Spatstat Package Presentation Handout

By Emily Tate

### Introduction:

“Spatstat” analyzes spatial point patterns in 2D. It can be used across a variety of disciplines that use statistics which makes it applicable to me as a marine science major. Although this package specializes in 2D interpretations it can also create 3D objects using the pp3 class. There are hundreds of function options. Luckily, the options are well-documented to facilitate applying them to analysizing and visiualizing data.

The most recent update, “Puppy zoomies”, includes features including: fast kernel estimation on a linear network using 2D kernel; nonparametric maximum likelihood estimation of ‘rho’; extension of Scott’s rule for bandwidth, selection, cross-validated bandwidth selection on a linear network; more support for character-valued images; and random thinning of clumps. Datasets in spatstat are lazy-loaded, so they can be used by just typing the name of the dataset.

### Features and Basic Usage:

Spatstat has options for: \* Creating and manipulating data \* Exploratory Data Analysis \* Model fitting with Cox and cluster models \* Model fitting with Poisson and Gibbs models \* Model fitting with determinantal point processes \* Model fitting with spatial logistic regression

### Main types of spatial data that can be

* ppp point pattern
* owin window (spatial region)
* im pixel image
* psp line segment pattern
* tess tessellation
* pp3 three-dimensional point pattern
* ppx point pattern in any number of dimensions
* lpp point pattern on a linear network

#### Package Application

#Install the Spatstat package and data to practice with  
install.packages('spatstat') #uncomment this line to run code

## Installing package into '/home/tate4109/R/x86\_64-pc-linux-gnu-library/3.6'  
## (as 'lib' is unspecified)

library('spatstat')

## spatstat 1.61-0 (nickname: 'Puppy zoomies')   
## For an introduction to spatstat, type 'beginner'

install.packages('spatstat.utils')

## Installing package into '/home/tate4109/R/x86\_64-pc-linux-gnu-library/3.6'  
## (as 'lib' is unspecified)

library('spatstat.utils')

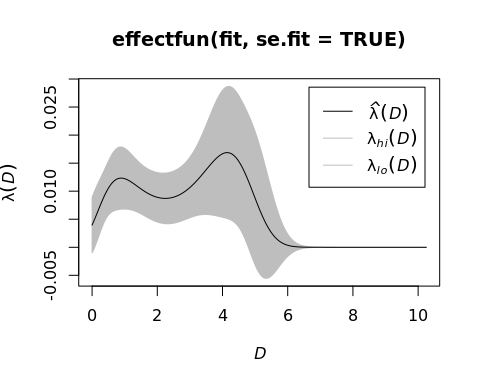
beginner  
# -== Welcome to the 'spatstat' package! ==-  
  
#For a friendly introduction to spatstat, type the command  
# vignette('getstart')  
#which displays the document "Getting Started with Spatstat".  
  
#For an overview of all capabilities, type   
# help(spatstat)  
  
#View the documentation for any command/function 'foo' by typing   
# help(foo)  
  
#Activate the graphical help interface by typing  
# help.start()  
  
#To handle spatial data in the 'shapefile' format, see the document  
#"Handling shapefiles in the spatstat package", by typing  
# vignette('shapefiles')  
#  
#For a complete course on spatstat, see the book  
# "Spatial Point Patterns: Methodology and Applications with R"  
#by Baddeley, Rubak and Turner, Chapman and Hall/CRC Press, December 2015.  
#For a summary of changes to spatstat since the book was finished, type  
# vignette('updates')  
#Visit the website  
# www.spatstat.org  
#for updates and free chapters.  
#  
#For news about the very latest version of spatstat, type  
# latest.news

### Examples

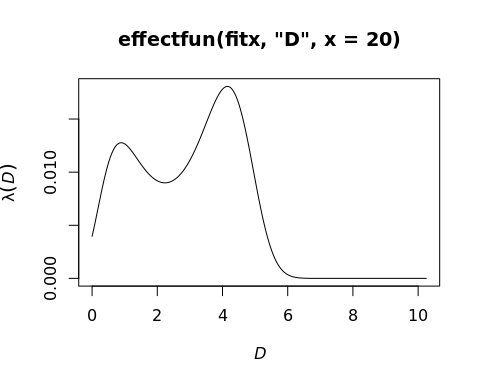
##### effectfun ~ Computes Fitted Effect of a Spatial Covariate in a Point Process Model  
X <- copper$SouthPoints  
D <- distfun(copper$SouthLines)  
fit <- ppm(X ~ polynom(D, 5))  
effectfun(fit)

## Function value object (class 'fv')  
## for the function D -> lambda(D)  
## ..........................................  
## Math.label Description   
## D D value of covariate D  
## lambda hat(lambda)(D) fitted intensity   
## ..........................................  
## Default plot formula: lambda~D  
## where "." stands for 'lambda'  
## Recommended range of argument D: [0.00018776, 10.251]  
## Available range of argument D: [0.00018776, 10.251]

plot(effectfun(fit, se.fit=TRUE))



fitx <- ppm(X ~ x + polynom(D, 5))  
plot(effectfun(fitx, "D", x=20))



#### rpoispp ~ Generate Poisson Point Pattern  
# uniform Poisson process with intensity 100 in the unit square  
pp <- rpoispp(100)  
# uniform Poisson process with intensity 1 in a 10 x 10 square  
pp <- rpoispp(1, win=owin(c(0,10),c(0,10)))  
# plots should look similar !  
# inhomogeneous Poisson process in unit square  
# with intensity lambda(x,y) = 100 \* exp(-3\*x)  
# Intensity is bounded by 100  
pp <- rpoispp(function(x,y) {100 \* exp(-3\*x)}, 100)  
# How to tune the coefficient of x  
lamb <- function(x,y,a) { 100 \* exp( - a \* x)}  
pp <- rpoispp(lamb, 100, a=3)  
# pixel image  
Z <- as.im(function(x,y){100 \* sqrt(x+y)}, unit.square())  
pp <- rpoispp(Z)  
# randomising an existing point pattern  
rpoispp(intensity(cells), win=Window(cells))

## Planar point pattern: 49 points  
## window: rectangle = [0, 1] x [0, 1] units

rpoispp(ex=cells)

## Planar point pattern: 40 points  
## window: rectangle = [0, 1] x [0, 1] units

### Other Information:

#### Type ‘beginner’ for an introduction to spatstat. From there, the command vignette(‘getstart’) will displays the document “Getting Started with Spatstat”.

#### The website for the book “Spatial Point Patterns: Methodology and Applications with R” by Baddeley, Rubak and Turner, Chapman and Hall/CRC Press, December 2015. can be found at <http://book.spatstat.org/>.

#### A comprehensive collection of resources relating to the package can be accessed at <http://spatstat.org/download.html>.

#### Spatstat Quick Reference guide: <http://spatstat.org/resources/spatstatQuickref.pdf>

#### PDF of Collecting and Handling Point Pattern Data: <http://book.spatstat.org/sample-chapters/chapter03.pdf>

#### Examples of function uses: <https://cran.r-project.org/web/packages/spatstat/spatstat.pdf>