Spatial Statistics

This library is in two parts, on kriging and on point processes.

1 Kriging

We provide three datasets, but `topo` is used most. There functions `correlogram(nint)` and `variogram(nint)` to plot the empirical functions over `nint` equally spaced intervals. You need to write your own function for the covariance model, or use `expcov, gaucov` or `sphercov`.

The functions `trmat, ppmat` and `semat` evaluate the trend and kriged surface and the predicition standard error on a grid, for plotting via `contour, image` or `persp`.

`openlook()`	
`library(spatial)`	
`par(pty="s")`	Use square plots, as plot region is square
`topo.ls <- surf.ls(2, topo)`	quadratic trend surface
`correlogram(topo.ls, 25)`	
`d <- seq(0,7,0.1);lines(d, expcov(d, 0.7))`	and compare with exponential model
`topo.gls <- surf.gls(2, expcov, topo, d=0.7)`	by GLS
`trsurf <- trmat(topo.gls, 0, 6.5, 0, 6.5, 50)`	evaluate it at 51×51 grid in square
`contour(trsurf)`	contour map
`points(topo)`	add data points
`image(trsurf)`	grey-level map
`points(topo)`	
`prsurf <- prmat(topo.gls, 0, 6.5, 0, 6.5, 50)`	kriged surface
`contour(prsurf)`	
`points(topo)`	
`image(prsurf)`	
`points(topo)`	
`sesurf <- semat(topo.gls, 0,6.5,0,6.5,30)`	SE of prediction surface
`contour(sesurf, levels=c(22,25))`	choose own contours
`points(topo)`	
`persp(trsurf)`	perspective plot of trend surface
`pout <- persp(prsurf)`	and of predicted surface
`points(perspp(topo$x, topo$y, topo$z, pout))`	and put points on

There are three other datasets you can try. `shkap` is a set of permeability measurements at an oil field in the USSR. `npr` contains both permability and porosity measurements on the *US Naval Petroleum Reserve 1*. For permeability measurements you may wish to transform, e.g. take logs.

You may also like to try S's `interp` function:

```
insurf <- interp(topo$x, topo$y, topo$z)
contour(insurf)
persp(insurf)
```

2 Point Processes

The software will perform K-analyses for real data and for simulations, so one can compare data with models. All plots are in fact of $L = \sqrt{K/\pi}$.

The S-Plus functions available are:

`pp <- ppinit("file")`	initializes a `pp` object
`ppset(xl=0, xu=1, yl=0, yu=1)`	initializes a blank region, for simulations
`Kfn(pp, fs, k=100)`	actually L-fn with full scale `fs` and `k` divisions
`Kenvl(fs, nsim, simfn)`	finds max and min of `nsim` simulations
`Kaver(fs, nsim, simfn)`	finds averages
`Psim(n)`	Poisson process conditional on `n`
`Strauss(n, c, r)`	
`SSI(n, r)`	
`pplik(pp, r)`	estaimte c for Strauss process

The data can, of course, be plotted by `points(pp)`.

A sample session is:

```
openlook()
library(spatial)
towns <- ppinit("towns.dat")
par(pty="s")
plot(Kfn(towns, 40), type="s")
plot(Kfn(towns, 10), type="s", xlab="distance", ylab="L(t)")
for(i in 1:10) lines(Kfn(Psim(69), 10))
lims <- Kenvl(10,100,Psim(69))
lines(lims$x,lims$l, lty=2)
lines(lims$x,lims$u, lty=2)
lines(Kaver(10,25,Strauss(69,0.5,3.5)))
```

The possible simulated models are all conditional on the number of points, and are *Binomial*, `Psim`, *Matern's sequential inhibition* process, `SSI` and the *Strauss* process, `Strauss`. The Binomial process is just a Poisson process conditioned on the total number of points, which are then uniform and independent in a study region, so the counts in sub-regions are Binomial. The SSI process puts down points sequentially, never putting a point within distance `r` of an existing point but otherwise at random. The Strauss process with `c=0` conditions a Poisson process on having no pair of points closer than `r` apart; in general it has density of a set of points $\{x_1, \ldots, x_n\}$ proportional to $c^{\#\text{ R-close pairs of points}}$.

Datasets

Datasets available in the library include:

`caveolae.dat`	on human muscle cell walls (big)
`davis.dat`	magnetite crystals in a rock
`drumlin.dat`	in Co. Down
`eagles.dat`	Golden eagles nest sites in Scotland
`fig1b.dat`	Synthetic examples
`fig1c.dat`	
`fig2a.dat`	
`fig2b.dat`	
`fig3a.dat`	
`fig3b.dat`	
`fig3c.dat`	
`grocery.dat`	Grocery stores in Ljubljana
`nztrees.dat`	NZ trees

`pereg.dat`	peregrine falcons nests in Scotland
`pines.dat`	Swedish pine trees
`redwood.dat`	redwood tree seedlings
`schools.dat`	
`stowns1.dat`	Spanish towns, re-digitized
`tokyo.dat`	meat stores in Tokyo
`towns.dat`	Spanish market towns

The library is automatically searched for files, then the current directory (or the path specified). Sources of the datasets are in file `PP.files`. Those not using a library may like to set suitably

```
assign("sp.lib.name", "any path/", frame=0)
```