

# Package ‘vvsculptor’

July 22, 2025

**Title** Apply Manipulations to Data Frames

**Version** 0.4.10

## Description

Provides a set of functions for manipulating data frames in accordance with specific business rules. In addition, it includes wrapper functions for commonly used functions from the popular 'tidyverse' package, making it easy to integrate these functions into data analysis workflows. The package is designed to streamline data preprocessing and help users quickly and efficiently perform data transformations that are specific to their business needs.

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.2.3

**Imports** dplyr

**NeedsCompilation** no

**Author** Tomer Iwan [aut, cre, cph]

**Maintainer** Tomer Iwan <t.iwan@vu.nl>

**Repository** CRAN

**Date/Publication** 2023-01-30 16:20:02 UTC

## Contents

correct_model_levels . . . . .	2
strict_left_join . . . . .	2
<b>Index</b>	<b>4</b>

---

correct\_model\_levels     *Correct model levels*

---

### Description

Correct level names for modelling and in the use of ROC curve/AUC.

### Usage

```
correct_model_levels(data)
```

### Arguments

data                      String with level names.

### Value

Corrected level names.

### Examples

```
data <- data.frame(id = c(1,2,3),
  name = c("Alice", "Bob", "Charlie"),
  gender = factor(c("Female", "Male", "Female"), levels = c("Female", "Male")))

correct_model_levels(data)
# returns a data frame with factor levels of the variable gender corrected to "Female" and "Male"

data <- data.frame(id = c(1,2,3),
  name = c("Alice", "Bob", "Charlie"),
  gender = factor(c("Female", "Male", "Female")))
correct_model_levels(data)
# returns a data frame with factor levels of the variable gender corrected to "F" and "M"
```

---

strict\_left\_join             *strict\_left\_join*

---

### Description

A wrapper around dplyr's left\_join, with an error message if duplicate values are present in the matching fields in y. This will prevent duplicating rows. See dplyr::left\_join .

### Usage

```
strict_left_join(x, y, by = NULL, ...)
```

**Arguments**

<code>x</code>	data frame x (left)
<code>y</code>	data frame y (right)
<code>by</code>	unquoted variable names to join.
<code>...</code>	Pass further arguments to <code>dplyr::left_join</code>

**Value**

merged data frame

**See Also**

[left\\_join](#)

**Examples**

```
left_df <- data.frame(id = c(1, 2, 3), name = c("Alice", "Bob", "Charlie"))
right_df <- data.frame(id = c(1, 2, 4), age = c(20, 25, 30))
strict_left_join(left_df, right_df, by = "id")
```

# Index

`correct_model_levels`, [2](#)

`left_join`, [3](#)

`strict_left_join`, [2](#)