STAT 5810
Applied Spatial Statistics
Fall Semester 2000

Dr. Jürgen Symanzik

Utah State University

Department of Mathematics and Statistics

3900 Old Main Hill

Logan, UT 84322–3900

Tel.: (435) 797–0696

FAX: (435) 797–1822

e-mail: `symanzik@sunfs.math.usu.edu`

Contents

09/06/00

1 Introduction

1.1 Spatial vs. Non–Spatial Data Analysis

Spatial Analysis:
"quantitative study of phenomena that are located in space"

Spatial Data Analysis/Spatial Statistics:

- observational data available on some process that operates in space (and time)
- interested in methods that describe or explain the bahavior of this process and its relationship to other spatial phenomena
- objective of analysis:
  - increase basic understanding of the process
  - assess evidence in favor of hypotheses concerning it
  - predict values in areas with no observations
- typical data:
  - sample of observations from the process of interest
  - data often very "noisy"
  - data not independent

1.2 Classes of Problems in Spatial Data Analysis

1.2.1 Spatial Point Patterns

- locations of particular point events (e.g., occurrence of a disease, species habitat)
- is there a "significant" pattern, e.g., cluster?
- sometimes one additional variable (or attribute) is known: time of occurrence of each point event

1.2.2 Spatially Continuous Data

- set of point locations with continuous (or discrete) measurements (e.g., amount of rainfall, concentration of a toxic in a river)
- we are usually not interested in locations at which sample measurements were taken (locations may have been randomly chosen or are predetermined, e.g., towns, measurement stations, mobile observer)
- interested in patterns in the measurement values at sample locations
- "model" (or estimate) values of the variable of interest at locations where no measurements were taken
- "spatially continuous" means that measurements can be (theoretically) taken at any location on the earth's (or land) surface
- due to cost and time constraints, we can take measurements only at a limited number of locations

1.2.3 Area Data

- spatial data that has been aggregated to areal units (e.g., counties, health service areas — HSA's, remotely sensed images)
- attributes associated with areas often do not vary continuously over space
- often, given areas are considered to be the only spatial locations at which attributes can be measured (but occasionally, we have missing data, e.g., data that has not been reported for a particular area, invalid measurements due to instrument failure or cloud coverage in case of remotely sensed images)
- model spatial pattern over the fixed areas and determine possible explanations for this pattern

1.2.4 Spatial Interaction Data

- data on "flows" that link a set of locations (either areas or points)
- interested in arrangement of flows (e.g., shipping of goods, customers driving to a shopping mall)
- build models of such flows

- use these models to predict how flows may change under certain scenarios (e.g., if a supermarket is build at location X, how are other supermarkets effected?)

1.3 Concepts in Spatial Data Analysis

1.3.1 Visualizing Spatial Data

- "see" the data to be analyzed, e.g., dot map, choropleth map, micromaps
- simple statistical plots
- often the use of a GIS is involved

1.3.2 Exploring Spatial Data

- "feel" the data
- descriptions of the data
- gain a priori assumptions about the data
- identification of extreme values (outliers)
- develop hypotheses about the data
- more sophisticated graphical plots and statistical summaries

1.3.3 Modelling Spatial Data

- if we want to test hypotheses about the data or estimate some of the parameters, visualization and exploratory methods are not enough
- however, there is not always a clear distinction possible between "visualization" / "exploration" and "exploration" / "modelling"
- simple (non–spatial) statistical models are linear regression, analysis of variance, or (in a wider sense) t–tests (where we want to establish that the data originates from a particular distribution)

1.4 Statistical Notation

1.4.1 Basic Notation

$$Y(\text{or } X, \ldots) = \textbf{random variable (rv)}\text{, representing a stochastic phenomenon of interest}$$

$$f_Y(y) = \begin{cases} \textbf{probability mass function (pmf)} & \\ = \text{probability that rv } Y \text{ takes value } y, & \text{if } Y \text{ is discrete} \\ \textbf{probability density function (pdf)} & \\ = \text{probability density of rv } Y \text{ at value } y, & \text{if } Y \text{ is continuous} \end{cases}$$

$$\begin{aligned} P(Y \in [a,b]) &= \begin{cases} \sum_{y=a}^{b} f_Y(y), & \text{if } Y \text{ is discrete} \\ \int_a^b f_Y(y)dy, & \text{if } Y \text{ is continuous} \end{cases} \\ &= \text{probability that rv } Y \text{ takes value in range } [a,b] \end{aligned}$$

$$\begin{aligned} F_Y(y) = P(Y \in (-\infty, y]) &= \begin{cases} \sum_{u=-\infty}^{y} f_Y(u), & \text{if } Y \text{ is discrete} \\ \int_{-\infty}^{y} f_Y(u)du, & \text{if } Y \text{ is continuous} \end{cases} \\ &= \text{probability that rv } Y \text{ takes value} \le y \\ &= \textbf{cumulative distribution function (cdf)} \end{aligned}$$

$$\begin{aligned} E(Y) &= \begin{cases} \sum_{y=-\infty}^{\infty} y f_Y(y), & \text{if } Y \text{ is discrete} \\ \int_{-\infty}^{\infty} y f_Y(y)dy, & \text{if } Y \text{ is continuous} \end{cases} \\ &= \textbf{expected value (or mean) of rv } Y \end{aligned}$$

$$\begin{aligned} E(g(Y)) &= \begin{cases} \sum_{y=-\infty}^{\infty} g(y) f_Y(y), & \text{if } Y \text{ is discrete} \\ \int_{-\infty}^{\infty} g(y) f_Y(y)dy, & \text{if } Y \text{ is continuous} \end{cases} \\ &= \textbf{expected value (or mean) of a fuction } g \textbf{ of rv } Y \end{aligned}$$

e.g., $g(Y) = (Y - E(Y))^2$

$\Longrightarrow E(g(Y)) = E((Y - E(Y))^2) = Var(Y) =$ **variance or rv** Y

$\sqrt{Var(Y)} =$ **standard deviation of rv** Y

1.4.2 Generalization to More Than 1 RV

$$
\begin{aligned}
\underline{Y} = (Y_1, Y_2) \text{ or } (X, Y) &= \textbf{bivariate random variable} \\
\underline{Y} = (Y_1, \ldots, Y_n) &= n\textbf{–variate random variable (or random vector)} \\
f_{X,Y}(x, y) &= \textbf{joint pdf (or joint pmf)}, \text{ associated with rv } X \text{ taking} \\
& \text{ value } x \text{ and rv } Y \text{ taking value } y \\
Cov(X, Y) &= E((X - E(X))(Y - E(Y))) \\
&= E(X \cdot Y) - E(X) \cdot E(Y) \\
&= \textbf{covariance between rv's } X \textbf{ and } Y\textbf{,} \\
& \text{ providing the approximate “similarity” between } X \text{ and } Y \\
\rho(X, Y) &= \frac{Cov(X, Y)}{\sqrt{Var(X)}\sqrt{Var(Y)}} \\
&= \textbf{correlation between rv's } X \textbf{ and } Y\textbf{,} \\
& \text{ providing information on “linearity” between } X \text{ and } Y; \\
& \text{ always } -1 \leq \rho(X, Y) \leq 1 \\
& \rho(X, Y) = +/-1\text{: strict positive/negative linear relationship} \\
& \rho(X, Y) \text{ close to } +/-1\text{: strong positive/negative linear relationship} \\
& \rho(X, Y) \text{ close to } 0\text{: no linear relationship}
\end{aligned}
$$

Note that rv's X and Y are **independent**, if probabilistic behavior of either one remains the same no matter what value the other might take. Then,

$$f_{X,Y}(x, y) = f_X(x) f_Y(y)$$

and

$$Cov(X, Y) = 0, \quad \rho(X, Y) = 0.$$

1.4.3 Generalization to Spatial RV's

$$
\begin{aligned}
\mathcal{R} \text{ or } \mathcal{D} &= \textbf{region (or domain) of interest} \\
\underline{s} = (s_1, s_2) \in \mathcal{R} &= \textbf{point location} \text{ (in some “rectangular” coordinate system)} \\
& \text{ in region of interest; often } \underline{s} \in \mathcal{R} \subseteq \{(x, y) \mid x \in I\!R, y \in I\!R\} = I\!R^2 \\
\underline{s}_1 = (s_{11}, s_{12}), \underline{s}_2 = (s_{21}, s_{22}) \in \mathcal{R} &: \text{ 2 point locations}
\end{aligned}
$$

$$\begin{aligned}\{Y(\underline{s}) \mid \underline{s} \in \mathcal{R}\} &= \textbf{spatial stochastic process}\\ &\ \text{(possibly non–independent) rv's } Y(\underline{s}), \text{ where } \mathcal{R}\\ &\ \text{consists of finite or infinite many point locations}\\ (y_1, y_2, \ldots) = (y(\underline{s}_1), y(\underline{s}_2), \ldots) &= \text{data observed from rv's } \{Y_1, Y_2, \ldots\} = \{Y(\underline{s}_1), Y(\underline{s}_2), \ldots\}\\ &= \text{“realization” of the spatial process}\end{aligned}$$

1.5 First and Second Order Effects

When modelling:

- one realization of a spatial process not enough to describe the entire world
- need "reasonable" assumptions about the spatial process, e.g., type of distribution (but parameters left unspecified)
- use realization(s) to estimate unknown parameters

Example:
Assuming that ozone level only depends on location (which is not very realistic, but might be used as a very simple model), we could make the following assumptions:

(i) $\{Y(\underline{s}) \mid \underline{s} \in \mathcal{R}\}$ are independent

(ii) distributions of $Y(\underline{s})$ only differ in the mean value but otherwise are identical

(iii) mean value is simple linear function of location, e.g.,

$$E(Y(\underline{s})) = \beta_0 + \beta_1 s_1 + \beta_2 s_2$$

where $\underline{s} = (s_1, s_2)$

(iv) each $Y(\underline{s})$ has a Normal distribution with mean as in (iii) and constant variance σ^2, i.e.,

$$Y(\underline{s}) \sim N(\beta_0 + \beta_1 s_1 + \beta_2 s_2, \sigma^2)$$

We use realization $(y_1, y_2, \ldots)$ to estimate parameters $\beta_0, \beta_1, \beta_2$. ■

09/11/00

Problem:
Unfortunately, independence (as assumed in (i) in the Example above) is very rare for spatial data. Typically, there exists a degree of spatial correlation. Thus, we should incorporate spatial dependence into our model.

Many spatial phenomena can be described as a mixture of **first order effects** and **second order effects**:

First Order Effect: variation in the mean value of a process in space, i.e., a global (or large scale) trend

Second Order Effect: result of the spatial correlation structure, i.e., local (or small scale) effects

<u>Important Types of Second Order Effects:</u>

- $\{Y(\underline{s}) \mid \underline{s} \in \mathcal{R}\}$ is **stationary** (or **homogeneous**) if its statistical properties are independent of the absolute location in $\mathcal{R}$, i.e., $E(Y(\underline{s}))$ and $Var(Y(\underline{s}))$ are constant in $\mathcal{R}$. Also, this implies that $Cov(Y(\underline{s}_i), Y(\underline{s}_j))$ for $\underline{s}_i \neq \underline{s}_j$ depends only on the relative locations (i.e., distance and direction) of $\underline{s}_i$ and $\underline{s}_j$ (but it does not depend on their absolute locations in $\mathcal{R}$).

- $\{Y(\underline{s}) \mid \underline{s} \in \mathcal{R}\}$ is **isotropic** if it is stationary and $Cov(Y(\underline{s}_i), Y(\underline{s}_j))$ depends only on the distance between $\underline{s}_i$ and $\underline{s}_j$ but not on the direction in which they are separated.

- If mean, variance, or covariance structure "drifts" over $\mathcal{R}$, the process exhibits **non–stationarity** (or **heterogeneity**).

<u>Note:</u>

- Stationarity and first/second order effects are artefacts introduced by the modeller to describe nature, but they do not exist themselves in nature.

- Any statistical (spatial) model is an abstraction of reality, but not reality itself.

- When designing models (or selecting the "best" among several equally good models), always use your expert knowledge on the subject area to see whether a model makes sense.

2 Analysis of Spatial Point Patterns (SPP)

2.1 Introduction

Notation:

$$\begin{aligned} \mathcal{R} &= \text{region of interest} \\ \underline{s}_i = (s_{i1}, s_{i2}) &= \text{point location } i\text{, i.e., } x \text{ and } y \text{ coordinates where the } i^{th} \text{ ``event'' took place} \\ (\underline{s}_1, \underline{s}_2, \ldots) &= \text{spatial point pattern, i.e., series of point location} \end{aligned}$$

Question:
Do observed events show any systematic pattern or are they located at random in $\mathcal{R}$? Is there any clustering or some regularity?

Assumption:
We assume that all events in the region of interest have been observed, i.e., we have a **mapped point pattern** (in contrast to a **sampled point pattern**, where only events in a sample of different areas of the whole region of interest are recorded — often used in forestry when measuring trees, biology when counting animals, etc.).

One Possible Presentation of a SPP:
Number of events occurring in arbitrary sub–areas $\mathcal{A}$ of region $\mathcal{R}$:

$$\{Y(\mathcal{A}) \mid \mathcal{A} \subseteq \mathcal{R}\},$$

where $Y(\mathcal{A})$ is a rv representing the number of events in sub–area $\mathcal{A}$.
Then, $E(Y(\mathcal{A}))$ and $Cov(Y(\mathcal{A}_i), Y(\mathcal{A}_j))$ depend on the size of the areas involved. Not particularly useful.

Better Presentation of a SPP:
Characterize first and second order properties of the SPP based on the limiting behavior of these quantities "per unit area".

- First order properties described as **intensity** $\lambda(\underline{s})$ = mean number of events per unit area at point $\underline{s}$:
$$\lambda(\underline{s}) = \lim_{ds \to 0} \frac{E(Y(\underline{ds}))}{ds}$$
where $\underline{ds}$ is a small region around point $\underline{s}$ and ds is the area of region $\underline{ds}$.

For a stationary point process, we have

$$\lambda(\underline{s}) = \lambda,$$

i.e., a constant over $\mathcal{R}$. Therefore,

$$E(Y(\mathcal{A})) = \lambda A,$$

where A is the area of $\mathcal{A} \subseteq \mathcal{R}$.

- Second order properties described as **second order intensity** $\gamma(\underline{s}_i, \underline{s}_j)$ = relationship between numbers of events in pairs of areas in $\mathcal{R}$:

$$\gamma(\underline{s}_i, \underline{s}_j) = \lim_{ds_i \to 0, ds_j \to 0} \frac{E(Y(\underline{ds}_i)Y(\underline{ds}_j))}{ds_i \; ds_j},$$

where $\underline{ds}_i$ ($\underline{ds}_j$) is a small region around point $\underline{s}_i$ ($\underline{s}_j$) and ds_i (ds_j) is the area of region $\underline{ds}_i$ ($\underline{ds}_j$).

For a stationary point process, we have

$$\gamma(\underline{s}_i, \underline{s}_j) = \gamma(\underline{s}_i - \underline{s}_j) = \gamma(\underline{h}),$$

i.e., the second order intensity only depends on the vector difference $\underline{h}$ (direction and distance) between $\underline{s}_i$ and $\underline{s}_j$ but not on the absolute locations in $\mathcal{R}$.

For an isotropic point process, we have

$$\gamma(\underline{s}_i, \underline{s}_j) = \gamma(h),$$

i.e., the second order intensity only depends on the length h of the distance vector $\underline{h}$ between $\underline{s}_i$ and $\underline{s}_j$ but not on its orientation.

2.2 Visualizing Spatial Point Patterns

Dot Map (i.e., scatterplot) of spatial point pattern:

- gives initial impression where events occurred in the region of interest
- problem with large number of events due to overplotting
- misleading when population varies over region of interest and occurrence of events is related to the number of individuals (e.g., diseases); low population density $\leftrightarrow$ few events; high population density $\leftrightarrow$ many events

Solution:

- Density equalized map (or cartogram) [see later this semester]
- Density plot [see next section]

2.3 Exploring Spatial Point Patterns

2.3.1 Quadrat Methods

- partition $\mathcal{R}$ into sub–regions of equal area (called **quadrats**) and use counts of the number of events in each quadrat to summarize the spatial pattern; then plot 2–dimensional histogram
- this plot shows how intensity varies over $\mathcal{R}$
- original point pattern has been transformed into area data [see more details later when discussing area data]
- many details of the originally observed pattern are lost
- with many small quadrats, high variability in quadrat counts (many empty cells!)
- better: "moving window", i.e., a suitably sized "window" is moved over a fine grid of locations in $\mathcal{R}$; intensity at each point is estimated from the event count per unit area within the "window" centered on that point; as a result, we get a spatially smooth estimate how $\lambda(\underline{s})$ is varying over $\mathcal{R}$; an extension of this is kernel estimation [see next section]

2.3.2 Kernel Estimation

Kernel Estimation was originally developed to obtain a smooth estimate of a univariate or multivariate pdf from an observed sample of observations, i.e., a smooth histogram.

Estimating the intensity of a SPP is very similar to estimating a bivariate pdf. Therefore, bivariate kernel estimation can be easily adapted to give an estimate of the intensity.

<u>Notation:</u>

$$\begin{aligned} \underline{s} &= \text{arbitrary location in } \mathcal{R} \\ \underline{s}_1, \ldots, \underline{s}_n &= \text{locations of the } n \text{ observed events} \end{aligned}$$

Intensity $\lambda(\underline{s})$ at location $\underline{s}$ estimated by

$$\hat{\lambda}_\tau(\underline{s}) = \frac{1}{\delta_\tau(\underline{s})} \sum_{i=1}^{n} \frac{1}{\tau^2} k\left(\frac{\underline{s} - \underline{s}_i}{\tau}\right),$$

where

$$k(\cdot) = \text{a suitably chosen bivariate pdf, called } \textbf{kernel},$$

which is symmetric about the origin

$$
\begin{aligned}
\tau > 0 &= \textbf{bandwidth} \text{ that determines amout of smoothing} \\
& \text{(essentially the radius of a disc centered on } \underline{s} \text{ within which} \\
& \text{points } \underline{s}_i \text{ will contribute "significantly" to } \hat{\lambda}_\tau(\underline{s})) \\
\delta_\tau(\underline{s}) &= \int_{\mathcal{R}} \frac{1}{\tau^2} k\left(\frac{\underline{s}-\underline{u}}{\tau}\right) d\underline{u} \\
&= \textbf{edge correction}\text{, i.e., the volume under the scaled} \\
& \text{kernel centerd on } \underline{s} \text{ which lies "inside" } \mathcal{R}
\end{aligned}
$$

For most "reasonable" choices of the pdf for $k(\cdot)$, the kernel estimate $\hat{\lambda}_\tau(\underline{s})$ will be very similar for a given bandwidth τ.

Frequently used is the **quartic kernel**:

$$
k(\underline{z}) = \begin{cases} \frac{3}{\pi}(1 - \underline{z}\underline{z}^T)^2, & \text{if } \underline{z}\underline{z}^T \leq 1 \\ 0, & \text{otherwise} \end{cases}
$$

Therefore,

$$
\hat{\lambda}_\tau(\underline{s}) = \frac{1}{\delta_\tau(\underline{s})} \sum_{h_i \leq \tau} \frac{3}{\pi\tau^2} \left(1 - \frac{h_i^2}{\tau^2}\right)^2
$$

where

$$
h_i = \text{distance between } \underline{s} \text{ and event location } \underline{s}_i
$$

Note that the summation is only over values of h_i which do not exceed τ. The region of influence in which observed events contribute to $\hat{\lambda}_\tau(\underline{s})$ is a circle of radius τ centered on $\underline{s}$.

<u>Choice of τ:</u> impacts the "stretch" of the region around $\underline{s}$

- τ too large: $\hat{\lambda}_\tau(\underline{s})$ will be flat, local features will be obscured
- τ too small: $\hat{\lambda}_\tau(\underline{s})$ often a collection of spikes centered on the $\underline{s}_i$'s
- "rough" recommendation: $\tau = \frac{0.68}{n^2}$ when $\mathcal{R}$ is the unit square and n is the number of observed events in $\mathcal{R}$

09/18/00

2.3.3 Nearest Neighbor Distances

Nearest Neighbor Event–Event Distance W:
distance between a randomly chosen event and the nearest neighboring event

Nearest Neighbor Point–Event Distance X:
distance between a randomly selected point in $\mathcal{R}$ and the nearest neighboring observed event (particularly useful in sampling)

- both nearest neighbor distances provide useful information about inter–event interactions at a small physical scale
- if there is a large scale variation in the intensity of a point pattern over $\mathcal{R}$, these measures are not very meaningful

Estimation of W and X:
Use of the **empirical cdf's** $\hat{G}(w)$ of W and $\hat{F}(x)$ of X:

$$\hat{G}(w) = \frac{\#(w_i \leq w)}{n}$$

$$\hat{F}(x) = \frac{\#(x_i \leq x)}{m}$$

where

$\# =$ "the number of"
$n =$ number of events in $\mathcal{R}$
$m =$ number of random points sampled

Useful Plot: $\hat{G}(w)$ or $\hat{F}(x)$ against w or x
Interpretation:
first steep, then flat: clustering due to inter–event attraction
first flat, then steep: inter–event repulsion or regularity

2.3.4 The K Function

Nearest neighbor distances only consider smallest scales of pattern; information on larger scales of pattern is ignored; therefore, we better use an estimate of the **reduced second moment measure** or **K function** of the observed process.

The K function relates to the second order properties of an isotropic process; if there are large scale first order effects, spatial dependence indicated by the K function might be due to first order effects rather than to interaction effects.

Definition:

$$\lambda K(h) = E(\#\text{ events within distance } h \text{ of an arbitrary event})$$

where

$$\begin{aligned}
\# &= \text{“the number of”} \\
E &= \text{expectation} \\
\lambda &= \text{intensity (constant over } \mathcal{R} \text{ for isotropic process)}
\end{aligned}$$

If R = area of $\mathcal{R}$, then the expected number of events in $\mathcal{R}$ is λR. Therefore, the expected number of ordered pairs of events at distance at most h apart, with the first event in $\mathcal{R}$, is

$$\lambda R \cdot \lambda K(h) = \lambda^2 R K(h)$$

Estimation of $K(h)$:

$$\hat{K}(h) = \frac{1}{\lambda^2 R} \sum_i \sum_{j, i \neq j} I_h(d_{ij})$$

where

$$\begin{aligned}
d_{ij} &= \text{distance between } i^{th} \text{ and } j^{th} \text{ observed events in } \mathcal{R} \\
I_h(d_{ij}) &= \text{indicator function} \\
&= \begin{cases} 1, & \text{if } d_{ij} \leq h \\ 0, & \text{otherwise} \end{cases}
\end{aligned}$$

Note that

$$\sum_i \sum_{j, i \neq j} I_h(d_{ij})$$

is the observed number of ordered pairs of events at distance at most h apart.

Since intensity λ is unknown, we use

$$\begin{aligned}
\hat{\lambda} &= \frac{n}{R} \\
\Longrightarrow \hat{K}(h) &= \frac{R}{n^2} \sum_i \sum_{j, i \neq j} I_h(d_{ij})
\end{aligned}$$

Problem:
Points that are close to the edge: summation excludes pairs of events for which the second event is outside $\mathcal{R}$ (and thus is unobservable). Therefore, we need some **edge correction**.

Solution:
Consider circle centered on event i, passing through point j. Let w_{ij} = proportion of the circumference of the circle which lies within $\mathcal{R}$ = conditional probability that an event is observed in $\mathcal{R}$, given that it is within distance d_{ij} from the i^{th} event. Then,

$$\hat{K}(h) = \frac{R}{n^2} \sum_i \sum_{j, i \neq j} \frac{I_h(d_{ij})}{w_{ij}}$$

Note that the calculation of w_{ij} is simple for rectangular or circular $\mathcal{R}$, but can be computationally intensive for arbitrarily shaped $\mathcal{R}$. Also, when using edge correction algorithms, h should be relatively small compared to $\mathcal{R}$. In practice, this is no problem since we do not attempt to explore second order effects which operate on the same physical scale as the dimensions of $\mathcal{R}$.

Useful Plot: $\hat{K}(h)$ against h
Interpretation:
For a "random" spatial point process, we assume that the probability of the occurrence of an event at any point in $\mathcal{R}$ is independent of the other events that have occurred and the probability of the occurrence of an event is equal over the whole $\mathcal{R}$. Thus, for such a "random" process,

$$E(\# \text{ of events within distance } h \text{ of an arbitrary event}) = \lambda \pi h^2$$

$\Longrightarrow K(h) = \pi h^2$ for a homogeneous process with no spatial dependence

$\Longrightarrow$ under regularity: $K(h) < \pi h^2$ for h small (compared to $\mathcal{R}$)

under clustering: $K(h) > \pi h^2$ for h small (compared to $\mathcal{R}$)

To compare $\hat{K}(h)$ with πh^2, we often use the transformation

$$\hat{L}(h) = \sqrt{\frac{\hat{K}(h)}{\pi}} - h$$

and plot $\hat{L}(h)$ against h.

$\Longrightarrow$ positive peaks: clustering for these distances of h

negative peaks: regularity for these distances of h

Note that S+SpatialStats uses the slightly different transformation

$$\hat{L}(h) = \sqrt{\frac{\hat{K}(h)}{\pi}}$$

which will result in a straight line through the origin with slope 1 when plotting $\hat{L}(h)$ against h for a "random" process.

