**Quantifying Point Patterns**

|  |
| --- |
| 1. Create a folder called **PPA**. On DIA 322 computers, you might want to create this folder in your user Documents folder (e.g. C:\Users\jdoe\Documents\ PPA). On the DIA 222 computers, you might want to create this folder on the D: drive under D:\course number\user name\ (e.g. D:\ES212\jdoe\PPA). 2. Download the [PPA.zip](Point_Pattern_Analysis_files/PPA.zip) data for this exercise and extract the files to your newly created PPA directory. |

This tutorial makes use of a tropical forest census which plots the positions of 3605 trees of the species Beilschmiedia pendula (Lauraceae) in a 1000 by 500 meter rectangular sampling region in the tropical rainforest of Barro Colorado Island. This version of the data was pulled from the [spatstat R package](http://www.spatstat.org).

*Source: Condit, R. (1998)*Tropical Forest Census Plots*. Springer-Verlag, Berlin and R.G. Landes Company, Georgetown, Texas.*

Contents

[Step 1: Density based: Gridded quadrat count 1](#_Toc496874921)

[Step 2: Density based: Tessellated quadrat count 6](#_Toc496874922)

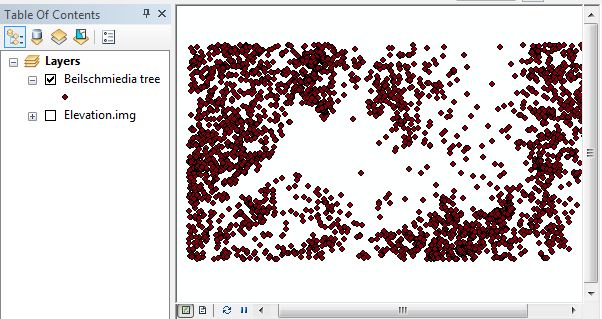
[Step 3: Density based: Point density tool 15](#_Toc496874923)

[Step 4: Density based: Kernel density tool 18](#_Toc496874924)

[Step 5: Distance Based: nearest neighbor tool 20](#_Toc496874925)

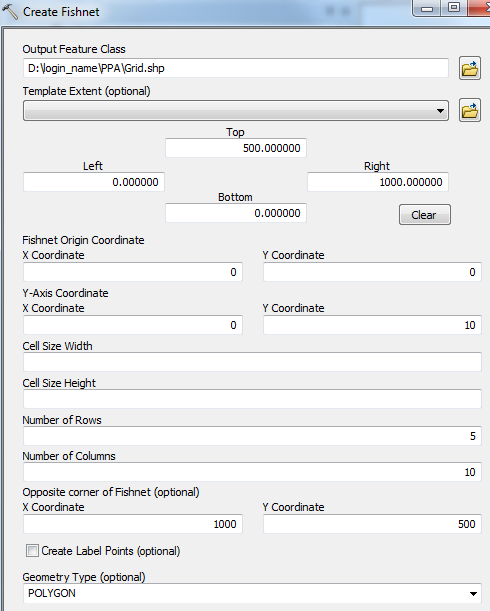
1. Density based: Gridded quadrat count

Open the **Point Pattern.mxd** document.

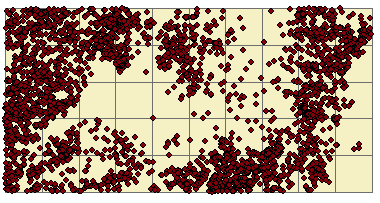


The map consists of a layer, **Beilschmiedia tree**, that plots the location of 3605 trees. It is assumed to be a complete census of trees. The **Elevation.img** layer is a raster of elevation values recorded in meters.

First, we’ll create a uniform grid (**5 rows by 10 columns**) using the **Data Management >> Sampling >> Create Fishnet** tool from ArcToolBox. Make sure to set the grid extent such that the x-axis covers a range of 0 to 1000 (meters) and the y-axis covers a range of 0 to 500 (meters). Set the output type to **Polygon** and name it **Grid.shp**.

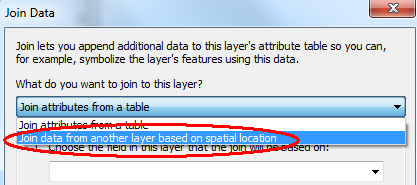


Note that the map’s coordinate system is not explicitly defined, but as long as we know that the trees were recorded in a local Cartesian coordinate system that preserves distance, we should be fine.

