Data Binning

Binning is a way to group a number of more or less continuous values into a smaller number of "bins". For example, if you have data about a group of people, you might want to arrange their ages into a smaller number of age intervals. Numeric columns can also be temporarily grouped by right-clicking on a column selector and clicking Auto-bin Column.

There is also an option to group categorical values into bins. This is useful when you have more categorical values in a column than you find necessary. Your visualization may for example show sales of apples, pears, oranges and limes, but you are interested in citrus fruit sales compared to apples and pears sales. Then oranges and limes can be grouped into a bin.

Note: A special use case of this binning method is grouping values that are misspelt or differ due to other reasons. For example, if a column contains values like “apple” and “appel”, or “UK” and “United Kingdom”, you can group these values into bins.

Data binning, **bucketing** is a data pre-processing method used to minimize the effects of small observation errors. The original data values are divided into small intervals known as bins and then they are replaced by a general value calculated for that bin. This has a smoothing effect on the input data and may also reduce the chances of overfitting in the case of small datasets
There are 2 methods of dividing data into bins:

1. **Equal Frequency Binning:** bins have an equal frequency.
2. **Equal Width Binning :** bins have equal width with a range of each bin are defined as [min + w], [min + 2w] …. [min + nw] where **w = (max – min) / (no of bins).**

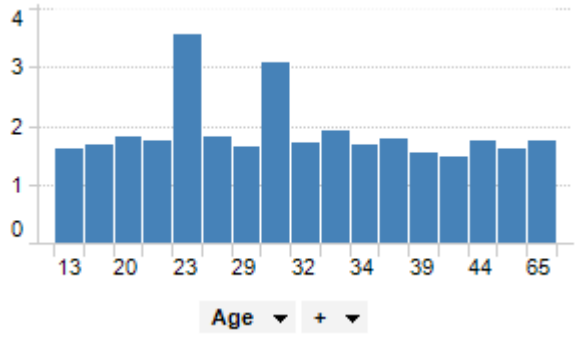
Equal frequency:
Input:[5, 10, 11, 13, 15, 35, 50, 55, 72, 92, 204, 215]

Output:
[5, 10, 11, 13]
[15, 35, 50, 55]
[72, 92, 204, 215]
Equal Width:
Input: [5, 10, 11, 13, 15, 35, 50, 55, 72, 92, 204, 215]

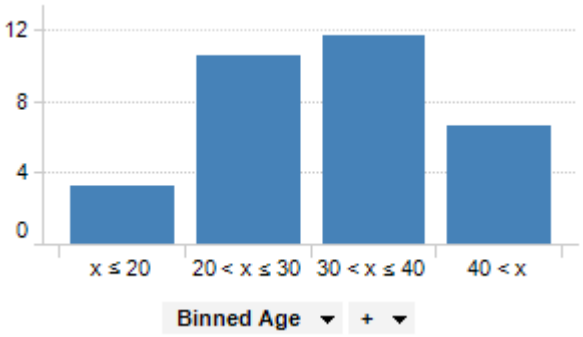
Output:
[5, 10, 11, 13, 15, 35, 50, 55, 72]
[92]
[204, 215]

Example of binning continuous data:

The data table contains information about a number of persons.

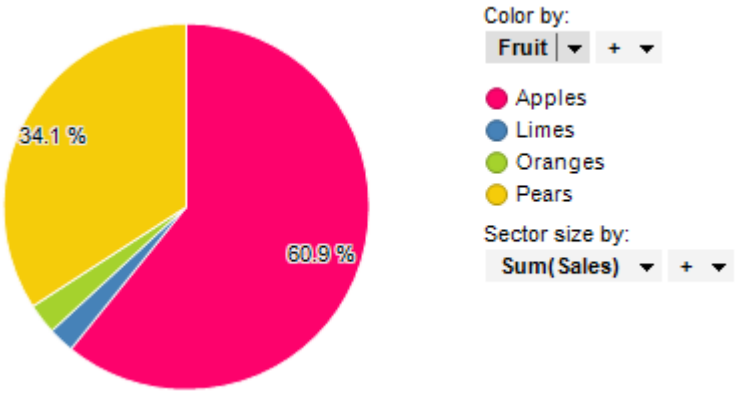


By binning the age of the people into a new column, data can be visualized for the different age groups instead of for each individual.

