Topic 2: Studying Poisson process

In this topic we we will further investigate properties of spatial point processes. In particular, we consider the Poisson model and the following related data analysis methods:

- **Fitting a Poisson process** to spatial data in R.
- Introduction to **marked Poisson processes**.
- **Manipulating marks** in R.
- Investigation of **intensities of marked Poisson processes** in R.
- Species interactions and **cross-type pair correlation functions**.

Fit model for spatial process

To fit Poisson processes in SPATSTAT the intensity function $\lambda(u)$ must be loglinear in the parameter $\boldsymbol{\theta}$:

$$\log \lambda_{\boldsymbol{\theta}}(u) = \boldsymbol{\theta} \cdot S(u),$$

where $S(u)$ is a real-valued or vector-valued function of location u. The form of S is arbitrary so this is not much of a restriction. In practice $S(u)$ could be a function of the spatial coordinates of u, or an observed covariate, or a mixture of both.

To fit the homogeneous Poisson model we use:

```
> library(spatstat)
> data(bei)
>
> fit<-ppm(bei, ~1)
```

```
> fit
Stationary Poisson process
Intensity: 0.007208
Estimate       S.E.    CI95.lo
log(lambda) -4.932564 0.01665742 -4.965212
CI95.hi Ztest      Zval
log(lambda) -4.899916   *** -296.1182

> cdf.test(fit, function(x, y) {y})

Spatial Kolmogorov-Smirnov test of CSR in two
dimensions
data:  covariate function(x, y) { evaluated at points of bei
    and transformed to uniform distribution under CSR
    D = 0.10415, p-value < 2.2e-16
    alternative hypothesis: two-sided
```

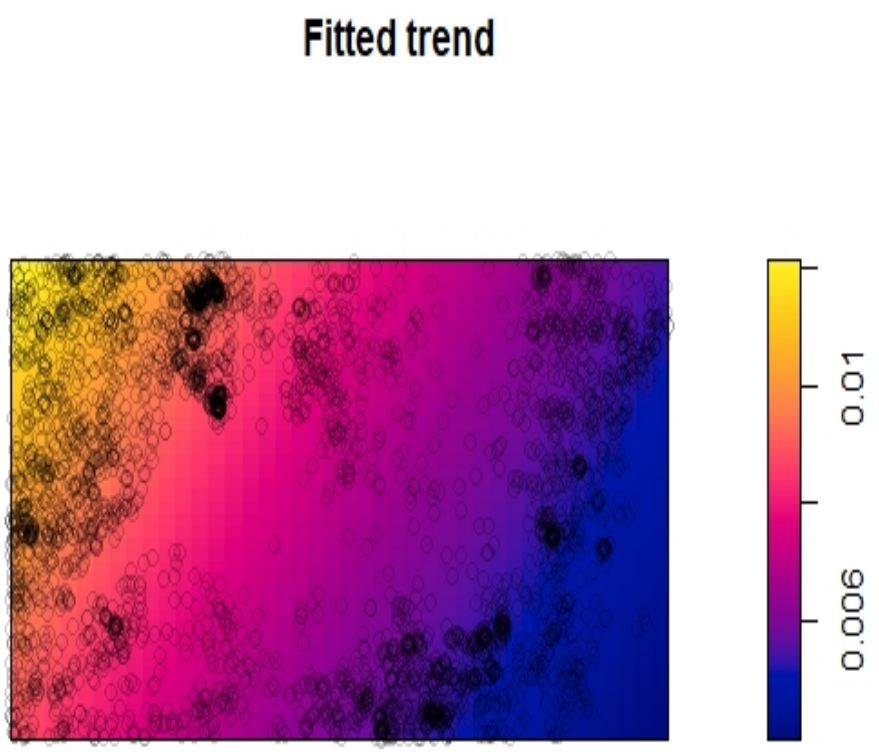
To fit an inhomogeneous Poisson model with the intensity

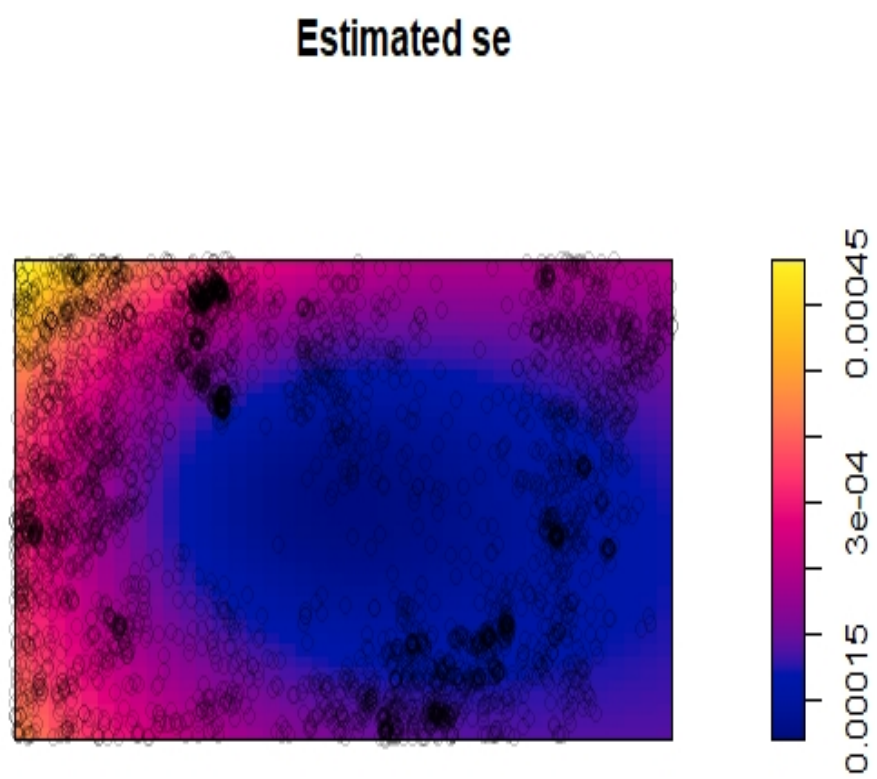
$$\lambda_{\boldsymbol{\theta}}(x, y) = exp(\theta_0 + \theta_1 x + \theta_2 y)$$

