Package ‘VSdecomp’

October 12, 2022

Title Variance and Skewness Decomposition

Version 0.1.1

Description Provides decomposition methods for the skewness or variance of a variable (e.g., wage).
By breaking distribution moments into independent components, users can
analyze changes in distributions across time or between groups.

License GPL-3

Encoding UTF-8

RoxygenNote 7.1.1

Imports Hmisc, ggplot2, reshape, lfe, rlang

Suggests testthat

NeedsCompilation no

Author Elad Guttman [aut, cre],
Oren Danieli [aut]

Maintainer Elad Guttman <eladguttman@mail.tau.ac.il>

Repository CRAN

Date/Publication 2021-05-18 08:00:08 UTC

R topics documented:

`linear_projection` *Linear Projection*

Description

estimates the linear model $y = \beta * X + \epsilon$ and returns its linear components, grouped according to X.list.

Usage

```
linear_projection(
  y,
  X.list,
  data,
  wgt = rep(1, nrow(data)),
  year = rep(1, nrow(data)),
  comp.names = NULL
)
```

Arguments

`y`	a character specifying the name of the outcome variable (e.g. "wage"). Note that this variable is standardized before it's projected onto X.
`X.list`	a list containing the names of all the variables needed for the linear projection, grouped according to the components will later be used in the skewness decomposition. For example: for X.list = list("x1", c("x2", "x3")) the following components are returned: $\beta 1X1$, ($\beta 2X2 + \beta 3X3$), ϵ. Currently interactions aren't supported, so the user should insert them manually.
`data`	a data frame with all the variables specified in X.list and y.
`wgt`	an optional vector of weights.
`year`	an optional vector of years. if provided, the projection is done for each year separately.
`comp.names`	an optional vector specifying name for each component.

Value

a matrix with the (centered) components specified by X.list + residuals. Note that each row is summed (up to a constant) to the standardized version of y, and each column to 0 (both by year).

Examples

```
#gen data
n <- 1000
X <- matrix(rnorm(n*3), ncol = 3)
colnames(X) <- c("x1", "x2", "x3")
beta <- c(1,2,3)
```

```
wage <- X %*% beta + rnorm(n)
dat <- as.data.frame(cbind(wage, X))
colnames(dat)[1] <- "wage"
res <- linear_projection("wage", X.list = list("x1", c("x2", "x3")), data = dat)
#each row is summed (up to a constant) to the standardized wage:
stand_wage <- (wage - mean(wage)) / sd(wage)
diff <- apply(res, 1, sum) - stand_wage
summary(diff)
```

---

plot.vs_decomp *Plot Method for 'vs_decomp' Objects*

Description

Plot Method for 'vs_decomp' Objects

Usage

```
## S3 method for class 'vs_decomp'
plot(
  x,
  plot.comp = NULL,
  comp.labels = NULL,
  fill.colors = NULL,
  trunc.negative = TRUE,
  ...
)
```

Arguments

`x`	a 'vs_decomp' object.
`plot.comp`	a vector of up to 3 components to plot. can be either character with component names or numeric with component indices. other components not specified by plot.comp are summed to one additional component. default is to plot the first 3 components. another option is to additionally specify "all other terms" in order to control for the position of each component. (relevant only for the linear decomposition).
`comp.labels`	a vector of the same length as `plot.comp`, which provides the labels to be shown in the graph. default is to use the original component names. (relevant only for the linear decomposition).
`fill.colors`	colors to fill the areas. see `scale_fill_manual` for more details.
`trunc.negative`	whether to truncate negative values with 0. default is TRUE. this flag is necessary because `geom_area`, on which the function is based, does not work well with a combination of positive and negative values.
`...`	further arguments passed to or from other methods.

Value

A ggplot object containing all the relevant information for the plot.

`summary.vs_decomp` *Summary Method for 'vs_decomp' Objects*

Description

Summary Method for 'vs_decomp' Objects

Usage

```
## S3 method for class 'vs_decomp'
summary(object, r = 3, sum.comp = NULL, ...)
```

Arguments

`object`	a 'vs_decomp' object.
`r`	a number of decimal places to use.
`sum.comp`	a vector of up to 3 components to summarize. can be either character with component names or numeric with component indices. other components not specified by sum.comp are summed to one additional component. default is to summarize the first 3 components.
`...`	further arguments passed to or from other methods.

Value

No return value.

`vs_decomp` *Skewness and Variance Decomposition*

Description

decompose the skewness or the variance of an outcome vector into independent components, either using one or many variables.

Usage

```
vs_decomp(
  y = NULL,
  X,
  wgt = rep(1, nrow(X)),
  moment = "skewness",
  year = rep(1, nrow(X))
)
```

Arguments

`y`	an outcome vector (necessary only when decomposing by one variable).
`X`	when decomposing by one variable, a matrix or data frame containing the variable for the decomposition. when decomposing by more than one variable, a matrix or data frame containing the linear components of y, calculated by `linear_projection`.
`wgt`	an optional vector of weights.
`moment`	the moment on which the decomposition method is applied. either "variance" (second moment) or "skewness" (third moment).
`year`	an optional vector of years. if provided, the decomposition is calculated by year. otherwise, the decomposition is calculated for the whole sample.

Value

A "vs_decomp" object. This object is a list containing the estimated components and their standard errors.

Examples

```
#generate data
n <- 100
men <- rbinom(n, 1, 0.5)
black <- rbinom(n, 1, 0.5)
year <- c(rep(2019, n/2), rep(2020, n/2))
rwage <- function(x){
  m <- x[1]; b <- x[2]; y <- x[3];
  exp(rnorm(1, mean = 1*m + 1*b, sd = 1 + 2020 - y))
}
dat <- data.frame(men, black, year)
dat$wage <- apply(dat, 1, rwage)
#skewness decomposition by one variable:
variable <- as.matrix(dat$men)
decomp_by_gender <- vs_decomp(y = dat$wage, X = variable,
                              moment = "skew", year = dat$year)
summary(decomp_by_gender)
plot(decomp_by_gender)

#skewness decomposition by more than one variable (=linear skewness decomposition):
#first we need to decompose yearly wages into their linear components
#(check ?linear_projection for more details):
wage_linear_comp <- linear_projection(y = "wage",
                                      X.list = list("men", "black"),
                                      data = dat, year = dat$year)
#using the linear components, we can calculate skewness decomposition:
decomp_by_mult_var <- vs_decomp(X = wage_linear_comp,
                                year = dat$year)
#up to 3 components can be summarized or plotted:
colnames(decomp_by_mult_var$components)
#let's take "3cov(epsilon^2,men)" and "3cov(epsilon^2,black)"
summary(decomp_by_mult_var, sum.comp = c(7, 8))
```

```
plot(decomp_by_mult_var, plot.comp = c(7, 8))
```

`wtd_skew` *Weighted Skewness*

Description

computes weighted version of the skewness estimator.

Usage

```
wtd_skew(x, wgt = rep(1, length(x)))
```

Arguments

`x`	a numeric vector
`wgt`	an optional vector of weights.

Value

scalar

Index