Package 'pandoc'

August 24, 2023

Title Manage and Run Universal Converter 'Pandoc' from 'R'

Version 0.2.0

Description Provides a set of tools to install, manage and run several
'Pandoc' versions.

License MIT + file LICENSE

URL <https://github.com/cderv/pandoc>, <https://cderv.github.io/pandoc/>

BugReports <https://github.com/cderv/pandoc/issues>

Depends R (>= 3.5.0)

Imports fs, rappdirs, rlang (>= 1.0.1)

Suggests covr, gh, rmarkdown (>= 2.14), testthat (>= 3.0.0), tibble,
withr

Config/Needs/website tidyverse/tidytemplate, dplyr, tidyr, purrr

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.2.3

NeedsCompilation no

Author Christophe Dervieux [aut, cre]
(<<https://orcid.org/0000-0003-4474-2498>>),
Posit Software, PBC [cph, fnd]

Maintainer Christophe Dervieux <cderv@posit.co>

Repository CRAN

Date/Publication 2023-08-24 11:00:08 UTC

R topics documented:

---

pandoc_activate   *Activate a specific Pandoc version to be used*

Description

This function will set the specified version as the default version for the session. By default, the
default active version in the most recent one among the installed version (nightly version excluded.)

Usage

```
pandoc_activate(
  version = "latest",
  rmarkdown = getOption("pandoc.activate_rmarkdown", TRUE),
  quiet = FALSE
)
```

Arguments

version	This can be either:

- "latest" for the latest release
- A version number (e.g "2.11.4") for a specific version
- "nightly" for the last pandoc development built daily

rmarkdown	if TRUE (the default) and **rmarkdown** is available, this will also set the pandoc version as the default one to use with **rmarkdown** by calling rmarkdown::find_pandoc(). Default behavior can be changed globally by setting option pandoc.activate_rmarkdown.
quiet	TRUE to suppress messages.

Value

invisibly, the previous active version.

Special behavior in an interactive session

If the version to activate is not yet installed, the user will be prompted to choose to install the version.

Default active version

When the package is loaded, an active version is set to the first Pandoc binary found between:

- the latest Pandoc version installed with this package (e.g "2.14.2")
- the version shipped with RStudio IDE. (version = "rstudio")
- a version available in PATH (version = "system")

Examples

```
# activate version 2.18, including for use with rmarkdown package
pandoc_activate("2.18")

# activate only for this package functions and not rmarkdown
pandoc_activate("2.18", rmarkdown = FALSE)
```

---

pandoc_available	*Check if active Pandoc version meet a requirement*

Description

This function allows to test if an active Pandoc version meets a min, max or in between requirement. See pandoc_activate() about active version.

Usage

```
pandoc_available(min = NULL, max = NULL)
```

Arguments

`min`	Minimum version expected.
`max`	Maximum version expected

Details

If `min` and `max` are provided, this will check the active version is in-between two versions. If non is provided (keeping the default `NULL` for both), it will check for an active version and return `FALSE` if none is active.

Value

logical. `TRUE` if requirement is met, `FALSE` otherwise.

Examples

```
# Is there an active version available ?
pandoc_available()
# check for a minimum requirement
pandoc_available(min = "2.11")
# check for a maximum version
pandoc_available(max = "2.18")
# only returns TRUE if Pandoc version is between two bounds
pandoc_available(min = "2.11", max = "2.12")
```

---

pandoc_available_releases

Fetch all versions available to install

Description

This function will fetch information from Github <https://github.com/jgm/pandoc> about available Pandoc versions released.

Usage

```
pandoc_available_releases()
```

Value

character vector of all available release.

See Also

[pandoc_install()](), [pandoc_installed_versions()]()

Examples

---

pandoc_bin   *Get path to the pandoc binary*

Description

Get path to the pandoc binary

Usage

```
pandoc_bin(version = "default")

pandoc_bin_browse(version = "default")
```

Arguments

version   Version to use. Default will be the "default" version. Other possible value are

- A version number e.g "2.14.1"
- The nightly version called "nightly"
- The latest installed version with "latest"
- Pandoc binary shipped with RStudio IDE with "rstudio"
- Pandoc binary found in PATH with "system"

Details

pandoc_bin_browse() allows to open in OS explorer the folder where pandoc_bin() is at, when in interactive mode only.

Value

Absolute path to the pandoc binary of the requested version.

Examples

```
pandoc_bin()
pandoc_bin("2.18")
pandoc_bin("nightly")
pandoc_bin("rstudio")
pandoc_bin("system")
pandoc_bin_browse("2.18")
```

---

pandoc_browse_citation

Open Pandoc's documentation about citation processing

Description

Open Pandoc's documentation about citation processing

Usage

```
pandoc_browse_citation()
```

Value

Open the webpage at the place regarding citation processing in Pandoc.

Examples

```
pandoc_browse_citation()
```

---

pandoc_browse_exit_code

Open Pandoc's documentation about exit codes

Description

Open Pandoc's documentation about exit codes

Usage

```
pandoc_browse_exit_code()
```

Value

Open the webpage at the place regarding exit code thrown by Pandoc.

Examples

```
pandoc_browse_exit_code()
```

---

pandoc_browse_extension

Open Pandoc's documentation about an extension

Description

Open Pandoc's documentation about an extension

Usage

```
pandoc_browse_extension(extension = NULL)
```

Arguments

extension   One of the supported extension. See [pandoc_list_extensions()](). As the Pandoc MANUAL only concerns the last released Pandoc's version, if the URL is incorrect this could mean the extensions has changed.

Value

Open the webpage at the place regarding the required extension.

Examples

```
pandoc_browse_extension()
pandoc_browse_extension("auto_identifiers")
```

---

pandoc_browse_manual   *Open Pandoc's MANUAL*

Description

Open Pandoc's MANUAL

Usage

```
pandoc_browse_manual(id = NULL)
```

Arguments

id   One of the id available in the HTML page (usually for anchor link).

Value

Open the Pandoc's MANUAL

References

<https://pandoc.org/MANUAL.html>

Examples

```
# open MANUAL home page
pandoc_browse_manual()
# open MANUAL at math part
pandoc_browse_manual("math")
```

---

pandoc_browse_option *Open Pandoc's documentation about a command line option*

Description

Open Pandoc's documentation about a command line option

Usage

```
pandoc_browse_option(option = NULL)
```

Arguments

option   One of the supported **long form** command line option. As the Pandoc MAN-
                  UAL only concerns the last released Pandoc's version, if the URL is incorrect
                  this could mean the option has changed.

Value

Open the webpage at the place regarding the required option

Examples

```
pandoc_browse_option()
pandoc_browse_option("embed-resources")
```

---

pandoc_browse_release   *Open Pandoc's release page in browser*

Description

Open Pandoc's release page in browser

Usage

```
pandoc_browse_release(version = "latest")
```

Arguments

version   One of pandoc release version number (e.g '2.11.2') or `"latest"`.

Value

Open the web page in browser in interactive mode or print the url

References

<https://github.com/jgm/pandoc/releases>

Examples

```
pandoc_browse_release()
pandoc_browse_release("2.14")
```

---

pandoc_citeproc_bin   *Get path to the pandoc-citeproc binary.*

Description

This function will return the path to `pandoc-citeproc` if available. It will only work with `version` of Pandoc installed by this package.

Usage

```
pandoc_citeproc_bin(version = "default")
```

Arguments

version   Version to use. Default will be the `"default"` version. Other possible value are

   • A version number e.g `"2.14.1"`
   • The nightly version called `"nightly"`
   • The latest installed version with `"latest"`
   • Pandoc binary shipped with RStudio IDE with `"rstudio"`
   • Pandoc binary found in PATH with `"system"`

Value

the path to `pandoc-citeproc` binary if it exists. Since Pandoc 2.11, the citeproc filter has been included into Pandoc itself and is no more shipped as a binary filter.

