In [ ]:

```
       BIVARIATE   ANALYSIS
```

In [ ]:

```
K.SASIKIRAN
MCA(R)
2019202049
```

In [1]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import math
df=pd.read_csv("suv_data.csv")
df.head(5)
```

Out[1]:

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	Male	19	19000	0
1	15810944	Male	35	20000	0
2	15668575	Female	26	43000	0
3	15603246	Female	27	57000	0
4	15804002	Male	19	76000	0

In [2]:

```
print("number of Users Purchased:  "+str(len(df.index)))
```

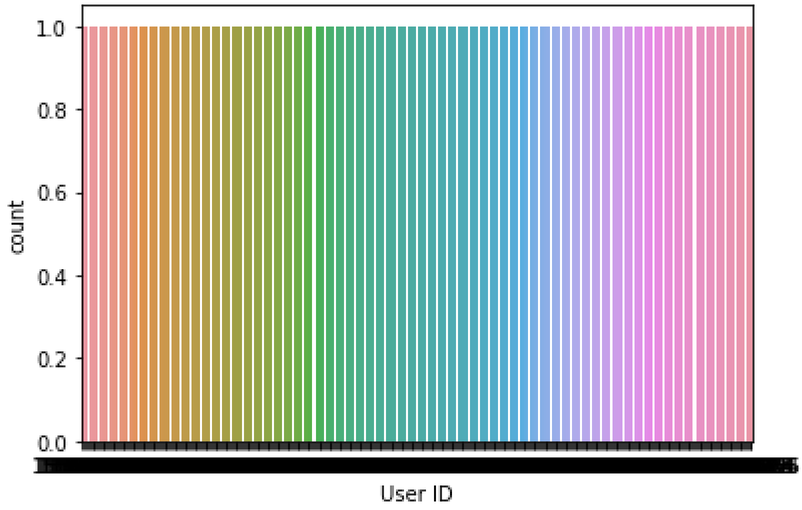
```
number of Users Purchased:  400
```

In [3]:

```
sns.countplot(x="User ID",data=df)
```

Out[3]:

```
<AxesSubplot:xlabel='User ID', ylabel='count'>
```

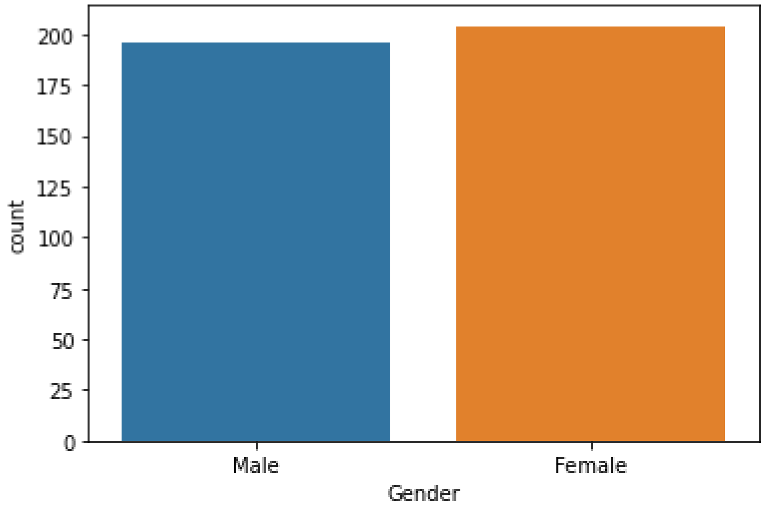


In [4]:

```
sns.countplot(x="Gender",data=df)
```

Out[4]:

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

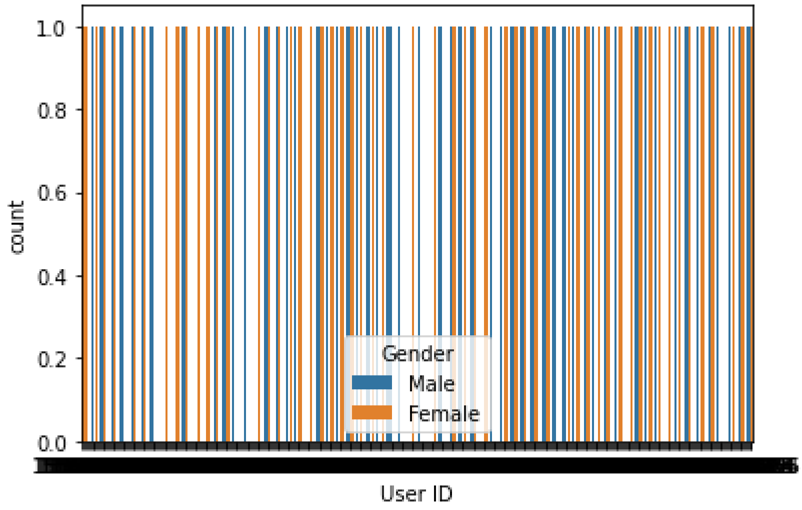


In [5]:

```
sns.countplot(x="User ID",hue="Gender",data=df)
```

Out[5]:

```
<AxesSubplot:xlabel='User ID', ylabel='count'>
```

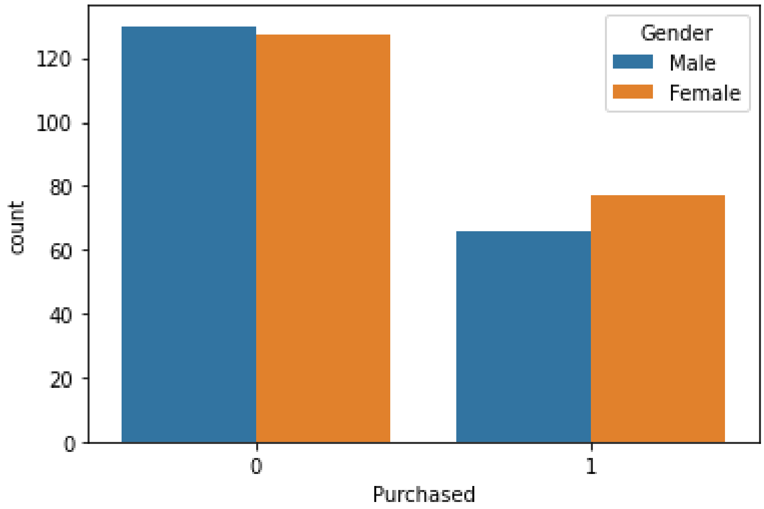


In [6]:

```
sns.countplot(x="Purchased",hue="Gender",data=df)
```

Out[6]:

```
<AxesSubplot:xlabel='Purchased', ylabel='count'>
```



In [7]:

```
df.["Age"].plot.hist()
```

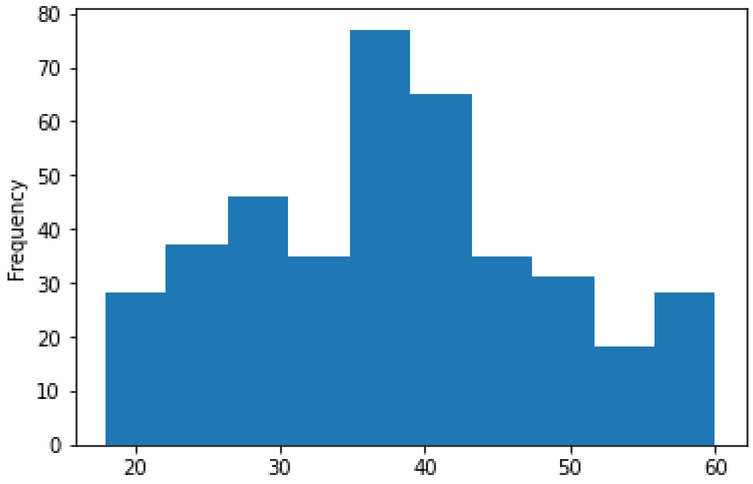
```
  File "<ipython-input-7-2a3dfb671752>", line 1
    df.["Age"].plot.hist()
       ^
SyntaxError: invalid syntax
```

In [8]:

```
df["Age"].plot.hist()
```

Out[8]:

```
<AxesSubplot:ylabel='Frequency'>
```

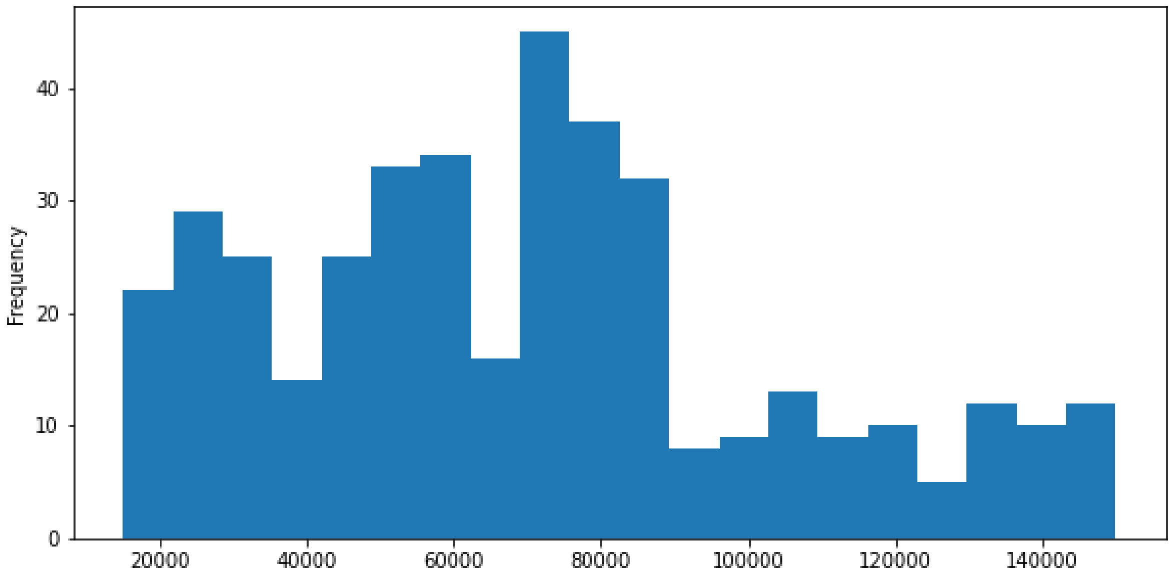


In [9]:

```
df["EstimatedSalary"].plot.hist(bins=20,figsize=(10,5))
```

Out[9]:

```
<AxesSubplot:ylabel='Frequency'>
```



In [10]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 5 columns):
 #   Column           Non-Null Count  Dtype 
---  ------           --------------  ----- 
 0   User ID          400 non-null    int64 
 1   Gender           400 non-null    object
 2   Age              400 non-null    int64 
 3   EstimatedSalary  400 non-null    int64 
 4   Purchased        400 non-null    int64 
dtypes: int64(4), object(1)
memory usage: 14.1+ KB
```

In [11]:

```
df.isnull()
```

Out[11]:

