

Package ‘basepenguins’

July 22, 2025

Title Convert Files that Use 'palmerpenguins' to Work with 'datasets'

Version 0.1.0

Description From 'R' 4.5.0, the 'datasets' package includes the penguins and penguins_raw data sets popularised in the 'palmerpenguins' package. 'basepenguins' takes files that use the 'palmerpenguins' package and converts them to work with the versions from 'datasets' ('R' >= 4.5.0). It does this by removing calls to library(palmerpenguins) and making the necessary changes to column names. Additionally, it provides helper functions to define new files paths for saving the output and a directory of example files to experiment with.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2.9000

URL <https://github.com/EllaKaye/basepenguins>,
<https://ellakaye.github.io/basepenguins/>

BugReports <https://github.com/EllaKaye/basepenguins/issues>

Depends R (>= 4.2.0)

Suggests knitr, rmarkdown, testthat (>= 3.0.0), withr

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation no

Author Ella Kaye [aut, cre, cph] (ORCID:
<<https://orcid.org/0000-0002-7300-3718>>),
Heather Turner [aut] (ORCID: <<https://orcid.org/0000-0002-1256-3375>>),
Achim Zeileis [ctb] (ORCID: <<https://orcid.org/0000-0003-0918-3766>>)

Maintainer Ella Kaye <ella.kaye@gmail.com>

Repository CRAN

Date/Publication 2025-04-08 08:40:06 UTC

Contents

convert_files	2
example_files	4
files_to_convert	6
output_paths	7
Index	8

convert_files	<i>Convert files to use datasets versions of penguins and penguins_raw</i>
---------------	----------------------------------------------------------------------------

Description

These functions convert files that use the **palmerpenguins** package to use the versions of penguins and penguins_raw included in the datasets package in R >= 4.5.0. They removes calls to library(palmerpenguins) and make necessary changes to some variable names (see Details section below).

Usage

```
convert_files(input, output, extensions = c("R", "r", "qmd", "rmd", "Rmd"))
convert_files_inplace(input, extensions = c("R", "r", "qmd", "rmd", "Rmd"))
convert_dir(input, output, extensions = c("R", "r", "qmd", "rmd", "Rmd"))
convert_dir_inplace(input, extensions = c("R", "r", "qmd", "rmd", "Rmd"))
```

Arguments

input	For convert_files() and convert_files_inplace(): A character vector of file paths to convert. Can be relative or absolute paths. For convert_dir() and convert_dir_inplace(): A string with a path to a directory of files to convert.
output	For convert_files(): A character vector of output file paths, or NULL`` to modify files in place. I put. Can be absolute or relative paths. For convert_dir(): A string with the output director to modify the files in the directory in place.
extensions	A character vector of file extensions to process, defaults to R scripts and R Mark-down and Quarto documents.

Details

Files are converted by:

- Replacing library(palmerpenguins) with the empty string ""
- Replacing data("penguins", package = "palmerpenguins") with data("penguins", package = "datasets")
- Replacing variable names:

```

- bill_length_mm -> bill_len
- bill_depth_mm -> bill_dep
- flipper_length_mm -> flipper_len
- body_mass_g -> body_mass

```

- Replacing `ends_with("_mm")` with `starts_with("flipper_")`, `starts_with("bill_")`

Non-convertible files (those without the specified extensions) are copied to the output location if output is provided, but are not modified.

If the output files or directory do not (yet) exist, they will be created (recursively if necessary).

Replacing `ends_with("_mm")` with `starts_with("flipper_")`, `starts_with("bill_")` ensures that modified R code will always run. `starts_with("flipper_")` isn't intuitively necessary, as there is only one variable starting with "flipper_", in penguins, but this code will not error inside `dplyr::(select)`, even if `flipper_len` isn't in the data frame (trying to select `flipper_len` directly will cause an error if that column isn't in the data frame). In an educational context, we suggest manually editing the converted files to replace `starts_with("flipper_")` to `flipper_len` if appropriate. To facilitate this, the functions documented here produce a message indicating the files and line numbers where the `ends_with("_mm")` substitution was made.

Value

A list (returned invisibly) with two components:

- `changed`: A character vector of paths for files that were modified.
- `not_changed`: A character vector of paths for files that were not modified. Files are not changed if they do not load the `palmerpenguins` package via `library(palmerpenguins)`, `library('palmerpenguins')` or `library("palmerpenguins")`, or if they do not have one of the specified extensions.

For both the `changed` and `not_changed` vectors, these will be subsets of the output paths, if they were provided, with the corresponding input paths as names. If output was not specified, then these vectors will be subsets of the input paths, and the vectors will not be named.

See Also

`example_dir()`, `output_paths()`

Examples

```

# Note that all examples below write output to a temporary directory
# and file paths are relative to that directory.

# For all examples below, use a copy of the examples provided by the package,
# copied to an "examples" directory in the working directory
example_dir("examples")

# Single file - new output
result <- convert_files("examples/penguins.R", "penguins_new.R")
cat(readLines("penguins_new.R"), sep = "\n") # view changes

```

```
# Single file - copy, then modify that in place
file.copy("examples/penguins.R", "penguins_copy.R")
convert_files_inplace("penguins_copy.R")

# Multiple files - new output locations
input_files <- c("examples/penguins.R", "examples/nested/penguins.qmd")
output_files <- output_paths(input_files, dir = "new_dir")
convert_files(input_files, output_files)

# Directory - new output location
result <- convert_dir("examples", "new_directory")
result # see `changed` and `not_changed` files

# Overwrite the files in "examples"
result <- convert_dir_inplace("examples")
result # see `changed` and `not_changed` files
```

example_files

List or find example files from basepenguins package

Description

These functions provides access to example files included with the basepenguins package. When `example_files()` is called with `path = NULL`, it lists available example files. When called with a specific path, it returns the full path to that file. `example_dir()` provides the path to the directory containing the examples, and also takes a `copy.dir` argument which, if specified, will copy all the example files to a new directory. This is useful for testing the `convert_dir_inplace()` and `convert_files_inplace()` functions without overwriting package files.

