

In [1]:

```
import numpy as np
import pandas as pd
import scipy.stats as st
import matplotlib.pyplot as plt
import statsmodels.api as sm
import math
```

## Giugno 2019

### Esercizio 0

$Y \sim UnifDisc(s)$

$X \sim Bern(p)$  con  $p=0.8$

#### 0.1.1-0.1.2-0.2.1-0.2-2

grafici dispersione e ripartizione

#### 0.2.3

$$E(X) = p$$
$$Var(X) = p(1 - p)$$

## Esercizio 1

$$\overline{X_n} = \sum \frac{X_i}{n}$$

### 1.1

$$E(\overline{X}) = p$$

### 1.2

$$Var(\overline{X}) = \frac{1}{n} Var(X)$$

### 1.3

$T_n$  stimatore per il valore atteso

$$T_n = \overline{X_n}$$

### 1.4

$$E(T_n) = E\left(\sum \frac{X_i}{n}\right) = \frac{1}{n} \sum E(X_i) = \frac{1}{n} n E(X) = E(X) = p \text{ non è distorto}$$

### 1.5

$U_n$  stimatore varianza non distorto

$$E(Var(X)) = \sum (1-p)p = np(1-p)$$

$$E(U_n) = E(nT_n(1 - T_n))$$

??????????

## Esercizio 2

$$0 < \delta < 1, \epsilon > 0$$

### 2.1

$$P(|\overline{X}_n - p| \leq \epsilon) \geq 1 - \delta \approx P(|X^*| \leq \frac{\epsilon}{\sqrt{p(1-p)}} \sqrt{n})$$

$$P(|\frac{X_n - p}{\frac{\sigma}{\sqrt{n}}}| \leq \frac{\epsilon \sqrt{n}}{\sqrt{p(1-p)}} \geq 1 - \delta$$

### 2.2

$$P(|\overline{X} - p| \leq \epsilon) \geq 1 - \delta$$

$$2\Phi(2\epsilon\sqrt{n}) - 1 \geq 1 - \delta$$

$$\Phi(2\epsilon\sqrt{n}) \geq 1 - \frac{\delta}{2}$$

### 2.3

$$\delta = 0.05, \epsilon = 0.01$$

$$\Phi(2\epsilon\sqrt{n}) \geq 1 - \frac{\delta}{2} = \Phi(0.02\sqrt{n}) \geq 1 - 0.025$$

$$\Phi(0.02\sqrt{n}) \geq 0.975$$

$$\sqrt{n} \geq \frac{\Phi^{-1}(0.975)}{0.02}$$

In [10]:

```
X = st.norm()
n = (X.ppf(0.975)/0.02)**2
n
```

Out[10]:

9603.647051735312

In [ ]:

```
imp = pd.read_csv("impiantitermici.csv", sep=";", decimal=".", parse_dates=True)
imp[:5]
```

## Esercizio 3

### 3.1

In [ ]:

```
len(imp)
```

## 3.2

In [ ]:

```
print("scalare")
```

## 3.3

In [ ]:

```
len(imp[imp['GENERATORE_DATA_INST'] < '01/01/1940'])
```

## 3.4

In [ ]:

```
mask1 = imp['POTENZA_IMPIANTO_RISC'] > 15.0  
mask2 = imp['POTENZA_IMPIANTO_RISC'] < 35.0  
mask3 = imp['GENERATORE_COMBUSTIBILE'] == 'GAS NATURALE'  
selezione = imp[mask1 & mask2 & mask3]
```

## 3.5

In questo caso l'eterogeneità è massima perchè tutti i valori sono uguali, in quanto abbiamo posto come condizione di considerare come combustibile il gas naturale

## 3.6

In [ ]:

```
selezione['EDIFICIO_CATEGORIA'].unique()
```

## 3.7

In [ ]:

```
selezione['EDIFICIO_CATEGORIA'].mode()
```

## 3.8

In [ ]:

```
len(selezione[selezione['EDIFICIO_CATEGORIA'] == 'E1'])
```

### 3.9

In [ ]:

```
#9  
len(selezione[selezione['EDIFICIO_CATEGORIA'] == 'E1'])/len(selezione) * 100
```

In [ ]:

```
### Esercizio 4 non si riesce a fare il confronto fra date
```