	User ID	Gender	Age	EstimatedSalary	Purchased
0	False	False	False	False	False
1	False	False	False	False	False
2	False	False	False	False	False
3	False	False	False	False	False
4	False	False	False	False	False
...	...	...	...	...	...
395	False	False	False	False	False
396	False	False	False	False	False
397	False	False	False	False	False
398	False	False	False	False	False
399	False	False	False	False	False

400 rows × 5 columns

In [13]:

```
df.isnull().sum()
```

Out[13]:

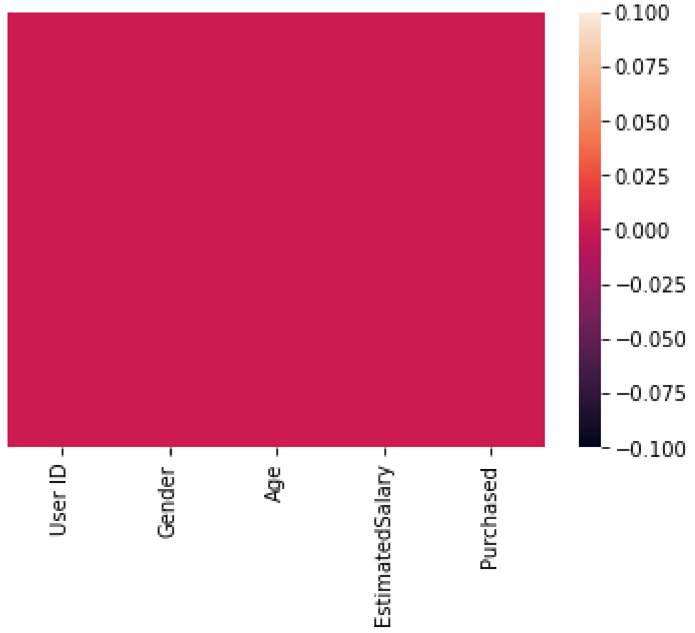
```
User ID            0
Gender             0
Age                0
EstimatedSalary    0
Purchased          0
dtype: int64
```

In [14]:

```
sns.heatmap(df.isnull() , yticklabels=False)
```

Out[14]:

```
<AxesSubplot:>
```

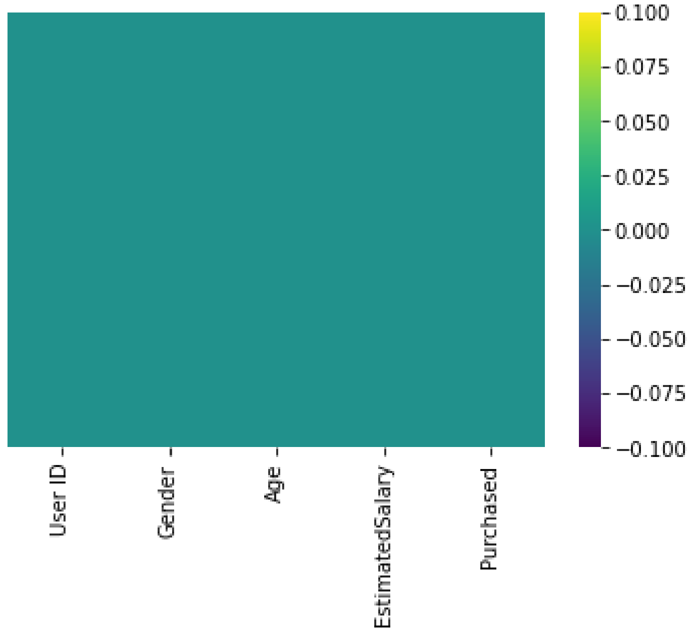


In [15]:

```
sns.heatmap(df.isnull() , yticklabels=False , cmap="viridis")
```

Out[15]:

```
<AxesSubplot:>
```

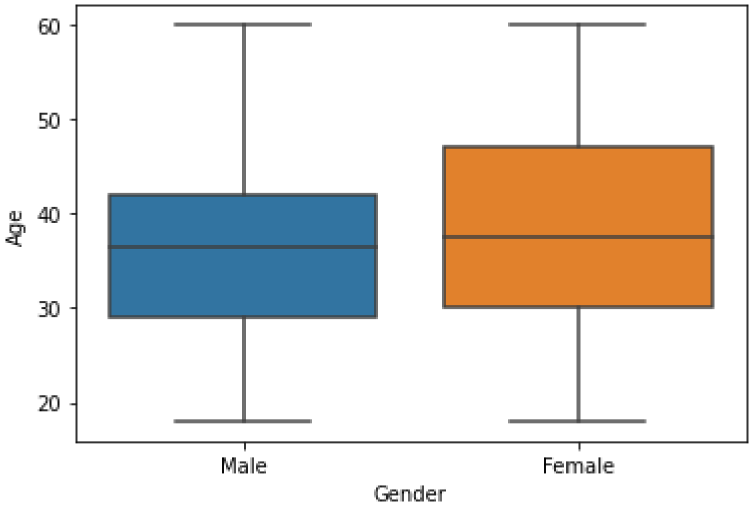


In [16]:

```
sns.boxplot(x="Gender",y="Age",data=df)
```

Out[16]:

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

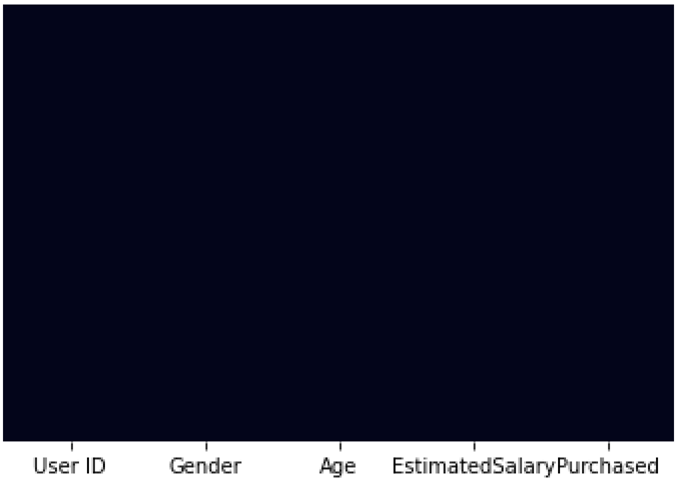


In [17]:

```
df.dropna(inplace=True)
sns.heatmap(df.isnull() , yticklabels=False , cbar=False)
```

Out[17]:

```
<AxesSubplot:>
```



In [18]:

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

Out[18]:

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	Male	19	19000	0
1	15810944	Male	35	20000	0
2	15668575	Female	26	43000	0
3	15603246	Female	27	57000	0
4	15804002	Male	19	76000	0
5	15728773	Male	27	58000	0
6	15598044	Female	27	84000	0
7	15694829	Female	32	150000	1
8	15600575	Male	25	33000	0
9	15727311	Female	35	65000	0

In [19]:

```
pd.get_dummies(df['Gender'], drop_first=True)
```

Out[19]:

	Male
0	1
1	1
2	0
3	0
4	1
...	...
395	0
396	1
397	0
398	1
399	0

400 rows × 1 columns

In [20]:

```
pd.get_dummies(df['Gender'])
```

Out[20]:

	Female	Male
0	0	1
1	0	1
2	1	0
3	1	0
4	0	1
...	...	...
395	1	0
396	0	1
397	1	0
398	0	1
399	1	0

400 rows × 2 columns

In [21]:

```
pd.get_dummies(df['Gender'] , drop_first=True)
```

Out[21]:

	Male
0	1
1	1
2	0
3	0
4	1
...	...
395	0
396	1
397	0
398	1
399	0

400 rows × 1 columns

In [23]:

```
pur=pd.get_dummies(df['Purchased'])
pur.head(10)
```

Out[23]:

	0	1
0	1	0
1	1	0
2	1	0
3	1	0
4	1	0
5	1	0
6	1	0
7	0	1
8	1	0
9	1	0

In [24]:

```
df=pd.concat([df,pur],axis=1)
```

In [25]:

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

Out[25]:

	User ID	Gender	Age	EstimatedSalary	Purchased	0	1
0	15624510	Male	19	19000	0	1	0
1	15810944	Male	35	20000	0	1	0
2	15668575	Female	26	43000	0	1	0
3	15603246	Female	27	57000	0	1	0
4	15804002	Male	19	76000	0	1	0
5	15728773	Male	27	58000	0	1	0
6	15598044	Female	27	84000	0	1	0
7	15694829	Female	32	150000	1	0	1
8	15600575	Male	25	33000	0	1	0
9	15727311	Female	35	65000	0	1	0

In [44]:

```
X=df.drop("EstimatedSalary",axis=1)
y=df["EstimatedSalary"]
```

In [45]:

```
from sklearn.model_selection import train_test_split
```

In [46]:

```
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.3,random_state=1)
```

In [47]:

```
from sklearn.linear_model import LogisticRegression
```

In [48]:

```
logmodel=LogisticRegression()
```

In [51]:

```
from sklearn.metrics import confusion_matrix
```

In [ ]:

```
-----------------WITH MY DATASET FUTURE TOP 50 RESTAURANTS-------------------
```

In [54]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import math
df=pd.read_csv("Future50.csv")
df.head(5)
```

Out[54]:

	Rank	Restaurant	Location	Sales	YOY_Sales	Units	YOY_Units	Unit_Volume	Franchisir
0	1	Evergreens	Seattle, Wash.	24	130.50%	26	116.70%	1150	N
1	2	Clean Juice	Charlotte, N.C.	44	121.90%	105	94.40%	560	Ye
2	3	Slapfish	Huntington Beach, Calif.	21	81.00%	21	90.90%	1370	Ye
3	4	Clean Eatz	Wilmington, N.C.	25	79.70%	46	58.60%	685	Ye
4	5	Pokeworks	Irvine, Calif.	49	77.10%	50	56.30%	1210	Ye

In [55]:

```
print("number of Restaurants :  "+str(len(df.index)))
```

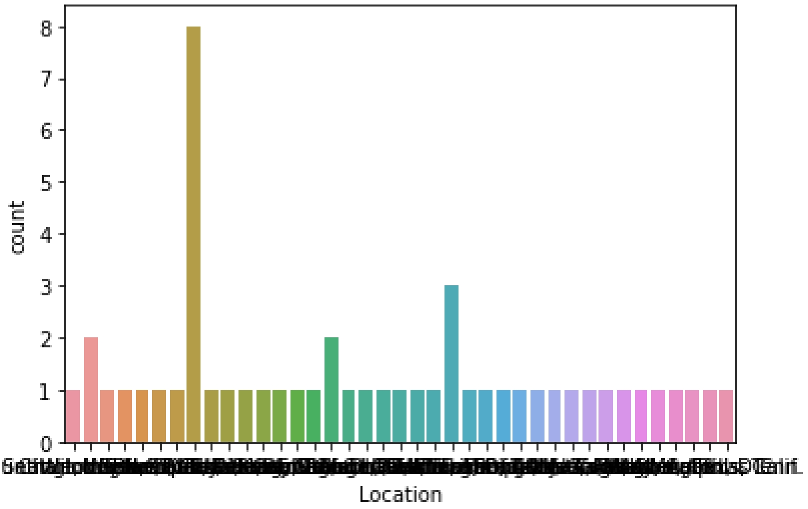
```
number of Restaurants :  50
```

In [56]:

```
sns.countplot(x="Location",data=df)
```

Out[56]:

```
<AxesSubplot:xlabel='Location', ylabel='count'>
```

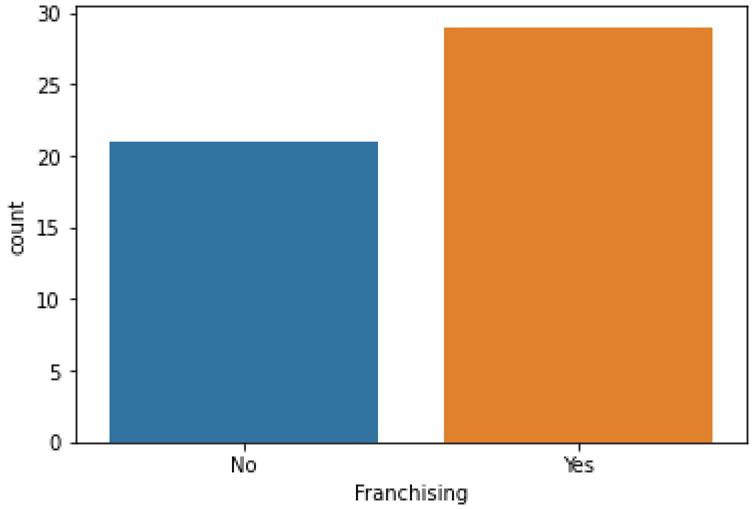


In [57]:

```
sns.countplot(x="Franchising",data=df)
```

Out[57]:

```
<AxesSubplot:xlabel='Franchising', ylabel='count'>
```

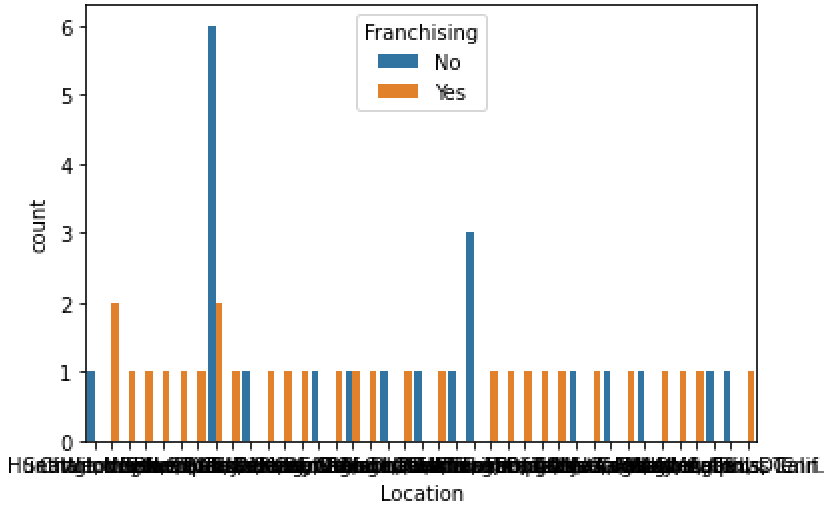


In [58]:

```
sns.countplot(x="Location",hue="Franchising",data=df)
```

Out[58]:

```
<AxesSubplot:xlabel='Location', ylabel='count'>
```

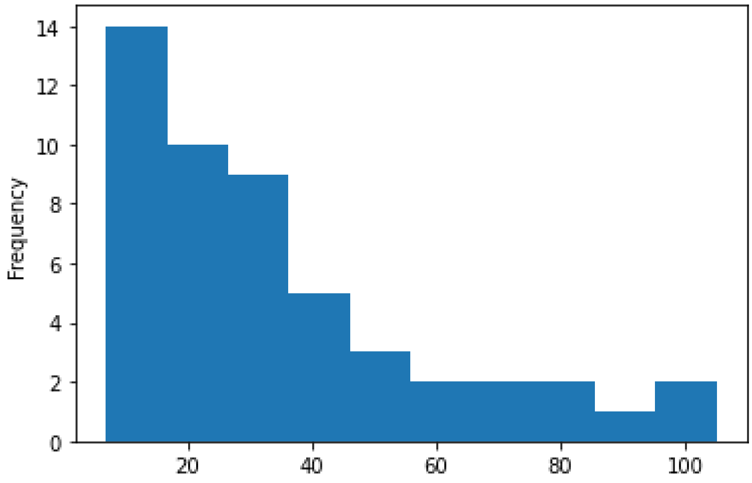


In [60]:

```
df["Units"].plot.hist()
```

Out[60]:

```
<AxesSubplot:ylabel='Frequency'>
```

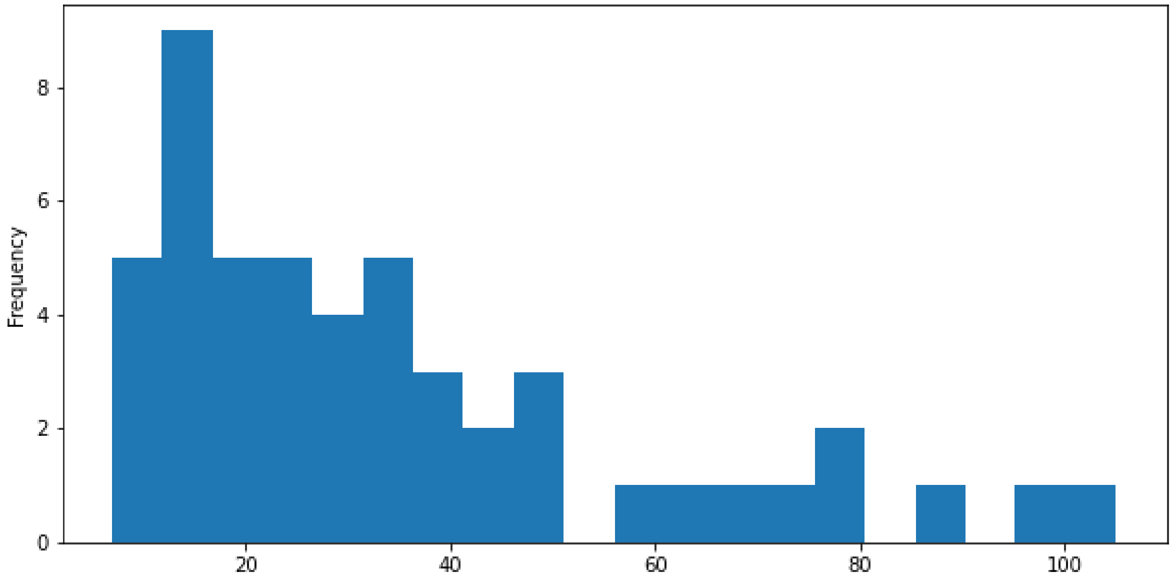


In [61]:

```
df["Units"].plot.hist(bins=20,figsize=(10,5))
```

Out[61]:

```
<AxesSubplot:ylabel='Frequency'>
```



In [62]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50 entries, 0 to 49
Data columns (total 9 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   Rank          50 non-null     int64 
 1   Restaurant    50 non-null     object
 2   Location      50 non-null     object
 3   Sales         50 non-null     int64 
 4   YOY_Sales     50 non-null     object
 5   Units         50 non-null     int64 
 6   YOY_Units     50 non-null     object
 7   Unit_Volume   50 non-null     int64 
 8   Franchising   50 non-null     object
dtypes: int64(4), object(5)
memory usage: 2.6+ KB
```

In [63]:

```
df.isnull()
```

Out[63]:

	Rank	Restaurant	Location	Sales	YOY_Sales	Units	YOY_Units	Unit_Volume	Franchising
0	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False
5	False	False	False	False	False	False	False	False	False
6	False	False	False	False	False	False	False	False	False
7	False	False	False	False	False	False	False	False	False
8	False	False	False	False	False	False	False	False	False
9	False	False	False	False	False	False	False	False	False
10	False	False	False	False	False	False	False	False	False
11	False	False	False	False	False	False	False	False	False
12	False	False	False	False	False	False	False	False	False
13	False	False	False	False	False	False	False	False	False
14	False	False	False	False	False	False	False	False	False
15	False	False	False	False	False	False	False	False	False
16	False	False	False	False	False	False	False	False	False
17	False	False	False	False	False	False	False	False	False
18	False	False	False	False	False	False	False	False	False
19	False	False	False	False	False	False	False	False	False
20	False	False	False	False	False	False	False	False	False
21	False	False	False	False	False	False	False	False	False
22	False	False	False	False	False	False	False	False	False
23	False	False	False	False	False	False	False	False	False
24	False	False	False	False	False	False	False	False	False
25	False	False	False	False	False	False	False	False	False
26	False	False	False	False	False	False	False	False	False
27	False	False	False	False	False	False	False	False	False
28	False	False	False	False	False	False	False	False	False
29	False	False	False	False	False	False	False	False	False
30	False	False	False	False	False	False	False	False	False
31	False	False	False	False	False	False	False	False	False
32	False	False	False	False	False	False	False	False	False
33	False	False	False	False	False	False	False	False	False

	Rank	Restaurant	Location	Sales	YOY_Sales	Units	YOY_Units	Unit_Volume	Franchisin
34	False	False	False	False	False	False	False	False	Fals
35	False	False	False	False	False	False	False	False	Fals
36	False	False	False	False	False	False	False	False	Fals
37	False	False	False	False	False	False	False	False	Fals
38	False	False	False	False	False	False	False	False	Fals
39	False	False	False	False	False	False	False	False	Fals
40	False	False	False	False	False	False	False	False	Fals
41	False	False	False	False	False	False	False	False	Fals
42	False	False	False	False	False	False	False	False	Fals
43	False	False	False	False	False	False	False	False	Fals
44	False	False	False	False	False	False	False	False	Fals
45	False	False	False	False	False	False	False	False	Fals
46	False	False	False	False	False	False	False	False	Fals
47	False	False	False	False	False	False	False	False	Fals
48	False	False	False	False	False	False	False	False	Fals
49	False	False	False	False	False	False	False	False	Fals

In [64]:

```
df.isnull().sum()
```

Out[64]:

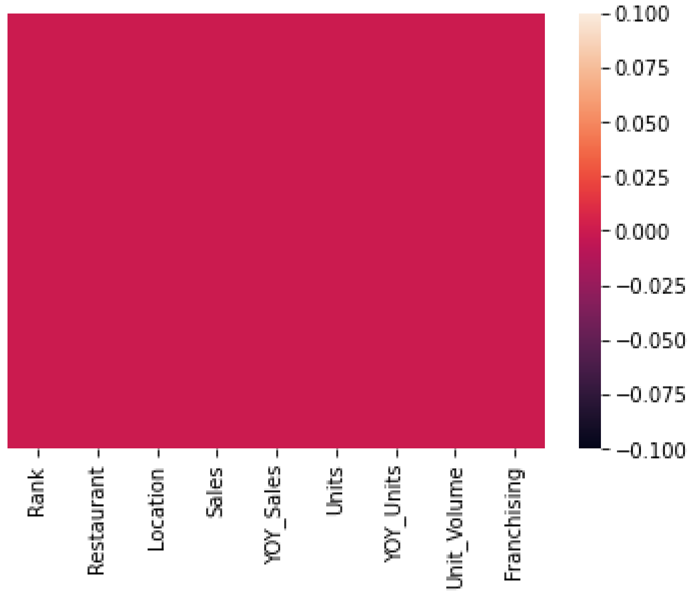
```
Rank           0
Restaurant     0
Location       0
Sales          0
YOY_Sales      0
Units          0
YOY_Units      0
Unit_Volume    0
Franchising    0
dtype: int64
```

In [65]:

```
sns.heatmap(df.isnull() , yticklabels=False)
```

Out[65]:

```
<AxesSubplot:>
```

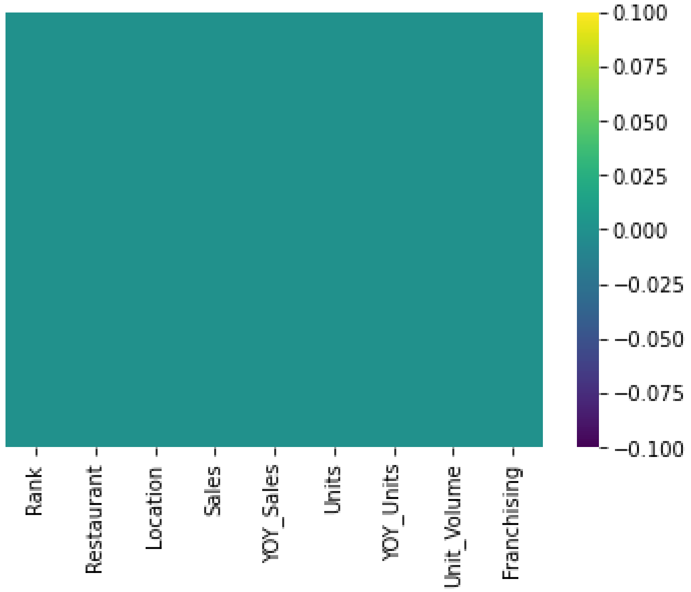


In [66]:

```
sns.heatmap(df.isnull() , yticklabels=False , cmap="viridis")
```

Out[66]:

```
<AxesSubplot:>
```

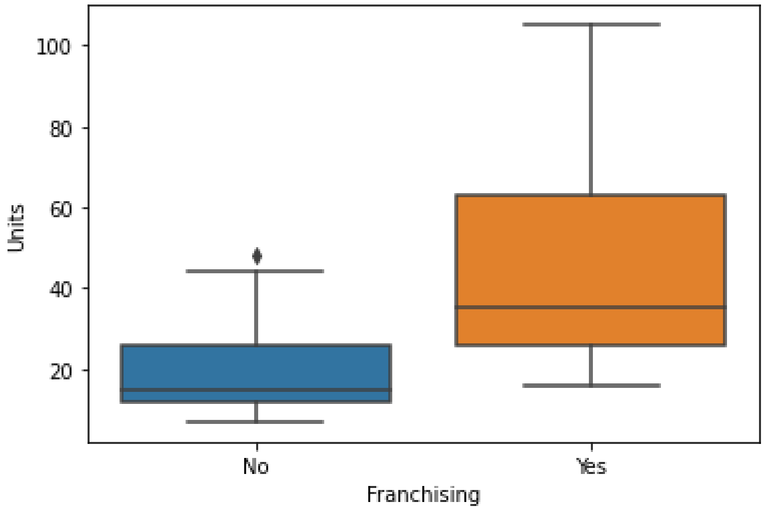


In [68]:

```
sns.boxplot(x="Franchising",y="Units",data=df)
```

Out[68]:

```
<AxesSubplot:xlabel='Franchising', ylabel='Units'>
```

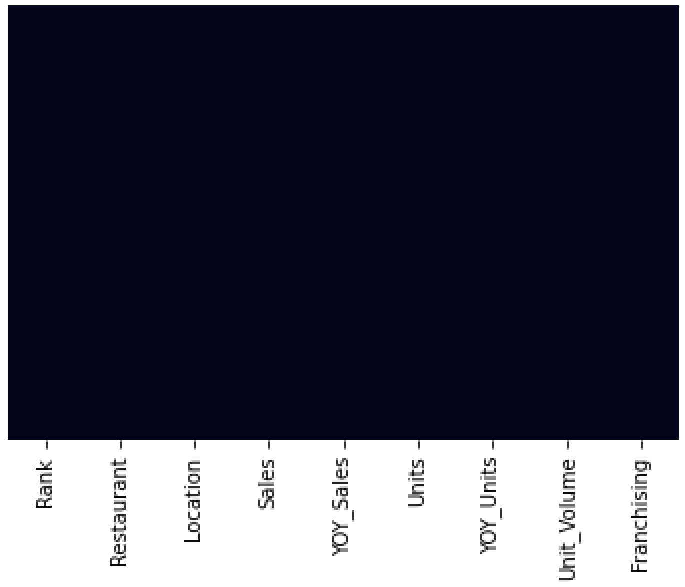


In [69]:

```
df.dropna(inplace=True)
sns.heatmap(df.isnull() , yticklabels=False , cbar=False)
```

