1

Farm-to-Table

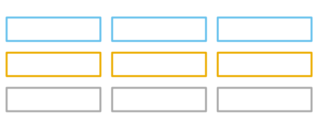
On-Farm Trial Data

2

Introduction

3

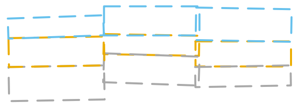
Uniformity Trial



- Plots of uniform size arranged in a lattice
- Spacing between plots can be controlled
- Units are exchangeable
  - Errors can be considered independent

4

Yield Monitor Data



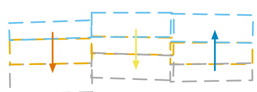
- Plots not uniform size and not at predetermined locations
- May be overlap between adjacent plots
- Units are not exchangeable
  - Errors can not be considered independent

5

Calculating Yield

$$Yield(mass/unit\ area) = \frac{Mass}{Width(unit) \times Length(unit)}$$

- Plots can be harvested independently, both in space and time.



$$Yield(mass/unit\ area) = \frac{Flow(mass/s)}{Width(unit) \times Speed(unit/s)}$$

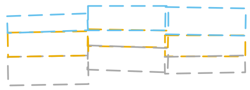
- Flow and speed are continuous processes, ordered in time.
- There can be mixing or lag in the processes, making errors dependent on time.

6

Types of Spatial Data

- Areal/Lattice
  - Sum or count over unit area.
- Geostatistical
  - Continuous value at a geo-tagged point.
- Point
  - Discrete value at a geo-tagged point.

Areal/Lattice

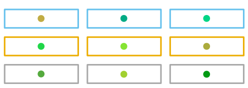


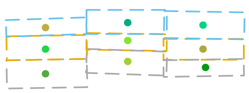
- Lattice data are sampled from a defined grid.
- Neighbors (rook, bishop, queen) are clearly defined

- Areal data come from arbitrary geographical divisions.
- Neighbors are not always clear.

7

Geostatistical Data





- Areal or lattice data can be mapped to geostastical data by associating measurements with a single point.
- Most of the concepts we will be covering are associated with geostatistical data.

8

Autocorrelated Data

9

Notation

10

We will be considering models of the forms

$$y_i = \mu + e_i \qquad y(\mathbf{s}_i) = \mu(\mathbf{s}_i) + e(\mathbf{s}_i)$$

where

- y_i is an outcome of interest, typically *yield*
- μ is a mean or expected value
- e_i is a random variable, *iid*, typically $\sim \mathcal{N}(0,\sigma^2)$
- $\mathbf{s}_i$ is a point in space, typically denoted (x,y)
- h is a measure of distance between two points

Blangiardo, M., & Cameletti, M. (2015). Spatial and Spatio-temporal Bayesian Models with R - INLA. John Wiley & Sons.

Gaussian Field

11

The set of values

$$y(\mathbf{s}_1), y(\mathbf{s}_2), \ldots, y(\mathbf{s}_n)$$

is a produced by a Gaussian field if, for every *i=1…n*,

- $y(\mathbf{s}_1), y(\mathbf{s}_2), \ldots, y(\mathbf{s}_n)$ has a multivariate normal distribution,
- mean $\boldsymbol{\mu} = \mu(\mathbf{s}_1), \mu(\mathbf{s}_2), \ldots, \mu(\mathbf{s}_n)$
- structure covariance

$$\boldsymbol{\Sigma}; \mathrm{Var}(\mathbf{s}_i) = \boldsymbol{\Sigma}_{ii} \text{ and } Cov(\mathbf{s}_i, \mathbf{s}_j) = \boldsymbol{\Sigma}_{ij}$$

Gaussian Field

12

A Gaussian field can be considered

- stationary if it has a constant mean

$$\mu(\mathbf{s}_1) = \mu(\mathbf{s}_2) = \ldots = \mu(\mathbf{s}_n)$$

- second-order stationary if it has constant variance dependent on distance and not location

$$Cov(\mathbf{s}_i, \mathbf{s}_j) = Cov(h(\mathbf{s}_i, \mathbf{s}_j))$$

- isotropic if variance is dependent only on distance and not direction

$$Cov(\mathbf{s}_i, \mathbf{s}_j) = Cov(\|h(\mathbf{s}_i, \mathbf{s}_j)\|)$$

13

Yield Maps

- We don't expect a yield map to be a Gaussian Field.
- We do want any errors in a yield map to be a Gaussian Field
- By analogy
  - One-way ANOVA
    - Data = Model + Error
  - Yield Map
    - Yield Monitor Data = Fertility Map + Gaussian Field

Schabenberger, O., & Gotway, C. A. (2005). Statistical Methods for Spatial Data Analysis. Chapman & Hall/CRC.

14

Autocorrelation

- Suppose we have a sequence of observations.
- We commonly assume these observations are i.i.d.
- If the observations are not independent, then each observation will have some relationship (dependence) on the preceding observation.

15

Sequential Processes

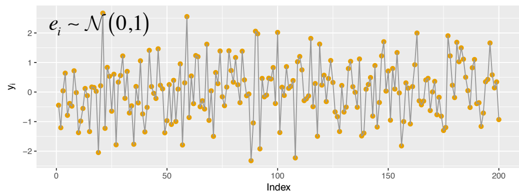
- White Noise
- Random Walk
- Autoregressive
- Moving Average
- Polynomial Trend

Borrowing largely from Cressie, N., & Wikle, C. K. (2011). Statistics for Spatio-Temporal Data. John Wiley & Sons.

16

White Noise

$$y_i = \mu + e_i$$

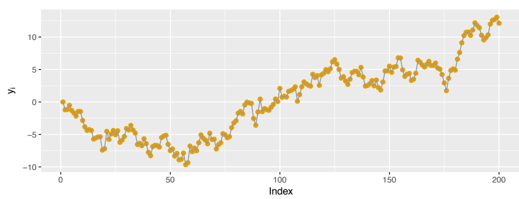


since e_i are *iid*, so are y_i

17

Random Walk

$$y_i = \mu + y_{i-1} + e_i$$

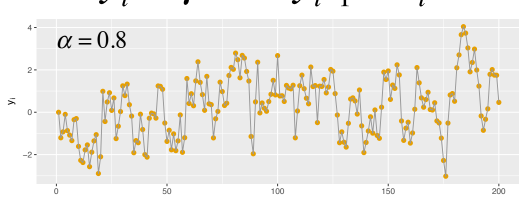


for simplicity, we let $\mu = 0$

18

Autoregressive

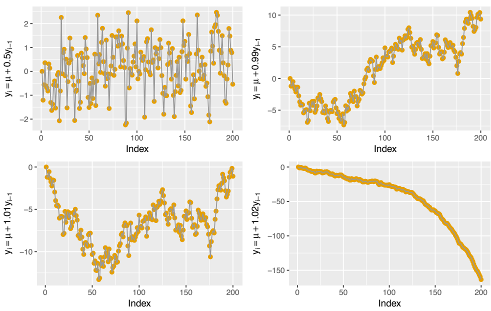
$$y_i = \mu + \alpha y_{i-1} + e_i$$



- white noise process when $\alpha = 0$
- random walk when $\alpha = 1$

19

Autoregressive

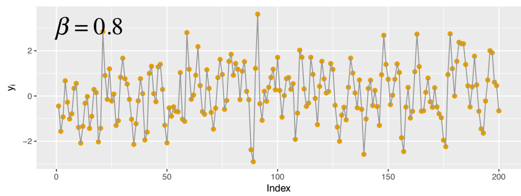


- explosive process when $|\alpha| > 1$

20

Moving Mean

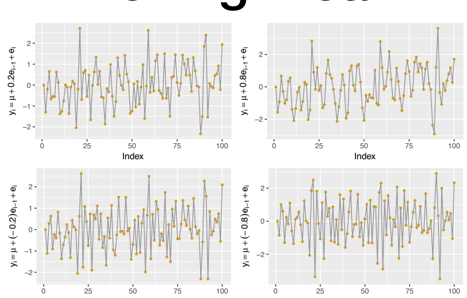
$$y_i = \mu + \beta e_{i-1} + e_i$$



- white noise process when $\beta = 0$
- a constant response to an unmeasurable random variable

