Package 'MapGAM'

November 2, 2026

Type Package

Title Mapping Smoothed Effect Estimates from Individual-Level Data

Version 1.4

Date 2026-10-30

Description Contains functions for mapping odds ratios, hazard ratios, or other effect estimates using individual-level data such as case-control study data, using generalized additive models (GAMs) or Cox models for smoothing with a two-dimensional predictor (e.g., geolocation or exposure to chemical mixtures) while adjusting linearly for confounding variables, using methods described by Kelsall and Diggle (1998), Webster at al. (2006), and Bai et al. (2020). Includes convenient functions for mapping point estimates and confidence intervals, efficient control sampling, and permutation tests for the null hypothesis that the two-dimensional predictor is not associated with the outcome variable (adjusting for confounders).

License GPL-3

Depends R (>= 2.10.0), sp, gam, survival

Imports sf, colorspace, PBSmapping

Suggests maps, mapproj

NeedsCompilation no

Author Lu Bai [aut],
Scott Bartell [aut, cre] (ORCID:
<https://orcid.org/0000-0001-7797-2906>),
Robin Bliss [aut],
Veronica Vieira [aut]

Maintainer Scott Bartell <sbartell@uci.edu>

Repository CRAN

Date/Publication 2026-11-02 16:30:02 UTC

Contents

MapGAM-package 2
AIC.modgam 4

beertweets . . . 5
CAdata . . . 6
CAgrid . . . 6
CAmap . . . 7
coef.gamcox . . . 8
coef.modgam . . . 8
colormap . . . 9
dls . . . 11
formula.gamcox . . . 13
formula.modgam . . . 14
gamcox . . . 14
MAdata . . . 17
MAmap . . . 18
MapGAM.version . . . 18
modgam . . . 19
mypredict.gam . . . 25
optspan . . . 27
plot.modgam . . . 28
predgrid . . . 31
predict.gamcox . . . 33
print.gamcox . . . 34
print.modgam . . . 35
residuals.gamcox . . . 36
residuals.modgam . . . 36
sampcont . . . 37
summary.gamcox . . . 39
summary.modgam . . . 40
toformula . . . 41
trimdata . . . 42

Index **45**

`MapGAM-package` *Mapping Smoothed Effect Estimates from Individual-Level Spatial Data*

Description

Contains functions for estimating and mapping hazard ratios, odds ratios, rate ratios, and continuous outcomes from individual-level data such as case-control study data, using generalized additive models (GAMs) or extended Cox models. This allows for smoothing with a two-dimensional predictor such as geolocation or exposure to chemical mixtures, while adjusting for confounding variables using methods described by Kelsall and Diggle (1998), Webster at al. (2006), Bai et al. (2020), and Tang et al. (2023). Includes convenient functions for mapping, efficient control sampling, confidence intervals, and permutation tests for the null hypothesis that the two-dimensional predictor is not associated with the outcome variable, adjusting for confounders.

Details

Package:	MapGAM
Type:	Package
Version:	1.3
Date:	2023-07-12
License:	GPL-3

Typical spatial applications will start with the predgrid function to create a regular grid of points within the study area, with optional map boundaries (e.g., a country, state, or regional map available in the maps package, or uploaded from a shapefile using the st_read function in the sf package). Crude or adjusted odds ratios, hazard ratios, or continuous outcomes are then estimated at each grid point using the modgam function to smooth by geolocation. Finally, the predicted values (and optionally, "clusters"–areas of signficantly increased or decreased values determined via permutation tests or confidence intervals) are plotted using the plot or colormap functions. Most users will use the plot function to create maps from modgam objects, but colormap is provided for backwards compatibility as well as for creating maps from models fit using functions other than modgam. The trimdata and sampcont functions can be used to restrict data to those within map boundaries and to conduct simple or spatiotemporal stratified sampling from eligible controls. The optspan function can be used to find an optimal span size for the LOESS smoother used by the modgam function; it is automatically used within the modgam function when the span size is not provided by the user. These functions can also be applied to non-spatial data when two-dimensional smoothing is of interest, such as investigation of the effects of a mixture of two chemicals.

Author(s)

Lu Bai, Scott Bartell, Robin Bliss, and Veronica Vieira

Send bug reports to <sbartell@uci.edu>.

References

Bai L, Gillen DL, Bartell SM, Vieira VM. Mapping smoothed spatial effect estimates from individual-level data: MapGAM. The R Journal 2020, 12:32-48. doi:10.32614/RJ2020001

Hastie TJ, Tibshirani RJ. Generalized Additive Models. (Chapman & Hall/CRC Monographs on Statistics & Applied Probability, Boca Raton, Florida, 1990).

Kelsall J, Diggle P. Spatial variation in risk of disease: a nonparametric binary regression approach. J Roy Stat Soc C-App 1998, 47:559-573. doi:10.1111/14679876.00128

Tang IW, Bartell SM, Vieira VM. Unmatched Spatially Stratified Controls: A simulation study examining efficiency and precision using spatially-diverse controls and generalized additive models. Spatial and Spatio-temporal Epidemiology 2023, 45:100584. doi:10.1016/j.sste.2023.100584

Vieira V, Webster T, Weinberg J, Aschengrau A, Ozonoff D. Spatial analysis of lung, colorectal, and breast cancer on Cape Cod: An application of generalized additive models to case-control data. Environmental Health 2005, 4:11. doi:10.1186/1476069X411

Webster T, Vieira V, Weinberg J, Aschengrau A. Method for mapping population-based case-control studies using Generalized Additive Models. International Journal of Health Geographics 2006, 5:26. doi:10.1186/1476072X526

Young RL, Weinberg J, Vieira V, Ozonoff A, Webster TF. A power comparison of generalized additive models and the spatial scan statistic in a case-control setting. International Journal of Health Geographics 2010, 9:37. doi:10.1186/1476072X937

See Also

trimdata, sampcont, predgrid, optspan, modgam, colormap. plot.modgam.

Examples

```
# Load synthetic data and a preformatted base map
data(MAmap)
data(MAdata)

# Create a grid on the base map
gamgrid <- predgrid(MAdata, map=MAmap)
# Fit a GAM with a smooth term for spatial location
fit1 <- modgam(data=MAdata, rgrid=gamgrid, m="crude", sp=0.5)
# Display odds ratio estimates on the base map
plot(fit1, MAmap, exp=TRUE)

#### See colormap and modgam help files for more examples
```

AIC.modgam *AIC of the modgam Object*

Description

This function summarizes the AIC of a modgam object produced by modgam.

Usage

```
## S3 method for class 'modgam'
 AIC(object, ...)
```

Arguments

object	a modgam object.
...	extra arguments for S3 generic, ignored by AIC.modgam.

Author(s)

Lu Bai

Send bug reports to <sbartell@uci.edu>.

See Also

modgam gamcox, predict.gamcox.

`beertweets` *Geocoded Tweets with Beer Indicator*

Description

Geocoded tweets from Twitter, with an indicator variable for any mention of beer, time stamp, and state of origin.

Usage

```
data(MAmap)
```

Format

A data frame with 10000 observations on 5 variables:

`beer` 1 for tweets about beer, 0 for other tweets.

`longitude` geocoded longitude.

`latitude` geocoded latitude.

`state` a factor with the name of the state.

`time` a list of POSIXlt format dates and times for the tweets.

Details

A sample of geocoded tweets from within in the contiguous US, from June to October of 2012. Tweets mentioning beer (cases) are oversampled by a factor of 100. Geocoding is typically at the level of city or town; tweets that could not be geocoded were excluded from this data set.

Source

Dr. Matthew Zook, University of Kentucky, floatingsheep.org

Examples

```
data(beertweets)
attach(beertweets)
plot(longitude,latitude,col=beer+1)  # beer in red, non-beer in black
```

CAdata *Deidentified Survival Data for California*

Description

Survival time, censoring status, and geolocations (jittered to preserve anonymity) for 5000 ovarian cancer patients in California, using a Lambert projection (in meters). CAmap is a map of California using the same projection.

Usage

```
data(CAdata)
```

Format

A data frame with 5000 observations on the following 6 variables.

`time` survival time.

`event` censoring status.

`X` projected X coordinate.

`Y` projected Y coordinate.

`AGE` patient age.

`INS` insurance type: `Mng` (Managed Care: managed care, HMO, PPO and other private insurance), `Mcr` (Medicare), `Mcd` (Medicaid), `Oth`: (Other Insurance: military, county-funded), `Uni` (Not Insured, self-pay) and `Unk` (Unknown).

Examples

```
data(CAdata)
summary(CAdata)
plot(CAdata$X,CAdata$Y)
```

CAgrid *A Grid on State of California*

Description

An evenly-spaced grid of prediction points approximately 5km apart that extended across the latitude and longitude coordinates of participants' locations throughout California with over 7,500 points. Areas with sparse population data are clipped.

Usage

```
data(CAgrid)
```

Format

A data frame with 7579 geolocations.

`X` projected X coordinate.

`Y` projected Y coordinate.

Source

Dr. Veronica Vieira, University of California, Irvine

Examples

```
data(CAgrid)
data(CAmap)
plot(CAmap)
points(CAgrid)
```

CAmap *Map of California*

Description

A map of the outline of CA in SpatialPolygonsDataFrame format, converted from an ESRI shapefile using the `readShapePoly` function in the **maptools** package.

Usage

```
data(CAmap)
```

Format

The format is class SpatialPolygonsDataFrame (package "sp").

