D[i] | Département Informatique

UNIVERSITÉ TOULOUSE III PAUL SABATIER — Université de Toulouse

Faculté Sciences et Ingénierie

Licence Informatique

Connexion à une base de données & création de graphes avec JFreeChart

Florent CABRIC

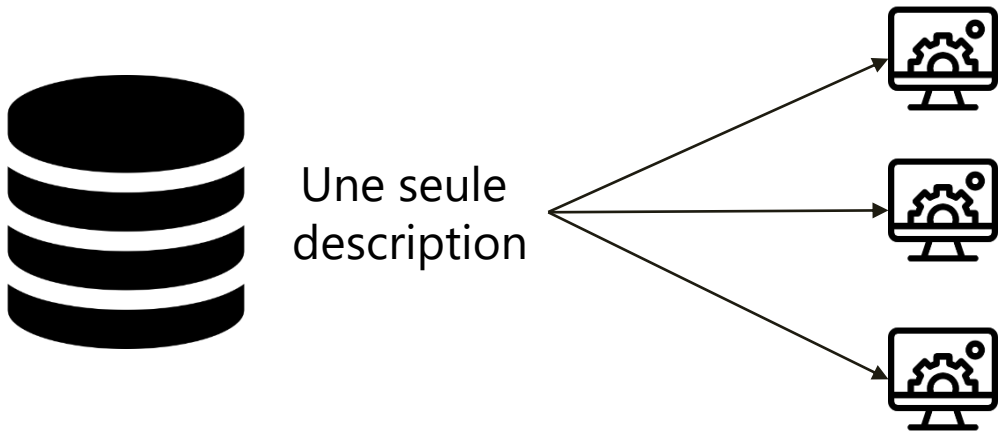
Florent.cabric@irit.fr

Rappel : comment créer une base de données héberger localement

Système de Gestion de Base de données (SGBD)

- BD : Ensemble structuré de données apparentées qui modélisent un univers réel

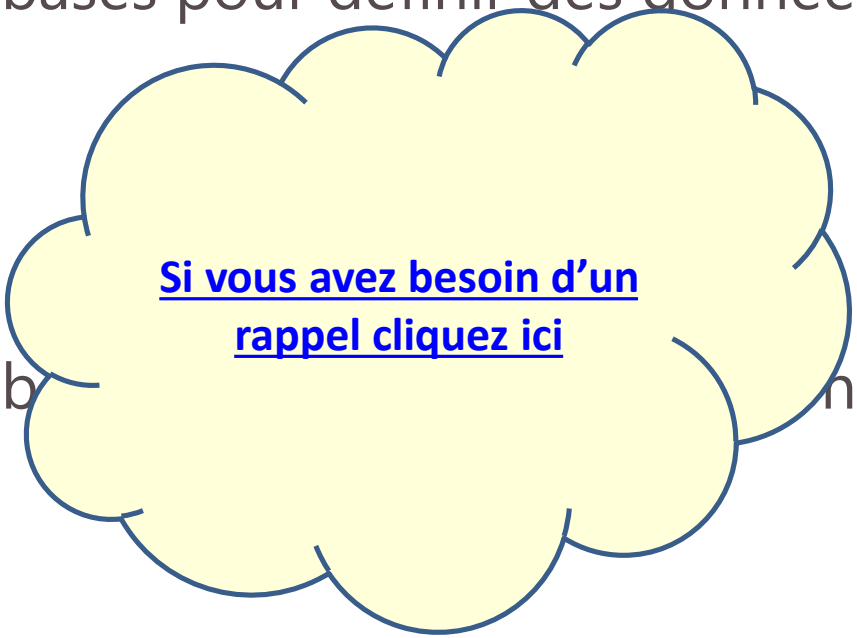
- SGBD : Système permettant de gérer une BD partagée par plusieurs utilisateurs simultanément



Structured Query Language (SQL)

- Langage standard de description et manipulation des base de données relationnelles

- Commandes de bases pour définir des données :
  - CREATE
  - DROP
  - ALTER



- Commandes de b nnées :
  - SELECT
  - INSERT
  - UPDATE
  - DELETE

WampServer

- Plate-forme de développement web sous **Windows**

- Permet de faire fonctionner localement tout un tas de contenus web

- Contient
  - Serveur Apache
  - Serveur MySQL
  - Serveur MariaDB
  - Interpréteur PHP
  - phpMyAdmin

WampServer

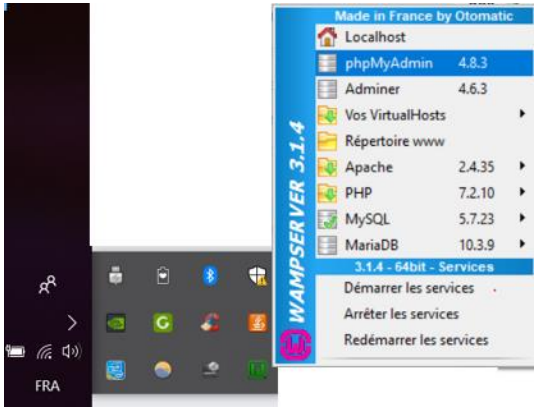

www.wampserver.com
Rechercher
Les plus visités
La Belle Verte, Toulous...
ZK FootballVideos
(3) Twitter
Les matchs en direct d...
LivresNoel
-43€89 sur Xiaomi Red...
Nîmes vs OM du 12/0...
Appart
Bordeaux et Nîmes fo...
UltimeDecathlon - UD...
Tipeee - Taranis News
TLS à BER, 21/12 — 24...
Débuter avec Firefox
ENGLISH РУССКИЙ
WampServer
Apache, PHP, MySQL sous Windows
DÉMARRER
TÉLÉCHARGER
FORMATION
FORUM
HÉBERGEMENT CLOUD
WAMPSERVER,
plate-forme de développement Web sous Windows
WampServer est une plate-forme de développement Web sous Windows pour des applications Web dynamiques à l'aide du serveur Apache2, du langage de scripts PHP et d'une base de données MySQL. Il possède également PHPMyAdmin pour gérer plus facilement vos bases de données.
EXPÉRIMENTER WAMPSERVER
CONTRIBUTION ALTER WAY
Cliquez ici pour télécharger
DÉMARRER AVEC WAMPSERVER
Comme vous allez le voir, WampServer s'installe facilement et son utilisation très intuitive permet de le configurer très rapidement (sans toucher aux fichiers de configuration).
INSTALLATION
Double-cliquez sur le fichier téléchargé et laissez vous guider. Tout est géré par l'installeur de WampServer. Par défaut, WampServer est livré avec les toutes dernières versions de Apache, MySQL et PHP.
Une fois installé, vous pourrez ajouter manuellement des versions supplémentaires d'Apache, PHP ou MySQL (Uniquement compilées VC9, VC10 ou VC11). Les explications pour le faire vous seront données sur le forum.
Chaque version de Apache, MySQL et PHP dispose de sa propre
FONCTIONNALITÉS
WampServer dispose également d'un « TrayIcon » vous permettant de gérer et configurer simplement vos serveurs, sans toucher aux fichiers de configuration.
Clic gauche sur l'icône de WampServer, vous pouvez notamment :
Gérer les services de Apache et MySQL
passer en mode online/offline (accessible à tous ou limité à localhost)
Installer et changer de version de Apache, MySQL et PHP
Gérer les paramètres de configuration de vos serveurs

phpMyAdmin

- phpMyAdmin est une application Web de gestion pour les systèmes de gestion de base de données MySQL

- Permet de créer facilement une base de données et d'y effectuer toutes les opérations nécessaires

- Attention ceci est un outil, tout peut être réalisé en SQL « pur »



Pour y accéder à partir de l'icone Wamp

D[i] | Département Informatique

UNIVERSITÉ TOULOUSE III PAUL SABATIER — Université de Toulouse

Faculté Sciences et Ingénierie

Gérer une base de données avec Java et JDBC

Java DataBase Connectivity (JDBC)

- JDBC est une API Java permettant d'accéder à une base de données relationnelles.

- JDBC permet de gérer trois activités :
  - Connexion à une source de données
  - L'envoi de requêtes et la mise à jour de la base de données
  - Le traitement des résultats résultant de votre requête

JDBC – Pseudo code

```
public void connectToAndQueryDatabase(String username, String password) {

    Connection con = DriverManager.getConnection(
                        "jdbc:myDriver:myDatabase",
                        username,
                        password);

    Statement stmt = con.createStatement();
    ResultSet rs = stmt.executeQuery("SELECT a, b, c FROM Table1");

    while (rs.next()) {
        int x = rs.getInt("a");
        String s = rs.getString("b");
        float f = rs.getFloat("c");
    }
}
```

1) Connexion à une source de données

JDBC – Pseudo code

```
public void connectToAndQueryDatabase(String username, String password) {

    Connection con = DriverManager.getConnection(
                        "jdbc:myDriver:myDatabase",
                        username,
                        password);

    Statement stmt = con.createStatement();
    ResultSet rs = stmt.executeQuery("SELECT a, b, c FROM Table1");

    while (rs.next()) {
        int x = rs.getInt("a");
        String s = rs.getString("b");
        float f = rs.getFloat("c");
    }
}
```

2) Envoyer des reqûetes

JDBC – Pseudo code

```
public void connectToAndQueryDatabase(String username, String password) {

    Connection con = DriverManager.getConnection(
                        "jdbc:myDriver:myDatabase",
                        username,
                        password);

    Statement stmt = con.createStatement();
    ResultSet rs = stmt.executeQuery("SELECT a, b, c FROM Table1");

    while (rs.next()) {
        int x = rs.getInt("a");
        String s = rs.getString("b");
        float f = rs.getFloat("c");
    }
}
```

3) Traiter le résultat de la rêquete

Utiliser JDBC

- Télécharger la dernière version de mysql-connector
  - https://mvnrepository.com/artifact/mysql/mysql-connector-java/8.0.13

- Installation…

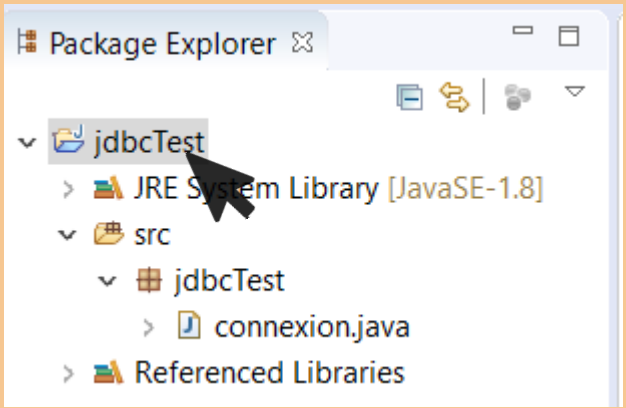
Etape 0 : créer votre projet sur eclipse

Utiliser JDBC

- Télécharger la dernière version de mysql-connector
  - https://mvnrepository.com/artifact/mysql/mysql-connector-java/8.0.13

- Installation…

Etape 1 : Clique droit sur le nom de votre projet. Ici jdbcTest

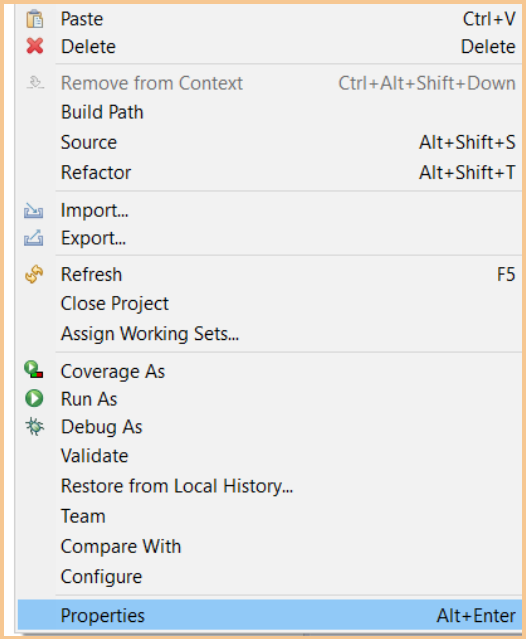


Utiliser JDBC

- Télécharger la dernière version de mysql-connector
  - https://mvnrepository.com/artifact/mysql/mysql-connector-java/8.0.13

- Installation…

Etape 2 : Clique sur l'item Properties

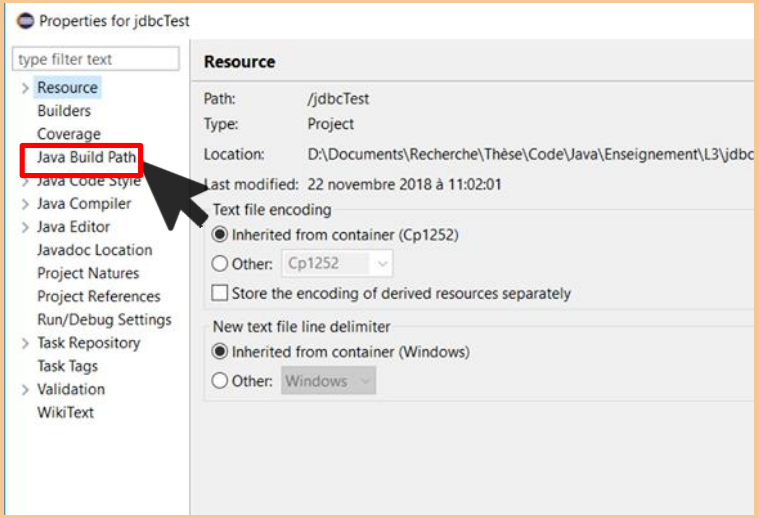


Utiliser JDBC

- Télécharger la dernière version de mysql-connector
  - https://mvnrepository.com/artifact/mysql/mysql-connector-java/8.0.13

- Installation…

Etape 3 : Dans la fenêtre Properties.
Clique sur l'item « Java Build Path »

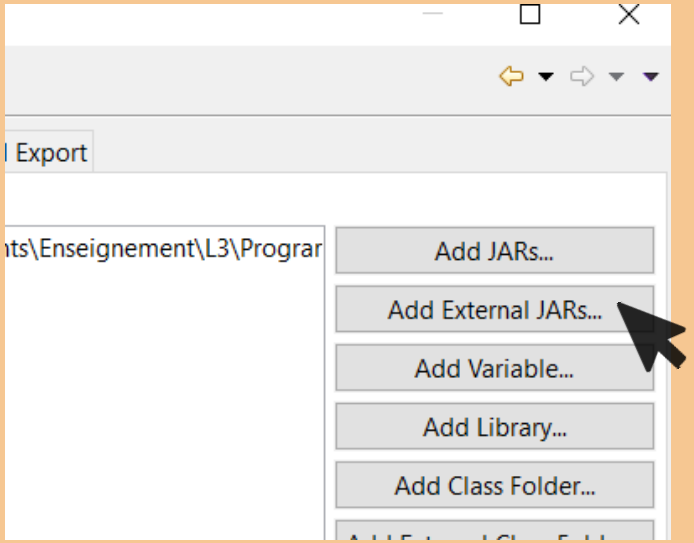


Utiliser JDBC

- Télécharger la dernière version de mysql-connector
  - https://mvnrepository.com/artifact/mysql/mysql-connector-java/8.0.13

- Installation…

Etape 4 : Dans Le menu Java build Path. Clique sur le bouton « add external Jars »



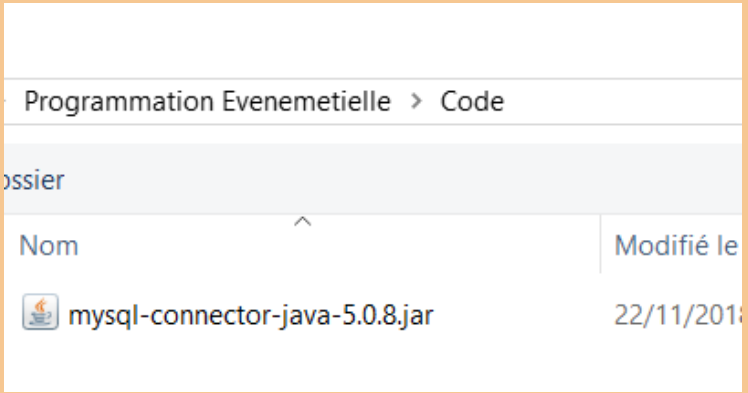
Utiliser JDBC

- Télécharger la dernière version de mysql-connector
  - https://mvnrepository.com/artifact/mysql/mysql-connector-java/8.0.13

- Installation…

Etape 5 : Chercher et cliquer sur .jar téléchargé au début.

Fin…



Etablir une connexion à une base de données

- JDBC peut se connecter en choisissant une des deux classes suivantes :
  - DataSource
  - **DriverManager**

- Appeler le driver jdbc (nécessaire selon votre ordinateur)

```
Class.forName("com.mysql.jdbc.Driver");
```

- Connexion à une base de données

```
Connection DriverManager.getConnection(connecteur, username , password);
```

Etablir une connexion à une base de données

- Adresse du connecteur

```
String connecteur = “jdbc:mysql://[host]:[port]/[votrenomdeBase”]
```

- Les valeurs par défaut
  - host = 127.0.0.1 (localhost)
  - Port 3306

- Login par défaut : « root »
- Mot de passe par défaut : « »

ATTENTION : vous ne devez pas les modifier pour le projet sinon nous ne pourrons pas nous connecter à votre base de données même si vous nous fournissez le mot de passe et le login !!!!

