Package ‘ramps’

August 24, 2022

Title Bayesian Geostatistical Modeling with RAMPS

Version 0.6.17

Date 2022-08-24

Maintainer Brian J Smith <brian-j-smith@uiowa.edu>

Depends coda, maps, Matrix, nlme

Imports fields, graphics, grDevices, methods, stats

Description Bayesian geostatistical modeling of Gaussian processes using a reparameterized and marginalized posterior sampling (RAMPS) algorithm designed to lower autocorrelation in MCMC samples. Package performance is tuned for large spatial datasets.

License GPL-2

URL https://www.jstatsoft.org/v25/i10

NeedsCompilation no

Author Brian J Smith [aut, cre],
Jun Yan [aut],
Mary Kathryn Cowles [aut]

Repository CRAN

Date/Publication 2022-08-24 18:00:02 UTC

R topics documented:

`corRCauchy` *Cauchy Spatial Correlation Structure*

Description

This function is a constructor for the 'corRCauchy' class, representing a Cauchy (rational quadratic) spatial correlation structure. Letting r denote the range, the correlation between two observations a distance d apart is $1/(1 + (d/r)^2)$.

Usage

```
corRCauchy(value = numeric(0), form = ~ 1,
           metric = c("euclidean", "maximum", "manhattan", "haversine"),
           radius = 3956)
```

Arguments

`value`	optional numeric "range" parameter value for the rational quadratic correlation structure, which must be greater than zero. Defaults to `numeric(0)`, which results in a range of 90% of the minimum distance being assigned to the parameter when `object` is initialized.
`form`	one-sided formula of the form `~ S1+...+Sp`, specifying spatial covariates `S1` through `Sp`. Defaults to `~ 1`, which corresponds to using the order of the observations in the data as a covariate.
`metric`	optional character string specifying the distance metric to be used. The currently available options are `"euclidean"` for the root sum-of-squares of distances; `"maximum"` for the maximum difference; `"manhattan"` for the sum of the absolute differences; and `"haversine"` for the great-circle distance (miles)

	between longitude/latitude coordinates. Partial matching of arguments is used, so only the first three characters need to be provided. Defaults to `"euclidean"`.
`radius`	radius to be used in the haversine formula for great-circle distance. Defaults to the Earth's radius of 3,956 miles.

Value

Object of class `'corRCauchy'`, also inheriting from class `'corRSpatial'`, representing a rational quadratic spatial correlation structure.

Note

When `"haversine"` is used as the distance metric, longitude and latitude coordinates must be given as the first and second covariates, respectively, in the formula specification for the `form` argument.

Author(s)

Brian Smith <brian-j-smith@uiowa.edu> and Jose Pinheiro <Jose.Pinheiro@pharma.novartis.com>, and Douglas Bates <bates@stat.wisc.edu>

References

Cressie, N.A.C. (1993), "Statistics for Spatial Data", J. Wiley & Sons.

Venables, W.N. and Ripley, B.D. (1997) "Modern Applied Statistics with S-plus", 2nd Edition, Springer-Verlag.

See Also

corRClasses

Examples

```
sp1 <- corRCauchy(form = ~ x + y + z)

spatDat <- data.frame(x = (0:4)/4, y = (0:4)/4)

cs1Cauchy <- corRCauchy(1, form = ~ x + y)
cs1Cauchy <- Initialize(cs1Cauchy, spatDat)
corMatrix(cs1Cauchy)

cs2Cauchy <- corRCauchy(1, form = ~ x + y, metric = "man")
cs2Cauchy <- Initialize(cs2Cauchy, spatDat)
corMatrix(cs2Cauchy)
```

corRClasses — *Spatial Correlation Structure Classes*

Description

Standard classes of spatial correlation structures available for the `georamps` function.

Spatial Structures:

`corRCauchy` Cauchy correlation.

`corRExp` exponential correlation.

`corRExpwr` powered exponential correlation.

`corRGaus` Gaussian correlation.

`corRGneit` Gneiting approximation to Gaussian correlation.

`corRLin` linear correlation.

`corRMatern` Matern correlation.

`corRSpher` spherical correlation.

`corRWave` sine wave correlation.

Spatio-Temporal Structures:

`corRExp2` exponential correlation.

`corRExpwr2` powered exponential correlation.

Temporally Integrated Spatial Structure:

`corRExpwr2Dt` powered exponential correlation.

Note

Users may define their own `corRStruct` classes by specifying a `constructor` function and, at a minimum, methods for the functions `corMatrix` and `coef`.

Author(s)

Brian Smith <brian-j-smith@uiowa.edu> and Jose Pinheiro <Jose.Pinheiro@pharma.novartis.com>, and Douglas Bates <bates@stat.wisc.edu>

See Also

`corRCauchy`, `corRExp`, `corRExp2`, `corRExpwr`, `corRExpwr2`, `corRExpwr2Dt`, `corRGaus`, `corRGneit`, `corRLin`, `corRMatern`, `corRSpher` `corRWave` corRExp — *Exponential Spatial Correlation Structure*

Description

This function is a constructor for the 'corRExp' class, representing an exponential spatial correlation structure. Letting r denote the range, the correlation between two observations a distance d apart is $\exp(-d/r)$.

Usage

```
corRExp(value = numeric(0), form = ~ 1,
        metric = c("euclidean", "maximum", "manhattan", "haversine"),
        radius = 3956)
```

Arguments

`value`	optional numeric “range” parameter value for the exponential correlation structure, which must be greater than zero. Defaults to `numeric(0)`, which results in a range of 90% of the minimum distance being assigned to the parameter when `object` is initialized.
`form`	one-sided formula of the form `~ S1+...+Sp`, specifying spatial covariates `S1` through `Sp`. Defaults to `~ 1`, which corresponds to using the order of the observations in the data as a covariate.
`metric`	optional character string specifying the distance metric to be used. The currently available options are `"euclidean"` for the root sum-of-squares of distances; `"maximum"` for the maximum difference; `"manhattan"` for the sum of the absolute differences; and `"haversine"` for the great-circle distance between longitude/latitude coordinates. Partial matching of arguments is used, so only the first three characters need to be provided. Defaults to `"euclidean"`.
`radius`	radius to be used in the haversine formula for great-circle distance. Defaults to the Earth’s radius of 3,956 miles.

Value

Object of class 'corRExp', also inheriting from class 'corRSpatial', representing an exponential spatial correlation structure.

Note

When `"haversine"` is used as the distance metric, longitude and latitude coordinates must be given as the first and second covariates, respectively, in the formula specification for the `form` argument.

Author(s)

Brian Smith <brian-j-smith@uiowa.edu> and Jose Pinheiro <Jose.Pinheiro@pharma.novartis.com>, and Douglas Bates <bates@stat.wisc.edu>

References

Cressie, N.A.C. (1993), "Statistics for Spatial Data", J. Wiley & Sons.

Venables, W.N. and Ripley, B.D. (1997) "Modern Applied Statistics with S-plus", 2nd Edition, Springer-Verlag.

See Also

corRClasses

Examples

```
sp1 <- corRExp(form = ~ x + y + z)

spatDat <- data.frame(x = (0:4)/4, y = (0:4)/4)

cs1Exp <- corRExp(1, form = ~ x + y)
cs1Exp <- Initialize(cs1Exp, spatDat)
corMatrix(cs1Exp)

cs2Exp <- corRExp(1, form = ~ x + y, metric = "man")
cs2Exp <- Initialize(cs2Exp, spatDat)
corMatrix(cs2Exp)
```

---

corRExp2 — *Non-Separable Exponential Spatio-Temporal Correlation Structure*

Description

This function is a constructor for the `'corRExp2'` class, representing a non-separable spatial correlation structure. Letting rs denote the spatial range, rt the temporal range, and $lambda$ the space-time interaction, the correlation between two observations a distance d apart in space and t in time is $\exp(-d/rs - t/rt - \lambda(d/rs)(t/rt))$.

