REDUCED SECOND MOMENT MEASURE (*K*-FUNCTION)

- It measures the departure from CSR at a wide range of spatial scales from short distances (large scale) to long distances (small scale).
- Assumes first order stationarity over the whole study area, i.e., $\lambda(\mathbf{s}) = \text{constant } \forall \mathbf{s}$.
- *R* is the area of the study region R, i.e., $R = |\mathrm{R}|$
- The ***expected number of events*** in the study region R is $E[Y(\mathrm{R})] = \lambda \cdot R = n$
- Conceptional definition of a *K*-function:
  $\lambda \cdot K(h) = E\left[\#\left(\text{events within distance } h \text{ of an } \textit{arbitrary} \text{ event under CSR}\right)\right]$
- Implementation of the *K*-function with an indicator function $I_h(d_{ij})$ for a given threshold distance h:

  $$I_h(d_{ij}) = \begin{cases} 1 \text{ for } d_{ij} < h \\ 0 \text{ otherwise} \end{cases}$$

  The expected number of events within the study region R is $E[Y(\mathbb{R})] = \lambda \cdot R = n$.

$$\lambda \cdot \hat{K}(h) = \frac{\sum_{i=1}^{n} \sum_{j \neq i, j=1}^{n} I_h(d_{ij})}{\underbrace{\lambda \cdot R}_{=n}}$$

$$\Leftrightarrow \hat{K}(h) = \frac{\sum_{i=1}^{n} \sum_{j \neq i, j=1}^{n} I_h(d_{ij})}{\lambda \cdot n} = \frac{R \cdot \sum_{i=1}^{n} \sum_{j=1, j \neq i}^{n} I_h(d_{ij})}{n^2}$$

- Edge corrections are performed as for the $F(x)$ and $G(w)$ functions.

THE *K*-FUNCTION FOR ALL SCALES

- The expected number of events under a CSR process is proportional to the area of a disk with radius *h*

$$\underbrace{\pi \cdot h^2}_{=\text{Area circle of radius } h} \cdot \hat{\lambda} \text{ with } \hat{\lambda} = n/R$$

(the intensity cancels out in the equations below):

  - If $\hat{K}(h) - \pi \cdot h^2 = 0$ for a given range of *h*, then is the point pattern process random at this range
  - If $\hat{K}(h) - \pi \cdot h^2 > 0$ for a given range of *h*, then is the point pattern process clustered at this range
  - If $\hat{K}(h) - \pi \cdot h^2 < 0$ for a given range of *h*, then is the point pattern process regular at this range
  - Alternative specification of the statistic as *L*-function becomes $\hat{L}(h) = \sqrt{\dfrac{\hat{K}(h)}{\pi}} - h$.

    The interpretation of $\hat{L}(h)$ remains the same.

- The function $\hat{K}(h) - \pi \cdot h^2$ can be plotted against *h* to give some indication of clustering or dispersion at different scales. Zero indicates CSR at a given distance *h*.

- See **OH Bailey Fig 3.8**. There is an indication of clustering at small to medium scales *h* (Note: CSR is indicated by 1) :

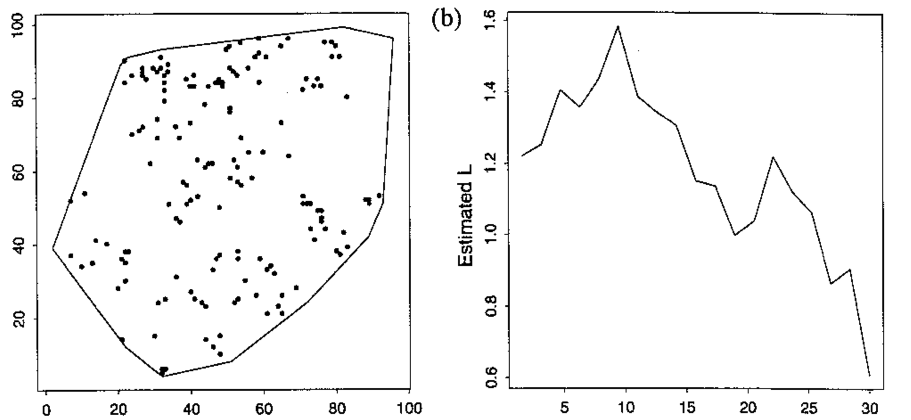


Fig. 3.8 (a) Juvenile offenders in Cardiff and (b) associated $\hat{L}$ function

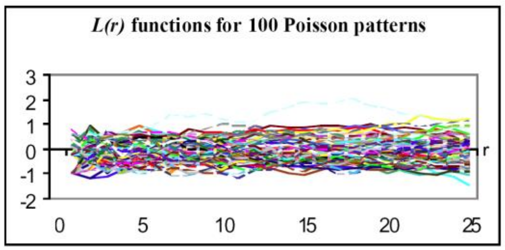
- Properties of *K*-functions:
  - Uses all inter-event information (all distances in the distance matrix)
  - Gives indication for non-randomness over the whole range of scales *h* and not just the short nearest neighbor distances.
  - Besides simple spatial randomness *K*-functions can also be used for other point processes (e.g., parent-children process).

- One can also evaluate *local* K-functions associated with the reference location $\mathbf{s}_i$

$$K_i(h) = \frac{1}{\lambda} \cdot E\left[\sum_{j=1, j\neq i}^{n} I_h(d_{ij})\right]$$

K-FUNCTION TEST FOR CSR

- Recall under CSR we have $K(h) = \pi \cdot h^2$ but there is no formal way to test for a significant departure of $\hat{K}(h)$
- One way to develop confidence intervals is to perform m simulations of a CSR process with the same number of events in the study area and calculate the *K*-functions for each simulated pattern.



L(r) estimates obtained for 100 realizations of a Poisson process of 100 points in a square study area of 100 x 100 m.

- Determine from the set of simulated *K*-functions the upper 97.5 $\mathrm{U}(h)$ and lower 2.5 $\mathrm{L}(h)$ percentiles for any value *h.*
- If the observed $\hat{K}$ function is within the 95% confidence interval for a given scale (i.e., distance range *h*) then within that range we cannot reject the assumption of CSR.
- This approach is also applicable for other than CSR spatial point processes by simulating these processes and comparing the empirically observed *K*-function against the ensemble of the simulated *K*-functions for those

processes,

$\Rightarrow$ If the empirically observed *K*-function is within the envelop of the simulated *K*-functions then the observed pattern is not different from the assumed spatial process.

AN EXAMPLE: INDISTINGUISHABLE BUT DIFFERENT SPATIAL POINT PROCESSES

- The **Cox process** is a *heterogeneous* Poisson process (first order stationarity is violated) where the intensity $\lambda(\mathbf{s}_i)$ varies randomly:
  - $\lambda(\mathbf{s}_i)$ is drawn from a probability distribution
  - Conditionally on $\lambda(\mathbf{s}_i)$ the number of events in A_i is simulated.
- **Example:** $\lambda(\mathbf{s}_i)$ follow a Gamma-distribution and the points within A_i are CSR, i.e., they follow a Poisson distribution.
- The resulting Cox point process leads to a *Negative Binomial* distribution $\Pr(X = x|\pi, n) = \binom{x+n-1}{n-1} \cdot (1-\pi)^x \cdot \pi^n$ of the grid cell counts X.
- The **Poisson clustering process** (Neyman-Scott process) is also a two stage doubly stochastic process:
  - A parent distribution of events is generated according to CSR.
  - The number of off-springs at a parent event is generated by an alternative distribution.
    [a] The number of off-springs for each parent is randomly determined.

[b[The off-springs are distributed randomly around their parent following a distance decay relationship with diminishing likelihood the further they are away from their parent.

- **Example:** The parent distribution of events follows a homogeneous Poisson process. The number of off-springs follows logarithmic distribution.
- The cell counts for this parent off-spring process again follow *Negative Binomial* distribution.
- <u>Consequence:</u> by observing a *Negative Binomial* distribution of cell counts we ***cannot*** distinguish whether [a] a heterogenous Cox process or [b] a two stage Poisson clustering process generated the underlying distribution of events.
- Show Negative Binomial event distribution in **Davis p 299-308.**

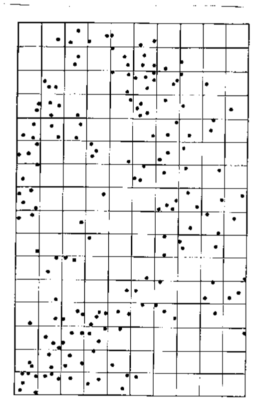


FIGURE 5.11 Locations of oil field discovery wells in part of the Eastern Shelf area of the Permian Basin, Fisher and Noland counties, Texas. Quadrats are approximately 10 square miles in size.

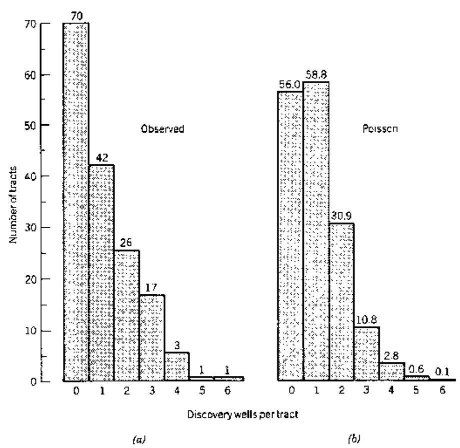


FIGURE 5.12 Histograms showing (*a*) observed numbers of discovery wells per tract in an area of the Permian Basin, and (*b*) the number expected if fields are distributed randomly according to a Poisson model.

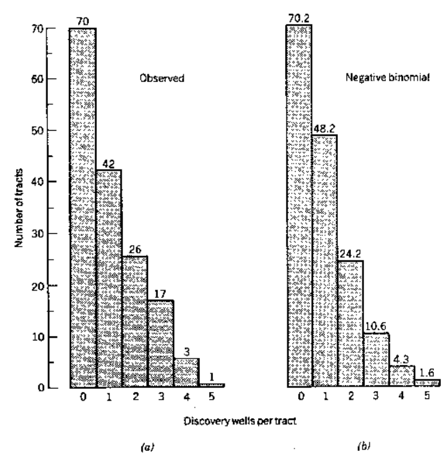


FIGURE 5.13 Histograms showing (*a*) observed numbers of discovery wells per tract in an area of the Permian Basin, and (*b*) the number expected

NON-COMPLETE SPATIAL RANDOM PROCESSES

- Generating random spatial patterns

`runifpoint`	generate n independent uniform random points
`rpoint`	generate n independent random points
`rmpoint`	generate n independent multitype random points
`rpoispp`	simulate the (in)homogeneous Poisson point process
`rmpoispp`	simulate the (in)homogeneous multitype Poisson point process
`runifdisc`	generate n independent uniform random points in disc
`rstrat`	stratified random sample of points
`rsyst`	systematic random sample of points
`rMaternI`	simulate the Matérn Model I inhibition process
`rMaternII`	simulate the Matérn Model II inhibition process
`rSSI`	simulate Simple Sequential Inhibition process
`rStrauss`	simulate Strauss process (perfect simulation)
`rHardcore`	simulate Hard Core process (perfect simulation)
`rDiggleGratton`	simulate Diggle-Gratton process (perfect simulation)
`rDGS`	simulate Diggle-Gates-Stibbard process (perfect simulation)
`rNeymanScott`	simulate a general Neyman-Scott process
`rPoissonCluster`	simulate a general Poisson cluster process
`rMatClust`	simulate the Matérn Cluster process
`rThomas`	simulate the Thomas process
`rGaussPoisson`	simulate the Gauss-Poisson cluster process
`rCauchy`	simulate Neyman-Scott Cauchy cluster process
`rVarGamma`	simulate Neyman-Scott Variance Gamma cluster process
`rcell`	simulate the Baddeley-Silverman cell process
`rmh`	simulate Gibbs point process using Metropolis-Hastings
`simulate.ppm`	simulate Gibbs point process using Metropolis-Hastings
`runifpointOnLines`	generate n random points along specified line segments
`rpoisppOnLines`	generate Poisson random points on specified line segments

Table 4.9. *Random point pattern generators in* `spatstat`.