21

Moving Mean

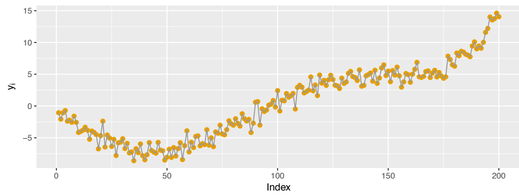


- more high-frequency behaviors $\beta < 0$
- more low-frequency behaviors $\beta > 0$

22

Trend + Error

$$y_i = \mu(i) + e_i$$



- non-stationary $\mu(i) \neq \mu(j), i \neq j$
- specifically $\mu(i) = poly(i,5) = \beta_0 + \beta_1 i + \ldots + \beta_6 i^5$

23

Model Order

- We can extend autocorrelated models to include any number of indexed values. The maximum number of indexed values gives the order of the model, i.e.

$$y_i = \mu + \alpha_1 y_{i-1} + \alpha_2 y_{i-2} + \ldots + e_i$$

- Further, we can write either AR(1) or MA(1) as a infinite series of the opposite process. That is, we can back-substitute from the formula above to produce

$$y_n = \sum_{k=0}^{\infty} \alpha^k e_{n-k}$$

24

ARIMA

- Thus, in many cases the simplest (i.e. most parsimonious) model includes both AR and MA components.

- We'll explore these processes more using the arima function in R. This function accepts a series of observations and a parameter specifying AR order, differencing (which we'll skip) and MA order.

```
arima(white.noise,c(1,0,1))
arima(random.walk,c(1,0,1))
arima(autoregressive,c(1,0,1))
arima(moving.average,c(1,0,1))
arima(trend.error,c(1,0,1))
```

25

ARIMA

$$y(1,0,1)_i = \mu + \alpha_1 y_{i-1} + \beta_1 e_{i-1} + e_i$$

$$y(2,0,2)_i = \mu + \alpha_1 y_{i-1} + \alpha_2 y_{i-2} + \beta_1 e_{i-1} + \beta_2 e_{i-2} + e_i$$

Series	Model	AIC	mean	ar1	ar2	ma1	ma2
White Noise	(1,0,1)	554.98	0.0694	0.9731		-1.0000	
	(2,0,2)	546.79	0.0603	0.2085	-0.9937	-0.1737	1.0000
Random Walk	(1,0,1)	560.40	3.1070	0.9892		-0.0103	
	(2,0,2)*	560.79	12.6650	1.7487	-0.7487	-0.8219	0.0240
Autoregressive	(1,0,1)	554.22	0.2924	0.6962		0.0747	
	(2,0,2)	557.56	0.2791	1.5940	-0.6086	-0.8290	-0.0972
Moving Average	(1,0,1)	556.97	0.1046	-0.0035		0.7840	
	(2,0,2)	555.78	0.1073	0.2590	-0.2111	0.5141	-0.1082

* In arima(random.walk, c(2, 0, 2)) :
possible convergence problem: optim gave code = 1

26

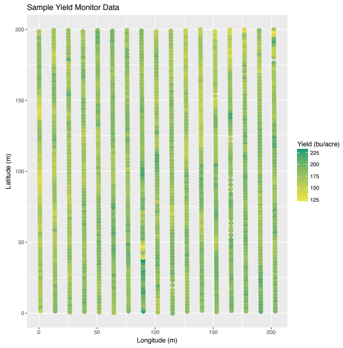
Yield Monitor Data

- Now we consider example yield monitor data.
- The sample data is a corn field, 2015, where I've trimmed the edges. I've also numbered the harvest strips as passes; we'll consider pass 14.

27

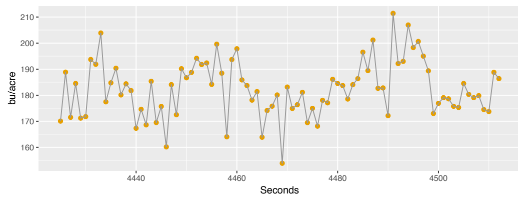
Sample Field

Longitude
Latitude
Distance (ft)
Yield (bu/ac)
YldMassWet (lb/ac)
Moisture (%)
Heading (degree)
LonM (m)
LatM (m)



28

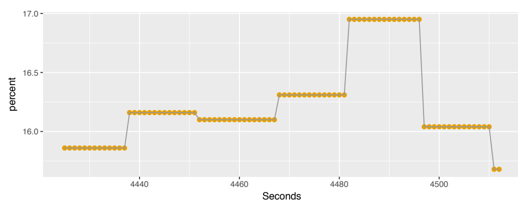
Yield



Model	AIC	mean	ar1	ar2	ar3	ma1	ma2	ma3
ARIMA(1,0,0)	660	182	0.32					
ARIMA(0,0,1)	663	182				0.20		
ARIMA(1,0,1)	**655**	**182**	**0.80**			**-0.53**		
ARIMA(2,0,2)	657	182	0.16	0.45		0.04	-0.16	
ARIMA(3,0,3)	656	182	-0.87	0.40	0.71	1.18	0.18	-0.49

29

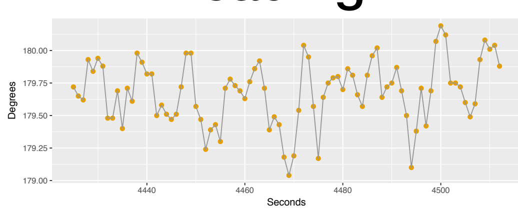
Moisture



Model	AIC	mean	ar1	ar2	ar3	ma1	ma2	ma3
ARIMA(1,0,0)	**-102**	**16.1**	**0.94**					
ARIMA(0,0,1)	-5.4	16.2				0.77		
ARIMA(1,0,1)	-100	16.1	0.93			0.03		
ARIMA(2,0,2)	-99.5	16.2	1.94	-0.94		-1.03	0.03	
ARIMA(3,0,3)	-96.4	16.1	-0.19	0.12	0.85	1.21	1.12	0.12

30

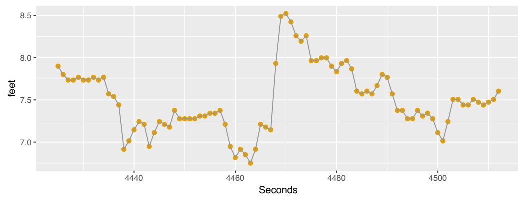
Heading



Model	AIC	mean	ar1	ar2	ar3	ma1	ma2	ma3
ARIMA(1,0,0)	-35.2	180	0.81					
ARIMA(0,0,1)	-34.2	180				0.62		
ARIMA(1,0,1)	**-39.1**	**180**	**0.42**			**0.35**		
ARIMA(2,0,2)	-38.1	180	0.51	-0.30		0.26	0.31	
ARIMA(3,0,3)	-35.7	180	0.38	0.61	-0.48	0.51	-0.53	-0.04

31

Distance



Model	AIC	mean	ar1	ar2	ar3	ma1	ma2	ma3
ARIMA(1,0,0)	-67.1	7.55	0.91					
ARIMA(0,0,1)	-9.9	7.50				0.92		
ARIMA(1,0,1)	-81.9	7.53	0.82			0.56		
ARIMA(2,0,2)	-80.0		0.15	0.59		1.21	0.29	
ARIMA(3,0,3)	-76.1		0.13	0.63	0.00	1.23	0.27	-0.03

32

Estimating Correlation

33

Simple Autoregression

$$\hat{\rho}_1 = r_1 = \frac{\sum_{2}^{n}(y_i - \bar{y})(y_{i-1} - \bar{y})}{\sum_{1}^{n}(y_i - \bar{y})^2}$$

- lag-1 autocorrelation coefficient
- compare this to Pearson's correlation coefficient between two variables *x* and *y*

