

T.P. V - Matrices & Statistiques

Code Capter : 2c2d-893913

I - Calculs sur des vecteurs

Solution de l'exercice 1.

```
import numpy as np

notes = np.array([18, 12, 15, 10, 14])

moyenne = np.mean(notes)
print("moyenne", moyenne)

ecarttype = np.std(notes)
print("écart-type", ecarttype)
```

□

Solution de l'exercice 2.

1.

```
import numpy as np

def u(n):
    u = 1
    for i in range(1, n+1):
        u = u * 3/i
    return u
```

2. À l'aide de la fonction précédente, afficher les valeurs de u_5 et u_{20} .

```
print("u5", u(5))
print("u20", u(20))
```

3.

```
U = np.ones((21, 1))
for i in range(1, 21):
    U[i] = u(i)

S = np.cumsum(U)
```

4.

```
import matplotlib.pyplot as plt
plt.figure()
X = np.arange(0, 21, 1)
plt.plot(X, S, 'o')
plt.show()
```

5. Il semble que la suite (s_n) converge vers 20. □

II - Statistiques bivariées

Solution de l'exercice 4.

1.

```
n, p = np.shape(M)
print("Nombre d'arbres : ", n)
```

2.

```
plt.figure()
plt.scatter(M[:,0], M[:,1])
plt.show()
```

3. À l'aide d'un unique appel de fonction, calculer les circonférence, hauteur et volume moyens.

```
print("Caractéristiques moyennes", np.mean(M, 0))
```

4.

```
plt.figure()
plt.scatter(M[:,0], M[:,2])
plt.plot(np.mean(M[:,0]), np.mean(M[:,2]), "xr")
plt.show()
```

5.

```
print("Coefficient de corrélation",\
      np.corrcoef(M[:,0], M[:,2])[0, 1])
```

6.

```
(a, b) = np.polyfit(M[:,0], M[:,2], 1)
T = np.arange(np.min(M[:,0]) - 1, np.max(M[:,0]) + 1)

def droite(x):
    return (a * x + b)

plt.figure()
plt.scatter(M[:,0], M[:,2])
plt.plot(np.mean(M[:,0]), np.mean(M[:,2]), "xr")
plt.plot(T, droite(T), 'g')
plt.show()
```

□