Usage

```
corRExp2(value = numeric(0), form = ~ 1,
         metric = c("euclidean", "maximum", "manhattan", "haversine"),
         radius = 3956)
```

Arguments

`value`	optional numeric vector of three parameter values for the exponential correlation structure, corresponding to the "spatial range", "temporal range", and "space-time interaction". The range parameter values must be greater than zero, and the interaction greater than or equal to zero. Defaults to `numeric(0)`, which results in ranges of 90% of the minimum distances and an interaction of 0 being assigned to the parameters when `object` is initialized.

form	one-sided formula of the form ~ S1+...+Sp+T, specifying spatial covariates S1 through Sp and the times T at which measurement were taken.
metric	optional character string specifying the distance metric to be used. The currently available options are "euclidean" for the root sum-of-squares of distances; "maximum" for the maximum difference; "manhattan" for the sum of the absolute differences; and "haversine" for the great-circle distance (miles) between longitude/latitude coordinates. Partial matching of arguments is used, so only the first three characters need to be provided. Defaults to "euclidean".
radius	radius to be used in the haversine formula for great-circle distance. Defaults to the Earth's radius of 3,956 miles.

Value

Object of class 'corRExp2', inheriting from class 'corRSpatioTemporal', representing a non-separable spatial correlation structure.

Note

When "haversine" is used as the distance metric, longitude and latitude coordinates must be given as the first and second covariates, respectively, in the formula specification for the form argument.

Author(s)

Brian Smith <brian-j-smith@uiowa.edu>

References

Cressie, N. and Huang, H.-C. (1993) "Classes of Nonseperable, Spatio-Temporal Stationary Covariance Functions", *Journal of the American Statistical Association*, 94, 1330-1340.

See Also

corRClasses

Examples

```
sp1 <- corRExp2(form = ~ x + y + t)

spatDat <- data.frame(x = (0:4)/4, y = (0:4)/4, t=(0:4)/4)

cs1Exp <- corRExp2(c(1, 1, 1), form = ~ x + y + t)
cs1Exp <- Initialize(cs1Exp, spatDat)
corMatrix(cs1Exp)

cs2Exp <- corRExp2(c(1, 1, 1), form = ~ x + y + t, metric = "man")
cs2Exp <- Initialize(cs2Exp, spatDat)
corMatrix(cs2Exp)
```

`corRExpwr` *Powered Exponential Spatial Correlation Structure*

Description

This function is a constructor for the `'corRExpwr'` class, representing a powered exponential spatial correlation structure. Letting r denote the range and p the shape, the correlation between two observations a distance d apart is $\exp(-(d/r)^p)$.

Usage

```
corRExpwr(value = numeric(0), form = ~ 1,
          metric = c("euclidean", "maximum", "manhattan", "haversine"),
          radius = 3956)
```

Arguments

`value` optional numeric vector of two parameter values for the powered exponential correlation structure, corresponding to the "range" and "shape". The range parameter value must be greater than zero, and the shape in the interval (0, 2]. Defaults to `numeric(0)`, which results in a range of 90% of the minimum distance and a shape of 1 being assigned to the parameter when `object` is initialized.

`form` one-sided formula of the form `~ S1+...+Sp`, specifying spatial covariates `S1` through `Sp`. Defaults to `~ 1`, which corresponds to using the order of the observations in the data as a covariate.

`metric` optional character string specifying the distance metric to be used. The currently available options are `"euclidean"` for the root sum-of-squares of distances; `"maximum"` for the maximum difference; `"manhattan"` for the sum of the absolute differences; and `"haversine"` for the great-circle distance (miles) between longitude/latitude coordinates. Partial matching of arguments is used, so only the first three characters need to be provided. Defaults to `"euclidean"`.

`radius` radius to be used in the haversine formula for great-circle distance. Defaults to the Earth's radius of 3,956 miles.

Value

Object of class `'corRExpwr'`, also inheriting from class `'corRSpatial'`, representing a powered exponential spatial correlation structure.

Note

When `"haversine"` is used as the distance metric, longitude and latitude coordinates must be given as the first and second covariates, respectively, in the formula specification for the `form` argument.

Author(s)

Brian Smith `<brian-j-smith@uiowa.edu>`

References

Cressie, N.A.C. (1993), "Statistics for Spatial Data", J. Wiley & Sons.

Venables, W.N. and Ripley, B.D. (1997) "Modern Applied Statistics with S-plus", 2nd Edition, Springer-Verlag.

See Also

corRClasses

Examples

```
sp1 <- corRExpwr(form = ~ x + y + z)

spatDat <- data.frame(x = (0:4)/4, y = (0:4)/4)

cs1Expwr <- corRExpwr(c(1, 1), form = ~ x + y)
cs1Expwr <- Initialize(cs1Expwr, spatDat)
corMatrix(cs1Expwr)

cs2Expwr <- corRExpwr(c(1, 1), form = ~ x + y, metric = "man")
cs2Expwr <- Initialize(cs2Expwr, spatDat)
corMatrix(cs2Expwr)
```

---

corRExpwr2 — *Non-Separable Powered Exponential Spatio-Temporal Correlation Structure*

Description

This function is a constructor for the `'corRExpwr2'` class, representing a non-separable spatial correlation structure. Letting rs denote the spatial range, ps the spatial shape, rt the temporal range, pt the temporal shape, and $lambda$ the space-time interaction, the correlation between two observations a distance d apart in space and t in time is $\exp(-(d/rs)^ps - (t/rt)^pt - \lambda(d/rs)^ps(t/rt)^pt)$.

Usage

```
corRExpwr2(value = numeric(0), form = ~ 1,
           metric = c("euclidean", "maximum", "manhattan", "haversine"),
           radius = 3956)
```

Arguments

`value` optional numeric vector of five parameter values for the powered exponential correlation structure, corresponding to the "spatial range", "spatial shape", "temporal range", "temporal shape", and "space-time interaction". The range parameter values must be greater than zero, the shapes in the interval (0, 2], and the interaction greater than or equal to zero. Defaults to `numeric(0)`, which results in ranges of 90% of the minimum distances, shapes of 1, and an interaction of 0 being assigned to the parameters when `object` is initialized.

form	one-sided formula of the form ~ S1+...+Sp+T, specifying spatial covariates S1 through Sp and the times T at which measurement were taken.
metric	optional character string specifying the distance metric to be used. The currently available options are "euclidean" for the root sum-of-squares of distances; "maximum" for the maximum difference; "manhattan" for the sum of the absolute differences; and "haversine" for the great-circle distance (miles) between longitude/latitude coordinates. Partial matching of arguments is used, so only the first three characters need to be provided. Defaults to "euclidean".
radius	radius to be used in the haversine formula for great-circle distance. Defaults to the Earth's radius of 3,956 miles.

Value

Object of class 'corRExpwr2', inheriting from class 'corRSpatioTemporal', representing a non-separable spatial correlation structure.

Note

When "haversine" is used as the distance metric, longitude and latitude coordinates must be given as the first and second covariates, respectively, in the formula specification for the form argument.

Author(s)

Brian Smith <brian-j-smith@uiowa.edu>

References

Cressie, N. and Huang, H.-C. (1993) "Classes of Nonseperable, Spatio-Temporal Stationary Covariance Functions", *Journal of the American Statistical Association*, 94, 1330-1340.

Gneiting, T. (2002) "Nonseparable, stationary covariance functions for space-time data", *Journal of the American Statistical Association*, 97, 590-600.

