Multivariate spatial models and the `multiKrig` class

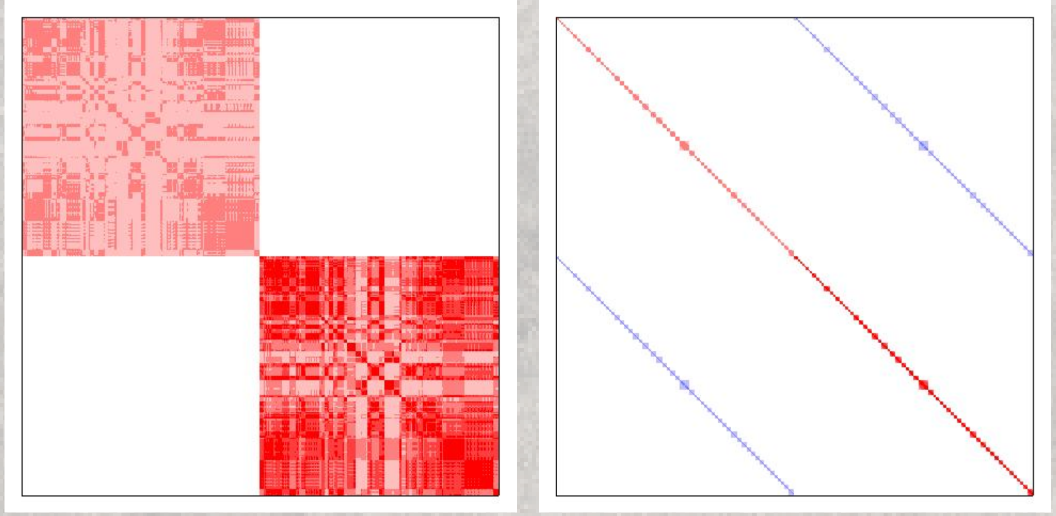
Stephan R. Sain, IMAGe, NCAR

SAMSI Summer School on Spatial Statistics

July 28–August 1, 2009

Outline

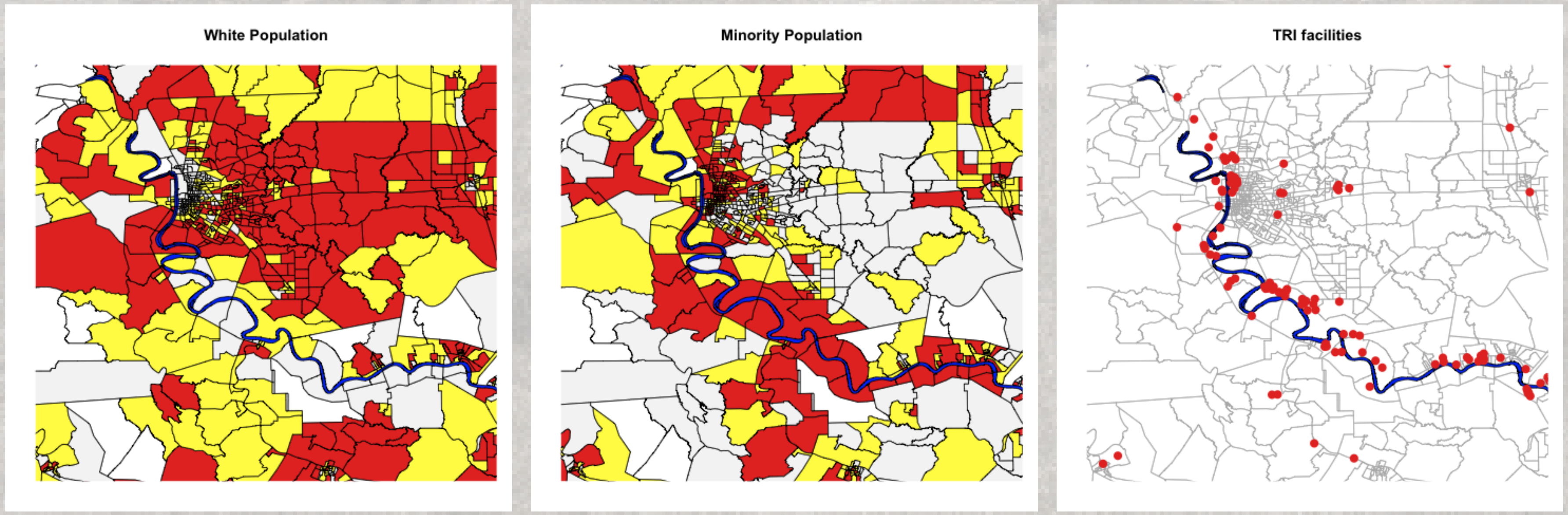
- Overview of multivariate spatial regression models.
- The `multiKrig` class
  - Case study: NC temperature and precipitation.
- Some future directions and issues.