one can use:

```
> fit1 <- ppm(bei, ~x + y)
> plot(fit1, how = "image", se = FALSE)
> plot(fit1, se=TRUE)

> cdf.test(fit1, function(x, y) {y})
Spatial Kolmogorov-Smirnov test of
inhomogeneous Poisson process in two
dimensions
data:  covariate function(x, y) {evaluated at points of bei
    and transformed to uniform distribution under fit1
    D = 0.080715, p-value < 2.2e-16
    alternative hypothesis: two-sided
```


Fitted trend
0.01
0.006


Estimated se
0.00045
3e-04
0.00015

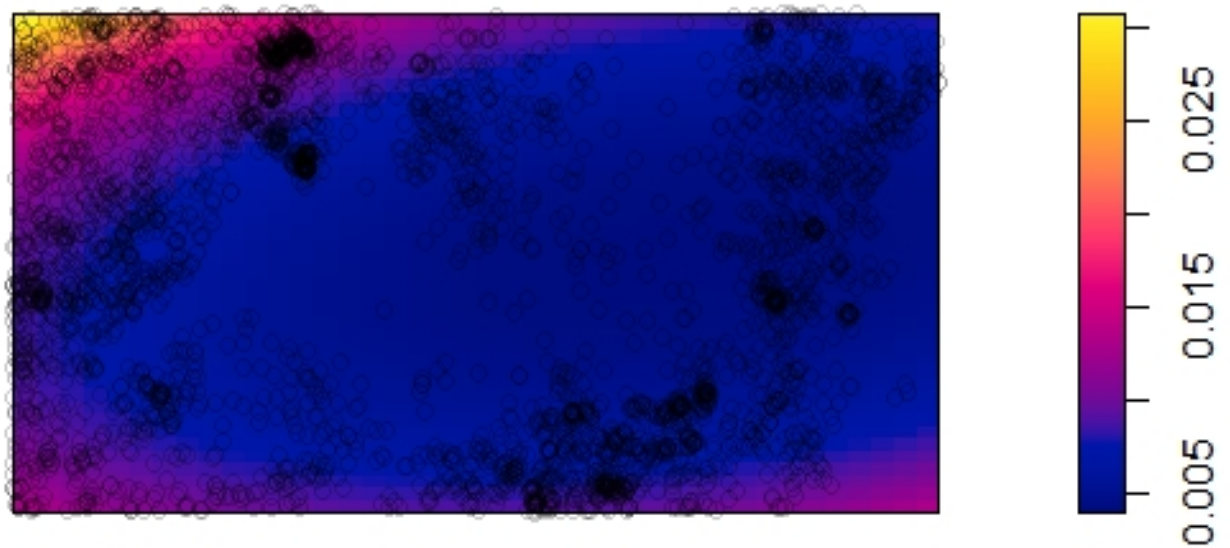
To fit an inhomogeneous Poisson model with an intensity that is log-quadratic one can use:

```
> fit2<-ppm(bei, ~polynom(x, y, 2))
> plot(fit2, how = "image", se = FALSE)

> cdf.test(fit2, function(x, y) {y})
Spatial Kolmogorov-Smirnov test of
inhomogeneous Poisson process in two
dimensions

data:  covariate function(x, y) {evaluated at points of bei
    and transformed to uniform distribution under fit2
    D = 0.030613, p-value = 0.00233
    alternative hypothesis: two-sided
```

