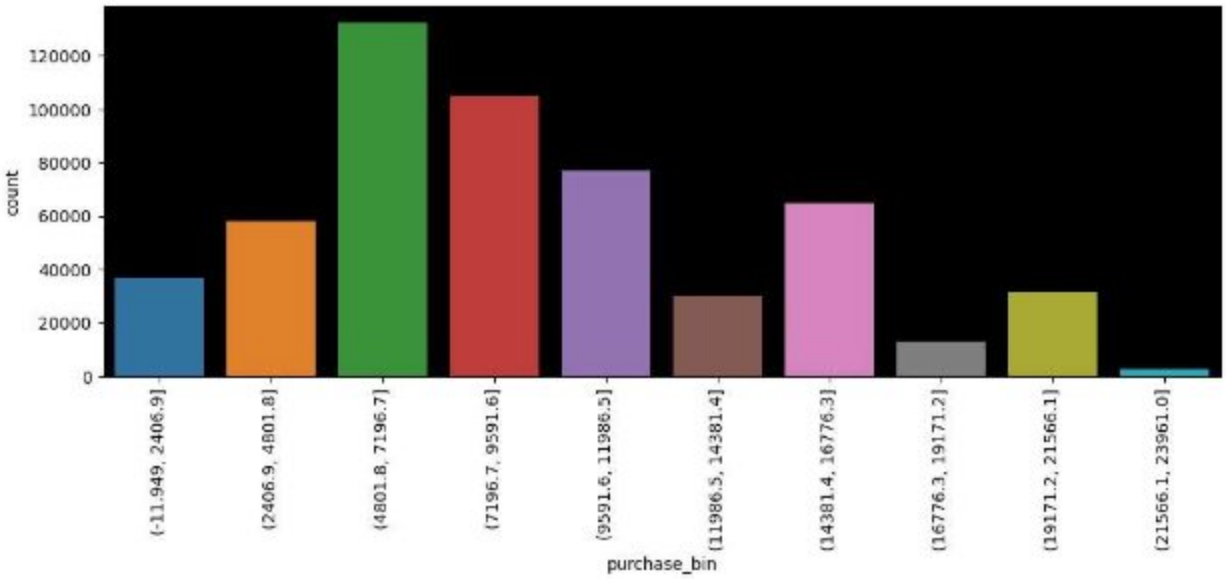
Out[30]:

	User_ID	Purchase
0	1004277	10536909
1	1001680	8699596
2	1002909	7577756
3	1001941	6817493
4	1000424	6573609
5	1004448	6566245
6	1005831	6512433
7	1001015	6511314
8	1003391	6477160
9	1001181	6387961

Statement: The provided dataset contains User IDs and their respective purchase values. Insights from this data suggest that User 1004277 made the highest purchase (10,536,909), indicating a potentially valuable customer. Additionally, users 1001680, 1002909, and 1001941 also made substantial purchases, highlighting potential high-value segments. Understanding and targeting these users may be crucial for maximizing revenue. Further analysis can help identify trends, such as user behavior or product preferences, to inform marketing and business strategies.

Optimising the Range of Products with Purchase

In [217…
```
plt.figure(figsize=(12,4))
df["purchase_bin"]=pd.cut(df["Purchase"],bins=10)
sns.countplot(data=df,x="purchase_bin")
plt.xticks(rotation=90)
ax=plt.gca()
ax.set_facecolor('black')
plt.show()
```



Business Insights: The graph illustrates the distribution of purchase amounts across product categories. It reveals that the highest concentration of purchases falls within the range of 4,801 to 7,196, indicating that customers within this range exhibit significant potential for making purchases. Conversely, the lowest purchase range, from 21,566 to 23,961, has the fewest number of transactions, suggesting that customers are less likely to make purchases within this price bracket.

Top userID with maximum number of purchases

In [31]:
```
df.groupby(by=['User_ID'])['Purchase'].count().sort_values(ascending=False)[:10].reset
```

Out[31]:

	User_ID	Purchase
0	1001680	1026
1	1004277	979
2	1001941	898
3	1001181	862
4	1000889	823
5	1003618	767
6	1001150	752
7	1001015	740
8	1005795	729
9	1005831	727

Statement: In this dataset, we observe user IDs and their corresponding purchase values. User 1001680 made the highest purchase at 1026, suggesting strong engagement. However, User 1004277, despite being second, has a lower purchase value (979), indicating potential for increased spending. Users 1001941, 1001181, and 1000889 also exhibit significant purchases,

warranting attention for retention and upselling efforts. Monitoring and analyzing these user behaviors can lead to improved revenue growth strategies.

Occupation category having the contribution in Purchase

In [32]:
```
occcupation = pd.DataFrame(df.groupby(by=['Occupation'])['Purchase'].sum())
occcupation['percentage']  = occcupation['Purchase']/occcupation['Purchase'].sum()*100
occcupation
```

Out[32]:

Occupation	Purchase	percentage
0	635406958	12.469198
1	424614144	8.332609
2	238028583	4.671062
3	162002168	3.179123
4	666244484	13.074352
5	113649759	2.230258
6	188416784	3.697482
7	557371587	10.937835
8	14737388	0.289206
9	54340046	1.066367
10	115844465	2.273327
11	106751618	2.094889
12	305449446	5.994126
13	71919481	1.411345
14	259454692	5.091527
15	118960211	2.334470
16	238346955	4.677310
17	393281453	7.717738
18	60721461	1.191595
19	73700617	1.446298
20	296570442	5.819885

Statement: This dataset presents purchase amounts and their respective percentages for different occupation categories. Notably, occupations 0, 4, and 7 contribute significantly to the total purchases, accounting for approximately 36% of the total. Occupations 8, 9, 13, and 18 make smaller contributions, suggesting potential growth opportunities through targeted marketing efforts. Occupations 2, 3, 5, and 19 have relatively lower percentages, indicating

room for improvement in engaging these segments to increase their contribution to overall sales.

Optimising the max sales of product

In [33]:
```
datasales = df.groupby(by=['Product_ID'])['Purchase'].count().sort_values(ascending=Fa
datasales
```

Out[33]:
```
Product_ID
P00265242    1880
P00025442    1615
P00110742    1612
P00112142    1562
P00057642    1470
P00184942    1440
P00046742    1438
P00058042    1422
P00145042    1406
P00059442    1406
Name: Purchase, dtype: int64
```

Statement: This dataset provides the sales quantities for different product IDs. Product P00265242 is the top seller with 1880 units sold, followed by P00025442 and P00110742 at 1615 and 1612 units, respectively. These insights highlight the best-performing products, which could be prioritized for marketing and inventory management. Additionally, it's important to investigate the characteristics of these top-selling products to identify customer preferences and potentially replicate their success with similar items.

Contributing the product category with purchase

In [35]:
```
pdp = pd.DataFrame(df.groupby(by=['Product_Category'])['Purchase'].sum())
pdp['percentage'] = pdp['Purchase']/pdp['Purchase'].sum()*100
pdp
```

Out[35]:

Product_Category	Purchase	percentage
1	1910013754	37.482024
2	268516186	5.269350
3	204084713	4.004949
4	27380488	0.537313
5	941835229	18.482532
6	324150302	6.361111
7	60896731	1.195035
8	854318799	16.765114
9	6370324	0.125011
10	100837301	1.978827
11	113791115	2.233032
12	5331844	0.104632
13	4008601	0.078665
14	20014696	0.392767
15	92969042	1.824420
16	145120612	2.847840
17	5878699	0.115363
18	9290201	0.182310
19	59378	0.001165
20	944727	0.018539

Statement: This dataset displays purchase amounts and their corresponding percentages for various product categories. Product category 1 dominates with 37.48% of total purchases, indicating its high popularity among customers. Categories 5 and 8 also make substantial contributions at 18.48% and 16.77%, respectively. Conversely, categories 4, 9, 12, and 13 have minimal shares, suggesting potential areas for growth or optimization in the product catalog.

In [36]:
```
df[:1]
```

Out[36]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_
0	1000001	P00069042	F	0-17	10	A	2	

In [37]:
```
stc = pd.DataFrame(df.groupby(by=['Stay_In_Current_City_Years'])['Purchase'].sum())
stc['percentage'] = stc['Purchase']/stc['Purchase'].sum()*100
stc
```

Out[37]:

Stay_In_Current_City_Years	Purchase	percentage
0	682979229	13.402754
1	1792872533	35.183250
2	949173931	18.626547
3	884902659	17.365290
4+	785884390	15.422160

Statement: The dataset reveals purchase amounts and percentages based on the number of years customers have stayed in their current city. Customers who have stayed for 1 year account for the largest share at 35.18%, followed by those with 0 years (13.40%). This suggests that relatively new residents contribute significantly to overall sales.

In [38]:
```
df.groupby(by=['City_Category'])['Marital_Status'].count()
```

Out[38]:
```
City_Category
A    147720
B    231173
C    171175
Name: Marital_Status, dtype: int64
```

Contributing Marriage status towards in Cities

In [39]:
```
pd.crosstab(index=df['City_Category'], columns=df['Marital_Status'],normalize=True)
```

Out[39]:

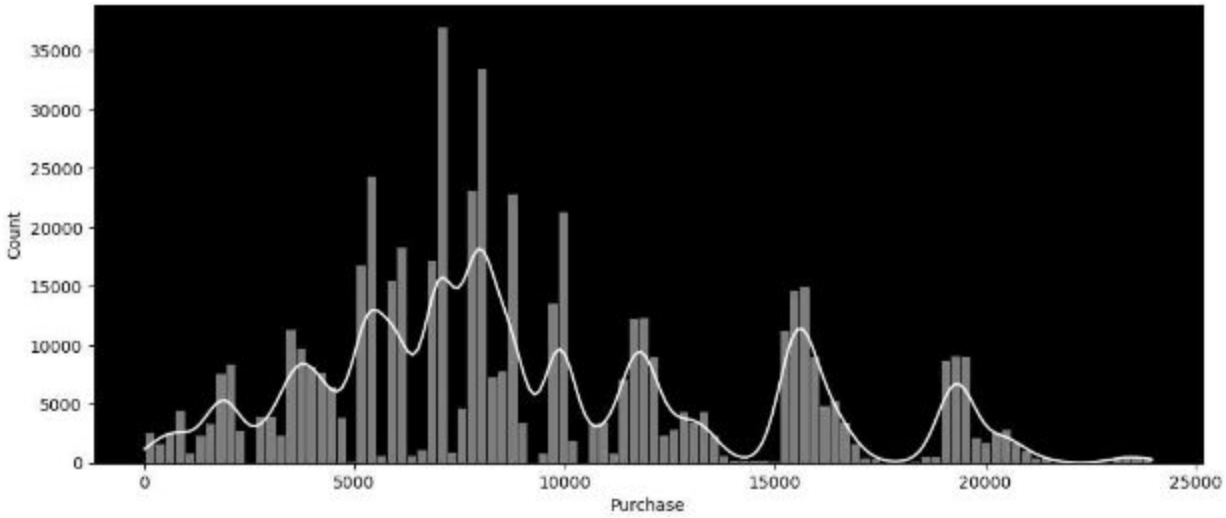
City_Category	Marital_Status 0	1
A	0.165749	0.102800
B	0.248553	0.171710
C	0.176046	0.135143

Statement: The table provides insights into the relationship between marital status and city category concerning customer proportions. In City Category B, a higher proportion of customers (24.86%) are married (Marital_Status 1) compared to City Category A (10.28%) and City Category C (13.51%). This suggests potential opportunities for businesses to tailor marketing strategies or product offerings to meet the needs and preferences of married individuals, especially in City Category B, where they represent a significant customer segment.

