Package 'k5'

January 24, 2023

Title Kiernan Nicholls Miscellaneous

Version 0.0.5

Description Quality of life functions for interactive programming. Shortcuts for common combinations of functions or different default arguments. Not to be used in production level scripts, but useful for exploring and quickly manipulating data for easy analysis. Also imports a variety of packages to facilitate the installation of those imported packages on the host machine.

License GPL-3

URL https://github.com/kiernann/k5, https://kiernann.github.io/k5/

BugReports https://github.com/kiernann/k5/issues

Depends R (>= 2.10)

Imports aws.s3 (>= 0.3.21), clipr (>= 0.8.0), dplyr (>= 1.0.10), fs (>= 1.5.2), ggplot2 (>= 3.4.0), glue (>= 1.6.2), lubridate (>= 1.9.0), magrittr (>= 2.0.3), pacman (>= 0.5.1), readr (>= 2.1.3), stringr (>= 1.5.0), tibble (>= 3.1.8), usethis (>= 2.1.6), utils

Suggests covr (>= 3.6.1), crayon (>= 1.5.2), gluedown (>= 1.0.6), here (>= 1.0.1), httr (>= 1.4.4), janitor (>= 2.1.0), knitr (>= 1.41), readxl (>= 1.4.1), rvest (>= 1.0.3), scales (>= 1.2.1), testthat (>= 3.1.6)

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

NeedsCompilation no

Author Kiernan Nicholls [aut, cre]

Maintainer Kiernan Nicholls <kiernann@protonmail.com>

Repository CRAN

Date/Publication 2023-01-24 10:50:02 UTC

R topics documented:

`aws_info` *List all objects in an AWS bucket*

Description

`aws_info()` uses `aws.s3::get_bucket_df()` to return a tibble of stored objects in a remote bucket, like `fs::dir_info()` returns information on a local directory.

`aws_ls()` also uses `aws.s3::get_bucket_df()` but only returns the filenames within the bucket as a named `fs_path` character vector, like `fs::dir_ls()`) returns files in a local directory.

Usage

```
aws_info(bucket = aws_bucket(), prefix = NULL, max = Inf, ...)

aws_ls(bucket = aws_bucket(), prefix = NULL, ...)

aws_bucket(bucket = getOption("aws.bucket"), set = FALSE)
```

Arguments

bucket	Character string with the name of the bucket. If you use the same bucket frequently, you can set a default through an option named that can be retrieved with aws_bucket().
prefix	Character string that limits the response to keys that begin with the specified prefix.
max	Number of objects to return.
...	Additional arguments passed to aws.s3::s3HTTP().
set	If TRUE, print instructions for setting the option.

Value

A list of objects on the AWS bucket.

Examples

```
aws_info("1000genomes", max = 10)
aws_ls("irs-form-990", max = 1)
```

---

contract_convert *Convert contract names to factor intervals*

Description

Can perform one of three **rough** conversions:

1. For interval contracts (e.g., "220 - 229", "9% or more", etc.), convert the character strings to proper interval notation.
2. For contracts with multiple discrete outcomes (e.g., Candidate names), convert the character vector to simple factors.
3. For markets with a single binary question (e.g., "Will the Democrats have a brokered convention in 2020?"), contracts returned are always "Yes" which is converted to TRUE.

Usage

```
contract_convert(x, decimal = FALSE)
```

Arguments

x	A character vector of contract names.
decimal	Should percentages be converted to decimals?

Value

A interval factor, unique factor, or logical vector.

---

copy_last *Copy the last object to the clipboard*

Description

Use `clipr::write_clip()` to write the last value as a character vector to the system clipboard.

Usage

```
copy_last(x = .Last.value)
```

Arguments

x	The object to view, usually left as base::.Last.value.

Details

The value of the internal evaluation of a top-level R expression is always assigned to `.Last.value` before further processing (e.g., printing).

Value

The same `.Last.value` as *before* copied, invisibly.

---

count2 *Count the way you want*

Description

A wrapper around `dplyr::count()` with `sort` set to `TRUE` by default and the an additional column created by default containing the proportional fraction each observation makes of the whole.

