

Análisis de presencias con procesos de puntos

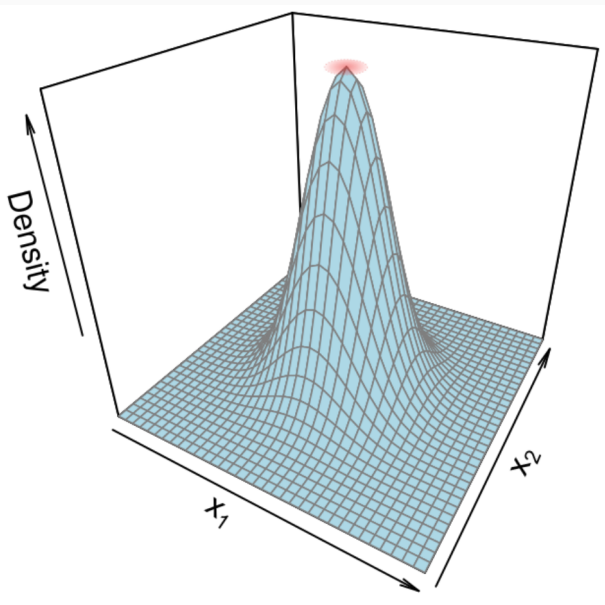
Tutorial intermedio de spatstat

Gerardo Martín

2022-06-29

Simulación de presencias

Especificación de un centroide



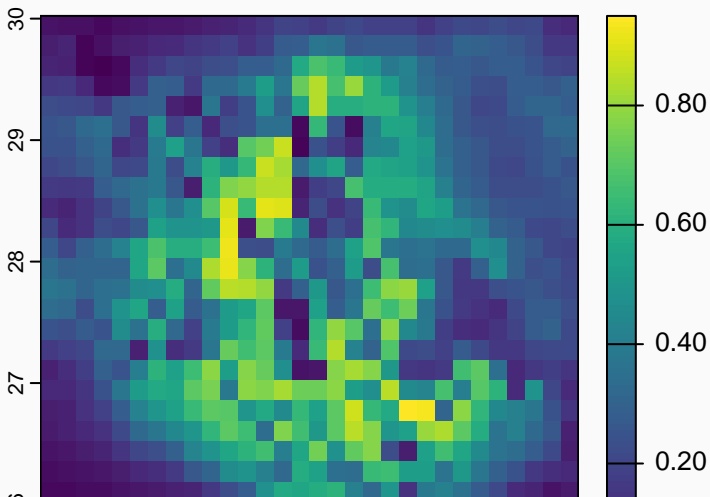
Código - generando favorabilidad “verdadera”

```
centroide <- global(r, mean)
r.df <- as.data.frame(r, xy = T)
covar <- cov(r.df[, 3:5])
md <- mahalanobis(r.df[, 3:5], center = centroide$mean, cov = covar)
head(md)
```

```
##           1           2           3           4           5           6
## 5.846738 6.383437 6.443874 7.296541 6.475630 6.066614
```

