

# Análisis de presencias con procesos de puntos

Tutorial básico de spatstat

---

Gerardo Martín

2022-06-29

# Introducción

---

# ¿Qué es spatstat?

- `spat` = Spatial
- `stat` = Statistics
- Paquete de **R** para hacer estadística espacial de procesos de puntos
  - Sintaxis típica de R básico
  - Objetos específicos del paquete

Por desarrolladores del paquete

# ¿Qué veremos aquí?

1. Instalación y carga
2. Formateo
3. Transformación de otros paquetes a **spatstat**
4. Análisis exploratorio
  - 4.1. Detección de autocorrelación
  - 4.2. Análisis de respuesta a covariables
  - 4.3. Propuesta de modelos en relación a covariables
5. Ajuste de modelo

## Instalación y carga

---

Instalación

```
install.packages("spatstat")
```

Carga

```
library(spatstat)
```

Necesitamos también (para leer raster de covariables):

```
library(terra)
```

```
library(foreach)
```



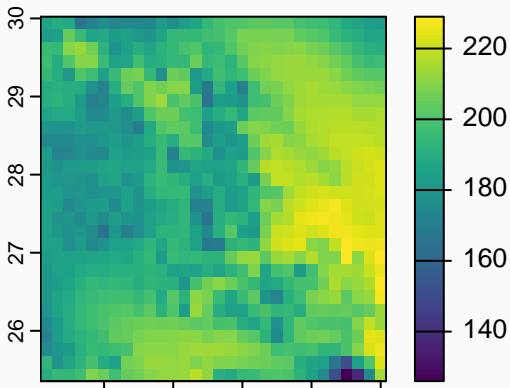
## Formateo

---

Variables ambientales

## Carga de rasters

```
archivos <- list.files("Datos-ejemplos/", "tif",  
                      full.names = T,  
                      recursive = F)  
  
r <- rast(archivos)  
  
plot(r[[1]])
```



Si hacemos:

```
class(r)
## [1] "SpatRaster"
## attr(,"package")
## [1] "terra"
```

vemos que el tipo de objeto es `SpatRaster`, pero `spastat` utiliza `im`

Transformación en lote con `imFromStack`

Para cargar `imFromStack`:

```
source("Funciones-spatstat/imFromStack.R")
```

Y se usa:

```
r.im <- imFromStack(r)
class(r.im)

## [1] "list"
```

Cálculo de intensidad se hace con la “ventana”. Utilizaremos la función `winFromRaster`:

```
w <- as.owin(r.im[[1]])
```

Para verificar la clase del objeto:

```
class(w)
```

```
## [1] "owin"
```

Vamos a simular las coordenadas  $x$  y  $y$  de un proceso de puntos:

```
set.seed(984573)
puntos <- as.data.frame(r, xy = T)[sample(1:840, 200),]
puntos$x <- puntos$x + rnorm(200, 0, 0.05)
puntos$y <- puntos$y + rnorm(200, 0, 0.05)
```

El formato que requiere spatstat es ppp:

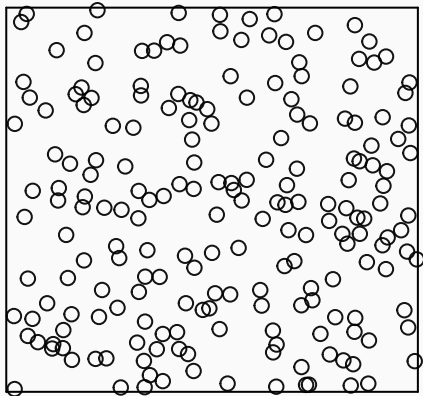
```
puntos.ppp <- ppp(x = puntos$x,
                 y = puntos$y,
                 window = w,
                 check = F)

class(puntos.ppp)

## [1] "ppp"
```

```
plot(puntos.ppp)
```