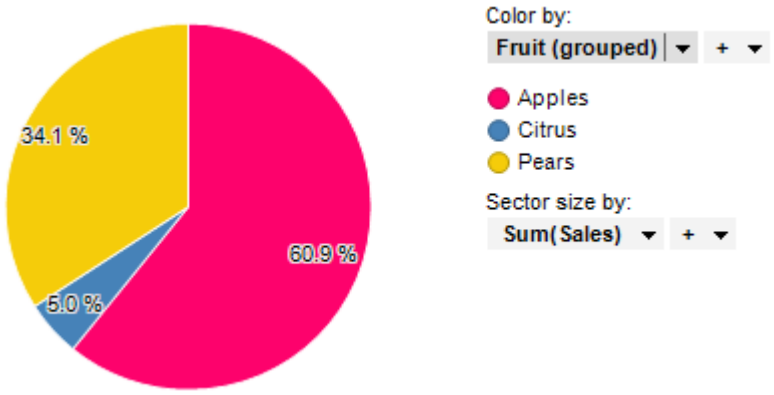


Example of binning categorical data

The pie chart shows sales per apples, limes, oranges and pears.



Below oranges and limes have been grouped into a bin called “Citrus”.



Binning in Scatterplots

Scatterplots are a straightforward way to visualize the data distribution in a XY plane, especially when we are looking for trends or clusters. But when you have a dataset with a large number of points, many of these data points can overlap. This overalpping effect can make difficult to see any trends or clusters.

For example, let's take these two different datasets which are represented in the following scatterplots (see Figures 1 and 2). The first scatterplot unequivocally shows the linear trend underlying the dataset. Instead the second scatterplot apparently shows a uniform distribution of the data points throughout the XY plane (actually there are a lot of overlapping points that are not visible).

