Karlsruher Institut für Technologie

Analyse räumlicher Punktprozesse mit R

Andreas Reichenbacher, Julia Hörrmann | 17. Juni, 2011

Institut für Stochastik - AG Stochastische Geometrie

Bericht über ein Summer Camp des

CENTRE FOR STOCHASTIC GEOMETRY
AND ADVANCED BIOIMAGING
FACULTY OF SCIENCE
AARHUS UNIVERSITY

Gliederung

Spatstat

- Eine open-source R-Bibliothek für räumliche Statistik von Adrian Baddeley und Rolf Turner.

Spatstat

- Eine open-source R-Bibliothek für räumliche Statistik von Adrian Baddeley und Rolf Turner.
- cran.r-project.org

Spatstat

- Eine open-source R-Bibliothek für räumliche Statistik von Adrian Baddeley und Rolf Turner.
- cran.r-project.org
- www.spatstat.org

Spatstat

- Eine open-source R-Bibliothek für räumliche Statistik von Adrian Baddeley und Rolf Turner.
- cran.r-project.org
- www.spatstat.org
- mind. R-Version 2.10.0

Spatstat unterstützt

- Erzeugen, Manipulation und Plotten von Punktmustern

Spatstat unterstützt

- Erzeugen, Manipulation und Plotten von Punktmustern
- Datenanalyse

Spatstat unterstützt

- Erzeugen, Manipulation und Plotten von Punktmustern
- Datenanalyse
- Simulation von Punktprozessen

Spatstat unterstützt

- Erzeugen, Manipulation und Plotten von Punktmustern
- Datenanalyse
- Simulation von Punktprozessen
- parametrische Modellanpassung

Spatstat unterstützt

- Erzeugen, Manipulation und Plotten von Punktmustern
- Datenanalyse
- Simulation von Punktprozessen
- parametrische Modellanpassung
- Hypothesentests

Spatstat unterstützt

- Erzeugen, Manipulation und Plotten von Punktmustern
- Datenanalyse
- Simulation von Punktprozessen
- parametrische Modellanpassung
- Hypothesentests
- Residuenplots und Modelldiagnostik

Spatstat unterstützt

- Erzeugen, Manipulation und Plotten von Punktmustern
- Datenanalyse
- Simulation von Punktprozessen
- parametrische Modellanpassung
- Hypothesentests
- Residuenplots und Modelldiagnostik
- 2-dim. Mosaike

Spatstat unterstützt

- Erzeugen, Manipulation und Plotten von Punktmustern
- Datenanalyse
- Simulation von Punktprozessen
- parametrische Modellanpassung
- Hypothesentests
- Residuenplots und Modelldiagnostik
- 2-dim. Mosaike
- Plotten von 3-dim. Punktmustern

Spatstat unterstützt

- Erzeugen, Manipulation und Plotten von Punktmustern
- Datenanalyse
- Simulation von Punktprozessen
- parametrische Modellanpassung
- Hypothesentests
- Residuenplots und Modelldiagnostik
- 2-dim. Mosaike
- Plotten von 3-dim. Punktmustern

Klassen in spatstat

- **ppp**: planar point pattern

Klassen in spatstat

- **ppp**: planar point pattern
- **owin**: spatial region (Beobachtungsfenster)

Klassen in spatstat

- **ppp**: planar point pattern
- **owin**: spatial region (Beobachtungsfenster)
- **im**: pixel image

Klassen in spatstat

- **ppp**: planar point pattern
- **owin**: spatial region (Beobachtungsfenster)
- **im**: pixel image
- **tess**: tessellation

Klassen in spatstat

- **ppp**: planar point pattern
- **owin**: spatial region (Beobachtungsfenster)
- **im**: pixel image
- **tess**: tessellation
- **pp3**: 3-d point pattern

Klassen in spatstat

- **ppp**: planar point pattern
- **owin**: spatial region (Beobachtungsfenster)
- **im**: pixel image
- **tess**: tessellation
- **pp3**: 3-d point pattern

```
> library(spatstat)
> data(humberside)
```

