

# Package ‘CoDaImpact’

July 21, 2025

**Title** Interpreting CoDa Regression Models

**Version** 0.1.0

**Description** Provides methods for interpreting CoDa (Compositional Data) regression models along the lines of “Pairwise share ratio interpretations of compositional regression models” (Dargel and Thomas-Agnan 2024) <[doi:10.1016/j.csda.2024.107945](https://doi.org/10.1016/j.csda.2024.107945)>. The new methods include variation scenarios, elasticities, elasticity differences and share ratio elasticities. These tools are independent of log-ratio transformations and allow an interpretation in the original space of shares. ‘CoDaImpact’ is designed to be used with the ‘compositions’ package and its ecosystem.

**License** GPL (>= 3)

**Encoding** UTF-8

**RoxygenNote** 7.3.1

**URL** <https://github.com/LukeCe/CoDaImpact>,  
<https://lukece.github.io/CoDaImpact/>

**BugReports** <https://github.com/LukeCe/CoDaImpact/issues>

**Imports** methods, compositions

**Depends** R (>= 2.10)

**LazyData** true

**Suggests** rmarkdown, knitr, sf, tinytest

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Lukas Dargel [aut, cre] (ORCID:  
<<https://orcid.org/0000-0002-4839-506X>>),  
Christine Thomas-Agnan [aut] (ORCID:  
<<https://orcid.org/0000-0002-6430-3110>>),  
Rodrigue Nasr [ctb],  
Sijia Pan [ctb],  
Iban Rendo Barreiro [ctb],  
Shuyao Li [ctb]

**Maintainer** Lukas Dargel <lukas.dargel@mailbox.org>  
**Repository** CRAN  
**Date/Publication** 2024-03-23 10:30:02 UTC

Contents

car_market . . . . .	2
CoDa_path . . . . .	3
CoDa_seq . . . . .	5
coef.lmCoDa . . . . .	6
confint.lmCoDa . . . . .	7
election . . . . .	8
fitted.lmCoDa . . . . .	9
Impacts . . . . .	10
lmCoDa . . . . .	11
predict.lmCoDa . . . . .	12
residuals.lmCoDa . . . . .	13
rice_yields . . . . .	13
ShareRatioElasticities . . . . .	14
ToSimplex . . . . .	16
toulouse_retail . . . . .	17
VariationScenario . . . . .	18
VariationTable . . . . .	19
<b>Index</b>	<b>22</b>

---

car_market	<i>French car market data</i>
------------	-------------------------------

---

Description

This data set shows monthly data of the French car market between 2003 and 2015. The market is divided into 5 main segments (A to E), according to the size of the vehicle chassis. Morais et. al (2018) first used this data to compare compositional and Dirichlet models for market shares.

Usage

car\_market

Format

An object of class `data.frame` with 152 rows and 10 columns.

## Details

- SEG\_: Corresponds to the shares of sales in each of the five market segments A,B,C,D and E. Where A are the smallest cars and E the largest. The segmentation is explained in [Wikipedia](#).
- GDP: GDP figures in millions at current prices
- HOUSEHOLD\_EXPENDITURE: total household expenditure in millions at previous years prices
- GAS\_PRICE: Corresponds to the gas price including VAT.
- SCRAPPING\_SUBSIDY: A dummy indicating periods where the French government provided subsidies for scrapping a car.

## Author(s)

Lukas Dargel, Christine Thomas-Agnan

## Source

- The figures for GDP and household expenditure are originally provided by the The National Institute of Statistics and Economic Studies (INSEE).
- The gas prices are from the OECD.
- The market share of each segment of come from a simulation by Renault.

