Package 'GeoAdjust'

March 20, 2023

Type Package

Title Accounting for Random Displacements of True GPS Coordinates of Data

Version 1.0.1

Description The purpose is to account for the random displacements (jittering) of true survey household cluster center coordinates in geostatistical analyses of Demographic and Health Surveys program (DHS) data. Adjustment for jittering can be implemented either in the spatial random effect, or in the raster/distance based covariates, or in both. Detailed information about the methods behind the package functionality can be found in two preprints.
Umut Altay, John Paige, Andrea Riebler, Geir-Arne Fuglstad (2022) <arXiv:2202.11035v2>.
Umut Altay, John Paige, Andrea Riebler, Geir-Arne Fuglstad (2022) <arXiv:2211.07442v1>.

Encoding UTF-8

RoxygenNote 7.2.2

Suggests INLA, knitr, rmarkdown, testthat (>= 3.0.0), rgdal, maptools

Config/testthat/edition 3

Depends R (>= 3.5)

LinkingTo TMB, RcppEigen

License GPL (>= 2)

Imports raster, sp, stats, rgeos, SUMMER, Matrix, ggplot2, fields, geosphere, TMB

Additional_repositories https://inla.r-inla-download.org/R/stable/

NeedsCompilation yes

Author Umut Altay [cre, aut],
John Paige [aut],
Geir-Arne Fuglstad [aut],
Andrea Riebler [aut]

Maintainer Umut Altay <altayumut.ua@gmail.com>

Repository CRAN

Date/Publication 2023-03-20 19:30:07 UTC

R topics documented:

convertDegToKM 2
convertDegToMollweide 3
convertKMToDeg 3
covMatern 4
dummy 5
estimateModel 5
gridCountry 6
meshCountry 7
plotPred 8
predRes 9
prepareInput 10
print.res 11

Index **13**

`convertDegToKM` *Converts a set of coordinates in degrees into a new set of coordinates in The Universal Transverse Mercator (UTM) zone:37 coordinate system*

Description

Converts a set of coordinates in degrees into a new set of coordinates in The Universal Transverse Mercator (UTM) zone:37 coordinate system

Usage

```
convertDegToKM(loc)
```

Arguments

`loc`	A two column matrix of coordinates (The first column is longitude and the second column is latitude).

Value

A two column matrix of coordinates in The Universal Transverse Mercator (UTM) zone:37 (https://www.usgs.gov/faqs/what-does-term-utm-mean- utm-better-or-more-accurate-latitudelongitude).

Examples

```
path1 <- system.file("extdata", "geoData.rda", package = "GeoAdjust")
load(path1)
locDegree <- cbind(surveyData$long, surveyData$lat)
locKM <- convertDegToKM(loc = locDegree)
head(locKM)
```

convertDegToMollweide *Converts a set of coordinates in degrees into a new set of coordinates in Mollweide coordinate system.*

Description

Converts a set of coordinates in degrees into a new set of coordinates in Mollweide coordinate system.

Usage

```
convertDegToMollweide(loc)
```

Arguments

loc	A two column matrix of coordinates (The first column is longitude and the second column is latitude).

Value

A two column matrix of coordinates in Mollweide (https://pubs.usgs.gov/pp/1395/report.pdf) coordinate system.

Examples

```
path1 <- system.file("extdata", "geoData.rda", package = "GeoAdjust")
load(path1)
locDegree <- cbind(surveyData$long, surveyData$lat)
locMoll <- convertDegToMollweide(loc = locDegree)
head(locMoll)
```

convertKMToDeg *Converts a set of coordinates in The Universal Transverse Mercator (UTM) zone:37 coordinate system into a new set of coordinates in degrees.*

Description

Converts a set of coordinates in The Universal Transverse Mercator (UTM) zone:37 coordinate system into a new set of coordinates in degrees.

Usage

```
convertKMToDeg(loc)
```

Arguments

loc	A two column matrix of coordinates in The Universal Transverse Mercator (UTM) zone:37 coordinate system (https://www.usgs.gov/faqs/what-does-term-utm-mean-utm-better-or-more-accurate-latitudelongitude).