- Interaction Process:
  - Matern process (**`Matern`**) is derived from a homogeneous parent Poisson process with intensity κ and each parent has $Pois(\mu)$ and the off-springs are uniform distributed in a disc with radius r. Subsequently the parents of off-springs are discarded.

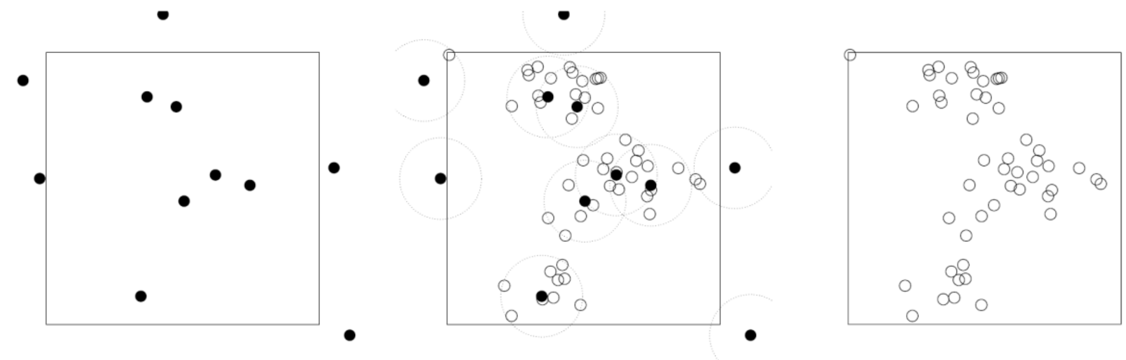


Figure 5.15. *Formation of Matérn cluster process.* Left: *parent points are generated according to a homogeneous Poisson process.* Middle: *for each parent, a Poisson random number of offspring (with mean μ offspring) are generated, and placed independently and uniformly in a circle of radius R around the parent.* Right: *the offspring constitute the Matérn cluster process.*

- Inhibition Process
  - The Matern Model I (**`MaternI`**) process is based on a homogenous Poisson process and then each point pair that lies closer than the distance r is deleted.

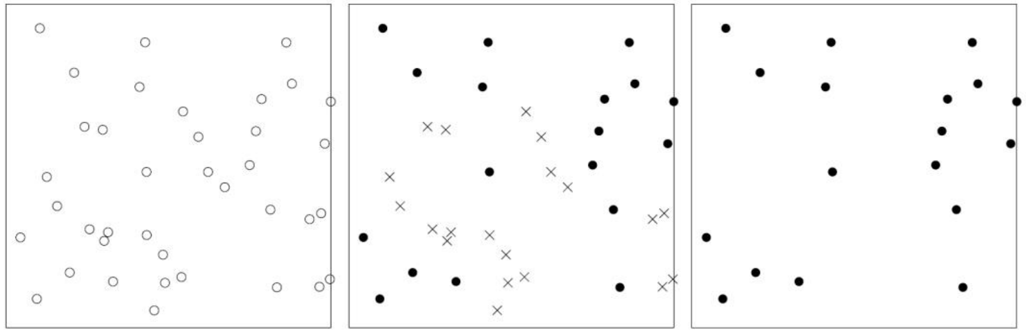


Figure 5.16. *Formation of Matérn's Model I.* Left: *points are generated by Poisson process.* Middle: *any point lying within distance R of another point is deleted* $(\times)$. Right: *resulting thinned process. Unit square window, interaction distance* $R = 0.1$.

- The Matern II (**`MaternII`**) process is similar to the Matern I process, however, depending the arrival time of the points only one point in the pair of close points is deleted.