Klassen in spatstat

- **ppp**: planar point pattern
- **owin**: spatial region (Beobachtungsfenster)
- **im**: pixel image
- **tess**: tessellation
- **pp3**: 3-d point pattern

```
> library(spatstat)
> data(humberside)

> humberside

marked planar point pattern: 203 points
multitype, with levels = case control
window: polygonal boundary
enclosing rectangle: [4690, 5411] x [4150, 4758] units (one u
```

Klassen in spatstat

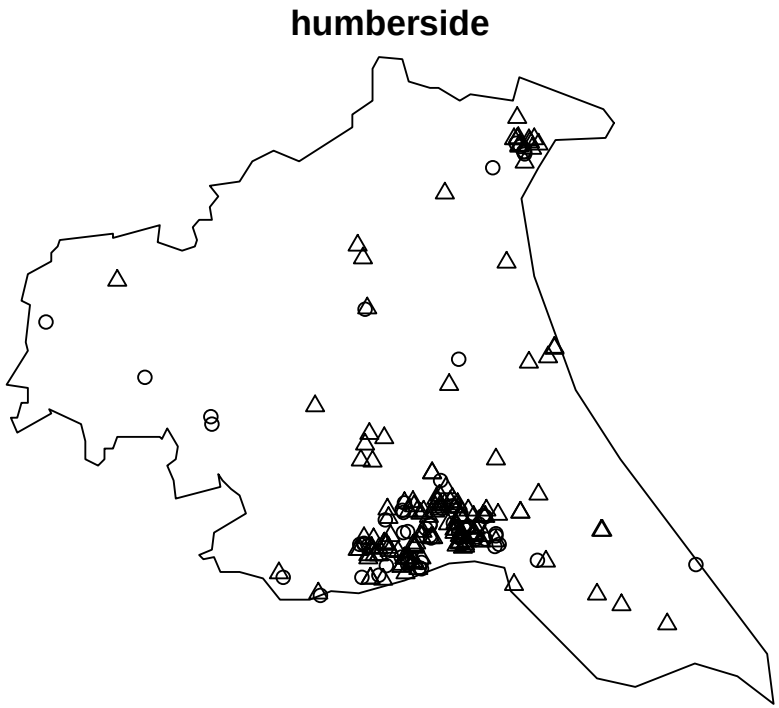
```
> summary(humberside)
Marked planar point pattern: 203 points
Average intensity 0.000993 points per square unit (one unit =

*Pattern contains duplicated points*
Multitype:
        frequency proportion intensity
case           62      0.305  0.000303
control       141      0.695  0.000690

Window: polygonal boundary
single connected closed polygon with 102 vertices
enclosing rectangle: [4690, 5411]x[4150, 4758]units
Window area =  204487 square units
Unit of length: 100 metres
```

Klassen in spatstat

```
> plot(humberside)
```



Befehle zur ppp-Klasse

- plot.ppp (aufgerufen durch plot(„ppp-Objekt“))

Befehle zur ppp-Klasse

- plot.ppp (aufgerufen durch plot(„ppp-Objekt“))
- summary

Befehle zur ppp-Klasse

- plot.ppp (aufgerufen durch plot(„ppp-Objekt“))
- summary
- coords (liest x- und y-Werte aus)

Befehle zur ppp-Klasse

- plot.ppp (aufgerufen durch plot(„ppp-Objekt“))
- summary
- coords (liest x- und y-Werte aus)
- marks (liest Marken aus)

Befehle zur ppp-Klasse

- plot.ppp (aufgerufen durch plot(„ppp-Objekt“))
- summary
- coords (liest x- und y-Werte aus)
- marks (liest Marken aus)
- Objekt$window (liest Beobachtungsfenster aus, Klasse **owin**)

Befehle zur ppp-Klasse

- plot.ppp (aufgerufen durch plot(„ppp-Objekt“))
- summary
- coords (liest x- und y-Werte aus)
- marks (liest Marken aus)
- Objekt$window (liest Beobachtungsfenster aus, Klasse **owin**)
- unmark (löscht Marken)

Befehle zur ppp-Klasse

- plot.ppp (aufgerufen durch plot(„ppp-Objekt“))
- summary
- coords (liest x- und y-Werte aus)
- marks (liest Marken aus)
- Objekt$window (liest Beobachtungsfenster aus, Klasse **owin**)
- unmark (löscht Marken)
- split (kategorielle Marken benötigt: teilt Punktprozess auf)