Details

State Plane Projected coordinate system, North America Datum 1983 (NAD_1983_StatePlane_California_I_FIPS_0401) for California, using standard parallels 40.00000000 and 41.66666667. The latitude of origin is 39.3, the central meridian is -122.0 and the projection units are meters (False Easting: 200000 m; False Northing: 500000 m).

Source

Dr. Veronica Vieira, University of California, Irvine

Examples

```
data(CAmap)
plot(CAmap)
```

coef.gamcox *Coefficents of the* gamcox *Object*

Description

This function provides coefficients of a gamcox object produced by gamcox.

Usage

```
## S3 method for class 'gamcox'
 coef(object,...)
```

Arguments

object	a gamcox object.
...	.

Author(s)

Lu Bai

Send bug reports to <sbartell@uci.edu>.

See Also

modgam gamcox, predict.gamcox.

Examples

```
data(CAdata)
fit <- gamcox(Surv(time,event)~lo(X,Y)+AGE+INS,data=CAdata, span = 0.2)
coef(fit)
```

coef.modgam *Coefficients of the* modgam *Object*

Description

This function provides coefficients of a modgam object produced by modgam.

Usage

```
## S3 method for class 'modgam'
 coef(object, ...)
```

Arguments

object	a modgam object.
...	extra arguments for S3 generic, ignored by coef.modgam.

Author(s)

Lu Bai

Send bug reports to <sbartell@uci.edu>.

See Also

modgam gamcox, predict.gamcox.

---

colormap *Maps Predicted Values and Clusters on a Two-Dimentional Map*

Description

Displays a color image map, including a legend, scale bar, and optional North arrow, showing values for a grid of points. Irregular grids are allowed. Also draws contour lines for regions of signficantly increased or decreased predicted values ("clusters").

Usage

```
colormap(obj, map=NULL, exp=FALSE, add=FALSE,  mapmin=NULL, mapmax=NULL,
         col.seq=rev(rainbow(201,start=0,end=0.66)), anchor=FALSE,
         border.gray = 0.3, contours="none",
         contours.drawlabels=FALSE, contours.lty=1,
         contours.lwd=1, contours.levels, contours.labcex=0.7,
         interval.exclude=0, arrow=TRUE, axes=FALSE, ptsize=0.9, mai,
         legend.name = "predicted values",legend.cex=1,
         legend.add.line,alpha,...)
```

Arguments

obj	(Required) A list containing at least these two elements: "grid" (a 2 column data frame with X and Y coordinates) and "fit" (the value to displayed on the map).
map	Can be used to map fit on a base map from the map function in the **maps** package, or on a base map produced by the readOGR function in the **rgdal** package. readOGR reads maps from external files in the ESRI shapefile format. map=NULL produces a color image without any base map.
exp	If exp=T, fit will be displayed in exponential scale.
add	Use add=T to add the color map to an existing plot. This will often result in loss of the legend and scale, which are added outside of the normal map boundaries. add is ignored when a map is provided using the map argument.

`mapmin` The minimum value for the color scale legend. When not specified, it will be set to the minimum predicted value.

`mapmax` The maximum value for the color scale legend. When not specified, it will be set to the maximum predicted value.

`col.seq` The color sequence (palette) used to display the predicted values on the map. The default is a rainbow palette, but divergent or sequential palettes may be better for some applications. See the colorspace library and the example below for other options.

`anchor` Use `anchor=TRUE` to center the color palette on `legend.add.line`. When `anchor=FALSE` (the default) or when `legend.add.line` is not specified, the color palette will be centered halfway between mapmin and mapmax. `anchor=TRUE` is recommended when using a divergent palette.

`border.gray` Gray scale for the border of `map`, ranges from 0 to 1. It is white with `border.gray=1`, and black with `border.gray=0`. The defalt scale is 0.9.

`contours` When `contours="interval"`, two other elements `conf.low` and `conf.high` must be included in `obj`, and regions excluding a user-specified `interval.exclude` will be circled on the map. By specifying an element name in `obj` for `contours`, contour lines for the specified element will be added on the map, e.g. `contours="fit"` or `contours="exp.fit"` for a `modgam` object. The default is "none" which produces no contour lines.

`contours.drawlabels`
Use `contours.drawlabels = TRUE` add labels on contour lines.

`contours.lty` The line type for contour lines

`contours.lwd` The width for contour lines.

`contours.levels`
The levels for contour lines. When `exp=T`, the levels will also be used in an exponential scale.

`contours.labcex`
cex for contour labelling. This is an absolute size, not a multiple of `par("cex")`

`interval.exclude`
If `contours="interval"`, regions with confidence intervals excluding `interval.exclude` will be circled

`arrow` Use arrow=T to add a North arrow to the map.

`axes` Use axes=T to add axes to the map (useful for chemical mixture isoboles).

`ptsize` The size of the points used to fill in colors on the map. Increase to remove white space inside the map or decrease to remove color outside the map boundaries. NOTE: white space can also be eliminated by increasing the grid size in `predgrid`, which is often preferable as it results in a higher resolution map.

`mai` The margins of the plot. Details see par,

`legend.name` The name of displayed for the legend bar.

`legend.cex` A numerical value giving the amount by which legend text should be magnified relative to the default.

`legend.add.line`
A numerical value at which a line will be added on the legend bar to indicate the color for the value.

alpha	Levels
...	Other auguments past to plot function

Value

Produces an image map. If the base map is in readOGR format a scale bar is included, showing whatever measurement units are recorded in the shapefile. If the units are missing from the shapefile conversion they are assumed to be meters.

Author(s)

Scott Bartell, Lu Bai, Robin Bliss, and Veronica Vieira

Send bug reports to <sbartell@uci.edu>.

See Also

trimdata, predgrid, plot.modgam.

Examples

```
data(MAdata)
data(MAmap)
obj <- list(grid=data.frame(MAdata$Xcoord,MAdata$Ycoord),fit=MAdata$Mercury)
colormap(obj, MAmap, legend.name = "mercury")

# map the same data using a divergent color palette anchored to the median
if (require(colorspace)) {
  newpal <- diverge_hsv(201)   # from the colorspace library
  colormap(obj, MAmap, legend.name = "mercury", col.seq = newpal,
 legend.add.line=round(median(obj$fit),2), anchor = TRUE)
  }
```

---

dls *Calculating Derivatives of Partial Likelihood for Cox Proportional Hazard Additive Models*

Description

Calculate the log partial likelihood and derivatives with respect to the subject log hazard ratio (compared to the baseline) for Cox proportional hazard additive model described in gamcox. Results are used to update estimates in gamcox function.

Usage

```
dls(Y,X,which,eta,span=0.5,adjust=TRUE)
```

Arguments

Y	a list including two elements: time for survival times and event for censoring statu.
X	a data frame containing the variables in the model. The data must be structured so that the X and Y coordinates for two-dimensional predictor (e.g., geolocation) are in the 1st and 2nd columns, respectively.
which	matrix index for smooth term.
eta	current estimated subject log hazard ratio compared to the baseline.
span	smoothing parameter that been used to smoothing the second derivative of the log partial likelihood.
adjust	adjust=TRUE means there are confounders included in the model.

Details

For data that having tied failure times, Efron's approximation method is used to calculate the log partial likelihood and correspongding derivatives. Let η denote the log hazard ratio, and l denote the partial likelihood. When fitting a Cox proportional hazard additive model, η is updated by

$$\eta^{new} = \eta^{old} - \frac{dl/d\eta}{smooth(d^2l/d\eta^2)}$$

Value

deltaeta	difference between the input eta and the new updated eta.
w	inverse of smoothed second derivatives.
l	partial likelihood baed on input eta.

Author(s)

Lu Bai and Scott Bartell

Send bug reports to <sbartell@uci.edu>.

References

Hastie TJ, Tibshirani RJ. Generalized Additive Models. (Chapman & Hall/CRC Monographs on Statistics & Applied Probability, Boca Raton, Florida, 1990).

Bristow RE, Chang J, Ziogas A, Gillen DL, Bai L, Vieira VM. Spatial Analysis of Advanced-stage Ovarian Cancer Mortality in California. American Journal of Obstetrics and Gynecology 2015, 213(1), e1-43).

See Also

gamcox, predict.gamcox.

Examples

```
data(CAdata)
Y = CAdata[,c("time","event")]
X = CAdata[,c(3:5)]
eta = coxph(Surv(time,event)~AGE,data=CAdata)$linear.predictors
result = dls(Y,X,1:2,eta,span=0.2)
plot(eta,result$deltaeta)
```

`formula.gamcox` *Formula of the* gamcox *Object*

Description

This function provides the formula of a gamcox object produced by gamcox.

Usage

```
## S3 method for class 'gamcox'
 formula(x,...)
```

Arguments

x	a gamcox object.
...	.

Author(s)

Lu Bai

Send bug reports to <sbartell@uci.edu>.

See Also

modgam gamcox, predict.gamcox.

Examples

```
data(CAdata)
fit <- gamcox(Surv(time,event)~lo(X,Y)+AGE+INS,data=CAdata, span = 0.2)
formula(fit)
```

formula.modgam *Formula of the* modgam *Object*

Description

This function provide the formula of a modgam object produced by modgam.

Usage

```
## S3 method for class 'modgam'
 formula(x, ...)
```

Arguments

x	a modgam object.
...	extra arguments for S3 generic, ignored by formula.modgam.

Author(s)

Lu Bai

Send bug reports to <sbartell@uci.edu>.

See Also

modgam gamcox, predict.gamcox.

gamcox *Fit a Cox Additive Model with a Two-Dimensional Smooth*

Description

Fits a Cox proportional hazard additive model with a two-dimensional LOESS smooth for geolocation (or any other two-dimensional predictor). gamcox uses the backfitting algorithm to combine the smoothing and fitting methods. The smoothing method currently supported is local regression (LOESS).

Usage

```
gamcox(formula, data, subset, weights, span=0.5, I.span=0.2, degree = 1,
       loess.trace = "exact", Maxiter = 40, tol = 1e-07)

gamcox.fit(Y, X, smooth.frame, weights, span=0.5, I.span=0.2, degree = 1,
           loess.trace = "exact", Maxiter = 40, tol = 1e-07)
```

Arguments

formula	a formula expression (required), with the response on the left of a ~ oprator, and the predictors on the right. The response must be a survival object as returned by the Surv function. A built-in nonparametric smoothing term is indicated by lo for loess smooth terms. The two-dimensional predictor (e.g.,geolocation) should be specified as auguments in lo().
data	a data frame containing the variables in the model.If not found in data, the variables are taken from environment(formula).
Y, X	for gamcox.fit: Y is a list including two elements: time for survival times and event for censoring status. X is a data frame containing the variables in the model. The data must be structured so that the X and Y coordinates for two-dimensional predictor (e.g., geolocation) are in the 1st and 2nd columns, respectively.
smooth.frame	the model matrix for the smooth term.
subset	an optional vector specifying a subset of observations to be used in the fitting process.
weights	an optional vector of weights to be used in the fitting process.
span	the span size for the LOESS smooth of the two-dimensional predictor, which controls the degree of smoothing.
I.span	the span size for the LOESS smooth of the Fisher information.
degree	the degree of the polynomials to be used for LOESS smooth, normally 1 or 2.
loess.trace	whether the trace of the smoother matrix be computed exactly ("exact") or approximately ("approximate"). It is recommended to use "approximate" for more than about 1000 data points.
Maxiter	the maximum number of iterations in backfitting algorithm.
tol	the tolerence threshold for convergence.

Details

The model used to fit the data is a Cox proportional hazard additive model with a LOESS smooth for a two-dimensional predictor such as geolocation (Hastie and Tibshirani, 1990):

$$\boldsymbol{\lambda}(t) = \boldsymbol{\lambda}_0(t) \exp\{S(x_i, y_i) + \mathbf{Z_i}\boldsymbol{\beta}\}$$

where $\boldsymbol{\lambda}(t)$ is the hazard at time *t* for participant *i*, x_i and y_i are predictor coordinates for participant *i* (i.e., projected distance east and north, respectively, from an arbitrarily defined origin location), *S(.,.)* is a 2-dimensional smoothing function (currently LOESS), $\mathbf{Z_i}$ is a row vector of covariate values for participant *i*, and $\boldsymbol{\beta}$ is a vector of unknown regression parameters. See the references for more details.

Value

gamcox returns an object of class gamcox. It can be examined by print and predict.

`coefficients` a named vector of coefficients for the parametric part of the additive predictors, which multiply the columns of the model matrix. The names of the coefficients are the names of the single-degree-of-freedom effects (the columns of the model matrix). If the model is overdetermined there will be missing values in the coefficients corresponding to inestimable coefficients.

`additive.predictors` the additive fit, given by the product of the model matrix and the coefficients, plus the columns of the `$smooth` component.

`smooth` estimated smoothing term. Nonlinear part of the spatial effect on survival rates.

`var` the approximate pointwise variances for the columns of smooth.

`residuals` the residuals from the final weighted additive fit; also known as residuals, these are typically not interpretable without rescaling by the weights.

`deviance` up to a constant, minus twice the maximized log-likelihood. Similar to the residual sum of squares. Where sensible, the constant is chosen so that a saturated model has deviance zero.

`df.residual` the residual degrees of freedom.

`aic` AIC of the fitted model.

Author(s)

Lu Bai

Send bug reports to <sbartell@uci.edu>.

References

Hastie TJ, Tibshirani RJ. Generalized Additive Models. (Chapman & Hall/CRC Monographs on Statistics & Applied Probability, Boca Raton, Florida, 1990).

Bristow RE, Chang J, Ziogas A, Gillen DL, Bai L, Vieira VM. Spatial Analysis of Advanced-stage Ovarian Cancer Mortality in California. American Journal of Obstetrics and Gynecology 2015, 213(1), e1-43)

See Also

modgam, predict.gamcox.

Examples

```
data(CAdata)
data(CAmap)

fit <- gamcox(Surv(time,event)~AGE + factor(INS) + lo(X,Y),data=CAdata,
span=0.2,loess.trace="approximate")
fit
pred = predict(fit)
colormap(list(fit=pred$pred,grid=data.frame(X=CAdata$X,Y=CAdata$Y)),map=CAmap,
border.gray=0.5)
```

MAdata *Synthetic Case-Control Data for Massachusetts*

Description

90 cases and 910 controls with random smoking covarate values and random geolocations within Massachusetts, geocoded on a Lambert projection (in meters). MAmap is a map of Massachusetts using the same projection.

Usage

```
data(MAdata)
```

Format

A data frame with 1000 observations on the following 6 variables.

`Case` 0 for controls, 1 for cases.

`Xcoord` projected X coordinate.

`Ycoord` projected Y coordinate.

`Smoking` 0 for nonsmokers, 1 for smokers.

`Mercury` continuous variable for mercury exposure.

`Selenium` continuous variable for selenium exposure.

Details

Lambert conformal conic projection for the State of Massachusetts, using standard parallels 41.71666667 and 42.68333333. The latitude of origin is 41.0, the central meridian is -71.5, and the projection units are meters (False Easting: 200000 m; False Northing: 750000 m).

Source

Simulated data provided by package authors

Examples

```
data(MAdata)
summary(MAdata)
attach(MAdata)
# map participants, cases in red and controls in black
plot(Xcoord,Ycoord,col=Case+1)
```

MAmap *Map of Massachusetts*

Description

A map of the outline of MA in SpatialPolygonsDataFrame format, converted from an ESRI shapefile using the `readShapePoly` function in the **maptools** package.

Usage

```
data(MAmap)
```

Format

The format is class SpatialPolygonsDataFrame (package "sp").

Details

Lambert conformal conic projection for the State of Massachusetts, using standard parallels 41.71666667 and 42.68333333. The latitude of origin is 41.0, the central meridian is -71.5, and the projection units are meters (False Easting: 200000 m; False Northing: 750000 m).

Source

Dr. Veronica Vieira, University of California, Irvine

Examples

```
data(MAmap)
plot(MAmap)
```

MapGAM.version *Revision Dates for MapGAM Functions*

Description

Print version and dates for MapGAM package and functions.

Usage

```
MapGAM.version()
```

Author(s)

Scott Bartell Send bug reports to <sbartell@uci.edu>.

See Also

MapGAM-package,

---

modgam *Fit a Generalized Additive Model (GAM) with a Two-Dimensional Smooth and Make Predictions*

Description

Fits a crude or adjusted regression on a user-supplied grid for spatial analysis using a generalized additive model with a two-dimensional LOESS smooth for geolocation (or any other two-dimensional predictor). Includes optional pointwise confidence intervals and permutation tests for global and local tests of the null hypothesis that the two-dimensional predictor (e.g., geolocation) is not associated with the outcome. Most applications will pass the output of this function to `plot.modgam` to map the resulting odds ratios or other effect estimates.

Usage

```
modgam(formula, rgrid, data, subset, offset, family = binomial(), permute = 0,
     conditional = TRUE, pointwise = FALSE, m = "adjusted", sp = seq(0.05,0.95,0.05),
       degree=1, keep=FALSE, type=c("spatial","all"), reference = "median",
       se.fit = FALSE, alpha = 0.05, verbose = TRUE, ...)
```

Arguments

`formula`	a formula expression, with the response on the left of a ~ oprator, and the predictors on the right. If fitting a Cox additive model, the response must be a survival object as returned by the Surv function. A built-in nonparametric smoothing term is indicated by `lo` for loess smooth terms. The two-dimensional predictor (e.g.,geolocation) should be specified as auguments in `lo()`. `formula` can be missing if `data` and `family` are specified.
`rgrid`	a data frame containing the values of X and Y coordinates at which to generate predictions. If missing, the predictions will be generated for the two-dimentional predictor in `data`. The `predgrid` function can be used to supply an appropriate rgrid for rdata, with the grid clipped according to any specified base map (see examples below).
`data`	a data frame containing the variables in the model. If `formula` is specified, `data` is optional. In this way, if not found in `data`, the variables are taken from `environment(formula)`. If `formula` is missing, the data must be structured so that the outcome is in the 1st column (1st and 2nd colums for survival data) and the X and Y coordinates for two-dimensional predictor (e.g., geolocation) are in the 2nd and 3rd columns (3rd and 4th columns for survival data), respectively. If more than 3 (4 for survival data) columns are provided and `m = "adjusted"`, the additional columns will be entered as linear predictors in the GAM model.

`family` a description of the error distribution and link function to be used in the model. This can be a character string naming a family function, a family function or the result of a call to a family function. (See `family` for details of family functions. Note that unlike other packages, MapGAM defaults to the binomial family and logit link, i.e., a logistic model.). If `formula` is missing, `family = "survival"` must be specified to fit a Cox proportional hazard additive model.

`subset` an optional vector specifying a subset of observations to be used in the fitting process.

`offset` name in data and rgrid that will be used as offset in the model. It will be ignored if offset is specified in `formula`. `offset` will be ignored if `family="survival"`

`permute` the number of permutations of the data set for testing the significance of the two-dimensional predictor. `permute = 0` (default) produces no permutation tests. If `permute > 0`, the paired coordinates for the two-dimensional predictor are randomly permuted in order to simulate the distribution of results under the null hypothesis that location is not associated with the outcome. 1000 permutations are recommended for reasonable accuracy of p-values. WARNING: Because each permutation requires refitting the GAM, permutation tests can be quite slow.

`conditional` a logical value indicating whether to run a conditional or unconditional permutation test; this argument is used only when `permute > 0`. The default is a conditional permutation test: the span size is held fixed throughout all permutations even if `sp = NULL` to optimize the span size for the original data (see Young et al., 2012). The unconditional permutation test repeats the span size optimization for each permutation, which is more conservative but takes about 20 times longer to compute. If `conditional = FALSE` the sp argument is ignored when fitting both the original and permuted data sets.

`pointwise` a logical value indicated whether to include pointwise permutation tests for every grid point. This argument is only used when `permute > 0`. Using `se.fit = TRUE` to determine grid points with significantly different outcomes is usually preferable to using pointwise permutation tests.

`m` model type for the GAM. Options are "crude" or "adjusted" (default). If "crude", only the smooth for the two-dimensional predictor is included in the model (i.e., "crude" is a synonym for "unadjusted"). If "adjusted", all covariates in the data set (columns >= 4) are included in the model.

`sp` the span size for the LOESS smooth of the two-dimensional predictor. If `sp` is a vector an optimal span size will be determined using the `optspan` function, using the vector as candidate values. By default the optimal span is chosen from among a sequence of values ranging from 0.05 to 0.95, in increments of 0.05. See the help file for `optspan` for more details. It will be ignored if span is indicated in `formula`.

`degree` the degree of the polynomials to be used for LOESS smooth, normally 1 or 2. It will be ignored if the degree is indicated in `formula`.

`keep` a logical value indicating whether to store and return the pointwise odds ratios, and if conditional = FALSE the span size, for every permuted data set. These values aren't necessary for mapping or cluster identification, and storing them slows the permutation tests, so the default is `keep = FALSE`.

type	use type="spatial"(default for censored survival data) to estimate the spatial effect, or type="all"(default for univariate outcome) to estimate the effect of all covariates included in the model.
reference	referent for the predicted values or effects. If reference="none", the output will be predicted values at each grid point for either the spatial effect (type="spatial") or the linear predictor including all covariates (type="all"). If reference = "median", the output will be the difference between the estimated effect/predictor at each grid point and the median effect/predictor (e.g., the log odds ratio for a binomial model with logit link). If reference = "mean", the output will be the difference between the estimated effect/predictor at each grid point and the mean effect/predictor. If reference is a data frame indicating a specific geolocation, the output will be the difference between the estimated effect/predictor at each grid point and the effect/predictor at the geolocation specified by reference.
se.fit	if TRUE, pointwise standard errors and confidence intervals are computed along with the predictions. Results can be passed to plot.modgam with contours = "interval" to identify clusters of the outcome (e.g., spatial regions with statistically significant outcomes–e.g., unusually high or low risks).
alpha	the significace level used for confidence intervals when se.fit = TRUE
verbose	a logical value indicating whether to print the GAM model statement, the percentile rank for the global deviance statistic in the permutation test, and the progress of the permutation test (report completion of every 10 permutations). The default is verbose = TRUE.
...	further arguments to be passed to the gam or gamcox function (e.g., weights).

Details

The model used to fit the data is a generalized additive model with a LOESS smooth for a two-dimensional predictor such as geolocation (Hastie and Tibshirani, 1990; Kelsall and Diggle, 1998; Webster et al., 2006). Although any family and link function supported by the gam function is supported, the default binomial family with logit link yields the following model:

$$\ln\left(\frac{\pi_i}{1-\pi_i}\right) = S(x_i, y_i) + \mathbf{Z_i}\boldsymbol{\beta}$$

where π_i is the probability that the outcome is 1 for participant *i*, x_i and y_i are predictor coordinates for participant *i* (i.e., projected distance east and north, respectively, from an arbitrarily defined origin location), $S(.,.)$ is a 2-dimensional smoothing function (currently LOESS), $\mathbf{Z_i}$ is a row vector of covariate values for participant *i*, and $\boldsymbol{\beta}$ is a vector of unknown regression parameters (including an intercept). When a permutation test is requested, for each permutation the paired X and Y coordinates in the data set are randomly reassigned to participants, consistent with the null hypothesis that the geolocation (or another two-dimensional predictor entered in place of X and Y) is not associated with the outcome. Note that this procedure intentionally preserves associations between other covariates and the outcome variable so the permutation test reflects the significance of geolocation. See the references for more details.

Value

modgam returns an object of class modgam. It can be examined by print and plot.

`grid` a data frame with X and Y coordinates from rgrid.

`span` the span size used for the LOESS smooth. If `keep = TRUE` and `conditional = FALSE`, a vector with optimized span sizes for the original data set and each of the permuted data sets.

`gamobj` the `gam` or `gamcox` model object from the fit to the data.

`predobj` the `mypredict.gam` or `predict.gamcox` model object from the fit to the data.

`family` the family and link function used for the model.

`fit` the predicted values for each point on the grid. For Gaussian family models, predicted values are for the outcome variable at each grid point. For non-Gaussian families, predicted values are for the spatial effect alone on the scale of the linear predictor, compared to a referent specified through augument `reference` (e.g., for a binomial family logit link model with `reference = "median"`, the fit values are the difference in log odds of the outcome between the grid point and the median log odds, holding other predictors constant). If predicted responses are desired for non-Gaussian families, they can be obtained from the fit values by setting `reference="none"` and applying the inverse link function.

`exp.fit` the exponential of `fit`. Primarily useful for logit or log link functions (e.g., this provides odds ratios for a binomial family with logit link function.)

`global.pvalue` the p-value based on the likelihood ratio test comparing models with and without geolocation (or other two-dimensional predictor).

If `se.fit = TRUE` then the following values are provided:

`se` an estimated standard error for each predicted value.

`conf.low` the lower bounds for pointwise (1-`alpha`) confidence intervals.

`conf.high` the higher bounds for pointwise (1-`alpha`) confidence intervals.

If `permute > 0` then the following values are also provided:

`global.permt` the p-value based on the deviance statistic comparing models with and without geolocation (or other two-dimensional predictor), using the distribution of deviance statistics from the permuted data sets to represent the null distribution. For a test of H0: geolocation is unassociated with the outcome variable (adjusting for any covariates if `m = "adjusted"`), reject H0 if the percentile rank is below alpha. WARNING: by default `modgam` uses a conditional permutation test which produces inflated type I error rates; Young et al. (2012) recommend using alpha=0.025 to limit the type I error rate to approximately 5%.

`pointwise.permt`
Only provided when `pointwise = TRUE`. For each point on the grid, the percentile rank of the local linear predictor for the model compared to the local linear predictor distribution from the permuted data sets. This result can be used to define clusters of the outcome (e.g., spatial regions with statistically significant outcomes–e.g., unusually high or low risks), as in `plot.modgam` with `contours="permrank"`.

If `permute > 0` and `keep = T` then the following values are also provided:

`permutations` a matrix containing null permuted values for each point on the grid, with the results for each permutation in a separate column. For the binomial family and logit link these are provided as odds ratios, otherwise they are reported as linear predictors.

`deviances.permt`
a vector containing deviances for each permutation.

Warning

Permutation tests are computationally intensive, often requiring several hours or more.

Author(s)

Lu Bai, Scott Bartell, Robin Bliss, and Veronica Vieira

Send bug reports to <sbartell@uci.edu>.

References

Bai L, Gillen DL, Bartell SM, Vieira VM. Mapping smoothed spatial effect estimates from individual-level data: MapGAM. The R Journal 2020, 12:32-48. doi:10.32614/RJ2020001

Hastie TJ, Tibshirani RJ. Generalized Additive Models. (Chapman & Hall/CRC Monographs on Statistics & Applied Probability, Boca Raton, Florida, 1990).

Kelsall J, Diggle P. Spatial variation in risk of disease: a nonparametric binary regression approach. J Roy Stat Soc C-App 1998, 47:559-573. doi:10.1111/14679876.00128

Tang IW, Bartell SM, Vieira VM. Unmatched Spatially Stratified Controls: A simulation study examining efficiency and precision using spatially-diverse controls and generalized additive models. Spatial and Spatio-temporal Epidemiology 2023, 45:100584. doi:10.1016/j.sste.2023.100584

Vieira V, Webster T, Weinberg J, Aschengrau A, Ozonoff D. Spatial analysis of lung, colorectal, and breast cancer on Cape Cod: An application of generalized additive models to case-control data. Environmental Health 2005, 4:11. doi:10.1186/1476069X411

Webster T, Vieira V, Weinberg J, Aschengrau A. Method for mapping population-based case-control studies using Generalized Additive Models. International Journal of Health Geographics 2006, 5:26. doi:10.1186/1476072X526

Young RL, Weinberg J, Vieira V, Ozonoff A, Webster TF. A power comparison of generalized additive models and the spatial scan statistic in a case-control setting. International Journal of Health Geographics 2010, 9:37. doi:10.1186/1476072X937

See Also

`predgrid`, `optspan`, `plot.modgam`, `colormap`, `gam`.

Examples

```
# Load base map in SpatialPolygonsDataFrame format
# This map was read from ESRI shapefiles using the readShapePoly function (soon to be deprecated)
data(MAmap)
# Load data and create grid on base map
```

```
data(MAdata)
gamgrid <- predgrid(MAdata, map=MAmap)
# Fit crude logistic GAM to the MA data using span size of 50%
# and predict odds ratios for every point on gamgrid
fit1 <- modgam(data=MAdata, rgrid=gamgrid, m="crude", sp=0.5)
# Summary statistics for pointwise crude odds ratios
summary(fit1$exp.fit)
# Summary stats for pointwise crude log odds (linear predictor)
summary(fit1$fit)

# fit adjusted GAM using span size of 50%,
# including a (too small) conditional permutation test
fit2 <- modgam(data=MAdata, rgrid=gamgrid, permute=25, m="adjusted", sp=0.5)
fit2

# fit adjusted GAM by specifiying formula
fit2.f <- modgam(Case~lo(Xcoord,Ycoord)+Smoking + Mercury + Selenium,data=MAdata,
          rgrid=gamgrid, sp=0.5)
fit2.f

# Detailed example with a continuous outcome variable, map projections,
# and data trimming:  investigating tweet times by geolocation
# NOTE: this example requires the maps and mapproj packages
# Convert base map and beer tweet data locations to US Albers projection
# for better distance estimates than using (lat,long) as (X,Y) coords
if(require(maps) & require(mapproj)) {
 USmap <- map("state",projection="albers",parameters=c(29.5,45.5),
  plot=FALSE,fill=TRUE,col="transparent")
 data(beertweets)
 # Reuse last map projection to convert data coordinates
 XY <- mapproject(beertweets$longitude,beertweets$latitude)[1:2]
 jtime <- julian(beertweets$time)
 # Convert tweet dates and times to time of day (24-hour)
 tweettime <- as.numeric(jtime-trunc(jtime))*24
 beerproj <- data.frame(tweettime, XY[1], XY[2], beertweets$beer)
 # Generate grid on the US map, trimmed to range of beer data
 USgrid <- predgrid(beerproj, map=USmap)
 # Fit adjusted model--adjusting for beer indicator variable
 fit3 <- modgam(data=beerproj, rgrid=USgrid, family=gaussian,
  reference="none", m="adjusted", sp=0.05)
 # Get summary statistics for predicted tweet times across grid points
 summary(fit3$fit)
}

# Smoothing survival rates over geolocations with adjustement of Age and Insurance
# Including generating pointwise standard errors and confidence intervals
data(CAdata)
data(CAgrid)
data = CAdata[1:1000,] # use a subset of the California data
# manually set the categorical variables as factors
data$INS = factor(data$INS)
# no formula needed if data are arranged in a specific order (see \code{data} argument)
fit4 <- modgam(data=data, rgrid=CAgrid, family="survival", sp=0.2)
```

```
fit4
# fit the same model using the formula argument, with data columns in any order
fit4.f <- modgam(Surv(time,event)~AGE+factor(INS)+lo(X,Y), data=data,
                rgrid = CAgrid, family="survival", sp=0.2)

# Smoothing for two-dimensional chemical exposure instead of geolocation
# case status in 1st column, mercury and selenium in 2nd and 3rd columns
ma2 <- MAdata[,c(1,5:6)]
expgrid <- predgrid(ma2)
fit5 <- modgam(data=ma2,rgrid=expgrid,sp=.5,m="crude")
summary(fit5$exp.fit)
# plot the results, with mercury on the X axis and selenium on the Y axis
plot(fit5, contours="response", arrow=FALSE, axes=TRUE)
```

`mypredict.gam` *Prediction for GAM Fits*

Description

Obtains spatial effects predictions and optionally estimates standard errors and confidence intervals of those predictions from a fitted generalized additive model object.

Usage

```
mypredict.gam(object, newdata, se.fit = FALSE, type=c("all","spatial"),
              reference = "median", level = 0.05,verbose=FALSE)
```

Arguments

`object`	a fitted `gam` object.
`newdata`	a data frame containing the values at which predictions are required. This argument can be missing, in which case predictions are made at the same values used to compute the object. Only two-dimentional predictor need be present by name in newdata.
`se.fit`	if TRUE, pointwise standard errors and confidence intervals are computed along with the predictions.
`type`	use `type="spatial"` to estimate spatial effect, and use `type="all"`(default) to estimate the effect of all covariates included in the model.
`reference`	the type of reference for the estimated effect. If `reference = "median"`, the output will be the estimated effect difference (or log ratio) compared to the median effect. If `reference = "mean"`,the output will be the estimated effect difference (or log-ratio) compared to the mean effect.If `reference` is a data frame indicating a specific geolocation, the output will be the estimated effect difference (or log-ratio) compared to the effect of the geolocation specified by `reference`.
`level`	the siginificance level used when `se.fit=TRUE`.
`verbose`	a logical value indicating whether to print filling values for newdata. The default is `verbose = FALSE`.

Value

pred	the estimated effect difference or (log ratio) compare to the effect specified by reference.
se	the standard errors along with the predictions.
conf.low	the lower bounds for pointwise (1-level) confidence intervals.
conf.high	the higher bounds for pointwise (1-level) confidence intervals.

Author(s)

Lu Bai

Send bug reports to <sbartell@uci.edu>.

References

Hastie TJ, Tibshirani RJ. Generalized Additive Models. (Chapman & Hall/CRC Monographs on Statistics & Applied Probability, Boca Raton, Florida, 1990).

See Also

modgam, predict.gamcox.

Examples

```
data(MAdata)
data(MAmap)
gamgrid <- predgrid(MAdata, map=MAmap)

fit <- gam(Case~lo(Xcoord,Ycoord,span=0.2)+Smoking,data=MAdata,family=binomial())
pred1 = mypredict.gam(fit)
colormap(list(fit=pred1$pred,grid=data.frame(X=MAdata$X,Y=MAdata$Y)),map=MAmap)

pred2 = mypredict.gam(fit,gamgrid)
colormap(list(fit=pred2$pred,grid=data.frame(X=gamgrid$X,Y=gamgrid$Y)),map=MAmap)

pred3 = mypredict.gam(fit,gamgrid,se.fit=TRUE)
colormap(list(fit=pred3$pred,conf.low = pred3$conf.low, conf.high = pred3$conf.high,
          grid=data.frame(X=gamgrid$X,Y=gamgrid$Y)),map=MAmap,contours = "interval")
```

optspan *Determine the Optimal Span Size for* modgam

Description

Determines the optimal span size for modgam, a spatial generalized additive model (GAM) with a two-dimensional LOESS smooth for location, by minimizing the AIC.

Usage

```
optspan(formula, data, offset,spans = seq(0.05, 0.95, by = 0.05), m ="adjusted",
        family = binomial(), verbose =TRUE, degree = 1, ...)
```

Arguments

formula	a formula expression for the model, with the response on the left of a ~ oprator, and the predictors on the right. If fitting a Cox additive model, the response must be a survival object as returned by the Surv function. A built-in nonparametric smoothing term is indicated by lo for loess smooth terms. The two-dimensional predictor (e.g.,geolocation) should be specified as auguments in lo(). formula can be missing if data is well structured and family is specified.
data	a data frame containing the variables in the model. data can be missing if formula is specified. If formula is missing, the data must be structured so that the outcome variable is in the 1st (1st and 2nd for survival data) column and X and Y location values are in the 2nd and 3rd (3rd and 4th for survival data) columns. Any additional columns will be entered as covariates with linear effects in the model.
offset	the name in data that will be used as offset in the model. It will be ignored if offset is specified in formula.
spans	a vector of candidate span sizes that the optimal span is chosen from.
m	the model type. Options are "crude" or "adjusted" (default). If "crude", only the spatial smooth term for location is included in the model. If "adjusted", all covariates in the data set (columns >= 4) are included in the model.
family	a family function or the result of a call to a family function. (See family for details of family functions. famil = "survival" for survival data.
verbose	a logical argument; if TRUE shows the AIC for each candidate span size.
degree	a smoothing parameter used in modgam.
...	any additional arguments to pass to the gam function.

Value

The optimal span size, determined by minimizing the AIC

Note

This function does not return model predictions–only the optimal span size. To obtain model predictions use the modgam function.

Author(s)

Lu Bai, Scott Bartell, Robin Bliss, and Veronica Vieira

Send bug reports to <sbartell@uci.edu>.

See Also

predgrid, modgam, colormap.

Examples

```
data(MAdata)
optspan(data=MAdata, m="crude")

data(CAdata)
optspan(Surv(time,event)~AGE+factor(INS)+lo(X,Y), data=CAdata,
        spans=c(0.1,0.2,0.3),family="survival")
```

---

plot.modgam *Maps Predicted Values and Clusters for* modgam *Objects*

Description

Displays a color image map, including a legend, scale bar, and optional North arrow, showing crude or adjusted odds ratios (or linear predictors) for a grid of points. Irregular grids are allowed. Also draws contour lines for regions of signficantly increased or decreased values of the outcome variable ("clusters"), if permutation ranks are provided. Designed to display modgam objects but can be used with other model results if the modgamobj list contains the correct elements.