## References

Joanna Morais, Christine Thomas-Agnan & Michel Simioni (2018) Using compositional and Dirichlet models for market share regression, Journal of Applied Statistics, 45:9, 1670-1689, DOI: 10.1080/02664763.2017.138986

---

CoDa_path	<i>Create a linear path in the simplex by defining a direction and a step size</i>
-----------	--

---

## Description

Create a linear path in the simplex by defining a direction and a step size

## Usage

```
CoDa_path(
  comp_direc,
  comp_from,
  step_size = 0.01,
  n_steps = 100,
  add_opposite = FALSE,
  dir_from_start = FALSE
)
```

**Arguments**

<code>comp_direc</code>	A numeric vector, defining a direction in the simplex
<code>comp_from</code>	A numeric vector, an initial point in the simplex - defaults to a balanced composition, which represents the origin in the simplex
<code>step_size</code>	A numeric, indicating the step size
<code>n_steps</code>	A numeric, indicating the number of steps to be taking from <code>comp_from</code>
<code>add_opposite</code>	A logical, if TRUE steps in the opposite direction are also computed
<code>dir_from_start</code>	A logical, if TRUE the direction is calculated from the difference between <code>comp_from</code> and <code>comp_direc</code>

**Details**

The function is very similar to `CoDa_seq()`. However, of drawing a line between a starting and end point it uses only a starting point and a direction.

**Value**

A data.frame frame where each row corresponds to one compositional vector

**Author(s)**

Lukas Dargel

**See Also**

`CoDa_seq`

**Examples**

```
# three steps that go from the origin towards the defined direction
comp_direc <- c(A = .4, B = .35, C = .25)
CoDa_path(comp_direc, n_steps = 3)

# we can draw the path that is defined by this direction
comp_direc <- c(A = .4, B = .35, C = .25)
compositions::plot.acomp(CoDa_path(comp_direc, n_steps = 10))
compositions::plot.acomp(CoDa_path(comp_direc, n_steps = 100))
compositions::plot.acomp(CoDa_path(comp_direc, add_opposite = TRUE))

# using the same direction we can draw a new path that does not go through the origin
comp_direc <- c(A = .4, B = .35, C = .25)
comp_from <- c(.7, .2, .1)
compositions::plot.acomp(CoDa_path(comp_direc, comp_from, n_steps = 10))
compositions::plot.acomp(CoDa_path(comp_direc, comp_from, n_steps = 100))
compositions::plot.acomp(CoDa_path(comp_direc, comp_from, add_opposite = TRUE))
```

```
# the balanced composition does not define a direction by itself
comp_origin <- c(A = 1/3, B = 1/3, C= 1/3) # corresponds to a zero vector in real space
try(CoDa_path(comp_origin, comp_from, add_opposite = TRUE))

# with the dir_from_start option the direction is derived
# from the simplex line connecting two compositions
path_origin <- CoDa_path(
  comp_direc = comp_origin,
  comp_from = comp_from,
  add_opposite = TRUE,
  dir_from_start = TRUE,
  step_size = .1)
compositions::plot.acomp(path_origin)
compositions::plot.acomp(comp_origin, add = TRUE, col = "blue", pch = 19)
compositions::plot.acomp(comp_from, add = TRUE, col = "red", pch = 19)
```

---

CoDa\_seq

*A sequence connecting two points in a simplex*


---

## Description

A sequence connecting two points in a simplex

## Usage

```
CoDa_seq(comp_from, comp_to, n_steps = 100, add_opposite = FALSE)
```

## Arguments

comp_from	A numeric vector, representing the initial compositions
comp_to	A numeric vector, representing the final compositions.
n_steps	An integer, indicating the number of steps used to go from comp_from to comp_to
add_opposite	A logical, if TRUE the path in the opposite direction is added

## Details

The sequence is evenly spaced and corresponds to a straight line in the simplex geometry. If no end point is provided the line will connect the initial point with the first summit of the simplex. Since exact zeros are not handled by the ilr they are replaced by a small constant.

## Value

A data.frame frame where each row corresponds to one compositional vector

## Author(s)

Lukas Dargel

**See Also**

simplex\_increment

**Examples**

```
# path to the first summit of the simplex
start_comp <- c(A =.4,B = .35, C= .25)
compositions::plot.acomp(CoDa_seq(start_comp))
compositions::plot.acomp(CoDa_seq(start_comp, add_opposite = TRUE))

# path to an edge of the simplex
end_comp <- c(0,.8,.2)
compositions::plot.acomp(CoDa_seq(start_comp, end_comp))
compositions::plot.acomp(CoDa_seq(start_comp, end_comp,add_opposite = TRUE))
```

---

coef.lmCoDa	<i>Predictions, fitted values, residuals, and coefficients in CoDa models</i>
-------------	---

---

**Description**

These functions work as in the usual lm object. They additionally offer the possibility use the space argument which transforms them into directly into clr space or in the simplex.

**Usage**

```
## S3 method for class 'lmCoDa'
coef(object, space = NULL, split = FALSE, ...)
```

**Arguments**

object	class "lmCoDa"
space	a character indicating in which space the prediction should be returned. Supported are the options c("clr", "simplex").
split	logical, if TRUE the coefficients are reported as a list instead of a matrix, where list structure reflects the explanatory variables of the model
...	not used

**Value**

a matrix

**Author(s)**

Lukas Dargel

---

confint.lmCoDa	<i>Confidence Intervals for CoDa Models</i>
----------------	---

---

**Description**

Dargel and Thomas-Agnan (2024) show to compute variances and confidence intervals for parameters of CoDa models in log-ratio spaces.

Of particular interest are the clr parameters since they can be directly interpreted as differences from an average elasticity.

Another option is interpret the difference in clr parameters as these coincide with the difference in elasticities.

**Usage**

```
## S3 method for class 'lmCoDa'
confint(object, parm, level = 0.95, y_ref = NULL, obs = NULL, ...)
```

**Arguments**

object	class "lmCoDa"
parm	a character, indicating the name of one explanatory variable
level	a numeric, indicating the confidence level required
y_ref	an optional argument that indicates the reference component of the response variable using its name or its position. This argument is only used in the Y-compositional model. If it is supplied confidence intervals of difference are used instead of the direct intervals of the parameters.
obs	an optional integer that indicates one observation when this argument is supplied the function return the observation dependent elasticity
...	passed on to confint()

**Details**

Since CoDa models are often multivariate this function only allows to specify one explanatory variable at a time. The output is also more complex than the usual one for "lm" classes, because we have to indicate the component of Y and X. With `confint.lm()` it is still possible to compute the usual the confidence intervals.

**Value**

data.frame

**Author(s)**

Lukas Dargel

References

- Dargel, Lukas and Christine Thomas-Agnan, “Pairwise share ratio interpretations of compositional regression models”, Computational Statistics & Data Analysis 195 (2024), p. 107945

Examples

```
## ==== Y-compositional model ====
res <- lmCoDa(
  ilr(cbind(left, right, extreme_right)) ~
  ilr(cbind(Age_1839, Age_4064)) +
  ilr(cbind(Educ_BeforeHighschool, Educ_Highschool, Educ_Higher)) +
  unemp_rate,
  data = head(election, 20))

## ---- CI for scalar X
# CI for clr parameters
confint(res, "unemp_rate")
# CI for difference in clr parameters (coincides with difference in the semi elasticity)
confint(res, "unemp_rate", y_ref = 1)

## ---- CI for compositional X
# CI for clr parameters
confint(res, "cbind(Age_1839, Age_4064)")

# CI for difference in clr parameters (coincides with difference in the elasticity)
confint(res, "cbind(Age_1839, Age_4064)", y_ref = 1)
```

---

election	<i>Results of french departmental elections in 2015</i>
----------	---

---

Description

The data is used by Nguyen et. al (2020) and originally disseminated by the French ministry (Ministère de l’Intérieur et des Outre-Mer). Information about the population characteristics comes from the french national statistics institute (INSEE).

Usage

election

Format

An object of class `data.frame` with 95 rows and 13 columns.



## Details

- left, right, extreme\_right: Vote shares during the election grouped into three blocks
- Age\_1839, Age\_4064, Age\_65plus: Share of the population falling into one of three age categories
- Educ\_BeforeHighschool, Educ\_Highschool, Educ\_Higher: Share of the population having completed a certain level of education.
- asset\_owner\_rate: The proportion of people who own assets
- income\_taxpayer\_rate: The proportion of people who pay income tax
- foreigner\_rate: The proportion of foreigners

## Author(s)

Lukas Dargel, Christine Thomas-Agnan

## Source

- <https://www.data.gouv.fr/fr/datasets/elections-departementales-2015-resultats-par-bureaux-de-vote>
- <https://www.insee.fr/fr/accueil>

## References

Nguyen THA, Laurent T, Thomas-Agnan C, Ruiz-Gazen A. Analyzing the impacts of socio-economic factors on French departmental elections with CoDa methods. J Appl Stat. 2020 Dec 9;49(5):1235-1251. doi: 10.1080/02664763.2020.1858274. PMID: 35707505; PMCID: PMC9041641.

---

fitted.lmCoDa	<i>Predictions, fitted values, residuals, and coefficients in CoDa models</i>
---------------	---

---

## Description

These functions work as in the usual lm object. They additionally offer the possibility use the space argument which transforms them into directly into clr space or in the simplex.

## Usage

```
## S3 method for class 'lmCoDa'
fitted(object, space = NULL, ...)
```

## Arguments

object	class "lmCoDa"
space	a character indicating in which space the prediction should be returned. Supported are the options c("clr", "simplex").
...	passed on to <code>predict.lm()</code>

**Value**

matrix or vector

**Author(s)**

Lukas Dargel

---

Impacts	<i>Computation of elasticities in CoDa regression models</i>
---------	--

---

**Description**

This function computes elasticities and semi-elasticities for CoDa regression model. where we have to distinguish four cases:

- Y and X are both compositional: this leads to an elasticity
- Y is compositional and X is scalar: this leads to a semi-elasticity
- Y is scalar and X is compositional: this leads to a semi-elasticity
- Y and X are both scalar: this case is not implemented as it leads to constant marginal effects

**Usage**

```
Impacts(object, Xvar = NULL, obs = 1)
```

**Arguments**

object	an object of class "lmCoDa"
Xvar	a character indicating the name of one explanatory variable
obs	a numeric that refers to the indicator of one observation

**Details**

The mathematical foundation for elasticity computations in CoDa model come from Morais and Thomas-Agnan (2021). Dargel and Thomas-Agnan (2024) present further results and illustrations.

**Value**

a matrix

**Author(s)**

- Lukas Dargel
- Rodrigue Nasr

## References

- Dargel, Lukas and Christine Thomas-Agnan, “Pairwise share ratio interpretations of compositional regression models”, Computational Statistics & Data Analysis 195 (2024), p. 107945
- Morais, Joanna and Christine Thomas-Agnan. "Impact of covariates in compositional models and simplicial derivatives." Austrian Journal of Statistics 50.2 (2021): 1-15.

## Examples

```
res <- lmCoDa(YIELD ~ PRECIPITATION + ilr(TEMPERATURES), data = head(rice_yields,20))
Impacts(res, Xvar = "TEMPERATURES")
```

---

lmCoDa

*Estimating CoDa regression models*


---

## Description

This is a thin wrapper around `lm()` followed by `ToSimplex()`, which allows to create a `lmCoDa` object in one step.

## Usage

```
lmCoDa(formula, data, ...)
```

## Arguments

formula	as in <code>lm()</code>
data	as in <code>lm()</code>
...	arguments passed on to <code>lm()</code>

## Value

an object of class "lm" and "lmCoDa" if the formula include at least one log-transformation

## Author(s)

Lukas Dargel

## See Also

`lm()`, `ToSimplex()`, `compositions::ilr()`, `compositions::alr()`

**Examples**

```
# XY-compositional model
res <- lmCoDa(
  ilr(cbind(left, right, extreme_right)) ~
  ilr(cbind(Educ_BeforeHighschool, Educ_Highschool, Educ_Higher)),
  data = head(election, 20))

# X-compositional model
res <- lmCoDa(YIELD ~ PRECIPITATION + ilr(TEMPERATURES), data = head(rice_yields, 20))
```

---

predict.lmCoDa	<i>Predictions, fitted values, residuals, and coefficients in CoDa models</i>
----------------	---

---

**Description**

These functions work as in the usual `lm` object. They additionally offer the possibility use the `space` argument which transforms them into directly into `clr` space or in the simplex.

