Lecture 9
Point Processes

Dennis Sun
Stats 253

July 21, 2014

Outline of Lecture

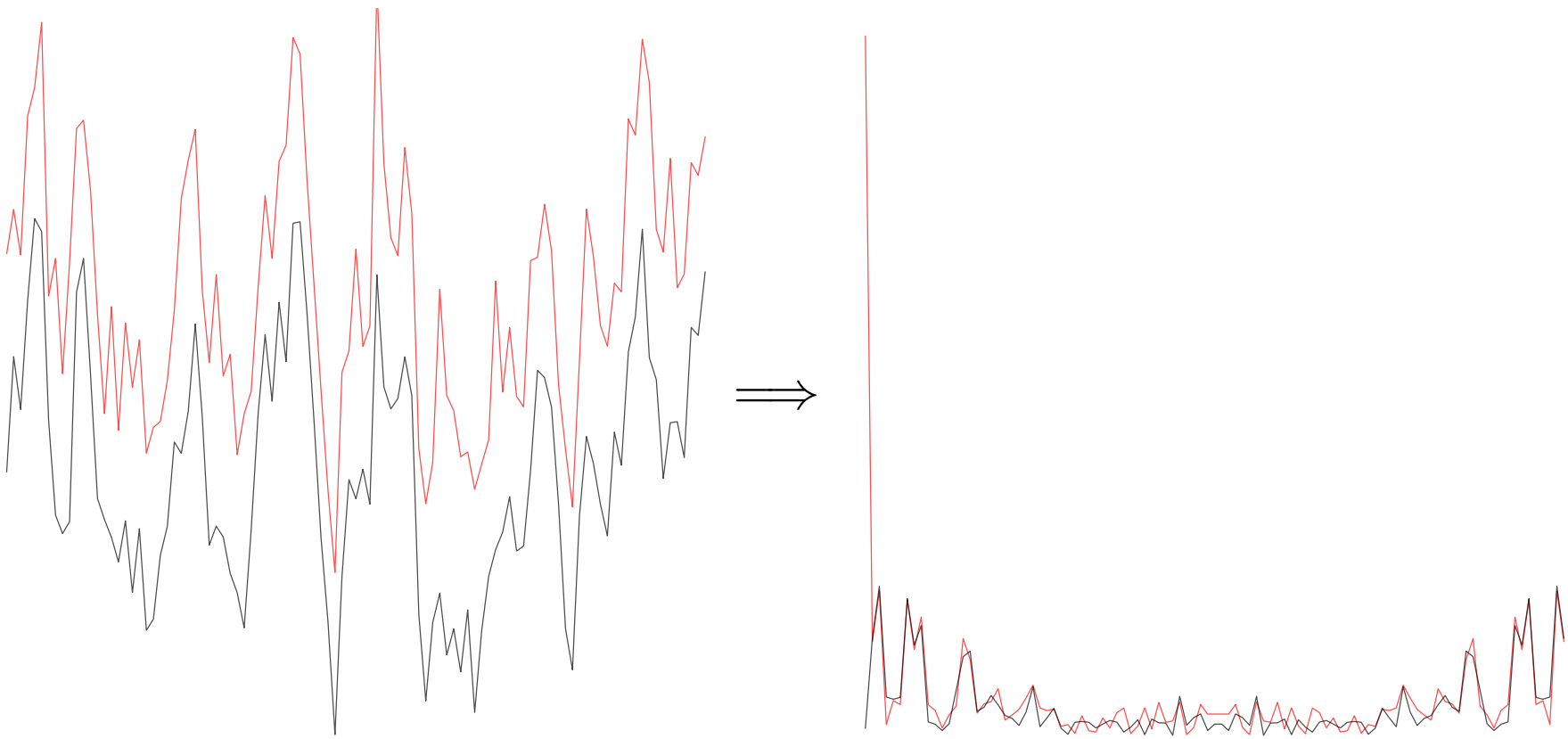
1. Last Words about the Frequency Domain
2. Point Processes in Time and Space
3. Inhomogeneous Poisson Processes
4. Second-Order Properties
5. Wrapping Up

Where are we?

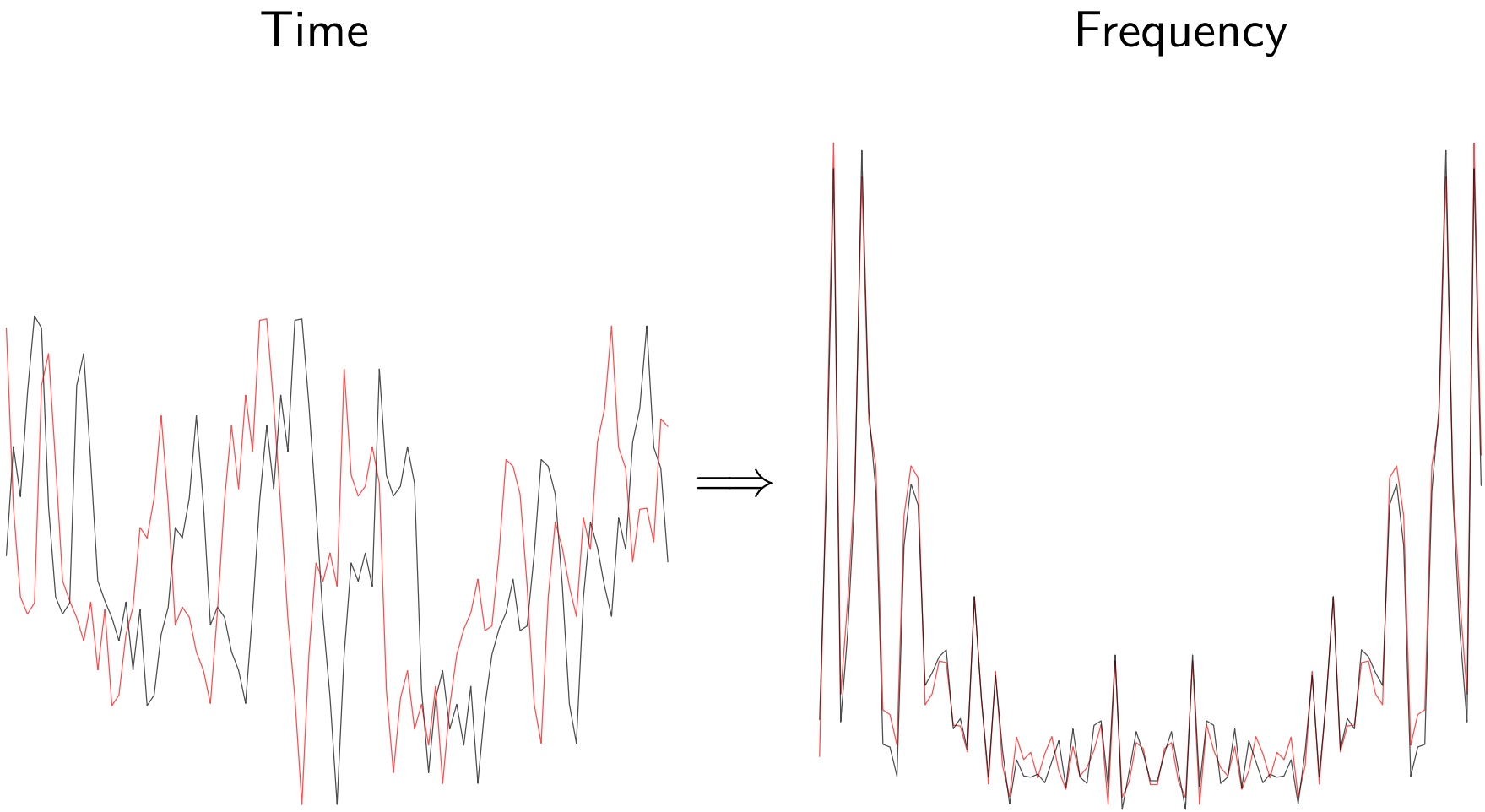
Why is the frequency domain useful?

Time

Frequency



Why is the frequency domain useful?



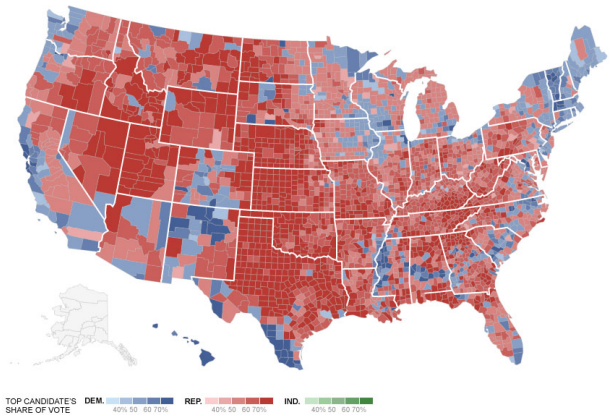
Why is the frequency domain useful?

The frequency domain more often captures our intuition about when two signals are "similar".

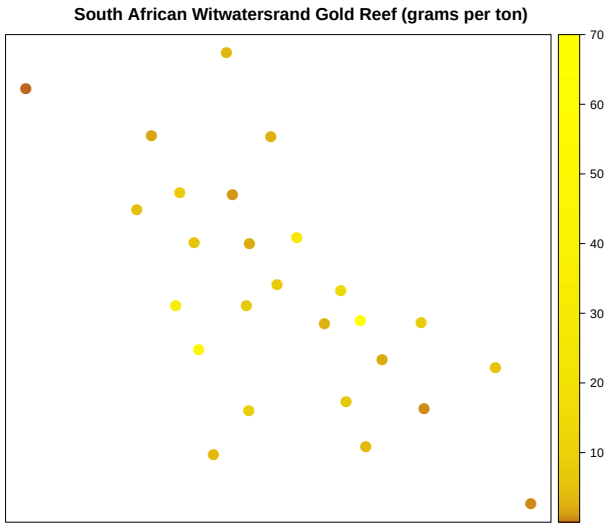
Where are we?

Trinity of Spatial Statistics

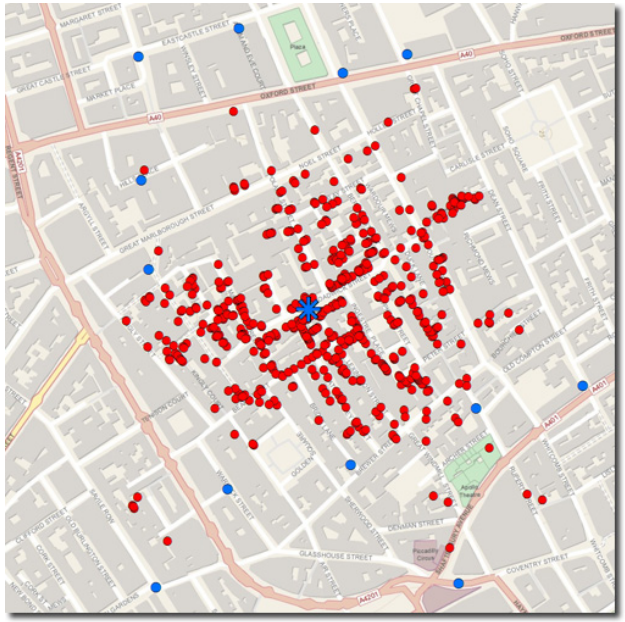
Today we complete the "trinity" of spatial statistics...