Data Distribution among Purchase Range

In [40]:
```
plt.figure(figsize=(12,5))
sns.histplot(data=df, x='Purchase', bins=100, kde=True,color="white")
ax=plt.gca()
```

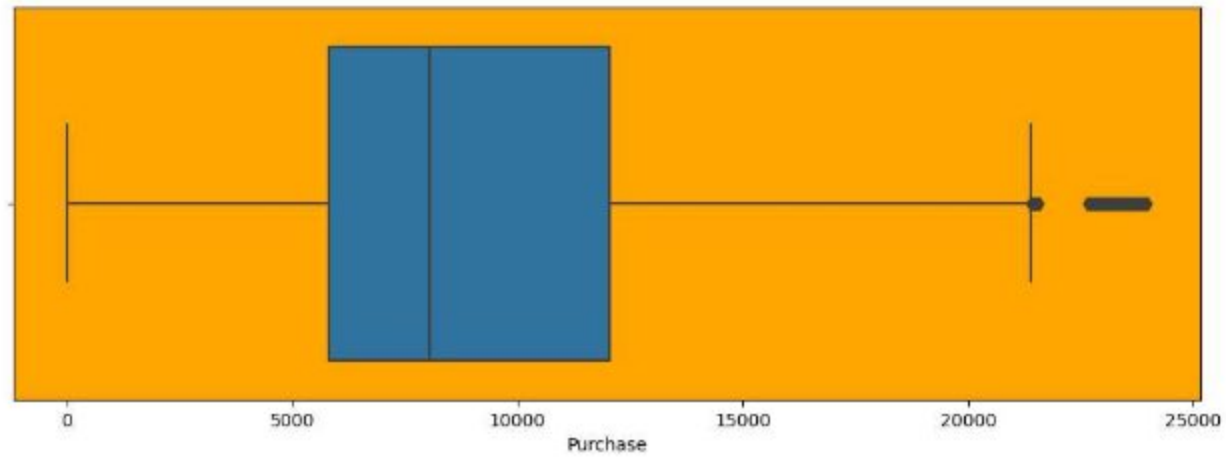
```
ax.set_facecolor('black')
plt.show()
```



Statement: The chart illustrates the distribution of product purchases within a predetermined budget range, typically ranging from 0 to approximately 23,000. Notably, the majority of purchases tend to cluster around the average range of approximately 9,000. This information is crucial for businesses to understand their customers' spending patterns and can guide pricing strategies and product offerings to align with these preferences and budget constraints.

Outlier Detection || Contributing with Purchase

In [41]:
```
plt.figure(figsize=(12,4))
sns.boxplot(data=df, x='Purchase')
ax=plt.gca()
ax.set_facecolor('orange')
plt.show()
```

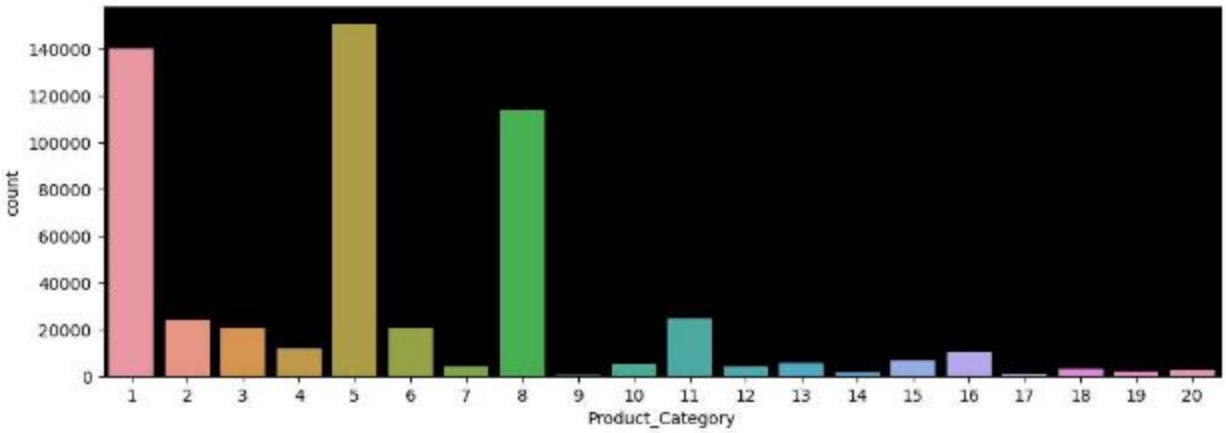


Statement: The boxplot visualizes the distribution of purchase amounts within a predefined range, typically spanning from 0 to around 25,000 purchase amount. Notably, the plot indicates that only a small fraction of purchases falls above the 22,000 mark but below 25,000. These outliers suggest that a few customers are making exceptionally high-value purchases, which

may warrant special attention in terms of customer engagement and targeted marketing strategies to capitalize on their substantial spending.

Category of Product Analysis based on users

In [42]:
```
plt.figure(figsize=(12,4))
sns.countplot(data=df, x='Product_Category')
ax=plt.gca()
ax.set_facecolor('black')
plt.show()
```

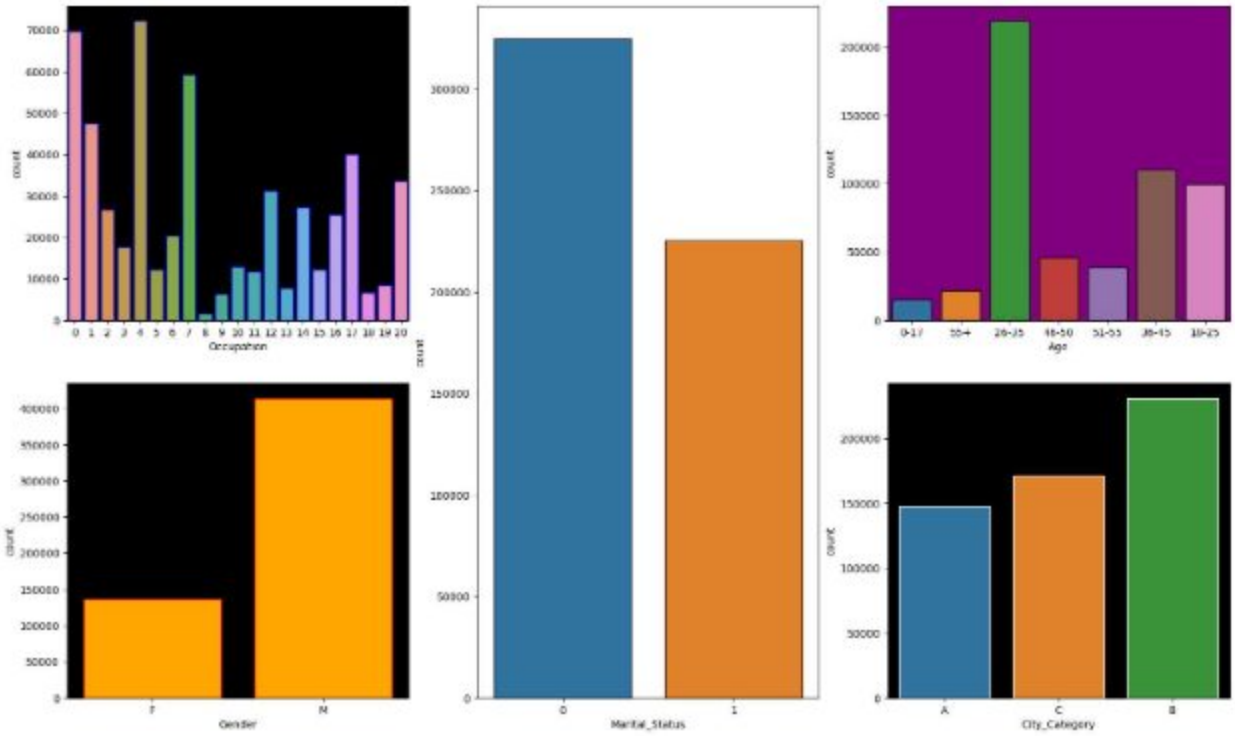


Statement: The graph indicates the most popular product categories among customers. Categories 1, 5, and 8 stand out as the highest-selling products, signifying their strong demand and profitability. Conversely, categories 2, 3, and 11 represent products with average sales. Businesses can focus resources on promoting and optimizing offerings within the top-performing categories to maximize revenue and customer satisfaction.

Analysis towards Keys Factors

In [30]:
```
plt.figure(figsize=(20,12))
plt.subplot(2,3,1)
sns.countplot(data=df, x='Occupation',ec='b')
ax=plt.gca()
ax.set_facecolor('black')
plt.subplot(2,3,3)
sns.countplot(data=df, x='Age', ec='k')
ax=plt.gca()
ax.set_facecolor('purple')
plt.subplot(2,3,4)
sns.countplot(data=df, x='Gender',ec='r', fc='orange')
ax=plt.gca()
ax.set_facecolor('black')
plt.subplot(2,3,6)
sns.countplot(data=df, x='City_Category',ec='w')
ax=plt.gca()
ax.set_facecolor('black')
plt.subplot(1,3,2)
sns.countplot(data=df, x='Marital_Status',ec='k')
ax=plt.gca()
```

```
ax.set_facecolor('white')
plt.show()
```



Insights: The set of graphs provides valuable insights into various customer segments:

1.Marital Status Impact: The data suggests that a larger portion of customers are categorized as "Single" (Marital Status 0) rather than "Engaged." Businesses may tailor their marketing strategies to target this single demographic effectively.

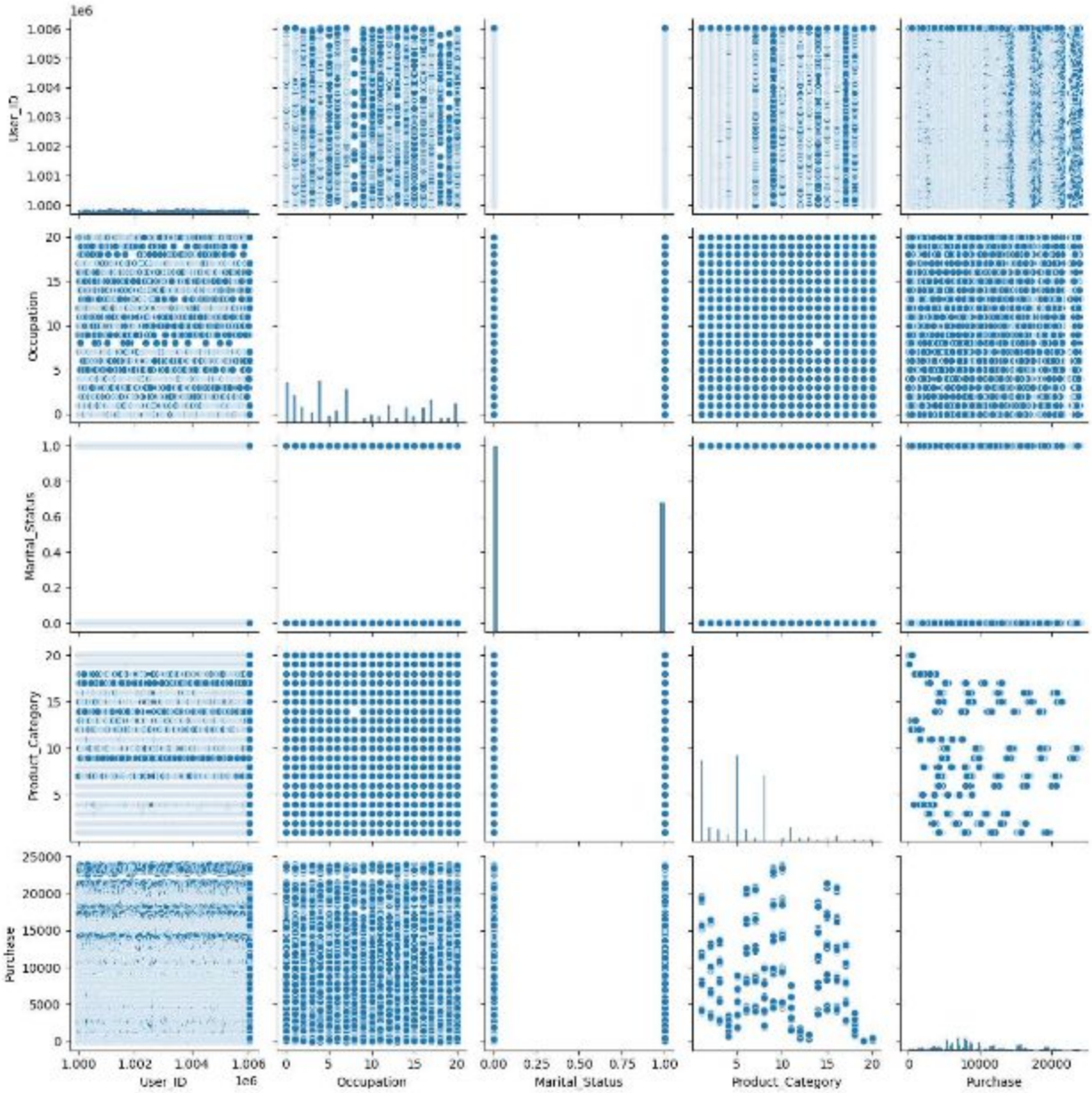
2.Age Influence: The most prominent age group contributing to product purchases and capacity lies between 26 and 35 years old. Understanding this can guide product development and marketing efforts toward this age group.