Fitted trend

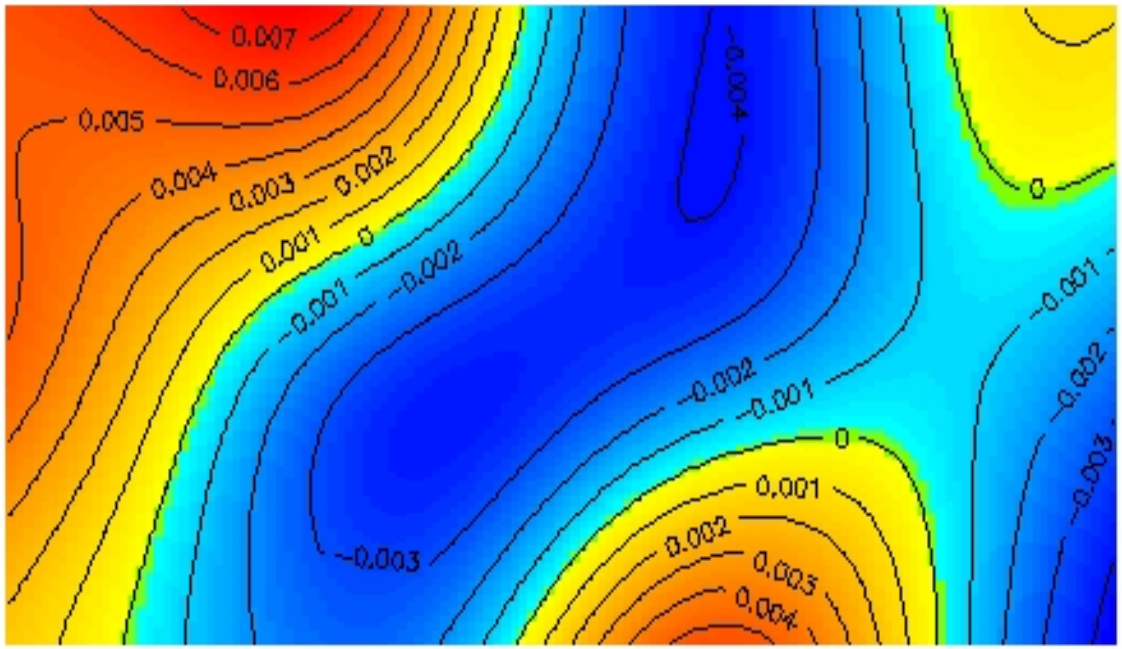


Residuals from the fitted model are an important diagnostic tool. For a fitted Poisson process model, with fitted intensity $\hat{\lambda}(u)$, the predicted number of points falling in any region B is $\int_B \hat{\lambda}(u)du$.

The residual in each region B is the observed minus predicted number of points in B.

A useful way to visualize the residuals is to smooth them:

```
> diagnose.ppm(fit,which = "smooth")
```

