6. JAVAFX CHARTS

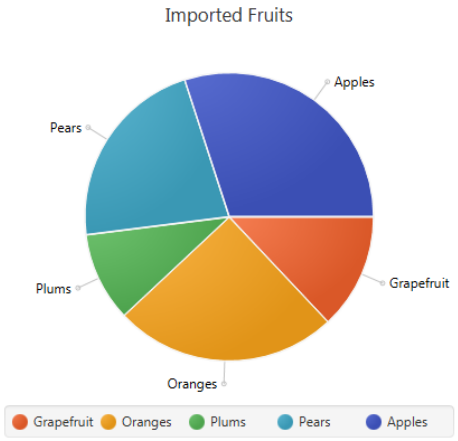
Interfaces Persona Computador

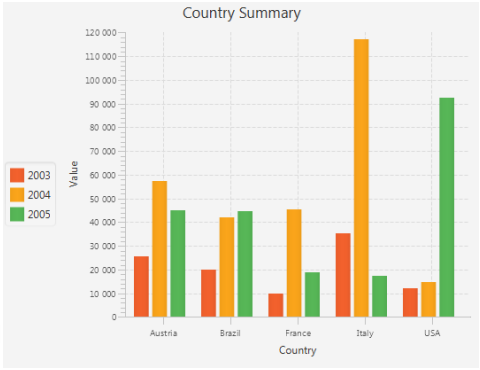
Depto. Sistemas Informáticos y Computación

UPV

Summary

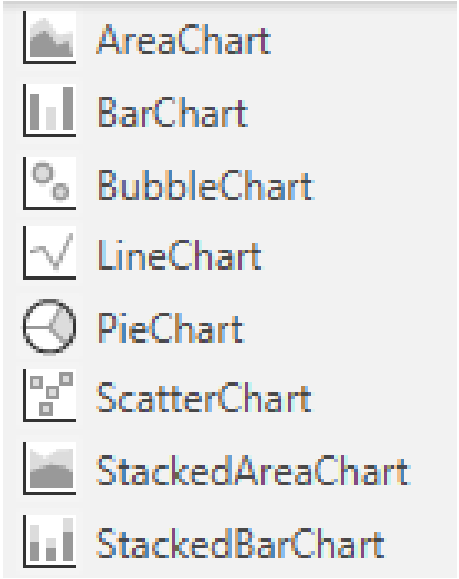
- Charts in JavaFX 8
  - Pie chart
  - Line chart
  - Area chart
  - Bubble chart
  - Scatter chart
  - Bar chart
  - Operations with charts
- Example
- Exercise





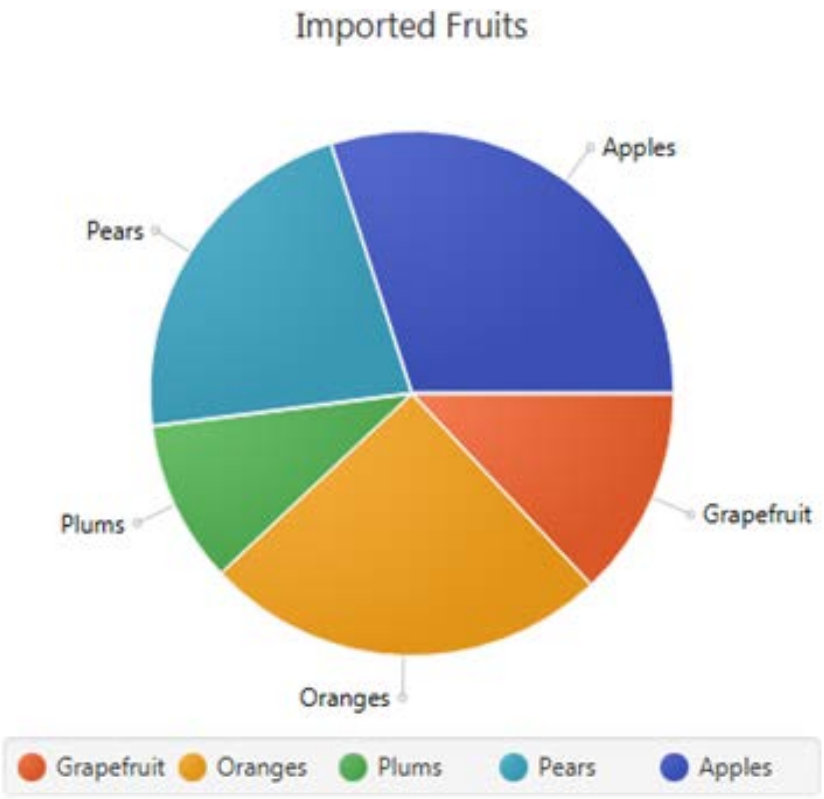
Introduction

- Charts in JavaFX can be implemented using code or using Scene Builder
- First, we will see how to implement them with code
- Then we will implement a chart using Scene Builder
- We recommend using Scene Builder
- All JavaFX charts display 2D data, except the pie chart (1D) and the bubble chart (3D)



Pie Chart

- Displays data in a form of a circle divided into wedges each representing a percentage corresponding to a value