3.City Category Preferences: City Category "B" is the most populated, indicating that it's crucial for businesses to focus on serving the needs of customers residing in this category.

4.Gender-Based Trends: Males significantly contribute to product purchases compared to females. Recognizing this trend allows businesses to customize products and marketing strategies to cater to male customers effectively.

In [220…

```
sns.pairplot(df)
plt.show()
```



Insights:The provided graph displays a pair plot depicting the relationships between various categories. It is evident that each category appears to be self-contained and independent, with no discernible correlations or connections between them. This suggests that the columns representing these categories do not exhibit any meaningful interdependence in the context of our business analysis.

Inference after computing the average female and male expenses.

In [10]:
```
Avg_pur_gender = pd.DataFrame(df.groupby(by=['Gender'])['Purchase'].mean())
Avg_pur_gender["percentage"] = Avg_pur_gender['Purchase']/Avg_pur_gender['Purchase'].r
Avg_pur_gender
```

Out[10]:

Gender	Purchase	percentage
F	8734.565765	96.131649
M	9437.526040	103.868351

Business Insights: The table presents average purchase amounts and percentages based on gender. On average, male customers spend 103.87% more than female customers(96.13%). This significant spending difference suggests that businesses may benefit from tailoring their marketing strategies and product offerings to better target and engage male customers, potentially increasing overall revenue and profitability.

inference after computing the total female and male expenses.

In [100…

```
total_pur_gender = pd.DataFrame(df.groupby(by=['Gender'])['Purchase'].sum())
total_pur_gender["percentage"] = total_pur_gender['Purchase']/total_pur_gender['Purcha
total_pur_gender
```

Out[100]:

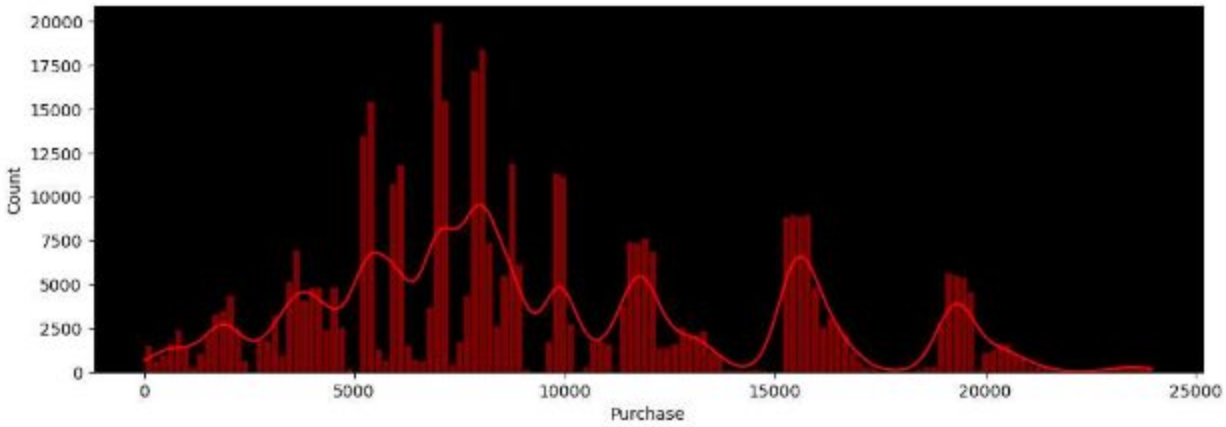
Gender	Purchase	percentage
F	1186232642	23.278576
M	3909580100	76.721424

Business Insights: The table provides a breakdown of purchase amounts and percentages by gender. Notably, male customers account for a substantial majority, contributing 76.72% of the total purchase value, while female customers account for 23.28%. This suggests that males have a more significant impact on overall sales. Businesses may consider tailoring their marketing strategies and product offerings to better target and engage male customers to maximize revenue and profitability.

Inference Computing the average male expenses.

In [32]:

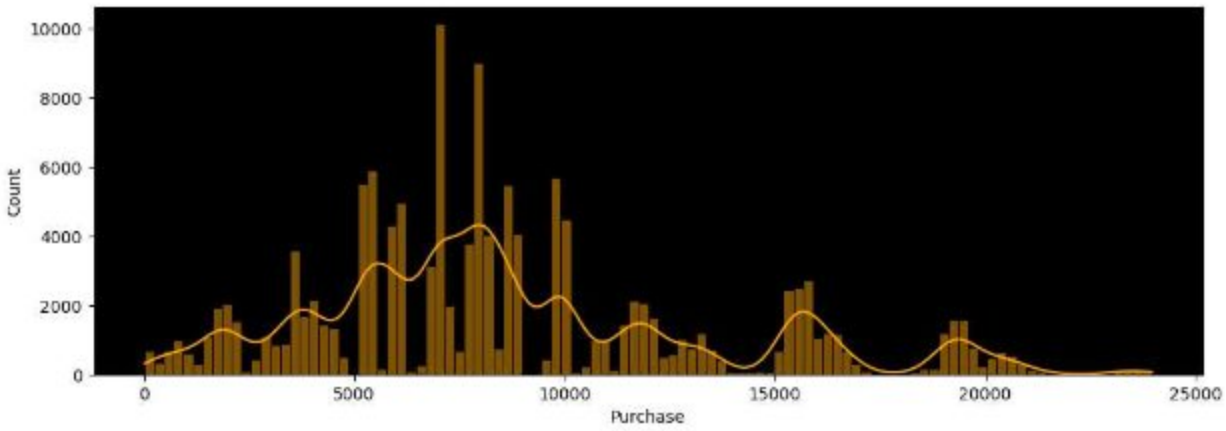
```
plt.figure(figsize=(12,4))
male = df[df['Gender']=="M"]
sns.histplot(data=male, x='Purchase',kde=True,color="Red",bins="auto")
ax=plt.gca()
ax.set_facecolor('black')
plt.show()
```



Business Insights: The histogram illustrates the distribution of purchase amounts among male customers. It reveals that a dense concentration of purchases falls below 5000, indicating a prevalence of lower-value transactions. The average purchase amount predominantly falls between 6000 and 8000. Additionally, there is another notable cluster of purchases between 15000 and 20000. These insights can guide pricing strategies, product recommendations, and promotions, optimizing revenue by catering to different spending patterns within the male customer segment.

Inference Computing the average female expenses.

In [62]:
```
plt.figure(figsize=(12,4))
female = df[df['Gender']=="F"]
sns.histplot(data=female, x='Purchase',kde=True,color="Orange")
ax=plt.gca()
ax.set_facecolor('black')
plt.show()
```

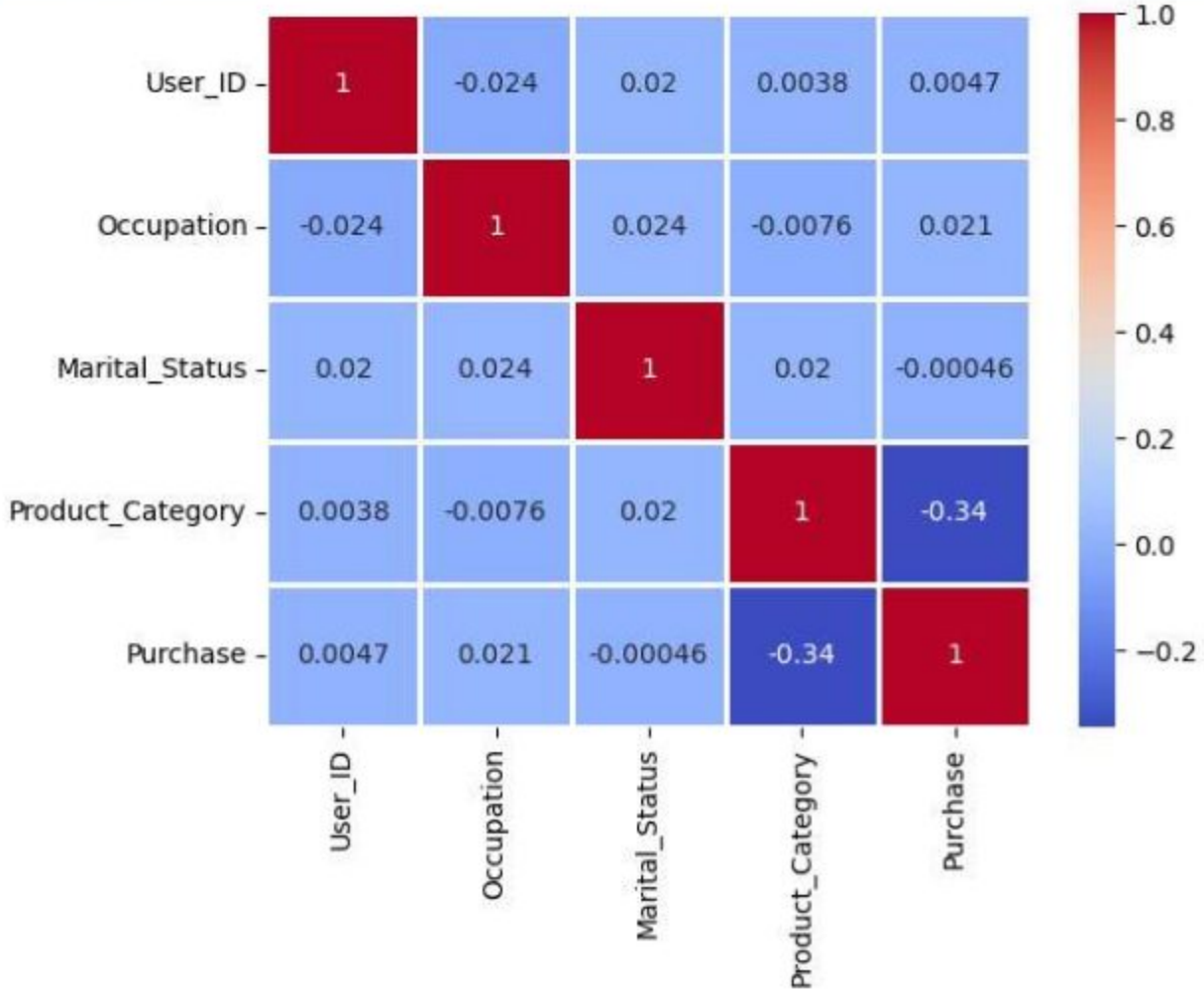


Business Insights: The histogram showcasing female customers' purchase behavior reveals that a significant portion of their purchases falls within lower price ranges, particularly below 5000 and even below 3000. The average purchase amount hovers between 6000 and 7000. Interestingly, there are notable spikes in spending at 11000, 15000, and 20000. Businesses can leverage these insights to tailor marketing strategies, product offerings, and promotions to cater to different spending preferences within the female customer segment, potentially optimizing sales and customer satisfaction.

showcasing correlation Among

In [4]:
```
sns.heatmap(data=df.corr(),annot=True,cmap='coolwarm', linewidths=1)
plt.show()
```

