Package ‘codaredistlm’

December 22, 2022

Type Package

Title Compositional Data Linear Models with Composition Redistribution

Version 0.1.0

Date 2022-12-22

Description Provided data containing an outcome variable, compositional variables and additional covariates (optional); linearly regress the outcome variable on an isometric log ratio (ilr) transformation of the linearly dependent compositional variables. The package provides predictions (with confidence intervals) in the change (delta) in the outcome/response variable based on the multiple linear regression model and evenly spaced reallocations of the compositional values. The compositional data analysis approach implemented is outlined in Dumuid et al. (2017a) <doi:10.1177/0962280217710835> and Dumuid et al. (2017b) <doi:10.1177/0962280217737805>.

License GPL-2

URL `https://github.com/tystan/codaredistlm`

BugReports `https://github.com/tystan/codaredistlm/issues`

Encoding UTF-8

LazyData true

Imports compositions, ggplot2, broom, knitr

Suggests testthat

RoxygenNote 7.1.2

NeedsCompilation no

Author Ty Stanford [aut, cre] (<`https://orcid.org/0000-0002-8570-5493`>),
Charlotte Lund Rasmussen [aut]
(<`https://orcid.org/0000-0003-2193-5019`>),
Dot Dumuid [aut] (<`https://orcid.org/0000-0003-3057-0963`>)

Maintainer Ty Stanford <`tystan@gmail.com`>

Repository CRAN

Date/Publication 2022-12-22 19:50:06 UTC

R topics documented:

append_ilr_coords *Add ILR coordinates to a data.frame containing composition variables*

Description

Add ILR coordinates to a data.frame containing composition variables

Usage

```
append_ilr_coords(dataf, comps, psi)
```

Arguments

`dataf`	data.frame containing composition variables
`comps`	character vector of composition variable names in dataf
`psi`	ilrBase passed to `compositions::ilr()`

check_input_args *Sanity checks for arguments passed to predict_delta_comps()*

Description

Sanity checks for arguments passed to predict_delta_comps()

Usage

```
check_input_args(dataf, y, comps, covars, deltas)
```

Arguments

`dataf` A `data.frame` containing data

`y` Name (as string/character vector of length 1) of outcome variable in `dataf`

`comps` Character vector of names of compositions in `dataf`. See details for more information.

`covars` Character vector of covariates names (non-comp variables) in `dataf` or NULL for none (default).

`deltas` A vector of time-component changes (as proportions of compositions , i.e., values between -1 and 1). Optional.

Details

Throws errors for any problematic input. Returns `TRUE` invisibly if no issues found.

check_strictly_positive_vals *Check if compositional variable are strictly greater than 0*

Description

Check if compositional variable are strictly greater than 0

Usage

```
check_strictly_positive_vals(dataf, comps, tol = 1e-06)
```

Arguments

`dataf` data.frame containing composition variables

`comps` character vector of composition variable names in dataf

`tol` a numeric value that compositional values are expected to be greater or equal than. 1e-6 is deafult

Value

If any compositional values are found to be strictly less than `tol` and erro is thrown. Returns `TRUE` invisibly otherwise.

`cols_exist` *Check whether columns exist in a data.frame*

Description

Check whether columns exist in a data.frame

Usage

```
cols_exist(dataf, cols)
```

Arguments

`dataf`	a data.frame
`cols`	character vector of columns to be checked in `dataf`

Value

An error if all `cols` not present in `dataf`. Returns `TRUE` invisibly otherwise.

`compare_two_lm` *Statistical test of the collective significance of the ilr variables*

Description

Statistical test of the collective significance of the ilr variables

Usage

```
compare_two_lm(y_str, X1, X2)
```

Arguments

`y_str`	a string representation of the column in `X1` (and `X2`) that is the outcome
`X1`	a data.frame or matrix that contains a subset of the predictor variables in `X2` and outcome variable
`X2`	a data.frame or matrix that contains the predictor variables and outcome variable

Value

Returns `NULL` invisibly. The ANOVA analysis is printed to the console, that is, the statistical test of whether the additional predictors in `X2` improve the model significantly from the model with only the subset of predictors in `X1`.

