

# Package: jimstools (via r-universe)

October 24, 2024

**Title** Tools for R

**Version** 0.4.0

**Description** What the package does (one paragraph).

**License** MIT + file LICENSE

**URL** <https://github.com/jimbrig/jimstools>

**BugReports** <https://github.com/jimbrig/jimstools/issues>

**Imports** attachment (>= 0.2.5.9000), checkmate, cli, crayon, data.table, devtools, dplyr, DT, fs, gert, glue, grDevices, here, htmltools, listviewer, lubridate, magrittr, openxlsx, purrr, qs, readr, rlang (>= 0.1.2), rstudioapi, shiny, stats, stringr, tcltk, tibble, tidyr, usethis, utils, whisker, whoami

**Suggests** covr, desc, knitr, rmarkdown, roxygen2, testthat

**VignetteBuilder** knitr

**Remotes** thinkR-open/attachment

**Config/testthat/edition** 3

**Encoding** UTF-8

**Language** en-US

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.2.1

**Repository** <https://jimbrig.r-universe.dev>

**RemoteUrl** <https://github.com/jimbrig/jimstools>

**RemoteRef** HEAD

**RemoteSha** 7c4be8d91857df6948990d174a146ebb39a26f0e

## Contents

|            |   |
|------------|---|
| aliases    | 3 |
| backup_raw | 4 |
| badge      | 4 |

|                                  |    |
|----------------------------------|----|
| browse_fontawesome . . . . .     | 5  |
| caching . . . . .                | 5  |
| coalesce_join . . . . .          | 6  |
| collapseRows . . . . .           | 7  |
| compare_to_prior . . . . .       | 7  |
| create_project . . . . .         | 8  |
| create_rstudio_project . . . . . | 8  |
| cs . . . . .                     | 9  |
| date_formats . . . . .           | 9  |
| default_colors . . . . .         | 10 |
| document_df . . . . .            | 10 |
| download_btn . . . . .           | 11 |
| editenv . . . . .                | 11 |
| editprof . . . . .               | 11 |
| expand_from_vars . . . . .       | 12 |
| extract_after . . . . .          | 12 |
| extract_before . . . . .         | 13 |
| extract_between . . . . .        | 13 |
| extract_date . . . . .           | 14 |
| extract_num . . . . .            | 14 |
| feedback . . . . .               | 15 |
| get_os . . . . .                 | 16 |
| get_rstudio_projects . . . . .   | 16 |
| group_by_occurrence . . . . .    | 17 |
| htmlspecialchars . . . . .       | 17 |
| indent . . . . .                 | 18 |
| inform . . . . .                 | 19 |
| in_rstudio . . . . .             | 20 |
| is_inst . . . . .                | 20 |
| is_valid_email . . . . .         | 21 |
| launch_rstudio . . . . .         | 21 |
| multi_filt . . . . .             | 22 |
| open_docs . . . . .              | 23 |
| open_gh_repo . . . . .           | 23 |
| open_in_rstudio . . . . .        | 23 |
| open_pkgdown . . . . .           | 24 |
| open_project . . . . .           | 24 |
| operators . . . . .              | 24 |
| paste_winslash . . . . .         | 25 |
| print_date . . . . .             | 26 |
| pull_unique . . . . .            | 26 |
| quick_table . . . . .            | 27 |
| random_password . . . . .        | 27 |
| render_template . . . . .        | 28 |
| save_pdf . . . . .               | 28 |
| search . . . . .                 | 29 |
| set_wd . . . . .                 | 31 |
| shiny_icons . . . . .            | 31 |

|                           |           |
|---------------------------|-----------|
| <i>aliases</i>            | 3         |
| shutdown . . . . .        | 32        |
| startup . . . . .         | 33        |
| user_info . . . . .       | 33        |
| use_gh_linguist . . . . . | 34        |
| view_list . . . . .       | 35        |
| <b>Index</b>              | <b>36</b> |

---

|         |                |
|---------|----------------|
| aliases | <i>Aliases</i> |
|---------|----------------|

---

### Description

These simple aliases can be useful to avoid operator precedence ambiguity, or to make use of indents from commas within your text editor. The all-caps versions accept single-length (capable of 'short-circuits') logical conditions only.

Neithers and nors are identical except have slightly different short-circuits. NOR uses negation once so may be quicker if the first argument is very, very prompt.

### Usage

AND(x, y)

OR(x, y)

nor(x, y)

neither(x, y)

NOR(x, y)

NEITHER(x, y)

pow()

XOR(x, y)

### Arguments

x, y                      Logical conditions.

---

|            |                        |
|------------|------------------------|
| backup_raw | <i>Backup Raw Data</i> |
|------------|------------------------|

---

### Description

This function copies raw data from an external location (i.e. a mapped network drive) to a specified local project directory. In addition, a snapshot of the files being copied is taken and stored in the specified "cache\_dir" so that files are only re-copied if changed.