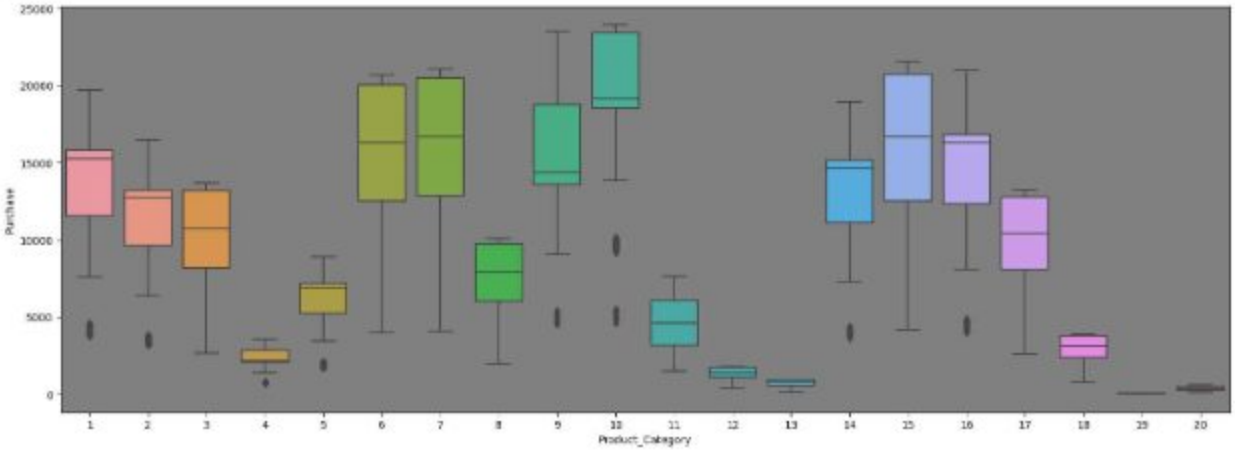
```
C:\Users\HP\AppData\Local\Temp\ipykernel_1420\3774175373.py:1: FutureWarning: The def
ault value of numeric_only in DataFrame.corr is deprecated. In a future version, it w
ill default to False. Select only valid columns or specify the value of numeric_only
to silence this warning.
  sns.heatmap(data=df.corr(),annot=True,cmap='coolwarm', linewidths=1)
```



Insights: There appears to be a concerning lack of positive correlation between our products and various categorical and numerical variables. Typically, strong relationships are expected between products and user-related factors, but our data suggests that the product-to-product correlations are quite weak, often falling below the 0.7 threshold. Furthermore, we've observed negative correlations between product category and purchase behavior, as well as between user ID and occupation. These findings indicate potential challenges in optimizing product recommendations and understanding user preferences.

Detecing outliers through all Segments with Purchase

In [10]:
```
plt.figure(figsize=(20,7))
sns.boxplot(data=df,x="Product_Category",y='Purchase')
ax=plt.gca()
ax.set_facecolor('grey')
plt.show()
```

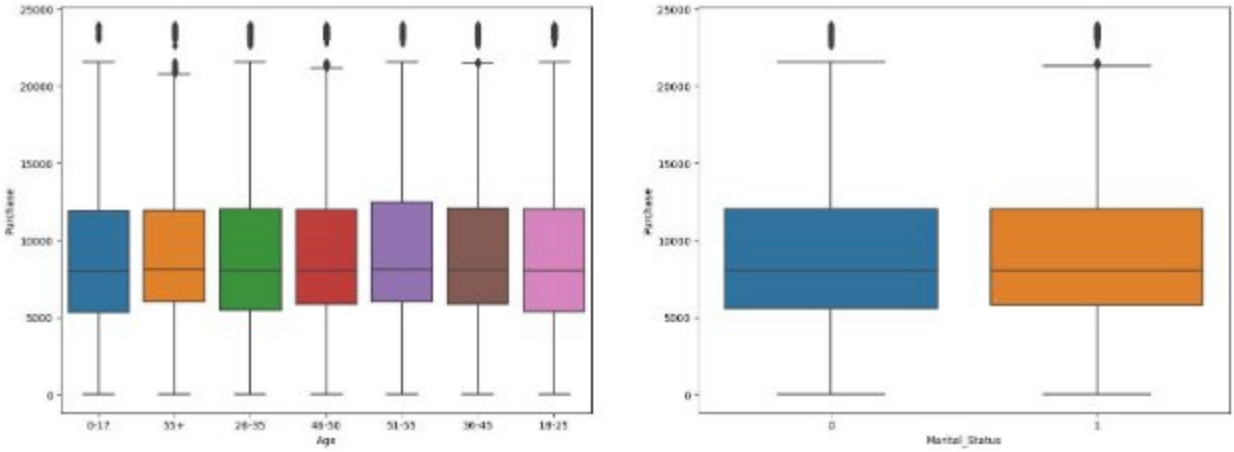


Insights: The boxplots presented above offer valuable insights into how specific product categories relate to customer purchasing behavior. We focused on four particular products (12, 13, 19, and 20) that exhibit minimal interest and consequently have the smallest customer segments making purchases. These products generally show low engagement from our customer base.

On the other hand, products 6, 7, and 15 stand out as having a more widespread appeal, attracting a larger portion of our customer population. The median purchase amount for these products hovers around 16,000, which is notably close to the overall average purchase amount. This suggests that these products are consistently purchased by a broad customer base.

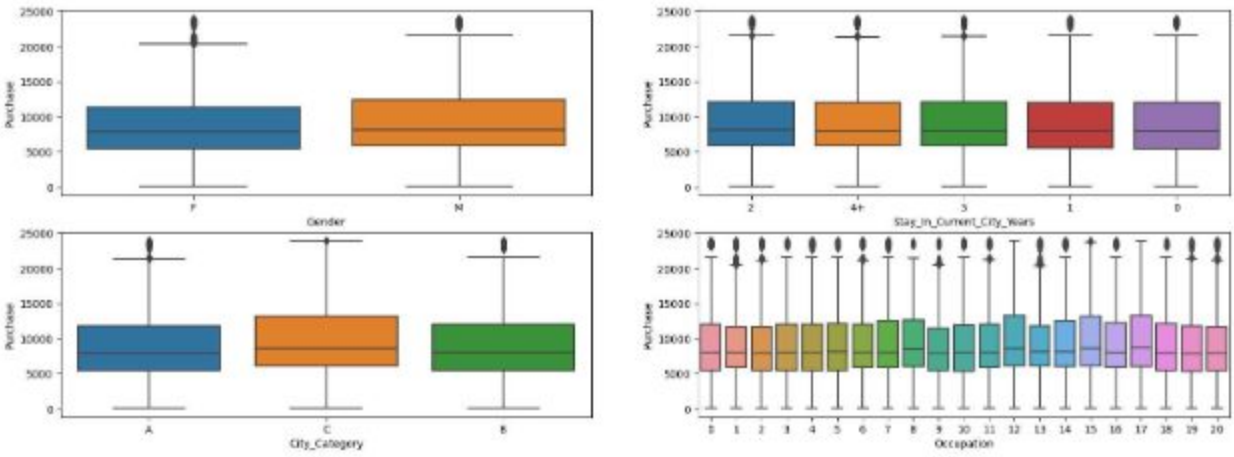
Business view: In contrast, products 1, 2, 4, 5, 8, 10, 14, and 16 have a distinct pattern of exceptional purchases. Despite being of relatively low interest, they still manage to achieve purchases that exceed expectations. Understanding the factors driving these exceptional purchases could be crucial in optimizing our product offerings and marketing strategies.

In [15]:
```
plt.figure(figsize=(20,7))
plt.subplot(1,2,1)
sns.boxplot(data=df, x="Age",y="Purchase")
plt.subplot(1,2,2)
sns.boxplot(data=df, x="Marital_Status",y="Purchase")
plt.show()
```



Insights: The first boxplot illustrates how age impacts purchasing behavior, with noticeable variations and outliers. Notably, customers aged 55+, 46-50, and 36-45 exhibit significant outliers, suggesting complex spending patterns among older demographics. The second boxplot indicates relatively even distribution between categories 0 and 1, but outliers are more prevalent in category 1. This highlights potential opportunities for targeted strategies to engage customers in category 1 and explore underlying factors behind these outliers.

In [24]:
```
plt.figure(figsize=(20,7))
plt.subplot(2,2,1)
sns.boxplot(data=df, x="Gender", y="Purchase")
plt.subplot(2,2,2)
sns.boxplot(data=df, x="Stay_In_Current_City_Years", y="Purchase")
plt.subplot(2,2,3)
sns.boxplot(data=df, x="City_Category", y="Purchase")
plt.subplot(2,2,4)
sns.boxplot(data=df, x="Occupation", y="Purchase")
plt.show()
```

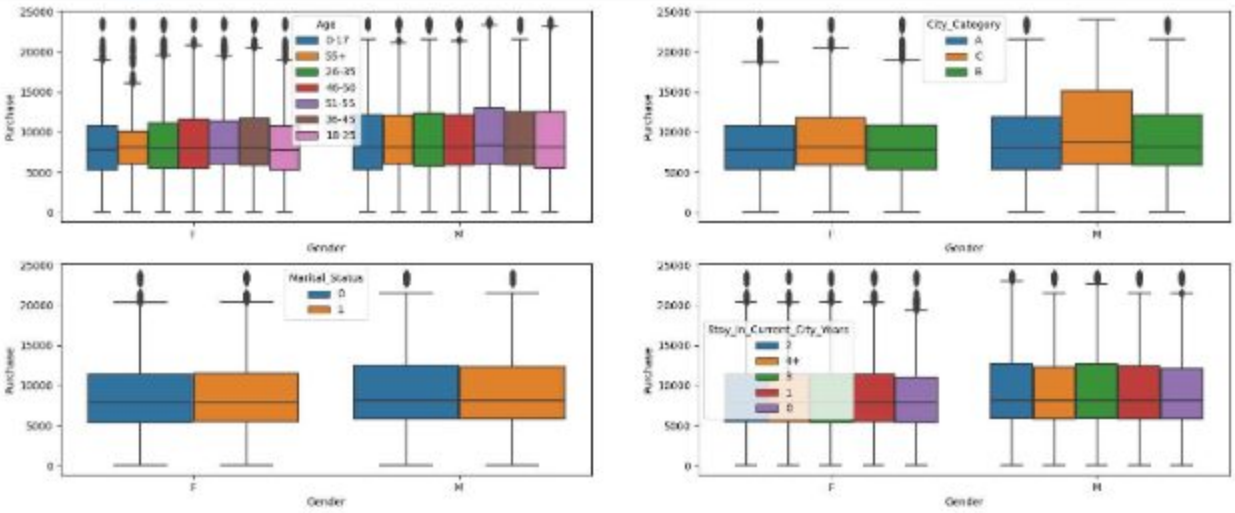


Insights: The gender-based analysis of purchase behavior reveals distinct patterns. Males display a wider range of purchasing habits compared to females, with notable outliers among females. Interestingly, current city years exhibit a relatively uniform influence across all purchase categories. However, category C stands out with a broad range of purchases and minimal outliers compared to other categories. Occupations demonstrate consistent influence across all levels, implying that various segments within this dataset share similar purchasing tendencies.

These insights emphasize the significance of gender and product category in understanding customer behavior for tailored marketing strategies.

Multivariate Analaysis through the Gender and Purchase with Category

In [9]:
```
plt.figure(figsize=(20,8))
plt.subplot(2,2,1)
sns.boxplot(data=df, x="Gender", y="Purchase", hue="Age")
plt.subplot(2,2,2)
sns.boxplot(data=df, x="Gender", y="Purchase", hue="City_Category")
plt.subplot(2,2,3)
sns.boxplot(data=df, x="Gender", y="Purchase", hue="Marital_Status")
plt.subplot(2,2,4)
sns.boxplot(data=df, x="Gender", y="Purchase", hue="Stay_In_Current_City_Years")
plt.show()
```



Insights: The boxplot pictorial represetation states that the major of the people who lies in age above 55+ well shown tendency a very shrink and small bunch of range fallsin female category, where as the female tends to have a high amount of urchase startegy where the outliers falls a big portion of it. The Male with the C city type tends to have a high portion of segmented values while female purchajsing powwrr is lower thsn male in all citiueds categories.

Average Spending of Customers

In [6]:
```
avgspent = df.groupby(by=['User_ID','Gender'])['Purchase'].sum()
avgspent = avgspent.reset_index()
avgspent
```

Out[6]:

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

5891 rows × 3 columns

Average number of Customers (Gender- Wise)

In [70]:
```
avgspent['Gender'].value_counts()
```

Out[70]:
```
M    4225
F    1666
Name: Gender, dtype: int64
```

Insights: Our customer data shows that there are 4,225 male customers and 1,666 female customers, highlighting a gender imbalance.

Total Spent

In [74]:
```
avgspent.groupby(by=['Gender'])['Purchase'].sum()
```

Out[74]:
```
Gender
F    1186232642
M    3909580100
Name: Purchase, dtype: int64
```

Insights: Females account for a total purchase value of $1,186,232,642, while males have a higher total purchase value of$ 3,909,580,100.