`create_comparison_matrix` *Creates row-wise perturbations of compositions from the mean composition*

Description

Creates row-wise perturbations of compositions from the mean composition

Usage

```
create_comparison_matrix(comparisons, comps, mean_comps)
```

Arguments

`comparisons`	currently two choices: `"one-v-one"` or `"prop-realloc"` (default).
`comps`	the names (character vector) of the compositional variables
`mean_comps`	the mean composition of `comps`

Details

`comparisons = "one-v-one"` creates a matrix with `length(comps)` columns and `length(comps) * (length(comps) - 1)` rows. The rows contain all pairs of variables with 1 and -1 values.

`comparisons = "prop-realloc"` creates a matrix with `length(comps)` columns and `length(comps)` rows. Each rows contains a 1 value for a compositional variable and the remaining values sum to -1 proportional to the `mean_comps` value for those variables.

Note that for both `comparisons` options the net change is 0 (each row sums to 0).

`create_v_mat` *Create ilr basis matrix (V)*

Description

Create ilr basis matrix (V)

Usage

```
create_v_mat(n_comp)
```

Arguments

`n_comp`	the number of compositional variables

Value

A `n_comp` by `n_comp - 1` matrix where each column relates to one ilr variable

The ilr basis made so that the numerator (`+` values) for the `i`th column is in the `i`th row. All values below the `+` value in the column are set to `-1` (the denominator).

The ilr basis for 3 compositional vars is `(2, -1, -1)/sqrt(6), (0, 1, -1)/sqrt(2)`.

The ilr basis for 4 comp vars is `(3, -1, -1, -1)/sqrt(12), (0, 2, -1, -1)/sqrt(6), (0, 0, 1, -1)/sqrt(2)`.

etc

`extract_lm_quantities` *Extract critical quantities from a lm object (for confidence interval calculations)*

Description

Extract critical quantities from a lm object (for confidence interval calculations)

Usage

```
extract_lm_quantities(lm_X, alpha = 0.05)
```

Arguments

`lm_X`	a lm object
`alpha`	level of significance. Defaults to 0.05.

Value

A list containing the `lm` model matrix (`dmX`), the inverse of `t(dmX) x dmX` (`XtX_inv`), the standard error (`s_e`), the estimated single column beta matrix (`beta_hat`), and the critical value of the relevant degrees of freedom t-dist (`crit_val`).

`fairclough` *Data from Fairclough (2017). Fitness, fatness and the reallocation of time between children's daily movement behaviours: an analysis of compositional data*

Description

A dataset containing z_bmi (outcome), time-use compositions (sl,sb,lpa,mvpa), and covariates from the Fairclough (2017) paper. The data can be found in supp file 7 of the paper at https://link.springer.com/article/10.1186/s129
017-0521-z.

Usage

```
data(fairclough)
```

Format

A data frame with 169 rows and 21 variables

Details

The variables in the data are as follows:

- child_id
- school
- sex
- decimal_age
- imd_decile
- height mass
- bmi
- z_bmi
- itof_grade
- waist_circ
- whtr
- shuttles_20m
- wear_time
- sed
- lpa
- mpa
- vpa
- mvpa
- sleep
- min_in_day

References

Fairclough, Stuart J. and Dumuid, Dorothea and Taylor, Sarah and Curry, Whitney and McGrane, Bronagh and Stratton, Gareth and Maher, Carol and Olds, Timothy. Fitness, fatness and the reallocation of time between children's daily movement behaviours: an analysis of compositional data. International Journal of Behavioral Nutrition and Physical Activity, 2017. 14(1): 64.

fat_data *Randomly generated data to simulate child fat percentage regressed on time-use compositional data*

Description

A dataset containing fat percentage (outcome), time-use compositions (sl,sb,lpa,mvpa), and covariates (sibs,parents,ed). Note sl+sb+lpa+mvpa=1440 minutes for each subject. The variables are as follows:

Usage

```
data(fat_data)
```

Format

A data frame with 100 rows and 8 variables

Details

- fat. child fat percentage (11.29–29.99)
- sl. daily sleep in minutes (283–765)
- sb. sedentary behaviour in minutes (354–789)
- lpa. low-intensity physical activity in minutes (157–507)
- mvpa. moderate- to vigorous-intensity physical activity in minutes (35–155)
- sibs. number of siblings (0,1,2,3,4)
- parents. number of parents/caregivers at home (1,2)
- ed. education level of parent(s) (0=high school, 1=diploma, 2=degree)

fit_lm *fit linear model based on input data.frame*

Description

fit linear model based on input data.frame

Usage

```
fit_lm(y_str, X, verbose = TRUE)
```

Arguments

`y_str`	a string representation of the column in `X` that is the outcome
`X`	a data.frame or matrix that contains the predictor and outcome variables
`verbose`	if `TRUE` (default), a model summary will be printed to the console

Value

A `lm` object where the `y_str` column has been regressed against the remaining columns of `X` (with an intercept term as well).

`is_deltacomp_obj` *Is object that is returned from* `pred_delta_comps()`*?*

Description

Is object that is returned from `pred_delta_comps()`?

Usage

```
is_deltacomp_obj(x)
```

Arguments

`x`	object to be tested

Value

Boolean TRUE or FALSE

`is_lm_mod` *Is object that is returned from* `lm()`*?*

Description

Is object that is returned from `lm()`?

Usage

```
is_lm_mod(x)
```

Arguments

`x`	object to be tested

Value

Boolean TRUE or FALSE

is_null_or_na *Catch NULL, empty and objects containing NAs*

Description

Catch NULL, empty and objects containing NAs

Usage

```
is_null_or_na(x)
```

Arguments

`x`	object to be tested

Value

Boolean. If object is NULL, empty or contains NA then TRUE returned. FALSE otherwise.

plot_delta_comp *Plot redistributed time-use predictions from compositional ilr multiple linear regression model fit*

Description

Plot redistributed time-use predictions from compositional ilr multiple linear regression model fit by `predict_delta_comps()`

Usage

```
plot_delta_comp(dc_obj, comp_total = NULL, units_lab = NULL)
```

Arguments

`dc_obj`	A `deltacomp_obj` object returned from the function `predict_delta_comps`
`comp_total`	A numeric scalar that is the original units of the composition to make the x-axis the original scale instead of in the range `[min(delta), max(delta)]` in (-1, 1).
`units_lab`	Character string of the units of the compositions relating to `comp_total` to add to the x-axis label

Value

Returns a plot object from the `ggplot2` package (that is, class of `gg` and `ggplot`).

Author(s)

Ty Stanford <tystan@gmail.com>

Examples

```
data(fairclough)

deltacomp_df <-
  predict_delta_comps(
    dataf = fairclough,
    y = "z_bmi",
    comps = c("sleep","sed","lpa","mvpa"),
    covars = c("decimal_age","sex"),
    deltas =  seq(-20, 20, by = 5) / (24 * 60),
    comparisons = "prop-realloc",
    alpha = 0.05
  )
class(deltacomp_df)

plot_delta_comp(
  dc_obj = deltacomp_df,
  comp_total = 24 * 60,
  units_lab = "min"
)

deltacomp_df <-
  predict_delta_comps(
    dataf = fairclough,
    y = "z_bmi",
    comps = c("sleep","sed","lpa","mvpa"),
    covars = c("decimal_age","sex"),
    deltas =  seq(-20, 20, by = 5) / (24 * 60),
    comparisons = "one-v-one",
    alpha = 0.05
  )

plot_delta_comp(
  dc_obj = deltacomp_df,
  comp_total = 24 * 60,
  units_lab = "min"
)
```

---

predict_delta_comps *Get predictions from compositional ilr multiple linear regression model*

Description

Provided the data (containing outcome, compositional components and covariates), fit a ilr multiple linear regression model and provide predictions from reallocating compositional values pairwise amunsnst the components model.

