

Package: dht (via r-universe)

September 13, 2024

Title A Collection of Functions to Assist Building DeGAUSS Containers

Version 1.2.3

Description degauss helper tools are used to develop and run DeGAUSS containers.

License GPL (>= 3)

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.2

Suggests testthat (>= 3.0.0), sf (>= 0.8-1), renv (>= 0.8.3), tibble, knitr, rmarkdown

Imports cli (>= 2.2.0), magrittr, readr (>= 2.0.0), dplyr (>= 1.0.0), tidyrr (>= 1.0.0), glue (>= 1.4.2), fs (>= 1.5.0), withr (>= 2.2.0), purrr (>= 0.3.4), stringr (>= 1.4.0), whisker (>= 0.4), ps (>= 1.6.0), prettyunits, tidyselect

Depends R (>= 2.10)

Config/testthat/edition 3

URL <http://degauss.org/dht>

VignetteBuilder knitr

BugReports <https://github.com/degauss-org/dht/issues>

Repository <https://degauss-org.r-universe.dev>

RemoteUrl <https://github.com/degauss-org/dht>

RemoteRef HEAD

RemoteSha f0b59bbb91697a55074d065642ed012b652e7d31

Contents

address_is_institutional	2
address_is_nonaddress	3

address_is_po_box	3
check_dates	4
check_end_after_start_date	4
check_for_column	5
check_ram	6
clean_address	6
core_lib_images	7
create_degauss_menu_data	7
degauss_colors	8
degauss_menu	8
degauss_run	9
expand_dates	9
find_docker	10
get_degauss_core_lib_env	10
get_degauss_env_dockerfile	11
greeting	12
has_docker	13
make_degauss_command	13
qlibrary	14
read_lat_lon_csv	14
use_degauss_compose	15
use_degauss_container	16
write_geomarker_file	17
Index	19

address_is_institutional

check if address is a known Cincinnati institutional address

Description

check if address is a known Cincinnati institutional address

Usage

address_is_institutional(address)

Arguments

address character vector of address text

Details

Print the function to view the source and the complete list of addresses considered to be Cincinnati insitutional addresses; Note that addresses in other cities might be erroneously categorized as institutional (e.g., "3333 Burnet Ave Syracuse NY 13206")

Value

logical vector; TRUE when address contains some text indicating Cincinnati Children's Hospital, Ronald McDonald House, Jobs and Family Services for Hamilton and Butler Counties in Ohio, or Stetson Square

address_is_nonaddress *check if address text is not actually an address*

Description

check if address text is not actually an address

Usage

```
address_is_nonaddress(address)
```

Arguments

address character vector of address text

Value

logical vector; TRUE when address text is "verify", "foreign", "foreign country", "unknown", or blank.

address_is_po_box *check if address is a PO Box*

Description

check if address is a PO Box

Usage

```
address_is_po_box(address)
```

Arguments

address character vector of address text

Value

logical vector; TRUE when address text contains some variation of "PO Box"

`check_dates`*check format of dates from DeGAUSS container input file*

Description

check format of dates from DeGAUSS container input file

Usage

```
check_dates(date, allow_missing = FALSE)
```

Arguments

`date` vector of dates to be checked for formatting

`allow_missing` logical. defaults to FALSE, resulting in an error if any dates are missing.

Details

ISO formatted dates (i.e., "%Y-%m-%d" or YYYY-MM-DD) will stay the same U.S. standard slash formatted dates (common to Microsoft Excel; e.g., "%m/%d/%y" or MM/DD/YY, "%m/%d/%Y" or MM/DD/YYYY) will be reformatted to ISO format Any unrecognized input will cause an error and the user will be instructed to reformat their dates.

