Package 'BiDimRegression'

February 17, 2022

Version 2.0.1

Date 2022-02-17

Title Calculates the Bidimensional Regression Between Two 2D Configurations

Imports Formula, methods

Depends R (>= 4.1.0)

Description Calculates the bidimensional regression between two 2D configurations following the approach by Tobler (1965).

License GPL-3

URL https://CRAN.R-project.org/package=BiDimRegression/,
https://github.com/alexander-pastukhov/bidim-regression/,
https://alexander-pastukhov.github.io/bidim-regression/

BugReports https://github.com/alexander-pastukhov/bidim-regression/issues/

Encoding UTF-8

NeedsCompilation no

Repository CRAN

RoxygenNote 7.1.2

LazyData true

Suggests testthat, knitr, rmarkdown, dplyr, ggplot2

VignetteBuilder knitr

Author Claus-Christian Carbon [aut] (<https://orcid.org/0000-0002-3446-9347>),
Alexander Pastukhov [aut, cre]
(<https://orcid.org/0000-0002-8738-8591>)

Maintainer Alexander Pastukhov <pastukhov.alexander@gmail.com>

Date/Publication 2022-02-17 13:02:10 UTC

R topics documented:

`anova.lm2` *Anova for lm2 objects*

Description

Anova for lm2 objects, returns a table with pairwise comparisons between models or, if only one model was supplied, with the null model.

Usage

```
## S3 method for class 'lm2'
anova(object, ...)
```

Arguments

`object`	an object of class "lm2"
`...`	further objects of class "lm2"

Value

an anova data frame

See Also

`lm2`

Examples

```
lm2euc <- lm2(depV1+depV2~indepV1+indepV2, NakayaData, transformation = 'Euclidean')
lm2aff <- lm2(depV1+depV2~indepV1+indepV2, NakayaData, transformation = 'Affine')
anova(lm2euc, lm2aff)
```

`BiDimRegression` *Calculates the bidimensional regression between two 2D configurations*

Description

Calculates the bidimensional regression between two 2D configurations using both Euclidean and Affine transformations following the approach by Tobler (1965). This function assumes strict data format and returns all coefficients and statistics in a single structure. Same functionality is now re-implemented in a R-friendly style, see `lm2` function.

Usage

```
BiDimRegression(coord)
```

Arguments

`coord`	table that must contain two columns for dependent variables (named `depV1` and `depV2`) and two columns for independent variables (named `indepV1` and `indepV2`).

Value

an S3 class `BiDimRegression` containing all essential measures of the bidimensional regression

- `euclidean.r,affine.r` - the regression coefficient, defined analogously to Pearson's r.
- `euclidean.rsqr,affine.rsqr` - the squared regression coefficient.
- `euclidean.diABSqr,affine.diABSqr` - the squared distortion index for dependent variables; following Waterman and Gordon's (1984) extension of the bidimensional regression, it provides a measure of comparison of distortions, but the range of values is 0 to 1 following Friedman and Kohler (2003).
- `euclidean.dMaxABSqr,affine.dMaxABSqr` - the maximal squared distortion index for dependent variables.
- `euclidean.diXYSqr,affine.diXYSqr` - the distortion index for independent variables.
- `euclidean.dMaxXYSqr,affine.dMaxXYSqr` - the maximal squared distortion index for independent variables.
- `euclidean.scaleFactorX,affine.scaleFactorX` - the scaling factor of the first dimension (1.0 means no scaling; values below 1.0 indicate a contraction, values above 1.0 indicate an expansion).
- `euclidean.scaleFactorY,affine.scaleFactorY` - the scaling factor of the second dimension.
- `euclidean.angleDEG,affine.angleDEG` - the rotation angle **in degrees**.
- `euclidean.shear,affine.shear` - shearing of the transformed configuration, always zero for the Euclidean transformation.