Out[69]:

```
<AxesSubplot:>
```



In [70]:

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

Out[70]:

	Rank	Restaurant	Location	Sales	YOY_Sales	Units	YOY_Units	Unit_Volume	Franchisir
0	1	Evergreens	Seattle, Wash.	24	130.50%	26	116.70%	1150	N
1	2	Clean Juice	Charlotte, N.C.	44	121.90%	105	94.40%	560	Ye
2	3	Slapfish	Huntington Beach, Calif.	21	81.00%	21	90.90%	1370	Ye
3	4	Clean Eatz	Wilmington, N.C.	25	79.70%	46	58.60%	685	Ye
4	5	Pokeworks	Irvine, Calif.	49	77.10%	50	56.30%	1210	Ye
5	6	Playa Bowls	Belmar, N.J.	39	62.90%	76	28.80%	580	Ye
6	7	The Simple Greek	Blue Bell, Pa.	24	52.50%	36	33.30%	775	Ye
7	8	Melt Shop	New York, N.Y.	20	39.60%	19	35.70%	1260	Ye
8	9	Creamistry	Yorba Linda, Calif.	24	36.80%	60	27.70%	465	Ye
9	10	Joella's Hot Chicken	Louisville, Ky.	29	35.50%	17	30.80%	1930	N

In [71]:

```
pd.get_dummies(df['Franchising'], drop_first=True)
```

Out[71]:

```
    Yes
 0    0
 1    1
 2    1
 3    1
 4    1
 5    1
 6    1
 7    1
 8    1
 9    0
10    1
11    1
12    1
13    0
14    0
15    1
16    1
17    1
18    0
19    1
20    0
21    1
22    0
23    1
24    0
25    0
26    0
27    1
28    0
29    1
30    1
31    0
32    0
33    0
```

```
    Yes
34    1
35    1
36    1
37    0
38    0
39    1
40    0
41    1
42    0
43    1
44    1
45    1
46    0
47    0
48    0
49    1
```

In [72]:

```
pd.get_dummies(df['Franchising'])
```

Out[72]:

	No	Yes
0	1	0
1	0	1
2	0	1
3	0	1
4	0	1
5	0	1
6	0	1
7	0	1
8	0	1
9	1	0
10	0	1
11	0	1
12	0	1
13	1	0
14	1	0
15	0	1
16	0	1
17	0	1
18	1	0
19	0	1
20	1	0
21	0	1
22	1	0
23	0	1
24	1	0
25	1	0
26	1	0
27	0	1
28	1	0
29	0	1
30	0	1
31	1	0
32	1	0
33	1	0

	No	Yes
34	0	1
35	0	1
36	0	1
37	1	0
38	1	0
39	0	1
40	1	0
41	0	1
42	1	0
43	0	1
44	0	1
45	0	1
46	1	0
47	1	0
48	1	0
49	0	1

In [73]:

```
pd.get_dummies(df['Franchising'] , drop_first=True)
```

Out[73]:

	Yes
0	0
1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	1
9	0
10	1
11	1
12	1
13	0
14	0
15	1
16	1
17	1
18	0
19	1
20	0
21	1
22	0
23	1
24	0
25	0
26	0
27	1
28	0
29	1
30	1
31	0
32	0
33	0

	Yes
34	1
35	1
36	1
37	0
38	0
39	1
40	0
41	1
42	0
43	1
44	1
45	1
46	0
47	0
48	0
49	1

In [74]:

```
pur=pd.get_dummies(df['Location'])
pur.head(10)
```

Out[74]:

	Agoura Hills, Calif.	Anaheim, Calif.	Atlanta, Ga.	Belmar, N.J.	Blue Bell, Pa.	Charlotte, N.C.	Columbus, Ohio	Conway, Ark.	Denver, Colo.	Doral, Fla.
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	1	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	1	0	0	0	0	0	0
6	0	0	0	0	1	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0

10 rows × 39 columns

In [75]:

```
df=pd.concat([df,pur],axis=1)
```

In [76]:

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

Out[76]:

	Rank	Restaurant	Location	Sales	YOY_Sales	Units	YOY_Units	Unit_Volume	Franchisir
0	1	Evergreens	Seattle, Wash.	24	130.50%	26	116.70%	1150	N
1	2	Clean Juice	Charlotte, N.C.	44	121.90%	105	94.40%	560	Ye
2	3	Slapfish	Huntington Beach, Calif.	21	81.00%	21	90.90%	1370	Ye
3	4	Clean Eatz	Wilmington, N.C.	25	79.70%	46	58.60%	685	Ye
4	5	Pokeworks	Irvine, Calif.	49	77.10%	50	56.30%	1210	Ye
5	6	Playa Bowls	Belmar, N.J.	39	62.90%	76	28.80%	580	Ye
6	7	The Simple Greek	Blue Bell, Pa.	24	52.50%	36	33.30%	775	Ye
7	8	Melt Shop	New York, N.Y.	20	39.60%	19	35.70%	1260	Ye
8	9	Creamistry	Yorba Linda, Calif.	24	36.80%	60	27.70%	465	Ye
9	10	Joella's Hot Chicken	Louisville, Ky.	29	35.50%	17	30.80%	1930	N

10 rows × 48 columns

In [ ]: