Simulating spatial data

- We've talked about simulating simple point patterns
  - Inference was via simulation
  - Does observed summary function "look like" simulated patterns?
- Now consider simulating geostatistical and areal data
- Given a set of locations $\boldsymbol{s}$, a model, and parameter values want to generate a set of values for $\boldsymbol{Z(s)}$
- Focus on values from normal distributions, want $N(\mu, \sigma^2)$
- If $\boldsymbol{Z}$ independent, easy: generate $\boldsymbol{Z} \sim N(0, 1)$ calculate: $\sigma \boldsymbol{Z} + \mu$
- If spatially correlated: want $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$

Why simulate data?

- Want to know about some summary of the spatial data
  - What proportion of the Swiss Zura has Zn > 10?
  - Compute from map of prediction
  - Many summary statistics: ignoring uncertainty $\Rightarrow$ biased summary
  - Better to simulate 5-10 data sets, summarize each, average
- To better understand uncertainty
  - In a summary, or a map
- Inference when theory inadequate
  - Often inadequate with non-normal distributions
  - Or when looking at the covariance parameters

Simulating correlated data

- a brute-force algorithm:
  - calculate μ or $\mu(s)$ for each location if trend
  - determine $\boldsymbol{\Sigma}$ from geostat model or equ's for CAR/SAR
  - calculate $\boldsymbol{C}$ = Cholesky square-root decomposition of $\boldsymbol{\Sigma}$. $\boldsymbol{C}'\boldsymbol{C} = \boldsymbol{\Sigma}$
  - simulate vector of standard normals, $\boldsymbol{Z} \sim N(\boldsymbol{0}, \boldsymbol{I})$
  - return $\boldsymbol{Z(s)} = \boldsymbol{\mu} + \boldsymbol{C}'\boldsymbol{Z}$
- Detail:
  - Matrix algebra defines $\boldsymbol{C}$ as a lower triangular matrix
  - R chol() function returns an upper triangular matrix,
  - Above formulae are correct for R parameterization

Why does this work?

- Mean:

$$\text{E } \boldsymbol{Z(s)} = \mu + \boldsymbol{C}' \text{ E } \boldsymbol{Z} = \mu$$

- Variance:

$$\text{Var } \boldsymbol{Z(s)} = \boldsymbol{C}' \text{ Var } \boldsymbol{Z}\boldsymbol{C} = \boldsymbol{C}'\boldsymbol{I}\boldsymbol{C} = \boldsymbol{C}'\boldsymbol{C} = \boldsymbol{\Sigma}$$

- Distribution: linear combinations of normals are normal
- Example:

$$\boldsymbol{\Sigma} = \begin{bmatrix} 3 & 2 & 1 \\ 2 & 3 & 2 \\ 1 & 2 & 3 \end{bmatrix}, \quad \boldsymbol{C} \approx \begin{bmatrix} 1.75 & 1.15 & 0.58 \\ 0 & 1.29 & 1.03 \\ 0 & 0 & 1.26 \end{bmatrix}$$

  - $Z_{s_1} = 1.73 Z_1$
  - $Z_{s_2} = 1.15 Z_1 + 1.29 Z_2$
  - $Z_{s_3} = 0.58 Z_1 + 1.03 Z_2 + 1.26 Z_3$

- Timing: k = 50 observations,
  - simulate 1000 sets separately: 7.18 sec
  - simulate all 1000 simultaneously: 0.04 sec
  - Difference is time req. to calculate the Cholesky
- Practical use:
  - either calculate $\boldsymbol{C}$ once, do $\boldsymbol{Z(s)}$ "by hand"
  - or, simulate many sets, use as needed
- Cholesky algorithm fails if $\boldsymbol{\Sigma}$ is large,
- In fact, working with $\boldsymbol{\Sigma}$ is difficult
  - 1000 locations, $\boldsymbol{\Sigma}$ is 1000 x 1000 - huge