Usage

```
example_files(path = NULL, full.names = FALSE, recursive = TRUE)
```

```
example_dir(copy.dir = NULL)
```

Arguments

<code>path</code>	Character string. If <code>NULL</code> (default), lists all available example files. If specified, returns the full path to the specified file or directory.
<code>full.names</code>	Logical. If <code>TRUE</code> , returns full file paths rather than relative paths. Only used when <code>path = NULL</code> . Default is <code>FALSE</code> .
<code>recursive</code>	Logical. If <code>TRUE</code> , lists files in subdirectories recursively. Only used when <code>path = NULL</code> . Default is <code>TRUE</code> .
<code>copy.dir</code>	Character string. A directory name or path to a directory into which the files in the example directory will be copied. If <code>NULL</code> (default) the files will not be copied.

Details

There are four example files in the example directory:

- penguins.R - an R script using the palmerpenguins package
- no_penguins.Rmd - an Rmarkdown file with `ends_with("_mm")`, but *not* in the context of the palmerpenguins package
- nested/penguins.qmd - a Quarto document that uses the palmerpenguins package
- nested/not_a_script.md - contains `library(palmerpenguins)`, but is not a script, i.e. not one of the default extensions for the convert functions.

Value

`example_files`:

- If `path = NULL`, a character vector of available file/directory names.
- If `path` is specified, the absolute path to the requested file.

`example_dir`:

- If `copy.dir = NULL`, the absolute path to the directory containing all examples.
- If `copy.dir` is specified, the specified directory is created if it doesn't already exist, and all example files are copied into it, preserving nesting structure. The absolute path to the directory is returned invisibly.

Examples

```
# List all files in the example directory provided by the package
example_files()

# Get the full path to a specific example files
example_files("penguins.R") # path/to/basepenguins/extdata/penguins.R
example_files("nested/penguins.qmd")

# Get the path to the directory containing the example files
example_dir() # path/to/basepenguins/extdata/

# Copy all files in the example directory
example_dir(".") # copy example files into working directory
example_dir("examples") # create subdirectory
```

files_to_convert	<i>List files to convert in a directory</i>
------------------	---------------------------------------------

Description

This function lists all files in a directory (and subdirectories) that match the specified file extensions. It can be used as a helper to find files paths to pass to `convert_files()` and `convert_files_inplace()`, or to preview which files those functions, as well as `convert_dir()` and `convert_dir_inplace()` will look to convert. It can also be used as input to `output_paths()` to help generate output paths for new files.

Usage

```
files_to_convert(
  dir,
  full.names = FALSE,
  extensions = c("R", "r", "qmd", "rmd", "Rmd")
)
```

Arguments

<code>dir</code>	A character string specifying the directory path to search
<code>full.names</code>	Logical. If TRUE, returns full file paths rather than relative paths. Default is FALSE.
<code>extensions</code>	A character vector of file extensions to filter by. Default is <code>c("R", "r", "qmd", "rmd", "Rmd")</code> . If NULL or empty, returns all files.

Value

A character vector of file paths that match the specified extensions.

See Also

[convert_files\(\)](#), [convert_files_inplace\(\)](#), [convert_dir\(\)](#), [convert_dir_inplace\(\)](#), [output_paths\(\)](#).

Examples

```
example_dir <- example_dir() # Get examples directory
files_to_convert(example_dir)
files_to_convert(example_dir, full.names = TRUE)
files_to_convert(example_dir, extensions = "R")
files_to_convert(example_dir, extensions = NULL) # all files
```

output_paths	<i>Generate modified file paths by adding prefixes and/or suffixes</i>
--------------	------------------------------------------------------------------------

Description

This function takes a vector of file paths and returns a vector of modified paths with prefixes and/or suffixes added to the filenames. It's useful for generating output paths for the `convert_files()` and `convert_dir()` functions. `files_to_convert()` is useful for generating the input paths.

Usage

```
output_paths(paths, prefix = "", suffix = "_new", dir = NULL)
```

Arguments

paths	A character vector of file paths to modify
prefix	A character string to add at the beginning of each filename. Default is the empty string "".
suffix	A character string to add at the end of each filename, before the extension. Default is "_new".
dir	An optional character string specifying the output directory. If provided, the modified filenames will be placed in this directory. This is useful if paths are relative and a different output directory is required. Default is NULL.

Value

A character vector of modified file paths with the specified prefixes and suffixes. The original paths are preserved as names of the returned vector.

See Also

[convert_files\(\)](#), [files_to_convert\(\)](#)

Examples

```
# Get all convertible files from examples directory
input_files <- files_to_convert(example_dir())

# Generate output paths with "_converted" suffix
output_paths(input_files, suffix = "_converted")

# Add both prefix and suffix and place in a new directory
output_paths(
  input_files,
  prefix = "base_",
  suffix = "_converted",
  dir = "~/new_dir"
)
```

Index

`convert_dir (convert_files), 2`
`convert_dir()`, [6](#)
`convert_dir_inplace (convert_files), 2`
`convert_dir_inplace()`, [6](#)
`convert_files, 2`
`convert_files()`, [6](#), [7](#)
`convert_files_inplace (convert_files), 2`
`convert_files_inplace()`, [6](#)

`example_dir (example_files), 4`
`example_dir()`, [3](#)
`example_files, 4`

`files_to_convert, 6`
`files_to_convert()`, [7](#)

`output_paths, 7`
`output_paths()`, [3](#), [6](#)