Usage

```
## S3 method for class 'modgam'
plot(x, map = NULL, exp = FALSE, add = FALSE, intervals=TRUE,
       mapmin = NULL, mapmax = NULL, col.seq = diverge_hsv(201), anchor=FALSE,
       border.gray = 0.3, contours=c("none","response","permrank","interval"),
       contours.drawlabels=FALSE, contours.lty=1, contours.lwd=1,
      contours.levels, contours.labcex=0.7, arrow=TRUE, axes=FALSE, ptsize=0.9,
       alpha=0.05, mai, legend.name = "predicted values", legend.cex=1, ...)
```

Arguments

x	(required) an object of class modgam, usually returned by the modgam function.
map	can be used to map predicted values on a base map from the map function in the **maps** package, or on a base map produced from an external shapefile using the st_read function in **sf** package. Note that MapGAM used to include examples using the readShapePoly function in the **maptools** package, but that package was removed from the CRAN repository in late 2023. map=NULL produces a color image without any base map.
exp	if exp=T, fit will be displayed in exponential scale.
add	use add=T to add the color map to an existing plot. This will often result in loss of the legend and scale, which are added outside of the normal map boundaries. add is ignored when a map is provided using the map argument.
intervals	if the modgam object contains confidence intervals, the lower and higher intervals will be plotted if intervals=T.
mapmin	the minimum value for the color scale legend.
mapmax	the maximum value for the color scale legend.
col.seq	The color sequence (palette) used to display the predicted values on the map.
anchor	Use anchor=TRUE to center the color palette on legend.add.line. When anchor=FALSE or when legend.add.line is not specified, the color palette will be centered halfway between mapmin and mapmax. anchor=TRUE is recommended when using a divergent palette.
border.gray	gray scale for the border of map, ranges from 0 to 1. It is white with border.gray=1, and black with border.gray=0. The defalt scale is 0.3.
contours	use contours="response" to add contour lines for the predicted response, for example to draw isoboles for mixtures of exposures. Use contours="permrank" to add contour lines for pointwise p-values computed from the permutation ranks, at alpha/2 and (1-alpha)/2. use contours = "intervals" to add contour lines for areas where the confidence intervals do not include 0. The default is "none" which produces no contour lines.
contours.drawlabels	use contours.drawlabels = TRUE add labels on contour lines.
contours.lty	the line type for contour lines.
contours.lwd	the width for contour lines.
contours.levels	the levels for contour lines. When exp=T, the levels will also be used in an exponential scale.
contours.labcex	cex for contour labelling. This is an absolute size, not a multiple of par("cex")
arrow	use arrow=T to add a North arrow to the map.
axes	use axes=T to add axes to the map (useful for chemical mixture isoboles).
ptsize	the size of the points used to fill in colors on the map. Increase to remove white space inside the map or decrease to remove color outside the map boundaries. NOTE: white space can also be eliminated by increasing the grid size in predgrid, which is often preferable as it results in a higher resolution map.

alpha	the nominal pointwise type I error rate; only used when contours="permrank".
mai	the margins of the plot. Details see par.
legend.name	the name of displayed for the legend bar.
legend.cex	a numerical value giving the amount by which legend text should be magnified relative to the default.
...	other arguments to be passed to colormap function.

Value

Produces an image map showing crude or adjusted linear predictors (or odds ratios). If the base map is in readShapePoly format a scale bar is included. The scale bar assumes that the X and Y coordinates are provided in meters.

Warning

Note that the contour lines use a pointwise nominal type I error rate of alpha; the chance of a type I error occurring for **at least** one point on the map is much higher, typically approaching 100% at alpha=0.05, because a spatial prediction grid generally contains many points.

Author(s)

Scott Bartell, Lu Bai, Robin Bliss, and Veronica Vieira

Send bug reports to <sbartell@uci.edu>.

See Also

trimdata, predgrid, optspan, modgam, colormap.

Examples

```
data(MAmap)
data(MAdata)
# Create a grid on the base map
MAgrid <- predgrid(MAdata, map=MAmap)
# fit crude GAM model to the MA data using span size of 50%
fit1 <- modgam(data=MAdata, rgrid=MAgrid, m="crude", sp=0.5)
# Plot a map showing crude odds ratios
plot(fit1, MAmap)

#### A detailed example including map projections and data trimming
# NOTE: this example requires the maps and mapproj packages
# Convert base map and beer tweet data locations to US Albers projection
# projected coords yield better distance estimates than (lat,long)
if(require(maps) & require(mapproj)) {
  USmap <- map("state",projection="albers",parameters=c(29.5,45.5),
               plot=FALSE,fill=TRUE,col="transparent")
  data(beertweets)
  case <- beertweets$beer
  # Reuse last map projection to convert data coordinates
  XY <- mapproject(beertweets$longitude,beertweets$latitude)[1:2]
```

```
  beerproj <- data.frame(case, XY[1], XY[2])
  # Generate grid on the US map, trimmed to range of beer data
  USgrid <- predgrid(beerproj, map=USmap)
  # Fit unadjusted model--geolocation only
  fit2 <- modgam(data=beerproj, rgrid=USgrid, m="unadjusted", sp=0.05)
  dev.new(width=7,height=5)
  plot(fit2, USmap)
  title(main="Beer Tweet Odds Ratios")
}
```

---

predgrid *Create a Grid and Clip It to a Map and Data Bounds*

Description

Creates a data frame containing a rectangular grid of points to cover the range of X and Y coordinates provided in a data set, and trims the grid so that the points do not extend beyond the boundaries shown on a map. Users can omit a data set to create a grid covering the whole map, or omit a map to create a grid covering the whole data set.

Usage

```
predgrid(dataXY = NULL, map = NULL, nrow = 100, ncol = 100, X = NULL, Y = NULL)
```

Arguments

dataXY	A data frame with X and Y coordinates. If the data frame has more than 2 columns, the X and Y coordinates must be in the 2nd and 3rd columns, respectively (the format for the data argument in the modgam function). If the data frame has only 2 columns, the X coordinates must be in the 1st column and Y coordinates in the 2nd column. If no data are provided for this argument or for the X and Y arguments, the grid will cover the entire map.
map	a map for clipping the grid, provided in a recognized format. Supported map classes include "map" (the format produced by the maps package), "sf" (the ISO standard format supported by the sf package), "Raster" (used by the raster package), or "Spatial" (a legacy format used by the maptools package). To load a map from an ESRI shapefile into R, use the st_read function in the sf library, the readOGR function in the rgdal package, or the readShapePoly function in the maptools package. If map = NULL (default), a rectangular grid is produced covering the range of geolocations within dataXY. For Raster format maps, the grid will be rectangular, rather than clipped to the raster boundaries. If no map is provided the grid will cover the whole data set.
nrow	the number of rows in the grid (default=100).
ncol	the number of columns in the grid (default=100).

X — a vector of X coordinates, supplied instead of (or in addition to) dataXY. If both X and dataXY are provided, X will be used instead of the corresponding column of dataXY.

Y — a vector of Y coordinates, supplied instead of (or in addition to) dataXY. If both Y and dataXY are provided, Y will be used instead of the corresponding column of dataXY.

Details

predgrid creates a grid of dimensions nrow*ncol using the range of X and Y coordinates (e.g., longitude and latitude) in the data frame supplied as dataXY. If the map argument is used, the function trimdata is automatically used to clip the grid. Users should be sure to use the same projection for the map and the data; putting both on the same plot can help reveal differing projections. If the map centroid is not in the range of the data a warning message is printed; this might indicate differing projections but can occur naturally when the data were not sampled from the entire extent of the map, or when map boundaries are concave.

Value

A data frame with X and Y coordinates in the first two columns. The column names from dataXY are used for the output. If the columns of dataXY are unnamed then the names "X" and "Y" are assigned to the data frame.

Author(s)

Veronica Vieira, Scott Bartell, and Robin Bliss

Send bug reports to <sbartell@uci.edu>.

See Also

trimdata, optspan, modgam, colormap.

Examples

```
# define a rectangular 100x100 grid covering the MA data
data(MAdata)
gamgrid <- predgrid(MAdata)
# plot the grid points
plot(gamgrid$Xcoord, gamgrid$Ycoord, cex=0.1, col="red")
# and the data locations
points(MAdata$Xcoord,MAdata$Ycoord)

# But that grid extends beyond the state boundaries and into the ocean!
# Better to also clip the grid to a map of MA using the following code:

# Clip a 50x50 grid covering the MA data to a map of MA
data(MAmap)
gamgrid2 <- predgrid(MAdata, map=MAmap, nrow=50, ncol=50)
# plot the MA map and grid points
plot(MAmap)
```

```
points(gamgrid2, cex=0.1, col="red")
```

`predict.gamcox` *Prediction Method for* `gamcox` *Fits*

Description

Obtains spatial effects predictions and optionally estimates standard errors and confidence intervals of those predictions from a fitted Cox proportional hazard additive model object.

Usage

```
## S3 method for class "gamcox"
## S3 method for class 'gamcox'
 predict(object, newdata = object$data, se.fit = FALSE, type=c("spatial","all"),
        reference = "median", level = 0.05, verbose=FALSE,...)
```

Arguments

`object`	a fitted `gamcox` object.
`newdata`	a data frame containing the values at which predictions are required. This argument can be missing, in which case predictions are made at the same values used to compute the object. A comprehensive effect (hazard ratio) of the covariates included in `newdata` will be predicted.
`se.fit`	if `TRUE`, pointwise standard errors and confidence intervals are computed along with the predictions.
`type`	use `type="spatial"`(default) to estimate spatial effect, and use `type="all"` to estimate the effect of all covariates included in the model.
`reference`	the type of reference for the estimated effect. If `reference = "median"`, the output will be the estimated hazard ratio (or log-ratio) compared to the median effect. If `reference = "mean"`,the output will be the estimated hazard ratio (or log-ratio) compared to the mean effect.If `reference` is a data frame indicating a specific geolocation, the out put will be the estimated hazard ratio (or log ratio) compared to the hazard of the geolocation specified by `reference`.
`level`	the confidence level for condifence bands.
`verbose`	a logical value indicating whether to print filling values for newdata. The default is `verbose = FALSE`.
`...`	extra arguments for S3 generic, ignored by `predict.gamcox`.