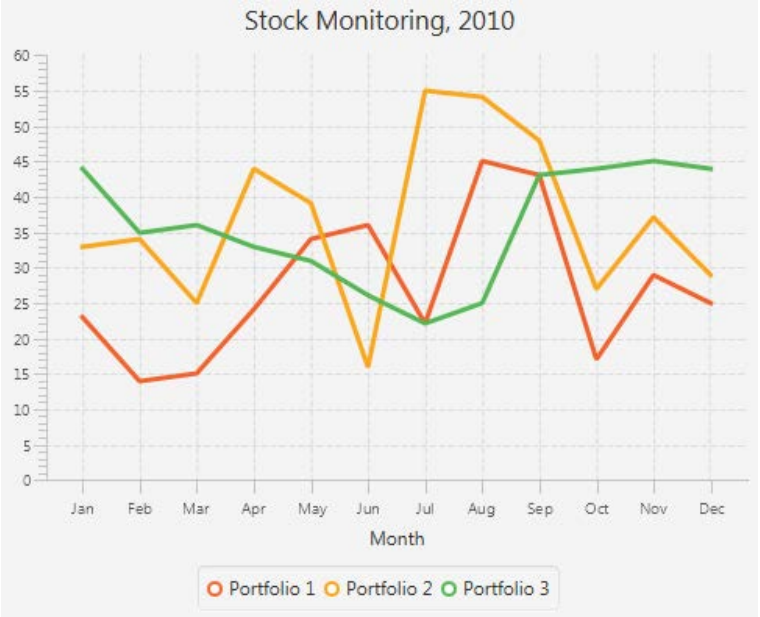
Pie Chart

- To display a chart:
  - Create a *PieChart*
  - Create an observable list to store the chart data
  - Data are of type *PieChart.Data*, each containing a *String* and the numeric value associated to that *String*

```
PieChart chart = new PieChart();
ObservableList<PieChart.Data> pieChartData =
       FXCollections.observableArrayList(
       new PieChart.Data("Grapefruit", 13),
       new PieChart.Data("Oranges", 25),
       new PieChart.Data("Plums", 10),
       new PieChart.Data("Pears", 22),
       new PieChart.Data("Apples", 30));
chart.setData(pieChartData);
chart.setTitle("Imported Fruits");
```

Line Chart

- Displays a set of 2D points connected by straight lines
- It is made of two axes, the plot of data points, the legend and an optional title
- It can have one or more data series

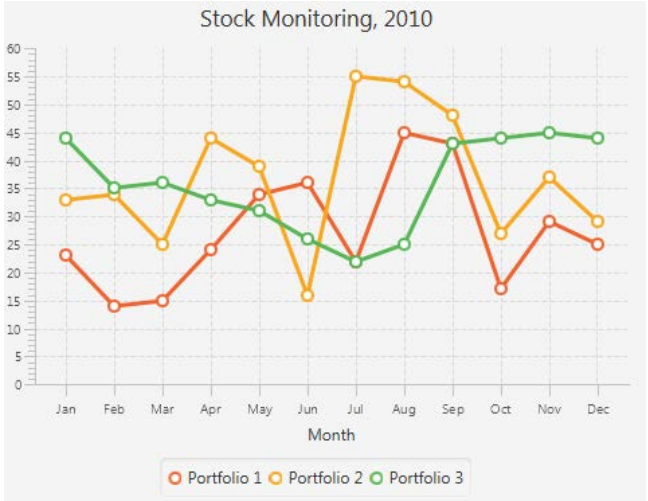


Line Chart

- With a single data series:

```
CategoryAxis xAxis = new CategoryAxis();
NumberAxis yAxis = new NumberAxis();
xAxis.setLabel("Month");

LineChart<String,Number> lineChart =
        new LineChart<>(xAxis,yAxis);

lineChart.setTitle("Stock Monitoring, 2010");

XYChart.Series series = new XYChart.Series();
series.setName("My portfolio");

series.getData().add(new XYChart.Data("Jan", 23));
series.getData().add(new XYChart.Data("Feb", 14));
...
series.getData().add(new XYChart.Data("Nov", 29));
series.getData().add(new XYChart.Data("Dec", 25));

Scene scene  = new Scene(lineChart,800,600);
lineChart.getData().add(series);
```

Line Chart



- With three data series:

```
XYChart.Series series1 = new XYChart.Series();
series1.setName("Portfolio 1");
series1.getData().add(new XYChart.Data("Jan", 23));
...
series1.getData().add(new XYChart.Data("Dec", 25));

XYChart.Series series2 = new XYChart.Series();
series2.setName("Portfolio 2");
series2.getData().add(new XYChart.Data("Jan", 33));
...
series2.getData().add(new XYChart.Data("Dec", 29));

XYChart.Series series3 = new XYChart.Series();
series3.setName("Portfolio 3");
series3.getData().add(new XYChart.Data("Jan", 44));
...
series3.getData().add(new XYChart.Data("Dec", 44));

Scene scene  = new Scene(lineChart,800,600);
lineChart.getData().addAll(series1, series2, series3);
```

Line Chart

- Options:
  - Move the X axis to the top of the chart

```
xAxis.setSide(Side.TOP);
```

  - Disable the symbols used for the data points

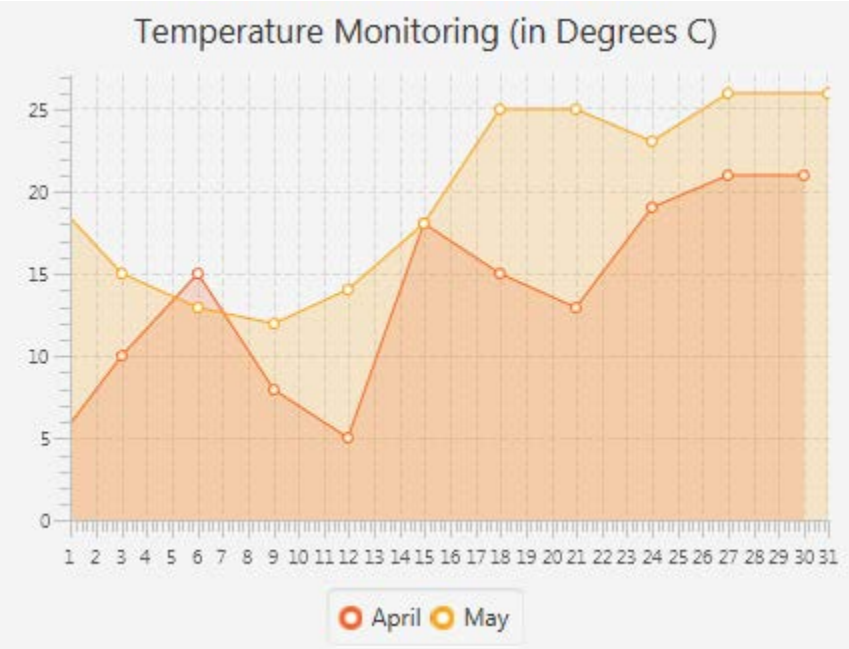
```
lineChart.setCreateSymbols(false);
```