Value

A two column matrix of coordinates in degrees (the first column is longitude and the second column is latitude).

Examples

```
path1 <- system.file("extdata", "geoData.rda", package = "GeoAdjust")
load(path1)
locKM <- cbind(surveyData$east, surveyData$north)
locDegree <- convertKMToDeg(loc = locKM)
head(locDegree)
```

---

covMatern *Creates a Matern covariance matrix.*

Description

Creates a Matern covariance matrix.

Usage

```
covMatern(dMat = NULL, range = NULL, stdDev = NULL)
```

Arguments

dMat	A distance matrix between the locations.
range	Spatial range in kilometers.
stdDev	The marginal variance.

Value

Matern covariance matrix.

Examples

```
if(requireNamespace("INLA")){
path1 <- system.file("extdata", "geoData.rda", package = "GeoAdjust")
load(path1)
loc <- cbind(surveyData$east, surveyData$north)
space.range <- 114
space.sigma <- 1
covMat <- covMatern(dMat = as.matrix(dist(loc)),
range = space.range, stdDev = space.sigma)
}
```

dummy *Roxygen commands*

Description

Roxygen commands

Usage

```
dummy()
```

estimateModel *Estimates model parameters*

Description

Estimates model parameters

Usage

```
estimateModel(
  data = NULL,
  nNodes = NULL,
  options = NULL,
  priors = NULL,
  n.sims = NULL
)
```

Arguments

data	A data input list that is created by prepareInput() function.
nNodes	number of mesh nodes.
options	A list containing two components, namely, random and covariates, representing the spatial random field and covariates. Values of 1 and 0 turn the accounting for jittering in these components on and off.
priors	A list of six components. Beta is a vector of two elements and passes the parameters of the Gaussian prior that will be assigned to the covariates (including the intercept). The first element of it is the mean and the second one is the standard deviation of Gaussian prior. Range is a value representing the median range in kilometers, which will be used for constructing the PC (Penalized-complexity) priors. USpatial and alphaSpatial are the threshold and probability of crossing the threshold for the variance prior. UNugget and alphaNug are the threshold and probability of crossing the threshold for the prior on the nugget standard deviation. UNugget and alphaNug should be included in the priors argument, but they will only be used when the likelihood is Gaussian.
n.sims	number of samples to be drawn for each model parameter

Value

Model estimation results of class called res. The output consists of four elements: A data frame containing the estimated model parameters and the corresponding 95 The optimized core model object from autodifferentiation of TMB, A matrix containing the sampled coefficient effect sizes and the random effect coefficients, A character string indicating the likelihood type in the model.

Examples

```
path1 <- system.file("extdata", "exampleInputData.rda", package = "GeoAdjust")
path2 <- system.file("extdata", "exampleMesh.rda", package = "GeoAdjust")
load(path1)
load(path2)
nNodes = exampleMesh[['n']]
results <- estimateModel(data = exampleInputData, nNodes = nNodes,
options = list(random = 1, covariates = 1), priors = list(beta = c(0,1),
range = 114, USpatial = 1, alphaSpatial = 0.05, UNugget = 1, alphaNug = 0.05), n.sims = 1000)
```

---

gridCountry *Creates a grid of locations within the bounding box of the national borders of a country of interest.*

Description

Creates a grid of locations within the bounding box of the national borders of a country of interest.

Usage

```
gridCountry(admin0 = NULL, res = NULL)
```

Arguments

admin0	A SpatialPolygonsDataFrame object representing the national (admin0) level borders of the country.
res	A value representing the resolution in kilometers.

Value

A list. The first element of the list, predRast, is the prediction raster. The second element of the list, loc.pred, is a data frame containing the grid of coordinates of the cell centers (both in degrees and in kilometers) of the prediction raster.

Examples

```
path1 <- system.file("extdata", "geoData.rda", package = "GeoAdjust")
load(path1)
grid <- gridCountry(admin0 = adm0, res = 5)
```

`meshCountry` *Creates a constrained refined Delaunay triangulation mesh based on the country borders.*

Description

Creates a constrained refined Delaunay triangulation mesh based on the country borders.