- `euclidean.ttestDF,affine.ttestDF` - degrees of freedom (DF) for the t-tests regarding the model parameters (alphas and betas).
- `euclidean.alpha1.*,euclidean.alpha2.*,affine.alpha1.*,affine.alpha2.*` - intercept vectors, information includes `.coeff` for coefficient, `.SE` for standard error, `tValue` for *t*-statistics, and `pValue` for significance.
- `euclidean.beta1.*,euclidean.beta2.*,affine.beta1.*,affine.beta2.*,affine.beta3.*,affine.beta4.*` - slope vectors, information includes `.coeff` for coefficient, `.SE` for standard error, `tValue` for *t*-statistics, and `pValue` for significance.
- `euclidean.fValue,affine.fValue` - F-statistics, following the advice of Nakaya (1997).
- `euclidean.df1,affine.df1` - degrees of freedom of the nominator used for the F-statistics propagated by Nakaya (1997); df1 = p-2, with p is the number of elements needed to calculate the referring model: p=4 for the Euclidean and p=6 for the affine geometry Nakaya, 1997, Table 1.
- `euclidean.df2,affine.df2` - degrees of freedom of the denominator used for the F-statistics propagated by Nakaya (1997); df2 = 2n-p, with p is the number of elements needed to calculate the referring model (see df1) and n is the number of coordinate pairs.
- `euclidean.pValue,affine.pValue` - the significance level based on the preceding F-statistics.
- `euclidean.dAICso,affine.dAICso` - the AIC difference between the regarding bidimensional regression model and the bidimensional null model (S0) according to Nakaya (1997), formula 56.
- `eucVSaff.*` - statistical comparison between Euclidean and Affine models, include `.fValue` for F-statistics, `.df1` and `.df2` for the degrees of freedom, `.pValue` for the significance level, and `.dAIC` for AIC difference between two models.

See Also

`lm2`

Examples

```
resultingMeasures <- BiDimRegression(NakayaData)
print(resultingMeasures)
```

CarbonExample1Data — *Data from Carbon, C. C. (2013). BiDimRegression: Bidimensional Regression Modeling Using R. \ Journal of Statistical Software, Code Snippets, 52(1), 1-11 (URL http://www.jstatsoft.org/v52/c01/)*

Description

Example 1 from the domain of aesthetics to show how the method can be utilized for assessing the similarity of two portrayed persons, actually the Mona Lisa in the world famous Louvre version and the only recently re-discovered Prado version

Usage

```
data(CarbonExample1Data)
```

Format

A data frame with 36 observations on the following 4 variables.

`depV1` a numeric vector

`depV2` a numeric vector

`indepV1` a numeric vector

`indepV2` a numeric vector

Examples

```
data(CarbonExample1Data)
## maybe str(CarbonExample1Data) ; plot(CarbonExample1Data) ...
```

`CarbonExample2Data` *Data from Carbon, C. C. (2013). BiDimRegression: Bidimensional Regression Modeling Using R. \ Journal of Statistical Software, Code Snippets, 52(1), 1-11 (URL http://www.jstatsoft.org/v52/c01/)*

Description

Example 2 originates from the area of geography and inspects the accuracy of different maps of the city of Paris which were created over the last 350 years as compared to a recent map

Usage

```
data(CarbonExample2Data)
```

Format

A data frame with 13 observations on the following 4 variables.

`depV1` a numeric vector

`depV2` a numeric vector

`indepV1` a numeric vector

`indepV2` a numeric vector

Examples

```
data(CarbonExample2Data)
## maybe str(CarbonExample2Data) ; plot(CarbonExample2Data) ...
```

`CarbonExample3Data` *Data from Carbon, C. C. (2013). BiDimRegression: Bidimensional Regression Modeling Using R. \ Journal of Statistical Software, Code Snippets, 52(1), 1-11 (URL http://www.jstatsoft.org/v52/c01/)*

Description

Example 3 focuses on demonstrating how good a cognitive map recalculated from averaged cognitive distance data fits with a related real map

Usage

```
data(CarbonExample3Data)
```

Format

A data frame with 10 observations on the following 4 variables.

- `depV1` a numeric vector
- `depV2` a numeric vector
- `indepV1` a numeric vector
- `indepV2` a numeric vector

Examples

```
data(CarbonExample3Data)
## maybe str(CarbonExample3Data) ; plot(CarbonExample3Data) ...
```

`EyegazeData` *Eye gaze calibration data*

Description

A dataset containing a monocular eye gaze recording with calibration sequence. Courtesy of Bamberger Baby Institut (BamBI).

Usage

```
EyegazeData
```

Format

A data frame with 365 rows and 6 variables:

time sample timestamp, in milliseconds

x, y recorded gaze, in internal eye tracker units

target_x, target_y location of the calibration target on the screen, in pixels

target index of the target within the sequence

...

`FriedmanKohlerData1` *Data from Friedman, A., & Kohler, B. (2003). Bidimensional regression: Assessing the configural similarity and accuracy of cognitive maps and other two-dimensional data sets. Psychological Methods, 8(4), 468-491.*

Description

Data from Friedman, A., & Kohler, B. (2003). Bidimensional regression: Assessing the configural similarity and accuracy of cognitive maps and other two-dimensional data sets. Psychological Methods, 8(4), 468-491.

Usage

```
data(FriedmanKohlerData1)
```

Format

A data frame with 4 observations on the following 4 variables.