  - Use numbers instead of strings in the X axis

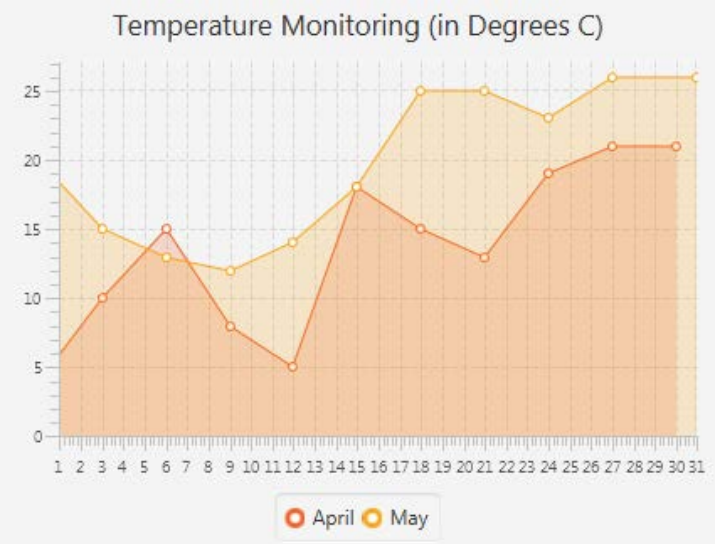
```
NumberAxis xAxis = new NumberAxis();
NumberAxis yAxis = new NumberAxis();
//creating the chart
LineChart<Number,Number> lineChart =
       new LineChart<>(xAxis,yAxis);

//defining a series
XYChart.Series series = new XYChart.Series();
//populating the series with data
series.getData().add(new XYChart.Data(1, 23));
...
```

Area Chart

- Similar to a line chart, it paints with color the area between the X axis and the lines
- Supports one or more data series



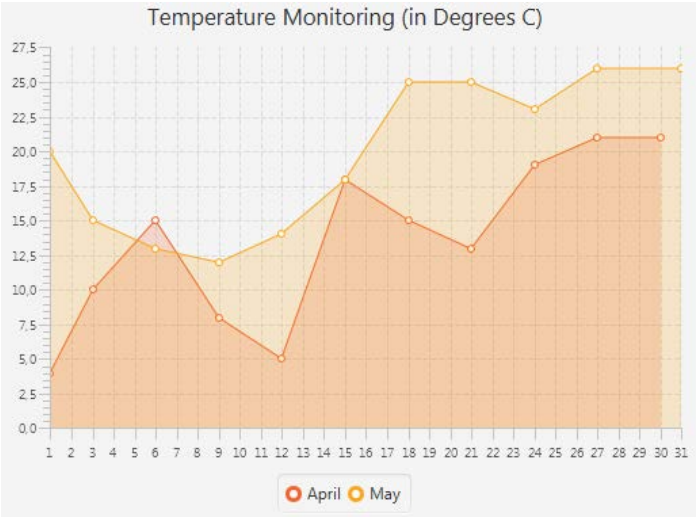
Area Chart



- Created like a line chart

```
stage.setTitle("Area Chart Sample");
NumberAxis xAxis = new NumberAxis(1, 31, 1);
NumberAxis yAxis = new NumberAxis();
AreaChart<Number,Number> ac =
    new AreaChart<>(xAxis,yAxis);
ac.setTitle("Temperature Monitoring (in Degrees C)");

XYChart.Series seriesApril= new XYChart.Series();
seriesApril.setName("April");
seriesApril.getData().add(new XYChart.Data(1, 4));
...
seriesApril.getData().add(new XYChart.Data(30, 21));

XYChart.Series seriesMay = new XYChart.Series();
seriesMay.setName("May");
seriesMay.getData().add(new XYChart.Data(1, 20));
...
seriesMay.getData().add(new XYChart.Data(31, 26));

ac.getData().addAll(seriesApril, seriesMay);
```

Area Chart



- Options
  - In the declaration of the X axis

```
NumberAxis xAxis = new NumberAxis(1, 31, 1);
```

    you can select the minimum(1), the maximum (31) and the distance between tick marks (1)
  - Also, you can do that explicitly:

```
xAxis.setLowerBound(1);
xAxis.setUpperBound(31);
xAxis.setTickUnit(1);
```

  - To remove the minor tick marks:

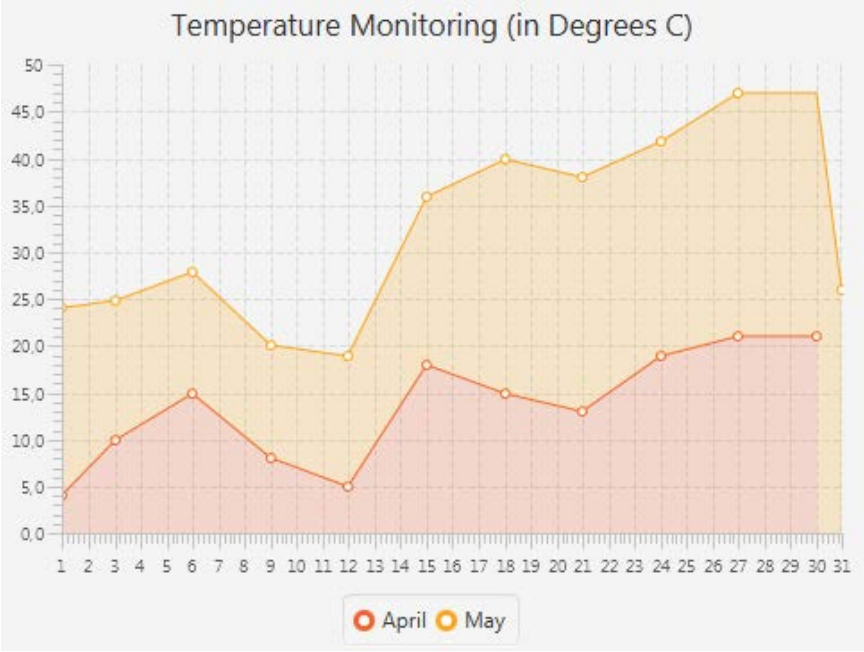
```
xAxis.setMinorTickCount(0);
```

  - To remove tick marks and labels, respectively:

```
xAxis.setTickMarkVisible(false);
xAxis.setTickLabelsVisible(false);
```