Examples

```
# Look into current active version
pandoc_citeproc_bin()


# Look into a specific version
pandoc_citeproc_bin("2.9.2")
```

---

pandoc_convert   *Run Pandoc to convert a document or a text*

Description

Main usage of Pandoc is to convert some text from a format into another. This function will do just that:

   • converting from a file or directly from text
   • writing to a file or directly to console

Usage

```
pandoc_convert(
  file = NULL,
  text = NULL,
  from = "markdown",
  to,
  output = NULL,
  standalone = FALSE,
  args = c(),
  version = "default"
)
```

Arguments

`file, text`	One or the other should be provided
`from`	Format to convert from. This must be one of the format supported by Pandoc. Default will be markdown. This correspond to the --from/-f CLI flag
`to`	Format to convert to. This must be one of the format supported by Pandoc. This correspond to the --to/-t CLI flag.
`output`	Pass a path to a file to write the result from Pandoc conversion into a file. This corresponds to the --output/-o flag
`standalone`	Should appropriate header and footer be included ? This corresponds to --standalone/-s CLI flag
`args`	Any other flag supported by Pandoc CLI. See https://pandoc.org/MANUAL.html#options
`version`	Version to use. Default will be the "default" version. Other possible value are

- A version number e.g "2.14.1"
- The nightly version called "nightly"
- The latest installed version with "latest"
- Pandoc binary shipped with RStudio IDE with "rstudio"
- Pandoc binary found in PATH with "system"

Value

`output` is provided, the absolute file path. If not, the output of `pandoc` binary run.

Examples

```
pandoc::pandoc_convert(text = "_This will be emphasize_", to = "latex")


pandoc::pandoc_convert(text = "**This will be bold**", to = "html", version = "2.11.4")
```

`pandoc_export_data_file`

Export Pandoc internal data file

Description

This correspond to the --print-default-data-file CLI flag using also --output to write a export a data file built in Pandoc.

Usage

```
pandoc_export_data_file(file, output = file, version = "default")

pandoc_export_reference_doc(type = c("docx", "pptx"), version = "default")
```

Arguments

file
: One of data file name included in Pandoc (e.g `reference.pptx`, `styles.html`)

output
: Path where to export the file. Default to working directory with the same file name.

version
: Version to use. Default will be the `"default"` version. Other possible value are

- A version number e.g `"2.14.1"`
- The nightly version called `"nightly"`
- The latest installed version with `"latest"`
- Pandoc binary shipped with RStudio IDE with `"rstudio"`
- Pandoc binary found in PATH with `"system"`

type
: one of `docx` or `pptx` depending on the reference doc to export.

Details

`pandoc_export_reference_doc()` is a helper to quickly get the reference doc for Word document (`reference.docx`) or Powerpoint document (`reference.pptx`)

Value

the `output` (invisibly) where export has been done

Examples

```
# export style.html file included in Pandoc HTML template
pandoc_export_data_file("styles.html", output = "custom.html")
# export css file used for epub by default
pandoc_export_data_file("epub.css")
```

```
pandoc_export_reference_doc("docx")
pandoc_export_reference_doc("pptx")
```

```
pandoc_export_reference_doc("pptx", version = "2.11.4")
```

---

pandoc_export_highlight_theme

Export highlighting style as JSON file

Description

Pandoc highlighting can be customize using a JSON .theme file, passed to --highlight-style= flag. This function allows to generate the JSON version of one of the supported highlighting style.

