In [1]:

```python
import pandas as pd
import numpy as np
import scipy as sp
from scipy import stats
%matplotlib inline
import matplotlib.pyplot as plt
import statsmodels.api as smf
import statsmodels.formula.api as smf
import matplotlib as mp1
import seaborn as sns
from statsmodels.stats.proportion import proportions_ztest
```

In [2]:

```python
#Read CSV
Cutlets=pd.read_csv("C:/Users/rjas/Downloads/ds assignments/3a/Cutlets.csv")
```

In [3]:

```python
Cutlets
```

Out[3]:

	Unit A	Unit B
0	6.8090	6.7703
1	6.4376	7.5093
2	6.9157	6.7300
3	7.3012	6.7878
4	7.4488	7.1522
5	7.3871	6.8110
6	6.8755	7.2212
7	7.0621	6.6606
8	6.6840	7.2402
9	6.8236	7.0503
10	7.3930	6.8810
11	7.5169	7.4059
12	6.9246	6.7652
13	6.9256	6.0380
14	6.5797	7.1581
15	6.8394	7.0240
16	6.5970	6.6672
17	7.2705	7.4314
18	7.2828	7.3070
19	7.3495	6.7478
20	6.9438	6.8889
21	7.1560	7.4220
22	6.5341	6.5217
23	7.2854	7.1688
24	6.9952	6.7594
25	6.8568	6.9399
26	7.2163	7.0133
27	6.6801	6.9182
28	6.9431	6.3346
29	7.0852	7.5459
30	6.7794	7.0992
31	7.2783	7.1180
32	7.1561	6.6965
33	7.3943	6.5780
34	6.9405	7.3875

In [ ]:

In [4]:

```python
Cutlets.dtypes
```

Out[4]:

```
Unit A    float64
Unit B    float64
dtype: object
```

In [5]:

```python
Cutlets.size
```

Out[5]: 70

In [6]:

```python
Cutlets.shape
```

Out[6]: (35, 2)

In [7]:

```python
Cutlets.columns
```

Out[7]: Index(['Unit A', 'Unit B'], dtype='object')

In [8]:

```python
Cutlets.axes
```

Out[8]:

```
[RangeIndex(start=0, stop=35, step=1),
 Index(['Unit A', 'Unit B'], dtype='object')]
```

In [9]:

```python
Cutlets.ndim
```

Out[9]: 2

In [10]:

```python
Cutlets.values
```

Out[10]:

```
array([[6.809 , 6.7703],
       [6.4376, 7.5093],
       [6.9157, 6.73  ],
       [7.3012, 6.7878],
       [7.4488, 7.1522],
       [7.3871, 6.811 ],
       [6.8755, 7.2212],
       [7.0621, 6.6606],
       [6.684 , 7.2402],
       [6.8236, 7.0503],
       [7.393 , 6.881 ],
       [7.5169, 7.4059],
       [6.9246, 6.7652],
       [6.9256, 6.038 ],
       [6.5797, 7.1581],
       [6.8394, 7.024 ],
       [6.597 , 6.6672],
       [7.2705, 7.4314],
       [7.2828, 7.307 ],
       [7.3495, 6.7478],
       [6.9438, 6.8889],
       [7.156 , 7.422 ],
       [6.5341, 6.5217],
       [7.2854, 7.1688],
       [6.9952, 6.7594],
       [6.8568, 6.9399],
       [7.2163, 7.0133],
       [6.6801, 6.9182],
       [6.9431, 6.3346],
       [7.0852, 7.5459],
       [6.7794, 7.0992],
       [7.2783, 7.118 ],
       [7.1561, 6.6965],
       [7.3943, 6.578 ],
       [6.9405, 7.3875]])
```