Average Spent

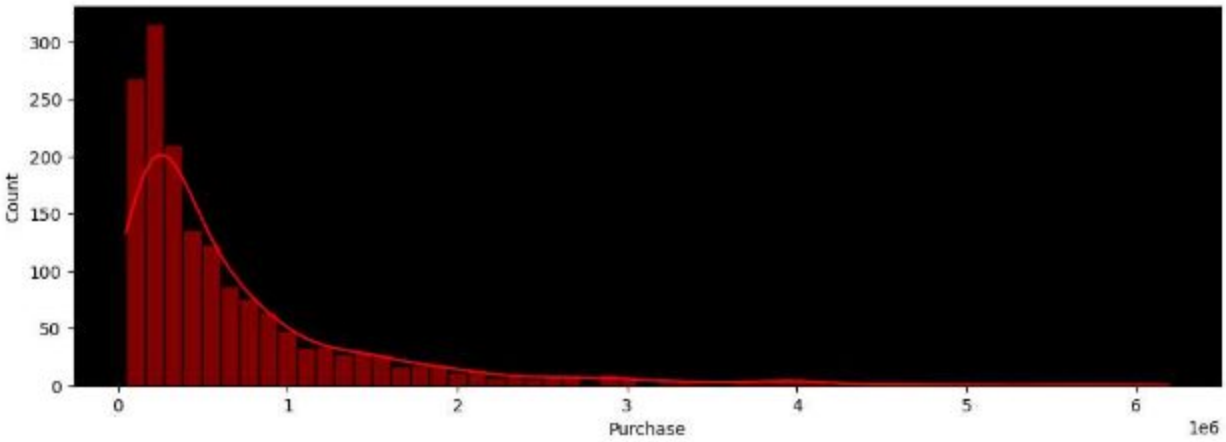
In [75]:
```
avgspent.groupby(by=['Gender'])['Purchase'].mean()
```

Out[75]:
```
Gender
F    712024.394958
M    925344.402367
Name: Purchase, dtype: float64
```

Insights: On average, males have higher purchase amounts at approximately $925,344, while females spend an average of about 712,024$, indicating a gender-based spending disparity.

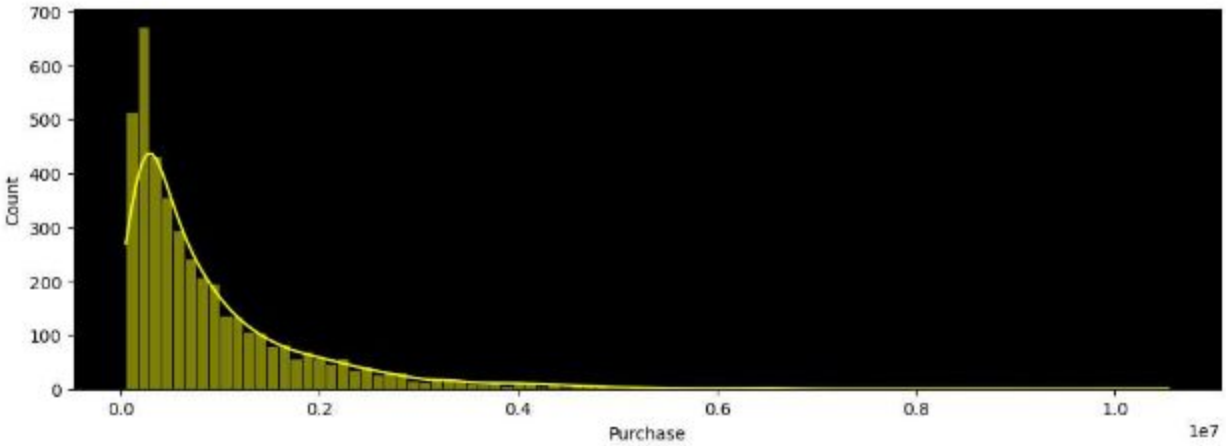
Female Average spending

In [67]:
```
plt.figure(figsize=(12,4))
sns.histplot(data=avgspent,x=avgspent[avgspent['Gender']=="F"]['Purchase'],kde=True,co
ax=plt.gca()
ax.set_facecolor('black')
plt.show()
```



Male Average spending

In [66]:
```
plt.figure(figsize=(12,4))
sns.histplot(data=avgspent,x=avgspent[avgspent['Gender']=="M"]['Purchase'],kde=True,co
ax=plt.gca()
ax.set_facecolor('black')
plt.show()
```



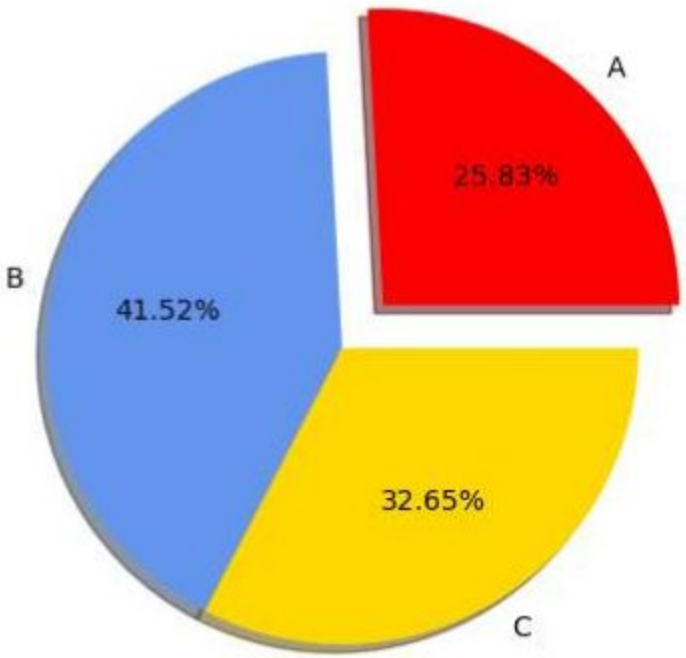
Cities Contributing Purchase Power

In [4]:
```
cities =df.groupby(by=['City_Category'])['Purchase'].sum().to_frame().reset_index()
cities['Average'] = cities['Purchase'].mean()
cities
```

Out[4]:

	City_Category	Purchase	Average
0	A	1316471661	1.698604e+09
1	B	2115533605	1.698604e+09
2	C	1663807476	1.698604e+09

In [17]:
```
from matplotlib.pyplot import pie, show
plt.pie(cities['Purchase'],autopct='%1.2f%%',labels=['A','B','C'],colors=['red',"corn
show()
```



Insights: The table shows purchase amounts in different city categories: A(26%), B(42%), and C(33%). Category B has the highest total purchase value which is 42% of purchase area, indicating it generates the most revenue. However, when considering the average purchase per category, all three are close, suggesting similar spending patterns across cities. This suggests potential for growth or targeted marketing efforts in Category B to maximize profits. The exploded part insures the least selling purchase fromt he city names a "A".

In [7]:
```
avgspent
```

Out[7]:

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

5891 rows × 3 columns

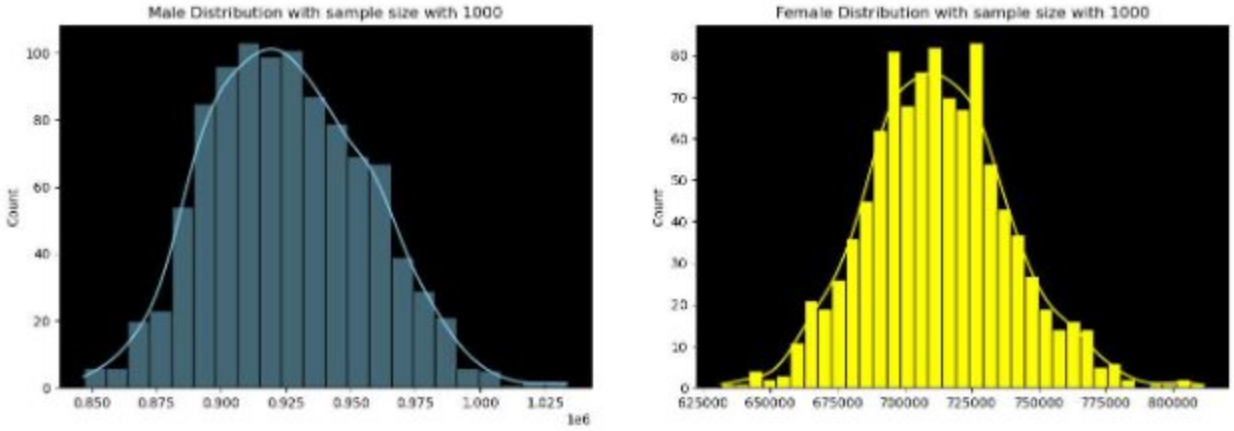
In [8]:
```
avgspent_male = avgspent[avgspent['Gender']=='M']
avgspent_female = avgspent[avgspent['Gender']=='F']
```

Finding the sample(sample size=1000) for avg purchase amount for males and females