09/25/00

2.4 Modelling Spatial Point Patterns

2.4.1 Complete Spatial Randomness (CSR)

For events that follow a **homogeneous Poisson process** over the study region $\mathcal{R}$, the following holds:

- for the set of rv's $\{Y(\mathcal{A}) \mid \mathcal{A} \in \mathcal{R}\}$, it holds that $Y(\mathcal{A}_i)$ and $Y(A_j)$ are independent for any choices of $\mathcal{A}_i$ and A_j
- the probability distribution of $Y(\mathcal{A})$ is a Poisson distribution with mean value λA, where A is the area of $\mathcal{A}$, i.e., the pdf of $Y(\mathcal{A})$ is

  $$f_{Y(\mathcal{A})}(y) = \frac{(\lambda A)^y}{y!} e^{-\lambda A},$$

  where

  $$\lambda \;=\; \text{constant intensity} \;=\; \text{mean number of events per unit area}$$

- conditional on n (the total number of events in $\mathcal{R}$), events are independently and uniformly distributed over $\mathcal{R}$, i.e.,
  - any event has an equal probabilty of occurring at any position in $\mathcal{R}$
  - position of any event is independent of the position of any other event (i.e., events do not interact with one another)
- simulation of n events from a homogeneous Poisson process:
  - enclose $\mathcal{R}$ in a rectangle $\{(x, y) \mid x_1 \leq x \leq x_2, y_1 \leq y \leq y_2\}$
  - generate events with x–coordinates from a $U(x_1, x_2)$ distribution and y–coordinates from a $U(y_1, y_2)$ distribution
  - "reject" any event with coordinates (x, y) that does not lie in $\mathcal{R}$
  - continue until n events have been obtained in $\mathcal{R}$
- CSR represents null hypothesis against which to assess whether an observed pattern is regular, clustered, or random

2.4.2 Simple Quadrat Tests for CSR

Index of Dispersion Test

- let $(x_1, \ldots, x_m)$ be the counts of the number of events in m quadrats, either randomly scattered in $\mathcal{R}$, or alternatively forming a regular grid covering the whole of $\mathcal{R}$
- idea: if these counts follow a Poisson distribution (recall that for a Poisson distribution with parameter λ, it holds $E(X) = Var(X) = \lambda$), one would expect the mean and variance of the counts to be equal
- use test statistic
$$X^2 = \frac{(m-1)s^2}{\overline{x}} = \frac{\sum_{i=1}^{m}(x_i - \overline{x})^2}{\overline{x}}$$
where
$$\begin{aligned} \overline{x} &= \text{mean of the observed counts} \\ s^2 &= \text{variance of the observed counts} \end{aligned}$$
- under CSR, the theoretical probability distribution of X^2 can be approximated by a χ^2_{m-1} distribution (i.e., a Chi–squared distribution with $m-1$ degrees of freedom), provided that $m > 6$ and $\overline{x} > 1$
- significance test compares observed value of X^2 with percentiles of the χ^2_{m-1} distribution
- significantly large (or small) values of X^2 indicate clustered (or regular) departures from CSR
- $\frac{s^2}{\overline{x}}$ is called **index of dispersion**
- $\frac{s^2}{\overline{x}} - 1$ is called **index of cluster size** (ICS)
- for CSR, $E(ICS) = 0$
if $ICS > 0 \implies$ clustering (number of extra events)
if $ICS < 0 \implies$ regularity (deficiency in events)

Estimation of λ Related to Dispersion Test:

- similarly, if we assume that intensity λ is a constant and CSR holds, use $(x_1, \ldots, x_m)$ based on m quadrats that are randomly scattered in $\mathcal{R}$ to estimate λ as
$$\hat{\lambda} = \frac{\overline{x}}{Q} = \frac{\sum_{i=1}^{m} x_i}{mQ}$$

where

$$Q = \text{area of each quadrat (constant)}$$

- since
$$Var(\hat{\lambda}) \overset{approx}{\equiv} \frac{\hat{\lambda}}{mQ},$$
an approximate 95% confidence interval for $\hat{\lambda}$ is
$$\hat{\lambda} \pm 2\sqrt{\frac{\hat{\lambda}}{mQ}}$$

- note that if m is large, randomly scattered quadrats are likely to overlap and the x_i counts will not be independent; also overlaps with the edge of $\mathcal{R}$ can be a problem

2.4.3 Nearest Neighbor Tests for CSR

Recall the nearest neighbor event–event distance W and nearest neighbor point–event distance X with cdf's $G(w)$ and $F(x)$. Unfortunately, theoretical distributions $G(w)$ of W and $F(x)$ of X under CSR are hard (or even impossible) to determine since they depend on the shape of $\mathcal{R}$ due to edge effects. However, theoretical distributions of W and X can be obtained when edge effects are ignored.

Let $\mathcal{A}$ = circle of radius x around a randomly chosen point

$\Longrightarrow A$ = area of $\mathcal{A} = \pi x^2$

$\Longrightarrow$ under CSR, probability that no events fall in $\mathcal{A}$ is

$$f_{Y(\mathcal{A})}(0) = \frac{(\lambda A)^0}{0!} e^{-\lambda A} = e^{-\lambda \pi x^2}$$

$\Longrightarrow$ under CSR, cdf $F(x)$ of nearest neighbor point–event distance X is

$$F(x) = P(X \leq x) = 1 - e^{-\lambda \pi x^2}, \text{ for } x \geq 0$$

$\Longrightarrow$ under CSR,
$$\pi X^2 \sim Exp(\lambda)$$

and
$$2\pi\lambda X^2 \sim \chi^2_2$$

where $S \sim Exp(\lambda)$ has pdf

$$f_S(s) = \lambda e^{-\lambda s}, \text{ for } s \geq 0, \lambda > 0$$

$$\Longrightarrow E(S) = \frac{1}{\lambda}, Var(S) = \frac{1}{\lambda^2}$$

and $T \sim \chi_2^2$ has pdf

$$f_T(t) = \frac{1}{2} e^{-\frac{1}{2}t}, \text{ for } t \geq 0$$

$$\Longrightarrow E(T) = 2, Var(T) = 4$$

$\Longrightarrow E(X) = \frac{1}{2\sqrt{\lambda}}$ and $Var(X) = \frac{4-\pi}{4\lambda\pi}$

Also, if $X_1, \ldots, X_n$ are independent nearest neighbor point–event distances, then

$$2\pi\lambda \sum_{i=1}^{n} X_i^2 \sim \chi_{2n}^2$$

Under CSR, similar arguments hold for the nearest neighbor event–event distance W, i.e., cdf $G(w)$ is

$$G(w) = P(W \leq w) = 1 - e^{-\lambda\pi w^2}, \text{ for } w \geq 0$$

$\Longrightarrow E(W), Var(W)$ are the same as $E(X), Var(X)$

Knowledge of theoretical distribution of W and X under CSR allows to derive (approximate) sampling distributions under CSR of major summary statistics of observed nearest neighbor distances. Theory requires that nearest neighbor distances have been independently sampled from study region $\mathcal{R}$ and have not been biased by edge effects.

If all nearest neighbor distances are taken into account, these are not independent (in the extreme: 2 events – both nearest neighbor distances are identical). It is recommended to use a sample of size m nearest neighbor distances, where $m \leq \frac{n}{10}$ and n = total number of events in $\mathcal{R}$. We have to be careful with a mapped point pattern, i.e., when all events in $\mathcal{R}$ have been observed, and usually cannot use all n nearest neighbor distances.

Also, nearest neighbor distances of points located near the edges are biased. Use edge correction to avoid these points or fix the problem.

Commonly Used Tests to Detect Departure from CSR that are based on summary statistics of m randomly sampled nearest neighbor event–event distances $(w_1, \ldots, w_m)$ or nearest neighbor point–event distances $(x_1, \ldots, x_m)$ are

- Clark–Evans Test
- Hopkins Test
- Byth & Ripley test

Clark–Evans Test:
Compare

$$\overline{w} = \frac{\sum_{i=1}^{m} w_i}{m}$$

with the percentage points of a

$$N(\frac{1}{2\sqrt{\lambda}}, \frac{4-\pi}{4\lambda\pi m})$$

distribution.

Since λ is unknown, we have to estimate λ by $\hat{\lambda} = \frac{n}{R}$, where n is the number of events in $\mathcal{R}$ and $R =$ area of $\mathcal{R}$. In this case, it is also desirable to use all n event–event distances rather than a random sample of m of them.

An approximate correction to $E(\overline{W})$ and $Var(\overline{W})$ is

$$\begin{aligned} E(\overline{W}) &= 0.5\sqrt{\frac{R}{n}} + 0.051\frac{P}{n} + 0.041\frac{P}{n^{3/2}} \\ Var(\overline{W}) &= 0.070\frac{R}{n^2} + 0.037P\sqrt{\frac{R}{n^5}} \end{aligned}$$

if all nearest neighbor distances are used ($m = n$), P is the perimeter of the study area $\mathcal{R}$, and $\mathcal{R}$ is reasonably shaped (and not very combusted).

In this case, we compare

$$\overline{w} = \frac{\sum_{i=1}^{n} w_i}{n}$$

with percentage points of a

$$N(E(\overline{W}), Var(\overline{W}))$$

distribution.

Note that we reject H_0 : "CSR" if $\overline{w}$ is either too small (i.e., clustering) or too large (i.e., regularity).

10/02/00

Hopkins Test:
Compare

$$\frac{\sum_{i=1}^{m} x_i^2}{\sum_{i=1}^{m} w_i^2}$$

with percentage points of a $F_{2m,2m}$ distribution.

Idea:

- in clustered patterns, point–event distances x_i will be large relative to the event–event distances w_i
- in regular patterns, event–event distances w_i will be large relative to the point–event distances x_i

Byth & Ripley Test:

Compare

$$\frac{1}{m}\sum_{i=1}^{m}\frac{x_i^2}{x_i^2+w_i^2}$$

with percentage points of a $N(\frac{1}{2}, \frac{1}{12m})$ distribution, where the x_i values are randomly paired with the w_i values.

The idea behind this test is similar to the idea behind the Hopkins test.

Tests based on empirical cdf's $\hat{G}(w)$ of W and $\hat{F}(x)$ of X:

After edge correction, plot $\hat{G}(w)$ or $\hat{F}(x)$ against their theoretical equivalents under CSR, i.e.,

$$G(w) = 1 - e^{-\lambda\pi w^2} \quad \text{or} \quad F(x) = 1 - e^{-\lambda\pi x^2}.$$

"Large" differences in the plot indicate departures from CSR. What is "large"? We will use simulation to answer this question:

- Simulate CSR process of n points (where n is the number of events in the observed process) m times in the same region $\mathcal{R}$, resulting in $\hat{G}_i(w), i = 1, \ldots, m$ (or $\hat{F}_i(x), i = 1, \ldots, m$), i.e., m empirical cdf's (do *not* apply edge correction here when calculating these empirical cdf's)
- $$\overline{G}(w) = \frac{1}{m}\sum_{i=1}^{m}\hat{G}_i(w) \quad \text{and} \quad \overline{F}(x) = \frac{1}{m}\sum_{i=1}^{m}\hat{F}_i(x)$$

  are simulation estimates of $G(w)$ and $F(x)$
- Define **upper and lower simulation envelopes**:

  $$\begin{aligned} U(w) &= \max_{i=1,\ldots,m} \hat{G}_i(w) \\ L(w) &= \min_{i=1,\ldots,m} \hat{G}_i(w) \end{aligned}$$

  (similarly for $U(x)$ and $L(x)$)

- Plot $\hat{G}(w)$ (without edge correction) against $\overline{G}(w)$ and add $U(w)$ and $L(w)$ (similarly for $\hat{F}(x)$ and $U(x)$ and $L(x)$)

- Under CSR, plot should be roughly linear and at 45 degrees;
  in case of clustering, plot of $\hat{G}(w)$ against $\overline{G}(w)$ will lie above the 45 degrees line;
  in case of regularity, plot of $\hat{G}(w)$ against $\overline{G}(w)$ will lie below the 45 degrees line

- $U(w)$ and $L(w)$ help to assess significance of departures from the 45 degrees line since

  $$P(\hat{G}(w) > U(w)) = P(\hat{G}(w) < L(w)) = \frac{1}{m+1}.$$

  Thus,

  $$P(\hat{G}(w) > U(w) \text{ or } \hat{G}(w) < L(w)) = \frac{2}{m+1}.$$

  Obviously, if we just make $m = 1$ simulation run, it is $U(w) = L(w)$ and

  $$P(\hat{G}(w) \neq U(w)) = 1.$$

  We need $m = 39$ to obtain

  $$P(\hat{G}(w) > U(w) \text{ or } \hat{G}(w) < L(w)) = \frac{2}{39+1} = \frac{2}{40} = 0.05,$$

  i.e., to obtain a test at the 5% significance level.

  Note that we have to look at each w separately. If it happens for some w that $\hat{G}(w)$ falls outside the region defined by $U(w)$ and $L(w)$, this may still be explained by pure chance and does not necessarily denote a departure from CSR. However, if $\hat{G}(w)$ falls outside the region defined by $U(w)$ and $L(w)$ for many values of w, this certainly cannot be explained by pure chance any more, i.e., in fact, there would be a departure from CSR.

2.4.4 K Function Tests for CSR

Recall from Section 2.3.4 that under CSR,

$$K(h) = \pi h^2$$

and

$$L(h) = \sqrt{\frac{K(h)}{\pi}} - h = 0,$$

i.e., a plot of $\hat{L}(h)$ vs. h is identical to the horizontal axis.

Since the sampling distributions of $\hat{K}(h)$ and $\hat{L}(h)$ are difficult to determine (when considering the problem to build in appropriate edge correction), simulation might be a good idea again:

Similar to the simulation of $\hat{G}(w)$ and $\hat{F}(x)$ described in the previous section, we can simulate a CSR process of n points m times and construct

$$\begin{aligned} U(h) &= \max_{i=1,\ldots,m} \hat{L}_i(h) \\ L(h) &= \min_{i=1,\ldots,m} \hat{L}_i(h) \end{aligned}$$

It holds

$$P(\hat{L}(h) > U(h)) = P(\hat{L}(h) < L(h)) = \frac{1}{m+1}.$$

If we take $m = 99$, it holds

$$P(\hat{L}(h) > U(h)) = \frac{1}{99+1} = \frac{1}{100} = 0.01,$$

i.e., we obtain a 1–sided test at the 1% significance level that rejects H_0: "CSR" in favor of H_A: "Clustering at distance h" in the case that $\hat{L}(h) > U(h)$.

2.4.5 Other Models for Spatial Point Patterns

<u>Models for Clustering:</u>

- Heterogeneous Poisson process
- Cox process
- Poisson cluster process
- Markov point process

<u>Models for Regularity:</u>

- Simple inhibition process
- Markov point process

<u>Heterogeneous Poisson Process:</u>

- constant intensity λ of CSR process is replaced by variable intensity $\lambda(\underline{s})$
- occurrence of any event remains independent of that of any other event
- a simple non–stationary point process with first order effects only

- Simulation:
  - simulate CSR process on $\mathcal{R}$ with intensity $\lambda_{max} = \max_{\underline{s} \in \mathcal{R}} \lambda(\underline{s})$
  - independently retain an event at location $\underline{s}$ with probability $\frac{\lambda(\underline{s})}{\lambda_{max}}$
  - realization of the heterogeneous Poisson process consists of the retained events

Cox Process:

- extension of the heterogeneous Poisson process
- intensity $\lambda(\underline{s})$ varies randomly (and not deterministically)
- also called "doubly stochastic" process since $\lambda(\underline{s})$ is drawn from a distribution over $\mathcal{R}$ and then, conditional on the value of $\lambda(\underline{s})$, events from a heterogeneous Poisson process with intensity $\lambda(\underline{s})$ are created
- process is stationary if and only if the probability distribution from which $\lambda(\underline{s})$ is generated is stationary; otherwise it is non–stationary
- Simulation:
  - first simulate the probability distribution for $\lambda(\underline{s})$ on $\mathcal{R}$
  - then use the rejection sampling technique introduced for the heterogeneous Poisson process

Poisson Cluster Process:

- explicit incorporation of a spatial clustering mechanism into the model
- parent events form a CSR process
- each parent produces a random number of offsprings that are realized independently and identically for each parent according to a pdf $f(\cdot)$
- positions of the offsprings relative to their parents are independently and identically distributed according to some bivariate pdf $g(\cdot)$
- the final process consists of the offsprings only
- this process is stationary
- if $g(\cdot)$ is radially symmetric, this process is also isotropic

- Simulation:
  - first simulate the parent process to obtain the location of parent events
  - for each parent event simulate independently the number of offspring events according to $f(\cdot)$
  - independently locate each of the offspring events around its parent event according to $g(\cdot)$
  - the offsprings form a realization of the process
  - to avoid edge effects, the parents must be simulated over a region which is larger than $\mathcal{R}$ such that the contribution of offsprings falling into $\mathcal{R}$ from parents falling outside $\mathcal{R}$ is not lost

Note:

- certain Poisson cluster processes and certain Cox processes can be statistically indistinguishable, in particular when the offspring distribution $f(\cdot)$ in the Poisson cluster process is another Poisson distribution
- thus, we may not be able to distinguish between a heterogeneous rate of occurrence (first order heterogeneity and spatial independence) and spatial interaction (dependence in form of stationary second order effects)
- e.g., in case of a rare disease, we may not be able to distinguish between an explanation in terms of varying exposure to risk (e.g., worse health conditions in a city than in a rural area) or in terms of infection due to contacts between individual cases (e.g., more people living in a city and thus a higher risk to become infected than in a rural area)

Simple Inhibition Process:

- incorporate minimum permissible distance between any two events
- Model I (**Hard Core Process**): "thin" a CSR process by deleting all pairs of events at a distance less than δ apart
- Model II (**Sequential Spatial Inhibition, SSI**): generate a CSR process one point at a time and discard points if they lie within a distance δ of any previously generated point (or within a distance δ of any previously retained point); this procedure terminates when a prespecified number of points has been placed or when it is impossible to continue
- these processes are stationary

- Simulation:
  - directly mimick the process by which these processes are defined

Markov Point Process: 10/09/00

- more flexibility for the modelling of regularity than a simple inhibition process, e.g., it might be unlikely but not impossible that two events occur in close proximity
- can be mathematically complex models
- simpler ones are *pairwise interaction processes*, e.g., the Strauss process

Strauss Process:

- points in $\mathcal{R}$ are considered neighbors if they are less than some distance $\delta > 0$ apart
- joint density function for n point locations $(\underline{s}_1, \ldots, \underline{s}_n)$ in $\mathcal{R}$ which contains m distinct pairs of neighbors is specified as

$$f(\underline{s}_1, \ldots, \underline{s}_n) = \alpha \beta^n \gamma^m, \quad \beta > 0,\ 0 \leq \gamma \leq 1$$

  where
  - α is a normalizing constant
  - β reflects the intensity of the process
  - γ describes the interactions between neighbors
    * $\gamma = 1$ gives a homogeneous Poisson process with intensity β
    * $\gamma = 0$ results in a simple inhibition process that contains no events at a distance less than or equal to δ
    * $0 < \gamma < 1$ represents some form of regularity; the smaller γ and the larger m (i.e., the number of distinct pairs of neighbors that did occur at a distance less than or equal to δ), the less likely it becomes that another event occurs in proximity to these n events
    * $\gamma > 1$ yields a chaotic and unstable process (and no clustering!)

2.4.6 Fitting Models other than CSR to Spatial Point Patterns

First use exploratory methods and subject knowledge to identify possible models.

Main questions:

- how much does the intensity vary over $\mathcal{R}$?

- how much of this can be contributed to heterogeneity and how much to spatial dependence?

Possible approaches:

- use kernel methods to estimate intensity over $\mathcal{R}$
- use K function to capture spatial dependence (assuming the process is isotropic)
- based on this exploratory work and subject knowledge, suggest a model, e.g., heterogeneous Poisson process, Poisson cluster process, etc.

New questions:

- how to estimate any unknown parameter(s) in such a model?
- how to assess the "fit" of the model to the data?

Answers:

- No general answer is possible — answers highly depend on the suggested alternative model, e.g.
- When a heterogeneous Poisson process might be a possible alternative, kernel methods could be used to estimate parameters of simple "trend" models for the first order variations in intensity (these models may include other covariates thought to influence the intensity, e.g., closeness to a city, age structure of the population, etc.).
- When we assume an isotropic process with spatial dependence might be an alternative, we might try to compare theoretical K functions of different isotropic point process models to the one estimated for the observed point pattern; this will also allow to estimate the parameters of such a model:
  Assume the proposed model incorporates a parameter vector θ. Then the theoretical K function will also depend on θ, i.e., $K(h, \theta)$. We might estimate θ by the value $\hat{\theta}$ which minimizes
  $$\int_0^{h_0} [(\hat{K}(h))^c - (K(h,\theta))^c]dh,$$
  where
  - $\hat{K}(h)$ is the estimated K function of the observed point pattern
  - h_0 and c are "tuning constants" to provide desirable estimation properties

e.g., for a Poisson cluster procss with Poisson numbers of offsprings per parent and $g(\cdot)$ a radially symmetric Normal distribution with variance σ^2, it is

$$K(h) = \pi h^2 + \frac{1 - e^{-h^2/4\sigma^2}}{\rho},$$

where ρ is the intensity of the parent CSR process and $\theta = (\rho, \sigma)$.

- Often, complex theoretical considerations and intensive computation are required to deal with models other than CSR processes.
- Simulation approaches may be required when it is difficult to obtain a theoretical K function (which is often the case due to edge effects); once again we can work with "simulation envelopes" that originiate from simulations under our alternative model.

10/16/00

2.5 Further Methods for Point Patterns

2.5.1 Analysis of Multiple Types of Events

- so far: all events that occurred were of one type
- now: two or more different types of events
- question: is the pattern in the occurrences of one type of events related to the pattern in the occurrences of another type of events, e.g., does the spatial distribution of one type of events "explain" the distribution of the other?
- more general: is there **independence** between types of events as opposed to either **attraction** or **repulsion**?
- similar to CSR which is providing the basis for one type of events, we now make independence the basis for bivariate (or multivariate) point patterns
- note that independence does *not* mean that all types of events are CSR processes (e.g., two clustered processes may be independent); instead, independence means that the overall pattern in events is made up from independent component processes, one for each of the types of events
- see literature how to generalize Poisson process, Poisson cluster process, Cox process, etc. to incorporate the idea of linked processes which either display attraction or repulsion, e.g., **linked Cox process** or **linked pair process**

Simple χ^2 Approach to Test for Independence in the Spatial Distribution of Two Types of Events:

- based on counts of the number of each event in quadrats over $\mathcal{R}$ (quadrats may be randomly scattered or a regular grid over $\mathcal{R}$)

- present results in a "presence–absence" form in a 2×2 table, consisting of the number of quadrats $c_{ij}, i, j = 1, 2$, which represent both types of events (c_{22}), neither type (c_{11}), type 1 but not type 2 (c_{21}), and type 2 but not type 1 (c_{12}):

Type 1 \ Type 2	Absent	Present
Absent	c_{11}	c_{12}
Present	c_{21}	c_{22}

- we can use a standard χ_1^2 test of independence in a 2×2 table

- test statistic is

$$X^2 = \frac{(c_{11}c_{22} - c_{12}c_{21})^2 \sum_i \sum_j c_{ij}}{(\sum_j c_{1j})(\sum_j c_{2j})(\sum_i c_{i1})(\sum_i c_{i2})}$$

- compare X^2 with percentage points of χ_1^2 distribution; significantly large values indicate lack of independence between the two patterns

- a lot of information is ignored in this test; it would be better to generalize nearest neighbor methods and the K function to be used in tests of independence for multivariate point patterns

Use of Nearest Neighbor Distribution Functions $G_{ij}(h)$ and $F_j(h)$:

- $G_{ij}(h)$ is the probability that the distance from a randomly chosen type i event to the nearest event of type j is less than or equal to h

- $F_j(h)$ is the probability that the distance from a randomly chosen point to the nearest event of type j is less than or equal to h

- estimates $\hat{F}_j(h)$ and $\hat{G}_{ij}(h)$ of $F_j(h)$ and $G_{ij}(h)$ can be obtained from a mapped point pattern using an extension of the method described in Section 2.3.3 (edge corrections can be included as well)

- if individual point processes in a multivariate point pattern are independent, the distribution of nearest neighbor distances to events of type j from an origin of measurement should be the same, no matter whether the origin of measurement is a randomly chosen point or a randomly chosen type i event ($i \neq j$), i.e.,

$$G_{ij}(h) = F_j(h),\ i \neq j,$$

if independence holds

- plot estimates $\hat{F}_j(h)$ and $\hat{G}_{ij}(h)$ against h in a single diagram with h varying between 0 and the maximum nearest neighbor event–event distance or point–event distance
- use non–parametric test (e.g., **Kolmogorov–Smirnov test**) to obtain evidence for differences in the two estimated distributions $\hat{F}_j$ and $\hat{G}_{ij}$
- other (crude) methods for bivariate point patterns exist (see literature)

Use of the Cross K Function $K_{ij}(h)$:

- we consider multivariate point patterns where λ_j defines intensity of type j events
- Definition:

$$\lambda_j K_{ij}(h) = E(\#\text{ type } j \text{ events} \leq h \text{ from an arbitrary type } i \text{ event})$$

- $K_{ii}(h)$ is just the univariate K function for events of type i
- $K_{ij}(h), i \neq j$, is called the **cross K function**
- under independence, locations of type j events are random with respect to the locations of type i events (no matter whether the distribution of either type i or type j events is clustered, regular, or random when each process is considered separately)
- theoretically:
  - $K_{ij}(h) = \pi h^2$ under independence
  - $K_{ij}(h) < \pi h^2$ under repulsion
  - $K_{ij}(h) > \pi h^2$ under attraction
- test of independence: compare the empirical cross K function $\hat{K}_{ij}(h)$ with the theoretical cross K function $K_{ij}(h)$
- in case of a bivariate point pattern with n_1 type 1 events and n_2 type 2 events over a study region $\mathcal{R}$ with area R, $K_{12}(h)$ may be estimated similar to the edge corrected univariate case discussed in Section 2.3.4:

$$\tilde{K}_{ij}(h) = \frac{R}{n_1 n_2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \frac{I_h(d_{ij})}{w_{ij}}$$

where

$$
\begin{aligned}
d_{ij} &= \text{distance between } i^{th} \text{ type 1 event and } j^{th} \text{ type 2 event in } \mathcal{R} \\
I_h(d_{ij}) &= \text{indicator function} \\
&= \begin{cases} 1, & \text{if } d_{ij} \leq h \\ 0, & \text{otherwise} \end{cases} \\
w_{ij} &= \text{proportion of the circumference of the circle centered} \\
&\quad\ \text{on } i^{th} \text{ type 1 event that is passing through} \\
&\quad\ \text{the } j^{th} \text{ type 2 event which lies within } \mathcal{R}
\end{aligned}
$$

- theoretically, $K_{12}(h) = K_{21}(h)$ but in practice it often happens that $\tilde{K}_{12}(h) \neq \tilde{K}_{21}(h)$
- we should use the following improved estimate instead:

$$\hat{K}_{12}(h) = \frac{n_2 \tilde{K}_{12}(h) + n_1 \tilde{K}_{21}(h)}{n_1 + n_2}$$

- similarly to univariate point patterns (see Section 2.3.4), we define

$$\hat{L}_{ij}(h) = \sqrt{\frac{\hat{K}_{ij}(h)}{\pi}} - h$$

- plot of $\hat{L}_{11}(h), \hat{L}_{22}(h)$, and $\hat{L}_{12}(h)$ vs h in one diagram reveals the tendency for individual component patterns to depart from CSR as well as attraction (positive peaks for the cross L function) or repulsion (negative peaks for the cross L function) — regardless of whether individual patterns are random, clustered, or regular
- to test for independence, we cannot make any further assumptions regarding any individual pattern to obtain simulation envelopes of K and L; instead, we need to preserve the individual component event patterns in any simulation run
- idea: use simulations which involve random shifts of an entire component pattern relative to the other (in case of a rectangular $\mathcal{R}$, use the toroidal idea, i.e., events moved outside of $\mathcal{R}$ reappear in $\mathcal{R}$ on the opposite side; for irregular $\mathcal{R}$, only use the subregion of $\mathcal{R}$ which lies within the shifted study region); repeating this shifting approach multiple times yields simulation envelopes for K and L

10/23/00

2.5.2 Space–Time Clustering

- when the occurrence of events has been recorded, a question of interest could be whether space and time interact, i.e., do events cluster in space when looking at a short time span only?
- this question is of interest in epidemiology, natural catastrophes (e.g., occurrence of earth quakes), etc.