**puntos.ppp**



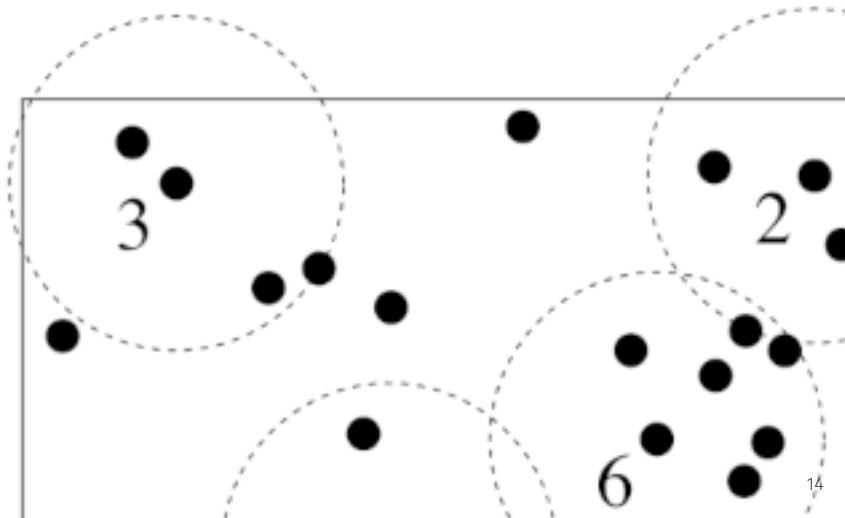


## Análisis exploratorio

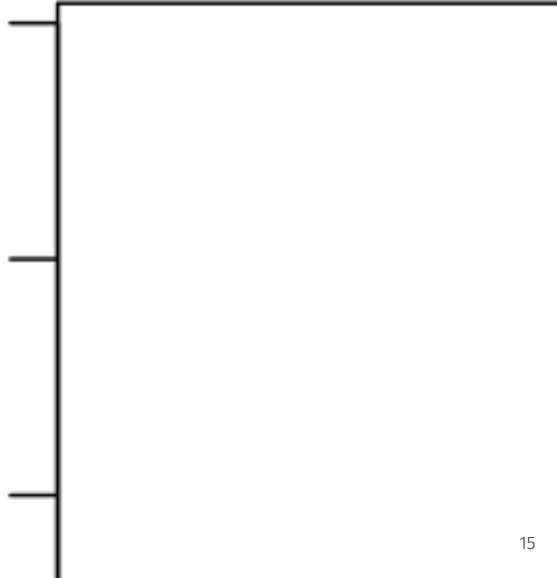
---

# Detección/medición de autocorrelación

- Análisis visual



$K(r)$   
0.10    0.15    0.20

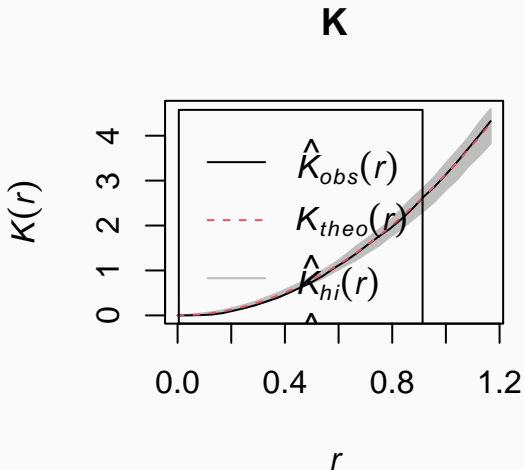


```
K <- envelope(puntos.ppp, fun = Kest, nsim = 39)
```

- `puntos.ppp` objeto que contiene los puntos
- `fun = Kest` función que se estimará
- `nsim = 39` número de simulaciones para IC al 95%

## Función de Ripley

```
K <- envelope(puntos.ppp, fun = Kest, nsim = 39)  
plot(K)
```



- Función `rhohat` de `spatstat`
  - Gráfica de intensidad en relación a covariable
  - Suavizada
- Implementada en lotes con `plotQuantIntens`

Uso de `plotQuantIntens`:

```
Q <- pixelquad(X = puntos.ppp, W = as.owin(w))
```

```
source("Funciones-spatstat/plotQuantIntens.R")
```

```
plotQuantIntens(imList = r.im,  
                noCuts = 5,  
                Quad = Q,  
                p.pp = puntos.ppp,  
                dir = "",  
                name = "Respuestas")
```

```
## pdf
```

```
## 2
```

## Ajuste del modelo

---



- Las variables ambientales no están correlacionadas
- Que la intensidad es log-lineal en relación a covariables

Se usa función ppm

```
names(r.im) <- paste("Var", 1:3, sep = ".")  
m1 <- ppm(Q = puntos.ppp,  
          trend = ~ Var.1,  
          covariates = r.im)
```

- Q es el proceso de puntos
- `trend` es la fórmula del modelo
- `covariates` es la lista de variables ambientales

## Resultados

```
summary(m1)

## Point process model
## Fitted to data: puntos.ppp
## Fitting method: maximum likelihood (Berman-
Turner approximation)
## Model was fitted using glm()
## Algorithm converged
## Call:
## ppm.ppp(Q = puntos.ppp, trend = ~Var.1, covariates = r.im)
## Edge correction: "border"
## [border correction distance r = 0 ]
## -----
-----
## Quadrature scheme (Berman-Turner) = data + dummy + weights
##
```

## Comparación de modelos

```
m2 <- ppm(Q = puntos.ppp,  
          trend = ~ Var.2,  
          covariates = r.im)
```

```
m3 <- ppm(Q = puntos.ppp,  
          trend = ~ Var.3,  
          covariates = r.im)
```

```
AIC(m1); AIC(m2); AIC(m3)
```

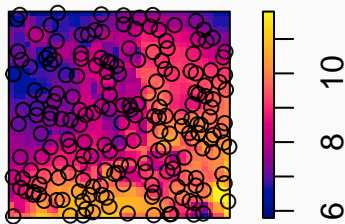
```
## [1] -458.1175
```

```
## [1] -459.3864
```

```
## [1] -455.3752
```

```
plot(m2, trend = T, se = F)
```

## Fitted trend



Para concluir

---

- Verificar significancia de efectos
- Verificar residuales (lo que el modelo no explicó)
- Supuestos
  - Pertinencia del “área de calibración”
  - Pertinencia del sistema de referencia de coordenadas
  - Supuesto de independencia rara vez se cumple
  - Proponer modelos de interacción o gaussianos