See Also

corRClasses

Examples

```
sp1 <- corRExpwr2(form = ~ x + y + t)

spatDat <- data.frame(x = (0:4)/4, y = (0:4)/4, t=(0:4)/4)

cs1Expwr <- corRExpwr2(c(1, 1, 1, 1, 1), form = ~ x + y + t)
cs1Expwr <- Initialize(cs1Expwr, spatDat)
corMatrix(cs1Expwr)

cs2Expwr <- corRExpwr2(c(1, 1, 1, 1, 1), form = ~ x + y + t, metric = "man")
cs2Expwr <- Initialize(cs2Expwr, spatDat)
corMatrix(cs2Expwr)
```

corRExpwr2Dt	*Non-Separable Temporally Integrated Powered Exponential Spatial Correlation Structure*

Description

This function is a constructor for the 'corRExpwr2Dt' class, representing a non-separable spatial correlation structure for temporally integrated measurements. Letting rs denote the spatial range, ps the spatial shape, rt the temporal range, and $lambda$ the space-time interaction, the correlation between two observations a distance d apart in space and t in time is $\exp(-(d/rs)^p s - t/rt - \lambda(d/rs)^p s(t/rt))$.

Usage

```
corRExpwr2Dt(value = numeric(0), form = ~ 1,
             metric = c("euclidean", "maximum", "manhattan", "haversine"),
             radius = 3956)
```

Arguments

`value` optional numeric vector of four parameter values for the powered exponential correlation structure, corresponding to the "spatial range", "spatial shape", "temporal range", and "space-time interaction". The range parameter values must be greater than zero, the shape in the interval (0, 2], and the interaction greater than or equal to zero. Defaults to `numeric(0)`, which results in ranges of 90% of the minimum distances, a shape of 1, and an interaction of 0 being assigned to the parameters when `object` is initialized.

`form` one-sided formula of the form ~ `S1+...+Sp+T1+T2`, specifying spatial covariates `S1` through `Sp` and the times `(T1, T2)` at which measurement periods begin and end, respectively.

`metric` optional character string specifying the distance metric to be used. The currently available options are `"euclidean"` for the root sum-of-squares of distances; `"maximum"` for the maximum difference; `"manhattan"` for the sum of the absolute differences; and `"haversine"` for the great-circle distance (miles) between longitude/latitude coordinates. Partial matching of arguments is used, so only the first three characters need to be provided. Defaults to `"euclidean"`.

`radius` radius to be used in the haversine formula for great-circle distance. Defaults to the Earth's radius of 3,956 miles.

Value

Object of class 'corRExpwr2Dt', also inheriting from class 'corRSpatial', representing a non-separable spatial correlation structure.

Note

When `"haversine"` is used as the distance metric, longitude and latitude coordinates must be given as the first and second covariates, respectively, in the formula specification for the `form` argument.

Author(s)

Brian Smith <brian-j-smith@uiowa.edu>

References

Cressie, N. and Huang, H.-C. (1993) “Classes of Nonseperable, Spatio-Temporal Stationary Covariance Functions”, *Journal of the American Statistical Association*, 94, 1330-1340.

Smith, B.J. and Oleson, J.J. (2007) “Geostatistical Hierarchical Model for Temporally Integrated Radon Measurements”, *Jounal of Agricultural, Biological, and Environmental Statistics*, in press.

See Also

corRClasses

Examples

```
sp1 <- corRExpwr2Dt(form = ~ x + y + t1 + t2)

spatDat <- data.frame(x = (0:4)/4, y = (0:4)/4, t1=(0:4)/4, t2=(1:5)/4)

cs1ExpwrDt <- corRExpwr2Dt(c(1, 1, 1, 1), form = ~ x + y + t1 + t2)
cs1ExpwrDt <- Initialize(cs1ExpwrDt, spatDat)
corMatrix(cs1ExpwrDt)

cs2ExpwrDt <- corRExpwr2Dt(c(1, 1, 1, 1), form = ~ x + y + t1 + t2, metric = "man")
cs2ExpwrDt <- Initialize(cs2ExpwrDt, spatDat)
corMatrix(cs2ExpwrDt)
```

---

corRGaus *Gaussian Spatial Correlation Structure*

Description

This function is a constructor for the 'corRGaus' class, representing a Gaussian spatial correlation structure. Letting r denote the range, the correlation between two observations a distance d apart is $\exp(-(d/r)^2)$.

Usage

```
corRGaus(value = numeric(0), form = ~ 1,
         metric = c("euclidean", "maximum", "manhattan", "haversine"),
         radius = 3956)
```

Arguments

- `value` — optional numeric “range” parameter value for the Gaussian correlation structure, which must be greater than zero. Defaults to `numeric(0)`, which results in a range of 90% of the minimum distance being assigned to the parameter when `object` is initialized.
- `form` — one-sided formula of the form `~ S1+...+Sp`, specifying spatial covariates `S1` through `Sp`. Defaults to `~ 1`, which corresponds to using the order of the observations in the data as a covariate.
- `metric` — optional character string specifying the distance metric to be used. The currently available options are `"euclidean"` for the root sum-of-squares of distances; `"maximum"` for the maximum difference; `"manhattan"` for the sum of the absolute differences; and `"haversine"` for the great-circle distance (miles) between longitude/latitude coordinates. Partial matching of arguments is used, so only the first three characters need to be provided. Defaults to `"euclidean"`.
- `radius` — radius to be used in the haversine formula for great-circle distance. Defaults to the Earth’s radius of 3,956 miles.

Value

Object of class `'corRGaus'`, also inheriting from class `'corRSpatial'`, representing a Gaussian spatial correlation structure.

Note

When `"haversine"` is used as the distance metric, longitude and latitude coordinates must be given as the first and second covariates, respectively, in the formula specification for the `form` argument.

Author(s)

Brian Smith <brian-j-smith@uiowa.edu> and Jose Pinheiro <Jose.Pinheiro@pharma.novartis.com>, and Douglas Bates <bates@stat.wisc.edu>

References

Cressie, N.A.C. (1993), “Statistics for Spatial Data”, J. Wiley & Sons.

Venables, W.N. and Ripley, B.D. (1997) “Modern Applied Statistics with S-plus”, 2nd Edition, Springer-Verlag.

See Also

corRClasses

Examples

```
sp1 <- corRGaus(form = ~ x + y + z)

spatDat <- data.frame(x = (0:4)/4, y = (0:4)/4)
```

```
cs1Gaus <- corRGaus(1, form = ~ x + y)
cs1Gaus <- Initialize(cs1Gaus, spatDat)
corMatrix(cs1Gaus)

cs2Gaus <- corRGaus(1, form = ~ x + y, metric = "man")
cs2Gaus <- Initialize(cs2Gaus, spatDat)
corMatrix(cs2Gaus)
```

---

corRGneit *Gneiting Spatial Correlation Structure*

Description

This function is a constructor for the 'corRGneit' class, representing the Gneiting approximation to the Gaussian correlation structure. Letting r denote the range, the correlation between two observations a distance $d < r/s$ apart is $(1 + 8sx + 25(sx)^2 + 32(sx)^3)(1 - sx)^8$, where s = 0.301187465825. If $d \geq r/s$ the correlation is zero.

Usage

```
corRGneit(value = numeric(0), form = ~ 1,
          metric = c("euclidean", "maximum", "manhattan", "haversine"),
          radius = 3956)
```

Arguments

value	optional numeric "range" parameter value for the Gneiting correlation structure, which must be greater than zero. Defaults to `numeric(0)`, which results in a range of 90% of the minimum distance being assigned to the parameter when `object` is initialized.
form	one-sided formula of the form ~ `S1+...+Sp`, specifying spatial covariates `S1` through `Sp`. Defaults to ~ `1`, which corresponds to using the order of the observations in the data as a covariate.
metric	optional character string specifying the distance metric to be used. The currently available options are `"euclidean"` for the root sum-of-squares of distances; `"maximum"` for the maximum difference; `"manhattan"` for the sum of the absolute differences; and `"haversine"` for the great-circle distance between longitude/latitude coordinates. Partial matching of arguments is used, so only the first three characters need to be provided. Defaults to `"euclidean"`.
radius	radius to be used in the haversine formula for great-circle distance. Defaults to the Earth's radius of 3,956 miles.

Value

Object of class 'corRGneit', also inheriting from class 'corRSpatial', representing the Gneiting spatial correlation structure.

