R is an increasingly popular freeware alternative to S-plus, available from the web at www.r-project.org.
Spatial data analysis with geoR

- R is an increasingly popular freeware alternative to S-plus, available from the web at www.r-project.org.

- geoR is an R add-on package with many functions for geostatistical analysis
Spatial data analysis with geoR

- R is an increasingly popular freeware alternative to S-plus, available from the web at www.r-project.org.

- geoR is an R add-on package with many functions for geostatistical analysis.

- geoR (and the akima library) can be easily installed from http://cran.us.r-project.org
  
  - in Windows click on Packages and Install from CRAN to select geoR
  
  - in Linux/Unix use the install.package function from within R; see also help(install.package)
Spatial data analysis with \texttt{geoR}

\texttt{R} is an increasingly popular freeware alternative to \texttt{S-plus}, available from the web at \url{www.r-project.org}.

\texttt{geoR} is an \texttt{R} add-on package with many functions for geostatistical analysis.

\texttt{geoR} (and the \texttt{akima} library) can be easily installed from \url{http://cran.us.r-project.org}

- in Windows click on \texttt{Packages and Install from CRAN} to select \texttt{geoR}
- in Linux/Unix use the \texttt{install.package} function from within \texttt{R}; see also \texttt{help(install.package)}

This tutorial describes methods for kriging and related geostatistical operations available in \texttt{geoR}; online at:

\url{www.biostat.umn.edu/~brad/data/booknew.R}
Exploratory tools in R

Consider scallops data in data frame `myscallops`
Consider scallops data in data frame `myscallops`

**Recall:** Often helpful to create **image** plots and place **contour** lines on the plot.
Exploratory tools in \texttt{R}

- Consider scallops data in data frame \texttt{myscallops}
- \textbf{Recall:} Often helpful to create \texttt{image} plots and place \texttt{contour} lines on the plot.
- \textbf{Invoke} \texttt{akima} library: \texttt{library(akima)}
Exploratory tools in R

- Consider scallops data in data frame `myscallops`
- **Recall:** Often helpful to create *image* plots and place *contour* lines on the plot.
- **Invoke** `akima library:` `library(akima)`
- `int.scp <- interp.new(myscallops$long, myscallops$lat, myscallops$lgcatch)`
Consider scallops data in data frame `myscallops`

Recall: Often helpful to create `image` plots and place `contour` lines on the plot.

Invoke `akima` library: `library(akima)`

```
int.scp <- interp.new(myscallops$long, myscallops$lat, myscallops$lgcatch)

image(int.scp,
      xlim=range(myscallops$long),
      ylim=range(myscallops$lat))
```
Consider scallops data in data frame `myscallops`.

Recall: Often helpful to create image plots and place contour lines on the plot.

Invoke `akima` library: `library(akima)`

`int.scp <- interp.new(myscallops$long, myscallops$lat, myscallops$lgcatch)`

`image(int.scp, xlim=range(myscallops$long), ylim=range(myscallops$lat))`

`contour(int.scp, add=T)`
Consider scallops data in data frame `myscallops`

**Recall:** Often helpful to create image plots and place contour lines on the plot.

Invoke **akima library:** `library(akima)`

- `int.scp <- interp.new(myscallops$long, myscallops$lat, myscallops$lgcatch)`
- `image(int.scp, xlim=range(myscallops$long), ylim=range(myscallops$lat))`
- `contour(int.scp, add=T)`
- `persp(int.scp, xlim=..., ylim=...)`
Image plot with contour lines
Variogram fitting with \texttt{geoR}

\begin{verbatim}
myscallops <- (myscallops$long, myscallops$lat, myscallops$lgcatch)
\end{verbatim}
Variogram fitting with geoR

- `myscallops <- (myscallops$long, myscallops$lat, myscallops$lgcatch)`
- **Crucial**: Create a `geodata` object
myScallops <- (myScallops$long, myScallops$lat, myScallops$lgcatch)

Crucial: Create a geodata object

scallops.geo <- as.geodata(myScallops, coords.col=1:2, data.col=3)
Variogram fitting with \texttt{geoR}

\begin{itemize}
  \item \texttt{myscallops <- (myscallops$long, myscallops$lat, myscallops$lgcatch)}
  \item \textbf{Crucial:} Create a \texttt{geodata} object
  \item \texttt{scallops.geo <- as.geodata(myscallops, coords.col=1:2, data.col=3)}
  \item Next a \texttt{variogram} object is created.
\end{itemize}
Variogram fitting with **geoR**

- `myscallops <- (myscallops$long, myscallops$lat, myscallops$lgcatch)`

**Crucial:** Create a `geodata` object

- `scallops.geo <- as.geodata(myscallops, coords.col=1:2, data.col=3)`

Next a `variogram` object is created.

