

Spatial Pattern

In this tutorial we are going to deal with recognizing one of the most basic patterns of a plant population: whether individuals are spatially closer or further apart than would be expected if they were simply randomly distributed (i.e., the location of an individual does not improve the prediction of where other individuals may be)).

Target

Investigate the spatial pattern in plant populations and discuss what underlying processes could generate the observed patterns. First of all, however, we need to define some concepts.



Context

A spatial pattern is a predictable structure that can be detected and quantified. In general, a pattern is considered to be a different structure than a random one, however, in the case of spatial patterns (and others as well) the random pattern can also be considered a pattern, after all it has some



predictability (eg in relation to the average number of individuals)) and can be detected and quantified. There are several metrics used to describe the distribution of individuals that are able to differentiate, with greater or lesser efficiency, the three basic spatial patterns: random, homogeneous and aggregated.

Spatial Patterns

- random: the distribution of individuals is not different from what would be expected by a random distribution;
- regular or homogeneous: individuals are regularly spaced. It is also called dispersed pattern, as it is related to the greatest possible distance between individuals;
- aggregate: individuals are closer than expected by a random pattern, forming clusters.

Detecting a spatial pattern can be important both for understanding the mechanisms that generate the pattern, for deciding on the sampling method and scale, and for planning the management of a population. Some desirable properties of a spatial pattern measure are:

- clearly differentiate the pattern;
- not be affected by: sample size, population density, or variation in sample size and shape;
- be statistically tractable: able to calculate the uncertainty of the value and test for differences between samples.

For this practice we will use a point randomness estimate called K-Ripley. First we will use simulated distribution data with different patterns and then use the same technique to detect the spatial pattern in a natural population.

Tutorial

Multiscale Patterns

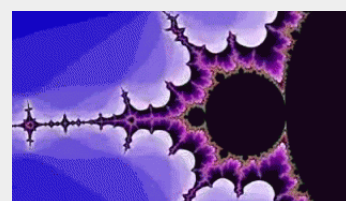


In this practice we will quantify the spatial pattern using a multiscale method. Multiscale methods allow, with a single metric, to assess how the spatial pattern varies with scale. We will describe the spatial pattern for the total set of individuals in a population in a delimited area and we will assess the pattern from the scale of the neighborhood of individuals to the wider scale of the population.

For practice we will use a little program called [Programita](#), made by researcher Thorsten Wiegand to quantify spatial patterns using multiscale measures based on distance between points. To download the **Programita** manual click

[here](#).

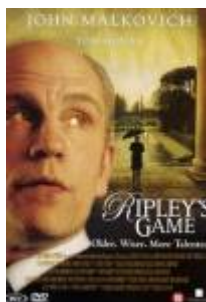
In **Programita** there are several measurements that can be used to



calculate the spatial pattern, we will use two of them: Ripley's **L** and **O-ring**.

Both are point-based approaches, which use the calculation of point-to-point distances within a bounded area. These measures can be used for univariate analyses, that is, identifying the pattern for a single class of points, or for bivariate analyses, which identifies the pattern between two types of points. Bivariate analyses can be used in the context of populations to verify whether individuals of a given stage are spatially associated with another, or in the context of community structuring to analyze whether there is attraction or repulsion in the occurrence of one species in relation to another.

Ripley's K



Ripley's K is a measure of the average density around each point. For each point in the study area, the density inside a circle of radius r centered on the point (gray area of the figure) is calculated. Then, an average of these values obtained for all points is calculated.

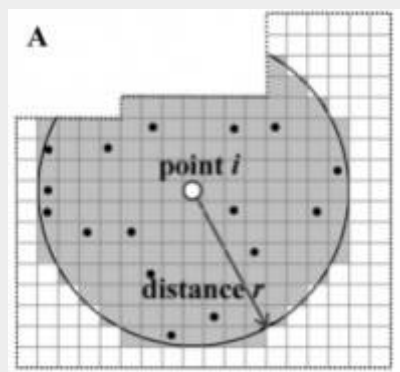


Figure: Implementation of Ripley's L statistic: counting the number of points far from i inside the circle of radius r . Taken from Wiegand & Moloney (2004).

The operation is repeated for different r values, which allows continuous evaluation of the K value for different scales.

$$K(r) = \frac{\sum_{i \neq j} I(d_{ij} < r)}{n} \frac{1}{\lambda}$$

Where:

- d_{ij} is the distance from point i to point j ;

- $I(\{d_{ij} < r\})$ indicator function, being 1 if the point j is at a distance less than r from the point i and 0 if the point j is outside this radius r around i ;
- n is the total number of points;
- λ is the density of the points.