Note

When "haversine" is used as the distance metric, longitude and latitude coordinates must be given as the first and second covariates, respectively, in the formula specification for the form argument.

Author(s)

Brian Smith <brian-j-smith@uiowa.edu>

References

Gneiting, T. (1999), "Correlation Functions for Atmospheric Data Analysis", *Quarterly Journal of the Royal Meteorological Society*, 125(559), 2449-2464.

Venables, W.N. and Ripley, B.D. (1997) "Modern Applied Statistics with S-plus", 2nd Edition, Springer-Verlag.

See Also

corRClasses

Examples

```
sp1 <- corRGneit(form = ~ x + y + z)

spatDat <- data.frame(x = (0:4)/4, y = (0:4)/4)

cs1Gneit <- corRGneit(1, form = ~ x + y)
cs1Gneit <- Initialize(cs1Gneit, spatDat)
corMatrix(cs1Gneit)

cs2Gneit <- corRGneit(1, form = ~ x + y, metric = "man")
cs2Gneit <- Initialize(cs2Gneit, spatDat)
corMatrix(cs2Gneit)
```

---

corRLin *Linear Spatial Correlation Structure*

Description

This function is a constructor for the 'corRLin' class, representing a linear spatial correlation structure. Letting r denote the range, the correlation between two observations a distance $d < r$ apart is $1 - (d/r)$. If $d \geq r$ the correlation is zero.

Usage

```
corRLin(value = numeric(0), form = ~ 1,
        metric = c("euclidean", "maximum", "manhattan", "haversine"),
        radius = 3956)
```

Arguments

value
: optional numeric "range" parameter value for the linear correlation structure, which must be greater than zero. Defaults to `numeric(0)`, which results in a range of 90% of the minimum distance being assigned to the parameter when `object` is initialized.

form
: one-sided formula of the form `~ S1+...+Sp`, specifying spatial covariates `S1` through `Sp`. Defaults to `~ 1`, which corresponds to using the order of the observations in the data as a covariate.

metric
: optional character string specifying the distance metric to be used. The currently available options are `"euclidean"` for the root sum-of-squares of distances; `"maximum"` for the maximum difference; `"manhattan"` for the sum of the absolute differences; and `"haversine"` for the great-circle distance (miles) between longitude/latitude coordinates. Partial matching of arguments is used, so only the first three characters need to be provided. Defaults to `"euclidean"`.

radius
: radius to be used in the haversine formula for great-circle distance. Defaults to the Earth's radius of 3,956 miles.

Value

Object of class `'corRLin'`, also inheriting from class `'corRSpatial'`, representing a linear spatial correlation structure.

Note

When `"haversine"` is used as the distance metric, longitude and latitude coordinates must be given as the first and second covariates, respectively, in the formula specification for the `form` argument.

Author(s)

Brian Smith <brian-j-smith@uiowa.edu> and Jose Pinheiro <Jose.Pinheiro@pharma.novartis.com>, and Douglas Bates <bates@stat.wisc.edu>

References

Cressie, N.A.C. (1993), "Statistics for Spatial Data", J. Wiley & Sons.

Venables, W.N. and Ripley, B.D. (1997) "Modern Applied Statistics with S-plus", 2nd Edition, Springer-Verlag.

See Also

corRClasses

Examples

```
sp1 <- corRLin(form = ~ x + y + z)

spatDat <- data.frame(x = (0:4)/4, y = (0:4)/4)
```

```
cs1Lin <- corRLin(1, form = ~ x + y)
cs1Lin <- Initialize(cs1Lin, spatDat)
corMatrix(cs1Lin)

cs2Lin <- corRLin(1, form = ~ x + y, metric = "man")
cs2Lin <- Initialize(cs2Lin, spatDat)
corMatrix(cs2Lin)
```

---

corRMatern *Matern Spatial Correlation Structure*

Description

This function is a constructor for the 'corRMatern' class, representing a Matern spatial correlation structure. Letting r denote the range, and s the scale, the correlation between two observations a distance d apart is $1/(2^{s-1}\Gamma(s))(d/r)^s K_s(d/r)$.

Usage

```
corRMatern(value = numeric(0), form = ~ 1,
           metric = c("euclidean", "maximum", "manhattan", "haversine"),
           radius = 3956)
```

Arguments

value	optional numeric vector of two parameter values for the Matern correlation structure, corresponding to the "range" and "scale". The range parameter value must be greater than zero, and the scale in the interval (0, 2]. Defaults to numeric(0), which results in a range of 90% of the minimum distance and a scale of 0.5 being assigned to the parameter when object is initialized.
form	one-sided formula of the form ~ S1+...+Sp, specifying spatial covariates S1 through Sp. Defaults to ~ 1, which corresponds to using the order of the observations in the data as a covariate.
metric	optional character string specifying the distance metric to be used. The currently available options are "euclidean" for the root sum-of-squares of distances; "maximum" for the maximum difference; "manhattan" for the sum of the absolute differences; and "haversine" for the great-circle distance (miles) between longitude/latitude coordinates. Partial matching of arguments is used, so only the first three characters need to be provided. Defaults to "euclidean".
radius	radius to be used in the haversine formula for great-circle distance. Defaults to the Earth's radius of 3,956 miles.

Value

Object of class 'corRMatern', also inheriting from class 'corRSpatial', representing a Matern spatial correlation structure.

Note

When "haversine" is used as the distance metric, longitude and latitude coordinates must be given as the first and second covariates, respectively, in the formula specification for the form argument.

Author(s)

Brian Smith <brian-j-smith@uiowa.edu>

References

Cressie, N.A.C. (1993), "Statistics for Spatial Data", J. Wiley & Sons.

Venables, W.N. and Ripley, B.D. (1997) "Modern Applied Statistics with S-plus", 2nd Edition, Springer-Verlag.

See Also

corRClasses

Examples

```
sp1 <- corRMatern(form = ~ x + y + z)

spatDat <- data.frame(x = (0:4)/4, y = (0:4)/4)

cs1Matern <- corRMatern(c(1, 1), form = ~ x + y)
cs1Matern <- Initialize(cs1Matern, spatDat)
corMatrix(cs1Matern)

cs2Matern <- corRMatern(c(1, 1), form = ~ x + y, metric = "man")
cs2Matern <- Initialize(cs2Matern, spatDat)
corMatrix(cs2Matern)
```

---

corRSpher *Spherical Spatial Correlation Structure*

Description

This function is a constructor for the 'corRSpher' class, representing a spherical spatial correlation structure. Letting r denote the range, the correlation between two observations a distance $d < r$ apart is $1 - 1.5(d/r) + 0.5(d/r)^3$. If $d \geq r$ the correlation is zero.

Usage

```
corRSpher(value = numeric(0), form = ~ 1,
          metric = c("euclidean", "maximum", "manhattan", "haversine"),
          radius = 3956)
```

Arguments

`value`	optional numeric "range" parameter value for the spherical correlation structure, which must be greater than zero. Defaults to `numeric(0)`, which results in a range of 90% of the minimum distance being assigned to the parameter when `object` is initialized.
`form`	one-sided formula of the form `~ S1+...+Sp`, specifying spatial covariates `S1` through `Sp`. Defaults to `~ 1`, which corresponds to using the order of the observations in the data as a covariate.
`metric`	optional character string specifying the distance metric to be used. The currently available options are `"euclidean"` for the root sum-of-squares of distances; `"maximum"` for the maximum difference; `"manhattan"` for the sum of the absolute differences; and `"haversine"` for the great-circle distance (miles) between longitude/latitude coordinates. Partial matching of arguments is used, so only the first three characters need to be provided. Defaults to `"euclidean"`.
`radius`	radius to be used in the haversine formula for great-circle distance. Defaults to the Earth's radius of 3,956 miles.

Value

An object of class `'corRSpher'`, also inheriting from class `'corRSpatial'`, representing a spherical spatial correlation structure.

Note

When `"haversine"` is used as the distance metric, longitude and latitude coordinates must be given as the first and second covariates, respectively, in the formula specification for the `form` argument.

Author(s)

Jose Pinheiro <Jose.Pinheiro@pharma.novartis.com>, Douglas Bates <bates@stat.wisc.edu>, and Brian Smith <brian-j-smith@uiowa.edu>

References

Cressie, N.A.C. (1993), "Statistics for Spatial Data", J. Wiley & Sons.

Venables, W.N. and Ripley, B.D. (1997) "Modern Applied Statistics with S-plus", 2nd Edition, Springer-Verlag.

See Also

`corRClasses`

Examples

```
sp1 <- corRSpher(form = ~ x + y + z)

spatDat <- data.frame(x = (0:4)/4, y = (0:4)/4)
```

```
cs1Spher <- corRSpher(1, form = ~ x + y)
cs1Spher <- Initialize(cs1Spher, spatDat)
corMatrix(cs1Spher)

cs2Spher <- corRSpher(1, form = ~ x + y, metric = "man")
cs2Spher <- Initialize(cs2Spher, spatDat)
corMatrix(cs2Spher)
```

---

corRWave *Sine Wave Spatial Correlation Structure*

Description

This function is a constructor for the 'corRWave' class, representing a sine wave spatial correlation structure. Letting r denote the range, the correlation between two observations a distance d apart is $\sin(d/r)/(d/r)$.

Usage

```
corRWave(value = numeric(0), form = ~ 1,
         metric = c("euclidean", "maximum", "manhattan", "haversine"),
         radius = 3956)
```

Arguments

value	optional numeric "range" parameter value for the sine wave correlation structure, which must be greater than zero. Defaults to numeric(0), which results in a range of 90% of the minimum distance being assigned to the parameter when object is initialized.
form	one-sided formula of the form ~ S1+...+Sp, specifying spatial covariates S1 through Sp. Defaults to ~ 1, which corresponds to using the order of the observations in the data as a covariate.
metric	optional character string specifying the distance metric to be used. The currently available options are "euclidean" for the root sum-of-squares of distances; "maximum" for the maximum difference; "manhattan" for the sum of the absolute differences; and "haversine" for the great-circle distance (miles) between longitude/latitude coordinates. Partial matching of arguments is used, so only the first three characters need to be provided. Defaults to "euclidean".
radius	radius to be used in the haversine formula for great-circle distance. Defaults to the Earth's radius of 3,956 miles.

Value

Object of class 'corRWave', also inheriting from class 'corRSpatial', representing a sine wave spatial correlation structure.

Note

When "haversine" is used as the distance metric, longitude and latitude coordinates must be given as the first and second covariates, respectively, in the formula specification for the form argument.

Author(s)

Brian Smith <brian-j-smith@uiowa.edu>

References

Cressie, N.A.C. (1993), "Statistics for Spatial Data", J. Wiley & Sons.

Venables, W.N. and Ripley, B.D. (1997) "Modern Applied Statistics with S-plus", 2nd Edition, Springer-Verlag.

See Also

corRClasses

Examples

```
sp1 <- corRWave(form = ~ x + y + z)

spatDat <- data.frame(x = (0:4)/4, y = (0:4)/4)

cs1Wave <- corRWave(1, form = ~ x + y)
cs1Wave <- Initialize(cs1Wave, spatDat)
corMatrix(cs1Wave)

cs2Wave <- corRWave(1, form = ~ x + y, metric = "man")
cs2Wave <- Initialize(cs2Wave, spatDat)
corMatrix(cs2Wave)
```

DIC *Deviance Information Criterion*

Description

Calculates the Deviance Information Criterion (DIC) for comparisons of georamps model fits.

Usage

```
    ## S3 method for class 'ramps'
DIC(object, ...)
```

Arguments

object	object returned by georamps.
...	some methods for this generic require additional arguments. None are used in this method.

Value

An numeric vector with the following two elements:

DIC value of the Deviance Information Criterion.

pD effective number of model parameters.

Author(s)

Brian Smith <brian-j-smith@uiowa.edu>

References

Spiegelhalter, D.J., Best, N.G., Carlin, B.P., and van der Linde, A. (2002) “Bayesian Measures of Model Complexity and Fit”, *Journal of the Royal Statistical Society - Series B*, 64, 583-639.

See Also

georamps

Examples

```
## DIC calculation for georamps example results

## Not run:
DIC(NURE.fit)

## End(Not run)
```

expand.chain *Expand MCMC Samples for georamps Model Fits*

Description

Generates additional posterior samples for georamps model fits by restarting the MCMC sampler at the last set of sampled parameter values.

Usage

```
expand.chain(object, n)
```

Arguments

object object returned by georamps.

n additional number of times to iterate the MCMC sampler.

Value

'ramps' object containing the previously and newly sampled parameter values.

Author(s)

Brian Smith <brian-j-smith@uiowa.edu>

See Also

georamps

Examples

```
## Generate 25 additional samples for the georamps example

## Not run:
fit <- expand.chain(NURE.fit, 25)

## End(Not run)
```

---

genUSStateGrid — *Generating a Grid over a US State*

Description

This function generate a grid of points over a US state with given increment size or resolution.

Usage

```
genUSStateGrid(state, incr = NULL, resolution = NULL)
```

Arguments

state	the name of a US state.
incr	a numeric vector of length 2 specifying the increment in longitude and latitude.
resolution	a numeric vector of length 2 specifying the size of the grid in longitude and latitude.

Value

A data.frame:

lon	longitude of the grid point.
lat	latitude of the grid point.
id	the id number of the county in which the grid point is located.
county	the name of the county in which the grid point is located.

Author(s)

Jun Yan <jun.yan@uconn.edu>

See Also

genUSStateSites

Examples

```
mygrid <- genUSStateGrid('iowa', resolution=c(8, 4))
map('state', 'iowa')
points(mygrid)
```

---

genUSStateSites *Generating Random Sites in a US State*

Description

A completely spatial random set of point is generated for a US state.

Usage

```
genUSStateSites(state, nsites)
```

Arguments

state	the name of a US state.
nsites	the number of sites needed.

Value

A matrix of longitude and latitude....

See Also

genUSStateGrid

---

georamps *Bayesian Geostatistical Model Fitting with RAMPS*

Description

General function for fitting Bayesian geostatistical models using the reparameterized and marginalized posterior sampling (RAMPS) algorithm of Yan et al. (2007).

Usage

```
georamps(fixed, random, correlation, data, subset, weights,
         variance = list(fixed = ~ 1, random = ~ 1, spatial = ~ 1),
         aggregate = list(grid = NULL, blockid = ""), kmat = NULL,
         control = ramps.control(...), contrasts = NULL, ...)
```

Arguments

fixed
: two-sided linear "formula" object describing the main effects in the mean structure of the model, with the response on the left of a ~ operator and the terms, separated by + operators, on the right.

random
: optional one-sided formula of the form ~ 1 | g, specifying random intercepts for groups defined by the factor g. Several grouping variables may be simultaneously specified, separated by the * operator, as in ~ 1 | g1 * g2 * g3. In such cases, the levels of each variable are pasted together and the resulting factor used to group the observations. Missing NA values may be given in the grouping variable to omit random effects for the associated measurements.

correlation
: 'corRSpatial' object describing the spatial correlation structure. See the corRClasses documentation for a listing of the available structures.

data
: optional data frame containing the variables named in fixed, random, correlation, weights, variance, and subset.

subset
: optional expression indicating the subset of rows in data that should be used in the fit. This can be a logical vector, or a numerical vector indicating which observation numbers are to be included, or a character vector of the row names to be included. All observations are included by default.

weights
: optional numerical vector of measurement error variance (inverse) weights to be used in the fitting process. Defaults to a value of 1 for point-source measurements and the number of grid points for areal measurements (see the aggregate argument below).

variance
: optional list of one-sided formulas, each of the form ~ g where g defines a grouping factor for the following elements: fixed for measurement error variances; random for random effects error variances; and spatial for spatial variances. A single variance is assumed in each case by default.

aggregate
: optional list of elements: grid a data frame of coordinates to use for Monte Carlo integration over geographic blocks at which areal measurements are available; and blockid a character string specifying the column by which to merge the areal measurements in data with the grid coordinates in grid. Merging is only performed for blockid values that are common to both datasets. All observations in data are treated as point-source measurements by default.

kmat
: optional $n \times s$ design matrix for mapping spatial sites to outcome responses, where n is the number of responses and s the number of unique sites. Unique sites are ordered first according to those supplied to the data argument and second to those supplied to the aggregate argument. Defaults to kmat[i,j] = 1 / N[i] if site j is one of N[i] measurement sites contributing to response i; otherwise kmat[i,j] = 0. Rows or columns of zeros are not supported.

control
: list of parameters for controlling the fitting process. See the ramps.control documentation for details.

contrasts
: optional list. See the contrasts.arg of model.matrix.