Usage

```
count2(x, ..., wt = NULL, sort = TRUE, prop = TRUE, sum = NULL)

count_vec(x, sort = TRUE, prop = TRUE, sum = NULL)
```

Arguments

x	A data frame.
...	Variables to group by.
wt	Frequency weights.
sort	If TRUE, will show the largest groups at the top.
prop	If TRUE, compute the fraction of marginal table.
sum	Column to replace with a cumulative sum (n, p, or np).

Value

A tibble of element counts

Examples

```
count2(iris, Species)
```

---

count_diff *Count set difference*

Description

Find the length of the set of difference between x and y vectors.

Usage

```
count_diff(x, y, ignore.case = FALSE)
```

Arguments

x	A vector to check.
y	A vector to compare against.
ignore.case	logical; if FALSE, the pattern matching is case sensitive and if TRUE, case is ignored during matching.

Details

```
sum(x %out% y)
```

Value

The number of *unique* values of x not in y.

See Also

Other counting wrappers: count_in(), count_na(), count_out(), na_in(), na_out(), na_rep(), prop_distinct(), prop_in(), prop_na(), prop_out(), what_in(), what_out()

Examples

```
# only unique values are checked
count_diff(c("VT", "NH", "ZZ", "ZZ", "ME"), state.abb)
```

`count_in` *Count in*

Description

Count the total values of `x` that are `%in%` the vector `y`.

Usage

```
count_in(x, y, na.rm = TRUE, ignore.case = FALSE)
```

Arguments

`x`	A vector to check.
`y`	A vector to compare against.
`na.rm`	logical; Should `NA` be ignored?
`ignore.case`	logical; if `FALSE`, the pattern matching is case sensitive and if `TRUE`, case is ignored during matching.

Details

`sum(x %out% y)`

Value

The sum of `x` present in `y`.

See Also

Other counting wrappers: `count_diff()`, `count_na()`, `count_out()`, `na_in()`, `na_out()`, `na_rep()`, `prop_distinct()`, `prop_in()`, `prop_na()`, `prop_out()`, `what_in()`, `what_out()`

Examples

```
count_in(c("VT", "NH", "ZZ", "ME"), state.abb)
```

count_na *Count missing*

Description

Count the total values of `x` that are `NA`.

Usage

```
count_na(x)
```

Arguments

`x`	A vector to check.

Details

```
sum(is.na(x))
```

Value

The sum of `x` that are `NA`

See Also

Other counting wrappers: `count_diff()`, `count_in()`, `count_out()`, `na_in()`, `na_out()`, `na_rep()`, `prop_distinct()`, `prop_in()`, `prop_na()`, `prop_out()`, `what_in()`, `what_out()`

Examples

```
count_na(c("VT", "NH", NA, "ME"))
```

count_out *Count out*

Description

Count the total values of `x` that are are `%out%` of the vector `y`.

Usage

```
count_out(x, y, na.rm = TRUE, ignore.case = FALSE)
```

Arguments

x	A vector to check.
y	A vector to compare against.
na.rm	logical; Should NA be ignored?
ignore.case	logical; if FALSE, the pattern matching is case sensitive and if TRUE, case is ignored during matching.

Details

```
sum(x %out% y)
```

Value

The sum of x absent in y.

See Also

Other counting wrappers: count_diff(), count_in(), count_na(), na_in(), na_out(), na_rep(), prop_distinct(), prop_in(), prop_na(), prop_out(), what_in(), what_out()

Examples

```
count_out(c("VT", "NH", "ZZ", "ME"), state.abb)
```

---

file_age *File modification date age*

Description

The period of time since a system file was modified.

Usage

```
file_age(...)
```

Arguments

...	Arguments passed to file.info(), namely character vectors containing file paths. Tilde-expansion is done: see path.expand().

Value

A Period class object.

Examples

```
file_age(system.file("README.md", package = "campfin"))
```

file_encoding *File Encoding*

Description

Call the `file` command line tool with option `-i`.

Usage

```
file_encoding(path)
```

Arguments

`path`	A local file path or glob to check.

Value

A tibble of file encoding.

filter_rx *Filter a data frame by a regular expression*

Description

A shortcut for `dat %>% filter(str_detect(column, "\\d"))`.