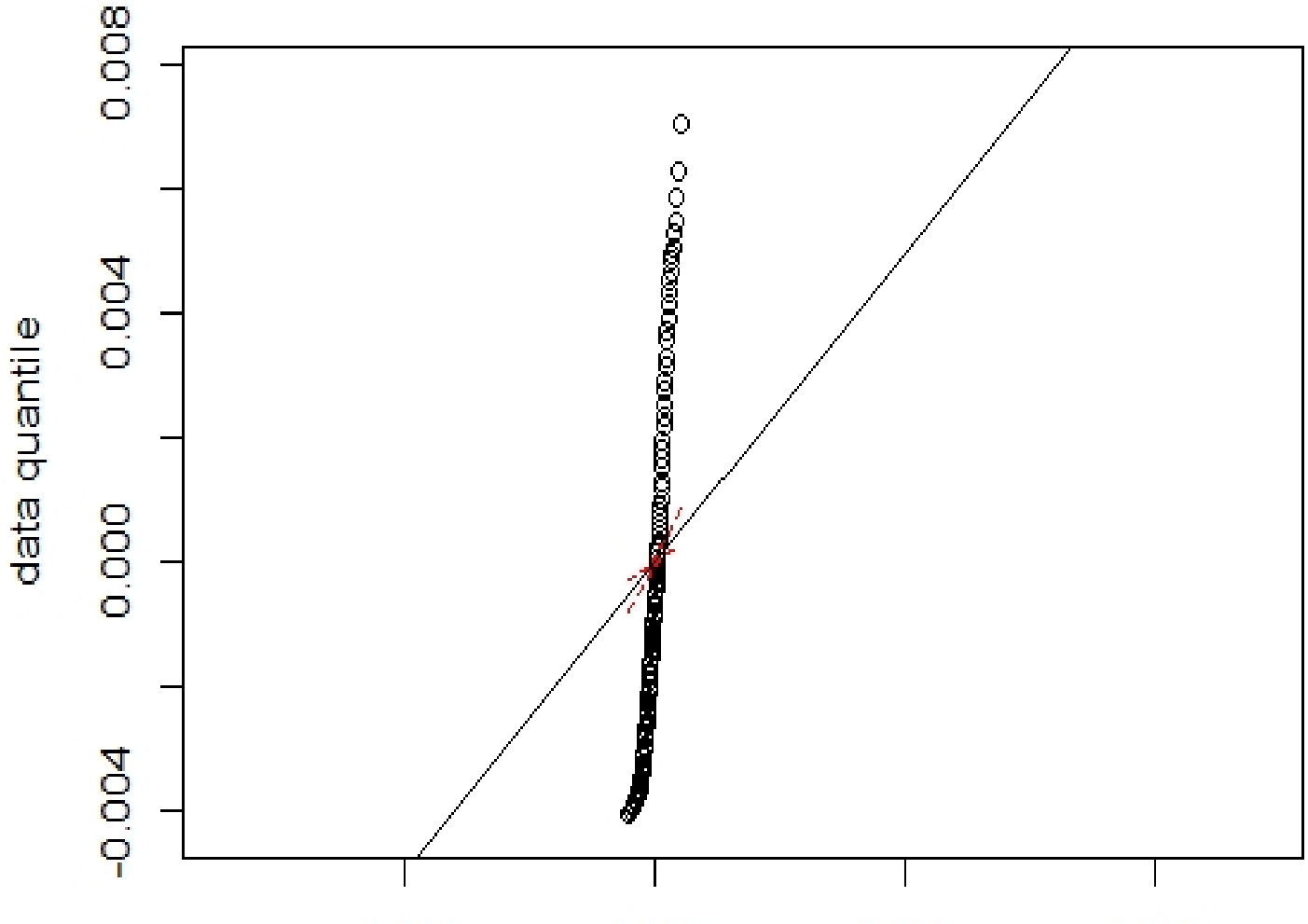


In this example the smoothed residual image contains a visible trend, suggesting that the model is inappropriate.

To validate the interaction terms in a point process model, we should plot the distribution of the residuals. The appropriate tool is a Q-Q plot:

```
> qqplot.ppm(fit)
Extracting model information...Evaluating trend...done.
Simulating 100 realisations...
Diagnostic info:
simulated patterns contained an average of 3615.7 points.
Calculating quantiles...averaging.....Done.
```

This shows a Q-Q plot of the smoothed residuals for a uniform Poisson model fitted to the cells data, with pointwise 5% critical envelopes from simulations of the fitted model. This indicates that the uniform Poisson model is grossly inappropriate for the data.


data quantile
0.008
0.004
0.000
-0.004

Marked Point Processes

The points may have extra information called **marks** attached to them. Thus a marked point process on R^d is a random sequence $M = \{x_n; m_n\}$ from which the points x_n constitute a point process (not marked) in R^d and the m_n are the marks corresponding to the x_n.

Specific examples or marked points are:

- For x the position of a tree, m the stem diameter of the tree;
- For x the centre of an atom, m the type of the atom;
- For x the location (suitably defined) of a convex compact set, m the centred (shifted to origin) set itself.

The marks can be continuous variables, as in the first example, indicators of types as in the second example or actually very complicated indeed, as in the last example.

Manipulating marks

The following tools can manipulate marks in a point pattern:

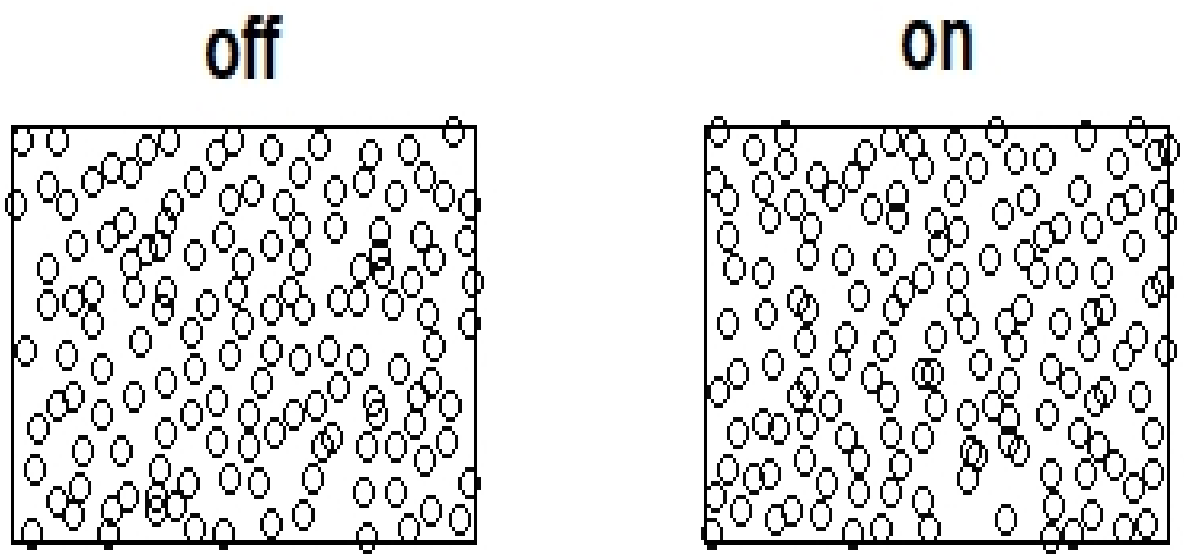
- MARKS() extract marks
- MARKS()<- attach marks to a point pattern
- %MARK% attach marks to a point pattern
- UNMARK remove marks from point pattern

For example, **longleaf** data are tree locations marked by diameters in centimeters. To convert the marks from diameters to circular areas use

```
> data(longleaf)
> str(marks(longleaf))
num [1:584] 32.9 53.5 68 17.7 36.9 51.6 66.4 17.7 21.9 25.7 ...
> d <- marks(longleaf)
> marks(longleaf) <- (pi/4)*d^2
> str(marks(longleaf))
num [1:584] 850 2248 3632 246 1069 ...
```