### Usage

```
backup_raw(from_dir, to_dir, cache_dir = "cache", force = FALSE)
```

### Arguments

|           |   |
|-----------|---|
| from_dir  | Directory to copy from.                         |
| to_dir    | Directory to copy to.                           |
| cache_dir | Directory to store cached file snapshots.       |
| force     | Logical indicating if copying should be forced. |

### Value

Invisible - this function is used for its side effects

---

|       |  |
|-------|--|
| badge | <i>Create README badges via img.shields.io</i> |
|-------|--|

---

### Description

Create README badges via img.shields.io

### Usage

```
badge(label, value, color, url)
```

### Arguments

|       |             |
|-------|-------------|
| label | badge label |
| value | badge value |
| color | badge color |
| url   | url         |

### Value

character string to be pasted into README

---

|                    |                                       |
|--------------------|---------------------------------------|
| browse_fontawesome | <i>Open fontawesome icons webpage</i> |
|--------------------|---------------------------------------|

---

**Description**

Open fontawesome icons webpage

**Usage**

```
browse_fontawesome()
```

---

|         |                                  |
|---------|----------------------------------|
| caching | <i>Caching Utility Functions</i> |
|---------|----------------------------------|

---

**Description**

Quick caching utility read/write functions utilizing the qs package.

**Usage**

```
write_cache(x, name = NULL, cache_dir = "cache", overwrite = TRUE)
```

```
read_cache(x, name = NULL, cache_dir = "cache")
```

**Arguments**

|           |                            |
|-----------|----------------------------|
| x         | object to cache            |
| name      | name of object to read in. |
| cache_dir | path to cache directory.   |
| overwrite | logical (default = TRUE)   |

**Value**

invisibly returns object passed to the function.

x

invisibly attaches object to parent global environment

**See Also**

[qs::qread\(\)](#) and [qs::qsave\(\)](#)

**Examples**

```
## Not run:
mydata <- mtcars
write_cache(mydata) # will save to 'cache/mydata'.
write_cache(mydata, "mydata-v2", cache_dir = "data/temp") # will save to 'data/temp/mydata-v2'

# read back in
read_cache(mydata)

## End(Not run)
```

---

coalesce\_join

*Coalesce Join*


---

**Description**

Coalesce Join

**Usage**

```
coalesce_join(
  x,
  y,
  by = NULL,
  suffix = c(".x", ".y"),
  join = dplyr::full_join,
  ...
)
```

**Arguments**

|        |                         |
|--------|-------------------------|
| x      | data.frame              |
| y      | data.frame to join with |
| by     | variables to join by    |
| suffix | common variables suffix |
| join   | type of join            |
| ...    | passed to join function |

**Value**

joined data.frame with values replaced

---

|              |                      |
|--------------|----------------------|
| collapseRows | <i>Collapse Rows</i> |
|--------------|----------------------|

---

**Description**

for each group, sets the top row of the group to the group's value. All other rows in the group are set to "". See the example

**Usage**

```
collapseRows(df, variable)
```

**Arguments**

|          |                                |
|----------|--------------------------------|
| df       | a data frame                   |
| variable | group variable to be collapsed |

**Value**

a data frame with an updated variable column

---

|                  |                         |
|------------------|-------------------------|
| compare_to_prior | <i>Compare to Prior</i> |
|------------------|-------------------------|

---

**Description**

Compares data across evaluations.

**Usage**

```
compare_to_prior(data, col)
```

**Arguments**

|      |                                       |
|------|---------------------------------------|
| data | current data                          |
| col  | columns to compare (character vector) |

**Value**

new data frame with added columns for changes and priors

---

|                |                       |
|----------------|-----------------------|
| create_project | <i>Create Project</i> |
|----------------|-----------------------|

---

**Description**

Create Project

**Usage**

```
create_project(  
  path,  
  rstudio = rstudioapi::isAvailable(),  
  open = rlang::is_interactive(),  
  fields = list()  
)
```

**Arguments**

|         |  |
|---------|--|
| path    | Path   |
| rstudio | Boolean - Use RStudio?   |
| open    | Boolean - Open?  |
| fields  | passed to <a href="#">usethis::use_description()</a> 's fields argument. |

**Value**

RStudio project (side-effect)

---

|                        |                               |
|------------------------|-------------------------------|
| create_rstudio_project | <i>Create RStudio Project</i> |
|------------------------|-------------------------------|

---

**Description**

Create RStudio Project

**Usage**

```
create_rstudio_project(name, path = ".", ...)
```

**Arguments**

|      |  |
|------|--|
| name | name of project  |
| path | defaults to "."  |
| ...  | passed to <a href="#">render_template()</a> 's data argument |



**Value**

same as `usethis::create_package()`.

---

cs

*Vector of character strings from list of unquoted names*

---

**Description**

Vector of character strings from list of unquoted names

**Usage**

```
cs(...)
```

**Arguments**

... dots to quote and `c()`.  
Returns a vector of character strings from list of unquoted names. (Copied directly from the Hmisc package).

**Examples**

```
cs(a, cat, dog)
```

---

date\_formats

*Open Date Formatting man page.*

---

**Description**

Because I always forget.