Value

reformatted vector of dates, or an error if dates could not be reformatted

Examples

```
date <- c("1/1/21", "1/2/21", "1/3/21")
check_dates(date)
```

`check_end_after_start_date`*check that end_date occurs after start_date*

Description

check that end_date occurs after start_date

Usage

```
check_end_after_start_date(start_date, end_date)
```

Arguments

start_date vector of start dates
end_date vector of end dates

Examples

```
## Not run:
start_date <- check_dates(c("1/1/21", "1/2/21", "1/3/21"))
end_date <- check_dates(c("1/7/21", "1/8/21", "1/9/20"))
check_end_after_start_date(start_date, end_date)

## End(Not run)
```

check_for_column	<i>check for specified columns and corresponding column types</i>
------------------	---

Description

check for specified columns and corresponding column types

Usage

```
check_for_column(d, column_name, column, column_type = NULL)
```

Arguments

d input dataframe
column_name character string defining name of column to be checked
column character vector to be checked (e.g., d\$column_name)
column_type (optional) desired column type to be checked for (e.g., 'character')

Value

if column_name exists in d and is of the correct column_type, nothing is returned. if column_name does not exist in d, an error is thrown. if column is not of the correct column_type, a warning is shown.

Examples

```
## Not run:
d <- tibble::tribble(
  ~"id", ~"value",
  "123", 123,
  "456", 456
)
check_for_column(d, "id", d$id, "double")
check_for_column(d, "id2", d$id2, "double")
```

```
## End(Not run)
```

check_ram	<i>check_ram</i>
-----------	------------------

Description

Checks for amount of system RAM and warns a potential DeGAUSS user if it might be too low

Usage

```
check_ram(minimum_ram = 4)
```

Arguments

minimum_ram	minimum recommended GB of RAM (as numeric value)
-------------	--

clean_address	<i>clean address text strings for geocoding</i>
---------------	---

Description

clean address text strings for geocoding

Usage

```
clean_address(address)
```

Arguments

address	character vector of address text
---------	----------------------------------

Value

vector of character strings with non-alphanumerics (except dashes, which are left in for +4 ZIP issues) and excess white space removed.

core_lib_images *list the DeGAUSS images in the core library*

Description

list the DeGAUSS images in the core library

Usage

```
core_lib_images(geocoder = TRUE)
```

Arguments

geocoder logical; include "geocoder" in core image list?

Value

names of DeGAUSS images in the core library as a character vector

Examples

```
core_lib_images()
core_lib_images(geocoder = FALSE)
```

create_degauss_menu_data
create data for use in DeGAUSS menu

Description

create data for use in DeGAUSS menu

Usage

```
create_degauss_menu_data(core_lib_env = get_degauss_core_lib_env())
```

Arguments

core_lib_env a data.frame of info about the DeGAUSS core image library created with get_degauss_core_lib_env()

Value

data.frame of information about core images with arguments separated into names and default values as well as an added example DeGAUSS command

Examples

```
dht:::create_degauss_menu_data()
```

degauss_colors	<i>Access the DeGAUSS color palette</i>
----------------	---

Description

Access the DeGAUSS color palette

Usage

```
degauss_colors(n)
```

Arguments

n which DeGAUSS color(s): 1-darkblue, 2-lightblue, 3-pink, 4-lightgrey, 5-purple, 6-teal)

Value

a named character string or vector of named character strings containing RGB colors in hexadecimal

Examples

```
degauss_colors(2)
degauss_colors(1:4)
plot(1:6, rep(1, 6), col = degauss_colors(1:6), pch = 19, cex = 10)
```

degauss_menu	<i>DeGAUSS Menu</i>
--------------	---------------------

Description

Run an interactive shiny application to find geomarkers available within DeGAUSS based on categories and input data characteristics. At launch, it will download the latest information about DeGAUSS images in the core library. Suggested DeGAUSS commands are automatically created and displayed for use.

Usage

```
degauss_menu()
```

degauss_run	<i>run a DeGAUSS container</i>
-------------	--------------------------------

Description

This function uses temporary CSV files and DeGAUSS commands as system calls to docker. Because of this approach, caching of geocoding results or reuse of intermediate downloaded data files are not possible, *unless called from the same R session*. See the examples for a workaround.

Usage

```
degauss_run(.x, image, version = "latest", argument = NA, quiet = FALSE)
```

Arguments

.x	a data.frame or tibble to be input to a DeGAUSS container
image	name of DeGAUSS image
version	version of DeGAUSS image; will use latest version if not specified
argument	optional argument
quiet	suppress output from DeGAUSS container?

Value

.x with additional returned DeGAUSS columns

Examples

```
## create a memoised version of degauss_run so repetitive calls are cached
## this can be useful during development of DeGAUSS pipelines
## Not run:
fc <- memoise::cache_filesystem(fs::path(fs::path_wd(), "data-raw"))
degauss_run <- memoise::memoise(degauss_run, omit_args = c("quiet"), cache = fc)

## End(Not run)
```

expand_dates	<i>expand dates between start_date and end_date</i>
--------------	---

Description

expand dates between start_date and end_date

Usage

```
expand_dates(d, by)
```

Arguments

d data.frame or tibble with columns called 'start_date' and 'end_date'
 by time interval to expand dates (e.g., 'day', 'week', etc)

Value

long data.frame or tibble with column called 'date' including all dates between start_date and end_date

Examples

```
## Not run:
d <- data.frame(
  start_date = check_dates(c("1/1/21", "1/2/21", "1/3/21")),
  end_date = check_dates(c("1/7/21", "1/8/21", "1/9/21"))
)
expand_dates(d, by = "day")

## End(Not run)
```

find_docker	<i>find the path to the docker executable</i>
-------------	---

Description

Error if docker cannot be found or if the docker daemon is not running in the background.

Usage

```
find_docker()
```

Value

path to Docker executable found using Sys.which("docker")

get_degauss_core_lib_env	<i>get DeGAUSS metadata on all images in the Rhrefhttps://degauss.org/available_imagescore library</i>
--------------------------	---

Description

get DeGAUSS metadata on all images in the **core library**

Usage

```
get_degauss_core_lib_env(...)
```

Arguments

```
... arguments passed to core_lib_images()
```

Value

data.frame of DeGAUSS metadata

Examples

```
get_degauss_core_lib_env(geocoder = FALSE)
```

```
get_degauss_env_dockerfile
```

get DeGAUSS metadata online or from a Dockerfile

Description

These functions look in a Dockerfile (locally or online) to extract environment variables corresponding to DeGAUSS image metadata.

Usage

```
get_degauss_env_dockerfile(  
  dockerfile_path = fs::path_join(c(getwd(), "Dockerfile"))  
)  
  
get_degauss_env_online(name = "fortunes")
```

Arguments

```
dockerfile_path  
                path to Dockerfile  
  
name            name of DeGAUSS container to download Dockerfile from
```

Details

Metadata on DeGAUSS images are defined using environment variables. Specifically within a Dockerfile, this is defined as ENV instructions where the name of the environment variable begins with `degauss_`, for example `"degauss_name"`, or `"degauss_version"`. It is assumed that each ENV instruction is on its own line and defines only one environment variable.

Value

named vector of DeGAUSS metadata

Examples

```
## Not run:
use_degauss_dockerfile(version = "0.1")
get_degauss_env_dockerfile()
get_degauss_env_dockerfile()["degauss_version"]

## End(Not run)
get_degauss_env_online("fortunes")
get_degauss_env_online("fortunes")["degauss_version"]
```

greeting

display DeGAUSS greeting message in console

Description

if not supplied as arguments, greeting-specific values (geomarker_name, version, description) are read in from the environment variables specified in the Dockerfile and made available when running the container; these include degauss_name, degauss_version, and degauss_description

Usage

```
greeting(
  geomarker_name = Sys.getenv("degauss_name"),
  version = Sys.getenv("degauss_version"),
  description = Sys.getenv("degauss_description")
)
```

Arguments

geomarker_name name of the geomarker, must be the name used in the degauss.org url
 version container version number as a character string
 description brief description of the container; finishes the sentence "This container returns..."