Lattice Data

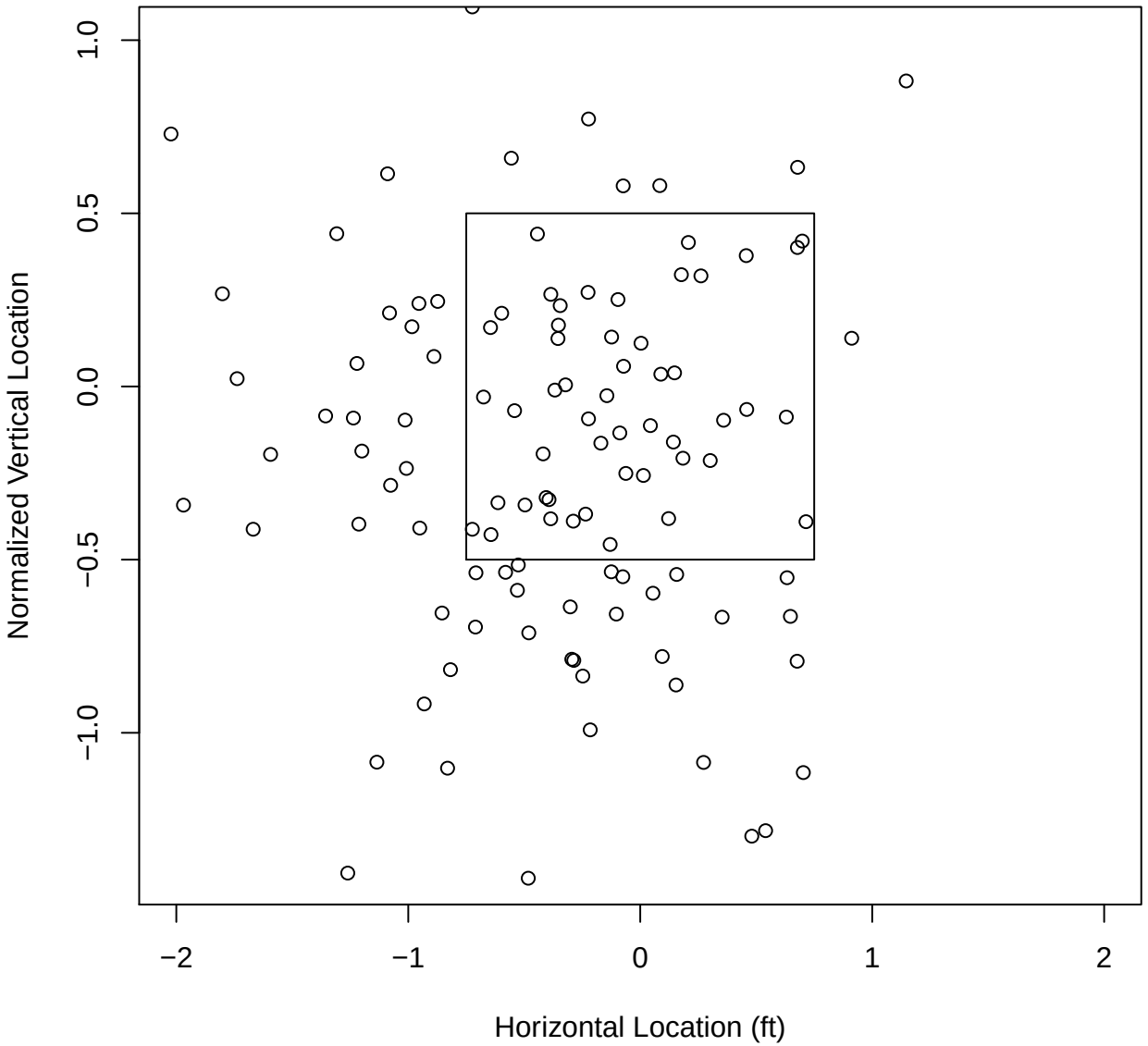


Geostatistics

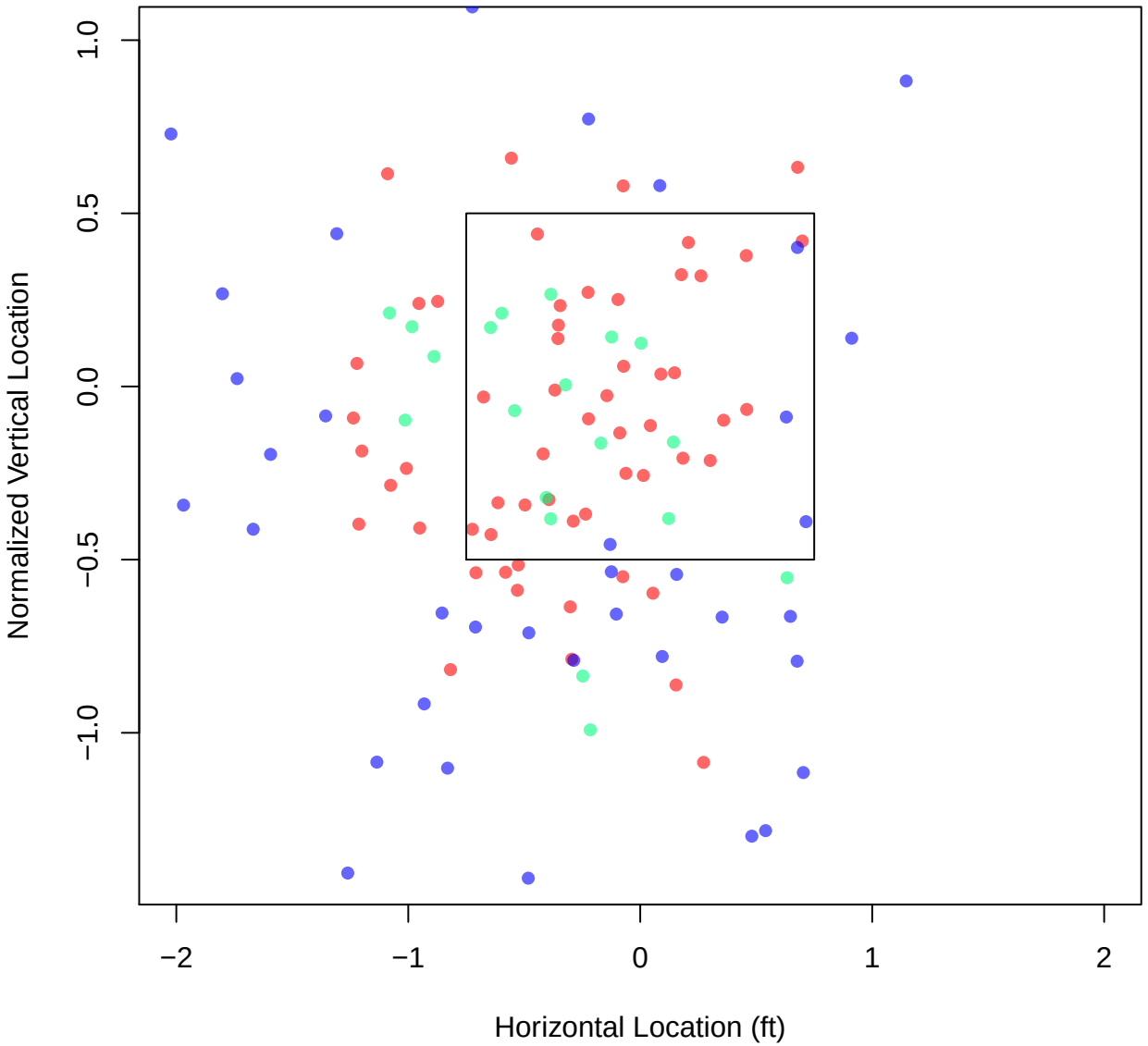


Point Processes

Point Processes



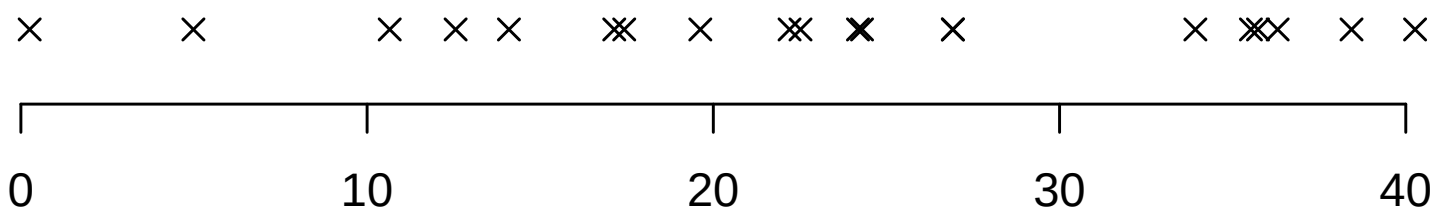
Marked Point Processes



Point Processes

- The distinguishing feature of point processes is that the locations $\boldsymbol{s}_i$ are now random.
- There may or may not be labels $y(\boldsymbol{s}_i)$ associated with the points.
- Basic model: Poisson processes

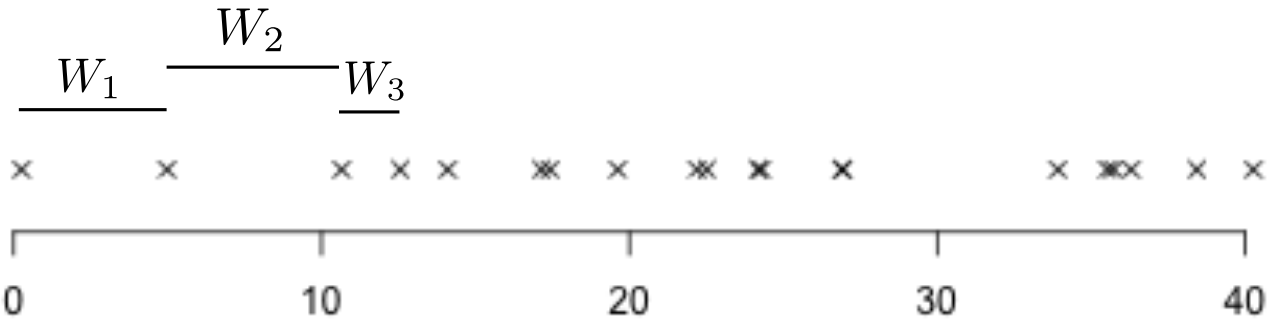
Poisson Processes in Time



- $N(t)$: number of events that have occurred up to time t
- Properties:
  1. $N(t+h) - N(t) \sim \text{Pois}(\lambda h)$
  2. If $s < t < u < v$, then $N(t) - N(s)$ is independent of $N(v) - N(u)$.
- **Question:** Is this **well-defined**? (i.e., Can I simulate a process that has these properties?)

Simulating a Temporal Poisson Process: Method 1

- Generate **waiting times** $W_i \overset{iid}{\sim} \text{Exp}(\lambda)$.