**Usage**

```
date_formats()
```

---

|                |                                      |
|----------------|--------------------------------------|
| default_colors | <i>Vectors of colors for figures</i> |
|----------------|--------------------------------------|

---

**Description**

Creates different vectors of related colors that may be useful for figures.

**Usage**

```
default_colors()
```

**Value**

Vector of character strings representing colors, in RGB.

**Examples**

```
plot(1:4,rep(1,4),col=default_colors(),pch=19,cex=5)
plot(1:16,rep(1,16),col=default_colors(),pch=19,cex=5)
```

---

|             |  |
|-------------|--|
| document_df | <i>Prints framework for documenting a data frame</i> |
|-------------|--|

---

**Description**

Prints the standard fframework for documenting a data frame in an R package.

**Usage**

```
document_df(df, num_spaces = 4)
```

**Arguments**

|            |   |
|------------|---|
| df         | data frame object; can be a data.frame, data.table, or tibble |
| num_spaces | number of a spaces per tab; default is 4                      |

**Value**

nothing

**Examples**

```
set.seed(0)
dat <- data.frame(x = c(LETTERS[1:5]),
                  y = c(sample(1:100, 5)))
document_df(dat)
```

---

|              |                        |
|--------------|------------------------|
| download_btn | <i>Download Button</i> |
|--------------|------------------------|

---

**Description**

Download button without opening up new window in browser when downloaded.

**Usage**

```
download_btn(outputId, label = "Download", class = NULL, icon = NULL, ...)
```

**Arguments**

|          |                   |
|----------|-------------------|
| outputId | output id         |
| label    | label             |
| class    | class             |
| icon     | icon              |
| ...      | passed to tags\$a |

**Value**

download button HTML

---

|         |                |
|---------|----------------|
| editenv | <i>editenv</i> |
|---------|----------------|

---

**Description**

quickly edit .Renviron

**Usage**

```
editenv()
```

---

|          |                 |
|----------|-----------------|
| editprof | <i>editprof</i> |
|----------|-----------------|

---

**Description**

Quickly edit .Rprofile

**Usage**

```
editprof()
```

---

|                  |  |
|------------------|--|
| expand_from_vars | <i>Create a Skeleton 'Grid' for Data based on combinations of unique column values</i> |
|------------------|--|

---

**Description**

Create a Skeleton 'Grid' for Data based on combinations of unique column values

**Usage**

```
expand_from_vars(data, vars)
```

**Arguments**

|      |                  |
|------|------------------|
| data | data.frame       |
| vars | variables to use |

**Value**

expanded data.frame

---

|               |   |
|---------------|---|
| extract_after | <i>Extract all characters in a string after a match</i> |
|---------------|---|

---

**Description**

If a string has more than one match, this will only return the first one.

**Usage**

```
extract_after(full_string, after_string, trim_spaces = TRUE)
```

**Arguments**

|              |  |
|--------------|--|
| full_string  | The full character string to extract from.   |
| after_string | A pattern specifying the string should be extracted after this.  |
| trim_spaces  | logical T/F, if there are any spaces within the extracted string, they will be removed. Default is TRUE. |

**Value**

A string

**Examples**

```
extract_after("What does the fox say?", after_string = "What does the")
```

---

|                |  |
|----------------|--|
| extract_before | <i>Extract all characters in a string before a match</i> |
|----------------|--|

---

**Description**

If a string has more than one match, this will only return the first one.

**Usage**

```
extract_before(full_string, before_string, trim_spaces = TRUE)
```

**Arguments**

|               |  |
|---------------|--|
| full_string   | The full character string to extract from.   |
| before_string | A pattern specifying the string should be extracted before this.   |
| trim_spaces   | logical T/F, if there are any spaces within the extracted string, they will be removed. Default is TRUE. |

**Value**

A string

**Examples**

```
extract_before("What does the fox say?", before_string = "say")
```

---

|                 |   |
|-----------------|---|
| extract_between | <i>Extract a portion of a string between two patterns</i> |
|-----------------|---|

---

**Description**

If a string has more than one match, this will only return the first one.

**Usage**

```
extract_between(full_string, after_string, before_string, trim_spaces = TRUE)
```

**Arguments**

|               |  |
|---------------|--|
| full_string   | The full character string to extract from.   |
| after_string  | A pattern specifying the string should be extracted after this.  |
| before_string | A pattern specifying the string should be extracted before this.   |
| trim_spaces   | logical T/F, if there are any spaces within the extracted string, they will be removed. Default is TRUE. |

**Value**

A string

**Examples**

```
extract_between("What does the fox say?", after_string = "What does the", before_string = "say")
```

---

|              |                     |
|--------------|---------------------|
| extract_date | <i>Extract Date</i> |
|--------------|---------------------|

---

**Description**

Extracts dates from a character string with format "mm-dd-yyyy" or "mm/dd/yyyy".