Usage

```
pandoc_export_highlight_theme(
  style = "pygments",
  output = style,
  version = "default"
)
```

Arguments

style	One of the support highlighting style. (See pandoc_list_highlight_style()).
output	Path (without extension) where to export the JSON .theme file. By default, the file will be located in working directory and named based on the parameter style (i.e <style>.theme).
version	Version to use. Default will be the "default" version. Other possible value are

- A version number e.g "2.14.1"
- The nightly version called "nightly"
- The latest installed version with "latest"
- Pandoc binary shipped with RStudio IDE with "rstudio"
- Pandoc binary found in PATH with "system"

Details

The .theme extension is required and it will be enforced in during the export by this function.

Value

the filename where the theme has been exported.

Note

This correspond to the --print-highlight-style CLI flag using also --output to write a export a data file built in Pandoc.

Examples

```
# export tango theme used by Pandoc highlighting to `tango.theme` file
pandoc_export_highlight_theme("tango")
pandoc_export_highlight_theme("pygments", output = "my_theme.theme")
pandoc_export_highlight_theme("zenburn", version = "system")
```

---

pandoc_export_template

Retrieve Pandoc template for a format

Description

This correspond to the --print-default-template CLI flag. With this function, one can easily export default LaTeX template for example.

Usage

```
pandoc_export_template(format = "markdown", output = NULL, version = "default")
```

Arguments

format   One of Pandoc format using a text template. (e.g html, latex, revealjs)

output   Path where to save the file. If not provided, the default, template content will be print to the console.

version   Version to use. Default will be the "default" version. Other possible value are

- A version number e.g "2.14.1"
- The nightly version called "nightly"
- The latest installed version with "latest"
- Pandoc binary shipped with RStudio IDE with "rstudio"
- Pandoc binary found in PATH with "system"

Value

If output is not provided, the content of the template will be printed and return as one string (invisibly). If output is provided, the file path of the output (invisibly).

Examples

```
pandoc_export_template()
```

```
pandoc_export_template("latex", output = "default.tex", version = "system")
```

---

pandoc_install   *Install a pandoc binary for Github release page*

Description

Binary releases of Pandoc are available on its release page. By default, this function will install the latest available version. `pandoc_install_nightly()` is a wrapper for `pandoc_install("nightly")`. `pandoc_update()` is an alias for `pandoc_install()` default behavior.

Usage

```
pandoc_install(version = "latest", force = FALSE)

pandoc_update()

pandoc_install_nightly(n_last = 1L)
```

Arguments

version   This can be either:
               - "latest" for the latest release
               - A version number (e.g "2.11.4") for a specific version
               - "nightly" for the last pandoc development built daily

force   To set to TRUE to force a re-installation

n_last   Set to n as integer to install the n-th from last nightly build. Default is last available build (1L)

Details

Pandoc versions are installed in user data directories with one folder per version. See `pandoc_locate()`.

Only one nightly version is available at a time as there should be no need to switch between them. The latest nightly will be installed over the current one if any. Installing nightly version is useful for example to test a bug against the very last available built version.

Value

Invisibly, the path where the binary is installed otherwise. NULL if already installed.

Examples

```
# Install the latest pandoc version
pandoc_install() # or pandoc_update()
pandoc_uninstall("latest")


# Install a specific pandoc version
```

```
pandoc_install("2.11.4")
pandoc_uninstall("2.11.4")


# Install last nightly build of pandoc
pandoc_install_nightly()
pandoc_uninstall("nightly")
```

---

pandoc_installed_versions

Check Pandoc versions already installed

Description

- `pandoc_installed_versions()` lists all versions already installed
- `pandoc_installed_latest()` returns the most recent installed version
- `pandoc_is_installed()` allows to check for a specific installed version

Usage

```
pandoc_installed_versions()

pandoc_installed_latest()

pandoc_is_installed(version, error = FALSE, ask = FALSE)
```

Arguments

version	This can be either:
	• `"latest"` for the latest release
	• A version number (e.g `"2.11.4"`) for a specific version
	• `"nightly"` for the last pandoc development built daily
error	if TRUE an error will be raised if the result is FALSE
ask	if TRUE, the user will be prompt in an interactive

Value

A character vector of installed versions or a logical for `pandoc_is_installed()`. It will return NULL is no versions are installed.