Código - viendo la favorabilidad

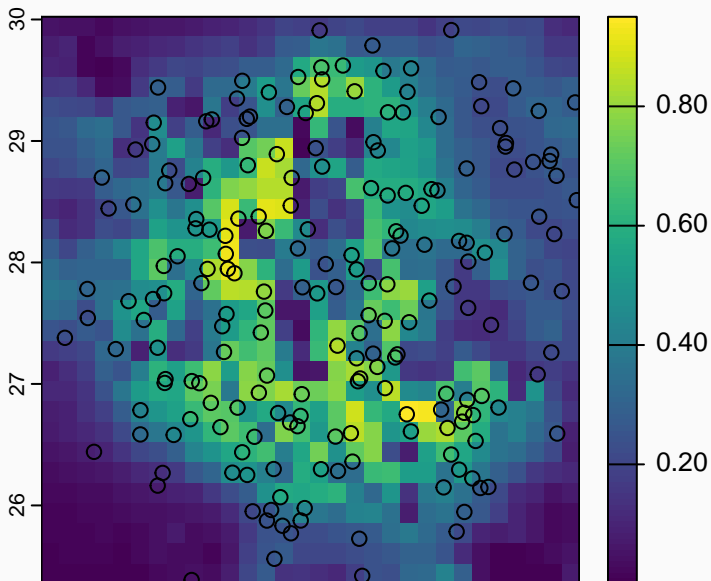
```
md.r <- rast(data.frame(r.df[, 1:2], md))  
md.exp <- exp(-0.5*md.r)  
plot(md.exp)
```



```
set.seed(182)
sam <- sample(1:nrow(r.df), 200, prob = exp(-0.5*md))
puntos.2 <- data.frame(r.df[, 1:2][sam,])
puntos.2$x <- puntos.2$x + rnorm(200, 0, 0.05)
puntos.2$y <- puntos.2$y + rnorm(200, 0, 0.05)
```

Código - favorabilidad y puntos

```
plot(md.exp); points(puntos.2)
```



Formateo para spatstat

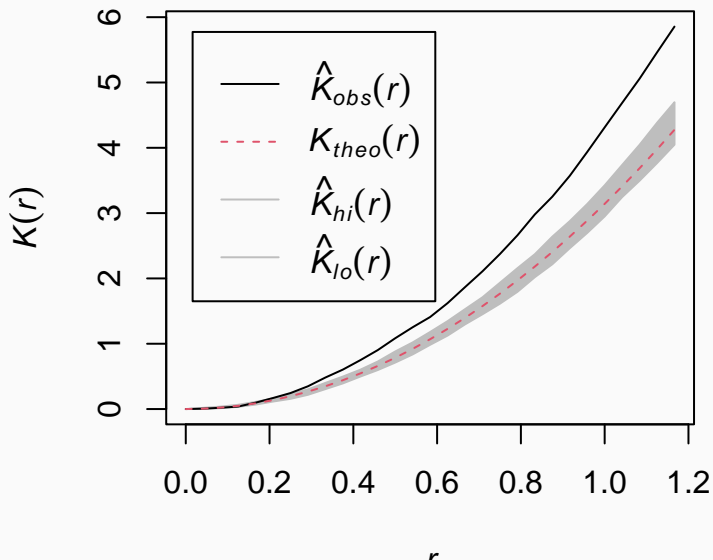
```
source("Funciones-spatstat/imFromStack.R")  
source("Funciones-spatstat/winFromRaster.R")  
source("Funciones-spatstat/plotQuantIntens.R")
```

```
r.im <- imFromStack(r)
names(r.im) <- paste("Var", 1:3, sep = ".")
w <- as.owin(r.im[[1]])
puntos.2.ppp <- ppp(x = puntos.2$x,
                    y = puntos.2$y,
                    window = w,
                    check = F)
Q <- pixelquad(X = puntos.2.ppp, W = as.owin(w))
```

Análisis exploratorio

```
K <- envelope(puntos.2.ppp, fun = Kest, nsim = 39)

## Generating 39 simulations of CSR ...
## 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 1
## 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36,
## 39.
##
## Done.
```

K

1. Pareciera que el proceso está levemente autocorrelacionado
2. No sabemos de momento si afectará al modelo
3. Debemos poner atención al modelo ajustado

```
plotQuantIntens(imList = r.im,  
                noCuts = 5,  
                Quad = Q,  
                p.pp = puntos.2.ppp,  
                dir = "",  
                name = "Respuestas-centroide")
```

```
## pdf
```