**Usage**

```
extract_date(string)
```

**Arguments**

string            character string

**Value**

date object

**Examples**

```
extract_date("data-raw/lossrun-as-of-2019-01-31.xlsx")
```

---

|             |                                      |
|-------------|--------------------------------------|
| extract_num | <i>Extract numbers from a string</i> |
|-------------|--------------------------------------|

---

**Description**

Extract numbers from a string

**Usage**

```
extract_num(string)
```

**Arguments**

string            String to pull numbers from

**Value**

String of numbers

---

`feedback`*Feedback Message Utility Functions*

---

**Description**

A set of helper functions for providing verbose feedback to the developer using this packages functions.

This function allows a package developer to automatically add these functions to their package.

**Usage**

```
use_feedback_helpers(pkg = ".")  
  
msg_field(x)  
  
msg_value(x)  
  
msg_done(x)  
  
msg_bullet(x, bullet = cli::symbol$bullet)  
  
msg_err(x)  
  
msg_path(x)  
  
msg_info(x)  
  
msg_code(x)  
  
msg_feedback(x)
```

**Arguments**

|                     |  |
|---------------------|--|
| <code>pkg</code>    | Package - defaults to "."  |
| <code>x</code>      | The string passed to various <code>msg_</code> functions.                          |
| <code>bullet</code> | What to use for the message's bullet. Defaults to <code>cli::symbol\$bullet</code> |

**See Also**

- [usethis::ui-questions\(\)](#)
- [cli::list\\_symbols\(\)](#)

Other Feedback Utilities: [indent\(\)](#), [inform\(\)](#)

get\_os *Determine Operating System (OS)*

---

**Description**

Determine the operating system (OS) of your machine.

**Usage**

```
get_os()
```

```
is_32bit_os()
```

```
is_64bit_os()
```

**Value**

OS name in lower case: windows, mac, linux, etc.

**See Also**

<https://www.r-bloggers.com/identifying-the-os-from-r/>

**Examples**

```
get_os()
```

---

get\_rstudio\_projects *Get RStudio Projects*

---

**Description**

Pulls projects list from RStudio's AppData files.

**Usage**

```
get_rstudio_projects(exclude_nonexistent = TRUE)
```

**Arguments**

```
exclude_nonexistent  
Defaults to TRUE
```

**Value**

```
data.frame
```



---

group\_by\_occurrence    *group\_by\_occurrence*

---

**Description**

Group loss data by occurrence ID.

**Usage**

```
group_by_occurrence(dat)
```

**Arguments**

dat                    loss\_data by claim ID

**Value**

data grouped by occurrence

**Examples**

```
## Not run:
dat <- readRDS("data/shiny-data.rds") %>%
  filter(
    eval_dt == as.Date("2018-12-31"),
    member == "Ace Endico Corporation"
  )
test <- by_occurrence(dat)

## End(Not run)
```

---

htmlspecialchars            *Replace HTML special characters with HTML entities*

---

**Description**

The characters c("&", "'", '"', "<", ">") will be replaced with c("&amp;", "&quot;", "&#039;", "&lt;", "&gt;"), respectively.

**Usage**

```
htmlspecialchars(string)
```

**Arguments**

string                    the string with (or w/o) HTML special chars

**Value**

the string with special chars replaced.

**References**

<https://www.php.net/manual/en/function.htmlspecialchars.php>

**See Also**

[gsub](#)

**Examples**

```
htmlspecialchars("<a href = 'https://yihui.org'>Yihui</a>")  
# &lt;a href = &#039;https://yihui.org&#039;&gt;Yihui&lt;/a&gt;
```

---

indent

*Indent*

---

**Description**

Indentation around various msg\_ feedback functions.

**Usage**

```
indent(x, first = " ", indent = first)
```

**Arguments**

|        |  |
|--------|--|
| x      | The string passed to various msg_ functions. |
| first  | what to indent with - defaults to " ".       |
| indent | indentation of next line - defaults to first |

**Value**

string

**See Also**

Other Feedback Utilities: [feedback](#), [inform\(\)](#)

---

|        |               |
|--------|---------------|
| inform | <i>Inform</i> |
|--------|---------------|

---

## Description

A wrapper around `rlang::inform()` for providing feedback to developers using this packages functions.

## Usage

```
inform(...)
```

## Arguments

... Arguments passed on to `rlang::inform`

`message` The message to display, formatted as a **bulleted list**. The first element is displayed as an *alert* bullet prefixed with ! by default. Elements named "x", "i", "v", "x", and "!" are formatted as regular, info, success, failure, and error bullets respectively. See [Formatting messages with cli](#) for more about bulleted messaging.

If a message is not supplied, it is expected that the message is generated **lazily** through `cnd_header()` and `cnd_body()` methods. In that case, `class` must be supplied. Only `inform()` allows empty messages as it is occasionally useful to build user output incrementally.

If a function, it is stored in the header field of the error condition. This acts as a `cnd_header()` method that is invoked lazily when the error message is displayed.

`class` Subclass of the condition.

`body, footer` Additional bullets.

`use_cli_format` Whether to format message lazily using `cli` if available. This results in prettier and more accurate formatting of messages. See `local_use_cli()` to set this condition field by default in your package namespace.