Value

`pred`	the estimated log hazards ratio compared to the effect specified by `reference`.
`se`	the standard errors along with the predictions.
`conf.low`	the lower bounds for pointwise (1-`level`) confidence intervals.
`conf.high`	the higher bounds for pointwise (1-`level`) confidence intervals.

Author(s)

Lu Bai

Send bug reports to <sbartell@uci.edu>.

References

Hastie TJ, Tibshirani RJ. Generalized Additive Models. (Chapman & Hall/CRC Monographs on Statistics & Applied Probability, Boca Raton, Florida, 1990).

See Also

modgam, predict.gamcox.

Examples

```
data(CAdata)
data(CAmap)
fit <- gamcox(Surv(time,event)~AGE + factor(INS) + lo(X,Y),data=CAdata,
       span=0.2,loess.trace="approximate")
fit
pred1 = predict(fit)
colormap(list(fit=pred1$pred,grid=data.frame(X=CAdata$X,Y=CAdata$Y)),map=CAmap,
         border.gray=0.5)

data(CAgrid)
pred2 = predict(fit,CAgrid[,c("X","Y")])
colormap(list(fit=pred2$pred,grid=data.frame(X=CAgrid$X,Y=CAgrid$Y)),map=CAmap,
        border.gray=0.5, legend.name="log hazard ratio")

## Circle significant areas based on the confidence intervals specified by conf.low and conf.high
pred3 = predict(fit,CAgrid[,c("X","Y")],se.fit=TRUE)
colormap(list(fit=pred3$pred,conf.low = pred3$conf.low, conf.high = pred3$conf.high,
          grid=data.frame(X=CAgrid$X,Y=CAgrid$Y)),map=CAmap,border.gray = 0.7,
          contours = "interval",legend.name="log hazard ratio")
```

---

print.gamcox *Print the gamcox Object*

Description

This function displays information about a gamcox object produced by gamcox.

Usage

```
## S3 method for class 'gamcox'
 print(x,...)
```

Arguments

x	a gamcox object.
...	.

Author(s)

Lu Bai

Send bug reports to <sbartell@uci.edu>.

See Also

modgam gamcox, predict.gamcox.

Examples

```
data(CAdata)
fit <- gamcox(Surv(time,event)~lo(X,Y)+AGE+INS,data=CAdata, span = 0.2)
print(fit)
```

---

print.modgam *Print the modgam Object*

Description

This function displays information about a modgam object produced by modgam.

Usage

```
## S3 method for class 'modgam'
 print(x, ...)
```

Arguments

x	a modgam object.
...	extra arguments for S3 generic, ignored by print.modgam.

Author(s)

Lu Bai

Send bug reports to <sbartell@uci.edu>.

See Also

modgam gamcox, predict.gamcox.

residuals.gamcox *Residuals of the gamcox Object*

Description

This function provides residuals of a gamcox object produced by gamcox.

Usage

```
## S3 method for class 'gamcox'
 residuals(object,...)
```

Arguments

object	a gamcox object.
...	.

Author(s)

Lu Bai

Send bug reports to <sbartell@uci.edu>.

See Also

modgam gamcox, predict.gamcox.

Examples

```
data(CAdata)
fit <- gamcox(Surv(time,event)~lo(X,Y)+AGE+INS,data=CAdata, span = 0.2)
residuals(fit)
```

residuals.modgam *Residuals of the modgam Object*

Description

This function provides residuals of a modgam object produced by modgam.

Usage

```
## S3 method for class 'modgam'
 residuals(object, ...)
```

Arguments

object	a modgam object.
...	extra arguments for S3 generic, ignored by residuals.modgam.

Author(s)

Lu Bai

Send bug reports to <sbartell@uci.edu>.

See Also

modgam gamcox, predict.gamcox.

---

sampcont *Unmatched Control Sampling*

Description

Take all cases and a random sample of controls from a data frame. Simple random sampling and spatially stratified random sampling are available. For spatially statified random sampling, strata can be defined by region, or by region and additional stratification variables (see Tang et al., 2023 for examples and simulation comparisons). If no specific regions are specified with stratified sampling, the function will create a regular grid for spactially stratified sampling.

Usage

```
sampcont(rdata, type = "stratified", casecol=1, Xcol=2, Ycol=3, regions = NULL,
  addstrat = NULL, times = NULL, n = 1, nrow = 100, ncol = 100)
```

Arguments

rdata	a data frame with case status in the casecol column (by default, the 1st column), and the geocoordinates (e.g., X and Y) in the Xcol and Ycol columns (by default, the 2nd and 3rd columns). Additional columns are not used in the sampling scheme but are retained in the sampled data frame.
casecol	the column number in rdata for the case status (coded with 0 for controls, 1 for cases).
Xcol	the column number in rdata for the X geocoordinate.
Ycol	the column number in rdata for the Y geocoordinate.
type	"stratified" (default) or "simple". If "simple" then a simple random sample of n controls (rows of rdata with outcome=0) is obtained. If "stratified" then a stratified random sample of controls is obtained, with up to n controls per stratum. Sampling strata are defined by the regions and times arguments. All cases (rows with outcome=1) are taken for the sample regardless of the value supplied for type.

regions	a vector of length equal to the number of rows in rdata, used to construct sampling strata. Only used if type = "stratified". If regions = NULL then the function will define regions as a vector of specific grid cells on a regular grid with nrow rows and ncol columns. If times = NULL then the nonempty regions are used as the sampling strata. If times is a vector, then the sampling strata are all nonempty combinations of regions and times.
addstrat	a vector of length equal to the number of rows in rdata, used along with regions to construct sampling strata. If addstrat = NULL then the sampling strata are defined only by the regions argument. If addstrat is a vector, then the sampling strata are all nonempty combinations of regions and addstrat. Continuous variables should generally be binned before being passed through this argument, as there are no efficiency gains if each value in addstrat is unique.
times	included for backward compatibility; now replaced by the addstrat argument which serves the same purpose.
n	the number of controls to sample from the eligible controls in each stratum. All available controls will be taken for strata with fewer than n eligible controls.
nrow	the number of rows used to create a regular grid for sampling regions. Only used when regions = NULL.
ncol	the number of columns used to create a regular grid for sampling regions. Only used when regions = NULL.

Value

rdata	a data frame with all cases and a random sample of controls.
w	the inverse probability weights for the rows in rdata. Important to include as weights in subsequent analyses.
ncont	the total number of controls in the sample.
type	statified or simple sampling, as specified by the same argument described above.
gridsize	a vector with the numbers of rows and columns for the stratified sampling grid.
grid	the stratified sampling grid in PolySet format.

Author(s)

Scott M. Bartell and Ian W. Tang <sbartell@uci.edu>.

References

Tang IW, Bartell SM, Vieira VM. Unmatched Spatially Stratified Controls: A simulation study examining efficiency and precision using spatially-diverse controls and generalized additive models. Spatial and Spatio-temporal Epidemiology 2023, 45:100584. doi:10.1016/j.sste.2023.100584

See Also

modgam

Examples

```
#### load beertweets data, which has 719 cases and 9281 controls
data(beertweets)
# take a simple random sample of 1000 controls
samp1 <- sampcont(beertweets, type="simple", n=1000)

# take a stratified random sample of controls on a 80x50 grid
samp2 <- NULL

samp2 <- sampcont(beertweets, nrow=80, ncol=50)

# Compare locations for the two sampling designs (cases in red)
par(mfrow=c(2,1), mar=c(0,3,4,3))
plot(samp1$rdata$longitude, samp1$rdata$latitude, col=3-samp1$rdata$beer,
 cex=0.5, type="p", axes=FALSE, ann=FALSE)
# Show US base map if maps package is available
mapUS <- require(maps)
if (mapUS) map("state", add=TRUE)
title("Simple Random Sample, 1000 Controls")

if (!is.null(samp2)) {
 plot(samp2$rdata$longitude, samp2$rdata$latitude,
  col=3-samp2$rdata$beer, cex=0.5, type="p", axes=FALSE,
  ann=FALSE)
 if (mapUS) map("state", add=TRUE)
 title(paste("Spatially Stratified Sample,",samp2$ncont,"Controls"))
 }

par(mfrow=c(1,1))

## Note that weights are needed in statistical analyses
# Prevalence of cases in sample--not in source data
mean(samp1$rdata$beer)
# Estimated prevalence of cases in source data
weighted.mean(samp1$rdata$beer, w=samp1$w)
## Do beer tweet odds differ below the 36.5 degree parallel?
# Using full data
glm(beer~I(latitude<36.5), family=binomial, data=beertweets)
# Stratified sample requires sampling weights
if (!is.null(samp2)) glm(beer~I(latitude<36.5), family=binomial,
 data=samp2$rdata, weights=samp2$w)
```

---

summary.gamcox *Summarize the* gamcox *Object*

Description

This function summarizes information about a gamcox object produced by gamcox.

Usage

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

Arguments

object	a gamcox object.
...	.

Author(s)

Lu Bai

Send bug reports to <sbartell@uci.edu>.