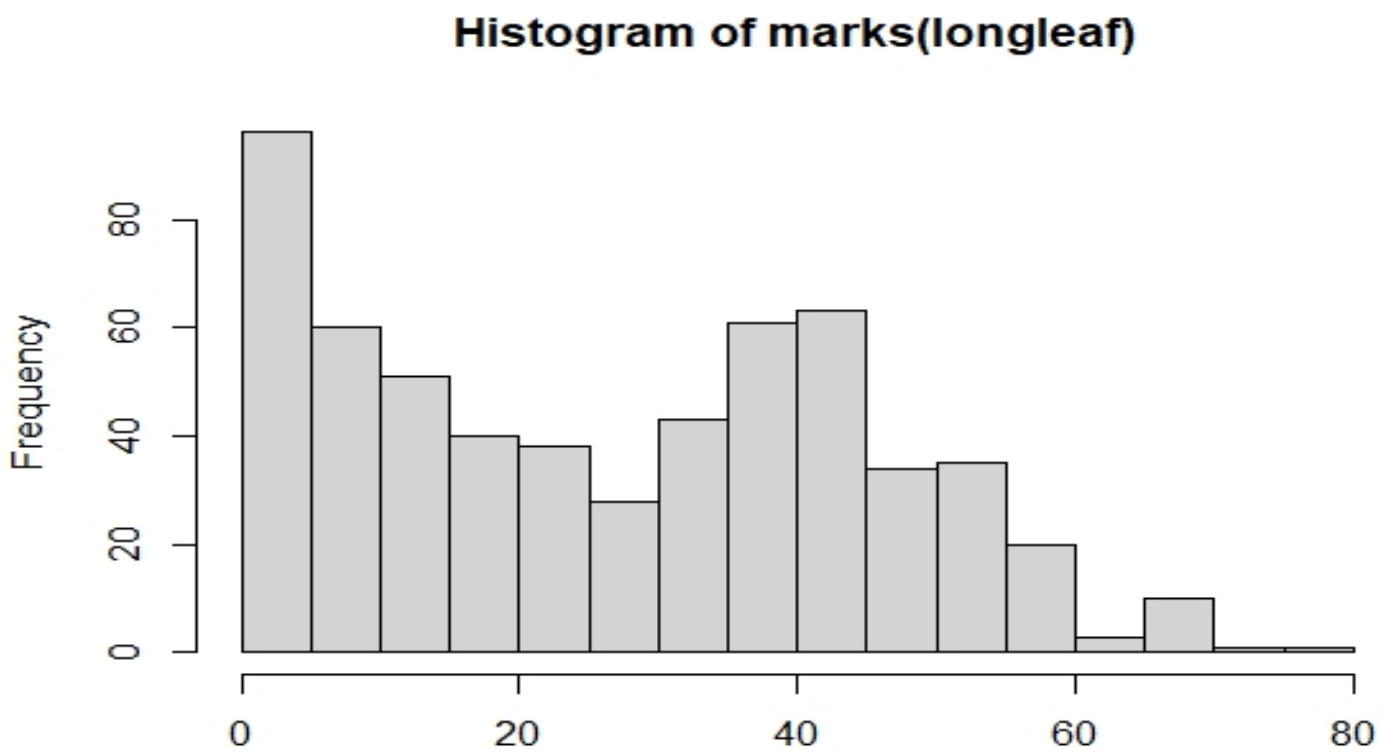
A multitype point pattern can be separated into the sub-patterns of points of each type, using the **split** command:

```
> data(amacrine)
> plot(split(amacrine))
```