If set to TRUE, message should be a character vector of individual and unformatted lines. Any newline character "\\n" already present in message is reformatted by cli's paragraph formatter. See [Formatting messages with cli](#).

`.file` A connection or a string specifying where to print the message. The default depends on the context, see the `stdout` vs `stderr` section.

`.subclass` **[Deprecated]** This argument was renamed to `class` in `rlang` 0.4.2 for consistency with our conventions for class constructors documented in <https://adv-r.hadley.nz/s3.html#s3-subclassing>.

`.frequency` How frequently should the warning or message be displayed? By default ("always") it is displayed at each time. If "regularly", it is displayed once every 8 hours. If "once", it is displayed once per session.

`.frequency_id` A unique identifier for the warning or message. This is used when `.frequency` is supplied to recognise recurring conditions. This argument must be supplied if `.frequency` is not set to "always".

**Value**

feedback in console

**See Also**

[rlang::inform\(\)](#)

Other Feedback Utilities: [feedback](#), [indent\(\)](#)

---

in\_rstudio

*In RStudio*

---

**Description**

Determine if user is in an RStudio Environment

**Usage**

in\_rstudio()

**Value**

logical

---

is\_inst

*is\_inst*

---

**Description**

Check if a package is installed.

**Usage**

is\_inst(pkg)

**Arguments**

pkg                    string: package to search installation path for

**Details**

This function provides a lightweight alternative to [utils::installed.packages\(\)](#) by using [nzchar](#) instead.

**Value**

logical (TRUE/FALSE)

**References**

- [Check for installed packages before running install.packages-Stack Overflow](#)

**Examples**

```
## Not run:  
is_inst("dplyr")  
  
## End(Not run)
```

---

|                |   |
|----------------|---|
| is_valid_email | <i>Ensures a text string is a valid email address</i> |
|----------------|---|

---

**Description**

Ensures a text string is a valid email address

**Usage**

```
is_valid_email(x)
```

**Arguments**

x                      string

**Value**

logical

---

|                |  |
|----------------|--|
| launch_rstudio | <i>Launch a new project-less RStudio session</i> |
|----------------|--|

---

**Description**

Call this function as an addin to launch a new project-less RStudio session without closing your currently open RStudio project(s).

**Usage**

```
launch_rstudio()
```

---

`multi_filt`*Multi Filt*

---

## Description

Filter across multiple variables.

## Usage

```
multi_filt(data, vars, filts)
```

## Arguments

|                    |           |
|--------------------|-----------|
| <code>data</code>  | data      |
| <code>vars</code>  | variables |
| <code>filts</code> | filters   |

## Details

NOTE: this function currently only supports character and factor level filtering by using the `dplyr::filter(<var> %in% c(<vars>))` syntax. For numeric and date level filtering will need to incorporate expressions like `<= 5` as filter arguments.

## Value

a filtered data.frame or tibble.

## References

[dplyr::filter\(\)](#) [rlang tidyeval tidyeval glue syntax](#)

## Examples

```
library(dplyr)
data <- datasets::iris
str(data)

iris_filtered <- multi_filt(data, vars = list("Species"), filts = list(c("setosa", "versicolor")))

str(iris_filtered) # filtered for only Species %in% c("setosa" and "versicolor")
```

---

|           |   |
|-----------|---|
| open_docs | <i>Open local version of pkgdown site for jimstools</i> |
|-----------|---|

---

**Description**

Open local version of pkgdown site for jimstools

**Usage**

```
open_docs()
```

---

|              |                                   |
|--------------|-----------------------------------|
| open_gh_repo | <i>Open Github Repository URL</i> |
|--------------|-----------------------------------|

---

**Description**

Open Github Repository URL

**Usage**

```
open_gh_repo()
```

---

|                 |                             |
|-----------------|-----------------------------|
| open_in_rstudio | <i>Open file in RStudio</i> |
|-----------------|-----------------------------|

---

**Description**

Function tries opening a file in RStudio.

**Usage**

```
open_in_rstudio(path, ...)
```

**Arguments**

|      |   |
|------|---|
| path | (string) Path to file.  |
| ...  | Further arguments to <a href="#">rstudioapi::navigateToFile()</a> . |

**See Also**

- [rstudioapi::navigateToFile\(\)](#),
- [fs::file\\_show\(\)](#), [browseURL\(\)](#),
- [utils::file.edit\(\)](#)

---

|              |   |
|--------------|---|
| open_pkgdown | <i>Open pkgdown site of the package</i> |
|--------------|---|

---

**Description**

Open pkgdown site of the package

**Usage**

```
open_pkgdown()
```

---

|              |                             |
|--------------|-----------------------------|
| open_project | <i>Open RStudio Project</i> |
|--------------|-----------------------------|

---

**Description**

Open RStudio Project

**Usage**

```
open_project()
```

**Value**

invisibly returns an RStudio project

---

|           |                  |
|-----------|------------------|
| operators | <i>Operators</i> |
|-----------|------------------|

---

**Description**

Extensions of base operators.