In [258...
```
genders = ["M", "F"]

sample_size = 1000

samples_work_repeat = 1000
male_means = []
female_means = []

for i in range(samples_work_repeat):
    male_mean = avgspent_male.sample(sample_size, replace=True)['Purchase'].mean()
    female_mean = avgspent_female.sample(sample_size,replace=True)['Purchase'].mean()

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

In [72]:
```
plt.figure(figsize=(16,5))
plt.subplot(1,2,1)
sns.histplot(male_means,color="skyblue",ec="k",kde=True)
plt.title("Male Distribution with sample size with 1000")
ax=plt.gca()
ax.set_facecolor('black')
plt.subplot(1,2,2)
sns.histplot(female_means,bins=35,fc="yellow",ec='k',kde=True)
plt.title("Female Distribution with sample size with 1000")
ax=plt.gca()
ax.set_facecolor('black')
plt.show
```

Out[72]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



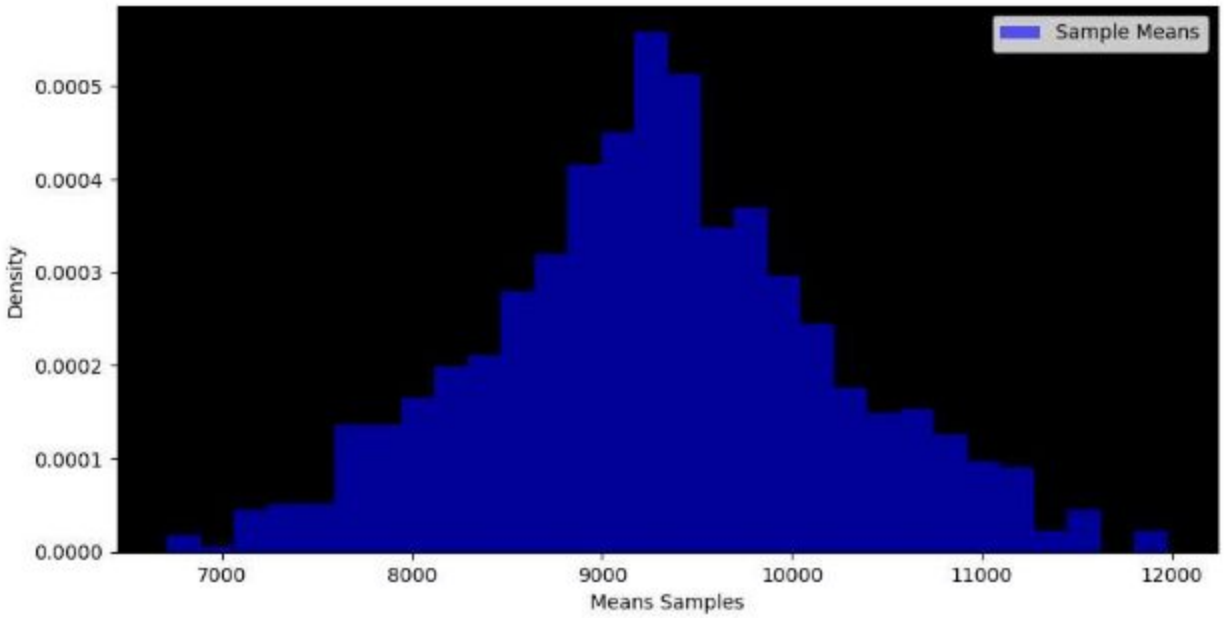
Business Insights (Male): The histogram of male purchase means shows a relatively normal distribution. This suggests that, on average, male customers exhibit consistent spending patterns. Businesses can rely on this consistency to tailor marketing strategies and product offerings to meet the expectations of their male customer base.

Business Insights (Female): The histogram of female purchase means appears to have a wider spread compared to males. This suggests greater variability in spending habits among female customers. Businesses may benefit from segmenting their female customer base to cater to different spending preferences and providing targeted promotions to specific segments.

Finding the Samples with Purchases with CLT

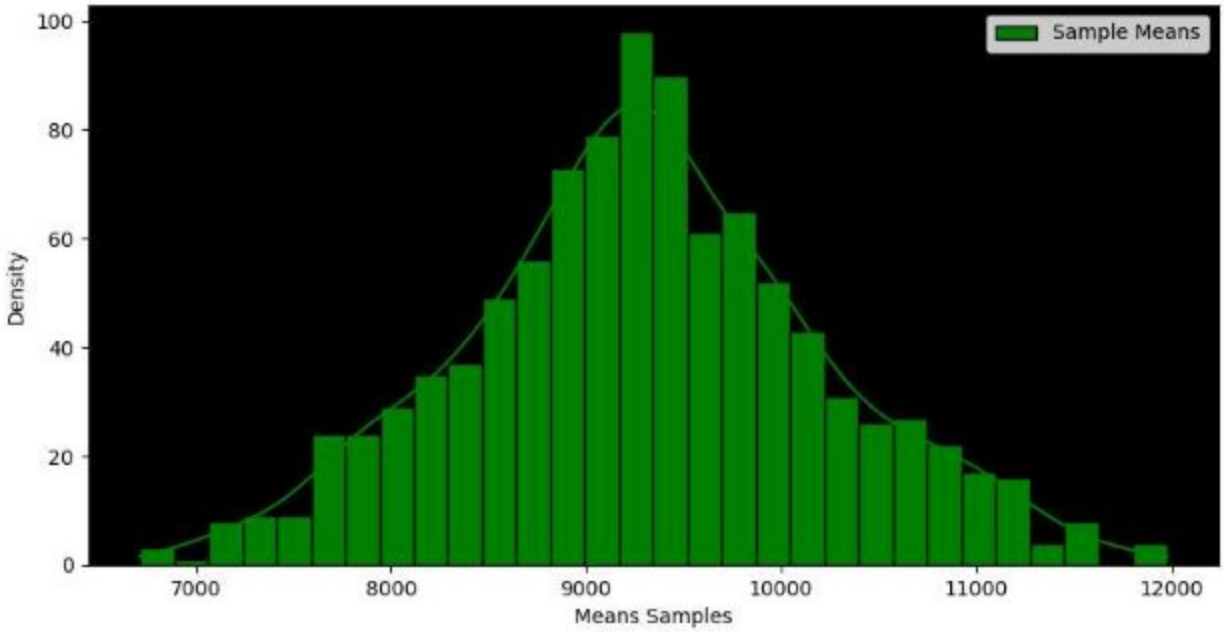
In [113…
```
population_mean  = df['Purchase'].mean()
population_stddev = df['Purchase'].std()
population_size  = len(df['Purchase'])
num_samples = 1000
sample_size = 30
sample_means = []

for i in range(num_samples):
    # Generating a random sample from the population
    sample = np.random.normal(population_mean, population_stddev, sample_size)

    # Calculating the sample mean and add it to the list
    sample_mean = np.mean(sample)
    sample_means.append(sample_mean)
```

In [288…
```
plt.figure(figsize=(10,5))
plt.hist(sample_means, bins=30, density=True, alpha=0.6, color='b', label='Sample Mear
plt.legend()
ax=plt.gca()
plt.xlabel("Means Samples")
plt.ylabel("Density")
ax.set_facecolor('black')
plt.show()
```



In [303…

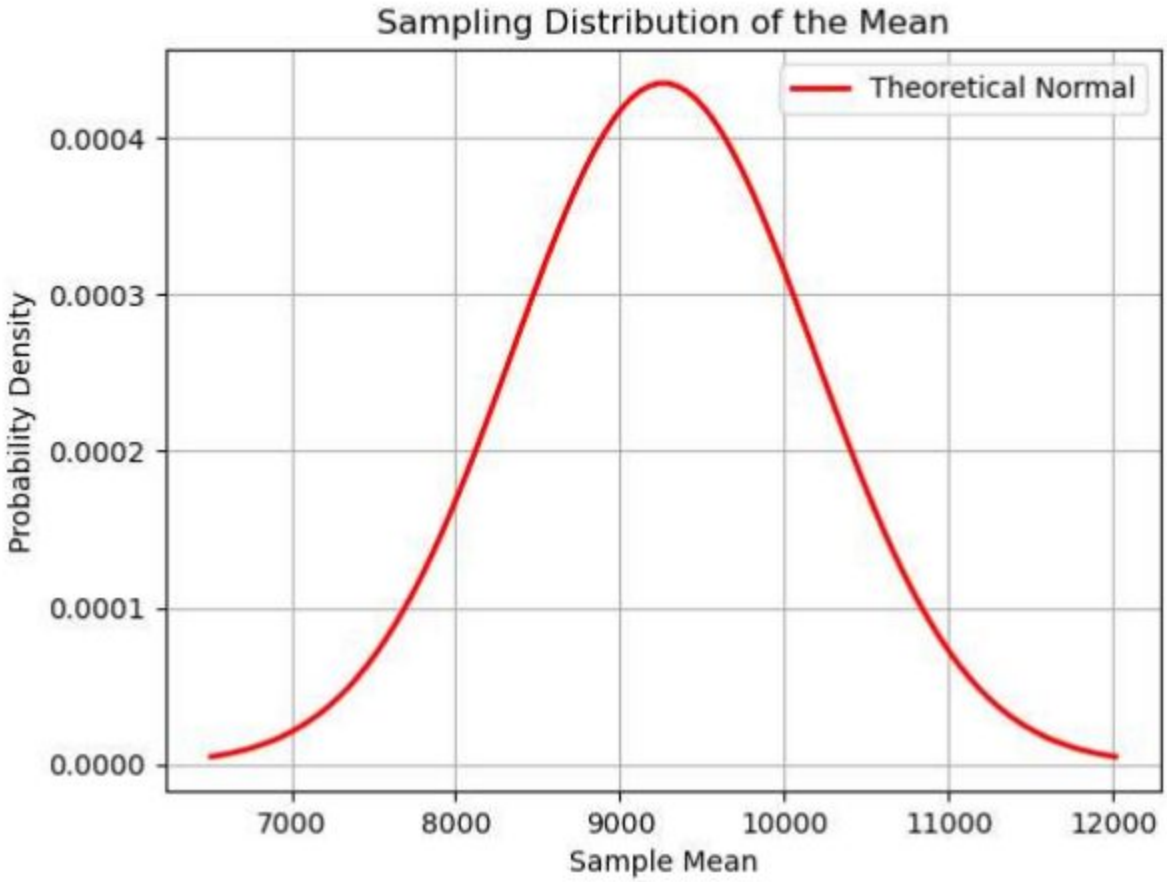
```
plt.figure(figsize=(10,5))
sns.histplot(sample_means,kde=True,label="Sample Means",bins=30,fc='g')
plt.legend()
ax=plt.gca()
plt.xlabel("Means Samples")
plt.ylabel("Density")
ax.set_facecolor('black')
plt.show()
```



Insights: When examining the purchase data, which represents the population mean, we took a sample and calculated the sample mean along with its standard deviation. Our analysis revealed that the distribution of sample means formed a bell-shaped curve, closely resembling a normal distribution. This observation suggests that the population mean itself follows a normal distribution, aligning with the Central Limit Theorem, which underscores the significance of this statistical principle in our analysis.

In [141…

```
sample_mean_theoretical = population_mean
sample_stddev_theoretical = population_stddev / np.sqrt(sample_size)
sample_stddev_theoretical
x = np.linspace(sample_mean_theoretical - 3*sample_stddev_theoretical,
                sample_mean_theoretical + 3*sample_stddev_theoretical, 100)
pdf = (1/(sample_stddev_theoretical * np.sqrt(2 * np.pi))) * \
      np.exp(-(x - sample_mean_theoretical)**2 / (2 * sample_stddev_theoretical**2))

# Plot the theoretical normal distribution
plt.plot(x, pdf, 'r', lw=2, label='Theoretical Normal')

plt.title('Sampling Distribution of the Mean')
plt.xlabel('Sample Mean')
plt.ylabel('Probability Density')
plt.legend()
plt.grid(True)

plt.show()
```



Observation for Confidence Interval at 90%, 95%, 99% for MALE and FEMALE

Observing at 90% Confidence Interval for Sample Size 1000 at z=90

population Average Mean

In [238…
```
print("Population Average Amount Spent for Male: {:.2f}".format(avgspent_male['Purchas
print("Population Average Amount Spent for Female: {:.2f}\n".format(avgspent_female['P
```

```
Population Average Amount Spent for Male: 925344.40
Population Average Amount Spent for Female: 712024.39
```

Sample Average Mean

In [239…
```
print("Sample Average Amount spent for Male: {:.2f}".format(np.mean(male_means)))
print("Sample Average Amount spent for Female: {:.2f}\n".format(np.mean(female_means)))
```

```
Sample Average Amount spent for Male: 925827.52
Sample Average Amount spent for Female: 712358.43
```

Sample Standard Deviation

In [243…
```
print("The std dev for Sample for Male: {:.2f}".format(pd.Series(male_means).std()))
print("The std dev for Sample for Female: {:.2f}".format(pd.Series(female_means).std()))
```

```
The std dev for Sample for Male: 30762.23
The std dev for Sample for Female: 26403.44
```

Sample Error (Std()/sqrt(n))

In [244…
```
print("Sample Standard Error for Male: {:.2f}".format(pd.Series(male_means).std()/np.s
print("Sample Standard Error for Female: {:.2f}\n".format(pd.Series(female_means).std(
```

```
Sample Standard Error for Male: 972.79
Sample Standard Error for Female: 834.95
```

In [245…
```
sample_mean_male=np.mean(male_means)
sample_mean_female=np.mean(female_means)

sample_std_male=pd.Series(male_means).std()
sample_std_female=pd.Series(female_means).std()

sample_std_error_male=sample_std_male/np.sqrt(1000)
sample_std_error_female=sample_std_female/np.sqrt(1000)
```

In [259…
```
#Calculating Confidence Interval with Formula: (mu(sample)+z*std),(mu(sample)-z*std) a
```

In [282…
```
z90 = 1.645
First_scale_male_z90=z90*sample_std_error_male + sample_mean_male
last_scale_male_z90=sample_mean_male - z90*sample_std_error_male

first_scale_female_z90=z90*sample_std_error_female + sample_mean_female
last_scale_female_z90=sample_mean_female - z90*sample_std_error_female
```

Confidence Interval at 90%

In [283…
```
print("Male_Confidence_interval_z90: ",[First_scale_male,last_scale_male])
print("Female_Confidence_intervalz90: ",[first_scale_female,last_scale_female])
```

```
Male_Confidence_interval_z90:  [927302.8204837536, 923277.6722862463]
Female_Confidence_intervalz90:  [712726.8087989037, 709434.5699990961]
```

Insights: Male Population Insight: With a 90% confidence level, we estimate that the average or relevant population parameter for males falls within the range of approximately 923,277.67 to 927,302.82. This information can be valuable for businesses targeting or analyzing the male demographic, helping them make informed decisions regarding marketing, product development, or resource allocation.

Female Population Insight: Similarly, for the female population, our 90% confidence interval suggests that the parameter of interest is likely to be within the range of approximately 709,434.57 to 712,726.81. This insight is crucial for businesses with a focus on females, aiding them in strategic planning, customer engagement, and market positioning.

Observing at 95% Confidence Interval for Sample Size 1000 at z=95

In [262…
```
sample_mean_male=np.mean(male_means)
sample_mean_female=np.mean(female_means)

sample_std_male=pd.Series(male_means).std()
sample_std_female=pd.Series(female_means).std()

sample_std_error_male=sample_std_male/np.sqrt(1000)
sample_std_error_female=sample_std_female/np.sqrt(1000)
```

In [268…
```
# Calculating Confidence Interval with Formula: (mu(sample)+z*std),(mu(sample)-z*std)
```

In [264…
```
z95 = 1.96
First_scale_male95=z95*sample_std_error_male + sample_mean_male
last_scale_male95=sample_mean_male - z95*sample_std_error_male

first_scale_female95=z95*sample_std_error_female + sample_mean_female
last_scale_female95=sample_mean_female - z95*sample_std_error_female
```

Confidence Interval at 95%

In [266…
```
print("Male_Confidence_interval: ",[First_scale_male95,last_scale_male95])
print("Female_Confidence_interval: ",[first_scale_female95,last_scale_female95])
```