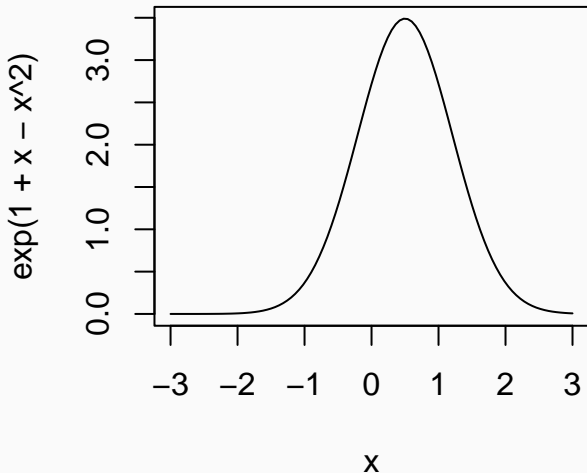
```
## 2
```

Ver archivo de gráficas

Consideraciones para proponer modelos

Curvas con forma de campana → fórmula cuadrática

```
curve(exp(1 + x - x^2), from = -3, 3)
```



Ecuación lineal:

$$y = \alpha + \beta_1 x_1 + \cdots + \beta_n x_n$$

Ecuación polinomial de 2º grado

$$y = \alpha + \beta_1 x_1 + \beta'_1 x_1^2 + \cdots + \beta_n x_n + \beta'_n x_n^2$$

Recordemos que $y = \log \lambda$

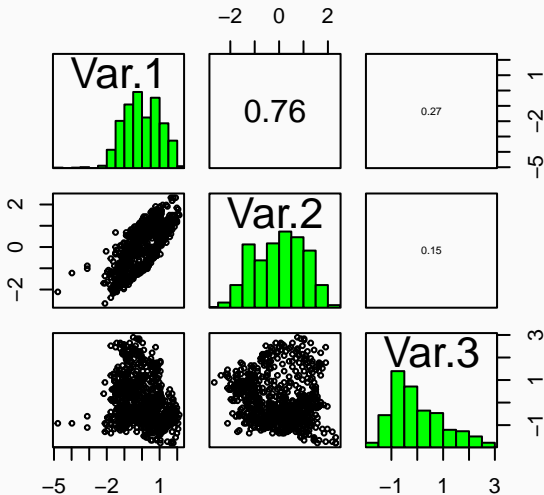
¿Qué variables podemos incluir en el mismo modelo?

Regla de oro: Aquellas que no estén correlacionadas

- Que x_1 no sea predictor de x_2
- No se puede atribuir efecto de x_1 ó x_2 sobre λ
- Necesitamos medir correlación entre pares de variables (**pairs**)

Medición de correlación entre covariables

`pairs(r)`



Podemos incluir en el mismo modelo:

1. Var.1 y Var.3
2. Var.2 y Var.3

Por lo tanto la fórmula polinomial

$$\log \lambda = \alpha + \beta_1 x_1 + \beta_1' x_1^2 + \beta_2 x_2 + \beta_2' x_2^2 +$$

En R:

1. `~ Var.1 + Var.3 + I(Var.1^2) + I(Var.3^2)`
2. `~ Var.2 + Var.3 + I(Var.2^2) + I(Var.3^2)`

```
m1 <- ppm(Q = puntos.2.ppp,  
          trend = ~ Var.1 + Var.3 + I(Var.1^2) + I(Var.3^2),  
          covariates = r.im)  
m2 <- ppm(Q = puntos.2.ppp,  
          trend = ~ Var.2 + Var.3 + I(Var.2^2) + I(Var.3^2),  
          covariates = r.im)
```

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

```
## [1] -509.172
```

```
## [1] -521.2113
```