Exemple de code de connexion

```
Connection con;
try {
        con = DriverManager.getConnection(
                "jdbc:mysql://localhost/personnesvoitures",
                "root",
                "");
}
Catch(SQLException e) {
        e.printStackTrace();
}
```

Faire une requête à la base de données

- Objet Statement
  - Interface représentant une requête Sql
  - Les objets statements s'executent
  - Renvoient des objets de types ResultSet

- Création d'un Statement

```
Statement stmt = con.createStatement();
```

- Exécution d'un Statement

```
ResultSet stmt.executeQuery(String requete)
```

- Fermeture d'un Statement

```
Stmt.close();
```

Exemple de requête

```
Statement stmt = connexion.createStatement();
ResultSet rst = stmt.executeQuery("SELECT Prenom FROM" +
        "personnesvoitures.personnes");
```

Parcourir le résultat d'une requête

- Objet ResultSet
  - Utilise un curseur sur une liste de tuples
- Par défaut le curseur est avant la première ligne
- Récupérer la valeur d'une colonne d'un tuple

```
resultSet.get[Type](“[Nom_de_la_colonne”]);
String s = resultSet.getString(“Nom_de_la_colonne”);
int i = resultSet.getInt(“Nom_de_la_colonne”);
```

- Avancer dans la liste des tuples

```
resultSet.next(); //Retoune faux si le pointeur est après la dernière ligne
```

Exemple de code

```
Statement stmt = connexion.createStatement();
ResultSet rst = stmt.executeQuery("SELECT Prenom , DateNaissance" +
" FROM personnesvoitures.personnes");

While(rst.next()) {
        String prenom = rst.getString("Prenom");
        java.sql.Date birth = rst.getDate("DateNaissance");
        }
```

Affichage prénom et date naissance :

```
Connection con;
try {
con = DriverManager.getConnection(
                "jdbc:mysql://localhost/personnesvoitures", #connecteur
                "root", #login
                "");      #password
Statement stmt = con.createStatement();
ResultSet rst = stmt.executeQuery("SELECT Prenom ,"+
" DateNaissance FROM personnesvoitures.personnes");
while (rst.next()) {
   String s = rst.getString("Prenom");
   Date d = rst.getDate("DateNaissance");
   System.out.println(s + " est né " + d.toString());
                    }
} catch (SQLException e) {
 // TODO Auto-generated catch block
   e.printStackTrace();
}
```

Affichage prénom et date naissance :

```
Connection con;
try {
con = DriverManager.getConnection(
                "jdbc:mysql://localhost/personnesvoitures", #connecteur
                "root", #login
                "");      #password
Statement stmt = con.createStatement();
ResultSet rst = stmt.executeQuery("SELECT Prenom ,"+
" DateNaissance FROM personnesvoitures.personnes");
while (rst.next()) {
   String s = rst.getString("Prenom");
   Date d = rst.getDate("DateNaissance");
   System.out.println(s + " est né " + d.toString());
                    }
} catch (SQLException e) {
 // TODO Auto-generated catch block
   e.printStackTrace();
}
```

1) Connexion

Affichage prénom et date naissance :

```
Connection con;
try {
con = DriverManager.getConnection(
                "jdbc:mysql://localhost/personnesvoitures", #connecteur
                "root", #login
                "");      #password
Statement stmt = con.createStatement();
ResultSet rst = stmt.executeQuery("SELECT Prenom ,"+
" DateNaissance FROM personnesvoitures.personnes");
while (rst.next()) {
   String s = rst.getString("Prenom");
   Date d = rst.getDate("DateNaissance");
   System.out.println(s + " est né " + d.toString());
                    }
} catch (SQLException e) {
 // TODO Auto-generated catch block
   e.printStackTrace();
}
```

2) Préparation de la requête

Affichage prénom et date naissance :

```
Connection con;
try {
con = DriverManager.getConnection(
                "jdbc:mysql://localhost/personnesvoitures", #connecteur
                "root", #login
                "");      #password
Statement stmt = con.createStatement();
ResultSet rst = stmt.executeQuery("SELECT Prenom ,"+
" DateNaissance FROM personnesvoitures.personnes");
while (rst.next()) {
   String s = rst.getString("Prenom");
   Date d = rst.getDate("DateNaissance");
   System.out.println(s + " est né " + d.toString());
                }
} catch (SQLException e) {
   // TODO Auto-generated catch block
   e.printStackTrace();
}
```

3) Traitement du résultat de la requête

Création de graphes avec jFreeChart

Information Visualisation

The use of computer-supported, interactive, visual representations of abstract data in order to amplify cognition.

Stuart K. Card

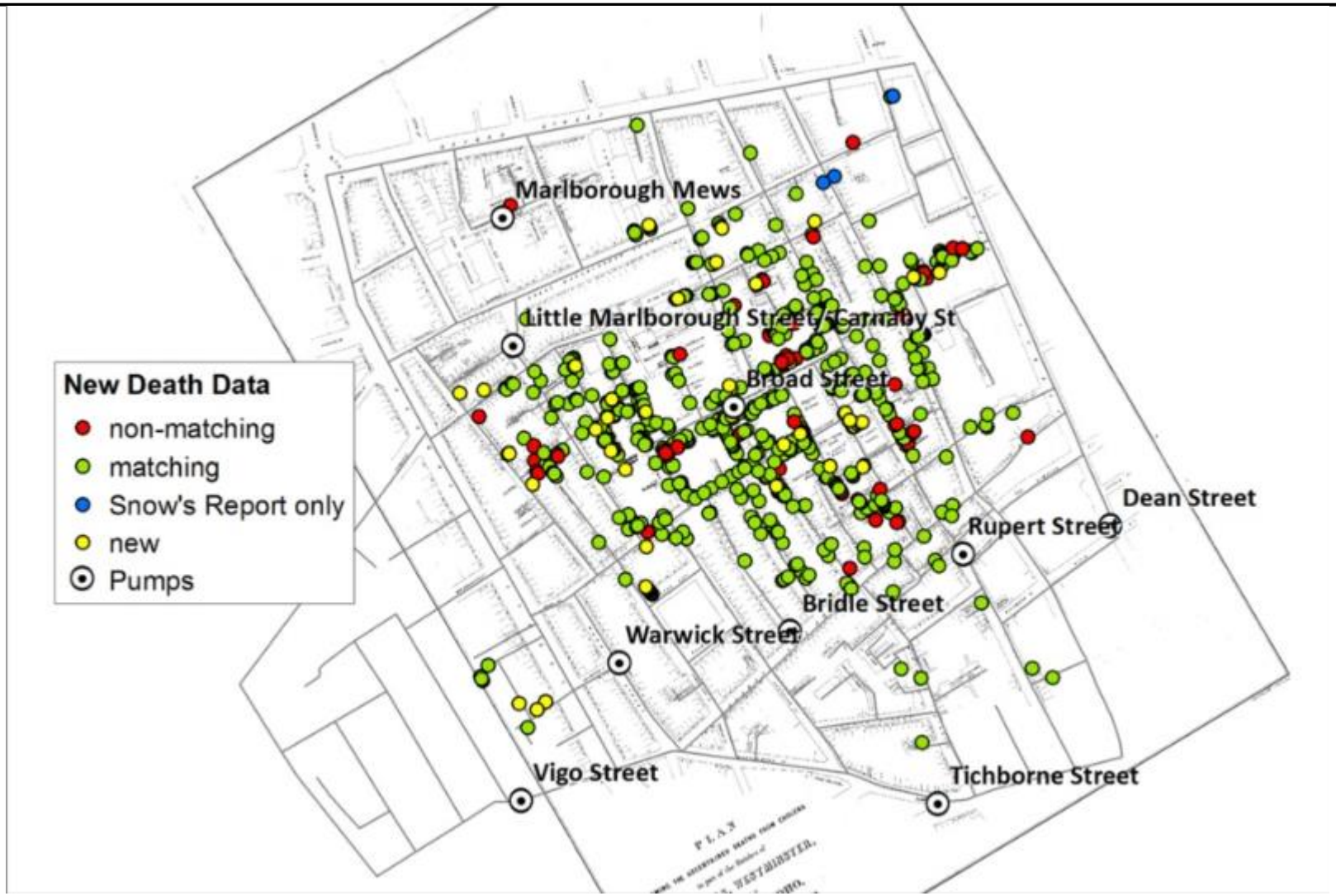
John Mackinlay

Ben Shneiderman

Exemples

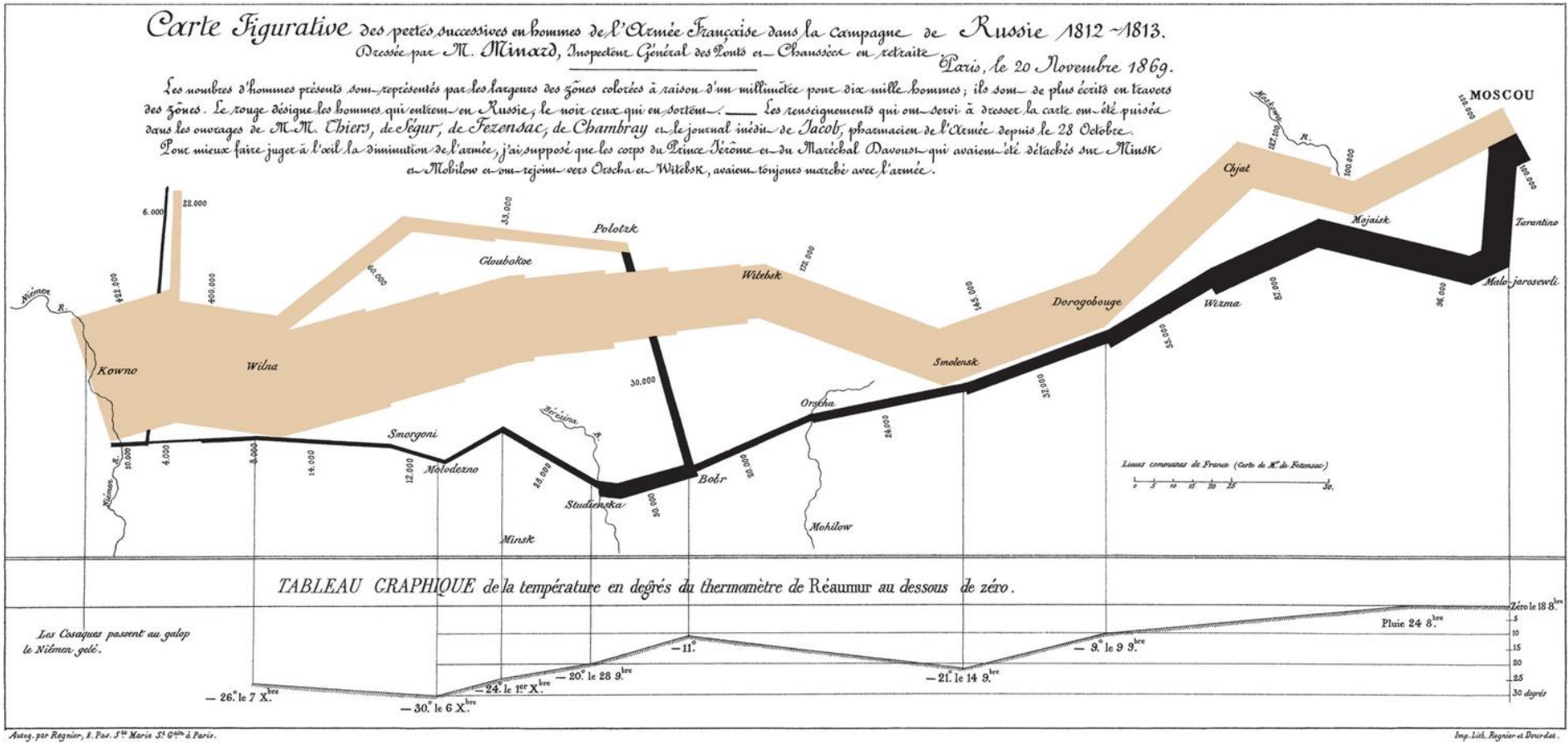
John Snow
1854

Exemples



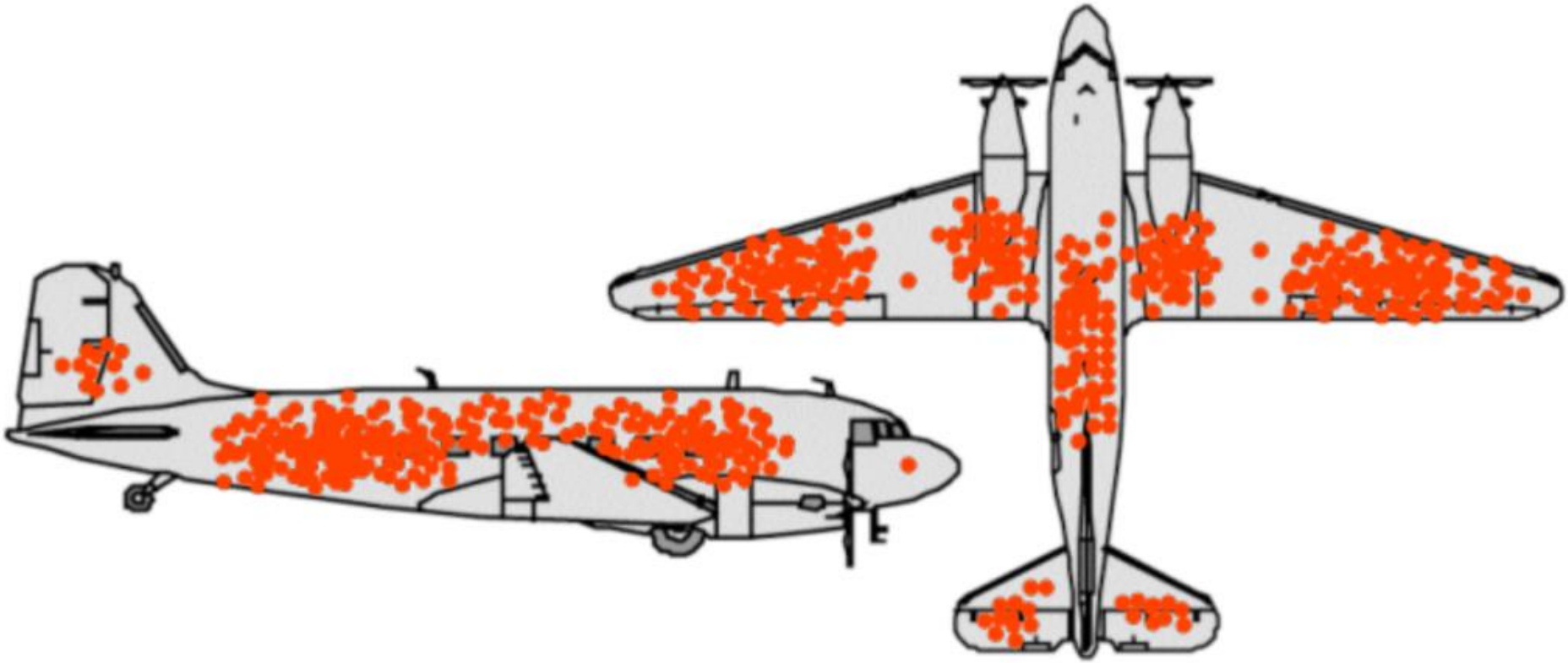
Shiode, N., Shiode, S., Rod-Thatcher, E., Rana, S., & Vinten-Johansen, P. (2015). The mortality rates and the space-time patterns of John Snow's cholera epidemic map. *International journal of health geographics*, *14*(1), 21.

Exemples

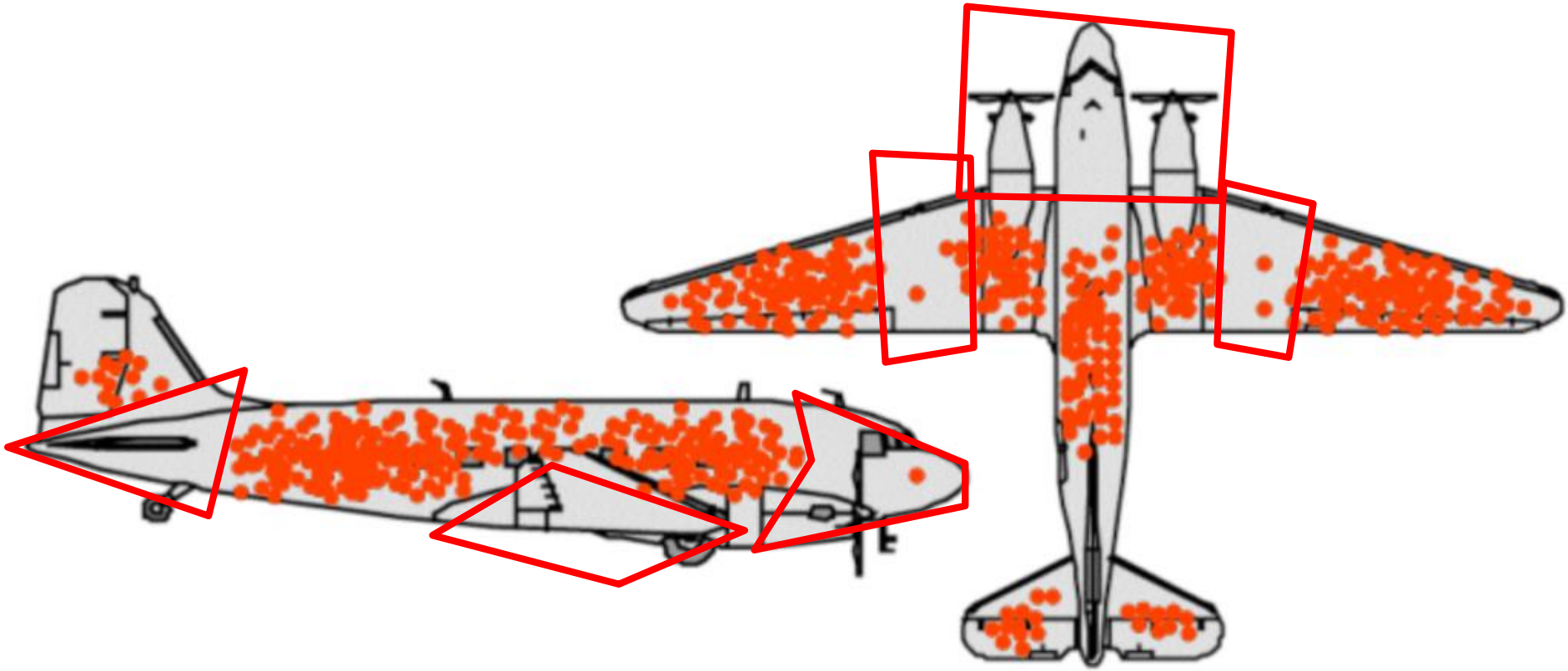


Charles Joseph Minard, 1869

Petit jeu…



Petit jeu...



