

# Package: zutils (via r-universe)

March 21, 2025

**Title** Zhang Liang's miscellaneous utilities

**Version** 0.0.10

**Description** This package mainly contains some miscellaneous utilities that I use frequently.

**License** MIT + file LICENSE

**URL** <https://psychelzh.github.io/zutils/>

**Imports** dplyr, rlang, stringr, tidyr, tidyselect, utils

**Suggests** testthat (>= 3.0.0)

**Config/testthat/edition** 3

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.1

**Config/pak/sysreqs** libicu-dev

**Repository** <https://psychelzh.r-universe.dev>

**RemoteUrl** <https://github.com/psychelzh/zutils>

**RemoteRef** HEAD

**RemoteSha** 436966e7b43eeb9ef0f58c37c7e05d73832ee06d

## Contents

call_full . . . . .	2
cautiously . . . . .	2
select_list . . . . .	3
separate_wider_dsv . . . . .	3

<b>Index</b>	<b>5</b>
--------------	----------

---

call_full	<i>Create a call to a function with its arguments</i>
-----------	---

---

**Description**

This is basically a wrapper around `rlang::call2()` that allows you to extract the arguments from a function and pass them to `rlang::call2()` without having to type them out.

**Usage**

```
call_full(.fn, ...)
```

**Arguments**

<code>.fn</code>	The function to call.
<code>...</code>	Arguments to pass to the function.

**Value**

A call to the function with its arguments.

---

cautiously	<i>Cautiously evaluate an expression</i>
------------	--

---

**Description**

This function is useful for when you want to evaluate an expression, but you want to catch any errors and return a default value instead. Note the error message will be printed as a warning as the name of the function suggests.

**Usage**

```
cautiously(.f, otherwise = NULL)
```

**Arguments**

<code>.f</code>	A function to modify. See <code>as_function()</code> for details.
<code>otherwise</code>	A value to return if the expression throws an error.

**Value**

A function that evaluates the expression and returns the result or the default value if an error is thrown.

---

select_list	<i>Tidy select for list</i>
-------------	-----------------------------

---

### Description

A tidy select interface for lists. See `tidyselect::eval_select()` for details.

### Usage

```
select_list(.l, ...)
```

### Arguments

`.l` A `list()` object.  
`...` One or more unquoted expressions separated by commas.

### Value

A list with the selected elements.

---

separate_wider_dsv	<i>Separate a column into multiple columns</i>
--------------------	--

---

### Description

This is a wrapper around `tidyr::separate_wider_regex()` that allows to split a column into multiple columns. The column contains so-called delimiter separated values (DSV) and the values are extracted using regular expressions.

### Usage

```
separate_wider_dsv(  
  data,  
  col,  
  names,  
  ...,  
  patterns = NULL,  
  delim = "_",  
  prefix = NULL,  
  suffix = NULL  
)
```

**Arguments**

<code>data</code>	A data frame.
<code>col</code>	<tidy-select> Column to separate.
<code>names</code>	Names of the new columns. Use NA if you want a component not to be in the output.
<code>...</code>	Additional arguments passed to <code>tidyr::separate_wider_regex()</code> .
<code>patterns</code>	Regular expressions to extract the values from the column. If NULL, the pattern will match any character non-greedily. The length of the vector must be equal to the number of new columns.
<code>delim</code>	Delimiter used in the column to separate different pieces of values.
<code>prefix, suffix</code>	Prefix and suffix to be removed from the target column to retrieve the values.

**Value**

A data frame with the separated columns.

# Index

`as_function()`, 2

`call_full`, 2

`cautiously`, 2

`list()`, 3

`rlang::call2()`, 2

`select_list`, 3

`separate_wider_dsv`, 3

`tidyr::separate_wider_regex()`, 3, 4

`tidyselect::eval_select()`, 3