Analizar los efectos estimados

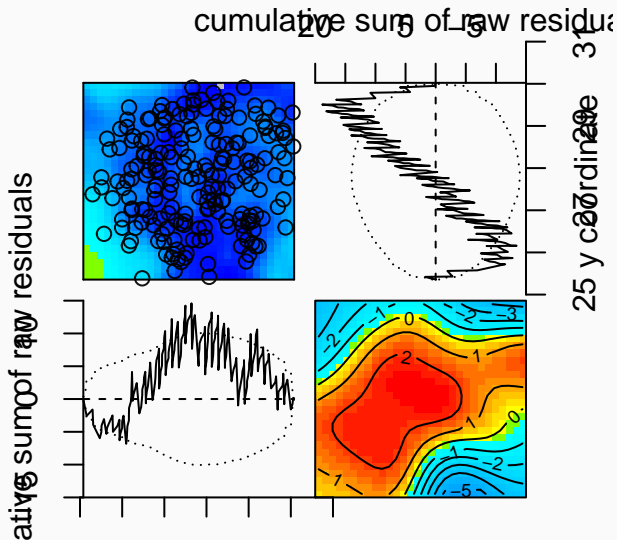
```
sum.m1 <- summary(m1)
knitr::kable(sum.m1$coefs.SE.CI[, 1:5])
```

	Estimate	S.E.	CI95.lo	CI95.hi	Ztest
(Intercept)	2.7089019	0.1116046	2.4901610	2.9276429	***
Var.1	0.1163265	0.0883434	-0.0568233	0.2894764	
Var.3	-0.2208983	0.1118989	-0.4402161	-0.0015805	*
I(Var.1^2)	-0.3046825	0.0875528	-0.4762828	-0.1330821	***
I(Var.3^2)	-0.5240862	0.1173717	-0.7541304	-0.2940420	***

Diagnóstico - Residuales

```
par(mar = c(2,2,2,2))
```

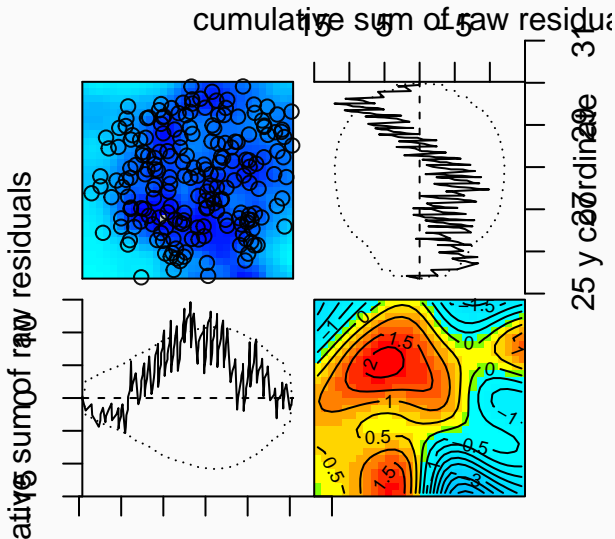
```
diagnose.ppm(m1, main = "", cex.axis = 0.25)
```



Diagnóstico - Residuales

```
par(mar = c(2,2,2,2))
```

```
diagnose.ppm(m2, main = "", cex.axis = 0.25)
```



Diagnóstico - Ripley

```
K1 <- envelope(m1, fun = Kest, nsim = 39)
```

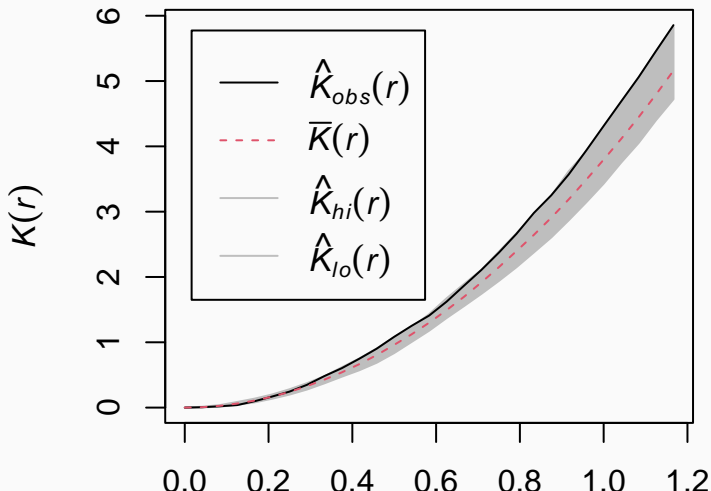
```
## Generating 39 simulated realisations of fitted Poisson model  
## 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 1  
## 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36,  
## 39.  
##  
## Done.
```

```
K2 <- envelope(m2, fun = Kest, nsim = 39)
```

```
## Generating 39 simulated realisations of fitted Poisson model  
## 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 1  
## 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36,  
## 39.  
##  
## Done.
```