Usage

```
filter_rx(dat, col, pattern, ...)
```

Arguments

`dat`	A data frame with a character column to filter.
`col`	The column containing a character vector to input.
`pattern`	Pattern to look for..
`...`	Additional arguments passed to `stringr::str_detect()`.

Value

A subset of rows from `dat`.

gaa *GAA Team Abbreviations by Season and Team ID*

Description

GAA Team Abbreviations by Season and Team ID

Usage

```
gaa
```

Format

A data frame with 74 rows and 3 variables:

seasonId The fantasy season integer

teamId The team ID integer

abbrev The normalized *owner* abbreviation for that year ...

last_value *Return the last value*

Description

A function shortcut for accessing .Last.value.

Usage

```
last_value(x = .Last.value)
```

Arguments

x	The object to return, usually left as base::.Last.value.

Details

The value of the internal evaluation of a top-level R expression is always assigned to .Last.value (in package:base) before further processing (e.g., printing).

Value

The same `.Last.value` as *before* viewing, invisibly.

load.packages *Save and load packages from file*

Description

Save and load packages from file

Usage

```
load.packages(path = NULL, install = FALSE)

save.packages(x = NULL, path = tempfile())
```

Arguments

path	The path to a text file containing package names. If NULL (default), then the default list is read from k5/inst/PACKAGES.
install	If TRUE, install missing packages.
x	A character vector of package names to save. If NULL (default), use all currently attached packages.

Value

The list of packages, invisibly.

na_in *Remove in*

Description

Set NA for the values of x that are %in% the vector y.

Usage

```
na_in(x, y, ignore.case = FALSE)
```

Arguments

x	A vector to check.
y	A vector to compare against.
ignore.case	logical; if FALSE, the pattern matching is case sensitive and if TRUE, case is ignored during matching.

Value

The vector x missing any values in y.

See Also

Other counting wrappers: count_diff(), count_in(), count_na(), count_out(), na_out(), na_rep(), prop_distinct(), prop_in(), prop_na(), prop_out(), what_in(), what_out()

Examples

```
na_in(c("VT", "NH", "ZZ", "ME"), state.abb)
na_in(1:10, seq(1, 10, 2))
```

---

na_out *Remove out*

Description

Set NA for the values of x that are %out% of the vector y.

Usage

```
na_out(x, y, ignore.case = FALSE)
```

Arguments

x	A vector to check.
y	A vector to compare against.
ignore.case	logical; if FALSE, the pattern matching is case sensitive and if TRUE, case is ignored during matching.

Value

The vector x missing any values not in y.

See Also

Other counting wrappers: count_diff(), count_in(), count_na(), count_out(), na_in(), na_rep(), prop_distinct(), prop_in(), prop_na(), prop_out(), what_in(), what_out()

Examples

```
na_out(c("VT", "NH", "ZZ", "ME"), state.abb)
na_out(1:10, seq(1, 10, 2))
```

na_rep *Remove repeated character elements*

Description

Set NA for the values of x that contain a single repeating character and no other characters.

Usage

```
na_rep(x, n = 0)
```

Arguments

x	A vector to check.
n	The minimum number times a character must repeat. If 0, the default, then any string of one character will be replaced with NA. If greater than 0, the string must contain greater than n number of repetitions.

Details

Uses the regular expression `"^(.)\\1+$"`.

Value

The vector x with NA replacing repeating character values.

See Also

Other counting wrappers: `count_diff()`, `count_in()`, `count_na()`, `count_out()`, `na_in()`, `na_out()`, `prop_distinct()`, `prop_in()`, `prop_na()`, `prop_out()`, `what_in()`, `what_out()`

Examples

```
na_rep(c("VT", "NH", "ZZ", "ME"))
```

print_all *Print all rows of elements*

Description

Print up to the `getOption("max.print")` and ask the user if they want to print more than that. This is most useful when printing tibbles with more than 10 rows but less than `getOption("max.print")`.