Multivariate Spatial Regression

- Motivation: increased interest in scientific problems with many different spatial layers.
  - Geographic information systems (GIS).

- Examples: disease mapping, remote sensing, climate and weather,. . .



A Spatial Regression Model

- A spatial regression model:

$$\underset{(n\times 1)}{\mathbf{Y}} = \underset{(n\times q)(q\times 1)}{\mathbf{X}\boldsymbol{\beta}} + \underset{(n\times 1)}{\mathbf{h}} + \underset{(n\times 1)}{\boldsymbol{\epsilon}}$$

  where

  – $\text{E}[\mathbf{h}] = \mathbf{0}$, $\text{Var}[\mathbf{h}] = \boldsymbol{\Sigma}_{\mathbf{h}}$

  – $\text{E}[\boldsymbol{\epsilon}] = \mathbf{0}$, $\text{Var}[\boldsymbol{\epsilon}] = \sigma^2\mathbf{I}$.

  – $\mathbf{h}$ and $\boldsymbol{\epsilon}$ are independent.

- $\mathbf{Y} \sim \mathcal{N}(\mathbf{X}\boldsymbol{\beta}, \mathbf{V})$, $\mathbf{V} = \boldsymbol{\Sigma}_{\mathbf{h}} + \sigma^2\mathbf{I}$

- $\widehat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{V}^{-1}\mathbf{X})^{-1}\mathbf{X}'\mathbf{V}^{-1}\mathbf{Y}$, $\hat{\mathbf{h}} = \boldsymbol{\Sigma}_{\mathbf{h}}\mathbf{V}^{-1}(\mathbf{Y} - \mathbf{X}\widehat{\boldsymbol{\beta}})$

Multivariate Regression

- A multivariate, multiple regression model:

$$\underset{(n \times p)}{\mathbf{Y}} = \underset{(n \times q)(q \times p)}{\mathbf{X}\boldsymbol{\beta}} + \underset{(n \times p)}{\boldsymbol{\epsilon}}$$

  where

  – Each of the n rows of $\mathbf{Y}$ represents a p-vector observation.

  – Each of the p columns of $\boldsymbol{\beta}$ represent regression coefficients for each variable.

  – The rows of $\boldsymbol{\epsilon}$ represents a collection of iid error vectors with zero mean and common covariance matrix, $\boldsymbol{\Sigma}$.

Multivariate Regression

- MLEs are straightforward to obtain:

$$\underset{(q \times p)}{\widehat{\boldsymbol{\beta}}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$$
$$\underset{(p \times p)}{\widehat{\boldsymbol{\Sigma}}} = \frac{1}{n}\mathbf{Y}'\mathbf{P}\mathbf{Y}$$

  where $\mathbf{P} = \mathbf{I} - \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$.

- Note that the columns of $\widehat{\boldsymbol{\beta}}$ can be obtained through p univariate regressions.

Vec and Kronecker

- The Kronecker product of an $m \times n$ matrix $\mathbf{A}$ and an $r \times q$ matrix $\mathbf{B}$ is an $mr \times nq$ matrix:

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \cdots & a_{2n}\mathbf{B} \\ \vdots & & \ddots & \vdots \\ a_{m1}\mathbf{B} & a_{m2}\mathbf{B} & \cdots & a_{mn}\mathbf{B} \end{bmatrix}$$

- Some properties:

$$\begin{aligned} \mathbf{A} \otimes (\mathbf{B} + \mathbf{C}) &= \mathbf{A} \otimes \mathbf{B} + \mathbf{A} \otimes \mathbf{C} \\ \mathbf{A} \otimes (\mathbf{B} \otimes \mathbf{C}) &= (\mathbf{A} \otimes \mathbf{B}) \otimes \mathbf{C} \\ (\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) &= \mathbf{AC} \otimes \mathbf{BD} \\ (\mathbf{A} \otimes \mathbf{B})' &= \mathbf{A}' \otimes \mathbf{B}' \\ (\mathbf{A} \otimes \mathbf{B})^{-1} &= \mathbf{A}^{-1} \otimes \mathbf{B}^{-1} \\ |\mathbf{A} \otimes \mathbf{B}| &= |\mathbf{A}|^m |\mathbf{B}|^n \end{aligned}$$

Vec and Kronecker

- The vec-operator stacks the columns of a matrix:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \qquad \text{vec}(\mathbf{A}) = \begin{bmatrix} a_{11} \\ a_{21} \\ a_{12} \\ a_{22} \end{bmatrix}$$

- Some properties:

$$\begin{aligned} \text{vec}(\mathbf{AXB}) &= (\mathbf{B}' \otimes \mathbf{A})\, \text{vec}\, \mathbf{X} \\ \text{tr}(\mathbf{A}'\mathbf{B}) &= \text{vec}(\mathbf{A})' \text{vec}(\mathbf{B}) \\ \text{vec}(\mathbf{A} + \mathbf{B}) &= \text{vec}(\mathbf{A}) + \text{vec}(\mathbf{B}) \\ \text{vec}(\alpha \mathbf{A}) &= \alpha\, \text{vec}(\mathbf{A}) \end{aligned}$$

Multivariate Regression Revisited

- Rewrite the multivariate, multiple regression model:

$$\begin{array}{ccccc} \text{vec}(\mathbf{Y}) & = & (\mathbf{I}_p \otimes \mathbf{X})\,\text{vec}(\boldsymbol{\beta}) & + & \text{vec}(\boldsymbol{\epsilon}) \\ (np \times 1) & & (np \times qp)(qp \times 1) & & (np \times 1). \end{array}$$

- What is Var[vec $\boldsymbol{\epsilon}$]?

- What is the GLS estimator for vec($\boldsymbol{\beta}$)?

A Multivariate Spatial Model

- Extend the multivariate, multiple regression model:

$$\underset{(np \times 1)}{\text{vec}(\mathbf{Y})} = \underset{(np \times qp)(qp \times 1)}{(\mathbf{I}_p \otimes \mathbf{X})\,\text{vec}(\boldsymbol{\beta})} + \underset{(np \times 1)}{\text{vec}(\mathbf{h})} + \underset{(np \times 1),}{\text{vec}(\boldsymbol{\epsilon})}$$

where

$$\text{Var}[\text{vec}(\mathbf{h})] = \boldsymbol{\Sigma}_{\mathbf{h}} = \begin{bmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} & \cdots & \boldsymbol{\Sigma}_{1p} \\ \boldsymbol{\Sigma}'_{12} & \boldsymbol{\Sigma}_{22} & \cdots & \boldsymbol{\Sigma}_{2p} \\ \vdots & & \ddots & \vdots \\ \boldsymbol{\Sigma}'_{1p} & \boldsymbol{\Sigma}'_{2p} & \cdots & \boldsymbol{\Sigma}_{pp} \end{bmatrix}$$

$$\text{Var}[\text{vec}(\boldsymbol{\epsilon})] = \boldsymbol{\Sigma} \otimes \mathbf{I}_n$$

A Multivariate Spatial Model

- One simplification to the spatial covariance matrix is to use a Kronecker form:

$$\begin{aligned} \Sigma_{\mathbf{h}} &= \boldsymbol{\rho} \otimes \mathbf{K} \\ &= \begin{bmatrix} \rho_{11}\mathbf{K} & \rho_{12}\mathbf{K} & \cdots & \rho_{1p}\mathbf{K} \\ \rho_{12}\mathbf{K} & \rho_{22}\mathbf{K} & \cdots & \rho_{2p}\mathbf{K} \\ \vdots & & \ddots & \vdots \\ \rho_{1p}\mathbf{K} & \rho_{2p}\mathbf{K} & \cdots & \rho_{pp}\mathbf{K} \end{bmatrix} \end{aligned}$$

  where

  - $\boldsymbol{\rho}$ is a $p \times p$ matrix of scale parameters
  - $\mathbf{K}$ is an $n \times n$ spatial covariance.

A Multivariate Spatial Model

- Extend the multivariate, multiple regression model:

$$\underset{(np \times 1)}{\text{vec}(\mathbf{Y})} = \underset{(np \times qp)(qp \times 1)}{(\mathbf{I}_p \otimes \mathbf{X})\,\text{vec}(\boldsymbol{\beta})} + \underset{(np \times 1)}{\text{vec}(\mathbf{h})} + \underset{(np \times 1)}{\text{vec}(\boldsymbol{\epsilon})}$$

OR

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{h} + \boldsymbol{\epsilon}$$

- Now everything follows...