Better algorithms

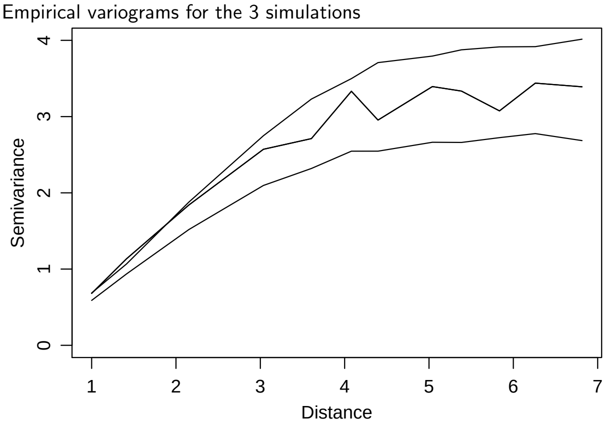
- Many choices: usual goal is reduce memory demand
- RandomFields has 11 for Gaussian data
- concept for one: "turning bands" algorithm
  - simulate a direction θ_k (will have many of these)
  - simulate Z's in chunks along that line (1D problem)
  - for any s, project s (in 2D) onto the line, record Z at that projected location
  - repeat for many (e.g. 10 - 15) directions, average contributions from all directions
  - picture on next slide (will be hand-drawn)
  - The detail is relating the 2D covariance function for $\boldsymbol{Z(s)}$ to the corresponding 1D covariance function for the line
  - The advantage is not memory intensive
    - don't have to work with N×N matrices
    - so can use for LARGE problems
  - And extremely fast
    - Because easy to simulate chunks along a line
  - Turning bands is my 'go-to' algorithm, but glad I don't have to code it

Unconditional and conditional simulation

- Cholesky and turning bands generate unconditional simulations
- Have similar trends and spatial correlation as the data
  - But, μ and $\boldsymbol{\Sigma}$ will be similar
  - And more similar with large sample size
- But, no connection to the observed values
  - Z may look very different
  - Specifically new $Z(s)$s at a sample location will vary
- Demonstrate with 3 simulated datasets
- Show the empirical variograms (as lines) then 3 data plots

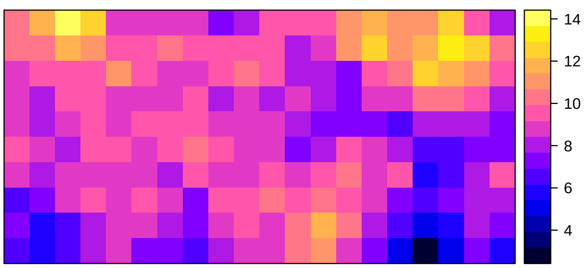
Unconditional simulation:

Empirical variograms for the 3 simulations



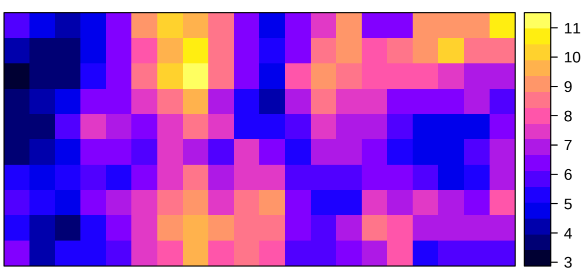
Unconditional simulation

Matern, k=1, p.sill=4, nugget=1, range=3, 20 x 10 grid



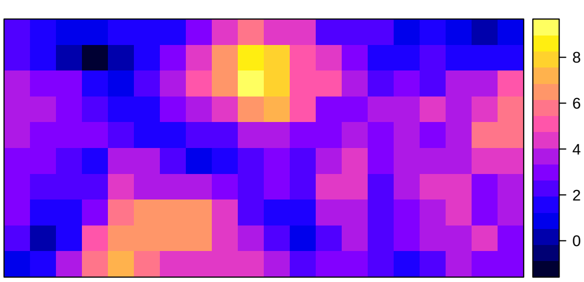
Unconditional simulation

Matern, k=1, p.sill=4, nugget=1, range=3, 20 x 10 grid



Unconditional simulation

Matern, k=1, p.sill=4, nugget=1, range=3, 20 x 10 grid



Conditional Simulation:

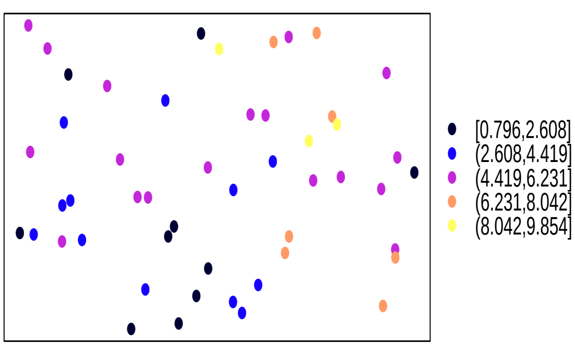
- Honor the observed data
  - Simulate of new values conditional on obs. values
  - predictions at any observed location are **always** the original value
- Given values at obs. locations, simulate values at other points
- Two sets of locations:
  - $\boldsymbol{s}_c$: locations in the original data set
  - $\boldsymbol{s}_n$: new locations where you want conditional predictions
- And one observed set of values: $Z(\boldsymbol{s}_c)$
- want to simulate $Z(\boldsymbol{s}_n$, the new random values at $\{\boldsymbol{s}_n\}$