...
: further arguments passed to or from other methods.

Value

An object of class 'ramps' containing the following elements:

params	'mcmc' object of monitored model parameters with variable labels in the column names and MCMC iteration numbers in the row names.
z	'mcmc' object of monitored latent spatial parameters with variable labels in the column names and MCMC iteration numbers in the row names.
loglik	vector of data log-likelihood values at each MCMC iteration.
evals	vector of slice sampler evaluations at each MCMC iteration.
call	the matched function call to georamps.
y	response vector.
xmat	design matrix for the main effects.
terms	the 'terms' object for xmat.
xlevels	list of the factor levels for xmat.
etype	grouping factor for the measurement error variances.
weights	weights used in the fitting process.
kmat	matrix for mapping the spatial parameters to the observed data.
correlation	specified 'corRSpatial' object for the spatial correlation structure.
coords	matrix of unique coordinates for the measurement and grid sites.
ztype	grouping factor for the spatial variances.
wmat	matrix for mapping the random effects to the observed data.
retype	grouping factor for the random effects variances.
control	a list of control parameters used in the fitting process.

Author(s)

Brian Smith <brian-j-smith@uiowa.edu>, Jun Yan <jun.yan@uconn.edu>, and Kate Cowles <kate-cowles@uiowa.edu>

References

Yan, J., Cowles, M.K., Wang, S., and Armstrong, M. (2007) "Parallelizing MCMC for Bayesian Spatiotemporal Geostatistical Models", *Statistics and Computing*, 17(4), 323-335.

Smith, B. J., Yan, J., and Cowles, M. K. (2008) "Unified Geostatistical Modeling for Data Fusion and Spatial Heteroskedasticity with R Package ramps", *Journal of Statistical Software*, 25(10), 1-21.

See Also

corRClasses, ramps.control, mcmc, DIC.ramps, plot.ramps, predict.ramps, summary.ramps, window.ramps

Examples

```
## Not run:
## Load the included uranium datasets for use in this example
data(NURE)

## Geostatistical analysis of areal measurements
NURE.ctrl1 <- ramps.control(
   iter = 25,
   beta = param(0, "flat"),
   sigma2.e = param(1, "invgamma", shape = 2.0, scale = 0.1, tuning = 0.75),
   phi = param(10, "uniform", min = 0, max = 35, tuning = 0.50),
   sigma2.z = param(1, "invgamma", shape = 2.0, scale = 0.1)
)

NURE.fit1 <- georamps(log(ppm) ~ 1,
   correlation = corRExp(form = ~ lon + lat, metric = "haversine"),
   weights = area,
   data = NURE,
   subset = (measurement == 1),
   aggregate = list(grid = NURE.grid, blockid = "id"),
   control = NURE.ctrl1
)
print(NURE.fit1)
summary(NURE.fit1)


## Analysis of point-source measurements
NURE.ctrl2 <- ramps.control(
   iter = 25,
   beta = param(0, "flat"),
   sigma2.e = param(1, "invgamma", shape = 2.0, scale = 0.1, tuning = 0.75),
   phi = param(10, "uniform", min = 0, max = 35, tuning = 0.5),
   sigma2.z = param(1, "invgamma", shape = 2.0, scale = 0.1)
)

NURE.fit2 <- georamps(log(ppm) ~ 1,
   correlation = corRExp(form = ~ lon + lat, metric = "haversine"),
   data = NURE,
   subset = (measurement == 2),
   control = NURE.ctrl2
)
print(NURE.fit2)
summary(NURE.fit2)


## Joint analysis of areal and point-source measurements with
## prediction only at grid sites
NURE.ctrl <- ramps.control(
   iter = 25,
   beta = param(rep(0, 2), "flat"),
   sigma2.e = param(rep(1, 2), "invgamma", shape = 2.0, scale = 0.1, tuning = 0.75),
   phi = param(10, "uniform", min = 0, max = 35, tuning = 0.5),
```

```
    sigma2.z = param(1, "invgamma", shape = 2.0, scale = 0.1),
    z.monitor = NURE.grid
)

NURE.fit <- georamps(log(ppm) ~ factor(measurement) - 1,
    correlation = corRExp(form = ~ lon + lat, metric = "haversine"),
    variance = list(fixed = ~ measurement),
    weights = area * (measurement == 1) + (measurement == 2),
    data = NURE,
    aggregate = list(grid = NURE.grid, blockid = "id"),
    control = NURE.ctrl
)
print(NURE.fit)
summary(NURE.fit)


## Discard initial 5 MCMC samples as a burn-in sequence
fit <- window(NURE.fit, iter = 6:25)
print(fit)
summary(fit)

## Deviance Information Criterion
DIC(fit)

## Prediction at unmeasured sites
ct <- map("state", "connecticut", plot = FALSE)
lon <- seq(min(ct$x, na.rm = TRUE), max(ct$x, na.rm = TRUE), length = 20)
lat <- seq(min(ct$y, na.rm = TRUE), max(ct$y, na.rm = TRUE), length = 15)
grid <- expand.grid(lon, lat)

newsites <- data.frame(lon = grid[,1], lat = grid[,2],
                       measurement = 1)
pred <- predict(fit, newsites)

plot(pred, func = function(x) exp(mean(x)),
     database = "state", regions = "connecticut",
     resolution = c(200, 150), bw = 5,
     main = "Posterior Mean",
     legend.args = list(text = "ppm", side = 3, line = 1))

plot(pred, func = function(x) exp(sd(x)),
     database = "state", regions = "connecticut",
     resolution = c(200, 150), bw = 5,
     main = "Posterior Standard Deviation",
     legend.args = list(text = "ppm", side = 3, line = 1))

## End(Not run)
```

NURE *Dataset of USGS NURE Uranium Measurements*

Description

Connecticut, USA, areal and point-source uranium measurements from the United States Geological Survey (USGS) National Uranium Resource Evaluation (NURE) project.

Usage

```
data(NURE)
```

Format

The following variables are provided in the `NURE` data frame:

`ppm` uranium measurements in parts per million.

`measurement` type of measurement: 1 = areal, 2 = point-source.

`lon` longitude coordinates of point-source measurements.

`lat` latitude coordinates of point-source measurements.

`easting` Universal Transverse Mercator easting coordinates - projected distances from the central meridian.

`northing` Universal Transverse Mercator northing coordinates - projected distances from the equator.

`county` counties from which measurements were taken.

`area` county land mass areas in square miles.

`id` unique identifiers for measured counties or sites.

A grid of coordinates is provided by the `NURE.grid` data frame to facilitate Monte Carlo integration in geostatistical modeling of areal measurements. The included columns are

`lon` longitude coordinates of grid sites.

`lat` latitude coordinates of grid sites.

`id` county identifiers.

Areal measurements in `NURE` can be matched to the grid coordinates in `NURE.grid` via the shared `"id"` variable.

References

Duval, J.S., Jones, W.J., Riggle, F.R., and Pitkin, J.A. (1989) “Equivalent uranium map of conterminous United States”, USGS Open-File Report 89-478.

