Package 'lavaan.printer'

September 15, 2024

Title Helper Functions for Printing 'lavaan' Outputs

Version 0.1.0

Description Helpers for customizing selected outputs from 'lavaan' by Rosseel (2012) <doi:10.18637/jss.v048.i02> and print them. The functions are intended to be used by package developers in their packages and so are not designed to be user-friendly. They are designed to be let developers customize the tables by other functions. Currently the parameter estimates tables of a fitted object are supported.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.2

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Imports utils, methods, lavaan

Depends R (>= 4.0.0)

Config/testthat/edition 3

URL `https://sfcheung.github.io/lavaan.printer/`

VignetteBuilder knitr

NeedsCompilation no

Author Shu Fai Cheung [aut, cre] (<`https://orcid.org/0000-0002-9871-9448`>)

Maintainer Shu Fai Cheung <`shufai.cheung@gmail.com`>

Repository CRAN

Date/Publication 2024-09-15 20:00:02 UTC

Contents

parameterEstimates_table_list

Convert an Estimates Table to a List of Data Frames

Description

Create a list of data frames from the output of `lavaan::parameterEstimates()`, formatted in nearly the same way the argument `output = "text"` does.

Usage

```
parameterEstimates_table_list(
  object,
  ...,
  fit_object = NULL,
  se_also_to_na = character(0),
  se_not_to_na = character(0),
  drop_cols = "std.nox",
  rename_cols = character(0),
  est_funs = list(),
  header_funs = list(),
  footer_funs = list(),
  est_funs_args = NULL,
  header_funs_args = NULL,
  footer_funs_args = NULL
)

print_parameterEstimates_table_list(
  x,
  nd = 3,
  by_group = TRUE,
  drop_cols = character(0),
  na_str = " "
)
```

Arguments

`object`	It can a data frame similar in form to the output of `lavaan::parameterEstimates()`, or a `lavaan` object (e.g., the output of `lavaan::sem()`). If it is a `lavaan` object, then `lavaan::parameterEstimates()` will be called to generate the parameter estimates table.
`...`	If `object` is a `lavaan` object, then these are the optional arguments to be passed to `lavaan::parameterEstimates()` when it is called.
`fit_object`	(Optional). The `lavaan` object for getting additional information, if they are not available in `object`, and added as attributes to `object`. It essentially does what `lavaan::parameterEstimates()` does when setting `output` to `"text"`.

`se_also_to_na`	Columns for which cells will be set to `NA` if the standard error of a parameter is zero, which is assumed to mean that this parameter is fixed. By default, these columns are included and no need to specify them for this argument: `"z"`, `"pvalue"`, `"t"`, `"df"`, `"ci.lower"`, and `"ci.upper"`. To exclude one of these columns from `se_als_to_na`, add it to `se_not_to_na`.
`se_not_to_na`	Columns for which cells will *not* be set to `NA` even if the standard error of a parameter is zero. Column names that appear here *override* `se_also_to_na`. Therefore, if `"z"`, `"pvalue"`, `"t"`, `"df"`, `"ci.lower"`, and `"ci.upper"` are included in this argument, they will also not be set to `NA`.
`drop_cols`	The names of columns to be dropped from the printout. It can be the names after being renamed by `rename_cols`, or the original names before being renamed (i.e., the names in `object`), provided that the names in `object` are stored in the attribute `"original_name"` of each column, which is done by default by `parameterEstimates_table_list()`.
`rename_cols`	If any columns are to be renamed, this is named character vector, with the names being the original names and the values being the new names. For example, `c("pvalue" = "P(|>z|)")` renames the column `"pvalue"` to `"P(|z|)"`. It is recommended to quote the names too because they may not be standard names.
`est_funs`	If supplied, it should be a list of functions to be applied to each parameter estimates table, applied in the same order they appear in the list. It can be used create new columns or modify existing columns. Usually, this should be done *before* calling `parameterEstimates_table_list()` but provided as an option.
`header_funs`	If supplied, it should be a list of functions to be applied to `object` to generate the header sections. See `Details` on the expected format of the output of these functions.
`footer_funs`	If supplied, it should be a list of functions to be applied to `object` to generate the footer sections. See `Details` on the expected format of the output of these functions.
`est_funs_args`	If supplied, it must be a "list of list(s)". The length of this list must be equal to the number of functions in `est_funs`. Each sub-list is the list of arguments to be used when calling a function in `est_funs`. It must be an empty `list()` if no additional arguments are to be used when calling a function in `est_funs`.
`header_funs_args`	If supplied, it must be a "list of list(s)". The length of this list must be equal to the number of functions in `header_funs`. Each sub-list is the list of arguments to be used when calling a function in `header_funs`. It must be an empty `list()` if no additional arguments are to be used when calling a function in `header_funs`.
`footer_funs_args`	If supplied, it must be a "list of list(s)". The length of this list must be equal to the number of functions in `footer_funs`. Each sub-list is the list of arguments to be used when calling a function in `footer_funs`. It must be an empty `list()` if no additional arguments are to be used when calling a function in `footer_funs`.
`x`	The object to be printed. Should be the output of `parameterEstimates_table_list()`.
`nd`	The number of decimal places to be displayed for numeric cells.

by_group	If TRUE, the default, tables will be grouped by groups first, and then by grouped by sections (e.g., Latent Variables, Regressions, etc.). If FALSE, then the tables will be grouped by sections first, and then grouped by groups. No effect if the number of groups is equal to one.
na_str	The string to be used for cells with NA. Default is " ", a white space.

