# APS7 - Vitor Liu

In [1]:

```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
from math import *
from scipy import stats
%matplotlib inline
```

# Questão 1

In [2]:

```
a = [0.8411, 0.8191, 0.8182, 0.8125, 0.8580, 0.8532, 0.8483, 0.8276, 0.8042, 0.8730,
0.8282, 0.8359, 0.8750, 0.7983, 0.8660]
n = len(a)
mi0 = 0.82
alp = 0.05
```

In [3]:

```
a_ = np.mean(a)
s = np.std(a,ddof=1)

print("x linha = {}; s = {}" .format(a_, s))
```

```
x linha = 0.83724; s = 0.024557099642611345
```

## a) x linha = 0.837; s = 0.024

## b) Hipótese nula (H0): média = 0,82.

## Hipótese alternativa: média > 0,82.

## O teste de hipótese vai ter região unicaudal

In [4]:

```
t = (a_-mi0)/(s/sqrt(n))
print("t_obs = {}" .format(t))
```

```
t_obs = 2.718978782525142
```

## c) t_obs = 2.718

In [5]:

```
tc = stats.t.ppf(1-alp, loc = 0, scale = 1, df = n-1)
print("t_critico = {}" .format(tc))
```

```
t_critico = 1.7613101357748562
```

## d) Como t_obs (2.718) é maior que o t_crítico (1.761), é possível afirmar que a hipótese nula é falsa

In [6]:

```
pt = 1 - stats.t.cdf(t, loc = 0, scale =1, df = n-1)
print("p_t = {}" .format(pt))
```

```
p_t = 0.008313368681510891
```

## e) Sabendo que o valor de alpha é 0.05, ou 5%, o valor do valor p de t_obs, que é 0.008, ou 0.8% (menor que alpha), comprova a conclusão do item D.

In [7]:

```
p_alp = 1 - stats.t.cdf(a_, loc = mi0, scale = s/sqrt(n), df = n-1)
p_alp
```

Out[7]:

```
0.0083133686815108909
```

In [8]:

```
ac = stats.t.ppf(1- alp,loc = mi0, scale = s/sqrt(n), df = n-1)
ac
```

Out[8]:

```
0.83116779098679039
```

## f) Refazendo o teste de hipótese utilizando os valores não padronizados, o x critico é menor que o x médio, e o valor de p de x é menor que alpha.

# Questão 2

In [9]:

```
x = [23.01, 22.22, 22.04, 22.62, 22.59]
n = len(x)
alp = 0.05
mi0 = 22.5
```

In [10]:

```
s = np.std(x,ddof=1)
x_ = np.mean(x)
print(x_)

p_x = stats.t.cdf(x_, loc = mi0, scale = s/sqrt(n), df = n-1)
print(p_x)
```

```
22.496
0.491135375192
```

# a) A hipótese nula (H0: mi0 = 22.5) é verdadeira pois, como a média da amostra é menor que a média dada, o valor p da média amostral é maior que o valor de alpha.

In [11]:

```
xc = stats.t.ppf(1- alp/2,loc = mi0, scale = s/sqrt(n), df = n-1), stats.t.ppf(alp/2,loc =
xc
```

Out[11]:

```
(22.969752851090053, 22.030247148909947)
```

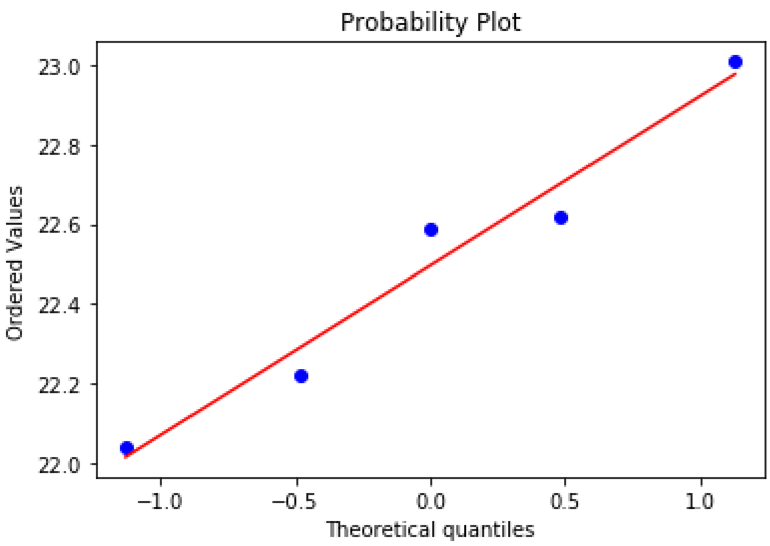
# b) Valores críticos = 22.03 e 22.969; Zona de rejeição < 22.03 e > 22.969

In [12]:

```
stats.probplot(x, dist='norm', plot=plt)
```

Out[12]:

```
((array([-1.12899754, -0.48565271,  0.        ,  0.48565271,  1.12899754]),
  array([ 22.04,  22.22,  22.59,  22.62,  23.01])),
 (0.42681026992664972, 22.496000000000002, 0.98042421170935712))
```




# c) A distribuição da variável pode ser considerada uma

## distribuição pois o valor do coeficiente de determinação (R quadrado) é de 0.9612 (0.980424^2). Portanto, o teste t pode ser feito para essa variável.

# Questão 3

In [13]:

```
x = [129.26, 204.49, 116.89, 106.4 , 95.3 , 123.35, 92.3 , 300.02, 264.34, 168.27, 80.02, 1
n = len(x)
x_ = np.mean(x)
s = np.std(x)
x_, s
```

Out[13]:

```
(176.8377777777778, 107.82680977462347)
```

In [14]:

```
alp = 5

l = []
for i in range(20000):
    l.append(np.random.choice(x, size=n, replace=True).mean())

xc = stats.scoreatpercentile(l, alp), stats.scoreatpercentile(l, 100-alp)
xc
```

Out[14]:

```
(136.99994444444445, 221.22277777777774)
```

## a) Para a média: Limite inferior = 136.999; Limite superior = 221.222

In [15]:

```
alp = 5

l = []
for i in range(20000):
    l.append(np.random.choice(x, size=n, replace=True).std())

xc = stats.scoreatpercentile(l, alp), stats.scoreatpercentile(l, 100-alp)
xc
```

Out[15]:

```
(64.101178269399753, 135.31894675548094)
```

## b) Para o desvio padrão: Limite inferior = 64.101; Limite superior = 135.318.