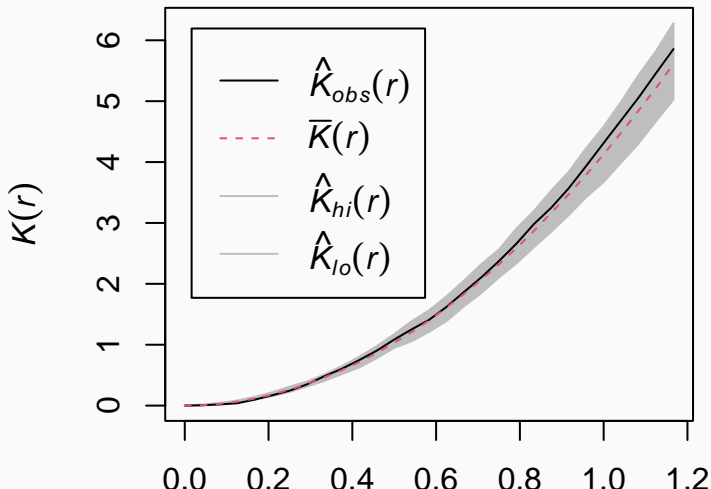
```
plot(K1, cex = 0.5)
```

K1



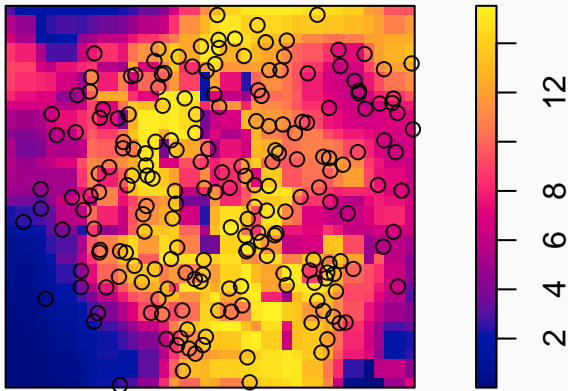
```
plot(K2, cex = 0.5)
```

K2



- AIC menor para **m1**
- Residuales dentro de tolerancia para **m1**
- Prueba de ripley correcta para ambos modelos
 - No parece necesario modelar autocorrelación (lo haremos a continuación)
- Evidencia *favorece* a **m1**

```
plot(m1, se = F, main = "")
```

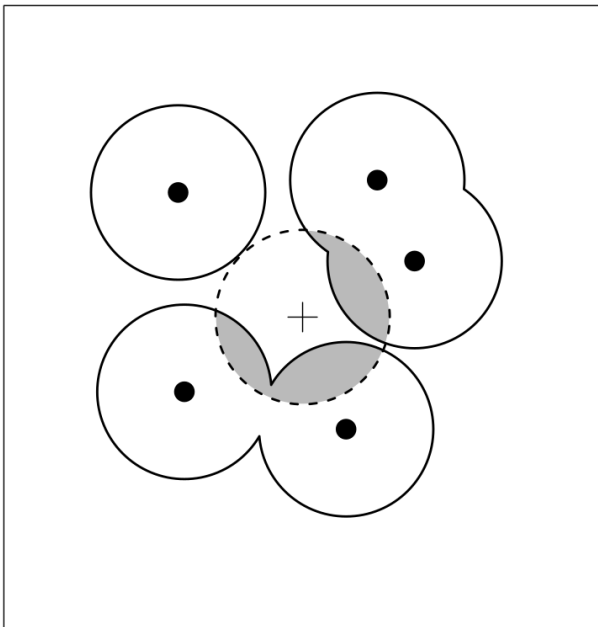


```
pred <- predict(m1)
pred.r <- rast(pred)
writeRaster(pred.r, "Predicción-m1.tif",
            overwrite = T)
```