Befehle zur ppp-Klasse

- plot.ppp (aufgerufen durch plot(„ppp-Objekt“))
- summary
- coords (liest x- und y-Werte aus)
- marks (liest Marken aus)
- Objekt$window (liest Beobachtungsfenster aus, Klasse **owin**)
- unmark (löscht Marken)
- split (kategorielle Marken benötigt: teilt Punktprozess auf)
- density(·,sigma=·) (schätzt Intensitätsfunktion, Klasse **im**; benutzt Gaußkern)

Befehle zur ppp-Klasse

- plot.ppp (aufgerufen durch plot(„ppp-Objekt“))
- summary
- coords (liest x- und y-Werte aus)
- marks (liest Marken aus)
- Objekt$window (liest Beobachtungsfenster aus, Klasse **owin**)
- unmark (löscht Marken)
- split (kategorielle Marken benötigt: teilt Punktprozess auf)
- density(·,sigma=·) (schätzt Intensitätsfunktion, Klasse **im**; benutzt Gaußkern)

Daten einlesen, ppp-Objekte erzeugen

Daten (x-Werte, y-Werte, Marken) in Text-Datei (oder csv-Datei)

Daten einlesen, ppp-Objekte erzeugen

Daten (x-Werte, y-Werte, Marken) in Text-Datei (oder csv-Datei)

```
> mydata <- read.table("anthills.txt", header = TRUE)
```

Daten einlesen, ppp-Objekte erzeugen

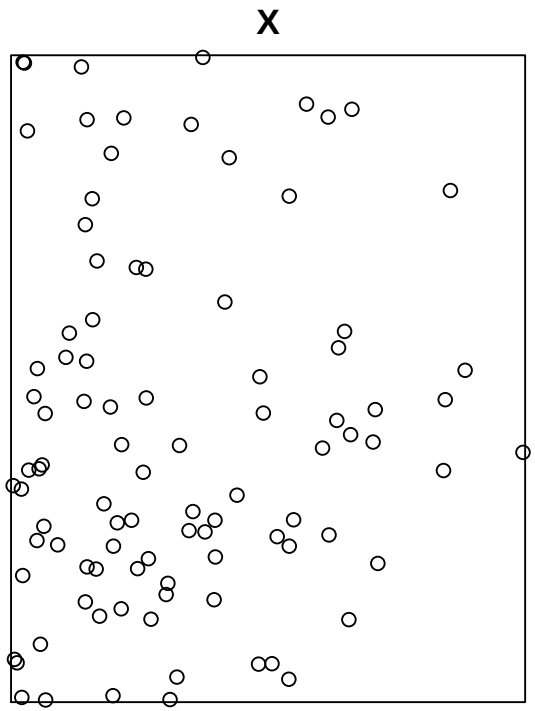
Daten (x-Werte, y-Werte, Marken) in Text-Datei (oder csv-Datei)

```
> mydata <- read.table("anthills.txt", header = TRUE)
> X <- ppp(x, y, xrange, yrange)
```

Daten einlesen, ppp-Objekte erzeugen

Daten (x-Werte, y-Werte, Marken) in Text-Datei (oder csv-Datei)

```
> mydata <- read.table("anthills.txt", header = TRUE)
> X <- ppp(x, y, xrange, yrange)
```