Exemples de graphes selon le temps

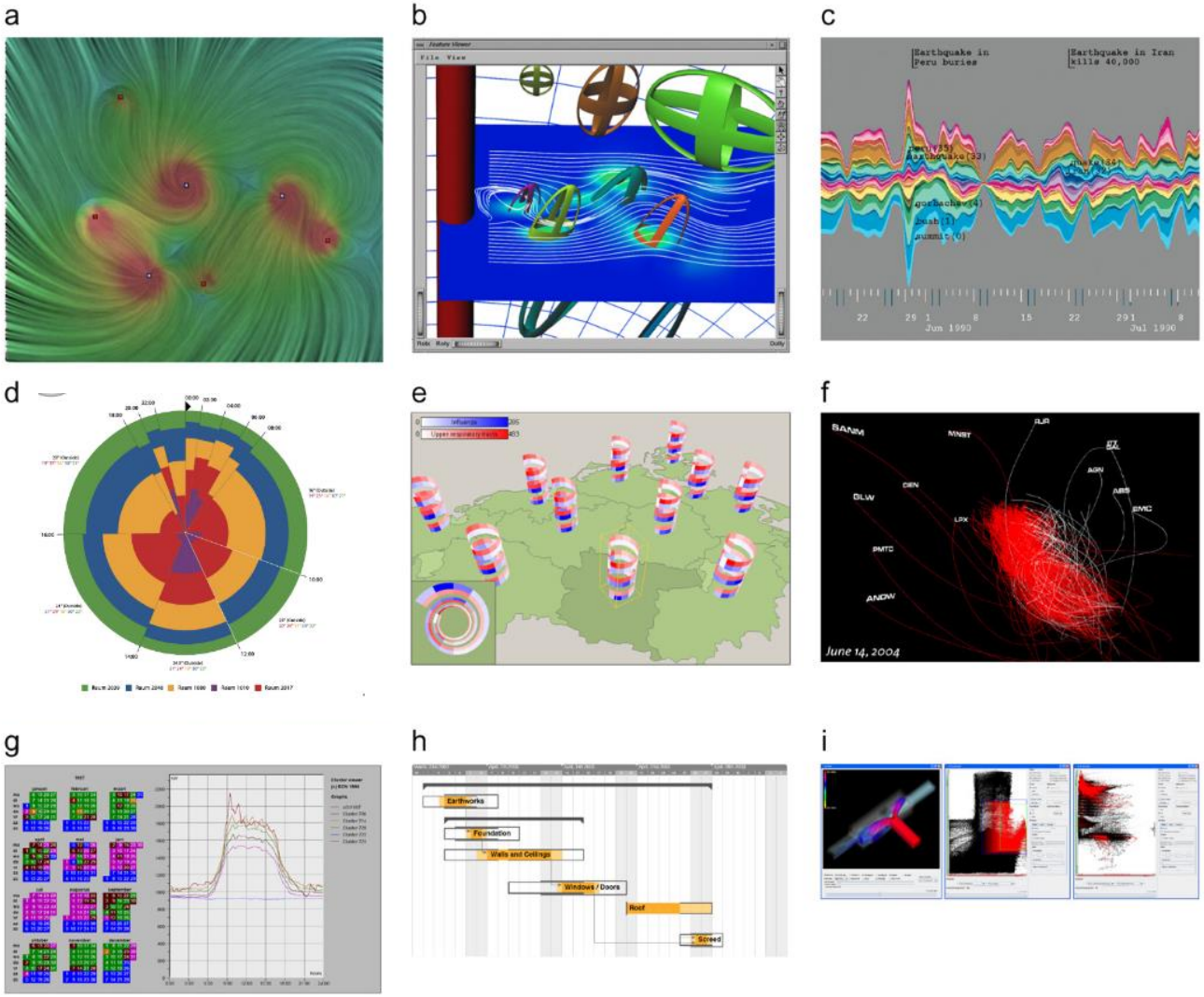


Fig. 3. Examples of techniques for visualizing time-oriented data.

Graphes

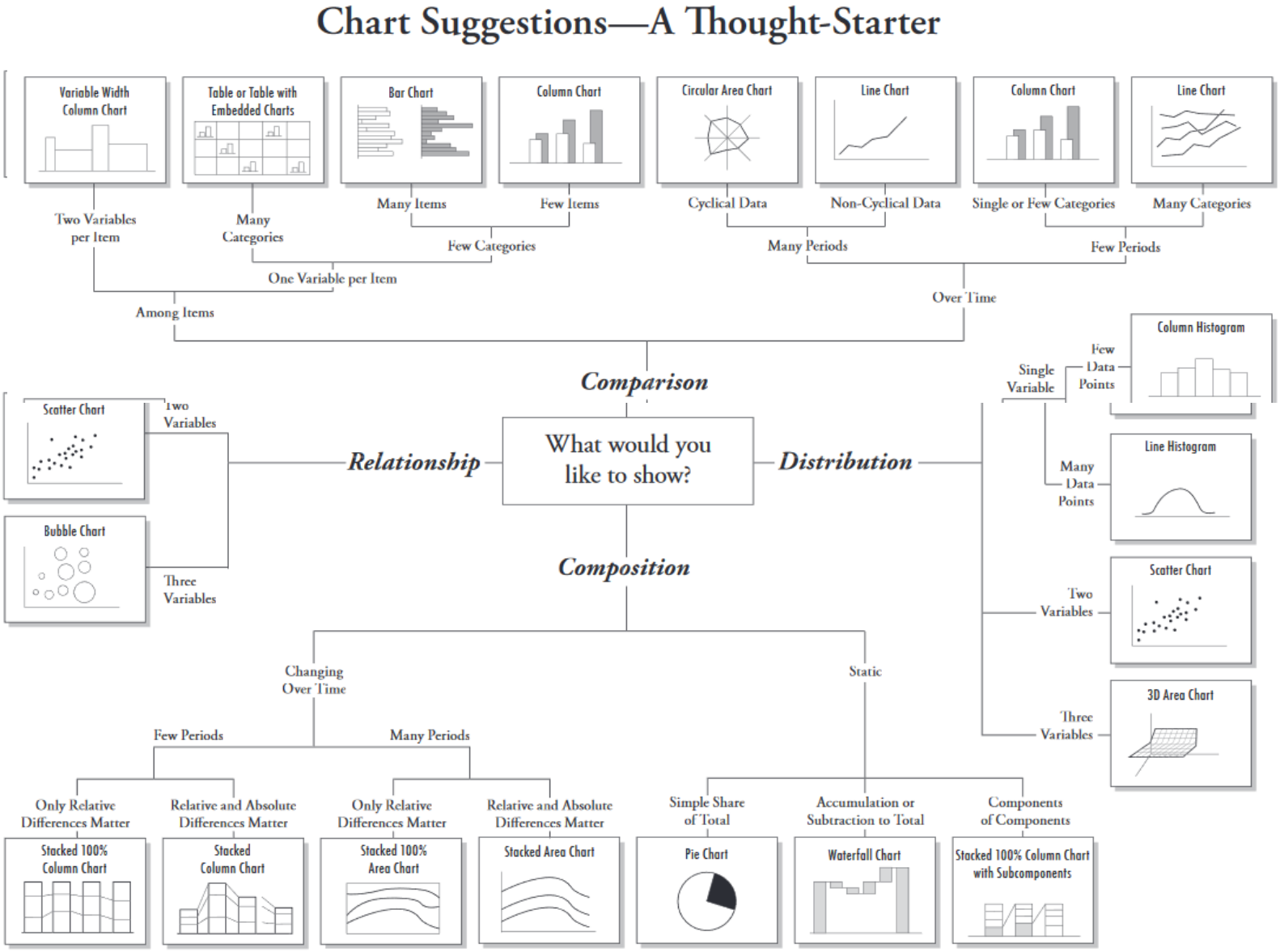


Chart Suggestions—A Thought-Starter
Variable Width Column Chart
Table or Table with Embedded Charts
Bar Chart
Column Chart
Circular Area Chart
Line Chart
Column Chart
Line Chart
Two Variables per Item
Many Categories
Many Items
Few Items
Cyclical Data
Non-Cyclical Data
Single or Few Categories
Many Categories
Few Categories
Many Periods
Few Periods
One Variable per Item
Among Items
Over Time
Comparison
Scatter Chart
Two Variables
Bubble Chart
Three Variables
Relationship
What would you like to show?
Distribution
Composition
Column Histogram
Single Variable
Few Data Points
Line Histogram
Many Data Points
Scatter Chart
Two Variables
3D Area Chart
Three Variables
Changing Over Time
Static
Few Periods
Many Periods
Only Relative Differences Matter
Relative and Absolute Differences Matter
Only Relative Differences Matter
Relative and Absolute Differences Matter
Simple Share of Total
Accumulation or Subtraction to Total
Components of Components
Stacked 100% Column Chart
Stacked Column Chart
Stacked 100% Area Chart
Stacked Area Chart
Pie Chart
Waterfall Chart
Stacked 100% Column Chart with Subcomponents

Graphes

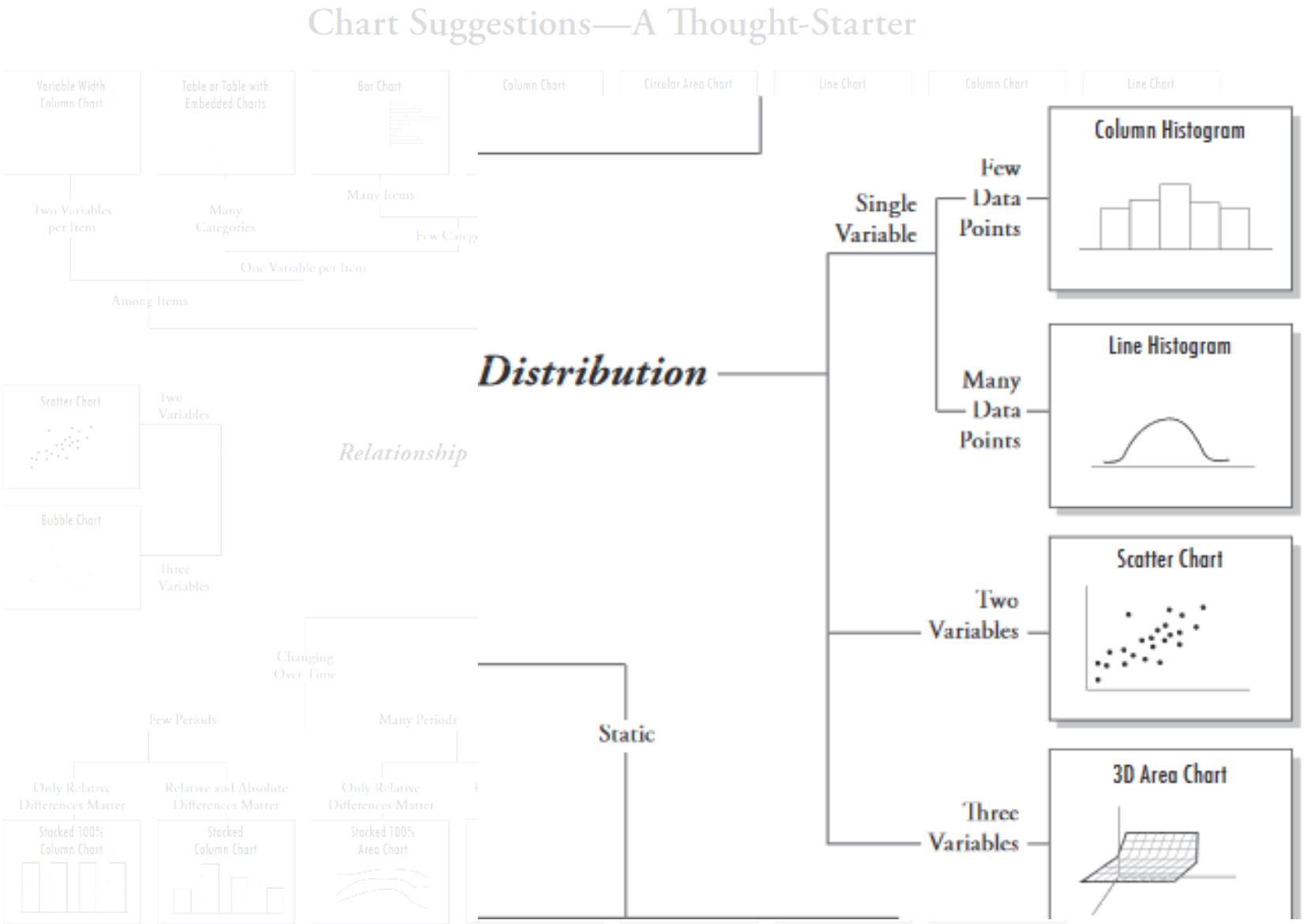


Chart Suggestions—A Thought-Starter
Distribution
Single Variable
Few Data Points
Column Histogram
Many Data Points
Line Histogram
Two Variables
Scatter Chart
Three Variables
3D Area Chart
Relationship
Two Variables
Scatter Chart
Three Variables
Bubble Chart
Static
Changing Over Time
Few Periods
Many Periods
Only Relative Differences Matter
Relative and Absolute Differences Matter
Only Relative Differences Matter
Stacked 100% Column Chart
Stacked Column Chart
Stacked 100% Area Chart
Variable Width Column Chart
Table or Table with Embedded Charts
Bar Chart
Column Chart
Circular Area Chart
Line Chart
Column Chart
Line Chart
Many Items
Few Categories
Two Variables per Item
Many Categories
One Variable per Item
Among Items

Graphes

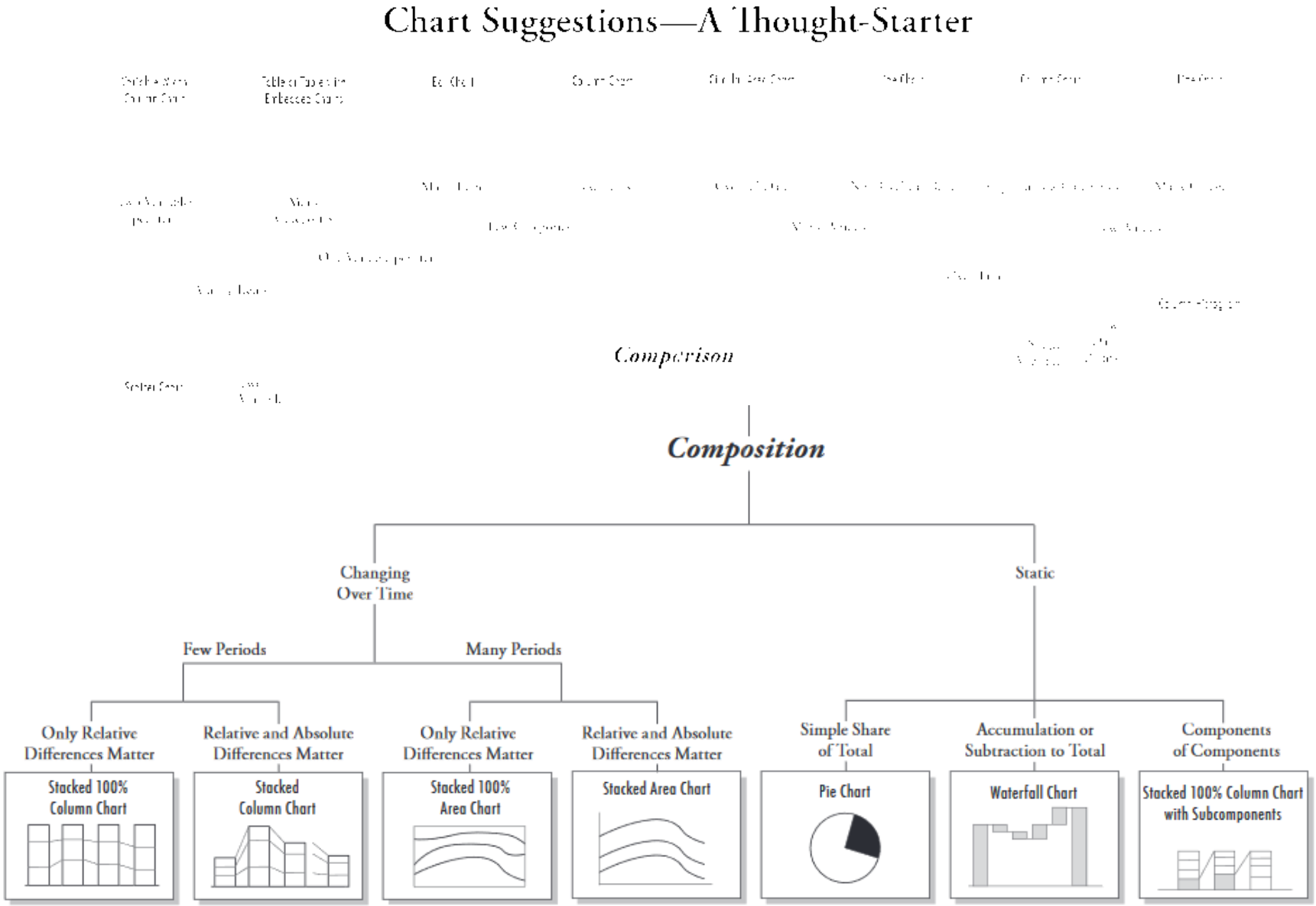


Chart Suggestions—A Thought-Starter
Comparison
Composition
Changing Over Time
Static
Few Periods
Many Periods
Only Relative Differences Matter
Relative and Absolute Differences Matter
Only Relative Differences Matter
Relative and Absolute Differences Matter
Simple Share of Total
Accumulation or Subtraction to Total
Components of Components
Stacked 100% Column Chart
Stacked Column Chart
Stacked 100% Area Chart
Stacked Area Chart
Pie Chart
Waterfall Chart
Stacked 100% Column Chart with Subcomponents