Usage

```
meshCountry(admin0 = NULL, max.edge = NULL, offset = NULL)
```

Arguments

`admin0`	A SpatialPolygonsDataFrame representing the country borders, in UTM:zone 37 coordinate system.
`max.edge`	A vector of two values. The first and the second elements of the vector represent the largest allowed triangle lengths for the inner and outer mesh, respectively.
`offset`	A value representing the extension distance for the inla.mesh.2d object

Value

A constrained refined Delaunay triangulation mesh created based on the country borders.

Examples

```
if(requireNamespace("INLA")){
path1 <- system.file("extdata", "adm0UTM37.rda", package = "GeoAdjust")
load(path1)
mesh.s <- meshCountry(admin0 = adm0UTM37, max.edge = c(25, 50), offset = -.08)
}
```

plotPred *Plots the predictions and the corresponding uncertainty (coefficient of variation)*

Description

Plots the predictions and the corresponding uncertainty (coefficient of variation)

Usage

```
plotPred(
  pred = NULL,
  predRaster = NULL,
  admin0 = NULL,
  admin1 = NULL,
  admin2 = NULL,
  rmPoly = NULL,
  locObs = NULL
)
```

Arguments

pred	A matrix that is the output of predRes() function.
predRaster	The prediction raster that is constructed by the gridCountry() function.
admin0	A SpatialPolygonsDataFrame representing the national level (admin0) borders of the country.
admin1	A SpatialPolygonsDataFrame representing the first level (admin1) subnational borders of the country.
admin2	A SpatialPolygonsDataFrame representing the second level (admin2) subnational borders of the country.
rmPoly	A number referring to the ID number of the admin2 level polygon that needs to be left uncolored. It can be set to NULL as well.
locObs	A data frame containing the coordinates of the observation points (DHS locations) in kilometers.

Value

A list of two ggplot objects. One of them (ggPred) shows the median predictions and the other one (ggUncertainty) shows the corresponding coefficient of variations across the country, respectively.

Examples

```
path1 <- system.file("extdata", "examplePredictionResults.rda", package = "GeoAdjust")
path2 <- system.file("extdata", "exampleGrid.rda", package = "GeoAdjust")
path3 <- system.file("extdata", "geoData.rda", package = "GeoAdjust")
load(path1)
```

```
load(path2)
load(path3)
plots = plotPred(pred = examplePredictionResults,
predRaster = exampleGrid[["predRast"]], admin0 = adm0,
admin1 = adm1, admin2 = NULL, rmPoly = NULL,
locObs = data.frame(East = surveyData$east, North = surveyData$north))
```

---

predRes *Predicts model outcomes at new locations.*

Description

Predicts model outcomes at new locations.

Usage

```
predRes(
  obj = NULL,
  predCoords = NULL,
  draws = NULL,
  nCov = NULL,
  covariateData = NULL,
  mesh.s = NULL,
  flag = NULL
)
```

Arguments

obj	The optimized core model object returned by estimateModel() function.
predCoords	A matrix containing the coordinates of the prediction locations in kilometers (UTM zone:37).
draws	A matrix containing 10.000 sampled values for each covariate effect size and 10.000 sampled values of random effect coefficients for each mesh node. It is one of the elements of the returning output list of estimateModel() function.
nCov	A value showing the number of covariates (including the intercept).
covariateData	A list containing the covariate rasters.
mesh.s	A mesh created based on the country borders.
flag	A value indicating the type of the likelihood that will be used. Pass 0 for Gaussian, 1 for binomial and 2 for Poisson likelihoods.