Knox's Test:

- simple space–time interaction test
- consider all $n \cdot (n-1)$ ordered pairs that can be formed from a set of n observed events
- for each pair, measure the spatial separation and the time interval between the two occurrences
- call S the number of pairs that are "close in space" and T the number of pairs that are "close in time"
- we can assign each pair of events to one of the four cells of a 2×2 contingency table, depending on whether the pair is / is not close in space and / or time
- let X be the number of pairs that are both "close in space" and "close in time"
- under independence of space and time, X has an approximate Poisson distribution with mean $\frac{ST}{n(n-1)}$
- reject H_0 : "independence of space and time" in favor of H_A : "clustering in space and time" if the observed value of X is unusually large with respect to the theoretical distribution
- the analyst has to determine what "close in space" and "close in time" means in a particular application
- in practice, this analysis is usually conducted for multiple distances and multiple time intervals

Mantel's Test:

- an extension of Knox's Test
- it uses the test statistic
$$M = \sum_i \sum_{j, j \neq i} x_{ij} y_{ij}$$
where

x_{ij}	represents the distance between events i and j
y_{ij}	represents the time interval between events i and j

- note: if we assign
$$x_{ij} = \begin{cases} 1, & \text{if events are close in distance} \\ 0, & \text{otherwise} \end{cases}$$
$$y_{ij} = \begin{cases} 1, & \text{if events are close in time} \\ 0, & \text{otherwise} \end{cases}$$
we obtain the same test statistic as used in Knox's Test
- suggested separation measures x_{ij} and y_{ij} are
$$x_{ij} = \frac{1}{c_s + d_{ij}}$$
$$y_{ij} = \frac{1}{c_t + t_{ij}}$$
where

d_{ij}	=	Euclidean distance between events i and j
t_{ij}	=	time interval between events i and j
c_s, c_t	=	constants that allow to address events that have identical space or time coordinates

- see Mantel (1967) for more details

Tests based on K Functions:

- if we consider only a few time periods, we may consider events at different time periods as different types of events and base our analysis upon $K_{ij}(h)$ (i.e., the cross K function discussed in Section 2.5.1) to detect attraction (= clustering in time) as opposed to independence
- alternative approach: define bivariate space–time K function $K(h,t)$, defined as the expected number of events within distance h and time interval t of an arbitrary event, scaled by the expected number of events per unit area and per unit time
- extend the methods of Section 2.3.4 to obtain an estimate $\hat{K}(h,t)$ of $K(h,t)$
- if processes operating in space and time are independent (i.e., no space–time interaction), then $K(h,t)$ should be the product of 2 K functions, i.e., one in space, $K_S(h)$, and one in time, $K_T(t)$
- we obtain estimates $\hat{K}_S(h)$ and $\hat{K}_T(t)$ from the observed data
- we consider the differences

$$\hat{D}(h,t) = \hat{K}(h,t) - \hat{K}_S(h)\hat{K}_T(t)$$

- in case of space–time interaction, there will be raised values of $\hat{D}(h,t)$ for some h and some t (best seen in a 3D–plot of $\hat{D}(h,t)$ vs distance and time)
- we do a formal assessment of the significance of the observed values of $\hat{D}(h,t)$ by simulation:
  - perform m simulations in each of which the n events are randomly labelled with the observed n time markers
  - we obtain m estimates $\hat{D}_i(h,t), i = 1, \ldots, m$
  - we compare the sum

    $$DS = \sum_h \sum_t \hat{D}(h,t)$$

    with the empirical frequency distribution of m such sums obtained from

    $$DS_i = \sum_h \sum_t \hat{D}_i(h,t), \; i = 1, \ldots, m$$

  - an extreme value of DS when compared with the m sums DS_i indicates evidence of space–time interaction

2.5.3 Correcting for the Spatial Variations in the Population at Risk

- in some applications where the hypothesis of clustering is of interest, comparison with CSR is not sensible because we have an expectation that there will be some natural spatial variation in the intensity of the process, e.g., in an epidemiological setting, the intensity of occurrences of the disease would be expected to vary with population density
- in such cases, we want to test H_0 : "heterogeneous Poisson process with varying intensity $\lambda(\underline{s})$" vs H_A : "clustering"
- first case: suppose we know variations in the "at risk population" in the study area $\mathcal{R}$, e.g., the number of individuals y_j in each of the m census districts $\mathcal{A}_j$ that are completely covering $\mathcal{R}$, where A_j = area of $\mathcal{A}_j$
  - we obtained a kernel estimate of the event intensity over $\mathcal{R}$ (ignoring the edge correction for simplicity) in Section 2.3.2:

$$\hat{\lambda}_\tau(\underline{s}) = \sum_{i=1}^{n} \frac{1}{\tau^2} k\left(\frac{\underline{s} - \underline{s}_i}{\tau}\right),$$

    which is an estimate of "events per unit area"
  - we can obtain an estimate of "events per unit population", $\hat{\rho}_\tau(\underline{s})$, by dividing $\hat{\lambda}_\tau(\underline{s})$ by an estimate of the population density at $\underline{s}$, e.g., $\frac{y_j}{A_j}$ if $\underline{s}$ falls in area $\mathcal{A}_j$
  - assessing $\hat{\rho}_\tau(\underline{s})$ over $\mathcal{R}$ addresses the real question of interest whether there is variation in intensity per unit population
- next case: suppose we can associate the m population values y_j with m specific points $\underline{s}'_j$ in $\mathcal{R}$, e.g., the center points of the areas $\mathcal{A}_j$
  - a possible kernel estimate of population density $\hat{\lambda}'_\tau(\underline{s})$ based on these m points is

$$\hat{\lambda}'_\tau(\underline{s}) = \sum_{j=1}^{m} \frac{1}{\tau^2} k\left(\frac{\underline{s} - \underline{s}'_j}{\tau}\right) y_j,$$

    which is an estimate of "population per unit area"
  - now an estimate of "events per unit population" can be calculated as

$$\hat{\rho}_\tau(\underline{s}) = \frac{\sum_{i=1}^{n} k\left(\frac{\underline{s} - \underline{s}_i}{\tau}\right)}{\sum_{j=1}^{m} k\left(\frac{\underline{s} - \underline{s}'_j}{\tau}\right) y_j}$$

- we can use different bandwidths τ and different kernel functions $k(\cdot)$ for the numerator and denominator above
- centering population values y_j at points $\underline{s}'_j$ may not be justified — see the section on area data for better solutions

- last case: suppose there is a "control" process of m events occurring at points $\underline{s}'_j$ in $\mathcal{R}$, i.e., a process that is related to population variations (e.g., locations of doctors and hospitals in an epidemiological setting)
  - now estimate "events per unit population" as

$$\hat{\rho}_\tau(\underline{s}) = \frac{\sum_{i=1}^{n} k\left(\frac{\underline{s} - \underline{s}_i}{\tau}\right)}{\sum_{j=1}^{m} k\left(\frac{\underline{s} - \underline{s}'_j}{\tau}\right)}$$

10/30/00

3 Analysis of Spatially Continuous Data

3.1 Introduction

- interest is no longer in the patterns of locations, but in the attribute values over $\mathcal{R}$ that have been observed at a predefined and fixed set of locations, i.e., the main interest is in patterns in the attribute values (and locations are simply a subset of all possible sites where attribute values could be recorded)
- we assume the data is spatially continuous over $\mathcal{R}$, i.e., observations (or measurements) can be taken (theoretically) at any location in $\mathcal{R}$
- values have been sampled at fixed point locations $\underline{s}_i = (s_{i1}, s_{i2})$
- examples of spatially continuous data are: rainfall, temperature, air pollution, corn yield in a field ($\longrightarrow$ precision agriculture), mineral concentration, etc.

<u>Notation:</u>

$$
\begin{aligned}
\mathcal{R} &= \text{region of interest} \\
\underline{s}_i \in \mathcal{R} &= \text{point location (in some "rectangular" coordinate system)} \\
& \text{ in region of interest where the } i^{th} \text{ observation has been recorded} \\
y_i = y(\underline{s}_i) &= \text{observation (univariate or multivariate) at location } \underline{s}_i \\
\underline{y} = (y_1, \ldots, y_n) &= \text{vector (or matrix) of all observations} \\
\{Y(\underline{s}) \mid \underline{s} \in \mathcal{R}\} &= \text{spatial stochastic process that varies spatially continuously} \\
& \text{ over } \mathcal{R} \text{ and has been sampled at fixed points} \\
Y_i = Y(\underline{s}_i) &= \text{random variable related to } y_i = y(\underline{s}_i) \\
\underline{Y} = (Y_1, \ldots, Y_n) &= \text{vector (or matrix) of all random variables related} \\
& \text{ to } \underline{y} = (y_1, \ldots, y_n)
\end{aligned}
$$

3.2 Visualizing Spatially Continuous Data

Proportional Symbol Maps:

- display a symbol that represents the numeric value of an attribute at each location $\underline{s}_i$
- symbols for univariate data may be circles (where the area relates to the numeric value), bars of fixed width (where the height relates to the numeric value), etc.
- symbols for multivariate data may be ray–glyphs and arcs (see Carr, Olsen & White, 1992)
- similar visualization techniques and exploration techniques are used for spatially continuous data and area data

3.3 Exploring Spatially Continuous Data

3.3.1 Summary Statistics

- summarize spatial statistical data, but first ignore its spatial relationship, e.g.,
  - stem and leaf plots
  - boxplots
  - histograms
  - density/kernel estimates
- see Fotheringham, Brunsdon & Charlton (2000), Chapter 4, for more details

3.3.2 Dynamic Statistical Graphics/Visual Data Mining

- similar, related key words are: Exploratory Data Analysis (EDA) and Exploratory Spatial Data Analysis (ESDA)
- explore spatial data in its geographic context and link maps with statistical displays, e.g.
  - scatterplots & scatterplot matrix
  - parallel coordinate plots
  - projection pursuit
  - grand tour
  - brushing

  - linked plots & maps/linked brushing

- References: e.g., Monmonier (1988), Monmonier (1989), Carr, Olsen & White (1992), MacDougall (1992), Fotheringham, Brunsdon & Charlton (2000), Chapter 4

- Software: e.g., XGobi and the ArcView/XGobi/XploRe environment; see Swayne, Cook & Buja (1998), Cook, Majure, Symanzik & Cressie (1996), Cook, Symanzik, Majure & Cressie (1997)

3.3.3 Spatial Moving Averages

- estimate $\mu(\underline{s}) = E(Y(\underline{s}))$ by averaging the values at neighboring sampled data points

- unweighted average: calculate the unweighted average of the sample values at the k sampling points nearest to $\underline{s}$, e.g., $k = 3$

- weighted average: use

$$\hat{\mu}(\underline{s}) = \sum_{i=1}^{n} w_i(\underline{s}) y_i$$

where

$$\begin{aligned} \sum_{i=1}^{n} w_i(\underline{s}) &= 1 \\ h_i &= \text{distance from } \underline{s} \text{ to } \underline{s}_i \\ \alpha &= \text{parameter that controls the degree of smoothing} \\ w_i(\underline{s}) &\propto \begin{cases} h_i^{-\alpha} & \text{or} \\ e^{-\alpha h_i} & \end{cases} \end{aligned}$$

often, $w_i(\underline{s}) = 0$ for $h_i \geq d_{max}$

3.3.4 Kernel Estimation

- similar to Section 2.3.2 and Section 2.5.3, we can use a kernel estimate (ignoring edge correction for simplicity) to estimate $\mu(\underline{s})$:

$$\hat{\mu}_\tau(\underline{s}) = \frac{\sum_{i=1}^{n} k\left(\frac{\underline{s} - \underline{s}_i}{\tau}\right) y_i}{\sum_{i=1}^{n} k\left(\frac{\underline{s} - \underline{s}_i}{\tau}\right)}$$

where a frequent choice for $k(\cdot)$ is the quartic kernel (see Section 2.3.2)

- kernel estimation and spatial moving averages are closely related: using

$$w_i(\underline{s}) = \frac{k\left(\frac{\underline{s}-\underline{s}_i}{\tau}\right)}{\sum_{j=1}^{n} k\left(\frac{\underline{s}-\underline{s}_j}{\tau}\right)}$$

  in Section 3.3.3 shows that a kernel estimate can be expressed as a spatial moving average with appropriate weights $w_i(\underline{s})$

11/06/00

3.3.5 Dynamic Statistical Graphics for Exploring Spatial Dependence

- References: Haslett, Bradley, Craig, Unwin & Wills (1991), Majure & Cressie (1997), Cook, Symanzik, Majure & Cressie (1997)
- Notation used throughout these references:

$\underline{Z}(\underline{s}_1), \ldots, \underline{Z}(\underline{s}_n)$ = multivariate spatial data observed at spatial locations $\underline{s}_1, \ldots, \underline{s}_n$

$\underline{Z}(\underline{s}) = (Z_1(\underline{s}), \ldots, Z_p(\underline{s}))^T = p$ components of the multivariate observation at location $\underline{s}$

Spatially Lagged Scatterplots:

- useful tool for the initial exploration of spatial dependence (for one variable or between variables)
- plot a scatterplot of $Z_l(\underline{s}_i)$ vs $Z_m(\underline{s}_j)$, where $Z_l(\underline{s}_i)$ represents variable l at location $\underline{s}_i$ and $Z_m(\underline{s}_j)$ represents variable m at location $\underline{s}_j$
- in case of strong spatial dependence, points in the scatterplot that relate to nearby sampling locations will be close to the line $y = x$; dispersion around this line will increase as distance between sampling locations increases
- static version: select some lag distances (e.g., 0 km, 50 km, 100 km) and plot a lagged scatterplot for those points that have distances within a small range of the lags (e.g., 0–10 km, 40–60 km, 90–110 km)
- interactive version: brush through a univariate plot of distances that is linked with the lagged scatterplot; as distance increases, brushed points in average should move away from the line $y = x$
- link with a map view and see if some locations behave differently from other locations
- if just one variable is plotted, locations at lag 0 will all fall on the line $y = x$

- if 2 different variables l and m are plotted, locations at lag 0 will be close to the line $y = x$ in case of a high positive correlation between the 2 variables

Variogram Cloud Plots:

- useful tool for exploring the spatial dependence in a data set and looking for spatial outliers (i.e., locations that behave differently from their neighbors but are not outliers on the global scale)
- plot $\gamma_{l,m}(\underline{s}_i, \underline{s}_j) = \sqrt{\mid Z_l(\underline{s}_i) - Z_m(\underline{s}_j) \mid}$ vs d, where d is the Euclidean distance between the locations $\underline{s}_i$ and $\underline{s}_j$
- in case of strong spatial dependence, variance in the attribute differences will increase with increasing distance between locations
- locations that are close to each other but have large attribute differences may be spatial outliers
- directional variogram clouds are useful when the spatial dependence is directional (**anisotropy**); if we assume that this is the case, we plot variogram clouds for specific directions (e.g., within $\pm 10^0$ of the N/S direction or within $\pm 20^0$ of the E/W direction)
- interactive version: brush through a plot of angles that is linked with the variogram cloud plot
- link with a map view and see if some locations behave differently from other locations or if direction has an impact
- slightly different versions of variogram cloud plots exist within the literature

Multivariate Variogram Cloud Plots:

- useful tool for the exploration of asymmetric, multivariate spatial dependence in a data set
- similar to the variogram cloud plot, but values $\gamma_{l,m}$ and $\gamma_{m,l}$ are plotted separately vs d, i.e., $\gamma_{l,m}(\underline{s}_i, \underline{s}_j)$ is plotted left of 0 and $\gamma_{m,l}(\underline{s}_i, \underline{s}_j)$ is plotted right of 0 (for locations $\underline{s}_i$, $\underline{s}_j$ that lie in a certain direction)
- if $l = m$, this plot is simply the variogram cloud, mirrored at the axis $y = 0$
- if $l \neq m$, this plot is also called **cross–variogram cloud plot**
- in case of asymmetry in the data, this can be detected in the cross–variogram cloud plot and suggests a shifted lag relationship

- asymmetry often occurs in environmental data, geology, etc.

- Example:

$$
\begin{aligned}
\underline{s}_1 &= \text{city on the coast} \\
\underline{s}_2 &= \text{city in the mountains} \\
Z_1 &= \text{maximum temperature in January} \\
Z_2 &= \text{maximum temperature in August}
\end{aligned}
$$

	Z_1	Z_2
$\underline{s}_1$	50	70
$\underline{s}_2$	20	80

Assume that $\underline{s}_2$ is located east of $\underline{s}_1$.
Question: are temperature changes from January to August symmetric from E to W?
To answer this, we consider only pairs of locations where the second locations is east of the first location. Thus:

$$
\begin{aligned}
\gamma_{1,2}(\underline{s}_1, \underline{s}_2) &= \sqrt{\mid 50 - 80 \mid} = \sqrt{30}\text{: plotted left of 0} \\
\gamma_{1,2}(\underline{s}_2, \underline{s}_1) &= \sqrt{\mid 20 - 70 \mid} = \sqrt{50}\text{: not plotted since } \underline{s}_1 \text{ not east of } \underline{s}_2 \\
\gamma_{2,1}(\underline{s}_1, \underline{s}_2) &= \sqrt{\mid 70 - 20 \mid} = \sqrt{50}\text{: plotted right of 0} \\
\gamma_{2,1}(\underline{s}_2, \underline{s}_1) &= \sqrt{\mid 80 - 50 \mid} = \sqrt{30}\text{: not plotted since } \underline{s}_1 \text{ not east of } \underline{s}_2
\end{aligned}
$$

3.3.6 Covariogram and Variogram

Notation:

$$
\begin{aligned}
\{Y(\underline{s}) \mid \underline{s} \in \mathcal{R}\} &= \text{spatial stochastic process} \\
E(Y(\underline{s})) &= \mu(\underline{s}) \\
Var(Y(\underline{s})) &= \sigma^2(\underline{s}) \\
C(\underline{s}_i, \underline{s}_j) &= E((Y(\underline{s}_i) - \mu(\underline{s}_i))(Y(\underline{s}_j) - \mu(\underline{s}_j))) \\
&= \text{covariance at locations } \underline{s}_i \text{ and } \underline{s}_j \\
\rho(\underline{s}_i, \underline{s}_j) &= \frac{C(\underline{s}_i, \underline{s}_j)}{\sigma(\underline{s}_i)\sigma(\underline{s}_j)} \\
&= \text{correlation at locations } \underline{s}_i \text{ and } \underline{s}_j \\
C(\underline{s}_i, \underline{s}_i) &= \sigma^2(\underline{s}_i)
\end{aligned}
$$

Stationarity:

- Recall: a process is called stationary if $\mu(\underline{s}) = \mu$ and $\sigma^2(\underline{s}) = \sigma^2$ for all locations $\underline{s}$ and $C(\underline{s}_i, \underline{s}_j) = C(\underline{s}_i - \underline{s}_j) = C(\underline{h})$, i.e., $C(\underline{s}_i, \underline{s}_j)$ only depends on the vector difference $\underline{h}$ (i.e., the direction and distance of separation) between $\underline{s}_i$ and $\underline{s}_j$
- $C(\underline{h})$ is called the **covariance function** or **covariogram** of the stationary process
- $\rho(\underline{h})$ is called the **correlogram**
- $C(\underline{0}) = \sigma^2$

Isotropy:

- Recall: a process is called isotropic if it is stationary and $C(\underline{s}_i, \underline{s}_j)$ is only a function of the distance h between $\underline{s}_i$ and $\underline{s}_j$, i.e., the length of the vector $\underline{h}$ (and not of its direction)
- $C(\underline{s}_i, \underline{s}_j) = C(h)$
- $\rho(\underline{s}_i, \underline{s}_j) = \rho(h)$

Intrinsic Stationarity:

- weaker assumption than stationarity
- $E(Y(\underline{s} + \underline{h}) - Y(\underline{s})) = 0$
- $Var(Y(\underline{s} + \underline{h}) - Y(\underline{s})) = 2\gamma(\underline{h})$, i.e., constant variance in the differences between values at locations separated by a given distance and direction $\underline{h}$
- $2\gamma(\underline{h})$ is called the **variogram**
- $\gamma(\underline{h})$ is called the **semi–variogram** (but many authors such as Bailey/Gatrell speak of $\gamma(\underline{h})$ as the variogram)
- for a stationary process, covariogram, correlogram, and variogram are directly related:

$$\begin{aligned}
\rho(\underline{h}) &= \frac{C(\underline{h})}{\sigma^2} \\
\gamma(\underline{h}) &= \frac{Var(Y(\underline{s} + \underline{h}) - Y(\underline{s}))}{2} \\
&= \frac{1}{2}(Var(Y(\underline{s} + \underline{h})) + Var(Y(\underline{s})) - 2C(Y(\underline{s} + \underline{h}), Y(\underline{s}))) \\
&= \frac{1}{2}(\sigma^2 + \sigma^2 - 2C(\underline{h})) \\
&= \sigma^2 - C(\underline{h})
\end{aligned}$$

11/13/00

Estimation of $C(\underline{h})$ and $\gamma(\underline{h})$:

- to be able to estimate $C(\underline{h})$ or $\gamma(\underline{h})$ from an observed spatial process, we have to make some assumptions with respect to stationarity about the process — otherwise, without repetition, it is not possible to estimate second order properties

- it depends on the analyst's judgement whether to assume stationarity, intrinsic stationarity, or even isotropy

- sample estimator of the variogram based on the **method–of–moments** is

$$2\hat{\gamma}(\underline{h}) = \frac{1}{n(\underline{h})} \sum_{\underline{s}_i - \underline{s}_j = \underline{h}} (y_i - y_j)^2,$$

  where the summation is over all pairs of observed data points with a vector separation of $\underline{h}$ and where $n(\underline{h})$ is the number of these pairs

- note that for many practical situations, all $\underline{h}$'s are different — to obtain an estimate for $2\hat{\gamma}(\underline{h})$, we aggregate all $\underline{h}$'s that are within specified tolerance limits with respect to distance and direction and then calculate $2\hat{\gamma}(\underline{h})$ for this set of $\underline{h}$'s

- under isotropy:

$$2\hat{\gamma}(h) = \frac{1}{n(h)} \sum_{|\underline{s}_i - \underline{s}_j| = h} (y_i - y_j)^2,$$

  where h is the length of the separation vector $\underline{h}$

- sample estimator of the covariogram of a stationary process based on the **method–of–moments** is

$$\hat{C}(\underline{h}) = \frac{1}{n(\underline{h})} \sum_{\underline{s}_i - \underline{s}_j = \underline{h}} (y_i - \overline{y})(y_j - \overline{y}),$$

  where $\overline{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ is the mean of all observed sample values, the summation is over all pairs of observed data points with a vector separation of $\underline{h}$ and where $n(\underline{h})$ is the number of these pairs

- same simplifications under isotropy as for $2\hat{\gamma}(\underline{h})$

- in theory, it holds for a stationary process that $\gamma(\underline{h}) = \sigma^2 - C(\underline{h})$; however, in practice, usually $\hat{\gamma}(\underline{h}) \neq \hat{C}(0) - \hat{C}(\underline{h})$

- theoretically, $\gamma(0) = 0$, but due to sampling errors and small scale variability, it often is $\hat{\gamma}(0) \neq 0$; this offset at the origin is called **nugget effect**

- a variogram that only has a nugget effect, i.e., that is horizontal except at the origin, corresponds to a process with no spatial dependence

- for a stationary process, the sample variance should rise to an upper bound ($= \sigma^2$), called **sill**; the distance at which this bound is reached is called **range**

- recall from your introductory statistics class that $\overline{X}$ is an unbiased estimate for the population mean μ, i.e., $E(\overline{X}) = \mu$, but it is not very robust; a single outlier has a huge impact on $\overline{X}$; alternatively, the median $\tilde{X}$ is not unbiased, but robust, i.e., an outlier has almost no impact on $\tilde{X}$

- similarly, the method–of–moments estimators $2\hat{\gamma}(\underline{h})$ and $\hat{C}(\underline{h})$ are not very robust and can be easily distorted by outliers (that are frequent in spatial data)

- robust estimators exist (see Cressie (1993) for example), e.g., for $2\gamma(\underline{h})$:

$$2\overline{\gamma}(\underline{h}) = \frac{1}{0.457 + \frac{0.494}{n(\underline{h})}} \left(\frac{1}{n(\underline{h})} \sum_{\underline{s}_i - \underline{s}_j = \underline{h}} \sqrt{| y_i - y_j |} \right)^4$$

3.4 Modelling Spatially Continuous Data

3.4.1 Trend Surface Analysis

- a simple approach to model global or large scale variations in the mean value of a spatially continuous process

- the conventional **multiple regression model** is

$$Y(\underline{s}) = \underline{x}^T(\underline{s})\underline{\beta} + \epsilon(\underline{s})$$

where

$Y(\underline{s})$	is the rv representing the process at point $\underline{s}$
$\underline{x}^T(\underline{s})\underline{\beta}$	is the trend, i.e., the mean $E(Y(\underline{s})) = \mu(\underline{s})$ of $Y(\underline{s})$
$\epsilon(\underline{s})$	is a rv with mean 0 representing departures from the trend
$\underline{x}^T(\underline{s})$	is a $1 \times p$ vector that consists of p functions of the spatial coordinates (s_1, s_2) of a point $\underline{s}$; e.g., for a **linear trend surface** $\underline{x}^T(\underline{s}) = (1, s_1, s_2)$ and for a **quadratic surface** $\underline{x}^T(\underline{s}) = (1, s_1, s_2, s_1^2, s_2^2, s_1 s_2)$
$\underline{\beta}$	is a $p \times 1$ parameter vector that is estimated from the observed data

- standard regression assumptions are: errors $\epsilon(\underline{s})$ at different points $\underline{s}$ have constant variance σ^2 and are independent (i.e., their covariance is 0) $\Longrightarrow$ no second order effects are present in the process $Y(\underline{s})$

- we can use **ordinary least squares** to fit the model to the observed data y_i, i.e., obtain estimates $\underline{\hat{\beta}}$ of $\underline{\beta}$:

$$\begin{aligned}\underline{\hat{\beta}} &= (\underline{X}^T\underline{X})^{-1}\underline{X}^T\underline{y} \\ Var(\underline{\hat{\beta}}) &= \sigma^2(\underline{X}^T\underline{X})^{-1}\end{aligned}$$

  where

  $$\underline{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \text{ is the vector of observed values at sample sites } \begin{pmatrix} \underline{s}_1 \\ \vdots \\ \underline{s}_n \end{pmatrix}$$

  $\underline{X}$ is a $n \times p$ matrix with row vectors $\underline{x}^T(\underline{s}_i), i = 1, \ldots, n$;

  $$\text{e.g., for a linear trend surface } \underline{X} = \begin{pmatrix} 1 & s_{11} & s_{12} \\ 1 & s_{21} & s_{22} \\ \vdots & & \\ 1 & s_{n1} & s_{n2} \end{pmatrix}$$

- predicted value at sample site $\underline{s}_i$: $\hat{y}_i = \underline{x}^T(\underline{s}_i)\underline{\hat{\beta}}$

- residuals: $e_i = y_i - \hat{y}_i$

- estimate of σ^2:

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n-p}$$

- the **coefficient of determination** R^2 is a measure of the goodness of fit and indicates the proportion of variation explained by the model:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \overline{y}_i)^2}$$

  where $0 \leq R^2 \leq 1$ and $R^2 = 1$ if, and only if, $Y(\underline{s})$ can be fully explained by the regression equation

- note that one basic assumption, i.e., the $\epsilon(\underline{s})$ are independent, is often violated for spatial data and, therefore, residuals from such a regression will be spatially correlated; in addition, we often have **heteroscedasticity**, i.e., $Var(\epsilon(\underline{s}))$ is not constant over $\mathcal{R}$

- if this happens (either one or both), we cannot use conventional confidence intervals for the coefficients β_i, goodness of fit tests and confidence intervals for predictions will be invalid, and the model cannot be used for local prediction

- literature on regression provides some remedies that might resolve some of the problems, e.g., **transformation** of the response variable y_i, **weighted regression**, or **generalized least squares** (see next section)

3.4.2 Generalized Least Squares

- we relax the assumption of independence of the residuals and now assume that there exists some covariance structure

- the generalized least squares model is

$$Y(\underline{s}) = \underline{x}^T(\underline{s})\underline{\beta} + U(\underline{s})$$

  where $U(\underline{s})$ are zero mean rv's representing the errors that are not necessarily independent at different points $\underline{s}$; instead, errors have some covariance $C(\cdot, \cdot)$

- we can use **generalized least squares** to fit the model to the observed data y_i, i.e., obtain estimates $\hat{\underline{\beta}}$ of $\underline{\beta}$:

$$\begin{aligned} \hat{\underline{\beta}} &= (\underline{X}^T \underline{C}^{-1} \underline{X})^{-1} \underline{X}^T \underline{C}^{-1} \underline{y} \\ Var(\hat{\underline{\beta}}) &= (\underline{X}^T \underline{C}^{-1} \underline{X})^{-1} \end{aligned}$$

  where

  $\underline{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$ is the vector of observed values at sample sites $\begin{pmatrix} \underline{s}_1 \\ \vdots \\ \underline{s}_n \end{pmatrix}$

  $\underline{X}$ is a $n \times p$ matrix with row vectors $\underline{x}^T(\underline{s}_i), i = 1, \ldots, n$

  $\underline{C}$ is the $n \times n$ variance–covariance matrix, i.e.,
  $C_{ij} = Cov(U(\underline{s}_i), U(\underline{s}_j)), i \neq j$ and $C_{ii} = Var(U(\underline{s}_i))$

- weighted least squares are a special case where $\underline{C}$ is a diagonal matrix, i.e., $C_{ii} \neq 0$ and $C_{ij} = 0$ for $i \neq j$

- generalized least squares allow to include first and second order variation into our statistical model

- usually, the variance–covariance matrix $\underline{C}$ is unknown; we have to find methods that allow us to find estimates of the elements of $\underline{C}$ (see next section)

3.4.3 Models for Variograms

- recall: sample variograms and covariograms only make sense as estimates of the covariance structure under the assumption of some form of stationarity; if the process is not stationary, sample variogram/covariogram may be dominated by first order effects and provide no information about the second order effects
- even if there is no first order effect, i.e., the mean of the process is constant, non–stationarity means that the covariance structure does not repeat itself in different parts of $\mathcal{R}$ and, therefore, we cannot obtain estimates by averaging over different parts of $\mathcal{R}$
- therefore, we only focus on models for covariance structures of stationary processes in this lecture
- if the observed data shows a drift or a trend (e.g., the sample variogram does not rise to an upper bound), we can remove this trend by methods from Section 3.4.1 (Ordinary least squares) and use the residuals for the modelling of the covariance structure; theoretically, we could use this covariance structure to re–estimate the trend by methods from Section 3.4.2 (Generalized least squares) to obtain better residuals, and iterate this process
- in practice, using ordinary least squares will be sufficient in many cases to obtain residuals that can be used to estimate the covariance structure, in particular when using variograms (instead of covariograms)
- conditions that must hold for the covariance function of any spatial process are
  - symmetry, i.e., $C(\underline{s}_i, \underline{s}_j) = C(\underline{s}_j, \underline{s}_i)$
  - non–negative definiteness, i.e., $\sum_{i=1}^{n}\sum_{i=1}^{n} \alpha_i \alpha_j C(\underline{s}_i, \underline{s}_j) \geq 0$ for all n, for all $\alpha_1, \ldots, \alpha_n$, for all $\underline{s}_1, \ldots, \underline{s}_n$
- most commonly used (valid) variogram models for stationary (isotropic) processes are:
  - **spherical model**:

$$\gamma(h) = \begin{cases} \sigma^2 \left(\frac{3h}{2r} - \frac{h^3}{2r^3} \right), & \text{if } 0 \leq h \leq r \\ \sigma^2, & \text{otherwise} \end{cases}$$

    where r is the range and σ^2 is the sill
  - **exponential model**:

$$\gamma(h) = \sigma^2(1 - \exp(-\frac{3h}{r}))$$

– **Gaussian model**:

$$\gamma(h) = \sigma^2(1 - \exp(-\frac{3h^2}{r^2}))$$

<u>**Nugget Effect:**</u>

- a **nugget effect** can be introduced into all (three) variogram models by the inclusion of a constant term $a > 0$ which represents the discontinuity at the origin (while $\gamma(0)$ remains 0)
- most commonly used (valid) variogram models for stationary (isotropic) processes that include a nugget effect $a > 0$ are:
  - **spherical model with nugget effect**:

    $$\gamma(h) = \begin{cases} a + (\sigma^2 - a)\left(\frac{3h}{2r} - \frac{h^3}{2r^3}\right), & \text{if } 0 < h \le r \\ 0, & \text{if } h = 0 \\ \sigma^2, & \text{otherwise} \end{cases}$$

    where r is the range and σ^2 is the sill
  - **exponential model with nugget effect**:

    $$\gamma(h) = \begin{cases} a + (\sigma^2 - a)(1 - \exp(-\frac{3h}{r})), & \text{if } h > 0 \\ 0, & \text{if } h = 0 \end{cases}$$

  - **Gaussian model with nugget effect**:

    $$\gamma(h) = \begin{cases} a + (\sigma^2 - a)(1 - \exp(-\frac{3h^2}{r^2})), & \text{if } h > 0 \\ 0, & \text{if } h = 0 \end{cases}$$

- the ratio of nugget effect and sill is called **relative nugget effect**; a relative nugget effect of 100% (i.e., $a = \sigma^2$) corresponds to a complete lack of spatial dependence

<u>**Anisotropy:**</u>

- in case of **anisotropy** (the spatial dependence depends on the direction), anisotropic covariance structures may be defined, replacing h by $\underline{h}$
- **geometric anisotropy:** range changes with direction but sill remains constant
- **zonal anisotropy:** sill changes with direction but range remains constant
- mixtures of these effects are also possible

- Kaluzny et al. (1998) discuss how to address anisotropy in S+SpatialStats (see Section 4.1.4)

Variogram Fitting:

- often, variogram models are fitted to the observed sample variogram "by eye":
  - range and sill can be guessed from the sample variogram
  - the nugget effect can be obtained by fitting a straight line to the first few points in the sample variogram and extrapolating this line back to the origin
- iterate this process by making small adjustments to the fitted parameters and see how well the new curve relates to the points in the sample variogarm
- do not overfit!
- Kaluzny et al. (1998) discuss how to fit a theoretical variogram model in S+SpatialStats (see Section 4.2.2)

11/20/00

3.4.4 Simple Kriging

- primary interest lies in the prediction of the values of a spatially continuous variable
- assume model

$$Y(\underline{s}) = \underline{x}^T(\underline{s})\underline{\beta} + U(\underline{s})$$

  where $U(\underline{s})$ is a zero mean process with covariance function $C(\cdot,\cdot)$

- since $U(\underline{s})$ is not entirely unpredictable, we can do a better prediction than just using the estimated mean value $\mu(\underline{s}) = \underline{x}^T(\underline{s})\underline{\hat{\beta}}$
- use knowledge of the covariance function of the residual process $U(\underline{s})$ combined with the observed values at sample points $\underline{s}_i$, i.e., add local component to the prediction at $\underline{s}$ (in addition to the mean) — this is called **kriging**, named after the geologist D. G. Krige who developed a preliminary version of this method

Simple Kriging:

- assume first order component $\mu(\underline{s})$ is known in advance (and does not have to be estimated from the observed data) and therefore can be subtracted from the original sample observations y_i to provide residuals u_i

- assume that the zero mean process $U(\underline{s})$ has a known variance σ^2 and known covariance function $C(\cdot,\cdot)$
- since we assume $C(\cdot,\cdot)$ to be known, we do not need the assumption of stationarity
- note that no kriging method needs the assumption of stationarity; however, some form of stationarity is needed to be able to estimate $C(\cdot,\cdot)$
- simple kriging means: find an estimate $\hat{u}(\underline{s})$ for a value $u(\underline{s})$ of the rv $U(\underline{s})$ at the location $\underline{s}$, given the observed values u_i of the rv's $U(\underline{s}_i)$ at the n sample locations $\underline{s}_i$
- $\hat{y}(\underline{s}) = \mu(\underline{s}) + \hat{u}(\underline{s})$ is the predicted value of the rv $Y(\underline{s})$ at the point $\underline{s}$
- consider weighted linear combinations of the observed residuals u_i:

$$\hat{u}(\underline{s}) = \sum_{i=1}^{n} \lambda_i(\underline{s}) u(\underline{s}_i),$$

  where the weights $\lambda_i(\underline{s})$ depend on the particular location $\underline{s}$ and $\hat{u}(\underline{s})$ is the realization of the statistic $\hat{U}(\underline{s})$
- to minimize the expected mean square error $E((\hat{U}(\underline{s}) - U(\underline{s}))^2)$, we use

$$\hat{u}(\underline{s}) = \lambda^T(\underline{s})\underline{u} = \underline{c}^T(\underline{s})\underline{C}^{-1}\underline{u},$$

  where

$\underline{C}$	is the $(n \times n)$ matrix of covariances $C(\underline{s}_i, \underline{s}_j)$ between all possible pairs of the n sample sites
$\underline{c}(\underline{s})$	is an $(n \times 1)$ column vector of covariances $C(\underline{s}, \underline{s}_i)$ between the prediction point $\underline{s}$ and each of the n sample sites
$\underline{u}$	is the $(n \times 1)$ column vector of the observed values of the rv's $U(\underline{s}_i)$ at each of the n sample sites $\underline{s}_i$

- the related **minimized expected mean square error** is

$$E((\hat{U}(\underline{s}) - U(\underline{s}))^2) = \sigma^2 - \underline{c}^T(\underline{s})\underline{C}^{-1}\underline{c}(\underline{s}) = \sigma_e^2,$$

  where $\sigma^2 = Var(U(\underline{s})) = C(\underline{s}, \underline{s})$
- σ_e^2 is also called **mean squared prediction error** or **kriging error**
- assuming that the prediction error is (approximately) Normal distributed, a 95% confidence interval for the prediction $\hat{y}(\underline{s})$ is $\hat{y}(\underline{s}) \pm 1.96\sigma_e$

3.4.5 Ordinary and Universal Kriging

Ordinary Kriging:

- we assume that the process $Y(\underline{s})$ we want to predict has a constant mean, i.e., $\mu(\underline{s}) = \mu$, which relates to a trend model where $\underline{x}^T(\underline{s})$ is the scalar 1 and $\underline{\beta} = \beta_1 = \mu$
- instead of estimating μ by generalized least squares and predicting the residual process by simple kriging, i.e., performing 2 steps, ordinary kriging forms the prediction $\hat{y}(\underline{s})$ of the original process $Y(\underline{s})$ at $\underline{s}$ in one step, based on a weighted linear combination of the observed values y_i at the sample sites $\underline{s}_i$
- consider weighted linear combinations of the rv's $Y(\underline{s}_i)$:

  $$\hat{y}(\underline{s}) = \sum_{i=1}^{n} w_i(\underline{s}) y(\underline{s}_i),$$

  where the weights $w_i(\underline{s})$ are chosen such that the mean value of the statistic $\hat{Y}(\underline{s})$ that is related to $\hat{y}(\underline{s})$ is constrained to be μ
- the expected mean square error is

  $$E((\hat{Y}(\underline{s}) - Y(\underline{s}))^2) = w^T(\underline{s})\underline{C}w(\underline{s}) + \sigma^2 - 2w^T(\underline{s})\underline{c}(\underline{s}),$$

  where $w(\underline{s}) = \begin{pmatrix} w_1(\underline{s}) \\ \vdots \\ w_n(\underline{s}) \end{pmatrix}$ and $\underline{C}$ and $\underline{c}(\underline{s})$ as defined before (see Section 3.4.4)
- to minimize the expected mean square error under the constraint that the sum of the weights is 1, i.e., $w^T(\underline{s})\underline{1} = 1$ (where $\underline{1}$ is an $(n \times 1)$ column vector of 1's), we introduce the augmented matrix $\underline{C}_+$ and the augmented vectors $w_+(\underline{s})$ and $\underline{c}_+(\underline{s})$:

  $$\underline{C}_+ = \begin{pmatrix} C(\underline{s}_1, \underline{s}_1) & \dots & C(\underline{s}_1, \underline{s}_n) & 1 \\ \vdots & & \vdots & \vdots \\ C(\underline{s}_n, \underline{s}_1) & \dots & C(\underline{s}_n, \underline{s}_n) & 1 \\ 1 & \dots & 1 & 0 \end{pmatrix}$$

  $$w_+(\underline{s}) = \begin{pmatrix} w_1(\underline{s}) \\ \vdots \\ w_n(\underline{s}) \\ v(\underline{s}) \end{pmatrix}$$

$$\underline{c}_+(\underline{s}) \quad = \quad \begin{pmatrix} C(\underline{s},\underline{s}_1) \\ \vdots \\ C(\underline{s},\underline{s}_n) \\ 1 \end{pmatrix}$$

- the solution that leads to the minimum expected mean square error is

$$w_+(\underline{s}) = \underline{C}_+^{-1}\underline{c}_+(\underline{s})$$

- the related **minimized expected mean square error** (**mean square prediction error**, **kriging variance**) is

$$\sigma_e^2 = \sigma^2 - \underline{c}_+^T(\underline{s})\underline{C}_+^{-1}\underline{c}_+(\underline{s})$$

- to obtain the prediction $\hat{y}(\underline{s})$, determine $w_+(\underline{s})$, extract $w(\underline{s})$, and calculate

$$\hat{y}(\underline{s}) = w^T(\underline{s})\underline{y},$$

where $\underline{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$, i.e., the vector of the original observations y_i

- in practice, $C(\cdot,\cdot)$ is unknown and has to be derived from a globally estimated variogram model; this variogram may be estimated from the original observations y_i since we assume that $\mu(\underline{s})$ is constant over $\mathcal{R}$

- in this scenario, $C(h) = \sigma^2 - \gamma(h)$, and the ordinary kriging equations can be expressed using the variogram:

$$\begin{aligned} w_+(\underline{s}) &= \underline{\Gamma}_+^{-1}\underline{\gamma}_+(\underline{s}) \\ \sigma_e^2 &= \underline{\gamma}_+^T(\underline{s})\underline{\Gamma}_+^{-1}\underline{\gamma}_+(\underline{s}) \end{aligned}$$

where $\underline{C}_+$ and $\underline{c}_+(\underline{s})$ in our earlier equations are replaced by $\underline{\Gamma}_+$ and $\underline{\gamma}_+(\underline{s})$

Universal Kriging:

- an extension of ordinary kriging to accommodate a global trend, i.e., $\mu(\underline{s}) = \underline{x}^T(\underline{s})\underline{\beta}$

- similar to ordinary kriging, universal kriging forms the prediction $\hat{y}(\underline{s})$ directly in one step by using a weighted linear combination of the observed values y_i at the sample sites $\underline{s}_i$

- define

$$
\begin{aligned}
\underline{C}_+ &= \begin{pmatrix}
C(\underline{s}_1, \underline{s}_1) & \dots & C(\underline{s}_1, \underline{s}_n) & x_1(\underline{s}_1) & \dots & x_p(\underline{s}_1) \\
\vdots & & \vdots & \vdots & & \vdots \\
C(\underline{s}_n, \underline{s}_1) & \dots & C(\underline{s}_n, \underline{s}_n) & x_1(\underline{s}_n) & \dots & x_p(\underline{s}_n) \\
x_1(\underline{s}_1) & \dots & x_1(\underline{s}_n) & 0 & \dots & 0 \\
\vdots & & \vdots & \vdots & & \vdots \\
x_p(\underline{s}_1) & \dots & x_p(\underline{s}_n) & 0 & \dots & 0
\end{pmatrix} \\
w_+(\underline{s}) &= \begin{pmatrix}
w_1(\underline{s}) \\ \vdots \\ w_n(\underline{s}) \\ v_1(\underline{s}) \\ \vdots \\ v_p(\underline{s})
\end{pmatrix} \\
\underline{c}_+(\underline{s}) &= \begin{pmatrix}
C(\underline{s}, \underline{s}_1) \\ \vdots \\ C(\underline{s}, \underline{s}_n) \\ x_1(\underline{s}) \\ \vdots \\ x_p(\underline{s})
\end{pmatrix}
\end{aligned}
$$

- using $\underline{C}_+$ and $\underline{c}_+(\underline{s})$, we obtain a solution that is similar to the solution from ordinary kriging:

$$
\begin{aligned}
w_+(\underline{s}) &= \underline{C}_+^{-1} \underline{c}_+(\underline{s}) \\
\sigma_e^2 &= \sigma^2 - \underline{c}_+^T(\underline{s}) \underline{C}_+^{-1} \underline{c}_+(\underline{s})
\end{aligned}
$$

- again, we extract $w(\underline{s})$ from $w_+(\underline{s})$ and calculate

$$
\hat{y}(\underline{s}) = w^T(\underline{s}) \underline{y}
$$

- there are some doubts about the usefulness of universal kriging when compared to explicitly estimating a trend $\underline{x}^T(\underline{s})\underline{\beta}$ by generalized least squares; in particular, to obtain a variogram model, we have to detrend the data anyway

3.4.6 Other Types of Kriging

Robust Kriging:

- an extension that is particularly resistant to the effect of extreme or outlying observations
- see Cressie (1993), Section 3.3, for details

Disjunctive Kriging:

- instead of using a predictor that is linear in the data, disjunctive kriging looks for an optimal predictor among the class of all linear combinations of univariate functions of the data
- see Cressie (1993), Section 5.1, for details

Block Kriging:

- prediction of the average value of a spatially continuous variable over an area (or block) $\mathcal{A}$ rather than prediction at a specific point location $\underline{s}$
- see Bailey/Gatrell (1995), Section 6.2.1, for details

Co–Kriging:

- include one (or more) secondary variables recorded at the same sites in the prediction of the primary variable of interest
- particularly useful if observations of the primary variable are missing at some locations but for the secondary variables the observations exist
- see Bailey/Gatrell (1995), Section 6.2.2, and Cressie (1993), Section 3.2.3, for details

11/27/00

4 Analysis of Area Data

4.1 Introduction

- attribute of interest does not vary spatially continuously over $\mathcal{R}$, but has values only within a fixed set of areas (or zones) over $\mathcal{R}$
- sites where data occurs may constitute a regular lattice (e.g., plots in agricultural field trials, pixels in remote sensing) or may consist of irregular areal units (e.g., census tracts, health service areas —HSA's)
- other terms for **area data** are **regional data** or **lattice data**
- main interest is not prediction since there are no values "between areas" and we usually have observed values in all possible areas (unless we have missing data, e.g., due to instrument failure in case of remote sensing data or unreported data for HSA's)
- instead, the main interest is in the detection and explanation of spatial patterns or trends in the area values

Notation:

$$
\begin{aligned}
\mathcal{R} &= \text{region of interest} \\
\mathcal{A}_i \in \mathcal{R} &= \text{area in region of interest where the } i^{th} \text{ observation has been} \\
& \text{ recorded, } i = 1, \ldots, n \text{ (finite), } \bigcup_{i=1}^{n} \mathcal{A}_i = \mathcal{R},\ \mathcal{A}_i \cap \mathcal{A}_j = \emptyset, i \neq j \\
y_i = y(\mathcal{A}_i) &= \text{observation (univariate or multivariate) in area } \mathcal{A}_i \\
\underline{y} = (y_1, \ldots, y_n) &= \text{vector (or matrix) of all observations} \\
\{Y(\mathcal{A}_i) \mid \mathcal{A}_i \subset \mathcal{R}, i = 1, \ldots, n\} &= \text{spatial stochastic process that varies over areas } \mathcal{A}_i \text{ where} \\
& \text{ data for all areas has been recorded} \\
Y_i = Y(\mathcal{A}_i) &= \text{random variable (or vector) related to } y_i = y(\mathcal{A}_i) \\
\underline{Y} = (Y_1, \ldots, Y_n) &= \text{vector (or matrix) of all random variables} \\
& \text{ related to } \underline{y} = (y_1, \ldots, y_n)
\end{aligned}
$$

Note:

- observations are not a sample of the possible locations at which attribute values could occur, but, instead, they are one possible realization of values of the attribute at all locations

- theoretically, this represents a sample from a "super population" of all possible realizations of the process over the areas $\mathcal{A}_1, \ldots, \mathcal{A}_n$ that might ever occur

- area data may be based on spatially continuous data (e.g., air pollution) where the average for an entire area has been reported or on spatial point patterns (e.g., number of deaths due to a particular disease) where disclosure restrictions do not permit to indicate individual cases but only totals in a given area

4.2 Visualizing Area Data

- similar visualization techniques and exploration techniques are used for spatially continuous data and area data; here we list methods that are unique to area data; for other methods, see Sections 3.2 and 3.3

Choropleth Maps:

- maps where each of the areas $\mathcal{A}_i$ is colored or shaded according to a discrete scale (typically 6 to 10 levels) based on the value of the attribute of interest within that area

- a major problem is the selection of appropriate classes (see Monmonier, 1988; Monmonier, 1996)

- another problem is the selection of appropriate colors; newer research even contradicts previous research which kind of color schemes should be used (see Brewer, 1997; Brewer, MacEachren, Pickle, Herrmann, 1997)

- extensions of choropleth maps exist, e.g., to represent the reliability of the statistical data (see MacEachren, Brewer, Pickle, 1998)

- without careful selection of classes and colors, choropleth maps can be very misleading (although they are widely used and easily obtainable within many GIS's and statistical packages)

- small regions (e.g., Washington D.C. in the US; Luxembourg, Andorra, and Liechtenstein in Europe) are almost invisible on a choropleth map

- if areas are based on arbitrary boundaries (e.g., administrative units), any observed pattern across the areas may be influenced by the selection of the boundaries as well as the underlying spatial distribution of the attribute values; this is known as the **modifiable area unit problem**

Micromaps (Linked Micromap Plots):

- link multiple maps with one (or more) columns of statistical displays
- no need to place data in classes since the original data is shown (also possible to display confidence intervals, time series, boxplots, etc. in the statistical displays)
- choice of color is simple since usually only 5 to 8 areas are highlighted on each micromap and thus only 5 to 8 colors (or gray–shades) are needed
- less emphasis on "correct" map — small regions are enhanced in the map and regions far away from the main area are shifted towards the main area; outlines and boundaries of regions are smoothed
- references, e.g., Carr, Pierson (1996), Carr, Olsen, Courbois, Pierson, Carr (1998), Symanzik, Carr, Axelrad, Wang, Wong, Woodruff (1999)
- colored on–line versions of micromap papers available on the Web:
  - Carr et al. (1996, 1998) in the Statistical Computing & Statistical Graphics Newsletter at
    `http://cm.bell-labs.com/cm/ms/who/cocteau/newsletter/index.html`
  - Symanzik at al. (1999) at
    `http://www.math.usu.edu/~symanzik/papers/JSpapers.html`

Cartogram (Density Equalized Map):

- transform regions in such a way to make their areas proportional to the corresponding attribute values while maintaining the spatial contiguity of the regions
- see Bailey/Gatrell (1995), Section 7.3, for details

4.3 Exploring Area Data

4.3.1 Proximity Measures for Area Data

- one crude approach woule be the use of Euclidean distances between the **centroids**, i.e., the geographical centers, of each of the areas
- this may ignore some of the aspects of the spatial nature of the areas, e.g., the centroids may not even fall into the corresponding areas
- a general approach is to define an $(n \times n)$ **spatial proximity matrix** $\underline{W}$ where elements W_{ij} represent a measure of the spatial proximity of areas $\mathcal{A}_i$ and $\mathcal{A}_j$

- possible choices for W_{ij} are, for example:

$$W_{ij} = \begin{cases} 1, & \text{centroid of } \mathcal{A}_j \text{ is one of the } k \text{ nearest centroids to that of } \mathcal{A}_i \\ 0, & \text{otherwise} \end{cases}$$

$$W_{ij} = \begin{cases} 1, & \text{centroid of } \mathcal{A}_j \text{ is within some specified distance of that of } \mathcal{A}_i \\ 0, & \text{otherwise} \end{cases}$$

$$W_{ij} = \begin{cases} d_{ij}^{\gamma}, & \text{if inter–centroid distance } d_{ij} < \delta \ (\delta > 0, \gamma < 0) \\ 0, & \text{otherwise} \end{cases}$$

$$W_{ij} = \begin{cases} 1, & \mathcal{A}_j \text{ shares a common boundary with } \mathcal{A}_i \\ 0, & \text{otherwise} \end{cases}$$

$$W_{ij} = \frac{l_{ij}}{l_i}, \text{ where } l_{ij} \text{ is the length of the common boundary between } \mathcal{A}_i \text{ and } \mathcal{A}_j \text{ and } l_i \text{ is the perimeter of } \mathcal{A}_i$$

$$W_{ij} = \begin{cases} 1, & \mathcal{A}_j \text{ shares a common boundary with } \mathcal{A}_i \\ \frac{1}{x+1}, & \text{otherwise, where } x \text{ is the minimum number of areas} \\ & \text{to be passed on a path from } \mathcal{A}_i \text{ to } \mathcal{A}_j \end{cases}$$

- $\underline{W}$ does not have to be symmetric
- we can specify proximity measures of different orders (called **spatial lags**), e.g., $\underline{W}^{(1)}$ represents spatial proximity of the areas at spatial lag 1, e.g., neighbors within some distance band or "first" nearest neighbors and $\underline{W}^{(2)}$ represents spatial proximity of the areas at spatial lag 2, e.g., neighbors within the next distance band or "second" nearest neighbors, etc.

4.3.2 Spatial Moving Averages

- a method that allows us to explore how the mean value μ_i of the attribute of interest varies across the region of interest
- $$\hat{\mu}_i = \frac{\sum_{j=1}^{n} W_{ij} y_j}{\sum_{j=1}^{n} W_{ij}}$$
- if $\underline{W}$ has been standardized such that row sums equal 1, this reduces to

$$\hat{\mu}_i = \sum_{j=1}^{n} W_{ij} y_j$$

4.3.3 Median Polish

- a technique resistant to extreme values or outliers that allows us to identify broad spatial trends in the mean values μ_i
- data y_i must originate from a regular grid of the areas (e.g., agricultural field trials, pixels in remote sensing)
- use y_{ij} (and μ_{ij}) to denote data (and mean values) in the i^{th} row and j^{th} column of the lattice, $i = 1, \ldots, r,\; j = 1, \ldots, s$
- treat y_{ij}'s as if they are cell entries in a two–way table and obtain an additive decomposition of each cell as

  $$y_{ij} = \mu + \mu_i + \mu_j + \epsilon_{ij}$$

  where

  $$\begin{aligned} \mu &= \text{fixed overall effect} \\ \mu_i &= \text{fixed row effect} \\ \mu_j &= \text{fixed column effect} \\ \epsilon_{ij} &= \text{random error} \end{aligned}$$

  and

  $$\mu_{ij} = \mu + \mu_i + \mu_j = \text{ overall mean}$$

- we could use an ordinary analysis of variance approach to obtain $\hat{\mu}, \hat{\mu}_i$, and $\hat{\mu}_j$ based on the row and column means $\overline{y}_{i.}$ and $\overline{y}_{.j}$; however, an approach that is more robust to extreme values and uses medians instead of means is favored for spatial data
- median polish algorithm:
  - add an extra $(r+1)^{th}$ row and an extra $(s+1)^{th}$ column to the table and initialize the new cells with 0
  - each cell value is replaced by the difference between y_{ij} and the median of its row ($m = \text{median}\{y_{i1}, \ldots, y_{is}\}$):

    $y_{ij,new} = y_{ij,old} - m$
  - $y_{i,s+1}$ is set equal to its previous value plus the median of the row:

    $y_{i,s+1,new} = y_{i,s+1,old} + m$
  - repeat the previous 2 steps, but now working through the columns (including the $(s+1)^{th}$ column) and adding the medians to the $(r+1)^{th}$ column

- repeat this procedure, working through the rows again, then through the columns again, etc., until no cell value changes by more than some small tolerance value $\delta > 0$
- during this process, the initial table is replaced by a table of residuals; the extra column contains robust estimates of the μ_i and the extra row contains robust estimates of the μ_j; the cell $y_{r+1,s+1}$ contains a robust estimate of μ; thus, $\hat{\mu}_{ij} = \hat{\mu} + \hat{\mu}_i + \hat{\mu}_j$
- example with $r = 3$ rows and $s = 2$ columns:

	3	5	0	4 (median row 1)
	7	9	0	8 (median row 2)
	6	20	0	13 (median row 3)
	0	0	0	

$\longrightarrow$	-1	1	4	
	-1	1	8	
	-7	7	13	
	0	0	0	
	-1	1	8	(median column 1, 2, 3)

$\longrightarrow$	0	0	-4	0 (median row 1)
	0	0	0	0 (median row 2)
	-6	6	5	0 (median row 3)
	-1	1	8	

$\longrightarrow$	0	0	-4	
	0	0	0	
	-6	6	5	
	-1	1	8	
	0	0	0	(median column 1, 2, 3)

no further change during the last step — median polish done

12/04/00

4.3.4 Kernel Estimation

- exact kernel estimates would use the specific shape of the areas $\mathcal{A}_i$; since this is computationally intensive, many kernel methods avoid using information on the shapes of the $\mathcal{A}_i$'s and typically associate each observation y_i with a point location $\underline{s}_i$, e.g., the centroid of the area $\mathcal{A}_i$ or a major center of population in the area $\mathcal{A}_i$
- in this case, kernel estimates for spatially continuous data and area data are identical:

$$\hat{\mu}_\tau(\underline{s}) = \frac{\sum_{i=1}^{n} k\left(\frac{\underline{s}-\underline{s}_i}{\tau}\right) y_i}{\sum_{i=1}^{n} k\left(\frac{\underline{s}-\underline{s}_i}{\tau}\right)}$$

- interpretation of $\hat{\mu}_\tau(\underline{s})$ is that this value does not represent a value at a single point $\underline{s}$ but in a small area around $\underline{s}$

4.3.5 Spatial Correlation and Correlogram

- methods to explore the second order properties of the spatial process
- variogram and covariogram could be used under the assumption that the attribute values are located at some point within each $\mathcal{A}_i$, e.g., the centroid
- better measures exist that make use of the spatial proximity matrix $\underline{W}$, introduced in Section 4.3.1
- most frequently used methods estimate **spatial correlation** (or, more precisely, **spatial autocorrelation** since this involves the same variable at different spatial locations) rather than **spatial covariance**
- two widely used measures for spatial autocorrelation are:
  - **Moran's I:** for a spatial proximity matrix $\underline{W}$, the spatial autocorrelation in the attribute values y_i is calculated as:

$$I = \frac{n \sum_{i=1}^{n} \sum_{j=1}^{n} W_{ij}(y_i - \overline{y})(y_j - \overline{y})}{\left(\sum_{i=1}^{n} (y_i - \overline{y})^2\right) \left(\sum_{i=1}^{n} \sum_{j=1, j \neq i}^{n} W_{ij}\right)}$$

- **Geary's C:** for a spatial proximity matrix $\underline{W}$, the spatial autocorrelation in the attribute values y_i is calculated as:

$$C = \frac{(n-1)\sum_{i=1}^{n}\sum_{j=1}^{n} W_{ij}(y_i - y_j)^2}{2\left(\sum_{i=1}^{n}(y_i - \overline{y})^2\right)\left(\sum_{i=1}^{n}\sum_{j=1, j\neq i}^{n} W_{ij}\right)}$$

- note that Moran's I resembles the covariogram and Geary's C resembles the variogram introduced in Section 3.3.6

- neither I nor C is constrained to lie in the interval $[-1, 1]$ (however, each could be forced into this interval by dividing through its maximum possible bound)

- I and C are approximately Normal distributed with means $\mu_I = -\frac{1}{n-1}$ and $\mu_C = 1$; ranges r_I and r_C and variances σ_I^2 and σ_C^2 are functions of the W_{ij}'s; always, $C \geq 0$

- we can use z–tests for H_0 : "no spatial autocorrelation" as follows: calculate $\frac{I-\mu_I}{\sigma_I}$ or $\frac{C-\mu_C}{\sigma_C}$ and see whether this is significantly different from 0, e.g., using the SIDS data ($n = 100$) from Kaluzny et al. (1998), page 125–126, we get:

$$\frac{I - \mu_I}{\sigma_I} = \frac{0.2287 - (-\frac{1}{100})}{0.08786} = 2.717$$
$$P(\mid Z \mid \geq 2.717) \approx 0.0066 < 0.05 \text{ (where } Z \sim N(0,1))$$

  and

$$\frac{C - \mu_C}{\sigma_C} = \frac{0.7439 - 1}{0.1136} = -2.254$$
$$P(\mid Z \mid \geq 2.254) \approx 0.0244 < 0.05 \text{ (where } Z \sim N(0,1)),$$

  i.e., both tests are significant at the 5% level, rejecting H_0

- instead of using z–tests, we could also use permutation tests (see Bailey/Gatrell (1995), page 281, and Kaluzny et al. (1998) for this feature in S+SpatialStats)

- interpretation of the spatial autocorrelation tests:
  - non–significant result: no significant result for the selected spatial proximity matrix $\underline{W}$; if other meaningful $\underline{W}$'s exist, rerun tests for these $\underline{W}$'s
  - positive autocorrelation could be caused by a trend in the data

Correlogram:

- generalization of Moran's I or Geary's C to estimate spatial autocorrelation at different spatial lags
- based on spatial proximity matrix for lag k, $\underline{W}^{(k)}$
- e.g., Moran's I to estimate spatial autocorrelation at lag k:

$$I^{(k)} = \frac{n \sum_{i=1}^{n} \sum_{j=1}^{n} W_{ij}^{(k)} (y_i - \overline{y})(y_j - \overline{y})}{\left(\sum_{i=1}^{n} (y_i - \overline{y})^2 \right) \left(\sum_{i=1}^{n} \sum_{j=1, j \neq i}^{n} W_{ij}^{(k)} \right)}$$

  where $W_{ij}^{(k)}$ are the elements of $\underline{W}^{(k)}$
- plot spatial autocorrelation at lag k vs k, $k = 1, 2, 3, \ldots$
- note that values at neighboring lags of a correlogram are highly correlated; also, correlation at larger lags is in part a function of correlation at smaller lags; possible solution: partial correlogram

4.4 Modelling Area Data

4.4.1 Non–Spatial Regression Models

- technique similar to trend surface analysis (see Section 3.4.1) for spatially continuous data
- using one model for all n areas together:

$$\underline{Y} = \underline{X}\underline{\beta} + \underline{\epsilon}$$

  where

$$\underline{Y} = \begin{pmatrix} Y_1 \\ \vdots \\ Y_n \end{pmatrix} = \begin{pmatrix} Y(A_1) \\ \vdots \\ Y(A_n) \end{pmatrix}, \text{ i.e., a } (n \times 1) \text{ random vector}$$

$\underline{X}$ = $(n \times p)$ matrix of the values of p explanatory variables

$\underline{\epsilon}$ = $(n \times 1)$ vector of zero–mean rv's, representing the errors
(assuming $Var(\epsilon_i) = \sigma^2, Cov(\epsilon_i, \epsilon_j) = 0, i \neq j$,
i.e., errors are independent and have constant variance)

$\underline{\beta}$ = $(p \times 1)$ parameter vector that is estimated from the observed data

- we can use **ordinary least squares** to fit the model to the observed data y_i, i.e., obtain estimates $\underline{\hat{\beta}}$ of $\underline{\beta}$:

$$\begin{aligned} \underline{\hat{\beta}} &= (\underline{X}^T \underline{X})^{-1} \underline{X}^T \underline{y} \\ Var(\underline{\hat{\beta}}) &= \sigma^2 (\underline{X}^T \underline{X})^{-1} \end{aligned}$$

where $\underline{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$ is the vector of observed values in the areas $\mathcal{A}_i$

- see Section 3.4.1 for more details

4.4.2 Generalized Least Squares

- similar to Section 3.4.2
- generalized least squares model is

$$\underline{Y} = \underline{X}\underline{\beta} + \underline{U}$$

where $\underline{Y}, \underline{X}$, and $\underline{\beta}$ are as in Section 4.4.1 and $\underline{U}$ is a $(n \times 1)$ zero–mean vector of rv's, representing the errors, with variance–covariance matrix $\underline{C}$, i.e., $E(\underline{U}) = 0$ and $E(\underline{U}\underline{U}^T) = \underline{C}$

- since $\underline{C}$ is typically unknown, we have to find methods that allow us to find the elements of $\underline{C}$
- see Section 3.4.2 for more details

4.4.3 Spatial Regression Models

- instead of specifying $\underline{C}$ directly as in the previous section, we can specify it indirectly by an **interaction scheme**
- we include some extra parameters (that need to be estimated) in our model that describe relationships between variables and their neighboring values
- no need to assume stationarity for the second order component
- covariance structures do not have to be smooth functions of the Euclidean distance

Simultaneous Autoregressive Models (SAR):

- also called **autocorrelated errors model**
-
$$\begin{aligned}\underline{Y} &= \underline{X\beta} + \underline{U}\\ \underline{U} &= \rho\underline{WU} + \underline{\epsilon}\end{aligned}$$
  where $E(\underline{\epsilon}) = 0$ and $E(\underline{\epsilon\epsilon}^T) = \sigma^2\underline{I}$ and the proximity matrix $\underline{W}$ has been standardized to have row sums equal to 1
- one interaction parameter ρ has been included
- model can be rewritten as
$$\begin{aligned}\underline{Y} &= \underline{X\beta} + \rho\underline{WU} + \underline{\epsilon}\\ &= \underline{X\beta} + \rho\underline{W}(\underline{Y} - \underline{X\beta}) + \underline{\epsilon}\\ &= \underline{X\beta} + \rho\underline{WY} - \rho\underline{WX\beta} + \underline{\epsilon},\end{aligned}$$
  i.e., Y_i in the area $\mathcal{A}_i$ depends
  - on the surrounding values Y_j, $(j \neq i)$, through the term $\rho\underline{WY}$
  - on the general trend through $\underline{X\beta}$
  - on neighboring trend values through $\rho\underline{WX\beta}$
- covariance structure $\underline{C}$ can be expressed as
$$\begin{aligned}\underline{C} &= E(\underline{UU}^T)\\ &= \sigma^2((\underline{I} - \rho\underline{W})^T(\underline{I} - \rho\underline{W}))^{-1}\end{aligned}$$
  where $-1 \leq \rho \leq 1$
- see Bailey/Gatrell (1995), page 286, for an iterative approach to estimate $\underline{\beta}, \sigma^2$, and ρ

Conditional Autoregressive Models (CAR):

- more complex interaction schemes than SAR
-
$$\begin{aligned}\underline{Y} &= \underline{X\beta} + \underline{U}\\ \underline{U} &= \rho_1\underline{W}^{(1)}\underline{U} + \rho_2\underline{W}^{(2)}\underline{U} + \ldots + \underline{\epsilon}\end{aligned}$$
  where $\underline{W}^{(k)}$ is the proximity matrix related to spatial lag k

Other Departures from SAR:

- simplification of SAR by dropping the $\rho \underline{W} \underline{X} \underline{\beta}$ term:

$$\underline{Y} = \underline{X}\underline{\beta} + \rho \underline{W}\underline{Y} + \underline{\epsilon},$$

  i.e., second order variation expressed as a pure **autoregressive effect**

- further simplification by dropping the first order term $\underline{X}\underline{\beta}$:

$$\underline{Y} = \rho \underline{W}\underline{Y} + \underline{\epsilon},$$

  i.e., a **pure autoregressive model** where all variation in the process is captured through second order variation

Moving Average (MA) Model:

$$\underline{Y} = (\underline{I} + \rho \underline{W})\underline{\epsilon}$$

Note:
see Kaluzny et al. (1998), Section 5.3, how to fit a SAR, CAR, or MA spatial regression model in S+SpatialStats via the "slm" (spatial linear model) command

5 Useful Future Statistics Courses

Stat 5100	Linear Regression and Time Series	(H)
Stat 5110	Theory of Linear Models	(H)
Stat 5600	Applied Multivariate Statistics	(M)
Math 5710	Introduction to Probability	(L)
Math 5720	Introduction to Mathematical Statistics	(L)
Stat 6560	Graphical Methods	(M)

Note:

- courses labelled (H) might be of general interest for everyone who is working with spatial and non–spatial data; some of the topics discussed in Stat 5100 and Stat 5110 that are related to Stat 5810 are regression models, ordinary least squares, generalized least squares, etc.
- courses labelled (M) might be of medium interest for everyone who is working with spatial data but might be particulary useful for people that are working with multivariate data (more than one observation) or for studies where little to nothing about the data is known in advance; some of the topics discussed in Stat 6560 that are related to Stat 5810 are brushing, linked brushing, grand tour, parallel coordinate plots, etc.
- courses labelled (L) might be of low interest for *applied* spatial statistics but are essential as a basis to understand theoretical concepts of spatial statistics such as discussed in Cressie (1993); some of the topics discussed in Math 5710 and Math 5720 that are related to Stat 5810 are different distributions such as exponential and Normal distributions and their properties, properties of E, Var, and Cov, maximum likelihood estimation, etc.

— The End —

A References

Recommended Textbook

- Bailey, T. C., Gatrell, A. C. (1995): *Interactive Spatial Data Analysis*, Longman, Essex, England & Wiley, New York, NY.
  [on reserve: QA 278.2 .B35 1995 (ask for book and floppy)]

Recommended Book for S+SpatialStats

- Kaluzny, S. P., Vega, S. C., Cardoso, T. P., Shelly, A. A. (1998): *S+SpatialStats — User's Manual for Windows and UNIX*, Springer, New York, NY.
  [on reserve: QA278.2 .S18 1998]

Additional Textbooks on Spatial Statistics

- Cressie, N. A. C. (1993): *Statistics for Spatial Data (Revised Edition)*, Wiley, New York, NY.
  [on reserve: QA278.2 .C75 1993]
- Diggle, P. J. (1983): *Statistical Analysis of Spatial Point Patterns*, Academic Press, London, England.
  [on reserve: QH323.5 .D53x]
- Fotheringham, A. S., Brunsdon, C., Charlton, M. (2000): *Quantitative Geography — Perspectives on Spatial Data Analysis*, Sage, London, England.
  [recommended reading: Chapters 3 & 4]
- Isaaks, E. H., Srivastava, R. M. (1989): *An Introduction to Applied Geostatistics*, Oxford, New York, NY.
  [on reserve: QE33.2 .M3I83 1989]
- Monmonier, M. (1996): *How to Lie with Maps (Second Edition)*, University of Chicago Press, Chicago, IL.
  [recommended reading: pp. 40–42]
- Ripley, B. D. (1981): *Spatial Statistics*, Wiley, New York, NY.
  [on reserve: QA278.2 .R56]
  [recommended reading: Chapter 8]
- Wackernagel, H. (1998): *Multivariate Geostatistics (2nd, Completely Revised Edition)*, Springer, Berlin, Germany.
  [on reserve: QE 33.2 .S82 W33 1995 (1995 edition)]

Additional Books on S and S–Plus

- Becker, R. A., Chambers, J. M., Wilks, A. R. (1988): *The New S Language*, Wadsworth & Brooks/Cole, Pacific Grove, CA.
- Chambers, J. M. (1998): *Programming with Data — A Guide to the S Language*, Springer, New York, NY.
[on reserve: QA 76.73 .S15 C483 1998]
- Chambers, J. M., Hastie, T. J., Eds. (1993): *Statistical Models in S*, Chapman & Hall, New York, NY.
- Krause, A., Olson, M. (2000): *The Basics of S and S–Plus (Second Edition)*, Springer, New York, NY.
[on reserve: QA276.4 .K73 1997 (1997 edition)]
- Spector, P. (1994): *An Introduction to S and S–Plus*, Duxbury Press, Belmont, CA.
[on reserve: QA76.73 .S15 S63 1994]
- Venables, W. N., Ripley, B. D. (1999): *Modern Applied Statistics with S–Plus (Third Edition)*, Springer, New York, NY.
[on reserve: QA276.4 .V46 1997 (1997 edition)]

Articles etc.

- Chambers Receives Prestigious ACM Award (April 1999), Amstat News, p. 21.
- ASA Receives Gift to Endowed Statistical Computing Prize (July 1999), Amstat News, p. 9.
- Brewer, C. A. (1997): Spectral Schemes: Controversial Color Use on Maps, *Cartography and Geographic Information Systems*, Vol. 24, No. 4, pp. 203–220.
- Brewer, C. A., MacEachren, A. M., Pickle, L. W., Herrmann, D. (1997): Mapping Mortality: Evaluating Color Schemes for Choropleth Maps, *Annals of the Association of American Geographers*, Vol. 87, No. 3, pp. 411–438.
- Carr, D. B., Olsen, A. R., White, D. (1992): Hexagon Mosaic Maps for Display of Univariate and Bivariate Geographical Data, *Cartography and Geographic Information Systems*, Vol. 19, No. 4, pp. 228-236, 271.
- Carr, D. B., Pierson, S. M. (1996): Emphasizing Statistical Summaries and Showing Spatial Context with Micromaps, *Statistical Computing & Statistical Graphics Newsletter*, Vol. 7, No. 3, pp. 16–23.

- Carr, D. B., Olsen, A. R., Courbois, J.–Y. P., Pierson, S. M., Carr, D. A. (1998): Linked Micromap Plots: Named and Described, *Statistical Computing & Statistical Graphics Newsletter*, Vol. 9, No. 1, pp. 24–32.

- Cook, D., Majure, J. J., Symanzik, J., Cressie, N. (1996): Dynamic Graphics in a GIS: Exploring and Analyzing Multivariate Spatial Data Using Linked Software, *Computational Statistics*, Vol. 11, No. 4, pp. 467–480.

- Cook, D. Symanzik, J., Majure, J. J., Cressie, N. (1997): Dynamic Graphics in a GIS: More Examples Using Linked Software, *Computers & Geosciences*, Vol. 23, No. 4, pp. 371–385.

- Gotway, C. A., Hartford, A. H. (1996): Geostatistical Methods for Incorporating Auxiliary Information in the Prediction of Spatial Variables, *Journal of Agricultural, Biological, and Environmental Statistics*, Vol. 1, No. 1, pp. 17–39.

- Haslett, J., Bradley, R., Craig, P., Unwin, A., Wills, G. (1991): Dynamic Graphics for Exploring Spatial Data with Application to Locating Global and Local Anomalies, *The American Statistician*, Vol. 45, No. 3, pp. 234–242.

- MacDougall, E. B. (1992): Exploratory Analysis, Dynamic Statistical Visualization, and Geographic Information Systems, *Cartography and Geographic Information Systems*, Vol. 19, No. 4, pp. 237–246.

- MacEachren, A. M., Brewer, C. A., Pickle, L. W. (1998): Visualizing Georeferenced Data: Representing Reliability of Health Statistics, *Environment and Planning A*, Vol. 30, pp. 1547–1561.

- Majure, J. J., Cressie, N. (1997): Dynamic Graphics for Exploring Spatial Dependence in Multivariate Spatial Data, *Geographical Systems*, Vol. 4, No. 2, pp. 131–158.

- Monmonier, M. (1988): Geographical Representations in Statistical Graphics: A Conceptual Framework, *1988 Proceedings of the Section on Statistical Graphics*, American Statistical Association, Alexandria, VA, pp. 1–10.

- Monmonier, M. (1989): Geographic Brushing: Enhancing Exploratory Analysis of the Scatterplot Matrix, *Geographical Analysis*, Vol. 21, No. 1, pp. 81–84.

- Papritz, A., Stein, A. (1999): Spatial Prediction by Linear Kriging, In: Stein, A., Van der Meer, F., Gorte, B. (Eds.), *Spatial Statistics for Remote Sensing*, Kluwer, Dordrecht, Netherlands, pp. 83–113.

- Rowlingson, B. S., Diggle, P. J. (1993): SPLANCS: Spatial Point Pattern Analysis Code in S–Plus, *Computers & Geosciences*, Vol. 19, No. 5, pp. 627–655.

- Swayne, D. F., Cook, D., Buja, A. (1998): XGobi: Interactive Dynamic Data Visualization in the X Window System, *Journal of Computational and Graphical Statistics*, Vol. 7, No. 1, pp. 113–130.

- Symanzik, J., Kötter, T., Schmelzer, S., Klinke, S., Cook, D., Swayne, D. F. (1998): Spatial Data Analysis in the Dynamically Linked ArcView/XGobi/XploRe Environment, *Computing Science and Statistics*, Vol. 29, No. 1, pp. 561–569.

- Symanzik, J., Carr, D. B., Axelrad, D. A., Wang, J., Wong, D., Woodruff, T. J. (1999): Interactive Tables and Maps — A Glance at EPA's Cumulative Exposure Project Web Page, *1999 Proceedings of the Section on Statistical Graphics*, American Statistical Association, Alexandria, VA, pp. 94–99.

B Computer Lab Lectures

B.1 Introduction to S and S–Plus

http://www.math.usu.edu/~symanzik/teaching/2000_spatial/splus/sintro

```
Stat 5810 Applied Spatial Statistics                              Fall 2000


                      Introduction to S and S-Plus
                      ============================

           Wed 9/13/00, Mon 9/18/00, Wed 9/20/00 & Mon 9/25/00


Dr. Juergen Symanzik


The material presented in this introduction is an update from
a workshop originally held in January 1997. The original material
is available at
   http://math.usu.edu/~symanzik/teaching/1997_scourse/sintro


------------------------------------------------------------------------------

References:

Becker, R. A., Chambers, J. M., and Wilks, A. R. (1988): The New S Language,
Wadsworth & Brooks/Cole, Pacific Grove, CA.

Chambers, J. M. (1998): Programming with Data - A Guide to the S Language,
Springer, New York.
[On reserve]

Chambers, J. M., Hastie, T. J. (1993): Statistical Models in S,
Chapman & Hall, New York, London.

Kaluzny, S. P., Vega, S. C., Cardoso, T. P., Shelly, A. A. (1998):
S+SpatialStats - User's Manual for Windows and UNIX, Springer, New York.
[On reserve]

Krause, A., Olson, M. (2000): The Basics of S and S-Plus, 2nd Edition,
Springer, New York.
[1st Edition on reserve]

Spector, P. (1994): An Introduction to S and S-Plus,
Duxbury Press, Belmont, CA.
[On reserve]

Venables, W. N., Ripley, B. D. (1999): Modern Applied Statistics with S-Plus,
3rd Edition, Springer, New York.
[2nd Edition on reserve]

------------------------------------------------------------------------------

S/S-Plus on the WWW:

S-Plus @ MathSoft:
    http://www.splus.mathsoft.com/

StatLib - Software and extensions for the S (S-Plus) language:
    http://lib.stat.cmu.edu/S/
```

```
  in particular: S-scripts for producing figures in W.S.Cleveland's
  "Visualizing Data" book (Hobart Press, Summit, New Jersey, 1993):
    http://lib.stat.cmu.edu/S/visualizing.scripts

StatLib - S-news Archive:
    http://lib.stat.cmu.edu/s-news/

Dalhousie University, Halifax, Nova Scotia,
Department of Mathematics and Statistics, S-Plus Tutorial:
    http://www.mscs.dal.ca/Splus/contents.html

B. Sueselbeck, Universitaet Muenster, Germany, S-Plus Help System:
    http://www.uni-muenster.de/ZIV/Mitarbeiter/BennoSueselbeck/s-html/shelp.html

A search for "Splus" using "Excite" resulted in about 7680 hits on
9/13/00. In January 1997, the use of "Net Search" within Netscape
resulted in 1022 documents found. The number of documents on the Web
regarding S-Plus has been considerably increased over the last few
years. It is to be expected that even more documents on S-Plus
will be published on the Web in the near future.

-------------------------------------------------------------------------------

Setting up .Data Directories & Starting S-Plus:

% mkdir .Data
% Splus

# S-PLUS : Copyright (c) 1988, 1998 MathSoft, Inc.
# S : Copyright Lucent Technologies, Inc.
# Version 5.0 Release 3 for Sun SPARC, SunOS 5.5 : 1998
# Working data will be in .Data

!!! Warning: Different hardware platforms need different .Data directories !!!

If Splus finds a .Data directory, it will use this directory to store all
its data and user written functions. If NO .Data directory exists,
Splus will create a new data directory called Schapterxxxx, where xxxx
represents the process number of the Unix Splus process.
Note that every Schapterxxxx directory contains a .Data subdirectory.

% Splus

# Creating data directory for chapter
# S chapter initialized.
# S-PLUS : Copyright (c) 1988, 1998 MathSoft, Inc.
# S : Copyright Lucent Technologies, Inc.
# Version 5.0 Release 3 for Sun SPARC, SunOS 5.5 : 1998
# Working data will be in /home/symanzik/Schapter793


Exiting S-Plus:

> q()


If you want to use a previously created data directory Schapterxxxx
in another Splus session (that creates another data directory
Schapteryyyy), this can be done using the attach command:

> attach("Schapter793")

Note that new data and user written functions will be written to
the new Schapteryyyy/.Data directory. The old one will not be modified.
```

```
Alternatively, you can cd into the Schapterxxxx directory and
start Splus from there. Since in this case a local .Data directory
is available, Splus will use this directory. No new Schapteryyyy
directory will be created.

-------------------------------------------------------------------------------

Getting Help:

in S-Plus:

> help ("command")

e.g. help ("q")

-------------------------------------------------------------------------------

A first Example:

Starting a Graphics Window:

> motif()

The ":" operator generates a sequence of numbers:

> 0:10
 [1]  0  1  2  3  4  5  6  7  8  9 10
> 0:10 * .314
 [1] 0.000 0.314 0.628 0.942 1.256 1.570 1.884 2.198 2.512 2.826 3.140

Assignment: "<-" or "_"

> x0 <- 0:10
> x0
 [1]  0  1  2  3  4  5  6  7  8  9 10
> x1 <- 0:10 * .314
> x1
 [1] 0.000 0.314 0.628 0.942 1.256 1.570 1.884 2.198 2.512 2.826 3.140
> y _ sin (x1)
> y
 [1] 0.000000000 0.308865520 0.587527526 0.808736061 0.950859461 0.999999683
 [7] 0.951351376 0.809671788 0.588815562 0.310379910 0.001592653

Plotting:

> plot (x1, y)
> plot (x1, y, main = "Sine of x")
> plot (x1, y, main = "Sine of x", xlab = "X Coordinates", ylab = "Y Coordinates")
> plot (x1, y, main = "Sine of x", xlab = "X Coordinates", ylab = "Y Coordinates",
  type = "l")

-------------------------------------------------------------------------------

Question 1: Create a new plot of 101 points in the range [0, 2 pi] that
shows the cosine of these points. Points should be overlaid to the line
that connects the points. Hint: See help page for "par". Guess how to get pi.

-------------------------------------------------------------------------------

Data Structures/Accessing single components:

Variables:

> x1 _ 5
```

```
> x1
[1] 5
```

Vectors:

```
> x2 _ 1:10
> x2
 [1]  1  2  3  4  5  6  7  8  9 10
> x2[3]
[1] 3
```

Matrices:

```
> x3 _ matrix(1:10, 2, 5)
> x3
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    3    5    7    9
[2,]    2    4    6    8   10
> x3[1,]
[1] 1 3 5 7 9
> x3[,4]
[1] 7 8
> x3[2,3]
[1] 6

> x4 _ matrix(1:10, 2, 5, byrow = T)
> x4
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    2    3    4    5
[2,]    6    7    8    9   10
```