Graphes

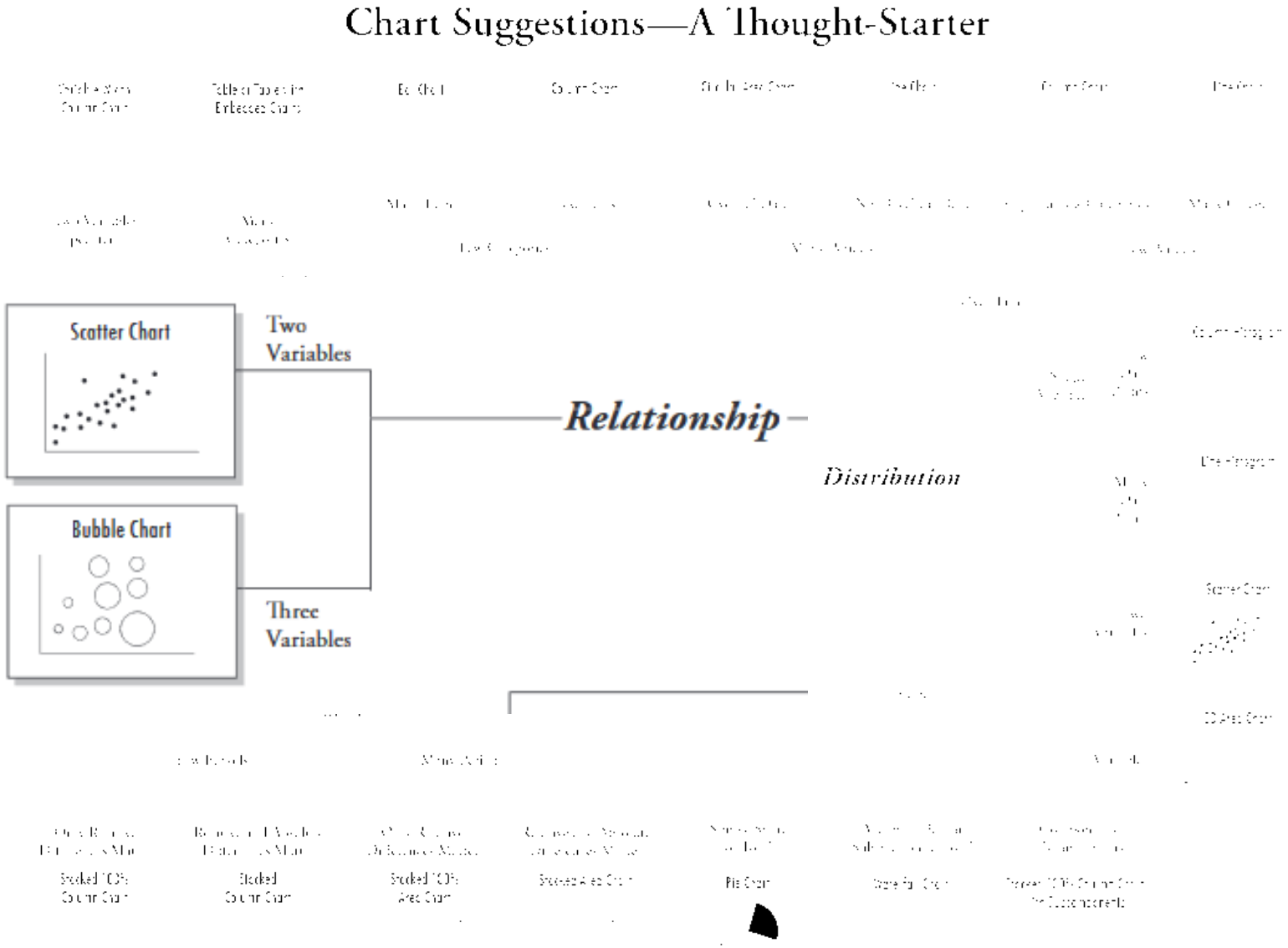


Chart Suggestions—A Thought-Starter
Scatter Chart
Two Variables
Bubble Chart
Three Variables
Relationship
Distribution

Graphes

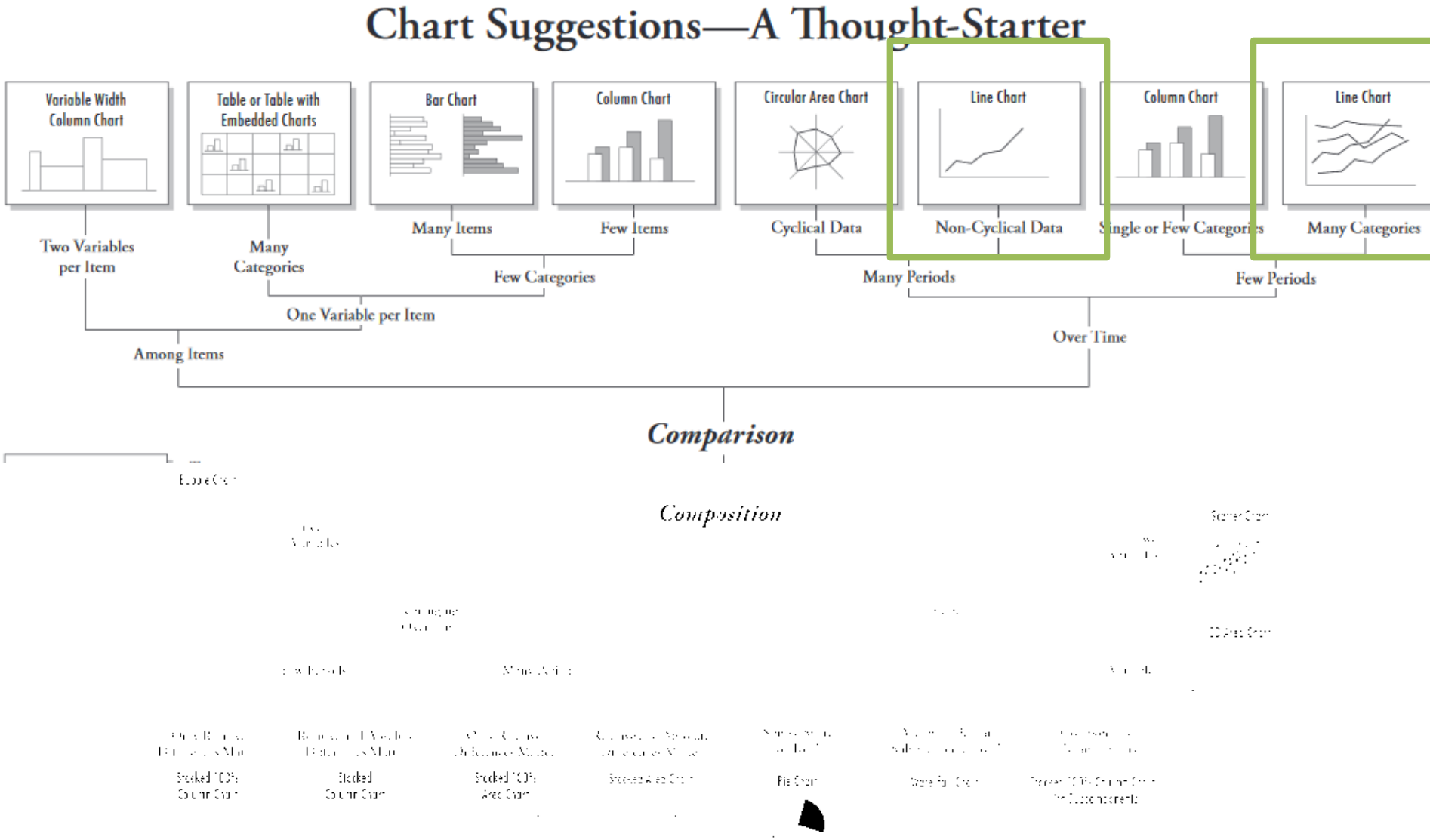


Chart Suggestions—A Thought-Starter
Variable Width Column Chart
Table or Table with Embedded Charts
Bar Chart
Column Chart
Circular Area Chart
Line Chart
Column Chart
Line Chart
Two Variables per Item
Many Categories
Many Items
Few Items
Cyclical Data
Non-Cyclical Data
Single or Few Categories
Many Categories
Few Categories
Many Periods
Few Periods
One Variable per Item
Over Time
Among Items
Comparison
Composition

Introduction JFreeChart

- JFreeChart open source API crée en février 2000 par David Gilbert

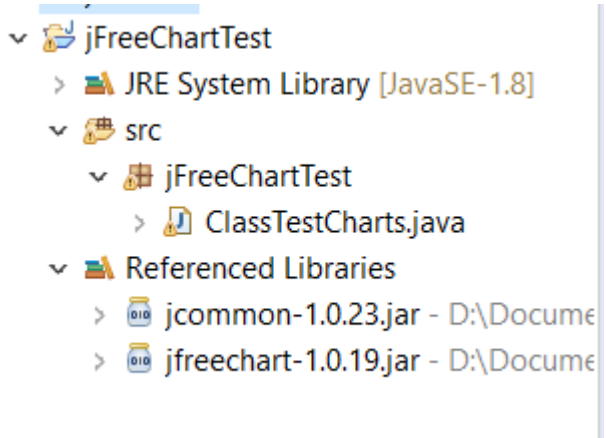
- Pourquoi cette libraire ?
  - API documenté
  - Supporte plusieurs types de graphes
  - Peut être utilisé côté client et côté serveur
  - Supporte plusieurs formats de sortie PNG, JPEG, PDF, SVG, etc.
  - Permet la modification des graphes

Installation JFreeChart

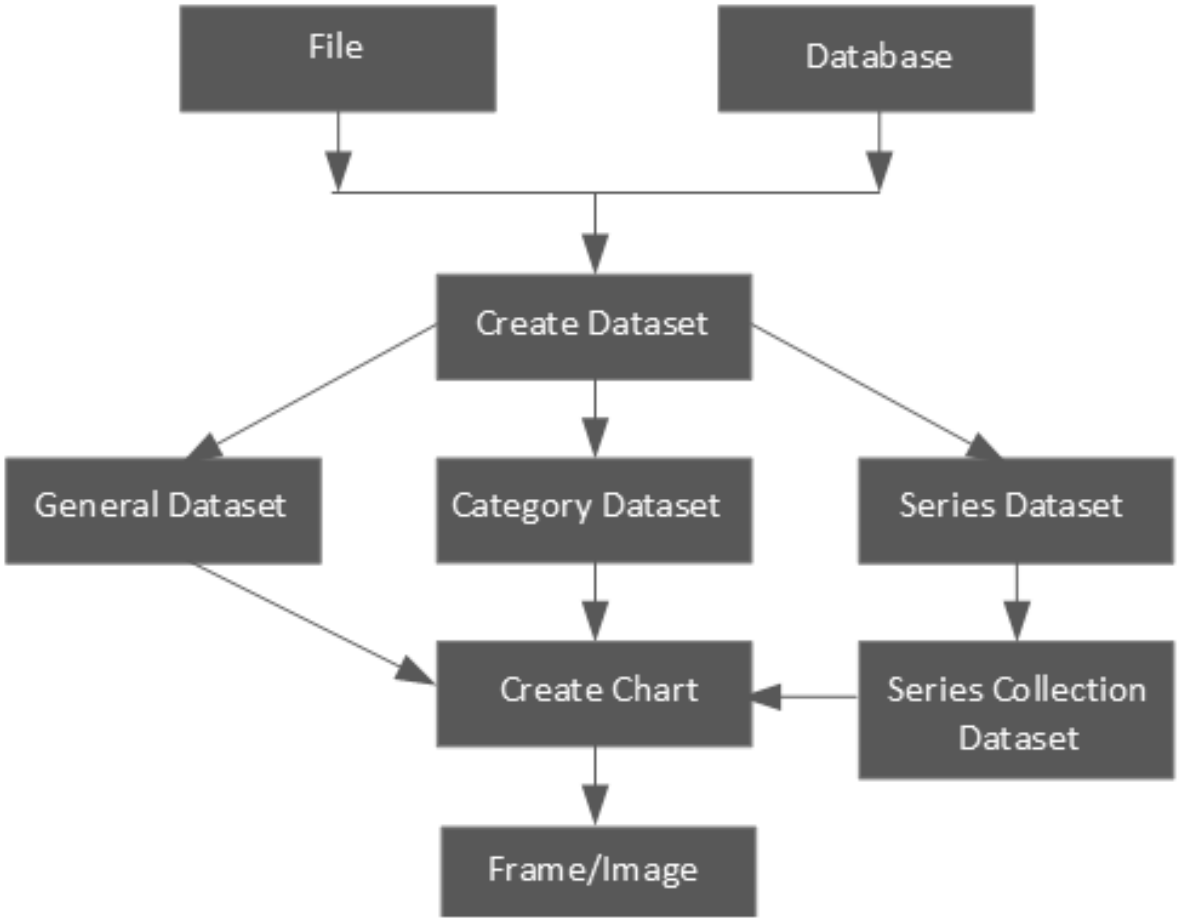
- Télécharger ici (ou sur https://sourceforge.net/projects/jfreechart/files/) et dézipper le fichier.

- Créer votre projet java.

- Ajouter les archives .jar suivantes (voir partie sur les bases de données)
  - jcommon (ou lien ici)
  - jfreechart (ou lien ici)

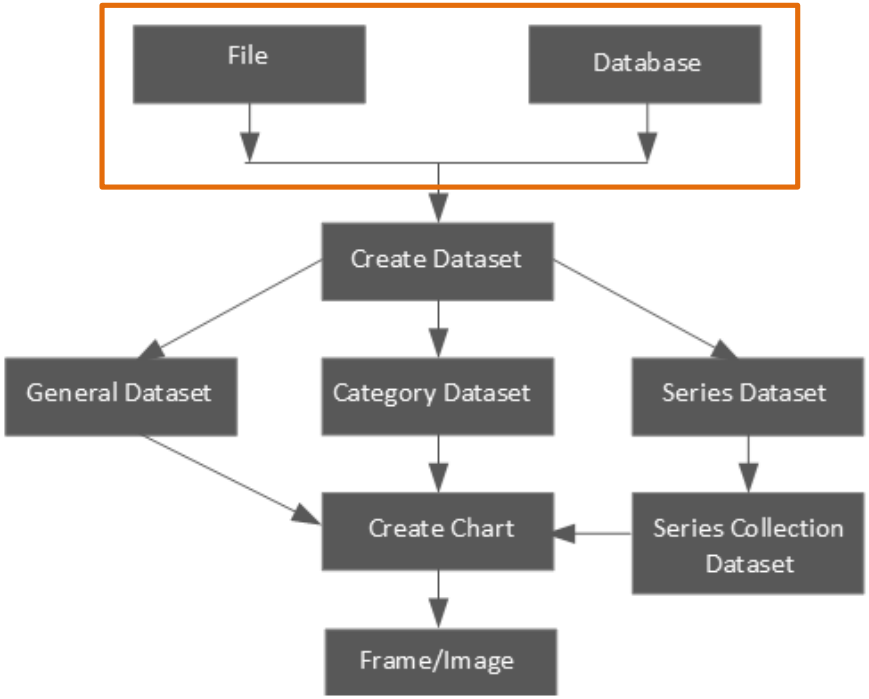


Architecture



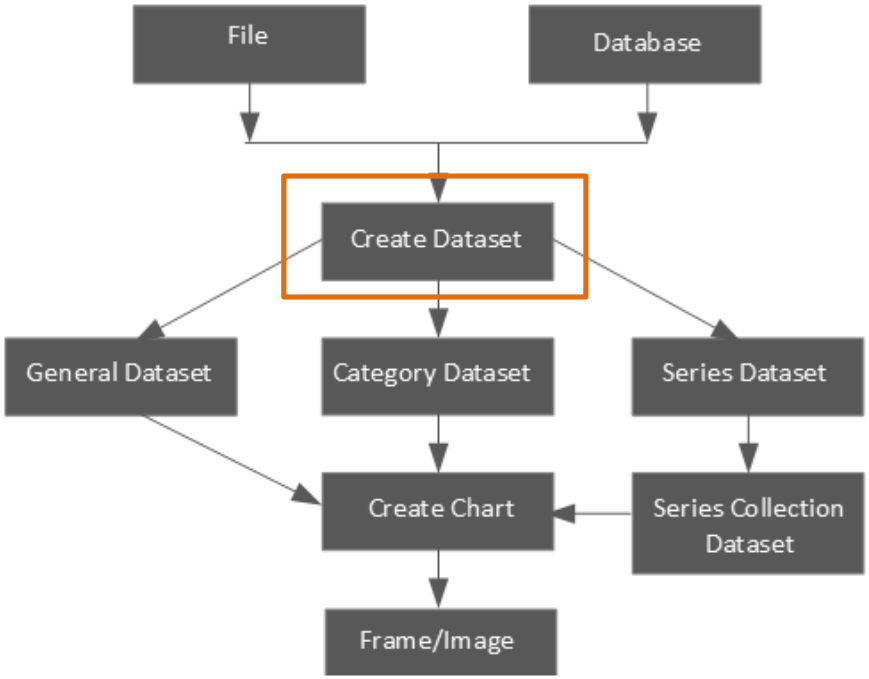
Architecture

- File
  - Stockage des données dans un fichier (.csv ou autres)
- Database
  - Stockage des données dans une base de données relationnelles (ce que l'on a vu avant)



