Statistics 222, Spatial Statistics.

<u>Outline for the day:</u>

1. Simulation.

2. Purely spatial processes, Papangelou intensity and

   Georgii Zessin Nguyen.

3. Kernel smoothing.

4. F, G, J, K, and L functions.

5. Exercises.

1. Simulation.

One can simulate spatial-temporal point processes by *thinning*.
Lewis, P. and Shedler, G. (1979). Simulation of nonhomogeneous poisson processes by thinning. *Naval Research Logistics Quarterly*, 26:403–413, 1979.

Jesper Møller

Suppose λ has some upper bound, B. $\lambda(t,x,y) \leq B$ everywhere.
First, simulate a stationary Poisson process N with intensity B.
For i = 1,2, ... keep point τ_i with probability $\lambda(\tau_i)/B$. We saw this in day 4 for simulating inhomogeneous Poisson processes, but it works for other processes too.

Boundary issues can be important in simulation. For Gibbs processes, for instance, the simulation can be biased because of missing points outside the observation region. For Hawkes processes, the simulation will tend to be biased by having too few points at the beginning of the simulation. One can have burn-in, by simulating points outside the observation region or before time 0, or in some cases some fancy weighting schemes can be done to achieve *perfect* simulation without burn-in. See Møller, J. and Waagepetersen, R. (2003). *Statistical Inference and Simulation for Spatial Point Processes*. Chapman and Hall, Boca Raton.

2. Purely spatial processes, Papangelou intensity and the Georgii-Zessin Nguyen formula.

For point processes in R^2, there is no natural ordering as there is in time. One could just use the x-coordinate in place of time and define a conditional intensity, but most models for spatial processes would be very awkward to define this way.

Instead, a more natural and useful tool is the Papangelou intensity, $\lambda(x,y)$, which is the conditional rate of points around location (x,y), given information on everywhere else. Letting

$$l(\theta) = \sum \log(\lambda(\tau_i)) - \int \lambda(x,y) \, dx \, dy,$$

where $\lambda(x,y)$ is the Papangelou intensity,
$l(\theta)$ is called the *pseudo-loglikelihood*.

A key formula for space-time point processes is called the *martingale formula*:
for any predictable function $f(t,x,y)$,

$$E \int f(t,x,y) \, dN = E \int f(t,x,y) \, \lambda(t,x,y) \, d\mu.$$

$$= E \sum_i f(t_i, x_i, y_i) = E \int f(t,x,y) \, \lambda(t,x,y) \, dt \, dx \, dy$$

For spatial point processes the corresponding formula,

$$E \int f(x,y) \, dN = E \int f(x,y) \, \lambda(x,y) \, dx \, dy$$

is called the Georgii-Zessin-Nguyen formula.
When $f = 1$, this means $EN(B) = E \int \lambda \, d\mu$.

3. Kernel smoothing.
A simple way to start summarizing a spatial point process is by kernel smoothing.

Suppose your observation region is B.
Let k(x,y) be a spatial density function, called a kernel, and construct, for each location (x,y),
$\lambda^{\wedge}(x,y) = \int_B k((x',y') - (x,y))\, dN(x',y') / \rho(x,y),$
where $\rho(x,y) = \int_B k((x',y') - (x,y))\, dx'\, dy'$ is an edge correction term.

The resulting function $\lambda^{\wedge}(x,y)$ is a natural estimator of $\lambda(x,y)$ and, if N is a Poisson process, is an unbiased estimator of $\lambda(x,y)$.
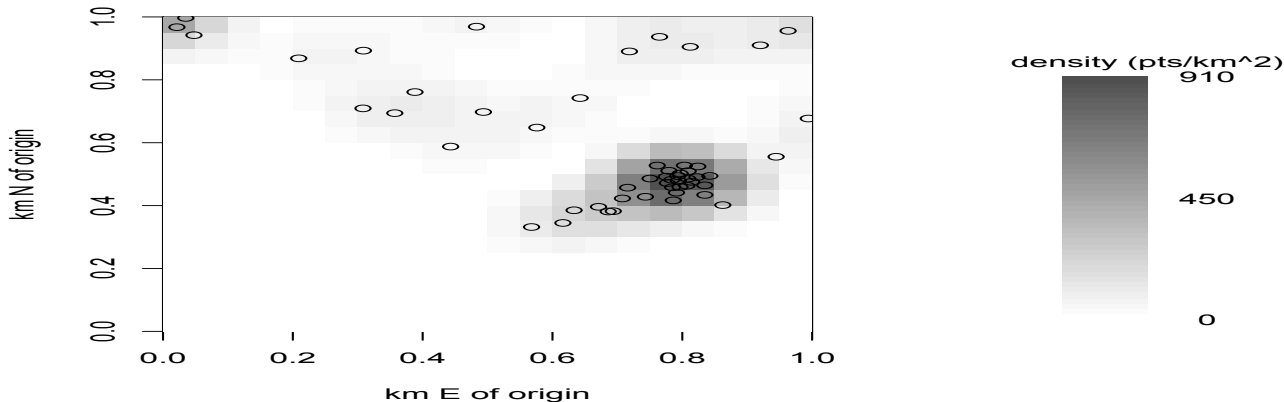
4. F, G, J, K, and L functions.

Let F(r) be the probability that the distance from a randomly chosen location to its nearest *point* of the process is ≤ r.

Let G(r) be the probability that the distance from a randomly chosen *point* to its nearest neighbor is ≤ r.

F is the empty space function and G is the nearest neighbor distribution function.

Matern (1971) showed that for a homogeneous Poisson process, $F(r) = G(r) = 1 - \exp(-\lambda \pi r^2)$.

Marie-Collette van Lieshout

Let $J(r) = (1-G(r)) / (1-F(r))$, for r such that $F(r)<1$.

$J > 1$ indicates inhibition, and $J < 1$ indicates clustering.

For a stationary Poisson process with rate μ, let

$K(r) = 1/\mu$ E(# of other points within distance *r* of a randomly chosen point).

K is the reduced 2nd moment measure or *Ripley*'s K-function (Ripley, 1976).

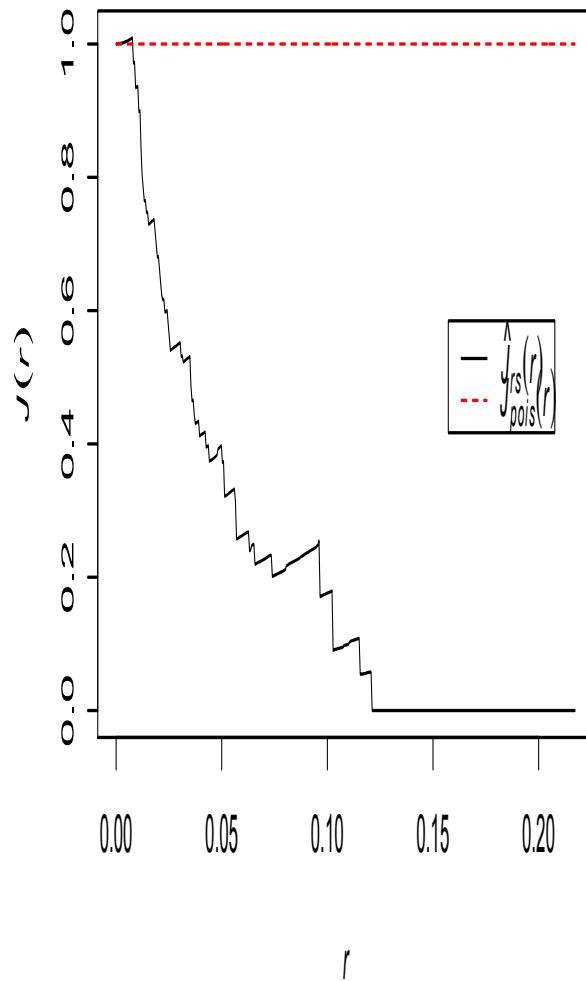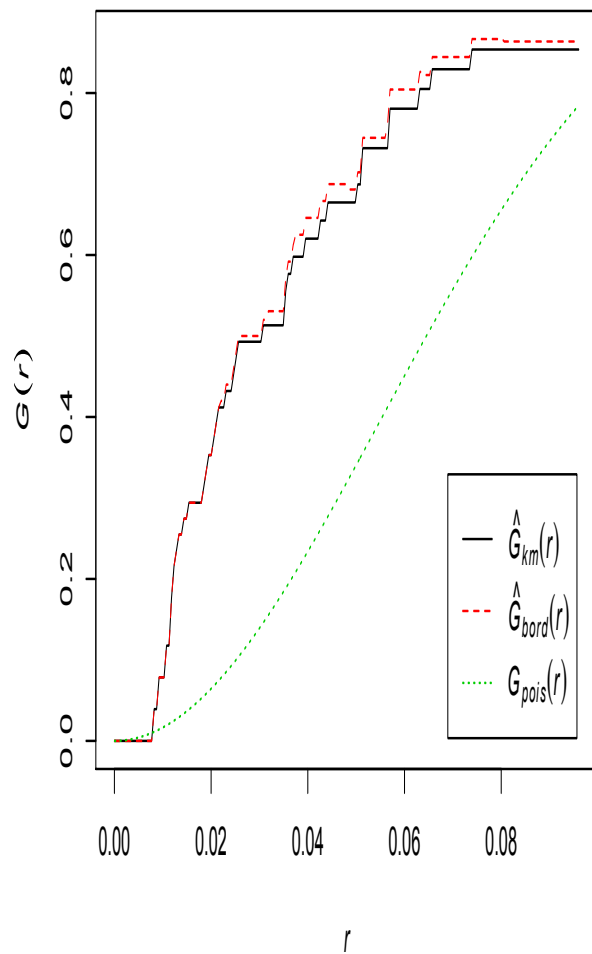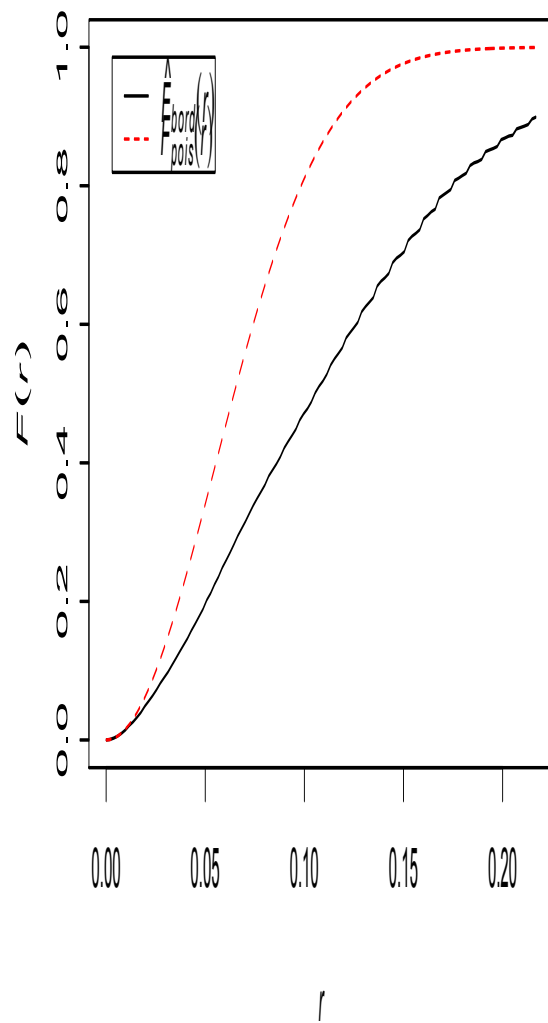van Lieshout, M.C. (2010). A *J*–function for inhomogeneous point processes. *Statistica Neerlandica*, 65(2), 183-201.