inverse of %in%

**Usage**

```
x %notin% vector
```

```
x %allin% vector
```

```
x %anyin% vector
```

```
x %nonein% vector
```

```
pattern %partin% vector
```



**Arguments**

|         |   |
|---------|---|
| x       | Numeric or character vector.  |
| vector  | Numeric or character vector.  |
| pattern | Character string containing <b>regular expressions</b> to be matched. |

**Value**

logical  
logical

**See Also**

[base::any\(\)](#), [base::match\(\)](#), and [base::all\(\)](#)  
[base::grepl\(\)](#)

**Examples**

```
data <- data.frame(
  id = 1:10,
  x = as.integer(runif(10, 0, 10))
)

data[data$id %notin% c(1, 3, 5, 7, 9), ]

1:2 %allin% 1:3 # TRUE
3:4 %allin% 1:3 # FALSE
3:4 %anyin% 1:3 # TRUE
4:5 %anyin% 1:3 # FALSE
3:4 %nonein% 1:3 # FALSE
4:5 %nonein% 1:3 # TRUE
"Bei" %partin% c("Beijing", "Shanghai") # TRUE
"bei" %partin% c("Beijing", "Shanghai") # FALSE
"[aeiou]ng" %partin% c("Beijing", "Shanghai") # TRUE
```

---

paste\_winslash

*Paste Winslash - Addin Function*

---

**Description**

Paste Winslash - Addin Function

**Usage**

```
paste_winslash()
```

**Details**

assign to keyboard shortcut Ctrl + Shift + P for easy access.

**Value**

normalized path character string

---

|            |  |
|------------|--|
| print_date | <i>Print the current date in a pretty format</i> |
|------------|--|

---

**Description**

Print the current date in a pretty format

**Usage**

```
print_date()
```

**Value**

string

**Examples**

```
print_date()
```

---

|             |  |
|-------------|--|
| pull_unique | <i>Pull all unique values for a variable</i> |
|-------------|--|

---

**Description**

Pull all unique values for a variable

**Usage**

```
pull_unique(data, var, sort = TRUE, decreasing = FALSE, names = TRUE)
```

**Arguments**

|            |                           |
|------------|---------------------------|
| data       | data.frame                |
| var        | variable name             |
| sort       | logical (default = TRUE)  |
| decreasing | logical (default = FALSE) |
| names      | logical (default = TRUE)  |

**Value**

vector

---

|             |  |
|-------------|--|
| quick_table | <i>htable</i> Create a quick html table using my favorites options for <a href="#">DT::datatable()</a> |
|-------------|--|

---

**Description**

htable Create a quick html table using my favorites options for [DT::datatable\(\)](#)

**Usage**

```
quick_table(df, ...)
```

**Arguments**

|     |   |
|-----|---|
| df  | data.frame to display                                 |
| ... | options sent through to <a href="#">DT::datatable</a> |

**Examples**

```
quick_table(mtcars)
```

---

|                 |   |
|-----------------|---|
| random_password | <i>Generate a random password with a specified length</i> |
|-----------------|---|

---

**Description**

This function generates a random password sampled from the ASCII table.

**Usage**

```
random_password(length = 12, replace = FALSE, extended = TRUE)
```

**Arguments**

|          |  |
|----------|--|
| length   | length of the password   |
| replace  | sample from the ASCII table with (TRUE) or without (FALSE) replacement?            |
| extended | if FALSE, use alphanumeric characters only; otherwise use all the ASCII characters |

**Value**

a character string

**See Also**

[sample](#)

**Examples**

```

random_password()
# set the seed to get fixed password every time; you may just remember the
# seed and forget the real password because it's reproducible
set.seed(123)
random_password()
# long password
random_password(20, TRUE)

```

---

|                 |                        |
|-----------------|------------------------|
| render_template | <i>Render Template</i> |
|-----------------|------------------------|

---

**Description**

Renders a template using `whisker::whisker.render()` for templates used in a fashion similar to `usethis::use_template()`.

**Usage**

```
render_template(template_path, out_path, data = list(), open = TRUE)
```

**Arguments**

|               |  |
|---------------|--|
| template_path | Path to template                                       |
| out_path      | Path to output the rendered template                   |
| data          | list to pass to <code>whisker::whisker.render()</code> |
| open          | boolean - open template after it renders?              |

**Value**

invisible

---

|          |                 |
|----------|-----------------|
| save_pdf | <i>save_pdf</i> |
|----------|-----------------|

---

**Description**

Copies the graphics contents of current device to PDF (a wrapper for `dev2.pdf`). Default size is 8.5 x 11 in landscape mode.

**Usage**

```
save_pdf(file, width = 11, height = 8.5, jpg = FALSE)
```

**Arguments**

|        |  |
|--------|--|
| file   | filename to save the image                   |
| width  | width of pdf image                           |
| height | height of pdf image                          |
| jpg    | convert to 300 dpi .jpg file after pdf save? |

---

search

*Generalized Search and Search Engine Examples*

---

**Description**

Use these functions to search the web directly from your browser, building advanced queries and supplying common useful R related domains.