Usage

```
print_all(x, ask = TRUE)
```

Arguments

x	Object to print, typically a data frame or vector.
ask	If the length of x exceeds getOption("max.print"), should the user be promoted confirm their intention to print everything. If FALSE, the maximum is printed without double checking: this can be **extremely** slow. The 'usethis' package must be installed for interactive confirmation.

Value

The object x (invisibly)

---

prop_distinct *Proportion missing*

Description

Find the proportion of values of x that are distinct.

Usage

```
prop_distinct(x)
```

Arguments

x	A vector to check.

Details

```
length(unique(x))/length(x)
```

Value

The ratio of distinct values x to total values of x.

See Also

Other counting wrappers: count_diff(), count_in(), count_na(), count_out(), na_in(), na_out(), na_rep(), prop_in(), prop_na(), prop_out(), what_in(), what_out()

Examples

```
prop_distinct(c("VT", "VT", NA, "ME"))
```

prop_in *Proportion in*

Description

Find the proportion of values of x that are %in% the vector y.

Usage

```
prop_in(x, y, na.rm = TRUE, ignore.case = FALSE)
```

Arguments

x	A vector to check.
y	A vector to compare against.
na.rm	logical; Should NA be ignored?
ignore.case	logical; if FALSE, the pattern matching is case sensitive and if TRUE, case is ignored during matching.

Details

```
mean(x %in% y)
```

Value

The proportion of x present in y.

See Also

Other counting wrappers: count_diff(), count_in(), count_na(), count_out(), na_in(), na_out(), na_rep(), prop_distinct(), prop_na(), prop_out(), what_in(), what_out()

Examples

```
prop_in(c("VT", "NH", "ZZ", "ME"), state.abb)
```

prop_na *Proportion missing*

Description

Find the proportion of values of x that are NA.

Usage

```
prop_na(x)
```

Arguments

x	A vector to check.

Details

```
mean(is.na(x))
```

Value

The proportion of values of x that are NA.

See Also

Other counting wrappers: count_diff(), count_in(), count_na(), count_out(), na_in(), na_out(), na_rep(), prop_distinct(), prop_in(), prop_out(), what_in(), what_out()

Examples

```
prop_na(c("VT", "NH", NA, "ME"))
```

prop_out *Proportion out*

Description

Find the proportion of values of x that are %out% of the vector y.

Usage

```
prop_out(x, y, na.rm = TRUE, ignore.case = FALSE)
```

Arguments

x	A vector to check.
y	A vector to compare against.
na.rm	logical; Should NA be ignored?
ignore.case	logical; if FALSE, the pattern matching is case sensitive and if TRUE, case is ignored during matching.

Details

```
mean(x %out% y)
```

Value

The proportion of x absent in y.

See Also

Other counting wrappers: count_diff(), count_in(), count_na(), count_out(), na_in(), na_out(), na_rep(), prop_distinct(), prop_in(), prop_na(), what_in(), what_out()

Examples

```
prop_out(c("VT", "NH", "ZZ", "ME"), state.abb)
```

---

read_delim_clip *Read a table from the clipboard*

Description

Use readr::read_delim() on a string copied to the clipboard. Defaults to tab separator like given when copying cells from spreadsheets.

Usage

```
read_delim_clip(delim = "\t", ...)
```

Arguments

delim	Single character used to separate fields within a record.
...	Additional arguments passed to readr::read_delim().

Value

A data frame read from the clipboard.

read_delim_dumb *Read a text file without column guessing*

Description

Use `readr::read_delim()` without specifying *any* column types. All columns are treated as character strings.

Usage

```
read_delim_dumb(file, delim = c(",", "\t", "|"), ...)

read_csv_dumb(file, ...)

read_tsv_dumb(file, ...)
```

Arguments

`file`	Either a path to a file, a connection, or literal data.
`delim`	Single character used to separate fields within a record.
`...`	Additional arguments passed to `readr::read_delim()`.

Value

A tibble data frame read from the file.

view_firefox *View an HTML document in Firefox*

Description

Take an XML document object, write to an HTML file, and open in Firefox.

Usage

```
view_firefox(html)
```

Arguments

`html`	An object which has the class `xml_document`, often from rvest.

