

Análise por *quadrat counts* em R

Elias T. Krainski & Paulo J. Ribeiro Jr.

Última Atualização: 2 de agosto de 2006

A análise por *quadrats* é uma técnica de análise estatística bastante simples, cujo objetivo é avaliar o padrão espacial de incidência de doenças em plantas. Para esta análise estão implementados até o momento o índice de dispersão binomial, o índice de dispersão Poisson e o modelo beta-binomial. Para uma análise conjunta de várias avaliações, também foi implementada a estimação da lei de Taylor, (Madden & Hughes 1995).

1 Contagens por *quadrats* - *quadrat counts*

Um *quadrat* é uma sub-região definida a partir do número de linhas e número de plantas nas linhas a serem abrangidas. Os dados de *Pinta Preta dos Citrus* de Itajobi serão usados como exemplo, cujo mapa está na Figura 1.

Carregando os dados:

```
> data(gItajobi)
> plot(gItajobi, pch = 20)
```

A função `quadrat.count()` implementa a contagem por *quadrats*. Os argumentos dessa função são:

```
> args(quadrat.count)
```

```
function (obj, dx, dy = dx, complet = TRUE, random = FALSE, N = NULL,
  p.quadrats = 1, p.quadrats.random = FALSE, criteria = 0.1,
  death = 1, healt = 0)
NULL
```

No *help* pode ser vista uma descrição detalhada de cada argumento. Os principais são `obj` que indica o objeto que contém os dados, `dx` que indica o número de linhas abrangido pelo *quadrat* e `dy` que indica o número de plantas na linha abrangido pelo *quadrat*. Por default `dy` é igual à `dx`, bastando informar apenas este argumento.

Com esta função podem ser avaliados todos os possíveis *quadrats* na dimensão especificado pelo número de linhas e plantas na linha. O objeto retornado é uma lista contendo o número de plantas doentes em cada *quadrat*, o número total de plantas em cada *q* e o número nominal de plantas em cada *quadrat*.

Contagem por *quadrats* 20 × 20:

```
> quadrat.count(gItajobi, 20)
```

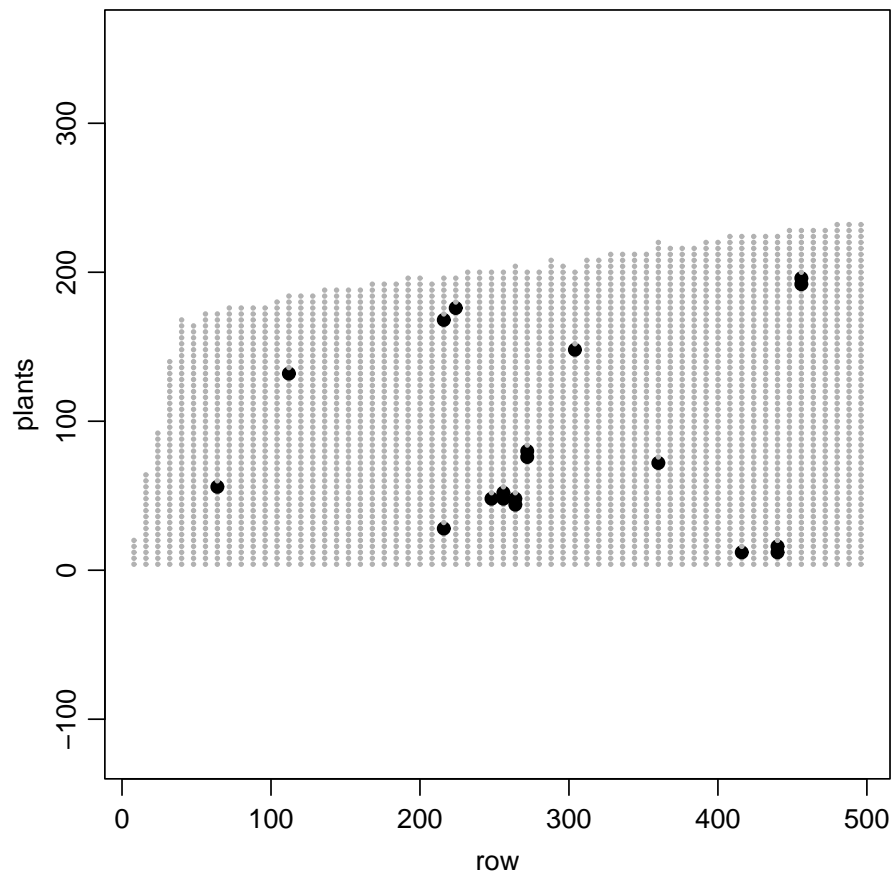


Figura 1: Incidência de pinta preta dos *Citrus* no talhão de Itajobi.

```
$y
      [,1] [,2] [,3]
[1,]    1    1    0
[2,]    8    1    2
[3,]    4    0    2
```

```
$n
      [,1] [,2] [,3]
[1,]  381  338   76
[2,]  400  400  196
[3,]  400  400  305
```

```
$dq
[1] 400
```

Observa-se no resultado o efeito da irregularidade das bordas.

Contagem por *quadrats* 20×10 :

```
> quadrat.count(gItajobi, 20, 10)
```

```
$y
      [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    0    1    0    1    0    0
[2,]    1    7    0    1    2    0
[3,]    3    1    0    0    2    0
```

```
$n
      [,1] [,2] [,3] [,4] [,5] [,6]
[1,]  195  186  173  165   76    0
[2,]  200  200  200  200  188    8
[3,]  200  200  200  200  200  105
```

```
$dq
[1] 200
```

Pode-se selecionar os *quadrats* aleatoriamente:

```
> quadrat.count(gItajobi, 20, 10, ran = T)
```

```
$y
[1] 0 1 1 1 1 0 0 2 0 3 5 0 6 0 2 0 1 5
```

```
$n
[1] 200 200 200 200 200 200 200 200 200 200 200 200 200 200 200 200 200
[18] 200
```

```
$dq
[1] 200
```

Pode-se selecionar os *quadrats* aleatoriamente e sem desconsiderar os *quadrats* incompletos:

```
> quadrat.count(gItajobi, 20, 10, ran = T, comp = F)
```

```
$y
```

```
[1] 1 1 0 2 0 2 0 0 1 1 3 0 3 0 0 0 6 0
```

```
$n
```

```
[1] 200 200 195 158 133 97 170 139 200 200 200 200 200 3 56 150 200  
[18] 140
```

```
$dq
```

```
[1] 200
```

2 Índice de dispersão binomial

O índice de dispersão binomial é baseado em dados de *quadrat counts*. É a razão da variância observada pela variância teórica. Todos os *quadrats* analisados devem ter o mesmo número de plantas. Quando as dimensões da matriz de dados não é múltipla das dimensões do *quadrat*, os dados “excedentes” são desconsiderados da análise. Se faltar alguma planta para completar um *quadrat* dentro do talhão, este também é desconsiderado da análise.

Argumentos da função:

```
> args(dispatch.quadrats)
```

```
function (data, dx, dy = dx, counts.return = FALSE, by.evaluations = FALSE,  
  death = 1, healt = 0, model = c("binomial", "Poisson", "beta-binomial"),  
  alpha = 0.05, random = FALSE, N = NULL, p.quadrats = 1, p.quadrats.random = FALSE,  
  complet = TRUE, evaluation = "all", digits = 5, verbose = FALSE,  
  bb.args = list(ini.p = NULL, ini.theta = NULL, usage = c("fitdistr",  
    "mle")), ...)
```

```
NULL
```

Fazendo a análise:

```
> disp.quadrats(gItajobi, dx = 7)
```

```
$`7x7`
```

	n	N	nN	p	obs.var	theor.var	index	p.value	pattern
Av1	49	46	2254	0.0071	0.00036	0.00014	2.47	0	Agregate

```
> disp.quadrats(gItajobi, dx = 10, dy = 10)
```

```
$`10x10`
```

	n	N	nN	p	obs.var	theor.var	index	p.value	pattern
Av1	100	23	2300	0.00696	0.00025	7e-05	3.61	0	Agregate

```
> disp.quadrats(gItajobi, dx = 10, dy = 10, random = TRUE)
```

```
$`10x10`
```

	n	N	nN	p	obs.var	theor.var	index	p.value	pattern
Av1	100	36	3600	0.00861	0.00021	9e-05	2.42	1e-05	Agregate

```
> disp.quadrats(gItajobi, dx = c(2, 2, 5), dy = c(2, 5, 10))
```

```
$`2x2`
```

	n	N	nN	p	obs.var	theor.var	index	p.value	pattern
Av1	4	738	2952	0.00644	0.00225	0.0016	1.41	0	Agregate

```
$`2x5`
```

	n	N	nN	p	obs.var	theor.var	index	p.value	pattern
Av1	10	286	2860	0.00664	0.00118	0.00066	1.79	0	Agregate

```
$`5x10`
```

	n	N	nN	p	obs.var	theor.var	index	p.value	pattern
Av1	50	50	2500	0.0068	5e-04	0.00014	3.7	0	Agregate

Havendo mais de uma avaliação feita ao longo do tempo no mesmo talhão, podemos optar a forma da saída: por avaliação ou por tamanhos de *quadrats*, usando o argumento *by.evaluations*.

Carregando um conjunto de dados validados de MSC:

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

Fazendo a análise das três primeiras avaliações:

```
> disp.quadrats(v303.geo, dx = 2:5, dy = 2 * (2:5), death = 1:3,  
+   evaluation = 1:3)
```

```
$`2x4`
```

	n	N	nN	p	obs.var	theor.var	index	p.value	pattern
Av1	8	108	864	0.0255	0.00489	0.00310	1.58	0.00013	Agregate
Av2	8	108	864	0.0301	0.00522	0.00365	1.43	0.00233	Agregate
Av3	8	108	864	0.0764	0.01193	0.00882	1.35	0.00893	Agregate

```
$`3x6`
```

	n	N	nN	p	obs.var	theor.var	index	p.value	pattern
Av1	18	38	684	0.0263	0.00179	0.00142	1.26	0.13484	Random
Av2	18	38	684	0.0322	0.00227	0.00173	1.32	0.09511	Random
Av3	18	38	684	0.0833	0.00797	0.00424	1.88	0.00097	Agregate

```
$`4x8`
```

	n	N	nN	p	obs.var	theor.var	index	p.value	pattern
Av1	32	20	640	0.0281	0.00194	0.00085	2.27	0.00121	Agregate
Av2	32	20	640	0.0344	0.00174	0.00104	1.67	0.03273	Agregate
Av3	32	20	640	0.0813	0.00539	0.00233	2.31	0.00098	Agregate

```
$`5x10`
```

	n	N	nN	p	obs.var	theor.var	index	p.value	pattern
Av1	50	9	450	0.0289	0.00051	0.00056	0.911	0.506	Random
Av2	50	9	450	0.0333	0.00070	0.00064	1.086	0.369	Random
Av3	50	9	450	0.0822	0.00234	0.00151	1.553	0.133	Random

```
> disp.quadrats(v303.geo, dx = 2, dy = 5, by.evaluations = FALSE,  
+   death = 1:3)
```

```

$`2x5`
      n  N  nN      p obs.var theor.var index p.value pattern
Av1  10  80  800  0.0262 0.00424   0.00256  1.66 0.00021 Agregate
Av2  10  80  800  0.0300 0.00441   0.00291  1.51 0.00219 Agregate
Av3  10  80  800  0.0737 0.01158   0.00683  1.70 0.00011 Agregate
Av4  10  80  800  0.0925 0.01640   0.00839  1.95 0.00000 Agregate
Av5  10  80  800  0.1025 0.01620   0.00920  1.76 0.00004 Agregate
Av6  10  80  800  0.1588 0.02423   0.01335  1.81 0.00001 Agregate
Av7  10  80  800  0.2575 0.03235   0.01912  1.69 0.00012 Agregate
Av8  10  80  800  0.2600 0.03129   0.01924  1.63 0.00037 Agregate
Av9  10  80  800  0.3187 0.03445   0.02171  1.59 0.00070 Agregate
Av10 10  80  800  0.3362 0.03702   0.02232  1.66 0.00021 Agregate
Av11 10  80  800  0.4925 0.04298   0.02499  1.72 0.00007 Agregate
Av12 10  80  800  0.5550 0.04554   0.02470  1.84 0.00001 Agregate
Av13 10  80  800  0.6188 0.05142   0.02359  2.18 0.00000 Agregate
Av14 10  80  800  0.6312 0.05028   0.02328  2.16 0.00000 Agregate
Av15 10  80  800  0.6650 0.05344   0.02228  2.40 0.00000 Agregate
Av16 10  80  800  0.6837 0.05429   0.02162  2.51 0.00000 Agregate
Av17 10  80  800  0.7712 0.03397   0.01764  1.93 0.00000 Agregate
Av18 10  80  800  0.8337 0.02556   0.01386  1.84 0.00001 Agregate
Av19 10  80  800  0.8387 0.02468   0.01352  1.82 0.00001 Agregate
Av20 10  80  800  0.8788 0.01916   0.01065  1.80 0.00002 Agregate
Av21 10  80  800  0.9337 0.01239   0.00619  2.00 0.00000 Agregate
Av22 10  80  800  0.9337 0.01239   0.00619  2.00 0.00000 Agregate
Av23 10  80  800  0.9337 0.01239   0.00619  2.00 0.00000 Agregate
Av24 10  80  800  0.9800 0.00289   0.00196  1.47 0.00402 Agregate
Av25 10  80  800  0.9800 0.00289   0.00196  1.47 0.00402 Agregate

```

3 Índice de dispersão de Poisson

Para o índice de dispersão de Poisson, a função considera os dados de todos os *quadrats* onde o número de plantas considerado é maior que um quarto do número de plantas definido pelo tamanho nominal do *quadrat* e não considera os dados de *quadrats* com menor número de plantas.

```
> disp.quadrats(gItajobi, dx = 10, model = "Pois")
```

```

$`10x10`
      n  np      p obs.var theor.var index p.value pattern
Av1  36 2896 0.00656   1.76    0.643  2.73      0 Agregate

```

```
> disp.quadrats(gItajobi, dx = 10, model = "Pois", random = TRUE)
```

```

$`10x10`
      n  np      p obs.var theor.var index p.value pattern
Av1  36 3600 0.00639   1.89    0.639  2.97      0 Agregate

```

4 Modelo beta-binomial

Assumindo o modelo binomial ou de Poisson, assume-se que a probabilidade de incidência no talhão é constante em todo o talhão, porém no caso de haver agregação essa suposição não é razoável e o índice de dispersão é maior que um. Supondo que a probabilidade de incidência é variável, pode-se modelar os dados de *quadrat counts* por uma distribuição beta-binomial.

A distribuição beta-binomial é uma composição da distribuição binomial com a densidade beta, (Skellam 1948). A distribuição beta-binomial tem dois parâmetros: π , que representa a incidência média no talhão, e θ que mede a variação da incidência. O parâmetro θ , no contexto dos dados de incidência de doenças, pode ser interpretado como um índice de agregação espacial da doença. Se $\theta > 0$, o padrão da doença é agregado (Griffiths 1979).

As estimativas de máxima verossimilhança para a incidência média p e o parâmetro de agregação θ do modelo beta-binomial não tem expressões fechadas. Para obter essas estimativas, utiliza-se um procedimento iterativo de minimização (maximização) numérica, que pode não convergir.

```
> disp.quadrats(v303.geo, dx = 2, dy = 5, death = 1:3, model = "beta-binomial")
```

```
$`2x5`
```

	N	n	nN	prob	theta	p.value	pattern
Av1	100	9.45	945	0.0238	0.0986	0.01229	Agregate
Av2	100	9.45	945	0.0283	0.0692	0.04202	Agregate
Av3	100	9.45	945	0.0778	0.0695	0.01948	Agregate
Av4	100	9.45	945	0.0948	0.0993	0.00174	Agregate
Av5	100	9.45	945	0.1041	0.0781	0.00829	Agregate
Av6	100	9.45	945	0.1570	0.0984	0.00147	Agregate
Av7	100	9.45	945	0.2520	0.0829	0.00392	Agregate
Av8	100	9.45	945	0.2568	0.0700	0.01075	Agregate
Av9	100	9.45	945	0.3175	0.0617	0.01882	Agregate
Av10	100	9.45	945	0.3357	0.0722	0.00781	Agregate
Av11	100	9.45	945	0.4925	0.0936	0.00124	Agregate
Av12	100	9.45	945	0.5522	0.1153	0.00016	Agregate
Av13	100	9.45	945	0.6086	0.1602	0.00000	Agregate
Av14	100	9.45	945	0.6210	0.1633	0.00000	Agregate
Av15	100	9.45	945	0.6530	0.1822	0.00000	Agregate
Av16	100	9.45	945	0.6714	0.1892	0.00000	Agregate
Av17	100	9.45	945	0.7616	0.1175	0.00013	Agregate
Av18	100	9.45	945	0.8219	0.1190	0.00016	Agregate
Av19	100	9.45	945	0.8262	0.1202	0.00015	Agregate
Av20	100	9.45	945	0.8730	0.1045	0.00097	Agregate
Av21	100	9.45	945	0.9272	0.1406	0.00018	Agregate
Av22	100	9.45	945	0.9272	0.1406	0.00018	Agregate
Av23	100	9.45	945	0.9272	0.1406	0.00018	Agregate
Av24	100	9.45	945	0.9711	0.1058	0.00730	Agregate
Av25	100	9.45	945	0.9711	0.1058	0.00730	Agregate

4.1 Detalhes da estimação

O procedimento de estimação dos parâmetro da distribuição beta-binomial, esta implementado em detalhes na função `betabinom.citrus()`.

```
> args(betabinom.citrus)

function (data, dx, dy = dx, check.args = NULL, ini.prob = NULL,
  ini.theta = NULL, usage = c("fitdistr", "mle"), ...)
NULL
```

Nos argumentos dessa função precisamos entrar com os dados no argumento `data`, as dimensões dos *quadrats* nos argumentos `dx` e `dy` e as informações dos códigos identificadores do *status* das plantas em forma de lista nomeada.

Duas funções podem ser utilizadas na minimização numérica: a `fitdistr()` do pacote **MASS**, (Venables & Ripley 2002) e a função `mle()` do pacote **stats4**. A primeira, retorna simplesmente as estimativas dos parâmetros e dos erros-padrão obtidos do hessiano numérico. A segunda, retorna um resultado que possibilita explorar as verossimilhanças perfilhadas. A escolha dessa função é informada no argumento `usage`.

Nos dados de PPC podemos utilizar a função `fitdistr`, fazendo:

```
> (bb.lita <- betabinom.citrus(gItajobi, dx = 10))

valid evaluations: 1
  prob      theta
0.00638  0.01934
(0.00236) (0.01330)
```

4.2 Verossimilhanças perfilhadas

Utilizando `usage='mle'`, retorna-se um objeto da classe `mle`. Com esse resultado, podemos estudar as verossimilhanças perfilhadas, para aproximar intervalos de confiança para as estimativas e visualizar.

```
> bb.ita <- betabinom.citrus(gItajobi, dx = 10, usage = "mle")
```

```
valid evaluations: 1
```

Obtendo o sumário:

```
> summary(bb.ita)
```

```
Maximum likelihood estimation
```

```
Call:
```

```
mle(minuslogl = function(prob, theta) -sum(dbetabinom(x = y,
  size = size, prob = prob, theta = theta, log = TRUE))), start = list(prob = ini.pr
  theta = ini.theta))
```

```
Coefficients:
```

	Estimate	Std. Error
prob	0.00638	0.00236
theta	0.01934	0.01330

```
-2 log L: 62.6
```

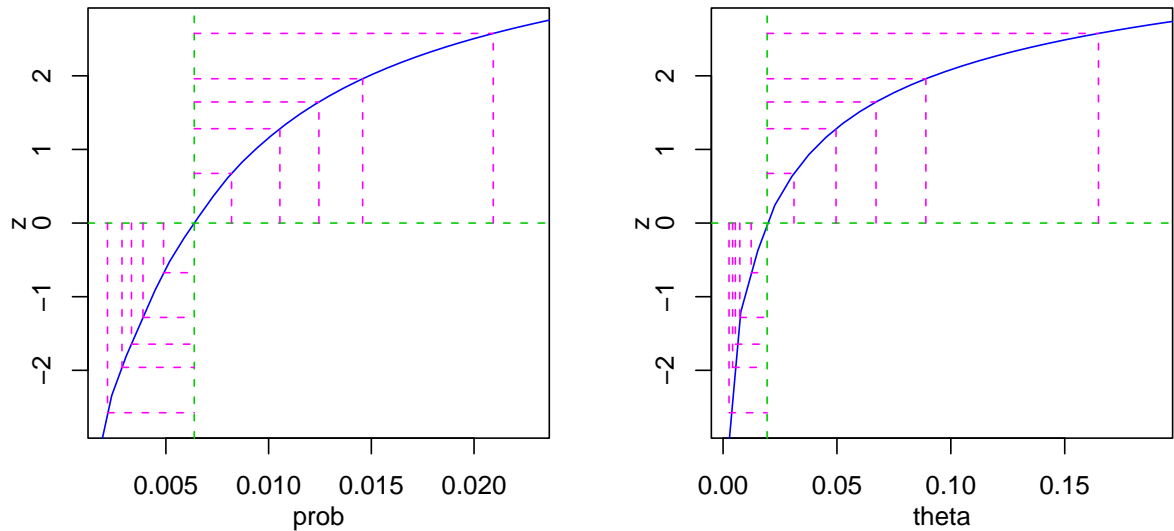



Figura 2: Visualização da verossimilhança perfilhada para os parâmetros da distribuição beta-binomial aplicada aos dados de Itajobi

Para obter o intervalo de confiança para as estimativas a partir das verossimilhanças perfilhadas, faz-se:

```
> prof <- profile(bb.ita)
> confint(prof)
```

```
      2.5 % 97.5 %
prob  0.00288 0.0146
theta 0.00463 0.0890
```

As verossimilhanças perfilhadas podem ser plotadas, como na Figura 2.

```
> par(mfrow = c(1, 2), mar = c(3, 3, 3, 0.5), mgp = c(2, 1,
+ 0))
> plot(prof, absVal = FALSE)
```

5 Lei de Taylor

A estimação dos parâmetros da Lei de Taylor é feita a partir do cálculo da variância observada e da variância esperada em observações feitas em vários talhões ou em várias ocasiões no mesmo talhão.

Usando a função `Taylor.citrus` podemos estimar os parâmetros da Lei de Taylor para usando as avaliações feitas em um talhão.

```
> Tay <- Taylor.citrus(v303.geo, dx = 5, random = FALSE, death = 1:3)
> Tay
```

```
      a      b
2.16 1.20
```

```
> summary(Tay)
```

```
Summary of disease incidence:
```

```
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.0257 0.2600 0.6110 0.5330 0.8300 0.9800
```

```
Estimates and confidence intervals of Taylor Law:
```

```
 2.5 % estimate 97.5 %
```

```
a 1.54      2.16  2.77
```

```
b 1.09      1.20  1.31
```

```
Thue an evidences of an agregatedpattern.
```

```
> TayR <- Taylor.citrus(v303.geo, dx = 5, random = TRUE, death = 1:3)
```

```
> TayR
```

```
      a      b
2.18 1.22
```

```
> summary(TayR)
```

```
Summary of disease incidence:
```

```
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.0294 0.2140 0.5600 0.5230 0.8080 0.9820
```

```
Estimates and confidence intervals of Taylor Law:
```

```
 2.5 % estimate 97.5 %
```

```
a 1.46      2.18  2.91
```

```
b 1.08      1.22  1.35
```

```
Thue an evidences of an agregatedpattern.
```

Podemos fazer o gráfico para as duas análises usando a função `plot()`. Esta função foi programada para mostrar a incidência no eixo horizontal e o índice de dispersão na vertical, Figura 3.

```
> par(mfrow = c(1, 2), mar = c(3, 3, 3, 0.5), mgp = c(2, 1,
+      0))
> plot(Tay)
> plot(TayR)
```

6 Número ótimo de *quadrats* aleatórios

Na análise por *quadrats* existe arbitrariedade na definição dos *qs*, quanto à tamanho, forma e o modo como são selecionados. Os *quadrats* podem ser fixos no talhão, definindo-os a partir de um extremo do talhão, ou aleatórios, definindo-os em qualquer posição no talhão. Na seleção aleatória dos *quadrats*, podemos selecionar uma amostra aleatória representativa. Então precisamos de um método que nos permita escolher o número ótimo de *quadrats* selecionados aleatoriamente.

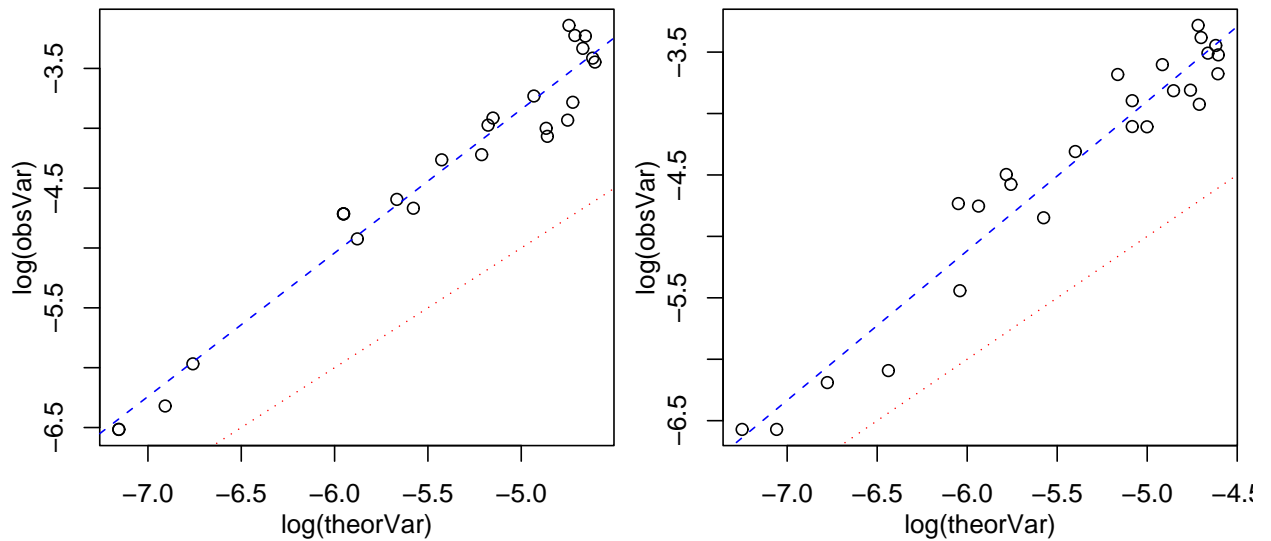


Figura 3: Logarítmo da variância observada versus logarítmo da variância teórica.

É razoável assumir que a análise feita a partir de *quadrats* selecionados aleatoriamente, será confiável em proporção ao número *quadrats*. Também é razoável assumir que o ganho em precisão na análise decai a medida em que aumenta-se o número de *quadrats*. Dessa forma, várias análises feitas com poucos *quadrats* selecionados aleatoriamente terá resultados mais variáveis que a análises feitas com um número maior de *quadrats*.

A metodologia implementada consiste em:

- realizar n análises com N *quadrats* selecionados aleatoriamente,
- realizar outras n análises com $N + 1$ *quadrats*
- repetir o passo anterior até um critério ser atingido

É difícil escolher um critério que seja eficaz para detectar que a estabilidade foi atingida para determinado N . As técnicas de controle de qualidade podem ser úteis para avaliar a estabilidade de uma medida ao longo de várias amostras. Para ilustrar, calculou-se o índice de dispersão binomial tomando-se, inicialmente N *quadrats* aleatórios com N entre 3 e 100, repetindo-se $n = 3$ vezes a análise para cada N .

```
> set.seed(153)
> res1 <- sim.N.quadrats(gItajobi, dx = 10, ini.N = 3, k = 100)
> qcc1 <- qcc(res1, type = "xbar")

> qcc2 <- qcc(res1, type = "R")
```

Na Figura 4 vê-se o gráfico de controle da média do índice de dispersão entre cada uma das 3 análises feitas para cada N . Os valores do eixo horizontal do gráfico representam o número de *quadrats* selecionados. Observa-se que a média do índice de dispersão binomial é mais variável quando se toma poucos *quadrats* e torna-se mais estável a medida que a análise é feita com maior número de *quadrats*.

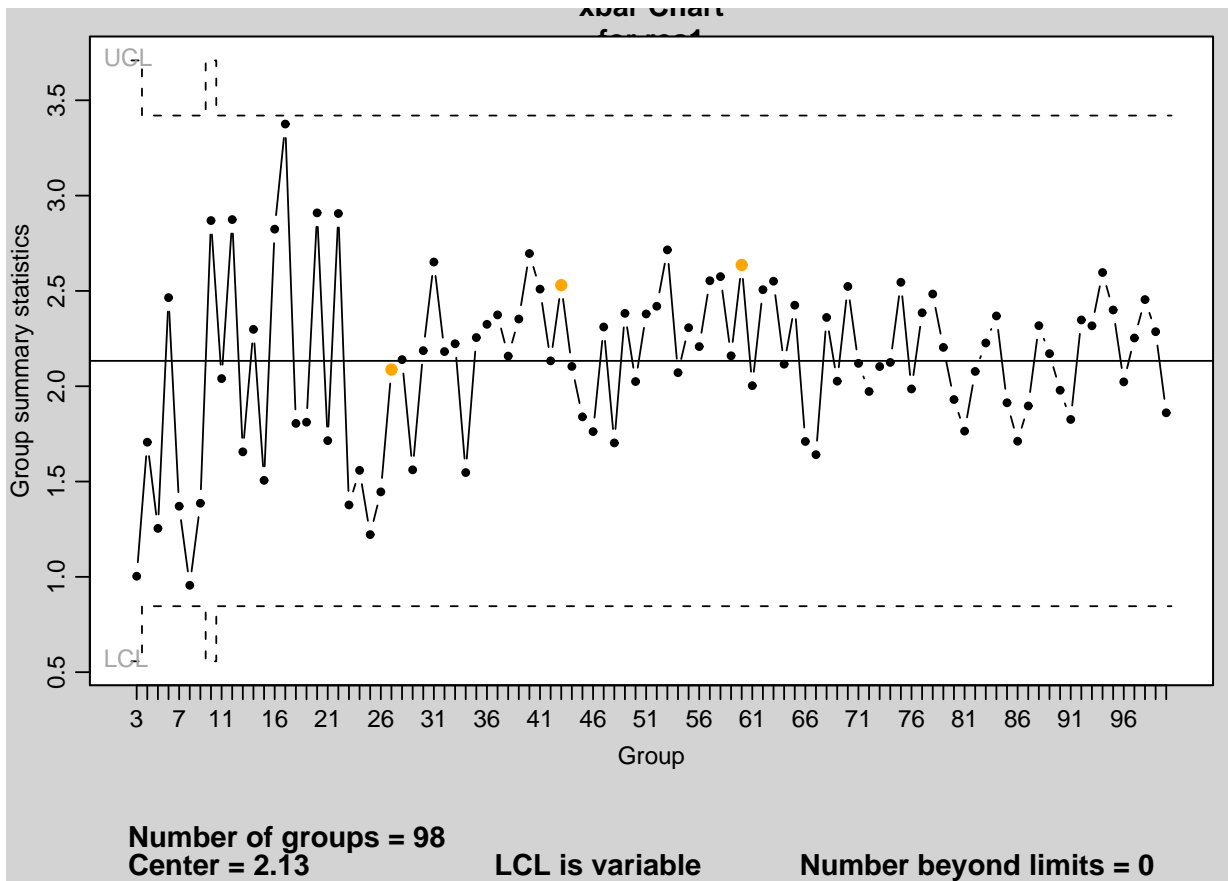


Figura 4: Carta de controle para média do índice de dispersão das 3 análises para análises de 3 a 100 *quadrats*