In [11]:

```python
Cutlets['Unit A'].value_counts()
```

Out[11]:

```
7.2828    1
6.9256    1
7.5169    1
6.8236    1
7.2783    1
6.5341    1
6.8755    1
7.1561    1
7.0621    1
6.9405    1
6.6801    1
6.9438    1
7.0852    1
7.2705    1
6.5797    1
6.6840    1
7.3943    1
6.5970    1
6.4376    1
6.9952    1
6.9157    1
7.4488    1
7.3871    1
6.9431    1
7.2854    1
7.3012    1
7.3495    1
7.1560    1
6.7794    1
6.8568    1
7.3930    1
6.8090    1
7.2163    1
6.9246    1
6.8394    1
Name: Unit A, dtype: int64
```

In [12]:

```python
Cutlets.describe()
```

Out[12]:

	Unit A	Unit B
count	35.000000	35.000000
mean	7.019091	6.964297
std	0.288408	0.343401
min	6.437600	6.038000
25%	6.831500	6.753600
50%	6.943800	6.939900
75%	7.280550	7.195000
max	7.516900	7.545900

In [13]:

```python
Cutlets.mean()
```

Out[13]:

```
Unit A    7.019091
Unit B    6.964297
dtype: float64
```

In [14]:

```python
Cutlets.std()
```

Out[14]:

```
Unit A    0.288408
Unit B    0.343401
dtype: float64
```

In [15]:

```python
Cutlets.median()
```

Out[15]:

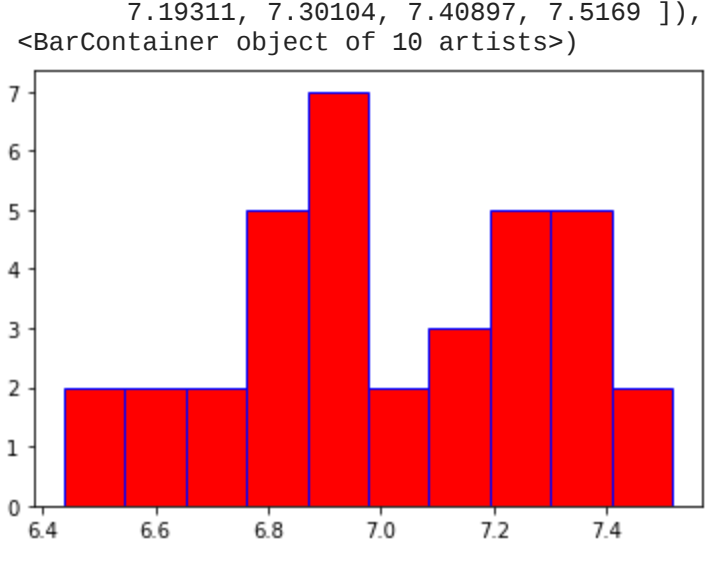
```
Unit A    6.9438
Unit B    6.9399
dtype: float64
```

In [16]:

```python
plt.hist(Cutlets["Unit A"],facecolor="red",edgecolor="blue",bins=10)
```

Out[16]:

```
(array([2., 2., 2., 5., 7., 2., 3., 5., 5., 2.]),
 array([6.4376 , 6.54553, 6.65346, 6.76139, 6.86932, 6.97725, 7.08518,
        7.19311, 7.30104, 7.40897, 7.5169 ]),
 <BarContainer object of 10 artists>)
```



In [17]:

```python
Cutlets_CI_B=stats.norm.interval(0.975,loc=6.96429,scale=0.3434)
```

In [18]:

```python
Cutlets_CI_B
```

Out[18]: (6.194592303340462, 7.733987696659539)

In [21]:

```python
Cutlets_CI_A=stats.norm.interval(0.975,loc=7.01909,scale=0.2884)
```