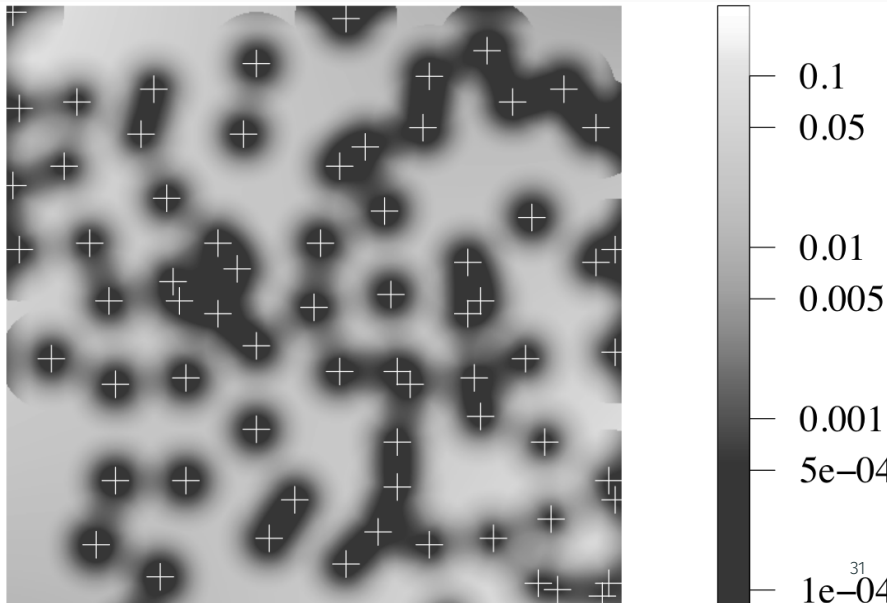
Modelando los efectos espaciales

- Estiman efecto aleatorio para puntos cercanos
- Sirven para procesos de exclusión o agregación moderada
- Hay varios tipos de interacciones entre puntos

¿Qué es interacción?



Tipos de interacciones



FUNCTION	MODEL
AreaInter	area-interaction process
BadGey	multiscale Geyer saturation process
Concom	connected component interaction
Geyer	Geyer saturation process
Hybrid	hybrid of several interactions
Ord	Ord model, user-supplied potential
OrdThresh	Ord model, threshold potential
Saturated	saturated model, user-supplied potential
SatPiece	multiscale saturation process
Triplets	Geyer triplet interaction process

Para generar un modelo de interacción

1. Establecer tamaño del búfer

```
rr <- data.frame(r=seq(1,5,by=1))
p <- profilepl(rr, Strauss,
               puntos.2.ppp ~ Var.1 + Var.3 + I(Var.1^2) + I(
               covariates = r.im, aic=F, rbord = 0.1)

## comparing 5 models...

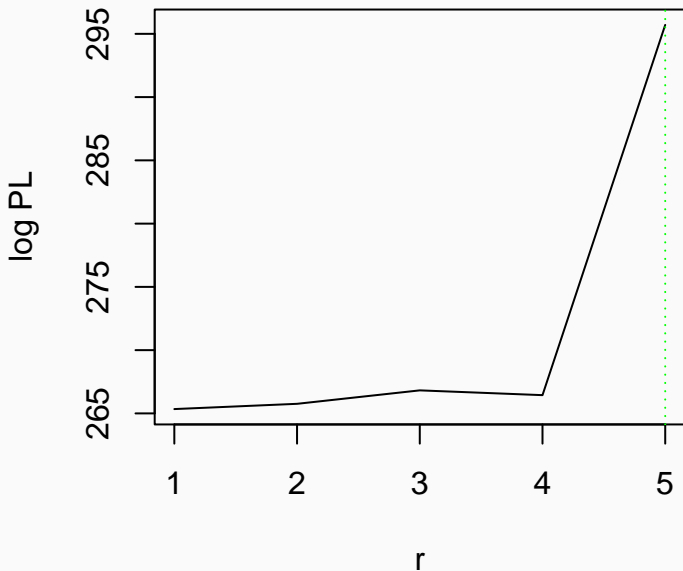
## 1, 2, 3, 4,
## 5.

## fitting optimal model...

## done.
```

