Package ‘psycCleaning’

November 5, 2023

Type Package
Title Data Cleaning for Psychological Analyses
Version 0.1.1
Description Useful for preparing and cleaning data. It includes functions to center data, reverse coding, dummy code and effect code data, and more.
License GPL (>= 3)
Encoding UTF-8
RoxygenNote 7.2.3
Imports dplyr, tidyr, tibble, data.table, rlang (>= 0.1.2)
Suggests roxygen2, covr, misty, testthat (>= 3.0.0)
Config/testthat/edition 3
Depends R (>= 2.10)
LazyData true
NeedsCompilation no
Author Jason Moy [aut, cre] (<https://orcid.org/0000-0001-8795-3311>)
Maintainer Jason Moy <jason.moyhj@gmail.com>
Repository CRAN
Date/Publication 2023-11-05 06:30:02 UTC

R topics documented:

center_grand_mean .................................................. 2
center_group_mean .................................................. 3
center_mlm ......................................................... 3
composite_score .................................................... 4
dummy_coding ....................................................... 5
effect_coding ......................................................... 6
listwise_deletion .................................................. 6
mlbook_data ......................................................... 7
Description

This function will compute grand-mean-centered scores.

Usage

center_grand_mean(data, cols, keep_original = TRUE)

Arguments

data A data.frame or a data.frame extension (e.g. a tibble).
cols Columns that need to be centered. See `dplyr::dplyr_tidy_select` for available options.
keep_original default is 'FALSE'. Set to 'TRUE' to keep original columns

Value

An object of the same type as .data. The output has the following properties: 1. Columns from .data will be preserved 2. Columns with scores that are grand-mean-centered.

Examples

center_grand_mean(iris,where(is.numeric))
center_group_mean

Description

This function will compute group-mean-centered scores.

Usage

center_group_mean(data, cols, group, keep_original = TRUE)

Arguments

data A data.frame or a data.frame extension (e.g. a tibble).
cols Columns that need to be centered. See `dplyr::dplyr_tidy_select` for available options.
group character. grouping variable
keep_original default is ‘TRUE’. Set to ‘FALSE’ to remove original columns

Value

An object of the same type as .data. The output has the following properties: 1. Columns from .data will be preserved 2. Columns with scores that are group-mean centered

Examples

center_group_mean(iris, where(is.numeric), group = Species)

center_mlm

Description

This function will group mean centered the scores at the level 1 and create a mean score for each group at L2.

Usage

center_mlm(data, cols, group, keep_original = TRUE)
composite_score

Arguments

- data: A data.frame or a data.frame extension (e.g. a tibble).
- cols: Columns that need to be centered. See `dplyr::dplyr_tidy_select` for available options.
- group: the grouping variable. Must be character.
- keep_original: default is ‘TRUE’. Set to ‘FALSE’ to remove original columns

Value

An object of the same type as .data. The output has the following properties: 1. Columns from .data will be preserved 2. Columns with L1 scores that are group-mean centered. 3. Columns with L2 aggregated means.

Examples

```r
center_mlm(iris,dplyr::ends_with("Length"),group = "Species")
```

---

**composite_score**  
**Composite column**

Description

The function will perform a row-wise aggregation which then divided by the total number of columns.

Usage

```r
composite_score(
  data,
  cols = dplyr::everything(),
  na.rm = FALSE,
  composite_col_name = "composited_column"
)
```

Arguments

- data: A data.frame or a data.frame extension (e.g. a tibble).
- cols: Columns that need to be composited See `dplyr::dplyr_tidy_select` for available options.
- na.rm: Ignore NA. The default is ‘FALSE’. If set to ‘TRUE’, the composite score will be ‘NA’ if there is one or more ‘NA’ in any of the columns.
- composite_col_name: Name for the new composited columns. Default is ‘composite_column’.
**dummy_coding**

**Value**

An object of the same type as `.data`. The output has the following properties: 1. Columns from `.data` will be preserved. 2. Columns with composited scores.

**Examples**

```r
test_df = data.frame(col1 = c(1,2,3,4), col2 = c(1,2,3,4), col3 = c(1,2,NA,4))
composite_df = composite_score(data = test_df)
dummy_coding(iris, Species)
```
**effect_coding**  
*Effect Coding*

**Description**
Create effect-coded columns, supporting tidyselect syntax to process multiple columns simultaneously.

**Usage**
effect_coding(data, cols, factor = FALSE)

**Arguments**
- `data`: A data.frame or a data.frame extension (e.g., a tibble).
- `cols`: Columns that need to be effect-coded. See `dplyr::dplyr_tidy_select` for available options.
- `factor`: The default is `FALSE`. If factor is set to `TRUE`, this function returns a tibble with effect-coded factors. If factor is set to `FALSE`, this function returns a tibble with effect-coded columns.

