Análise de processos pontuais em dados de Citrus

Elias T. Krainski & Paulo J. Ribeiro Jr.

Última Atualização: 2 de agosto de 2006

Na análise de processos pontuais, é analizado a ocorrência de plantas doentes. Podemos analizar os padrão espacial analizando as plantas doentes e os vizinhos doentes, usando a distância mínima média ou o número de vizinhos doentes. Também foram adaptados alguns métodos de análise implementados no pacote **splancs** (Rowlingson, Diggle, adapted, packaged for R by Roger Bivand, pcp functions by Giovanni Petris & goodness of fit by Stephen Eglen 2006). Os métodos adaptados foram a suavização por kernel e da função K de Ripley e respectivo envelope simulado.

1 Distância mínima média

Defina-se distância mínima a distância entre uma planta e a planta doente mais próxima a ela. O número de vizinhos próximos dado um raio é o número de plantas doentes a menos da distância do raio da planta. Calculando-se estas estatísticas para cada planta doente, obtemos a distância mínima média entre plantas doentes e o número de vizinhos próximos de plantas doentes.

Para a análise, aplica-se um teste Monte Carlo, comparando o valor observado no conjunto de dados e com valores obtidos sob hipótese nula de aleatoriedade espacial completa. O p-valor no teste de distância mínima média é pela proporção de valores sob a hipótese nula menores ou iguais ao valor observado; no teste dos vizinhos próximos é dado pela proporção de valores sob hipótese nula, maiores ou iguais ao valor observado.

A função `mmdist.test()` faz a análise por distância mínima e recebe como entrada os dados no formato `geodata`, o código atribuido às plantas doentes e o número de simulações.

Carregando um conjuntode dados de morte súbita de Citrus.

```
> data(v303.geo)
```

Vendo os argumentos da função

```
> args(mmdist.test)
```

```
function (obj, NMC = 99, evaluation = 1, death = 1, healt = 0)
NULL
```

Fazendo a análise:

```
> mmdt <- mmdist.test(v303.geo, NMC = 199, death = 1:3, evaluation = 1:3)
```

```
test evaluation: 1 2 3
```

```
> summary(mmdt)
```

