

Package ‘syntaxr’

July 23, 2025

Type Package

Title An 'SPSS' Syntax Generator for Multi-Variable Manipulation

Version 0.8.0

Author Alix Lahuec <alix.lahuec@mail.mcgill.ca>

Maintainer Alix Lahuec <alix.lahuec@mail.mcgill.ca>

URL <https://github.com/greenmeen/syntaxr>

BugReports <https://github.com/greenmeen/syntaxr/issues>

Description A set of functions for generating 'SPSS' syntax files from the R environment.

Imports magrittr

Suggests covr, haven, testthat

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

NeedsCompilation no

Repository CRAN

Date/Publication 2019-04-06 11:10:03 UTC

Contents

spss.compute.concat	2
spss.compute.max	2
spss.concat.new	3
spss.format.compute	3
spss.format.concat	4
spss.format.max	4
spss.rename	5
spss.rtrim	5
spss.string	6

Index	7
--------------	----------

spss.compute.concat	<i>Generate SPSS 'COMPUTE' syntax to compute the CONCAT() of two (sets of) variables.</i>
---------------------	---

Description

Generate SPSS 'COMPUTE' syntax to compute the CONCAT() of two (sets of) variables.

Usage

```
spss.compute.concat(str1, str2, names)
```

Arguments

str1	the first argument for CONCAT().
str2	the second argument for CONCAT().
names	the name(s) of the variable(s) to be created.

Examples

```
spss.compute.concat("feedback", "feedback_f", "enfr.feedback")
spss.compute.concat("feedback", "feedback_f", "enfr.feedback")
```

spss.compute.max	<i>Generate SPSS 'COMPUTE' syntax to compute the MAX() of two (sets of) variables.</i>
------------------	--

Description

Generate SPSS 'COMPUTE' syntax to compute the MAX() of two (sets of) variables.

Usage

```
spss.compute.max(var1, var2, append = "new.", ...)
```

Arguments

var1	the first argument for MAX(). Used for the naming of the output variable(s).
var2	the second argument for MAX().
append	specifies the text that should be appended to the name(s) of the variable(s) in var1 to create the output variable(s).
...	any additional arguments that can be passed to functions spss.format.max and spss.format.compute

Examples

```
spss.compute.max(c("dob", "income"), c("dob_f", "income_f"), append = "total.")
spss.compute.max(c("dob", "income"), c("dob_f", "income_f"))
```

spss.concat.new	<i>Generate SPSS 'STRING' syntax and 'COMPUTE' syntax to compute the CONCAT() of two (sets of) variables.</i>
-----------------	---

Description

Generate SPSS 'STRING' syntax and 'COMPUTE' syntax to compute the CONCAT() of two (sets of) variables.

Usage

```
spss.concat.new(str1, str2, append = "", name = "", ...)
```

Arguments

str1	the first argument for CONCAT(). Used for the naming of the output variable(s).
str2	the second argument for CONCAT().
append	specifies the text that should be appended to the name(s) of the variable(s) in str1 to create the output variable(s). Defaults to an empty string.
name	specifies the name of the new String variable to be created. Defaults to an empty string ; if left unspecified, the function will use the append parameter.
...	any additional arguments that can be passed to functions spss.string and spss.compute.concat

Examples

```
spss.concat.new(c("dob", "income"), c("dob_f", "income_f"), append = "total.")
spss.concat.new(c("dob", "income"), c("dob_f", "income_f"))
```

spss.format.compute	<i>Generate SPSS 'COMPUTE' syntax to carry out an operation on two (sets of) variables.</i>
---------------------	---

Description

Generate SPSS 'COMPUTE' syntax to carry out an operation on two (sets of) variables.

Usage

```
spss.format.compute(output, operation)
```

Arguments

output	specifies the name(s) of the output variable(s) of COMPUTE().
operation	specifies the computational operation to be carried out.

Examples

```
spss.format.compute("new.var", "MAX(var1,var2)")
spss.format.compute(c("dob", "comments"), c("MAX(dob,dob_f)", "CONCAT(comments,comments_f)"))
spss.format.compute("string.var", "CONCAT(string1,string2)")
```

spss.format.concat	<i>Generate SPSS 'concat()' syntax for two (sets of) variables.</i>
--------------------	---

Description

Generate SPSS 'concat()' syntax for two (sets of) variables.

Usage

```
spss.format.concat(var1, var2)
```

Arguments

var1	the first argument for concat().
var2	the second argument for concat().

Examples

```
spss.format.concat(c("dob", "income"), c("dob_f", "income_f"))
spss.format.concat("income", "income_f")
```

spss.format.max	<i>Generate SPSS 'MAX()' syntax for two (sets of) variables.</i>
-----------------	--

Description

Generate SPSS 'MAX()' syntax for two (sets of) variables.

Usage

```
spss.format.max(var1, var2)
```

Arguments

var1	the first argument for MAX().
var2	the second argument for MAX().

Examples

```
spss.format.max(c("dob", "income"), c("dob_f", "income_f"))
spss.format.max("income", "income_f")
```

spss.rename	<i>Generate SPSS 'RENAME' syntax to rename a (set of) variables into another.</i>
-------------	---

Description

Generate SPSS 'RENAME' syntax to rename a (set of) variables into another.

Usage

```
spss.rename(values, rename)
```

Arguments

values	the variable(s) to be renamed.
rename	the name(s) to use for renaming.

Examples

```
spss.rename(c("oldname1", "oldname2", "oldname3"), c("new1", "new2", "new3"))
spss.rename(c("oldname1", "oldname2"), c("new1", "new2"))
```

spss.rtrim	<i>Generate SPSS 'RTRIM' syntax to apply RTRIM() to a string variable.</i>
------------	--

Description

Generate SPSS 'RTRIM' syntax to apply RTRIM() to a string variable.

Usage

```
spss.rtrim(str)
```

Arguments

str	the string argument for RTRIM().
-----	----------------------------------

Examples

```
spss.rtrim("variable_to_be_trimmed")
```

`spss.string`*Generate SPSS 'STRING' syntax to create a variable of type string.*

Description

Generate SPSS 'STRING' syntax to create a variable of type string.

Usage

```
spss.string(names, string.format = "A15")
```

Arguments

<code>names</code>	the name(s) of the variable(s) to be created.
<code>string.format</code>	specifies the formatting to use when creating the string variable. Defaults to "A15".

Examples

```
spss.string("my.string")  
spss.string("long.string", string.format = "A40")
```

Index

spss.compute.concat, [2](#)
spss.compute.max, [2](#)
spss.concat.new, [3](#)
spss.format.compute, [3](#)
spss.format.concat, [4](#)
spss.format.max, [4](#)
spss.rename, [5](#)
spss.rtrim, [5](#)
spss.string, [6](#)