Package: rns (via r-universe)

September 4, 2024

Version 0.0.2

Date 2020-08-07

Title Functions to fit river network smoothers

Depends mgcv

LazyData yes

Description Functions to fit river network smoothers, ...

License file LICENSE

URL http://www.gov.scot/publications/scotland-river-temperature-monitoring-network-srtmn/

Encoding UTF-8

RoxygenNote 7.2.3

Repository https://faskally.r-universe.dev

RemoteUrl https://github.com/faskally/rns

RemoteRef HEAD

RemoteSha 843b48a9c574626de724ec379a42bb26e7c17e8a

Contents

Predict.matrix.rn.smooth 2
predict_RNS 2
smooth.construct.rn.smooth.spec 4

Index **5**

Predict.matrix.rn.smooth *River Network Smooths*

Description

For data observed over river networks, a simple river network smoother can be appropriate. These functions provide such a smoother class for mgcv.

Usage

```
## S3 method for class 'rn.smooth'
Predict.matrix(object, data)
```

Arguments

object	For the smooth.construct method a smooth specification object, usually generated by a term s(x,...,bs="rn", xt=list(polys=foo)). x is a factor variable giving labels for geographic districts, and the xt argument is obligatory: see details. For the Predict.Matrix method an object of class "rn.smooth" produced by the smooth.construct method.
data	a list containing just the data (including any by variable) required by this term, with names corresponding to object$term (and object$by). The by variable is the last element.

Value

An object of class "rn.smooth" or a matrix mapping the coefficients of the river network smooth to the predictions for the nodes listed in data.

predict_RNS *Predict with and without river network smoothers*

Description

Function to allow you to predict with and without river network smoothers in gam/bam models

Usage

```
predict_RNS(
  gmod,
  newdata,
  RNS_col,
  by_factor_col,
  to_drop_col = NULL,
```

```
  single_RNS = FALSE,
  single_RNS_factor_col = NULL,
  single_RNS_factor = NULL,
  type = "link",
  se.fit = FALSE
)
```

Arguments

gmod	Model that you are predicting with
newdata	Prediction dataset (data points you are predicting for)
RNS_col	column for the river network smoother 'riversmooth' in our case
by_factor_col	column/factors that the river network smoother is different for (e.g. by= in model call) in our case this may be CTMName or Lifestage
to_drop_col	the default is NULL - here any column names you wish to drop can be provide e.g. if you wanted to predict without a hydrometric area smoother NOTE: Random effects columns are automatically removed from the lpmatrix within the function and no longer need to be provided.Similarly for catchments without an RNS you do not need to provide the RNS columns. If you want to predict without RNS, say for partial effects plots, just add in the RNS column (in our case 'riversmooth) and function will generate the appropriate names to drop from lpmatrix
single_RNS	the default is FALSE - RNS will have multiple levels. If the RNS has only one level e.g. a single catchment or lifestage this should be changed to true
single_RNS_factor_col	the default is NULL - if the RNS has only one level (e.g. a single catchment or lifestage) the column where this level could be found in a prediction data should be included here ('CTMName' or 'Lifestage' in our case)
single_RNS_factor	the default is NULL - if the RNS has only one level (e.g. a single
type	the default is "link" (predictions) but can use lpmatrix
se.fit	the default is FALSE (no standard errors) NOTE: if se.fit=T the result returned is a list but using $fit or $se.fit after the model call lets you add each in a column

Details

This function allows you to predict for data in the following formats, using gam or bam models: With river networks smoothers where all prediction data points have an RNS e.g. predicting for observed catchments or lifestages Without river networks smoothers where no prediction data points have an RNS e.g. predicting to observed catchments or lifestages With river networks smoothers AND without river network smoothers in the prediction dataset e.g. predicting to both observed AND unobserved catchments or lifestages

The function has been optimised so there is no longer a requirement to subset data, predictions for all data can be done in one step. Furthermore, there is no need to provide a vector of RNS names or random effects names within the function call these are now taken from the model object and the column used to generate RNS or RE

The function can also deal with circumstances where the RNS may not have multiple levels e.g. a model for a single catchment or lifestage. Predictions can be made with and without RNS as above

Value

If type=="lpmatrix" then a matrix is returned which will give a vector of linear predictor values (minus any offest) at the supplied covariate values, when applied to the model coefficient vector. Otherwise, if se.fit is TRUE then a 2 item list is returned with items (both arrays) fit and se.fit containing predictions and associated standard error estimates, otherwise an array of predictions is returned.

`smooth.construct.rn.smooth.spec`

River Network Smoothers in GAMs

Description

Description here.

Usage

```
## S3 method for class 'rn.smooth.spec'
smooth.construct(object, data, knots)
```

Arguments

`object`	a smooth specification object, usually generated by a term s(...,bs="cr",...), s(...,bs="cs",...) or s(...,bs="cc",...).
`data`	a list containing just the data (including any by variable) required by this term, with names corresponding to object$term (and object$by). The by variable is the last element.
`knots`	a list containing any knots supplied for basis setup — in same order and with same names as data. Can be NULL. See details.

Details

The constructor is not normally called directly, but is rather used internally by gam. To use for basis setup it is recommended to use smooth.construct2.

Value

An object of class "rn.smooth". In addition to the usual elements of a smooth class documented under smooth.construct, this object will contain:

xp giving the knot locations used to generate the basis.

Index

Predict.matrix.rn.smooth, 2
predict_RNS, 2

smooth.construct.rn.smooth.spec, 4
smooth.construct2, *4*