The visual interpretation of K is not very intuitive as it is a cumulative function associated with the area of the circle relative to r . Ripley's L , in turn, is the standardization of this value:

$$L_{\{r\}} = \left(\sqrt{\frac{K_{\{r\}}}{\pi}} - r \right)$$

This transformation makes the value of L for a completely random distribution always be 0 and $L > 0$ indicates aggregation, while $L < 0$ indicates homogeneous pattern.

O-ring (O(r))



The **O-ring** stat is similar to Ripley's L , but based on a ring rather than a circle. It is measured by counting the number of points in a ring of radius r and fixed width. In the same way as L-Ripley, the intensities for different ring sizes are also calculated, keeping the width fixed.

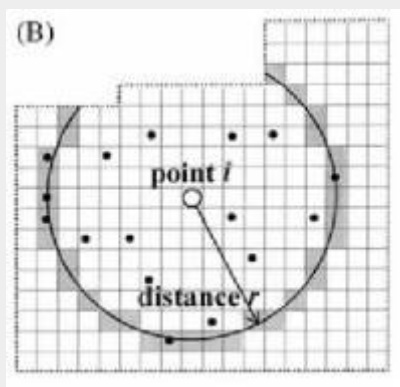


Figure: Implementation of the *O-ring* statistic: count of the number of points far from i along the radius r . Taken from Wiegand & Moloney (2004).

So we define $O(r)$ as: $O_{\{r\}} = L_{\{r\}} - L_{\{r-l\}}$

Where:

- $r-l$: is the radius minus the width of the ring ¹⁾

In complete spatial randomness $O(r) = \lambda$ (pattern intensity), when the pattern is aggregated $O(r) > \lambda$ and when it is homogeneous $O(r) < \lambda$

The measures $K_{\{r\}}$, $L_{\{r\}}$ or $O_{\{r\}}$ present theoretical analytical solutions for the pattern defined as Poisson process or Complete Spatial Randomness (CAE). That is, when the distribution of points in the studied space is not different from what was expected by chance. For a given density of points we were able to calculate these theoretical values for any radius. Thus, to interpret the spatial pattern of the observed points we need:

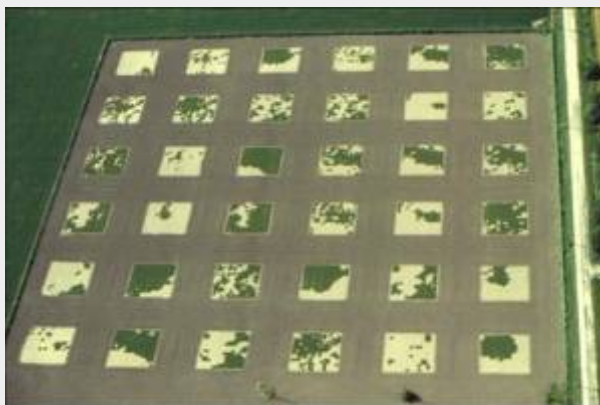


- calculate observed and theoretical values for CAE;
- compare these values;
- define when a difference is acceptable or not to claim that the pattern is different from random;

For the first two topics above, we used the formulas and calculated the values. To take the subjectivity out of the third, we can calculate confidence intervals or generate confidence envelopes²⁾ of confidence by computer simulations, to objectively define what is something different from what is expected for the CAE.

Simulted Points Pattern

Activity 1:



Which process created this points pattern?

General Instructions

- 1. Download the files related to spatial pattern 1 or 2. If open a page showing the data, right-click the link to save the file to your computer:

Data for Spatial Analysis

Default 1

- Population (all individuals, without distinguishing adults and young individuals)
- Parents (adults only)
- Young (young individuals only)
- Bivariate (all individuals, distinguishing adults and youth)

Default 2

- Population (all individuals, without distinguishing adults and young individuals)
- Parents (adults only)
- Young (young individuals only)
- Bivariate (all individuals, distinguishing adults and youth)

- if you don't have programita installed, download [programita here](#) in the same folder as the data file;
- unzip the file *programita.zip*;
- 2x click to open the executable file ProgramitaJulio2006.exe.

Welcome to **Programita**! Now let's open the data that we are going to work with.

