In [1]:

```
import pandas as pd
df=pd.read_csv('/Users/suraaj/Desktop/DSML/dsml-case-studies/Walmart CLT/walmart.csv')
```

In [2]:

```
df
```

Out[2]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years
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

In [4]:

```
df.shape
```

Out[4]:

```
(550068, 10)
```

In [5]:

```
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
```

In [6]:

```
df.groupby('Gender')['Purchase'].describe()
```

Out[6]:

Gender	count	mean	std	min	25%	50%	75%	max
F	135809.0	8734.565765	4767.233289	12.0	5433.0	7914.0	11400.0	23959.0
M	414259.0	9437.526040	5092.186210	12.0	5863.0	8098.0	12454.0	23961.0

In [7]:

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

Out[7]:

```
Gender
F    1666
M    4225
Name: User_ID, dtype: int64
```

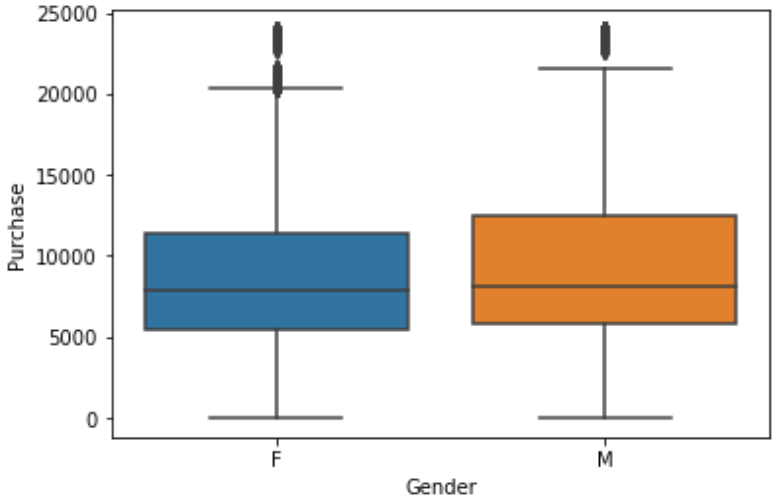
In [8]:

```
import seaborn as sbn


sbn.boxplot(x='Gender', y='Purchase', data=df)
```

Out[8]:

```
<AxesSubplot:xlabel='Gender', ylabel='Purchase'>
```

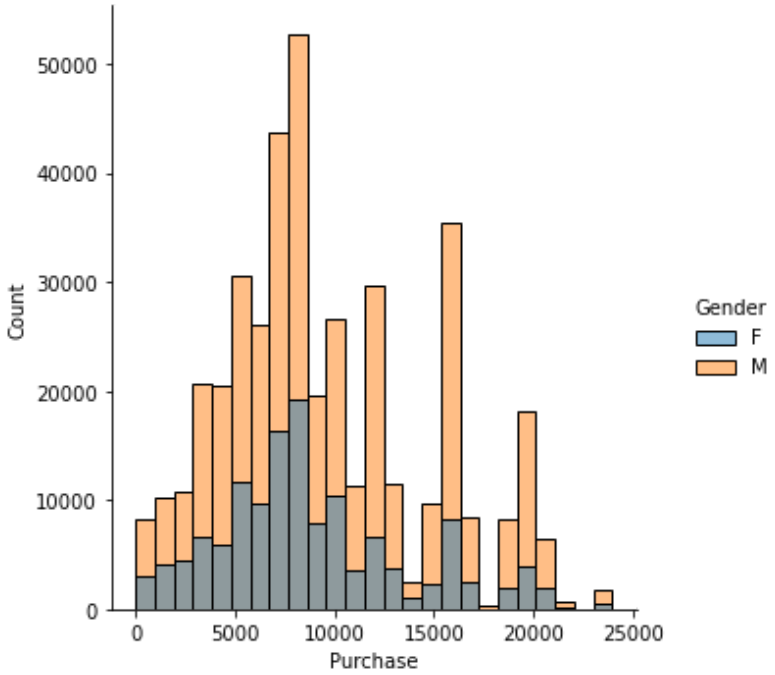


In [11]:

```
sbn.displot (x='Purchase', data=df,hue='Gender', bins=25)
```

Out[11]:

```
<seaborn.axisgrid.FacetGrid at 0x7fd6dac425b0>
```



In [12]:

```
df.groupby('Gender')['Purchase'].describe()
```

Out[12]:

Gender	count	mean	std	min	25%	50%	75%	max
F	135809.0	8734.565765	4767.233289	12.0	5433.0	7914.0	11400.0	23959.0
M	414259.0	9437.526040	5092.186210	12.0	5863.0	8098.0	12454.0	23961.0

In [13]:

```
df.sample(300).groupby('Gender')['Purchase'].describe()
```

Out[13]:

Gender	count	mean	std	min	25%	50%	75%	max
F	71.0	9139.929577	4637.239374	2020.0	6874.0	8002.0	11832.5	23138.0
M	229.0	9908.746725	5220.197782	62.0	6058.0	8716.0	13703.0	23451.0

In [14]:

```
df.sample(300).groupby('Gender')['Purchase'].describe()
```

Out[14]:

Gender	count	mean	std	min	25%	50%	75%	max
F	85.0	8406.258824	4356.823223	62.0	5365.0	7758.0	10033.0	19489.0
M	215.0	8976.200000	5070.919584	400.0	5312.5	7992.0	11920.5	23885.0

In [15]:

```
df.sample(300).groupby('Gender')['Purchase'].describe()
```

Out[15]:

Gender	count	mean	std	min	25%	50%	75%	max
F	81.0	8234.086420	5103.232013	60.0	5291.0	7184.0	9922.0	23064.0
M	219.0	9498.808219	4965.723675	695.0	5977.0	8018.0	12754.5	20636.0

