

/opt/pyenv-3.7.5/bin/jupyter notebook

```
In [1]: import pandas as pd
import numpy as np
```

Δημιουργία τυχαίου δείγματος 1024 στοιχείων από την  $\mathcal{N}(2, 5)$

```
In [2]: dataset = pd.DataFrame(5*np.random.randn(1024)+2, columns=['x'])
```

```
In [3]: dataset.head()
```

Out[3]:

	x
0	5.292863
1	-3.883816
2	-4.931237
3	2.112910
4	4.157163

Υπολογισμός του συντελεστή συμμετρίας Fisher-Pearson

$$G_1 = \frac{N^2}{(N-1)(N-2)} \frac{\frac{1}{N} \sum_{n=1}^N (x_n - \bar{X})^3}{s^3}$$

```
In [4]: dataset.skew()
```

Out[4]: x 0.022701  
dtype: float64

Με τη len() λαμβάνουμε πόσα στοιχεία έχει το σύνολο δεδομένων

```
In [5]: N = len(dataset)
```

Υπολογισμός του  $\sum_{n=1}^N (x_n - \bar{X})^3$

```
In [6]: dataset - dataset.mean()
```

```
Out[6]:
```

	x
0	3.483789
1	-5.692890
2	-6.740311
3	0.303837
4	2.348089
...	...
1019	0.376268
1020	0.589948
1021	-5.233179
1022	4.843270
1023	-0.844030

1024 rows × 1 columns

```
In [7]: sum3 = ((dataset - dataset.mean())**3).sum()
```

```
In [8]: s = dataset.std()
```

```
In [9]: G1 = N**2 / ((N-1)*(N-2)) * (1/N) * sum3 / s**3
```

```
In [10]: G1
```

```
Out[10]: x    0.022701
dtype: float64
```

### Υπολογισμός του μέτρου κύρτωσης

$$\text{kurt} = \frac{\frac{1}{N} \sum_{n=1}^N (x_n - \bar{X})^4}{s^4} - 3$$

```
In [11]: dataset.kurt()
```

```
Out[11]: x    0.158832
dtype: float64
```

### Άσκηση : Υπολογισμός του $\sum_{n=1}^N (x_n - \bar{X})^4$

```
In [12]: sum4 = ((dataset-dataset.mean())**4).sum()
```

```
In [13]: sum4
```

```
Out[13]: x    1.795067e+06  
dtype: float64
```

```
In [14]: kurt = (1/N * sum4)/s**4 - 3
```

```
In [15]: kurt
```

```
Out[15]: x    0.14605  
dtype: float64
```

**Άσκηση : Υπολογίστε τη μέση τιμή, τη διάμεσο, τα τεταρτημόρια για το dataset**

```
In [16]: dataset.describe()
```

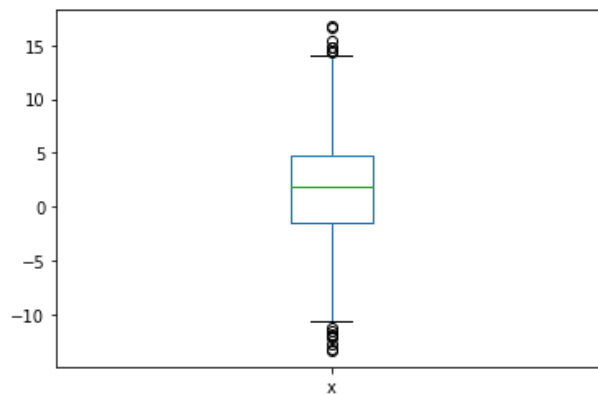
```
Out[16]:
```

	x
count	1024.000000
mean	1.809074
std	4.858517
min	-13.379307
25%	-1.433172
50%	1.895105
75%	4.799271
max	16.765703

**Άσκηση : Δημιουργήστε το Box-and-Whisker plot για το dataset**

```
In [17]: dataset.plot.box()
```

```
Out[17]: <matplotlib.axes._subplots.AxesSubplot at 0x7fcfd56f5860>
```



## scipy.stats

```
In [18]: import scipy.stats as st
```

## z-scores

$$z = \frac{x - \bar{X}}{s}$$

```
In [19]: x1 = 17.0
```

```
In [20]: z1 = (x1 - dataset.mean())/dataset.std()
```

```
In [21]: z1
```

```
Out[21]: x      3.126659  
dtype: float64
```

## Υπολογισμός του $P(Z \leq z_1)$ με την `scipy.stats`

```
In [22]: P1 = st.norm.cdf(z1)
```

```
In [23]: P1
```

```
Out[23]: array([0.99911598])
```

## Άσκηση : Επιλύστε με χρήση των `pandas` και `scipy.stats`

Σε ένα δημοψήφισμα με απάντηση ΝΑΙ / ΟΧΙ, το ΟΧΙ συγκέντρωσε το 60% των συνολικών ψήφων. Ποιά η πιθανότητα σε ένα τυχαίο δείγμα 500 ψηφοφόρων το ΝΑΙ να συγκεντρώνει τουλάχιστον το 50%;

```
In [24]: p = 0.6
```

```
In [25]: sigma_p_hat = (0.6*(1-0.6)/500)**(1/2)
```

```
In [26]: z = (0.5-p)/sigma_p_hat
```

```
In [27]: z
```

```
Out[27]: -4.564354645876383
```

```
In [28]: st.norm.cdf(z)
```

```
Out[28]: 2.505165978195235e-06
```