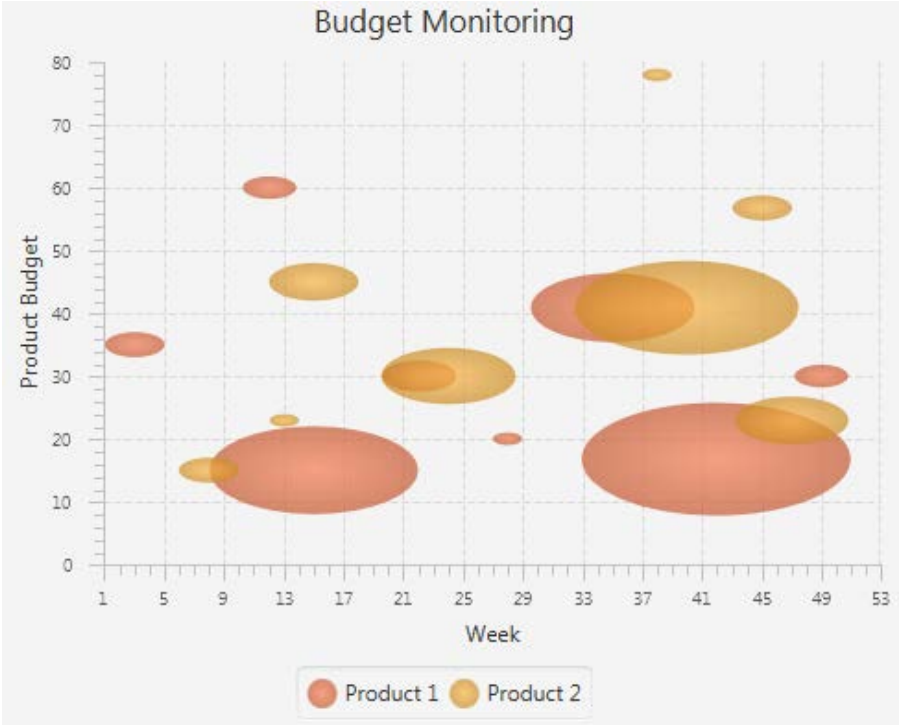
Stacked Area Chart

- It is an area chart where the second, third, etc. series are displayed showing the accumulated values of the previous series
- Use class *StackedAreaChart* instead of *AreaChart*



Bubble Chart

- It is an XY chart that displays bubbles for the points of the series
- The bubbles can have a different radii

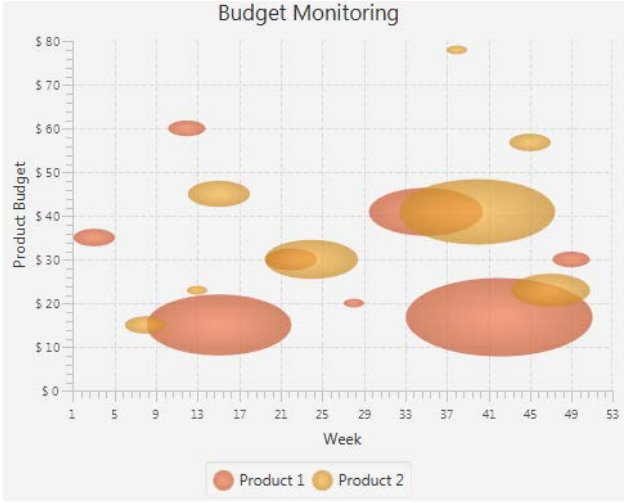


Bubble Chart

- The radius of each bubble is specified as an additional parameter in the *XYChart.Data()*
- The radius must be of type Number

```
NumberAxis xAxis = new NumberAxis(1, 53, 4);
NumberAxis yAxis = new NumberAxis(0, 80, 10);
BubbleChart<Number,Number> blc = new
            BubbleChart<>(xAxis,yAxis);

XYChart.Series series1 = new XYChart.Series();
series1.setName("Product 1");
series1.getData().add(new XYChart.Data(3, 35, 2));
...

XYChart.Series series2 = new XYChart.Series();
series2.setName("Product 2");
series2.getData().add(new XYChart.Data(8, 15, 2));
...

blc.getData().addAll(series1, series2);
```

Bubble Chart



- Options:
  - To format the labels of an axis

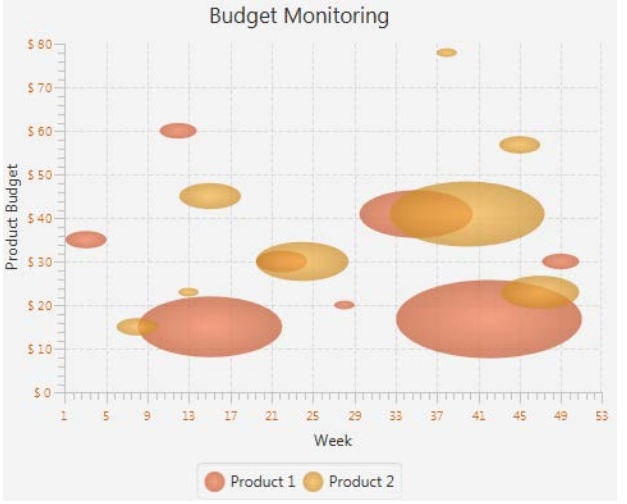
```
yAxis.setTickLabelFormatter(new
    NumberAxis.DefaultFormatter(yAxis,"$ ",null));
```