The `multiKrig` Class

- Create a multivariate spatial regression within the univariate `Krig` framework:

$$\mathbf{Y} = \mathbf{T}\boldsymbol{\beta} + \mathbf{h} + \boldsymbol{\epsilon}$$

$$\mathbf{Y} = \begin{bmatrix} \mathbf{Y}_1 \\ \vdots \\ \mathbf{Y}_p \end{bmatrix} \qquad \mathbf{T} = \mathbf{I}_p \otimes \mathbf{X} \qquad \mathbf{h} = \begin{bmatrix} \mathbf{h}_1 \\ \vdots \\ \mathbf{h}_p \end{bmatrix} \qquad \boldsymbol{\epsilon} = \begin{bmatrix} \boldsymbol{\epsilon}_1 \\ \vdots \\ \boldsymbol{\epsilon}_p \end{bmatrix}$$

The `multiKrig` Class

- Create a multivariate spatial regression within the univariate `Krig` framework:

$$\mathbf{Y} = \mathbf{T}\boldsymbol{\beta} + \mathbf{h} + \boldsymbol{\epsilon}$$

$$\begin{aligned}
\mathsf{E}[\mathbf{Y}] &= \mathbf{T}\boldsymbol{\beta} \\
\mathsf{Var}[\mathbf{Y}] &= \boldsymbol{\Sigma}_h + \boldsymbol{\Sigma}_{\boldsymbol{\epsilon}} \\
&= \begin{bmatrix} \rho_1 & & \\ & \ddots & \\ & & \rho_p \end{bmatrix} \otimes \mathbf{V}(\theta) + \begin{bmatrix} s_{11} & s_{12} & \cdots & s_{1p} \\ s_{21} & s_{22} & \cdots & s_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ s_{p1} & s_{p2} & \cdots & s_{pp} \end{bmatrix} \otimes \mathbf{I}_n \\
&= \rho_1 \mathbf{R} \otimes \mathbf{V}(\theta) + s_{11} \mathsf{S} \otimes \mathbf{I}_n \\
&= \rho_1 \left(\mathbf{R} \otimes \mathbf{V}(\theta) + \lambda \mathsf{S} \otimes \mathbf{I}_n \right)
\end{aligned}$$

The `multiKrig` Class

- Create a multivariate spatial regression within the univariate `Krig` framework:

$$\mathbf{Y} = \mathbf{T}\boldsymbol{\beta} + \mathbf{h} + \boldsymbol{\epsilon}$$

$$\begin{aligned} \mathsf{E}[\mathbf{Y}] &= \mathbf{T}\boldsymbol{\beta} \\ \mathsf{Var}[\mathbf{Y}] &= \boldsymbol{\Sigma}_h + \boldsymbol{\Sigma}_{\boldsymbol{\epsilon}} \\ &= \rho_1 \left(\mathbf{R} \otimes \mathbf{V}(\theta) + \lambda \mathsf{S} \otimes \mathbf{I}_n \right) \end{aligned}$$

- Given S, $\mathbf{R}$, and θ, use `Krig` to estimate $\boldsymbol{\beta}$, ρ_1 and λ.

The `multiKrig` Class

- Issues:
  - Specifying `x`, `Y`, and `Z`
  - Mean function (`null.function`)
  - Covariance function (`cov.function`)
  - Error function (`wght.function`)
- Estimation (S, $\mathbf{R}$, and θ)

`Krig` Function

```
Krig  <-  function (x, Y, Z,
          null.function = "Krig.null.function",
          cov.function = "stationary.cov",
          wght.function = NULL,
          null.args = NULL, cov.args = NULL, wght.args = NULL)
```

- `x` is an $n \times q$ matrix of spatial locations
- `Y` is a n-vector of observations observations
- `Z` is a $n \times q$ matrix of additional covariates

`multiKrig` Function

```
multiKrig  <-  function(s,Y,Z,
               cov.function="multi.cov",cov.args=NULL,
               wght.function="multi.wght",wght.args=NULL)
```

