Package ‘spatial’

February 20, 2015

Priority recommended
Version 7.3-9
Date 2015-02-10
Depends R (>= 3.0.0), graphics, stats, utils
Suggests MASS
Description Functions for kriging and point pattern analysis.
Title Functions for Kriging and Point Pattern Analysis
LazyLoad yes
ByteCompile yes
License GPL-2 | GPL-3
URL http://www.stats.ox.ac.uk/pub/MASS4/
NeedsCompilation yes
Author Brian Ripley [aut, cre, cph],
       Roger Bivand [ctb],
       William Venables [cph]
Maintainer Brian Ripley <ripley@stats.ox.ac.uk>
Repository CRAN
Date/Publication 2015-02-11 09:30:11

R topics documented:

anova.trls .................................................. 2
correlogram .................................................. 3
expcov ......................................................... 4
Kaver .......................................................... 5
Kenvl ........................................................... 6
Kfn .............................................................. 7
ppgetregion ................................................... 8
ppinit .......................................................... 8
pplik ........................................................... 9
Description

Compute analysis of variance tables for one or more fitted trend surface model objects; where \texttt{anova.trls} is called with multiple objects, it passes on the arguments to \texttt{anovalist.trls}.

Usage

```r
## S3 method for class 'trls'
anova(object, ...)
anovalist.trls(object, ...)
```

Arguments

- `object` A fitted trend surface model object from \texttt{surf.ls}
- `...` Further objects of the same kind

Value

\texttt{anova.trls} and \texttt{anovalist.trls} return objects corresponding to their printed tabular output.

References


See Also

\texttt{surf.ls}
correlogram

Examples

```r
library(stats)
data(topo, package="MASS")
topo0 <- surf.ls(0, topo)
topo1 <- surf.ls(1, topo)
topo2 <- surf.ls(2, topo)
topo3 <- surf.ls(3, topo)
topo4 <- surf.ls(4, topo)
anova(topo0, topo1, topo2, topo3, topo4)
summary(topo4)
```

---

**Description**

Compute spatial correlograms of spatial data or residuals.

**Usage**

```r
correlogram(krig, nint, plotit = TRUE, ...)
```

**Arguments**

- `krig` trend-surface or kriging object with columns `x`, `y`, and `z`
- `nint` number of bins used
- `plotit` logical for plotting
- `...` parameters for the plot

**Details**

Divides range of data into `nint` bins, and computes the covariance for pairs with separation in each bin, then divides by the variance. Returns results for bins with 6 or more pairs.

**Value**

- `x` and `y` coordinates of the correlogram, and `cnt`, the number of pairs averaged per bin.

**Side Effects**

Plots the correlogram if `plotit = TRUE`.

**References**


See Also

variogram

Examples

```r
data(topo, package="MASS")
topo.kr <- surf.ls(2, topo)
correlogram(topo.kr, 25)
d <- seq(0, 7, 0.1)
lines(d, expcov(d, 0.7))
```

### expcov

**Spatial Covariance Functions**

**Description**

Spatial covariance functions for use with `surf.gls`.

**Usage**

```r
expcov(r, d, alpha = 0, se = 1)
gaucov(r, d, alpha = 0, se = 1)
sphercov(r, d, alpha = 0, se = 1, D = 2)
```

**Arguments**

- `r` vector of distances at which to evaluate the covariance
- `d` range parameter
- `alpha` proportion of nugget effect
- `se` standard deviation at distance zero
- `D` dimension of spheres.

**Value**

vector of covariance values.

**References**


See Also

`surf.gls`
Kaver

Examples

data(topo, package="MASS")
topo.kr <- surf.ls(2, topo)
correlogram(topo.kr, 25)
d <- seq(0, 7, 0.1)
lines(d, expcov(d, 0.7))

kaver

Average K-functions from Simulations

Description

Forms the average of a series of (usually simulated) K-functions.

Usage

Kaver(fs, nsim, ...)

Arguments

fs full scale for K-fn
nsim number of simulations
... arguments to simulate one point process object

Value

list with components x and y of the average K-fn on L-scale.

