U.E. 5.2 Contrôle continu 4

le 12/10/2018

Nom, prénom : /10

Durée : 15 minutes.

Dans les questions qui concernent le langage Python, le respect de la syntaxe (et des espaces) est pris en compte dans la notation. Les questions sont indépendantes.

Accès à python, au cours autorisé et à internet autorisé.

Question 1 : Il vous est demandé **de renseigner le code** nécessaire pour répondre à chaque question. Un résultat numérique est attendu en plus du code à la question 3.

Le module numpy a été importé avec la commande : `import numpy as np`

1. Générer le vecteur Z avec 5000 points suivant la distribution gaussienne N(0,1) [code]

```
Z = np.random.normal(size=5000)
```

2. Compter le nombre d'éléments de Z supérieurs à -1 et inférieurs à 1 ? [code]

```
sum(abs(Z) < 1)
```

3. Quelle est la proportion d'éléments de Z supérieurs à -1 et inférieurs à 1 ? [code + valeurs]

```
sum(abs(Z) < 1)/len(Z)
```

Environ 68%

4. Expliquer les valeurs obtenues

Correspond à ce qu'on attend comme proportions pour une gaussienne à 1 sigma (~68,2%), ici sigma valant 1.

Question 2 : Le résultat de la résolution analytique pour les coefficients de la régression linéaire vue en cours est la suivante :

$$\hat{\theta}_1 = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sum(x_i - \bar{x})^2} = \frac{cov(x,y)}{var(x)}$$

$$\hat{\theta}_0 = \bar{y} - \hat{\theta}_1 \bar{x}$$

$\hat{\theta}_0$ et $\hat{\theta}_1$ sont les coefficients que l'on souhaite estimer, tels que

$$\hat{y} = \hat{\theta}_0 + \hat{\theta}_1 x$$

$\bar{x}$ et $\bar{y}$ sont les moyennes respectives des vecteurs `x` et `y`.

La covariance et la variance sont définies par :

$$cov(x,y) = \frac{1}{N} * \sum (x_i - \bar{x})(y_i - \bar{y})$$

$$var(x) = \frac{1}{N} * \sum (x_i - \bar{x})^2$$

avec $N = len(x) - 1$

Nous allons vous demander de créer plusieurs fonctions pour calculer au final les paramètres $\hat{\theta}_0$ et $\hat{\theta}_1$ de façon analytique. **L'ordre des questions n'est pas aléatoire !**

- **Dans ce qui suit, x et y sont des listes de nombres de même taille.**
- **Vous n'avez pas accès à numpy.**

1/ Créer la fonction `moyenne(x)` retournant la moyenne de la liste `x`

```
def moyenne(x):
     return sum(x)/len(x)
```

2/ Créer la fonction `produit_scalaire(x,y)` permettant de retourner le résultat du produit scalaire de `x` et `y`

```
def produit_scalaire(x,y):
     return sum([i*j for i,j in zip(x,y)])
```

3/ Créer la fonction `covariance(x,y)` permettant de retourner le résultat `du` calcul de la covariance, définie plus haut, en utilisant la fonction `produit_scalaire()`

```
def variance(x):
     return covariance(x,x)
```

4/ Créer la fonction `variance(x)` permettant de retourner le résultat du calcul de la variance, définie plus haut

```
def covariance(x,y):
     N = len(x)-1
     return 1/N*produit_scalaire(x-moyenne(x), y-moyenne(y))
```

5/ Créer la fonction `linear_parameters(x,y)` permettant de retourner les résultats de l'estimation des paramètres $\hat{\theta}_0$ et $\hat{\theta}_1$, à partir des équations définies plus haut.

```
def linear_parameters(x,y):
     theta_1 = covariance(x,y) / variance(x)
     theta_0 = moyenne(y) - theta_1*moyenne(x)
     return theta_0, theta_1
```