

Charger un dataset

La classe `BufferedReader` permet de lire le contenu d'un fichier :

```
BufferedReader <IDF_READER> = new BufferedReader(new FileReader(<PATH_TO_FILE>));  
BufferedReader reader = new BufferedReader(new FileReader("C:/seeds.txt"));
```

La méthode `readLine()` permet de lire une ligne de texte. Elle retourne `null` SI la fin du flux est atteinte, SINON une chaîne de caractères :

```
String <IDF_STR> = <IDF_READER>.readLine();
```

Déclarer une liste de `Double` :

```
ArrayList<Double> <IDF_LIST> = new ArrayList();
```

La méthode `add()` permet d'ajouter un élément à une liste de `Double` :

```
<IDF_LIST>.add(Double element);
```

La méthode `split()` permet de scinder une chaîne de caractères :

```
String[] <IDF_STR_ARRAY> = <IDF_STR>.split(<REGULAR_EXPRESSION>);  
String[] str_splited = str.split("\\t+");
```

La méthode `parseDouble()` permet de convertir une chaîne de caractère en `Double` :

```
Double <IDF_DOUBLE> = Double.parseDouble(<IDF_STR>);
```

Tracer un diagramme de dispersion

La classe `XYSeriesCollection` permet de créer une collection (dataset) de données :

```
XYSeriesCollection <IDF_DATASET> = new XYSeriesCollection();
```

La classe `XYSeries` permet de créer un ou plusieurs éléments de données (x, y) :

```
XYSeries <IDF_XYSeries> = new XYSeries("");
```

La méthode `add()` permet d'ajouter un élément de données (x, y) à `XYSeries` :

```
<IDF_XYSeries>.add(Double X, Double Y);
```

La méthode `addSeries()` permet d'ajouter la série `XYSeries` à la collection (dataset) :

```
<IDF_DATASET>.addSeries(<IDF_XYSeries>);
```

La méthode `ChartFactory.createScatterPlot()` de la classe `ChartFactory` permet de créer un diagramme de dispersion :

```
JFreeChart <IDF_SCATTER_PLOT> = ChartFactory.createScatterPlot("Title", "1st attribute name", "2nd attribute name",  
<IDF_DATASET>, PlotOrientation.<Orientation>, <boolean>, <boolean>, <boolean>);
```

La classe `ChartPanel` permet de créer un panneau pour ajouter un diagramme de dispersion `JFreeChart`:

```
ChartPanel <IDF_PANEL> = new ChartPanel(<IDF_SCATTER_PLOT>);
```

La classe `JFrame` permet de créer une fenêtre pour ajouter le panneau `ChartPanel` :

```
JFrame <IDF_FRAME> = new JFrame();
```

```
<IDF_FRAME>.add(<IDF_PANEL>);
```

```
<IDF_FRAME>.setVisible(true);
```

La classe `XYPlot` permet de modifier et personnaliser certaines propriétés du graphe :

```
XYPlot <IDF_PLOT> = (XYPlot) <IDF_SCATTER_PLOT>.getPlot();
```

La classe `ValueAxis` et la méthode `getDomainAxis()` permet de modifier et personnaliser certaines propriétés de l'axe x :

```
ValueAxis <IDF_AXIS_X> = <IDF_PLOT>.getDomainAxis();
```

La classe `ValueAxis` et la méthode `getRangeAxis()` permet de modifier et personnaliser certaines propriétés de l'axe y :

```
ValueAxis <IDF_AXIS_Y> = <IDF_PLOT>.getRangeAxis();
```

La classe `XYLineAndShapeRenderer` permet de modifier et personnaliser certaines propriétés des points (x, y) :

```
XYLineAndShapeRenderer <IDF_RENDERER> = new XYLineAndShapeRenderer();
```

```
<IDF_PLOT>.setRenderer(<IDF_RENDERER>);
```

Tracer un histogramme

La classe `HistogramDataset` permet de créer une collection (dataset) de données :

```
HistogramDataset <IDF_DATASET> = new HistogramDataset ();
```

La méthode `setType()` permet d'initialiser le type de l'histogramme :

```
<IDF_DATASET>.setType(HistogramType.<TYPE>);
```

La méthode `addSeries()` permet d'ajouter les données au dataset :

```
<IDF_DATASET>.addSeries("", double[] <VALUES>, <NUMBER_BINS>);
```

La méthode `ChartFactory.createHistogram()` de la classe `ChartFactory` permet de créer un diagramme de dispersion :

```
JFreeChart <IDF_HISTOGRAM> = ChartFactory.createHistogram("Title", "attribute name", "attribute frequency",  
<IDF_DATASET>, PlotOrientation.<Orientation>, <boolean>, <boolean>, <boolean>);
```