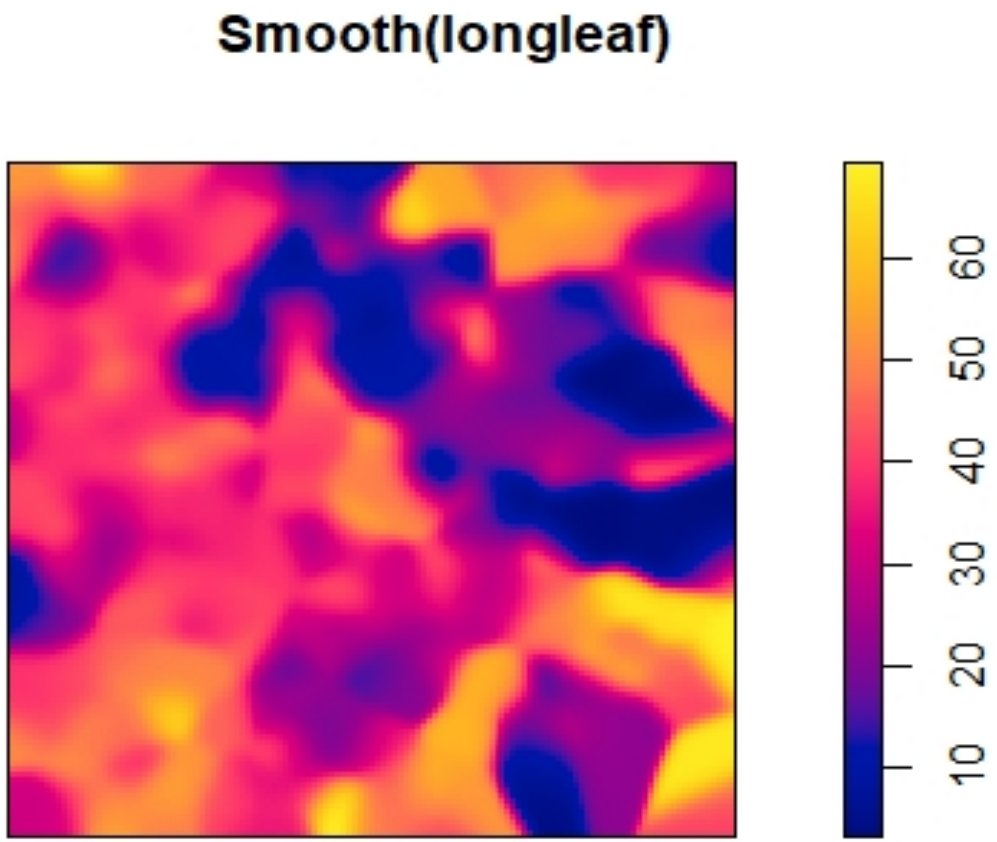
For a point pattern with marks that are numeric (real numbers or integers) or logical values, the mark values can be extracted using the marks function and inspected using the histogram or kernel density estimate:

```
> data(longleaf)
> hist(marks(longleaf))
```



To assess spatial trend in the marks, one way is to form a kernel regression smoother. This is computed by

```
> plot(Smooth(longleaf))
```



Intensity

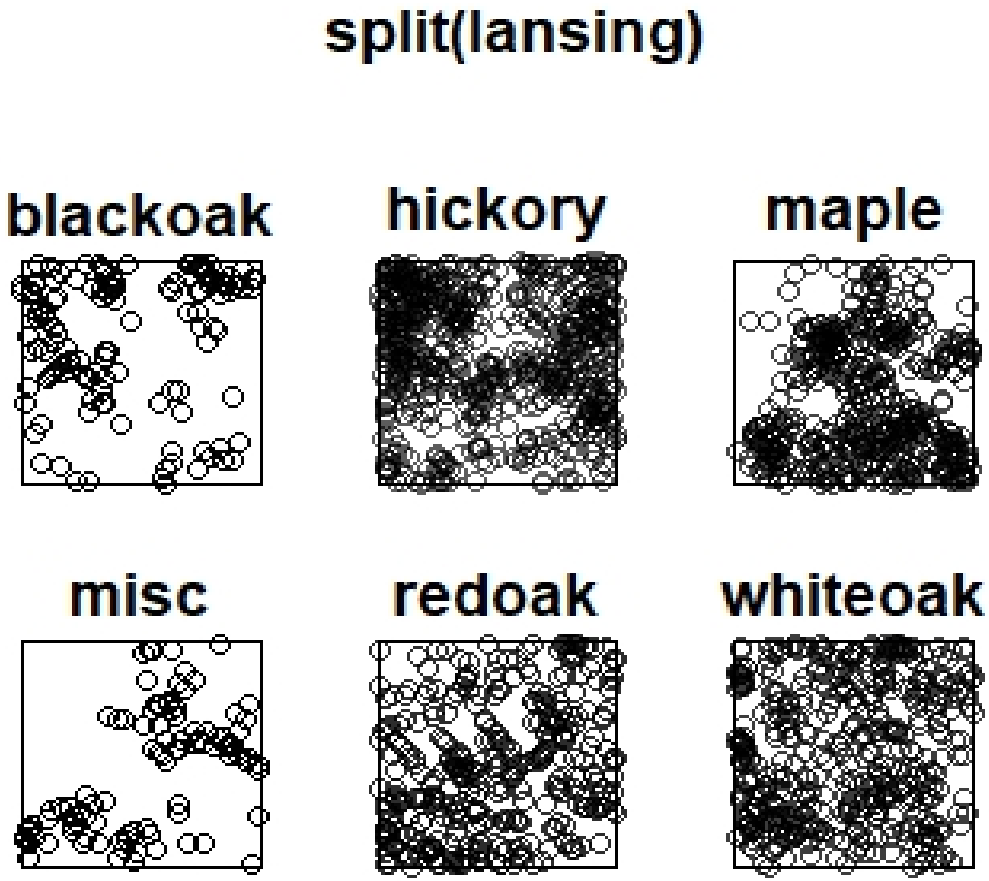
We will use the dataset **lansing.** The data come from an investigation of Lansing Woods, Clinton County, Michigan USA.

The data give the locations of 2251 trees and their botanical classification (into 6 different species: hickories, maples, red oaks, white oaks, black oaks and miscellaneous trees).

The original plot size (924 x 924 feet) has been rescaled to the unit square.