**Usage**

```
## S3 method for class 'lmCoDa'
predict(object, space = NULL, ...)
```

**Arguments**

<code>object</code>	class "lmCoDa"
<code>space</code>	a character indicating in which space the prediction should be returned. Supported are the options <code>c("clr", "simplex")</code> .
<code>...</code>	passed on to <code>predict.lm()</code>

**Value**

matrix or vector

**Author(s)**

Lukas Dargel

---

residuals.lmCoDa	<i>Predictions, fitted values, residuals, and coefficients in CoDa models</i>
------------------	---

---

**Description**

These functions work as in the usual `lm` object. They additionally offer the possibility use the `space` argument which transforms them into directly into `clr` space or in the simplex.

**Usage**

```
## S3 method for class 'lmCoDa'
residuals(object, space = NULL, ...)
```

**Arguments**

<code>object</code>	class "lmCoDa"
<code>space</code>	a character indicating in which space the prediction should be returned. Supported are the options <code>c("clr", "simplex")</code> .
<code>...</code>	passed on to <code>predict.lm()</code>

**Value**

matrix or vector

**Author(s)**

Lukas Dargel

---

rice_yields	<i>Data on the rice yields in the Vietnamese provinces</i>
-------------	--

---

**Description**

The data is presented in Trinh et al. (2023) for studying the impact of climate change on rice production in Vietnam.

It contains the following information:

- PROVINCE: a factor for the 63 provinces of Vietnam
- REGION: a factor with the 6 main regions
- YEAR: a numeric corresponding to the year
- YIELD: a numeric for the rice production in tons per hectare
- PRECIPITATION: a numeric for the annual precipitation in liters
- TEMPERATURES: a compositional variable represented as a matrix whose columns correspond to the proportion of days in a year where the maximal temperature (in Celsius degrees) falls into one of the three categories: "LOW" (from -6, to 25.1), "MEDIUM" (from 25.1 to 35.4) and "HIGH" (from 35.4 to 45).

Usage

rice\_yields

Format

An object of class `data.frame` with 1890 rows and 6 columns.

Author(s)

Lukas Dargel, Christine Thomas-Agnan

References

Thi-Huong Trinh, Michel Simioni, and Christine Thomas-Agnan, “Discrete and Smooth Scalar-on-Density Compositional Regression for Assessing the Impact of Climate Change on Rice Yield in Vietnam”, TSE Working Paper, n. 23-1410, February 2023.

---

ShareRatioElasticities
<i>Compute share ratio elasticities for CoDa models</i>

---

Description

In CoDa models with compositional dependent variable (Y) share ratio elasticities (SRE) allow to interpret the influence of compositional explanatory variables (X). The interpretation is analogous to usual elasticities:

- When the share ratio of X increases by 1% the share ratio of Y increases by SRE%
- The main difference to usual elasticities that, since X is compositional the change of X must be specified in terms of a direction in the simplex.

Usage

`ShareRatioElasticities(object, Xvar, Xdir = NULL)`

Arguments

object	an object of class "lmCoDa"
Xvar	a character indicating the name of the explanatory variable that changes
Xdir	a numeric vector, a single character, or NULL: <ul style="list-style-type: none"><li>• if numeric Xdir is taken as a fixed direction in the simplex</li><li>• if character Xdir is interpreted as one summit of the X composition and converted to the fixed direction towards this summit</li><li>• if NULL the share ratio elasticities are computed for variable directions corresponding the example in Dargel and Thomas-Agnan (2024) Lukas Dargel &amp; Christine Thomas-Agnan (2024) The link between multiplicative competitive interaction models and compositional data regression with a total, Journal of Applied Statistics, DOI: 10.1080/02664763.2024.2329923 )</li></ul>

**Details**

More details on this interpretation can be found in Dargel and Thomas-Agnan (2024) and in the accompanying vignette.