La classe `ChartPanel` permet de créer un panneau pour ajouter un histogramme `JFreeChart`:

```
ChartPanel <IDF_PANEL> = new ChartPanel(<IDF_HISTOGRAM>);
```

La classe `JFrame` permet de créer une fenêtre pour ajouter le panneau `ChartPanel`:

```
JFrame <IDF_FRAME> = new JFrame();  
<IDF_FRAME>.add(<IDF_PANEL>);  
<IDF_FRAME>.setVisible(true);
```

La classe `XYPlot` permet de modifier et personnaliser certaines propriétés du graphe :

```
XYPlot <IDF_PLOT> = (XYPlot) <IDF_HISTOGRAM>.getPlot();
```

La classe `ValueAxis` et la méthode `getDomainAxis()` permet de modifier et personnaliser certaines propriétés de l'axe x :

```
ValueAxis <IDF_AXIS_X> = <IDF_PLOT>.getDomainAxis();
```

La classe `ValueAxis` et la méthode `getRangeAxis()` permet de modifier et personnaliser certaines propriétés de l'axe y :

```
ValueAxis <IDF_AXIS_Y> = <IDF_PLOT>.getRangeAxis();
```

La classe `XYBarRenderer` permet de modifier et personnaliser certaines propriétés des points (x, y) :

```
XYBarRenderer <IDF_RENDERER> = new XYBarRenderer();  
<IDF_PLOT>.setRenderer(<IDF_RENDERER>);
```

Tracer une boîte à moustache

La classe `DefaultBoxAndWhiskerXYDataset` permet de créer une collection (dataset) de données :

```
DefaultBoxAndWhiskerXYDataset <IDF_DATASET> = new DefaultBoxAndWhiskerXYDataset("");
```

La méthode `BoxAndWhiskerCalculator.calculateBoxAndWhiskerStatistics()` permet de calculer les statistiques requises (Médiane, Q1, Q3, ...) pour un élément de données `BoxAndWhiskerItem` :

```
BoxAndWhiskerItem <IDF_DATA_ITEM> = BoxAndWhiskerCalculator.calculateBoxAndWhiskerStatistics(LIST <VALUES>);
```

La méthode `add()` permet d'ajouter un élément de données au dataset :

```
<IDF_DATASET>.add(<IDF_DATA_ITEM>, "", "");
```

La méthode `ChartFactory.createBoxAndWhiskerChart()` de la classe `ChartFactory` permet de créer une boîte à moustache :

```
JFreeChart <IDF_BOX_PLOT> = ChartFactory.createBoxAndWhiskerChart("Title", "attribute name", "attribute values", <IDF_DATASET>, <boolean>);
```

La classe `ChartPanel` permet de créer un panneau pour ajouter un histogramme `JFreeChart`:

```
ChartPanel <IDF_PANEL> = new ChartPanel (<IDF_BOX_PLOT>);
```

La classe `JFrame` permet de créer une fenêtre pour ajouter le panneau `ChartPanel` :

```
JFrame <IDF_FRAME> = new JFrame();  
<IDF_FRAME>.add(<IDF_PANEL>);  
<IDF_FRAME>.setVisible(true);
```

La classe `XYPlot` permet de modifier et personnaliser certaines propriétés du graphe :

```
XYPlot <IDF_PLOT> = (XYPlot) <IDF_BOX_PLOT>.getPlot();
```

La classe `ValueAxis` et la méthode `getDomainAxis()` permet de modifier et personnaliser certaines propriétés de l'axe x :

```
ValueAxis <IDF_AXIS_X> = <IDF_PLOT>.getDomainAxis();
```

La classe `ValueAxis` et la méthode `getRangeAxis()` permet de modifier et personnaliser certaines propriétés de l'axe y :

```
ValueAxis <IDF_AXIS_Y> = <IDF_PLOT>.getRangeAxis();
```

La classe `XYBarRenderer` permet de modifier et personnaliser certaines propriétés des points (x, y) :

```
XYBoxAndWhiskerRenderer <IDF_RENDERER> = new XYBoxAndWhiskerRenderer();  
<IDF_PLOT>.setRenderer(<IDF_RENDERER>);
```

Exercice :

Développer un programme en Java qui permet de :

- . Charger le dataset "seeds.txt",
- . Tracer un diagramme de dispersion des attributs N°6 & N°7,
- . Tracer un histogramme pour l'attribut N°6.