

# Lecture6 JavaFX 数据可视化

---

## 1. 可视化

数据可视化要时刻考虑到受众，一个可视化的消费者大致有三种

- 开发者自己
- 行业专家
- 所有人

### 为自己可视化数据

首先是自己，这个无所不知的专家最有可能在分析或算法开发中快速迭代

你的要求是尽可能清晰和快速地查看数据，设置情节标题、轴标签、平滑、图例或日期格式等内容可能并不重要，因为您已经清楚地知道您正在查看的内容

从本质上说，我们经常绘制数据以获得数据全景的快速概述，而不考虑别人如何看待它

### 为行业专家可视化数据

数据可视化的第二个消费者是行业专家

解决了数据科学问题之后，你准备分析，需要填充好坐标轴的标签，标题，确保任何一系列的数据都用图例描述，并确保你所创建的图像能够独立地讲述一个故事

即使它的视觉效果并不出众，你的同事和同事可能不会在意外表本身，而是在意你想要传达的信息

事实上，如果可视化中清楚地包含图形小部件和效果，那么就更容易对工作的优点做出科学的评估。当然，这种格式对于归档数据也是必不可少的，一个月后，如果现在不标记，你将不记得这些坐标轴是什么

### 为所有人可视化数据

第三类可视化消费者是其他人，这是需要创意和艺术的时候，因为仔细选择颜色和风格可以使数据看起来很棒

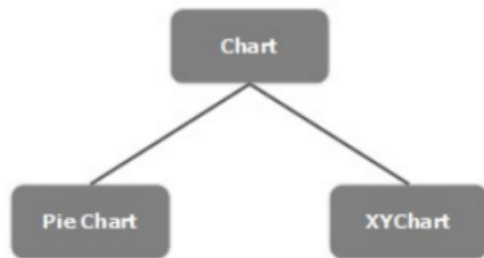
但是，要注意的是，你将花费大量的时间和精力来为这个级别的用户准备图像

使用 JavaFX 的另一个优点是通过鼠标选项允许交互性，这使得能够构建一个图形化应用程序，类似于许多熟悉的基于 web 的仪表板

## JavaFX - 图表

一般来说，图表是数据的图形表示，表示数据的图表有很多种，如柱状图、饼图、折线图、散点图等

JavaFX支持各种 `Pie Charts` 和 `XY Charts`



在 `XY Charts` 上表示的图表包括

- `AreaChart`
- `BarChart`
- `BubbleChart`
- `LineChart`
- `ScatterChart`
- `StackedAreaChart`
- `StackedBarChart`

## 2. 创建一个图表

为了创建一个图表，你需要

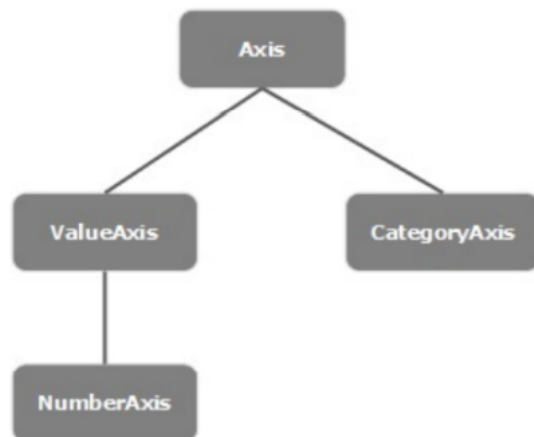
- 定义图表的坐标轴
- 实例化相应的类
- 准备并传递数据给图表

### 定义图表的坐标轴

通常，图表的轴可以用被表示成

- 数字：例如人口数量、年龄
- 种类：例如每周的星期，国家

在JavaFX中，`Axis` 是表示X或Y轴的抽象类，它有两个子类来定义每一种类型的轴，分别是 `CategoryAxis` 和 `NumberAxis`



## CategoryAxis

通过实例化这个类，可以定义(创建)一个 X 或 Y 轴，其中每个值代表一个类别

```
1 CategoryAxis xAxis = new CategoryAxis();
```

