In [1]:
```
import pandas as pd
```

In [3]:
```
df=pd.read_csv('walmart.csv')
```

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

Out[4]:

Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Status	Product_Category
P00069042	F	0-17	10	A	2	0	3
P00248942	F	0-17	10	A	2	0	1
P00087842	F	0-17	10	A	2	0	12
P00085442	F	0-17	10	A	2	0	12
P00285442	M	55+	16	C	4+	0	8

In [5]:
```
df['Gender'].value_counts()
```

Out[5]:
```
M    414259
F    135809
Name: Gender, dtype: int64
```

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

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

In [7]:
```
df.groupby('Gender')['User_ID'].nunique()
```

Out[7]:
```
Gender
F    1666
M    4225
Name: User_ID, dtype: int64
```

In [8]:
```
df.groupby('Gender')['Purchase'].describe()
```

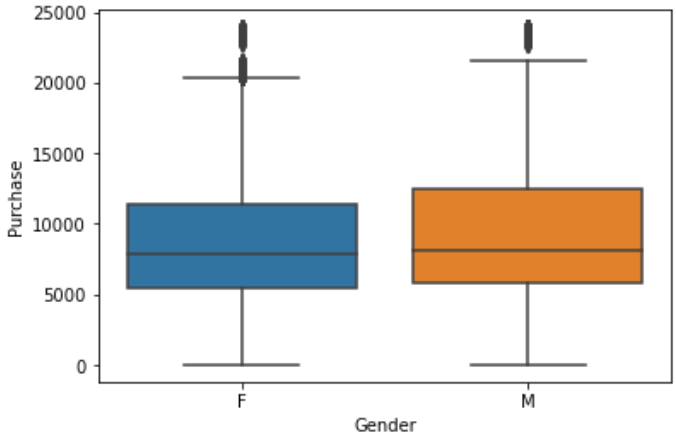
Out[8]:

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 [9]:
```
import seaborn as sbn

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

Out[9]:
```
<AxesSubplot:xlabel='Gender', ylabel='Purchase'>
```

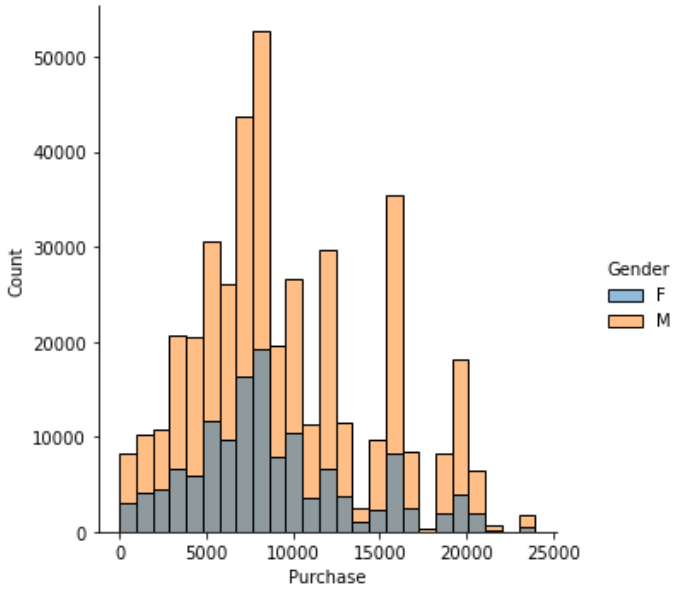


In [ ]:
```
#There is no major difference between the median of spend between males and females
#Therefore, we cannot say it with clarity, who is spending more.


#xFor pop-> CLT (checking who's spending more)
```

In [11]:
```
sbn.displot(x='Purchase', hue='Gender', data =df, bins=25)
```

Out[11]:
```
<seaborn.axisgrid.FacetGrid at 0x7faab05ac2b0>
```



In [13]:
```
sample = 300
```

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 [14]:
```
df.sample(300).groupby('Gender')['Purchase'].describe()
```

Out[14]:

Gender	count	mean	std	min	25%	50%	75%	max
F	84.0	9486.250000	5056.967497	49.0	6087.75	8599.5	11734.25	20529.0
M	216.0	9815.648148	5215.959408	138.0	5961.50	8601.0	13056.75	21267.0

In [15]:
```
df.sample(300).groupby('Gender')['Purchase'].describe()
```

Out[15]:

Gender	count	mean	std	min	25%	50%	75%	max
F	60.0	8668.350	3920.704649	570.0	6728.5	8014.5	9805.0	20027.0
M	240.0	9925.075	5248.062855	489.0	6101.5	8588.0	14213.0	23798.0

In [18]:
```
df.sample(300).groupby('Gender')['Purchase'].describe()
```

Out[18]:

Gender	count	mean	std	min	25%	50%	75%	max
F	71.0	8510.000000	4410.609841	1432.0	5403.5	7801.0	11906.0	19413.0
M	229.0	9347.868996	5174.740454	14.0	5930.0	8061.0	11976.0	23675.0

In [20]:
```
male_sample_means= [df[df['Gender']=='M'].sample (300, replace=True)['Purchase'].mea
```

In [21]:
```
female_sample_means= [df[df['Gender']=='F'].sample (300, replace=True)['Purchase'].m
```

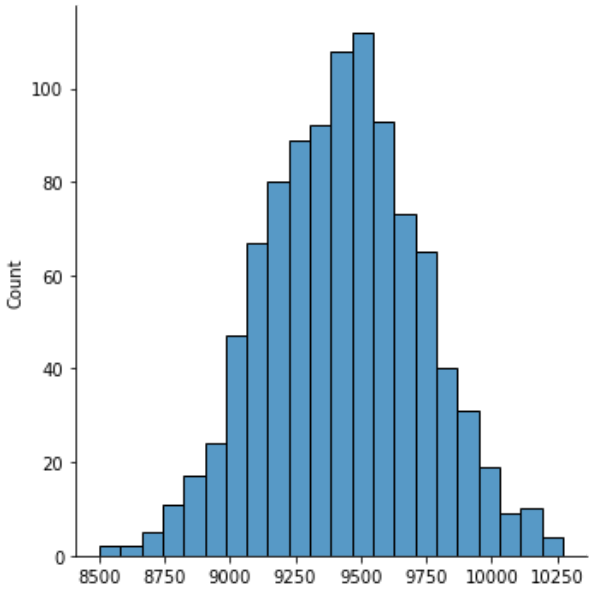
In [24]:
```
import numpy as np

np.mean(male_sample_means)
```

Out[24]: 9428.60124

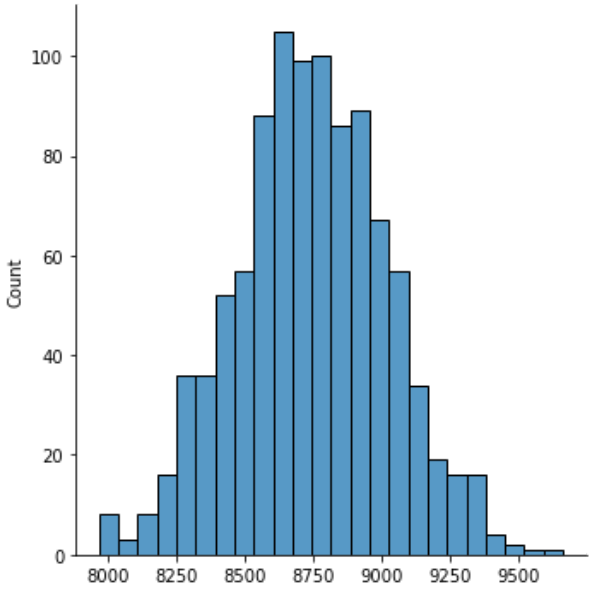
In [26]:
```
sbn.displot(male_sample_means)
```

Out[26]: <seaborn.axisgrid.FacetGrid at 0x7faab05ac6a0>



In [27]:
```
sbn.displot(female_sample_means)
```

Out[27]:
```
<seaborn.axisgrid.FacetGrid at 0x7faaf27d0160>
```



In [ ]:
```
#confidence interval to check average male and female spends
```

In [ ]:
```
upperlimit_males= means + Z-score * Standard Error
```

In [31]:
```
upperlimit_males= np.mean(male_sample_means) + 1.96 * np.std(male_sample_means)
```

In [30]:
```
lowerlimit_males= np.mean(male_sample_means) - 1.96 * np.std(male_sample_means)
```

In [32]:
```
lowerlimit_males, upperlimit_males
```

Out[32]:
```
(8844.223303978792, 10012.979176021208)
```

In [33]:
```
upperlimit_females= np.mean(female_sample_means) + 1.96 * np.std(female_sample_means
```

In [34]:
```
lowerlimit_females= np.mean(female_sample_means) - 1.96 * np.std(female_sample_means
```

In [35]:
```
lowerlimit_females, upperlimit_females
```

Out[35]:
```
(8201.490566781567, 9279.764179885102)
```

In [36]:
```
#---- percentile method

np.percentile(male_sample_means, [0.25, 97.25])
```

Out[36]:
```
array([8607.17971667, 9999.85486667])
```

In [37]:
```
np.percentile(female_sample_means, [0.25, 97.25])
```

Out[37]:
```
array([7997.36203333, 9288.94163333])
```

In [38]:
```
erlimit_females__check= np.mean(female_sample_means) + 1.96 * np.std(female_sample_me
erlimit_females__check= np.mean(female_sample_means) - 1.96 * np.std(female_sample_me
```

In [39]:
```
lowerlimit_females__check, upperlimit_females__check
```

Out[39]:
```
(8709.500295294065, 8771.754451372604)
```

In [ ]:
```
#uncertain because the CIs are overlapping

#what can be done to eliminate the overlap

1.increase the sample size
2. reduce the Confidence level
```