Introduction

The purpose of this tutorial is to teach you the basics of loading, manipulating and visualising data in Processing with the HiVis library.

Outcomes

By the end of this tutorial you will be able to:

- load data from a spreadsheet with HiVis;
- perform basic manipulation or “shaping” of the data with HiVis; and
- visualise the data using basic drawing commands in Processing.

Getting started

The first thing to do is install HiVis and another library, Beads, that we’ll use to “render” some data with sound.

1. Open Processing.
2. In the menus go to Sketch > Import Library > Add Library
3. In the filter box type in “hivis”
4. Click on the HiVis library and then click the Install button.
5. Repeat steps 3 and 4 for the Beads library.

Loading data from spreadsheets / Terminology

HiVis expects the data in a spreadsheet to be in a particular format. Each row of the spreadsheet should contain one record or “data point”. For example if the data is recording the name, hair colour and height of people then the spreadsheet will have columns for name, eye colour and height. The first row may optionally contain headings for the columns. Thus the spreadsheet will look something like this:

Name	Eye colour	Height
Fin	Brown	161
Tovar	Green	183

When HiVis loads this spreadsheet it will create a *DataTable*. The DataTable will consist of three *DataSeries*, one for each column. The series will be of length 2 as the first row will be used to label the series in the table. When manipulating the data we'll be able to refer to the series by their index in the spreadsheet or by their label (if present in the spreadsheet).

A note on indexing into DataTables and DataSeries: when referring to series in a data table or an element of data in a series by their index, the indexes start at **0**, not 1. So to get the first element you might type something like `myTable.getSeries(0)`, and the second would be `myTable.getSeries(1)`.

Exercise 1 – Scatter Plot

In the first exercise we'll look at producing a scatter plot. In this plot we draw a dot for each data point. The position of a dot (the x and y coordinates) is determined by two values from the data point.

1. Create a spreadsheet with three columns. Each column should have a header (make something up) followed by 10 rows of positive numerical values (whatever you like). Save the spreadsheet in Microsoft Excel format (xlsx).
2. In the Processing menus go to *File* > *Examples*. Then open *Contributed Libraries* > *HiVis* > *tutorials* > *T01_Scatter*.
3. Click Run and select the spreadsheet you created. You should see a scatter plot.
4. Try changing the values in the spreadsheet and saving it. The plot should be updated with the new values. Now try adding new data points in the spreadsheet and saving it.
5. Close the plot.
6. Take a moment to read through and get familiar with the code. Don't worry if some new commands or methods don't make sense immediately.
7. The plot is using the second and third columns/series (indices 1 and 2) for the x and y coordinates respectively. Change it to use different series for the coordinates and run it again.
8. The plot can only currently represent two values from each data point. Let's change it so that it can represent three values. We'll do this by changing the brightness of the dots according to the values in one of the series:
    a) Add a new DataSeries variable to contain the values for the brightness (see lines 13 and 14).

b) In the `fileSelected` method, set the new series variable to one of the series in the DataTable not used for the coordinates (see lines starting with `xSeries =` … and `ySeries =` …). Add your new line before the `xSeries =` … line (if you're wondering why, have a look at the line containing `if (ySeries != null)` in the `draw` method and see if you can figure it out).

c) In the `for` loop in the `draw` method extract the value from the new series into a new variable to store the brightness for the dot. Multiply the value by 255 because Processing expects colour values to have a value between 0 and 255 by default (see lines starting `float x =` … and `float y =` …).

d) Add a `fill` command just before the `ellipse` command to change the brightness of the dot, using the new brightness variable you added.

Exercise 2 – Pie Charts and Data Manipulation

In the second exercise we'll generate pie charts using data from the spreadsheet you made in Exercise 1. We'll select which data to chart and manipulate the data before visualising it.

1. In the Processing menus go to *File > Examples* (if it's not open already). Then open *Contributed Libraries > HiVis > tutorials > T02_Pie*.
2. Take a moment to read through and get familiar with the code **(ignoring the exercises listed at the top**, we're doing something much more exciting, no really**)**.
3. Click Run and then select your spreadsheet. You should see several pie charts nested inside each other.
4. Let's modify the code so that we only chart the first and third columns, and only rows 2 through 8 (counting from 1). Change the line
   ```
   data = HV.loadSpreadSheet(selection);
   ```
   to
   ```
   data = HV.loadSpreadSheet(selection).selectSeries(0,
   2).selectRowRange(1,7);
   ```
   [All on one line].
5. Click Run and see how it looks.
6. The `selectSeries` and `selectRowRange` methods can be used with any DataTable. There are many other methods for selecting series and rows from DataTables, and

rows/elements from DataSeries, check out the first two examples in *Contributed Libraries > HiVis> examples* after the tutorial.

7. Now let's pretend that there is an error in our data set: the values in the first and third columns accidentally had some amount subtracted from them when they were recorded. Let's modify the data to correct this. Change the line
   ```
   DataSeries series = data.getSeries(s);
   ```
   to
   ```
   DataSeries series = data.getSeries(s).add(10);
   ```
   [pick a value that is a bit smaller than the typical magnitude of the values in your data.]
8. Click Run and see how it looks.
9. Finally, let's pretend that what we're really interested in is the ratio of the first and third series. We want to chart the values in the first series divided by the corresponding values in the third series (while still correcting the values in the first and third series).
   a) Comment out the for loop (don't forget the matching closing brace).
   b) Change the line:
   ```
   DataSeries series = data.getSeries(s).add(10);
   ```
   to
   ```
   DataSeries series =
   data.getSeries(0).add(10).divide(data.getSeries(1).add(10));
   ```
   c) Click Run and check out your visualised shaped data.

Exercise 3 – Sonification

In the third exercise we're back to the scatter plot, but this time with a twist..

1. In the Processing menus go to *File > Examples* (if it's not open already). Then open *Contributed Libraries > HiVis > tutorials > T03_Sonification*.
2. Take a moment to read through and get familiar with the code.
3. Click Run and then select your spreadsheet.
4. With the volume turned up on your computer, try hovering the mouse pointer over the dots. Have fun annoying your friends by trying to compose a tune with your deft hovering skills.
5. Modify the code so that when hovering over a data point, text is displayed next to it that lists the values for that point in each series.