

# Package ‘listless’

August 29, 2016

**Type** Package

**Title** Convert Lists to Tidy Data Frames

**Version** 0.0-2

**Date** 2016-08-11

**Maintainer** Richard Cotton <richierocks@gmail.com>

**Depends** R (>= 3.0.0)

**Imports** magrittr (>= 1.5), tidyr (>= 0.5.1)

**Description** A lightweight utility for converting lists to tidy data frames.

**License** GPL-3

**RoxygenNote** 5.0.1

**NeedsCompilation** no

**Author** Richard Cotton [cre, aut]

**Repository** CRAN

**Date/Publication** 2016-08-12 12:39:49

## R topics documented:

list_depth . . . . .	1
list_str . . . . .	2
list_to_data.frame . . . . .	3

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

---

list_depth	<i>Get the depth of a list</i>
------------	--------------------------------

---

## Description

Gets the depth of a list (at its deepest point).

**Usage**

```
list_depth(l, prune_empty_elts = FALSE)
```

**Arguments**

`l` A variable, probably a list.

`prune_empty_elts` A logical value. Should empty elements be pruned without counting them?

**Value**

A non-negative integer of the deepest depth of the list.

**See Also**

This is loosely based upon `list.depth` (internal to older versions of the Zelig package), but gives different answers.

**Examples**

```
list_depth(list(1))
list_depth(list(1, list(2:3, 4:6)))

# Atomic variables have depth 0
list_depth(1)

# Empty elements can be pruned before counting
list_depth(list())
list_depth(list(), prune_empty_elts = TRUE)
```

---

list\_str

*Summarise the structure of a list*

---

**Description**

Summarise the structure of a list, in a tidy data frame.

**Usage**

```
list_str(l, name_variables = "names",
         stringsAsFactors = getOption("stringsAsFactors"))
```

**Arguments**

`l` A variable, probably a list.

`name_variables` A string. What should the columns formed from the names of `l` be called?

`stringsAsFactors` Should character columns be converted to factors?

**Value**

`data.frame`.

**See Also**

`str`, `list_to_data.frame`

**Examples**

```
l <- list(
  a = 1,
  b = matrix(1:6, 2),
  c = list(
    ca = y ~ x,
    list(cba = median, cbb = quote(1 + 1), cbc = expression(1 + 1)),
    cc = list(cca = as.name("xyz"))
  ),
  d = array(1:24, 2:4)
)
list_str(l)
```

---

<code>list_to_data.frame</code>	<i>Convert a list to a data frame</i>
---------------------------------	---------------------------------------

---

**Description**

Converts a list to a data frame, with names in multiple columns.

**Usage**

```
list_to_data.frame(l, name_variables = "names", values_variable = "values",
  stringsAsFactors = getOption("stringsAsFactors"))
```

**Arguments**

<code>l</code>	A variable, probably a list.
<code>name_variables</code>	A string. What should the columns formed from the names of <code>l</code> be called?
<code>values_variable</code>	A string. What should the columns formed from the values of <code>l</code> be called?
<code>stringsAsFactors</code>	Should character columns be converted to factors?

**Value**

`data.frame`.

**Examples**

```
(l <- list(
  a = 1,
  2:3,                                # missing names are blank
  c = list(ca = 4:6, 7:10, list(cca = 11:15)),
  d = list()                          # empty elt's silently ignored
))
list_to_data.frame(l)

# Custom column names
list_to_data.frame(l, c("group", "subgroup", "element"), "amount")
```

# Index

`data.frame`, 3

`list_depth`, 1

`list_str`, 2

`list_to_data.frame`, 3, 3

`str`, 3