Programita accepts text files with .dat and .asc extensions. They are files in text format, separated by tabs (or spaces). The data file has the following structure:

The first line contains general information about the data file:

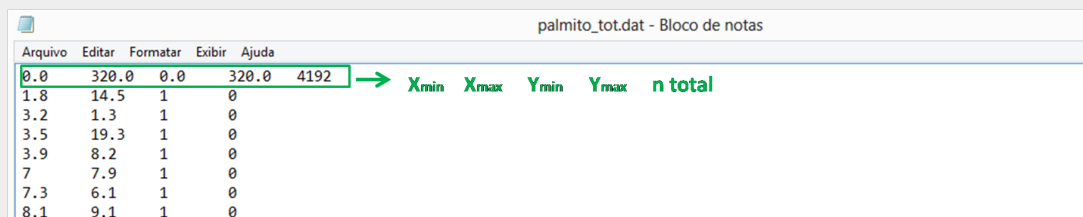
- minimum value of x;
- maximum value of x;
- minimum value of y;
- maximum value of y; and
- total number of individuals

From the second line on, are the data of the points that will be analyzed:

- first column with the x coordinates of the individuals;
- second column with the y coordinates of the individuals;
- third column with points of pattern 1 identified by 1 and pattern 2 by 0³⁾;
- fourth column with the points of pattern 1 identified by 0 and pattern 2 by 1⁴⁾.

In the case of univariate data, the third column will always be 1 and the fourth column will always be 0. For bivariate data the third and fourth columns will have values of 0 and 1 according to the point

pattern.




	X_{min}	X_{max}	Y_{min}	Y_{max}	n total
0.0	320.0	0.0	320.0	4192	
1.8	14.5	1	0		
3.2	1.3	1	0		
3.5	19.3	1	0		
3.9	8.2	1	0		
7	7.9	1	0		
7.3	6.1	1	0		
8.1	9.1	1	0		

Fig. Example of a .dat file in the format used in *Programita*.

Univariate Pattern: all points

- 1. Check if the *Input data file* window is showing the .dat files. If not, check that the programita executable file is in the same folder as the .dat files.



Depending on your browser configuration, the saved file may appear with a different extension (eg ".bin"). In this case it is necessary to change the file extension to ".dat".

- 2. in the menu on the left select the file **padrao"0X"all.dat**. In the case **X** will be 1 or 2 depending on your choice;

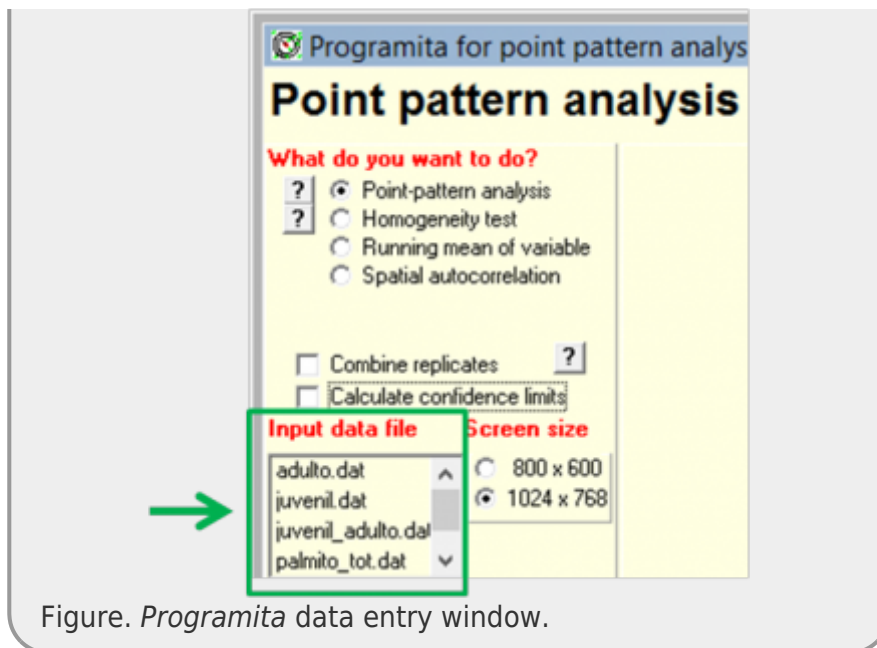
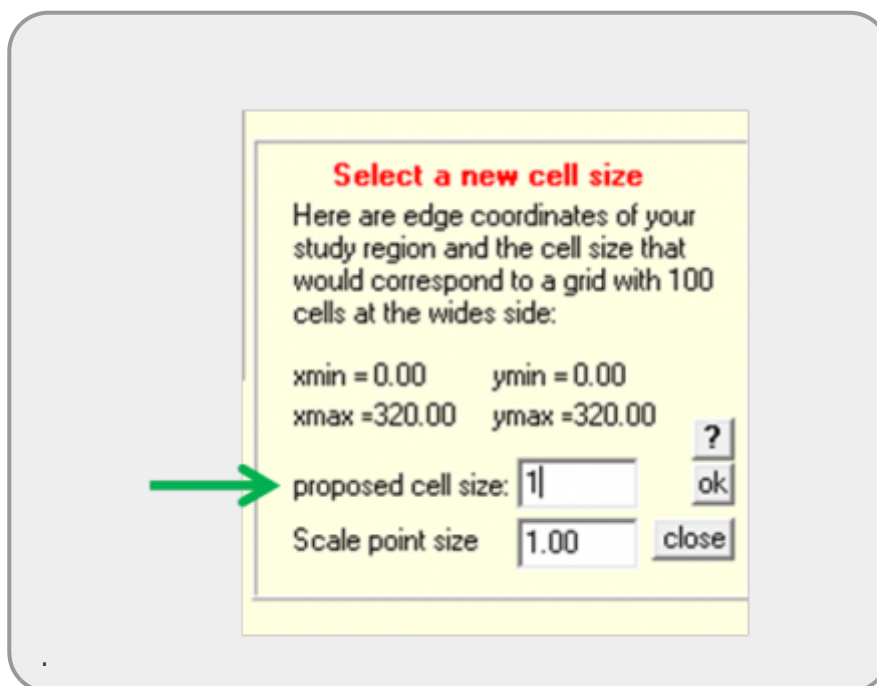