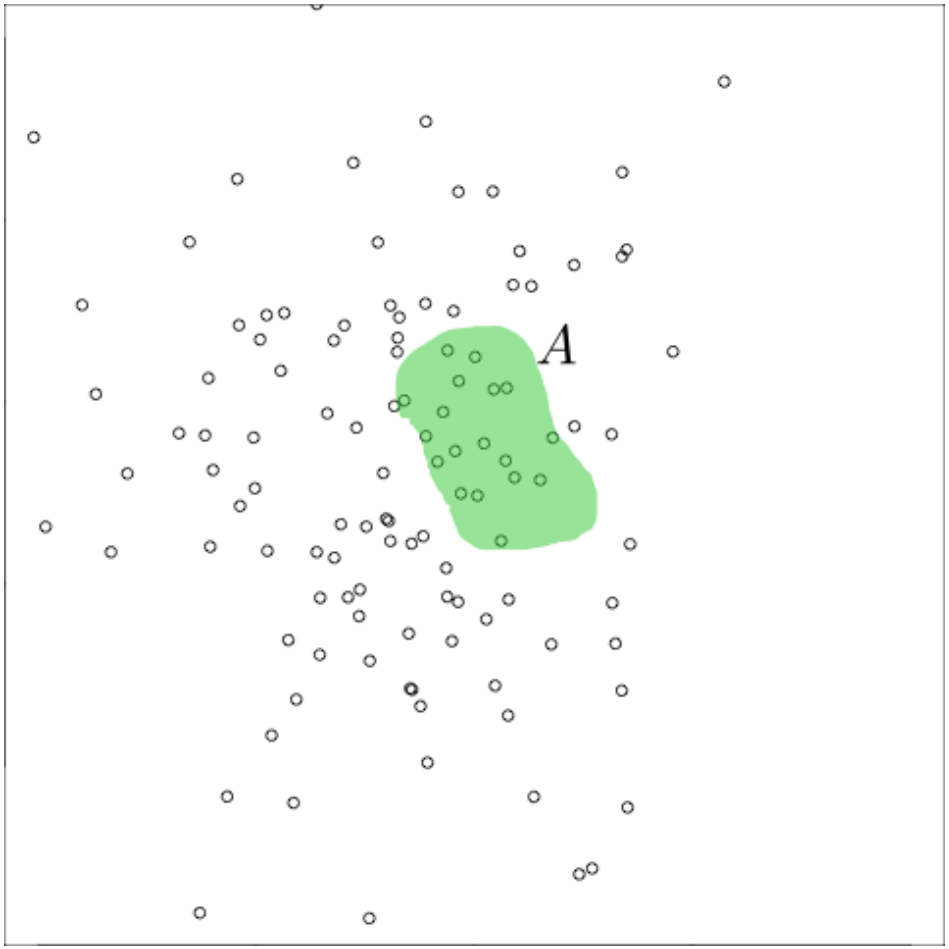


- Then, the first event occurs at time W_1, the second at time $W_1 + W_2$, etc.
- Formally, $N(t) = \max\{k : \sum_{i=1}^{k} W_i \leq t\}$.
- **Claim:** $N(t)$ has the desired properties.
  1. $N(t+h) - N(t) \sim \text{Pois}(\lambda h)$
  2. If $s < t < u < v$, then $N(t) - N(s)$ is independent of $N(v) - N(u)$.

Simulating a Temporal Poisson Process: Method 2

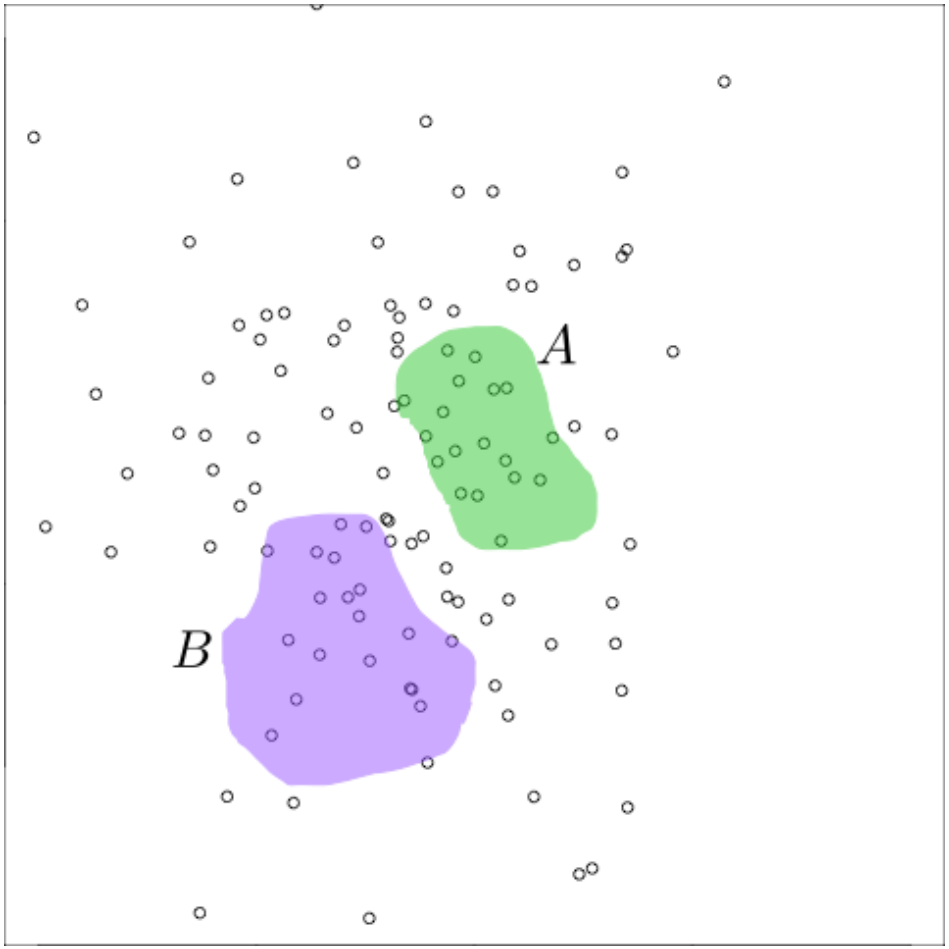
- This method simulates a Poisson process on $[0, T]$.
- Generate $N(T) \sim \text{Pois}(\lambda T)$.
- Generate $\mathbf{s_i} \sim \text{Unif}(0, T)$ for $i = 1, ..., N(T)$.
- **Question:** Why is $N(t + h) - N(t) \sim \text{Pois}(\lambda h)$?

How do we generalize this to space?



$$N(A) \sim \text{Pois}(\lambda|A|)$$

Poisson Processes in Space



$N(A)$ and $N(B)$ are independent if $A \cap B = \emptyset$.

Is this well-defined?

- We can generalize Method 2 for temporal processes:
  - Generate $N(D) \sim \text{Pois}(\lambda|D|)$.
  - Generate $\boldsymbol{s}_i \sim \text{Unif}(D)$, $i = 1, ..., N(D)$.
- Method 1 doesn't generalize. ☹
- Once again, the moral of this class:
  Temporal processes are easy, spatial processes are hard.

Parameter Estimation

- How do you estimate λ?
- Easy: $\hat{\lambda} = \frac{N(D)}{|D|}$.
- Many ways to justify this, e.g., MLE.

Where are we?