对于该轴，需要将类别列表和标签设置到该轴，如下所示

```
1 // 设置分类
2 xAxis.setCategories(FXCollections.
   <String>observableArrayList(Arrays.asList("name1", "name2", ...)));
3 // 设置 Label
4 xAxis.setLabel("name of the axis");
```

## NumberAxis

通过实例化这个类，可以定义(创建)一个 X 或 Y 轴，沿着这个轴，每个值代表一个数值

可以使用任何数字类型，`Long`、`Double`、`BigDecimal` 等

```
1 NumberAxis yAxis = new NumberAxis();
2 yAxis.setLabel("name of the axis")
```

## 实例化相应的类

要创建一个图表，实例化它各自的类

- 例如，如果想创建一个折线图，需要按如下方式实例化名为 `LineChart` 的类

```
1 LineChart linechart = new Linechart(xAxis, yAxis);
```

正如在上面的代码中观察到的，在实例化时，需要传递两个分别表示图表的 X 轴和 Y 轴的对象

## 准备并传递数据给图表

所有的 XY 图表都是沿着 XY 平面表示的，要在图表中绘制一组点，我们需要指定一系列XY坐标

`XYChart.Series` 类是的 `javafx.scene.chart` 包是一个类，使用它，你可以向图表发送数据，这个类保存一个命名为 `series` 的可观察对象列表，可以使用 `XYChart` 的 `getData()` 方法获得这个列表，如下所示

```
1 ObservableList list = series.getData();
```

其中 `series` 是 `XYChart.Series` 类的一个实例对象，你可以将数据添加到这个列表中，使用

```
1 list.add(new XYChart.Data(x-axis data, y-axis data));
```

这两行代码可以被合并在一起

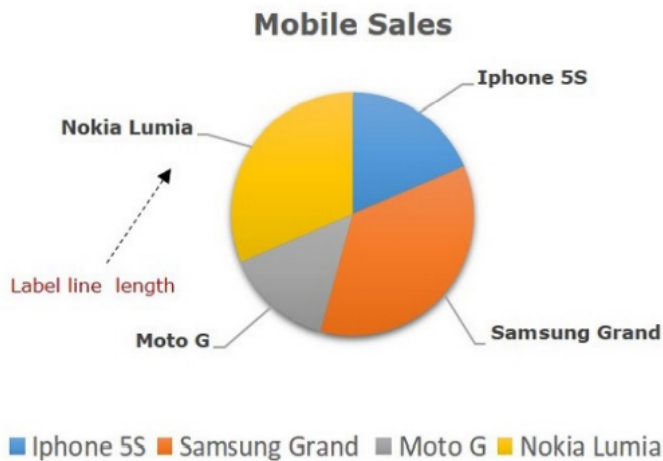
```
1 series.getData().add(new XYChart.Data(x-axis data, y-axis data));
```

### 3. 图表的创建与展示（以饼图为例）

#### 饼图 Pie Chart

饼图是将值表示为具有不同颜色的圆切片，这些切片被标记，并且每个切片对应的值在图表中表示

下面是一个饼图，以一个实例描述了不同公司的移动销售情况



在 JavaFX 中，饼图由 `PieChart` 表示，这个类属于 `javafx.scene.chart`

通过实例化这个类，你可以在 JavaFX 中创建一个饼图的 `node`

这个类由 5 个属性

- `clockwise`：布尔表达，将该操作符设置为 `true` 后，饼图中的数据切片将从饼图的起始角度开始顺时针排列
- `data`：一个 `ObservableList` 对象，其中装载了饼图需要的数据
- `labelLineLength`：一个整型变量，表示连接标签和饼图切片的线的长度
- `labelsVisible`：一个布尔运算符，将该操作符设置为 `true` 时，将绘制饼图的标签，默认情况下，该操作符被设置为 `true`
- `startAngle`：这是一个双类型操作符，它表示第一个饼图切片开始的角度

## 准备 ObservableList 对象

为了获得一个饼图，提供一个 `ObservableList` 对象

```
1 ObservableList<PieChart.Data> pieChartData = FXCollections.observableArrayList(  
2     new PieChart.Data("Iphone 5S", 13),  
3     new PieChart.Data("Samsung Grand", 25),  
4     new PieChart.Data("MOTO G", 10),  
5     new PieChart.Data("Nokia Lumia", 22));
```