  - To remove the grid lines of a chart

```
blc.setHorizontalGridLinesVisible(false);
blc.setVerticalGridLinesVisible(false);
```

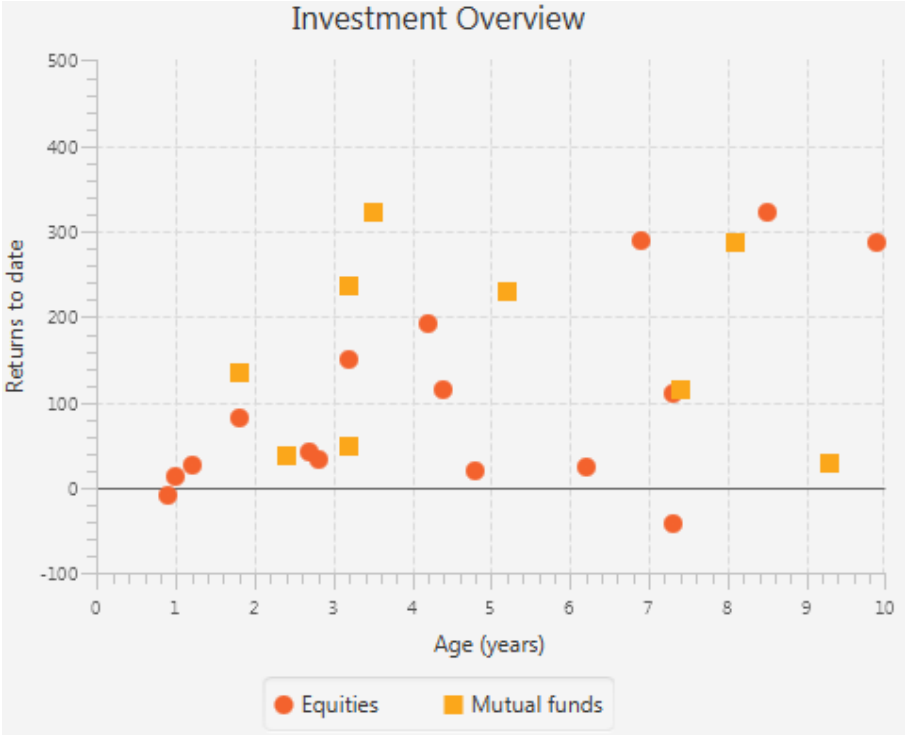
  - To change the color of the tick labels

```
xAxis.setTickLabelFill(Color.CHOCOLATE);
yAxis.setTickLabelFill(Color.CHOCOLATE);
```

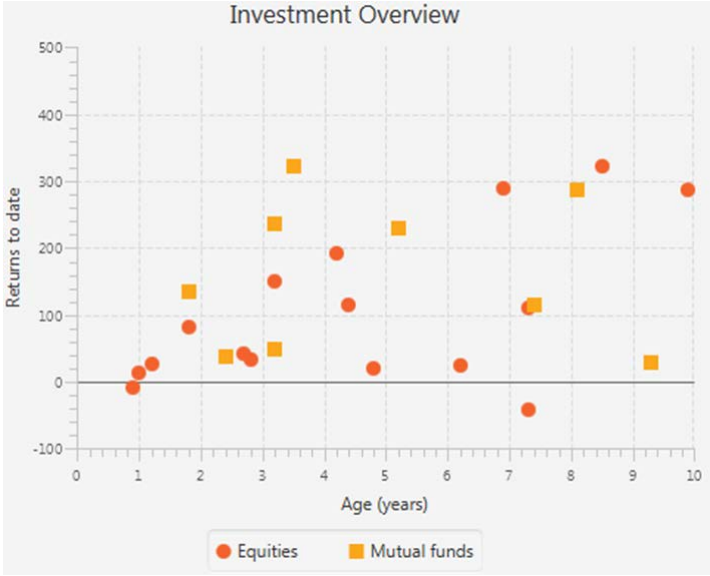


Scatter Chart

- Two-dimensional chart made of points given by pairs of XY values
- They are created like the other 2D charts