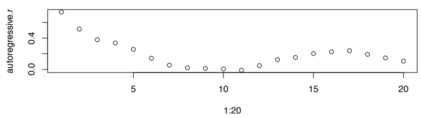
$$r = \frac{\sum_{1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{1}^{n}(y_i - \bar{y})^2}}$$

34

Lag classes

$$r_1, r_2, \ldots, r_k = \frac{\sum_{2}^{n}(y_i-\bar{y})(y_{i-1}-\bar{y})}{\sum_{1}^{n}(y_i-\bar{y})^2}, \frac{\sum_{3}^{n}(y_i-\bar{y})(y_{i-2}-\bar{y})}{\sum_{1}^{n}(y_i-\bar{y})^2}, \ldots, \frac{\sum_{k+1}^{n}(y_i-\bar{y})(y_{i-k}-\bar{y})}{\sum_{1}^{n}(y_i-\bar{y})^2}$$

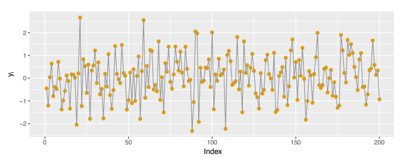
- We can compute correlation across different distances (lags) and plot the correlation by lag group (correlogram)

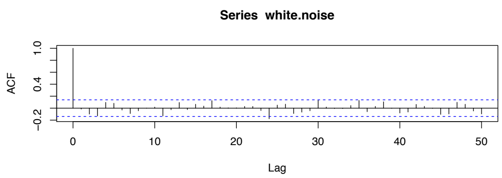


The `acf` function in R produces a nicer plot; we'll continue with that

35

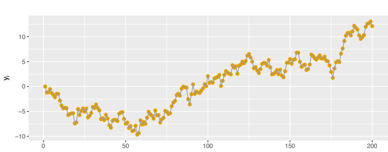
White Noise

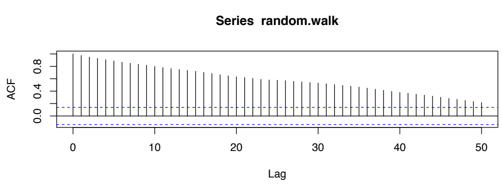




36

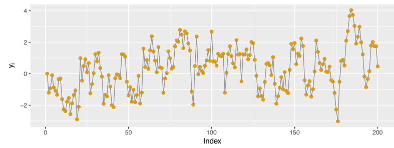
Random Walk

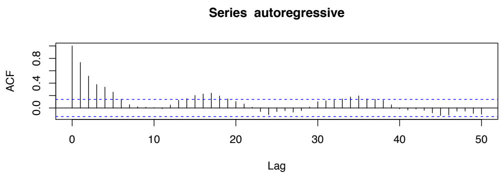




37

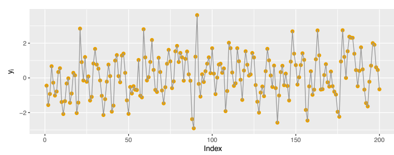
Autoregressive

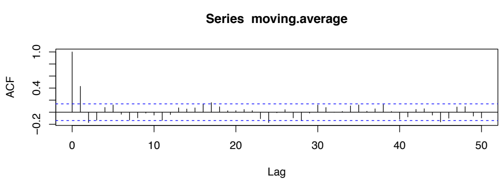




38

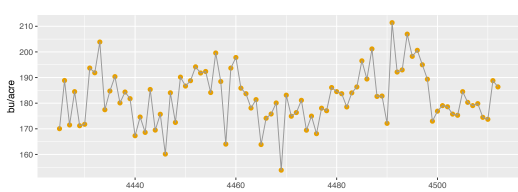
Moving Mean

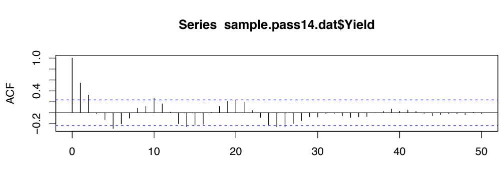




39

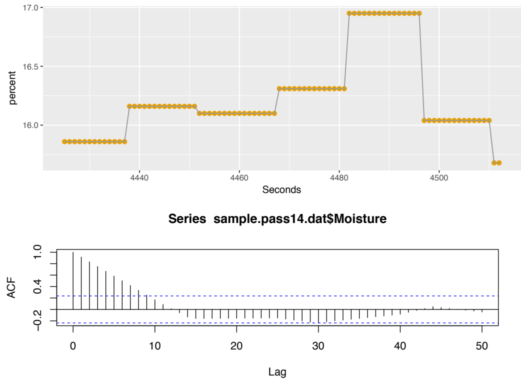
Yield





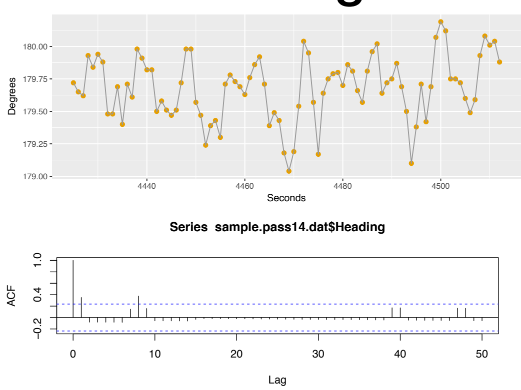
40

Moisture



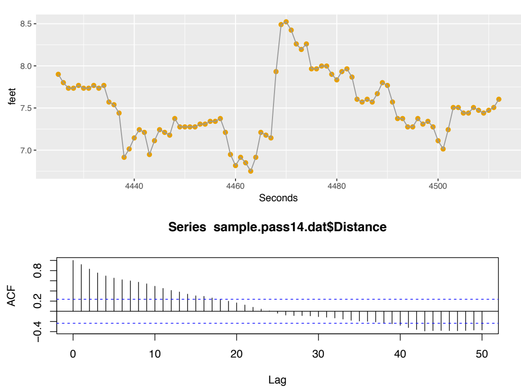
41

Heading



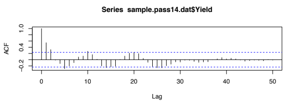
42

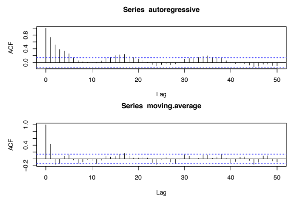
Distance



43

Yield

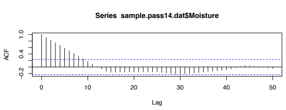


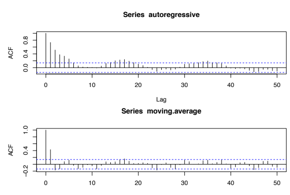


Model	AIC	mean	ar1	ar2	ar3	ma1	ma2	ma3
ARIMA(1,0,0)	660	182	0.32					
ARIMA(0,0,1)	663	182				0.20		
ARIMA(1,0,1)	**655**	**182**	**0.80**			**-0.53**		
ARIMA(2,0,2)	657	182	0.16	0.45		0.04	-0.16	
ARIMA(3,0,3)	656	182	-0.87	0.40	0.71	1.18	0.18	-0.49

44

Moisture

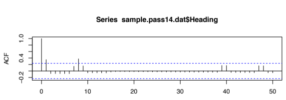


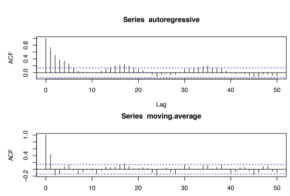


Model	AIC	mean	ar1	ar2	ar3	ma1	ma2	ma3
ARIMA(1,0,0)	**-102**	**16.1**	**0.94**					
ARIMA(0,0,1)	-5.4	16.2				0.77		
ARIMA(1,0,1)	-100	16.1	0.93			0.03		
ARIMA(2,0,2)	-99.5	16.2	1.94	-0.94		-1.03	0.03	
ARIMA(3,0,3)	-96.4	16.1	-0.19	0.12	0.85	1.21	1.12	0.12