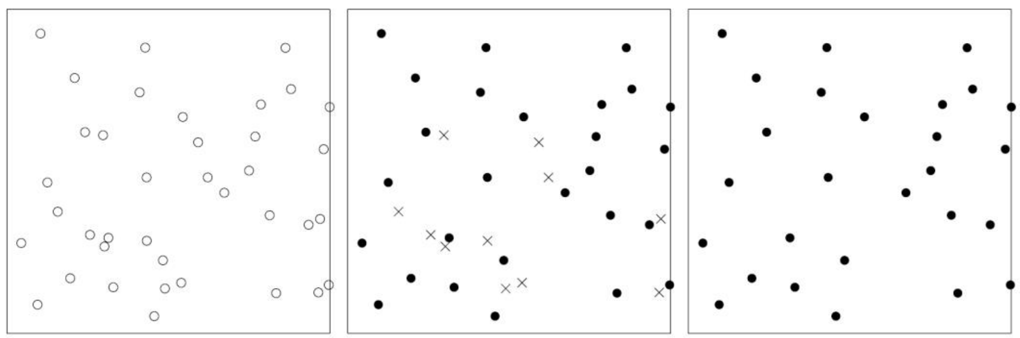


Figure 5.17. *Formation of Matérn's Model II.* Left: *points are generated by Poisson process and marked by independent arrival times.* Middle: *any point lying within distance R of an* earlier *point is deleted* $(\times)$. Right: *resulting thinned process. Unit square window, interaction distance* $R = 0.1$.

- The sequential inhibition process generates one Poisson point at a time and then discards it if it is closer than r to an already existing point.

INHOMOGENEOUS POINT PATTERN

- An inhomogeneous complete spatial random surface can be simulated by first establishing a probability surface, then generating a CSR proposal point and finally accepting the proposal at location $\mathbf{s}_i$ by the probability of the underlying surface at that location.

```
> lambda <- function(x,y) { 100 * (x^2+y) }
> X <- rpoispp(lambda, win=square(1))
```

The result is shown in Figure 5.14.

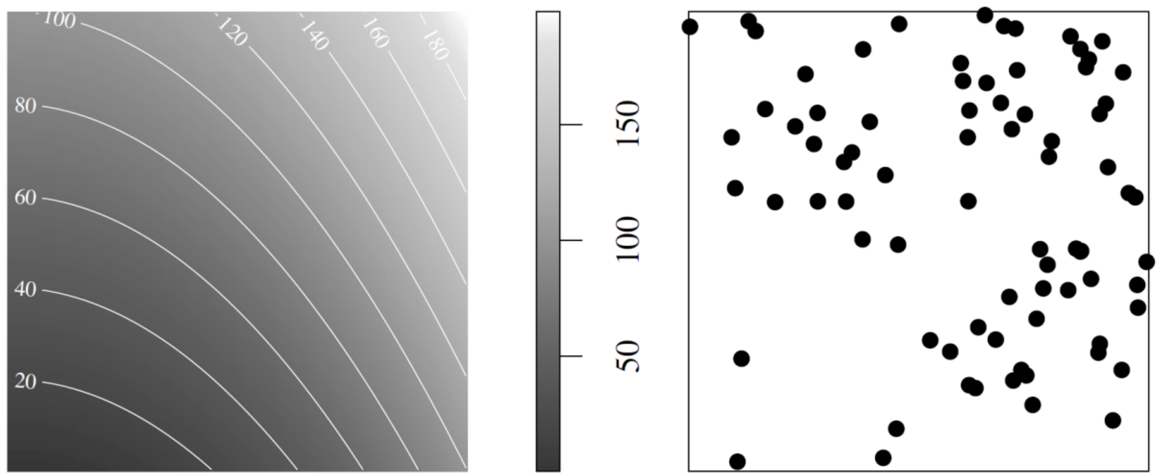


Figure 5.14. *Inhomogeneous Poisson process.* Left: *intensity function;* Right: *realisation of point process.*

- Run -script `SimulInhibClustAniso.R`. It also explores the impact of anisotropy.