Details

This function creates an output mimicking the output format when lavaan::parameterEstimates() is called with output set to "text". It only creates the output as a list of data frames, grouped in sections like Latent Variables and Regression, as in the printout of lavaan::parameterEstimates(). It does not format the content. The actual printing is to be done by print_parameterEstimates_table_list(), which will format the cells before printing them.

This function is not intended to be used by end-users. It is intended to be used inside other functions, such as a print method. Functions that add columns to the parameter estimates table of a lavaan object can use it and PRINT_parameterEstimates_table_list() to print the output in the lavaan-style, but with columns modified as needed and with additional header and/or footer sections added.

Therefore, it was developed with flexibility in mind, at the expense of user-friendliness.

Header and Footer Functions:

If a list of functions is supplied to header_funs or footer_funs, they will be used to generate the headers and/or footers. The first argument of these function will be one of the followings.

If object is a data frame like object, then the first argument is this object when calling thews functions.

If object is a lavaan object, then the first argument is the parameter estimates table generated by lavaan::parameterEstimates() with output = "text", header = TRUE.

The output of these functions should be one of the following formats.

It can be a data frame with two optional attributes: section_title and print_args. If section_title is not null, it will be printed by cat() before printing tbe section. The header or footer section will then be printed by print(). If print_args is set to be a list of named arguments, then they will be used when calling print(). For example, setting print_args to list(right = FALSE, row.names = FALSE) will print the data frame with these arguments.

It can also be any other object that can be printed. One possible case is a character vector of footnotes. In this case, we can add this attribute print_fun and set it to "cat", the name of the function to be used to print the section, and add the attribute print_args and set it to be a named list of arguments to be passed to print_fun.

Special treatment when print-fun is "cat"‘:

- The default of sep is "\n". To override this default, set the attribute print_args and set sep to something else.
- Each element in the object, which should be a character vector, is processed by strwrap() by default. Additional arguments to strwrap() can be passed by setting the attribute strwrap_args to a named list of the arguments for strwrap() (e.g., list(exdent = 2)). To disable this feature, set the attribute wrap_lines to FALSE.

These arguments header_funs and footer_args allow users to add header and footer sections and print them in the desired format.

Value

`parameterEstimates_table_list()`**:**

A list of data frames of the class `parameterEstimates_table_list`, with this structure.

- `group`: A list of data frames for each group. It is a list of length equal to one if the model has only one group. For each group, the content is a list of data frames, one for each section of the estimates.
- `model`: A list of tables for sections such as user-defined parameters (`"Defined Parameters"`) or model constraints (`"Constraints"`).
- `header`: A list of header sections.
- `footer`: A list of footer sections.

The decision of not having a `print` method is intentional. It is intended to be used by other the `print` methods of other classes, to create the formatted list of tables, and then print it by calling `print_parameterEstimates_table_list()` internally.

`print_parameterEstimates_table_list()`**:**

The original input, `x`, is returned invisibly. Called for its side effect to print the content of `x`.

Limitations

These function do not yet support multilevel models.

Author(s)

Shu Fai Cheung https://orcid.org/0000-0002-9871-9448

See Also

`print_parameterEstimates_table_list()` for the printing function, and `lavaan::parameterEstimates()` for generating the parameter estimates table.

Examples

```
# Adapted from the help of lavaan::cfa()
library(lavaan)
mod <- "
visual  =~ x1 + x2 + x3
textual =~ x4 + x5 + x6
speed   =~ x7 + x8 + x9
"
fit <- cfa(mod,
           data = HolzingerSwineford1939)
est <- parameterEstimates_table_list(fit,
                                     rename_cols = c("P(>|z|)" = "pvalue",
                                                     "S.E." = "SE"))
print_parameterEstimates_table_list(est,
                                    drop = "Z")
fit2 <- cfa(mod,
            data = HolzingerSwineford1939,
            group = "school")
```

```
est2 <- parameterEstimates_table_list(fit2)
# The tables in the same group are printed together (default)
print_parameterEstimates_table_list(est2,
                                    by_group = TRUE)
# The table are grouped by section then by group
print_parameterEstimates_table_list(est2,
                                    by_group = FALSE)

# A simple function to add "***" for parameters with p-values < .001
add_sig <- function(object,
                    pvalue = "pvalue") {
    tmp <- object[, pvalue, drop = TRUE]
    if (!is.null(tmp)) {
        tmp[is.na(tmp)] <- 1
        tmp2 <- ifelse(tmp < .001, "***", "")
        i <- match(pvalue, colnames(object))
        out <- data.frame(object[, 1:i],
                          Sig = tmp2,
                          object[, seq(i + 1, ncol(object))])
      }
    out
  }

est3 <- parameterEstimates_table_list(fit2,
                                      est_funs = list(add_sig))
print_parameterEstimates_table_list(est3)
```

Index