The Inhomogeneous Poisson Process

- This model is restrictive: assumes points are equally likely to be anywhere in D.
- Generalization: suppose there is a density $\lambda(\cdot)$ on D. Then

$$N(A) \sim \text{Pois}\left(\int_A \lambda(\boldsymbol{s})\, d\boldsymbol{s}\right).$$

- This is called an **inhomogeneous Poisson process**.

Testing for Homogeneity

- How do we test if our process is homogeneous? (Another term is **complete spatial randomness**.)
- Let's try to generate a homogeneous process in R.
- General strategy for devising tests:
  - Come up with some test statistic, e.g.,

$$\hat{G}(r) = \frac{1}{N(D)} \#\{\boldsymbol{s}_i \text{ whose nearest neighbor is closer than } r\}$$

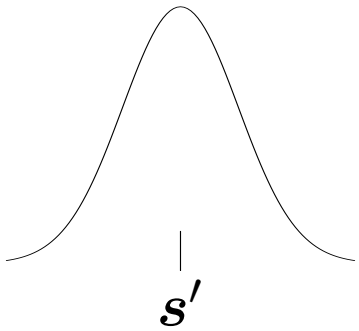
$$\hat{K}(r) = \frac{1}{\hat{\lambda}} \frac{\#\{(i,j) : d(\boldsymbol{s}_i, \boldsymbol{s}_j) \leq r\}}{N(D)}$$

  - Simulate test statistic under null hypothesis.
  - Compare with observed value of test statistic.

Estimating $\lambda(\cdot)$

Kernel density estimation: Choose a kernel K that integrates to 1:

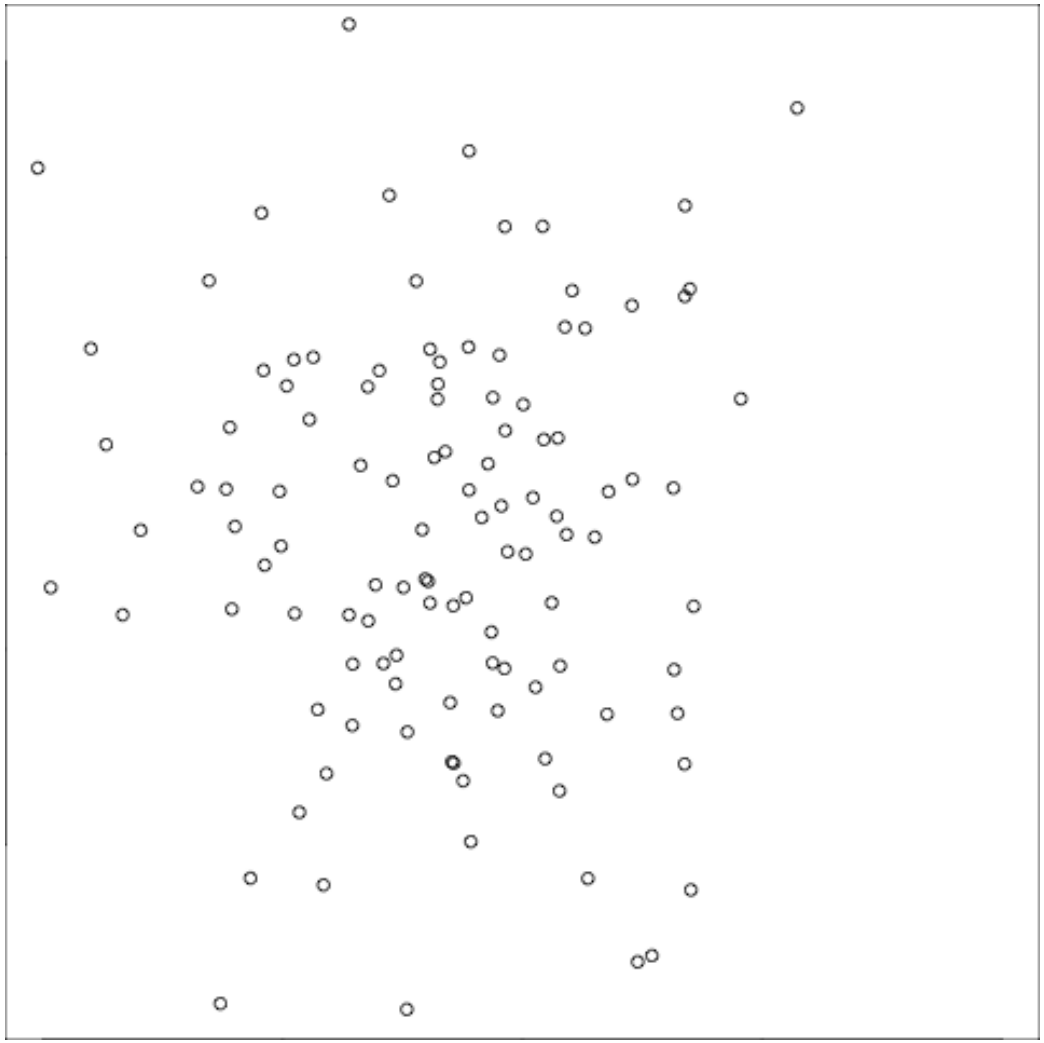
$$\int K(\boldsymbol{s}, \boldsymbol{s}')\, d\boldsymbol{s} = 1.$$



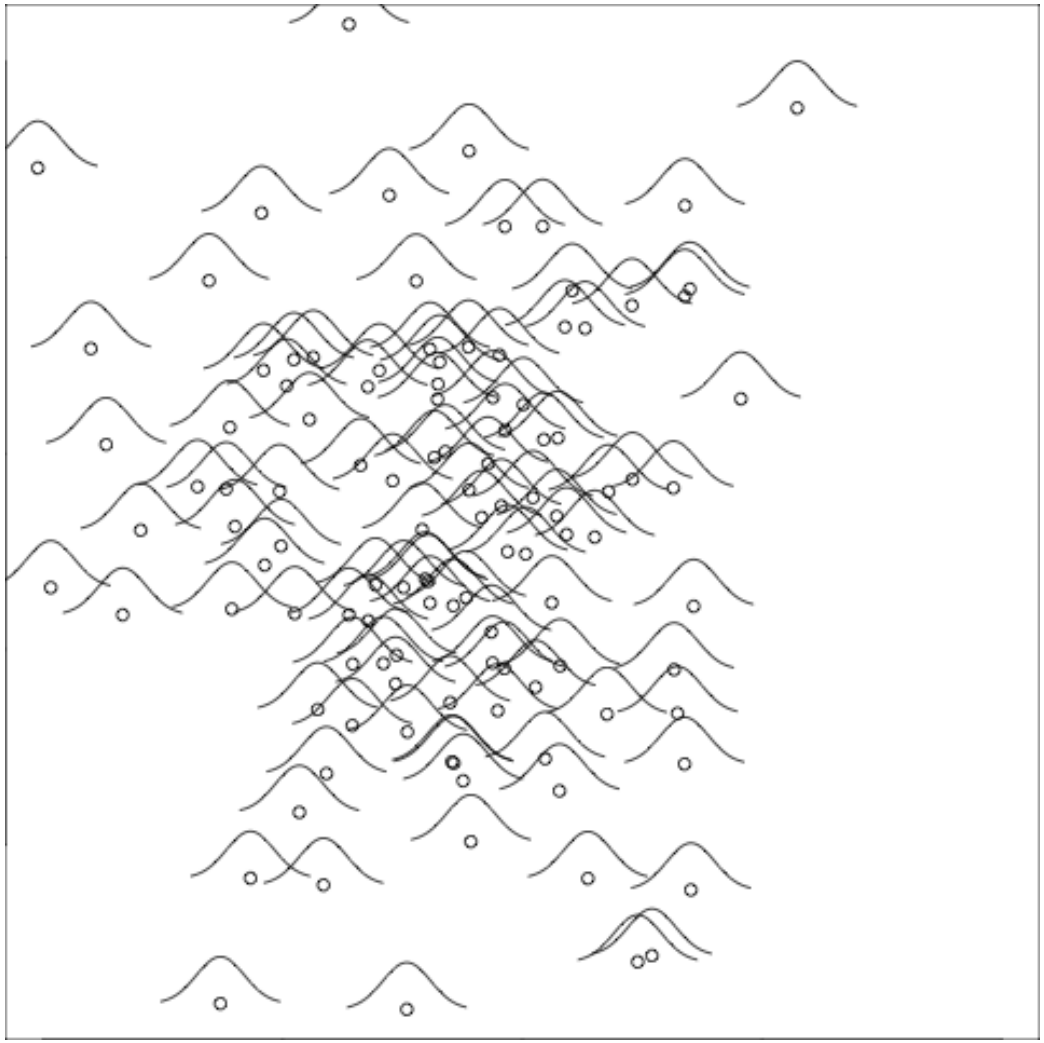
Place a kernel centered at each $\boldsymbol{s}_i$. Then the estimator is