**Value**
An object of the same type as `.data`. The output has the following properties: 1. Columns from `.data` will be preserved. 2. Columns that are effect-coded.

**Examples**
effect_coding(iris, Species)

---

**listwise_deletion**  
*Listwise deletion*

**Description**
Perform listwise deletion (the entire rows is disregarded if the row has one ‘NA’ value)

**Usage**
listwise_deletion(data, cols = dplyr::everything())

**Arguments**
- `data`: A data.frame or a data.frame extension (e.g., a tibble).
- `cols`: Columns that need to use listwise deletion. See `dplyr::dplyr_tidy_select` for available options.
**Value**

An object of the same type as .data with rows removed if the row has one `NA` value

**Examples**

```r
test_df = data.frame(col1 = c(1,2,3),col2 = c(1,NA,3),col3 = c(1,2,NA))
listwise_deletion(test_df,col1:col2) # you can see that the row with NA in col3 is not deleted
```

**Description**


**Usage**

```
mlbook_data
```

**Format**

A data frame with 3758 rows and 34 variables:

- `schoolnr` School ID
- `pupilNR_new` Student Identifier (Level 1 units)
- `langPOST` Student language score
- `ses` Student socioeconomic score, grand-mean centered (in points, M = 0))
- `IQ_verb` Student verbal IQ, grand-mean centered (in points, M = 0)
- `sex` Student binary gender, 1 = female, 0 = not female
- `Minority` Student minority status, 1 = minoritized, 0 = not minoritized
- `denomina` School-level religious denominations, 5 categories
- `female_dum` Dummy coded sex
- `female_eff` Effect-coded sex
- `female_CMC` Group-mean-centered of female_eff
- `fempct_agg` Aggregated mean female_dum for each school
- `Zfempct_agg` Z-scored aggregated mean female_dum for each school
- `ses_CMC` Group-mean-centered SES
- `Zses_CMC` Z-scored group-mean-centered SES
- `ses_agg` Aggregated mean SES for each school
- `Zses_agg` Z-scored aggregated mean SES for each school

**Source**

[https://www.stats.ox.ac.uk/~snijders/mlbook.htm](https://www.stats.ox.ac.uk/~snijders/mlbook.htm)
### recode_item

**Recode values of a data frame**

**Description**

Recode values of a data frame

**Usage**

```r
recode_item(data, cols, code_from = NULL, code_to = NULL, retain_code = NULL)
```

**Arguments**

- `data`: A data.frame or a data.frame extension (e.g. a tibble).
- `cols`: Columns that need to be recoded. See `dplyr::dplyr_tidy_select` for available options.
- `code_from`: vector. the order must match with vector for `code_to`
- `code_to`: vector. the order must match with vector for `code_from`
- `retain_code`: vector. Specify the values to be retain

**Value**

An object of the same type as `.data`. The output has the following properties:
1. Columns except the recoded columns from `.data` will be preserved
2. Recoded columns

**Examples**

```r
pre_recoded_df = tibble::tibble(x1 = 1:5, x2 = 5:1)
recoded_df = recode_item(pre_recoded_df, cols = dplyr::contains('x'),
                         code_from = 1:5,
                         code_to = 5:1)
```

---

### summarize_missing_values

**Count the number of missing values**

**Description**

It counts the number of missing (i.e., ‘NA’) values in each column.
z_scored_grand_mean

Usage

```r
summarize_missing_values(
  data,
  cols = dplyr::everything(),
  group = NULL,
  verbose = TRUE,
  return_result = FALSE
)
```

Arguments

- `data`: A data.frame or a data.frame extension (e.g. a tibble).
- `cols`: Columns that need to be checked for missing values. See `dplyr::dplyr_tidy_select` for available options.
- `group`: character. count missing values by group.
- `verbose`: default is `TRUE`. Print the missing value data frame.
- `return_result`: default is `FALSE`. Return `data_frame` if set to yes.

Value

An object of the same type as .data. that specified the number of NA values of the columns (only when `return_result = TRUE`).

Examples

```r
df1 = data.frame(col1 = c(1,2,3),col2 = c(1,NA,3),col3 = c(1,2,NA))
summarize_missing_values(df1,everything())
```

---

z_scored_grand_mean  Grand mean z-score

Description

This function will compute z-scores with respect to the grand mean.