45

Heading

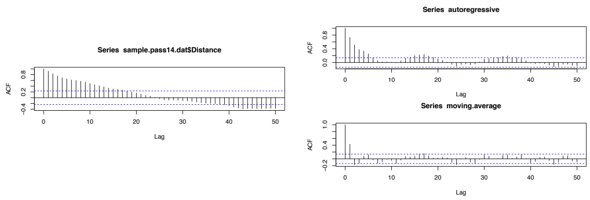




Model	AIC	mean	ar1	ar2	ar3	ma1	ma2	ma3
ARIMA(1,0,0)	-35.2	180	0.81					
ARIMA(0,0,1)	-34.2	180				0.62		
ARIMA(1,0,1)	**-39.1**	**180**	**0.42**			**0.35**		
ARIMA(2,0,2)	-38.1	180	0.51	-0.30		0.26	0.31	
ARIMA(3,0,3)	-35.7	180	0.38	0.61	-0.48	0.51	-0.53	-0.04

46

Distance



Model	AIC	mean	ar1	ar2	ar3	ma1	ma2	ma3
ARIMA(1,0,0)	-67.1	7.55	0.91					
ARIMA(0,0,1)	-9.9	7.50				0.92		
ARIMA(1,0,1)	-81.9	7.53	0.82			0.56		
ARIMA(2,0,2)	-80.0		0.15	0.59		1.21	0.29	
ARIMA(3,0,3)	-76.1		0.13	0.63	0.00	1.23	0.27	-0.03

47

(Semi)Variograms

48

Theoretical Variogram

- We can relate variance to distance by the (semi)variogram

$$\gamma(h) = \frac{1}{2} Var\left[y(\mathbf{s}_i) - y(\mathbf{s}_i + h) \right]$$

- where h is a distance and may be scalar or vector. Scalar implies that the variogram is the same at all points on a ring around **s**.

- This function is dependent only in distance h and not on location, thus is not valid if **s** are not stationary.

- Strictly speaking, a semi-variogram is half a variogram, but the two terms are often used inter-changeably.

Bachmaier, M., & Backes, M. (2008). Variogram or semivariogram? Understanding the variances in a variogram. Precision Agriculture, 9(3), 173–175.

49

Variance Estimate

Consider that for univariate data, with

$$y_i = \mu + e_i \qquad e \sim \mathcal{N}\left(0, \sigma^2\right)$$

we can estimate variance by

$$\hat{\sigma}^2 = s^2 = \frac{\sum_1^n \left(y_i - \bar{y}\right)^2}{n-1}$$

We can also estimate variance, independent of the mean, by

$$\hat{\sigma}^2 = s^2 = \frac{1}{2} \frac{\sum_{i \neq j}^n \left(y_i - y_j\right)^2}{n\left(n-1\right)}$$

50

Empirical Variogram

Similarly, we can compute an empirical variorum by computing variances for lag classes, denoted by **h**

$$g(\mathbf{h}) = \frac{1}{2} \frac{1}{n(\mathbf{h})} \Sigma_{i=1}^{n(\mathbf{h})} [y(\mathbf{s}) - y(\mathbf{s} + \mathbf{h})]^2$$

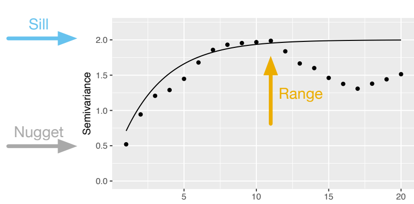
In this case, **h** represents a range of distances, i.e. $0 \leq h_1 < 5$, $5 \leq h_2 < 10$, …, and n(**h**) are the number of pairs of points that lie within that distance of each other.

We then plot *g* versus *h.*

51

Components of a Variogram

- To make use of a variogram for operations such as kriging, we need to find a measure of variance for arbitrary distances. We do this by fitting a smoothing function to the empirical variogram.
- There are typically three components to the smoothing function:



52

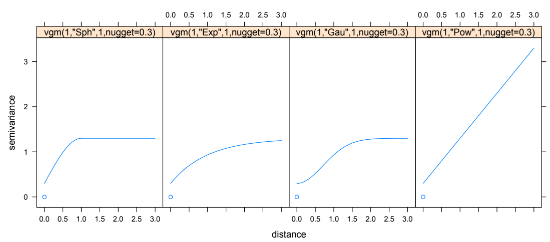
Variogram Models

- Nugget $$g(h)=\begin{cases} 0 & h=0 \\ c & otherwise \end{cases}$$
- Spherical $$g(h)=\begin{cases} c\times\left[1.5\left(\frac{h}{a}\right)-0.5\left(\frac{h}{a}\right)^3\right] & h\le a \\ c & otherwise \end{cases}$$
- Exponential $$g(h)=c\times\left[1-\exp\left(\frac{-3h}{a}\right)\right]$$
- Gaussian $$g(h)=c\times\left[1-\exp\left(\frac{-3h^2}{a^2}\right)\right]$$
- Power $$g(h)=c\times h^{\omega}, 0<\omega<2$$

53

Variogram Models

```
show.vgms(models = c("Sph", "Exp", "Gau", "Pow"), nugget = 0.3)
```



from `library(gstat)`

54

gstat variogram

- There are several S3 methods associated with the variogram function in gstat. The version I find simplest is the ‘formula’ method
- `variogram(object, locations = coordinates(data), data, …)`
- To produce a variogram for our yield data, we use the call

```
> Yield.var <- variogram(Yield~1,
                         locations=~LonM+LatM,
                         data=sample.dat)
> head(Yield.var)
    np      dist     gamma dir.hor dir.ver   id
1  3054  3.486770  83.84119       0       0 var1
2  5465  9.947748 134.04349       0       0 var1
…
```

gstat variogram

```
> Yield.var <- variogram(Yield~1,
                         locations=~LonM+LatM,
                         data=sample.dat)
```

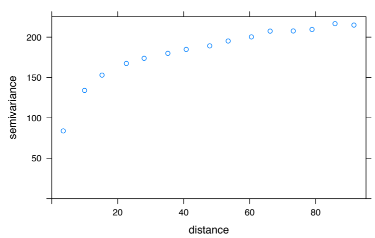
- The formula `Yield~1` implies a stationary (constant mean) model
- Locations may also be specified by assigning an attribute to data, but this is usually associated with other spatial packages (i.e. `sp`)

```
> library(sp)
> coordinates(sample.dat) <- ~LonM+LatM
> Yield.var <- variogram(Yield~1, data=sample.dat)
```

55

plot variogram

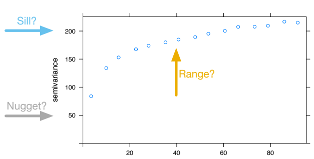
```
> plot(Yield.var)
```



56

Fitting an Empirical Variogram

- Most variogram models are non-linear, so there is no simple least-squares solution. We usually require initial guesses. These are specified as a *vgm* object.
  - `vgm(psill=150,model="Sph",range=40,nugget=50)`
- `psill` is a partial sill, given by the difference between the sill and the nugget



57

58

Fitting an Empirical Variogram

- Sometimes fitting is easy

```
> fit.variogram(Yield.var, vgm(psill=200, model="Sph",
range=40, nugget=50))
  model     psill    range
1   Nug  64.12461  0.00000
2   Sph 126.97065 30.91816
```

59

Fitting an Empirical Variogram

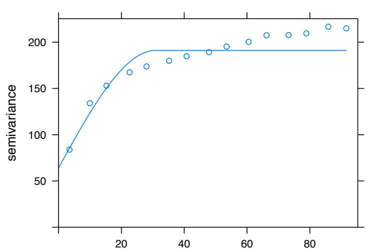
- Sometimes not

```
> Heading.var <- variogram(Heading~1, data=sample.dat)
> fit.variogram(Heading.var, vgm(0.25,"Sph",20,0.1))
In fit.variogram(Heading.var, vgm(0.25, "Sph", 20, 0.1)) :
No convergence after 200 iterations: try different initial
values?
> Heading.vgm <- fit.variogram(Heading.var, vgm(1000,"Sph",
20,500))
  model    psill    range
1   Nug 1959.372 0.000000
2   Sph 9752.226 9.370907
Warning message:
In fit.variogram(Heading.var, vgm(1000, "Sph", 20, 500)) :
  singular model in variogram fit
```

60

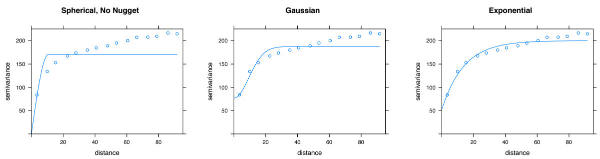
Plotting a Fitted Variogram

```
> plot(Yield.var, Yield.vgm)
```



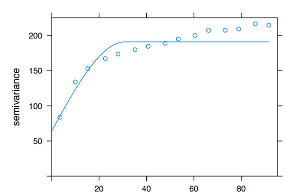
61

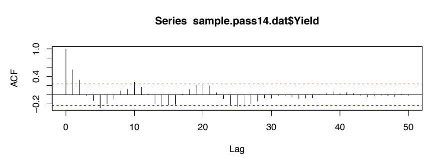
Models, Yield



62

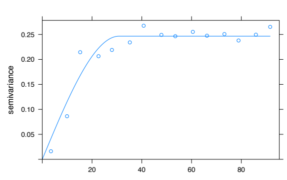
Yield

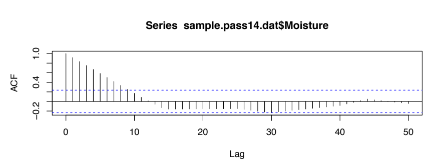




63

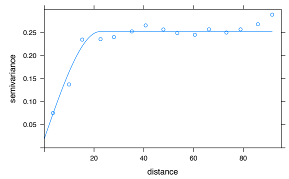
Moisture

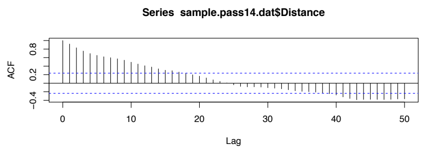




64

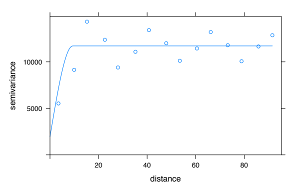
Distance

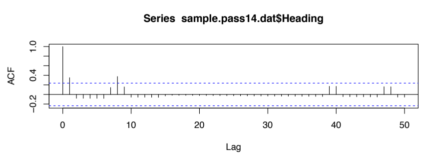




65

Heading





66

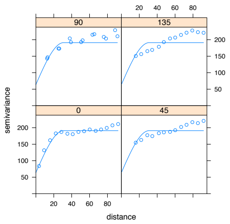
Anisotropy

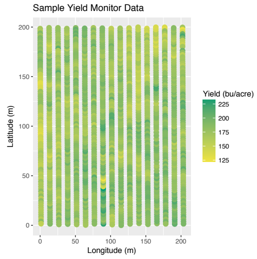
- A Gaussian field is isotropic only if covariance depends on distance and not on direction.
- We can check for this by computing variograms at different angles.

```
> Yield.ani.var <- variogram(Yield~1,
          locations=~LonM+LatM,
          data=sample.dat,
          alpha=c(0,45,90,135))
> plot(Yield.ani.var,model=Yield.vgm)
```

67

Yield

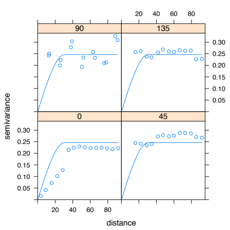


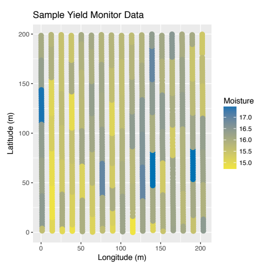


`gstat` doesn't automatically fit anisotropy, so we would need to eyeball an estimate.

68

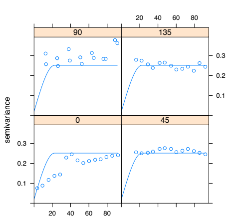
Moisture

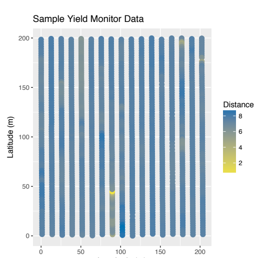




69

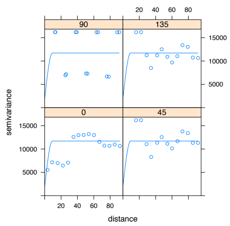
Distance

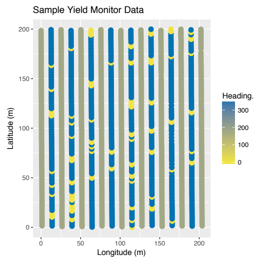




70

Heading





71

Kriging

72

Kriging Equations

- Suppose we wish to estimate a value for an unobserved point **s′**. We can do this by assigning weights and summing over observed points by

$$\hat{y}(\mathbf{s}') = \Sigma_{i=1}^{n} w_i y(\mathbf{s}_i)$$

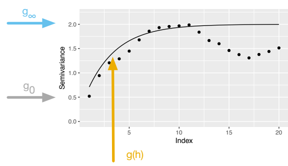
- We choose ***w*** such that the sum of all ***w*** is 1; this produces a weighted average over all observed points

73

Kriging Weights

- We estimate a (semi)covariance at distance h using the (semi)variogram model

$$\hat{C}(h') = g_{\infty} - g_0 - g(h')$$



Adapted from Plant, R. E. (2012). Spatial Data Analysis in Ecology and Agriculture Using R. CRC Press.

74

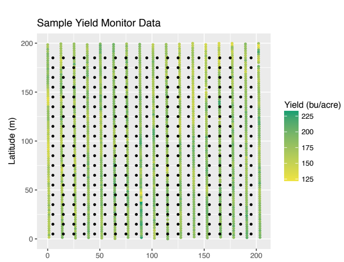
Kriging Computations

- For each pair of points (**s**′, $\mathbf{s}_i$) we can assign a w_i proportional to *C(h)*, where *h* is the distance between **s**′and $\mathbf{s}_i$
- This weight has to be computed for each new point **s**′ and typically involves computationally expensive matrix inversions.

75

New Points

- It's not very interesting to krige a single new point.
- To illustrate kriging, we'll use our sample trial map to estimate yields on a uniform grid.
- Here, we have points on a 20x20m grid, starting at (10,10)



76

kriging in gstat

```
> Yield.krig <- krige(id="Yield",
                      formula=Yield~1,
                      data = sample.dat,
                      newdata = sample4.grd,
                      model = Yield.vgm,
                      maxdist = 100,
                      locations=~LatM + LonM)
```

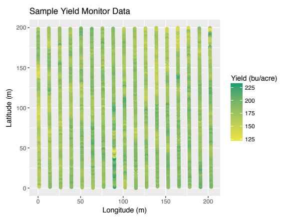
- The syntax for kriging is similar to the syntax for fitting a variogram.
- Note that we add the fitted variogram along with original and new data.
- Strictly speaking, ordinary kriging would use all original data points (as opposed to local kriging, which limits the range), but I've included a `maxdist` to save computing time.