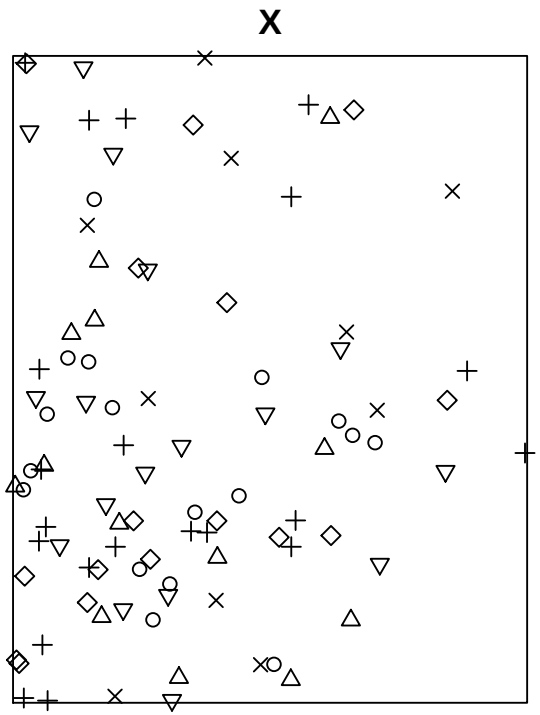


Daten einlesen, ppp-Objekte erzeugen

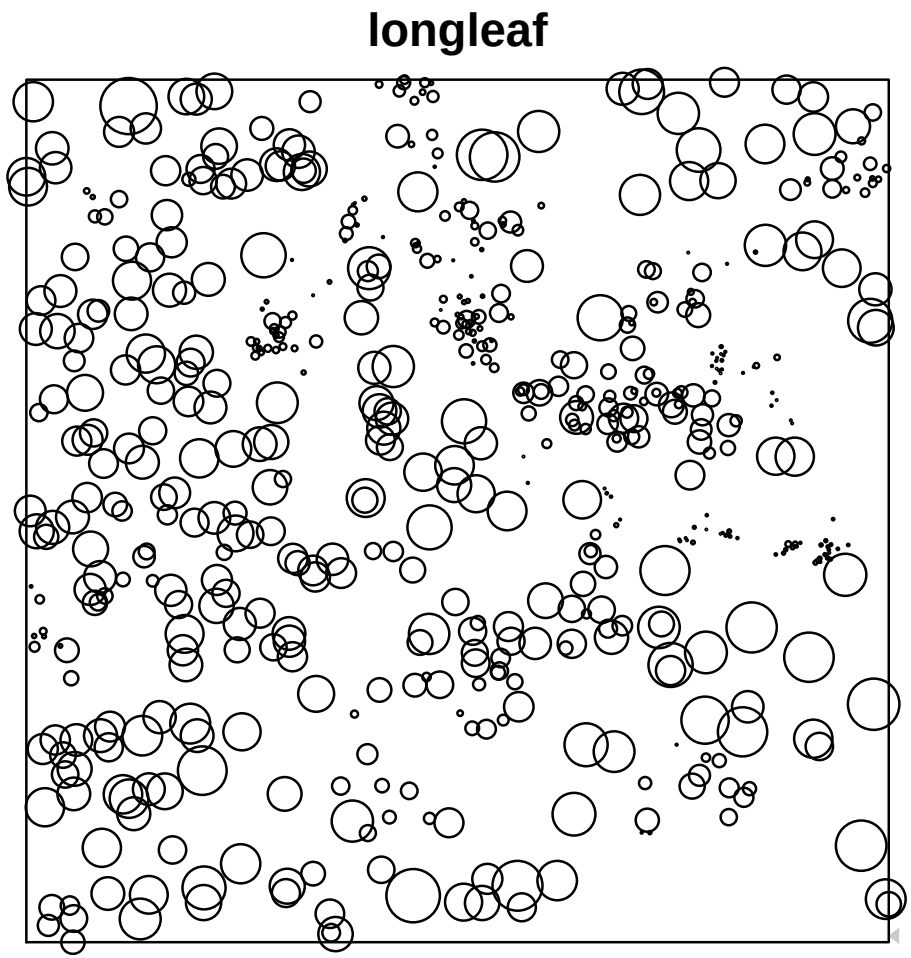
```
> X <- ppp(x, y, xrange, yrange, marks = m)
```

Daten einlesen, ppp-Objekte erzeugen

```
> X <- ppp(x, y, xrange, yrange, marks = m)
e f g h i j
1 2 3 4 5 6
```



stetige Marken



Beobachtungsfenster

rechteckige Fenster

```
> owin(xrange,yrange)
```

Beobachtungsfenster

rechteckige Fenster

```
> owin(xrange,yrange)
> owin(c(0, 3), c(1, 2))
window: rectangle = [0, 3] x [1, 2] units
```