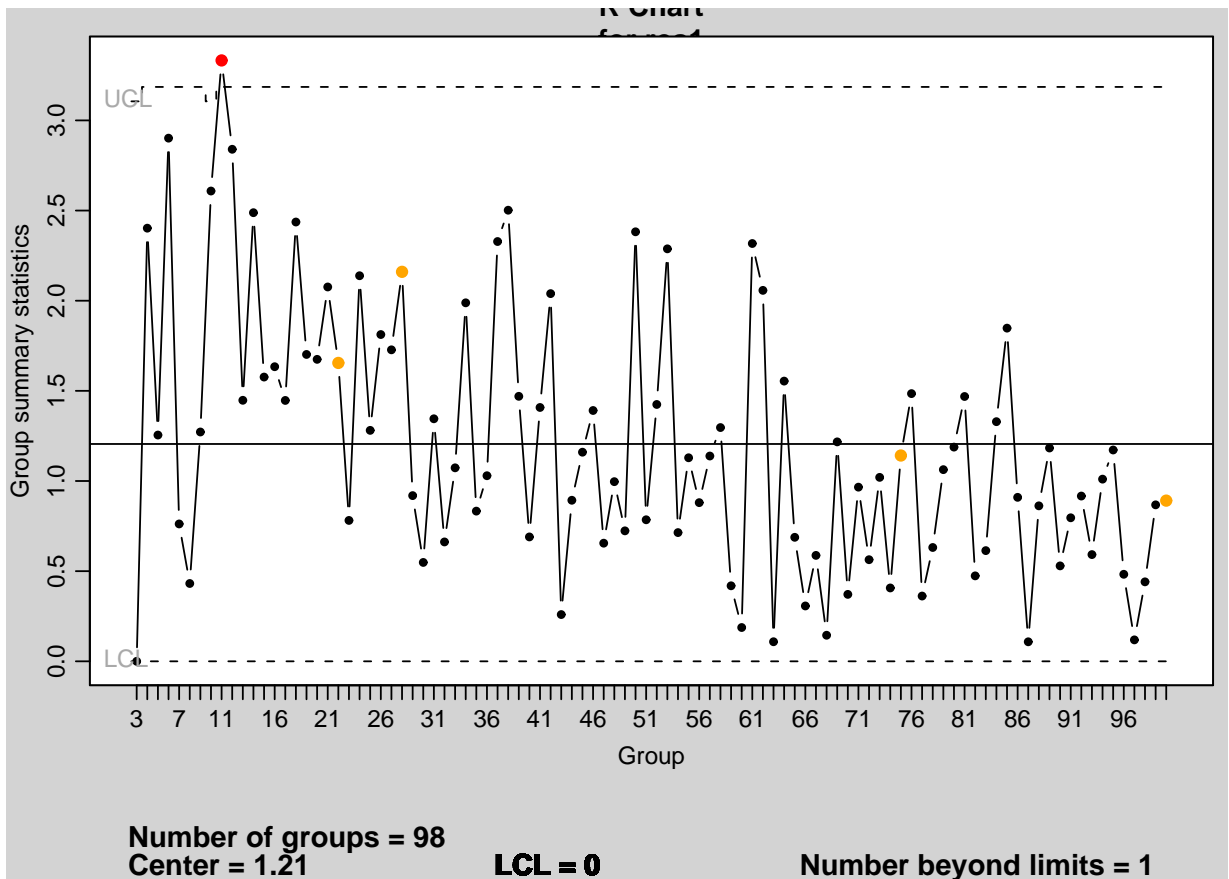


Figura 5: Carta de controle para a amplitude do índice de dispersão das 3 análises para cada análises de 3 a 100 *quadrats*

Na Figura 5 vê-se o gráfico de controle da amplitude do índice de dispersão entre cada uma das 3 análises feitas para cada N . Observa-se que a amplitude entre o índice de dispersão das três análises diminui a medida que o número de *quadrats* aumenta.

Nos gráficos de controle os pontos coloridos em vermelho representam pontos fora dos limites de controle e os pontos laranja aparecem quando uma seqüência de 5 valores ocorre abaixo ou acima da média. Esses pontos são detectados pelas funções `beyond.limits()` e `violating.runs()` do pacote **qcc**, respectivamente. No caso do gráfico da amplitude, quando ocorre um ponto laranja, muito provavelmente este estará abaixo da média, indicando que houve uma estabilização da amplitude abaixo da amplitude média.

