ENSAE TD noté rattrapage, mardi 21 février 2017

Le programme devra être imprimé et rendu au chargé de TD. Toutes les questions valent 2 points. Vous êtes libres d'utiliser numpy ou non à toutes les questions.

Quelques fonctions dont vous pourriez avoir besoin :

```
@                    ou numpy.dot
numpy.array
numpy.abs
numpy.mean
numpy.sum
numpy.diag
numpy.linalg.inv
numpy.maximum
numpy.ones
numpy.reciprocal
random.randint
sort
X.reshape
X.shape
X.T
```

1

1) A l'aide du module random, générer un ensemble d'entiers aléatoires compris entre 0 et 100 excepté le premier égal à 1000.

```
def ensemble_aleatoire(n):
    ....
    return
```

2) La médiane d'un ensemble de points $\{X_1, ..., X_n\}$ est une valeur X_M telle que :

$$\sum_i \mathbb{1}_{\{X_i<X_m\}} = \sum_i \mathbb{1}_{\{X_i>X_m\}}$$

Autrement dit, il y a autant de valeurs inférieures que supérieures à X_M. On obtient cette valeur en triant les éléments par ordre croissant et en prenant celui du milieu.

Ecrire une fonction qui calcule la médiane.

```
def mediane(ensemble):
    ....
    return
```

3) Lorsque le nombre de points est pair, la médiane peut être n'importe quelle valeur dans un intervalle. Modifier votre fonction de façon à ce que la fonction précédente retourne le milieu de la fonction.

4) Pour un ensemble de points $E = \{X_1, ..., X_n\}$, on considère la fonction suivante :

$$f(x) = \sum_{i=1}^{n} |x - X_i|$$

.

On suppose que la médiane X_M de l'ensemble E n'appartient pas à E : $X_M \notin E$. Que vaut $f'(X_M)$? On acceptera le fait que la médiane est le seul point dans ce cas.

5) On suppose qu'on dispose d'un ensemble d'observations (X_i, Y_i) avec $X_i, Y_i \in \mathbb{R}$. La régression linéaire consiste en une relation linéaire $Y_i = aX_i + b + \epsilon_i$ qui minimise la variance du bruit. On pose :

$$E(a, b) = \sum_i (Y_i - (aX_i + b))^2$$

On cherche a, b tels que :

$$a^*, b^* = \arg\min E(a, b) = \arg\min \sum_i (Y_i - (aX_i + b))^2$$

La fonction est dérivable et on trouve :

$$\frac{\partial E(a,b)}{\partial a} = -2 \sum_i X_i (Y_i - (aX_i + b)) \text{ et } \frac{\partial E(a,b)}{\partial b} = -2 \sum_i (Y_i - (aX_i + b))$$

Il suffit alors d'annuler les dérivées. On résoud un système d'équations linéaires. On note :

$$\mathbb{E}(X) = \frac{1}{n} \sum_{i=1}^{n} X_i \text{ et } \mathbb{E}(Y) = \frac{1}{n} \sum_{i=1}^{n} Y_i$$
$$\mathbb{E}(X^2) = \frac{1}{n} \sum_{i=1}^{n} X_i^2 \text{ et } \mathbb{E}(XY) = \frac{1}{n} \sum_{i=1}^{n} X_i Y_i$$

Finalement :

$$a^* = \frac{\mathbb{E}(XY) - \mathbb{E}(X)\mathbb{E}(Y)}{\mathbb{E}(X^2) - (\mathbb{E}(X))^2} \text{ et } b^* = \mathbb{E}(Y) - a^* \mathbb{E}(X)$$

Lorsqu'on a plusieurs dimensions pour X, on écrit le problème d'optimisation, on cherche les coefficients β^* qui minimisent :

$$E(\beta) = \sum_{i=1}^{n} (y_i - X_i \beta)^2 = \|Y - X\beta\|^2$$

La solution est :

$$\beta^* = (X'X)^{-1} X'Y$$

Ecrire une fonction qui calcule ce vecteur optimal :

```
def regression_lineaire(X, Y):
    ....
    return
```

6) Ecrire une fonction qui transforme un vecteur en une matrice diagonale :

```
def matrice_diagonale(W):
    ....
    return
```

7) On considère maintenant que chaque observation est pondérée par un poids w_i. On veut maintenant trouver le vecteur β qui minimise :

$$E(\beta) = \sum_{i=1}^{n} w_i \left(y_i - X_i\beta\right)^2 = \left\| W^{\frac{1}{2}}(Y - X\beta) \right\|^2$$

Où $W = diag(w_1, ..., w_n)$ est la matrice diagonale. La solution est :

$$\beta_* = (X'WX)^{-1}X'WY$$

.

Ecrire une fonction qui calcule la solution de la régression pondérée.

```
def regression_lineaire_ponderee(X, Y, W):
    ....
    return
```

8) Ecrire une fonction qui calcule les quantités suivantes :

$$z_i = \frac{1}{\max\left(\delta, |y_i - X_i\beta|\right)}$$

```
def calcule_z(X, beta, Y, W, delta=0.0001):
    ....
    return
```

9) On souhaite coder l'algorithme suivant :

1. $w_i^{(1)} = 1$
2. $\beta_{(t)} = (X'W^{(t)}X)^{-1}X'W^{(t)}Y$
3. $w_i^{(t+1)} = \frac{1}{\max\left(\delta, \left|y_i - X_i\beta^{(t)}\right|\right)}$
4. $t = t + 1$
5. Retour à l'étape 2.

```
def algorithm(X, Y, delta=0.0001):
    ....
    return
```

10) On pose Y le vecteur aléatoire de la question 1. X est un vecteur de même dimension constant et égale à 1. Calculer les quatre valeurs suivantes :

1. $algorithm(X, Y)$
2. $regression\_lineaire(X, Y)$
3. $mediane(Y)$
4. $moyenne(Y)$

Que constatez-vous ?