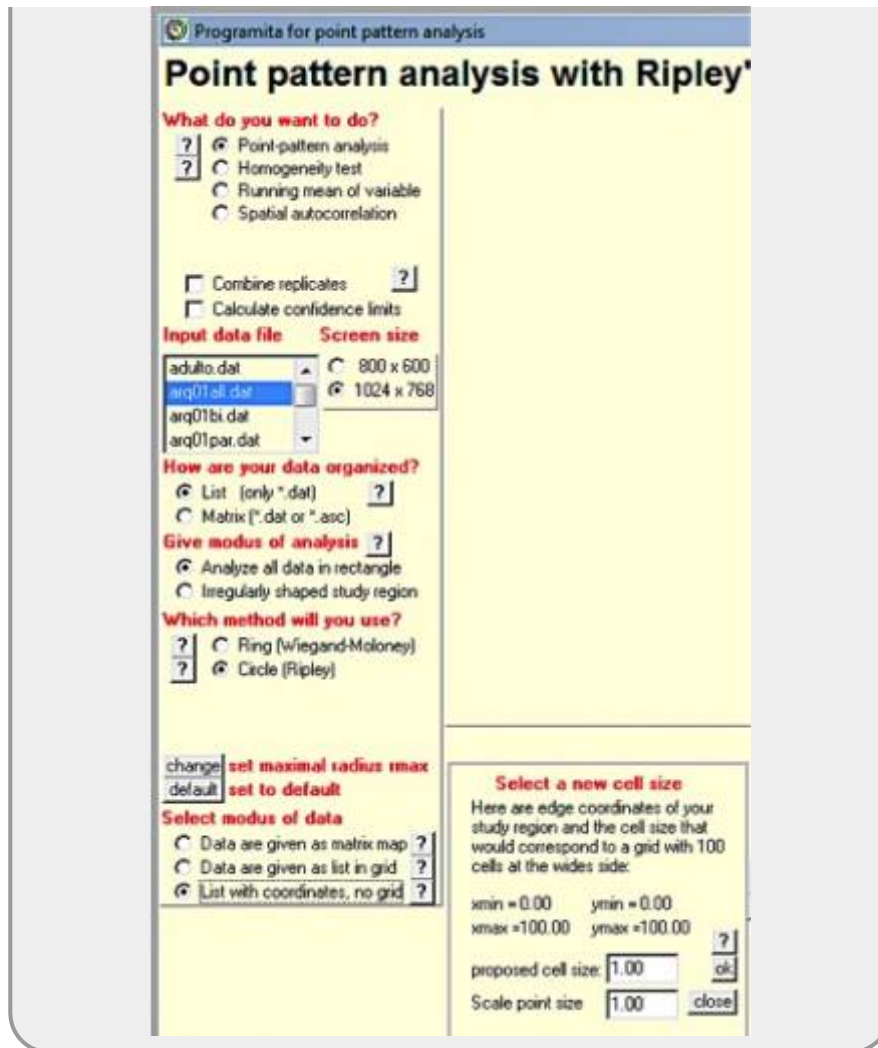
Figure. Programita data entry window.