- `s` is an $n \times q$ matrix of spatial locations
- `Y` is an $n \times p$ matrix of observations
- `Z` is either:
  - a $n \times q$ matrix of additional covariates, or
  - a list of $n \times q_i$ matrices of additional covariates

`multiKrig` Function

```
multiKrig  <-  function(s,Y,Z,
               cov.function="multi.cov",cov.args=NULL,
               wght.function="multi.wght",wght.args=NULL){
               ⋮
               d <- ncol(Y)
               n <- nrow(Y)
               Y <- c(Y)
               ⋮
               x <- expand.grid(1:n,1:d)
               nZ <- kronecker(diag(d),cbind(s,Z))
               ⋮
               obj  <-  Krig(x=x,Y=c(Y),Z=nZ,method="REML",
                        null.function="multi.null",
                        cov.function=cov.function,cov.args=cov.args,
                        wght.function=wght.function,wght.args=wght.args)
               ⋮
               }
```

- Y <- c(Y)

$$\mathbf{Y} = \begin{bmatrix} \mathbf{Y}_1 \\ \vdots \\ \mathbf{Y}_p \end{bmatrix}$$

```
multiKrig  <-  function(s,Y,Z,
               cov.function="multi.cov",cov.args=NULL,
               wght.function="multi.wght",wght.args=NULL){
               ⋮
               d <- ncol(Y)
               n <- nrow(Y)
               Y <- c(Y)
               ⋮
               x <- expand.grid(1:n,1:d)
               nZ <- kronecker(diag(d),cbind(s,Z))
               ⋮
               obj  <-  Krig(x=x,Y=c(Y),Z=nZ,method="REML",
                        null.function="multi.null",
                        cov.function=cov.function,cov.args=cov.args,
                        wght.function=wght.function,wght.args=wght.args)
               ⋮
               }
```

- `x <- expand.grid(1:n,1:d)`

$$\mathbf{x} = \begin{bmatrix} 1 & 1 \\ 2 & 1 \\ \vdots & \\ n & 1 \\ 1 & 2 \\ \vdots & \\ n & p \end{bmatrix}$$

```
multiKrig  <-  function(s,Y,Z,
               cov.function="multi.cov",cov.args=NULL,
               wght.function="multi.wght",wght.args=NULL){
               ⋮
               d <- ncol(Y)
               n <- nrow(Y)
               Y <- c(Y)
               ⋮
               x <- expand.grid(1:n,1:d)
               nZ <- kronecker(diag(d),cbind(s,Z))
               ⋮
               obj  <-  Krig(x=x,Y=c(Y),Z=nZ,method="REML",
                        null.function="multi.null",
                        cov.function=cov.function,cov.args=cov.args,
                        wght.function=wght.function,wght.args=wght.args)
               ⋮
               }
```

- `nZ <- kronecker(diag(d),cbind(s,Z))` $\quad Z = \begin{bmatrix} s\,Z & & \\ & \ddots & \\ & & s\,Z \end{bmatrix}$

```
multiKrig  <-  function(s,Y,Z,
               cov.function="multi.cov",cov.args=NULL,
               wght.function="multi.wght",wght.args=NULL){
               ⋮
               d <- ncol(Y)
               n <- nrow(Y)
               Y <- c(Y)
               ⋮
               x <- expand.grid(1:n,1:d)
               nZ <- kronecker(diag(d),cbind(s,Z))
               ⋮
               obj  <-  Krig(x=x,Y=c(Y),Z=nZ,method="REML",
                        null.function="multi.null",
                        cov.function=cov.function,cov.args=cov.args,
                        wght.function=wght.function,wght.args=wght.args)
               ⋮
               }
```

```
multi.null  <-  function(x,Z=NULL,drop.Z=FALSE){
                data <- data.frame(a=as.factor(x[,2]))
                X <- model.matrix(~a,data=data,contrasts=list(a="contr.treatment"))
                ⋮
                return(cbind(X,Z))
                }
```

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & \mathbf{s\,Z} & 0 & 0 \\ 1 & 1 & 0 & 0 & \mathbf{s\,Z} & 0 \\ 1 & 0 & 1 & 0 & 0 & \mathbf{s\,Z} \end{bmatrix}$$

```
multiKrig  <-  function(s,Y,Z,
               cov.function="multi.cov",cov.args=NULL,
               wght.function="multi.wght",wght.args=NULL){
               ⋮
               obj  <-  Krig(x=x,Y=c(Y),Z=nZ,method="REML",
                        null.function="multi.null",
                        cov.function=cov.function,cov.args=cov.args,
                        wght.function=wght.function,wght.args=wght.args)
               ⋮
               }
```

