Package 'datapasta'

October 13, 2022

Title R Tools for Data Copy-Pasta

Version 3.1.0

Description
RStudio addins and R functions that make copy-pasting vectors and tables to text painless.

Depends R (>= 3.3.0)

Suggests tibble (>= 1.2), testthat, knitr, rmarkdown, utils, covr, data.table

Imports readr (>= 1.2.0), clipr (>= 0.3.0), rstudioapi (>= 0.6), methods

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.0.1

URL https://github.com/milesmcbain/datapasta

BugReports https://github.com/milesmcbain/datapasta/issues

VignetteBuilder knitr

NeedsCompilation no

Author Miles McBain [aut, cre] (<https://orcid.org/0000-0003-2865-2548>),
Jonathan Carroll [aut] (<https://orcid.org/0000-0002-1404-5264>),
Mark Dulhunty [ctb],
Andrew Collier [ctb],
Sharla Gelfand [aut],
Suthira Owlarn [aut] (<https://orcid.org/0000-0002-3258-1415>),
Garrick Aden-Buie [aut] (<https://orcid.org/0000-0002-7111-0077>)

Maintainer Miles McBain <miles.mcbain@gmail.com>

Repository CRAN

Date/Publication 2020-01-17 12:40:02 UTC

R topics documented:

`clipboard_context` *custom_context*

Description

the _context functions define lists of parameters for text formatting. The specific contexts return hard-coded values appropriate to the context they describe, while custom_context allows definition of new contexts for custom formatting.

Usage

```
clipboard_context()

rstudio_context()

console_context()

markdown_context()

custom_context(
  output_mode = "console",
  nspc = 2,
  indent_context = 0,
  indent_head = TRUE
)
```

Arguments

output_mode	A named output mode, controls the target of the _paste functions options are "rstudioapi" or "console"
nspc	The number of spaces for each indent level in the output context
indent_context	The number of spaces applied initially to all lines in the output context
indent_head	Logical. Apply the indent_context to the to the header row? Use FALSE if targeting cursor location.

Value

an output context. An input to _paste, _format, _construct functions used to format whitespace.

---

dfdt_construct *dfdt_construct*

Description

Parse the current clipboard as a table and return in data.frame format.

Usage

```
dfdt_construct(input_table, oc = console_context(), class = NULL)
```

Arguments

input_table	an optional R object to parse instead of the clipboard.
oc	an optional output context that defines the target and indentation.
class	either data.frame or data.table.

Value

a character string containing the input formatted as a data.frame definition.

`df_format` *df_format*

Description

Parse the current clipboard as a table and paste to the clipboard in data.frame format.

Usage

```
df_format(input_table, output_context = clipboard_context())
```

Arguments

`input_table`	an optional input tibble or data.frame to format.
`output_context`	an optional output context that defines the target and indentation.

Value

nothing.

`df_paste` *df_paste*

Description

Parse either: the current clipboard, or a supplied argument, as a table and paste in at the cursor location in data.frame format.

Usage

```
df_paste(input_table, output_context = guess_output_context())
```

Arguments

`input_table`	an optional input tibble or data.frame to format.
`output_context`	an optional output context that defines the target and indentation. The default behaviour is target the rstudioapi and fall back to console if it is not available.

Value

nothing.

`dmdclip` *dmdclip*

Description

Formats input for presentation in markdown as a preformatted chunk and inserts it onto the clipboard. Ready for pasting to Stack Overflow or Github.

Usage

```
dmdclip(input)
```

Arguments

`input`	a vector, data.frame, or tibble

Value

nothing

`dpasta` *dpasta*

Description

Formats input and inserts at either the current cursor or the console.

Usage

```
dpasta(input)
```

Arguments

`input`	a vector, data.frame, or tibble

Value

nothing

dp_set_decimal_mark *dp_set_decimal_mark*

Description

A function to optionally set the decimal mark if in a locale where it is not '.'. Will allow "3,14" to be parsed as 3.14, normally would be parsed as 314. Will also handle spaces in numbers.

Usage

```
dp_set_decimal_mark(mark)
```

Arguments

mark	The decimal mark to use when parsing "number" type data, as guessed by readr::guess_parser.

Value

NULL.

dp_set_max_rows *dp_set_max_rows*

Description

dp_set_max_rows

Usage

```
dp_set_max_rows(num_rows)
```

Arguments

num_rows	The number of rows of an input at which any of tribble_construct() or df_construct() will abort parsing. Datapasta is untested on large tables. Use at own risk.

`dt_format` *dt_format*

Description

Parse the current clipboard as a table and paste to the clipboard in data.table format.

Usage

```
dt_format(input_table, output_context = clipboard_context())
```

Arguments

`input_table`	an optional input tibble or data.frame to format.
`output_context`	an optional output context that defines the target and indentation.

Value

nothing.