**Value**

a data.frame

**Author(s)**

Lukas Dargel

**References**

- Dargel, Lukas and Christine Thomas-Agnan, “Pairwise share ratio interpretations of compositional regression models”, Computational Statistics & Data Analysis 195 (2024), p. 107945

**Examples**

```
### XY-compositional model
res <- lmCoDa(
  ilr(cbind(left, right, extreme_right)) ~
  ilr(cbind(Educ_BeforeHighschool, Educ_Highschool, Educ_Higher)),
  data = head(election, 20))

## Focus on changes in the education composition
educ_comp <- "cbind(Educ_BeforeHighschool, Educ_Highschool, Educ_Higher)"

## case 1
## changes towards the summit "Educ_Higher" as (fixed) direction
SRE1 <- ShareRatioElasticities(res, Xvar = educ_comp, Xdir = "Educ_Higher")

SRE1[1,]
# Result: SRE=Inf
# cannot be interpreted because, for this direction,
# the relative change in the share ratio of X (Highschool / BeforeHighschool) is zero
SRE1[7,]
# Result: SRE=0.9
# when the ratio of X (Higher / BeforeHighschool) increases by 1%
# the ratio of Y (right / left) increases by about 0.9%

## case 2
## numeric vector as (fixed) direction
SRE2 <- ShareRatioElasticities(res, Xvar = educ_comp, Xdir = exp(c(0,0,1)))
identical(SRE1,SRE2) # exp(c(0,0,1)) is the direction that points to the third summit

## case 3
## variable directions with Xdir = NULL
## In this case the direction depends components used for the share ratio of X
## In particular the component of X in the numerator grows
## by the same rate as the denominator decreases
SRE3 <- ShareRatioElasticities(res, Xvar = educ_comp, Xdir = NULL)
```

```
SRE3[1,]
# Result: SRE=-2.8
# when the ratio of X (Highschool / BeforeHighschool) increases by 1%
# the ratio of Y (right / left) decreases by about -2.8%
```

---

ToSimplex

---

*Converting Linear Models to CoDa models*


---

## Description

The function converts the output of a "lm" to the "lmCoDa" class, which offers additional tools for the interpretation of a CoDa regression models. Most of the work is done by the [transformationSummary\(\)](#) function, which has its own documentation page, but should be reserved for internal use.

## Usage

```
ToSimplex(object)
```

## Arguments

object                    an object of class "lmCoDa"

## Value

an object of class "lm" and "lmCoDa" if the formula include at least one log-transformation

## Author(s)

- Lukas Dargel
- Rodrigue Nasr

## See Also

[lm\(\)](#), [lmCoDa\(\)](#)

## Examples

```
# XY-compositional model
res <- lm(
  ilr(cbind(left, right, extreme_right)) ~
  ilr(cbind(Educ_BeforeHighschool, Educ_Highschool, Educ_Higher)),
  data = head(election, 20))
res <- ToSimplex(res)

# X-compositional model
res <- lm(YIELD ~ PRECIPITATION + ilr(TEMPERATURES), data = head(rice_yields, 20))
res <- ToSimplex(res)
```



---

toulouse_retail	<i>Simulated retail data for nine shopping malls in the city of Toulouse</i>
-----------------	--

---

**Description**

This data set provides an example for the use of CoDa models in geomarketing applications. The data is simulated, but realistic in the sense that the parameters used for the simulation were estimated on a real, but confidential data set (Dargel and Thomas-Agnan 2024).

**Usage**

toulouse\_retail

**Format**

An object of class sf (inherits from data.table, data.frame) with 428 rows and 6 columns.

