Python Practice 3

Probability and Statistics Programming (Sejong University)¶

Date: 2019.05.13(By: S.M. Riazul Islam)

Confidence Interval

Finding z/t Critical Values

In [21]:
```
# z-crtical values
# 100*(1-alpha)% Confidence Level

import scipy.stats as sp
alpha=0.05
sp.norm.ppf(1-alpha, loc=0, scale=1) # One-sided; ppf=percent point function
```

Out[21]: 1.6448536269514722

In [10]:
```
# z-crtical values
# 100*(1-alpha)% Confidence Level

import scipy.stats as sp
alpha=0.05
sp.norm.ppf(1-alpha/2, loc=0, scale=1) # Two-sided
```

Out[10]: 1.959963984540054

In [19]:
```
# t-crtical values
# 100*(1-alpha)% Confidence Level

import scipy.stats as sp
alpha=0.05
sp.t.ppf(1-alpha, df=4) # One-sided; df=degrees of fredom
```

Out[19]: 2.13184678133629

In [20]:
```
# t-crtical values
# 100*(1-alpha)% Confidence Level

import scipy.stats as sp
alpha=0.05
sp.t.ppf(1-alpha/2, df=4) # Two-sided; df=degrees of fredom
```

Out[20]: 2.7764451051977987

Finding Confidence Interval from a Dataset

In [5]:
```
import scipy.stats as sp
import numpy as np
import random
data = sp.norm.rvs(loc=5, scale=2, size=1000) # Dummy data creaion
```

In [3]:
```
# select 40 data randomly
random.shuffle(data)     # shuffling data for randomization
sample=data[0:39]        # Select first 40 data
sample
```

Out[3]:
```
array([3.36279428, 5.82074126, 4.35515673, 0.55956141, 5.40747593,
       4.42548187, 7.2679764 , 3.93362727, 5.70989225, 7.17753465,
       0.76002705, 5.38796465, 7.32896825, 2.49800773, 0.82878237,
       6.5796981 , 3.87537448, 4.37941019, 5.7685642 , 4.74356547,
       5.60335784, 2.76737548, 3.65747905, 4.81596705, 5.95379591,
       9.28694471, 4.3384802 , 6.84263376, 7.56020637, 5.08440045,
       8.84636786, 4.643632  , 5.89429687, 4.188482  , 7.69416368,
       2.92477777, 6.34425461, 3.68424952, 7.2857122 ])
```

In [12]:
```
# use z-score because Sample size is large (Also population distribution is norma
xbar=np.mean(sample)    # Sample mean
sigmabar=2/np.sqrt(40)  # Sample standard deviation
lvalue=xbar-1.96*sigmabar      # Lower bound of 95% Confidence Interval
rvalue=xbar+1.96*sigmabar
print("95% CI: ",'(',lvalue,rvalue,')')
```

```
95% CI:  ( 4.446531575499767 5.686144418285771 )
```

Use Python Function for Calculating CI

In [11]:
```
sp.norm.interval(0.95,loc=xbar,scale=sigmabar)  #xbar is the sample mean calculat
```

Out[11]:
```
(4.4465429645882075, 5.686133029197331)
```