Beobachtungsfenster

rechteckige Fenster

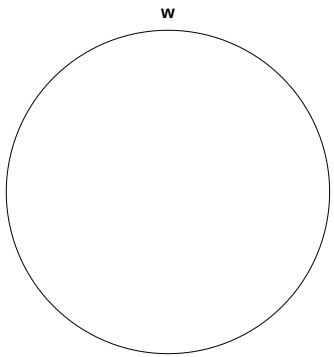
```
> owin(xrange,yrange)
```

```
> owin(c(0, 3), c(1, 2))
```

```
window: rectangle = [0, 3] x [1, 2] units
```

kreisförmige Fenster

```
> w <- disc(radius = 3, centre = c(0, 0))
> plot(w)
```

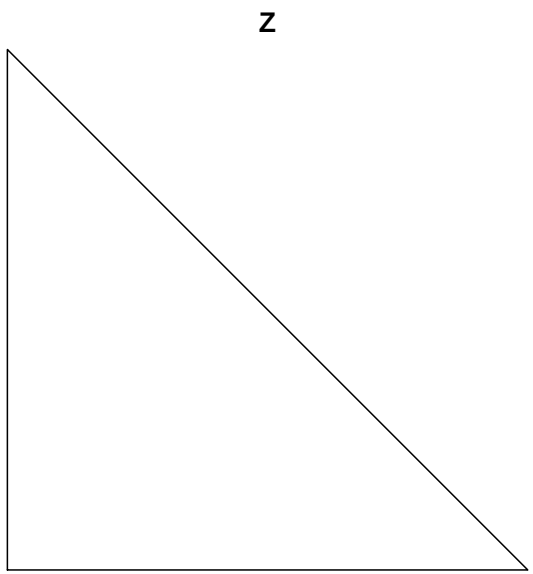


polygonale Fenster

nur für nicht-kommerzielle Zwecke, benötigen

```
> spatstat.options(gpclib = TRUE)
```

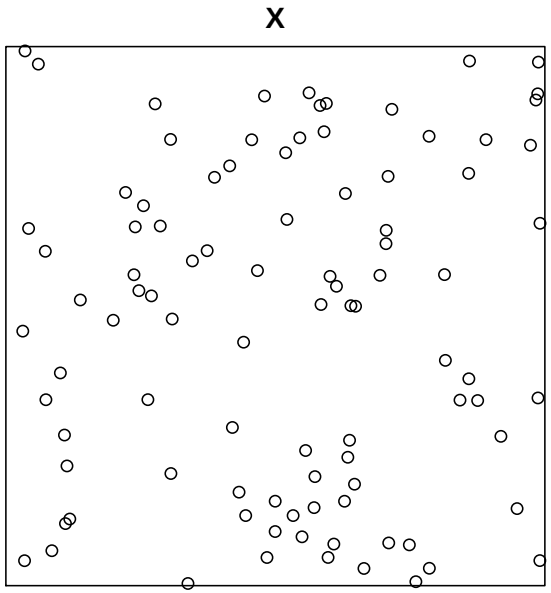
```
> Z <- owin(poly = list(x = c(0, 1, 0), y = c(0, 0, 1)))
> plot(Z)
```



wichtige Befehle:
area.owin (Fläche)
diameter (Durchmesser)
perimeter (Umfang)

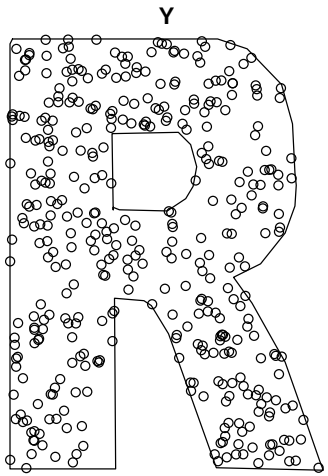
Simulation von Poissonpunktprozessen

```
> X <- rpoispp(100)
> plot(X)
```