Smith, S.M.(2006) “National Geochemical Database Reformatted Data from the National Uranium Resource Evaluation (NURE) Hydrogeochemical and Stream Sediment Reconnaissance (HSSR) Program”, USGS Open-File Report 97-492.

Examples

```
data(NURE)

## Map areal and point-source measurements
ppm1 <- NURE$ppm[NURE$measurement == 1]
level <- (max(ppm1) - ppm1) / diff(range(ppm1))
map("county", "connecticut", fill = TRUE, col = gray(level))
title("Connecticut Uranium Measurements")
points(NURE$lon, NURE$lat)

## Map grid sites
map("county", "connecticut")
title("Regular Grid of Coordinates")
points(NURE.grid$lon, NURE.grid$lat)
```

`param` *Initialization of georamps Model Parameters*

Description

Function used in conjunction with `ramps.control` to specify the initial values and prior distributions used in calls to `georamps`.

Usage

```
param(init, prior = c("flat", "invgamma", "normal", "uniform", "user"), tuning,
    ...)
```

Arguments

`init`	numerical vector of initial parameter values. `NA` elements will be replaced with random draws from the prior distribution when possible.
`prior`	character string specifying the prior distribution. This must be one of `"flat"`, `"invgamma"`, `"normal"`, `"uniform"`, or `"user"`, with default `"flat"`, and may be abbreviated to a unique prefix.
`tuning`	numerical tuning values the slice-simplex routine in the MCMC sampler.
`...`	hyperparameters of the specified prior distribution. See details below.

Details

The supported prior distributions and associated hyperparameters are:

`"flat"` Flat prior with no hyperparameters.

`"invgamma"` Inverse-gamma with hyperparameters `shape > 0` and `scale > 0` such that $f(x) = scale^{shape}/\Gamma(shape)x^{-shape-1}\exp(-scale/x)$.

"normal" Normal with hyperparameters mean and variance such that $f(x) = (2\pi)^{-n/2}|\Sigma|^{-1/2}\exp(-1/2(x-\mu)'\Sigma^{-1}(x-\mu))$. The variance hyperparameter must be positive definite and may be supplied either as a vector (independence) or a matrix.

"uniform" Uniform with hyperparameters min and max > min such that $f(x) = 1/(max - min)$.

"user" Use-defined function supplied as hyperparameter f which takes a single numeric vector of length and order equal to the associated model parameters and whose returns values are proportional to the prior distribution.

The number of model parameters to be initialized is determined by length(init). Missing values occurring in the supplied init vector will be replaced with draws from the prior distribution, for all but the "flat" specification.

Value

A list of class 'param' containing the following components:

init	numerical vector of initial parameter values.
prior	character string specifying the prior distribution.
tuning	numerical vector of tuning values of length(init).
...	hyperparameters of the specified prior distribution.

Author(s)

Brian Smith <brian-j-smith@uiowa.edu>

See Also

georamps, ramps.control

Examples

```
## Initial values for a flat prior
param(rep(0, 2), "flat")

## Random generation of initial values for an inverse-gamma prior
param(rep(NA, 2), "invgamma", shape = 2.0, scale = 0.1)

## Independent normal priors
param(rep(0, 2), "normal", mean = c(0, 0), variance = c(100, 100))

## Correlated normal priors
npv <- rbind(c(100, 25), c(25, 100))
param(rep(0, 2), "normal", mean = c(0, 0), variance = npv)

## Uniform prior and MCMC tuning parameter specification
param(10, "uniform", min = 0, max = 100, tuning = 0.5)
```

plot *Posterior Spatial Distribution Plots*

Description

Creates surface maps of posterior spatial distributions from `georamps` or `predict.ramps`.

Usage

```
   ## S3 method for class 'ramps'
plot(x, type = c("i", "c", "w"), col = tim.colors(64), func = mean,
        sites = FALSE, database = NULL, regions = ".", resolution = c(64, 64),
        bw = 1, ...)

   ## S3 method for class 'predict.ramps'
plot(x, type = c("i", "c", "w"), col = tim.colors(64), func = mean,
        database = NULL, regions = ".", resolution = c(64, 64), bw = 1, ...)
```

Arguments

`x`	object returned by `georamps` or `predict.ramps`.
`type`	type of plot to produce: `"i"` = `image.plot` (default), `"c"` = `contour` and `image`, and `"w"` = `drape.plot` wireframe.
`col`	vector of colors such as that generated by `rainbow`, `heat.colors`, `topo.colors`, `terrain.colors`, or similar functions.
`func`	function defining the posterior summary statistic to be plotted.
`sites`	logical value indicating whether to include the measurements sites in the plot.
`database`	character string naming a geographical database for the mapping of geographic boundaries. See `map` documentation for details.
`regions`	character vector naming the polygons to draw. See `map` documentation for details.
`resolution`	numerical vector of length 2 specifying the number of pixels (width x height) for the surface image.
`bw`	numerical value specifying the bandwidth used for smoothing the spatial surface as a percentage of the diagonal length of the plot region. Defaults to 1% of the diagonal length.
`...`	additional arguments passed to the underlying plotting function associated with the specified `type` argument.

Author(s)

Brian Smith <brian-j-smith@uiowa.edu>

See Also

georamps predict.ramps contour drape.plot image image.plot map

Examples

```
## Surface maps of the georamps example results

## Not run:
plot(NURE.fit, database = "state", regions = "connecticut",
     resolution = c(200, 150), bw = 5,
     main = "Spatial Process Posterior Mean")

## End(Not run)
```

---

predict.ramps *Prediction Method for georamps Model Fits*

Description

Obtains prediction of main effects plus spatial variability from a georamps model fit.

Usage

```
   ## S3 method for class 'ramps'
predict(object, newdata, type = c("response", "spatial", "error", "random"), ...)
```

Arguments

object	object returned by georamps.
newdata	data frame containing covariate values for the main effect, unmeasured spatial coordinates, and (if applicable) spatial variance indices with which to predict.
type	character string specifying the type of spatial prediction to perform. The default value "response" provides spatial prediction which includes measurement error and non-spatial random effects; "spatial" excludes measurement error and non-spatial random effects from the prediction; "error" excludes non-spatial random effects; and "random" excludes measurement error.
...	some methods for this generic require additional arguments. None are used in this method.

Details

Prediction will be performed only at the coordinates in newdata that differ from those used in the initial georamps model fitting. In particular, overlapping coordinates will be excluded automatically in the prediction.

Value

'predict.ramps' object, inheriting from class 'matrix', of samples from the posterior predictive distribution. Labels for the samples at each new coordinate are supplied in the returned column names and MCMC iteration numbers in the row names. A matrix containing the new coordinates is supplied in the coords attribute of the object.

Author(s)

Brian Smith <brian-j-smith@uiowa.edu>

See Also

georamps plot.predict.ramps, window.predict.ramps,

Examples

```
## Prediction for georamps example results

## Not run:
ct <- map("state", "connecticut", plot = FALSE)
lon <- seq(min(ct$x, na.rm = TRUE), max(ct$x, na.rm = TRUE), length = 20)
lat <- seq(min(ct$y, na.rm = TRUE), max(ct$y, na.rm = TRUE), length = 15)
grid <- expand.grid(lon, lat)

newsites <- data.frame(lon = grid[,1], lat = grid[,2],
                       measurement = 1)
NURE.pred <- predict(NURE.fit, newsites)

par(mfrow=c(2,1))
plot(NURE.pred, func = function(x) exp(mean(x)),
     database = "state", regions = "connecticut",
     resolution = c(200, 150), bw = 5,
     main = "Posterior Mean",
     legend.args = list(text = "ppm", side = 3, line = 1))
plot(NURE.pred, func = function(x) exp(sd(x)),
     database = "state", regions = "connecticut",
     resolution = c(200, 150), bw = 5,
     main = "Posterior Standard Deviation",
     legend.args = list(text = "ppm", side = 3, line = 1))

## End(Not run)
```

---

ramps.control *Auxiliary for Controlling georamps Model Fitting*

Description

Auxiliary function that provides a user interface to control the georamps model fitting algorithm.