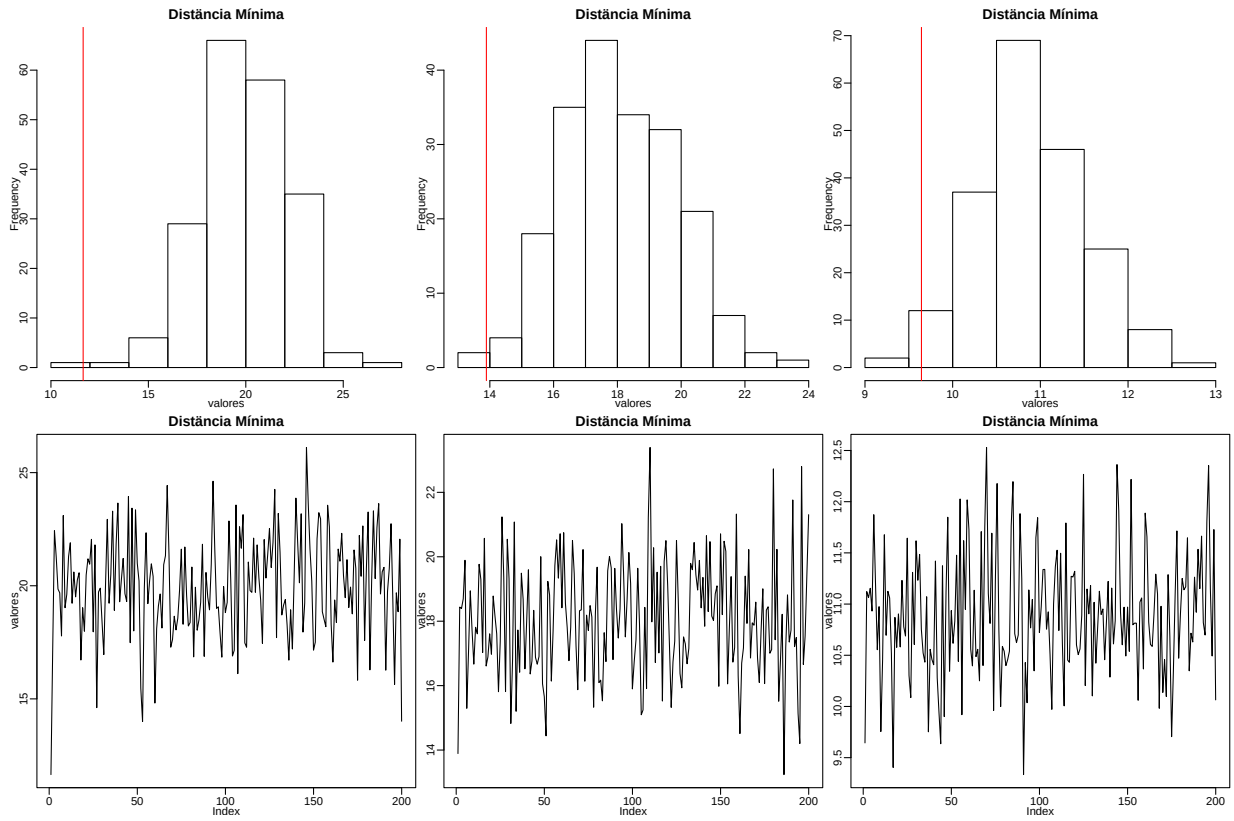

Figura 1: Vizualização dos resultados do teste de Monte Carlo para a distância mínima média.

```
Results for 199 Monte Carlo simulations!
Observed: 11.7 13.9 9.64
Randoms:
         Av1  Av2   Av3
Min.    14.0 13.2  9.33
1st Qu. 18.5 16.8 10.50
Median  19.9 17.9 10.90
Mean    20.0 18.1 10.90
3rd Qu. 21.6 19.4 11.20
Max.    26.1 23.4 12.50
P-value: 0.005 0.01 0.02
```

Pode-se também fazer o histograma e o plot das distâncias (Figura 1) usando os comandos:

```
> par(mfrow = c(2, 3), mar = c(2, 2, 2, 0.1), mgp = c(1.2,
+     0.5, 0))
> hist(mmdt, main = "Distäncia Mínima", evaluation = 1:3)
> plot(mmdt, main = "Distäncia Mínima", evaluation = 1:3)
```

2 Número médio de vizinhos doentes

Na análise do número de vizinhos próximos, pode-se usar a função `neigh.test`. Nesta função deve-se entrar com os dados em formato `geodata`, o número de simulações, o(s) código(s) que identifica(m) plantas doentes e o raio de vizinhança. Na Figura 2 vizualiza-se os resultados.

```
> args(neigh.test)

function (obj, NMC = 99, death = 1, ray = 30, evaluation = 1,
    edge.cor = TRUE)
NULL

> neit <- neigh.test(v303.geo, NMC = 199, death = 1:3, evaluation = 1:3)

test evaluation: 1 2 3

> summary(neit)

Results for 199 Monte Carlo simulations!
Observed: 3.61 4.12 10.1
Randoms:
          Av1  Av2  Av3
Min.    0.983 1.34 5.73
1st Qu. 1.770 2.11 6.47
Median  2.060 2.39 6.77
Mean    2.100 2.43 6.79
3rd Qu. 2.410 2.70 7.07
Max.    3.610 4.14 8.48
P-value: 0.01 0.01 0.005

> par(mfrow = c(2, 3), mar = c(2, 2, 2, 0.1), mgp = c(1.2,
+     0.5, 0))
> hist(neit, main = "Número de Vizinhos", evaluation = 1:3)
> plot(neit, main = "Número de Vizinhos", evaluation = 1:3)
```

3 Suavização por *kernel*

O kernel para a incidência de plantas doentes, pode ser feito usando a função `kernel2d.citrus()`. Esta função faz o *kernel* uma ou mais avaliações feitas. Os parâmetros de suavização são os mesmos da função `kernel2d()` do pacote **splancs**.

Na função `kernel2d.citrus()` deve-se entrar com os dados usando o argumento `pts`. O polígono envolvente da área pode ser informado no argumento `poly` ou selecionar uma das opções do argumento `borders` para a definição automática do polígono envolvente.

```
> args(kernel2d.citrus)

function (obj, poly = NULL, h0, nx = 100, ny = 100, evaluation = 1,
    death = 1, ratio = TRUE, kernel = c("quartic", "gaussian"))
NULL
```

No exemplo, consideramos as avaliações 19 a 21 e que todas as plantas com códigos 1, 2 ou 3 são as que têm incidência.

```
> ker20 <- kernel2d.citrus(v303.geo, h0 = 20, eval = 19:21,
+     death = 1:3)
```

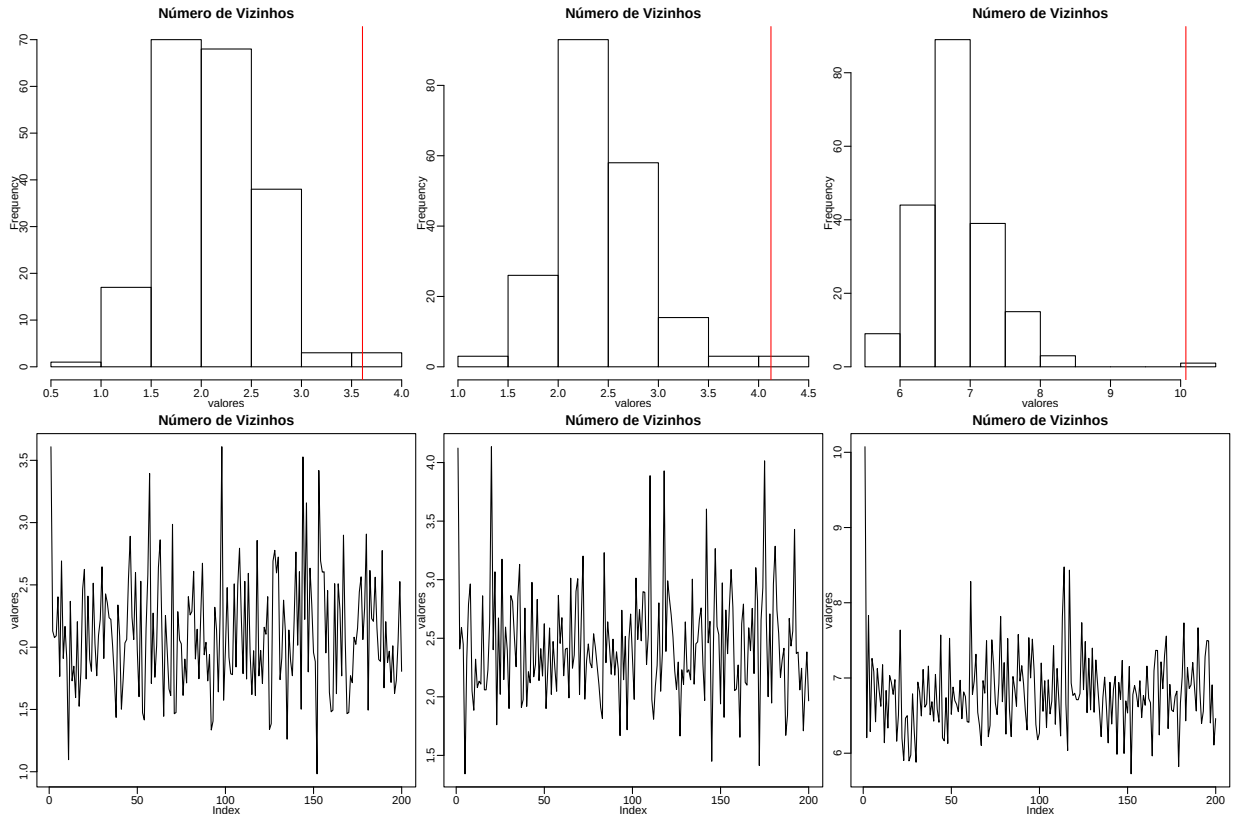

Figura 2: Vizualização dos resultados do teste de Monte Carlo para o número médio de vizinhos doentes.

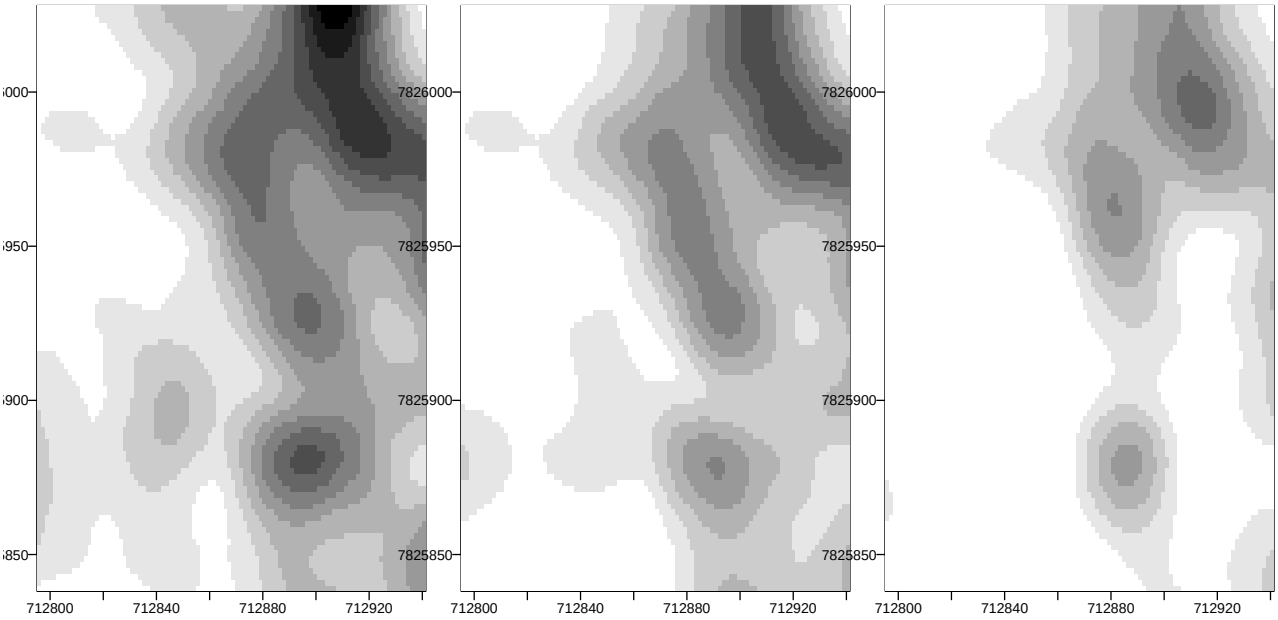

Figura 3: Kernel com Escala de Cores Global

Também foi adaptada a função `image` para vizualizar o *kernel*. Podemos vizualizar o kernel de dois modos diferentes, usando uma escala de cores global para todas as avaliações ou usando uma escala local.

```
> par(mfrow = c(1, 3), mar = c(2, 2, 1, 0.1), mgp = c(1, 0.5,
+     0), las = 1)
> image(ker20, zlim = "global", col = gray(seq(0, 1, l = 11)))
```

Usando escala global, podemos vizualizar regiões de maior intensidade e também o aumento da incidência ao longo do tempo. Usando uma escala individual, para cada avaliação, podemos vizualizar mais nitidamente as regiões de maior incidência dentro do talhão.

```
> par(mfrow = c(1, 3), mar = c(2, 2, 1, 0.1), mgp = c(1, 0.5,
+     0), las = 1)
> image(ker20, zlim = "individual", col = gray(seq(0, 1, l = 11)))
```

O Kernel em três dimensões (espaço-temporal) pode ser feito utilizando a função `kernel3d()` do **splancs**. Porém, os dados precisam ser convertidos para uma matriz contendo três colunas: coordenadas das plantas doentes em cada avaliação e o tempo. Para isso, foi implementada a função `as.Tpoints()` para converter os dados. Como exemplo, tomamos as avaliações 19 a 21 e aproveitamos alguns resultados do kernel espacial.

```
> a <- as.Tpoints(v303.geo, death = 1:3, eval = 19:21, ref.time = "01/01/2001",
+     form.date = "dd/mm/yyyy")
> k3 <- kernel3d(a[, 1:2], a[, 3], ker20[[1]]$x, ker20[[1]]$y,
+     seq(510, 750, by = 30), 20, 100)
```

```
> par(mfrow = c(3, 3), mar = c(0.3, 0.3, 0.1, 0.1), mgp = c(0.2,
+     0.1, 0))
> for (i in 1:9) image(k3$xgr, k3$ygr, k3$v[, , i], asp = 1,
+     xlab = "", ylab = "", main = 0 + i, col = gray(seq(0,
+         1, l = 11)))
```

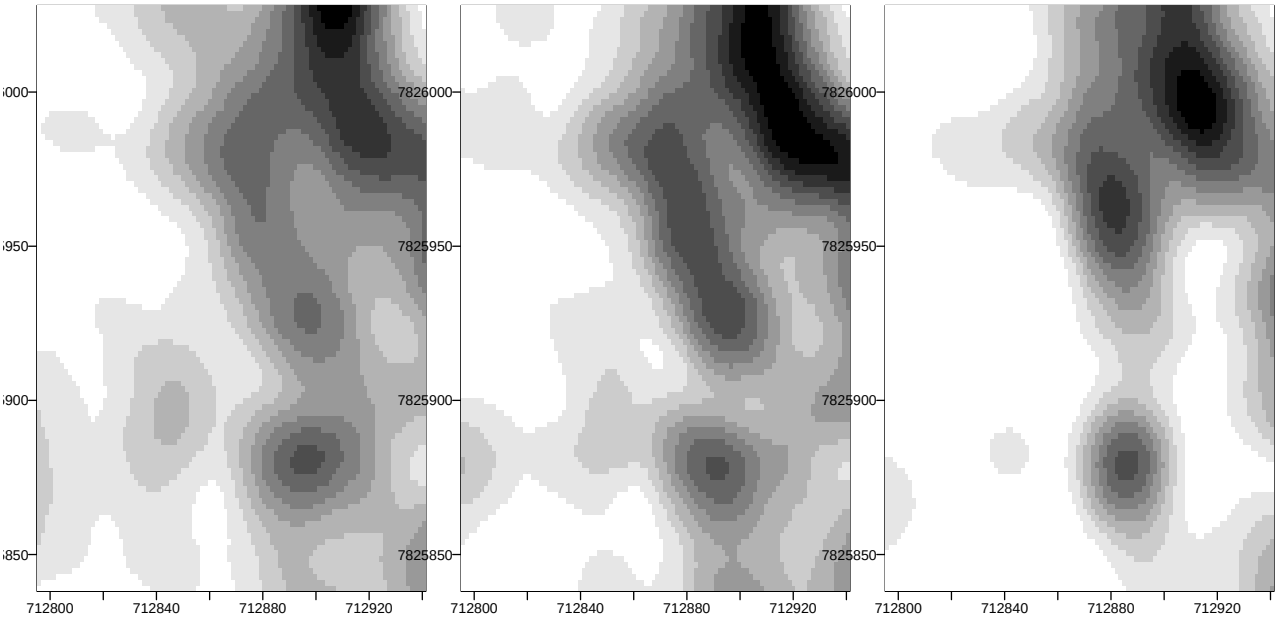

Figura 4: Kernel com Escala de Cores Global

4 Função K de Ripley

A função K, $K(t)$ é dada por

$$K(t) = \lambda^{-1}\mathrm{E}[N_0(t)]$$

, onde $N_0(t)$ é o número de eventos distante menos de t de um evento arbitrario. Esta função é mais comumente estimada utilizando-se a correção do efeito de borda proposta por Ripley (Diggle 2003).

O envelope de confiança para o padrão aleatório é contruido estimando a função K sob padrão de aleatoriedade espacial completa, para várias realizações desse processo em um poligno de mesma forma e tamanho que o observado nos dados, com número de pontos sendo o observado nos dados.

Podemos construir o envelope de confiança da função K de Ripley usando a função `Kenv.csr.citrus`. Os argumentos dessa função são:

```
> args(Kenv.csr.citrus)

function (nptg, poly = NULL, nsim = 49, s = seq(1, 50, 3), quiet = FALSE,
    evaluation = 1, borders = c("sbox", "bbox", "chull"), death = 1)
NULL
```

Entra-se com os dados em formato citrus e com o vetor de códigos que definem os eventos em estudo, argumentos `nptg` e `death` respectivamente. O poligono envolvente pode ser informado no argumento `poly` ou definido a partir dos dados, usando-se a uma das três funções: `chull`, `bbox` ou `sbox`, escolhida no argumento `borders`. O vetor para definir os raios para os quais será calculado os valores da função K é informado no argumento `s`. No argumento `evaluation` pode-se informar em qual avaliação será avaliada.

```
> Ken <- Kenv.csr.citrus(v303.geo, evaluation = 1:25, death = 1:3,
+     nsim = 19)
```

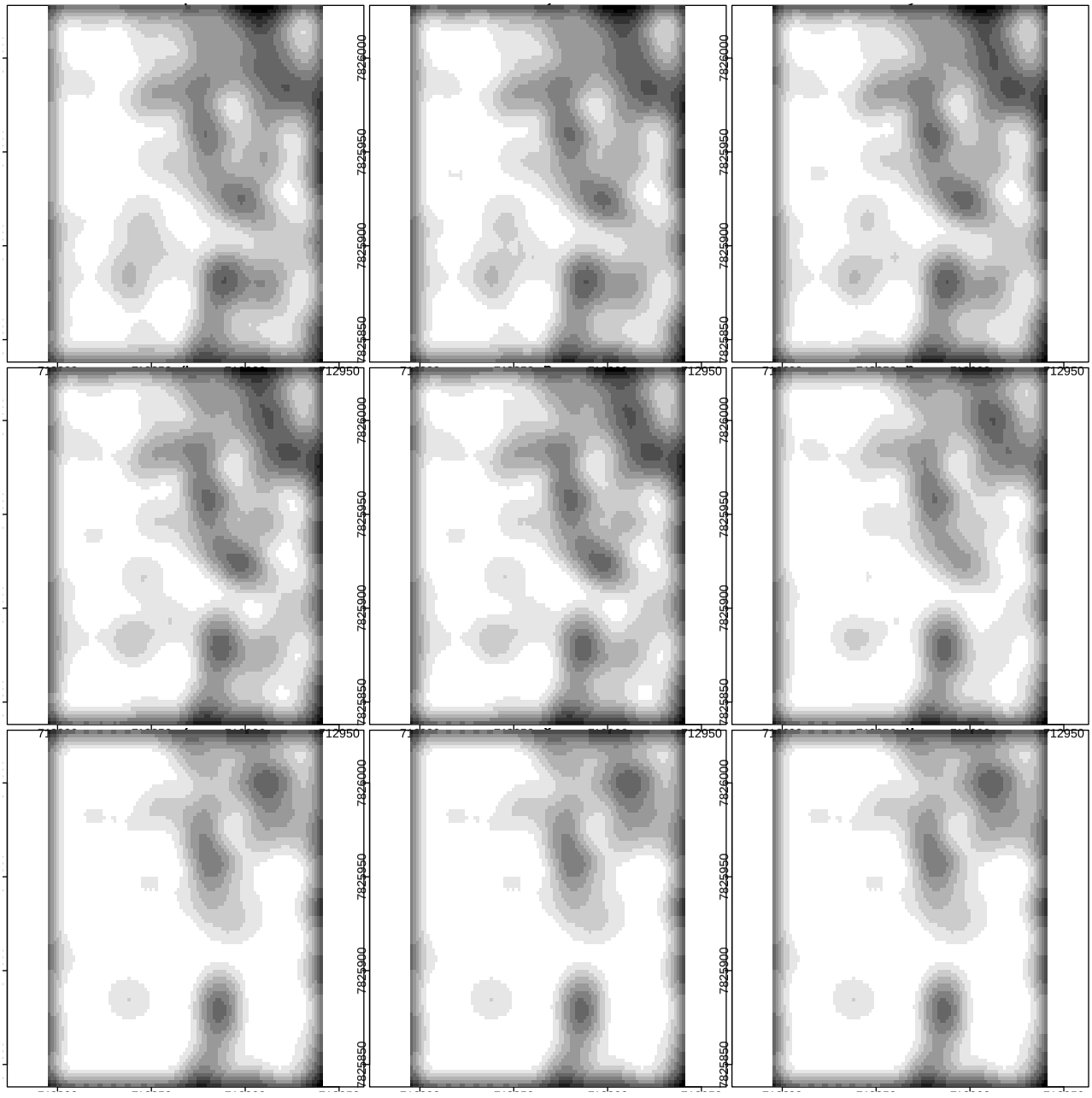

Figura 5: Kernel com Escala de Cores Global

A vizualização gráfica é mais facilmente interpretada vizualizando-se os valores padronizados da função K, através de

$$K^*(t) = \sqrt{K(t)/\pi} - t$$

. Usando a função `plot` no resultado da função `Kenv.csr.citrus`, podemos ver o envelope para os valores padronizados. Isto é ilustrado na Figura 6 prodizida com os comandos a seguir.

```
> par(mfrow = c(5, 5), mar = c(1.5, 1.5, 1.5, 0.1), mgp = c(1,
+     0.5, 0), las = 1)
> plot(Ken, main = "", xlab = "", ylab = "")
```

No gráfico vizualiza-se a intensidade média no talhão, que é a proporção de plantas doentes no talhão em cada avaliação.