and references therein gives extensions to the inhomogeneous Poisson process and to marked point processes.

4. F, G, J, K, and L functions, continued.

4. F, G, J, K, and L functions, continued.

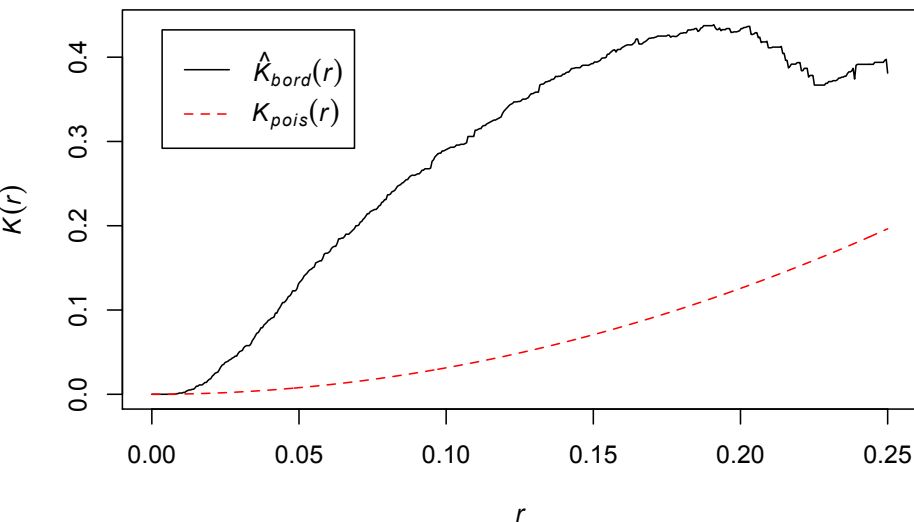K(r) = 1/μ E(# of other points within distance *r* of a randomly chosen point).

K is estimated in the obvious way using data, but various edge correction ideas are available.