Conditional simulation: the usual algorithm

- Uses three sets of predictions to make sure that $Z(\boldsymbol{s}_c)$ are constant
- calculate kriging predictions $= Z^*(\boldsymbol{s}_n)$ using values at $Z(\boldsymbol{s}_c)$
- simulate unconditional random field at $\{\boldsymbol{s}_c\} = Z^\circ(\boldsymbol{s}_c)$
- simulate 2nd unconditional random field at $\{\boldsymbol{s}_n\} = Z^\circ(\boldsymbol{s}_n)$
- calculate kriging predictions $= Z^\dagger(\boldsymbol{s}_n)$ using values at $Z^\circ(\boldsymbol{s}_c)$
- return $Z^*(\boldsymbol{s}_n) + Z^\circ(\boldsymbol{s}_n) - Z^\dagger(\boldsymbol{s}_n)$

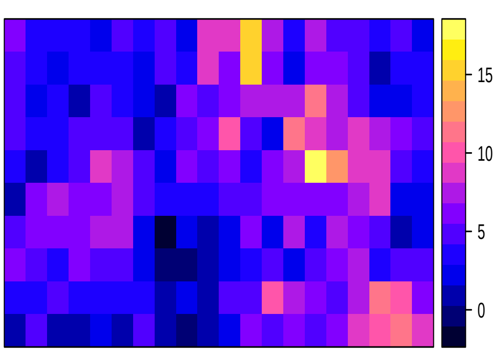
Conditional simulation in pictures

Observed data (simulated values, not a "real" dataset)