Usage

```
ramps.control(iter = 1000, beta, sigma2.e, phi, sigma2.z, sigma2.re,
              z.monitor = TRUE, mpdfun = c("mpdbeta", "mpdbetaz"), file)
```

Arguments

`iter`	numerical value indicating the number of consecutive MCMC samples to generate, or a vector indicating specific iterations to monitor.
`beta`	`'param'` object of initial values and hyperparameters for the main effects coefficients. Flat priors are currently supported for these parameters. Argument is optional if no main effects appear in the model.
`sigma2.e`	`'param'` object of initial values and hyperparameters for the measurement error variances. Inverse-gamma priors are currently supported. Argument is optional if no measurement error variances appear in the model.
`phi`	`'param'` object of initial values and hyperparameters for the spatial correlation parameters. Uniform and user-defined priors are currently supported. Argument is optional if no correlation parameters appear in the model.
`sigma2.z`	`'param'` object of initial values and hyperparameters for the spatial variances. Inverse-gamma priors are currently supported. Argument is optional if no spatial variances appear in the model.
`sigma2.re`	`'param'` object of initial values and hyperparameters for the random effects variances. Inverse-gamma priors are currently supported. Argument is optional if no random effects appear in the model.
`z.monitor`	logical value indicating whether to monitor the latent spatial parameters, or data frame containing a subset of the coordinates at which to monitor the parameters.
`mpdfun`	character string giving the type of marginalized posterior density used for slice sampling and calculation of the data likelihood. Default is marginalization with respect to the beta parameters `"mpdbeta"`, and the alternative is with respect to both the beta and z parameters `"mpdbetaz"`. The latter may provide faster MCMC sampling when analyzing data with multiple observation per measurement site. The two options generate samples from the same posterior distribution.
`file`	vector or list of character strings specifying external files to which to save monitored parameters. Elements of the object named "params" and "z" will be taken to be the output files for model parameters and latent parameters, respectively. If these element names are not supplied, then the first element is taken to be the "params" output file and the second the "z" output file. Defaults to no external outputting of monitored parameters.

Details

Tuning parameters may be set for the `sigma2` and `phi` arguments via the `param` function. If a user-defined prior is specified, then tuning parameters must be supplied and are taken to be the initial widths of the slice sampling windows. Otherwise, tuning parameters are taken to be factors by which the initial widths are multiplied. Separate tuning parameters may be set for each of the arguments. However, only the minimum of all `sigma2` tuning parameters is used in the sampling of those parameters.

Value

A list containing the following components:

- `iter` sorted numerical vector of unique MCMC iterations to be monitored.
- `beta` `'param'` object of initial values for the main effects coefficients.
- `sigma2.e` `'param'` object of initial values for the measurement error variances.
- `phi` `'param'` object of initial values for the spatial correlation parameters.
- `sigma2.z` `'param'` object of initial values for the spatial variances.
- `sigma2.re` `'param'` object of initial values for the random effects variances.
- `z` list with element: `monitor` containing a logical monitoring indicator for the latent spatial parameters or a data frame of coordinates at which to monitor the parameters.
- `mpdfun` character string specifying the marginalized posterior distribution.
- `file` list with elements: `params` and `z` character strings specifying external files to which to save monitored model and spatial parameters.
- `expand` non-negative integer value indicating the starting point of the MCMC sampler, initialized to zero.

Author(s)

Brian Smith <brian-j-smith@uiowa.edu>

See Also

`georamps`, `param`

Examples

```
ctrl <- ramps.control(
   iter = seq(1, 100, by = 2),
   beta = param(rep(0, 2), "flat"),
   sigma2.e = param(rep(1, 2), "invgamma", shape = 2.0, scale = 0.1),
   phi = param(10, "uniform", min = 0, max = 100, tuning = 0.5),
   sigma2.z = param(1, "invgamma", shape = 2.0, scale = 0.1),
   file = c("params.txt", "z.txt")
)
```

---

simJSS *Dataset of Simulated Measurements from JSS Publication*

Description

Simulated Iowa, USA, areal and point-source measurements analyzed in the Working Example of the ramps package paper published in *Journal of Statistical Software*.

Usage

```
data(simJSS)
```

Format

The following variables are provided in the `simIowa` data frame:

`areal` type of measurement: 1 = areal, 0 = point-source.

`y` simulated measurement.

`id` unique identifiers for measurements.

`siteId` unique identifiers for point-source measurement sites.

`lon` longitude coordinates of point-source measurements.

`lat` latitude coordinates of point-source measurements.

`weights` number of sites per measurement.

A grid of coordinates is provided by the `simGrid` data frame to facilitate Monte Carlo integration in geostatistical modeling of areal measurements. The included columns are

`lon` longitude coordinates of grid sites.

`lat` latitude coordinates of grid sites.

`id` county identifiers.

`county` county names.

Areal measurements in `simIowa` can be matched to the grid coordinates in `simGrid` via the shared `"id"` variable.

Details

Areal and point-source observations were generated from from a geostatistical model using the county structure in the state of Iowa, USA. There are 99 counties in the state. Areal observations were generated from each as county averages from a uniform grid of 391 sites - approximately 4 sites per county. An additional 600 point-source observations were generated from a set of 300 unique sites sampled from a uniform distribution in Iowa.

An exponential correlation structure with a range parameter of 10 was used for the underlying Gaussian spatial structure. Measurement errors were generated with variances of 0.25 for point-source data and 0.09 for areal data. Site-specific non-spatial random effects were generated with a variance 0.16. One fixed effects covariate with coefficient equal to 0.5 was included as an indicator for areal observations.

References

Smith, B. J., Yan, J., and Cowles, M. K. (2008) “Unified Geostatistical Modeling for Data Fusion and Spatial Heteroskedasticity with R Package ramps”, *Journal of Statistical Software*, 25(10), 1-21.

Examples

```
data(simJSS)

## Map areal and point-source measurements
y <- simIowa$y[simIowa$areal == 1]
level <- (max(y) - y) / diff(range(y))
map("county", "iowa", fill = TRUE, col = gray(level))
title("Simulated Iowa Measurements")
points(simIowa$lon, simIowa$lat)

## Map grid sites
map("county", "iowa")
title("Regular Grid of Coordinates")
points(simGrid$lon, simGrid$lat)
```

---

summary.ramps *Posterior Summaries of georamps Model Fits*

Description

Posterior summaries of georamps model parameters.

Usage

```
    ## S3 method for class 'ramps'
summary(object, ...)
```

Arguments

object	object returned by georamps.
...	additional arguments to be passed to summary.mcmc.

Value

Two sets of summary statistics for each model parameter. Sample mean, standard deviation, naive standard error of the mean, and time-series-based standard error are included in the first set. Quantiles are included in the second.

Author(s)

Brian Smith <brian-j-smith@uiowa.edu>

See Also

georamps summary.mcmc

Examples

```
## Posterior summaries for georamps example results

## Not run:
summary(NURE.fit)

## End(Not run)
```

---

window — *Subsetting of MCMC Sampler Results*

Description

Post-processing function to subset the MCMC iterations in `georamps` or `predict.ramps` results.

Usage

```
   ## S3 method for class 'ramps'
window(x, iter, ...)

   ## S3 method for class 'predict.ramps'
window(x, iter, ...)
```

Arguments

`x`	object returned by `georamps` or `predict.ramps`.
`iter`	numerical vector specifying the MCMC iterations to subset.
`...`	some methods for this generic require additional arguments. None are used in this method.

Value

Subsetted object of the same class as the one supplied.

Author(s)

Brian Smith `<brian-j-smith@uiowa.edu>`

See Also

`georamps` `predict.ramps`

Examples

```
## Exclude first five iterations of the georamps example results

## Not run:
fit <- window(NURE.fit, iter = 6:25)
print(fit)
summary(fit)

## End(Not run)
```

Index