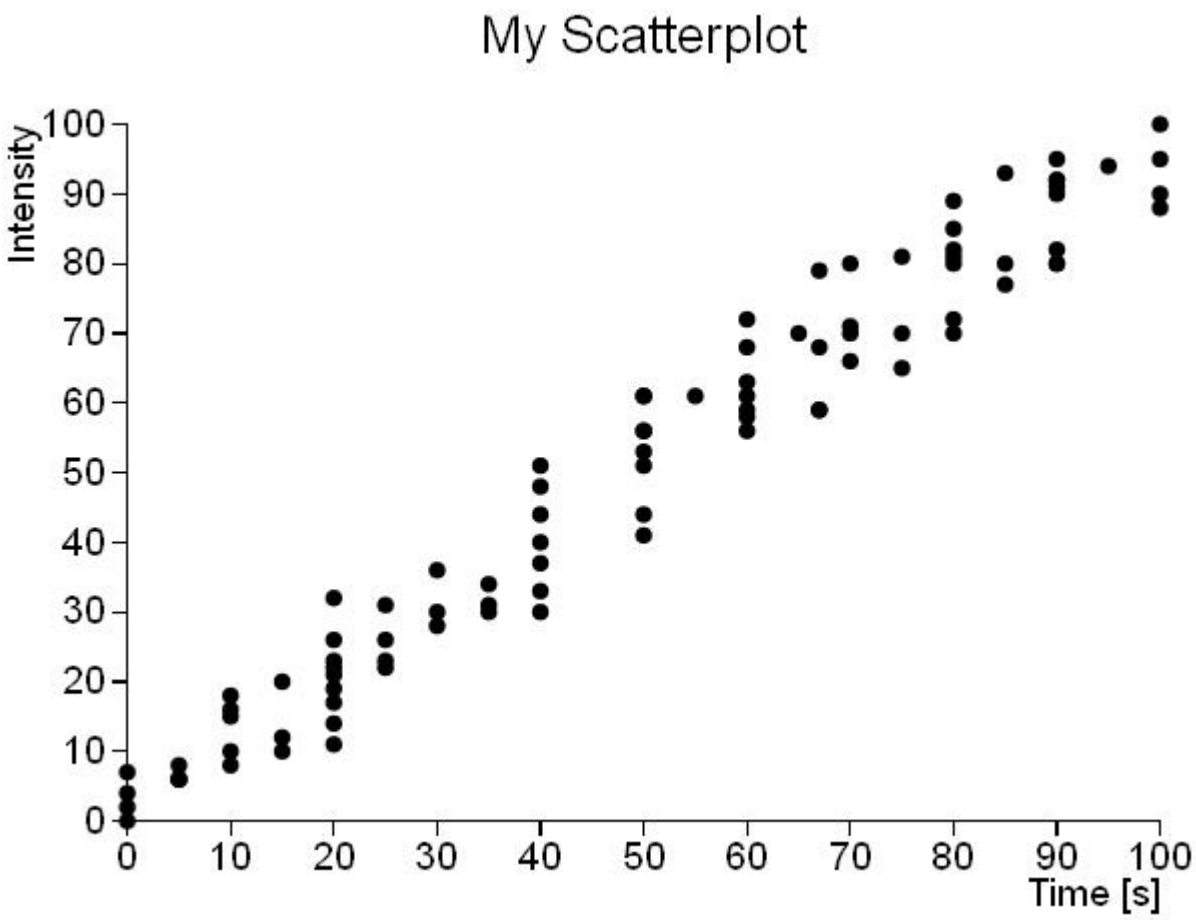


Fig.1: a scatterplot showing a linear trend

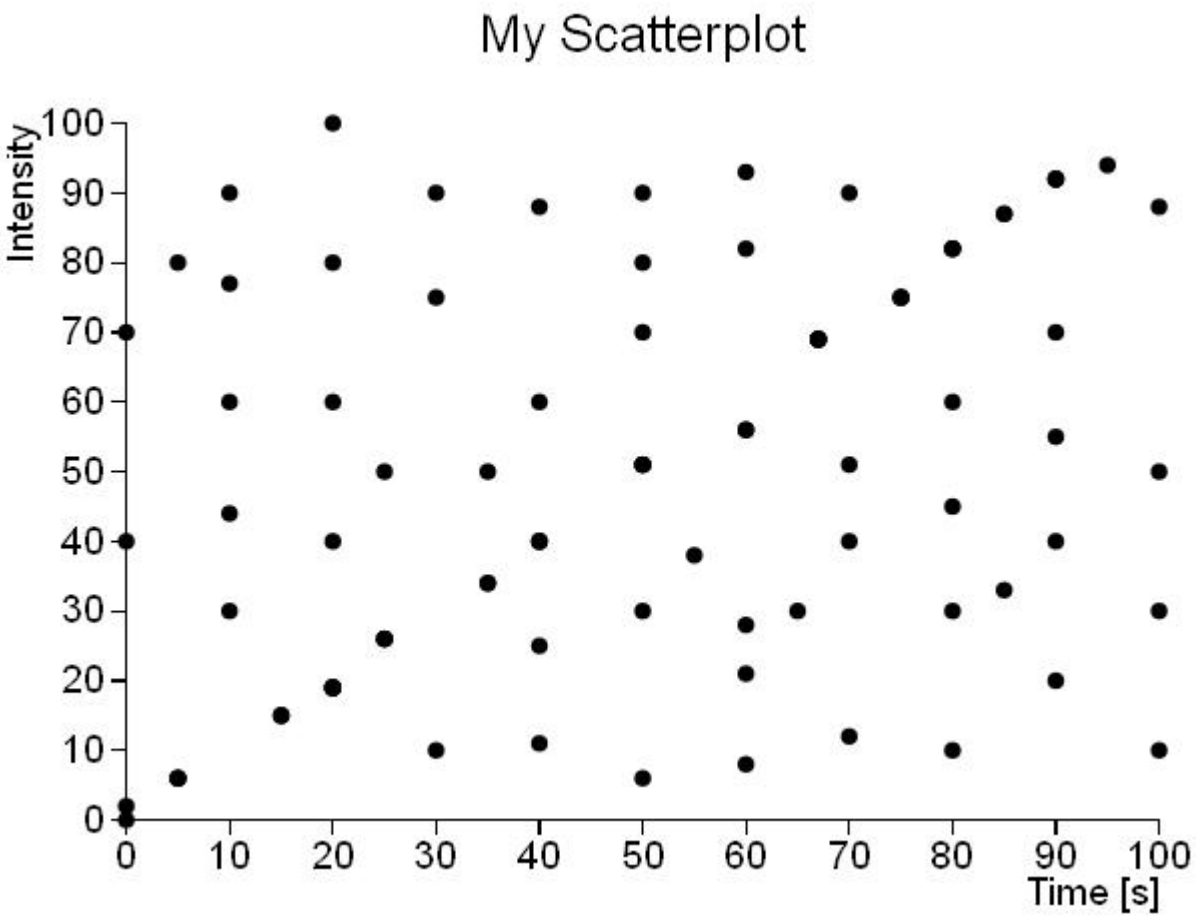


Fig.2: a "sparse" scatterplot

Note: I have used this "sparse" dataset to make easier to understand the concepts covered in this article. The following picture (see Figure 3) shows a more real case.