$$\hat{\lambda}(\boldsymbol{s}) = \sum_{i=1}^{N(D)} K(\boldsymbol{s}, \boldsymbol{s}_i).$$

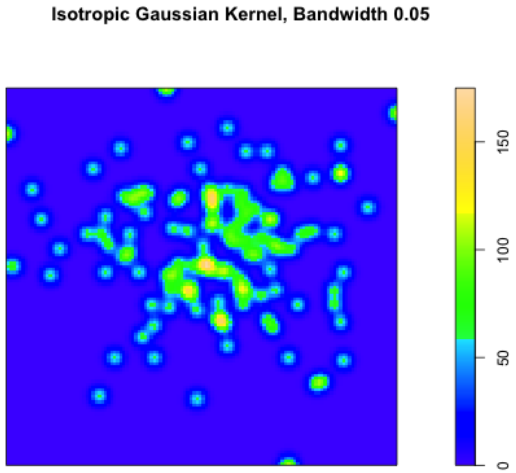
Estimating $\lambda(\cdot)$

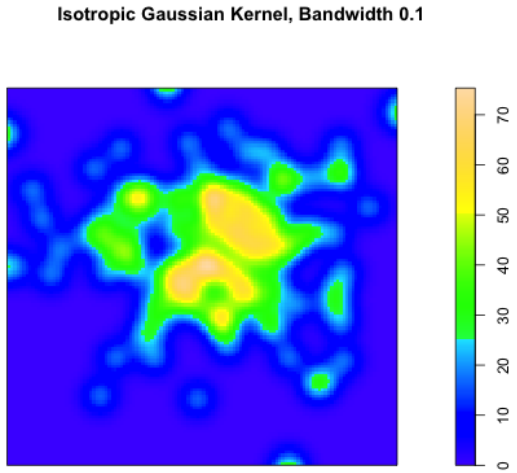


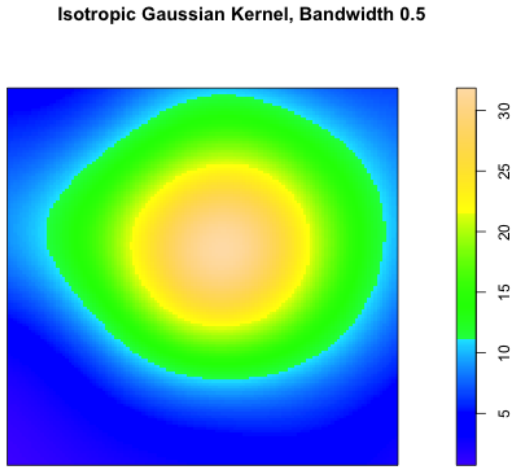
Estimating $\lambda(\cdot)$



Estimating $\lambda(\cdot)$







May need to correct for edge effects:

$$\hat{\lambda}(\boldsymbol{s}) = \sum_{i=1}^{N(D)} \frac{1}{\int_D K(\boldsymbol{s}, \boldsymbol{s}_i)\, d\boldsymbol{s}_i} K(\boldsymbol{s}, \boldsymbol{s}_i).$$

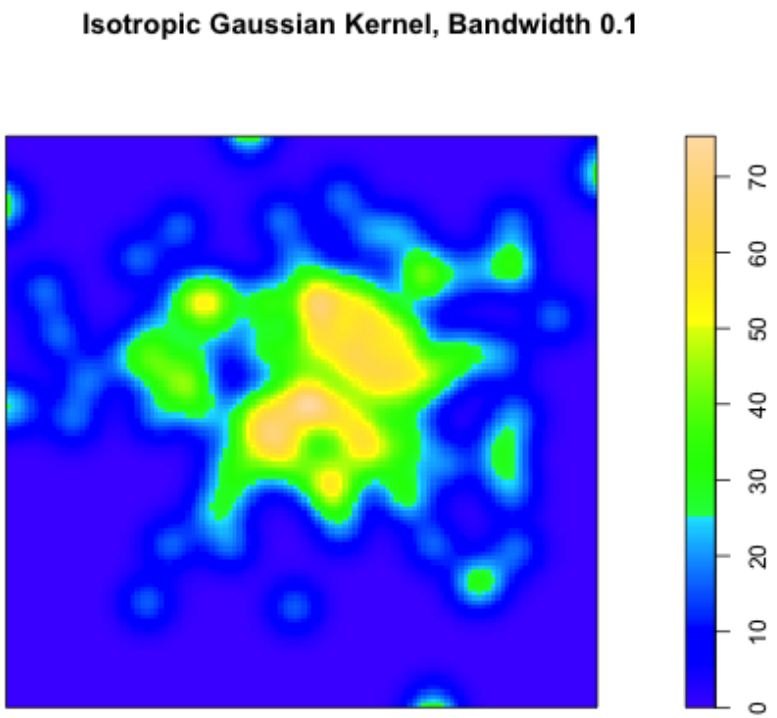
Final Issues

- How do you simulate from an inhomogeneous point process?
- Also possible to model $\lambda(\boldsymbol{s})$ parametrically, e.g.,

$$\log \lambda(\boldsymbol{s}) = \boldsymbol{x}(\boldsymbol{s})^T \boldsymbol{\beta}$$

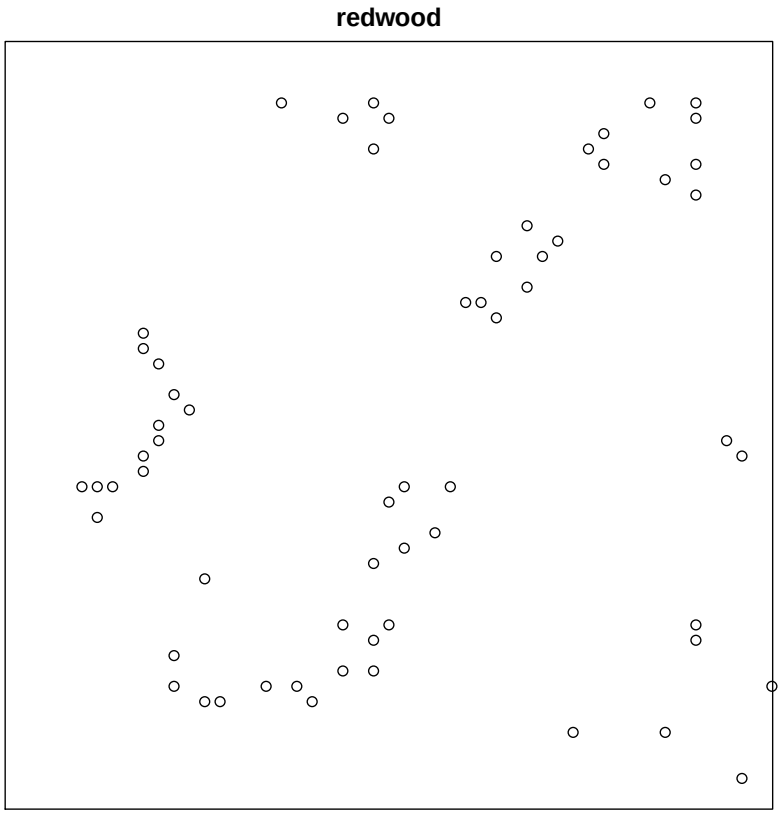
Where are we?