Query Github's internal search engine.

Query a search on [rseek.org](https://rseek.org).

Query a search on the R Nabble Forum. Nabble is an innovative search engine for R messages.

**Usage**

```
search_online(s, query_url = "https://google.com/search?q=")
```

```
search_gh(  
  s,  
  type = "all",  
  language = NULL,  
  topic = NULL,  
  user = NULL,  
  org = NULL  
)
```

```
search_rseek(s)
```

```
search_finzi(s)
```

```
search_nabble(s)
```

```
search_google(s)
```

```
search_rproject(s)
```

```
search_metacran(s)
```

## Arguments

|           |  |
|-----------|--|
| s         | string to search for                                     |
| query_url | string representing the URL to query; defaults to Google |
| type      | what to search for, see details for options              |
| language  | optional language filter                                 |
| topic     | optional topic filter                                    |
| user      | optional user filter                                     |
| org       | optional org filter                                      |

## Details

See <https://cloud.r-project.org/search.html> which showcases Google's advanced search feature to query only R-Project domain sites via the [Google Search Engine](#).

## Functions

- `search_gh()`:
- `search_rseek()`:
- `search_finzi()`:
- `search_nabble()`:
- `search_google()`:
- `search_rproject()`:
- `search_metacran()`:

## References

- <http://www.sashagoodman.com/>
- <http://finzi.psych.upenn.edu/search.html>
- <http://finzi.psych.upenn.edu/search/manual.html#query>
- <https://cloud.r-project.org/search.html>
- <http://n4.nabble.com/help/Answer.jtp?id=31>

## Examples

```
# default search on google
search_online("polished.tech")

# search github (note: &ref=opensearch)
search_online("polished", "https://github.com/search?q=%s&ref=opensearch")

# search Github with language:r, org:tychobra for 'polished' (note: '%3A' represents a ':')
search_online("polished", "https://github.com/search?q=%s+language%3Ar+org%3Atychobra")
search_gh("websocket", language = "r", topic = "rshiny")

# search your org
search_gh("polished", org = "tychobra")
```

---

|        |   |
|--------|---|
| set_wd | <i>Set working directory to the path of <b>current</b> script</i> |
|--------|---|

---

**Description**

Set working directory to the path of **current** script

**Usage**

```
set_wd(dir = NULL)
```

**Arguments**

|     |   |
|-----|---|
| dir | NULL (default) or a character string specifying the working directory.<br>If NULL, set working directory to the path of <b>the current R script</b> . |
|-----|---|

**See Also**

[setwd](#)

**Examples**

```
## Not run:  
set_wd() # set working directory to the path of the current R script  
  
set_wd("D:/") # "\" is not allowed, you should use "/"  
  
set_wd("../") # set working directory to the parent directory  
  
## End(Not run)
```

---

|             |                    |
|-------------|--------------------|
| shiny_icons | <i>Shiny Icons</i> |
|-------------|--------------------|

---

**Description**

Shiny Icons

**Usage**

```
shiny_icons()
```

**Value**

character vector of shiny package icons

shutdown

*Shut down the operating system with the command 'shutdown'*

---

**Description**

There is a command shutdown in both Windows and Linux, and this function uses it to shut down a computer.

**Usage**

```
shutdown(wait = 0)
```

**Arguments**

`wait`                    time to wait before shutting down (in seconds); passed to [Sys.sleep](#)

**Details**

After the time `wait` has passed, R will execute `shutdown -s -t 0` (for Windows) or `shutdown -h now` to shut down the computer.

**Value**

The status code of [system](#).

**See Also**

[system](#), [Sys.sleep](#)

**Examples**

```
if (interactive()) {  
  # when your code is extremely time-consuming, you may need this function;  
  # e.g. you go to sleep, and R is running long computation...  
  # complex graphics... and long long computation... at last,  
  shutdown()  
  # the next day you wake up, "thank you, R" :)  
}
```



---

startup

*R Startup Functions*

---

**Description**

Use these utility functions to quickly edit your .Rprofile, .Renviron, or other startup files.

**Usage**

```
editrprof(scope = c("user", "project"))
```

```
editrenv(scope = c("user", "project"))
```

**Arguments**

scope            scope - user or project

**Value**

invisible

**See Also**

See [base::Startup\(\)](#) for details on the R specific startup process.

---

user\_info

*Get User-Related Information*

---

**Description**

Get user-related information.

**Usage**

```
user_info()
```

**Value**

invisibly returns os\_info vector

**Examples**

```
user_info()
```

---

use\_gh\_linguist      *GitHub Linguist Exclusions*

---

### Description

This function creates a `.gitattributes` file in the root of your project in order to exclude certain file types from being detected as a certain programming language on GitHub. See details for more information.

### Usage

```
use_gh_linguist(excludes = c("*.js", "*.html", "*.css"))
```

### Arguments

`excludes`      Character vector of wildcards indicating extensions to be labelled as linguist-vendored and ignored by GitHub's language detector `github-linguist`.