- `scallops.var <- variog(scallops.geo, estimator.type=``'classical''``)`
Variogram fitting with **geoR**

- \[ \text{myscallops} <- (\text{myscallops}\_\text{long}, \text{myscallops}\_\text{lat}, \text{myscallops}\_\text{lgcatch}) \]

- **Crucial:** Create a \textit{geodata} object

- \[ \text{scallops.geo} <- \text{as.geodata}(\text{myscallops}, \text{coords.col}=1:2, \text{data.col}=3) \]

- Next a \textit{variogram} object is created.

- \[ \text{scallops.var} <- \text{variog}(\text{scallops.geo}, \text{estimator.type}=``\text{classical}''') \]

- A \textit{robust} variogram is obtained by setting \text{estimator.type} = “robust”
**Variogram fitting with geoR**

- `myscallops <- (myscallops$long, myscallops$lat, myscallops$lgcatch)`
- **Crucial:** Create a `geodata` object
  - `scallops.geo <- as.geodata(myscallops, coords.col=1:2, data.col=3)`
- Next a `variogram` object is created.
  - `scallops.var <- variog(scallops.geo, estimator.type="classical")`
- A **robust** variogram is obtained by setting `estimator.type = "robust"`
- **Plots:** `plot(scallops.var)`
Variograms: classical and robust

(a)

(b)
geoR provides a wide range of covariance functions for fitting variograms:
Variogram fitting in `geoR`

`geoR` provides a wide range of covariance functions for fitting variograms:

- exponential, gaussian, spherical, matérn etc.
geoR provides a wide range of covariance functions for fitting variograms:

- exponential, gaussian, spherical, matérn etc.

The function `variofit` estimates the sill, the range, and the nugget parameters under a specified covariance model.
Variogram fitting in \texttt{geoR}

\texttt{geoR} provides a wide range of covariance functions for fitting variograms:

- exponential, gaussian, spherical, matérn etc.

The function \texttt{variofit} estimates the sill, the range, and the nugget parameters under a specified covariance model.

Example: \texttt{scallops.var.fit <- variofit(scallops.var, ini.cov.pars = c(1.0,5.0), cov.model=’’exponential’’, fix.nugget=FALSE, nugget=1.0)}
Likelihood model fitting

Both **ML** and **REML** are implemented through `geoR` function `likfit`.
Both **ML** and **REML** are implemented through GeoR function `likfit`.

```r
scallops.lik.fit <- likfit(scallops.geo, ini.cov.pars=c(1.0,2.0), cov.model = "exponential", trend = "cte", fix.nugget = FALSE, nugget = 1.0, nospatial = TRUE, method.lik = "ML")
```
Both **ML** and **REML** are implemented through `geoR` function `likfit`.

```r
scallops.lik.fit <- likfit(scallops.geo, ini.cov.pars=c(1.0,2.0), cov.model = "'exponential'", trend = "'cte'", fix.nugget = FALSE, nugget = 1.0, nospatial = TRUE, method.lik = "'ML'")
```

The option `trend = "'cte'"` means a spatial regression model with constant mean.
Likelihood model fitting

- Both **ML** and **REML** are implemented through the `geoR` function `likfit`.

```r
callops.lik.fit <- likfit(callops.geo, ini.cov.pars=c(1.0,2.0), cov.model = "exponential", trend = "cte", fix.nugget = FALSE, nugget = 1.0, nospatial = TRUE, method.lik = "ML")
```

- The option `trend = "cte"` means a spatial regression model with constant mean.

- **Output:**

```
likfit: estimated model parameters:
    beta tausq sigmasq phi
2.3748 0.0947 5.7675 0.2338
```
Bayesian kriging is carried out by the function `krige.bayes`
Bayesian kriging in `geoR`

Bayesian kriging is carried out by the function `krige.bayes`

This is a handy tool improved upon the likelihood methods by providing posterior samples of all the model parameters, which lead to estimation of their variability.
Bayesian kriging is carried out by the function `krige.bayes`.

This is a handy tool improved upon the likelihood methods by providing posterior samples of all the model parameters, which lead to estimation of their variability.

The `krige.bayes` function is less versatile than WinBUGS.
Bayesian kriging in geoR

Bayesian kriging is carried out by the function `krige.bayes`

This is a handy tool improved upon the likelihood methods by providing posterior samples of all the model parameters, which lead to estimation of their variability.

The `krige.bayes` function is less versatile than WinBUGS:
- more limited in the types of models it can handle
Bayesian kriging in geoR

- Bayesian kriging is carried out by the function \texttt{krige.bayes}

- This is a handy tool improved upon the likelihood methods by providing posterior samples of \textit{all} the model parameters, which lead to estimation of their variability.