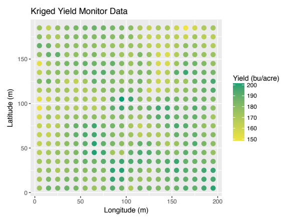
77

krige results

- We can extract the predicted values and plot

```
sample4.grd$Yield.krig <- Yield.krig$Yield.pred
```





78

Spatial Correlation

79

Everything depends on everything else, but closer things more so

–Tobler's first law of geography

80

Moran I

- We write Moran's I as

$$I = \frac{n}{\Sigma_i \Sigma_j w_{ij}} \frac{\Sigma_i \Sigma_j w_{ij}(y_i - \overline{y})(y_j - \overline{y})}{\Sigma_i (y_i - \overline{y})^2}$$

- where we simplify by denoting

$$y_i = y(\mathbf{s}_i)$$

81

Neighbor Weights

- We compute I using a matrix of weights W associated with each pairwise distance, such that

$$w_{ii} = 0$$

- Sometimes W is binary,

$$w_{ij} = \begin{cases} 1 & i,j \text{ are neighbors} \\ 0 & \text{otherwise} \end{cases}$$

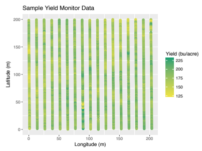
- We'll use a weight determined by the distance between points

$$w_{ij} = 1/h(\mathbf{s}_i, \mathbf{s}_j)$$

82

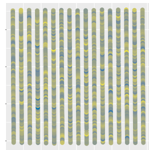
Examples

Yield

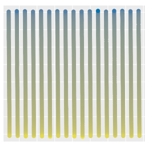


Using points from yield data, we create different patterns

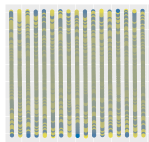
White Noise



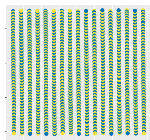
Ordered



Disordered



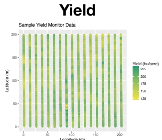
Checkerboard



83

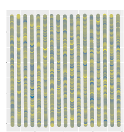
Moran I

Yield

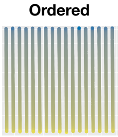


Pattern	I
Ordered	0.3659194
Yield	0.1027251
Expected	-0.00066844
Disordered	-0.00184221
White Noise	-0.00224158
Checkerboard	-0.02915056

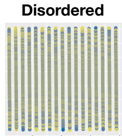
White Noise



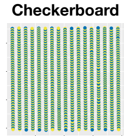
Ordered



Disordered



Checkerboard



84

Other Measures

- Geary C

$$C = \frac{N-1}{\Sigma_i 2\Sigma_j w_{ij}} \frac{\Sigma_i \Sigma_j w_{ij}(y_i - y_j)^2}{\Sigma_i (y_i - \bar{y})^2}$$

- Getis-Ord G

$$G = \frac{\Sigma_i \Sigma_j w_{ij} \times y_i \times y_j}{\Sigma_i \Sigma_j y_i \times y_j}$$

$$I = \frac{N}{\Sigma_i \Sigma_j w_{ij}} \frac{\Sigma_i \Sigma_j w_{ij}(y_i - \bar{y})(y_j - \bar{y})}{\Sigma_i (y_i - \bar{y})^2}$$

85

Comparison : *I*, *C*, *G*

Pattern	I	p(I)	C	p(C)	G	p(G)
Ordered	0.3324	<0.001	0.5467	<0.001	0.0602	<0.001
Yield	0.0979	<0.001	0.8869	<0.001	0.0603	<0.001
Expected	-0.0007		1.00		0.0599	
Disordered	-0.0016	0.4914	0.8806	<0.001	0.0600	0.06143
White Noise	-0.0021	0.3293	1.0038	0.4773	0.0599	0.7708
Checkerboard	-0.0288	<0.001	1.0281	<0.001	0.0599	0.985

Statistics computed using `spdep` functions, "two-sided", under randomization

86

LISA

- Local Indicators of Spatial Association
- Two key properties
  - LISA for each observation is an indicator of similar values around that observation
  - The sum of all LISA is proportional to a global measure of spatial correlation

Anselin, L. (1995). Local Indicators of Spatial Association - LISA. Geographical Analysis, 27.

87

Local I

- We write a local Moran I by

$$I_i = \frac{n \times (y_i - \bar{y})}{\Sigma_i (y_i - \bar{y})^2} \Sigma_j w_{ij} (y_j - \bar{y})$$

- It follows that

$$I = \Sigma_i \frac{I_i}{N}$$

88

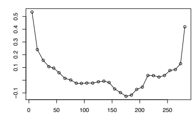
LISA plots

- Our sample yield data has a very large number of points; too many to examine local correlation, individually.
- We can visualize degree of local spatial correlation using LISA plots (from the package ncf).
- This package produces a variation on a bubble plot:
  - Size of the symbol is determine by the magnitude of the difference between the point value and a grand mean
  - shape and color are determined by sign of the difference - red circles are positive and black squares are negative.
  - Shapes are filled if the point has significant local correlation

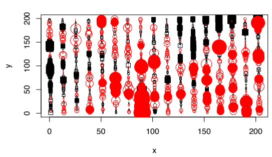
89

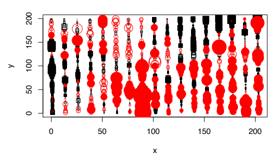
Neighborhood

- The correlogram for Yield suggests an effective range of 60m, but correlation is small at 20m.



- It takes longer to compute more neighbors, so we compare 6 and 60

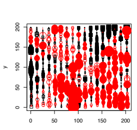




90

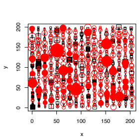
Examples

Yield

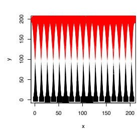


neighborhood=60

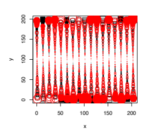
White Noise



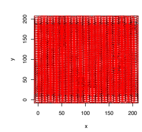
Ordered



Disordered



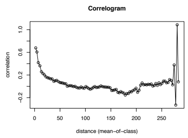
Checkerboard



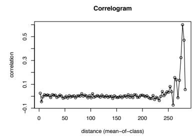
91

Correlograms

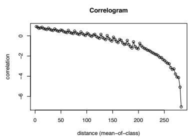
Yield



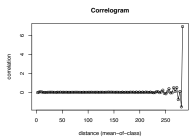
White Noise



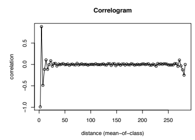
Ordered



Disordered

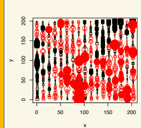
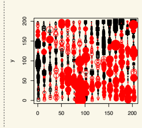
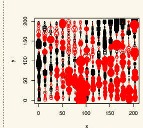
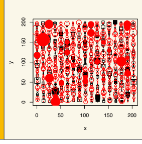
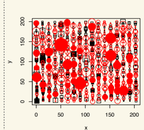
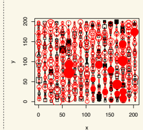


Checkerboard



92

White Noise and Neighborhoods

Pattern	6	30	60
Yield			
White Noise			

93

Trend Analysis

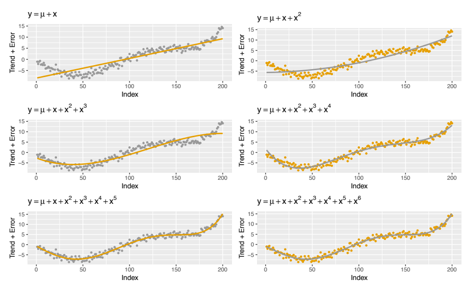
Polynomial Regression

- Polynomial functions are frequently used to fit arbitrary curves to data.
- Increasing the degree of a polynomial allows more points to be fitted.
- A polynomial of degree n can be fit exactly to (n+1) data points.
- This will be commonly done if the data are not stationary

```
> arima(trend.error,c(1,0,1))
Error in arima(trend.error, c(1, 0, 1)) : non-stationary AR part
from CSS
```

94

Polynomial Regression




```
lm(trend.error ~ 1 + x + I(x^2) + I(x^3) + I(x^4) + I(x^5))
lm(trend.error ~ poly(x,5))
```

95

Detrending

- We wish to detrend a series by fitting to an appropriate polynomial model, leaving only white noise, i.e.