`depV1` a numeric vector

`depV2` a numeric vector

`indepV1` a numeric vector

`indepV2` a numeric vector

Examples

```
data(FriedmanKohlerData1)
## maybe str(FriedmanKohlerData1) ; plot(FriedmanKohlerData1) ...
```

FriedmanKohlerData2 — *Data from Friedman, A., & Kohler, B. (2003). Bidimensional regression: Assessing the configural similarity and accuracy of cognitive maps and other two-dimensional data sets. Psychological Methods, 8(4), 468-491.*

Description

Data from Friedman, A., & Kohler, B. (2003). Bidimensional regression: Assessing the configural similarity and accuracy of cognitive maps and other two-dimensional data sets. Psychological Methods, 8(4), 468-491.

Usage

```
data(FriedmanKohlerData2)
```

Format

A data frame with 4 observations on the following 4 variables.

- `depV1` a numeric vector
- `depV2` a numeric vector
- `indepV1` a numeric vector
- `indepV2` a numeric vector

Examples

```
data(FriedmanKohlerData2)
## maybe str(FriedmanKohlerData2) ; plot(FriedmanKohlerData2) ...
```

lm2 — *Fitting Bidimensional Regression Models*

Description

lm2 is used to fit bidimensional linear regression models using Euclidean and Affine transformations following the approach by Tobler (1965).

Usage

```
lm2(formula, data, transformation)
```

Arguments

`formula` a symbolic description of the model to be fitted in the format `A + B ~ C + D`, where `A` and `B` are dependent and `C` and `D` are independent variables

`data` a data frame containing variables for the model.

`transformation` the transformation to be used, either `'euclidean'`, `'affine'`, or `'projective'`.

Value

lm2 returns an object of class "lm2". An object of class "lm" is a list containing at least the following components:

`transformation` string with the transformation type (`euclidean`, `affine`, or `projective`)

`npredictors` number of predictors used in the model: 4 for euclidean, 6 for affine, 8 for projective.

`df_model, df_residual`
degrees of freedom for the model and for the residuals

`transformation_matrix`
`3x3` transformation matrix

`coeff` transformation coefficients, with `a` denoting the intercept terms.

`transformed_coeff`
`scale`, `angle`, and `sheer` coefficients, depends on transformation.

`fitted_values` data frame containing fitted values for the original data set

`residuals` data frame containing residuals for the original fit

`r.squared, adj.r.squared`
R-squared and adjusted R-squared.

`F, p.value` F-statistics and the corresponding p-value, given the `df_model` and `df_residual` degrees of freedom.

`dAIC` Akaike Information Criterion (AIC) difference between the regression model and the null model. A negative values indicates that the regression model is better. See *Nakaya (1997)*.

`distortion_index`
Distortion index following *Waterman and Gordon (1984)*, as adjusted by *Friedman and Kohler (2003)*

`lm` an underlying linear model for `Euclidean` and `affine` transformations.

`formula` formula, describing input and output columns

`data` data used to fit the model

`Call` function call information, incorporates the `formula`, `transformation`, and `data`.

See Also

`anova.lm2` `BiDimRegression`

Examples

```
lm2euc <- lm2(depV1 + depV2 ~ indepV1 + indepV2, NakayaData, 'euclidean')
lm2aff <- lm2(depV1 + depV2 ~ indepV1 + indepV2, NakayaData, 'affine')
lm2prj <- lm2(depV1 + depV2 ~ indepV1 + indepV2, NakayaData, 'projective')
anova(lm2euc, lm2aff, lm2prj)
predict(lm2euc)
summary(lm2euc)
```

`NakayaData` *Data from Nakaya, T. (1997). Statistical inferences in bidimensional regression models. Geographical Analysis, 29(2), 169-186.*

Description

Data from Nakaya, T. (1997). Statistical inferences in bidimensional regression models. Geographical Analysis, 29(2), 169-186.

Usage

```
data(NakayaData)
```

Format

A data frame with 19 observations on the following 4 variables.

`depV1` a numeric vector

`depV2` a numeric vector

`indepV1` a numeric vector

`indepV2` a numeric vector

Examples

```
data(NakayaData)
## maybe str(NakayaData) ; plot(NakayaData) ...
```

predict.lm2 *Predict method for Bidimensional Regression Model Fits*

Description

Predicted values based on the bidimensional regressional model object.

Usage

```
## S3 method for class 'lm2'
predict(object, newdata, ...)
```

Arguments

object	an object of class "lm2"
newdata	An optional two column data frame with independent variables. If omitted, the fitted values are used.
...	optional arguments

Value

a two column data frame with predicted values for dependent variables.

See Also

lm2

Examples

```
lm2euc <- lm2(depV1+depV2~indepV1+indepV2, NakayaData, transformation = 'Euclidean')
predict(lm2euc, NakayaData[, 3:4])
```

Index