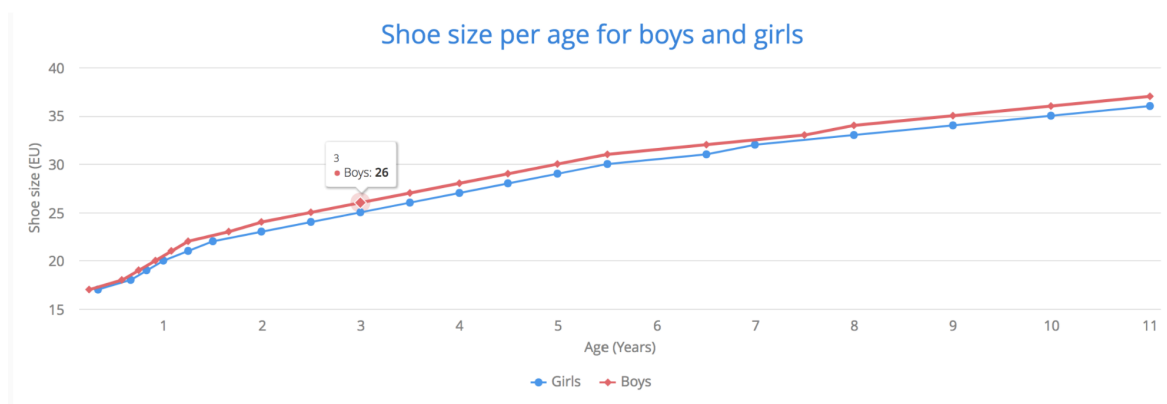


Exercise 1: Shoe size chart

The goal of this exercise is to demonstrate how you can show some data with Vaadin Charts.

In this exercise, you are supposed to make a shoe size chart, which has two data series, one for boys one for girls. You can get the data from ChartData, with methods `getGirlsData()` and `getBoysData()` respectively.

The X axis should be of age in years. and Y axis should be the shoe size. Note: LineChart should be suitable chart type for this kind of data. But feel free to try out different chart types. In the end, it should look something like this:



Exercise 2: Weather Chart

Now you have mastered the basic skill of using Vaadin Charts, let's do something a bit more complicated. In this exercise, you are supposed to make a Weather chart, which will display the weather information of Turku.

You can get the weather information from `ChartData` class, with method `getWeatherData()`.

To accomplish this exercise, you should know that:

- 1.1. How to display multiple chart types in one chart.
- 1.2. How to use `DataProviderSeries`.
- 1.3. How to enable Timeline for Charts.

You should set X axis type to be `AxisType.DATETIME`.

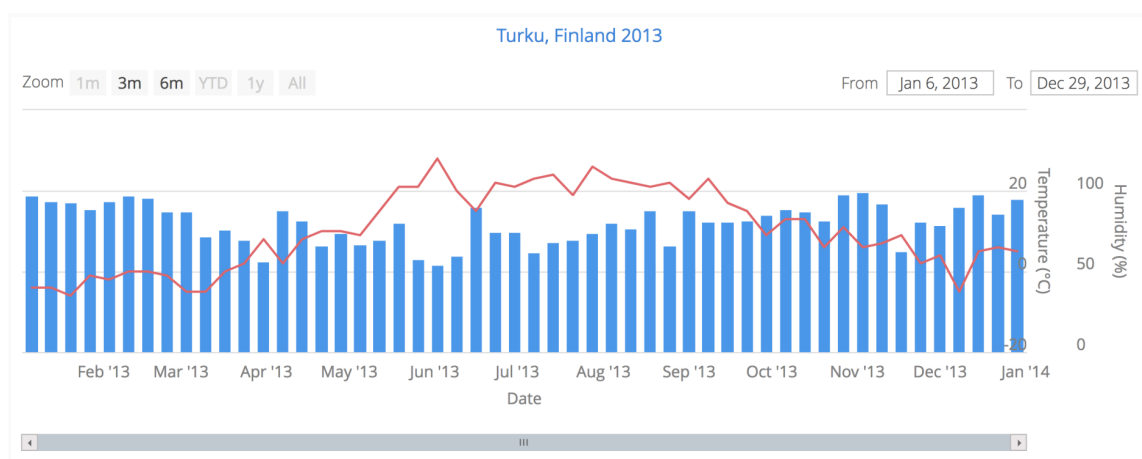
The value for X axis should be of type `LocalDateTime`, while `WeatherInfo` has the type of `Instant`, to convert `Instant` to `LocalDateTime`, you can use: `LocalDateTime date = LocalDateTime.ofInstant(point.getInstant(), ZoneId.of("Europe/Helsinki"));`

To display multiple chart types, you simply give different `PlotOptions` to different `DataSet`s.

Let's use a `LineChart` for the Temperature and `ColumnChart` for Humidity. As you can see from `ChartData`, that you have quite a lot weather data, from 2013-01-06 to 2013-12-31, for every single day.

You don't really need this many, so you can wrap the data into a `ListDataProvider`, and filter the data out, a datum per week should be sufficient, for example, we can take data from every Sunday. To enable Timeline is really easy, you can just say `charts.setTimeline(true)`.

For this exercise, let's also hide the navigator. In the end, it should look like:



Exercise 3: Range Data Chart

Let's try displaying the same data in a different format, and try using a different type of Series as well. In this exercise, our goal is to display the temperature ranges of our weather data using an Area Range chart. We'll also try out some additional functionality.

This time, the X axis should be a DATETIME axis. Enable the Crosshair functionality of the X axis so you'll see better where the mouse is hovering over the data. Add a Tooltip to the Chart's Configuration so you can modify the way the information is displayed on hover - add a Value Suffix like "°C" so the value's type is indicated correctly.

The interesting part of the exercise is getting the data out correctly. The ChartsData getMinMaxTemperaturesByDay should return a matching database for the RangeSeries constructor. Since the X axis is of the type DATETIME, you should produce an array of Number arrays - Number [] [] - where each array entry is a three-value array of { date, min temperature, max temperature }. The date value should be a Unix milliseconds timestamp you get out of java.util.Date:s getTime().

The output should look like this:

