Análisis de presencias con procesos de puntos

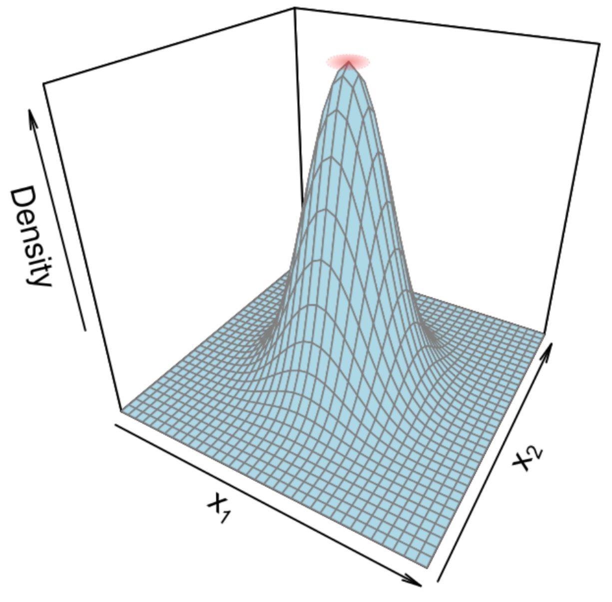
Tutorial intermedio de spatstat

Gerardo Martín

2022-06-29

Simulación de presencias

Especificación de un centroide



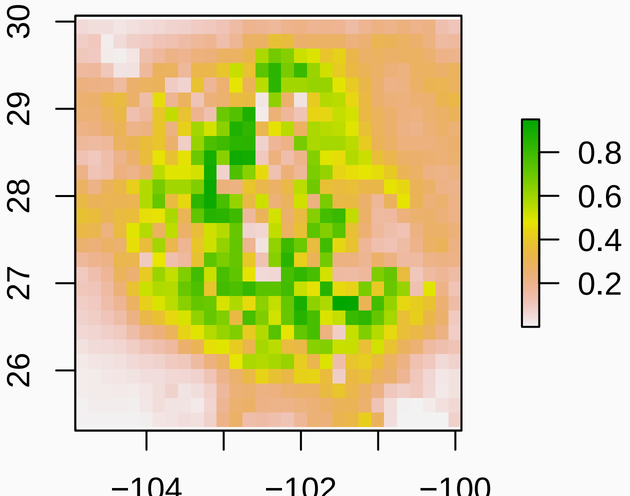
Código - generando favorabilidad “verdadera”

```
centroide <- cellStats(r, mean)
r.df <- data.frame(rasterToPoints(r))
covar <- cov(r.df[, 3:5])
md <- mahalanobis(r.df[, 3:5], center = centroide, cov = cova
head(md)
```

```
## [1] 5.846738 6.383437 6.443874 7.296541 6.475630 6.066614
```

Código - viendo la favorabilidad

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



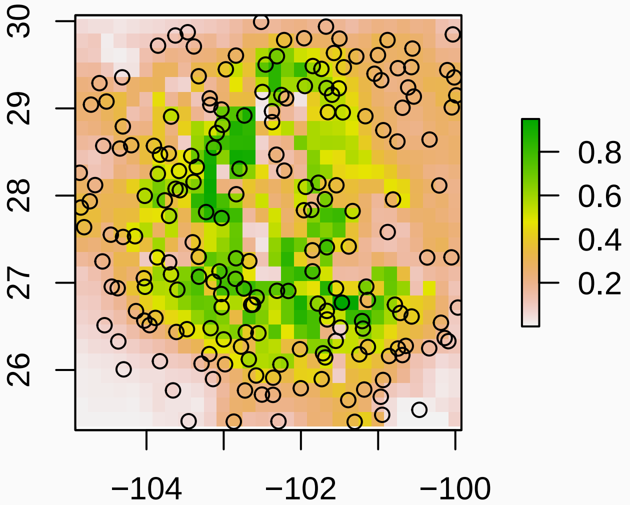
Código - simulando los puntos

```r
set.seed(182)
puntos.2 <- dismo::randomPoints(mask = md.exp,
                                n = 200,
                                prob = T)
```

```
## Warning in .couldBeLonLat(x, warnings = warnings): CRS is NA.
## longitude/latitude
```

```r
puntos.2 <- data.frame(puntos.2)
puntos.2$x <- puntos.2$x + rnorm(200, 0, 0.05)
puntos.2$y <- puntos.2$y + rnorm(200, 0, 0.05)
```

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



Formateo para spatstat

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

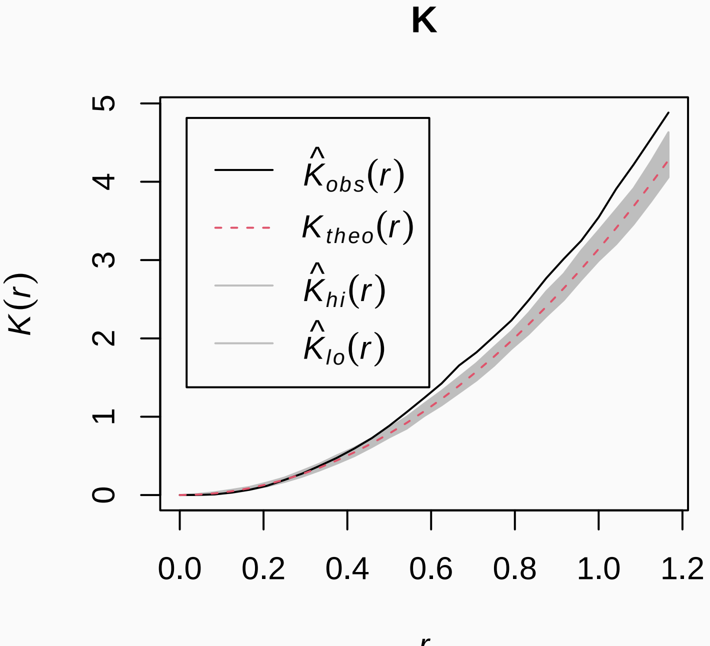
Formateo rápido

```r
r.im <- imFromStack(r)
w <- winFromRaster(r)
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
##
## Done.
```


K
$\hat{K}_{obs}(r)$
$K_{theo}(r)$
$\hat{K}_{hi}(r)$
$\hat{K}_{lo}(r)$
$K(r)$
0
1
2
3
4
5
0.0
0.2
0.4
0.6
0.8
1.0
1.2
r

Autocorrelación - notas

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

Respuestas a variables

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

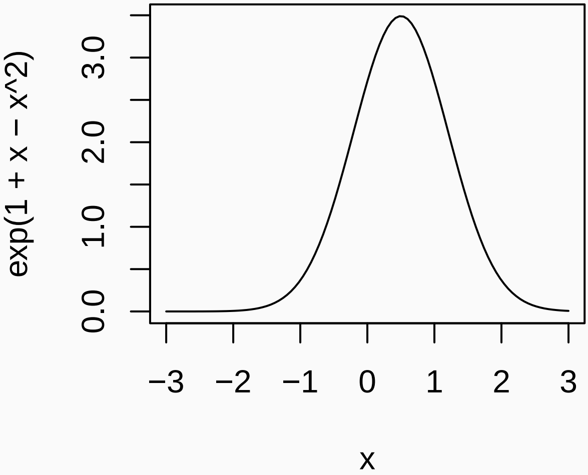
```
## pdf
##   2
```

Ver archivo de gráficas

Consideraciones para proponer modelos

Curvas con forma de campana $\longrightarrow$ 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 2^o grado

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

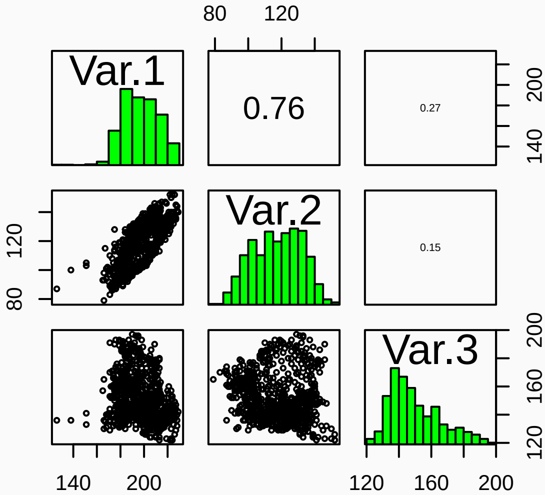
Recordemos que $y = \log \lambda$

¿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`)

```
pairs(r)
```


Var.1
0.76
0.27
Var.2
0.15
Var.3
80
120
200
140
120
80
200
160
120
140
200
120
160
200

Variables *compatibles*

Podemos incluir en el mismo modelo:

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

Por lo tanto las 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)`

Ajustando los modelos

```r
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)
```

Comparando los modelos

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

```
## [1] -473.2666
```

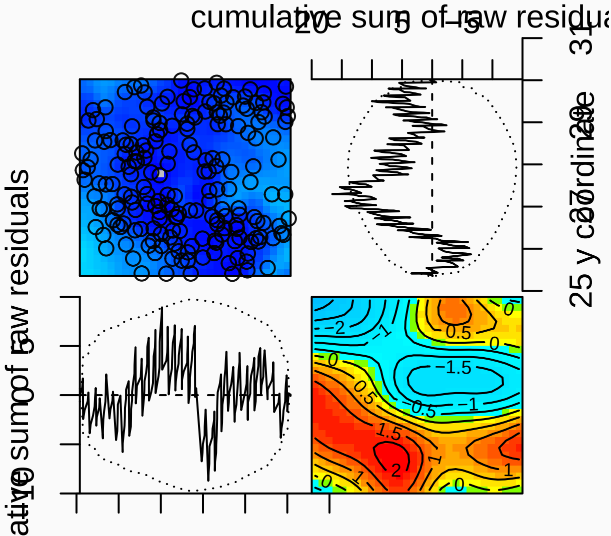
```
## [1] -468.7745
```

```
summary(m1)
```

```
## Point process model
##   Fitting   method:   maximum   likelihood   (Berman-
Turner approximation)
## Model was fitted using glm()
## Algorithm converged
## Call:
## ppm.ppp(Q = puntos.2.ppp, trend = ~Var.1 + Var.3 + I(Var.1^2)
##     I(Var.3^2), covariates = r.im)
## Edge correction: "border"
##  [border correction distance r = 0 ]
## -------------------------------------------------
--------------------------
## Quadrature scheme (Berman-Turner) = data + dummy + weights
##
```

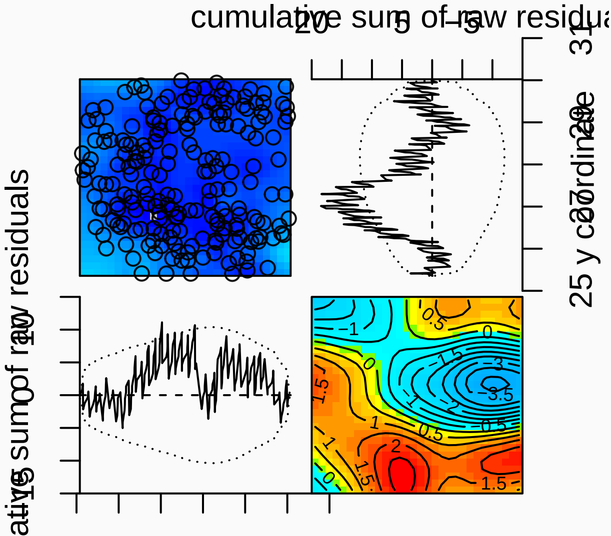
Diagnóstico - Residuales

```r
par(mar = c(2,2,2,2))
diagnose.ppm(m1, main = "", cex.axis = 0.25)
```



Diangnóstico - Residuales

```
par(mar = c(2,2,2,2))
diagnose.ppm(m2, main = "", cex.axis = 0.25)
```



Diagnóstico - Ripley

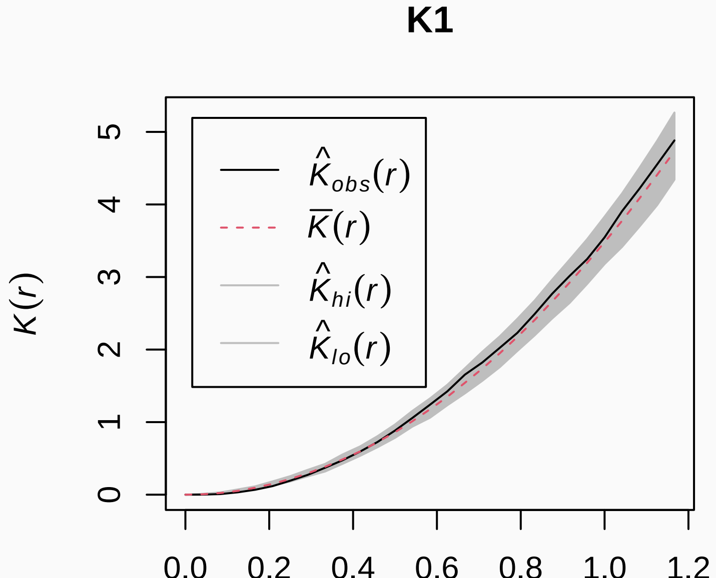
```r
K1 <- envelope(m1, fun = Kest, nsim = 39)
```

```
## 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
##
## Done.
```

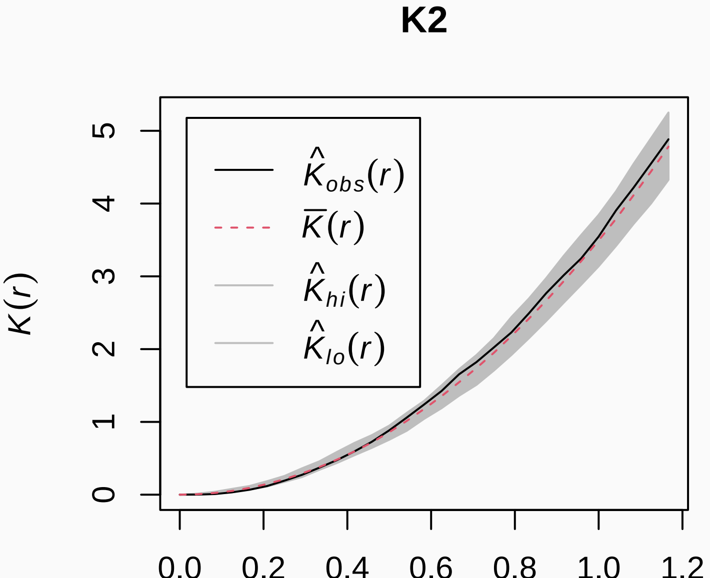
```r
K2 <- envelope(m2, fun = Kest, nsim = 39)
```

```
## 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
##
## Done.
```

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



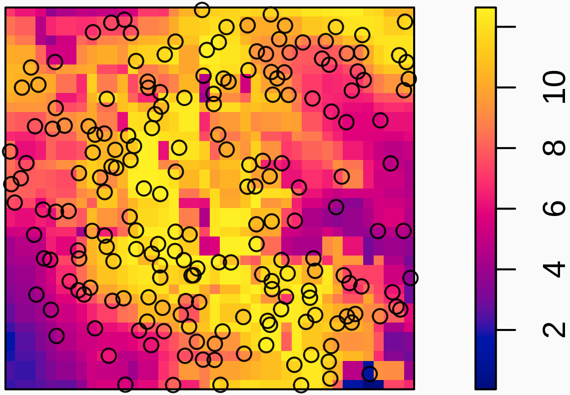
```r
plot(K2, cex = 0.5)
```



Resumen del análisis

- AIC menor para `m1`
- Residuales dentro de tolerancia para `m1`
- Prueba de ripley correcta para ambos modelos
    - No parece necesario modelar autocorrelación
- Evidencia *favorece* a `m1`

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



Guardando los resultados

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

```
## Warning in .gd_SetProject(object, ...): NOT UPDATED FOR PROJ
```

Alternativas de modelación

- Respuestas “hinge”: Regresión por partes
- Respuestas no lineales: Suavizadores GAM
- Interacciones entre variables
- LASSO con paquete `ppmlasso` (Warton y Renner 2013)