## 创造 PieChart 对象

将 `ObservableList` 对象传入 `PieChart` 的构造器中

```
1 // 创建一个饼图  
2 PieChart pieChart = new PieChart(pieChartData);
```

或者使用 `setData()` 方法

```
1 pieChart.setData(pieChartData);
```

## 给 PieChart 设置属性

```
1 pieChart.setTitle("Mobile Scales"); // 设置标题  
2 pieChart.setClockwise(true); // 设置切片顺序  
3 pieChart.setLabelLineLength(50); // 设置标签和饼图切片的线的长度  
4 pieChart.setLabelVisible(true); // 设置标签可见  
5 pieChart.setStartAngle(180); // 设置起始的角度
```

## 创建一个 Group 对象

将在上一步中创建的 `PieChart` 对象作为参数（一个 `node`）传递给 `Group` 类的构造函数，将其添加到组中

```
1 Group root = new Group(pieChart);
```

## 创建 Scene、Stage 并展示

```
1 Scene scene = new Scene(group, 600, 300);
2 primaryStage.setScene(scene);
3 primaryStage.setTitle("Sample Application");
4 primaryStage.show();
```

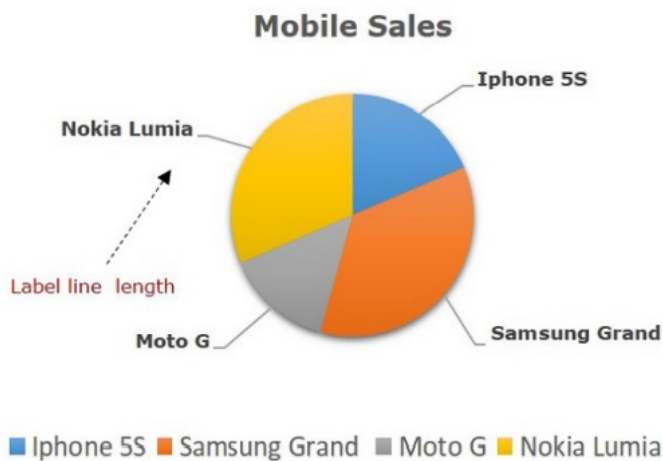
展示方法与 Lecture5 的写法类似

## 4. JavaFX 图表

### 饼图 Pie Chart

饼图是将值表示为具有不同颜色的圆切片，这些切片被标记，并且每个切片对应的值在图表中表示

下面是一个饼图，以一个实例描述了不同公司的移动销售情况



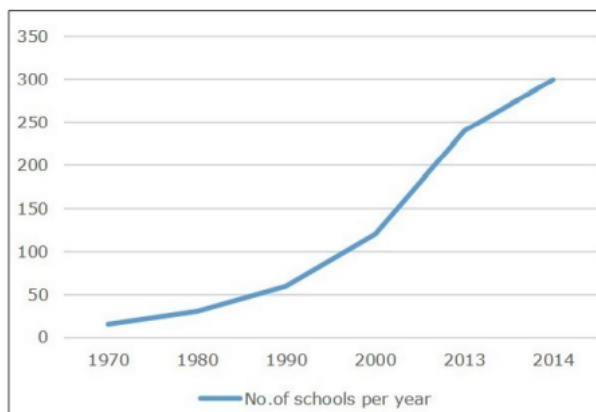
在 JavaFX 中，饼图由 `PieChart` 表示，这个类属于 `javafx.scene.chart`

通过实例化这个类，你可以在 JavaFX 中创建一个饼图的 `node`

### 折线图 Line Chart

折线图或线形图以一系列由直线段连接的数据点（标记）的形式显示信息

折线图显示了数据如何在相同的时间频率变化



在 JavaFX 中，折线图由 `LineChart` 表示，这个类属于 `javafx.scene.chart`

通过实例化这个类，你可以在 JavaFX 中创建一个折线图的 `node`

## 面积图 Area Chart