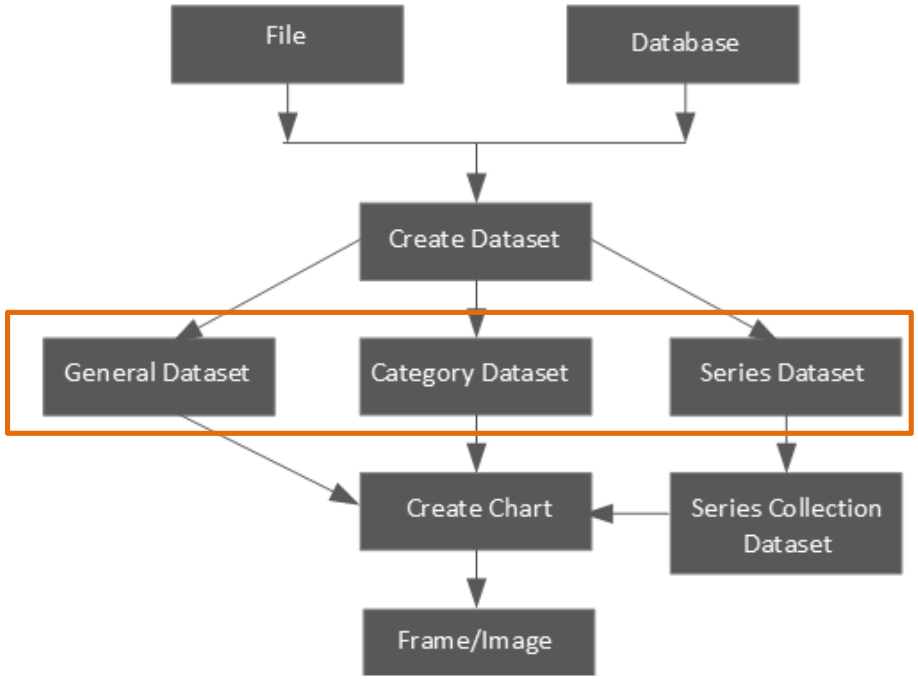
Architecture

- Create Dataset
  - Permet à partir de la source de créer un objet Dataset



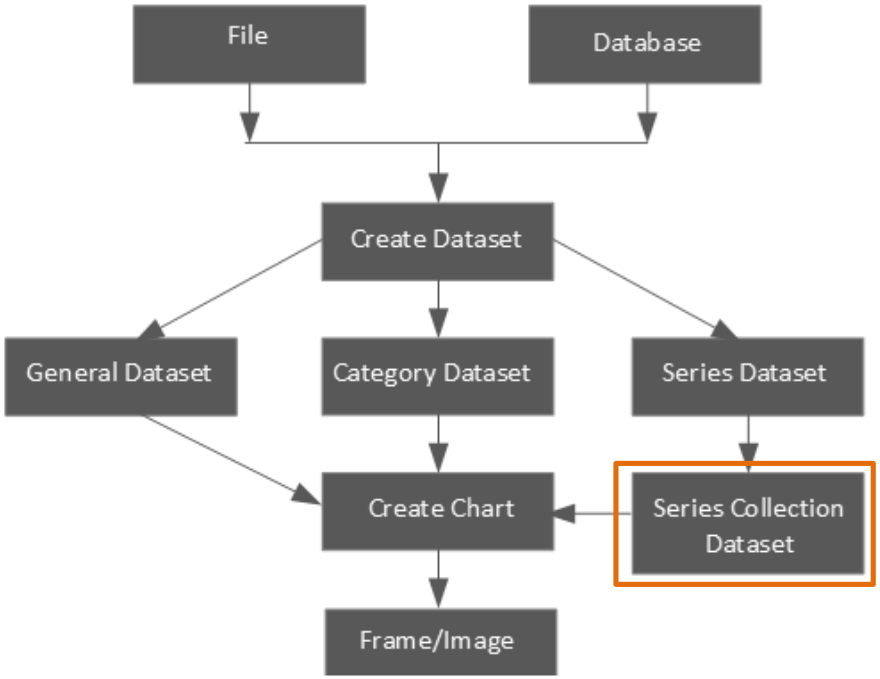
Architecture

- General Dataset
  - Type utilisé pour les camemberts
- Category Dataset
  - Type utilisé pour les histogrammes et les courbes
- Series Dataset
  - Stocke des séries de données et construire des lignes



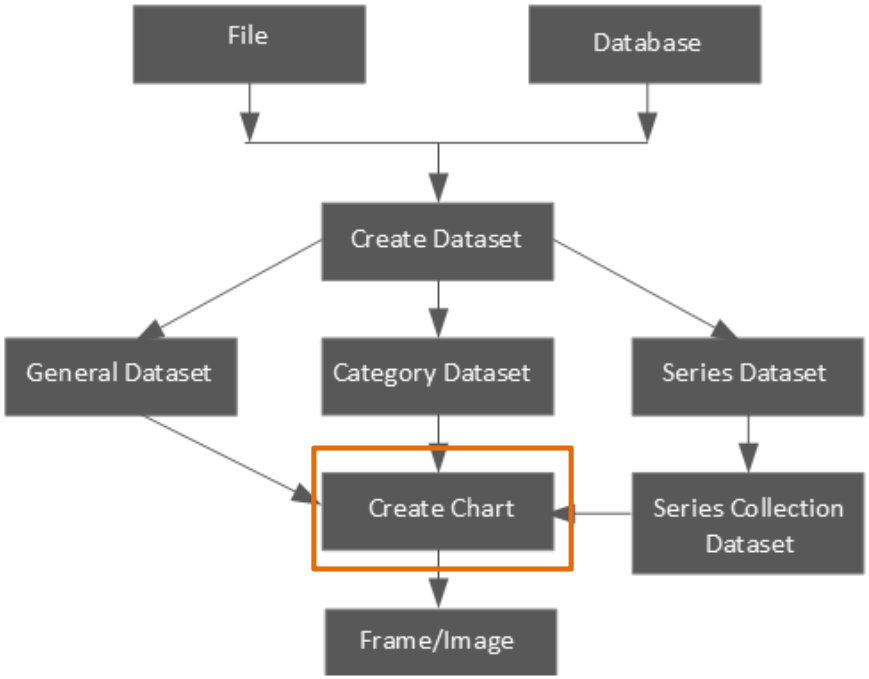
Architecture

- Series Collection Dataset
  - Chaque série est ajouté dans un object de type collection



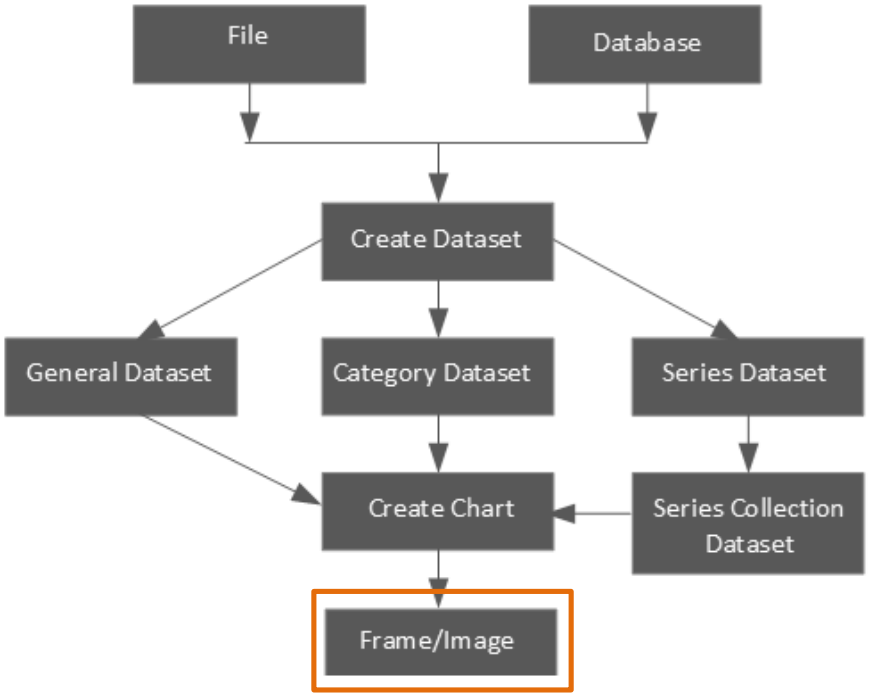
Architecture

- Create Chart
  - Cette méthode quand elle s'éxécutera créera le diagramme final



Architecture

- Frame/Image
  - Le graphe est affiché dans une fenêtre Swing



Création de graphes et d'une frame

- Création de graphes à l'aide de la classe ChartFactory()

- Construction d'une fenêtre permettant d'afficher un graphe

```
ChartFrame (java.lang.Frame String, JfreeChart chart)
```

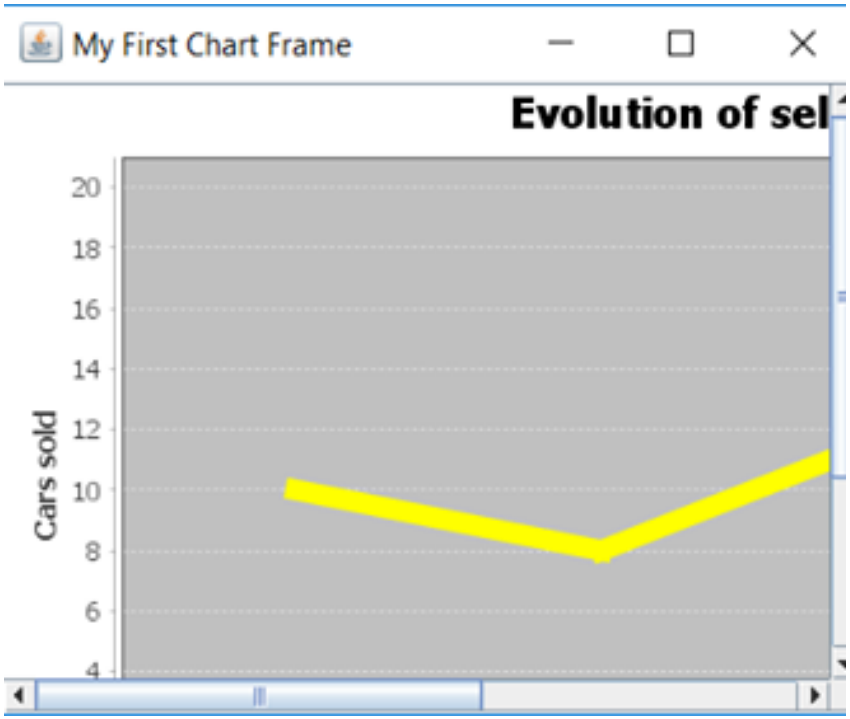
```
ChartFrame (java.lang.Frame String, JfreeChart chart; Boolean scrollpane)
```

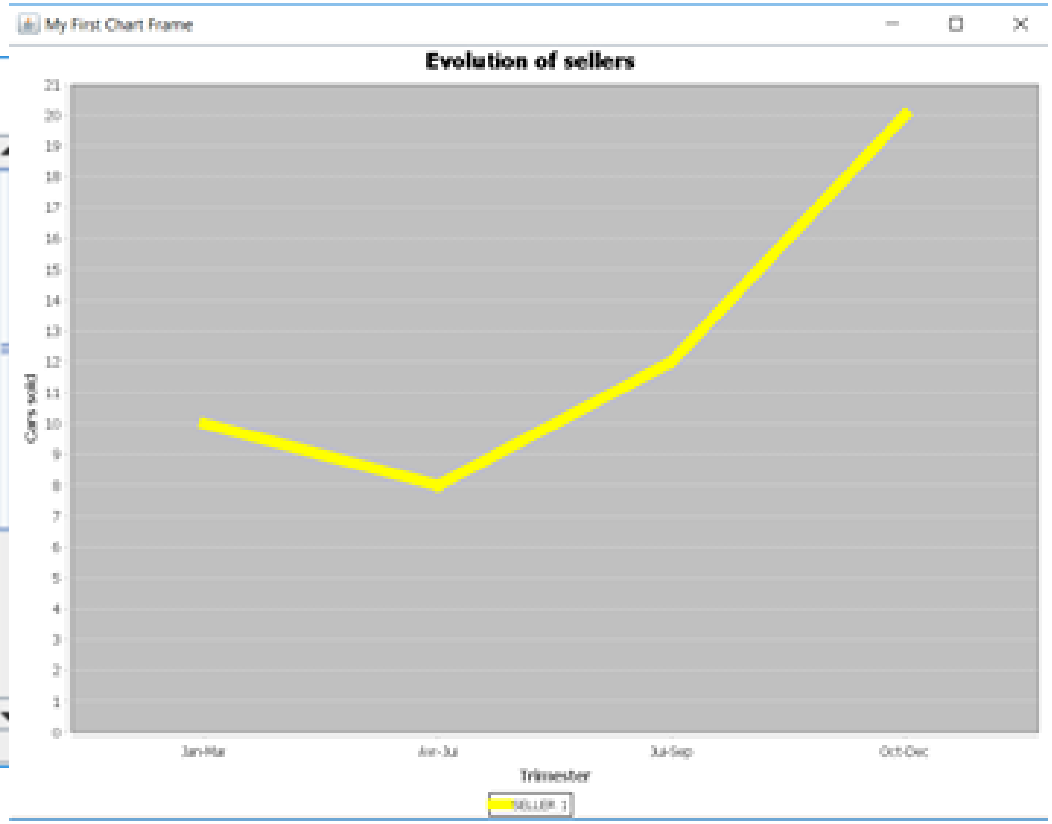
- Méthode pour retourner le chart panel d'une frame

```
getChartPanel()
```

Exemple de code

```
JFreeChart chart = createLineChart(createCategoryDataset());
ChartFrame cf = new ChartFrame("My First Chart Frame", chart , true);
cf.setVisible(true);
```





ChartPanel

- Comme un composant un Swing pour afficher un graphe JFreeChart

- Constructeur permettant d'afficher un graphe

```
ChartPanel(JFreeChart chart)
```

```
ChartPanel(JFreeChart chart, boolean useBuffer)
```

```
ChartPanel(JFreeChart chart, boolean properties, boolean save, boolean print, Boolean zoom, Boolean tooltips)
```

- Méthode pour changer la taille

```
setPreferredSize(Dimension)
```

Exemple de code

```
JFrame f = new JFrame("My JFrame but what will be in it ?");
JFreeChart chart = createLineChart(createCategoryDataset2());

ChartPanel cp = new ChartPanel(chart,true);

JLabel j = new JLabel("I Created a chart above ?");

JPanel jpanelText = new JPanel();
jpanelText.setLayout(new
FlowLayout(FlowLayout.CENTER));
jpanelText.add(j);

JSplitPane jsp = new JSplitPane(0,cp,
jpanelText);
jsp.setDividerLocation(450);
f.add(jsp);

f.setSize(1620,900);
f.setVisible(true);
```

JfreeChart

- Le cœur de l'api -> contient les méthodes pour créer les graphes

- Constructeurs pour créer un nouveau graphe

```
JFreeChart(Plot p)
```

```
JFreeChart(String title, Plot plot)
```

- Pour obtenir un Plot

```
Plot getPlot()
```

```
XYPlot getXYPlot()
```

Création du jeu de données

- Plusieurs interface permettent de créer plusieurs types de jeux de donnée
  - PieDataset -> PieChart
  - CategoryDataset -> BarChart, LineChart
  - XYDataset -> LineChart, TimeSeries

AbstractDataset, AbstractIntervalXYDataset, AbstractSeriesDataset, AbstractXYDataset, AbstractXYZDataset, CategoryTableXYDataset, CategoryToPieDataset, CyclicXYItemRenderer.OverwriteDataSet ,DefaultBoxAndWhiskerCategoryDataset, DefaultBoxAndWhiskerXYDataset, DefaultCategoryDataset, DefaultHeatMapDataset, DefaultHighLowDataset, DefaultIntervalCategoryDataset, DefaultIntervalXYDataset, DefaultKeyedValueDataset, DefaultKeyedValues2DDataset, DefaultKeyedValuesDataset, DefaultMultiValueCategoryDataset, DefaultOHLCDataset, DefaultPieDataset, DefaultStatisticalCategoryDataset, DefaultTableXYDataset, DefaultValueDataset, DefaultWindDataset, DefaultXYDataset, DefaultXYZDataset, DynamicTimeSeriesCollection, HistogramDataset, JDBCCategoryDataset, JDBCPieDataset, JDBCXYDataset, MatrixSeriesCollection, OHLCSeriesCollection, SimpleHistogramDataset, SlidingCategoryDataset, SlidingGanttCategoryDataset, TaskSeriesCollection, TimePeriodValuesCollection, TimeSeriesCollection, TimeTableXYDataset, VectorSeriesCollection, WaferMapDataset, XIntervalSeriesCollection, XYBarDataset, XYIntervalSeriesCollection, XYSeriesCollection, XYTaskDataset, YIntervalSeriesCollection

```
DefaultPieDataset dataset = new DefaultPieDataset();
```