`dt_paste` *dt_paste*

Description

Parse either: the current clipboard, or a supplied argument, as a table and paste in at the cursor location in data.table format.

Usage

```
dt_paste(input_table, output_context = guess_output_context())
```

Arguments

`input_table`	an optional input tibble or data.frame to format.
`output_context`	an optional output context that defines the target and indentation. The default behaviour is target the rstudioapi and fall back to console if it is not available.

Value

nothing.

guess_output_context *guess_output_context*

Description

Return the a list containing the guessed output target context, either rstudio or the console.

Usage

```
guess_output_context()
```

Value

a list containing the output target, space size of indent, and number of indents at context.

guess_sep *guess_sep*

Description

Guesses the separator based on a simple heuristic over the first 10 or less rows: The separator chosen is the one that leads to the most columns, whilst parsing the same number of columns for each line (var=0). The guessing algorithm ignores blank lines - which are lines that contain only the separator. Options are in ‘c(",","\t","\|",";")‘

Usage

```
guess_sep(char_vec)
```

Arguments

char_vec	a table from the clipboard in character vector form.

Value

the separator selected to parse char_vec as a table

nchar_type *nchar_type*

Description

nchar_type

Usage

```
nchar_type(df_col_row, df_col_type)
```

Arguments

df_col_row	a character string
df_col_type	the type the string will be converted to.

Value

The number of characters wide this data would be in when rendered in text

nquote_str *Count the number of quotes in a string*

Description

Count the number of quotes in a string

Usage

```
nquote_str(char_vec)
```

Arguments

char_vec	the string to count quotes in

Value

a number, possibly 0.

pad_to *pad_to*

Description

Left pad string to a certain size. A helper function for getting spacing in table correct.

Usage

```
pad_to(char_vec, char_length)
```

Arguments

char_vec	character vector.
char_length	length to pad to.

Value

char_vec left-padded with spaces to char_length.

parse_vector *parse_vector*

Description

Pastes data from clipboard as a vertically formatted character vector on multiple lines. One line is used per element. Considers ‘,‘, ‘tab‘, ‘newline‘ as delimiters.

Usage

```
parse_vector(input_vector)
```

Arguments

input_vector	an optional character vector to attempt to break up, and escape.

Value

A vector parsed from the clipboard as ether a character string or a character vector. The type attribute contains the type guessed by ‘readr‘.

read_clip_tbl_guess *read_clip_table_guess*

Description

Similar to read_clip_tbl from clipr, however it will error if there are less than 2 rows and it tries to guess the separator.

Usage

```
read_clip_tbl_guess(x = NULL, ...)
```

Arguments

x	contents of the clipboard
...	arguments passed to read.table

Value

a parsed table from the clipboard. Separator is guessed.

render_type *render_type*

Description

Renders a character vector as R types for pasting into Rstudio. Strings are quoted. Numbers, NA, logicals etc are not.

Usage

```
render_type(char_vec, char_type)
```

Arguments

char_vec	a character vector containing text to be rendered as the type indicated by type_str
char_type	a string describing the type of char_vec

Value

A vector parsed from the clipboard as ether a character string or a character vector. The type attribute contains the type guessed by ‘readr‘.

render_type_pad_to *render_type_pad_to*

Description

Based on a type and length, render a character string as the type in text. Pad to the desired length.

Usage

```
render_type_pad_to(char_vec, char_type, char_length)
```

Arguments

char_vec	a character vector
char_type	a string type from readr::guess_parser
char_length	a string length to pad to.

Value

a string containing the representation of char_vec as char_type in the RStudio source editor, left-padded with spaces to char_length.

tortellini *wrap the datapasta around itself*

Description

wrap the datapasta around itself

Usage

```
tortellini(col_struct, defn_width = 80, indent_context = 0, add_comma = TRUE)
```

Arguments

col_struct	input structure - a split apart column definition
defn_width	total number of characters in a line (includes column name and indent on line 1)
indent_context	the level of indent in spaces in the current editor pane
add_comma	add one final comma to the end of the wrapped column def? Useful when pasting together columns.

Value

w wrapped string

tribble_construct *tribble_construct*

Description

Parse the current clipboard as a table, or use the table argument supplied, and return as a character string.

Usage

```
tribble_construct(input_table, oc = console_context())
```

Arguments

input_table	an optional input 'data.frame'. If 'input_table' is supplied, then nothing is read from the clipboard.
oc	an optional output context that defines the target and indentation. Default is console. Table is output as 'tribble()' call. Useful for creating reproducible examples.

Value

The parsed table text.

tribble_format *tribble_format*

Description

Parse the current clipboard as a table, or use the table argument supplied, and paste to the clipboard in tribble format.

Usage

```
tribble_format(input_table, output_context = console_context())
```

Arguments

input_table	an optional input 'data.frame'. If 'input_table' is supplied, then nothing is read from the clipboard.
output_context	an optional output context that defines the target and indentation. Default is console. Table is output as 'tribble()' call. Useful for creating reproducible examples.