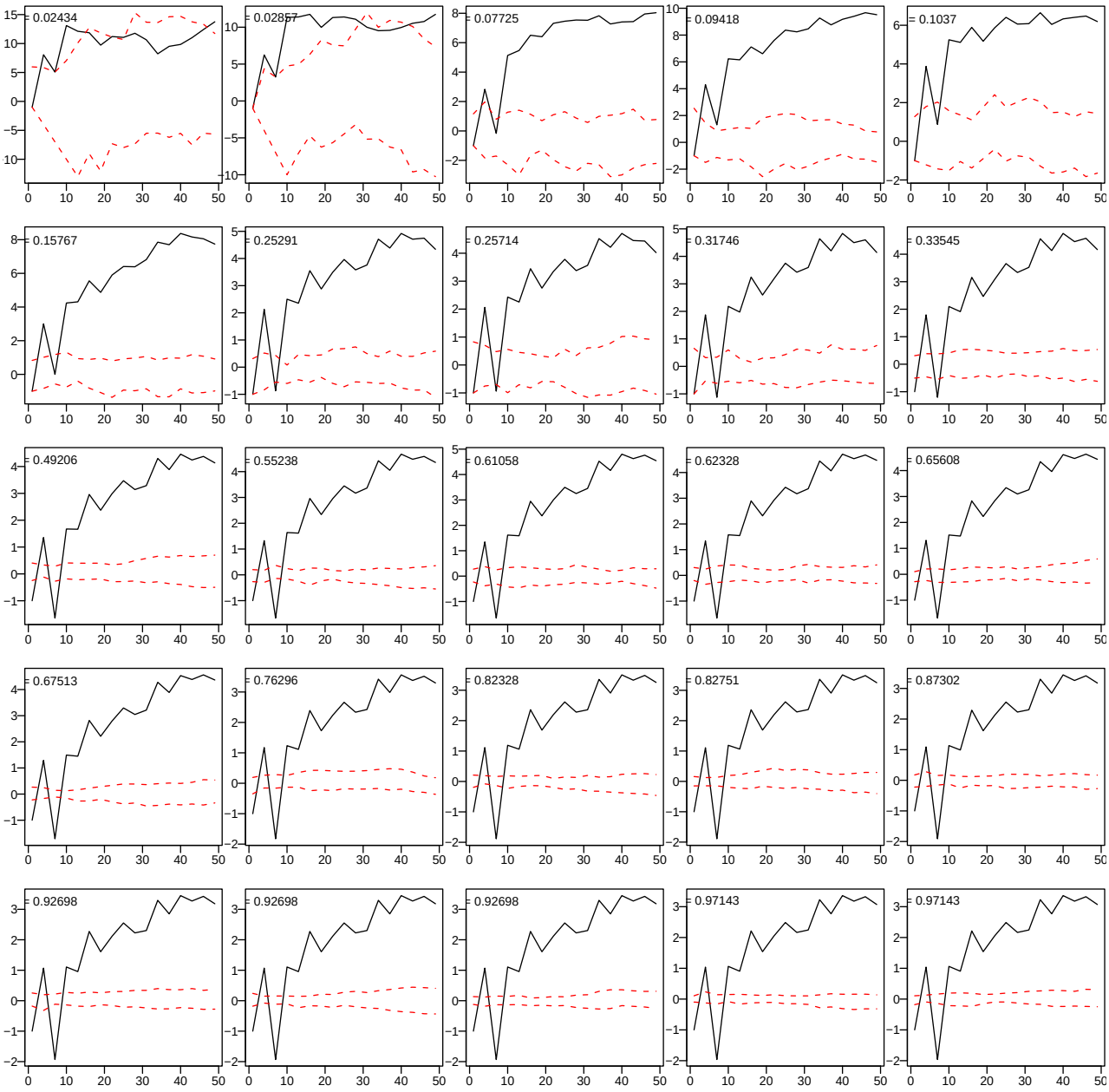

Figura 6: Envelope de Confiança da Função K de Ripley

Agradecimentos

Este trabalho foi desenvolvido como parte das atividades do convênio firmado entre o Fundo de Defesa da Citricultura (FUNDECITRUS) e o Departamento de Estatística da Universidade Federal do Paraná e financiado pelo FUNDECITRUS.

Referências

Diggle, P. J. (2003). *Statistical Analysis of Spatial Point Patterns*, Oxford University Press Inc.

Rowlingson, B., Diggle, P., adapted, packaged for R by Roger Bivand, pcp functions by Giovanni Petris & goodness of fit by Stephen Eglen (2006). *splancs: Spatial and Space-Time Point Pattern Analysis*. R package version 2.01-17.
URL: *http://www.maths.lancs.ac.uk/ rowlings/Splancs/*">*http://www.maths.lanc
wlings/Splancs/