For `pandoc_is_installed()`, TRUE if only the required version is installed. If FALSE and ask is TRUE, the user will be prompt for installing the version.

Examples

```
pandoc_installed_versions()
pandoc_installed_latest()
pandoc_is_installed("2.19.2")
pandoc_installed_latest()
```

---

pandoc_is_active   *Is a pandoc version active ?*

Description

Is a pandoc version active ?

Usage

```
pandoc_is_active(version)
```

Arguments

version   This can be either:

- "latest" for the latest release
- A version number (e.g "2.11.4") for a specific version
- "nightly" for the last pandoc development built daily

Value

TRUE is the version provided is currently the active one (i.e the one used when special "default" keyword is used).

See Also

[pandoc_bin()](#)

Examples

```
# is Pandoc 2.19.2 active ?
pandoc_is_active("2.19.2")
# is it the Pandoc in PATH which is active ?
pandoc_is_active("system")
```

---

pandoc_list_abbreviations

List system default abbreviations

Description

Pandoc uses this list in the Markdown reader. Strings found in this list will be followed by a non-breaking space, and the period will not produce sentence-ending space in formats like LaTeX. The strings may not contain spaces.

Usage

```
pandoc_list_abbreviations(version = "default")
```

Arguments

version   Version to use. Default will be the "default" version. Other possible value are

- A version number e.g "2.14.1"
- The nightly version called "nightly"
- The latest installed version with "latest"
- Pandoc binary shipped with RStudio IDE with "rstudio"
- Pandoc binary found in PATH with "system"

Details

This correspond to the option --abbreviations as CLI flag.

Value

a character version of system default abbreviation known by Pandoc

Examples

```
pandoc_list_abbreviations()


# check abbreviations available in a specific Pandoc's version
pandoc_list_abbreviations("2.11.4")
```

---

pandoc_list_extensions

List supported extensions for a format

Description

Pandoc has a system of extensions to activate or deactivate some features. Each format have a set of activated by default extensions and other supported extensions than can be activated.

Usage

```
pandoc_list_extensions(format = "markdown", version = "default")
```

Arguments

format
: One for the supported input or output formats. See [pandoc_list_formats()](#). It corresponds to call

version
: Version to use. Default will be the "default" version. Other possible value are

  - A version number e.g "2.14.1"
  - The nightly version called "nightly"
  - The latest installed version with "latest"
  - Pandoc binary shipped with RStudio IDE with "rstudio"
  - Pandoc binary found in PATH with "system"

Details

All the extensions for the last Pandoc version released are available in https://pandoc.org/MANUAL.html.

Value

a data.frame (or a tibble if available) with 3 columns:

  - format: One of the Pandoc format
  - extensions : name of the extensions
  - default: Is the extensions activated by default or not ?

Examples

```
pandoc_list_extensions("markdown")
pandoc_list_extensions("gfm")


# target a specific version
pandoc_list_extensions("html", version = "system")
```

---

pandoc_list_formats   *List available supported formats*

Description

List available supported formats

Usage

```
pandoc_list_formats(type = c("input", "output"), version = "default")
```

Arguments

type   Either list input or output formats. It corresponds to call --list-input-formats
                and --list-output-formats respectively.

version   Version to use. Default will be the "default" version. Other possible value are

                • A version number e.g "2.14.1"
                • The nightly version called "nightly"
                • The latest installed version with "latest"
                • Pandoc binary shipped with RStudio IDE with "rstudio"
                • Pandoc binary found in PATH with "system"

Value

a data.frame (or a tibble if available) with 2 column:

• type (input or output)

• formats (name of the formats that can be used as input or output)

Examples

```
# which input formats are available
pandoc_list_formats()
# which output formats are available
pandoc_list_formats()


# target a specific version
pandoc_list_formats("input", version = "system")
```

---

pandoc_list_highlight_languages

List supported languages for Pandoc syntax highlighting

Description

This function is useful to retrieve the supported languages by Pandoc's syntax highlighter. These are the values that can be used as fenced code attributes to trigger the highlighting of the block for the requested language. See `fenced_code_attributes` extensions flag.