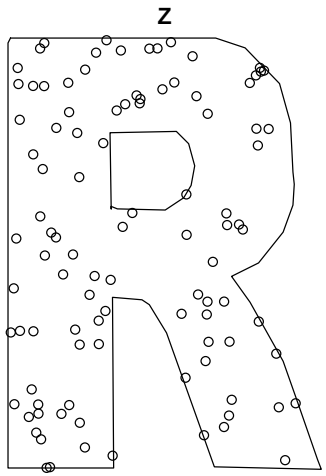
Simulation von Poissonpunktprozessen

```
> data(letterR)
> area.owin(letterR)
[1] 3.697251
> Y <- rpoispp(100, win = letterR)
> plot(Y)
```



Simulation von Poissonpunktprozessen

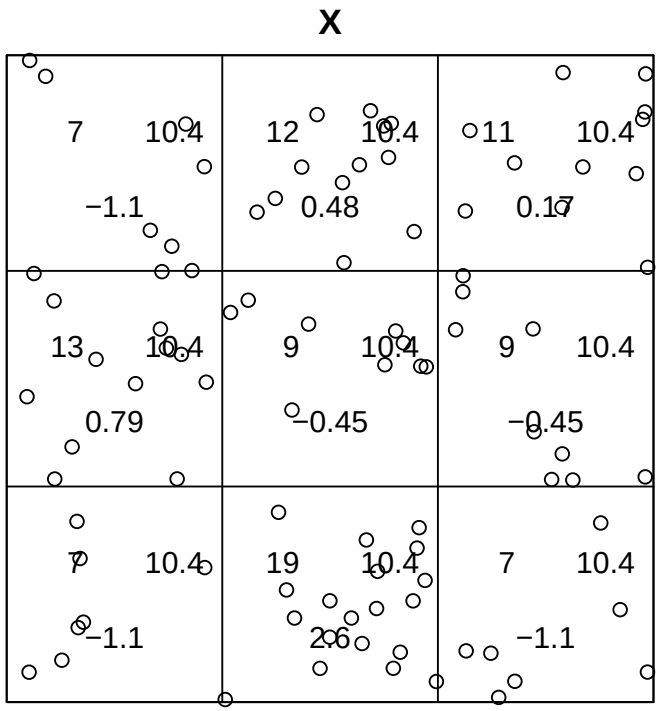
```
> Z <- runifpoint(100, win = letterR)
> plot(Z)
```



Test auf homogenen Poisson-PP

- χ^2-Test basierend auf Quadratcount

```
> plot(X)
> plot(quadrat.test(X, nx = 3, ny = 3), add = T)
```

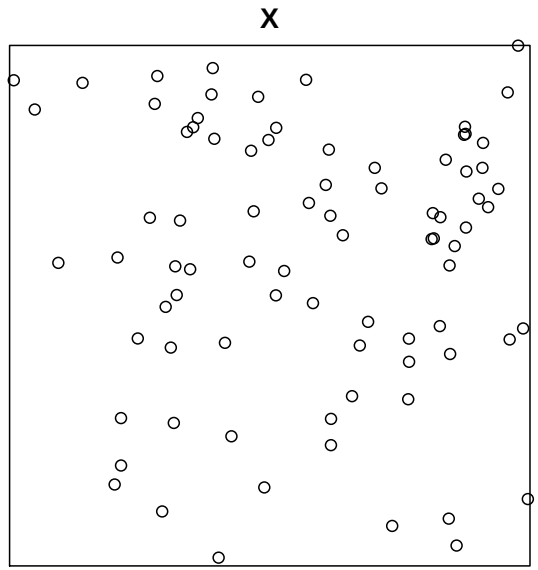


Inhomogene Poisson-PPe

```
> lambda <- function(x, y) {
+     100 * (x + y)
+ }
> X <- rpoispp(lambda)
```

Inhomogene Poisson-PPe

```
> lambda <- function(x, y) {
+     100 * (x + y)
+ }
> X <- rpoispp(lambda)
```



- Es gilt

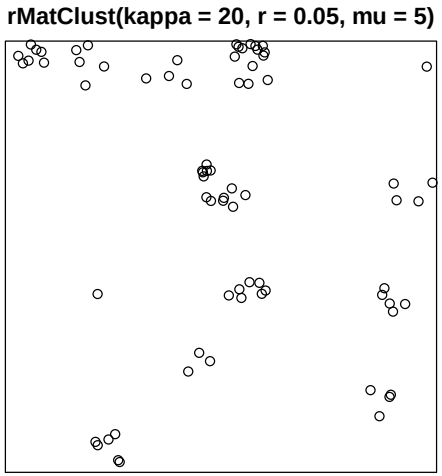
$$\Theta_X(B) = \int_B \lambda(u)du$$

für $B \in \mathcal{B}(\mathbb{R}^2)$.