- The \texttt{krige.bayes} function is less versatile than \texttt{WinBUGS}:
  - more limited in the types of models it can handle
  - updating is not through MCMC methods
Bayesian kriging in \texttt{geoR}

- Bayesian kriging is carried out by the function \texttt{krige.bayes}

- This is a handy tool improved upon the likelihood methods by providing posterior samples of \textit{all} the model parameters, which lead to estimation of their variability.

- The \texttt{krige.bayes} function is less versatile than \texttt{WinBUGS}:
  - more limited in the types of models it can handle
  - updating is not through MCMC methods

- \textbf{BUT}: an advantage is it offers a wider range of covariance functions
Bayesian kriging example

Fitting a constant mean spatial regression model to the scallops data:
Bayesian kriging example

Fitting a constant mean spatial regression model to the scallops data:

```r
scallops.bayes1 <- krig.bayes(scallops.geo, locations = "no", borders = NULL, model =
model.control(trend.d = "cte",
cov.model = "exponential"), prior =
prior.control(beta.prior = "flat",
sigmasq.prior = "reciprocal",
tausq.rel.prior = "uniform",
tausq.rel.discrete=seq(from=0.0, to=1.0, by=0.01)))
```
Bayesian kriging example

- Fitting a constant mean spatial regression model to the scallops data:

  ```r
  scallops.bayes1 <- krig.bayes(scallops.geo, locations = "no", borders = NULL, model = model.control(trend.d = "cte", cov.model = "exponential"), prior = prior.control(beta.prior = "flat", sigmasq.prior = "reciprocal", tausq.rel.prior = "uniform", tausq.rel.discrete=seq(from=0.0, to=1.0, by=0.01)))
  ```

- For predictions set `locations = PredLoc` where `PredLoc` contains the locations to predict.
Bayesian kriging example

- Obtaining the posterior quantiles:
Bayesian kriging example

- Obtaining the posterior quantiles:
  ```r
  out <- scallops.krige.bayes$posterior
  ```
Bayesian kriging example

- Obtaining the posterior quantiles:
  - `out <- scallops.krige.bayes$posterior`
  - `out <- out$sample`
Bayesian kriging example

Obtaining the posterior quantiles:

```r
out <- scallops.krige.bayes$posterior
out <- out$sample
beta.qnt <- quantile(out$beta, c(0.50, 0.025, 0.975))
```
Bayesian kriging example

Obtaining the posterior quantiles:

```r
out <- scallops.krige.bayes$posterior
out <- out$sample
beta.qnt <- quantile(out$beta, c(0.50, 0.025, 0.975))
phi.qnt <- quantile(out$phi, c(0.50, 0.025, 0.975))
```
Bayesian kriging example

Obtaining the posterior quantiles:

```r
out <- scallops.krige.bayes$posterior
out <- out$sample
beta.qnt <- quantile(out$beta, c(0.50, 0.025, 0.975))
phi.qnt <- quantile(out$phi, c(0.50, 0.025, 0.975))
sigmasq.qnt <- quantile(out$sigmasq, c(0.50, 0.025, 0.975))
```
Bayesian kriging example

Obtaining the posterior quantiles:

```r
out <- scallops.krige.bayes$posterior
out <- out$sample
beta.qnt <- quantile(out$beta, c(0.50, 0.025, 0.975))
phi.qnt <- quantile(out$phi, c(0.50, 0.025, 0.975))
sigmasq.qnt <- quantile(out$sigmasq, c(0.50, 0.025, 0.975))
tausq.rel.qnt <- quantile(out$tausq.rel, c(0.50, 0.025, 0.975))
```
Output

> beta.qnt

<table>
<thead>
<tr>
<th></th>
<th>50%</th>
<th>2.5%</th>
<th>97.5%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1.931822</td>
<td>-6.426464</td>
<td>7.786515</td>
</tr>
</tbody>
</table>

> phi.qnt

<table>
<thead>
<tr>
<th></th>
<th>50%</th>
<th>2.5%</th>
<th>97.5%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.5800106</td>
<td>0.2320042</td>
<td>4.9909913</td>
</tr>
</tbody>
</table>

> sigmasq.qnt

<table>
<thead>
<tr>
<th></th>
<th>50%</th>
<th>2.5%</th>
<th>97.5%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>11.225002</td>
<td>4.147358</td>
<td>98.484722</td>
</tr>
</tbody>
</table>

> tausq.rel.qnt

<table>
<thead>
<tr>
<th></th>
<th>50%</th>
<th>2.5%</th>
<th>97.5%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.03</td>
<td>0.00</td>
<td>0.19</td>
</tr>
</tbody>
</table>

Note: tausq.rel refers to the ratio of the nugget variance to the spatial variance, and is seen to be negligible here.