In [22]:

```python
Cutlets_CI_A
```

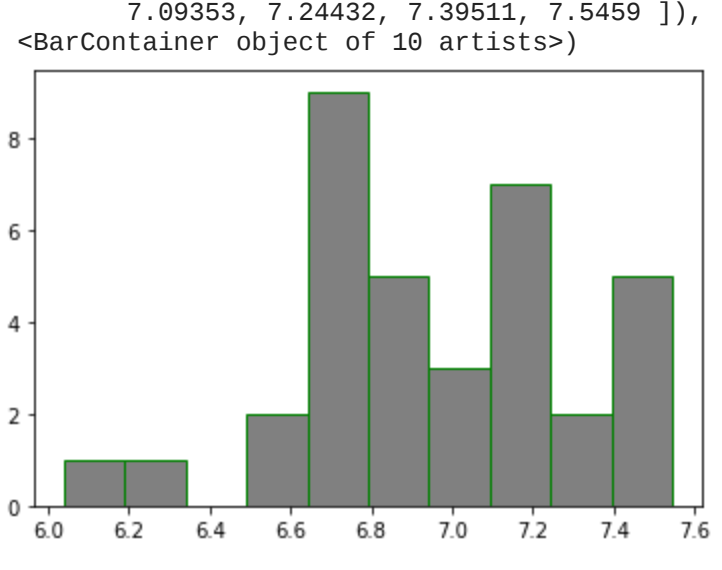
Out[22]: (6.372669453358734, 7.665510546641267)

In [23]:

```python
plt.hist(Cutlets["Unit B"],facecolor="grey",edgecolor="green",bins=10)
```

Out[23]:

```
(array([1., 1., 0., 2., 9., 5., 3., 7., 2., 5.]),
 array([6.038  , 6.18879, 6.33958, 6.49037, 6.64116, 6.79195, 6.94274,
        7.09353, 7.24432, 7.39511, 7.5459 ]),
 <BarContainer object of 10 artists>)
```



In [26]:

```python
Unit_A=Cutlets['Unit A'].mean()
Unit_B=Cutlets['Unit B'].mean()
print('Unit A Mean= ',Unit_A, '\nUnit B Mean = ',Unit_B)
print('Unit A Mean > Unit_B Mean = ' ,Unit_A>Unit_B)
```

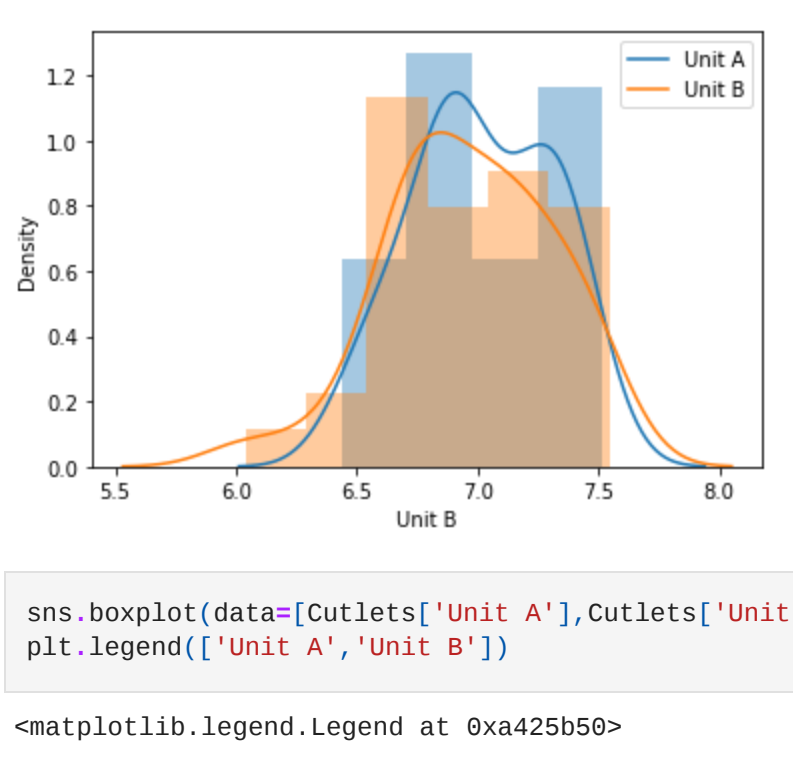
```
Unit A Mean=  7.01909142857143
Unit B Mean =  6.964297142857142
Unit A Mean > Unit_B Mean =  True
```