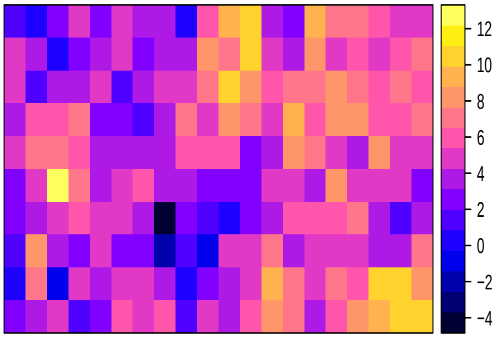
Conditional simulation in pictures

Conditional simulation # 1



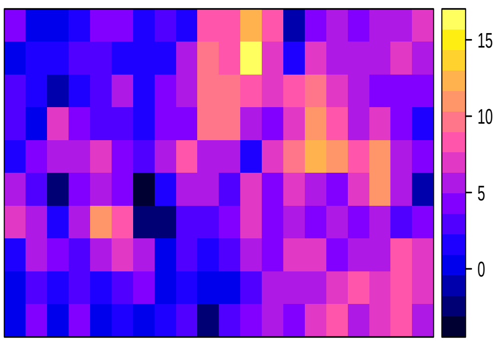
Conditional simulation in pictures

Conditional simulation # 2



Conditional simulation in pictures

Conditional simulation # 3



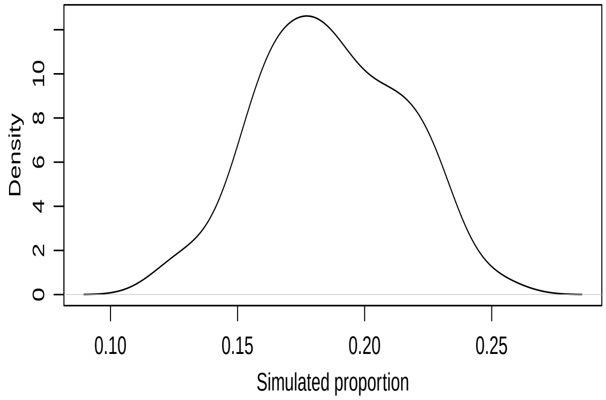
Conditional simulation properties

- If $\boldsymbol{s}$ is a conditioning point (obs. value), i.e. one of the locations in the $\{\boldsymbol{s}_c\}$ set,
  - 1st kriging prediction: $Z^*(\boldsymbol{s}_c) =$ obs. value, $Z(\boldsymbol{s}_c)$
  - 2nd kriging prediction: $Z^\dagger(\boldsymbol{s}_c) =$ obs. value, $Z^\circ(\boldsymbol{s}_c)$
  - so returned value is obs. value, $Z(\boldsymbol{s}_c)$
- If $\boldsymbol{s}_n$ is far from any obs. loc, $\boldsymbol{s}_c$:
  - $Z^*(\boldsymbol{s}_n) = \mu$ and $Z^\dagger(\boldsymbol{s}_n) = \mu$
  - so return the unconditional predictions, $Z^\circ(\boldsymbol{s}_n)$
- Both behaviours for extreme situations "make sense"
- Usually only used for geostat data.
  - With areal data, have an obs. value for all regions in study area

How could we use this?

- Given observed values, what fraction of the area > 7?
  - Estimate by ordinary kriging to predict at fine grid
  - Estimate proportion of predictions > 7
  - I don't have that estimate: let's say it's 20% of area
- How uncertain?
  - Conditional simulation given data
  - Three simulations: 19%, 18.5%, 21.5%
  - 100 simulations: mean = 18.7%, sd = 2.8%

Conditional estimates of proportion > 7



Simulating point patterns

- Have seen simulating CSR, without discussing details
- Big question: is N known or random?
  - Known: every realization has 100 (or 224, or 59) points
    Binary process: N fixed
  - Random: $N \sim$ some distribution, N not constant
    Poisson process: $N \sim$ Pois $(\lambda\, A)$
    - simulate N, then simulate locations of N events

Simulating point patterns

- 2nd question: is study area rectangular or irregular
  - rectangular, L_x by L_y: $X \sim$ Unif $(0, L_x)$, $Y \sim$ Unif $(0, L_y)$
  - irregular:
    - find bounding box
    - simulate within bounding box
    - keep observations within study region
  - How many events to simulate in the bounding box?
    - Poisson: $N_{bb} \sim$ Pois (λ BBox area), gives Pois (λA) in study area
    - Binary: $N_{bb} = 1.2\lambda$ BBox area
      keep first N events. 1.2 is ad hoc. Can also simulate sequentially.

Simulating locations with trend

- What if $\lambda(s) = f(X(s))$?
- Use a rejection algorithm (Lewis and Shedler)
  - Find $L_m = \max \lambda(s)$ in the study region
  - Simulate $L_m A$ locations $(s_1, s_2, \cdots s_k)$
  - Calculate $p_i = \lambda(s_i)/L_m$ for each event
  - Retain the point with probability p_i
    - i.e., simulate $U_i \sim$ Unif $(0, 1)$ for each event
    - retain the point if $U_i \le p_i$
- Intensity at location $s_i = L_m p_i = \lambda(s_i)$

Simulating non-Poisson processes

- Neyman-Scott: follow the definition
  - Simulate k locations for mothers
  - For each mom, simulate $N_i \sim$ Pois (μ) # of daughters
  - Simulate locations of each daughter around Mom
- Strauss (inhibition) processes
  - Harder, usually done with a sequential algorithm
  - given set of locations (current events)
  - simulate potential location of next event, s_{new}
  - use inhibition model to calculate $\lambda(s_{new})$
  - retain with probability $\lambda(s_{new})/\lambda$

Pattern reconstruction

- What if you don't have a model (or don't trust your model)?
- Pattern reconstruction generates random patterns "like" some observed pattern
- You specify what characteristics that should match
  - such as $K(r)$ and nearest-neighbor distance $D(r)$
- Basic idea, to match observed locations O
  - Simulate an arbitrary set of locations: L_1
  - Randomly delete one location and simulate another: L_2
  - For both sets, L_1 and L_2 calculate "Energy"
    - quantifies discrepancy between O and L_i
  - Keep the set with the lower energy
    - i.e., keep the new location if it improves the fit
  - Repeat until arbitrarily close to observed pattern

Pattern reconstruction

- An example of simulated annealing, a technique for optimization of difficult problems
- Lots of details that I'm skipping
- More complete descriptions are:
  - Wiegand and Moloney, pp 276-287
  - Illian et al. pp 407-415
- Wiegand et al 2013 *Ecography* considered which summary statistics provide the most information for reconstructing patterns
- Implemented in the shar library (species-habitat associations)
  - Look at the relationships between species occurrences and habitat information
  - Need to account for potential correlation in occurrence.