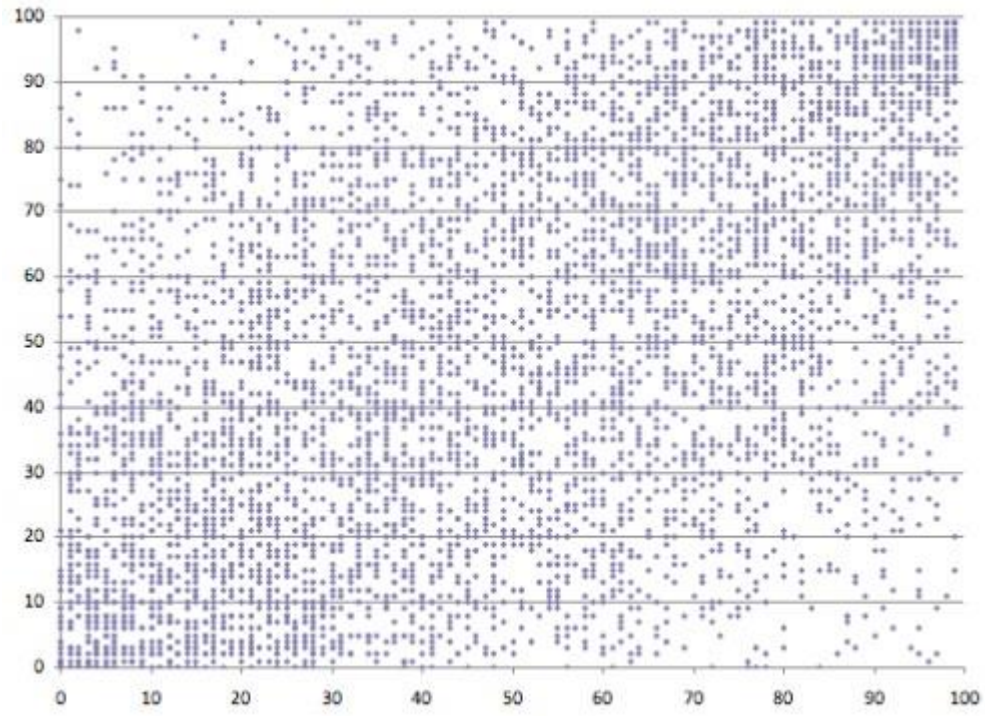


Fig.3: a real sparse scatterplot

If you analyze in detail the second dataset, it turns out that multiple data points are occupying the same place in the scatterplot, thereby distorting the visualization of the data distribution on the XY plane. Thus, a scatterplot can only represent point density up to a certain threshold.