Lists:

```
> x5 _ list (x1, x4, x2)
> x5
[[1]]:
[1] 5

[[2]]:
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    2    3    4    5
[2,]    6    7    8    9   10

[[3]]:
 [1]  1  2  3  4  5  6  7  8  9 10
> x5[[3]]
 [1]  1  2  3  4  5  6  7  8  9 10
> x5[[2]][,4]
[1] 4 9
```

Characters:

```
> x6 _ c("alpha", "b", "3")
> x6
[1] "alpha" "b"     "3"
```

Logical values:

```
> x7 _ c(T, F, F)
> x7
[1] T F F

> x8 _ c(F, T, T)
> x8
[1] F T T
```

```
> x6[x7]
[1] "alpha"
> x6[x8]
[1] "b" "3"
```

Formulas and Basic Functions:

```
> sum (x2)
[1] 55
> mean (x2)
[1] 5.5
> var (x2)
[1] 9.166667

> x2 * x2
 [1]   1   4   9  16  25  36  49  64  81 100
> exp (x2)
 [1]     2.718282     7.389056    20.085537    54.598150   148.413159
 [6]   403.428793  1096.633158  2980.957987  8103.083928 22026.465795
> log (x2)
 [1] 0.0000000 0.6931472 1.0986123 1.3862944 1.6094379 1.7917595 1.9459101
 [8] 2.0794415 2.1972246 2.3025851
```

Some Important Functions:

c combines its arguments into a single vector:

```
> data.vector <- c(1:3,5:7,9,11)
> data.vector
[1]  1  2  3  5  6  7  9 11
```

rep replicates its first argument as many times as specified by its second argument:

```
> rep(1,length(data.vector))
[1] 1 1 1 1 1 1 1 1
> rep(data.vector,2)
[1]  1  2  3  5  6  7  9 11  1  2  3  5  6  7  9 11
```

seq generates a sequence of numbers. The ":" operator is a special case
of the seq function:

```
> seq(1,10)   # the same as 1:10
[1]  1  2  3  4  5  6  7  8  9 10
> seq(1,10,by=.5)
 [1]  1.0  1.5  2.0  2.5  3.0  3.5  4.0  4.5  5.0  5.5  6.0  6.5  7.0  7.5  8.0
[16]  8.5  9.0  9.5 10.0
> seq(1,10,length=50)
 [1]  1.000000  1.183673  1.367347  1.551020  1.734694  1.918367  2.102041
 [8]  2.285714  2.469388  2.653061  2.836735  3.020408  3.204082  3.387755
[15]  3.571429  3.755102  3.938776  4.122449  4.306122  4.489796  4.673469
[22]  4.857143  5.040816  5.224490  5.408163  5.591837  5.775510  5.959184
[29]  6.142857  6.326531  6.510204  6.693878  6.877551  7.061224  7.244898
[36]  7.428571  7.612245  7.795918  7.979592  8.163265  8.346939  8.530612
[43]  8.714286  8.897959  9.081633  9.265306  9.448980  9.632653  9.816327
[50] 10.000000
```

cut creates categories by cutting continuous data:

```
> cut (data.vector, 3)
[1] 1 1 1 2 2 2 3 3
attr(, "levels"):
[1] "0.9+ thru  4.3" "4.3+ thru  7.7" "7.7+ thru 11.1"
```

sort returns an vector which is a sorted version of the input:

```
> sort (rep(data.vector,2))
 [1]  1  1  2  2  3  3  5  5  6  6  7  7  9  9 11 11
```

Question 2: Calculate the temperature in degrees Celsius (C) for all
Fahrenheit (F) degrees in the range -40 to 100 in steps of 10.
The formula is: C = (F - 32) / 1.8

Logical Operations:

```
> x2
 [1]  1  2  3  4  5  6  7  8  9 10
> x9 _ 11:2
> x9
 [1] 11 10  9  8  7  6  5  4  3  2

> x2 >= x9
 [1] F F F F F T T T T T
> x2 < x9
 [1] T T T T T F F F F F

> x2[x2 < x9]
[1] 1 2 3 4 5
```

Dealing with NA's:

```
> x10 _ -5:5
> x10
 [1] -5 -4 -3 -2 -1  0  1  2  3  4  5

> x11 _ log (x10)
Warning messages:
  NAs generated in: log(x)
> x11
 [1]        NA        NA        NA        NA        NA      -Inf 0.0000000
 [8] 0.6931472 1.0986123 1.3862944 1.6094379

> median (x11)
[1] NA
> median (x11, na.rm = T)
[1] 0.8958797
> mean (x11[8:9])
[1] 0.8958797

> is.na(x11)
 [1] T T T T T F F F F F F
> x11 [ ! is.na(x11)]
[1]      -Inf 0.0000000 0.6931472 1.0986123 1.3862944 1.6094379

> median (x11 [ ! is.na(x11)])
[1] 0.8958797
```

```
-------------------------------------------------------------------------------

Matrix Algebra:

> x12 _ matrix(1:4, 2, 2)
> x12
     [,1] [,2]
[1,]    1    3
[2,]    2    4
> x12 * x12
     [,1] [,2]
[1,]    1    9
[2,]    4   16

Matrix Multiplication:

> x12 %*% x12
     [,1] [,2]
[1,]    7   15
[2,]   10   22

Transpose:

> t (x12)
     [,1] [,2]
[1,]    1    2
[2,]    3    4

Inverse:

> solve (x12)
     [,1] [,2]
[1,]   -2  1.5
[2,]    1 -0.5

> x12 %*% solve(x12)
             [,1]          [,2]
[1,] 1.000000e+00 -4.440892e-16
[2,] 8.881784e-16  1.000000e+00

Apply a Function to Sections of an Array
(MARGIN parameter: 1 indicates rows and 2 indicates columns):

> apply (x12, 1, exp)
          [,1]      [,2]
[1,]  2.718282  7.389056
[2,] 20.085537 54.598150
> apply (x12, 2, exp)
         [,1]     [,2]
[1,] 2.718282 20.08554
[2,] 7.389056 54.59815


Better example for apply:

> matrix (9:1, 3, 3)
     [,1] [,2] [,3]
[1,]    9    6    3
[2,]    8    5    2
[3,]    7    4    1
> apply (matrix (9:1, 3, 3), 1, sort)
     [,1] [,2] [,3]
[1,]    3    2    1
[2,]    6    5    4
```

```
[3,]    9    8    7
> apply (matrix (9:1, 3, 3), 2, sort)
     [,1] [,2] [,3]
[1,]    7    4    1
[2,]    8    5    2
[3,]    9    6    3
```

Reading in Data:

The contents of the file js.data is:

```
12 34 45 67
31 45 43 76
```

```
> jsread _ scan ("/home/symanzik/splus/js.data")
> jsread
[1] 12 34 45 67 31 45 43 76

> jsread2 _ matrix( scan ("/home/symanzik/splus/js.data"), 2, 4)
> jsread2
     [,1] [,2] [,3] [,4]
[1,]   12   45   31   43
[2,]   34   67   45   76

> jsread3 _ matrix( scan ("/home/symanzik/splus/js.data"), 2, 4, byrow = T)
> jsread3
     [,1] [,2] [,3] [,4]
[1,]   12   34   45   67
[2,]   31   45   43   76
```

Finally, the right format!

Reading in and Executing entire Files:

```
> source ("/home/symanzik/splus/sourcetest")
```

Reading Tables of Data - data frame:

```
> jstable _ read.table ("/home/symanzik/splus/js.data")
> jstable
  V1 V2 V3 V4
1 12 34 45 67
2 31 45 43 76
> dim(jstable)
[1] 2 4
> dimnames(jstable)
[[1]]:
[1] "1" "2"

[[2]]:
[1] "V1" "V2" "V3" "V4"
```

A standard data set:

```
 state _ data.frame (state.center, state.x77, row.names = state.abb)
> state [1:5, ]
           x       y Population Income Illiteracy Life.Exp Murder HS.Grad
AL  -86.7509 32.5901       3615   3624        2.1    69.05   15.1    41.3
AK -127.2500 49.2500        365   6315        1.5    69.31   11.3    66.7
AZ -111.6250 34.2192       2212   4530        1.8    70.55    7.8    58.1
```

```
AR  -92.2992 34.7336       2110   3378        1.9    70.66   10.1    39.9
CA -119.7730 36.5341      21198   5114        1.1    71.71   10.3    62.6
   Frost   Area
AL    20  50708
AK   152 566432
AZ    15 113417
AR    65  51945
CA    20 156361

> dimnames(state.x77)[[2]]
[1] "Population" "Income"     "Illiteracy" "Life Exp"   "Murder"
[6] "HS Grad"    "Frost"      "Area"

> state _ data.frame (state.center, state.x77,
  name = state.name, region = state.region, row.names = state.abb)
> state [1:5, ]
           x       y Population Income Illiteracy Life.Exp Murder HS.Grad
AL  -86.7509 32.5901       3615   3624        2.1    69.05   15.1    41.3
AK -127.2500 49.2500        365   6315        1.5    69.31   11.3    66.7
AZ -111.6250 34.2192       2212   4530        1.8    70.55    7.8    58.1
AR  -92.2992 34.7336       2110   3378        1.9    70.66   10.1    39.9
CA -119.7730 36.5341      21198   5114        1.1    71.71   10.3    62.6
   Frost   Area       name region
AL    20  50708    Alabama  South
AK   152 566432     Alaska   West
AZ    15 113417    Arizona   West
AR    65  51945   Arkansas  South
CA    20 156361 California   West

------------------------------------------------------------------------------

Attaching Other People's Directories:

Check availability of data at the UNIX level:

tolkien% pwd
/home/symanzik/splus
tolkien% ls -l
total 6
-rw-r--r--   1 symanzik staff         24 Sep 13 14:53 js.data
-rw-r--r--   1 symanzik staff         87 Sep 13 14:53 sourcetest
-rw-r--r--   1 symanzik staff         68 Sep 13 15:03 splusdemodata

In Splus:

> splusdemodata
Error: Object "splusdemodata" not found
> attach ("/home/symanzik/splus")   [needs full path]
> splusdemodata
 [1]  1  2  3  4  5  6  7  8  9 10


Create a startup function:

.First _ function()
{
 print ("Attaching JS directory")
 attach ("/home/symanzik/splus")
 print (splusdemodata)
}

quit Splus, restart, and see what happens


Removing this function again:
```

```
> rm (.First)
```

```
Similarly:

> xx _ 1
> xx
[1] 1
> rm (xx)
> xx
Error: Object "xx" not found
```

```
------------------------------------------------------------------------------

Random Number Generation:

> runif (100)
  [1] 0.960659164 0.937460009 0.044101933 0.764618514 0.705857692 0.503550521
  [7] 0.928648222 0.840273120 0.547101672 0.487805105 0.398984732 0.263519622
 [13] 0.925924629 0.428514566 0.960061379 0.298337657 0.577215644 0.488645311
 [19] 0.159736846 0.182527061 0.213183025 0.265985790 0.732723788 0.643868097
 [25] 0.897489898 0.499814727 0.576943601 0.905813668 0.014425047 0.746565899
 [31] 0.421574195 0.494985684 0.725713329 0.071925794 0.279651628 0.975036673
 [37] 0.509669911 0.772227993 0.980533611 0.714941630 0.471561028 0.119847733
 [43] 0.716741297 0.303822581 0.775616636 0.136684868 0.017757831 0.791862543
 [49] 0.723880269 0.880876297 0.412518194 0.222057927 0.805116300 0.549268416
 [55] 0.404414877 0.554270512 0.868483483 0.031665142 0.217252275 0.803386038
 [61] 0.226833535 0.510908458 0.375442212 0.494554662 0.645866242 0.783326366
 [67] 0.951738726 0.836408332 0.127163784 0.931654924 0.698016137 0.049239687
 [73] 0.912288268 0.926898122 0.116276071 0.348985139 0.501943978 0.858758209
 [79] 0.381377962 0.006793533 0.158040394 0.860326129 0.474985190 0.932033515
 [85] 0.795743512 0.256793795 0.637563583 0.655271475 0.706128572 0.373451429
 [91] 0.086245694 0.258623377 0.417373099 0.636045368 0.011345901 0.517812594
 [97] 0.315782589 0.701276027 0.249960655 0.514386513
> rnorm (100)
  [1] -0.917764850 -1.761528002  0.303701970 -0.524866891  1.467455342
  [6]  0.453631521  0.407779687  0.536222095  0.075956898  0.323955633
 [11] -1.353166483 -2.422615027  0.344129949  2.464569565  2.990625941
 [16] -1.556408910  1.271734502  1.544720216  0.803162292 -0.585806580
 [21]  0.887564070 -2.356727153  1.269861580 -1.108051749  0.562733355
 [26]  0.245423365  0.291905162  0.986758660  0.101259514  0.303514806
 [31]  0.044807527  1.435044924 -2.459295013  0.907460530  0.508864764
 [36] -1.118441465  0.513715984  0.582297380  2.286813997 -0.065091333
 [41] -0.898497927  0.687970758 -0.762046062 -1.021641427  1.349464485
 [46]  1.359369817  2.248522469 -1.270765249  0.198357403  2.484674224
 [51] -1.604709652  1.351729921 -0.582867798 -0.485981545  0.963504212
 [56] -0.563410076  1.323822092 -0.873647927 -1.700700570 -1.421790493
 [61]  0.796397824 -0.248718984 -0.827949229  0.749582735  1.097697150
 [66] -2.233537687  0.065925376 -0.315599656 -1.242237825 -0.967936934
 [71] -1.294310424 -0.511646201 -1.183767255 -0.443604339 -0.168885605
 [76]  0.417472634 -0.100217195  0.583684538 -0.282444864  0.189648235
 [81]  0.009124648 -1.957229831 -0.843219234 -1.142369140 -1.625863978
 [86] -0.593153260 -0.277026588 -1.036525482 -0.194730628 -0.952137384
 [91]  0.372314331 -0.345667301  0.066380799 -0.712903265 -1.673258866
 [96] -0.646916651 -0.446425318  0.489922573  0.449102068  0.008685083

------------------------------------------------------------------------------

More on Plotting:

> hist (runif(100))
> hist (runif(100))
> hist (runif(100))
```

```
> hist (rnorm(100))
> hist (rnorm(100))
> hist (rnorm(100))

> boxplot (rnorm(100))
> boxplot (rnorm(100))
> boxplot (rnorm(100))

> boxplot (runif(100), runif(100), rnorm(100), rnorm(100))

> plot (sort (rnorm(100)), qnorm( (1:100 - 0.5) / 100))
> plot (sort (rnorm(100)), qnorm( (1:100 - 0.5) / 100))
> plot (sort (rnorm(100)), qnorm( (1:100 - 0.5) / 100))

Creating multiple plots on 1 page:

> par(mfrow = c(2,3))
> plot (sort (rnorm(100)), qnorm( (1:100 - 0.5) / 100))
> plot (sort (rnorm(100)), qnorm( (1:100 - 0.5) / 100))
> plot (sort (rnorm(100)), qnorm( (1:100 - 0.5) / 100))
> plot (sort (rnorm(100)), qnorm( (1:100 - 0.5) / 100))
> plot (sort (rnorm(100)), qnorm( (1:100 - 0.5) / 100))
> plot (sort (rnorm(100)), qnorm( (1:100 - 0.5) / 100))

-----------------------------------------------------------------------------

Q3:

There exist several other distributions such as
beta, cauchy, chisq, exp, f, gamma, lnorm, logis, norm, stab, t, unif.

Select 4 of them, look up their parameters and see how a QQplot of
random numbers vs quantiles looks. Plot all these QQplots on
one single side. Give each plot an individual title.
Select a main title.

-----------------------------------------------------------------------------

Writing Functions:

myrexp _ function (mu, num = 1)
{
 cat ("This is my exp random function based on Inversion.\n", fill = T)
 cat ("Calculating random value(s) from exponential with mu = ")
 print (mu)
 cat ("", fill = T)
 cat ("That's all the text !", fill = T)

 -mu * log (runif(num))
}


> myrexp (1)
This is my exp random function based on Inversion.

Calculating random value(s) from exponential with mu = [1] 1

That's all the text !
[1] 2.141541


Also try:
myrexp (2, 100)

hist (myrexp (2, 100))
```

```
-----------------------------------------------------------------------------

Iterations:

For loop (known number of iterations):

> for (i in 1:5) print (i^2)
[1] 1
[1] 4
[1] 9
[1] 16
[1] 25

> for (i in c(3, 6, 10)) print (i^2)
[1] 9
[1] 36
[1] 100


While loop (check condition at the start):

Solution to the famous birthday problem: Assuming birthdays
are uniformly distributed over the year, how many people do
I have to select randomly to obtain a probability >= 0.5
that at least 2 people have the same birthday, i.e.,
P(at least 2 identical birthdays) = 1 - P(no identical birthday)

n _ 1
pno _ 1
while (pno > 0.5)
{
  pno _ pno * (365 - n) / 365
  n _ n + 1
}

> n
[1] 23
> pno
[1] 0.4927028


Repeat loop (check condition at the end):

myfactorial _ function (n)
{
  if (n == 0)
    return (1)
  fact _ 1
  num _ 1
  repeat
  {
    fact _ fact * num
    num _ num + 1
    if (num > n) break
  }
  fact
}

> myfactorial(0)
[1] 1
> myfactorial(1)
[1] 1
> myfactorial(2)
[1] 2
```

```
> myfactorial(3)
[1] 6
> myfactorial(4)
[1] 24

-------------------------------------------------------------------------------

What to avoid:

!!! Vectorize whenever possible !!!

veryslow _ function (num)
{
 print (unix("date"))

 x _ 1:num

 for (i in 1:num)
   x[i] _ x[i] * 2

 print (unix("date"))

 x
}

> test1 _ veryslow (10000)
[1] "Wed Sep 13 15:20:54 MDT 2000"
[1] "Wed Sep 13 15:21:03 MDT 2000"

> test1 _ veryslow (10000)
[1] "Wed Sep 13 15:21:20 MDT 2000"
[1] "Wed Sep 13 15:21:29 MDT 2000"


veryfast _ function (num)
{
 print (unix("date"))

 x _ 1:num * 2
 print (unix("date"))

 x
}

> test2 _ veryfast (10000)
[1] "Wed Sep 13 15:22:27 MDT 2000"
[1] "Wed Sep 13 15:22:27 MDT 2000"

> test2 _ veryfast (10000)
[1] "Wed Sep 13 15:22:31 MDT 2000"
[1] "Wed Sep 13 15:22:31 MDT 2000"

-------------------------------------------------------------------------------

Fitting Linear Models:

> attach (state)
> summary (state)
         x                 y           Population        Income
 Min.   :-127.20   Min.   :27.87   Min.   :  365   Min.   :3098
 1st Qu.:-104.20   1st Qu.:35.55   1st Qu.: 1079   1st Qu.:3993
 Median : -89.90   Median :39.62   Median : 2838   Median :4519
 Mean   : -92.46   Mean   :39.41   Mean   : 4246   Mean   :4436
 3rd Qu.: -78.98   3rd Qu.:43.14   3rd Qu.: 4968   3rd Qu.:4813
 Max.   : -68.98   Max.   :49.25   Max.   :21200   Max.   :6315
```

```
   Illiteracy       Life.Exp         Murder          HS.Grad
 Min.   :0.500   Min.   :67.96   Min.   : 1.400   Min.   :37.80
 1st Qu.:0.625   1st Qu.:70.12   1st Qu.: 4.350   1st Qu.:48.05
 Median :0.950   Median :70.67   Median : 6.850   Median :53.25
 Mean   :1.170   Mean   :70.88   Mean   : 7.378   Mean   :53.11
 3rd Qu.:1.575   3rd Qu.:71.89   3rd Qu.:10.670   3rd Qu.:59.15
 Max.   :2.800   Max.   :73.60   Max.   :15.100   Max.   :67.30

     Frost             Area                   name            region
 Min.   :  0.00   Min.   :  1049   Wyoming      : 1   Northeast    : 9
 1st Qu.: 66.25   1st Qu.: 36990   Wisconsin    : 1   South        :16
 Median :114.50   Median : 54280   West Virginia: 1   North Central:12
 Mean   :104.50   Mean   : 70740   Washington   : 1   West         :13
 3rd Qu.:139.70   3rd Qu.: 81160   Virginia     : 1
 Max.   :188.00   Max.   :566400   Vermont      : 1
                                   (Other)      :44


> state.fit _ lm (Income ~ Illiteracy + HS.Grad + Murder, state)


> state.fit
Call:
lm(formula = Income ~ Illiteracy + HS.Grad + Murder, data = state)

Coefficients:
 (Intercept) Illiteracy  HS.Grad   Murder
    2017.962  -176.8359 45.18885 30.47528

Degrees of freedom: 50 total; 46 residual
Residual standard error: 490.1023


> coef (state.fit)
 (Intercept) Illiteracy  HS.Grad   Murder
    2017.962  -176.8359 45.18885 30.47528


> summary (state.fit)

Call: lm(formula = Income ~ Illiteracy + HS.Grad + Murder, data = state)
Residuals:
   Min     1Q Median    3Q  Max
 -1068 -274.7 -40.47 182.7 1204

Coefficients:
                 Value Std. Error    t value   Pr(>|t|)
(Intercept)  2017.9618   750.1040     2.6902     0.0099
 Illiteracy  -176.8359   187.2546    -0.9444     0.3499
    HS.Grad    45.1889    11.5143     3.9246     0.0003
     Murder    30.4753    26.6988     1.1414     0.2596

Residual standard error: 490.1 on 46 degrees of freedom
Multiple R-Squared: 0.4028
F-statistic: 10.34 on 3 and 46 degrees of freedom, the p-value is 2.551e-05

Correlation of Coefficients:
           (Intercept) Illiteracy HS.Grad
Illiteracy -0.5520
   HS.Grad -0.9758      0.5061
    Murder -0.1324     -0.5811     0.0485


> par(mfrow = c(2,3))
```

```
> plot (state.fit)


> state.fit2_ update (state.fit, . ~ . - Murder - Illiteracy)


> summary (state.fit2)

Call: lm(formula = Income ~ HS.Grad, data = state)
Residuals:
   Min     1Q Median    3Q  Max
 -1083 -277.4 -34.15 241.5 1238

Coefficients:
                Value Std. Error   t value  Pr(>|t|)
(Intercept) 1931.1047  462.7391     4.1732    0.0001
    HS.Grad   47.1623    8.6161     5.4738    0.0000

Residual standard error: 487.1 on 48 degrees of freedom
Multiple R-Squared: 0.3843
F-statistic: 29.96 on 1 and 48 degrees of freedom, the p-value is 1.579e-06

Correlation of Coefficients:
        (Intercept)
HS.Grad -0.9889


> plot (state.fit2)


Other models:

  Analysis of Variance Model (aov)
  Generalized Linear Model (glm)
  Nonlinear Mixed Effects Model (nlme)
  Nonlinear Least Squares Regression (nls)

------------------------------------------------------------------------------

Creating ps output files for future use:

> dev.off ()                 [turn off previous graphics device]
 null device
           1
> postscript ("MYFILE")      [open ps graphics device]
> plot (state.fit2)          [plot us usual]
> dev.off ()                 [write to file - only this creates a correct file]
Generated postscript file "MYFILE".
 null device
           1

Outside S-Plus:

ghostview MYFILE              [to see the file on your screen]

lpr MYFILE                    [to print the file]

------------------------------------------------------------------------------

Additional Capabilities:


Calling Fortran and C

Having C call S-Plus
```

```
Object-oriented stuff

Advanced modeling

Advanced graphics

Additional packages based on S-Plus, e.g., spatial

-----------------------------------------------------------------------------
```

B.2 Introduction to S and S–Plus — Answers

http://www.math.usu.edu/~symanzik/teaching/2000_spatial/splus/answers

```
Stat 5810 Applied Spatial Statistics                              Fall 2000


                Introduction to S and S-Plus - Answers
                ====================================

          Wed 9/13/00, Mon 9/18/00, Wed 9/20/00 & Mon 9/25/00


Dr. Juergen Symanzik

-----------------------------------------------------------------------------

Q1:

> quest1 _ 0:100 * 2 * pi / 100
> quest1
  [1] 0.00000000 0.06283185 0.12566371 0.18849556 0.25132741 0.31415927
  [7] 0.37699112 0.43982297 0.50265482 0.56548668 0.62831853 0.69115038
 [13] 0.75398224 0.81681409 0.87964594 0.94247780 1.00530965 1.06814150
 [19] 1.13097336 1.19380521 1.25663706 1.31946891 1.38230077 1.44513262
 [25] 1.50796447 1.57079633 1.63362818 1.69646003 1.75929189 1.82212374
 [31] 1.88495559 1.94778745 2.01061930 2.07345115 2.13628300 2.19911486
 [37] 2.26194671 2.32477856 2.38761042 2.45044227 2.51327412 2.57610598
 [43] 2.63893783 2.70176968 2.76460154 2.82743339 2.89026524 2.95309709
 [49] 3.01592895 3.07876080 3.14159265 3.20442451 3.26725636 3.33008821
 [55] 3.39292007 3.45575192 3.51858377 3.58141563 3.64424748 3.70707933
 [61] 3.76991118 3.83274304 3.89557489 3.95840674 4.02123860 4.08407045
 [67] 4.14690230 4.20973416 4.27256601 4.33539786 4.39822972 4.46106157
 [73] 4.52389342 4.58672527 4.64955713 4.71238898 4.77522083 4.83805269
 [79] 4.90088454 4.96371639 5.02654825 5.08938010 5.15221195 5.21504380
 [85] 5.27787566 5.34070751 5.40353936 5.46637122 5.52920307 5.59203492
 [91] 5.65486678 5.71769863 5.78053048 5.84336234 5.90619419 5.96902604
 [97] 6.03185789 6.09468975 6.15752160 6.22035345 6.28318531
> plot (quest1, cos (quest1), type = "o")


Q2:

> seq (-40, 100, by = 10)
 [1]  -40  -30  -20  -10    0   10   20   30   40   50   60   70   80   90  100
> (seq (-40, 100, by = 10) - 32) / 1.8
 [1] -40.000000 -34.444444 -28.888889 -23.333333 -17.777778 -12.222222
 [7]  -6.666667  -1.111111   4.444444  10.000000  15.555556  21.111111
[13]  26.666667  32.222222  37.777778


Q3:

par (mfrow = c(2,2), oma = c (0, 0, 4, 0))
plot (sort (rbeta(100, 1, 1)), qbeta( (1:100 - 0.5) / 100, 1, 1), main = "Beta")
plot (sort (rf(100, 5, 8)), qf( (1:100 - 0.5) / 100, 5, 8), main = "F")
plot (sort (rt(100, 5)), qt( (1:100 - 0.5) / 100, 5), main = "t")
plot (sort (rgamma(100, 5)), qgamma( (1:100 - 0.5) / 100, 5), main = "Gamma")
mtext (side = 3, line = 0, cex = 2, outer = TRUE, "Several Distributions")
```