Usage

```r
z_scored_grand_mean(data, cols, keep_original = TRUE)
```

Arguments

- `data`: A data.frame or a data.frame extension (e.g. a tibble).
- `cols`: Columns that need to be centered. See `dplyr::dplyr_tidy_select` for available options.
- `keep_original`: default is `FALSE`. Set to `TRUE` to keep original columns.
z_scored_group_mean

Value

An object of the same type as .data. The output has the following properties: 1. Columns from .data will be preserved 2. Columns with scores that are z-scored

Examples

```r
z_scored_grand_mean(iris, where(is.numeric))
```

Description

This function will compute group-mean-centered scores, and then z-scored the group-mean-centered scores with respect to the grand mean.

Usage

```r
z_scored_group_mean(data, cols, group, keep_original = TRUE)
```

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>data</td>
<td>A data.frame or a data.frame extension (e.g. a tibble).</td>
</tr>
<tr>
<td>cols</td>
<td>Columns that need to be centered. See <code>dplyr::dplyr_tidy_select</code> for available options.</td>
</tr>
<tr>
<td>group</td>
<td>the grouping variable. If you need to pass multiple group variables, try to use quos(). Passing multiple group variables is not tested.</td>
</tr>
<tr>
<td>keep_original</td>
<td>default is ‘FALSE’. Set to ‘TRUE’ to keep original columns</td>
</tr>
</tbody>
</table>

Value

return a dataframe with the columns z-scored (replace existing columns)

Examples

```r
z_scored_group_mean(iris, dplyr::ends_with("Petal.Width"), "Species")
```
**z_scored_mlm**

**Z-scored for multilevel analyses**

**Description**

This function will group mean centered the scores at the level 1 and create an aggregated mean score for each group at L2. After that, the group-mean-centered L1 scores and mean L2 scores will be z-scored with respect to the grand mean. Please see ‘center_mlm’ if you want to use the version without the z-scoring.

**Usage**

```r
z_scored_mlm(data, cols, group, keep_original = TRUE)
```

**Arguments**

- `data`: A data.frame or a data.frame extension (e.g. a tibble).
- `cols`: Columns that need to be centered. See ‘dplyr::dplyr_tidy_select’ for available options.
- `group`: The grouping/cluster variable.
- `keep_original`: default is ‘TRUE’. Set to ‘FALSE’ to remove original columns.

**Value**

An object of the same type as .data. The output has the following properties: 1. Columns from .data will be preserved 2. Columns with L1 scores that are group-mean centered then grand-mean z-scored. 3. Columns with L2 aggregated means that are z-scored.

**Examples**

```r
z_scored_mlm(iris, dplyr::ends_with(‘/quotesingle.VarLength/quotesingle.Var’), group = ‘/quotesingle.VarSpecies/quotesingle.Var’)
```

---

**z_scored_mlm_categorical**

**Z-scored for multilevel analyses**

**Description**

This is a specialized function for mean centering categorical variables. There are two cases where this function should be used instead of the generic ‘center_mlm’. 1. This function should be used when you need group mean centering for non-dummy-coded variables at L1. Variables at L2 are always dummy-coded as they represent the percentage of subjects in that group. 2. This function should be used whenever you want to z-score the aggregated L2 means.

---
Usage

z_scored_mlm_categorical(
  data,
  cols,
  dummy_coded = NA,
  group,
  keep_original = TRUE
)

Arguments

data  A data.frame or a data.frame extension (e.g. a tibble).
cols  Dummy-coded or effect-coded columns for group-mean centering. Support 'dplyr::dplyr_tidy_select' options.
dummy_coded  Dummy-coded variables (cannot be effect-coded) for L2 aggregated means. Support 'dplyr::dplyr_tidy_select' options.
group  the grouping variable. Must be character
keep_original  default is ‘FALSE’. Set to ‘TRUE’ to keep original columns

Value

An object of the same type as .data. The output has the following properties: 1. Columns from .data will be preserved 2. Columns with L1 scores that are group-mean centered 3. Columns with L2 aggregated means (i.e., percentage) that are z-scored

Examples

z_scored_mlm_categorical(mlbook_data, cols = 'female_eff', dummy_coded = 'female_dum', 'schoolnr')
Index

* datasets
  mlbook_data, 7

center_grand_mean, 2
center_group_mean, 3
center_mlm, 3
composite_score, 4
dummy_coding, 5
effect_coding, 6
listwise_deletion, 6
mlbook_data, 7
recode_item, 8
summarize_missing_values, 8
z_scored_grand_mean, 9
z_scored_group_mean, 10
z_scored_mlm, 11
z_scored_mlm_categorical, 11