In these cases, we need to choose another type of visualization which uses binning methods, in order to plot the point density rather the point themselves. A binned representation is a technique of data aggregation which may reveal patterns not readily apparent in a scatterplot.

Binning

Binning is a technique of data aggregation used for grouping a dataset of N values into less than N discrete groups. In this article we are considering only the case of datasets build up of (x,y) points distributed on a XY plane, but this technique is applicable in other cases. This technique is based on extremely simple concepts.

- the XY plane is uniformly tiled with polygons (squares, rectangles or hexagons).
- the number of points falling in each bin (tile) are counted and stored in a data structure.
- the bins with count > 0 are plotted using a color range (heatmap) or varying their size in proportion to the count.

Note: If we consider the case of monodimensional datasets, the binning technique generates **histograms**.

Rectangular Binning

The simplest binning method use square tiles, and for most purposes this suffices, taking advantage of its computational simpliticy.

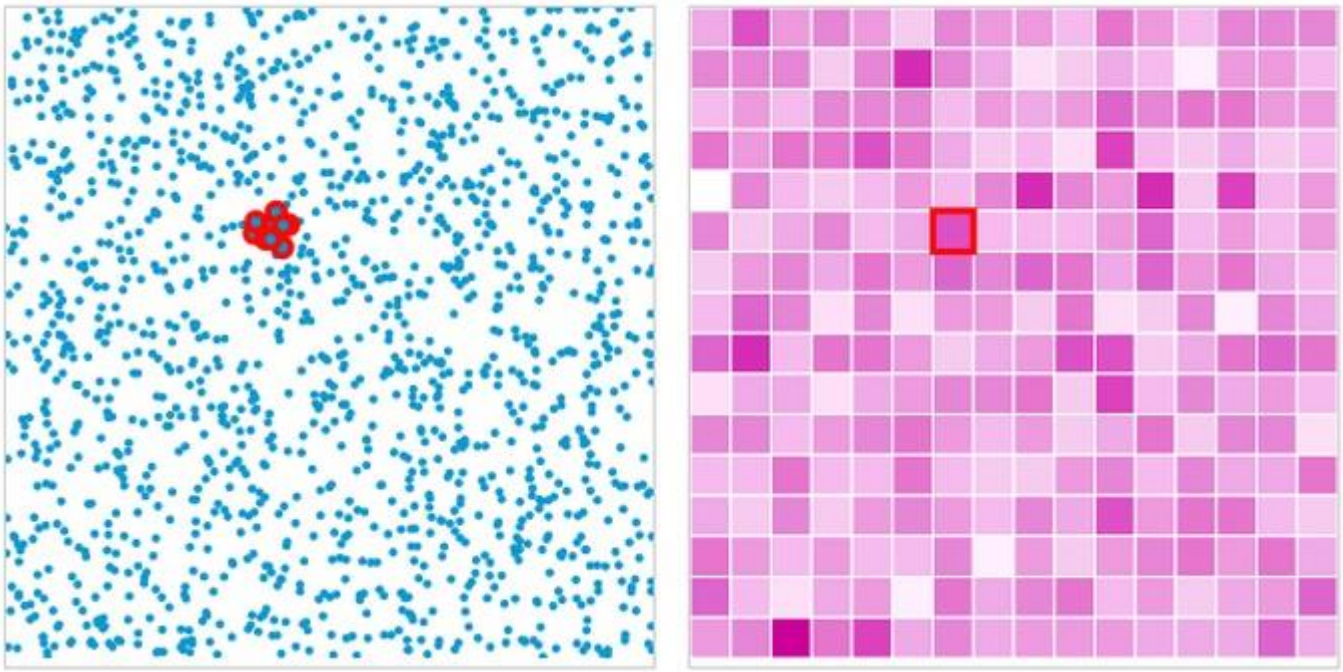


Fig.4: rectangular binning

If you are interested to develop charts using the rectangular binnings method, there is a **tutorial** about this topic showing how to make it using the JavaScript library D3.

Hexagonal binning

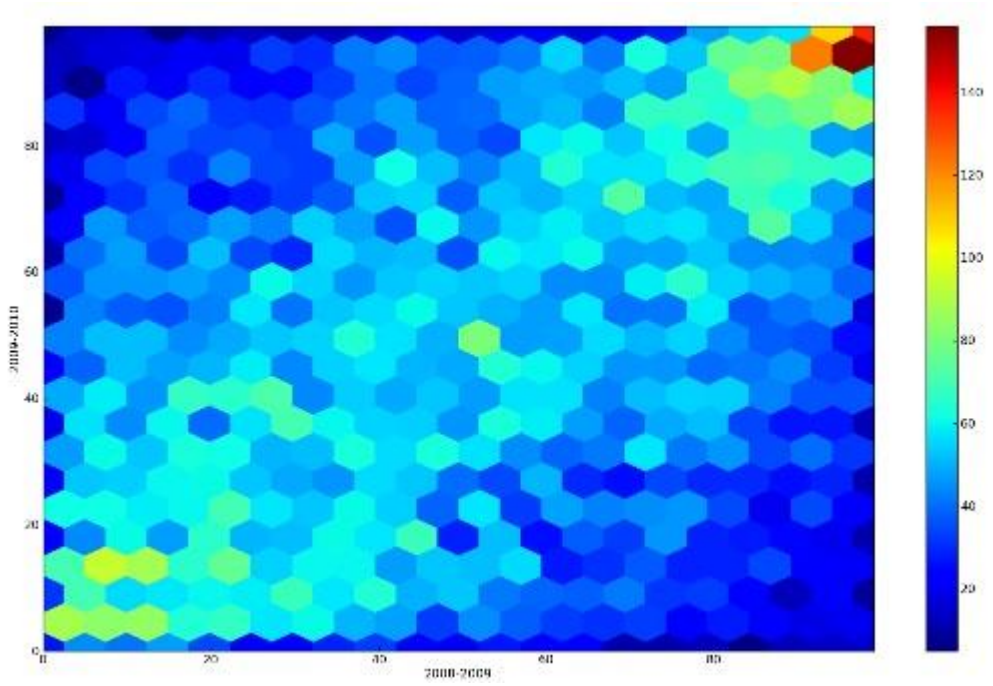


Fig.5: a hexagonal binning

This technique was first described in 1987 (*D.B.Carr et al. Scatterplot Matrix Techniques for large N, Journal of the American Statistical Association, No.389 pp 424-436*). There are many reasons for using hexagons instead of squares for binning a 2D surface as a plane. The most evident is that hexagons are more similar to circle than square. This translates in more efficient data aggregation around the bin center. This can be seen by looking at some particular properties of hexagons and, especially, of the hexagonal tessellation.

- Hexagon is the polygon with the maximum number of sides for a regular tessellation of a 2D plane.

This makes the hexagonal binning the most efficient and compact division of 2D data space.

Fig.6: the hexagonal tessellation

In fact, although you can create many pattern using two or more types of polygons, this is not possible if you are using the same polygon if this has more than 6 sides. Only triangles, squares and hexagon can create them.

- In an hexagonal binning, adjacent hexagons shares edge borders and not only vertex borders.

Instead in square and triangular binning, triangles and square share only a vertex border with some adjacent.

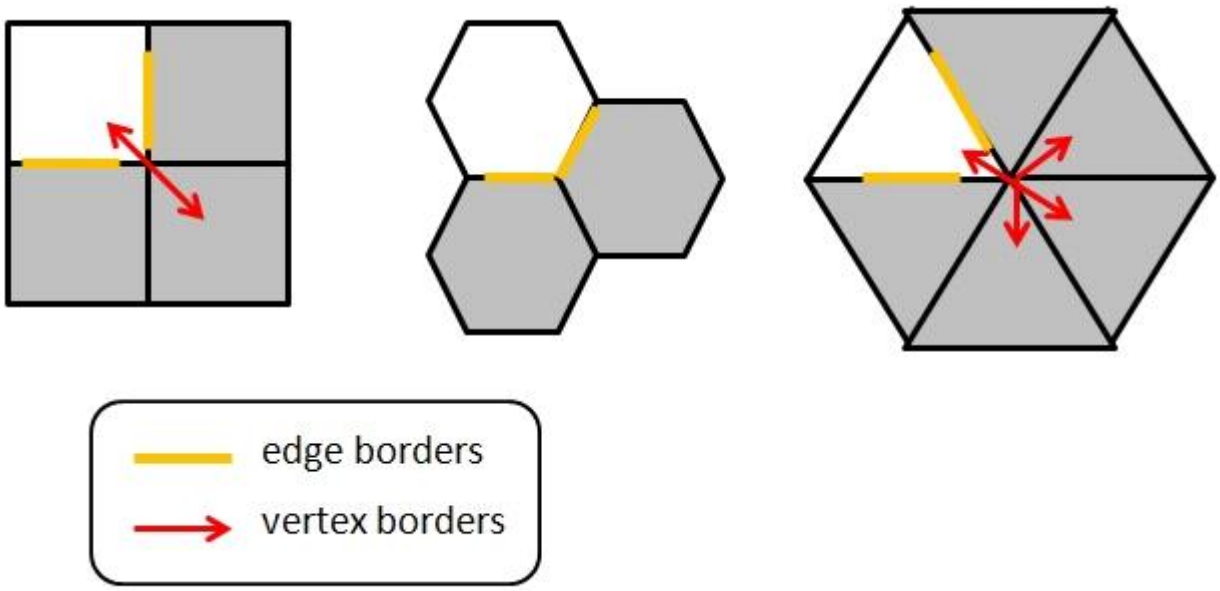


Fig:7. sharings of the borders in different tessellations

Considering polygons with equal area, the more similar to a circle this polygon is, the closer to the center the border points are (especially vertices).

Thus any point inside a hexagon is closer to the center of any given point in an equal area square or triangle would be. This is because square and triangles have more acute angles.

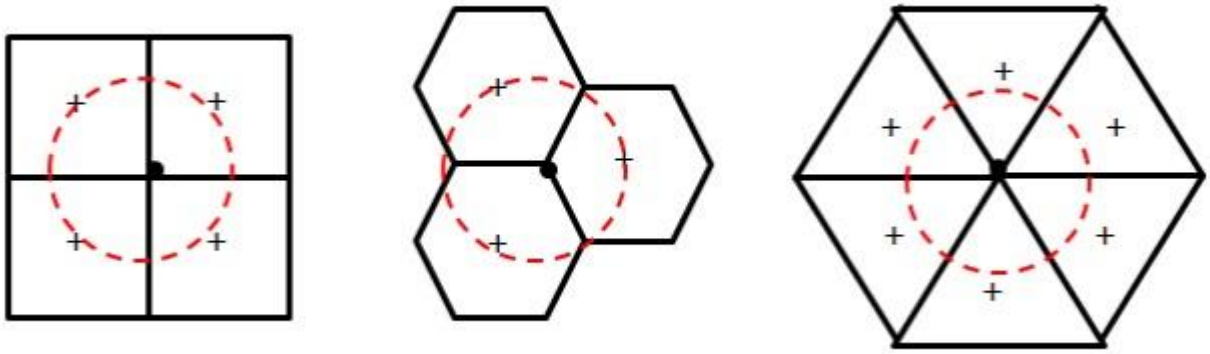


Fig.8: distances of vertices point from the centers

Sparse Scatterplot in Hexagonal Binning

Now that we have an idea of what the hexagonal binning is, let's submit the dataset which generated a "sparse" scatterplot to a hexagonal binning. This is the result:

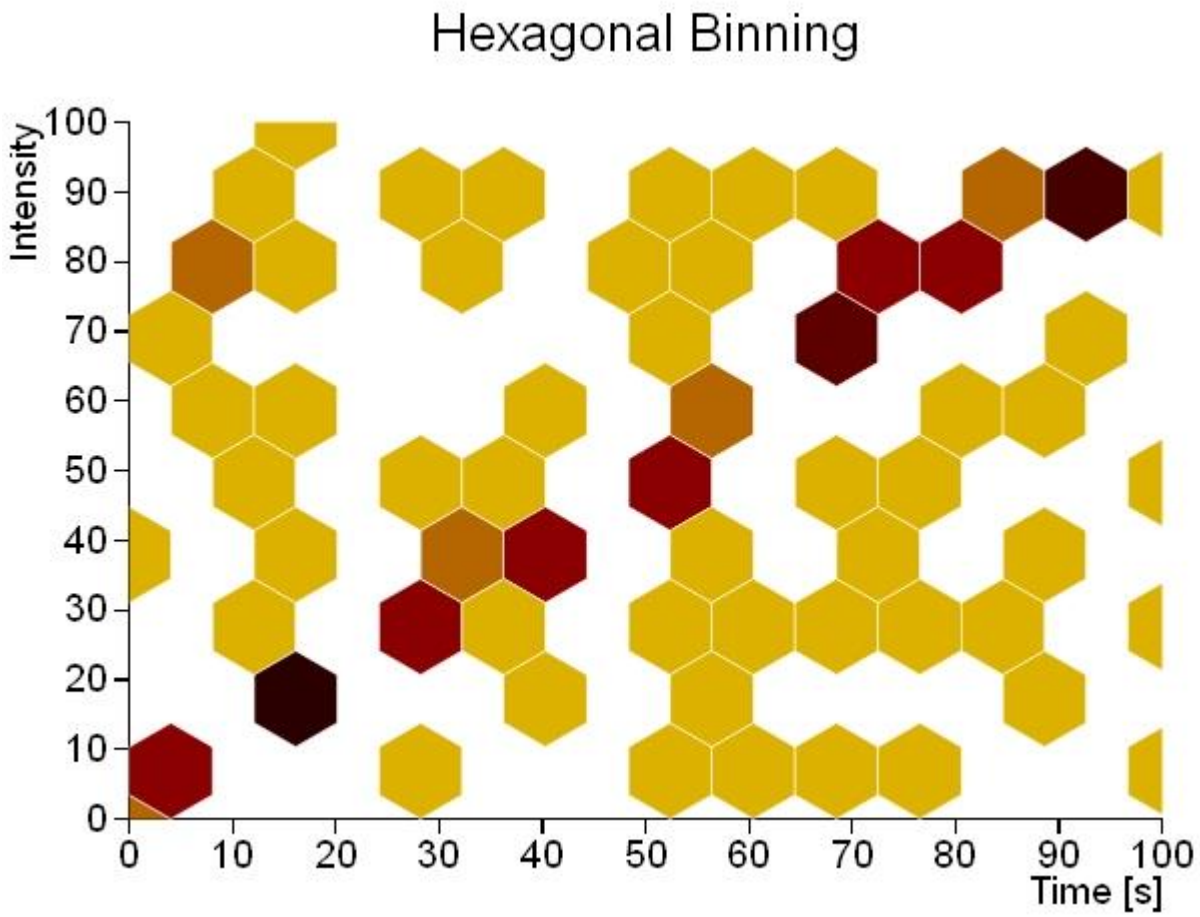


Fig.9: hexagonal binnings applied to the sparse dataset