Usage

```
pandoc_list_highlight_languages(version = "default")
```

Arguments

version   Version to use. Default will be the "default" version. Other possible value are

- A version number e.g "2.14.1"
- The nightly version called "nightly"
- The latest installed version with "latest"
- Pandoc binary shipped with RStudio IDE with "rstudio"
- Pandoc binary found in PATH with "system"

Value

a character vector of supported languages to use as fenced code attributes.

Examples

```
pandoc_list_highlight_languages()


# check languages available in a specific Pandoc's version
pandoc_list_highlight_languages("2.11.4")
```

---

pandoc_list_highlight_style

List supported styles for Pandoc syntax highlighting

Description

Pandoc includes a highlighter which offer a styling mechanism to specify the coloring style to be used in highlighted source code. This function returns the supported values which can be specify at pandoc command line using the `--highlight-style=` flag.

Usage

```
pandoc_list_highlight_style(version = "default")
```

Arguments

version   Version to use. Default will be the `"default"` version. Other possible value are

- A version number e.g `"2.14.1"`
- The nightly version called `"nightly"`
- The latest installed version with `"latest"`
- Pandoc binary shipped with RStudio IDE with `"rstudio"`
- Pandoc binary found in PATH with `"system"`

Value

a character vector of supported highlighting style name to use.

Examples

```
pandoc_list_highlight_style()


# check style available in a specific Pandoc's version
pandoc_list_highlight_style("2.11.4")
```

---

pandoc_locate *Locate a specific Pandoc version installed with this package*

Description

This package helps install and manage Pandoc binaries in a specific folder. This function helps with finding the path to those specific versions of Pandoc. See `pandoc_bin()` for another way of getting paths to pandoc binaries

Usage

```
pandoc_locate(version = "default")
```

Arguments

version   Version to use. Default will be the `"default"` version. Other possible value are

- A version number e.g `"2.14.1"`
- The nightly version called `"nightly"`
- The latest installed version with `"latest"`
- Pandoc binary shipped with RStudio IDE with `"rstudio"`
- Pandoc binary found in PATH with `"system"`

Value

Path of Pandoc binaries root folder if version is available.

See Also

`pandoc_install()`

Examples

```
# where is the default active version located ?
pandoc_locate()

pandoc::pandoc_is_installed("2.11.4")
# where is a specific installed version located
pandoc_locate("2.11.4")
# return root folder of installed versions
pandoc_locate(NULL)
```

---

pandoc_run *Run the pandoc binary from R*

Description

This function is a thin wrapper around the pandoc binary and allow to pass any arguments supported by the Pandoc binary.

Usage

```
pandoc_run(args, version = "default")
```

Arguments

args   Character vector, arguments to the pandoc CLI command

version   Version to use. Default will be the "default" version. Other possible value are

- A version number e.g "2.14.1"
- The nightly version called "nightly"
- The latest installed version with "latest"
- Pandoc binary shipped with RStudio IDE with "rstudio"
- Pandoc binary found in PATH with "system"

Value

The output of running pandoc binary

Examples

```
# Run any command line argument (prefer `pandoc_convert()` for conversion )
pandoc::pandoc_run(c("--version"))


pandoc::pandoc_run(c("--list-input-formats"), version = "system")
```

---

pandoc_uninstall *Uninstall a Pandoc version*

Description

You can run [pandoc_installed_versions()](pandoc_installed_versions) to see which versions are currently installed on the system.