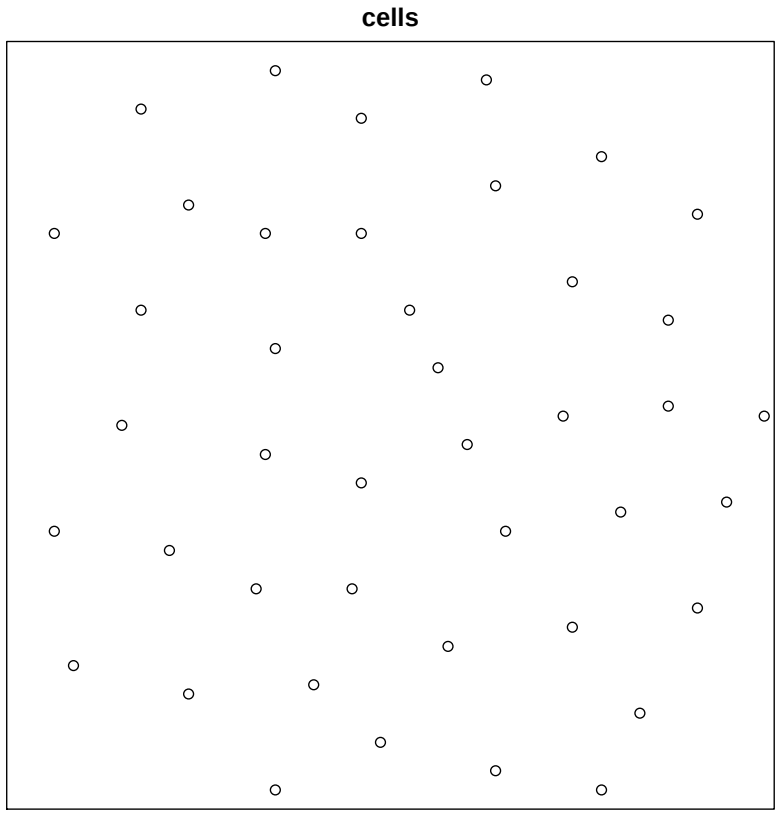
Why Second-Order Properties?



Processes may still exhibit **clustering** or **inhibition**.

Examples of Clustering and Inhibition





Second-Order Intensity

$$\lambda(\boldsymbol{s}) = \lim_{|d\boldsymbol{s}| \to 0} \frac{\mathrm{E}(N(d\boldsymbol{s}))}{|d\boldsymbol{s}|} \qquad \mu(\boldsymbol{s}) = \mathrm{E}(y(\boldsymbol{s}))$$

$$\lambda_2(\boldsymbol{s}, \boldsymbol{s}') = \lim_{|d\boldsymbol{s}| \to 0} \frac{\mathrm{E}(N(d\boldsymbol{s})N(d\boldsymbol{s}'))}{|d\boldsymbol{s}||d\boldsymbol{s}'|} \qquad \Sigma(\boldsymbol{s}, \boldsymbol{s}') = \mathrm{E}(y(\boldsymbol{s})y(\boldsymbol{s}')) - \mu(\boldsymbol{s})\mu(\boldsymbol{s}')$$

λ_2 is called the **second-order intensity**.

A process is stationary if $\lambda(\boldsymbol{s}) \equiv \lambda$ and $\lambda_2(\boldsymbol{s}, \boldsymbol{s}') = \lambda_2(\boldsymbol{s} - \boldsymbol{s}')$.

Ripley's K-function

- λ_2 is hard to estimate.
- For stationary and isotropic processes, K-function is used instead:

$$K(r) = \frac{1}{\lambda}\mathrm{E}\#\{\text{events within distance } r \text{ of a randomly chosen event}\}$$

$$= \frac{1}{\lambda}\mathrm{E}\left[\frac{1}{N(D)}\sum_{i=1}^{N(D)}\sum_{j\neq i} 1\{d(\boldsymbol{s}_i, \boldsymbol{s}_j) \leq r\}\right]$$

- This has a natural estimator:

$$\hat{K}(r) = \frac{1}{\hat{\lambda}}\frac{\#\{(i,j) : d(\boldsymbol{s}_i, \boldsymbol{s}_j) \leq r\}}{N(D)}$$

- General strategy for fitting models:

$$\underset{\theta}{\text{minimize}} \int_0^{r_0} w(r)(\hat{K}(r) - K_\theta(r))^2 \, dr$$

Handling Inhomogeneity

- What if $\lambda(\boldsymbol{s}) \not\equiv \lambda$ but is known?
- **Solution:** Re-weight distances by chance of observing event.

$$K_I(r) = \mathrm{E}\left[\frac{1}{|D|}\sum_{i=1}^{N(D)}\sum_{j\neq i}\frac{1\{d(\boldsymbol{s}_i,\boldsymbol{s}_j)\leq r\}}{\lambda(\boldsymbol{s}_i)\lambda(\boldsymbol{s}_j)}\right]$$

$$\hat{K}_I(r) = \frac{1}{|D|}\sum_{i=1}^{N(D)}\sum_{j\neq i}\frac{1\{d(\boldsymbol{s}_i,\boldsymbol{s}_j)\leq r\}}{\lambda(\boldsymbol{s}_i)\lambda(\boldsymbol{s}_j)}$$

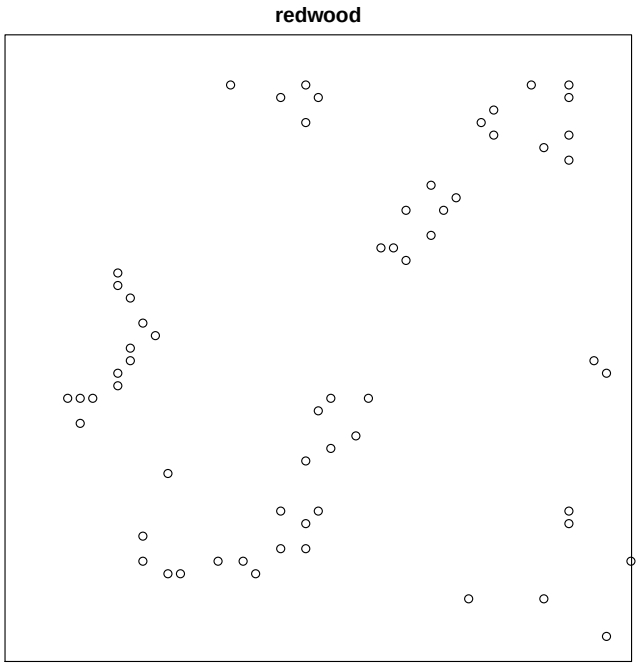
- All computations proceed with K_I and $\hat{K}_I$ instead of K and $\hat{K}$.