```
Male_Confidence_interval:  [927302.8204837536, 923277.6722862463]
Female_Confidence_interval:  [712726.8087989037, 709434.5699990961]
```

Insights: Male Population (95% Confidence): For males, the parameter of interest is estimated to fall within 927302.82, 923277.67 with 95% confidence. This provides businesses targeting males with a highly reliable range for decision-making regarding marketing and resource allocation.

Female Population (95% Confidence): The 95% confidence interval for females is 712726.80, 709434.5, indicating a strong level of confidence in the parameter estimate. Businesses focusing on females can use this interval for strategic planning, customer engagement, and market positioning with a high degree of certainty.

Observing at 99% Confidence Interval for Sample Size 1000 at z=99

In [271…
```
sample_mean_male=np.mean(male_means)
sample_mean_female=np.mean(female_means)

sample_std_male=pd.Series(male_means).std()
sample_std_female=pd.Series(female_means).std()

sample_std_error_male=sample_std_male/np.sqrt(1000)
sample_std_error_female=sample_std_female/np.sqrt(1000)
```

In [273…
```
# Calculating Confidence Interval with Formula: (mu(sample)+z*std),(mu(sample)-z*std)
```

In [280…
```
z99 = 2.576
First_scale_male99=z99*sample_std_error_male + sample_mean_male
last_scale_male99=sample_mean_male - z99*sample_std_error_male

first_scale_female99=z99*sample_std_error_female + sample_mean_female
last_scale_female99=sample_mean_female - z99*sample_std_error_female
```

Confidence Interval at 99%

In [286…
```
print("Male_Confidence_interval: ",[First_scale_male99,last_scale_male99])
print("Female_Confidence_interval: ",[first_scale_female99,last_scale_female99])
```

```
Male_Confidence_interval:  [927935.3437719333, 922645.1489980666]
Female_Confidence_interval:  [713244.160610302, 708917.2181876978]
```

Insights: Male Population Insight (99% Confidence): The 99% confidence interval for males 922,645.15, 927,935.34], indicates that we are highly confident in the estimated parameter's range. Businesses targeting males can make strategic decisions with a strong level of assurance regarding marketing, product development, and resource allocation.

Female Population Insight (99% Confidence): For females, the 99% confidence interval 708,917.22, 713,244.16 provides a robust basis for decision-making. Businesses focusing on females can rely on this interval for strategic planning, customer engagement, and market positioning with a high degree of confidence.

In [7]:
```
df.head()
```

Out[7]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_
0	1000001	P00069042	F	0-17	10	A	2	
1	1000001	P00248942	F	0-17	10	A	2	
2	1000001	P00087842	F	0-17	10	A	2	
3	1000001	P00085442	F	0-17	10	A	2	
4	1000002	P00285442	M	55+	16	C	4+	

CLT AND CONFIDENCE INTERVAL FOR MARITAL STATUS:

In [9]:
```
avg_marital = df.groupby(['User_ID','Marital_Status'])['Purchase'].sum()
avg_marital = avg_marital.reset_index()
avg_marital.head()
```

Out[9]:

	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

In [13]:
```
avg_married =avg_marital[avg_marital['Marital_Status']==1]
avg_unmarried =avg_marital[avg_marital['Marital_Status']==0]
```

In [14]:
```
print(avg_married)
print(avg_unmarried)
```

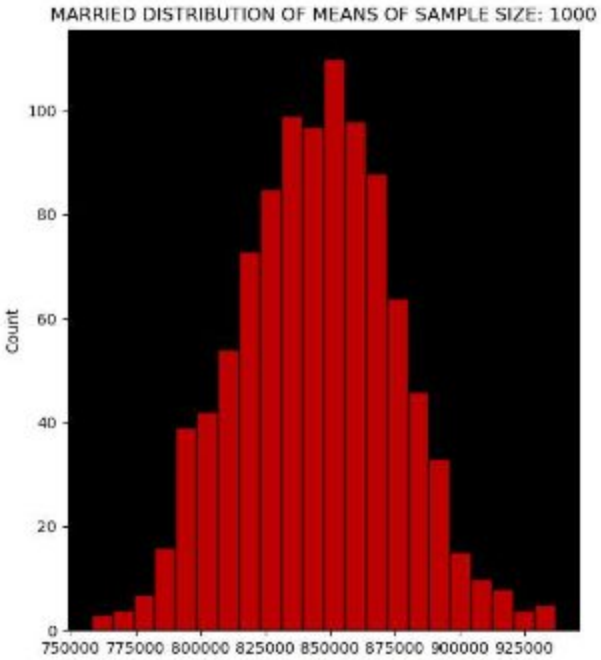
```
      User_ID  Marital_Status  Purchase
3     1000004               1    206468
4     1000005               1    821001
6     1000007               1    234668
7     1000008               1    796593
9     1000010               1   2169510
...       ...             ...       ...
5879  1006029               1    157436
5880  1006030               1    737361
5883  1006033               1    501843
5886  1006036               1   4116058
5889  1006039               1    590319

[2474 rows x 3 columns]
      User_ID  Marital_Status  Purchase
0     1000001               0    334093
1     1000002               0    810472
2     1000003               0    341635
5     1000006               0    379930
8     1000009               0    594099
...       ...             ...       ...
5884  1006034               0    197086
5885  1006035               0    956645
5887  1006037               0   1119538
5888  1006038               0     90034
5890  1006040               0   1653299

[3417 rows x 3 columns]
```

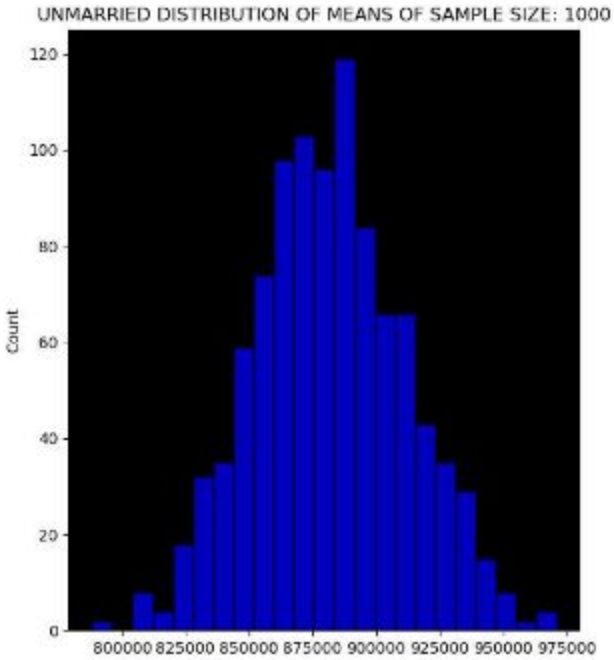
average expenses of customer depending upon their marital status

In [28]:

```
sample_size = 1000
num_repitions = 1000

married_means = []
unmarried_means = []

for i in range(num_repitions):
    avg_married = avg_marital[avg_marital['Marital_Status']==1].sample(sample_size, re
    avg_unmarried = avg_marital[avg_marital['Marital_Status']==0].sample(sample_size,

    married_means.append(avg_married)
    unmarried_means.append(avg_unmarried)


plt.figure(figsize =(13,7))
plt.subplot(1,2,1)
sns.histplot(married_means, bins='auto',color = 'red')
plt.title("MARRIED DISTRIBUTION OF MEANS OF SAMPLE SIZE: 1000")
ax=plt.gca()
ax.set_facecolor('black')


plt.subplot(1,2,2)
sns.histplot(unmarried_means, bins='auto',color='blue')
plt.title("UNMARRIED DISTRIBUTION OF MEANS OF SAMPLE SIZE: 1000")
ax=plt.gca()
```

```
ax.set_facecolor('black')

plt.show()
```





Insights: From the analysis, it is evident that the samples exhibit a normal distribution for both married and Unmarried customers. Additionally, we can glean valuable business insights by noting that the means of these sample sets closely align with the population mean, as stipulated by the central limit theorem. This suggests that our data is statistically reliable, enabling us to make more confident decisions based on this information.

In [29]:
```
avg_marital['Marital_Status'].value_counts()
```

Out[29]:
```
0    3417
1    2474
Name: Marital_Status, dtype: int64
```

This analysis reveals a noteworthy business insight: the unique count of occurrences for unmarried customers is significantly higher, standing at approximately 3417, as opposed to Unmarried, where the count is 2474 as Married. This disparity underscores the importance of catering to the needs and preferences of unmarried customers, as they constitute a larger portion of our target demographic.

CALCULATING 90 % CONFIDENCE INTERVAL FOR AVERAGE EXPENSES FOR THE MARRIED AND UNMARRIED HAVING 1000 SAMPLE SIZE

At 90 % confidence Interval

In [ ]:
```
#Calculating Confidence Interval with Formula: (mu(sample)+z*std),(mu(sample)-z*std) c
```

In [30]:
```
z90 = norm.ppf((1 + 0.90)/2)
z90 = z90.round(3)
```

```
z90
```

Out[30]: 1.645

In [39]:
```
sample_mean_married=np.mean(married_means)
sample_mean_unmarried=np.mean(unmarried_means)
print(sample_mean_married)
print(sample_mean_unmarried)
```

```
845126.611096
881612.4409859999
```

In [41]:
```
sample_std_married=pd.Series(married_means).std()
sample_std_unmarried=pd.Series(unmarried_means).std()
print(sample_std_married)
print(sample_std_unmarried)
```

```
30315.62730099818
29837.110649661743
```

In [43]:
```
sample_std_error_married=sample_std_married/np.sqrt(1000)
sample_std_error_unmarried=sample_std_unmarried/np.sqrt(1000)
print(sample_std_error_married)
print(sample_std_error_unmarried)
```

```
958.6643096793717
943.5322845139738
```

calculate confidence interrval using formula CI = mean(sample) + (z(confidence percentage) * sample standard error)

In [45]:
```
Upper_Limit_married = (sample_mean_married + (z90 * (sample_std_error_married)))
Lower_Limit_married = (sample_mean_married - (z90 * (sample_std_error_married)))
```

In [46]:
```
Upper_Limit_unmarried = (sample_mean_unmarried + (z90 * (sample_std_error_unmarried)))
Lower_Limit_unmarried = (sample_mean_unmarried - (z90 * (sample_std_error_unmarried)))
```

In [48]:
```
print("Married_CI: ",[Lower_Limit_married,Upper_Limit_married])
print("unmarried_CI: ",[Lower_Limit_unmarried,Upper_Limit_unmarried])
```

```
Married_CI:  [843549.6083065774, 846703.6138854225]
unmarried_CI:  [880060.3303779744, 883164.5515940253]
```

CALCULATING 95 % CONFIDENCE INTERVAL FOR AVERAGE EXPENSES FOR THE MARRIED AND UNMARRIED HAVING 1000 SAMPLE SIZE

At 95 % confidence Interval

In [50]:
```
z95 = norm.ppf((1 + 0.95)/2)
z95 = z95.round(3)
z95
```

Out[50]: 1.96

calculate confidence interrval using formula CI = mean(sample) + (z(confidence percentage) * sample standard error)

In [52]:
```
Upper_Limit_married = (sample_mean_married + (z95 * (sample_std_error_married)))
Lower_Limit_married = (sample_mean_married - (z95 * (sample_std_error_married)))
```

In [54]:
```
Upper_Limit_unmarried = (sample_mean_unmarried + (z95 * (sample_std_error_unmarried)))
Lower_Limit_unmarried = (sample_mean_unmarried - (z95 * (sample_std_error_unmarried)))
```

In [56]:
```
print("Married_CI: ",[Lower_Limit_married,Upper_Limit_married])
print("Unmarried_CI: ",[Lower_Limit_unmarried,Upper_Limit_unmarried])
```