Usage

```
predict_delta_comps(
  dataf,
  y,
  comps,
  covars = NULL,
  deltas = c(0, 10, 20)/(24 * 60),
  comparisons = c("prop-realloc", "one-v-one")[1],
  alpha = 0.05
)
```

Arguments

`dataf` A `data.frame` containing data

`y` Name (as string/character vector of length 1) of outcome variable in `dataf`

`comps` Character vector of names of compositions in `dataf`. See details for more information.

`covars` Optional. Character vector of covariates names (non-comp variables) in `dataf`. Defaults to NULL.

`deltas` A vector of time-component changes (as proportions of compositions , i.e., values between -1 and 1). Optional. Changes in compositions to be computed pairwise. Defaults to 0, 10 and 20 minutes as a proportion of the 1440 minutes in a day (i.e., approximately `0.000`, `0.007` and `0.014`).

`comparisons` Currently two choices: `"one-v-one"` or `"prop-realloc"` (default). Please see details for explanation of these methods.

`alpha` Optional. Level of significance. Defaults to 0.05.

Details

Values in the `comps` columns must be strictly greater than zero. These compositional values are NOT assumed to be constrained to (0, 1) values as the function normalises the compositions row-wise to sum to 1 in part of it's processing of the dataset before analysis.

Please see the `deltacomp` package `README.md` file for examples and explanation of the `comparisons = "prop-realloc"` and `comparisons = "one-v-one"` options.

Value

Messages are printed to the console as the function tests the inputs, produces the isometric log ratios (ilrs), fits the linear model and produces the redistributed time-use predictions (with confidence intervals).

Returns a `data.frame` of the time-use redistribution predictions (and 95% confidence intervals) with the following columns:

- `comp+`: the compositional variable with the addition of the `delta` value
- `comp-`: the compositional variable with the subtraction of the `delta` value
- `delta`: the time-use redistribution value

- alpha: significance level for the 100(1-alpha)% confidence interval
- delta_pred: the predicted mean change in the outcome variable
- ci_lo: the lower limit of 100(1-alpha)% confidence interval corresponding to delta_pred
- ci_up: the upper limit of 100(1-alpha)% confidence interval corresponding to delta_pred
- sig: "*" if the delta_pred is significantly different from 0 at the alpha level (empty string otherwise)

The data.frame has a class of deltacomp_obj which denotes there are additional attributes of the returned object accessible using attr(*, "attribute_name").

The possible values for "attribute_name" are:

- dataf: a data.frame of the predictors (covariates and ilrs)
- y: a vector of the outcome variable
- comps: a character vector of the time-use composition names
- lm: the lm object of the multiple linear regression fit (using y and dataf from above)
- deltas: the redistributed time-use values used in the predictions
- comparisons: "one-v-one" or "prop-realloc" provided as the comparisons argument
- alpha: significance level for the 100(1-alpha)% confidence intervals
- ilr_basis: the ilr change of basis matrix V
- mean_pred: a single row data.frame with the predicted mean outcome (fit column) value from the "average" set of predictors

Author(s)

Ty Stanford <tystan@gmail.com>

Examples

```
predict_delta_comps(
  dataf = fat_data,
  y = "fat",
  comps = c("sl", "sb", "lpa", "mvpa"),
  covars = c("sibs", "parents", "ed"),
  deltas = seq(-60, 60, by = 5) / (24 * 60),
  comparisons = "one-v-one",
  alpha = 0.05
)

delta_comp_out <- predict_delta_comps(
  dataf = fat_data,
  y = "fat",
  comps = c("sl", "sb", "lpa", "mvpa"),
  covars = NULL,
  deltas = seq(-60, 60, by = 5) / (24 * 60),
  comparisons = "prop-realloc",
  alpha = 0.05
)
```

```
# get the mean prediction from the returned object
attr(delta_comp_out, "mean_pred")
```

`print_ilr_trans` *Print the ilr transformation of provided composition parts to console*

Description

Print the ilr transformation of provided composition parts to console

Usage

```
print_ilr_trans(comps)
```

Arguments

`comps`	a character vector of compositional parts

Value

a character vector of representing the ilr transformation of the `comps` is returned invisibly as the function's purpose is simply to print to the R console

Index