Usage

```
pandoc_uninstall(version)
```

Arguments

version   which version to uninstalled.

Value

TRUE (invisibly) if uninstalling is successful.

See Also

[pandoc_install()](#)

Examples

```
pandoc_install("2.19.2")
pandoc_is_installed("2.19.2")
pandoc_uninstall("2.19.2")
```

---

pandoc_version   *Get Pandoc version*

Description

This is calling pandoc --version to retrieve the version of Pandoc used. A special treatment is done for *nightly* version as Pandoc project does not use a development version scheme between released versions. This function will add a .9999 suffix to the version reported by Pandoc.

Usage

```
pandoc_version(version = "default")
```

Arguments

version   Version to use. Default will be the "default" version. Other possible value are

- A version number e.g "2.14.1"
- The nightly version called "nightly"
- The latest installed version with "latest"
- Pandoc binary shipped with RStudio IDE with "rstudio"
- Pandoc binary found in PATH with "system"

Value

The version number for pandoc binary as a [base::numeric_version()](#) object.

Examples

```
pandoc::pandoc_version()


pandoc::pandoc_version(version = "system")
```

---

rstudio_pandoc   *Retrieve path and version of Pandoc shipped with RStudio*

Description

RStudio IDE ships with a pandoc binary. The path is stored in RSTUDIO_PANDOC environment variable. Theses function are helper to easily use this specific version.

Usage

```
pandoc_rstudio_version()

pandoc_rstudio_bin()
```

Value

pandoc_rstudio_version() returns the version number for pandoc binary used by RStudio IDE as a [base::numeric_version()](#) object.

pandoc_system_bin() returns absolute path to the pandoc binary used by RStudio IDE.

See Also

[pandoc_version()](#), [pandoc_bin()](#)

Examples

```
pandoc_rstudio_bin()
```

---

system_pandoc *Retrieve path and version of Pandoc found on the system PATH*

Description

Pandoc can also be installed on a system and available through the PATH. Theses function are helper to easily use this specific version.

Usage

```
pandoc_system_version()

pandoc_system_bin()
```

Value

pandoc_system_version() returns the version number for pandoc binary found in PATH as a
base::numeric_version() object.

pandoc_system_bin() returns absolute path to the pandoc binary found in PATH.

See Also

pandoc_version(), pandoc_bin()

Examples

```
pandoc_system_bin()
```

---

with_pandoc_version *Execute any code with a specific Pandoc version*

Description

This function allows to run any R code by changing the active pandoc version to use without modifying the R session state.

Usage

```
with_pandoc_version(
  version,
  code,
  rmarkdown = getOption("pandoc.activate_rmarkdown", TRUE)
)

local_pandoc_version(
  version,
  rmarkdown = getOption("pandoc.activate_rmarkdown", TRUE),
  .local_envir = parent.frame()
)
```

Arguments

version	This can be either:

- `"latest"` for the latest release
- A version number (e.g `"2.11.4"`) for a specific version
- `"nightly"` for the last pandoc development built daily

code	Code to execute with the temporary active Pandoc version.
rmarkdown	if TRUE (the default) and **rmarkdown** is available, this will also set the pandoc version as the default one to use with **rmarkdown** by calling `rmarkdown::find_pandoc()`. Default behavior can be changed globally by setting option `pandoc.activate_rmarkdown`.
.local_envir	The environment to use for scoping.

Details

This is inspired from **withr** package.

Value

The results of the evaluation of the code argument.

Examples

```
# Run with pandoc without activating the version for rmarkdown::render()
with_pandoc_version("system",
  pandoc_bin(),
  rmarkdown = FALSE
)


with_pandoc_version("2.11.4", rmarkdown::find_pandoc(), rmarkdown = TRUE)


local({
  local_pandoc_version("2.11.4")
```

```
  pandoc::pandoc_locate()
  rmarkdown::find_pandoc()
})
rmarkdown::find_pandoc()
```

Index