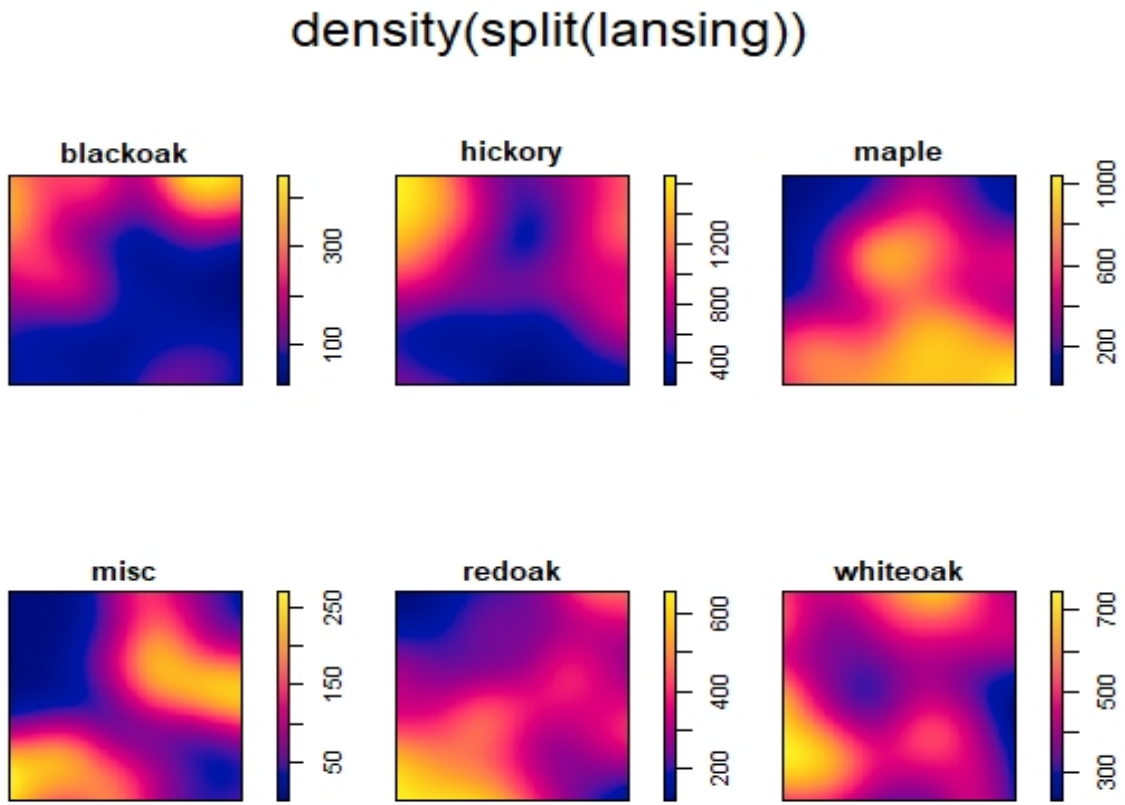
To visualise different species use the commands

```
> data(lansing)
> plot(split(lansing))
```


split(lansing)
blackoak
hickory
maple
misc
redoak
whiteoak

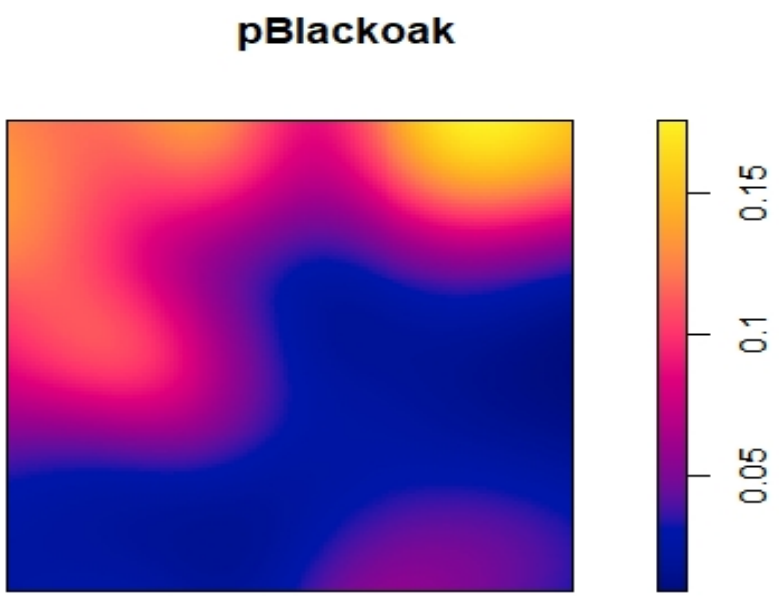
Estimates of the frequency distribution of species and the intensities of species can be obtain by

```
> summary(lansing)
> plot(density(split(lansing)))
```



The relative proportions of intensity can be computed using

```
> Y <- density(split(lansing))
> attach(Y)
> pBlackoak <- eval.im(blackoak/(blackoak + hickory +
+ maple + misc + redoak + whiteoak))
> plot(pBlackoak)
> detach(Y)
```