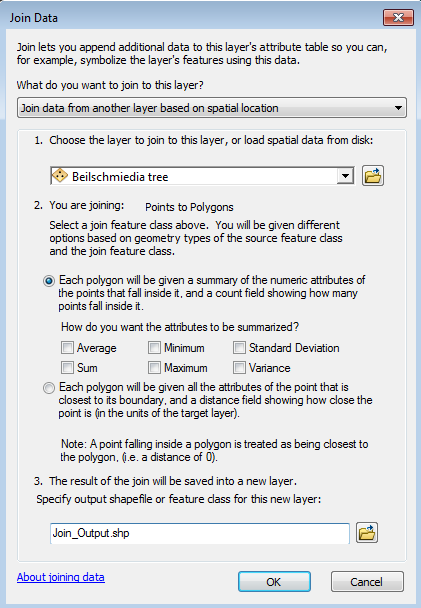


Next, we will tally the number of trees within each grid cell.

**Right-click** on the grid.shp layer and select **Joins and Relates >> Join**. In the past we’ve use this tool to join tables, here, we’ll use the spatial join option instead.

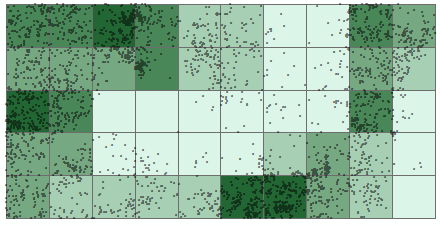


The layer whose features will be tallied in the grid.shp layer is the **Beilschmiedia** tree layer. By default, this tool will compute the total number of points in each grid cell so we do not need to compute any other summary. We’ll name the output **Join\_Output.shp**.



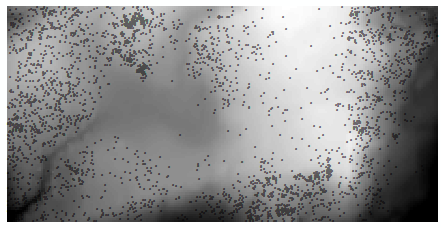
The output shapefile has a new field called **Count\_** that tallies the points in each polygon.

Change Join\_Output’s symbology to reflect the **Count\_** values.

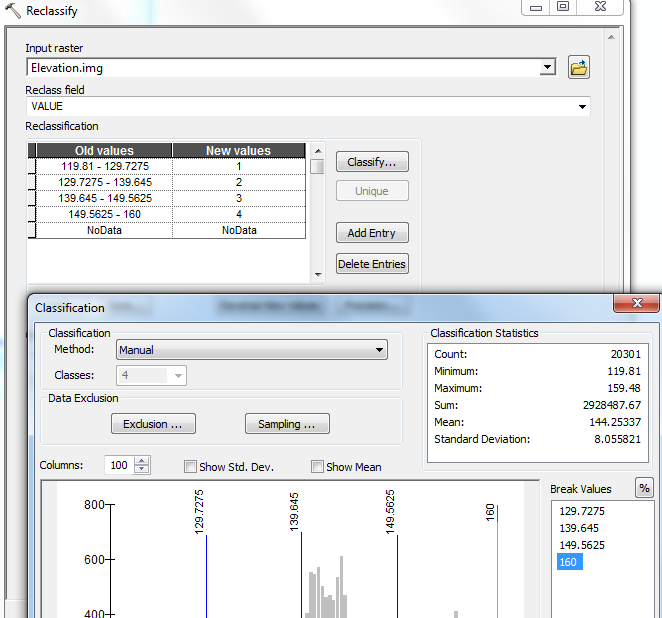


1. Density based: Tessellated quadrat count

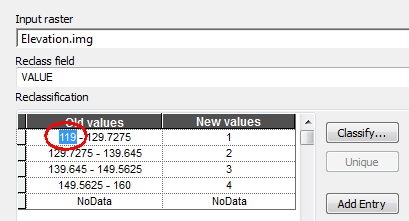
In the last step, you created quadrats from a uniform grid. Quadrats need not be uniform in shape and in size. In this step, you will create quadrats based on elevation intervals for the purpose of counting the number of trees at different elevation ranges.