Modeling Clustering: Neyman-Scott Process

- Parent events are generated from Poisson process with intensity $\rho(\cdot)$.
- Each parent has S offspring, where $S \overset{iid}{\sim} p_S$.
- Offspring are located i.i.d. around their parent, according to some density function $f(\cdot)$.
- The process is the resulting offspring.

[**Check:** If $\rho(\cdot) \equiv \rho$ and $f(\boldsymbol{s}) = f(||\boldsymbol{s}||)$, then stationary and isotropic.]

Modeling Clustering: Neyman-Scott Process

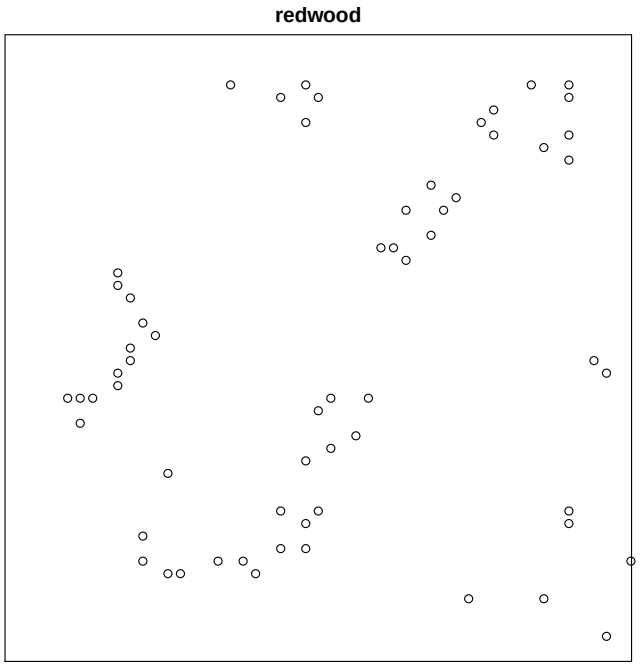



```
> cauchy.estK(redwood)
```

Fits Neyman-Scott process with Cauchy kernel:

- parents from Poisson process with intensity κ
- $S \sim \text{Pois}(\mu)$
- $f(\boldsymbol{s}) = \frac{1}{2\pi\omega^2}\left(1 + \frac{||\boldsymbol{s}||^2}{\omega^2}\right)^{-3/2}$

Modeling Clustering: Neyman-Scott Process




```
Minimum contrast fit (object of class minconf
Model: Cauchy process
Fitted by matching theoretical K function to
Parameters fitted by minimum contrast ($par):
       kappa         eta2
12.446917419  0.008454113
Derived parameters of Cauchy process ($modelp
      kappa       omega          mu
12.44691742  0.04597312  4.98115300
Converged successfully after 259 iterations
Starting values of parameters:
kappa  eta2
    1     1
Domain of integration: [ 0 , 0.25 ]
Exponents: p= 2, q= 0.25
```

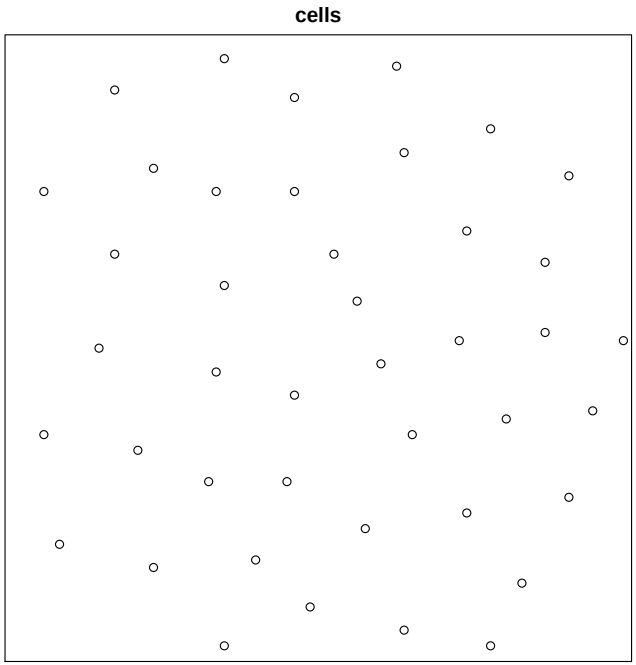
Modeling Inhibition: Strauss Process

$$p(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n) = \frac{\beta^n \gamma^{\#\{(i,j):\ d(\boldsymbol{s}_i, \boldsymbol{s}_j) \leq \delta\}}}{\alpha(\beta, \gamma)}$$

$\gamma \in [0, 1]$, so last term penalizes processes that have points close together.

- $\gamma = 1$: no inhibition
- $\gamma = 0$: no event can be within δ of another ("hard-core" model)

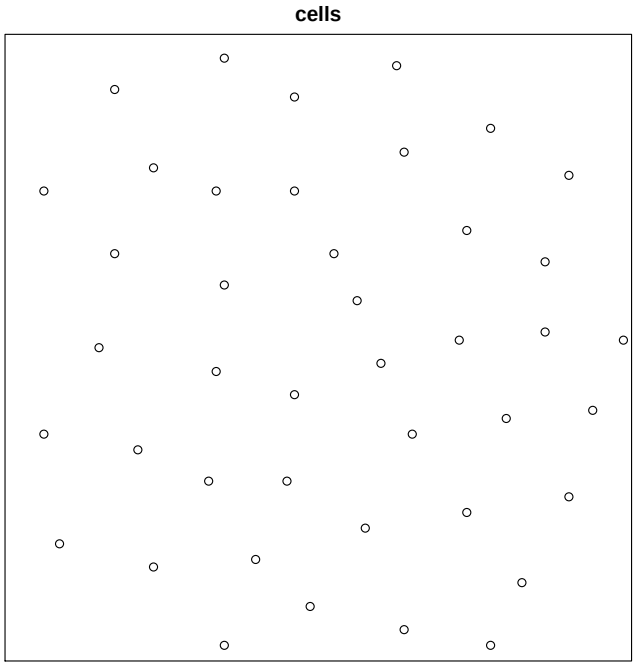
Modeling Inhibition: Strauss Process




```
> ppm(Q = cells, trend = ~ 1,
      interaction = Strauss(0.1))
```

Fits Strauss model with $\delta = 0.1$.

Modeling Inhibition: Strauss Process




```
Stationary Strauss process

First order term:
    beta
563.0123

Interaction: Strauss process
interaction distance: 0.1
Fitted interaction parameter gamma:  0.0389

Relevant coefficients:
Interaction
  -3.247058

For standard errors, type coef(summary(x))
```

Where are we?

Summary

- Point processes are distinguished by the randomness of the locations.
- Poisson processes are the simplest model for point processes.
- More complex models (Neyman-Scott, Strauss) can capture second-order interactions.
- Use the `spatstat` package in R!

Homeworks

- Homework 3 deadline extended to Friday.
- Winners of prediction competition will be announced next Monday.
- Homework 4a and b: students doing a project need only complete one of these.
- Homework 4a will be posted Wednesday, 4b by the end of the week.

Projects

- Please submit your project proposal if you haven't done so already!
- I will start responding to them today.