B.3 Introduction to S+SpatialStats

http://www.math.usu.edu/~symanzik/teaching/2000_spatial/splus/sspatialintro

```
Stat 5810 Applied Spatial Statistics                                Fall 2000


                      Introduction to S+SpatialStats
                      ==============================

                        Wed 9/27/00 & Wed 10/18/00


Dr. Juergen Symanzik

-------------------------------------------------------------------------------

References:

Kaluzny, S. P., Vega, S. C., Cardoso, T. P., Shelly, A. A. (1998):
S+SpatialStats - User's Manual for Windows and UNIX, Springer, New York.

-------------------------------------------------------------------------------

Loading the Spatial Statistics module:

> module (spatial)

> .First _ function() {module (spatial) }

-------------------------------------------------------------------------------

Analyzing Spatial Point Patterns:
=================================

SPP data sets included with S+SpatialStats:

bramble
lansing
quakes.bay
quakes.wash

-------------------------------------------------------------------------------

Creating a Spatial Point Pattern:

> summary (quakes.bay)
    longitude         latitude       magnitude          date
    Min.:-123.3      Min.:36.01      Min.:2.500      Min.: 734
 1st Qu.:-121.7   1st Qu.:36.61   1st Qu.:2.600   1st Qu.:2771
  Median:-121.5    Median:36.84    Median:2.800    Median:4588
    Mean:-121.5      Mean:37.00      Mean:2.947      Mean:4549
 3rd Qu.:-121.2   3rd Qu.:37.36   3rd Qu.:3.100   3rd Qu.:6437
    Max.:-119.8      Max.:38.50      Max.:5.800      Max.:8024

       hour           minute          second
    Min.: 0.00      Min.: 0.00      Min.: 0.00
 1st Qu.: 5.00   1st Qu.:15.00   1st Qu.:14.60
  Median:11.00    Median:31.00    Median:30.20
    Mean:11.47      Mean:30.09      Mean:30.07
 3rd Qu.:18.00   3rd Qu.:45.00   3rd Qu.:45.20
    Max.:23.00      Max.:59.00      Max.:59.90
```

```
> quakes.spp _ spp (quakes.bay)
> summary (quakes.spp)
Total number of points: 2049
Coordinate extents :
 longitude :  -123.33 , -119.78
 latitude :  36.01 , 38.50
Number of boundary vertices : 4
Other covariates :
    magnitude          date           hour           minute
    Min.:2.500      Min.: 734      Min.: 0.00      Min.: 0.00
 1st Qu.:2.600   1st Qu.:2771   1st Qu.: 5.00   1st Qu.:15.00
  Median:2.800    Median:4588    Median:11.00    Median:31.00
    Mean:2.947      Mean:4549      Mean:11.47      Mean:30.09
 3rd Qu.:3.100   3rd Qu.:6437   3rd Qu.:18.00   3rd Qu.:45.00
    Max.:5.800      Max.:8024      Max.:23.00      Max.:59.00

      second
    Min.: 0.00
 1st Qu.:14.60
  Median:30.20
    Mean:30.07
 3rd Qu.:45.20
    Max.:59.90


> par (pty = "s")     # square plotting region
> plot (quakes.spp, boundary = T)
> bbox(quakes.spp)
$x:
[1] -123.33 -123.33 -119.78 -119.78

$y:
[1] 38.50 36.01 36.01 38.50


-------------------------------------------------------------------------------

Creating a subset:

> quakes.spp$magnitude[1:100]
  [1] 3.0 2.8 2.6 2.5 3.7 2.5 2.6 3.9 2.7 2.5 3.2 2.5 2.7 3.4 3.0 3.1 3.1 2.9
 [19] 2.6 3.0 3.1 4.1 3.1 3.4 3.0 3.7 3.5 3.0 3.5 2.7 2.6 3.0 4.7 2.7 2.8 2.7
 [37] 2.7 3.1 3.8 2.8 2.8 2.6 2.7 2.9 2.7 2.6 3.1 2.7 2.5 2.6 3.2 2.7 2.6 3.1
 [55] 2.6 3.3 2.8 3.0 3.0 2.6 2.5 3.6 2.9 3.2 3.7 3.4 2.5 3.0 2.6 2.7 2.5 2.6
 [73] 3.6 2.5 3.6 2.7 2.7 2.8 2.7 2.9 2.9 3.7 2.5 2.7 2.5 2.8 2.7 3.3 2.7 3.0
 [91] 3.4 3.7 2.5 2.8 2.6 3.6 2.6 3.2 3.1 3.3

> attach(quakes.spp)
> magnitude[1:100]
 1   2   3   4   5   6   7   8   9  10  11  12  13  14 15  16  17  18  19 20
 3 2.8 2.6 2.5 3.7 2.5 2.6 3.9 2.7 2.5 3.2 2.5 2.7 3.4  3 3.1 3.1 2.9 2.6  3

  21  22  23  24 25  26  27 28  29  30  31 32  33  34  35  36  37  38  39  40
 3.1 4.1 3.1 3.4  3 3.7 3.5  3 3.5 2.7 2.6  3 4.7 2.7 2.8 2.7 2.7 3.1 3.8 2.8

  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57 58 59  60
 2.8 2.6 2.7 2.9 2.7 2.6 3.1 2.7 2.5 2.6 3.2 2.7 2.6 3.1 2.6 3.3 2.8  3  3 2.6

  61  62  63  64  65  66  67 68  69  70  71  72  73  74  75  76  77  78  79
 2.5 3.6 2.9 3.2 3.7 3.4 2.5  3 2.6 2.7 2.5 2.6 3.6 2.5 3.6 2.7 2.7 2.8 2.7

  80  81  82  83  84  85  86  87  88  89 90  91  92  93  94  95  96  97  98
 2.9 2.9 3.7 2.5 2.7 2.5 2.8 2.7 3.3 2.7  3 3.4 3.7 2.5 2.8 2.6 3.6 2.6 3.2

  99 100
```

```
 3.1 3.3

> quakes40.spp _ spp (quakes.bay[magnitude >= 4.0,], boundary = bbox(quakes.spp))
> summary (quakes40.spp)
Total number of points: 94
Coordinate extents :
 longitude :  -122.73 , -120.16
 latitude :  36.11 , 38.49
Number of boundary vertices : 4
Other covariates :
    magnitude          date           hour           minute
    Min.:4.000      Min.: 806      Min.: 0.00      Min.: 0.00
 1st Qu.:4.100   1st Qu.:3568   1st Qu.: 6.00   1st Qu.:11.25
  Median:4.400    Median:5094    Median:11.00    Median:30.50
    Mean:4.469      Mean:4982      Mean:11.73      Mean:29.16
 3rd Qu.:4.700   3rd Qu.:7157   3rd Qu.:18.00   3rd Qu.:43.75
    Max.:5.800      Max.:7941      Max.:23.00      Max.:59.00

      second
    Min.: 0.00
 1st Qu.:13.03
  Median:26.95
    Mean:28.23
 3rd Qu.:45.10
    Max.:59.70


par (mfrow = c(1, 2))
par (pty = "s")
plot (quakes.spp, boundary = T, main = "All Earthquakes")
points (quakes40.spp, col = 2, pch = "x")
plot (quakes40.spp, boundary = T, main = "Earthquakes with Magnitude >= 4.0")

-----------------------------------------------------------------------------

Different plots of intensities:

par (mfrow = c(3, 2))
par (pty = "s")
quakes.bin _ intensity (quakes.spp, method = "binning", nx = 30, ny = 30, span = 0.5)
plot (quakes.spp)
contour (quakes.bin)
image (quakes.bin)
persp (quakes.bin)
quakes.kernel _ intensity (quakes.spp, method = "kernel", bw = 0.5)
contour (quakes.kernel)
image (quakes.kernel)

-----------------------------------------------------------------------------

Nearest neighbor methods:

par (mfrow = c(2, 2))
quakes.ghat _ Ghat (quakes.spp)
title (main = "All Earthquakes")
quakes40.ghat _ Ghat (quakes40.spp)
title (main = "Earthquakes >= 4.0")

plot (c(0, 0.5), c(0, 1.0), type = "n", xlab = "dist", ylab = "Ghat")
points (quakes.ghat)
plot (c(0, 0.5), c(0, 1.0), type = "n", xlab = "dist", ylab = "Ghat")
points (quakes40.ghat)

-----------------------------------------------------------------------------
```

```
K & L function:

par (mfrow = c(2, 2))
quakes.khat _ Khat (quakes.spp)
title (main = "All Earthquakes")
quakes40.khat _ Khat (quakes40.spp)
title (main = "Earthquakes >= 4.0")

quakes.lhat _ Lhat (quakes.spp)
abline(0, 1)
quakes40.lhat _ Lhat (quakes40.spp)
abline(0, 1)

----------------------------------------------------------------------------

Creating a CSR process:

par (mfrow = c(2, 2))
csr _ make.pattern (n = 2049)
plot (csr)
csr.bin _ intensity (csr, method = "binning", nx = 30, ny = 30, span = 0.1)
persp (csr.bin)
Khat (csr)
Lhat (csr)
abline(0, 1)

----------------------------------------------------------------------------
----------------------------------------------------------------------------

Creating CSR differently:

par (mfrow = c(1, 2))
csr _ make.pattern (n = 2049)
plot (csr, main = "CSR with fixed n")
summary(csr)
#Total number of points: 2049
#Coordinate extents :
# x :  0.0005649887 , 0.9985041954
# y :  8.638017e-06 , 9.985471e-01
#Number of boundary vertices : 4

csr2 _ make.pattern (process = "poisson", lambda = 2049)
plot (csr2, main = "CSR defined via lambda")
summary(csr2)
#Total number of points: 2073
#Coordinate extents :
# x :  0.0004788809 , 0.9997592210
# y :  0.0001384751 , 0.9999346910
#Number of boundary vertices : 4

----------------------------------------------------------------------------

Creating point patterns other than CSR:

par (mfrow = c(3, 3))

ssi0.02 _ make.pattern (100, process = "SSI", radius = 0.02)
plot (ssi0.02, main = "SSI with radius 0.02")

ssi0.05 _ make.pattern (100, process = "SSI", radius = 0.05)
plot (ssi0.05, main = "SSI with radius 0.05")

ssi0.08 _ make.pattern (100, process = "SSI", radius = 0.08)
plot (ssi0.08, main = "SSI with radius 0.08")
```

```
strauss0.0 _ make.pattern (100, process = "Strauss", radius = 0.08, cpar = 0.0)
plot (strauss0.0, main = "Strauss process with radius 0.08 and cpar = 0.0")

strauss0.5 _ make.pattern (100, process = "Strauss", radius = 0.08, cpar = 0.5)
plot (strauss0.5, main = "Strauss process with radius 0.08 and cpar = 0.5")

strauss1.0 _ make.pattern (100, process = "Strauss", radius = 0.08, cpar = 1.0)
plot (strauss1.0, main = "Strauss process with radius 0.08 and cpar = 1.0")


cluster0.02 _ make.pattern (100, process = "cluster", radius = 0.02, cpar = 10)
plot (cluster0.02, main = "Cluster process with radius 0.02")

cluster0.05 _ make.pattern (100, process = "cluster", radius = 0.05, cpar = 10)
plot (cluster0.05, main = "Cluster process with radius 0.05")

cluster0.08 _ make.pattern (100, process = "cluster", radius = 0.08, cpar = 10)
plot (cluster0.08, main = "Cluster process with radius 0.08")

-----------------------------------------------------------------------------

Simulation envelopes:

quakes.spp _ spp (quakes.bay)

par (mfrow = c(1, 1))
quakes.khat _ Khat (quakes.spp, plot = F)
plot (quakes.khat$values[quakes.khat$values[,1] < 0.5, ])
title ("Khat and Simulation Envelopes")

quakes.kenv _ Kenv (quakes.spp, nsims = 4, add = F)
quakes.index _ quakes.kenv$dist < 0.5
lines (quakes.kenv$dist[quakes.index], quakes.kenv$lower[quakes.index])
lines (quakes.kenv$dist[quakes.index], quakes.kenv$upper[quakes.index])

(quakes.kenv$upper[quakes.index] - quakes.kenv$lower[quakes.index])[1:10]

# [1] 0.0001416664 0.0003322114 0.0007732138 0.0008911356 0.0008334583
# [6] 0.0011996496 0.0011787788 0.0012590687 0.0018577206 0.0014500613

-----------------------------------------------------------------------------

Dealing with nearest neighbor distances:

xmin _ min (bbox(quakes.spp)$x)
xmax _ max (bbox(quakes.spp)$x)
ymin _ min (bbox(quakes.spp)$y)
ymax _ max (bbox(quakes.spp)$y)

quakes.ext _ matrix (c(xmin, xmax, quakes.spp$longitude,
   ymin, ymax, quakes.spp$latitude), 2051, 2)

quakes.neighbors _ find.neighbor (quakes.ext, k = 2)
quakes.neighbors[1:10,]

#   index1 index2  distances
# 1      1    760 1.14109596
# 2      1      1 0.00000000
# 3      2   1561 1.61969133
# 4      2      2 0.00000000
# 5      3    815 0.10000000
# 6      3      3 0.00000000
# 7      4    292 0.03162278
# 8      4      4 0.00000000
```

```
# 9      5   1803 0.03162278
#10      5      5 0.00000000

nrow(quakes.neighbors)
# [1] 4102

plot (quakes.spp)
lines (c(xmin, quakes.spp$longitude[quakes.neighbors[1,2] - 2]),
       c(ymin, quakes.spp$latitude[quakes.neighbors[1,2] - 2]) )
lines (c(xmax, quakes.spp$longitude[quakes.neighbors[3,2] - 2]),
       c(ymax, quakes.spp$latitude[quakes.neighbors[3,2] - 2]) )

quakes.diff _ quakes.neighbors[quakes.neighbors[,3] > 0, ]
nrow(quakes.diff)
# [1] 1067

quakes.diff2 _ quakes.neighbors[quakes.neighbors[,1] != quakes.neighbors[,2], ]
nrow(quakes.diff2)
# [1] 2375

--------------------------------------------------------------------------------
```

B.4 How to Solve a Problem in S–Plus

http://www.math.usu.edu/~symanzik/teaching/2000_spatial/splusfunctionwriting.full

```
Stat 5810 Applied Spatial Statistics                             Fall 2000

                      How to Solve a Problem in S-Plus
                      ================================

                         Mon 10/9/00 & Wed 10/11/00


Dr. Juergen Symanzik

------------------------------------------------------------------------------

Write S-Plus functions that implement the quadrat plot
from Section 2.3.1. One function might be used to do the
binning, i.e., to count how many points fall in each quadrat.
Another function might be used to create a 3D plot.
Instead of drawing a full 3D histogram, you can
simply work with 'persp'. The user should call only the
main function with 3 input parameters, where xbins and
ybins are optional:

  spp: a nx2 matrix of xy-coordinates representing a spatial point pattern
  xbins: number of bins >= 1 in x direction, default = 10
  ybins: number of bins >= 1 in y direction, default = 10

Test your functions with the quakes.bay data set. Also create
some data from a bivariate Normal distribution and use this
to check your result.

------------------------------------------------------------------------------

How to proceed:

1) For the binning function:

a) test your input parameters
b) set up a matrix that contains the counts
c) loop through your data and count the number of events
   in each bin - this can be done as follows:

   let
      xmin = smallest observed x-value
      xmax = largest observed x-value (assume xmin < xmax)
      deltax = xmax - xmin
      xbins = number of available bins (e.g., 5)

   assign x to the following bin:
      trunc ( (x - xmin) / deltax * xbins + 1)   if x < xmax
      xbins   if x = xmax

   do the same for the y-direction
d) test your function !!!


2) For the plotting function:

a) call persp
b) test your function !!!
```

3) Using the quakes.bay data set and data from a Normal distribution:

a) extract longitude and latitude as x & y coordinates
b) create the spp matrix
c) call the function
d) test your functions !!

----

1) Coding of the binning function:

```
binning _ function (spp, xbins = 10, ybins = 10)
{
  if (xbins < 1 || ybins < 1)
  {
    cat ("xbin and ybin must be >= 1.", fill = T)
    return (matrix(-1, 1, 1))
  }

  bincount _ matrix(0, xbins, ybins)

  x _ spp[,1]
  y _ spp[,2]

  xmin _ min (x)
  xmax _ max (x)
  ymin _ min (y)
  ymax _ max (y)

  deltax _ xmax - xmin
  deltay _ ymax - ymin

  for (i in 1:nrow(spp))
  {
    xx _ ifelse (x[i] == xmax, xbins, trunc ( (x[i] - xmin) / deltax * xbins + 1))
    yy _ ifelse (y[i] == ymax, ybins, trunc ( (y[i] - ymin) / deltay * ybins + 1))
    bincount[xx, yy] _ bincount[xx, yy] + 1
  }

  cat (sum(bincount))
  cat (" events have been binned.", fill = T)

  return (bincount)
}
```

2) Coding of the drawing function:

```
drawquadrats _ function (spp, xbins = 10, ybins = 10)
{
  binmat _ binning (spp, xbins, ybins)
  if (binmat[1,1] == -1)
    cat ("No plot possible.", fill = T)
  else
    persp (binmat)
  "OK"
}
```

3) Using the quakes.bay data set:

```
x _ quakes.bay$longitude
y _ quakes.bay$latitude
spp _ matrix (c(x, y), length(x), 2)
```

```
drawquadrats (spp)
drawquadrats (spp, 10, 15)

par (mfrow = c(1, 2))
plot (quakes.spp)
drawquadrats (spp)
```

4) Testing:

```
x _ quakes.bay$longitude
y _ quakes.bay$latitude

x _ x[1:100]
y _ y[1:100]

xmin _ min (x)
xmax _ max (x)
ymin _ min (y)
ymax _ max (y)

deltax _ xmax - xmin
deltay _ ymax - ymin

xbins _ 5
ybins _ 5

ifelse (x == xmax, xbins, trunc ( (x - xmin) / deltax * xbins + 1))

spp _ matrix (c(x, y), 100, 2)
binning (spp)
binning (spp, 0, 1)
binning (spp, 1, 1)
binning (spp, 2, 2)
binning (spp, 10, 15)

drawquadrats (spp)
drawquadrats (spp, 0, 1)
drawquadrats (spp, 1, 1)
drawquadrats (spp, 2, 2)
drawquadrats (spp, 10, 15)


# test a bivariate normal with correlation 0
xn _ rnorm (1000)
yn _ rnorm (1000)
nn _ matrix (c(xn, yn), 1000, 2)
par (mfrow = c(1, 2))
plot (nn)
drawquadrats (nn)

nn2 _ nn[(nn[,1] < 0) & (nn[,2] < 0),]
par (mfrow = c(1, 2))
plot (nn2)
drawquadrats (nn2)
```

B.5 Analysis of the Volcanic Craters Data by Bailey/Gatrell

http://www.math.usu.edu/~symanzik/teaching/2000_spatial/volcanic_craters.fullversion

```
Stat 5810 Applied Spatial Statistics                              Fall 2000

         Analysis of the Volcanic Craters Data by Bailey/Gatrell
         =======================================================

                        Wed 10/11/00 & Mon 10/16/00


Dr. Juergen Symanzik

-------------------------------------------------------------------------------

1) Load the volcanic craters data from the Web, i.e., from
  http://math.usu.edu/~symanzik/teaching/2000_spatial/infomap_data/Volcano.txt
Store it locally in a file.
Open the file in a text editor.
Remove the headings and the "'" symbols.
Then store the data set locally.


2) Reading in the Data and Creating a SPP:

vol _ matrix( scan ("/home/symanzik/splus/Volcano.dat"), 120, 4, byrow = T)
vol.coord _ matrix( c(vol[,2], vol[,3]), 120, 2)
vol.spp _ spp (vol.coord)
summary (vol.spp)


3) Simple Plots: Data and Kernel Estimates

par (mfrow = c(2,2))
par (pty = "s")
plot (vol.spp, main = "Volcanic Craters")
   # should be similar to Figure 3.1 in Bailey/Gatrell

vol.kernel100 _ intensity (vol.spp, method = "kernel", bw = 100)
image (vol.kernel100)

vol.kernel220 _ intensity (vol.spp, method = "kernel", bw = 220)
image (vol.kernel220)

vol.kernel500 _ intensity (vol.spp, method = "kernel", bw = 500)
image (vol.kernel500)
   # should be similar to Figure 3.5 in Bailey/Gatrell


4) Nearest Neighbor Plots, Khat & Lhat function:

par (mfrow = c(2,2))
vol.ghat _ Ghat (vol.spp)
   # should be similar to Figure 3.6 in Bailey/Gatrell
vol.khat _ Khat (vol.spp)
vol.lhat _ Lhat (vol.spp)
abline(0, 1)
   # positive peaks between 100 and 1500 - probably clustering!


5) Khat and Simulation Envelopes:

par (mfrow = c(1,2))
```

```
vol.khat _ Khat (vol.spp, plot = F)
plot (vol.khat$values[vol.khat$values[,1] < 500, ])
title ("Khat & Simulation Envelopes, dist < 500")

vol.kenv _ Kenv (vol.spp, nsims = 39, add = F)
vol.index _ vol.kenv$dist < 500
lines (vol.kenv$dist[vol.index], vol.kenv$lower[vol.index])
lines (vol.kenv$dist[vol.index], vol.kenv$upper[vol.index])
   # if there was any doubt - certainly clustering at these distances!


plot (vol.khat$values[vol.khat$values[,1] < 100, ])
title ("Khat & Simulation Envelopes, dist < 100")

vol.index _ vol.kenv$dist < 100
lines (vol.kenv$dist[vol.index], vol.kenv$lower[vol.index])
lines (vol.kenv$dist[vol.index], vol.kenv$upper[vol.index])
   # regularity not very likely at small distances!
```

B.6 Analysis of the Pines Data by Ripley & Venables/Ripley

http://www.math.usu.edu/~symanzik/teaching/2000_spatial/pines.final

```
Stat 5810 Applied Spatial Statistics                           Fall 2000

         Analysis of the Pines Data by Ripley & Venables/Ripley
         ======================================================

                              Mon 10/23/00


Dr. Juergen Symanzik

-------------------------------------------------------------------------------


1) Reading in the Data and Creating a SPP:

pines _ matrix( scan ("/home/symanzik/splus/pines.dat"), 71, 2, byrow = T)
pines.spp _ spp (pines / 10, boundary = bbox(x = c(0, 10), y = c(0, 10)))
summary (pines.spp)


2) ghat.env function from Kaluzny et al...

ghat.env _ function (n, s, dist,
  boundary = bbox(x = c(0,1), y = c(0,1)))
{
  hold _ matrix(0, s, length(dist))
  for (i in 1:s)
  {
    hold[i, ] _ Ghat(make.pattern(n, boundary = boundary),
      dist.ghat = dist, plot = F)[,2]
  }

  mn _ apply(hold, 2, mean)
  Up _ apply(hold, 2, max)
  Down _ apply(hold, 2, min)
  return(data.frame(mn, Up, Down))
}


3) Simple Plot: Figure 8.7 (a) in Ripley

par (mfrow = c(2,4))
par (pty = "s")
plot (pines.spp, main = "Pine Trees")


4) Lhat and envelopes: Figure 8.7 (c) in Ripley

pines.lhat _ Lhat (pines.spp)
title ("Lhat")
abline(0, 1)

pines.env _ Lenv (pines.spp, nsims = 50, process = "binomial",
  boundary = bbox(x = c(0, 10), y = c(0, 10)))

# concentrate on interval from 0.0 to 2.0

pines.hatdist _ pines.lhat$values[,1] < 2.0
plot (pines.lhat$values[pines.hatdist,])
title ("Lhat in [0.0, 2.0]")
```

```
lines (pines.env$dist[pines.hatdist],
  pines.env$upper[pines.hatdist])
lines (pines.env$dist[pines.hatdist],
  pines.env$lower[pines.hatdist])
abline(0, 1)
```

5) Nearest Neighbor Plot Ghat and envelopes: should be Figure 8.7 (d) in Ripley

```
pines.ghat _ Ghat (pines.spp, plot = F)

pines.sdist = seq(0.0, 1.6, by = 0.05)

pines.genv _ ghat.env (n = 71, s = 100, pines.sdist,
  boundary = bbox(x = c(0, 10), y = c(0, 10)))

plot (c(0, 1.6), c(0, 1.0), xlab = "distance", ylab = "Ghat", type = "n")
title ("Ghat in [0.0, 1.6]")
points (pines.ghat)
lines (pines.sdist, pines.genv$Up, lty = 2)
lines (pines.sdist, pines.genv$Down, lty = 2)
```

6) Strauss process - Simple Plot

```
pines.strauss _ make.pattern (71, process = "Strauss",
  radius = 0.7, cpar = 0.2, boundary = bbox(x = c(0, 10), y = c(0, 10)))

plot(pines.strauss)
```

7) Strauss process - Lhat and envelopes: Figure 8.7 (e) in Ripley

```
pines.strlhat _ Lhat (pines.strauss)
pines.strkenv _ Lenv (pines.strauss, nsims = 50, process = "Strauss",
  radius = 0.7, cpar = 0.2, boundary = bbox(x = c(0, 10), y = c(0, 10)))
title ("Lhat for Strauss")

# concentrate on interval from 0.0 to 2.0

pines.strhatdist _ pines.strlhat$values[,1] < 2.0
plot (pines.strlhat$values[pines.strhatdist,])
title ("Lhat for Strauss in [0.0, 2.0]")
lines (pines.strkenv$dist[pines.strhatdist],
  pines.strkenv$upper[pines.strhatdist])
lines (pines.strkenv$dist[pines.strhatdist],
  pines.strkenv$lower[pines.strhatdist])
```

B.7 Introduction to XGobi and ArcView/XGobi/XploRe

http://www.math.usu.edu/~symanzik/teaching/2000_spatial/xgobiinfo

```
Stat 5810 Applied Spatial Statistics                                Fall 2000

              Introduction to XGobi and ArcView/XGobi/XploRe
              ==============================================

            Wed 10/25/00, Mon 10/30/00, Mon 11/6/00, Wed 11/8/00


Dr. Juergen Symanzik

-----------------------------------------------------------------------------


1) First time initialization:

Add the following lines to your .cshrc-private file:

############################################################

# ArcView/XGobi/XploRe stuff

setenv AXX_BASE /home/symanzik/axx
setenv AV2XGOBI_HOME $AXX_BASE/av_dir
setenv XGOBIDIR $AXX_BASE/xgobi
setenv USEREXT $AXX_BASE/av_dir
alias xgobi $XGOBIDIR/bin/xgobi

setenv XPL4HOME $AXX_BASE/xplore
setenv XPL4PROG $XPL4HOME
setenv XPL4LIB $XPL4HOME/lib
setenv XPL4DATA $XPL4HOME/data

setenv HOST tolkien

############################################################

!!! Note - put the name of the machine you are using most frequently
in the last setenv HOST command; if you work on another machine, reissue
the command "setenv HOST currentmachinename", i.e., use the name
of the machine you are currently working at

These commands will be executed each time you log on again. To activate
them for the current session, type:
   source .cshrc-private

Try
   xgobi $XGOBIDIR/data_xgobi/places

XGobi should pop up with the "places" data set

-----------------------------------------------------------------------------

2) XGobi Background:

XGobi is a dynamical statistical graphics program, developed
by Debby Swayne, Di Cook, and Andreas Buja.

It supports one-dimensional plots, scatterplots, scatterplot matrizes,
parallel coordinate plots, the grand tour, linked brushing, and much more.
It also provides exploratory features that are useful for spatial data sets.
```

```
+ The XGobi home page is
   http://www.research.att.com/areas/stat/xgobi/

Many papers on the use of XGobi are available through this page.


+ A book on XGobi is currently in the making. Draft chapters are
available at
   http://www.public.iastate.edu/~dicook/xgobi-book/xgobi.html

Most useful for our class are the Introduction and the chapter on Space.


+ Additional publications how to use XGobi and ArcView/XGobi/XploRe
are available through my "Publications & Videos" Web page.

------------------------------------------------------------------------------

3) XGobi Basics:

+ Click right mouse button: On-line help menu

+ Inside Variable Circle:
  Click left mouse button: Select variable on horizontal axis

  Click middle mouse button: Select variable on vertical axis

+ On Menu Item:
  Click and hold left mouse button

+ In Brushing Mode:
  Left mouse button moves brush

  Middle mouse button resizes brush

  Moving pointer to another location and clicking left mouse button
  causes brush to jump to this new location

+ Scatterplot Matrix:
  All variables are used that are currently active in the Grand Tour View

+ Persistent Brushing (in contrast to Transient Brushing) permanently
  keeps the brushing information

------------------------------------------------------------------------------

4) Basic XGobi and ArcView/XGobi/XploRe Tutorial:

  http://www.math.usu.edu/~symanzik/teaching/2000_spatial/workshop_epa.pdf

Note that this has been created on a different machine.

If you want to use XGobi with any of the data sets listed in this tutorial,
you have to use
   xgobi $XGOBIDIR/data_xgobi/data_file

You should have set $XGOBIDIR as described in 1) above. data_file is the
name of the sample file you want to explore. There are other sample files
available in $XGOBIDIR/data_xgobi. Note that the files norm.500.5 and
norm.diffvar.500.5 are not available in the current XGobi distribution.

------------------------------------------------------------------------------
------------------------------------------------------------------------------
```

5) Use of ArcView/XGobi:

Before starting:

Type "setenv" and check that the environment variable HOST is set to
the name of the machine you are currently using. If this is not the case,
type the command "setenv HOST currentmachinename", i.e., use the name
of the machine you are currently working at.