We’ll first reclassify the raster into **four** elevation intervals (each classification bin covering an equal range) using **Spatial Analyst Tools >> Reclass >> Reclassify.** We will create equal interval classes so make sure to select the **Equal Interval** method from the **Classification** window. You will probably want to bump the upper range value to 160 to ensure that *all* input values are reclassified.

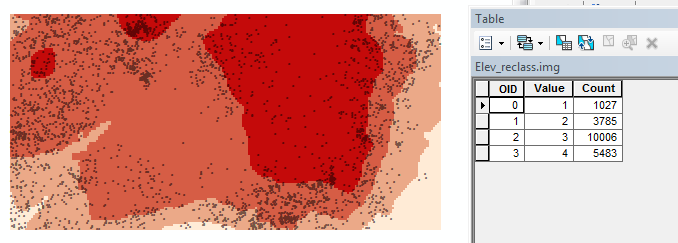


Before running the Reclassify tool, we might want to set the lower bound to a smaller value like 119, to make sure that we are not excluding values due to rounding errors. This can be done directly in the reclassification table.

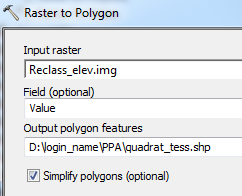


Name the output **Reclass\_elev.img** then click **OK** to run the geoprocess.

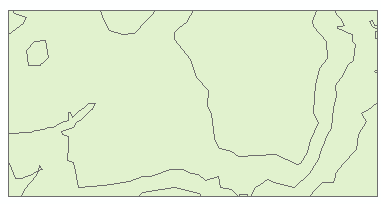
Next, we’ll convert the categorical raster to a vector layer. Note that this new raster has an attributes table whereas the original raster did not. This is to be expected since the newly created raster has just 4 unique integer values (ArcMap will not create a raster attributes table if there are too many unique pixel values).



In ArcToolbox, select **Conversion Tools >> From Raster >> Raster to Polygon** and populate the fields as follows:

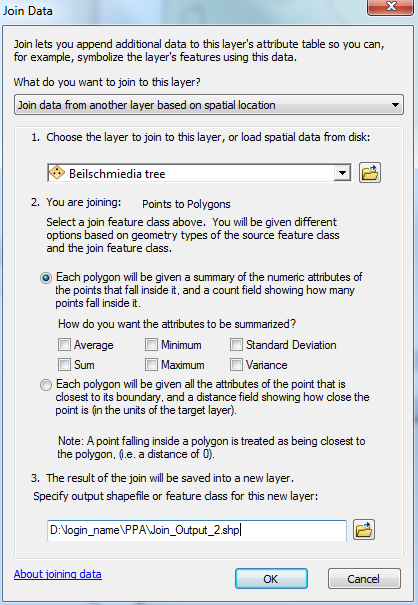


The output is a polygon vector layer.



Next, we’ll tally the number of points in each tessellated surface using the *spatial join* workflow described earlier.

Right-click the quadrat\_tess layer and select **Joins and Relates >> Join**. Populate the fields as follows:

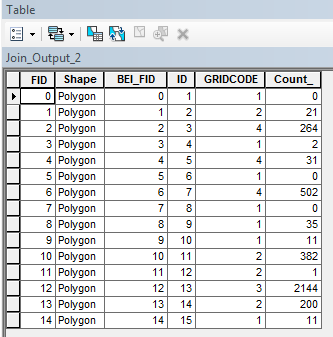


Click **OK** to perform the spatial join.

Symbolize the new layer using the **Count\_** field.

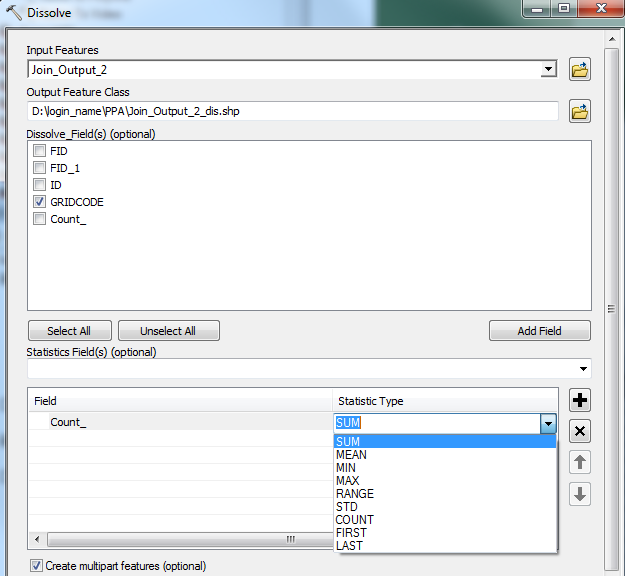


If you look at the attributes table, you’ll note the one-to-one relationship between polygons and attribute values.