References


See Also

Kfn, Kenvl

Examples

towns <- ppinit("towns.dat")
par(pty="s")
plot(Kfn(towns, 40), type="b")
plot(Kfn(towns, 10), type="b", xlab="distance", ylab="L(t)"
for(i in 1:10) lines(Kfn(Psim(69), 10))
lims <- Kenvl(10,100,Psim(69))
lines(lims$x,lims$lower, lty=2, col="green")
lines(lims$x,lims$upper, lty=2, col="green")
lines(Kaver(10,25,Strauss(69,0.5,3.5)), col="red")
**Kenvl**

*Compute Envelope and Average of Simulations of K-fns*

**Description**

Computes envelope (upper and lower limits) and average of simulations of K-fns

**Usage**

```r
Kenvl(fs, nsim, ...)
```

**Arguments**

- `fs`: full scale for K-fn
- `nsim`: number of simulations
- `...`: arguments to produce one simulation

**Value**

list with components

- `x`: distances
- `lower`: min of K-fns
- `upper`: max of K-fns
- `aver`: average of K-fns

**References**


**See Also**

`kfn`, `kaver`

**Examples**

```r
towns <- ppinit("towns.dat")
par(pty="s")
plot(Kfn(towns, 40), type="b")
plot(Kfn(towns, 10), type="b", xlab="distance", ylab="L(t")"
for(i in 1:10) lines(Kfn(Psim(69), 10))
lims <- Kenvl(10,100,Psim(69))
lines(lims$x,lims$lower, lty=2, col="green")
lines(lims$x,lims$upper, lty=2, col="green")
lines(Kaver(10,25,Strauss(69,0.5,3,5)), col="red")
```
Kfn

Compute K-fn of a Point Pattern

Description

Actually computes $L = \sqrt{K/\pi}$.

Usage

Kfn(pp, fs, k=100)

Arguments

pp      a list such as a pp object, including components x and y
fs      full scale of the plot
k       number of regularly spaced distances in (0, fs)

Details

relies on the domain D having been set by ppinit or ppregion.

Value

A list with components

x       vector of distances
y       vector of L-fn values
k       number of distances returned – may be less than k if fs is too large
dmin    minimum distance between pair of points
lm      maximum deviation from $L(t) = t$

References


See Also

ppinit, ppregion, Kaver, Kenvl

Examples

towns <- ppinit("towns.dat")
par(pty="s")
plot(Kfn(towns, 10), type="s", xlab="distance", ylab="L(t)")
ppgetregion  
*Get Domain for Spatial Point Pattern Analyses*

**Description**
Retrieves the rectangular domain \((x_L, x_U) \times (y_L, y_U)\) from the underlying C code.

**Usage**

```r
ppgetregion()
```

**Value**
A vector of length four with names `c("xl", "xu", "yL", "yu")`.

**References**

**See Also**

- `ppregion`

ppinit  
*Read a Point Process Object from a File*

**Description**
Read a file in standard format and create a point process object.

**Usage**

```r
ppinit(file)
```

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>file</code></td>
<td>string giving file name</td>
</tr>
</tbody>
</table>

**Details**
The file should contain
- the number of points
- a header (ignored)
- `xL xu yL yU scale` 
x y (repeated n times)
pplik

Value

class "pp" object with components x, y, x1, xu, yl, yu

Side Effects

Calls ppregion to set the domain.

References


See Also

ppregion

Examples

towns <- ppinit("towns.dat")
par(pty="s")
plot(Kfn(towns, 10), type="b", xlab="distance", ylab="L(t)")

pplik

Pseudo-likelihood Estimation of a Strauss Spatial Point Process

Description

Pseudo-likelihood estimation of a Strauss spatial point process.

Usage

pplik(pp, R, ng=50, trace=FALSE)

Arguments

pp a pp object
R the fixed parameter R
ng use a ng x ng grid with border R in the domain for numerical integration.
trace logical? Should function evaluations be printed?

Value

estimate for c in the interval [0, 1].

References

See Also

Strauss

Examples

```r
pines <- ppinit("pines.dat")
pplik(pines, 0.7)
```

---

**ppregion**  
*Set Domain for Spatial Point Pattern Analyses*

**Description**

Sets the rectangular domain \((x_l, x_u) \times (y_l, y_u)\).