Start ArcView:
  arcview &

In the prompt, click on "Open an existing project", then "OK".
CD into /home/symanzik/axx/av_dir/av_data and select the file
"AV2XG_DEMO_UTall.apr" (and click OK).

Select "File", then "Extensions", activate the "ArcView XGobi link",
and click "OK". A new menu button "XGobi" will appear in the command
line.

Activate a theme, e.g., "Places", select "XGobi", then "Basic Link",
hit the "Green Diamond", chose a field for row labels in XGobi
(e.g., Name2), and click "OK". Then start brushing on either side...

To stop XGobi, click on the "Red Diamond". Always stop XGobi first,
then quit ArcView. In case that XGobi crashes or you save your project
with the current symbols visible in the ArcView map view, you can
select "Edit" and "Select All Graphics", followed by "Edit" and
"Cut Graphics" which removes the symbols from the map view.

If you made changes to this demo project and want to keep these
changes for a future session, you have to use the option
"File" and "Save Project As" and indicate a directory for which you
have write permission. If you just try "File" and "Save Project",
you get the error message "ArcView cannot save or create this file"
since you do not have write permission to my directories.

----

6) On-line help on ArcView/XGobi/XploRe is available at
  http://www.math.usu.edu/~symanzik/axx/
and then following the link to the ArcView 3.1 - XGobi - XploRe
Environment Version 2.000 or by directly selecting
  http://www.math.usu.edu/~symanzik/axx/axx2000/

Unfortunately, there are some bugs in the current version
of ArcView/XGobi/XploRe. In particular, try to avoid brushing
too many points at a time in the Variogram Cloud, Lagged
Scatterplot, and Multivariate Variogram Cloud version of the link.
Instead, add points by slowly moving the brush.

Also, the CDF version of the link does not work at this stage.

----

7) XploRe is a statistical computing environment, developed at
Humboldt University zu Berlin, Germany, that has been added as
an analytical statistical package to the ArcView/XGobi link in 1997.

The main XploRe Web page is
  http://www.xplore-stat.de
and a US mirror is available at
  http://www.galaxy.gmu.edu/~xplore

In short, XploRe can be started from within XGobi and it contains
the spatial statistics functions described in Venables/Ripley (1999).
Thus, we have the link ArcView <-> XGobi <-> XploRe

A tutorial how to start XploRe from within XGobi and conduct
linked brushing is available at
  http://www.xplore-stat.de/tutorials/rpcframe1.html
  http://www.galaxy.gmu.edu/~xplore/tutorials/rpcframe1.html

A tutorial on spatial statistics in XploRe is available at
  http://www.xplore-stat.de/tutorials/spatialframe1.html
  http://www.galaxy.gmu.edu/~xplore/tutorials/spatialframe1.html

To start XploRe from within XGobi, select "Tools" and "Start XploRe"
within XGobi. To select variables to be passed from XGobi to XploRe,
select "Tools" and "Variable List" and mark those variables with a "+"
you want to use in XploRe. Then select "File" and "XploRe (pass variables)"
to pass the data into XploRe. On the XploRe side, you can then access
the data from XGobi via the variable name XGobiData.

B.8 Introduction to S+SpatialStats (2)

http://www.math.usu.edu/~symanzik/teaching/2000_spatial/splus/sspatialSCD

```
Stat 5810 Applied Spatial Statistics                                 Fall 2000


                    Introduction to S+SpatialStats (2)
                    ==================================

                             Wed 11/15/00


Dr. Juergen Symanzik

------------------------------------------------------------------------------

References:

Kaluzny, S. P., Vega, S. C., Cardoso, T. P., Shelly, A. A. (1998):
S+SpatialStats - User's Manual for Windows and UNIX, Springer, New York.

------------------------------------------------------------------------------

Analyzing Spatially Continuous Data:
====================================

Spatially continuous data sets included with S+SpatialStats:

aquifer
coal.ash
iron.ore
scallops

------------------------------------------------------------------------------

Basic Plots for  Spatially Continuous Data:

par(mfrow = c(1, 3))
plot(aquifer)

par(mfrow = c(2, 2), pty = "s")
plot(aquifer$easting, aquifer$northing)
hist(aquifer$head)

> summary (aquifer)
     easting            northing             head
    Min.:-145.20      Min.:  9.414      Min.:1024
 1st Qu.: -21.30   1st Qu.: 33.680   1st Qu.:1548
  Median:  11.66    Median: 59.160    Median:1797
    Mean:  16.89      Mean: 79.360      Mean:2002
 3rd Qu.:  70.90   3rd Qu.:131.800   3rd Qu.:2540
    Max.: 112.80      Max.:184.800      Max.:3571


aqu.interp _ interp (x = aquifer$easting, y = aquifer$northing,
  z = aquifer$head / 1000)
contour (aqu.interp)
image (aqu.interp)

par(mfrow = c(1, 1), pty = "s")
persp (aqu.interp, zlab = "Z / 1000", eye = c(300, -1500, 15))
cloud (head ~ easting * northing, data = aquifer)
```

```
-----------------------------------------------------------------------------

Variogram Clouds and Variograms:

aqu.varcloud _ variogram.cloud (head ~ loc(easting, northing), data = aquifer)

> aqu.varcloud[1:10,]
    distance iindex jindex    gamma
 1  79.25931      1      2 592960.5
 2  61.29274      1      3 240818.0
 3  79.92831      1      4 491040.5
 4  82.80194      1      5  42632.0
 5  79.52998      1      6  28322.0
 6  84.51441      1      7  58140.5
 7 100.20856      1      8  55444.5
 8 107.15201      1      9  31250.0
 9  89.97830      1     10      2.0
10 104.17854      1     11  35112.5


aqu.varmom _ variogram (head ~ loc(easting, northing), data = aquifer)

> aqu.varmom
     distance      gamma  np azimuth
 1   2.140951   7918.208  12       0
 2   6.768118  16691.490  52       0
 3  13.761070  18584.938  72       0
 4  20.591498  28335.207  87       0
 5  26.630336  29734.284  97       0
 6  33.884961  43063.465  85       0
 7  40.737678  60964.620  75       0
 8  47.272285  66935.436 110       0
 9  54.343456  81013.876 105       0
10  61.152581 101877.250 114       0
11  67.763697 123492.171 117       0
12  74.583602 114557.912 142       0
13  81.541152 170210.029 140       0
14  87.940181 212991.636 154       0
15  94.835540 238276.277 179       0
16 101.764432 299421.709 206       0
17 108.262561 329103.978 183       0
18 115.246276 370136.633 162       0
19 121.928166 351661.036 179       0
20 128.654657 399034.124 125       0

> summary(aqu.varmom)
Call:
variogram.formula(formula = head ~ loc(easting, northing), data = aquifer)
      lag nlag  maxdist
 6.776539   20 135.5308
     distance           gamma              np       azimuth
    Min.:  2.141      Min.:  7918      Min.: 12.0   0:20
 1st Qu.: 32.070   1st Qu.: 39730   1st Qu.: 86.5
  Median: 64.460    Median:108200    Median:115.5
    Mean: 64.490      Mean:153200      Mean:119.8
 3rd Qu.: 96.570   3rd Qu.:253600   3rd Qu.:156.0
    Max.:128.700      Max.:399000      Max.:206.0

aqu.varrobust _ variogram (head ~ loc(easting, northing), data = aquifer,
  method = "robust")


par(mfrow = c(2, 2))

plot (aqu.varcloud, main = "Variogram Cloud")
```

```
plot (aqu.varmom, main = "Variogram (MoM)")

plot (aqu.varrobust, main = "Variogram (Robust)")

-----------------------------------------------------------------------------

Theoretical Variogram Models and Fitting a Theoretical Variogram:

par (mfrow = c (2, 2))
vdist _ 1:240
vrange _ 120
vsill _ 600000
plot (vdist, exp.vgram(distance = vdist, range = vrange, sill = 600000),
  type = "l")
plot (vdist, spher.vgram(distance = vdist, range = vrange, sill = 600000),
  type = "l")
plot (vdist, gauss.vgram(distance = vdist, range = vrange, sill = 600000),
  type = "l")
plot (aqu.varrobust, main = "Variogram (Robust)")

> model.variogram(aqu.varrobust, fun = gauss.vgram,
    range = vrange / 2, sill = 600000)
Select a number to change a parameter (or 0 to exit):
 Current objective = 666341708467
1: range - current value: 60.
2: sill - current value: 600000
3: nugget - current value: 0
Selection: 1
New range : 90
Select a number to change a parameter (or 0 to exit):
 Current objective = 128252334042
1: range - current value: 90.
2: sill - current value: 600000
3: nugget - current value: 0
...

-----------------------------------------------------------------------------

Trend Surface Models:

aqu.lm1 _ lm (head ~ easting + northing, data = aquifer, x = T, y = T)

> aqu.lm1
Call:
lm(formula = head ~ easting + northing, data = aquifer, x = T, y = T)

Coefficients:
 (Intercept)   easting  northing
    2591.327 -6.751872 -5.986175

Degrees of freedom: 85 total; 82 residual
Residual standard error: 203.3075


aqu.lm2 _ lm (head ~ easting * northing, data = aquifer, x = T, y = T)

> aqu.lm2
Call:
lm(formula = head ~ easting * northing, data = aquifer, x = T, y = T)

Coefficients:
 (Intercept)   easting  northing easting:northing
    2626.994 -8.288185 -6.647954       0.02453368
```

```
Degrees of freedom: 85 total; 81 residual
Residual standard error: 191.9555


par (mfrow = c (2, 3))
plot (aqu.lm1)
plot (aqu.lm2)


par (mfrow = c (1, 2))
b0 _ coefficients(aqu.lm1)[1]
b1 _ coefficients(aqu.lm1)[2]
b2 _ coefficients(aqu.lm1)[3]
aqu.pred _ b0 + b1 * aquifer$easting + b2 * aquifer$northing

aqu.interppred _ interp (x = aquifer$easting, y = aquifer$northing,
  z = aqu.pred / 1000)

persp (aqu.interp, zlab = "Z / 1000", eye = c(300, -1500, 15))
persp (aqu.interppred, zlab = "Z / 1000", eye = c(300, -1500, 15))

------------------------------------------------------------------------------
```

B.9 Introduction to S+SpatialStats (3)

http://www.math.usu.edu/~symanzik/teaching/2000_spatial/splus/sspatialAREA

```
Stat 5810 Applied Spatial Statistics                              Fall 2000


                    Introduction to S+SpatialStats (3)
                    ==================================

                             Wed 11/29/00


Dr. Juergen Symanzik

-----------------------------------------------------------------------------

References:

Kaluzny, S. P., Vega, S. C., Cardoso, T. P., Shelly, A. A. (1998):
S+SpatialStats - User's Manual for Windows and UNIX, Springer, New York.

-----------------------------------------------------------------------------

Analyzing Area (Lattice) Data:
==============================

Area (lattice) data sets included with S+SpatialStats:

coal.ash
iron.ore
sids  (and sids.neighbor)
sids2 (and sids.neighbor)
wheat

-----------------------------------------------------------------------------

Basic Plots for Area Data:

par(mfrow = c(1, 4))
plot(iron.ore)

par(mfrow = c(2, 2), pty = "s")
plot(iron.ore$easting, iron.ore$northing)
hist(iron.ore$ore)

> summary (iron.ore)
     easting          northing          ore          residuals
    Min.: 1.000      Min.:1.000      Min.:45.40      Min.:-8.2800
 1st Qu.: 5.750   1st Qu.:3.000   1st Qu.:52.58   1st Qu.:-1.5060
  Median: 9.000    Median:5.000    Median:55.85    Median: 0.0000
    Mean: 9.089      Mean:5.009      Mean:54.97      Mean:-0.2546
 3rd Qu.:12.250   3rd Qu.:7.000   3rd Qu.:57.80   3rd Qu.: 1.4580
    Max.:17.000      Max.:9.000      Max.:60.80      Max.: 5.8970

iron.interp _ interp (x = iron.ore$easting, y = iron.ore$northing,
  z = iron.ore$ore - 45)
contour (iron.interp)
image (iron.interp)

par(mfrow = c(1, 1), pty = "s")
persp (iron.interp, zlab = "Z - 45", eye = c(30, -15, 30))
cloud (ore ~ easting * northing, data = iron.ore)
```

```
------------------------------------------------------------------------------

Median Polish:

iron.mp _ twoway(ore ~ easting + northing, data = iron.ore)

> iron.mp
$grand:
[1] 54.51337

$row:
       1         2         3          4        5 6          7        8
 -3.55307 -3.173534 -1.553066 -0.1936951 1.546783 0 0.04672623 2.926464

        9
 4.186394

$col:
        1         2           3 4         5         6        7          8
 -7.289996 -4.939993 -0.03676397 0 -1.089887 -0.330037 1.300032 -0.2198053

          9      10       11       12       13       14         15       16
 -0.1797631 2.16008 2.339876 4.210004 2.139691 0.539767 -0.4367636 2.763235

       17
 2.080321

$residuals:
  [1]  1.429686e+00 -1.429686e+00  2.479685e+00 -2.479685e+00 -9.030775e-01
  [6] -1.523543e+00 -7.829162e-01  1.766081e-01 -1.766081e-01  5.766678e-01
  ...
[111]  0.000000e+00 -5.340417e+00

$call:
twoway.formula(formula = ore ~ easting + northing, data = iron.ore)

# calculate the predicted data

iron.pred _ iron.ore$ore - iron.mp$residuals

# convert vectors to matrices, determine range of both matrices,
# and plot original data and predicted data

iron.mat _ tapply(iron.ore$ore,
             list(iron.ore$easting, iron.ore$northing),
             function(x) x
           )

iron.predmat _ tapply(iron.pred,
             list(iron.ore$easting, iron.ore$northing),
             function(x) x
           )

iron.resmat _ tapply(iron.mp$residuals,
             list(iron.ore$easting, iron.ore$northing),
             function(x) x
           )

zmin _ min(iron.mat, iron.predmat, na.rm = T)
zmax _ max(iron.mat, iron.predmat, na.rm = T)
zminres _ min(iron.resmat, na.rm = T)
zmaxres _ max(iron.resmat, na.rm = T)

par(mfrow = c(2, 2), pty = "s")
image(iron.mat, zlim = c(zmin, zmax), main = "Observed")
```

```
image(iron.predmat, zlim = c(zmin, zmax), main = "Predicted")
image(iron.resmat, zlim = c(zminres, zmaxres), main = "Residuals")

------------------------------------------------------------------------------

Finding Spatial Neighbors:

# for a full regular grid

iron.neigh _ neighbor.grid (nrow = 17, ncol = 9,
  neighbor.type = "first.order")

> iron.neigh[1:10,]
Total number of spatial units =  153
(Matrix was NOT defined as symmetric)
   row.id col.id weights matrix
 1      1      2       1      1
 2      1     18       1      1
 3      2      1       1      1
 4      2      3       1      1
 5      2     19       1      1
 6      3      2       1      1
 7      3      4       1      1
 8      3     20       1      1
 9      4      3       1      1
10      4      5       1      1


# for an irregular structure

iron.neighirreg _ find.neighbor (cbind(iron.ore$easting, iron.ore$northing),
  quadtree = quad.tree(cbind(iron.ore$easting, iron.ore$northing)),
  max.dist = 1.1)

> iron.neighirreg[1:10,]
   index1 index2 distances
 1      1      1         0
 2      1      2         1
 3      1      3         1
 4      2      2         0
 5      2      1         1
 6      2      4         1
 7      3      3         0
 8      3      4         1
 9      3      1         1
10      3      6         1

iron.neighirreg1 _ iron.neighirreg[iron.neighirreg[,3] > 0,]

> iron.neighirreg1[1:10,]
   index1 index2 distances
 2      1      2         1
 3      1      3         1
 5      2      1         1
 6      2      4         1
 8      3      4         1
 9      3      1         1
10      3      6         1
12      4      3         1
13      4      2         1
14      4      7         1

iron.neighirreg2 _ spatial.neighbor(row.id = iron.neighirreg1[,1],
  col.id = iron.neighirreg1[,2],
  matrix.id = rep(1, length(iron.neighirreg1[,1])))
```

```
> iron.neighirreg2[1:10,]
Total number of spatial units =  112
(Matrix was NOT defined as symmetric)
   row.id col.id weights matrix
 2      1      2       1      1
 3      1      3       1      1
 5      2      1       1      1
 6      2      4       1      1
 8      3      4       1      1
 9      3      1       1      1
10      3      6       1      1
12      4      3       1      1
13      4      2       1      1
14      4      7       1      1

------------------------------------------------------------------------------

Spatial Autocorrelation:

iron.cor1 _ spatial.cor(iron.ore$ore, neighbor = iron.neighirreg2,
  statistic = "moran", sampling = "free", npermutes = 1000)

> iron.cor1

        Spatial Correlation Estimate

Statistic = "moran" Sampling = "free"

Correlation =  0.4075
Variance    =  0.004903
Std. Error  =  0.07002

Normal statistic =  5.949
Normal p-value (2-sided) =  2.705e-9

Null Hypothesis:  No spatial autocorrelation

Summary of the permutation-correlations :
    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 -0.1990 -0.0538 -0.0100 -0.0081  0.0334  0.2381

permutation p-value = 0

----

iron.cor2 _ spatial.cor(iron.ore$ore, neighbor = iron.neighirreg2,
  statistic = "geary", sampling = "free", npermutes = 1000)

> iron.cor2

        Spatial Correlation Estimate

Statistic = "geary" Sampling = "free"

Correlation =  0.5464
Variance    =  0.005537
Std. Error  =  0.07441

Normal statistic =  -6.097
Normal p-value (2-sided) =  1.082e-9

Null Hypothesis:  No spatial autocorrelation

Summary of the permutation-correlations :
```

```
  Min. 1st Qu. Median  Mean 3rd Qu.  Max.
 0.790 0.949   1.000  1.000 1.050   1.259

permutation p-value = 0

----

iron.cor3 _ spatial.cor(iron.mp$residuals, neighbor = iron.neighirreg2,
  statistic = "moran", sampling = "free", npermutes = 1000)

> iron.cor3

        Spatial Correlation Estimate

Statistic = "moran" Sampling = "free"

Correlation =  0.0712
Variance    =  0.004903
Std. Error  =  0.07002

Normal statistic =  1.146
Normal p-value (2-sided) =  0.252

Null Hypothesis:  No spatial autocorrelation

Summary of the permutation-correlations :
    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 -0.2374 -0.0568 -0.0093 -0.0095  0.0368  0.2284

permutation p-value = 0.129

----

iron.cor4 _ spatial.cor(iron.mp$residuals, neighbor = iron.neighirreg2,
  statistic = "geary", sampling = "free", npermutes = 1000)

> iron.cor4

        Spatial Correlation Estimate

Statistic = "geary" Sampling = "free"

Correlation =  0.8555
Variance    =  0.005537
Std. Error  =  0.07441

Normal statistic =  -1.942
Normal p-value (2-sided) =  0.05209

Null Hypothesis:  No spatial autocorrelation

Summary of the permutation-correlations :
  Min. 1st Qu. Median  Mean 3rd Qu.  Max.
 0.762 0.949   1.002  1.003 1.056   1.271

permutation p-value = 0.025

------------------------------------------------------------------------------

Micromaps:

source('/home/symanzik/splus/spat_penn1.s')

The following code will be executed:
```

```
# File:         penn1.s
# Programmer:   Dan Carr
# Date:         Revised June 6, 1999
# Purpose       Pennsylvania LM dot plot for benzene

# The data: penn.ben
# and boundary files: penn.bnd
# are in a dump file

data.restore('/home/symanzik/splus/penn1_panel.dmp')
data.restore('/home/symanzik/splus/penn1.dmp')

#data.restore('penn1_panel.dmp')
#data.restore('penn1.dmp')


# History of data file creation
# benzene data
#penn.ben <- read.table(file='penn_benzene.txt',sep=',')
#names(penn.ben) <- c('Tracts', 'Mean', 'Median', 'Min', 'P25', 'P75', 'Max')

vals <- penn.ben$Median
tnams <- row.names(penn.ben)
nams <- penn.bnd$countyid[tnams]


#sorting
ord <- order(vals)
ord <- rev(ord)
vals <- vals[ord]
nams <- nams[ord]
tnams <- tnams[ord]

# scaling
rx <- range(vals)
rx <- mean(rx) + .57*diff(rx)*c(-1,1)
ry <- c(0,1)
a <- .25
grid.x <- inbounds(rx)


id <- penn.bnd$id
bnd.rx <- penn.bnd$rx
bnd.rx <- mean(bnd.rx)+ 1.04*diff(bnd.rx)*c(-.5,.5)

bnd.ry <- penn.bnd$ry
bnd.ry <- mean(bnd.ry)+ 1.04*diff(bnd.ry)*c(-.5,.5)


micolors <- matrix(c(
  .00, .00, .00,
 1.00,1.00,1.00,
  .78, .78, .78,
 1.00, .1, .1 ,
 1.00, .50, .00,
  .10,1.00, .00,
  .10, .50,1.00,
  .70, .40,1.00,
  .20, .20, .20,
 1.00,1.00, .80,
  .78, .78, .78,
  .50, .50, .50,
  .80, .50,1.00),ncol=3,byrow=T)
ps.options(black.and.white="false",rasters=600)
#postscript(file='penn1.ps',colors=micolors,hori=F,width=8.5,height=10)
```

```
motif("hpage")
par(xpd=T)



if(names(dev.cur())=='postscript'){
   pch<- 183
   outline <- T
} else {
   pch<- 16
   outline <- F
}

panels <- panel.layout(nrow=8,ncol=6,
                        col.size=c(1.5,1.0,1.3,1.5,1.0,1.3),
                        col.sep=c(0,0,0,.15,0,0,0),
                        row.sep=c(0,0,0,0,.15,0,0,0,0),bottom.mar=.2,
                        left.mar=0,top.mar=.7,borders=c(.25,.25,.25,.25))

pan.outline <- panel.layout(nrow=2,ncol=6,
                        col.size=c(1.5,1.0,1.3,1.5,1.0,1.3),
                        col.sep=c(0,0,0,.15,0,0,0),
                        row.sep=c(0,.15,0),bottom.mar=.2,
                        left.mar=0,top.mar=.7,borders=c(.25,.25,.25,.25))

pan.title <- panel.layout(nrow=1,ncol=1,bottom.mar=.2,
                        left.mar=0,top.mar=.7,borders=c(.25,.25,.25,.25))


ie <- cumsum(c(rep(c(4,4,4,5),3),c(4,4,4,4)))
npanel <- length(ie)
ib <- c(1,ie[-npanel]+1)

for (i in 1:length(ib)){
   subs <- seq(ib[i],ie[i])
   gnams <- nams[subs]
   txtnams <- tnams[subs]
   if(i <= 8){
     pan.r <- i
     pan.c1 <- 1
     pan.c2 <- 2
     pan.c3 <- 3
     if(i<=4)cont <- nams[ib[1]:ie[4]] else
             cont <- nams[ib[5]:ie[8]]
   } else {
     pan.r <- i-8
     pan.c1 <- 4
     pan.c2 <- 5
     pan.c3 <- 6
     if(i <= 12) cont <- nams[ib[9]:ie[12]] else
                 cont <- nams[ib[13]:ie[16]]
   }
#plot maps
   panel.select(panels,pan.r,pan.c1)
   panel.scale(bnd.rx,bnd.ry)

#  background
   back <- is.na(match(penn.bnd$fid,cont))
   tx <- penn.bnd$x[back]
   ty <- penn.bnd$y[back]
   polygon(tx,ty,col=11,border=F)
   polygon(tx,ty,col=2,density=0,lwd=0)

#  midground
```

```
   tx <- penn.bnd$x[!back]
   ty <- penn.bnd$y[!back]
   polygon(tx,ty,col=10,border=F)
   polygon(tx,ty,col=1,density=0,lwd=0)

#  highlight
   good <- !is.na(match(penn.bnd$fid,gnams))
   tx <- penn.bnd$x[good]
   ty <- penn.bnd$y[good]
   pen <- match(penn.bnd$id,gnams,0)
   pen <- pen[pen>0]+3
   polygon(tx,ty,col=pen,border=F)
   polygon(tx,ty,col=1,density=0)

#  final outline
  polygon(penn.bnd$out.x,penn.bnd$out.y,density=0,lwd=0)

# _________________ plot id's___________________
   panel.select(panels,pan.r,pan.c2)
   panel.scale(c(0,1),ry)
   nums <- rev(seq(along=subs))
   n <- length(nums)
   y <- (nums-a)/(n+1-2*a)
   text(rep(.025,n),y,txtnams,adj=0,cex=.65)

# plot dots
   panel.select(panels,pan.r,pan.c3)
   panel.scale(rx,ry)
   panel.fill(col=3)
   panel.mygrid(x=grid.x,col=2)
   panel.outline(col=2)
   if(pan.r==8){
      axis(side=1,at=grid.x,labels=as.character(grid.x),cex=.7,mgp=c(2,.2,0))
      text(mean(rx),-.3," g/m ",adj=.5,cex=.7)
      points(mean(rx)-.12*diff(rx),-.34,pch=109,font=13,cex=.7)
      text(mean(rx)+.13*diff(rx),-.27,"3",adj=.5,cex=.5)
   }
   nums <- rev(seq(along=subs))
   n <- length(nums)+1
   y <- (nums-a)/(n-2*a)
   for (j in 1:length(nums)){
      points(vals[subs[j]],y[j],col=1,cex=1.9,pch=pch)
      points(vals[subs[j]],y[j],col=j+3,cex=1.6,pch=pch)
   }
}
   panel.select(pan.outline,1,3)
   panel.scale(rx,ry)
   panel.outline()

   panel.select(pan.outline,2,3)
   panel.scale(rx,ry)
   panel.outline()
   panel.select(pan.outline,1,6)
   panel.scale(rx,ry)
   panel.outline()
   panel.select(pan.outline,2,6)
   panel.scale(rx,ry)
   panel.outline()

   panel.select(pan.title,margin='top')
   panel.scale()
   text(.5,.9,'Pennslyvania: Median of Modeled 1990 Census Tract Benzene Concentrations',adj=.5)
   text(.25,.3,'Above County Median',cex=.9)
   text(.75,.3,'Below County Median',cex=.9)
```
