## Importing modules

In [53]:

```
import numpy as np
import statsmodels
import scipy.stats as sts
import matplotlib.pyplot as plt
```

In [5]:

```
norm = sts.norm
```

# Q2

In [152]:

```
u = 12
X = [13.51, 10.51, 13.71, 13.8, 10.18, 11.42, 9.99, 13.51]
```

In [145]:

```
X_ = np.mean(X); X_
```

Out[145]:

```
12.07875
```

In [146]:

```
sig2 = np.var(X)
sig = sig2**0.5; sig
```

Out[146]:

```
1.6040451793824264
```

In [ ]:

```
Ho: u=12
H1: u>12
```

In [147]:

```
u=12

n = len(X)
alpha=0.02
```

In [148]:

```
n
```

Out[148]:

8

***Using the z-test and p-value***

In [149]:

```
sig_ = sig / n**0.5 ; sig_
```

Out[149]:

0.5671156118354529

In [150]:

```
z_ = (X_ - u) / sig_ ; z_
```

Out[150]:

0.1388605750864932

In [143]:

```
z_a = norm.ppf(1-0.02, 0, 1) ; z_a
```

Out[143]:

2.0537489106318225

In [62]:

```
# p = P(Z > z_)
p = 1 - norm.cdf(x=z_, loc=0, scale=1) ; p
```

Out[62]:

0.4447801627966188

***valor de x limite para rejeição***

In [68]:

```
z_lim = norm.ppf(1-alpha, loc=0, scale=1) ; z_lim
```

Out[68]:

2.0537489106318225

In [64]:

```
x_lim = (z_lim * sig_) + u ; x_lim
```

Out[64]:

12.932822171028349

# Q2 b) Pag 195

In [74]:

```
alpha = 0.01
```

In [80]:

```
alpha/2
```

Out[80]:

```
0.005
```

In [81]:

```
1 - alpha/2
```

Out[81]:

```
0.995
```

In [82]:

```
v = n-1 ; v
```

Out[82]:

```
7
```

In [83]:

```
s_2 = (n * sum([i**2 for i in X]) - sum(X)**2) / (n*(n-1)); s_2
```

Out[83]:

```
2.940526785714318
```

In [153]:

```
sum([i**2 for i in X])
```

Out[153]:

```
1187.7533
```

In [154]:

```
sum(X)**2
```

Out[154]:

```
9337.356899999999
```

In [84]:

```
X_2 = 0.989
```

In [85]:

```
_X_2 = 20.278
```

In [86]:

```
(n-1) * s_2 / X_2
```

Out[86]:

```
20.812626390293453
```

In [87]:

```
(n-1) * s_2 / _X_2
```

Out[87]:

```
1.015074834796342
```

# Q3

In [171]:

```
n = 150
S = 57

e = 0.08
alpha = 0.1
```

In [159]:

```
p_ = S / n ;
q_ = 1 - p_; p_
```

Out[159]:

```
0.38
```

In [160]:

```
n * p_, n*(1-p_)
```

Out[160]:

```
(57.0, 93.0)
```

n*p e n*(1-5) devem ser maiores que 5

In [168]:

```
z_a2 = norm.ppf(1-alpha, 0, 1) ; z_a2
```

Out[168]:

```
1.2815515655446004
```

In [169]:

```
n = z_a2**2 * p_ * (1 - p_) / e**2 ; n
```

Out[169]:

```
60.4599081577026
```

In [170]:

```
n = z_a2**2 / (4*e**2) ; n
```

Out[170]:

```
64.15525059178968
```

# 3b

In [172]:

```
p_ - z_a2 * ( p_*q_/n )**0.5
```

Out[172]:

```
0.32921001330909827
```

In [173]:

```
p_ + z_a2 * ( p_*q_/n )**0.5
```

Out[173]:

```
0.43078998669090174
```