Scatter Chart

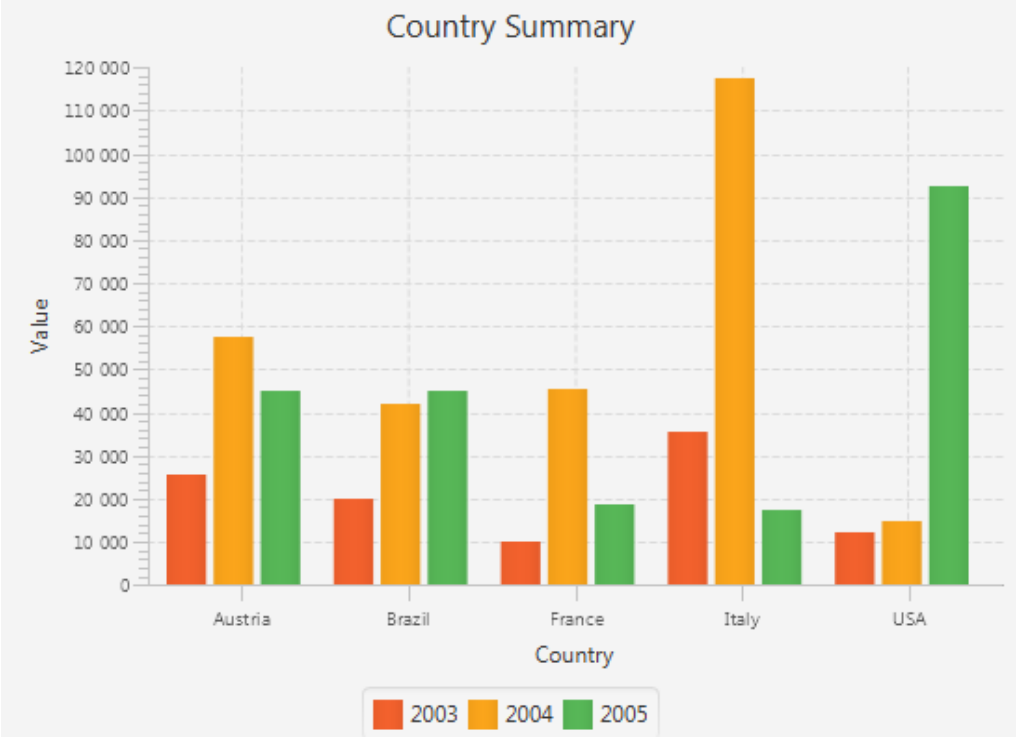


- Created like the other 2D charts:

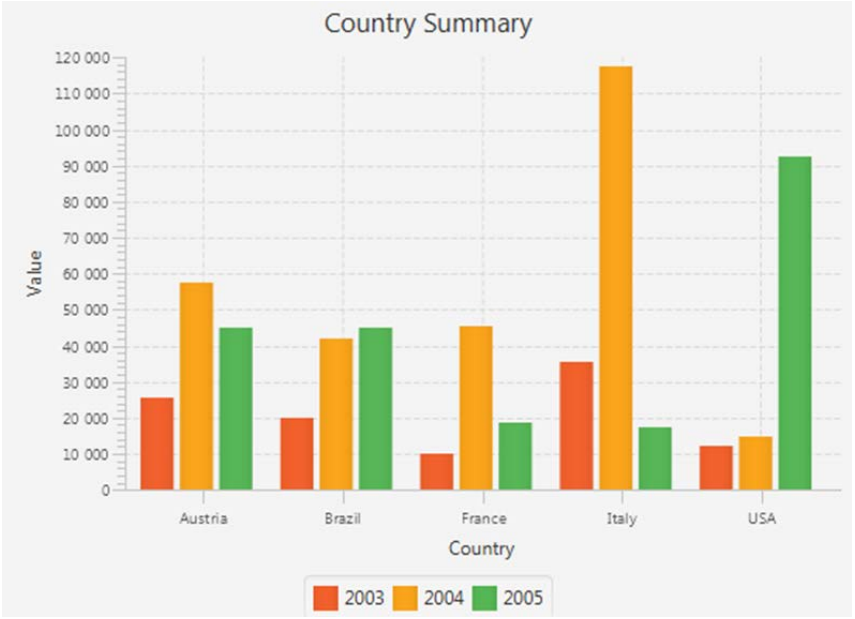
```
NumberAxis xAxis = new NumberAxis(0, 10, 1);
NumberAxis yAxis = new NumberAxis(-100, 500, 100);
ScatterChart<Number,Number> sc = new
            ScatterChart<>(xAxis,yAxis);
xAxis.setLabel("Age (years)");
yAxis.setLabel("Returns to date");
sc.setTitle("Investment Overview");

XYChart.Series series1 = new XYChart.Series();
series1.setName("Equities");
series1.getData().add(new XYChart.Data(4.2, 193.2));
...

XYChart.Series series2 = new XYChart.Series();
series2.setName("Mutual funds");
series2.getData().add(new XYChart.Data(5.2, 229.2));
...

sc.getData().addAll(series1, series2);
```

Bar Chart

- It is a two-dimensional chart where data is represented as bars
- They support one or more data series



Bar Chart



- Sample code

```
CategoryAxis xAxis = new CategoryAxis();
NumberAxis yAxis = new NumberAxis();
BarChart<String,Number> bc =
            new BarChart<>(xAxis,yAxis);
bc.setTitle("Country Summary");
xAxis.setLabel("Country");
yAxis.setLabel("Value");

XYChart.Series series1 = new XYChart.Series();
series1.setName("2003");
series1.getData().add(new XYChart.Data("Austria", 25601.34));
...
XYChart.Series series2 = new XYChart.Series();
series2.setName("2004");
series2.getData().add(new XYChart.Data("Austria", 57401.85));
...
XYChart.Series series3 = new XYChart.Series();
series3.setName("2005");
series3.getData().add(new XYChart.Data("Austria", 45000.65));
...
bc.getData().addAll(series1, series2, series3);
```

