# Introduction

Weighted estimates are considered standard when using spatial data. This is because they take into account the spatial distribution of the data and can compensate for factors like spatial autocorrelation.

Weights can be derived from the original sampling design information used to select the sampled locations included in an analysis. When those original polygons are unavailable, density partitions and Thiessen polygon can provide an alternative way to derive weights.

Density partitioning splits the area of interest being analyzed into discrete polygons with similar levels of sampling, e.g., low, medium, and high density of sampled locations. This is relatively coarse on its own, but does provide weights which reflect the expectation of spatial autocorrelation: points in high density areas will have lower weights and therefore individually lower influence on estimates than points in low density areas.

Thiessen polygons are defined using centroids, single points within the area of interest, and each represents the area which is closer to its own centroid than any other centroid. Centroids can be placed within the area of interest in various ways, but placing them randomly avoids introducing unnecessary bias. Because the centroids may be placed randomly, there is effectively an infinite number of valid sets of Thiessen polygons possible for an analysis. This permits a model-based approach to the analyses in which many different sets of Thiessen polygons are found and used for weighted analyses and then the estimates from those analyses can be combined via bootstrapping to get a single estimate.

Taken together, density partitions and Thiessen polygons can provide a responsible weighting scheme which allows the use of data from multiple overlapping sampling designs without needing specific information about the original sampling frames or spatial stratification definitions.

## Defining parameters

### Density partitions

#### Partition counts

### Thiessen polygons

#### Polygon counts

#### Minimum point counts

#### Number of solutions