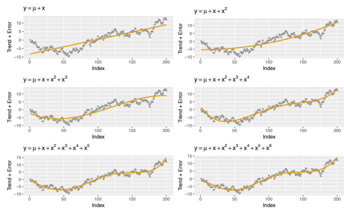
```
x <- 1:length(trend.error)
arima(lm(trend.error ~ poly(x,1))$residuals, c(1,0,0))
```

Model	AIC	mean	ar1	s^2
poly(x,1)	687.95	0.3591	0.8636	1.759
poly(x,2)	678.24	0.1242	0.7971	1.679
poly(x,3)	670.1	0.0828	0.7205	1.614
poly(x,4)	612.3	-0.0029	0.3055	1.213
poly(x,5)	548.87	0.0003	-0.0416	0.8838
poly(x,6)	547.68	0.0002	-0.0456	0.8785
poly(x,7)	547.67	0.0002	-0.0455	0.8785

`trend.error` was simulated by fitting a 5th degree polynomial to `random.walk` and adding random error with `mean=0` and `sd=1`

96

Random Walk



Model	AIC	mean	ar1	s^2
poly(x,1)	553.82	0.9972	0.9554	0.895
poly(x,2)	548.38	0.3102	0.9265	0.8731
poly(x,3)	543.93	0.2257	0.8981	0.8552
poly(x,4)	532.11	-0.0468	0.8128	0.8084
poly(x,5)	528.68	-0.0356	0.7666	0.7954
poly(x,6)	526.22	-0.0039	0.7536	0.7859
poly(x,7)	519.85	-0.0034	0.7255	0.7616

97

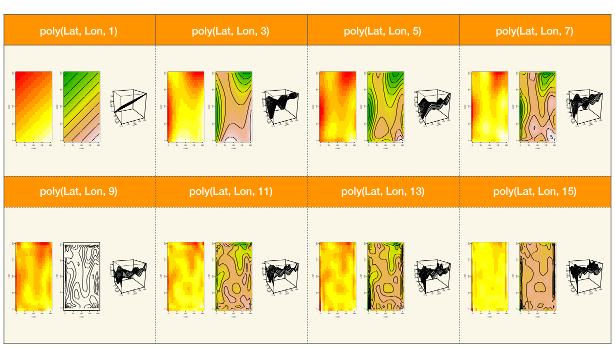
Trend Surface

- We can extend this concept to two-dimensions by fitting to a polynomial in two variables.
- Generically, this can be termed a response surface (think of response to different levels of two factors), while trend surface is more common in the geostatistical literature.
- We'll be using the package `rsm` to visualize polynomial trends.
- The general call will be

```
> Yieldn.lm <- lm(Yield ~ poly(LonM, LatM, degree=n),
data=sample.dat)
> image(Yieldn.lm, LatM ~ LonM)
> contour(Yieldn.lm,LatM ~ LonM, image = TRUE)
> persp(Yieldn.lm, LatM ~ LonM, zlab = "Yield, Poly n")
```

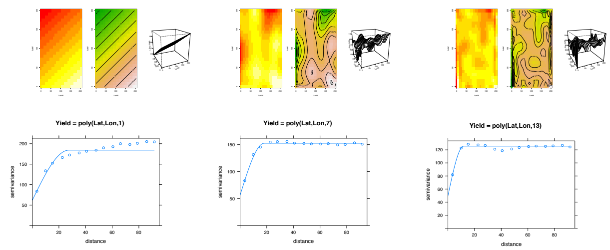
98

Trend Surface Fit



99

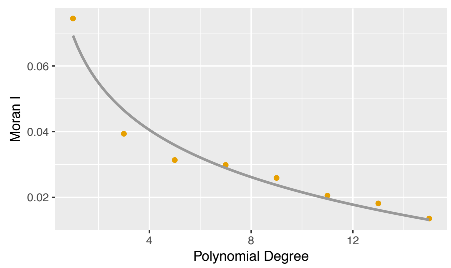
Residual Variograms



Model	nugget	partial sill	range
Yield = poly(x,1)	61.46898	122.74176	27.4485
Yield = poly(x,7)	56.19666	96.42960	18.20175
Yield = poly(x,13)	49.29429	76.39657	11.85668

100

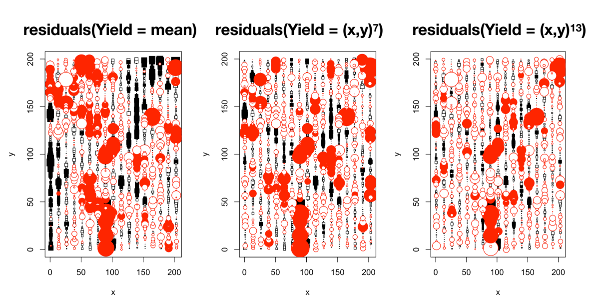
Residual Moran *I*



expected = -0.0006684492

101

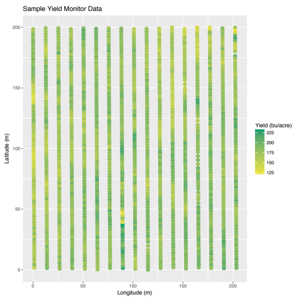
LISA - polynomial residuals

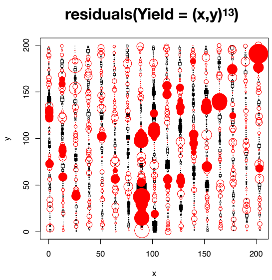


102

103

LISA - outliers





104

Examples

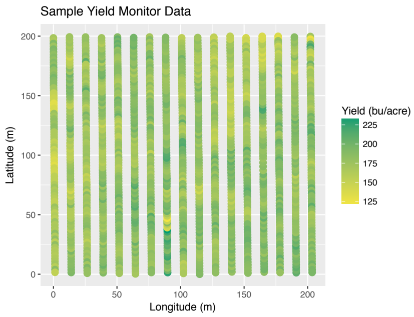
105

Grid Cells

- Our goal is to map randomly sample yield monitor data to a uniformly sampled grid.
- Our grid will be a 20x20m lattice. We divide a field into squares of 20 meters per side.
- We want an estimate of yield per grid cell.
- We will compare three methods
  - Grid cell means
    - We compute a simple arithmetic average over all yield points that fall within the bounds of the cell
  - Trend estimated means
    - We use a linear polynomial trend to interpolate yield at four uniformly selected points within each grid cell and compute the average of these four samples
  - Kriged means
    - We use a kriging to interpolate yield at four uniformly selected points within each grid cell and compute the average of these four samples

