# Package 'tidymodlr'

August 26, 2024

**Title** An R6 Class to Perform Analysis on Long Tidy Data

**Version** 1.0.0

**Author** David Hammond [aut, cre]

**URL** https://github.com/david-hammond/tidymodlr

**BugReports** https://github.com/david-hammond/tidymodlr/issues

**Maintainer** David Hammond <anotherdavidhammond@gmail.com>

**Description** Transforms long data into a matrix form to allow for ease of input into modelling packages for regression, principal components, imputation or machine learning. It does this by pivoting on user defined columns, generating a key-value table for variable names to ensure one-to-one mappings are preserved. It is particularly useful when the indicator names in the columns are long descriptive strings, for example ``Energy imports, net (% of energy use)''. High level analysis wrapper functions for correlation and principal components analysis are provided.

**License** MIT + file LICENSE

**Suggests** testthat (>= 3.0.0)

**Config/testthat/edition** 3

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**Imports** R6, dplyr, tidyr, tm, corrr, FactoMineR

**Depends** R (>= 2.10)

**LazyData** true

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2024-08-26 12:30:02 UTC

## Contents

---

`tidymodlr-package` `tidymodlr`*: Modelling with tidy long data*

---

## Description

`tidymodlr` transforms long data into a matrix form to allow for ease of input into modelling packages for regression, principal components, imputation or machine learning.

## Details

In many fields it is common to have data in tidy long data, with the rows representing many variables, but only **one** column representing the values (see `?wb` for an example).

`tidymodlr` is particularly useful when the indicator names in the columns are long descriptive strings, for example 'Energy imports, net (% of energy use)'. In such cases a straight `pivot wider` generates column names that are not only cumbersome, but also generate errors in many standard modelling packages that require `base` column names.

High level analysis functions for correlation, imputation and principals components analysis are provided.

## Author(s)

**Maintainer**: David Hammond <anotherdavidhammond@gmail.com>

## See Also

Useful links:

- https://github.com/david-hammond/tidymodlr
- Report bugs at https://github.com/david-hammond/tidymodlr/issues

---

`make_key_value` *Generate a key value table with unique key for a set of text*

---

## Description

Given a vector of characters, this will return a data frame of a unique `key` column (of, where possible, 3 characters) and `value` column listing the unique elements of the original `text`.

## Usage

```
make_key_value(text)
```

## Arguments

| | |
|---|---|
| `text` | The text to abbreviate and create a key value table for |

### Value

df A `Key Value` table

### Examples

```
data(wb)
make_key_value(wb$indicator)
```

---

`tidymodl` *Creates a model matrix style R6 class for modelling with long tidy data*

---

### Description

Creates a model matrix style R6 class for modelling with long tidy data

### Public fields

`data (data.frame())`
: The original tidy long data frame

`parent (data.frame())`
: The parent identifiers of the original data

`child (data.frame())`
: The model matrix version of the data

`key (data.frame())`
: A `key value` table that links the parent and child data.frames.

### Methods

#### Public methods:

- `tidymodl$new()`
- `tidymodl$assemble()`
- `tidymodl$print()`
- `tidymodl$correlate()`
- `tidymodl$pca()`
- `tidymodl$clone()`

**Method** `new()`**:** Creates a new instance of this R6 class.

Create a new tidymodl object.

*Usage:*

```
tidymodl$new(df, pivot_column, pivot_value)
```

*Arguments:*

`df` A tidy long data frame

`pivot_column` The column name on which the pivot will occur

`pivot_value` The column name of the values to be pivotted

*Returns:* A new `tidymodl` object.

**Method** `assemble()`**:** Adds a results matrix

*Usage:*

```
tidymodl$assemble(newdata, format = "long")
```

*Arguments:*

`newdata` A new data set to append. Needs to be either:

- A vector of length equal to the number of rows in the model matrix. For example, the output of `predict()` of a `lm` model. In this case the function returns a data.frame of dimensions `c(nrow(parent), ncol(parent) + 1)`
- A data.frame/matrix of equal dimensions of the model matrix. For example, the output of `xgb_impute()`. In this case the function returns a data.frame of dimensions `c(nrow(data), ncol(data) + 1)`

`format` The desired format of the returned data frame, can either be "long" or "wide".

*Details:* This returns a completed data.frame for four use cases based on user preference of the desired format.

- **Format "long":**
  - **Use Case 1 - "newdata" is a vector of length nrow(child):** The function returns a combined data frame of the parent data and the "newdata" in a new column. Useful when the user wants to append an output of, for example, `predict` for a `lm` regression model.
  - **Use Case 2 - "newdata" is a matrix of dimensions dim(child):** The function returns a data.frame of the original data in long format with the "newdata" in a new column. Useful when the user wants to append an output of, for example, `xgb_impute` for all original data.
- **Format "wide":**
  - **Use Case 3 - "newdata" is a vector of length nrow(child):** The function returns a combined data frame of the parent data and the "newdata" in a new column. Useful when the user wants to append an output of, for example, `predict` for a `lm` regression model.
  - **Use Case 4 - "newdata" is a matrix of dimensions dim(child):** The function returns a data.frame of the original data in wide format with the "newdata" as replacing the child matrix of the original data. Useful when the user *is only* interested in using the output of, for example, `xgb_impute` for all original data.

*Returns:* df A Data Frame

**Method** `print()`**:** Prints the key and the head matrix

*Usage:*

```
tidymodl$print()
```

**Method** `correlate()`**:** Correlates and reutrns pearson values

*Usage:*

```
tidymodl$correlate()
```

*Returns:* df A Correlation Matrix of class `cor_df` (see corrr)

**Method** `pca()`**:** Provides high level principal components analysis

*Usage:*

```
tidymodl$pca()
```

*Returns:* df A principle components of class PCA (see FactoMineR

**Method** `clone()`**:** The objects of this class are cloneable with this method.

*Usage:*

```
tidymodl$clone(deep = FALSE)
```

*Arguments:*

deep Whether to make a deep clone.

## Note

Use Cases 1 and 3 return identical results.

## Examples

```
data(wb)
mdl <- tidymodl$new(wb,
                    pivot_column = "indicator",
                   pivot_value = "value")
### Use mdl$child for modelling
fit <- lm(data = mdl$child, gni ~ gcu + ppt)

### Can be used to add a yhat value for processed data

nc <- ncol(mdl$child)
nr <- nrow(mdl$child)
dm <- nc * nr
dummy <- matrix(runif(dm),
                ncol = nc) |>
                data.frame()
names(dummy) = names(mdl$child)
tmp <- mdl$assemble(dummy)

# In built correlation function
mdl$correlate()

tmp <- mdl$pca()
plot(tmp, choix = "var")
```

---

wb *Dummy Long Tidy Data*

---

## Description

A dataset from the World Bank of a dummy data. The variables are as follows:

## Usage

```
data(wb)
```

## Format

A data frame with 975 rows and 4 variables

- **iso3c:** isocode of a country.
- **indicator:** World Bank indicator.
- **year:** Year of observation
- **value:** Value of observation

# Index