We will want to reduce the number of records to just four (one for each unique elevation interval) using the dissolve tool. Note that the elevation interval value is named GRIDCODE (this is the default vector attribute name when converting from raster to vector).

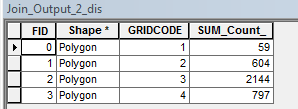
From the **Geoprocess** pull-down menu select **Dissolve** and populate the fields as follows:



In the statistics field, we choose to **SUM** the **Count\_** values by unique elevation intervals.

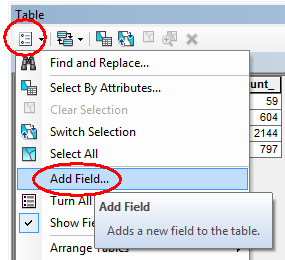
Click **OK** to dissolve the layer.

The new shapefile now has just four records (a one-to-many relationship).

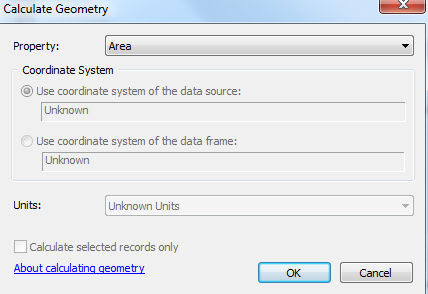


We could compare the counts between all four elevation intervals however, the area of each interval may not be the same (think of the modifiable aerial unit problem). So we should normalize the count to area. This will require that we create two new attribute fields: **Area** and **Density** (both data types stored as **Float**).

Add two new fields from **Add Field** option. Make sure to set their types to **Float**.

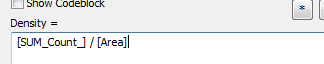


For the Area field, compute its area via **right-clicking the column header >> Calculate Geometry**.



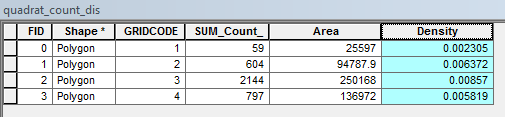
Note that the coordinate system is in meters (even though it’s not explicitly defined in the layer’s CS).

For the Density field, type in the expression [SUM\_Count\_]/[Area] via **right-clicking the column header >> Field Calculator**:



Click **OK** to compute the values.

The attributes table should look like this:

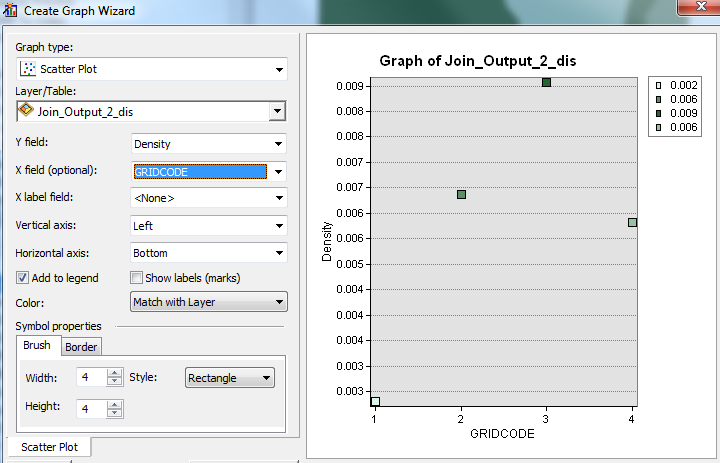


**Symbolize** join\_Output\_2\_dis using the **Density** Attribute.



Finally, we can make use of ArcGIS’ rudimentary chart tool to plot count vs elevation range.

From the **View** pull-down menu select **Graphs >> Create Graphs** and populate the fields as follows:



The plot suggests a peak tree density at the third elevation interval (elevation of 139.6 m to 149.6 m, roughly). Note that the point symbol colors match those of the input shapefile. This facilitates matching plot points to map polygons.