**Usage**

```r
ppregion(x_l = 0, x_u = 1, y_l = 0, y_u = 1)
```

**Arguments**

- `x_l`  
  Either `x_l` or a list containing components `x_l`, `x_u`, `y_l`, `y_u` (such as a point-process object)

- `x_u`

- `y_l`

- `y_u`

**Value**

`none`

**Side Effects**

Initializes variables in the C subroutines.

**References**


**See Also**

`ppinit`, `ppgetregion`
predict.trls   Predict method for trend surface fits

Description

Predicted values based on trend surface model object

Usage

## S3 method for class 'trls'
predict(object, x, y, ...)

Arguments

- object: Fitted trend surface model object returned by `surf.ls`
- x: Vector of prediction location eastings (x coordinates)
- y: Vector of prediction location northings (y coordinates)
- ...: further arguments passed to or from other methods.

Value

`predict.trls` produces a vector of predictions corresponding to the prediction locations. To display the output with `image` or `contour`, use `trmat` or convert the returned vector to matrix form.

References


See Also

`surf.ls`, `trmat`

Examples

data(topo, package="MASS")
  topo2 <- surf.ls(2, topo)
  topo4 <- surf.ls(4, topo)
  x <- c(1.78, 2.21)
  y <- c(6.15, 6.15)
  z2 <- predict(topo2, x, y)
  z4 <- predict(topo4, x, y)
  cat("2nd order predictions:", z2, 
       "4th order predictions:", z4, 
       "n")
prmat

Evaluate Kriging Surface over a Grid

Description
Evaluate Kriging surface over a grid.

Usage
prmat(obj, xl, xu, yl, yu, n)

Arguments
- obj: object returned by surf.gls
- xl: limits of the rectangle for grid
- xu: limits of the rectangle for grid
- yl: limits of the rectangle for grid
- yu: limits of the rectangle for grid
- n: use n x n grid within the rectangle

Value
list with components x, y and z suitable for contour and image.

References

See Also
surf.gls, trmat, semat

Examples
```
data(topo, package="MASS")
topo.kr <- surf.gls(2, expcov, topo, d=0.7)
prsurf <- prmat(topo.kr, 0, 6.5, 0, 6.5, 50)
contour(prsurf, levels=seq(700, 925, 25))```
**Psim**

*Simulate Binomial Spatial Point Process*

**Description**

Simulate Binomial spatial point process.

**Usage**

```r
Psim(n)
```

**Arguments**

- `n` number of points

**Details**

relies on the region being set by `ppinit` or `ppregion`.

**Value**

list of vectors of x and y coordinates.

**Side Effects**

uses the random number generator.

**References**


**See Also**

`SSI`, `Strauss`

**Examples**

```r
towns <- ppinit("towns.dat")
par(pty="s")
plot(Kfn(towns, 10), type="s", xlab="distance", ylab="L(t)")
for(i in 1:10) lines(Kfn(Psim(69), 10))
```
semat

Evaluate Kriging Standard Error of Prediction over a Grid

Description

Evaluate Kriging standard error of prediction over a grid.

Usage

semat(obj, xl, xu, yl, yu, n, se)

Arguments

obj          object returned by surf.gls
xl           limits of the rectangle for grid
xu           yl
yu           use n x n grid within the rectangle
n            standard error at distance zero as a multiple of the supplied covariance. Otherwise estimated, and it assumed that a correlation function was supplied.
se

Value

list with components x, y and z suitable for contour and image.

References


See Also

surf.gls, trmat, prmat

Examples

data(topo, package="MASS")
topo.kr <- surf.gls(2, expcov, topo, d=0.7)
prsurf <- prmat(topo.kr, 0, 6.5, 0, 6.5, 50)
contour(prsurf, levels=seq(700, 925, 25))
ssurf <- semat(topo.kr, 0, 6.5, 0, 6.5, 30)
contour(ssurf, levels=c(22,25))
**Description**

Simulates SSI (sequential spatial inhibition) point process.

**Usage**

`SSI(n, r)`

**Arguments**

- `n`: number of points
- `r`: inhibition distance

**Details**

uses the region set by `ppinit` or `ppregion`.

**Value**

list of vectors of x and y coordinates

**Side Effects**

uses the random number generator.

**Warnings**

will never return if `r` is too large and it cannot place `n` points.

**References**


**See Also**

`psim`, `strauss`

**Examples**