Value

The html object, invisibly.

view_last *View the last object*

Description

Invoke a spreadsheet-style data viewer on a matrix-like R object. In a non-interactive session, the object is returned invisibly and nothing is "viewed".

Usage

```
view_last(x = .Last.value)
```

Arguments

x	The object to view, usually left as base::.Last.value.

Details

The value of the internal evaluation of a top-level R expression is always assigned to `.Last.value` before further processing (e.g., printing).

Value

The same `.Last.value` as *before* viewing, invisibly.

what_in *Which in*

Description

Return the values of `x` that are `%in%` of the vector `y`.

Usage

```
what_in(x, y, ignore.case = FALSE)
```

Arguments

x	A vector to check.
y	A vector to compare against.
ignore.case	logical; if `FALSE`, the pattern matching is case sensitive and if `TRUE`, case is ignored during matching.

Details

```
x[which(x %in% y)]
```

Value

The elements of x that are %in% y.

See Also

Other counting wrappers: count_diff(), count_in(), count_na(), count_out(), na_in(), na_out(), na_rep(), prop_distinct(), prop_in(), prop_na(), prop_out(), what_out()

Examples

```
what_in(c("VT", "DC", NA), state.abb)
```

---

what_out *Which out*

Description

Return the values of x that are %out% of the vector y.

Usage

```
what_out(x, y, na.rm = TRUE, ignore.case = FALSE)
```

Arguments

x	A vector to check.
y	A vector to compare against.
na.rm	logical; Should NA be ignored?
ignore.case	logical; if FALSE, the pattern matching is case sensitive and if TRUE, case is ignored during matching.

Details

```
x[which(x %out% y)]
```

Value

The elements of x that are %out% y.

See Also

Other counting wrappers: count_diff(), count_in(), count_na(), count_out(), na_in(), na_out(), na_rep(), prop_distinct(), prop_in(), prop_na(), prop_out(), what_in()

Examples

```
what_out(c("VT", "DC", NA), state.abb)
```

word_count *Count file words, lines, and bytes*

Description

Invoke system tool to print newline, word, and byte counts for each file.

Usage

```
word_count(path, count = "")
```

Arguments

path	Character vector of file paths.
count	The type of element to count, see details.

Details

One of five options or an empty string (default):

1. "lines" for newline characters (separating lines).
2. "words" for words separated by white space.
3. "chars" for individual characters.
4. "bytes" for total bytes, differs with multibyte characters.
5. "max" for the maximum display width of longest line.

Value

A data frame of counts by file.

write_delim_clip *Write a table from the clipboard*

Description

Use `readr::format_delim()` on a data frame to copy a string to the clipboard. Defaults to tab separator like given when copying cells from spreadsheets.

Usage

```
write_delim_clip(x, delim = "\t", ...)
```

Arguments

x	A data frame to write to clipboard.
delim	Single character used to separate fields within a record.
...	Additional arguments passed to readr::format_delim().

Value

Invisibly, the input data frame.

---

write_last *Write the last value to disk*

Description

The value of the internal evaluation of a top-level R expression is always assigned to .Last.value before further processing (e.g., printing).

Usage

```
write_last(file = tempfile(), x = .Last.value, ...)

save_last(file = tempfile(), x = .Last.value, ...)
```

Arguments

file	File or connection to write to.
x	The object to write, usually left as base::.Last.value.
...	Additional arguments passed to the writing function (see Details).

Details

Four types of files are written, based on object class:

1. For data frames, a tab-separated file via readr::write_tsv().
2. For vectors, a newline-separated file via readr::write_lines().
3. For ggplots, a raster image (by default) via ggplot2::ggsave().
4. For other objects, an uncompressed data file via readr::write_rds().

Value

The created file path, invisibly.

`%out%` *Inverted match*

Description

`%out%` is an inverted version of the infix `%in%` operator.

Usage

```
x %out% table
```

Arguments

`x`	vector: the values to be matched. Long vectors are supported.
`table`	vector or `NULL`: the values to be matched against.

Details

`%out%` is currently defined as `"%out%" <- function(x, table) match(x, table, nomatch = 0) == 0`

Value

logical; if `x` is not present in `table`

Examples

```
c("A", "B", "3") %out% LETTERS
```

Index