Details

greeting message includes name, version, and brief description of container, as well as a link to more information about the specific geomarker

Examples

```
## Not run:
greeting("roads", "0.4", "returns proximity and length of nearby major roadways")

## End(Not run)
```

has_docker	<i>is docker available?</i>
------------	-----------------------------

Description

is docker available?

Usage

has_docker()

Value

TRUE if find_docker() succeeds; FALSE otherwise

make_degauss_command	<i>create a Rhrefhttps://degauss.org/using_degauss.html#DeGAUSS_CommandsDeGAUSS command</i>
----------------------	---

Description

create a **DeGAUSS command**

Usage

```
make_degauss_command(
  input_file = "my_address_file_geocoded.csv",
  image,
  version = "latest",
  argument = NA,
  docker_cmd = find_docker()
)
```

Arguments

input_file	name of input file
image	name of DeGAUSS image
version	version of DeGAUSS image
argument	optional argument
docker_cmd	path to docker executable

Value

DeGAUSS command as a character string

Examples

```
make_degauss_command(image = "geocoder", version = "3.2.0", docker_cmd = "docker")
make_degauss_command(image = "geocoder", version = "3.2.0", argument = "0.4", docker_cmd = "docker")
make_degauss_command(image = "geocoder", version = "3.2.0", docker_cmd = "/usr/local/bin/docker")
```

qlibrary	<i>wrapper for base::library() that automatically supresses package startup messages</i>
----------	--

Description

note that renv will not pickup dependencies loaded using this function and it is recommended to use something like `withr::with_message_sink("/dev/null", library(dplyr))` instead

Usage

```
qlibrary(...)
```

Arguments

... arguments passed to `base::library()`

Examples

```
## Not run:
qlibrary(dplyr)

## End(Not run)
```

read_lat_lon_csv	<i>read in and format input file for DeGAUSS container</i>
------------------	--

Description

read in and format input file for DeGAUSS container

Usage

```
read_lat_lon_csv(
  filename,
  nest_df = FALSE,
  sf_out = FALSE,
  project_to_crs = NULL
)
```

Arguments

filename	name of input file, probably opt\$filename if inside container
nest_df	logical. If TRUE, data is nested on lat/lon. Defaults to FALSE.
sf_out	logical. If TRUE, data is converted as an sf object. Defaults to FALSE.
project_to_crs	(optional) if sf_out=TRUE, the crs to which input data is projected. If unspecified and sf_out=TRUE, the crs defaults to 4326.

Value

a list with two elements. The first is the raw_data as it is read in from the input file. The second is a tibble nested on lat and lon to prevent duplication of geomarker computations. If sf_out=TRUE the second is an sf object.

Examples

```
## Not run:
d <- read_lat_lon_csv(filename = "test/my_address_file_geocoded.csv")
d <- read_lat_lon_csv(
  filename = "test/my_address_file_geocoded.csv",
  sf_out = TRUE, project_to_crs = 5072
)

## End(Not run)
```

use_degauss_compose *use DeGAUSS compose file*

Description

creates a docker-compose yaml file in current working directory

Usage

```
use_degauss_compose(...)
```

Arguments

... arguments passed to render_degauss_template (overwrite)

use_degauss_container *use DeGAUSS container template*

Description

Creates all the necessary files to create a DeGAUSS container. The container/geomarker name is assumed to be the basename of the working directory and the version of R and renv is taken from the calling environment. This function calls all of the individual `dht::use_degauss_*`() functions to create the following:

- Dockerfile
- Makefile
- README.md
- entrypoint.R
- .dockerignore
- test/my_address_file_geocoded.csv
- LICENSE GPL license
- .github/workflows/build-deploy-pr.yaml
- .github/workflows/build-deploy-release.yaml