- 3. In **How your data are organized** select **List**
- 4. Let's start using Ripley's L so in **Which method to use** select **Circle**
- 5. In **Select modus of data** select **List with coordinates in the grid**. When selecting this option, a window will appear with the option **Select a new cell size**:



- 6. If you have less than 500 points, change the **proposed cell size** to 1. Otherwise, leave it at the program's default.
- 7. Done all that, you should be like this:





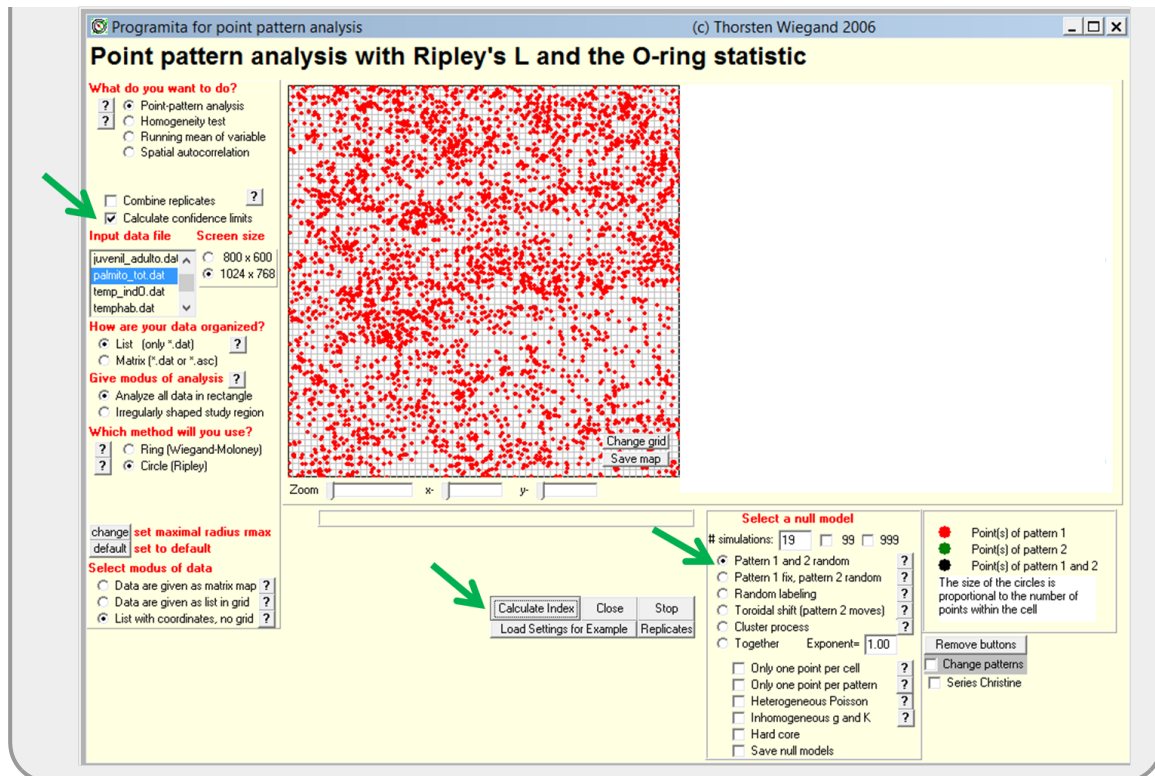
- 8. You can now take a deep breath and hit the **Calculate index** button;

The program's visual output is a map where individuals appear in red dots, following the coordinates of the data file. The graph in the upper right corresponds to the L-Ripley value for different radii. In this graphic output it is possible to analyze how the spatial pattern varies according to the scale.

However, this is not enough to state at what scales the population is aggregated. For this we need to compare the observed result with the pattern that would be generated by the completely random distribution of points. This null model is called **complete spatial randomness**. To generate this model by simulation it is necessary to relocate the same number of points randomly in the same area. If we do this over and over again, it is possible to generate a confidence envelope (similar to the confidence interval) in which the random distribution pattern is found. If the observed values are contained within the envelope we can conclude that our pattern is not different from the random one.

To do this you must:

- 9. select the option **Calculate confidence limits** and;
- 10. in the window **Select a null model** select the null model **Pattern 1 and 2 random**;
- 11. Make sure your screen looks like the picture and click the **Calculate index** button again.