**Details**

- ID\_IRIS: Identifies the geographic region.
- POP: Population number within the region
- MEDIAN\_INCOME: The median income in the region
- dist\_km: Distances from the region to all nine shopping malls.
- visits: The share of visitors that coming from the region and going to each of the malls.
- geometry: The geometry (polygon) of the region
- The "mall\_locations" and the "simulation\_parameters" are given as additional attributes.

**Author(s)**

Lukas Dargel, Christine Thomas-Agnan

**Source**

- The figures for POP and MEDIAN\_INCOME come from the French census data prided by INSEE.
- The polygon geometry is provided by the IGN.
- The locations of the nine shopping malls around the city center are derived from online mapping services (Google Maps and OpenStreetMap).
- The distances (dist\_km) are derived from location information.
- The number of shopping trips (visits) are simulated by the authors.

**References**

Lukas Dargel & Christine Thomas-Agnan (2024) “The link between multiplicative competitive interaction m

---

VariationScenario	<i>Scenarios for variation in CoDa regressions models</i>
-------------------	---

---

## Description

Scenarios of this type are illustrated in Dargel and Thomas-Agnan (2024). They allow to evaluate how the response variable (Y) in a CoDa model would evolve under a hypothetical scenario for linear changes in one explanatory variable (X). When the changing explanatory variable is compositional the term "linear" is understood with respect to the geometry of the simplex.

## Usage

```
VariationScenario(
  object,
  Xvar,
  Xdir,
  obs = 1,
  inc_size = 0.1,
  n_steps = 100,
  add_opposite = TRUE,
  normalize_Xdir = TRUE
)
```

## Arguments

object	an object of class "lmCoDa"
Xvar	a character indicating the name of the explanatory variable that changes
Xdir	either character or numeric, to indicate the direction in which Xvar should change <ul style="list-style-type: none"> <li>• when character this should be one of the components of X, in which case the direction is the corresponding vertex of the simplex</li> <li>• when numeric this argument is coerced to a unit vector in the simplex</li> <li>• (when Xvar refers to a scalar variable this argument is ignored)</li> </ul>
obs	a numeric indicating the observation used for the scenario
inc_size	a numeric indicating the distance between each point in the scenario of X
n_steps	a numeric indicating the number of points in the scenario
add_opposite	a logical, if TRUE the scenario also includes changes in the opposite direction
normalize_Xdir	a logical, if TRUE the direction Xdir scaled to have an Aitchison norm of 1, allowing to interpret inc_size as the Aitchison distance

## Details

The linear scenario for X is computed with `seq()` in the scalar case and with `CoDa_seq()` in the compositional case. The corresponding changes in Y are computed with the prediction formula, where we exploit the fact that only in one variable is changing.

**Value**

a data.frame containing the scenario of X and the corresponding predicted values of Y

**Author(s)**

Lukas Dargel

**References**

- Dargel, Lukas and Christine Thomas-Agnan, “Pairwise share ratio interpretations of compositional regression models”, Computational Statistics & Data Analysis 195 (2024), p. 107945

**Examples**

```
# ---- model with scalar response ----
res <- lmCoDa(YIELD ~ PRECIPITATION + ilr(TEMPERATURES), data = head(rice_yields,20))
VariationScenario(res, Xvar = "TEMPERATURES", Xdir = "MEDIUM", n_steps = 5)
VariationScenario(res, Xvar = "PRECIPITATION", n_steps = 5)

# ---- model with compositional response ----
res <- lmCoDa(ilr(cbind(left, right, extreme_right)) ~
  ilr(cbind(Age_1839, Age_4064)) +
  ilr(cbind(Educ_BeforeHighschool, Educ_Highschool, Educ_Higher)) +
  log(unemp_rate),
  data = head(election))

VariationScenario(res, Xvar = "cbind(Age_1839, Age_4064)", Xdir = "Age_1839", n_steps = 5)
VariationScenario(res, "log(unemp_rate)", n_steps = 5)
```

---

VariationTable

*Effects of infinitesimal changes in CoDa models*


---

**Description**

This function allows to evaluate how a change in an explanatory variables impacts the response variable in a CoDa regression model. The changes are calculated based from the approximate formal presented in Dargel and Thomas-Agnan (2024). Changes in the response variables are provided as data.frame and the underlying changes in the explanatory variable are given as attributes.