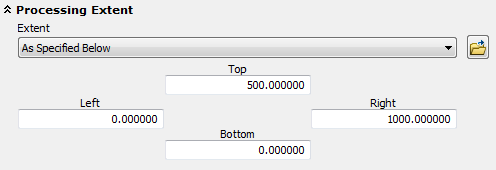
1. Density based: Point density tool

In this step we’re creating a density map of tree counts.

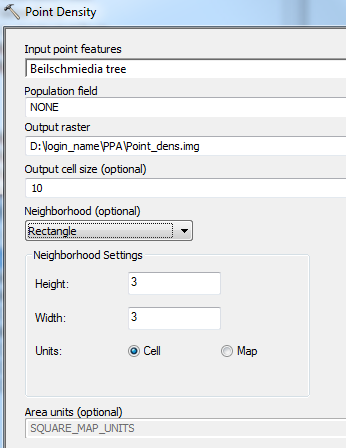
In ArcToolbox open **Spatial Analyst Tools >> Density >> Point Density.**

Since the geoprocess will create a new raster from a vector layer, it may not be a bad idea to explicitly define the raster extent since the tool will default to the smallest rectangle encompassing the input point layer.

Open the **Environments** setting and define the extent as outlined below:

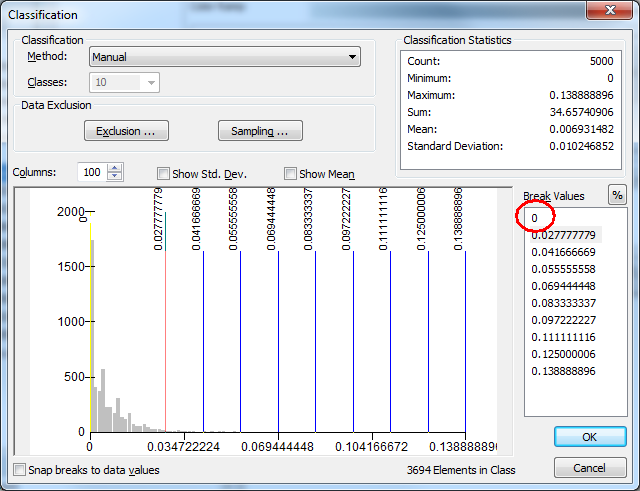


Click OK to close the **Environments** setting and populate the **Point Density** tool fields as follows:

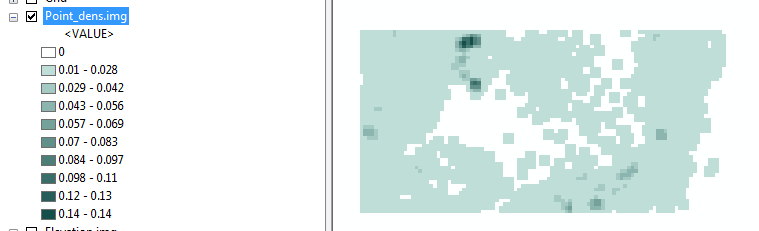


Click **OK** to run the geoprocess.

Set the new raster’s symbology classification scheme to **equal interval** (10 classes) and assign *No Color* to pixel values of 0.

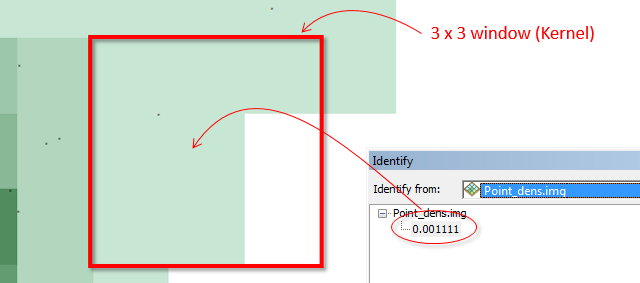


Your output should look something like this.



Each pixel is assigned the number of points within a 3x3 pixel search window then divided by the area of that search window. For example, if a cell has one point inside the search window, its output value will be:

1 point / (9 cells \* 100m2 area per cell) = 0.0011 points per square meter (within a 3x3 search window)

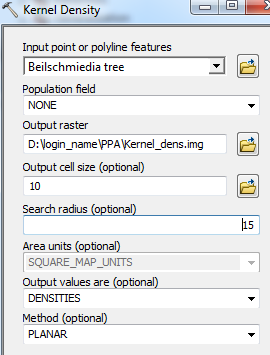


Note that the point density function is a *focal operation*; behind the scenes, ArcMap converts the point layer to a raster before computing the output density values.