In [17]:

```
#clt


sample_x =500
interations = 1000


male_sample_means= [df[df['Gender']=='M'].sample(sample_x, replace=True)['Purchase'].r
```

In [18]:

```
female_sample_means= [df[df['Gender']=='F'].sample(sample_x, replace=True)['Purchase'
```

In [20]:

```
male_sample_means
```

```
 9249.03,
 9168.842,
 9409.446,
 9586.638,
 9545.48,
 9781.004,
 9369.132,
 9167.41,
 8860.914,
 9460.984,
 9347.136,
 9581.654,
 9587.51,
 9353.43,
 9113.448,
 9224.642,
 9790.7,
 9490.792,
 9295.742,
 9590.736,
```

In [21]:

```
import numpy as np

np.mean(male_sample_means)
```

Out[21]:

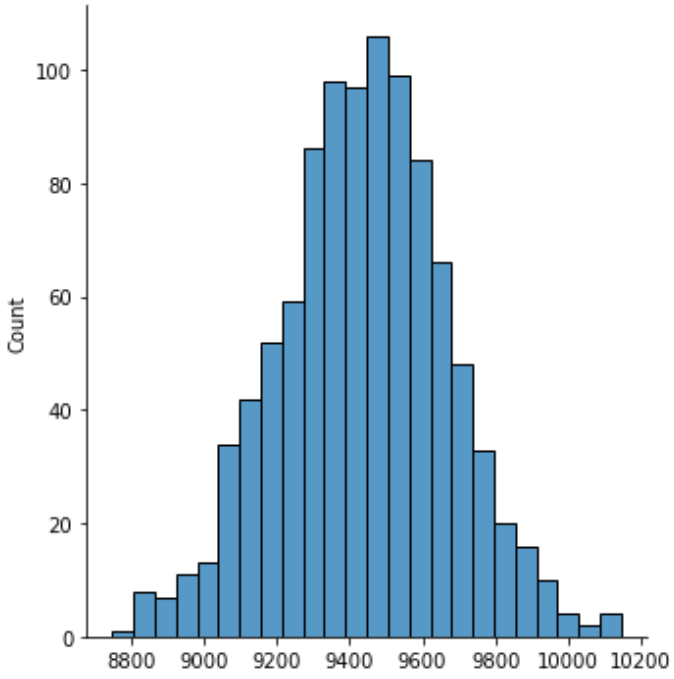
```
9436.801242
```

In [22]:

```
sbn.displot(male_sample_means)
```

Out[22]:

```
<seaborn.axisgrid.FacetGrid at 0x7fd6dad16190>
```

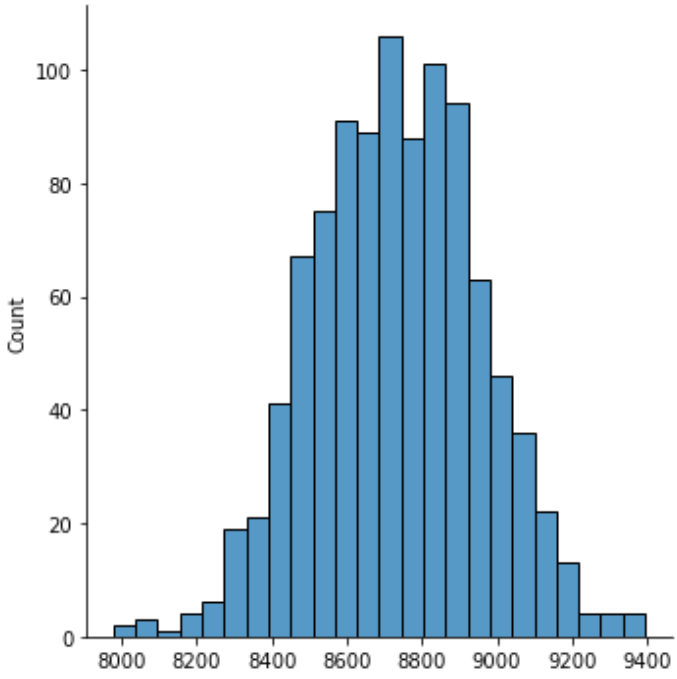


In [23]:

```
sbn.displot(female_sample_means)
```

Out[23]:

```
<seaborn.axisgrid.FacetGrid at 0x7fd6e86f7f40>
```



In [ ]:

```
#calculate confidence intervals


2 ways
1. Z score
2. Percentile
```

In [24]:

```
male_upper_limit= np.mean(male_sample_means) + 1.96 * np.std(male_sample_means)
male_lower_limit= np.mean(male_sample_means) - 1.96 * np.std(male_sample_means)
```

In [25]:

```
female_upper_limit= np.mean(female_sample_means) + 1.96 * np.std(female_sample_means)
female_lower_limit= np.mean(female_sample_means) - 1.96 * np.std(female_sample_means)
```

In [26]:

```
(male_lower_limit,male_upper_limit )
```

Out[26]:

```
(8986.440489768998, 9887.161994231)
```

In [27]:

```
(female_lower_limit,female_upper_limit )
```

Out[27]:

```
(8295.904195839254, 9164.842364160746)
```

In [ ]:

```
#1. Increase the sample
#2. Decreasing the confidence interval
#3. Do it for other columns
```

In [28]:

```
np.percentile(male_sample_means,[2.5, 97.5])
```

Out[28]:

```
array([8971.369  , 9888.83395])
```

In [29]:

```
np.percentile(female_sample_means,[2.5, 97.5])
```

Out[29]:

```
array([8302.6396 , 9156.31215])
```

In [ ]:

```
```