Bar Chart

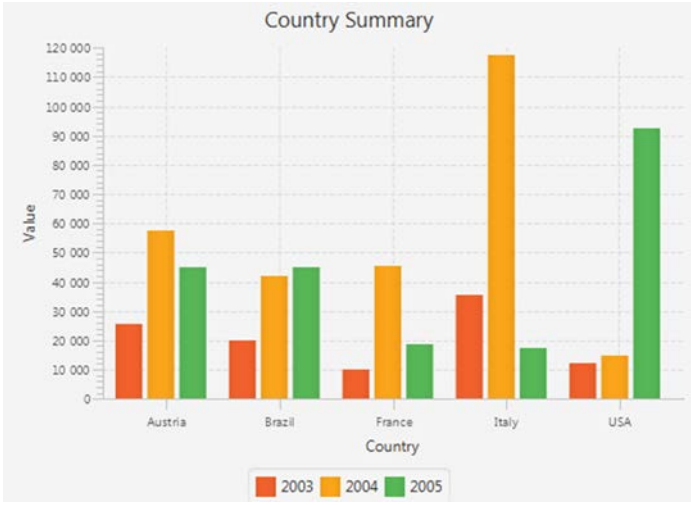
- Options:
  - Set the gap between bars

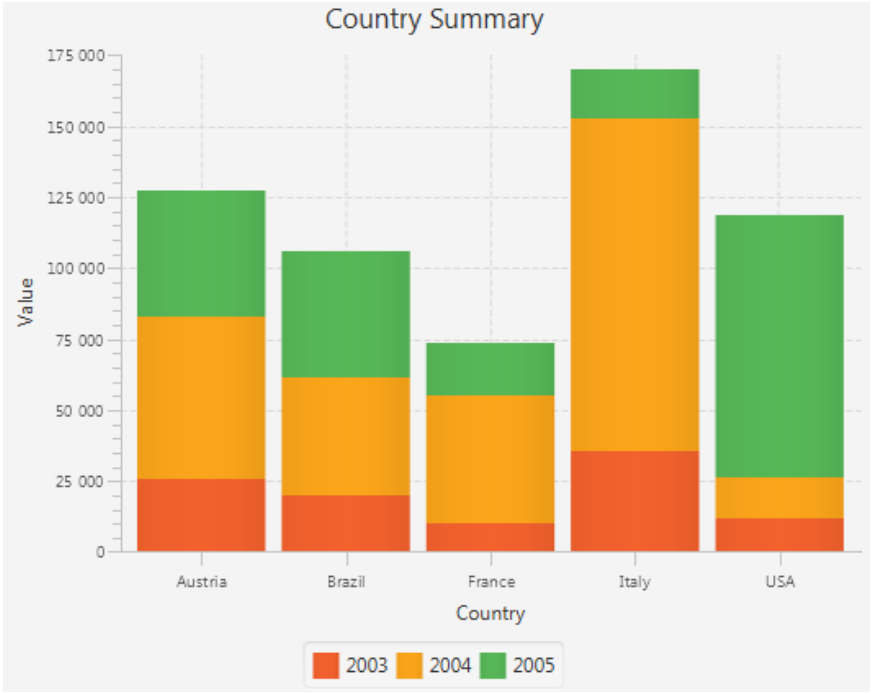
    ```
    barChart.setBarGap(3);
    ```

  - Set the gap between categories

    ```
    barChart.setCategoryGap(20);
    ```

- Stacked bar chart:
  - In the vertical axis the areas of the bars show cumulative values
  - In the sample chart: the value 125000 for Austria indicates the cumulative value for 2003, 2004 and 2005
  - Use class *StackedBarChart*



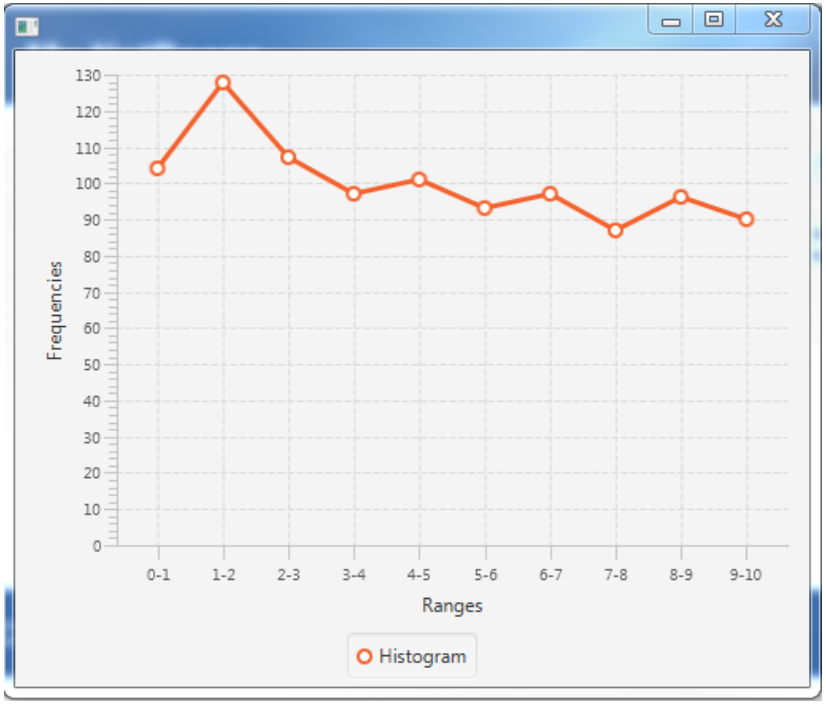


JavaFX Charts

- Support CSS style sheets
- Should be created from within Scene
- Note that in Scene Builder:
  - The X axis is always a *CategoryAxis*
  - The Y axis is a *NumberAxis*
  - Therefore, the data is *XYChart.Data(String, Number)*
  - Except for the *BubbleChart* and the *StackedAreaChart* where both axes are *NumberAxis*
  - *You can edit the FXML by hand to change the definition of the axes*
- For more information, including complete examples:
  - http://docs.oracle.com/javase/8/javafx/user-interface-tutorial/charts.htm