If the simulation is taking too long

- hit the *stop* button next to the *Calculate index*;
- select another “*modus of data*” and then select again *list with coordinate,...*;
- in the *Select a new cell size* window, change *proposed cell size* to 2;
- in the window *Select a null model* change # *simulations* to 20;
- press the *Calculate index* button again;

Describe the observed pattern

Programita allows you to graphically follow the simulation over time 😊. It is possible to observe that at each simulation a random distribution of individuals is generated and the L-Ripley values are recalculated. At the end, the graph is generated with the observed values from the data file, accompanied by the confidence envelope generated from the simulation of complete spatial randomness. Values outside the confidence interval indicate the existence of a spatial pattern other than random.

Tip: Make a **Print Screen** of your results to save the graph of each analysis

you do throughout the practice.

- 12. Do the same procedure, but in **Which method to use** select **Ring**
- 13. Compare the results between the L-Ripley and the O-Ring.

Activity

- repeat parsing for files with:
 - the points of the parents (adults): *pattern"0X"par.dat* e;
 - the points of the associated points - offspring (youth): *pattern"0X"prole.dat*;
- interpret the result for each type of point;

Bivariate Pattern: two classes of points

Programita allows the analysis of point pattern of one class in relation to another. For this, it is necessary to differentiate the points in the data file, using 0 or 1 in columns 3 and 4, as shown in the figure below, in a file that distinguished adult from juvenile individuals:

Arquivo	Editar	Formatar	Exibir	Ajuda
0.0	320.0	0.0	320.0	4192
1.8	14.5	1	0	
3.2	1.3	1	0	
3.5	19.3	1	0	
3.9	8.2	1	0	
7	7.9	0	1	
7.3	6.1	1	0	

↑ ↑
adulto juvenil

Let's now analyze the pattern of the associated points (PROLE) in relation to the parents (PAR), following the same procedure as before.

- 1. select the file with the separation of classes of parent and associated points: *pattern"0X"bi.dat*;
- 2. under **What do you want to do** select the option **Point-pattern analysis**
- 3. in **How your data are organized** select **List**

- 4. in this case, we are interested in the cumulative scale pattern analysis to understand how far there is aggregation, so in *Which method to use* select *L-Ripley*
- 5. in **Select modus of data** select **List with coordinates in the grid**
- 6. to test whether there is aggregation of PROLE points in relation to PAR , we will use the confidence envelope. select option **Calculate confidence limits** and select null model **Pattern 1 fix, 2 random**.
- 7. run the analysis by pressing: **Calculate index**
- 8. Interpret the results.

Discover the algorithm

Algorithm is a sequence of steps to perform a task. The points of the data files were generated by a very simple algorithm in two phases: first the parental points were generated and then the associated points (offspring). Describe a sequence of tasks ⁵⁾ which would be able to generate the point distribution (including both point classes) that you observed from your data file.

Distribution of Palm Hearts in the Forest



The palm heart tree (*Euterpe edulis* Mart.) is a very characteristic species of the Atlantic forests and usually occurs with high densities in more preserved areas. We will now analyze the data referring to a population of palm hearts that occurs in a portion of Restinga forest on Ilha do Cardoso, Brazil. Data

were collected in the years 2009/2010 in an area of 10.24ha (320m x 320m).

We prepared three files in the format read by *Programita*:

1. data from juveniles (trunk diameter between 1 and 5 cm):
juvenil.dat
2. data from adult individuals (trunk diameter > 5 cm):
adulto.dat
3. juveniles and adults (default 1 adult, default 2 juvenile): [juvenil_adulto.dat](#)

Using the tools available in *Programita* to describe spatial patterns:

- of the total population of palm hearts;
- juveniles only and;
- adults only.

Investigate whether the distribution of juveniles is associated with that of adults.

Patterns & Processes Get together in a group of 2-4 students and discuss what possible processes could generate the patterns described.

1)

equal to the inner radius of the ring

2)

equivalent to confidence interval obtained by numerical simulation

3)

in the case of bivariate data

4)

also in the case of data with two types of points

5)

eg: generate 10 x values from a uniform random distribution from 0 to 100; generate values from a sequence of 10 to 90 at every interval of 5 as y....

From:

<http://ecovirtual.ib.usp.br/> -

Permanent link:

http://ecovirtual.ib.usp.br/doku.php?id=en:ecovirt:roteiro:pad_spat



Last update: **2022/09/15 13:35**