Cluster-Prozesse

- Matérn-Cluster-Prozess

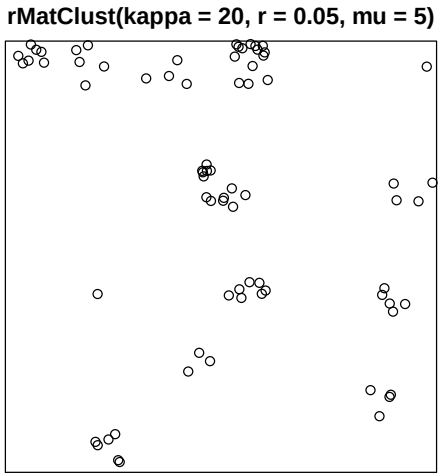
```
> plot(rMatClust(kappa = 20, r = 0.05, mu = 5))
```



Cluster-Prozesse

- Matérn-Cluster-Prozess

```
> plot(rMatClust(kappa = 20, r = 0.05, mu = 5))
```

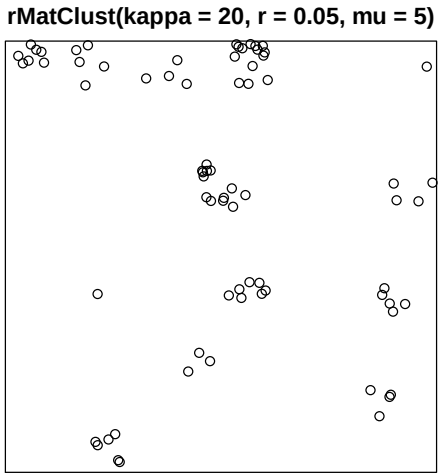


- Thomas-Prozess

Cluster-Prozesse

- Matérn-Cluster-Prozess

```
> plot(rMatClust(kappa = 20, r = 0.05, mu = 5))
```



- Thomas-Prozess
- Gauss-Poisson-Prozess

Punktmuster in 3 Dimensionen

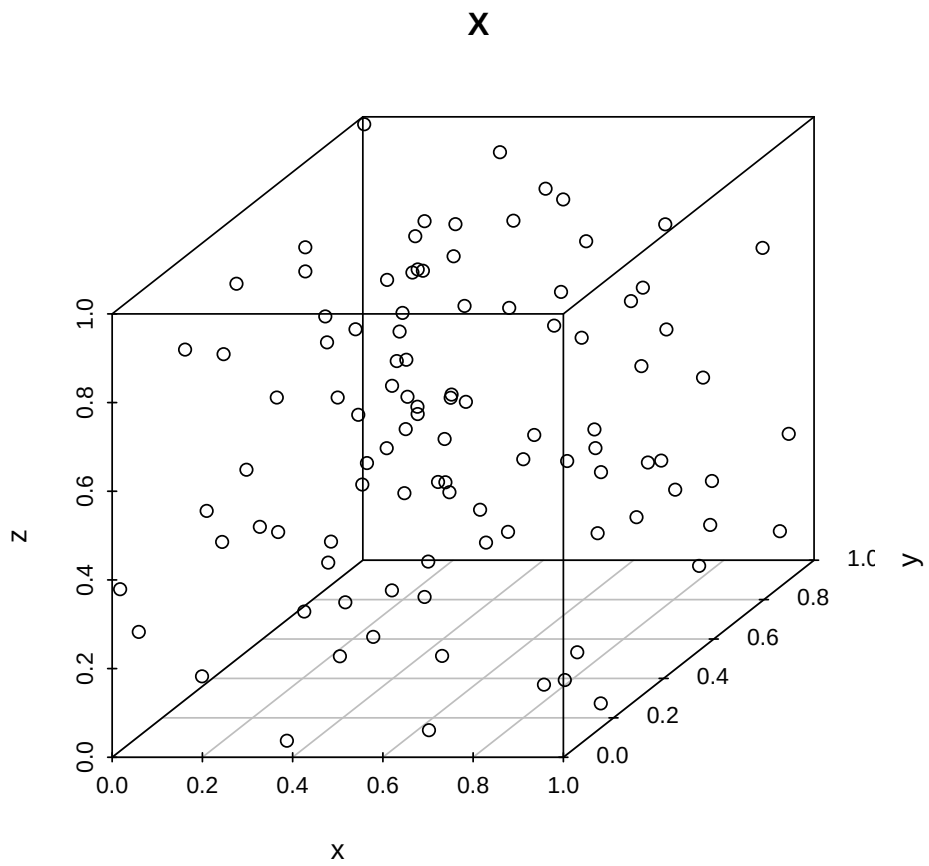
Basisfunktionen seit kurzem in spatstat implementiert.

```
> x <- runif(100)
> y <- runif(100)
> z <- runif(100)
> X <- pp3(x, y, z, box3(c(0, 1)))
> summary(X)

Three-dimensional point pattern
100 points
Box: [0, 1] x [0, 1] x [0, 1] units
Volume 1 cubic unit
Average intensity 100 points per cubic unit
```