1. Density based: Kernel density tool

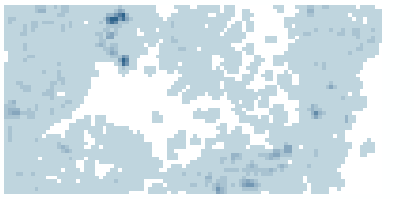
ArcMap offers two density tools: a point density (covered in the last step) and a kernel density tool. This would seem to suggest that these tools perform distinct operations when in fact the point density tool is simply a special case of a kernel density function whereby all input values in the focal operation are assigned equal weight. ArcMap’s Kernel density tool applies a quartic function that assigns a non-uniform weight to each point based on its proximity to the output cell. This tool tends to generate smoother density rasters.

In ArcToolbox open **Spatial Analyst Tools >> Density >> Kernel Density.** Set the extent in the **Environments** window as outlined in the previous step then populate the fields as follows:



Note that this tool only allows you to define a kernel by its search radius and not by cell grids. Here, we use a 15 m search radius.

Next, generate a symbology scheme similar to the one used in the last step (make sure to set 0 pixel values to no color).

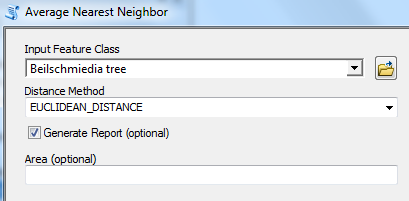


Note the smoother appearance of density values.

1. Distance Based: nearest neighbor tool

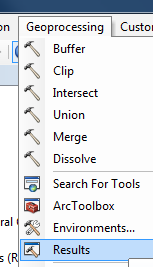
So far, we’ve explored density based approaches to quantify point patterns. Density based analysis usually focuses on a point pattern’s [first order property](https://mgimond.github.io/Spatial/point-pattern-analysis.html#first-and-second-order-effects)—i.e. its distribution vis-à-vis location. Another property of interest is a point pattern’s spatial interaction, a [second order effect](https://mgimond.github.io/Spatial/point-pattern-analysis.html#first-and-second-order-effects). A statistic that can be used to quantify a point pattern’s second order property is the average nearest neighbor (ANN) statistic.

In ArcToolbox, open **Spatial Statistics Tools >> Analyzing Patterns >> Average Nearest Neighbor** and populate the fields as follows (make sure to check off **Generate Report**).

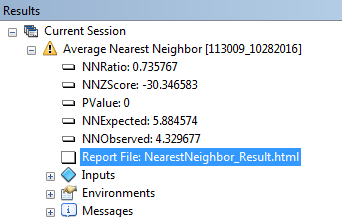


Click **OK** to run the geoprocess.

The output is not a geoprocess but a report saved as an HTML file. To view this file, open the **Results** tab from the **Geoprocessing** pull-down menu.



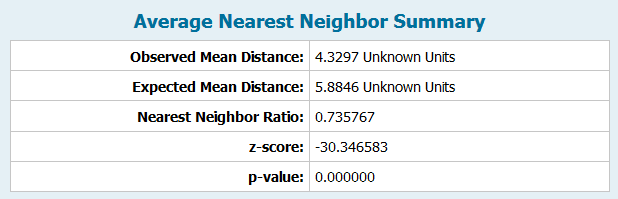
In the Results tab, expand **Current Session >> Average Nearest Neighbor** (if you ran more than one ANN analysis, pick the top-most instance).



You might notice a yellow warning symbol next to the geoprocess. It’s simply indicating that the tool does not recognize this layer as being in a projected (Cartesian) coordinate system. Recall that the file has no defined CS, but it is assumed that the tree locations were located on a planar coordinate system. This is a good reminder that this tool measures planar distances and *not* geodesic distances.

Double-click on the **Report File:NearestNeighbor\_Result** link.

You will ignore the infographic at the top of the report (this tutorial does not cover hypothesis testing) and focus on the first row of the summary table which gives us the average distance between all nearest neighbor combinations.



The coordinate system was never explicitly defined in this layer so the software defines the units as “unknown”. This is fine since we know that the units are in meters. The output indicates that the average distance between first order neighboring trees is about **4.3 meters**.

© Manny Gimond, 10/27/2017