Para generar un modelo de interacción

```
plot(p, main = "")
```



Para generar un modelo de interacción

Un radio de tamaño 2 minimiza la pseudo-verosimilitud, de modo que el modelo de interacción con la fórmula de m1 es:

```
m1.int <- ppm(Q = puntos.2.ppp,  
             trend = ~ Var.2 + Var.3 + I(Var.2^2) + I(Var.3^2),  
             covariates = r.im,  
             AreaInter(rr$r[p$iopt]), rbord = 0.1) #Interacción
```

```
sum.int <- summary(m1.int)
knitr::kable(sum.int$coefs.SE.CI[, 1:4])
```



```
coef(m1)
```

```
## (Intercept)      Var.1      Var.3 I(Var.1^2) I(Var.3^2)
##      2.7089019      0.1163265    -0.2208983    -0.3046825    -
0.5240862
```

```
coef(m1.int)
```

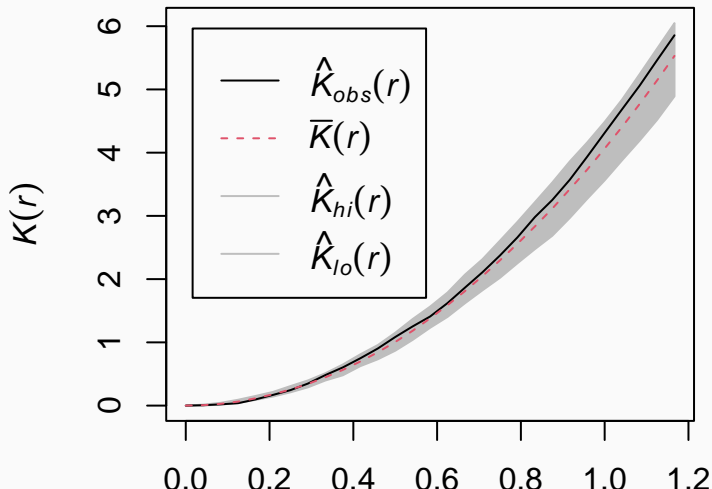
```
## (Intercept)      Var.2      Var.3 I(Var.2^2) I(Var.3^2) Inter
##      2.87549775      0.08478505    -0.00164637    -0.42014097    -
0.52671442           NA
```

```
K.int <- envelope(m1.int, Kest, nsim = 39)

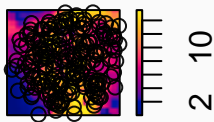
## Generating 39 simulated realisations of fitted Gibbs model .
## 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 1
## 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36,
## 39.
##
## Done.
```

`plot(K.int)`

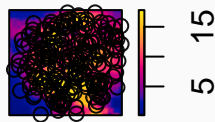
K.int



Poisson



Interacción



Proceso Cox log-Gaussiano

- En MPPs
 - Intensidad es explicada por covariables si
 - Covariables rara vez explican puntos agregados
- Gaussiano = Distribución normal
 - Efecto aleatorio con distribución normal multivariada

$$\log \lambda_i = \alpha + \beta_1 x_{1,i} + \cdots + G(u_i, v_i)$$

- α es el intercepto global - $G(u_i)$ es el intercepto aleatorio para cada píxel
- Cuando todas las $x = 0$, la intensidad en el píxel i es $\exp(\alpha + G(u_i))$

¿Con qué se ajusta un LGCP en R?