Value

A matrix containing the mean, median,standard deviation and the lower and the upper bounds of 95

Examples

```
if(requireNamespace("INLA")){
path1 <- system.file("extdata", "exampleInputData.rda", package = "GeoAdjust")
path2 <- system.file("extdata", "exampleMesh.rda", package = "GeoAdjust")
path3 <- system.file("extdata", "exampleGrid.rda", package = "GeoAdjust")
load(path1)
load(path2)
load(path3)
nNodes = exampleMesh[['n']]
results <- estimateModel(data = exampleInputData, nNodes = nNodes,
options = list(random = 1, covariates = 1), priors = list(beta = c(0,1),
range = 114, USpatial = 1, alphaSpatial = 0.05, UNugget = 1, alphaNug = 0.05), n.sims = 1000)
pred = predRes(obj = results[["obj"]],
predCoords = cbind(exampleGrid[["loc.pred"]]["east"],
exampleGrid[["loc.pred"]]["north"]),
draws = results[["draws"]], nCov = 1,
mesh.s = exampleMesh, covariateData = NULL, flag = 1)
}
```

`prepareInput` *Prepares input data list for the model estimation with estimateModel() function.*

Description

Prepares input data list for the model estimation with estimateModel() function.

Usage

```
prepareInput(
  response = NULL,
  locObs = NULL,
  likelihood,
  jScale = NULL,
  urban = NULL,
  mesh.s = NULL,
  adminMap = NULL,
  nSubAPerPoint = 10,
  nSubRPerPoint = 10,
  covariateData = NULL
)
```

Arguments

`response`	A list containing the number of trials (ns) and number of successes (ys) for the binomial response, response values (ys) for the Gaussian response or the Poisson counts for the Poisson response.

locObs	A matrix containing the coordinates of DHS survey cluster centers in kilometers.
likelihood	A value indicating which likelihood should be used (0, 1 or 2 for Gaussian, binomial or Poisson, respectively).
jScale	Jittering scale, where 1 represents the default DHS jittering scheme.
urban	A vector containing the urbanization classification of the administrative area that each cluster center is initially located within (U for urban and R for rural).
mesh.s	A triangulation mesh.
adminMap	A SpatialPolygonsDataFrame object containing the borders of the administrative area level, which was respected while the cluster centers were initially being jittered (can be obtained from https://gadm.org).
nSubAPerPoint	A value representing the number of unique sub-integration point angles per integration point.
nSubRPerPoint	A value representing the number of unique sub-integration point radii per integration point.
covariateData	A list containing the covariate rasters.

Value

A list containing the input for estimateModel() function.

Examples

```
if(requireNamespace("INLA")){
path1 <- system.file("extdata", "geoData.rda", package = "GeoAdjust")
path2 <- system.file("extdata", "exampleMesh.rda", package = "GeoAdjust")
load(path1)
load(path2)
inputData <- prepareInput(response = list(ys = surveyData$ys, ns = surveyData$ns),
locObs = cbind(surveyData$east, surveyData$north),
likelihood = 1, jScale = 1,
urban = surveyData$urbanRural, mesh.s = exampleMesh, adminMap = adm1,
nSubAPerPoint = 10, nSubRPerPoint = 10,
covariateData = NULL)
}
```

---

print.res *Prints the output of estimateModel() function.*

Description

Prints the output of estimateModel() function.

Usage

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

Arguments

x	A list containing the model estimation output, returned by estimateModel() function.
...	not used

Value

Prints the model estimation results of class res as a table that shows the estimated model parameters and the corresponding 95 interval lengths.

Examples

```
path1 <- system.file("extdata", "exampleInputData.rda", package = "GeoAdjust")
path2 <- system.file("extdata", "exampleMesh.rda", package = "GeoAdjust")
load(path1)
load(path2)
nNodes = exampleMesh[['n']]
results <- estimateModel(data = exampleInputData, nNodes = nNodes,
options = list(random = 1, covariates = 1), priors = list(beta = c(0,1),
range = 114, USpatial = 1, alphaSpatial = 0.05, UNugget = 1, alphaNug = 0.05), n.sims = 1000)
print(results)
```

Index

convertDegToKM, 2
convertDegToMollweide, 3
convertKMToDeg, 3
covMatern, 4

dummy, 5

estimateModel, 5

gridCountry, 6

meshCountry, 7

plotPred, 8
predRes, 9
prepareInput, 10
print.res, 11