In [27]:

```python
sns.distplot(Cutlets['Unit A'])
sns.distplot(Cutlets['Unit B'])
plt.legend(['Unit A','Unit B'])
```

```
C:\Users\rjas\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\rjas\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

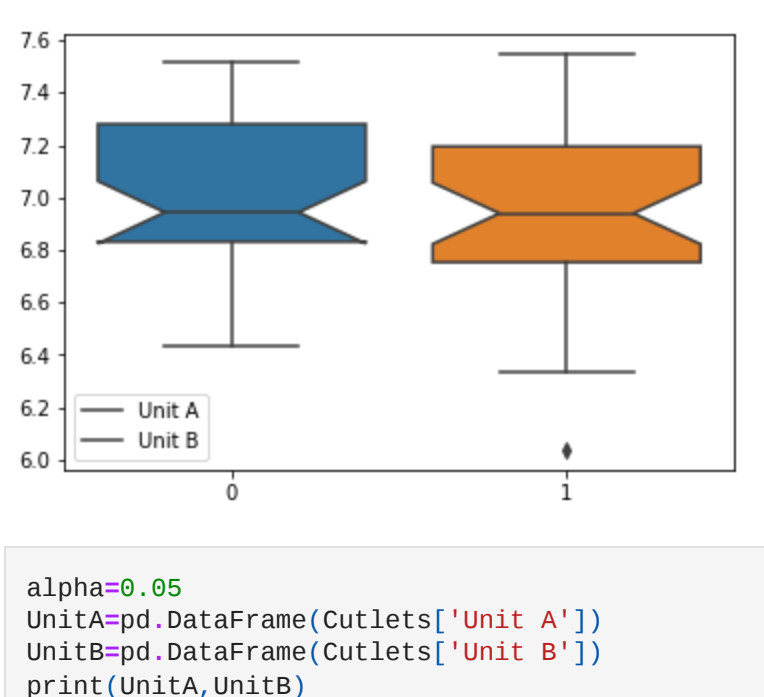
Out[27]: <matplotlib.legend.Legend at 0x56fbe50>



In [29]:

```python
sns.boxplot(data=[Cutlets['Unit A'],Cutlets['Unit B']],notch=True)
plt.legend(['Unit A','Unit B'])
```

Out[29]: <matplotlib.legend.Legend at 0xa425b50>



In [30]:

```python
alpha=0.05
UnitA=pd.DataFrame(Cutlets['Unit A'])
UnitB=pd.DataFrame(Cutlets['Unit B'])
print(UnitA,UnitB)
```

```
    Unit A
0   6.8090
1   6.4376
2   6.9157
3   7.3012
4   7.4488
5   7.3871
6   6.8755
7   7.0621
8   6.6840
9   6.8236
10  7.3930
11  7.5169
12  6.9246
13  6.9256
14  6.5797
15  6.8394
16  6.5970
17  7.2705
18  7.2828
19  7.3495
20  6.9438
21  7.1560
22  6.5341
23  7.2854
24  6.9952
25  6.8568
26  7.2163
27  6.6801
28  6.9431
29  7.0852
30  6.7794
31  7.2783
32  7.1561
33  7.3943
34  6.9405     Unit B
0   6.7703
1   7.5093
2   6.7300
3   6.7878
4   7.1522
5   6.8110
6   7.2212
7   6.6606
8   7.2402
9   7.0503
10  6.8810
11  7.4059
12  6.7652
13  6.0380
14  7.1581
15  7.0240
16  6.6672
17  7.4314
18  7.3070
19  6.7478
20  6.8889
21  7.4220
22  6.5217
23  7.1688
24  6.7594
25  6.9399
26  7.0133
27  6.9182
28  6.3346
29  7.5459
30  7.0992
31  7.1180
32  6.6965
33  6.5780
34  7.3875
```

In [34]:

```python
tStat,pValue =sp.stats.ttest_ind(UnitA,UnitB)
print("p-Value:{0} T-Statistic:{1}".format(pValue,tStat))
```

```
p-Value:[0.47223947] T-Statistic:[0.72286887]
```

In [37]:

```python
if pValue <0.05:
  print('we reject null hypothesis')
else:
  print('we accept null hypothesis')
```

```
we accept null hypothesis
```

Inference is that there is no significant difference in the diameters of Unit A and Unit B