- Frecuentista - `spatstat` (rápido poco preciso)
- Bayesiano
 - `RINLA` (moderadamente rápido, moderadamente preciso)
 - `lgcp` (muuuuy lento, bastante preciso)
- Frecuentista son aproximaciones, y Bayesiano son estimaciones *verdaderas*


```
m1.lgcp <- kppm(puntos.2.ppp,  
               trend = ~ Var.2 + Var.3 + I(Var.2^2) + I(Var.  
               covariates = r.im,  
               clusters = "LGCP",  
               statistic = "K", # K de Ripley  
               method = "mincon") # Modelo de varianza
```

```
sum.lgcp <- summary(m1.lgcp)
knitr::kable(sum.lgcp$coefs.SE.CI[, 1:4])
```

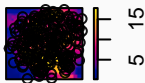
	Estimate	S.E.	CI95.lo	CI95.hi
(Intercept)	2.8710581	0.3194624	2.2449234	3.4971929
Var.2	0.1113775	0.0921967	-0.0693247	0.2920797
Var.3	0.0049808	0.1157026	-0.2217922	0.2317538
I(Var.2^2)	-0.4756438	0.1028281	-0.6771832	-0.2741044
I(Var.3^2)	-0.5258717	0.1119727	-0.7453342	-0.3064093

```
knitr::kable(sum.m1$coefs.SE.CI[, c(1, 2, 3, 4)])
```

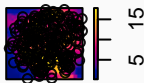
	Estimate	S.E.	CI95.lo	CI95.hi
(Intercept)	2.7089019	0.1116046	2.4901610	2.9276429
Var.1	0.1163265	0.0883434	-0.0568233	0.2894764
Var.3	-0.2208983	0.1118989	-0.4402161	-0.0015805
I(Var.1^2)	-0.3046825	0.0875528	-0.4762828	-0.1330821
I(Var.3^2)	-0.5240862	0.1173717	-0.7541304	-0.2940420

```
par(mfrow = c(1, 3))  
plot(m2, se = F, trend = T, main = "Poisson")  
plot(m1.int, se = F, trend = T, cif = F, main = "Interacción")  
plot(m1.lgcp, what = "intensity", main = "LGCP")
```

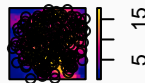
Poisson



Interacción



LGCP

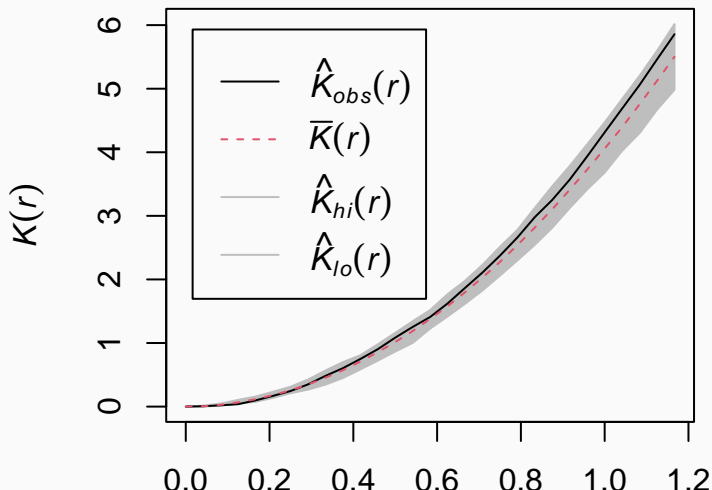


```
K.lgcp <- envelope(m1.lgcp, Kest, nsim = 39)

## Generating 39 simulated realisations of fitted cluster model
## 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 1
## 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36,
## 39.
##
## Done.
```

`plot(K.lgcp)`

K.lgcp



Conclusiones

- Modelo Poisson
 - Más simple, y no parece tener problemas
 - IC de estimaciones más amplios que LGCP
- Interacción
 - IC más amplios que MPP
- LGCP
 - Función K más cercana a expectativa teórica ### Alternativas de modelación
- Respuestas bisagra: Regresión por partes
- Respuestas no lineales: Suavizadores GAM
- Interacciones entre variables