

TP Modélisation à événements discrets

GnuPlot

Exercice 1

L'objectif de cet exercice est de construire un programme qui permettra de générer aléatoirement des fichiers de données comportant une ou plusieurs colonnes.

Une colonne sera modélisée par la classe `Data` comportant trois attributs :

- `min` qui sera de type réel et représentera la valeur la plus petite que l'on pourra retrouver sur cette colonne.
- `max` qui sera de type réel et représentera la valeur la plus grande que l'on pourra retrouver sur cette colonne.
- `rand` qui indiquera si la valeur sera tirée aléatoirement ou non. Si la valeur n'est pas tirée aléatoirement alors on se basera sur le nombre de lignes devant figurer dans le fichier de données pour fixer l'intervalle entre deux valeurs successives.

La classe `Data` comportera aussi les méthodes d'accès classiques : `getXXX` et `setXXX`.

La classe principale `GenerateData` comportera les attributs suivants :

- `numberOfLines` qui sera de type entier et indiquera le nombre de lignes devant figurer dans le fichier de données généré. Par défaut cette valeur sera initialisée à 10 donc chaque colonne comportera 10 valeurs.
- `columns` qui sera de type `ArrayList<Data>` servira à stocker les différentes données décrivant les colonnes.

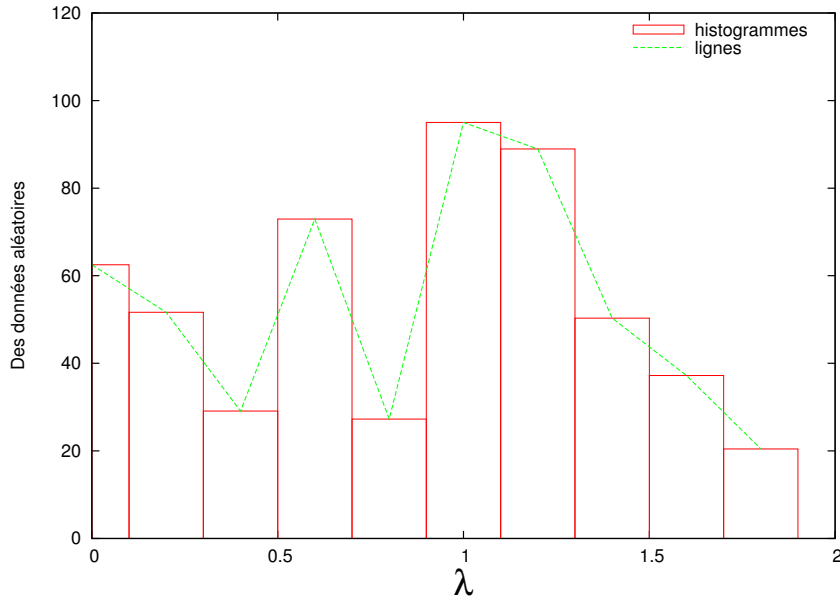
Cette classe comportera aussi les méthodes suivantes :

- `GenerateData()` : un constructeur par défaut permettant d'initialiser le nombre de lignes (`numberOfLines`) à 10 et la variable `columns`.
- `setNumberOfLines()` et `getNumberOfLines()` : les méthodes d'accès à l'attribut `numberOfLines`.
- `addDataColumn (double min, double max, boolean rand)` : Cette méthode crée une nouvelle donnée de type `Data` représentant donc une colonne et l'ajoute dans la liste prévue à cet effet.
- `writeDataFile (String filename)` : Cette méthode permettra de générer un fichier de données. Il faudra vérifier qu'il existe au moins une colonne de spécifiée. Afin de vérifier l'endroit où sera écrit le fichier, vous pouvez utiliser la méthode : `System.getProperty("user.dir")`.

Une méthode principale devra permettre de tester le programme et de générer une courbe.

Exercice 2

Il s'agit de tracer deux courbes à partir de données générées par le biais du programme de l'exercice 1. Les valeurs utilisées pour l'axe des abscisses (première colonne du fichier de données) seront comprises entre 0 et 2.0. Les valeurs pour l'axe des ordonnées (seconde colonne du fichier de données) seront générées aléatoirement entre 20 et 100. Vous utiliserez alors les données pour tracer une courbe et des histogrammes.



Exercice 3

Il s'agit de créer une seconde version de la classe `GenerateData` afin que les résultats, fournis par le programme, ne soient plus basés sur une seule série de données générées aléatoirement mais sur un ensemble de séries (environ 1000) pour lesquelles le générateur sera réinitialisé au début du tirage de chaque série.

L'objectif de ce type de manipulation est d'augmenter la fiabilité statistique des tirages et par conséquent des résultats fournis.

