Module 6 establishes that clustering is synonymous with distance - records are considered in relation to their distance from other records and clusters are considered in relation to their distance from other clusters. K-means clustering is a specific, partitional form of cluster analysis – noted for its relative ease of computation compared to other methods of clustering. In this method, records are grouped into k clusters based on (again) their distance from those clusters. But how are those distances calculated? As Shmueli’s Data Mining for Business Analytics lays out, there are a number of methods for computing this. Euclidean distance is the most common - in this, we calculate the square root of the sum of all squared differences between two records. Other types include statistical distance, which considers correlation between measurements as part of its calculation; Manhattan distance, which uses absolute differences as opposed to squared; and maximum coordinate distance, which takes into account only the maximum measurement between two given data points.

With more than one way to calculate the distances used in our cluster model, how do we determine which method will work best? Data Mining for Business Analytics tells us that whatever method we do choose has to satisfy a few requirements – namely, that the distance values are non-negative, self-proximate, symmetrical, and that they display the principle of triangle inequality (grab any three records; calculate the distance between any two of them, and that distance should not exceed the sum of the distances between the other two pairs). But beyond just fulfilling those principles, there are a few other things we should consider when picking our preferred equation, such as the scale of each measurement, the presence of outliers in the data, the way measurements should relate to each other, and, perhaps most importantly, the domain that we are working in with our model (more plainly, the what that our data represents).

How might this look in practice? Take the field I work in, Philanthropy. One project I could conceivably work on might be to segment a pool of potential donors into groups that represent the area that they are most likely to donate to – in Healthcare Philanthropy where I work specifically, examples of these areas might be Neurology, Oncology, Cardiology, or Gastroenterology. We then might choose to cluster individuals into these segments based on variables that can include their patient visit history (their number of appointments in each of these areas), and their donation history (the number of and total sum of donations they’ve made to comparable areas at other organizations). Now, what form of distance measurement should we use for this model? On the surface, patient history and donation history don’t have any appearance of correlation with each other. This could indicate that Euclidean distance might be appropriate for this model, as it ignores any relationship between variables. The scale of these variables also does not appear to be an issue; while the number of appointments is measured in single or double digits and the donation sums can be measured in thousands or even millions, these numerical variables can easily be normalized by z-score, meaning that Euclidean distance can still be used even though it is scale dependent. However, these variables could be susceptible to outliers, particularly the sum of donations variable (one large mega-gift of say, $100 million would be more than enough to skew our calculations). Knowing that Euclidean distance can be sensitive to outliers like these, it may be more prudent to use Manhattan distance for our clustering model on account of its emphasis on absolute distance rather than squared distance (which will compound the effects of the outlier values to a much greater degree than an absolute calculation would).

**References**

Shmueli, G., Bruce, P. C., Gedeck, P., & Patel, N. R. (2020). Data mining for Business Analytics: Concepts, techniques and applications in Python. John Wiley & Sons, Inc.