```r
towns <- ppinit("towns.dat")
par(pty = "s")
plot(Kfn(towns, 10), type = "b", xlab = "distance", ylab = "L(t)")
lines(Kaver(10, 25, SSI(69, 1.2)))
```
Strauss

Simulates Strauss Spatial Point Process

Description

Simulates Strauss spatial point process.

Usage

strauss(n, c=0, r)

Arguments

- `n`: number of points
- `c`: parameter c in [0, 1]. `c = 0` corresponds to complete inhibition at distances up to r.
- `r`: inhibition distance

Details

Uses spatial birth-and-death process for 4n steps, or for 40n steps starting from a binomial pattern on the first call from an other function. Uses the region set by `ppinit` or `ppregion`.

Value

list of vectors of x and y coordinates

Side Effects

uses the random number generator

References


See Also

`psim`, `ssi`

Examples

towns <- ppinit("towns.dat")
par(pty="s")
plot(Kfn(towns, 10), type="b", xlab="distance", ylab="L(t)")
lines(Kaver(10, 25, Strauss(69,0.5,3.5)))
**surf.gls**

*Fits a Trend Surface by Generalized Least-squares*

**Description**

Fits a trend surface by generalized least-squares.

**Usage**

```
surf.gls(np, covmod, x, y, z, nx = 1000, ...)
```

**Arguments**

- `np`: degree of polynomial surface
- `covmod`: function to evaluate covariance or correlation function
- `x`: x coordinates or a data frame with columns `x`, `y`, `z`
- `y`: y coordinates
- `z`: z coordinates. Will supersede x$z
- `nx`: Number of bins for table of the covariance. Increasing adds accuracy, and increases size of the object.
- `...`: parameters for covmod

**Value**

list with components

- `beta`: the coefficients
- `x`
- `y`
- `z`: and others for internal use only.

**References**


**See Also**

`trmat`, `surf.ls`, `prmat`, `semat`, `expcov`, `gaucov`, `sphercov`
Examples

```r
library(MASS)  # for eqscplot
data(topo, package="MASS")
topo.kr <- surf.gls(2, expcov, topo, d=0.7)
trsurf <- trmat(topo.kr, 0, 6.5, 0, 6.5, 50)
eqscplot(trsurf, type = "n")
contour(trsurf, add = TRUE)

prsurf <- prmat(topo.kr, 0, 6.5, 0, 6.5, 50)
contour(prsurf, levels=seq(700, 925, 25))
ifesurf <- semat(topo.kr, 0, 6.5, 0, 6.5, 30)
eqscplot(sesurf, type = "n")
contour(sesurf, levels = c(22, 25), add = TRUE)
```

surf.ls  

*Fits a Trend Surface by Least-squares*

Description

Fits a trend surface by least-squares.

Usage

```r
surf.ls(np, x, y, z)
```

Arguments

- `np` degree of polynomial surface
- `x` x coordinates or a data frame with columns x, y, z
- `y` y coordinates
- `z` z coordinates. Will supersede x$z

Value

- list with components
  - `beta` the coefficients
  - `x`
  - `y`
  - `z` and others for internal use only.

References


See Also

trls.influence

Examples

library(MASS)  # for eqscplot
data(topo, package="MASS")
topo.kr <- surf.ls(2, topo)
trsurf <- trmat(topo.kr, 0, 6.5, 0, 6.5, 50)
eqscplot(trsurf, type = "n")
contour(trsurf, add = TRUE)
points(topo)
eqscplot(trsurf, type = "n")
contour(trsurf, add = TRUE)
plot(topo.kr, add = TRUE)
title(xlab = "Circle radius proportional to Cook's influence statistic")

trls.influence

Regression diagnostics for trend surfaces

Description

This function provides the basic quantities which are used in forming a variety of diagnostics for checking the quality of regression fits for trend surfaces calculated by surf.ls.

Usage

trls.influence(object)

## S3 method for class 'trls'
plot(x, border = "red", col = NA, pch = 4, cex = 0.6,
    add = FALSE, div = 8, ...)

Arguments

object, x Fitted trend surface model from surf.ls
div scaling factor for influence circle radii in plot.trls
add add influence plot to existing graphics if TRUE
border, col, pch, cex, ...
    additional graphical parameters

Value

trls.influence returns a list with components:

r raw residuals as given by residuals.trls
hii diagonal elements of the Hat matrix
stresid standardised residuals
Di Cook's statistic
References


See Also

`surf.ls`, `influence.measures`, `plot.lm`

Examples

