

Practical session: R library spatstat

Load the library: `library(spatstat)`

Examples of point pattern data:

- `amacrine` (rabbit amacrine cells, locations and 2 types)
- `anemones` (sea anemones data, locations and sizes)
- `ants` (ant nests data, locations and 2 types)
- `bei` (tropical rainforest trees, locations)
- `betacells` (cat retinal ganglia data, locations, 2 types and sizes)
- `bramblecanes` (Bramble Canes data, locations and 3 types)
- `cells` (biological cells data, locations)
- `chorley` (cancer data, locations and 2 types)
- `finpines` (Finnish Pines data, locations and 2 size measures)
- `hamster` (hamster tumour data, locations and 2 types)
- `japanesepines` (Japanese Pines data, locations)
- `lansing` (Lansing Woods data, locations and 6 types)
- `longleaf` (Longleaf Pines data, locations and sizes)
- `nztrees` (trees data, locations)
- `ponderosa` (ponderosa pine trees data, locations)
- `redwood` (redwood saplings data, locations)
- `spruces` (Spruce trees in Saxonia, locations and sizes)
- `swedishpines` (Swedish pines data, locations)
- `urkiola` (Urkiola Woods data, locations and 2 types)

Task 1: Choose some of the point pattern data above and use `summary` to print a summary of the point pattern. Plot the data by using `plot` or `plot.ppp`.

Choose a few point patterns which you would like to investigate further. Pick, for example, one pattern that looks regular and one that looks clustered, and one multitype pattern and one marked pattern (with size information).

Task 2: Estimate and plot the summary statistics F, G, J and K by using `Fest`, `Gest`, `Jest`, and `Kest` or `allstats` (`allstats` works only for unmarked data, the marks can be ignored by using `unmark.`) Try also the functions `Lest` and `pcf`. Use `help` to see a description of these functions.

For multitype point patterns, (e.g. `urkiola` or `lansing`), you can plot the functions first for the whole pattern and then separately for each type, and compare them. Try to explain the differences you see (if you see any).

Task 3: To test complete spatial randomness: function `envelope`. Take some of the data sets and try this function. Plot the result. You can try different summary functions (default is the K function).

Note that the envelopes are pointwise envelopes.

Task 4: Fit a Matern cluster model for one of the clustered point patterns by using the function `kppm(X, clusters="MatClust")` (`X` is the point pattern).

You can save the result e.g. by writing `modell1<-kppm(X, clusters="MatClust")`. In the results, `kappa` is the intensity of the parent process and `r` is the cluster radius.

Use `plot.kppm(modell1)` to plot the estimated K function for the data and for the fitted model, and `plot(envelope(modell1))` to check the goodness-of-fit of the model (envelopes are based on the estimated model).

You may also want to simulate the fitted model by using `simulate(modell1)`.

You can try also other cluster processes, e.g. Thomas process (`clusters="Thomas"`), where the daughter points are distributed according to an isotropic Gaussian kernel centered in the parent point. In the results `kappa` is the intensity of the parent (Poisson) process and `sigma2` is the variance of the Gaussian distribution. The log Gaussian Cox process (`clusters="LGCP"`) is another alternative.

Task 5: To fit a Strauss model for one of the regular point patterns, one can use the function `ppm(X, ~1, Strauss(r1))` (`r1` is the interaction radius you have to fix). You can play around with different `r1` values. You can also try to use the summary functions K, L or the pair-correlation function to decide which `r1` to use.

You can save the result by writing `model2<-ppm(X, ~1, Strauss(r1))` and use `plot(envelope(model2))` to check the goodness-of-fit of the model.

Simulate by using `simulate(model2)`.

You can also try a Strauss model with a hard-core (`Strauss(r1)` replaced by `StraussHard(r1, hc)`) (or e.g. `LennartJones`, use help to see description of other models).

Task 6: For multitype data, you can estimate and plot the summary functions separately for each type.

Use [help](#) to see description of [Gcross](#), [Kcross](#), and [Jcross](#) (if only two types of points) or [Gmulti](#), [Kmulti](#) and [Jmulti](#) (if more than two types of points, specify the types you want to study). Apply these functions to some of the data sets to see whether one would suggest any dependence between points of two types either by just plotting the result or by using [envelope](#).

Task 7: For marked data, you can estimate and plot the mark correlation function by using [markcorr](#). Use [help](#) to read about the function. Use also [envelope](#).