See Also

modgam gamcox, predict.gamcox.

Examples

```
data(CAdata)
fit <- gamcox(Surv(time,event)~lo(X,Y)+AGE+INS,data=CAdata, span = 0.2)
summary(fit)
```

---

summary.modgam *Summarize the modgam Object*

Description

This function summarizes information about a modgam object produced by modgam.

Usage

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

Arguments

object	a modgam object.
...	extra arguments for S3 generic, ignored by summary.modgam.

Author(s)

Lu Bai

Send bug reports to <sbartell@uci.edu>.

See Also

modgam gamcox, predict.gamcox.

---

toformula *Build a Formula Based on Data for* modgam *Function*

Description

Based on the arguments in modgam, build a formula used for model fitting.

Usage

```
toformula(formula, data,m="adjusted",surv = FALSE, span=0.5,degree=2,
          smooth = TRUE,offset='')
```

Arguments

formula	a user specified formula expression (for modgam).
data	a well structured data frame with the two geolocation parameters in the first and second columns. data is ingnored if formula is specified.
m	use m="adjusted" to specify a model including adjusted confounders. Use m="unadjusted" to specify a unadjusted model.
surv	surv=TRUE specifies a model for censored surival dataset.
span, degree	smoothing parameters used for smoothing functions (see details in modgam) if smooth=TRUE.
smooth	smooth=TRUE specifies a smooth term in the model.
offset	the name for offset.

Value

The function returns the formula for the model.

Author(s)

Lu Bai

Send bug reports to <sbartell@uci.edu>.

See Also

modgam, .

Examples

```
data(CAdata)
toformula(data=CAdata, surv=TRUE)
```

trimdata *Trim a Data Set To Map Boundaries*

Description

Takes a subset of a data frame: returns rows with geolocations inside the boundaries determined by a map. If rectangle=FALSE, strict map boundaries are used. If rectangle=TRUE, a rectangular boundary is determined based on the range of X and Y coordinates for the map, along with an optional buffer.

Usage

```
trimdata(rdata, map, Xcol=2, Ycol=3, rectangle=F, buffer=0.05)
```

Arguments

rdata	the data set (required). If rdata has only 2 columns then they are assumed to be the X and Y coordinates, in that order. If rdata has more than 2 columns then identify the positions of the X and Y coordinates with Xcol and Ycol, respectively.
map	a map for clipping the grid, provided in a recognized format. Supported map classes include "map" (the format produced by the maps package), "sf" (the ISO standard format for maps, supported by the sf package), "Raster" (used by the raster package), or "Spatial" (a legacy format used by the maptools package). To load a map from an ESRI shapefile into R, use the st_read function in the sf library, the readOGR function in the rgdal package, or the readShapePoly function in the maptools package (soon to be deprecated). For Raster format maps, the grid will be rectangular, rather than clipped to the raster boundaries.
Xcol	the column number of rdata for the X coordinates of geolocation. Only used if rdata has >2 columns.
Ycol	the column number of rdata for the Y coordinates of geolocation. Only used if rdata has >2 columns.
rectangle	if rectangle=FALSE (default), only data with geolocations strictly within the map boundaries are retained. If rectangle=TRUE, only data with geolocations within a rectangular boundary are retained. The rectangular boundary coordinates are determined using the ranges of X and Y coordinates for the map, along with an optional buffer.
buffer	a fraction of the map range to add to either side of the rectangular boundary before trimming the data (default 5%). The buffer argument is ignored if rectangle=FALSE. If a vector of length 2 is supplied, the first value as the buffer fraction for the X coordinate range and the second value is used as the buffer fraction for the Y coordinate range.

Details

Functions from the sf package are used to convert map formats and clip the grid to points inside the map boundaries. Be sure to use `fill=T` when using the `map` function to produce maps for `trimdata`: maps produced using the default `fill=F` argument do not work properly with this function! If the map bounding box centroid is not in the range of the data, a warning message is printed; this might indicate differing projections, but can occur naturally when the data were not sampled from the entire extent of the map, or when map boundaries are concave.

Value

A subset of the rdata data frame containing only those rows with geolocations within the specified boundaries.

Author(s)

Veronica Vieira, Scott Bartell, and Robin Bliss

Send bug reports to <sbartell@uci.edu>.

See Also

predgrid, optspan, modgam, colormap.

Examples

```
# The following examples use the "maps" package, which includes its own maps
if (require(maps)) {
  data(beertweets)

  ### Trim beer tweet data to US base map, and plot them
  basemap1 <- map("usa", fill=TRUE, col="transparent")
  dUS <- trimdata(beertweets, basemap1)
  # Plot tweet locations (beer tweets in red)
  points(dUS$longitude, dUS$latitude, col=dUS$beer+1, cex=0.5)
  dev.off()  # clear map settings

  ### Trim beer tweet data to Texas base map, and plot them
  basemap2 <- map("state", regions="texas", fill=TRUE, col="transparent")
  dTX <- trimdata(beertweets, basemap2)
  # Plot tweet locations (beer tweets in red)
  points(dTX$longitude, dTX$latitude, col=dTX$beer+1, cex=0.5)
}

## Not run:
# This additional example (not run automatically by the example function)
# uses the sf and RCurl packages to read in an external ESRI shapefile for Maine
shpfileloc <- "https://www2.census.gov/geo/tiger/GENZ2025/shp/cb_2025_23_cousub_500k.zip"
# if (require(sf) && require(RCurl) && url.exists(shpfileloc)) {
  # download example shapefile zip from github, and unzip
  zippath <- paste(tempdir(),"cb_2025_23_cousub_500k.zip",sep="/")
  download.file(shpfileloc,
```

```
            destfile = zippath, mode='wb')
  unzip(zippath, exdir = tempdir())

  # read shapefile into sf format
  shppath <- paste(tempdir(),"cb_2025_23_cousub_500k.shp",sep="/")
  basemap0 <- st_read(shppath)

  # Create example data by randomly sampling within bounding box
  rs <- st_bbox(basemap0)   # get ranges of X and Y
  MEdata <- data.frame(X=runif(300,rs[1],rs[3]), Y=runif(300,rs[2],rs[4]))
  plot(basemap0["NAME"], reset=FALSE)
  plot(st_as_sf(MEdata,coords=1:2), add=TRUE)  # plot data in black

  # trim data to basemap, and plot trimmed data with red X's
  dME <- trimdata(MEdata, basemap0)
  plot(st_as_sf(dME,coords=1:2), col="red", pch="X", add=TRUE)

  dev.off()  # clear map settings
#  }

## End(Not run)
```

Index

∗ **datasets**
beertweets, 5
CAdata, 6
CAgrid, 6
CAmap, 7
MAdata, 17
MAmap, 18
MapGAM-package, 2
∗ **hplot**
colormap, 9
MapGAM-package, 2
plot.modgam, 28
∗ **misc**
AIC.modgam, 4
coef.gamcox, 8
coef.modgam, 8
colormap, 9
dls, 11
formula.gamcox, 13
formula.modgam, 14
gamcox, 14
MapGAM-package, 2
modgam, 19
mypredict.gam, 25
optspan, 27
plot.modgam, 28
predgrid, 31
predict.gamcox, 33
print.gamcox, 34
print.modgam, 35
residuals.gamcox, 36
residuals.modgam, 36
sampcont, 37
summary.gamcox, 39
summary.modgam, 40
toformula, 41
trimdata, 42
∗ **optimize**
optspan, 27
∗ **package**
MapGAM-package, 2
∗ **smooth**
AIC.modgam, 4
coef.gamcox, 8
coef.modgam, 8
colormap, 9
dls, 11
formula.gamcox, 13
formula.modgam, 14
gamcox, 14
MapGAM-package, 2
modgam, 19
mypredict.gam, 25
optspan, 27
plot.modgam, 28
predict.gamcox, 33
print.gamcox, 34
print.modgam, 35
residuals.gamcox, 36
residuals.modgam, 36
summary.gamcox, 39
summary.modgam, 40
toformula, 41
∗ **version**
MapGAM.version, 18

AIC.modgam, 4

beertweets, 5

CAdata, 6
CAgrid, 6
CAmap, *6*, 7
coef.gamcox, 8
coef.modgam, 8
colormap, *4*, 9, *23*, *28*, *30*, *32*, *43*

dls, 11

family, *20*, *27*

formula.gamcox, 13
formula.modgam, 14

gam, *21–23*
gamcox, *4*, *8, 9*, *11–14*, 14, *21, 22*, *34–37*, *39–41*

MAdata, 17
MAmap, *17*, 18
MapGAM *(*MapGAM-package*)*, 2
MapGAM-package, 2
MapGAM.version, 18
modgam, *4*, *8, 9*, *13, 14*, *16*, 19, *26–28*, *30*, *32*, *34–38*, *40, 41*, *43*
mypredict.gam, *22*, 25

optspan, *4*, *20*, *23*, 27, *30*, *32*, *43*

par, *10*, *30*
plot, *11*
plot.modgam, *4*, *11*, *23*, 28
predgrid, *4*, *11*, *23*, *28*, *30*, 31, *43*
predict.gamcox, *4*, *8, 9*, *12–14*, *16*, *22*, *26*, 33, *34–37*, *40, 41*
print.gamcox, 34
print.modgam, 35

residuals.gamcox, 36
residuals.modgam, 36

sampcont, *4*, 37
summary.gamcox, 39
summary.modgam, 40

toformula, 41
trimdata, *4*, *11*, *30*, *32*, 42