Création d'un pie chart

- Création d'un jeu de données

```
DefaultPieDataset dataset = new DefaultPieDataset();
```

- Ajouter une donnée

```
setValue(String key, double value)
```

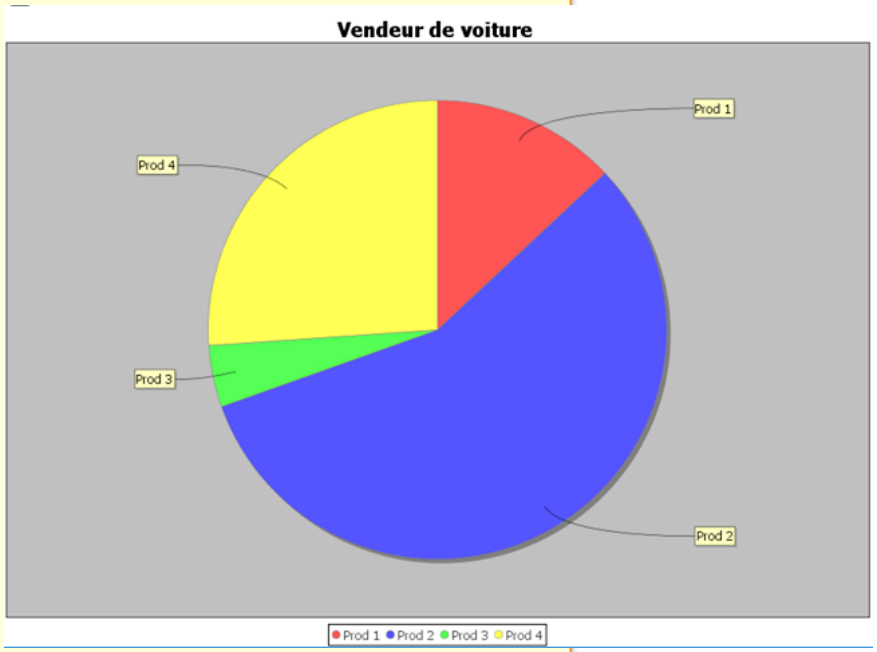
- Création du graphe à partir des données

```
ChartFactory.createPieChart(String title, PieDataset dataset)
```

```
ChartFactory.createPieChart(String title, PieDataset dataset, Boolean legend
, Boolean tooltips, Boolean urls)
```

Exemple de code

```
private static PieDataset createDataset() {
        DefaultPieDataset dataset = new
        DefaultPieDataset();
        dataset.setValue("Prod 1", 30);
        dataset.setValue("Prod 2", 130);
        dataset.setValue("Prod 3", 10);
        dataset.setValue(“Prod 4”, 60);
        return dataset;
    }

private static JFreeChart createChart(PieDataset d) {
        JFreeChart chart = ChartFactory.createPieChart("Vendeur de
        voiture ",d,true,true, false);
        return chart;
    }
```



Création d'un Bar chart

- Création d'un jeu de données

```
DefaultCategoryDataset dataset = new DefaultCategoryDataset( );
```

- Ajouter une donnée

```
setValue(double value, Comparable rowKey, Comparable columnKey)
```

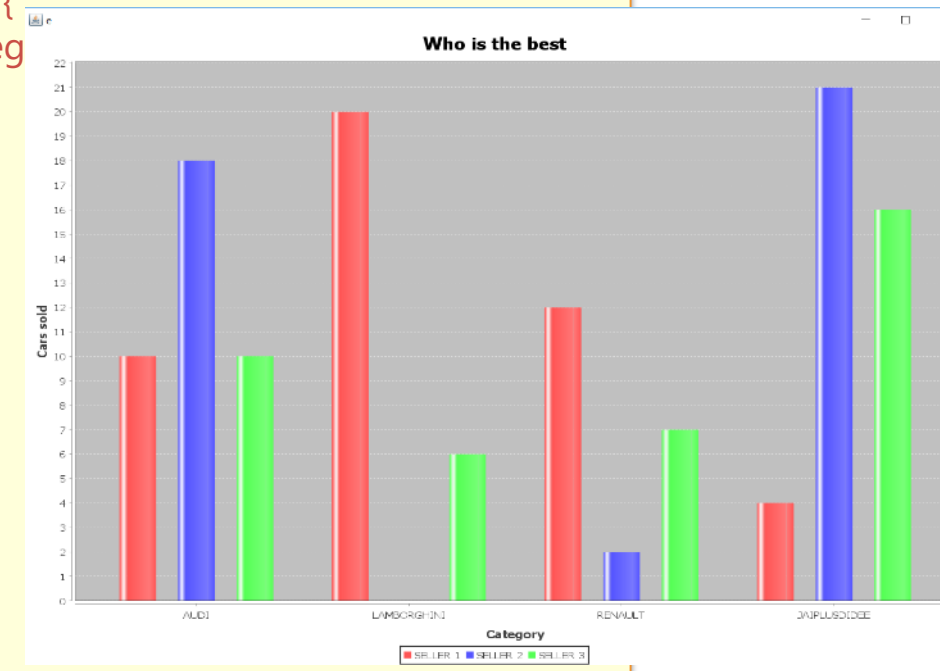
- Création du graphe à partir des données

```
createBarChart(String title,String categoryAxisLabel,
String valueAxisLabel,CategoryDataset dataset)
```

```
ChartFactory.createBarChart(String title, String categoryAxisLabel,
String valueAxisLabel, CategoryDataset dataset, PlotOrientation orientation,
 boolean legend, boolean tooltips, boolean urls)
```

Exemple de code

```
private static CategoryDataset createCategoryDataset() {
      DefaultCategoryDataset dataset = new DefaultCateg
      dataset.setValue(10, "SELLER 1" , "AUDI");
      dataset.setValue(20, "SELLER 1" , "LAMBORGHINI");
      dataset.setValue(12, "SELLER 1" , "RENAULT");
      dataset.setValue(4, "SELLER 1" , "JAIPLUSDIDEE");

      dataset.setValue(18, "SELLER 2" , "AUDI");
      dataset.setValue(0, "SELLER 2" , "LAMBORGHINI");
      dataset.setValue(2, "SELLER 2" , "RENAULT");
      dataset.setValue(21, "SELLER 2" , "JAIPLUSDIDEE");

      dataset.setValue(10, "SELLER 3" , "AUDI");
      dataset.setValue(6, "SELLER 3" , "LAMBORGHINI");
      dataset.setValue(7, "SELLER 3" , "RENAULT");
      dataset.setValue(16, "SELLER 3" , "JAIPLUSDIDEE");

      return dataset;
   }
private static JFreeChart createChart(CategoryDataset d) {
     JFreeChart chart = ChartFactory.createBarChart("Who is the best" , "Category",
"Cars sold", d, PlotOrientation.VERTICAL, true, true, false);
     return chart;
   }
```



Création d'un Line chart avec un categoryDataset

- Création d'un jeu de données

```
DefaultCategoryDataset dataset = new DefaultCategoryDataset();
```

- Ajouter une donnée

```
setValue(double value, Comparable rowKey, Comparable columnKey)
```

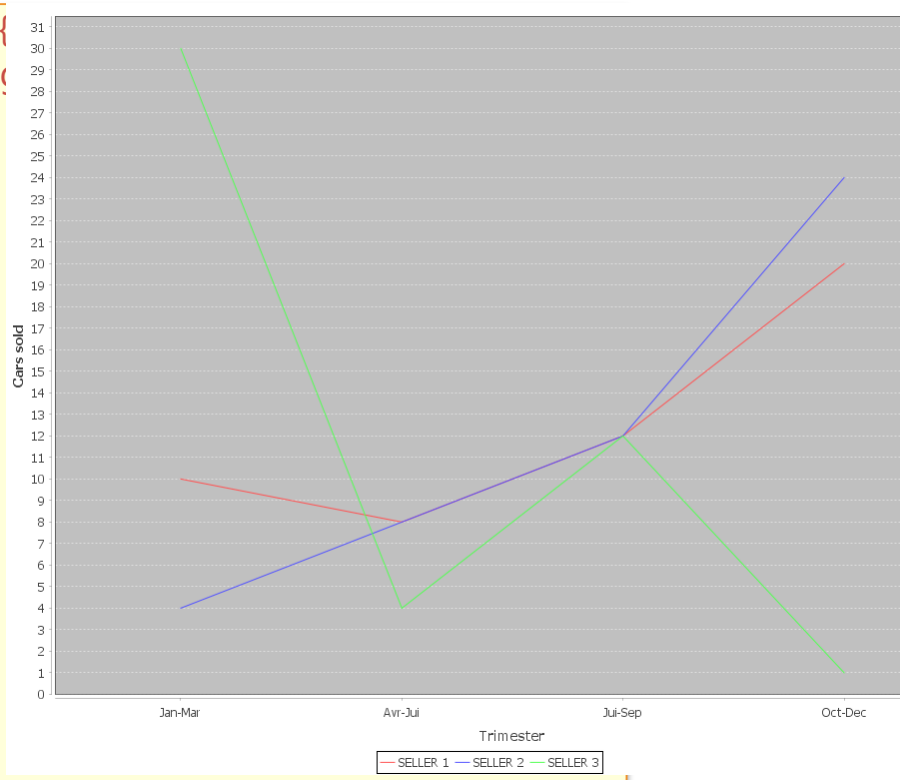
- Création du graphe à partir des données

```
ChartFactory.createLineChart(String title, String categoryAxisLabel,
String valueAxisLabel, CategoryDataset dataset)
```

```
ChartFactory.createLineChart(String title, String categoryAxisLabel,
String valueAxisLabel, CategoryDataset dataset, PlotOrientation orientation
, boolean legend, boolean tooltips, boolean urls)
```

Exemple de code

```
 private static CategoryDataset createCategoryDataset() {
        DefaultCategoryDataset dataset = new DefaultCateg
         dataset.setValue(10, "SELLER 1" , "Jan-Mar");
        dataset.setValue(8, "SELLER 1" , "Avr-Jui");
        dataset.setValue(12, "SELLER 1" , "Jui-Sep");
        dataset.setValue(20, "SELLER 1" , "Oct-Dec");

        dataset.setValue(4, "SELLER 2" , "Jan-Mar");
        dataset.setValue(8, "SELLER 2" , "Avr-Jui");
        dataset.setValue(12, "SELLER 2" , "Jui-Sep");
        dataset.setValue(24, "SELLER 2" , "Oct-Dec");

        dataset.setValue(30, "SELLER 3" , "Jan-Mar");
        dataset.setValue(4, "SELLER 3" , "Avr-Jui");
        dataset.setValue(12, "SELLER 3" , "Jui-Sep");
        dataset.setValue(1, "SELLER 3" , "Oct-Dec");
        return dataset;
    }
 private static JFreeChart createChart(CategoryDataset d) {
JFreeChart chart = ChartFactory.createLineChart("Evolution of sellers" , "Trimester",
"Cars sold", d, PlotOrientation.VERTICAL, true, true, false);
return chart;
    }
```



Création d'un Line chart avec un XYDataset

- Création d'un jeu de données (une collection de séries XY)

```
XYSeriesCollection dataset = new XYSeriesCollection();
```

- Créer une série

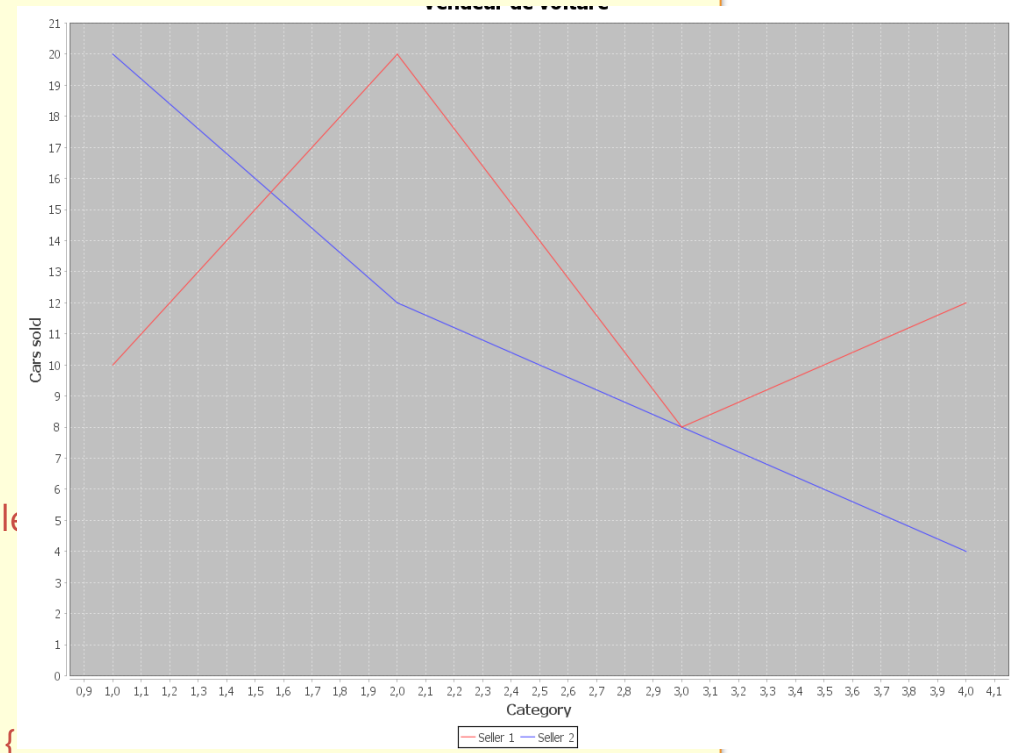
```
XYSeries serie = new XYSeries(String name);
```

- Ajout d'une valeur dans la série

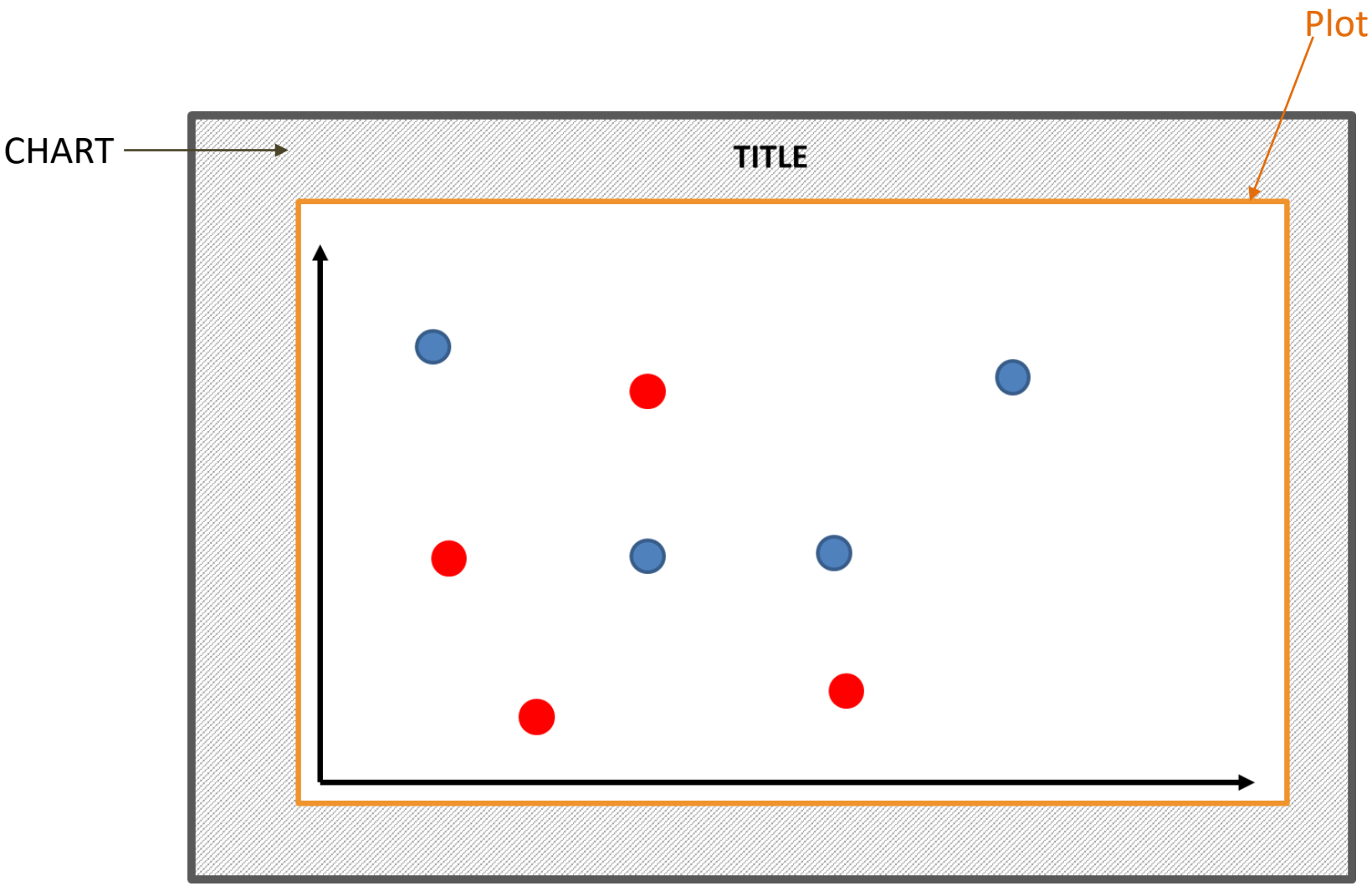
```
XYSeriesCollection .addSeries(double valueX, double valueY)
```

