

# Package ‘df2yaml’

January 26, 2023

**Type** Package

**Title** Convert Dataframe to 'YAML'

**Version** 0.3.1

**Maintainer** Yabing Song <songyb0519@gmail.com>

**Description** The 'df2yaml' aims to simplify the process of converting 'dataframe' to 'YAML' <<https://yaml.org/>>. The 'dataframe' with multiple key columns and one value column will be converted to the multi-level hierarchy.

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.1.1

**Imports** dplyr, rraply, tibble, yaml, magrittr, rlang

**Suggests** rmarkdown, prettydoc, knitr

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Yabing Song [aut, cre]

**Repository** CRAN

**Date/Publication** 2023-01-26 12:50:02 UTC

## R topics documented:

df2yaml	2
df2yaml2	3

<b>Index</b>	<b>4</b>
--------------	----------

---

`df2yaml`*Convert dataframe to YAML.*

---

## Description

Convert dataframe to YAML.

## Usage

```
df2yaml(  
  df,  
  key_col,  
  val_col,  
  val_sep = ";",  
  key_sep = ":",  
  rm_quote = TRUE,  
  out_yaml = NULL  
)
```

## Arguments

<code>df</code>	Dataframe.
<code>key_col</code>	The columns used as keys.
<code>val_col</code>	The columns used as values, this column can contain key: value pairs.
<code>val_sep</code>	The separator used to separate different key:value pairs in <code>val_col</code> . Default: ";".
<code>key_sep</code>	The separator used to separate key and value. Default: ":".
<code>rm_quote</code>	Logical value, whether to remove single quotes. Default: TRUE.
<code>out_yaml</code>	Output YAML file. Default: NULL (return string).

## Value

NULL (write YAML) or string (if `out_yaml` is NULL).

## Examples

```
library(df2yaml)  
test_file <- system.file("extdata", "df2yaml_l3.txt", package = "df2yaml")  
test_data <- read.table(file = test_file, header = TRUE, sep = "\t")  
df2yaml(df = test_data, key_col = c("paras", "subcmd"), val_col = "values")
```

---

`df2yaml2`*Convert dataframe to YAML.*

---

**Description**

Convert dataframe to YAML.

**Usage**

```
df2yaml2(  
  df,  
  key_col = c("paras", "subcmd"),  
  val_col = "values",  
  val_sep = ";",  
  key_sep = ":",  
  out_yaml = NULL  
)
```

**Arguments**

<code>df</code>	Dataframe.
<code>key_col</code>	The columns used as keys, up to 2.
<code>val_col</code>	The columns used as values, this column can contain key: value pairs.
<code>val_sep</code>	The separator used to separate different key:value pairs in <code>val_col</code> . Default: ";".
<code>key_sep</code>	The separator used to separate key and value. Default: ":".
<code>out_yaml</code>	The output yaml file. Default: NULL (string).

**Value**

NULL (write YAML) or string (if `out_yaml` is NULL).

**Examples**

```
library(df2yaml)  
test_file <- system.file("extdata", "df2yaml_l3.txt", package = "df2yaml")  
test_data <- read.table(file = test_file, header = TRUE, sep = "\t")  
df2yaml(df = test_data, key_col = c("paras", "subcmd"), val_col = "values")
```

# Index

df2yam1, [2](#)  
df2yam12, [3](#)