Usage

```
use_degauss_container(geomarker = getwd(), version = "0.1.0", ...)
```

```
use_degauss_dockerfile(geomarker = getwd(), version, ...)
```

```
use_degauss_makefile(geomarker = getwd(), ...)
```

```
use_degauss_readme(geomarker = getwd(), version = "0.1.0", ...)
```

```
use_degauss_github_readme_rmd(geomarker = getwd(), ...)
```

```
use_degauss_entrypoint(geomarker = getwd(), version = "0.1.0", ...)
```

```
use_degauss_dockerignore(geomarker = getwd(), ...)
```

```
use_degauss_tests(geomarker = getwd(), ...)
```

```
use_degauss_license(geomarker = getwd(), ...)
```

```
use_degauss_github_actions(geomarker = getwd(), ...)
```


Arguments

geomarker	path to folder where DeGAUSS container files are to be added; defaults to the current working directory
version	string of version number used in freshly created README and entrypoint.R; defaults to "0.1.0"
...	arguments passed to render_degauss_template (overwrite)

write_geomarker_file *write geomarker output to file*

Description

write geomarker output to file

Usage

```
write_geomarker_file(
  d,
  raw_data = NULL,
  filename,
  geomarker_name = Sys.getenv("degauss_name"),
  version = Sys.getenv("degauss_version"),
  argument = NULL
)
```

Arguments

d	input nest on .row with added geomarker column(s)
raw_data	original unnested input data, defaults to NULL (for use when nest_df = FALSE in read_lat_lon_csv)
filename	name of input file, probably opt\$filename if inside container
geomarker_name	name of the geomarker; defaults to degauss environment variable degauss_name
version	container version number as a character string; defaults to degauss environment variable degauss_version
argument	optional information to append after the image version number that was specified using a degauss argument; for example, a selected buffer radius, care site, or geocoding threshold

Value

output file is written to working directory

Examples

```
## Not run:
write_geomarker_file(d$d, d$raw_data,
  filename = "test/my_address_file_geocoded.csv",
  geomarker = "roads", version = "0.4"
)

## End(Not run)
```

Index

address_is_institutional, [2](#)
address_is_nonaddress, [3](#)
address_is_po_box, [3](#)

check_dates, [4](#)
check_end_after_start_date, [4](#)
check_for_column, [5](#)
check_ram, [6](#)
clean_address, [6](#)
core_lib_images, [7](#)
create_degauss_menu_data, [7](#)

degauss_colors, [8](#)
degauss_menu, [8](#)
degauss_run, [9](#)

expand_dates, [9](#)

find_docker, [10](#)

get_degauss_core_lib_env, [10](#)
get_degauss_env_dockerfile, [11](#)
get_degauss_env_online
 ([get_degauss_env_dockerfile](#)),
 [11](#)
greeting, [12](#)

has_docker, [13](#)

make_degauss_command, [13](#)

qlibrary, [14](#)

read_lat_lon_csv, [14](#)

use_degauss_compose, [15](#)
use_degauss_container, [16](#)
use_degauss_dockerfile
 ([use_degauss_container](#)), [16](#)
use_degauss_dockerignore
 ([use_degauss_container](#)), [16](#)

use_degauss_entrypoint
 ([use_degauss_container](#)), [16](#)
use_degauss_github_readme_rmd
 ([use_degauss_container](#)), [16](#)
use_degauss_github_actions
 ([use_degauss_container](#)), [16](#)
use_degauss_license
 ([use_degauss_container](#)), [16](#)
use_degauss_makefile
 ([use_degauss_container](#)), [16](#)
use_degauss_readme
 ([use_degauss_container](#)), [16](#)
use_degauss_tests
 ([use_degauss_container](#)), [16](#)

write_geomarker_file, [17](#)