Exemple de code

```
private static XYSeriesCollection createXYDataset() {
XYSeries seller1 = new XYSeries("Seller 1");
        seller1.add(1, 10);
        seller1.add(2, 20);
        seller1.add(3, 8);
        seller1.add(4, 12);

        XYSeries seller2 = new XYSeries("Seller 2");
        seller2.add(1, 20);
        seller2.add(2, 12);
        seller2.add(3, 8);
        seller2.add(4, 4);

        XYSeriesCollection dataset = new XYSeriesColle
        dataset.addSeries(seller1);
        dataset.addSeries(seller2);
        return dataset; }

 private static JFreeChart createChart(XYDataset d) {
JFreeChart chart = ChartFactory.createLineChart("Evolution of sellers" , "Trimester",
"Cars sold", d, PlotOrientation.VERTICAL, true, true, false);
return chart;
    }
```



Différence entre Chart et Plot



Modifier l'apparence d'un graphe

- Sur l'objet Chart les modifications se font sur
  - Les bordures du graphe
  - Le titre du graphe
  - Le fond

- Méthodes sur les bordure

```
setBorderVisible() || setBorderPaint() || setBorderStroke()
```

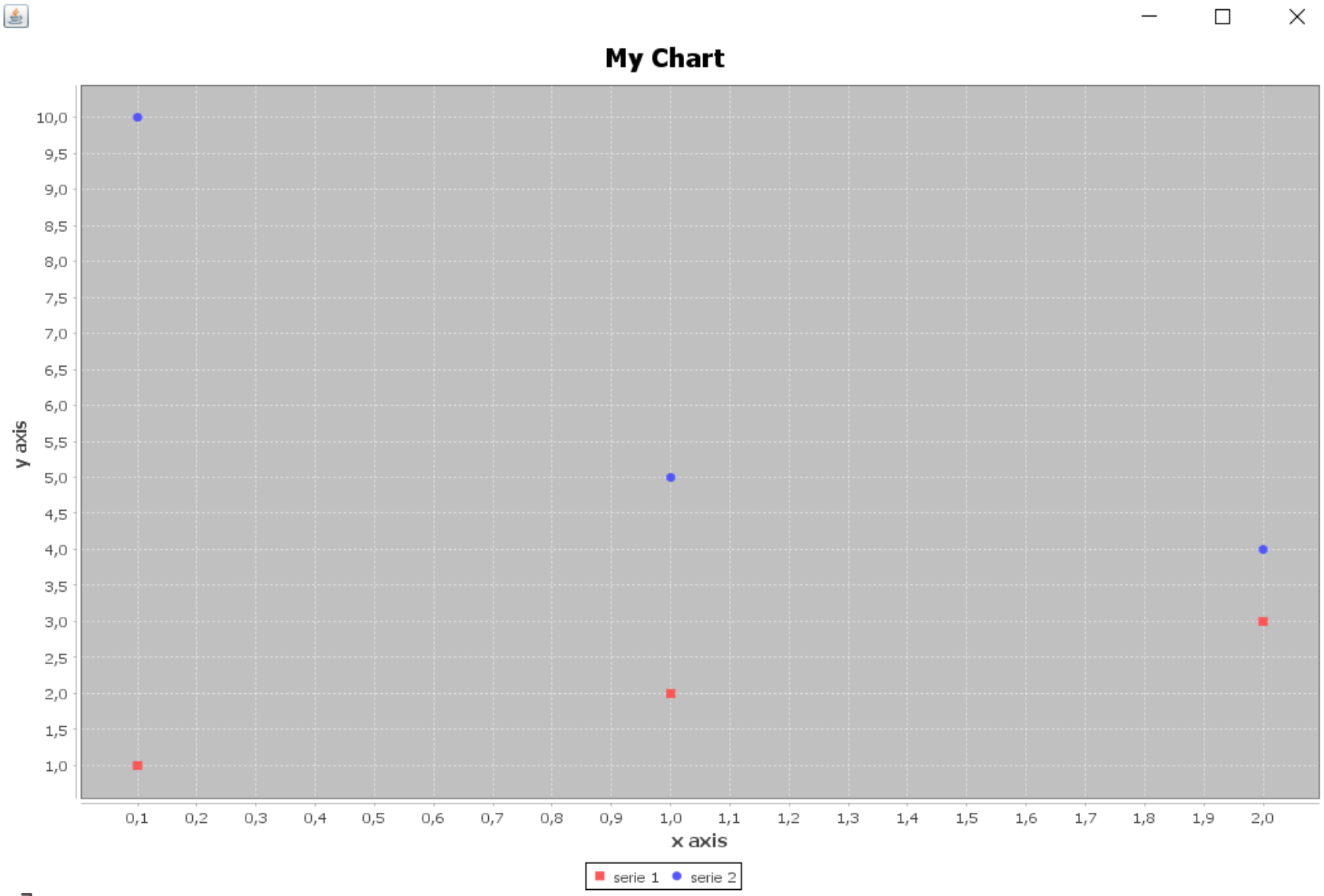
- Récupérer et instancier titres et sous-titres

```
TextTitle getTitle()              || chart.SetTitle(TextTitle t)
Title getSubtitle(int indice) ||addSubtitle(TextTitle t)
```

- Modifier les TextTitle

```
setText(String s) ||.setFont(Font f) || setBackgroundPaint(Paint p)
setTextAlignement(HorizontalAlignement a)
```

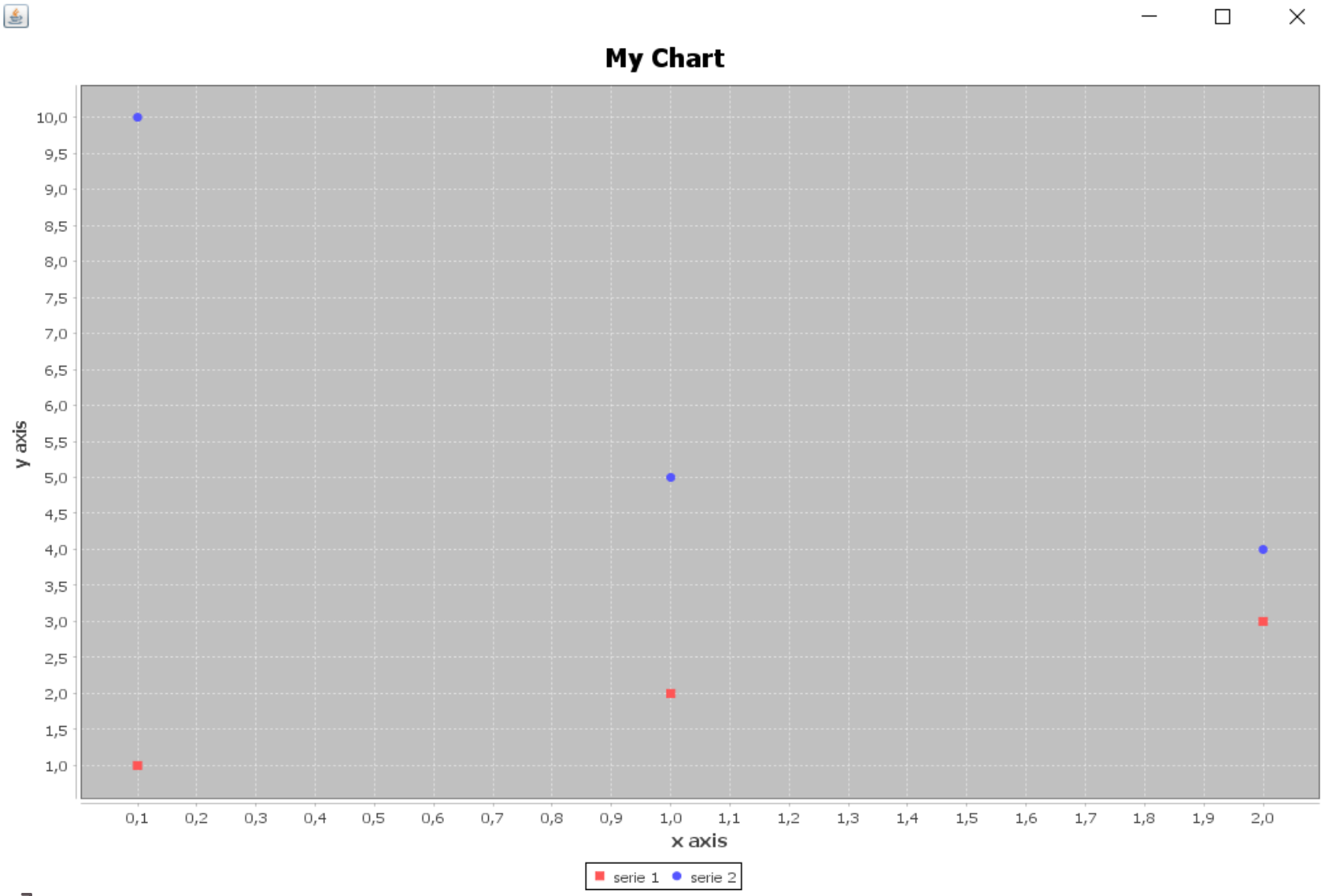
Code de base


My Chart
y axis
10,0
9,5
9,0
8,5
8,0
7,5
7,0
6,5
6,0
5,5
5,0
4,5
4,0
3,5
3,0
2,5
2,0
1,5
1,0
0,1
0,2
0,3
0,4
0,5
0,6
0,7
0,8
0,9
1,0
1,1
1,2
1,3
1,4
1,5
1,6
1,7
1,8
1,9
2,0
x axis
serie 1
serie 2

Code de base

```
chart.setBackgroundPaint(new Color(125,125,0));
Font f = new Font("Verdana", Font.BOLD, 18);
TextTitle title =new TextTitle("My new Title", f);
title.setPaint(new Color(255,255,255));
chart.setTitle(title);
```

Code de base


My Chart
y axis
10,0
9,5
9,0
8,5
8,0
7,5
7,0
6,5
6,0
5,5
5,0
4,5
4,0
3,5
3,0
2,5
2,0
1,5
1,0
0,1
0,2
0,3
0,4
0,5
0,6
0,7
0,8
0,9
1,0
1,1
1,2
1,3
1,4
1,5
1,6
1,7
1,8
1,9
2,0
x axis
serie 1
serie 2

Modifier l'apparence d'un graphe v2

- Récupérer un plot (exemple avec un XYDataset)

```
XYPlot getXYPlot( )
```

- Modifier la couleur de fond

```
Plot.setBackgroundPaint(Paint p)
```

- Modifier l'orientation

```
Plot.setOrientation (PlotOrientation p)
```

- Modifier l'échelle

```
(NumberAxis) plot.getDomain(Axis)
```

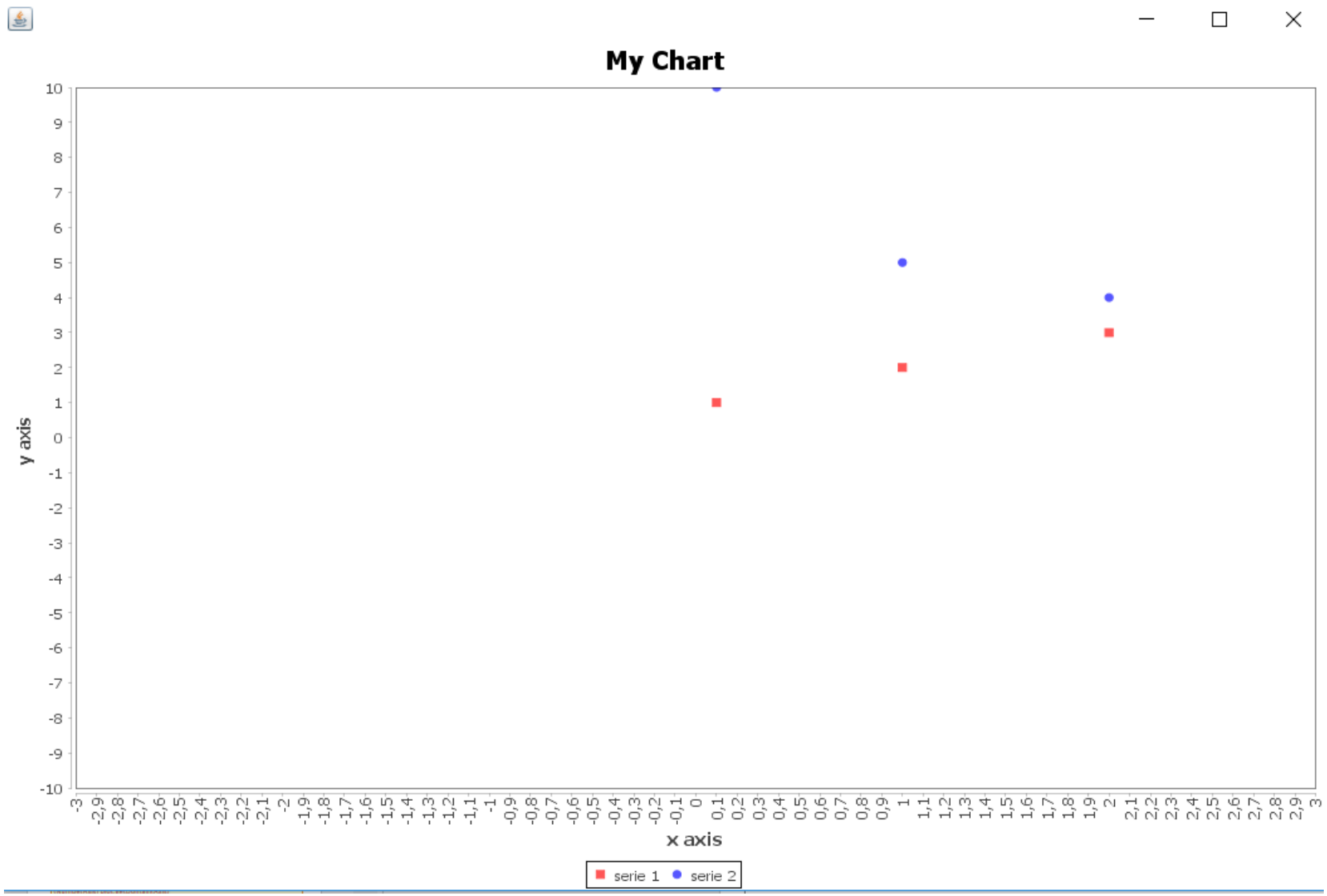
```
Domain.setRange(minvalue, maxvalue)
```

```
Domain.setTickUnit(NumberTickUnit unit)
```

Code de base

```
XYPlot plot = chart.getXYPlot();
plot.setBackgroundPaint(Color.WHITE);
plot.setOrientation(PlotOrientation.VERTICAL);
NumberAxis domain = (NumberAxis) plot.getDomainAxis();
 domain.setRange(-3.00, 3.00);
 domain.setTickUnit(new NumberTickUnit(0.1));
 domain.setVerticalTickLabels(true);
 NumberAxis range = (NumberAxis) plot.getRangeAxis();
 range.setRange(-10.0, +10.0);
 range.setTickUnit(new NumberTickUnit(1));
```

Code modifié



Modifier le rendu des données

- Créer un « renderer » pour les traits entre les donnes ou la forme

```
XYLineAndShapeRenderer renderer = new XYLineAndShapeRenderer( );
XYShapeRenderer renderer = new XYShapeRenderer( );
XYLineRenderer renderer = new XYLineRenderer( );
```

- Modifier la couleur ou l'epaisseur de la ligne

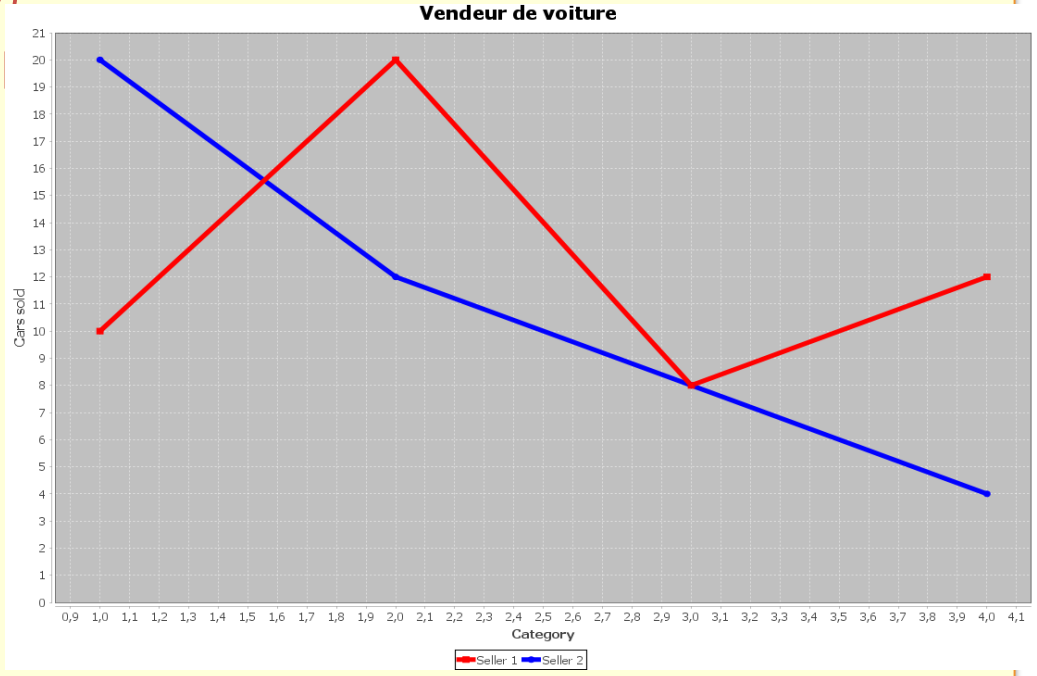
```
setSeriesPaint(int series, Paint paint) setSeriesStroke(int series, Stroke stroke)
```