Value

Nothing.

tribble_paste *tribble_paste*

Description

Parse the current clipboard as a table, or use the table argument supplied, and paste in at the cursor location in tribble format.

Usage

```
tribble_paste(input_table, output_context = guess_output_context())
```

Arguments

input_table	an optional input 'data.frame'. If 'input_table' is supplied, then nothing is read from the clipboard.
output_context	an optional output context that defines the target and indentation. Default is to guess between rstudio and console. Table is output as 'tribble()' call. Useful for creating reproducible examples.

Value

Nothing.

vector_construct *vector_construct*

Description

Returns a formatted character string, either from clipboard or supplied argument, as a vector definition. Considers ',', 'tab', 'newline' as delimiters. If a single character string is passed as an argument, it will be split to form a vector.

Usage

```
vector_construct(input_vector, oc = console_context())
```

Arguments

input_vector	An input vector to be formatted for output. If supplied, no data is read from the clipboard.
oc	an optional output context that defines the output indentation.

Value

A string containing the input formatted as a vector definition.

vector_construct_vertical *vector_construct_vertical*

Description

Returns a formatted string, either from clipboard or supplied argument, as a vertically formatted character vector over many lines. Considers ‘,‘, ‘tab‘, ‘newline‘ as delimiters. If a single character string is passed as an argument, it will be split to form a vector.

Usage

```
vector_construct_vertical(input_vector, oc = console_context())
```

Arguments

input_vector	An input vector to be formatted for output. If supplied, no data is read from the clipboard.
oc	an optional output context that defines the output target and indentation. The default behaviour is to target the rstudioapi and fall back to console if it is not available.

Value

a string containing the input formatted as a vector definition.

vector_format *vector_format*

Description

Writes data to the clipboard, either from clipboard or supplied argument. Writes a horizontally formatted character vector on a single line. Considers ‘,‘, ‘tab‘, ‘newline‘ as delimiters. If a single character string is passed as an argument, it will be split to form a vector.

Usage

```
vector_format(input_vector, output_context = console_context())
```

Arguments

input_vector	An input vector to be formatted for output. If supplied, no data is read from the clipboard.
output_context	an optional output context that defines the output indentation.

Value

nothing.

`vector_format_vertical` *vector_format_vertical*

Description

Writes data to clipboard, either from clipboard or supplied argument, as a vertically formatted character vector over many lines. Considers ‘,‘, ‘tab‘, ‘newline‘ as delimiters. If a single character string is passed as an argument, it will be split to form a vector.

Usage

```
vector_format_vertical(input_vector, output_context = clipboard_context())
```

Arguments

`input_vector`	An input vector to be formatted for output. If supplied, no data is read from the clipboard.
`output_context`	an optional output context that defines the output target and indentation. The default behaviour is to target the rstudioapi and fall back to console if it is not available.

Value

nothing.

`vector_paste` *vector_paste*

Description

Pastes data, either from clipboard or supplied argument, as a horizontally formatted character vector on a single line. Considers ‘,‘, ‘tab‘, ‘newline‘ as delimiters. If a single character string is passed as an argument, it will be split to form a vector.

Usage

```
vector_paste(input_vector, output_context = guess_output_context())
```

Arguments

input_vector	An input vector to be formatted for output. If supplied, no data is read from the clipboard.
output_context	an optional output context that defines the output target and indentation. The default behaviour is to target the rstudioapi and fall back to console if it is not available.

Value

nothing.

---

vector_paste_vertical *vector_paste_vertical*

Description

Pastes data, either from clipboard or supplied argument, as a vertically formatted character vector over many lines. Considers ‘,‘, ‘tab‘, ‘newline‘ as delimiters. If a single character string is passed as an argument, it will be split to form a vector.

Usage

```
vector_paste_vertical(input_vector, output_context = guess_output_context())
```

Arguments

input_vector	An input vector to be formatted for output. If supplied, no data is read from the clipboard.
output_context	an optional output context that defines the output target and indentation. The default behaviour is to target the rstudioapi and fall back to console if it is not available.

Value

nothing.

zzz_rs_dfiddle *dfiddle*

Description

An addin to fiddle your RStudio selections to better things. Make a selection in RStudio and dfiddle will update it inline. Good for: Converting Text to vectors (‘c()‘), pivoting between horizontal and vertical vectors, reflowing tribble() and data.frame() definitions to have nice indenting and padding.

Usage

```
zzz_rs_dfiddle()
```

Value

a fiddled version of your selection (invisibly)

zzz_rs_toggle_quotes *Toggle Quotes*

Description

An addin to toggle between quotes and bare vectors. Applies to a vector selected in an RStudio source editor. Works with horizontal or vertical form.

Usage

```
zzz_rs_toggle_quotes()
```

Value

The toggled vector (invisibly).

Index