For a stationary Poisson process, $K(r) = \pi r^2$, so one may consider
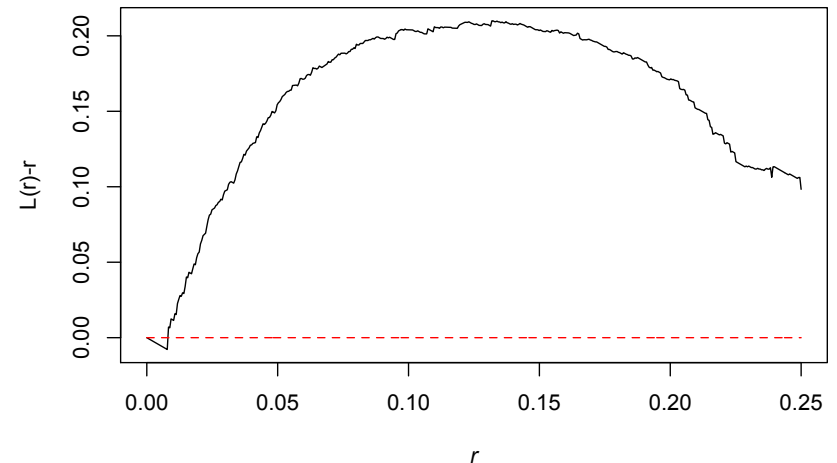$L(r) = \sqrt{(K(r)/\pi)}$.
For a stationary Poisson process, $L(r) - r = 0$ and $\hat{L}(r) - r$ should be approximately 0.

5. More exercises.

Which of the following is not typically true of the MLE of a spatial-temporal point process?

a. It is unbiased.
b. It is consistent.
c. It is asymptotically normal.
d. It is asymptotically efficient.

5. More exercises.

Which of the following is not typically true of the MLE of a spatial-temporal point process?

a. It is unbiased.
b. It is consistent.
c. It is asymptotically normal.
d. It is asymptotically efficient.

5. More exercises.

Suppose N is a spatial point process with clustering for distances ≤ d.
Let F(r) be the empty space function and let G(r) be the nearest neighbor distribution
function.
Which of the following is true.

a. F(d) = G(d).
b. F(d) < G(d).
c. F(d) > G(d)

5. More exercises.

Suppose N is a spatial point process with clustering for distances ≤ d.
Let F(r) be the empty space function and let G(r) be the nearest neighbor distribution function.
Which of the following is true.

a. F(d) = G(d).
b. F(d) < G(d).
c. F(d) > G(d)

Entering data and kernel smoothing example.

First, input 54 points using the mouse.
n = 54
plot(c(0,1),c(0,1),type="n",xlab="longitude",ylab="latitude",
                    main="locations")

```
  x1 = rep(0,n)
  y1 = rep(0,n)
  for(i in 1:n){
  z1 = locator(1)
  x1[i] = z1$x
  y1[i] = z1$y
  points(x1[i],y1[i])
  }
```

PLOT THE POINTS WITH A 2D KERNEL SMOOTHING IN GREYSCALE PLUS A LEGEND

```
library(splancs)
bdw = sqrt(bw.nrd0(x1)^2+bw.nrd0(y1)^2)  ## possible default bandwidth
b1 = as.points(x1,y1)
bdry = matrix(c(0,0,1,0,1,1,0,1,0,0),ncol=2,byrow=T)
z = kernel2d(b1,bdry,bdw)
attributes(z)
par(mfrow=c(1,2))
image(z,col=gray((64:20)/64),xlab="km E of origin",ylab="km N of
    origin")
points(b1)
x4 = seq(min(z$z),max(z$z),length=100)
```

```
plot(c(0,10),c(.8*min(x4),1.2*max(x4)),type="n",
        axes=F,xlab="",ylab="")
image(c(-1:1),x4,matrix(rep(x4,2),ncol=100,byrow=T),
add=T,col=gray((64:20)/64))
text(2,min(x4),as.character(signif(min(x4),2)),cex=1)
text(2,(max(x4)+min(x4))/2,
        as.character(signif((max(x4)+min(x4))/2,2)),cex=1)
text(2,max(x4),as.character(signif(max(x4),2)),cex=1)
mtext(s=3,l=-3,at=1,"density (pts/km^2)")

library(spatstat)
b2 = as.ppp(b1,c(0,1,0,1))
k = Kest(b2,correction="border")

plot(k, main="K function")

plot(k, sqrt(./pi)-r ~ r, ylab="L(r)-r", main="L function",legend=F)
```

```
plot(Fest(b2,correction="border"),main="")
plot(Gest(b2,correction="border"),main="")
plot(Jest(b2,correction="border"),main="")
```