- Modifier la forme

```
renderer.setSeriesShape(int series, Shape shape)
```

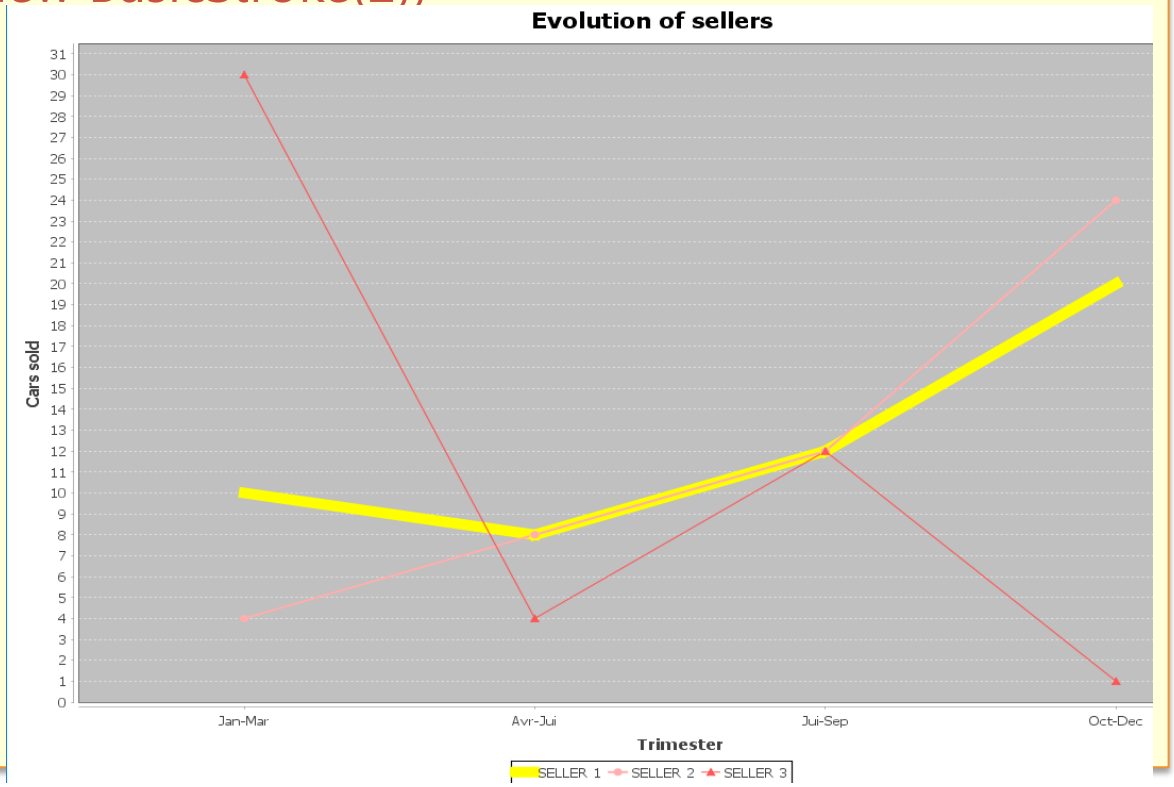
Exemple de code

```
XYPlot plot = chart.getXYPlot();

XYLineAndShapeRenderer renderer =
new XYLineAndShapeRenderer();
//COLOR
renderer.setSeriesPaint(0,Color.RED);
renderer.setSeriesPaint(1,Color.BLU
//EPAISSEUR DES LIGNES
B = new BasicStroke(5.0f);
renderer.setSeriesStroke(0, B);
renderer.setSeriesStroke(1, B);
plot.setRenderer(renderer);
```



Exemple de code avec un LineGraph

```
LineAndShapeRenderer renderer = new LineAndShapeRenderer();
renderer.setSeriesPaint(0, Color.YELLOW);
renderer.setSeriesPaint(1, Color.PINK);
renderer.setSeriesStroke(0, new BasicStroke(10));
renderer.setSeriesStroke(1, new BasicStroke(2))
 plot.setRenderer(renderer);
```



ChartUtilities

- Classe contenant une multitude d'utilitaires dont les fonctions d'export de graphes dans d'autres formats (PNG,JPEG)

- Constructeur

```
ChartUtilities()
```

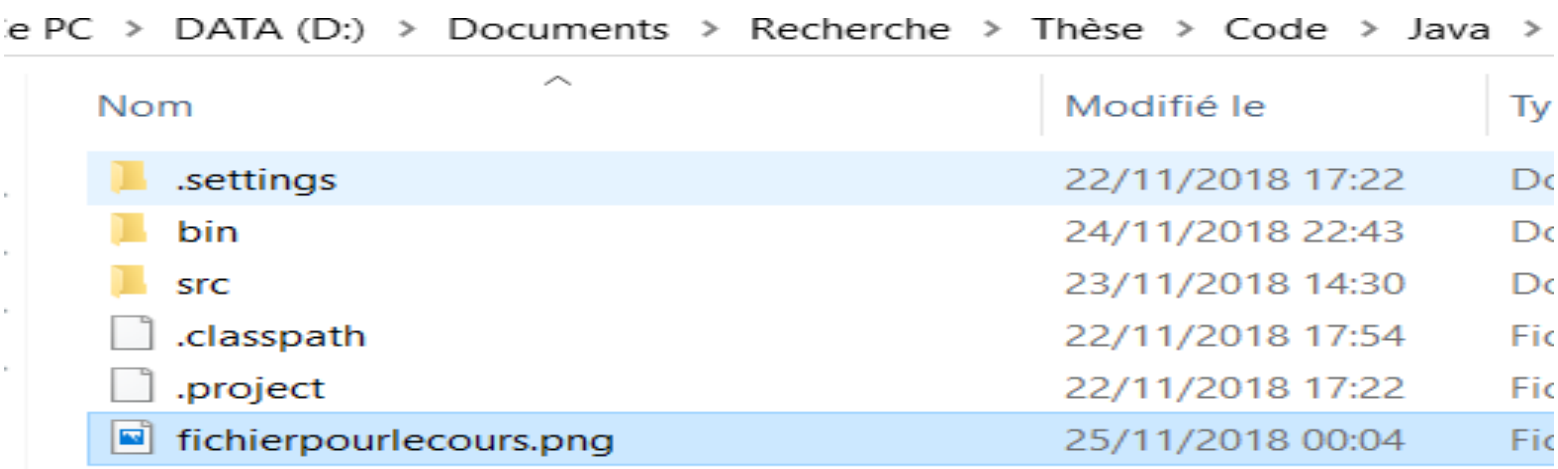
- Exporter les graphes

```
saveChartAsPNG(java.io.File file, JfreeChart chart, int width, int height)
```

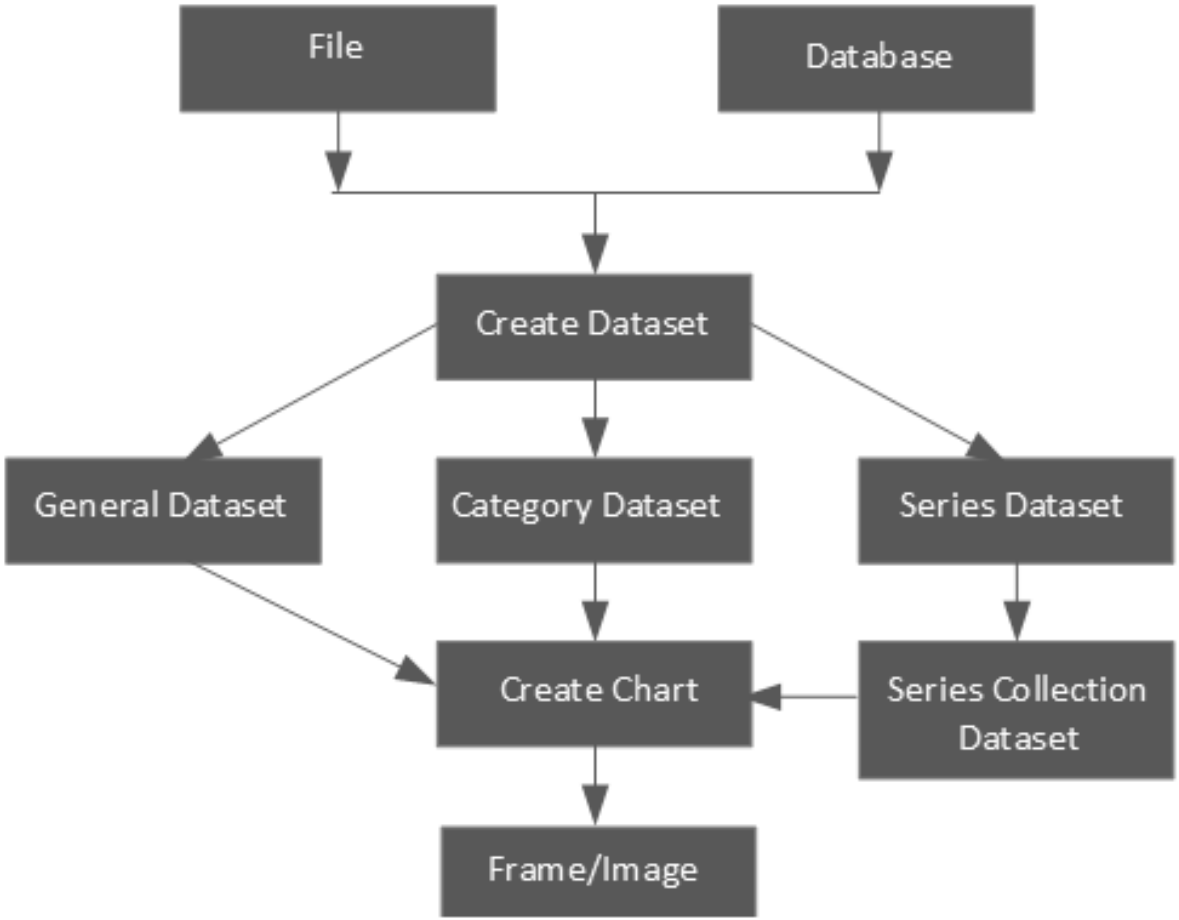
```
saveChartAsJPEG(java.io.File file, JfreeChart chart, int width, int height)
```

Exemple de code avec un LineGraph

```
JFreeChart chart = createLineChart(createCategoryDataset());
File f = new File(".\\fichierpourlecours.png");
ChartUtilities.saveChartAsPNG(f, chart, 1920, 1080);
```



Architecture



Créer graphe à partir d'une base de données v1

- Rappelez-vous… 3 étapes pour récupérer les valeurs d'une base de données
  - Connexion à la base
  - Requête sur la base
  - Traitement de la requête

- Initialisation du jeu de données voulu

```
DefaultPieDataset dataset = new DefaultPieDataset( );
```

```
DefaultCategoryDataset dataset = new DefaultCategoryDataset( );
```

- Dans le traitement de la requête

```
dataset.setValue( resultSet.getString("NOM_COLONNE"), … , …)
```

JDBCCategoryDataset

- Permet de garder en cache les données provenant directement de la base de données.

- Constructeur à partir d'une connexion et créeant dataset vide

```
JDBCCategoryDataset(Connection connection)
```

- Constructeur peuplant un dataset

```
JDBCCategoryDataset(Connection connection, String query)
```

- Constructeur créant une connexion

```
JDBCCategoryDataset(String url, String driverName,
 String user, String passwd)
```

- Méthode permettant de rafraichir le dataset à partir de la base

```
executeQuery(String query)
```

JDBCCategoryDataset exemple

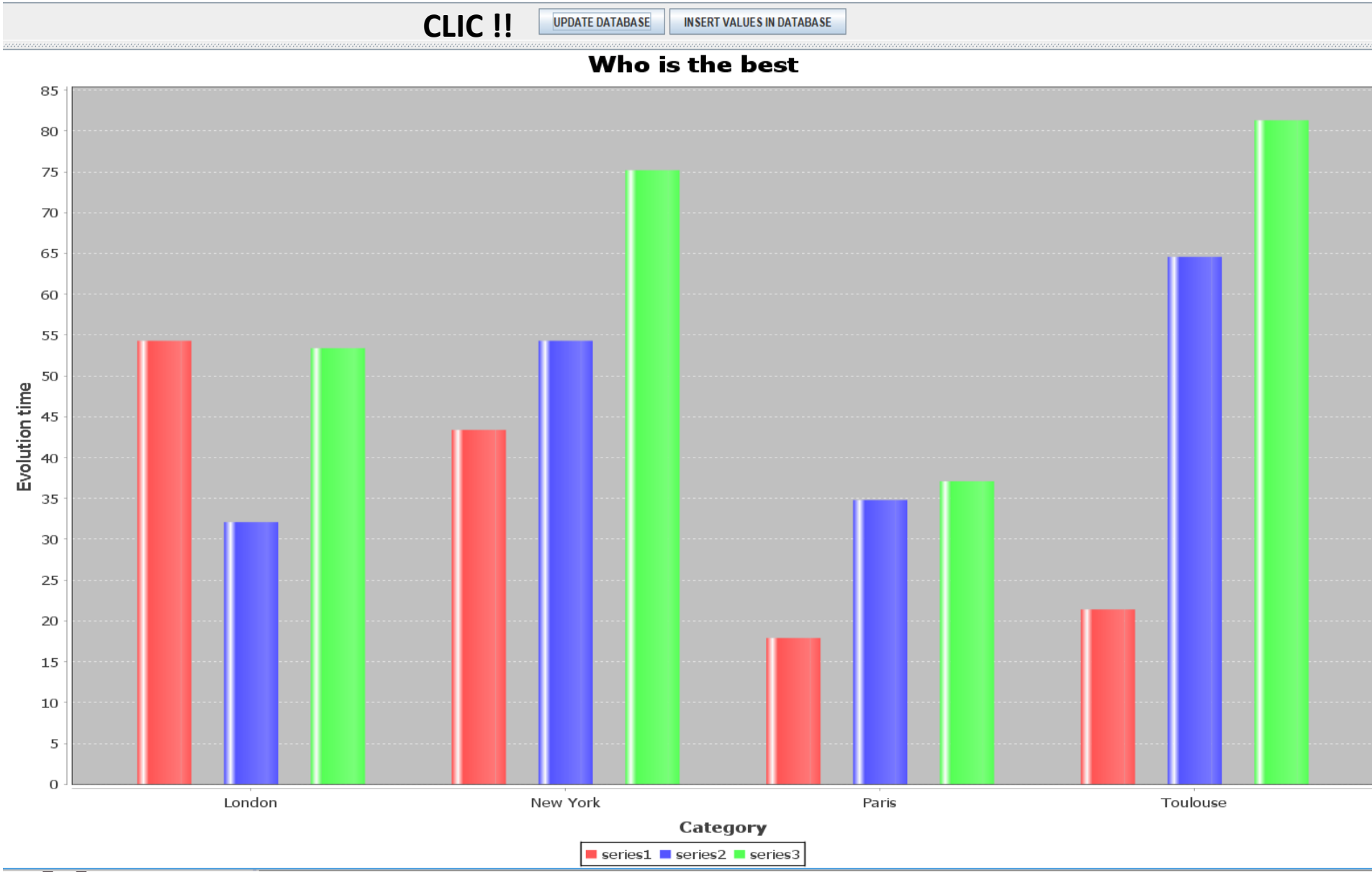
```
Class.forName("com.mysql.jdbc.Driver");
JFreeChart chart = null;
con = DriverManager.getConnection( "jdbc:mysql://localhost/jfreechartdb", "root", "");
Statement stmt = con.createStatement();
ResultSet rst = stmt.executeQuery("SELECT * FROM jfreechartdb.categorydata1");
String query = "SELECT * FROM jfreechartdb.categorydata1";
 chart = createBarChart(new JDBCCategoryDataset(con,query));
JFrame f = new JFrame();
f.add(new ChartPanel(chart));
```

Création de deux boutons pour rendre le graphe dynamique

```
JPanel jp1 = new JPanel();
jp1.setLayout(new FlowLayout());
query = "SELECT * FROM jfreechartdb.categorydata1";
JButton b1 = new JButton("UPDATE DATABASE");
b1.addActionListener(new ActionListener() {
   @Override
   public void actionPerformed(ActionEvent e) {
       dts.executeQuery(query);
    });
JButton b2 = new JButton("INSERT VALUES IN DATABASE");
b2.addActionListener(new ActionListener() {
     public void actionPerformed(ActionEvent e) {
       Statement s = con.createStatement();
    s.executeUpdate("INSERT INTO categorydata1 " +
"VALUES (\"Toulouse\",   21.4 , 64.6, 81.3)");
            });
```

Rendre le graphe dynamique

Rendre le graphe dynamique



JDBCPieDataset & JDBCXYDataset

- Ce sont les mêmes fonctions que pour la classe JDBCCategoryDataset

- 3 constructeurs

```
JDBCPieDataset ou JDBCXYDataset (Connection connection)
```

```
JDBCPieDataset ou JDBCXYDataset (Connection connection, String query)
```

```
JDBCPieDataset ou JDBCXYDataset  (String url, String driverName,
 String user, String passwd)
```

- Une méthode pour exécuter les requêtes

```
executeQuery(String query)
```

Credits for icons & liens utiles

Icon made by freepik from www.flaticon.com

Base de données

- https://docs.oracle.com/javase/tutorial/jdbc/index.html

JFREECHART

- https://www.tutorialspoint.com/jfreechart/jfreechart_referenced_apis.htm
- http://www.jfree.org/jfreechart/api/javadoc/overview-summary.html
- http://www.cin.ufpe.br/~ags/jFreeChart/Object.Refinery.The.JFreeChart.Class.Library.Developer.Guide.v1.0.9.Jan.2008.pdf