In [1]:
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import scipy
from scipy import stats
import statsmodels.api as sm
import pylab
import scipy.stats as stats
```

In [2]:
```python
cutlets = pd.read_csv ('/Users/acer/Sandesh Pal/Data Science Assgn/Hypothesis/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 [4]:
```python
cutlets.shape
```

Out[4]:
```
(35, 2)
```

Since the Unit A and Unit B are two different samples put together in a single dataset, hence we'll split them into two separate datasets

In [5]:
```python
unitA = cutlets['Unit A']
```

In [6]:
```python
unitB = cutlets['Unit B']
```

In [7]:
```python
unitA.describe()
```

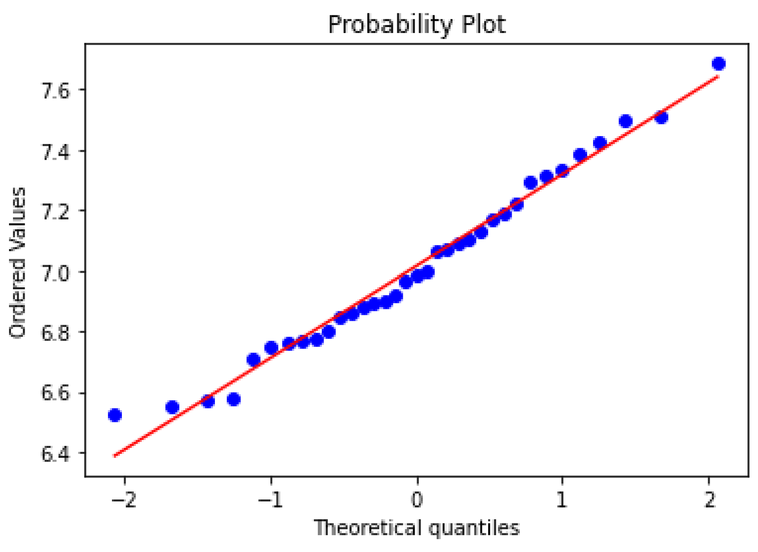
Out[7]:
```
count    35.000000
mean      7.019091
std       0.288408
min       6.437600
25%       6.831500
50%       6.943800
75%       7.280550
max       7.516900
Name: Unit A, dtype: float64
```

In [8]:
```python
unitB.describe()
```

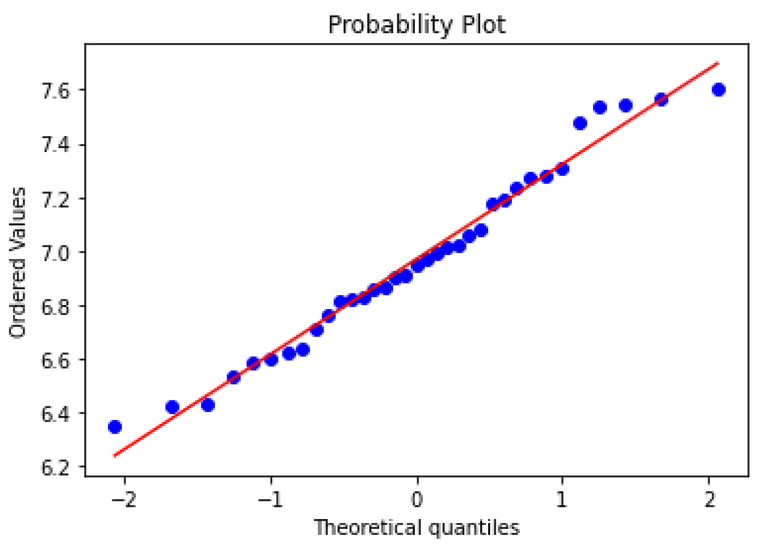
Out[8]:
```
count    35.000000
mean      6.964297
std       0.343401
min       6.038000
25%       6.753600
50%       6.939900
75%       7.195000
max       7.545900
Name: Unit B, dtype: float64
```

Checking if the samples are normally distributed

In [9]:
```python
measurements = np.random.normal(loc = 7.019091, scale =0.288408 , size=35)
stats.probplot(measurements, dist="norm", plot=pylab)
pylab.show()
```



In [10]:
```python
measurements = np.random.normal(loc = 6.964297, scale =  0.343401 , size=35)
stats.probplot(measurements, dist="norm", plot=pylab)
pylab.show()
```



By looking at their qqplots, it is evident that the samples are approximately normally distributed

Hence, now we'll proceed for sample t- test is equality of Means test. Sample Mean will tell us which program is better.

Ho= Averages of diameters of Unit A is equal to Averages of diameters of unit B

Ha= Averages of diameters of Unit A is not equal to Averages of diameters of unit B

In [11]:
```python
stats.ttest_ind (unitA, unitB)
```

Out[11]:
```
Ttest_indResult(statistic=0.7228688704678061, pvalue=0.4722394724599501)
```

In [12]:
```python
#Since pvalue(=0.47) > alpha(=0.05), hence we can't reject the null hypothesis
```

In [13]:
```python
#Conclusion: there is no significant difference in the diameters of Unit A and Unit B at 5% significance level
```

In [ ]: