Visualizing Data Reduction in Three Dimensions

Kevin McCoy

November 9, 2023

Summary

In the advent of the information age, John Tukey astutely warned the mathematics and statistics community that soon, data would be everywhere. It's true: in fact, all of the worlds digital data continues to double every two years. It was for this reason that Tukey advises us to become more familiar with visualizing data. You may be wondering how this is different then any other day to day work of mathematician. Well, the answer is that when assessing data we use formalized models we call hypotheses. However, these are not hypotheses of a typical sense. Instead, hypotheses are reference situations that we can compare the data to. Now which model do we choose, given a set of data? One can use data visualizations for this purpose.

Tukey argues that most visualizations in the current literature were used wastefully, to present summaries that could be illustrated in even simpler ways. Instead, a good visualization "reveals the unexpected" and makes the complex easier to perceive". The ultimate purpose here is to suggest the analyst's next step, or next insight. Lastly, our picturing of the data must be sensitive to both the hypotheses we have and have not considered.

Elaborating on this idea, Tukey argues that our own mental pictures of data are equally important, thus making it very important for us to understand matrices. Matrices can represent a linear transformation, a quadratic form, or a change of coordinates. A matrix also always involves two specific vector spaces and also two specific coordinate systems. To simplify our understanding of a matrix down to "$n \times p$ matrix has n rows and p columns" is a large source of our own errors and misunderstandings.

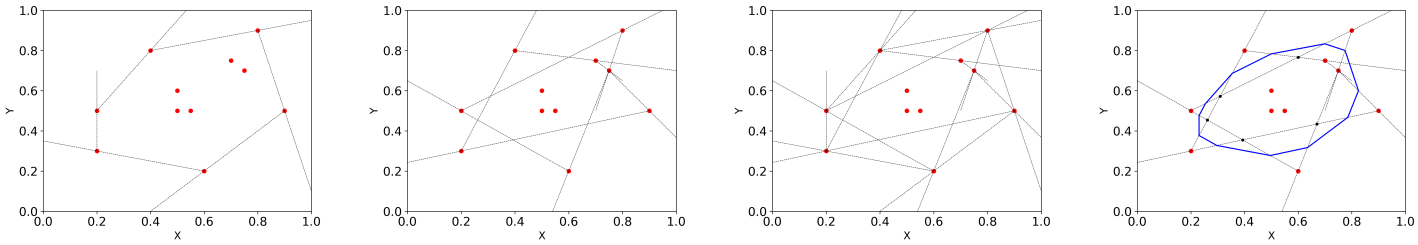
Typically, the cumulative distribution function (CDF) is defined as $F_X(x) = \mathbb{P}(X \leq x)$. However, Tukey argues that the alternative definition of $F_X(x) = \mathbb{P}(X < x) + \frac{1}{2}\mathbb{P}(y = x)$ is preferable, as it comes with the desirable quality that the CDF of $-y$ is exactly $1 - F_X(x)$, including any discontinuities, and makes Fourier inversion exact at any such discontinuities. This idea can be extended to empirical CDFs as well, where it is better to define:

$$n \cdot F_n(x) = \#(y_i < x) + \frac{1}{2}\#(y_i = x) \qquad (1)$$

The statistician is usually only concerned with symmetric functions of the sample $y_1, y_2, \ldots, y_n$. The most general function is the order statistics, $y_{(1)}, y_{(2)} \ldots, y_{(n)}$. The distribution of the order statistics, $F(y_{(i)})$ depends only on n and i, and not the underlying distribution F, so long as F is continuous. The median is then $med(F(y_{(i)})) \lessapprox \frac{i-\frac{1}{3}}{n+\frac{1}{3}}$, thus $med(y_{(i)}) \lessapprox F^{-1}\left[\frac{i-\frac{1}{3}}{n+\frac{1}{3}}\right]$. This fraction is a much better representation of the 'ith of n' then any alternative. Now we can write a further revised eCDF as:

$$F_n(x) = \frac{\#(y_i < x) + \frac{1}{2}\#(y_i = x) + \frac{1}{6}}{n + \frac{1}{3}} \qquad (2)$$

The order statistics serve as a natural basis for data reduction as they have a common correlation structure, with equal correlations for equal values of i_1/i_2. Thus one we can summarize all order statistics with only $2\log_2 n$ of them. On particularly auspicious example is depth, where:



(a) Points and their 1-1/2 Polygon (b) The Same Points and their 2-1/2 Polygon (c) The Points and their 2-Loop (d) Construction of the 2 Polygon

Figure 1: A recreation of the original figure 2 using Python. This code works for any set of points, but is shown above using the original points for clarity.

$$\text{depth of median} = \frac{1}{2}(1+n)$$
$$\text{depth of hinge} = \frac{1}{2}(1+\lfloor\text{depth of median}\rfloor)$$
$$\text{depth of eight} = \frac{1}{2}(1+\lfloor\text{depth of hinge}\rfloor)$$
$$\dots$$

From these selected order statistics, $L_k, \dots, L_2, L_1, M, U_1, U_2, \dots, U_k$, we can define the middle $M_j = (U_j + L_j)/2$, and the spread $S_j = U_j - L_j$. These can further be normalized against a standard Gaussian and used to assess skew.

Tukey finishes by generalizing the concept of order statistics to the affine plane. First we define $y_{(i)}$ as the point with $\geq i$ points to the left or on it, and $\leq i-1$ strictly to its left. In the plane, then, the (i,j) line will have $\geq i$ points to the left or on it, and $\leq j$ strictly to its left. For any $i \neq n$, there is one and ony one $(i, i-1)$ line, the set of which form a closed curve of lines of depth i, called the i-loop. The set of $(i, i-2)$ lines form the closed $(i-\frac{1}{2})$-polygon. Finally, the i-polygon is defined by the midpoints of the segments cut off by the $(i-\frac{1}{2})$-polygon from the extensions of the sides of the $(i+\frac{1}{2})$-polygon.

Extensions into 3D

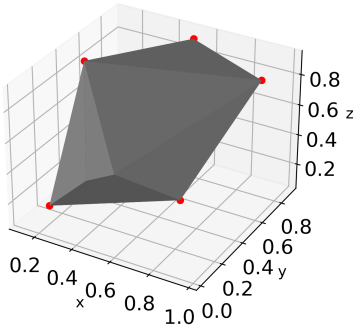
Extending this idea into 3D is incredibly non-trivial. However, we can start by define the (i,j) plane as any "directed" plane with $\geq i$ points to above or on it, and $\leq j$ strictly above it. Thus we can construct the $(1-\frac{1}{2})$-polyhedron by the collection of all $(i, i-2)$ planes. The $(2-\frac{1}{2})$-polyhedron can be constructed similarly, with the intersection of these two shapes forming the 2-Loop. Finally, the i-polyhedron is formed by the centroids of the plane segments cut off by the $(i-\frac{1}{2})$-polyhedron from the extensions of the sides of the $(i+\frac{1}{2})$-polyhedron.

Difficulties in 3D

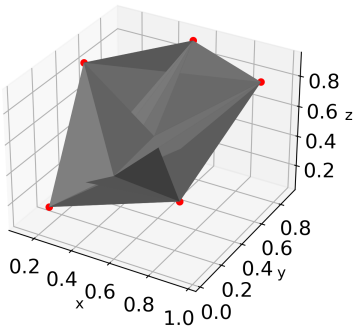
The transition from 2D into 3D poses unique challenges not faced by the original transition into 2D. The most pressing challenge it is impossible to fully convey 3D information on a 2D report / computer screen. Depth can be communicated via coloring and shadows, but some information is always lost. Furthermore, as we are drawing surfaces in 3D, it

Automatic 2D Recreation

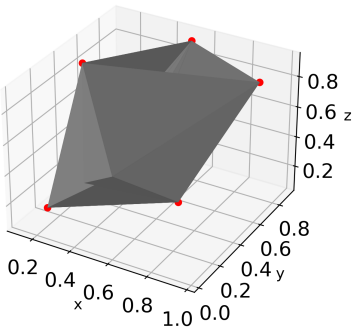
In the original paper, Figure 2 illustrates Tukey's proposed method to extend order statistics into two dimensions. The figure is hand drawn, and can be done algorithmically using any modern programming language. Figure 1 shows the process being done using similar data to the data in the original paper.



(a) Points and their 1-1/2 Polyhedron



(b) The Same Points and their 2-1/2 Polyhedron



(c) The Points and their 2-Loop

Figure 2: 11 points drawn randomly from Unif(0,1), with each dimension being i.i.d. In (c), the 2-loop is obstructed, but remains inside the surfaces plotted.

will always be impossible to see the opposite side of any surface (at least without printing two pictures for a distinct surface). To remedy this, an interactive 3D viewer is available in the code repository. There are also .gif images that rotate over the 3D surfaces.

There are mathematical challenges, too. First, what is 'left' of a plane? I define it as being 'on top of' the plane, with the intention that this direction is usually radially outward from the centroid of the data. The integer polyhedron is also incredibly difficult to program, for a number of reasons. First, you must first find the lines of intersections between any two planes (if they do intersect, but do not overlap). Then you have to extend the planes back from this line of intersection, and find the centroid of these extensions. Finally, one can draw the convex hull of these centroids. As of now, I have not been fully able to implement this construction.

Code

The code created to generate the outputs displayed in this report can be found on my personal GitHub, here.

References

[1] John W Tukey. "Mathematics and the picturing of data". In: *Proceedings of the International Congress of Mathematicians, Vancouver, 1975*. Vol. 2. 1975, pp. 523–531.