**Usage**

```
VariationTable(
  object,
  Xvar,
  Xdir,
  obs = 1,
```

```

    inc_size = 0.1,
    inc_rate = NULL,
    Ytotal = 1,
    normalize_Xdir = TRUE
  )

```

### Arguments

<code>object</code>	an object of class "lmCoDa"
<code>Xvar</code>	a character indicating the name of the explanatory variable that changes
<code>Xdir</code>	either character or numeric, to indicate the direction in which <code>Xvar</code> should change <ul style="list-style-type: none"> <li>• when character this should be one of the components of <code>X</code>, in which case the direction is the corresponding vertex of the simplex</li> <li>• when numeric this argument is coerced to a unit vector in the simplex</li> <li>• (when <code>Xvar</code> refers to a scalar variable this argument is ignored)</li> </ul>
<code>obs</code>	a numeric indicating the observation used for the scenario
<code>inc_size</code>	a numeric indicating the distance between each point in the scenario of <code>X</code>
<code>inc_rate</code>	a numeric that can be used as a parameterization of the step size
<code>Ytotal</code>	a numeric indicating the total of <code>Y</code>
<code>normalize_Xdir</code>	a logical, if <code>TRUE</code> the direction <code>Xdir</code> scaled to have an Aitchison norm of 1, allowing to interpret <code>inc_size</code> as the Aitchison distance

### Value

`data.frame`

### Author(s)

- Lukas Dargel
- Rodrigue Nasr

### References

- Dargel, Lukas and Christine Thomas-Agnan, "Pairwise share ratio interpretations of compositional regression models", *Computational Statistics & Data Analysis* 195 (2024), p. 107945

### Examples

```

# XY-compositional model
res <- lmCoDa(
  ilr(cbind(left, right, extreme_right)) ~
  ilr(cbind(Educ_BeforeHighschool, Educ_Highschool, Educ_Higher)),
  data = head(election, 20))

# Focus on changes in the education composition
educ_comp <- "cbind(Educ_BeforeHighschool, Educ_Highschool, Educ_Higher)"

```

```
# ... changes towards a summit towards a summit (higher share of people with lower education)
VariationTable(res, educ_comp, Xdir = "Educ_BeforeHighschool")

# ... same changes using a compositional vector as direction
VariationTable(res, educ_comp, Xdir = c(.5,.25,.25))

# ... changes in a more general direction and for a different observation
VariationTable(res, educ_comp, Xdir = c(.35,.45,.10), obs = 2)
```

# Index

- \* **data**
  - car\_market, [2](#)
  - election, [8](#)
  - rice\_yields, [13](#)
  - toulouse\_retail, [17](#)
- car\_market, [2](#)
- CoDa\_path, [3](#)
- CoDa\_seq, [5](#)
- CoDa\_seq(), [4](#), [18](#)
- coef.lmCoDa, [6](#)
- compositions::alr(), [11](#)
- compositions::ilr(), [11](#)
- confint.lm(), [7](#)
- confint.lmCoDa, [7](#)
- election, [8](#)
- fitted.lmCoDa, [9](#)
- Impacts, [10](#)
- lm(), [11](#), [16](#)
- lmCoDa, [11](#)
- lmCoDa(), [16](#)
- predict.lm(), [9](#), [12](#), [13](#)
- predict.lmCoDa, [12](#)
- residuals.lmCoDa, [13](#)
- rice\_yields, [13](#)
- seq(), [18](#)
- ShareRatioElasticities, [14](#)
- ToSimplex, [16](#)
- ToSimplex(), [11](#)
- toulouse\_retail, [17](#)
- transformationSummary(), [16](#)
- VariationScenario, [18](#)
- VariationTable, [19](#)