```
Married_CI:  [843247.6290490284, 847005.5931429715]
Unmarried_CI:  [879763.1177083525, 883461.7642636473]
```

CALCULATING 99 % CONFIDENCE INTERVAL FOR AVERAGE EXPENSES FOR THE MARRIED AND UNMARRIED HAVING 1000 SAMPLE SIZE

At 99 % confidence Interval

In [59]:
```
z99 = norm.ppf((1 + 0.99)/2)
z99 = z99.round(3)
z99
```

Out[59]:
```
2.576
```

calculate confidence interrval using formula CI = mean(sample) + (z(confidence percentage) * sample standard error)

In [60]:
```
Upper_Limit_married = (sample_mean_married + (z99 * (sample_std_error_married)))
Lower_Limit_married = (sample_mean_married - (z99 * (sample_std_error_married)))
```

In [61]:
```
Upper_Limit_unmarried = (sample_mean_unmarried + (z99 * (sample_std_error_unmarried)))
Lower_Limit_unmarried = (sample_mean_unmarried - (z99 * (sample_std_error_unmarried)))
```

In [66]:
```
print("Married_CI: ",[Lower_Limit_married,Upper_Limit_married])
print("Unmarried_CI: ",[Lower_Limit_unmarried,Upper_Limit_unmarried])
```

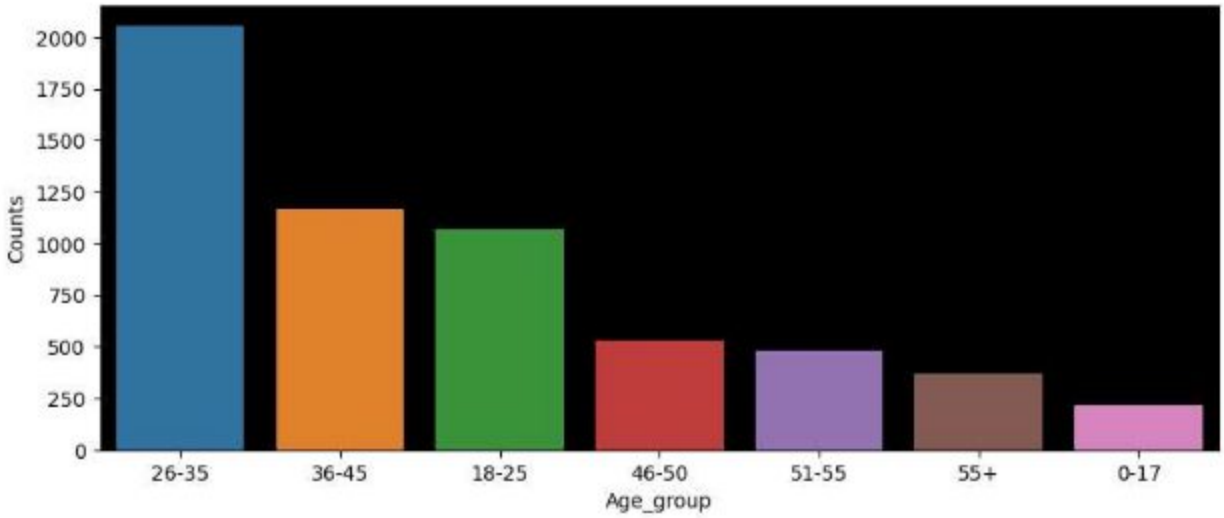
```
Married_CI:  [842657.0918342659, 847596.130357734]
Unmarried_CI:  [879181.9018210919, 884042.9801509079]
```

Calculating the CLT and CI for the Age groups with the purchase mean

In [108…
```
avg_age = df.groupby(['User_ID', 'Age'])[['Purchase']].sum()
avg_age = avgamt_age.reset_index()

avg_age = avg_age['Age'].value_counts()
avg_age=pd.DataFrame(avg_age).reset_index()
avg_age.rename(columns={"index":"Age_group","Age":"Counts"}, inplace=True)
```

In [110…
```
plt.figure(figsize =(10,4))
sns.barplot(data=avg_age, x=avg_age['Age_group'],y=avg_age['Counts'])
ax=plt.gca()
```

```
ax.set_facecolor('black')
plt.show()
```



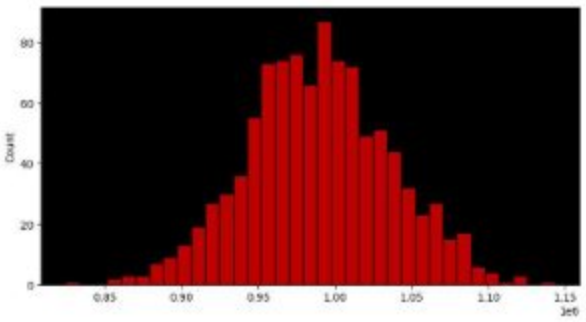
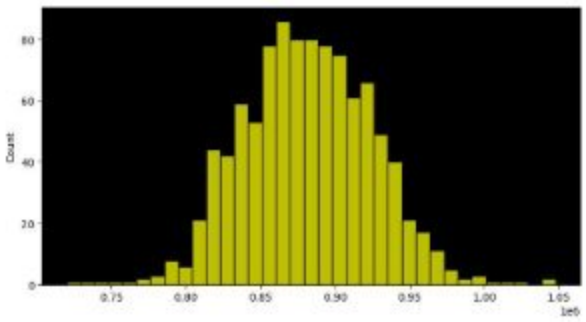
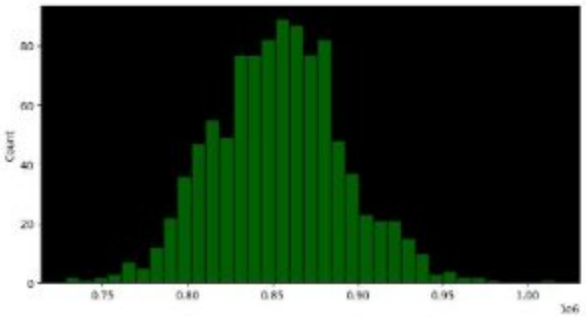
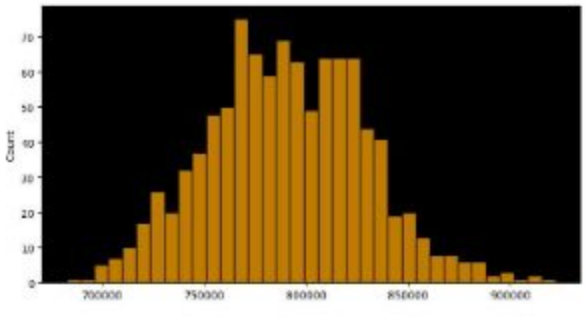
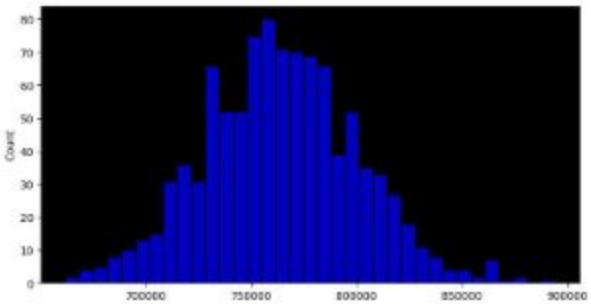
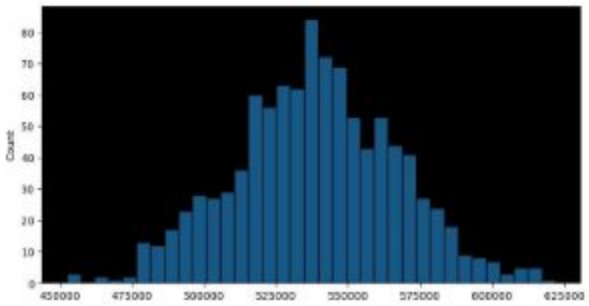
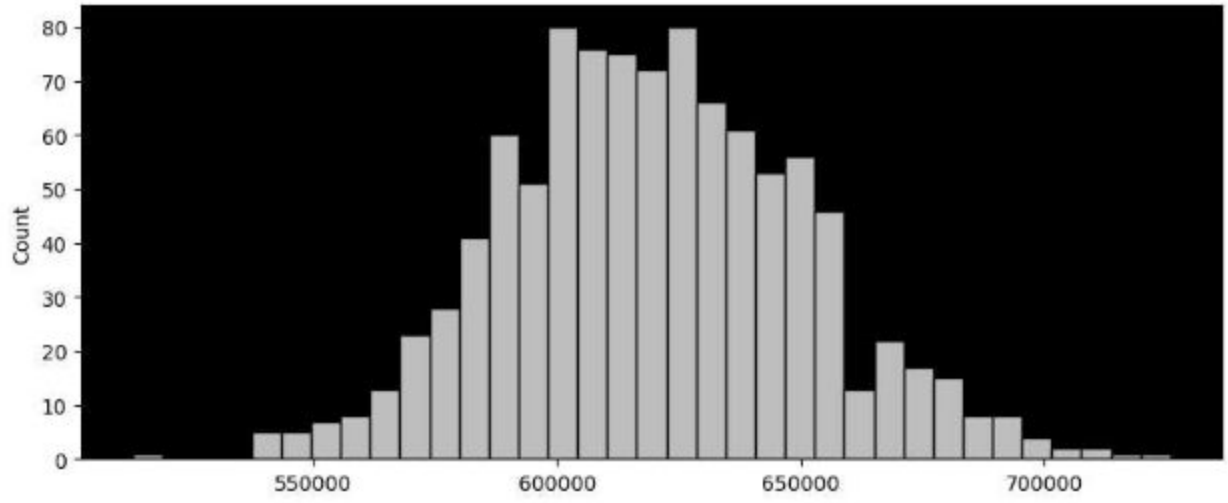
In [122…

```
sample_size = 500
num_repitions = 1000

all_sample_means = {}

age_intervals = ['26-35', '36-45', '18-25', '46-50', '51-55', '55+', '0-17']
for i in age_intervals:
    all_sample_means[i] = []

for i in age_intervals:
    for j in range(num_repitions):

        mean = avgamt_age[avgamt_age['Age']==i].sample(sample_size, replace=True)['Pur
        all_sample_means[i].append(mean)
```

In [135…

```
plt.figure(figsize=(20,16))
plt.subplot(3,2,1)
sns.histplot(all_sample_means['26-35'],bins=35,color='red')
ax=plt.gca()
ax.set_facecolor('black')
plt.subplot(3,2,2)
sns.histplot(all_sample_means['36-45'],bins=35,color='yellow')
ax=plt.gca()
ax.set_facecolor('black')
plt.subplot(3,2,3)
sns.histplot(all_sample_means['18-25'],bins=35,color='green')
ax=plt.gca()
ax.set_facecolor('black')
plt.subplot(3,2,4)
sns.histplot(all_sample_means['46-50'],bins=35,color='orange')
ax=plt.gca()
ax.set_facecolor('black')
plt.subplot(3,2,5)
sns.histplot(all_sample_means['51-55'],bins=35,color='Blue')
ax=plt.gca()
ax.set_facecolor('black')
plt.subplot(3,2,6)
sns.histplot(all_sample_means['55+'],bins=35)
ax=plt.gca()
```

```
ax.set_facecolor('black')

plt.show()

plt.figure(figsize=(10, 4))
sns.histplot(all_sample_means['0-17'],bins=35,color='white')
ax=plt.gca()
ax.set_facecolor('black')
plt.show()
```

Insights: The analysis segments customers into distinct age intervals, such as '26-35', '36-45', '18-25', '46-50', '51-55', '55+', and '0-17'. This segmentation allows for a more granular

understanding of purchasing behavior across various age groups.

Business Inisghts: The analysis indicates that the sample means exhibit a normal distribution pattern across all age groups. Additionally, a noteworthy business insight derived from this analysis is that the means of these samples closely approximate the population mean, aligning with the expectations of the central limit theorem. This suggests that our collected data is statistically reliable and can serve as a solid foundation for making informed business decisions related to customer purchasing behavior across different age groups.