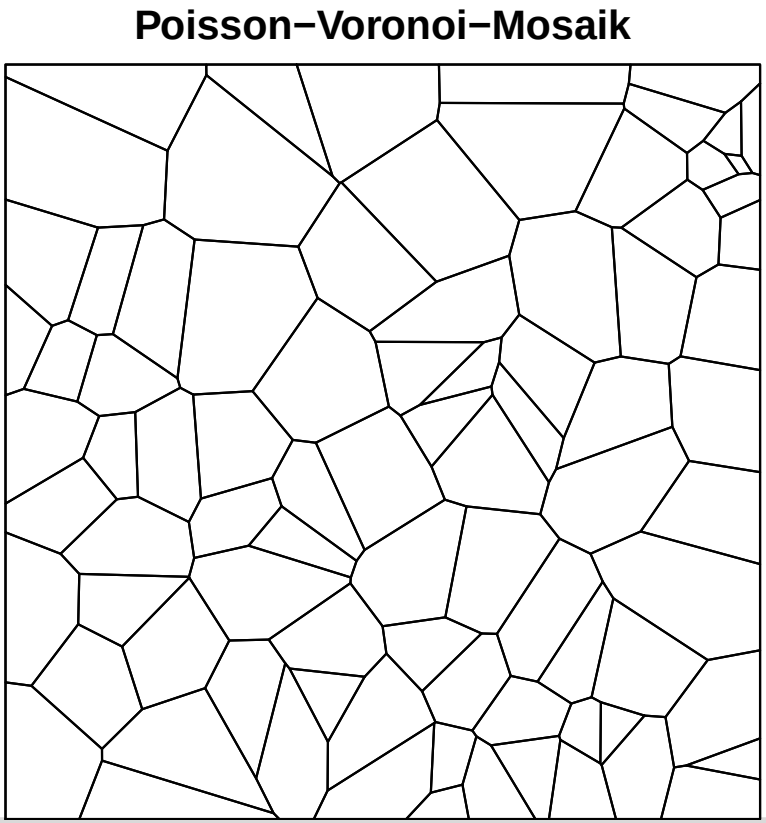
Punktmuster in 3 Dimensionen

```
> plot(X)
```



Mosaike

```
> Y <- rpoispp(100)
> plot(dirichlet(Y), main = "Poisson-Voronoi-Mosaik")
```



Sweave

Sweave is used to combine R and LaTeX.
First a command and its result is printed:

```
> 2 + 2
```

```
[1] 4
```

You can also show the command but not the result using `results=hide` (you can only see this and other options in the `.Rnw` file):

```
> 2 + 2
```

Quellen

A. Baddeley and R. Turner. Practical maximum pseudolikelihood for spatial point patterns (with discussion). Australian and New Zealand Journal of Statistics, 42(3):283-322, 2000.

Ausführliches Skript:
http://www.csiro.au/resources/Spatial-Point-Patterns-in-R.html