Example of chart

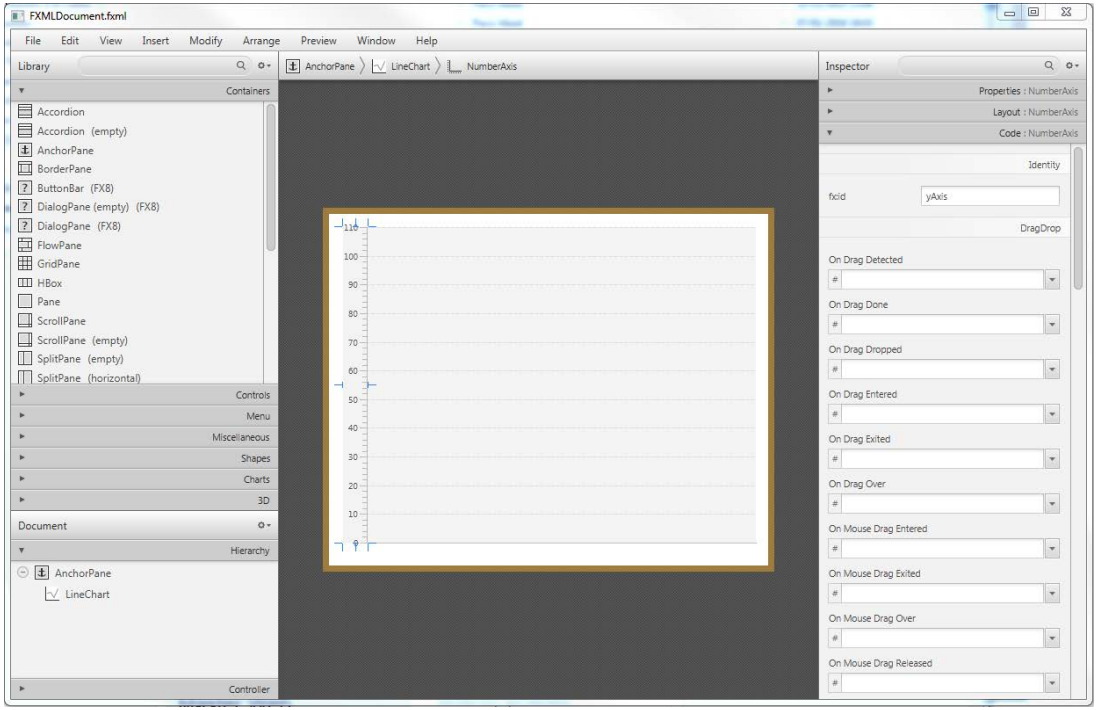
- In this example we will build a chart that shows the histogram of a set of random numbers
- If the numbers were uniformly distributed in the range, the histogram should show a flat line



Example of chart

- In Scene Builder, drag and drop a LineChart from the Charts section of the Library to the scene
- Set the *fx:id*s for the axes of the chart. Just select them in the scene

```
@FXML
private LineChart<String,Number>
        chart;

@FXML
private CategoryAxis xAxis;

@FXML
private NumberAxis yAxis;
```



Example of chart

- Then, generate the random numbers and compute their histogram

```
int hist[] = new int[10];

for (int i = 0; i < hist.length; i++)
    hist[i] = 0;

for (int j = 0; j < 1000; j++) {
    double value = Math.random() * 10;
    hist[(int)Math.floor(value)]++;
}
```

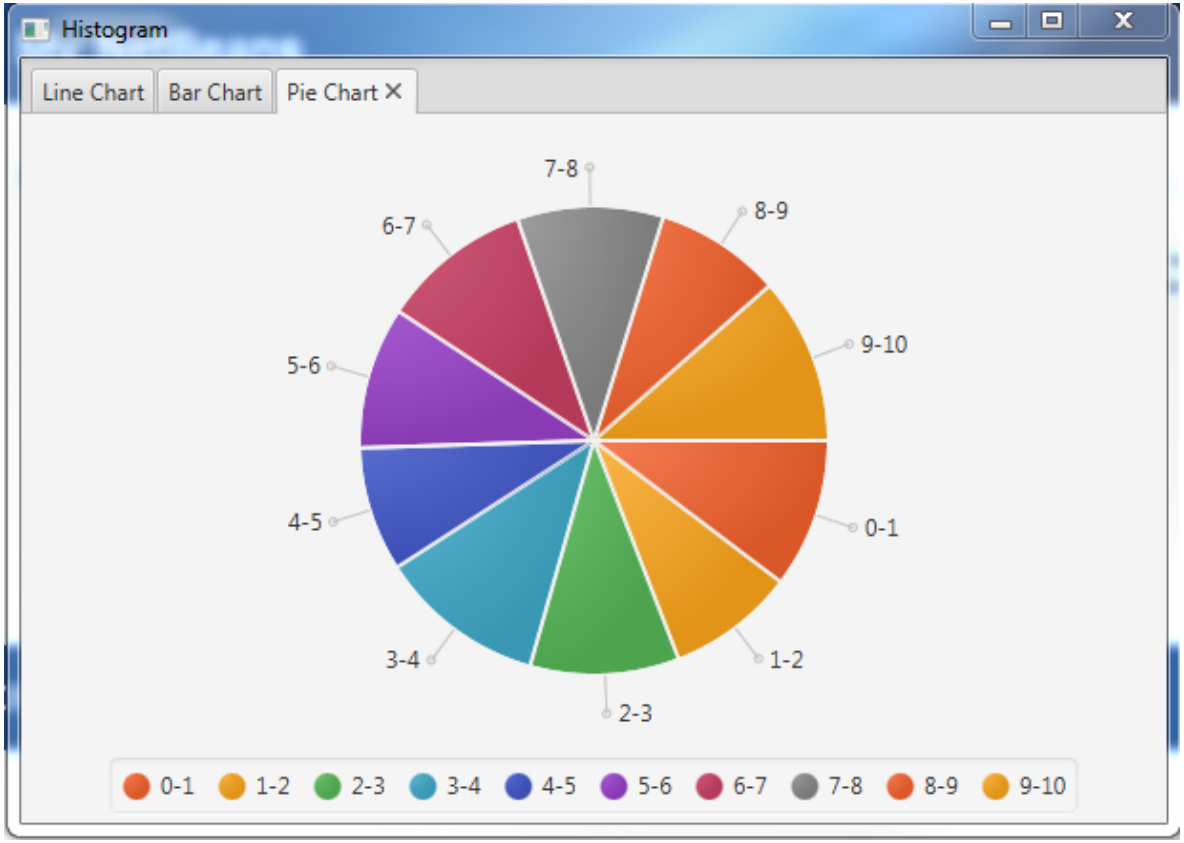
Example of chart

- Finally create a series with the data points and add it to the chart

```
xAxis.setLabel("Ranges");
yAxis.setLabel("Frequencies");

XYChart.Series<String,Number> series = new XYChart.Series();
for (int i = 0; i < hist.length; i++)
    series.getData().add(new XYChart.Data<>(i + "-" + (i+1), hist[i]));

series.setName("Histogram");
chart.getData().add(series);
```

Exercise

- Given the previous code, implement other types of charts



References

- http://docs.oracle.com/javase/8/javafx/user-interface-tutorial/charts.htm