A função `sim.N.quadrats()` realiza a análise por *quadrats* aleatórios iniciando com valor para N informado e repetindo n vezes. Quando se atinge certo número k de valores de N , é feito um gráfico de controle para detectar se houve *violating runs*. Se não houve, continua-se fazendo a análise para números crescentes de *quadrats* e refaz-se o gráfico de controle para os últimos k valores de N considerados.

```
> args(sim.N.quadrats)

function (obj, dx, dy = dx, complet = TRUE, ini.N = 5, n = 3,
  k = 10, stat = "index", death = 1, healt = 0, model = c("binomial",
  "Poisson", "beta-binomial"), type = "R", alpha = 0.05,
  evaluation = 1, N.max = 100, verbose = FALSE)
NULL
```

```
> (res2 <- sim.N.quadrats(gItajobi, dx = 10))
```

```
      am1  am2  am3
5  0.753 0.753 1.512
6  0.403 3.015 6.061
7  1.090 1.451 3.704
8  1.482 3.708 4.329
9  4.879 1.548 3.060
10 1.064 1.856 4.173
11 3.791 2.195 2.715
12 0.914 1.774 1.077
13 0.835 0.821 2.270
14 1.469 1.277 2.206
15 1.767 4.264 4.027
16 0.946 4.283 3.135
17 3.036 0.911 1.638
18 3.742 1.630 0.884
19 0.927 1.352 4.297
```

Na Figura 6 visualiza-se um gráfico de controle para a amplitude dos valores do índice de dispersão obtidos da análise.

```
> qcc3 <- qcc(res2, type = "R")
```

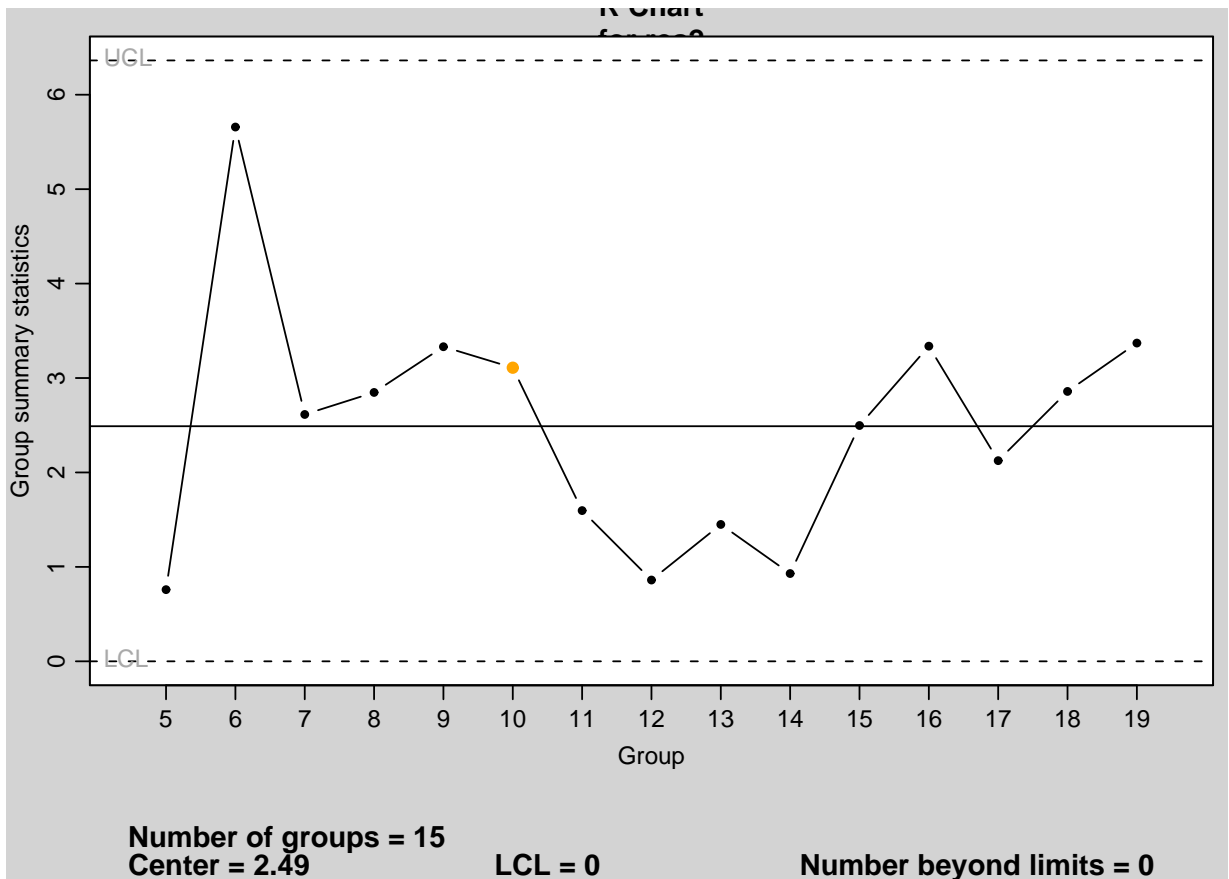


Figura 6: Carta de controle para a amplitude do índice de dispersão das 3 análises para cada valor de N a partir de 5 e interrompido pelo critério adotado

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

Griffiths, D. A. (1979). Maximum likelihood estimation for the beta-binomial distribution and an application to household distribution of total number of a disease, *Biometrics* .

Madden, L. V. & Hughes, G. (1995). Plant disease incidence: Distributions, heterogeneity, and temporal analysis, *Phytopathology* .

Skellam, J. G. A. (1948). A probability distribution derived from the binomial distribution by regarding the probability of success as variable between the sets of trials, *Journal of Royal Statistical Society, Series B* .

Venables, W. N. & Ripley, B. D. (2002). *Modern Applied Statistics with S*, Fourth edition.