Covariance Function

- Issue: $\mathbf{x}$ is now a matrix of indices.
- Solution: pass the spatial locations as an argument to the covariance function.

```
multiKrig   <-   function(s,Y,Z,
                 cov.function="multi.cov",cov.args=NULL,
                 wght.function="multi.wght",wght.args=NULL){
                 ⋮
                 cov.args$s <- s
                 obj   <-   Krig(x=x,Y=c(Y),Z=nZ,method="REML",
                            null.function="multi.null",
                            cov.function=cov.function,cov.args=cov.args,
                            wght.function=wght.function,wght.args=wght.args)
                 ⋮
                 }

multi.cov   <-   function(x1,x2,marginal=FALSE,C=NA,s,theta,rho,smoothness){
                 ⋮
                 ind <- unique(x1[,2])
                 temp   <-   stationary.cov(s[x1[,1][x1[,2]==1],],
                             s[x2[,1][x2[,2]==1],],
                             Covariance="Matern",theta=theta,smoothness=smoothness)
                 if   (length(ind)>1){
                      for   (i in 2:length(ind)){
                            temp2   <-   rho[i-1]*stationary.cov(s[x1[,1][x1[,2]==i],],
                                         s[x2[,1][x2[,2]==i],],
                                         Covariance="Matern",theta=theta,smoothness=smoothness)
                            d1 <- dim(temp)
                            d2 <- dim(temp2)
                            temp   <-   rbind(cbind(temp,matrix(0,d1[1],d2[2])),
                                        cbind(matrix(0,d1[1],d2[2]),temp2))
                      }
                 }
                 ⋮
                 return(temp)
                 }
```

Weight Function

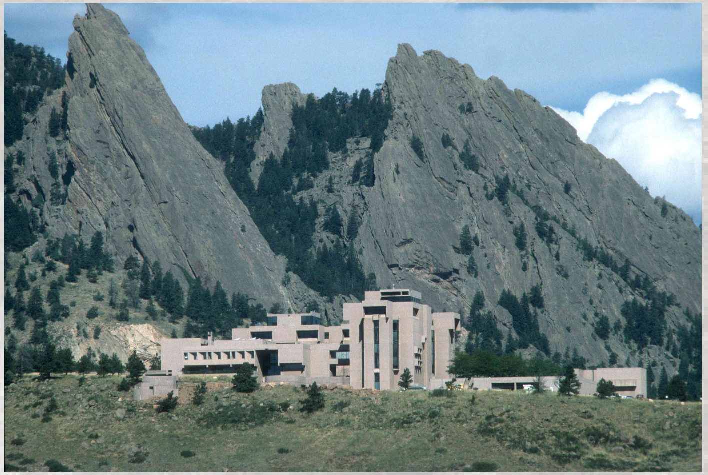
```
multiKrig  <-  function(s,Y,Z,
               cov.function="multi.cov",cov.args=NULL,
               wght.function="multi.wght",wght.args=NULL){
               ⋮
               obj  <-  Krig(x=x,Y=c(Y),Z=nZ,method="REML",
                        null.function="multi.null",
                        cov.function=cov.function,cov.args=cov.args,
                        wght.function=wght.function,wght.args=wght.args)
               ⋮
               }

multi.wght  <-  function(x,sp){
                n <- length(unique(x[,1]))
                ⋮
                return(kronecker(solve(S),diag(n)))
                }
```

Parameter Estimation and Other Issues

- `Krig` will estimate $\boldsymbol{\beta}$, ρ_1 and λ.
  - REML
  - GCV (not quite there...)
- How to estimate S, $\mathbf{R}$, and θ?
- Nonseparable covariance functions.
- Estimation/prediction for missing observations.
- Implementation in `fields`.

Thanks!



ssain@ucar.edu
www.image.ucar.edu/~ssain

- Sain, S.R., Mearns, L., Shrikant, J., and Nychka, D. (2006), "A multivariate spatial model for soil water profiles," *Journal of Agricultural, Biological, and Environmental Statistics*, **11**, 462-480.

- Sain, S.R. and Cressie, N. (2007), "A spatial model for multivariate lattice data," *Journal of Econometrics*, **140**, 226-259, doi:10.1016/j.jeconom.2006.09.010.