```r
library(MASS) # for eqscplot
data(topo, package = "MASS")
topo2 <- surf.ls(2, topo)
infl.topo2 <- trls.influence(topo2)
(cand <- as.data.frame(infl.topo2)[abs(infl.topo2$stresid) > 1.5, ])
cand.xy <- topo[as.integer(rownames(cand)), c("x", "y")]
trsurf <- trmat(topo2, 0, 6.5, 0, 6.5, 50)
eqscplot(trsurf, type = "n")
contour(trsurf, add = TRUE, col = "grey")
plot(topo2, add = TRUE, div = 3)
points(cand.xy, pch = 16, col = "orange")
text(cand.xy, labels = rownames(cand.xy), pos = 4, offset = 0.5)
```

trmat

*Evaluate Trend Surface over a Grid*

Description

Evaluate trend surface over a grid.

Usage

`trmat(obj, x1, xu, yl, yu, n)`

Arguments

- `obj` object returned by `surf.ls` or `surf.gls`
- `x1` limits of the rectangle for grid
- `xu` `yl` `yu`
- `n` use `n x n` grid within the rectangle

Value

list with components `x`, `y` and `z` suitable for contour and image.
variogram

References


See Also

surf.ls, surf.gls

Examples

data(topo, package="MASS")
topo.kr <- surf.ls(2, topo)
trsurf <- trmat(topo.kr, 0, 6.5, 0, 6.5, 50)

---

**variogram**

*Compute Spatial Variogram*

**Description**

Compute spatial (semi-)variogram of spatial data or residuals.

**Usage**

variogram(krig, nint, plotit = TRUE, ...)

**Arguments**

- **krig**: trend-surface or kriging object with columns x, y, and z
- **nint**: number of bins used
- **plotit**: logical for plotting
- **...**: parameters for the plot

**Details**

Divides range of data into nint bins, and computes the average squared difference for pairs with separation in each bin. Returns results for bins with 6 or more pairs.

**Value**

x and y coordinates of the variogram and cnt, the number of pairs averaged per bin.

**Side Effects**

Plots the variogram if plotit = TRUE
References


See Also

correlogram

Examples

```r
data(topo, package="MASS")
topo.kr <- surf.ls(2, topo)
variogram(topo.kr, 25)
```
Index

*Topic spatial
  anova.trls, 2
  correlogram, 3
  expcov, 4
  Kaver, 5
  Kenvl, 6
  Kfn, 7
  ppgetregion, 8
  ppinit, 8
  pplik, 9
  ppregion, 10
  predict.trls, 11
  prmat, 12
  Psim, 13
  SSI, 15
  Strauss, 16
  surf.gls, 17
  surf.ls, 18
  trls.influence, 19
  trmat, 20
  variogram, 21
  prmat, 12
  plmat, 14
  prmat, 17
  trmat, 19
  variogram, 4

anova.trls, 2
anovalist.trls (anova.trls), 2
correlogram, 3, 22
expcov, 4, 17
gaucov, 17
gaucov (expcov), 4
influence.measures, 20
Kaver, 5, 6, 7
Kenvl, 5, 6, 7
Kfn, 5, 6, 7
plot.lm, 20
plot.trls (trls.influence), 19
ppgetregion, 8, 10
ppinit, 7, 8, 10
pplik, 9
ppregion, 7–9, 10
predict.trls, 11
prmat, 12, 14, 17
Psim, 13, 15, 16
semat, 12, 14, 17
sphercov, 17
sphercov (expcov), 4
SSI, 13, 15, 16
Strauss, 10, 13, 15, 16
surf.gls, 4, 12, 14, 17, 19, 21
surf.ls, 2, 11, 17, 18, 20, 21
trls.influence, 19
trmat, 11, 12, 14, 17, 19, 20
variogram, 4, 21