### Details

By default this function will write `*.js`, `*.html`, and `*.css` to a `.gitattributes` file in order for `github-linguist` to ignore Javascript, HTML, and CSS languages in your repositories language section.

The resulting file will look like so:

```
# github linguist exclusions:
*.js linguist-vendored
*.html linguist-vendored
*.css linguist-vendored
```

### Value

invisibly returns the text written to `.gitattributes`

### Examples

```
## Not run:
use_gh_linguist()

## End(Not run)
```

---

`view_list`*view\_list*

---

**Description**

Interactive view of list structures with `listviewer::jsonedit()`

**Usage**

```
view_list(list, ...)
```

**Arguments**

|                   |   |
|-------------------|---|
| <code>list</code> | list to display   |
| <code>...</code>  | options sent through to <code>listviewer::jsonedit()</code> |

**Examples**

```
view_list(mtcars)
```

# Index

- \* **Feedback Utilities**
  - feedback, 15
  - indent, 18
  - inform, 19
- \* **actuarial**
  - compare\_to\_prior, 7
  - group\_by\_occurrence, 17
- \* **feedback**
  - feedback, 15
- \* **open files**
  - open\_in\_rstudio, 23
- \* **open**
  - open\_in\_rstudio, 23
- \* **operators**
  - operators, 24
- \* **paths and dirs**
  - open\_in\_rstudio, 23
- \* **search engines**
  - search, 29
- \* **system**
  - get\_os, 16
- \* **utilities**
  - default\_colors, 10
- %allin% (operators), 24
- %anyin% (operators), 24
- %nonein% (operators), 24
- %notin% (operators), 24
- %partin% (operators), 24
- aliases, 3
- AND (aliases), 3
  
- backup\_raw, 4
- badge, 4
- base::all(), 25
- base::any(), 25
- base::grepl(), 25
- base::match(), 25
- base::Startup(), 33
- browse\_fontawesome, 5
  
- browseURL(), 23
  
- caching, 5
- cli::list\_symbols(), 15
- cnd\_body(), 19
- cnd\_header(), 19
- coalesce\_join, 6
- collapseRows, 7
- compare\_to\_prior, 7
- create\_project, 8
- create\_rstudio\_project, 8
- cs, 9
  
- date\_formats, 9
- default\_colors, 10
- document\_df, 10
- download\_btn, 11
- dplyr::filter(), 22
- DT::datatable, 27
- DT::datatable(), 27
  
- editenv, 11
- editprof, 11
- editrenv (startup), 33
- editrprof (startup), 33
- expand\_from\_vars, 12
- extract\_after, 12
- extract\_before, 13
- extract\_between, 13
- extract\_date, 14
- extract\_num, 14
  
- feedback, 15, 18, 20
- Formatting messages with cli, 19
- fs::file\_show(), 23
  
- get\_os, 16
- get\_rstudio\_projects, 16
- group\_by\_occurrence, 17
- gsub, 18

htmlspecialchars, 17

in\_rstudio, 20

indent, 15, 18, 20

inform, 15, 18, 19

is\_32bit\_os (get\_os), 16

is\_64bit\_os (get\_os), 16

is\_inst, 20

is\_valid\_email, 21

launch\_rstudio, 21

listviewer::jsonedit(), 35

local\_use\_cli(), 19

msg\_bullet (feedback), 15

msg\_code (feedback), 15

msg\_done (feedback), 15

msg\_err (feedback), 15

msg\_feedback (feedback), 15

msg\_field (feedback), 15

msg\_info (feedback), 15

msg\_path (feedback), 15

msg\_value (feedback), 15

multi\_filt, 22

NEITHER (aliases), 3

neither (aliases), 3

NOR (aliases), 3

nor (aliases), 3

nzchar, 20

open\_docs, 23

open\_gh\_repo, 23

open\_in\_rstudio, 23

open\_pkgdown, 24

open\_project, 24

operators, 24

OR (aliases), 3

paste\_winslash, 25

pow (aliases), 3

print\_date, 26

pull\_unique, 26

qs::qread(), 5

qs::qsave(), 5

quick\_table, 27

random\_password, 27

read\_cache (caching), 5

render\_template, 28

render\_template(), 8

rlang::inform, 19

rlang::inform(), 19, 20

rstudioapi::navigateToFile(), 23

sample, 27

save\_pdf, 28

search, 29

search\_finzi (search), 29

search\_gh (search), 29

search\_google (search), 29

search\_metacran (search), 29

search\_nabble (search), 29

search\_online (search), 29

search\_rproject (search), 29

search\_rseek (search), 29

set\_wd, 31

setwd, 31

shiny\_icons, 31

shutdown, 32

startup, 33

Sys.sleep, 32

system, 32

use\_feedback\_helpers (feedback), 15

use\_gh\_linguist, 34

user\_info, 33

usethis::create\_package(), 9

usethis::use\_description(), 8

usethis::use\_template(), 28

utils::file.edit(), 23

utils::installed.packages(), 20

view\_list, 35

whisker::whisker.render(), 28

write\_cache (caching), 5

XOR (aliases), 3