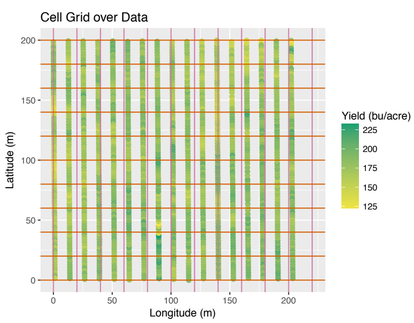
106

Yield Monitor Samples



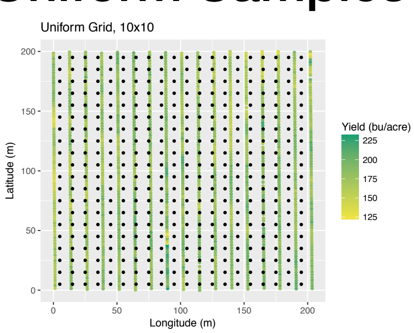
107

Grid Boundaries



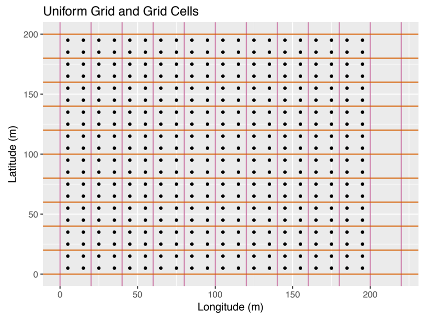
108

Uniform Samples



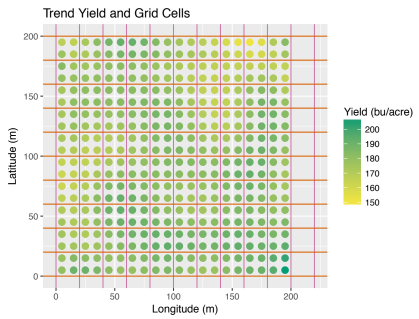
109

Uniform Samples



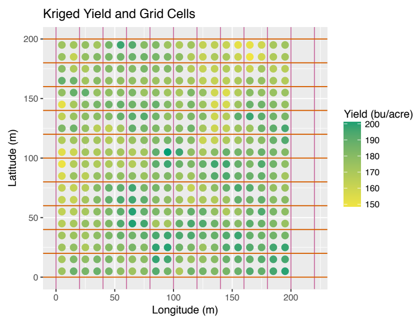
110

Trend Estimates



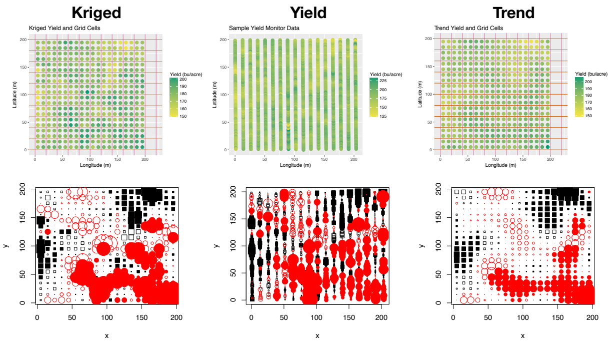
111

Kriged Estimates



112

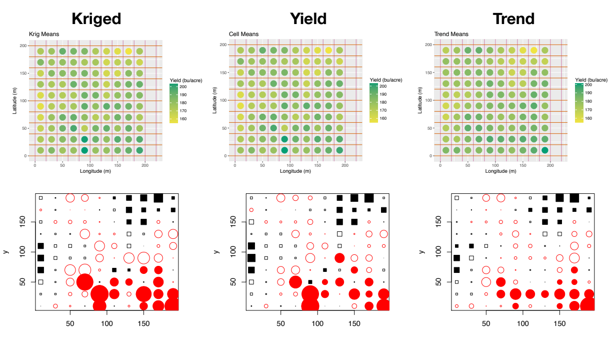
Comparing Interpolation



I = 0.1342357 I = 0.1027251 I = 0.1856037

113

Comparing Cell Means

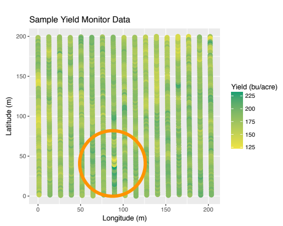


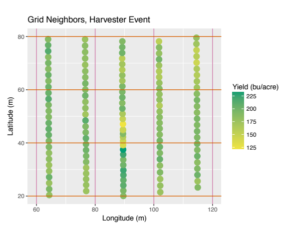
I = 0.1195487 I = 0.09819354 I = 0.1430269

114

Harvester Event

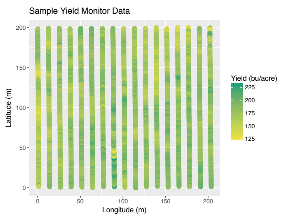
- There is a series of about 20 samples that suggest some sampling error due to a harvester event. We'll look at this in more detail

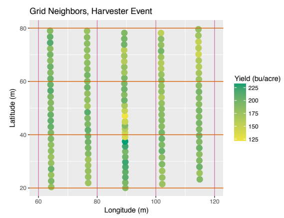




Harvester Event

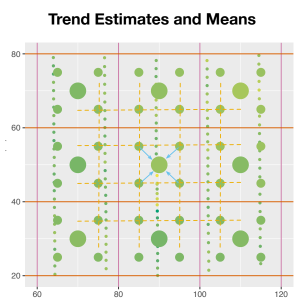
- There is a series of about 20 samples that suggest some sampling error due to a harvester event. We'll look at this in more detail

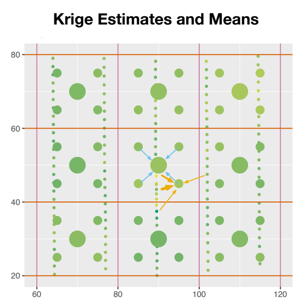




115

Harvester Event





116

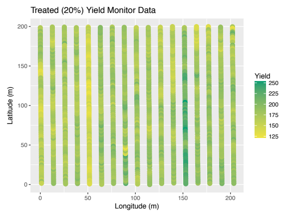
Detecting Strips

- We simulate a strip trial by adding or subtracting a constant to all yield samples in a single pass.
- We will use local indicators of spatial correlation to try to detect the treated strips.
- We will compare yield values and detrended residuals, and different values for simulated treatment effect.

117

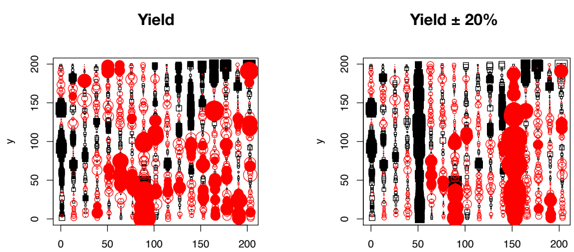
118

Yield ± 20%



119

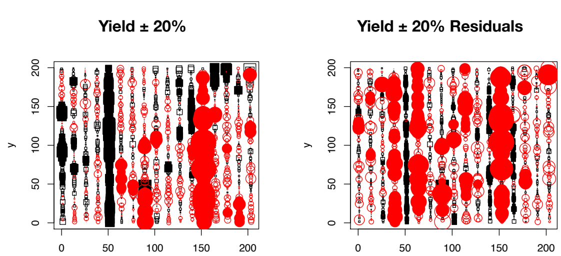
LISA, Yield



neighborhood = 10m

120

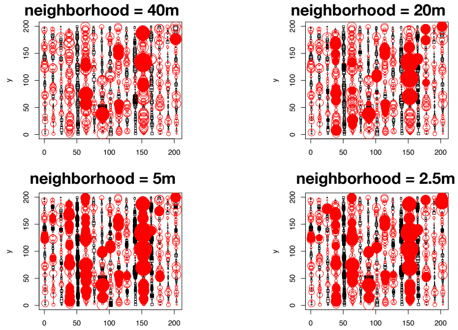
LISA, Yield



neighborhood = 10m

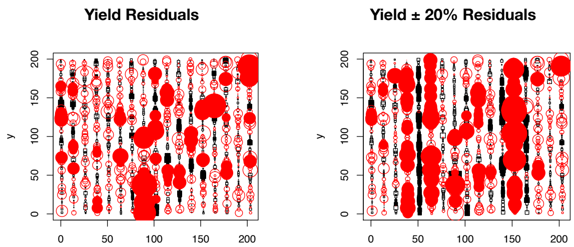
121

LISA, Yield ± 20% Residuals



122

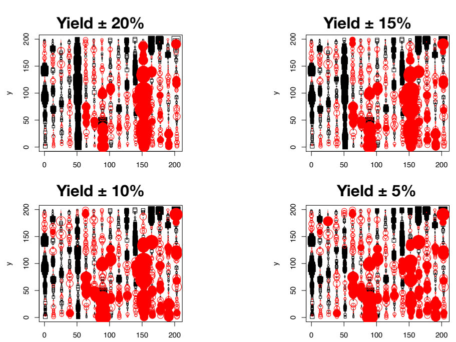
LISA, Yield Residuals



neighborhood = 10m

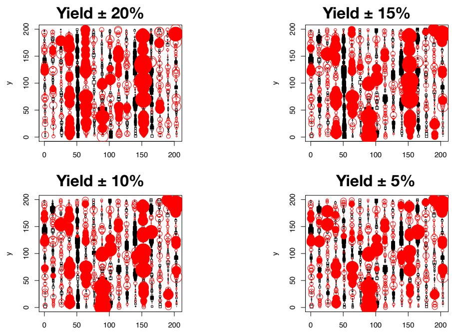
123

LISA, Yield



neighborhood = 10m

LISA, Yield Residuals



neighborhood = 10m