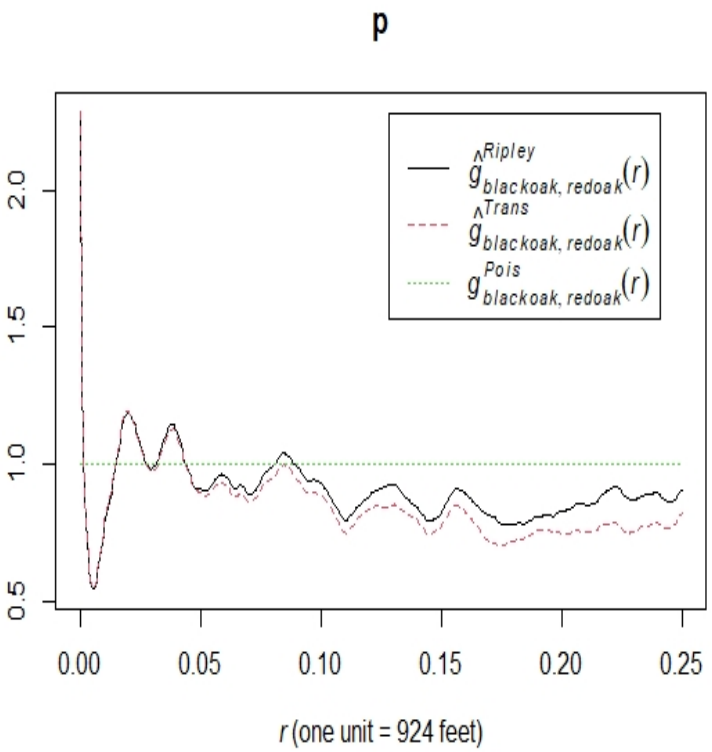
Marks interaction

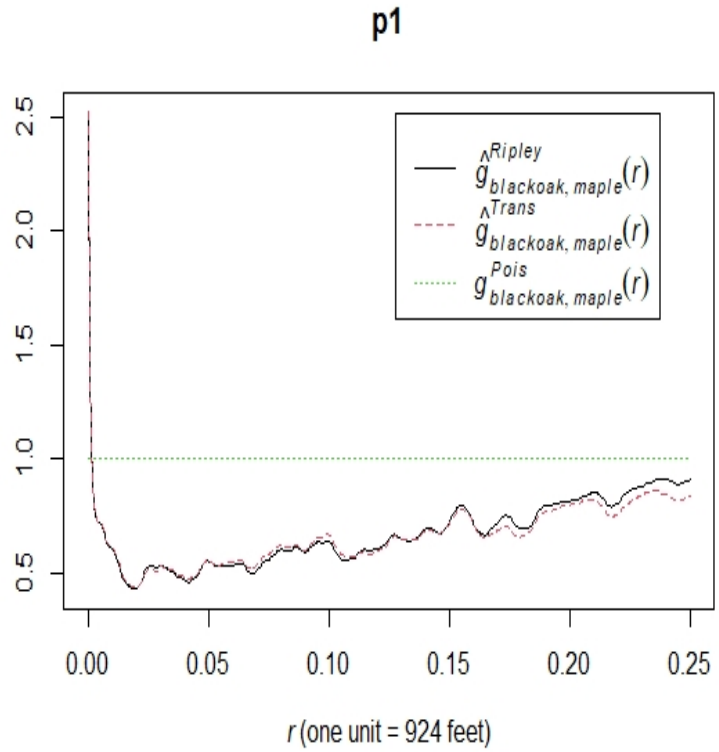
To assess within- and between-marks(species) interaction one can compute exploratory correlation functions. **pcfcross** calculates an estimate of the **cross-type pair correlation function** for a multitype point pattern:

```
> p <- pcfcross(lansing, "blackoak", "redoak")
> plot(p)
> p1 <- pcfcross(lansing, "blackoak", "maple")
> plot(p1)
```

The plots suggest that there is a strong inhibition between black oaks and red oaks at short and long scales and no interaction at medium scale.

While there is inhibition between black oaks and maples at all scales except very small distances.


p
$\hat{g}^{Ripley}_{blackoak,\ redoak}(r)$
$\hat{g}^{Trans}_{blackoak,\ redoak}(r)$
$g^{Pois}_{blackoak,\ redoak}(r)$
0.5
1.0
1.5
2.0
0.00
0.05
0.10
0.15
0.20
0.25
r (one unit = 924 feet)


p1
$\hat{g}^{Ripley}_{blackoak,\ maple}(r)$
$\hat{g}^{Trans}_{blackoak,\ maple}(r)$
$g^{Pois}_{blackoak,\ maple}(r)$
0.5
1.0
1.5
2.0
2.5
0.00
0.05
0.10
0.15
0.20
0.25
r (one unit = 924 feet)

Key R commands

marks(x,...)	*extracts or changes marks attached to a point pattern*
Smooth(X)	*perform spatial smoothing*
split(x)	*divides the data into the groups*
density(x,...)	*computes kernel density estimates*
eval.im(expr,...)	*evaluates expression involving one or more pixel images*
pcfcross(X, i, j)	*estimates the cross-type pair correlation function for a multitype point pattern*