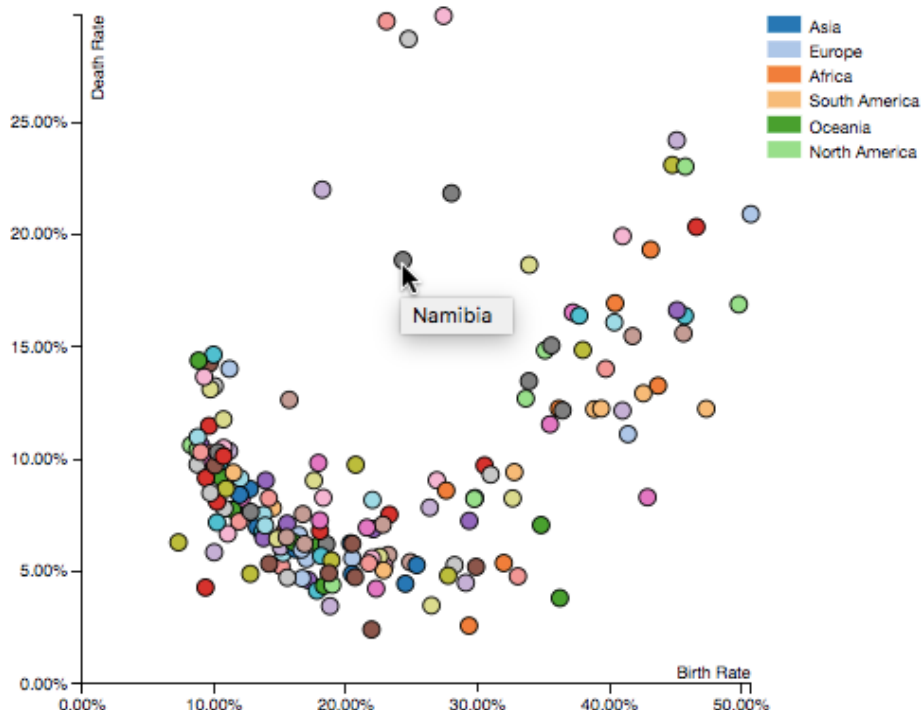


Assignment 1

In this assignment, you are expected to create a scatterplot which is sometimes called bubble plot. Scatterplots are effective for abstract tasks of providing overviews and characterizing distributions, and specifically for finding outliers and extreme values. Also, it is highly effective for the abstract task of judging the correlation between two attributes.

You are going to work on a dataset of country statistics of 169 different countries/regions around the world which includes 14 statistic descriptors, such as, population, area size, GDP, birth rate and death rate. The dataset is given in “countries_of_world.csv”. The following figure shows how your scatterplot should look like.



Requirements:

1. Creating a scatterplot of Birth Rate and Death Rate (vertical axis 15, horizontal axis 15, data points 15, legend 15);
2. Adopting percentage scale to both axes (15);
3. Using color to encode regions (15);
4. Adding a tooltip to a data points which will show the name of the country when a mouse hovers on it (10);

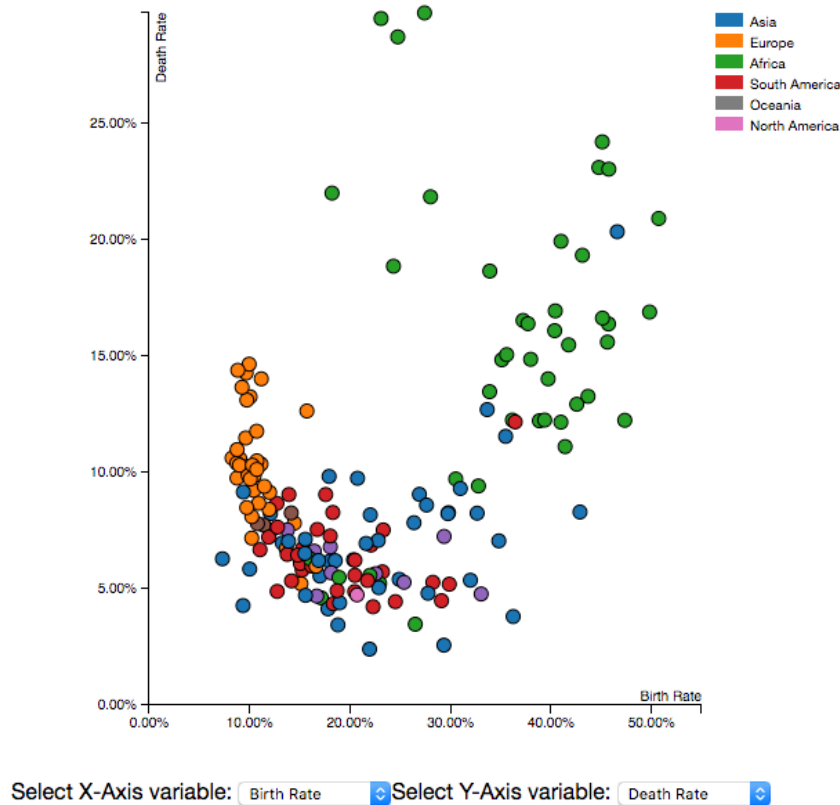
Bonus:

As you may notice, some data points are overlapping in the figure. So, please develop a remove function which is able to remove a point with a click on it (5).

Assignment 2

In this assignment, you will handle the multiple-dimension data with D3.js. We have built a scatterplot in Assignment 1, but it only shows the relations between two attributes: Birth Rate and Death Rate. Actually, we have 14 attributes in the given dataset, so there are 91 different attribute pairs in total. In order to observe the relation in each attribute pair, we have to draw 91 Scatterplots. Of course, it is inadvisable and intractable to plot so many figures in a web page. One solution for this is to use drop-lists.

You need to design two drop-lists for x-axis and y-axis respectively, which allows users to select the attributes, and naturally, the scatterplot should switch along with their selections. Besides, you will put your animation designs into the scatterplot. When the attributes are switched, the plots should shift to their new positions correspondingly. You need to design animations for this progress and implement them with d3.js. The following figure is an example.



Requirements:

1. Creating drop lists for both axes (20);
2. Updating the axes and data points in the scatterplot according to the selected attributes (20);
3. Adding animation to the data points when they shifting to new positions (20);
4. Adding animation to the points: When the mouse hovers on a point, it will be enlarged; when the mouse moves out, the point will turn back to normal size (20);
5. Zoom in/out is enabled (20).

Before you start:

In this task, you need to load the CSV file from our local disks. However, some browsers do not allow users to reading local files due to security reasons. One solution is to build a **local web server** on your laptop. This can simply be done with Python by the following steps,

1. open a terminal window
2. navigate to the directory where your html files in: **cd** the path to your folder
3. For Python 2 input: **python -m SimpleHTTPServer 8000**
For Python 3 input: **python -m http.server 8000**
4. start a web browser, visit the address: **http://localhost:8000**, the files in your entire directory are listed there.